ALY6040 Final Project Milestone 1: EDA

Yinan Zhou

October 3, 2024

1. Introduction

Our proposal of analyzing diamonds based on the dataset diamonds.csv has been approved. This dataset contains the 10 attributes of 53940 diamonds. We assume the data owners are diamond merchants, who are eager to discover practical strategies of evaluating the diamond prices. Therefore, we select `price` as the target, and use the remaining 9 varaibles: `carat`, `cut`, `color`, `clarity`, `x`, `y`, `z`, `depth` and `table` as the features. The definition of each variable is available in Appendix A.

This report will cover the first part of our final project: Exploratory Data Analysis (EDA), which is a crucial first step in any data analysis or modeling process. It involves examining and summarizing the main characteristics of the data, often with visual methods, to discover patterns, spot anomalies, check assumptions, and decide how to proceed with more complex analysis.

In the following sections, we will first process the data in the following sequence: provide an overview of the dataset, check for missing data and duplication, identify outliers and suspicious data, and verify equations to clean the dataset. Next, we will analyze the characteristics of the data, including data distributions and correlations between variables. Finally, we will summarize our findings and centralize the answers to all assignment questions in Appendix B.

2. Data Cleaning

2.1 Overview

We load the data using function `read.csv`, and store it in the variable `raw_data`.

The raw data has 53940 rows and 11 columns. However, the first column 'X' is the useless index. We removed it. Therefore, the dataset has 53940 entries and 10 variables.

The target variable is price, which is numerical, ranging from 326 to 18823. The average price of the 53940 diamonds is 3933. The rest 9 features consists of 3 categorical variables and 6 numerical variables. Due to space constraints, details are printed in Appendix C1.

```
raw_data <- read.csv("diamonds.csv") # Reading the data set as a dataframe
diamonds <- raw_data[-1] # Remove the column of entry indices
dim(diamonds) # Print dimensions of the dataset
```

```
## [1] 53940    10
```

2.2 Missing Data and Duplications

We subtract the row numbers without missing data from the total number of rows to get the rows with missing data. Fortunately, this dataset do not have any missing value.

However, we catch 146 duplicated rows (not include their first appearance). These duplicated entry pairs have exactly the same price and all other attributes. After removing all duplications, 53794 entries remains in the dataset.

```
# Print number of rows with missing values
nrow(diamonds) - sum(complete.cases(diamonds))
```

```
## [1] 0
```

```
# Find columns with only one unique value
useless_columns <- sapply(diamonds, function(x) length(unique(x)) == 1)
# Check the duplicated entries
print(paste("The number of duplicated row:", sum(duplicated(diamonds))))
```

```
## [1] "The number of duplicated row: 146"
```

```
duplicated_rows <- diamonds[duplicated(diamonds)
                        | duplicated(diamonds,fromLast = TRUE), ]
print(duplicated_rows[288:289,])# Print the last 2 duplicated rows
```

```
##       carat  cut color clarity depth table price    x    y    z
## 52861   0.5 Fair     E     VS2    79    73  2579 5.21 5.18 4.09
## 52862   0.5 Fair     E     VS2    79    73  2579 5.21 5.18 4.09
```

```
diamonds_clean <- diamonds[!duplicated(diamonds),]# Remove duplicated rows
```

2.3 Outliers and Suspicious Data

The outliers of numerical variables are visualized using boxplots, shown in Appendix C2 due to space constraints. The outliers are represented by dots in the box plots. There are many outliers in every numerical variable.

