DATA ANALISYS FINAL PROJECT

Boris Ivanov, Hugo González, Hamza Tebri and Nikolas Zannettou

2024-01-22

Contents

1.SET UP

1.1LIBRARIES AND DATASETS

```
# Loading Libraries
library(readxl)       # For reading Excel files
library(openxlsx)     # Enhanced Excel file reading and writing
library(dplyr)        # Data manipulation
```

```
##
## 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
```

```r
library(tidyverse)    # Data manipulation and visualization
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.4
## v ggplot2   3.4.3     v stringr   1.5.0
## v lubridate 1.9.2     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```r
library(ggplot2)      # For data visualization
library(DataExplorer) # Automating exploratory data analysis

# Loading Datasets
base_path <- "/Users/hugogonzalez/Desktop/BIDA /DATA ANALISYS /"
Orders <- read_excel(paste0(base_path, "AEKI_Data.xlsx"), sheet = "Orders")
Products <- read_excel(paste0(base_path, "AEKI_Data.xlsx"), sheet = "Products")
Returns <- read_excel(paste0(base_path, "AEKI_Data.xlsx"), sheet = "Returns")
Demographics <- read_excel(paste0(base_path, "AEKI_Data.xlsx"), sheet = "Demographics")
World_Cities <- read.csv(paste0(base_path, "worldcities.csv"))
Orders_2016 <- read_excel(paste0(base_path, "AEKI_2016.xlsx"))
```

1.2RENAMING

```r
# Standardizing The Datasets

# Renaming Columns in Products Dataset
colnames(Products)[colnames(Products) == "ID Product"] <- "Product ID"

# Function to Replace Spaces and Hyphens in Column Names
rename_columns <- function(data) {
    colnames(data) <- gsub(" |-", "_", colnames(data))
    return(data)
}

# Applying the Function to Rename Columns
Orders <- rename_columns(Orders)
Products <- rename_columns(Products)
Returns <- rename_columns(Returns)
Demographics <- rename_columns(Demographics)
Orders_2016 <- rename_columns(Orders_2016)

# Dropping Columns in Orders Dataset
```

```r
Orders <- subset(Orders, select = -Row_ID)
Orders_2016 <- subset(Orders_2016, select = -Row_ID)

# Renaming and Filtering Columns in World Cities Dataset
World_Cities <- World_Cities %>%
    select(city, country, admin_name, lat, lng, population, id) %>%
    rename(City_ID = id, City_Population = population, City = city, State = admin_name, Country = count
    filter(Country == "United States")

# Renaming Specific City Names in World Cities Dataset
World_Cities$City[World_Cities$City == "New York"] <- "New York City"
World_Cities$City[World_Cities$City == "Port St. Lucie"] <- "Port Saint Lucie"
World_Cities$City[World_Cities$City == "McAllen"] <- "Mcallen"
World_Cities$City[World_Cities$City == "St. Cloud" & World_Cities$State == "Minnesota"] <- "Saint Cloud
World_Cities$City[World_Cities$City == "St. Petersburg"] <- "Saint Petersburg"
World_Cities$City[World_Cities$City == "St. Charles"] <- "Saint Charles"
World_Cities$City[World_Cities$City == "St. Louis"] <- "Saint Louis"
World_Cities$City[World_Cities$City == "St. Peters"] <- "Saint Peters"
World_Cities$City[World_Cities$City == "St. Paul"] <- "Saint Paul"
World_Cities$City[World_Cities$City == "Milford city"] <- "Milford"
World_Cities$City[World_Cities$City == "Novi" & World_Cities$State == "Michigan"] <- "Canton"
```

2.DATA MANIPULATION

```r
# Create a new dataframes with selected columns from the original data frame
Orders_Processing <- Orders
Products_Processing <- Products
Orders_2016_Processing <- Orders_2016
Returns_Processing <- Returns
Citites_Processing <- World_Cities
Demographics_Processing <- Demographics

# Products Dataframe Preparation

# Dropping the duplicate rows from the Product ID column
# Dropping the duplicate rows from the Product ID column
Products_Processing <- Products_Processing[!duplicated(Products_Processing$Product_ID), ]


# Rename the values in columnn Category:  "Office Suplies"  to "Office Supplies"
Products_Processing$Category[Products_Processing$Category == "Office Suplies"] <- "Office Supplies"

# Merge Data Frames: Orders and Products

# Combine the two data sets based on the Product ID column
Orders_Processing <- merge(Orders_Processing, Products_Processing, by = "Product_ID")

#Merge Data Frames: Orders_Products and Orders_2016

# Combine the two data sets based on the Order ID column
Orders_Processing <- rbind(Orders_Processing, Orders_2016_Processing)
```

```r
# Merge Data Frames: Orders_Processing and Returns
Orders_Processing <- Orders_Processing %>%
    left_join(Returns_Processing[, c("Order_ID", "Returned")], by = "Order_ID")

# Replace NA in the Returned column with "no"
Orders_Processing$Returned[is.na(Orders_Processing$Returned)] <- "No"

# Fix Values in Country Column

# Add observations with Italy values into another dataset
Orders_Wrong_Entries <- Orders_Processing[Orders_Processing$Country == "Italy", ]

# Drop the observations with Italy values from the original dataset
Orders_Processing <- Orders_Processing[Orders_Processing$Country != "Italy", ]

# Discount

Orders_Processing <- Orders_Processing %>%
    mutate(Discount = case_when(
        Discount == 1.4 ~ 0.4,
        Discount == 1.6 ~ 0.6,
        TRUE ~ Discount
    ))

# Quantity Column

# Put it in the Orders_Wrong_Entries df
Orders_Wrong_Entries <- rbind(Orders_Wrong_Entries, Orders_Processing[Orders_Processing$Quantity == 19,

# Drop it from the original dataset
Orders_Processing <- Orders_Processing[Orders_Processing$Quantity != 19, ]

# We drop it beacause sales are 0,0002

#Profit Column
# Add the observations with profit more than 22638.48 to the Orders_Wrong_Entries df
Orders_Wrong_Entries <- rbind(Orders_Wrong_Entries, Orders_Processing[Orders_Processing$Profit > 22638.

# Drop the observations with profit more than 22638.48 from the original dataset
Orders_Processing <- Orders_Processing[Orders_Processing$Profit <= 22638.48, ]

# Products

# Add Test value from sub-category column to Wrong_Entries df
Orders_Wrong_Entries <- rbind(Orders_Wrong_Entries, Orders_Processing[Orders_Processing$Sub_Category ==

# Drop Test value from sub-category column
Orders_Processing <- Orders_Processing[Orders_Processing$Sub_Category != "Test", ]

# Merging Data Frames: Orders_Processing with Cities
# Rename Orange to East Orange ###
Orders_Processing$City[Orders_Processing$City == "Orange"] <- "East Orange"
```

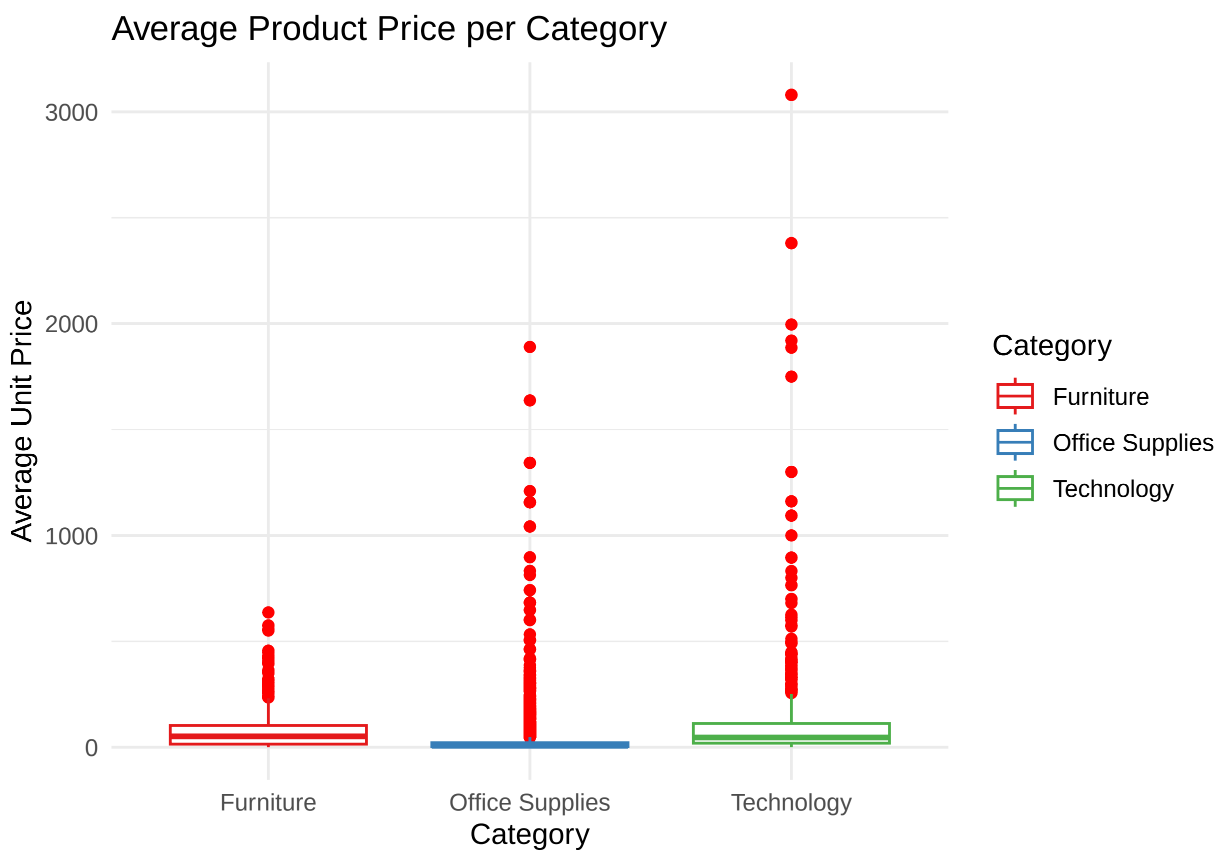
```r
# Merge the datasets again
Orders_Processing <- Orders_Processing %>%
    left_join(Citites_Processing, by = c("City" = "City", "State" = "State", "Country" = "Country"))


# Merging Data Frames: Orders_Processing with Demographics
# Perform the left join
Orders_Processing <- Orders_Processing %>%
    left_join(Demographics_Processing, by = "State")

#Extracting Time Data

# ExtraActing the year

# Extract the year and add it as a new column in the new dataset
Orders_Processing$Year <- as.numeric(format(Orders_Processing$Order_Date, "%Y"))

#Extracting the month
# Extract the month and add it as a new column in the new dataset
Orders_Processing$Month <- as.numeric(format(Orders_Processing$Order_Date, "%m"))

# Extracting the day

# Extract the day and add it as a new column in the new dataset
Orders_Processing$Day <- as.numeric(format(Orders_Processing$Order_Date, "%d"))

# Extracting the week-day

Orders_Processing$Day_of_Week <- weekdays(Orders_Processing$Order_Date)

#Calculating Processing Days

# Calucalate processing days based on Order_Date and Ship_Date
# Calculate the distance between the Order_Date and Ship_Date
Orders_Processing$Processing_Days <- as.numeric(difftime(Orders_Processing$Ship_Date, Orders_Processing


# Calculating Product Prices
# Calculate Net Sales and Unit Price
Orders_Processing <- Orders_Processing %>%
    mutate(
        Net_Sales = Sales - (Sales * Discount), # Calculate Net Sales
        Unit_Price = Net_Sales / Quantity # Calculate Unit Price
    )


# Calculate Avg Min Max

Orders_Processing <- Orders_Processing %>%
    group_by(Product_ID, Year) %>%
    mutate(
        Average_Unit_Price = mean(Unit_Price, na.rm = TRUE),
        Max_Unit_Price = max(Unit_Price, na.rm = TRUE),
```

```
        Min_Unit_Price = min(Unit_Price, na.rm = TRUE)
    ) %>%
    ungroup()


# Calculate Total Order, Profit

Orders_Processing <- Orders_Processing %>%
    group_by(Order_ID) %>%
    mutate(
        Total_Order_Price = sum(Net_Sales, na.rm = TRUE), # Total Net Sales per order
        Total_Order_Profit = sum(Profit, na.rm = TRUE) # Total Profit per order
    ) %>%
    ungroup()
```

3.DESCRIPTIVE ANALISYS

3.1 PRODUCT CATEGORIES

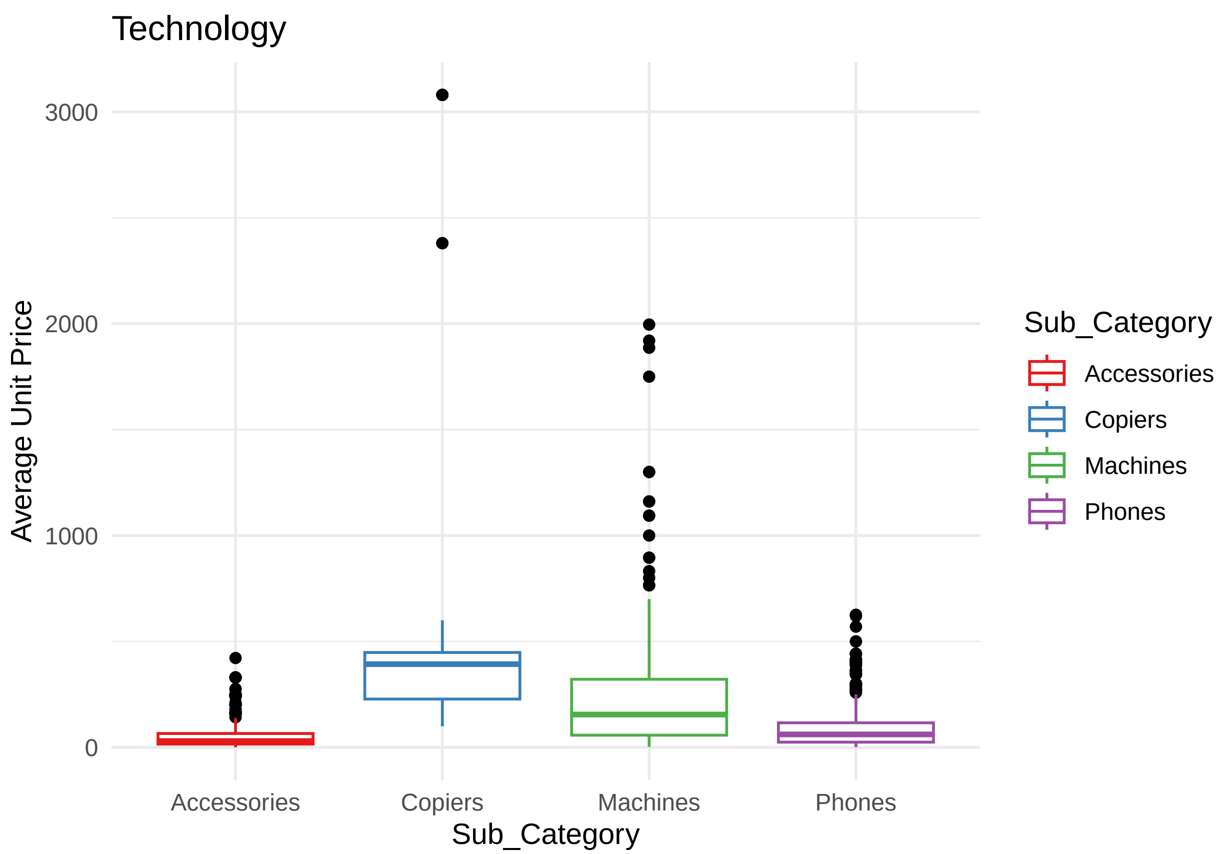
```
ggplot(Orders_Processing, aes(x = Category, y = Average_Unit_Price, color = Category)) +
    geom_boxplot(outlier.color = "#ff0000") +
    ggtitle("Average Product Price per Category") +
    xlab("Category") +
    ylab("Average Unit Price") +
    scale_color_brewer(palette = "Set1") +
    theme_minimal()
```



The plot reveals significant variations in product pricing across different categories, indicating the presence of both high ticket and low ticket products. Notably, technology products include the most expensive items. The presence of outliers predominantly on the higher side suggests a right-skewed (or positively skewed) distribution, implying that the bulk of the data is concentrat ed towards the lower end. Given this complexity, it is somewhat challenging to discern the precise dynamics at play. Therefore, to gain a clearer understanding, let's proceed to plot the data per sub-category.Plotting The Sub Categories

3.1.1TECHNOLOGY

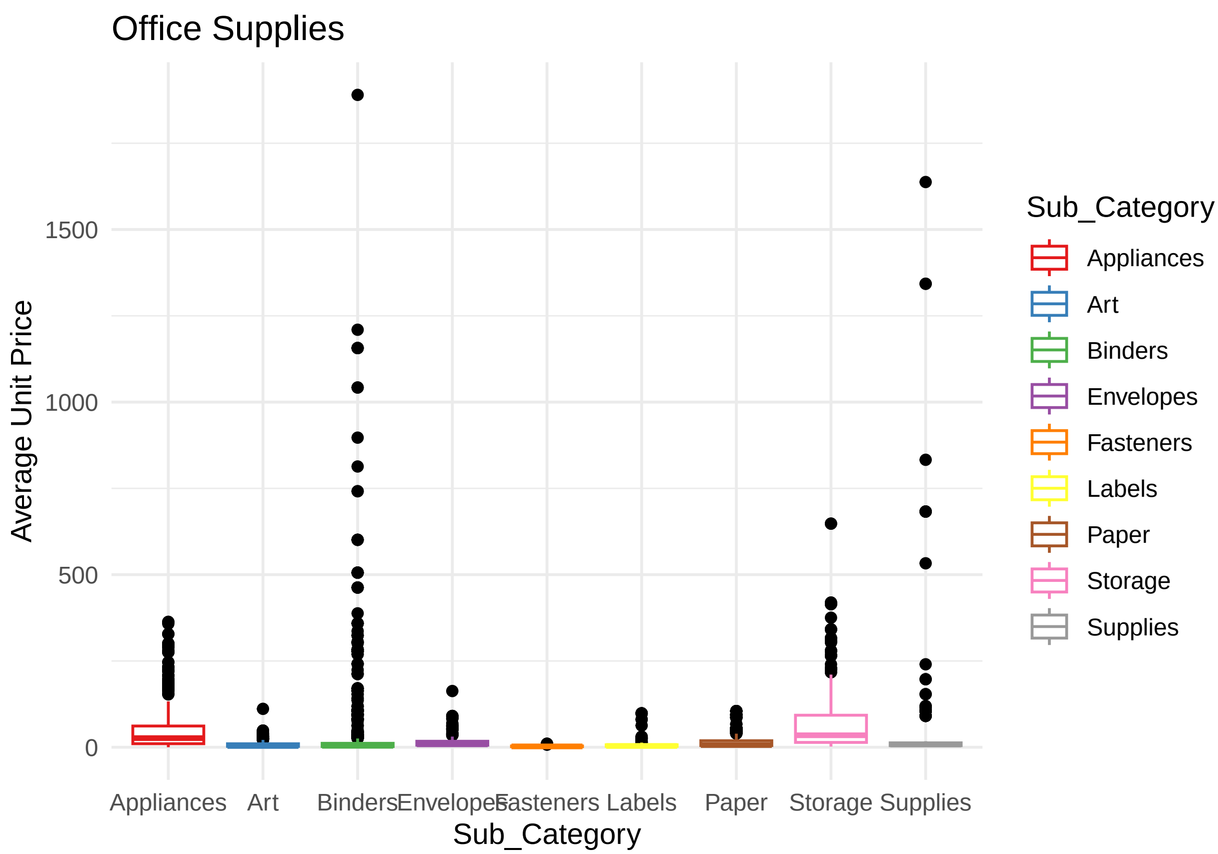
```
Orders_Processing %>%
    filter(Category == "Technology") %>%
    ggplot(aes(x = Sub_Category, y = Average_Unit_Price, color = Sub_Category)) +
    geom_boxplot(outlier.color = "black") +
    ggtitle("Technology") +
    xlab("Sub_Category") +
    ylab("Average Unit Price") +
    scale_color_brewer(palette = "Set1") +
    theme_minimal()
```



From the dots in the plot, it is evident that outliers exist in every sub-category. The subcategory that stands out with the most significant outliers is 'Copiers'. It is also the most expensive sub-category, with its lowest value being higher than the median of most other sub-categories. The median value across all sub-categories falls below 500, indicating a general trend toward lower-priced items. Additionally, the majority of values are clustered between the 25th and 75th percentiles, highlighting a concentration of data within this range.

3.1.2OFFICE SUPPLIES

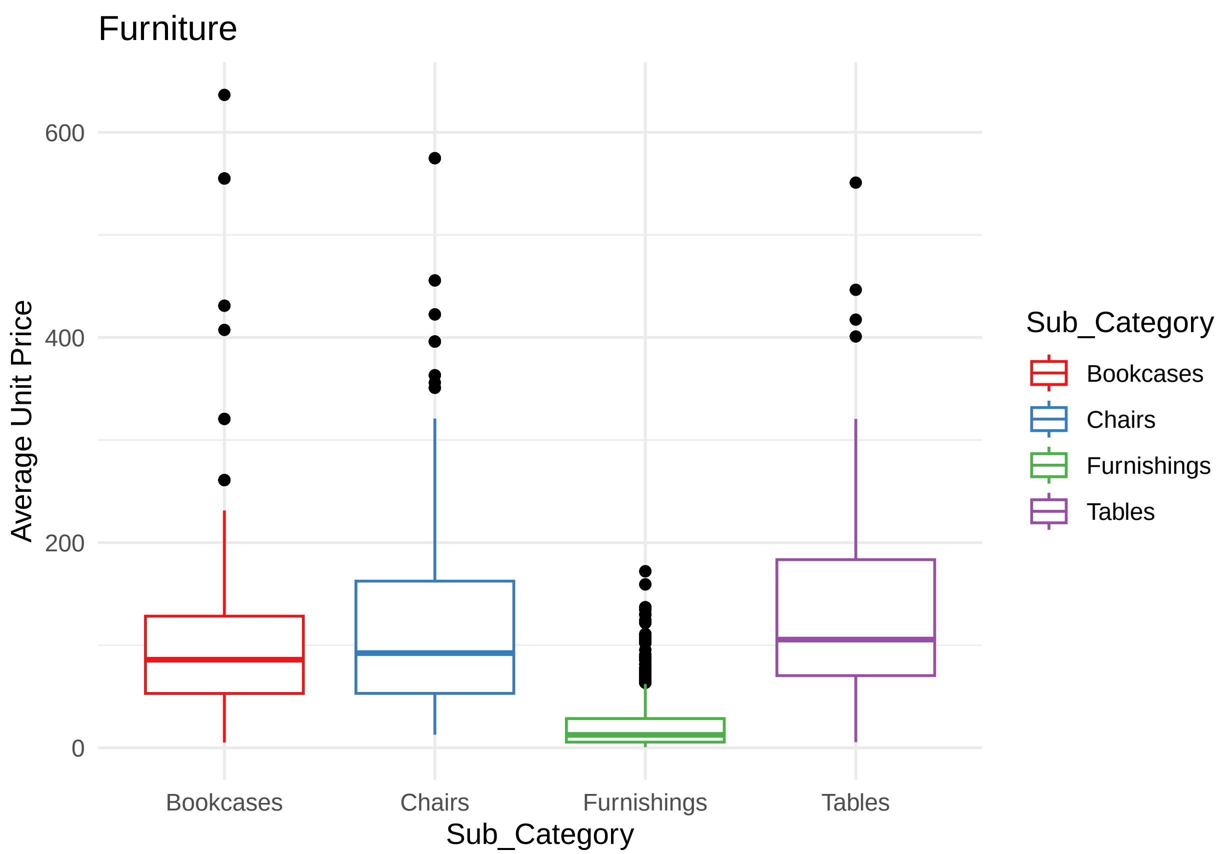
```
Orders_Processing %>%
    filter(Category == "Office Supplies") %>%
    ggplot(aes(x = Sub_Category, y = Average_Unit_Price, color = Sub_Category)) +
    geom_boxplot(outlier.color = "black") +
    ggtitle("Office Supplies") +
    xlab("Sub_Category") +
    ylab("Average Unit Price") +
    scale_color_brewer(palette = "Set1") +
    theme_minimal()
```



This Category encompasses the greatest number of sub-categories. Observing the dots in the plot reveals the presence of outliers in every sub-category. Notably, the sub-category with the highest outliers is 'Binders', followed closely by 'Supplies'. Additionally, it's worth noting that the median value across all sub-categories remains below 250, indicating a general trend toward lower median prices within these sub-categories.

3.1.3FURNITURE

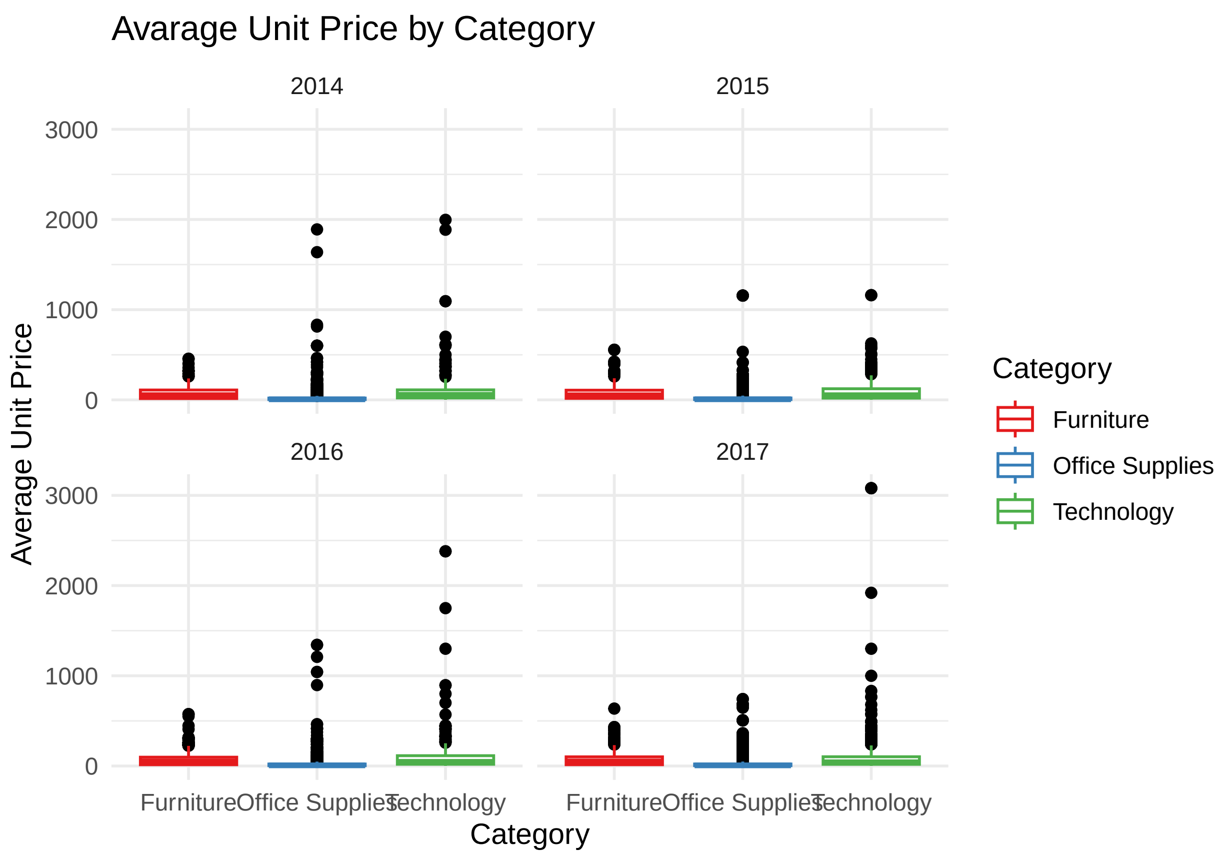
```
Orders_Processing %>%
    filter(Category == "Furniture") %>%
    ggplot(aes(x = Sub_Category, y = Average_Unit_Price, color = Sub_Category)) +
    geom_boxplot(outlier.color = "black") +
    ggtitle("Furniture") +
    xlab("Sub_Category") +
    ylab("Average Unit Price") +
    scale_color_brewer(palette = "Set1") +
    theme_minimal()
```



In 2014 and 2015, outliers were closer to the median, while 2016 and 2017 saw a wider spread, with 2017 featuring a notable product priced above 3000. Post-2014, Office Supplies' prices began aligning more closely with the median, indicating a trend towards price stabilization in this category.

3.1.4PER YEAR

```
ggplot(Orders_Processing, aes(x=Category, y= Average_Unit_Price, color = Category)) +
geom_boxplot(outlier.color = "black") +
facet_wrap(~ Year, scales = "fixed", nrow = 3,) +
ggtitle("Avarage Unit Price by Category") +
xlab("Category") +
ylab("Average Unit Price") +
scale_color_brewer(palette = "Set1") +
theme_minimal()
```



In 2014 and 2015, outliers were closer to the median, while 2016 and 2017 saw a wider spread, with 2017 featuring a notable product priced above 3000. Post-2014, Office Supplies' prices began aligning more closely with the median, indicating a trend towards price stabilization in this category.

