PSTAT126-Final-V2

2024-06-07

PART - 1 Data Description and Descriptive Statistics

```
Diamond_dataset <- read.csv("/users/robeh/desktop/DiamondsPrices2022.csv")
attach(Diamond_dataset)
head(Diamond_dataset)
```

```
##   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
```

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
set.seed(1)
diamonds_sample <- sample_n(Diamond_dataset, 500)

library(skimr)
# Have to use this instead of skim, because I get error trying to load the histogram
# due to a latex error
skim_without_charts(diamonds_sample)
```

Table 1: Data summary

Name	diamonds_sample
Number of rows	500

Number of columns	11
Column type frequency:	
character	3
numeric	8
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
cut	0	1	4	9	0	5	0
color	0	1	1	1	0	7	0
clarity	0	1	2	4	0	8	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100
X	0	1	26231.28	15342.65	121.00	12185.25	26185.00	39254.25	53917.00
carat	0	1	0.84	0.54	0.23	0.40	0.72	1.08	5.01
depth	0	1	61.87	1.44	56.90	61.20	61.90	62.60	70.20
table	0	1	57.37	2.12	52.00	56.00	57.00	59.00	66.00
price	0	1	4248.63	4345.51	362.00	917.00	2657.00	5374.50	18757.00
x	0	1	5.79	1.21	0.00	4.68	5.75	6.56	10.74
y	0	1	5.80	1.21	0.00	4.71	5.80	6.59	10.54
z	0	1	3.58	0.77	0.00	2.90	3.55	4.07	6.98

Description of Variables and Observational Unit

Main Variables Being Testing:

The **Color** variable is a color grading scale that ranges from D to H (in the data set) and rates the colorless or near colorless of a white diamond. Diamonds in the ranges of D to F are the best that contain no other color and are of the highest quality. Diamonds in the ranges of G to J are still high quality but contain slight traces of color.

The **Clarity** variable rates the presence of internal or external blemishes, known as inclusions within the diamond. In the data set the ratings go from VVS2, VS1, VS2, SI1, SI2. VVS2 means that inclusions are very difficult to see under 10x magnification. VS means that inclusions are difficult to see under 10x magnification. VS2 means that inclusions are somewhat easier to see under 10x magnification, but still minor. SI1 means that inclusions are noticeable under 10x magnification. SI2 means that inclusions are more easily noticeable under 10x magnification.

The **Price** variable determines the monetary value of a diamond. There are a lot of factors that go into determining the value of a diamond like cut, clarity, color, shape, and many more.

The **Table** of a diamond refers to the large flat face when looking down at the top of the diamond. In the data set the variable is a percentage of how big the table is compared to the total diameter of the diamond.

The **Depth** of a diamond is the distance from the large flat surface of the top of the diamond, known as the table, to the bottom of the diamond, known as the culet. In the data set the variable is a percentage of how deep a diamond in relation to its width.

The **Observational Unit** of the data set is a single diamond out of the total 54,000 diamonds that make up the data set. Each contain the variables stated before as well as 6 other variables that are outside the scope of this project.

Other Variables in The Data Set:

The **Carat** refers to the weight unit that measures gemstones, in this case it is measuring diamonds. One carat is equal to 200 milligrams.

The **Cut** variable is a categorical variable is a rating that is an important piece of the puzzle that determines how good the diamond is overall. It goes into how the diamond looks visually as well as how well the diamond is crafted.

The **X**, **Y**, and **Z** variable just refers to the dimensions of the diamond. The **x** is the length in mm, the **y** is the width in mm, and the **z** is the depth in mm.

