UNIVERSITÉ D'ANGERS

Faculté des sciences

Département de Mathématique

Projet Data Mining 2 - Analyse du prix des diamants avec des méthodes de régression (OLS, PCR et PLS)

Rapport présenté pour le MASTER 1 Mention Data-Science

Par : ED-DAKI Issam

Sous la direction de Mr. Daniel Christophe

ANGERS, 20 mai 2021

Sommaire

1. Introduction

Disons que le diamant est un morceau de carbone, qui a traversé l'enfer sur terre à des pressions et des températures extrêmes. Les diamants sont rares, fantaisistes, désirables et méritent d'être conservés.Dans le marché du diamant on trouve que les prix ne sont pas fixes en faite ils différent en fonction de certaines caractéristiques (comme les dimension par exemple).Dans le cadre de cette rapport nous allons essayer d'étudier le prix d'un diamant en connaissant les valeur de certaines variables qui peuvent expliquer ou non ce prix.

2. La packages nécessaire

Commençons par importer les package dont on aura besoin tout au long de cette analyse:

```
#Librairies----
library(ggplot2)
library(gplots)
library(dplyr)
library(corrplot)
library(factoextra)
library(FactoMineR)
library(questionr)
library(lmtest)
library(pls)
library(plsr)
library(car)
```

3. La base de données

Dans cette étude nous travaillons avec la base de donnée "diamonds" disponible dans la librairie ggplot2 du logiciel R Ce jeu de données contient 53940 enregistrements (lignes) et de 10 variables (colones).Voici les premier 4 premier lignes:

Changons maintenant le nom des colonnes pour avoir plus de lisibilité :

```
colnames(diamonds) <- c("carat",
                        "cut",
                        "color",
                        "clarity",
                        "fdepth",
                        "table",
```

```
                    "price",
                    "length",
                    "width",
                    "depth")
head(diamonds,4)
```

```
## # A tibble: 4 x 10
##   carat cut     color clarity fdepth table price length width depth
##   <dbl> <ord>   <ord> <ord>    <dbl> <dbl> <int>  <dbl> <dbl> <dbl>
## 1 0.23  Ideal   E     SI2       61.5    55   326   3.95  3.98  2.43
## 2 0.21  Premium E     SI1       59.8    61   326   3.89  3.84  2.31
## 3 0.23  Good    E     VS1       56.9    65   327   4.05  4.07  2.31
## 4 0.290 Premium I     VS2       62.4    58   334   4.2   4.23  2.63
```

le jeu de donnée contient les 10 variables:

-quantitative:

carat : représente le poids d'un diamant en carat (1 carat=0.2 gramme)

table : représente la largeur du sommet du diamant par rapport au point le plus .

depth : représente le pourcentage de profondeur totale. Cette variable est une fonction des 3 variables suivantes (2 * z / (x + y))

x : représente la longueur d'un diamant, en mm.

y: représente la largeur d'un diamant, en mm.

z: représente la profondeur d'un diamant, en mm.

price : variable quantitative représentant le prix d'un diamant, en dollars.(la variable d'interet) qualitative:

cut : variable qualitative ordonnée représentant la qualité de la coupe d'un diamant. 5 modalités : Fair<Good<VeryGood<Premium<Ideal

color : variable qualitative ordonnée représentant la couleur d'un diamant. 7 modalités : D<E<F<G<H<I<J (D représentant la meilleure couleur, J la pire. Le classement semble avoir été fait dans le sens inverse de la variable cut)

clarity : variable qualitative ordonnée mesurant la clareté d'un diamant. 8 modalités : I1<SI2<SI1<VS2<VS1<VVS12<VVS1<IF (I1 représentant la pire clareté, IF la meilleure)

4. Analyse Exploratoire

4.1 Analyse univaririée

```
summary(diamonds)
```

```
##      carat               cut        color        clarity          fdepth
##  Min.   :0.2000   Fair      : 1610   D: 6775   SI1    :13065   Min.
```

```
:43.00
##  1st Qu.:0.4000   Good     : 4906   E: 9797   VS2    :12258   1st
Qu.:61.00
##  Median :0.7000   Very Good:12082   F: 9542   SI2    : 9194   Median
:61.80
##  Mean   :0.7979   Premium  :13791   G:11292   VS1    : 8171   Mean
:61.75
##  3rd Qu.:1.0400   Ideal    :21551   H: 8304   VVS2   : 5066   3rd
Qu.:62.50
##  Max.   :5.0100                     I: 5422   VVS1   : 3655   Max.
:79.00
##                                     J: 2808   (Other): 2531
##      table           price           length          width
##  Min.   :43.00   Min.   :  326   Min.   : 0.000   Min.   : 0.000
##  1st Qu.:56.00   1st Qu.:  950   1st Qu.: 4.710   1st Qu.: 4.720
##  Median :57.00   Median : 2401   Median : 5.700   Median : 5.710
##  Mean   :57.46   Mean   : 3933   Mean   : 5.731   Mean   : 5.735
##  3rd Qu.:59.00   3rd Qu.: 5324   3rd Qu.: 6.540   3rd Qu.: 6.540
##  Max.   :95.00   Max.   :18823   Max.   :10.740   Max.   :58.900
##
##      depth
##  Min.   : 0.000
##  1st Qu.: 2.910
##  Median : 3.530
##  Mean   : 3.539
##  3rd Qu.: 4.040
##  Max.   :31.800
##
```

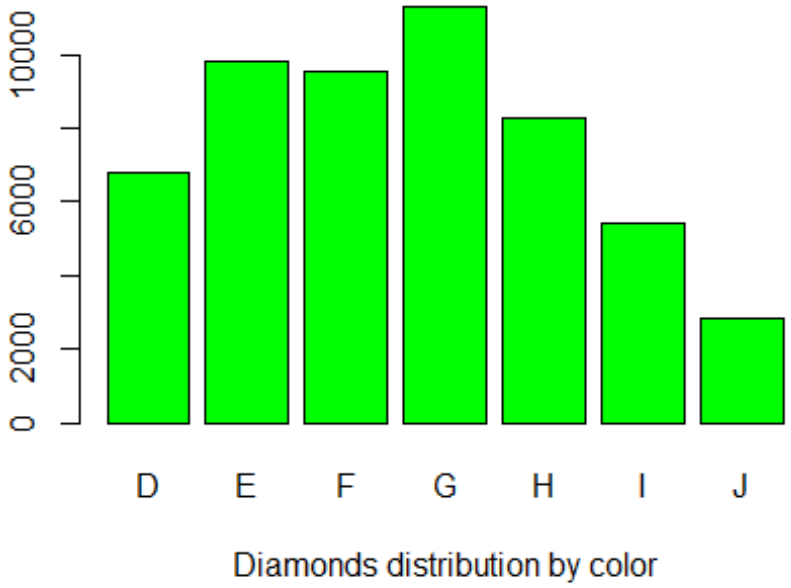
Dans le tableu ci-dessus nous pemet de dire que :

1 :les diamants ayant pour la variable **cut** la modalité “ideal” sont les plus répandus.Pour la variable **clarity** ce sont les modalités représentant une qualité intermédiaire.conçernant la variable **color** on voit que les modalités (D,E,G,H,I) sont relativement bien représentés par rapport à la modalité J,lorsqu’on va prendre un échantillon du jeu de données on doit garder les proportion d’aaprition des différentes modalité.

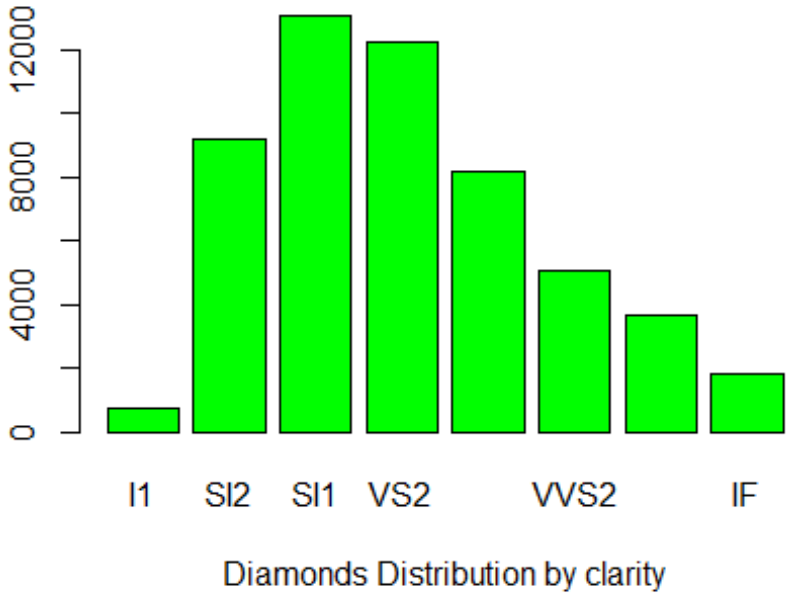
2 : les valeurs de **carat** sont tré dispersée ,et por la variable **price** on remarque que le quantile “3rd Qu.: 5324” ce qui veut dire que 75% des prix sont inférieurs à 5324 dollars. Donc par la suite il faut supprimer les valeurs aberrantes en faisant une étude des valeur extrême de cette variable.

```
plot(diamonds$cut, xlab="Diamonds distribution by cut",col = 'green')
```


```
plot(diamonds$color, xlab="Diamonds distribution by color",col = 'green')
```



