Shaping Value: A Regression Analysis of Diamond Characteristics and Pricing

Mettu Sathvika (811352339)

2024-12-06

Introduction:

The diamond data set provides detailed information about different attributes of diamonds,such as their carat, cut, color, clarity, and price.It is commonly used in data analysis to explore how various characteristics influence the pricing of diamonds, offering valuable insights into the factors that determine their market value.

Research Questions:

1. What are the key factors that influence diamond pricing?
2. How does the carat weight of a diamond affect its price?
3. To what extent do factors like cut and clarity contribute to the price of diamonds?
4. Is there a notable difference in price across different diamond colors and cuts?

Description of Variables Price:

Response Variable: The cost of the diamond, which is the variable we are trying to predict. **Regressor Variable:** Carat (Quantitative): The weight of the diamond, expressed in carats. Cut (Categorical): The quality of the diamond's cut, categorized into grades like Ideal, Premium,and Good. Color (Categorical): The color rating of the diamond, ranging from D(colorless) to Z. Clarity (Categorical): The clarity level of the diamond, ranging from Flawless to Included. Depth (Quantitative): The depth percentage of the diamond. Table (Quantitative): The table percentage, which refers to the size of the diamond's top surface relative to its overall size.

```
library(ggplot2)
library(car)
library(MASS)
diamond_data <- read.csv("C:\\Users\\sathv\\Downloads\\Diamondsdata.csv")
str(diamond_data)
```

```
## 'data.frame':    7999 obs. of  10 variables:
##  $ carat  : num  0.23 0.21 0.23 0.29 0.31 0.24 0.24 0.26 0.22 0.23 ...
##  $ cut    : chr  "Ideal" "Premium" "Good" "Premium" ...
##  $ color  : chr  "E" "E" "E" "I" ...
##  $ clarity: chr  "SI2" "SI1" "VS1" "VS2" ...
##  $ depth  : num  61.5 59.8 56.9 62.4 63.3 62.8 62.3 61.9 65.1 59.4 ...
##  $ table  : num  55 61 65 58 58 57 57 55 61 61 ...
##  $ price  : int  326 326 327 334 335 336 336 337 337 338 ...
##  $ x      : num  3.95 3.89 4.05 4.2 4.34 3.94 3.95 4.07 3.87 4 ...
##  $ y      : num  3.98 3.84 4.07 4.23 4.35 3.96 3.98 4.11 3.78 4.05 ...
##  $ z      : num  2.43 2.31 2.31 2.63 2.75 2.48 2.47 2.53 2.49 2.39 ...
```

The “summary(diamond_data)” function in R provides a concise statistical overview of each variable in the dataset. For quantitative variables, it returns key metrics like minimum, maximum, mean, median, and quartiles, helping to understand the data distribution. For categorical variables, it shows the frequency of each category. This function is crucial for exploratory data analysis, as it identifies central tendencies, variability, and potential outliers, while also helping to distinguish between numerical and categorical variables, making it a foundational step for data cleaning and model preparation.

```
summary(diamond_data)
```

```
##      carat            cut                color              clarity         
##  Min.   :0.2000   Length:7999        Length:7999        Length:7999       
##  1st Qu.:0.7200   Class :character   Class :character   Class :character  
##  Median :0.9000   Mode  :character   Mode  :character   Mode  :character  
##  Mean   :0.8278                                                           
##  3rd Qu.:1.0000                                                           
##  Max.   :1.5200                                                           
##      depth           table          price             x                y         
##  Min.   :43.00   Min.   :49.0   Min.   : 326   Min.   :3.790   Min.   :3.750  
##  1st Qu.:61.00   1st Qu.:56.0   1st Qu.:2990   1st Qu.:5.760   1st Qu.:5.770  
##  Median :61.90   Median :58.0   Median :3492   Median :6.080   Median :6.090  
##  Mean   :61.86   Mean   :57.8   Mean   :3290   Mean   :5.946   Mean   :5.947  
##  3rd Qu.:62.70   3rd Qu.:59.0   3rd Qu.:4014   3rd Qu.:6.350   3rd Qu.:6.340  
##  Max.   :71.80   Max.   :69.0   Max.   :4459   Max.   :7.530   Max.   :7.420  
##        z         
##  Min.   :0.000  
##  1st Qu.:3.550  
##  Median :3.770  
##  Mean   :3.677  
##  3rd Qu.:3.930  
##  Max.   :4.870
```

Interpretation : The summary statistics indicate that the dataset contains 7,999 observations. For quantitative variables, carat ranges from 0.2 to 1.52, with a mean of 0.83 and a median of 0.9, while depth ranges from 43% to 71.8% with a mean of 61.86%. The table variable spans 49 to 69, with a mean of 57.8. Price varies from $326 to $4,459, with a mean of $3,290 and a median of $3,492. Dimensions (‘x’, ‘y’, and ‘z’) show similar distributions, with minimum values starting around 3.75 and maximums reaching 7.53. The categorical variables, including cut, color, and clarity, are encoded as character data, indicating the dataset’s mix of numerical and categorical attributes, suitable for regression analysis.

Regression Analysis: For this dataset , regression analysis can be used to model the relationship between the price of a diamond (dependent variable) and its characteristics such as carat,cut,clarity,and depth (independent variables). The goal is to predict diamond prices based on these features.The data set supports multiple linear regression, where the relationship is assumed to be linear.Key assumptions include linearity, independence, and normally distributed residuals.The effectiveness of the model is evaluated using metrics like R-squared,p-values, and residual plots to check for model assumptions.

