UNIVERSITÉ
DE MONTPELLIER

TP1 : Régressions régularisées

Analyse Multivariée

Rédigé par :
GOUJILI Nouhaila CORDOVAL Chloë
- Master 1 MIND

2020/2021

1 Introduction :

L'objetif de ce TP est de présenter une introduction aux méthodes régularisées.

Pour cela on s'intéresse aux données des logements dakarois figurant dans le fichier "*LogtsDK*.csv" ; qui représente les variables de taille, de qualité et les variables de situation. Nous en bénéficierons de ces trois thèmes pour modéliser le loyer le mieux possible.

On commencera dans un premier temps par effectuer une régréssion sur composantes principales (ACP) de chaque thème ; on modélisera le loyer à partir des composantes des trois thèmes explicatifs.

En second lieu, nous allons réaliser une classification sur les variables afin de déterminer une bonne partition. On va simuler le loyer à partir de ces composantes et retrouver les coefficients des variables originelles dans le prédicteur linéaire.

Au final, on utilisera la régression PLS pour déterminer le meilleur nombre de composantes, et retrouver ainsi ; les coefficients des variables originelles dans le prédicteur linéaire. En revanche, nous affectuerons les régressions ridge, LASSO et en conséquence, nous allons comparer l'ajustement et les coefficients des modèles obtenus avec ceux des méthodes principales.

En dernier lieu, on tranchera quels sont les avantages et inconvénients que nous avons pu reconnaître des différentes méthodes et quel modèle nous allons prendre en compte en définitive.

2 Importation des données :

Nous travaillerons sur une base de données portant sur des logements dakarois, outre les variables de taille et de qualité, ainsi que les variables de situation. L'objectif est de prédire et de modéliser le loyer *le mieux possible*.

Chargement des librairies :

```
library(ade4) #Librairie qui permet l'implentation de fonctions statistiques
#et graphiques
library(FactoMineR) #Il permet de réaliser des analyses classiques telles que
l'analyse en composantes principales (ACP), l'analyse des correspondances (AC)
l'analyse des correspondances multiples (ACM) ainsi que des analyses plus
avancées.
library(glmnet) #Permet d'ajuster l'ensemble du chemin de régularisation lasso
ou élastique-net pour la régression linéaire
library(corrplot)#Permet de visualiser une matrice de corrélation par
corrélogramme
```

Lecture des données :

```
library(readr)
logtsDK <- read.delim("logtsDK.csv")#Pour importer cet ensemble de données
```

Transformation des variables qualitatives en indicatrices :

Les variables de la colonne 13 à 29 représentent des variables qualitatives. Pour pouvoir les exploiter dans le futur, nous devons les transformer en indicatrices.

```
logtsDK_fact <- logtsDK[,13:29]
logtsDKnomIndic <- acm.disjonctif(logtsDK_fact)#Tableau disjonctif complet
```

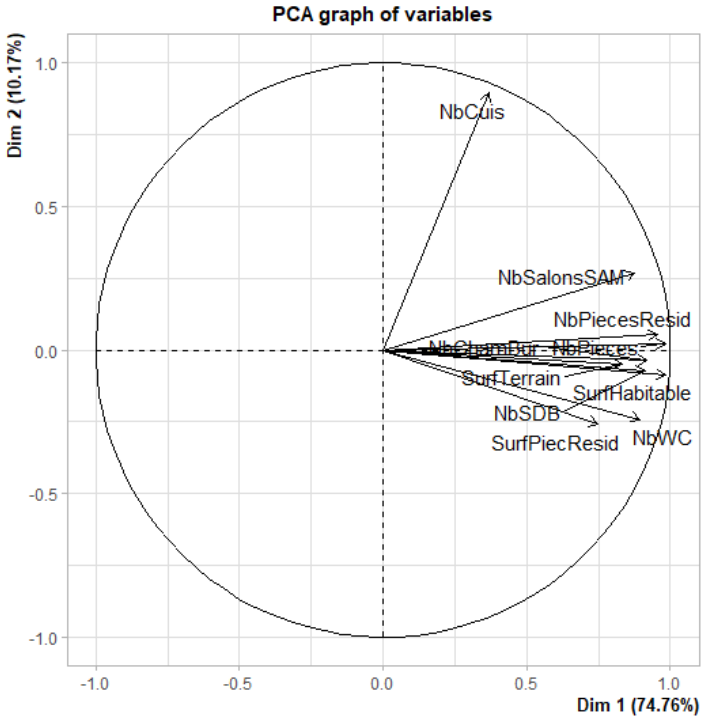
3 Régression sur composantes principales :

Le modèle linéaire formule la variable y (loyer) comme la somme de la combinaison linéaire $X\beta$ avec X le groupe des variables de taille, qualité et situation (variables explicatives) ; et d'un bruit ϵ et il s'agit de minimiser la norme carrée de ϵ.

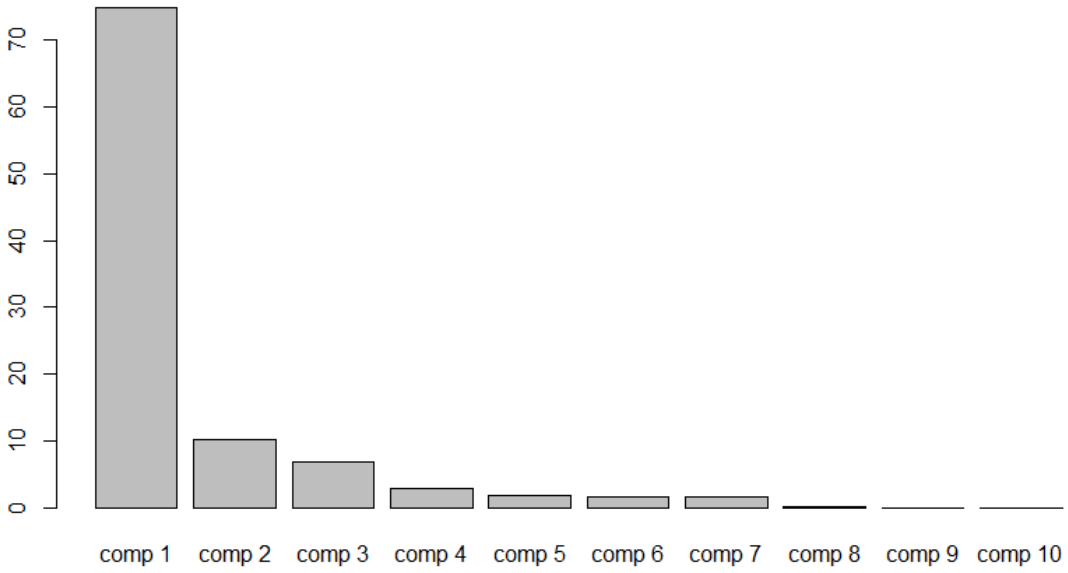
3.1 ACP de chaque thème (selon la nature de ses variables) :

Nous allons réaliser une ACP de chaque thème, et comme les deux thèmes qualité et situation désignent des variables qualitatives on va procéder par une ACM sur ces deux thèmes et on commencera par faire une regression sur les composantes retenues pour chaque thème. Ensuite nous allons modéliser le loyer sur tous les thèmes simultanément en éliminant les composantes qui n'ont pas de rôle significatif.

Pour le **Thème Taille :** on trouve le le graphique suivant :



Cette image n'a retenu de X (ici groupe de variables de taille) que l'information utile pour la prédiction du loyer et on peut s'en servir pour procéder dans la suite à une classification automatique afin de simplifier la description de ce nuage.



On remarque un décroissement des valeurs propres et un décrochement assez fort entre la première valeur propre et la deuxième mais on ne capte pas assez d'inertie, ainsi un décrochement entre la valeur propre 5 et la sixième. Comme le sous espace de dimension 5 capture 96.5% de l'inertie des variables de taille, on pourrait se dire qu'il est suffisant de conserver que les 5 premières composantes principales. Si on l'avait conserver la sixième composante que l'on pensait de peu importante on aurait obtenu un R^2 plus élevé.

L'ajustement du modèle linéaire de la variable y (loyer) sur les composantes des variables de taille a donc besoin de cette sixième composante. Le tout est de savoir si la sixième composante représente effectivement du bruit ou simplement une information un peu marginale. Même si la composante 6 n'est pas approprement parlé du bruit, elle est de rang assez élevé et risque d'être fort peu intérprétable.

En extrayant de X des composantes utiles pour régresser y (le loyer) dessus : Il s'agit de modéliser et prédire linéairement le loyer à partir des variables de qualité, de taille et de situation.

Après une régression sur composantes principales des variables de taille ; on remarque que les composantes principales les plus utiles sont la première, et la dernière.

```
Call:
lm(formula = logtsDK$Loyer ~ PCA1$ind$coord, data = as.data.frame(logtsDK))

Residuals:
    Min      1Q  Median      3Q     Max
-436.25 -184.03  -10.35  121.94  908.92

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)            356.05      40.87   8.712 2.75e-10 ***
PCA1$ind$coordDim.1    117.40      14.95   7.854 3.13e-09 ***
PCA1$ind$coordDim.2    -35.28      40.52  -0.871   0.3899
PCA1$ind$coordDim.3     78.97      49.19   1.605   0.1174
PCA1$ind$coordDim.4   -105.36      75.84  -1.389   0.1735
PCA1$ind$coordDim.5   -260.68      96.66  -2.697   0.0107 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 261.7 on 35 degrees of freedom
Multiple R-squared:  0.6796,    Adjusted R-squared:  0.6338
F-statistic: 14.85 on 5 and 35 DF,  p-value: 7.961e-08
```

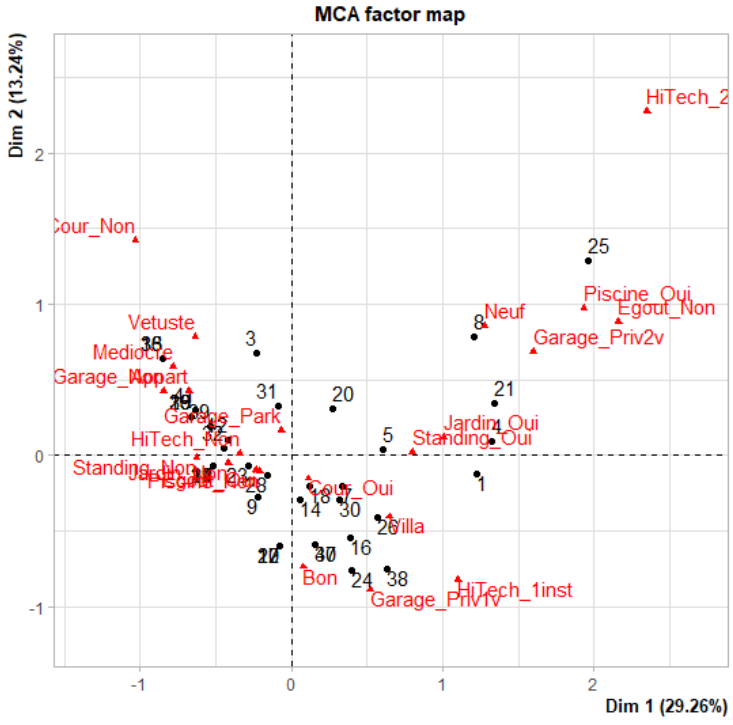
Par conséquence, nous allons effectuer une MCO que sur les deux composantes 1 et 5 :

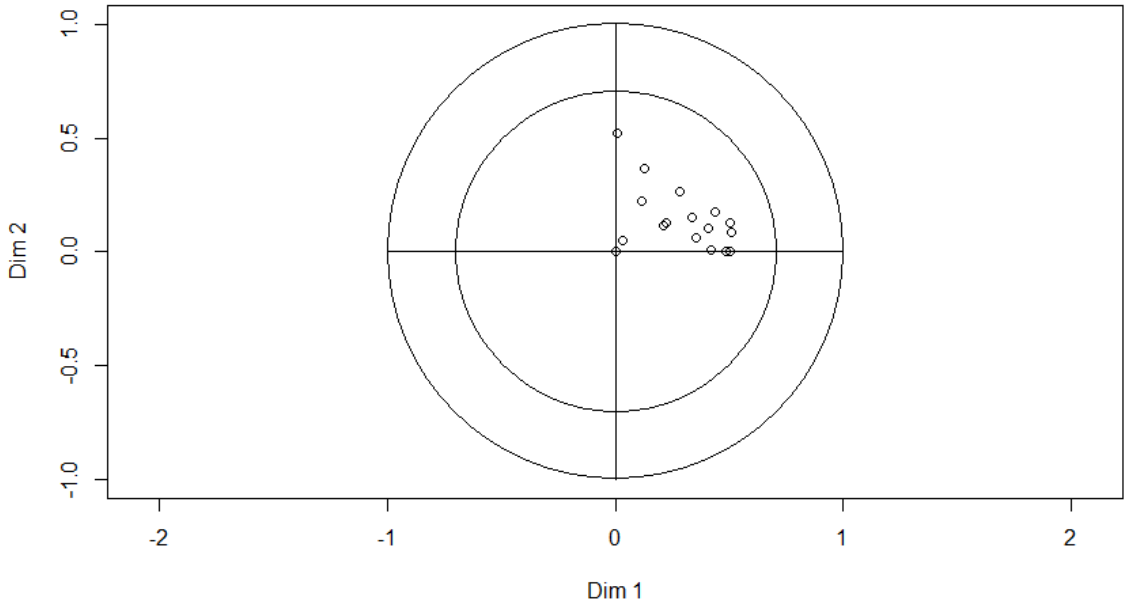
```
Call:
lm(formula = logtsDK$Loyer ~ PCA1$ind$coord[, c(1, 5)], data = as.data.frame(logtsDK))

Residuals:
    Min      1Q  Median      3Q     Max
-471.72 -166.46   -2.39  128.45  865.44

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                     356.05      42.07   8.463 2.84e-10 ***
PCA1$ind$coord[, c(1, 5)]Dim.1  117.40      15.39   7.630 3.51e-09 ***
PCA1$ind$coord[, c(1, 5)]Dim.5 -260.68      99.50  -2.620   0.0126 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 269.4 on 38 degrees of freedom
Multiple R-squared:  0.6314,    Adjusted R-squared:  0.612
F-statistic: 32.54 on 2 and 38 DF,  p-value: 5.826e-09
```

On trouve un $R^2 = 0.63$. En effet ; notre groupe C de composante ; $C = [c^1, c^5]$ est de la forme : $X[u_1, u_5]$ où les u sont des vecteurs de la base duale des variables. D'où : $\hat{y} = C\hat{\gamma}$.

C'est plus intéressant de faire la régression sur les 2 composantes que sur une seule. Ainsi, y'a pas de confusion entre les composantes ; c'est la vraie-significativité.

