



Analysis of Super Market Sales

Chandi Rupasinghe

Data description

Context

The supermarket industry is experiencing significant growth in densely populated cities, accompanied by intense market competition. In this context, we have access to a data set containing historical sales data from a supermarket company, specifically from three different branches over a span of three months. This data set lends itself well to the application of predictive data analytics methods.

In the rapidly growing supermarket industry, R programming proves to be a valuable asset for analyzing historical sales data and gaining insights into market trends and competition. By leveraging R's data manipulation capabilities, I efficiently preprocess and clean the supermarket's historical sales dataset, ensuring its suitability for predictive analytics. R's extensive library ecosystem, including packages like dplyr and tidyr, enables to perform advanced data wrangling tasks such as aggregations, filtering, and reshaping, facilitating further analysis. Through the utilization of time series analysis techniques available in R, we can forecast future sales patterns, enabling the supermarket to make accurate predictions and adjust operations accordingly.

Utilizing R's visualization packages, such as ggplot2, I create visually appealing charts and graphs to effectively communicate sales trends, customer behavior, and other key insights. R's statistical analysis capabilities enable to conduct hypothesis testing and identify significant factors that influence sales performance across branches. By employing R's clustering algorithms, I can segment customers based on their purchasing patterns and demographics, helping the supermarket company tailor marketing strategies to different customer segments. Through sentiment analysis techniques available in R, I can analyze customer feedback data to assess satisfaction levels and identify areas for improvement, ultimately enhancing customer loyalty.

[Link to the Data Set \(Click here\)](#)

Attribute information

Variable	Description
Invoice id	Computer generated sales slip invoice identification number
Branch	Branch of supercenter (3 branches are available identified by A, B and C)
City	Location of supercenters
Customer type	Type of customers, recorded by Members for customers using member card and Normal for without member card
Gender	Gender type of customer
Product line	General item categorization groups - Electronic accessories, Fashion accessories, Food and beverages, Health and beauty, Home and lifestyle, Sports and travel
Unit price	Price of each product in \$
Quantity	Number of products purchased by customer
Tax	5% tax fee for customer buying
Total	Total price including tax
Date	Date of purchase (Record available from January 2019 to March 2019)
Time	Purchase time (10am to 9pm)
Payment	Payment used by customer for purchase (3 methods are available – Cash, Credit card and Ewallet)
COGS	Cost of goods sold
Gross margin percentage	Gross margin percentage
Gross income	Gross income
Rating	Customer stratification rating on their overall shopping experience (On a scale of 1 to 10)

Data Understanding

Dimension of the data set

```
data= supermarket_sales...Sheet1 <- read.csv("C:/Users/Chandi/Downloads/supermarket_sales - Sheet1.csv")
dim(data)

## [1] 1000 17
```

This data set has 1000 rows and 17 columns.

Column names of the data set

```
colnames(data)

## [1] "Invoice.ID"      "Branch"
## [3] "City"           "Customer.type"
## [5] "Gender"         "Product.line"
## [7] "Unit.price"     "Quantity"
## [9] "Tax.5."        "Date"
## [11] "Total"         "Time"
## [13] "Payment"       "cogs"
## [15] "gross.margin.percentage" "gross.income"
## [17] "Rating"
```

Rename columns

```
library(dplyr)

## Warning: package 'dplyr' was built under R version 4.3.1

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

data= rename(data, total.price = Total)
data= rename(data, cost.of.goods = cogs)
data= rename(data, tax = Tax.5.)
data= rename(data, date = Date)
colnames(data)

## [1] "Invoice.ID"      "Branch"
## [3] "City"           "Customer.type"
## [5] "Gender"         "Product.line"
```

```
## [7] "Unit.price"      "Quantity"
## [9] "tax"             "date"
## [11] "total.price"     "Time"
## [13] "Payment"         "cost.of.goods"
## [15] "gross.margin.percentage" "gross.income"
## [17] "Rating"
```

Structure of the data set

```
str(data)

## 'data.frame':  1000 obs. of  17 variables:
## $ Invoice.ID      : chr  "750-67-8428" "226-31-3081" "631-41-3108" "123-19-1176" ...
## $ Branch         : chr  "A" "C" "A" "A" ...
## $ City           : chr  "Yangon" "Naypyitaw" "Yangon" "Yangon" ...
## $ Customer.type   : chr  "Member" "Normal" "Normal" "Member" ...
## $ Gender         : chr  "Female" "Female" "Male" "Male" ...
## $ Product.line    : chr  "Health and beauty" "Electronic accessories" "Home and lifestyle"
## "Health and beauty" ...
## $ Unit.price      : num  74.7 15.3 46.3 58.2 86.3 ...
## $ Quantity        : int   7 5 7 8 7 7 6 10 2 3 ...
## $ tax             : num  26.14 3.82 16.22 23.29 30.21 ...
## $ date            : chr   "01-05-2019" "03-08-2019" "03-03-2019" "1/27/2019" ...
## $ total.price     : num  549 80.2 340.5 489 634.4 ...
## $ Time            : chr   "13:08" "10:29" "13:23" "20:33" ...
## $ Payment         : chr  "Ewallet" "Cash" "Credit card" "Ewallet" ...
## $ cost.of.goods    : num  522.8 76.4 324.3 465.8 604.2 ...
## $ gross.margin.percentage: num   4.76 4.76 4.76 4.76 4.76 ...
## $ gross.income     : num  26.14 3.82 16.22 23.29 30.21 ...
## $ Rating          : num   9.1 9.6 7.4 8.4 5.3 4.1 5.8 8 7.2 5.9 ...
```

Change the classes of selected variables

```
character_columns <- c("Branch", "Customer.type", "Gender", "Product.line", "Payment")

data <- data %>%
  mutate(across(all_of(character_columns), as.factor))
str(data)

## 'data.frame':  1000 obs. of  17 variables:
## $ Invoice.ID      : chr  "750-67-8428" "226-31-3081" "631-41-3108" "123-19-1176" ...
## $ Branch         : Factor w/ 3 levels "A","B","C": 1 3 1 1 1 3 1 3 1 2 ...
## $ City           : chr  "Yangon" "Naypyitaw" "Yangon" "Yangon" ...
## $ Customer.type   : Factor w/ 2 levels "Member","Normal": 1 2 2 1 2 2 1 2 1 1 ...
## $ Gender         : Factor w/ 2 levels "Female","Male": 1 1 2 2 2 2 1 1 1 1 ...
## $ Product.line    : Factor w/ 6 levels "Electronic accessories",...: 4 1 5 4 6 1 1 5 4 3 ...
## $ Unit.price      : num  74.7 15.3 46.3 58.2 86.3 ...
## $ Quantity        : int   7 5 7 8 7 7 6 10 2 3 ...
## $ tax             : num  26.14 3.82 16.22 23.29 30.21 ...
## $ date            : chr   "01-05-2019" "03-08-2019" "03-03-2019" "1/27/2019" ...
```

```
## $ total.price      : num  549 80.2 340.5 489 634.4 ...
## $ Time            : chr  "13:08" "10:29" "13:23" "20:33" ...
## $ Payment         : Factor w/ 3 levels "Cash","Credit card",...: 3 1 2 3 3 3 3 3 2 2 ...
## $ cost.of.goods    : num  522.8 76.4 324.3 465.8 604.2 ...
## $ gross.margin.percentage: num  4.76 4.76 4.76 4.76 4.76 ...
## $ gross.income     : num  26.14 3.82 16.22 23.29 30.21 ...
## $ Rating           : num  9.1 9.6 7.4 8.4 5.3 4.1 5.8 8 7.2 5.9 ...
```

In this dataset, several variables have been converted from character variables to factor variables. The variables that have been modified are as follows: "Branch", "Customer.type", "Gender", "Product.line", and "Payment".

Summarize the dataset

```
summary(data)
```

```
## Invoice.ID      Branch      City      Customer.type  Gender
## Length:1000    A:340      Length:1000    Member:501    Female:501
## Class :character B:332      Class :character Normal:499    Male :499
## Mode :character C:328      Mode :character
##
##
##
##      Product.line Unit.price   Quantity      tax
## Electronic accessories:170 Min. :10.08 Min. : 1.00 Min. : 0.5085
## Fashion accessories :178 1st Qu.:32.88 1st Qu.: 3.00 1st Qu.: 5.9249
## Food and beverages :174 Median :55.23 Median : 5.00 Median :12.0880
## Health and beauty :152 Mean :55.67 Mean : 5.51 Mean :15.3794
## Home and lifestyle :160 3rd Qu.:77.94 3rd Qu.: 8.00 3rd Qu.:22.4453
## Sports and travel :166 Max. :99.96 Max. :10.00 Max. :49.6500
## date      total.price      Time      Payment
## Length:1000 Min. : 10.68 Length:1000 Cash :344
## Class :character 1st Qu.: 124.42 Class :character Credit card:311
## Mode :character Median : 253.85 Mode :character Ewallet :345
##      Mean : 322.97
##      3rd Qu.: 471.35
##      Max. :1042.65
## cost.of.goods gross.margin.percentage gross.income      Rating
## Min. : 10.17 Min. :4.762 Min. : 0.5085 Min. : 4.000
## 1st Qu.:118.50 1st Qu.:4.762 1st Qu.: 5.9249 1st Qu.: 5.500
## Median :241.76 Median :4.762 Median :12.0880 Median : 7.000
## Mean :307.59 Mean :4.762 Mean :15.3794 Mean : 6.973
## 3rd Qu.:448.90 3rd Qu.:4.762 3rd Qu.:22.4453 3rd Qu.: 8.500
## Max. :993.00 Max. :4.762 Max. :49.6500 Max. :10.000
```

The given data set represents information related to invoices and transactions. It includes various variables such as dates, invoice IDs, branch, city, customer type, gender, product line, unit price,

quantity, tax, total price, payment method, cost of goods, gross margin percentage, gross income, and ratings.

The data set consists of 1,000 records, with invoices spanning from January 1, 2019, to March 30, 2019. The branches are labeled as A, B, and C, with the distribution of invoices among them being 340, 332, and 328, respectively. The data set covers multiple cities.

Regarding customer details, there are two customer types: "Member" and "Normal." The data set also includes information about the gender of customers, with an almost equal distribution between males and females.

The product line column provides insights into the types of products sold, with categories such as electronic accessories, fashion accessories, food and beverages, health and beauty, home and lifestyle, and sports and travel. The unit price of products ranges from \$10.08 to \$99.96.

Quantity indicates the number of items purchased, with values ranging from 1 to 10. The tax applied to each transaction ranges from 0.5085 to 49.6500, while the total price varies from \$10.68 to \$1042.65.

Payment methods include cash, credit card, and e-wallet, with respective frequencies of 344, 311, and 345 transactions. The cost of goods ranges from \$10.17 to \$993.00, and the gross margin percentage remains constant at 4.762% for all transactions. Gross income, representing the profit earned, ranges from 0.5085 to 49.6500.

