SALES DATA ANALYSIS PROJECT REPORT

Project Title:

Analyzing Sales Data

Tools Used:

• Jupyter Notebook

Technologies:

• Business Intelligence

Domain:

• E-Commerce

1. Problem Definition

Objective:
Analyze Amazon (Superstore) sales data to understand sales trends, identify top-performing products, and optimize inventory and marketing strategies.

2. Data Collection

Dataset Used:
Sample - Superstore.csv (obtained from Kaggle/Amazon dataset)

Key Features:
- Order ID, Order Date, Ship Date
- Product Name, Category, Sub-Category
- Sales, Quantity, Discount, Profit
- Customer Segment, City, State, Region, Country

3. Data Exploration

• Dataset loaded using pandas.

• Checked data types, null values, and basic statistical description.

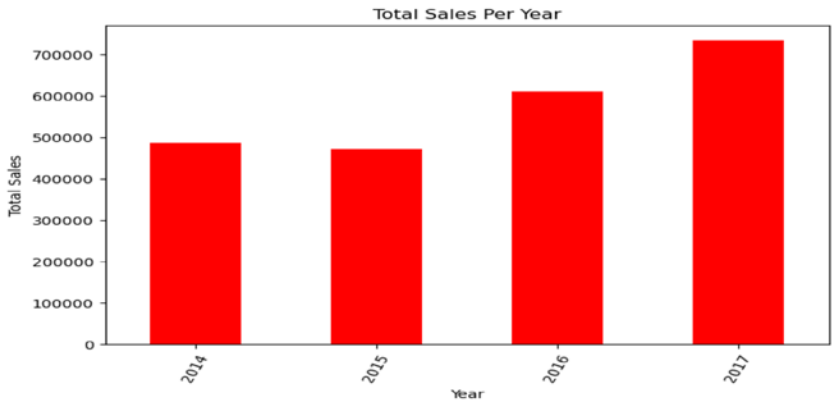
- No missing values found; data was clean and ready for analysis.

4. Visual Data Analysis

1.Sales Over Time (Yearly)

- Analyzed yearly sales performance using bar chart.
- Observed growth and decline patterns year-wise.

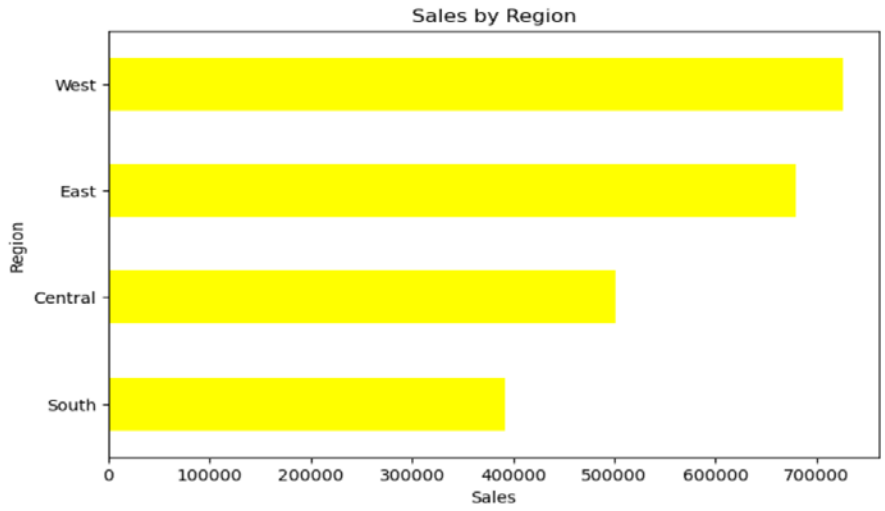
```
#visual analysis

# Sales Over Time (Yearly)
sales_by_year = df.groupby('Order Year')['Sales'].sum()
plt.figure(figsize=(8,5))
sales_by_year.plot(kind='bar', color='red')
plt.title("Total Sales Per Year")
plt.xlabel("Year")
plt.ylabel("Total Sales")
plt.xticks(rotation=65)
plt.show()
```