Model Fitting The following code creates a linear regression model (`model`) to predict the price of diamonds (price) using carat, cut, clarity, and depth as predictor variables from the diamond_data dataset. The lm() function fits a linear model by estimating coefficients for the predictors that minimize the sum of squared residuals. The summary(model) function then provides detailed information about the model, including the estimated coefficients, their standard errors, t-values, and p-values to assess statistical significance. Additionally, it displays model metrics such as the residual standard error, R-squared, adjusted R-squared, and the F-statistic, which help evaluate the model's overall fit and performance.

```
model <- lm(price ~ carat + cut + clarity + depth, data = diamond_data)
summary(model)
```

```
##
## Call:
## lm(formula = price ~ carat + cut + clarity + depth, data = diamond_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2303.45  -267.36    21.31   287.74  2523.68
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1763.225    205.442  -8.583  < 2e-16 ***
## carat         4960.368     25.456 194.863  < 2e-16 ***
## cutGood        319.692     25.092  12.741  < 2e-16 ***
## cutIdeal       497.649     24.046  20.695  < 2e-16 ***
## cutPremium     280.527     24.575  11.415  < 2e-16 ***
## cutVery Good   387.060     24.005  16.124  < 2e-16 ***
## clarityIF     2162.374     49.767  43.450  < 2e-16 ***
## claritySI1    1414.265     30.433  46.471  < 2e-16 ***
## claritySI2    1130.439     29.497  38.324  < 2e-16 ***
## clarityVS1    1700.796     32.576  52.209  < 2e-16 ***
## clarityVS2    1552.486     31.619  49.099  < 2e-16 ***
## clarityVVS1   2057.625     38.828  52.993  < 2e-16 ***
## clarityVVS2   1895.579     36.457  51.995  < 2e-16 ***
## depth          -13.473      3.122  -4.316 1.61e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 420.2 on 7985 degrees of freedom
## Multiple R-squared:  0.8383, Adjusted R-squared:  0.8381
## F-statistic:  3185 on 13 and 7985 DF,  p-value: < 2.2e-16
```

Interpretation : The regression model for predicting diamond prices shows that carat, cut, clarity, and depth significantly influence the price. The carat coefficient of 4960.37 suggests that each additional carat increases the price by approximately 4960.37, while cut and clarity also have substantial positive effects, with Ideal cut and Internally Flawless clarity having the highest impact. The depth coefficient is negative (-13.47), indicating a slight decrease in price with increasing depth, but it has a smaller effect compared to other variables.The model explains 83.83% of the variance in diamond prices (R-squared), with all predictors being highly significant(p-values < 0.001). The F-statistic confirms that the model as a whole is statistically significant. These results highlight the strong relationship between diamond characteristics and price, with carat being the most influential predictor.

Hypothesis Testing:

The 'anova(model)' function performs an analysis of variance (ANOVA) on the linear model created using the 'lm()' function. It partitions the total variation in the response variable (price) into components explained by each predictor (carat, cut,clarity, and depth) and the residuals (unexplained variation). The output includes the degrees of freedom , sum of squares, mean square, F-value, and p-value for each predictor. These metrics help determine the relative contribution of each predictor to the model and assess their statistical significance in explaining the variation in the response variable. Predictors with low p-values significantly impact the model.

```
anova(model)
```

```
## Analysis of Variance Table
##
## Response: price
##             Df     Sum Sq    Mean Sq   F value    Pr(>F)
## carat        1 6385804590 6385804590 36169.209 < 2.2e-16 ***
## cut          4  157396600   39349150   222.874 < 2.2e-16 ***
## clarity      7  763067647  109009664   617.431 < 2.2e-16 ***
## depth        1    3288570    3288570    18.627 1.609e-05 ***
## Residuals 7985 1409780616     176554
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

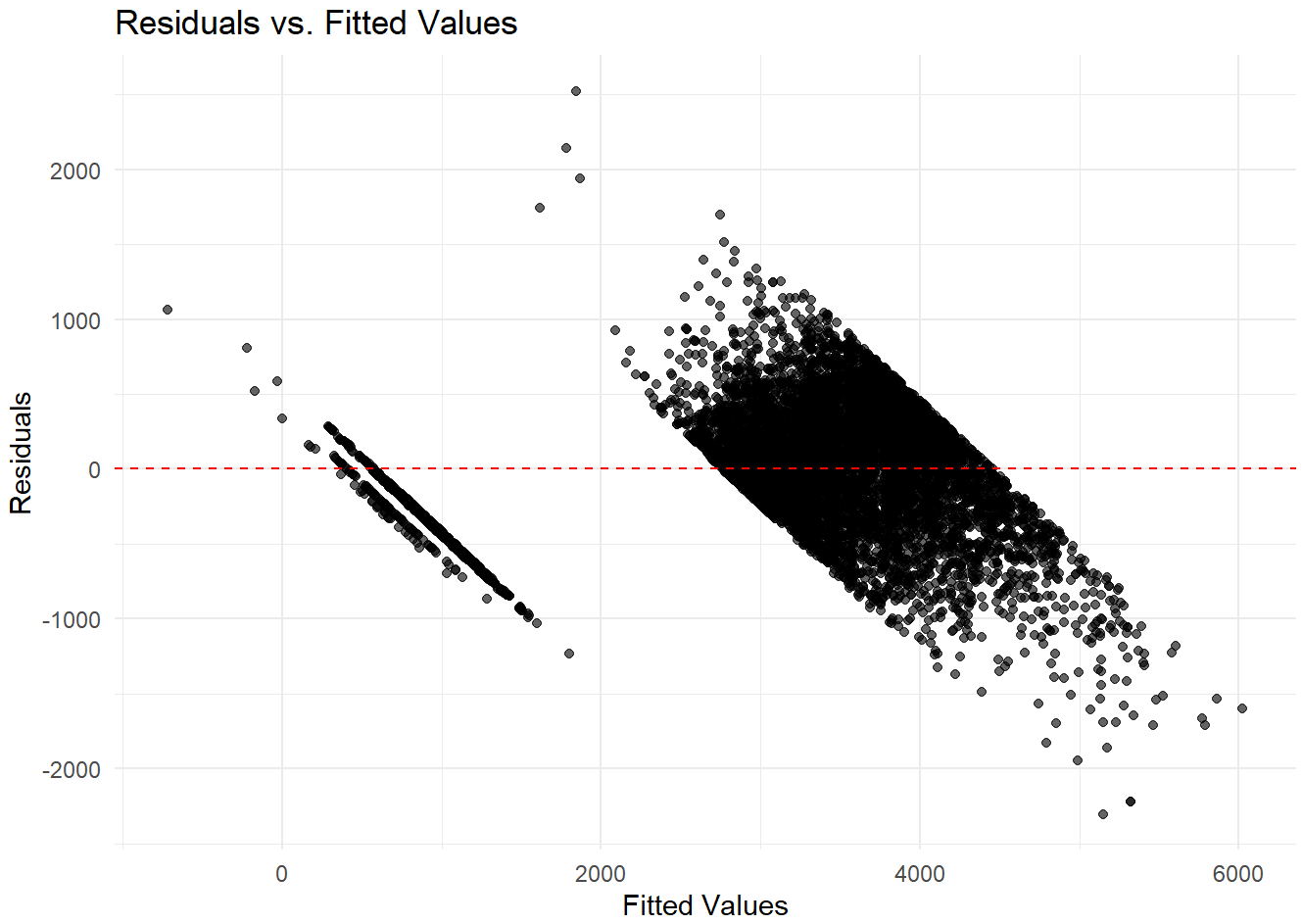
Interpretation: The ANOVA table for the regression model indicates that all predictors carat, cut, clarity, and depth are highly significant with very low p-values (< 0.001), meaning they all significantly contribute to explaining the variation in diamond prices. Carat has the largest impact, with an F-value of 36169.209, suggesting it is the most influential factor on price.Cut and clarity also have strong effects, with F-values of 222.874 and 617.431, respectively, indicating that different cut categories and clarity levels significantly affect price.Depth has a smaller but still significant effect with an F-value of 18.627. The residuals indicate some unexplained variation in the model,suggesting that other factors not included in the model may also influence diamond prices.

Model Diagnostics: In the diamonds data set, regression analysis models the relationship between the diamond price (dependent variable) and features like carat, cut, clarity, and depth (independent variables). Model diagnostics are essential to validate assumptions: linearity, normality of residuals, homoscedasticity, and the absence of influential points. Residual plots check for linearity and constant variance, while a Q-Q plot ensures normality. Cook's Distance identifies influential points, and the Variance Inflation Factor detects multicollinearity. Ensuring these assumptions hold strengthens the model's reliability and helps make accurate predictions.

The following code creates a residuals vs. fitted values plot to evaluate the performance and assumptions of the linear regression model. It first calculates the fitted values and residuals and adds them as new columns to the 'diamond_data' dataset. Using 'ggplot2', it plots the residuals against the fitted values, adding a dashed horizontal line at zero to represent the ideal residual baseline. The plot is customized with a dark blue color for points,

transparency for better visibility, and labeled axes. This diagnostic plot helps check for violations of key regression assumptions, such as linearity, constant variance, and independence, with random scatter around the line at zero indicating a well-fitted model.