Some outliers are suspicious. For example, the entries with x = 0 or y = 0 or z = 0. Any one of these variables equals 0 should be wrong. We print 3 examples of such outliers. There are two situations: (1) x = 0, y = 0 and z = 0; and (2) x > 0, y > 0 and z = 0. For the first situation, we directly remove the corresponding entries because they are unable to be calculated. For the second situation, we calculate the z value using the known formula: depth = 200*z/(x+y). After our processing, only 7 suspicious outliers need be removed. Now there are 53787 entries in the dataset.

```
numeric_data <- diamonds_clean[,sapply(diamonds_clean,is.numeric)]
# Collect all suspicious outliers with x = 0 or y = 0 or z = 0
suspicious_outliers <- diamonds_clean[(diamonds_clean$x == 0)
              | (diamonds_clean$y == 0) | (diamonds_clean$z == 0),]
print(paste("The number of suspicious outliers:", nrow(suspicious_outliers)))
```

```
## [1] "The number of suspicious outliers: 19"
```

```r
print(suspicious_outliers[17:19,]) # Print the last 3 suspicious outliers.
```

```
##       carat     cut color clarity depth table price    x    y z
## 27740  2.80    Good     G     SI2  63.8    58 18788 8.90 8.85 0
## 49557  0.71    Good     F     SI2  64.1    60  2130 0.00 0.00 0
## 51507  1.12 Premium     G      I1  60.4    59  2383 6.71 6.67 0
```

```r
# Fix the suspicious outliers that can be fixed
outliers_can_fix <- which(diamonds_clean$x > 0
                          & diamonds_clean$y > 0 & diamonds_clean$z == 0)
diamonds_clean$z[outliers_can_fix] <- diamonds_clean$depth[outliers_can_fix] *
  (diamonds_clean$x[outliers_can_fix] + diamonds_clean$y[outliers_can_fix])/200

remaining_suspicious_outliers <- diamonds_clean[(diamonds_clean$x == 0)
              | (diamonds_clean$y == 0) | (diamonds_clean$z == 0),]
print(paste("The remaining suspicious outliers:", nrow(remaining_suspicious_outliers)))
```

```
## [1] "The remaining suspicious outliers: 7"
```

```r
# Remove the 7 remaining suspicious outliers with x=0, y=0 and z=0
diamonds_clean <- diamonds_clean[!(diamonds_clean$x == 0 |
                  diamonds_clean$y == 0 | diamonds_clean$z == 0), ]
```

2.4 Clean Dataset

Until now, we have already cleaned the data by (1) removing the useless columns of the row index; (2) removing 146 repeated rows; (3) fixing 12 rows with z = 0, x, y > 0; (4) removing 7 rows with z = 0, x = 0 and y = 0, which are the suspicious outliers that is unable to fix.

Lastly, we verify that each entry conforms to the predefined equation: 100*depth = z/mean(x,y). Due to the `x`, `y`, and `z` values being recorded with only two decimal places, we allow a tolerance of 3% to account for rounding differences when verifying the accuracy of the calculated depth against the recorded depth. A total of 49 entries show a mismatch between the calculated and recorded depths. The maximum error is 902%! This entry calculated depth is 619.3 while the recorded depth is 61.8. These entries are removed to ensure the reliability of the dataset. Observing the boxplots again in Appendix C2, we decide that larger outliers in `y` and `z` should also be removed. We store the cleaned dataset in `diamonds_cleaned`, which contains 53738 rows and 10 columns.

We do not consider `depth` as a duplicated attribute, even though it can be derived from other attributes. This is because `depth` results from feature engineering, potentially providing a better representation of diamond pricing compared to the original attributes such as `x`, `y`, `z`.

```r
# Verify the equation of calculating depth
diamonds_clean$calculated_depth <- with(diamonds_clean, round(100*z / ((x + y) / 2),1))
diamonds_clean$depth_error <- with(diamonds_clean,
                                   round(abs(depth - calculated_depth) /depth,2))
invalid_entries <- diamonds_clean[diamonds_clean$depth_error>0.03,]
print(paste("The number of entries with invalid depth or x,y,z:", nrow(invalid_entries)))
```

```
## [1] "The number of entries with invalid depth or x,y,z: 49"
```

```r
# Print the 3 entries with mismatch depth.
print(invalid_entries[37:39,c("x", "y", "z", "carat",
                      "depth", "calculated_depth", "depth_error")])
```

```
##          x     y     z carat depth calculated_depth depth_error
## 48411 5.12  5.15 31.80  0.51  61.8           619.3        9.02
## 48833 5.16  6.20  3.25  0.53  62.7            57.2        0.09
## 49190 5.15 31.80  5.12  0.51  61.8            27.7        0.55
```

```r
# Clean the dataset: remove entries with more than 3% depth error or larger outliers
diamonds_cleaned <- diamonds_clean[(diamonds_clean$depth_error <= 0.03)
                                   & (diamonds_clean$y<30 | diamonds_clean$z <30), ]
diamonds_cleaned$calculated_depth <- NULL; diamonds_cleaned$depth_error <- NULL
write.csv(diamonds_cleaned, "diamonds_cleaned.csv", row.names = FALSE) # Save csv
dim(diamonds_cleaned)
```

```
## [1] 53738    10
```

