Final Project

Sashwat Saha, Kevin Brunia

2025-06-05

Part 1

1.1

```r
set.seed(6425)
df <- read_csv("Diamonds Prices2022.csv")
```

```
## New names:
## Rows: 53943 Columns: 11
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (3): cut, color, clarity dbl (8): ...1, carat, depth, table, price, x, y, z
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

```r
df <- df[sample(nrow(df), 1000), ]
```

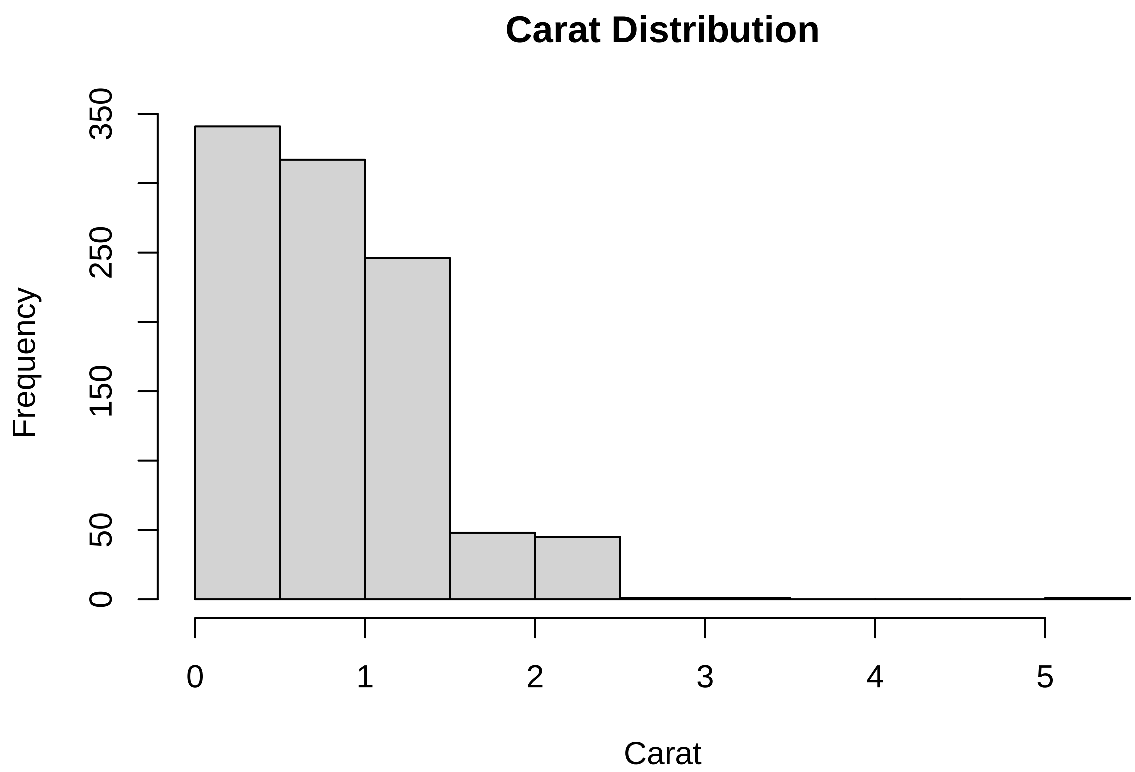
1.2

```r
summary(df)
```

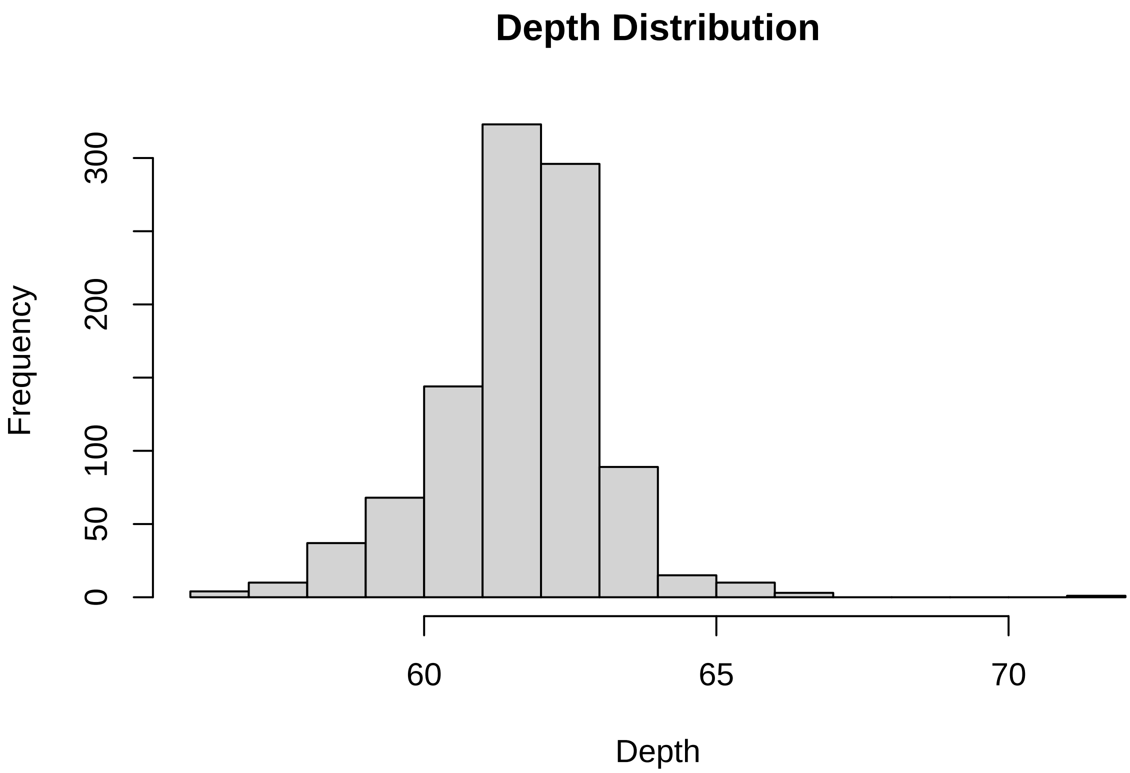
```
##       ...1           carat            cut                color          
##  Min.   :   39   Min.   :0.230   Length:1000        Length:1000       
##  1st Qu.:13602   1st Qu.:0.400   Class :character   Class :character  
##  Median :26536   Median :0.710   Mode  :character   Mode  :character  
##  Mean   :26716   Mean   :0.824                                        
##  3rd Qu.:39425   3rd Qu.:1.060                                        
##  Max.   :53867   Max.   :5.010                                        
##    clarity              depth           table           price        
##  Length:1000        Min.   :56.70   Min.   :52.00   Min.   :  384.0  
##  Class :character   1st Qu.:61.00   1st Qu.:56.00   1st Qu.:  980.8  
##  Mode  :character   Median :61.80   Median :57.00   Median : 2545.0  
##                     Mean   :61.68   Mean   :57.48   Mean   : 4177.2  
##                     3rd Qu.:62.50   3rd Qu.:59.00   3rd Qu.: 5660.5  
##                     Max.   :71.30   Max.   :68.00   Max.   :18710.0  
```

