Linear Regression Analysis on Diamond Prices

Alexis Alfaro Naranjo

2025-05-25

Introduction

Diamonds have long been in demand in the luxury market and are a strong symbol of one's status. Diamond prices are driven by quantifiable physical attributes such as cut, clarity, and carat weight, therefore enabling the construction of predictive pricing models utilizing multiple linear regression. Provided the high and persistent demand for diamonds, understanding how each attribute contributes to price provides invaluable insight and allows for better decision-making in the jewelry market. By statistically modeling the relationship between diamond attributes and price, this project aims to identify the most influential predictors and improve pricing transparency for consumers and sellers.

Specifically, this project analyzes a dataset that contains 2,000 randomly sampled diamonds. The primary goal of this analysis is to model the relationships between different diamonds with different attributes and their market prices.

Method

I began this project by importing a CSV file that contains a random sample of 2,000 observations of diamond sales. I then proceed to generate a reproducible random sample by setting a fixed random seed in order to ensure consistency across all analysis. The dataset contains both continuous and categorical predictors. The continuous variables include carat, depth, table, price, and physical dimensions (x, y, z). Whereas categorical variables include cut, color, and clarity. These values are explored through utilizing summary statistics and visualizations.

Utilizing this sampled dataset, histograms were constructed for all continuous variables in order to assess their distributions and therefore identify potential skewness, outliers, and deviations from normality.

Results

Part - 1: Data Description and Descriptive Statistics

We import the data from the diamonds dataset:

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

Then selecting a random sample:

```r
set.seed(5252025)

d_rs <- d_data[sample(nrow(d_data), 2000),]
```

```r
# Summary and Structure of the Dataset
kable(summary(d_rs), caption = "Summary Statistics for Diamond Dataset")
```

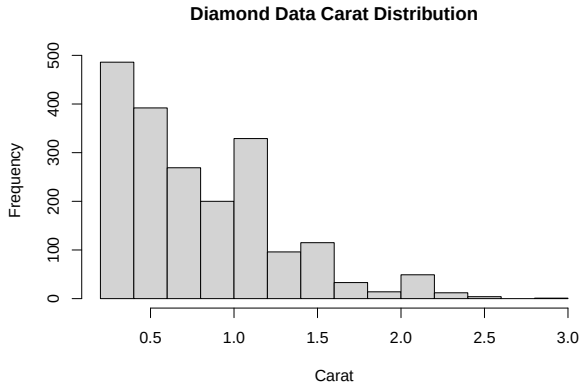
Table 1: Summary Statistics for Diamond Dataset

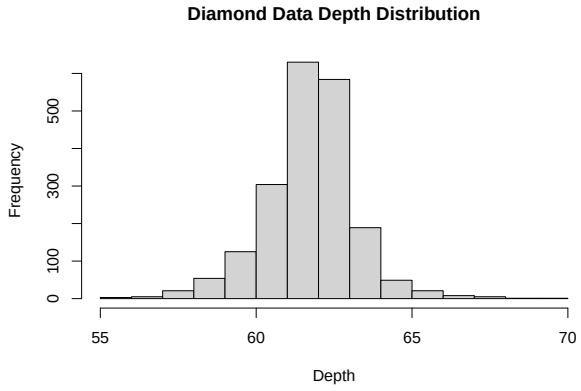
X	carat	cut	color	clarity	depth	table	price	x	y	z
Min. : 97	Min. :0.2100	Length:2000	Length:2000	Length:2000	Min. :55.20	Min. :52.00	Min. : 368	Min. :3.850	Min. :3.840	Min. :0.000
1st Qu.:13584	1st Qu.:0.4100	Class :character	Class :character	Class :character	1st Qu.:61.00	1st Qu.:56.00	1st Qu.: 1015	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 a series of histograms for these random variables.

```r
# 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")
```



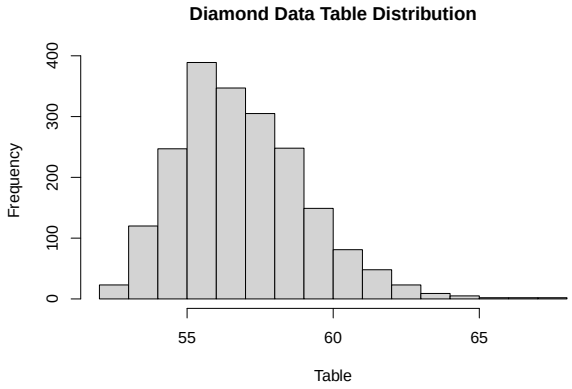


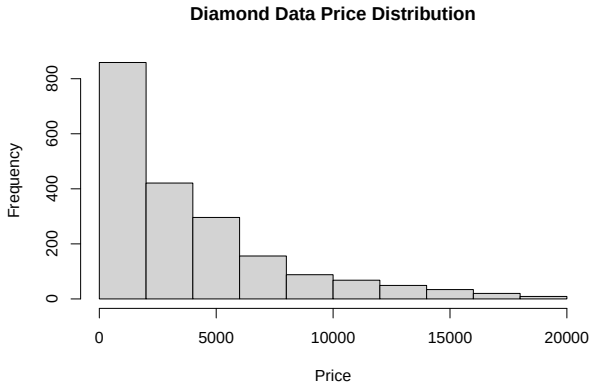
The Carat distribution appears to be not normal, with a heavy skew to the right. Most of the data lies around < 1.25 carat range, with a small amount of the other falls above 1.25 carat, therefore constitutes the right skewness.

