final project

Sumaiya Alvi

2025-03-15

Part 1 - Data Description and Descriptive Statistics

```
# upload the data
diamonds <- read.csv("diamondsdata.csv")
set.seed(03152025)
```

1.

```
library(dplyr)
# select random sample of 1000
# 2 categorical variables: cut, color
# 3 independent quantities: carat, depth, price
d_sample <- diamonds %>%
  select(carat, cut, color, depth, price) %>%
  sample_n(1000)
```

2.

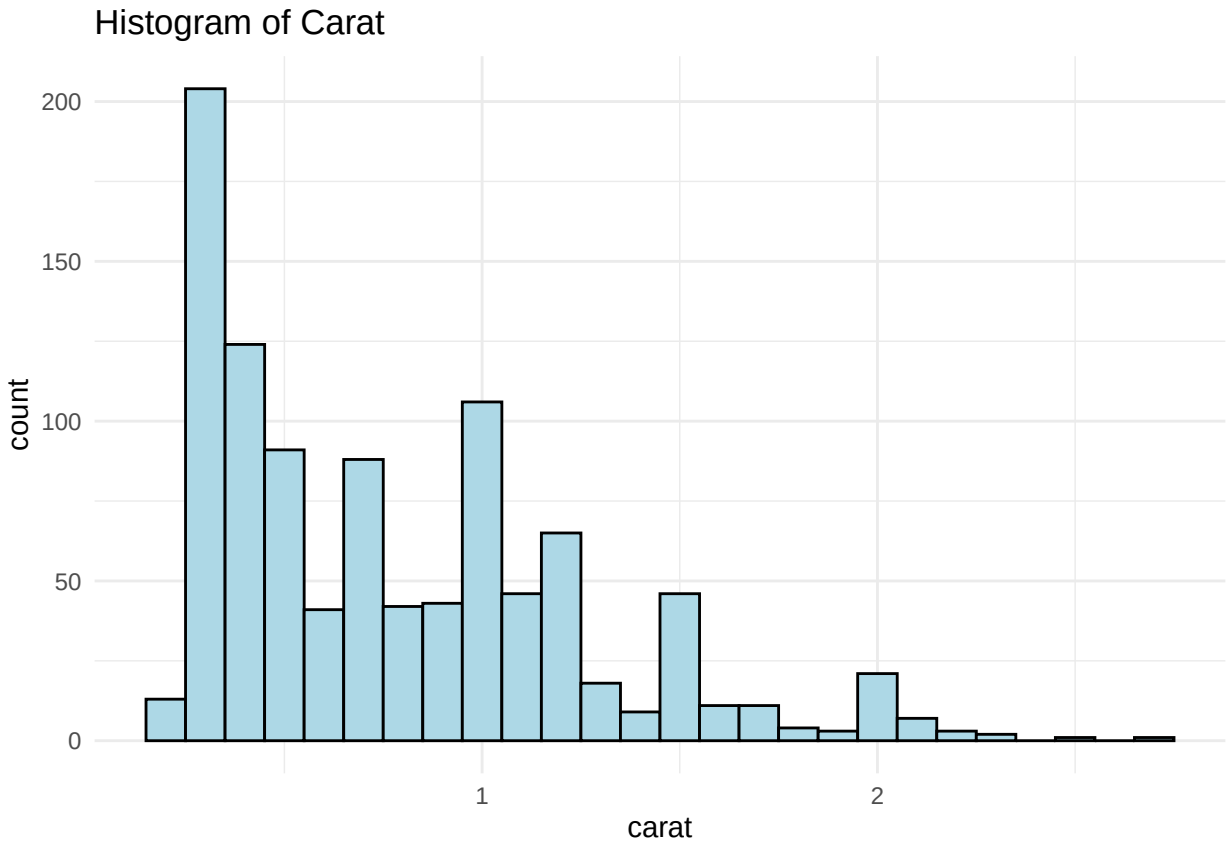
```
library(ggplot2)

# 1. summary statistics of diamonds dataset
summary(d_sample)
```

```
##      carat             cut                color               depth      
##  Min.   :0.210   Length:1000        Length:1000        Min.   :56.20  
##  1st Qu.:0.400   Class :character   Class :character   1st Qu.:61.00  
##  Median :0.700   Mode  :character   Mode  :character   Median :61.80  
##  Mean   :0.787                                         Mean   :61.71  
##  3rd Qu.:1.050                                         3rd Qu.:62.50  
##  Max.   :2.720                                         Max.   :70.60  
##      price      
##  Min.   :  351  
##  1st Qu.:  927  
##  Median : 2360  
##  Mean   : 3805  
##  3rd Qu.: 5128  
##  Max.   :18766
```

```r
str(d_sample)
```

```
## 'data.frame':    1000 obs. of  5 variables:
##  $ carat: num  0.36 0.3 0.5 1.53 1.53 0.85 0.41 0.8 1.03 0.83 ...
##  $ cut  : chr  "Premium" "Ideal" "Good" "Very Good" ...
##  $ color: chr  "F" "H" "G" "E" ...
##  $ depth: num  58.5 60.7 57.9 62.6 62 61.6 62.3 61.1 61.9 61.6 ...
##  $ price: int  869 465 1316 12386 10836 3023 1367 4388 4932 3841 ...
```

