Regression Analysis of Diamond Prices

Lucas Childs

2025-03-17

Part 1 - Data Description and Descriptive Statistics

Regression is an essential tool used to make quantitative predictions. This project will focus on using it to predict diamond prices given the `Diamonds` dataset from Kaggle.

First, one must understand the variables and their relationships.

1)

```r
set.seed(203854)
library(ggplot2)
library(gridExtra)
diamonds <- read.csv("Diamonds_Prices_2022.csv",
                     colClasses = c("carat" = "numeric",
                                    "cut" = "factor",
                                    "color" = "factor",
                                    "clarity" = "factor",
                                    "depth" = "numeric",
                                    "table" = "numeric",
                                    "price" = "numeric",
                                    "x" = "numeric",
                                    "y" = "numeric",
                                    "z" = "numeric"))

# Random sample of 1000 observations
n <- 1000

diamonds_sample <- diamonds[sample(nrow(diamonds), size = n), ]

# Remove index variable
diamonds_sample <- subset(diamonds_sample, select = -c(X))

# Rename real indices
rownames(diamonds_sample) <- 1:nrow(diamonds_sample)
```

2)

```r
summary(diamonds_sample)
```

```
##      carat                cut      color      clarity        depth      
##  Min.   :0.2000   Fair     : 27   D:131   SI1    :232   Min.   :55.20  
##  1st Qu.:0.3975   Good     : 79   E:171   VS2    :216   1st Qu.:61.10  
##  Median :0.7000   Ideal    :400   F:172   SI2    :173   Median :61.90  
##  Mean   :0.7917   Premium  :263   G:213   VS1    :157   Mean   :61.79  
##  3rd Qu.:1.0400   Very Good:231   H:159   VVS2   :108   3rd Qu.:62.50  
##  Max.   :2.6100                   I:104   VVS1   : 74   Max.   :68.60  
##                                   J: 50   (Other): 40                  
##      table           price             x               y        
##  Min.   :53.00   Min.   :  367   Min.   :3.860   Min.   :3.840  
##  1st Qu.:56.00   1st Qu.:  914   1st Qu.:4.688   1st Qu.:4.690  
##  Median :57.00   Median : 2366   Median :5.680   Median :5.685  
##  Mean   :57.45   Mean   : 3937   Mean   :5.713   Mean   :5.709  
##  3rd Qu.:59.00   3rd Qu.: 5306   3rd Qu.:6.520   3rd Qu.:6.510  
##  Max.   :73.00   Max.   :18768   Max.   :8.760   Max.   :8.700  
##                                                                 
##        z        
##  Min.   :2.300  
##  1st Qu.:2.890  
##  Median :3.500  
##  Mean   :3.528  
##  3rd Qu.:4.040  
##  Max.   :5.530  
## 
```

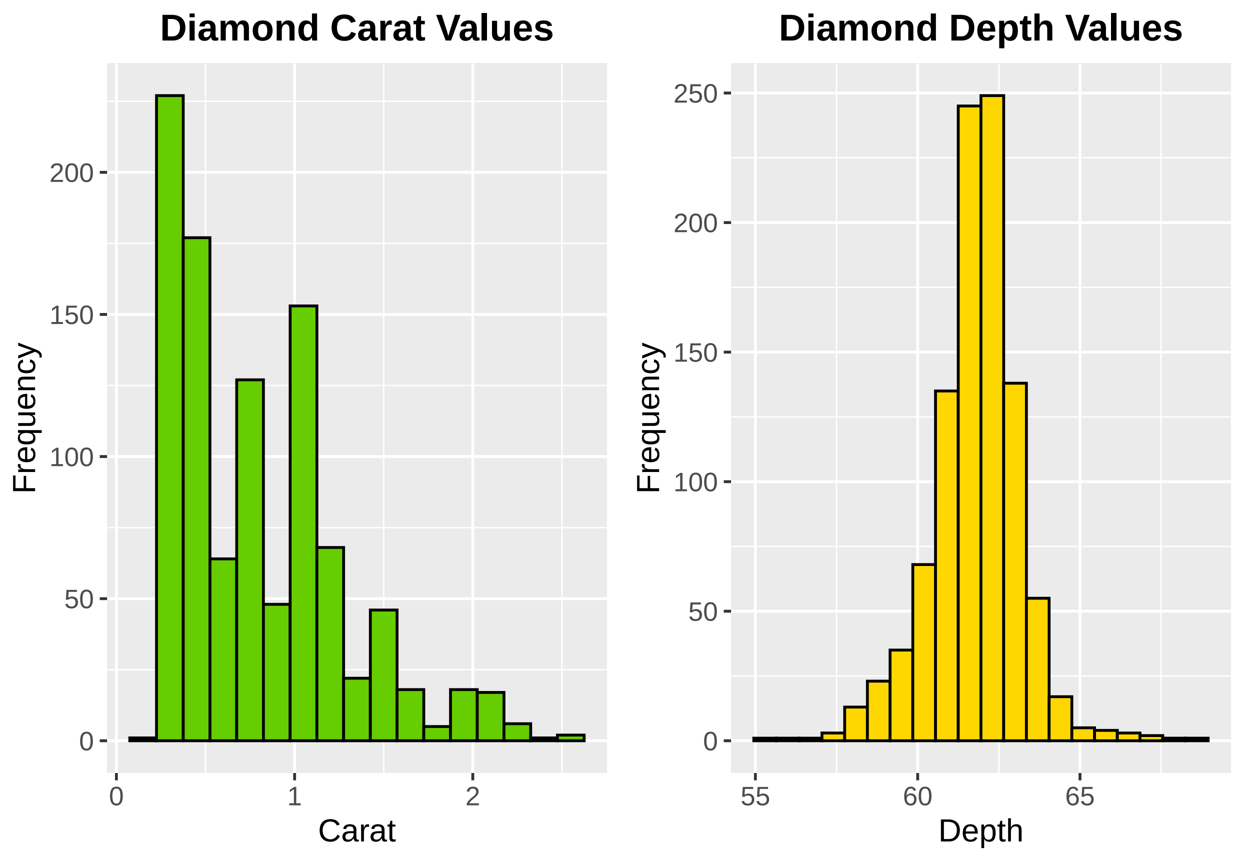
```r
str(diamonds_sample)
```

```
## 'data.frame':    1000 obs. of  10 variables:
##  $ carat  : num  1.22 0.3 1.5 0.75 1.08 0.58 0.31 1.16 0.67 1.42 ...
##  $ cut    : Factor w/ 5 levels "Fair","Good",..: 5 3 5 4 3 4 4 3 4 4 ...
##  $ color  : Factor w/ 7 levels "D","E","F","G",..: 2 1 4 4 5 2 6 4 3 3 ...
##  $ clarity: Factor w/ 8 levels "I1","IF","SI1",..: 4 6 4 4 4 3 3 4 6 5 ...
##  $ depth  : num  63.3 60.9 60.7 62.7 62.1 61.6 59.6 61.9 62.5 58.4 ...
##  $ table  : num  56 57 61 58 56 59 58 56 59 59 ...
##  $ price  : num  5739 911 8580 2268 4344 ...
##  $ x      : num  6.78 4.35 7.33 5.82 6.54 5.38 4.44 6.73 5.58 7.36 ...
##  $ y      : num  6.75 4.32 7.36 5.76 6.6 5.33 4.39 6.77 5.53 7.32 ...
##  $ z      : num  4.28 2.64 4.46 3.63 4.08 3.3 2.63 4.18 3.47 4.29 ...
```

```r
g1 <- ggplot(data = diamonds_sample, mapping = aes(x = carat)) +
  geom_histogram(binwidth = 0.15,fill = "chartreuse3", color = "black") +
  ggtitle("Diamond Carat Values") +
  xlab("Carat") + ylab("Frequency") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10))

g2 <- ggplot(data = diamonds_sample, mapping = aes(x = depth)) +
  geom_histogram(binwidth = 0.7, fill = "gold1", color = "black") +
  ggtitle("Diamond Depth Values") +
  xlab("Depth") + ylab("Frequency") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10))

grid.arrange(g1, g2, ncol = 2)
```