2.Region-wise Sales

- Visualized sales distribution across different regions.
- Helpful in identifying strong and weak regions for sales.

```
#Region-wise Sales
region_sales = df.groupby('Region')['Sales'].sum().sort_values()
plt.figure(figsize=(8,5))
region_sales.plot(kind='barh', color='yellow')
plt.title("Sales by Region")
plt.xlabel("Sales")
plt.show()
```



3.Sales vs Profit Scatter Plot (with Regression Line)

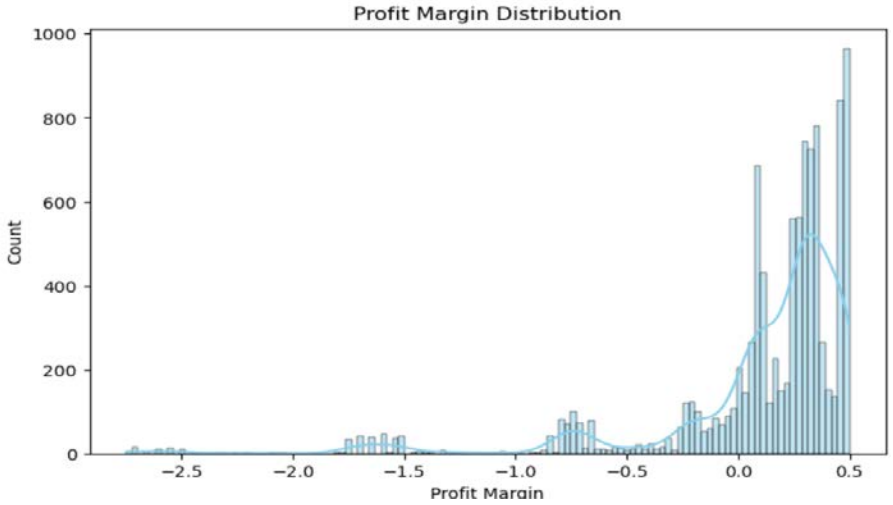
- Visualized linear relationship between sales and profit.
- Observed positive correlation visually.

```
#Sales vs Profit Scatter Plot (with Regression Line)
plt.figure(figsize=(8,5))
sns.regplot(x='Sales', y='Profit', data=df, scatter_kws={'alpha':0.3}, line_kws={"color":"red"})
plt.title("Sales vs Profit")
plt.xlabel("Sales")
plt.ylabel("Profit")
plt.grid(True)
plt.show()
```

4.Profit Margin Distribution

- Calculated and plotted profit margin distribution.
- Identified majority of transactions cluster around certain profit margins.

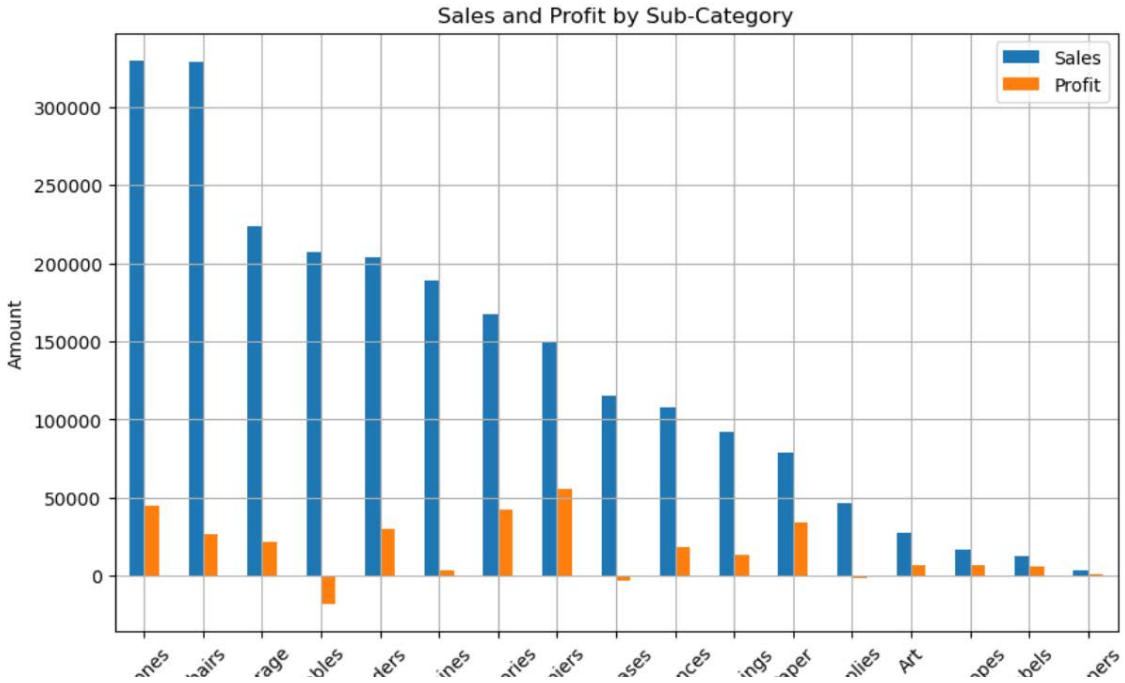
```
#Profit Margin Distribution
df['Profit Margin'] = df['Profit'] / df['Sales']
plt.figure(figsize=(8,5))
sns.histplot(df['Profit Margin'], kde=True, color='skyblue')
plt.title("Profit Margin Distribution")
plt.xlabel("Profit Margin")
plt.show()
```