Finally, the data set includes a rating column, indicating customer satisfaction. Ratings range from 4.000 to 10.000, with an average rating of approximately 6.973.

In summary, the data set provides detailed information about invoices, transactions, customer types, product categories, prices, quantities, taxes, payment methods, costs, margins, incomes, and customer ratings.

Data Preparation

Check for missing values.

```
sum(is.na(data))
```

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

The presented data set exhibits a remarkable characteristic as it contains no missing values. Each observation within the data set possesses complete information for all variables. This attribute ensures the reliability and integrity of the data, allowing for comprehensive analyses and accurate interpretations.

Check for duplicate values.

```
sum(duplicated(data))
```

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

The data set at hand showcases a notable characteristic whereby it does not contain any duplicate values. Each entry within the data set is unique and distinct, ensuring the absence of redundant or replicated observations. This characteristic enhances the reliability and accuracy of the data, as each value represents a distinct entity or event. The absence of duplicates simplifies data analysis and interpretation, as there is no need to account for or handle multiple occurrences of the same value.

Convert variables to their correct data types:

```
library(lubridate)
```

```
data$date = parse_date_time(data$date, orders = c("mdy", "dmy", "ymd"))
```

```
data$date = format(data$date, "%Y-%m-%d")
```

```
cleaned_dates = data$date
```

```
data$date = cleaned_dates
```

```
head(data$date, 5)
```

```
## [1] "2019-01-05" "2019-03-08" "2019-03-03" "2019-01-27" "2019-02-08"
```

The given data set includes a date column, which is a valuable component for time series analysis. However, initially, the dates in the data set were not in a consistent format. Fortunately, all the dates in the data set have now been standardized to adhere to the same date format.

Moving Date into first column

```
data <- data %>%
```

```
  relocate(date, .before = Invoice.ID)
```

```
head(data, 3)
```

```
##      date Invoice.ID Branch  City Customer.type Gender
## 1 2019-01-05 750-67-8428   A   Yangon      Member Female
## 2 2019-03-08 226-31-3081   C Naypyitaw    Normal Female
## 3 2019-03-03 631-41-3108   A   Yangon      Normal  Male
##      Product.line Unit.price Quantity  tax total.price Time
## 1 Health and beauty    74.69      7 26.1415  548.9715 13:08
## 2 Electronic accessories    15.28     5  3.8200  80.2200 10:29
## 3 Home and lifestyle    46.33     7 16.2155  340.5255 13:23
##      Payment cost.of.goods gross.margin.percentage gross.income Rating
## 1 Ewallet      522.83      4.761905    26.1415  9.1
## 2 Cash         76.40      4.761905     3.8200  9.6
## 3 Credit card   324.31      4.761905    16.2155  7.4
```

Moving the date column to the index enhances the overall data understanding and enables more effective analysis of the temporal aspects present in the data set. It provides a solid foundation for conducting comprehensive time series analysis and extracting valuable insights related to the behavior and dynamics of the observed phenomena over time.

Creating an Informative Table of Unique Values for factor variables in R

```
## Unique values for different variables
branch_values <- unique(data$Branch)
customer_type_values <- unique(data$Customer.type)
city_values <- unique(data$City)
payment_values <- unique(data$Payment)
product_line_values <- unique(data$Product.line)
# Create a data frame for the table

# Create a data frame for the table
unique_values_table <- data.frame(
  Variable = c("Branch", "Customer Type", "City", "Payment", "Product Line"),
  Unique_Values = supply(list(branch_values, customer_type_values, city_values,
    payment_values, product_line_values), paste, collapse = ", ")
)

print(unique_values_table)
```

RStudio: Notebook Output

Description: df [5 × 2]

Variable <chr>	Unique_Values <chr>
Branch	A, C, B
Customer Type	Member, Normal
City	Yangon, Naypyitaw, Mandalay
Payment	Ewallet, Cash, Credit card
Product Line	Health and beauty, Electronic accessories, Home and lifestyle, Sports and travel, Food and beverages, Fashion accessories

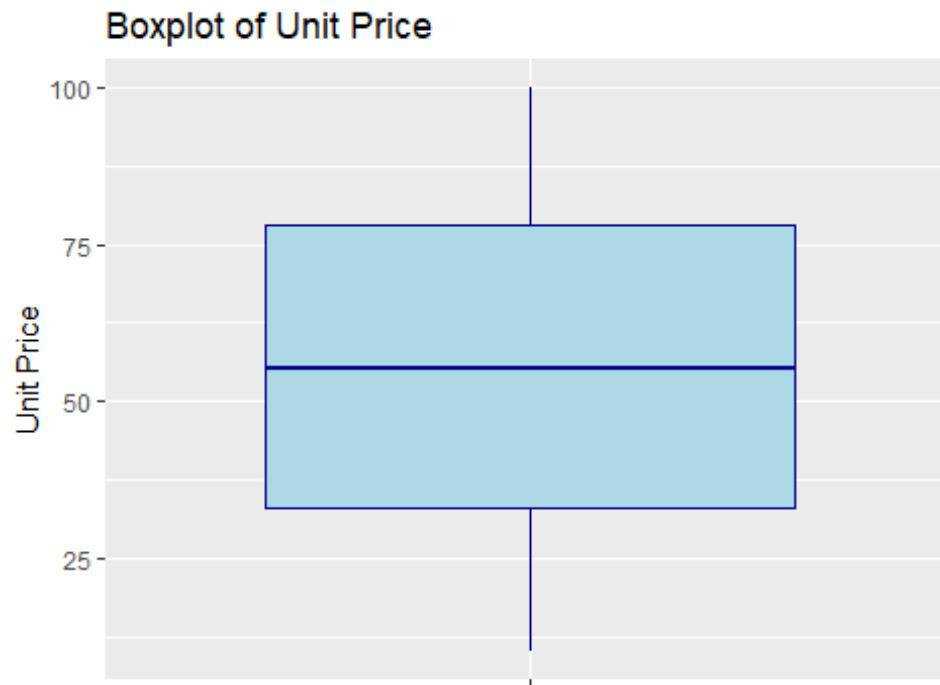
5 rows

Identify outliers using boxplot

Identifying outliers using a box plot is a powerful technique that helps in detecting and understanding the presence of extreme values in a data set. Outliers are data points that significantly differ from the majority of the observations and can have a significant impact on statistical analysis, modeling, and decision-making processes. By utilizing a box plot, analysts can gain valuable insights into the data distribution and identify potential outliers for further investigation.

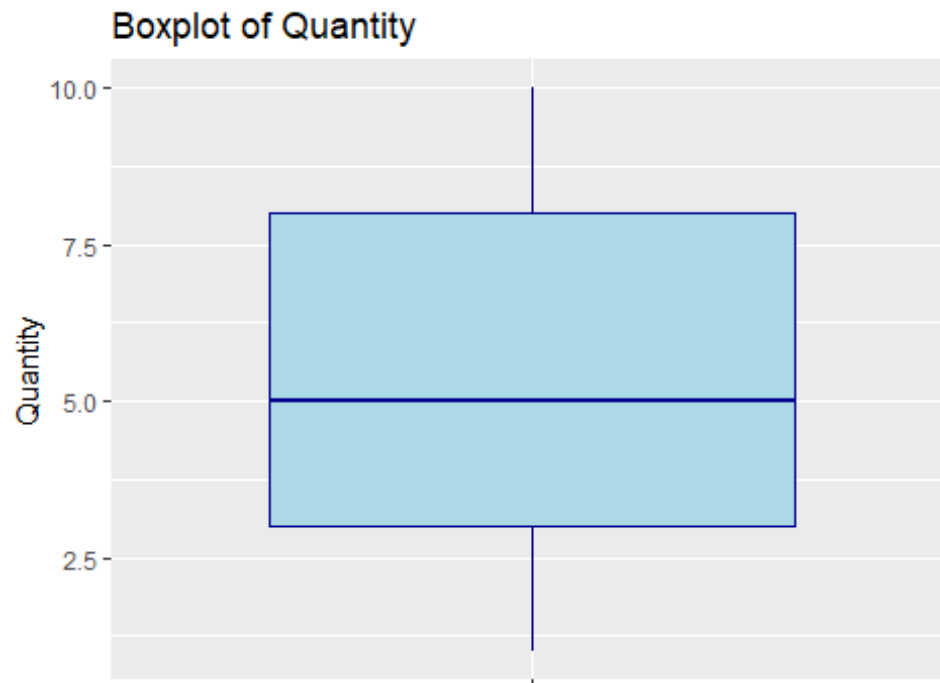
```
library(ggplot2)

ggplot(data, aes(x = "", y = Unit.price)) +
  geom_boxplot(fill = "lightblue", color = "darkblue", outlier.color = "red") +
  labs(title = "Boxplot of Unit Price",
       x = "",
       y = "Unit Price")
```

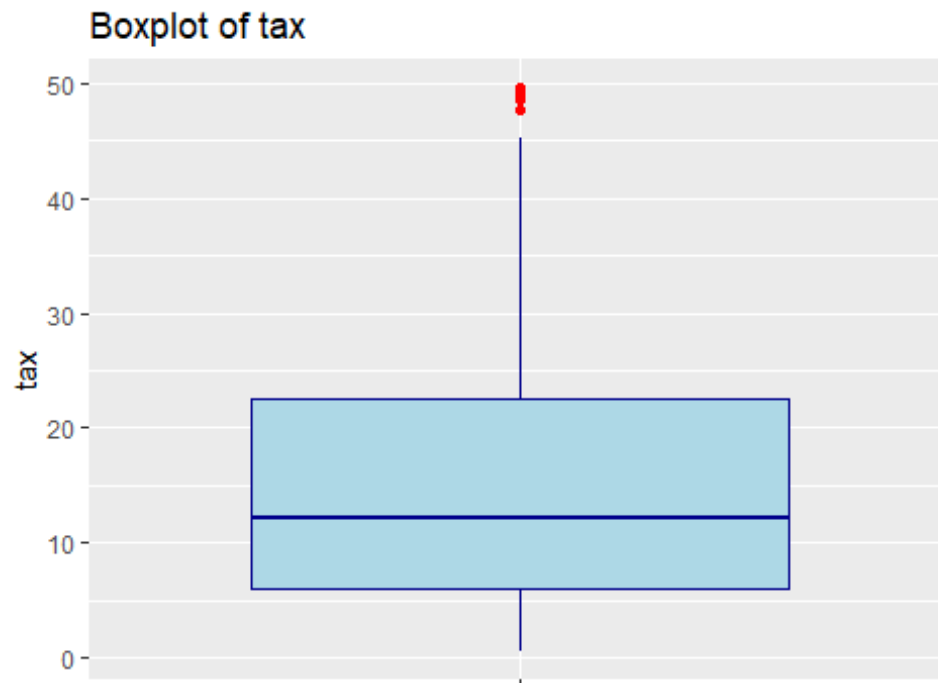
```
library(ggplot2)

# Create a boxplot of Unit.price
ggplot(data, aes(x = "", y = Quantity)) +
  geom_boxplot(fill = "lightblue", color = "darkblue", outlier.color = "red") +
  labs(title = "Boxplot of Quantity",
       x = "",
       y = "Quantity")
```



