Diamonds - Final Assignment

Aina Belloni

16/3/2021

Diamonds dataset

Searching on the Internet I found this dataset called `Diamonds` in the website Kaggle.com. I decided to take into consideration the information related to some diamonds contained in this file because I think it's interesting see how some variables that describe a diamond can influentiate the determination of the price of that diamond.

The dataset downloaded from the website was very big, I decided to reduce from the beginning the dimension of that from 53940 observations to 10000.

I also transformed all the variables containing numbers in numeric variables with the function `as.numeric` and the qualitative/categorical variables as factor variables using the function `as.factor`.

At the end I remove the first variable which was the ID number of the diamonds which simply matched the row number of the dataset.

The goal of this project is to understand what kind of data we are dealing with, determinate the best linear model that allows us to predict the price of any diamond and understand the problems that we can encounter during the analysis.

Load the data

```
setwd("~/CATTOLICA/Applied linear models/assignments")
diamonds<-read.csv2("diamonds.csv", header=T, sep=",",dec=".")
```

```
str(diamonds)
```

```
## 'data.frame':    10000 obs. of  10 variables:
##  $ carat  : num  0.51 2.04 0.32 1.04 0.31 1.58 0.91 0.38 0.41 0.51 ...
##  $ cut    : Factor w/ 5 levels "Fair","Good",..: 2 4 3 4 3 4 5 5 4 2 ...
##  $ color  : Factor w/ 7 levels "D","E","F","G",..: 2 5 2 4 4 7 5 6 4 1 ...
##  $ clarity: Factor w/ 8 levels "I1","IF","SI1",..: 6 4 6 5 6 3 5 4 6 4 ...
##  $ depth  : num  63.6 62.7 61.3 60.2 62.4 59.8 62.1 62.7 61.3 63.9 ...
##  $ table  : num  54 56 57 58 55 58 59 58 57 55 ...
##  $ price  : num  1662 15760 702 6360 698 ...
##  $ x      : num  5.09 8.12 4.42 6.65 4.32 7.61 6.17 4.57 4.8 5.1 ...
##  $ y      : num  5.06 8.06 4.45 6.64 4.3 7.57 6.2 4.61 4.76 5.04 ...
##  $ z      : num  3.23 5.07 2.72 4 2.69 4.54 3.84 2.88 2.93 3.24 ...
```

The dataset now includes information about 10000 diamonds, in particular we have 10000 rows/observations and 10 variables (columns). The variables are :

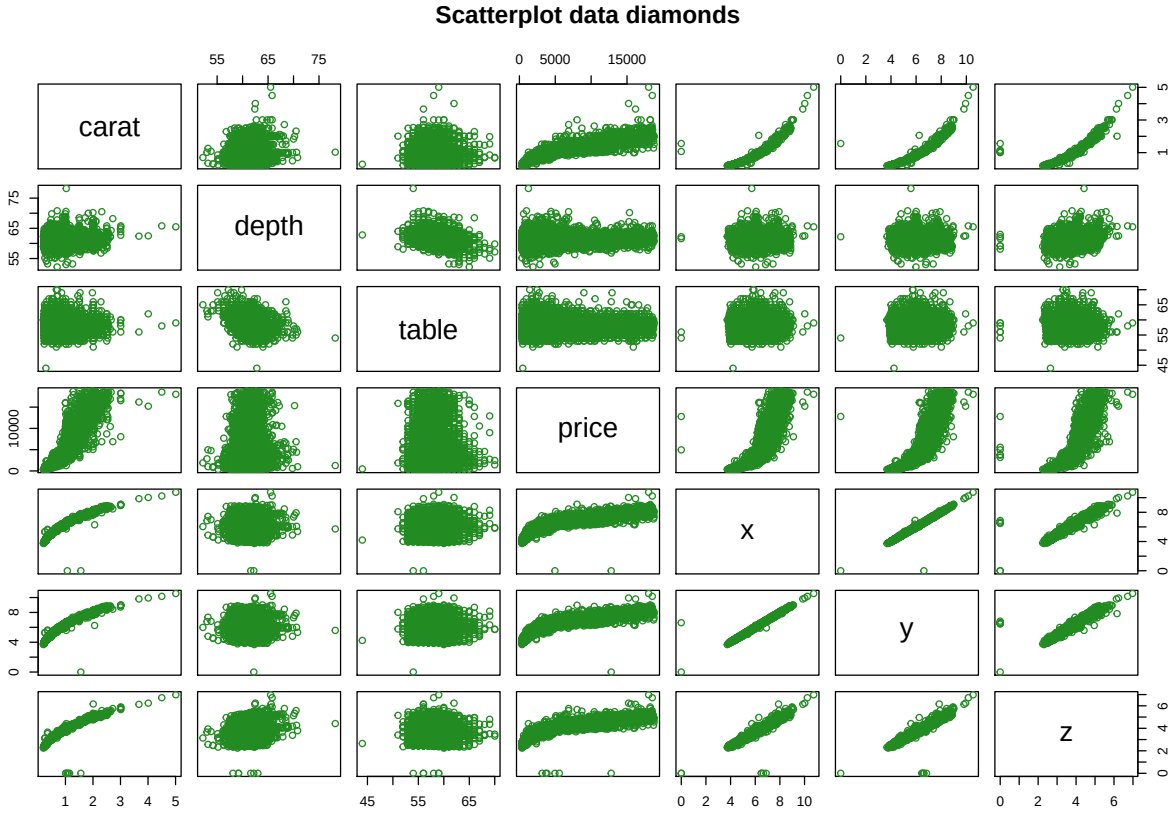
- `carat` : the number of carats, so the weight of the diamond

- `cut` : the quality of the cut (Fair, Good, Very Good, Premium, Ideal) where Fair is the worst and Ideal the best
- `color` : the color of the diamond (D,E,F,G,H,I,J) where D is the best and J the worst
- `clarity` : a measurement of how clear the diamond is (I1, SI2, SI1, VS2, VS1, VVS2, VVS1, IF) where I1 is the worst and IF the best.
- `depth` : total depth percentage
- `table` : width of top of diamond relative to the widest point
- `x`,`y` and `z` : the 3 dimensions of the diamond respectively length, width and depth

Exploratory analysis

Since the variables `cut`, `color` and `clarity` are categorical, I exclude them from the scatterplot matrix.