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")
```



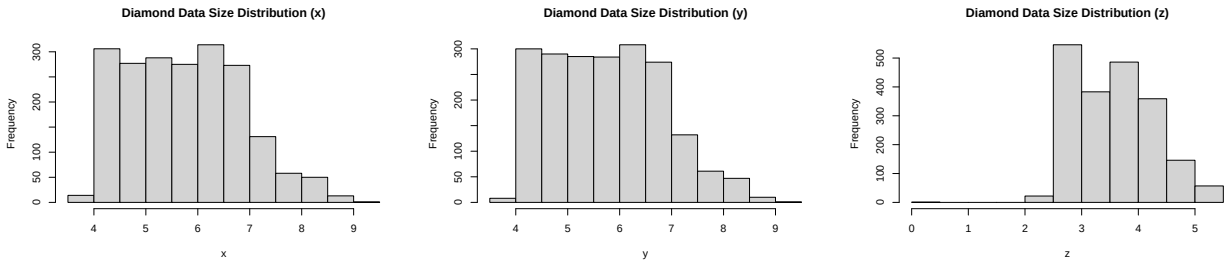


The table distribution appears to be generally normal with a slight right skewness, centering around the value of 56. The price distribution, however, appears to be not normal, with a extremely heavy skew to the right.

```
# Histogram for Table and Price
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")
```



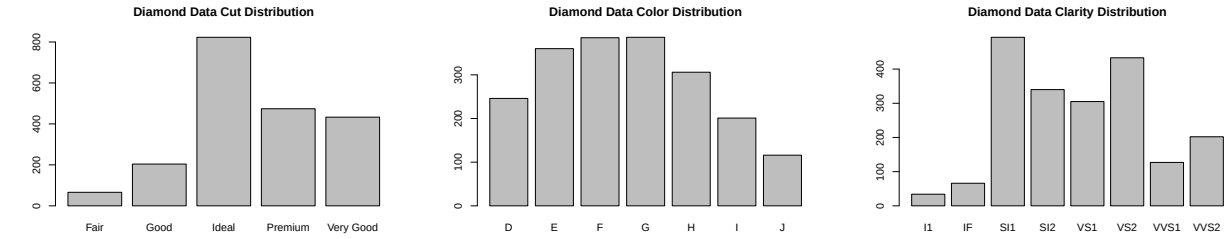
The x, y, and z variables' distributions are approximately normal, but not perfect. Specifically, both x and y distributions skews to the right, while z has a distributions skews to the left.

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

```
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 appears to be approximately normal with a slight skewness 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 appears to be approximately normal as well, but with a slight skewness to the right. In context, we may interpret as the most popular color found in the data set is "F".

Lastly, the clarity distribution appears to be normal also, but visually the normality is not as strong as the previous two variables as it has a bimodel shape and a slight skewness to the left.

```
d_rs_num <- d_rs[, -c(3, 4, 5)]

cor_mat <- cor(d_rs_num)
cor_df <- round(as.data.frame(cor_mat), 3)

cor_df2 <- cbind(Variable = rownames(cor_df), cor_df)

kable(cor_df2, caption = "Correlation Matrix of Numeric Diamond Variables")
```

Table 2: Correlation Matrix of Numeric Diamond Variables

	Variable	X	carat	depth	table	price	x	y	z
X	X	1.000	-0.403	-0.059	-0.097	-0.333	-0.429	-0.431	-0.428
carat	carat	-0.403	1.000	0.034	0.212	0.922	0.979	0.978	0.969
depth	depth	-0.059	0.034	1.000	-0.308	0.007	-0.028	-0.030	0.098
table	table	-0.097	0.212	-0.308	1.000	0.135	0.232	0.225	0.183
price	price	-0.333	0.922	0.007	0.135	1.000	0.888	0.889	0.879
x	x	-0.429	0.979	-0.028	0.232	0.888	1.000	0.999	0.982
y	y	-0.431	0.978	-0.030	0.225	0.889	0.999	1.000	0.982
z	z	-0.428	0.969	0.098	0.183	0.879	0.982	0.982	1.000

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.

```
#Running multiple linear regression
model1 <- lm(price~ ., data = d_rs)

model1_tidy <- tidy(model1)
kable(model1_tidy, digits = 3,
      caption = "Multiple Linear Regression Coefficients (Full Model)")
```

Table 3: Multiple Linear Regression Coefficients (Full Model)

term	estimate	std.error	statistic	p.value
(Intercept)	4173.931	2197.824	1.899	0.058
X	0.004	0.002	2.269	0.023
carat	11548.771	269.395	42.869	0.000
cutGood	622.816	159.623	3.902	0.000
cutIdeal	858.932	160.751	5.343	0.000
cutPremium	787.323	153.115	5.142	0.000
cutVery Good	729.346	157.637	4.627	0.000
colorE	-107.829	87.654	-1.230	0.219
colorF	-270.504	87.145	-3.104	0.002
colorG	-430.949	88.011	-4.897	0.000
colorH	-802.194	92.472	-8.675	0.000
colorI	-1338.042	103.800	-12.891	0.000
colorJ	-2227.149	123.681	-18.007	0.000
clarityIF	5049.144	233.098	21.661	0.000
claritySI1	3508.801	194.983	17.995	0.000
claritySI2	2626.083	195.463	13.435	0.000
clarityVS1	4333.297	200.101	21.655	0.000
clarityVS2	4187.127	195.581	21.409	0.000
clarityVVS1	4736.713	214.948	22.037	0.000

term	estimate	std.error	statistic	p.value
clarityVVS2	4795.982	205.610	23.326	0.000
depth	-65.191	25.185	-2.589	0.010
table	-51.087	14.750	-3.464	0.001
x	-1884.809	530.293	-3.554	0.000
y	602.918	529.165	1.139	0.255
z	175.519	253.972	0.691	0.490

```
#Observe the summary statistics
kable(glance(model1), digits = 3,
      caption = "Model Fit Statistics (Full Model)")
```