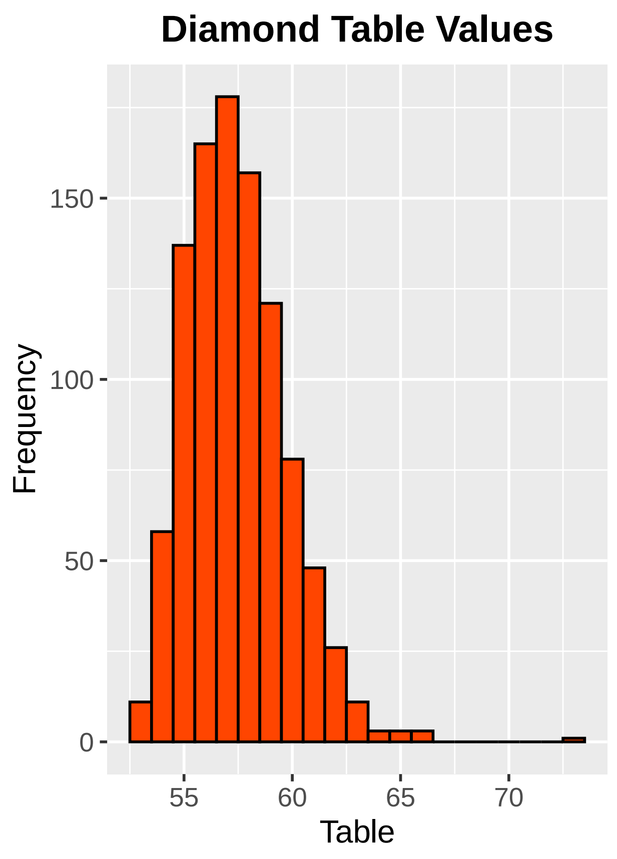


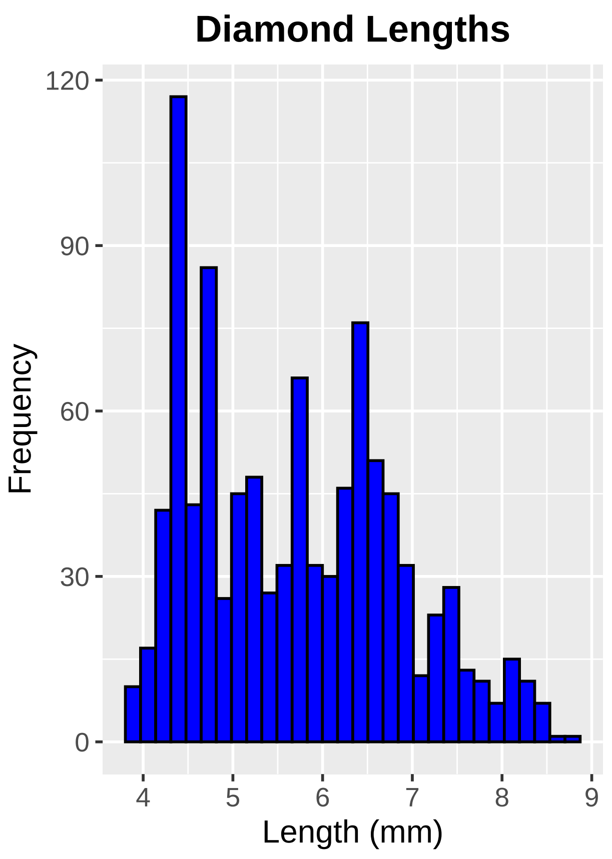
Based on the carat histogram, most diamonds are about 0.3 carats with very few above 2 carats. Diamond depth appears normal, with most depth about 62 (unknown units).

```r
g3 <- ggplot(data = diamonds_sample, mapping = aes(x = table)) +
  geom_histogram(binwidth = 1, fill = "orangered", color = "black") +
  ggtitle("Diamond Table Values") +
  xlab("Table") + ylab("Frequency") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10))

g4 <- ggplot(data = diamonds_sample, mapping = aes(x = x)) +
  geom_histogram(fill = "blue", color = "black") +
  ggtitle("Diamond Lengths") +
  xlab("Length (mm)") + ylab("Frequency") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10))
```

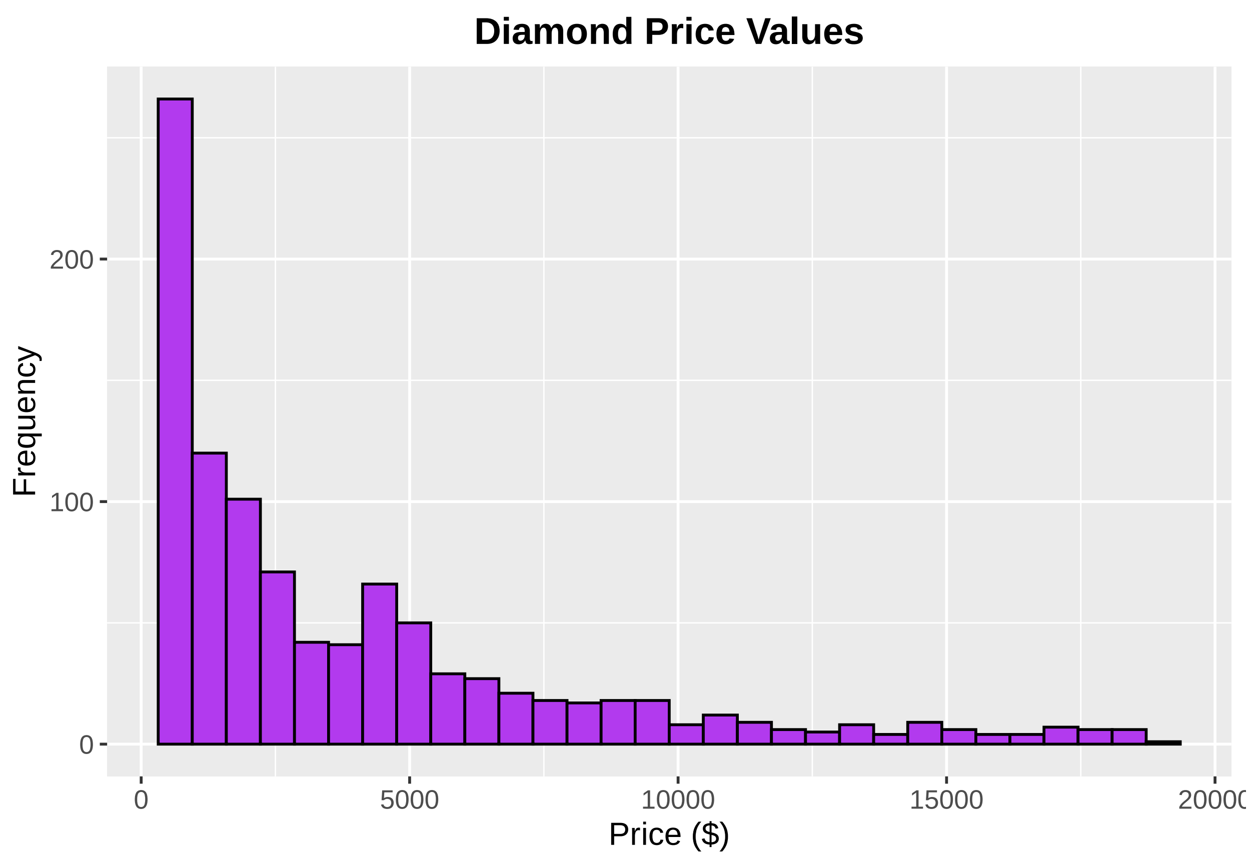
```r
grid.arrange(g3, g4, ncol = 2)
```





Diamond table also appears normal with most values at around 56. Diamond length is most frequent at 4.4 mm with few diamonds deeper than 8.5 mm.

```r
ggplot(data = diamonds_sample, mapping = aes(x = price)) +
  geom_histogram(fill = "darkorchid2", color = "black") +
  ggtitle("Diamond Price Values") +
  xlab("Price ($)") + ylab("Frequency") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10))
```