```
##        x                y                z
##  Min.   : 3.910   Min.   : 3.950   Min.   :2.360
##  1st Qu.: 4.740   1st Qu.: 4.740   1st Qu.:2.930
##  Median : 5.750   Median : 5.740   Median :3.545
##  Mean   : 5.793   Mean   : 5.797   Mean   :3.573
##  3rd Qu.: 6.610   3rd Qu.: 6.582   3rd Qu.:4.050
##  Max.   :10.740   Max.   :10.540   Max.   :6.980
```

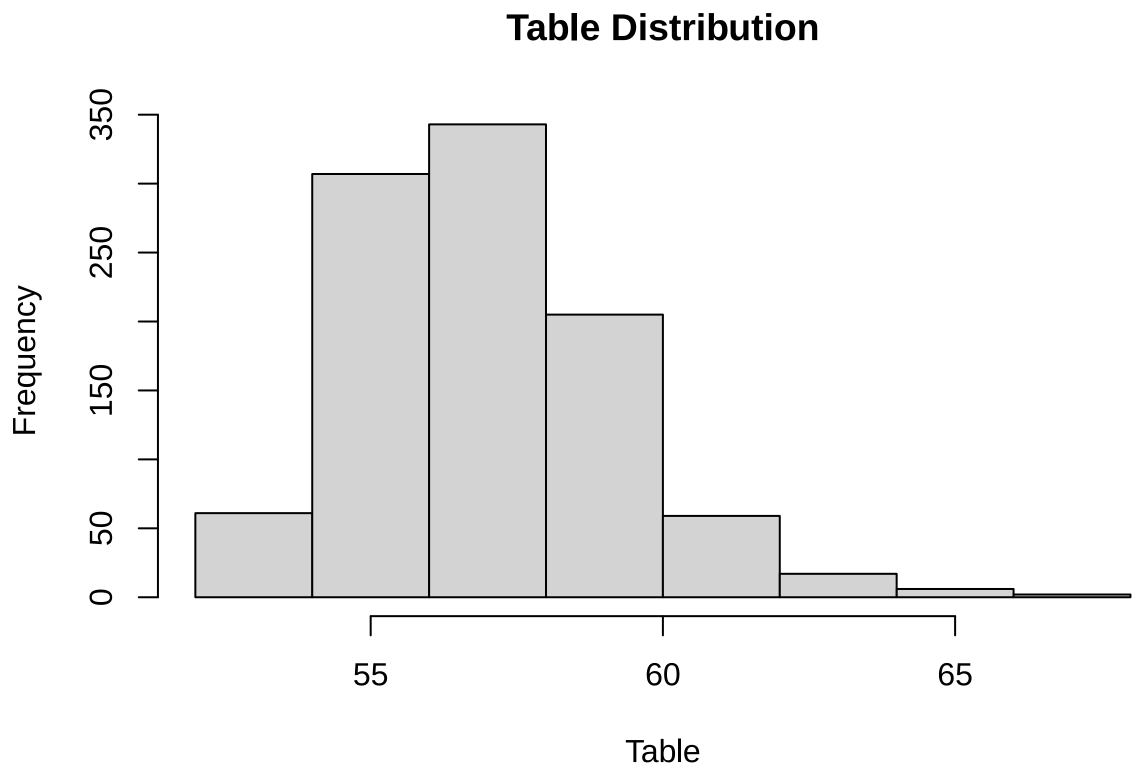
```r
hist(df$carat, main="Carat Distribution", xlab="Carat")
```



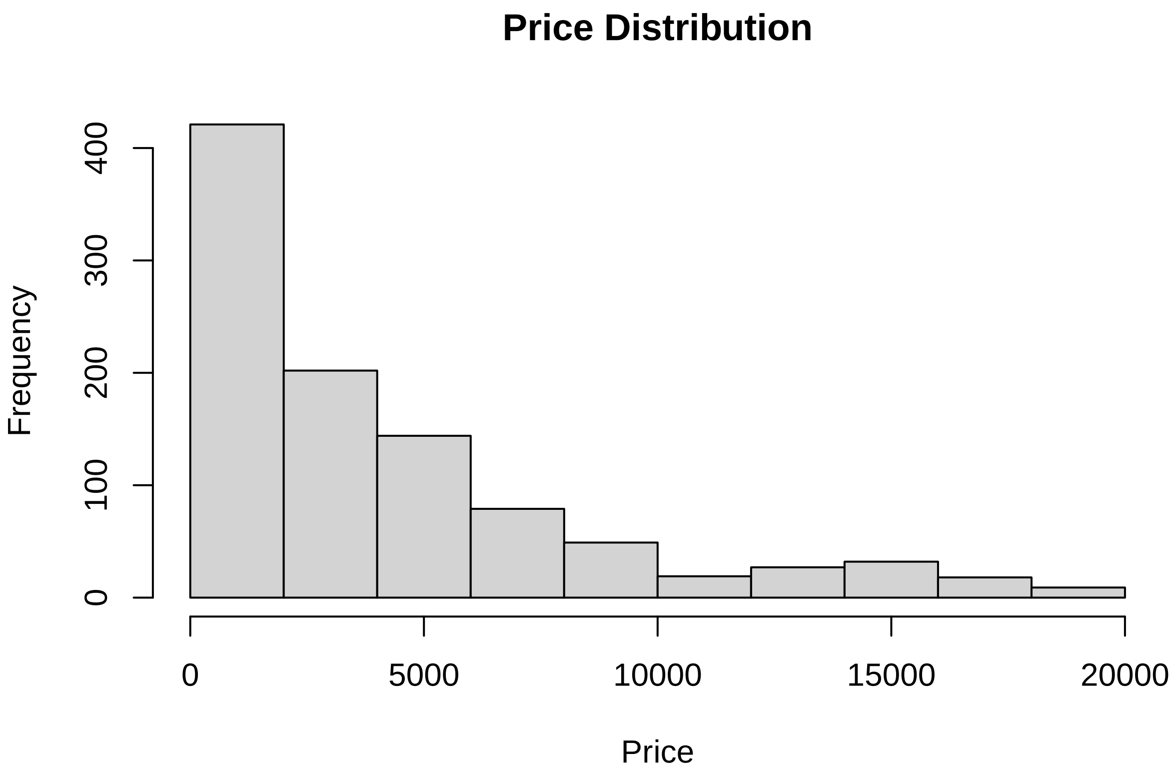

