Math 32 Course Project

Armaan Kapoor

November 26, 2019

Introduction

My dataset is called “diamonds.csv”. It is adopted from https://www.kaggle.com/shivam2503/diamonds (https://www.kaggle.com/shivam2503/diamonds). This is a classical dataset that contains data on about 53940 diamonds with 10 different variables, with one of them being price.

Purpose

The main goal of this project is to use the data to ultimately devise a model (albeit a best-fit line or model) that can be used to determine the price of a diamond based on the other 9 variables.

Side Goal

I also want to conjure up other analytical data, such as correlation, between other variables beside the price. For example, I will show the correlation between clarity and carat of a diamond, to show how a high number of one variable may correlate to a high number of the other variable. Ultimately, I hope to use this project to showcase all my R and data analytical skills I learned in Math 32.

Data Analysis

Holistic Look

With the code below, can see the first couple of values with the head() function and the columns. With the summary() function I get satistics about each column in the data set.

What each column is: 1. Carat = Carat weight of the diamond 2. Cut = Cut quality of the diamond 3. Color = Color of the diamond. D being the best and J as the worst 4. Depth = Depth percentage: The height of a diamond, measured from the culet to the table, divided by its average girdle diameter 5. Table = table percentage: The width of the diamond’s table expressed as a percentage of its average diameter 6. Price = Price of the diamond 7. X = Length mm 8. Y = Width mm 9. Z = Depth mm

```
diamonds = read.csv(file="diamonds.csv", header=TRUE, sep = ",")

head(diamonds)
```

```
##   X carat       cut color clarity depth table price    x    y    z
## 1 1  0.23     Ideal     E     SI2  61.5    55   326 3.95 3.98 2.43
## 2 2  0.21   Premium     E     SI1  59.8    61   326 3.89 3.84 2.31
## 3 3  0.23      Good     E     VS1  56.9    65   327 4.05 4.07 2.31
## 4 4  0.29   Premium     I     VS2  62.4    58   334 4.20 4.23 2.63
## 5 5  0.31      Good     J     SI2  63.3    58   335 4.34 4.35 2.75
## 6 6  0.24 Very Good     J    VVS2  62.8    57   336 3.94 3.96 2.48
```

The summary() command produces an output similar to the table() function on those columns that are not numeric. The ones that are calculates the Minimum, 1st Quartile, Median, Mean, 3rd Quartile, and Maximum.

```
summary(diamonds)
```

```
##        X             carat               cut        color        clarity
##  Min.   :    1   Min.   :0.2000   Fair     : 1610   D: 6775   SI1    :13065
##  1st Qu.:13486   1st Qu.:0.4000   Good     : 4906   E: 9797   VS2    :12258
##  Median :26971   Median :0.7000   Ideal    :21551   F: 9542   SI2    : 9194
##  Mean   :26971   Mean   :0.7979   Premium  :13791   G:11292   VS1    : 8171
##  3rd Qu.:40455   3rd Qu.:1.0400   Very Good:12082   H: 8304   VVS2   : 5066
##  Max.   :53940   Max.   :5.0100                     I: 5422   VVS1   : 3655
##                                                     J: 2808   (Other): 2531
##      depth           table           price             x
##  Min.   :43.00   Min.   :43.00   Min.   :  326   Min.   : 0.000
##  1st Qu.:61.00   1st Qu.:56.00   1st Qu.:  950   1st Qu.: 4.710
##  Median :61.80   Median :57.00   Median : 2401   Median : 5.700
##  Mean   :61.75   Mean   :57.46   Mean   : 3933   Mean   : 5.731
##  3rd Qu.:62.50   3rd Qu.:59.00   3rd Qu.: 5324   3rd Qu.: 6.540
##  Max.   :79.00   Max.   :95.00   Max.   :18823   Max.   :10.740
##
##        y                z
##  Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 4.720   1st Qu.: 2.910
##  Median : 5.710   Median : 3.530
##  Mean   : 5.735   Mean   : 3.539
##  3rd Qu.: 6.540   3rd Qu.: 4.040
##  Max.   :58.900   Max.   :31.800
##
```

Priming the dataset

Now we need to see if all the numeric data is actually numeric. This is done in the below chunk by running the command is.numeric() on each of the columns that are supposed to contain all the numbers.

```
carat <- sapply(diamonds$carat, is.numeric)
table(carat) #returned all 53940 are TRUE
#Because all are numeric, we must turn all values to numbers
diamonds$carat <- sapply(diamonds$carat, as.numeric)
head(diamonds$carat)

depth <- sapply(diamonds$depth, is.numeric)
table(depth) #returned all 53940 are TRUE
#Because all are numeric, we must turn all values to numbers
diamonds$depth <- sapply(diamonds$depth, as.numeric)
head(diamonds$depth)

tab <- sapply(diamonds$table, is.numeric)
table(tab) #returned all 53940 are TRUE
#Because all are numeric, we must turn all values to numbers
diamonds$tab <- sapply(diamonds$table, as.numeric)
head(diamonds$tab)

price <- sapply(diamonds$price, is.numeric)
table(price) #returned all 53940 are TRUE
#Because all are numeric, we must turn all values to numbers
diamonds$price <- sapply(diamonds$price, as.numeric)
head(diamonds$price)

x <- sapply(diamonds$x, is.numeric)
table(x) #returned all 53940 are TRUE
#Because all are numeric, we must turn all values to numbers
diamonds$x <- sapply(diamonds$x, as.numeric)
head(diamonds$x)

y <- sapply(diamonds$y, is.numeric)
table(y) #returned all 53940 are TRUE
#Because all are numeric, we must turn all values to numbers
diamonds$y <- sapply(diamonds$y, as.numeric)
head(diamonds$y)

z <- sapply(diamonds$z, is.numeric)
table(z) #returned all 53940 are TRUE
#Because all are numeric, we must turn all values to numbers
diamonds$z <- sapply(diamonds$z, as.numeric)
head(diamonds$z)
```

A Side note:

We have three variables X, Y, Z. Looking at them individually is a solution but what would be more helpful if they were looked as one variable, a combination of X, Y, and Z, which would be X*Y*Z, also known as the volume.