```
diamond_data$model_fitted <- fitted(model)
diamond_data$model_residuals <- residuals(model)
ggplot(diamond_data, aes(x = model_fitted, y = model_residuals)) + geom_point(alpha = 0.6, color
= "black") +
geom_hline(yintercept = 0, linetype = "dashed", color = "red") + labs(title = "Residuals vs. Fit
ted Values",x = "Fitted Values",y = "Residuals") +  theme_minimal()
```

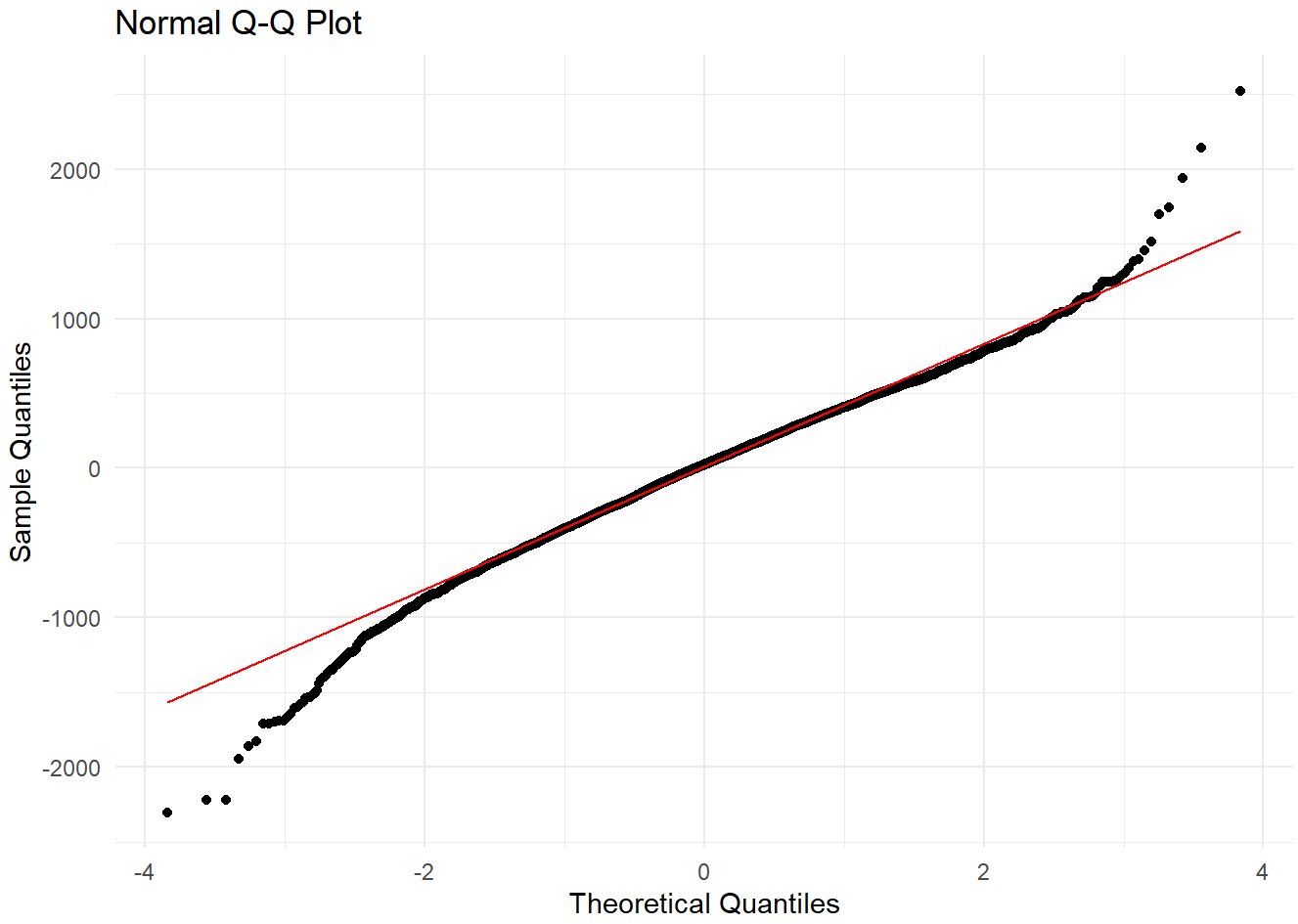


Interpretation : The Residuals vs. Fitted Values plot helps assess the adequacy of the linear regression model. Ideally, if the points are randomly scattered around the horizontal dashed line at zero, it indicates that the model fits the data well, with no systematic pattern in the residuals. This suggests that the assumptions of linearity and homoscedasticity are likely satisfied. However, if the plot shows a distinct pattern, such as a funnel shape or curvature, it indicates potential issues like heteroscedasticity or non-linearity, meaning the model might need adjustments or additional predictors. The semi-transparent black points make it easier to spot overlapping data, while the dashed line at zero represents the ideal scenario where the model's predictions are close to the actual values.

The following code creates a Normal Q-Q plot to assess the normality of residuals from a linear regression model. First, a new data frame 'residuals_df' is created, which contains the residuals from the model and the theoretical quantiles computed using the 'qqnorm()' function . Then, 'ggplot()' is used to generate the Q-Q plot, with 'stat_qq()' plotting the actual sample quantiles against the theoretical quantiles and 'stat_qq_line()' adding a reference line in

purple to indicate the expected alignment for normally distributed data. The plot is titled “Normal Q-Q Plot” and includes axis labels for the theoretical and sample quantiles. The ‘theme_minimal()’ function is applied for a clean and simple plot appearance.

