UNIFIED MENTOR

Supermarket Grocery Sales - Retail Analytics Report

Submitted by – Ishita

Intern ID - UMIP277158

Internship Domain- Data Science Intern

Introduction

Every day, the supermarket industry produces vast amounts of sales data that reveal information about consumer buying patterns, product demand, and regional performance. Businesses can improve their operations by gaining important insights from the analysis of this data about inventory management, sales patterns, and customer preferences.
Supermarkets deal with issues such seasonal sales fluctuations, inventory control, and shifting consumer demand. Businesses may make well-informed decisions that increase profitability, boost customer satisfaction, and optimize supply chain operations by utilizing data analytics. The goal of this project is to methodically handle and examine supermarket sales data in order to find important insights that can inform strategic business choices. We will investigate a number of supermarket sales-related topics through this analysis, including determining the top-selling items, comprehending regional variations in sales performance, and forecasting future sales patterns. The findings will also be made easier to understand through the use of visualization approaches. Supermarket managers and business analysts will benefit from the study's conclusions as they make data-driven decisions, enhance their marketing plans, and guarantee the best possible product availability.

The complete data analysis process will be covered in this paper, including exploratory analysis, machine learning modeling, performance evaluation, and data preprocessing. The objective is to show how data analytics can turn unstructured sales data into insightful knowledge that helps the retail industry.

Data Preprocessing

1. **Handling Missing Values**:
   - Missing values were identified using functions like is.na() and sum(is.na()).
   - Rows with missing values were removed to ensure the integrity of the dataset. This was essential as missing data can lead to biased or inaccurate results.

2. **Removing Duplicates**:
   - Duplicate entries were checked using duplicated() and removed to prevent uneven results. This ensures that each observation in the dataset is unique and contributes equally to the analysis.

3. **Encoding Categorical Variables**:
   - Categorical variables were converted into numerical formats using techniques such as one-hot encoding or label encoding. This allows the model to interpret categorical data effectively.

4. **Date Formatting**:
   - Date values were converted to appropriate date formats using the as.Date() function. Proper date formatting is crucial for conducting time-series analysis, as it allows for accurate chronological sorting and manipulation of date-related data.

Exploratory Data Analysis (EDA)

1. **Sales Distribution Across Categories**:
    - Histograms and bar plots were created using ggplot2 to visualize the distribution of sales across different categories. This helps identify which categories perform best and which may need further analysis or improvement.

```
ggplot(data, aes(x = Category, y = Sales)) +
geom_bar(stat = 'identity', fill = 'blue') +
ggtitle('Sales Distribution by Category') +
xlab('Category') +
ylab('Total Sales')
```

2. **Trends Over Time**:
    - Time-series plots were generated to examine sales trends over time. Line graphs were used to illustrate how sales fluctuate month by month or year by year, providing insights into seasonal trends and overall growth.

```
ggplot(sales_over_time, aes(x = Order.Date, y = Total.Sales)) +
geom_line(color = 'green') +
ggtitle('Sales Trends Over Time') +
xlab('Order Date') +
ylab('Total Sales')
```

3. **Correlation Between Numeric Features**:
    - Correlation matrices and scatter plots were utilized to explore relationships between numeric features. This analysis helps identify potential predictors for the sales model.

```
 cor_matrix <- cor(numeric_data)
ggplot(data = as.data.frame(cor_matrix), aes(x = Var1, y = Var2)) +
geom_tile(aes(fill = value), color = 'white') +
scale_fill_gradient2(low = 'blue', high = 'red', mid = 'white', limit = c(-1, 1), name = 'Correlation') +
theme_minimal() +
ggtitle('Correlation Matrix')
```

Predictive Modeling

1. **Splitting the Dataset**:
   - The dataset was divided into training and testing sets using the createDataPartition() function. An 80-20 split was utilized, ensuring that the model is trained on a substantial portion of the data while retaining a separate set for validation.

```
set.seed(42)
train_index <- createDataPartition(target, p = 0.8, list = FALSE)
X_train <- features[train_index, ]
X_test <- features[-train_index, ]
y_train <- target[train_index]
y_test <- target[-train_index]
```