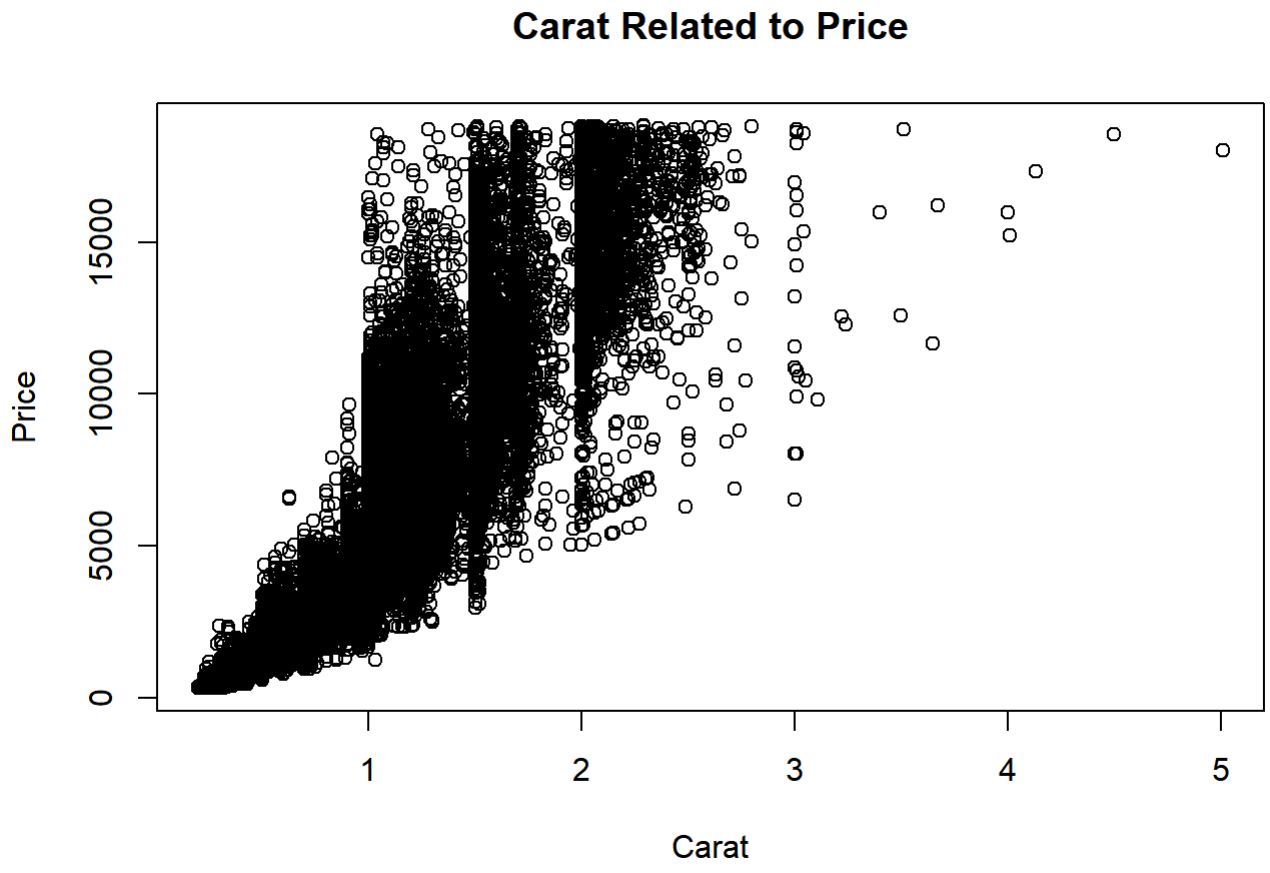
```
diamonds$volume = diamonds$x*diamonds$y*diamonds$z

head(diamonds)
```

```
##   X carat       cut color clarity depth table price    x    y    z tab   volume
## 1 1  0.23     Ideal     E     SI2  61.5    55   326 3.95 3.98 2.43  55 38.20203
## 2 2  0.21   Premium     E     SI1  59.8    61   326 3.89 3.84 2.31  61 34.50586
## 3 3  0.23      Good     E     VS1  56.9    65   327 4.05 4.07 2.31  65 38.07688
## 4 4  0.29   Premium     I     VS2  62.4    58   334 4.20 4.23 2.63  58 46.72458
## 5 5  0.31      Good     J     SI2  63.3    58   335 4.34 4.35 2.75  58 51.91725
## 6 6  0.24 Very Good     J    VVS2  62.8    57   336 3.94 3.96 2.48  57 38.69395
```

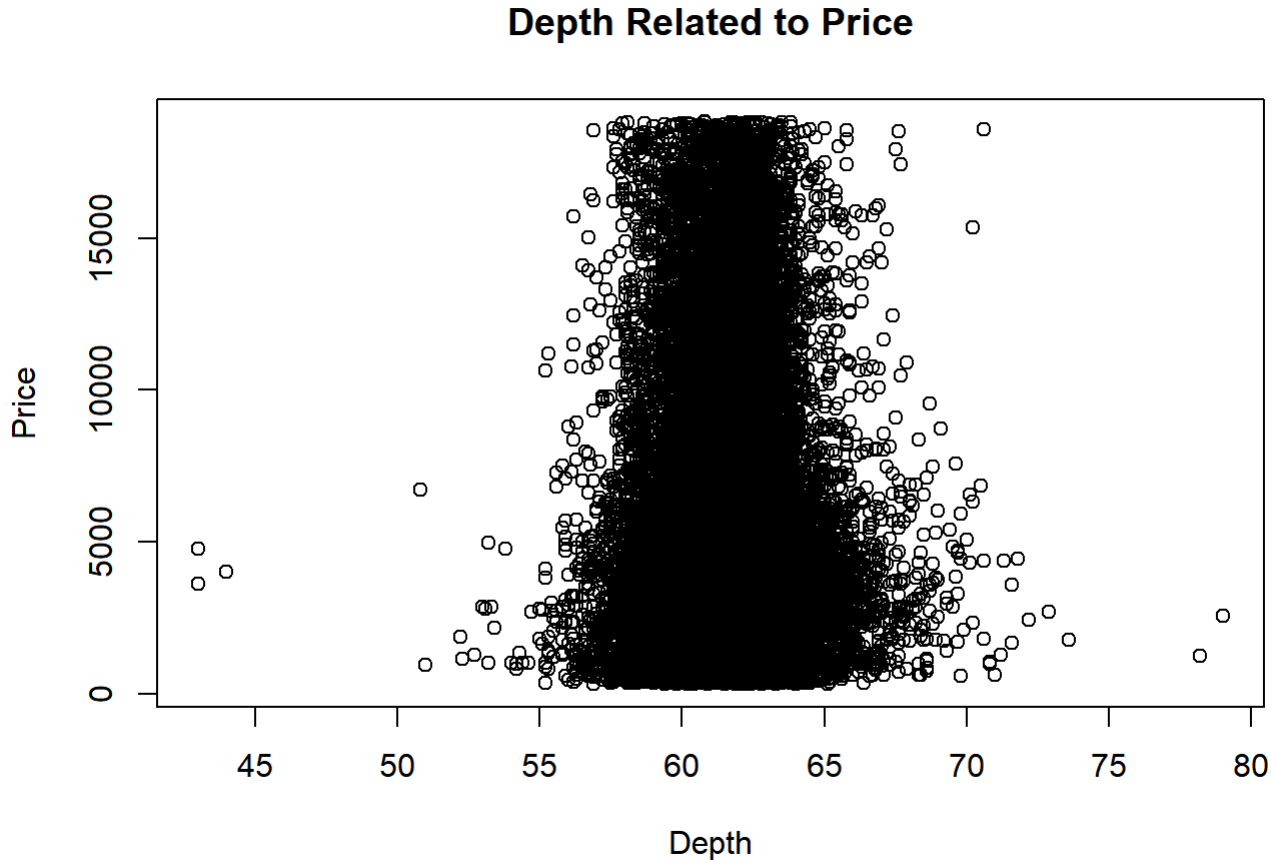
Correlation Between Numeric Variables and Price

Here I simply plotted each of the main variables including volume and excluding x, y, and z, against the price column. The purpose of this is to get an overview of the data and determine any easily-recognized patterns