```
 residuals_df <- data.frame(residuals = residuals(model))
 residuals_df$theoretical_quantiles <- qqnorm(residuals_df$residuals, plot.it = FALSE)$x
 ggplot(residuals_df, aes(sample = residuals)) + stat_qq() + stat_qq_line(color = "red") + labs
(title = "Normal Q-Q Plot", x = "Theoretical Quantiles", y = "Sample Quantiles") + theme_minimal
()
```

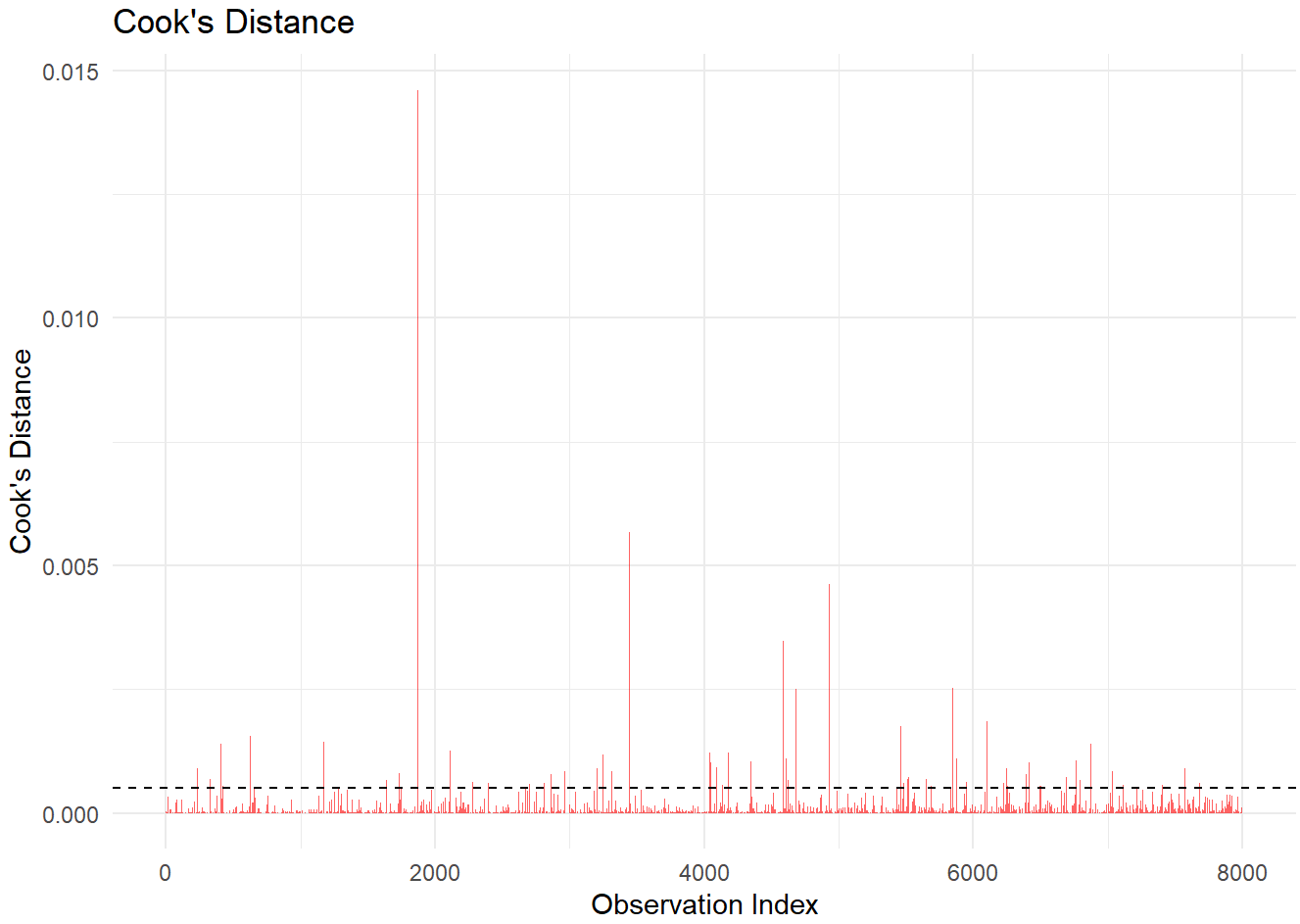


Interpretation: The Normal Q-Q Plot helps assess if the residuals of the regression model follow a normal distribution. If the points closely follow the purple line, it suggests that the residuals are normally distributed, indicating that the normality assumption holds. This is important for the validity of statistical tests like t-tests and confidence intervals. However,if the points deviate significantly from the line, particularly at the ends , it indicates non-normality in the residuals, such as skewness or heavy tails. In such cases, you might need to transform the data or consider alternative modeling approaches to address the violation of normality.

The following code creates a Scale-Location plot to assess the homoscedasticity assumption of the residuals in the regression model. It first calculates the fitted values (model_fitted) and residuals (model_residuals) for the dataset. Then, it computes the standardized residuals using rstandard(model) and applies the square root transformation to the absolute value of these residuals to stabilize their variance. The ggplot() function is used to plot the square root of standardized residuals against the fitted values, with points displayed in dark blue. A red smoothing line is added using geom_smooth() to highlight any patterns in the spread of residuals. The plot is titled “Scale-Location Plot” with appropriately labeled axes. If the points are randomly scattered around the horizontal axis without a clear pattern, it suggests constant variance, while patterns indicate heteroscedasticity .

Cooks Distance: This code calculates and visualizes Cook's Distance to identify influential data points in the regression model. The cooks.distance() function computes Cook's Distance for each observation, which measures the influence of each data point on the model's parameters. The ggplot() function is used to create a bar plot, where the x-axis represents the index of each observation, and the y-axis shows the corresponding Cook's Distance values. The bars are filled with a purple color and partially transparent (alpha=0.6) for better visibility. A dashed horizontal line is added at a threshold value of 4/n (where n is the number of observations) to help identify observations with Cook's Distance greater than this threshold, which are considered influential. The plot is titled "Cook's Distance," and labels are added for the axes. The theme_minimal() function is applied to provide a clean, simple visual style. This plot helps detect outliers or influential points that coul disproportionately affect the regression results.