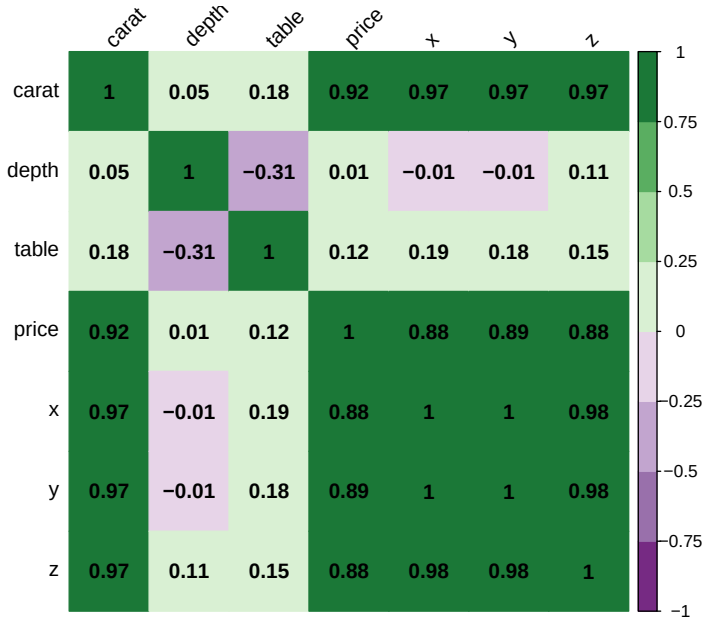
```
plot(diamonds[,-c(2,3,4)], col="forestgreen", main="Scatterplot data diamonds")
```



We see a strong linear correlation between the variables `x`,`y` and `z`, and also these variables and the `carat` and `price` ones. Another quite important relationship is between `price` and `carat` variables. For the others the relationship is unclear.

To deepen the analysis we can also see the correlation matrix.

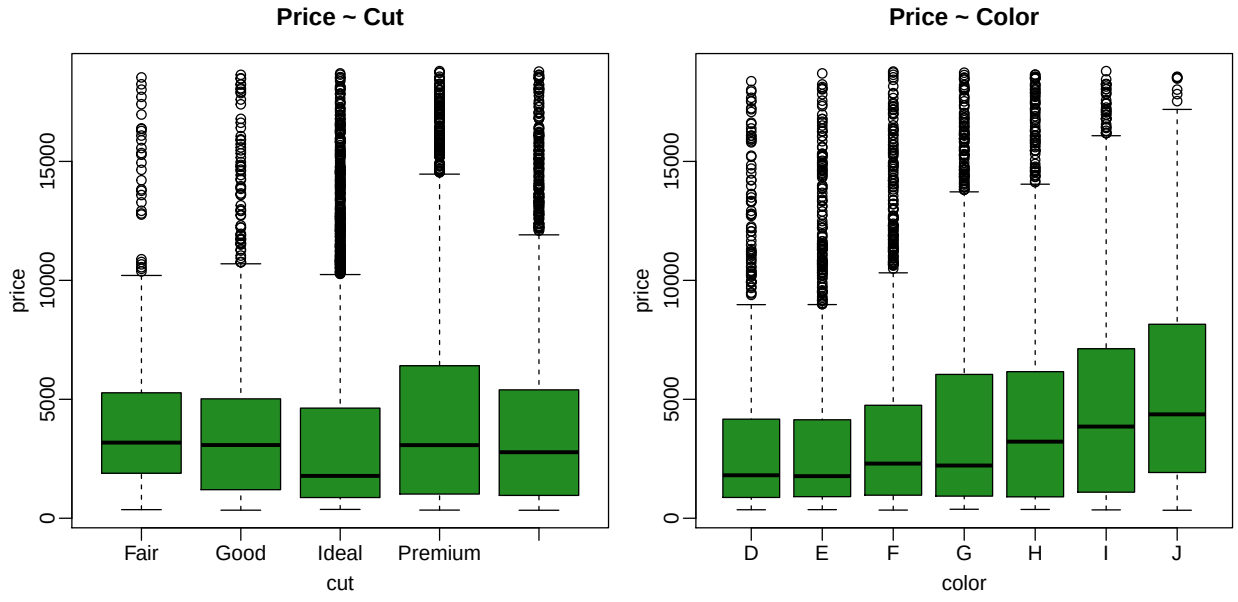
```
corr=cor(diamonds[,-c(2,3,4)])

library("corrplot")
corrplot(corr,method="color", addCoef.col="black", tl.col="black",tl.srt=45,
         col = brewer.pal(n = 8, name = "PRGn"))
```



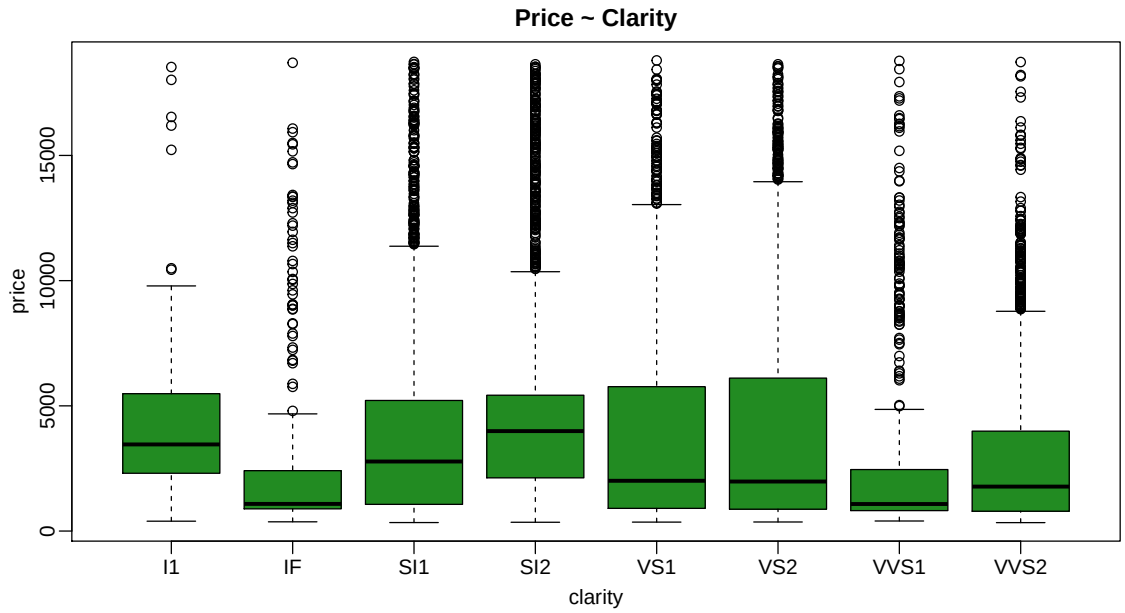
There are many variables strongly correlated.

