# CS23334-FUNDAMENTALS OF DATA SCIENCE

**DEVA DHARSHINI P(240701107)**

## 2.) PANDAS LIBRARY-BASIC CONCEPT

**Aim:**

To analyze and visualize sales data from an Excel dataset using Python libraries — *pandas*, *numpy*, *matplotlib*, and *seaborn* — by performing data cleaning, summarization, and generating insightful visualizations.

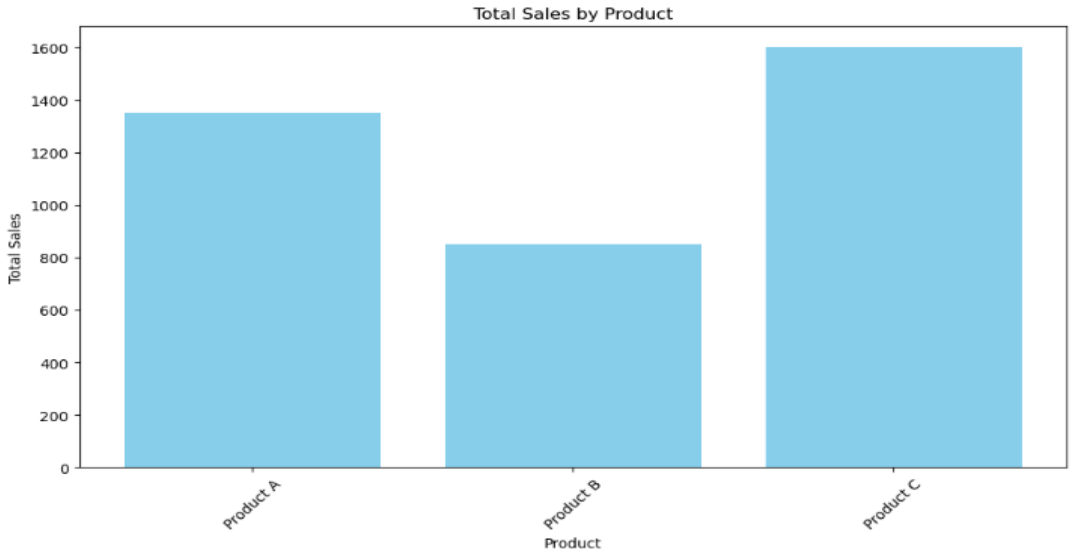
**Code:**

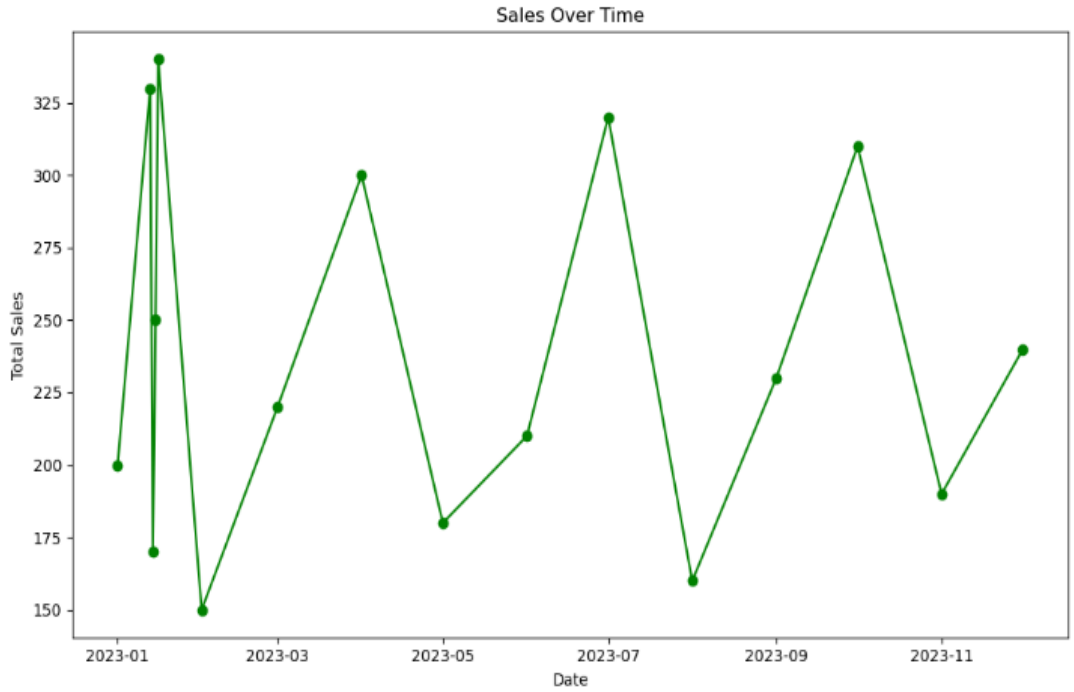
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

file_path = r"C:\Users\Deva Dharshini P\Downloads\sales_data.xlsx"