3.2 PRICE CATEGORIES

3.2.1 CLASSIFYING THE PRODUCTS

```
# Creating the Classification
data_classification <- Orders_Processing %>%
    mutate(Price_Range_Category = case_when(
        Average_Unit_Price > 1000 ~ "Above 1000",
        Average_Unit_Price > 500 & Average_Unit_Price <= 1000 ~ "500-1000",
        Average_Unit_Price > 400 & Average_Unit_Price <= 500 ~ "400-500",
        Average_Unit_Price > 300 & Average_Unit_Price <= 400 ~ "300-400",
        Average_Unit_Price > 250 & Average_Unit_Price <= 300 ~ "250-300",
        Average_Unit_Price > 200 & Average_Unit_Price <= 250 ~ "200-250",
        Average_Unit_Price > 150 & Average_Unit_Price <= 200 ~ "150-200",
        Average_Unit_Price > 100 & Average_Unit_Price <= 150 ~ "100-150",
        Average_Unit_Price > 50 & Average_Unit_Price <= 100 ~ "50-100",
        Average_Unit_Price > 25 & Average_Unit_Price <= 50 ~ "25-50",
        Average_Unit_Price > 10 & Average_Unit_Price <= 25 ~ "10-25",
        Average_Unit_Price > 5 & Average_Unit_Price <= 10 ~ "5-10",
        Average_Unit_Price > 0 & Average_Unit_Price <= 5 ~ "0-5",
```

```
        TRUE ~ "Other" # Catch-all for any unexpected cases
    ))
```

- We decide to do this classification to smooth out the data because in every category and sub category price are significantly different. There are sub-categories which have products that cost less than 10 and more than 500 so this makes them different in terms of quantity, revenue and profit.

3.2.2 Calculating Yearly Statistics for Each Price Range Classification

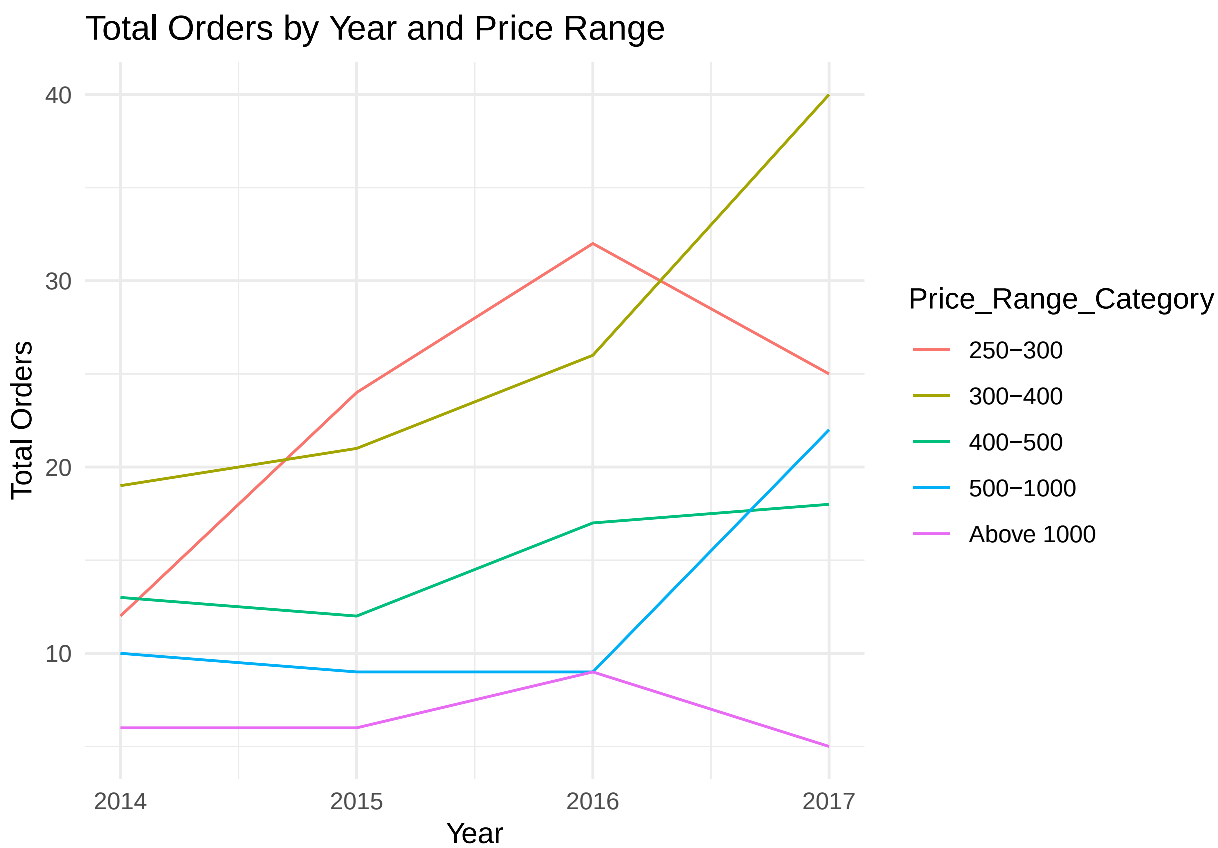
```
aggregated_data <- data_classification %>%
    group_by(Year, Price_Range_Category) %>%
    summarise(
        Total_Orders = n(), # Count total number of orders
        Total_Gross_Revenue = sum(Sales, na.rm = TRUE), # Sum total sales
        Total_Net_Revenue = sum(Net_Sales, na.rm = TRUE), # Sum total net sales
        Total_Profit = sum(Profit, na.rm = TRUE), # Sum total profit
        Total_Products = n_distinct(Product_ID), # Count unique products
        Total_Number_of_Returns = sum(Returned == "Yes", na.rm = TRUE), # Count total number of returns
        Order_After_Returns = n() - sum(Returned == "Yes", na.rm = TRUE), # Count total number of order
        Gross_Revenue_After_Returns = sum(Sales, na.rm = TRUE) - sum(Sales[Returned == "Yes"], na.rm = 
        Net_Revenue_After_Returns = sum(Net_Sales, na.rm = TRUE) - sum(Net_Sales[Returned == "Yes"], na
        Profit_After_Returns = sum(Profit, na.rm = TRUE) - sum(Profit[Returned == "Yes"], na.rm = TRUE)
        Avg_Unit_Price_Category = mean(Average_Unit_Price, na.rm = TRUE), # Average unit price
        Avg_Discount = mean(Discount, na.rm = TRUE), # Average discount
        Avg_Quantity = mean(Quantity, na.rm = TRUE), # Average quantity
        .groups = "drop"
    )

head(aggregated_data)
```

```
## # A tibble: 6 x 15
##    Year Price_Range_Category Total_Orders Total_Gross_Revenue Total_Net_Revenue
##   <dbl> <chr>                       <int>               <dbl>             <dbl>
## 1  2014 0-5                           546               7453.             5946.
## 2  2014 10-25                         365              26054.            22149.
## 3  2014 100-150                       122              67201.            58172.
## 4  2014 150-200                        72              53176.            43593.
## 5  2014 200-250                        45              46852.            38878.
## 6  2014 25-50                         240              38567.            33374.
## # i 10 more variables: Total_Profit <dbl>, Total_Products <int>,
## #   Total_Number_of_Returns <int>, Order_After_Returns <int>,
## #   Gross_Revenue_After_Returns <dbl>, Net_Revenue_After_Returns <dbl>,
## #   Profit_After_Returns <dbl>, Avg_Unit_Price_Category <dbl>,
## #   Avg_Discount <dbl>, Avg_Quantity <dbl>
```

3.2.3 Total Orders During the years for Price Ranges Between Above 1000 and 250-300

```
aggregated_data %>%
    filter(Price_Range_Category == "Above 1000" | Price_Range_Category == "500-1000" | Price_Range_Cate
```

```
    ggplot(aes(x = Year, y = Total_Orders, group = Price_Range_Category, color = Price_Range_Category))
    geom_line() +
    theme_minimal() +
    labs(title = "Total Orders by Year and Price Range", x = "Year", y = "Total Orders")
```

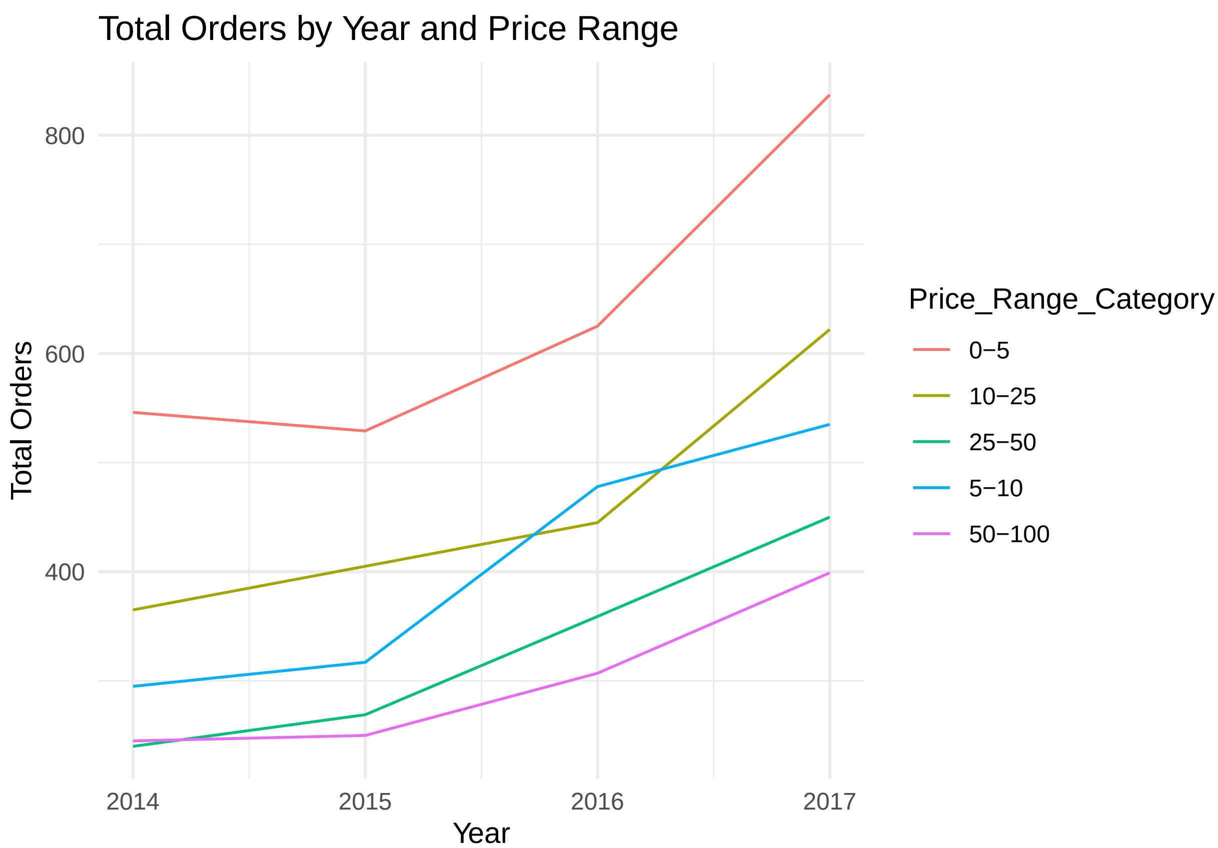


3.2.4Total Orders During the years for Price Ranges Between 200-250 and 100-150

```
aggregated_data %>%
    filter(Price_Range_Category == "200-250" | Price_Range_Category == "150-200" | Price_Range_Category
    ggplot(aes(x = Year, y = Total_Orders, group = Price_Range_Category, color = Price_Range_Category))
    geom_line() +
    theme_minimal() +
    labs(title = "Total Orders by Year and Price Range", x = "Year", y = "Total Orders")
```

3.2.5Total Orders During the years for Price Ranges Between 50-100 and 0-5

```
aggregated_data %>%
    filter(Price_Range_Category == "50-100" | Price_Range_Category == "25-50" | Price_Range_Category ==
    ggplot(aes(x = Year, y = Total_Orders, group = Price_Range_Category, color = Price_Range_Category))
    geom_line() +
    theme_minimal() +
    labs(title = "Total Orders by Year and Price Range", x = "Year", y = "Total Orders")
```



3.3 Total Number of Orders and Returns

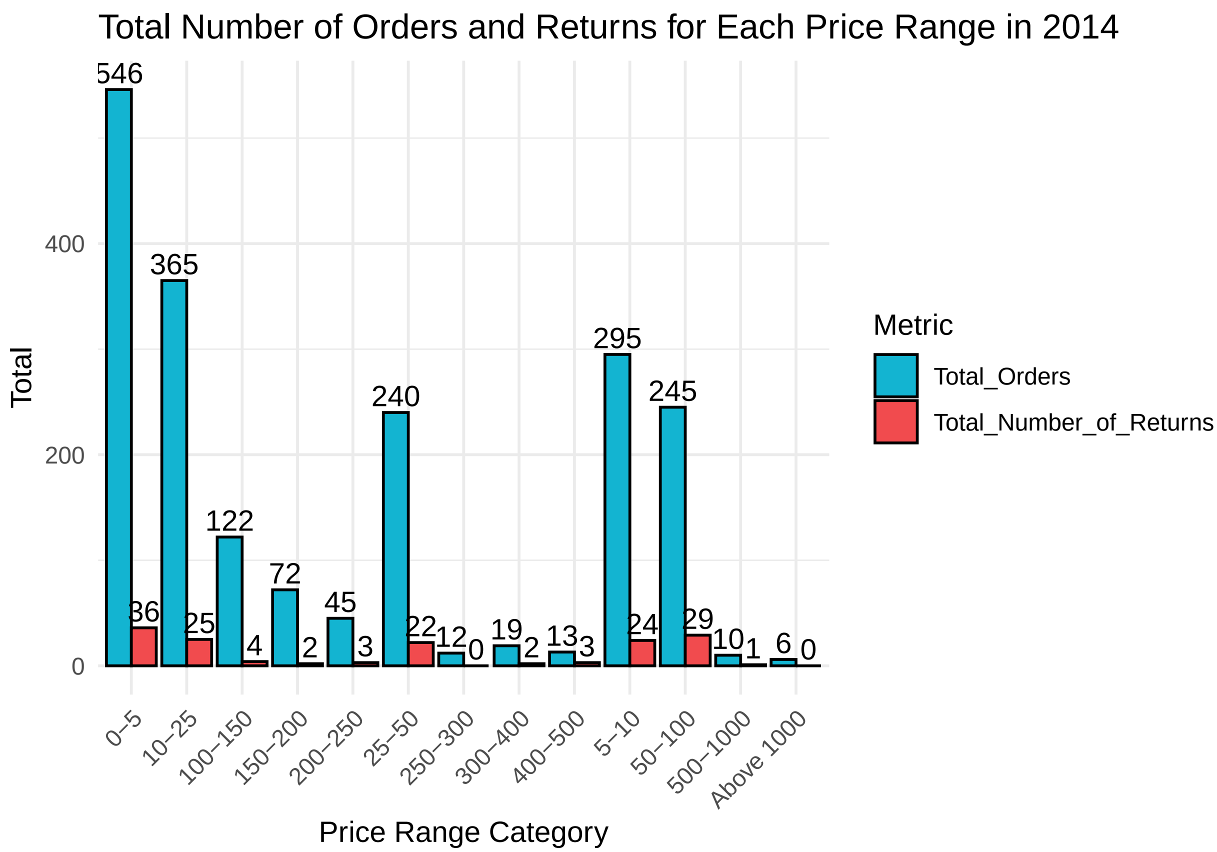
3.3.1 Reshaping the data

```
long_data_orders_returns <- aggregated_data %>%
    pivot_longer(
        cols = c(Total_Orders, Total_Number_of_Returns),
        names_to = "Metric",
        values_to = "Value"
    ) %>%
    mutate(Metric = factor(Metric, levels = c("Total_Orders", "Total_Number_of_Returns")))
```

3.3.2 2014

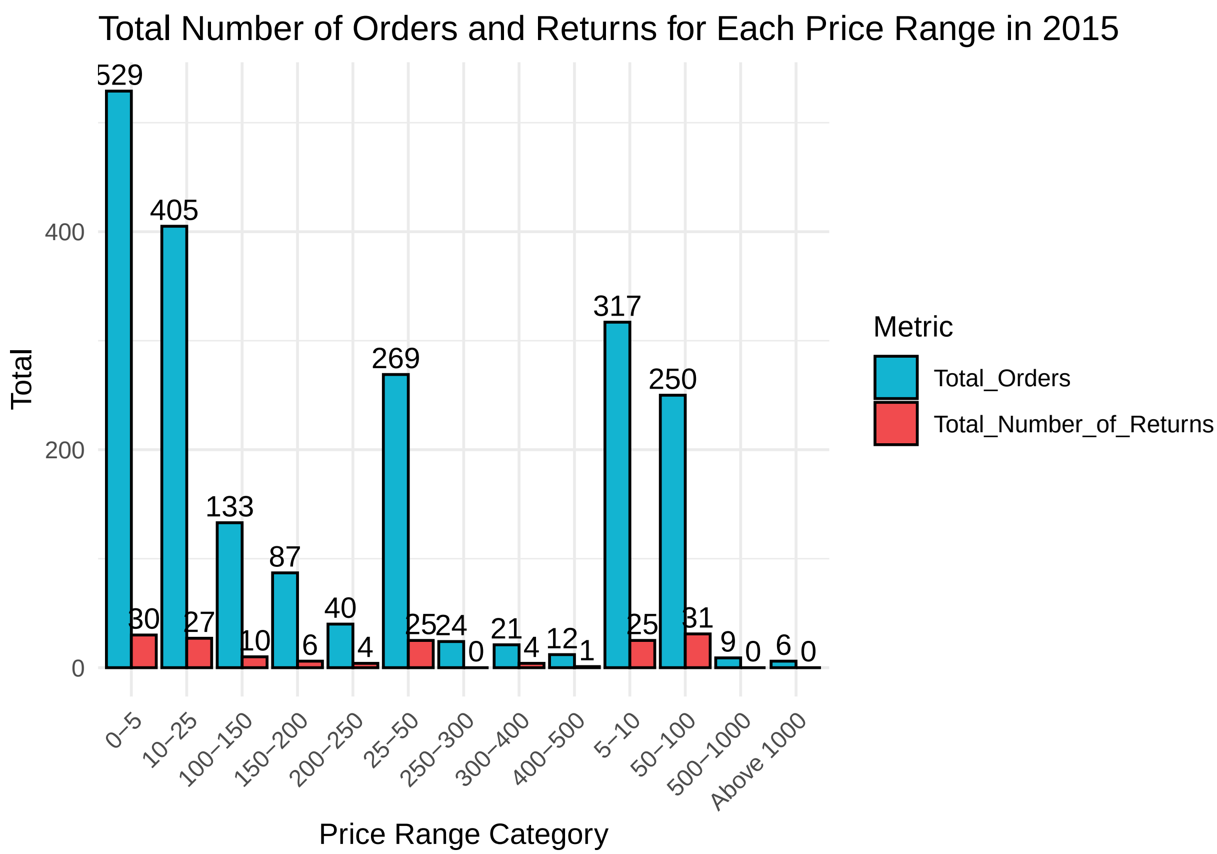
```
ggplot(long_data_orders_returns %>% filter(Year == 2014), aes(x = Price_Range_Category, y = Value, fill
  geom_bar(stat = "identity", position = position_dodge(width = 0.9), color = "black") +
  geom_text(aes(label = Value), vjust = -0.3, position = position_dodge(width = 0.9)) +
  scale_fill_manual(values = c("Total_Orders" = "#13b4d1", "Total_Number_of_Returns" = "#f14b4e")) +
  theme_minimal() +
  labs(
    title = "Total Number of Orders and Returns for Each Price Range in 2014",
    x = "Price Range Category",
```

```r
    y = "Total"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



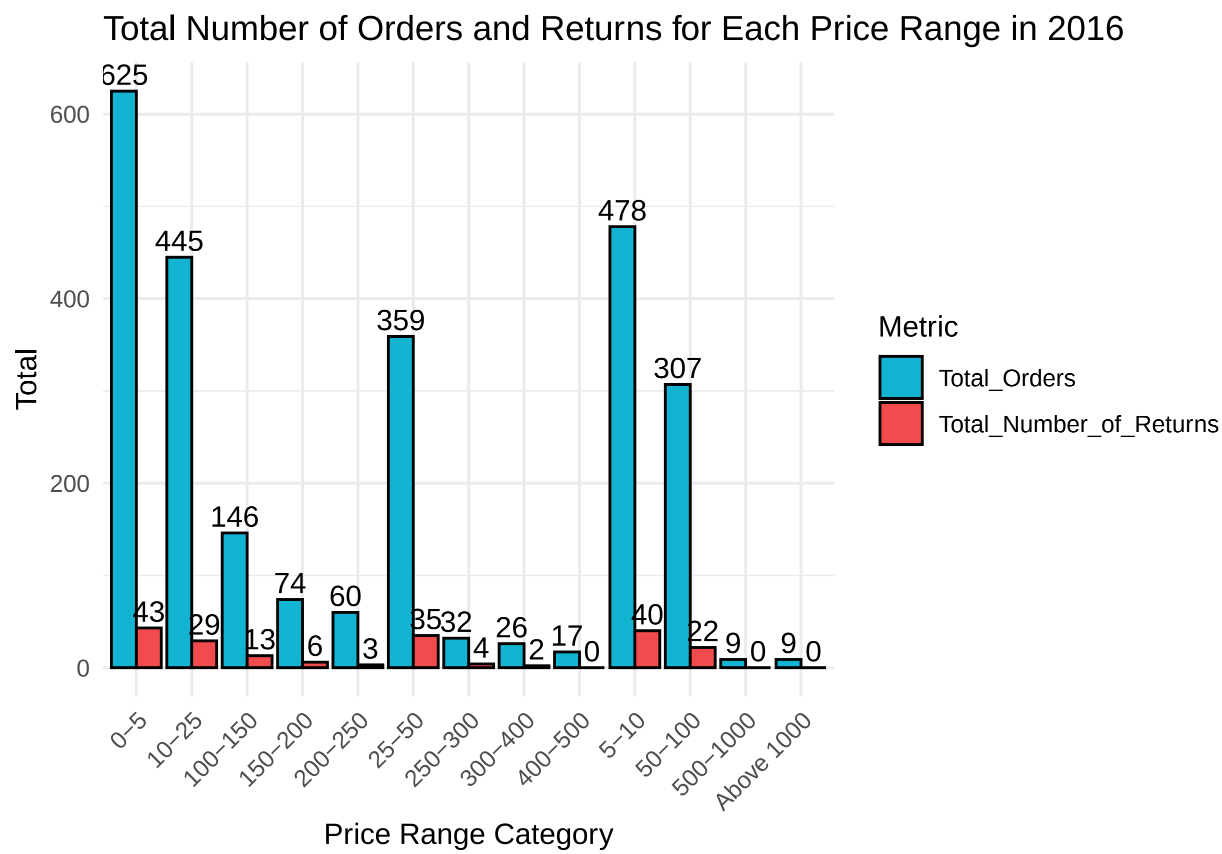
3.3.3 2015

```r
ggplot(long_data_orders_returns %>% filter(Year == 2015), aes(x = Price_Range_Category, y = Value, fill
  geom_bar(stat = "identity", position = position_dodge(width = 0.9), color = "black") +
  geom_text(aes(label = Value), vjust = -0.3, position = position_dodge(width = 0.9)) +
  scale_fill_manual(values = c("Total_Orders" = "#13b4d1", "Total_Number_of_Returns" = "#f14b4e")) +
  theme_minimal() +
  labs(
    title = "Total Number of Orders and Returns for Each Price Range in 2015",
    x = "Price Range Category",
    y = "Total"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



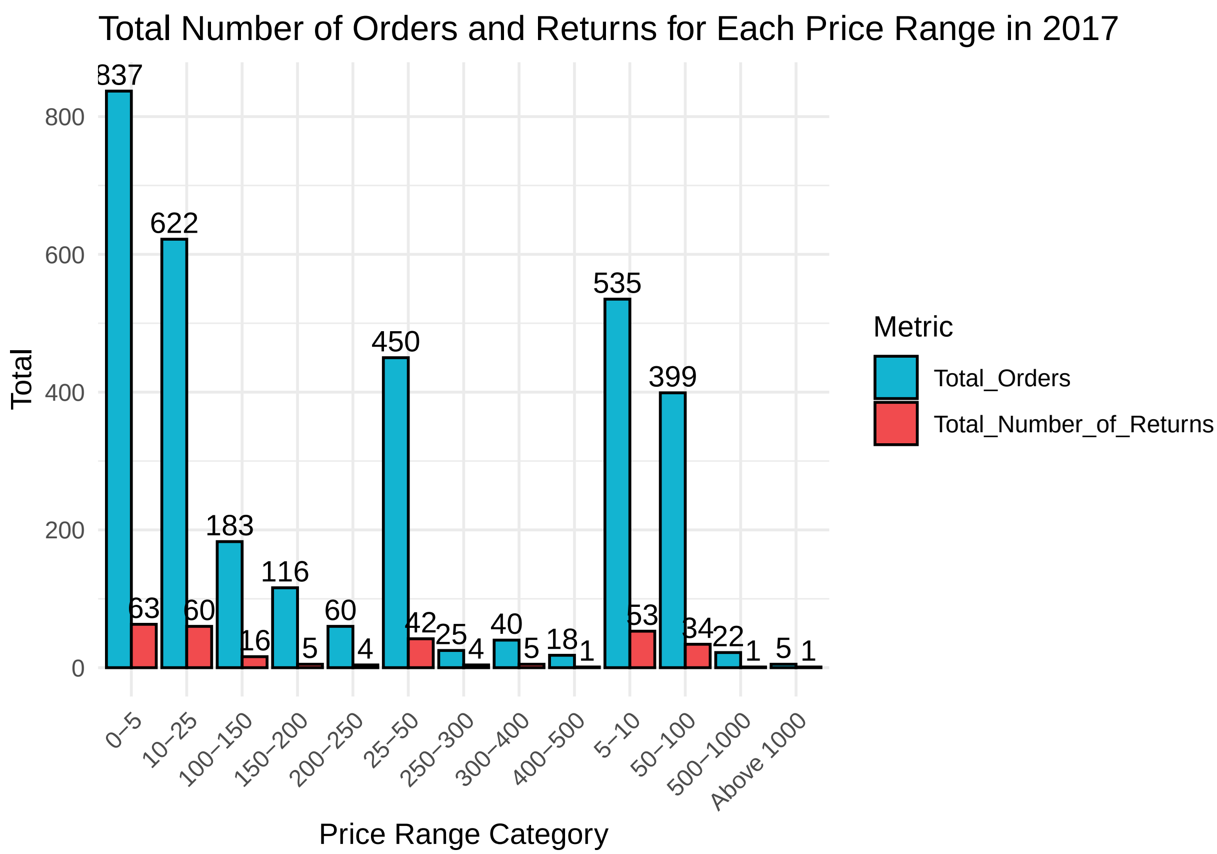
3.3.4 2016

```
ggplot(long_data_orders_returns %>% filter(Year == 2016), aes(x = Price_Range_Category, y = Value, fill
  geom_bar(stat = "identity", position = position_dodge(width = 0.9), color = "black") +
  geom_text(aes(label = Value), vjust = -0.3, position = position_dodge(width = 0.9)) +
  scale_fill_manual(values = c("Total_Orders" = "#13b4d1", "Total_Number_of_Returns" = "#f14b4e")) +
  theme_minimal() +
  labs(
    title = "Total Number of Orders and Returns for Each Price Range in 2016",
    x = "Price Range Category",
    y = "Total"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



3.3.4 2017

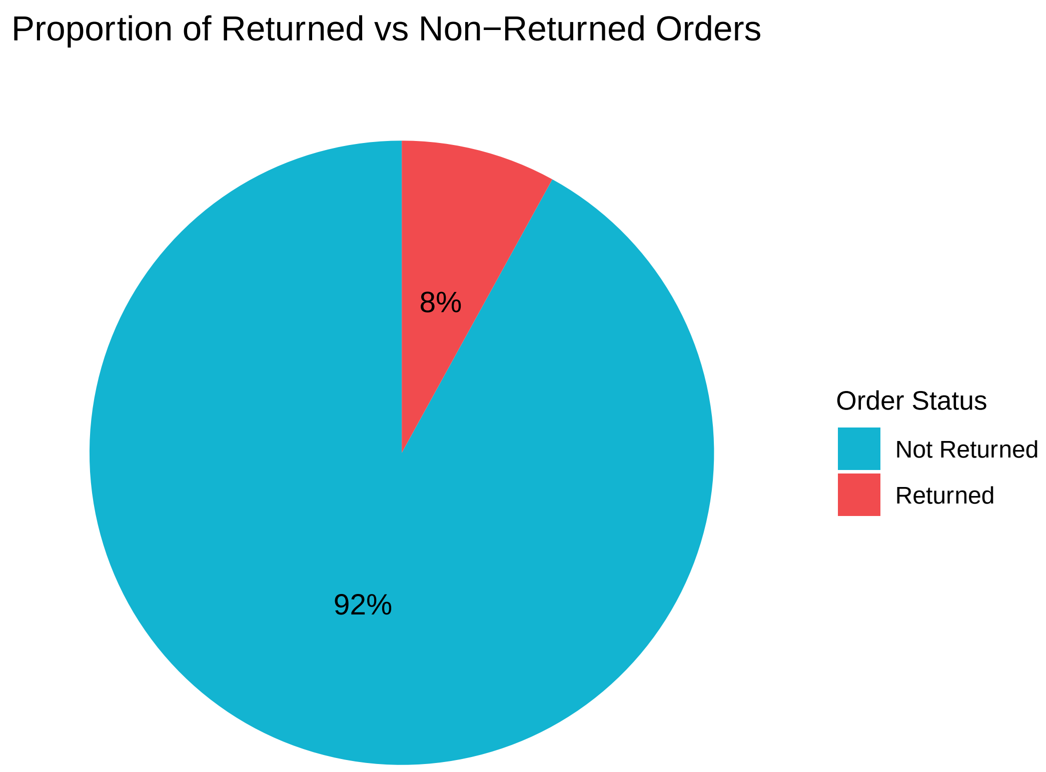
```
ggplot(long_data_orders_returns %>% filter(Year == 2017), aes(x = Price_Range_Category, y = Value, fill
  geom_bar(stat = "identity", position = position_dodge(width = 0.9), color = "black") +
  geom_text(aes(label = Value), vjust = -0.3, position = position_dodge(width = 0.9)) +
  scale_fill_manual(values = c("Total_Orders" = "#13b4d1", "Total_Number_of_Returns" = "#f14b4e")) +
  theme_minimal() +
  labs(
    title = "Total Number of Orders and Returns for Each Price Range in 2017",
    x = "Price Range Category",
    y = "Total"
  ) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



