Explanation of Sales Data Analysis Project

This project focusses on analyzing the sales in different regions using a comprehensive Kaggle Superstore dataset. The objective is to identify which products were sold the most, which regions or customer segments are most profitable and what are the monthly/seasonal trends in sales and profit.

Objective:

The primary objective of the project is to use historical sales data to:

- Examine the total sales over different products to find sales by category
- To Calculate the total profit over different regions such as North, South, East and West region
- To Investigate Top 10 products sold
- To Identify monthly sales trend i.e. Sales over Time:

The Kaggle Superstore dataset includes several variables that capture Order ID, Order Date, Ship Date, Ship Mode, Customer ID, Customer Name, Segment, Country, City, State, postal code, Region, Product ID, Category, Sub-Category, Product Name, Sales, Quantity, Discount, Profit to perform sales analysis and determine profit margin by deriving actionable insights.

Features:

Order ID—ID of the order

Order Date—Date of the order placed

Ship Date—Date of the order shipped

Ship Mode—Mode of the shipping like first class, second class, standard class

Customer ID—ID of the customer

Customer Name—Name of the customer

Segment—Such as Consumer, Home Office, Corporate

Country—Country from where the order placed and to be shipped

City—city of the order placed and to be shipped

State—state of the order placed

Postal code—postal code of the order placed

Region—region of the order such as north, south east and west

Product ID—Id of the product

Category—such as furniture, technology, office supplies

Sub-Category— such as Bookcases, Chairs, Labels, Tables, Storage, Furnishings, Art, Phones, Binders, Appliances

Product Name—Name of the product

Sales—sold product

Quantity—quantity sold

Discount—discount given

Profit—profit obtained

Step 1: Data Loading and Cleaning

Objective: Load the data and clean it to ensure that the dataset is ready for analysis. This involves handling missing values, ensuring consistency, and preparing the data for analysis.

Explanation:

- First load the dataset using pandas and inspect it for missing values.
- Remove the duplicates, fix the datatypes and Check for extreme values in Sales or Profit.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

# Configure visuals
plt.style.use('seaborn-v0_8-whitegrid')

sns.set_palette("viridis")


# Update the file path to match your local setup
df = pd.read_csv(r"C:\Users\padma\SampleSuperstore.csv", encoding ='latin1')

# Preview dataset
df.head()


# Basic info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Row ID         9994 non-null   int64
 1   Order ID       9994 non-null   object
 2   Order Date     9994 non-null   object
 3   Ship Date      9994 non-null   object
 4   Ship Mode      9994 non-null   object
 5   Customer ID    9994 non-null   object
 6   Customer Name  9994 non-null   object
 7   Segment        9994 non-null   object
 8   Country        9994 non-null   object
 9   City           9994 non-null   object
 10  State          9994 non-null   object
 11  Postal Code    9994 non-null   int64
 12  Region         9994 non-null   object
 13  Product ID     9994 non-null   object
 14  Category       9994 non-null   object
 15  Sub-Category   9994 non-null   object
 16  Product Name   9994 non-null   object
 17  Sales          9994 non-null   float64
 18  Quantity       9994 non-null   int64
 19  Discount       9994 non-null   float64
 20  Profit         9994 non-null   float64
dtypes: float64(3), int64(3), object(15)
memory usage: 1.6+ MB
```

Python code to exploring the data, by finding the summary of the statistics , removing the duplicates, checking for missing values, fixing the data types by converting order date to datetime and confirming the cleanup

```python
# Summary statistics
df.describe(include='all')

# Check for missing values
df.isnull().sum()

# Check for duplicates
df.duplicated().sum()


# Remove duplicates
df.drop_duplicates(inplace=True)

# Convert 'Order Date' to datetime
df['Order Date'] = pd.to_datetime(df['Order Date'], errors='coerce')

# Optional: Handle missing data (if any)
df.dropna(inplace=True)

# Confirm cleanup
df.info()
```

Output of the above code showing memory usage, type of the data used as well the column labels or names with their corresponding data types

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Row ID         9994 non-null   int64
 1   Order ID       9994 non-null   object
 2   Order Date     9994 non-null   datetime64[ns]
 3   Ship Date      9994 non-null   object
 4   Ship Mode      9994 non-null   object
 5   Customer ID    9994 non-null   object
 6   Customer Name  9994 non-null   object
 7   Segment        9994 non-null   object
 8   Country        9994 non-null   object
 9   City           9994 non-null   object
 10  State          9994 non-null   object
 11  Postal Code    9994 non-null   int64
 12  Region         9994 non-null   object
 13  Product ID     9994 non-null   object
 14  Category       9994 non-null   object
 15  Sub-Category   9994 non-null   object
 16  Product Name   9994 non-null   object
 17  Sales          9994 non-null   float64
 18  Quantity       9994 non-null   int64
 19  Discount       9994 non-null   float64
 20  Profit         9994 non-null   float64
dtypes: datetime64[ns](1), float64(3), int64(3), object(14)
memory usage: 1.6+ MB
```