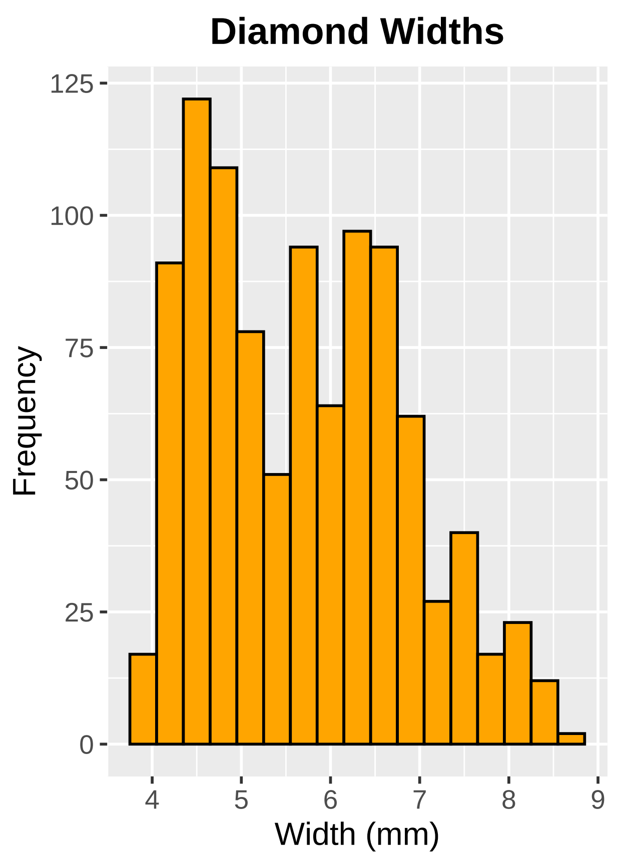


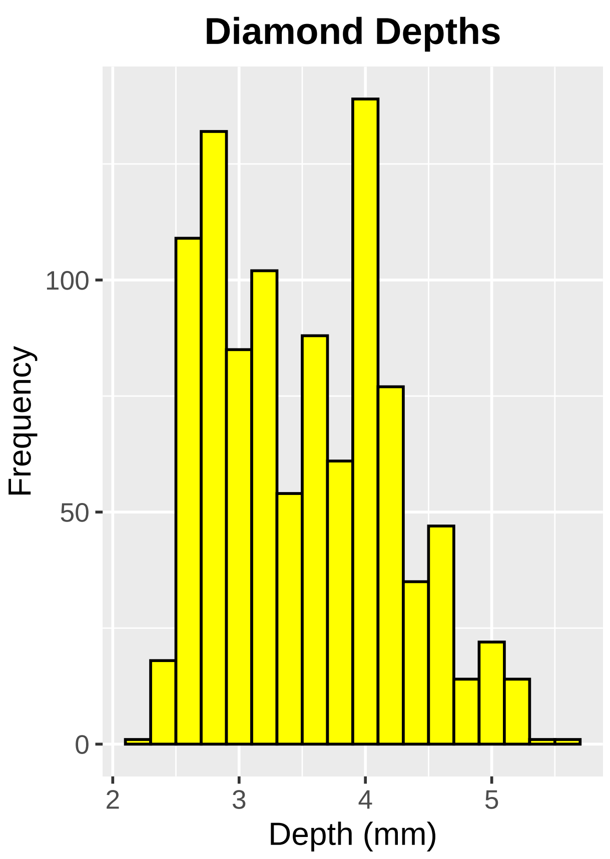
Diamond price is right skewed, with most diamonds priced at approximately $600.

```r
g5 <- ggplot(data = diamonds_sample, mapping = aes(x = y)) +
  geom_histogram(binwidth = 0.3, fill = "orange", color = "black") +
  ggtitle("Diamond Widths") +
  xlab("Width (mm)") + ylab("Frequency") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10))

g6 <- ggplot(data = diamonds_sample, mapping = aes(x = z)) +
  geom_histogram(binwidth = 0.2, fill = "yellow", color = "black") +
  ggtitle("Diamond Depths") +
  xlab("Depth (mm)") + ylab("Frequency") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10))

grid.arrange(g5, g6, ncol = 2)
```

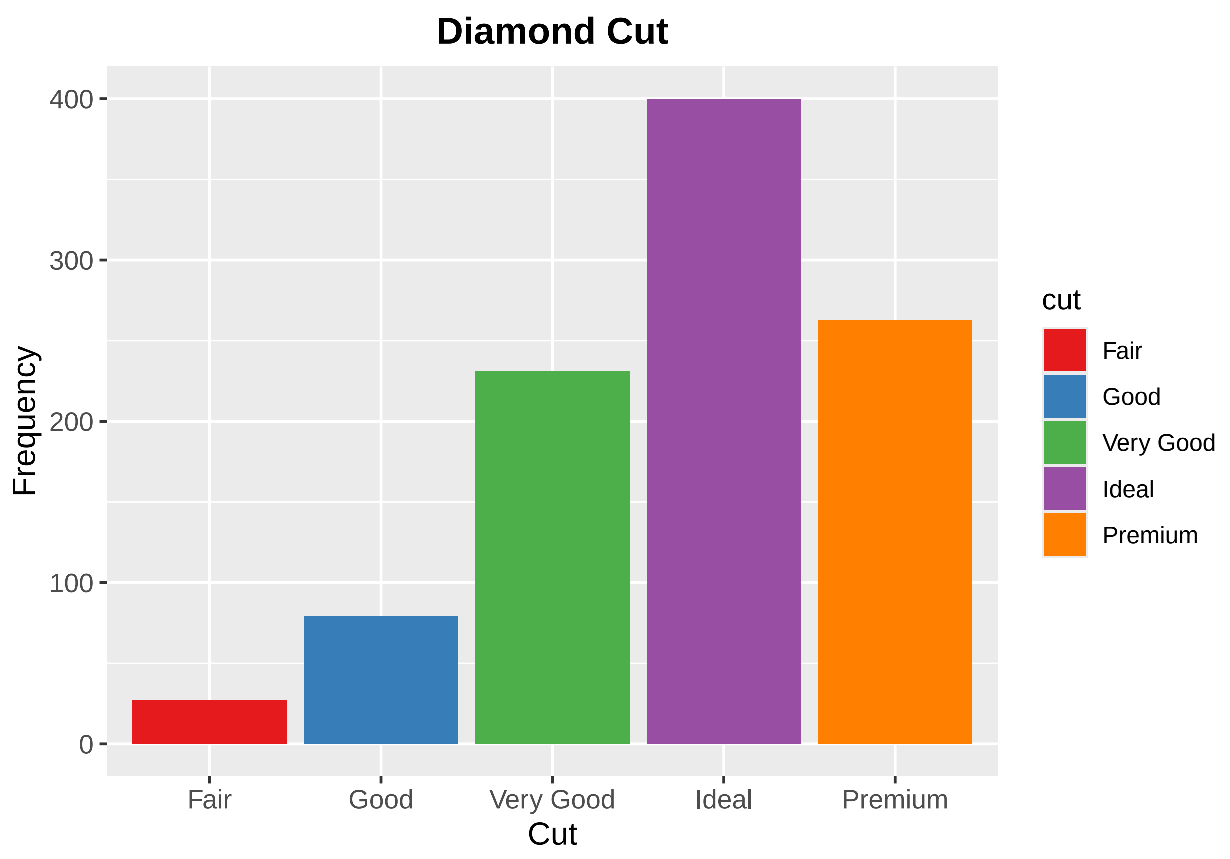




Both diamond width and depth in mm are similarly distributed and appear roughly normal. The diamonds are more often wide than they are deep.