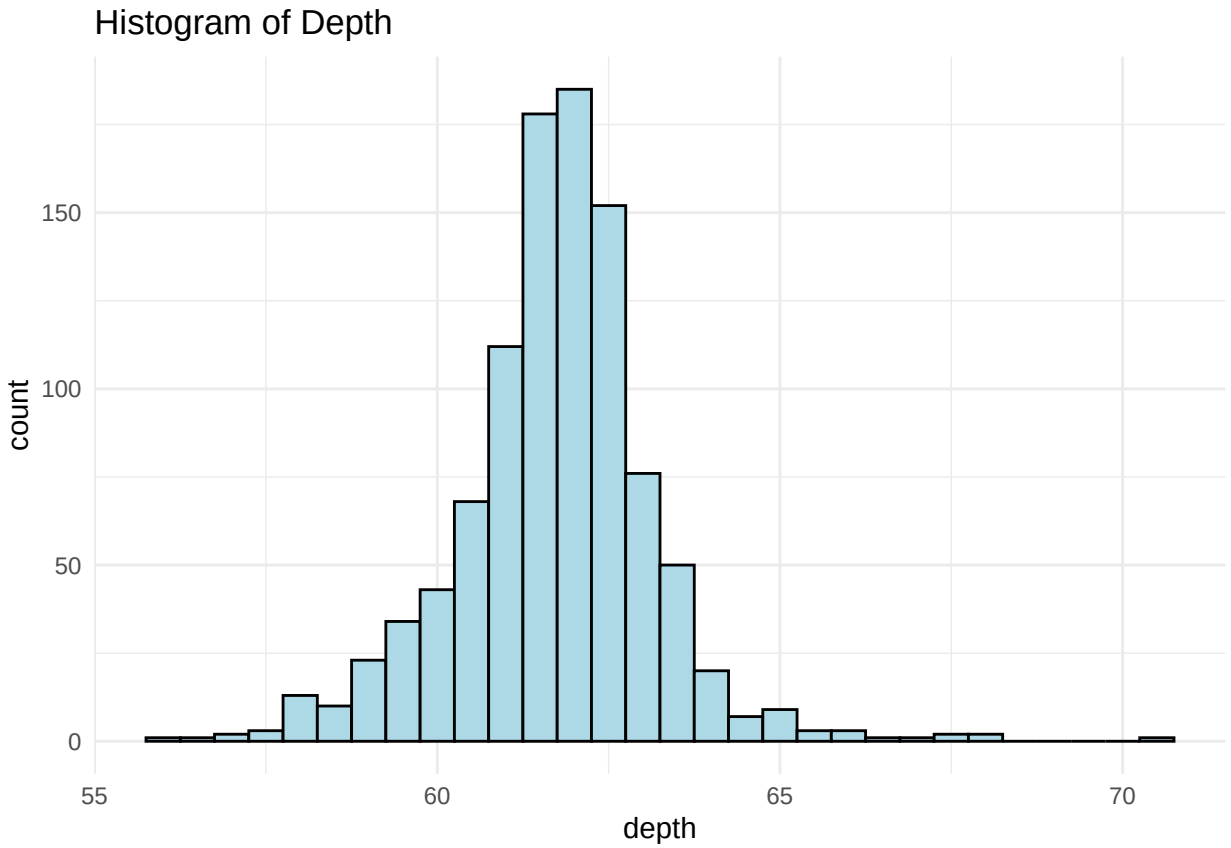
```r
# 2. histograms for continuous variables

# carat
ggplot(d_sample, aes(x = carat)) +
  geom_histogram(binwidth = 0.1, fill = "lightblue", color = "black") +
  ggtitle("Histogram of Carat") +
  theme_minimal()
```



carat: The carat histogram is right skewed, which suggests that the sample has mostly lower carat diamonds, and fewer high carat diamonds. The highest frequency is for diamonds with a carat value between 0 and 1, and the frequencies drops off as the carat count increases to 2 and beyond. This makes sense since higher carat diamonds are usually more rare, and more expensive.

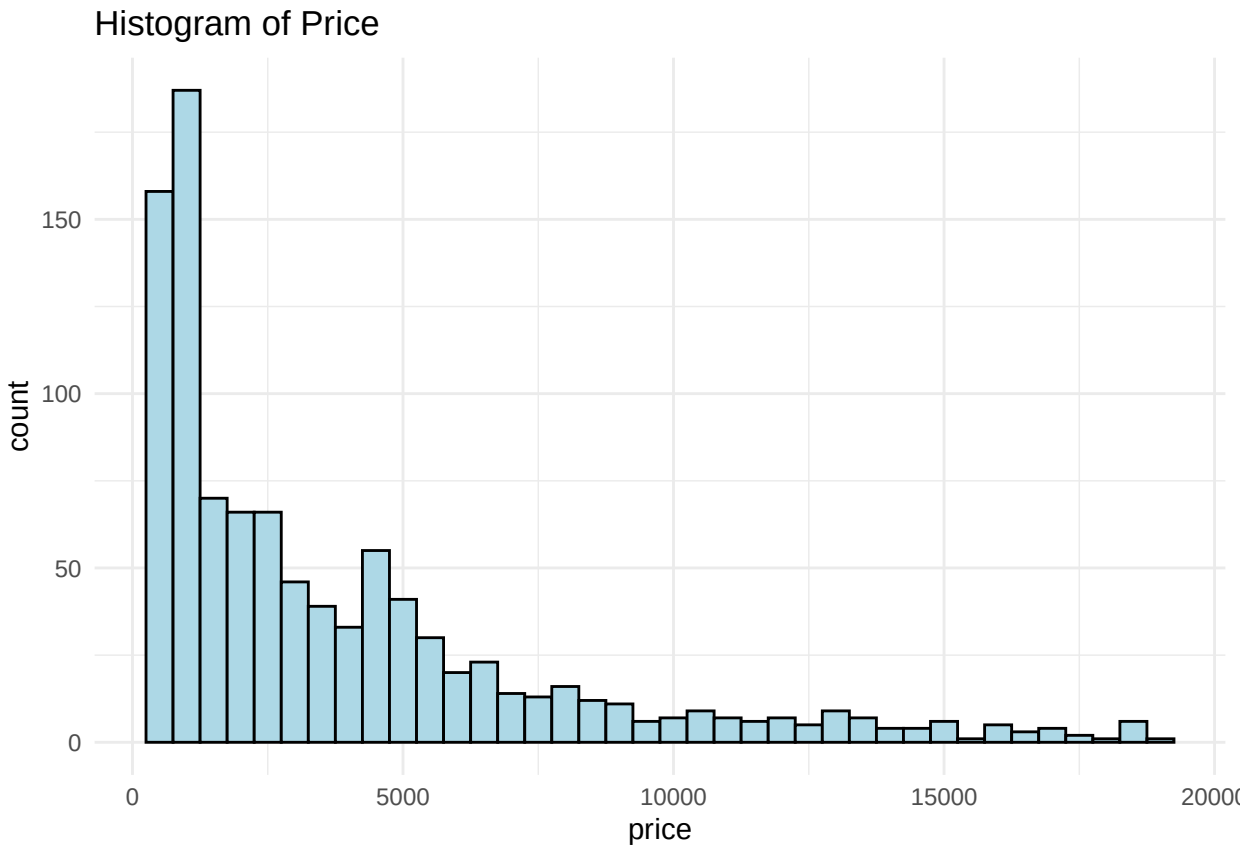
```r
# depth
ggplot(d_sample, aes(x = depth)) +
  geom_histogram(binwidth = 0.5, fill = "lightblue", color = "black") +
```

```
  ggtitle("Histogram of Depth") +
  theme_minimal()
```



