Regression analysis in R

Marius G. Sampid

31 October 2018

Introduction

Regression is the relationship between one or more independent variables and the expected value of the corresponding dependent variable. Regression analysis is the most popular type of data analysis method used to test hypothesis about how the world works and for forecasting. For example, how change in the money value will affect interest rates, what will be the unemployment rate in the next year, etc.

In this report, I look at the relationship between price and weight of diamond (carat), price and length (x), width (y), depth (z) and total depth of diamond, using the publicly available diamond data set, which can be found in https://vincentarelbundock.github.io/Rdatasets/datasets.html.

Reading data into R

```
# Get the address/path where data is stored:
address <-
"C:/Users/marius/Desktop/deskTopFiles/DataAnalysis_R_Python/SampleData.csv"


Data <- read.csv(address) # Load data into R and store in an object called
'Data'

head(Data) # See the first 6 rows of the data set. A quick snapshot of the
data to make sure that the data was correctly imported into R

##   carat       cut color clarity depth table price    x    y    z
## 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  0.29   Premium     I     VS2  62.4    58   334 4.20 4.23 2.63
## 5  0.31      Good     J     SI2  63.3    58   335 4.34 4.35 2.75
## 6  0.24 Very Good     J    VVS2  62.8    57   336 3.94 3.96 2.48


myData <- Data[,c(1,5:10)] # Isolate the numeric variables in the data set.
```

Let's look at the Pearson's correlation coefficient, (ρ_r), between the numeric variables:

```
cor(myData[,c(1,5:10)])
```

```
##             carat       depth      table      price           x           y
## carat 1.00000000  0.02822431  0.1816175  0.9215913  0.97509423  0.95172220
## depth 0.02822431  1.00000000 -0.2957785 -0.0106474 -0.02528925 -0.02934067
## table 0.18161755 -0.29577852  1.0000000  0.1271339  0.19534428  0.18376015
## price 0.92159130 -0.01064740  0.1271339  1.0000000  0.88443516  0.86542090
## x     0.97509423 -0.02528925  0.1953443  0.8844352  1.00000000  0.97470148
## y     0.95172220 -0.02934067  0.1837601  0.8654209  0.97470148  1.00000000
## z     0.95338738  0.09492388  0.1509287  0.8612494  0.97077180  0.95200572
##                z
## carat 0.95338738
## depth 0.09492388
## table 0.15092869
## price 0.86124944
## x     0.97077180
## y     0.95200572
## z     1.00000000
```

Before we proceed, let's look at some strict assumptions associated with linear models for regression analysis: (1.) All variables are continuous numeric variables and not categorical ones, (2.) the data is free of missing values and outliers, (3.) there's a linear relationship between the dependent and independent variables, (4.) all explanatory or independent variables are independent of one another, and finally, (5.) the residuals are normally distributed.

We can see from the Pearson's correlation matrix that there is a strong linear relationship between price and weight of diamond (carat), price and length of diamond (x), price and width of diamond (y), price and depth of diamond (z) with ρ_rs of 0.92, 0.88, 0.87, and 0.86, respectively. The correlation between price and the width of top diamond relative to widest point (table) and total depth and price are quite small; i.e: 0.127 and -0.011, respectively. Thus, as price of diamond increase the total depth decreases and vice versa because ρ_r is negative.

For the regression analysis, we need a small sample of our data set. The data contains over 5000 observations, so it will make no sense to try to fit a linear regression model to all data points as it will be too compact, and figures will be difficult to analyze.

Let's collect a random sample of 50 observations from our data set for regression analysis:

```
set.seed(10)
MySampleData <- myData[sample(1:nrow(myData), 50, replace=FALSE),] # Collect
a random sample of 50 observations without replacement.
```

```
head(MySampleData) # see first few lines of sample data
```

```
##        carat       cut color clarity depth table price    x    y    z
## 27374  0.32     Ideal     G     SI1  61.5    55   647 4.40 4.44 2.72
## 16547  1.07 Very Good     G     VS1  58.6    60  6610 6.65 6.76 3.93
## 23027  0.30 Very Good     H    VVS1  61.0    55   631 4.35 4.39 2.66
## 37384  0.40     Ideal     I      IF  62.2    56   982 4.71 4.75 2.94
## 4592   1.11   Premium     D      I1  61.9    58  3655 6.66 6.63 4.11
## 12159  1.30   Premium     J     SI1  61.5    58  5176 7.02 6.97 4.30
```

```
dim(MySampleData) # check the dimension of the sample data
```

```
## [1] 50 10
```

