Global Sales Own Project

Yuliana Naddaf

2024-06-04

1. DATA SET DESCRIPTION:

The data set used in this project is "Global Superstore Orders 2016", which includes various details about the store's sales transactions from 2012 to 2015. It contains information about orders, customers, products, and various sales metrics.

2. VARIABLES:

- *Order_Date:* The date the order was placed.

- *Ship_Date:* The date the order was shipped.

- *Customer_ID:* Unique identifier for customers.

- *Product_ID:* Unique identifier for products.

- *Category:* The main category of the product (for example, Furniture, Technology).

- *Sub_Category:* Subcategories within each main category (e.g. Chairs, Telephones).

- *Sales:* The sales amount for each order.

- *Quantity:* The quantity of products ordered.

- *Discount:* The discount applied to the order.

- *Profit:* The profit obtained from each order.

- *Shipping_Cost:* The shipping cost of the order.

- *Other variables* include customer names, order priority, and regions.

3. OBJECTIVES OF THE PROJECT:

The primary goal of this project is to analyze sales data to gain insight into sales performance, customer behavior, and profitability trends. The analysis will help identify key factors that influence sales and profits, allowing data-driven decision making to improve business strategies.

4. DATA LOADING AND INITIAL EXPLORATION

- *Data Loading:* We use the `readxl` library to read the Excel file "Global Superstore Orders 2015.xlsx". This library is very useful for working with Excel files in R.

- *Initial Exploration:* Once the data is loaded, we use functions like `head()`, `summary()`, and `str()` to see the first rows of the dataset, get summary statistics, and understand the structure of the data (types of variables and their content).

5. DATA CLEANING

- *Identifying Outliers:* We use the Interquartile Range (IQR) method to identify and remove outliers in key variables such as Sales, Profit, and Shipping Cost.

Profit Data Outliers

The interquartile range (IQR) for sales is 220.2946, which indicates the variability of sales between the 25th and 75th percentiles of the data. To identify outliers, boundaries are established based on the IQR. The lower bound is -299.6832. This means that any sales value below -299.6832 is considered an outlier.The upper bound is 581.4951 determines that any sales value upper this values is an outlier.

Profit Data Outliers

For the profit data, the IQR is 26.6748. The lower bound for identifying outliers is -40.0254, and the upper bound is 66.6738. Any profit values outside this range are considered outliers.

Shipping Cost Data Outliers

Finally, for the shipping cost data, the IQR is 9.325. The lower bound is -12.0075, and the upper bound is 25.2925. Shipping cost values outside this range are considered outliers and are filtered out.

these steps ensure that the data used for further analysis is clean and free from extreme values that could affect the results. This improves the accuracy and reliability of any statistical or predictive models applied to the dataset.

6. EXPLORATORY DATA ANALYSIS (EDA)

Exploratory Data Analysis (EDA) is the process of analyzing datasets to summarize their main characteristics, often using visual methods. This helps to discover patterns, detect outliers, and check assumptions.

Initial EDA (Before Data Cleaning):

- *Initial Data Visualization:* Before cleaning the data, we create exploratory graphs to understand the initial distribution and detect outliers. For example, boxplots are created for Sales, Discounts, Shipping Costs, and Profits.

Post-Cleaning EDA (After Data Cleaning):

- *Cleaned Data Visualization:* After cleaning the data and removing outliers, we repeat the exploratory graphs to confirm that the anomalies data has been removed and to better understand the distribution of the cleaned data.

- *Category Summaries:* We create summaries by category to better understand the behavior of Sales and Profits in different market segments.

7. FEATURE ENGINEERING

Feature engineering is an important process in data analysis and building predictive models. It involves creating new variables or transforming existing ones to improve the model's predictive power. In the dataset of the "Global Superstore Orders 2016", various feature engineering steps were performed to capture patterns and trends not obvious in the original variables. Here the details of the steps:

1) Creating Temporal Features:

-*Year and Month of Order:* Extracted the year (Year) and month (Month) from the order date (Order_Date) to capture seasonal and annual sales trends.

-*Day of the Week:* Extracted the day of the week (DayOfWeek) from the order date to identify sales patterns throughout the week.

2) Aggregated Historical Features:

-*Past Sales:* Calculated the cumulative sum of sales (Past_Sales) for each customer to understand their historical buying behavior.

-*Rolling Average Sales*: Used a rolling average (Rolling_Average_Sales) to smooth out sales fluctuations and capture short-term trends.

3) Customer Behavior Features:

-*Purchase Frequency:* Counted the number of purchases made by each customer (Purchase_Frequency), indicating customer loyalty.

-*Customer Lifetime Value:* Calculated the cumulative sales value (Customer_Lifetime_Value) for each customer, measuring their total value to the business.

4) Discount-Based Features:

-*Average Discount by Category:* Calculated the average discount (Average_Discount) by category and created a binary variable (Discount_Above_Average) to indicate if a specific discount is above average.

-*Discount-Quantity Interaction:* Created an interaction feature (Discount_Quantity_Interaction) between discount and quantity sold.

5 and 6) Other Features:

-*Shipping Cost per Sale:* Calculated the shipping cost per unit of sale (Shipping_Cost_Per_Sale) and binarized it to identify high shipping costs (High_Shipping_Cost).

-*Cumulative Sales:* Calculated the cumulative sales sum (Total_Sales_Accum) for each customer.

-*Above-Average Sales:* Created a binary variable (Sales_Above_Average) to indicate if a sale is above average.

8. DATA MODELING GMB

Data modeling create using statistical and machin learning techniques to build models that can predict future outcomes based on historical data. In this project, were used some approachs to model sales and profits with the following steps

a) Splitting Data into Training and Test Sets:

***Time-Based Split:* Divided the data into training sets (2012-2014) and test sets (2015) to evaluate model performance on unseen data.**

b) Machine Learning Models:

***Generalized Boosted Models (GBM):* Used a GBM model to predict profits (Profit). This model is suitable for capturing complex relationships and handling non-linear interactions.**

Applying and Interpretation of GBM Model Results

- Target Variable: Profit

- Predictor Variables: All variables except Orde_ID, Order_Date, Ship_Date, Customer_ID, City, State, Product_ID, and Product_Name, plus Month, Year, Past_Sales, Rolling_Average_Sales, Purchase_Frequency, Discount_Above_Average, Total_Sales_Accum, Sales_Above_Average, and Shipping_Cost_Per_Sale.

Model Settings:

- *Distribution:* "gaussian" (for regression problems)

- *Number of Trees:* 600

- *Interaction Depth:* 4

- *Learning Rate:* 0.01

- *Cross-Validation Folds:* 5

- *Minimum Observations per Node:* 10

Importance of Features

The analysis showed which variables had the most impact on predicting `Profit`. The most important variables were:

1. *Discount* (32.22%)

2. *Sales* (25.29%)

3. *Customer_Name* (24.16%)

Other significant features founded were:

4. *Cost_Good* (8.78%)

5. *Original_Price* (2.94%)

6. *Past_Sales* (2.64%)

And less important features such as:

- *Category*, *Average_Discount*, *Customer_Lifetime_Value*, among others, with a relative influence of 0%.

Applying the Model to the Test Set and computing the RMSE to evaluate the model's perfomance I had the followings result:

RMSE: 14.48 indicates the average size of the prediction error. A lower RMSE is better and suggests a more accurate model.

The range of Profit in the dataset was determined and goes from -40.02 to 66.66

Conclusion

The range of profit in the data varies from -40.02 to 66.66, indicating that the largest observed losses were -40.02 and the highest gains were 66.66. The RMSE (Root Mean Squared Error) of the model is 14.48, which means that, on average, the model's predictions diverge by 14.48 units from the actual profit values.

This RMSE is relatively small compared to the total range of profits (106.68 units), suggesting that the model performs well. Although the model does not predict the exact profit every time, its prediction errors are reasonably small in relation to the total variability of the observed profits. In summary, the model can predict profits with acceptable accuracy, and its prediction errors are manageable.

9. DATA MODELING LINEAR REGRESSION

Multiple Linear Regression: **Applied multiple linear regression to analyze the impact of variables like discount, sub-category and sales on profits.**

Interpretation of the results Linear Regression 1

The coefficient for Discount is highly significant and negative (-5487.00), confirming that higher discounts significantly reduce profits.Several coefficients for the interactions of subcategory and sales are also significant, suggesting that there are differences in the impact of the discount depending on the subcategory.

This model has a better fit than the following regression linear, as indicated by the higher adjusted R-squared value (0.528 vs. 0.3214).indicating that approximately 52.8% of the variability in benefits can be explained by the predictor variables included in the model.

On the other hand The Residual Standard Error of 12.55 indicates that the typical prediction error made by the model is about 12.55 units of Profit, we can consider a RSE of 12.55 is relatively small compared to the total width of the earnings range 106.68 units (min -40.02 and max 66.66. This indicates that, on average, the model prediction errors are much smaller than the total variability of observed benefits.

The F-statistic of 573.7, along with the degrees of freedom (67 for the numerator and 34231 for the denominator), and the extremely low p-value (< 2.2e-16), indicates that the regression model is statistically significant. This means that the group of predictors in the model collectively contribute significantly to explaining the variability in the response variable, which in this case is Profit.

In simpler words, the model does a good job explaining the data, and the factors we used to predict Profit work well. The very low p-value shows that it's very unlikely these results happened by chance, which makes us trust the model's predictions.

Interpretation of the results Linear Regression 2

The discount coefficient is highly significant and negative (-5001.00), indicating that for each additional unit of discount, the profit decreases by 5001 units, reconfirming once again that the discount variable has an important impact in the variability of the profit.

The F-statistic (1355) with degrees of freedom of 12 and 34286 and an very low p-value (< 2.2e-16) indicates that the regression model is globally significant. This means the model is useful for explaining the variability in the response variable (profit) and it is unlikely that all the coefficients in the model are zero.

Nevertheless the R-squared value (0.3214).indicating that approximately 32.14% of the variability in benefits can be explained by the predictor variables included in the model wich is less than the first model lnear regression

Conclusion

**** The regression model highlights a significant negative impact of discounts on profits, while the months do not show a strong effect. The model explains a moderate proportion of the variability in profits, and while it is statistically significant, there is room for improvement in predictive accuracy.**

10. Recommendations and final Conclusions

Based on the analysis and data modeling, several actionable conclusions and recommendations were made to improve business strategies:

a) I identifying Profitable Products:

Most Profitable Sub-Categories: **Identified sub-categories with the highest profits, such as copiers and phones in the technology category. Recommended focusing marketing and sales efforts on these products.**

Least Profitable Sub-Categories: **Identified sub-categories with low or negative profits, such as tables in the furniture category. I Suggest to review pricing and cost strategies for these products.**

b) Discount Strategies:

Effective Discounts: **Analyzed discount patterns and their impact on sales and profits. Recommended adjusting discounts to maximize profits without sacrificing sales volume.**

Regional Variations: **Identified differences in discounts and profits by region. Suggested customizing discount strategies by region to optimize sales.**

c) Inventory Management:

Sales Trends: **Analyzed sales trends over time to improve inventory management and reduce storage costs.**

Demand Forecasting: **Recommended using predictive models to forecast demand and adjust inventory accordingly.**

d) Customer Behavior:

Customer Loyalty: **Identified customers with high purchase frequency and lifetime value. Suggested implementing loyalty programs to retain valuable customers.**

Customer Segmentation: **Recommended segmenting customers based on their buying behavior and tailoring marketing strategies for each segment.**

e) Shipping Cost Optimization:

```
# Project Store Global Sales 2012-2015

# 4. DATA LOADING AND INITIAL EXPLORATION

options(repos = c(CRAN = "https://cloud.r-project.org/"))
Sys.setenv(LANG = "en_US.UTF-8")

# Load necessary packages
install.packages("rmarkdown")
```

Reducing Costs: **Analyzed shipping costs and their impact on profits. We suggested to negotiate lower shipping rates and to optimize shipping routes.**

```
## Installing package into 'C:/Users/nadda/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

##
##   There is a binary version available but the source version is later:
##           binary source needs_compilation
## rmarkdown   2.26   2.27             FALSE
```

```
## installing the source package 'rmarkdown'
```

```
install.packages("knitr")
```

```
## Installing package into 'C:/Users/nadda/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

## also installing the dependency 'xfun'

##
##   There are binary versions available but the source versions are later:
##       binary source needs_compilation
## xfun    0.43   0.44              TRUE
## knitr   1.46   1.47             FALSE
##
##   Binaries will be installed
## package 'xfun' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'xfun'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\nadda\AppData\Local\R\win-library\4.2\00LOCK\xfun\libs\x64\xfun.dll to
## C:\Users\nadda\AppData\Local\R\win-library\4.2\xfun\libs\x64\xfun.dll:
## Permission denied

## Warning: restored 'xfun'

##
## The downloaded binary packages are in
##  C:\Users\nadda\AppData\Local\Temp\RtmpewMKaI\downloaded_packages

## installing the source package 'knitr'

## Warning in install.packages("knitr"): installation of package 'knitr' had
## non-zero exit status
```

```
install.packages("xfun")
```

```
## Installing package into 'C:/Users/nadda/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

##
##   There is a binary version available but the source version is later:
##      binary source needs_compilation
## xfun   0.43   0.44              TRUE
##
##   Binaries will be installed
## package 'xfun' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'xfun'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\nadda\AppData\Local\R\win-library\4.2\00LOCK\xfun\libs\x64\xfun.dll to
## C:\Users\nadda\AppData\Local\R\win-library\4.2\xfun\libs\x64\xfun.dll:
## Permission denied

## Warning: restored 'xfun'

##
## The downloaded binary packages are in
##  C:\Users\nadda\AppData\Local\Temp\RtmpewMKaI\downloaded_packages
```

```
install.packages("readxl")
```

```
## Installing package into 'C:/Users/nadda/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

## package 'readxl' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'readxl'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\nadda\AppData\Local\R\win-library\4.2\00LOCK\readxl\libs\x64\readxl.dll
## to C:\Users\nadda\AppData\Local\R\win-library\4.2\readxl\libs\x64\readxl.dll:
## Permission denied

## Warning: restored 'readxl'

##
## The downloaded binary packages are in
##  C:\Users\nadda\AppData\Local\Temp\RtmpewMKaI\downloaded_packages
```

```
install.packages("dplyr")
```

```
## Installing package into 'C:/Users/nadda/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

## package 'dplyr' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'dplyr'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\nadda\AppData\Local\R\win-library\4.2\00LOCK\dplyr\libs\x64\dplyr.dll
## to C:\Users\nadda\AppData\Local\R\win-library\4.2\dplyr\libs\x64\dplyr.dll:
## Permission denied

## Warning: restored 'dplyr'

##
## The downloaded binary packages are in
##  C:\Users\nadda\AppData\Local\Temp\RtmpewMKaI\downloaded_packages
```

```r
install.packages("ggplot2")
```

```
## Installing package into 'C:/Users/nadda/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

## package 'ggplot2' successfully unpacked and MD5 sums checked
##
## The downloaded binary packages are in
##  C:\Users\nadda\AppData\Local\Temp\RtmpewMKaI\downloaded_packages
```

```r
library(readxl)
library(dplyr)
```

```
##
## 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(ggplot2)
library(dplyr)

# Set working directory and load data

getwd()
```

```
## [1] "C:/Users/nadda/OneDrive/Documents/Project Machine Learning/ProjectoOwnProject"
```

```r
setwd("/Users/nadda/OneDrive/Documents/Project Machine Learning")
sales_data <- read_excel("Global Superstore Orders 2015.xlsx")
head(sales_data)
```

```
## # A tibble: 6 x 25
##   Row_ID Orde_ID   Order_Date          Ship_Date           Ship_Mode Customer_ID
##    <dbl> <chr>     <dttm>              <dttm>              <chr>     <chr>
## 1  38123 CA-2014-~ 2014-10-03 00:00:00 2014-10-10 00:00:00 Standard~ TC-2098014~
## 2  39450 CA-2015-~ 2015-03-24 00:00:00 2015-03-26 00:00:00 First Cl~ RB-1936014~
## 3  35487 CA-2015-~ 2015-11-18 00:00:00 2015-11-23 00:00:00 Standard~ HL-1504014~
## 4  40336 CA-2014-~ 2014-12-18 00:00:00 2014-12-22 00:00:00 Standard~ AB-1010514~
## 5  35395 CA-2012-~ 2012-09-22 00:00:00 2012-09-27 00:00:00 Standard~ SC-2009514~
## 6  12069 ES-2015-~ 2015-09-08 00:00:00 2015-09-14 00:00:00 Standard~ PJ-1883564
## # i 19 more variables: Customer_Name <chr>, Segment <chr>, City <chr>,
## #   State <chr>, Country <chr>, Region <chr>, Market <chr>, Product_ID <chr>,
## #   Category <chr>, Sub_Category <chr>, Product_Name <chr>, Sales <dbl>,
## #   Quantity <dbl>, Discount <dbl>, Profit <dbl>, Shipping_Cost <dbl>,
## #   Order_Priority <chr>, Original_Price <dbl>, Cost_Good <dbl>
```

```r
summary(sales_data)  # Get summary statistics which can help in identifying anomalies
```

```
##      Row_ID        Orde_ID            Order_Date
##  Min.   :    1   Length:51290       Min.   :2012-01-01 00:00:00.00
##  1st Qu.:12823   Class :character   1st Qu.:2013-06-19 00:00:00.00
##  Median :25646   Mode  :character   Median :2014-07-08 00:00:00.00
##  Mean   :25646                      Mean   :2014-05-11 21:26:49.16
##  3rd Qu.:38468                      3rd Qu.:2015-05-22 00:00:00.00
##  Max.   :51290                      Max.   :2015-12-31 00:00:00.00
##    Ship_Date                       Ship_Mode         Customer_ID
##  Min.   :2012-01-03 00:00:00.00   Length:51290       Length:51290
##  1st Qu.:2013-06-23 00:00:00.00   Class :character   Class :character
##  Median :2014-07-12 00:00:00.00   Mode  :character   Mode  :character
##  Mean   :2014-05-15 20:42:42.75
##  3rd Qu.:2015-05-26 00:00:00.00
##  Max.   :2016-01-07 00:00:00.00
##  Customer_Name        Segment              City              State
##  Length:51290       Length:51290       Length:51290       Length:51290
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    Country             Region             Market           Product_ID
##  Length:51290       Length:51290       Length:51290       Length:51290
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##    Category         Sub_Category       Product_Name           Sales
##  Length:51290       Length:51290       Length:51290       Min.   :    0.444
##  Class :character   Class :character   Class :character   1st Qu.:   30.759
##  Mode  :character   Mode  :character   Mode  :character   Median :   85.053
##                                                           Mean   :  246.491
##                                                           3rd Qu.:  251.053
##                                                           Max.   :22638.480
##     Quantity         Discount          Profit         Shipping_Cost
##  Min.   : 1.000   Min.   :0.0000   Min.   :-6599.98   Min.   :  1.002
##  1st Qu.: 2.000   1st Qu.:0.0000   1st Qu.:    0.00   1st Qu.:  2.610
##  Median : 3.000   Median :0.0000   Median :    9.24   Median :  7.790
##  Mean   : 3.477   Mean   :0.1429   Mean   :   28.61   Mean   : 26.479
##  3rd Qu.: 5.000   3rd Qu.:0.2000   3rd Qu.:   36.81   3rd Qu.: 24.450
##  Max.   :14.000   Max.   :0.8500   Max.   : 8399.98   Max.   :933.570
##  Order_Priority     Original_Price       Cost_Good
##  Length:51290       Min.   :    0.80   Min.   :   -0.64
##  Class :character   1st Qu.:   34.38   1st Qu.:   24.95
##  Mode  :character   Median :   95.04   Median :   70.27
##                     Mean   :  272.74   Mean   :  217.65
##                     3rd Qu.:  278.18   3rd Qu.:  217.98
##                     Max.   :33957.72   Max.   :35744.51
```

```r
str(sales_data)      # Check the structure to understand data types
```

```
## tibble [51,290 x 25] (S3: tbl_df/tbl/data.frame)
##  $ Row_ID        : num [1:51290] 38123 39450 35487 40336 35395 ...
##  $ Orde_ID       : chr [1:51290] "CA-2014-TC20980140-41915" "CA-2015-RB19360140-42087" "CA-2015-HL150
##  $ Order_Date    : POSIXct[1:51290], format: "2014-10-03" "2015-03-24" ...
##  $ Ship_Date     : POSIXct[1:51290], format: "2014-10-10" "2015-03-26" ...
##  $ Ship_Mode     : chr [1:51290] "Standard Class" "First Class" "Standard Class" "Standard Class" ..
##  $ Customer_ID   : chr [1:51290] "TC-209801402" "RB-193601404" "HL-150401406" "AB-101051402" ...
##  $ Customer_Name : chr [1:51290] "Tamara Chand" "Raymond Buch" "Hunter Lopez" "Adrian Barton" ...
##  $ Segment       : chr [1:51290] "Corporate" "Consumer" "Consumer" "Consumer" ...
##  $ City          : chr [1:51290] "Lafayette" "Seattle" "Newark" "Detroit" ...
##  $ State         : chr [1:51290] "Indiana" "Washington" "Delaware" "Michigan" ...
##  $ Country       : chr [1:51290] "United States" "United States" "United States" "United States" ...
##  $ Region        : chr [1:51290] "Central US" "Western US" "Eastern US" "Central US" ...
##  $ Market        : chr [1:51290] "USCA" "USCA" "USCA" "USCA" ...
##  $ Product_ID    : chr [1:51290] "TEC-CO-3691" "TEC-CO-3691" "TEC-CO-3691" "OFF-BI-4345" ...
##  $ Category      : chr [1:51290] "Technology" "Technology" "Technology" "Office Supplies" ...
##  $ Sub_Category  : chr [1:51290] "Copiers" "Copiers" "Copiers" "Binders" ...
##  $ Product_Name  : chr [1:51290] "Canon imageCLASS 2200 Advanced Copier" "Canon imageCLASS 2200 Adva
##  $ Sales         : num [1:51290] 17500 14000 10500 9893 9450 ...
##  $ Quantity      : num [1:51290] 5 4 3 13 5 14 4 5 11 9 ...
##  $ Discount      : num [1:51290] 0 0 0 0 0 0 0.2 0 0 0 ...
##  $ Profit        : num [1:51290] 8400 6720 5040 4946 4630 ...
##  $ Shipping_Cost : num [1:51290] 349 20 363 499 656 ...
##  $ Order_Priority: chr [1:51290] "Medium" "Medium" "Medium" "Medium" ...
##  $ Original_Price: num [1:51290] 17500 14000 10500 9893 9450 ...
##  $ Cost_Good     : num [1:51290] 8751 7260 5097 4448 4164 ...
```

```r
print(colnames(sales_data))
```

