Exp.No:04	Multivariate Analysis
07-04-23	

Aim: To Perform Multivariate analysis on the given data sets.

Algorithm:

STEP 1: Import the built libraries required to perform EDA and outlier removal.

STEP 2: Read the given csv file

STEP 3: Convert the file into a dataframe and get information of the data.

STEP 4: Return the objects containing counts of unique values using (value_counts()).

STEP 5: Plot the counts in the form of Histogram or Bar Graph.

STEP 6: Use seaborn the bar graph comparison of data can be viewed.

STEP 7: Find the pairwise correlation of all columns in the dataframe.corr()

STEP 8: Save the final data set into the file

Program & Output:

1.SuperStore dataset

Code:

SuperStore Dataset

```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot  as plt
df=pd.read_csv("/content/SuperStore.csv")
df.head()
```

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country	City	State	Postal Code	Region	Product ID	Category	Sub-Category
0	1	CA-2017-152156	08-11-2017	11-11-2017	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420.0	South	FUR-BO-10001798	Furniture	Bookcases
1	2	CA-2017-152156	08-11-2017	11-11-2017	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420.0	South	FUR-CH-10000454	Furniture	Chairs
2	3	CA-2017-138688	12-06-2017	16-06-2017	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California	90036.0	West	OFF-LA-10000240	Office Supplies	Labels

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9800 entries, 0 to 9799
Data columns (total 18 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   Row ID         9800 non-null   int64  
 1   Order ID       9800 non-null   object 
 2   Order Date     9800 non-null   object 
 3   Ship Date      9800 non-null   object 
 4   Ship Mode      9800 non-null   object 
 5   Customer ID    9800 non-null   object 
 6   Customer Name  9800 non-null   object 
 7   Segment        9800 non-null   object 
 8   Country        9800 non-null   object 
 9   City           9800 non-null   object 
 10  State          9800 non-null   object 
 11  Postal Code    9789 non-null   float64
 12  Region         9800 non-null   object 
 13  Product ID     9800 non-null   object 
 14  Category       9800 non-null   object 
 15  Sub-Category   9800 non-null   object 
 16  Product Name   9800 non-null   object 
 17  Sales          9800 non-null   float64
dtypes: float64(2), int64(1), object(15)
```

```
df.describe()
```

	Row ID	Postal Code	Sales
count	9800.000000	9789.000000	9800.000000
mean	4900.500000	55273.322403	230.769059
std	2829.160653	32041.223413	626.651875
min	1.000000	1040.000000	0.444000
25%	2450.750000	23223.000000	17.248000
50%	4900.500000	58103.000000	54.490000
75%	7350.250000	90008.000000	210.605000
max	9800.000000	99301.000000	22638.480000

```
df.isnull().sum()
```

```
Row ID           0
Order ID         0
Order Date       0
Ship Date        0
Ship Mode        0
Customer ID      0
Customer Name    0
Segment          0
Country          0
City             0
State            0
Postal Code     11
Region           0
Product ID       0
Category         0
Sub-Category     0
Product Name     0
Sales            0
dtype: int64
```

