# Linear Regression Analysis on Diamond Prices

Jimmy Wu

2025-05-25

## Introduction

Diamonds have long held a prominent place in the luxury industry, symbolizing wealth, elegance, and commitment. As the centerpiece of engagement rings and jewelry, their market value is not only driven by intrinsic physical characteristics such as size, clarity, and cut, but also by branding, consumer perception, and global demand trends. Given their economic and emotional significance, understanding what truly drives diamond pricing is critical for both consumers seeking value and sellers aiming for competitive pricing strategies. By statistically modeling the relationship between diamond features and their prices, we can conclude patterns and support more informed decision-making.

In this project, we investigate the factors that influence diamond pricing by analyzing a dataset of 2,000 randomly sampled diamonds. The primary goal is to explore and model the relationship between various physical and categorical characteristics of diamonds and their market prices. Given the The project adopts a structured modeling approach that progresses from descriptive statistics to simple and multiple regression analyses, and finally to model refinement and validation.

## Method

We begin by performing an exploratory data analysis to understand the distribution of each variable and detect any patterns or anomalies. Histograms and bar plots are used to assess the shape and spread of continuous and categorical variables respectively, while a correlation matrix helps identify linear relationships that may influence model selection later. This initial step reveals strong correlations between diamond size metrics (carat, x, y, z) and price, while also pointing out potential multicollinearity issues among the size dimensions.

Next, we build and evaluate a simple linear regression model using carat as the sole predictor of price, followed by diagnostic checks of model assumptions. After detecting violations of normality and heteroscedasticity, we apply logarithmic transformations, which substantially improve model performance and validity. The model is then extended by testing additional predictors, with selection guided by improvements in adjusted $R^2$ and reductions in residual error.

Finally, we apply stepwise AIC-based model selection to determine the most parsimonious model, followed by variance inflation factor (VIF) checks to address multicollinearity. The refined model is used to generate confidence and prediction intervals for a new data point, offering insights into both the expected average price and variability of individual diamond prices. Altogether, this project provides a thorough exploration of regression modeling techniques while delivering practical findings about diamond price determinants.

# Results

## Part - 1: Data Description and Descriptive Statistics

We start by read in the data and randomly select 2000 observations.

```
d_data <- read.csv("Diamonds Prices2022.csv")

set.seed(5252025)
d_rs <- d_data[sample(nrow(d_data), 2000),]
```

We first need to identify the continuous random variable of interest, using `str` function.

```
summary(d_rs)[,2]
```

```
##
## "Min.   :0.2100  " "1st Qu.:0.4100  " "Median :0.7100  " "Mean   :0.8054  "
##
## "3rd Qu.:1.0425  " "Max.   :3.0000  "
```

```
kable(summary(d_rs), caption = "Summary of Each Diamond Variable")
```

Table 1: Summary of Each Diamond Variable

| X | carat | cut | color | clarity | depth | table | price | x | y | z |
|---|---|---|---|---|---|---|---|---|---|---|
| Min. : 97 | Min. :0.2100 | Length:2000 | Length:2000 | Length:2000 | Min. :55.20 | Min. :52.00 | Min. : 368 | Min. :3.850 | Min. :3.840 | Min. :0.000 |
| 1st Qu.:13584 | 1st Qu.:0.4100 | Class :character | Class :character | Class :character | 1st Qu.:61.00 | 1st Qu.:56.00 | 1st Qu.: 1015 | 1st Qu.:4.770 | 1st Qu.:4.780 | 1st Qu.:2.950 |
| Median :27068 | Median :0.7100 | Mode :character | Mode :character | Mode :character | Median :61.80 | Median :57.00 | Median : 2538 | Median :5.720 | Median :5.740 | Median :3.540 |
| Mean :27150 | Mean :0.8054 | NA | NA | NA | Mean :61.76 | Mean :57.38 | Mean : 3910 | Mean :5.764 | Mean :5.767 | Mean :3.559 |
| 3rd Qu.:41246 | 3rd Qu.:1.0425 | NA | NA | NA | 3rd Qu.:62.50 | 3rd Qu.:59.00 | 3rd Qu.: 5392 | 3rd Qu.:6.530 | 3rd Qu.:6.540 | 3rd Qu.:4.040 |
| Max. :53921 | Max. :3.0000 | NA | NA | NA | Max. :69.50 | Max. :68.00 | Max. :18797 | Max. :9.320 | Max. :9.190 | Max. :5.500 |

Through the `str` function, we identified the following as the continuous random variables of interest: carat, depth, table, price, x, y, and z. We now create summary statistics and a series of histograms for these random variables.

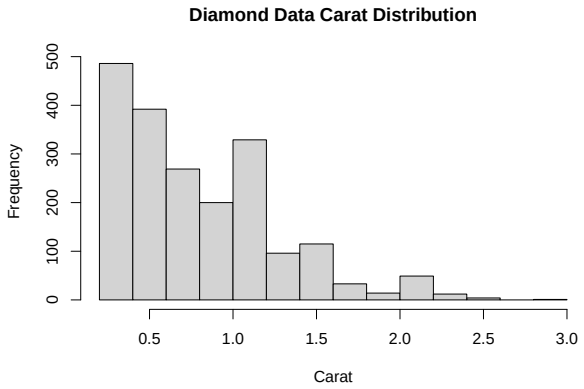
```
# Summary and Structure of the Dataset
kable(summary(d_rs)[, 2:6])
```

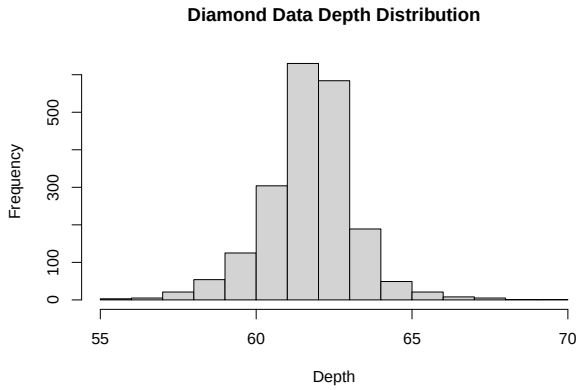
| carat | cut | color | clarity | depth |
|---|---|---|---|---|
| Min. :0.2100 | Length:2000 | Length:2000 | Length:2000 | Min. :55.20 |
| 1st Qu.:0.4100 | Class :character | Class :character | Class :character | 1st Qu.:61.00 |
| Median :0.7100 | Mode :character | Mode :character | Mode :character | Median :61.80 |
| Mean :0.8054 | NA | NA | NA | Mean :61.76 |
| 3rd Qu.:1.0425 | NA | NA | NA | 3rd Qu.:62.50 |
| Max. :3.0000 | NA | NA | NA | Max. :69.50 |

```
kable(summary(d_rs)[, 7:11])
```

| table | price | x | y | z |
|---|---|---|---|---|
| Min. :52.00 | Min. : 368 | Min. :3.850 | Min. :3.840 | Min. :0.000 |
| 1st Qu.:56.00 | 1st Qu.: 1015 | 1st Qu.:4.770 | 1st Qu.:4.780 | 1st Qu.:2.950 |
| Median :57.00 | Median : 2538 | Median :5.720 | Median :5.740 | Median :3.540 |
| Mean :57.38 | Mean : 3910 | Mean :5.764 | Mean :5.767 | Mean :3.559 |
| 3rd Qu.:59.00 | 3rd Qu.: 5392 | 3rd Qu.:6.530 | 3rd Qu.:6.540 | 3rd Qu.:4.040 |
| Max. :68.00 | Max. :18797 | Max. :9.320 | Max. :9.190 | Max. :5.500 |

```
# Histogram for Carat and Depth
hist(d_rs$carat,
     main = "Diamond Data Carat Distribution",
     xlab = "Carat")

hist(d_rs$depth,
     main = "Diamond Data Depth Distribution",
     xlab = "Depth")
```



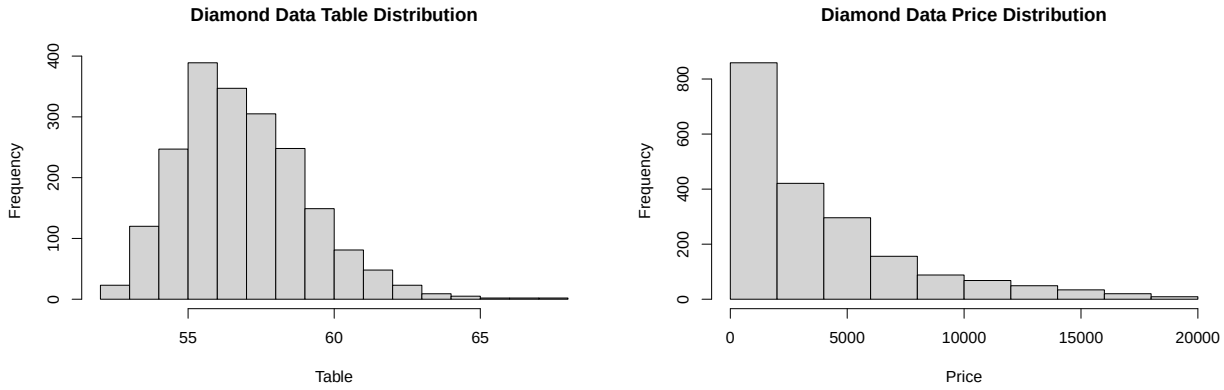




The Carat distribution appears to be not normal, with a heavy skew to the right. Most of the data lies around $< 1.25$ carat range, with a small amount of the other falls above 1.25 carat, therefore constitutes the right skewness.