```r
hist(df$depth, main="Depth Distribution", xlab="Depth")
```



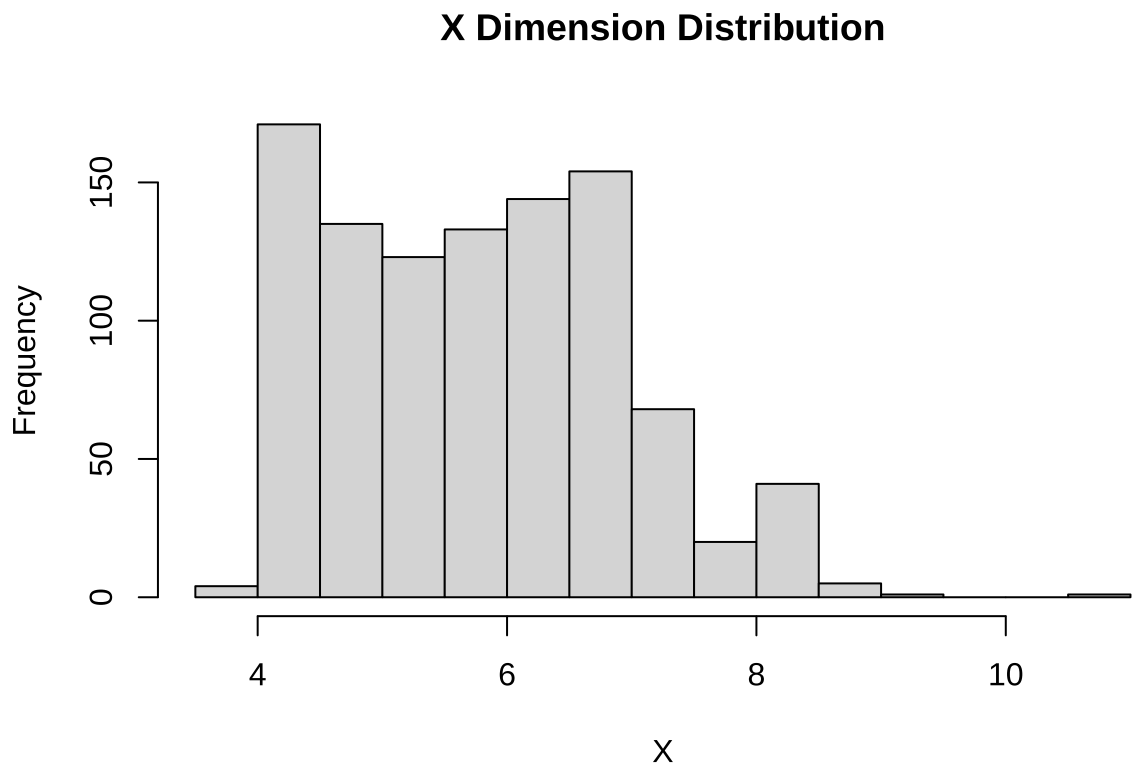