2. **Normalization**:
   - The training data was normalized using centering and scaling methods. Normalization ensures that all features contribute equally to the model training, preventing features with larger scales from dominating the learning process.

```
preprocess_params <- preProcess(X_train, method = c("center", "scale"))
X_train <- predict(preprocess_params, X_train)
```

```
X_test <- predict(preprocess_params, X_test)
```

3. **Training the Model**:

    - A linear regression model was trained using the train() function from the caret package. The model learned the relationship between the features and the target variable (sales).

```
model <- train(X_train, y_train, method = 'lm')
```

4. **Making Predictions**:

    - Predictions were generated for the test set using the trained model. This step is crucial for evaluating the model's performance.

```
y_pred <- predict(model, X_test)
```

5. **Model Evaluation**:

    - The model's performance was evaluated using Mean Squared Error (MSE) and R-squared metrics. MSE measures the average squared difference between predicted and actual values, while R-squared indicates the proportion of variance explained by the model.

```
mse <- mean((y_test - y_pred)^2, na.rm = TRUE)
r2 <- cor(y_test, y_pred, use = "complete.obs")^2
print(paste('Mean Squared Error:', mse))
print(paste('R-squared:', r2))
```

R – CODE

```
# Load Libraries
library(ggplot2)
library(dplyr)
library(lubridate)
library(caret)
library(corrplot)
library(readr)
library(reshape2)

# Load Dataset
df <- read.csv("C:/Users/ishit/Downloads/supermarket datatset unified/Supermart Grocery Sales - Retail Analytics Dataset.csv", stringsAsFactors = TRUE)

# Data Preprocessing
df <- na.omit(df)
df <- df[!duplicated(df), ]

# Convert Order Date
df$Order.Date <- as.Date(df$Order.Date, format='%Y-%m-%d')
df$Order.Day <- day(df$Order.Date)
df$Order.Month <- month(df$Order.Date)
df$Order.Year <- year(df$Order.Date)

# Encode categorical variables
df$Category <- as.numeric(factor(df$Category))
df$Sub.Category <- as.numeric(factor(df$Sub.Category))
df$City <- as.numeric(factor(df$City))
df$Region <- as.numeric(factor(df$Region))
```

```
df$State <- as.numeric(factor(df$State))
df$Sales <- as.numeric(df$Sales)

# Exploratory Data Analysis
# Boxplot
ggplot(df, aes(x=factor(Category), y=Sales)) +
  geom_boxplot(fill='lightblue') +
  ggtitle('Sales Distribution by Category') +
  xlab('Category') +
  ylab('Sales')

# Ensure Order.Date is in Date format
df$Order.Date <- as.Date(df$Order.Date)

# Filter out NA values and summarize profit
profit_over_time <- df %>%
  filter(!is.na(Profit)) %>%  # Remove NA values
  group_by(Order.Date) %>%
  summarise(Total.Profit = sum(Profit, na.rm = TRUE))

# Plot the profit data as a bar plot
ggplot(profit_over_time, aes(x = Order.Date, y = Total.Profit)) +
  geom_bar(stat = 'identity', fill = 'green') +  # Use bar plot with filled color
  ggtitle('Total Profit Over Time') +
  ylab('Total Profit') +
  xlab('Order Date') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))  # Rotate x-axis labels for better
visibility
```

```
# Correlation Matrix
cor_matrix <- cor(df %>% select_if(is.numeric))
corrplot(cor_matrix, method='color', type='upper', tl.cex=0.8)

# Sales by City
ggplot(df, aes(x=factor(City), y=Sales)) +
  geom_bar(stat='summary', fun=mean, fill='skyblue') +
  ggtitle('Average Sales by City') +
  xlab('City') +
  ylab('Average Sales')

# Sales by Sub-Category
ggplot(df, aes(x=factor(Sub.Category), y=Sales)) +
  geom_boxplot(fill='lightgreen') +
  ggtitle('Sales Distribution by Sub-Category') +
  xlab('Sub-Category') +
  ylab('Sales')

# Pie Chart: Region-wise Sales
region_sales <- df %>% group_by(Region) %>% summarise(Total = sum(Sales))
region_sales$Region <- as.factor(region_sales$Region)
pie(region_sales$Total, labels=region_sales$Region, main="Sales Distribution by Region")

# Feature Selection and Model Building
features <- df %>% select(-c(Order.ID, Customer.Name, Order.Date, Sales))
target <- df$Sales
target

#split the dataset
```

```
set.seed(42)
train_index <- createDataPartition(target, p=0.8, list=FALSE)
X_train <- features[train_index, ]
X_test <- features[-train_index, ]
y_train <- target[train_index]
y_test <- target[-train_index]

# Normalize Data
preprocess_params <- preProcess(X_train, method=c("center", "scale"))
X_train <- predict(preprocess_params, X_train)
X_test <- predict(preprocess_params, X_test)

# Train Linear Regression Model
model <- train(X_train, y_train, method='lm')
y_pred <- predict(model, X_test)

# Check for NA values in predictions and actuals
if (any(is.na(y_test)) || any(is.na(y_pred))) {
  stop("NA values found in predictions or actuals.")
}

# Evaluate the Model
mse <- mean((y_test - y_pred)^2, na.rm = TRUE)  # Ensure NA removal
r2 <- cor(y_test, y_pred, use = "complete.obs")^2  # Use complete cases for correlation

# Print results
print(paste('Mean Squared Error:', mse))
print(paste('R-squared:', r2))
```