The Depth distribution, however, appears to be normal, centering around the value of 62.5. Although there are slightly more data falling before 62.5, the overall shape of the distribution still appears to be bell-shaped.

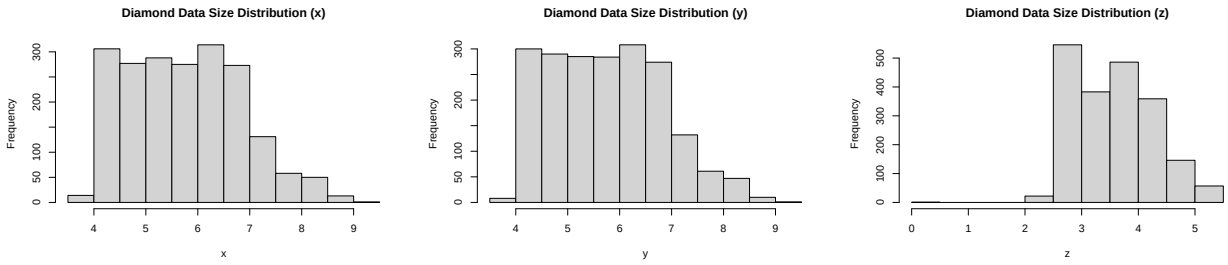
```
# Histogram for Table and Price
hist(d_rs$table,
     main = "Diamond Data Table Distribution",
```

```
     xlab = "Table")

hist(d_rs$price,
     main = "Diamond Data Price Distribution",
     xlab = "Price")
```




The table distribution appears to be generally normal with a slight right skewness, centering around the value of 56. The price distribution, however, appears to be not normal, with a extremely heavy skew to the right.

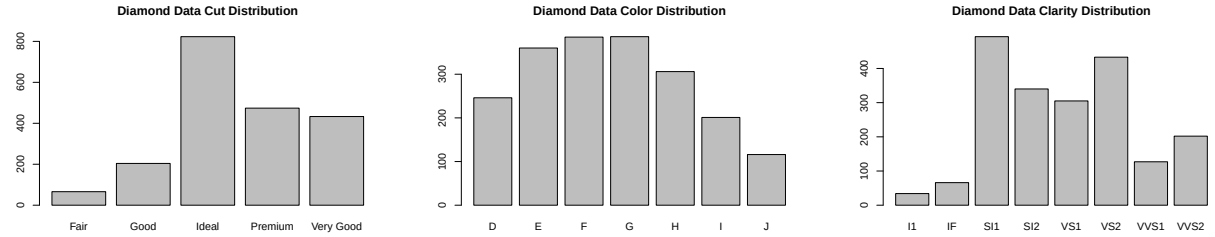
```
# Histogram for Diamond Data Size x, y and z
hist(d_rs$x,
     main = "Diamond Data Size Distribution (x)",
     xlab = "x")

hist(d_rs$y,
     main = "Diamond Data Size Distribution (y)",
     xlab = "y")

hist(d_rs$z,
     main = "Diamond Data Size Distribution (z)",
     xlab = "z")
```




The x, y, and z variables' distributions are unimodal. Specifically, both x and y look roughly bell-shaped and skews to the right, and z has a strongly right-skewed.

As of the categorical variables, the model has three: cut, color, and clarity. We now create a series of barplots for them.

```r
barplot(table(d_rs$cut),
  main="Diamond Data Cut Distribution"
  )

barplot(table(d_rs$color),
  main="Diamond Data Color Distribution"
  )

barplot(table(d_rs$clarity),
        main="Diamond Data Clarity Distribution"
        )
```




As of the cut distribution, it is skewed to the left. In context, we may interpret as most of the diamond in the data set are cut "ideally".

As of the color distribution, it is skewed to the right. In context, we may interpret that the peak of the dataset is color E~G.

Lastly, clarity peaks at the SI1 level and less at the I1 level. We can see two obvious peaks in SI1 and VS2, so the distribution is bimodal (two modes).

We now determine if there is any correlation between these variables.

```r
# The input of the `cor` function has to be numeric
# Thus, we first remove the three categorical variables in the model.
d_rs_num <- d_rs[,-c(3,4,5)] # index 3,4,5 is cut, color, and clarity, correspondingly

kable(summary(cor(d_rs_num)))
```

| X | carat | depth | table | price | x | y | z |
|---|---|---|---|---|---|---|---|
| Min. :-0.4305 | Min. :-0.4026 | Min. :-0.30767 | Min. :-0.30767 | Min. :-0.3330 | Min. :-0.4291 | Min. :-0.4305 | Min. :-0.4278 |
| 1st Qu.:-0.4281 | 1st Qu.: 0.1674 | 1st Qu.:-0.03739 | 1st Qu.: 0.07681 | 1st Qu.: 0.1028 | 1st Qu.: 0.1666 | 1st Qu.: 0.1609 | 1st Qu.: 0.1622 |
| Median :-0.3678 | Median : 0.9458 | Median :-0.01053 | Median : 0.19772 | Median : 0.8831 | Median : 0.9331 | Median : 0.9335 | Median : 0.9241 |
| Mean :-0.1473 | Mean : 0.5864 | Mean : 0.08928 | Mean : 0.19776 | Mean : 0.5483 | Mean : 0.5777 | Mean : 0.5765 | Mean : 0.5833 |
| 3rd Qu.:-0.0871 | 3rd Qu.: 0.9779 | 3rd Qu.: 0.04984 | 3rd Qu.: 0.22636 | 3rd Qu.: 0.8975 | 3rd Qu.: 0.9863 | 3rd Qu.: 0.9862 | 3rd Qu.: 0.9820 |
| Max. : 1.0000 | Max. : 1.0000 | Max. : 1.00000 | Max. : 1.00000 | Max. : 1.0000 | Max. : 1.0000 | Max. : 1.0000 | Max. : 1.0000 |

From the correlation matrix, we can observe several strong linear relationships among the variables. Most notably, carat is highly correlated with price (correlation of around 0.92) and also with the physical dimensions

x, y, and z (correlation all above 0.96), which makes sense as larger diamonds tend to cost more.

Similarly, price is also strongly correlated with x, y, and z (correlation all around 0.88 or higher). On the other hand, depth and table show weak correlation with most other variables.The extremely high correlation among x, y, and z (correlation above 0.98) suggests potential multicollinearity.

Overall, the matrix indicates that carat and physical size are the primary drivers of price, while depth and table play a smaller role.

With an understanding of the model's variables, we now run the multiple linear regression model using all these variables and observe the summary statistics.

```
model1 <- lm(price ~ ., data = d_rs)
model1_summary <- summary(model1)

coefs <- as.data.frame(model1_summary$coefficients)
colnames(coefs)[4] <- "Pr"
coefs$Term <- rownames(coefs)
rownames(coefs) <- NULL

rsq <- model1_summary$r.squared
adj_rsq <- model1_summary$adj.r.squared
sigma <- model1_summary$sigma
f_stat <- model1_summary$fstatistic[1]
df1 <- model1_summary$fstatistic[2]
df2 <- model1_summary$fstatistic[3]

overview <- data.frame(
  `R-squared` = rsq,
  `Adjusted R-squared` = adj_rsq,
  `Residual Std. Error` = sigma,
  `F-statistic` = f_stat,
  `df1 (num)` = df1,
  `df2 (den)` = df2
)

kable(coefs, caption = "Model Coefficients", digits = 5, escape = FALSE)
```

