



# **AI MSE-1 Assignment**

**(AI101B)**

**MCA 2<sup>nd</sup> SEM (II-D)**



**Submitted by:**  
**supervision of:**

**Tannoo Shukla (202410116100220)**

**Tannu Modanwal (202410116100221)**

**Trapti Rajput (202410116100224)**

**Tanya Singh (202410116100222)**

**Under the**

**Ms. KOMAL SALGOTRA**

# **TITLE: SIMPLE SALES DATA VISUALIZATION**

## **INTRODUCTION:**

Sales data visualization is a crucial aspect of business intelligence, helping organizations analyze and interpret their sales performance effectively. By transforming raw sales data into graphical representations, businesses can gain insights into market trends, customer behavior, and product performance. Sales visualization simplifies complex numerical data, making it easier for stakeholders to identify patterns and make informed decisions.

We will utilize various data visualization techniques, such as bar charts, line graphs, and pie charts, to represent sales performance in an intuitive manner. Furthermore, we will analyze key sales metrics, including revenue growth, sales distribution across regions, and product sales trends. This project aims to provide actionable insights that can drive business strategy and improve overall sales performance.

In this analysis, we concentrate on two main features of sales data visualization:

- **Sales Trend Analysis:** Using historical sales data to track revenue patterns, seasonal fluctuations, and growth trends. Line graphs and area charts are commonly used for this purpose.
- **Comparative Analysis:** Examining product performance, regional sales, and customer preferences through bar charts, pie charts, and heatmaps. These visualizations help businesses optimize their sales strategies.

This research will provide valuable insights into how sales data visualization enhances decision-making, improves sales forecasting, and identifies business growth opportunities. By leveraging effective visualization techniques, businesses can gain a competitive edge in the market.

## **CODE:**

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

# Load sales data
file_path = "/content/sales_data.csv" # Update this path as needed
sales_data = pd.read_csv(file_path)

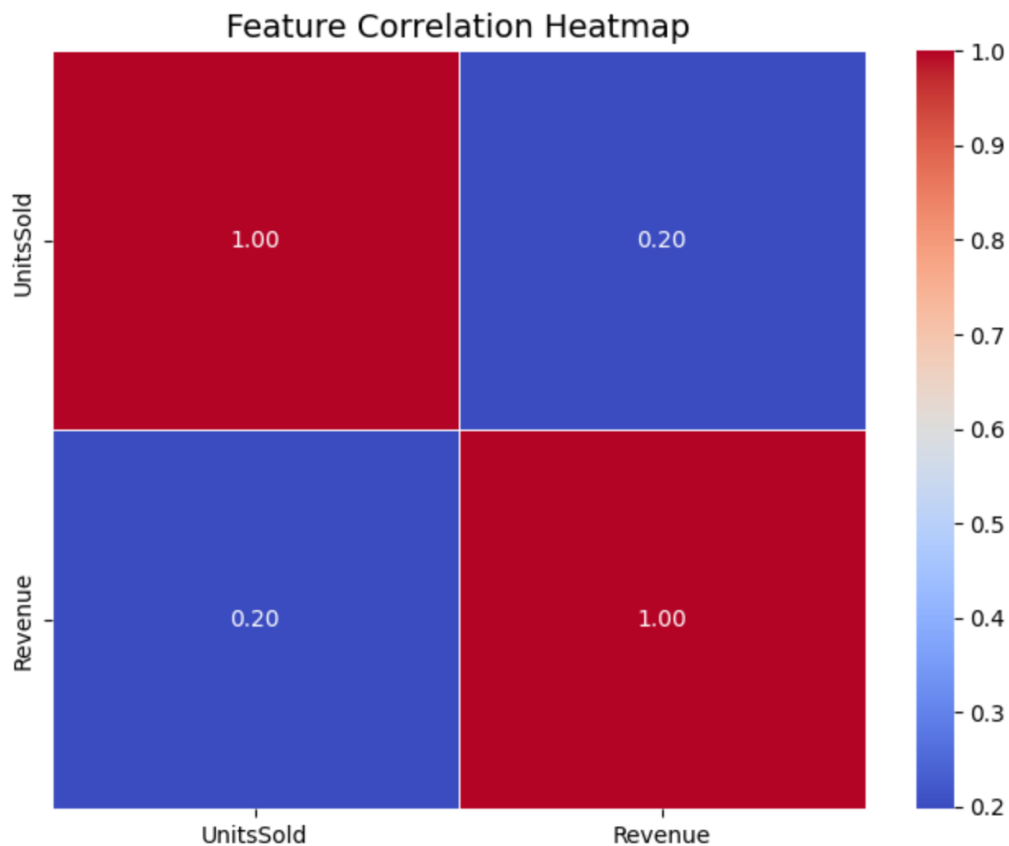
# Convert 'Date' column to datetime if it exists
if 'Date' in sales_data.columns:
    sales_data['Date'] = pd.to_datetime(sales_data['Date'])