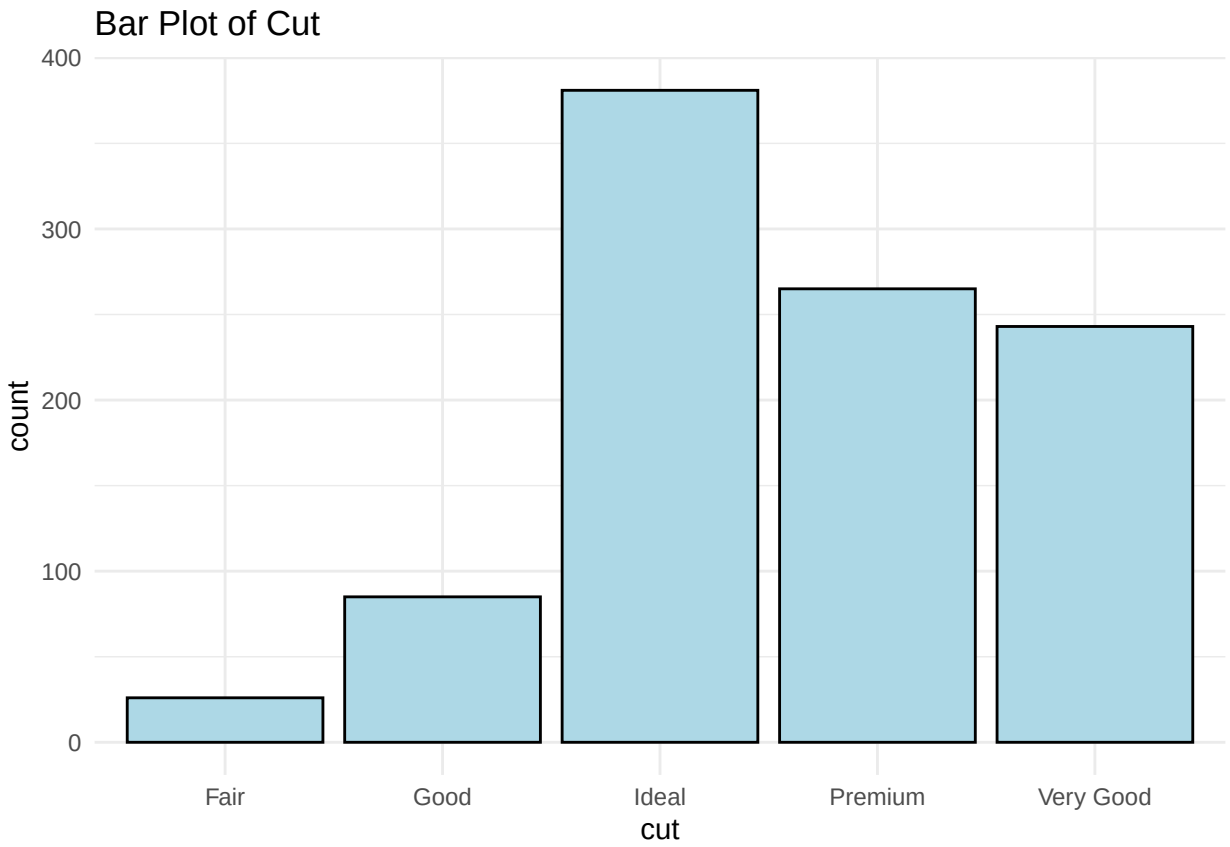
depth: The depth histogram appears to follow a normal distribution, which is centered around ~62. This means that overall, most of the diamonds are cut to a standard depth of around 61-62, and there are few outliers.

```
# price
ggplot(d_sample, aes(x = price)) +
  geom_histogram(binwidth = 500, fill = "lightblue", color = "black") +
  ggtitle("Histogram of Price") +
  theme_minimal()
```



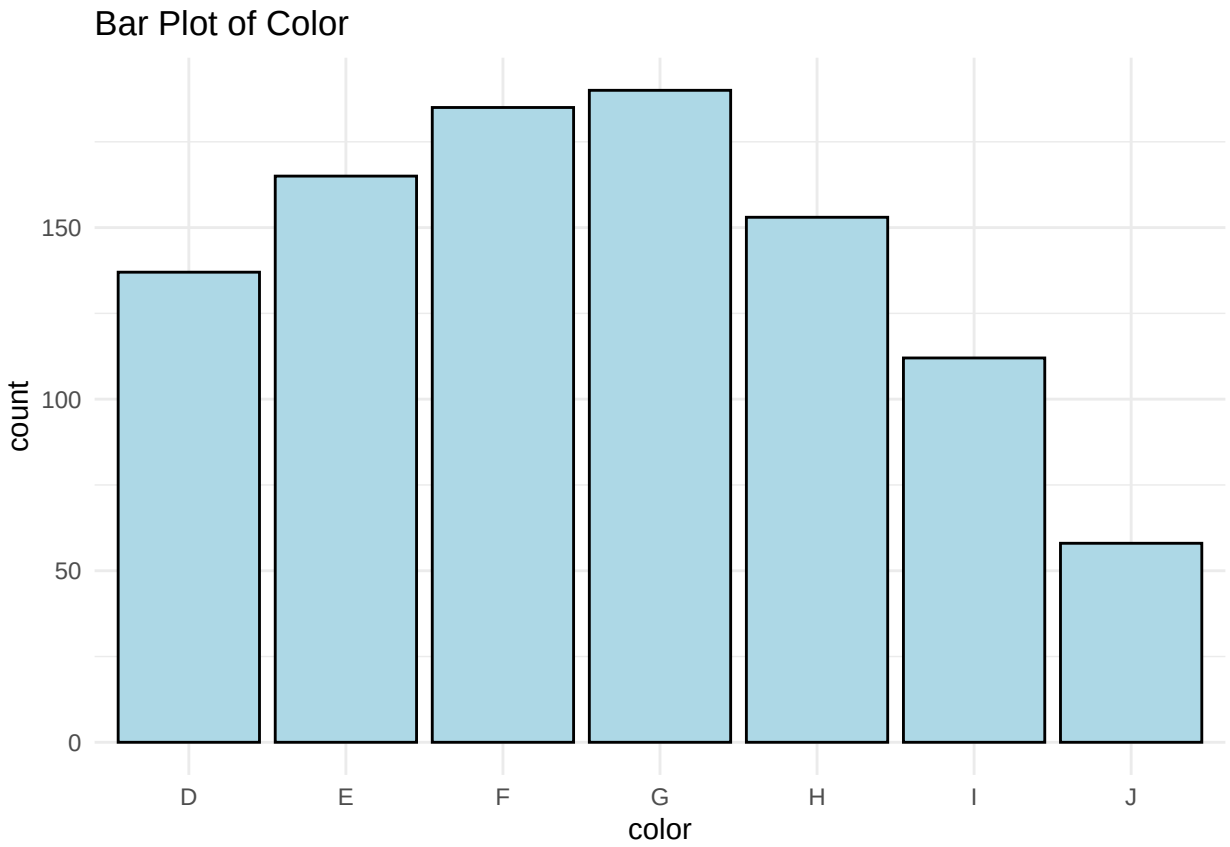
price: The price histogram is significantly right skewed, which suggests that the sample has diamonds which are mostly on the lower price end, with the peak around ~$1000. The frequencies steadily decrease as the price increases. Again, this makes sense since more expensive diamonds are more rare, and we know from the carat histogram that most of the diamonds are of a lower carat.

```
# 3. bar plots for categorical variables (cut, color)

# cut
ggplot(d_sample, aes(x = cut)) +
  geom_bar(fill = "lightblue", color = "black") +
  ggtitle("Bar Plot of Cut") +
  theme_minimal()
```



cut: The cut bar plot shows that most of the cuts are described as "Ideal". Then, "Premium" and "Very Good" cuts are next most common, with "Good" and "Fair" cuts being significantly lower. This means that most of the diamonds are of a good enough quality to meet the "Ideal" standard, if not better.

```
#color
ggplot(d_sample, aes(x = color)) +
  geom_bar(fill = "lightblue", color = "black") +
  ggtitle("Bar Plot of Color") +
  theme_minimal()
```



color: The color bar plot shows that the color distributions are relatively even, except for the J color which is far less frequent than the other colors. This could mean that it is a rarer color, or that there is less demand for it than more popular colors, like F or G.