Faisons maintenant l'ACM pour **Thème Qualité :**



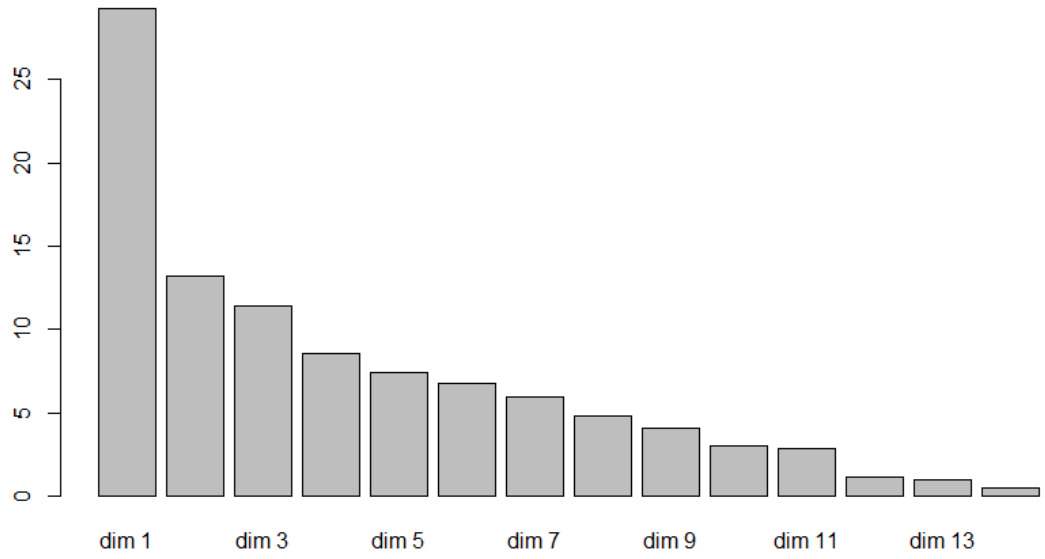


Les variables de qualité sont bien représentées par la dimension 1 .

La faible part de variance expliquée par les deux premiers axes est une caractéristique de l'ACM (qui donne généralement des mesures pessimistes de l'information extraite). Les résultats de cette analyse ne devront pas être pris pour l'absolu ; cependant notre analyse nous permettra de dégager un profil des logements selon leurs caractéristiques.

Affichons les valeurs propres de l'ACM pour le thème qualité :

```
> acm1$eig
        eigenvalue percentage of variance cumulative percentage of variance
dim 1  0.455103199             29.2566342                           29.25663
dim 2  0.205879007             13.2350790                           42.49171
dim 3  0.177207061             11.3918825                           53.88360
dim 4  0.133360905              8.5732010                           62.45680
dim 5  0.115060789              7.3967650                           69.85356
dim 6  0.104964086              6.7476912                           76.60125
dim 7  0.092226129              5.9288226                           82.53008
dim 8  0.074637114              4.7981002                           87.32818
dim 9  0.063820044              4.1027171                           91.43089
dim 10 0.047270799              3.0388371                           94.46973
dim 11 0.044941295              2.8890833                           97.35881
dim 12 0.018169704              1.1680524                           98.52687
dim 13 0.015022607              0.9657390                           99.49260
dim 14 0.007892816              0.5073953                          100.00000
```



On remarque de même ; un décrochement entre la valeur propre 5 et la sixième. Le sous espace de dimension 5 capte environ 69.85% de l'inertie des variables de qualité. Et qu'à partir de la composante 14 qu'on capte environ 100% de l'inertie ; on retrouve tout l'espace des variables de qualité.

Nous effectuerons une régression sur les composantes principales, et on trouve les résultats suivants :

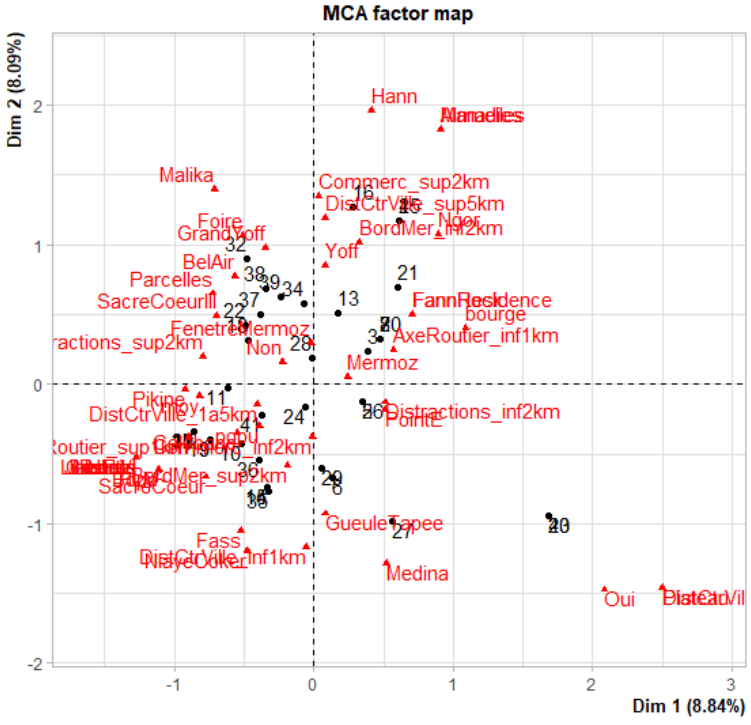
```
lm(formula = logtsDK$Loyer ~ acm1$ind$coord, data = as.data.frame(logtsDK))

Residuals:
    Min      1Q  Median      3Q     Max
-570.24 -116.31  -29.31   60.72  972.12

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)           356.05      43.18   8.246 1.02e-09 ***
acm1$ind$coordDim 1   467.12      64.00   7.298 1.58e-08 ***
acm1$ind$coordDim 2   103.82      95.16   1.091   0.2827
acm1$ind$coordDim 3  -102.86     102.57  -1.003   0.3228
acm1$ind$coordDim 4  -270.32     118.23  -2.286   0.0284 *
acm1$ind$coordDim 5  -187.74     127.29  -1.475   0.1492
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 276.5 on 35 degrees of freedom
Multiple R-squared:  0.6424,    Adjusted R-squared:  0.5913
F-statistic: 12.57 on 5 and 35 DF,  p-value: 5.037e-07
```

On observe que les composantes principales les plus utiles sont la première et la quatrième ; d'où nous allons faire une MCO que sur ces deux composantes et on trouve :

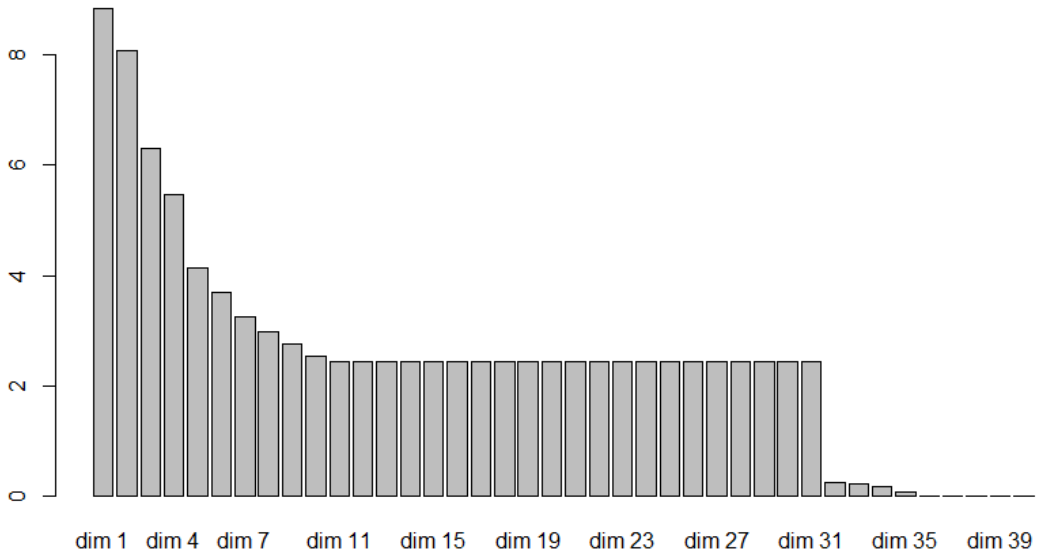
```
Call:
lm(formula = logtsDK$Loyer ~ acm1$ind$coord[, c(1, 4)], data = as.data.frame(logtsDK))

Residuals:
    Min      1Q  Median      3Q     Max
-514.92 -135.88  -45.69   52.96  986.92

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                     356.05      43.95   8.101 8.39e-10 ***
acm1$ind$coord[, c(1, 4)]Dim 1  467.12      65.15   7.170 1.45e-08 ***
acm1$ind$coord[, c(1, 4)]Dim 4 -270.32     120.35  -2.246   0.0306 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 281.4 on 38 degrees of freedom
Multiple R-squared:  0.5977,    Adjusted R-squared:  0.5765
F-statistic: 28.23 on 2 and 38 DF,  p-value: 3.066e-08
```

Pour le R^2, on constate qu'on a perdu de l'information, mais on a bien la première et la quatrième composante qui jouent un rôle significatif.

Pour le **Thème Situation :**



On trace maintenant un histogramme des valeurs propres de l'ACM sur les variables de situation :



On effectuerons une régression sur les composantes afin de savoir celle les plus utiles.

```
Call:
lm(formula = logtsDK$Loyer ~ acm2$ind$coord, data = as.data.frame(logtsDK))

Residuals:
    Min      1Q  Median      3Q     Max
-563.21 -100.14   -6.09   83.91 1266.41

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)            356.05      48.79   7.297 1.59e-08 ***
acm2$ind$coordDim 1    385.35      72.49   5.316 6.17e-06 ***
acm2$ind$coordDim 2    169.63      75.77   2.239   0.0316 *
acm2$ind$coordDim 3    162.69      85.77   1.897   0.0661 .
acm2$ind$coordDim 4   -193.66      92.19  -2.101   0.0430 *
acm2$ind$coordDim 5    -62.80     105.81  -0.594   0.5566
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 312.4 on 35 degrees of freedom
Multiple R-squared:  0.5433,    Adjusted R-squared:  0.478
F-statistic: 8.327 on 5 and 35 DF,  p-value: 2.894e-05
```

Les composantes les plus significatives sont la **première**, la **deuxième** et la **quatrième** composante.

Nous allons maintenant modéliser le loyer sur tous les thèmes simultanément, en éliminant les composantes qui n'ont pas de rôle significatif.

```
Call:
lm(formula = logtsDK$Loyer ~ PCAT$ind$coord, data = as.data.frame(logtsDK))

Residuals:
    Min      1Q  Median      3Q     Max
-274.66  -96.46  -31.16   71.71  845.08

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)           356.05      33.83  10.524 2.19e-12 ***
PCAT$ind$coordDim.1    78.17       8.14   9.603 2.42e-11 ***
PCAT$ind$coordDim.2   -63.89      13.96  -4.578 5.70e-05 ***
PCAT$ind$coordDim.3    23.61      15.09   1.565  0.12655
PCAT$ind$coordDim.4   -12.38      15.83  -0.782  0.43931
PCAT$ind$coordDim.5    48.99      17.15   2.856  0.00718 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 216.6 on 35 degrees of freedom
Multiple R-squared:  0.7804,    Adjusted R-squared:  0.749
F-statistic: 24.88 on 5 and 35 DF,  p-value: 1.295e-10
```

On trouve les coefficients de la modélisation du loyer à partir des thèmes simultanément :

(Intercept)	double [1]	356.0488
PCATindcoordDim.1	double [1]	78.17106
PCATindcoordDim.2	double [1]	-63.89304
PCATindcoordDim.3	double [1]	23.60978
PCATindcoordDim.4	double [1]	-12.38333
PCATindcoordDim.5	double [1]	48.98559

On observe que les composantes qui sont utiles sont la première, la deuxième et la cinquième composante. D'où, nous allons faire une MCO que sur ces trois composantes.

```
Call:
lm(formula = logtsDK$Loyer ~ PCAT$ind$coord[, c(1, 2, 5)], data = as.data.frame(logtsDK))

Residuals:
    Min      1Q  Median      3Q     Max
-288.35  -99.20  -28.09   59.82  885.65

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                      356.049     34.315  10.376 1.66e-12 ***
PCAT$ind$coord[, c(1, 2, 5)]Dim.1   78.171      8.256   9.468 1.99e-11 ***
PCAT$ind$coord[, c(1, 2, 5)]Dim.2  -63.893     14.155  -4.514 6.26e-05 ***
PCAT$ind$coord[, c(1, 2, 5)]Dim.5   48.986     17.399   2.815  0.00776 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 219.7 on 37 degrees of freedom
Multiple R-squared:  0.7612,    Adjusted R-squared:  0.7418
F-statistic: 39.31 on 3 and 37 DF,  p-value: 1.356e-11
```

D'après le graphique ci-dessus, on remarque qu'on a perdu de l'information lors de cette régression, mais on s'aperçoit que notre choix est bon, pour la régression que sur ces composantes. En effet, toutes ces composantes sont significatives dans notre modèle.

On retrouve les coefficients des variables après la régression sur la première, deuxième et cinquième composante :

```
> coef(MCO_T)
                      (Intercept) PCAT$ind$coord[, c(1, 2, 5)]Dim.1 PCAT$ind$coord[, c(1, 2, 5)]Dim.2
                        356.04878                          78.17106                          -63.89304
PCAT$ind$coord[, c(1, 2, 5)]Dim.5
                         48.98559
```

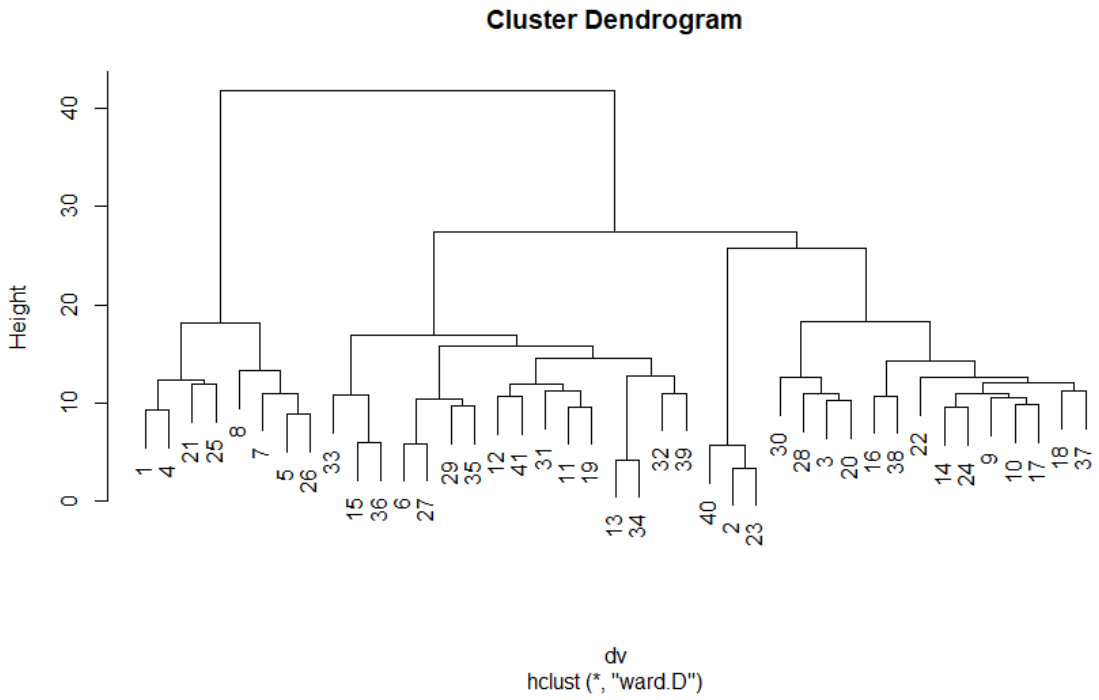
Dans la suite, nous allons modéliser le loyer à partir des composantes des trois thèmes explicatifs retenues précedemment, et on trouve les résultats suivants :

```
lm(formula = logtsDK$Loyer ~ PCA1$ind$coord[, 1] + acm1$ind$coord[,
    c(1, 4)] + acm2$ind$coord[, c(1, 2, 4)], data = as.data.frame(logtsDK))

Residuals:
    Min      1Q  Median      3Q     Max
-276.59  -93.06  -29.76   58.13  824.08

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                       356.05      33.69  10.568 2.78e-12 ***
PCA1$ind$coord[, 1]                69.86      24.47   2.854 0.007294 **
acm1$ind$coord[, c(1, 4)]Dim 1    200.72     106.91   1.877 0.069057 .
acm1$ind$coord[, c(1, 4)]Dim 4   -149.82      94.85  -1.580 0.123458
acm2$ind$coord[, c(1, 2, 4)]Dim 1  223.56      56.14   3.982 0.000341 ***
acm2$ind$coord[, c(1, 2, 4)]Dim 2 -109.09      69.30  -1.574 0.124691
acm2$ind$coord[, c(1, 2, 4)]Dim 4  -57.35      67.01  -0.856 0.398085
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 215.7 on 34 degrees of freedom
Multiple R-squared:  0.7885,    Adjusted R-squared:  0.7511
```

On obtient un R^2 de 0.7885, et on constate que les composantes utiles sont la première composante qu'on a déja retenu pour le thème taille, ainsi que la première composante du thème situation.