5.Sales and Profit by Sub-Category

- Combined sales and profit in a bar chart by sub-category.
- Identified which sub-categories perform best in both metrics.

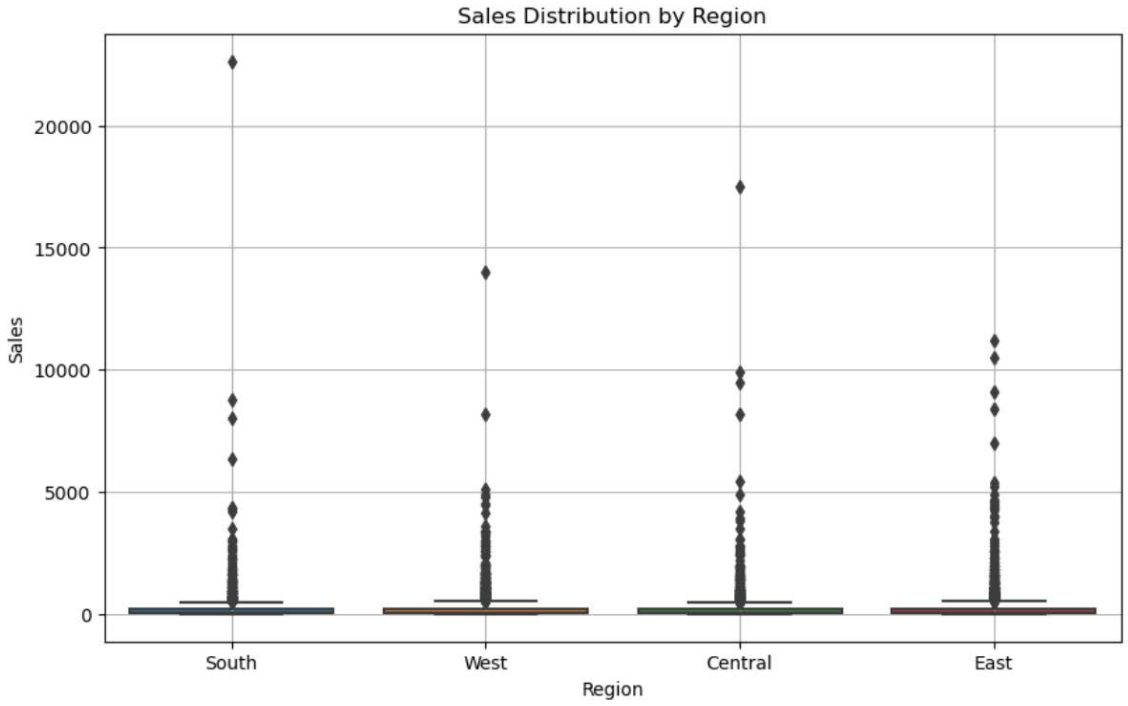
```
#Top Sub-Category Sales and Profit Combined Barplot
subcat_sales = df.groupby('Sub-Category')['Sales'].sum()
subcat_profit = df.groupby('Sub-Category')['Profit'].sum()
subcat_df = pd.DataFrame({'Sales': subcat_sales, 'Profit': subcat_profit})
subcat_df.sort_values('Sales', ascending=False).plot(kind='bar', figsize=(10,6))
plt.title("Sales and Profit by Sub-Category")
plt.ylabel("Amount")
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



6.Sales Distribution by Region (Boxplot)

- Analyzed how sales are distributed across regions.
- Detected outliers and spread across regions.

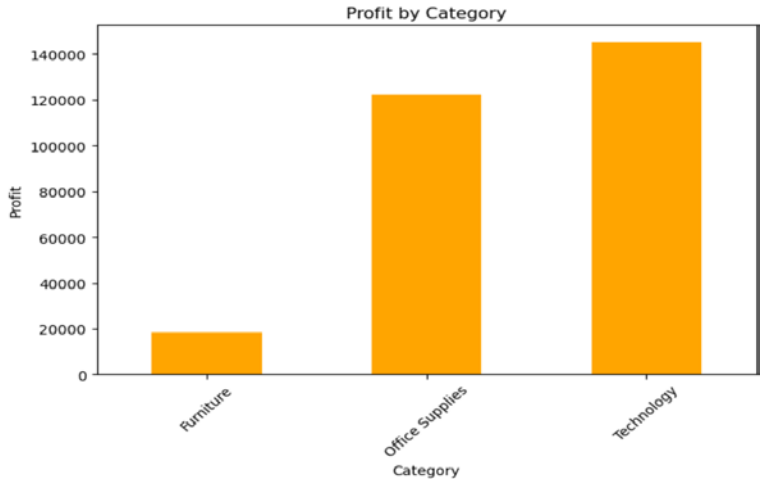
```
#-Sales Distribution Boxplot Across Regions
plt.figure(figsize=(10,6))
sns.boxplot(x='Region', y='Sales', data=df)
plt.title("Sales Distribution by Region")
plt.ylabel("Sales")