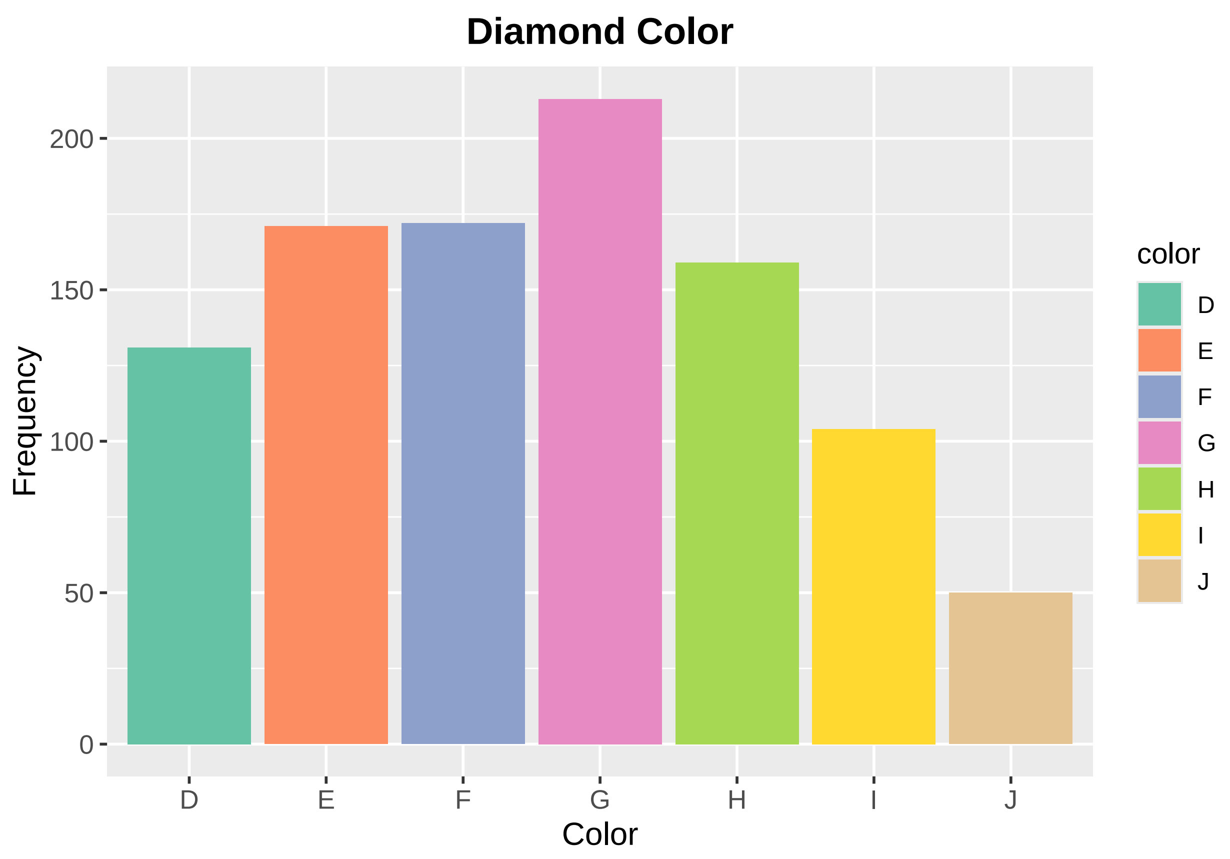
```
diamonds_sample$cut <- factor(diamonds_sample$cut, levels = c("Fair", "Good", "Very Good", "Ideal", "Pr

ggplot(data = diamonds_sample, mapping = aes(x = cut, fill = cut)) +
  geom_bar() +
  xlab("Cut") +
  ylab("Frequency") +
  ggtitle("Diamond Cut") +
  scale_x_discrete() +
  scale_fill_brewer(palette = "Set1") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10))
```



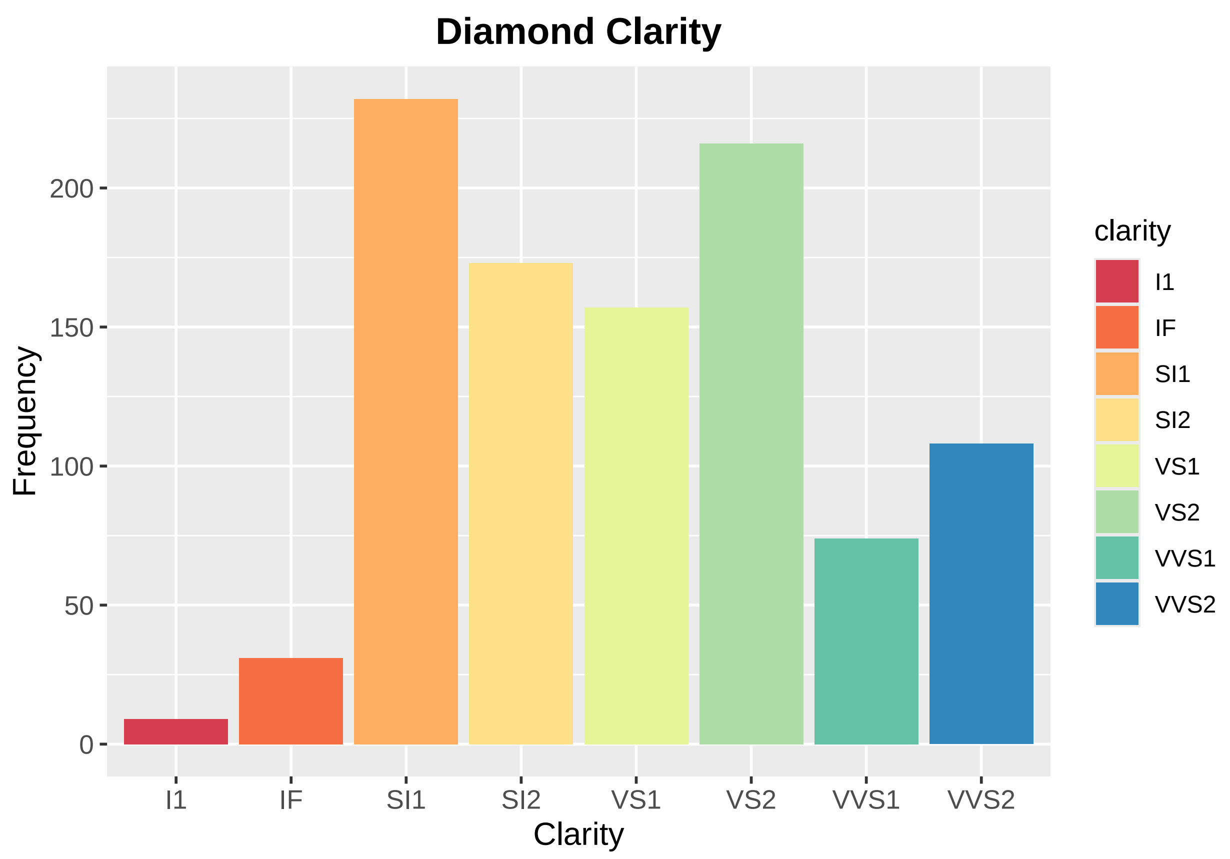
Evidently, there are the most diamonds with an Ideal cut where the next most common are Premium and then Very Good cuts.

```
ggplot(data = diamonds_sample, mapping = aes(x = color, fill = color)) +
  geom_bar() +
  xlab("Color") +
  ylab("Frequency") +
  ggtitle("Diamond Color") +
  scale_fill_brewer(palette = "Set2") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10))
```



The most common diamond color is G, followed by E and F in the colorless category.

```
ggplot(data = diamonds_sample, mapping = aes(x = clarity, fill = clarity)) +
  geom_bar() +
  xlab("Clarity") +
  ylab("Frequency") +
  ggtitle("Diamond Clarity") +
  scale_fill_brewer(palette = "Spectral") +
  theme(plot.title = element_text(hjust = 0.5, face = "bold", size = 14),
        axis.title = element_text(size = 12),
        axis.text = element_text(size = 10))
```



The most common diamond clarity metric is VS2, followed by SI1 and SI2.

3)

Choose 3 quantitative and 2 categorical variables intuitively:

```
# I chose price, carat, depth, color, clarity
model1 <- lm(price ~ carat + depth + color + clarity, data = diamonds_sample)
```

Examine correlation between the variables:

```
# Run the correlation matrix on just the quantitative variables (and variables we've chosen)
cor(diamonds_sample[, -c(2, 3, 4, 6, 8, 9, 10)])
```

```
##           carat        depth        price
## carat 1.0000000  0.030551105  0.926960988
## depth 0.0305511  1.000000000 -0.003093351
## price 0.9269610 -0.003093351  1.000000000
```

Out of the quantitative variables I chose: `price`, `carat`, and `depth`, only `price` and `carat` are very correlated, `depth` is not very correlated at all to the other variables.