3.3.5 Percentages of Non Returned vs Returned

```
order_summary <- Orders_Processing %>%
    group_by(Returned) %>%
    summarise(Count = n(), .groups = "drop") %>%
    mutate(Percentage = Count / sum(Count) * 100)

ggplot(order_summary, aes(x = "", y = Count, fill = factor(Returned))) +
    geom_bar(width = 1, stat = "identity") +
    coord_polar(theta = "y") +
    geom_text(aes(label = paste0(round(Percentage, 1), "%")),
        position = position_stack(vjust = 0.5)
    ) +
    scale_fill_manual(
        values = c("#13b4d1", "#f14b4e"),
        labels = c("Not Returned", "Returned"),
        name = "Order Status"
    ) +
    labs(
        title = "Proportion of Returned vs Non-Returned Orders",
        x = NULL,
        y = NULL
    ) +
    theme_minimal() +
```

```
    theme(
        axis.line = element_blank(),
        axis.text = element_blank(),
        axis.ticks = element_blank(),
        panel.grid = element_blank(),
        legend.title = element_text(size = 10)
    )
```

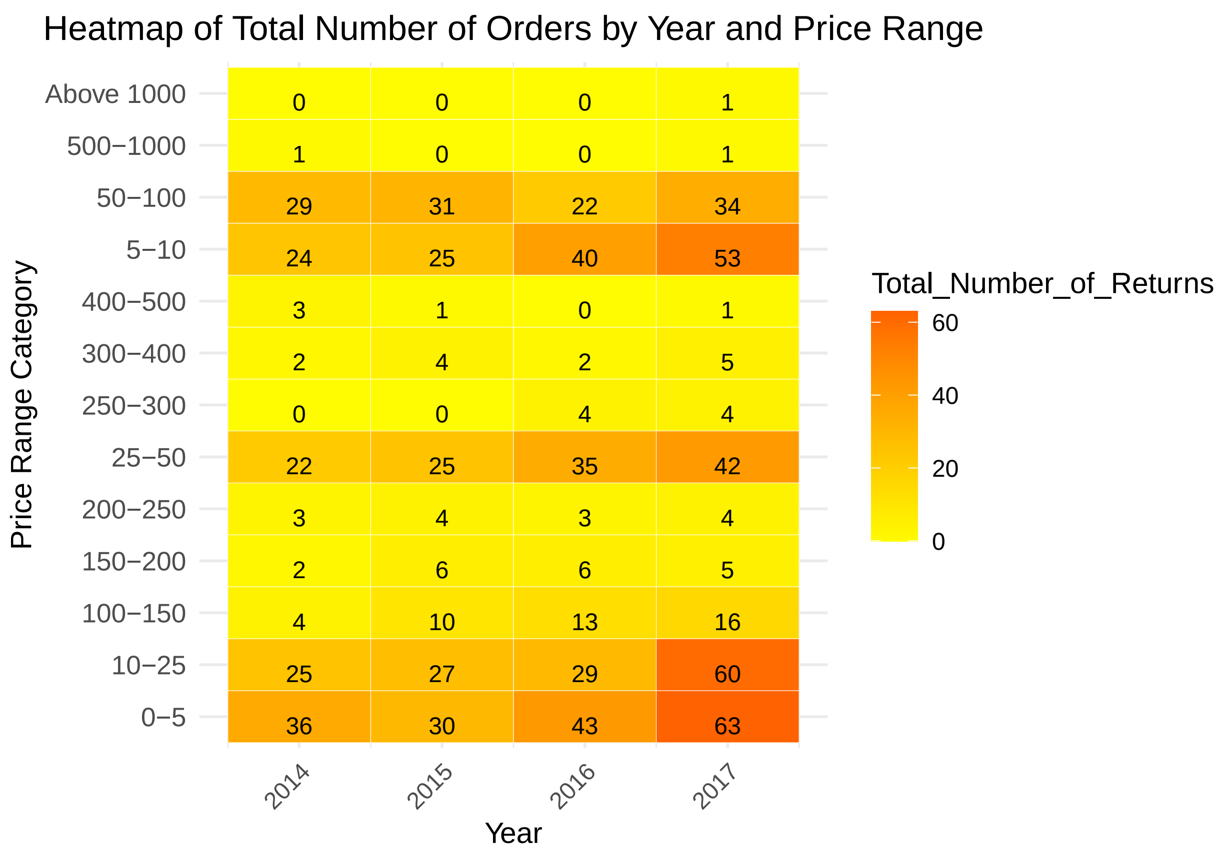


3.3.6 Heatmap of Year, Price Range and Total_Number of Orders

```
ggplot(aggregated_data, aes(x = Year, y = Price_Range_Category, fill = Total_Orders)) +
    geom_tile(color = "white") +
    geom_text(aes(label = Total_Orders), color = "black", size = 3, vjust = 1) +
    scale_fill_gradient2(low = "#00ff9d", high = "#0011ff", mid = "#117dd0", midpoint = median(aggregate
    theme_minimal() +
    theme(
        axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1),
        axis.text.y = element_text(size = 10),
        plot.title = element_text(hjust = 0.5)
    ) +
    labs(title = "Heatmap of Total Number of Orders by Year and Price Range", x = "Year", y = "Price Ra
```

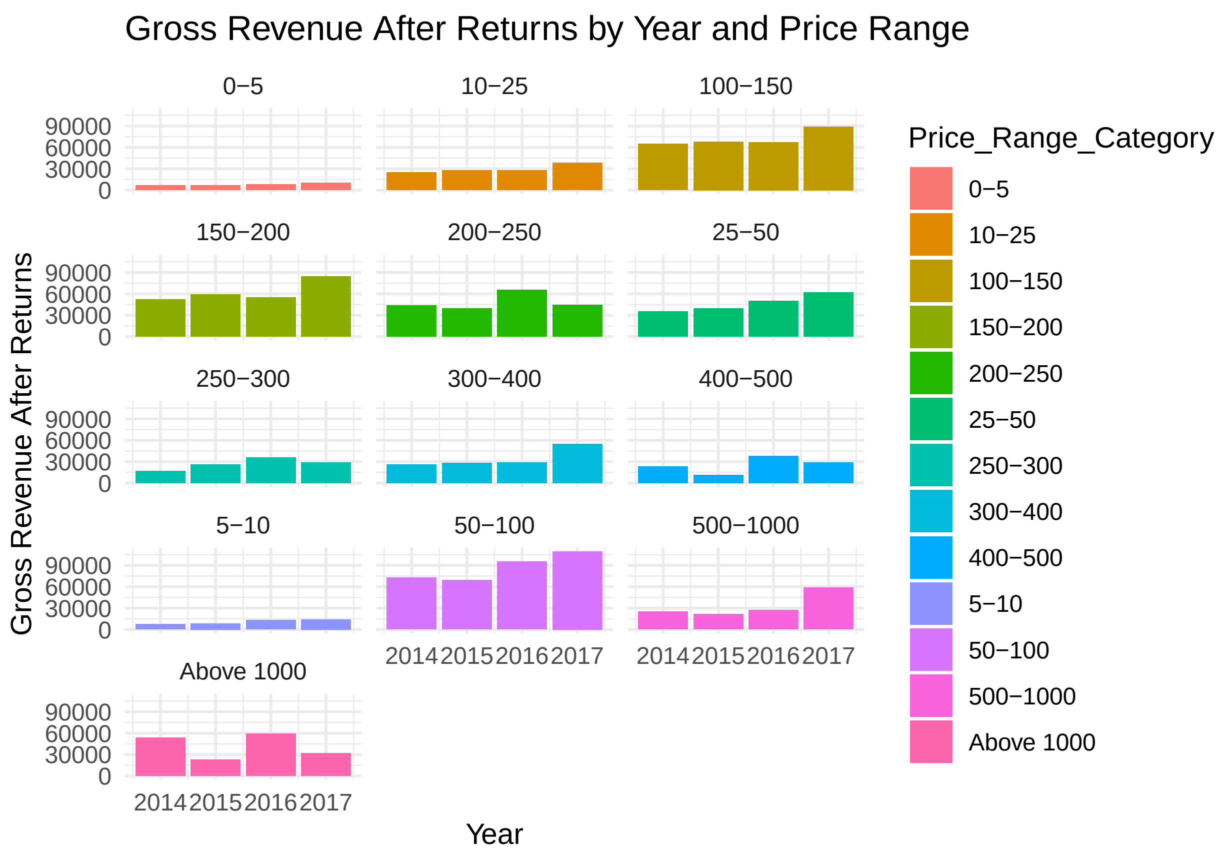
3.3.7 Heatmap of Year, Price Range and Total_Number of Returns

```
ggplot(aggregated_data, aes(x = Year, y = Price_Range_Category, fill = Total_Number_of_Returns)) +
    geom_tile(color = "white") +
    geom_text(aes(label = Total_Number_of_Returns), color = "black", size = 3, vjust = 1) +
    scale_fill_gradient2(low = "#fffb00", high = "red", mid = "red", midpoint = median(aggregated_data$T
    theme_minimal() +
    theme(
        axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1),
        axis.text.y = element_text(size = 10),
        plot.title = element_text(hjust = 0.5)
    ) +
    labs(title = "Heatmap of Total Number of Orders by Year and Price Range", x = "Year", y = "Price Ra
```



3.3.8 Gross_Revenue_After Returns

```
ggplot(aggregated_data, aes(x = Year, y = Gross_Revenue_After_Returns, fill = Price_Range_Category)) +
    geom_bar(stat = "identity", position = "dodge") +
    facet_wrap(~Price_Range_Category, ncol = 3) +
    theme_minimal() +
    labs(title = "Gross Revenue After Returns by Year and Price Range", x = "Year", y = "Gross Revenue A
```



3.4 Net Revenue and Profit

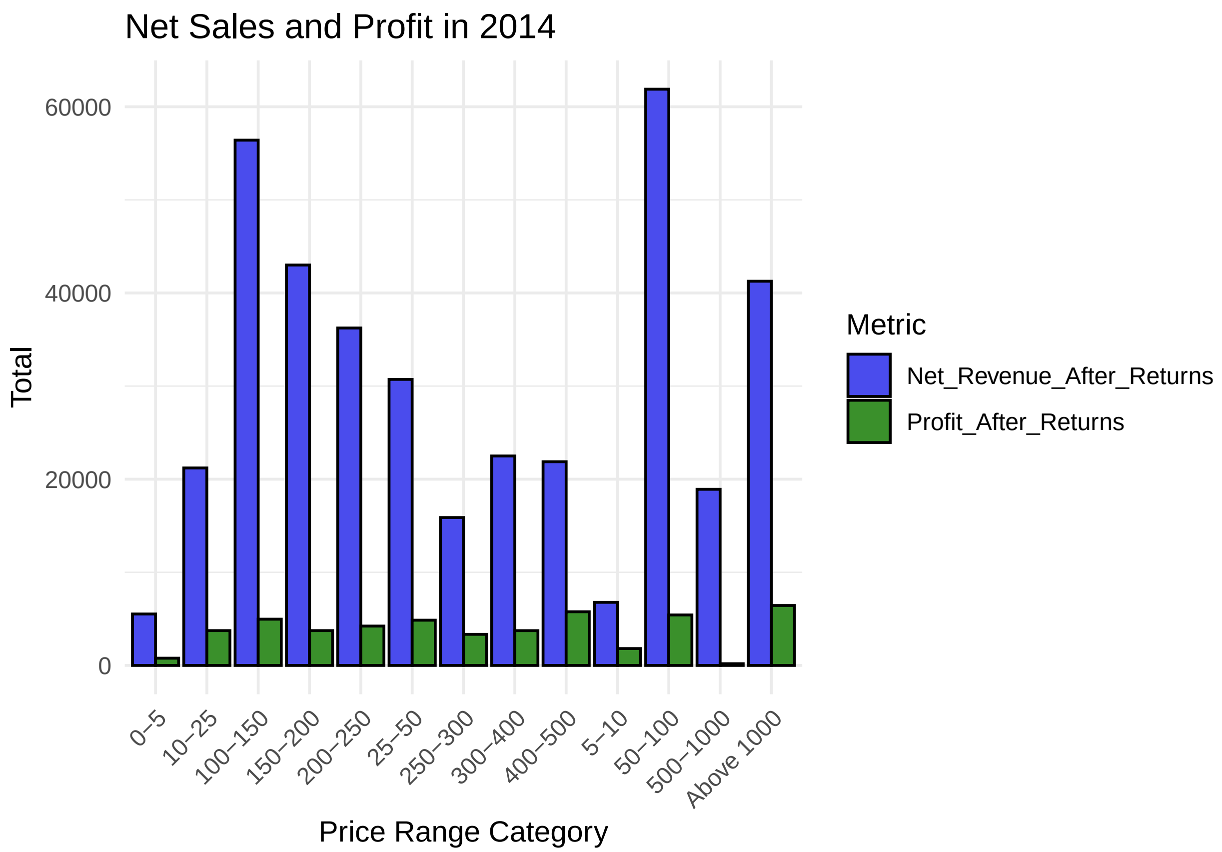
3.4.1 Reshaping the Data

```
long_data_netrev_profit <- aggregated_data %>%
    pivot_longer(
        cols = c(Net_Revenue_After_Returns, Profit_After_Returns),
        names_to = "Metric",
        values_to = "Value"
    ) %>%
    mutate(Metric = factor(Metric, levels = c("Net_Revenue_After_Returns", "Profit_After_Returns")))
```

3.4.2 2014

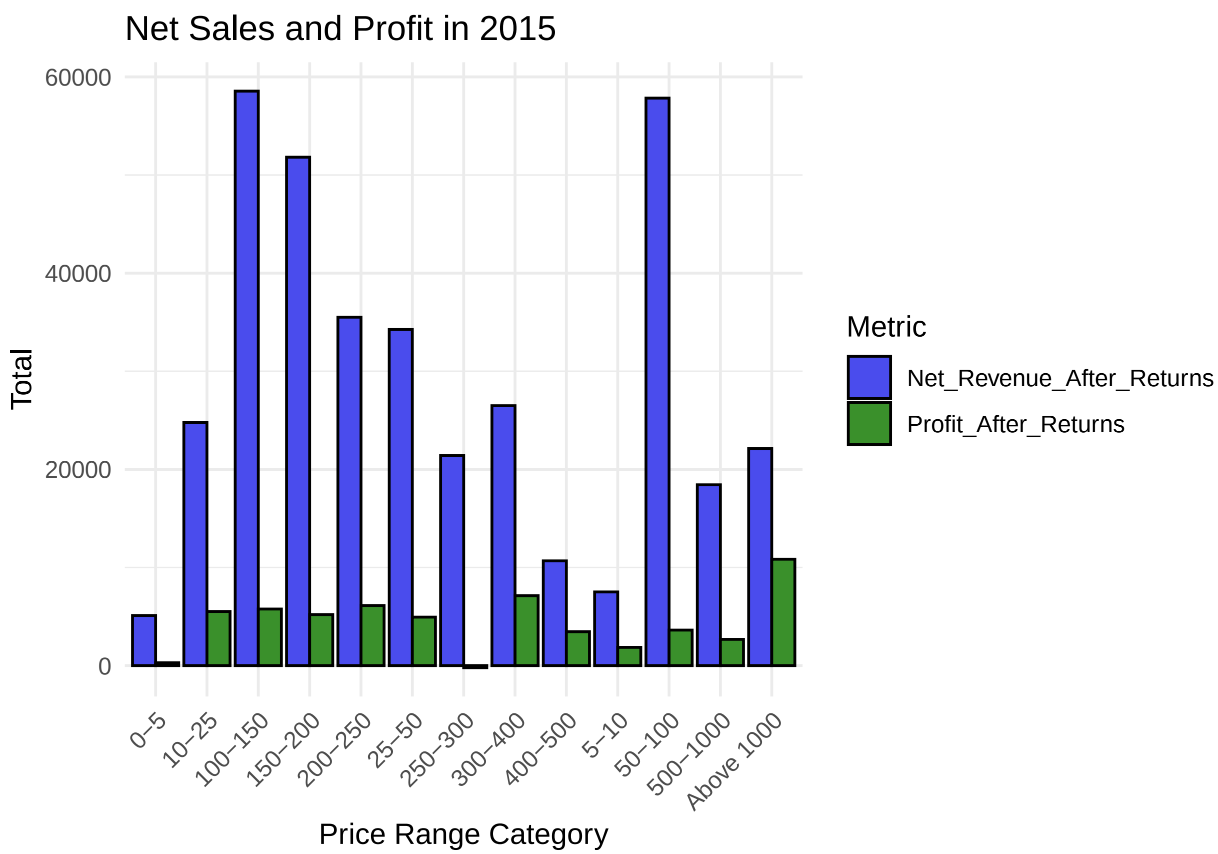
```
ggplot(long_data_netrev_profit %>% filter(Year == 2014), aes(x = Price_Range_Category, y = Value, fill =
    geom_bar(stat = "identity", position = position_dodge(width = 0.9), color = "black") +
    scale_fill_manual(values = c("Net_Revenue_After_Returns" = "#4a4ced", "Profit_After_Returns" = "#3a9
    theme_minimal() +
    labs(
        title = "Net Sales and Profit in 2014",
        x = "Price Range Category",
        y = "Total"
```

```
    ) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



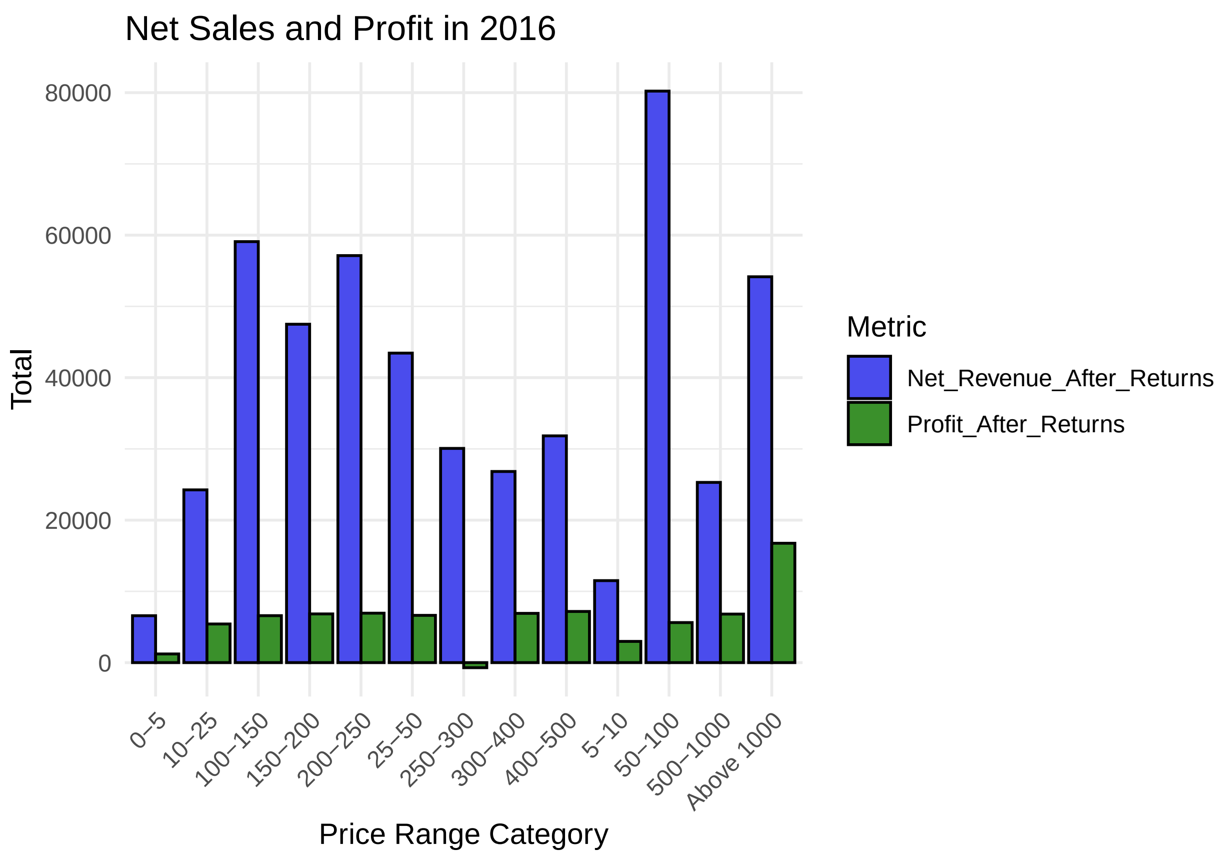
3.4.3 2015

```
ggplot(long_data_netrev_profit %>% filter(Year == 2015), aes(x = Price_Range_Category, y = Value, fill =
    geom_bar(stat = "identity", position = position_dodge(width = 0.9), color = "black") +
    scale_fill_manual(values = c("Net_Revenue_After_Returns" = "#4a4ced", "Profit_After_Returns" = "#3a9
    theme_minimal() +
    labs(
        title = "Net Sales and Profit in 2015",
        x = "Price Range Category",
        y = "Total"
    ) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



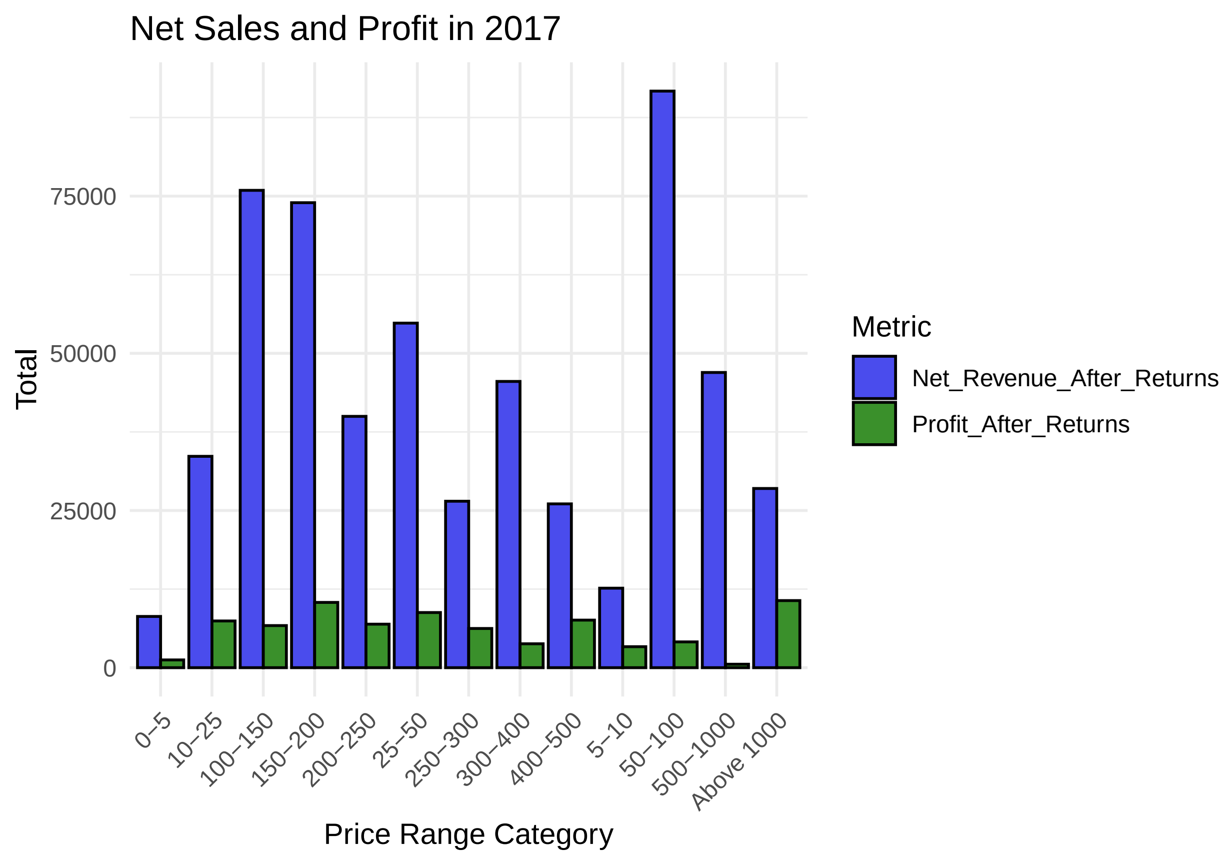
3.4.4 2016

```
ggplot(long_data_netrev_profit %>% filter(Year == 2016), aes(x = Price_Range_Category, y = Value, fill =
    geom_bar(stat = "identity", position = position_dodge(width = 0.9), color = "black") +
    scale_fill_manual(values = c("Net_Revenue_After_Returns" = "#4a4ced", "Profit_After_Returns" = "#3a9
    theme_minimal() +
    labs(
        title = "Net Sales and Profit in 2016",
        x = "Price Range Category",
        y = "Total"
    ) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



3.4.5 2017

```
ggplot(long_data_netrev_profit %>% filter(Year == 2017), aes(x = Price_Range_Category, y = Value, fill =
    geom_bar(stat = "identity", position = position_dodge(width = 0.9), color = "black") +
    scale_fill_manual(values = c("Net_Revenue_After_Returns" = "#4a4ced", "Profit_After_Returns" = "#3a9
    theme_minimal() +
    labs(
        title = "Net Sales and Profit in 2017",
        x = "Price Range Category",
        y = "Total"
    ) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

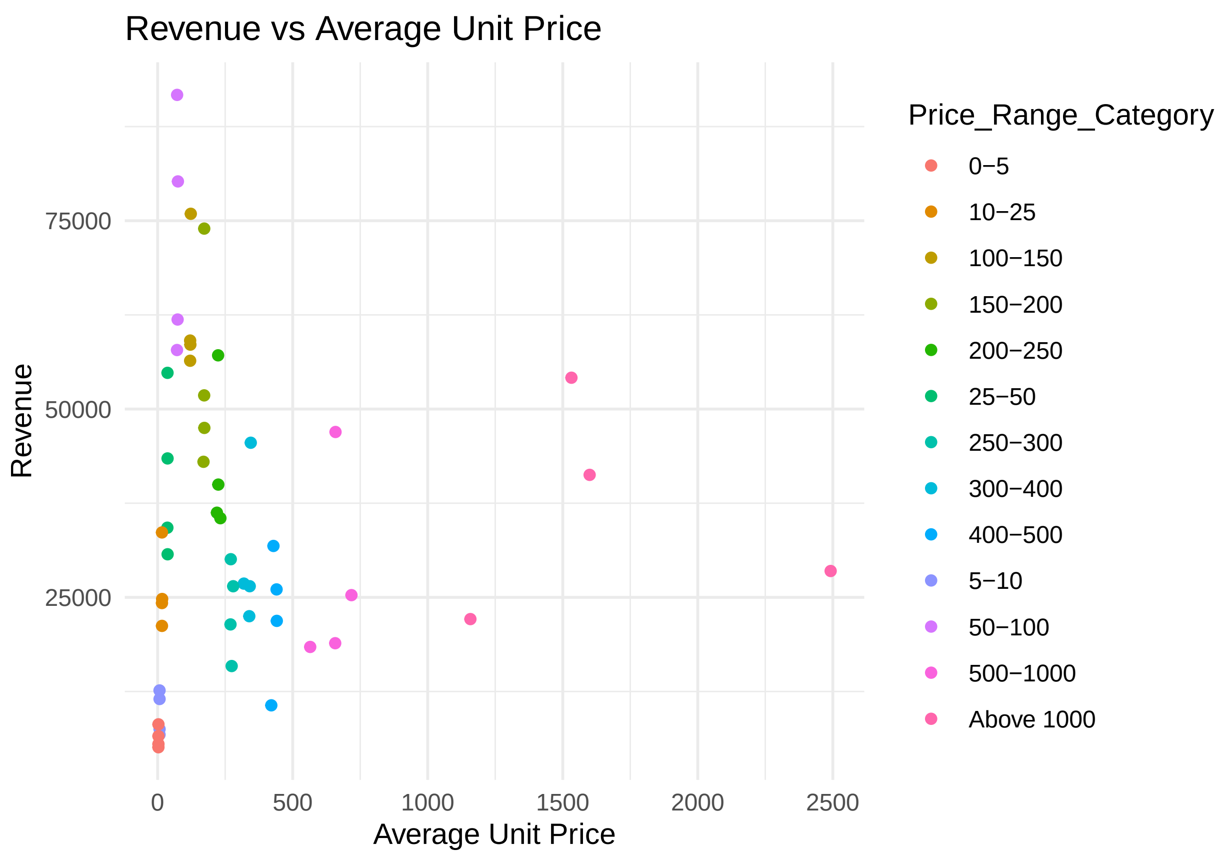


3.4.6 Total Orders vs Average Unit Price by Price Range During The Years after returns

```
ggplot(aggregated_data, aes(x = Avg_Unit_Price_Category, y = Order_After_Returns, color = Price_Range_Ca
    geom_point() +
    theme_minimal() +
    labs(title = "Total Orders vs Average Unit Price", x = "Average Unit Price", y = "Total Orders")
```