df = pd.read_excel(file_path)

print("First 5 rows of dataset:")
print(df.head())

print("\nMissing Values:")
print(df.isnull().sum())

if 'Sales' in df.columns:
    df['Sales'].fillna(df['Sales'].mean(), inplace=True)

for col in ['Product', 'Quantity', 'Region']:
    if col in df.columns:
        df.dropna(subset=[col], inplace=True)

print("\nSummary Statistics:")
print(df.describe())

if {'Product', 'Sales', 'Quantity'}.issubset(df.columns):
    product_summary = df.groupby('Product').agg({
        'Sales': 'sum',
        'Quantity': 'sum'
    }).reset_index()

    print("\nTotal Sales and Quantity by Product:")
    print(product_summary)

    plt.figure(figsize=(10, 6))
    plt.bar(product_summary['Product'], product_summary['Sales'], color='skyblue')
```

```
    plt.xlabel('Product')
    plt.ylabel('Total Sales')
    plt.title('Total Sales by Product')
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()

if 'Date' in df.columns:
    df['Date'] = pd.to_datetime(df['Date'], errors='coerce')
    sales_over_time = df.groupby('Date').agg({'Sales': 'sum'}).reset_index()

    plt.figure(figsize=(10, 6))
    plt.plot(sales_over_time['Date'], sales_over_time['Sales'], color='green', marker='o')
    plt.xlabel('Date')
    plt.ylabel('Total Sales')
    plt.title('Sales Over Time')
    plt.tight_layout()
    plt.show()

if {'Region', 'Product', 'Sales'}.issubset(df.columns):
    pivot_table = df.pivot_table(values='Sales', index='Region', columns='Product',
                                 aggfunc=np.sum, fill_value=0)
    print("\nSales by Region and Product (Pivot Table):")
    print(pivot_table)

correlation_matrix = df.select_dtypes(include=[np.number]).corr()
print("\nCorrelation Matrix:")
print(correlation_matrix)

plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Matrix Heatmap')
plt.show()
```

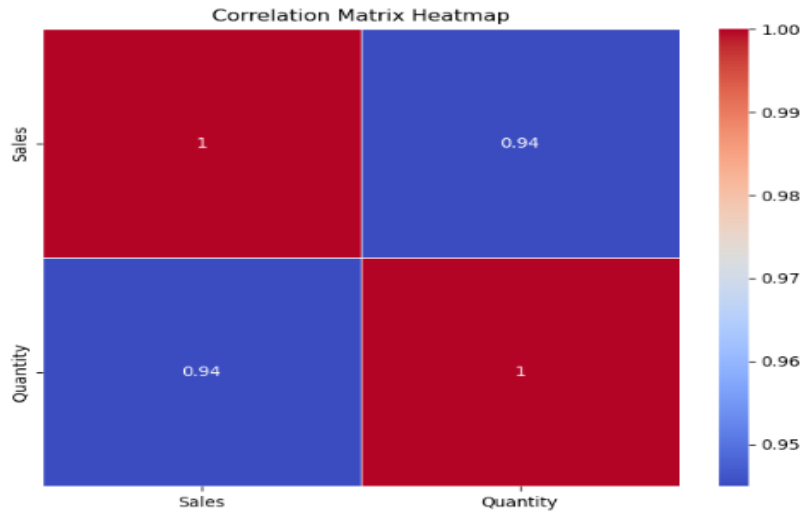
## OUTPUT:

```
First 5 rows of dataset:
                  Date     Product  Sales  Quantity Region
0  2023-01-01 00:00:00  Product A    200         4  North
1  2023-02-01 00:00:00  Product B    150         3  South
2  2023-03-01 00:00:00  Product A    220         5  North
3  2023-04-01 00:00:00  Product C    300         6   East
4  2023-05-01 00:00:00  Product B    180         4   West

Missing Values:
Date        0
Product     0
Sales       0
Quantity    0
Region      0
dtype: int64

Summary Statistics:
            Sales    Quantity
count   16.000000  16.000000
mean   237.500000   5.375000
std     64.031242   1.746425
min    150.000000   3.000000
25%    187.500000   4.000000
50%    225.000000   5.500000
75%    302.500000   7.000000
max    340.000000   8.000000

Total Sales and Quantity by Product:
      Product  Sales  Quantity
0  Product A   1350        33
1  Product B    850        17
2  Product C   1600        36
```




```
Sales by Region and Product (Pivot Table):
Product  Product A  Product B  Product C
Region
East             0          0       1600
North         1350          0          0
South            0        480          0
West             0        370          0

Correlation Matrix:
              Sales  Quantity
Sales     1.000000  0.944922
Quantity  0.944922  1.000000
```




**Result:**

The sales dataset was successfully analyzed and visualized using Python.
Missing values were handled, and graphs showing total sales by product, sales over time, and a correlation heatmap were displayed correctly.