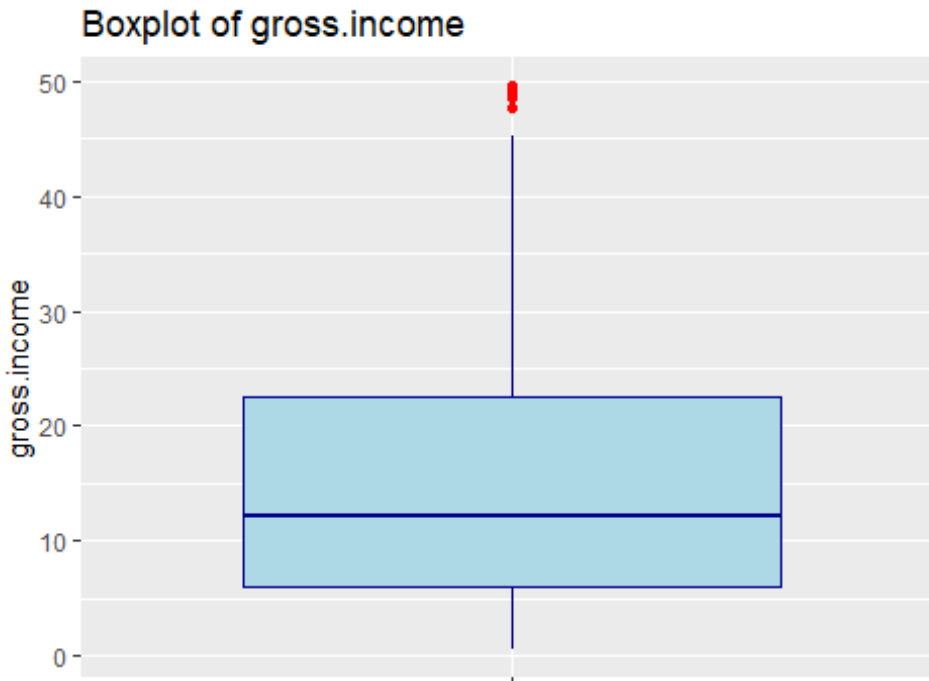
```
library(ggplot2)

# Create a boxplot of Unit.price
ggplot(data, aes(x = "", y = tax)) +
  geom_boxplot(fill = "lightblue", color = "darkblue", outlier.color = "red") +
  labs(title = "Boxplot of tax",
       x = "",
       y = "tax")
```



```
library(ggplot2)

# Create a boxplot of Unit.price
ggplot(data, aes(x = "", y = gross.income)) +
  geom_boxplot(fill = "lightblue", color = "darkblue", outlier.color = "red") +
  labs(title = "Boxplot of gross.income",
       x = "",
       y = "gross.income")
```



Note:

Upon analyzing the data set using a box plot, it was observed that the variables "tax" and "gross income" contain outliers. Outliers are data points that deviate significantly from the majority of observations and exhibit extreme values in relation to the rest of the data set.

The presence of outliers in the "tax" and "gross income" variables can have important implications for data analysis and decision-making processes. These outliers may arise due to various factors, such as data entry errors, measurement inaccuracies, or genuine extreme values in the underlying phenomenon being measured.

Exploratory Data Analysis

The data presents the counts of various categories within different variables in the data set. These counts provide valuable insights into the distribution and prevalence of each category.

The "Branch" variable indicates three different branches, namely Branch A, Branch B, and Branch C. The counts reveal that Branch A has the highest count with 340 occurrences, followed closely by Branch B with 332 occurrences, and Branch C with 328 occurrences. This information highlights the varying representation of branches within the data set.

Moving on to the "Customer Type" variable, it can be observed that there are two categories: "Member" and "Normal." The counts indicate that there are 501 occurrences of customers categorized as "Member" and 499 occurrences of customers categorized as "Normal." This suggests a relatively balanced distribution of customer types in the data set.

The "Payment" variable provides insights into the different modes of payment used by customers. The counts show that cash is the most frequently used payment method with 344 occurrences, followed by credit card with 311 occurrences, and e-wallet with 345 occurrences. These counts shed light on the popularity and adoption of different payment methods among customers.

Lastly, the "Product Line" variable represents various product categories. The counts reveal the distribution of products across different lines. The highest count is observed in the "Fashion accessories" category with 178 occurrences, followed by "Food and beverages" with 174 occurrences, "Sports and travel" with 166 occurrences, "Home and lifestyle" with 160 occurrences, "Electronic accessories" with 170 occurrences, and "Health and beauty" with 152 occurrences. These counts provide valuable information about the popularity and demand for different product lines within the data set.

In summary, the provided counts offer valuable insights into the distribution and occurrence of different categories within the "Branch," "Customer Type," "Payment," and "Product Line" variables. This information can be further utilized to understand patterns, preferences, and trends within the data set.

```
# Table for Product Line
product_table <- table(data$Product.line)
product_table <- as.data.frame(product_table)
colnames(product_table) <- c("Product Line", "Count")
kable(product_table, format = "markdown", align = c("l", "r"))
```

Product Line	Count
Electronic accessories	170
Fashion accessories	178
Food and beverages	174
Health and beauty	152
Home and lifestyle	160
Sports and travel	166

```
# Pie Chart for Product Line
```

```
library(ggplot2)
```

```
product_pie <- ggplot(product_table, aes(x = "", y = Count, fill = `Product Line`)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar("y", start = 0) +
  labs(title = "Distribution of Product Line") +
```

```
  theme_void()
```

```
product_pie
```

Distribution of Product Line

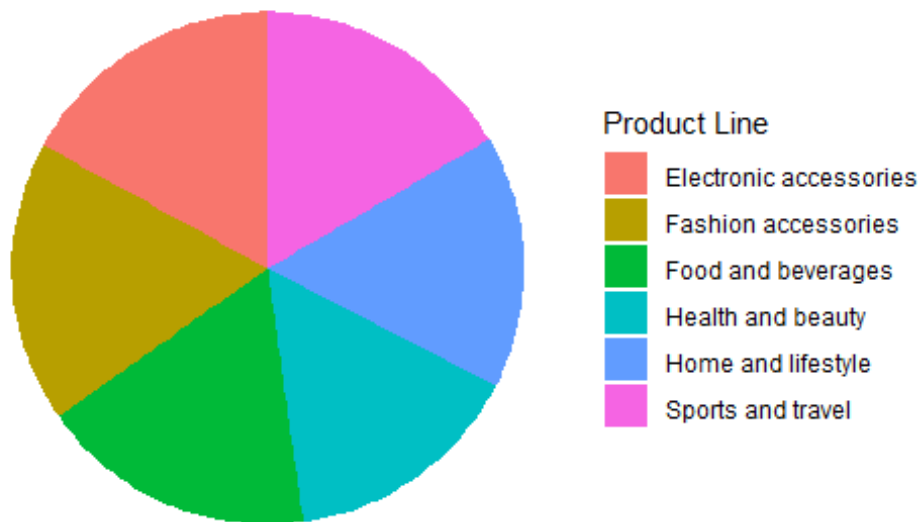


Table for Branch

```
branch_table <- table(data$Branch)
branch_table <- as.data.frame(branch_table)
colnames(branch_table) <- c("Branch", "Count")
kable(branch_table, format = "markdown", align = c("l", "r"))
```

Branch	Count
A	340
B	332
C	328

Pie Chart for Branch

```
library(ggplot2)

branch_pie <- ggplot(branch_table, aes(x = "", y = Count, fill = Branch)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar("y", start = 0) +
  labs(title = "Distribution of Branch") +

  theme_void()

branch_pie
```

Distribution of Branch

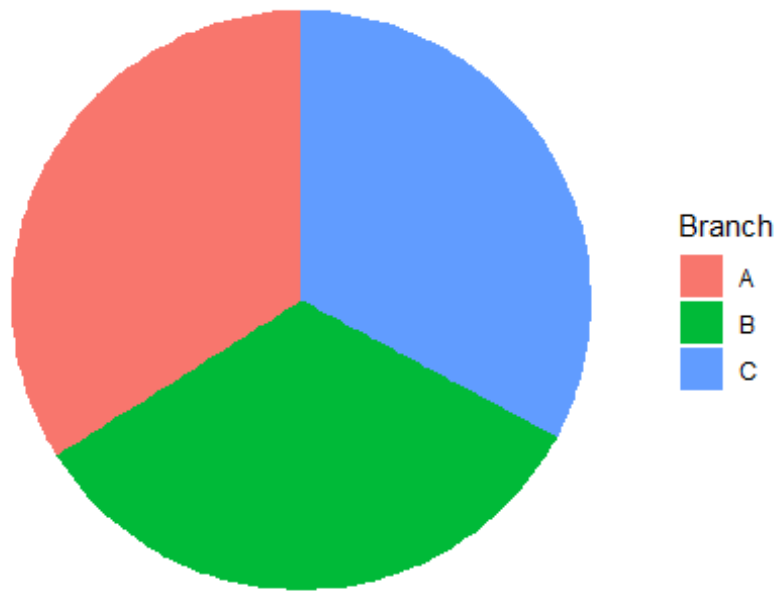


Table for Customer Type

```
customer_table <- table(data$Customer.type)
customer_table <- as.data.frame(customer_table)
colnames(customer_table) <- c("Customer_Type", "Count")
kable(customer_table, format = "markdown", align = c("l", "r"))
```

Customer_Type	Count
Member	501
Normal	499

Pie Chart for Customer Type

```
library(ggplot2)

customer_pie <- ggplot(customer_table, aes(x = "", y = Count, fill = Customer_Type)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar("y", start = 0) +
  labs(title = "Distribution of Customer Type") +

  theme_void()

customer_pie
```