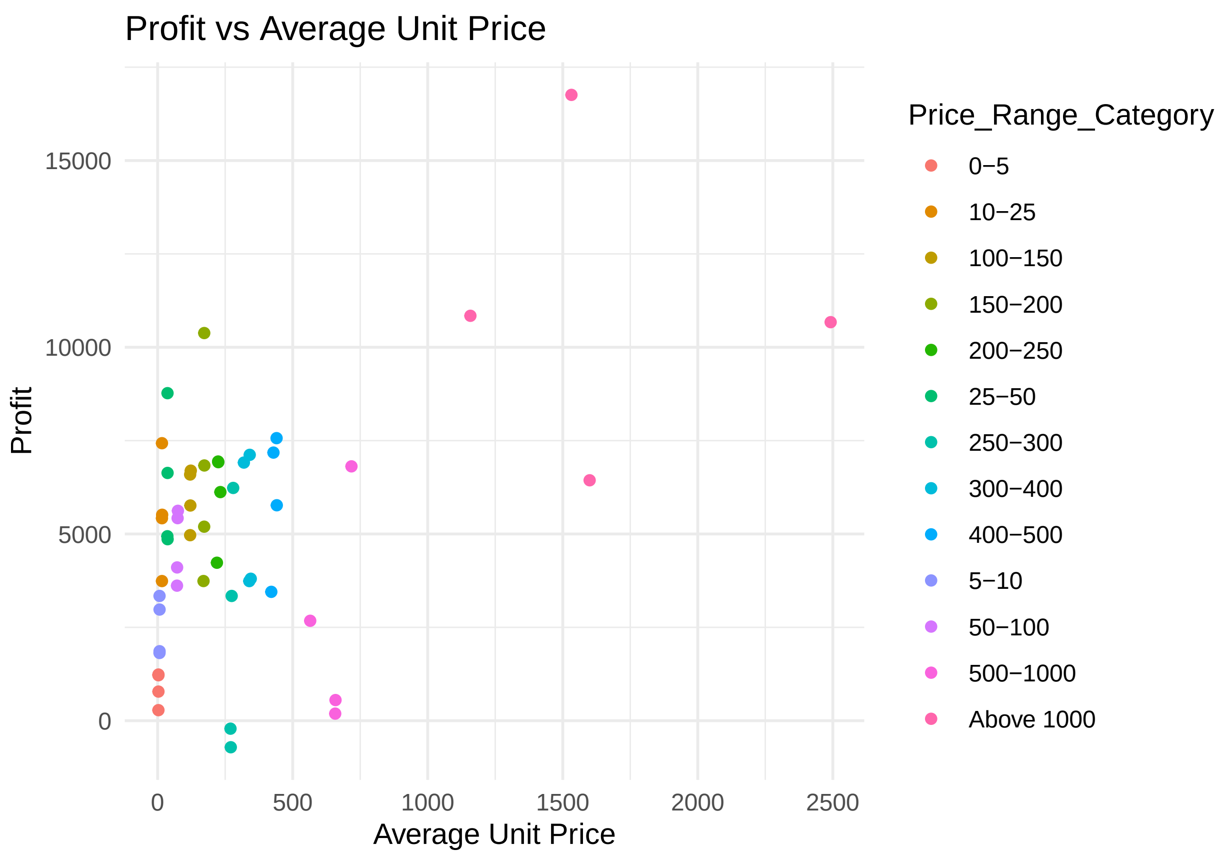
3.4.7 Total Revenue vs Average Unit Price by Price Range During The Years

```
ggplot(aggregated_data, aes(x = Avg_Unit_Price_Category, y = Net_Revenue_After_Returns, color = Price_Ra
    geom_point() +
    theme_minimal() +
    labs(title = "Revenue vs Average Unit Price", x = "Average Unit Price", y = "Revenue")
```



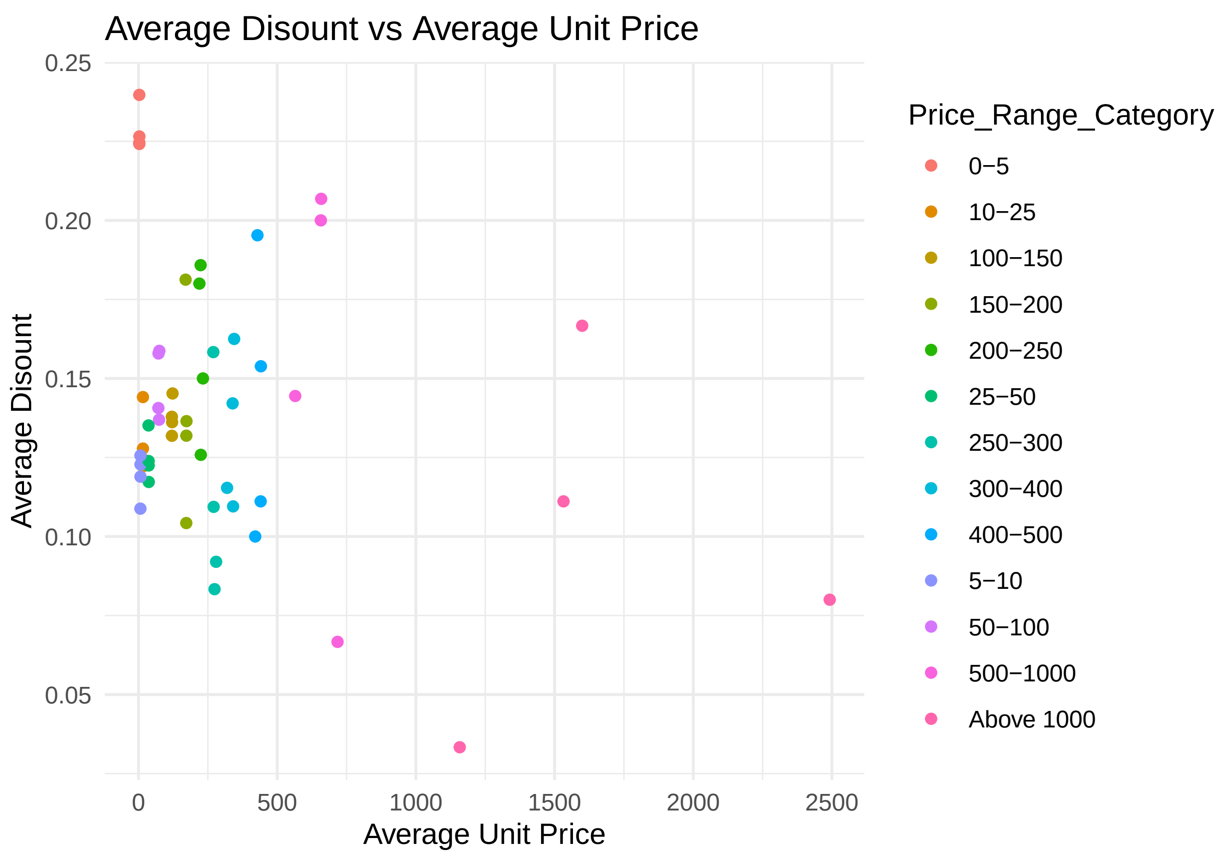
3.4.8 Total Profit vs Average Unit Price by Price Range During The Years

```
ggplot(aggregated_data, aes(x = Avg_Unit_Price_Category, y = Profit_After_Returns, color = Price_Range_(
    geom_point() +
    theme_minimal() +
    labs(title = "Profit vs Average Unit Price", x = "Average Unit Price", y = "Profit")
```



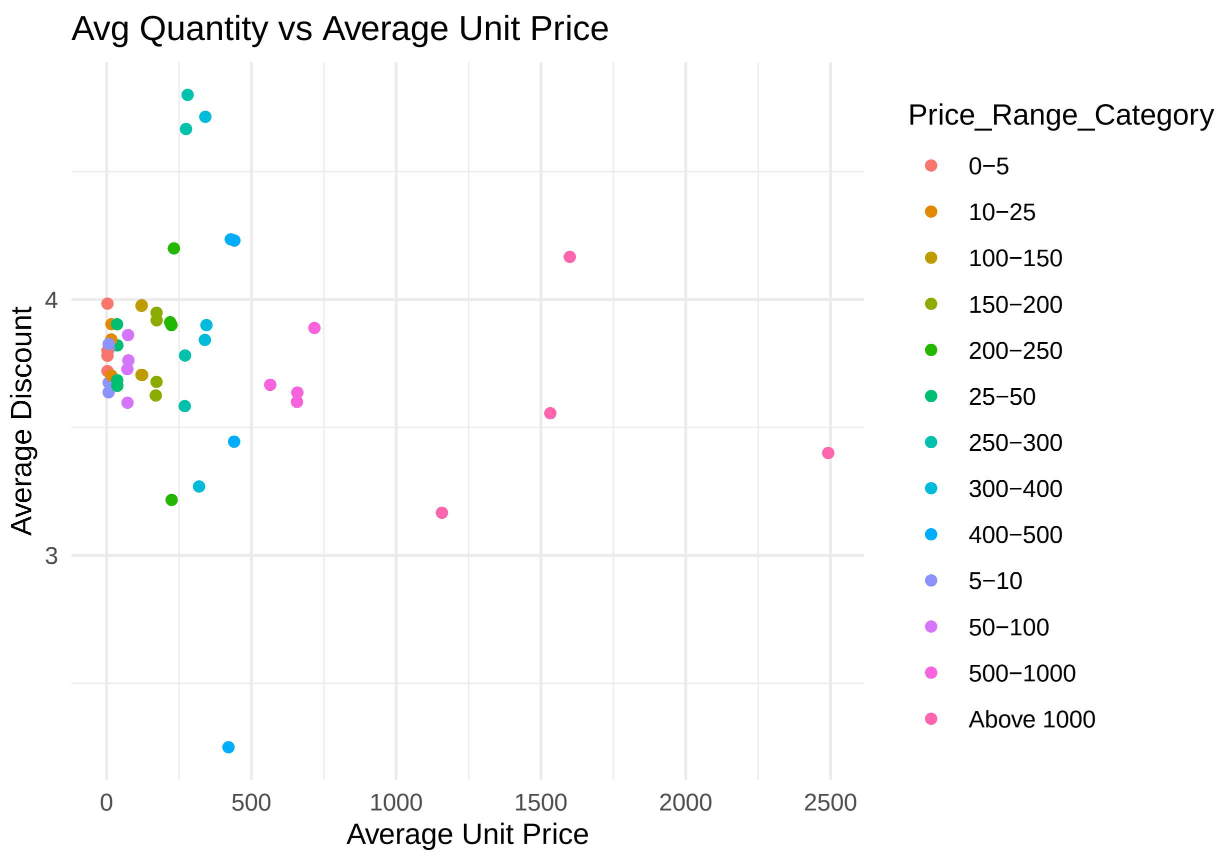
3.4.9 Avg_Discount vs Average Unit Price by Price Range During The Years

```
ggplot(aggregated_data, aes(x = Avg_Unit_Price_Category, y = Avg_Discount, color = Price_Range_Category)
    geom_point() +
    theme_minimal() +
    labs(title = "Average Disount vs Average Unit Price", x = "Average Unit Price", y = "Average Disoun
```



3.4.10 Avg_Quantity vs Average Unit Price by Price Range During The Years

```
ggplot(aggregated_data, aes(x = Avg_Unit_Price_Category, y = Avg_Quantity, color = Price_Range_Category)
    geom_point() +
    theme_minimal() +
    labs(title = "Avg Quantity vs Average Unit Price", x = "Average Unit Price", y = "Average Discount")
```



3.5Category and Sub_Category Analysis

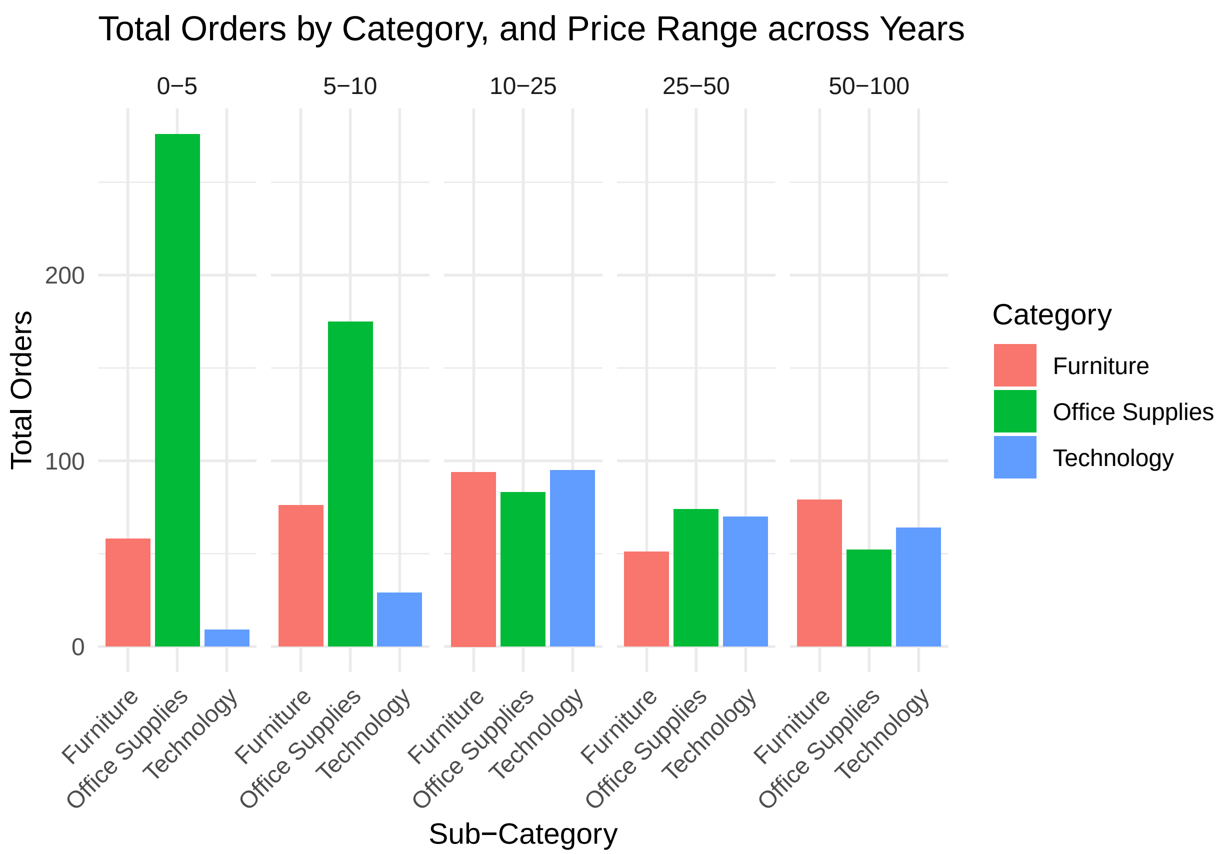
3.5.1Aggregating the data

```
aggregated_data_subcategory <- data_classification %>%
    mutate(Price_Range_Category = factor(Price_Range_Category,
        levels = c("0-5", "5-10", "10-25", "25-50", "50-100", "100-150", "150-200", "200-250", "250-300
        ordered = TRUE
    )) %>%
    group_by(Year, Price_Range_Category, Category, Sub_Category) %>%
    summarise(Total_Orders = n(), .groups = "drop")
```

3.5.2 Total orders for Low price ranges

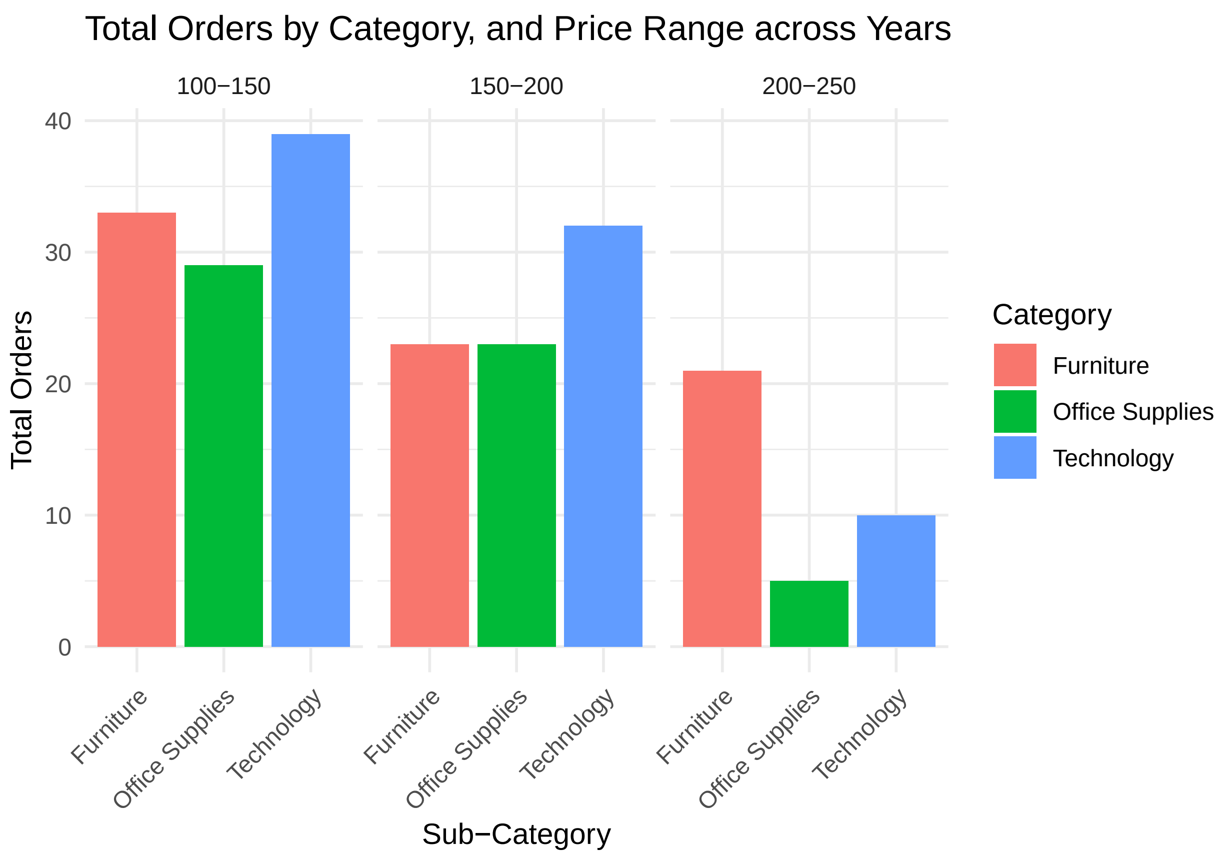
```
aggregated_data_subcategory %>%
    filter(Price_Range_Category == "50-100" | Price_Range_Category == "25-50" | Price_Range_Category ==
    ggplot(aes(x = Category, y = Total_Orders, fill = Category)) +
    geom_bar(stat = "identity", position = position_dodge()) +
    facet_wrap(~Price_Range_Category, scales = "free_x", ncol = 5) +
    theme_minimal() +
    labs(
```

```
        title = "Total Orders by Category, and Price Range across Years",
        x = "Sub-Category",
        y = "Total Orders"
    ) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



3.5.3 Middle Price Ranges

```
aggregated_data_subcategory %>%
    filter(Price_Range_Category == "200-250" | Price_Range_Category == "150-200" | Price_Range_Category
    ggplot(aes(x = Category, y = Total_Orders, fill = Category)) +
    geom_bar(stat = "identity", position = position_dodge()) +
    facet_wrap(~Price_Range_Category, scales = "free_x", ncol = 5) +
    theme_minimal() +
    labs(
        title = "Total Orders by Category, and Price Range across Years",
        x = "Sub-Category",
        y = "Total Orders"
    ) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

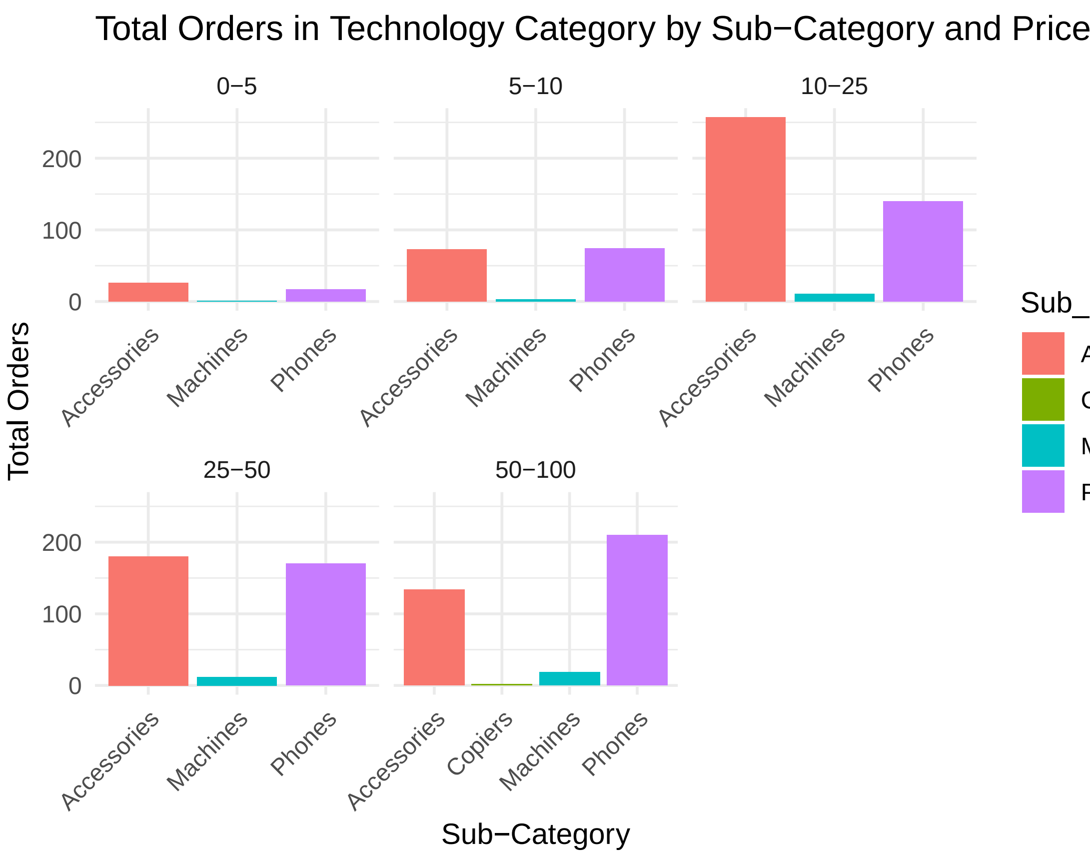


3.5.4 High Price Ranges

```r
aggregated_data_subcategory %>%
    filter(Price_Range_Category == "Above 1000" |
             Price_Range_Category == "500-1000" |
             Price_Range_Category == "400-500" |
             Price_Range_Category == "300-400" |
             Price_Range_Category == "250-300") %>%
    ggplot(aes(x = Category, y = Total_Orders, fill = Category)) +
    geom_bar(stat = "identity", position = position_dodge()) +
    facet_wrap(~Price_Range_Category, scales = "free_x", ncol = 5) +
    theme_minimal() +
    labs(
        title = "Total Orders by Category, and Price Range across Years",
        x = "Sub-Category",
        y = "Total Orders"
    ) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

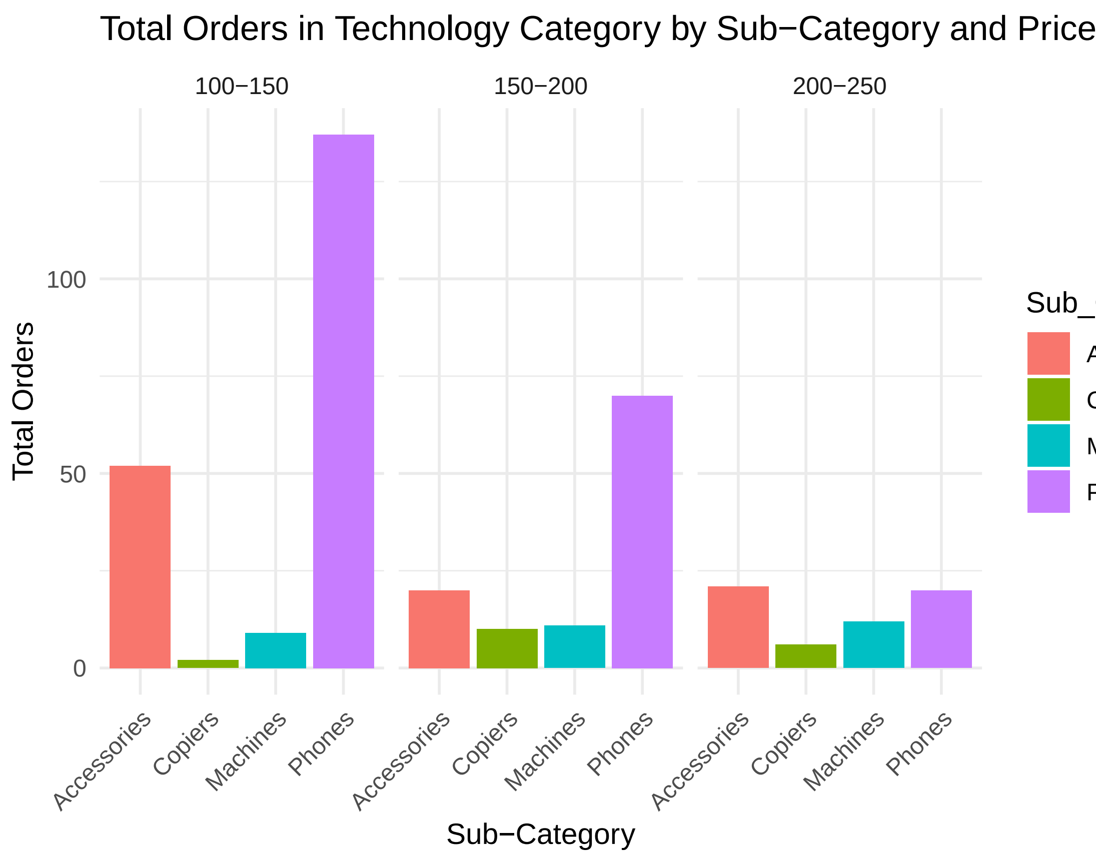
3.5.5 Technology Price ranges

```
ggplot(
    aggregated_data_subcategory %>%
        filter(Category == "Technology" &
            (Price_Range_Category == "50-100" |
                Price_Range_Category == "25-50" |
                Price_Range_Category == "10-25" |
                Price_Range_Category == "5-10" |
                Price_Range_Category == "0-5")),
    aes(x = Sub_Category, y = Total_Orders, fill = Sub_Category)
) +
    geom_bar(stat = "identity") +
    facet_wrap(~Price_Range_Category, scales = "free_x", ncol = 3) +
    theme_minimal() +
    labs(
        title = "Total Orders in Technology Category by Sub-Category and Price Range",
        x = "Sub-Category",
        y = "Total Orders"
    ) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