```
fit0 <- lm(price ~ table + clarity, data = diamonds_sample)
summary(fit0)
```

```
## 
## Call:
## lm(formula = price ~ table + clarity, data = diamonds_sample)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
##  -5925  -2879  -1389   1269  15061 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1903.40    5739.44  -0.332    0.740
## table         122.75      91.19   1.346    0.179
## clarityIF   -3186.79    2101.70  -1.516    0.130
## claritySI1  -1249.88    1789.89  -0.698    0.485
## claritySI2    978.29    1806.25   0.542    0.588
## clarityVS1  -1092.38    1820.13  -0.600    0.549
## clarityVS2   -743.14    1791.18  -0.415    0.678
## clarityVVS1 -2482.48    1909.95  -1.300    0.194
## clarityVVS2 -1790.74    1855.09  -0.965    0.335
## 
## Residual standard error: 4244 on 491 degrees of freedom
## Multiple R-squared:  0.06139,    Adjusted R-squared:  0.0461 
## F-statistic: 4.014 on 8 and 491 DF,  p-value: 0.0001244
```

From the results we can see the the variables clarity and table don't have a significant effect on the price variable. When looking at the R-Squared value, we see that they only account for 6.14% of the variance in price.

```
table(diamonds_sample$color)
```

```
## 
##   D   E   F   G   H   I   J 
##  63  83  78 109  81  51  35
```

```
diamonds_sample$D = ifelse(diamonds_sample$color == "D", 1, 0)
diamonds_sample$E = ifelse(diamonds_sample$color == "E", 1, 0)
diamonds_sample$F = ifelse(diamonds_sample$color == "F", 1, 0)
diamonds_sample$G = ifelse(diamonds_sample$color == "G", 1, 0)
diamonds_sample$H = ifelse(diamonds_sample$color == "H", 1, 0)
diamonds_sample$I = ifelse(diamonds_sample$color == "I", 1, 0)
diamonds_sample$J = ifelse(diamonds_sample$color == "J", 1, 0)
attach(diamonds_sample)
```

```
## The following objects are masked from Diamond_dataset:
##
##     carat, clarity, color, cut, depth, price, table, x, X, y, z

## The following object is masked from package:base:
##
##     F
```

```
fit2 <- lm(price ~ depth + E + F + G + H + I + J, data = diamonds_sample)
summary(fit2)
```

```
##
## Call:
## lm(formula = price ~ depth + E + F + G + H + I + J, data = diamonds_sample)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -6098  -2664  -1166   1394  13964
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 3656.786   8141.406   0.449 0.653515
## depth         -7.025    131.909  -0.053 0.957551
## E           -378.775    703.899  -0.538 0.590745
## F             20.820    713.491   0.029 0.976732
## G           1607.431    666.962   2.410 0.016315 *
## H           1310.871    711.979   1.841 0.066199 .
## I           2907.470    795.530   3.655 0.000285 ***
## J           3239.448    888.378   3.646 0.000294 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4210 on 492 degrees of freedom
## Multiple R-squared:  0.07466,    Adjusted R-squared:  0.0615
## F-statistic: 5.671 on 7 and 492 DF,  p-value: 2.577e-06
```

Now looking at depth and color (using dummy variables) we can see how alike the last test they have a very small significance on price. Back to the R-Squared value it shows that they only account for 7.47% of the variance in the price variable.

```
fit3 <- lm(price ~ color + clarity + depth + table, data = diamonds_sample)
summary(fit3)
```