3. Exploratory Analysis

3.1 Data Distribution

We visualize the 7 continuous numerical variables by the histograms and visualize the 3 categorical variables by the bar plots, as shown in Appendix C3. The variables `depth` and `table` are normally distributed. The distributions of `carat`, `price`, `x`, `y`, and `z` are right-skewed, meaning most diamonds have lower carat, lower price and smaller dimensions. The distribution of `cut` is fairly uneven, with a large proportion of diamonds rated as "Very Good," followed by "Premium" and "Ideal." There are fewer "Fair" and "Good" diamonds. The color distribution is relatively balanced, with most diamonds having a color grade between D and G. Grades H to J appear less frequently. Most diamonds fall into the SI1 and VS2 clarity categories, indicating slight inclusions or very slight inclusions. The I1 and IF categories are much less common.

3.2 Variable Correlation

For the 7 numerical variables, we use `corrplot()` to visualizes the correlation matrix, as shown in Appendix C4. Our interpretations are: (1) Carat, dimensions (x, y, z), and price are strongly related to each other and (2) Depth and table do not significantly affect the price of diamonds, and depth shows a weak inverse relationship with carat size.

4. Discussion

We have completed the EDA section of the final project. Though the carefully sanity checks, we have cleaned the raw dataset by removing one column and 202 entries, where 146 are duplicated, 7 are obvious wrong outliers, 49 are suspicious data. Moreover, we have fixed 12 suspicious outliers using the given equations. In the context of exploration, we have investigated the distribution of each variable and the correlations between numerical variables. The 5 right-skewed varaibles are strongly related to each other. The variables `depth` and `table` are normally distributed, and they do not significantly affect the price of diamonds.

We have saved the cleaned dataset into a new csv file for our next step, which is data mining, to provide a clear guidance for the data owners to estimate the price of diamonds. Due to space constraints, the full discussion can be found in Appendix B.

Appendix A: Dataset Information

Definition of all variables in the dataset (ranges are listed in the parenthesis):

- price: price in US dollars ($326–$18,823)
- carat: weight of the diamond (0.2–5.01)
- cut: quality of the cut (Fair, Good, Very Good, Premium, Ideal)
- color: diamond color, from J (worst) to D (best)
- clarity: a measurement of how clear the diamond is (I1 (worst), SI2, SI1, VS2, VS1, VVS2, VVS1, IF (best))
- x: length in mm (0–10.74)
- y: width in mm (0–58.9)
- z: depth in mm (0–31.8)
- depth: total depth percentage = z / mean(x, y) = 2 * z / (x + y) (43–79)
- table: width of top of diamond relative to widest point (43–95)

Appendix B: Answers to Assignment Questions

- **What did you do with the data in the context of exploration?**

In the context of exploration, we checked and processed the data in five steps:

(1) we viewed the dimensions of the data: 53940 entries × 10 variables, as well as the type and statistic of each variable.

(2) we checked the missing and duplicated data: 0 missing and 146 duplicated. All of them are removed from the dataset.

(3) we checked the outliers and suspicious data: 7 obvious outliers and 49 suspicious data with mismatch depths are removed from the dataset. In addition, 12 outliers with z = 0 are fixed by using the given formula.

(4) we investigated the distribution of each variables: 2 are normally distributed, 5 are right-skewed distributed, 2 categorical are uneven, and 1 categorical is relatively balanced.

