TASK3-EXPLORATORY DATA ANALYSIS- RETAIL

BY- VATSAL OJHA

MOUNTING GDRIVE ONTO THE COLAB NOTEBOOK

```
from google.colab import drive
drive.mount('/content/drive',force_remount=True)
print("MOUNTED SUCCESSFULLY")
```

```
Mounted at /content/drive
MOUNTED SUCCESSFULLY
```

IMPORTING THE NECESSARY LIBRARIES

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(color_codes=True)
```

IMPORTING AND DISPLAYING DATA HEAD AND TAIL

```
path='/content/drive/MyDrive/Colab Notebooks/SampleSuperstore.csv'
data=pd.read_csv(path)
print(data.head())
print(data.tail())
```

```
        Ship Mode    Segment        Country  ... Quantity Discount    Profit
0    Second Class   Consumer  United States  ...        2     0.00   41.9136
1    Second Class   Consumer  United States  ...        3     0.00  219.5820
2    Second Class  Corporate  United States  ...        2     0.00    6.8714
3  Standard Class   Consumer  United States  ...        5     0.45 -383.0310
4  Standard Class   Consumer  United States  ...        2     0.20    2.5164

[5 rows x 13 columns]
           Ship Mode   Segment        Country  ... Quantity Discount   Profit
9989    Second Class  Consumer  United States  ...        3      0.2   4.1028
9990  Standard Class  Consumer  United States  ...        2      0.0  15.6332
9991  Standard Class  Consumer  United States  ...        2      0.2  19.3932
9992  Standard Class  Consumer  United States  ...        4      0.0  13.3200
9993    Second Class  Consumer  United States  ...        2      0.0  72.9480

[5 rows x 13 columns]
```

HERE I HAVE PRINTED THE DATA TYPE OF EACH COLUMN . A BEFORE-HAND ACCOUNT OF THIS IS A MUST FOR BETTER DATA ANALYTICS.THEN I AM PRINTING THE TOTAL NULL VALUES PRESENT IN THE ENTIRE DATASET ALONG WITH THE DATASET SHAPE

```
print(data.dtypes)
nulls=data.isnull().sum().sum()
print("Total null values=",nulls)
print("Dataset Shape=",data.shape)
```

```
Ship Mode        object
Segment          object
Country          object
City             object
State            object
Postal Code       int64
Region           object
Category         object
Sub-Category     object
Sales           float64
Quantity          int64
Discount        float64
Profit          float64
dtype: object
Total null values= 0
Dataset Shape= (9994, 13)
```

```
data.describe()
```

	Postal Code	Sales	Quantity	Discount	Profit
count	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000
mean	55190.379428	229.858001	3.789574	0.156203	28.656896
std	32063.693350	623.245101	2.225110	0.206452	234.260108
min	1040.000000	0.444000	1.000000	0.000000	-6599.978000
25%	23223.000000	17.280000	2.000000	0.000000	1.728750
50%	56430.500000	54.490000	3.000000	0.200000	8.666500
75%	90008.000000	209.940000	5.000000	0.200000	29.364000
max	99301.000000	22638.480000	14.000000	0.800000	8399.976000

NEXT I HAVE FOUND OUT THE NUMBER OF UNIQUE VALUES PRESENT IN DIFFERENT COLUMNS OF THE DATASET

```
data.nunique()
```

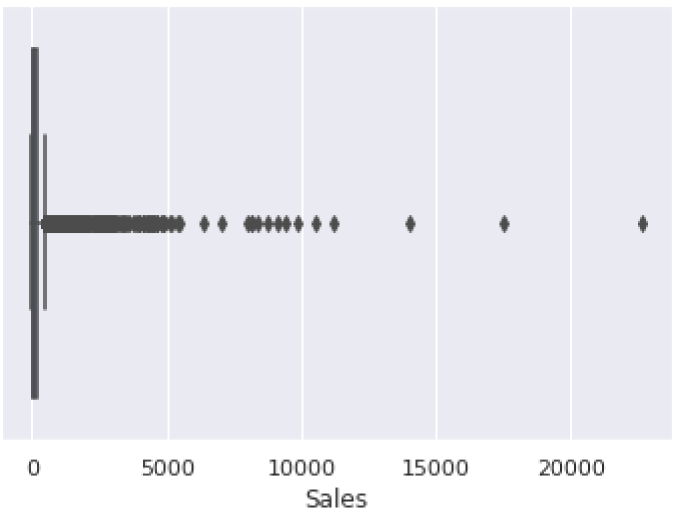
```
Ship Mode         4
Segment           3
Country           1
City            531
State            49
Postal Code     631
Region            4
Category          3
Sub-Category     17
```