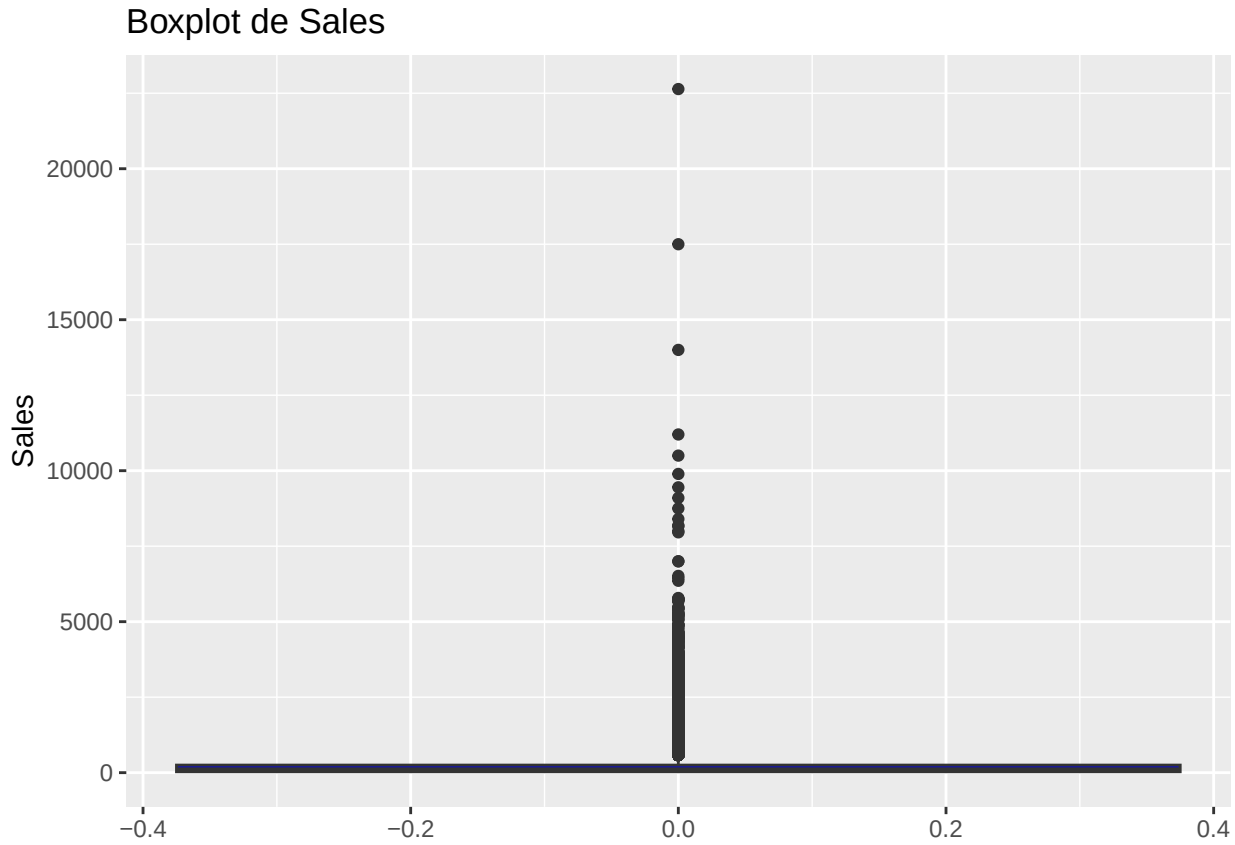
```
##  [1] "Row_ID"         "Orde_ID"        "Order_Date"     "Ship_Date"
##  [5] "Ship_Mode"      "Customer_ID"    "Customer_Name"  "Segment"
##  [9] "City"           "State"          "Country"        "Region"
## [13] "Market"         "Product_ID"     "Category"       "Sub_Category"
## [17] "Product_Name"   "Sales"          "Quantity"       "Discount"
## [21] "Profit"         "Shipping_Cost"  "Order_Priority" "Original_Price"
## [25] "Cost_Good"
```

```r
# Summarize Sales and Profit by Category
category_summary <- sales_data %>%
  group_by(Category) %>%
  summarise(
    Total_Sales = sum(Sales, na.rm = TRUE),
    Total_Profit = sum(Profit, na.rm = TRUE)
  )

# Print the summary
print(category_summary)
```

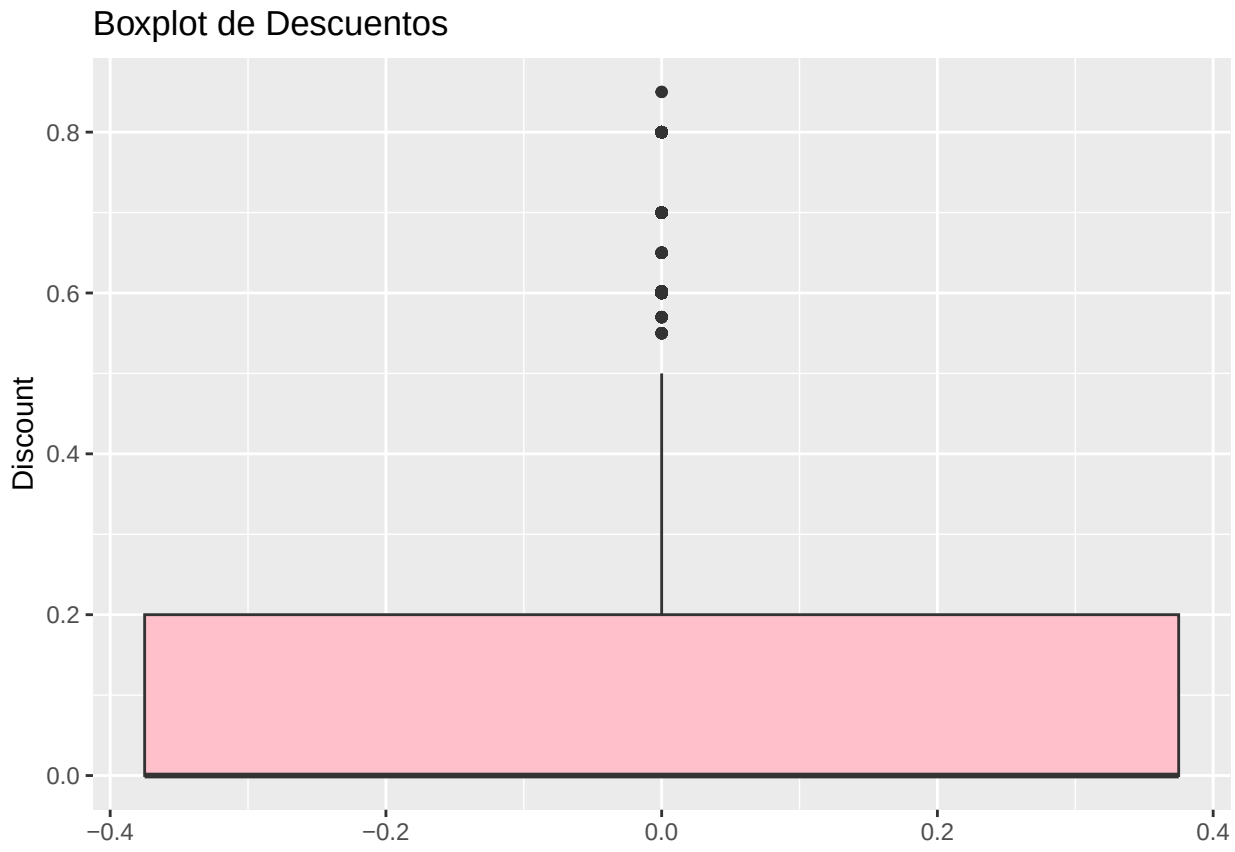
```
## # A tibble: 3 x 3
##   Category        Total_Sales Total_Profit
```

```
##   <chr>                <dbl>        <dbl>
## 1 Furniture          4110452.      285083.
## 2 Office Supplies    3787493.      518596.
## 3 Technology         4744557.      663779.
```

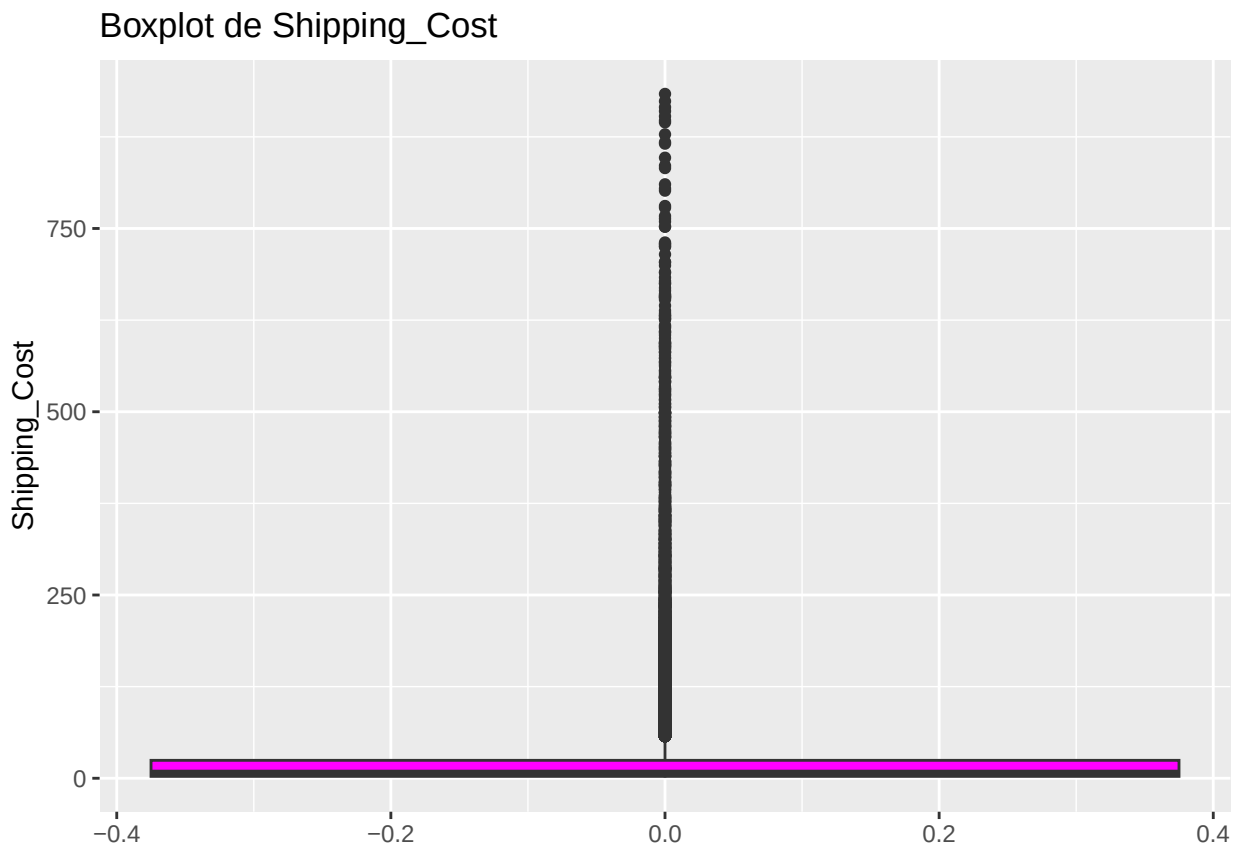
```
# The following four boxplots are essential for understanding the data's distribution, identifying outl
#OUTIERS PLOT

# Boxplot para Sales
ggplot(sales_data, aes(y = Sales)) +
  geom_boxplot(fill = "blue") +
  ggtitle("Boxplot de Sales")
```



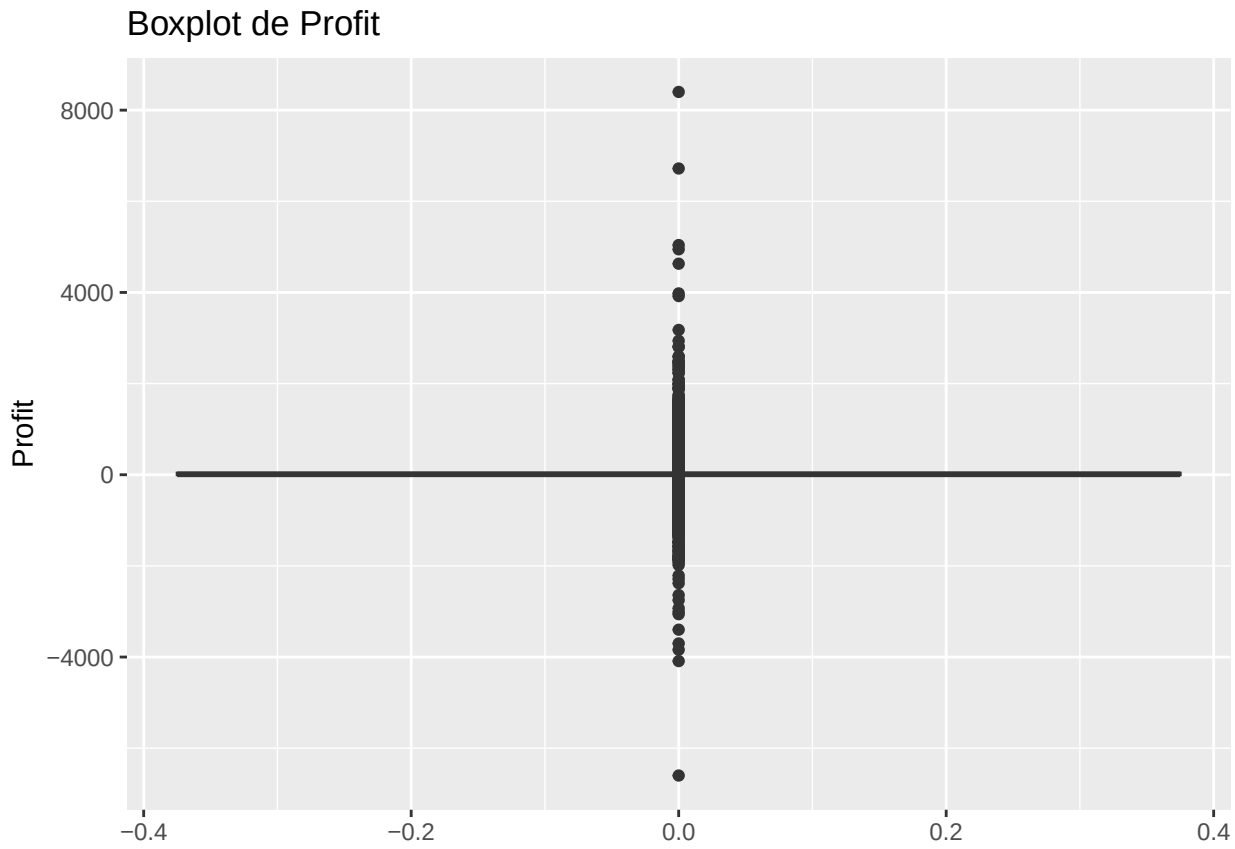

```
# Boxplot para Descuentos
ggplot(sales_data, aes(y = Discount)) +
  geom_boxplot(fill = "pink") +
  ggtitle("Boxplot de Descuentos")
```




```
# Boxplot para Shipping Cost
ggplot(sales_data, aes(y = Shipping_Cost)) +
  geom_boxplot(fill = "magenta") +
  ggtitle("Boxplot de Shipping_Cost")
```




```
# Boxplot para Profit
ggplot(sales_data, aes(y = Profit)) +
  geom_boxplot(fill = "violet") +
  ggtitle("Boxplot de Profit")
```




```
# 5. DATA CLEANING

# OUTLIER CLEANING METHOD
# Method Using Interquartile Range (IQR) for SALES

# Calculating the IQR
Q1 <- quantile(sales_data$Sales, 0.25, na.rm = TRUE)
Q3 <- quantile(sales_data$Sales, 0.75, na.rm = TRUE)
IQR <- Q3 - Q1
print(IQR)
```

```
##      75%
## 220.2946
```

```
# Defining boundaries to identify outliers
lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR
print(lower_bound)
```

```
##       25%
## -299.6832
```

```
print(upper_bound)
```

```
##      75%
## 581.4951
```

```r
# Filtering out the outliers
data <- sales_data[sales_data$Sales >= lower_bound & sales_data$Sales <= upper_bound, ]


# Method Using Interquartile Range (IQR) for PROFIT

# Calculating the IQR
Q1 <- quantile(data$Profit, 0.25, na.rm = TRUE)
Q3 <- quantile(data$Profit, 0.75, na.rm = TRUE)
IQR <- Q3 - Q1
print(IQR)
```

```
##     75%
## 26.6748
```

```r
# Defining boundaries to identify outliers
lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR
print(lower_bound)
```

```
##      25%
## -40.0254
```

```r
print(upper_bound)
```

```
##     75%
## 66.6738
```

```r
# Filtering out the outliers
data_1 <- data[data$Profit >= lower_bound & data$Profit <= upper_bound, ]

# Method Using Interquartile Range (IQR) for Shipping Cost

# Calculating the IQR
Q1 <- quantile(data_1$Shipping_Cost, 0.25, na.rm = TRUE)
Q3 <- quantile(data_1$Shipping_Cost, 0.75, na.rm = TRUE)
IQR <- Q3 - Q1
print(IQR)
```

```
##   75%
## 9.325
```

```r
# Defining boundaries to identify outliers
lower_bound <- Q1 - 1.5 * IQR
upper_bound <- Q3 + 1.5 * IQR
print(lower_bound)
```

```
##      25%
## -12.0075
```

```r
print(upper_bound)
```

```
##     75%
## 25.2925
```

```r
# Filtering out the outliers
data_2 <- data_1[data_1$Shipping_Cost >= lower_bound & data_1$Shipping_Cost <= upper_bound, ]
```

```r
# DATA FORMATTING (formatting of dataset variables)

# Formatting the data
data_2 <- data_2 %>%
  mutate(
    # Convert dates to Date type
    Order_Date = as.Date(Order_Date, format = "%Y-%m-%d"),
    Ship_Date = as.Date(Ship_Date, format = "%Y-%m-%d"),

    # Ensure numeric fields are treated as numeric
    Sales = as.numeric(gsub("[$,]", "", Sales)),
    Quantity = as.numeric(Quantity),
    Discount = as.numeric(sub("%$", "", Discount)) / 100,  # Convert percentage to decimal
    Profit = as.numeric(gsub("[$,]", "", Profit)),
    Shipping_Cost = as.numeric(Shipping_Cost),
    Original_Price = as.numeric(gsub("[$,]", "", Original_Price)),
    Cost_Good = as.numeric(gsub("[$,]", "", Cost_Good)),

    # Handle categorical data by converting them to factors
    Ship_Mode = as.factor(Ship_Mode),
    Customer_ID = as.factor(Customer_ID),
    Customer_Name = as.factor(Customer_Name),
    Segment = as.factor(Segment),
    City = as.factor(City),
    State = as.factor(State),
    Country = as.factor(Country),
    Region = as.factor(Region),
    Market = as.factor(Market),
    Product_ID = as.factor(Product_ID),
    Category = as.factor(Category),
    Sub_Category = as.factor(Sub_Category),
    Product_Name = as.factor(Product_Name),
    Order_Priority = as.factor(Order_Priority)
  )

# Verification Data Structure
str(data_2)
```

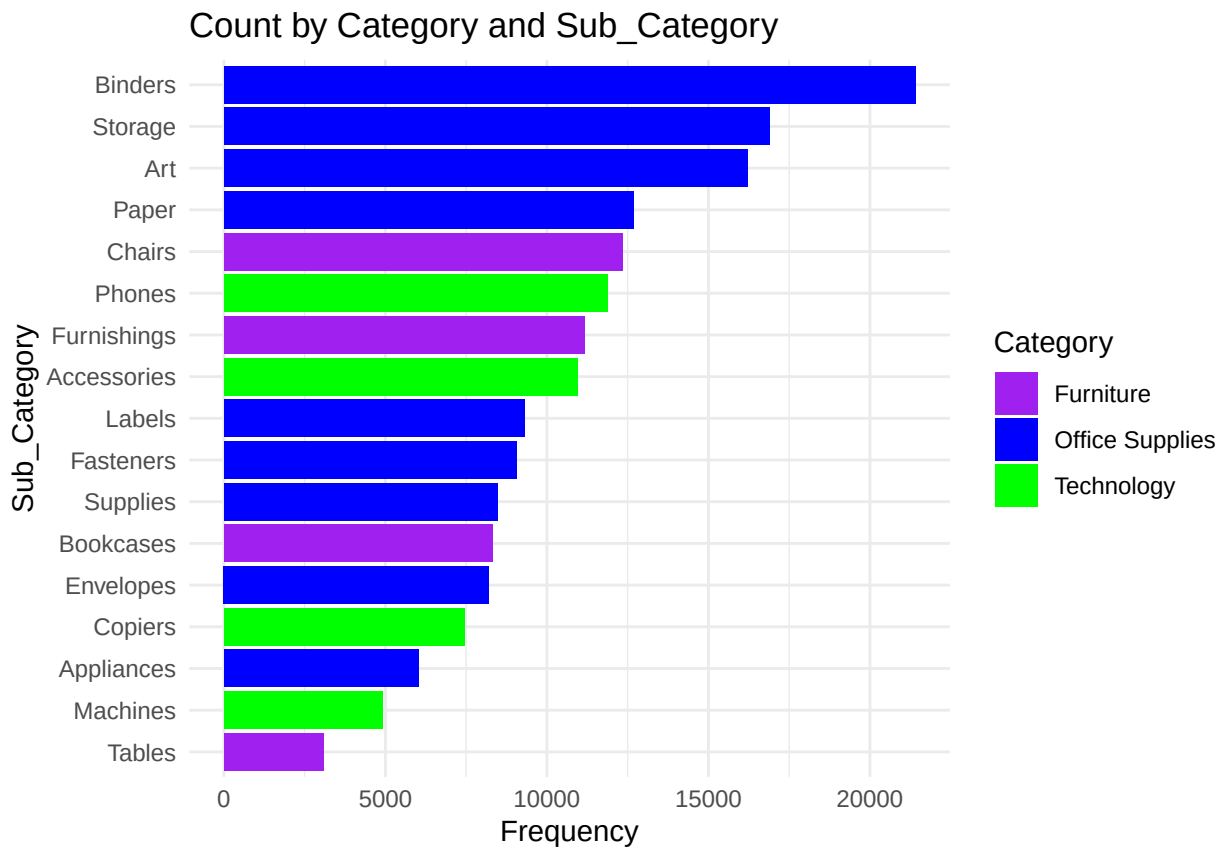
```
## tibble [34,299 x 25] (S3: tbl_df/tbl/data.frame)
##  $ Row_ID        : num [1:34299] 45663 37594 37939 46303 20344 ...
##  $ Orde_ID       : chr [1:34299] "RO-2013-SF10065107-41591" "CA-2012-AJ10780140-41270" "CA-2015-PO188
##  $ Order_Date    : Date[1:34299], format: "2013-11-13" "2012-12-27" ...
##  $ Ship_Date     : Date[1:34299], format: "2013-11-17" "2012-12-31" ...
##  $ Ship_Mode     : Factor w/ 4 levels "First Class",..: 4 4 4 4 2 4 4 4 4 4 ...
```

```
##  $ Customer_ID   : Factor w/ 14817 levels "AA-10315102",..: 12782 682 11272 13995 10448 1330 2776 104
##  $ Customer_Name : Factor w/ 796 levels "Aaron Bergman",..: 672 52 592 741 576 107 155 559 618 144 .
##  $ Segment       : Factor w/ 3 levels "Consumer","Corporate",..: 1 2 1 1 2 1 3 1 2 1 ...
##  $ City          : Factor w/ 3406 levels "Aachen","Aalst",..: 460 364 1363 1509 1908 2150 2871 2871 8
##  $ State         : Factor w/ 1063 levels "'Ajman","'Amman",..: 166 613 433 654 804 687 848 848 634 30
##  $ Country       : Factor w/ 161 levels "Afghanistan",..: 119 154 154 120 7 7 45 45 154 44 ...
##  $ Region        : Factor w/ 23 levels "Canada","Caribbean",..: 9 10 6 9 13 13 4 4 6 11 ...
##  $ Market        : Factor w/ 5 levels "Africa","Asia Pacific",..: 3 5 5 3 2 2 4 4 5 1 ...
##  $ Product_ID    : Factor w/ 3295 levels "FUR-BO-3177",..: 736 2044 1644 3018 41 2606 2698 2698 1336
##  $ Category      : Factor w/ 3 levels "Furniture","Office Supplies",..: 2 2 2 3 1 3 3 3 2 3 ...
##  $ Sub_Category  : Factor w/ 17 levels "Accessories",..: 2 13 11 12 5 1 1 1 4 7 ...
##  $ Product_Name  : Factor w/ 3295 levels "\"While you Were Out\" Message Book, One Form per Page",..
##  $ Sales         : num [1:34299] 191 142 133 266 333 ...
##  $ Quantity      : num [1:34299] 2 4 9 1 3 3 3 3 4 2 ...
##  $ Discount      : num [1:34299] 0 0 0 0 0.001 0.001 0 0 0 0 ...
##  $ Profit        : num [1:34299] 66.7 66.6 66.6 66.6 66.6 ...
##  $ Shipping_Cost : num [1:34299] 22.7 11.3 17.9 17.7 10.7 ...
##  $ Order_Priority: Factor w/ 4 levels "Critical","High",..: 2 4 2 4 4 2 4 4 4 4 ...
##  $ Original_Price: num [1:34299] 191 142 133 266 367 ...
##  $ Cost_Good     : num [1:34299] 101.2 63.8 48.7 182.2 289.3 ...
```