(5) we investigated the correlations between numerical variables: the 5 right-skewed distributed variables are strongly related to each other. 2 normally distributed variables `depth` and `table` are less related to the diamond price.

- **How many entries are in the dataset?**

In the raw dataset, there are 53940 entries.

In the cleaned dataset, there are 53738 entries.

- **Was there missing data? Duplications? How clean was the data?**

There was no missing data.

There were 146 duplicated rows.

The data was clean in general: Most of the entries were good. Only 0.4% entries were duplicated or suspicious.

- **Were there outliers or suspicious data?**

Yes, each variable had outliers by observing the boxplots. There were 19 obvious wrong outliers in variables `x`, `y` and `z`, we fixed 12 of them by using the given depth formula. Only 7 obvious wrong outliers were removed.

There are 49 suspicious data when we are verifying the depth formula. We removed them to get a cleaned dataset.

- **What did you do to clean the data?**

We cleaned the data by removing duplicated rows, removing obvious wrong outliers, fixing wrong outliers using existed information, and removing suspicious data with mismatch `depth`.

- **What did you find? What intrigued you about the data? Why does that matter?**

After cleaning the dataset, we found the following:

(1) The variables `depth` and `table` are normally distributed. The distributions of `carat`, `price`, `x`, `y`, and `z` are right-skewed, meaning most diamonds have lower carat, lower price and smaller dimensions. The distribution of `cut` is fairly uneven, with a large proportion of diamonds rated as "Very Good," followed by "Premium" and "Ideal." There are fewer "Fair" and "Good" diamonds. The color distribution is relatively balanced, with most diamonds having a color grade between D and G. Grades H to J appear less frequently. Most diamonds fall into the SI1 and VS2 clarity categories, indicating slight inclusions or very slight inclusions. The I1 and IF categories are much less common.

(2) Carat, dimensions (x, y, z), and price are strongly related to each other.

(3) Depth and table do not significantly affect the price of diamonds, and depth shows a weak inverse relationship with carat size.

There are several things intrigued me about the data, for example:

(1) Figure out the attributes that largely affect the price of diamonds.

(2) Find a method to predict the price of diamonds given the attributes.

They matter because they are what the data owner would like to discover from the dataset.

- **What would your proposed next steps be?**

The next steps we proposed to do: (1) Select several models and fit them to the regression problem of predicting the price.

(2) Evaluate the model qualities based on the testing mean square error (MSE).

(3) Interpret the results of analysis and explain what the results mean for the data owner.

(4) Provide recommendations for actions to be taken based on the data mining and interpolation.

- **What business questions do you plan to answer with your data mining?**

The business questions we plan to answer:

(1) What is the critical attributes that affect the price of diamonds?

(2) How to estimate the price of diamonds based on the attributes?

(3) Can we tell a price of a diamond is overestimated or underestimated? We will suggest the data owner to buy underestimated diamonds and sell overestimated diamonds.

Appendix C : Code Results Occupying Space

C1. Subsection 2.1: Variable types and statistics

```
# Print variable types and statistics
summary(diamonds)
```