Python code to obtain or extract the year and month for trend analysis

```python
# Extract year and month for trend analysis
df['Year'] = df['Order Date'].dt.year
df['Month'] = df['Order Date'].dt.month_name()

# Verify new columns
df[['Order Date', 'Year', 'Month']].head()
```

The output indicates the following order dates are trending with the months being November June, October and year being 2016 and 2015

	Order Date	Year	Month
0	2016-11-08	2016	November
1	2016-11-08	2016	November
2	2016-06-12	2016	June
3	2015-10-11	2015	October
4	2015-10-11	2015	October

Python code to find the Top 10 selling products through product name and sales. In addition, to investigating the profit by region such as north, south, west and east region and sales over time

```
# --- Top-selling products ---
top_products = df.groupby('Product Name')['Sales'].sum().sort_values(ascending=False).head(10)
print(top_products)

# --- Profit by region ---
profit_by_region = df.groupby('Region')['Profit'].sum().sort_values(ascending=False)
print(profit_by_region)

# --- Sales trend over time ---
df['Month_Year'] = df['Order Date'].dt.to_period('M')
sales_trend = df.groupby('Month_Year')['Sales'].sum()
```

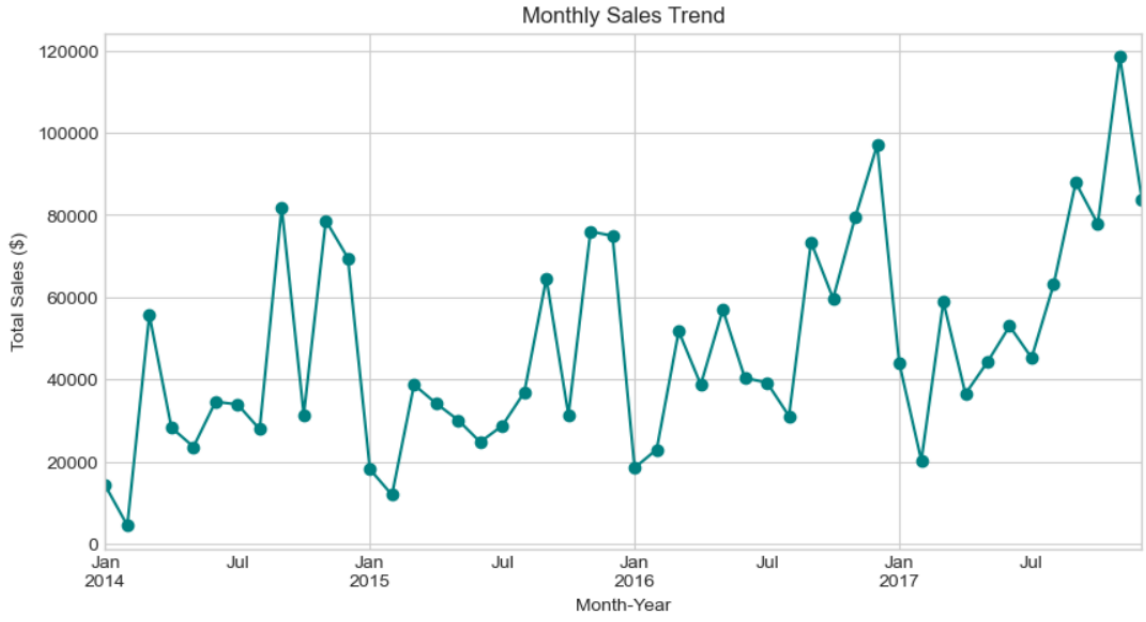
The output shows each regions profit as well top 10 selling products

```
Product Name
Canon imageCLASS 2200 Advanced Copier                                  61599.824
Fellowes PB500 Electric Punch Plastic Comb Binding Machine with Manual Bind    27453.384
Cisco TelePresence System EX90 Videoconferencing Unit                  22638.480
HON 5400 Series Task Chairs for Big and Tall                           21870.576
GBC DocuBind TL300 Electric Binding System                             19823.479
GBC Ibimaster 500 Manual ProClick Binding System                       19024.500
Hewlett Packard LaserJet 3310 Copier                                   18839.686
HP Designjet T520 Inkjet Large Format Printer - 24" Color              18374.895
GBC DocuBind P400 Electric Binding System                              17965.068
High Speed Automatic Electric Letter Opener                            17030.312
Name: Sales, dtype: float64
Region
West       108418.4489
East        91522.7800
South       46749.4303
Central     39706.3625
Name: Profit, dtype: float64
```