```
 library(ggplot2)
 model <- lm(price ~ carat + cut + clarity + depth, data = diamond_data)
 cooks_dist <- cooks.distance(model)
 ggplot(data = NULL, aes(x = seq_along(cooks_dist), y = cooks_dist)) + geom_bar(stat = "identit
y", fill = "red", alpha=0.6) + geom_hline(yintercept = 4 / length(cooks_dist), color = "black",
linetype = "dashed") + labs(title = "Cook's Distance", x = "Observation Index", y = "Cook's Dist
ance") + theme_minimal()
```

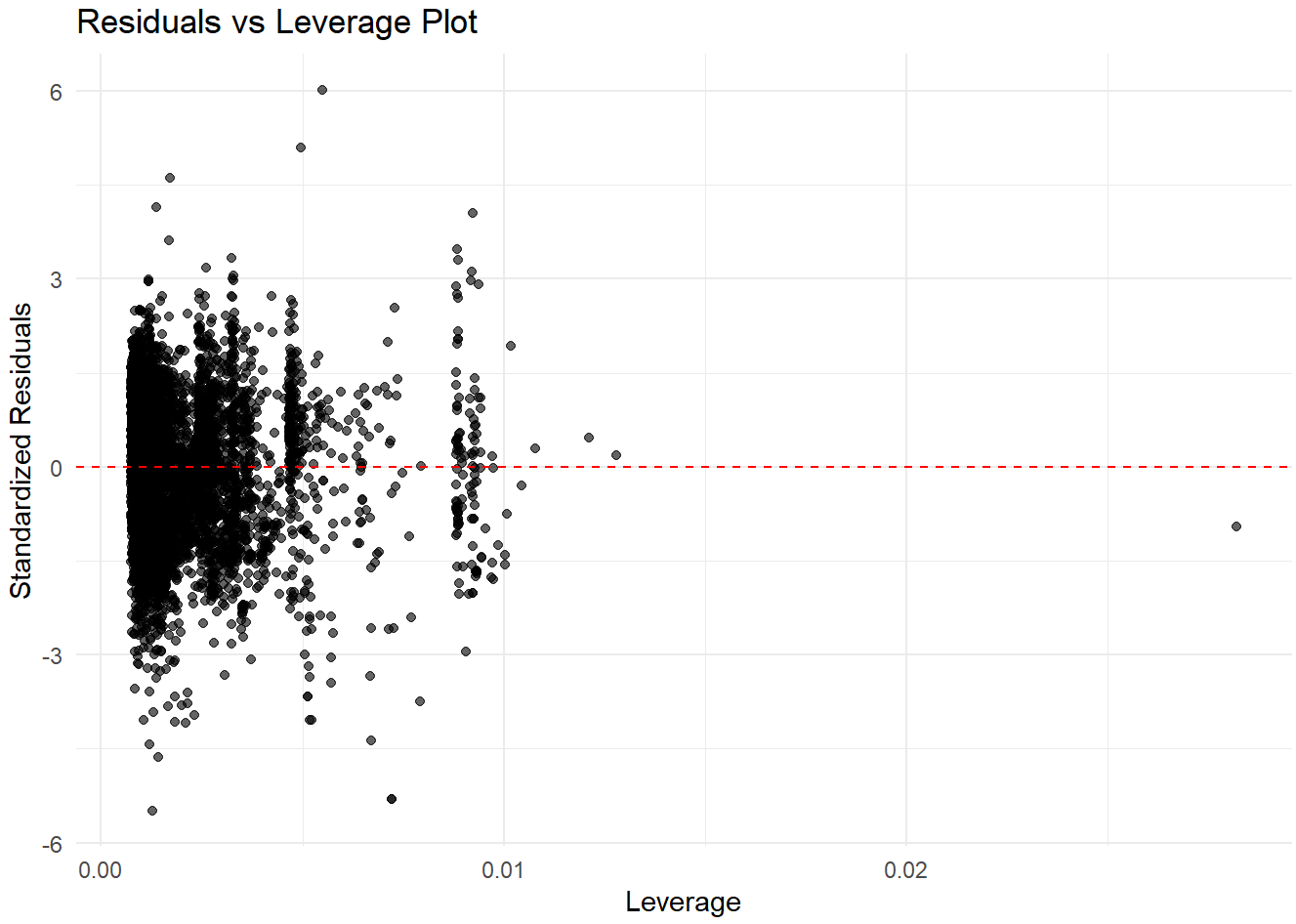


Interpretation: The output of the Cook's Distance plot helps identify influential data points in the regression model. Each bar in the plot represents the Cook's Distance for an individual observation, indicating how much influence that particular data point has on the estimated model parameters. The horizontal dashed line at 4/n (where n is the number of observations) serves as a threshold to identify influential points. If a data point's Cook's Distance exceeds this threshold, it suggests that the point has a disproportionate impact on the model and could be an

outlier .In the plot, observations with large Cook's Distance values should be carefully examined, as they may distort the regression results. Ideally, most bars should be below the threshold, indicating that most observations have minimal influence on the model.

This code generates a Residuals vs. Leverage plot to identify influential data points in the linear regression model. It first fits a linear model to predict price using carat, cut, clarity, and depth as predictors. The hatvalues function calculates the leverage for each observation, which measures how far the predictor values are from the average of all predictors, and the rstandard function calculates standardized residuals, which indicate how far each observation is from the predicted value relative to the overall variance. The ggplot() function is then used to create a scatter plot with leverage on the x-axis and standardized residuals on the y-axis. A horizontal dashed line is added at 0 to indicate the baseline for residuals, and the plot is titled "Residuals vs Leverage Plot" with appropriate axis labels. The plot helps identify points with high leverage or large residuals, which may be influential and could disproportionately affect the model's estimates.