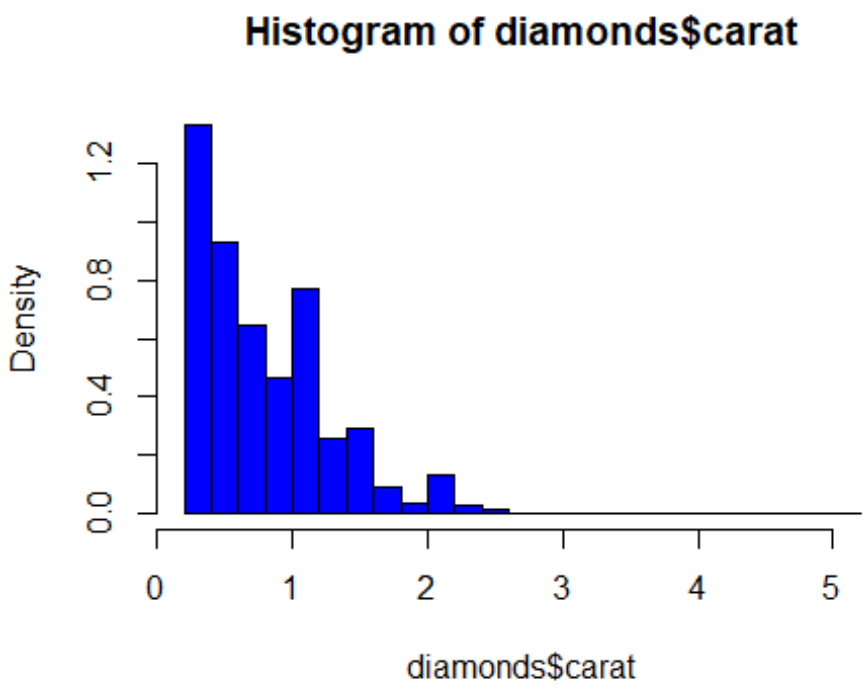

```
plot(diamonds$clarity, xlab="Diamonds Distribution by clarity",col = 'green')
```



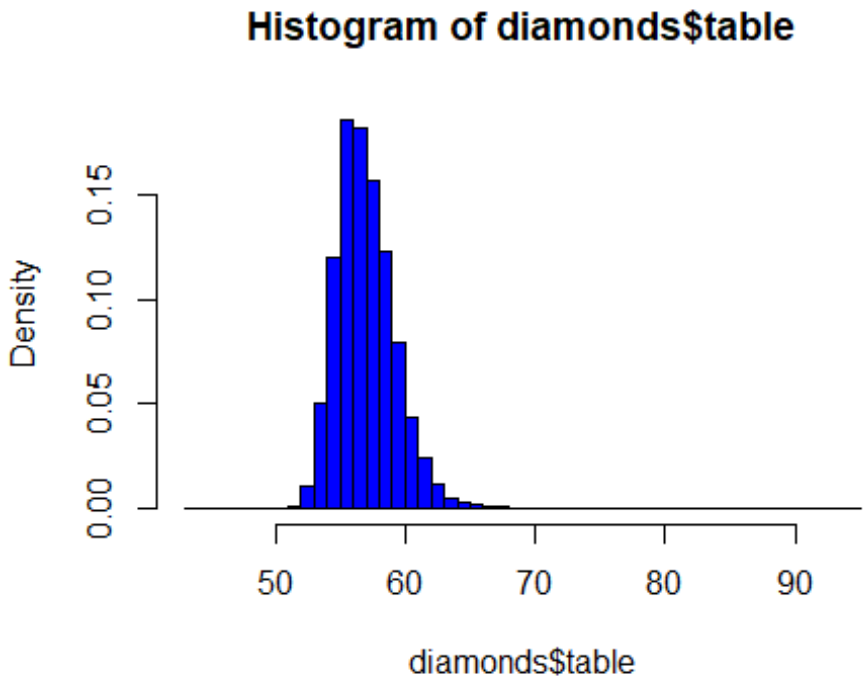
ces graphique valident les conclusion qu'on a fait avant concernant les variables qualitatives.

Regardons maintenant la distribution des variables quantitatives :

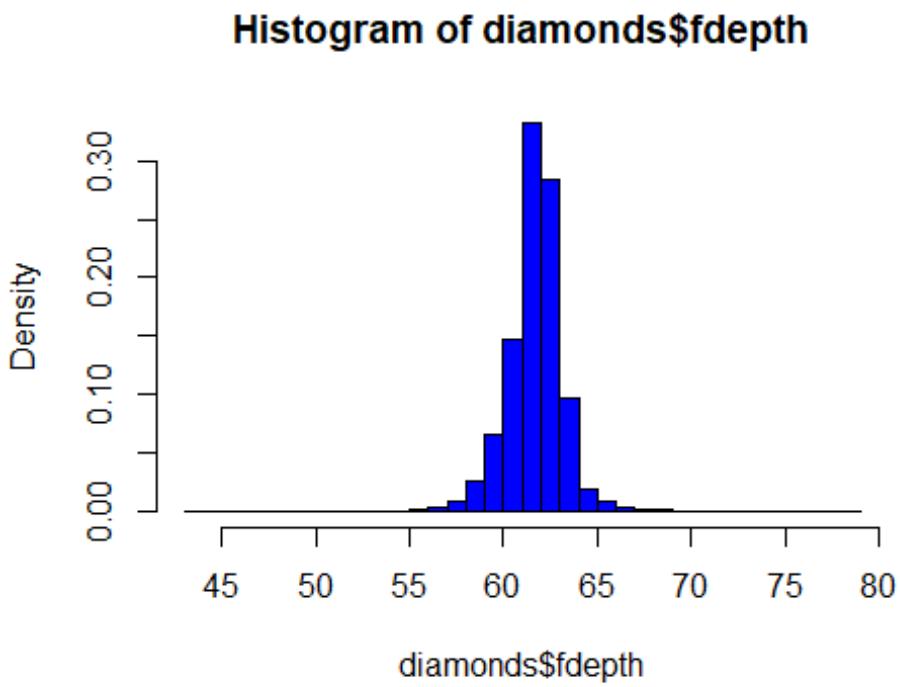
```
hist(diamonds$carat,breaks=25,col = 'blue',prob=TRUE)
```



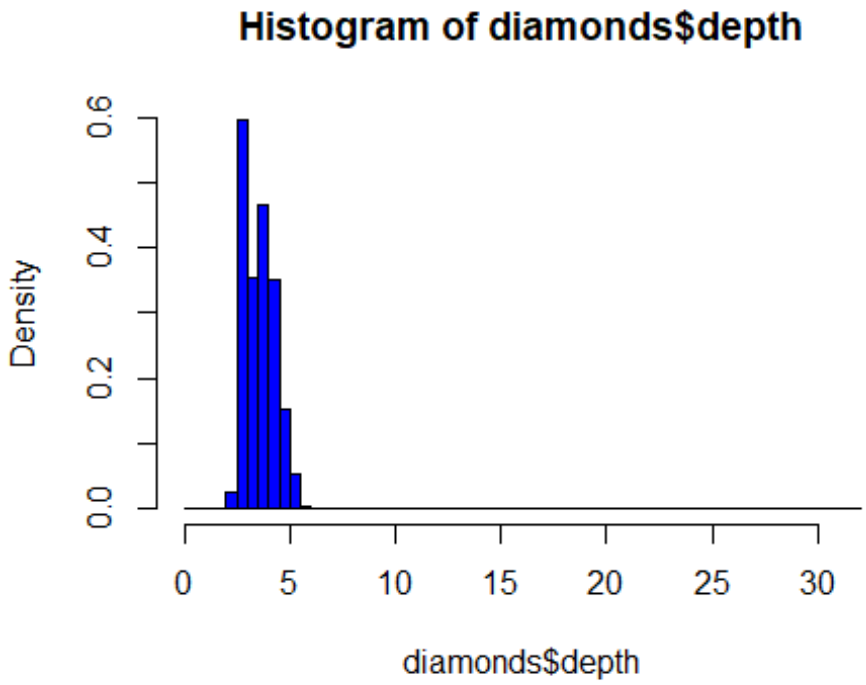

```r
hist(diamonds$table,breaks=50,col = 'blue',prob=TRUE)
```



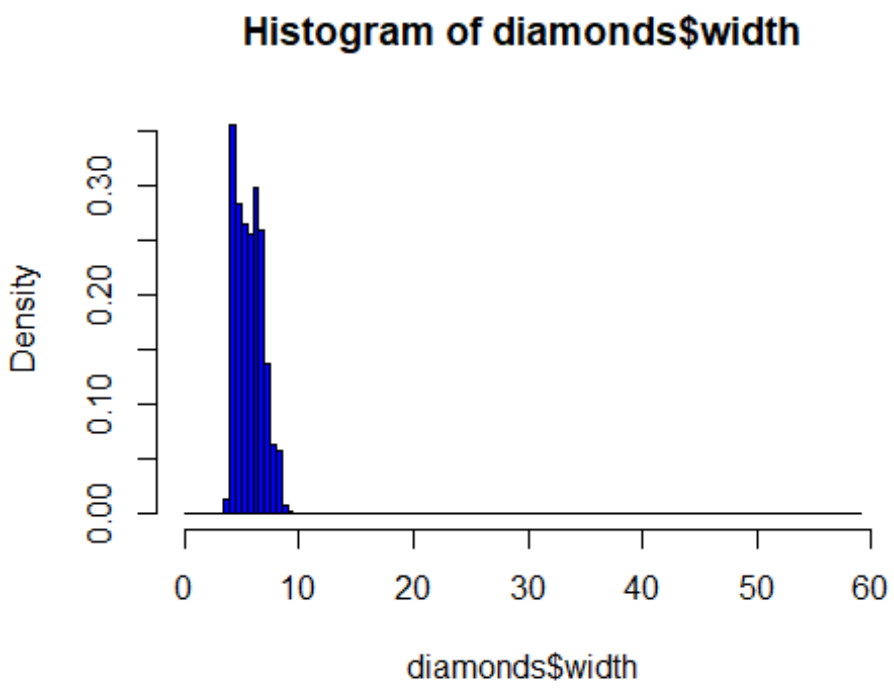

```r
hist(diamonds$fdepth,breaks=50,col = 'blue',prob=TRUE)
```