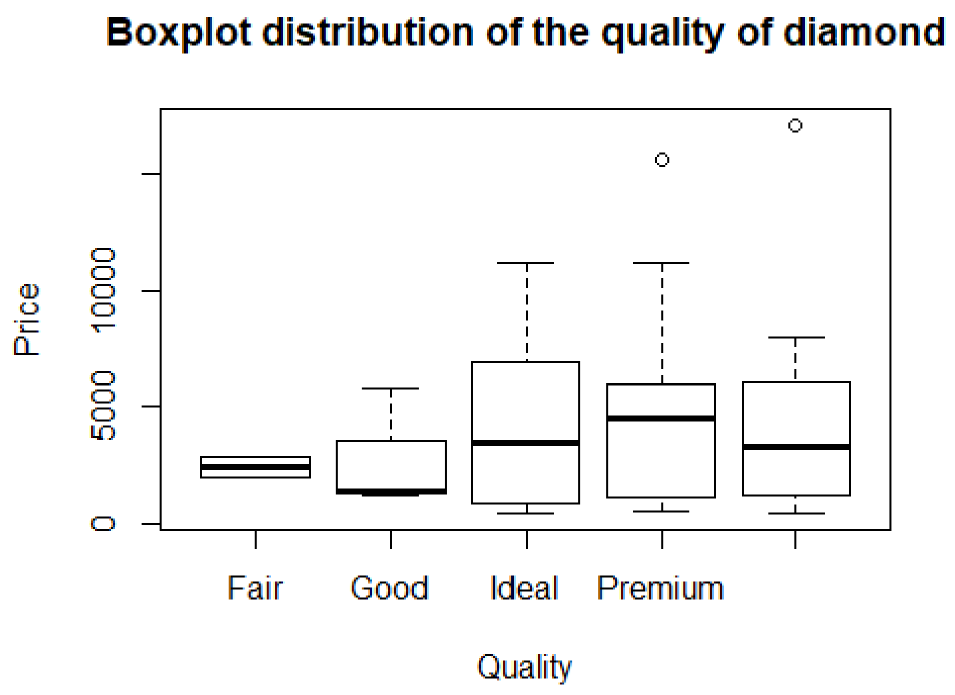
We can see the summary statistics of our sample data by calling the basicStats( ) function in the fBasics package:

```
library(fBasics) # Load package for the function basicStats()
basicStats(MySampleData[,c(1,5:10)]) # see summary statistics of the numeric
variables of our sample data
```

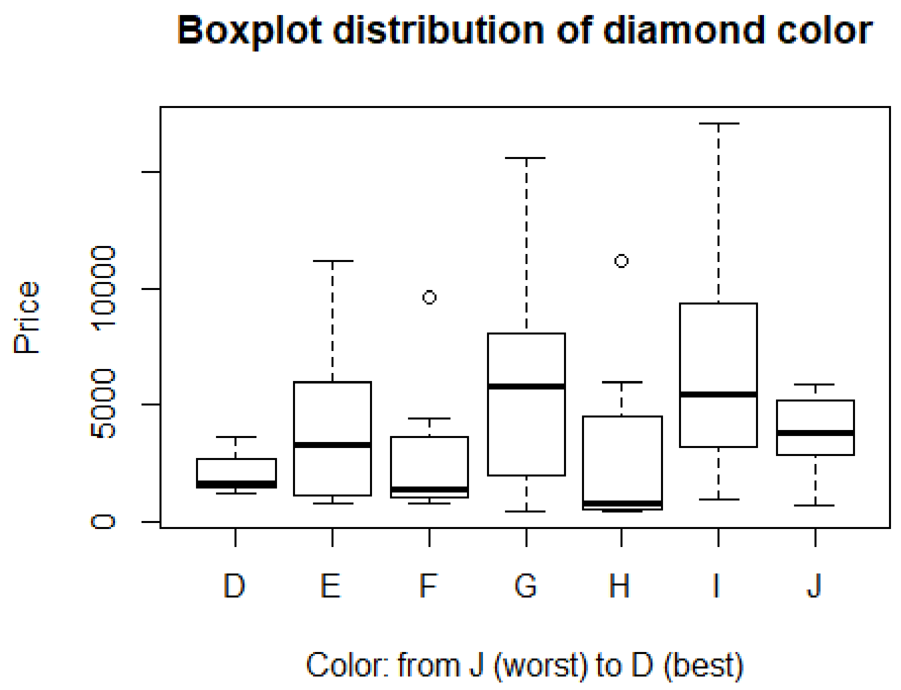
```
##                  carat       depth       table        price          x
## nobs         50.000000   50.000000   50.000000 5.000000e+01  50.000000
## NAs           0.000000    0.000000    0.000000 0.000000e+00   0.000000
## Minimum       0.300000   55.800000   54.000000 4.470000e+02   4.260000
## Maximum       2.310000   65.000000   70.000000 1.706200e+04   8.340000
## 1. Quartile   0.402500   60.700000   56.000000 1.041000e+03   4.745000
## 3. Quartile   1.200000   62.525000   58.000000 5.952750e+03   6.782500
## Mean          0.860000   61.472000   57.542000 4.349120e+03   5.902800
## Median        0.915000   61.550000   57.000000 3.458000e+03   6.125000
## Sum          43.000000 3073.600000 2877.100000 2.174560e+05 295.140000
## SE Mean       0.066167    0.211449    0.369623 5.673837e+02   0.159924
## LCL Mean      0.727032   61.047078   56.799215 3.208920e+03   5.581420
## UCL Mean      0.992968   61.896922   58.284785 5.489320e+03   6.224180
## Variance      0.218906    2.235527    6.831057 1.609621e+07   1.278792
## Stdev         0.467874    1.495168    2.613629 4.012009e+03   1.130837
## Skewness      0.612368   -0.942684    2.195652 1.245406e+00   0.011449
## Kurtosis     -0.006977    2.621585    8.077126 1.110025e+00  -1.297226
##                   y          z
## nobs         50.000000  50.000000
## NAs           0.000000   0.000000
## Minimum       4.300000   2.610000
## Maximum       8.430000   5.270000
## 1. Quartile   4.755000   2.950000
## 3. Quartile   6.775000   4.137500
## Mean          5.912400   3.630400
## Median        6.180000   3.880000
## Sum         295.620000 181.520000
## SE Mean       0.159480   0.098307
## LCL Mean      5.591913   3.432845
## UCL Mean      6.232887   3.827955
## Variance      1.271692   0.483212
## Stdev         1.127693   0.695135
## Skewness      0.018580   0.070567
## Kurtosis     -1.267273  -1.110161
```

