Simple & Multiple Linear Regression Model Project

Billy Dang

2024-06-04

Contents

Introduction:

Today we will be exploring the **Diamonds Prices** data set from Kaggle which contains information on prices and attributes of almost 54,000 diamonds. We'll examine the relationships between some of these attributes (cut, price, clarity, etc.) and the price of a given diamond. We'll start by doing some exploratory data analysis before delving into inference/prediction and modeling. In our section on modeling, we'll employ feature engineering and a multiple linear regression model to assess the impact and relationship between these variables.

Data set name: Diamonds (Diamonds Prices2022.csv)

We will be using the following packages:

```r
library(dplyr)
library(skimr)
library(tinytex)
library(tidyverse)
library(ggplot2)
library(ggthemes)
library(reshape)
library(corrplot)
library(pracma)
library(car)
library(generics)
library(broom)
```

Variable Information

Variable Name	Description
`carat`	unit of weight for diamonds: 200mg per carat
`cut`	the quality of the cut (Fair, Good, Very Good, Premium, Ideal)
`color`	the degree of colorlessness by comparing a stone under controlled lighting (J (worst), I, H, G, F, E, D (best))
`clarity`	the measurement of how clear the diamond is (I1 (worst), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (best))
`depth`	the total depth percentage of the diamond calculated by dividing the diamond's total depth by its average diameter and multiplying by 100
`table`	width of top of diamond relative to widest point
`price`	the price of the diamond
`x`	the length of the diamond in mm
`y`	the width of the diamond in mm
`z`	the depth of the diamond in mm

Part 1a: Data Cleaning

Data Transform

Before proceeding with the analysis, it is beneficial to consider data transformation. In linear regression, it is essential to work with continuous variables, as this method models the relationship between a continuous response variable (dependent variable) and one or more predictor variables, which can be either continuous or categorical. For our dataset, most numeric variables are already in float/double format suitable for linear regression, while the ‘price’ & ‘X’ variable is appropriately kept as an integer. However, the variables “cut”, “color”, and “clarity” variables are categorical variables formatted as characters, we will need to mutate these ordinal values to rank their inherent value (i.e. quality of cut). After using the factor() function to do this, we’ll need to use the as.numeric() function as well when generating our models later.

```
diamonds$cut <- factor(diamonds$cut,
                       levels = c("Fair", "Good", "Very Good", "Premium", "Ideal"))
diamonds$color <- factor(diamonds$color,
                       levels = c("J", "I", "H", "G", "F", "E", "D"))
diamonds$clarity <- factor(diamonds$clarity,
                       levels = c("I1", "SI2", "SI1", "VS2", "VS1", "VVS2", "VVS1", "IF"))
```

While there appears to be no immediate need for any numerical transformations within our dataset, notice how x, y, z aren’t exactly descriptive variable names. Let’s adjust this to facilitate easier coding and better readability. We’ll also adjust ‘depth’ to ‘depth_percentage’ to be more concise in it’s actual definition. Another feature to note is the ‘X’ variable column. This must be a unique identifier for the diamond which we’ll leave in for now. We just need to be careful not to include it in our regression model selection.

```
colnames(diamonds)[6] <- "depth_percentage"
colnames(diamonds)[9] <- "length"
colnames(diamonds)[10] <- "width"
colnames(diamonds)[11] <- "depth"
```

Handling Null Values

In the Diamonds dataset, we’ll look for missing values encoded as ‘NA’ or represented by blank cells. A comprehensive check for missing values is essential to ensure data integrity and reliability. Using the `sapply()` function, we assessed missingness across all variables. We found no empty values or null values across all columns. The table below shows missing value counts for each variable, which should all be 0.

```
# Check for empty values in each column
sapply(diamonds, function(x) sum(x == "", na.rm = TRUE))
```

```
##                X            carat              cut            color 
##                0                0                0                0 
##          clarity depth_percentage            table            price 
##                0                0                0                0 
##           length            width            depth 
##                0                0                0
```

```
#checking if there are null values
sapply(diamonds, function(x) sum(is.na(x)))
```

```
##                 X             carat               cut             color 
##                 0                 0                 0                 0 
##           clarity depth_percentage             table             price 
##                 0                 0                 0                 0 
##            length             width             depth 
##                 0                 0                 0 
```

After performing the following due diligence to make sure our data is clean and ready for use, we'll take a sample from the original data (performing the same transformations) for our next section in Exploratory Data Analysis.

Part 1b: Exploratory Data Analysis

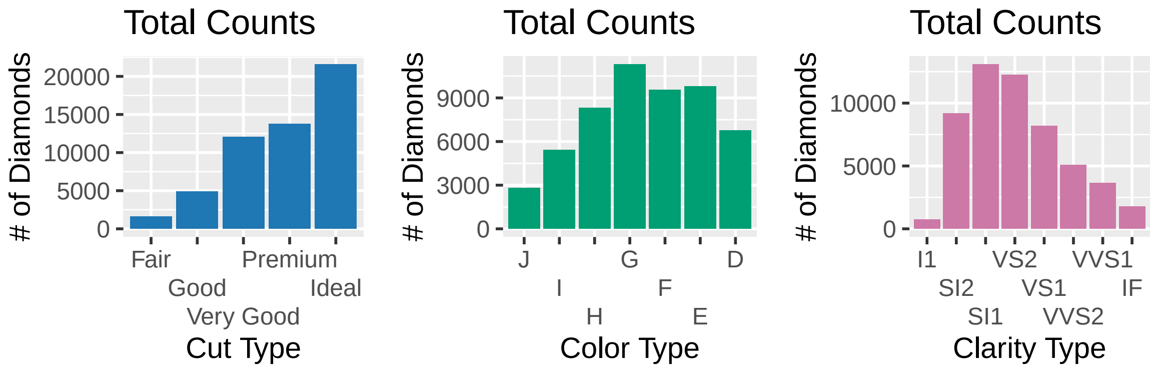
Visual Distribution Comparisons

Let's take a look at the distribution of diamonds across its various ordinal attributes from the original data set, and then take a sample of n = 500 and compare them. We'll initially take a visual approach by graphing them to see if there are any notable discrepancies as we will be working with this sample for the rest of the report.