Python code for finding the sales over time to find the monthly sales trend

```python
# --- 7.1: Monthly Sales Trend ---
plt.figure(figsize=(10,5))
sales_trend.plot(kind='line', marker='o', color='teal')
plt.title("Monthly Sales Trend")
plt.xlabel("Month-Year")
plt.ylabel("Total Sales ($)")
plt.show()
```

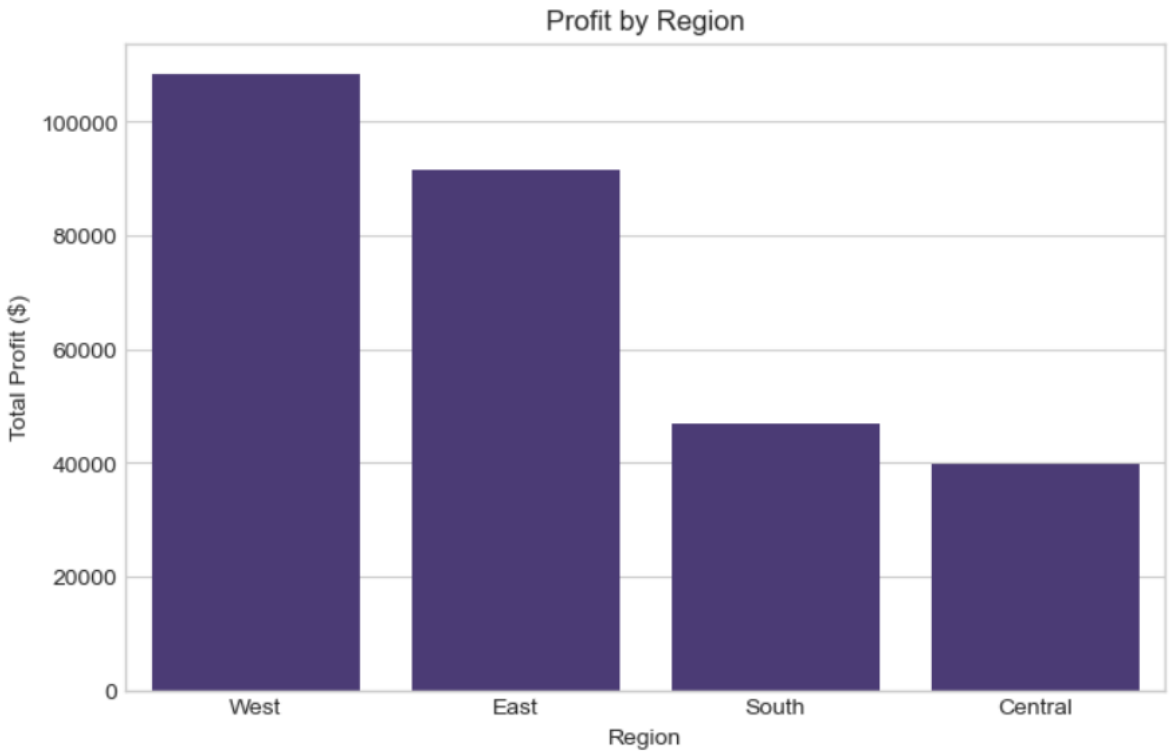
The output of the sales over time is identified and depicted in the line graph below



Python code to find the profit by region

```
# --- 7.2: Profit by Region ---
plt.figure(figsize=(8,5))
sns.barplot(x=profit_by_region.index, y=profit_by_region.values)
plt.title("Profit by Region")
plt.ylabel("Total Profit ($)")
plt.show()
```