```
Sales          5825
Quantity         14
Discount         12
Profit         7287
dtype: int64
```

FOLLOWING ARE A SERIES OF BOXPLOT GRAPHS FOR A NUMBER OF FEATURES PRESENT IN THE DATASET.THIS WOULD HELP IN UNDERSTANDING WHERE ARE THE VALUES OF DIFFERERNT FEATURES CONCENTRATED AND WHAT ALL VALUES ARE OUTLIERS(IF ANY)

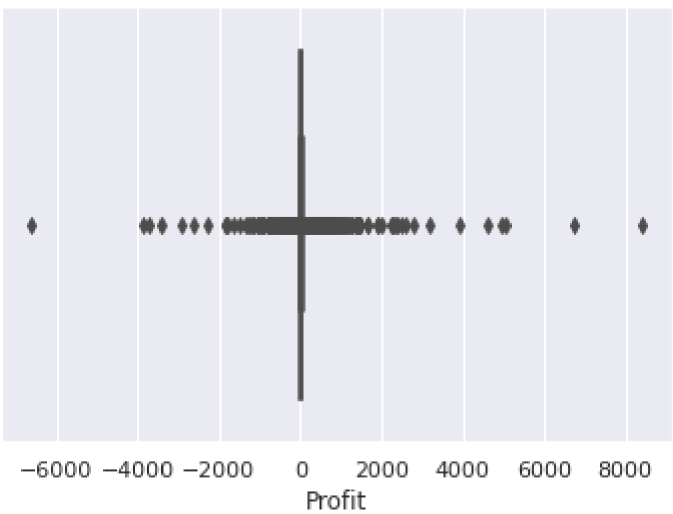
```
sns.boxplot(x=data['Sales'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f199c3dc790>
```




```
sns.boxplot(x=data['Profit'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f199c38e890>
```

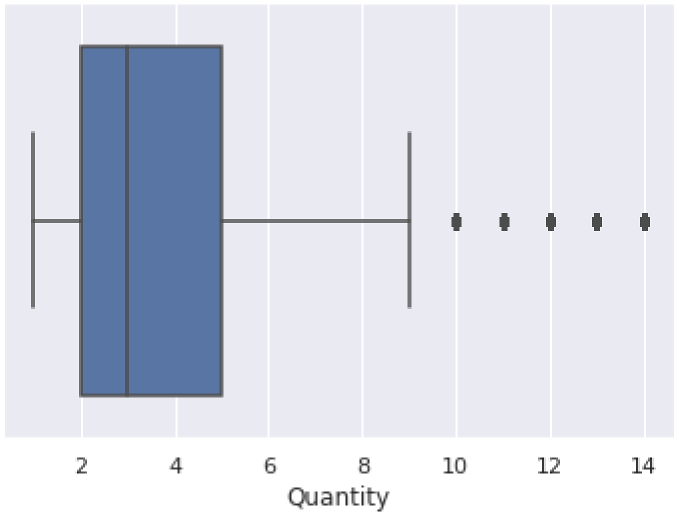



