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 it is necessary 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*YZ*, also known as the volume.

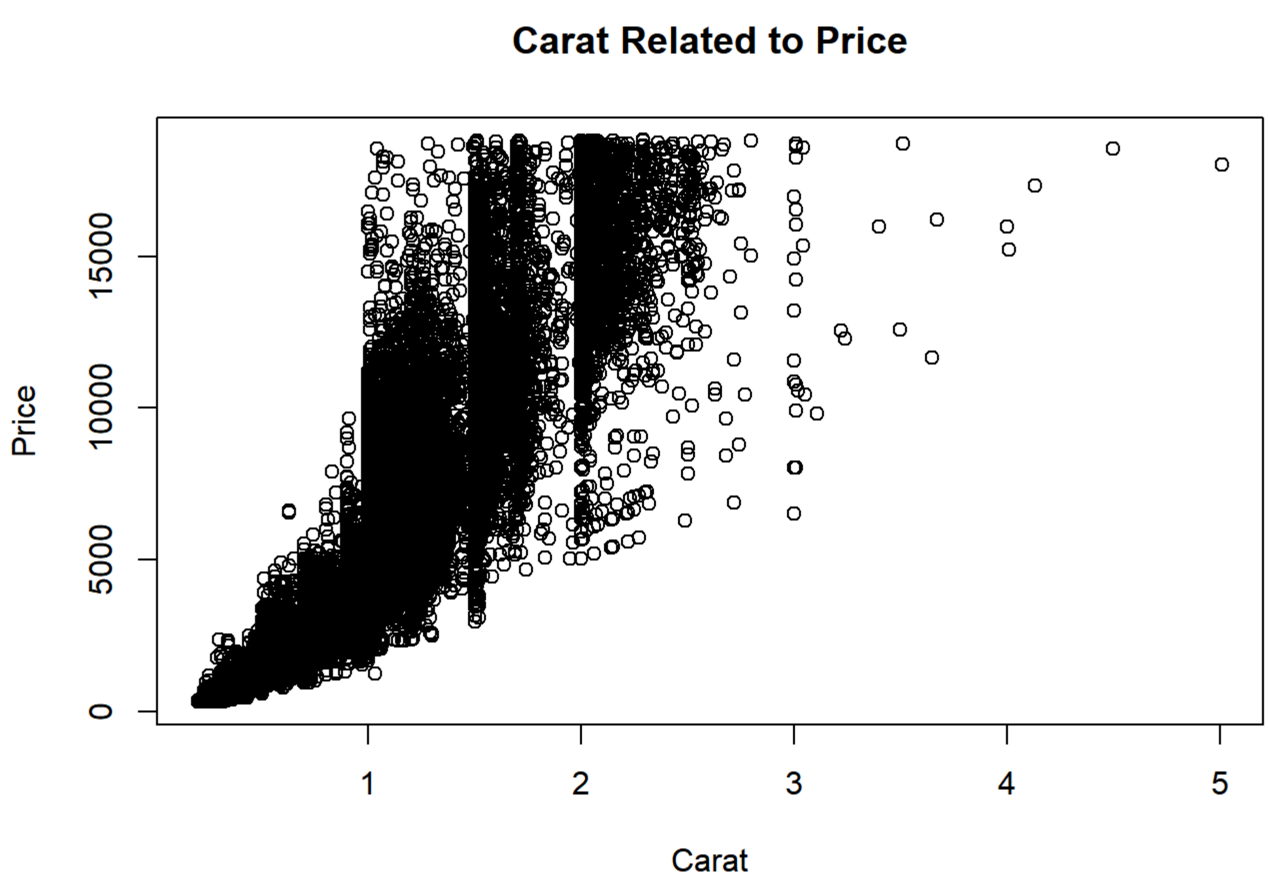
```
diamonds$volume = diamonds$x*diamonds$y*diamonds$z

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
```

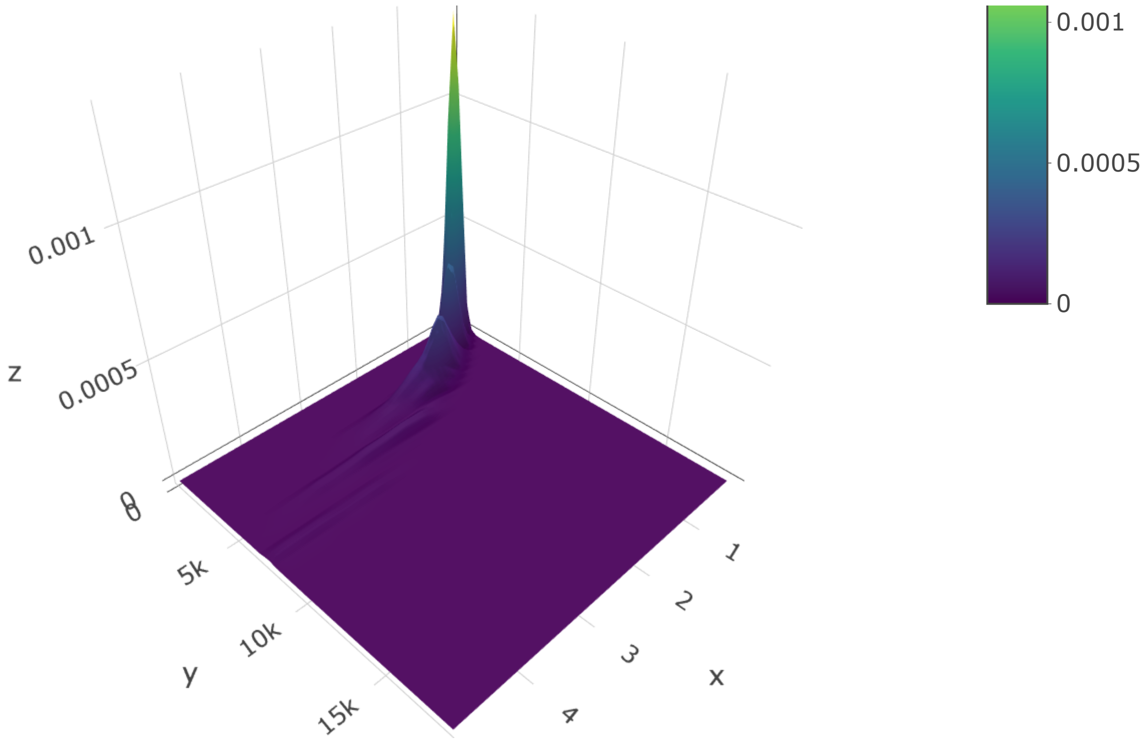
Correlation Between Numeric Variables and Price

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. Due to an overlap in points on the scatter plot it is hard to determine whether a blot of points represents 10 points or 10000 points, thus a 3d representation of the density of the plot is also shown after each corresponding graph.

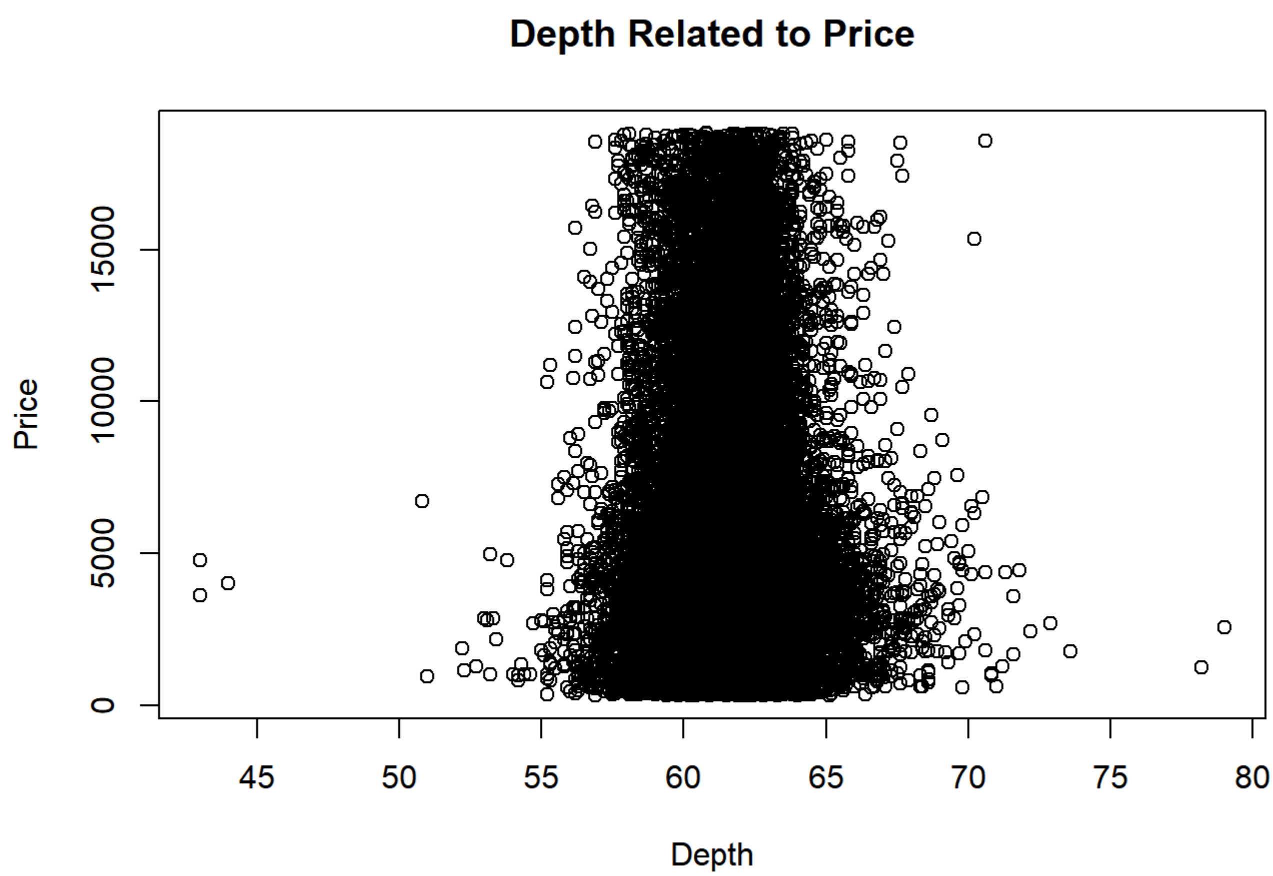
```
plot(diamonds$carat, diamonds$price, main = "Carat Related to Price", xlab="Carat",ylab="Price")
```



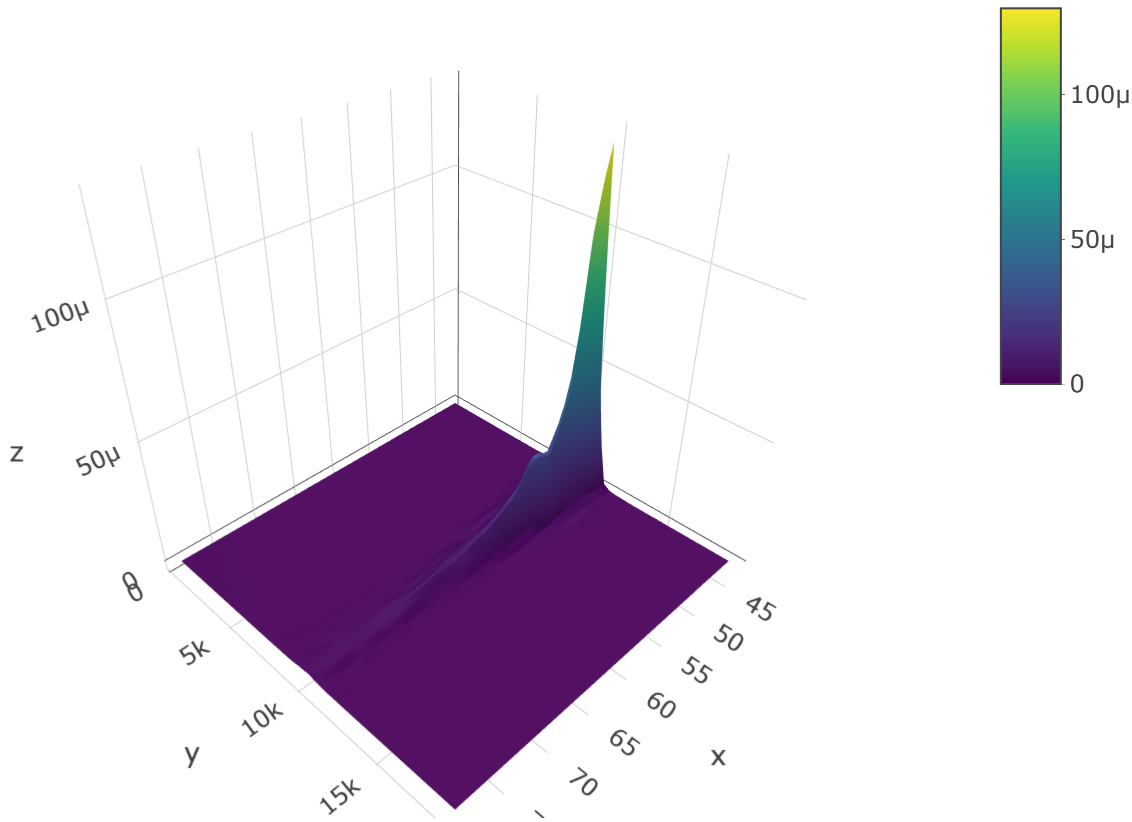

```
kd <- with(diamonds, MASS::kde2d(diamonds$carat, diamonds$price, n = 50))
plot_ly(x = kd$x, y = kd$y, z = kd$z) %>% add_surface()
```



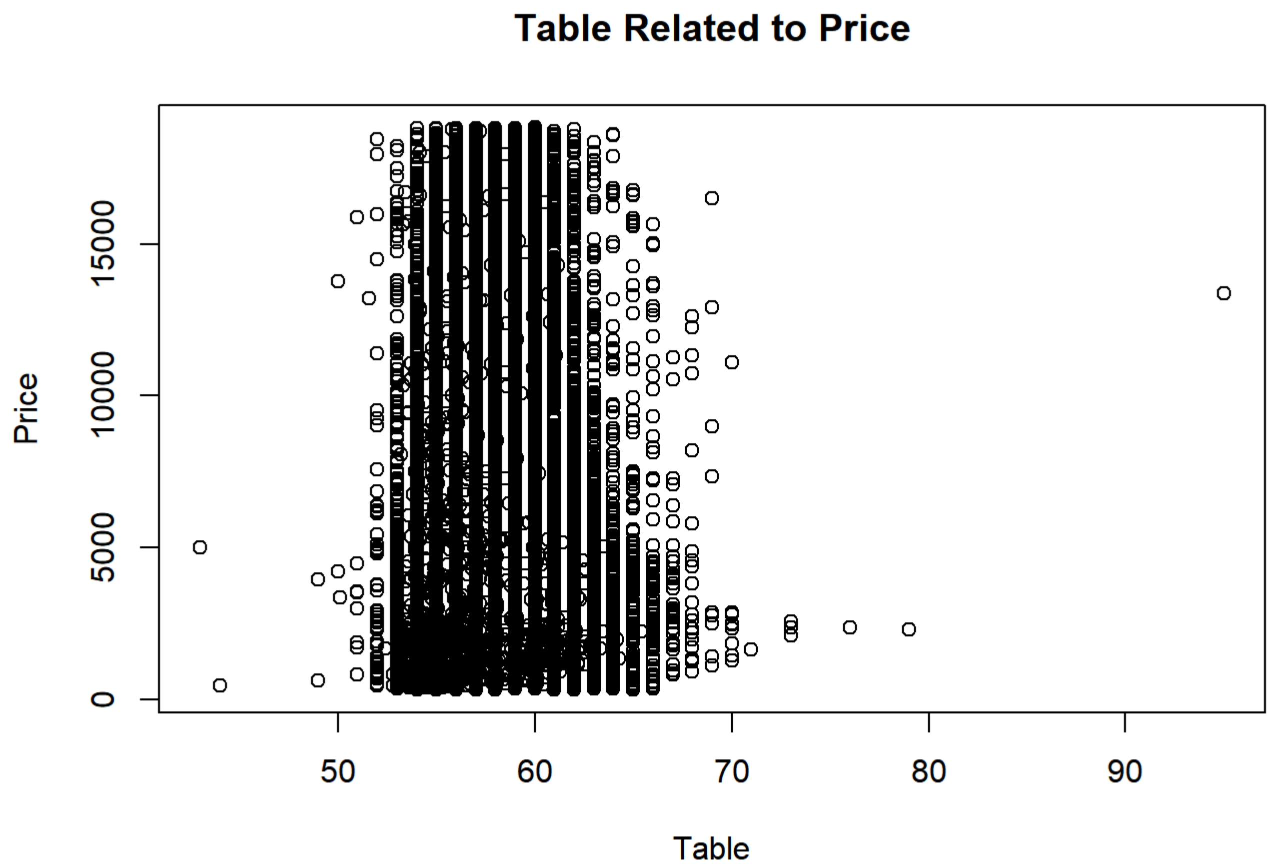

```
plot(diamonds$depth, diamonds$price, main = "Depth Related to Price", xlab="Depth",ylab="Price")
```