4)

Multiple Linear regression:

```r
summary(model1)
```

```
##
## Call:
## lm(formula = price ~ carat + depth + color + clarity, data = diamonds_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4327.1  -680.0  -225.8   518.5  6059.4
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5771.25    1791.39  -3.222  0.00132 **
## carat        8957.76      90.55  98.921  < 2e-16 ***
## depth         -19.67      27.62  -0.712  0.47652
## colorE       -167.91     138.68  -1.211  0.22628
## colorF       -417.91     139.68  -2.992  0.00284 **
## colorG       -343.89     135.54  -2.537  0.01133 *
## colorH       -850.67     143.33  -5.935 4.07e-09 ***
## colorI      -1313.96     162.47  -8.088 1.78e-15 ***
## colorJ      -2328.68     202.45 -11.502  < 2e-16 ***
## clarityIF    5924.95     457.61  12.948  < 2e-16 ***
## claritySI1   3973.54     407.44   9.753  < 2e-16 ***
## claritySI2   3079.32     408.64   7.535 1.10e-13 ***
## clarityVS1   4900.58     412.31  11.886  < 2e-16 ***
## clarityVS2   4634.69     409.84  11.309  < 2e-16 ***
## clarityVVS1  5574.18     428.19  13.018  < 2e-16 ***
## clarityVVS2  5305.51     420.56  12.615  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1188 on 984 degrees of freedom
## Multiple R-squared:  0.9156, Adjusted R-squared:  0.9143
## F-statistic: 711.5 on 15 and 984 DF,  p-value: < 2.2e-16
```

5)

Per the summary statistics, the overall model p-value is statistically significant, and most predictors are as well, except for `depth` and "colorE". Adjusted R^2 is relatively high and so is residual standard error.

Part 2 - Simple Linear Regression Comparison

1)

Simple linear regression model:

```r
slr <- lm(price ~ carat, data = diamonds_sample)
```

2)

Summary statistics interpretation:

```r
summary(slr)
```

```
##
## Call:
## lm(formula = price ~ carat, data = diamonds_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6257.3  -858.3    38.6   555.0  8123.6
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2338.54      93.71  -24.95   <2e-16 ***
## carat        7926.94     101.55   78.06   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1523 on 998 degrees of freedom
## Multiple R-squared:  0.8593, Adjusted R-squared:  0.8591
## F-statistic:  6093 on 1 and 998 DF,  p-value: < 2.2e-16
```

With a p-value of 2.2e-16, we reject the null hypothesis H_0 at significance level $\alpha = 0.05$ and conclude that we have statistically significant evidence that carat impacts diamond price. The F-statistic is used to calculate this p-value, in which we want a higher F-statistic so that unexplained variance is low.

Adjusted R^2 represents the change in y, diamond price, explained by the combination of the independent variables, which is just carat in the simple linear regression model. Since multiple R^2 increases as more predictors are added to the model, adjusted R^2 accounts for this by dividing SSE and SST by their respective degrees of freedom. But since there is only one predictor, we jut look at normal R^2 which is multiple R^2, 0.86, a pretty good fit, but it could be better since the best R^2 would be 1.

Residual standard error represents the average deviation of the actual data points from the predicted values of the model, where a simple linear regression model was used in this instance. So on average, the model deviates 1523 dollars from the actual diamond prices.

Under "Residuals", Min represents the distance between the regression line and the data point farthest below it, 1Q is the first quartile so 25% of residuals lie below this value, Median represents the median so the value that is separating the lower half of the values from the upper half, 3Q is the third quartile so 25% of residuals lie above this value, and Max represents the distance between the regression line and the data point farthest above it. So, the data point furthest underneath the fitted regression line is -6257 dollars away, 25% of the residuals lies below -858, the median is 38.6, 25% of residuals lie above 555, and the distance between the regression line and the data point farthest above it is 8124 dollars.

Under "Coefficients" the intercept estimate, -2339, represents the expected diamond price when not considering carats, 7927 represents the change in diamond price for every 1 carat added to the diamond. The standard errors of the estimates represents the measure of the average deviation of the estimated coefficients from its true population value. So the intercept deviates an average of 94 dollars from the true intercept value and the coefficient for carat deviates an average of 102 dollars from the true slope coefficient value.

The t-values represent how many standard errors the estimated coefficients are from 0, so the intercept is -25 standard errors away from 0 while the slope term is 78 standard errors away from 0. Finally, the p-value of carat tells us that carat has a statistically significant effect on diamond price, which is the same p-value as the global p-value, since there is only one predictor.

```
confint(slr, level = 0.95)
```

```
##                 2.5 %    97.5 %
## (Intercept) -2522.433 -2154.641
## carat        7727.656  8126.220
```

```
predict(slr, interval = "prediction", level = 0.95)[1:5, ]
```

```
##        fit       lwr       upr
## 1 7332.327  4341.886 10322.767
## 2   39.544 -2951.284  3030.372
## 3 9551.869  6559.316 12544.423
## 4 3606.666   617.432  6595.900
## 5 6222.555  3232.781  9212.330
```

The 95% confidence interval is interpreted as: out of 100 intervals, 95 of them will contain the true intercept and slope coefficient, between -2522 and -2155 and between 7728 and 8126, respectively.

In the 95% prediction interval, "fit" is the predicted diamond price of each observation that the simple linear regression model outputted. The lower and upper bounds represent the interval that the true price could take. The wide interval is due to the high residual standard error that is present in this model. And, out of 100 intervals, 95 of them will contain the true i-th data point.

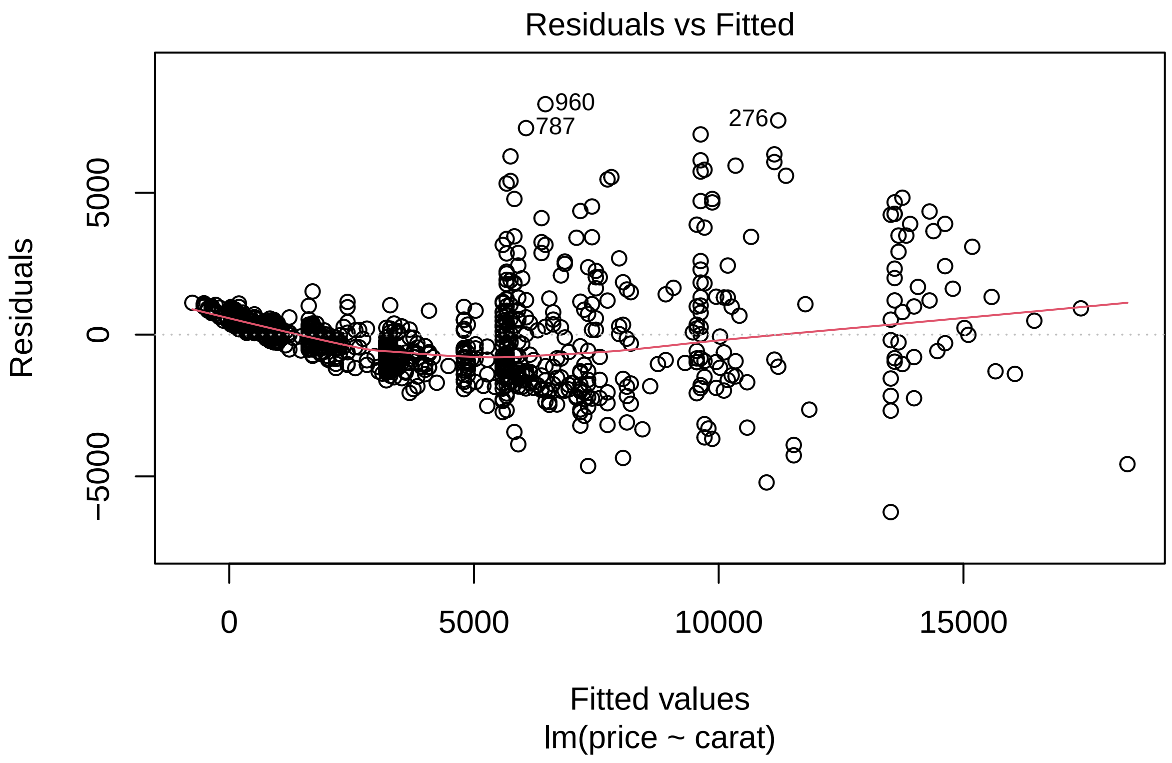
Examining the plot of residuals vs fitted values gives insight into whether the model is linear or not and maintains homoscedasticity. This will be analysed in the next question, testing the linear regression model assumptions.