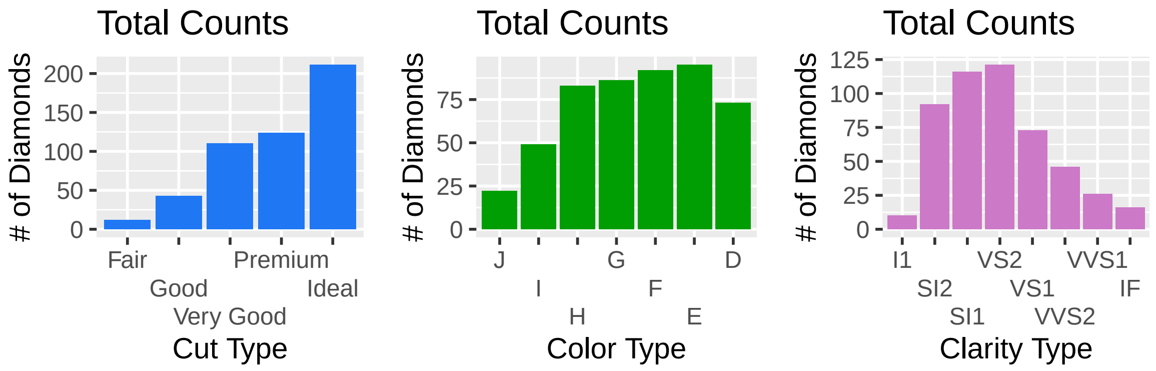
```r
# Select a random sample
diamond_sample = sample_n(diamonds, size = 500)
#Performing necessary transformations to sampled dataset as well
colnames(diamond_sample)[6] <- "depth_percentage"
colnames(diamond_sample)[9] <- "length"
colnames(diamond_sample)[10] <- "width"
colnames(diamond_sample)[11] <- "depth"

diamond_sample$cut <- factor(diamond_sample$cut,
                        levels = c("Fair", "Good", "Very Good", "Premium", "Ideal"))
diamond_sample$color <- factor(diamond_sample$color,
                        levels = c("J", "I", "H", "G", "F", "E", "D"))
diamond_sample$clarity <- factor(diamond_sample$clarity,
                        levels = c("I1", "SI2", "SI1", "VS2", "VS1", "VVS2", "VVS1", "IF"))
```

Original Dataset



Sampled dataset:



It seems the distribution is pretty much the same for cut, color, and clarity between the original and sampled dataset. We can see that for within both frequency distribution graphs (sample & original) 'Fair' is the least common cut type, with ideal being the most common as well. Moving on to the color type distributions we notice our first difference with G being the most common color type in the original vs. the sample's 'E' color type appearing most frequently. However, they do share the same least common color type 'J'. Finally, as we look at the counts for clarity type we notice the same pattern as cut type (where the original & sample datasets share the same most/least common types).

As for the other variables considered (depth, price, carat) we will look at and compare the appropriate summary statistics below.

Summary Statistics Exploration

After using the `skimr` package in R, we were able to generate a comprehensive summary statistic list for the two Diamond datasets. From this, it is relatively safe to conclude that the sampling distributions is close to and representative of the population data. Now, let's see if we can identify any initial relationships between the variables within the newly sampled dataset (which we will be using for the rest of the report). We'll display a couple of the notable statistics below:

```
summary(diamonds$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     326     950    2401    3933    5324   18823
```

```
summary(diamond_sample$price)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     379    1018    2432    3848    4889   18281
```

```
summary(diamonds$carat)
```

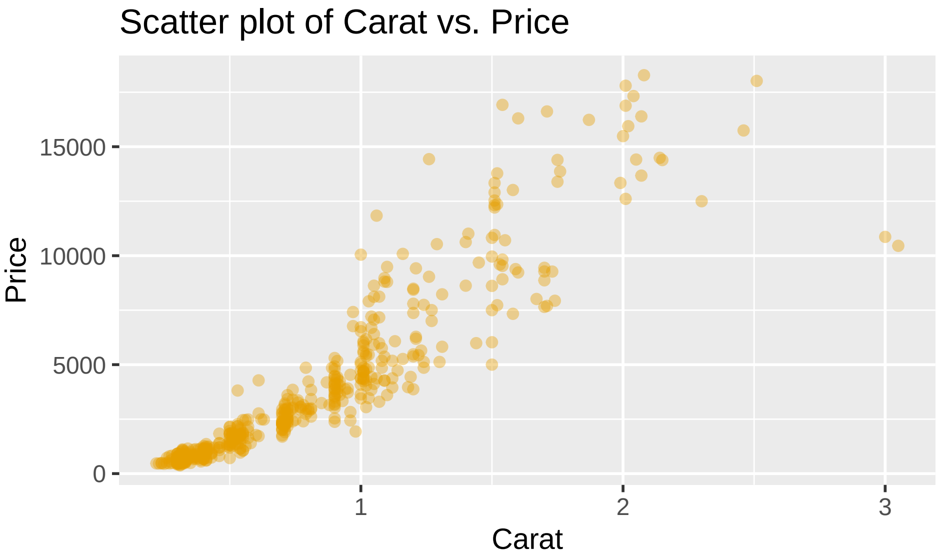
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2000  0.4000  0.7000  0.7979  1.0400  5.0100
```

```
summary(diamond_sample$carat)
```

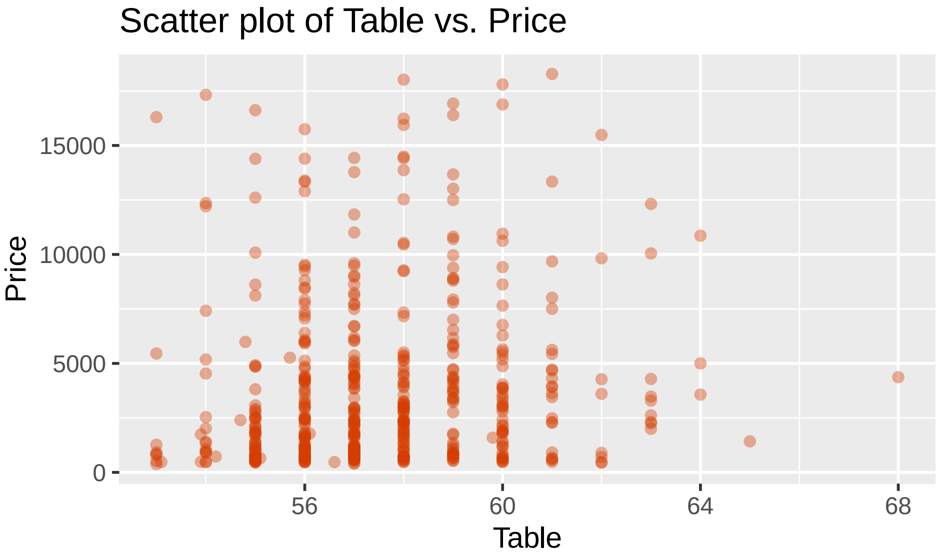
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.220   0.400   0.710   0.797   1.022   3.050
```