Qualitative variables in relation with the price variable

```
par(mfrow=c(1,2),mar=c(3,3,3,.5), mgp=c(1.7,.5,0))
boxplot(price~cut, data=diamonds, col="forestgreen", main="Price ~ Cut")
boxplot(price~color, data=diamonds, col="forestgreen", main="Price ~ Color")
```




```
par(mfrow=c(1,1),mar=c(3,4,2,4))
boxplot(price~clarity, data=diamonds, col="forestgreen", main="Price ~ Clarity")
```



Best subset selection of the variables

Now we consider a model whose response is the variable `price` and regressors are all the other variables in the dataset. We try also to add an interaction between the weight of the diamond (`carat`) and the length of it (`x`).

Using the `regsubsets()` function we perform a Best Subset Selection, whose number of models considered would be $2^p = 2^{24} = 16777216$.

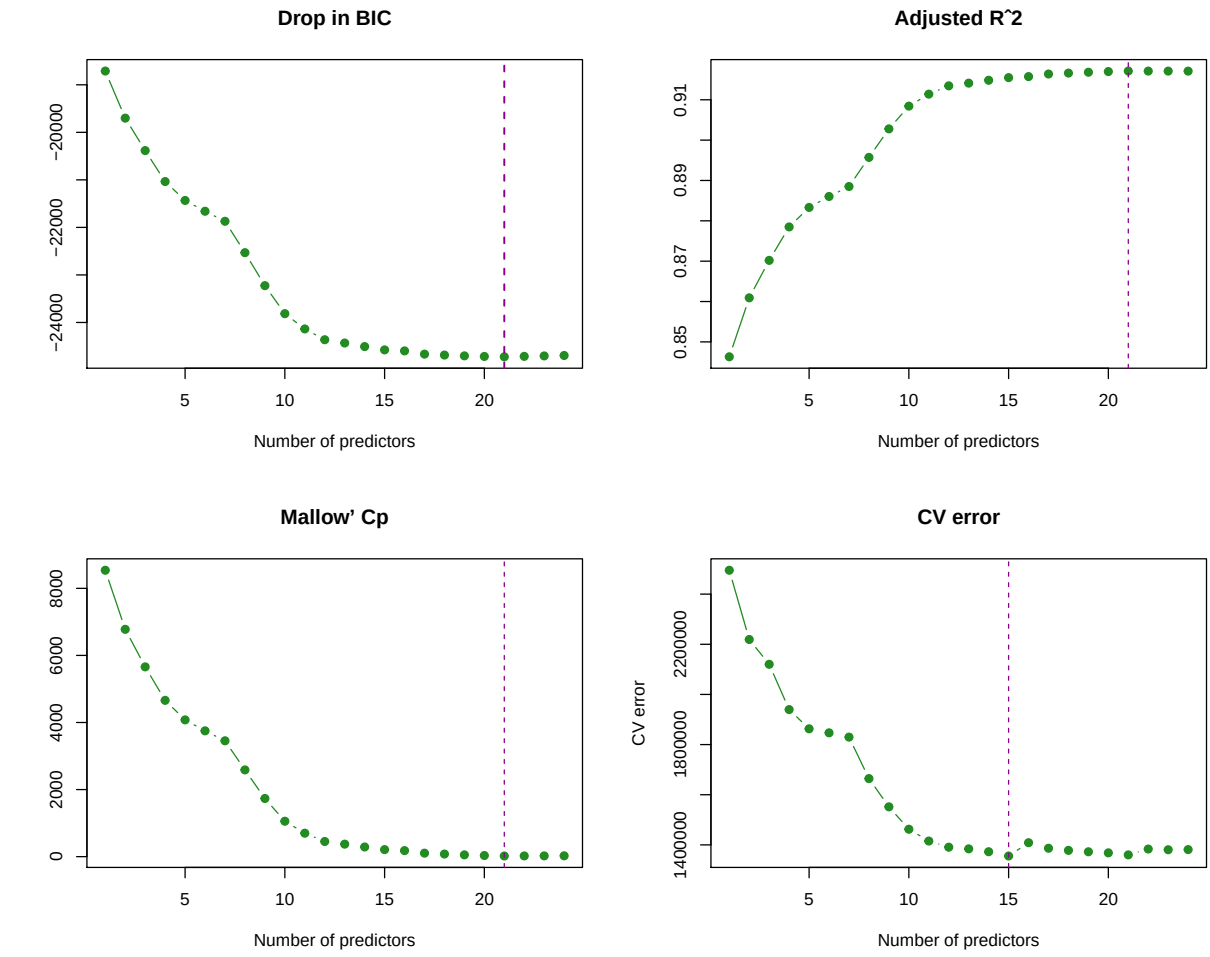
```
library(leaps)
ols_bss <- regsubsets(price ~ .+(carat*x),data=diamonds, nvmax=24)
summ_bss <- summary(ols_bss)
```

Now that the function provides us the best 24 models, each one containing respectively $1, 2, 3, ..., 24$ predictors, we have to choose the best among those, using different criteria.

- BIC : Bayesian information criteria
- Mallow's Cp
- Adjusted R^2
- Cross-Validation error

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

plot(summ_bss$bic, type="b", pch=19,
xlab="Number of predictors", ylab="", main="Drop in BIC", col="forestgreen")
abline (v=which.min(summ_bss$bic),col = "magenta4", lty=2,lwd=1.5)

plot(summ_bss$adjr2, type="b", pch=19,
xlab="Number of predictors", ylab="", main="Adjusted R^2",col="forestgreen")
abline (v=which.max(summ_bss$adjr2),col = "magenta4", lty=2)

plot(summ_bss$cp, type="b", pch=19,
xlab="Number of predictors", ylab="", main="Mallow' Cp",col="forestgreen")
abline (v=which.min(summ_bss$cp),col = "magenta4", lty=2)
```

```r
p <- 24
k <- 10
set.seed (1)
folds <- sample (1:k,nrow(diamonds),replace =TRUE)
cv.errors <- matrix (NA ,k, p, dimnames =list(NULL , paste (1:p) ))
for(j in 1:k){
  best.fit =regsubsets (price ~ .+(carat*x), data=diamonds[folds!=j,], nvmax=24)
  for(i in 1:p) {
    mat <- model.matrix(as.formula(best.fit$call[[2]]),diamonds[folds==j,])
    coefi <- coef(best.fit ,id = i)
    xvars <- names(coefi )
    pred <- mat[,xvars ]%*% coefi
    cv.errors[j,i] <- mean( (diamonds$price[folds==j] - pred)^2)
  }
}
cv.mean <- colMeans(cv.errors)

plot(cv.mean ,type="b",pch=19, xlab="Number of predictors",
     ylab="CV error", col="forestgreen", main="CV error")
abline(v=which.min(cv.mean), col="magenta4", lty=2)
```



The number of predictors for the best model chosen, according to the following criteria.