3)

Linear regression model assumptions:

Linearity:

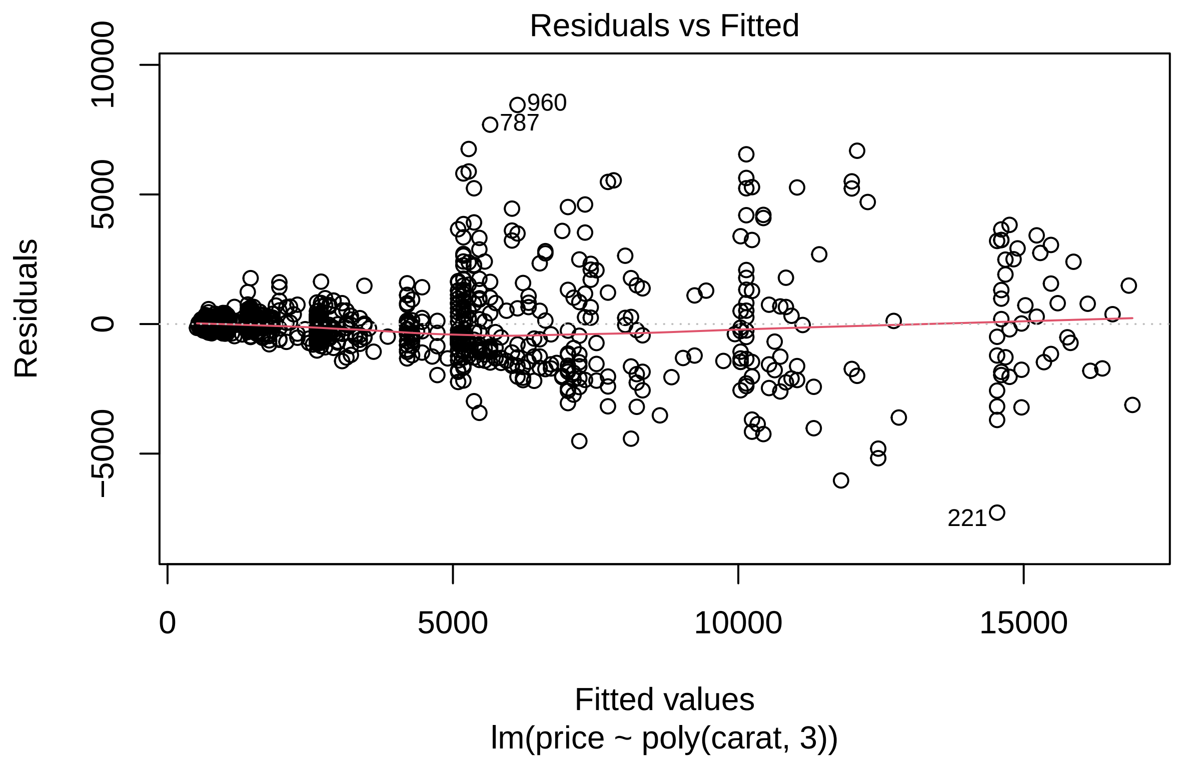
```r
plot(slr, which = 1)
```



The residuals vs fitted values plot shows a slight pattern away from 0 and the residuals follow a curved, non-linear pattern. To remedy this, let's transform the predictor, carat, with a polynomial.

```r
slr_2 <- lm(price ~ poly(carat, 3), data = diamonds_sample)

plot(slr_2, which = 1)
```



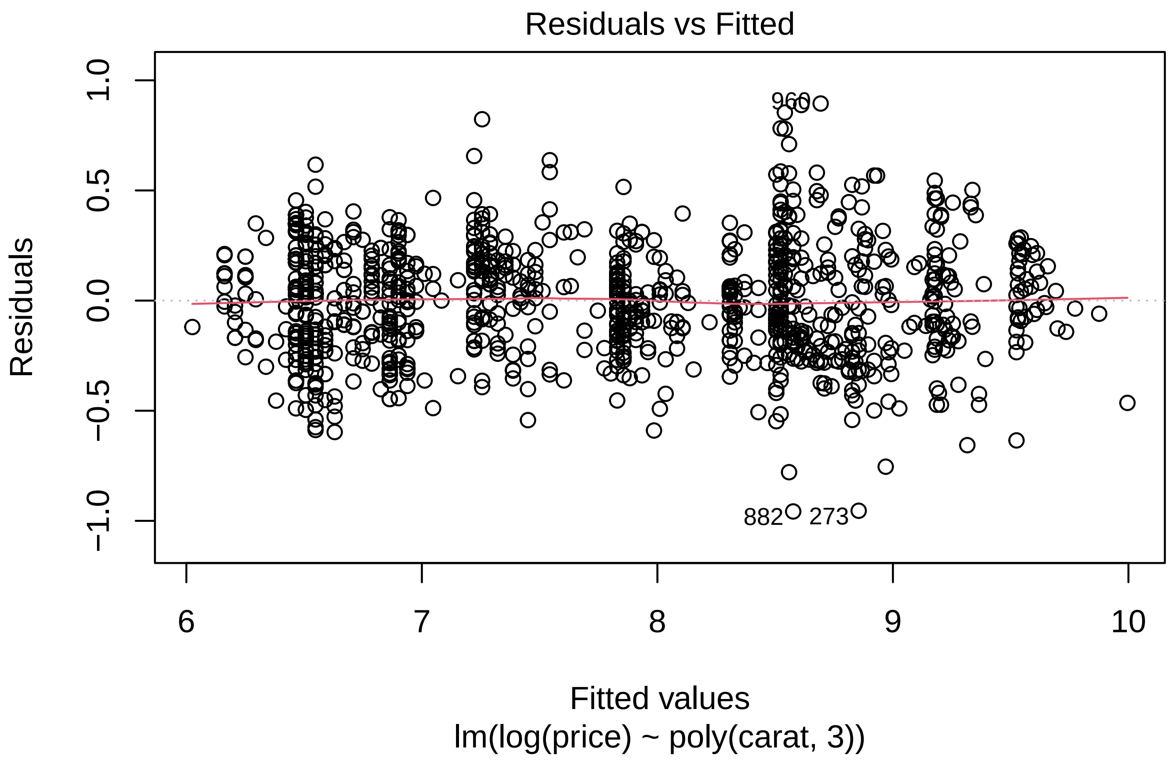
Now the residuals are randomly scattered around 0 and follow a more linear relationship.

However, variance does not appear constant since the residuals fan out. Let's try using a logarithmic function to transform the response.

Homoscedasticity:

```
slr_3 <- lm(log(price) ~ poly(carat, 3), data = diamonds_sample)

plot(slr_3, which = 1)
```



Now the residuals have constant variance, homoscedasticity, since their vertical spread is more consistent throughout the fitted values.

Normality of the residuals is not a concern because we have a sufficiently large sample size of $n = 1000$, so the residuals will approximate to a normal distribution by the central limit theorem.

4)

```
summary(slr_3)
```

```
##
## Call:
## lm(formula = log(price) ~ poly(carat, 3), data = diamonds_sample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.95739 -0.17599 -0.00307  0.17676  0.89455
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)      7.773262   0.008012 970.157  < 2e-16 ***
## poly(carat, 3)1 30.165444   0.253374 119.055  < 2e-16 ***
## poly(carat, 3)2 -9.164624   0.253374 -36.170  < 2e-16 ***
## poly(carat, 3)3  1.976103   0.253374   7.799 1.57e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2534 on 996 degrees of freedom
## Multiple R-squared:  0.9398, Adjusted R-squared:  0.9396
## F-statistic:  5181 on 3 and 996 DF,  p-value: < 2.2e-16
```

After the required transformations, R^2 increased from 0.86 to 0.94 and residual standard error decreased from 1523 dollars to 0.25 dollars. So, the fit of the model is much better now.

5)

Testing more predictors (assessing adjusted R^2):

```
model4 <- lm(log(price) ~ poly(carat, 3) + depth + color + clarity,
             data = diamonds_sample)
```

After adding the predictor `depth` into the transformed model, adjusted R^2 increased from 0.9396 to 0.9401. Then, adding `color` to this model increased adjusted R^2 to 0.952. Furthermore, `clarity` increased adjusted R^2 to 0.983.