plt.grid(True)
plt.show()
```



7.Category-wise Profit

- Compared profit generated by each category.
- Identified which product categories are most profitable.

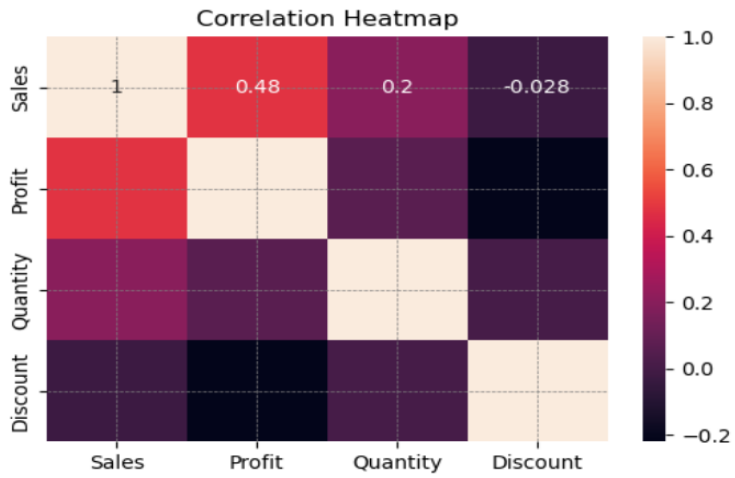
```
#  Category-wise Profit
category_profit = df.groupby('Category')['Profit'].sum()
plt.figure(figsize=(8,5))
category_profit.plot(kind='bar', color='orange')
plt.title("Profit by Category")
plt.ylabel("Profit")
plt.xticks(rotation=45)
plt.show()
```



8.Correlation Heatmap

- Analyzed correlation between Sales, Profit, Quantity, and Discount.
- Found that Sales and Profit have positive correlation.

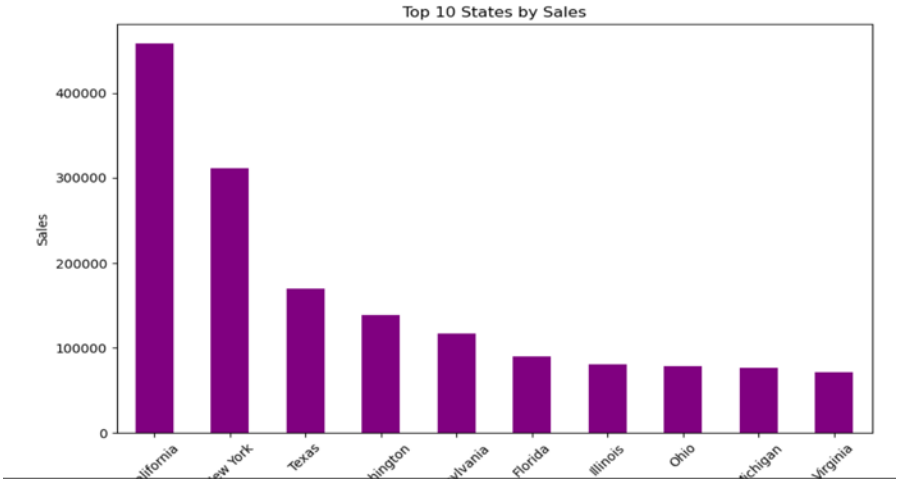
```
[35]: # Correlation Heatmap
plt.figure(figsize=(6,4))
sns.heatmap(df[['Sales', 'Profit', 'Quantity', 'Discount']].corr(), annot=True)
plt.title("Correlation Heatmap")
plt.grid(color='gray', linestyle='--', linewidth=0.5)
plt.show()
```



9.Top 10 States by Sales

- Identified top performing states by total sales.
- Useful for regional marketing and operations.