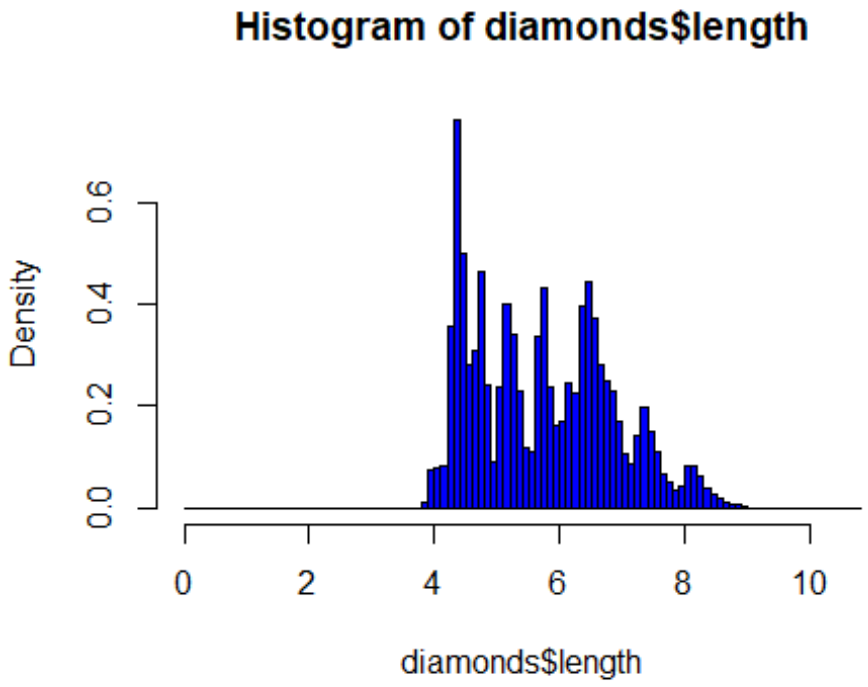

```r
hist(diamonds$depth,breaks=50,col = 'blue',prob=TRUE)
```



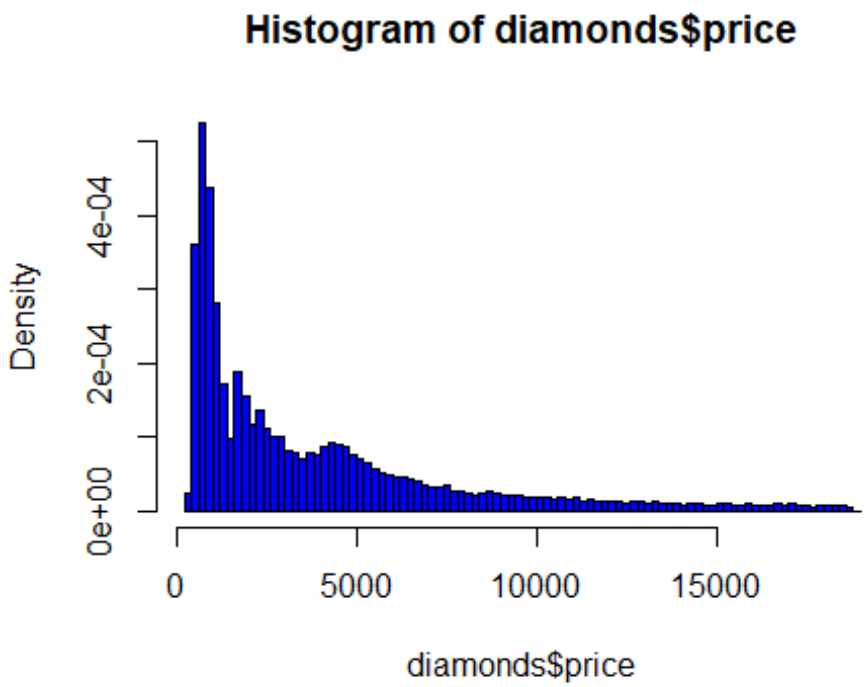

```r
hist(diamonds$width,breaks=100,col = 'blue',prob=TRUE)
```




```r
hist(diamonds$length,breaks=100,col = 'blue',prob=TRUE)
```




```r
hist(diamonds$price,breaks=100,col = 'blue',prob=TRUE)
```

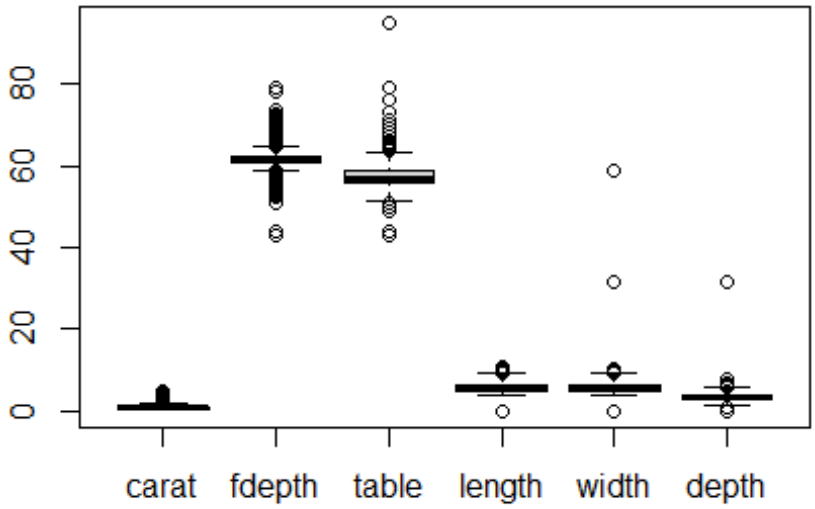


La majorité des diamants on des poids compris entre 0.2 et 1.5 carats, ce qui correspond à des poids en gramme compris entre 0.04 et 0.3. Nous avons donc à faire dans cette base de données principalement à de petits diamants.

Les variables **table**, **fdepth**, **depth** et **width** ont des distributions gaussiennes alors que la variable **length** pas du tout. La longeur des diamants de ce jeu de données est donc très hétérogène comparé aux deux autres variables de mesure que sont **depth** et **width** .

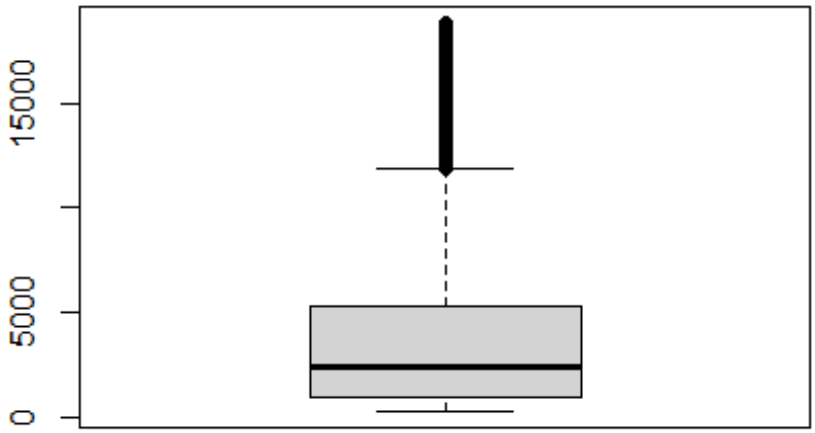
regardant les boxplots des variables quantitatives de notre jeu de donnée:

```
boxplot(diamonds[,c(1,5,6,8,9,10)])
```



On voit **fdepth** et **table** ont beaucoup de points aberrantes. Les variables **width** et **depth** également mais dans une moindre mesure. On veillera dans la suite à mettre ces valeurs aberrant en supplémentaire pour les utiliser lors de l'interprétation et non lors de l'apprentissage du modèle. (Il peut s'agir d'erreurs de mesure et pas d'une valeur extreme)

```
boxplot(diamonds$price)
```



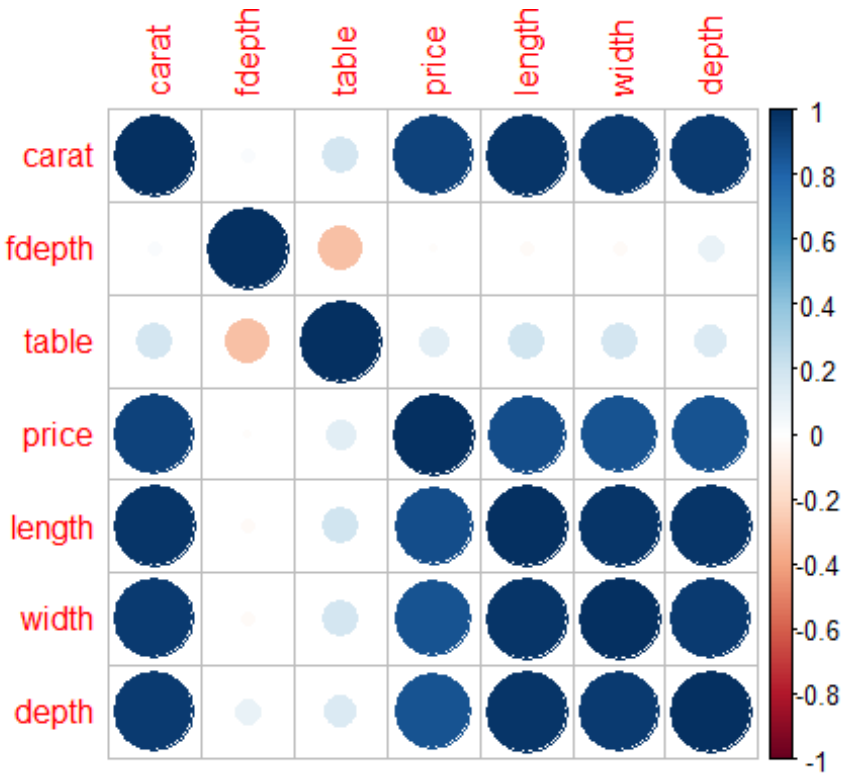
ce dernier graphe valide la remarque qu'on a fait avant donc la variable **price** contient beaucoup de points aberrantes et les grandes valeurs que prend la variable ne sont pas juste des valeurs extremes donc on va enlever les enregistrement associés à ces points.

```
data=diamonds
Q=quantile(data$price, probs=c(.25, .75), na.rm = FALSE)
iqr <- IQR(data$carat)
up <-  Q[2]+1.5*iqr # Upper Range
low<- Q[1]-1.5*iqr # Lower Range

data<- subset(data, data$carat> (Q[1] - 1.5*iqr) & data$carat <
(Q[2]+1.5*iqr))
```

4.2 Analyse bivaririée

Maintenant on s'intéresse au reltions entre les variable 2 à 2 autrement dit corrélation pour les variables quantitatives ,indépendance pour les variables qualitatives : ## **4.2.1 corrélations**

```
corrplot::corrplot(cor(diamonds[,-c(2,3,4)]))  #Corrélation des variables
quantitatives
```



On remarque grâce à ce graphique des corrélations que les variables **length**, **width**, **depth** et **carat** sont très corrélées positivement et qu'elles sont également toutes très corrélées positivement à la variable à expliquer **price**. Le prix d'un diamant semble donc à première vu être déterminé par ses dimensions et son poids (en carat), en excluant les variables qualitatives de l'étude.