```
##
## Call:
## lm(formula = price ~ color + clarity + depth + table, data = diamonds_sample)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -6918  -2626   -977   1273  14461
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2208.20   10943.39  -0.202 0.840171
## colorE       -358.89     686.70  -0.523 0.601467
## colorF        203.41     694.11   0.293 0.769606
## colorG       2321.92     665.84   3.487 0.000533 ***
## colorH       1383.56     696.20   1.987 0.047452 *
## colorI       2931.89     774.62   3.785 0.000173 ***
## colorJ       3090.46     871.77   3.545 0.000431 ***
## clarityIF   -3464.93    2051.20  -1.689 0.091821 .
## claritySI1  -1009.11    1733.98  -0.582 0.560863
## claritySI2   1223.92    1752.06   0.699 0.485164
## clarityVS1  -1188.61    1769.17  -0.672 0.502001
## clarityVS2   -594.74    1738.07  -0.342 0.732363
## clarityVVS1 -2766.10    1861.11  -1.486 0.137861
## clarityVVS2 -1661.21    1799.34  -0.923 0.356344
## depth          -4.35     131.99  -0.033 0.973721
## table         109.64      90.89   1.206 0.228258
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4084 on 484 degrees of freedom
## Multiple R-squared:  0.1434, Adjusted R-squared:  0.1169
## F-statistic: 5.404 on 15 and 484 DF,  p-value: 3.838e-10
```

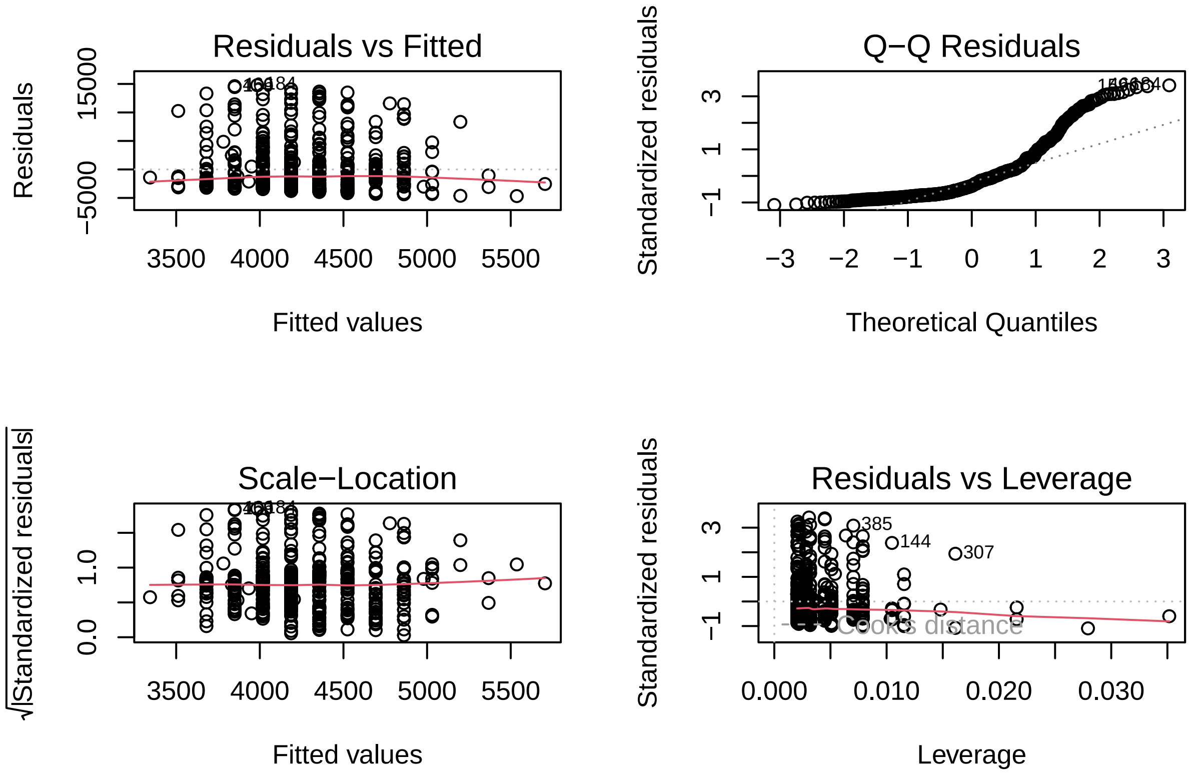
Constructing a model with all the variables that I randomly chose, it is apparent that the ones I chose don't have that big of an impact when it comes to the effect of the price. In the last summary, it does show a lot of significance stars but in the end the R-Squared value only comes out to 0.1434, something I'll have to work on from here on out on this project.

Comments on Part 1

I was surprised that the variables I choose to test showed that they didn't have that big an effect on the overall price of the diamond. The variables I chose had a very low R-Squared value which I wasn't really expected. However, after looking at the data again I realized that the percentages of the depth and table don't can't really quantify the diamond. They give a percentage of the proportion, but diamond that are way different in size could still have the same depth or table, so it makes sense that these values don't really have a big affect on the price.

PART - 2 Simple Linear Regression