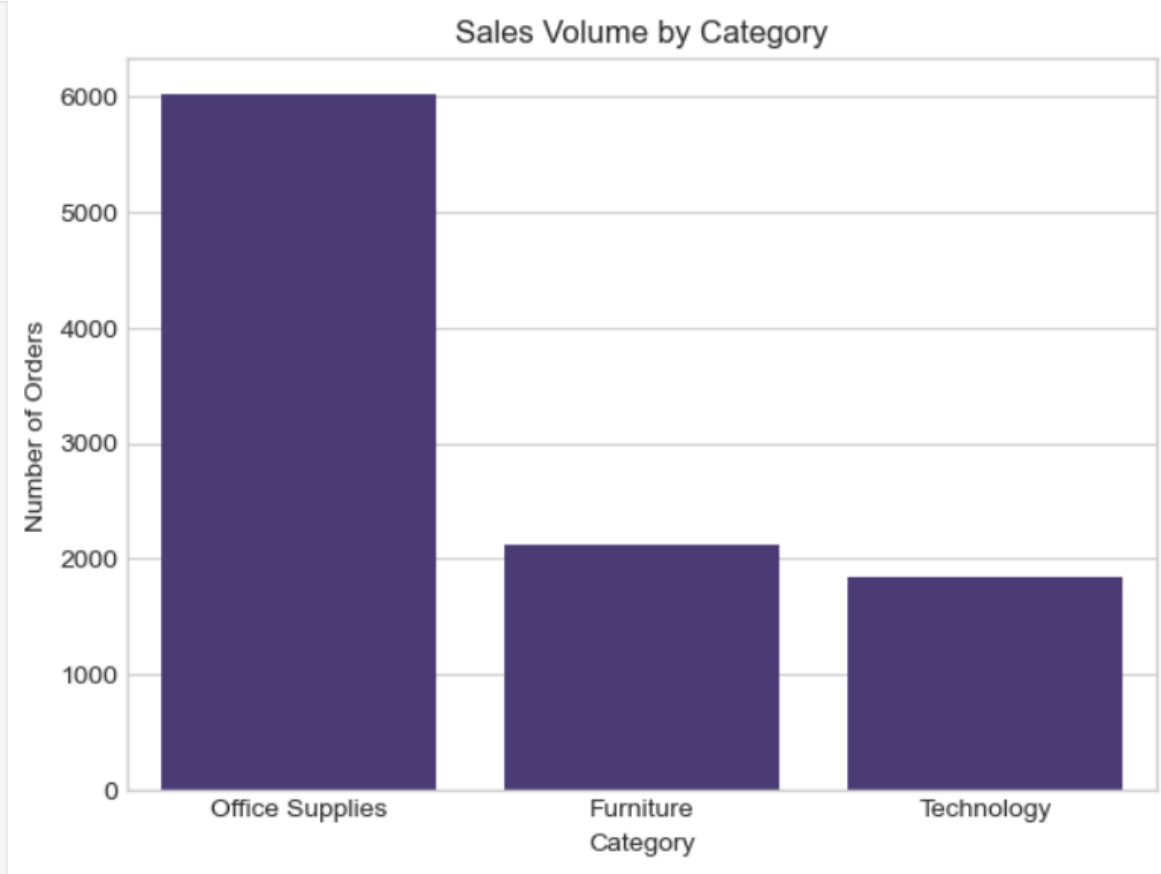
The output of the profit by region depicted through bar graph



Python code to find the sales by category

```
# --- 7.3: Sales by Category ---
plt.figure(figsize=(7,5))
sns.barplot(x=df['Category'].value_counts().index, y=df['Category'].value_counts().values)
plt.title("Sales Volume by Category")
plt.xlabel("Category")
plt.ylabel("Number of Orders")
plt.show()
```