Actual vs Predicted Plot

ggplot(data.frame(Actual=y_test, Predicted=y_pred), aes(x=Actual, y=Predicted)) +

  geom_point(color='blue') +

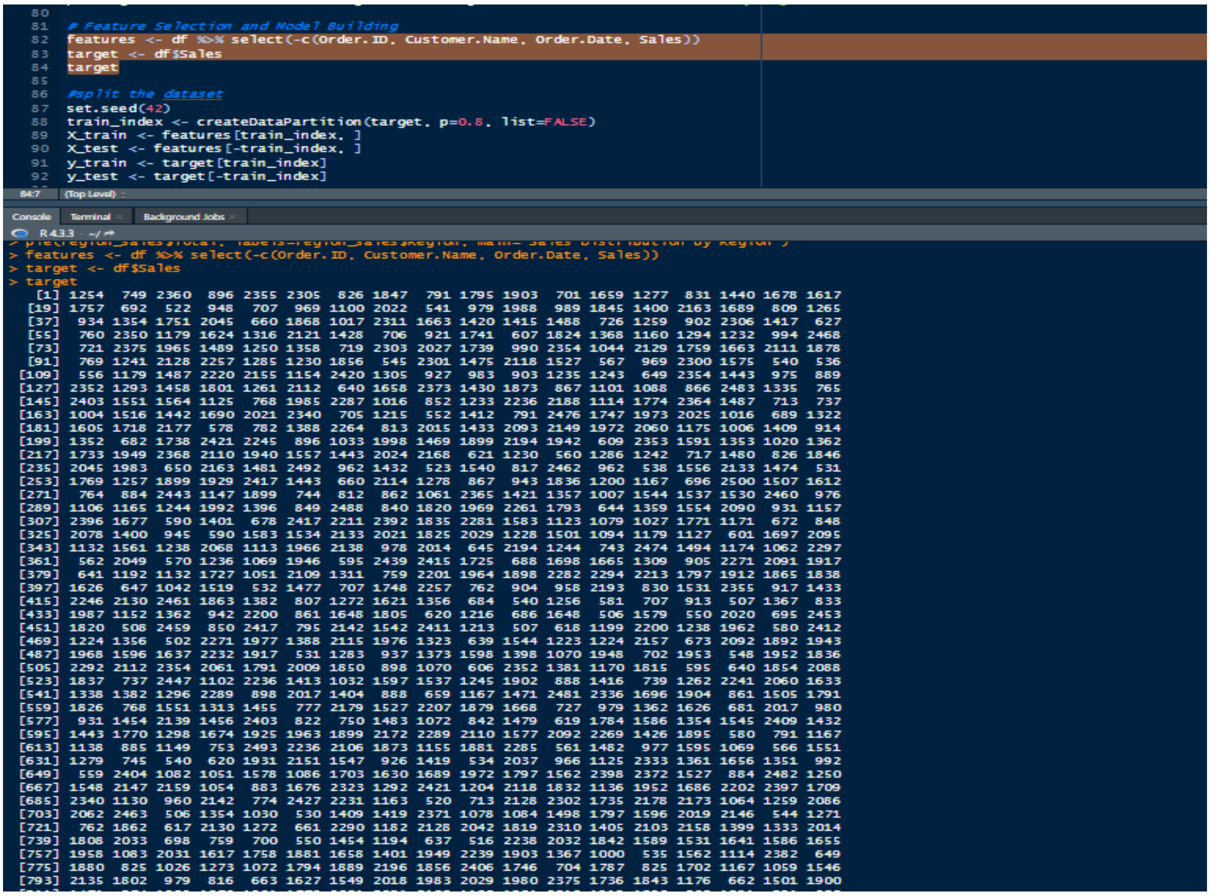
  geom_abline(slope=1, intercept=0, color='red') +