3.2 Classification sur variables :

On procède par une classifiction hiérarchique sur les variables avec l'indice WARD, pour objectif d'avoir un gain minimum d'inertie intra-classe à chaque agrégation ; et on obtient le dendrogramme de la classification ci-dessous :



Sur celui-ci, on remarque 4 zones de décroissance rapide, déterminant des classes entre lesquelles des différences sont significatives :

- Partition de 2 classes
- Partition de 3 classes
- Partition de 4 classes
- Partition de 7 classes

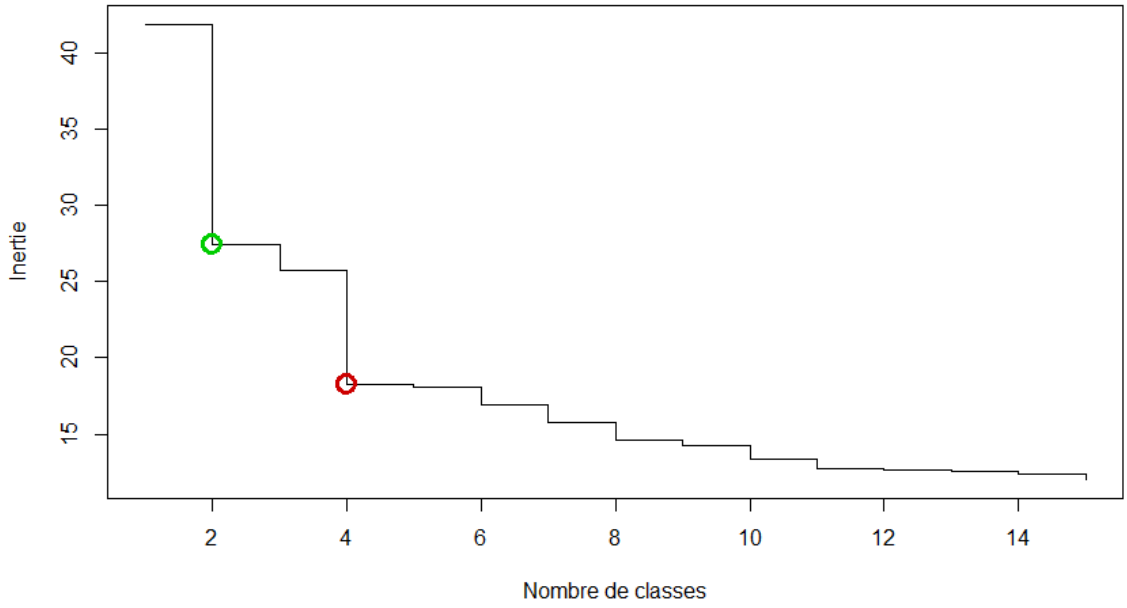
A partir de ces quatre possibilités, il s'agit d'évaluer laquelle des partitions est non seulement optimale en terme statistique, mais aussi et surtout la plus **pertinente** pour l'analyse.

Pour déterminer une bonne partition, nous allons calculer le R^2 de chacune.

On présente ci-dessous un tableau qui désigne ces indicateurs de chaque ensemble de parties, afin de choisir le bon, c'est-à-dire celui qui convient au bon nombre de partition.

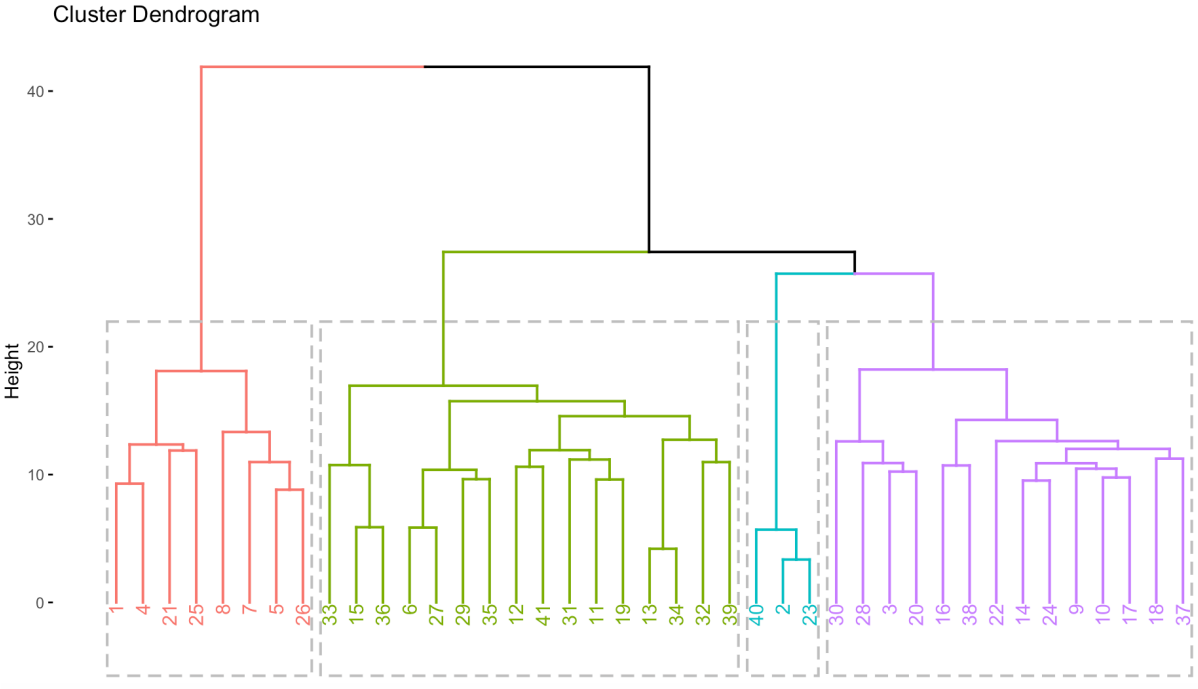
Nombre de classes	R^2de la partition
2 classes	0.2215363
3 classes	0.2802125
4 classes	0.3256438
7 classes	0.4389535

On regarde dans la suite ; l'évolution de l'inertie en fonction de nombres de classes :



On voit ici qu'une découpe en 4 catégories est pertinent (saut à 4 catégories) : en rouge.

On remarque de suite que la partition en 4 classes répond le mieux à l'exigence de l'indicateur d'inertie qui est égale à **0.3256438**.



Dans la suite, on regroupe les variables de chaque classe dans des grappes (Clusters), et on procède par une analyse en composantes principales (ACP) pour chaque grappe.

Ainsi on modélise le loyer à partir des composantes de chaque Cluster, et on trouve les résultats suivants :

```
lm(formula = log(logtsDK[, 2]) ~ ACP1$ind$coord[, 1] + ACP2$ind$coord[,
    1] + ACP3$ind$coord[, 1] + ACP4$ind$coord[, 1])

Residuals:
    Min      1Q  Median      3Q     Max
-1.0562 -0.4418 -0.1076  0.3491  1.7565

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)           5.2203     0.1047  49.863  < 2e-16 ***
ACP1$ind$coord[, 1]   0.3714     0.1067   3.482  0.00132 **
ACP2$ind$coord[, 1]  -0.1271     0.1423  -0.893  0.37767
ACP3$ind$coord[, 1]  -0.1450     0.1860  -0.779  0.44080
ACP4$ind$coord[, 1]   0.3075     0.1546   1.989  0.05435 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6704 on 36 degrees of freedom
Multiple R-squared:  0.7157,    Adjusted R-squared:  0.6841
F-statistic: 22.66 on 4 and 36 DF,  p-value: 2.043e-09
```

On observe que les composantes qui sont utiles et qu'on peut prendre en compte sont les composantes de la première ACP faite sur la première et la quatrième classe. On refait notre modélisation que sur ces deux composantes, et on obtient les résultats ci-desous :

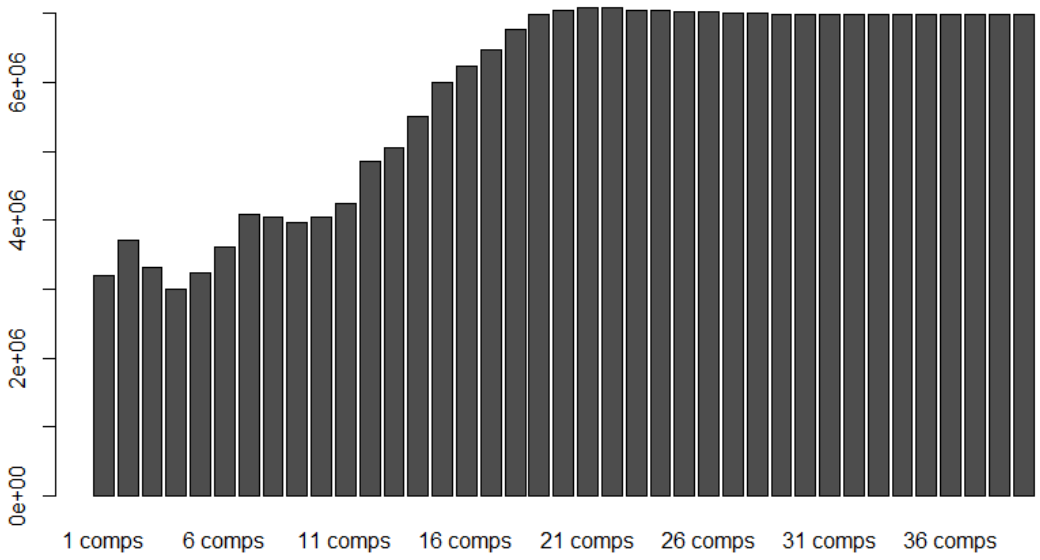
```
Call:
lm(formula = log(logtsDK[, 2]) ~ ACP1$ind$coord[, 1] + ACP4$ind$coord[,
    1])

Residuals:
    Min      1Q  Median      3Q     Max
-1.1164 -0.4108 -0.1797  0.3811  1.8409

Coefficients:
                    Estimate Std. Error t value Pr(>|t|)
(Intercept)           5.2203     0.1035   50.42  < 2e-16 ***
ACP1$ind$coord[, 1]   0.2995     0.0803    3.73 0.000624 ***
ACP4$ind$coord[, 1]   0.1679     0.0827    2.03 0.049376 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.663 on 38 degrees of freedom
Multiple R-squared:  0.7065,    Adjusted R-squared:  0.6911
F-statistic: 45.74 on 2 and 38 DF,  p-value: 7.658e-11
```

De même, on a perdu de l'information, mais on constate que les composantes de la première et la quatrième classe qui comptent le plus et qui sont significatives.

L'intérêt de cette classification sur variables, c'est d'avoir des composantes qui sont peu corrélées entre elles ; on a des composantes qui sont facilement interprétables, parce que chacune est interprétable à son cluster. Et ensuite, elles sont exogènes ; en effet, on a fabriqué ces composantes sans regarder la variable réponse y (loyer).

4 Régression PLS :

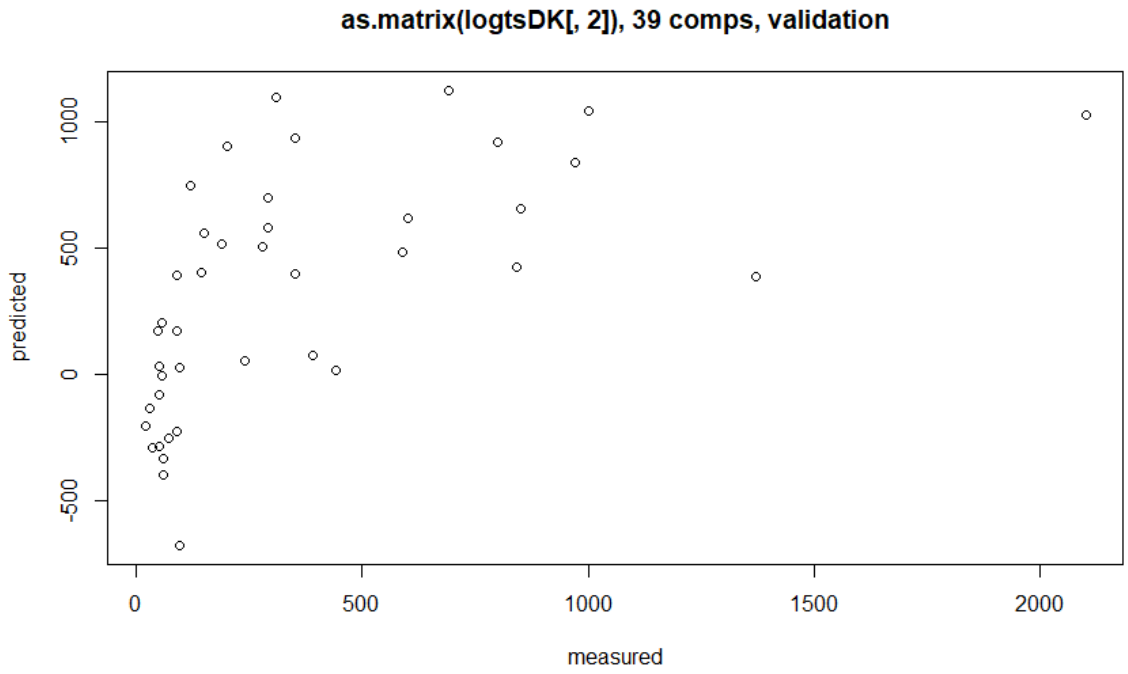
Nous allons faire une régression PLS du loyer et déterminer combien de composantes prédictives qu'on doit retenir.