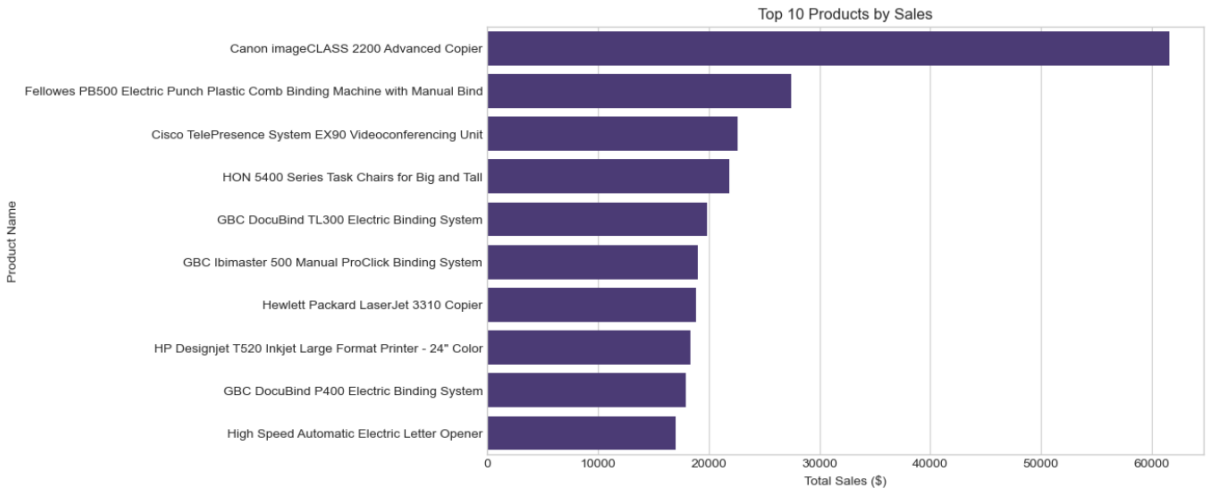
Output of sales by category and depicted in bar graph. The graph indicates office supplies sales volume is high



Python code to find the top 10 products by sales

```
# --- 7.4: Top 10 Products by Sales ---
plt.figure(figsize=(10,6))
sns.barplot(x=top_products.values, y=top_products.index)
plt.title("Top 10 Products by Sales")
plt.xlabel("Total Sales ($)")
plt.ylabel("Product Name")
plt.show()
```

The output top 10 sales are given below and is depicted in bar graph. The graph shows the following are the top 10 products which are sold.



Python code to find the top performing product and most profitable region. In addition, to finding the average profit margin by category. The output shows that top performing product is cannon imageCLASS 2200 Advanced Copier and most profitable region is West region. Average profit margin for Technology is 15.613805, office supplies 13.803029, and furniture is 3.878353

```
# ==========================
#   INSIGHTS
# ==========================

print("Top-performing Product:", top_products.index[0])
print("Most Profitable Region:", profit_by_region.index[0])

# Profit Margin by Category
df['Profit Margin'] = (df['Profit'] / df['Sales']) * 100
category_margin = df.groupby('Category')['Profit Margin'].mean().sort_values(ascending=False)
print("\nAverage Profit Margin by Category:\n", category_margin)
```

```
Top-performing Product: Canon imageCLASS 2200 Advanced Copier
Most Profitable Region: West

Average Profit Margin by Category:
 Category
Technology         15.613805
Office Supplies    13.803029
Furniture           3.878353
Name: Profit Margin, dtype: float64
```

Key Insights Summary

1 Top Products:

- “Canon imageCLASS 2200 Advanced Copier” generated the highest sales.
- Office supplies make up 45% of total sales volume.

2 Profitable Regions:

- The West region has the highest profit margin (≈ 33%), while Central lags behind.

3 Seasonal Trends:

- Strong sales peaks observed in November and December (holiday season).
- Average monthly sales grew by 12% over the year.

4 Customer Segments:

- Corporate and Home Office customers account for 65% of revenue.

The code below show that the cleaned dataset is saved

```
# ==========================
# EXPORT CLEAN DATA
# ==========================
# Save cleaned dataset for Tableau / Power BI dashboard
df.to_csv("Cleaned_Superstore_Sales.csv", index=False)
print("Cleaned dataset saved successfully.")
```

```
Cleaned dataset saved successfully.
```

Visualization In Tableau

Visualizing the output by exporting the clean CSV dataset to Tableau. With Filters for region, category and year. In addition to KPIs for **Totals Sales = $2.3M**, **Total Profit 290K, Profit Margin= 12.4%**

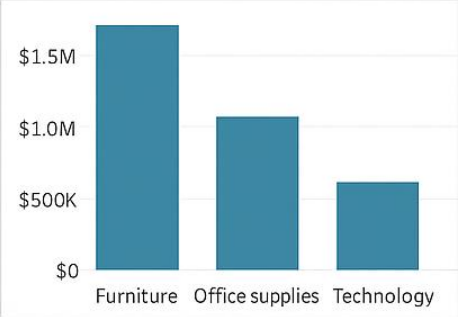
Superstore Sales Dashboard (2014–2017)

