Title: Peer-graded Assignment: ETL and EDA Using R

Name: Muhammad Rizqi Winnel Adnin
****

*Load needed Package

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.3
```

```
## Warning: package 'lubridate' was built under R version 4.4.3
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.4     ✓ tidyr     1.3.1
## ✓ purrr     1.0.4
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
library(lubridate)
```

*Read the Data

```
df <- read.csv("mod4_peer_rev_data.csv")
```

1. Transform columns so that datatypes are appropriate. Specifically ensure that the CustomerCode variable is formatted as character, any other categorical variable is set as factor, and date column is set as a date type (Date/POSIXlt/POSIXct). 3 points

```
df$CustomerCode <- as.character(df$CustomerCode)
df$Department <- as.factor(df$Department)
df$Category <- as.factor(df$Category)
df$Date <- mdy(df$Date)
summary(df)
```

```
##       Date              Department                 Category
##  Min.   :2014-01-02   Entrees: 5670   Chicken           :9046
##  1st Qu.:2014-09-11   Kabobs :18129   Beef              :6249
##  Median :2015-06-15   Sides  :10633   Yogurt            :5898
##  Mean   :2015-07-05                   Beef and Broccoli:2885
##  3rd Qu.:2016-04-18                   Rice              :2835
##  Max.   :2017-04-03                   Lamb Chops        :2785
##                                       (Other)           :4734
##  CustomerCode           Price          Quantity
##  Length:34432       Min.   : 3.00   Min.   : 1.00
##  Class :character   1st Qu.:12.00   1st Qu.: 8.00
##  Mode  :character   Median :25.00   Median :11.00
##                     Mean   :22.81   Mean   :11.31
##                     3rd Qu.:33.00   3rd Qu.:15.00
##                     Max.   :50.00   Max.   :24.00
##                     NA's   :10      NA's   :7
```

2. Display and interpret the summaries for the Quantity and Price columns. 4 points