# Correlation Heatmap (only for numeric columns)
plt.figure(figsize=(8, 6))

# Selecting only numeric columns
numeric_data = sales_data.select_dtypes(include=['number'])

if not numeric_data.empty: # Ensure at least one numeric column exists
    sns.heatmap(numeric_data.corr(), annot=True, cmap="coolwarm",
fmt=".2f", linewidths=0.5)
    plt.title("Feature Correlation Heatmap", fontsize=14)
    plt.show()
else:
    print("No numeric columns available for correlation heatmap.")
```

## SCREENSHOTS:



```
import pandas as pd
import matplotlib.pyplot as plt

# Sample Sales Data (You can replace it with a CSV file or actual dataset)
data = {
    'Month': ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun'],
    'Sales': [1000, 1500, 1200, 1700, 1600, 1800], # Monthly sales in dollars
    'Product_A': [400, 600, 500, 800, 750, 900], # Sales for Product A
    'Product_B': [600, 900, 700, 900, 850, 900], # Sales for Product B
    'Region': ['North', 'South', 'East', 'West', 'North', 'South']
}

# Convert dictionary to DataFrame
df = pd.DataFrame(data)

# 1. Line Chart - Monthly Sales Trend
```

```

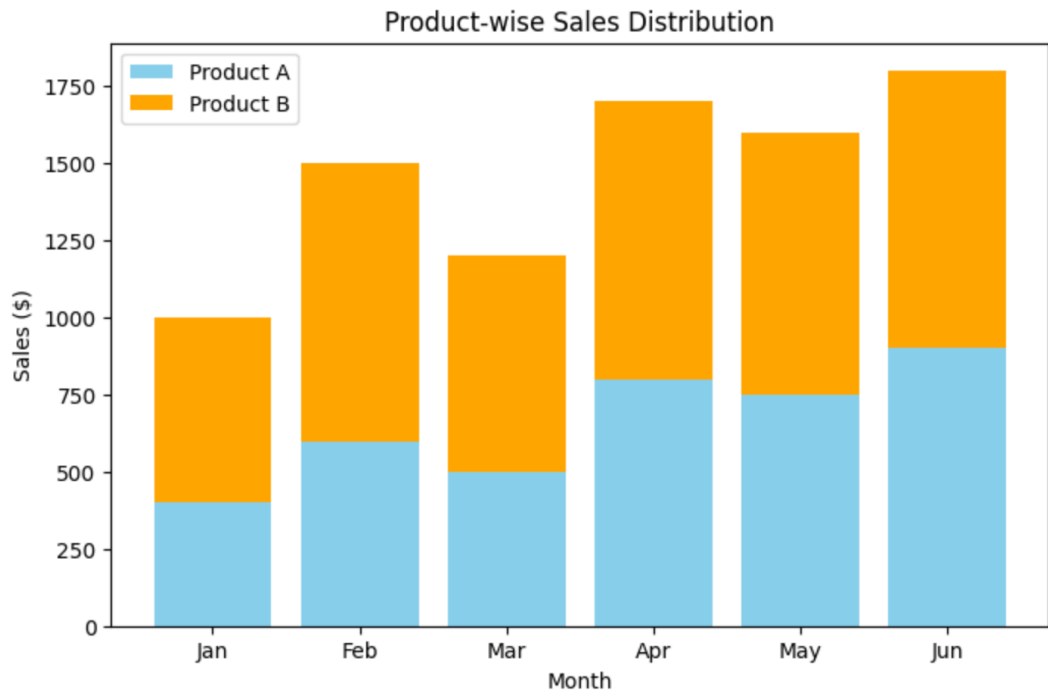
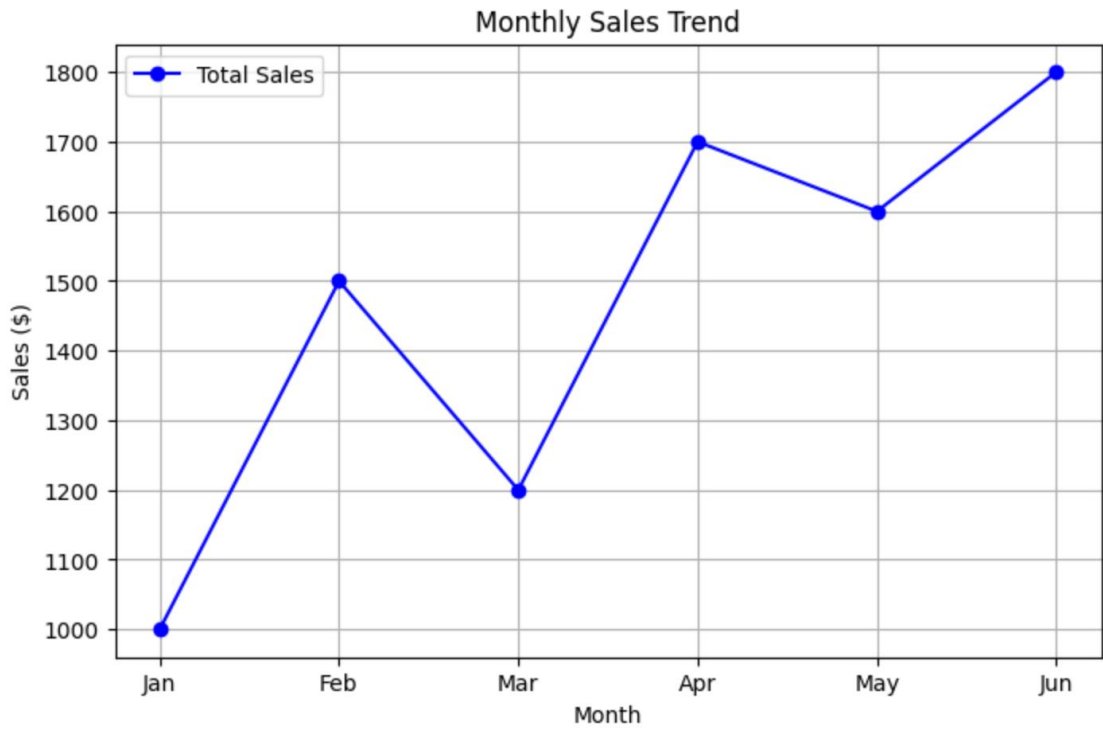
plt.figure(figsize=(8, 5))
plt.plot(df['Month'], df['Sales'], marker='o', linestyle='-',
color='b', label="Total Sales")
plt.xlabel('Month')
plt.ylabel('Sales ($)')
plt.title('Monthly Sales Trend')
plt.legend()
plt.grid(True)
plt.show()

# 2. Bar Chart - Sales by Product
plt.figure(figsize=(8, 5))
plt.bar(df['Month'], df['Product_A'], label='Product A',
color='skyblue')
plt.bar(df['Month'], df['Product_B'], bottom=df['Product_A'],
label='Product B', color='orange')
plt.xlabel('Month')
plt.ylabel('Sales ($)')
plt.title('Product-wise Sales Distribution')
plt.legend()
plt.show()