Initial Relationship Discoveries

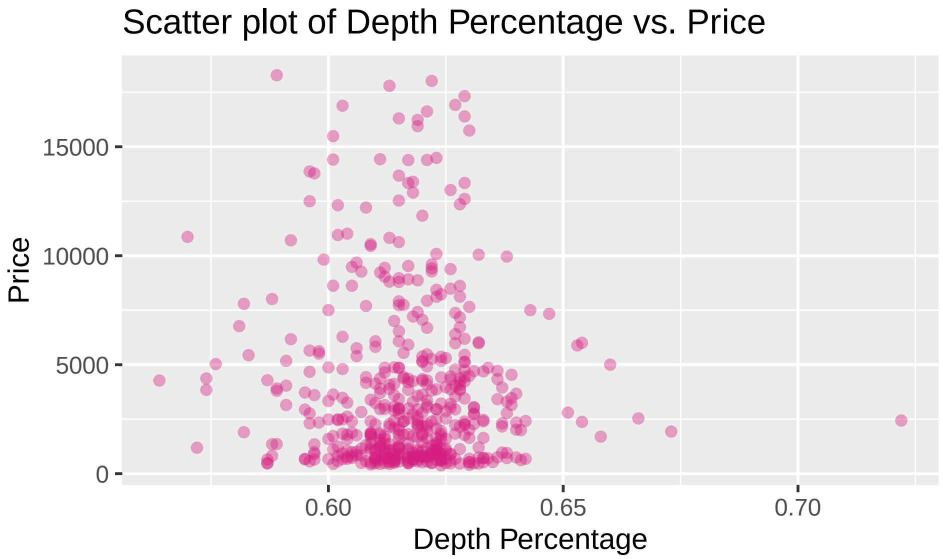
After deciding that our sample is sufficiently representative, we'll create some plots to examine possible relationships between our variables. In this case, we initially focus on depicting "Price" as our dependent variable and looking at how it changes against different types of diamond traits.



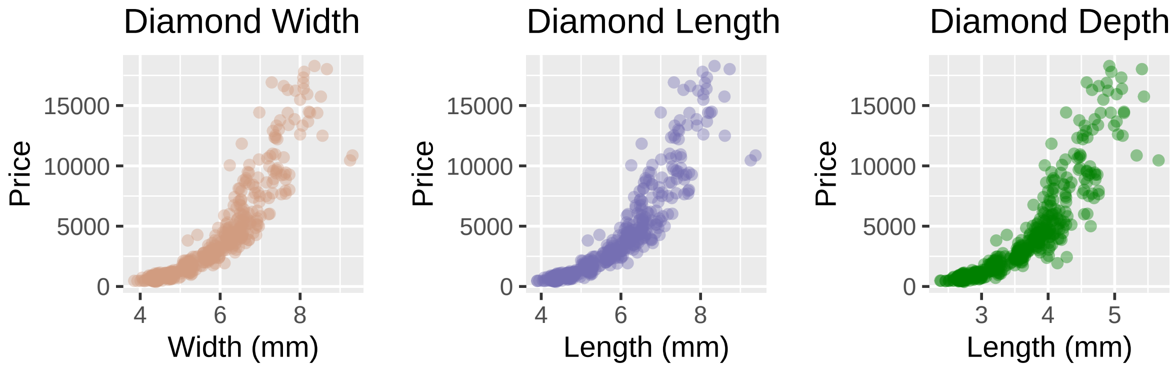
By producing a scatterplot of a diamond's price against its carat size, we see that on average, as a diamond increases in weight so does its market value. This makes intuitive sense, so let's move onto our other variables to find out more about what else may affect a diamond's price.



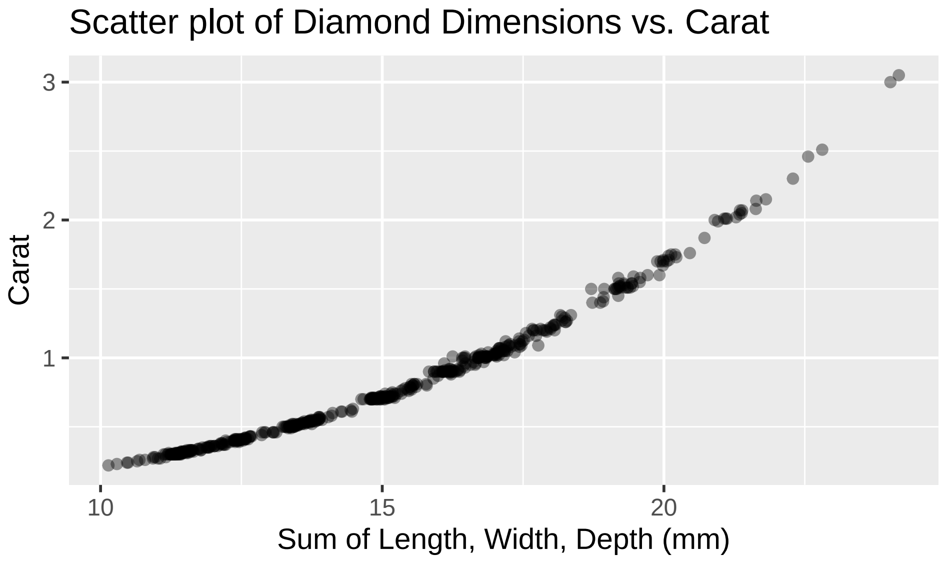
Here we've made a scatter plot of Diamond price relative to its table. Based off the graph alone, it is hard to deduce any relationship between the two variables and a closer examination would be required to draw any conclusions. For now, let's move onto the next variables and see if there are more obvious correlations.



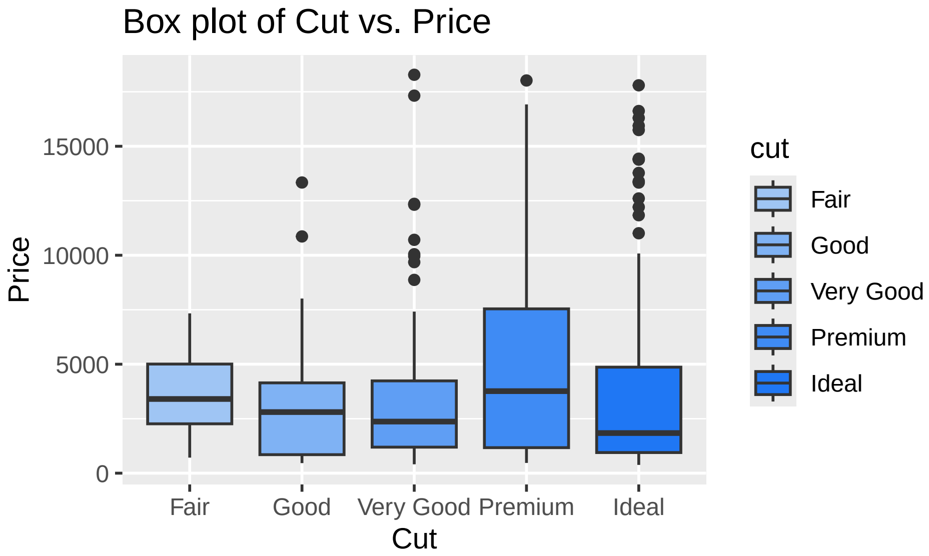
Once again, like the graph before it, the plot of Depth Percentage vs. Price fails to yield any concrete relationship between the two variables. However, there seem to be a clear range where most diamonds fall into in terms of depth percentage (.60 - .65).