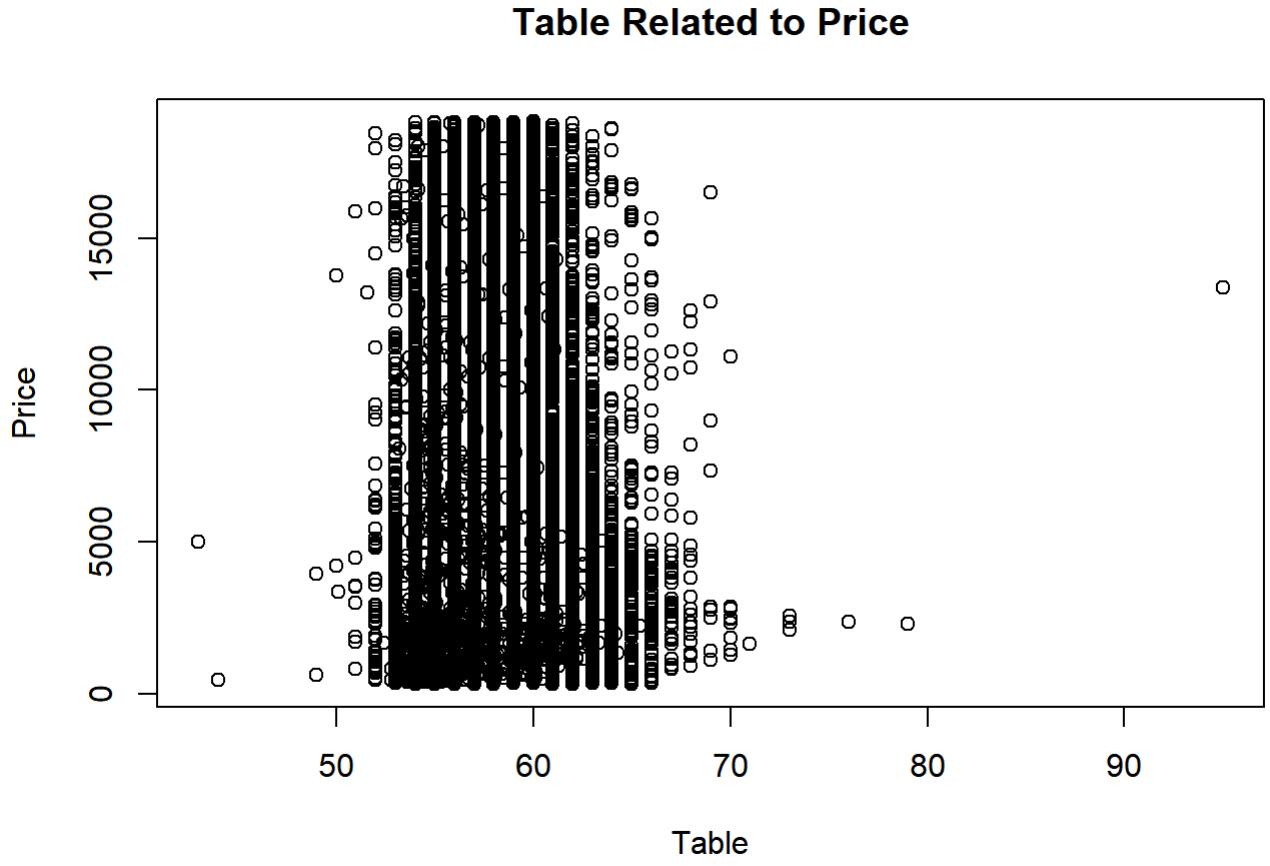
```
plot(diamonds$carat, diamonds$price, main = "Carat Related to Price", xlab="Carat",ylab="Price")
```



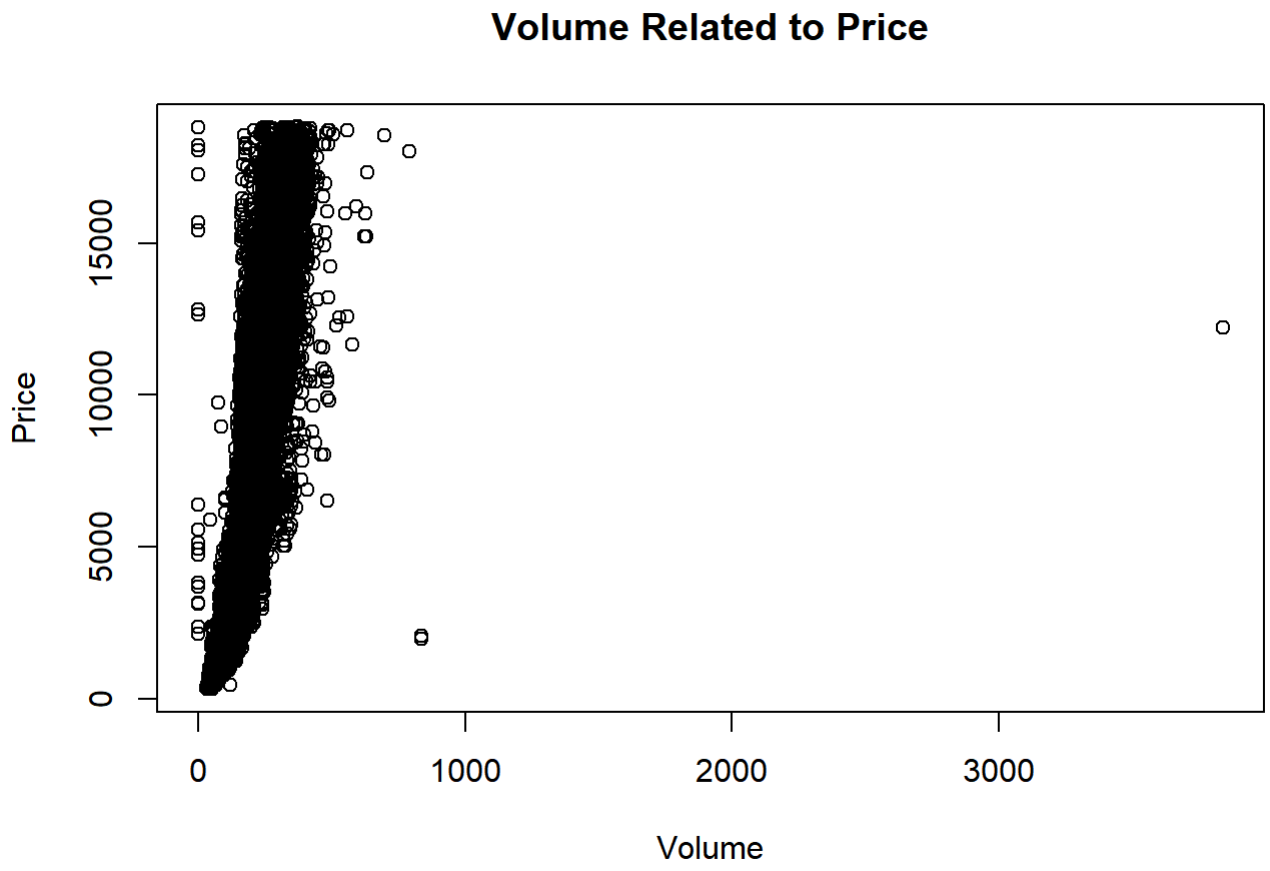

```
plot(diamonds$depth, diamonds$price, main = "Depth Related to Price", xlab="Depth",ylab="Price")
```




```
plot(diamonds$table, diamonds$price, main = "Table Related to Price", xlab="Table",ylab="Price")
```