```r
hist(df$table, main="Table Distribution", xlab="Table")
```



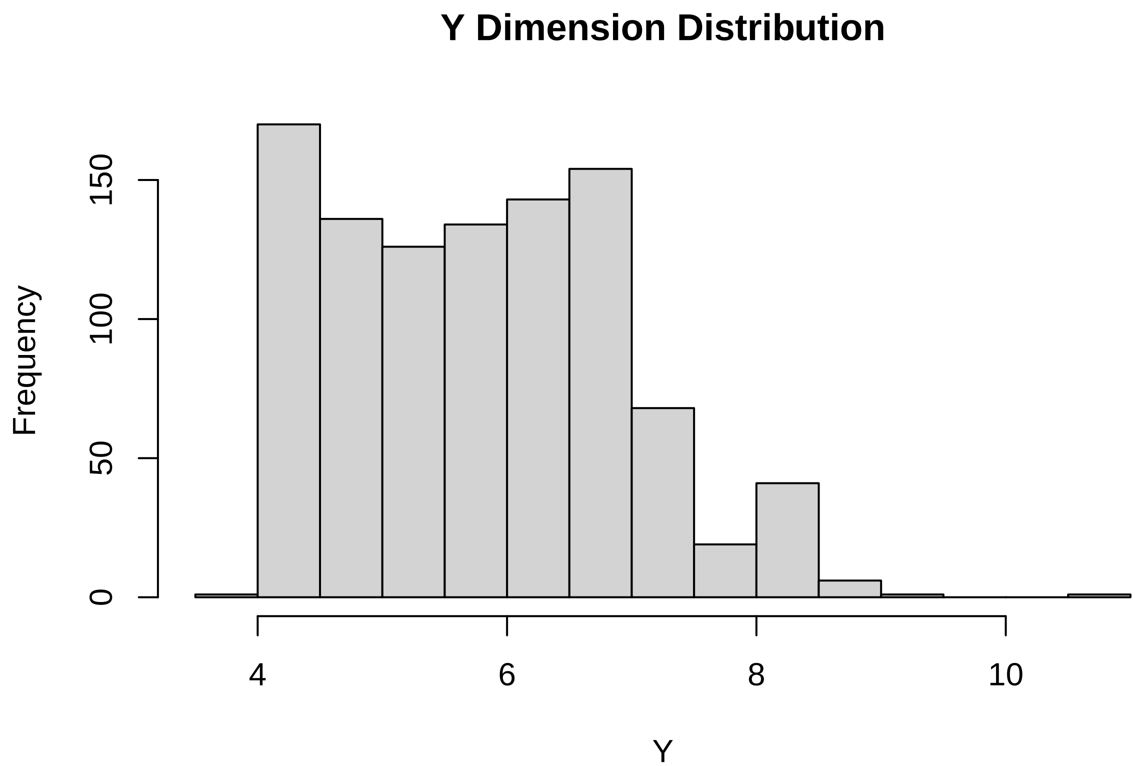

```r
hist(df$price, main="Price Distribution", xlab="Price")
```



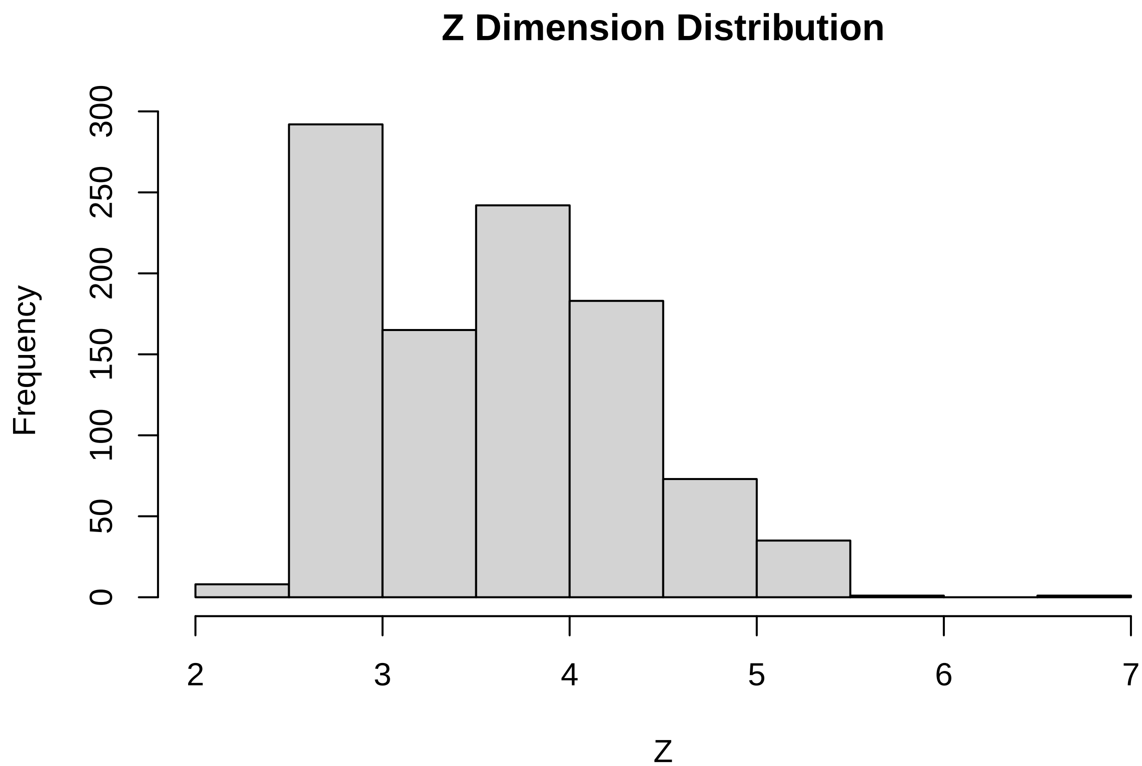

```
hist(df$x, main="X Dimension Distribution", xlab="X")
```



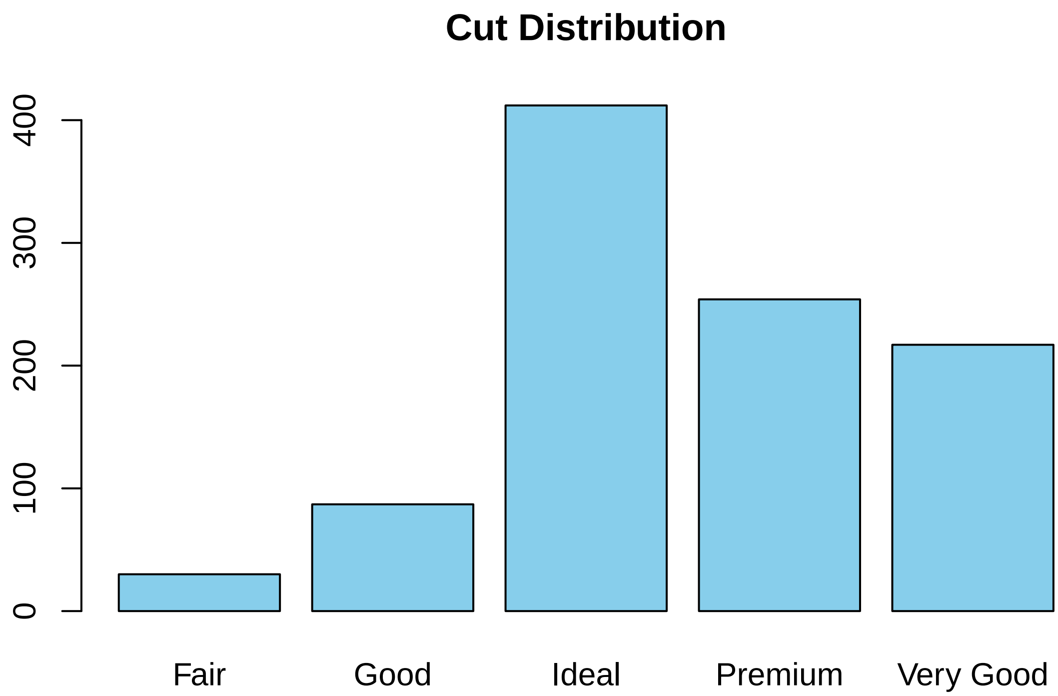