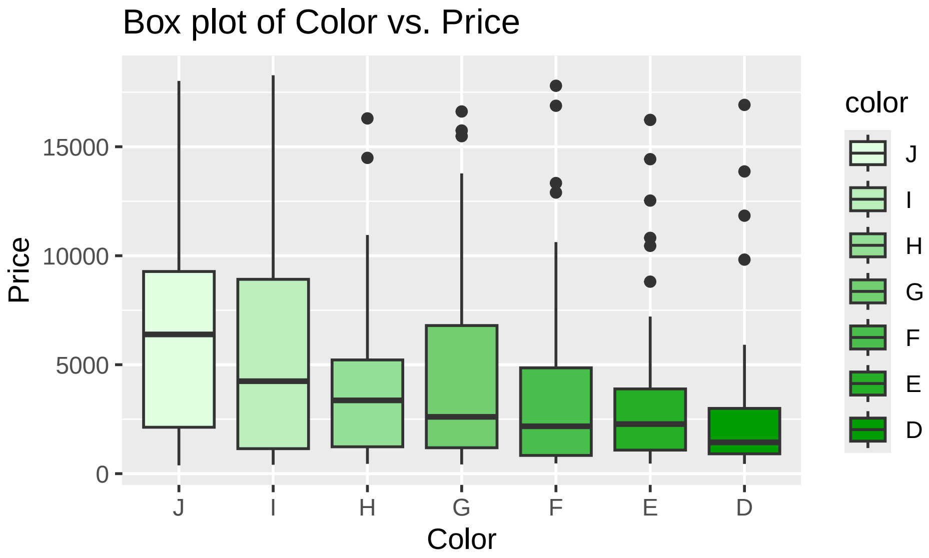
For the three variables above: (Width, Length, Depth) the graphs follow a nearly identical curve to that of carat vs. price. This makes intuitive sense as the relationship of a Diamonds weight will most likely have a strong positive correlation to its size & dimensions. We'll confirm this again by making another plot of the sum of a diamonds dimensions against its carat value below:



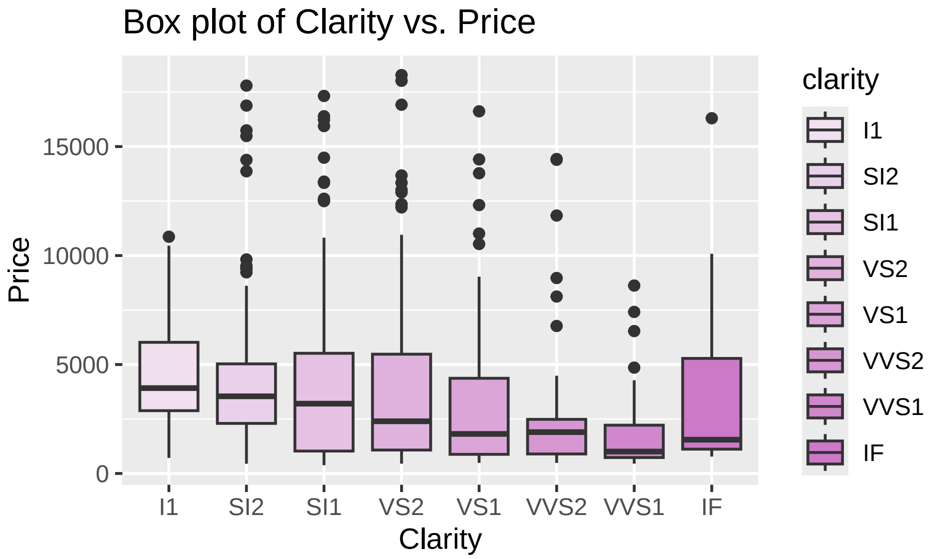
As expected, the graph reveals a strong relationship between the variables.



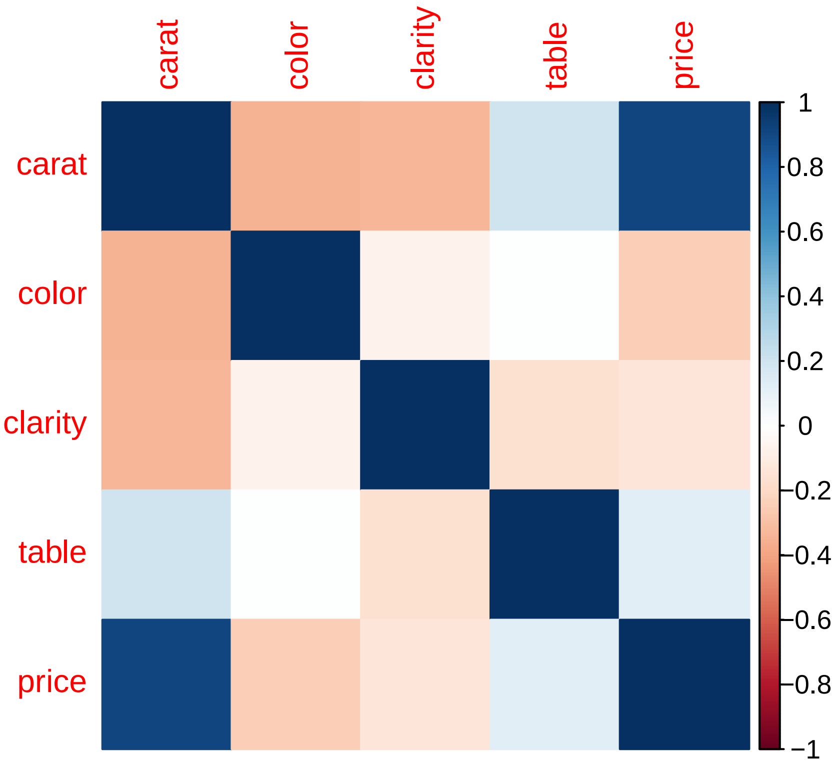
Here, we produced a box plot of a diamond's price as it relates to cut type from worst (**Fair**) to best (**Ideal**). Examining the graph yields unexpected results as there seems to be a decline in price as the quality of cut increases (with the exception being **Premium**). As there are many more factors to consider and tests to be made, we'll hold off on drawing any conclusions here.



Here, we produced a box plot of a diamond's price as it relates to colorlessness from worst (**J**) to best (**D**). Surprisingly we see a gradual decline in median diamond prices as the rating for color increases in ranking. There also seems to be a decline in the spread of diamond prices as color rating increases as well (with the exception being the **G** rating).



Here, we produced a box plot of a diamond's price as it relates to clarity type. Like the graph beforehand, it seems the median value of a diamond actually drops as the quality of the clarity rises. As there a multitude of factors working in tandem to ultimately determine the value of a diamond, we must perform further analysis before making concrete claims on the relationship of such factors and price. We'll start by choosing 3 quantitative and 2 categorical variables from our options above to create a correlation matrix. This will give us a more thorough examination of the connection between these Diamond variables. We'll then run a preliminary multiple linear regression model using all these variables and observe the summary statistics.