```
print(data['Profit'].sum(axis=0))
```

```
286397.0217
```

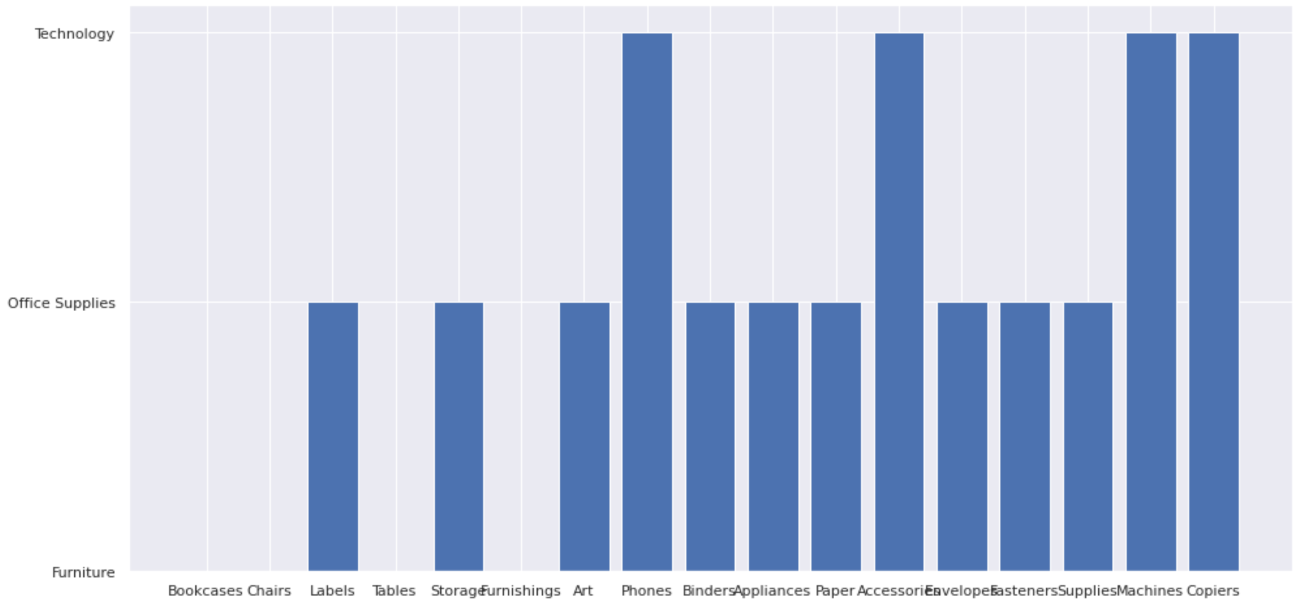
```
sns.boxplot(x=data['Quantity'])
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f199bec36d0>
```



BAR GRAPH BETWEEN SUB-CATEGORY AND CATEGORY

```
plt.figure(figsize=(16,8))
plt.bar('Sub-Category','Category',data=data)
plt.show()
```

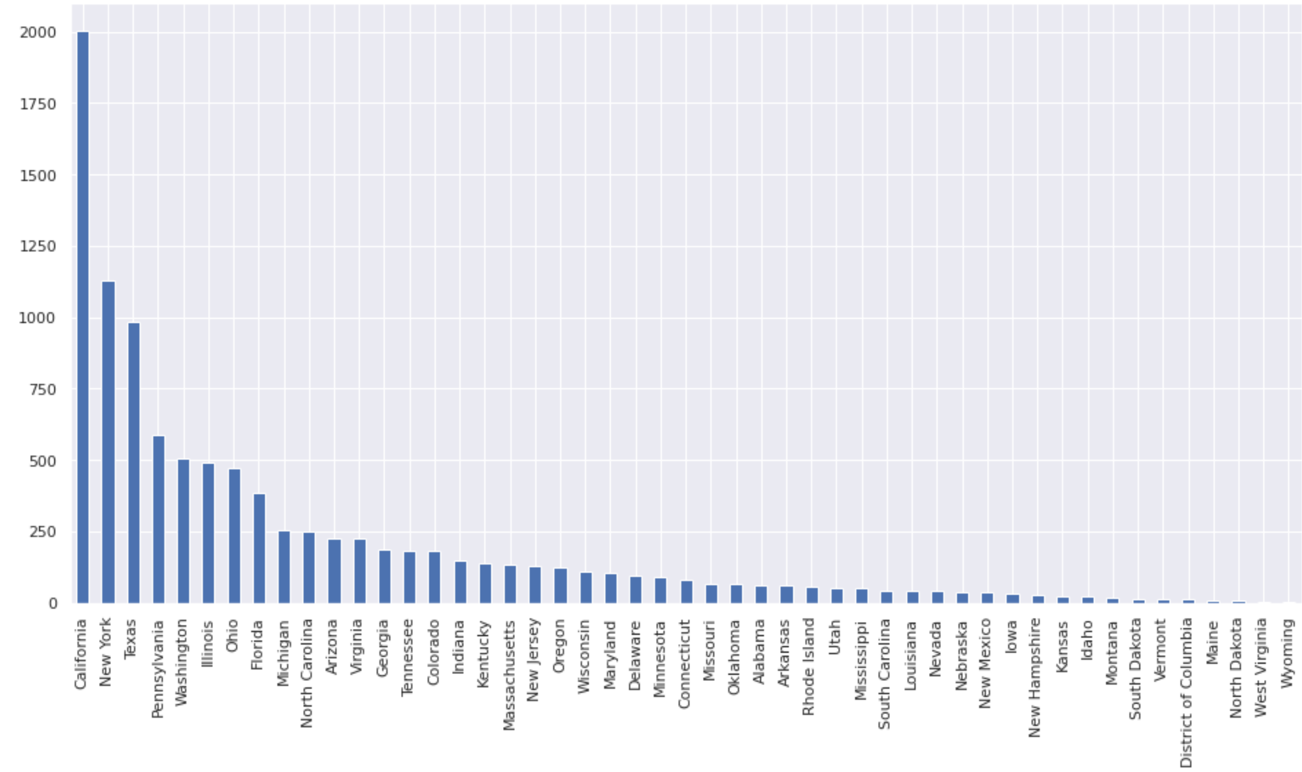


NEXT ARE A SERIES OF COUNT GRAPHS THAT IS HOW MANY DATA DO WE HAVE IF THE VALUES OF A PARTICULAR COLUMN IS TO BE CONSIDERED THE DIFFERNTIATING FACTOR