We can visualize the distribution of the categorical variables in our sample data using boxplots:

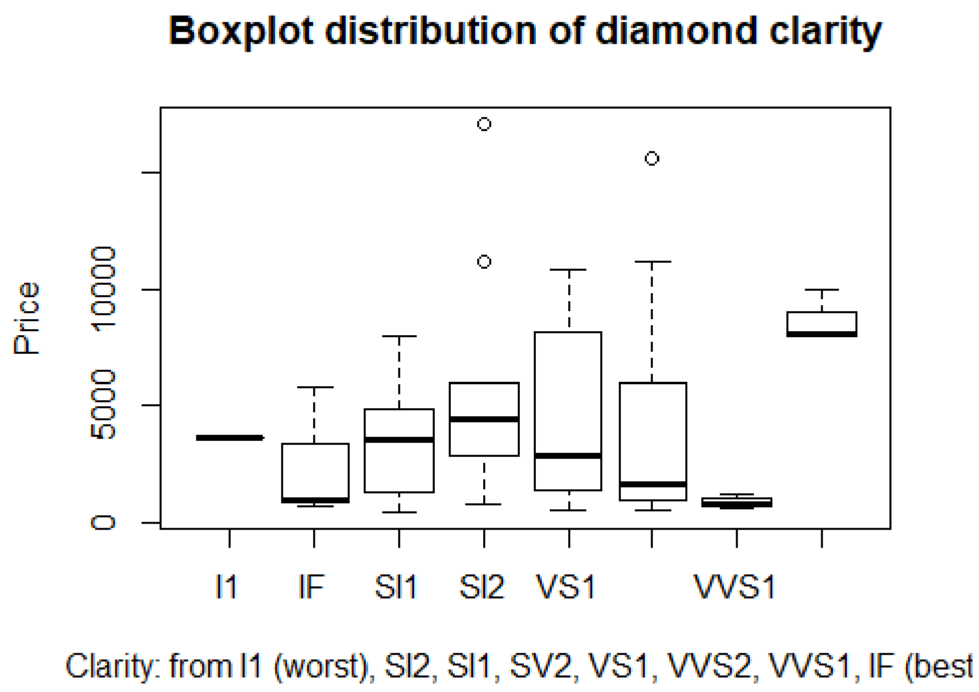
```
boxplot(MySampleData$price~MySampleData$cut, xlab ="Quality", ylab = "Price",
main = "Boxplot distribution of the quality of diamond") # distribution of
sample data based on quality of diamond.
```



```
boxplot(MySampleData$price~MySampleData$color, xlab ="Color: from J (worst)
to D (best)", ylab = "Price", main = "Boxplot distribution of diamond color")
# distribution of sample data based on color of diamond.
```



```
boxplot(MySampleData$price~MySampleData$clarity, xlab ="Clarity: from I1
(worst), SI2, SI1, SV2, VS1, VVS2, VVS1, IF (best)", ylab = "Price", main =
"Boxplot distribution of diamond clarity") # distribution of sample data
based on clarity of diamond.
```



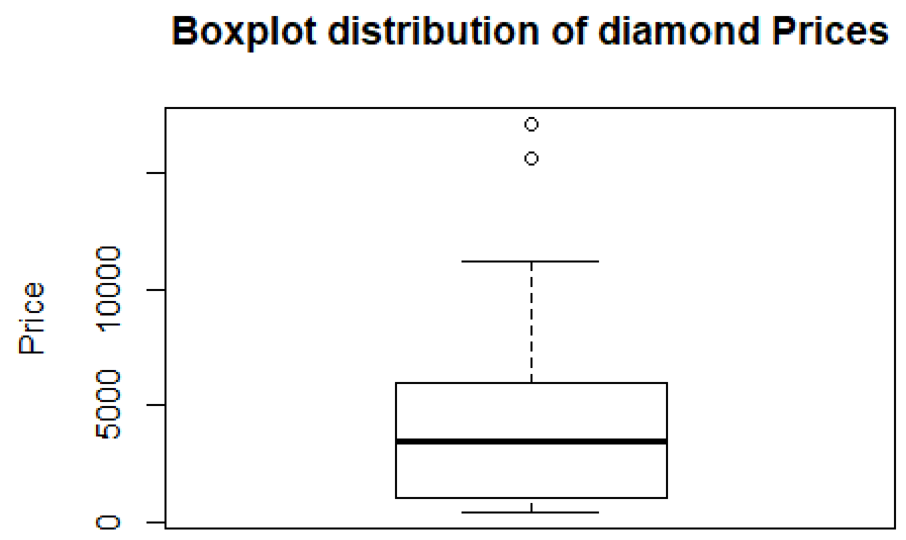
Let's model the relationship between price and weight of diamond (carat). That is, we want to see how a unit change in the weight of diamond will affect price.

```
cor(MySampleData$price, MySampleData$carat) # See the correlation coefficient
between price and carat (weight of diamond)

## [1] 0.9353939
```