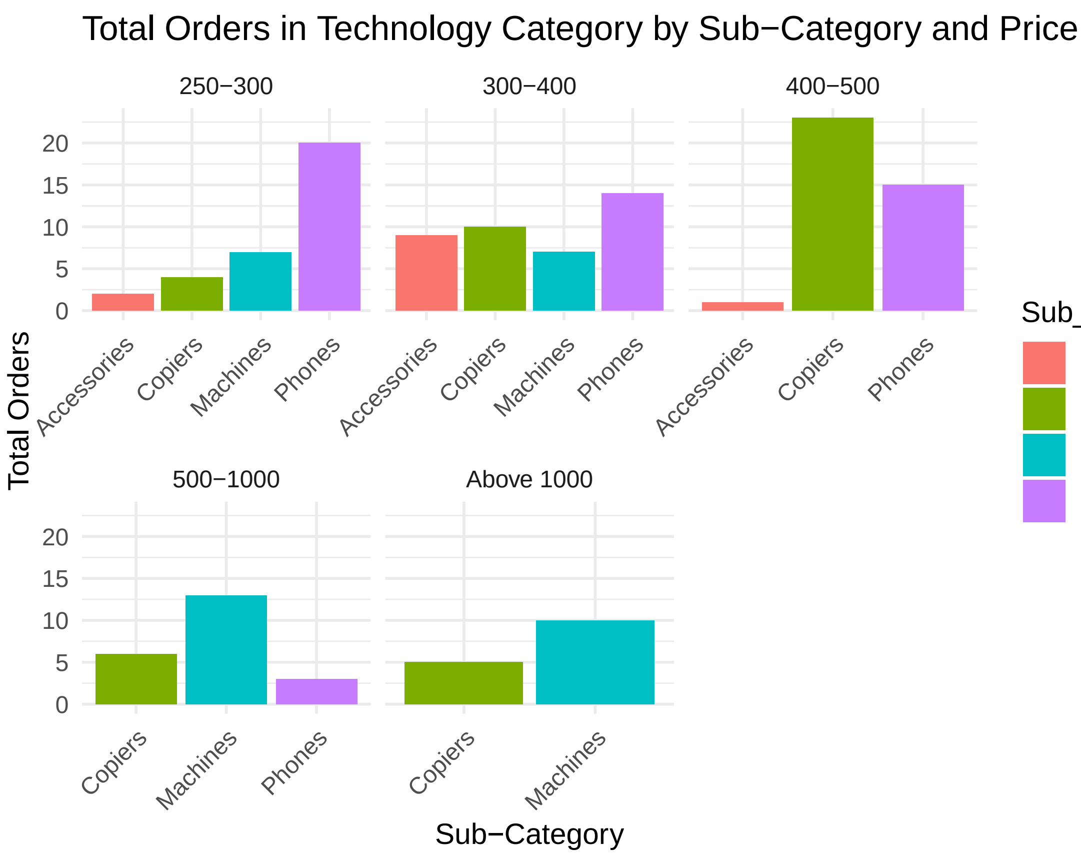
3.5.5.1 Low Price Range

```
ggplot(
    aggregated_data_subcategory %>%
        filter(Category == "Technology" &
            (Price_Range_Category == "200-250" |
               Price_Range_Category == "150-200" |
               Price_Range_Category == "100-150")),
    aes(x = Sub_Category, y = Total_Orders, fill = Sub_Category)
) +
    geom_bar(stat = "identity") +
    facet_wrap(~Price_Range_Category, scales = "free_x", ncol = 3) +
    theme_minimal() +
    labs(
        title = "Total Orders in Technology Category by Sub-Category and Price Range",
        x = "Sub-Category",
        y = "Total Orders"
    ) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



3.5.5.2 Mid Price Range

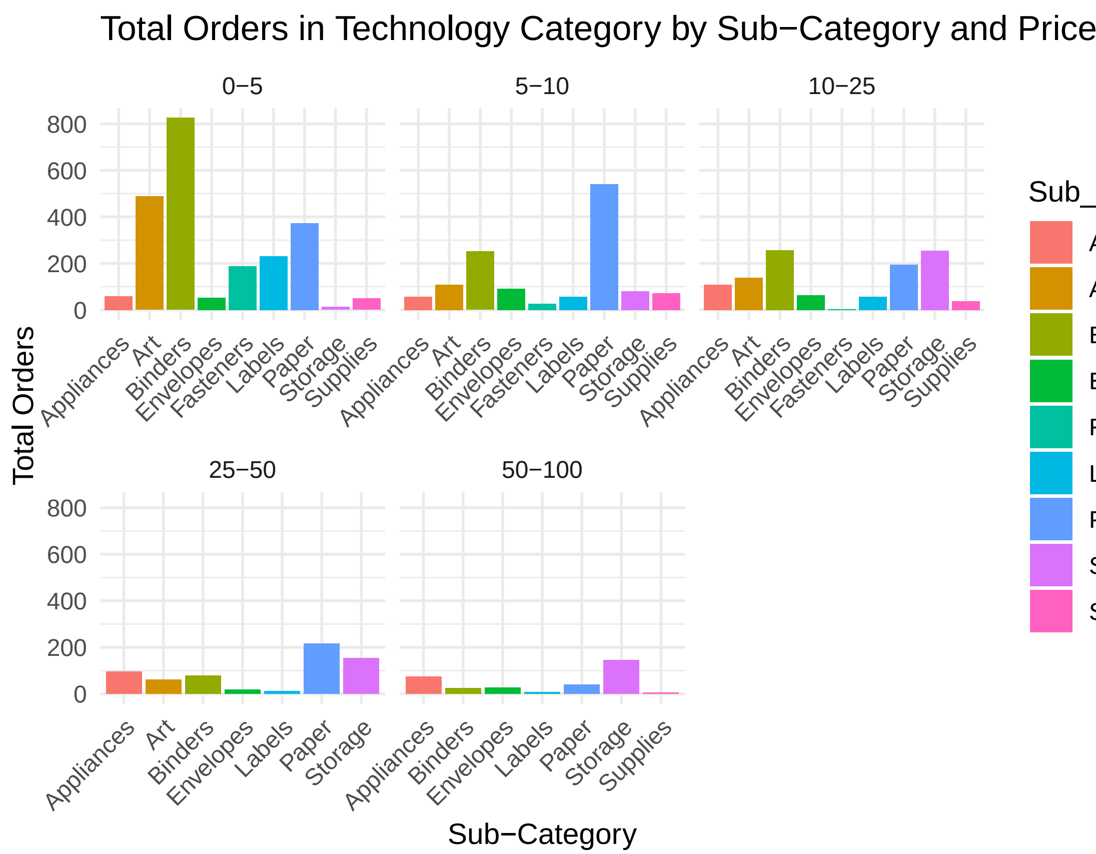
```r
ggplot(
    aggregated_data_subcategory %>%
        filter(Category == "Technology" &
            (Price_Range_Category == "Above 1000" |
               Price_Range_Category == "500-1000" |
               Price_Range_Category == "400-500" |
               Price_Range_Category == "300-400" |
               Price_Range_Category == "250-300")),
    aes(x = Sub_Category, y = Total_Orders, fill = Sub_Category)
) +
    geom_bar(stat = "identity") +
    facet_wrap(~Price_Range_Category, scales = "free_x", ncol = 3) +
    theme_minimal() +
    labs(
        title = "Total Orders in Technology Category by Sub-Category and Price Range",
        x = "Sub-Category",
        y = "Total Orders"
    ) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



3.5.5.3 High Price Range

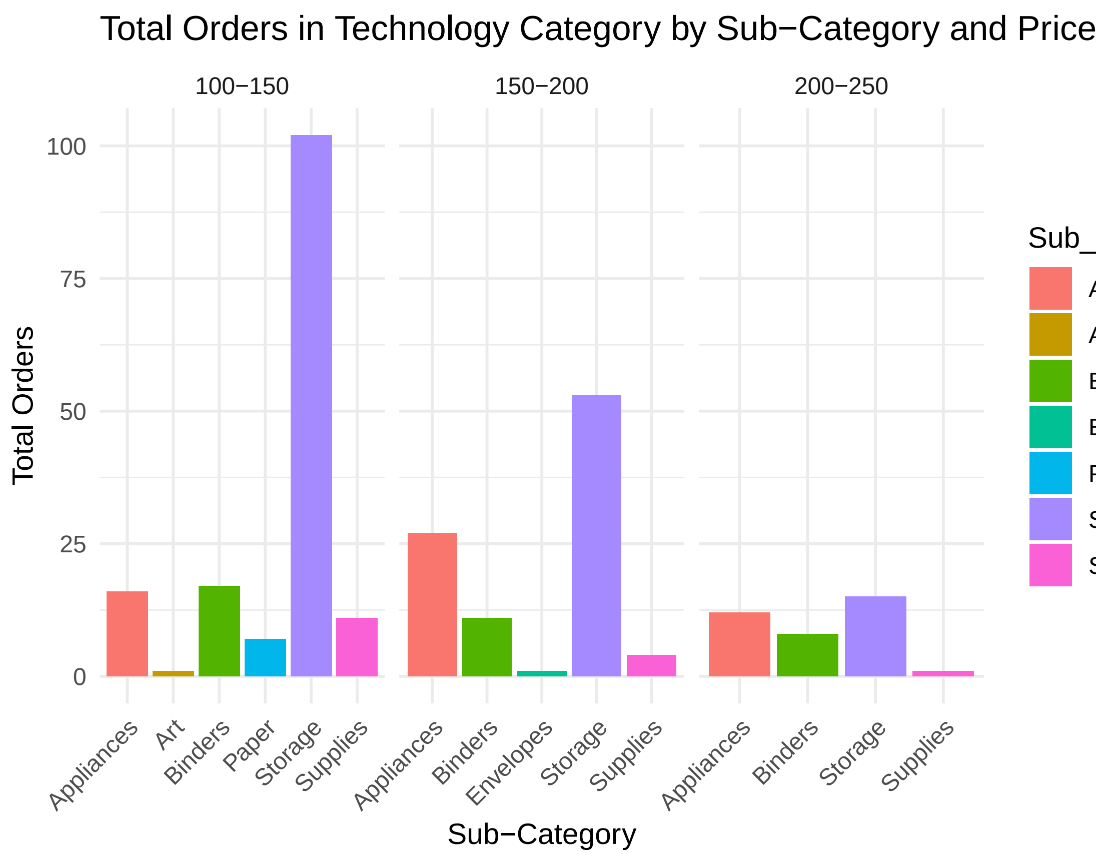
3.5.6 Office Supplies

```
ggplot(
    aggregated_data_subcategory %>%
        filter(Category == "Office Supplies" &
            (Price_Range_Category == "50-100" |
                Price_Range_Category == "25-50" |
                Price_Range_Category == "10-25" |
                Price_Range_Category == "5-10" |
                Price_Range_Category == "0-5")),
    aes(x = Sub_Category, y = Total_Orders, fill = Sub_Category)
) +
    geom_bar(stat = "identity") +
    facet_wrap(~Price_Range_Category, scales = "free_x", ncol = 3) +
    theme_minimal() +
    labs(
        title = "Total Orders in Technology Category by Sub-Category and Price Range",
        x = "Sub-Category",
        y = "Total Orders"
    ) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



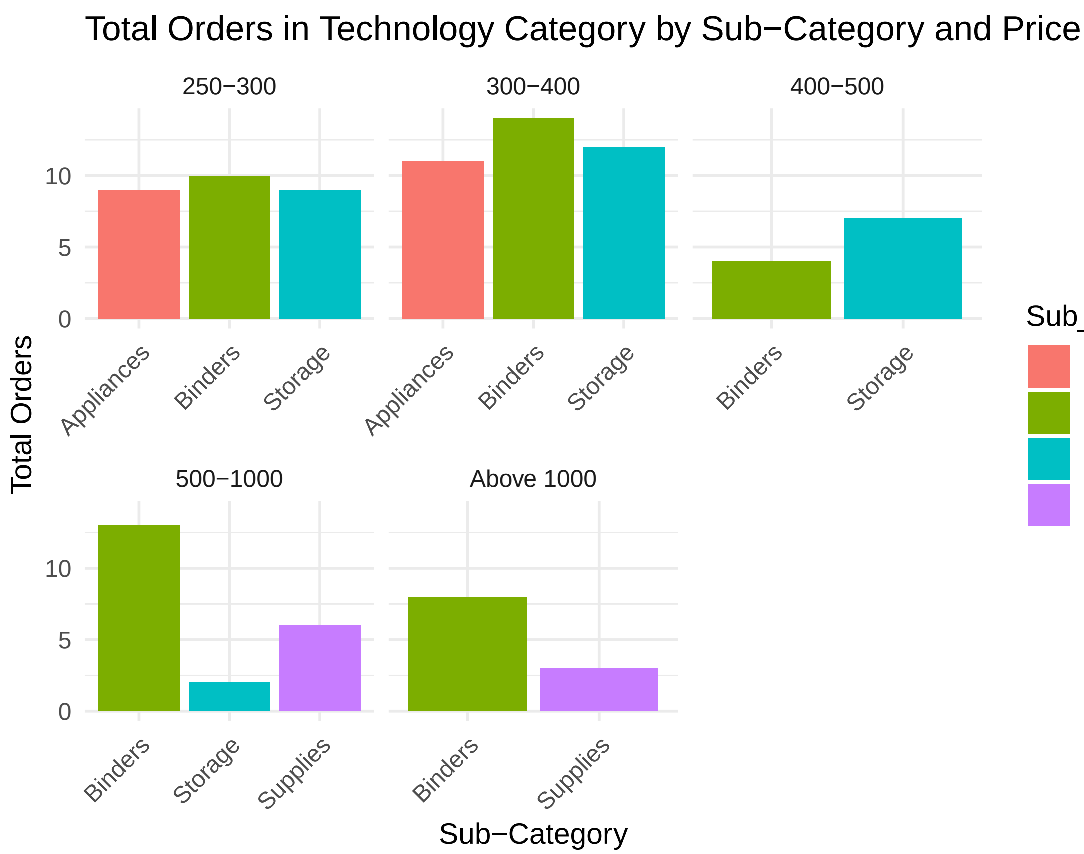
3.5.6.1 Low Price Range

```
ggplot(
    aggregated_data_subcategory %>%
        filter(Category == "Office Supplies" &
            (Price_Range_Category == "200-250" |
               Price_Range_Category == "150-200" |
               Price_Range_Category == "100-150")),
    aes(x = Sub_Category, y = Total_Orders, fill = Sub_Category)
) +
    geom_bar(stat = "identity") +
    facet_wrap(~Price_Range_Category, scales = "free_x", ncol = 3) +
    theme_minimal() +
    labs(
        title = "Total Orders in Technology Category by Sub-Category and Price Range",
        x = "Sub-Category",
        y = "Total Orders"
    ) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



3.5.6.2 Mid Price Range

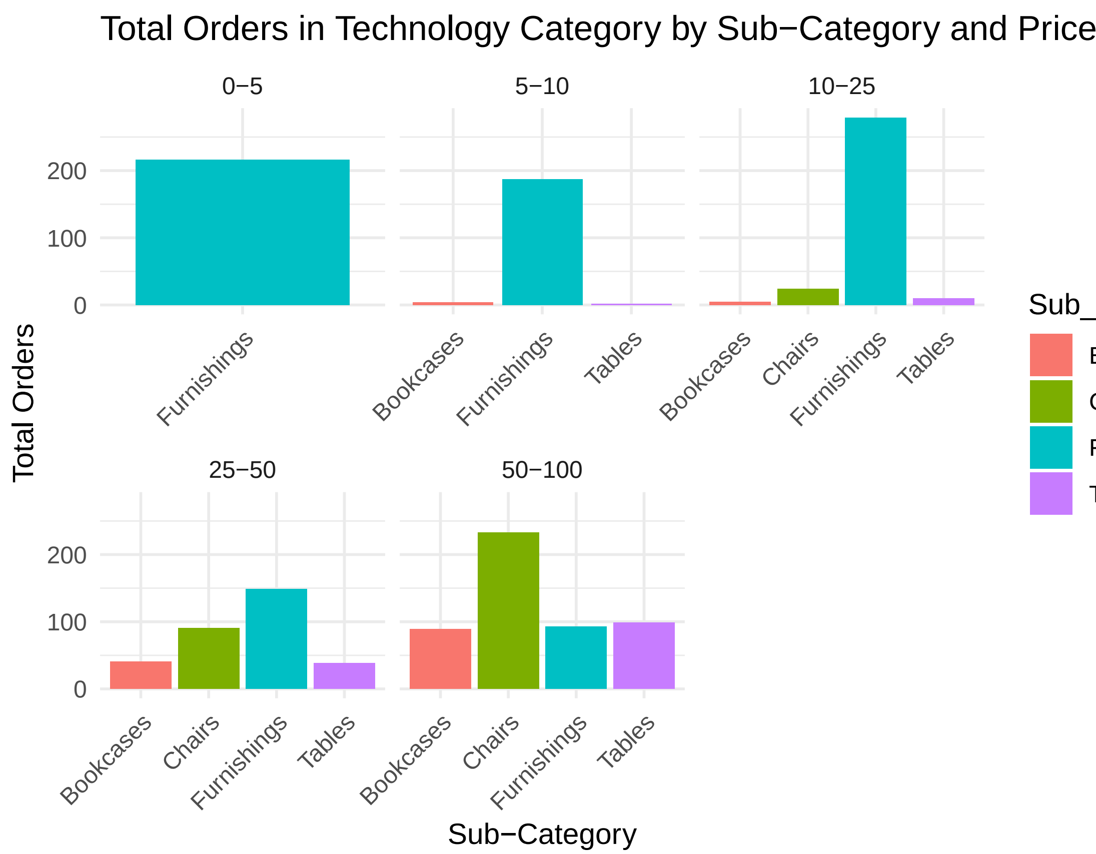
```
ggplot(
    aggregated_data_subcategory %>%
        filter(Category == "Office Supplies" &
            (Price_Range_Category == "Above 1000" |
               Price_Range_Category == "500-1000" |
               Price_Range_Category == "400-500" |
               Price_Range_Category == "300-400" |
               Price_Range_Category == "250-300")),
    aes(x = Sub_Category, y = Total_Orders, fill = Sub_Category)
) +
    geom_bar(stat = "identity") +
    facet_wrap(~Price_Range_Category, scales = "free_x", ncol = 3) +
    theme_minimal() +
    labs(
        title = "Total Orders in Technology Category by Sub-Category and Price Range",
        x = "Sub-Category",
        y = "Total Orders"
    ) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



3.5.6.3 High Price Range

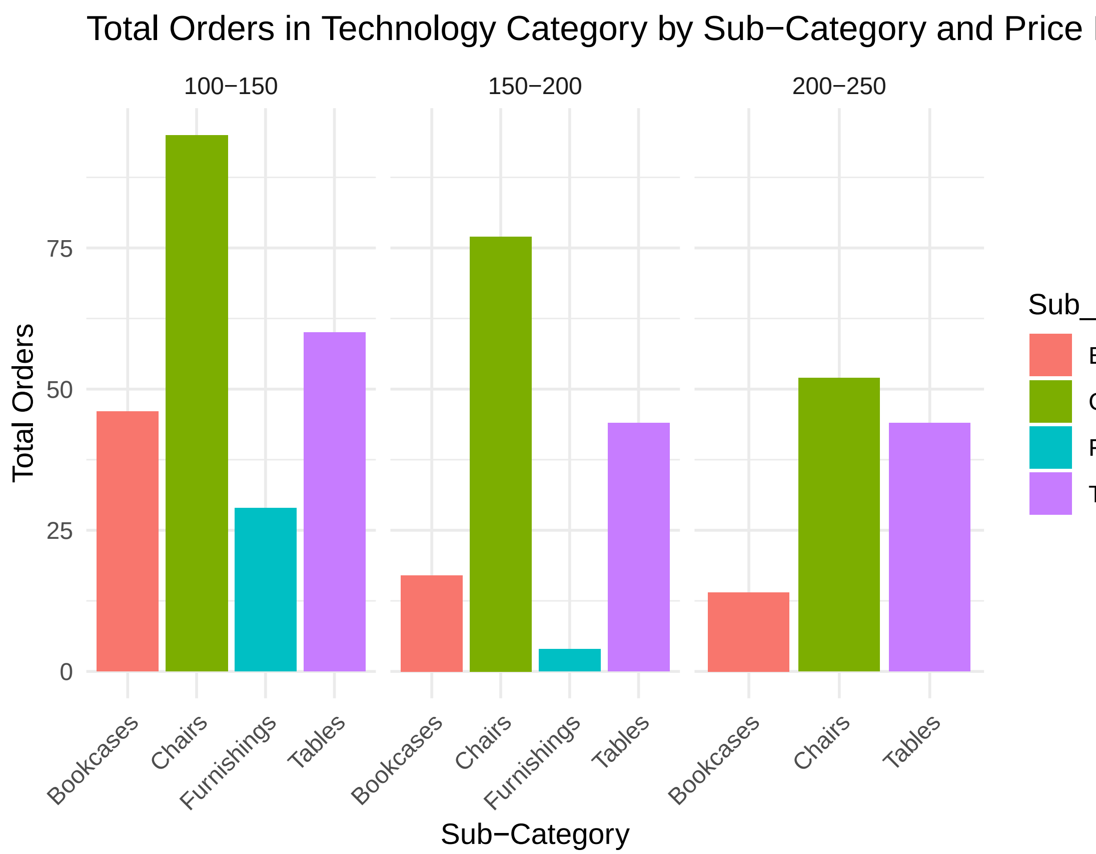
3.5.7 Furniture

```
ggplot(
    aggregated_data_subcategory %>%
        filter(Category == "Furniture" &
            (Price_Range_Category == "50-100" |
                Price_Range_Category == "25-50" |
                Price_Range_Category == "10-25" |
                Price_Range_Category == "5-10" |
                Price_Range_Category == "0-5")),
    aes(x = Sub_Category, y = Total_Orders, fill = Sub_Category)
) +
    geom_bar(stat = "identity") +
    facet_wrap(~Price_Range_Category, scales = "free_x", ncol = 3) +
    theme_minimal() +
    labs(
        title = "Total Orders in Technology Category by Sub-Category and Price Range",
        x = "Sub-Category",
        y = "Total Orders"
    ) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



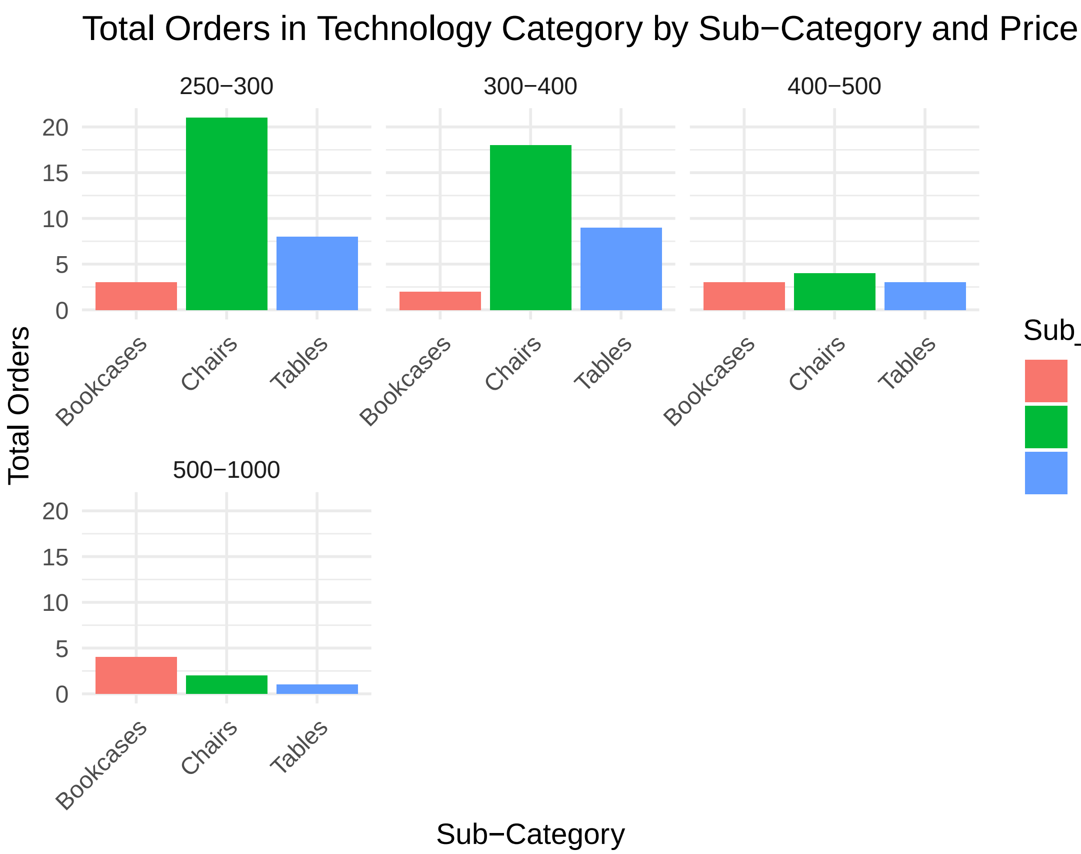
3.5.7.1 Low Price Range

```r
ggplot(
    aggregated_data_subcategory %>%
        filter(Category == "Furniture" &
            (Price_Range_Category == "200-250" |
              Price_Range_Category == "150-200" |
              Price_Range_Category == "100-150")),
    aes(x = Sub_Category, y = Total_Orders, fill = Sub_Category)
) +
    geom_bar(stat = "identity") +
    facet_wrap(~Price_Range_Category, scales = "free_x", ncol = 3) +
    theme_minimal() +
    labs(
        title = "Total Orders in Technology Category by Sub-Category and Price Range",
        x = "Sub-Category",
        y = "Total Orders"
    ) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



3.5.7.2 Mid Price Range

```
ggplot(
    aggregated_data_subcategory %>%
        filter(Category == "Furniture" &
            (Price_Range_Category == "Above 1000" |
               Price_Range_Category == "500-1000" |
               Price_Range_Category == "400-500" |
               Price_Range_Category == "300-400" |
               Price_Range_Category == "250-300")),
    aes(x = Sub_Category, y = Total_Orders, fill = Sub_Category)
) +
    geom_bar(stat = "identity") +
    facet_wrap(~Price_Range_Category, scales = "free_x", ncol = 3) +
    theme_minimal() +
    labs(
        title = "Total Orders in Technology Category by Sub-Category and Price Range",
        x = "Sub-Category",
        y = "Total Orders"
    ) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



3.5.7.3 High Price Range

3.6 Discounts

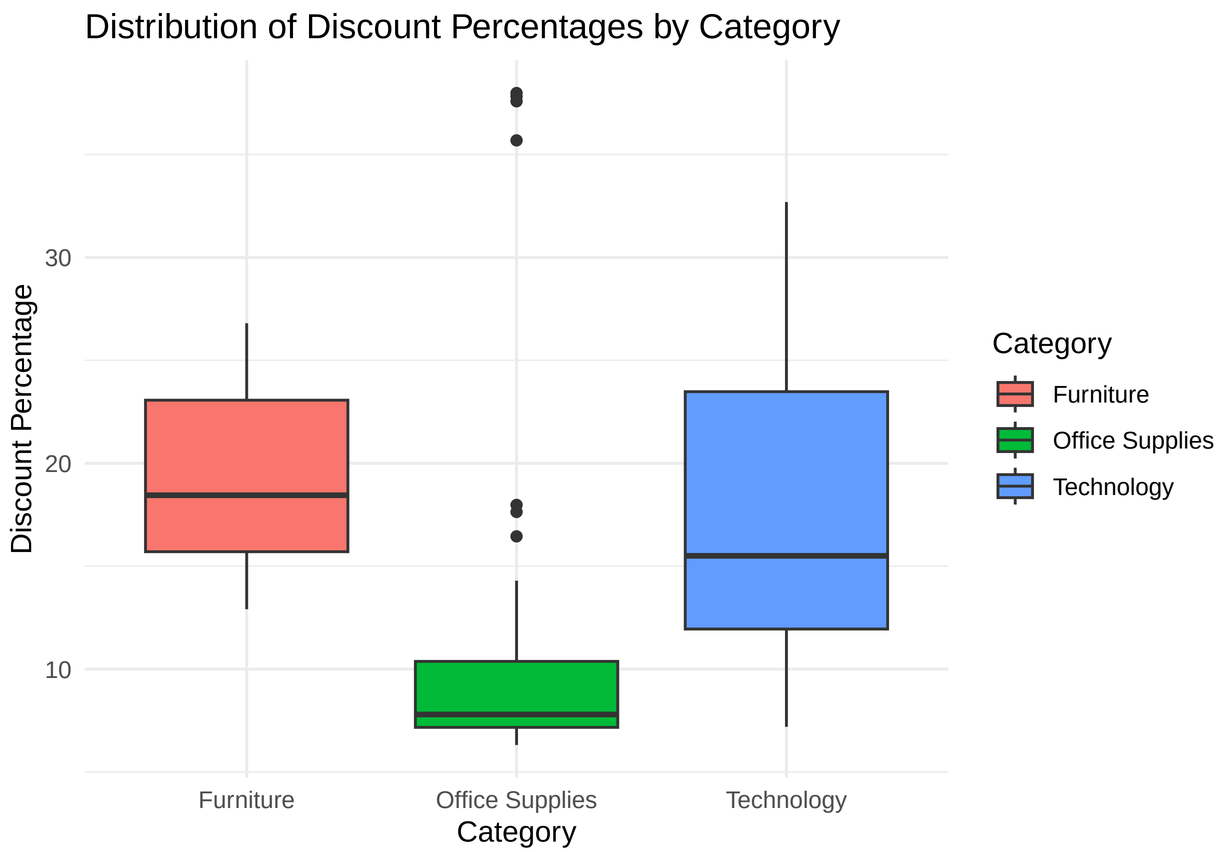
3.6.1 Aggregating the data for Discount and Sales Analysis

```
discount_aggregated <- Orders_Processing %>%
    mutate(
        Discount_Amount = Sales - Net_Sales, # Calculate discount amount for each order
        Discount_Percentage = Discount * 100 # Convert discount to percentage format
    ) %>%
    group_by(Year, Category, Sub_Category) %>% # Group by year, category and sub-category
    summarise(
        Total_Discount = sum(Discount_Amount), # Total discount amount
        Average_Discount_Percentage = mean(Discount_Percentage), # Average discount percentage
        .groups = "drop"
    )
```

3.6.2 Plotting Average Discount Percentage for each Category

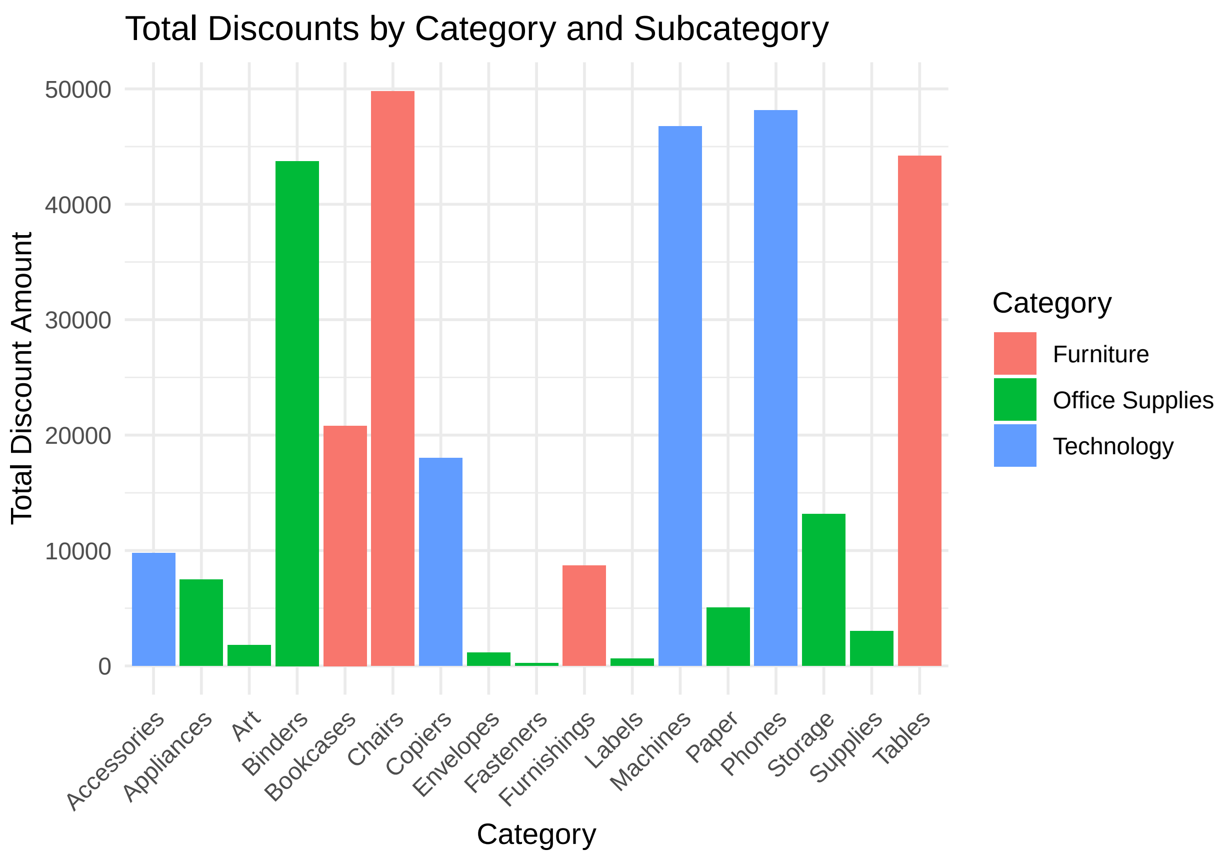
```
ggplot(discount_aggregated, aes(x = Category, y = Average_Discount_Percentage, fill = Category)) +
    geom_boxplot() +
    theme_minimal() +
    labs(
```

```
        title = "Distribution of Discount Percentages by Category",
        x = "Category",
        y = "Discount Percentage"
    )
```