```
SLR <- lm(price ~ table, data = diamonds_sample)
par(mfrow=c(2,2))
plot(SLR)
```



```
summary(SLR)
```

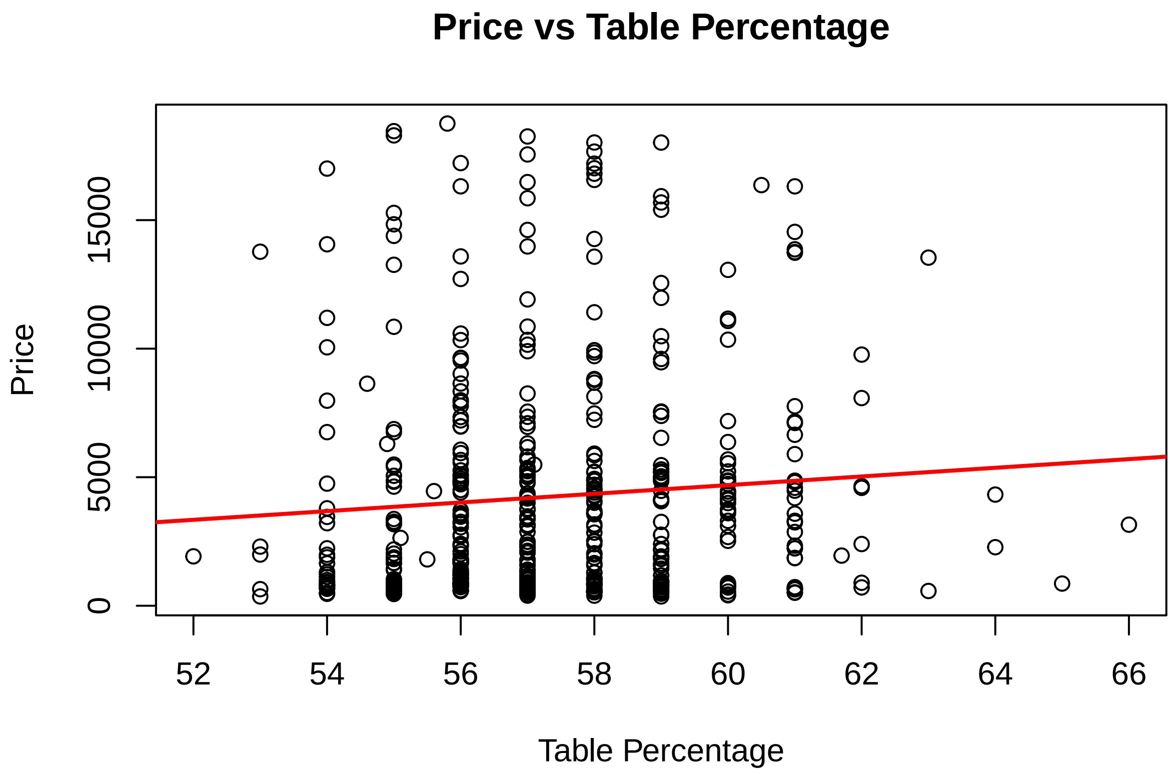
```
##
## Call:
## lm(formula = price ~ table, data = diamonds_sample)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -4673  -3091  -1614   1100  14772
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5426.41    5248.71  -1.034   0.3017
## table         168.66      91.43   1.845   0.0657 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4335 on 498 degrees of freedom
## Multiple R-squared:  0.006786,   Adjusted R-squared:  0.004791
## F-statistic: 3.402 on 1 and 498 DF,  p-value: 0.06569
```