3.

3 quantitative variables: price, carat, depth 2 categorical variables: cut, color

```
# correlation between quantitative variables
correlation_matrix <- cor(d_sample %>% select(price, carat, depth))
correlation_matrix
```

```
##              price      carat       depth
## price 1.000000000 0.91442781 0.003869853
## carat 0.914427814 1.00000000 0.025771416
## depth 0.003869853 0.02577142 1.000000000
```

price v. carat: The correlation between price and carat is ~0.92, which suggests that as the carat of the diamond increases, the price of the diamond also increases significantly.

price v. depth: The correlation between price and depth is ~0.02, which suggests that there is close to no linear relationship between the two, and price is relatively unaffected by depth.

carat v. depth: The correlation between carat and depth is ~0.03, which also suggests that there is close to no relationship between diamond weight and depth.

```r
# correlation between price and cut
anova_cut <- aov(price ~ cut, data = d_sample)
summary(anova_cut)
```

```
##              Df    Sum Sq  Mean Sq F value Pr(>F)
## cut           4 1.592e+08 39804553   2.645 0.0323 *
## Residuals   995 1.498e+10 15050610
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value for the anova test comparing price and cut quality is 0.0323. At a significance level of $\alpha = 0.05$, this suggests that there is a statistically significant difference in price across different levels of cut quality.

```r
# correlation between price and color
anova_color <- aov(price ~ color, data = d_sample)
summary(anova_color)
```

```
##              Df    Sum Sq   Mean Sq F value   Pr(>F)
## color         6 7.359e+08 122653189   8.459 5.66e-09 ***
## Residuals   993 1.440e+10  14500157
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value for the anova test comparing price and color is 5.66e-09. At a significance level of $\alpha = 0.05$, this suggests that there is a strong statistically significant difference in price for different colors.

4.

```r
model1 <- lm(price ~ carat + depth + cut + color, data = d_sample)
summary(model1)
```

```
##
## Call:
## lm(formula = price ~ carat + depth + cut + color, data = d_sample)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9344.3  -741.6   -86.3   612.8 12020.6
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -6421.51    2338.83  -2.746  0.00615 **
## carat         8089.60     104.19  77.642  < 2e-16 ***
## depth           37.13      35.59   1.043  0.29709
## cutGood       1128.50     342.17   3.298  0.00101 **
## cutIdeal      2117.32     318.09   6.656 4.65e-11 ***
## cutPremium    1763.06     329.35   5.353 1.07e-07 ***
## cutVery Good  1833.79     319.70   5.736 1.29e-08 ***
## colorE          59.36     165.44   0.359  0.71981
```

```
## colorF         201.82     161.92   1.246  0.21291
## colorG         175.15     162.00   1.081  0.27989
## colorH        -716.69     170.36  -4.207 2.82e-05 ***
## colorI        -958.22     186.20  -5.146 3.21e-07 ***
## colorJ       -1899.96     230.28  -8.251 5.01e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1425 on 987 degrees of freedom
## Multiple R-squared:  0.8675, Adjusted R-squared:  0.8659
## F-statistic: 538.7 on 12 and 987 DF,  p-value: < 2.2e-16
```

5.

The data is relatively similar to what I would expect to see out of a model analyzing different factors relationship to diamond price. The carat coefficient is 8139.74, which means that for every increase in 1 carat, the diamond price increases by about $8139.74. This makes sense, since higher carat diamonds are very expensive. We saw earlier that price and depth had close to no linear relationship. Here we see that the depth p-value is ~0.5, which further proves that depth is not statistically significant in predicting diamond price.

Some of the colors (H, I, J) are statistically significant and have lower p-values than the rest of the colors. They also have negative coefficients, which suggests that these colors tend to result in a lower price. For colors E, F, and G, the higher p-values suggest that they don't have a strong difference in price from the baseline (color D).

The model has a good fit with $R^2 = 0.8669$, which means that 86.69% of the variability in diamond prices can be explained by the regression variables (cut, carat, depth, etc). The adjusted R^2 value is 0.8652, which is very close to 0.8669, suggesting that adding the variables doesn't inflate the R^2 value unnecessarilya.

The F-statistic is 535.5 with a very small p-value (<2.2e-16) which indicates that the model is highly significant.

Part 2 Simple Linear Regression (Part 1 continuation)

1.

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

```
##
## Call:
## lm(formula = price ~ carat, data = d_sample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11699.1   -829.7     -2.1    552.2  12312.6
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2206.25      97.87  -22.54   <2e-16 ***
```

```
## carat        7637.99     107.02   71.37   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1576 on 998 degrees of freedom
## Multiple R-squared:  0.8362, Adjusted R-squared:  0.836
## F-statistic:  5094 on 1 and 998 DF,  p-value: < 2.2e-16
```

2.

The given regression equation from model2 is $\hat{price} = 7725.96carat - 2194.63$, which means that the price increases by $7725.96 per carat, and that a 0 carat diamond is worth -$2194.63 (theoretical).

The null hypothesis H_0 of our test is that there is no impact on diamond price for different carat values. We can reject the null hypothesis since our p value is very small, <2e-16. Thus, diamond carat is a significant predictor of price.

We know that the model has a good fit because R^2 is 0.8465, and it is very close in value to the adjusted R^2, which prioritizes model fit to provide a better measure while avoiding overfitting. Also, since we only have one predictor so far, the two R^2 values would be close.

The Residual Standard Error value tells us that the typical error in diamond price prediction after using this model is $1516. This is unsurprising since our model doesn't include other important price predictors, such as color, cut, etc.

```
confint(model2)
```

```
##                 2.5 %    97.5 %
## (Intercept) -2398.309 -2014.199
## carat        7427.991  7847.998
```

The confidence interval for the carat predictor tells us that the true effect of increasing carat falls between 7521.57 and 7930.34, with a 95% certainty. Furthermore, since this interval doesn't include 0 we know that the carat value is a significant predictor of price.

