```
 Main Menu
1. Load Dataset
2. Explore Data
3. Perform DataFrame Operations
4. Clean and Handle Missing Data
5. Generate Descriptive Statistics
6. Data Visualization
7. Save and Display Last Plot
8. Exit
Enter your choice:  1
Enter CSV file path:  C:\Users\Ritu\Desktop\DATA_ANAYLIST\synthetic_sales_data_2000.csv
Dataset loaded successfully!

 Main Menu
1. Load Dataset
2. Explore Data
3. Perform DataFrame Operations
4. Clean and Handle Missing Data
5. Generate Descriptive Statistics
6. Data Visualization
7. Save and Display Last Plot
8. Exit
Enter your choice:  2


 Explore Data Menu
1. Explore Dataset
2. Combine Data
3. Remove Duplicate Rows
4. Back to Main Menu
Enter your option:  1
```

```
Enter your option:  1

== Explore Data ==
1. First 5 rows
2. Last 5 rows
3. Columns
4. Data types
5. Info
6.Go back
Enter choice:  1
       Date  Product    Sales Region  Profit
0  01-01-23   Tablet   600.45  North  132.75
1  01-01-23   Tablet   291.57  South   30.16
2  01-01-23   Laptop  1388.55   East  298.97
3  01-01-23   Tablet   784.23   East   96.28
4  01-01-23  Monitor  1079.46   West  168.74

== Explore Data ==
1. First 5 rows
2. Last 5 rows
3. Columns
4. Data types
5. Info
6.Go back
Enter choice:  2
          Date     Product    Sales   Region  Profit
1995  13-05-24  Smartphone   841.00    South  129.26
1996  13-05-24     Monitor  1022.30  Central  297.84
1997  14-05-24    Keyboard    61.11     East   14.22
1998  14-05-24  Smartphone   165.59    South   46.68
```

```
1998  14-05-24  Smartphone   165.59    South   46.68
1999  14-05-24  Smartphone   192.88    North   55.46

== Explore Data ==
1. First 5 rows
2. Last 5 rows
3. Columns
4. Data types
5. Info
6.Go back
Enter choice:  3
Index(['Date', 'Product', 'Sales', 'Region', 'Profit'], dtype='object')

== Explore Data ==
1. First 5 rows
2. Last 5 rows
3. Columns
4. Data types
5. Info
6.Go back
Enter choice:  4
Date        object
Product     object
Sales      float64
Region      object
Profit     float64
dtype: object

== Explore Data ==
1. First 5 rows
```

```
2. Last 5 rows
3. Columns
4. Data types
5. Info
6.Go back
Enter choice:  5
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 5 columns):
 #   Column   Non-Null Count  Dtype
---  ------   --------------  -----
 0   Date     2000 non-null   object
 1   Product  2000 non-null   object
 2   Sales    2000 non-null   float64
 3   Region   2000 non-null   object
 4   Profit   1999 non-null   float64
dtypes: float64(2), object(3)
memory usage: 78.3+ KB

== Explore Data ==
1. First 5 rows
2. Last 5 rows
3. Columns
4. Data types
5. Info
6.Go back
Enter choice:  6

 Explore Data Menu
1. Explore Dataset
```

```
2. Combine Data
3. Remove Duplicate Rows
4. Back to Main Menu
Enter your option:  2
Enter CSV path of dataset to combine:  C:\Users\Ritu\Desktop\DATA_ANAYLIST\synthetic_sales_data_2000.csv
Datasets combined. Total rows: 4000

 Explore Data Menu
1. Explore Dataset
2. Combine Data
3. Remove Duplicate Rows
4. Back to Main Menu
Enter your option:  3
Duplicates removed: 2000
Total rows now: 2000

 Explore Data Menu
1. Explore Dataset
2. Combine Data
3. Remove Duplicate Rows
4. Back to Main Menu
Enter your option:  4

 Main Menu
1. Load Dataset
2. Explore Data
3. Perform DataFrame Operations
4. Clean and Handle Missing Data
5. Generate Descriptive Statistics
6. Data Visualization
7. Save and Display Last Plot
8. Exit
Enter your choice:  3

 DataFrame Operations Menu
1. Convert DataFrame to NumPy Array
2. Search / Sort / Filter Data
3. Aggregate Functions
4. Statistical Analysis
5. Back to Main Menu
Enter your option:  1
Index(['Date', 'Product', 'Sales', 'Region', 'Profit'], dtype='object')
Enter column names to convert to NumPy (comma-separated, or leave blank for all):  Sales,Profit

Data successfully converted into a NumPy array.