6)

Multicollinearity

```
faraway::vif(model4)
```

```
## poly(carat, 3)1 poly(carat, 3)2 poly(carat, 3)3           depth          colorE
##        1.308211        1.065089        1.051713        1.039785        1.934645
##          colorF          colorG          colorH          colorI          colorJ
##        1.983807        2.189185        1.952729        1.766067        1.384420
##       clarityIF      claritySI1      claritySI2      clarityVS1      clarityVS2
##        4.473862       21.122803       17.068201       15.994517       20.240409
##     clarityVVS1     clarityVVS2
##        8.941582       12.122205
```

Using VIF, it can be seen that some levels of the categorical variable `clarity` have VIF values higher than 10, but VIF measures relationships among variables, not between categorical variable levels, so we'll keep them unchanged. The other variables do not have any multicollinearity present among them.

7)

An interesting aspect of the multicollinearity check was in the third degree polynomial transformed `carat` predictor. When an orthogonal polynomial is used instead of a raw polynmoial, there is no multicollinearity among the degrees of the carat polynomial.

Part 3 - AIC & Predictions

1)

Variable Selection using stepwise AIC:

```
fullmodel <- lm(log(price) ~ poly(carat, 3) + depth + color + clarity,
                data = diamonds_sample)
nullmodel <- lm(log(price) ~ 1, data = diamonds_sample)

model5 <- step(nullmodel, scope = formula(fullmodel), direction = "both", trace = 0)
summary(model5)
```

```
##
## Call:
## lm(formula = log(price) ~ poly(carat, 3) + clarity + color, data = diamonds_sample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.53206 -0.08800  0.00154  0.09453  0.47866
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)       7.19366    0.04683 153.623  < 2e-16 ***
## poly(carat, 3)1 33.73760    0.15378 219.393  < 2e-16 ***
## poly(carat, 3)2 -9.39886    0.13869 -67.767  < 2e-16 ***
## poly(carat, 3)3  2.40239    0.13763  17.455  < 2e-16 ***
## clarityIF         1.14020    0.05172  22.046  < 2e-16 ***
## claritySI1        0.61963    0.04618  13.417  < 2e-16 ***
## claritySI2        0.45875    0.04626   9.918  < 2e-16 ***
## clarityVS1        0.83646    0.04654  17.972  < 2e-16 ***
## clarityVS2        0.77481    0.04626  16.748  < 2e-16 ***
## clarityVVS1       1.03436    0.04838  21.381  < 2e-16 ***
## clarityVVS2       0.97053    0.04744  20.458  < 2e-16 ***
## colorE           -0.05265    0.01571  -3.353 0.000831 ***
## colorF           -0.09118    0.01587  -5.745 1.22e-08 ***
## colorG           -0.13971    0.01535  -9.100  < 2e-16 ***
## colorH           -0.26071    0.01623 -16.066  < 2e-16 ***
## colorI           -0.35912    0.01849 -19.419  < 2e-16 ***
## colorJ           -0.51316    0.02295 -22.359  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1345 on 983 degrees of freedom
## Multiple R-squared:  0.9833, Adjusted R-squared:  0.983
## F-statistic:  3609 on 16 and 983 DF,  p-value: < 2.2e-16
```

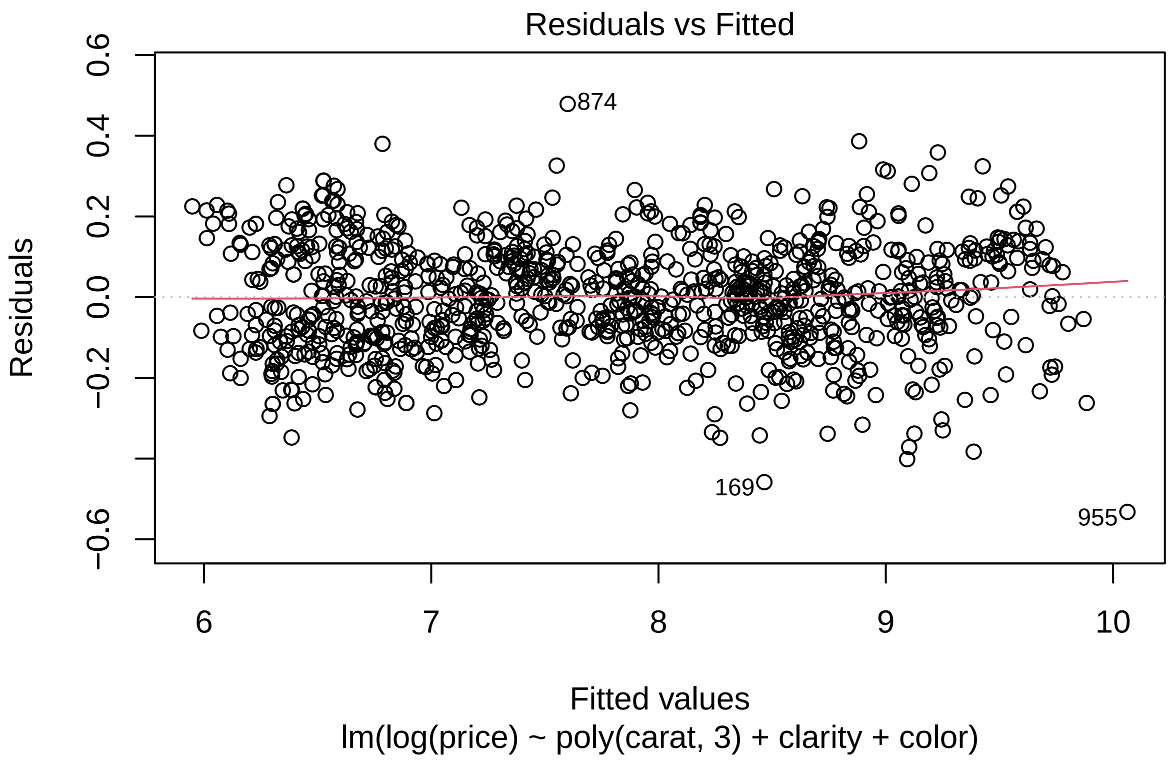
After running stepwise AIC on the transformed model, AIC chose a model with the third degree polynomial transformed `carat` predictor, `clarity`, and `color`, eliminating `depth` from `model4` that was previously created in Part 2. Adjusted R^2 remains at 0.983, but since the individual effect of `depth` was not statistically significant, that predictor is now gone. And, all individual effects are statistically significant, even at the Bonferroni-adjusted significance level ($\alpha_{Bonferroni} = \frac{0.05}{16} = 0.003$).

```r
faraway::vif(model5)
```

```
## poly(carat, 3)1 poly(carat, 3)2 poly(carat, 3)3       clarityIF      claritySI1
##        1.307796        1.063823        1.047594        4.443907       21.016031
##      claritySI2      clarityVS1      clarityVS2     clarityVVS1     clarityVVS2
##       16.928966       15.856476       20.044455        8.869377       11.990386
##          colorE          colorF          colorG          colorH          colorI
##        1.933723        1.983665        2.185148        1.947275        1.762432
##          colorJ
##        1.383695
```

Multicollinearity remains unchanged from `model4`, and once again, it can be seen that some levels of the categorical variable `clarity` have VIF values higher than 10, but VIF measures relationships among variables, not between categorical variable levels, so we'll keep them unchanged.

```r
plot(model5, which = 1)
```



The model assumptions still hold, `model5` is linear and maintains homoscedasticity.