Table 4: Model Fit Statistics (Full Model)

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.925	0.924	1050.645	1015.633	0	24	-16739.62	33531.23	33676.86	2180112906	1975	2000

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

```
#Begin with one predictor and one response variable
#Conduct a simple linear regression
model2 <- lm(price ~ carat, data = d_rs)
kable(tidy(model2), digits = 3,
      caption = "Simple Linear Regression: Price vs Carat")
```

Table 5: Simple Linear Regression: Price vs Carat

term	estimate	std.error	statistic	p.value
(Intercept)	-2245.921	66.502	-33.772	0
carat	7643.866	71.698	106.612	0

```
kable(glance(model2), digits = 3,
      caption = "Model Fit Statistics for Simple Linear Regression Model")
```

Table 6: Model Fit Statistics for Simple Linear Regression Model

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.85	0.85	1475.291	11366.11	0	1	-17430.1	34866.2	34883	4348616218	1998	2000

In this simple linear regression analysis, 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 .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 just ran the model in Q1. The interpretation of the summary statistics are as follow:

- Estimates: For each unit change in carat, we expect the response variable (which is price in this case) to increase by 7643.9.
- Residual standard error: A residual standard error of 1475 indicates that predicted diamond prices deviate from observed prices by approximately $1,475 on average.
- Multiple R-squared: In this model, 85.05% of the variance can be explained.
- Adjusted R-squared: After adjusting (eliminating) the explanatory power of the model in consideration of correlation among independent variable (multicollinearity), 85.04% of the variance can be explained by our model.

```
ci <- confint(model2, level = 0.95) # We use traditional significance level of 0.05

ci_df <- as.data.frame(ci)
colnames(ci_df) <- c("Lower 95%", "Upper 95%")

kable(ci_df, digits = 3,
      caption = "95% Confidence Intervals for Simple Linear Regression Coefficients")
```

Confidence Intervals of the model are:

Table 7: 95% Confidence Intervals for Simple Linear Regression Coefficients

	Lower 95%	Upper 95%
(Intercept)	-2376.341	-2115.500
carat	7503.256	7784.477

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.

```
# We specified carat to be 0.5, or else the `predict` function
# will return prediction interval for all carat values
ddf = data.frame(carat = 0.5)

pred <- predict(model2, ddf, interval = "prediction", level = 0.95)

pred_df <- as.data.frame(pred)

kable(pred_df, digits = 2,
      caption = "95% Prediction Interval for a 0.5-Carat Diamond")
```

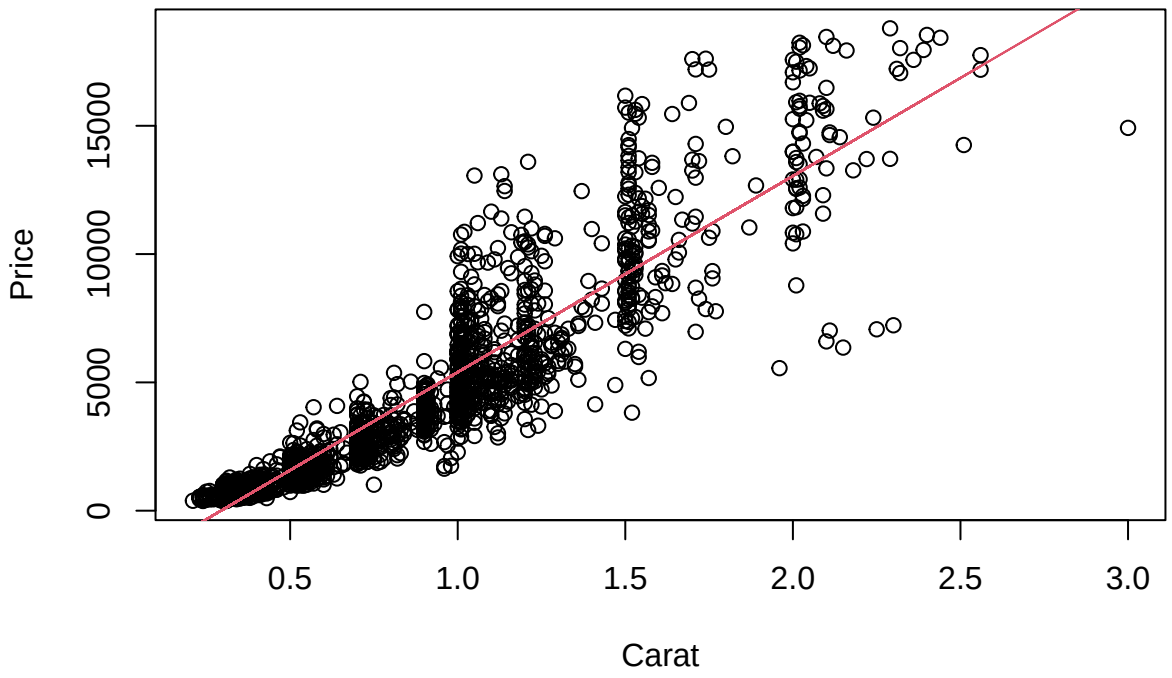
Prediction Interval for the model:

Table 8: 95% Prediction Interval for a 0.5-Carat Diamond

fit	lwr	upr
1576.01	-1318.3	4470.33

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.