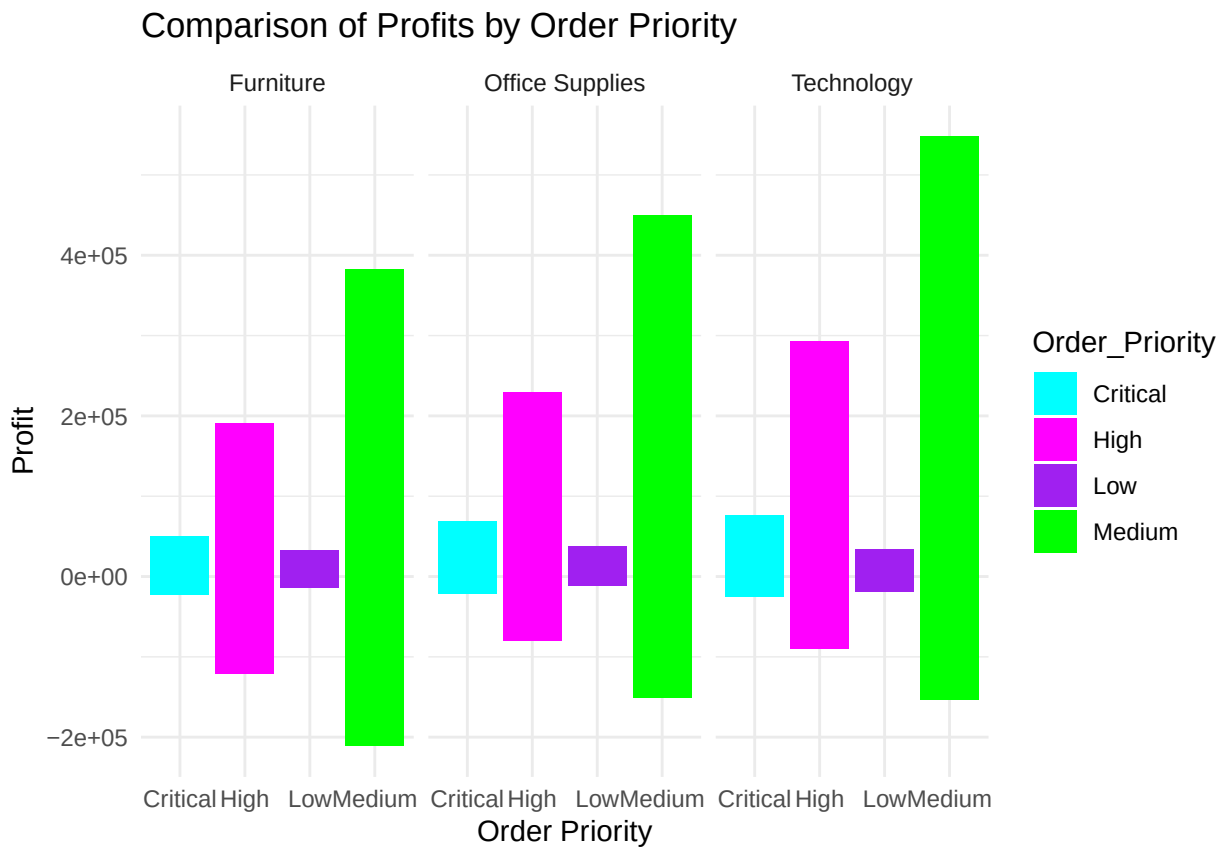
```
# 6. EXPLORATORY DATA ANALYSIS (EDA)

#EXPLORATORY WITH THE ENTIRE DATASET ANALYSIS
#Chart 1
#This chart helps to identify which sub-categories are the most and least popular within
#each main category, providing insights into sales trends and inventory management.
#The chart "Count by Category and Sub_Category" shows the frequency of items sold across
#different sub-categories within Furniture, Office Supplies, and Technology. Office Supplies,
#particularly Binders, Storage, and Art, dominate in sales frequency. Furniture categories
#like Chairs and Furnishings, and Technology items like Phones and Accessories
#also show significant counts. Overall, Office Supplies lead in sales volume, highlighting
#key trends in inventory management.


# Group by category and subcategory and sum the quantities

frequency_data <- sales_data %>%
  group_by(Category, Sub_Category) %>%
  summarise(Frequency = sum(Quantity, na.rm = TRUE)) %>%
  ungroup()
```

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

```
# Create the chart
ggplot(frequency_data, aes(x = Frequency, y = reorder(Sub_Category, Frequency), fill = Category)) +
  geom_bar(stat = "identity") +
  labs(title = "Count by Category and Sub_Category", x = "Frequency", y = "Sub_Category") +
  theme_minimal() +
  scale_fill_manual(values = c("Furniture" = "purple", "Office Supplies" = "blue",
  "Technology" = "green"))
```




```r
#Chart 2
# The chart "Comparison of Profits by Order Priority" illustrates profits across different
# categories (Furniture, Office Supplies, Technology) and order priorities (Critical,
# High, Low, Medium). Technology shows the highest profits, particularly with Medium priority
# orders. Office Supplies and Furniture also perform well with High and Medium priorities.
# Critical and Low priorities generally yield lower profits or losses across all categories.
# The chart highlights that Medium priority orders are most profitable, especially in the
# Technology category.
# Create the chart
ggplot(sales_data, aes(x = Order_Priority, y = Profit, fill = Order_Priority)) +
  geom_bar(stat = "identity") +
  facet_wrap(~Category, scales = "free_x") +
  scale_fill_manual(values = c("cyan", "magenta", "purple", "green")) +
  labs(
    title = "Comparison of Profits by Order Priority",
    x = "Order Priority",
    y = "Profit"
  ) +
  theme_minimal()
```



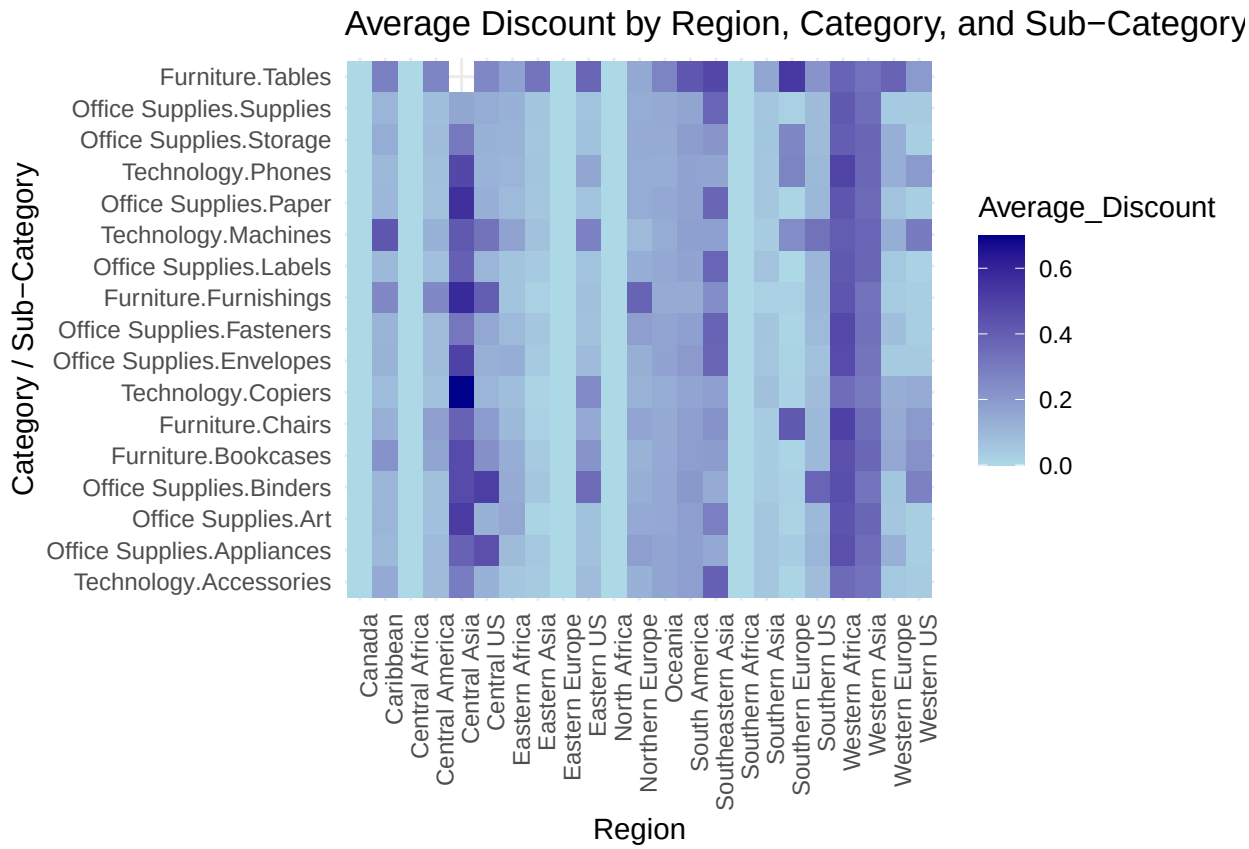

```
#Chart 3
install.packages("reshape2")
```

```
## Installing package into 'C:/Users/nadda/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

## package 'reshape2' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'reshape2'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\nadda\AppData\Local\R\win-library\4.2\00LOCK\reshape2\libs\x64\reshape2.dll
## to
## C:\Users\nadda\AppData\Local\R\win-library\4.2\reshape2\libs\x64\reshape2.dll:
## Permission denied

## Warning: restored 'reshape2'

##
## The downloaded binary packages are in
##  C:\Users\nadda\AppData\Local\Temp\RtmpewMKaI\downloaded_packages
```

```r
library(reshape2)

#Average Discount HEATMAP
#The chart "Average Discount by Region, Category, and Sub-Category" displays a heatmap
#of average discounts across various regions and product categories. Darker shades indicate
#higher average discounts. Notable patterns include lower discounts in the regions such as
#Canada, Central Afric Eastern Asia Eastern US Southern Africa show lower average discounts
#across most categories while in regions like Weatwrn Asia Weastwer Africa and Central Asia
#present the higher discount accross the Category and Sub Category. The chart highlights
#regional variations in discount strategies for different product sub-categories, suggesting
#that discount practices are tailored to specific regional markets.

avg_discount_data <- sales_data %>%
  group_by(Region, Category, Sub_Category) %>%
  summarise(Average_Discount = mean(Discount, na.rm = TRUE)) %>%
  ungroup()
```

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

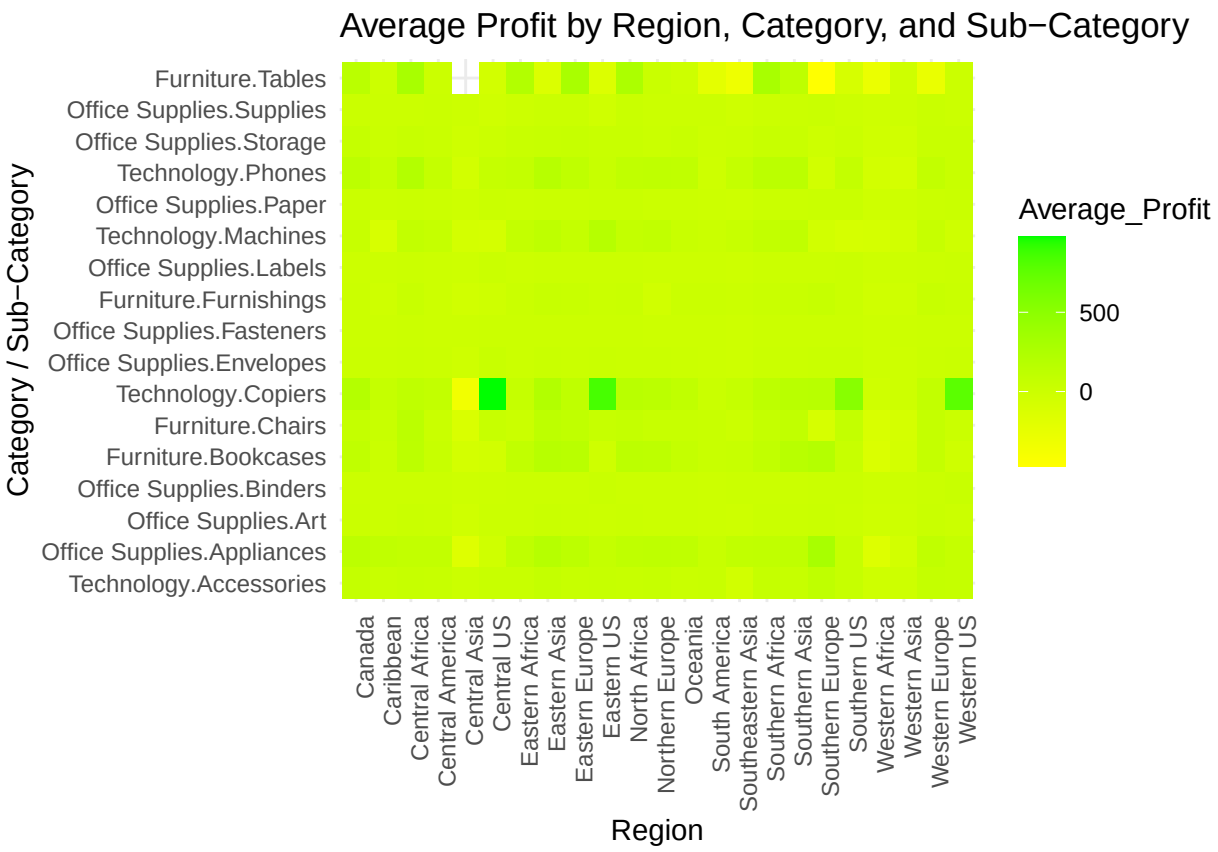
```r
ggplot(avg_discount_data, aes(x = Region, y = interaction(Category, Sub_Category),
  fill = Average_Discount)) + geom_tile() +
  labs(title = "Average Discount by Region, Category, and Sub-Category", x = "Region",
  y = "Category / Sub-Category") +
  scale_fill_gradient(low = "lightblue", high = "darkblue", na.value = "cyan") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```
#CHART 4
#The chart "Average Profit by Region, Category, and Sub-Category" presents a heatmap showing
#the average profit across various regions and product sub-categories. Darker shades of
#blue indicate higher average profits. Key insights include higher profits for certain
#sub-categories like Technology-Copiers in Central US, Eastern US, Southern US, and Western US.
#Most regions exhibit moderate to low average profits across sub-categories, with noticeable
#variations suggesting that profitability is influenced by both regional market conditions and
#product types. This visualization highlights areas where certain products are more profitable
#in specific regions.

#Average Profit HEATMAP
avg_profit_data <- sales_data %>%
  group_by(Region, Category, Sub_Category) %>%
  summarise(Average_Profit = mean(Profit, na.rm = TRUE)) %>%
  ungroup()
```

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

```
ggplot(avg_profit_data, aes(x = Region, y = interaction(Category, Sub_Category), fill = Average_Profit))
  geom_tile() +
  labs(title = "Average Profit by Region, Category, and Sub-Category", x = "Region",
  y = "Category / Sub-Category") +
  scale_fill_gradient(low = "#FFFF00", high = "#00FF00", na.value = "beige") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

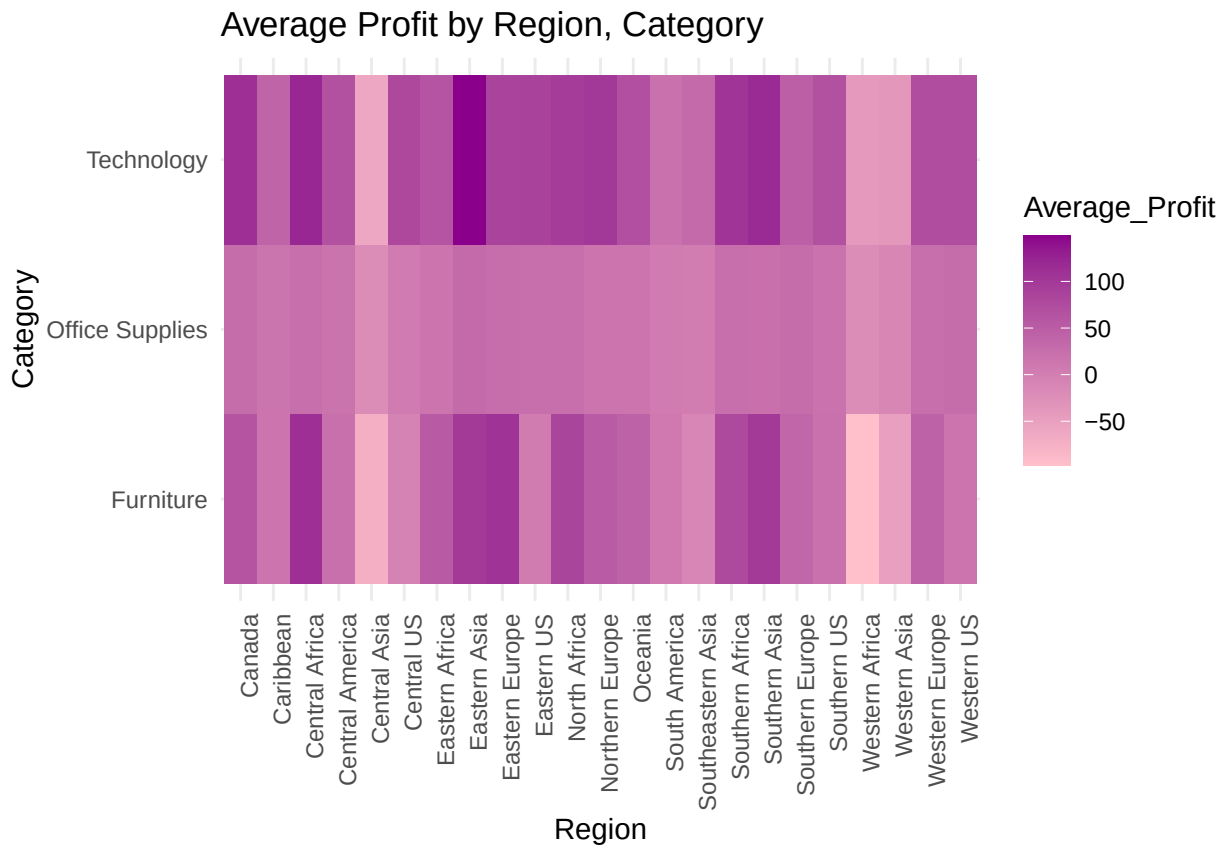



```
#CHART 5
#The chart "Average Profit by Region, Category" displays a heatmap illustrating the
#average profit for Technology, Office Supplies, and Furniture across various regions.
#Darker shades of blue represent higher average profits, while lighter shades and cyan
#indicate lower profits or losses. Key insights include higher profits in the Technology
#category for regions like Central Africa, Eastern Asia, Eastern Europe, Southern Asia
#and Southen Africa and lower profits or losses for Furniture in regions such as Western
#Africa and Central Asia. Office Supplies show moderate average profits across most regions.
#This visualization highlights how different product categories perform profit-wise in
#various regions, providing insights into regional profitability trends.

#Average Profit HEATMAP
avg_profit_data <- sales_data %>%
  group_by(Region, Category) %>%
  summarise(Average_Profit = mean(Profit, na.rm = TRUE)) %>%
  ungroup()
```

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

```
ggplot(avg_profit_data, aes(x = Region, y = interaction(Category), fill = Average_Profit)) +
  geom_tile() +
  labs(title = "Average Profit by Region, Category", x = "Region", y = "Category") +
```

```
  scale_fill_gradient(low = "#FFC0CB", high = "#8B008B", na.value = "beige") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```
#Chart 6 SCOPE (BY SUB_CATEGORY)
#The chart "Sales Comparison by Year and Sub-Category (2012 vs 2015)" shows the sales growth
#for various sub-categories from 2012 to 2015. Phones and Copiers exhibit the most significant
#sales increases, followed by Bookcases, Chairs, and Storage. Appliances, Machines, and
#Accessories show moderate growth, while Tables, Binders, and Furnishings have steady but less
#pronounced growth. Art, Supplies, Paper, Envelopes, Fasteners, and Labels remain relatively
#stable with minimal growth. This highlights the sub-categories with the highest and lowest sales
#growth over the period.

install.packages("lubridate")
```

```
## Installing package into 'C:/Users/nadda/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

## package 'lubridate' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'lubridate'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\nadda\AppData\Local\R\win-library\4.2\00LOCK\lubridate\libs\x64\lubridate.dll
```

```
## to
## C:\Users\nadda\AppData\Local\R\win-library\4.2\lubridate\libs\x64\lubridate.dll:
## Permission denied

## Warning: restored 'lubridate'

##
## The downloaded binary packages are in
##  C:\Users\nadda\AppData\Local\Temp\RtmpewMKaI\downloaded_packages
```

```
install.packages("ggrepel")
```

```
## Installing package into 'C:/Users/nadda/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

## package 'ggrepel' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'ggrepel'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\nadda\AppData\Local\R\win-library\4.2\00LOCK\ggrepel\libs\x64\ggrepel.dll
## to C:\Users\nadda\AppData\Local\R\win-library\4.2\ggrepel\libs\x64\ggrepel.dll:
## Permission denied