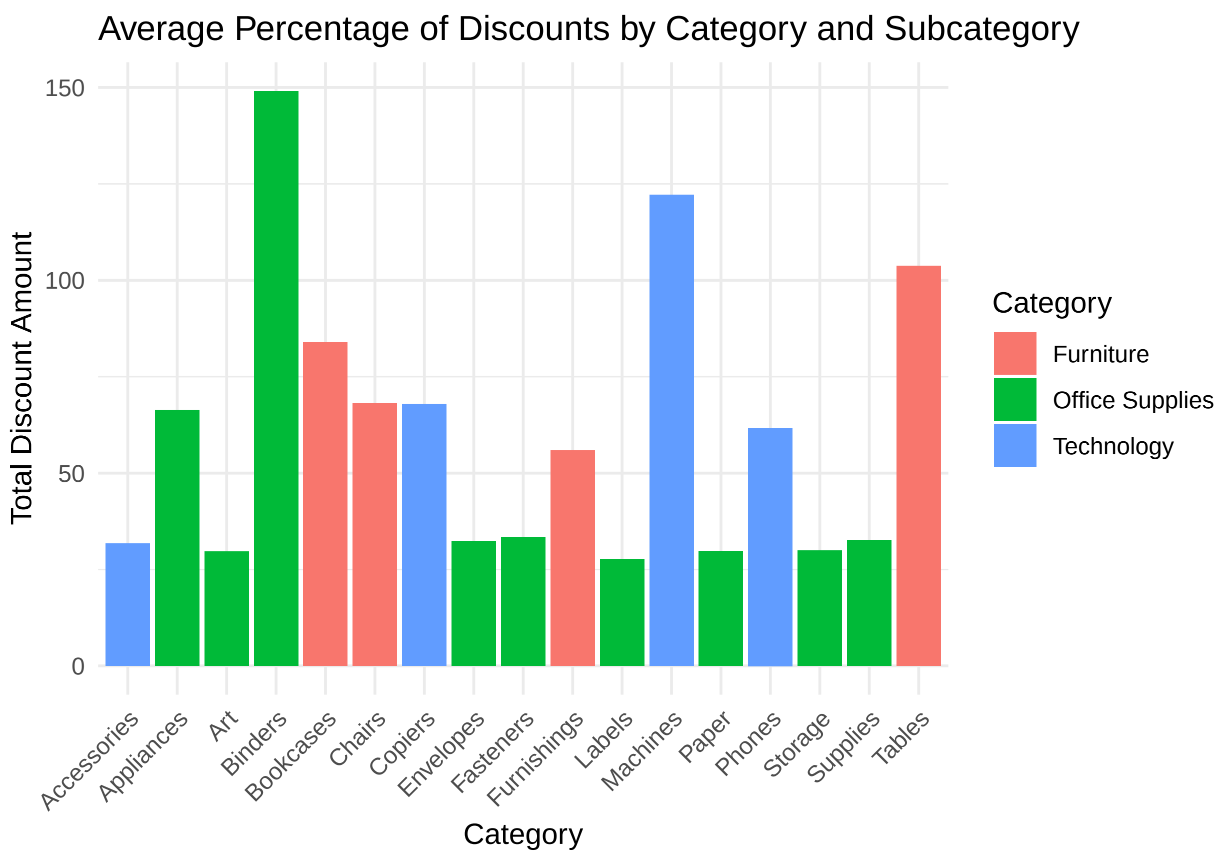
3.6.3 The total amount of discount for each Sub Category

```
ggplot(discount_aggregated, aes(x = Sub_Category, y = Total_Discount, fill = Category)) +
    geom_bar(stat = "identity") +
    theme_minimal() +
    labs(
        title = "Total Discounts by Category and Subcategory",
        x = "Category",
        y = "Total Discount Amount"
    ) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



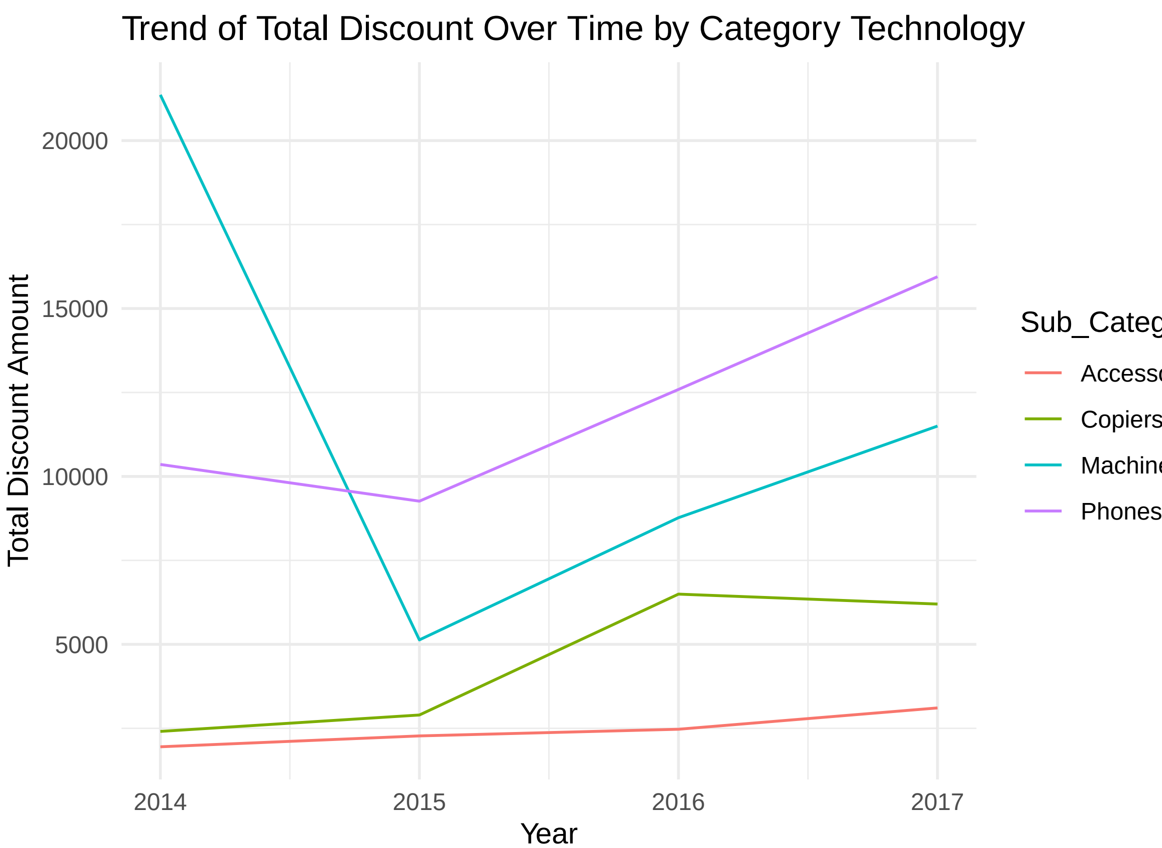
3.6.4 Average Discount Percentage for each Sub Category

```
ggplot(discount_aggregated, aes(x = Sub_Category, y = Average_Discount_Percentage, fill = Category)) +
    geom_bar(stat = "identity") +
    theme_minimal() +
    labs(
        title = "Average Percentage of Discounts by Category and Subcategory",
        x = "Category",
        y = "Total Discount Amount"
    ) +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



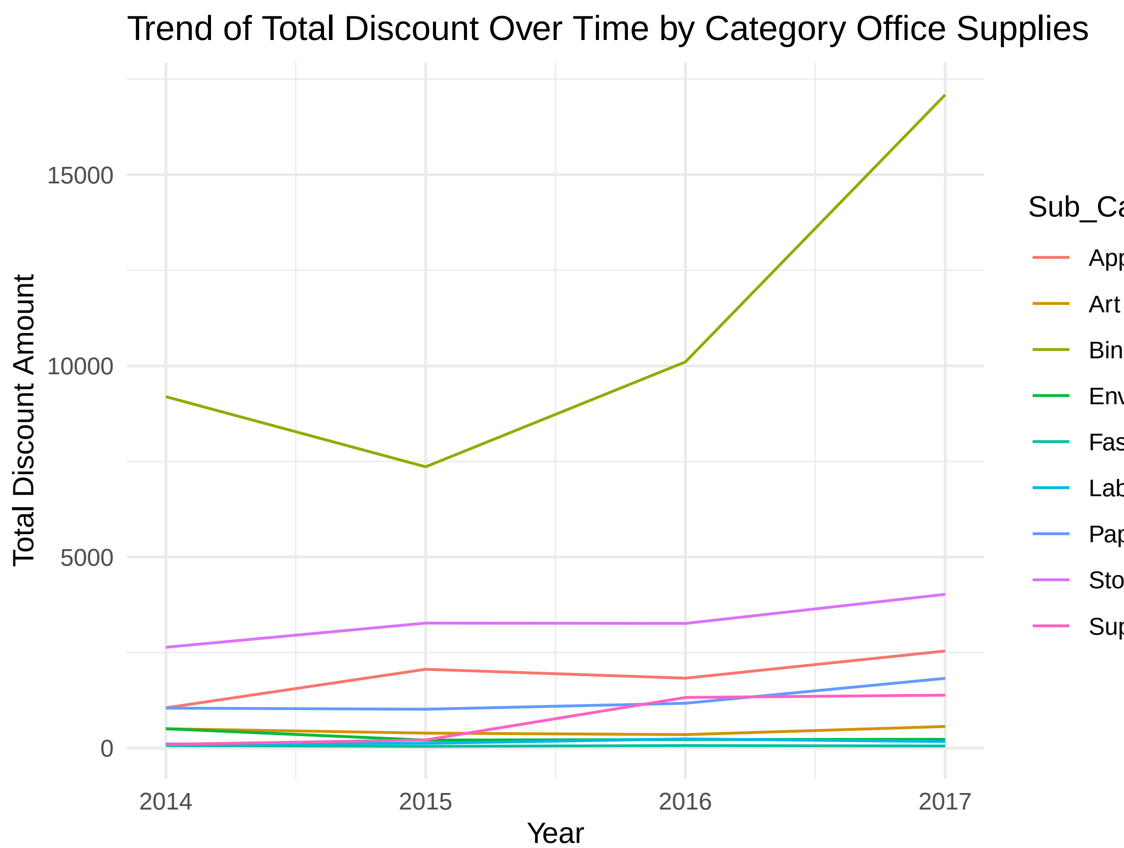
3.6.5 Total Discount

```
discount_aggregated %>%
filter(Category == "Technology") %>%
ggplot(aes(x = Year, y = Total_Discount, color = Sub_Category)) +
    geom_line() +
    theme_minimal() +
    labs(
        title = "Trend of Total Discount Over Time by Category Technology",
        x = "Year",
        y = "Total Discount Amount"
    )
```



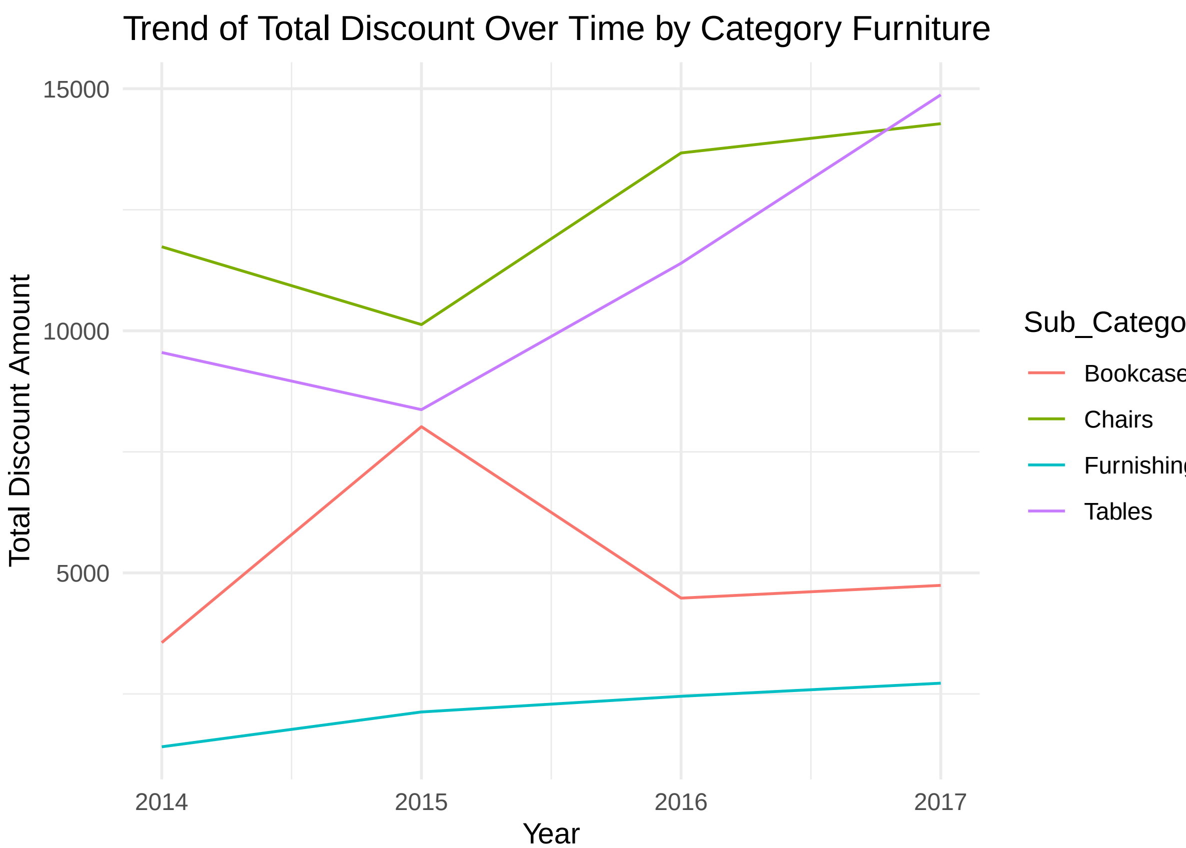
3.6.5.1 Technology

```
discount_aggregated %>%
filter(Category == "Office Supplies") %>%
ggplot(aes(x = Year, y = Total_Discount, color = Sub_Category)) +
    geom_line() +
    theme_minimal() +
    labs(
        title = "Trend of Total Discount Over Time by Category Office Supplies ",
        x = "Year",
        y = "Total Discount Amount"
    )
```



3.6.5.2 Office Supplies

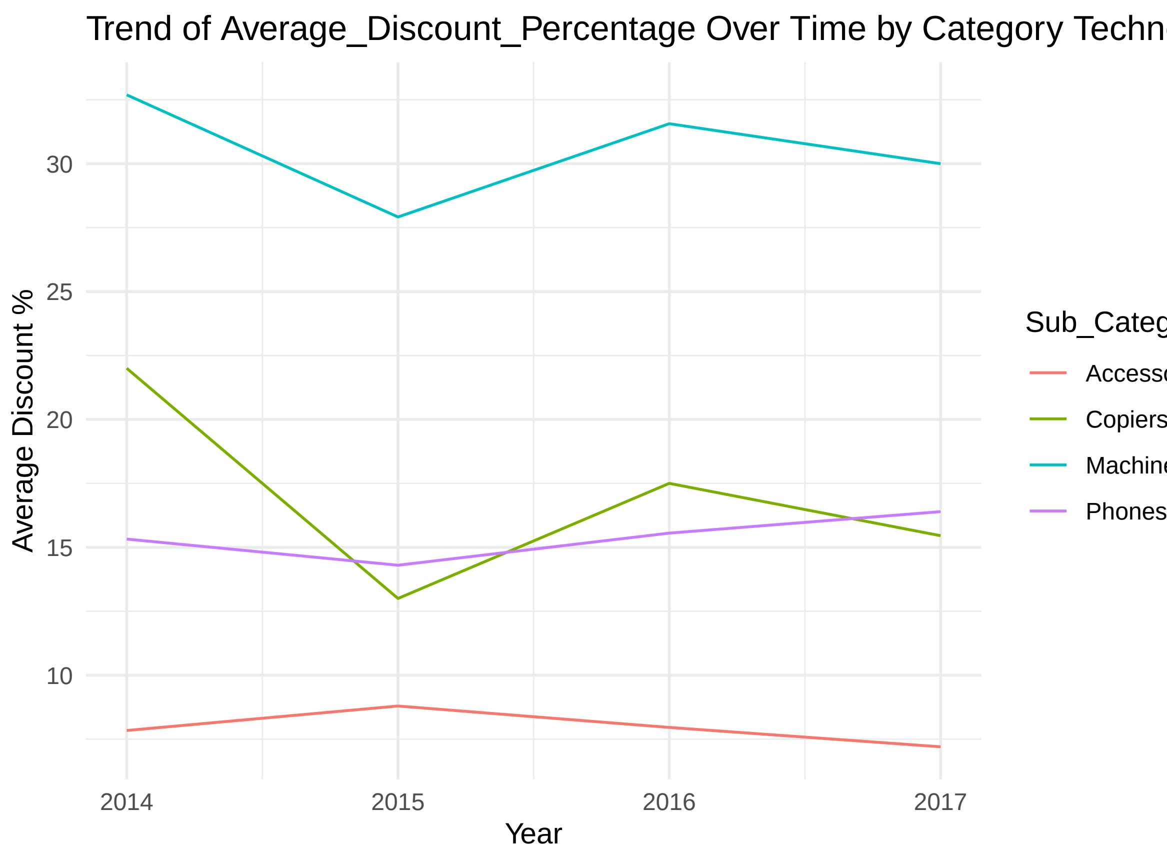
```
discount_aggregated %>%
filter(Category == "Furniture") %>%
ggplot(aes(x = Year, y = Total_Discount, color = Sub_Category)) +
    geom_line() +
    theme_minimal() +
    labs(
        title = "Trend of Total Discount Over Time by Category Furniture",
        x = "Year",
        y = "Total Discount Amount"
    )
```



3.6.5.3 Furniture

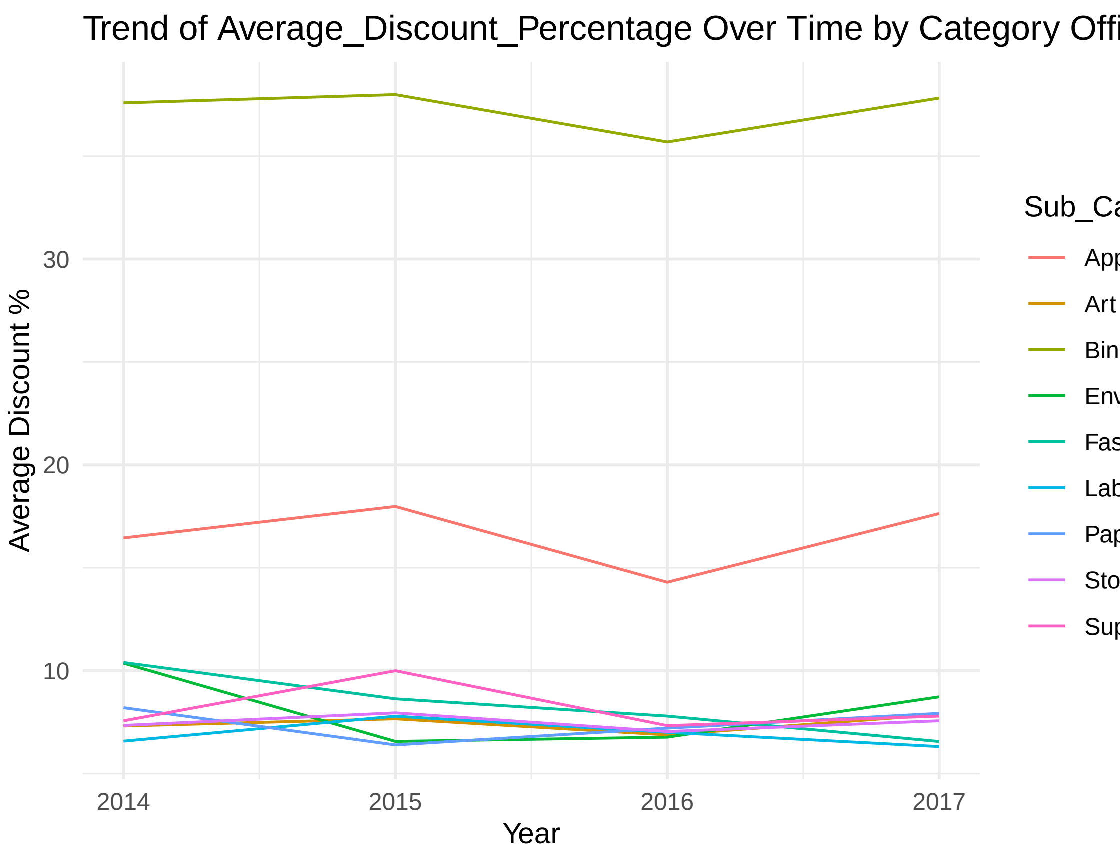
3.6.4 Average Discount Percentage

```
discount_aggregated %>%
filter(Category == "Technology") %>%
ggplot(aes(x = Year, y = Average_Discount_Percentage, color = Sub_Category)) +
    geom_line() +
    theme_minimal() +
    labs(
        title = "Trend of Average_Discount_Percentage Over Time by Category Technology",
        x = "Year",
        y = "Average Discount %"
    )
```



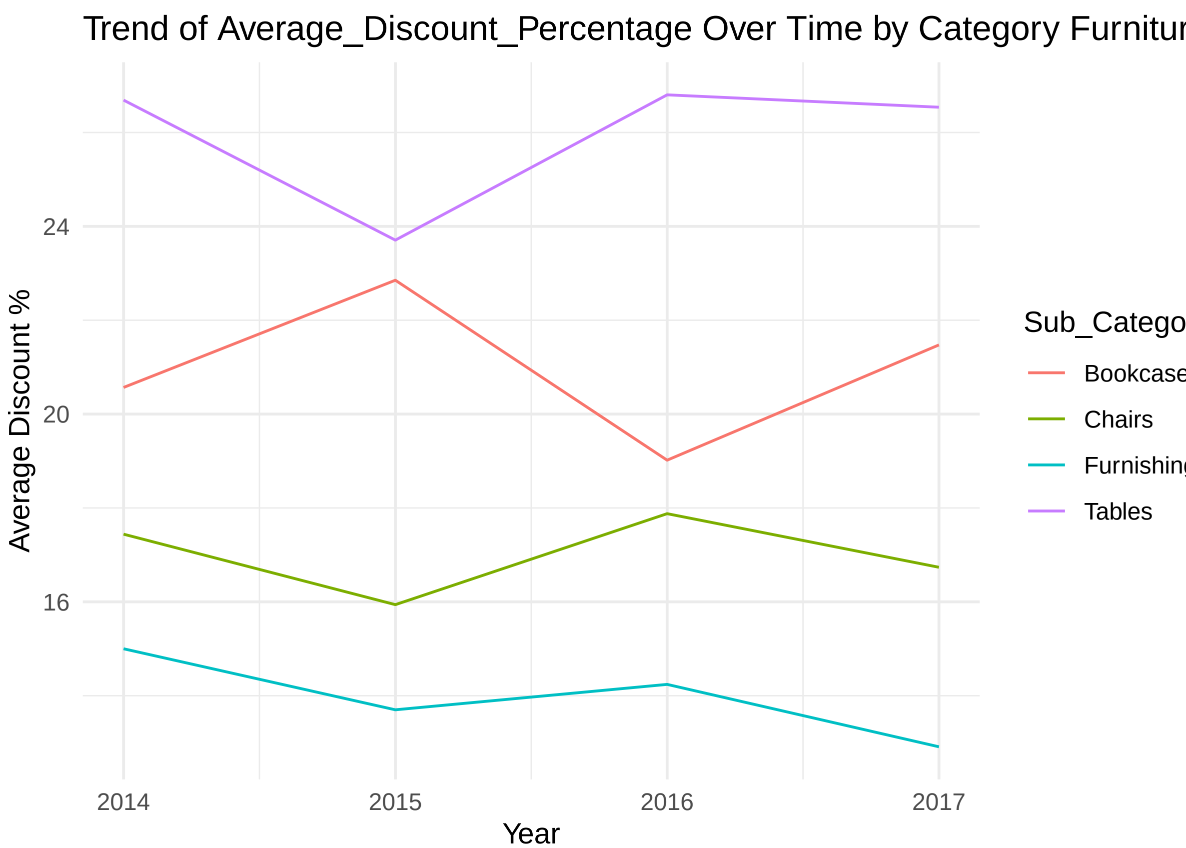
3.6.4.1 Technology

```
discount_aggregated %>%
filter(Category == "Office Supplies") %>%
ggplot(aes(x = Year, y = Average_Discount_Percentage, color = Sub_Category)) +
    geom_line() +
    theme_minimal() +
    labs(
        title = "Trend of Average_Discount_Percentage Over Time by Category Office Supplies ",
        x = "Year",
        y = "Average Discount %"
    )
```



3.6.4.2 Office Supplies

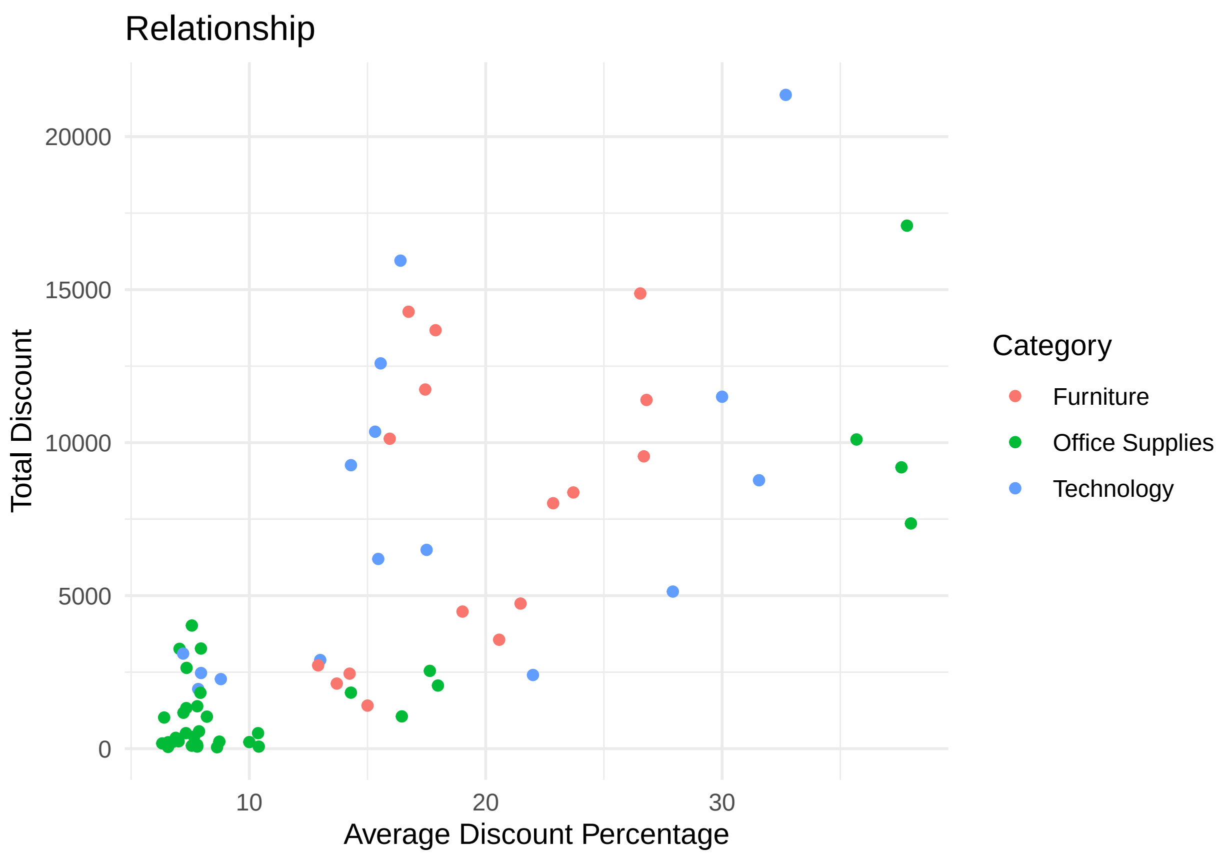
```
discount_aggregated %>%
filter(Category == "Furniture") %>%
ggplot(aes(x = Year, y = Average_Discount_Percentage, color = Sub_Category)) +
    geom_line() +
    theme_minimal() +
    labs(
        title = "Trend of Average_Discount_Percentage Over Time by Category Furniture",
        x = "Year",
        y = "Average Discount %"
    )
```