```
summary(select(df, Quantity, Price))
```

```
##     Quantity         Price
##  Min.   : 1.00   Min.   : 3.00
##  1st Qu.: 8.00   1st Qu.:12.00
##  Median :11.00   Median :25.00
##  Mean   :11.31   Mean   :22.81
##  3rd Qu.:15.00   3rd Qu.:33.00
##  Max.   :24.00   Max.   :50.00
##  NA's   :7       NA's   :10
```

For the Quantity column, the minimum value is 1, while the maximum value is 24. The median value provide an idea of the central tendency of sales, and the mean can indicate the average number of item sold per transaction. Any significant difference between the mean and median suggests skewness in the data. For the Price column, the minimum and maximum values indicate the range of product prices in the dataset. The median price helps identify the central value, while the mean price gives an overall sense of the average pricing. Large differences between the median and mean suggest the presence of high or low outliers affecting the price distribution.

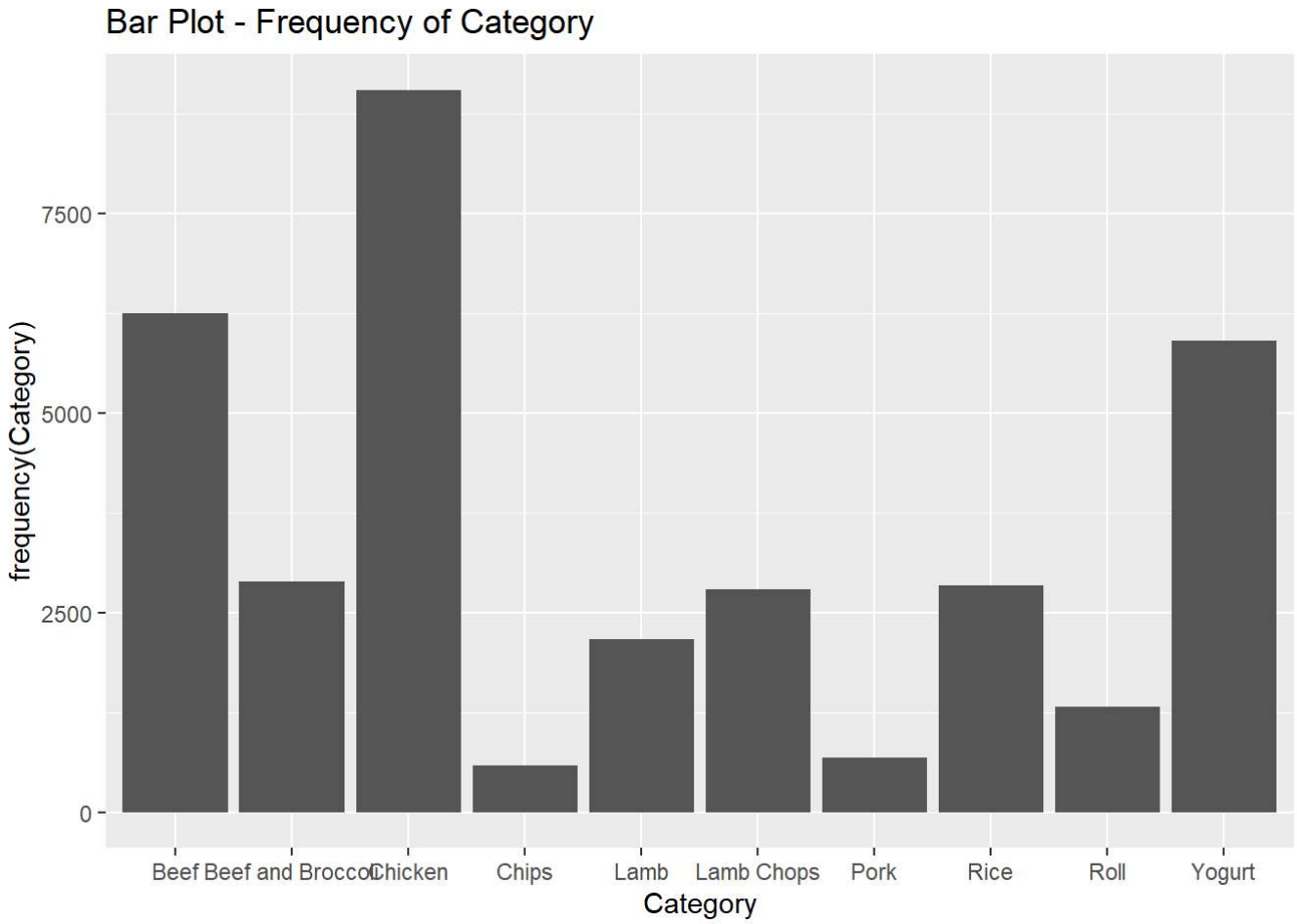
3. Display the count of NA values in each column. 1 point