```
confint(SLR)
```

```
##                    2.5 %    97.5 %
## (Intercept) -15738.76146 4885.9327
## table          -10.98675  348.2986
```

```
par(mfrow=c(1,1))
plot(diamonds_sample$table, diamonds_sample$price,
     main = "Price vs Table Percentage",
     xlab = "Table Percentage", ylab = "Price")

abline(SLR, col = "red", lwd = 2)
```



From the plot of this simple linear model between the price and the table we can start by looking at the Residuals vs. Fitted and see that there really isn't a pattern that emerges here. The points have a lot of fluctuation and many are very far away from zero showing that the prediction values aren't very significant.

When it comes to the Q-Q Residuals, the start of the graph doesn't look all too bad. The points form around that straight line from the beginning, but a little halfway through that line the points take a steep upturn and don't return back to the straight line visually showing that the residuals aren't normally distributed as the quantiles increase.

The Scale Location graph goes along with the trend of proving that our two variables aren't significant to each other. This part of the plot doesn't show the points forming a funnel shape that would suggest constant variance among the residuals, instead the points are scattered everywhere.

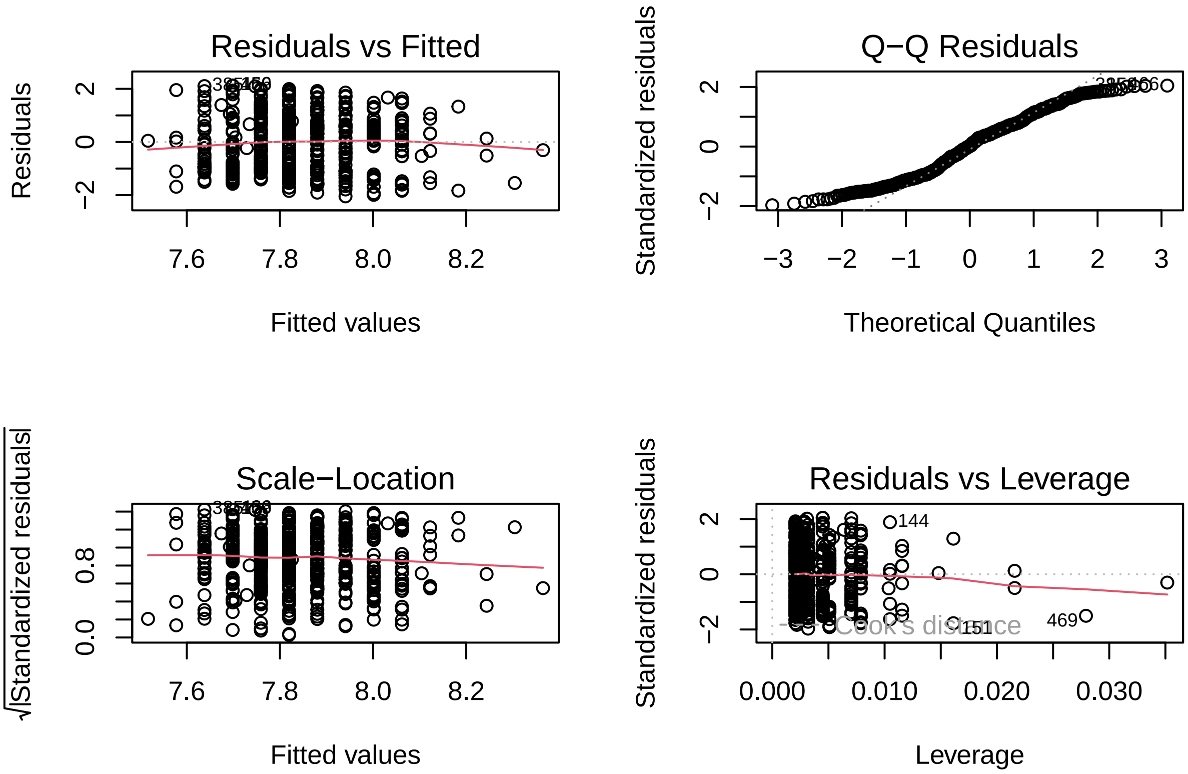
Lastly in the Residuals vs. Leverage, we see a decent amount of outliers that have a high leverage although they don't have high residuals, as the residuals go up the leverage trends to go down for these outliers. Still these outliers show that there can be an effect on our model because of them.

Now looking at the summary statistics, the table is coming to the same conclusion. We see that the p-values are very high and doesn't push is towards the idea of a significant affect. Moving onto the R-Squared value, we see that it is at an extremely low value of 0.006786, or that table accounts for 0.68% of the variance of the price.

Moving onto the confidence interval, we can see how the intercept and table both include 0 within the interval, proving that the parameters aren't significant. The large range at which the values can be from also show the little significance if any that the parameters have on each other.

Lastly, the graph of the price vs table with the plotted regression line is a clear visual that they have little to no relationship. The graph shows how the points do not end up along the line at all and many are scattered around from outliers as there isn't a clear pattern in their grouping.