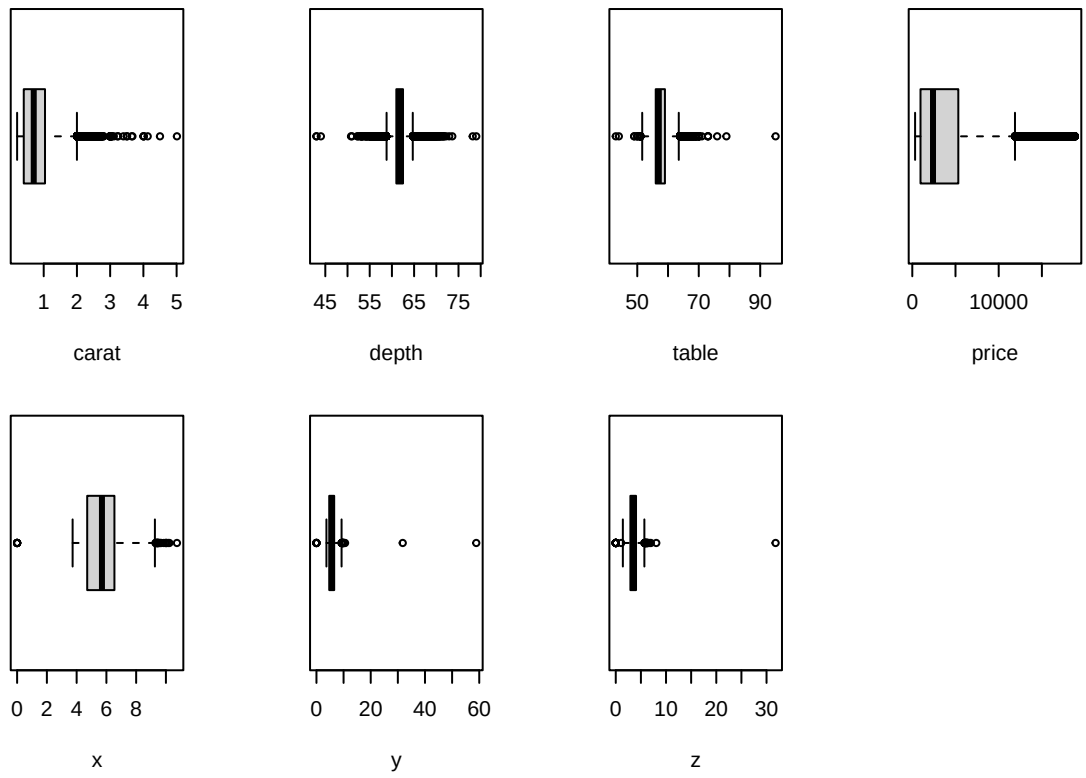
```
##      carat               cut                color              clarity
##  Min.   :0.2000   Length:53940       Length:53940       Length:53940
##  1st Qu.:0.4000   Class :character   Class :character   Class :character
##  Median :0.7000   Mode  :character   Mode  :character   Mode  :character
##  Mean   :0.7979
##  3rd Qu.:1.0400
##  Max.   :5.0100
##      depth           table           price             x
##  Min.   :43.00   Min.   :43.00   Min.   :  326   Min.   : 0.000
##  1st Qu.:61.00   1st Qu.:56.00   1st Qu.:  950   1st Qu.: 4.710
##  Median :61.80   Median :57.00   Median : 2401   Median : 5.700
##  Mean   :61.75   Mean   :57.46   Mean   : 3933   Mean   : 5.731
##  3rd Qu.:62.50   3rd Qu.:59.00   3rd Qu.: 5324   3rd Qu.: 6.540
##  Max.   :79.00   Max.   :95.00   Max.   :18823   Max.   :10.740
##        y                z
##  Min.   : 0.000   Min.   : 0.000
##  1st Qu.: 4.720   1st Qu.: 2.910
##  Median : 5.710   Median : 3.530
##  Mean   : 5.735   Mean   : 3.539
##  3rd Qu.: 6.540   3rd Qu.: 4.040
##  Max.   :58.900   Max.   :31.800
```

```
str(diamonds)
```

```
## 'data.frame':    53940 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 ...
```