```
colSums(is.na(df))
```

```
##         Date   Department     Category CustomerCode        Price     Quantity
##            0            0            0            0           10            7
```

4. Display a bar chart for Category column. The bar chart should display the frequency of each category. 2 points

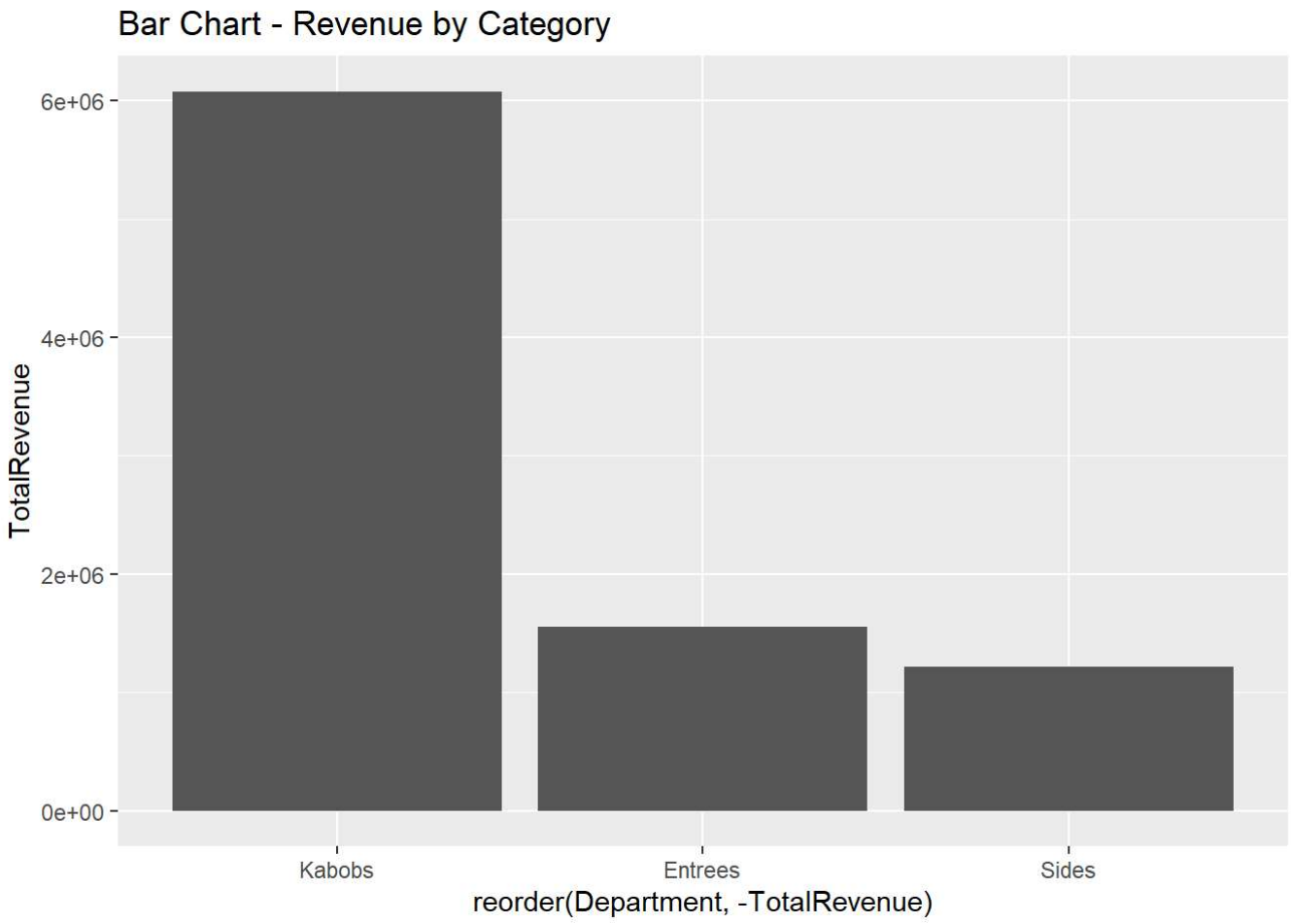
```
ggplot(df, aes(x = Category, y = frequency(Category))) +
  geom_col() +
  labs(title = 'Bar Plot - Frequency of Category')
```



5. Display the Departments and their revenue using a bar chart. Order the bars in a meaningful way. 4 points.

Hint: You will need to create a new column Revenue by multiplying Price and Quantity.