Table 5: Model Coefficients

| Estimate | Std. Error | t value | Pr | Term |
|---|---|---|---|---|
| 4173.93129 | 2197.82383 | 1.89912 | 0.05769 | (Intercept) |
| 0.00382 | 0.00168 | 2.26942 | 0.02335 | X |
| 11548.77058 | 269.39546 | 42.86921 | 0.00000 | carat |
| 622.81594 | 159.62301 | 3.90179 | 0.00010 | cutGood |
| 858.93205 | 160.75124 | 5.34324 | 0.00000 | cutIdeal |
| 787.32292 | 153.11464 | 5.14205 | 0.00000 | cutPremium |
| 729.34639 | 157.63730 | 4.62674 | 0.00000 | cutVery Good |
| -107.82929 | 87.65360 | -1.23018 | 0.21878 | colorE |
| -270.50420 | 87.14486 | -3.10408 | 0.00194 | colorF |
| -430.94856 | 88.01112 | -4.89652 | 0.00000 | colorG |
| -802.19414 | 92.47180 | -8.67501 | 0.00000 | colorH |
| -1338.04215 | 103.80026 | -12.89055 | 0.00000 | colorI |
| -2227.14926 | 123.68098 | -18.00721 | 0.00000 | colorJ |

| Estimate | Std. Error | t value | Pr | Term |
|---|---|---|---|---|
| 5049.14391 | 233.09802 | 21.66103 | 0.00000 | clarityIF |
| 3508.80055 | 194.98268 | 17.99545 | 0.00000 | claritySI1 |
| 2626.08330 | 195.46320 | 13.43518 | 0.00000 | claritySI2 |
| 4333.29668 | 200.10146 | 21.65550 | 0.00000 | clarityVS1 |
| 4187.12693 | 195.58077 | 21.40868 | 0.00000 | clarityVS2 |
| 4736.71331 | 214.94822 | 22.03653 | 0.00000 | clarityVVS1 |
| 4795.98234 | 205.60959 | 23.32567 | 0.00000 | clarityVVS2 |
| -65.19061 | 25.18454 | -2.58852 | 0.00971 | depth |
| -51.08660 | 14.74975 | -3.46356 | 0.00054 | table |
| -1884.80873 | 530.29281 | -3.55428 | 0.00039 | x |
| 602.91765 | 529.16544 | 1.13937 | 0.25469 | y |
| 175.51906 | 253.97185 | 0.69110 | 0.48959 | z |

```
kable(overview, caption = "Model Fit Statistics", digits = 5, escape = FALSE)
```

Table 6: Model Fit Statistics

| | R.squared | Adjusted.R.squared | Residual.Std..Error | F.statistic | df1..num. | df2..den. |
|---|---|---|---|---|---|---|
| value | 0.92505 | 0.92414 | 1050.645 | 1015.633 | 24 | 1975 |

The output consists of two tables summarizing the results of our `model1`. Table 5 shows the estimated effects of each predictor variable on `price`. Each row represents a predictor, either a continuous variable like `carat` or a dummy variable derived from categorical variables such as `cut`, `color`, and `clarity`. For instance, the estimated coefficient for `carat` is 11548.771, which suggests that for each additional `carat` (with all other factors held constant), the diamond's price increases by approximately $11,549. The Std. Error column indicates the variability of these estimates; for `carat`, the standard error is 269.395. The t value and Pr columns test the statistical significance of each coefficient. Variables such as `clarityVS2` ($t = 21.409$, $p < 0.001$) and `cutIdeal` ($t = 5.343$, $p < 0.001$) are highly significant, while others like `colorE` ($p = 0.219$) or `z` ($p = 0.490$) do not show statistical significance and may not meaningfully contribute to the model.

Table 6 summarizes the overall performance of the model. The R-squared value is 0.925, meaning that 92.5% of the variability in diamond prices is explained by the predictors included in the model. The Adjusted R-squared, which adjusts for the number of predictors, is slightly lower at 0.924, suggesting the model is not overfitted. The Residual Standard Error is 1050.645, representing the average deviation between observed and predicted prices. Additionally, the F-statistic is 1015.633 with degrees of freedom 24 and 1975, respectively, indicating that the model as a whole is highly significant. Together, these outputs confirm that the regression model fits the data well, with most predictors contributing meaningfully, though a few may warrant reconsideration or removal in further refinement.

In Part 1 of the project, several patterns and unexpected findings emerged.

First, the distribution of the continuous variables are well within the intuitive expectations. Carat and price, for example, were heavily right-skewed, which makes sense given that large diamonds are rarer and more expensive. The distributions for x, y, and z (which represent physical size dimensions) were mostly normal but showed signs of skewness, suggesting a general consistency in shape but with some outliers or measurement variability. The depth and table variables showed relatively normal distributions with only little skew, indicating that most diamonds fall within standard proportions.

For the categorical variables, some results were slightly surprising. The color variable showed a right-skewed distribution centered on F, which was not necessarily expected-this suggests that F is a particularly popular or available color grade.

Correlation analysis revealed very strong positive correlations between carat and price, as well as among the size dimensions x, y, and z. This was expected as, again, larger diamonds typically have higher prices. However, the strength of the correlations (all above 0.9) was particularly striking. These strong linear relationships suggest that carat and physical dimensions are closely tied to the value of a diamond. Conversely, depth and table had weak correlations with price and other variables, indicating their limited influence in pricing, at least linearly.

Overall, while some results (such as the skewed distributions and high correlations) were expected, the exact strength and structure of these relationships, especially the extreme multicollinearity among x, y, and z, were not really anticipated.

## PART - 2: Simple Linear Regression for Diamond Data

We now make a simple linear regression model before finding the optimal model.

```
model2 <- lm(price ~ carat, data = d_rs)
model2_summary <- summary(model2)

coefs <- as.data.frame(model2_summary$coefficients)
colnames(coefs)[4] <- "Pr"
coefs$Term <- rownames(coefs)
rownames(coefs) <- NULL

rsq <- model2_summary$r.squared
adj_rsq <- model2_summary$adj.r.squared
sigma <- model2_summary$sigma
f_stat <- model2_summary$fstatistic[1]
df1 <- model2_summary$fstatistic[2]
df2 <- model2_summary$fstatistic[3]

overview <- data.frame(
  `R-squared` = rsq,
  `Adjusted R-squared` = adj_rsq,
  `Residual Std. Error` = sigma,
  `F-statistic` = f_stat,
  `df1 (num)` = df1,
  `df2 (den)` = df2
)

kable(coefs, caption = "Model Coefficients", digits = 4, escape = FALSE)
```

Table 7: Model Coefficients

| Estimate | Std. Error | t value | Pr | Term |
|---|---|---|---|---|
| -2245.921 | 66.5019 | -33.7723 | 0 | (Intercept) |
| 7643.867 | 71.6980 | 106.6120 | 0 | carat |

```
kable(overview, caption = "Model Fit Statistics", digits = 4, escape = FALSE)
```

Table 8: Model Fit Statistics

| | R.squared | Adjusted.R.squared | Residual.Std..Error | F.statistic | df1..num. | df2..den. |
|---|---|---|---|---|---|---|
| value | 0.8505 | 0.8504 | 1475.291 | 11366.11 | 1 | 1998 |

In this simple linear regression model, carat was chosen as the predictor variable and price as the response variable due to the fact that a diamond's weight (measured in carats) largely impacts its market value. Therefore this decision is a logical starting point for comprehending how the characteristics and features of a diamond impact its market price.

The linear regression model in turn reveals that there is a strong positive relationship between carat and price. The r-squared value of 0.8505 reveals that there is approximately 85% of the variation in price that is due to the carat. Therefore carat is a strong predictor for the price.

We also create a confidenec interval and a prediction interval for the purpose of understanding.

```
# We use traditional significance level of 0.05
conf_df <- as.data.frame(confint(model2, level = 0.95))
conf_df$Term <- rownames(conf_df)
rownames(conf_df) <- NULL

kable(conf_df, caption = "95% Confidence Intervals for Model Coefficients")
```

Table 9: 95% Confidence Intervals for Model Coefficients

| 2.5 % | 97.5 % | Term |
|---|---|---|
| -2376.341 | -2115.500 | (Intercept) |
| 7503.256 | 7784.477 | carat |

The 95% confidence interval for the slope (coefficient of carat) is (7503.256, 7784.477). This means we are 95% confident that, in the population, each additional carat increases the average diamond price by between 7,503.26 and 7,784.48.