Distribution of Customer Type

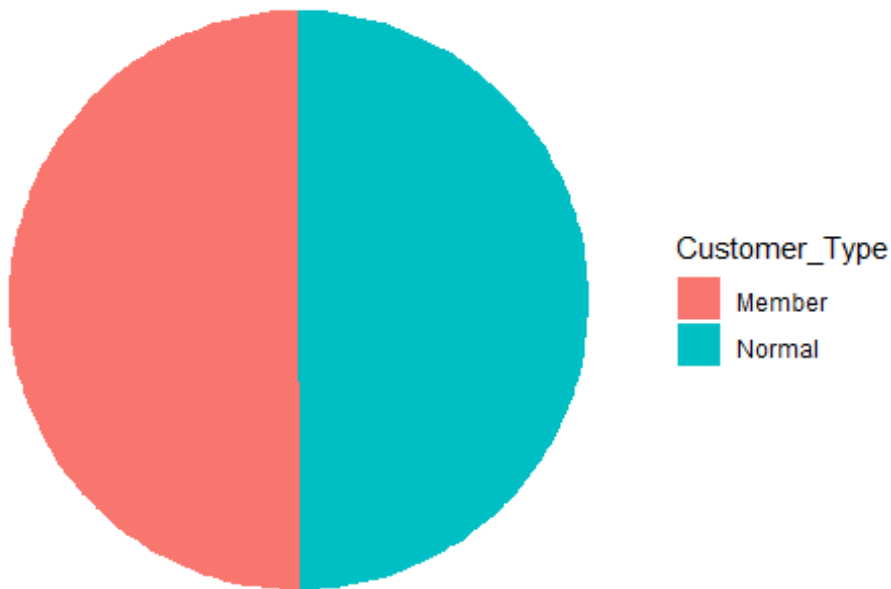


Table for Payment

```
payment_table <- table(data$Payment)
payment_table <- as.data.frame(payment_table)
colnames(payment_table) <- c("Payment", "Count")
kable(payment_table, format = "markdown", align = c("l", "r"))
```

Payment	Count
Cash	344
Credit card	311
Ewallet	345

Pie Chart for Payment

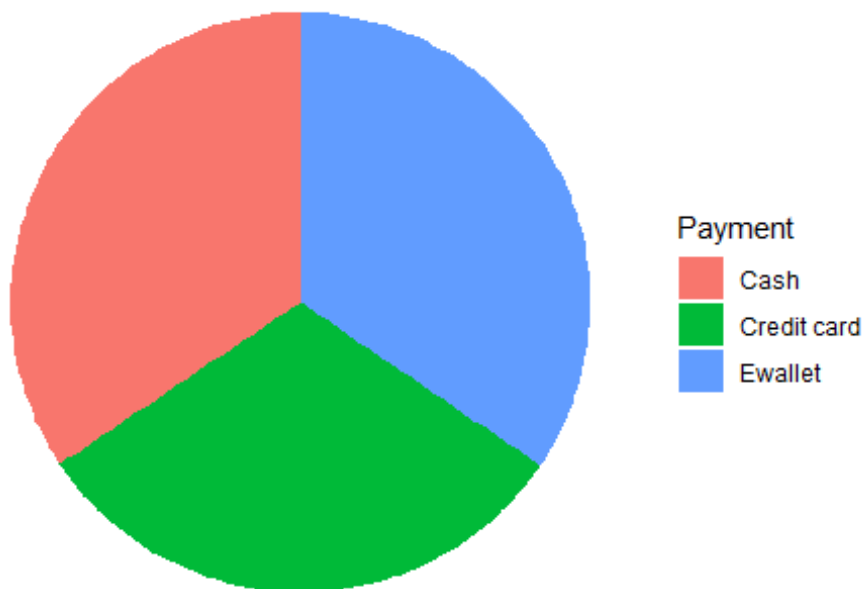
```
library(ggplot2)

payment_pie <- ggplot(payment_table, aes(x = "", y = Count, fill = Payment)) +
  geom_bar(stat = "identity", width = 1) +
  coord_polar("y", start = 0) +
  labs(title = "Distribution of Payment") +

  theme_void()

payment_pie
```


Distribution of Payment



Bar plot with colored bars

```
bar_plot <- ggplot(data, aes(x = Branch, y = Quantity, fill = Branch)) +  
  geom_bar(stat = "identity") +  
  labs(title = "Quantity by Branch", x = "Branch", y = "Quantity") +  
  theme_minimal() +  
  theme(plot.title = element_text(size = 16, face = "bold"),  
        legend.position = "none",  
        panel.grid.major = element_blank(),  
        panel.grid.minor = element_blank(),  
        panel.background = element_rect(fill = "white"))
```

Boxplot with colored boxes

```
boxplot1 <- ggplot(data, aes(x = Product.line, y = total.price, fill = Product.line)) +  
  geom_boxplot() +  
  labs(title = "Boxplot of Total Price by Product Line", x = "Product Line", y = "Total Price") +  
  theme_minimal() +  
  theme(plot.title = element_text(size = 16, face = "bold"),  
        legend.position = "none",  
        panel.grid.major = element_blank(),  
        panel.grid.minor = element_blank(),  
        panel.background = element_rect(fill = "white"))
```

Boxplot with colored boxes

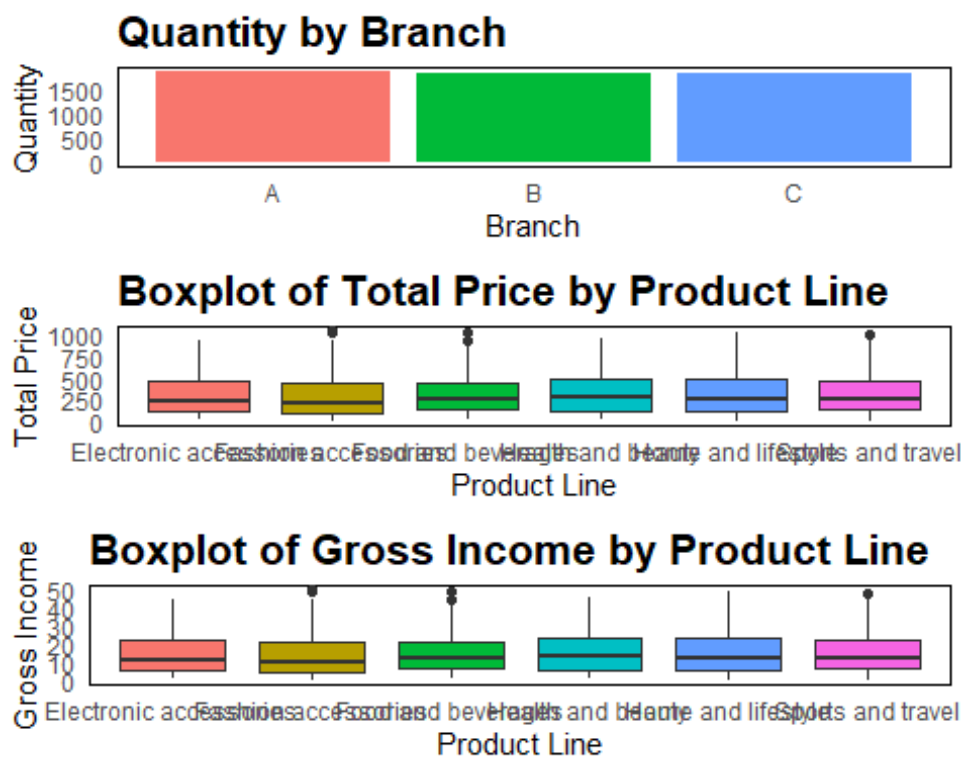
```
boxplot2 <- ggplot(data, aes(x = Product.line, y = gross.income, fill = Product.line)) +  
  geom_boxplot() +
```

```

labs(title = "Boxplot of Gross Income by Product Line", x = "Product Line", y = "Gross
Income") +
theme_minimal() +
theme(plot.title = element_text(size = 16, face = "bold"),
      legend.position = "none",
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      panel.background = element_rect(fill = "white"))

# Combine plots in a grid
grid.arrange(bar_plot, boxplot1, boxplot2, nrow = 3)

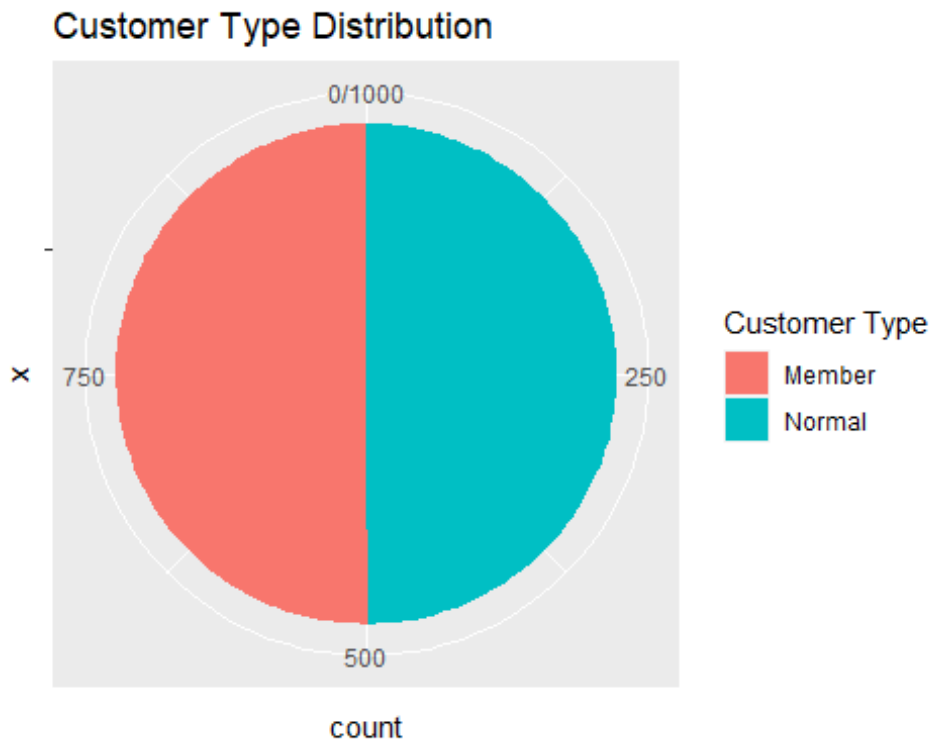
```



```

ggplot(data, aes(x = "", fill = Customer.type)) +
geom_bar(width = 1) +
coord_polar("y", start = 0) +
labs(fill = "Customer Type") +
ggtitle("Customer Type Distribution")

```



Analyzing Customer Rating Distribution

This histogram visualizes the distribution of the "Rating" variable in the data set. The x-axis represents the rating values, while the y-axis represents the count of observations corresponding to each rating value.

The histogram is constructed with bars representing different rating intervals. Each bar's height represents the frequency or count of occurrences of ratings falling within that particular interval. The fill color of each bar is determined by the rating value, with different colors indicating different rating levels.