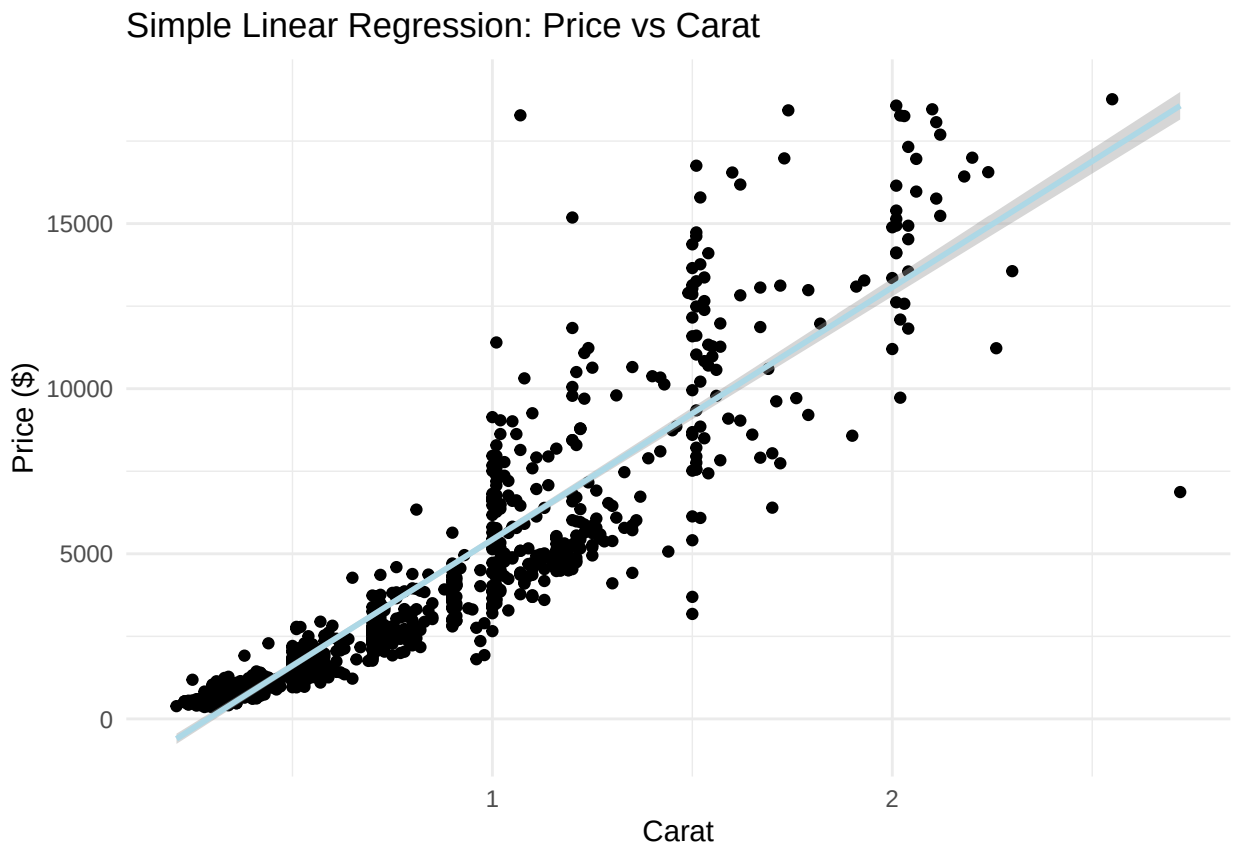
```
predict(model2, interval="confidence")[1:5,]
```

```
##          fit        lwr       upr
## 1  543.42446  410.72363  676.1253
## 2   85.14479  -56.37655  226.6661
## 3 1612.74368 1497.85013 1727.6372
## 4 9479.87800 9295.73275 9664.0232
## 5 9479.87800 9295.73275 9664.0232
```

In the shown confidence table values above, the "fit" column shows the predicted price for a given carat value, follow by the lower and upper bounds of the 95% confidence interval. We can see that the intervals are relatively small, which means that our model provides pretty precise predictions. For less narrow intervals (ie. 3rd row), we can assume that carat value is lower, which explains the higher variability.

```
ggplot(d_sample, aes(x = carat, y = price)) +
  geom_point() +
  geom_smooth(method="lm", color="lightblue") +
  labs(title = "Simple Linear Regression: Price vs Carat",
       x = "Carat", y = "Price ($)") +
  theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

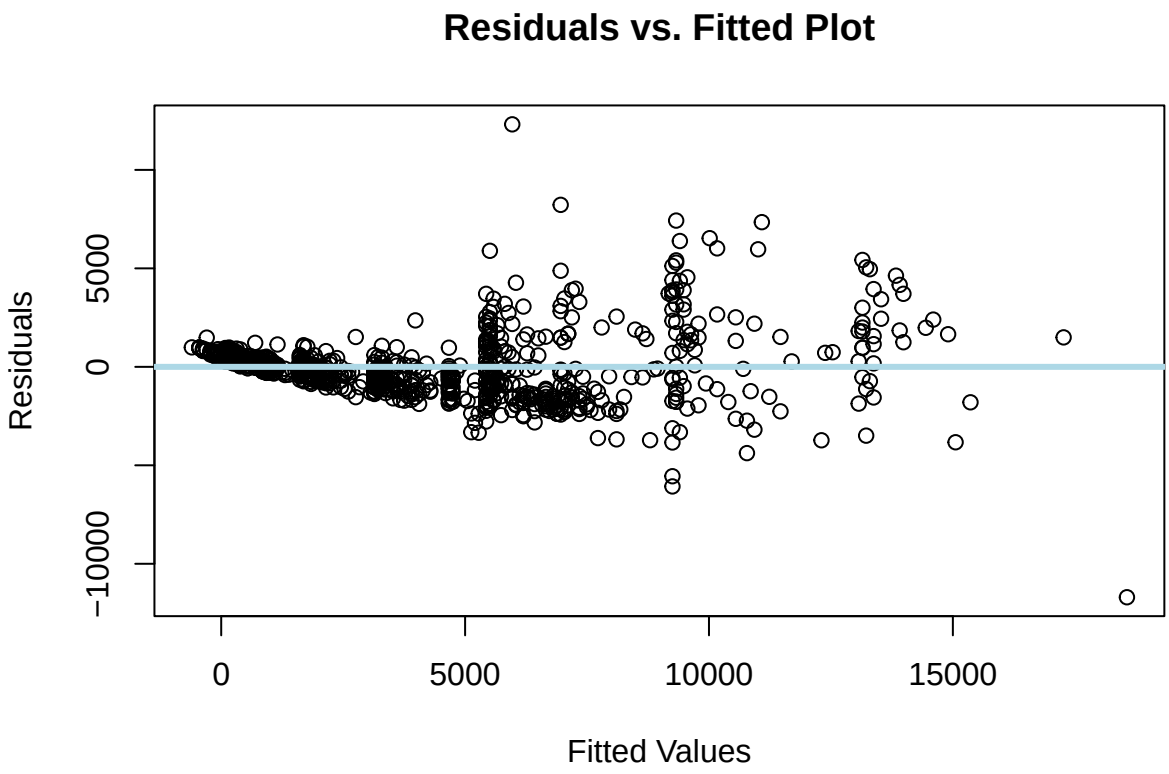


The scatter plot of carat vs. price shows that the data points generally follow the regression line, but there is some variance around the line. This suggests that carat is a strong predictor, but diamond price is still affected by other prices (this follows the conclusion we have drawn previously).

3.

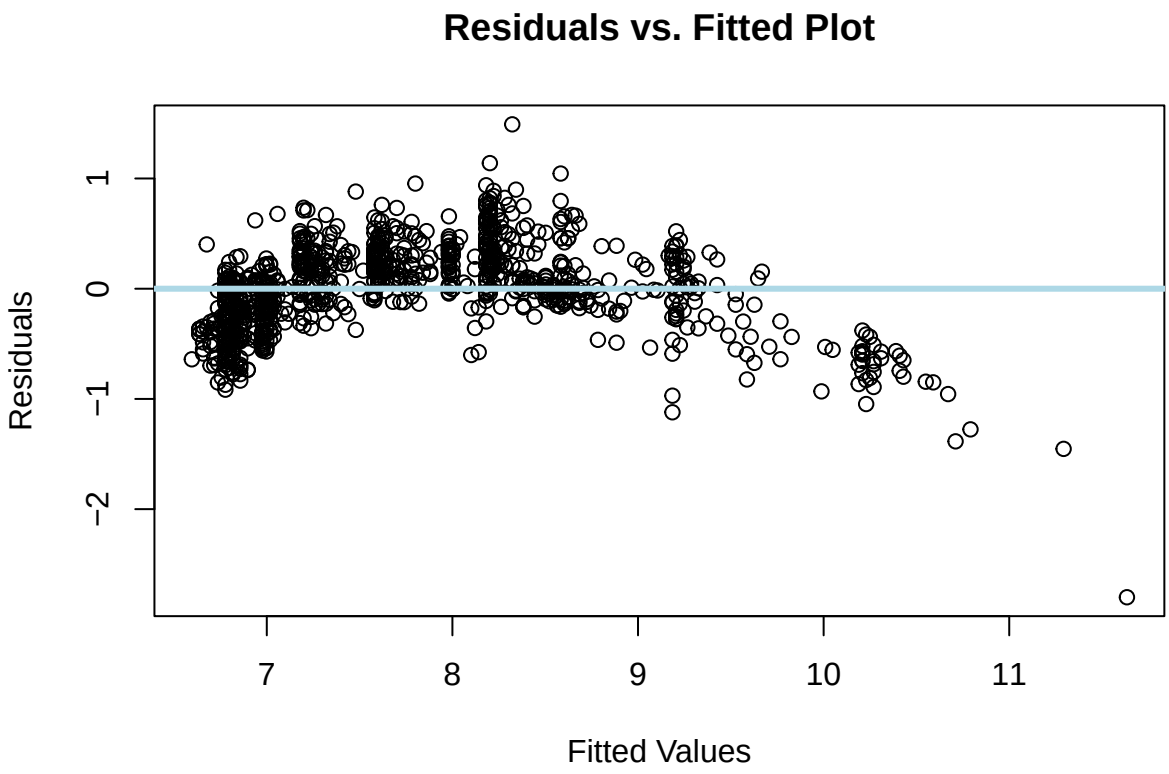
First, we'll plot residuals vs. fitted values to check the linearity and homoscedasticity assumptions