```
hist(df$y, main="Y Dimension Distribution", xlab="Y")
```



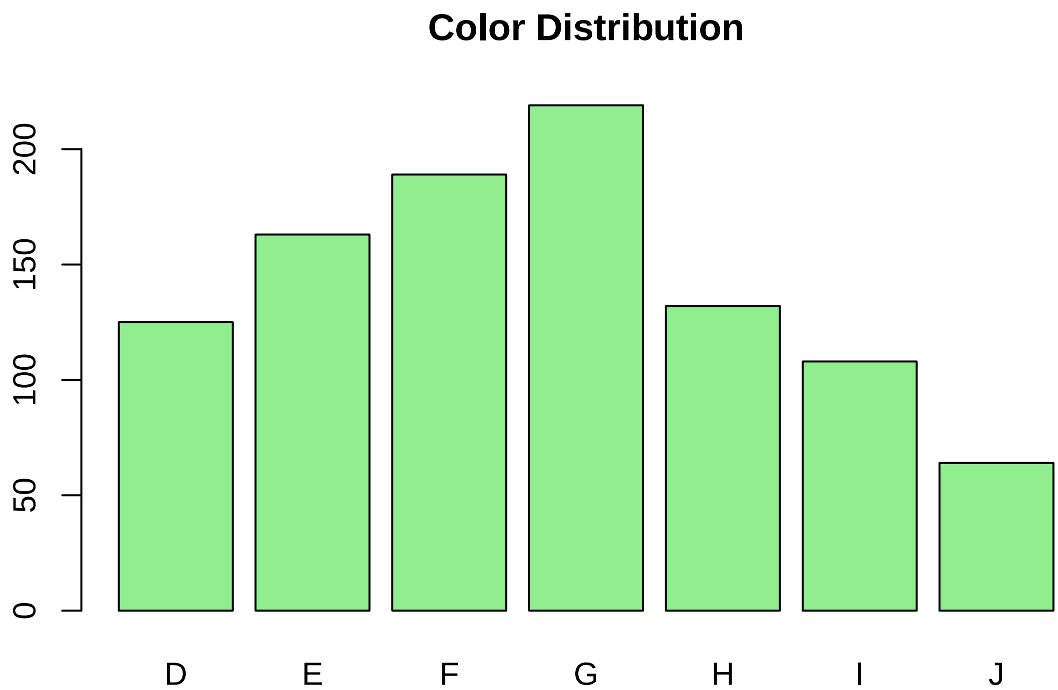

```r
hist(df$z, main="Z Dimension Distribution", xlab="Z")
```



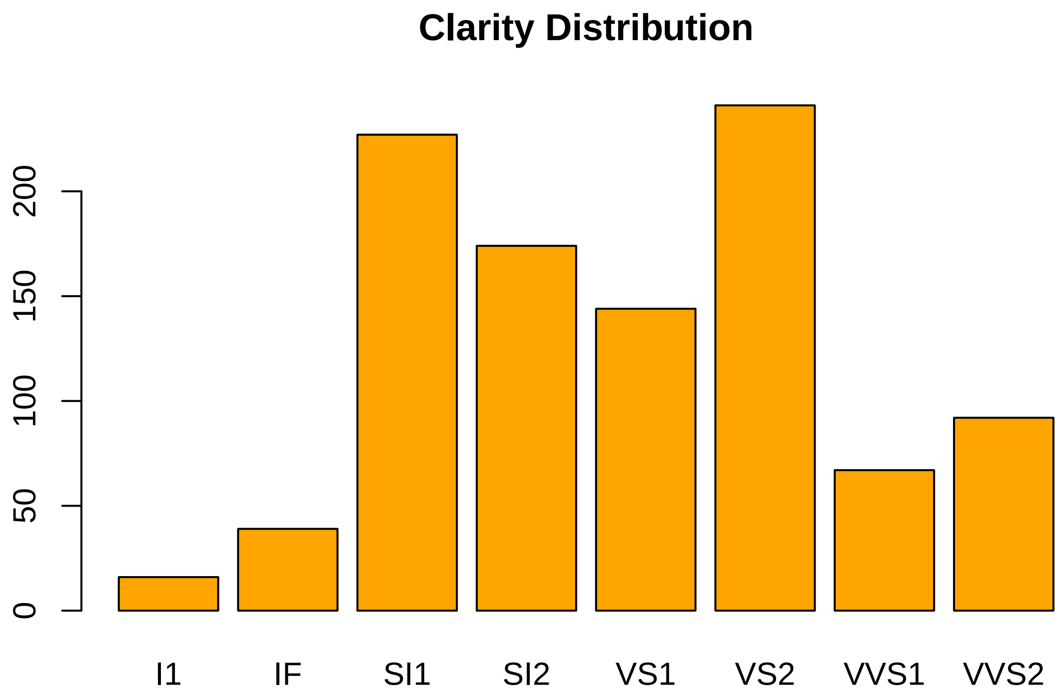