The 95% confidence interval for the intercept is (-2376.341, -2115.500). This indicates that, when carat = 0, the expected price would be between about -2376 and -2115.

```
new_data <- data.frame(carat = 0.5)

pred_df <- as.data.frame(predict(model2, newdata = new_data, interval = "prediction", level = 0.95))

kable(pred_df, caption = "95% Prediction Interval for a New Diamond (carat = 0.5)")
```

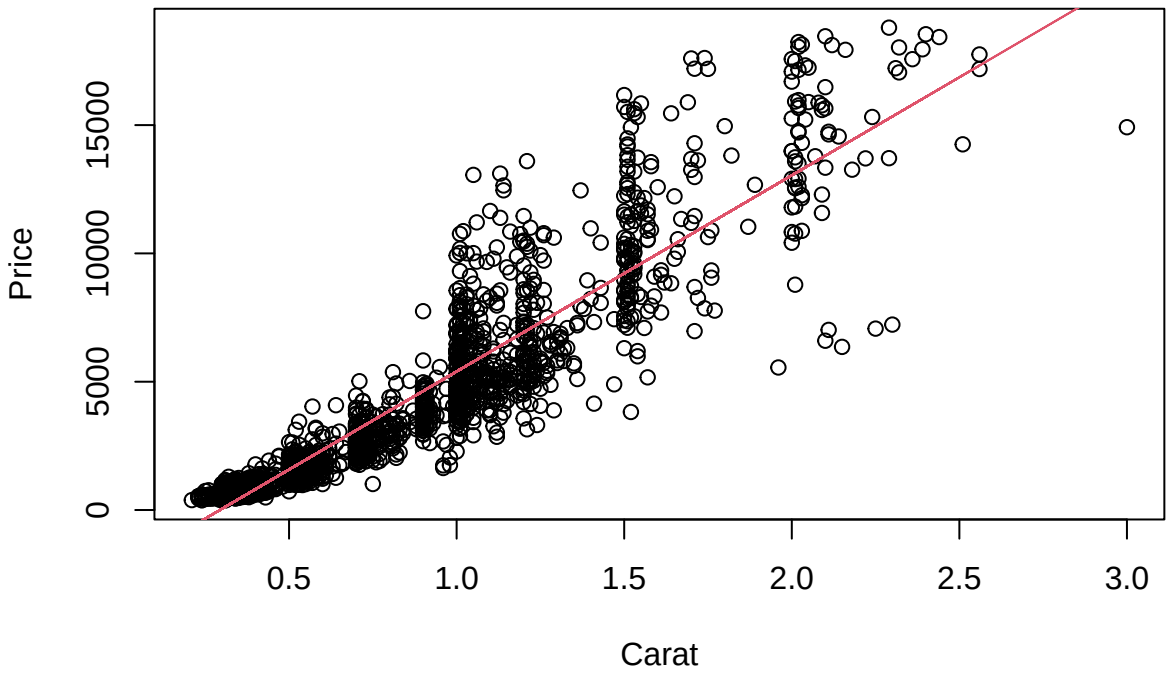
Table 10: 95% Prediction Interval for a New Diamond (carat = 0.5)

| fit | lwr | upr |
|---|---|---|
| 1576.013 | -1318.3 | 4470.325 |

For a new diamond with carat = 0.5, the 95% prediction interval is (-1318.3, 4470.325). This means that, based on the model, we expect the price of a single new 0.5-carat diamond to fall between -1318.3 and 4470.325 with 95% confidence.

At this point, to get an overview of how well the model fits, we plot the fitted value.

```
plot(d_rs$carat, d_rs$price, xlab="Carat", ylab="Price")
lines(d_rs$carat, model2$fitted.values, col=2)
```



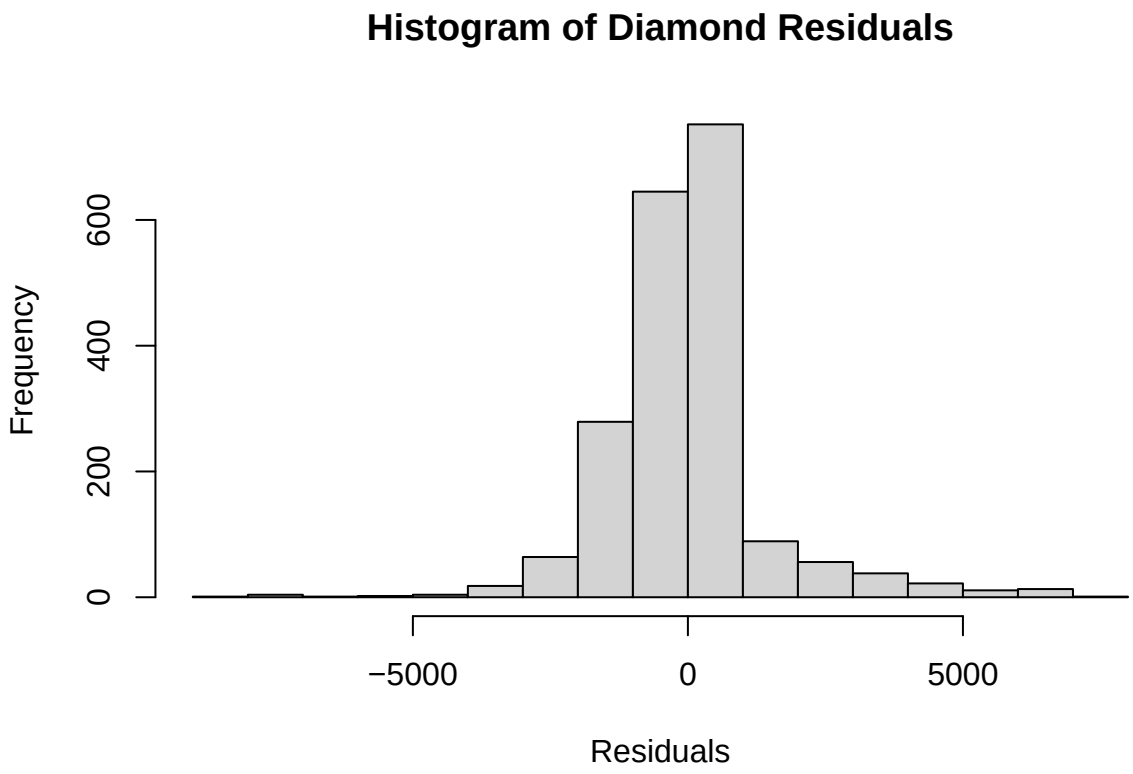

The scatterplot indicates a clear positive relationship between carat and price. However, there is also a spread around the fitted line, especially for higher carat values. This suggests that while carat is a strong predictor of price, there is substantial variability that cannot be explained by carat alone.

Therefore, to ensure that our test results are reliable, we now conduct a model checking on the following three assumptions on linear model:

1. **Normality of the Data**
2. **No structure to the Data**
3. **Equal Variances across Fitted Values**

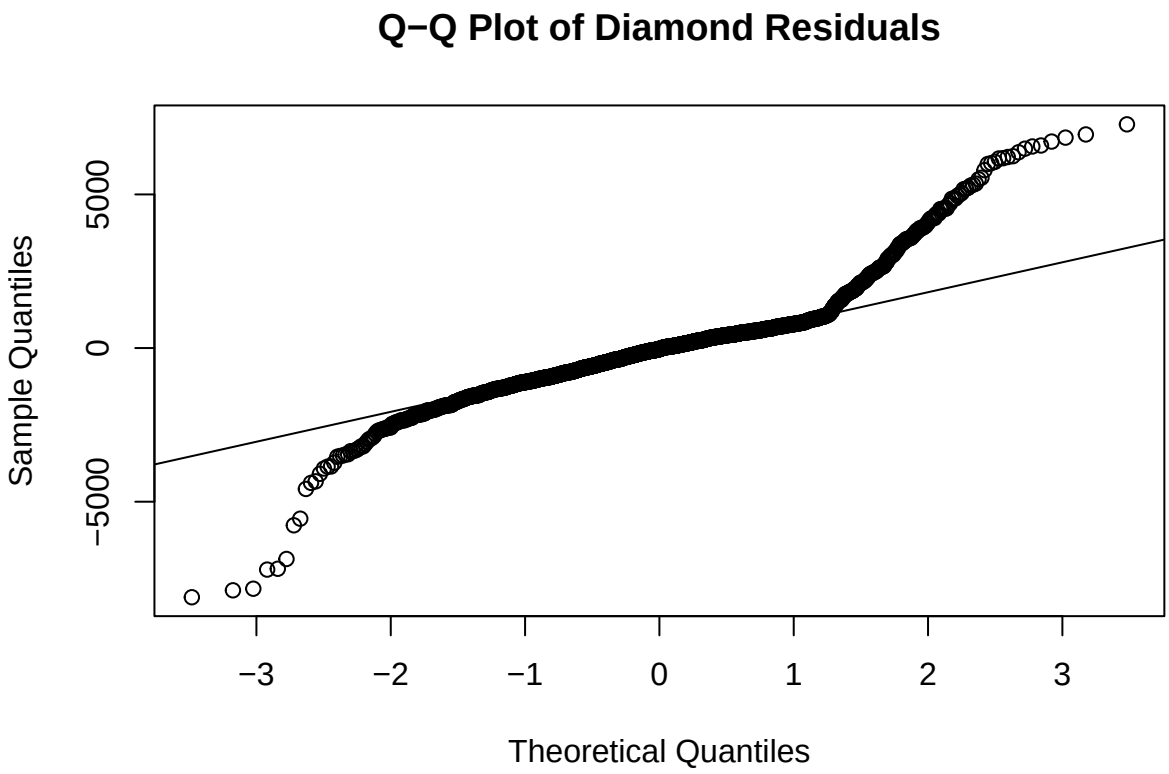
### Normality of the Data

```
# Histogram of the Residuals
hist(model2$residuals, main = "Histogram of Diamond Residuals", xlab = "Residuals")
```




