CS-23334-FUNDAMENTALS OF DATA SCIENCE

ABENANTHAN P 240701005

EXPERIMENT 2

Upload and Analyze the data set given in csv format and perform data preprocessing and visualization.

Visualize the following:

1. Sales over the product
2. Sales over time
3. Display the correlation matrix

AIM:

To analyze the given data and perform preprocessing and visualize the data as Bar Plot , Line Plot , Pivot Table and a Correlation Matrix

Algorithm:

Step 1: Data Loading and Preprocessing

Step 2: Data Visualization (Bar Plot, Line Plot, Pivot Table, Correlation Matrix)

Step 3: Interpretation and Reporting

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

# STEP 1: Upload the file manually
from google.colab import files
uploaded = files.upload()

# STEP 2: Load the uploaded file into a pandas DataFrame
# (Make sure the file name matches exactly what you upload, e.g.,
'Sales_Data.csv')
df = pd.read_csv('Sales_Data.csv')

# STEP 3: Display the first few rows
print(df.head())

# STEP 4: Check for missing values
print("\nMissing Values:\n", df.isnull().sum())

# STEP 5: Handle missing data
df['Sales'] = pd.to_numeric(df['Sales'], errors='coerce')
df['Sales'].fillna(df['Sales'].mean(), inplace=True)
df.dropna(subset=['Product', 'Quantity', 'Region'], inplace=True)

# STEP 6: Summary statistics
print("\nSummary Statistics:\n", df.describe())

# STEP 7: Group by product and calculate total sales and quantity
product_summary = df.groupby('Product').agg({
    'Sales': 'sum',
    'Quantity': 'sum'
}).reset_index()
print("\nProduct Summary:\n", product_summary)

# STEP 8: Bar plot of total sales by product
plt.figure(figsize=(10, 6))
plt.bar(product_summary['Product'], product_summary['Sales'])
plt.xlabel('Product')
plt.ylabel('Total Sales')
plt.title('Total Sales by Product')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# STEP 9: Line plot of sales over time
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'])
plt.xlabel('Date')
plt.ylabel('Total Sales')
plt.title('Sales Over Time')
plt.tight_layout()
plt.show()
```

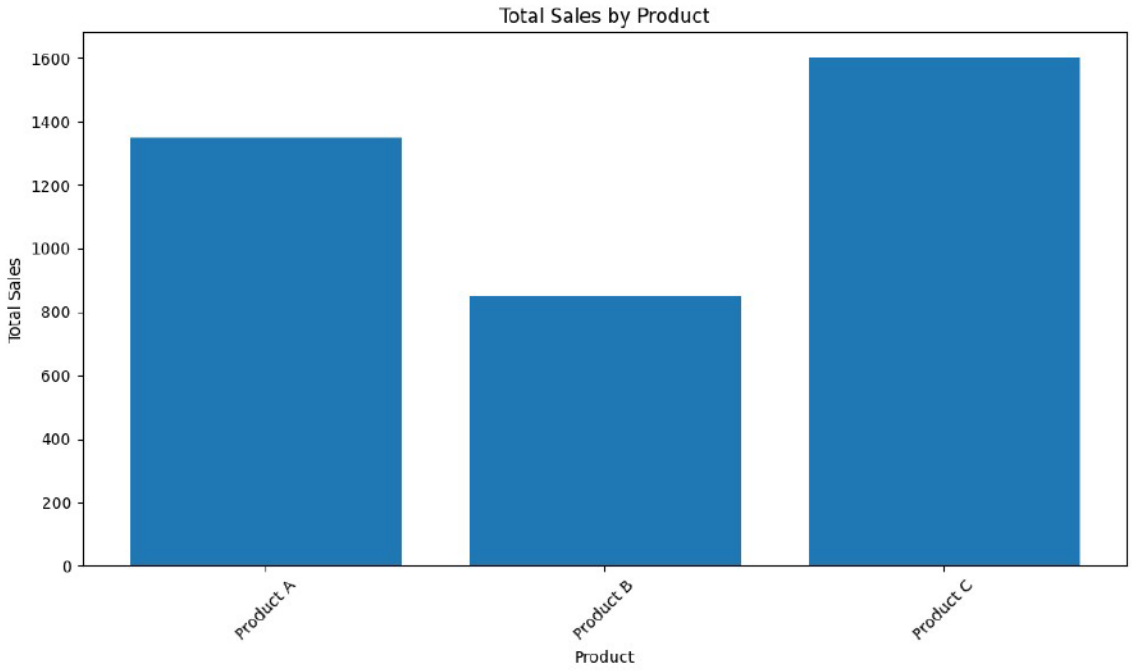
```
# STEP 10: Pivot table - Sales by Region and Product
pivot_table = df.pivot_table(values='Sales', index='Region',
columns='Product', aggfunc=np.sum, fill_value=0)
print("\nPivot Table:\n", pivot_table)

# STEP 11: Correlation matrix
correlation_matrix = df.select_dtypes(include=[np.number]).corr()
print("\nCorrelation Matrix:\n", correlation_matrix)