De plus, on remarque une légère corrélation négative entre les variables **table** et **fdepth**.(on peut valider ou non ces remarques en faisant des tests de corrélation deux à deux)

4.2.1 indépendance

On effectue des test d'écart à l'indépendance deux à deux(test de khi2):

```
tab_cont1=table(diamonds$cut, diamonds$color)
chisq.test(tab_cont1)

##
##  Pearson's Chi-squared test
##
## data:  tab_cont1
## X-squared = 310.32, df = 24, p-value < 2.2e-16
```

Au vu de la très faible p-value du test du khi-deux d'indépendance entre la variable **cut** et la variable **color**, on en déduit que ce deux variables ne sont pas indépendantes (on rejette Ho)

```
tab_cont2=table(diamonds$cut, diamonds$clarity)
chisq.test(tab_cont2)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tab_cont2
## X-squared = 4391.4, df = 28, p-value < 2.2e-16
```

On obtient le même résultat ici, les variables **cut** et **clarity** ne sont pas indépendantes.

```
tab_cont3=table(diamonds$color, diamonds$clarity)
chisq.test(tab_cont3)
```

```
##
##  Pearson's Chi-squared test
##
## data:  tab_cont3
## X-squared = 2047.1, df = 42, p-value < 2.2e-16
```

Les variables **color** et **clarity** ne sont pas non plus indépendantes.

4.3 Analyse multivaririée

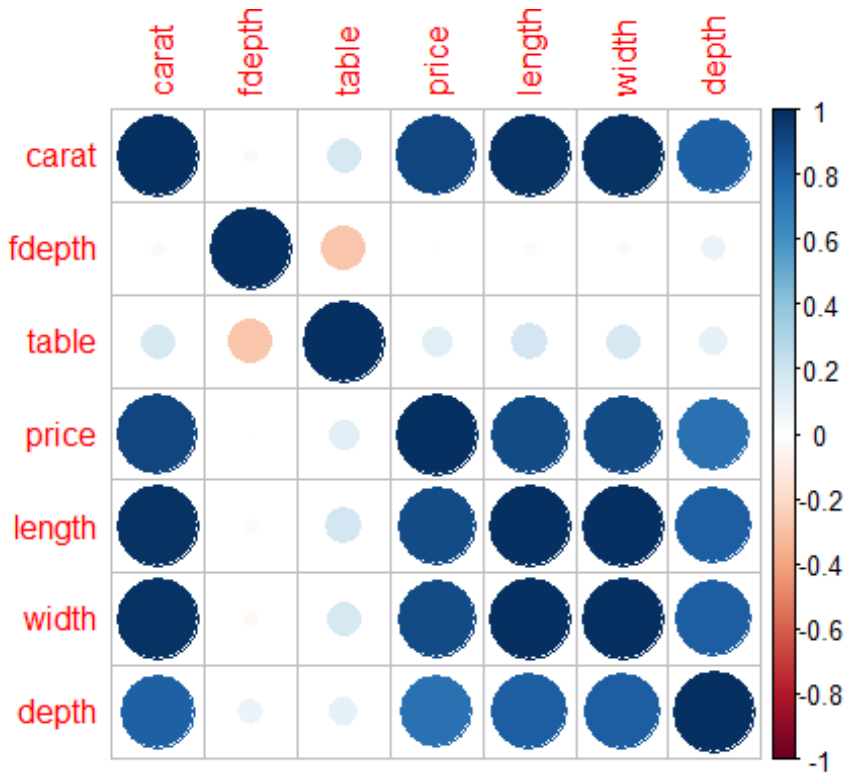
Dans cette partie étudier le jeu de données à l'aide de toutes les variables à la fois et non pas 2 à 2 ou une variable comme ce qu'on a fait avant mais d'abords prenons un échantillon qu'on appel data **data** de notre jeu de donnée initile pour rendre les temps de calculs plus rapides et les graphiques plus lisibles juste aprés avoir supprimer les valeurs aberrantes.

```
data=diamonds

Q=quantile(data$price, probs=c(.25, .75), na.rm = FALSE)
iqr <- IQR(data$price)
up <-  Q[2]+1.5*iqr # Upper Range
low<- Q[1]-1.5*iqr # Lower Range
data<- subset(data, data$price> (Q[1] - 1.5*iqr) & data$price <
(Q[2]+1.5*iqr))

#echantillonage
n=dim.data.frame(data)[1]
Ind=seq.int(1,n,10)
data=data[Ind,]
data
```

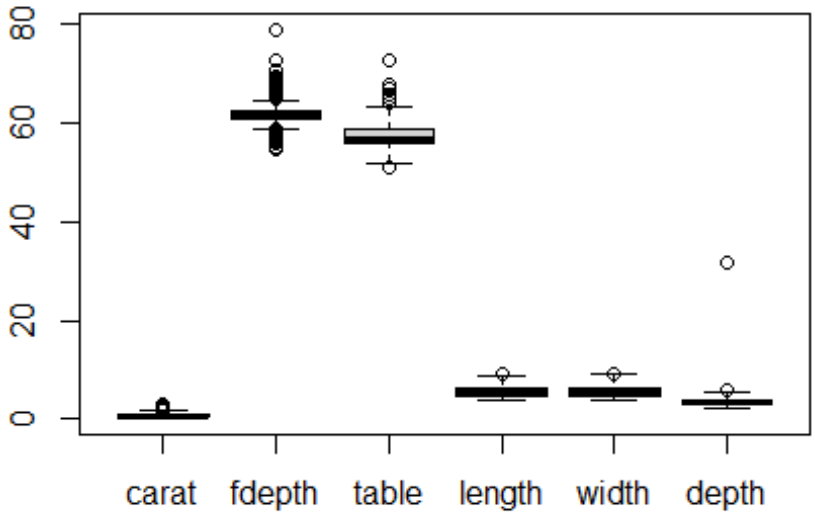
```
## # A tibble: 5,040 x 10
##    carat cut       color clarity fdepth table price length width depth
##    <dbl> <ord>     <ord> <ord>    <dbl> <dbl> <int>  <dbl> <dbl> <dbl>
##  1  0.23 Ideal     E     SI2       61.5    55   326   3.95  3.98  2.43
##  2  0.3  Good      J     SI1       64      55   339   4.25  4.28  2.73
##  3  0.3  Good      I     SI2       63.3    56   351   4.26  4.3   2.71
##  4  0.23 Very Good F     VS1       60      57   402   4     4.03  2.41
```

```
##  5  0.33 Ideal     I     SI2       61.2    56   403   4.49  4.5   2.75
##  6  0.24 Very Good F     SI1       60.9    61   404   4.02  4.03  2.45
##  7  0.35 Ideal     I     VS1       60.9    57   552   4.54  4.59  2.78
##  8  0.24 Very Good D     VVS1      61.5    60   553   3.97  4     2.45
##  9  0.26 Very Good E     VVS1      63.4    59   554   4     4.04  2.55
## 10  0.7  Ideal     E     SI1       62.5    57  2757   5.7   5.72  3.57
## # ... with 5,030 more rows
```

```
corrplot::corrplot(cor(data[,-c(2,3,4)]))
```




```
boxplot(data[,c(1,5,6,8,9,10)])
```



On peut voir que l'échantillon garde plus ou moins les mémes corrélations qu'au jeu de données initial et data ne contient que 5040 lignes,10 variables.