By observing the correlation matrix we can see some key relationships between the variables carat, color, clarity, table, and price. First, we see a very high correlation between **carat** and **price**, meaning as carat size increases, the price also tends to increase significantly (which was shown in our scatterplot earlier). Meanwhile, the correlation between **clarity** and **price** is weak which suggest there is little to no linear relationship between the two variables. Moving on to **color**, there is a weak negative correlation between the two which seems to indicate that diamonds with lower color grades tend to have higher prices. Another notable characteristic includes **table** and **color** which have close to no correlation whatsoever. So while this analysis suggests that **carat** and **color** are the strongest predictors of price, we'll place all these variables into a multiple linear regression model and observe.

```
#Run MLR model & observe summary statistics
model1 <- lm(price ~ color + clarity + table + carat, data = diamond_corr)
summary(model1)
```

```
##
## Call:
## lm(formula = price ~ color + clarity + table + carat, data = diamond_corr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11306.5   -629.3   -174.3    515.5   6215.9
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -621.47    1582.67  -0.393 0.694728
## color         280.23      37.03   7.568 1.87e-13 ***
## clarity       508.13      39.18  12.970  < 2e-16 ***
## table         -96.34      27.22  -3.539 0.000439 ***
## carat        8452.16     143.70  58.819  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1307 on 495 degrees of freedom
## Multiple R-squared:  0.8866, Adjusted R-squared:  0.8857
## F-statistic: 967.3 on 4 and 495 DF,  p-value: < 2.2e-16
```

Part 2: Simple Linear Regression

Report Question: Our Hypothesis & SLR Model

The analysis in our report aims to test whether various factors have a significant effect on the selling price of a diamond. We'll being with a simple linear regression model with our predictor **carat** and response **price**.

$H_0 : \beta_1 = 0$ - The carat of a diamond does not have a significant effect on its selling price.

$H_1 : \beta_1 \neq 0$ - The carat of a diamond does have a significant effect on its selling price.

We chose carat to be our most representative variable. The formula of our simple linear regression model is shown below:

$$Y = \beta_0 + \beta_1 X_1 + \epsilon$$

where β_0 is our intercept, β_1 is the coefficient for our predictor variable X_1, and ϵ is our error term. Interpreting, this means that β_0 represents the price of a diamond when its carat value is 0; β_1 represents the change in the price of a diamond given a one unit change in carat (X_1).

```
model2 <- lm(price ~ carat, data = diamond_sample)
summary(model2)
```

```
##
## Call:
## lm(formula = price ~ carat, data = diamond_sample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10135.5   -748.7    -32.4    453.9   7552.6
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2074.5      136.0  -15.26   <2e-16 ***
## carat         7430.5      146.5   50.72   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1558 on 498 degrees of freedom
## Multiple R-squared:  0.8378, Adjusted R-squared:  0.8375
## F-statistic:  2572 on 1 and 498 DF,  p-value: < 2.2e-16
```

Looking at summary statistics, there are many things to note. First we'll start by looking at the intercept β_0, which has a value of -2074.5. Interpreting this means that if the value of β_1 were to be 0, the predicted price would be -\$2074.5 for that given diamond (though we know intuitively that isn't necessarily plausible)

Moving onto the value for our predictor, (7430.5) which suggest that for every increase in a Diamonds carat size, we should expect an increase in value by $7430.5 for a given diamond. Now, returning to our hypothesis we see that the t-value for β_1 is 50.72. Since the P-value for this is <2e-16 (which is much less that a significance level of .05) we can reject the null hypothesis in favor of the alternative: which suggests that carat has a significant effect on the price of a diamond. So how much of the variability is actually explained by carat size? This lies in our R-Squared with a value of .8378. This means that approximately 83.78% of the variability is explained from our 1 predictor. Our adjusted R-Squared isn't far from this value either as we only have 1 predictor.

To further our analysis, we created confidence and prediction intervals of the range in the price of the diamonds in our sample using values of 'carat'. We used the median value of 'carat', in order to focus on the most typical observation found in our sample.

```
##        fit      lwr      upr
## 1 3201.145 3061.977 3340.313
```

```
##        fit      lwr      upr
## 1 3201.145 136.9411 6265.349
```

Interpreting: We are 95% confident that with a carat value of 0.72, a diamond's price according to our confidence interval lies between 3061.977 and 3340.313. Using a prediction interval, we can say we are 95% confident that a new observation (aka a new diamond) will fall between the range of 136.9411 and 6265.349.

Checking our Assumptions

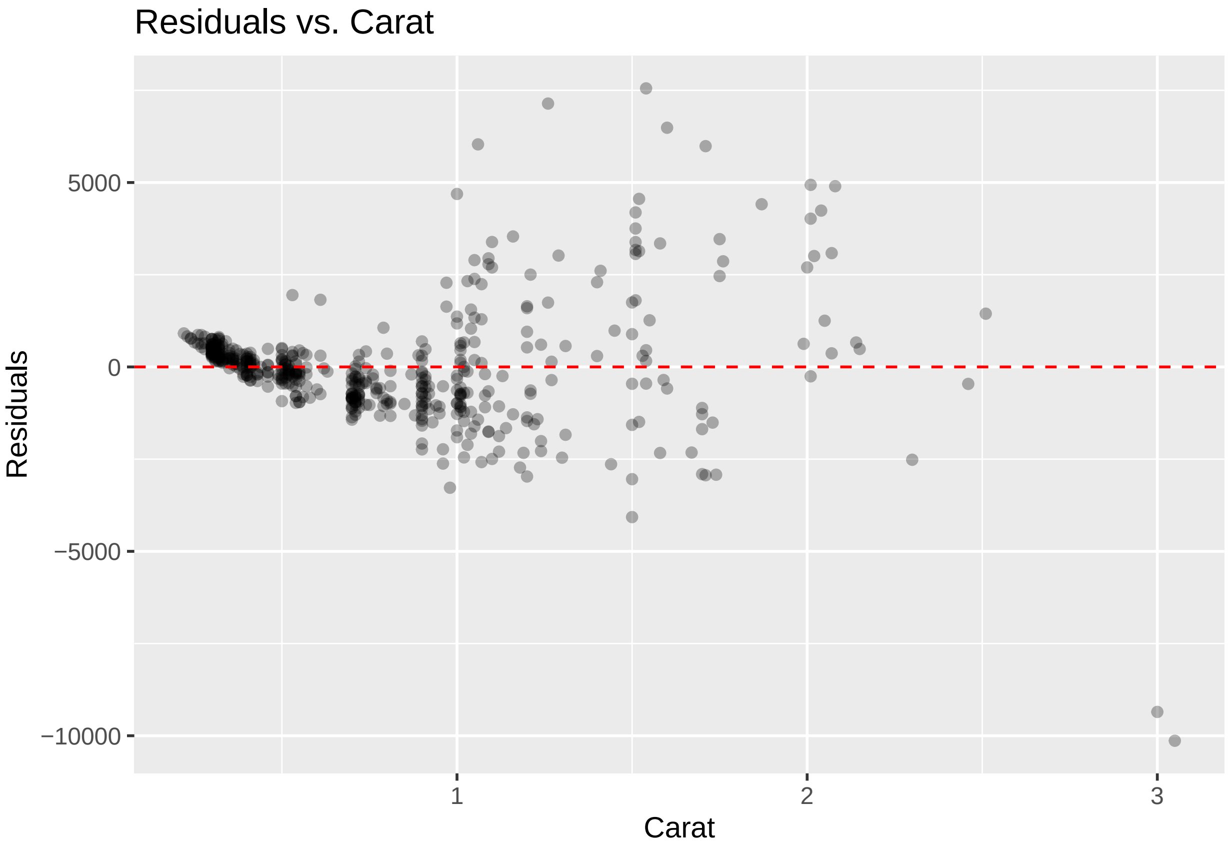
Before conducting any analyses on our model, we must first check four important assumptions of a simple linear regression model - linearity, independence, homoscedasticity, and normality. We will confirm these assumptions through different diagnostic tests and plots.