On remarque que la quatrième composante donne une meilleure performance, ensuite il y a une détérioration. En effet, il s'agit d'une erreur quadratique de prédiction qui est supérieure.

Dès qu'on aborde les autres composantes on est dans le sur-ajustement.

Ainsi, on regarde la prédiction selon le nombre de composantes retenues :



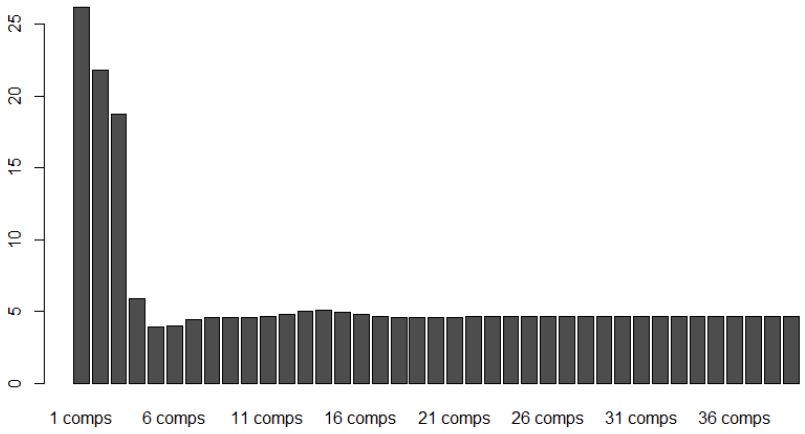
Lorsque l'on retient toutes les composantes, il n'y a pas de correspondance entre les variables mesurées et les variables prédites.

Donc dans la suite, on retient la seule première composante et on fait la prédiction. Ainsi on trouve un R^2 qui est égale à 0.64.

Afin de diminuer la variable y (ici le loyer) nous allons le diminuer en prenant son logarithme puis on refait la régression :

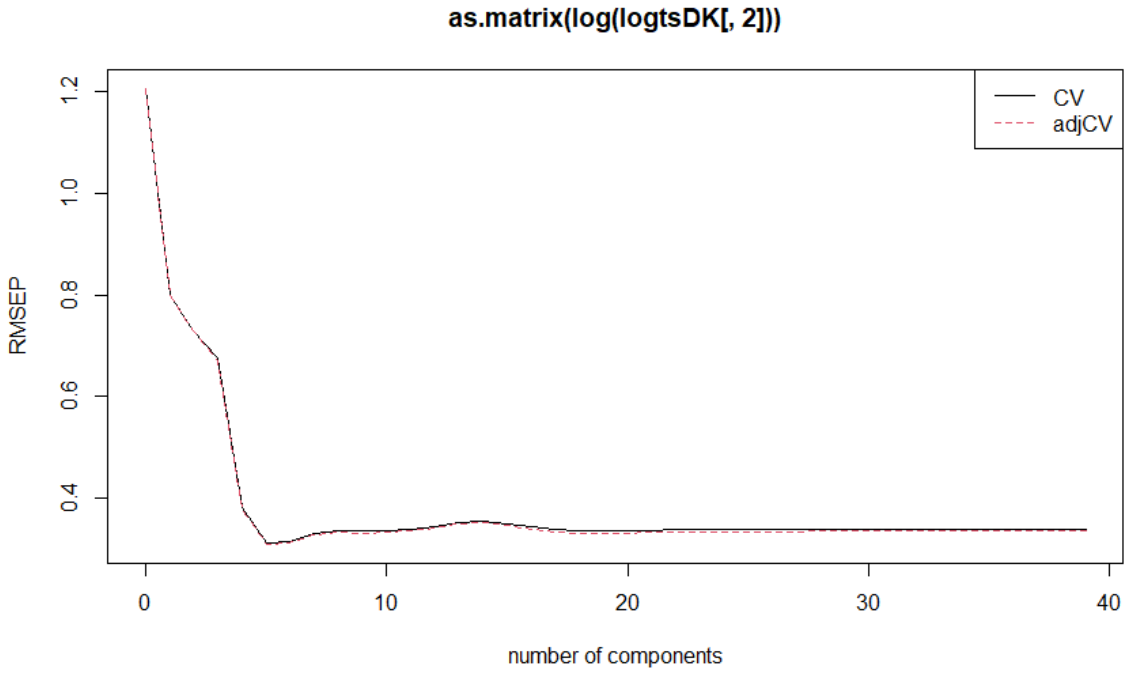
On présente ci-dessous, l'erreur quadratique moyenne de la prédiction (**PRESS**) :

```
> LDKLogYpls$validation$PRESS
                              1 comps  2 comps  3 comps 4 comps  5 comps  6 comps  7 comps  8 comps 9 comps 10 comps
as.matrix(log(logtsDK[, 2])) 26.17922 21.84084 18.75476 5.93485 3.926385 3.992414 4.421394 4.618986 4.56565 4.606414
                             11 comps 12 comps 13 comps 14 comps 15 comps 16 comps 17 comps 18 comps 19 comps 20 comps
as.matrix(log(logtsDK[, 2])) 4.673105 4.832199 5.068051 5.141585 4.992089 4.818286 4.634387 4.581417 4.571174 4.573765
                             21 comps 22 comps 23 comps 24 comps 25 comps 26 comps 27 comps 28 comps 29 comps 30 comps
as.matrix(log(logtsDK[, 2])) 4.601006 4.648525 4.660397 4.659966 4.662864 4.663757 4.667487 4.669187 4.669508 4.669568
                             31 comps 32 comps 33 comps 34 comps 35 comps 36 comps 37 comps 38 comps 39 comps
as.matrix(log(logtsDK[, 2])) 4.669651 4.669768 4.669744 4.669743 4.669741 4.669742 4.669742 4.669742 4.669742
```



D'après les graphiques ci-dessus, on peut retenir 5 composantes ; le passage de la quatrième à la cinquième composante fait gagner de l'information mais pas énormément.

Le graphique ci-dessous désigne la racine carrée de l'erreur moyenne quadratique par rapport aux nombres de composantes :



On s'aperçoit qu'on avait fait le bon choix pour les 5 composantes qu'on a retenu. Rajouter les quatres composantes pour arriver à la cinquième ça nous fait gagner en prédiction.

On peut le vérifier avec des valeurs précises, en utilisant la **fonction summary()** qui permet d'avoir la description statistique d'une variable ou d'une table de données.

```
VALIDATION: RMSEP
Cross-validated using 41 leave-one-out segments.
       (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps  10 comps  11 comps
CV           1.208   0.7991   0.7299   0.6763   0.3805   0.3095   0.3121   0.3284   0.3356   0.3337    0.3352    0.3376
adjCV        1.208   0.7985   0.7294   0.6696   0.3786   0.3074   0.3107   0.3263   0.3323   0.3303    0.3319    0.3340
       12 comps  13 comps  14 comps  15 comps  16 comps  17 comps  18 comps  19 comps  20 comps  21 comps  22 comps
CV       0.3433    0.3516    0.3541    0.3489    0.3428    0.3362    0.3343    0.3339    0.3340    0.3350    0.3367
adjCV    0.3392    0.3474    0.3498    0.3447    0.3386    0.3321    0.3302    0.3298    0.3299    0.3309    0.3326
       23 comps  24 comps  25 comps  26 comps  27 comps  28 comps  29 comps  30 comps  31 comps  32 comps  33 comps
CV       0.3371    0.3371    0.3372    0.3373    0.3374    0.3375    0.3375    0.3375    0.3375    0.3375    0.3375
adjCV    0.3330    0.3330    0.3331    0.3331    0.3333    0.3333    0.3333    0.3333    0.3333    0.3333    0.3333
       34 comps  35 comps  36 comps  37 comps  38 comps  39 comps
CV       0.3375    0.3375    0.3375    0.3375    0.3375    0.3375
adjCV    0.3333    0.3333    0.3333    0.3333    0.3333    0.3333
```

```
TRAINING: % variance explained
                              1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps  8 comps  9 comps  10 comps
X                               94.21    99.95    99.96    99.99    99.99    99.99    99.99    99.99    99.99     99.99
as.matrix(log(logtsDK[, 2]))    59.22    65.68    85.60    94.54    97.49    98.14    98.72    99.39    99.57     99.68
                              11 comps  12 comps  13 comps  14 comps  15 comps  16 comps  17 comps  18 comps  19 comps
X                                99.99     99.99    100.00    100.00    100.00    100.00    100.00    100.00    100.00
as.matrix(log(logtsDK[, 2]))     99.82     99.91     99.95     99.97     99.98     99.98     99.99     99.99     99.99
                              20 comps  21 comps  22 comps  23 comps  24 comps  25 comps  26 comps  27 comps  28 comps
X                                  100       100       100       100       100       100       100       100       100
as.matrix(log(logtsDK[, 2]))       100       100       100       100       100       100       100       100       100
                              29 comps  30 comps  31 comps  32 comps  33 comps  34 comps  35 comps  36 comps  37 comps
X                                  100       100       100       100       100       100       100       100       100
as.matrix(log(logtsDK[, 2]))       100       100       100       100       100       100       100       100       100
                              38 comps  39 comps
X                                  100       100
as.matrix(log(logtsDK[, 2]))       100       100
```

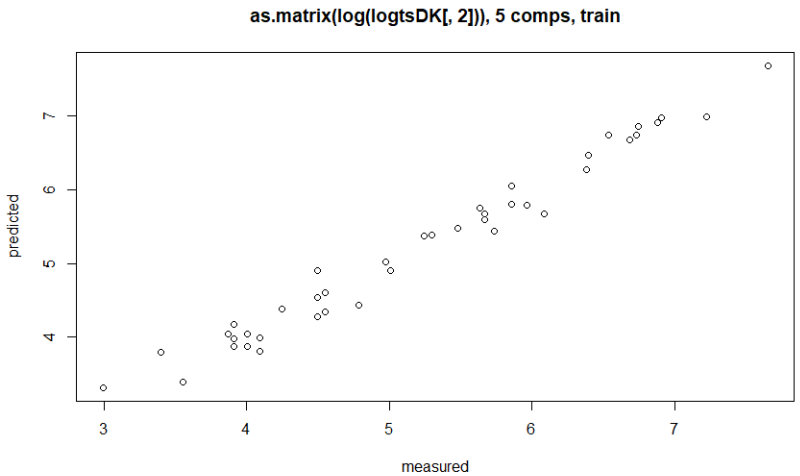
Dans le premier graphique ci-dessus, la ligne « CV » représente la validation croisée (Cross Validation).

Ses valeurs correspondent à la "racine carrée de l'erreur quadratique moyenne de prédiction".

On observe que la valeur la plus faible est effectivement obtenue pour 5 composantes (0.3095). Cela signifie que les erreurs de prédiction les plus faibles seront obtenues en utilisant 5 composantes pour la régression.

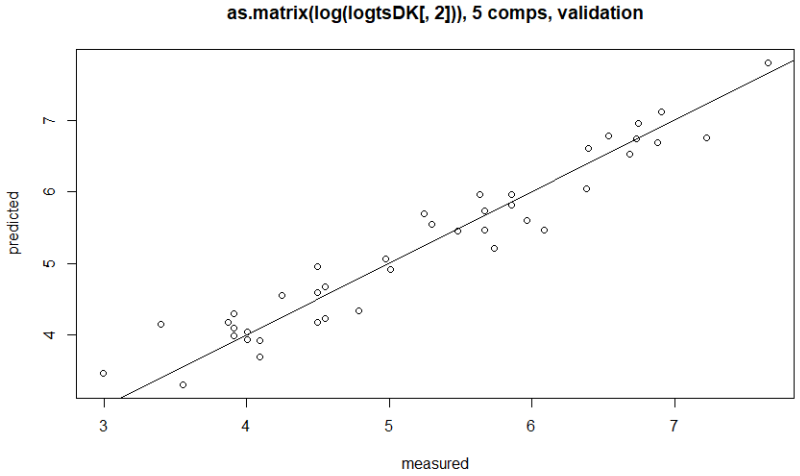
Le deuxième graphique lui, représente les valeurs des pourcentages de variance expliquée pour le modèle. On a 97.49 % de la variance expliquée par le modèle à cinq composantes.

Nous allons maintenant faire la prédiction avec les cinq composantes :



Ainsi, on retrouve bien le $R^2 = 0.9749$ qui est bien meilleur.

Concernant l'allure de notre graphe, on remarque bien une droite (cf graphique ci-dessous). De plus, au fur et à mesure que la variable y qui représente le loyer augmente, il y a homogénéité des écarts entre les autres variables.