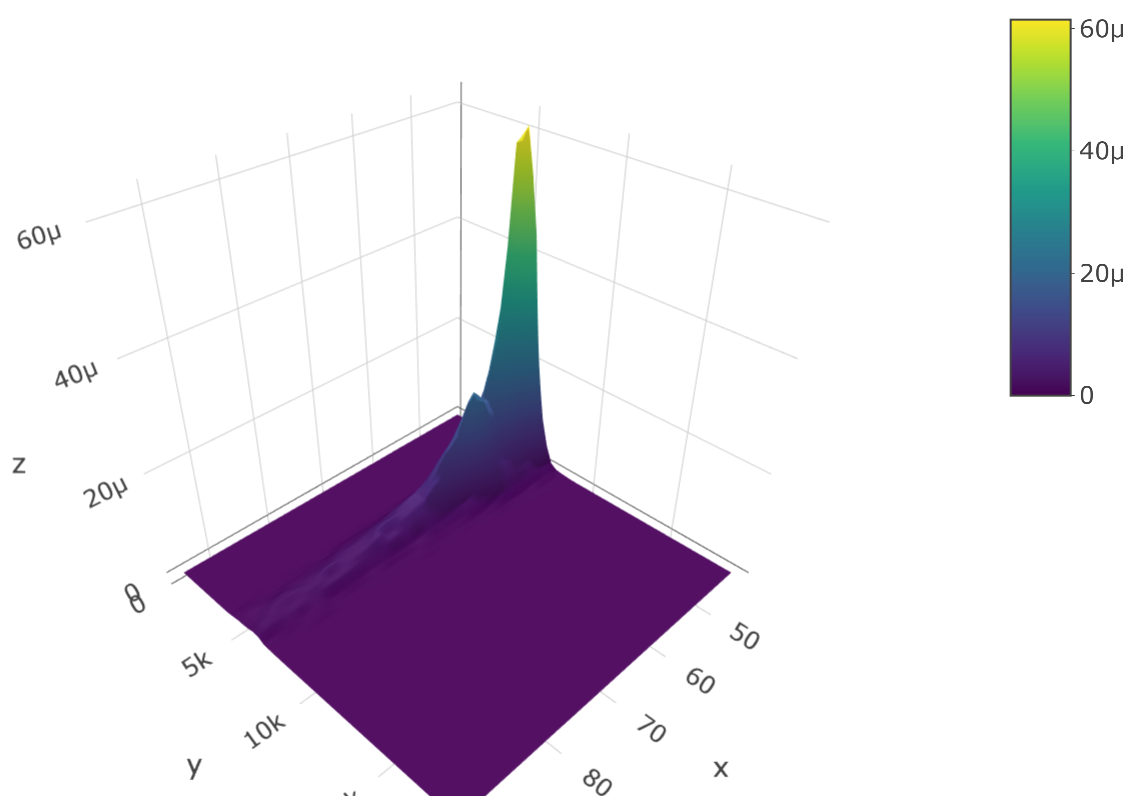
```
kd <- with(diamonds, MASS::kde2d(diamonds$depth, diamonds$price, n = 50))
plot_ly(x = kd$x, y = kd$y, z = kd$z) %>% add_surface()
```



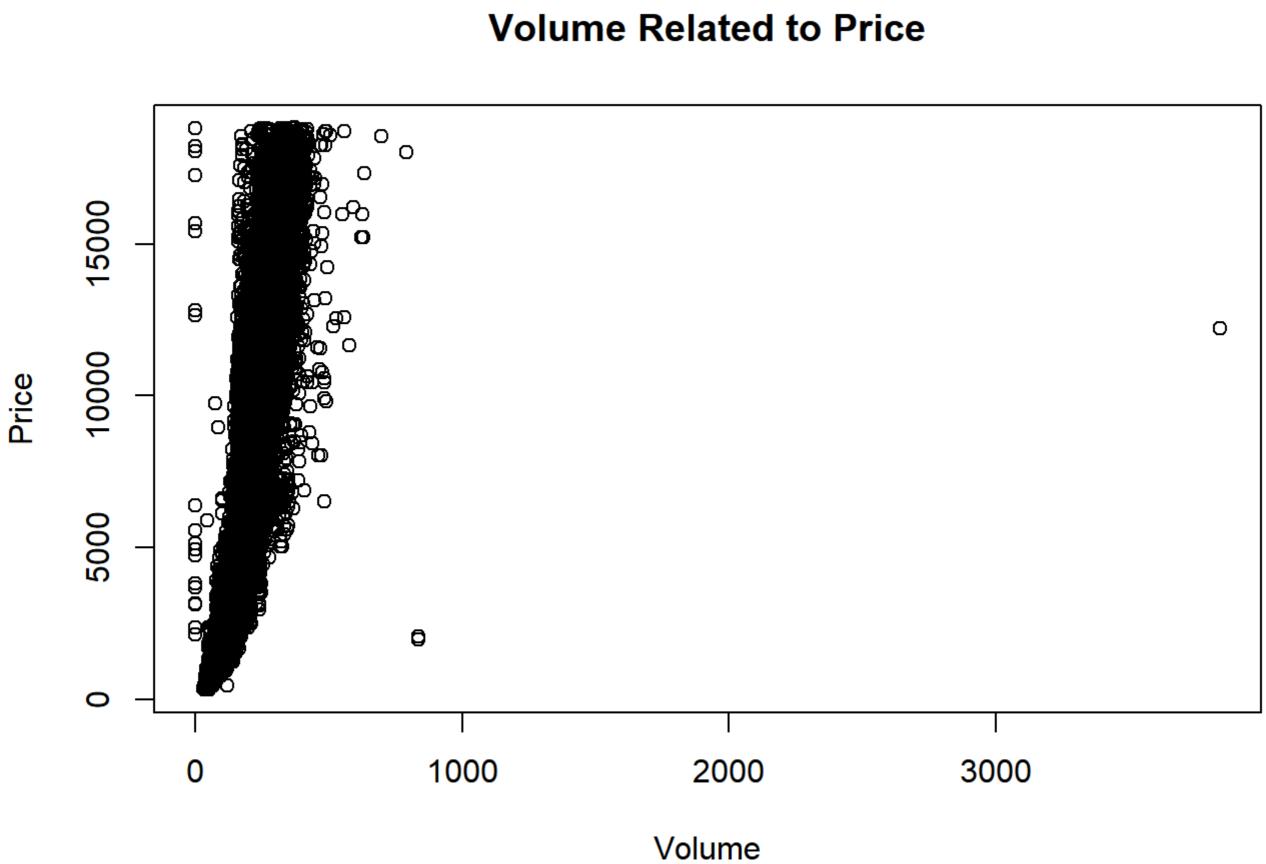

```
plot(diamonds$table, diamonds$price, main = "Table Related to Price", xlab="Table",ylab="Price")
```



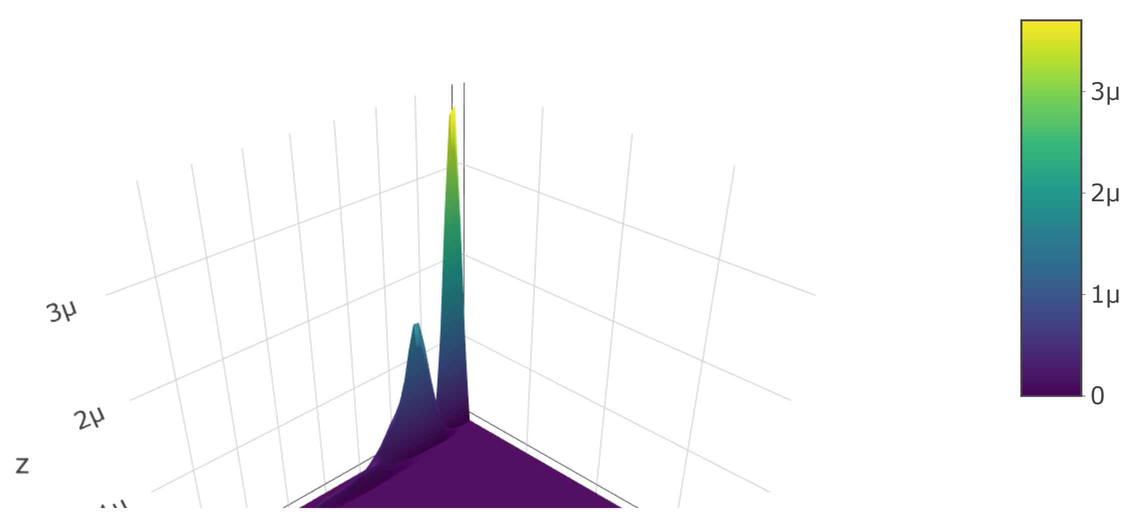

```
kd <- with(diamonds, MASS::kde2d(diamonds$table, diamonds$price, n = 50))
plot_ly(x = kd$x, y = kd$y, z = kd$z) %>% add_surface()
```

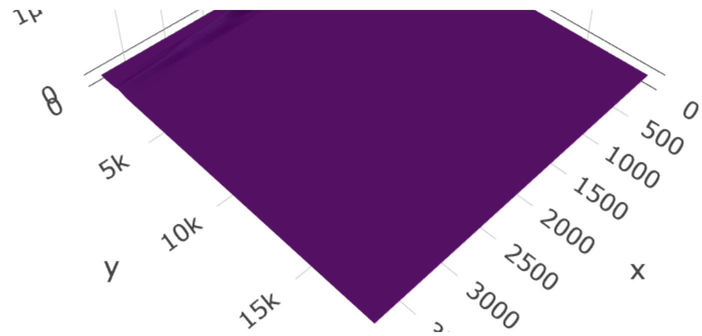



```
plot(diamonds$volume, diamonds$price, main = "Volume Related to Price", xlab="Volume",ylab="Pric
e")
```




```
kd <- with(diamonds, MASS::kde2d(diamonds$volume, diamonds$price, n = 50))
plot_ly(x = kd$x, y = kd$y, z = kd$z) %>% add_surface()
```

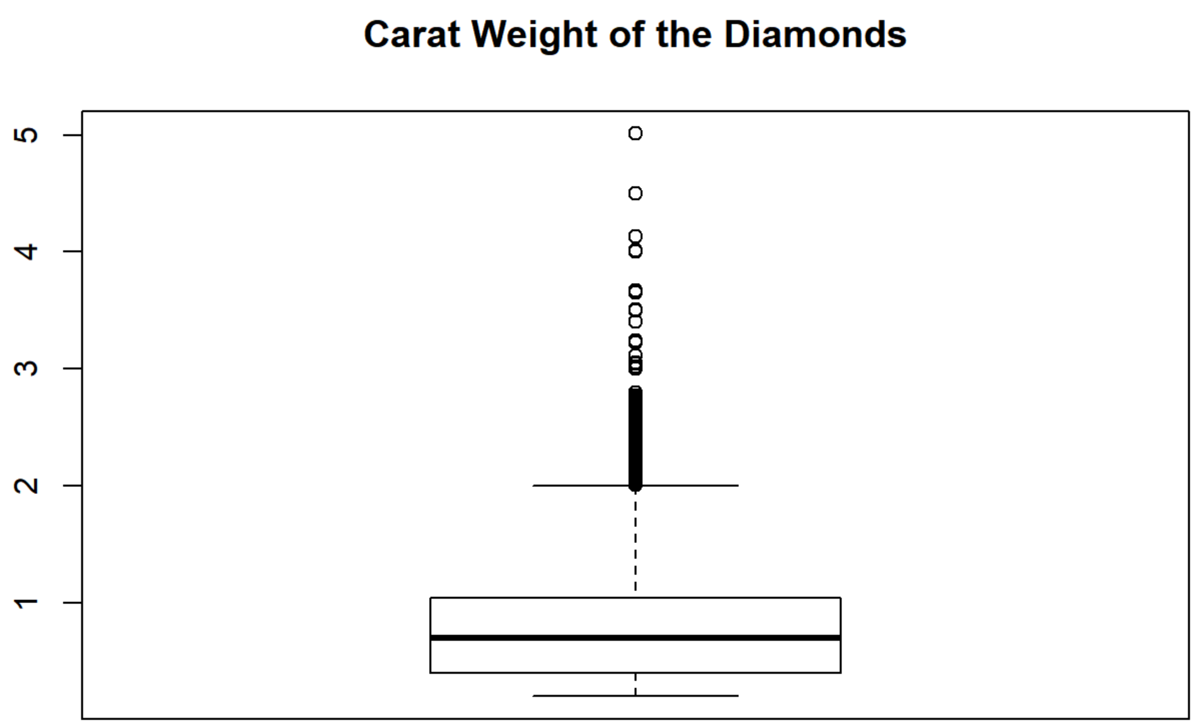




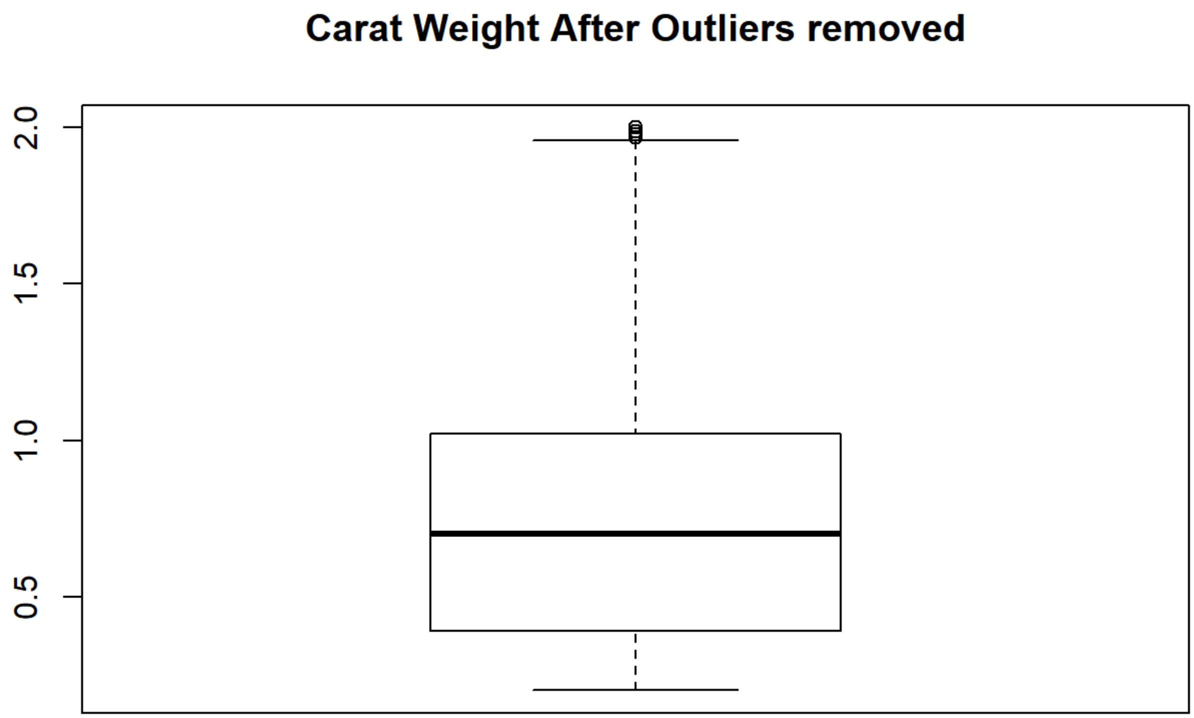
Correct for Outliers

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.