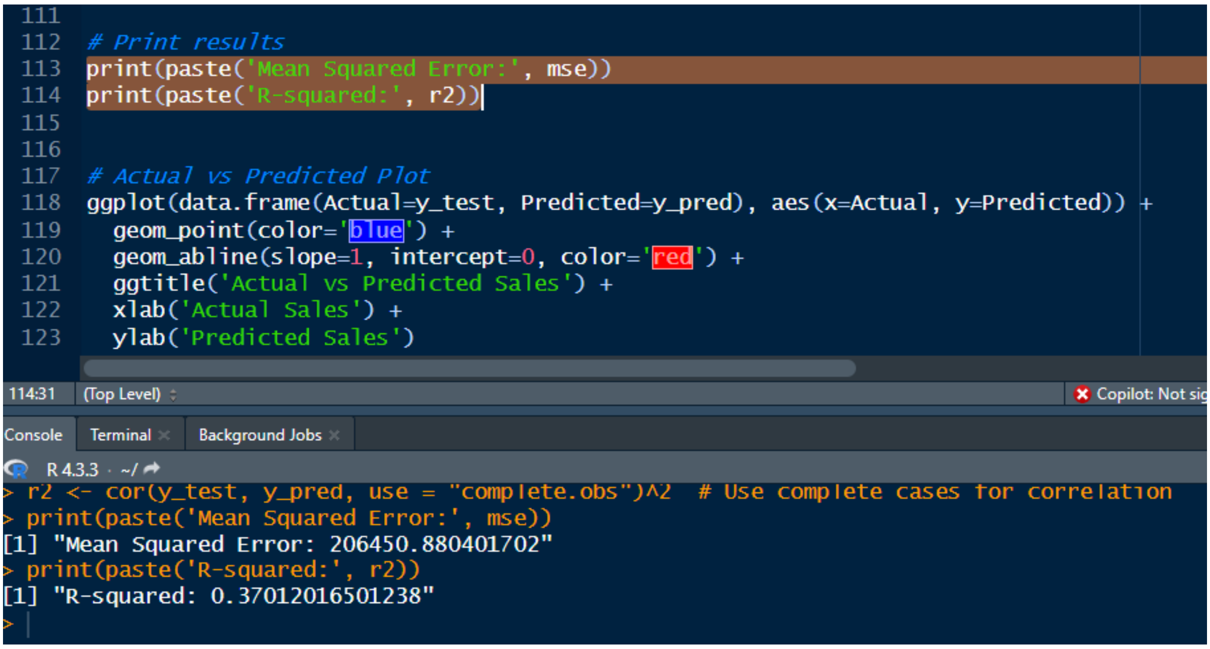
  ggtitle('Actual vs Predicted Sales') +