2)

```r
# prediction data (observation 17) with all variables model5 used as predictors
predict_data <- diamonds_sample[17, c(1, 3, 4)]
```

```r
# mean predicted value confidence interval
predict(model5, , newdata = predict_data, interval = "confidence", level = 0.95)
```

```
##         fit      lwr      upr
## 17 6.627857 6.596662 6.659052
```

```r
# prediction interval
predict(model5, newdata = predict_data, interval = "prediction", level = 0.95)
```

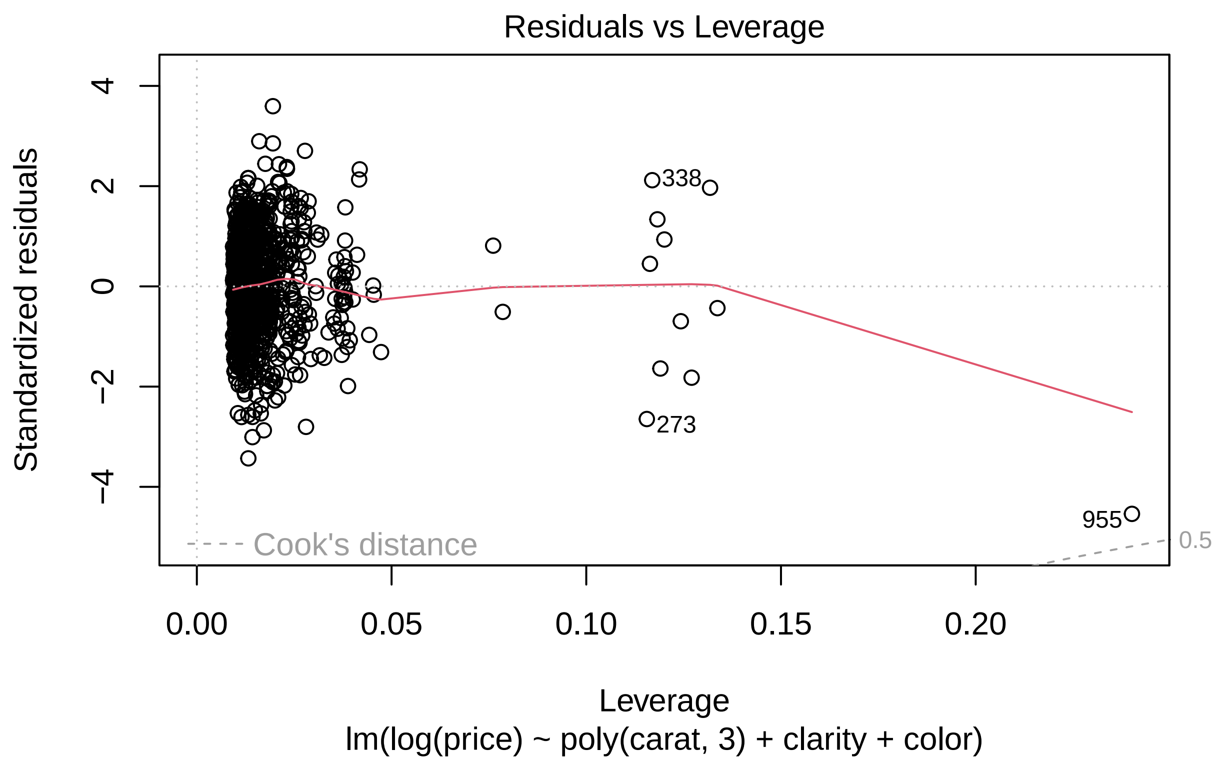
```
##         fit      lwr      upr
## 17 6.627857 6.362141 6.893573
```

Here are the confidence and prediction intervals for a mean predicted value and predicted value using the 17th observation in the dataset. However, these values are logarithmically transformed since the linear regression model used to make these predictions transformed diamond price logarithmically. So, to get the real diamond price predictions and intervals, we can apply an exponential transformation.

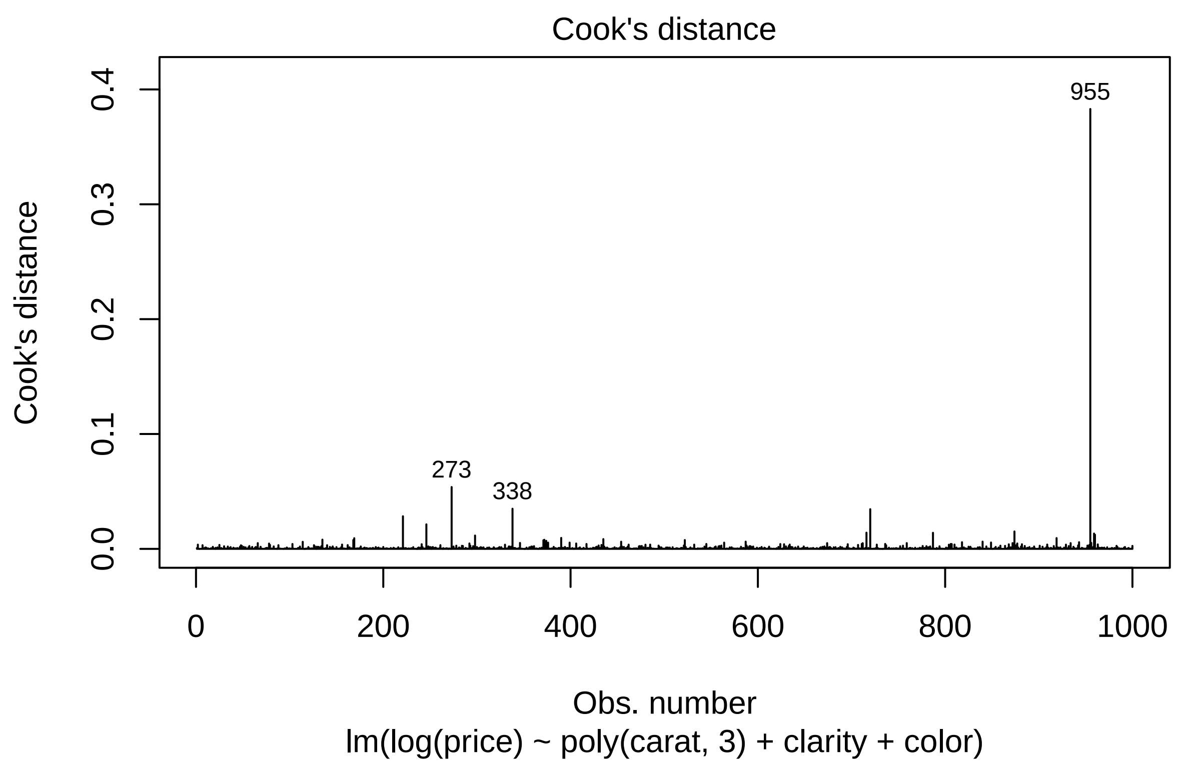
The predicted values would be 755.86 and the mean predicted value's confidence interval would be 732.65 to 779.81. The predicted value's prediction interval would be 579.49 to 985.92.

Leverage and influential points:

```r
plot(model5, which = 5)
```



```r
plot(model5, which = 4)
```



Looking at the residuals vs leverage graph, it can be seen that 3 points have high leverage. Observations 955, 273, and 338. Cook's distance confirms this, showing observation 955 with a Cook's distance of about 0.4, observation 273 with a Cook's distance of approximately 0.08, and observation 338 with a Cook's distance of approximately 0.06. Since these values are greater than the threshold for Cook's distance of $\frac{4}{n} = \frac{4}{1000} = 0.004$, then these observations could be removed from the model. However, in this case, we will not remove them, just make note of their influence.

3)

This report sought to fit a linear regression model for diamond price, choosing the most relevant predictors out of 4 chosen to do this, `carat`, `color`, `clarity`, and `depth`. The data was taken from the "Diamonds Prices" dataset on Kaggle, randomly sampling 1000 observations. Exploratory data analysis was ran on this subsetted dataset and the linear regression `summary` function output was interpreted. Additionally, diagnostics were run to check if model assumptions were upheld, in which a simple linear regression model was transformed for linearity and homoscedasticity with a 3rd degree polynomial on the predictor `carat`, and a logarithmic function on the response. Then, predictors were added one at a time to the transformed simple linear regression model, examining the adjusted R^2 value to determine goodness of fit. The predictors chosen were `carat`, `depth`, `color`, and `clarity`. The transformed model had a better fit than the model with the same predictors but without the transformation. Then, this model was compared to a model chosen using stepwise AIC on the previous model. The stepwise method eliminated the `depth` variable previously chosen, which kept adjusted R^2 the same but got rid of a statistically insignificant predictor. Multicollinearity was checked with variance inflation factors, appearing unconcerning, and finally leverage points and influential points were checked. Overall, the report helped to familiarize oneself with exploratory data analysis, feature engineering, and overfitting.