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



Plot

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.

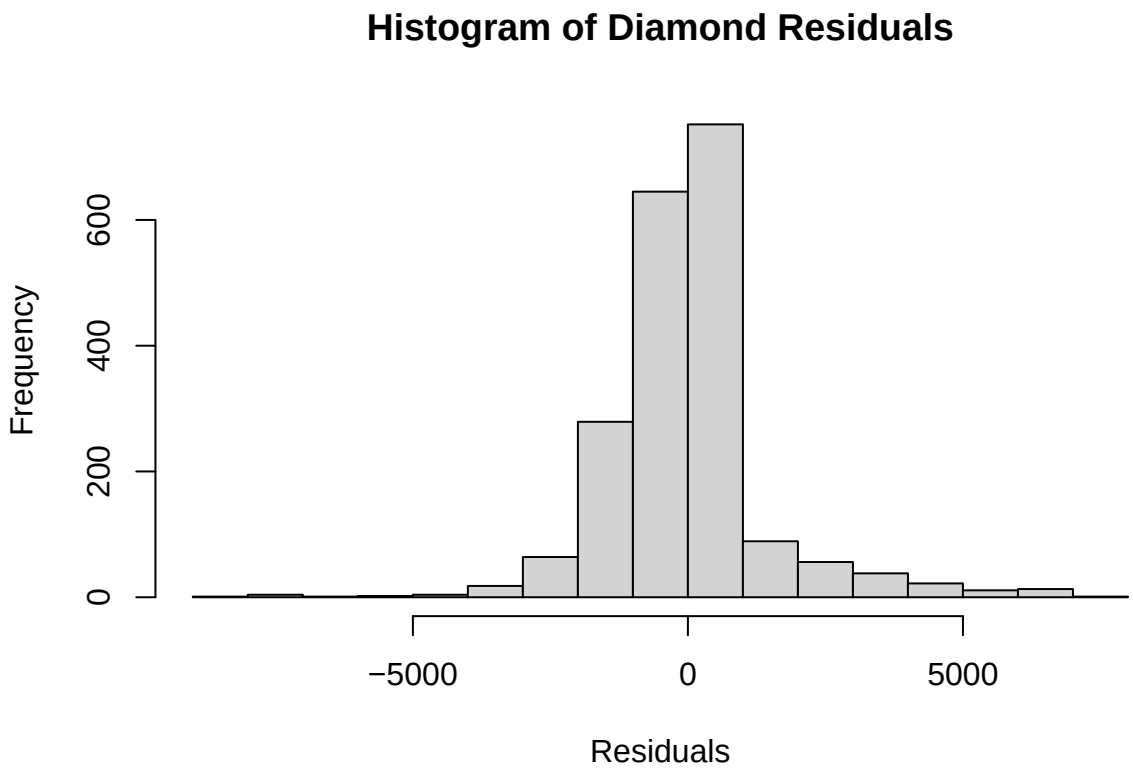
3. Test the assumptions and apply any necessary transformations to the response variable y or the predictor.

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**

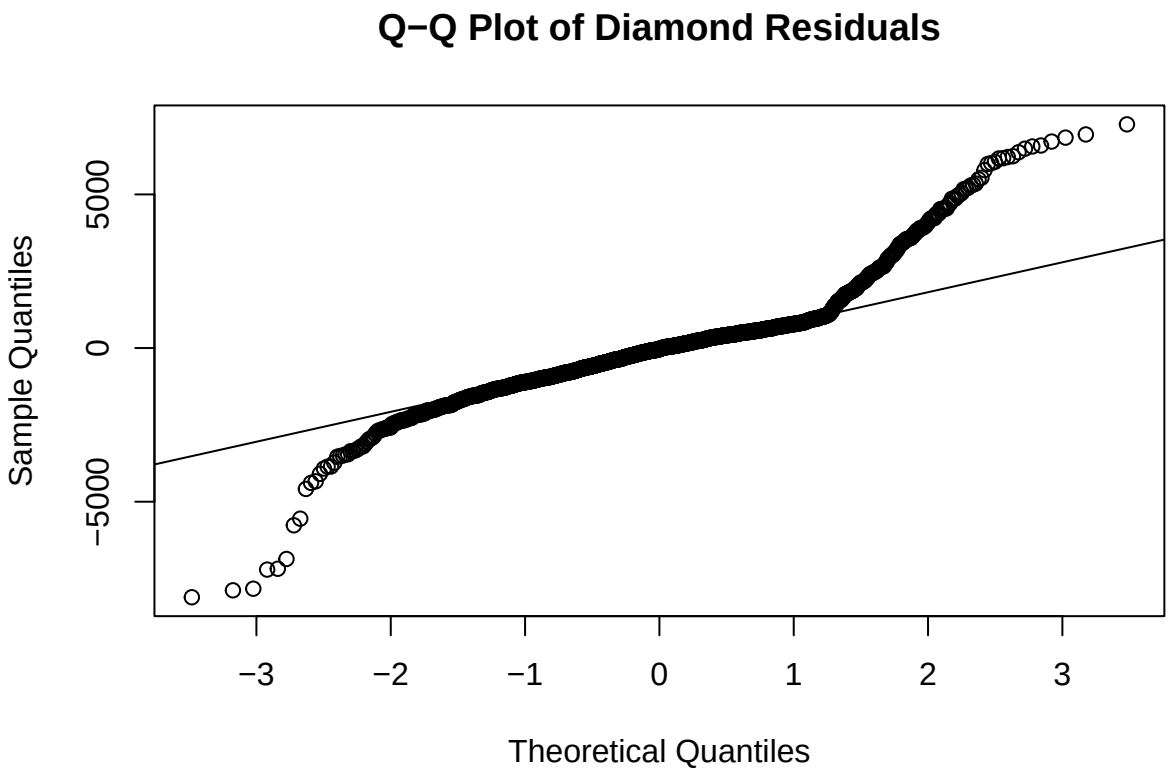
```r
# Histogram of the Residuals
hist(model2$residuals, main = "Histogram of Diamond Residuals", xlab = "Residuals")
```