```
SLR2 <- lm(log(price) ~ table, data = diamonds_sample)
par(mfrow=c(2,2))
plot(SLR2)
```

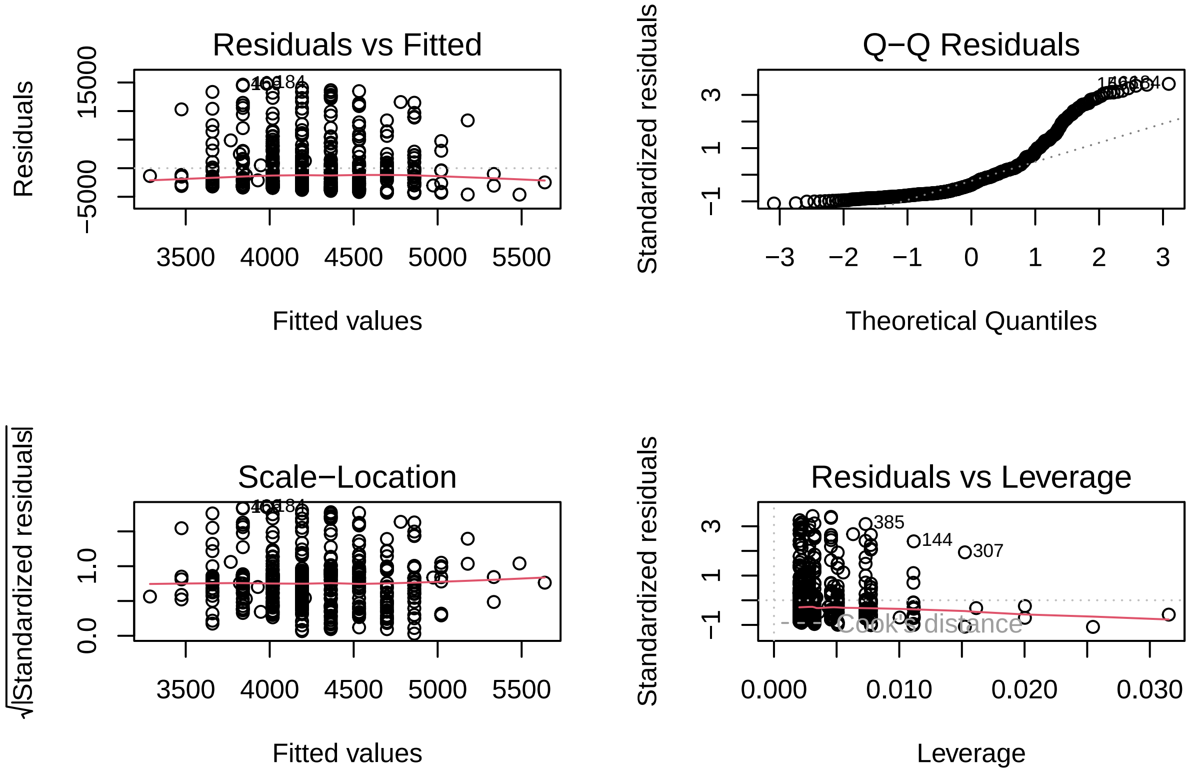