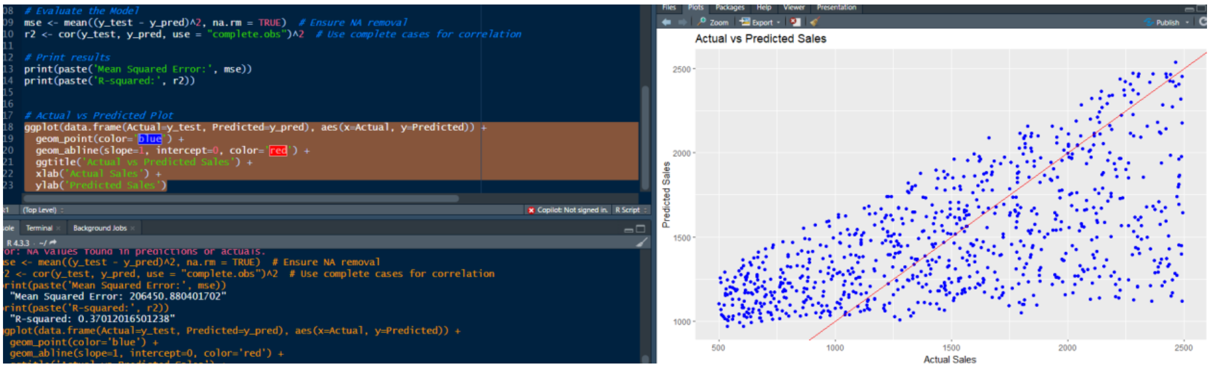
  xlab('Actual Sales') +

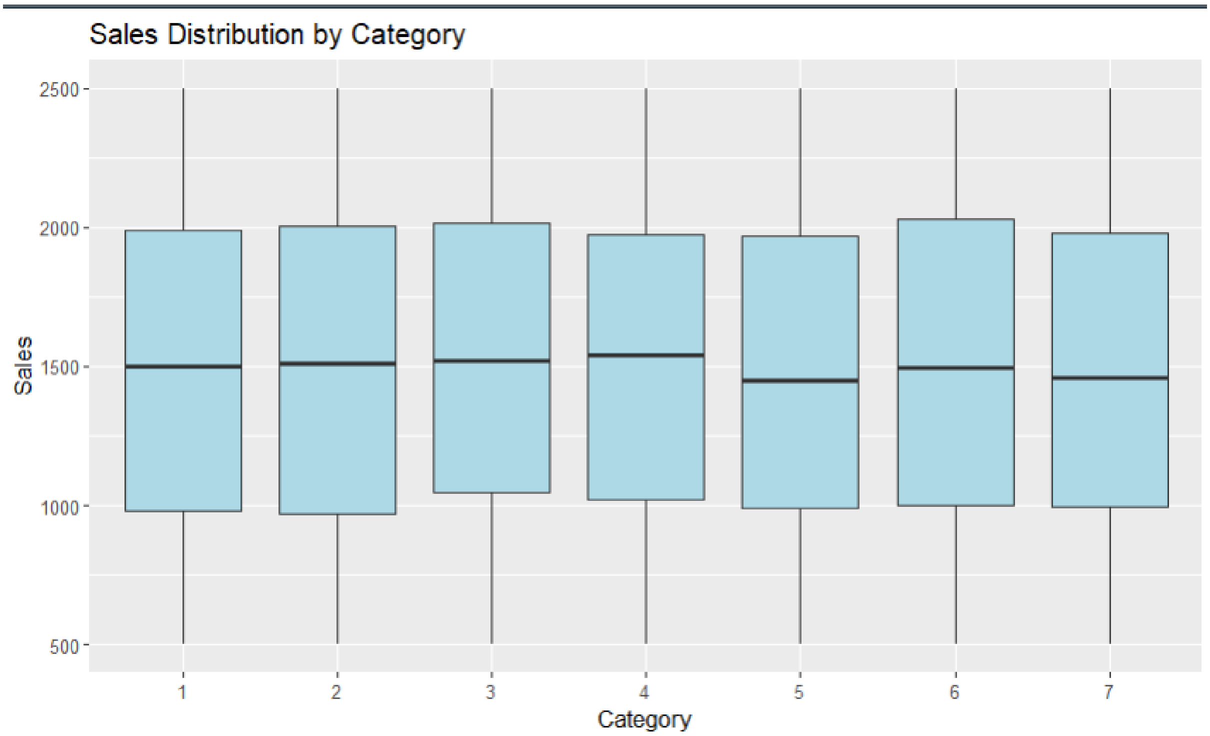
  ylab('Predicted Sales')