Dans la suite, on regarde la corrélation de la variable du loyer avec les autres variables explicatives pour les cinq composantes :

	Comp 1	Comp 2	Comp 3	Comp 4	Comp 5
SurfTerrain	0.990385171	-0.138337005	0.0002398295	-1.663050e-04	1.777622e-06
SurfHabitable	0.899451517	0.437019258	-0.0002748108	-7.809467e-04	3.552860e-04
SurfPiecResid	0.781935836	0.245471091	-0.3225266785	4.706671e-01	-3.051740e-02
NbPieces	0.824551638	0.479147822	0.2209499379	-1.310730e-01	-9.701728e-02
NbPiecesResid	0.774878198	0.531747845	0.1924836334	-2.282014e-01	-4.451819e-02
NbSDB	0.819733245	0.296924813	0.2283418196	1.282359e-02	-2.046653e-01
NbChamBur	0.747040090	0.518401539	0.1499996248	-2.888039e-01	-3.740840e-02
NbSalonsSAM	0.704514616	0.469560994	0.2608879852	-4.135583e-02	-5.378093e-02
NbWC	0.778695118	0.402030140	0.2117711081	9.608673e-02	-1.391414e-01
NbCuis	0.330904755	-0.010502152	0.2421217408	-5.424413e-02	-6.374613e-02
Type.Appart	-0.718821245	-0.413137418	0.1973739471	-6.268795e-02	-5.587752e-02
Type.Villa	0.718821245	0.413137418	-0.1973739471	6.268795e-02	5.587752e-02
Standing.Non	-0.500981659	-0.109997886	-0.4229686883	-3.248913e-01	1.147788e-01
Standing.Oui	0.500981659	0.109997886	0.4229686883	3.248913e-01	-1.147788e-01
Etat.Bon	-0.007531767	0.367046574	0.1712680760	1.549622e-01	3.581194e-01
Etat.Mediocre	-0.276836129	-0.259233126	-0.3557440952	-1.038299e-01	-2.343795e-01
Etat.Neuf	0.438593150	-0.142440935	0.1668706209	-8.364715e-02	-4.983829e-02
Etat.Vetuste	-0.148257063	-0.057650699	0.0354129438	7.773829e-05	-2.166024e-01
Jardin.Non	-0.565733717	0.172829170	-0.1195449375	-2.118147e-01	2.394500e-01
Jardin.Oui	0.565733717	-0.172829170	0.1195449375	2.118147e-01	-2.394500e-01

Cour.Non	-0.274585332	-0.183944205	-0.2041000168	1.007354e-01	-1.761303e-01
Cour.Oui	0.274585332	0.183944205	0.2041000168	-1.007354e-01	1.761303e-01
Piscine.Non	-0.464580446	-0.243198215	0.0694250700	-1.086425e-01	-5.359399e-02
Piscine.Oui	0.464580446	0.243198215	-0.0694250700	1.086425e-01	5.359399e-02
Garage.Non	-0.613058918	-0.337048835	-0.1476295320	-2.527746e-01	3.089216e-01
Garage.Park	-0.151462408	-0.167604062	0.2792691515	1.765326e-01	-4.778903e-01
Garage.Priv1v	0.349468962	0.256302756	-0.1524947227	2.287816e-01	-8.774653e-03
Garage.Priv2v	0.589624226	0.324000393	0.1374726965	-1.354518e-01	6.356304e-02
Egout.Non	0.475953892	0.297408723	0.0206252914	-3.502615e-02	9.494935e-02
Egout.Oui	-0.475953892	-0.297408723	-0.0206252914	3.502615e-02	-9.494935e-02
HiTech.1inst	0.291059952	0.323368977	-0.1178810275	1.222242e-01	5.330116e-02
HiTech.2inst	0.617303193	-0.202415682	0.1505408677	-2.536878e-01	-4.708349e-02
HiTech.Non	-0.595130964	-0.178382636	0.0233084470	2.888423e-02	-2.194580e-02
DistCtrville.0	-0.140677622	-0.027225363	0.5369006564	2.868468e-01	4.720810e-01
DistCtrville.1a5km	0.033989509	-0.100644522	-0.0246627000	7.288228e-02	-2.346497e-01
DistCtrville.inf1km	-0.216815170	-0.258112330	-0.0573307347	-1.218231e-01	1.697618e-01
DistCtrville.sup5km	0.211602778	0.326656033	-0.2356661006	-1.496103e-01	-1.442871e-01
Commerc.inf2km	-0.351804766	-0.188096974	0.2426887341	-9.768146e-02	3.365436e-01
Commerc.sup2km	0.351804766	0.188096974	-0.2426887341	9.768146e-02	-3.365436e-01
BordMer.inf2km	0.387925350	0.224403526	0.1553957317	5.972429e-02	-2.907157e-01
BordMer.sup2km	-0.387925350	-0.224403526	-0.1553957317	-5.972429e-02	2.907157e-01

Nous constatons que la première composante est celle qui compte le plus. En effet, c'est celle que la régression PLS (Partial Least Squares regression) trouve en premier ; il s'agit d'une composante de taille, sauf la variable "NbCuisine" qui est à l'écart.

A partir de la deuxième composante, il est rare de trouver des corrélations qui sont bonnes ; ça risque d'être des composantes correctives qui interviennent pour faire des corrections d'effet de la première composante.

On présente dans le graphique ci-dessous, les coefficients des variables du modèle prédictif fondé sur la première composante :

as.matrix(LDK)SurfTerrain	1.556931e-05
as.matrix(LDK)SurfHabitable	9.179786e-06
as.matrix(LDK)SurfPiecResid	5.055222e-07
as.matrix(LDK)NbPieces	3.904533e-07
as.matrix(LDK)NbPiecesResid	2.376130e-07
as.matrix(LDK)NbSDB	7.014745e-08
as.matrix(LDK)NbChamBur	1.626215e-07
as.matrix(LDK)NbSalonsSAM	7.499150e-08
as.matrix(LDK)NbWC	7.413271e-08
as.matrix(LDK)NbCuis	8.560182e-09
as.matrix(LDK)Type.Appart	-3.480572e-08
as.matrix(LDK)Type.Villa	3.480572e-08
as.matrix(LDK)Standing.Non	-3.615153e-08
as.matrix(LDK)Standing.Oui	3.615153e-08
as.matrix(LDK)Etat.Bon	1.416147e-08
as.matrix(LDK)Etat.Mediocre	-2.447340e-08
as.matrix(LDK)Etat.Neuf	1.473551e-08
as.matrix(LDK)Etat.Vetuste	-4.423580e-09

as.matrix(LDK)Jardin.Non	-2.412981e-08
as.matrix(LDK)Jardin.Oui	2.412981e-08
as.matrix(LDK)Cour.Non	-1.285506e-08
as.matrix(LDK)Cour.Oui	1.285506e-08
as.matrix(LDK)Piscine.Non	-1.366446e-08
as.matrix(LDK)Piscine.Oui	1.366446e-08
as.matrix(LDK)Garage.Non	-3.664136e-08
as.matrix(LDK)Garage.Park	-1.276604e-09
as.matrix(LDK)Garage.Priv1v	1.827677e-08
as.matrix(LDK)Garage.Priv2v	1.964119e-08
as.matrix(LDK)Egout.Non	1.445965e-08
as.matrix(LDK)Egout.Oui	-1.445965e-08
as.matrix(LDK)HiTech.1inst	1.069972e-08
as.matrix(LDK)HiTech.2inst	9.398685e-09
as.matrix(LDK)HiTech.Non	-2.009841e-08
as.matrix(LDK)DistCtrVille.0	8.674606e-09
as.matrix(LDK)DistCtrVille.1a5km	-1.573619e-09
as.matrix(LDK)DistCtrVille.inf1km	-1.128233e-08

as.matrix(LDK)Quartier.HLM	-9.881831e-10
as.matrix(LDK)Quartier.JetdEau	-2.218721e-09
as.matrix(LDK)Quartier.Libertel	-2.986893e-09
as.matrix(LDK)Quartier.LiberteVI	-5.561404e-10
as.matrix(LDK)Quartier.Malika	-2.769298e-09
as.matrix(LDK)Quartier.Mamelles	7.264257e-09
as.matrix(LDK)Quartier.Medina	-3.289932e-09
as.matrix(LDK)Quartier.Mermoz	1.880747e-09
as.matrix(LDK)Quartier.Ngor	3.342974e-09
as.matrix(LDK)Quartier.NiayeCoker	-2.570650e-09
as.matrix(LDK)Quartier.Parcelles	6.093702e-11
as.matrix(LDK)Quartier.Pikine	-2.570650e-09
as.matrix(LDK)Quartier.Plateau	8.674606e-09
as.matrix(LDK)Quartier.PointE	5.691461e-09
as.matrix(LDK)Quartier.SacreCoeur	1.026326e-09
as.matrix(LDK)Quartier.SacreCoeurIII	-4.787426e-10
as.matrix(LDK)Quartier.Yoff	-4.414265e-09

D'après les signes des coefficients représentés ci-dessus, toutes les variables de taille contribuent positivement à la variable prédite.

Par conséquent, le logarithme du loyer va augmenter.

En revanche, les modalités de la même variable sont opposées ; c'est la propriété de toute méthode qui minimise la norme des vecteurs des coefficients en cas de multicolinéarité.

On l'observe pour toutes les variables binaires.

Nous avons vu qu'il existait des variables ayant des valeurs contradictoires avec leurs modalités, par exemple la variable **Egout**.

On a intérêt a l'enlever parce qu'il se trouve que c'est un résidu historique et c'est le contraire de la logique.

Nous allons regarder maintenant les **coefficients des variables** pour cinq composantes :

as.matrix(LDK)SurfTerrain	9.800417e-04
as.matrix(LDK)SurfHabitable	-6.080843e-04
as.matrix(LDK)SurfPiecResid	-4.890128e-03
as.matrix(LDK)NbPieces	1.059518e-01
as.matrix(LDK)NbPiecesResid	2.486098e-02
as.matrix(LDK)NbSDB	2.051700e-02
as.matrix(LDK)NbChamBur	-3.947268e-02
as.matrix(LDK)NbSalonsSAM	6.433366e-02
as.matrix(LDK)NbWC	4.396108e-02
as.matrix(LDK)NbCuis	1.661278e-02
as.matrix(LDK)Type.Appart	-5.068254e-03
as.matrix(LDK)Type.Villa	5.068254e-03
as.matrix(LDK)Standing.Non	-7.684980e-02
as.matrix(LDK)Standing.Oui	7.684980e-02
as.matrix(LDK)Etat.Bon	7.685578e-02
as.matrix(LDK)Etat.Mediocre	-8.015038e-02
as.matrix(LDK)Etat.Neuf	1.507422e-02
as.matrix(LDK)Etat.Vetuste	-1.177962e-02

as.matrix(LDK)Etat.Vetuste	-1.177962e-02
as.matrix(LDK)Jardin.Non	-2.212229e-02
as.matrix(LDK)Jardin.Oui	2.212229e-02
as.matrix(LDK)Cour.Non	-4.260151e-02
as.matrix(LDK)Cour.Oui	4.260151e-02
as.matrix(LDK)Piscine.Non	5.304107e-04
as.matrix(LDK)Piscine.Oui	-5.304107e-04
as.matrix(LDK)Garage.Non	-3.485277e-02
as.matrix(LDK)Garage.Park	2.073348e-02
as.matrix(LDK)Garage.Priv1v	1.550092e-02
as.matrix(LDK)Garage.Priv2v	-1.381633e-03
as.matrix(LDK)Egout.Non	1.619285e-04
as.matrix(LDK)Egout.Oui	-1.619285e-04
as.matrix(LDK)HiTech.1inst	-1.099029e-02
as.matrix(LDK)HiTech.2inst	-7.777000e-03
as.matrix(LDK)HiTech.Non	1.876729e-02
as.matrix(LDK)DistCtrVille.0	9.423821e-02
as.matrix(LDK)DistCtrVille.1a5km	-2.308172e-02

as.matrix(LDK)Quartier.HLM	-1.364424e-03
as.matrix(LDK)Quartier.JetdEau	-6.722760e-03
as.matrix(LDK)Quartier.Libertel	-3.458633e-03
as.matrix(LDK)Quartier.LiberteVI	-6.971980e-03
as.matrix(LDK)Quartier.Malika	-1.337591e-02
as.matrix(LDK)Quartier.Mamelles	-4.089168e-03
as.matrix(LDK)Quartier.Medina	5.707091e-04
as.matrix(LDK)Quartier.Mermoz	1.409021e-02
as.matrix(LDK)Quartier.Ngor	8.687805e-03
as.matrix(LDK)Quartier.NiayeCoker	-1.619390e-04
as.matrix(LDK)Quartier.Parcelles	-1.799292e-02
as.matrix(LDK)Quartier.Pikine	2.318819e-03
as.matrix(LDK)Quartier.Plateau	9.423821e-02
as.matrix(LDK)Quartier.PointE	-2.977700e-04
as.matrix(LDK)Quartier.SacreCoeur	5.693544e-03
as.matrix(LDK)Quartier.SacreCoeurIII	-3.629445e-03
as.matrix(LDK)Quartier.Yoff	7.446335e-06

On remarque de même qu'il faudra recommencer cette analyse en agrégant les modalités "Vetuste" et "Médiocre" (comme pour la variable "Egout").

On observe ainsi que pour la variable "garage non" ; on a un effet qui augmente avec la qualité de la solution "garage". De plus, les variables "Hitech-Non" qui a un effet positif par contre les deux variables "Hitech.1inst" et "Hitech.2inst" ont un effet négatif. D'où on a une confusion d'effet.

Il se trouve qu'une partie de cette confusion d'effet qui est des corrections, permet visiblement de donner une **meilleure prédiction**.

```
> regLDKc5$coefficients[2:6]
LDKLogYpls5$scoresComp 1 LDKLogYpls5$scoresComp 2 LDKLogYpls5$scoresComp 3 LDKLogYpls5$scoresComp 4
             0.004253071              0.005680053              0.408712086              0.100948886
LDKLogYpls5$scoresComp 5
             0.277579705
> var(LDKLogYpls5$scores)
              Comp 1        Comp 2        Comp 3       Comp 4        Comp 5
Comp 1  4.657292e+04  3.913470e-12  1.358349e-13 4.725423e-13 -2.217983e-14
Comp 2  3.913470e-12  2.850608e+03 -1.290101e-14 4.461717e-14 -2.943782e-14
Comp 3  1.358349e-13 -1.290101e-14  1.696126e+00 1.838755e-15  1.048034e-16
Comp 4  4.725423e-13  4.461717e-14  1.838755e-15 1.248503e+01  1.778051e-16
Comp 5 -2.217983e-14 -2.943782e-14  1.048034e-16 1.778051e-16  5.443803e-01
```

D'après le graphique ci-dessus ; on observe qu'on a des composantes qui capturent extrêmement peu d'information par rapport aux deux premières. Autant que covariance, la troisième composante va faire mieux que la quatrième.

5 Régressions ridge & LASSO :

5.1 Régression Ridge :

Ayant diagnostiqué un problème mal conditionné mais désirant conserver toutes les variables explicatives pour des raisons d'interprétation, il est possible d'améliorer les propriétés numériques et la variance des estimations en considérant un **estimateur biaisé des paramètres** par procédure de régularisation.

Soit le modèle linéaire :

$$\boxed{Y = \tilde{X}\tilde{\beta} + \epsilon}$$

L'estimateur ridge de $\tilde{\beta}$ dans ce modèle est défini par :

$$\boxed{\tilde{\beta} = (X'WX + \lambda I)^{-1}X'WY}$$

$\diamond W = \frac{1}{n}I_n$

$\diamond$ λ qui est le paramètre de pénalité qui est positif

$\diamond X$ désigne la matrice des variables de taille, qualité et situation

$\diamond Y$ désigne le vecteur loyer qu'on veut modéliser

La fonction qui effectue la régession ridge nous permet de faire des régressions pénalisées ainsi de contôler le coefficient de pénalité.

	Df	%Dev	Lambda
1	82	0.00	946.70
2	82	3.53	903.70
3	82	3.69	862.60
4	82	3.86	823.40
5	82	4.04	786.00
6	82	4.23	750.20
7	82	4.42	716.10
8	82	4.62	683.60
9	82	4.83	652.50
10	82	5.05	622.90
11	82	5.28	594.60
12	82	5.52	567.50
13	82	5.77	541.70
14	82	6.03	517.10
15	82	6.30	493.60
16	82	6.58	471.20
17	82	6.88	449.80
18	82	7.18	429.30
19	82	7.50	409.80
20	82	7.84	391.20

...

```
 81   82 62.00  22.91
 82   82 63.19  21.87
 83   82 64.36  20.88
 84   82 65.52  19.93
 85   82 66.67  19.02
 86   82 67.80  18.16
 87   82 68.91  17.33
 88   82 70.00  16.54
 89   82 71.07  15.79
 90   82 72.12  15.07
 91   82 73.14  14.39
 92   82 74.15  13.73
 93   82 75.13  13.11
 94   82 76.09  12.51
 95   82 77.03  11.95
 96   82 77.94  11.40
 97   82 78.82  10.88
 98   82 79.69  10.39
 99   82 80.52   9.92
100   82 81.34   9.47
```

Le graphique ci-dessus représente, en première colonne, le nombre de coefficients non nuls pour chaque valeur de λ. On observe que λ a commencé très haut ; il a commencé à beaucoup pénaliser ; On a unλ qui est très fort ensuite ; il commence à diminuer et à 100 itérations ; il arrive à 9.47.