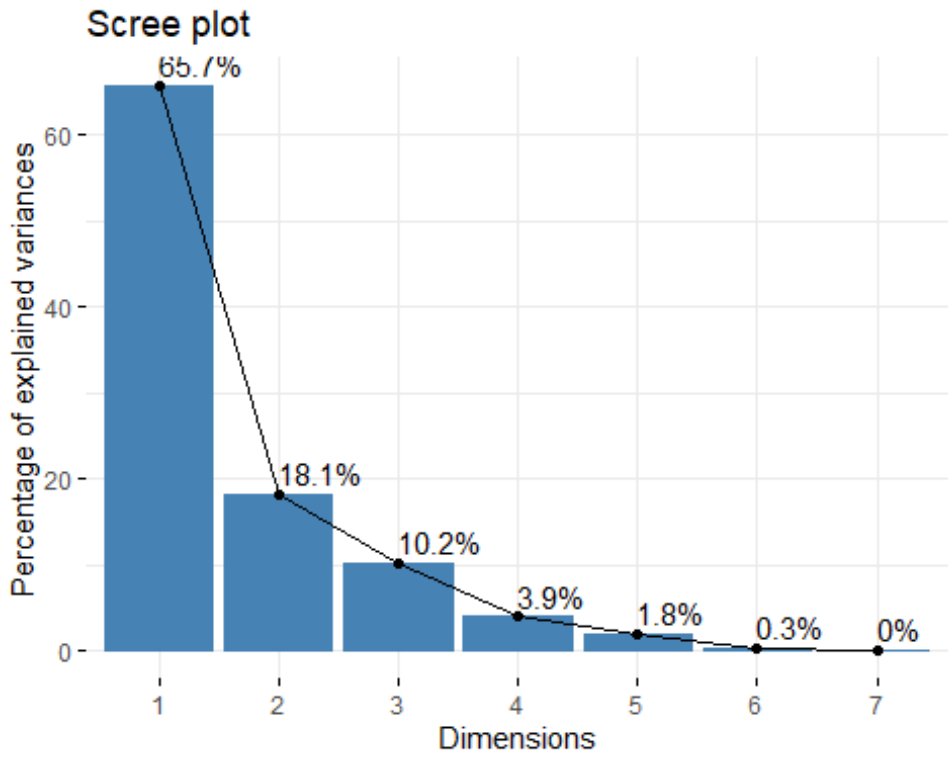
On fait maintenant à une analyse en composant principales sur notre échantillon:

```
data_acp_res=PCA(data,quali.sup=c(2,3,4),graph = FALSE)
```

–Choix des axes:

Regardant les pourcentage d'inirtie expliqués par chaqu'un des axes:

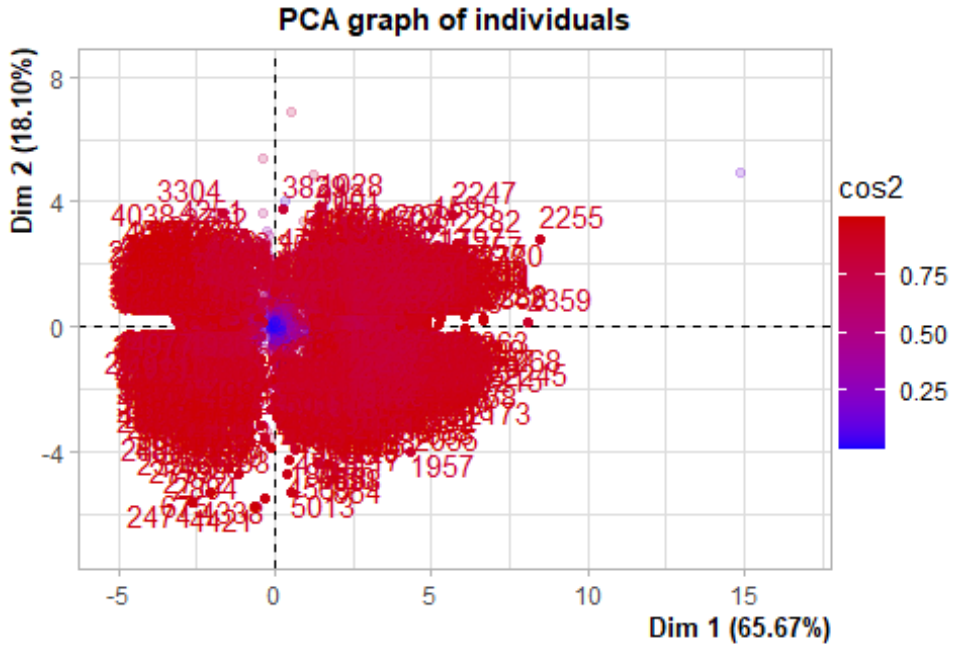
```
fviz_eig(data_acp_res, addlabels = TRUE)
```



On voit que les 2 premiers axes factoriels expliquent presque 84% de la variabilité de l'echantillon donc on garde uniquement le premier plans factoriel durant cette analyse.

– Graphe des individus:

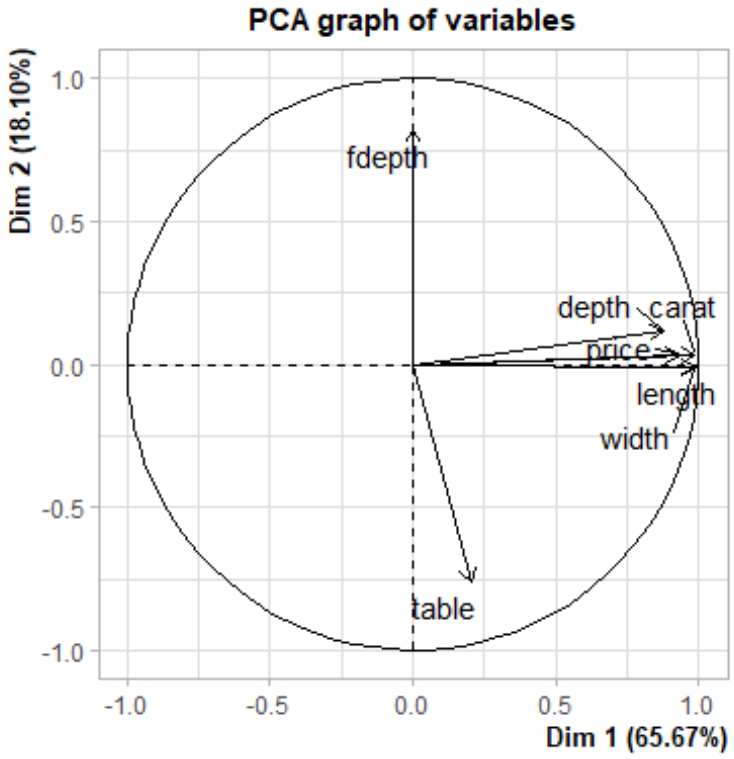
```
plot.PCA(data_acp_res,choix="ind",select="cos2 0.8",unselect=0.8,
habillage='cos2',invisible='quali')
```



On voit que le nuage des individus se répartit en 4 groupes chaque groupe dans un cadran du plan factoriel Nous voyons donc que plusieurs individus ont disparu de la représentation (notamment au centre), ce qui rend le nuage encore plus clairement divisé en 4 régions.

– Graphe des variables

```
plot.PCA(data_acp_res,choix="var",select="cos2 0.6",)
```



On voit alors que toutes les variables sont relativement bien représenté.

L'axe 1: est positivement corrélé aux variables: **price**, **carat**, **depth**, **width** et **length** cela confirme ce la remarque qu'on a fait dans la section précédente (analyse bivariée).On peut dire que l'axe 1 met en oppostion les diamants ayant des prix élevés, une longueur, largeur et profondeur importante et un poids (en carat) élevé aux diamants ayant un prix faible, une longueur, largeur, profondeur faible et un poids (en carat) faible.

L'axe 2:fortement lié à la variable **fdepth**(corrélation positive) ,à la variable **table**(corrélation négative).On peut dire que cette axe me en oposition les diamants ayant une forte de valeur pour la variable **fdepth** et une faible valeur pour la variable **table** et ceux ayant une faible valeur pour **fdepth** mais une forte pour **table**.

conclusion:

Nous venons de montrer qu'il y a beaucoup de corrélations entre les variables quantitatives et que les 3 variables explicatives qualitatives sont non-indépendantes. Cela nous posera le problème d'interprétation des résultast lors de nos différentes régressions qu'on va voir par qu'on veut effectuer par la suite à cause de la multicolinéarité. Il nous faudra donc certainement trouver un modèle pouvant s'affranchir de ces problèmes de multicolinéarité.

5. Analyse économétrique

Dans cette partie on va s'interesser à la variable **price** et on cherche à prédire cette variable (variable réponse) à partir des variables réstantes dans notre jeu de donnée (variable explicatives°.Pour cela on va utiliser plusieurs modéle plusieurs régressions multiples.

5.1. Régression linéaire multiple(moindre carré ordinaire)

Dans une régression linéaire multiple, la variable à expliquer Y est une variable quantitative et toutes les variables explicatives le sont également.

De plus, pour que les résulats d'une régression multiple soient exploitables, il faut que le modèle (Y=Xbeta+epsilon) vérifient plusieurs postulats:

1. P1 : les résidus epsilon sont centrés (espérance nulle)
2. P2 : les résidus ont même variance (homoscédasticité)
3. P3 : les résidus sont indépendants entre eux
4. P4 : les résidus sont des variables aléatoires Gaussiennes (param (0,sigma2))

Vérification des Postulats

```
#Reg_1=lm(data$price~data$carat + data$fdepth + data$table + data$length +
data$width+ data$depth)
Reg_1=lm(data$price~data$carat + data$fdepth + data$table + data$length +
data$width+ data$depth)
vif(Reg_1)
```

```
##  data$carat data$fdepth  data$table data$length  data$width  data$depth
##   33.039702    1.270024    1.144729  415.403202  398.323111    3.283232
```

On remarque grâce au test vif que nos variables sont très corrélées entre elles (on le savait déjà grâce à l'ACP effectuée précédemment), cela va donc sûrement poser problème pour la régression multiple qui supporte très mal les soucis de multicolinéarité.

Vérifions si les résidus sont distribués de manière gaussienne:

```
res=Reg_1$residuals
shapiro.test(res[1:5000]) #on ne prend que les 5000 premiers résidus car le
test sur R n'autorise pas plus de 5000 observations
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res[1:5000]
## W = 0.86487, p-value < 2.2e-16
```

On voit bien que le test de Shapiro réfute violamment cette hypothèse.