According to the histogram, the residuals of the linear model looks approximately bell-shaped, with a slight skewness to the right. Therefore, there should be little to no concerns about the normality of the data.

```
# Q-Q Plot
qqnorm(model2$residuals, main = "Q-Q Plot of Diamond Residuals")
qqline(model2$residuals)
```




According to the Q-Q plot, the residual data points lies close to the regression line from -2 to 1 quantiles, but heavily deviates from the regression line before -2 and after 1 quantiles. This suggests potential deviation from normal distribution, so we proceed to verify the normality by conducting Shapiro-Wilk Normality Test.

```
# Shapiro-Wilk Normality Test
shapiro_result <- shapiro.test(model2$residuals)

shapiro_df <- data.frame(
  Statistic = signif(shapiro_result$statistic, 5),
  P_Value = shapiro_result$p.value
)

kable(shapiro_df, caption = "Shapiro-Wilk Normality Test for Diamond Model Residuals")
```

Table 11: Shapiro-Wilk Normality Test for Diamond Model Residuals

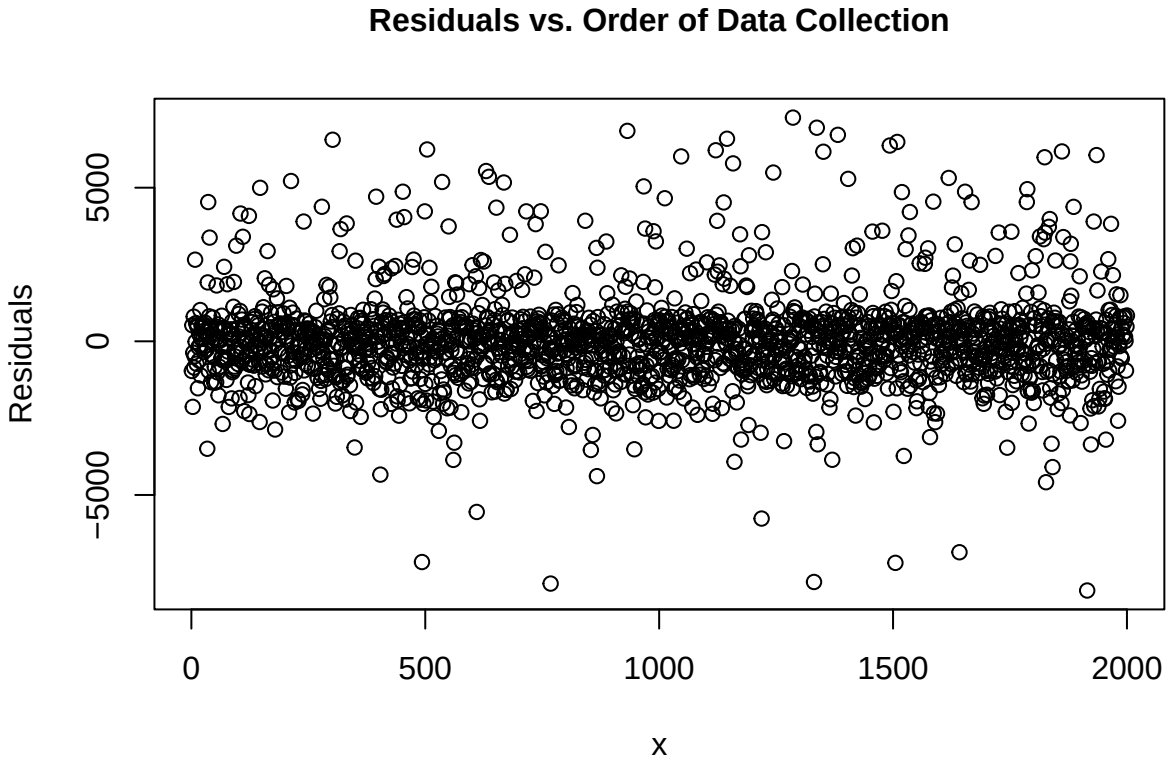
| | Statistic | P_Value |
|---|---|---|
| W | 0.88859 | 0 |

According to the Shapiro-Wilk Normality Test, we derive a p-value of $9.7705 \times 10^{-36}$ at $\alpha = 0.05$, which is very close to 0 as rounded in the table. Therefore, we reject the null hypothesis of the test that our data follows a normal distribution and conclude that our data does not follows a normal distribution.

All three approaches we used to examine the normality of our data - histogram, Q-Q plot, and shapiro-wilk normality test - indicates contradicting conclusions. While the no structure to the data assumption is met, the Q-Q plot indicates deviation from normality, which was later confirmed by the Shapiro-Wilk Normality Test. Thus, we conclude that the normality assumption of the test is violated and therefore **the normality assumption is not met**.

### Structure of the Data

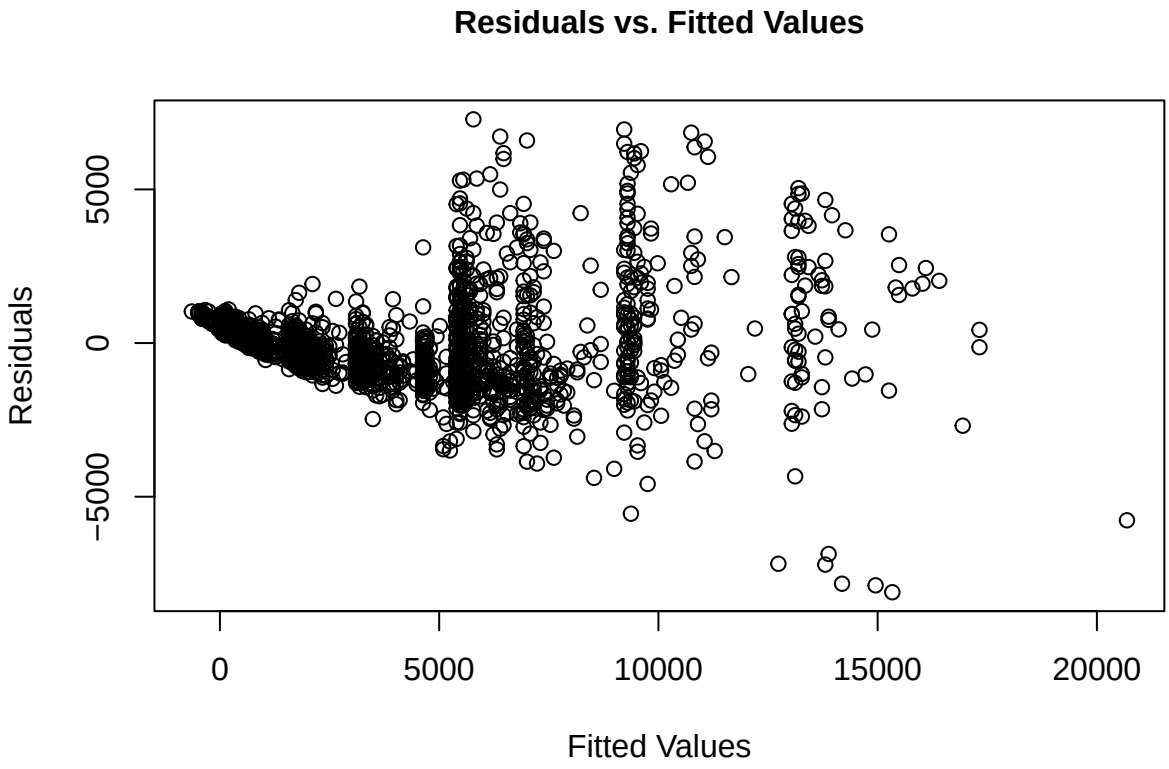
```
x <- 1:length(model2$residuals)

plot(model2$residuals ~ x, ylab="Residuals", cex.lab=1,
  main="Residuals vs. Order of Data Collection", cex.main=1)
```




The residuals' value, across the order of data collections, does not seems to go up or down. Therefore, there are no apparent patterns in the plot and so the **no structure to the data assumption is met**.

### Equal Variances across Fitted Values

```
plot(model2$residuals ~ model2$fitted.values,
  xlab="Fitted Values", ylab="Residuals", cex.lab=1,
  main="Residuals vs. Fitted Values", cex.main=1)
```




According to the plot, variances across different fitted values seems to be speading out as the fitted values increase. Therefore, the **equal variance across fitted value assumption is not met**.