```
barplot(table(df$cut), main="Cut Distribution", col="skyblue")
```




```r
barplot(table(df$color), main="Color Distribution", col="lightgreen")
```




```r
barplot(table(df$clarity), main="Clarity Distribution", col="orange")
```



Brief comments on data distribution:

The 'carat' is skewed to the right, with a mean of 0.824 meaning that higher carat diamonds are rarer. The 'depth' is approximately normally distributed with a mean of 61.8. The 'table' variable is skewed to the right with a mean of 57.48. The 'price' variable is skewed to the right with a mean of 4177.2. The 'x', 'y', and 'z' variables are distributed mostly uniformly except for the tails, with means 5.8 for x and y, and 3.57 for y.

1.3, 1.4

```
numeric_vars <- df[, sapply(df, is.numeric)]
cor_matrix <- cor(numeric_vars)
print(cor_matrix)
```

```
##               ...1      carat       depth      table       price           x
## ...1   1.00000000 -0.3323137  0.01878306 -0.1362890 -0.27254017 -0.37534327
## carat -0.33231370  1.0000000  0.01737120  0.2115364  0.91622782  0.96998281
## depth  0.01878306  0.0173712  1.00000000 -0.3313397 -0.02066524 -0.04785965
## table -0.13628899  0.2115364 -0.33133970  1.0000000  0.14619125  0.23494972
## price -0.27254017  0.9162278 -0.02066524  0.1461912  1.00000000  0.88845693
## x     -0.37534327  0.9699828 -0.04785965  0.2349497  0.88845693  1.00000000
## y     -0.37687091  0.9687674 -0.05067842  0.2315989  0.88988711  0.99901283
## z     -0.37322439  0.9709881  0.07719590  0.1900128  0.88483988  0.99166296
##                 y          z
## ...1  -0.37687091 -0.3732244
```

```
## carat  0.96876737  0.9709881
## depth -0.05067842  0.0771959
## table  0.23159893  0.1900128
## price  0.88988711  0.8848399
## x      0.99901283  0.9916630
## y      1.00000000  0.9912899
## z      0.99128988  1.0000000
```

```r
df$cut <- as.factor(df$cut)
df$color <- as.factor(df$color)
df$clarity <- as.factor(df$clarity)

model <- lm(price ~ ., data = df[, -1])  # remove 'Unnamed: 0'
summary(model)
```

```
##
## Call:
## lm(formula = price ~ ., data = df[, -1])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -14293.8   -717.9   -207.5    504.9   6140.3
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -66115.76    9693.67  -6.821 1.59e-11 ***
## carat          9375.84     355.73  26.356  < 2e-16 ***
## cutGood         932.01     277.74   3.356 0.000822 ***
## cutIdeal       1275.43     270.32   4.718 2.73e-06 ***
## cutPremium     1351.22     260.04   5.196 2.48e-07 ***
## cutVery Good   1033.26     263.92   3.915 9.67e-05 ***
## colorE         -251.79     149.39  -1.685 0.092224 .
## colorF         -338.90     145.94  -2.322 0.020429 *
## colorG         -630.14     144.40  -4.364 1.41e-05 ***
## colorH         -952.31     159.96  -5.953 3.66e-09 ***
## colorI        -1582.84     171.23  -9.244  < 2e-16 ***
## colorJ        -2607.74     200.49 -13.007  < 2e-16 ***
## clarityIF      5499.21     394.43  13.942  < 2e-16 ***
## claritySI1     3364.74     341.90   9.841  < 2e-16 ***
## claritySI2     2523.90     343.32   7.351 4.14e-13 ***
## clarityVS1     4393.19     350.01  12.551  < 2e-16 ***
## clarityVS2     4037.19     341.74  11.814  < 2e-16 ***
## clarityVVS1    5352.85     369.73  14.478  < 2e-16 ***
## clarityVVS2    4800.33     360.84  13.303  < 2e-16 ***
## depth           986.09     155.06   6.360 3.10e-10 ***
## table           -31.20      24.58  -1.269 0.204681
## x              1815.19    1151.79   1.576 0.115355
## y              7884.83    1186.04   6.648 4.93e-11 ***
## z            -15943.72    2473.61  -6.446 1.81e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1251 on 976 degrees of freedom
## Multiple R-squared:  0.9157, Adjusted R-squared:  0.9137
```

```
## F-statistic:   461 on 23 and 976 DF,  p-value: < 2.2e-16
```

1.5

After fitting the multiple linear regression model, several interesting findings emerged. At the 0.05 significance level, nearly all variables were found to have a statistically significant effect on the mean price of a diamond—except for table and x, which showed no significant impact on price. This is notable because one might expect that all physical characteristics of a diamond would influence its value, yet table, which represents the width of the diamond's top facet, appears to have limited predictive power. We also see an oddly high p-value for color E relative to other colors.

The strongest correlation with price was observed for the carat variable, which makes sense since larger diamonds generally cost more. Additionally, the x, y, and z dimensions (length, width, and depth) also showed a high correlation with price, likely because they scale with carat and reflect the diamond's physical size. However, it's interesting that even though x is highly correlated with both price and the other dimensions, it did not retain statistical significance in the full regression model. This could suggest multicollinearity, where the effect of x is absorbed by other, highly related variables like y, z, and carat.

Overall, the results align with expectations in many ways: larger diamonds with better quality tend to cost more. owever, the lack of importance of the aforementioned variables despite their correlation highlights the complex interrelationships among the physical attributes and how not all apparent associations translate into independent contributions when modeled together.

Part 2

2.1

```r
model1 <- lm(price ~ carat, data = df)

summary(model1)
```

```
##
## Call:
## lm(formula = price ~ carat, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -18693.7   -865.7    -63.9    503.4   8769.9
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -2227.0      103.8  -21.46   <2e-16 ***
## carat         7772.2      107.6   72.24   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1707 on 998 degrees of freedom
## Multiple R-squared:  0.8395, Adjusted R-squared:  0.8393
## F-statistic:  5219 on 1 and 998 DF,  p-value: < 2.2e-16
```

2.2