C2. Subsection 2.3: Boxplots of 7 numerical variables to find outliers

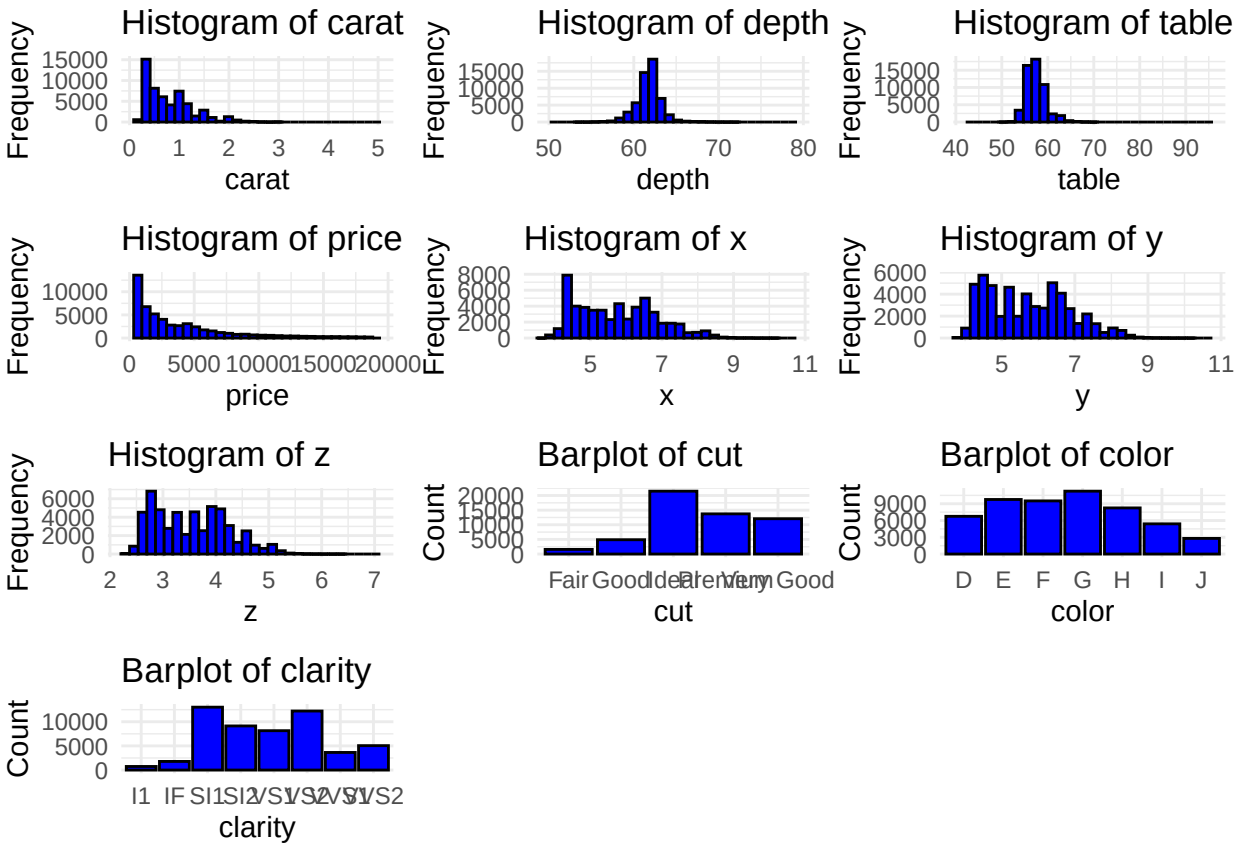
```
par(mfrow = c(2, 4), mar = c(4, 4, 2, 1), oma = c(1, 1, 1, 1))
for (i in 1:7){boxplot(numeric_data[, i],
                       xlab=colnames(numeric_data)[i],horizontal = TRUE)}
```



