Pandas library -Basic Concept

Aim:

To write the Python program to understand and perform basic data analysis operations using the Pandas library on the given dataset.

Algorithm:

1. Load the dataset using Pandas and preview its contents.
2. Check for missing values and handle them using appropriate methods.
3. Generate summary statistics to understand the data distribution.
4. Group the data based on a categorical column to summarize key metrics.
5. Prepare the grouped data for visualization.
6. Create a bar chart to represent total sales by product.

Program:

```
[4]: df.isnull().sum()

[4]: Date        0
     Product     0
     Sales       0
     Quantity    0
     Region      0
     dtype: int64

[5]: df['Sales'].fillna(df['Sales'].mean(), inplace=True)
     df.dropna(subset=['Product', 'Quantity', 'Region'], inplace=True)
     df.describe()
```
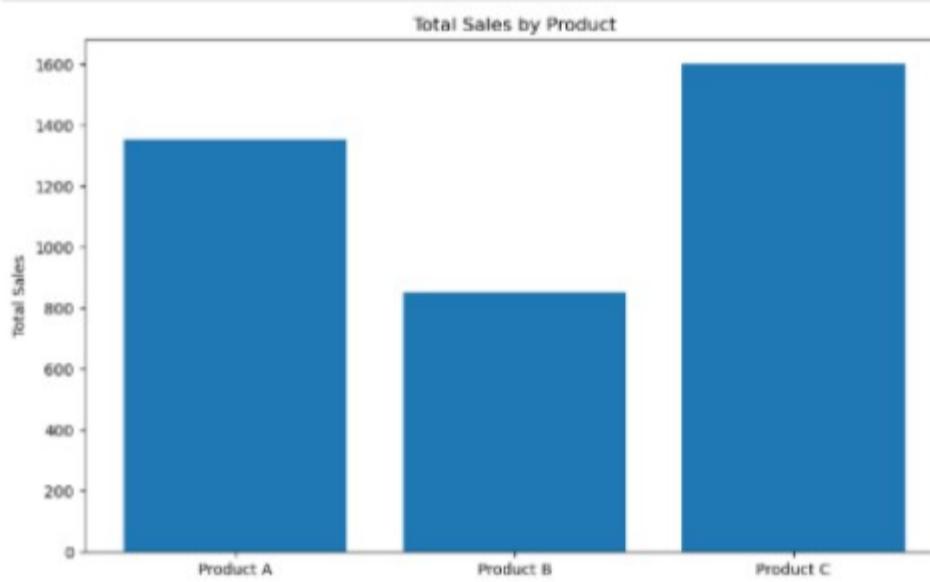
[5]:

	Sales	Quantity
count	16.000000	16.000000
mean	237.500000	5.375000
std	64.091242	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

```
[6]: product_summary = df.groupby('Product').agg({
     'Sales': 'sum',
     'Quantity': 'sum'
     }).reset_index()
     print(product_summary)

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

[7]: 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.show()
```
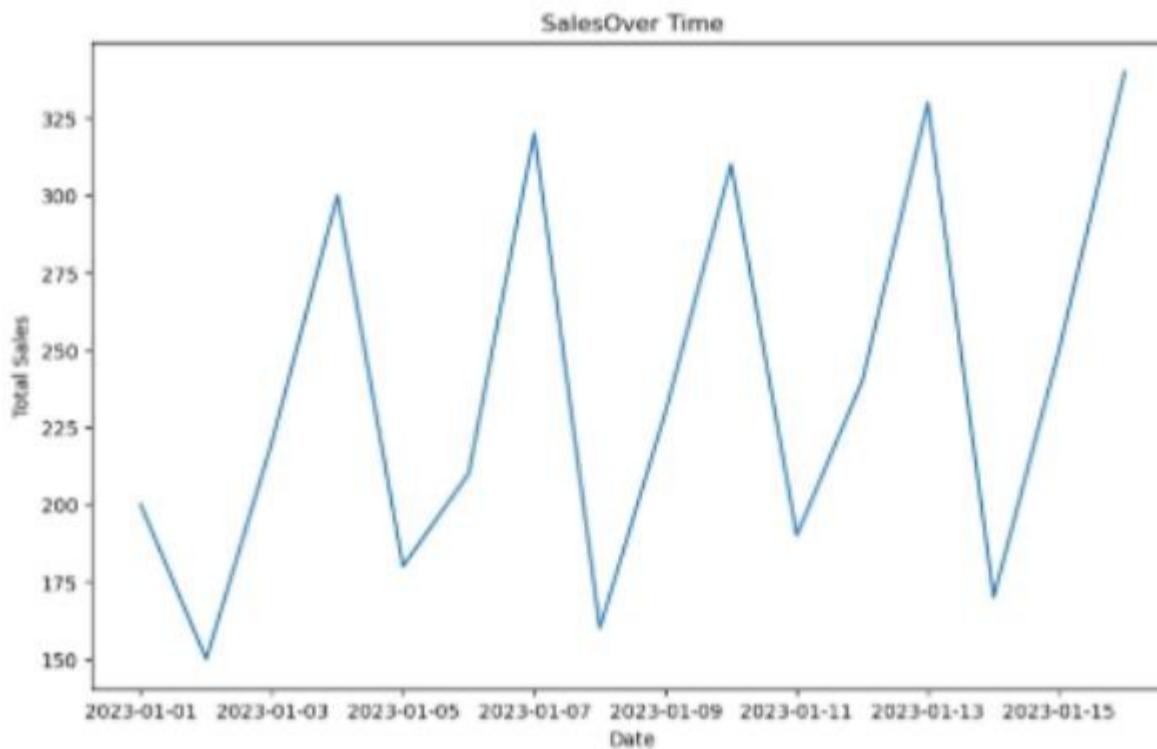
```
[8]: df["Date"] = pd.to_datetime(df["Date"] ,dayfirst = True)
     sales_over_time = df.groupby("Date").agg({"Sales": "sum"}).reset_index()
     plt.figure(figsize=(10, 6))
     plt.plot(sales_over_time["Date"],sales_over_time["Sales"])
     plt.xlabel("Date")
     plt.ylabel("Total Sales")
     plt.title("SalesOver Time")
     plt.show()
```