```
df$Revenue <- df$Price * df$Quantity
df1 <- df %>% group_by(Department) %>% summarise(TotalRevenue = sum(Revenue, na.rm = TRUE))
ggplot(df1, aes(x = reorder(Department, -TotalRevenue), y = TotalRevenue)) +
  geom_col() +
  labs(title = 'Bar Chart - Revenue by Category')
```



6. Create a histogram and box and whisker plot of the Price and Quantity columns. 4 points You are creating 4 different univariate plots:

a. box plot of the price column

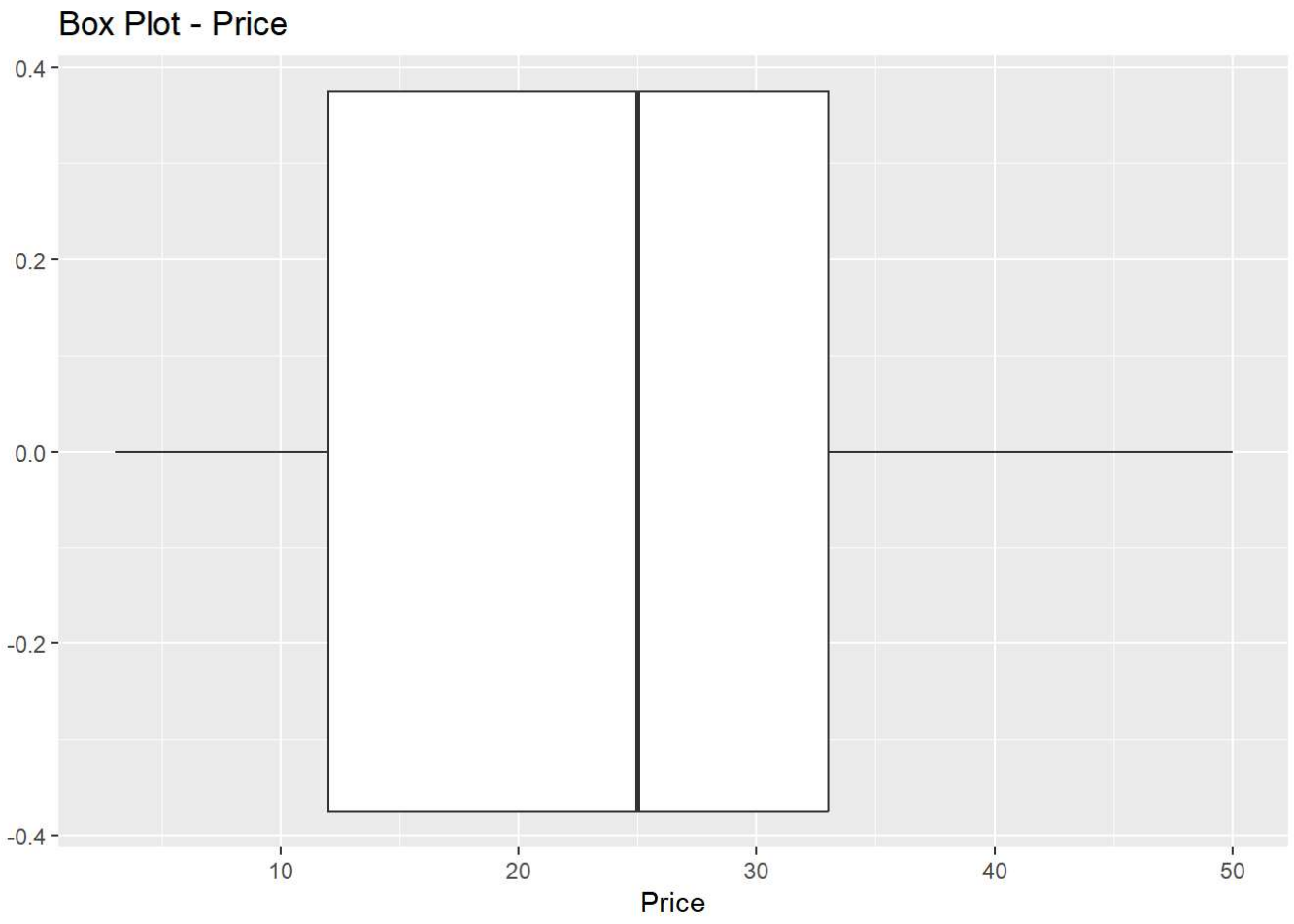
b. a histogram of the price column

c. a box plot of the quantity column

d. a histogram of the quantity column

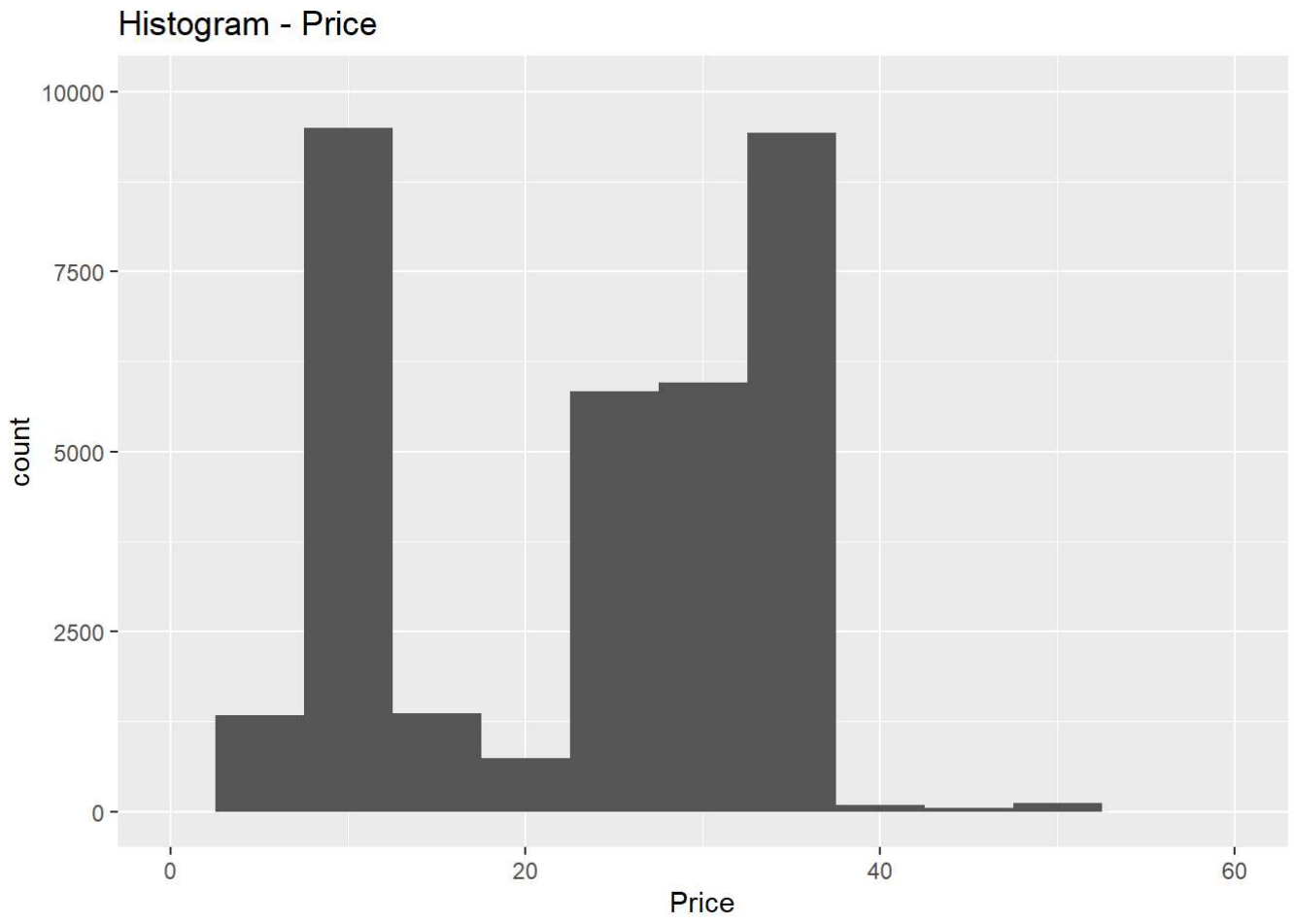
```
ggplot(df, aes(x=Price)) +
  geom_boxplot() +
  labs(title = 'Box Plot - Price')
```