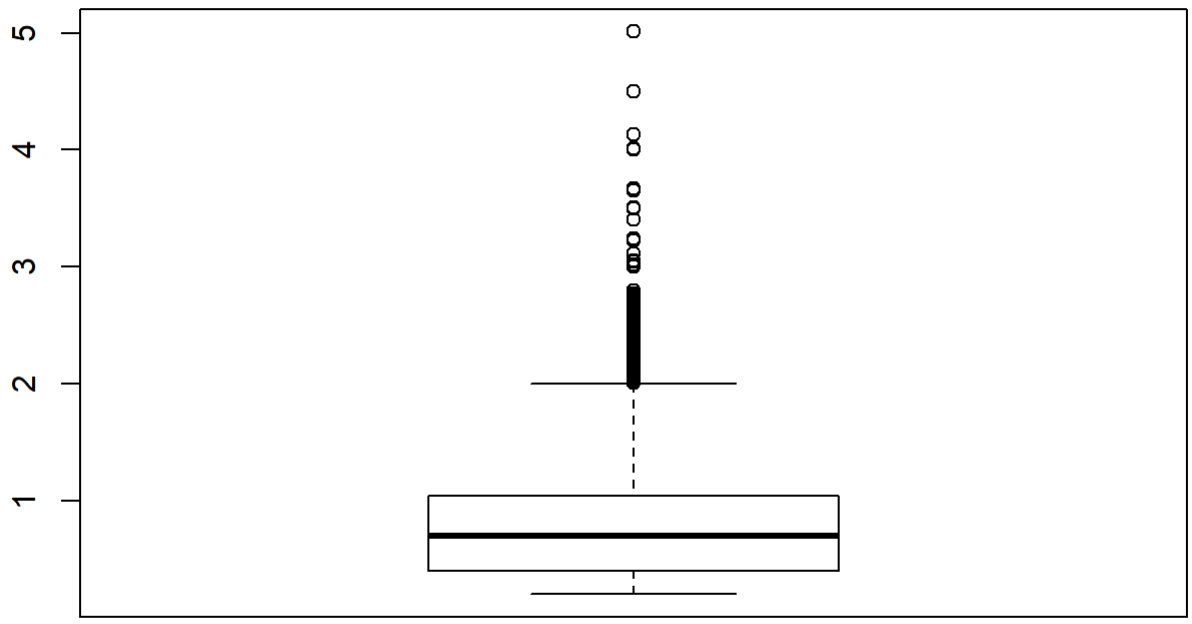

```
plot(diamonds$volume, diamonds$price, main = "Volume Related to Price", xlab="Volume",ylab="Pric
e")
```



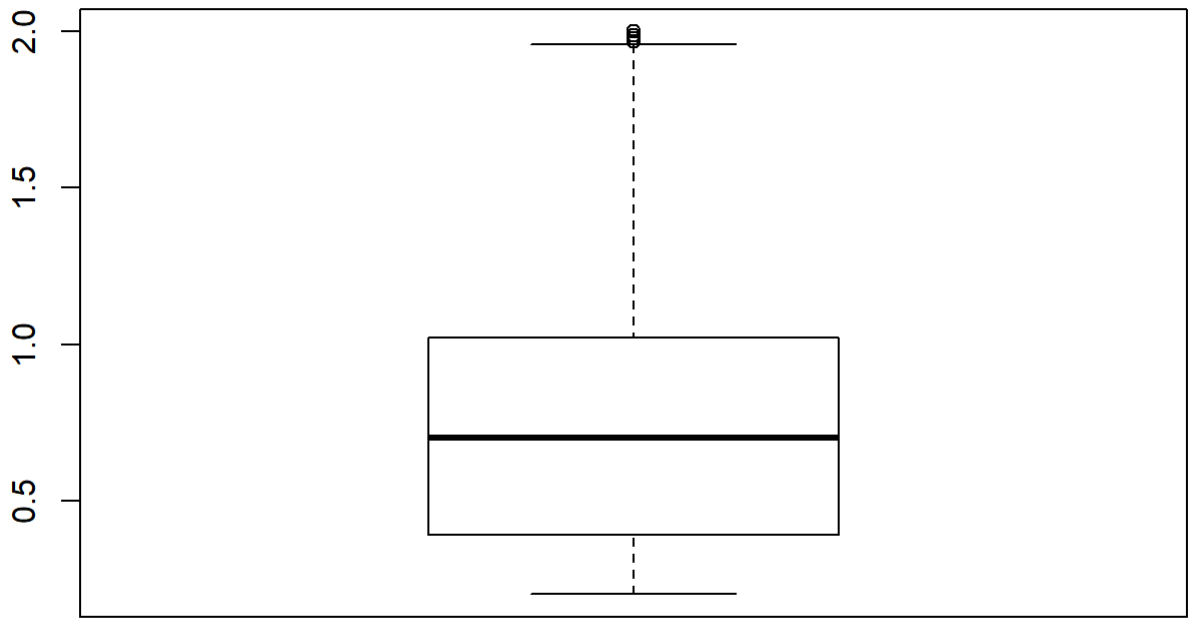
Correct for Outliers

These following graphs are the same as above, except now they will correct for outliers that hinder the overview of the graphs. This is done by using a box and wisker plot to see where the outliers exist, then comes the process of removing the outliers. When we have identified the outliers, then the entire row is excluded from the data frame.

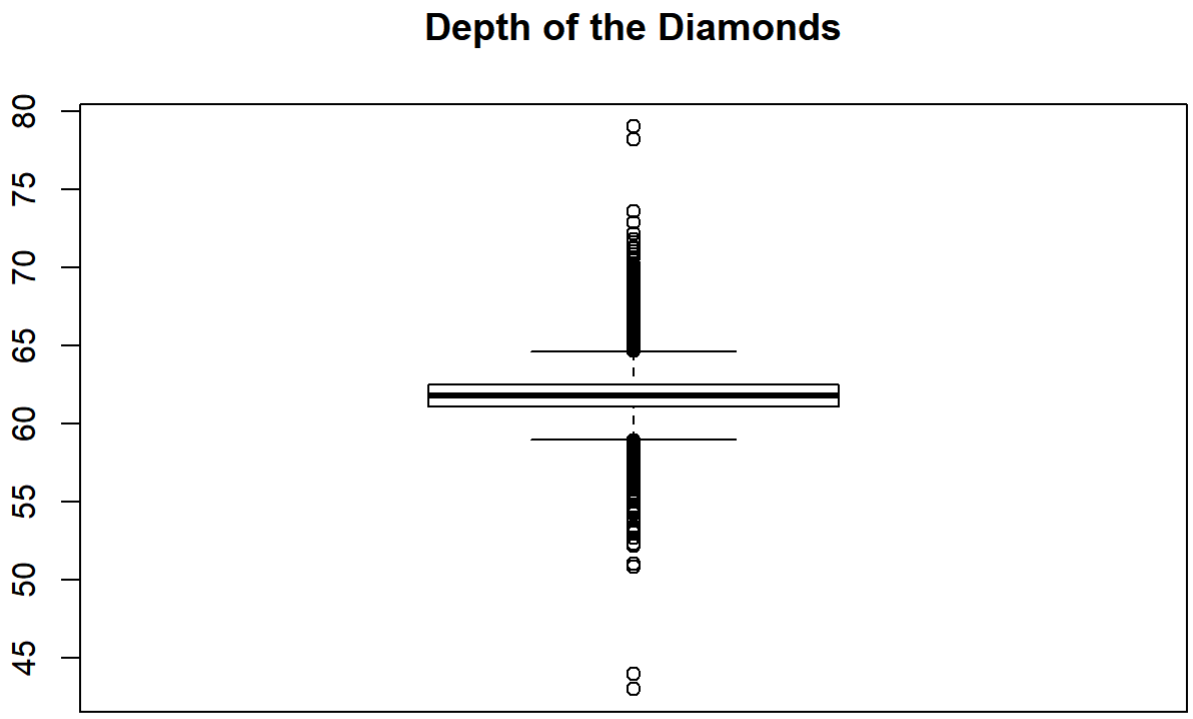
```
library(gridExtra)
outliers <- boxplot(diamonds$carat, main="Carat Weight of the Diamonds")$out
```