# STEP 12: Heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix')
plt.tight_layout()
plt.show()
```

Output:

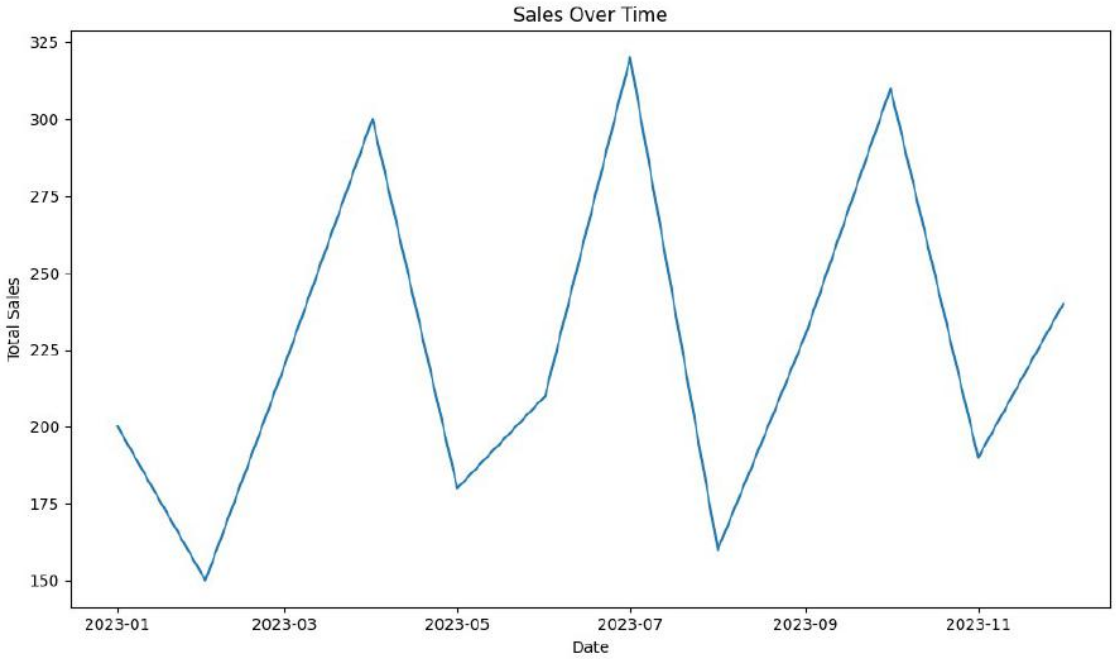
```
<IPython.core.display.HTML object>

Saving Sales_Data.csv to Sales_Data (1).csv
         Date    Product  Sales  Quantity Region
0  01-01-2023  Product A    200         4  North
1  02-01-2023  Product B    150         3  South
2  03-01-2023  Product A    220         5  North
3  04-01-2023  Product C    300         6   East
4  05-01-2023  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

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

Bar Plot:



Line Plot:

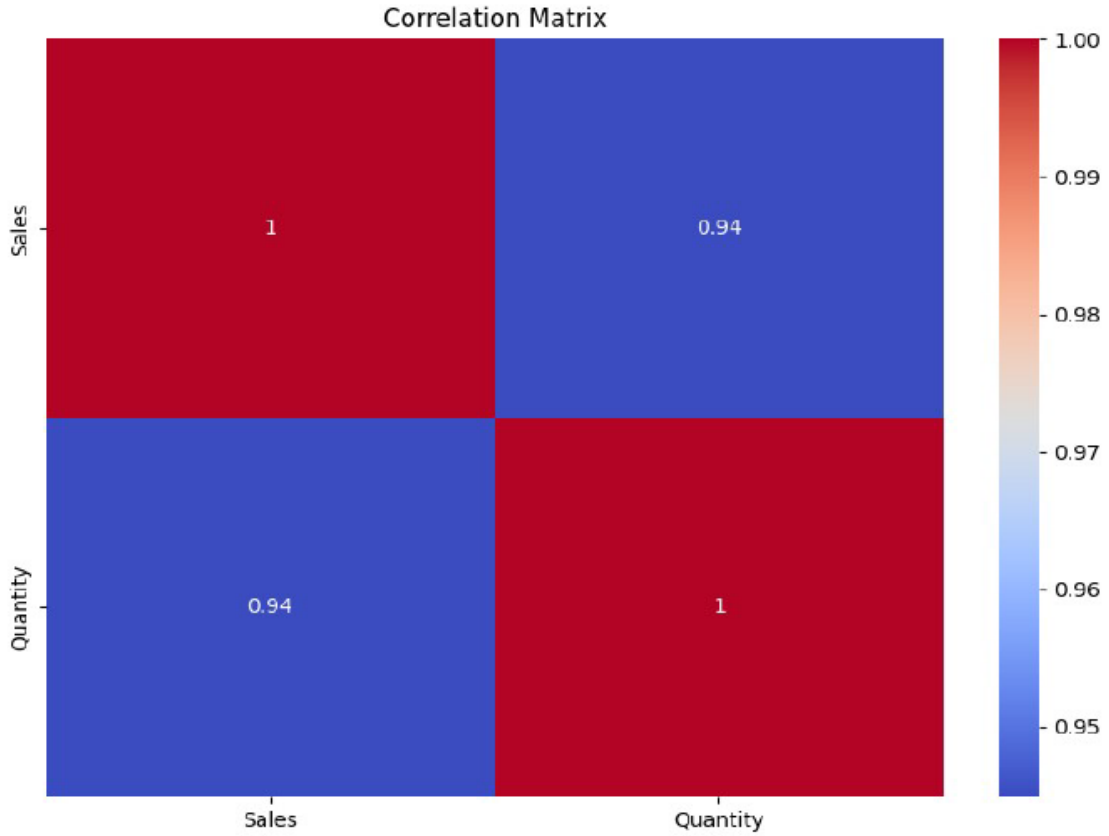


Pivot Table:

```
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:

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

/tmp/ipython-input-217718739.py:57: FutureWarning: The provided
callable <function sum at 0x7a4182919620> is currently using
DataFrameGroupBy.sum. In a future version of pandas, the provided
callable will be used directly. To keep current behavior pass the
string "sum" instead.
  pivot table = df.pivot table(values='Sales', index='Region',
columns='Product', aggfunc=np.sum, fill_value=0)
```



Result:

Thus the given dataset was preprocessed and visualized using a python code.