```
## Warning: Removed 10 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



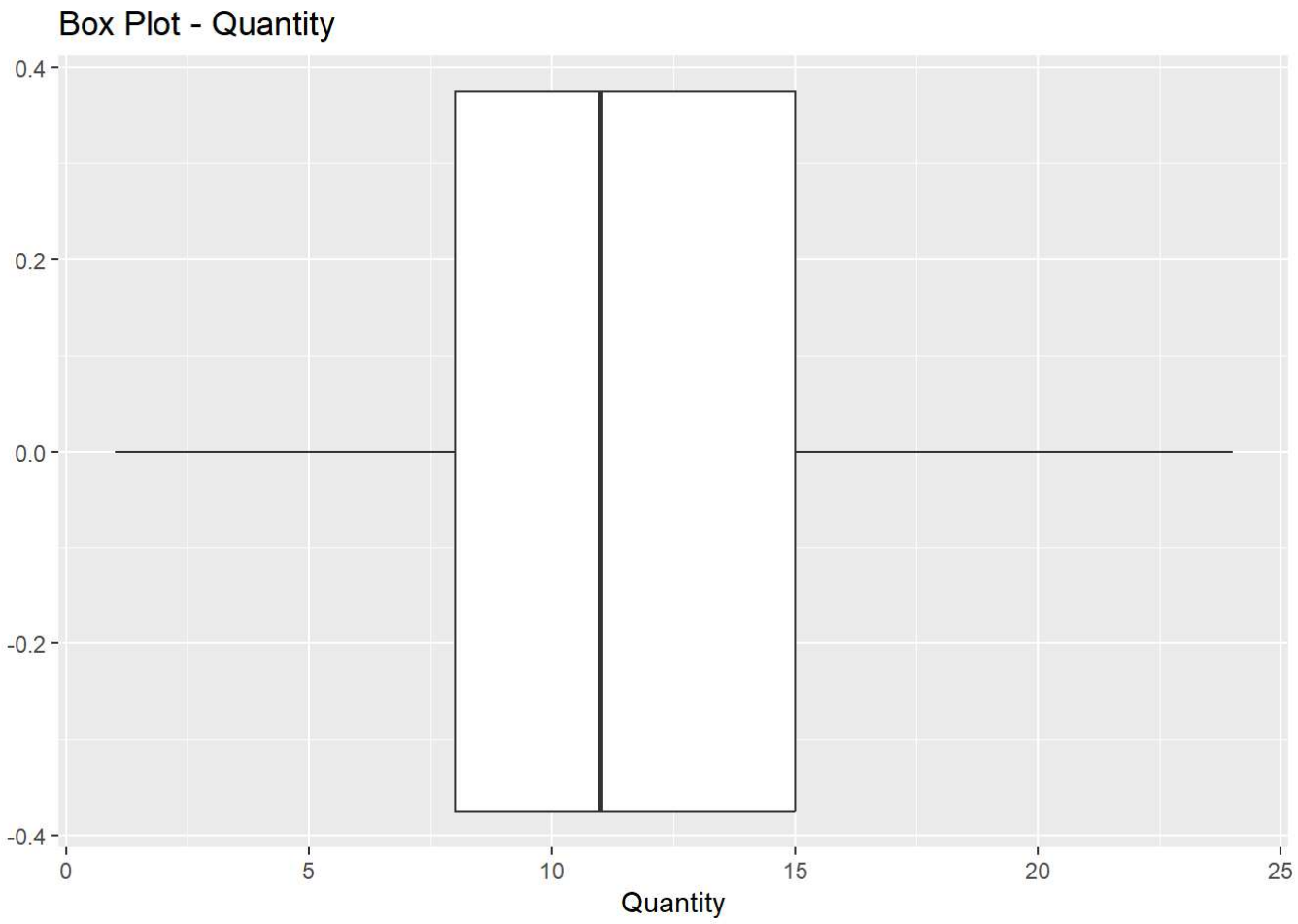

```
ggplot(df, aes(x=Price)) +
  geom_histogram(binwidth = 5) +
  coord_cartesian(xlim = c(0, 60), ylim = c(0, 10000)) +
  labs(title = 'Histogram - Price')
```

```
## Warning: Removed 10 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



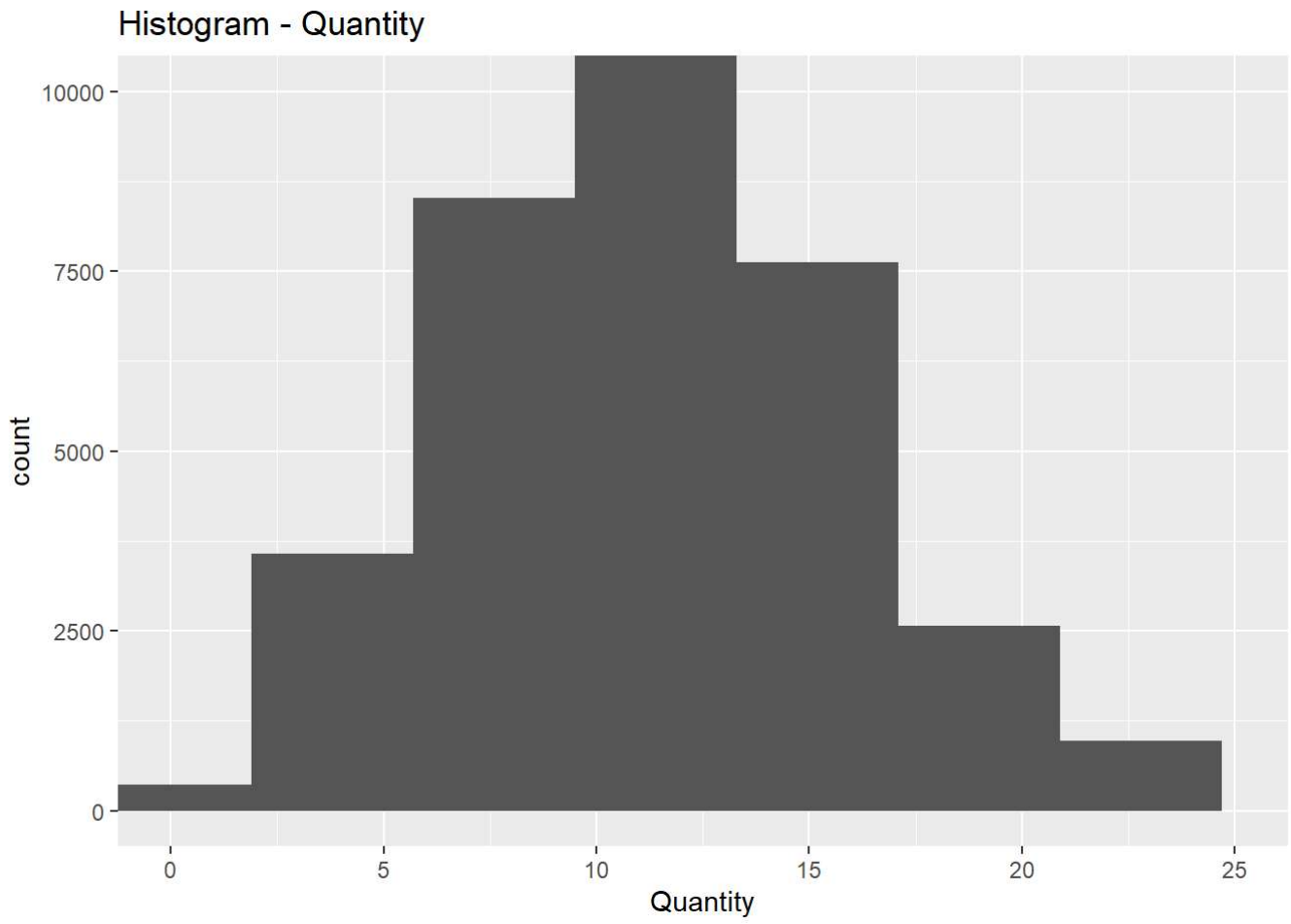