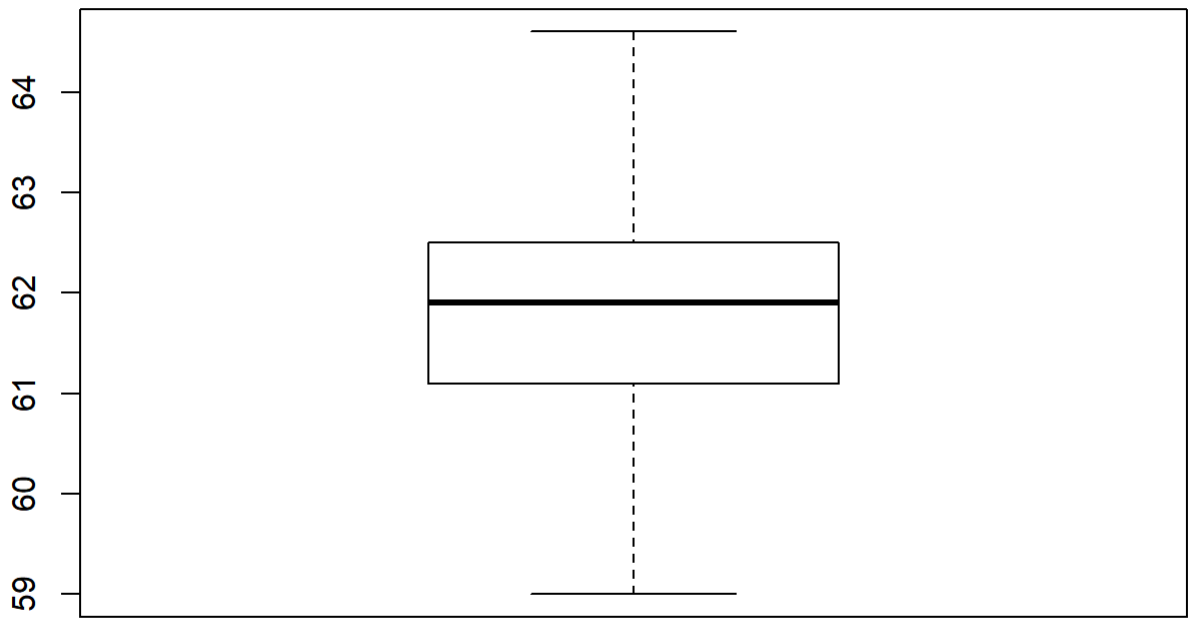

```
diamonds <- diamonds[-which(diamonds$carat %in% outliers),]
boxplot(diamonds$carat, main="Carat Weight After Outliers removed")
```



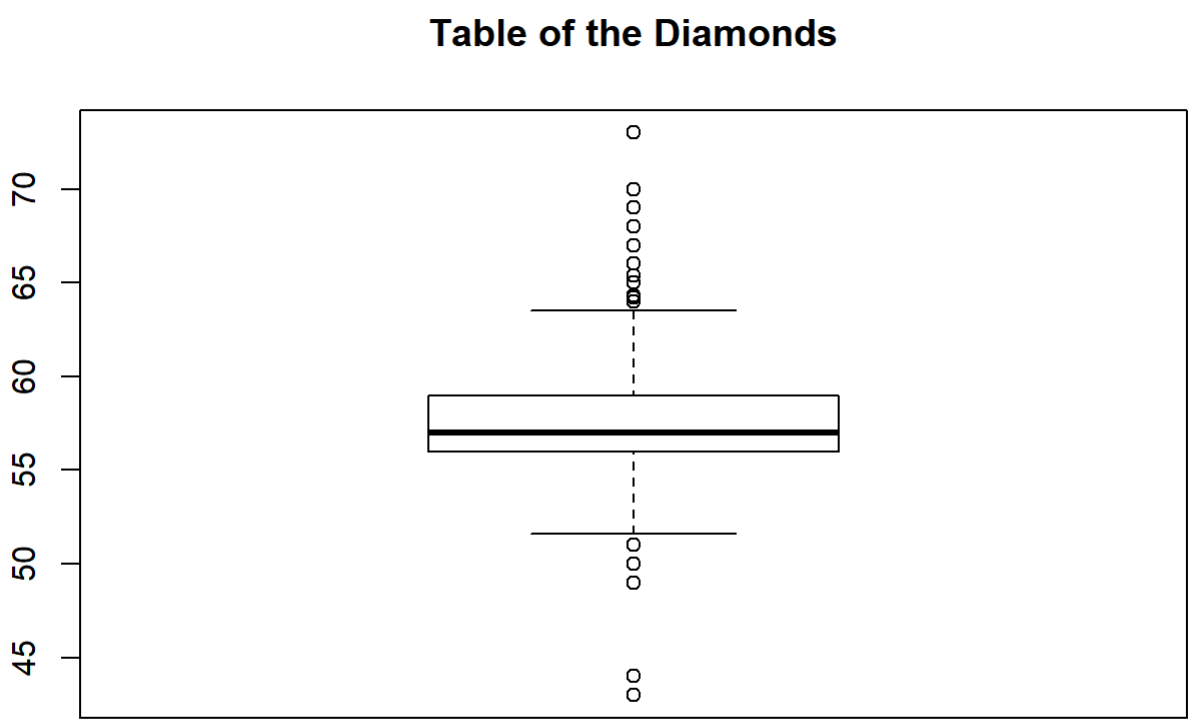

```
outliers <- boxplot(diamonds$depth, main="Depth of the Diamonds")$out
```