[[ 600.45  132.75]
 [ 291.57   30.16]
 [1388.55  298.97]
 ...
 [  61.11   14.22]
 [ 165.59   46.68]
 [ 192.88   55.46]]
Enter row index:  0
Enter column index:  1
Value: 132.75

 DataFrame Operations Menu
```

```
1. Convert DataFrame to NumPy Array
2. Search / Sort / Filter Data
3. Aggregate Functions
4. Statistical Analysis
5. Back to Main Menu
Enter your option:  2
1.Search
2.Sort
3.filter
4.back
enter choice 1
Index(['Date', 'Product', 'Sales', 'Region', 'Profit'], dtype='object')
Column to search:  Product
Value to search:  laptop
339
        Date Product    Sales   Region  Profit
2   01-01-23  Laptop  1388.55     East  298.97
8   02-01-23  Laptop  1162.58     West  324.46
21  05-01-23  Laptop  1049.84     East  165.38
23  05-01-23  Laptop   663.17  Central  220.45
27  06-01-23  Laptop   760.62     West  177.63
38  08-01-23  Laptop   571.75  Central   92.32
51  12-01-23  Laptop   503.67  Central  112.39
53  12-01-23  Laptop   559.00    North  147.75
55  13-01-23  Laptop  1239.98  Central  427.68
61  14-01-23  Laptop   864.49     West  129.97
1.Search
2.Sort
3.filter
4.back
enter choice 2
Index(['Date', 'Product', 'Sales', 'Region', 'Profit'], dtype='object')

Enter column name to sort by:  Region

Top 5 rows sorted by 'Region' (descending):
         Date     Product    Sales Region  Profit
1021  16-09-23      Laptop   802.88   West  245.52
1270  15-11-23     Monitor  1265.08   West  266.20
368   09-04-23      Laptop  1338.28   West  326.62
366   09-04-23  Smartphone   840.14   West  243.05
1675  23-02-24     Monitor   750.07   West  160.75
1.Search
2.Sort
3.filter
4.back
enter choice 3
Index(['Date', 'Product', 'Sales', 'Region', 'Profit'], dtype='object')
Enter numeric column to filter:  Sales
Keep rows where 'Sales' >= value:  1500

Filtered data where 'Sales' >= 1500.0:
Empty DataFrame
Columns: [Date, Product, Sales, Region, Profit]
Index: []
1.Search
2.Sort
```

```
3.filter
4.back
enter choice 3
Index(['Date', 'Product', 'Sales', 'Region', 'Profit'], dtype='object')
Enter numeric column to filter:  Sales
Keep rows where 'Sales' >= value:  800

Filtered data where 'Sales' >= 800.0:
        Date  Product    Sales   Region  Profit
2   01-01-23   Laptop  1388.55     East  298.97
4   01-01-23  Monitor  1079.46     West  168.74
6   02-01-23  Monitor  1363.63    South  235.90
8   02-01-23   Laptop  1162.58     West  324.46
9   02-01-23   Tablet   888.59    North  126.34
15  04-01-23  Monitor   884.19     East  251.95
17  04-01-23  Monitor   840.98    South  175.47
21  05-01-23   Laptop  1049.84     East  165.38
32  06-01-23  Monitor  1375.37    South  208.26
33  07-01-23  Monitor  1257.54  Central  386.98
1.Search
2.Sort
3.filter
4.back
enter choice 4
Index(['Date', 'Product', 'Sales', 'Region', 'Profit'], dtype='object')

 DataFrame Operations Menu
1. Convert DataFrame to NumPy Array
2. Search / Sort / Filter Data
```

```
3. Aggregate Functions
4. Statistical Analysis
5. Back to Main Menu
Enter your option:  3
Index(['Date', 'Product', 'Sales', 'Region', 'Profit'], dtype='object')
Enter numeric column to aggregate:  Sales

Aggregate for 'Sales':
Sum: 1022657.53
Mean: 511.33
Count: 2000

 DataFrame Operations Menu
1. Convert DataFrame to NumPy Array
2. Search / Sort / Filter Data
3. Aggregate Functions
4. Statistical Analysis
5. Back to Main Menu
Enter your option:  4
Index(['Date', 'Product', 'Sales', 'Region', 'Profit'], dtype='object')
Enter numeric column for stats:  Sales

Statistics for 'Sales':
Std Dev: 421.31
Variance: 177498.82
Quantiles:
0.25     82.2625
0.50    461.6300
```

```
0.75    791.4900
Name: Sales, dtype: float64

 DataFrame Operations Menu
1. Convert DataFrame to NumPy Array
2. Search / Sort / Filter Data
3. Aggregate Functions
4. Statistical Analysis
5. Back to Main Menu
Enter your option:  5

 Main Menu
1. Load Dataset
2. Explore Data
3. Perform DataFrame Operations
4. Clean and Handle Missing Data
5. Generate Descriptive Statistics
6. Data Visualization
7. Save and Display Last Plot
8. Exit
Enter your choice:  4

Handle Missing Data
1. Show missing rows
2. Fill with mean
3. Drop missing rows
4.Go back
Enter choice:  1
Date       0
Product    0
Sales      0
Region     0
Profit     1
dtype: int64

Handle Missing Data