## Warning: restored 'ggrepel'

##
## The downloaded binary packages are in
##  C:\Users\nadda\AppData\Local\Temp\RtmpewMKaI\downloaded_packages
```

```
library(ggrepel)
library(lubridate)
```

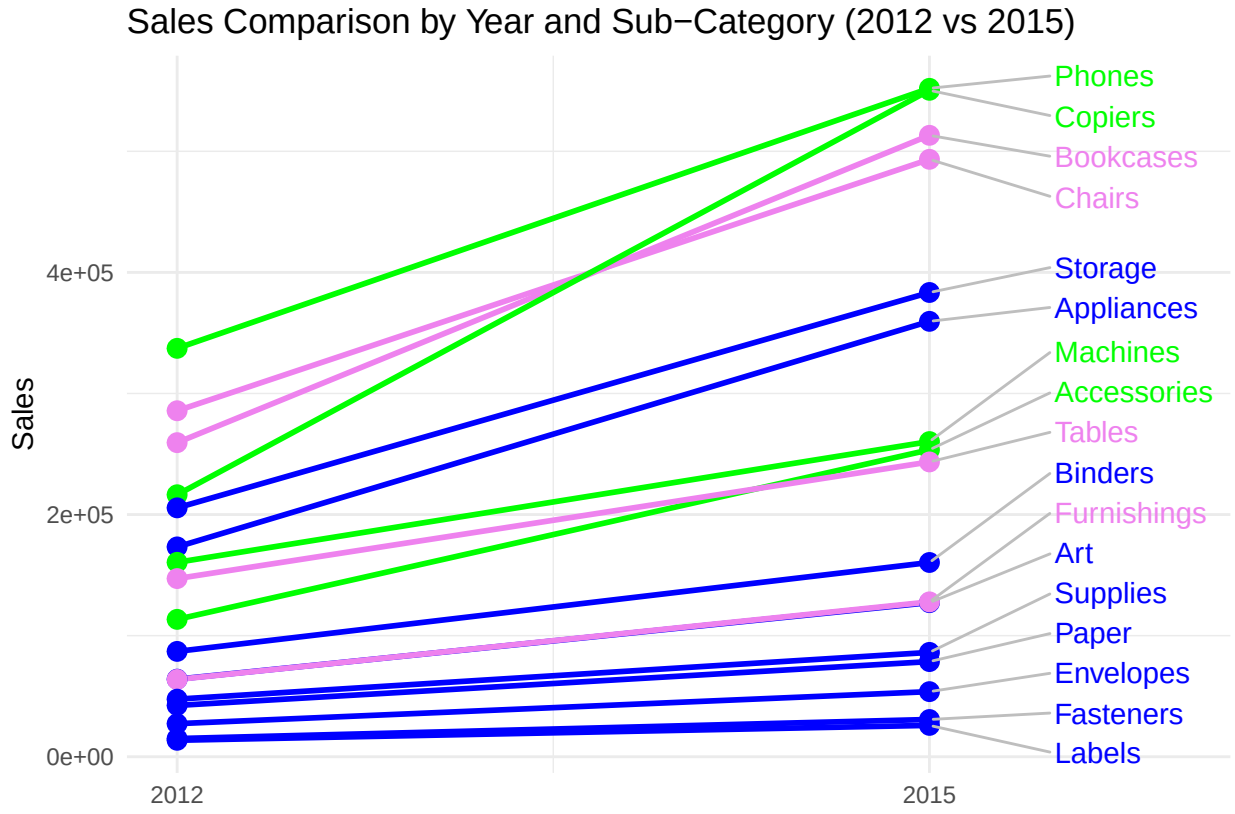
```
##
## Attaching package: 'lubridate'

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

```
sales_data$Year <- year(sales_data$Order_Date)

sales_by_year <- sales_data %>%
  group_by(Year, Sub_Category, Category) %>%
  summarise(Sales = sum(Sales, na.rm = TRUE)) %>%
  ungroup()
```

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

```r
sales_by_year_filtered <- sales_by_year %>%
  filter(Year %in% c(2012, 2015))

# Define a color palette with three colors for the categories
category_colors <- c("Furniture" = "violet", "Office Supplies" = "blue", "Technology" = "green")

ggplot(sales_by_year_filtered, aes(x = Year, y = Sales, group = Sub_Category, color = Category)) +
  geom_line(size = 1) +
  geom_point(size = 3) +
  geom_text_repel(data = subset(sales_by_year_filtered, Year == 2015),
                  aes(label = Sub_Category),
                  nudge_x = 0.5,
                  direction = "y",
                  hjust = 0,
                  segment.color = 'grey') +
  scale_color_manual(values = category_colors) +
  scale_x_continuous(breaks = c(2012, 2015), limits = c(2012, 2016)) +
  labs(title = "Sales Comparison by Year and Sub-Category (2012 vs 2015)", x = "", y = "Sales") +
  theme_minimal() +
  theme(legend.position = "none") +
  coord_cartesian(clip = 'off')
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

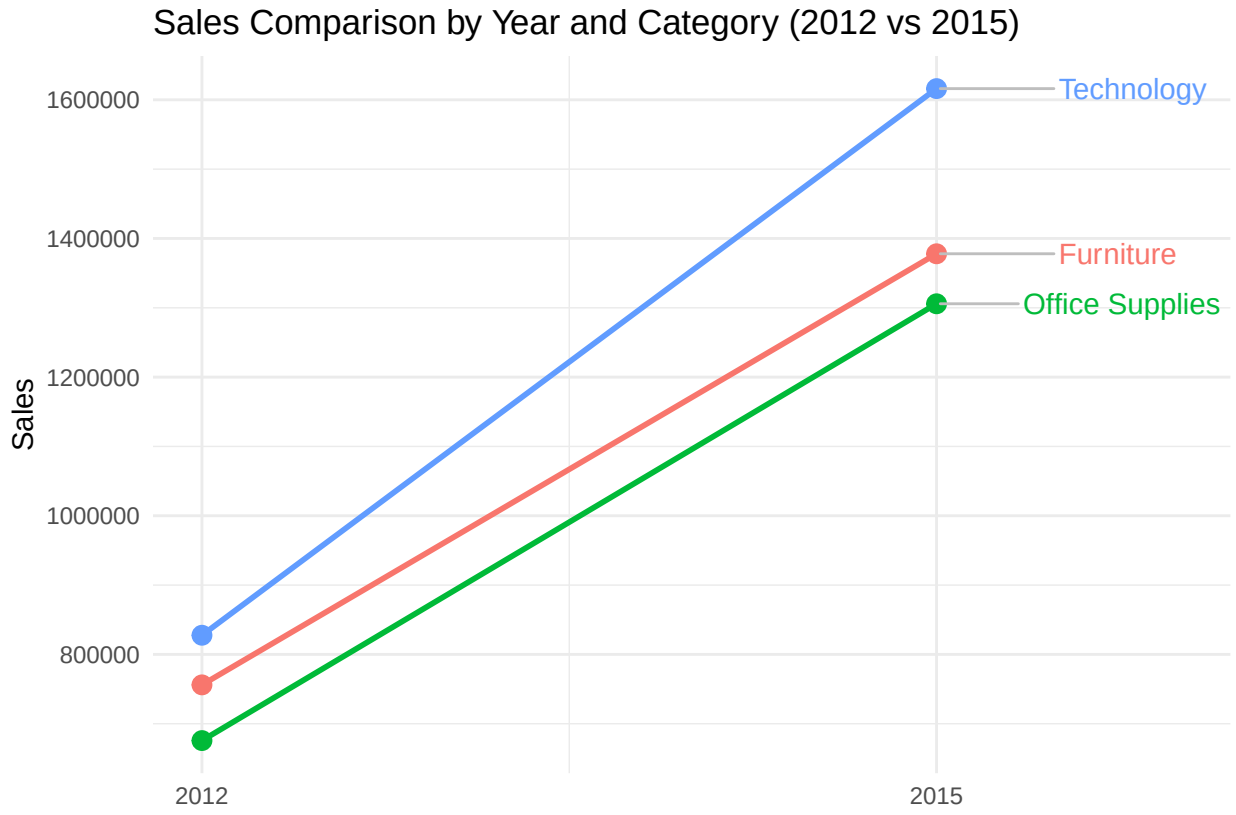



```
#Chart 7 SCOPE (BY CATEGORIA)
#The chart "Sales Comparison by Year and Category (2012 vs 2015)" shows the sales trends
#for Technology, Furniture, and Office Supplies from 2012 to 2015. Technology, represented
#in blue, shows the highest increase in sales, followed by Furniture in red and Office
#Supplies in green. Each category demonstrates significant growth over the period, with
#Technology leading in total sales by 2015

# Extract the year from Order_Date
sales_data$Year <- year(sales_data$Order_Date)

sales_by_year <- sales_data %>%
  group_by(Year, Category) %>%
  summarise(Sales = sum(Sales, na.rm = TRUE)) %>%
  ungroup()
```

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

```
sales_by_year_filtered <- sales_by_year %>%
  filter(Year %in% c(2012, 2015))

ggplot(sales_by_year_filtered, aes(x = Year, y = Sales, group = Category, color = Category)) +
  geom_line(size = 1) +
  geom_point(size = 3) +
  geom_text_repel(data = subset(sales_by_year_filtered, Year == 2015),
```

```
                  aes(label = Category),
                  nudge_x = 0.5,
                  direction = "y",
                  hjust = 0,
                  segment.color = 'grey') +
  scale_x_continuous(breaks = c(2012, 2015), limits = c(2012, 2016)) +
  labs(title = "Sales Comparison by Year and Category (2012 vs 2015)", x = "", y = "Sales") +
  theme_minimal() +
  theme(legend.position = "none") +
  coord_cartesian(clip = 'off')
```

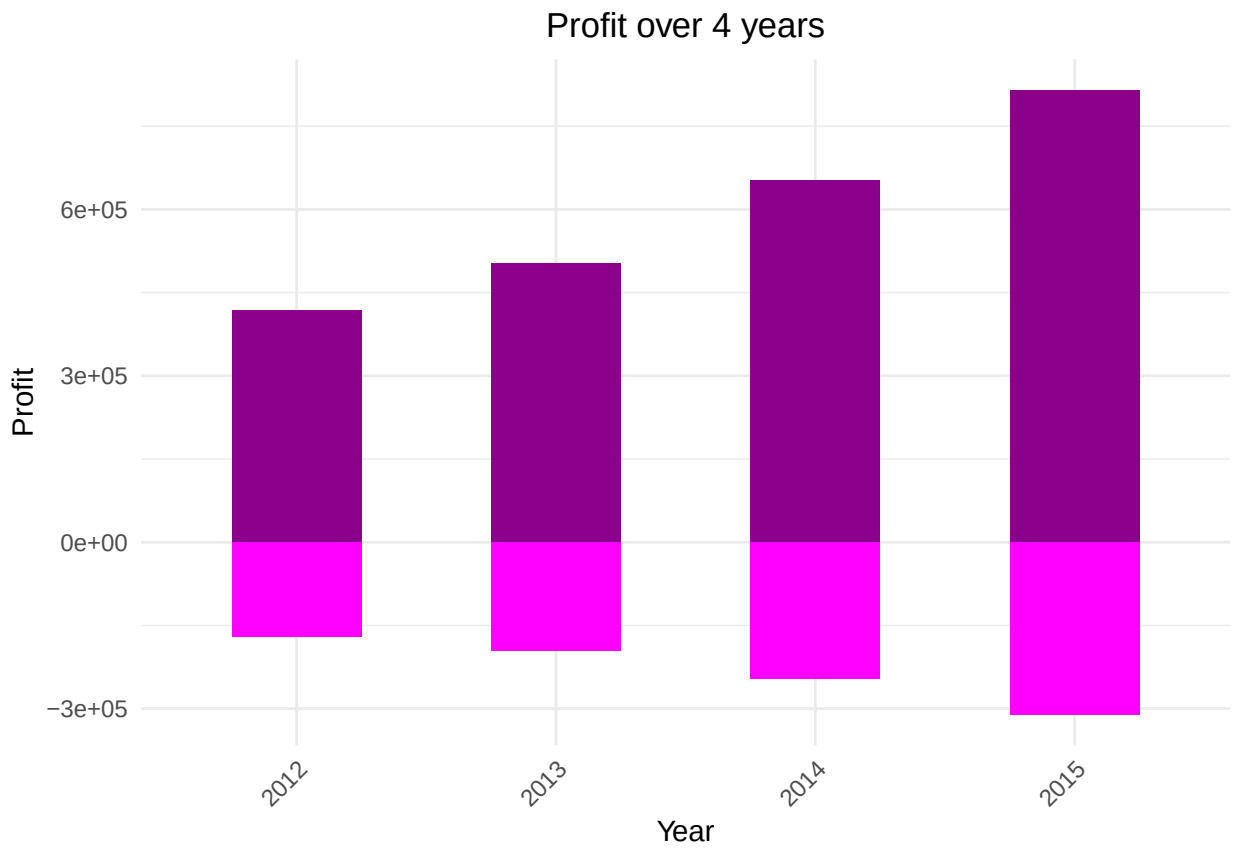



```
#Chart 8 Profit by year (With individual trasactions)
#Shows the Profit for each transaction within each year. Each bar represents the sum
#of transactions for that year, with colors indicating gains or losses.

profit_plot <- ggplot(sales_data, aes(x = factor(Year), y = Profit, fill = Profit < 0)) +
  geom_bar(stat = "identity", width = 0.5) +
  scale_fill_manual(values = c("TRUE" = "magenta", "FALSE" = "#8B008B")) +
  labs(title = "Profit over 4 years", y = "Profit", x = "Year") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme(plot.title = element_text(hjust = 0.5)) +
  guides(fill = FALSE)  # Ocultar la leyenda
```

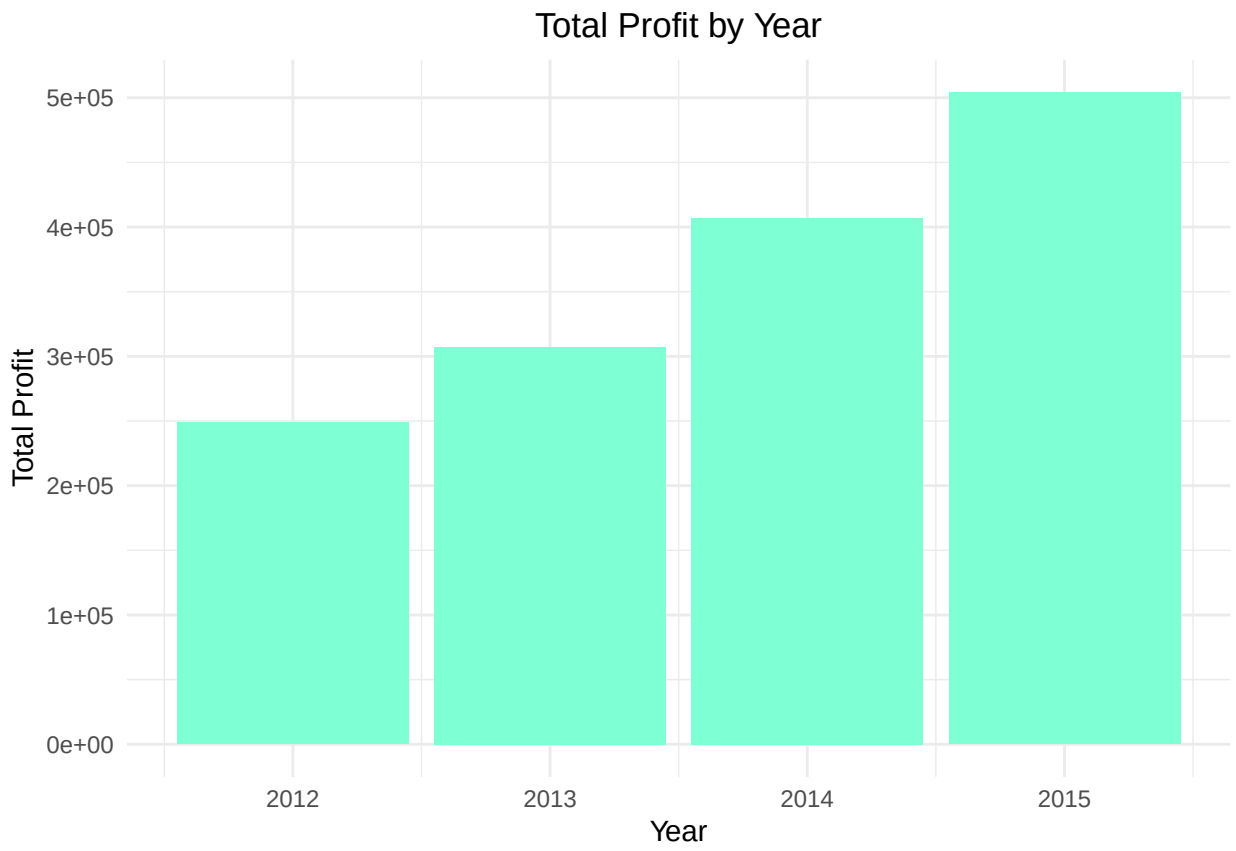
```
## Warning: The `<scale>` argument of `guides()` cannot be `FALSE`. Use "none" instead as
```

```
## of ggplot2 3.3.4.
## This warning is displayed once every 8 hours.
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was
## generated.
```

```
print(profit_plot)
```




```
#Chart 9 profit by year (without individual transactions)
#Shows the aggregated Total Profit for each year. Each bar represents the total sum of
#all profits for the year, without showing individual transactions.

sales_data <- sales_data %>%
  mutate(Year = year(Order_Date))

yearly_profit <- sales_data %>%
  group_by(Year) %>%
  summarise(Total_Profit = sum(Profit, na.rm = TRUE))

profit_plot <- ggplot(yearly_profit, aes(x = Year, y = Total_Profit)) +
  geom_bar(stat = "identity", fill = "#7FFFD4") +
  labs(title = "Total Profit by Year", x = "Year", y = "Total Profit") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))

print(profit_plot)
```

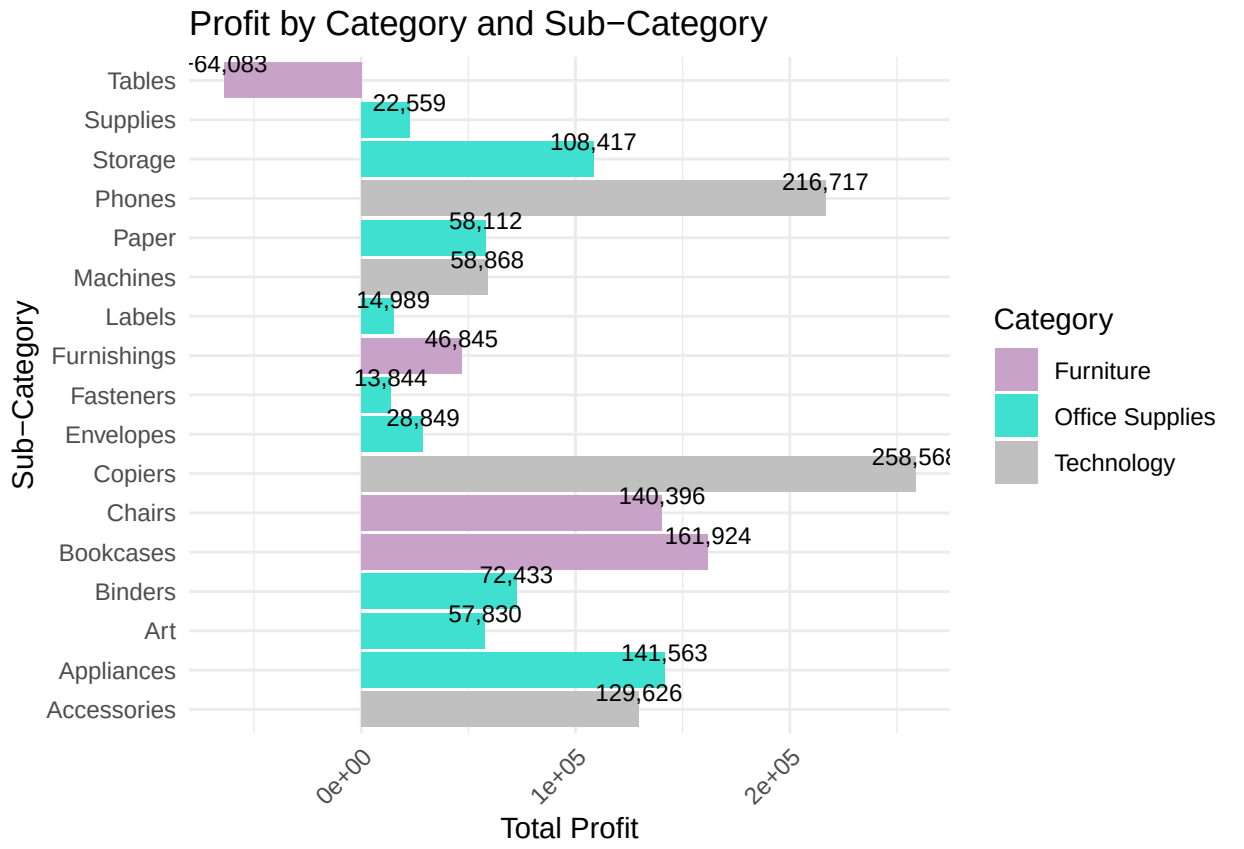



```
#Chart 10 Profit by Category and Subcategory

#The chart "Profit by Category and Sub-Category" shows that **Copiers** in the Technology
#category have the highest profit at 258,568, followed by **Phones** at 216,717. In contrast,
#Tables in the Furniture category show the most significant loss with a negative profit
#of -64,083. Technology sub-categories dominate the highest profits, while Furniture has the
#lowest profit with notable losses in Tables.


profit_data <- sales_data %>%
  group_by(Category, Sub_Category) %>%
  summarise(Total_Profit = sum(Profit, na.rm = TRUE)) %>%
  ungroup()
```

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

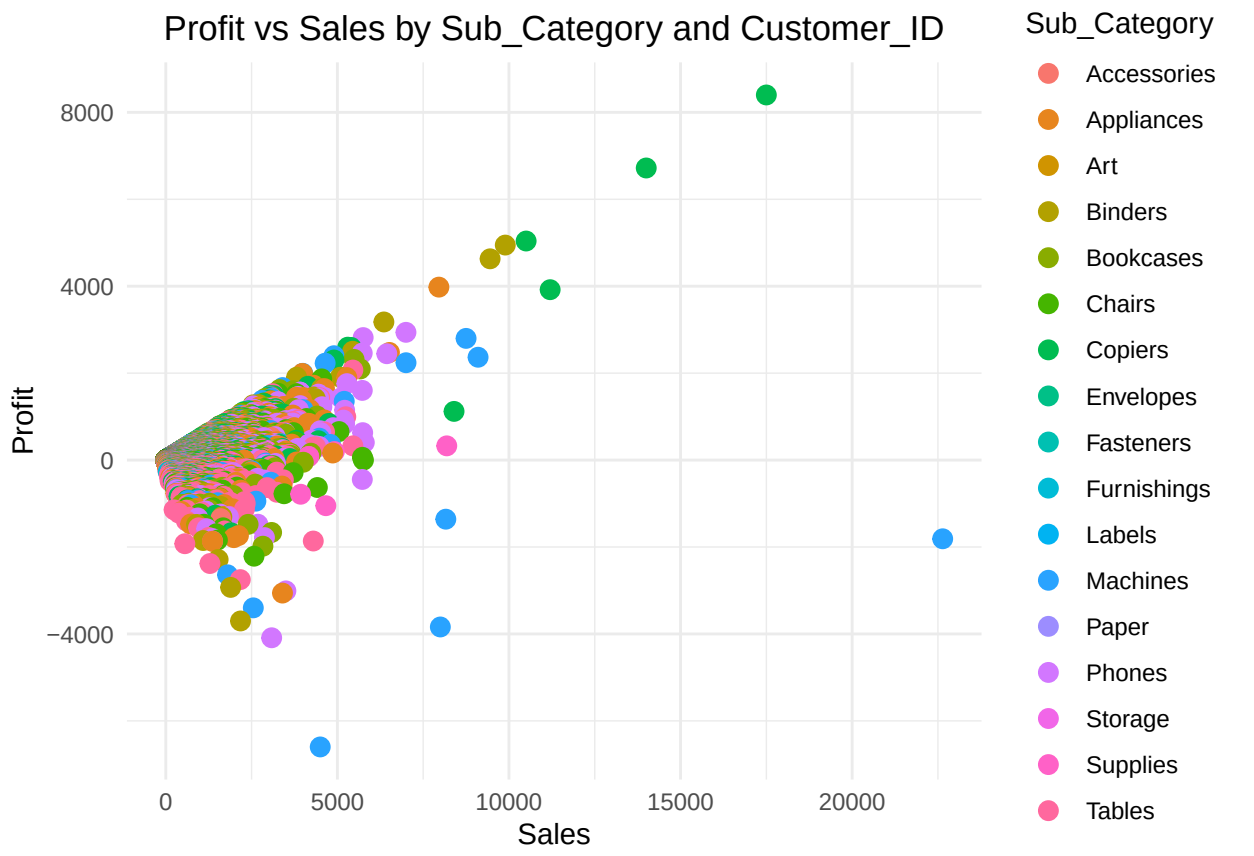
```
profit_plot <- ggplot(profit_data, aes(x = Total_Profit, y = Sub_Category , fill = Category)) +
  geom_bar(stat = "identity", position = "dodge") +
  geom_text(aes(label = scales::comma(Total_Profit)), vjust = -0.5, size = 3) +
  labs(title = "Profit by Category and Sub-Category", y = "Sub-Category", x = "Total Profit") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_manual(values = c("Furniture" = "#C8A2C8", "Office Supplies" = "turquoise",
  "Technology" = "#C0C0C0"))
```

```r
# Print the plot
print(profit_plot)
```




```r
#Chart 11 SCATTERPLOT BY CUSTOMER_ID
#The chart "Profit vs Sales by Sub_Category and Customer_ID" is a scatter plot that illustrates
#the relationship between sales and profit for various sub-categories, with each point representing
#a unique Customer_ID. The sub-categories are color-coded as shown in the legend.