```
plt.figure(figsize=(16,8))
```

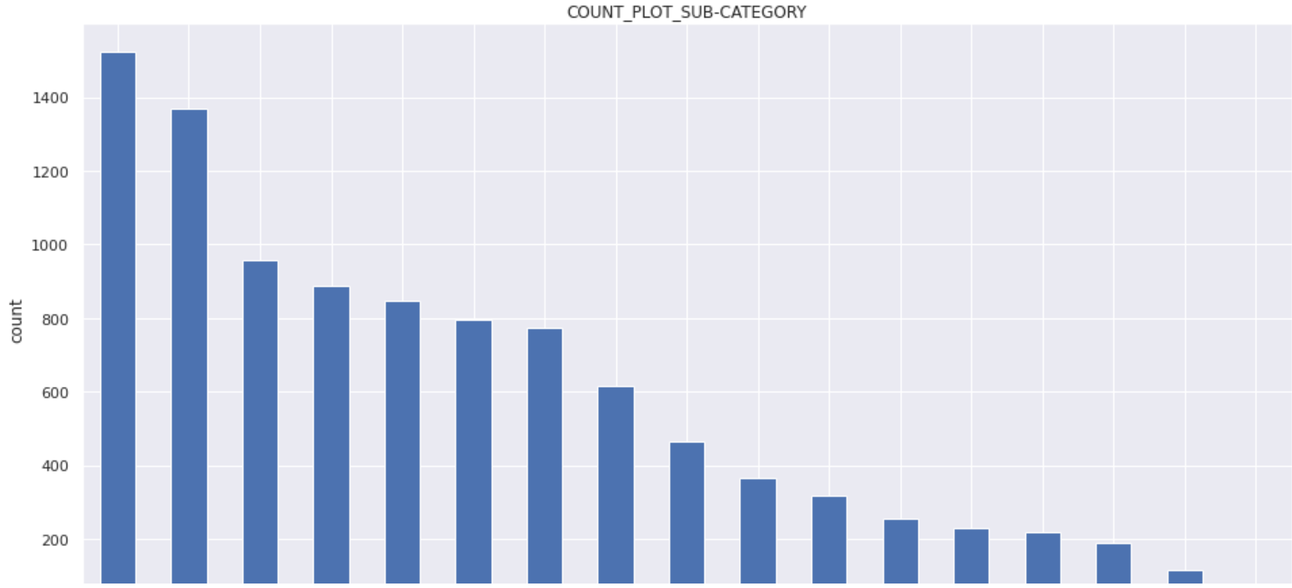
```
data['State'].value_counts().plot(kind='bar')
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f199bdc76d0>