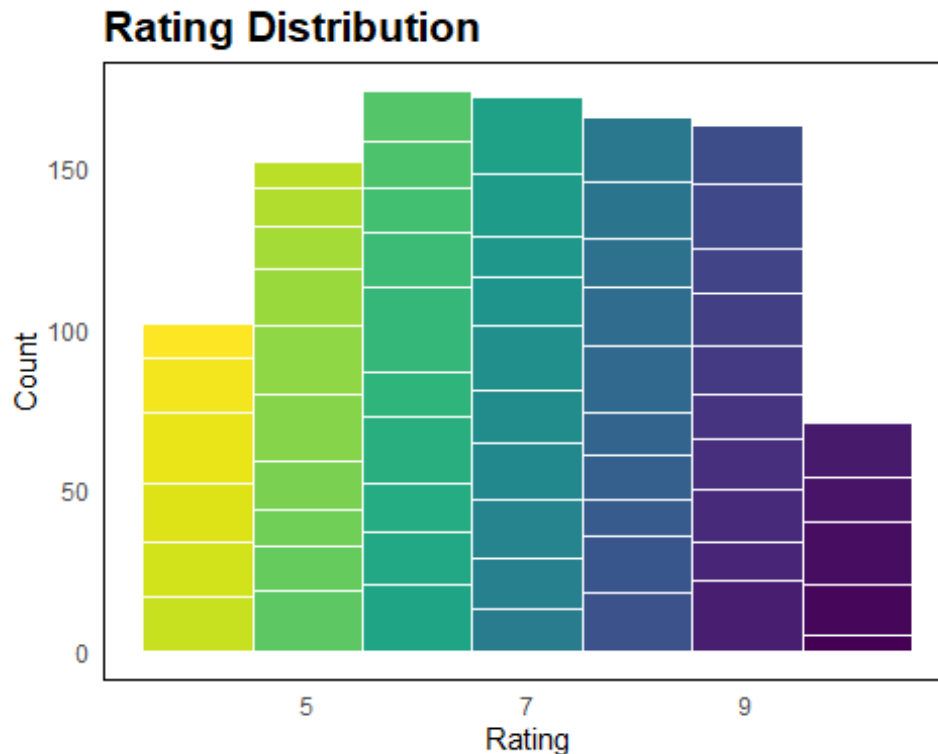
The provided information represents the distribution of ratings in the dataset. Each row corresponds to a specific rating value, and the "Count" column indicates the number of occurrences of that rating value in the dataset.

Breakdown of the information:

- Ratings range from 4 to 10, with increments of 0.1.
- The count of ratings varies for each rating value.
- The lowest count is 5, observed for the rating value of 10.
- The highest count is 26, observed for the rating value of 6.

```
ggplot(data, aes(x = Rating, fill = factor(Rating))) +
  geom_histogram(binwidth = 1, color = "white") +
  scale_fill_viridis_d(option = "D", direction = -1) +
```

```
labs(x = "Rating", y = "Count") +
ggtitle("Rating Distribution") +
theme_minimal() +
theme(plot.title = element_text(size = 16, face = "bold"),
      legend.position = "none",
      panel.grid.major = element_blank(),
      panel.grid.minor = element_blank(),
      panel.background = element_rect(fill = "white"))
```



Unveiling Product Line Distribution by Gender

There are 84 female customers who purchased electronic accessories. Similarly, there are 86 male customers who purchased electronic accessories, resulting in a total count of 170 for the electronic accessories product line. The same pattern applies to other gender and product line combinations.

The data concludes with the row and column totals, indicating that there are a total of 501 female customers, 499 male customers, and a grand total of 1000 customers across all categories.

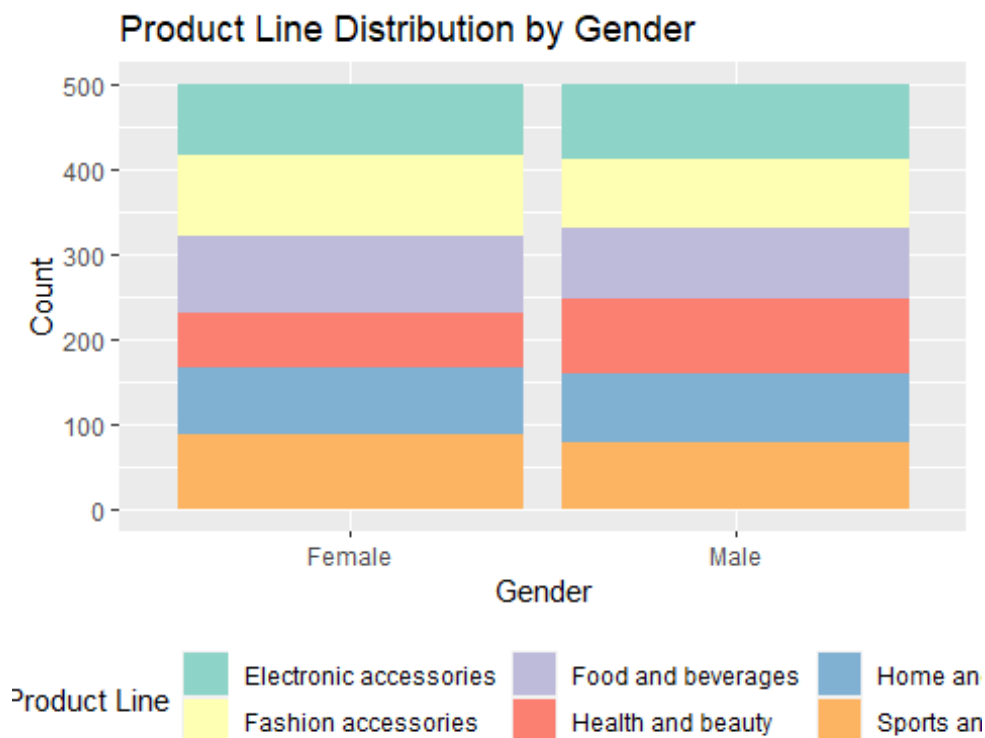
Overall, the data provides an overview of the distribution of customers based on gender and their preference for different product lines.

Product Line Distribution by Gender

The bar chart illustrates the distribution of product lines among different genders in the dataset. Each bar represents a specific product line, and the height of the bar indicates the count or

frequency of that product line. The bars are grouped by gender, allowing for a visual comparison of the distribution across different genders.

```
ggplot(data, aes(x = Gender, fill = Product.line)) +  
  geom_bar() +  
  labs(x = "Gender", y = "Count", fill = "Product Line") +  
  ggtitle("Product Line Distribution by Gender") +  
  theme(legend.position = "bottom") +  
  scale_fill_brewer(palette = "Set3")
```

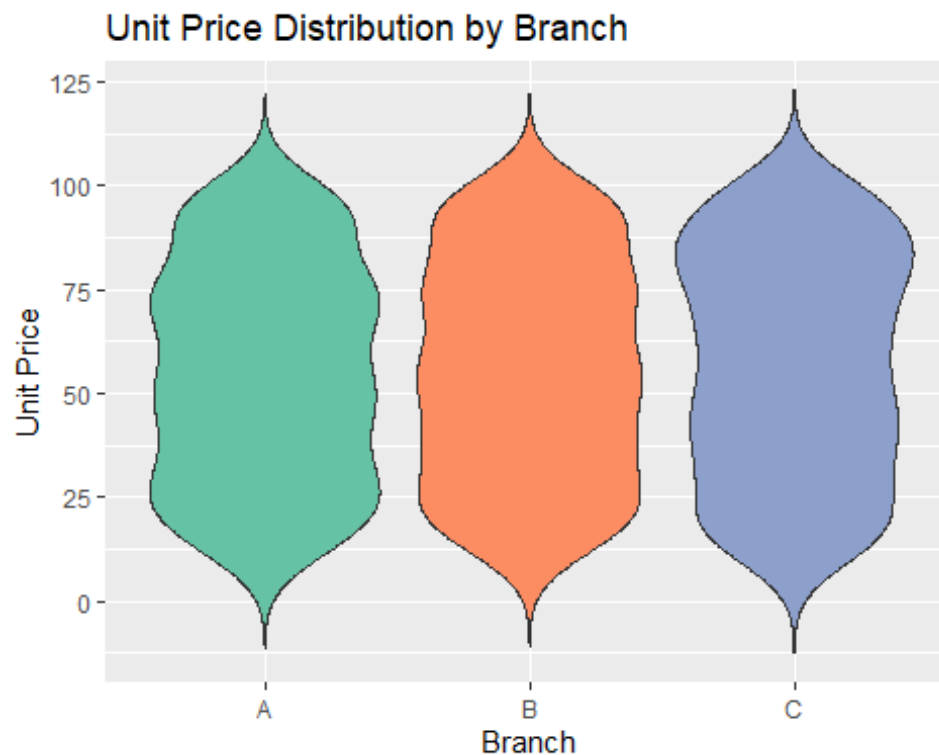


Unraveling the Distribution of Ratings by Product Line

The highest median rating among the product lines is observed for the "Food and beverages" category with a rating of 7.30, indicating a generally positive reception from customers. The "Health and beauty" product line has the lowest minimum rating of 4.01, suggesting some variations in customer opinions within this category. The "Fashion accessories" category has the highest count of ratings with a total of 178, indicating a significant level of customer engagement and feedback. The "Electronic accessories" and "Health and beauty" categories have the widest range of ratings, ranging from a minimum of 4.01 to a maximum of 10.0, indicating diverse customer experiences within these product lines.

Overall, the ratings for all product lines tend to be above average, with median ratings ranging from 6.70 to 7.30, suggesting positive customer satisfaction across different categories.

```
ggplot(data, aes(x = Branch, y = Unit.price, fill = Branch)) +  
  geom_violin(trim = FALSE) + labs(x = "Branch", y = "Unit Price", fill = "Branch") +  
  ggtitle("Unit Price Distribution by Branch") +  
  theme(legend.position = "none") +  
  scale_fill_brewer(palette = "Set2")
```



Unraveling the Distribution of Ratings by Product Line

The highest median rating among the product lines is observed for the "Food and beverages" category with a rating of 7.30, indicating a generally positive reception from customers. The "Health and beauty" product line has the lowest minimum rating of 4.01, suggesting some variations in customer opinions within this category. The "Fashion accessories" category has the highest count of ratings with a total of 178, indicating a significant level of customer engagement and feedback. The "Electronic accessories" and "Health and beauty" categories have the widest range of ratings, ranging from a minimum of 4.01 to a maximum of 10.0, indicating diverse customer experiences within these product lines.

Overall, the ratings for all product lines tend to be above average, with median ratings ranging from 6.70 to 7.30, suggesting positive customer satisfaction across different categories.