#Key Insights:
#1.Positive Correlation: There is a general positive correlation between sales and profit,
#meaning higher sales tend to be associated with higher profits.
#2.High Profit and Sales: Sub-categories such as Bookcases (green) and Appliances (orange)
#show high sales and correspondingly high profits.
#3.Low Profit and Sales: Some sub-categories, like Tables (pink) and Machines (blue),
#have lower sales and are closer to or below the zero-profit line, indicating they might
#not be performing well.
#4.Outliers: There are notable outliers, such as one point with high sales but very
#low profit, indicating a potential issue with that sub-category.
#5.Customer-Specific Data: The inclusion of Customer_ID adds a layer of granularity,showing
#the sales and profit performance at the individual customer level for each sub-category.

scatter_plot <- ggplot(sales_data, aes(x = Sales, y = Profit, color = Sub_Category)) +
  geom_point(size = 3) +
  labs(title = "Profit vs Sales by Sub_Category and Customer_ID",
       x = "Sales",
```

```
       y = "Profit",
       color = "Sub_Category") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))

print(scatter_plot)
```

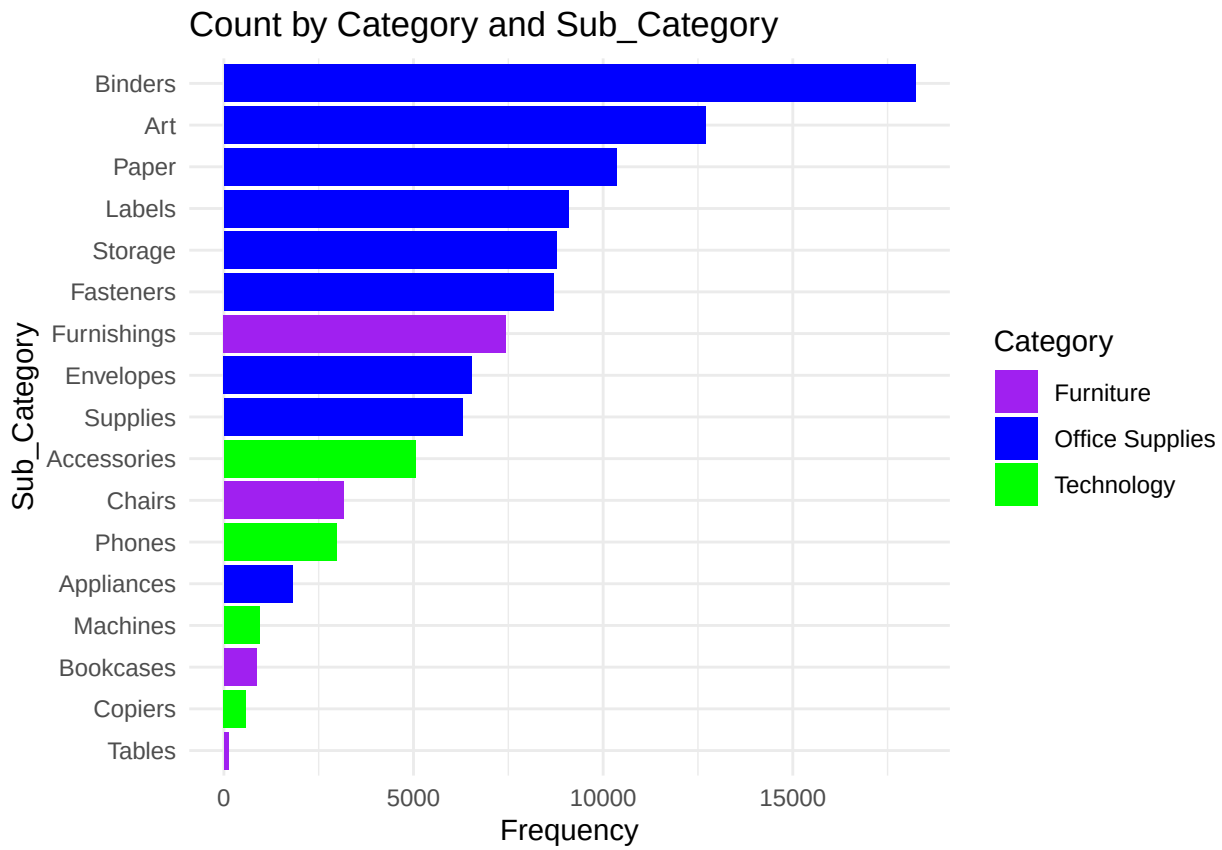



```
#EXPLORATORY DATASET ANALYSIS AFTER CLEANING DATA

#This chart identifies which sub-categories are the most and least popular within each
#main category, providing insights into sales trends and inventory management.
#Chart 1
frequency_data <- data_2 %>%
  group_by(Category, Sub_Category) %>%
  summarise(Frequency = sum(Quantity, na.rm = TRUE)) %>%
  ungroup()
```

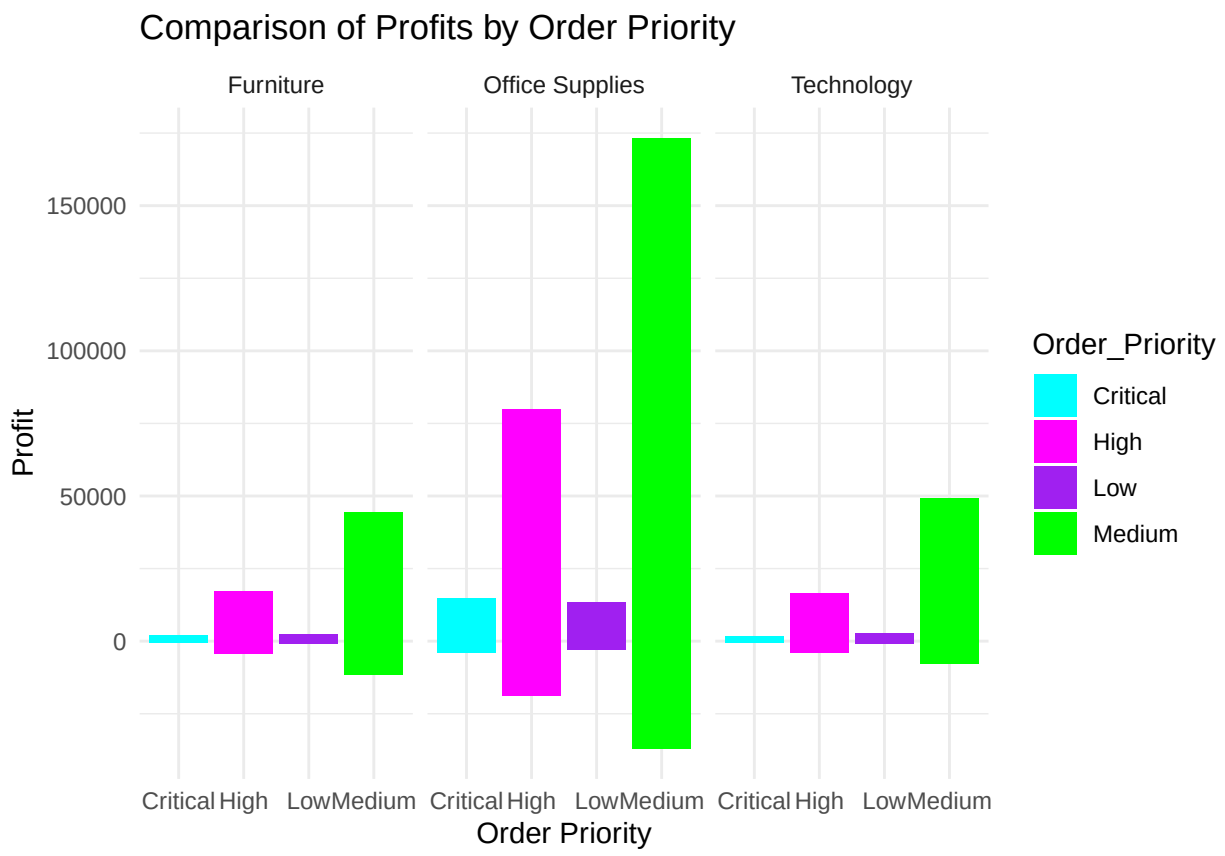
```
## `summarise()` has grouped output by 'Category'. You can override using the
## `.groups` argument.
```

```
# Crear el gráfico
ggplot(frequency_data, aes(x = Frequency, y = reorder(Sub_Category, Frequency), fill = Category)) +
  geom_bar(stat = "identity") +
  labs(title = "Count by Category and Sub_Category", x = "Frequency", y = "Sub_Category") +
```

```
  theme_minimal() +
  scale_fill_manual(values = c("Furniture" = "purple", "Office Supplies" = "blue", "Technology" = "gree
```



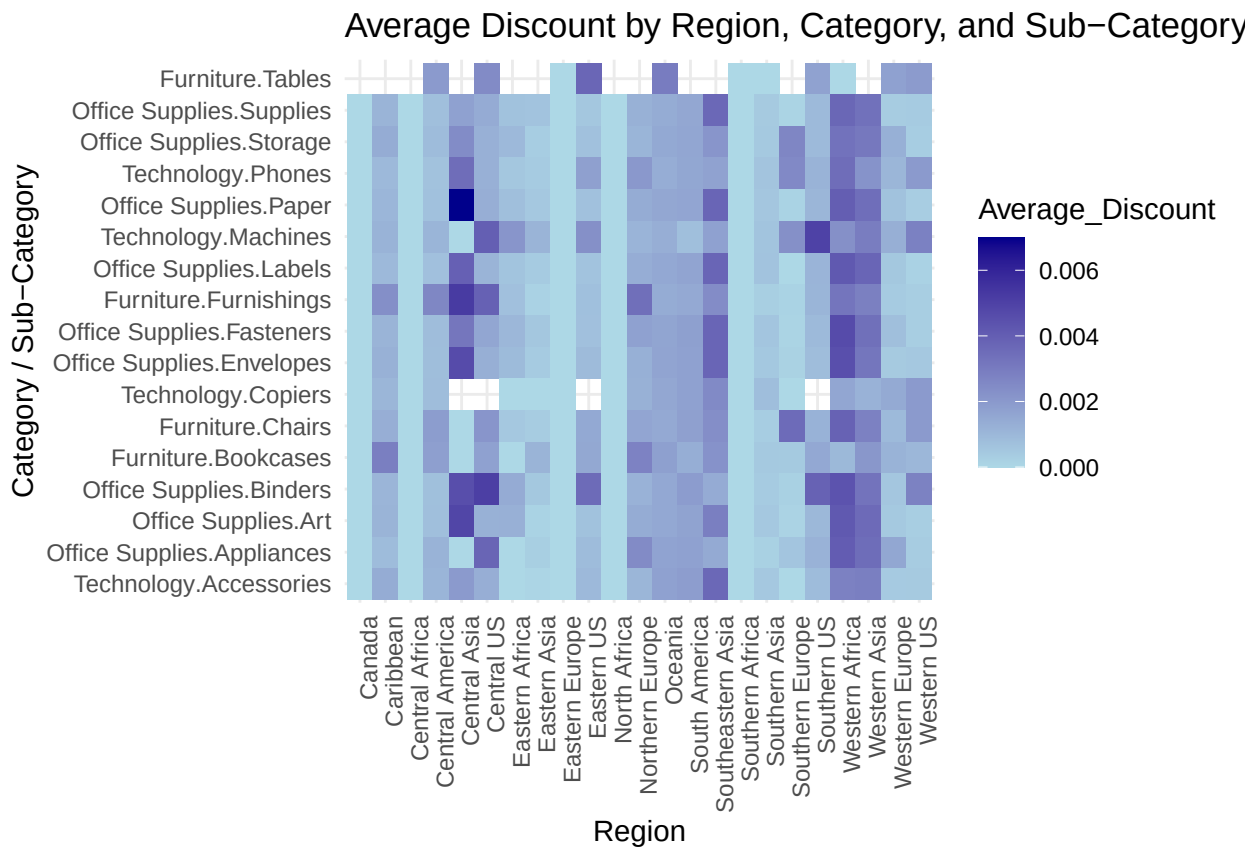

```
#Chart 2
# The chart "Comparison of Profits by Order Priority" illustrates profits across
# different categories (Furniture, Office Supplies, Technology) and order priorities
# (Critical, High, Low, Medium).

# Create the plot

ggplot(data_2, aes(x = Order_Priority, y = Profit, fill = Order_Priority)) +
  geom_bar(stat = "identity") +
  facet_wrap(~Category, scales = "free_x") +
  scale_fill_manual(values = c("cyan", "magenta", "purple", "green")) +
  labs(
    title = "Comparison of Profits by Order Priority",
    x = "Order Priority",
    y = "Profit"
  ) +
  theme_minimal()
```




```
#Chart 3
#The chart "Average Discount by Region, Category, and Sub-Category" displays a heatmap of
#average discounts across various regions and product categories. Darker shades indicate
#higher average discounts while lighter shades indicate lower profits or losses.

#Average Discount HEATMAP
avg_discount_data <- data_2 %>%
  group_by(Region, Category, Sub_Category) %>%
  summarise(Average_Discount = mean(Discount, na.rm = TRUE)) %>%
  ungroup()
```

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

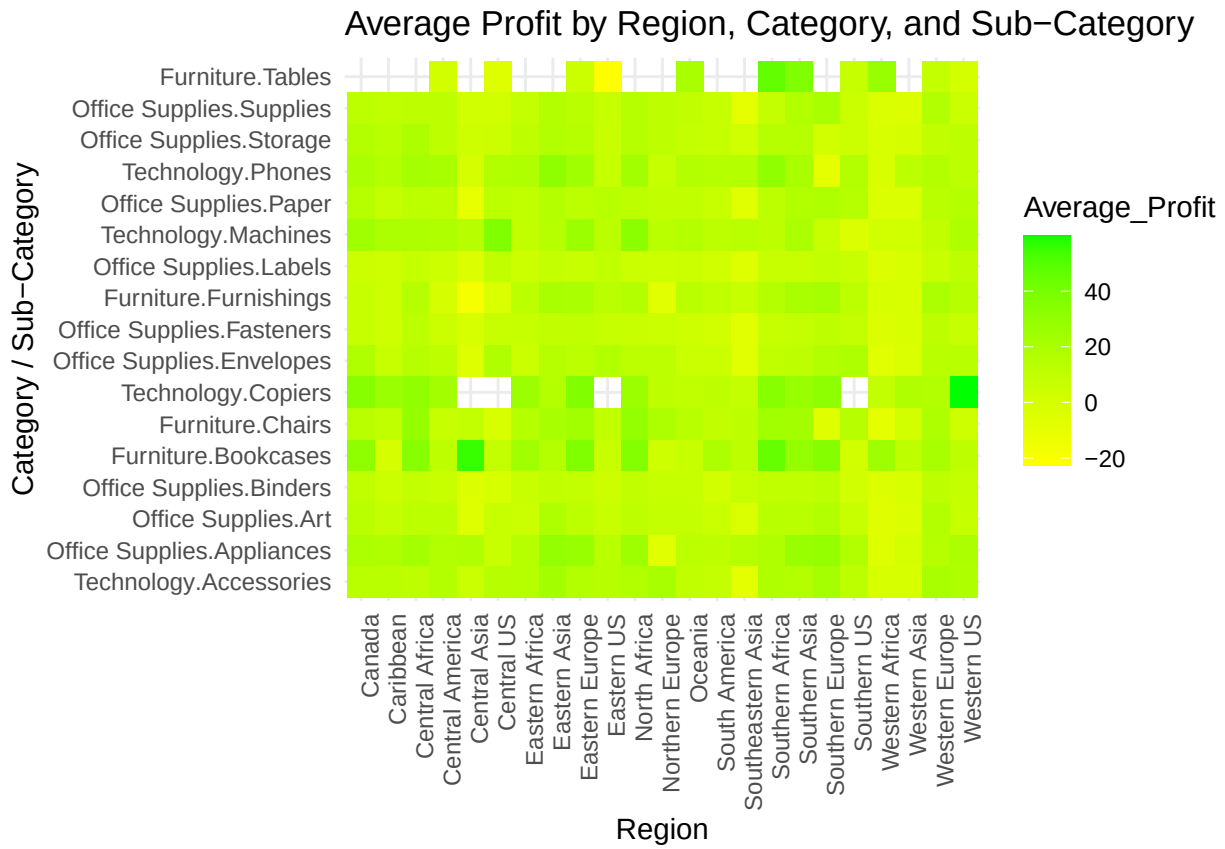
```
ggplot(avg_discount_data, aes(x = Region, y = interaction(Category, Sub_Category), fill = Average_Discou
  geom_tile() +
  labs(title = "Average Discount by Region, Category, and Sub-Category", x = "Region", y = "Category / S
  scale_fill_gradient(low = "lightblue", high = "darkblue", na.value = "cyan") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```
#CHART 4
#The chart "Average Profit by Region, Category" displays a heatmap illustrating the
#average profit for Technology, Office Supplies, and Furniture across various regions.
#Darker shades of blue represent higher average profits, while lighter shades and cyan
#indicate lower profits or losses. Key insights include higher profits in the Technology category for r
#Average Profit HEATMAP
avg_profit_data <- data_2 %>%
  group_by(Region, Category, Sub_Category) %>%
  summarise(Average_Profit = mean(Profit, na.rm = TRUE)) %>%
  ungroup()
```

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

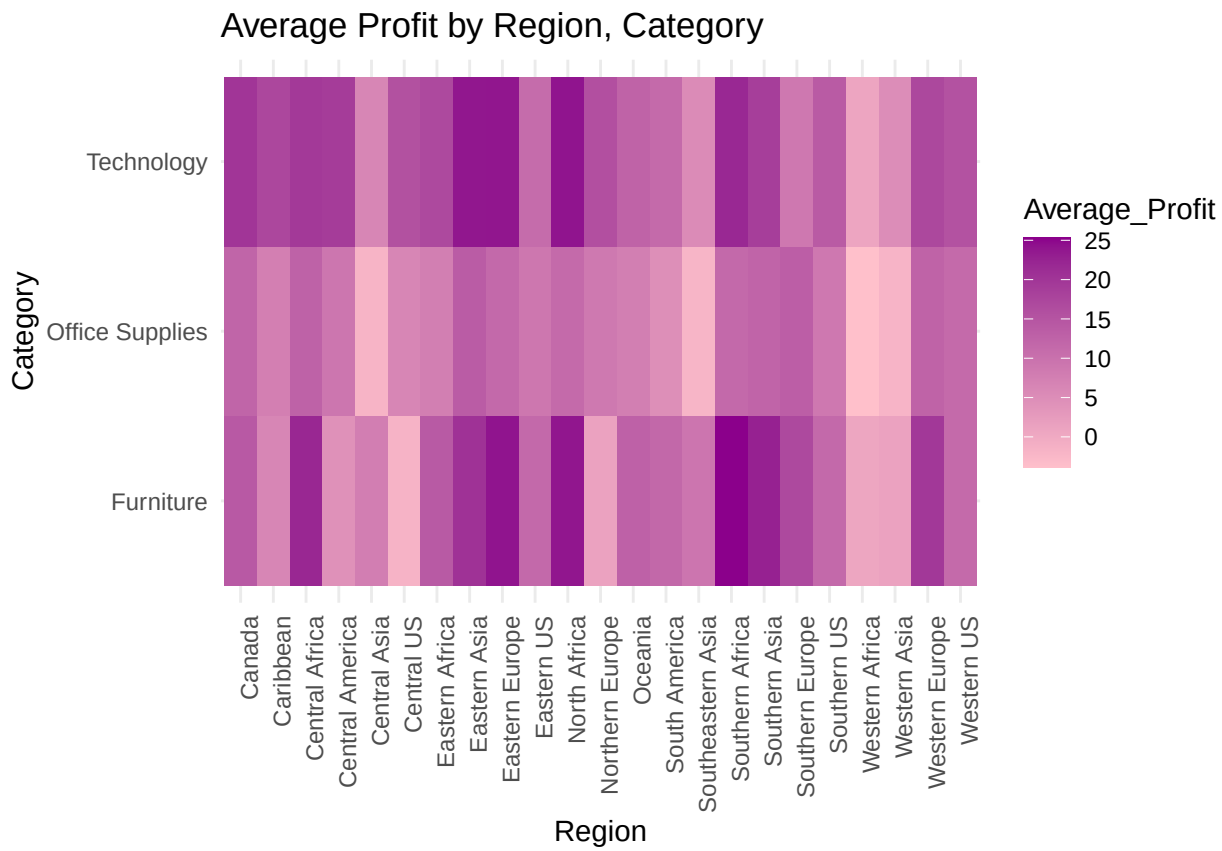
```
ggplot(avg_profit_data, aes(x = Region, y = interaction(Category, Sub_Category), fill = Average_Profit)
  geom_tile() + labs(title = "Average Profit by Region, Category, and Sub-Category", x = "Region",
  y = "Category / Sub-Category") +
  scale_fill_gradient(low = "#FFFF00", high = "#00FF00", na.value = "beige") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```
#CHART 5
#The chart "Average Profit by Region, Category" displays a heatmap illustrating the average
#profit for Technology, Office Supplies, and Furniture across various regions. Darker shades
#of blue represent higher average profits, while lighter shades and cyan indicate lower profits
#or losses.

avg_profit_data <- data_2 %>%
  group_by(Region, Category) %>%
  summarise(Average_Profit = mean(Profit, na.rm = TRUE)) %>%
  ungroup()
```

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

```
ggplot(avg_profit_data, aes(x = Region, y = interaction(Category), fill = Average_Profit)) +
  geom_tile() +
  labs(title = "Average Profit by Region, Category", x = "Region", y = "Category") +
  scale_fill_gradient(low = "#FFC0CB", high = "#8B008B", na.value = "beige") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```