```r
confint(model1)
```

```
##                 2.5 %    97.5 %
## (Intercept) -2430.617 -2023.303
## carat        7561.068  7983.302
```

```r
predict(model1, interval = "prediction")[1:5, ]
```

```
## Warning in predict.lm(model1, interval = "prediction"): predictions on current data refer to _future
```

```
##        fit       lwr       upr
## 1 5622.947  2272.084  8973.810
## 2 6167.000  2815.931  9518.069
## 3 5545.225  2194.386  8896.064
## 4 9819.927  6465.790 13174.064
## 5 1736.854 -1614.434  5088.143
```

The coefficient for carat is 7644.39, meaning that for every 1-unit increase in carat weight, the model predicts an increase of approximately $7,644 in price.

The intercept is -2191.34, which suggests that the model predicts a negative price when carat is zero—this is not meaningful in practice but expected for linear models outside of the data range.

The R^2 is 0.8547, meaning around 85.5% of the variability in diamond price is explained by carat alone.

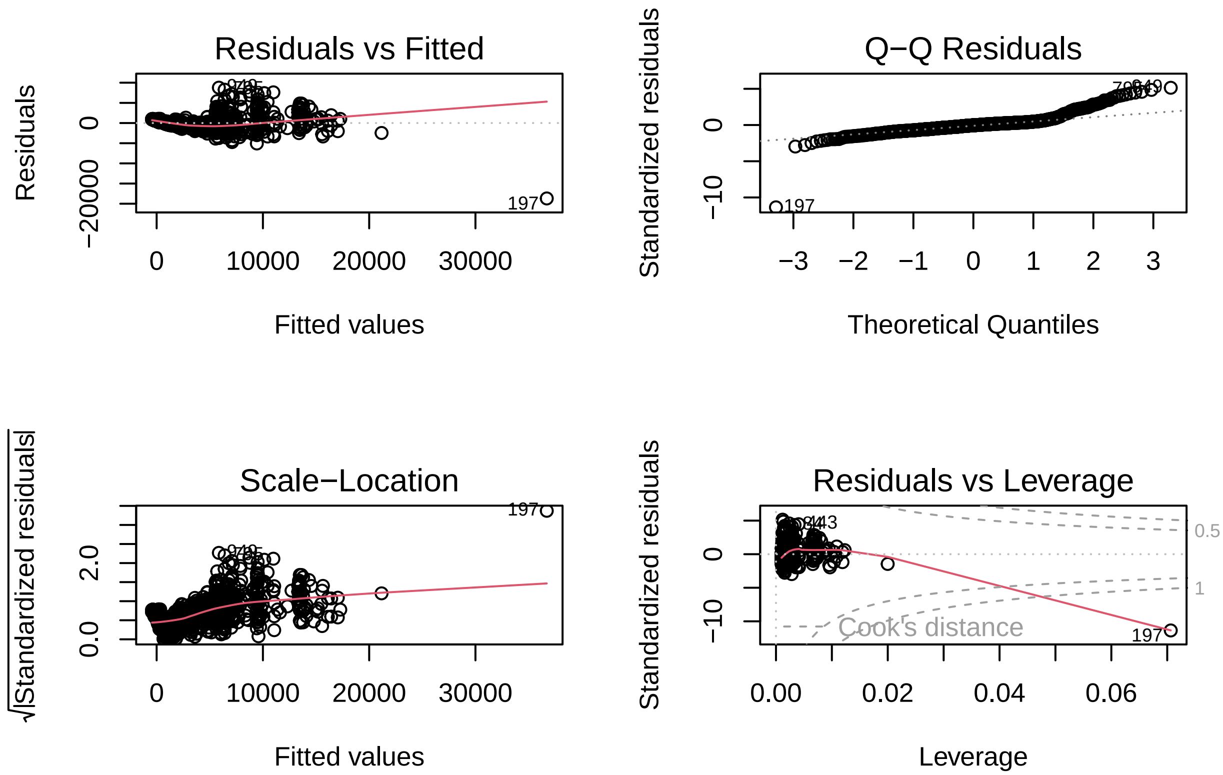
The F-statistic is very large (5870) with an extremely low p-value, strongly rejecting the null hypothesis that the model explains no variance.

The prediction intervals are very wide, especially at low and high carat values, reflecting variability in price not explained by carat alone.

Residual standard error is 1567, indicating considerable absolute prediction error.

2.3, 2.4

```r
par(mfrow = c(2, 2))
plot(model1)
```




```r
df$log_price <- log(df$price)
df$log_carat <- log(df$carat)

model1_log <- lm(log_price ~ log_carat, data = df)

summary(model1_log)
```

```
##
## Call:
## lm(formula = log_price ~ log_carat, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.36007 -0.16311 -0.00369  0.15789  1.07079
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 8.453921   0.009963   848.6   <2e-16 ***
## log_carat   1.678796   0.014415   116.5   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2676 on 998 degrees of freedom
## Multiple R-squared:  0.9315, Adjusted R-squared:  0.9314
## F-statistic: 1.356e+04 on 1 and 998 DF,  p-value: < 2.2e-16
```

The residuals show signs of non-constant variance and potential right-skewness, suggesting a transformation may improve the model.

The coefficient for log(carat) is 1.679, which means a 1% increase in carat is associated with a 1.68% increase in price. This makes interpretation more realistic for proportional change.

The R^2 increased to 0.9315, indicating that 93.1% of the variance in log(price) is now explained by log(carat), an improvement of almost 8 percentage points.

The residual standard error decreased dramatically to 0.2676, suggesting tighter model predictions.

The residuals are more symmetrically distributed with reduced heteroskedasticity, indicating better model assumptions are met.

2.5