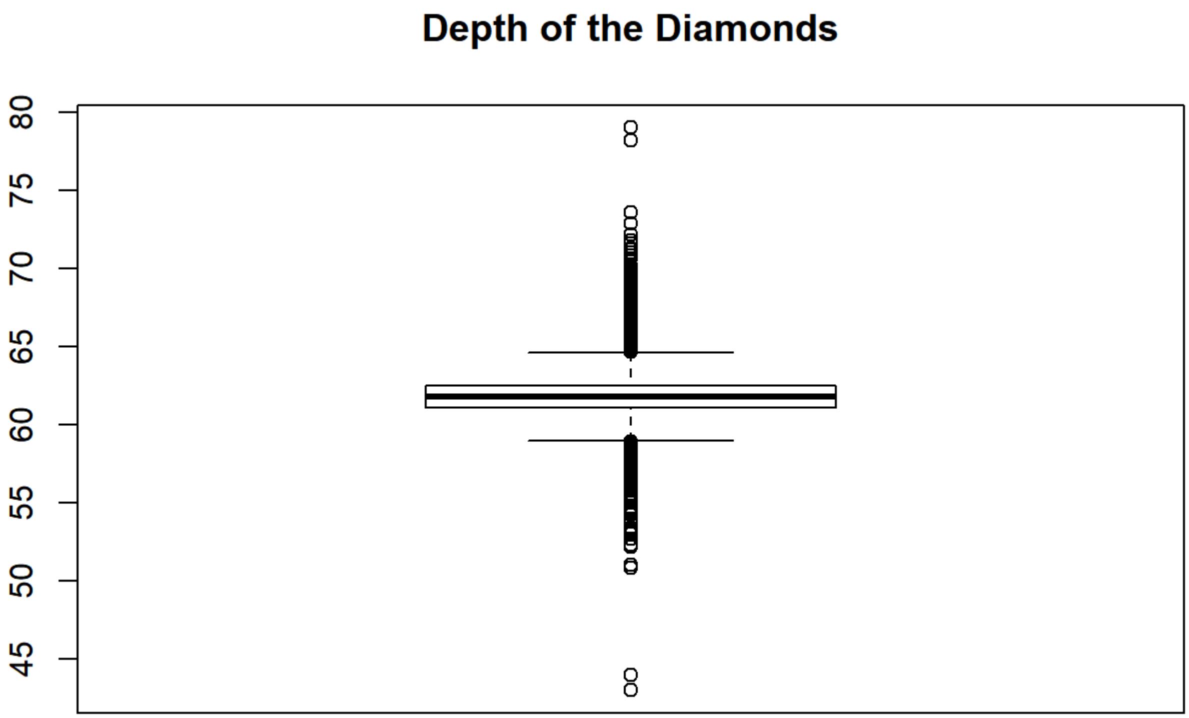
```
library(gridExtra)
outliers <- boxplot(diamonds$carat, main="Carat Weight of the Diamonds")$out
```



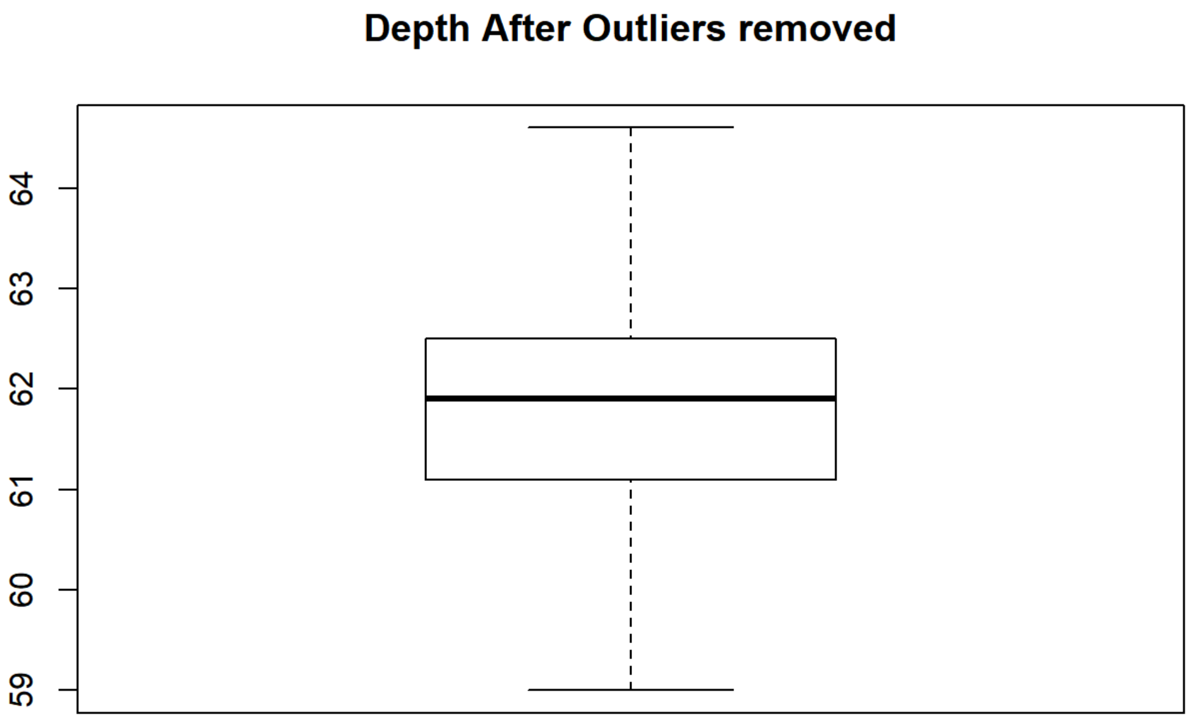

```
diamonds <- diamonds[-which(diamonds$carat %in% outliers),]
boxplot(diamonds$carat, main="Carat Weight After Outliers removed")
```



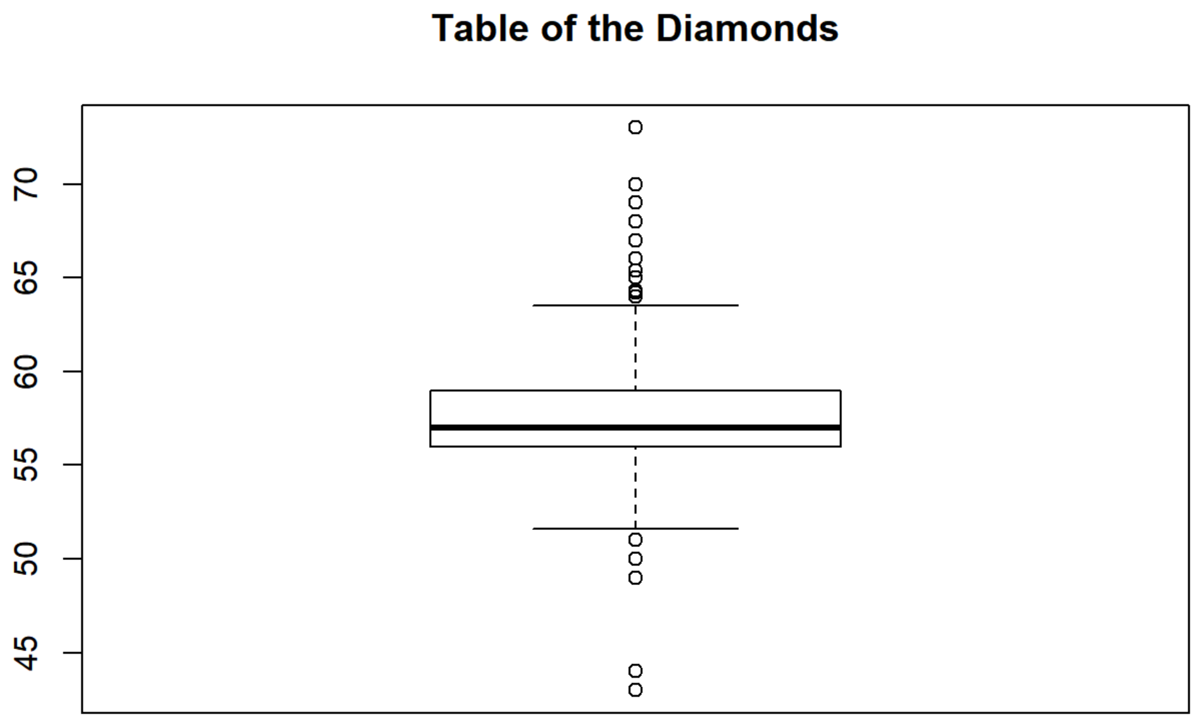

```
outliers <- boxplot(diamonds$depth, main="Depth of the Diamonds")$out
```



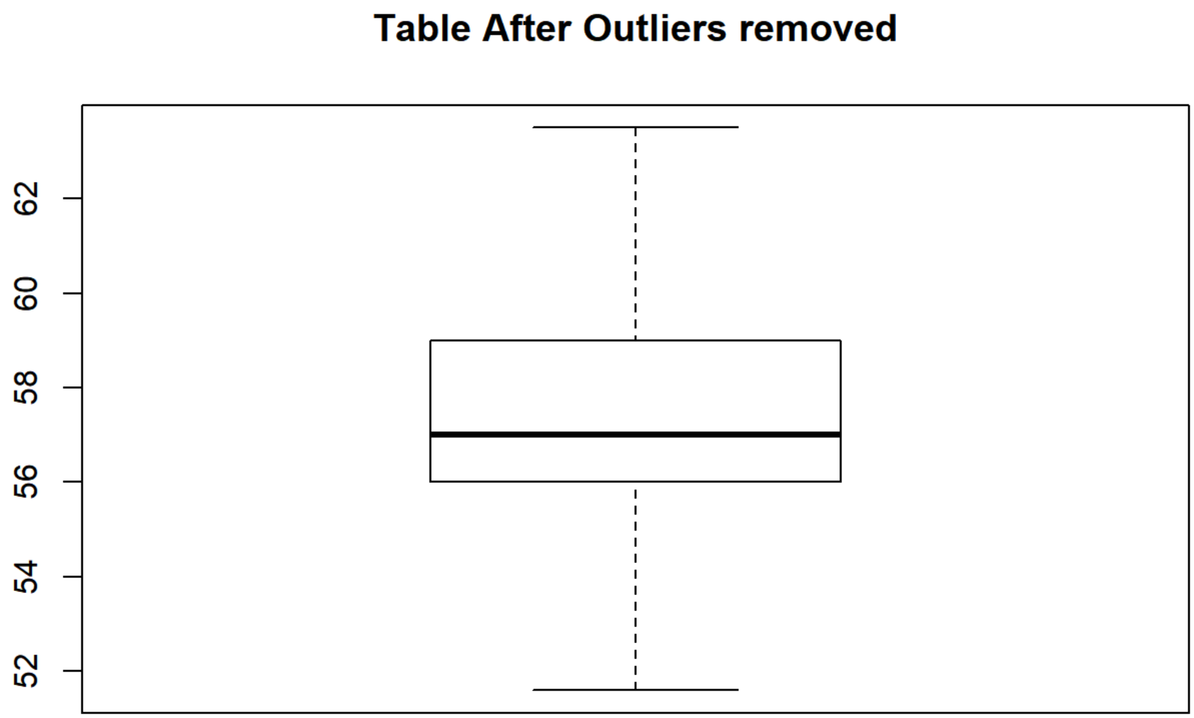

```r
diamonds <- diamonds[-which(diamonds$depth %in% outliers),]
boxplot(diamonds$depth, main="Depth After Outliers removed")
```



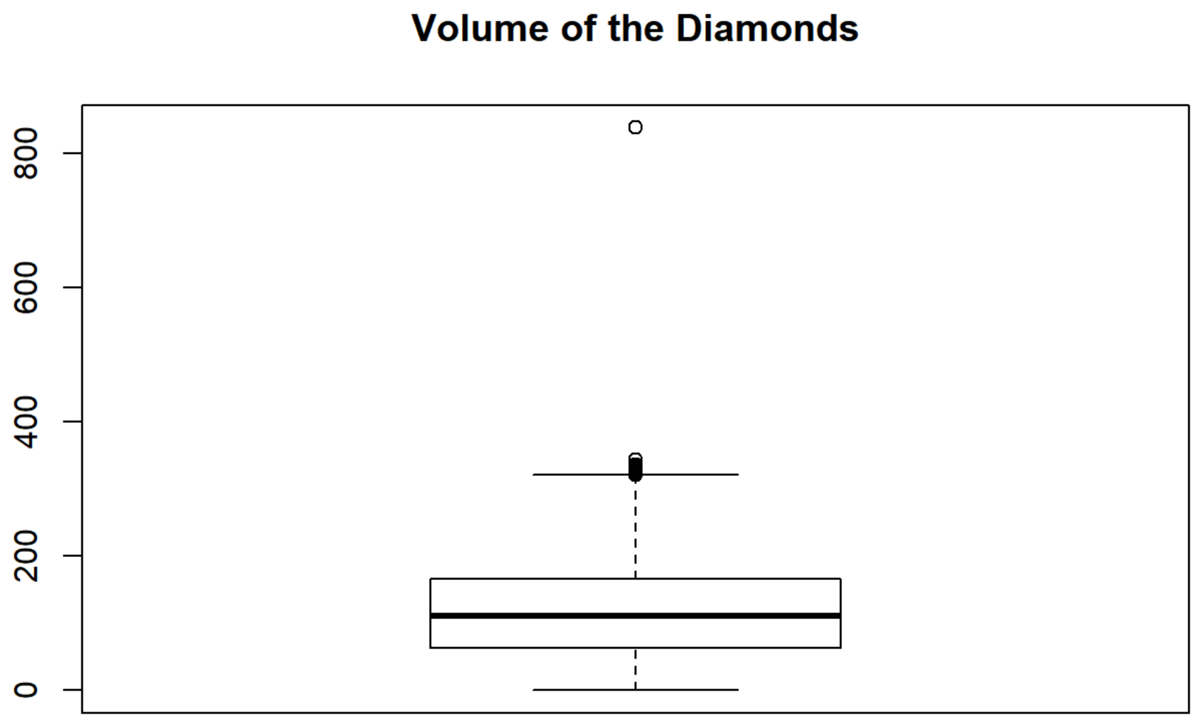

```
outliers <- boxplot(diamonds$table, main="Table of the Diamonds")$out
```



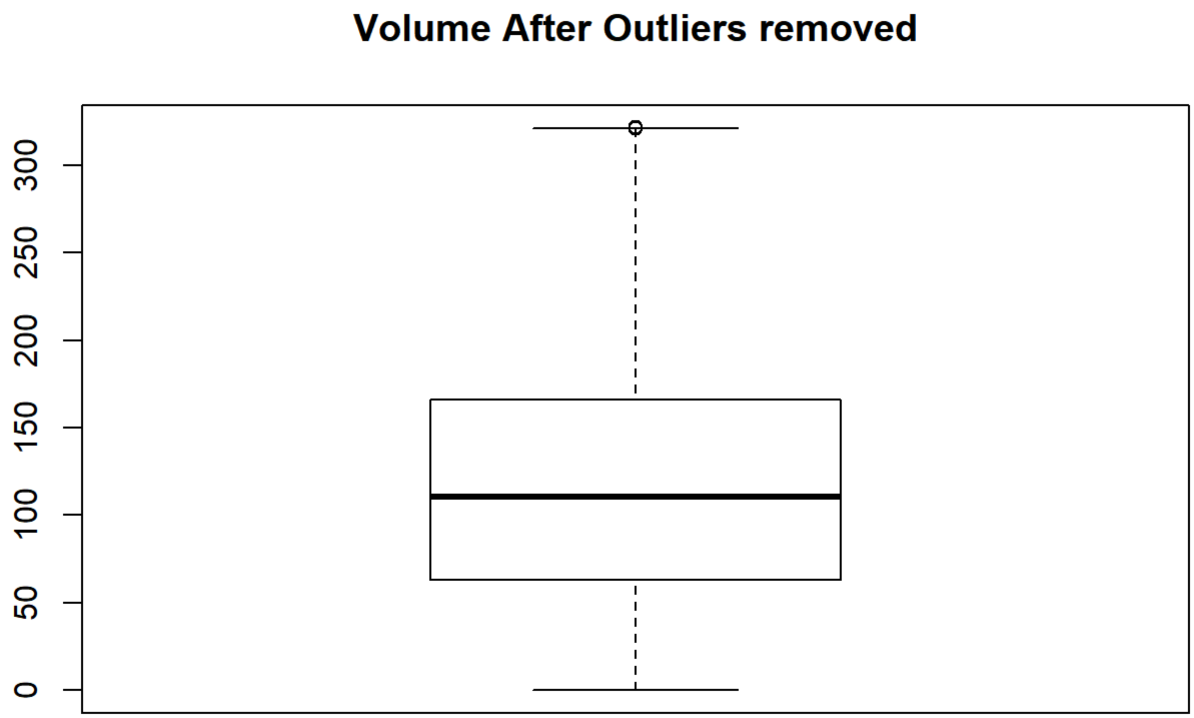

```
diamonds <- diamonds[-which(diamonds$table %in% outliers),]
boxplot(diamonds$table, main="Table After Outliers removed")
```