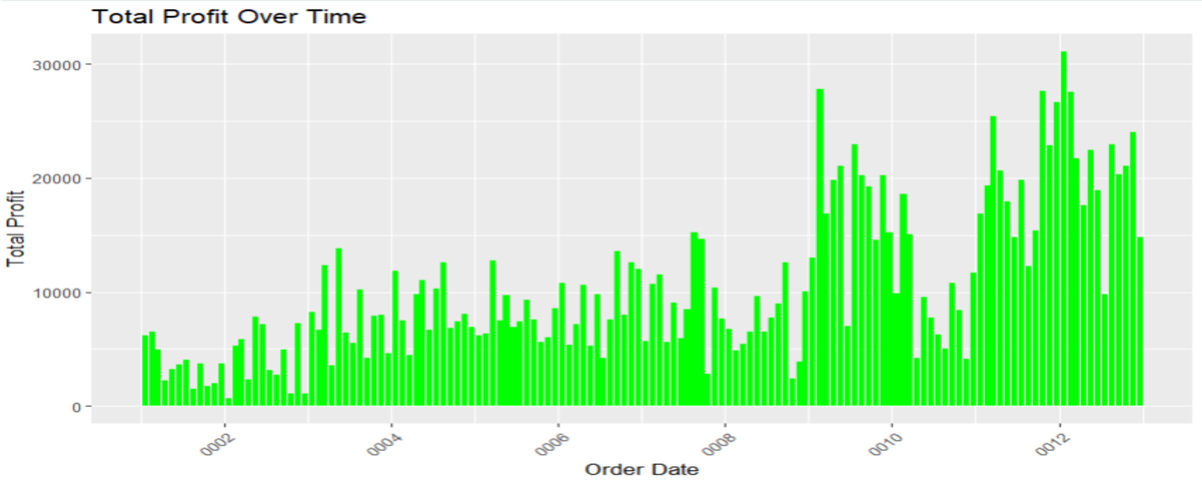
<u>Outputs</u>

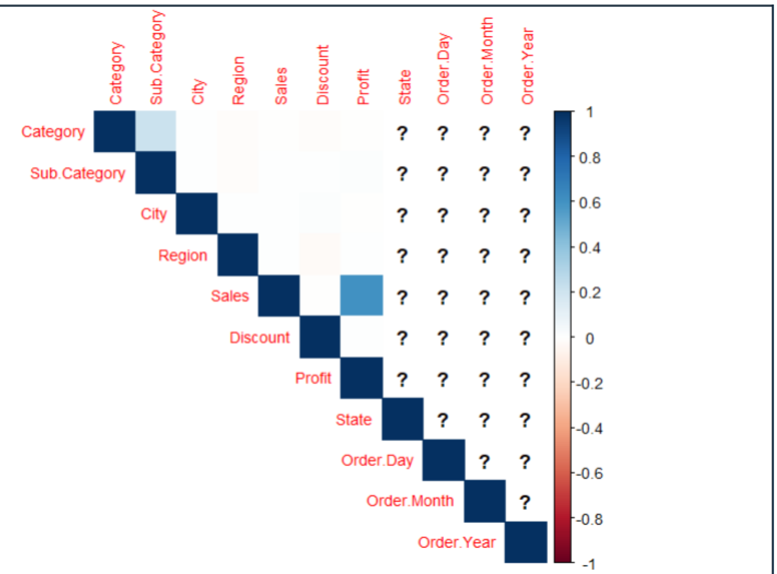


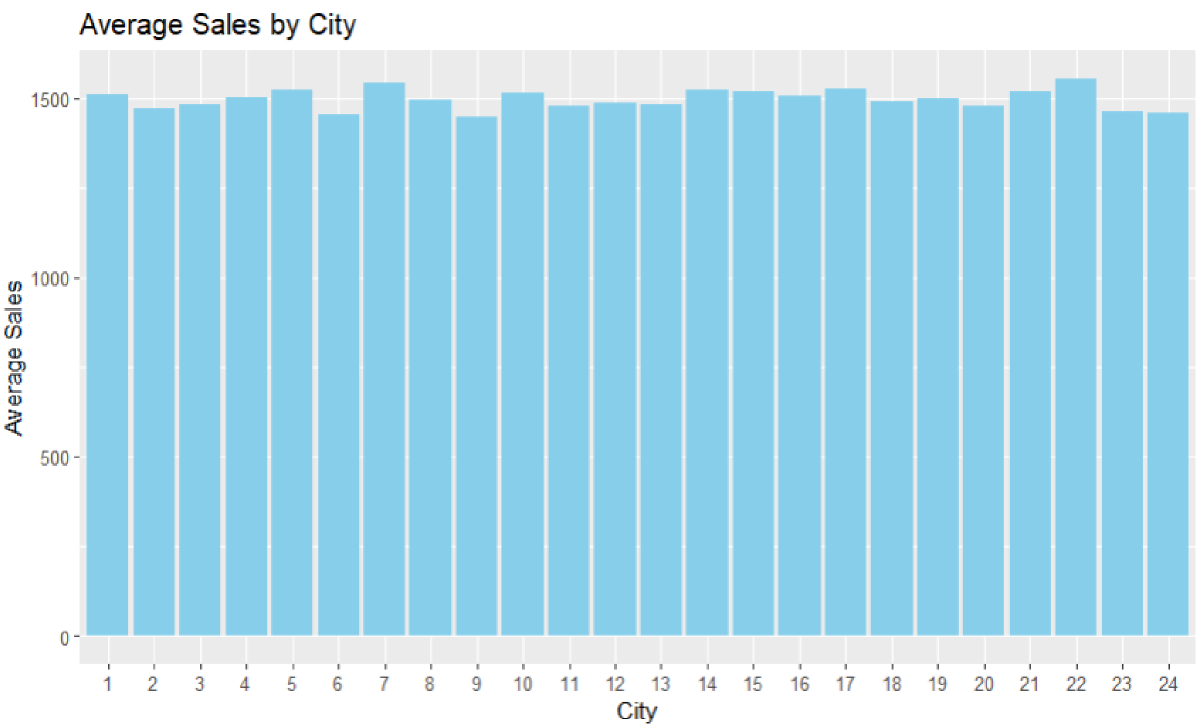

# Print results
print(paste('Mean Squared Error:', mse))
print(paste('R-squared:', r2))
# Actual vs Predicted Plot
ggplot(data.frame(Actual=y_test, Predicted=y_pred), aes(x=Actual, y=Predicted)) +
geom_point(color='blue') +
geom_abline(slope=1, intercept=0, color='red') +
ggtitle('Actual vs Predicted Sales') +
xlab('Actual Sales') +
ylab('Predicted Sales')
114:31
(Top Level)
Copilot: Not sig
Console
Terminal
Background Jobs
R 4.3.3 · ~/
> r2 <- cor(y_test, y_pred, use = "complete.obs")^2 # Use complete cases for correlation
> print(paste('Mean Squared Error:', mse))
[1] "Mean Squared Error: 206450.880401702"
> print(paste('R-squared:', r2))
[1] "R-squared: 0.37012016501238"
>