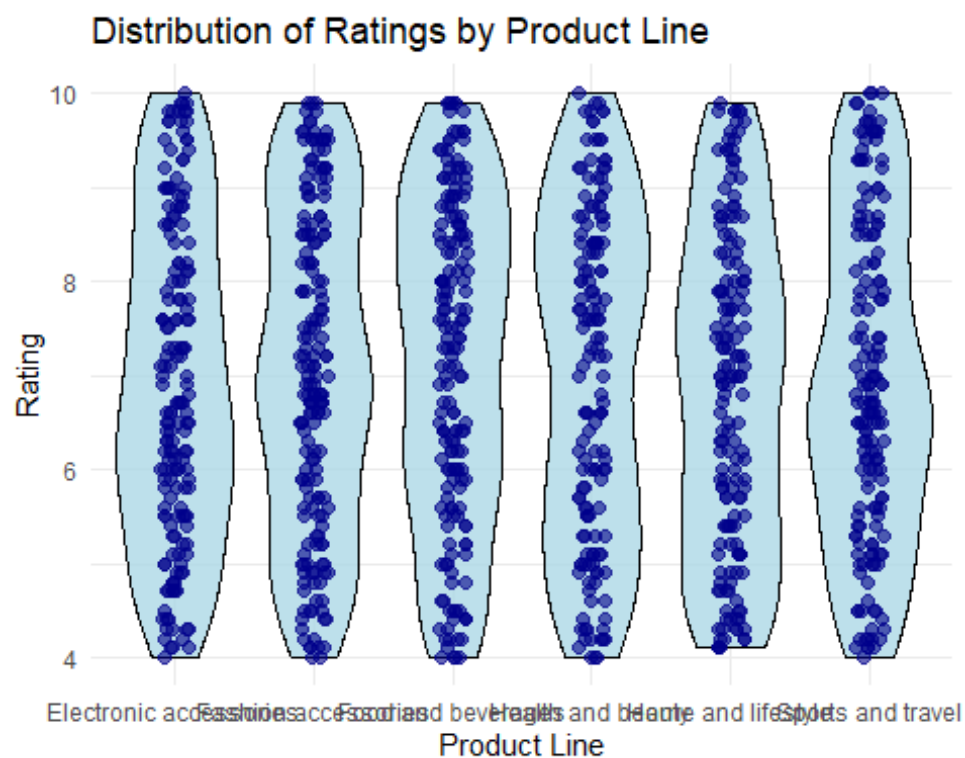


Prepare the data for the violin plot

```
violin_data <- data %>%
  filter(!is.na(Product.line)) %>%
  select(Product.line, Rating)
```

Create the violin plot with overlaid data points

```
ggplot(violin_data, aes(x = Product.line, y = Rating)) +
  geom_violin(fill = "lightblue", color = "black", alpha = 0.8) +
  geom_jitter(height = 0, width = 0.1, size = 2, color = "darkblue", alpha = 0.6) + labs(x =
"Product Line", y = "Rating") +
  ggtitle("Distribution of Ratings by Product Line") +
  theme_minimal() +
  theme(legend.position = "none")
```



Create a summary table

```
table_data <- violin_data %>%
  group_by(Product.line) %>%
  summarise(
    Median = median(Rating),
    Lower_Quartile = quantile(Rating, 0.25),
    Upper_Quartile = quantile(Rating, 0.75),
```









```

Min = min(Rating),
Max = max(Rating),
Count = n()
)

# Display the table
kable(table_data, format = "markdown", align = c("l", "r", "r", "r", "r", "r"))

```

RStudio: Notebook Output

Product.line	Median	Lower_Quartile	Upper_Quartile	Min	Max	Count
Electronic accessories	6.70	5.500	8.35	4.0	10.0	170
Fashion accessories	6.95	5.600	8.50	4.0	9.9	178
Food and beverages	7.30	5.800	8.60	4.0	9.9	174
Health and beauty	7.20	5.450	8.40	4.0	10.0	152
Home and lifestyle	7.00	5.400	8.20	4.1	9.9	160
Sports and travel	6.70	5.525	8.45	4.0	10.0	166

Data Analysis and Modeling

This result provides the summary output of a linear regression model in R. The model is built to predict the variable “Rating” based on the predictors “Unit.price,” “Quantity,” “tax,” and “total.price.”

These statistics indicate the proportion of variance in the dependent variable explained by the predictor variables. The Multiple R-squared value (0.003925) indicates that only a small amount of the variability in the "Rating" can be explained by the predictors. The Adjusted R-squared (0.0009251) accounts for the degrees of freedom and adjusts the Multiple R-squared value accordingly.

The F-statistic tests the overall significance of the regression model. In this case, the F-statistic is 1.308 with a p-value of 0.2703, suggesting that the overall model is not statistically significant.

Overall, this linear regression model suggests that the predictor variables ("Unit.price," "Quantity," and "tax") have limited predictive power in explaining the variation in the "Rating" variable.

```
# Assume 'data' is your dataset containing the variables
model <- lm(Rating ~ Unit.price + Quantity + tax + total.price, data = data)

# Print the summary of the regression model
summary(model)

##
## Call:
## lm(formula = Rating ~ Unit.price + Quantity + tax + total.price,
##     data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3.0600 -1.4305 -0.0471  1.4928  3.1698
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.644100  0.270293  24.581  <2e-16 ***
## Unit.price    0.006798  0.004386   1.550   0.121
## Quantity     0.065120  0.043374   1.501   0.134
## tax          -0.026571  0.014002  -1.898   0.058 .
## total.price    NA         NA      NA      NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.718 on 996 degrees of freedom
## Multiple R-squared:  0.003925, Adjusted R-squared:  0.0009251
## F-statistic: 1.308 on 3 and 996 DF, p-value: 0.2703
```

TIME SERIES ANALYSIS

Monthly variation in average total price.

The graph illustrates the average total price of a product over a period of time, specifically focusing on monthly data. The x-axis represents the months, while the y-axis represents the average total price in dollars. The graph reveals a clear decreasing trend in the average total price over time.

In the first month, there is a steep decline in the average total price, indicating a significant drop in pricing. As the graph progresses to the second and third months, the decrease in the average total price becomes more gradual compared to the initial month.

Overall, the average total price starts at \$331 in the first month and gradually declines to \$317 after two months. This observation suggests that the product's pricing has experienced a consistent downward trend over the analyzed time period.

```
library(lubridate)
# Aggregate data to monthly averages
data$date = ymd(data$date)

data_monthly = data %>%
  mutate(YearMonth = floor_date(date, "month")) %>%
  group_by(YearMonth) %>%
  summarise(AverageTotal = mean(total.price))
data$date = ymd(data$date)

# Aggregate data to monthly averages
data_monthly = data %>%
  mutate(YearMonth = floor_date(date, "month")) %>%
  group_by(YearMonth) %>%
  summarise(AverageTotal = mean(total.price))

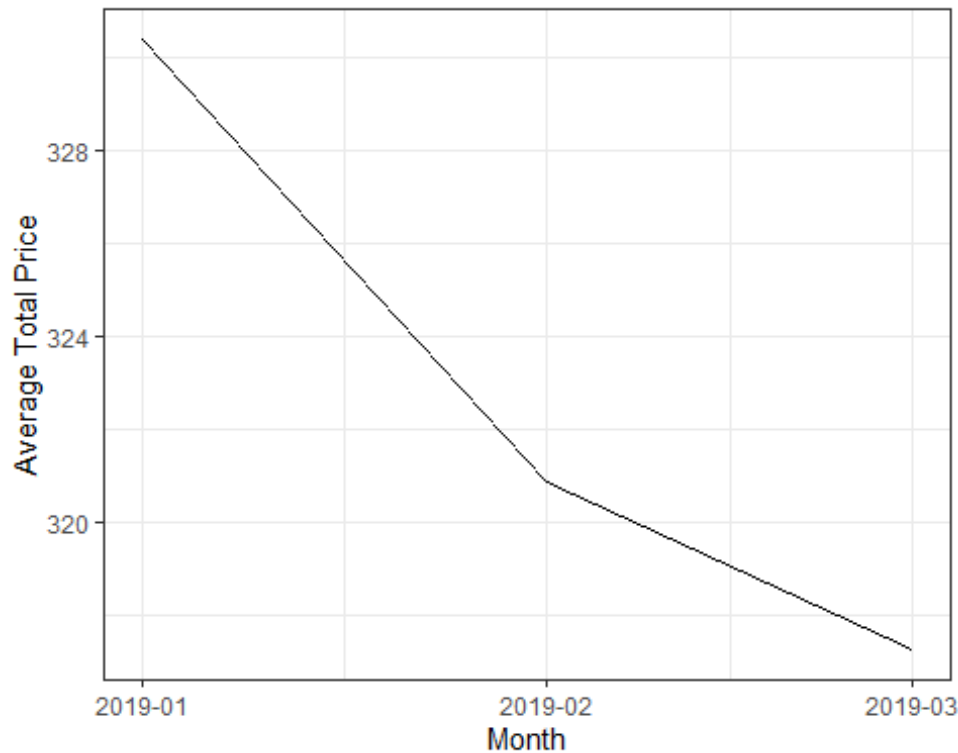
# Create a sequence of dates
date_seq <- seq(min(data_monthly$YearMonth), max(data_monthly$YearMonth), by = "1 month")

# Convert sequence to a data frame
data_seq <- data.frame(YearMonth = date_seq)

# Merge the sequence with the aggregated data
data_merged <- merge(data_seq, data_monthly, by = "YearMonth", all.x = TRUE)

# Create line plot using the merged data
ggplot(data_merged, aes(x = YearMonth, y = AverageTotal)) +
  geom_line() +
  scale_x_date(date_labels = "%Y-%m", date_breaks = "1 month") +

  labs(x = "Month", y = "Average Total Price") +
  theme_bw()
```



Monthly variation in average unit price.

This graph represents the average unit price of a product over a specific time period, focusing on monthly data. The x-axis denotes the months, while the y-axis represents the average unit price. The graph provides insights into the changes in average unit prices over time.