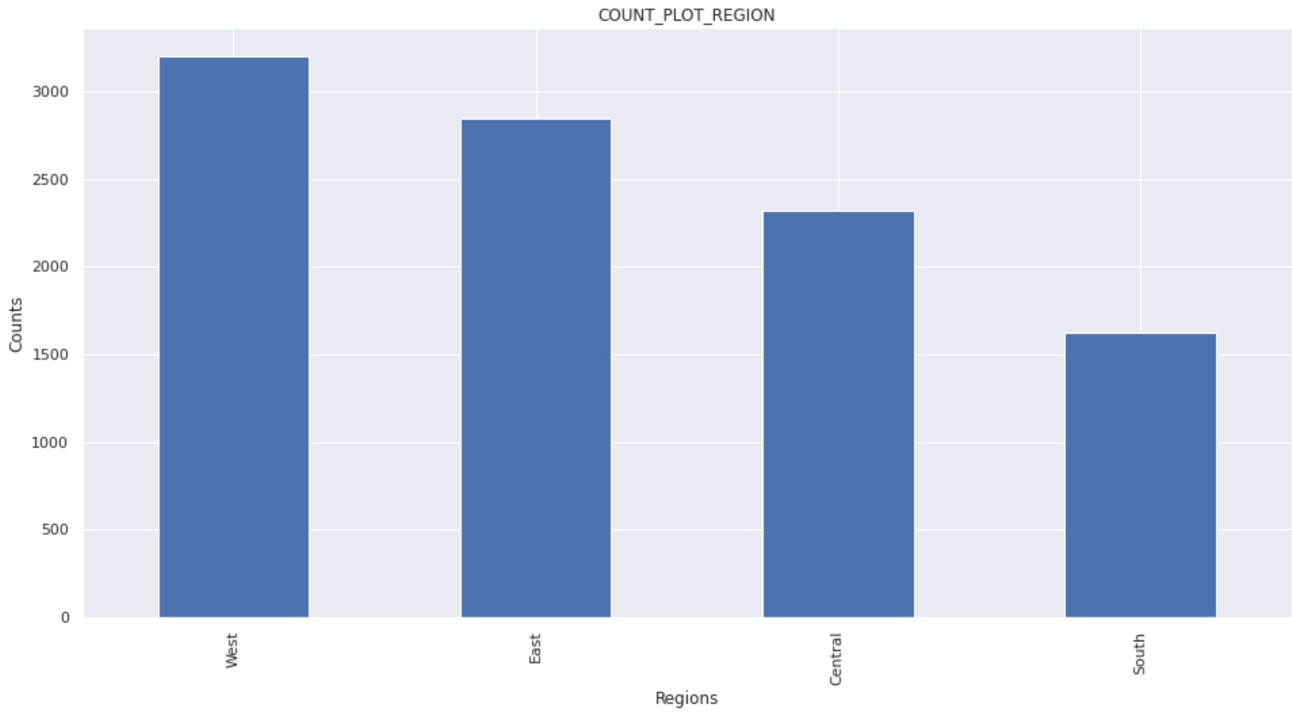

```
plt.figure(figsize=(16,8))
data['Sub-Category'].value_counts().plot(kind='bar')
plt.title("COUNT_PLOT_SUB-CATEGORY")
plt.xlabel("Sub-catrgories")
plt.ylabel("count")
```

```
Text(0, 0.5, 'count')
```




```
plt.figure(figsize=(16,8))
data['Region'].value_counts().plot(kind='bar')
plt.title("COUNT_PLOT_REGION")
plt.xlabel("Regions")
plt.ylabel("Counts")
```