```
ggplot(df, aes(x=Quantity)) +
  geom_boxplot() +
  labs(title = 'Box Plot - Quantity')
```

```
## Warning: Removed 7 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```




```
ggplot(df, aes(x=Quantity)) +
  geom_histogram(binwidth = 3.8) +
  coord_cartesian(xlim = c(0, 25), ylim = c(0, 10000)) +
  labs(title = 'Histogram - Quantity')
```

```
## Warning: Removed 7 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



7. Write a short essay (150-200 words) to compare the strengths and weaknesses of (1) Power BI and (2) Alteryx with that of R, for this kind of analysis. You may discuss how each of these fare in terms of replicability, ease of use, cost, ability to share results with others, scalability, etc. 7 points

R is a powerful and flexible tool for data analysis and visualization. Compared to Power BI and Alteryx, it offers full control over statistical modeling and data manipulation with relatively ease code-based programming. Power BI: This tool excels in interactive dashboards and ease of use. It is great for business reporting but has limited statistical capabilities. It is not as scriptable or reproducible as R. Alteryx: Known for its drag-and-drop interface, it simplifies data preparation and ETL processes. However, it is expensive and less flexible than R for custom statistical modeling. R: Open-source and cost-effective, R provides deep statistical and visualization capabilities. However, it has a steeper learning curve compared to Power BI and Alteryx because of its code-based program. Overall, R is the best choice for replicability and advanced analysis, while Power BI and Alteryx are better for business users who prioritize ease of use and interactive reporting and business analysis.