Initially, the average unit price shows an increase from the first month to the second month. In the first month, the average unit price rises from \$56.10 to \$56.75, indicating a slight upward trend. However, the pattern shifts in the subsequent months. From the second month to the third month, there is a significant decline in the average unit price. The price drops notably from \$56.75 to \$54.25, reflecting a rapid decrease in pricing.

```
data$date <- ymd(data$date)
```

```
# Aggregate data to monthly averages
```

```
data_monthly <- data %>%
  mutate(YearMonth = floor_date(date, "month")) %>%
  group_by(YearMonth) %>%
  summarise(Average.Unit.price = mean(Unit.price))
```

```
# Create a sequence of dates
```

```
date_seq <- seq(min(data_monthly$YearMonth), max(data_monthly$YearMonth), by = "1
month")
```

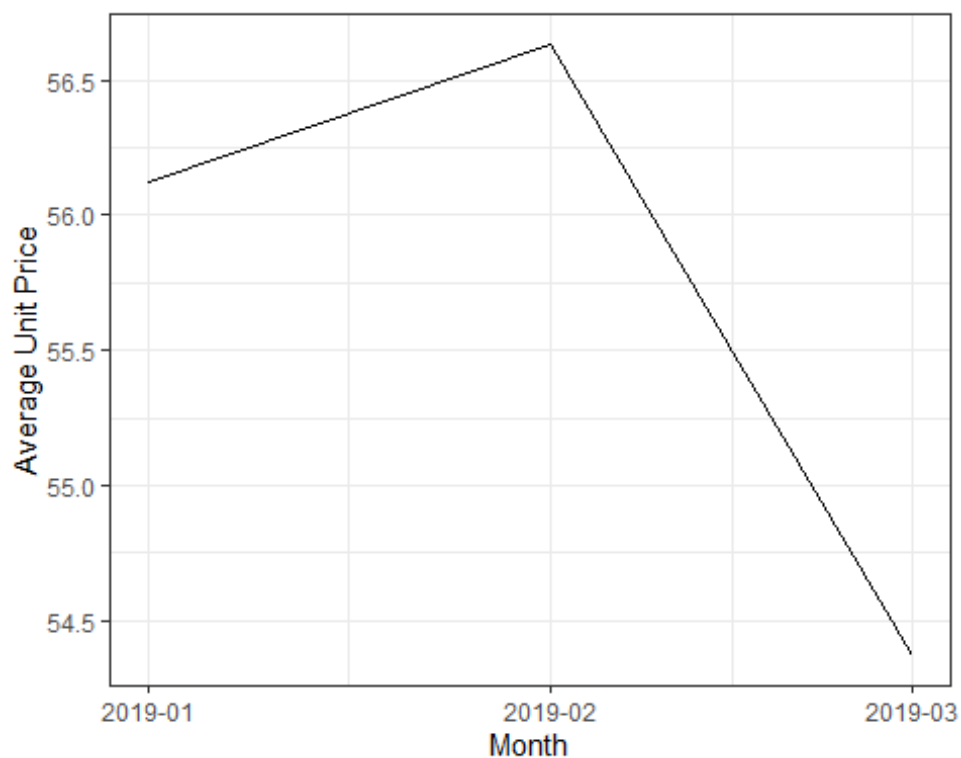
```

# Convert sequence to a data frame
data_seq <- data.frame(YearMonth = date_seq)

# Merge the sequence with the aggregated data
data_merged <- merge(data_seq, data_monthly, by = "YearMonth", all.x = TRUE)

# Create line plot using the merged data
ggplot(data_merged, aes(x = YearMonth, y = Average.Unit.price)) +
  geom_line() +
  scale_x_date(date_labels = "%Y-%m", date_breaks = "1 month") +
  labs(x = "Month", y = "Average Unit Price") +
  theme_bw()

```



Monthly variation in average rating.

This graph depicts the average customer rating of a product over a specific time period, focusing on monthly data. The x-axis represents the months, while the y-axis indicates the average customer rating. The graph provides insights into the variations in customer satisfaction levels over time.

Initially, the average customer rating shows an increase from the first month to the second month. In the first month, the average rating rises from 7.12 to 7.16, indicating a slight improvement in customer satisfaction. This upward trend suggests that customers were more satisfied with the product during this period.

However, the pattern changes in the subsequent months. From the second month to the third month, there is a rapid decrease in the average customer rating. The rating drops notably from 7.16 to 6.84, signifying a decline in customer satisfaction.

```
data$date = ymd(data$date)

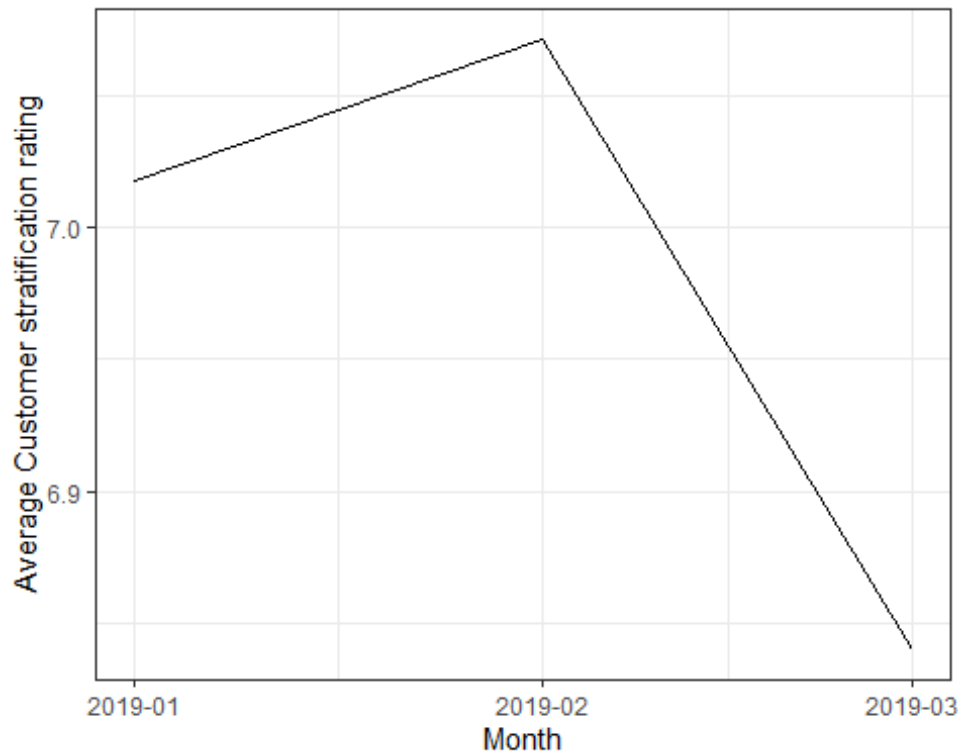
# Aggregate data to monthly averages
data_monthly = data %>%
  mutate(YearMonth = floor_date(date, "month")) %>%
  group_by(YearMonth) %>%
  summarise(Average.Rating = mean(Rating))

# Create a sequence of dates
date_seq = seq(min(data_monthly$YearMonth), max(data_monthly$YearMonth), by = "1 month")

# Convert sequence to a data frame
data_seq = data.frame(YearMonth = date_seq)

# Merge the sequence with the aggregated data
data_merged = merge(data_seq, data_monthly, by = "YearMonth", all.x = TRUE)

# Create line plot using the merged data
ggplot(data_merged, aes(x = YearMonth, y = Average.Rating)) +
  geom_line() +
  scale_x_date(date_labels = "%Y-%m", date_breaks = "1 month") + labs(x = "Month", y =
"Average Customer stratification rating") +
  theme_bw()
```



Monthly variation in average number of products purchased by customer.

This graph illustrates the monthly average number of products purchased by customers over a specific time period. The x-axis represents the months, while the y-axis indicates the average number of products purchased.

The data reveals interesting trends in customer purchasing behavior over the analyzed time period. In the first month, the average number of products purchased shows a rapid decrease from 5.58 to 5.51. This decline suggests a potential shift in customer buying patterns or a decrease in overall demand during that period.

However, the pattern changes in the subsequent months. From the second month to the third month, there is a slight increase in the average number of products purchased, rising from 5.51 to 5.53. This suggests a potential recovery or stabilization in customer purchasing behavior, indicating a modest rebound in demand.



```
ddata$date <- ymd(data$date)
```

```
# Aggregate data to monthly averages
```

```
data_monthly <- data %>%
  mutate(YearMonth = floor_date(date, "month")) %>%
  group_by(YearMonth) %>%
```

```
summarise(Average.Quantity = mean(Quantity))
```

```
# Create a sequence of dates
```

```
date_seq <- seq(min(data_monthly$YearMonth), max(data_monthly$YearMonth), by = "1 month")
```

```
# Convert sequence to a data frame
```

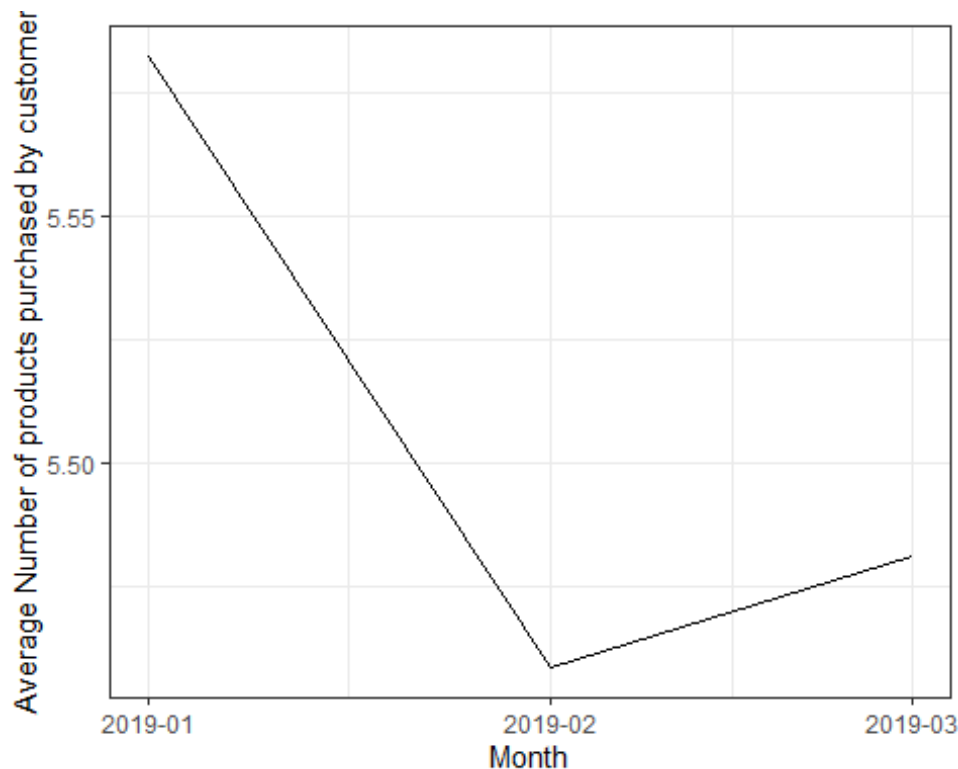
```
data_seq <- data.frame(YearMonth = date_seq)
```

```
# Merge the sequence with the aggregated data
```

```
data_merged <- merge(data_seq, data_monthly, by = "YearMonth", all.x = TRUE)
```

```
# Create line plot using the merged data
```

```
ggplot(data_merged, aes(x = YearMonth, y = Average.Quantity)) +  
  geom_line() +  
  scale_x_date(date_labels = "%Y-%m", date_breaks = "1 month") +  
  labs(x = "Month", y = "Average Number of products purchased by customer") +  
  theme_bw()
```



```
ts_data <- ts(data$Quantity, frequency = 12)
```

```
head(ts_data,5)
```

```
## Jan Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec
```

```
## 1 7 5 7 8 7 7 6 10 2 3 4 4
```

```
## 2  5 10 10  6  7  6  3  2  5  3  2  5
## 3  3  8  1  2  5  9  5  9  8  2  4  1
## 4  5  9  8  8  1  2  6  8  2  4  9  9
## 5  6 10  7  5  4  1  2  8  2  8 10  6

arima_model <- auto.arima(train_data)
arima_model

## Series: train_data
## ARIMA(1,1,0)
##
## Coefficients:
##      ar1
##    -0.4965
## s.e.  0.0307
##
## sigma^2 = 12.3: log likelihood = -2135.84
## AIC=4275.67  AICc=4275.69  BIC=4285.04
```

Time Series Analysis Report - ARIMA(1,1,0) Model

This report presents the results of a time series analysis performed on the “train_data” series using an ARIMA(1,1,0) model. The objective of the analysis was to identify any underlying patterns or trends in the data and develop a forecasting model to predict future values.