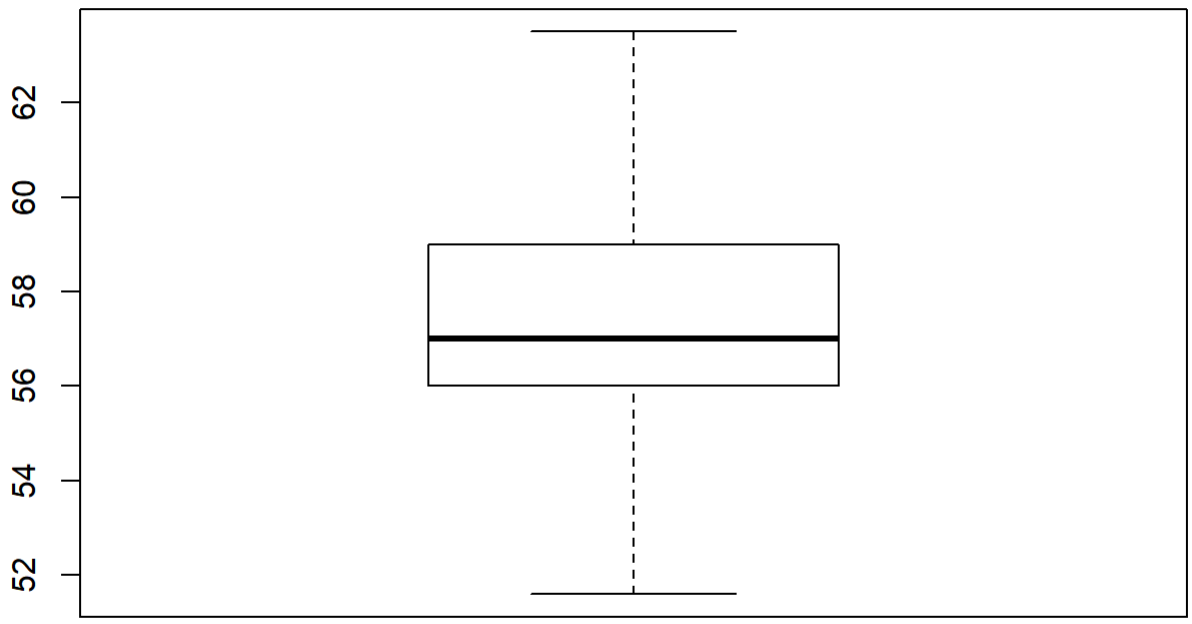

```
diamonds <- diamonds[-which(diamonds$depth %in% outliers),]
boxplot(diamonds$depth, main="Depth After Outliers removed")
```



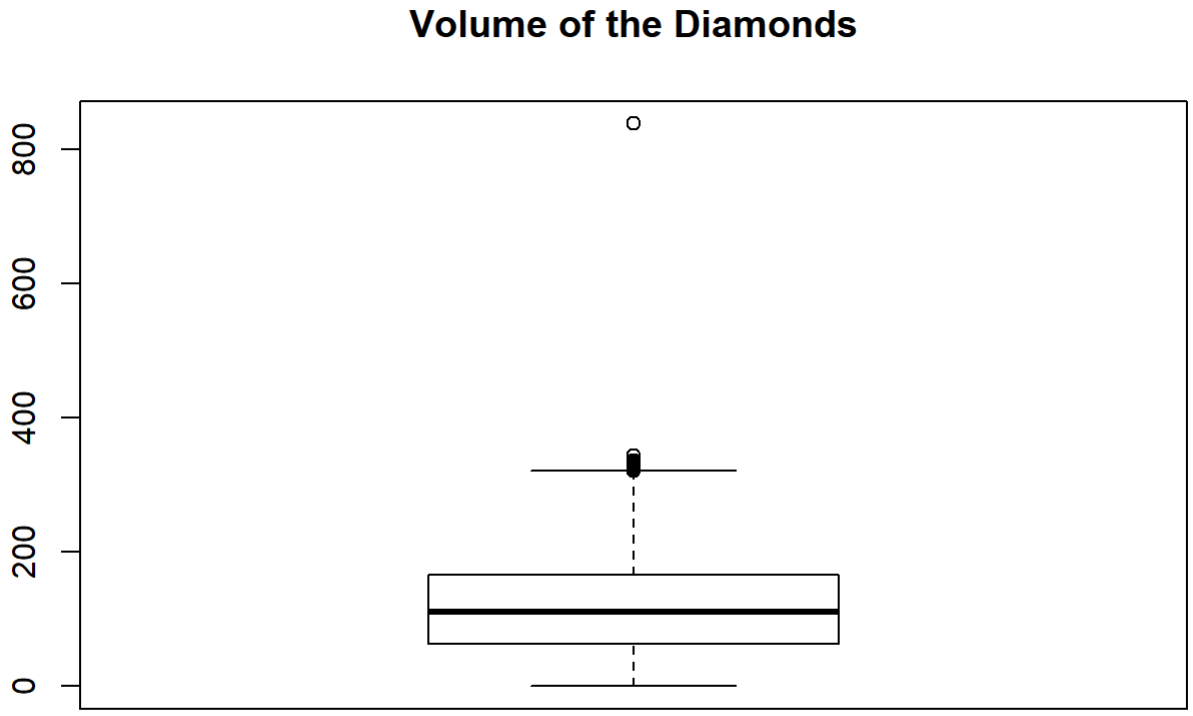

```
outliers <- boxplot(diamonds$table, main="Table of the Diamonds")$out
```



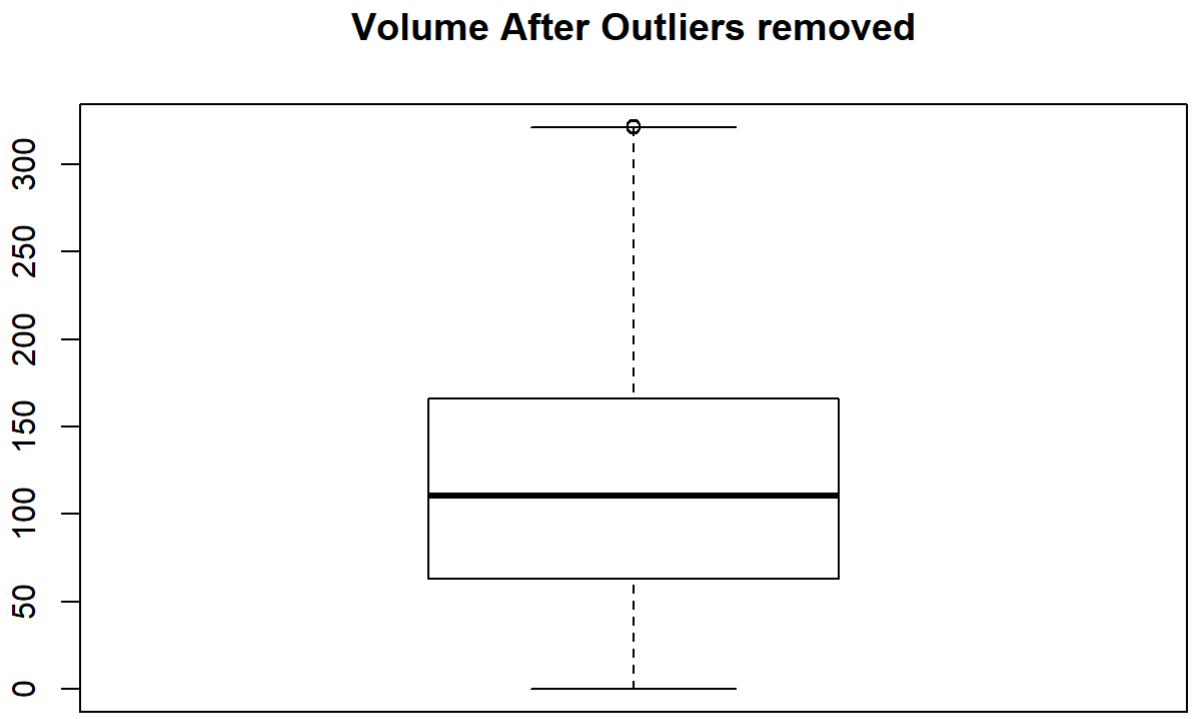

```
diamonds <- diamonds[-which(diamonds$table %in% outliers),]
boxplot(diamonds$table, main="Table After Outliers removed")
```




```
outliers <- boxplot(diamonds$volume, main="Volume of the Diamonds")$out
```