```
df['Postal Code']=df["Postal Code"].fillna(df['Postal Code'].mode()[0])
df.isnull().sum()
```

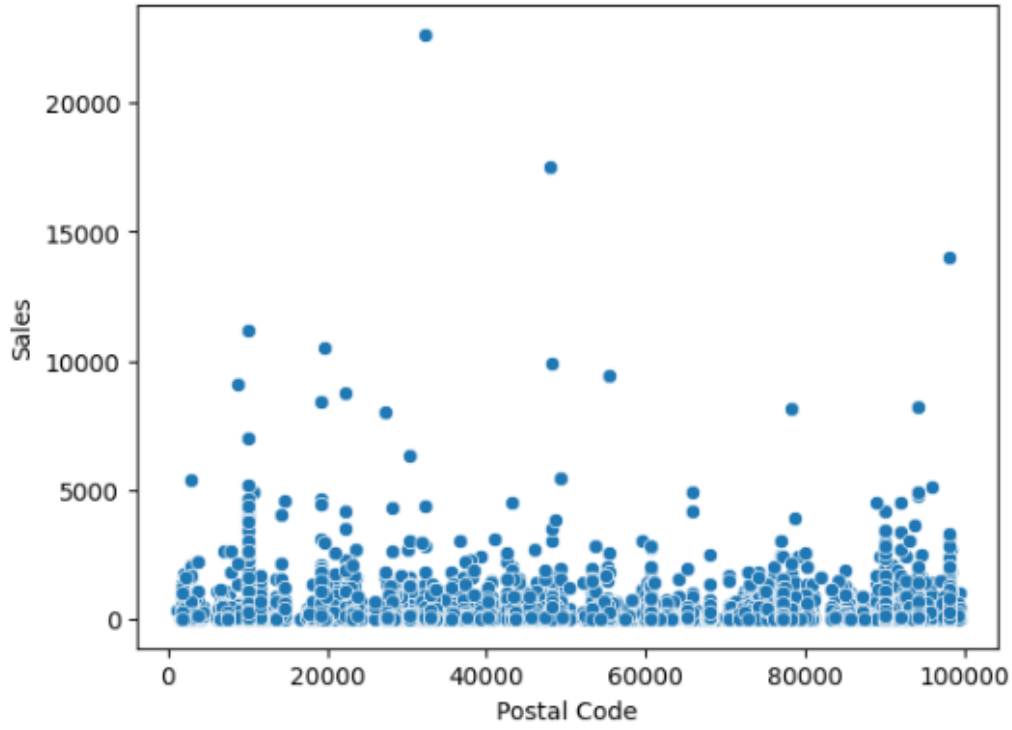
```
Row ID          0
Order ID        0
Order Date      0
Ship Date       0
Ship Mode       0
Customer ID     0
Customer Name   0
Segment         0
Country         0
City            0
State           0
Postal Code     0
Region          0
Product ID      0
Category        0
Sub-Category    0
Product Name    0
Sales           0
dtype: int64
```

```
df.dtypes
```

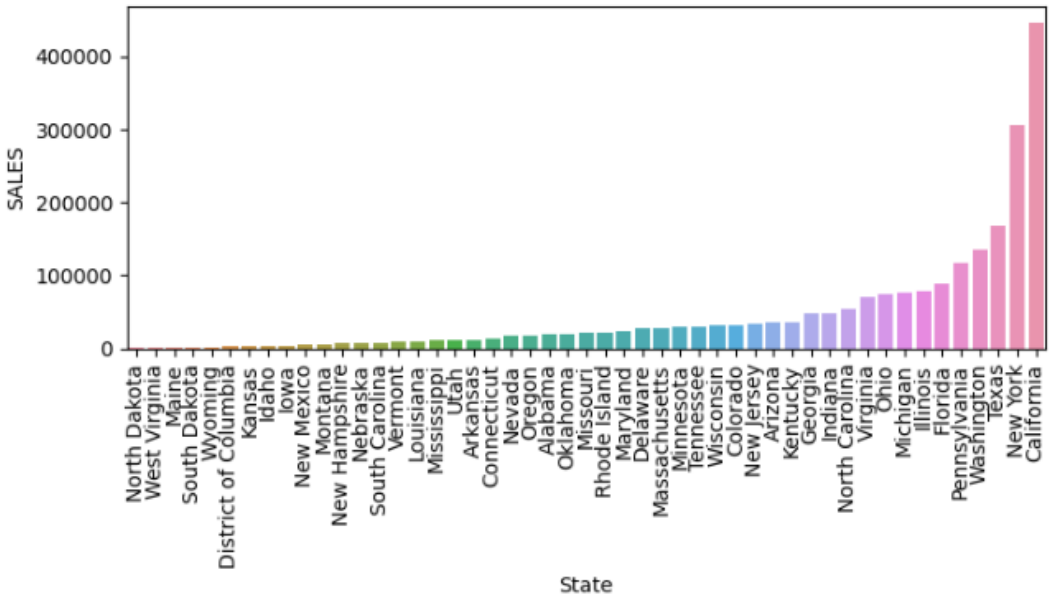
```
Row ID           int64
Order ID        object
Order Date      object
Ship Date       object
Ship Mode       object
Customer ID     object
Customer Name   object
Segment         object
Country         object
City            object
State           object
Postal Code    float64
Region          object
Product ID      object
Category        object
Sub-Category    object
Product Name    object
Sales          float64
dtype: object
```

