

[Super Store Sales Analysis Report]

Made by: Sufiyan Ahmed Khan



July 14, 2024

Internee.Pk

Contents

[Super Store Sales Analysis Documentation 2](#_Toc171814424)

[Introduction 2](#_Toc171814425)

[Dependencies 2](#_Toc171814426)

[Data Loading 2](#_Toc171814427)

[Data Exploration 2](#_Toc171814428)

[1. First and Last Five Rows of the Dataset 2](#_Toc171814429)

[2. Shape of the Dataset 2](#_Toc171814430)

[3. Columns in the Dataset 2](#_Toc171814431)

[4. Summary Information 3](#_Toc171814432)

[5. Missing Values Check 3](#_Toc171814433)

[6. Descriptive Statistics 3](#_Toc171814434)

[Date Manipulation 3](#_Toc171814435)

[Sales Trend Analysis 3](#_Toc171814436)

[Product Sales Analysis 4](#_Toc171814437)

[Shipping Mode Analysis 5](#_Toc171814438)

[Category and Sub-Category Profit Analysis 5](#_Toc171814439)

[Suggestions for Improvement 6](#_Toc171814440)

[Suggestions for Improvement in Analysis 6](#_Toc171814441)

# Super Store Sales Analysis Documentation

# Introduction

This analysis aims to understand the sales trends and performance of different products, categories, and shipping modes in the SuperStore dataset. The analysis includes data manipulation, visualization, and deriving insights to suggest strategies for improving sales and profitability.

# Dependencies

The following libraries are used for data manipulation and visualization:

* pandas
* matplotlib
* seaborn

# Data Loading

The dataset is loaded from an Excel file named superstore\_sales.xlsx.

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

df = pd.read\_excel('superstore\_sales.xlsx')

# Data Exploration

## First and Last Five Rows of the Dataset

df.head()

df.tail()

## Shape of the Dataset

df.shape

## Columns in the Dataset

df.columns

## Summary Information

df.info()

## Missing Values Check

df.isna().sum()

## Descriptive Statistics

df.describe()

# Date Manipulation

Extracting the minimum and maximum order dates:

df['order\_date'].min()

df['order\_date'].max()

# Creating a month\_year column from the order\_date:

df['month\_year'] = df['order\_date'].apply(lambda x: x.strftime('%Y-%m'))

# Sales Trend Analysis

# Grouping sales by month\_year and plotting the sales trend:

df\_trend = df.groupby('month\_year').sum()['sales'].reset\_index()

plt.figure(figsize=(16, 5))

plt.plot(df\_trend['month\_year'], df\_trend['sales'], color='#b80045')

plt.xticks(rotation='vertical', size=8)

plt.show()

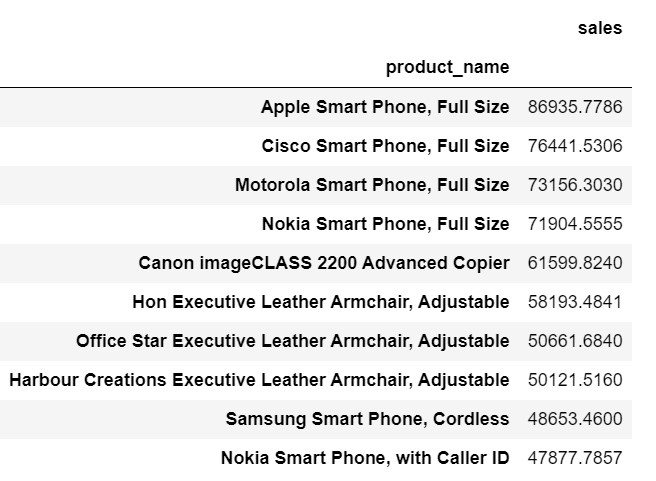


# Product Sales Analysis

1. **Top 10 Products by Sales**

prod\_sales = pd.DataFrame(df.groupby('product\_name').sum()['sales'])

prod\_sales = prod\_sales.sort\_values('sales', ascending=False)