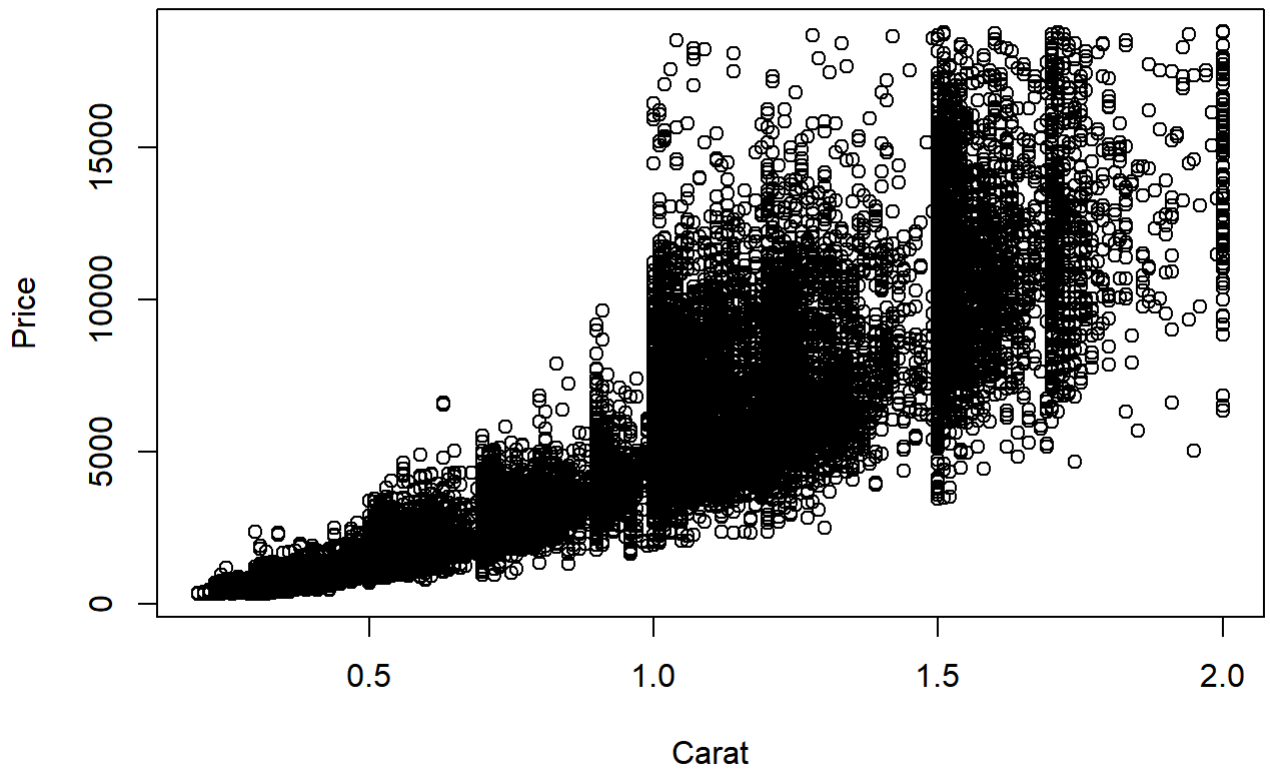

```
diamonds <- diamonds[-which(diamonds$volume %in% outliers),]
boxplot(diamonds$volume, main="Volume After Outliers removed")
```



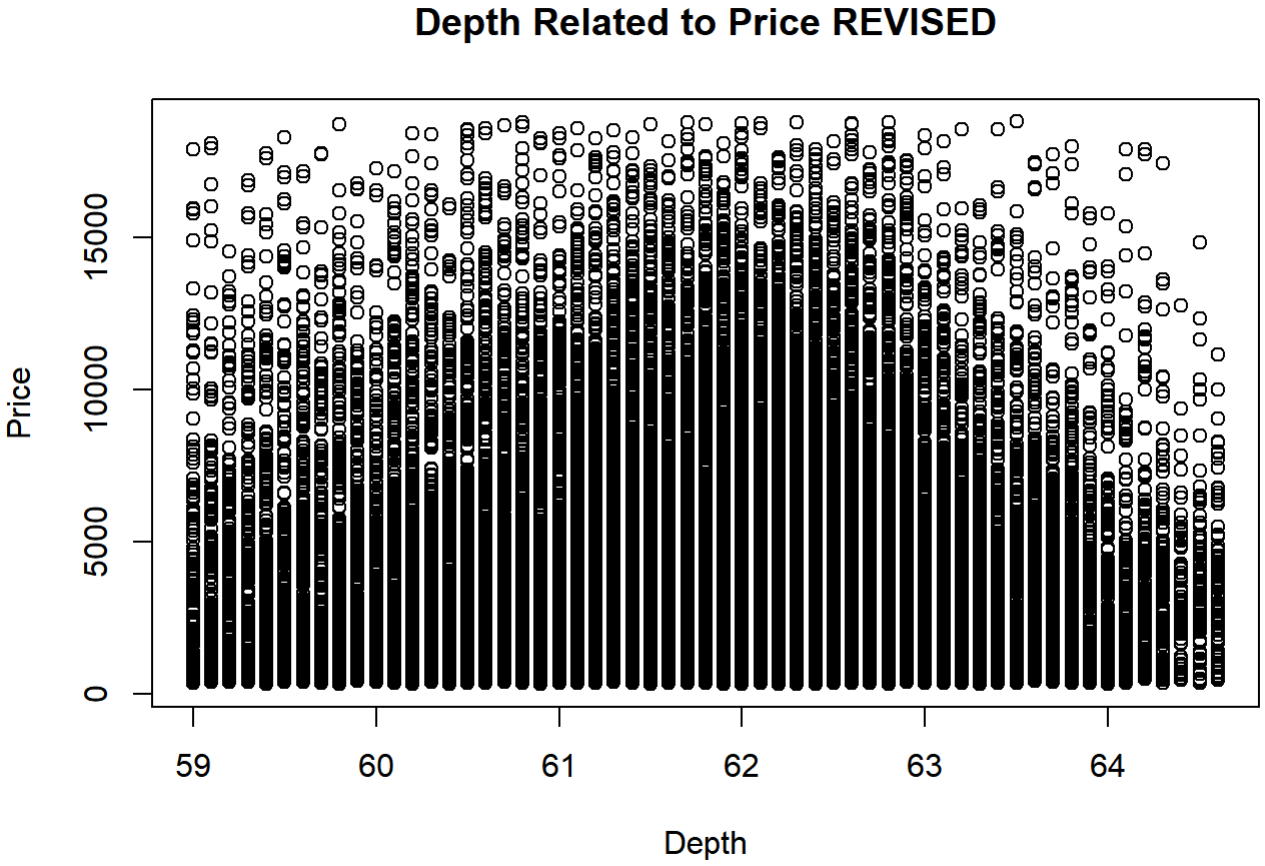
Plot Above Variables against Price

Now plotting against price will show any obvious relations/correlation between the variables. Then, a type of regression can be assigned