```
param <- c(which.min(summ_bss$bic),which.max(summ_bss$adjr2),
           which.min(summ_bss$cp),which.min(cv.mean))
names(param) <- c("BIC","Adj R^2", "Cp","CV-error")
param
```

```
##      BIC  Adj R\2102       Cp CV-error
##       21       21       21       15
```

We can see that the first 3 quantities give the same result, while the CV error is at his minimum for a very low number of parameters.

Looking at the summary of the `regsubset()` function I decided to take into consideration the model with 20 parameters, and the interaction is not included.

```
coef(ols_bss,20)
```

```
##  (Intercept)        carat      cutGood     cutIdeal   cutPremium cutVery Good
##  -1710.69642  10631.26787    726.18459   1034.46655    947.52449    881.44034
##       colorE       colorF       colorG       colorH       colorI       colorJ
##   -197.08386   -299.77121   -463.49354   -908.55939  -1474.73298  -2491.74012
##    clarityIF   claritySI1   claritySI2   clarityVS1   clarityVS2  clarityVVS1
##   5312.65183   3693.01834   2791.88833   4606.50535   4332.37240   5226.95343
##  clarityVVS2        depth            x
##   5028.73796    -45.20783   -769.04764
```

Collinearity issues

Now that we have our chosen model, we have to verify the presence of some problems, for example the collinearity.

```
ols <- lm(price ~ .-table-y-z, data=diamonds)

library(car)
vif(ols)[,1]
```

```
##     carat       cut     color   clarity     depth         x
## 20.229611  1.224178  1.191943  1.321608  1.193491 20.128518
```

We have collinearity for variable `carat` and `x`, I expected it since from the correlation matrix we have seen a high correlation between the two.

Since we have another variable of measure of the diamond which is the depth, we try to exclude the variable `x` from the model.

```
ols <- update(ols, ~ .-x, data=diamonds)
summ <- summary(ols)

vif(ols)[,1]
```

```
##    carat      cut    color  clarity    depth
## 1.310382 1.223710 1.181779 1.282690 1.126565
```

In fact, the collinearity is no longer an issue of our model.

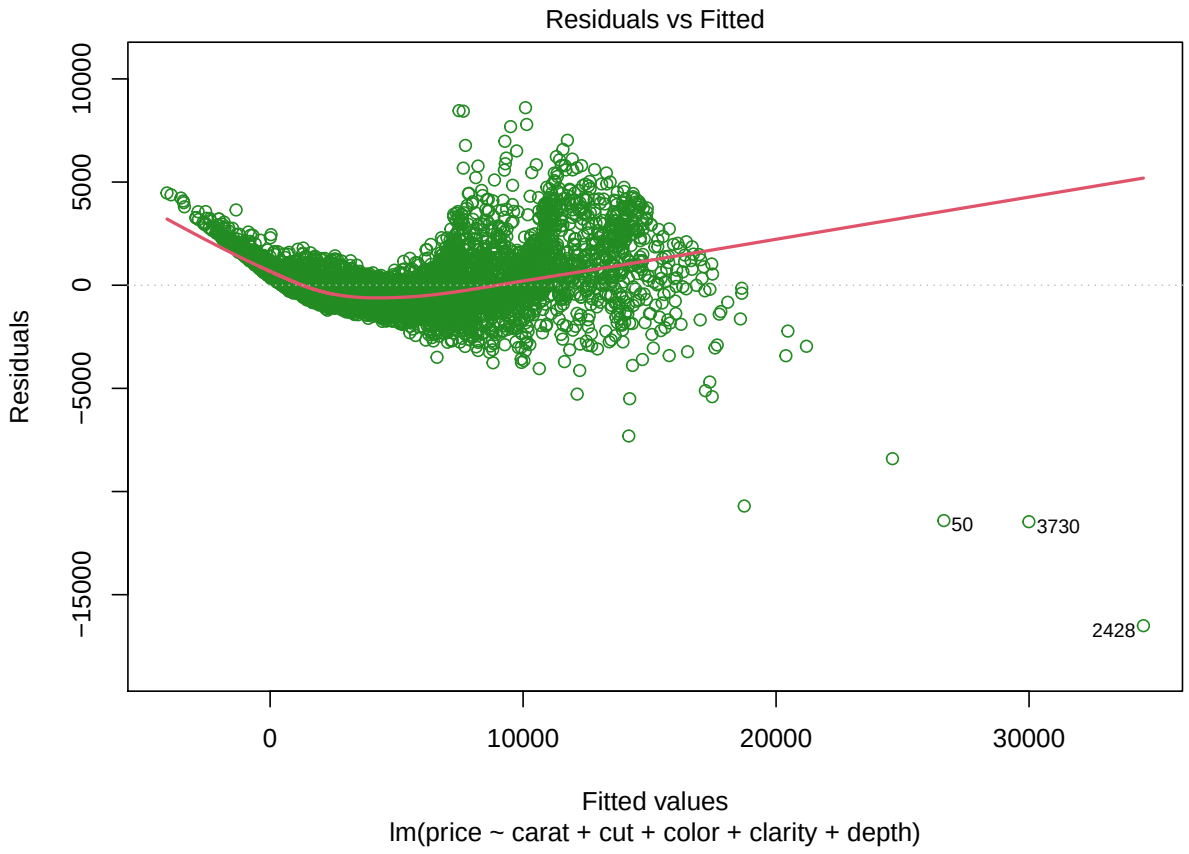
```
n <- nrow(diamonds)
p <- nrow(summ$coefficients)-1
```

Diagnostic

It's interesting to observe some different plots, in order to individuate if the model doesn't satisfy the assumptions made for the Multiple Linear Regression model.

First of all we plot the fitted values $\hat{y}$ versus the residuals.