Vérifions l'homoscédasticité des résidus:

```
bptest(Reg_1)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  Reg_1
## BP = 901.68, df = 6, p-value < 2.2e-16
```

En regardant de la valeur de la p_value, on rejette également l'hypothèse nulle d'égalité des variances des résidus.Vu que la variance n'est pas constante on essaie avec le modéle suivant(**log(price)** comme variable réponse) :

```
Reg_2=lm(log(data$price) ~ data$carat + log(data$fdepth) + log(data$table)
+data$length + data$width + data$depth)
res2=Reg_2$residuals
vif(Reg_2)
```

```
##       data$carat log(data$fdepth)  log(data$table)      data$length
##        33.102616         1.277858         1.151526        415.207181
##       data$width       data$depth
##       397.907354         3.283225
```

```
hist(res2,probability = T) #vérification graphique
```


```
shapiro.test(res2[1:5000])
```

```
##
##  Shapiro-Wilk normality test
##
## data:  res2[1:5000]
## W = 0.99856, p-value = 0.0001764
```

```
bptest(Reg_2)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  Reg_2
## BP = 19.041, df = 6, p-value = 0.004094
```

```
dwtest(Reg_2)
```

```
##
##  Durbin-Watson test
##
## data:  Reg_2
## DW = 1.5878, p-value < 2.2e-16
## alternative hypothesis: true autocorrelation is greater than 0
```

Malheureusement, même en passant au log du prix, les résidus ne vérifient toujours aucun des postulats (même si l'on note une nette amélioration pour la normalité et l'homoscédasticité). Même en essayant plusieurs transformations sur notre variable Y et nos variables explicatives (suppression des variables très corrélées, variables au carré, formules de boxCox), nous n'obtenons jamais vérification des postulats par les résidus. Nous sommes donc obligées à ce stade d'abandonner la régression multiple car non pertinente.

Nous allons maintenant effectuer des régressions différentes de la régression multiple en ce que celles ci peuvent régler les problèmes de multicolinéarité des variables.

5.2. Régression sur Composantes Principales

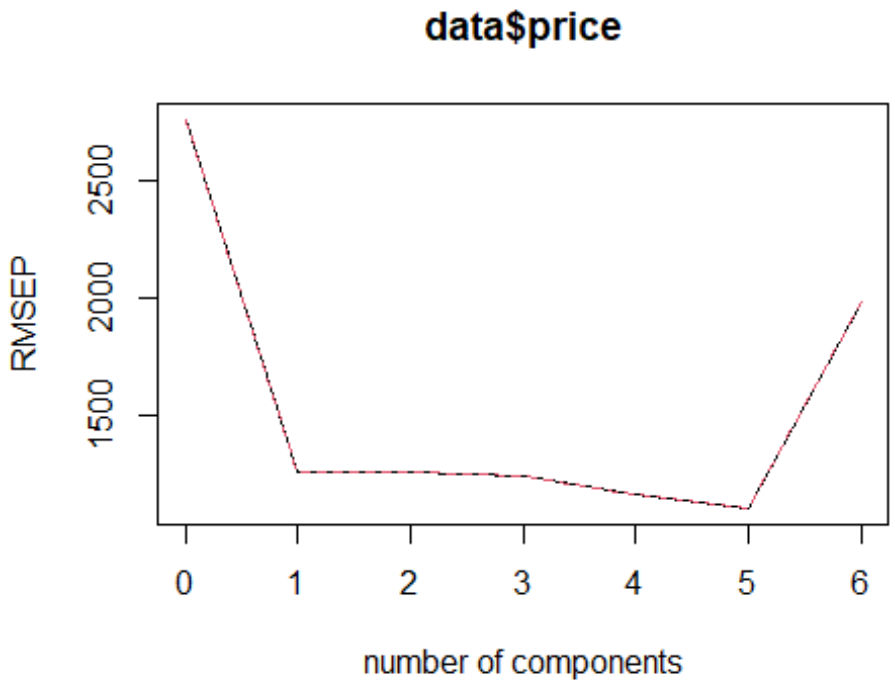
Dans ce modèle on combine entre régression linéaire multiple et analyse en composantes principales en faisant une regression de la variable reponse sur non pas les variables explicatives initial mais les variable artificielles de l'ACP (Axes factoriels):

```
Reg_pcr=pcr(data$price~ data$carat + data$fdepth + data$table + data$length +
data$width + data$depth, scale=T, validation ='LOO') #On standardise les
données car elles n'ont pas toutes la même unité
summary(Reg_pcr)
```

```
## Data:    X dimension: 5040 6
##  Y dimension: 5040 1
## Fit method: svdpc
## Number of components considered: 6
##
```

```
## VALIDATION: RMSEP
## Cross-validated using 5040 leave-one-out segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV            2764     1259     1254     1239     1159     1103     1982
## adjCV         2764     1259     1254     1239     1159     1103     1981
##
## TRAINING: % variance explained
##             1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## X             62.66    83.76    95.59    99.64    99.98   100.00
## data$price    80.08    80.24    80.73    82.55    84.38    84.54
```

```
validationplot(Reg_pcr)
```

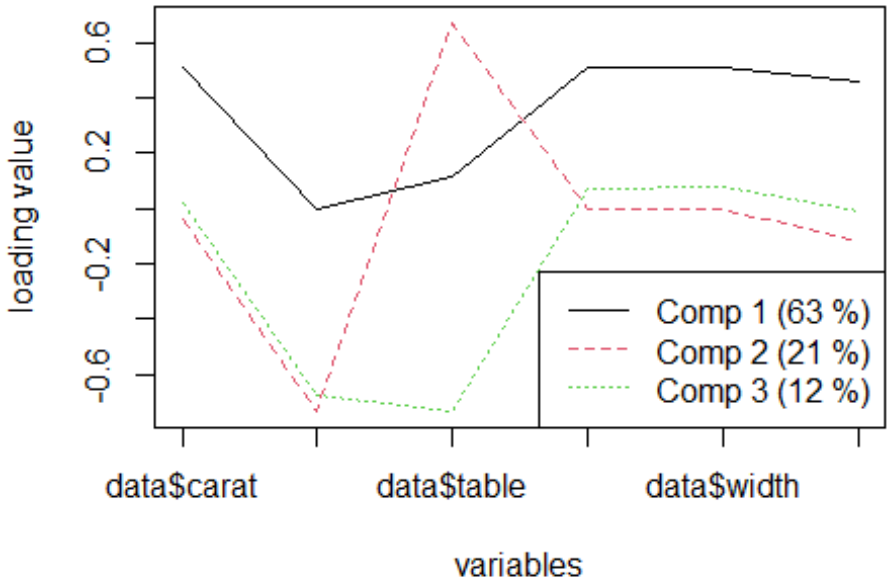


Le résumé de cette régression PCR nous montre que le nombre de composantes à retenir pour minimiser l'indicateur PRESS est de 6. Il n'y a donc aucune réduction de dimension. En regardant bien les pourcentages de variances expliquées pour X et Y, nous décidons de ne retenir que les 3 premières composantes car celles-ci expliquent 95.6% de la variance de X et plus de 80% de la variance de Y, ce qui est plutôt bon.

Signification des axes

Pour définir chaque axe retenu, nous allons représenter le graphe des loadings.

```
plot(Reg_pcr, "loadings", comps = 1:3, legendpos = "bottomright",
     labels = "names", xlab = "variables")
```




```
Yloadings(Reg_pcr)[,1:3]
```

```
##     Comp 1     Comp 2     Comp 3
## 1275.54611  -99.92297  230.01673
```

On retrouve donc les résultats de l'ACP pour les 2 premières dimensions : l'axe 1 met en opposition les diamants ayant des mesures (poids, longueur, largeur, profondeur) élévées à ceux en ayant des plus faibles.

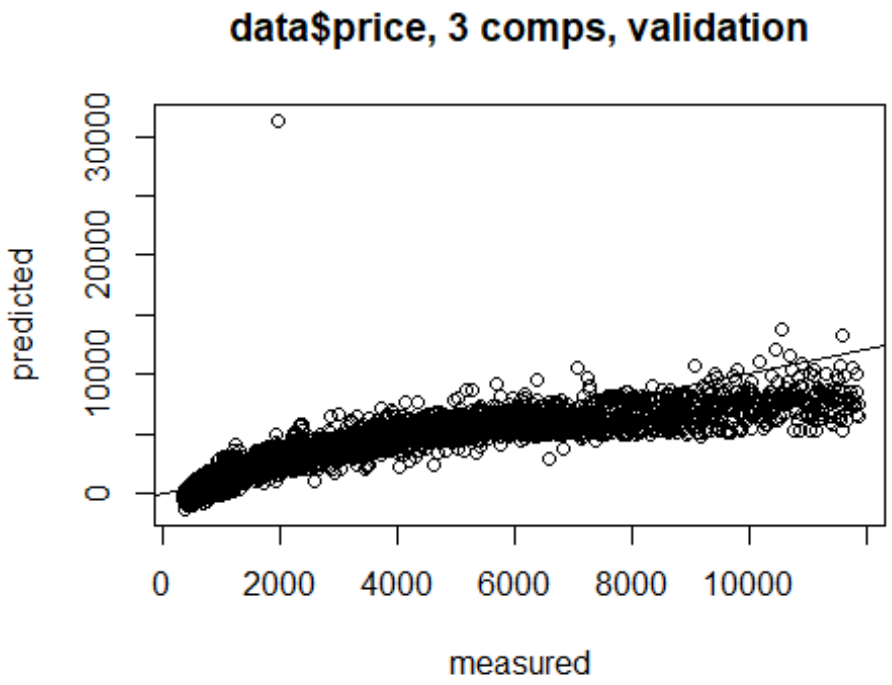
Pour l'axe 2, on trouve une forte corrélation positive avec la variable **table** et une forte corrélation négative avec la variable **fdepth** . L'axe 2 met donc en opposition les diamands ayant une forte valeur de **table** et une faible valeur de **fdpeth** à ceux ayant au contraire une faible valeur de **table** mais une valeur de **fdepth** élevée.

Enfin, l'axe 3 met lui en opposition les diamants ayant de forte valeurs de **fdepth** et **table** à ceux ayant des faibles valeurs pour les deux variables.

La variable à expliquer Y est elle très positivement corrélée à la dimension 1, comme déjà vu lors de l'ACP. Il semblerait donc que les principaux facteurs explicatifs du prix d'un diamants soit son poids (en carat) et ses dimensions.