1. Show missing rows
2. Fill with mean
3. Drop missing rows
4.Go back
Enter choice:  2
Filled with mean

Handle Missing Data
1. Show missing rows
2. Fill with mean
3. Drop missing rows
4.Go back
Enter choice:  1
Date       0
Product    0
Sales      0
Region     0
Profit     0
dtype: int64

Handle Missing Data
1. Show missing rows
2. Fill with mean
```

```
2. Fill with mean
3. Drop missing rows
4.Go back
Enter choice:  3
Dropped missing rows

Handle Missing Data
1. Show missing rows
2. Fill with mean
3. Drop missing rows
4.Go back
Enter choice:  4


 Main Menu
1. Load Dataset
2. Explore Data
3. Perform DataFrame Operations
4. Clean and Handle Missing Data
5. Generate Descriptive Statistics
6. Data Visualization
7. Save and Display Last Plot
8. Exit
Enter your choice:  5


 Descriptive Statistics Menu
1. Generate Descriptive Statistics
2. GroupBy and Transform
3. Back to Main Menu
Enter your option:  1
```

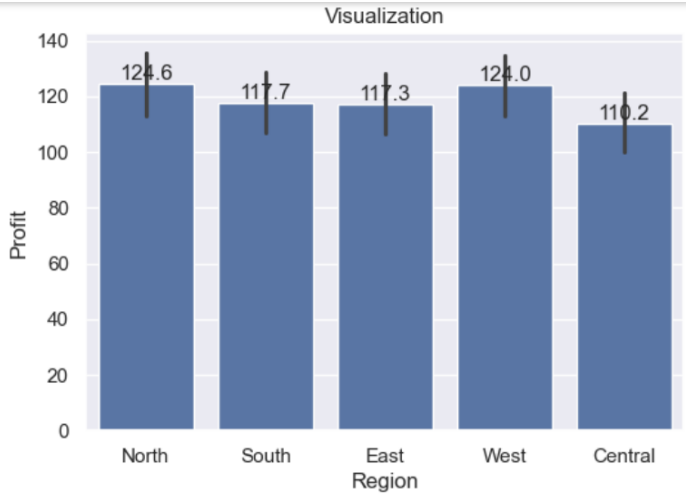
```
Index(['Date', 'Product', 'Sales', 'Region', 'Profit'], dtype='object')
Enter index columns (comma-separated):  Product
Enter value columns (comma-separated):  Sales
Enter aggregation function (sum, mean, count):  sum

Pivot Table:
                Sales
Product
Keyboard     18137.77
Laptop      306357.88
Monitor     329392.47
Mouse        18503.36
Smartphone  175032.23
Tablet      175233.82

--- Statistics Menu ---
1. Statistical Analysis
2. Pivot Table
3. Back to Main Menu
Enter Choice: 3

 Descriptive Statistics Menu
1. Generate Descriptive Statistics
2. GroupBy and Transform
3. Back to Main Menu
Enter your option:  2
Index(['Date', 'Product', 'Sales', 'Region', 'Profit'], dtype='object')

--- GroupBy & Transform Menu ---
```

```
=== GroupBy & Transform Menu ===
1. GroupBy + Aggregate (sum / mean)
2. Transform: Percentage of Group Total
3. Back
Enter choice:  1
Enter column to group by (e.g., Region):  Region
Enter numeric column (e.g., Sales):  Sales
Aggregation (sum / mean / count):  mean
Grouped Result:
Region
Central    469.090911
East       513.176813
North      537.291894
South      504.735914
West       530.645291
Name: Sales, dtype: float64