Linearity

In Part I, we saw in our scatter plot of carat vs. price that as the carat of a diamond increases, so does its price. Thus, the two variables appear to be linearly related. Once we add more predictors, we can check of linearity more thoroughly by plotting each predictor against the residuals and examining the graph.

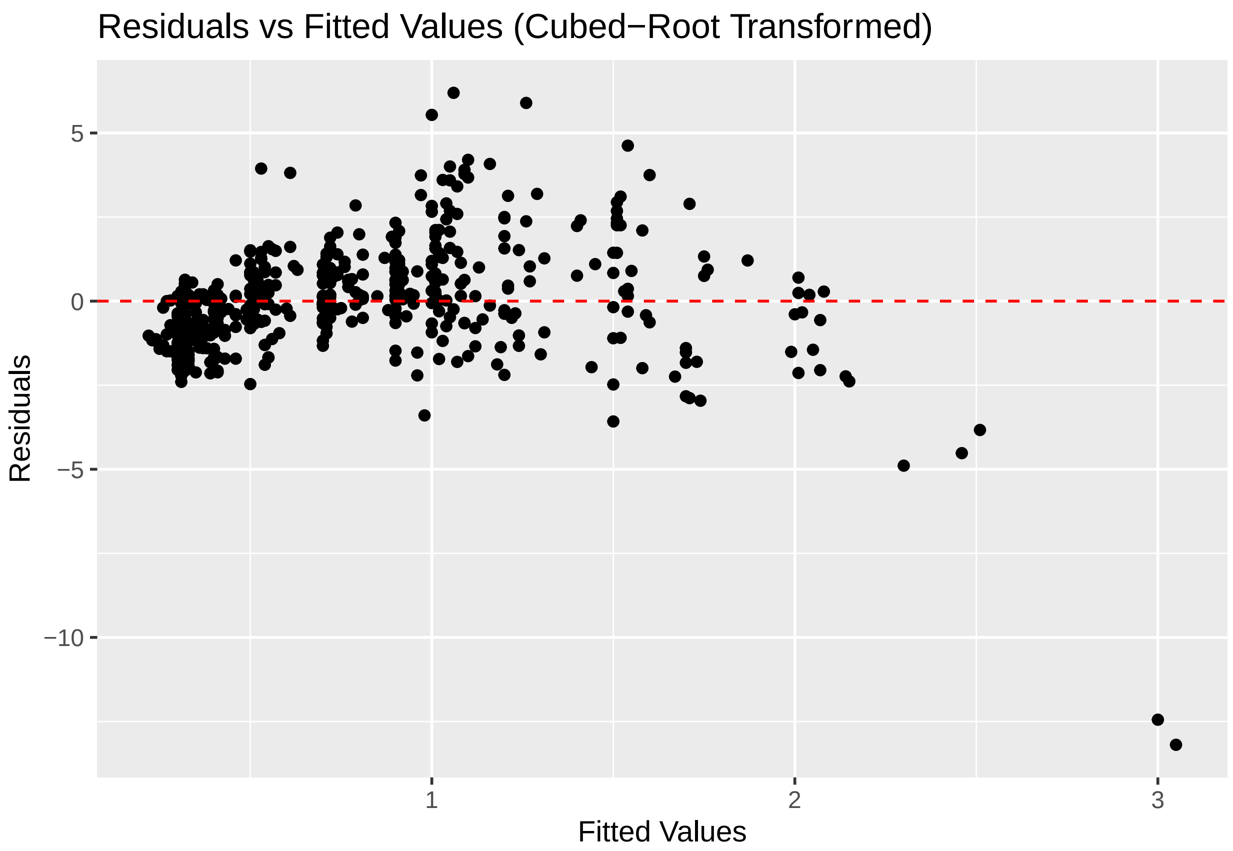
Independence

Next is independence: we can test for independence by plotting the residuals of our diamond sample versus carat. We can see from the plot below that there doesn't seem to be a discernible pattern of our residual points, and that they are mostly clustered around 0. Therefore, we can say that there is no relationship between carat (our explanatory variable) and the residuals, and that they are independent.



Homoscedasticity

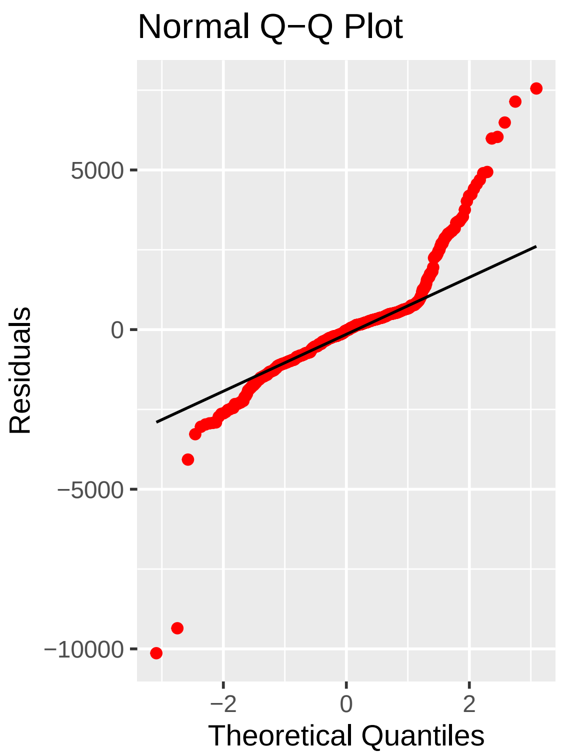
Error terms should have constant variance. If the plot of residual vs. fitted value resembles a funnel shape, indicating non-constant variance, it suggests heteroskedasticity. To ensure homoscedasticity, we check the scale-location plot, where residuals should not have an increasing spread or discernible pattern. As mentioned before, while the lack of a pattern indicates normality, the same cannot be said about constant variance as the residuals gradually get further apart from each other. We'll perform a cubed-square transform and observe how the residuals plot changes.



After performing a cubed root transformation to our Y variable, the graph looks alot better as there is limited spread, and no funnel shape. This means we have successfully transformed the data such that our assumption of homoscedasticity is better met. Though is is not perfect by any means, it will suffice for the purposes of this model and we'll move on to check out final assumption.