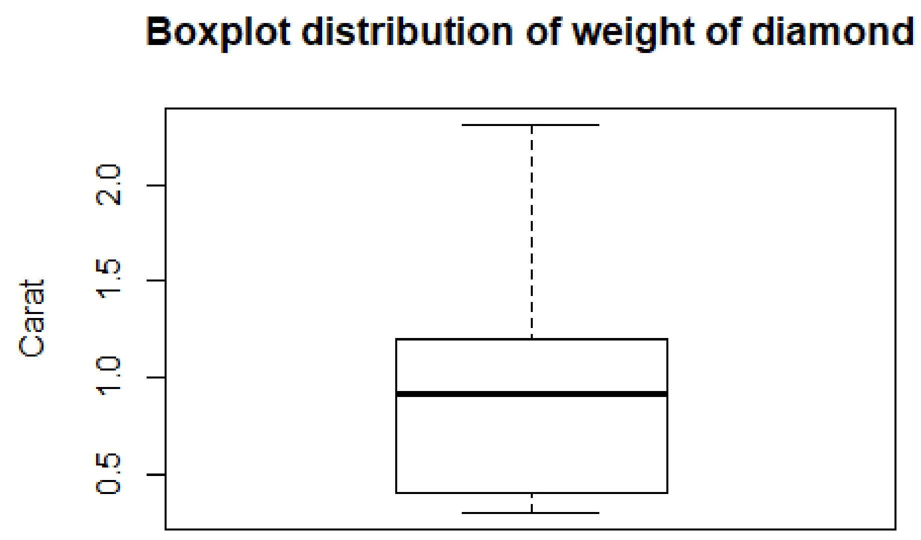
Taking the assumptions of linear models into consideration, we know that there's a strong linear relationship between the dependent (price) and independent (carat) variables in our sample data with a ρ_r of 93.54%, and that our data contains no missing values as seen in the summary statistics.

We can check for outliers in the price and carat variables in the sample data with the help of boxplots as follows:

```
boxplot(MySampleData$price, ylab = "Price", main = "Boxplot distribution of
diamond Prices")
```

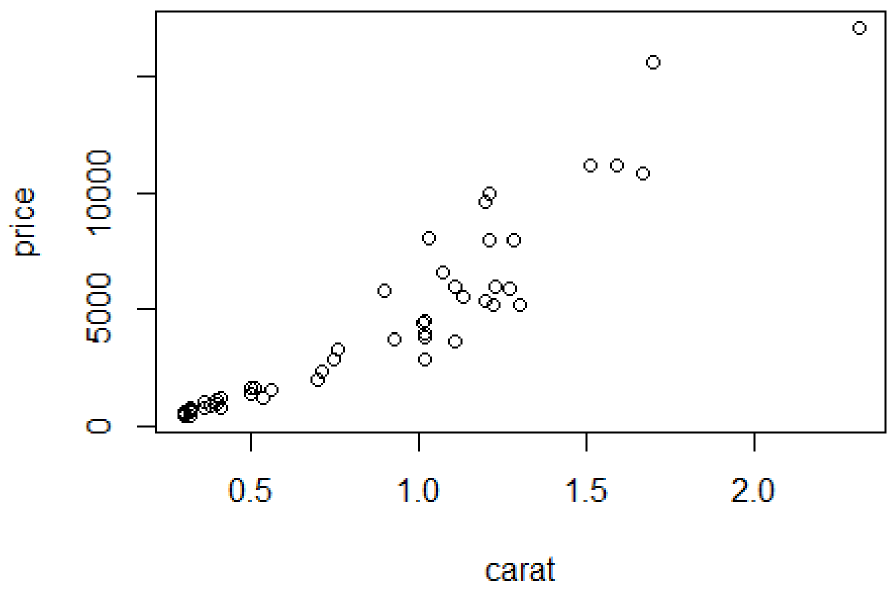


```r
boxplot(MySampleData$carat, ylab = "Carat", main = "Boxplot distribution of
weight of diamond")
```



Boxplots suggests that we have two data points in the price variable as outliers. These outliers are also visible in the scatter plot:

```r
plot(price ~ carat, data = MySampleData)
```



A “rule of thumb”: any point in a data set that is greater than $Q_3 + 1.5 * IQR$ or less than $Q_1 - 1.5 * IQR$ is considered an outlier, where Q_1 is the First Quartile, Q_3 is the Third Quartile, and IQR is the interquartile range.