3.6.4.3 Furniture

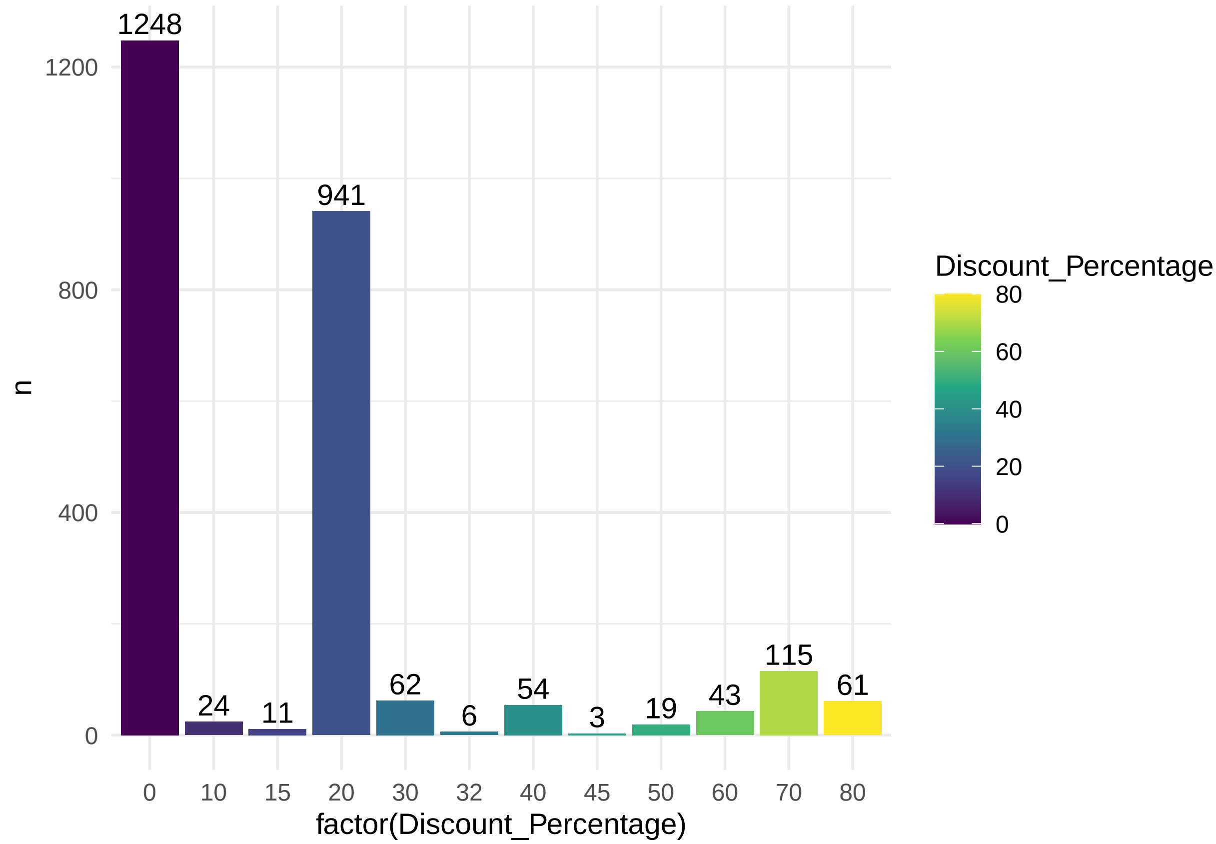
3.6.5 Relationship between Average Discount Percentage and Total Discount Amount

```
ggplot(discount_aggregated, aes(x = Average_Discount_Percentage, y = Total_Discount, color = Category))
    geom_point() +
    theme_minimal() +
    labs(
        title = "Relationship",
        x = "Average Discount Percentage",
        y = "Total Discount"
    )
```



3.6.6 Discount Frequency

```
discount_frequency <- Orders_2016_Processing %>%
    mutate(Discount_Percentage = Discount * 100) %>% # Convert to percentage format
    count(Discount_Percentage) %>%
    arrange(desc(n))

ggplot(discount_frequency, aes(x = factor(Discount_Percentage), y = n, fill = Discount_Percentage)) +
    geom_bar(stat = "identity") +
    geom_text(aes(label = n), vjust = -0.3, position = position_dodge(width = 0.9)) + # Display count o
    scale_fill_viridis_c() + # A more colorful palette
    theme_minimal()
```



3.3Map Plots

3.3.1 Orders

```
library(sf)
```

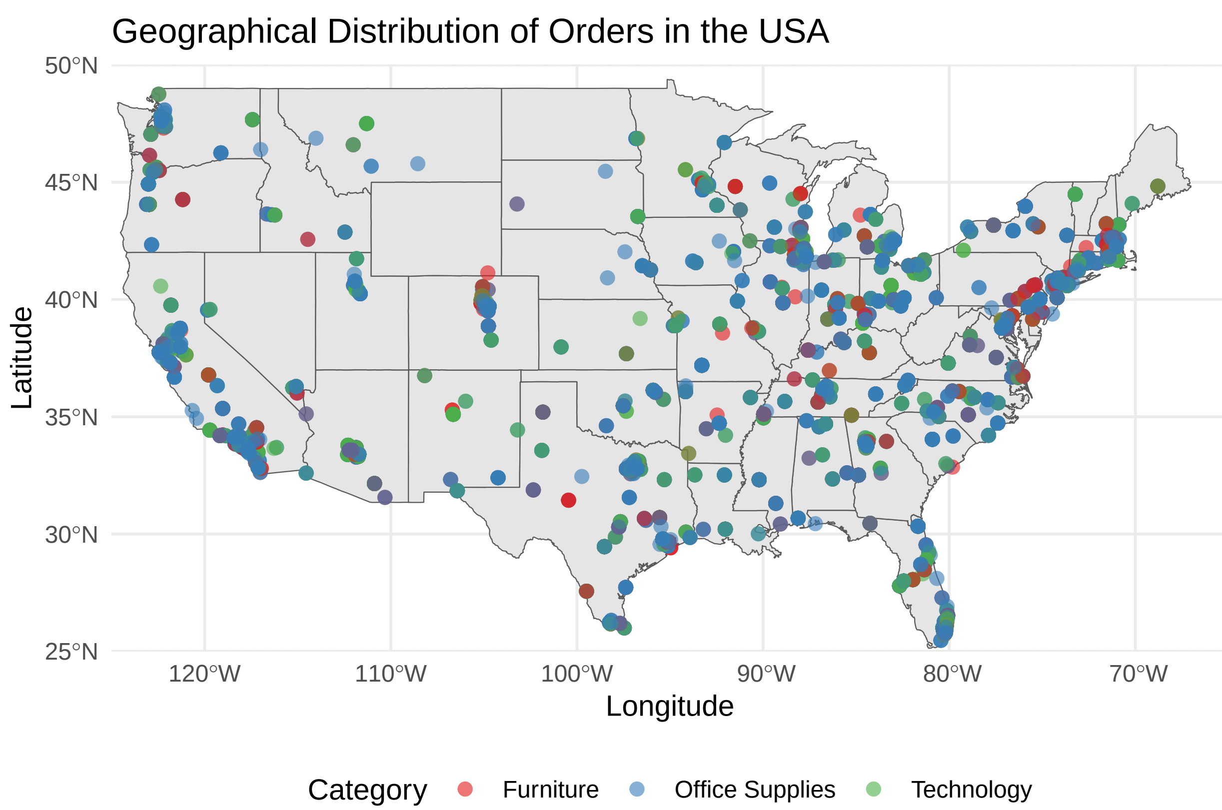
```
## Linking to GEOS 3.11.0, GDAL 3.5.3, PROJ 9.1.0; sf_use_s2() is TRUE
```

```
library(ggplot2)
library(maps)
```

```
##
## Attaching package: 'maps'

## The following object is masked from 'package:purrr':
##
##     map
```

```
# Base world map
world <- sf::st_as_sf(maps::map("world", plot = FALSE, fill = TRUE))

# US states map
```

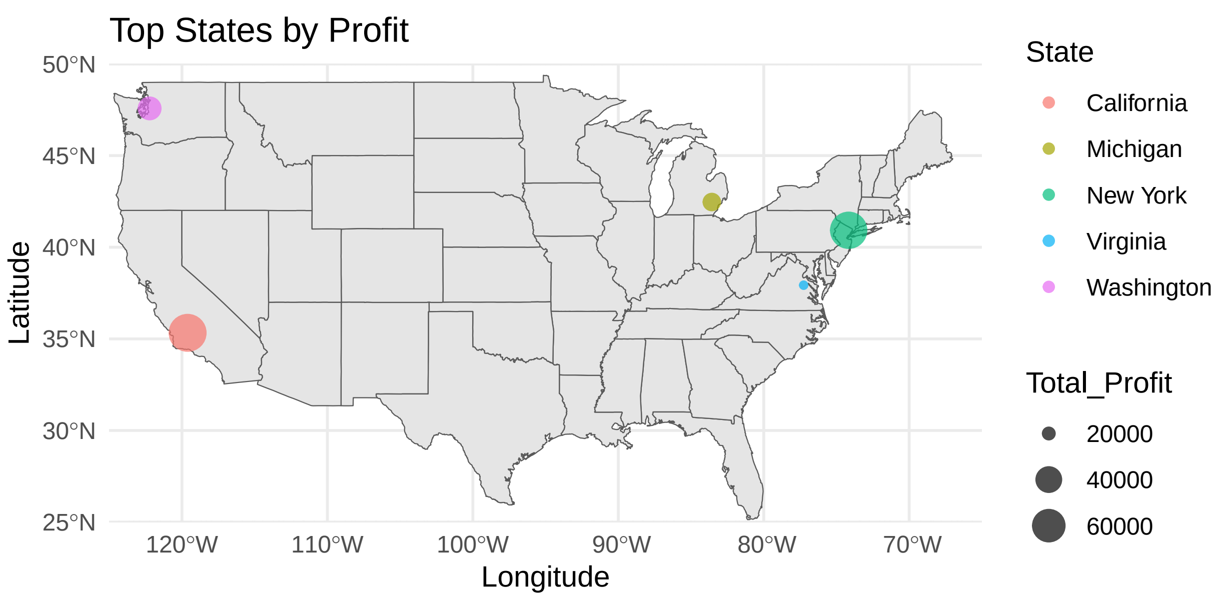
```r
us_states <- sf::st_as_sf(maps::map("state", plot = FALSE, fill = TRUE))

# ggplot code with Orders_Processing dataset
ggplot(data = us_states) +
    geom_sf() +
    geom_point(data = Orders_Processing, aes(x = lng, y = lat, color = Category), size = 2, alpha = 0.6)
    coord_sf(xlim = c(-125, -65), ylim = c(25, 50), expand = FALSE) +
    labs(
        title = "Geographical Distribution of Orders in the USA",
        x = "Longitude",
        y = "Latitude"
    ) +
    theme_minimal() +
    theme(legend.position = "bottom", legend.title.align = 0.5) +
    scale_color_brewer(type = "qual", palette = "Set1")
```



3.3.2 State

```r
state_aggregated_data <- Orders_Processing %>%
    group_by(State) %>%
    summarise(
        Total_Profit = sum(Profit, na.rm = TRUE), # Replace Net_Sales with your measure
        Avg_Lat = mean(lat, na.rm = TRUE),
```

```r
        Avg_Lng = mean(lng, na.rm = TRUE)
    ) %>%
    arrange(desc(Total_Profit))

top_states <- head(state_aggregated_data, 5)

ggplot() +
    geom_sf(data = us_states) +
    geom_point(data = top_states, aes(x = Avg_Lng, y = Avg_Lat, size = Total_Profit, color = State), alp
    coord_sf(xlim = c(-125, -65), ylim = c(25, 50), expand = FALSE) +
    labs(
        title = "Top States by Profit",
        x = "Longitude",
        y = "Latitude"
    ) +
    theme_minimal()
```



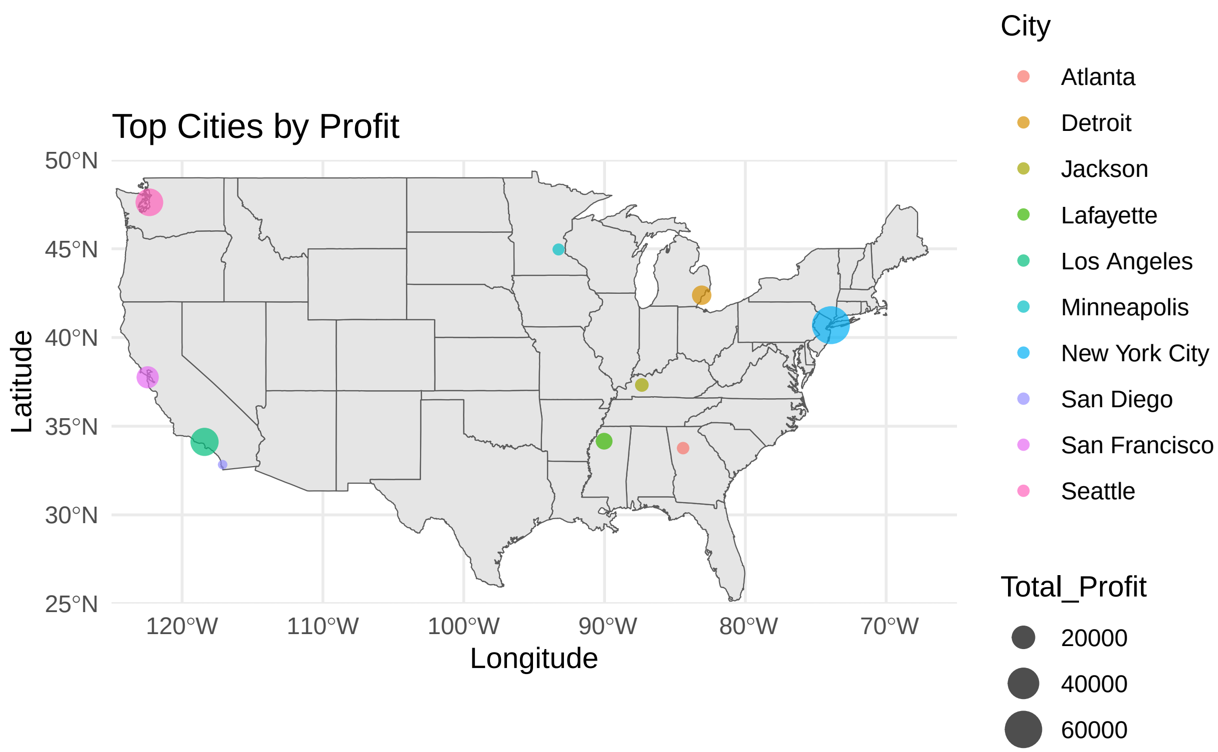

```r
print(top_states)
```

```
## # A tibble: 5 x 4
##   State       Total_Profit Avg_Lat Avg_Lng
##   <chr>              <dbl>   <dbl>   <dbl>
## 1 California        76363.    35.3  -120.
## 2 New York          74012.    40.9   -74.2
## 3 Washington        33403.    47.6  -122.
```

```
## 4 Michigan          24463.    42.5   -83.6
## 5 Virginia          18598.    37.9   -77.3
```

3.3.3 City

```
city_aggregated_data <- Orders_Processing %>%
    group_by(City) %>%
    summarise(
        Total_Profit = sum(Profit, na.rm = TRUE), # Replace Net_Sales with your measure
        Avg_Lat = mean(lat, na.rm = TRUE),
        Avg_Lng = mean(lng, na.rm = TRUE)
    ) %>%
    arrange(desc(Total_Profit))

# Top 10 cities
top_cities <- head(city_aggregated_data, 10)

ggplot() +
    geom_sf(data = us_states) +
    geom_point(data = top_cities, aes(x = Avg_Lng, y = Avg_Lat, size = Total_Profit, color = City), alpl
    coord_sf(xlim = c(-125, -65), ylim = c(25, 50), expand = FALSE) +
    labs(
        title = "Top Cities by Profit",
        x = "Longitude",
        y = "Latitude"
    ) +
    theme_minimal()
```




```
print(top_cities)
```

```
## # A tibble: 10 x 4
##    City          Total_Profit Avg_Lat Avg_Lng
##    <chr>                <dbl>   <dbl>   <dbl>
##  1 New York City       62010.    40.7   -73.9
##  2 Los Angeles         30435.    34.1  -118.
##  3 Seattle             29156.    47.6  -122.
##  4 San Francisco       17507.    37.8  -122.
##  5 Detroit             13182.    42.4   -83.1
##  6 Lafayette           10018.    34.2   -90.0
##  7 Jackson              7582.    37.3   -87.4
##  8 Atlanta              6994.    33.8   -84.4
##  9 Minneapolis          6825.    45.0   -93.3
## 10 San Diego            6377.    32.8  -117.
```

4.PREDICTIVE ANALISYS

4.1CORRELATION POPULATION AND SALES/PROFIT

```
# Aggregate sales and profits by state, and calculate average population for each state
statewise_data <- Orders_Processing %>%
```

```r
  group_by(State) %>%
  summarise(Total_Sales = sum(Sales),
            Total_Profit = sum(Profit),
            Average_Population = mean(Population2021)) %>%
  ungroup()

# Calculate correlations
sales_population_corr <- cor(statewise_data$Total_Sales, statewise_data$Average_Population)
profit_population_corr <- cor(statewise_data$Total_Profit, statewise_data$Average_Population)

# Print correlation coefficients
print(paste("Correlation between Sales and Population:", sales_population_corr))
```

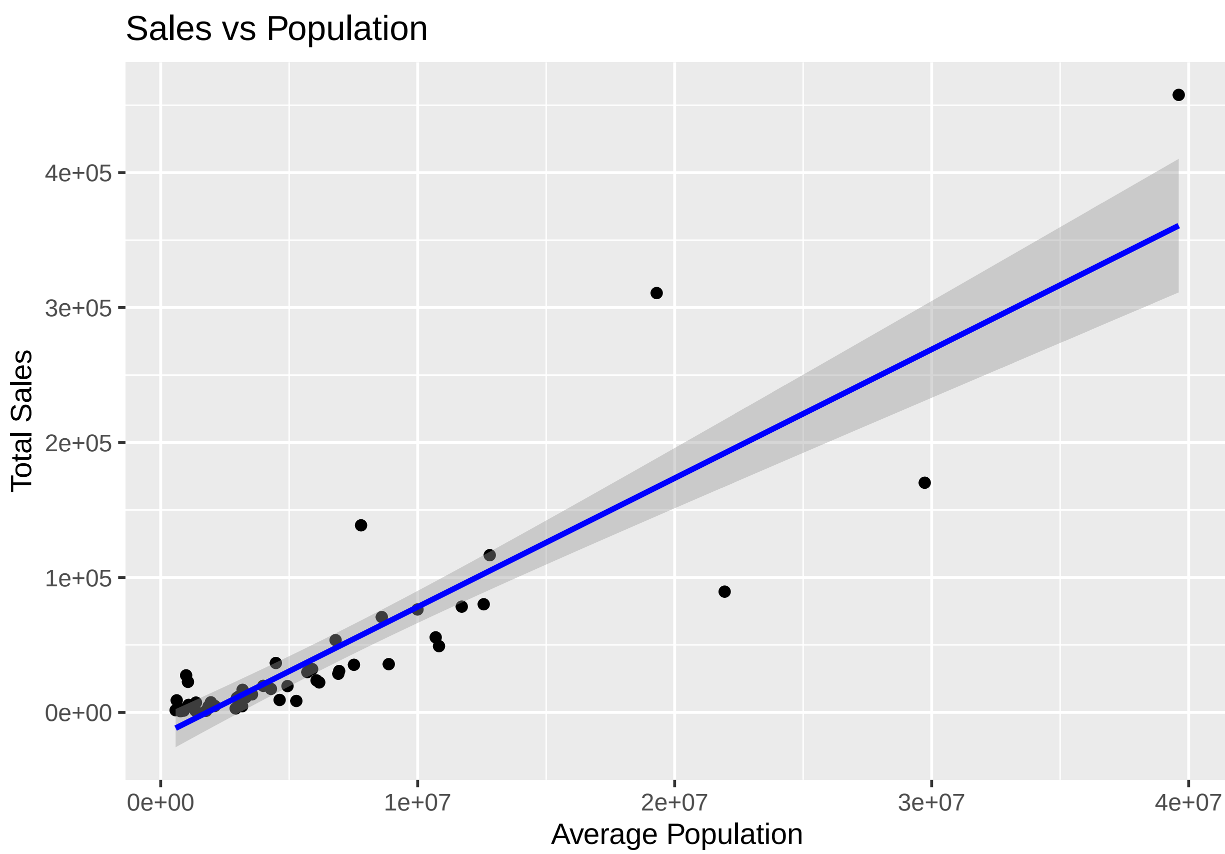
```
## [1] "Correlation between Sales and Population: 0.885694387954497"
```

```r
print(paste("Correlation between Profit and Population:", profit_population_corr))
```

```
## [1] "Correlation between Profit and Population: 0.389828202817969"
```

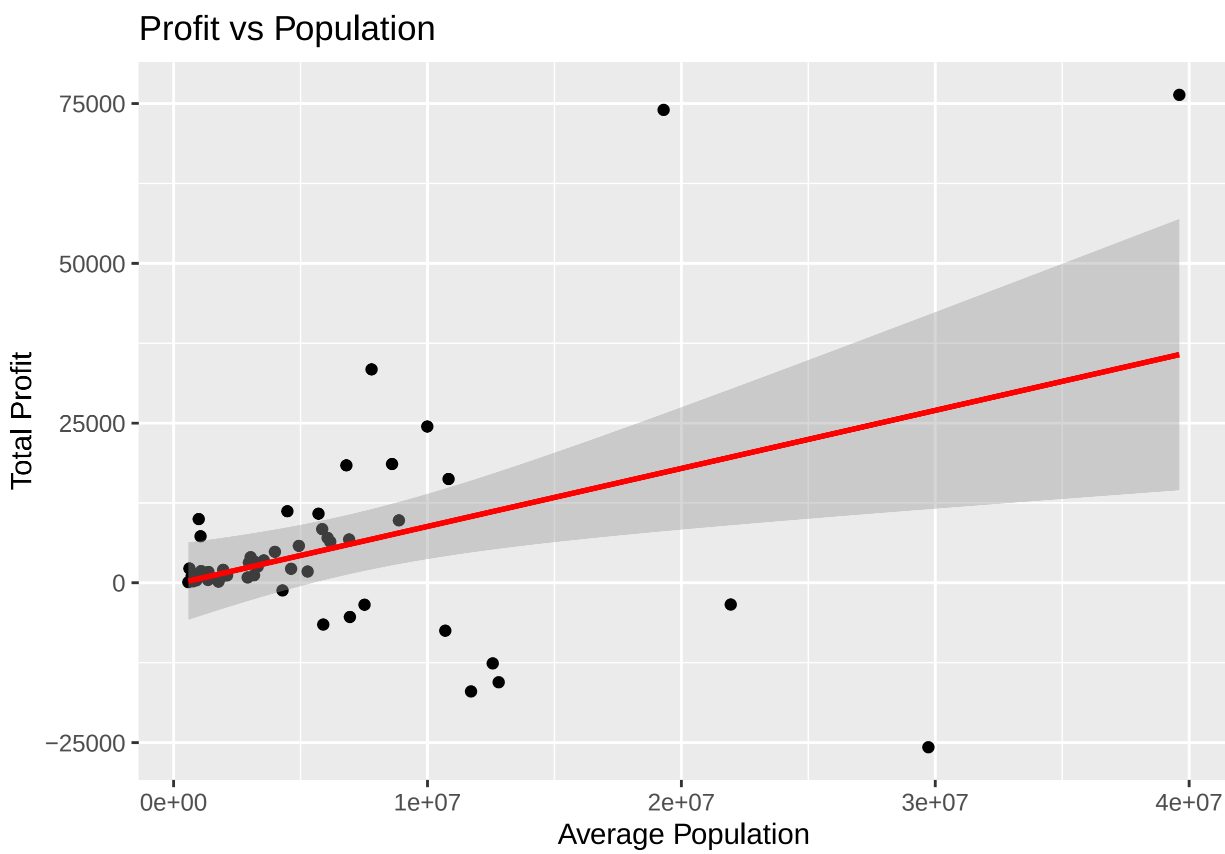
```r
# Scatter plot for Sales vs Population
ggplot(statewise_data, aes(x = Average_Population, y = Total_Sales)) +
  geom_point() +
  geom_smooth(method = "lm", color = "blue") +
  labs(title = "Sales vs Population", x = "Average Population", y = "Total Sales")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
# Scatter plot for Profit vs Population
ggplot(statewise_data, aes(x = Average_Population, y = Total_Profit)) +
  geom_point() +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "Profit vs Population", x = "Average Population", y = "Total Profit")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



In this section, the existing correlation between the total population of each state and sales and profits is analyzed. For sales, we can observe a strong correlation of 0.88, whereas for profits, it's 0.38. This seems to indicate that there are differences in the products purchased by citizens in each state, which subsequently affect profits and lead to this discrepancy between sales and profits. This is because the per capita GDP of different states can vary by more than 100%.

4.2 CORRELATION DISCOUNT AND SALES/PROFIT

```r
library(broom)


# Example analysis: Correlation between Discount and Sales
discount_sales_corr <- cor(Orders_Processing$Discount, Orders_Processing$Sales, use = "complete.obs")

# Fit a linear model
model <- lm(Sales ~ Discount, data = Orders_Processing)
summary(model) # To get R2 and other stats
```

```
##
## Call:
## lm(formula = Sales ~ Discount, data = Orders_Processing)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
##   -242.3  -211.8  -170.8   -21.9 22438.0
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  243.291      7.819  31.117   <2e-16 ***
## Discount     -85.557     30.187  -2.834   0.0046 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 623.1 on 9989 degrees of freedom
## Multiple R-squared:  0.0008035,  Adjusted R-squared:  0.0007035
## F-statistic: 8.033 on 1 and 9989 DF,  p-value: 0.004603
```

```r
model_fit <- glance(model)

# Print correlation coefficient and R2
print(paste("Correlation coefficient:", discount_sales_corr))
```

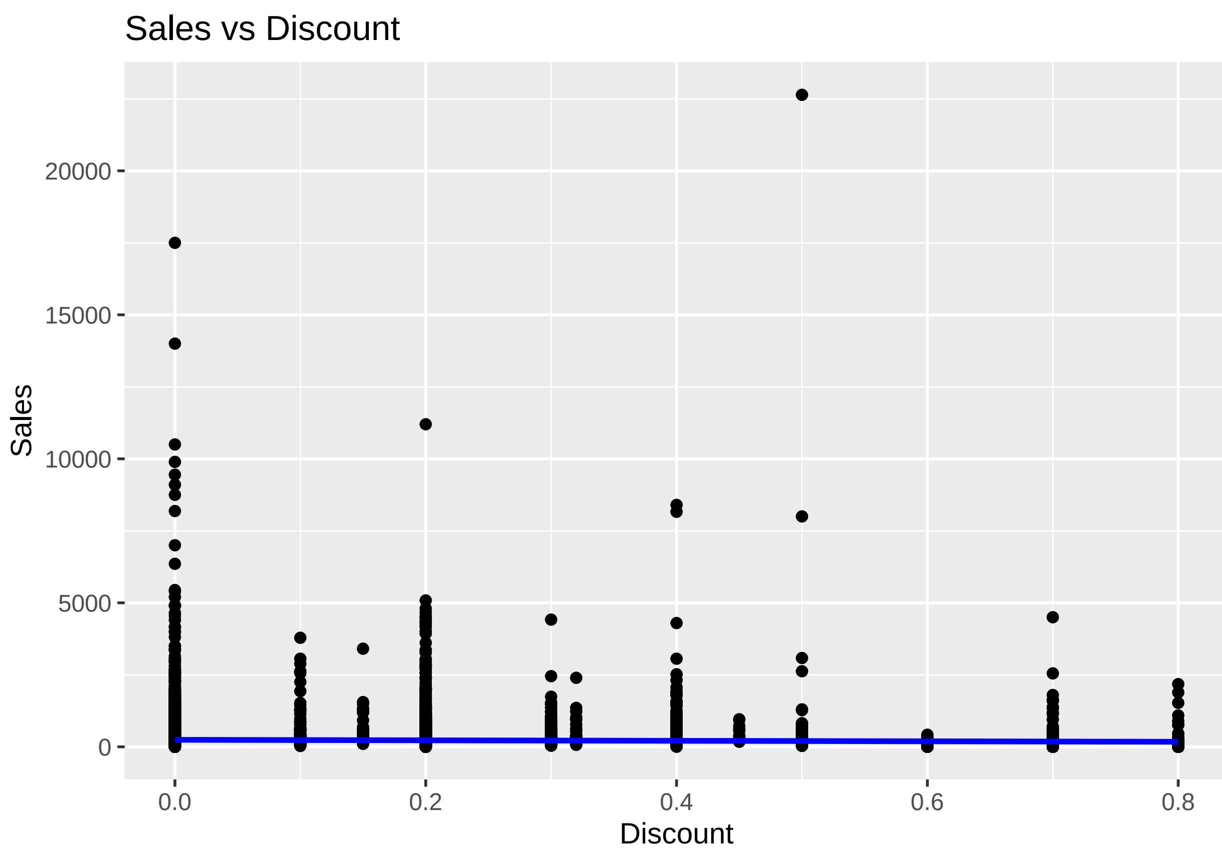
```
## [1] "Correlation coefficient: -0.0283465403114104"
```

```r
print(paste("R-squared:", model_fit$r.squared))
```

```
## [1] "R-squared: 0.00080352634762629"
```

```r
# Scatter plot with regression line
ggplot(Orders_Processing, aes(x = Discount, y = Sales)) +
  geom_point() +
  geom_smooth(method = "lm", color = "blue") +
  labs(title = "Sales vs Discount", x = "Discount", y = "Sales")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
# Calculating the correlation between Discount and Profit
discount_profit_corr <- cor(Orders_Processing$Discount, Orders_Processing$Profit, use = "complete.obs")

# Fit a linear regression model
model <- lm(Profit ~ Discount, data = Orders_Processing)
model_summary <- summary(model)
model_fit <- glance(model)

# Print correlation coefficient and R2
print(paste("Correlation coefficient between Discount and Profit:", discount_profit_corr))
```

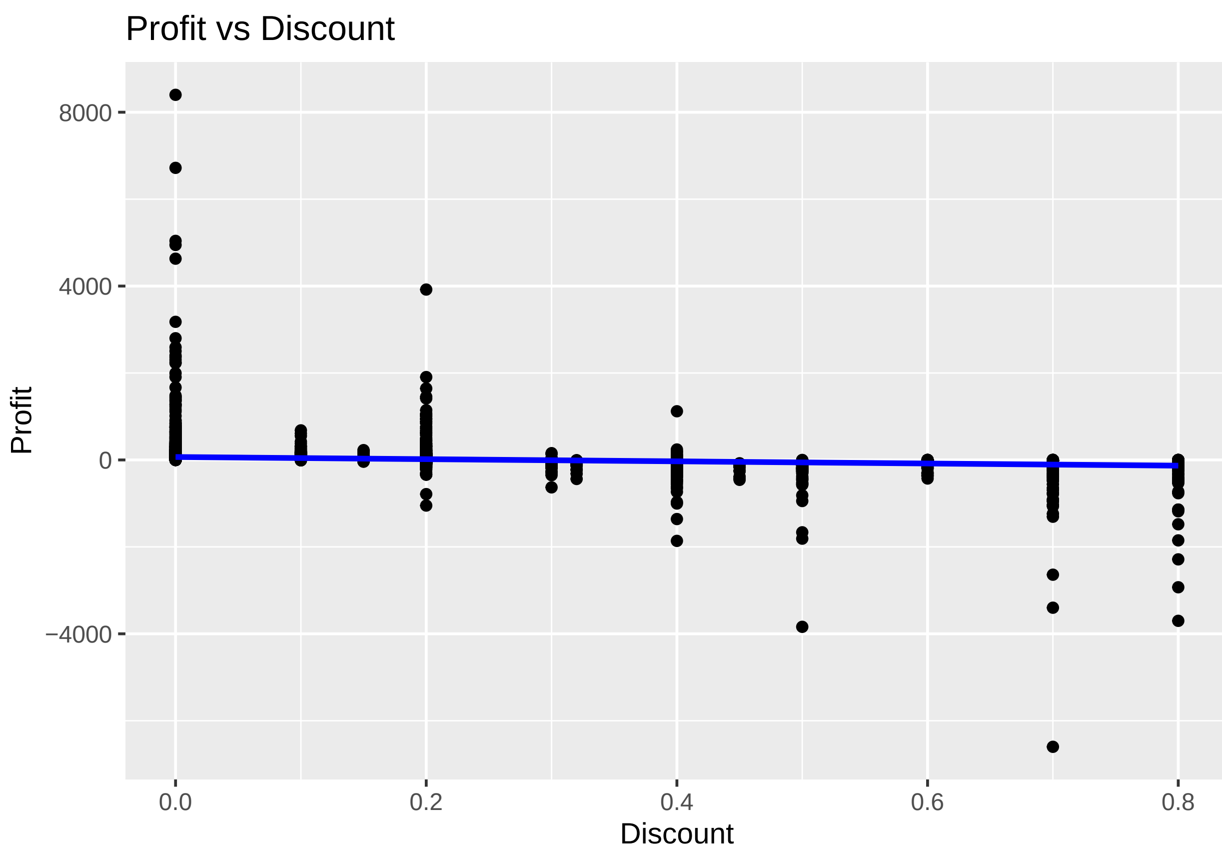
```
## [1] "Correlation coefficient between Discount and Profit: -0.219529377992477"
```

```r
print(paste("R-squared of the model:", model_fit$r.squared))
```

```
## [1] "R-squared of the model: 0.0481931478017577"
```

```r
# Scatter plot with regression line for Discount vs Profit
ggplot(Orders_Processing, aes(x = Discount, y = Profit)) +
  geom_point() +
  geom_smooth(method = "lm", color = "blue") +
  labs(title = "Profit vs Discount", x = "Discount", y = "Profit")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



In this section, we analyze the correlation between discounts and sales. Our intuition led us to believe that the correlation should always be positive, even if not very high, but positive nonetheless. However, upon analyzing the data over these four years, it seems that discounts should be eliminated, as the correlation is very weakly negative, very close to 0, at -0.03.