```
plot(model2$fitted.values, residuals(model2),
     xlab = "Fitted Values",
     ylab = "Residuals",
     main = "Residuals vs. Fitted Plot")
abline(h=0, col = "lightblue", lwd=3)
```



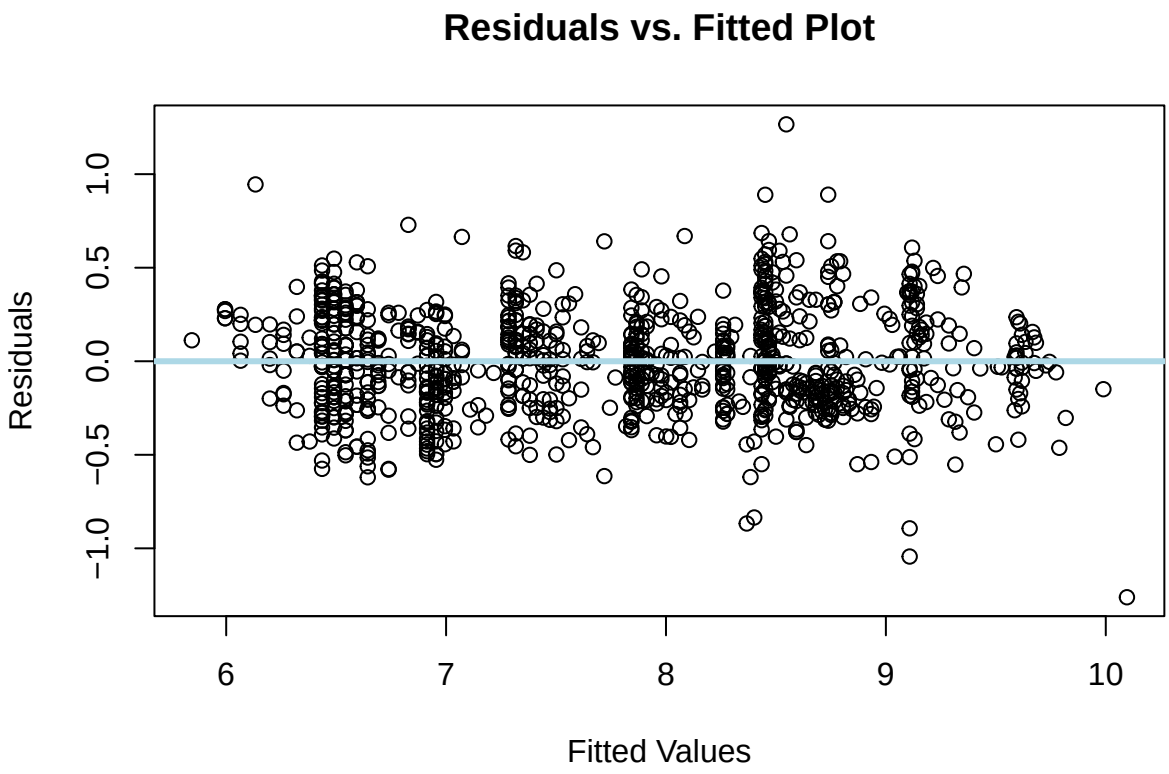
The data points seem to follow a systematic pattern (U-shape), so we can assume that the linearity assumption is not met. Also, the points seem to funnel out towards the center of the plot and then back in at the ends. Thus, we can't assume that the constant variance assumption is met. In order to fix this, we can transform the model by taking the logs of price and carat. Price first.