Normality

We can test the normality of our model by using a Q-Q plot. Points that generally create a straight line suggest an assumption of normality, which we can see from the points (colored yellow) in our Q-Q plot below, indicating normality. While inference depends on this assumption, it often still works well (due to the Central Limit Theorem) even if this assumption were to fail. With a sample size of 500 our inference should be fairly robust to the normality assumption. We will also perform a Shapiro-Wilk test to confirm normality as well.



```
##
##  Shapiro-Wilk normality test
##
## data:  diamond_sample$price
## W = 0.79444, p-value < 2.2e-16
```

```
##
##  Shapiro-Wilk normality test
##
## data:  diamond_sample$carat
## W = 0.8843, p-value < 2.2e-16
```

```
##
## Call:
## lm(formula = nthroot(price, 3) ~ carat, data = diamond_sample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.1938  -0.9281  -0.0596   0.7791   6.1942
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.7615     0.1474   45.87   <2e-16 ***
## carat         9.2778     0.1588   58.41   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.689 on 498 degrees of freedom
## Multiple R-squared:  0.8726, Adjusted R-squared:  0.8724
## F-statistic:  3412 on 1 and 498 DF,  p-value: < 2.2e-16
```

After we confirmed all of our necessary assumptions above, we ran the model again with the transformed variables. We then called a summary function on this model to look for any notable changes. Here we see that the intercept value is 6.7615 (vs. -2074.5 before the transformation) and the value of β_1 is now 9.2778 (vs. 7430.5). It is important to note that this does not mean for every increase in carat size, the price of a diamond is expected to increase by only ~$9.27 as we have to remember that we cube-rooted the response variable (**price**). Another notable change is the R-squared values which both jumped up ~4% for both the regular and adjusted values with the model now explaining ~87.25% of the variability. Meanwhile, carat was still considered significant with a t-value of 58.41. Let's start adding some more variables to the model and access if it improves. If it does, we'll keep it and add another; if it decreases the adjusted R-squared, we'll exclude it. (Don't worry, we'll be taking a more systematic approach in **Part 3**)

Adding Variables

After conducting an analysis on our simple linear regression model of 'carat' vs. 'price', we can start to add more attributes from our diamonds dataset to see how this may affect our adjusted R-squared, and in turn how well our model fits. We concluded that out of the 10 variables in our diamonds sample, 2 (color and table) were not included in our multiple linear regression model.

After creating 7 new models and running each one to see if the value of R-squared would improve, we arrived at a model with the following predictors: **carat, cut, clarity, depth_percentage, length, width, depth**. Running this multiple linear regression model ended with an R-Squared value of 95.05% and and adjusted R-squared of 94.49%. However it is important to note that a high or low R-squared isn't necessarily good or bad—it doesn't convey the reliability of the model or whether you've chosen the right regression. Especially in this case, a closer examination of some summary statistics and previous graphs we've made before (dimension vs. carat scatter plot & the correlation matrix) hint at some significant issues with this multiple linear regression model: **MULTICOLINEARITY** and **OVERFITTING**. This can be easily seen by applying the vif() function onto our model where carat, depth_percentage, length, width, and depth all yield a VIF value of over the commonly accepted range of ~5-10 (with length, width, and depth having scores over 800+). This makes sense intuitively as mentioned earlier in Part 1, and is a serious issue that creates an over fitting problem within our model. In Part 3, we'll use a more sound and technical approach in creating our best model. (For more information on what is being described above refer to Code Appendix below)

Part 3: Multiple Linear Regression

Depicted below are the summary statistics of the model created earlier from just looking at the R-squared value before and after adding a variable. In the following part, we'll be performing Backward elimination using AIC (Akaike Information Criterion) to prioritize predictive accuracy:

```
## 
## Call:
## lm(formula = nthroot(price, 3) ~ carat + cut + clarity + depth_percentage +
##     length + width + depth, data = diamond_sample)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -5.4309 -0.6119 -0.0366  0.5493  4.4082 
## 
## Coefficients:
```

```
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       -3.15799   11.64396  -0.271   0.7863
## carat             -0.93582    0.46915  -1.995   0.0466 *
## cut                0.18731    0.04614   4.060 5.71e-05 ***
## clarity            0.50071    0.03213  15.586  < 2e-16 ***
## depth_percentage -0.19676    0.18706  -1.052   0.2934
## length             1.26438    1.34872   0.937   0.3490
## width             -0.09060    1.27707  -0.071   0.9435
## depth              5.86056    3.01015   1.947   0.0521 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.059 on 492 degrees of freedom
## Multiple R-squared:  0.9505, Adjusted R-squared:  0.9498
## F-statistic:  1351 on 7 and 492 DF,  p-value: < 2.2e-16
```

Variable Selection: Backwards Elimination using AIC

```
# multiple linear regression model using AIC Backward Elimination
mlr_using_be <- lm(cubed_root_price ~ carat + cut + color + clarity + depth_percentage + table + length
step(mlr_using_be, direction = "backward")
```

```
## Start:  AIC=-91.77
## cubed_root_price ~ carat + cut + color + clarity + depth_percentage +
##     table + length + width + depth
##
##                    Df Sum of Sq    RSS     AIC
## - depth_percentage  1      0.03 399.88 -93.724
## - table             1      0.04 399.89 -93.714
## - carat             1      0.74 400.59 -92.839
## - width             1      0.83 400.68 -92.726
## - depth             1      0.97 400.81 -92.555
## <none>                          399.84 -91.766
## - length            1      2.21 402.06 -91.009
## - cut               1     17.14 416.99 -72.774
## - color             1    151.64 551.48  67.000
## - clarity           1    344.87 744.71 217.197
##
## Step:  AIC=-93.72
## cubed_root_price ~ carat + cut + color + clarity + table + length +
##     width + depth
##
##           Df Sum of Sq    RSS     AIC
## - table    1      0.03 399.91 -95.683
## - carat    1      0.72 400.60 -94.826
## - width    1      1.18 401.06 -94.249
## <none>                 399.88 -93.724
## - length   1      3.53 403.41 -91.324
## - cut      1     17.70 417.58 -74.067
## - depth    1     33.17 433.04 -55.884
## - color    1    152.71 552.59  66.000
## - clarity  1    345.19 745.07 215.437
```

```
##
## Step:  AIC=-95.68
## cubed_root_price ~ carat + cut + color + clarity + length + width +
##     depth
##
##           Df Sum of Sq    RSS     AIC
## - carat    1      0.72 400.63 -96.789
## - width    1      1.15 401.06 -96.248
## <none>                 399.91 -95.683
## - length   1      4.13 404.04 -92.546
## - cut      1     23.16 423.07 -69.538
## - depth    1     39.79 439.70 -50.254
## - color    1    152.96 552.87  64.258
## - clarity  1    345.17 745.08 213.440
##
## Step:  AIC=-96.79
## cubed_root_price ~ cut + color + clarity + length + width + depth
##
##           Df Sum of Sq    RSS     AIC
## - width    1      1.13 401.75 -97.386
## <none>                 400.63 -96.789
## - length   1      3.71 404.33 -94.187
## - cut      1     23.22 423.84 -70.621
## - depth    1     39.09 439.72 -52.239
## - color    1    157.25 557.88  66.765
## - clarity  1    344.57 745.19 211.517
##
## Step:  AIC=-97.39
## cubed_root_price ~ cut + color + clarity + length + depth
##
##           Df Sum of Sq    RSS     AIC
## <none>                 401.75 -97.386
## - cut      1     23.52 425.27 -70.938
## - depth    1     42.96 444.71 -48.595
## - length   1     68.20 469.95 -20.992
## - color    1    156.94 558.69  65.493
## - clarity  1    355.99 757.74 217.866
```

```
##
## Call:
## lm(formula = cubed_root_price ~ cut + color + clarity + length +
##     depth, data = diamond_sample)
##
## Coefficients:
## (Intercept)          cut        color      clarity       length        depth
##    -16.9711       0.2078       0.3536       0.5779       2.5532       3.3278
```

```
mlr_using_be <- lm(cubed_root_price ~ cut + color + clarity + length + depth, data = diamond_sample)
summary(mlr_using_be)
```

```
##
## Call:
## lm(formula = cubed_root_price ~ cut + color + clarity + length +
##     depth, data = diamond_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.1721 -0.5198 -0.0445  0.4942  3.0703
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -16.97115    0.39229 -43.262  < 2e-16 ***
## cut           0.20779    0.03864   5.378 1.16e-07 ***
## color         0.35355    0.02545  13.891  < 2e-16 ***
## clarity       0.57793    0.02762  20.922  < 2e-16 ***
## length        2.55317    0.27881   9.157  < 2e-16 ***
## depth         3.32780    0.45789   7.268 1.44e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9018 on 494 degrees of freedom
## Multiple R-squared:  0.964,  Adjusted R-squared:  0.9636
## F-statistic:  2644 on 5 and 494 DF,  p-value: < 2.2e-16
```