La deuxième colonne représente le pourcentage de variance expliqué pour chaque λ.

On observe que la dernière valeur de notre graphique defini ci-dessus, nous renvoie à un %DEV qui vaut 81.34.

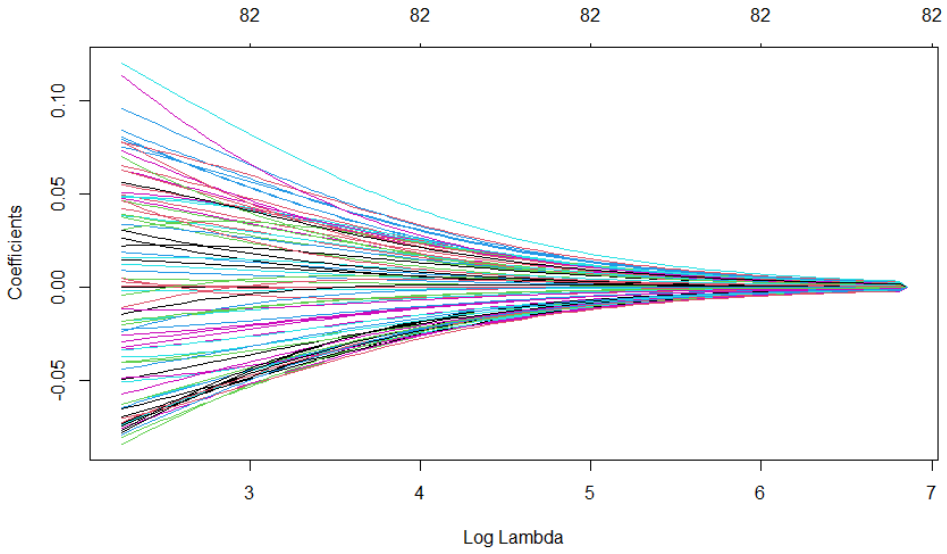
D'où, le R^2 est égal à 0.8134.

```
          SurfTerrain          SurfHabitable          SurfPiecResid              NbPieces          NbPiecesResid
         0.0001849058          0.0003614481          0.0045305744          0.0087954064          0.0122674857
                NbSDB             NbChamBur            NbSalonsSAM                 NbWC               NbCuis
         0.0477684989          0.0148582520          0.0419034645          0.0461485294          0.0837439606
          Type.Appart           Type.Villa          Standing.Non          Standing.Oui             Etat.Bon
        -0.0628236394          0.0628294111         -0.0651229391          0.0651265687          0.0377033877
        Etat.Mediocre             Etat.Neuf           Etat.Vetuste           Jardin.Non           Jardin.Oui
        -0.0643893009          0.0382952106         -0.0324671938         -0.0494561339          0.0494613569
             Cour.Non              Cour.Oui            Piscine.Non           Piscine.Oui           Garage.Non
        -0.0806689700          0.0806730621         -0.0506404096          0.0506487428         -0.0693381409
          Garage.Park          Garage.Priv1v          Garage.Priv2v             Egout.Non             Egout.Oui
         0.0029068701          0.0388790812          0.0751685591          0.0484981155         -0.0485062883
          HiTech.1inst          HiTech.2inst            HiTech.Non          DistCtrVille.0      DistCtrVille.1a5km
         0.0220888837          0.0773116813         -0.0404177032          0.1135000405         -0.0020991186
    DistCtrville.inf1km     DistCtrville.sup5km          Commerc.inf2km          Commerc.sup2km          BordMer.inf2km
        -0.0334554875         -0.0144577215          0.0046261711         -0.0046236760          0.0336949479
        BordMer.sup2km     Distractions.inf2km     Distractions.sup2km     AxeRoutier.inf1km     AxeRoutier.sup1km
        -0.0336974142          0.0732230848         -0.0732246821          0.0629453272         -0.0629462704
StandingQuartier.bourge     StandingQuartier.moy     StandingQuartier.popu    QuartierAffaires.Non    QuartierAffaires.Oui
         0.0957745461         -0.0183845305         -0.0753407843         -0.0780265142          0.0780252802
      Quartier.Almadies        Quartier.BelAir          Quartier.Bopp         Quartier.Castors       Quartier.Colobane
```

On remarque d'apres les coefficients présentés ci-dessus ; que les modalités de la même variable binaire sont opposées ; par exemple **Typeappart**, **Typevilla**. Les caractéristiques de taille n'ont pas tous des effets régularisés positifs. En revanche, on est assez proche de la régression classique ; d'où il y a une compensation d'effet entre certaines variables qui peuvent se compenser. Ainsi que pour les variables ordinales, en guide d'exemple ; **GaragePrivé1** et

GaragePrivé2 sont ordonnées en terme de qualité. De même la variable Egout ; toujours le même paradoxe : ça vient du fait de ne pas être accordés à l'égout. En effet ; il y a une part de l'effet qualité et aussi taille de logement qui va se reporter sur l'égout.

Dans la suite, on fait varier λ puis on regardera l'évolution (de la stabilité) des coefficients.

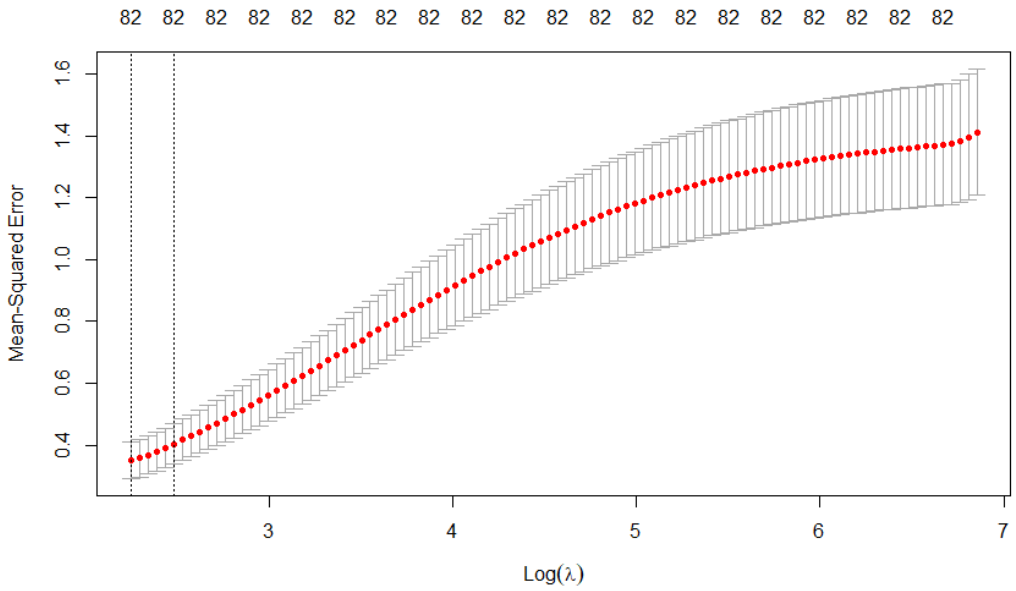


Sur le côté gauche, λ est proche de 0, il n'y a presque aucune contrainte sur les coefficients. Il s'agit des estimations des moindres carrées complètes.

Comme λ devient plus grand, il pousse les coefficients vers 0. À l'extrême, où λ est un peu plus grand, les coefficients sont tous essentiellement 0.

Entre les deux, ils sont réduits vers 0 à mesure que λ devient plus grand, mais pas uniformément.

Nous allons maintenant regarder l'erreur quadratique moyenne en fonction des valeurs de λ (cf graphique ci-dessous) :



La valeur minimale de MSE (Mean Squared Error) en validation croisée est 0.3697236.

Ainsi donc le λ qui correspond à cette valeur minimale vaut 9.466981.

Nous avons recommencer la regression avec le meilleur lambda obtenu précédemment puis nous trouvrons les coefficients du modèle :

	s0
(Intercept)	4.5504571462
SurfTerrain	0.0001847839
SurfHabitable	0.0003612079
SurfPiecResid	0.0045285472
NbPieces	0.0087914863
NbPiecesResid	0.0122634696
NbSDB	0.0477620380
NbChamBur	0.0148572778
NbSalonsSAM	0.0419043836
NbWC	0.0461536698
NbCuis	0.0837538080
Type.Appart	-0.0628372650
Type.Villa	0.0628458655
Standing.Non	-0.0651306722
Standing.Oui	0.0651397914
Etat.Bon	0.0377067427
Etat.Mediocre	-0.0643994251
Etat.Neuf	0.0383080260
Etat.Vetuste	-0.0324734934
Jardin.Non	-0.0494702152
Cour.Non	-0.0806814458
Cour.Oui	0.0806841932
Piscine.Non	-0.0506623728
Piscine.Oui	0.0506725636
Garage.Non	-0.0693591791
Garage.Park	0.0028997551
Garage.Priv1v	0.0388881931
Garage.Priv2v	0.0751981783
Egout.Non	0.0485229866
Egout.Oui	-0.0485280212
HiTech.1inst	0.0221007368
HiTech.2inst	0.0773392132
HiTech.Non	-0.0404310853
DistCtrVille.0	0.1134978890
DistCtrVille.1a5km	-0.0020987183
DistCtrVille.inf1km	-0.0334663693
DistCtrVille.sup5km	-0.0144515885
Commerc.inf2km	0.0046179802
Commerc.sup2km	-0.0046187218
BordMer.inf2km	0.0337037251
BordMer.sup2km	-0.0337032019

La régression ridge permet donc de contourner les problèmes de colinéarité même en présence d'un nombre important de variables explicatives ou prédictives.

La principale faiblesse de cette méthode est la non-selection des variables qui peut conduire à des erreurs d'interprétation. En effet, sans cette sélection toutes les variables sont concernées dans le modèle.

D'autres approches par pénalisation permettent également une sélection, c'est le cas de la **Régression Lasso**.

5.2 Régression LASSO :

La méthode Lasso correspond à la minimisation d'un critère des moindres carrés avec une pénalité de type l1. Soit $||\beta||_1 = \sum_{j=1}^{p} |\beta_j|$.

Soit le modèle linéaire :

$$\boxed{Y = X\beta + \epsilon}$$

L'estimateur Lasso de β dans ce modèle est défini par :

$$\boxed{\hat{\beta}_{Lasso} = argmin_{\beta, ||\beta||_1 \leq t}(||Y - X\beta||)^2}$$

Pour un t convenablement choisi.

Nous allons tout d'abord procéder à une validation croisée pour bien choisir la valeur de λ.

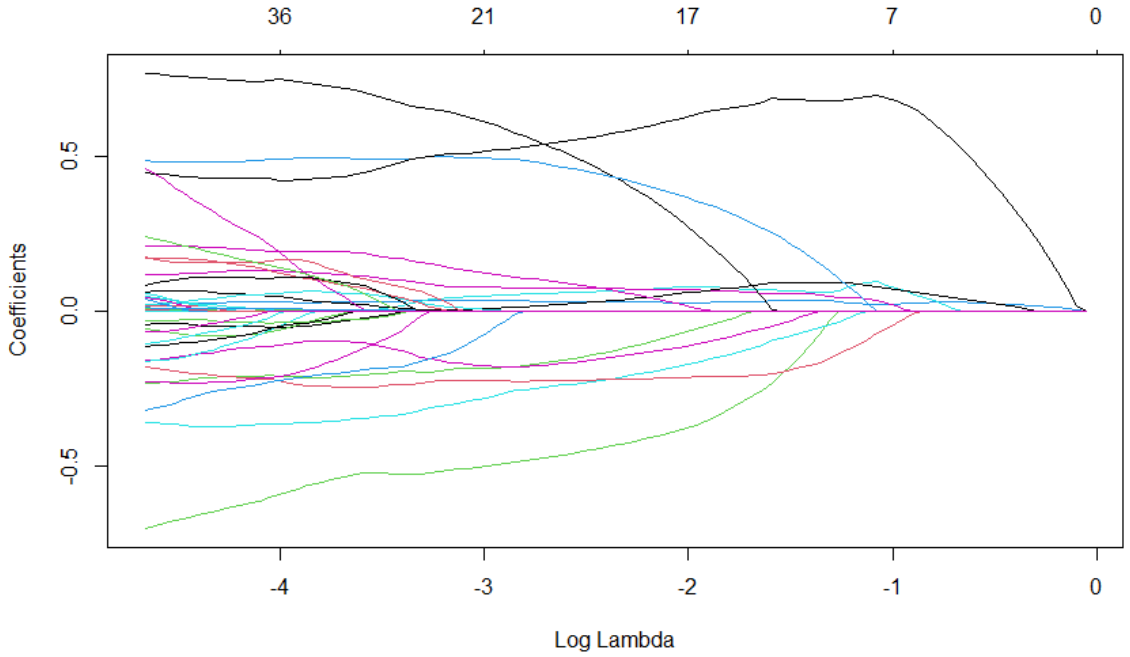
	Df	%Dev	Lambda
1	0	0.00	0.94670
2	3	6.34	0.90370
3	3	13.15	0.86260
4	3	19.36	0.82340
5	3	25.02	0.78600
6	4	30.18	0.75020
7	4	34.89	0.71610
8	4	39.17	0.68360
9	4	43.08	0.65250
10	4	46.65	0.62290
11	4	49.89	0.59460
12	4	52.85	0.56750
13	4	55.54	0.54170
14	4	58.00	0.51710
15	5	60.27	0.49360
16	5	62.37	0.47120
17	5	64.27	0.44980
18	5	66.01	0.42930
19	6	67.62	0.40980
20	7	69.25	0.39120

80	32	98.83	0.02400
81	32	98.90	0.02291
82	33	98.96	0.02187
83	35	99.02	0.02088
84	35	99.09	0.01993
85	35	99.14	0.01902
86	36	99.20	0.01816
87	37	99.25	0.01733
88	37	99.31	0.01654
89	36	99.35	0.01579
90	37	99.39	0.01507
91	38	99.42	0.01439
92	38	99.46	0.01373
93	39	99.49	0.01311
94	40	99.52	0.01251
95	40	99.55	0.01195
96	41	99.59	0.01140
97	41	99.61	0.01088
98	41	99.64	0.01039
99	41	99.67	0.00992
100	41	99.69	0.00947

On observe que la dernière valeur de notre graphique defini ci-dessus, nous renvoie à un %DEV qui vaut 99.69.

D'où, le R^2 est égal à 0.9969.