```
d_sample$log_price <- log(d_sample$price) #log transformation of price
model3 <- lm(log_price ~ carat, data = d_sample) #run new model
plot(model3$fitted.values, residuals(model3), #plot new model
     xlab = "Fitted Values",
     ylab = "Residuals",
     main = "Residuals vs. Fitted Plot")
abline(h=0, col = "lightblue", lwd=3)
```



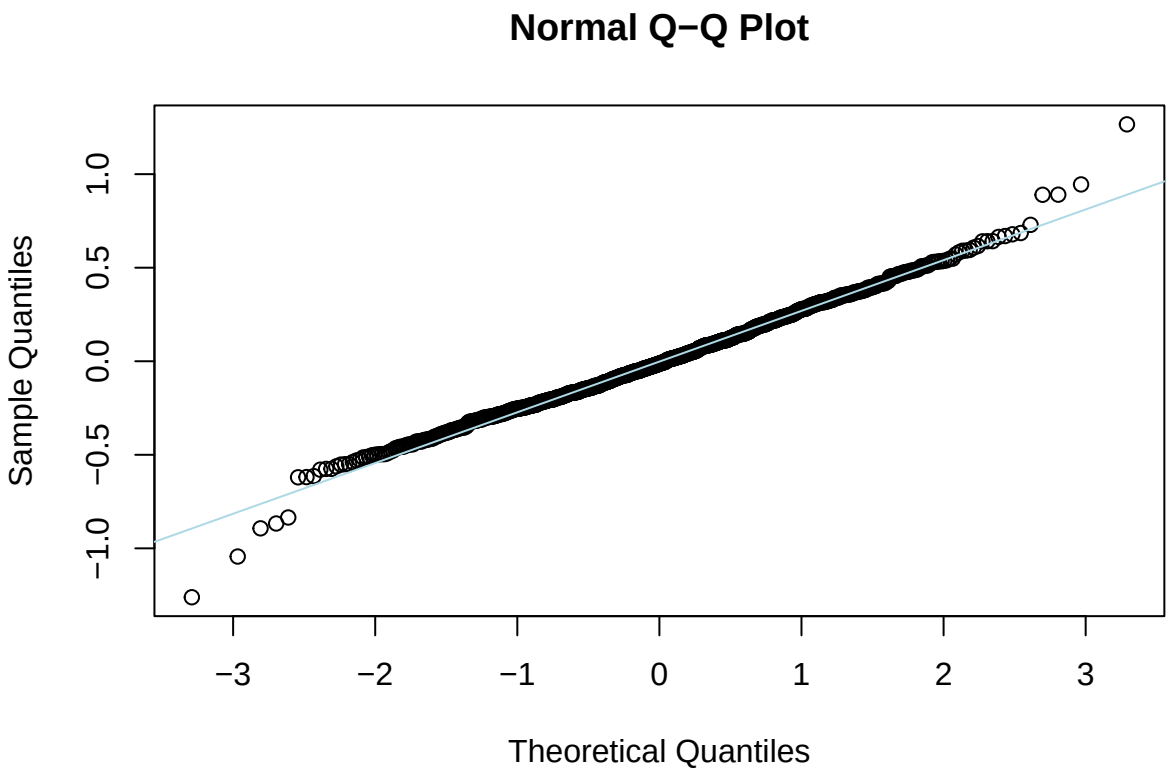
Now that we have transformed the model to account for homoscedasticity, it is even more clear that the data follows a structured pattern and therefore, the normality assumption is not met. We can apply a log transformation to carat to fix this.

```
d_sample$log_carat <- log(d_sample$carat) #log transformation of carat
model4 <- lm(log_price ~ log_carat, data = d_sample) #run new model
plot(model4$fitted.values, residuals(model4), #plot new model
     xlab = "Fitted Values",
     ylab = "Residuals",
     main = "Residuals vs. Fitted Plot")
abline(h=0, col = "lightblue", lwd=3)
```



Now the residuals v. fitted value plot has no structured pattern and doesn't funnel, so the normality and homoscedasticity assumptions are met. In order to double check that the normality assumption is met, we can analyze the qq-plot.

```
qqnorm(residuals(model4))
qqline(residuals(model4), col = "lightblue")
```



The data falls along the qq-line. Thus, we know that the normality assumption is met.

4.

```
summary(model4)
```

```
##
## Call:
## lm(formula = log_price ~ log_carat, data = d_sample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.26140 -0.18470 -0.01254  0.18097  1.26622
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.43495    0.01043   809.1   <2e-16 ***
## log_carat    1.66032    0.01456   114.1   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2704 on 998 degrees of freedom
## Multiple R-squared:  0.9288, Adjusted R-squared:  0.9287
## F-statistic: 1.301e+04 on 1 and 998 DF,  p-value: < 2.2e-16
```

Looking at the residuals category, we can see that the residuals are much more closely centered around zero than the linear model summary from before, which suggests that normality is closer to being met now.

Additionally, looking at the coefficients category reveals that the log(carat) coefficient is ~1.67, which is how much percentage the price increases by for a 1% increase in carat. It's highly statistically significant, since the p-value (<2e-16) is so low.

The residual standard error is 0.2699, which is relatively small and therefore means that the predictions are close to the actual values, indicating a good fit. Additionally, the large F-statistic (1.283e04) with a small p-value (<2.2e-16) indicates that the model is statistically significant.

One of the bigger differences are the R^2 and adjusted R^2 values, which are both 0.9278. Not only is this a significantly better fit than before, but the values are the same, which shows that there's no overfitting.

5.

Adding depth to the model resulted in a decrease of adjusted R^2 by 0.0001, which indicates that it doesn't have a significant impact on price and it can be dropped from the model.

Adding cut to the model resulted in an increase of adjusted R^2, by 0.0034. This is a small increase, but it is an increase nonetheless, which means that the model is improved by adding the cut predictor.

Adding color to the model also resulted in an increase of adjusted R^2, by 0.0124. Thus, the model is also improved by adding the color predictor.

Adding both color and cut to the model results in the greatest increase of adjusted R^2, by 0.0163. Therefore the optimal model to predict diamond price includes the carat, cut, and color predictors.

6.

The fact that the R^2 and adjusted R^2 values are so close (0.9447 and 0.9441, respectively) indicates that there is no overfitting.

In order to check that there is no multicollinearity, we can check the Variance Inflaction Factor (VIF)

```
faraway::vif(lm(log_price ~ log_carat + color + cut, data = d_sample))
```

```
##    log_carat       colorE       colorF       colorG       colorH       colorI
##     1.143243     1.854315     1.943031     1.983407     1.833825     1.673754
##       colorJ      cutGood     cutIdeal   cutPremium cutVery Good
##     1.410578     3.953025     9.920937     8.320261     7.948886
```

A VIF value > 5 suggests high collinearity, which some of the cut variables (cutIdeal, cutPremium, cutVery-Good) have. These variables represents different levels of the same category, which might be why they contribute redundancy to the model.

7.

It's interesting that there's multicollinearity with some of the cut variables, which are categorical variables. One possible solution to deal with this could be to reduce the number of similar predictors (cutIdeal, cut-Premium, cutVeryGood). It's important to remember that earlier we concluded that cut quality has a significant impact on diamond price.

Something else we concluded is that our adjusted model has a good ability to predict diamond price using the carat, cut, and color predictors, since the R^2 values are so high (~0.944). We were able to drop the depth predictor.

Part 3 (Part 2 continuation)

1.