```
plot(ols, which=1, lwd=2, col="forestgreen")
```



Variance of the errors

There is a strong evidence of non-constant variance of the errors, so there is heteroschedasticity.

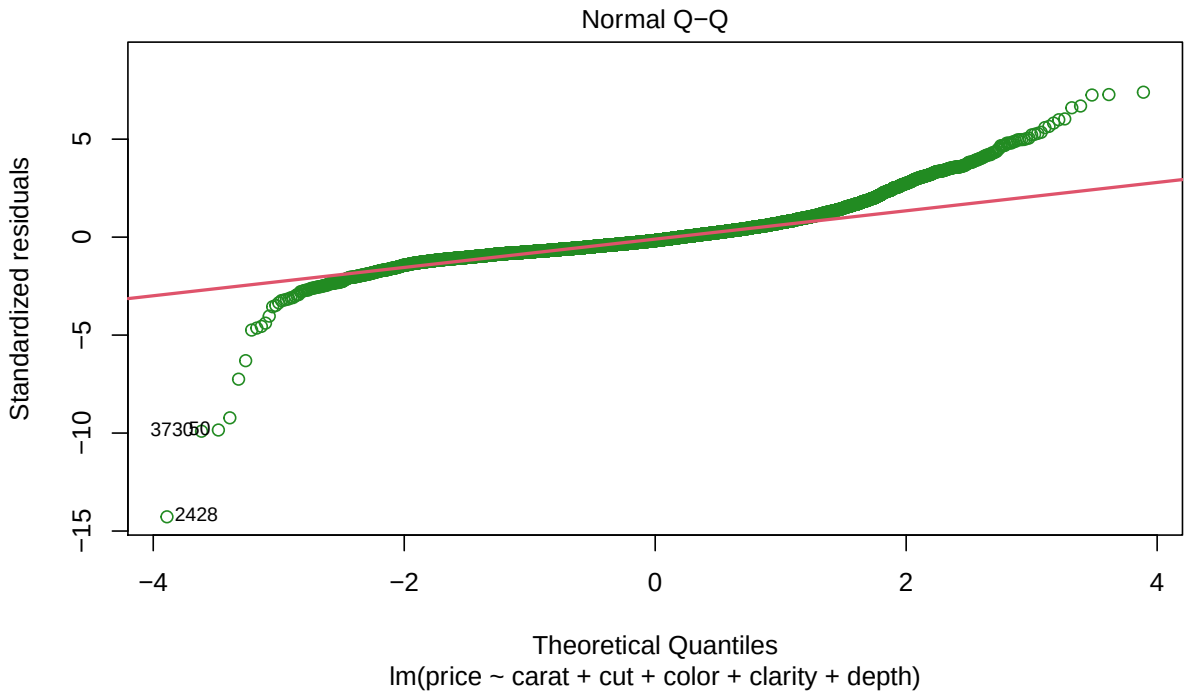
Linearity of the model

The fitted model line seems to have a slight paraboid shape, so we could say that the relationship between the response and the predictors is not very linear.

Normality assumption of the errors

To verify this assumption we have to provide the QQ-PLOT with the respected QQ-line and the result of the Shapiro-Wilk test.

```
plot(ols, which=2, col="forestgreen")
qqline(rstandard(ols), col=2,lwd=2)
```



Since the function of the Shapiro-Wilk test works if the number of observations is less or equal to 5000, I sample a smaller dataset.

```
newdiamonds <- diamonds[sample(nrow(diamonds),5000),]
ols_new <- lm(price ~ carat+cut+color+clarity+depth, data=newdiamonds)
shapiro.test(residuals(ols_new))
```

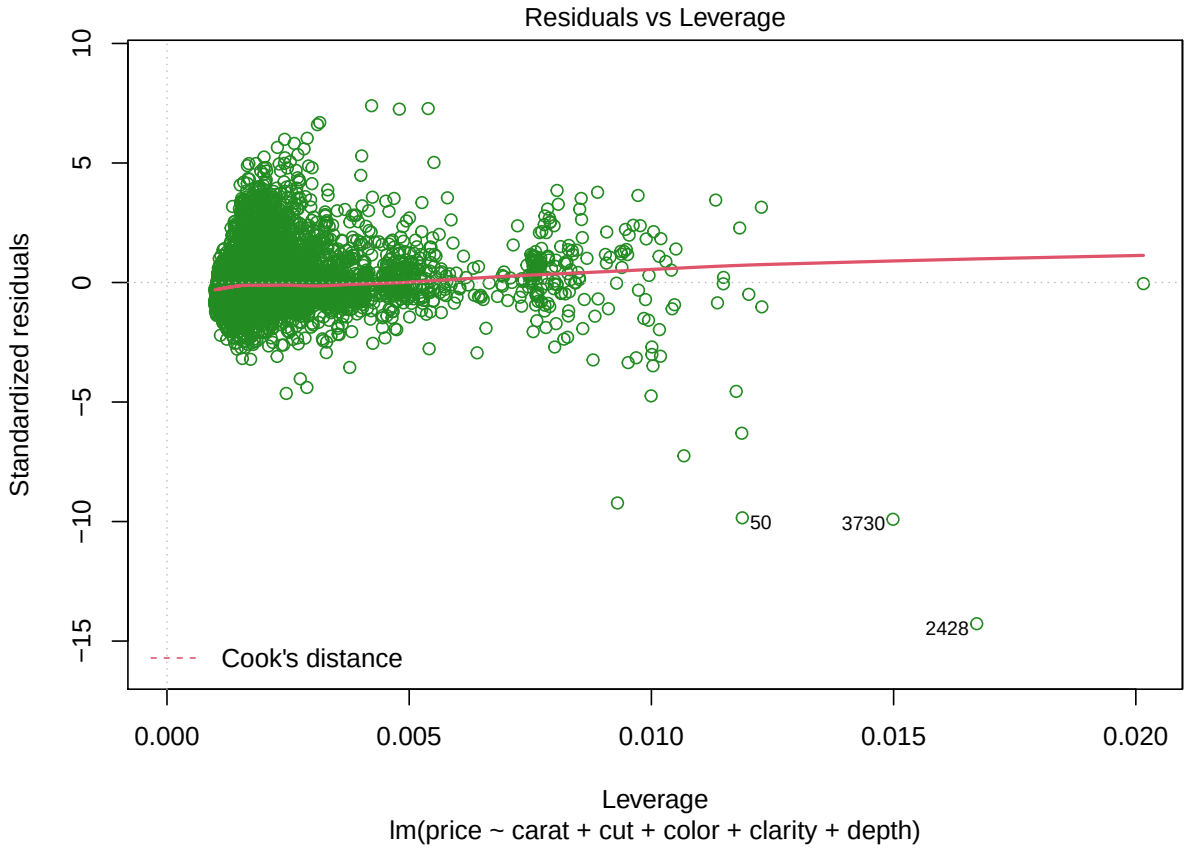
```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(ols_new)
## W = 0.87617, p-value < 2.2e-16
```

From the qq-plot and the test we can say that the hypothesis of normality of the model has to be rejected. Observing the plot we could say that we have a long tailed distribution, in this case we should change the inference on another distribution.

Large leverage points

We want now to individuate points with high leverage, so we observe the plot of the leverages in relation with the standardized residuals.