```
sns.scatterplot(x=df['Postal Code'],y=df['Sales'])
```

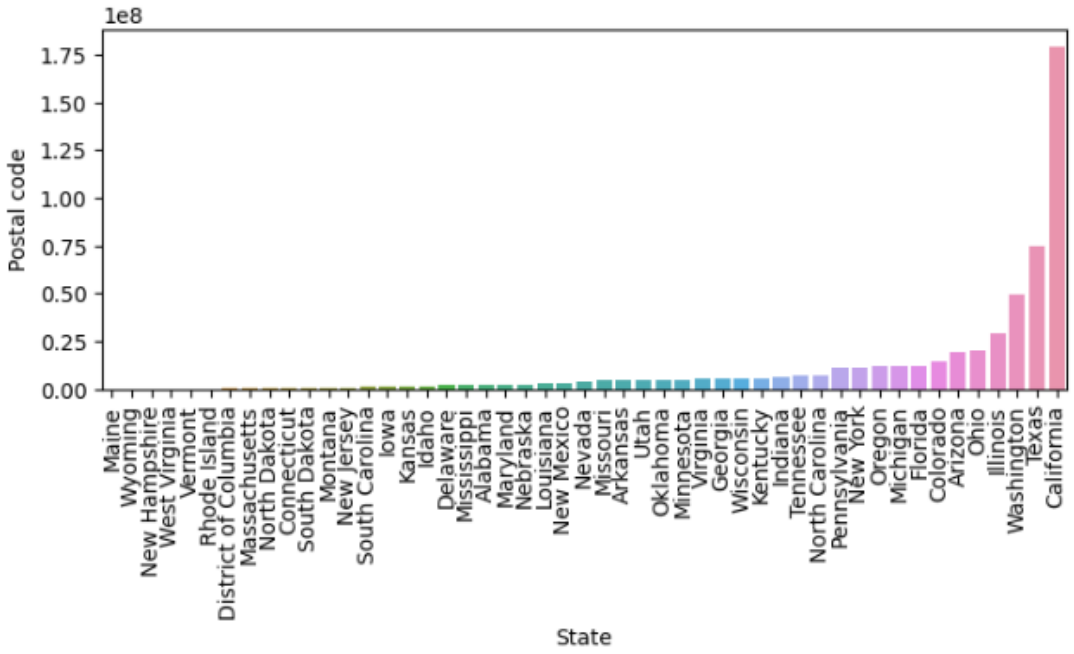
```
<Axes: xlabel='Postal Code', ylabel='Sales'>
```



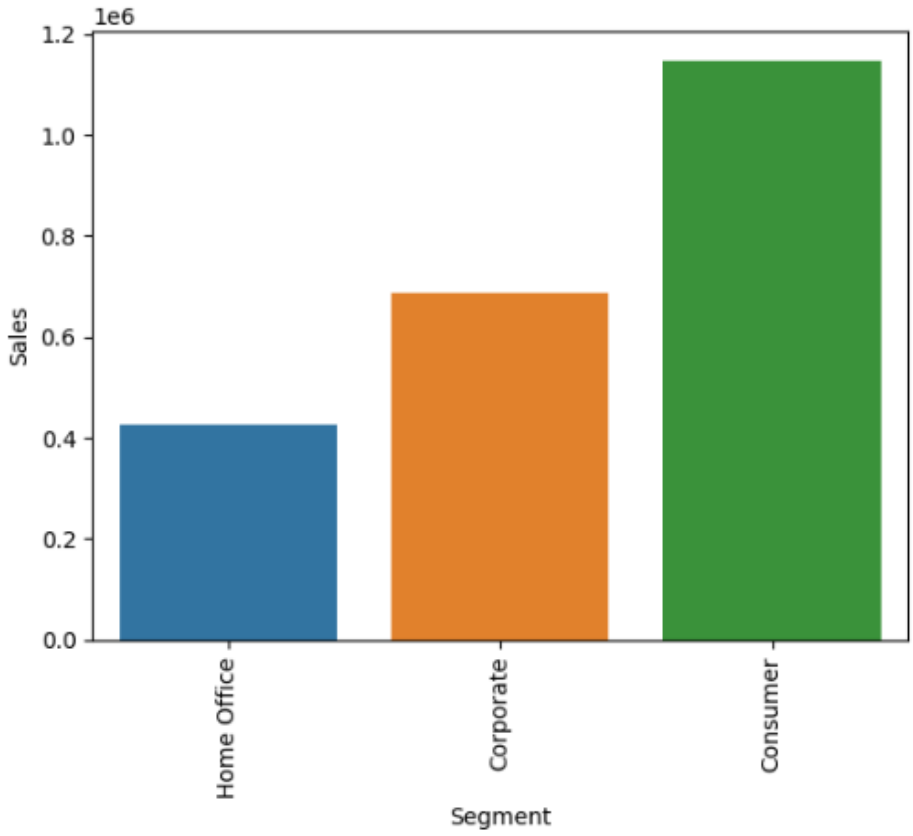