```
summary(SLR2)
```

```
##
## Call:
## lm(formula = log(price) ~ table, data = diamonds_sample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.04639 -0.96340  0.02053  0.76704  2.12499
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.36673    1.26083   3.463 0.000579 ***
## table         0.06058    0.02196   2.758 0.006029 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.041 on 498 degrees of freedom
## Multiple R-squared:  0.01504,    Adjusted R-squared:  0.01307
## F-statistic: 7.607 on 1 and 498 DF,  p-value: 0.006029
```

```
SLR3 <- lm(price ~ log(table), data = diamonds_sample)
par(mfrow=c(2,2))
plot(SLR3)
```



```
summary(SLR3)
```

```
##
## Call:
## lm(formula = price ~ log(table), data = diamonds_sample)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -4624  -3088  -1614   1094  14774
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -35673      21400  -1.667   0.0962 .
## log(table)      9860       5285   1.866   0.0627 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4335 on 498 degrees of freedom
## Multiple R-squared:  0.00694,    Adjusted R-squared:  0.004946
## F-statistic:  3.48 on 1 and 498 DF,  p-value: 0.0627
```

After testing the initial hypothesis it was apparent the the variables I was using didn't have much correlation between the two of them. To try and make the model better, I experimented with using the log function on the y and x variables of the model. As you can see I started with the log on the y or the price and made a difference by getting the R-Squared value to 0.01504. Although this is extremely low, within the scope of this test I more than doubled the original R-Squared value. After this, I tried to switch the log function onto the x variable, however it didn't have a R-Squared value nearly as good as my first transformation so I left the log on the price variable.

```
# The 1st model I was happy with
test_model_4 <- lm(log(price) ~ table + x + clarity, data = diamonds_sample)

# The 2nd model I was happy with
test_model_7 <- lm(log(price) ~ table + x + clarity + cut, data = diamonds_sample)
```

After realizing that my best model would come from the transformation of the price variable, I experimented with around 7 different models to try and find the best one. After all of them I ended up with two models that I was pretty happy with. The models were extremely close to each other when it came to their R-Squared values with 1 being 0.8696 and the second one being 0.8697. At this point I can say that the model has multicollinearity, because of how lowly it originally aligned with the first two variables to now how it is a good model because of the additional variables added.

In this part I found it interesting that with the two models I ended up choosing that their R-Squared values were so incredibly close to each other. I also thought it was interesting that although the first model was just barely lower it had one less variable than my second one. I would have thought that by adding the cut varibale to the model it would have done a little bit more to effect the model, but in this case it didn't.

Part 3 - Goal

```
test_model_4 <- lm(log(price) ~ table + x + clarity, data = diamonds_sample)
summary(test_model_4)
```