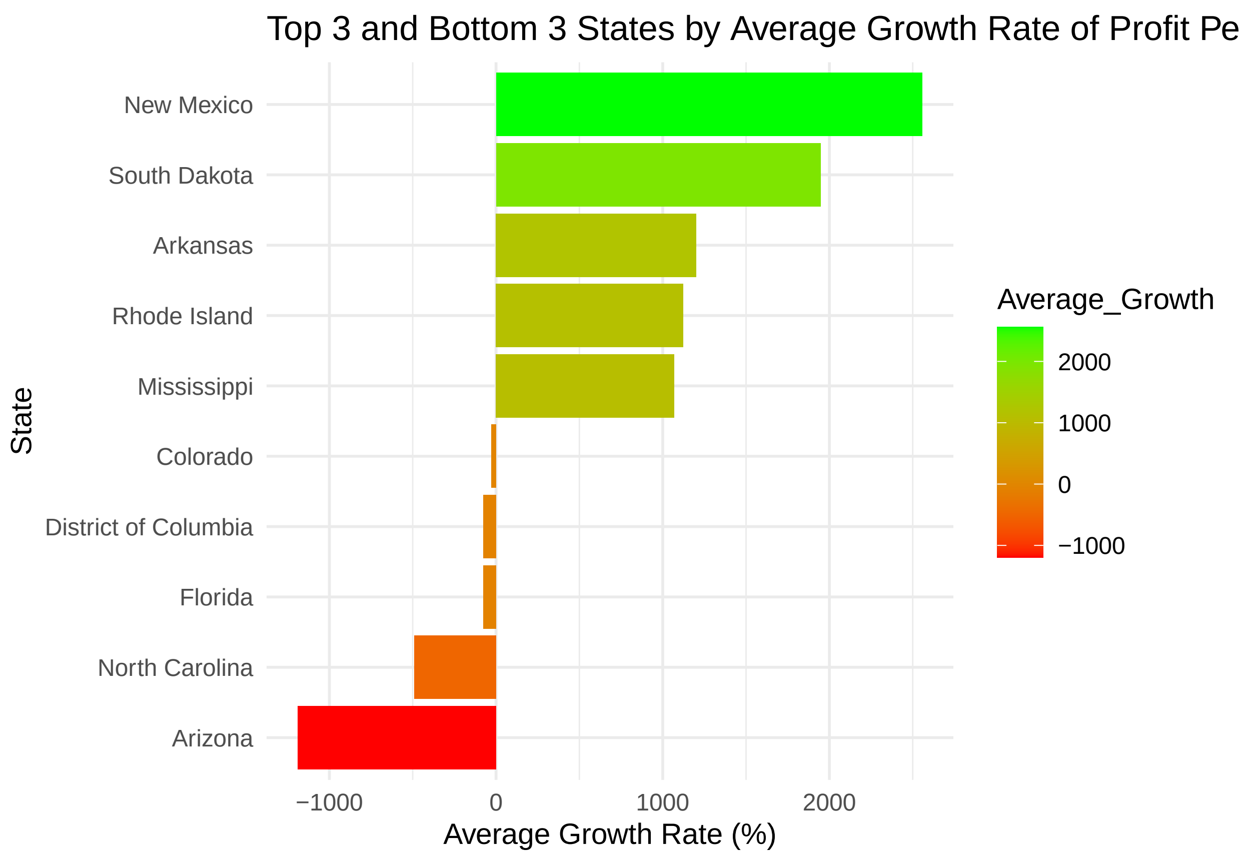
As expected, the correlation between profits and discounts is slightly negative at -0.22. This is because greater discounts lead to narrower profit margins for the company. Discounts are usually a company's strategy to increase sales at the expense of relative or absolute profit loss, but in this case, it doesn't seem to work. It's worth noting that the R-squared value is 0 for sales and 0.05 for profits, indicating that discounts have no impact on either of these variables. Therefore, they should be eliminated, as they represent a hindrance to AEKI.

4.3 GROWTH RATE OF PROFTS

```
# Assuming you have already calculated the yearly_state_profit_per_capita

state_profit_per_capita <- Orders_Processing %>%
  group_by(State, Year) %>%
  summarise(Total_Profit = sum(Total_Order_Profit, na.rm = TRUE),
            Population = mean(Population2021, na.rm = TRUE)) %>%
  mutate(Profit_Per_Capita = Total_Profit / Population)
```

```
## `summarise()` has grouped output by 'State'. You can override using the
## `.groups` argument.
```

```r
# Calculate the growth rate of profit per capita for each state
growth_rate_profit_per_capita <- state_profit_per_capita %>%
  arrange(State, Year) %>%
  group_by(State) %>%
  mutate(Profit_Per_Capita_Growth = (Profit_Per_Capita / lag(Profit_Per_Capita) - 1) * 100) %>%
  na.omit() %>%
  ungroup()

# Calculate average growth rate for each state
average_growth_rate <- growth_rate_profit_per_capita %>%
  group_by(State) %>%
  summarise(Average_Growth = mean(Profit_Per_Capita_Growth, na.rm = TRUE)) %>%
  ungroup()

# Select top 3 and bottom 3 states based on average growth rate
top_bottom_growth_states <- average_growth_rate %>%
  arrange(desc(Average_Growth)) %>%
  slice(c(1:5, (n()-4):n()))

# Plotting
ggplot(top_bottom_growth_states, aes(x = reorder(State, Average_Growth), y = Average_Growth, fill = Aver
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top 3 and Bottom 3 States by Average Growth Rate of Profit Per Capita",
       x = "State",
       y = "Average Growth Rate (%)") +
  theme_minimal() +
  scale_fill_gradient(low = "red", high = "green")  # Color gradient for visual appeal
```




```r
# Save the plot
ggsave("growth_rate_profit_per_capita_plot.png", width = 10, height = 8)
```

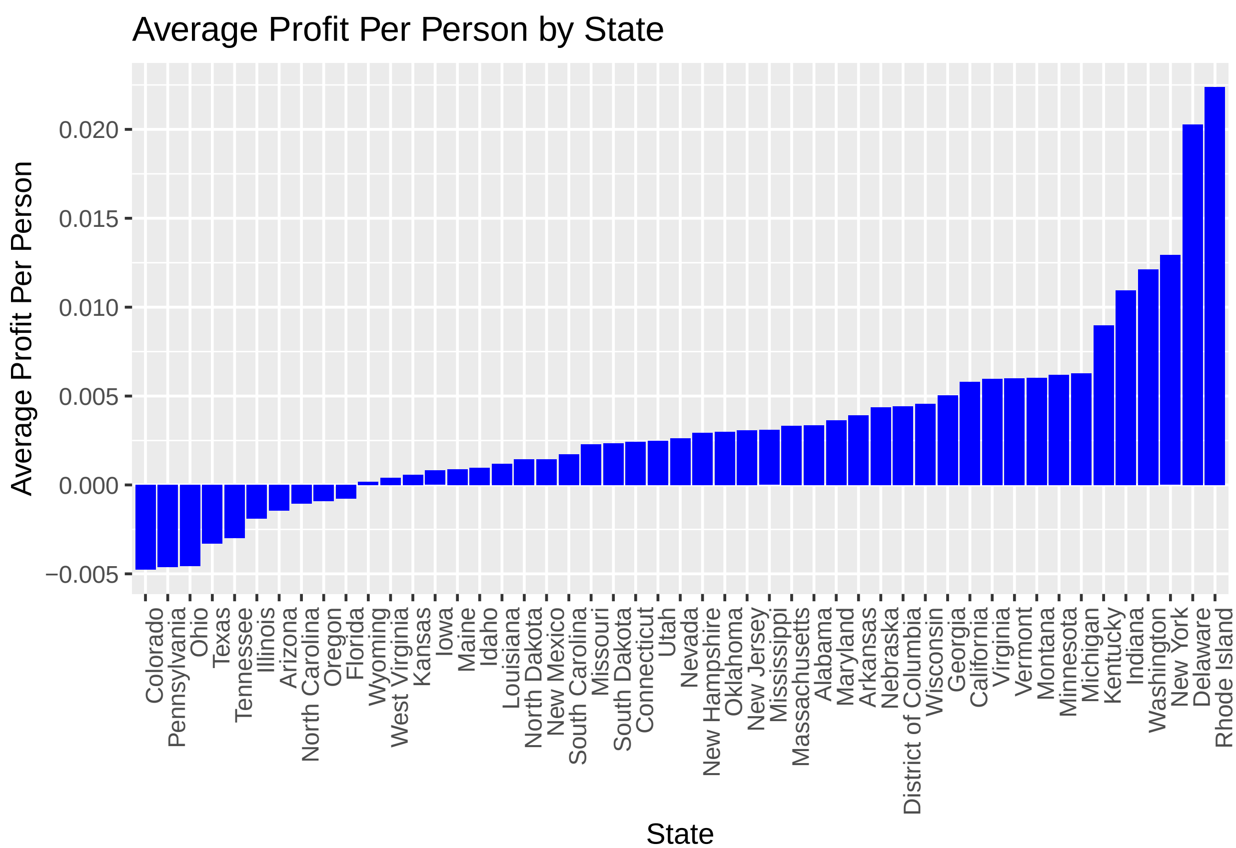
In this graph, we can observe the states where the percentage growth of profit per capita is the highest and lowest. Specifically, it shows the top 5 and bottom 5 states. This helps us see where we seem to be succeeding and where we should continue to invest, as well as areas where we may not be doing well.

It's important to note that the significant percentage growth observed in the first and last positions is due to the fact that the profit for some states is close to 0 or even negative. According to the graph, it appears that we should continue to invest in and focus on the states of Arkansas, Rhode Island, and Mississippi, where we observe growth close to 5000%. On the other hand, we should consider withdrawing from Arizona, which has a decrease of over 3000%. Additionally, it would be interesting to investigate our business activities in North Carolina and Tennessee.

4.4 PROFIT PER CAPITA FOR EACH STATE

```r
# Average profti per person dataset
result <- Orders_Processing %>%
  group_by(State) %>%
  summarise(Total_Profit = sum(Total_Order_Profit),
            Avg_Population = mean(Population2021)) %>%
  mutate(Average_Profit_Per_Person = Total_Profit / Avg_Population)

# Plot the Average Profit Per Person for each State
```

```r
ggplot(result, aes(x = reorder(State, Average_Profit_Per_Person), y = Average_Profit_Per_Person)) +
  geom_bar(stat = "identity", fill = "blue") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(title = "Average Profit Per Person by State",
       x = "State",
       y = "Average Profit Per Person")
```

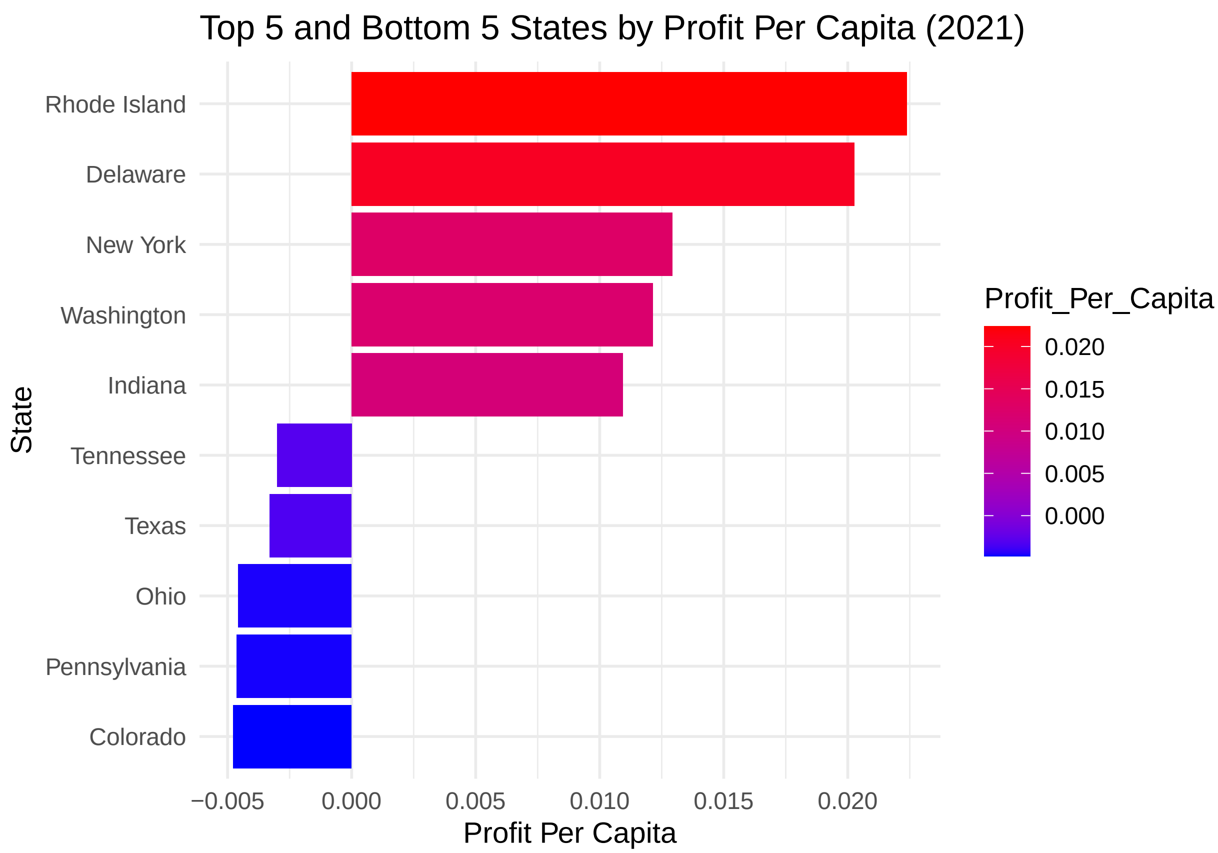


In this graph, we can see the profit per person for the company in different states, taking into account the total profit and the population in 2021. There are 10 states where the profit is negative, starting from Colorado to Oregon. Beyond that point, the profit is positive, with Delaware and Rhode Island having the highest profit per person. These states are known for their wealth, so it's not surprising to see them at the top.

4.5 HIGH 5 AND LOW 5

```r
# Orders_Processing <- read.csv("path_to_your_Orders_Processing_file.csv")

# Calculate total profit for each state and then divide by the population of 2021
state_profit_per_capita <- Orders_Processing %>%
  group_by(State) %>%
  summarise(Total_Profit = sum(Total_Order_Profit, na.rm = TRUE),
            Population = mean(Population2021, na.rm = TRUE)) %>%
  mutate(Profit_Per_Capita = Total_Profit / Population)
```

```r
# Arrange the data by Profit_Per_Capita
arranged_data <- state_profit_per_capita %>%
  arrange(desc(Profit_Per_Capita))

# Select the top 5 and bottom 5 states
top_bottom_states <- arranged_data %>%
  slice(c(1:5, (n()-4):n()))

# Plotting
ggplot(top_bottom_states, aes(x = reorder(State, Profit_Per_Capita), y = Profit_Per_Capita, fill = Profi
  geom_bar(stat = "identity") +
  coord_flip() +  # Flips the axes for better readability
  labs(title = "Top 5 and Bottom 5 States by Profit Per Capita (2021)",
       x = "State",
       y = "Profit Per Capita") +
  theme_minimal() +
  scale_fill_gradient(low = "blue", high = "red")  # Color gradient for visual appeal
```

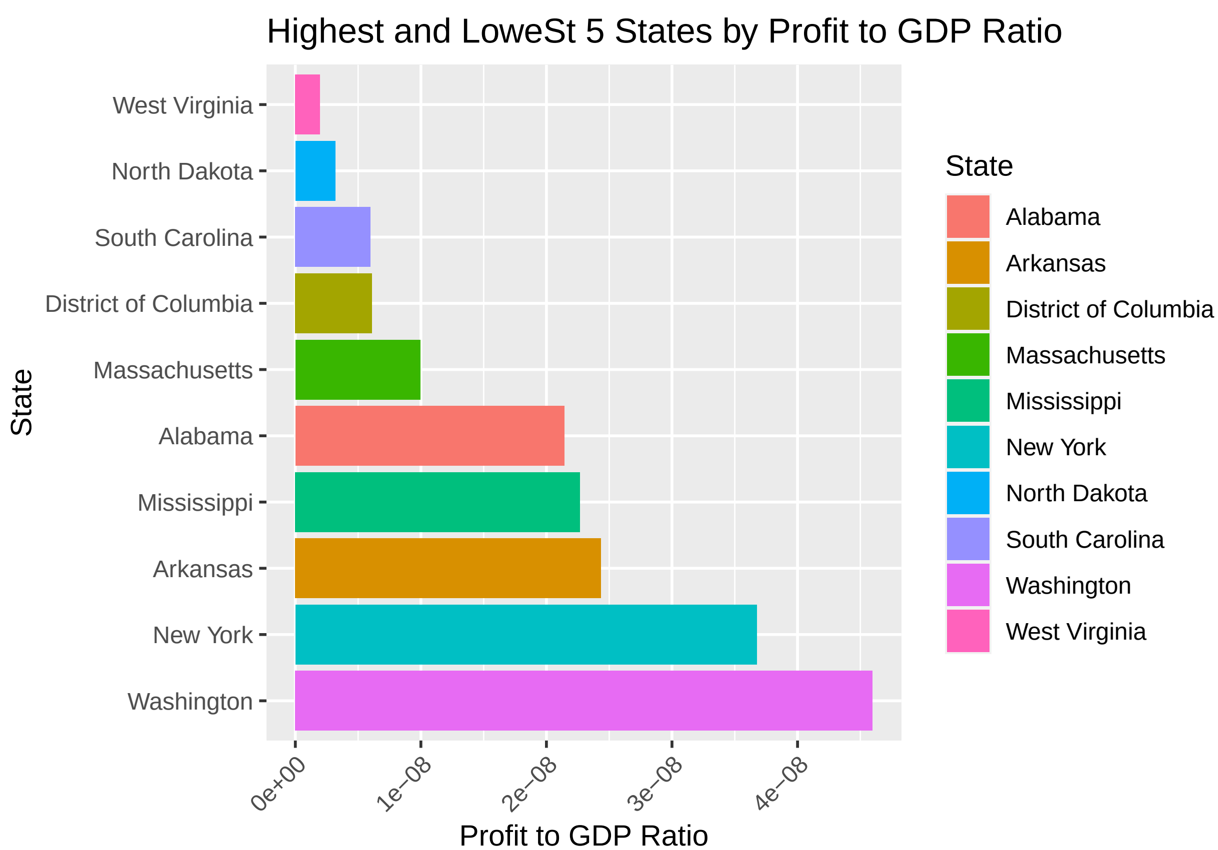


In this graph, we can see the profit-to-citizen ratio of AEKI in different states, specifically the top 5 and bottom 5. It appears that we should continue investing in Rhode Island and Delaware, as the ratio is very high in those states. However, in Colorado, Pennsylvania, Ohio, Texas, and Tennessee, the ratio is negative, indicating that we should consider divesting and revising our strategic plan in those locations.

4.6 PROFIT TO GDP FOR EACH STATE

```r
# Load the GDP dataset
gdp_data <- read_excel("/Users/hugogonzalez/Desktop/BIDA /DATA ANALISYS /GDP_PER.xlsx")

# Calculate Profit Per Person for each state
profit_per_person <- Orders_Processing %>%
  group_by(State) %>%
  summarise(Total_Profit = sum(Profit, na.rm = TRUE),
            Population = mean(Population2021, na.rm = TRUE),
            Profit_Per_Person = Total_Profit / Population) %>%
  ungroup()

# Identify common columns
common_columns <- intersect(names(result), names(gdp_data))

gdp_data <- gdp_data %>%
  rename(State = `State or Federal District`)

# Identifying the common columns
common_columns <- intersect(names(result), names(gdp_data))

# Adding the specific column 'Nominal GDP per Capita 2022' from gdp_data
additional_columns <- c(common_columns, "Nominal GDP per Capita 2022")

# Subsetting the results dataset to include only the common columns
results_common <- result[, common_columns]

# Subsetting the gdp_data dataset to include common columns and the additional column
gdp_data_common <- gdp_data[, additional_columns]

# Merging the datasets on the 'State' column
merged_data <- merge(results_common, gdp_data_common, by = "State")

# merge datasets of profit per person and gdp per capita
gdp_person_state <- merge(merged_data, profit_per_person, by = "State")

# Remove the dollar sign ('$') and any commas, then convert to numeric
gdp_person_state$`Nominal GDP per Capita 2022` <- as.numeric(gsub("[\\$,]", "", gdp_person_state$`Nomina

# Create a new column with the ratio
gdp_person_state$Profit_GDP_Ratio <- gdp_person_state$Profit_Per_Person / gdp_person_state$`Nominal GDP

# Sort the data frame based on Nominal GDP per Capita 2022 in descending order
sorted_data <- gdp_person_state[order(-gdp_person_state$`Nominal GDP per Capita 2022`), ]

# Select the highest 5 and lowest 5 rows
highest_5 <- head(sorted_data, 5)
lowest_5 <- tail(sorted_data, 5)

# Combine the highest and lowest 5 rows into a single data frame
combined_data <- rbind(highest_5, lowest_5)
```

```r
# Create a bar plot to visualize the selected values
ggplot(combined_data, aes(x = reorder(State, -Profit_GDP_Ratio), y = Profit_GDP_Ratio, fill = State)) +
  geom_bar(stat = "identity") +
  labs(title = "Highest and LoweSt 5 States by Profit to GDP Ratio",
       x = "State",
       y = "Profit to GDP Ratio") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_discrete() +
  coord_flip()
```



In this graph, we can observe the ratio of Aeki's corporate profit to the per capita GDP of the different states in 2022. The per capita GDP data has been extracted from Wikipedia. This ratio is very useful for identifying where the company has the greatest market opportunity, as there is a substantial difference among the states in the United States.

The states with the highest investment opportunity and where strong investments should be considered are West Virginia, North Dakota, South Carolina, and the District of Columbia. On the other hand, it seems that the states of Washington and New York may be saturated, possibly due to intense competition in the business sector.

5 CONCLUSIONS

5.1 CONCLUSION SUMMARY

5.1.1 Product Pricing and Category Analysis:

- **Price Range:** There's a wide range in product prices across categories, with the median value in most sub-categories falling below $250.
- **Notable Categories:** 'Copiers' and 'Bookcases' have significant outliers, with some products exceeding the $3,000 mark.
- **Yearly Trends:** Outliers were closer to the median in 2014 and 2015, indicating a narrower price range compared to 2016 and 2017.

5.2.2 Sales and Profit Correlation with State Population:

- **Sales Correlation:** A strong positive correlation (0.88) with state populations suggests increased sales in more populous states.
- **Profits Correlation:** Weaker correlation (0.38) with profits, hinting at variances in product preferences and economic conditions across states.

5.3.3 Impact of Discounts on Sales and Profits:

- **Sales Impact:** Negligible negative correlation (-0.03) with discounts, showing little effect on boosting sales.
- **Profits Impact:** Slight negative correlation (-0.22), suggesting discounts marginally reduce profits.

5.4.4 State-Specific Profit Analysis:

- **Top Performers:** Arkansas, Rhode Island, Mississippi show significant profit growth.
- **Underperformers:** Arizona shows a substantial decrease, while states like Colorado, Pennsylvania, Ohio, Texas, Tennessee have negative profit-to-citizen ratios.
- **Recommendation:** Intensify efforts in high-growth states and reassess strategies or divest in underperforming or negative ratio states.

5.5.5 Corporate Profit vs. Per Capita GDP Analysis:

- **High Opportunity States:** West Virginia, North Dakota, South Carolina, District of Columbia exhibit potential for market expansion.
- **Saturated Markets:** States like Washington and New York show high competition and market saturation.

5.2 Recommendations for AEKI’s Management:

5.2.1 Prioritize High-Value Product Segments:

- Focus on products in the ‘Copiers’ and ‘Bookcases’ categories, where significant pricing outliers suggest higher profitability.
- Develop premium marketing campaigns and display strategies for these high-value products, especially considering the notable peak in product pricing in 2017.

5.2.2 Strategic Expansion in Populous States:

- Leverage the strong sales correlation (0.88) with state populations to intensify marketing and distribution in populous states.
- Tailor product offerings to align with state-specific preferences and economic conditions, acknowledging the weaker profit correlation (0.38).

5.2.3 Rethink Discount Strategies:

- Given the negligible impact of discounts on sales (-0.03 correlation) and slight negative impact on profits (-0.22 correlation), gradually phase out blanket discount policies.
- Experiment with targeted promotions based on customer segments and purchasing behaviors instead of across-the-board discounts.

5.2.4 Focus on States with High Profit Growth:

- Increase investment in states like Arkansas, Rhode Island, and Mississippi, which show growth close to 5000%.
- Consider expanding customer service, distribution channels, and marketing efforts in these high-growth areas.

5.2.5 Address Underperformance in Specific States:

- Conduct a comprehensive review of operations in Arizona, considering its over 3000% decrease in profits, and in states with negative profit-to-citizen ratios like Colorado and Oregon.
- Explore strategies such as product line adjustments, cost reduction measures, or even exiting these markets if they remain unprofitable.

5.2.6 Capitalize on High Opportunity States:

- Identify expansion opportunities in states like West Virginia, North Dakota, South Carolina, and the District of Columbia, where there’s potential for significant market growth.
- Develop market entry or expansion strategies that consider the unique economic conditions and consumer demographics of these regions.

5.2.7 Adapt to Saturated Markets:

- In saturated markets like Washington and New York, differentiate AEKI's product offerings to stand out in competitive environments.
- Implement innovative marketing strategies and consider niche market penetration to maintain a strong presence in these regions.

5.2.8 Customized Product and Marketing Strategies:

- Use data-driven insights to understand customer preferences in different states, especially focusing on wealthier states and those with high corporate profit-to-GDP ratios.
- Develop state-specific marketing campaigns that resonate with local consumer needs and preferences.

5.2.9. Continuous Market Analysis:

- Regularly analyze market trends, consumer behavior, and economic indicators to stay ahead of changes and adapt strategies accordingly.
- Use data analytics to forecast demand, optimize inventory management, and identify emerging opportunities.