Data Analysis Report

## Read the Data

file\_path <- '/Users/Sour/RClassTools/cleaned\_data\_R.csv'  
df <- read.csv(file\_path)

## Transform Columns

df <- df %>%  
 mutate(  
 CustomerCode = as.character(CustomerCode),  
 Category = as.factor(Category),  
 Date = as.Date(Date, format='%m/%d/%Y')  
 )  
str(df)

## 'data.frame': 34432 obs. of 7 variables:  
## $ Date : Date, format: NA NA ...  
## $ Department : chr "Kabobs" "Sides" "Sides" "Sides" ...  
## $ Category : Factor w/ 10 levels "Beef","Beef and Broccoli",..: 7 8 8 8 1 6 2 10 8 6 ...  
## $ CustomerCode: chr "CWM11331L8O" "CWM11331L8O" "CXP4593H7E" "CWM11331L8O" ...  
## $ Price : num 28 9 9 9 25 18 26 12 9 12 ...  
## $ Quantity : num 11 5 14 6 7 13 9 6 11 22 ...  
## $ Revenue : num 308 45 126 54 175 234 234 72 99 264 ...

## Summaries for Quantity and Price Columns

summary(df$Quantity)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 1.00 8.00 11.00 11.31 15.00 24.00 7

summary(df$Price)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 3.00 12.00 25.00 22.81 33.00 50.00 10

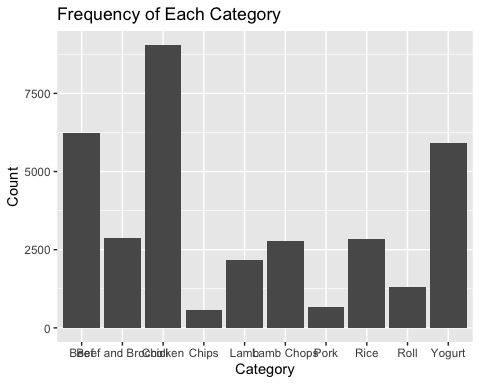
## Count of NA Values in Each Column

colSums(is.na(df))

## Date Department Category CustomerCode Price Quantity   
## 34432 0 0 0 10 7   
## Revenue   
## 17

## Bar Chart for Category Column

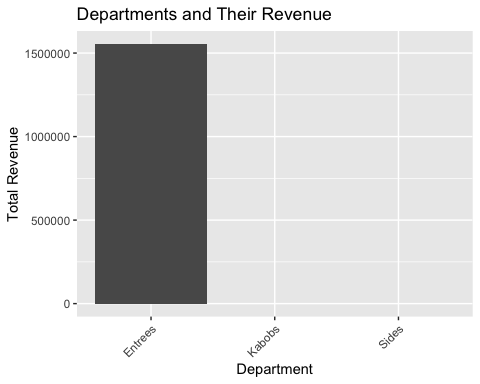
ggplot(df, aes(x = Category)) +  
 geom\_bar() +  
 labs(title = 'Frequency of Each Category', x = 'Category', y = 'Count')



## Departments and Their Revenue

df <- df %>%  
 mutate(Revenue = Price \* Quantity)  
department\_revenue <- df %>%  
 group\_by(Department) %>%  
 summarize(TotalRevenue = sum(Revenue)) %>%  
 arrange(desc(TotalRevenue))  
ggplot(department\_revenue, aes(x = reorder(Department, -TotalRevenue), y = TotalRevenue)) +  
 geom\_bar(stat = 'identity') +  
 labs(title = 'Departments and Their Revenue', x = 'Department', y = 'Total Revenue') +  
 theme(axis.text.x = element\_text(angle = 45, hjust = 1))

## Warning: Removed 2 rows containing missing values or values outside the scale range  
## (`geom\_bar()`).

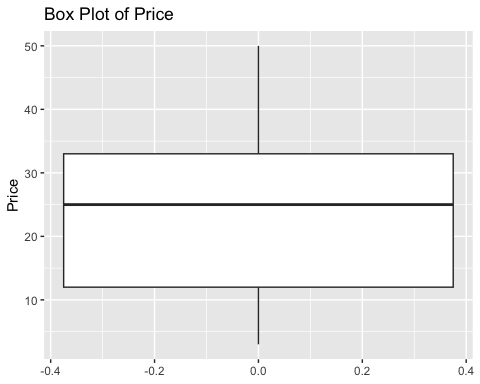


## Univariate Plots

### Box Plot of Price

ggplot(df, aes(y = Price)) +  
 geom\_boxplot() +  
 labs(title = 'Box Plot of Price', y = 'Price')