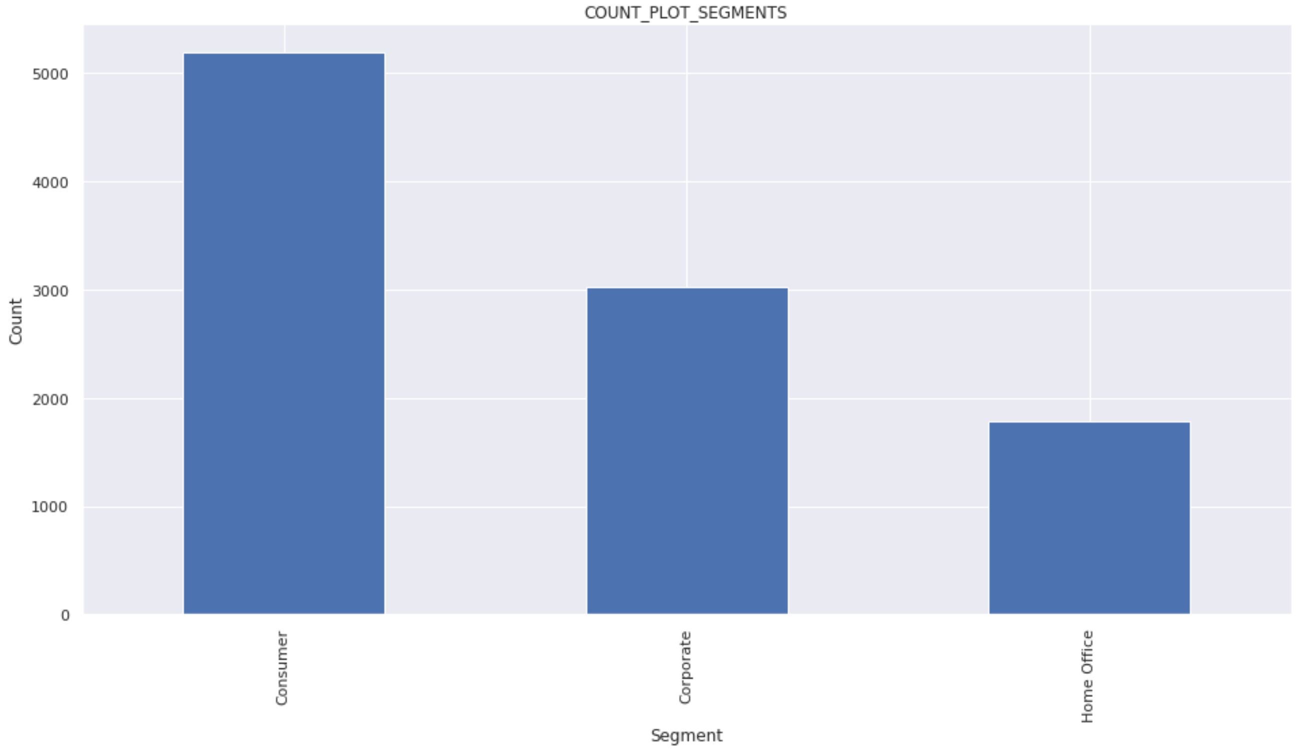
```
Text(0, 0.5, 'Counts')
```



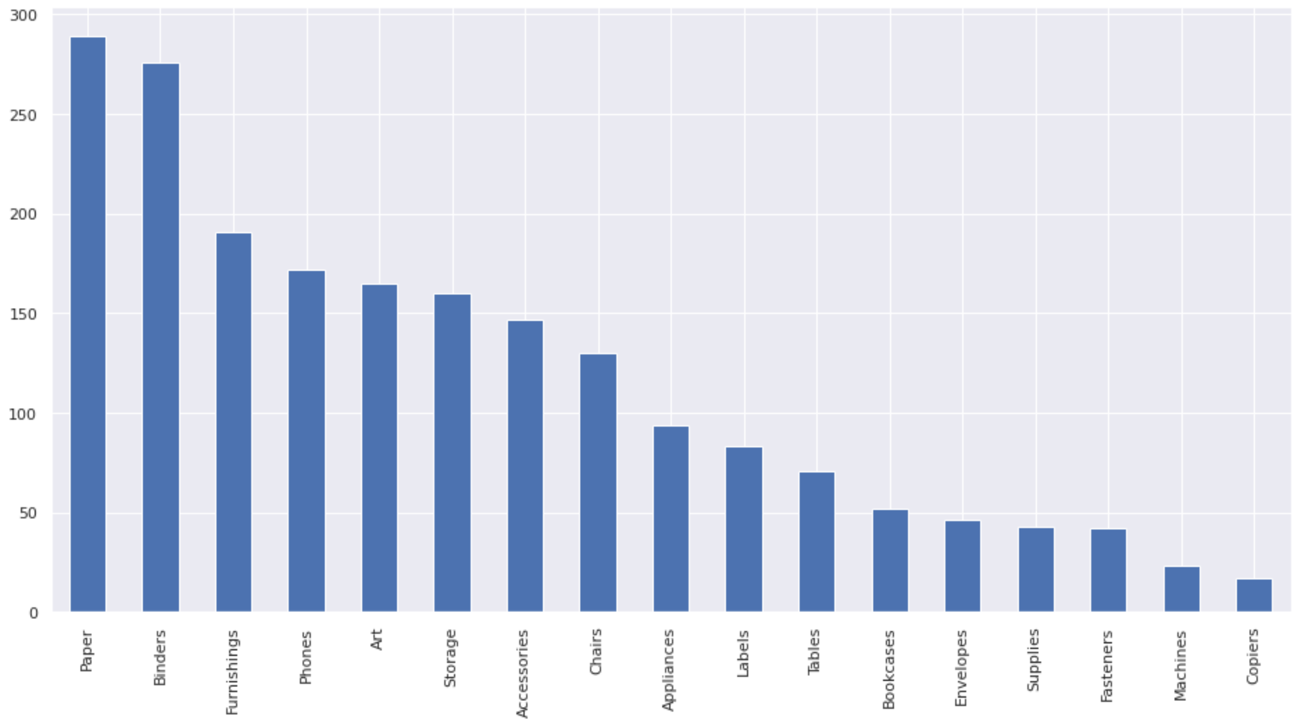

```
plt.figure(figsize=(16,8))
```

```
data['Segment'].value_counts().plot(kind='bar')
plt.title("COUNT_PLOT_SEGMENTS")
plt.xlabel("Segment")
plt.ylabel("Count")