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



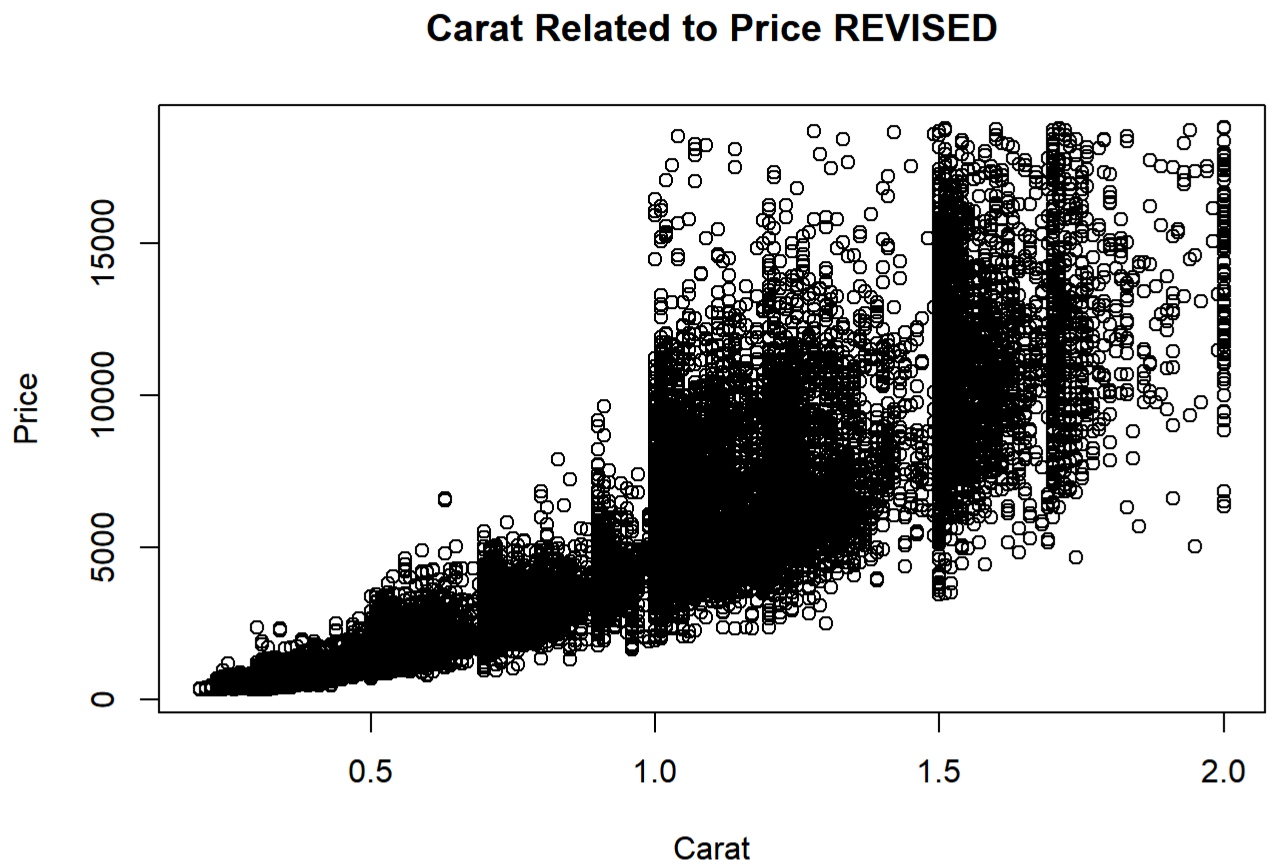

```
diamonds <- diamonds[-which(diamonds$volume %in% outliers),]
boxplot(diamonds$volume, main="Volume After Outliers removed")
```



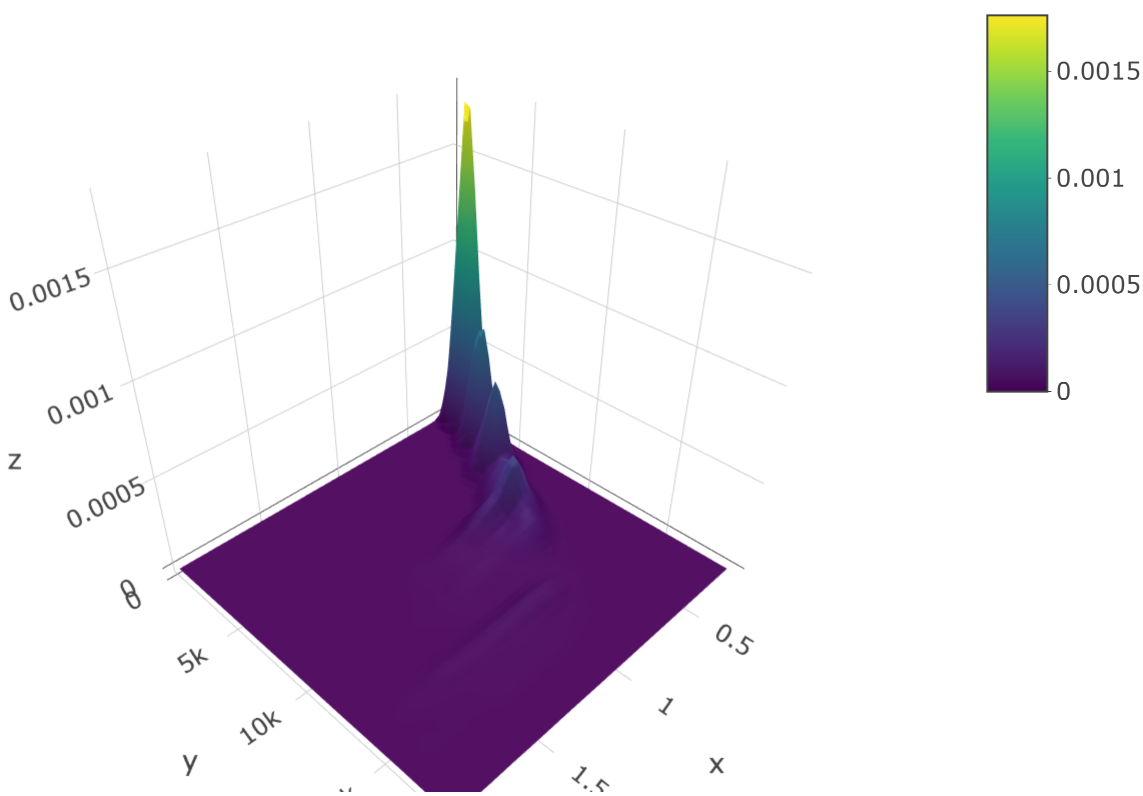
Plot Above Variables against Price

Now plotting against price will show any obvious relations/correlation between the variables. Then, a type of regression can be assigned. Note that these plots also come with corresponding 3d density plots same as before.

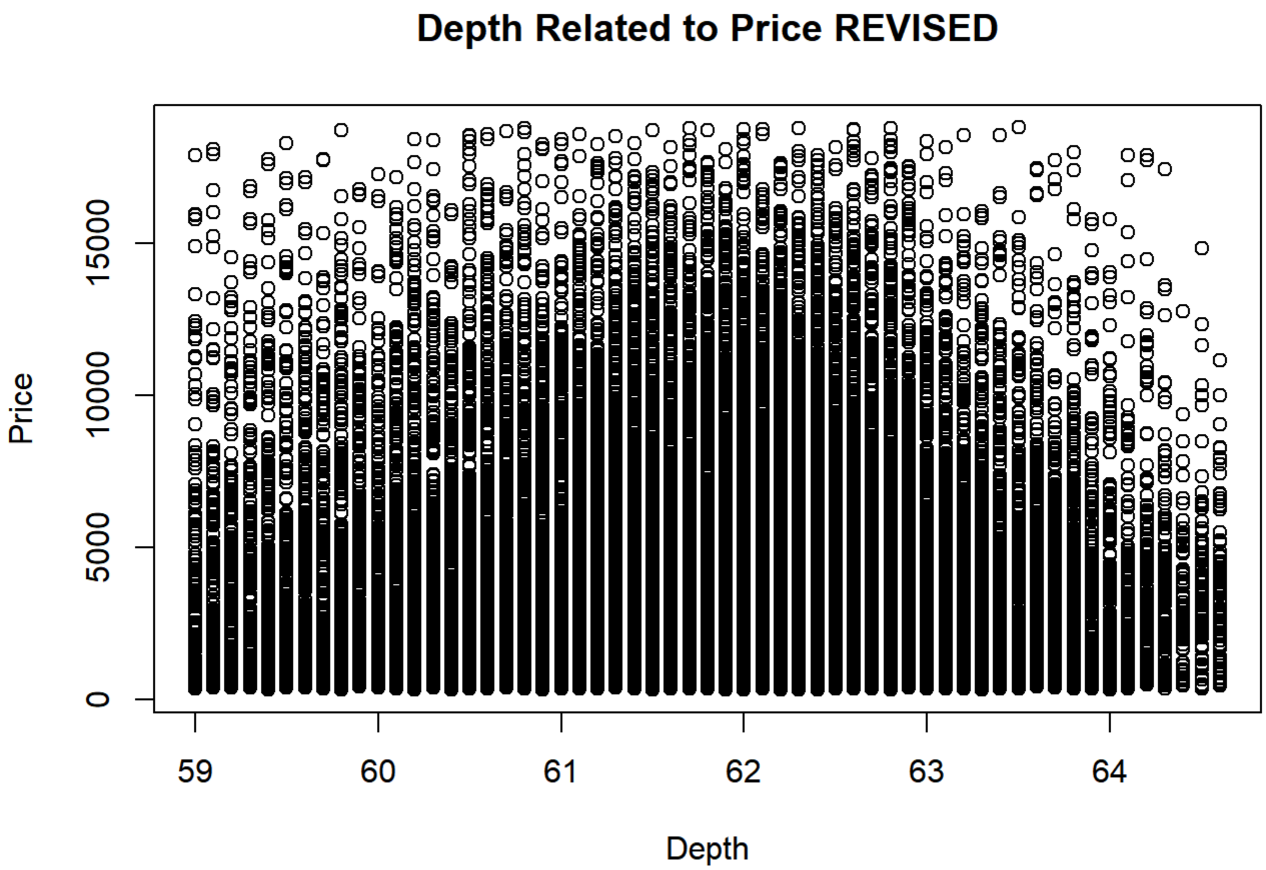
```
plot(diamonds$carat, diamonds$price, main = "Carat Related to Price REVISED", xlab="Carat",ylab=
"Price")
```



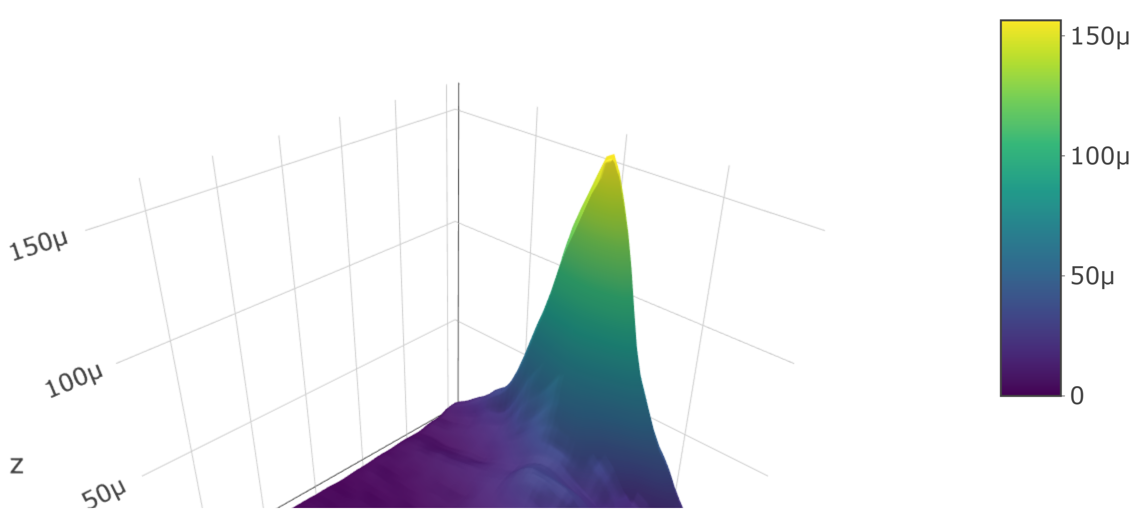

```
kd <- with(diamonds, MASS::kde2d(diamonds$carat, diamonds$price, n = 50))
plot_ly(x = kd$x, y = kd$y, z = kd$z) %>% add_surface()
```

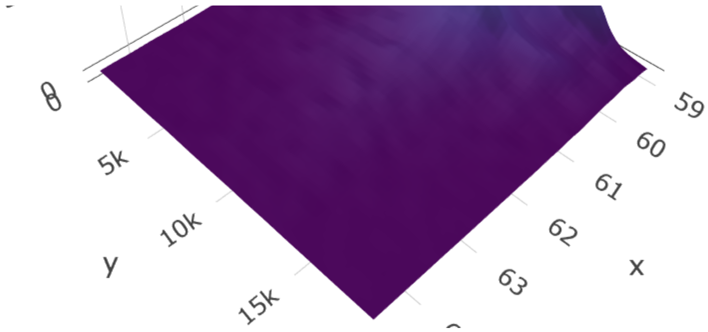



```
plot(diamonds$depth, diamonds$price, main = "Depth Related to Price REVISED", xlab="Depth",ylab=
"Price")
```

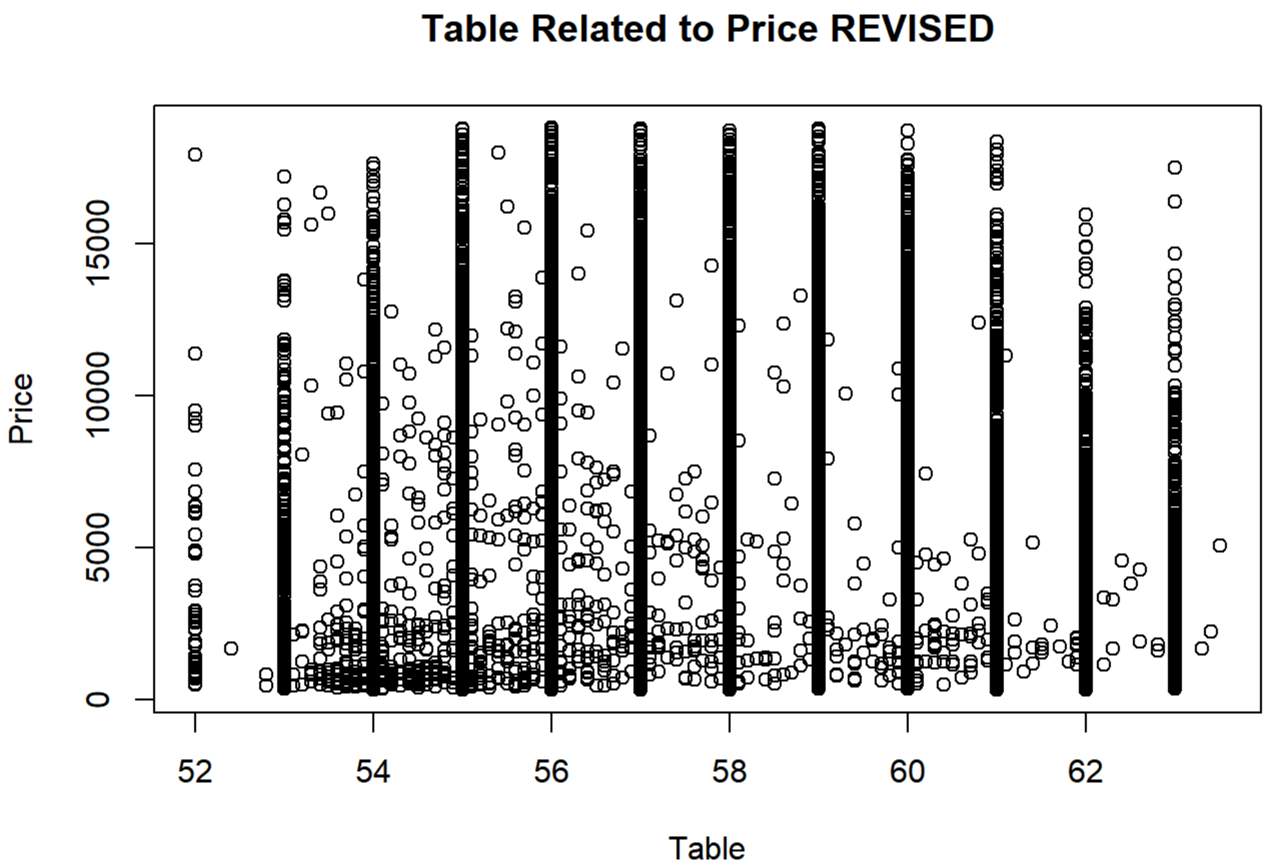



```
kd <- with(diamonds, MASS::kde2d(diamonds$depth, diamonds$price, n = 50))
plot_ly(x = kd$x, y = kd$y, z = kd$z) %>% add_surface()
```