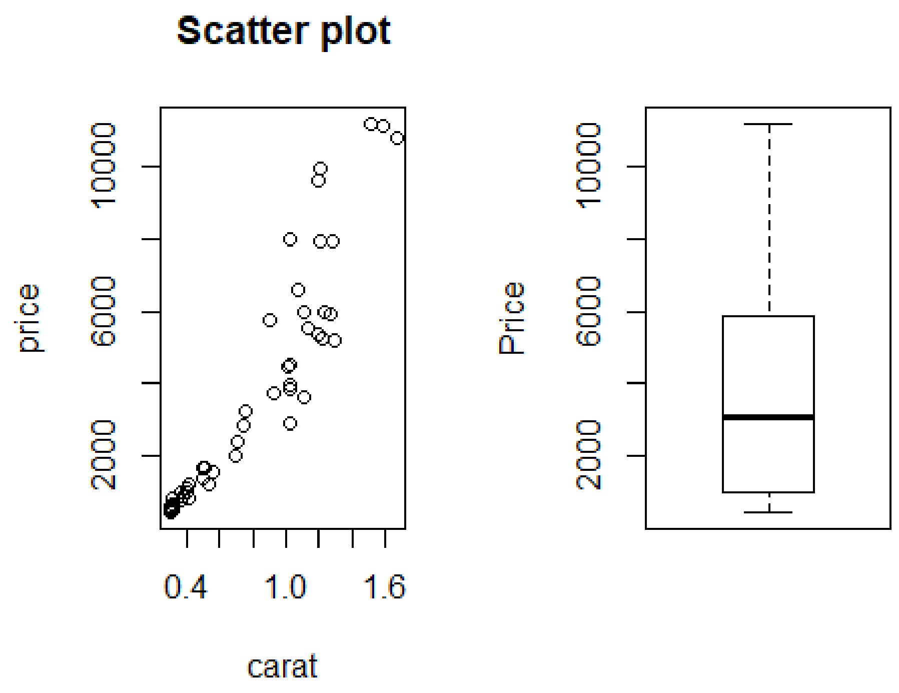
Let’s remove the outliers in the price variable and the corresponding rows in the carat variable. The outliers in the price variable are greater than $Q_3 + 1.5 * IQR$, so we remove these outliers as follows:

```r
#calculates the cutoff point for the upper outliers
cutoff_point <- quantile(MySampleData$price, 0.75) +
1.5*IQR(MySampleData$price)


cutoff_point

##      75%
## 13320.38
```

```
# delete corresponding rows in the sample data
index_cutoff_point <- which(MySampleData$price > cutoff_point)
MySampleData_2 <- MySampleData[-index_cutoff_point,]


cor(MySampleData_2$price, MySampleData_2$carat) # see new correlation
coefficient after deleting outliers

## [1] 0.9227177
```

Let's look again at the scatter plot of price and carat, and the boxplot of the price variable to be sure that the outliers have been removed

```
par(mfcol=c(1,2)) # set the screen to show a matrix of plots: 1 row two
columns
plot(price ~ carat, data = MySampleData_2, main="Scatter plot")
boxplot(MySampleData_2$price, ylab ="Price")
```



```
par(mfcol=c(1,1)) # set the screen back to its original format
```

We can see from the scatter plot and boxplot that there is a positive relationship between price and weight of diamond (carat) and no outliers. Thus, we can now proceed with our model.

We now employ the ordinary least squares (OLS) method and fit the linear regression model (Model_One):

$y = \beta_0 + \beta_1 X_1 + \mu_i,$

to our sample data with dependent variable (y) as price and independent variable (X) as weight of diamond (carat), where β_0 is the intercept, β_1 the slope, and μ the residuals.

We obtain the Best Linear Unbiased Estimator (BLUE) by minimizing the sum of the squared errors or residuals:

$$min \sum_{i=1}^{n} \mu_i^2$$

To achieve BLUE, the following properties must hold:

$$\sum_{i=1}^{n} \mu_i = 0$$

$$\sum_{i=1}^{n} X_i \, \mu_i = 0$$

$$\overline{y} = \hat{\beta}_0 + \hat{\beta}_1 \overline{X}.$$

```
library(MASS) # load package
mod1 <- lm(price ~ carat, data = MySampleData_2) # fit linear model and pass
it to an object called mod1.
```

We obtain the summary statistics of the fitted model with the summary( ) function as:

```
summary(mod1)

##
## Call:
## lm(formula = price ~ carat, data = MySampleData_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2488.6  -829.9   109.6   480.0  3198.6
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2082.4      408.0  -5.104  6.2e-06 ***
## carat         7304.0      449.9  16.235  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1257 on 46 degrees of freedom
## Multiple R-squared:  0.8514, Adjusted R-squared:  0.8482
## F-statistic: 263.6 on 1 and 46 DF,  p-value: < 2.2e-16
```

From the summary statistics, we deduced that the fitted model is:

$$\hat{y} = -2082.4 + 7304.0X_1.$$

The standard error (std. Error) tells us on average how much is each observation missing our predictions. The higher the standard error, the worse fitting the model is.

t-value = $coefficient/std.Error$. The higher the t-value, the more significant the variable is

The p-values test that the coefficients are zero for the null hypothesis. p-value is the smallest level of significance in which we will reject the null hypothesis in favor of the alternative hypothesis. Therefore, we reject the null hypothesis at 5% and 10% significant levels.