On regarde dans la suite, les coefficients en fonction de α :



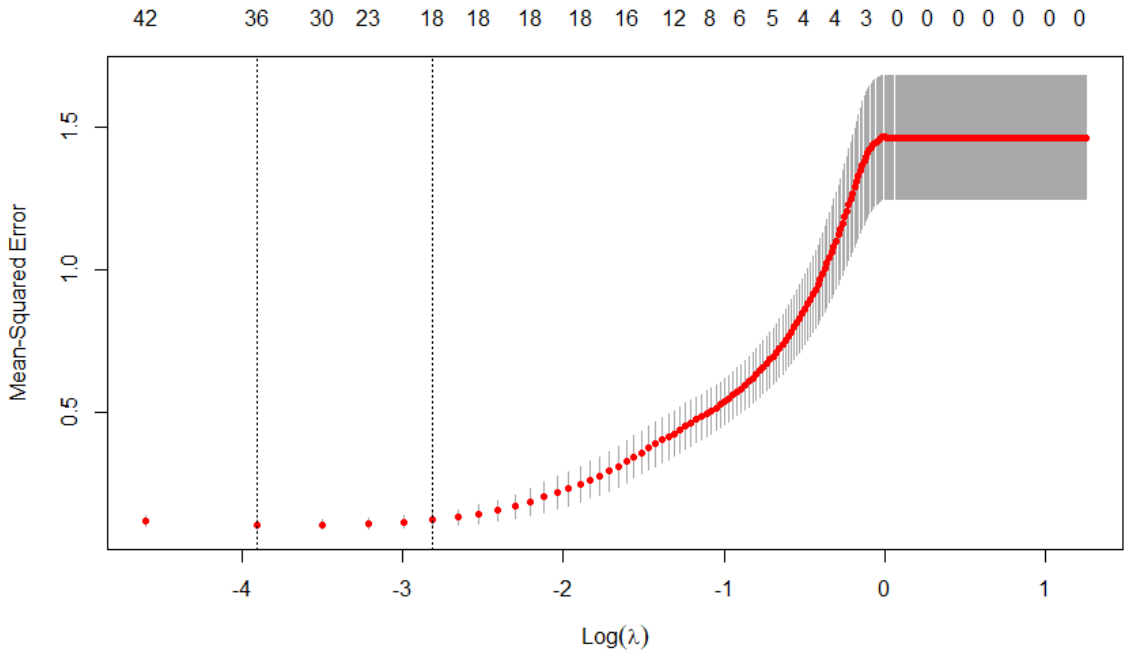
Comme on peut le voir, la **Régression Lasso** permet de supprimer des variables en mettant leur poids à zéro ; c'est le cas si deux variables sont corrélées.

L'une sera sélectionnée par la régression Lasso, l'autre supprimée.

SurfTerrain	SurfHabitable	SurfPiecResid	NbPieces	NbPiecesResid
6.004233e-04	0.000000e+00	1.127415e-02	2.253262e-02	0.000000e+00
NbSDB	NbChamBur	NbSalonsSAM	NbWC	NbCuis
5.689767e-02	0.000000e+00	1.164135e-01	6.307075e-02	1.731411e-02
Type.Appart	Type.Villa	Standing.Non	Standing.Oui	Etat.Bon
-2.327086e-01	2.109452e-13	0.000000e+00	0.000000e+00	2.098806e-02
Etat.Mediocre	Etat.Neuf	Etat.Vetuste	Jardin.Non	Jardin.Oui
-1.601989e-01	0.000000e+00	0.000000e+00	-4.312547e-02	1.531322e-13
Cour.Non	Cour.Oui	Piscine.Non	Piscine.Oui	Garage.Non
-7.045781e-01	3.519603e-13	0.000000e+00	0.000000e+00	-3.619065e-01
Garage.Park	Garage.Priv1v	Garage.Priv2v	Egout.Non	Egout.Oui
4.366410e-02	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
HiTech.1inst	HiTech.2inst	HiTech.Non	DistCtrVille.0	DistCtrVille.1a5km
0.000000e+00	0.000000e+00	0.000000e+00	7.709611e-01	0.000000e+00
DistCtrVille.inf1km	DistCtrVille.sup5km	Commerc.inf2km	Commerc.sup2km	BordMer.inf2km
0.000000e+00	0.000000e+00	1.718972e-01	0.000000e+00	6.971288e-03
BordMer.sup2km	Distractions.inf2km	Distractions.sup2km	AxeRoutier.inf1km	AxeRoutier.sup1km
0.000000e+00	4.863005e-01	-1.296599e-03	2.123460e-01	0.000000e+00
StandingQuartier.bourge	StandingQuartier.moy	StandingQuartier.popu	QuartierAffaires.Non	QuartierAffaires.Oui
4.484426e-01	0.000000e+00	-1.799222e-01	0.000000e+00	0.000000e+00
Quartier.Almadies	Quartier.BelAir	Quartier.Bopp	Quartier.Castors	Quartier.Colobane

D'après le graphique ci-dessus ; on remarque que pour quelques variables binaires il ya une des deux qui est suprimé.

Le graphique suivant indique les erreurs quadratiques moyennes en fonction des valeurs de λ :



On obtient la valeur minimale de λ qui est égale à 0.00947 avec la commande ***cvfit2$lambda.min*** , où *cvfit2* représente la fonction de la validation croisée pour bien choisir la valeur de λ. Ce choix de λ correspond à une erreur quadratique moyenne égale à 0.09327298.

On retrouve alors une erreur qui est plus petite que celle de la régression PLS et Ridge.

On relance la regression avec ce choix de lambda et on obtient les coefficients suivants :

SurfTerrain	SurfHabitable	SurfPiecResid	NbPieces	NbPiecesResid
6.004233e-04	0.000000e+00	1.127415e-02	2.253262e-02	0.000000e+00
NbSDB	NbChamBur	NbSalonsSAM	NbWC	NbCuis
5.689767e-02	0.000000e+00	1.164135e-01	6.307075e-02	1.731411e-02
Type.Appart	Type.Villa	Standing.Non	Standing.Oui	Etat.Bon
-2.327086e-01	2.109452e-13	0.000000e+00	0.000000e+00	2.098806e-02
Etat.Mediocre	Etat.Neuf	Etat.Vetuste	Jardin.Non	Jardin.Oui
-1.601989e-01	0.000000e+00	0.000000e+00	-4.312547e-02	1.531322e-13
Cour.Non	Cour.Oui	Piscine.Non	Piscine.Oui	Garage.Non
-7.045781e-01	3.519603e-13	0.000000e+00	0.000000e+00	-3.619065e-01
Garage.Park	Garage.Priv1v	Garage.Priv2v	Egout.Non	Egout.Oui
4.366410e-02	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
HiTech.1inst	HiTech.2inst	HiTech.Non	DistCtrVille.0	DistCtrVille.1a5km
0.000000e+00	0.000000e+00	0.000000e+00	7.709611e-01	0.000000e+00
DistCtrVille.inf1km	DistCtrVille.sup5km	Commerc.inf2km	Commerc.sup2km	BordMer.inf2km
0.000000e+00	0.000000e+00	1.718972e-01	0.000000e+00	6.971288e-03
BordMer.sup2km	Distractions.inf2km	Distractions.sup2km	AxeRoutier.inf1km	AxeRoutier.sup1km
0.000000e+00	4.863005e-01	-1.296599e-03	2.123460e-01	0.000000e+00
StandingQuartier.bourge	StandingQuartier.moy	StandingQuartier.popu	QuartierAffaires.Non	QuartierAffaires.Oui
4.484426e-01	0.000000e+00	-1.799222e-01	0.000000e+00	0.000000e+00

DistCtrVille.inf1km	DistCtrVille.sup5km	Commerc.inf2km	Commerc.sup2km	BordMer.inf2km
0.000000e+00	0.000000e+00	1.718972e-01	0.000000e+00	6.971288e-03
BordMer.sup2km	Distractions.inf2km	Distractions.sup2km	AxeRoutier.inf1km	AxeRoutier.sup1km
0.000000e+00	4.863005e-01	-1.296599e-03	2.123460e-01	0.000000e+00
StandingQuartier.bourge	StandingQuartier.moy	StandingQuartier.popu	QuartierAffaires.Non	QuartierAffaires.Oui
4.484426e-01	0.000000e+00	-1.799222e-01	0.000000e+00	0.000000e+00
Quartier.Almadies	Quartier.BelAir	Quartier.Bopp	Quartier.Castors	Quartier.Colobane
0.000000e+00	0.000000e+00	-5.675849e-02	4.125753e-02	-1.623327e-01
Quartier.Derkle	Quartier.Fann	Quartier.FannHock	Quartier.FannResidence	Quartier.Fass
4.633412e-01	8.387169e-02	0.000000e+00	1.743841e-01	-3.304288e-02
Quartier.FenetreMermoz	Quartier.Foire	Quartier.GrandYoff	Quartier.GueuleTapee	Quartier.Hann
0.000000e+00	0.000000e+00	-3.214242e-01	0.000000e+00	0.000000e+00
Quartier.HLM	Quartier.JetdEau	Quartier.LibarteI	Quartier.LibarteVI	Quartier.Malika
4.899427e-02	-2.288655e-01	0.000000e+00	-1.151605e-01	0.000000e+00
Quartier.Mamelles	Quartier.Medina	Quartier.Mermoz	Quartier.Ngor	Quartier.NiayeCoker
0.000000e+00	0.000000e+00	4.865839e-03	0.000000e+00	0.000000e+00
Quartier.Parcelles	Quartier.Pikine	Quartier.Plateau	Quartier.PointE	Quartier.SacreCoeur
0.000000e+00	2.399992e-01	1.122180e-02	-1.069969e-01	0.000000e+00
Quartier.SacreCoeurIII	Quartier.Yoff			
-6.740337e-02	0.000000e+00			

On fait encore mieux qu'avec la régression ridge ! En effet, le lasso a pour avantage de pouvoir sélectionner un sous-ensemble des variables explicatives afin de permettre une meilleure généralisation.

On produit un tableau synoptique des coefficients selon les méthodes, afin d'en faciliter la comparaison :

Variables \ Méthode	RCP thèmes	RCP Class Var *(4 classes)*	PLS *(sans log)*	Ridge	Lasso
SurfTerrain	-0.9015978499	0.23617140	1.271465	0.0001847839	3.689737e-04
SurfHabitable	-0.4312907882	-0.1251095	-3.797501	0.0003612079	.
SurfPiecResid	-0.0253149027	0.2234590	19.680514	0.0045285472	1.271086e-02
NbPieces	-0.0167691961	0.29604973	80.686445	0.0087914863	2.444574e-02
NbPiecesResid	-0.0102744528	0.29250859	-44.272942	0.0122634696	.
NbSDB	-0.0031195791	0.131350494	6.503057	0.0477620380	9.525848e-02
NbChambur	-0.0073873933	0.28205833	-124.898792	0.0148572778	.
NbSalonSAM	-0.0028870594	0.26580618	80.625851	0.0419043836	9.199129e-02
NbWC	-0.0030333584	0.24804408	220.032588	0.0461536698	4.842380e-02
NbCuis	-0.0003418059	0.12176344	-101.576257	0.0837538080	2.827871e-02
Type.Appart	0.32132128	.	-152.263769	-0.0628372650	-2.507841e-01
Type.Villa	-0.32132128	-0.101140388	152.263769	0.0628458655	.
Standing.Non	0.32546119	0.079438096	22.730178	-0.0651306722	.
Standing.Oui	-0.32546119	0.21352213	- 22.730178	0.0651397914	.
Etat.Bon	-0.08252816	0.021192565	-77.200550	0.0377067427	1.140692e-03
Etat.Mediocre	0.20235087	-0.13102419	-1.365114	-0.0643994251	-1.947687e-01
Etat.Neuf	-0.14931011	0.17025335	154.002137	0.0383080260	.
Etat.Vetuste	0.02948739	-0.05013664	-75.436473	-0.0324734934	.
Jardin.Non	0.28096828	0.07740654	156.576374	-0.0494702152	-7.175928e-02
Jardin.Oui	-0.28096828	0.17958306	- 156.576374	0.0494770517	1.122933e-16
Cour.Non	0.08293188	-0.13405684	82.575699	-0.0806814458	-7.461867e-01
Cour.Oui	-0.08293188	0.11107537	- 82.575699	0.0806841932	7.690091e-14
Piscine.Non	0.12915276	0.1111670	28.549217	-0.0506623728	.
Piscine.Oui	-0.12915276	0.20703316	- 28.549217	0.0506725636	.
Garage.Non	0.35548051	-0.22413194	-218.465669	-0.0693591791	-3.349738e-01
Garage.Park	-0.01841398	-0.03497982	-90.586728	0.0028997551	4.319967e-02
Garage.Priv1v	-0.20573983	.	101.505948	0.0388881931	.
Garage.Priv2v	-0.13132670	0.17615378	207.546449	0.0751981783	.
Egout.Non	-0.14322447	-0.076161151	9.099381	0.0485229866	.
Egout.Oui	0.14322447	-0.23523364	- 9.099381	-0.0485280212	.
HiTech.1inst	-0.15390251	.	79.313792	0.0221007368	.
HiTech.2inst	-0.06950076	.	135.699975	0.0773392132	.
HiTech.Non	0.22340326	-0.120368503	-215.013768	-0.0404310853	.
DistCtrVille.0	0.05979733	0.018958683	-5.566209	0.1134978890	6.827965e-01
DistCtrVille.1a5km	-0.18477847	-0.002804304	-127.653842	-0.0020987183	.
DistCtrVille.inf1km	-0.04759043	-0.071155945	44.683018	-0.0334663693	.
DistCtrVille.sup5km	0.17257157	0.14679135	88.537033	-0.0144515885	.
Commerc.inf2km	-0.13434571	-0.17353453	289.740604	0.0046179802	1.486377e-01
Commerc.sup2km	0.13434571	.	- 289.740604	0.0046187218	-1.086251e-16
BordMer.inf2km	0.32004593	-0.1023222	44.702990	0.0337037251	3.063656e-02
BordMer.sup2km	-0.32004593	0.060890620	- 44.702990	-0.0337032019	.
Distractions.inf2km	0.28718104	.	-56.724933	0.0732295375	5.182085e-01
Distractions.sup2km	-0.28718104	.	56.724933	-0.0732299978	.
AxeRoutier.inf1km	0.39021347	.	22.276303	0.0629503755	1.940270e-01
AxeRoutier.sup1km	-0.39021347	.	- 22.276303	-0.0629492205	.
StandingQuartier.bourge	0.37498042	.	85.741406	0.0957790192	4.561529e-01
StandingQuartier.moy	-0.20635373	.	-57.725242	-0.0183825207	.
StandingQuartier.popu	-0.16862669	.	-28.016164	-0.0753417853	-1.801859e-01
QuartierAffaires.Non	-0.06392544	.	- 172.209236	-0.0780218240	-3.708936e-03
QuartierAffaires.Oui	0.06392544	.	172.209236	0.0780225067	.
Quartier.Almadies	.	.	-11.517459	0.0305840265	.
Quartier.BelAir	.	.	-121.380190	0.0186856223	.
Quartier.Bopp	.	.	81.191041	-0.0736677972	.
Quartier.Castors	.	.	276.295502	-0.0256732819	4.629276e-02
Quartier.Colobane	.	.	-60.236247	-0.0744034727	-1.337399e-01
Quartier.Derkle	.	.	135.136143	-0.0700682076	5.411992e-01
Quartier.Fann	.	.	216.541097	0.0698208589	3.776828e-02
Quartier.FannHock	.	.	220.792797	0.0791635446	.
Quartier.FannResidence	.	.	147.217434	0.1202105620	2.513412e-01
Quartier.Fass	.	.	-131.925518	-0.0573161363	-7.003918e-03
Quartier.FenetreMermoz	.	.	85.108018	0.0262481935	.
Quartier.Foire	.	.	-92.016357	-0.0111182717	.
Quartier.GrandYoff	.	.	140.291855	-0.0842374423	-2.926742e-01