```
#  Top 10 States by Sales
state_sales = df.groupby('State')['Sales'].sum().sort_values(ascending=False).head(10)
plt.figure(figsize=(10,6))
state_sales.plot(kind='bar', color='purple')
plt.title("Top 10 States by Sales")
plt.ylabel("Sales")
plt.xticks(rotation=45)
plt.show()
```



5. Business Questions & Insights

1.Total Sales, Profit, and Orders

- Total Sales: `df['Sales'].sum()`
- Total Profit: `df['Profit'].sum()`
- Total Orders: `len(df)`

2. Average Order Value

average_order_value = df['Sales'].mean()

3. Top 5 Products by Sales

product_sales=df.groupby('ProductName')['Sales'].sum().sort_values(ascending= False).head(5)

4.**Top 5 Sub-Categories by Profit**

subcategory_profit=df.groupby('SubCategory')['Profit'].sum().sort_values(ascending=False).head(5)

5.Monthly Sales Trend

- Line plot showing seasonal/monthly trends.

6.Average Shipping Delay

avg_ship_delay = df['Ship Delay'].mean()

7. Correlation between Sales & Profit

correlation = df['Sales'].corr(df['Profit'])

6.Insights

- **Sales Trends:** Sales vary yearly; some growth and some decline.
- **Region Performance:** West and East regions lead in sales.
- **Category Profit:** Technology category gives highest profits.
- **Shipping:** Delays present; needs logistics improvement.
- **Sales vs Profit:** Positive but weak correlation; heavy discounts reduce profits.
- **Top States:** California, New York, Texas are top-selling states.
- **Average Order Value:** Moderate; upselling can improve.
- **Top Products/Sub-categories:** Copiers and Phones most profitable.
- **Seasonality:** Monthly sales fluctuate; seasonal marketing needed.
- **Discount Impact:** High discounts hurt profits.
- **Shipping Delay:** Outliers suggest logistics optimization required.

7. Technologies Used

Python Libraries: pandas, numpy, matplotlib, seaborn
Environment: Jupyter Notebook, VS Code