Given that, across all three linear model test assumptions, two of them (normality and equal variances) are not met, transformation of random variable is necessary.

```
model2_log <- lm(log(price) ~ log(carat), data = d_rs) # Apply log transformation
model2_log_summary <- summary(model2_log)

coefs <- as.data.frame(model2_log_summary$coefficients)
colnames(coefs)[4] <- "Pr"
coefs$Term <- rownames(coefs)
rownames(coefs) <- NULL

rsq <- model2_log_summary$r.squared
adj_rsq <- model2_log_summary$adj.r.squared
sigma <- model2_log_summary$sigma
f_stat <- model2_log_summary$fstatistic[1]
df1 <- model2_log_summary$fstatistic[2]
df2 <- model2_log_summary$fstatistic[3]

overview <- data.frame(
  `R-squared` = rsq,
  `Adjusted R-squared` = adj_rsq,
  `Residual Std. Error` = sigma,
  `F-statistic` = f_stat,
```

```
  `df1 (num)` = df1,
  `df2 (den)` = df2
)

kable(coefs, caption = "Model Coefficients", digits = 4, escape = FALSE)
```

Table 12: Model Coefficients

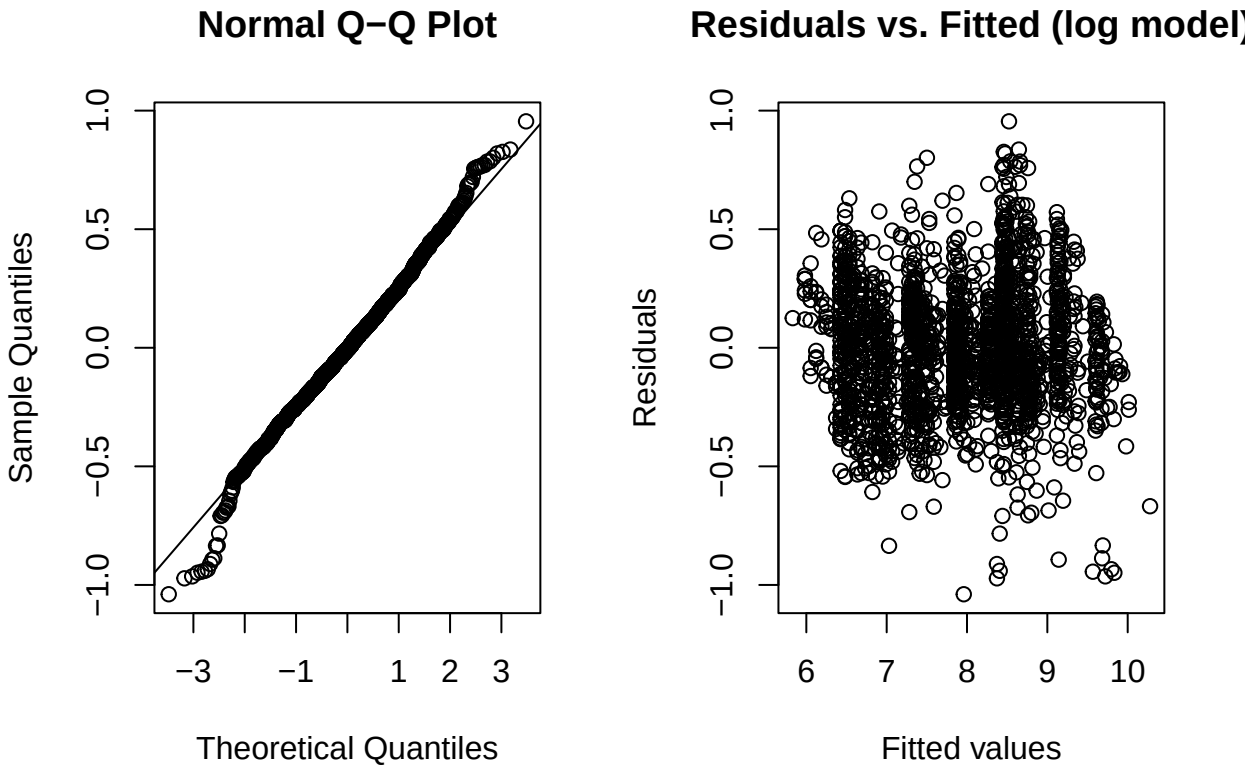
| Estimate | Std. Error | t value | Pr | Term |
|---|---|---|---|---|
| 8.4410 | 0.0071 | 1193.7187 | 0 | (Intercept) |
| 1.6725 | 0.0104 | 160.3844 | 0 | log(carat) |

```
kable(overview, caption = "Model Fit Statistics", digits = 4, escape = FALSE)
```

Table 13: Model Fit Statistics

| | R.squared | Adjusted.R.squared | Residual.Std..Error | F.statistic | df1..num. | df2..den. |
|---|---|---|---|---|---|---|
| value | 0.9279 | 0.9279 | 0.2638 | 25723.16 | 1 | 1998 |

```
par(mfrow = c(1, 2))
# Normality Plot
qqnorm(model2_log$residuals)
qqline(model2_log$residuals)

# Equal Variances
plot(model2_log$fitted.values, model2_log$residuals,
     main = "Residuals vs. Fitted (log model)",
     xlab = "Fitted values", ylab = "Residuals")
```




After transforming both the independent and response variable, the summary for `model2_log` shows that the linear relationship between carat and the `(log) price` remains very strong. The estimated coefficient for `log(carat)` is now 1.672505, meaning that for each one-unit increase in carat, the expected log price increases by approximately 1.672505. This is still statistically significant with a p-value less than 2.2e-16.

R-squared increased from 0,842 to 0.9279, which indicate a great improve in fitness. More importantly, the residual standard error significantly decreased from 1475 to 0.2638, indicating improved model fit in terms of residual variability on the log scale.

The normality of the data was greatly improved, as we can see that most of the data align closely to the regression line. However, as of the equal variance assumption, it was also greatly improved by the transformation but still not perfect. We still see some minor deviations of the variances, but overall it should be satisfactory.

With the improved `model2_log` model, we now start adding more predictors to `model2_log` to see if it improves the explanatory power.

For our `model2_log`, the Residual standard error is 0.2638, Multiple R-squared is 0.9279, and Adjusted R-squared is 0.9279 as well.

We now run a full model where we assessed every combination of independent variables.

Table 14: Model Coefficients

| Estimate | Std. Error | t value | Pr | Term |
|---|---|---|---|---|
| -5.0048 | 0.2863 | -17.4834 | 0.0000 | (Intercept) |
| 0.0000 | 0.0000 | -3.3349 | 0.0009 | X |
| -1.1045 | 0.0351 | -31.4780 | 0.0000 | carat |
| 0.0629 | 0.0208 | 3.0233 | 0.0025 | cutGood |

| Estimate | Std. Error | t value | Pr | Term |
|---|---|---|---|---|
| 0.1511 | 0.0209 | 7.2167 | 0.0000 | cutIdeal |
| 0.1153 | 0.0199 | 5.7818 | 0.0000 | cutPremium |
| 0.1136 | 0.0205 | 5.5308 | 0.0000 | cutVery Good |
| -0.0572 | 0.0114 | -5.0071 | 0.0000 | colorE |
| -0.0887 | 0.0114 | -7.8159 | 0.0000 | colorF |
| -0.1476 | 0.0115 | -12.8771 | 0.0000 | colorG |
| -0.2446 | 0.0120 | -20.3094 | 0.0000 | colorH |
| -0.3701 | 0.0135 | -27.3734 | 0.0000 | colorI |
| -0.5205 | 0.0161 | -32.3081 | 0.0000 | colorJ |
| 1.0763 | 0.0304 | 35.4503 | 0.0000 | clarityIF |
| 0.5838 | 0.0254 | 22.9866 | 0.0000 | claritySI1 |
| 0.4163 | 0.0255 | 16.3504 | 0.0000 | claritySI2 |
| 0.7932 | 0.0261 | 30.4347 | 0.0000 | clarityVS1 |
| 0.7399 | 0.0255 | 29.0462 | 0.0000 | clarityVS2 |
| 1.0091 | 0.0280 | 36.0449 | 0.0000 | clarityVVS1 |
| 0.9173 | 0.0268 | 34.2537 | 0.0000 | clarityVVS2 |
| 0.0666 | 0.0033 | 20.3048 | 0.0000 | depth |
| 0.0125 | 0.0019 | 6.4810 | 0.0000 | table |
| 0.8375 | 0.0691 | 12.1253 | 0.0000 | x |
| 0.5455 | 0.0689 | 7.9146 | 0.0000 | y |
| 0.0788 | 0.0331 | 2.3833 | 0.0173 | z |

Table 15: Model Fit Statistics

| | R.squared | Adjusted.R.squared | Residual.Std..Error | F.statistic | df1..num. | df2..den. |
|---|---|---|---|---|---|---|
| value | 0.9808 | 0.9806 | 0.1368 | 4208.64 | 24 | 1975 |

We derived Residual standard error of 0.1368, Multiple R-squared of 0.9808, and Adjusted R-squared of 0.9806. This indicate that the full model might have a better fit than our `model2_log` model.