Pouvoir prédictif du modèle

```
predplot(Reg_pcr,ncomp=3, line=T)
```



On voit grâce à ce graphique que la prédiction faite par notre modèle n'est pas très satisfaisante. La présence de prix prédits négatifs montre que le modèle peut surement être amélioré. Il semble aussi que le modèle estime bien les prix des diamands vendus entre 1000 et 8000 dollars mais beacoup moins bien les autres fourchettes de prix.

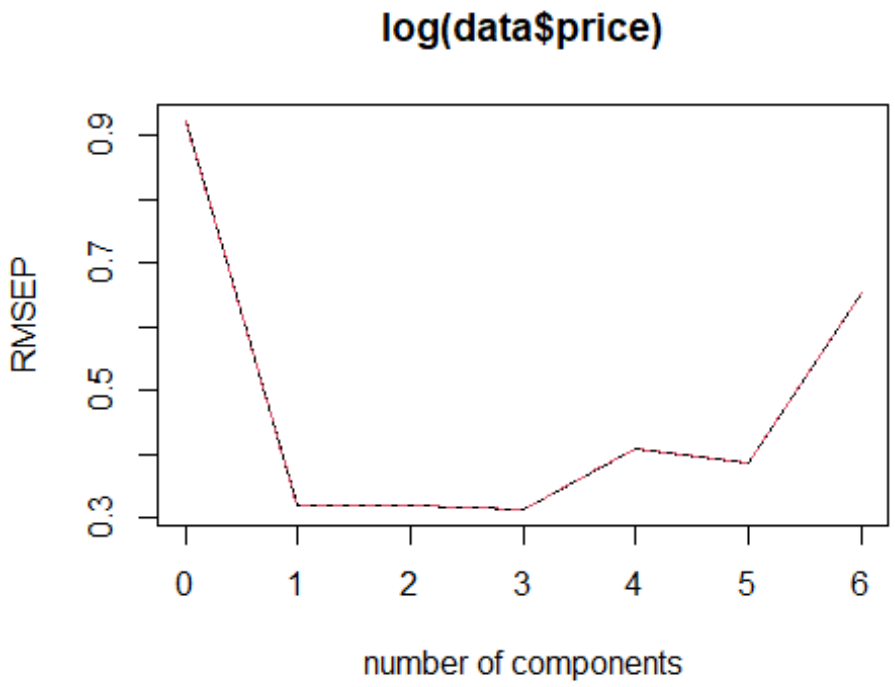
On va essayer de voir si expliquer le log du prix au lieu du prix lui-même améliore le modèle:

```
Reg_pcr2=pcr(log(data$price)~data$carat + data$fdepth + data$table +
data$length + data$width + data$depth, scale=T, validation ='LOO') #On
standardise les données car elles n'ont pas toutes la même unité
summary(Reg_pcr2)
```

```
## Data:    X dimension: 5040 6
##  Y dimension: 5040 1
## Fit method: svdpc
## Number of components considered: 6
##
## VALIDATION: RMSEP
## Cross-validated using 5040 leave-one-out segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV          0.9235   0.3205    0.319   0.3142   0.4101   0.3854   0.6521
## adjCV       0.9235   0.3205    0.319   0.3142   0.4101   0.3854   0.6520
##
## TRAINING: % variance explained
##                   1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
```

```
## X                  62.66    83.76    95.59    99.64    99.98   100.00
## log(data$price)    88.82    88.95    89.31    90.46    92.66    92.66
```

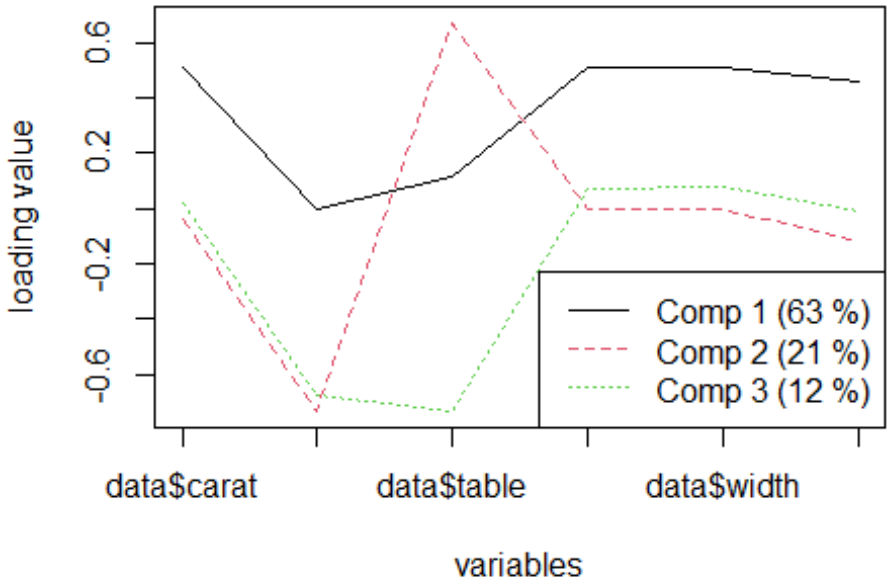
```
validationplot(Reg_pcr2)
```



Encore une fois, il faut 6 composantes pour minimiser l'indicateur PRESS mais on se rend compte que 4 composantes donne quasiment la même valeur. On voit donc le début d'une réduction de dimension. En observant les pourcentages de variances expliquées, en gardant également 3 composantes, on explique cette fois-ci plus de 89.31% de la variance de Y (log(price)) et plus de 95.6% de la variance de X. Ce modèle semble donc meilleur.

Vérifions si la signification des axes est toujours la même avec ce modèle:

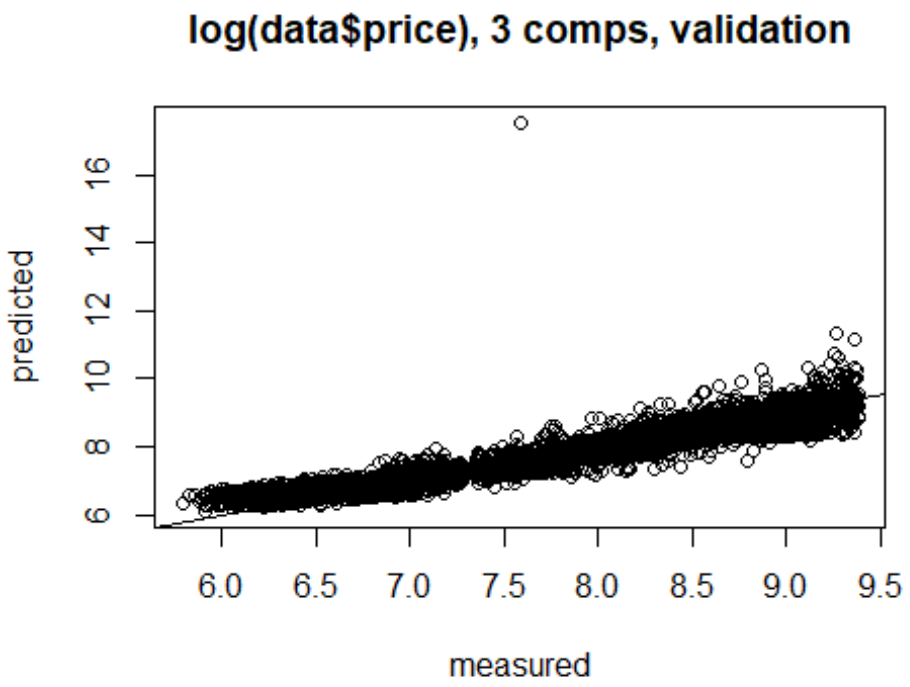
```
plot(Reg_pcr2, "loadings", comps = 1:3, legendpos = "bottomright",
     labels = "names", xlab = "variables")
```




```
Yloadings(Reg_pcr2)[,1:3]

##       Comp 1      Comp 2      Comp 3
##  0.44884214 -0.03026188  0.06543169
```

La définition des axes est inchangée et la variable Y (log(price)) est toujours plus corrélée à l'axe 1, donc on conserve les interprétations faites plus haut.

```
predplot(Reg_pcr2,ncomp=3, line=T)
```



On obtient cette fois une courbe beaucoup plus proche d'une droite. Nous allons donc garder ce modèle.

Coefficients du modèle

```
coef(Reg_pcr2,ncomp=3)
```

```
## , , 3 comps
##
##              log(data$price)
## data$carat        0.22979307
## data$fdepth      -0.02233260
## data$table       -0.01661784
## data$length       0.23347124
## data$width        0.23399176
## data$depth        0.20932185
```

Les coefficients de la régression permettent de quantifier l'impact de chaque variable sur la variable à expliquer Y(**log(price)**) lorsque l'on prend en compte 3 composantes principales.

Le prix d'un diamant est donc fortement positivement lié à sa largeur, sa longueur, son poids et sa profondeur. Par exemple, si l'on augmente d'une unité la variable carat, le log du prix du diamant augmente 0.23, toutes choses égales par ailleurs.

En revanche, le prix est négativement corrélé aux variables **fdepth** et **table**. Si l'on augmente d'une unité la variable **fdepth** alors le log du prix du diamant diminue de 0.022, toutes choses égales par ailleurs.

Conclusion

Cette méthode nous a donc permis de trouver un modèle satisfaisant pour expliquer non pas le prix mais le log du prix des diamants de la base de données.
Avec cette méthode, on voit que ce sont les dimensions et le poids d'un diamant qui influence le plus positivement sont prix. Des valeurs de table ou de fdepth trop grandes auront en revanche tendance à le faire diminuer.
Cependant,on peut encore pousser notre étude car la régression sur composantes principales est une méthode qui maximise uniquement la variance des X (variables explicatives) et non pas la variance de Y. On va donc pour finir notre étude, utiliser une autre méthode de régression sur variables latentes, la PLS.