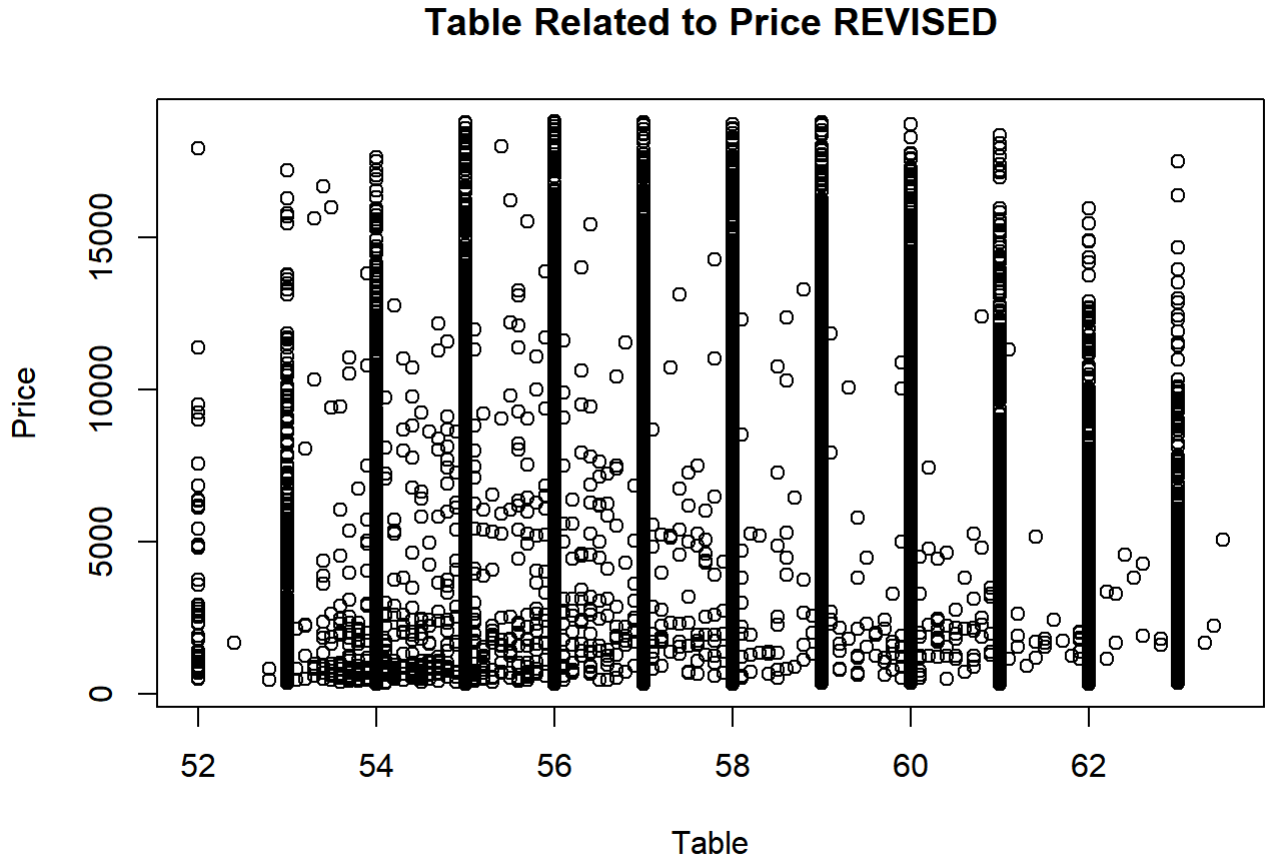
```
plot(diamonds$carat, diamonds$price, main = "Carat Related to Price REVISED", xlab="Carat",ylab=
"Price")
```



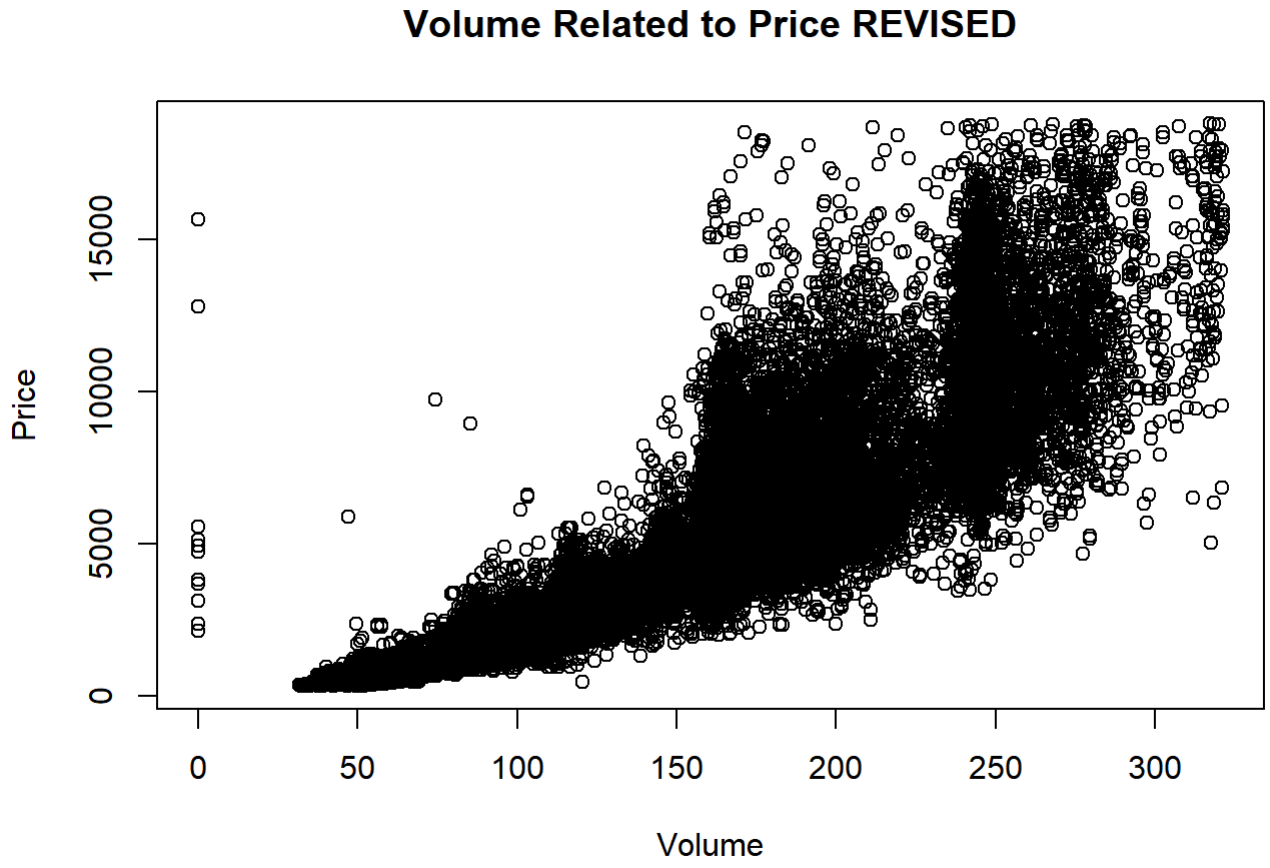

```
plot(diamonds$depth, diamonds$price, main = "Depth Related to Price REVISED", xlab="Depth",ylab=
"Price")
```




```
plot(diamonds$table, diamonds$price, main = "Table Related to Price REVISED", xlab="Table",ylab=
"Price")
```




```
plot(diamonds$volume, diamonds$price, main = "Volume Related to Price REVISED", xlab="Volume",yl
ab="Price")
```



HW #10 Questions

1. see above document
2. see above document
3. I have to do two main things. I will make regression functions for the plots against Price. The second things I will do is hopefully weigh all the variables to form one singular regression function to use.