```
##
## Call:
## lm(formula = log(price) ~ table + x + clarity, data = diamonds_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4399 -0.1773 -0.0032  0.1571  6.3879
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.800953   0.520033   3.463 0.000581 ***
## table       0.004773   0.008245   0.579 0.562947
## x           0.852563   0.015693  54.329  < 2e-16 ***
## clarityIF   1.082356   0.191608   5.649 2.74e-08 ***
## claritySI1  0.801843   0.161539   4.964 9.56e-07 ***
## claritySI2  0.609525   0.162582   3.749 0.000199 ***
## clarityVS1  0.886594   0.164836   5.379 1.16e-07 ***
## clarityVS2  0.825438   0.161856   5.100 4.87e-07 ***
## clarityVVS1 1.190079   0.174740   6.811 2.86e-11 ***
## clarityVVS2 1.024728   0.168862   6.068 2.59e-09 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.3819 on 490 degrees of freedom
## Multiple R-squared:  0.8696, Adjusted R-squared:  0.8672
## F-statistic: 363.2 on 9 and 490 DF,  p-value: < 2.2e-16
```

```r
test_model_7 <- lm(log(price) ~ table + x + clarity + cut, data = diamonds_sample)
summary(test_model_7)
```

```
##
## Call:
## lm(formula = log(price) ~ table + x + clarity + cut, data = diamonds_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4067 -0.1803 -0.0026  0.1562  6.3829
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.790824   0.624199   2.869   0.0043 **
## table        0.004256   0.009968   0.427   0.6696
## x            0.853247   0.015790  54.037  < 2e-16 ***
## clarityIF    1.065802   0.194540   5.479 6.89e-08 ***
## claritySI1   0.788007   0.164646   4.786 2.26e-06 ***
## claritySI2   0.591806   0.166025   3.565   0.0004 ***
## clarityVS1   0.870427   0.167984   5.182 3.23e-07 ***
## clarityVS2   0.807881   0.165138   4.892 1.36e-06 ***
## clarityVVS1  1.173137   0.178046   6.589 1.15e-10 ***
## clarityVVS2  1.008775   0.172163   5.859 8.57e-09 ***
## cutGood      0.042888   0.108907   0.394   0.6939
## cutIdeal     0.050346   0.101879   0.494   0.6214
## cutPremium   0.062581   0.101464   0.617   0.5377
## cutVery Good 0.054623   0.101645   0.537   0.5912
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.3833 on 486 degrees of freedom
## Multiple R-squared:  0.8697, Adjusted R-squared:  0.8663
## F-statistic: 249.6 on 13 and 486 DF,  p-value: < 2.2e-16
```

Analyze how it works and what you observe.

```r
AIC(test_model_4)
```

```
## [1] 468.3468
```

```r
AIC(test_model_7)
```

```
## [1] 475.9145
```

```r
BIC(test_model_4)
```

```
## [1] 514.7075
```

```r
BIC(test_model_7)
```

```
## [1] 539.1336
```

```r
final_model <- lm(log(price) ~ table + x + clarity, data = diamonds_sample)
summary(final_model)
```

```
## 
## Call:
## lm(formula = log(price) ~ table + x + clarity, data = diamonds_sample)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -1.4399 -0.1773 -0.0032  0.1571  6.3879 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 1.800953   0.520033   3.463 0.000581 ***
## table       0.004773   0.008245   0.579 0.562947    
## x           0.852563   0.015693  54.329  < 2e-16 ***
## clarityIF   1.082356   0.191608   5.649 2.74e-08 ***
## claritySI1  0.801843   0.161539   4.964 9.56e-07 ***
## claritySI2  0.609525   0.162582   3.749 0.000199 ***
## clarityVS1  0.886594   0.164836   5.379 1.16e-07 ***
## clarityVS2  0.825438   0.161856   5.100 4.87e-07 ***
## clarityVVS1 1.190079   0.174740   6.811 2.86e-11 ***
## clarityVVS2 1.024728   0.168862   6.068 2.59e-09 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 0.3819 on 490 degrees of freedom
## Multiple R-squared:  0.8696, Adjusted R-squared:  0.8672 
## F-statistic: 363.2 on 9 and 490 DF,  p-value: < 2.2e-16
```

Document the observations of finding the best model

```
# Give CIs for a mean predicted value and the PIs of a future predicted value for at least one combinat
```

Summarize the report.