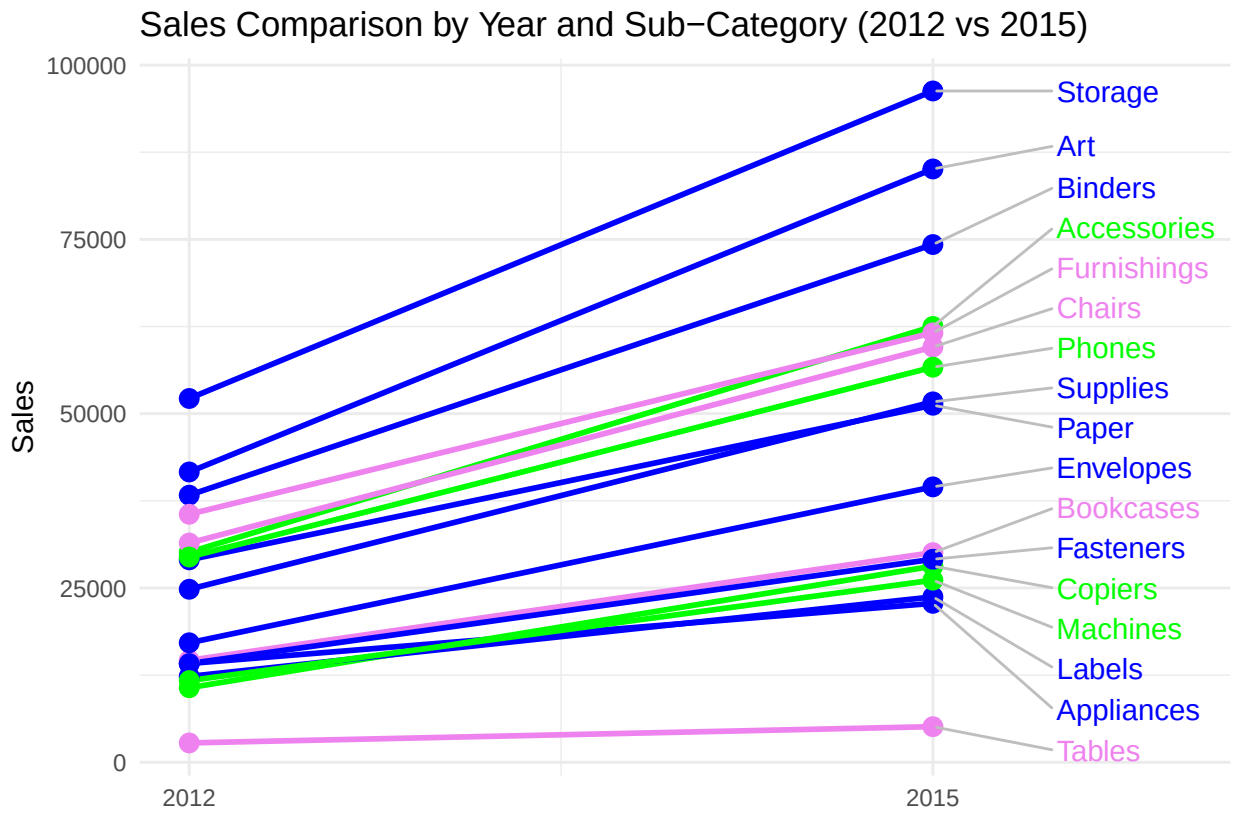



```
#Chart 6 SCOPE (BY SUB_CATEGORY)
#The chart "Sales Comparison by Year and Sub-Category (2012 vs 2015)" shows the sales
#growth for various sub-categories from 2012 to 2015.

data_2$Year <- year(data_2$Order_Date)

sales_by_year <- data_2 %>%
  group_by(Year, Sub_Category, Category) %>%
  summarise(Sales = sum(Sales, na.rm = TRUE)) %>%
  ungroup()
```

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

```
sales_by_year_filtered <- sales_by_year %>%
  filter(Year %in% c(2012, 2015))

# Define a color palette with three colors for the categories
category_colors <- c("Furniture" = "violet", "Office Supplies" = "blue", "Technology" = "green")

ggplot(sales_by_year_filtered, aes(x = Year, y = Sales, group = Sub_Category, color = Category)) +
  geom_line(size = 1) +
  geom_point(size = 3) +
  geom_text_repel(data = subset(sales_by_year_filtered, Year == 2015),
                  aes(label = Sub_Category),
```

```
                  nudge_x = 0.5,
                  direction = "y",
                  hjust = 0,
                  segment.color = 'grey') +
  scale_color_manual(values = category_colors) +
  scale_x_continuous(breaks = c(2012, 2015), limits = c(2012, 2016)) +
  labs(title = "Sales Comparison by Year and Sub-Category (2012 vs 2015)", x = "", y = "Sales") +
  theme_minimal() +
  theme(legend.position = "none") +
  coord_cartesian(clip = 'off')
```

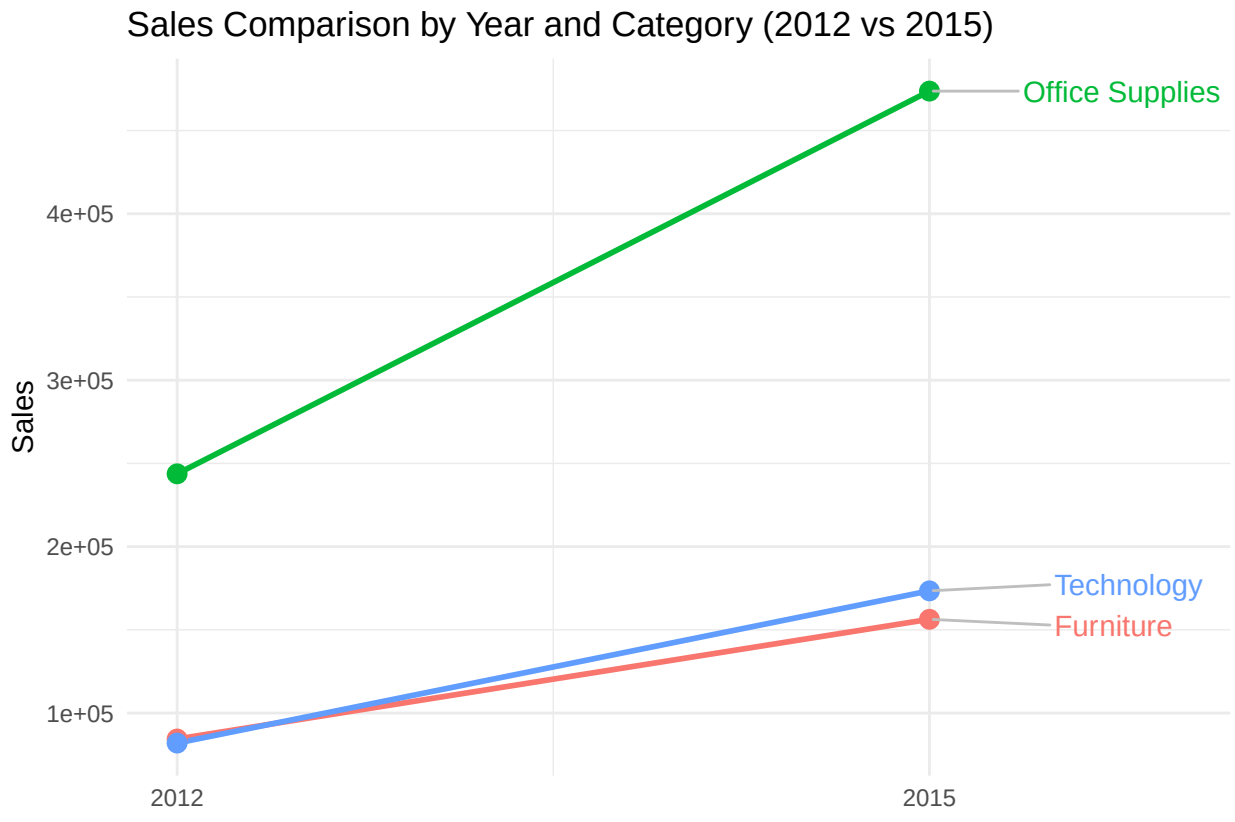



```
#Chart 7 SCOPE (BY CATEGORIA)
#The chart "Sales Comparison by Year and Category (2012 vs 2015)" shows the sales trends
#for Technology, Furniture, and Office Supplies from 2012 to 2015.


# Extract the year from Order_Date
data_2$Year <- year(data_2$Order_Date)

sales_by_year <- data_2 %>%
  group_by(Year, Category) %>%
  summarise(Sales = sum(Sales, na.rm = TRUE)) %>%
  ungroup()
```

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

```
## `.groups` argument.
```

```
sales_by_year_filtered <- sales_by_year %>%
  filter(Year %in% c(2012, 2015))
library(ggplot2)
library(ggrepel)

ggplot(sales_by_year_filtered, aes(x = Year, y = Sales, group = Category, color = Category)) +
  geom_line(size = 1) +
  geom_point(size = 3) +
  geom_text_repel(data = subset(sales_by_year_filtered, Year == 2015),
                  aes(label = Category),
                  nudge_x = 0.5,
                  direction = "y",
                  hjust = 0,
                  segment.color = 'grey') +
  scale_x_continuous(breaks = c(2012, 2015), limits = c(2012, 2016)) +
  labs(title = "Sales Comparison by Year and Category (2012 vs 2015)", x = "", y = "Sales") +
  theme_minimal() +
  theme(legend.position = "none") +
  coord_cartesian(clip = 'off')
```

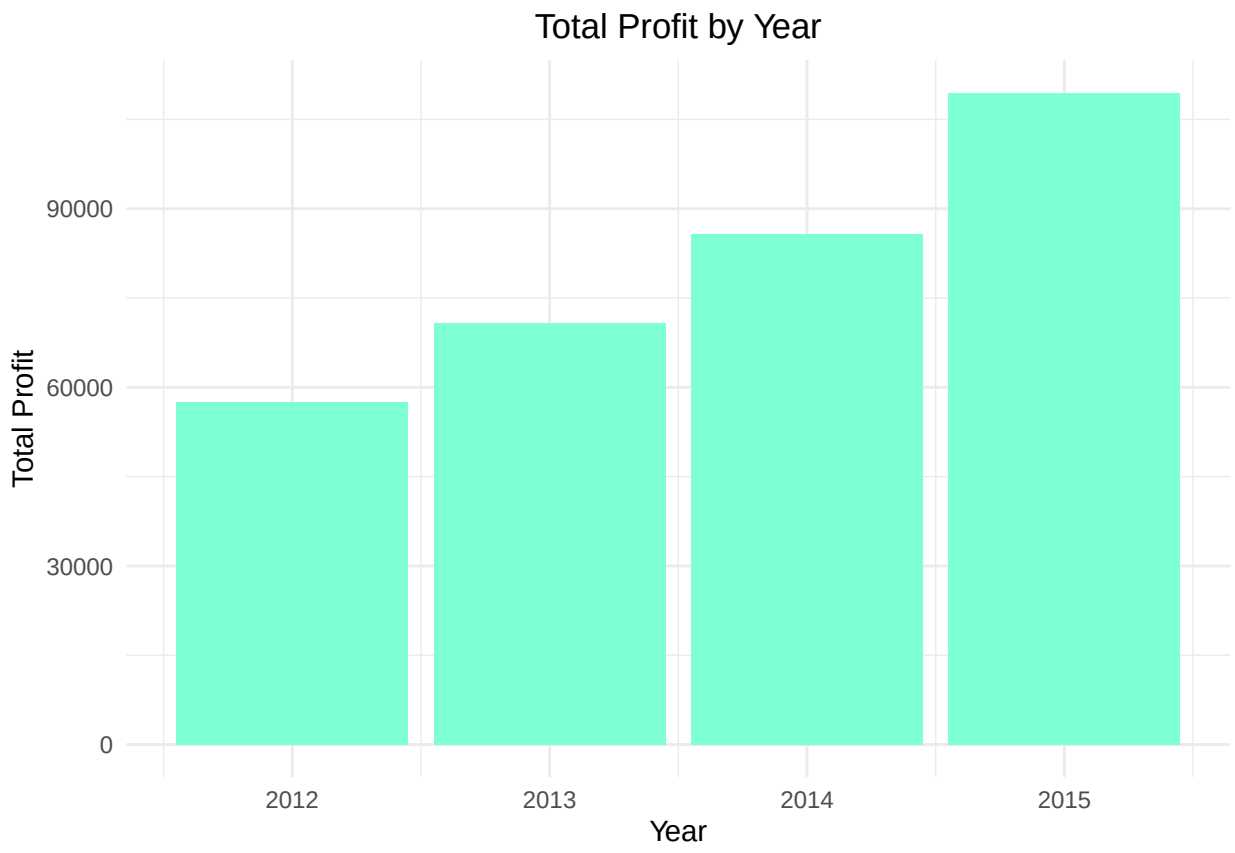



```
# Chart 8 Profit by year (With individual trasactions)
# Shows the Profit for each transaction within each year. Each bar represents the sum of
# transactions for that year, with colors indicating gains or losses.
```

```
profit_plot <- ggplot(data_2, aes(x = factor(Year), y = Profit, fill = Profit < 0)) +
  geom_bar(stat = "identity", width = 0.5) +
  scale_fill_manual(values = c("TRUE" = "magenta", "FALSE" = "#8B008B")) +
  labs(title = "Profit over 4 years", y = "Profit", x = "Year") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme(plot.title = element_text(hjust = 0.5)) +
  guides(fill = FALSE)  # Ocultar la leyenda

print(profit_plot)
```


```
#Chart 9 profit by year (without individual transactions)
#Shows the aggregated Total Profit for each year. Each bar represents the total sum of
#all profits for the year, without showing individual transactions.

data_2 <- data_2 %>%
  mutate(Year = year(Order_Date))

yearly_profit <- data_2 %>%
  group_by(Year) %>%
  summarise(Total_Profit = sum(Profit, na.rm = TRUE))

profit_plot <- ggplot(yearly_profit, aes(x = Year, y = Total_Profit)) +
  geom_bar(stat = "identity", fill = "#7FFFD4") +
  labs(title = "Total Profit by Year", x = "Year", y = "Total Profit") +
```

```
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))

print(profit_plot)
```

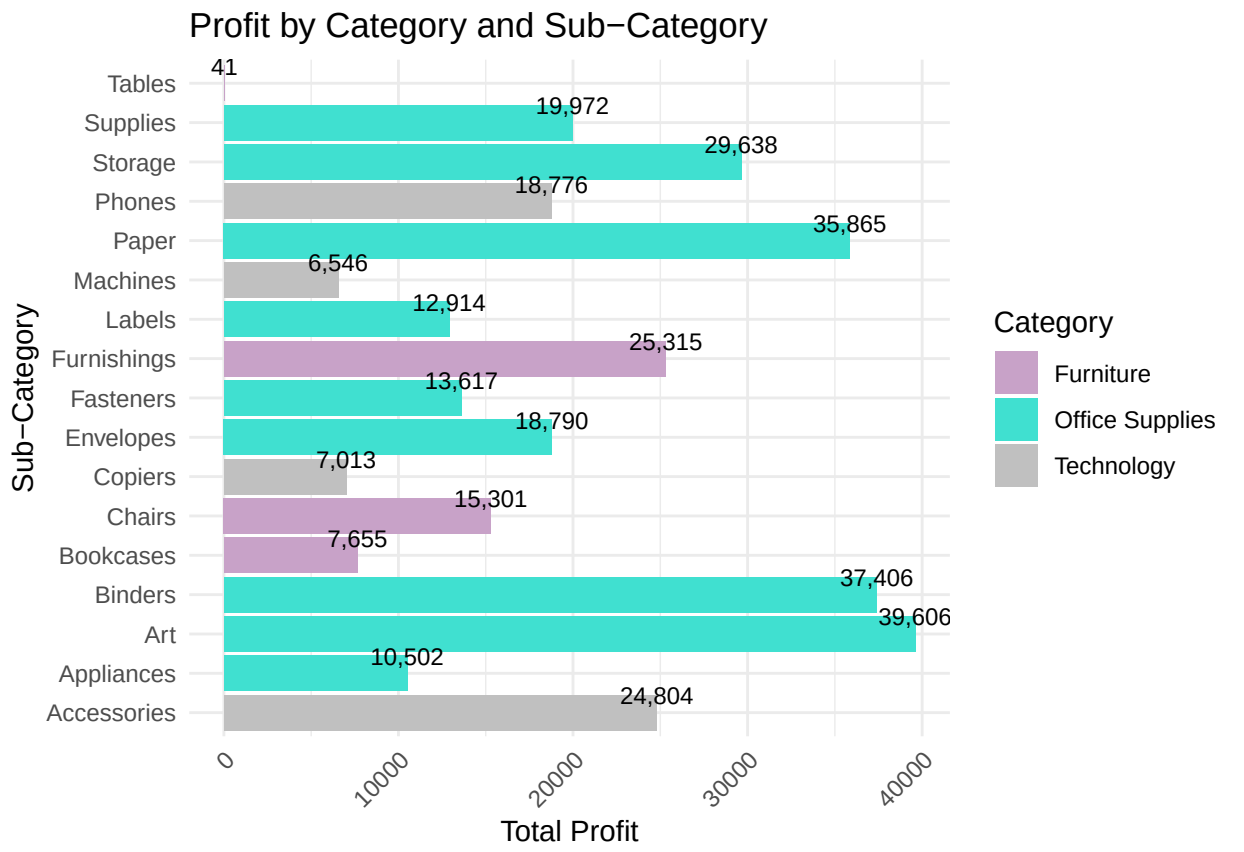



```
#Chart 10 Profit by Category and Subcategory
#The chart "Profit by Category and Sub-Category" shows the total profit for various
#sub-categories within the Furniture, Office Supplies, and Technology categories.
#Binders and Storage in Office Supplies have the highest profits, with 39,606 and 37,406
#respectively. Phones in Technology also perform well with a profit of 35,865. Conversely,
#Tables in Furniture have the lowest profit at 41. This highlights significant profitability
#differences across sub-categories, with Office Supplies generally showing higher profits
#compared to Furniture and Technology.

profit_data <- data_2 %>%
  group_by(Category, Sub_Category) %>%
  summarise(Total_Profit = sum(Profit, na.rm = TRUE)) %>%
  ungroup()
```

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

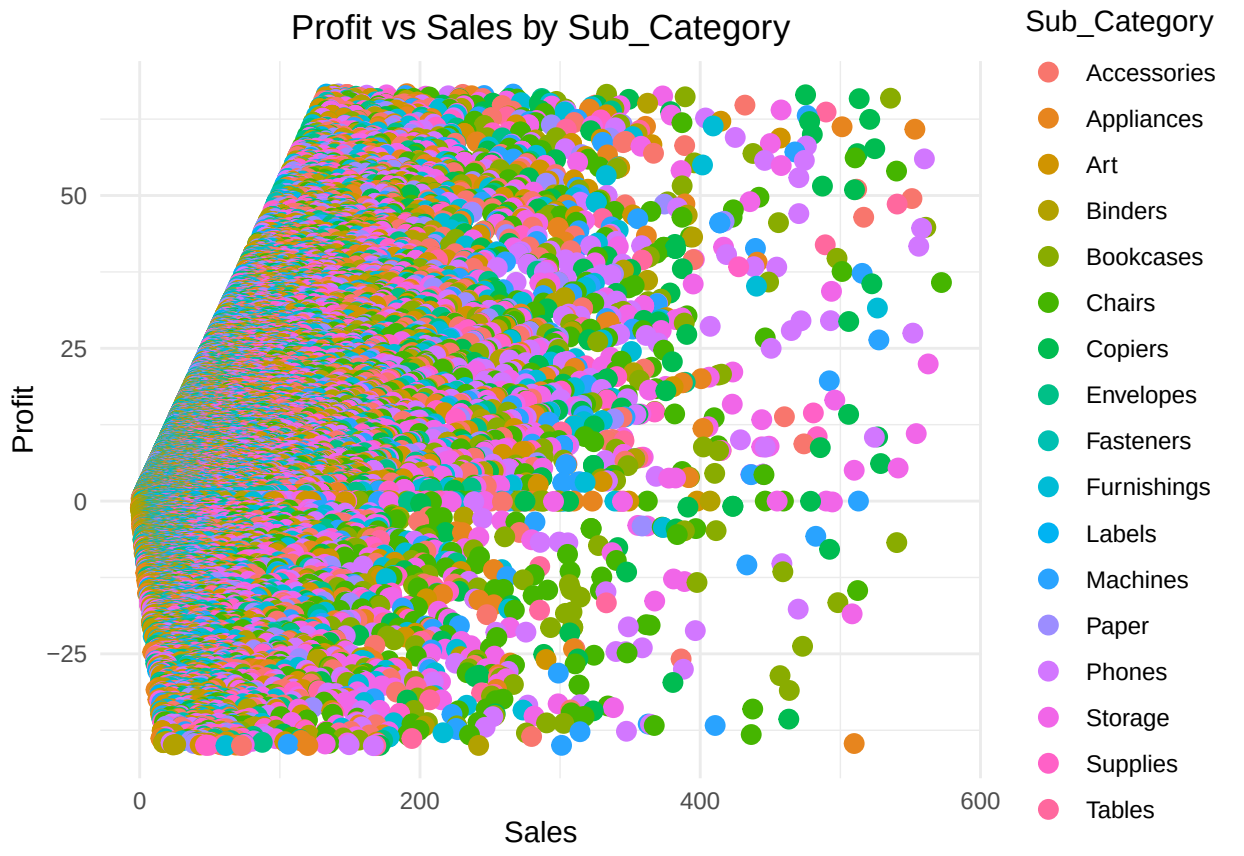
```
profit_plot <- ggplot(profit_data, aes(x = Total_Profit, y = Sub_Category , fill = Category)) +
  geom_bar(stat = "identity", position = "dodge") +
```

```
  geom_text(aes(label = scales::comma(Total_Profit)), vjust = -0.5, size = 3) +
  labs(title = "Profit by Category and Sub-Category", y = "Sub-Category", x = "Total Profit") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_manual(values = c("Furniture" = "#C8A2C8", "Office Supplies" = "turquoise", "Technology" =

# Mostrar el gráfico de ganancias
print(profit_plot)
```




```
#Chart 11 SCATTERPLOT BY CUSTOMER_ID
#The chart indicates a positive correlation between sales and profit across most
#sub-categories, with a dense clustering of data points around lower sales and
#profit values. Notably, sub-categories like Bookcases, Binders, and Phones show
#higher profit and sales, while others like Tables and Machines have more scattered
#lower values.

scatter_plot <- ggplot(data_2, aes(x = Sales, y = Profit, color = Sub_Category)) +
  geom_point(size = 3) +
  labs(title = "Profit vs Sales by Sub_Category",
       x = "Sales",
       y = "Profit",
       color = "Sub_Category") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

```r
print(scatter_plot)
```




```r
# 7. FEATURE ENGINEERING

# Load necessary libraries
# Then, apply your transformations
install.packages("zoo")
```

```
## Installing package into 'C:/Users/nadda/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

## package 'zoo' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'zoo'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\nadda\AppData\Local\R\win-library\4.2\00LOCK\zoo\libs\x64\zoo.dll to
## C:\Users\nadda\AppData\Local\R\win-library\4.2\zoo\libs\x64\zoo.dll: Permission
## denied

## Warning: restored 'zoo'

##
## The downloaded binary packages are in
##  C:\Users\nadda\AppData\Local\Temp\RtmpewMKaI\downloaded_packages
```

```r
library(dplyr)
library(lubridate)
library(zoo) # for rollapply
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
# FEATURE 1 Create Temporal Features
# This feature extracts relevant information from the order date. The intuition behind
# these features is to capture any seasonal, cyclical, or general trends over time that
# might be related to sales.

data_2 <- data_2 %>%
  mutate(
    Month = lubridate::month(Order_Date),
    Year = lubridate::year(Order_Date),
    DayOfWeek = lubridate::wday(Order_Date, label = TRUE)
  )

# FEATURES 2 Sort and then create Aggregated Historical Features
# This block organizes the data in chronological order and calculates measures that summarize
# past sales behavior, which can be useful for predicting future sales.
data_2 <- data_2 %>%
  arrange(Order_Date, Customer_ID) %>%
  group_by(Customer_ID) %>%
  mutate(
    Past_Sales = lag(Sales, order_by = Order_Date), # previous sale
    Rolling_Average_Sales = rollapply(Sales, width = 3, FUN = mean, partial = TRUE, fill = NA, align =
  ) %>%
  ungroup()

# FEATURES 3 Create Customer Behavior Features /
# This feature focuses on the customer's purchasing behavior and their relationship with the company, w

data_2 <- data_2 %>%
  group_by(Customer_ID) %>%
  mutate(
    Purchase_Frequency = n(), # counts the number of purchases by the customer
    Customer_Lifetime_Value = cumsum(Sales) # cumulative sum of sales
  ) %>%
  ungroup()

#FEATURES 4 Feature Based on Discounts
# Create discount bins
data_2$Discount_Bin <- cut(data_2$Discount, breaks = c(0, 0.05, 0.10, 0.20, 1),
                           labels = c("0-5%", "6-10%", "11-20%", ">20%"), include.lowest = TRUE)