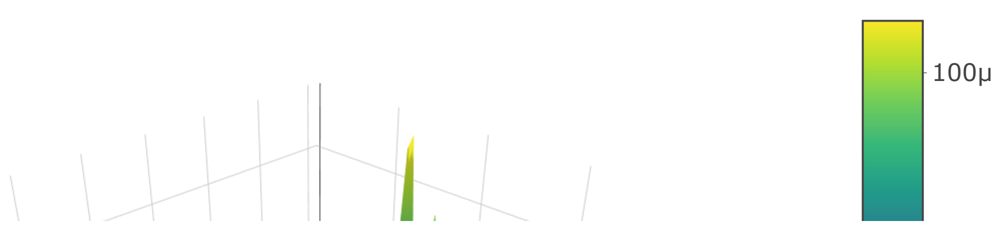


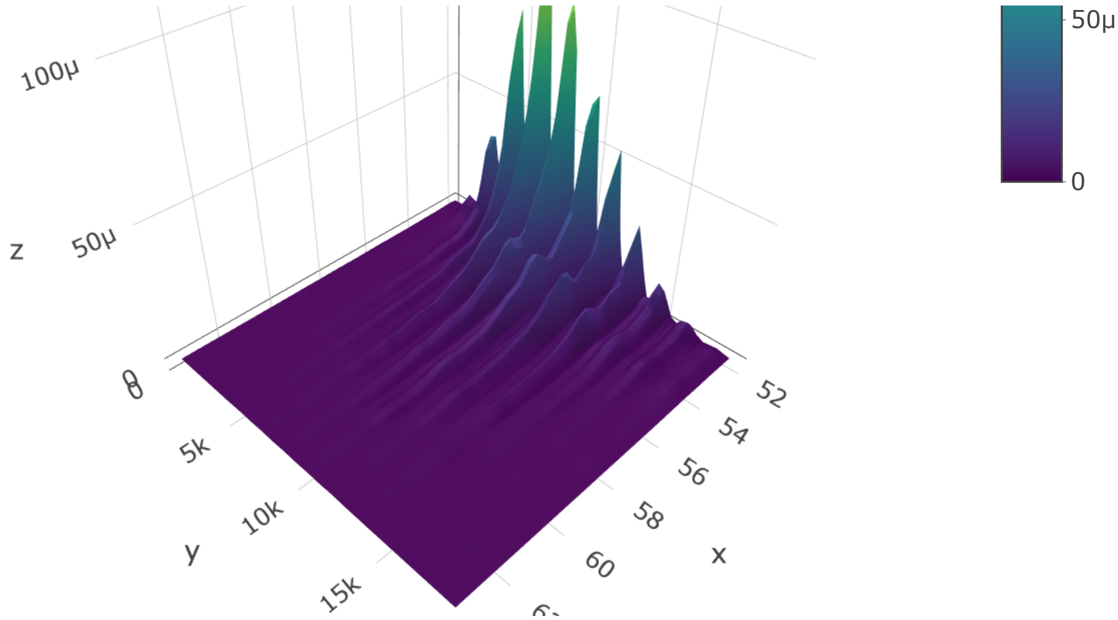



```
plot(diamonds$table, diamonds$price, main = "Table Related to Price REVISED", xlab="Table",ylab=
"Price")
```

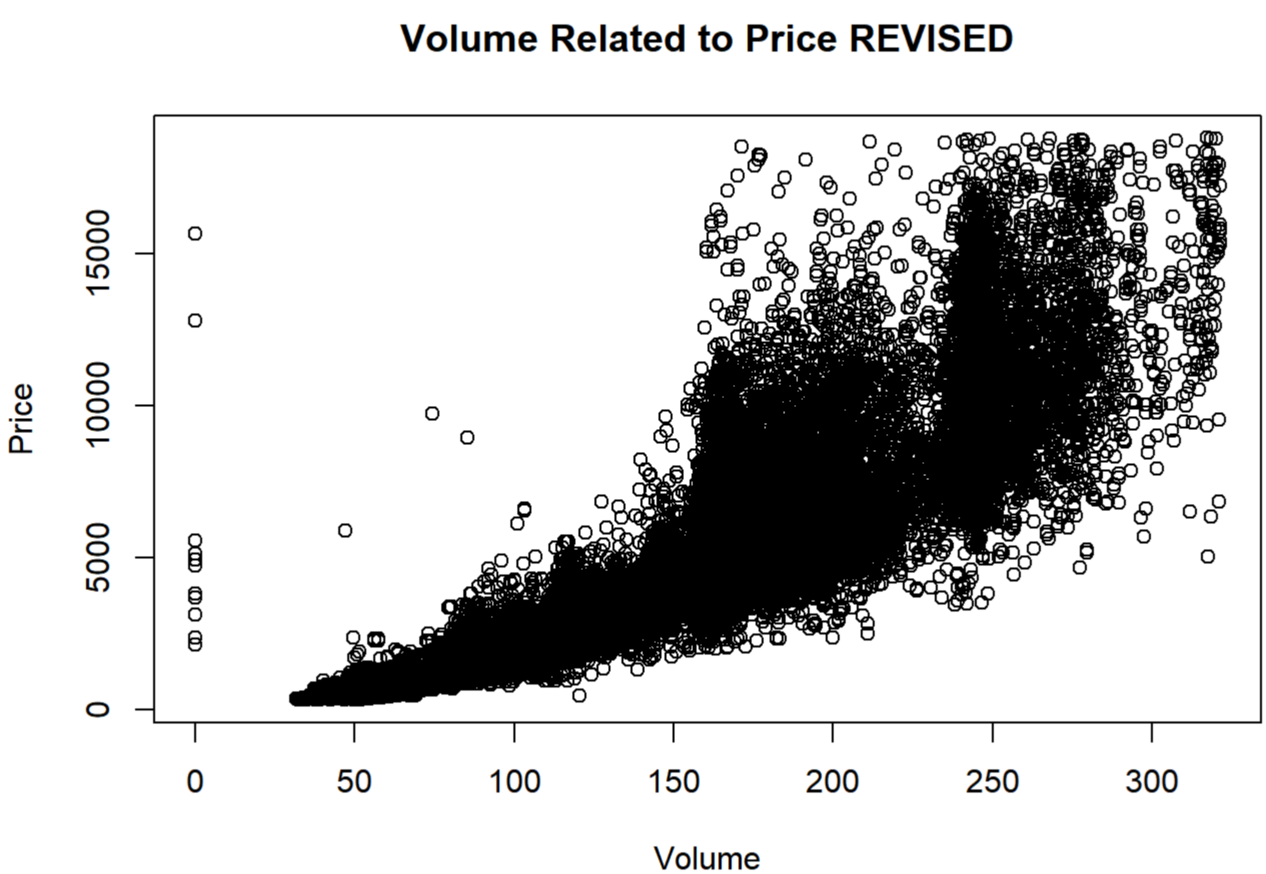



```
kd <- with(diamonds, MASS::kde2d(diamonds$table, diamonds$price, n = 50))
plot_ly(x = kd$x, y = kd$y, z = kd$z) %>% add_surface()
```

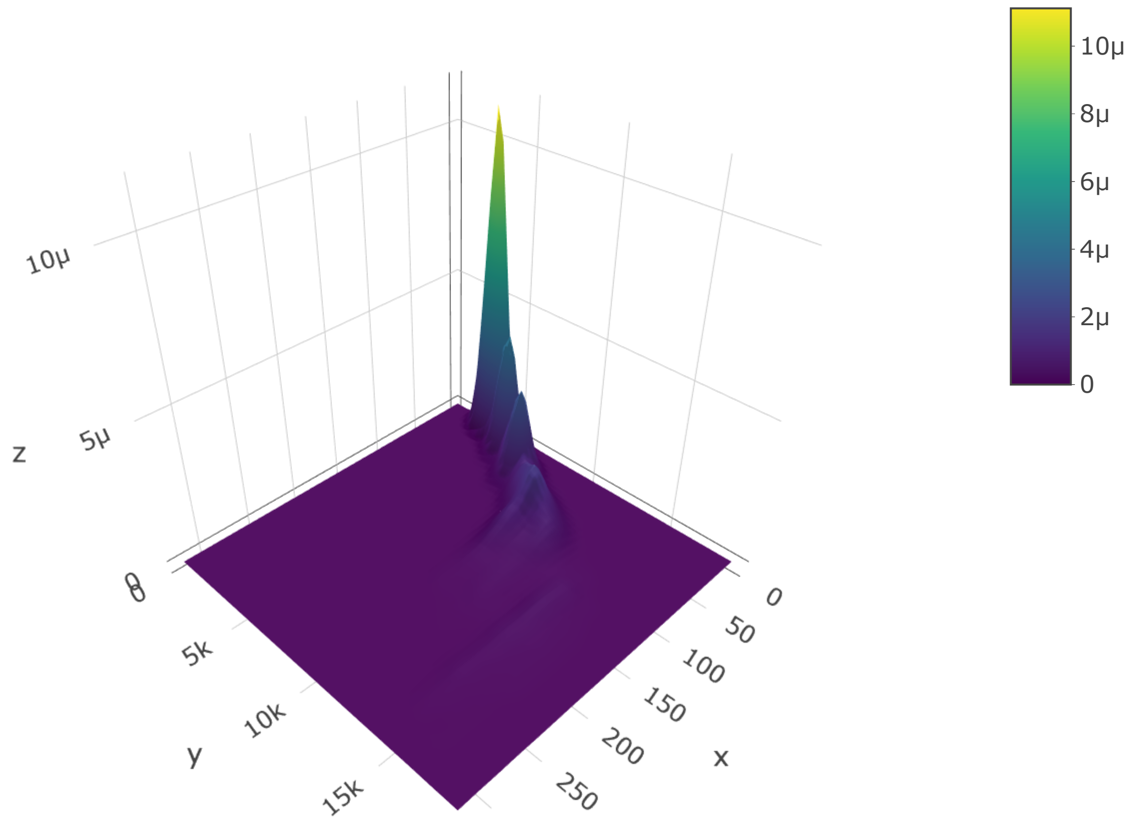





```
plot(diamonds$volume, diamonds$price, main = "Volume Related to Price REVISED", xlab="Volume",yl
ab="Price")
```




```
kd <- with(diamonds, MASS::kde2d(diamonds$volume, diamonds$price, n = 50))
plot_ly(x = kd$x, y = kd$y, z = kd$z) %>% add_surface()
```



Analysis

Now since the dataset has been primed, conducting simple linear regressions and determining relative correlations on the data set can help us get close to the main goal of this project.

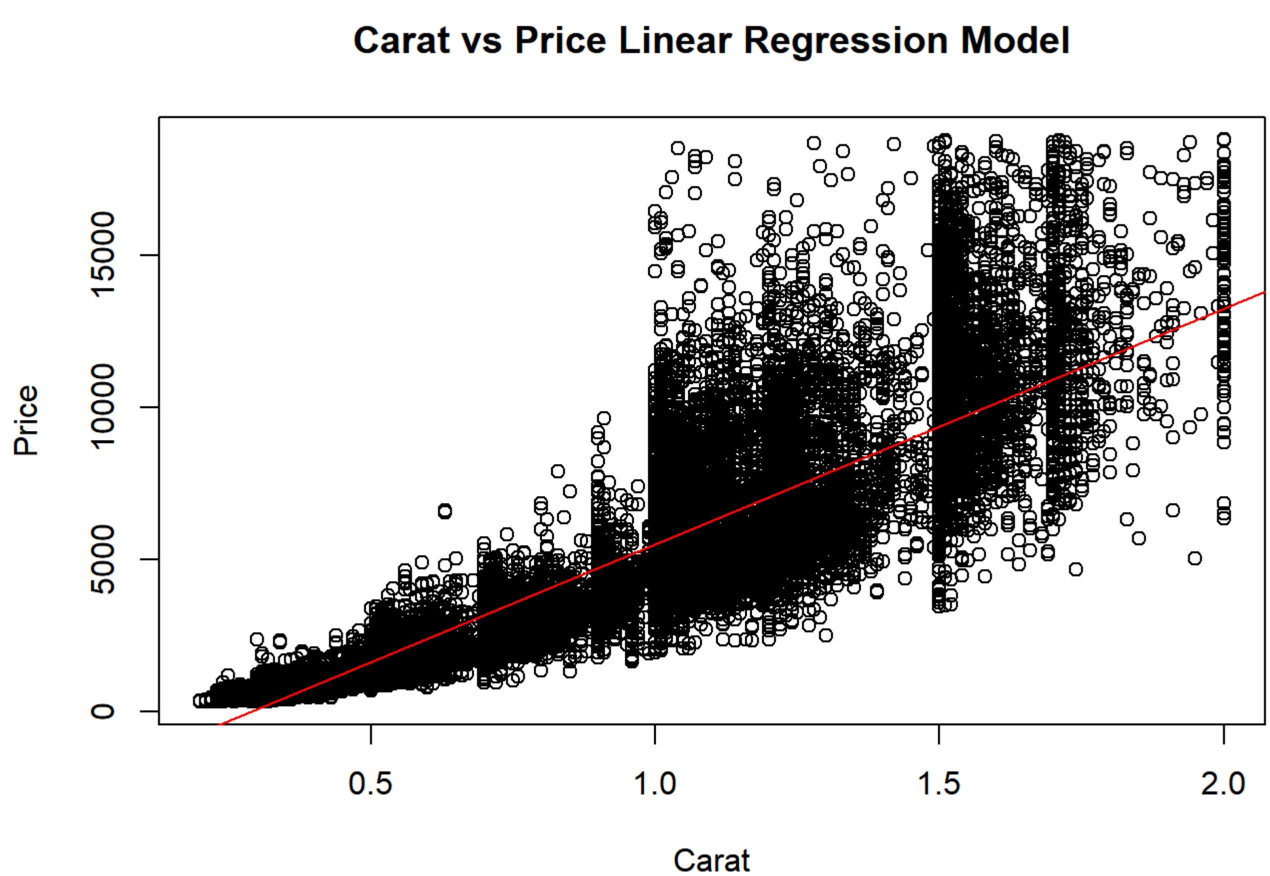
Linear Regression of Variables

The below r script will compute and print out the summary as well as the graph of each linear model formed by all the combinations of the four variables: carat, depth, table, and volume, with price

```
#CARAT vs PRICE
caratlm <- lm(diamonds$price ~ diamonds$carat)
summary(caratlm)
```

```
##
## Call:
## lm(formula = diamonds$price ~ diamonds$carat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7850.7  -766.0   -20.8   503.2 12703.2
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -2226.17      13.59  -163.8   <2e-16 ***
## diamonds$carat  7754.80      16.24   477.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1425 on 48802 degrees of freedom
## Multiple R-squared:  0.8238, Adjusted R-squared:  0.8238
## F-statistic: 2.281e+05 on 1 and 48802 DF,  p-value: < 2.2e-16
```

```
plot(diamonds$price ~ diamonds$carat, main = "Carat vs Price Linear Regression Model", xlab="Car
at",ylab="Price")
abline(caratlm, col="red")
```