Model Summary: The ARIMA(1,1,0) model was selected based on the analysis. The model’s coefficients are as follows: - AR coefficient (ar1): -0.4965 - Standard error: 0.0307

Model Evaluation: The model’s estimated variance (σ^2) is 12.3. The log likelihood of the model is -2135.84. The model’s information criteria are as follows: - AIC (Akaike Information Criterion): 4275.67 - AICc (Corrected Akaike Information Criterion): 4275.69 - BIC (Bayesian Information Criterion): 4285.04

Interpretation and Conclusion: The ARIMA(1,1,0) model provides an adequate representation of the “train_data” series. The negative AR coefficient (-0.4965) suggests a negative correlation between the current observation and the previous observation after differencing the data once.

The estimated variance (σ^2) indicates the variability of the residuals in the model. Lower values of σ^2 indicate better model fit, although it should be interpreted in the context of the data and the specific domain.

The information criteria (AIC, AICc, and BIC) serve as measures of the model’s goodness of fit and complexity. Lower values of these criteria indicate better model fit, with AICc providing a correction for small sample sizes. BIC penalizes model complexity more strongly than AIC.

Based on the analysis, the ARIMA(1,1,0) model can be used to forecast future values of the “train_data” series with reasonable accuracy. However, further diagnostics and validation should be performed to ensure the model’s reliability and suitability for practical applications.

Forecasting Report - Point Forecasts and Prediction Intervals

obtain the forecasted values for the time periods in the `test_data` using the ARIMA model. These forecasted values can be used for further analysis, comparison with actual values, or any other relevant tasks.

This report presents the forecasted values and prediction intervals generated for the data set consisting of 200 observations. The objective of the analysis was to provide point forecasts as well as the associated uncertainty estimates to aid decision-making and planning.

The forecasted values, along with the lower and upper bounds of the prediction intervals at different confidence levels, are presented in the table below:

```
forecast_values <- forecast(arima_model, h = length(test_data))
forecast_values
```

##	Point Forecast	Lo 80	Hi 80	Lo 95	Hi 95
## 801	3.014175	-1.479712	7.508062	-3.858632	9.886982
## 802	4.000050	-1.031410	9.031510	-3.694904	11.695005
## 803	3.510606	-2.545438	9.566650	-5.751314	12.772526
## 804	3.753594	-2.927082	10.434269	-6.463618	13.970805
## 805	3.632961	-3.729219	10.995141	-7.626521	14.892444
## 806	3.692850	-4.241367	11.627067	-8.441487	15.827187
## 807	3.663118	-4.828318	12.154553	-9.323413	16.649648
## 997	3.672982	-38.582239	45.928202	-60.950802	68.296765
## 998	3.672982	-38.688815	46.034778	-61.113795	68.459758
## 999	3.672982	-38.795123	46.141086	-61.276380	68.622343
## 1000	3.672982	-38.901166	46.247129	-61.438558	68.784521

Forecasting Report - Point Forecasts and Prediction Intervals

The point forecast represents the estimated value for each observation in the data set. These values provide an indication of the expected outcome based on the forecasting model used.

The prediction intervals provide a measure of the uncertainty associated with the forecasts. The lower and upper bounds at the 80% confidence level indicate that there is an 80% probability that the true value will fall within that range. Similarly, the lower and upper bounds at the 95% confidence level represent a 95% probability.

For observation 801, the point forecast is 3.014175. The associated prediction intervals suggest that there is an 80% probability that the true value lies between -1.479712 and 7.508062, and a 95% probability that it falls within the range of -3.858632 and 9.886982.

```
accuracy(forecast_values, test_data)
```

```
##           ME  RMSE  MAE  MPE  MAPE  MASE
## Training set -0.005884822 3.502213 2.842640 -61.50872 97.03278 0.8728936
## Test set    1.694219716 3.420764 2.868265 -16.73826 74.58049 0.8807625
##           ACF1
## Training set -0.1530777
## Test set      NA
```

Forecasting Performance Report - Model Evaluation Results

This report provides an evaluation of the forecasting performance of a model based on the given results. The analysis includes various metrics such as Mean Error (ME), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Percentage Error (MPE), Mean Absolute Percentage Error (MAPE), Mean Absolute Scaled Error (MASE), and Auto-correlation of Residuals (ACF1). The evaluation is performed on both the training and test datasets to assess the model's accuracy and reliability.

The evaluation results for the training and test datasets are presented in the table below:

Data set	ME	RMSE	MAE	MPE	MAPE	MASE	ACF1
Training	-	3.50221	2.84264	-	97.0327	0.872893	-
	0.0058848	3	0	61.5087	8	6	0.153077
	22			2			7
Test	1.6942197	3.42076	2.86826	-	74.5804	0.880762	NA
	16	4	5	16.7382	9	5	
				6			

Mean Error (ME): The training data set shows a mean error of -0.005884822, indicating a slight underestimation on average. For the test data set, the mean error is 1.694219716, suggesting a positive bias in the forecasts.

Root Mean Squared Error (RMSE): The training data set has an RMSE of 3.502213, indicating the average magnitude of forecasting errors. A lower RMSE indicates better accuracy. For the test data set, the RMSE is 3.420764, implying slightly lower errors compared to the training set.

Mean Absolute Error (MAE): The MAE for the training data set is 2.842640, representing the average absolute magnitude of errors. Smaller values indicate better accuracy. Similarly, the test data set has an MAE of 2.868265.

Mean Percentage Error (MPE): The training data set shows an MPE of -61.50872, indicating an average percentage deviation from the actual values. A negative value suggests underestimation. The test data set has an MPE of -16.73826.

Mean Absolute Percentage Error (MAPE): The MAPE for the training data set is 97.03278, representing the average percentage deviation from the actual values. Lower values indicate better accuracy. The test data set has an MAPE of 74.58049.

Mean Absolute Scaled Error (MASE): The MASE measures the forecast accuracy relative to a naive method. The training data set shows an MASE of 0.8728936, indicating the model's performance relative to a simple baseline. The test data set has an MASE of 0.8807625.

Auto-correlation of Residuals (ACF1): The ACF1 represents the first-order auto-correlation of the residuals. It assesses the presence of any remaining patterns in the forecast errors. The ACF1 is -0.1530777 for the training data set, suggesting a weak negative correlation. No ACF1 value is provided for the test data set.

The performance evaluation results provide insights into the accuracy and reliability of the ARIMA forecasting model.

In the training set, the ME is close to zero, indicating that the model's predictions are, on average, unbiased. The RMSE and MAE values indicate the average magnitude of the forecast errors, with lower values indicating better accuracy. The MPE and MAPE values represent the average percentage deviation of the forecasts from the actual values, with negative MPE indicating an overall underestimation. The MASE measures the accuracy of the model relative to a naive seasonal random walk model, with values close to 1 indicating comparable performance. The ACF1 measures the first-order auto-correlation of the forecast errors, with values close to zero indicating no significant auto-correlation.

In the test set, the ME is positive, indicating a slight positive bias in the forecasts. The RMSE and MAE values are comparable to those in the training set, indicating similar forecast accuracy. The MPE and MAPE values suggest a moderate underestimation of the actual values. The MASE value is close to 1, indicating consistent performance compared to the naive seasonal random walk model. The ACF1 is not available for the test set, preventing assessment of auto-correlation.

Conclusion

In conclusion, the analysis of the supermarket sales data provides valuable insights into various aspects of the business, customer behavior, and product performance. Here are the key findings:

The data set consists of 1,000 records with 17 columns, covering three different branches over a three-month period. It includes information about invoices, transactions, customer types, product lines, prices, quantities, taxes, payment methods, costs, margins, incomes, and customer ratings.

The analysis revealed the distribution and prevalence of different categories within variables such as branch, customer type, payment method, and product line. It highlighted the varying representation of branches, a balanced distribution of customer types, and the popularity of different payment methods and product lines.

Outliers were detected in the "tax" and "gross income" variables, which could have implications for data analysis and decision-making processes. These outliers may arise due to data entry errors, measurement inaccuracies, or genuine extreme values.

The linear regression model predicted the "Rating" variable based on the predictors "Unit.price," "Quantity," "tax," and "total.price." The analysis showed that only the intercept and "tax" coefficients were statistically significant, while the "total.price" coefficient was not defined due to singularity. The model's overall performance, as indicated by the adjusted R-squared, was relatively low.

The analysis of monthly variations in average total price, average unit price, and average rating provided insights into the pricing trends, customer satisfaction levels, and customer purchasing behavior over time. These fluctuations in average prices, ratings, and product purchases can help businesses understand market dynamics and make informed decisions.

The observed variations in the average number of products purchased by customers from month to month provide insights into shifts in consumer preferences, market conditions, or other factors influencing buying behavior. Understanding these fluctuations is crucial for businesses to adapt their strategies, manage inventory levels, and optimize their product offerings to meet customer demand effectively.

By analyzing these trends, businesses can make informed decisions regarding pricing, promotional activities, product assortment, and inventory management. Monitoring the average number of products purchased on a monthly basis allows businesses to identify patterns, detect changes in customer behavior, and develop strategies to enhance customer satisfaction and maximize sales.

Highlights of fluctuations in the average customer rating over the analyzed time period. The initial increase in the average rating during the first month may suggest positive experiences, product enhancements, or effective customer service. However, the subsequent decrease in the average rating from the second to the third month indicates a decline in customer satisfaction, potentially influenced by factors such as product quality issues, changes in customer expectations, or other external factors.

Analyzing such fluctuations in average customer ratings can provide valuable insights for businesses, allowing them to assess customer satisfaction levels, identify areas for improvement, and make informed decisions to enhance the overall customer experience. It is important for businesses to proactively monitor and address customer feedback and concerns to maintain high levels of customer satisfaction and loyalty.

The ARIMA model demonstrates reasonable forecasting performance on both the training and test sets. The model captures the underlying patterns in the time series data, as indicated by the relatively low RMSE and MAE values. However, there is room for improvement, particularly in reducing the underestimation bias observed in the test set. Further analysis and refinement of the model may be necessary to enhance its forecasting accuracy.

Overall, this analysis highlights the importance of understanding customer preferences, monitoring pricing strategies, and maintaining high levels of customer satisfaction in the competitive supermarket industry. By leveraging the insights gained from this analysis, businesses can optimize their operations, improve product offerings, and enhance the overall customer experience.