# Calculate the average discount by category
data_2 <- data_2 %>%
```

```
  group_by(Category) %>%
  mutate(Average_Discount = mean(Discount, na.rm = TRUE),
         Discount_Above_Average = ifelse(Discount > Average_Discount, 1, 0)) %>%
  ungroup()

data_2$Discount_Above_Average <- factor(data_2$Discount_Above_Average)

# Create an interaction feature between discount and quantity for all categories
data_2$Discount_Quantity_Interaction <- data_2$Discount * data_2$Quantity


# FEATURES 5 Based on Sales
data_2 <- data_2 %>%
  arrange(Customer_ID, Order_Date) %>%
  group_by(Customer_ID) %>%
  mutate(Total_Sales_Accum = cumsum(Sales)) %>%
  ungroup()

# Set a sales threshold, for example, the average sales
average_sales <- mean(data_2$Sales, na.rm = TRUE)

data_2 <- data_2 %>%
  mutate(Sales_Above_Average = ifelse(Sales > average_sales, 1, 0))

data_2$Sales_Above_Average <- factor(data_2$Sales_Above_Average)

# FEATURES 6 Based on Shipping Cost
data_2 <- data_2 %>%
  mutate(
    Shipping_Cost_Per_Sale = Shipping_Cost / Sales,  # Costo de envío por unidad de venta
    High_Shipping_Cost = ifelse(Shipping_Cost_Per_Sale > median(Shipping_Cost_Per_Sale, na.rm = TRUE),
  )

data_2$High_Shipping_Cost <- factor(data_2$High_Shipping_Cost)
```

```
# DATA DIVISION BY TIME SERIES (2012-2014 training data, 2015 test data)
library(dplyr)
library(lubridate)

# Being the data in 'data_frame' and the date column is 'Order_Date'
data_2$Order_Date <- as.Date(data_2$Order_Date, format = "%Y-%m-%d")

# Sort the data by date
data_2 <- data_2 %>% arrange(Order_Date)

# Find the date that divides the sets
summary(data_2$Order_Date)  # This will give you an idea of the distribution of the dates
```

```
##         Min.      1st Qu.       Median         Mean      3rd Qu.         Max.
## "2012-01-01" "2013-06-19" "2014-07-09" "2014-05-11" "2015-05-21" "2015-12-31"
```

```r
# Determine the cut-off index based on a percentage
corte_index <- floor(0.8 * nrow(data_2))

# Split the data using the index
train_data <- data_2[1:corte_index, ]
test_data  <- data_2[(corte_index + 1):nrow(data_2), ]

# Check the first rows of the data frame to confirm the presence of new features
head(train_data)
```

```
## # A tibble: 6 x 40
##   Row_ID Orde_ID       Order_Date Ship_Date  Ship_Mode Customer_ID Customer_Name
##    <dbl> <chr>         <date>     <date>     <fct>     <fct>       <fct>
## 1  48883 HU-2012-AT73~ 2012-01-01 2012-01-05 Second C~ AT-73557    Annie Thurman
## 2  11731 IT-2012-EM14~ 2012-01-01 2012-01-05 Second C~ EM-14140124 Eugene Moren
## 3  22255 IN-2012-JH15~ 2012-01-01 2012-01-08 Standard~ JH-159857   Joseph Holt
## 4  22253 IN-2012-JH15~ 2012-01-01 2012-01-08 Standard~ JH-159857   Joseph Holt
## 5  22254 IN-2012-JH15~ 2012-01-01 2012-01-08 Standard~ JH-159857   Joseph Holt
## 6  49550 CA-2012-MM72~ 2012-01-02 2012-01-06 Standard~ MM-726023   Magdelene Mo~
## # i 33 more variables: Segment <fct>, City <fct>, State <fct>, Country <fct>,
## #   Region <fct>, Market <fct>, Product_ID <fct>, Category <fct>,
## #   Sub_Category <fct>, Product_Name <fct>, Sales <dbl>, Quantity <dbl>,
## #   Discount <dbl>, Profit <dbl>, Shipping_Cost <dbl>, Order_Priority <fct>,
## #   Original_Price <dbl>, Cost_Good <dbl>, Year <dbl>, Month <dbl>,
## #   DayOfWeek <ord>, Past_Sales <dbl>, Rolling_Average_Sales <dbl>,
## #   Purchase_Frequency <int>, Customer_Lifetime_Value <dbl>, ...
```

```r
# Or get a summary to see all the columns
summary(train_data)
```

```
##      Row_ID        Orde_ID            Order_Date           Ship_Date
##  Min.   :    3   Length:27439       Min.   :2012-01-01   Min.   :2012-01-03
##  1st Qu.:12696   Class :character   1st Qu.:2013-03-17   1st Qu.:2013-03-22
##  Median :26774   Mode  :character   Median :2014-01-25   Median :2014-01-30
##  Mean   :26138                      Mean   :2014-01-02   Mean   :2014-01-06
##  3rd Qu.:39401                      3rd Qu.:2014-11-05   3rd Qu.:2014-11-09
##  Max.   :51290                      Max.   :2015-07-10   Max.   :2015-07-16
##
##           Ship_Mode           Customer_ID                Customer_Name
##  First Class   : 3646   AP-109151404:   19   Bart Watters       :   64
##  Same Day      : 1279   EM-1396082  :   14   Chloris Kastensmidt:   62
##  Second Class  : 5410   WB-218501404:   14   Steven Ward        :   62
##  Standard Class:17104   BC-11125120 :   13   Karl Braun         :   58
##                         CK-122051406:   13   Kristen Hastings   :   58
##                         CS-121757   :   13   Art Ferguson       :   57
##                         (Other)     :27353   (Other)            :27078
##         Segment                 City                  State
##  Consumer   :14107   New York City:  502   California     : 1154
##  Corporate  : 8346   Los Angeles  :  466   England        :  724
##  Home Office: 4986   Philadelphia :  300   New York       :  627
##                      San Francisco:  290   Texas          :  597
##                      Houston      :  243   Ile-de-France  :  454
##                      Seattle      :  242   New South Wales:  420
```

```
##                      (Other)      :25396   (Other)        :23463
##           Country                  Region               Market
##  United States: 5776   Central America: 3181   Africa      :2628
##  Mexico       : 1499   Western Europe : 2896   Asia Pacific:7225
##  Australia    : 1467   Western US     : 1836   Europe      :5869
##  France       : 1373   Oceania        : 1804   LATAM       :5698
##  Germany      : 1064   Eastern US     : 1627   USCA        :6019
##  China        :  875   South America  : 1553
##  (Other)      :15385   (Other)        :14542
##        Product_ID               Category        Sub_Category
##  OFF-FA-6129:  178   Furniture      : 3442   Binders  : 4426
##  OFF-BI-4828:   72   Office Supplies:20936   Art      : 3436
##  OFF-BI-3737:   69   Technology     : 3061   Storage  : 2556
##  OFF-AR-5923:   66                           Paper    : 2486
##  OFF-BI-2917:   59                           Fasteners: 2074
##  OFF-BI-3293:   57                           Labels   : 2067
##  (Other)    :26938                           (Other)  :10394
##                                  Product_Name       Sales
##  Staples                                :  178   Min.   :  0.444
##  Ibico Index Tab, Clear                 :   72   1st Qu.: 20.808
##  Cardinal Index Tab, Clear              :   69   Median : 45.528
##  Sanford Pencil Sharpener, Water Color:   66   Mean   : 68.858
##  Acco Index Tab, Clear                  :   59   3rd Qu.: 94.200
##  Avery Index Tab, Clear                 :   57   Max.   :572.160
##  (Other)                                :26938
##     Quantity         Discount            Profit        Shipping_Cost
##  Min.   : 1.000   Min.   :0.000000   Min.   :-40.020   Min.   : 1.002
##  1st Qu.: 2.000   1st Qu.:0.000000   1st Qu.:  0.600   1st Qu.: 1.900
##  Median : 2.000   Median :0.000000   Median :  6.450   Median : 4.050
##  Mean   : 3.012   Mean   :0.001325   Mean   :  9.443   Mean   : 6.309
##  3rd Qu.: 4.000   3rd Qu.:0.002000   3rd Qu.: 17.730   3rd Qu.: 8.830
##  Max.   :14.000   Max.   :0.008000   Max.   : 66.660   Max.   :25.290
##
##   Order_Priority  Original_Price       Cost_Good           Year
##  Critical: 1676   Min.   :  0.7992   Min.   : -0.59   Min.   :2012
##  High    : 7986   1st Qu.: 23.7894   1st Qu.: 16.41   1st Qu.:2013
##  Low     : 1285   Median : 49.7400   Median : 36.36   Median :2014
##  Medium  :16492   Mean   : 75.0219   Mean   : 59.27   Mean   :2013
##                   3rd Qu.: 99.5400   3rd Qu.: 74.99   3rd Qu.:2014
##                   Max.   :723.3057   Max.   :696.80   Max.   :2015
##
##      Month        DayOfWeek       Past_Sales      Rolling_Average_Sales
##  Min.   : 1.000   dim\\.:2474   Min.   :  0.444   Min.   :  0.556
##  1st Qu.: 4.000   lun\\.: 568   1st Qu.: 22.435   1st Qu.: 29.307
##  Median : 7.000   mar\\.:5030   Median : 50.092   Median : 56.640
##  Mean   : 6.912   mer\\.:4982   Mean   : 73.754   Mean   : 73.272
##  3rd Qu.:10.000   jeu\\.:4778   3rd Qu.:100.658   3rd Qu.: 99.155
##  Max.   :12.000   ven\\.:4646   Max.   :572.160   Max.   :572.160
##                   sam\\.:4961   NA's   :12483
##  Purchase_Frequency Customer_Lifetime_Value Discount_Bin   Average_Discount
##  Min.   : 1.000     Min.   :   0.556        0-5%  :27439   Min.   :0.001229
##  1st Qu.: 2.000     1st Qu.:  46.484        6-10% :    0   1st Qu.:0.001319
##  Median : 3.000     Median : 108.365        11-20%:    0   Median :0.001319
##  Mean   : 3.637     Mean   : 159.160        >20%  :    0   Mean   :0.001330
```

```
##  3rd Qu.: 5.000     3rd Qu.: 214.700                      3rd Qu.:0.001319
##  Max.   :20.000     Max.   :1616.871                      Max.   :0.001492
##
##  Discount_Above_Average Discount_Quantity_Interaction Total_Sales_Accum
##  0:18557                Min.   :0.000000              Min.   :   0.556
##  1: 8882                1st Qu.:0.000000              1st Qu.:  46.484
##                         Median :0.000000              Median : 108.365
##                         Mean   :0.003884              Mean   : 159.160
##                         3rd Qu.:0.006000              3rd Qu.: 214.700
##                         Max.   :0.112000              Max.   :1616.871
##
##  Sales_Above_Average Shipping_Cost_Per_Sale High_Shipping_Cost
##  0:17977             Min.   :0.003492       0:13664
##  1: 9462             1st Qu.:0.068233       1:13775
##                      Median :0.098449
##                      Mean   :0.123149
##                      3rd Qu.:0.150641
##                      Max.   :2.274775
##
```

```r
# Formatting the data
train_data <- train_data %>%
  mutate(
    # Convert dates to Date type
    Order_Date = as.Date(Order_Date, format = "%Y-%m-%d"),
    Ship_Date = as.Date(Ship_Date, format = "%Y-%m-%d"),

    # Ensure numeric fields are treated as numeric
    Sales = as.numeric(gsub("[$,]", "", Sales)),
    Quantity = as.numeric(Quantity),
    Discount = as.numeric(sub("%$", "", Discount)) / 100,  # Convert percentage to decimal
    Profit = as.numeric(gsub("[$,]", "", Profit)),
    Shipping_Cost = as.numeric(Shipping_Cost),
    Original_Price = as.numeric(gsub("[$,]", "", Original_Price)),
    Cost_Good = as.numeric(gsub("[$,]", "", Cost_Good)),

    # Handle categorical data by converting them to factors
    Ship_Mode = as.factor(Ship_Mode),
    Customer_ID = as.factor(Customer_ID),
    Customer_Name = as.factor(Customer_Name),
    Segment = as.factor(Segment),
    City = as.factor(City),
    State = as.factor(State),
    Country = as.factor(Country),
    Region = as.factor(Region),
    Market = as.factor(Market),
    Product_ID = as.factor(Product_ID),
    Category = as.factor(Category),
    Sub_Category = as.factor(Sub_Category),
    Product_Name = as.factor(Product_Name),
    Order_Priority = as.factor(Order_Priority)
  )
```

```r
# 8. DATA MODELING GMB

#GRADIANT BOOSTING MACHINE
# Step 1: Load necessary libraries
install.packages("readxl")
```

```
## Warning: package 'readxl' is in use and will not be installed
```

```r
install.packages("caret")
```

```
## Installing package into 'C:/Users/nadda/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

## package 'caret' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'caret'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\nadda\AppData\Local\R\win-library\4.2\00LOCK\caret\libs\x64\caret.dll
## to C:\Users\nadda\AppData\Local\R\win-library\4.2\caret\libs\x64\caret.dll:
## Permission denied

## Warning: restored 'caret'

##
## The downloaded binary packages are in
##  C:\Users\nadda\AppData\Local\Temp\RtmpewMKaI\downloaded_packages
```

```r
install.packages("gbm")
```

```
## Installing package into 'C:/Users/nadda/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)

## package 'gbm' successfully unpacked and MD5 sums checked

## Warning: cannot remove prior installation of package 'gbm'

## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\nadda\AppData\Local\R\win-library\4.2\00LOCK\gbm\libs\x64\gbm.dll to
## C:\Users\nadda\AppData\Local\R\win-library\4.2\gbm\libs\x64\gbm.dll: Permission
## denied

## Warning: restored 'gbm'

##
## The downloaded binary packages are in
##  C:\Users\nadda\AppData\Local\Temp\RtmpewMKaI\downloaded_packages
```

```r
install.packages("dplyr")
```

```
## Warning: package 'dplyr' is in use and will not be installed
```

```r
install.packages("caTools")
```

```
## Installing package into 'C:/Users/nadda/AppData/Local/R/win-library/4.2'
## (as 'lib' is unspecified)
```

```
## package 'caTools' successfully unpacked and MD5 sums checked
```

```
## Warning: cannot remove prior installation of package 'caTools'
```

```
## Warning in file.copy(savedcopy, lib, recursive = TRUE): problem copying
## C:\Users\nadda\AppData\Local\R\win-library\4.2\00LOCK\caTools\libs\x64\caTools.dll
## to C:\Users\nadda\AppData\Local\R\win-library\4.2\caTools\libs\x64\caTools.dll:
## Permission denied
```

```
## Warning: restored 'caTools'
```

```
##
## The downloaded binary packages are in
##  C:\Users\nadda\AppData\Local\Temp\RtmpewMKaI\downloaded_packages
```

```r
install.packages("lubridate")
```

```
## Warning: package 'lubridate' is in use and will not be installed
```

```r
install.packages("zoo")
```

```
## Warning: package 'zoo' is in use and will not be installed
```

```r
library(readxl)
library(gbm)
```

```
## Loaded gbm 2.1.9
```

```
## This version of gbm is no longer under development. Consider transitioning to gbm3, https://github.c
```

```r
library(dplyr)
library(caTools)
library(lubridate)
library(caret)
```

```
## Loading required package: lattice
```

```r
# Fit the GBM model
# The Profit is the target variable and I want to use all other variables as predictors
gbm_model <- gbm(Profit ~ . - Orde_ID - Order_Date - Ship_Date - Customer_ID - City - State
           - Product_ID - Product_Name + Month + Year + Past_Sales + Rolling_Average_Sales
           + Purchase_Frequency + Discount_Above_Average + Total_Sales_Accum + Sales_Above_Average
           + Shipping_Cost_Per_Sale, # The '.' indicates that all other variables are used as predictor
           data = train_data,
           distribution = "gaussian",  # Use 'gaussian' for regression problems
           n.trees = 600,  # Number of trees to grow
           interaction.depth = 4,  # Interaction depth (depth of each tree)
           shrinkage = 0.01,  # Learning rate
           cv.folds = 5,  # Number of folds for cross-validation
           n.minobsinnode = 10,  # Minimum number of observations in terminal nodes
           verbose = TRUE)  # Print information during the training process
```

```
## Warning in gbm.fit(x = x, y = y, offset = offset, distribution = distribution,
## : variable 24: Discount_Bin has no variation.
```

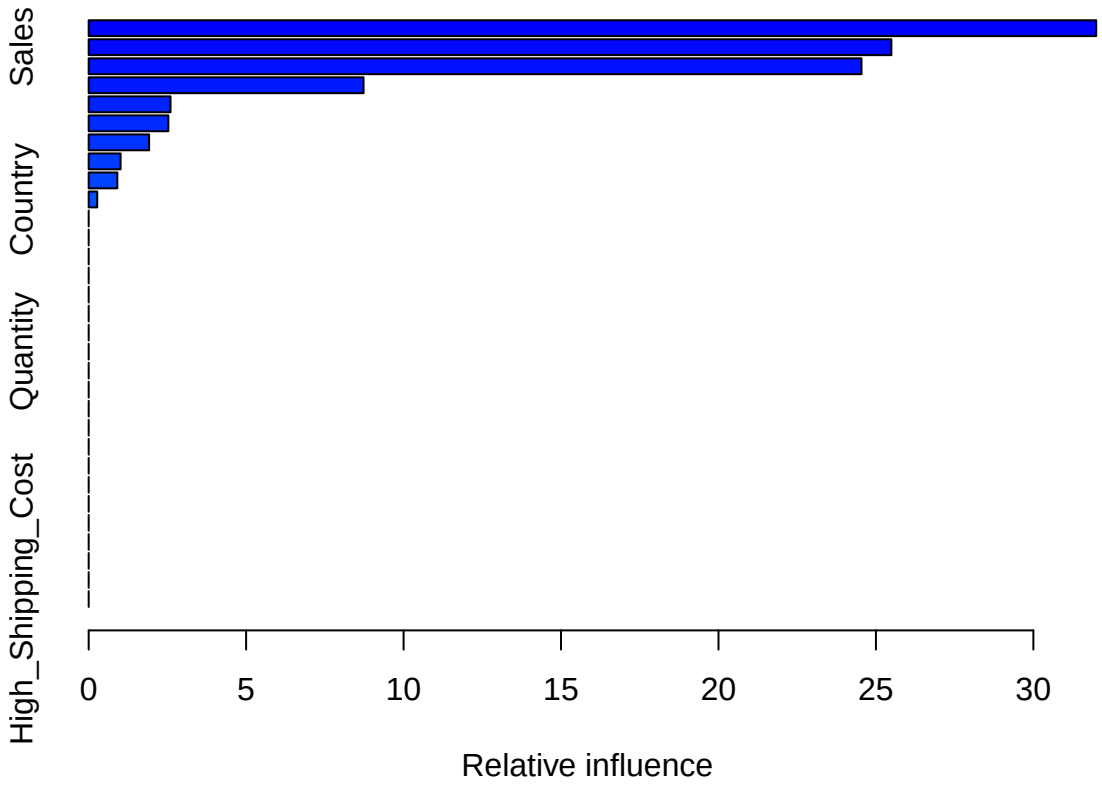
```
## Iter   TrainDeviance   ValidDeviance   StepSize   Improve
##      1      330.9489             nan     0.0100    3.2387
##      2      327.6493             nan     0.0100    3.1669
##      3      324.3917             nan     0.0100    3.0845
##      4      321.2379             nan     0.0100    2.9974
##      5      318.1530             nan     0.0100    2.9942
##      6      315.0843             nan     0.0100    2.9264
##      7      312.0919             nan     0.0100    2.8229
##      8      309.1345             nan     0.0100    2.8150
##      9      306.2953             nan     0.0100    2.7196
##     10      303.4558             nan     0.0100    2.6556
##     20      277.8248             nan     0.0100    2.2123
##     40      238.7360             nan     0.0100    1.4726
##     60      211.2161             nan     0.0100    1.1447
##     80      191.2202             nan     0.0100    0.6917
##    100      176.2682             nan     0.0100    0.4739
##    120      164.8209             nan     0.0100    0.3533
##    140      154.9988             nan     0.0100    0.2167
##    160      147.2117             nan     0.0100    0.2278
##    180      140.5866             nan     0.0100    0.2315
##    200      135.1571             nan     0.0100    0.0808
##    220      130.5448             nan     0.0100    0.0668
##    240      127.1533             nan     0.0100    0.0644
##    260      123.9584             nan     0.0100    0.0669
##    280      120.9781             nan     0.0100   -0.0523
##    300      118.4356             nan     0.0100    0.0444
##    320      115.7727             nan     0.0100    0.0538
##    340      113.7517             nan     0.0100   -0.0597
##    360      111.8506             nan     0.0100   -0.0536
##    380      109.9371             nan     0.0100    0.0013
##    400      108.2817             nan     0.0100   -0.0728
##    420      106.5787             nan     0.0100   -0.0008
##    440      105.1889             nan     0.0100   -0.0556
##    460      103.7176             nan     0.0100   -0.0645
##    480      102.3577             nan     0.0100   -0.0692
```

```
##    500      100.9850             nan     0.0100   -0.0566
##    520       99.7936             nan     0.0100   -0.0945
##    540       98.3856             nan     0.0100    0.0183
##    560       97.1280             nan     0.0100   -0.0888
##    580       95.9331             nan     0.0100   -0.0715
##    600       94.8885             nan     0.0100   -0.0656
```

```r
print(gbm_model)
```

```
## gbm(formula = Profit ~ . - Orde_ID - Order_Date - Ship_Date -
##     Customer_ID - City - State - Product_ID - Product_Name +
##     Month + Year + Past_Sales + Rolling_Average_Sales + Purchase_Frequency +
##     Discount_Above_Average + Total_Sales_Accum + Sales_Above_Average +
##     Shipping_Cost_Per_Sale, distribution = "gaussian", data = train_data,
##     n.trees = 600, interaction.depth = 4, n.minobsinnode = 10,
##     shrinkage = 0.01, cv.folds = 5, verbose = TRUE)
## A gradient boosted model with gaussian loss function.
## 600 iterations were performed.
## The best cross-validation iteration was 600.
## There were 31 predictors of which 10 had non-zero influence.
```

```r
# IMPORTANCE OF ENGINEERING FEATURES
# Adjust the margins (the numbers are the margins in lines for bottom, left, top, and right)
par(mar=c(4, 4, 2, 2))