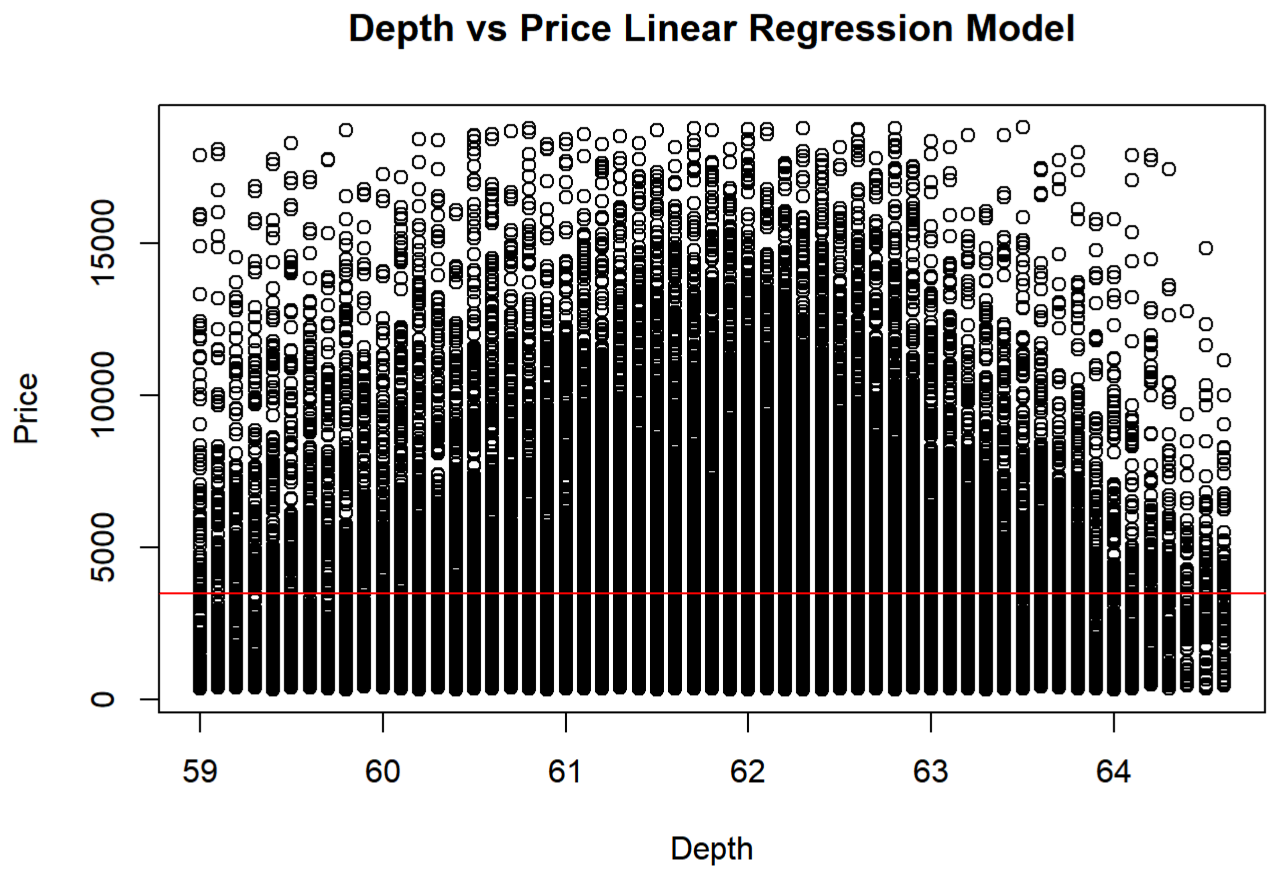

```
cat("EQUATION of the Line: y=", caratlm$coefficients[1], "+", caratlm$coefficients[2], "x")
```

```
## EQUATION of the Line: y= -2226.166 + 7754.804 x
```

```
#DEPTH vs PRICE
depthlm <- lm(diamonds$price ~ diamonds$depth)
summary(depthlm)
```

```
##
## Call:
## lm(formula = diamonds$price ~ diamonds$depth)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -3162  -2581  -1286   1452  15332
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3521.4175   880.5532   3.999 6.37e-05 ***
## diamonds$depth   -0.5524    14.2462  -0.039    0.969
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3394 on 48802 degrees of freedom
## Multiple R-squared:  3.081e-08,  Adjusted R-squared:  -2.046e-05
## F-statistic: 0.001504 on 1 and 48802 DF,  p-value: 0.9691
```

```
plot(diamonds$price ~ diamonds$depth, main = "Depth vs Price Linear Regression Model", xlab="Dep
th",ylab="Price")
abline(depthlm, col="red")
```



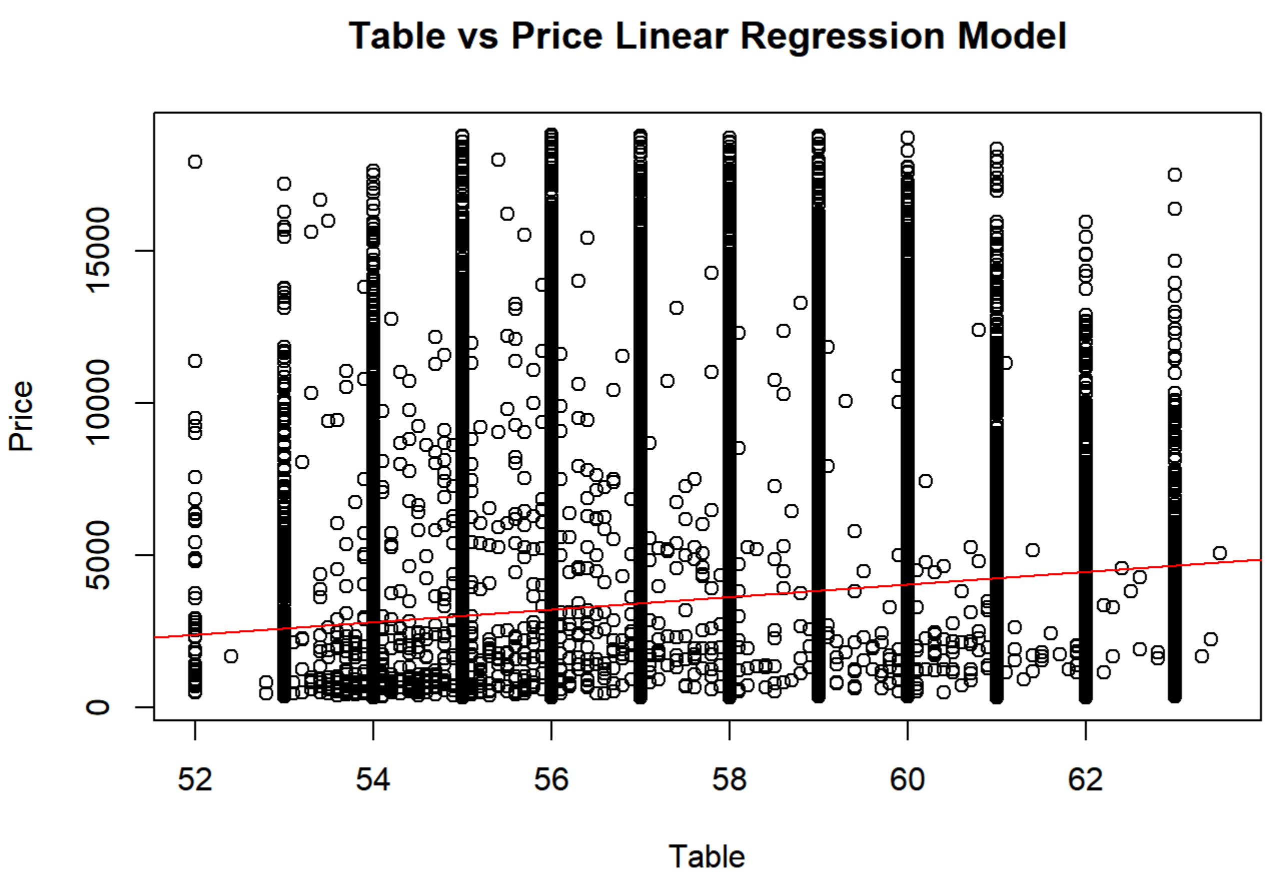

```
cat("EQUATION of the Line: y=", depthlm$coefficients[1], "+", depthlm$coefficients[2], "x")
```

```
## EQUATION of the Line: y= 3521.418 + -0.5524095 x
```

```
#TABLE vs PRICE
tablelm <- lm(diamonds$price ~ diamonds$table)
summary(tablelm)
```

```
##
## Call:
## lm(formula = diamonds$price ~ diamonds$table)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -4311  -2398  -1252   1384  15781
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -8271.706    434.541  -19.04   <2e-16 ***
## diamonds$table   205.389      7.585   27.08   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3369 on 48802 degrees of freedom
## Multiple R-squared:  0.0148, Adjusted R-squared:  0.01478
## F-statistic: 733.2 on 1 and 48802 DF,  p-value: < 2.2e-16
```

```
plot(diamonds$price ~ diamonds$table, main = "Table vs Price Linear Regression Model", xlab="Tab
le",ylab="Price")
abline(tablelm, col="red")
```



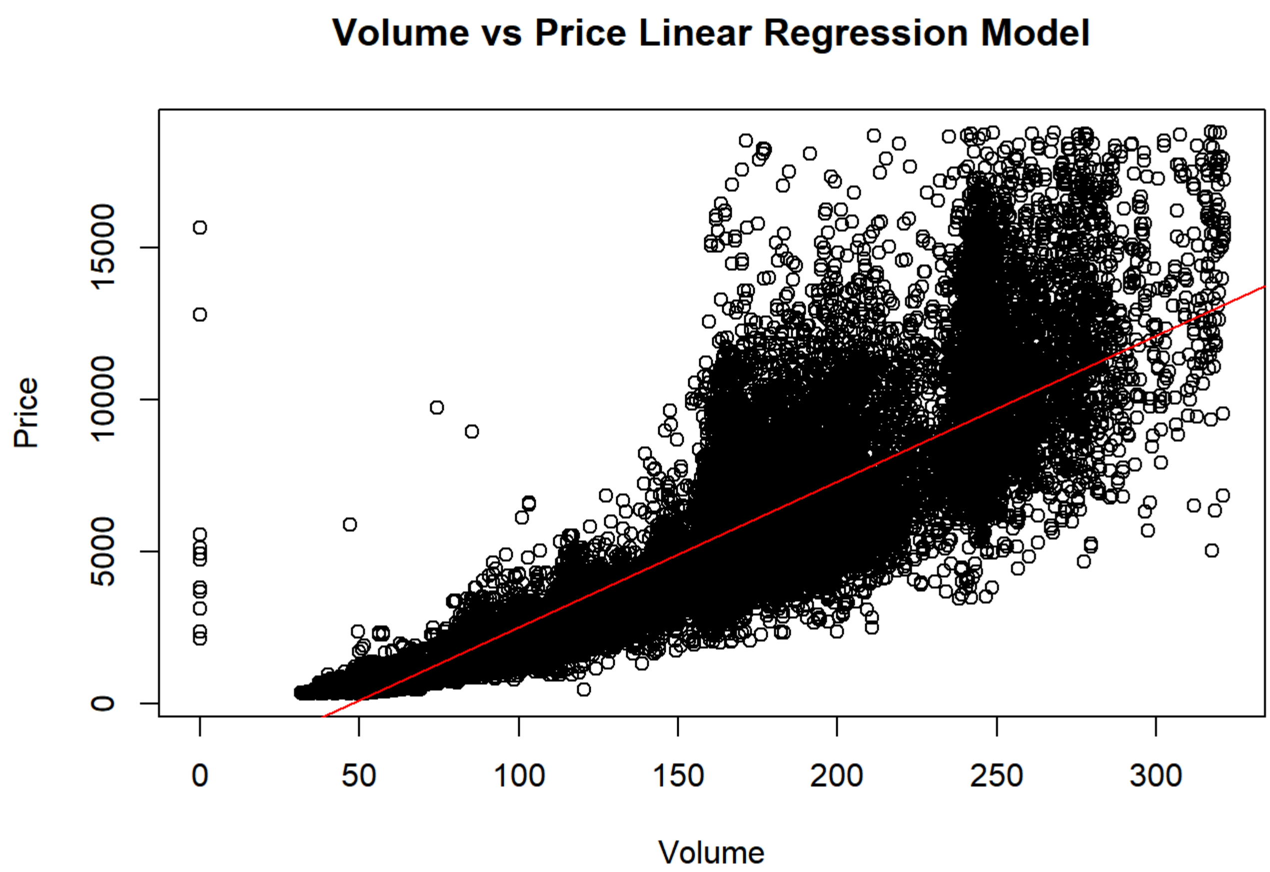

```
cat("EQUATION of the Line: y=", tablelm$coefficients[1], "+", tablelm$coefficients[2], "x")
```

```
## EQUATION of the Line: y= -8271.706 + 205.3894 x
```

```
#VOLUME vs PRICE
volumelm <- lm(diamonds$price ~ diamonds$volume)
summary(volumelm)
```

```
##
## Call:
## lm(formula = diamonds$price ~ diamonds$volume)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7931.0  -764.1   -25.2   501.6 17962.3
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -2276.2776    13.6316  -167.0   <2e-16 ***
## diamonds$volume    48.0314     0.1002   479.5   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1420 on 48802 degrees of freedom
## Multiple R-squared:  0.8249, Adjusted R-squared:  0.8249
## F-statistic: 2.299e+05 on 1 and 48802 DF,  p-value: < 2.2e-16
```

```
plot(diamonds$price ~ diamonds$volume, main = "Volume vs Price Linear Regression Model", xlab="V
olume",ylab="Price")
abline(volumelm, col="red")
```




```
cat("EQUATION of the Line: y=", volumelm$coefficients[1], "+", volumelm$coefficients[2], "x")
```

```
## EQUATION of the Line: y= -2276.278 + 48.0314 x
```

As it is clear that the table and the depth variables seem to have a random spread of data along with the really high errors and very small r^2 values and that these variables do not contribute much to the price of the diamonds. So, these variables would be dropped and their linear regression models are not going to be used.

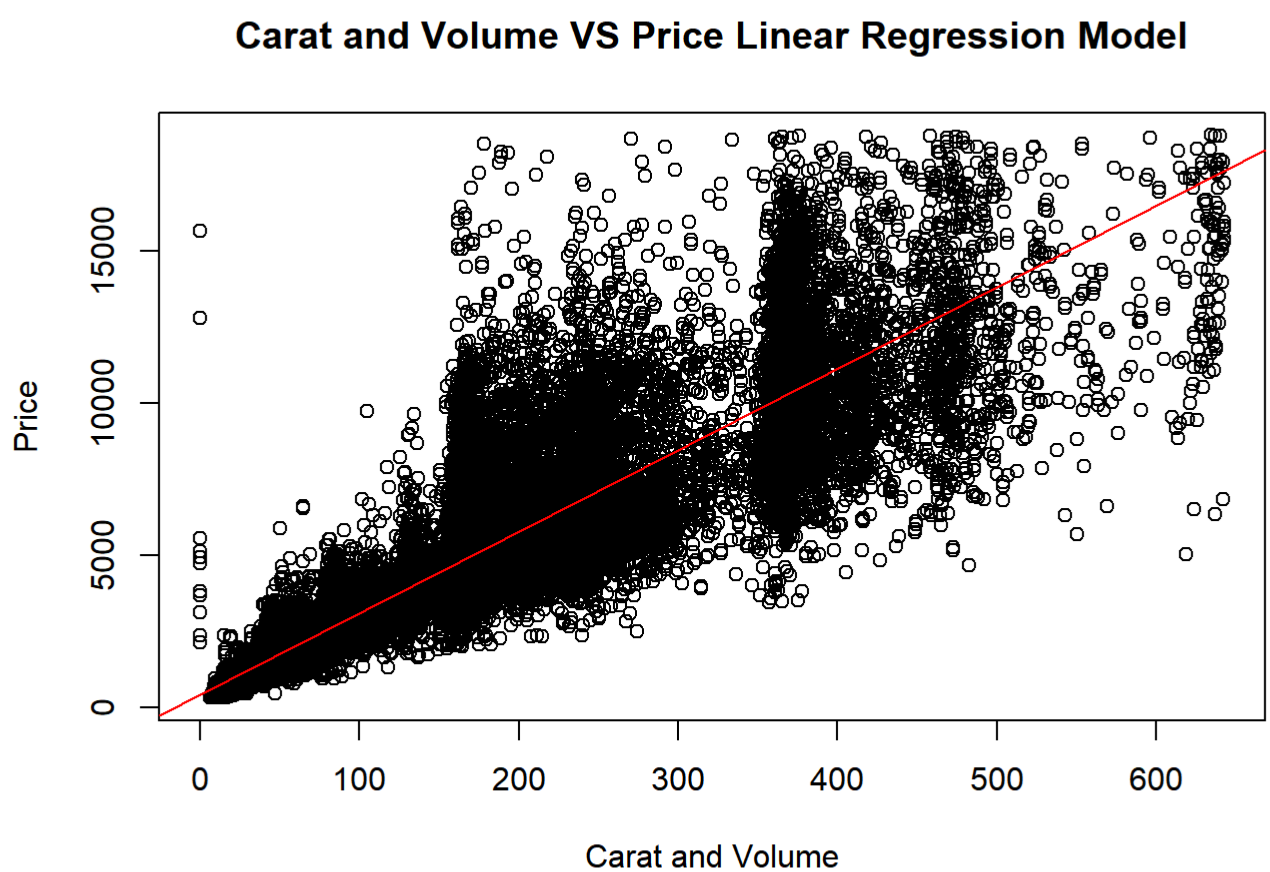
Linear Regression of a Combination of Variables