```
plot(ols, which=5, lwd=2, col="forestgreen")
```




```
hii <- influence(ols)$hat
n <- nrow(diamonds)
p <- nrow(summ$coefficients)-1
high_leverage=hii[which(hii>=(2*(p+1)/n))]

length(high_leverage)
```

```
## [1] 528
```

It seems that the data contain 528 observations with high leverage, from the plot we can individuate in particular 3 points with very high leverage.

In particular, these observations we are discussing are the observations with row number :

```
names(sort(high_leverage, decreasing = T)[1:3])
```

```
## [1] "9201" "2428" "3730"
```

Outliers

From the first plot, the residual plot, we can observe that there are 5 observations very far from the others, 3 of them are observation number 50, 3730 and 2428.

According to the rule of thumb, which says that an observation is a possible outlier if the standardize residual is $> |3|$, we have that possible outliers in our data are :

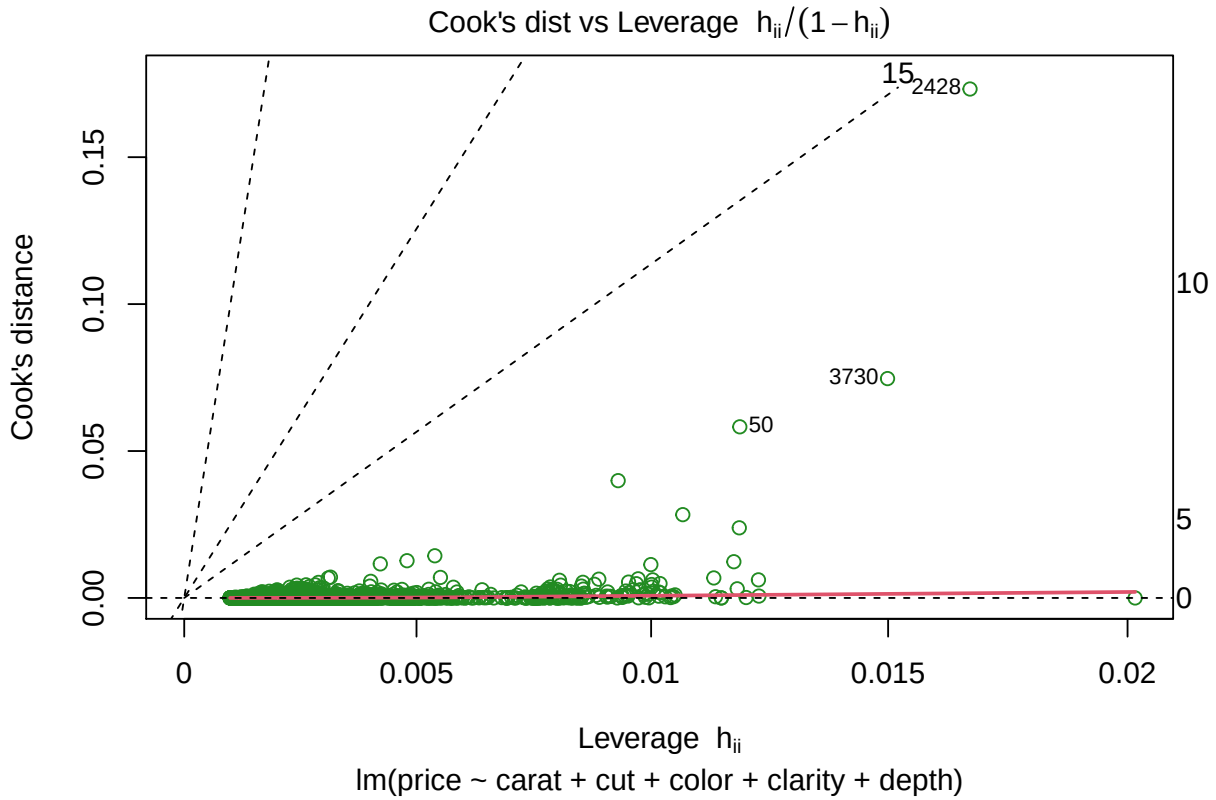
```
rsta <- rstandard(ols)
length(rsta[which(rsta>abs(3))])
```

```
## [1] 183
```

Influential points

In order to combine the information above on outliers and high leverage points we use a measure called Cook's distance that allow us to see which are the points that most influence the model.

```
plot(ols, which=6, lwd=2, col="forestgreen")
```



From the plot we see that there are at least 3 points that are far from the others, we calculate them.

```
cook <- cooks.distance(ols)
infl_points <- sort(cook,decreasing=T)[1:3]
infl_points
```

```
##       2428       3730         50
## 0.17319043 0.07465118 0.05823615
```

Improve the model

- For the non-constant variance we should transform the response, in particular a logarithmic transformation could be useful.
- For the non linearity of the model, we should verify which of the predictor has the most non linear relationships with the residuals, and in our case is the variable `carat`, and a logarithmic transformation also of this variable could improve the model.
- For the non normality of the errors, we ignore it for now.
- Since we have a very big dataset and only a few influential points, which don't exceed the threshold, I think it's not necessary to remove them from the fitting.

The new model will be the following one :

```
ols <- lm(log(price) ~ log(carat)+ cut + color + clarity + depth, data=diamonds)
summ <- summary(ols)
round(summ$coefficients,4)
```