R-squared tells us how much we have explained; the goodness of fit (GoF). R-squared =0.8514 imply that 85.14% of the variations in y-variable is being explained by the X-variable, and there is a residual of $100\% - 85.14\% = 14.86\%$ going unexplained. The higher the R-squared the better the GoF and vice versa.

Adjusted-R-squared gives us information about the explanatory power of the model as new variables are added to it. As useful variables are added into the model, degrees of freedom (DF) decrease and Adjusted-R-squared increases to reflect the increase power of the model. If the added variables are not useful to the model, both DF and Adjusted-R-squared decreases to reflect the reduced power of the model.

F-statistics is the overall hypothesis test that all the coefficients in the model are zero.

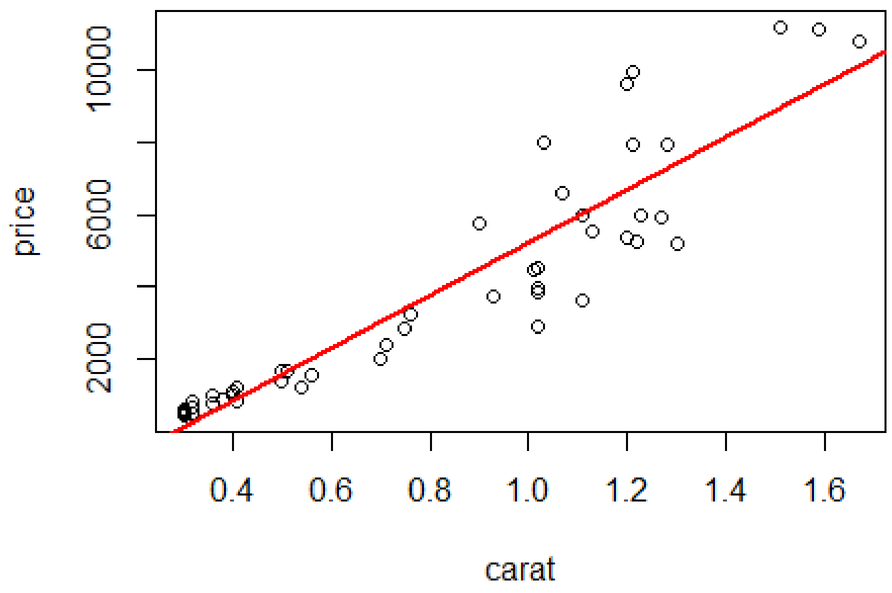
We could further explore the model with the attribute function; attributes( ), to see what particular attributes are stored in the object mod1 as:

```
attributes(mod1) # see what particular attributes are stored in the object
mod1

## $names
##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"
##
## $class
## [1] "lm"

 mod1$coefficients # get the coefficients of the fitted model

## (Intercept)       carat
##   -2082.403    7303.958

confint(mod1) # get the 95% confidence intervals of the coefficients

##                  2.5 %    97.5 %
## (Intercept) -2903.698 -1261.108
## carat        6398.373  8209.544

confint(mod1, level = 0.99) # get the 99% confidence intervals of the
coefficients
```

```
##                  0.5 %    99.5 %
## (Intercept) -3178.749 -986.0565
## carat        6095.092 8512.8243
```

Call the main plot and add the regression line for the model to the plot using the abline ( ) command:

```
plot(price~carat, data = MySampleData_2, main = " ")
abline(mod1, col = "red", lwd = 2)
```