```
model <- lm(price ~ carat + cut + clarity + depth, data = diamond_data)
diamond_data$leverage <- hatvalues(model)
diamond_data$std_residuals <- rstandard(model)
ggplot(diamond_data, aes(x = leverage, y = std_residuals)) + geom_point(alpha = 0.6, color = "bl
ack") +geom_hline(yintercept = 0, linetype = "dashed", color = "red") +  labs(title = "Residuals
vs Leverage Plot", x = "Leverage", y = "Standardized Residuals") + theme_minimal()
```



Interpretation: The Residuals vs. Leverage plot helps identify influential observations in the regression model. Points on the plot represent the relationship between each observation's leverage and its standardized residuals . Points with high leverage and large residuals are considered influential, meaning they have a disproportionate effect on the model's estimates. The horizontal dashed line at 0 indicates the ideal residual value, with points far

above or below this line suggesting poor fit. Observations that are both far from the center and have large residuals should be examined closely, as they could be outliers or influential points that may distort the model's accuracy.

Model Selection: Model selection involves choosing the best regression model by balancing model complexity and fit.Key techniques include using Adjusted R-squared, AIC, and BIC to assess the goodness of fit while penalizing unnecessary complexity.Stepwise selection methods,such as forward selection and backward elimination, iteratively add or remove predictors based on statistical criteria like AIC.Cross-validation helps evaluate model performance on unseen data, ensuring it generalizes well. The goal of model selection is to find the most efficient model that accurately predicts the outcome while avoiding over fitting.

Comparing Models: This R code fits three different linear regression models (model1, model2, and model3) using the lm() function, where each model predicts price based on different sets of predictor variables. model1 uses carat and depth, model2 uses carat, cut, and depth, and model3 uses carat,clarity, and depth. The AIC() function is then used to compute the Akaike Information Criterion for all four models, including the model . AIC is a measure that balances model fit with complexity, where a lower AIC indicates a better model. By comparing the AIC values, the code helps determine which model best fits the data without overfitting.

```
model1 <- lm(price ~ carat + depth, data = diamond_data)
model2 <- lm(price ~ carat + cut + depth, data = diamond_data)
model3 <- lm(price ~ carat + clarity + depth, data = diamond_data)
AIC(model, model1, model2, model3)
```

```
##        df      AIC
## model  15 119355.1
## model1  4 123305.0
## model2  8 122800.8
## model3 11 119915.8
```

Interpretation : The AIC and BIC values indicate that model (price ~ carat + cut + clarity + depth) is the best model among the four, as it has the lowest AIC (119355.1) and BIC (119459.9), suggesting it strikes the best balance between model fit and complexity.Model1 (price ~ carat + depth) has the highest AIC (123305.0) and BIC (123333.0), making it the least effective model in terms of predictive accuracy and complexity.Model2 (price ~ carat + cut + depth) and Model3 (price ~ carat + clarity + depth) have intermediate AIC and BIC values, indicating that while they fit the data reasonably well, they are not as efficient as the full model in explaining diamond price

```
 BIC(model, model1, model2, model3)
```

```
##        df      BIC
## model  15 119459.9
## model1  4 123333.0
## model2  8 122856.7
## model3 11 119992.7
```

Stepwise Selection : This R code performs forward stepwise regression to select the best subset of predictors for the regression model. It starts by fitting an empty model with only the intercept , which serves as the baseline. Then, it defines a full model that includes all available predictors (carat,cut,clarity, and depth). The step() function is

used to perform forward selection, where predictors are added one by one from the empty_model to the full_model based on statistical criteria . The selection process continues until no further improvements can be made, resulting in the best model with the most relevant predictors.

```
empty_model <- lm(price ~ 1, data = diamond_data)
full_model <- lm(price ~ carat + cut + clarity + depth, data = diamond_data)
step(empty_model, scope = list(lower = empty_model, upper = full_model), direction = "forward")
```

```
## Start:  AIC=111202
## price ~ 1
##
##           Df  Sum of Sq        RSS    AIC
## + carat    1 6385804590 2333533433 100660
## + clarity  7  581790957 8137547065 110664
## + cut      4   95215326 8624122697 111122
## + depth    1    6178145 8713159877 111198
## <none>                  8719338023 111202
##
## Step:  AIC=100660
## price ~ carat
##
##           Df Sum of Sq        RSS    AIC
## + clarity  7 801998395 1531535038  97306
## + cut      4 157396600 2176136833 100109
## + depth    1  17200208 2316333225 100603
## <none>                 2333533433 100660
##
## Step:  AIC=97305.55
## price ~ carat + clarity
##
##         Df Sum of Sq        RSS   AIC
## + cut    4 118465852 1413069186 96670
## + depth  1  17872645 1513662393 97214
## <none>               1531535038 97306
##
## Step:  AIC=96669.58
## price ~ carat + clarity + cut
##
##         Df Sum of Sq        RSS   AIC
## + depth  1   3288570 1409780616 96653
## <none>               1413069186 96670
##
## Step:  AIC=96652.95
## price ~ carat + clarity + cut + depth
```