After using the aforementioned technique for our variable selection, we ended with the a model using cut, color, clarity, length, and depth as our predictors for price. All of these variables are statistically significant at a .05 level (still significant at lower levels as well) and have an adjusted R-squared value of 96.36%. This model accounts for ~1.4% more variability explained by the model than our previous model. Going back to the variables, we see that all are positively correlated to the diamonds value and increases in any variable value (cut, color, etc.) is projected to increase the (cubed_root) price of a diamond. While we've attained statistically significant predictors that have resulted in a model with a high adjusted R-squared value, there are some issues with our model that we must note. Using the vif() function reveals high levels of multicollinearity within our model which, as one may recall, is a necessary assumption for an accurate model. The correlation exists between the length and depth variables (which we have noted before as variables subject to this issue) so let's create another model that removed that less statistically significant variable based on t-value (which in this case would be depth) and note any differences.

```r
vif(mlr_using_be)
```

```
##       cut     color   clarity    length     depth
##  1.101631  1.176387  1.238423 58.719178 59.489688
```

```r
adj_mlr <- lm(cubed_root_price ~ cut + color + clarity + length , data = diamond_sample)
summary(adj_mlr)
```

```
##
## Call:
## lm(formula = cubed_root_price ~ cut + color + clarity + length,
##     data = diamond_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.1671 -0.5338 -0.0285  0.5123  3.3888
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -16.31578    0.40127 -40.660  < 2e-16 ***
## cut           0.14367    0.03954   3.634 0.000309 ***
## color         0.33488    0.02661  12.583  < 2e-16 ***
## clarity       0.57043    0.02901  19.661  < 2e-16 ***
## length        4.55652    0.04400 103.550  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9478 on 495 degrees of freedom
## Multiple R-squared:  0.9601, Adjusted R-squared:  0.9598
## F-statistic:  2980 on 4 and 495 DF,  p-value: < 2.2e-16
```

```
vif(adj_mlr)
```

```
##      cut    color  clarity   length
## 1.044181 1.164402 1.236696 1.323971
```

Notable Issues & Adjustments

Here, we've created another multiple linear regression model using cut, clarity, color, and length only and ran it to produced the summary statistics noted above. Starting with the variables, we see that slight changes in their values with changes most notable in cut and length. The model now shows that for every increase in diamond length (mm) we expect a ~4.56 increase in the cubed-root price. (For purposes of interpretation, feel free to transform the response variable back as necessary.) Meanwhile, the adjusted R-squared value barely decreases to 95.98% which is less than a .5% decrease when removing 1 whole predictor. This seems to tell us that the model derived from Backward Elimination using AIC was over fitted with too many variables. This makes sense as these models can only find the "best" combination for the variables they started with. Omitting or having too many variables (especially those with high collinearity) relative to your sample size will only hurt the model building process. What are some ways to get around this? Well, one method was just shown above which drops the collinear variable(s). Another way we can avoid this issue is to remove the highly collinearity variables from the pool of predictors Forward/Backward/Stepwise regression can use in the first place. Or, we can also create a single variable that is some function of the problematic variables (as seen in Part 1 where we summed the dimensions). As we move onto generating confidence intervals for a mean predicted value and the predicted intervals of a future predicted value for a combination of X's, we'll use our adjusted model to meet the necessary assumptions of regression modelling.

Confidence & Prediction Intervals

In deciding the specific values to use in our confidence intervals and prediction intervals, we looked at the most common attributes in our diamond sample. We used the most common cut, color, and clarity in our sample ("Ideal", "G", "SI1", respectively), and the median length of our diamond sample; we feel that these choices help give our analysis a broader range of our sample to apply to.

Interpreting: The confidence interval suggests that we are 95% confident that the true mean price of diamonds with these attributes lies between $2480.732 and $2636.745. The prediction interval indicates that we are 95% confident that the price of an individual diamond with these attributes lies between $1646.661 and $3755.398.

```
##        fit      lwr      upr
## 1 2395.332 2320.758 2471.487
```

```
##        fit      lwr     upr
## 1 2395.332 1525.804 3544.86
```

Summary of Report

The analysis conducted in this report reveals that cut, color, clarity, and length significantly influence the price of a diamond in our sample with all variables having a strong, positive correlation. The multiple linear regression model including all of these factors explains a high amount of the variance in diamond price within our sample, with an adjusted R-squared value of 0.9598. Overall, this analysis aids in emphasizing that there is no singular factor that determines a diamond's price, nor that the highest value of each factor will always lead to the most valuable diamond. Of course, further research can be conducted to explore additional factors that are outside our current scope of inference (i.e. the location the diamond was found, what retailer each listed price was from, etc.) that may increase or devalue the price of a diamond.

Acknowledgements

Special thanks to David Nichols for his website on coloring for colorblindness https://davidmathlogic.com/colorblind/. Accessibility is key to any report and his guidance on the matter helped generate the graphs within this project.

Any lecture slides, labs, and datasets referenced are from PSTAT 126, courtesy of Professor Puja Pandey, University of California, Santa Barbara.

Appendix

All of the code to replicate the following project and graphs can be found at: https://github.com/billydanggg/Projects