```
states=df.loc[:,["State","Sales"]]
states=states.groupby(by=["State"]).sum().sort_values(by="Sales")
plt.figure(figsize=(8,3))
sns.barplot(x=states.index,y="Sales",data=states)
plt.xticks(rotation = 90)
plt.xlabel("STATES")
plt.ylabel("SALES")
plt.show()
```



```
[10] states=df.loc[:,["State","Postal Code"]]
     states=states.groupby(by=["State"]).sum().sort_values(by="Postal Code")
     plt.figure(figsize=(8,3))
     sns.barplot(x=states.index,y="Postal Code",data=states)
     plt.xticks(rotation = 90)
     plt.xlabel=("STATES")
     plt.ylabel("Postal code")
     plt.show()
```

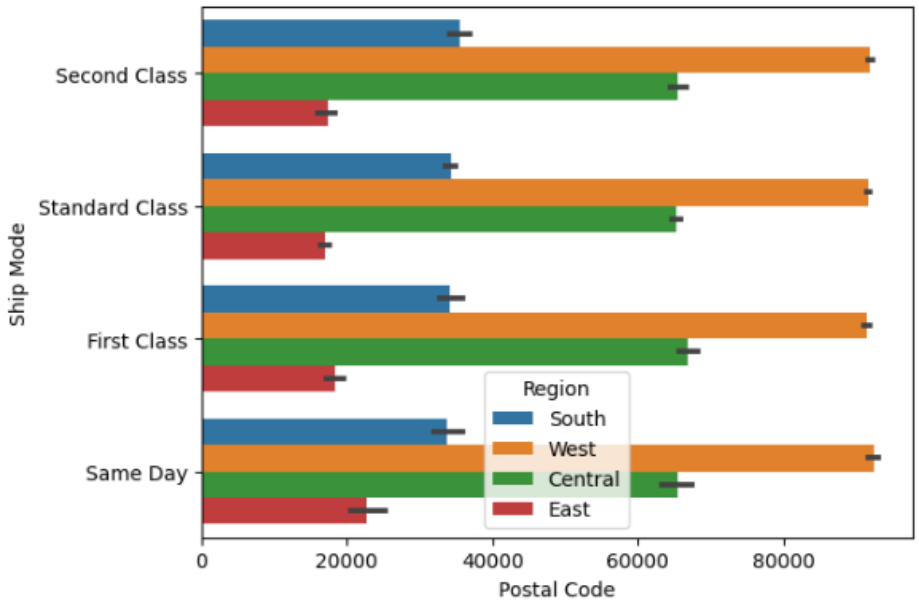


```
states=df.loc[:,["Segment","Sales"]]
states=states.groupby(by=["Segment"]).sum().sort_values(by="Sales")
#plt.figure(figsize=(10,7))
sns.barplot(x=states.index,y="Sales",data=states)
plt.xticks(rotation = 90)
plt.xlabel=("SEGMENT")
plt.ylabel=("SALES")
plt.show()
```



```
sns.barplot(x=df['Postal Code'],y=df['Ship Mode'],hue=df['Region'])
```

```
<Axes: xlabel='Postal Code', ylabel='Ship Mode'>
```




```
df.corr()
```

	Row ID	Postal Code	Sales
Row ID	1.000000	0.011723	0.001151
Postal Code	0.011723	1.000000	-0.025444
Sales	0.001151	-0.025444	1.000000

```
sns.heatmap(df.corr(),annot=True)
```

```
<Axes: >
```