```
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept)    7.9224     0.0656 120.6925   0.0000
## log(carat)     1.8834     0.0026 716.2986   0.0000
## cutGood        0.0860     0.0094   9.1702   0.0000
## cutIdeal       0.1649     0.0087  18.9429   0.0000
## cutPremium     0.1458     0.0089  16.3872   0.0000
## cutVery Good   0.1246     0.0088  14.0950   0.0000
## colorE        -0.0497     0.0050  -9.9836   0.0000
## colorF        -0.0935     0.0050 -18.6502   0.0000
## colorG        -0.1538     0.0049 -31.2836   0.0000
## colorH        -0.2460     0.0052 -47.3203   0.0000
## colorI        -0.3677     0.0059 -62.8045   0.0000
## colorJ        -0.5122     0.0073 -70.1161   0.0000
## clarityIF      1.0857     0.0137  79.3474   0.0000
## claritySI1     0.5735     0.0115  49.8325   0.0000
## claritySI2     0.4154     0.0116  35.8617   0.0000
## clarityVS1     0.7922     0.0118  67.2293   0.0000
## clarityVS2     0.7299     0.0116  63.0060   0.0000
## clarityVVS1    1.0100     0.0125  80.5937   0.0000
## clarityVVS2    0.9313     0.0121  76.6644   0.0000
## depth         -0.0009     0.0010  -0.9334   0.3506
```

Interpretation parameters and uncertainties

The fitted model has the following structure :

$$\hat{Y} = \hat{\beta}_0 + \hat{\beta}_1 X_1 + \hat{\beta}_2 X_2 + \hat{\beta}_3 X_3 + ... + \hat{\beta}_{19} X_{19}$$

Where:

- $\hat{\beta}_0$ is the intercept and is estimated to be 7.922. It corresponds to the expected logarithm of the price of the diamond in a state with no carats, color D, fair cut, clarity I1, no depth and no table.
- $\hat{\beta}_1$ is the coefficient for the logarithm of the number of carats (`log(carat)`) whose estimation is 1.883 which means that if the logarithm of the carats increase by 1 the log price rises by 1.883, keeping the other variables constant.
- $\hat{\beta}_2, \hat{\beta}_3, \hat{\beta}_4, \hat{\beta}_5$ are coefficients for the variable dummy `cut` which assumes 1 if it is the cut that corresponds to the diamond, 0 otherwise. If each of them are 0 the cut of the diamond is the "Fair" one, which is the baseline. The coefficients of those variables are all positive, it means that if the cut is not Fair (which is

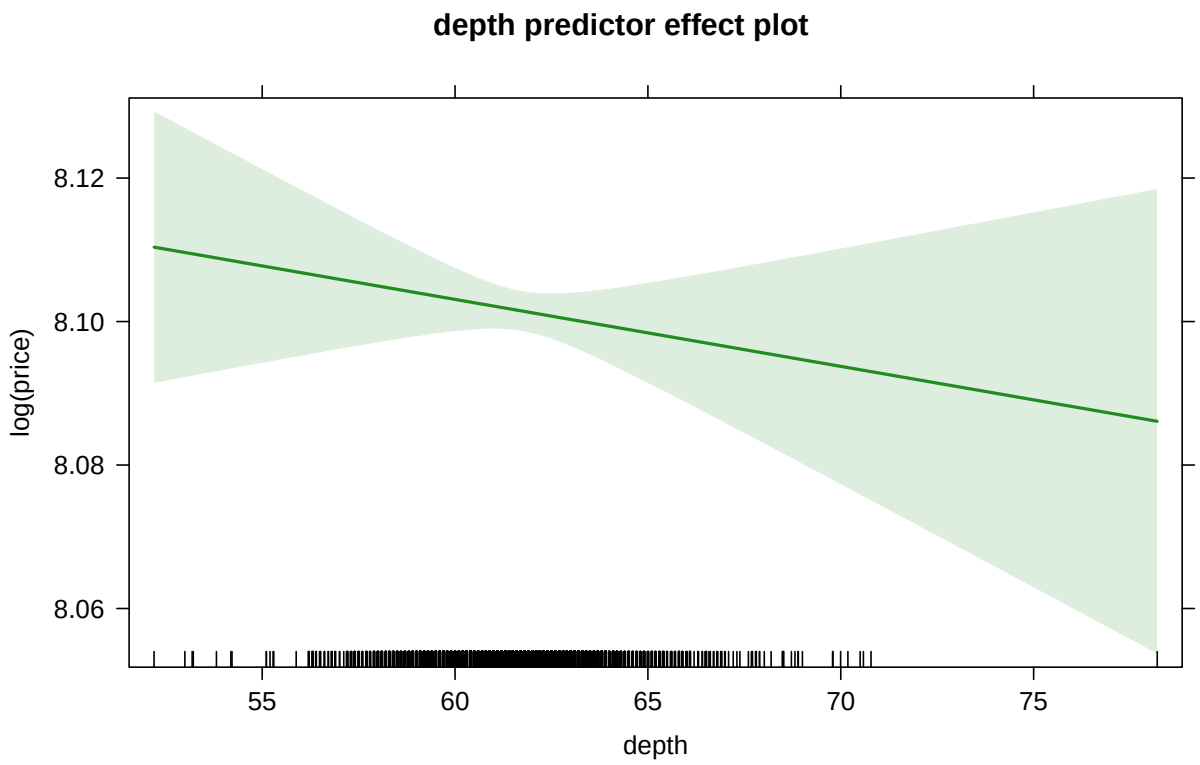
the worst cut) the logarithm of the price,keeping the other variables constant, increases respectively by 0.086, 0.165, 0.146 and 0.124 for cut Good, Ideal, Premium and Very Good.

- $\hat{\beta}_6, \hat{\beta}_7, \hat{\beta}_8, \hat{\beta}_9, \hat{\beta}_{10}, \hat{\beta}_{11}$ are coefficients for the dummy variable `color` which assumes 1 if it is the color that corresponds to the one of the diamond (E,F,G,H,I,J), 0 otherwise. If each of them are 0 the color is the color D, which is the baseline. The coefficients of those variables are all negative, it means that if the color is not color D (which is the best) the logarithm of the price, keeping the other variables constant, decreases respectively by 0.049, 0.093, 0.154, 0.246, 0.367 and 0.512 for color E,F,G,H,I and J.
- $\hat{\beta}_{12}, \hat{\beta}_{13}, \hat{\beta}_{14}, \hat{\beta}_{15}, \hat{\beta}_{16}, \hat{\beta}_{17}, \hat{\beta}_{18}$ are variables for the dummy `clarity` which assumes 1 if it is the clarity that corresponds to the one of the diamond (IF,SI1,SI2,VS1,VS2,VVS1,VVS2), 0 otherwise. If each of them are 0 the clarity is 'I1', which is the baseline. The coefficients of those variables are all positive, it means that if the clarity is not I1 (which is the worst) the log price, keeping the other variables constant, increases respectively by 1.086, 0.573, 0.415, 0.792, 0.729, 1.01 and 0.931 for clarity IF,SI1,SI2,VS1,VS2,VVS1 and VVS2.
- $\hat{\beta}_{19}$ is the coefficient for the variable `depth` , with all the other regressors in the model held fixed, increasing depth of a diamond by 1, decreases the logarithm of the price in average by 0.001, keeping the other variables constant.

The second column provide the estimated standard deviation for each coefficient and it's a measure that tells us the range in which probably would stay the coefficient.

So for example for the `depth` variable the value of $\hat{\beta}_{19}$ mostly stands in the range -0.0009 ± 0.0010 which is a very very high standard error. The same interpretation for each of the other coefficients, but for the others the standard error in proportion of the coefficient is a lower value.