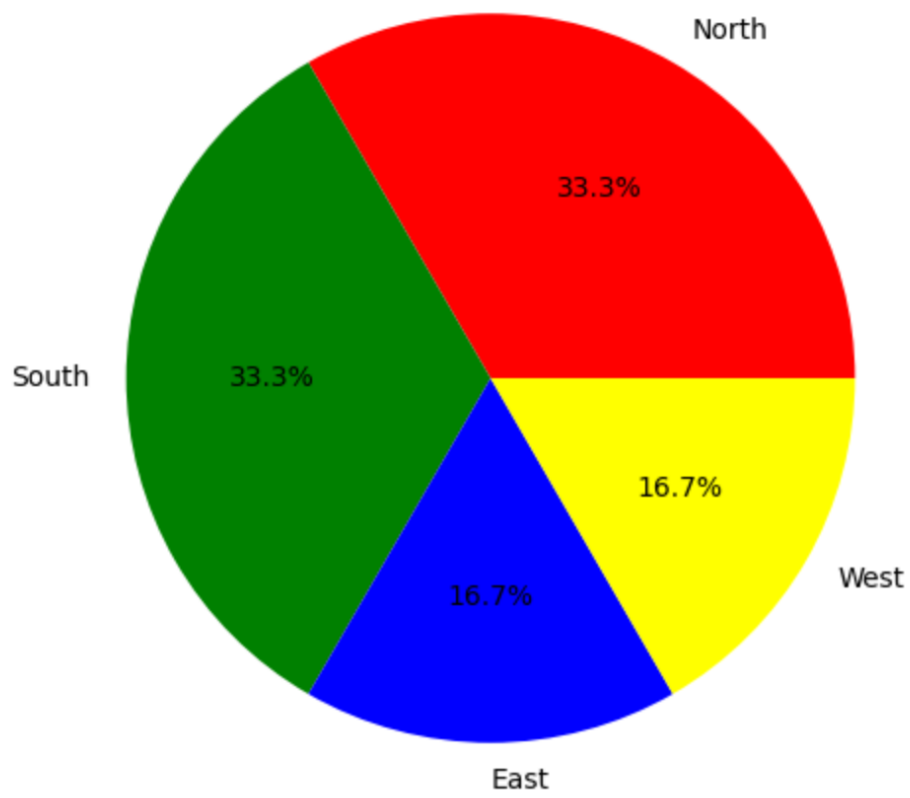
# 3. Pie Chart - Regional Sales Contribution
region_counts = df['Region'].value_counts()
plt.figure(figsize=(6, 6))
plt.pie(region_counts, labels=region_counts.index, autopct='%1.1f%%',
colors=['red', 'green', 'blue', 'yellow'])
plt.title('Sales Distribution by Region')
plt.show()

```

## **SCREENSHOTS:**



Sales Distribution by Region



# **METHODOLOGY:**

To effectively perform **sales data visualization**, we follow a structured methodology that includes **data collection, preprocessing, visualization, and analysis**. The following steps outline the approach used:

## Step 1: Data Collection and Preprocessing

- **Collect sales data** from sources like **CSV files, databases, or business reports**.
- The dataset should include **date, product category, region, and sales figures**.
- **Clean the data** by removing duplicates and handling missing values.
- Convert categorical values into appropriate numerical representations if necessary.

## Step 2: Data Analysis and Processing

- Compute **monthly sales trends, product-wise contributions, and regional performance**.
- Identify **top-selling products** and **seasonal patterns** in the sales data.
- Aggregate sales based on different categories for meaningful insights.

## Step 3: Data Visualization

After processing the data, **various visualization techniques** are used to represent sales trends effectively:

- **Sales Trend Analysis:**



- A **line chart** is plotted to show sales performance over time.
- Helps in identifying **seasonal trends** and **growth patterns**.
- **Product-wise Sales Comparison:**
  - A **bar chart** is created to compare sales across different products.
  - This helps in determining which products contribute the most revenue.
- **Regional Sales Distribution:**
  - A **pie chart** represents the percentage contribution of different regions.
  - Helps businesses focus on **high-demand areas**.

#### Step 4: User Interaction and Execution

- The user inputs a dataset, and the program processes it for **insights**.
- Sales figures are **visualized using Matplotlib and Seaborn** in Python.
- The **final output includes charts and graphs**, providing a clear sales overview.

#### Step 5: Business Insights & Decision Making

- Identify **peak sales periods** and optimize inventory management.
- Recognize **low-performing products** and take corrective measures.
- Plan targeted marketing strategies based on **regional sales trends**.