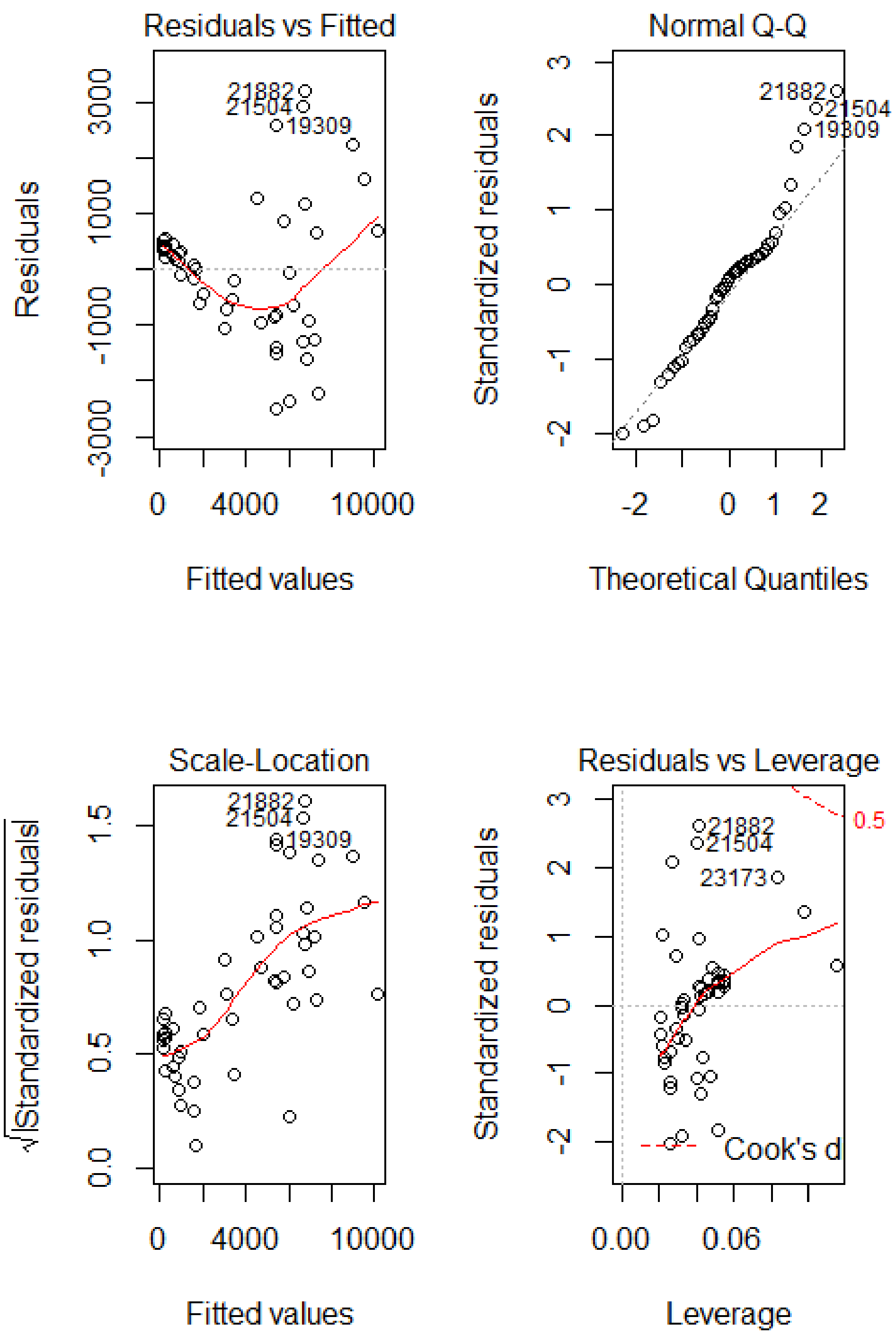


We can also produce the analysis of variance (anova) table for the linear regression model using the anova( ) command:

```
anova(mod1)

## Analysis of Variance Table
##
## Response: price
##           Df    Sum Sq   Mean Sq F value    Pr(>F)
## carat      1 416595581 416595581  263.57 < 2.2e-16 ***
## Residuals 46  72706379   1580573
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Let's look at the residual diagnostic plots. These diagnostic plots are very important for checking the model assumptions. For example, the residuals should be independent and normally distributed.

```r
par(mfcol=c(1,2)) # set the screen to show a matrix of plots: 1 row two columns
plot(mod1) # get residuals diagnostic plots
```



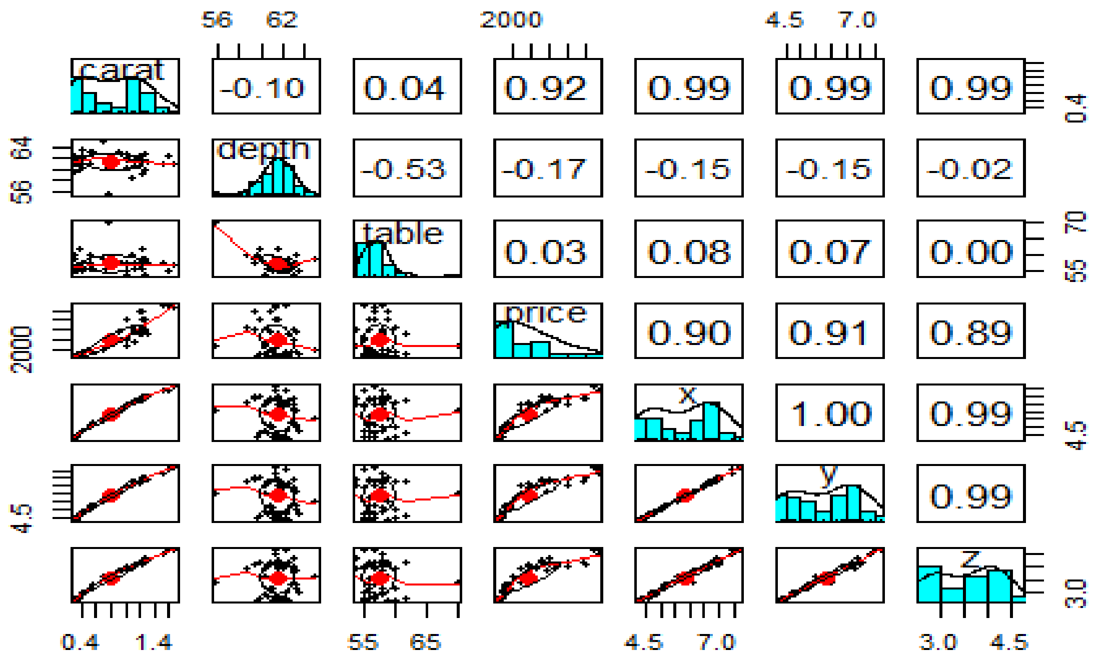
```
par(mfrow=c(1,1)) # set the screen back to its original format
```

The diagnostic plots show that the residuals are fairly normally distributed.

Multiple linear regression

One of the assumptions of multiple linear regression is that the explanatory variables are independent of each other. Let’s isolate the numeric variables in our sample data and visualize the relationship between the variables with the help of a scatter plot matrix as:

```
install.packages(“psych”) # install package
library(psych) # load package

pairs.panels(MySampleData_2[,c(1,5:10)])
```



Looking at the scatter plots above, a good candidate for the explanatory or independent variables should be carat, depth and table since the correlation between them are insignificant. We know that our data has no missing values and outliers, so we can go ahead and fit a multilinear regression model (Model_Two):

$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \mu_i,$

to our sample data, where price is the dependent variable (y), and independent variables X_1, X_2, and X_3 as carat, depth, and table, respectively. The model is as follows:

```
mod2 <- lm(price ~ carat + depth + table, data = MySampleData_2) # fit linear
model and pass it to an object called mod2.
```

```
summary(mod2)

##
## Call:
## lm(formula = price ~ carat + depth + table, data = MySampleData_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2980.6  -813.8    60.2   428.5  2870.1
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 16273.98   12057.20   1.350    0.184
## carat        7235.14     449.43  16.098   <2e-16 ***
## depth        -230.93     143.80  -1.606    0.115
## table         -71.65      82.15  -0.872    0.388
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1249 on 44 degrees of freedom
## Multiple R-squared:  0.8596, Adjusted R-squared:  0.8501
## F-statistic: 89.82 on 3 and 44 DF,  p-value: < 2.2e-16
```

The deduced model is:

$\hat{y} = 16273.98 + 7235.14X_1 - 230.93X_2 - 71.65X_3$.

Let's see how R-squared and Adjusted-R-squared will behave when we reduce the number of variables in the model and interchange them. That is, for Model_Three, we consider carat and depth as the explanatory variables:

$y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \mu_i$;

```
mod3 <- lm(price ~ carat + depth, data = MySampleData_2)

summary(mod3)

##
## Call:
## lm(formula = price ~ carat + depth, data = MySampleData_2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2659.99  -737.26    68.74   449.04  2890.79
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8064.4     7514.6   1.073    0.289
## carat         7243.4      448.1  16.163   <2e-16 ***
## depth         -164.4      121.6  -1.352    0.183
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1246 on 45 degrees of freedom
## Multiple R-squared:  0.8572, Adjusted R-squared:  0.8509
## F-statistic: 135.1 on 2 and 45 DF,  p-value: < 2.2e-16
```

For Model_Four, we consider carat and table as the explanatory variables:

$y = \beta_0 + \beta_1 X_1 + \beta_3 X_3 + \mu_i;$

```
mod4 <- lm(price ~ carat + table, data = MySampleData_2)

summary(mod4)

##
## Call:
## lm(formula = price ~ carat + table, data = MySampleData_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2494.5  -831.6   109.0   484.9  3201.0
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1986.718   4079.159  -0.487    0.629
## carat        7304.336    455.143  16.048   <2e-16 ***
## table          -1.671     70.853  -0.024    0.981
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1271 on 45 degrees of freedom
## Multiple R-squared:  0.8514, Adjusted R-squared:  0.8448
## F-statistic: 128.9 on 2 and 45 DF,  p-value: < 2.2e-16
```

And finally, we consider depth and table as the explanatory variables for Model_Five:

$y = \beta_0 + \beta_2 X_2 + \beta_3 X_3 + \mu_i,$

```
mod5 <- lm(price ~ depth + table, data = MySampleData_2)



summary(mod5)

##
## Call:
## lm(formula = price ~ depth + table, data = MySampleData_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3929.0 -2642.2  -545.3  1779.3  7452.5
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 37388.70   31109.29   1.202    0.236
## depth        -452.87     371.53  -1.219    0.229
## table         -99.61     213.18  -0.467    0.643
##
## Residual standard error: 3243 on 45 degrees of freedom
## Multiple R-squared:  0.0329, Adjusted R-squared:  -0.01008
## F-statistic: 0.7655 on 2 and 45 DF,  p-value: 0.4711
```

Looking at the summary statistics; moving from Model_Two to Model_Three, R-squared decreases and Adjusted-R-squared increases. This tells us that the explanatory power of the model is better without the table variable. In other words, adding the table variable to the model decreases the explanatory power of the model. Model_Five is totally useless as Adjusted-R-squared is almost zero. Therefore, in conclusion, the best model that we can use to forecast the price of diamond for the multilinear regression models is Model_Three, which from the summary statistics is:

$$\hat{y} = 8064.4 + 7243.4X_1 - 164.4X_2.$$