```
##
## Call:
## lm(formula = price ~ carat + clarity + cut + depth, data = diamond_data)
##
## Coefficients:
##  (Intercept)         carat     clarityIF    claritySI1    claritySI2
##     -1763.23       4960.37       2162.37       1414.27       1130.44
##   clarityVS1    clarityVS2   clarityVVS1   clarityVVS2       cutGood
##      1700.80       1552.49       2057.62       1895.58        319.69
##     cutIdeal    cutPremium  cutVery Good         depth
##       497.65        280.53        387.06        -13.47
```

```
step(full_model, direction = "backward")
```

```
## Start:  AIC=96652.95
## price ~ carat + cut + clarity + depth
##
##           Df  Sum of Sq        RSS    AIC
## <none>                  1409780616  96653
## - depth    1    3288570 1413069186  96670
## - cut      4  103881777 1513662393  97214
## - clarity  7  762859172 2172639788 100099
## - carat    1 6704023781 8113804397 110650
```

```
##
## Call:
## lm(formula = price ~ carat + cut + clarity + depth, data = diamond_data)
##
## Coefficients:
##  (Intercept)         carat       cutGood      cutIdeal    cutPremium
##     -1763.23       4960.37        319.69        497.65        280.53
## cutVery Good     clarityIF    claritySI1    claritySI2    clarityVS1
##       387.06       2162.37       1414.27       1130.44       1700.80
##   clarityVS2   clarityVVS1   clarityVVS2         depth
##      1552.49       2057.62       1895.58        -13.47
```

Interpretation: The stepwise regression process, using AIC as the selection criterion, identified the most significant predictors for diamond price. Starting with an empty model, carat was added first, significantly reducing the AIC to 100660. Adding clarity further improved the model, lowering the AIC to 97306, followed by cut, which resulted in an AIC of 96670. Finally, depth was included, bringing the AIC to 96653, which indicated the best-fitting model. The backward elimination process also confirmed that all four variables carat,clarity,cut, and depth were important, as removing any of them led to higher AIC values. Therefore, the final model includes these four predictors, which best explain the variation in diamond prices

#* Conclusion and Discussion:* This analysis showed that diamond prices are mainly influenced by carat weight, cut quality, and clarity, with carat weight having the most significant effect. Higher-quality cuts and clarity grades also contribute to higher prices, while depth has a smaller, negative impact.
The research questions were addressed effectively:

1. **Key factors influencing diamond pricing**: Carat, cut, and clarity were identified as major determinants.
2. **Effect of carat weight**: Heavier diamonds are significantly more expensive.
3. **Impact of cut and clarity**: Higher grades in these qualities add value to the price.

Limitations: Some factors, such as diamond color or market trends, were not included in the analysis, and there were minor signs of model assumption violations like non-linearity or varying variances. These could affect the accuracy of the results.
Future Improvements: Adding more variables and addressing assumption issues could improve the model. Testing the model on new data could also validate its reliability.

#Further Analysis: Robust Regression: Residual diagnostics from the original linear regression model revealed the presence of influential points, as indicated by Cook's Distance and other diagnostic plots. To mitigate the effects of these outliers, robust regression was employed. Robust regression is less sensitive to outliers and provides more stable coefficient estimates, ensuring that the model's results are not unduly influenced by extreme observations. Methodology: Using the "MASS" package in R, a robust regression model was fit to the diamond dataset. The method minimizes the influence of outliers by assigning weights to observations based on their residuals.

```
library(MASS)
robust_model <- rlm(price ~ carat + cut + clarity + depth, data = diamond_data)
summary(robust_model)
```

```
##
## Call: rlm(formula = price ~ carat + cut + clarity + depth, data = diamond_data)
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2383.95  -268.66    14.53   270.70  2581.25
##
## Coefficients:
##              Value      Std. Error t value
## (Intercept)  -1875.1810   200.7935    -9.3389
## carat         5117.6601    24.8796   205.6968
## cutGood        335.8979    24.5238    13.6968
## cutIdeal       514.0619    23.5021    21.8730
## cutPremium     303.4537    24.0187    12.6340
## cutVery Good   400.7224    23.4622    17.0795
## clarityIF     2076.0824    48.6407    42.6820
## claritySI1    1366.1613    29.7446    45.9298
## claritySI2    1074.6885    28.8295    37.2774
## clarityVS1    1658.7681    31.8392    52.0983
## clarityVS2    1520.7542    30.9039    49.2092
## clarityVVS1   2002.3750    37.9495    52.7642
## clarityVVS2   1847.0805    35.6319    51.8378
## depth          -13.0997     3.0512    -4.2934
##
## Residual standard error: 400.1 on 7985 degrees of freedom
```

The robust regression results showed the following coefficient estimates: -> Carat remained the most significant predictor, with a coefficient similar to the original linear regression model. -> Cut and clarity continued to have a strong influence on price, while depth had a smaller negative effect. -> The robust model reduced the impact of extreme data points, as reflected in improved residual patterns.

The robust regression model provided results consistent with the OLS model in terms of the significance of predictors but showed improved residual behavior. The influence of extreme data points was mitigated, as observed in a reduced range of residuals and Cook's Distance values. **Discussion:** Robust regression offers a valuable approach to improving model reliability when outliers or influential points are present. By reducing their impact, the robust model produces more stable coefficient estimates, making it a better choice for datasets like this. While the original OLS model provided similar insights, the robust regression ensures the results are less sensitive to unusual data points, enhancing their interpretability and robustness.