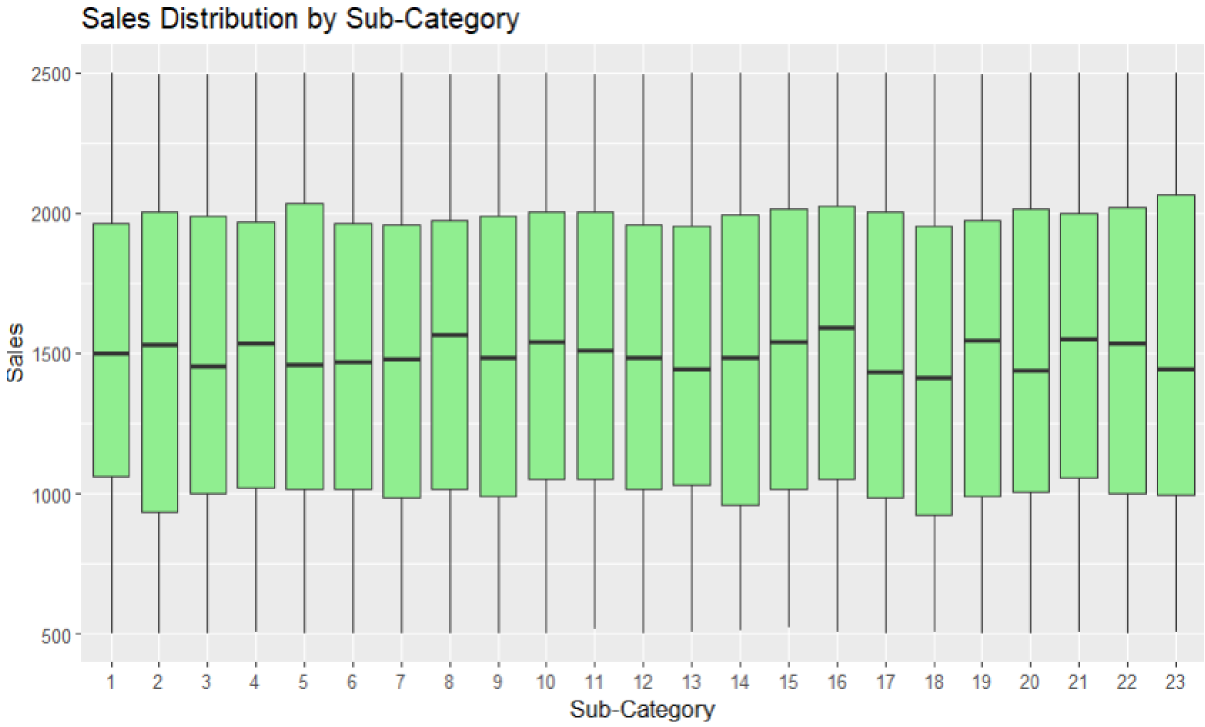

Actual vs Predicted Sales
Predicted Sales
Actual Sales
2500
2000
1500
1000
500
1000
1500
2000
2500

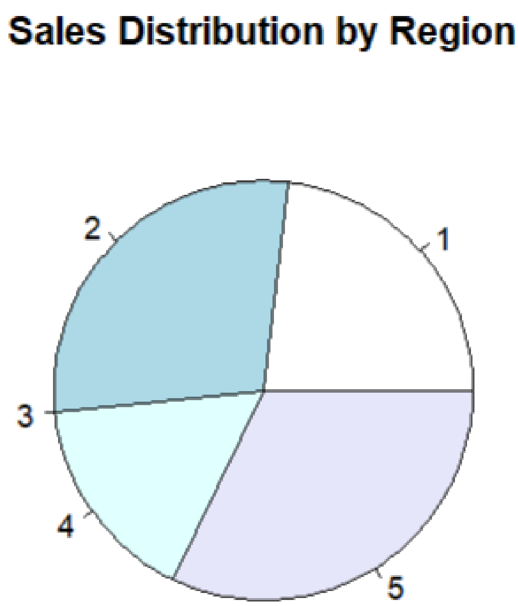

Sales Distribution by Category
Sales
2500
2000
1500
1000
500
1
2
3
4
5
6
7
Category


Total Profit Over Time
Total Profit
30000
20000
10000
0
0002
0004
0006
0008
0010
0012
Order Date


Category
Sub.Category
City
Region
Sales
Discount
Profit
State
Order.Day
Order.Month
Order.Year
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1


Average Sales by City
Average Sales
1500
1000
500
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
City


Sales Distribution by Sub-Category
Sales
2500
2000
1500
1000
500
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
Sub-Category


Sales Distribution by Region
1
2
3
4
5

CONCLUSION

The analysis provided valuable insights into product category performance, revealing high-revenue categories such as electronics and clothing, while identifying underperformers like home goods. By examining sales trends over time, we discovered seasonal variations, highlighting peaks during holidays and the need for timely marketing campaigns. Additionally, potential areas for optimization were identified, including inventory management to align stock with anticipated demand, targeted marketing strategies for both high-performing and underperforming categories, and customer segmentation to enhance engagement. Predictive modeling further enabled forecasting of future sales trends based on historical data, facilitating better resource allocation and risk mitigation. This comprehensive approach ensures informed decision-making and strategic planning for sustained growth in a competitive marketplace.