```
library(effects)
plot(predictorEffect("depth",ols,main=""),lines=list(col="forestgreen"))
```



Estimated standard error

```
summ$sigma
```

```
## [1] 0.1348546
```

It measures the error of the regression line's predictability, $\hat{\sigma} = 0.135$. It means that the errors for the logarithm of the price are mostly in the range ± 0.135

Coefficient of determination R^2

```
summ$r.squared
```

```
## [1] 0.982447
```

So $R^2 = 0.98$ and it's a very high value, it means that the regressors explain the 98% of the variation of the log(price).

Significance test on coefficients

In the summary of the model above, for each coefficient is indicated the T-test and the pvalue associated. The T-test is the statistical test for the test of the following hypothesis :

$$H_0 : \beta_j = 0 \longrightarrow T_{test} = \hat{\beta}_j / se(\hat{\beta}_j)$$

For all the coefficients except for the one associated to the `depth` variable, we can conclude that the p-value is very close to zero, so the null hypothesis is rejected. While $\hat{\beta}_{19}$ is non significant because the associated p-value is $> 5\%$.

In fact, we can provide the 95% confidence interval for each coefficient :

```
confint.lm(ols)
```

```
##                     2.5 %       97.5 %
## (Intercept)   7.793742789  8.051083278
## log(carat)    1.878213416  1.888521351
## cutGood       0.067598439  0.104354991
## cutIdeal      0.147856345  0.181988438
## cutPremium    0.128387214  0.163275222
## cutVery Good  0.107286204  0.141947222
## colorE       -0.059426426 -0.039920514
## colorF       -0.103276759 -0.083632028
## colorG       -0.163435624 -0.144161877
## colorH       -0.256141950 -0.235765150
## colorI       -0.379162474 -0.356210626
## colorJ       -0.526549609 -0.497909290
## clarityIF     1.058869639  1.112511511
## claritySI1    0.550893821  0.596008092
## claritySI2    0.392703648  0.438116243
## clarityVS1    0.769124215  0.815321805
## clarityVS2    0.707189023  0.752605167
## clarityVVS1   0.985426082  1.034556134
## clarityVVS2   0.907485988  0.955109936
## depth        -0.002893555  0.001026753
```

Only the variable `depth` contain the value 0 in the interval.

Test of a group of regressors

Since in the best subset selection the Cross Validation error was at his minimum for the model with only 15 variables, and this model wouldn't have included the `cut` variable, it's interesting to test the regressors for the dummy variable `cut`.

$$H_0 : \beta_2 = \beta_3 = \beta_4 = \beta_5 = 0$$

```
ols1 <- lm(log(price) ~ log(carat)+color+clarity+depth,data=diamonds)
anova(ols1,ols)
```

```
## Analysis of Variance Table
##
## Model 1: log(price) ~ log(carat) + color + clarity + depth
## Model 2: log(price) ~ log(carat) + cut + color + clarity + depth
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1   9984 191.57
## 2   9980 181.49  4    10.074 138.49 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The pvalue of the F test for those regressors is very low, so we have to reject the null hypothesis and we should remain with the original model.

Test all the regressors

$$H_0 : \beta_1 = \beta_2 = ... = \beta_{18} = \beta_{19} = 0$$

```
summ$fstatistic
```

```
##    value    numdf    dendf
## 29399.12    19.00  9980.00
```

```
1-pf(summ$fstatistic[1],summ$fstatistic[2],summ$fstatistic[3])
```

```
## value
##     0
```

Pvalue is very low, tend to 0, so we have to reject the null hypothesis, and the model is significant.

Prediction

We suppose to have a new information about a 1.5 carat diamond, good cut, with a H color and clarity type IF, whose depth is 60.

```
newdata = data.frame(carat=1.5, cut="Good",color="H",clarity="IF", depth=60)
predict(ols, newdata = newdata, interval="prediction",level=.95)
```

```
##        fit      lwr      upr
## 1 9.555762 9.290709 9.820816
```

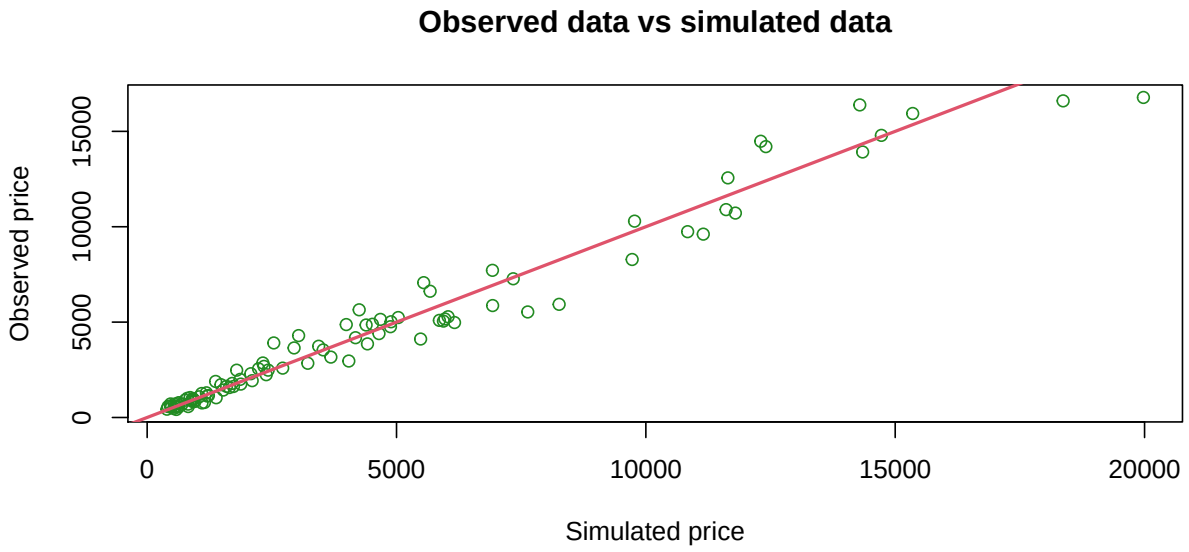
The prediction of the logarithm of the price is 9.55, and with a 95% of probability the real value would stay between 9.29 and 9.82.

Simulation of a new dataset

Given the estimated parameters fitted in the model above, we sample 100 data from the design matrix and we calculate the fitted values.

```
set.seed(1)
n <- 100
beta <- ols$coefficients ; sigma <- summary(ols)$sigma
X <- model.matrix(ols)
obs_simu <- sample(1:nrow(X),n)
new_X <- X[obs_simu,]

y_hat=new_X%*%beta+(rep(1,n)*rnorm(n, mean=0, sd=sigma))
fake_data <- data.frame(new_X[,-1],exp(y_hat))
fake_data[,1] <- exp(fake_data[,1])
colnames(fake_data)[c(1,20)] <- c("carat","price")
```

```
plot(fake_data$price, diamonds$price[obs_simu], xlab="Simulated price", ylab="Observed price",
     main="Observed data vs simulated data", col="forestgreen")
abline(0,1, col=2, lwd=2)
```



We used the fitting of the improved model to create a new dataset containing the new prices calculated on the base of the estimated parameters and taking into account also the normal distributed error for each observation. From the plot we should say that simulated and observed data are very similar, more for low prices than high prices.

Conclusions

What we can conclude from this project is that the Multiple Linear Model is not a perfect model for our data, but a good one, we had many issues and many assumptions were incorrect, but with some transformations and the good variables selected we obtain such a quite good model.

In particular variables containing information on the number of carats, the cut, the color, the clarity and the depth are very useful to predict the price of a diamond, those variables explain a very high percentage of the variability of the price for the multiple linear model.