=== GroupBy & Transform Menu ===
1. GroupBy + Aggregate (sum / mean)
2. Transform: Percentage of Group Total
3. Back
Enter choice:  2
Group by column (e.g., Region):  Region
Numeric column (e.g., Sales):  Sales
  Region    Sales  percent_of_group
0  North   600.45          0.282209
1  South   291.57          0.146616
2   East  1388.55          0.700983
3   East   784.23          0.395903
4   West  1079.46          0.474182
```

```
4   West  1079.46          0.474182

=== GroupBy & Transform Menu ===
1. GroupBy + Aggregate (sum / mean)
2. Transform: Percentage of Group Total
3. Back
Enter choice:  3

 Descriptive Statistics Menu
1. Generate Descriptive Statistics
2. GroupBy and Transform
3. Back to Main Menu
Enter your option:  6
Invalid option! Try again.

 Descriptive Statistics Menu
1. Generate Descriptive Statistics
2. GroupBy and Transform
3. Back to Main Menu
Enter your option:  3

 Main Menu
1. Load Dataset
2. Explore Data
3. Perform DataFrame Operations
4. Clean and Handle Missing Data
5. Generate Descriptive Statistics
6. Data Visualization
7. Save and Display Last Plot
```

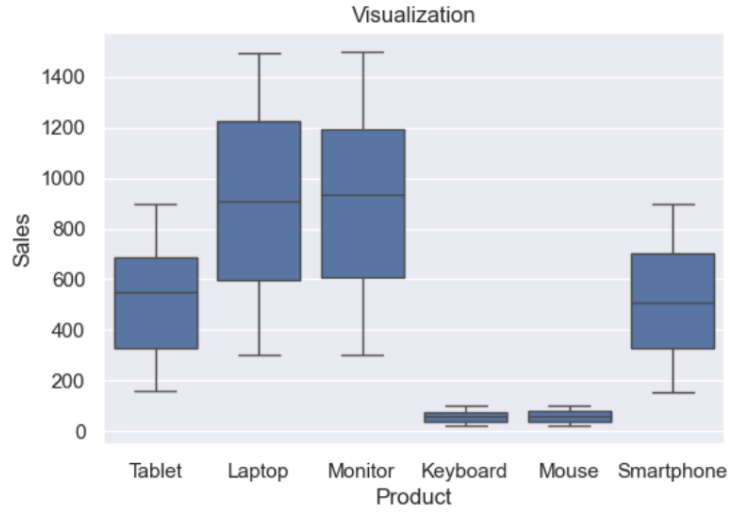
---

```
 7. Save and Display Last Plot
 8. Exit
 Enter your choice:  6
 Index(['Date', 'Product', 'Sales', 'Region', 'Profit', 'percent_of_group'], dtype='object')

 ====Visualization Menu ====
 1. Bar Plot
 2. Box Plot
 3. Scatter Plot
 4. Histogram
 5. Heatmap
 6. Pie Chart
 7. Stack Plot
 8. Go Back
 Enter Choice: 1
 Index(['Date', 'Product', 'Sales', 'Region', 'Profit', 'percent_of_group'], dtype='object')
 Enter x-axis column:(e.g., Region)  Region
 Enter y-axis column:(e.g., Sales)  Profit
```



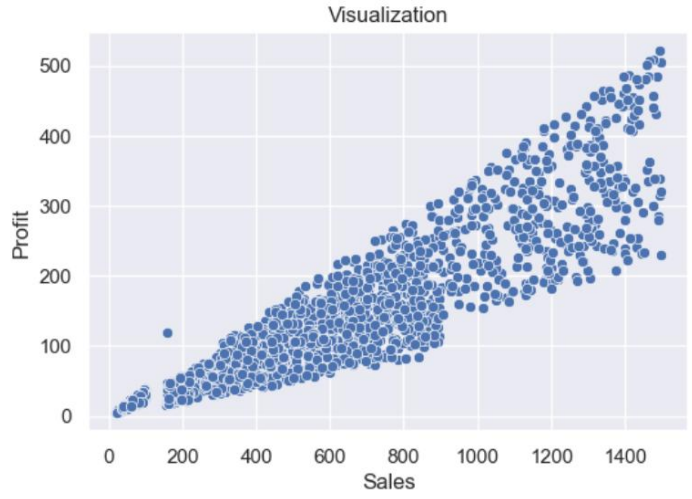

```
Plot displayed successfully!

====Visualization Menu ====
1. Bar Plot
2. Box Plot
3. Scatter Plot
4. Histogram
5. Heatmap
```

```
Enter Choice: 2
Index(['Date', 'Product', 'Sales', 'Region', 'Profit', 'percent_of_group'], dtype='object')
Enter x-axis column:(e.g., Category)  Product
Enter y-axis column:(e.g., Sales)  Sales
```



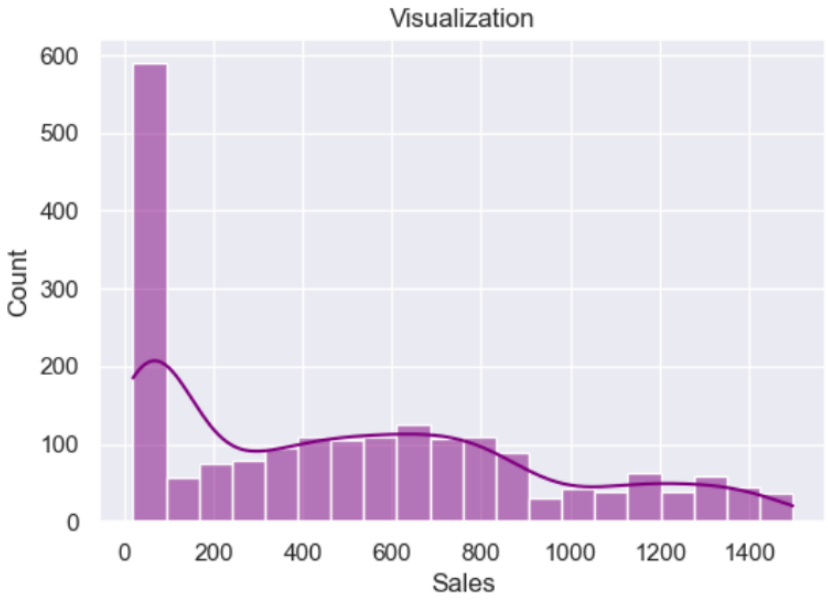

```
Plot displayed successfully!