SalesOver Time

```
[9]: pivot_table = df.pivot_table(values="Sales", index="Region", columns="Product",
     aggfunc=np.sum, fill_value=0)
     pivot_table
```

[9]:
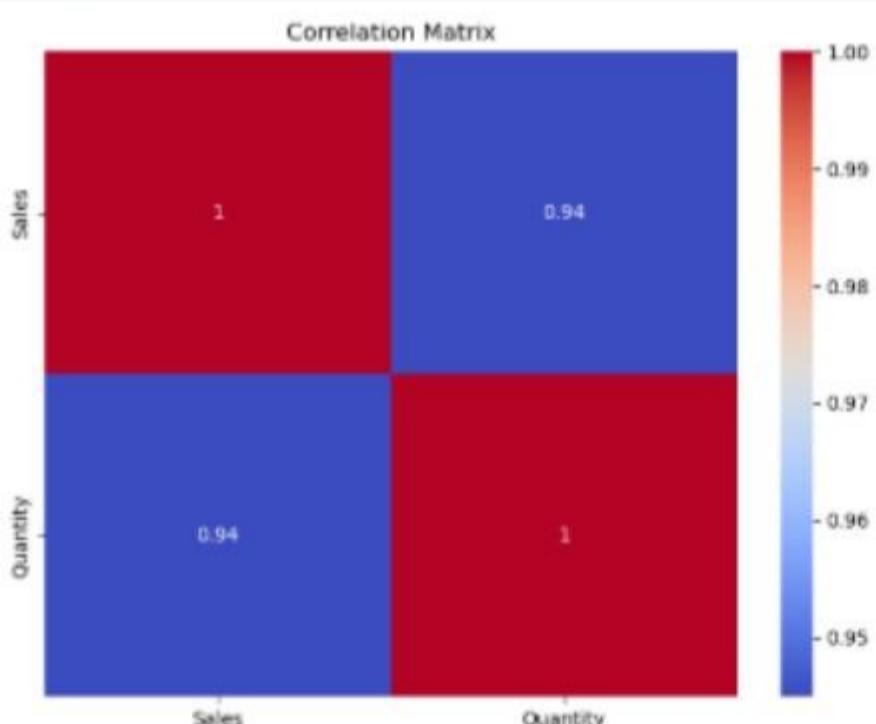
Product	Product A	Product B	Product C
Region			
East	0	0	1600
North	1350	0	0
South	0	480	0
West	0	370	0

```
[10]: numeric_df = df.select_dtypes(include="number")
      correlation_matrix = numeric_df.corr()
      correlation_matrix
```

[10]:

	Sales	Quantity
Sales	1.000000	0.944922
Quantity	0.944922	1.000000

```
[11]: import seaborn as sns
      plt.figure(figsize=(8, 6))
      sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm")
      plt.title("Correlation Matrix")
      plt.show()
```


Correlation Matrix

Result:

Thus, the Python program is executed successfully for analyzing the given dataset using basic Pandas operations and visualizing the results.