Now that only two NUMERICAL variables (carat and volume) are going to be used, it is possible to multiply these two variables together and be plotted along with their combined linear regression model.

```
cVmodel <- lm(diamonds$price ~ diamonds$carat * diamonds$volume)
x <- diamonds$carat * diamonds$volume
summary(cVmodel)
```

```
##
## Call:
## lm(formula = diamonds$price ~ diamonds$carat * diamonds$volume)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10676.4   -472.9    -45.4    246.0  15613.5
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      -581.9286    25.8246 -22.534   <2e-16 ***
## diamonds$carat                    545.3612   243.0185   2.244   0.0248 *
## diamonds$volume                    14.0462     1.5139   9.278   <2e-16 ***
## diamonds$carat:diamonds$volume     17.4083     0.2312  75.287   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1343 on 48800 degrees of freedom
## Multiple R-squared:  0.8434, Adjusted R-squared:  0.8434
## F-statistic: 8.764e+04 on 3 and 48800 DF,  p-value: < 2.2e-16
```

```
plot(diamonds$price ~ x, main = "Carat and Volume VS Price Linear Regression Model", xlab="Carat
and Volume",ylab="Price")
abline(lm(diamonds$price ~ x), col="red") # just for visual purposes
```



```
cat("EQUATION of the Line: y =", cVmodel$coefficients[1], "+", cVmodel$coefficients[2], "c +", c
Vmodel$coefficients[3], "v +", cVmodel$coefficients[4], "cv", "\nwhere c = carat and v = volume"
)
```

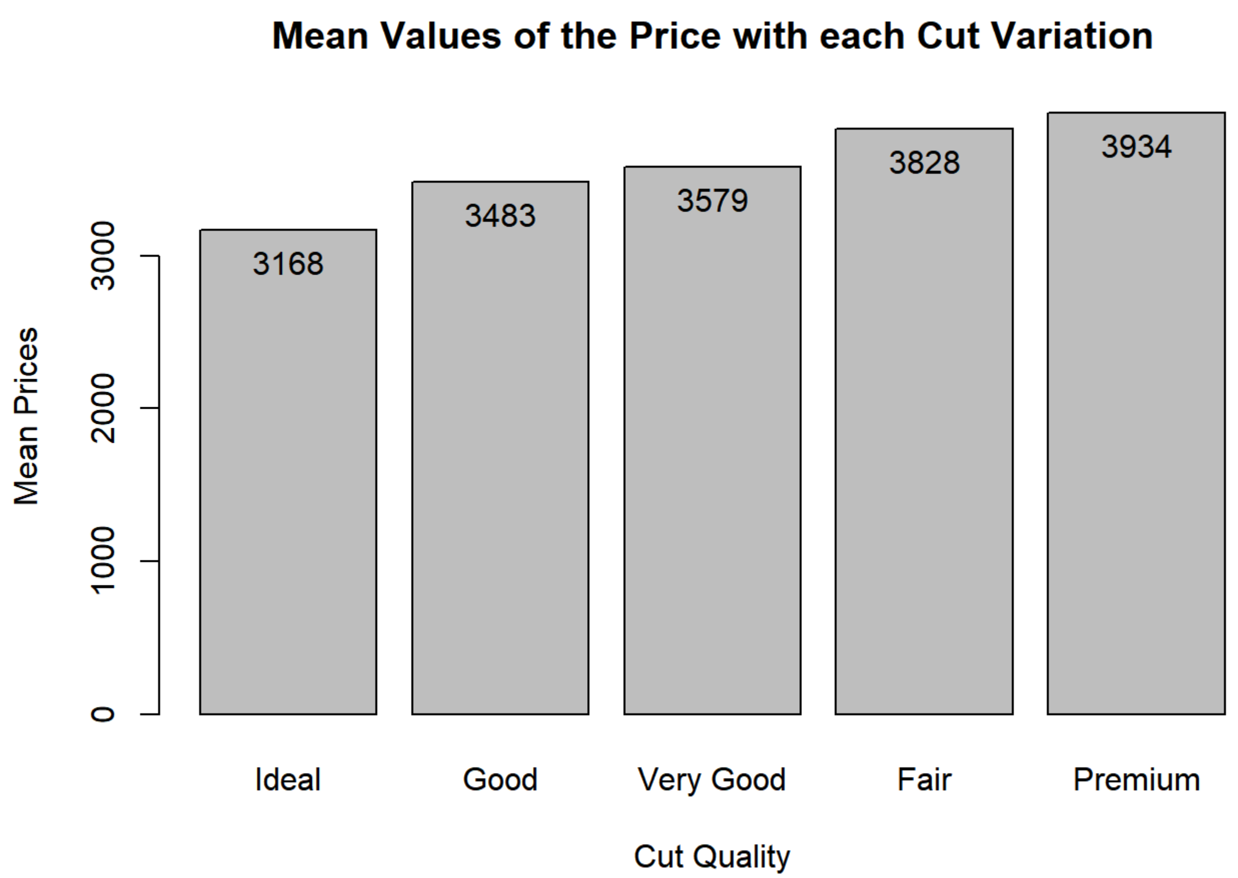
```
## EQUATION of the Line: y = -581.9286 + 545.3612 c + 14.04618 v + 17.40833 cv
## where c = carat and v = volume
```

The highest R^2 achieved in the previous plots was 0.8249 but with this model, we achieved a higher R^2 as 0.8434. This is some progress but including more variables would help with the model and increase the R^2 value.

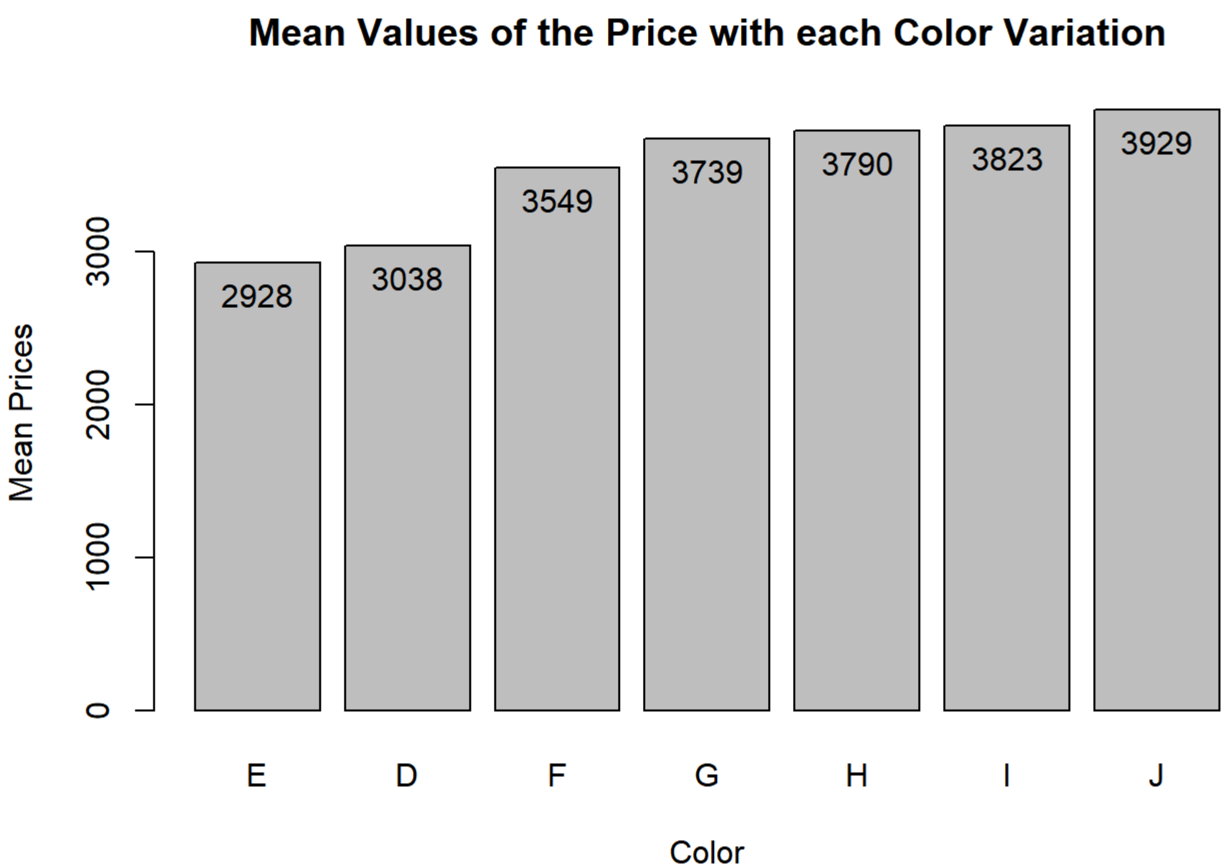
Qualitative Correlation

Of course, the data variables that are non-numerical should not be ruled out as one that does not make a big different in price. What the below R code does is calcuate the average of each non-numerical value against price and plots it in a bar graph

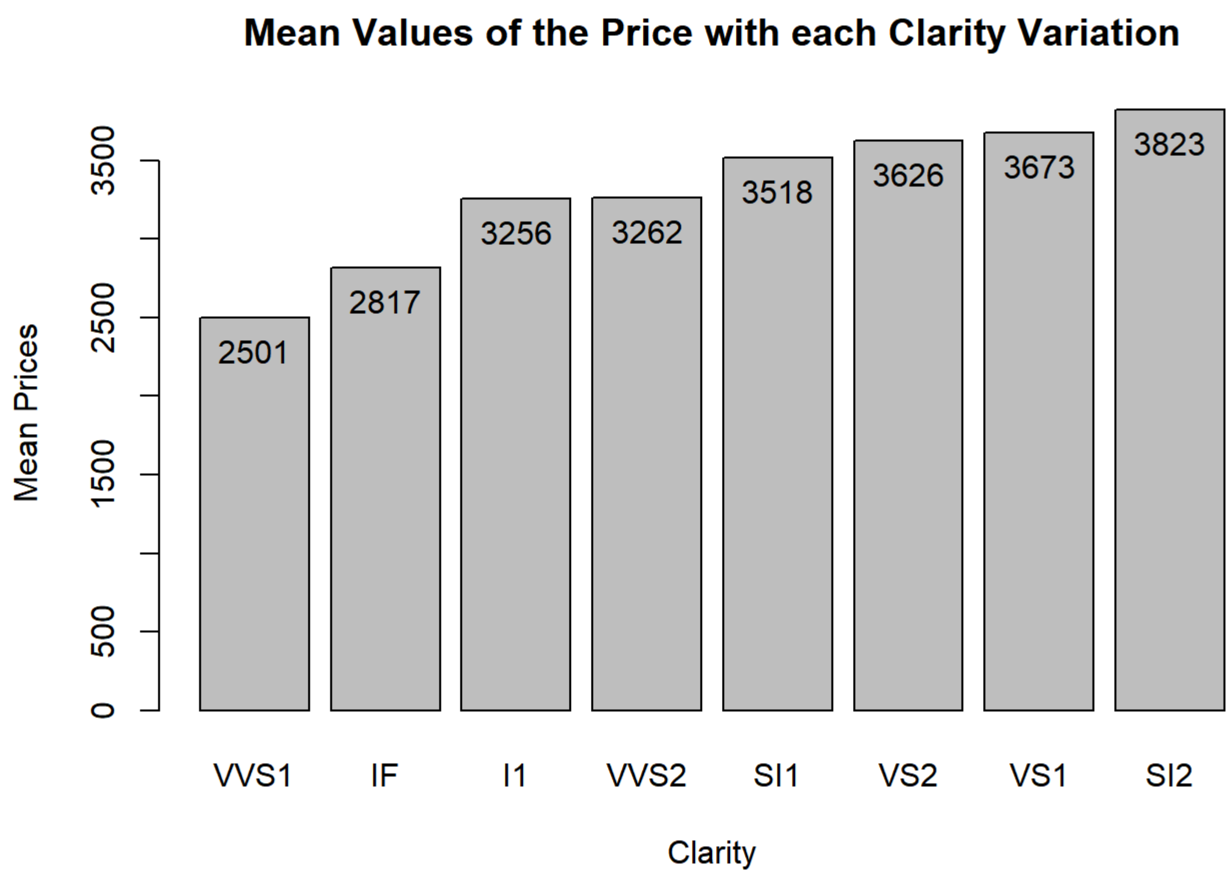
```
means_cut <- sort(with(diamonds, tapply(price, cut, mean)))
x<-barplot(means_cut,main="Mean Values of the Price with each Cut Variation", xlab="Cut Quality"
, ylab="Mean Prices")
text(x,means_cut-210,labels=as.character(floor(means_cut)))
```



```
means_color <- sort(with(diamonds, tapply(price, color, mean)))
x<-barplot(means_color,main="Mean Values of the Price with each Color Variation", xlab="Color",
 ylab="Mean Prices")
text(x,means_color-210,labels=as.character(floor(means_color)))
```




```
means_clarity <- sort(with(diamonds, tapply(price, clarity, mean)))
x<-barplot(means_clarity,main="Mean Values of the Price with each Clarity Variation", xlab="Clar
ity", ylab="Mean Prices")
text(x,means_clarity-210,labels=as.character(floor(means_clarity)))
```



Since we see a clear increase in the average price for each different value in the non-numerical value.

Ratio Calculation of Non-numerical variables

Since we know that there is an effect of these variables on price, the ratio of each value against the lowest one can be caluculated and inputed within the current linear model of carat and volume. These ratios also be inputed in new columns in the diamonds data frame.

```
means_cut <- means_cut/means_cut[1]
means_color <- means_color/means_color[1]
means_clarity <- means_clarity/means_clarity[1]

diamonds$means_cut = means_cut[diamonds$cut]
diamonds$means_color = means_color[diamonds$color]
diamonds$means_clarity = means_clarity[diamonds$clarity]
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
## 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
## 7 7  0.24 Very Good     I    VVS1  62.3    57   336 3.95 3.98 2.47  57 38.83087
##   means_cut means_color means_clarity
## 1  1.129814    1.037607      1.304503
## 2  1.208304    1.037607      1.302008
## 4  1.208304    1.305763      1.450031
## 5  1.099485    1.341794      1.304503
## 6  1.241950    1.341794      1.528706
## 7  1.241950    1.305763      1.468815
```