```

```
====Visualization Menu ====
1. Bar Plot
2. Box Plot
3. Scatter Plot
4. Histogram
5. Heatmap
6. Pie Chart
7. Stack Plot
8. Go Back
Enter Choice: 3
Index(['Date', 'Product', 'Sales', 'Region', 'Profit', 'percent_of_group'], dtype='object')
Enter x-axis column name:  Sales
Enter y-axis column name:  Profit
```



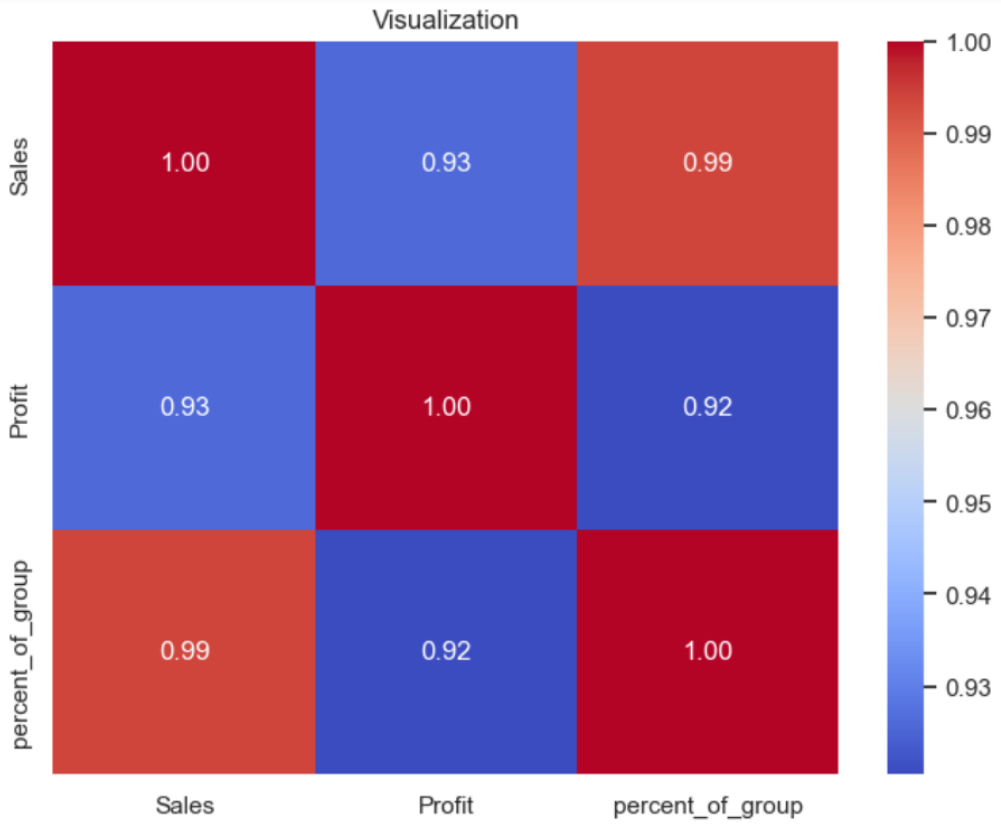

```
Plot displayed successfully!