Normality of the Data



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$statistic, 5),
  P_Value = signif(shapiro_result$p.value, 5)
)

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

Table 9: Shapiro-Wilk Normality Test for Diamond Model Residuals

	Statistic	P_Value
W	0.88859	0

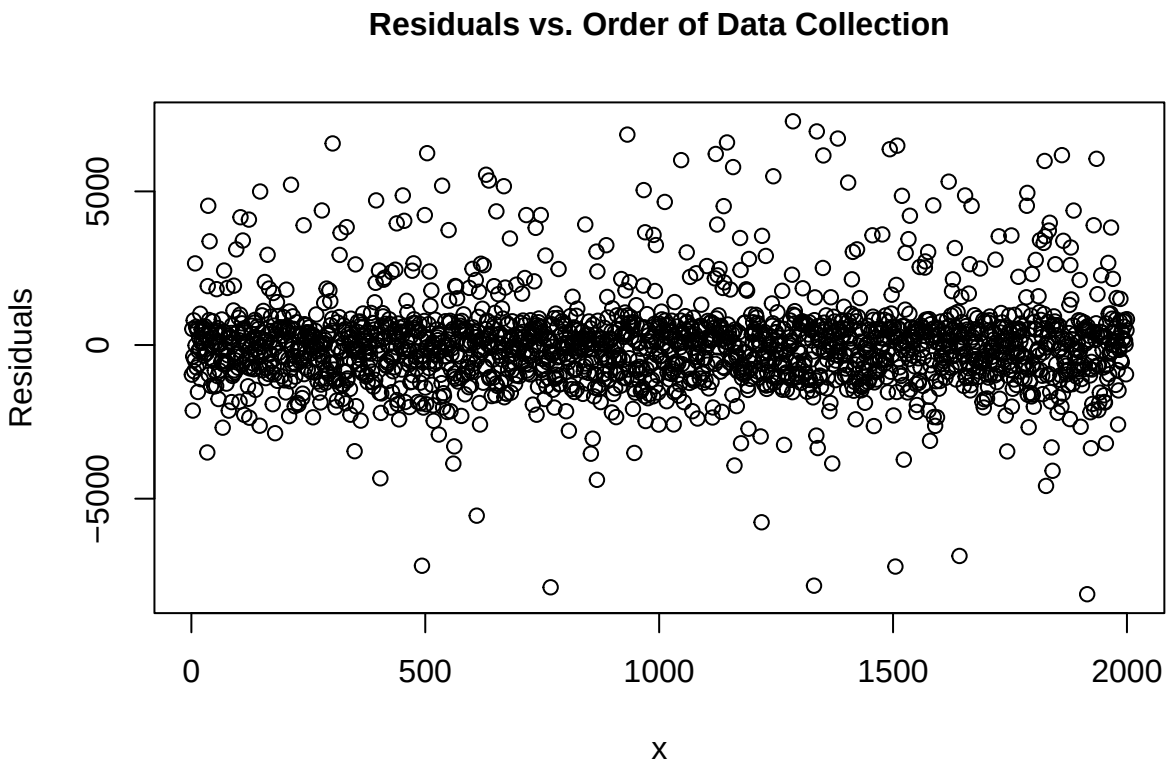
According to the Shapiro-Wilk Normality Test, we derive a p-value of 9.7705×10^{-36} at $\alpha = 0.05$. Therefore, we reject the null hypothesis of the test that our data follows a normal distribution and conclude that our data does not follows a normal distribution.

All three approaches we used to examine the normality of our data - histogram, Q-Q plot, and shapiro-wilk normality test - indicates contradicting conclusions. Thus, we conclude that the normality assumption of the test is violated and therefore **the normality assumption is not met**.