```r
df$color <- as.factor(df$color)
model2 <- lm(log_price ~ log_carat + depth + color, data = df)

summary(model1)$adj.r.squared
```

```
## [1] 0.8393126
```

```r
summary(model1_log)$adj.r.squared
```

```
## [1] 0.9313919
```

```r
summary(model2)$adj.r.squared
```

```
## [1] 0.945586
```

2.6

The model has improved between each iteration, from the first model, to model1, to model1 log, and model2.

After modeling the relationship between carat and price, the adjusted R^2 indicated a strong linear association. However, the residual plots suggested non-constant variance and skewness, prompting a log transformation. This greatly improved model diagnostics. Adding variables like depth and color increased the adjusted R^2, indicating improved model fit. Interestingly, while carat dominated predictive power, color added subtle variation, reinforcing the importance of quality-based features in diamond pricing.

Part 3

3.1

```r
full_model <- lm(log_price ~ log_carat + depth + table + cut + color + clarity, data = df)
step_model_aic <- stepAIC(full_model, direction = "both", trace = FALSE)
summary(step_model_aic)
```

```
##
## Call:
## lm(formula = log_price ~ log_carat + cut + color + clarity, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.56640 -0.08076 -0.00320  0.07788  0.42168
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.873584   0.037126 212.078  < 2e-16 ***
## log_carat     1.898411   0.007883 240.819  < 2e-16 ***
## cutGood       0.039958   0.027563   1.450 0.147468
## cutIdeal      0.140689   0.025115   5.602 2.75e-08 ***
## cutPremium    0.116373   0.025348   4.591 4.98e-06 ***
## cutVery Good  0.093556   0.025662   3.646 0.000281 ***
## colorE       -0.051429   0.015077  -3.411 0.000673 ***
## colorF       -0.099875   0.014707  -6.791 1.93e-11 ***
## colorG       -0.166814   0.014552 -11.463  < 2e-16 ***
## colorH       -0.256311   0.016104 -15.916  < 2e-16 ***
## colorI       -0.370323   0.017177 -21.560  < 2e-16 ***
## colorJ       -0.567222   0.020070 -28.262  < 2e-16 ***
## clarityIF     1.131291   0.039319  28.772  < 2e-16 ***
## claritySI1    0.614765   0.033930  18.119  < 2e-16 ***
## claritySI2    0.444113   0.034170  12.997  < 2e-16 ***
## clarityVS1    0.833218   0.034679  24.027  < 2e-16 ***
## clarityVS2    0.759074   0.033981  22.338  < 2e-16 ***
## clarityVVS1   1.049522   0.036779  28.536  < 2e-16 ***
## clarityVVS2   0.965956   0.035923  26.889  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1263 on 981 degrees of freedom
## Multiple R-squared:  0.985,  Adjusted R-squared:  0.9847
## F-statistic:  3575 on 18 and 981 DF,  p-value: < 2.2e-16
```

The final model was selected using stepwise regression based on AIC, balancing goodness of fit and model complexity. After evaluating multiple combinations of predictors, the final model retained log_carat, depth, and a subset of the quality indicators (cut, color, and clarity), all of which had meaningful contributions to predicting log_price.

3.2

```r
vif(step_model_aic)
```

```
##               GVIF Df GVIF^(1/(2*Df))
## log_carat 1.341789  1        1.158356
## cut       1.207200  4        1.023817
## color     1.242259  6        1.018242
## clarity   1.460332  7        1.027417
```

There is no significant multicollinearity present in the model since all VIF factors are close to 1.

3.3

```r
new_data <- data.frame(
    log_carat = log(1),
    depth = 61.5,
    table = 55,
    cut = factor("Ideal", levels = levels(df$cut)),
    color = factor("E", levels = levels(df$color)),
    clarity = factor("VS2", levels = levels(df$clarity))
)

ci_log <- predict(step_model_aic, newdata = new_data, interval = "confidence")

pi_log <- predict(step_model_aic, newdata = new_data, interval = "prediction")

ci_original <- exp(ci_log)
pi_original <- exp(pi_log)

ci_original
```

```
##        fit      lwr      upr
## 1 6135.937 5967.845 6308.763
```

```r
pi_original
```

```
##        fit      lwr     upr
## 1 6135.937 4781.154 7874.61
```

We are 95% confident that the mean price of diamonds with these characteristics lies between 5,967.85 and 6,308.76. This reflects the average expected value for all diamonds with these features.

We are 95% confident that the price of a single future diamond with these characteristics will fall between 4,781.15 and 7,874.61. This is wider than the CI because it accounts for both model uncertainty and individual variation in price.

3.4

In this analysis, we explored various factors influencing diamond prices using regression modeling techniques. Our initial investigation began with a simple linear regression model using carat as the sole predictor of price. This model had an adjusted R^2 of 0.8545, demonstrating a strong positive linear relationship between carat and price. However, the residual plots showed non-constant variance and skewness, prompting a log transformation.

After transforming both the response (price) and the predictor (carat) using log, the model significantly improved. The log model increased the adjusted R^2 to 93.1% and reduced the residual standard error, indicating a better fit and more reliable predictions.

Next, we applied stepwise regression using AIC to select a more comprehensive model that included additional variables like depth, cut, color, and clarity. The resulting model achieved an adjusted R^2 of 94.6%, indicating that a substantial portion of price variability can be explained by a combination of carat and quality characteristics. We verified this model using VIF analysis, which showed no major multicollinearity among the retained predictors.

For a sample diamond (1 carat, Ideal cut, E color, VS2 clarity, 61.5 depth, 55 table), we predicted a price of approximately 6,135.94. The 95% confidence interval for the mean price ranged from 5,967.85 to 6,308.76, while the 95% prediction interval for a single future diamond ranged more widely, from 4,781.15 to 7,874.61.

Throughout the modeling process, we found that:

carat was by far the strongest predictor. Variables like table and x had little to no significant impact. Log-transformation greatly improved model fit and interpretation. Quality-related factors like color and clarity refined the model's accuracy.

This analysis confirms that while carat is the primary driver of diamond price, incorporating other physical and qualitative attributes leads to a more accurate and interpretable predictive model.