```
best_model <- lm(log_price ~ log_carat + color + cut, data=d_sample)
summary(best_model)
```

```
##
## Call:
## lm(formula = log_price ~ log_carat + color + cut, data = d_sample)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.03192 -0.15332  0.00576  0.16320  1.15259
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.269708   0.049284 167.796  < 2e-16 ***
## log_carat     1.731741   0.013369 129.537  < 2e-16 ***
## colorE       -0.012982   0.026944  -0.482   0.6300
## colorF        0.022644   0.026365   0.859   0.3906
## colorG       -0.009848   0.026366  -0.374   0.7088
## colorH       -0.162739   0.027628  -5.890 5.28e-09 ***
## colorI       -0.281672   0.030129  -9.349  < 2e-16 ***
## colorJ       -0.386290   0.037317 -10.351  < 2e-16 ***
## cutGood       0.134222   0.052360   2.563   0.0105 *
## cutIdeal      0.330732   0.047634   6.943 6.94e-12 ***
## cutPremium    0.274045   0.048002   5.709 1.50e-08 ***
## cutVery Good  0.259902   0.048279   5.383 9.13e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2322 on 988 degrees of freedom
## Multiple R-squared:  0.948,  Adjusted R-squared:  0.9474
## F-statistic:  1636 on 11 and 988 DF,  p-value: < 2.2e-16
```

The residuals category is centered around 0, which suggests that linearity and homoscedasticity assumptions are met (we have already transformed the model to ensure so).

The coefficient for the log_carat predictor is 1.74408, with a p-value (<2e-16) < 0.05, indicating that carat has a high impact on diamond price. The same can be said for cutIdeal, cutPremium, and cutVeryGood. Thus, these predictors also have a significant impact on diamond price. Additionally, colorG, colorI, and colorJ are also statistically significant, all of which have a negative impact on log_price (since they have negative coefficients).

Earlier we analyzed the bar graph for color J and saw that its frequency was much lower than the other colors. Now we can assume that this is an unpopular color, which is why it is related to a lower price. The reference category here is color D, the most rare and desired diamond color. Therefore it makes sense that all of the other colors have a negative coefficient in comparison, since they would be less popular.

The R^2 value of 0.9447 indicates that around 94.47% of the variance in log price is explained by the model. The adjusted R^2 value of 0.9441 adjusts for the number of predictors while showing that 94.41% of the variance is still explained by the model. Since they are both high values, we know the model has good fit.

Finally, the high F statistic (1535) and small p-value (<2.2e-16) indicates that at least one of the predictors is significantly related to log_price, which we already know.

2.

```
#convert color and cut to factors
d_sample$color <- as.factor(d_sample$color)
d_sample$cut <- as.factor(d_sample$cut)

#store sample colors as levels in the assigned vectors
color_levels <- levels(d_sample$color)
cut_levels <- levels(d_sample$cut)

#assign new X predictor values
new_combo <- data.frame(
  log_carat = 0.5,
  color = factor("G", levels = color_levels),
  cut = factor("Ideal", levels = cut_levels)
) #carat = 0.5, color=G, cut=Ideal

CI <- predict(best_model, newdata = new_combo, interval = "confidence",
              level = 0.95)
PI <- predict(best_model, newdata = new_combo, interval = "prediction",
              level = 0.95)

#log price intervals
CI
```

```
##        fit      lwr      upr
## 1 9.456463 9.411433 9.501493
```

```
PI
```

```
##        fit      lwr      upr
## 1 9.456463 8.998487 9.914439
```

```
#price intervals
exp(CI)
```

```
##        fit      lwr      upr
## 1 12790.57 12227.39 13379.68
```

```
exp(PI)
```

```
##        fit      lwr      upr
## 1 12790.57 8090.832 20220.24
```

The 95% confidence interval for the log price with a 0.5 carat, G color, and ideal cut diamond is [9.411433, 9.501493]. Since this is the log price interval, we can take $\exp(\hat{price})$ and see that the price confidence

interval is [12227.39, 13379.68]. With 95% confidence, the true mean price for such a diamond falls within that interval.

The 95% prediction interval for the log price with the same characteristics is [8.998487, 9.914439], and the exp(PI) price interval is [8090.832, 20220.24]. It's much broader because it accounts for error from the fitted model, as well as error associated with future observations.

3.

In this project we analyzed a data set from Kaggle which included price for diamonds and several characteristics associated with the diamond such as carat, cut, color, depth, clarity, etc. We then narrowed down the data set by choosing a random sample of 1000 diamonds with only carat, cut, color, and depth as the predictors for price.

In part 1, we created visualizations for each predictor, and saw that most of the diamonds had a lower carat and a generally lower price (for diamonds), while having higher quality cuts and a distribution centered around a standard depth of about 61%. The distribution of colors was relatively even, but it was clear that one of the colors had a much lower frequency than others (J). After looking at diamond color scales, it appears that the least desired color is J, and the most desired is D.

We also found that price and carat are positively related, and price tends to increase as carat value does. The same conclusion was met for cut quality and color. As for depth, there was not a significant relationship to price.

After running a linear model on the diamonds sample, we saw that some of the colors have negative coefficients, and thus tend to lead to a lower price.

In part 2, we further explored the relationship between price and carat in a simple linear regression model. Our linear model test concluded that diamond carat value is a significant predictor of price. It was a pretty reliable conclusion because our R^2 values were significantly high, and close in value to each other (although not exactly the same), indicating the model had a good fit.

Analyzing the residuals vs. fitted plot for the simple linear regression model revealed that the data did not follow the normality or homoscedasticity assumptions. We then transformed the model by taking the log of both price and carat.

To round off part 2, we analyzed the difference in the adjusted R^2 values after adding the cut, depth, and color predictors to the model (individually first). We found that although cut and color improved the goodness of model fit, depth did not have a significant impact and was therefore dropped. Our final best model then included log price, log carat, cut, color, and the intercept. We ensured that there is no multicollinearity or overfitting in the model.

Finally, in part 3 we ran a linear model on the best model and were able to come to conclusions that lined up with what we previously inferred. Carat has a high impact on diamond price, as well as higher quality cuts. What was interesting is that the colors (compared to color D, the most sought after) had negative coefficients, and thus lead to a decrease in price compared to color D. Our final R^2 values were ~0.944 for both, which proves that the model has a good fit.

We then chose a combination of X's (0.5 carat, G color, "Ideal" cut), and produced a 95% diamond price confidence interval of [`$12,227.39`, `$13,379.68`] and a prediction interval of [`$8,090.832`, `$20220.24`]. As expected, the prediction interval is broader than the confidence interval for reasons previously stated.

In the future, there are some modifications we could make to improve the model further. For example, we could remove non-significant variables like color E and cutGood, since they don't significantly contribute to the price prediction model.