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

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

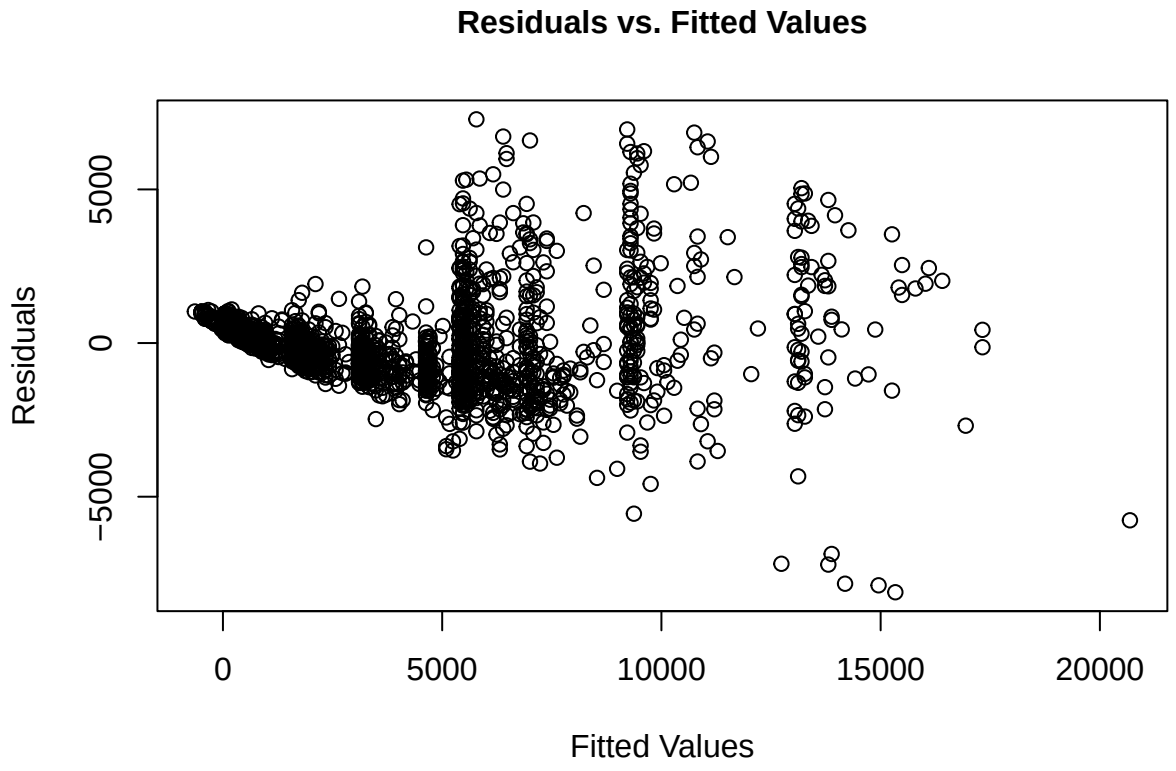
Structure of the Data



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**.

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

Equal Variances across Fitted Values

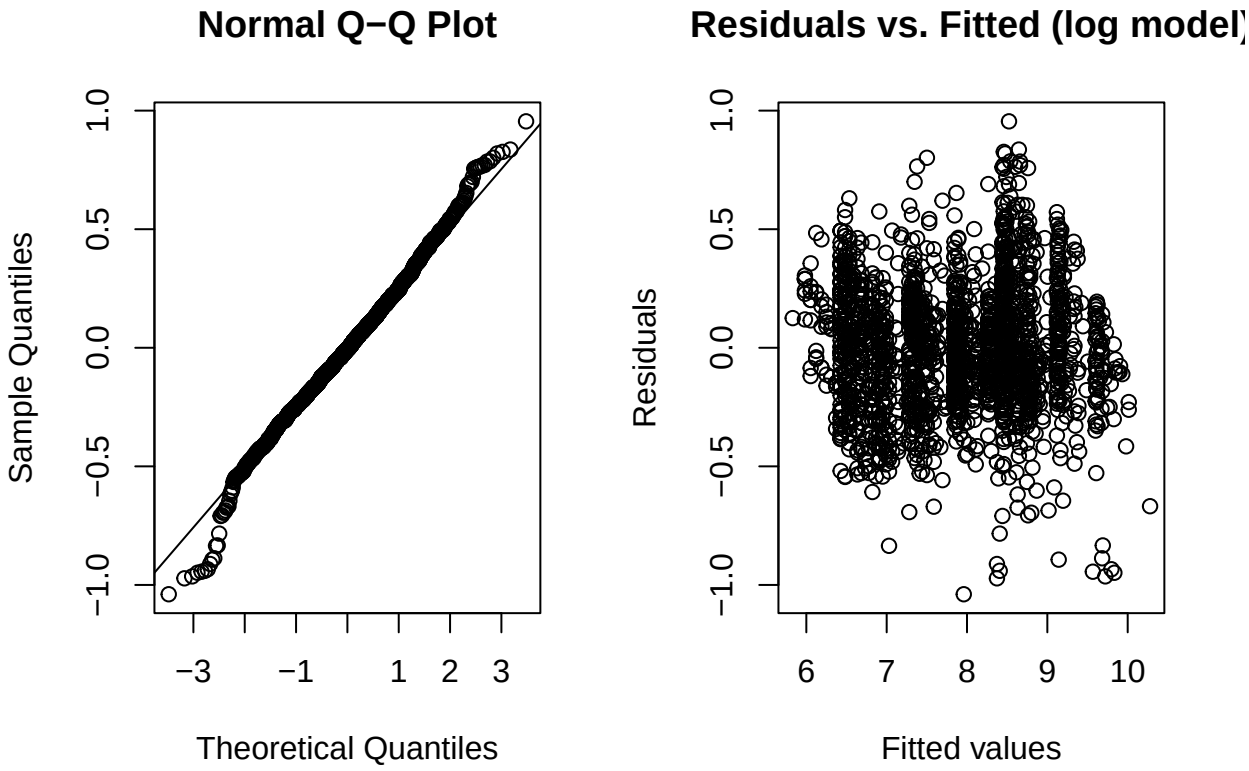


According to the plot, variances across different fitted values seems to be speading out as the fitted values increase. Therefore, the **equal variance across fitted value assumption is not met**.

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

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 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.