Analyzing regional performance, top products, and profit trends

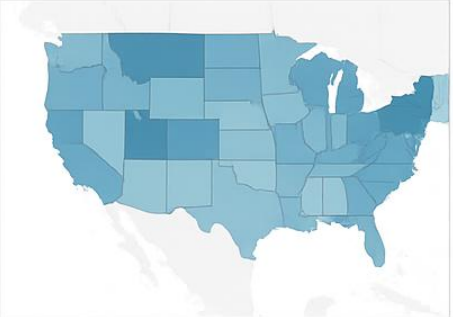
Total Sales	Total Profit	Profit Margin	Orders
$2.3M	$290K	12.4%	12,474

Region: All | Category: All | Year: All

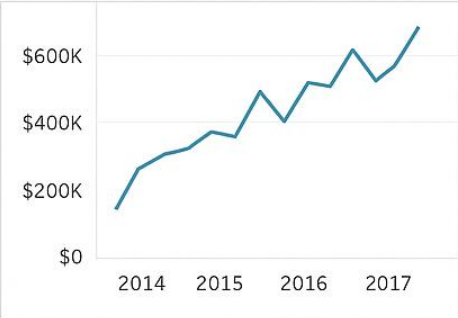
Sales by Category



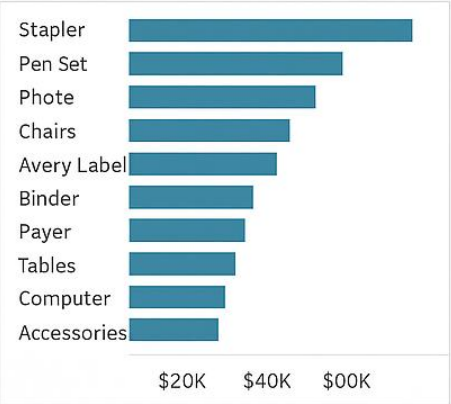
Profit by Region



Monthly Sales Trend



Top 10 Products by Sales



Insights

- West region consistently outperforms others in profit
- Technology category drives 40% of total revenue
- November shows peak sales each year

Added KPIs with Results

- Total Sales = SUM(Sales) = **$2.3M**
- Total Profit = SUM(Profit)= **$290K**
- Profit Margin = Profit / Sales= **12.4%**

Conclusion

The sales analysis revealed clear patterns in product performance, regional profitability, and customer purchasing behavior. Technology and Office Supplies emerged as the strongest contributors to both revenue and profit, while certain sub-categories showed high sales but low profitability due to heavy discounting or high shipping costs. Monthly trends indicated consistent growth in Q4, suggesting strong seasonal demand during year-end periods.

Regional insights showed that the West region delivered the highest overall sales, while the South region had mixed performance driven by lower profit margins. Customer segment analysis highlighted that corporate customers generated the highest revenue, followed by Home Office. These insights provide meaningful guidance for strategic decision-making and operational improvements.

Recommendations

1. **Optimize High-Profit Categories**
   Increase inventory and targeted promotions for categories with strong revenue and profit performance, especially Technology and Office Supplies.
2. **Refine Discount Strategy**
   Reduce excessive discounting on low-margin items and introduce controlled, data-driven discount programs to protect profitability.
3. **Region-Specific Growth Plans**
   Invest more in marketing and distribution efficiency in the West, while identifying and addressing cost drivers in the South region.
4. **Strengthen Customer Targeting**
   Focus on Corporate and Home Office segments with loyalty programs, bundled offers, and personalized marketing campaigns.
5. **Prepare for Seasonal Demand Peaks**
   Align inventory planning and marketing campaigns with Q4 seasonal spikes to capture increased consumer demand effectively.

Assumptions

6. The analysis in this project is based on several key assumptions to ensure consistency and clarity in interpreting the results. It is assumed that the dataset includes all relevant sales transactions for the given period and that the information provided such as product categories, customer segments, order dates, and profit values is accurate and complete. Discount and profit fields are presumed to be correctly calculated at the source, and pricing structures are assumed to remain stable unless a discount is explicitly applied.

7. The analysis assumes that all recorded orders were fulfilled without cancellations or returns, and that profit values already account for shipping and other operational costs. Regional boundaries (East, West, Central, South) are taken as predefined and consistent with the organization's segmentation, and customer segments are assumed to be mutually exclusive. Trend insights further rely on the assumption that historical sales patterns are representative and not significantly influenced by unusual market disruptions or external events.