# Obtain the feature importance
importance <- summary(gbm_model)
```

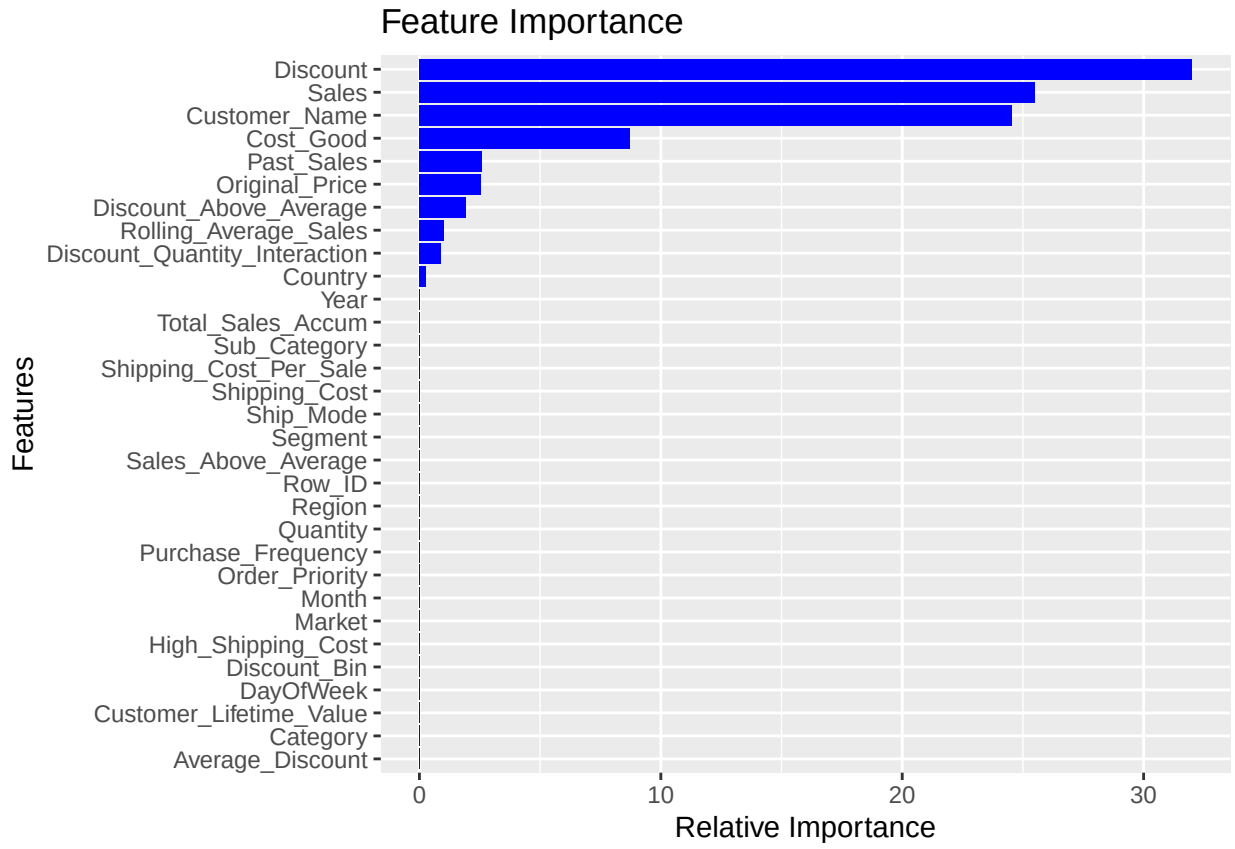



```
# Print the feature importance
print(importance)
```

```
##                                                          var    rel.inf
## Discount                                           Discount 31.9947893
## Sales                                                 Sales 25.4908449
## Customer_Name                                  Customer_Name 24.5416356
## Cost_Good                                          Cost_Good  8.7289341
## Past_Sales                                        Past_Sales  2.5989045
## Original_Price                                Original_Price  2.5304953
## Discount_Above_Average                Discount_Above_Average  1.9191297
## Rolling_Average_Sales                  Rolling_Average_Sales  1.0139810
## Discount_Quantity_Interaction Discount_Quantity_Interaction  0.9105358
## Country                                             Country  0.2707498
## Row_ID                                               Row_ID  0.0000000
## Ship_Mode                                         Ship_Mode  0.0000000
## Segment                                             Segment  0.0000000
## Region                                               Region  0.0000000
## Market                                               Market  0.0000000
## Category                                           Category  0.0000000
## Sub_Category                                   Sub_Category  0.0000000
## Quantity                                           Quantity  0.0000000
## Shipping_Cost                                 Shipping_Cost  0.0000000
## Order_Priority                               Order_Priority  0.0000000
## Year                                                 Year  0.0000000
## Month                                               Month  0.0000000
```

```
## DayOfWeek                                DayOfWeek  0.0000000
## Purchase_Frequency                Purchase_Frequency  0.0000000
## Customer_Lifetime_Value      Customer_Lifetime_Value  0.0000000
## Discount_Bin                          Discount_Bin  0.0000000
## Average_Discount                  Average_Discount  0.0000000
## Total_Sales_Accum                Total_Sales_Accum  0.0000000
## Sales_Above_Average        Sales_Above_Average  0.0000000
## Shipping_Cost_Per_Sale  Shipping_Cost_Per_Sale  0.0000000
## High_Shipping_Cost          High_Shipping_Cost  0.0000000
```

```r
# Create a data frame from the summary output
importance <- as.data.frame(importance)
names(importance) <- c("Feature", "RelativeImportance")

# Plot using ggplot2
ggplot(importance, aes(x = reorder(Feature, RelativeImportance), y = RelativeImportance)) +
  geom_col(fill = "blue") +
  coord_flip() +
  labs(title = "Feature Importance", x = "Features", y = "Relative Importance")
```




```r
# The chart "Feature Importance" shows the relative importance of various features in
# predicting a target variable. The most important features are **Discount**, **Sales**,
# and **Customer Name**, which have the highest relative importance scores. Other
# significant features include **Cost Good**, **Original Price**, and **Past Sales**.
# Features such as **Category**, **Average Discount**, and **Customer Lifetime Value**
```

```r
# are among the least important. This visualization highlights which factors are most
# influential in the model's predictions.

# FEATURES REMOVING VARIABLES We proceed to remove the following features because they are
# not relevant: High Shipping Cost, Customer Lifetime Value, Day of Week, Discount Bin,
# Average Discount

# APPLICATION OF THE MODEL ON THE TEST SET
# Predict using the test dataset
predictions <- predict(gbm_model, newdata = test_data, n.trees = 600)
print(predictions[1:10])
```

```
##  [1] -10.6795655 -14.0194516  27.4882782   7.8783738  -0.8452667   8.2034255
##  [7]  -7.0247883   8.6345362   7.9565159   6.9219630
```

```r
# MODEL PERFORMANCE CALCULATION (RMSE ERROR)
# Calculate RMSE
actual <- test_data$Profit  # Assuming 'Profit' is the variable to predict
rmse <- sqrt(mean((predictions - actual)^2))
print(paste("RMSE: ", rmse))
```

```
## [1] "RMSE:  14.3967382337813"
```

```r
data_2$Profit <- as.numeric(data_2$Profit)
min_profit <- min(data_2$Profit, na.rm = TRUE)  # na.rm = TRUE omits NA values in the calculation
max_profit <- max(data_2$Profit, na.rm = TRUE)
print(paste("The range of Profit goes from", min_profit, "to", max_profit))
```

```
## [1] "The range of Profit goes from -40.02 to 66.66"
```

```r
# 9. DATA MODELING LINEAR REGRESSION

# MULTIPLE LINEAR REGRESSION 1
Regression <- lm(Profit ~ Discount * Sub_Category * Sales, data = data_2)
print(Regression)
```

```
##
## Call:
## lm(formula = Profit ~ Discount * Sub_Category * Sales, data = data_2)
##
## Coefficients:
##                             (Intercept)                                 Discount
##                               1.104e+01                                -5.487e+03
##                  Sub_CategoryAppliances                      Sub_CategoryArt
##                               1.649e+00                                -6.907e+00
##                     Sub_CategoryBinders                 Sub_CategoryBookcases
##                              -7.387e+00                                 2.056e+01
##                      Sub_CategoryChairs                   Sub_CategoryCopiers
##                               5.382e+00                                 1.390e+01
##                   Sub_CategoryEnvelopes                 Sub_CategoryFasteners
##                              -6.443e+00                                -1.035e+01
```

```
##                Sub_CategoryFurnishings                     Sub_CategoryLabels
##                             -3.311e+00                             -1.089e+01
##                   Sub_CategoryMachines                      Sub_CategoryPaper
##                             6.486e+00                             -4.114e+00
##                    Sub_CategoryPhones                    Sub_CategoryStorage
##                             7.676e+00                             -2.682e+00
##                   Sub_CategorySupplies                     Sub_CategoryTables
##                             -4.514e+00                             -3.331e-01
##                                  Sales         Discount:Sub_CategoryAppliances
##                             1.029e-01                              1.057e+03
##               Discount:Sub_CategoryArt           Discount:Sub_CategoryBinders
##                             2.995e+03                              3.967e+03
##         Discount:Sub_CategoryBookcases            Discount:Sub_CategoryChairs
##                             -5.436e+03                             -1.317e+03
##           Discount:Sub_CategoryCopiers         Discount:Sub_CategoryEnvelopes
##                             -3.671e+03                              3.005e+03
##         Discount:Sub_CategoryFasteners       Discount:Sub_CategoryFurnishings
##                             4.768e+03                              1.946e+03
##            Discount:Sub_CategoryLabels          Discount:Sub_CategoryMachines
##                             4.842e+03                             -1.172e+03
##             Discount:Sub_CategoryPaper           Discount:Sub_CategoryPhones
##                             2.262e+03                             -2.338e+03
##           Discount:Sub_CategoryStorage          Discount:Sub_CategorySupplies
##                             1.598e+03                              2.491e+03
##            Discount:Sub_CategoryTables                         Discount:Sales
##                             -4.679e+03                             -1.400e+01
##           Sub_CategoryAppliances:Sales                  Sub_CategoryArt:Sales
##                             1.273e-02                              6.691e-02
##              Sub_CategoryBinders:Sales           Sub_CategoryBookcases:Sales
##                             9.143e-02                             -1.013e-01
##               Sub_CategoryChairs:Sales             Sub_CategoryCopiers:Sales
##                             -3.276e-02                             -8.201e-02
##             Sub_CategoryEnvelopes:Sales           Sub_CategoryFasteners:Sales
##                             8.040e-02                              1.605e-01
##          Sub_CategoryFurnishings:Sales              Sub_CategoryLabels:Sales
##                             3.820e-02                              1.918e-01
##             Sub_CategoryMachines:Sales               Sub_CategoryPaper:Sales
##                             -6.119e-02                             8.135e-02
##               Sub_CategoryPhones:Sales             Sub_CategoryStorage:Sales
##                             -5.141e-02                             -1.339e-02
##             Sub_CategorySupplies:Sales              Sub_CategoryTables:Sales
##                             4.000e-02                             -1.553e-02
##  Discount:Sub_CategoryAppliances:Sales         Discount:Sub_CategoryArt:Sales
##                             -2.287e+01                             -4.086e+01
##     Discount:Sub_CategoryBinders:Sales    Discount:Sub_CategoryBookcases:Sales
##                             -4.827e+01                              1.360e+01
##      Discount:Sub_CategoryChairs:Sales      Discount:Sub_CategoryCopiers:Sales
##                             -6.934e+00                              1.934e+01
##   Discount:Sub_CategoryEnvelopes:Sales    Discount:Sub_CategoryFasteners:Sales
##                             -4.450e+01                             -9.768e+01
## Discount:Sub_CategoryFurnishings:Sales       Discount:Sub_CategoryLabels:Sales
##                             -2.168e+01                             -8.925e+01
##    Discount:Sub_CategoryMachines:Sales        Discount:Sub_CategoryPaper:Sales
##                              1.181e+01                             -2.638e+01
```

```
##      Discount:Sub_CategoryPhones:Sales      Discount:Sub_CategoryStorage:Sales
##                              1.340e+01                              -1.376e+01
##    Discount:Sub_CategorySupplies:Sales       Discount:Sub_CategoryTables:Sales
##                             -4.851e+01                               3.110e+00
```

```
# Summarize the fitted model
summary(Regression)
```

```
##
## Call:
## lm(formula = Profit ~ Discount * Sub_Category * Sales, data = data_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -88.107  -6.135  -0.197   5.574  72.441
##
## Coefficients:
##                                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)                           1.104e+01  6.084e-01  18.146  < 2e-16
## Discount                             -5.487e+03  2.476e+02 -22.165  < 2e-16
## Sub_CategoryAppliances                1.649e+00  1.147e+00   1.438 0.150482
## Sub_CategoryArt                      -6.907e+00  6.973e-01  -9.904  < 2e-16
## Sub_CategoryBinders                  -7.387e+00  6.721e-01 -10.990  < 2e-16
## Sub_CategoryBookcases                 2.056e+01  1.937e+00  10.615  < 2e-16
## Sub_CategoryChairs                    5.382e+00  1.109e+00   4.852 1.23e-06
## Sub_CategoryCopiers                   1.390e+01  1.981e+00   7.017 2.30e-12
## Sub_CategoryEnvelopes                -6.443e+00  7.962e-01  -8.092 6.05e-16
## Sub_CategoryFasteners                -1.035e+01  7.818e-01 -13.233  < 2e-16
## Sub_CategoryFurnishings              -3.311e+00  7.828e-01  -4.229 2.35e-05
## Sub_CategoryLabels                   -1.089e+01  7.732e-01 -14.079  < 2e-16
## Sub_CategoryMachines                  6.486e+00  1.461e+00   4.438 9.12e-06
## Sub_CategoryPaper                    -4.114e+00  7.306e-01  -5.631 1.80e-08
## Sub_CategoryPhones                    7.676e+00  1.063e+00   7.218 5.39e-13
## Sub_CategoryStorage                  -2.682e+00  7.333e-01  -3.658 0.000255
## Sub_CategorySupplies                 -4.514e+00  8.139e-01  -5.546 2.95e-08
## Sub_CategoryTables                   -3.331e-01  6.021e+00  -0.055 0.955873
## Sales                                 1.029e-01  4.784e-03  21.517  < 2e-16
## Discount:Sub_CategoryAppliances       1.057e+03  3.906e+02   2.706 0.006815
## Discount:Sub_CategoryArt              2.995e+03  2.804e+02  10.680  < 2e-16
## Discount:Sub_CategoryBinders          3.967e+03  2.610e+02  15.198  < 2e-16
## Discount:Sub_CategoryBookcases       -5.436e+03  7.415e+02  -7.332 2.32e-13
## Discount:Sub_CategoryChairs          -1.317e+03  4.208e+02  -3.130 0.001749
## Discount:Sub_CategoryCopiers         -3.671e+03  8.091e+02  -4.537 5.72e-06
## Discount:Sub_CategoryEnvelopes        3.005e+03  3.169e+02   9.483  < 2e-16
## Discount:Sub_CategoryFasteners        4.768e+03  3.100e+02  15.381  < 2e-16
## Discount:Sub_CategoryFurnishings      1.946e+03  3.033e+02   6.415 1.42e-10
## Discount:Sub_CategoryLabels           4.842e+03  3.111e+02  15.565  < 2e-16
## Discount:Sub_CategoryMachines        -1.172e+03  4.782e+02  -2.452 0.014218
## Discount:Sub_CategoryPaper            2.262e+03  3.151e+02   7.179 7.19e-13
## Discount:Sub_CategoryPhones          -2.338e+03  4.381e+02  -5.338 9.46e-08
## Discount:Sub_CategoryStorage          1.598e+03  2.948e+02   5.420 6.00e-08
## Discount:Sub_CategorySupplies         2.491e+03  3.333e+02   7.473 8.02e-14
## Discount:Sub_CategoryTables          -4.679e+03  2.090e+03  -2.239 0.025170
## Discount:Sales                       -1.400e+01  2.790e+00  -5.019 5.21e-07
```

```
## Sub_CategoryAppliances:Sales            1.273e-02  8.578e-03   1.484 0.137816
## Sub_CategoryArt:Sales                   6.691e-02  6.274e-03  10.665  < 2e-16
## Sub_CategoryBinders:Sales               9.143e-02  6.490e-03  14.087  < 2e-16
## Sub_CategoryBookcases:Sales            -1.013e-01  9.712e-03 -10.431  < 2e-16
## Sub_CategoryChairs:Sales               -3.276e-02  7.473e-03  -4.384 1.17e-05
## Sub_CategoryCopiers:Sales              -8.201e-02  8.981e-03  -9.131  < 2e-16
## Sub_CategoryEnvelopes:Sales             8.040e-02  8.054e-03   9.983  < 2e-16
## Sub_CategoryFasteners:Sales             1.605e-01  1.152e-02  13.927  < 2e-16
## Sub_CategoryFurnishings:Sales           3.820e-02  6.781e-03   5.633 1.79e-08
## Sub_CategoryLabels:Sales                1.918e-01  1.399e-02  13.711  < 2e-16
## Sub_CategoryMachines:Sales             -6.119e-02  8.671e-03  -7.057 1.73e-12
## Sub_CategoryPaper:Sales                 8.135e-02  7.170e-03  11.346  < 2e-16
## Sub_CategoryPhones:Sales               -5.141e-02  7.098e-03  -7.243 4.48e-13
## Sub_CategoryStorage:Sales              -1.339e-02  5.860e-03  -2.285 0.022343
## Sub_CategorySupplies:Sales              4.000e-02  7.437e-03   5.378 7.57e-08
## Sub_CategoryTables:Sales               -1.553e-02  2.189e-02  -0.710 0.477947
## Discount:Sub_CategoryAppliances:Sales  -2.287e+01  4.802e+00  -4.762 1.93e-06
## Discount:Sub_CategoryArt:Sales         -4.086e+01  4.133e+00  -9.888  < 2e-16
## Discount:Sub_CategoryBinders:Sales     -4.827e+01  4.288e+00 -11.257  < 2e-16
## Discount:Sub_CategoryBookcases:Sales    1.360e+01  4.712e+00   2.886 0.003906
## Discount:Sub_CategoryChairs:Sales      -6.934e+00  3.923e+00  -1.768 0.077141
## Discount:Sub_CategoryCopiers:Sales      1.934e+01  4.851e+00   3.987 6.69e-05
## Discount:Sub_CategoryEnvelopes:Sales   -4.450e+01  5.102e+00  -8.721  < 2e-16
## Discount:Sub_CategoryFasteners:Sales   -9.768e+01  7.690e+00 -12.702  < 2e-16
## Discount:Sub_CategoryFurnishings:Sales -2.168e+01  3.874e+00  -5.594 2.23e-08
## Discount:Sub_CategoryLabels:Sales      -8.925e+01  9.221e+00  -9.679  < 2e-16
## Discount:Sub_CategoryMachines:Sales     1.181e+01  4.735e+00   2.495 0.012591
## Discount:Sub_CategoryPaper:Sales       -2.638e+01  5.275e+00  -5.001 5.73e-07
## Discount:Sub_CategoryPhones:Sales       1.340e+01  4.029e+00   3.326 0.000883
## Discount:Sub_CategoryStorage:Sales     -1.376e+01  3.482e+00  -3.951 7.81e-05
## Discount:Sub_CategorySupplies:Sales    -4.851e+01  5.185e+00  -9.357  < 2e-16
## Discount:Sub_CategoryTables:Sales       3.110e+00  9.590e+00   0.324 0.745719
##
## (Intercept)                           ***
## Discount                              ***
## Sub_CategoryAppliances
## Sub_CategoryArt                       ***
## Sub_CategoryBinders                   ***
## Sub_CategoryBookcases                 ***
## Sub_CategoryChairs                    ***
## Sub_CategoryCopiers                   ***
## Sub_CategoryEnvelopes                 ***
## Sub_CategoryFasteners                 ***
## Sub_CategoryFurnishings               ***
## Sub_CategoryLabels                    ***
## Sub_CategoryMachines                  ***
## Sub_CategoryPaper                     ***
## Sub_CategoryPhones                    ***
## Sub_CategoryStorage                   ***
## Sub_CategorySupplies                  ***
## Sub_CategoryTables
## Sales                                 ***
## Discount:Sub_CategoryAppliances       **
## Discount:Sub_CategoryArt              ***
```

```
## Discount:Sub_CategoryBinders           ***
## Discount:Sub_CategoryBookcases         ***
## Discount:Sub_CategoryChairs            **
## Discount:Sub_CategoryCopiers           ***
## Discount:Sub_CategoryEnvelopes         ***
## Discount:Sub_CategoryFasteners         ***
## Discount:Sub_CategoryFurnishings       ***
## Discount:Sub_CategoryLabels            ***
## Discount:Sub_CategoryMachines          *
## Discount:Sub_CategoryPaper             ***
## Discount:Sub_CategoryPhones            ***
## Discount:Sub_CategoryStorage           ***
## Discount:Sub_CategorySupplies          ***
## Discount:Sub_CategoryTables            *
## Discount:Sales                         ***
## Sub_CategoryAppliances:Sales
## Sub_CategoryArt:Sales                  ***
## Sub_CategoryBinders:Sales              ***
## Sub_CategoryBookcases:Sales            ***
## Sub_CategoryChairs:Sales               ***
## Sub_CategoryCopiers:Sales              ***
## Sub_CategoryEnvelopes:Sales            ***
## Sub_CategoryFasteners:Sales            ***
## Sub_CategoryFurnishings:Sales          ***
## Sub_CategoryLabels:Sales               ***
## Sub_CategoryMachines:Sales             ***
## Sub_CategoryPaper:Sales                ***
## Sub_CategoryPhones:Sales               ***
## Sub_CategoryStorage:Sales              *
## Sub_CategorySupplies:Sales             ***
## Sub_CategoryTables:Sales
## Discount:Sub_CategoryAppliances:Sales  ***
## Discount:Sub_CategoryArt:Sales         ***
## Discount:Sub_CategoryBinders:Sales     ***
## Discount:Sub_CategoryBookcases:Sales   **
## Discount:Sub_CategoryChairs:Sales      .
## Discount:Sub_CategoryCopiers:Sales     ***
## Discount:Sub_CategoryEnvelopes:Sales   ***
## Discount:Sub_CategoryFasteners:Sales   ***
## Discount:Sub_CategoryFurnishings:Sales ***
## Discount:Sub_CategoryLabels:Sales      ***
## Discount:Sub_CategoryMachines:Sales    *
## Discount:Sub_CategoryPaper:Sales       ***
## Discount:Sub_CategoryPhones:Sales      ***
## Discount:Sub_CategoryStorage:Sales     ***
## Discount:Sub_CategorySupplies:Sales    ***
## Discount:Sub_CategoryTables:Sales
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.55 on 34231 degrees of freedom
## Multiple R-squared:  0.5289, Adjusted R-squared:  0.528
## F-statistic: 573.7 on 67 and 34231 DF,  p-value: < 2.2e-16
```

```r
# MULTIPLE LINEAR REGRESSION 2

#Variable month
data_2 <- data_2 %>%
  mutate(month = month(Order_Date, label = TRUE))

Regression_month <- lm(Profit ~ month + Discount, data = data_2)
print(Regression_month)
```

```
##
## Call:
## lm(formula = Profit ~ month + Discount, data = data_2)
##
## Coefficients:
## (Intercept)      month.L      month.Q      month.C      month^4      month^5
##   1.606e+01    3.362e-01   -1.661e-01    1.621e-01   -2.532e-01    2.900e-01
##     month^6      month^7      month^8      month^9     month^10     month^11
##  -2.497e-01   -2.305e-02   -3.305e-01   -3.095e-01    7.821e-01    1.707e-02
##    Discount
##  -5.001e+03
```

```r
summary(Regression_month)
```

```
##
## Call:
## lm(formula = Profit ~ month + Discount, data = data_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -51.138 -10.727  -3.057   7.250  67.414
##
## Coefficients:
##               Estimate Std. Error  t value Pr(>|t|)
## (Intercept)  1.606e+01  1.009e-01  159.077  < 2e-16 ***
## month.L      3.362e-01  3.081e-01    1.091  0.27522
## month.Q     -1.661e-01  3.000e-01   -0.554  0.57979
## month.C      1.620e-01  2.976e-01    0.545  0.58604
## month^4     -2.532e-01  3.046e-01   -0.831  0.40587
## month^5      2.900e-01  3.041e-01    0.954  0.34017
## month^6     -2.497e-01  3.058e-01   -0.817  0.41403
## month^7     -2.305e-02  3.029e-01   -0.076  0.93933
## month^8     -3.305e-01  2.994e-01   -1.104  0.26970
## month^9     -3.095e-01  2.943e-01   -1.051  0.29305
## month^10     7.821e-01  2.859e-01    2.736  0.00623 **
## month^11     1.707e-02  2.879e-01    0.059  0.95272
## Discount    -5.001e+03  3.927e+01 -127.361  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.05 on 34286 degrees of freedom
## Multiple R-squared:  0.3217, Adjusted R-squared:  0.3214
## F-statistic:  1355 on 12 and 34286 DF,  p-value: < 2.2e-16
```