```

```
====Visualization Menu ====
1. Bar Plot
2. Box Plot
3. Scatter Plot
4. Histogram
5. Heatmap
6. Pie Chart
7. Stack Plot
8. Go Back
Enter Choice: 4
Index(['Date', 'Product', 'Sales', 'Region', 'Profit', 'percent_of_group'], dtype='object')
Enter numeric column for histogram (e.g., Sales):  Sales
Enter number of bins (e.g., 30):  20
```

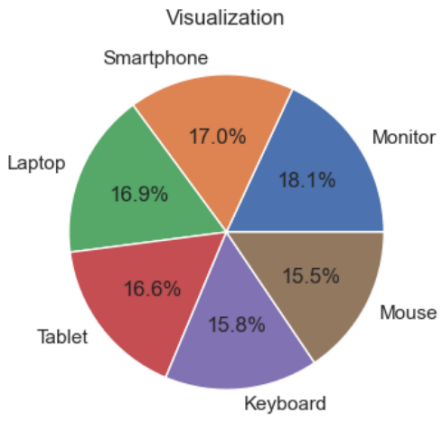



```
Plot displayed successfully!
```

```
Plot displayed successfully!

====Visualization Menu ====
1. Bar Plot
2. Box Plot
3. Scatter Plot
4. Histogram
5. Heatmap
6. Pie Chart
7. Stack Plot
8. Go Back
Enter Choice: 5
Index(['Date', 'Product', 'Sales', 'Region', 'Profit', 'percent_of_group'], dtype='object')
<Figure size 600x400 with 0 Axes>
```

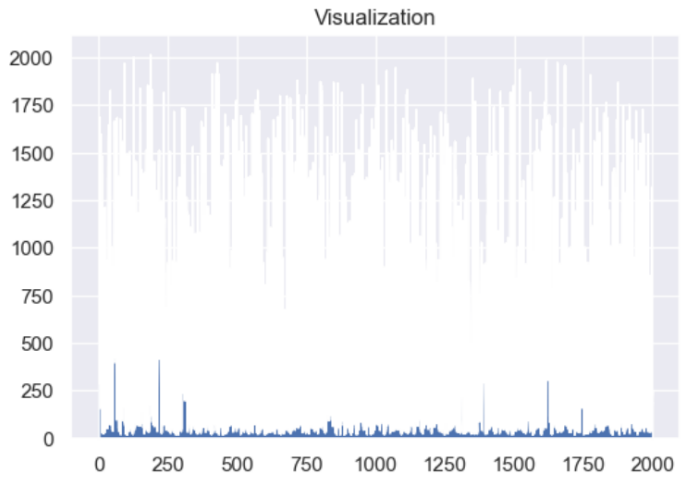



```
5. Heatmap
6. Pie Chart
7. Stack Plot
8. Go Back
Enter Choice: 6
Index(['Date', 'Product', 'Sales', 'Region', 'Profit', 'percent_of_group'], dtype='object')
Enter coloumn for pie chart:  Product
```




```
Plot displayed successfully!
```

```
7. Stack Plot
8. Go Back
Enter Choice: 7
Index(['Date', 'Product', 'Sales', 'Region', 'Profit', 'percent_of_group'], dtype='object')
```




```
Plot displayed successfully!

6. Pie Chart
7. Stack Plot
8. Go Back
Enter Choice: 8


 Main Menu
1. Load Dataset
2. Explore Data
3. Perform DataFrame Operations
4. Clean and Handle Missing Data
5. Generate Descriptive Statistics
6. Data Visualization
7. Save and Display Last Plot
8. Exit
Enter your choice:  7
Enter file name (e.g., scatter_plot.png):  seaborn.png
Visualization saved as C:\Users\Ritu\Desktop\DATA_ANAYLIST\seaborn.png successfully!

 Main Menu
1. Load Dataset
2. Explore Data
3. Perform DataFrame Operations
4. Clean and Handle Missing Data
5. Generate Descriptive Statistics
6. Data Visualization
7. Save and Display Last Plot
8. Exit
Enter your choice:  8
Exiting the program. Goodbye!
```