Quartier.GueuleTapee	.	.	46.626183	-0.0226322829	.
Quartier.Hann	.	.	-21.488693	0.0158698076	.
Quartier.HLM	.	.	-390.518876	-0.0117097743	9.730112e-02
Quartier.JetdEau	.	.	129.828832	-0.0767583908	-1.967823e-01
Quartier.LibertéI	.	.	8.388645	-0.0725814026	.
Quartier.LibertéVI	.	.	-173.283630	-0.0203300633	-1.258936e-01
Quartier.Malika	.	.	-62.223329	0.0790798776	.
Quartier.Mamelles	.	.	69.637409	0.0483702326	.
Quartier.Medina	.	.	-99.905867	-0.0294313553	.
Quartier.Mermoz	.	.	-134.484917	0.0303892493	.
Quartier.Ngor	.	.	-49.020569	0.0546192055	.
Quartier.NiayeCoker	.	.	333.810418	-0.0401851870	.
Quartier.Parcelles	.	.	-88.168137	-0.0238868286	.
Quartier.Pikine	.	.	252.341973	-0.0373187607	2.613676e-01
Quartier.Plateau	.	.	-5.566209	0.1134952551	1.072528e-01
Quartier.PointE	.	.	-189.358674	0.0561492437	8.564901e-02
Quartier.SacreCoeur	.	.	-271.769177	0.0464885059	.
Quartier.SacreCoeurIII	.	.	-191.043840	-0.0183166801	-5.259822e-02
Quartier.Yoff	.	.	-49.299660	-0.0438516877	.
R^2 (en %)	78%	71.57%	64%	81%	99%
PRESS	2741161		3201973	0.36972	0.09327

6 Conclusion :

Pour des performances si voisines ; c'est recommander de commencer par faire une régression PLS pour comprendre le phénomène, en effet ; pour régler PLS on joue sur le nombre des composantes qui est discret, donc le réglage sera moins fin. Ensuite on compare le modèle obtenu à celle des régressions pénalisées.

Limitation de la régression Ridge :

La régression Ridge diminue la complexité d'un modèle, mais ne réduit pas le nombre de variables puisqu'elle ne conduit jamais à un coefficient zéro, mais ne fait que le minimiser. Par conséquent, ce modèle n'est pas bon pour la réduction des fonctionnalités.

Limitation de la régression LASSO :

Lasso a parfois du mal avec certains types de données. Si le nombre de prédicteurs (p) est supérieur au nombre d'observations (n), Lasso choisira tout au plus les prédicteurs n comme non-nuls, même si tous les prédicteurs sont pertinents (ou peuvent être utilisés dans l'ensemble de test).

De plus, s'il y a deux variables très collinéaires ou plus, la régression LASSO sélectionne l'une d'entre elles au hasard, ce qui n'est pas bon pour l'interprétation des données.

Avec ce TP nous avons pu mettre en oeuvre de manière pratique la régression ridge et le lasso. La régularisation fonctionne bien dans les deux cas, et permet de réduire l'erreur totale du modèle à l'aide de l'ajout du biais qui quantifie la complexité. L'un diminue grandement l'influence de certaines variables sur le modèle tandis que l'autre peut directement les supprimer (mettre leur poids à zéro), et donc être parcimonieux.

En revanche, on s'aperçoit que l'erreur minimale est celle de la régression lasso, cela est expliqué par le fait que la régression lasso rend nulle les variables les moins utiles et qui ne

donnent pas d'information, et donc le bruit diminue.

Donc la régression Lasso présente le meilleur compromis lors de la modélisation du loyer. En effet, elle élimine les variables les moins utiles à la modélisation et pénalise les coefficients des variables tout en gardant un bon pourcentage de variance expliquée.

7 Références :

1. https ://www.math.univ-toulouse.fr/ besse/Wikistat/pdf/st-m-app-linSelect.pdf

2. https ://www.institut-numerique.org/232-presentation-des-resultats-de-lacm-52e75a826f27a

3. http ://larmarange.github.io/analyse-R/analyse-des-correspondances-multiples.htmlfactominer

4. https ://openclassrooms.com/fr/courses/4444646-entrainez-un-modele-predictif-lineaire/4507811-tp-comparez-le-comportement-du-lasso-et-de-la-regression-ridge

5. http ://eric.univ-lyon2.fr/ ricco/cours/slides/regularized$_{r}egression.pdf$

6. http ://eric.univ-lyon2.fr/ ricco/tanagra/fichiers/fr$_{T}anagra_{R}idge_{E}lasticnet_{R}.pdf$

7. https ://en.wikipedia.org/wiki/Lasso$_{(}statistics)$

8. https ://brilliant.org/wiki/ridge-regression/

8 ANNEXE :

Chargement librairies + Importation des données

```
library(ade4) #Librairie qui permet l'implentation de fonctions statistiques
#et graphiques
library(FactoMineR) #Il permet de réaliser des analyses classiques telles que
l'analyse en composantes principales (ACP), l'analyse des correspondances (AC)
l'analyse des correspondances multiples (ACM) ainsi que des analyses plus
avancées.
library(pls)
library(glmnet) #Permet d'ajuster l'ensemble du chemin de régularisation lasso
ou élastique-net pour la régression linéaire
library(corrplot)#Permet de visualiser une matrice de corrélation par
corrélogramme
library(readr)
logtsDK <- read.delim("logtsDK.csv")#Pour importer cet ensemble de données
```

Partie 1

```
Transformation des variables qualitatives en indicatrices:
logtsDK_taille <- logtsDK[,3:12]
logtsDK_fact <- logtsDK[,13:29]
logtsDKnomIndic <- acm.disjonctif(logtsDK_fact)
LDK = cbind(logtsDK_taille,logtsDKnomIndic)

Regression MCO directement:
logtsDK_taille<-as.matrix(logtsDK_taille)
MCO_taille<- lm(logtsDK$Loyer~logtsDK_taille,data= as.data.frame(logtsDK))
summary(MCO_taille)

#ACP par thème:
#Thème Taille:
PCA1<-PCA(logtsDK_taille)
plot.PCA(PCA1,choix = "var")
PCA1$var$cos2
corrplot(PCA1$var$cos2)
#Valeurs propres:
PCA1$eig
barplot(PCA1$eig[,2])
pca1 = prcomp(logtsDK_taille)
#La croissance des valeurs propres
pca1$rotation #Les composantes principales
pca1$sdev ##Les écarts-types "bruts"
#Régression sur composantes principales
pcr1<- lm(logtsDK$Loyer~PCA1$ind$coord,data= as.data.frame(logtsDK))
summary(pcr1)

#On remarque qu'après la régression sur les composantes principales; les plus
utiles sont la première, et la dernière.

#Régression MCO sur les CP 1 et 5:
MCO_taille2<- lm(logtsDK$Loyer~PCA1$ind$coord[,c(1,5)],data= as.data.frame
(logtsDK))
summary(MCO_taille2)

#C'est plus intéressant de faire la régression sur les 2 composantes que sur
une seule. Ainsi, y'a pas de confusion
#entre les composantes; c'est la vraie-significativité.


#Thème Qualité:
logtsDK_qualite <- LDK[,11:33]
PCA2<- PCA(logtsDK_qualite, scale=TRUE)
corrplot(PCA2$var$cos2)
```

```
#Valeurs propres:
PCA2$eig
barplot(PCA2$eig[,2])
#Régression sur composantes principales :
pcr2<- lm(logtsDK$Loyer~PCA2$ind$coord,data= as.data.frame(logtsDK))
summary(pcr2)
#MCO sur la première composante:
MCO_qualite<- lm(logtsDK$Loyer~PCA2$ind$coord[,1],data= as.data.frame(logtsDK))
summary(MCO_qualite)


#Thème Situation:
logtsDK_situation<- LDK[,34:50]
PCA3<- PCA(logtsDK_situation)
corrplot(PCA3$var$cos2)
#Valeurs propres:
PCA3$eig
barplot(PCA3$eig[,2])
#Régression sur composantes principales :
pcr3<- lm(logtsDK$Loyer~PCA3$ind$coord,data= as.data.frame(logtsDK))
summary(pcr3)
#MCO sur les composantes principales 1 et 3 :
MCO_surface<- lm(logtsDK$Loyer~PCA3$ind$coord[,c(1,3)],data= as.data.frame(logtsDK))
summary(MCO_surface)

#Simultanémenet:
PCAT<- PCA(LDK)
pcrT<- lm(logtsDK$Loyer~PCAT$ind$coord,data= as.data.frame(logtsDK))
summary(pcrT)

#Modélisation du loyer sur les thèmes simultanément:
MCO_T<- lm(logtsDK$Loyer~PCAT$ind$coord[,c(1,2,5)],data= as.data.frame(logtsDK))
summary(MCO_T)

#Coefficients des variables reconstitués selon la RCP :

#Classification sur variables:
library(ClustOfVar)
LDK = cbind(logtsDK_taille,logtsDKnomIndic)
LDK_cr<- scale(LDK)*sqrt(81/82
#Standardiser les variables
#On procéde par une classifiction hiérarchique sur les variables avec l'indice
WARD:
dv<- dist(LDK_cr,method="euclidean")
CAH<- hclust(d=dv,method="ward.D")
```

```
#Dendrogramme
plot(CAH)

#Coupure de l'arbre pour k=2classes:
PV2<-cutree(tree = CAH,k=2)
#Calcul du R^2 des variables avec la variable de classe; On stocke tous les
R^2 dans un seul vecteur:
R2_PV2= cbind(rep(0,ncol(LDK_cr)))
for (i in cbind(1:ncol(LDK_cr))) {R2_PV2[i] = summary(lm(LDK_cr[,i]~as.factor(PV2)))
$r.squared}

#Calcul du R^2 de la partition:
R2P_PV2<- mean(R2_PV2)
# On lance une boucle pour calculer R2 de la partiton pour k appartient à {3,4,..,8}
V <- rep(0,8)
for(i in 1:8) {
  PV<-cutree(tree = CAH,k=(i+1))
  R2_PV = cbind(rep(0,ncol(LDK_cr)))
  for ( j in cbind(1:ncol(LDK_cr))) {R2_PV[j] =
    summary(lm(LDK_cr[,j]~as.factor(PV))$r.squared)

  V[i]<- mean(R2_PV)}
}

#Le vecteur V représenté ci-dessus contient les R^2 de chaque partition
#D'après le vecteur V qui contient les R^2 de chaque partition; on peut choisir
7 partitions.
#En effet, l'amélioration de l'agrégation de 6 à 7 classes est plus intéressante
que l'agrégation de 7 à 8 classes.

#saut d'inertie:
inertie <- sort(CAH$height, decreasing = TRUE)
plot(inertie[1:20], type = "s", xlab = "Nombre de classes", ylab = "Inertie")

plot(inertie[1:20], type = "s", xlab = "Nombre de classes", ylab = "Inertie")
points(c(2, 4), inertie[c(2,4)], col = c("green3", "red3"), cex = 2, lwd = 3)

#Description de la partition en 4 classes:
P4 <- cutree(tree = CAH,k=4)
summary(P4)
```

Partie 2

Régression PLS

```
LDKpls<-plsr(as.matrix(logtsDK[,2])~as.matrix(LDK),validation="LOO")
LDKpls$validation$PRESS
barplot(LDKpls$validation$PRESS)
plot(LDKpls)


#Selon les composantes retenues:
LDKpls1 <- plsr(as.matrix(logtsDK[,2]) ~as.matrix(LDK), ncomp=1)
cor(logtsDK[,2],LDKpls1$fitted.values[,1,1])

cor(logtsDK[,2],LDKpls1$fitted.values[,1,1])^2

plot(LDKpls1)

#On passe au log de la variable y à prédire (ici c'est le loyer):
LDKLogYpls <- plsr(as.matrix(log(logtsDK[,2]))~ as.matrix(LDK),validation = "LOO")
plot(LDKLogYpls)
LDKLogYpls1 <- plsr(as.matrix(log(logtsDK[,2])) ~as.matrix(LDK),ncomp=1)
barplot(LDKLogYpls$validation$PRESS)
plot(RMSEP(LDKLogYpls), legendpos ="topright")
summary(LDKLogYpls)

#On fait la prédiction avec 5 composantes:
LDKLogYpls5 = plsr(as.matrix(log(logtsDK[,2])) ~as.matrix(LDK),ncomp=5)
plot(LDKLogYpls5)
cor(log(logtsDK[,2]),LDKLogYpls5$fitted.values[,1,5])
cor(log(logtsDK[,2]),LDKLogYpls5$fitted.values[,1,5])^2

#On trouve R^2=0.975; on régle sur le nombre des composantes.
plot(LDKLogYpls, ncomp = 5, line = TRUE)

cor(x=LDK,y=LDKLogYpls5$scores)

#Interprétation du modèle prédictif fondé sur la première composante :
regLDKc1 = lm(log(logtsDK[,2]) ~ LDKLogYpls1$scores)
LogLoyerModelPLS1 = as.matrix(LDKLogYpls1$coefficients[,1,])%*%as.matrix(regLDKc1$coefficient

#Interprétation du modèle prédictif fondé sur les 5 premières composantes :
regLDKc5 <- lm(log(logtsDK[,2]) ~ LDKLogYpls5$scores)
LogLoyerModelPLS5 <- as.matrix(LDKLogYpls5$coefficients[,1,])%*%as.matrix(regLDKc5$coefficien
regLDKc5$coefficients[2:6]
var(LDKLogYpls5$scores)
```

```
Régression Ridge

#La fonction qui effectue la régession ridge nous permet de faire des régresions
#pénalisées,  et nous permet de contôler le coefficient de pénaalité:
logLoy <- log(logtsDK[,2])
fit1 <- glmnet(x=as.matrix(LDK) , y=logLoy, family="gaussian",alpha=0)
fit1

#Evolution des coeff quand lambda augmente:
plot(fit1, xvar='lambda')

cvfit1<- cv.glmnet(x=as.matrix(LDK), y=logLoy,family="gaussian",alpha=0
#Choix de lambda
plot(cvfit1)
#Courbe log(lambda) vs MSE

#valeur min de MSE (en validation croisée)
print(min(cvfit1$cvm))
##lambda corresp.
print(cvfit1$lambda.min)

#On relance la regression avec le meilleur lambda:
fit <- glmnet(x=as.matrix(LDK) , y=logLoy, family="gaussian",alpha=0, lambda =9.466981)
#Coefficients du modèle obtenu:
coef(fit)

#lambda le plus élevé dont le MSE est inf.
#à la borne haute de l'intervalle de min(MSE)
cvfit1$lambda.1se

#ici R^2= 0.81 vs R^2= 0.97  pour PLS sur 5composantes
fit1$beta[,100]

#Comparaison des coefficients de PLS et Ridge:
fit1$beta[,100]
```