5.3 Partial Least Squares Regression

Dans cette méthode est on fait une régresssion linéaire sur des axes artificiels qui maximise non plus uniquement la inertie des variables explicative(cas précédent: méthode **PCR**) noté X mais la covariance de X et la variable réponse Y (méthode: **PLS**).Cette méthode est trés consistente pour résoudre le problème de multicolinéarité

On va effectuer une régression PLS en utilisant uniquement les variables explicatives quantitatives, pour pouvoir comparer les résultats obtenus aux résultats de la pcr effectuée précédemment.

Régression PLS

On applique la méthode directement sur la variable **log(price)**:

```
Reg_pls=plsr(log(price) ~ carat + fdepth + table + length + width + depth,
data=data, scale=TRUE, validation="LOO")
summary(Reg_pls)
```

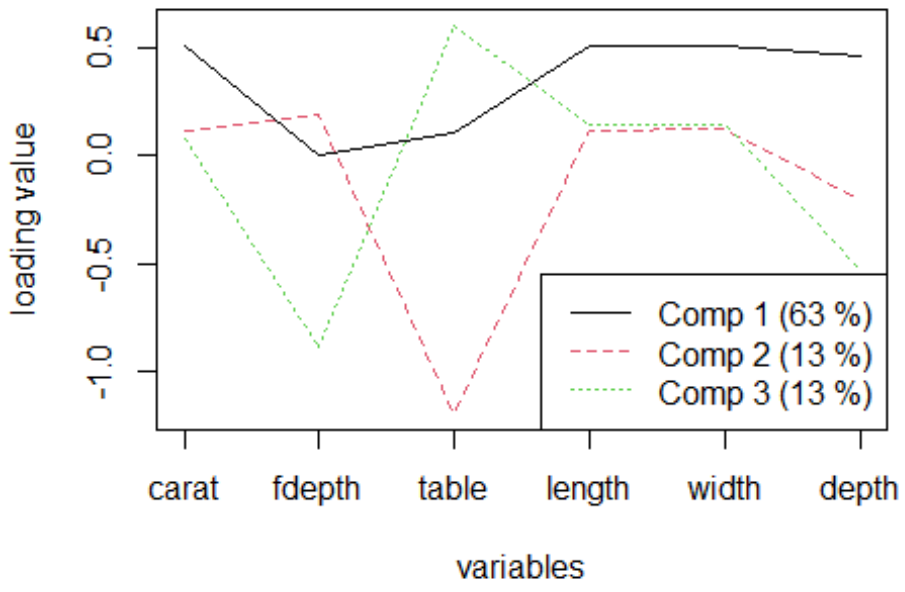
```
## Data:    X dimension: 5040 6
##  Y dimension: 5040 1
## Fit method: kernelpls
## Number of components considered: 6
##
## VALIDATION: RMSEP
## Cross-validated using 5040 leave-one-out segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV          0.9235   0.3182   0.3182   0.3209   0.4074   0.5104   0.6521
## adjCV       0.9235   0.3182   0.3182   0.3209   0.4073   0.5104   0.6520
##
```

```
## TRAINING: % variance explained
##             1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## X             62.64    76.01    88.87    99.36    99.98   100.00
## log(price)    89.18    90.21    90.76    91.07    92.66    92.66
```

Là encore, 6 composantes minimisent l'indicateur PRESS mais 4 composantes seulement donnent quasiment les mêmes résultats. Pour pouvoir comparer cette régression à la pcr précédente, nous allons également conserver 3 axes. Les 3 premiers axes expliquent 89% de la variance des X et quasiment 90% de la variance de Y (log(price)). Ne garder que 3 axes est donc bien justifié ici.

Signification des axes

```
plot(Reg_pls, "loadings", comps = 1:3, legendpos = "bottomright",
     labels = "names", xlab = "variables")
```




```
Yloadings(Reg_pls)[,1:3]

##      Comp 1      Comp 2      Comp 3
## 0.45018526 0.12987365 0.09520688
```

En regardant les loadings on retrouve pour la dimension 1 exactement la même définition que lors de nos différentes pcr. L'axe 1 met toujours en oppposition les diamants ayant de fortes valeurs pour les variables **carat**, **length**, **width** et **depth** aux diamants ayant de faible valeurs pour ces variables.

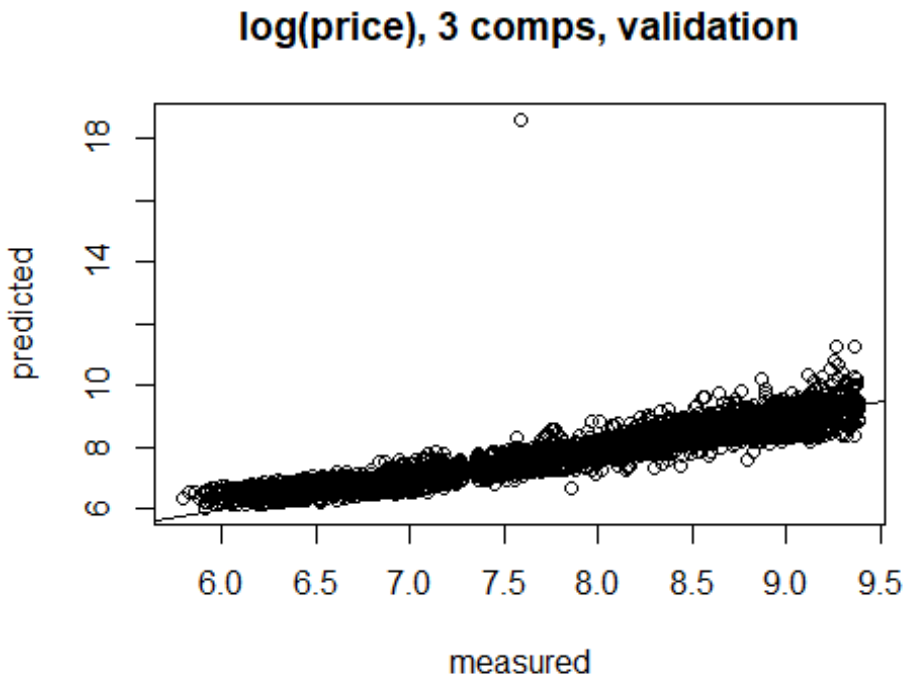
En revanche, la définition de l'axe 2 change. Lors de notre pcr, l'axe 2 mettait en opposition les diamants ayant une forte valeur pour la variale **table** et une faible valeur pour la variable **fdepth** à ceux ayant les caractéristiques inverses. Ici, la variable **table** est fortement négativement corrélée à l'axe 2 et la variable **fdepth** est plus faiblement positivement corrélée à l'axe 2. On a donc inversé le sens de la dmension 2 par rapport à la pcr.

Le 3ème axe est quant à lui uniquement définit par la variable **fdepth**, négativement corrélée à l'axe. La dimension 3 met donc en opposition les diamants ayant une faible valeur de **fdepth** à ceux ayant une forte valeur.

Les loadings montrent encore une fois que la variable cible **price** est plus corrélée avec la dimension 1 qu'avec toutes les autres dimensions. Cependant, le coefficient pour l'axe 2 est cette fois-ci positif et le coefficient pour l'axe 3 est plus élevé que pour la pcr. La variable **fdepth** semble jouer un rôle plus important dans l'explication du prix des diamants .

Qualité de prédiction du modèle

```
predplot(Reg_pls,ncomp=3, line=T)
```



On obtient également ici une courbe quasi-linéaire. Le modèle est donc satisfaisant. Les 2 méthodes, pcr et pls, donnent donc à peu de choses près les mêmes résulats pour ce modèle.

Coefficients du modèle

```
coef(Reg_pls,ncomp=3)
```

```
## , , 3 comps
##
##         log(price)
## carat   0.20740513
## fdepth -0.02937322
## table  -0.03407141
## length  0.32402797
## width   0.32861207
## depth   0.03302983
```

On voit tout d'abord que les signes des coefficients sont les mêmes que pour la pcr mais la valeur du coefficient pour carat est plus faible.

En ne regardant que les résultats que l'on obtient ici, on peut dire que si l'on augmente la largeur d'un diamant **width** d'une unité (le mm ici en l'occurence) alors le log du prix augmente de 0.32, toutes choses égales par ailleurs. Enfin, si l'on augmente d'une unité le variable **table** alors le log du prix du diamant diminue de 0.034, toutes choses égales par ailleurs.

Conclusion

Avec cette régression PLS les résultats les plus souvent revenus dans toute notre étude. Ce sont les variables **carat**, **length**, **width** et **depth** qui définissent le premier axe et qui sont le plus positivement corrélées à la variable à expliquer. Cependant, cette régression a aussi permis de montrer que la variable **fdepth** semble jouer un rôle plus important. Il en résulte que les coefficients de régression obtenus sont différents de ceux de la PCR. Nous avons interprété ces résultats dans la section précédente.

6. conclusion général

Pendant cette étude nous avons tirer les résultatat suivants:

ACP: L'analyse en composantes principales nous a elle permis de montrer que les diamants peuvent être séparés en 4 groupes, définis par les différentes variables quantitatives. Il est également apparu lors de cette analyse que ce sont les variables **carat**, **length**, **width** et **depth** qui sont le plus positivement corrélées à la variable **price** et qui influencent donc le plus positivement le prix d'un diamant (cette acp nous a donné une idée sur le mulcticolinéaritée entre le variables de de la base de données)

OLS: Malheureusement, nous n'avons pû réaliser de régression multiple sur nos données car celles-ci ne vérifient aucun des postulats nécessaires.

PCR: La régression PCR a elle aussi mis en avant la grande part explicative des variables sus-mentionnées dans l'explication du prix d'un diamant. De plus, cette régression a permis

de montrer qu'il était beaucoup plus judicieux et pertinent d'expliquer le log du prix d'un diamant par les variables quantitatives à notre disposition au lieu du prix lui-même.

PLS la régression PLS, bien qu'elle confirme les résultats déjà énoncés, a montré que la variable **fdepth** peut avoir une influence négative sur le prix assez importante.

Au final, on a trouvé deux fonctions qui nous permettent de prédire le prix d'un diamant.