Selection of the Final Regression Model

```
kable(
  tidy(model2_log),
  digits = 4,
  caption = "Log-Log Regression Coefficients: log(Price) vs log(Carat)"
)
```

Table 10: Log–Log Regression Coefficients: log(Price) vs log(Carat)

term	estimate	std.error	statistic	p.value
(Intercept)	8.4410	0.0071	1193.7187	0
log(carat)	1.6725	0.0104	160.3844	0

```
kable(
  glance(model2_log),
```

```
  digits = 4,
  caption = "Model Fit Statistics for Log-Log Regression Model"
)
```

Table 11: Model Fit Statistics for Log–Log Regression Model

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.9279	0.9279	0.2638	25723.16	0	1	-171.4019	348.8038	365.6065	138.9935	1998	2000

After transforming the response variable using the natural logarithm, the summary 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 fron 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.

Model Refinement and Feature Evaluation

In order to improve predictive performance beyond the baseline carat only model, additional diamond attributes were evaluated as candidate predictors. Variables were added iteratively and compared using adjusted R^2 and residual standard error in order to determine whether each feature contributed meaningful explanatory power.

The transformed model substantially outperforms the untransformed simple regression, with improved explanatory power and reduced residual variability. This result motivated further exploration of whether additional predictors could meaningfully enhance performace.

In a full model where we assessed every combination of independent variables, 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.

Table 12: Log-Price Regression Coefficients (Carat, Color, Table)

term	estimate	std.error	statistic	p.value
(Intercept)	6.4974	0.2150	30.2206	0.0000
carat	2.0549	0.0192	107.2700	0.0000
colorE	-0.0227	0.0304	-0.7491	0.4539
colorF	0.0251	0.0300	0.8388	0.4017
colorG	-0.0514	0.0300	-1.7135	0.0868
colorH	-0.1936	0.0318	-6.0952	0.0000
colorI	-0.2884	0.0355	-8.1152	0.0000
colorJ	-0.5133	0.0425	-12.0736	0.0000
table	-0.0042	0.0038	-1.1039	0.2698

Table 13: Model Fit Statistics for Log-Price Regression Model

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.8614	0.8609	0.3663	1547.287	0	8	-824.9946	1669.989	1725.998	267.206	1991	2000

Similar pattern is also evident in adding only some other independent variables into the model. For instance, in a model where color and table variables are added, we see a Residual standard error of 0.2383, Multiple R-squared of 0.9414, and Adjusted R-squared of 0.9411, indicating a better fit than our `model2_log` model.

Model Insights and Observations

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

Model Selection and Evaluation

In order to identify an optimal predictive model, formal model selection techniques were applied to balance goodness of fit with model complexity. Stepwise selection based on the Akaike Information Criterion (AIC) was used to compare possible models and remove predictors that did not meaningfully improve performance.

```
# 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, trace = FALSE)


# Coefficient table of final AIC-selected model
kable(
  tidy(aic_step),
  digits = 3,
  caption = "Regression Coefficients for AIC-Selected Model"
)
```

Table 14: Regression Coefficients for AIC-Selected Model

term	estimate	std.error	statistic	p.value
(Intercept)	3762.422	2005.977	1.876	0.061
X	0.004	0.002	2.224	0.026
carat	11574.972	268.710	43.076	0.000
cutGood	656.870	156.741	4.191	0.000
cutIdeal	887.387	158.748	5.590	0.000
cutPremium	788.237	152.960	5.153	0.000
cutVery Good	774.440	152.946	5.063	0.000
colorE	-106.293	87.645	-1.213	0.225
colorF	-266.436	87.091	-3.059	0.002
colorG	-430.805	87.951	-4.898	0.000
colorH	-801.549	92.472	-8.668	0.000
colorI	-1335.775	103.783	-12.871	0.000
colorJ	-2228.247	123.678	-18.017	0.000
clarityIF	5084.583	231.191	21.993	0.000
claritySI1	3537.968	193.447	18.289	0.000
claritySI2	2649.405	194.232	13.640	0.000
clarityVS1	4365.455	198.339	22.010	0.000
clarityVS2	4214.344	194.225	21.698	0.000
clarityVVS1	4765.117	213.663	22.302	0.000
clarityVVS2	4828.204	203.846	23.685	0.000
depth	-57.515	20.406	-2.819	0.005
table	-51.930	14.738	-3.524	0.000
x	-1188.352	115.588	-10.281	0.000

```
# Model fit statistics
kable(
  glance(aic_step),
  digits = 3,
  caption = "Model Fit Statistics for AIC-Selected Model"
)
```

Table 15: Model Fit Statistics for AIC-Selected Model

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.925	0.924	1050.661	1107.836	0	22	-16740.66	33529.32	33663.74	2182386531	1977	2000

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

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

# Coefficient table
kable(
  tidy(model_aic),
  digits = 3,
  caption = "Regression Coefficients for AIC-Selected Model"
)
```

Table 16: Regression Coefficients for AIC-Selected Model

term	estimate	std.error	statistic	p.value
(Intercept)	4376.451	1988.868	2.200	0.028
carat	11643.165	267.221	43.571	0.000
cutGood	638.220	156.673	4.074	0.000
cutIdeal	875.917	158.823	5.515	0.000
cutPremium	773.914	152.977	5.059	0.000
cutVery Good	756.688	152.890	4.949	0.000
colorE	-105.028	87.731	-1.197	0.231
colorF	-265.633	87.177	-3.047	0.002
colorG	-426.135	88.013	-4.842	0.000
colorH	-801.680	92.564	-8.661	0.000
colorI	-1326.602	103.805	-12.780	0.000
colorJ	-2218.085	123.717	-17.929	0.000
clarityIF	5105.205	231.235	22.078	0.000
claritySI1	3553.097	193.520	18.360	0.000
claritySI2	2659.490	194.373	13.682	0.000
clarityVS1	4377.468	198.463	22.057	0.000
clarityVS2	4236.068	194.172	21.816	0.000
clarityVVS1	4776.594	213.814	22.340	0.000
clarityVVS2	4854.171	203.715	23.828	0.000
depth	-61.941	20.329	-3.047	0.002
table	-51.983	14.752	-3.524	0.000
x	-1239.728	113.369	-10.935	0.000