However, adding some but not all of the variables might hurt the model's explanatory power.

Table 16: Model Coefficients

| Estimate | Std. Error | t value | Pr | Term |
|---|---|---|---|---|
| 6.4974 | 0.2150 | 30.2206 | 0.0000 | (Intercept) |
| 2.0549 | 0.0192 | 107.2700 | 0.0000 | carat |
| -0.0227 | 0.0304 | -0.7491 | 0.4539 | colorE |
| 0.0251 | 0.0300 | 0.8388 | 0.4017 | colorF |
| -0.0514 | 0.0300 | -1.7135 | 0.0868 | colorG |
| -0.1936 | 0.0318 | -6.0952 | 0.0000 | colorH |
| -0.2884 | 0.0355 | -8.1152 | 0.0000 | colorI |
| -0.5133 | 0.0425 | -12.0736 | 0.0000 | colorJ |
| -0.0042 | 0.0038 | -1.1039 | 0.2698 | table |

Table 17: Model Fit Statistics

| | R.squared | Adjusted.R.squared | Residual.Std..Error | F.statistic | df1..num. | df2..den. |
|---|---|---|---|---|---|---|
| value | 0.8614 | 0.8609 | 0.3663 | 1547.287 | 8 | 1991 |

In a model where color and table variables are added, we see a Residual standard error of 0.3663, Multiple R-squared of 0.8614, and Adjusted R-squared of 0.8609, indicating a less fit than our `model2_log` model. So we will exclude these two variables.

One of the most interesting aspects of this part was seeing how strong the linear relationship was between carat and price. The simple linear regression model already produced an $R^2$ value above 0.85, suggesting that carat alone explains a substantial portion of the variation in price, which aligns with general life experience as traditionally considered "larger" diamond (diamond with a greater carat value) tend to cost more.

However, when checking the assumptions of linear regression, we encountered two violations, specifically the normality of residuals and equal variance. We tend applied log function to both the response and independent variable, which made the log model satisfying all three model assumptions and thus greatly improve the situation. It was also informative to see how much the model improved after adding more predictors. Both the full model and a subset model that included color and table led to higher $R^2$ and lower residual standard errors.

## Part - 3: Finding Optimal Model

We will be using stepwise AIC technique to find the optimal model for the diamond dataset.

```
# We previously defined a log full model
# Here, we define a full model without transformation
library(MASS)
full_model <- lm(price ~ ., data = d_rs)

aic_step <- stepAIC(full_model)
```

```
## Start:  AIC=27853.48
## price ~ X + carat + cut + color + clarity + depth + table + x +
##     y + z
##
##           Df  Sum of Sq        RSS   AIC
## - z        1     527217 2180640123 27852
## - y        1    1432996 2181545902 27853
## <none>                  2180112906 27854
## - X        1    5685167 2185798073 27857
## - depth    1    7396293 2187509199 27858
## - table    1   13242087 2193354993 27864
## - x        1   13944886 2194057792 27864
## - cut      4   34394215 2214507121 27877
## - color    6  567782811 2747895717 28304
## - clarity  7 1251204425 3431317331 28747
## - carat    1 2028629993 4208742899 29167
##
## Step:  AIC=27851.96
## price ~ X + carat + cut + color + clarity + depth + table + x +
##     y
```

```
##
##           Df  Sum of Sq        RSS   AIC
## - y        1    1746408 2182386531 27852
## <none>                  2180640123 27852
## - X        1    5779747 2186419870 27855
## - depth    1    7970108 2188610231 27857
## - table    1   13307704 2193947827 27862
## - x        1   13454784 2194094907 27862
## - cut      4   34171886 2214812009 27875
## - color    6  567907892 2748548015 28303
## - clarity  7 1252774921 3433415044 28746
## - carat    1 2029786291 4210426414 29166
##
## Step:  AIC=27851.56
## price ~ X + carat + cut + color + clarity + depth + table + x
##
##           Df  Sum of Sq        RSS   AIC
## <none>                  2182386531 27852
## - X        1    5459978 2187846508 27855
## - depth    1    8769420 2191155951 27858
## - table    1   13705675 2196092206 27862
## - cut      4   35486187 2217872717 27876
## - x        1  116677884 2299064415 27954
## - color    6  568697711 2751084242 28303
## - clarity  7 1280662618 3463049148 28761
## - carat    1 2048314367 4230700898 29174
```

From the AIC method, the last AIC attempt has the lowest AIC value. Thus, we obtain the following model:

```r
model_aic <- lm(formula = price ~ carat + cut + color + clarity + depth + table + x, data = d_rs)
model_aic_summary <- summary(model_aic)

coefs <- as.data.frame(model_aic_summary$coefficients)
colnames(coefs)[4] <- "Pr"
coefs$Term <- rownames(coefs)
rownames(coefs) <- NULL

rsq <- model_aic_summary$r.squared
adj_rsq <- model_aic_summary$adj.r.squared
sigma <- model_aic_summary$sigma
f_stat <- model_aic_summary$fstatistic[1]
df1 <- model_aic_summary$fstatistic[2]
df2 <- model_aic_summary$fstatistic[3]

overview <- data.frame(
  `R-squared` = rsq,
  `Adjusted R-squared` = adj_rsq,
  `Residual Std. Error` = sigma,
  `F-statistic` = f_stat,
  `df1 (num)` = df1,
  `df2 (den)` = df2
)

kable(coefs, caption = "Model Coefficients", digits = 4, escape = FALSE)
```

Table 18: Model Coefficients

| Estimate | Std. Error | t value | Pr | Term |
|---|---|---|---|---|
| 4376.4506 | 1988.8675 | 2.2005 | 0.0279 | (Intercept) |
| 11643.1650 | 267.2212 | 43.5713 | 0.0000 | carat |
| 638.2205 | 156.6729 | 4.0736 | 0.0000 | cutGood |
| 875.9172 | 158.8226 | 5.5151 | 0.0000 | cutIdeal |
| 773.9145 | 152.9770 | 5.0590 | 0.0000 | cutPremium |
| 756.6881 | 152.8899 | 4.9492 | 0.0000 | cutVery Good |
| -105.0285 | 87.7310 | -1.1972 | 0.2314 | colorE |
| -265.6332 | 87.1770 | -3.0471 | 0.0023 | colorF |
| -426.1352 | 88.0134 | -4.8417 | 0.0000 | colorG |
| -801.6799 | 92.5641 | -8.6608 | 0.0000 | colorH |
| -1326.6023 | 103.8046 | -12.7798 | 0.0000 | colorI |
| -2218.0851 | 123.7166 | -17.9288 | 0.0000 | colorJ |
| 5105.2052 | 231.2348 | 22.0780 | 0.0000 | clarityIF |
| 3553.0971 | 193.5204 | 18.3603 | 0.0000 | claritySI1 |
| 2659.4901 | 194.3729 | 13.6824 | 0.0000 | claritySI2 |
| 4377.4677 | 198.4628 | 22.0569 | 0.0000 | clarityVS1 |
| 4236.0681 | 194.1722 | 21.8160 | 0.0000 | clarityVS2 |
| 4776.5944 | 213.8139 | 22.3400 | 0.0000 | clarityVVS1 |
| 4854.1707 | 203.7147 | 23.8283 | 0.0000 | clarityVVS2 |
| -61.9410 | 20.3291 | -3.0469 | 0.0023 | depth |
| -51.9829 | 14.7524 | -3.5237 | 0.0004 | table |
| -1239.7275 | 113.3691 | -10.9353 | 0.0000 | x |

```
kable(overview, caption = "Model Fit Statistics", digits = 4, escape = FALSE)
```

Table 19: Model Fit Statistics

| | R.squared | Adjusted.R.squared | Residual.Std..Error | F.statistic | df1..num. | df2..den. |
|---|---|---|---|---|---|---|
| value | 0.9248 | 0.924 | 1051.708 | 1158.045 | 21 | 1978 |

Again, for our transformed linear model in part 2, `model2_log`, the Residual standard error is 0.2638, Multiple R-squared is 0.9279, and Adjusted R-squared is 0.9279.

In our model obtained by AIC, the Residual standard error is 1052, Multiple R-squared is 0.9248, and Adjusted R-squared is 0.924. This is worse than `model2_log`, but slightly better than model2, which is `model2_log` before applying transformation. This may indicate that we also need to take transformation on `model_aic` due to violation of linear model assumption(s).