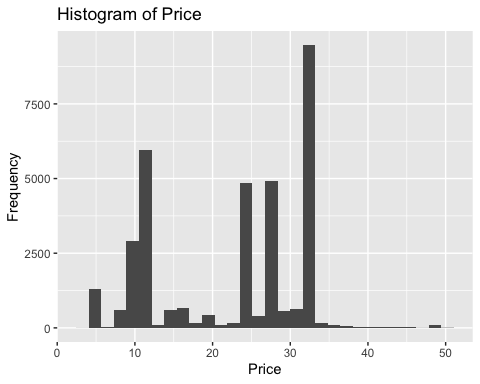
## Warning: Removed 10 rows containing non-finite outside the scale range  
## (`stat\_boxplot()`).



### Histogram of Price

ggplot(df, aes(x = Price)) +  
 geom\_histogram(bins = 30) +  
 labs(title = 'Histogram of Price', x = 'Price', y = 'Frequency')

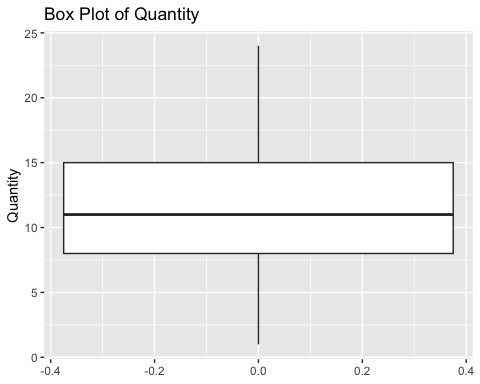
## Warning: Removed 10 rows containing non-finite outside the scale range  
## (`stat\_bin()`).



### Box Plot of Quantity

ggplot(df, aes(y = Quantity)) +  
 geom\_boxplot() +  
 labs(title = 'Box Plot of Quantity', y = 'Quantity')

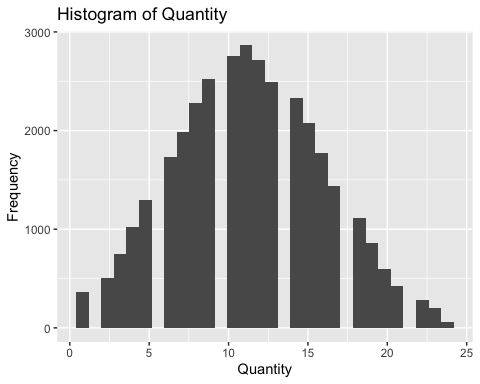
## Warning: Removed 7 rows containing non-finite outside the scale range  
## (`stat\_boxplot()`).



### Histogram of Quantity

ggplot(df, aes(x = Quantity)) +  
 geom\_histogram(bins = 30) +  
 labs(title = 'Histogram of Quantity', x = 'Quantity', y = 'Frequency')

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



## Essay: Comparison of Power BI, Alteryx, and R for Data Analysis

Power BI, Alteryx, and R each offer distinct advantages and drawbacks for data analysis. Power BI excels in data visualization and ease of use. Its intuitive interface and integration with Microsoft Office make it accessible to non-technical users. Power BI allows easy sharing of results through interactive dashboards and reports, enhancing collaborative efforts. However, it can be limited in handling very large datasets and may incur substantial costs for advanced features and extensive use.

Alteryx stands out for its powerful data preparation and blending capabilities. Its drag-and-drop workflow design simplifies complex data transformations and analytics, making it user-friendly for those without extensive coding knowledge. Alteryx also supports scalability with its ability to process large datasets efficiently. However, it is relatively expensive and might be less flexible than programming languages for customized analyses.

R, on the other hand, is a robust programming language renowned for its statistical analysis and data manipulation capabilities. It offers unmatched flexibility and is highly scalable, handling large datasets efficiently with packages like data.table and dplyr. R is open-source, making it cost-effective, but it requires a steeper learning curve. Sharing results can be challenging compared to Power BI’s dashboards, but tools like R Markdown and Shiny apps provide alternatives for reproducible research and interactive presentations.

In summary, Power BI is ideal for user-friendly visualization and collaboration, Alteryx for intuitive data preparation, and R for in-depth statistical analysis and scalability.