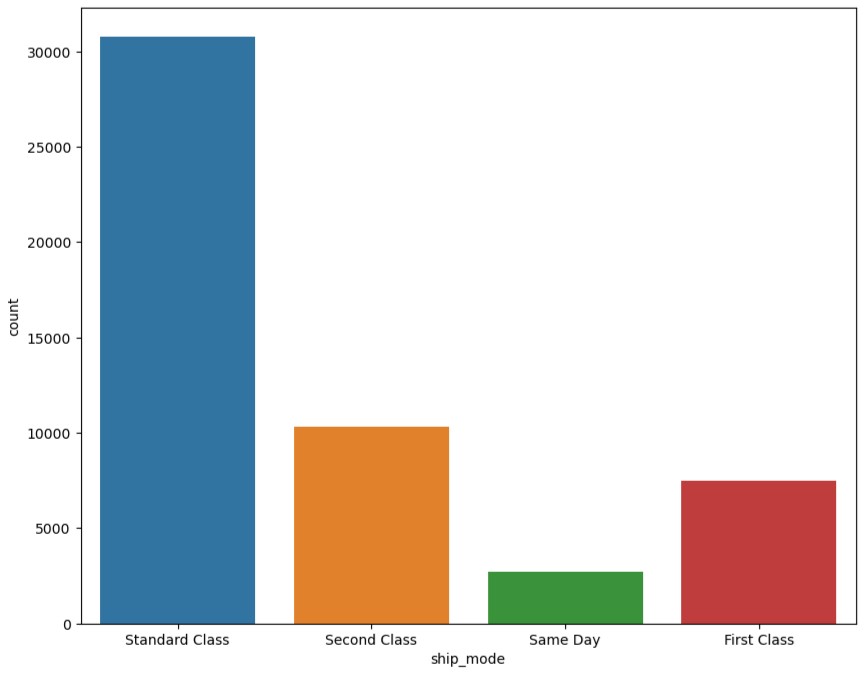
prod\_sales.head(10)  
  


1. **Top 10 Most Sold Products**

prod\_most\_sell = pd.DataFrame(df.groupby('product\_name').sum()['quantity'])

prod\_most\_sell = prod\_most\_sell.sort\_values('quantity', ascending=False)

prod\_most\_sell.head(10)



# Shipping Mode Analysis

Plotting the count of each shipping mode:

plt.figure(figsize=(10, 8))

sns.countplot(df['ship\_mode'])

plt.show()

# Category and Sub-Category Profit Analysis

# Grouping profit by category and sub-category:

cat\_subcat = pd.DataFrame(df.groupby(['category', 'sub\_category']).sum()['profit'])

cat\_subcat = cat\_subcat.sort\_values(['category', 'profit'], ascending=False)

cat\_subcat

# Suggestions for Improvement

1. **Focus on High-Performing Categories and Regions**
   * Invest in marketing and inventory for categories and regions with high sales to maximize revenue.
2. **Address Low-Performing Categories and Regions**
   * Investigate reasons for low sales and implement strategies to improve performance in these areas.
3. **Optimize Shipping Methods**
   * Analyze shipping costs and delivery times to optimize shipping methods for better customer satisfaction and reduced costs.

# Suggestions for Improvement in Analysis

1. **Handle Missing Values**
   * Address any missing values through appropriate imputation methods or data cleaning techniques.
2. **Data Normalization**
   * Normalize numerical data to ensure fair comparisons between different scales.
3. **Additional Visualizations**
   * Incorporate more visualizations such as bar charts, heatmaps, and box plots for a deeper understanding of the data.
4. **Advanced Analysis**
   * Consider advanced analysis techniques such as clustering, segmentation, and predictive modeling to uncover more insights.
5. **Interactive Dashboards**
   * Create interactive dashboards using tools like Tableau or Power BI for better data exploration and presentation.