C3. Subsection 3.1: histograms and bar plots

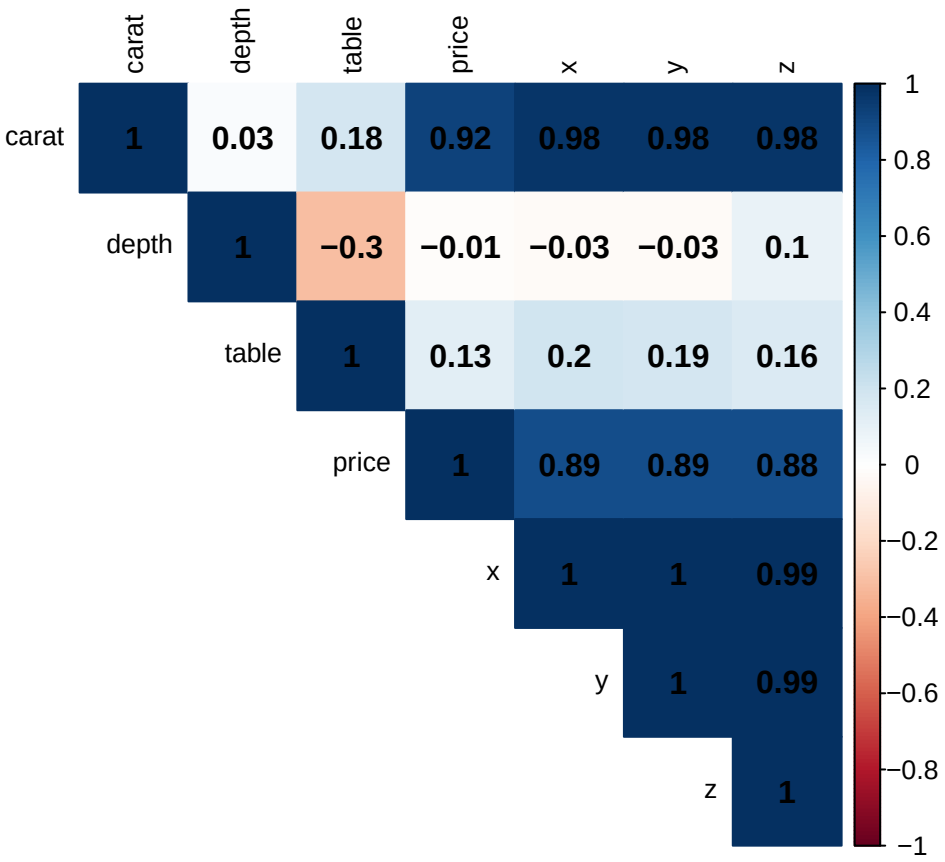
```r
# Load necessary libraries
library(ggplot2)
library(gridExtra)
# Separate numeric and categorical variables
numeric_vars <- names(diamonds_cleaned)[sapply(diamonds_cleaned, is.numeric)]
categorical_vars <- names(diamonds_cleaned)[sapply(diamonds_cleaned, is.factor)
                    | sapply(diamonds_cleaned, is.character)]
# Create an empty list to store plots
plot_list <- list()
# Visualize numerical variables using histograms
for (var in numeric_vars) {
  p <- ggplot(diamonds_cleaned, aes(x = .data[[var]])) +
    geom_histogram(fill = "blue", color = "black") +
    ggtitle(paste("Histogram of", var)) +
    theme_minimal() + xlab(var) + ylab("Frequency")
  plot_list[[length(plot_list) + 1]] <- p
}
# Visualize categorical variables using bar plots
for (var in categorical_vars) {
  p <- ggplot(diamonds_cleaned, aes(x = .data[[var]])) +
    geom_bar(fill = "blue", color = "black") +
    ggtitle(paste("Barplot of", var)) +
```

```
    theme_minimal() + xlab(var) + ylab("Count")
  plot_list[[length(plot_list) + 1]] <- p
}
# Arrange the plots in a 2 by 5 grid
grid.arrange(grobs = plot_list, ncol = 3, nrow = 4)
```



C4. Subsection 3.2: Correlation matrix

```
# Load necessary libraries
library(corrplot)
# Select only numeric variables from the dataset
numeric_vars <- diamonds_cleaned[sapply(diamonds_cleaned, is.numeric)]
# Compute the correlation matrix
cor_matrix <- cor(numeric_vars, use = "complete.obs")
# Visualize the correlation matrix
corrplot(cor_matrix, method = "color", type = "upper",
         tl.cex = 0.8, tl.col = "black", addCoef.col = "black")
```


carat
depth
table
price
x
y
z
carat 1 0.03 0.18 0.92 0.98 0.98 0.98
depth 1 −0.3 −0.01 −0.03 −0.03 0.1
table 1 0.13 0.2 0.19 0.16
price 1 0.89 0.89 0.88
x 1 1 0.99
y 1 0.99
z 1
1
0.8
0.6
0.4
0.2
0
−0.2
−0.4
−0.6
−0.8
−1