We now start detecting multicollinearity among the variables using the variance inflation factor (VIF).

```
library("car")
kable(vif(model_aic))
```

| | GVIF | Df | GVIF^(1/(2*Df)) |
|---|---|---|---|
| carat | 27.333336 | 1 | 5.228129 |
| cut | 2.210572 | 4 | 1.104239 |
| color | 1.261950 | 6 | 1.019577 |

| | GVIF | Df | GVIF^(1/(2*Df)) |
|---|---|---|---|
| clarity | 1.436353 | 7 | 1.026202 |
| depth | 1.585815 | 1 | 1.259291 |
| table | 1.946715 | 1 | 1.395247 |
| x | 27.393533 | 1 | 5.233883 |

In the output of `vif` function, x have a great vif values, so we remove them as it may indicate potential multicollinearity.

```
model_aic_vif <- lm(formula = price ~ carat + clarity + color + cut + table + depth, data = d_rs)
model_aic_vif_summary <- summary(model_aic_vif)

coefs <- as.data.frame(model_aic_vif_summary$coefficients)
colnames(coefs)[4] <- "Pr"
coefs$Term <- rownames(coefs)
rownames(coefs) <- NULL

rsq <- model_aic_vif_summary$r.squared
adj_rsq <- model_aic_vif_summary$adj.r.squared
sigma <- model_aic_vif_summary$sigma
f_stat <- model_aic_vif_summary$fstatistic[1]
df1 <- model_aic_vif_summary$fstatistic[2]
df2 <- model_aic_vif_summary$fstatistic[3]

overview <- data.frame(
  `R-squared` = rsq,
  `Adjusted R-squared` = adj_rsq,
  `Residual Std. Error` = sigma,
  `F-statistic` = f_stat,
  `df1 (num)` = df1,
  `df2 (den)` = df2
)

kable(coefs, caption = "Model Coefficients", digits = 4, escape = FALSE)
```

Table 21: Model Coefficients

| Estimate | Std. Error | t value | Pr | Term |
|---|---|---|---|---|
| -4099.7528 | 1885.6823 | -2.1741 | 0.0298 | (Intercept) |
| 8793.6619 | 60.9621 | 144.2480 | 0.0000 | carat |
| 5220.1759 | 237.8156 | 21.9505 | 0.0000 | clarityIF |
| 3502.5758 | 199.1771 | 17.5852 | 0.0000 | claritySI1 |
| 2621.2337 | 200.0792 | 13.1010 | 0.0000 | claritySI2 |
| 4383.7913 | 204.3213 | 21.4554 | 0.0000 | clarityVS1 |
| 4218.2471 | 199.8979 | 21.1020 | 0.0000 | clarityVS2 |
| 4920.3491 | 219.7101 | 22.3947 | 0.0000 | clarityVVS1 |
| 4899.1276 | 209.6865 | 23.3641 | 0.0000 | clarityVVS2 |
| -103.9324 | 90.3211 | -1.1507 | 0.2500 | colorE |
| -324.1858 | 89.5813 | -3.6189 | 0.0003 | colorF |
| -446.6223 | 90.5913 | -4.9301 | 0.0000 | colorG |
| -818.6039 | 95.2837 | -8.5912 | 0.0000 | colorH |

| Estimate | Std. Error | t value | Pr | Term |
|---|---|---|---|---|
| -1306.4314 | 106.8524 | -12.2265 | 0.0000 | colorI |
| -2128.7935 | 127.0915 | -16.7501 | 0.0000 | colorJ |
| 707.2088 | 161.1677 | 4.3880 | 0.0000 | cutGood |
| 916.6939 | 163.4666 | 5.6078 | 0.0000 | cutIdeal |
| 834.4008 | 157.3905 | 5.3015 | 0.0000 | cutPremium |
| 785.8183 | 157.3800 | 4.9931 | 0.0000 | cutVery Good |
| -53.2266 | 15.1875 | -3.5046 | 0.0005 | table |
| -2.5613 | 20.1689 | -0.1270 | 0.8990 | depth |

```
kable(overview, caption = "Model Fit Statistics", digits = 4, escape = FALSE)
```

Table 22: Model Fit Statistics

| | R.squared | Adjusted.R.squared | Residual.Std..Error | F.statistic | df1..num. | df2..den. |
|---|---|---|---|---|---|---|
| value | 0.9202 | 0.9194 | 1082.759 | 1141.565 | 20 | 1979 |

```
# To make sure that we did things right, we also run vif function on the new model
kable(vif(model_aic_vif))
```

| | GVIF | Df | GVIF^(1/(2*Df)) |
|---|---|---|---|
| carat | 1.342140 | 1 | 1.158508 |
| clarity | 1.384108 | 7 | 1.023490 |
| color | 1.240952 | 6 | 1.018153 |
| cut | 2.203054 | 4 | 1.103769 |
| table | 1.946599 | 1 | 1.395206 |
| depth | 1.472671 | 1 | 1.213537 |

The vif values for the independent variables of our new model all turns out to be less than 5, therefore model `model_aic_vif` should satisfy our expectations.

We also create a confidence interval and a prediction interval for model `model_aic_vif`.

```
# We define one combination of predictors, or else the model run every combination
# For simplicity, we use the first diamond data as our baseline
new_data <- data.frame(
  carat = 0.23,
  clarity = "SI2",
  color = "E",
  cut = "Ideal",
  table = 55.0,
  depth = 61.5
)

# Confidence interval
predict(model_aic_vif, newdata = new_data, interval = "confidence", level = 0.95)
```

```
##         fit       lwr     upr
## 1 -1728.197 -1916.493 -1539.9
```

```r
# Prediction interval
predict(model_aic_vif, newdata = new_data, interval = "prediction", level = 0.95)
```

```
##         fit       lwr      upr
## 1 -1728.197 -3859.996 403.6028
```

Using the final linear model, `model_aic_vif`, we predicted the diamond price for a diamond with the following characteristics: carat = 0.23, clarity = "SI2", color = "E", cut = "Ideal", table = 55.0, and depth = 61.5. The 95% confidence interval for the mean predicted price is [-1916.493, -1539.9] (all values here are rounded so that they all have two digit after the decimal). This means that we are 95% confident that the average price for all diamonds with these exact characteristics falls within this range.

However, the negative lower bound is not realistic in context as price cannot be negative. This indicate that the model is poor in estimating a certain range of data, meaning that a transformation of random variables might be necessary.

The 95% prediction interval for the price of a single future diamond with the same characteristics is [-3859.996, 403.6028]. Again, the negative lower bound is not realistic in context (price cannot be negative), and it further suggests that the model has substantial variability and possibly room for improvement, such as making transformation.

# Discussion

This project examined the factors influencing diamond prices through a structured analysis of a sample of 2000 diamonds randomly selected from the dataset using `sample` function. We began with an study of the data's structure and distributions. Continuous variables such as carat, price, and the physical dimensions (x, y, z) displayed skewed patterns, which aligns with the general expectation that rare and larger diamond tend to cost more. Categorical variables like cut, color, and clarity were unevenly distributed, with Ideal cut and F color being most common in the sample.

Initial correlation analysis revealed several strong linear relationships. Carat showed an extremely high correlation with both price and the x, y, z dimensions, while the size dimensions were also strongly correlated with each other (greater than 0.98), indicating multicollinearity. Depth and table, by contrast, had weak associations with price. A multiple linear regression model incorporating all predictors produced a high adjusted $R^2$ of around 0.9241, suggesting that the overall model was strong. However, signs of multicollinearity and some insignificant predictors still exists and therefore potential room of improvement still exist.

The next step of our study involved fitting a simple linear regression model with carat as the only predictor of price. This model alone achieved a strong adjusted $R^2$ of 0.8504. Despite this, residual plots showed violations of key regression assumptions, specifically non-normal residuals and non-constant variance. A log transformation of both price and carat was applied to address these issues. The transformed model performed better: it achieved a higher adjusted $R^2$ of 0.9279 and showed improved residual behavior with reduced standard error, confirming the benefits of transformation. Adding other predictors such as color and table to the transformed model made the model less fit, so we will exclude these variables.

We then conducted a model selection using stepwise AIC. The resulting model included carat, clarity, color, x, cut, table, and depth, and retained a strong adjusted $R^2$. However, variance inflation factor (VIF) analysis showed that there is potential multicollinearity between carat and x. To resolve this, both variables were removed to produce a final model that retained clarity, color, cut, table, and depth.

Using this final model, a prediction was made for a diamond with characteristics including carat = 0.23, clarity = "SI2", color = "E", cut = "Ideal", table = 55.0, and depth = 61.5. The 95% confidence interval for the mean predicted price was [-1916.493, -1539.9], while the prediction interval for a single future diamond was [-3859.996, 403.6028]. The negative price confidence interval indicate that we might need to

conduct transformation to improve the model's estimation ability. The wide prediction interval reflected high variability in individual prices and suggested that even with quality indicators included, diamond pricing remains complex and influenced by additional factors.