Final Model and Conclusion

Now that there are ratios for the non-numerical variables, it is possible to add it to our linear model of carat and volume to form a stronger linear model.

```
finalModel <- lm(diamonds$price ~ diamonds$carat * diamonds$volume * diamonds$means_cut * diamon
ds$means_color * diamonds$means_clarity)
x <- diamonds$carat * diamonds$volume * diamonds$means_cut * diamonds$means_color * diamonds$mea
ns_clarity
summary(finalModel)
```

```
##
## Call:
## lm(formula = diamonds$price ~ diamonds$carat * diamonds$volume *
##     diamonds$means_cut * diamonds$means_color * diamonds$means_clarity)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8833.4  -346.2   -46.6   202.6 14164.6
##
## Coefficients:
##
Estimate
## (Intercept)
281951.9
## diamonds$carat
7457334.8
## diamonds$volume
-52117.1
## diamonds$means_cut
-277805.0
## diamonds$means_color
-194137.0
## diamonds$means_clarity
-202416.9
## diamonds$carat:diamonds$volume
3442.0
## diamonds$carat:diamonds$means_cut
-6513373.0
## diamonds$volume:diamonds$means_cut
46169.8
## diamonds$carat:diamonds$means_color
-5753615.5
## diamonds$volume:diamonds$means_color
39908.3
## diamonds$means_cut:diamonds$means_color
195966.9
## diamonds$carat:diamonds$means_clarity
-5697295.9
## diamonds$volume:diamonds$means_clarity
39525.7
## diamonds$means_cut:diamonds$means_clarity
200113.6
## diamonds$means_color:diamonds$means_clarity
138181.4
## diamonds$carat:diamonds$volume:diamonds$means_cut
-3625.1
## diamonds$carat:diamonds$volume:diamonds$means_color
-2524.9
## diamonds$carat:diamonds$means_cut:diamonds$means_color
5023956.4
## diamonds$volume:diamonds$means_cut:diamonds$means_color
-35372.2
## diamonds$carat:diamonds$volume:diamonds$means_clarity
```

```
-2404.7
## diamonds$carat:diamonds$means_cut:diamonds$means_clarity
4963235.0
## diamonds$volume:diamonds$means_cut:diamonds$means_clarity
-34932.5
## diamonds$carat:diamonds$means_color:diamonds$means_clarity
4379166.7
## diamonds$volume:diamonds$means_color:diamonds$means_clarity
-30150.4
## diamonds$means_cut:diamonds$means_color:diamonds$means_clarity
-140397.9
## diamonds$carat:diamonds$volume:diamonds$means_cut:diamonds$means_color
2677.8
## diamonds$carat:diamonds$volume:diamonds$means_cut:diamonds$means_clarity
2609.1
## diamonds$carat:diamonds$volume:diamonds$means_color:diamonds$means_clarity
1766.8
## diamonds$carat:diamonds$means_cut:diamonds$means_color:diamonds$means_clarity
-3813189.0
## diamonds$volume:diamonds$means_cut:diamonds$means_color:diamonds$means_clarity
26662.9
## diamonds$carat:diamonds$volume:diamonds$means_cut:diamonds$means_color:diamonds$means_clarity
-1922.9
##
Std. Error
## (Intercept)
64122.6
## diamonds$carat
842021.9
## diamonds$volume
5290.6
## diamonds$means_cut
54680.6
## diamonds$means_color
53017.2
## diamonds$means_clarity
46341.9
## diamonds$carat:diamonds$volume
627.8
## diamonds$carat:diamonds$means_cut
710570.9
## diamonds$volume:diamonds$means_cut
4467.9
## diamonds$carat:diamonds$means_color
663874.2
## diamonds$volume:diamonds$means_color
4177.1
## diamonds$means_cut:diamonds$means_color
45241.3
## diamonds$carat:diamonds$means_clarity
625248.1
## diamonds$volume:diamonds$means_clarity
3921.1
## diamonds$means_cut:diamonds$means_clarity
```

```
39505.3
## diamonds$means_color:diamonds$means_clarity
38301.9
## diamonds$carat:diamonds$volume:diamonds$means_cut
535.0
## diamonds$carat:diamonds$volume:diamonds$means_color
508.4
## diamonds$carat:diamonds$means_cut:diamonds$means_color
560576.4
## diamonds$volume:diamonds$means_cut:diamonds$means_color
3529.6
## diamonds$carat:diamonds$volume:diamonds$means_clarity
463.8
## diamonds$carat:diamonds$means_cut:diamonds$means_clarity
526894.2
## diamonds$volume:diamonds$means_cut:diamonds$means_clarity
3306.9
## diamonds$carat:diamonds$means_color:diamonds$means_clarity
493054.7
## diamonds$volume:diamonds$means_color:diamonds$means_clarity
3096.3
## diamonds$means_cut:diamonds$means_color:diamonds$means_clarity
32673.1
## diamonds$carat:diamonds$volume:diamonds$means_cut:diamonds$means_color
433.3
## diamonds$carat:diamonds$volume:diamonds$means_cut:diamonds$means_clarity
395.2
## diamonds$carat:diamonds$volume:diamonds$means_color:diamonds$means_clarity
375.2
## diamonds$carat:diamonds$means_cut:diamonds$means_color:diamonds$means_clarity
415757.2
## diamonds$volume:diamonds$means_cut:diamonds$means_color:diamonds$means_clarity
2612.9
## diamonds$carat:diamonds$volume:diamonds$means_cut:diamonds$means_color:diamonds$means_clarity
319.7
##
t value
## (Intercept)
4.397
## diamonds$carat
8.856
## diamonds$volume
-9.851
## diamonds$means_cut
-5.081
## diamonds$means_color
-3.662
## diamonds$means_clarity
-4.368
## diamonds$carat:diamonds$volume
5.483
## diamonds$carat:diamonds$means_cut
-9.166
## diamonds$volume:diamonds$means_cut
```

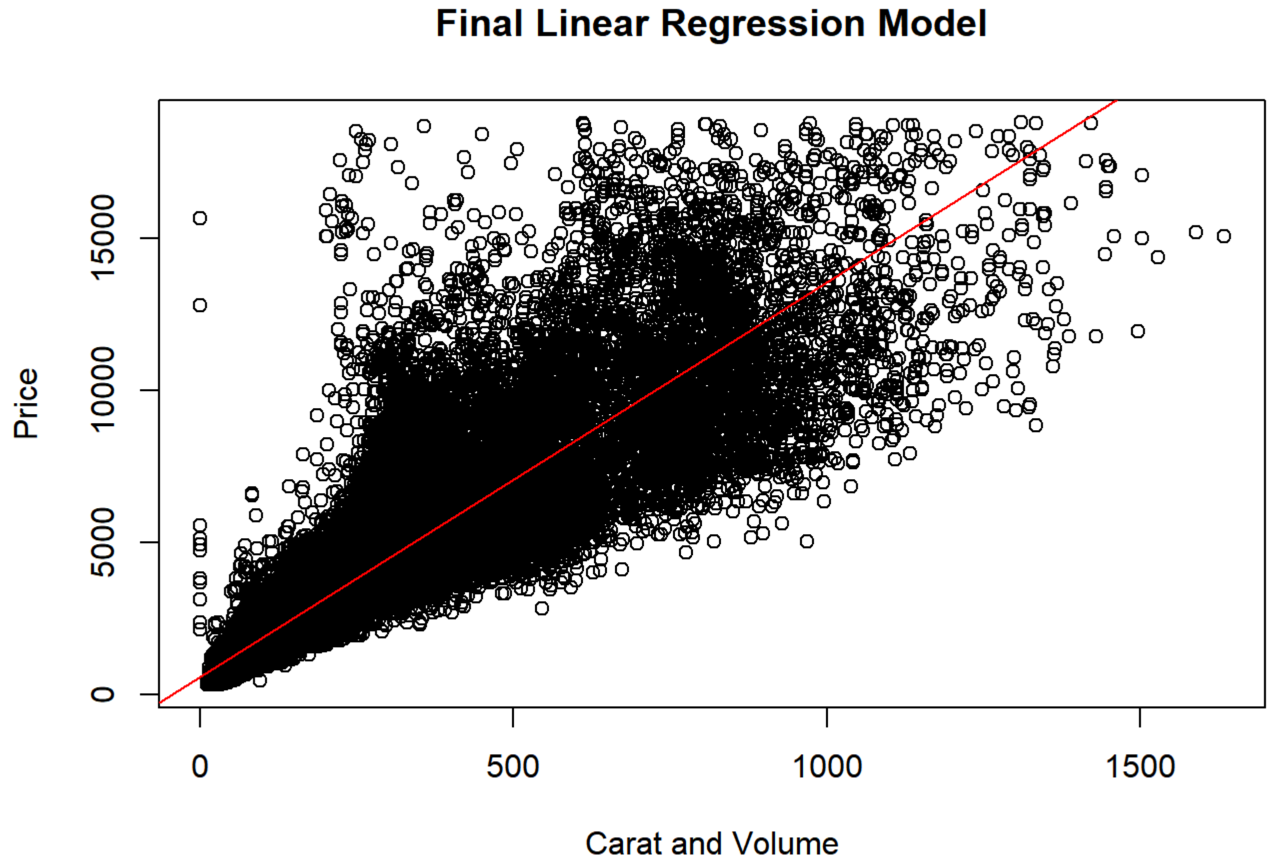
```
10.334
## diamonds$carat:diamonds$means_color
-8.667
## diamonds$volume:diamonds$means_color
9.554
## diamonds$means_cut:diamonds$means_color
4.332
## diamonds$carat:diamonds$means_clarity
-9.112
## diamonds$volume:diamonds$means_clarity
10.080
## diamonds$means_cut:diamonds$means_clarity
5.065
## diamonds$means_color:diamonds$means_clarity
3.608
## diamonds$carat:diamonds$volume:diamonds$means_cut
-6.776
## diamonds$carat:diamonds$volume:diamonds$means_color
-4.967
## diamonds$carat:diamonds$means_cut:diamonds$means_color
8.962
## diamonds$volume:diamonds$means_cut:diamonds$means_color
-10.022
## diamonds$carat:diamonds$volume:diamonds$means_clarity
-5.184
## diamonds$carat:diamonds$means_cut:diamonds$means_clarity
9.420
## diamonds$volume:diamonds$means_cut:diamonds$means_clarity
-10.564
## diamonds$carat:diamonds$means_color:diamonds$means_clarity
8.882
## diamonds$volume:diamonds$means_color:diamonds$means_clarity
-9.738
## diamonds$means_cut:diamonds$means_color:diamonds$means_clarity
-4.297
## diamonds$carat:diamonds$volume:diamonds$means_cut:diamonds$means_color
6.180
## diamonds$carat:diamonds$volume:diamonds$means_cut:diamonds$means_clarity
6.602
## diamonds$carat:diamonds$volume:diamonds$means_color:diamonds$means_clarity
4.709
## diamonds$carat:diamonds$means_cut:diamonds$means_color:diamonds$means_clarity
-9.172
## diamonds$volume:diamonds$means_cut:diamonds$means_color:diamonds$means_clarity
10.205
## diamonds$carat:diamonds$volume:diamonds$means_cut:diamonds$means_color:diamonds$means_clarity
-6.014
##
Pr(>|t|)
## (Intercept)
1.10e-05
## diamonds$carat
< 2e-16
## diamonds$volume
```

```
< 2e-16
## diamonds$means_cut
3.78e-07
## diamonds$means_color
0.000251
## diamonds$means_clarity
1.26e-05
## diamonds$carat:diamonds$volume
4.21e-08
## diamonds$carat:diamonds$means_cut
< 2e-16
## diamonds$volume:diamonds$means_cut
< 2e-16
## diamonds$carat:diamonds$means_color
< 2e-16
## diamonds$volume:diamonds$means_color
< 2e-16
## diamonds$means_cut:diamonds$means_color
1.48e-05
## diamonds$carat:diamonds$means_clarity
< 2e-16
## diamonds$volume:diamonds$means_clarity
< 2e-16
## diamonds$means_cut:diamonds$means_clarity
4.09e-07
## diamonds$means_color:diamonds$means_clarity
0.000309
## diamonds$carat:diamonds$volume:diamonds$means_cut
1.25e-11
## diamonds$carat:diamonds$volume:diamonds$means_color
6.84e-07
## diamonds$carat:diamonds$means_cut:diamonds$means_color
< 2e-16
## diamonds$volume:diamonds$means_cut:diamonds$means_color
< 2e-16
## diamonds$carat:diamonds$volume:diamonds$means_clarity
2.18e-07
## diamonds$carat:diamonds$means_cut:diamonds$means_clarity
< 2e-16
## diamonds$volume:diamonds$means_cut:diamonds$means_clarity
< 2e-16
## diamonds$carat:diamonds$means_color:diamonds$means_clarity
< 2e-16
## diamonds$volume:diamonds$means_color:diamonds$means_clarity
< 2e-16
## diamonds$means_cut:diamonds$means_color:diamonds$means_clarity
1.73e-05
## diamonds$carat:diamonds$volume:diamonds$means_cut:diamonds$means_color
6.44e-10
## diamonds$carat:diamonds$volume:diamonds$means_cut:diamonds$means_clarity
4.10e-11
## diamonds$carat:diamonds$volume:diamonds$means_color:diamonds$means_clarity
2.50e-06
## diamonds$carat:diamonds$means_cut:diamonds$means_color:diamonds$means_clarity
```

```
< 2e-16
## diamonds$volume:diamonds$means_cut:diamonds$means_color:diamonds$means_clarity
< 2e-16
## diamonds$carat:diamonds$volume:diamonds$means_cut:diamonds$means_color:diamonds$means_clarity
1.82e-09
##
## (Intercept)
***
## diamonds$carat
***
## diamonds$volume
***
## diamonds$means_cut
***
## diamonds$means_color
***
## diamonds$means_clarity
***
## diamonds$carat:diamonds$volume
***
## diamonds$carat:diamonds$means_cut
***
## diamonds$volume:diamonds$means_cut
***
## diamonds$carat:diamonds$means_color
***
## diamonds$volume:diamonds$means_color
***
## diamonds$means_cut:diamonds$means_color
***
## diamonds$carat:diamonds$means_clarity
***
## diamonds$volume:diamonds$means_clarity
***
## diamonds$means_cut:diamonds$means_clarity
***
## diamonds$means_color:diamonds$means_clarity
***
## diamonds$carat:diamonds$volume:diamonds$means_cut
***
## diamonds$carat:diamonds$volume:diamonds$means_color
***
## diamonds$carat:diamonds$means_cut:diamonds$means_color
***
## diamonds$volume:diamonds$means_cut:diamonds$means_color
***
## diamonds$carat:diamonds$volume:diamonds$means_clarity
***
## diamonds$carat:diamonds$means_cut:diamonds$means_clarity
***
## diamonds$volume:diamonds$means_cut:diamonds$means_clarity
***
## diamonds$carat:diamonds$means_color:diamonds$means_clarity
***
```

```
## diamonds$volume:diamonds$means_color:diamonds$means_clarity
***
## diamonds$means_cut:diamonds$means_color:diamonds$means_clarity
***
## diamonds$carat:diamonds$volume:diamonds$means_cut:diamonds$means_color
***
## diamonds$carat:diamonds$volume:diamonds$means_cut:diamonds$means_clarity
***
## diamonds$carat:diamonds$volume:diamonds$means_color:diamonds$means_clarity
***
## diamonds$carat:diamonds$means_cut:diamonds$means_color:diamonds$means_clarity
***
## diamonds$volume:diamonds$means_cut:diamonds$means_color:diamonds$means_clarity
***
## diamonds$carat:diamonds$volume:diamonds$means_cut:diamonds$means_color:diamonds$means_clarity
***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1036 on 48772 degrees of freedom
## Multiple R-squared:  0.9069, Adjusted R-squared:  0.9069
## F-statistic: 1.533e+04 on 31 and 48772 DF,  p-value: < 2.2e-16
```

```
plot(diamonds$price ~ x, main = "Final Linear Regression Model", xlab="Carat and Volume",ylab="P
rice")
abline(lm(diamonds$price ~ x), col="red") # just for visual purposes
```



Final Linear Regression Model
Price
0
5000
10000
15000
0
500
1000
1500
Carat and Volume

```
cat("EQUATION of the Line: y =", finalModel$coefficients[1], "+",
    finalModel$coefficients[2], "c +",
    finalModel$coefficients[3], "v +",
    finalModel$coefficients[4], "u +",
    finalModel$coefficients[5], "o +",
    finalModel$coefficients[6], "a +",
    finalModel$coefficients[7], "cv + ",
    finalModel$coefficients[8], "cu + ",
    finalModel$coefficients[9], "vu + ",
    finalModel$coefficients[10], "co + ",
    finalModel$coefficients[11], "vo + ",
    finalModel$coefficients[12], "uo + ",
    finalModel$coefficients[13], "ca + ",
    finalModel$coefficients[14], "va + ",
    finalModel$coefficients[15], "ua + ",
    finalModel$coefficients[16], "oa + ",
    finalModel$coefficients[17], "cvu + ",
    finalModel$coefficients[18], "cvo + ",
    finalModel$coefficients[19], "cuo + ",
    finalModel$coefficients[20], "vuo + ",
    finalModel$coefficients[21], "cva + ",
    finalModel$coefficients[22], "cua + ",
    finalModel$coefficients[23], "vua + ",
    finalModel$coefficients[24], "coa + ",
    finalModel$coefficients[25], "voa + ",
    finalModel$coefficients[26], "uoa + ",
    finalModel$coefficients[27], "cvuo + ",
    finalModel$coefficients[28], "cvua + ",
    finalModel$coefficients[29], "cvoa + ",
    finalModel$coefficients[30], "cuoa + ",
    finalModel$coefficients[31], "vuoa + ",
    finalModel$coefficients[32], "cvuoa",
    "\nwhere c = carat and v = volume and u = means_cut and o = means_color and a = means_clarit
y")
```

```
## EQUATION of the Line: y = 281951.9 + 7457335 c + -52117.11 v + -277805 u + -194137 o + -20241
6.9 a + 3442.02 cv +  -6513373 cu +  46169.8 vu +  -5753615 co +  39908.26 vo +  195966.9 uo +
-5697296 ca +  39525.66 va +  200113.6 ua +  138181.4 oa +  -3625.142 cvu +  -2524.948 cvo +  50
23956 cuo +  -35372.2 vuo +  -2404.714 cva +  4963235 cua +  -34932.48 vua +  4379167 coa +  -30
150.44 voa +  -140397.9 uoa +  2677.757 cvuo +  2609.128 cvua +  1766.833 cvoa +  -3813189 cuoa
+  26662.93 vuoa +  -1922.923 cvuoa
## where c = carat and v = volume and u = means_cut and o = means_color and a = means_clarity
```

Conclusion

With this final linear model calculated, the final R^2 value discovered is 0.9069 which is better than the original 0.8249 when carat was plotted against price alone. This model can be used to determine the price of a diamonds based on its individual characteristics within statistical significance due to the high R^2 value.

Future Analysis

In the future, perhaps instead of doing a simple linear regression model, I can add more polynomial regressions to account for any devistions from the line as the x value grows faster than the lines. Also a machine learning model can be trained using this data (split into test and training data), which may be more useful and accurate with a series of convolutional layers.