```
# Model fit statistics
kable(
  glance(model_aic),
  digits = 3,
  caption = "Model Fit Statistics for AIC-Selected Model"
)
```

Table 17: Model Fit Statistics for AIC-Selected Model

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.925	0.924	1051.708	1158.045	0	21	-16743.16	33532.32	33661.14	2187846508	1978	2000

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).

Multicollinearity Assessment

```r
library(car)
library(knitr)

vif_vals <- vif(model_aic)

vif_df <- as.data.frame(vif_vals)
colnames(vif_df) <- "VIF"

kable(vif_df, digits = 3,
      caption = "Variance Inflation Factors (VIF) for AIC-Selected Model")
```

Table 18: Variance Inflation Factors (VIF) for AIC-Selected Model

	VIF	NA	NA
carat	27.333	1	5.228
cut	2.211	4	1.104
color	1.262	6	1.020
clarity	1.436	7	1.026
depth	1.586	1	1.259
table	1.947	1	1.395
x	27.394	1	5.234

The variable x exhibits a high VIF, indicating substantial multicollinearity.

```r
# Final model after removing multicollinearity
model_aic_vif <- lm(
  price ~ carat + clarity + color + cut + table + depth,
  data = d_rs
)

# Coefficient table
kable(
  tidy(model_aic_vif),
  digits = 3,
  caption = "Regression Coefficients for Final Model After VIF Adjustment"
)
```

Table 19: Regression Coefficients for Final Model After VIF Adjustment

term	estimate	std.error	statistic	p.value
(Intercept)	-4099.753	1885.682	-2.174	0.030
carat	8793.662	60.962	144.248	0.000
clarityIF	5220.176	237.816	21.951	0.000
claritySI1	3502.576	199.177	17.585	0.000
claritySI2	2621.234	200.079	13.101	0.000
clarityVS1	4383.791	204.321	21.455	0.000
clarityVS2	4218.247	199.898	21.102	0.000
clarityVVS1	4920.349	219.710	22.395	0.000

term	estimate	std.error	statistic	p.value
clarityVVS2	4899.128	209.686	23.364	0.000
colorE	-103.932	90.321	-1.151	0.250
colorF	-324.186	89.581	-3.619	0.000
colorG	-446.622	90.591	-4.930	0.000
colorH	-818.604	95.284	-8.591	0.000
colorI	-1306.431	106.852	-12.227	0.000
colorJ	-2128.793	127.091	-16.750	0.000
cutGood	707.209	161.168	4.388	0.000
cutIdeal	916.694	163.467	5.608	0.000
cutPremium	834.401	157.390	5.301	0.000
cutVery Good	785.818	157.380	4.993	0.000
table	-53.227	15.188	-3.505	0.000
depth	-2.561	20.169	-0.127	0.899

```
# Model fit statistics
kable(
  glance(model_aic_vif),
  digits = 3,
  caption = "Model Fit Statistics for Final Model After VIF Adjustment"
)
```

Table 20: Model Fit Statistics for Final Model After VIF Adjustment

r.squared	adj.r.squared	sigma	statistic	p.value	df	logLik	AIC	BIC	deviance	df.residual	nobs
0.92	0.919	1082.759	1141.565	0	20	-16801.86	33647.71	33770.93	2320114073	1979	2000

```
# VIF table for final model
vif_vals <- vif(model_aic_vif)
vif_df <- as.data.frame(vif_vals)
colnames(vif_df) <- "VIF"

kable(
  vif_df,
  digits = 3,
  caption = "Variance Inflation Factors (VIF) for Final Model"
)
```

Table 21: Variance Inflation Factors (VIF) for Final Model

	VIF	NA	NA
carat	1.342	1	1.159
clarity	1.384	7	1.023
color	1.241	6	1.018
cut	2.203	4	1.104
table	1.947	1	1.395
depth	1.473	1	1.214

The vif values for the independent variables of our new model all turns out to be less than 5, therefore we should be good to go.

Prediction and Uncertainty Quantification

```
# 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 for mean price
ci_pred <- predict(
  model_aic_vif,
  newdata = new_data,
  interval = "confidence",
  level = 0.95
)

ci_df <- as.data.frame(ci_pred)

kable(
  ci_df,
  digits = 2,
  caption = "95% Confidence Interval for Mean Predicted Diamond Price"
)
```

Table 22: 95% Confidence Interval for Mean Predicted Diamond Price

fit	lwr	upr
-1728.2	-1916.49	-1539.9

```
# Prediction interval for a future diamond
pi_pred <- predict(
  model_aic_vif,
  newdata = new_data,
  interval = "prediction",
  level = 0.95
)

pi_df <- as.data.frame(pi_pred)

kable(
  pi_df,
```

```
  digits = 2,
  caption = "95% Prediction Interval for a Future Diamond Price"
)
```

Table 23: 95% Prediction Interval for a Future Diamond Price

fit	lwr	upr
-1728.2	-3860	403.6

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. The negative lower bounds highlight a limitation of model price on the original scale and further support the use of transformations.

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.

Conclusion and Key Findings

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 fit better.

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.