```

```
Text(0, 0.5, 'Count')
```

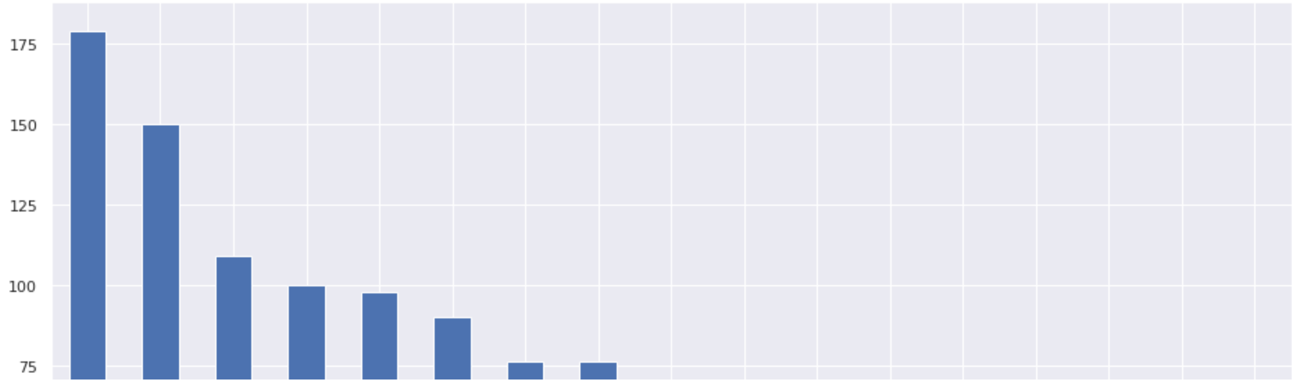



```
d_california=data[data["State"]=="California"]
plt.figure(figsize=(16,8))
d_california['Sub-Category'].value_counts().plot(kind='bar')
plt.show()
print(d_california.head())
```

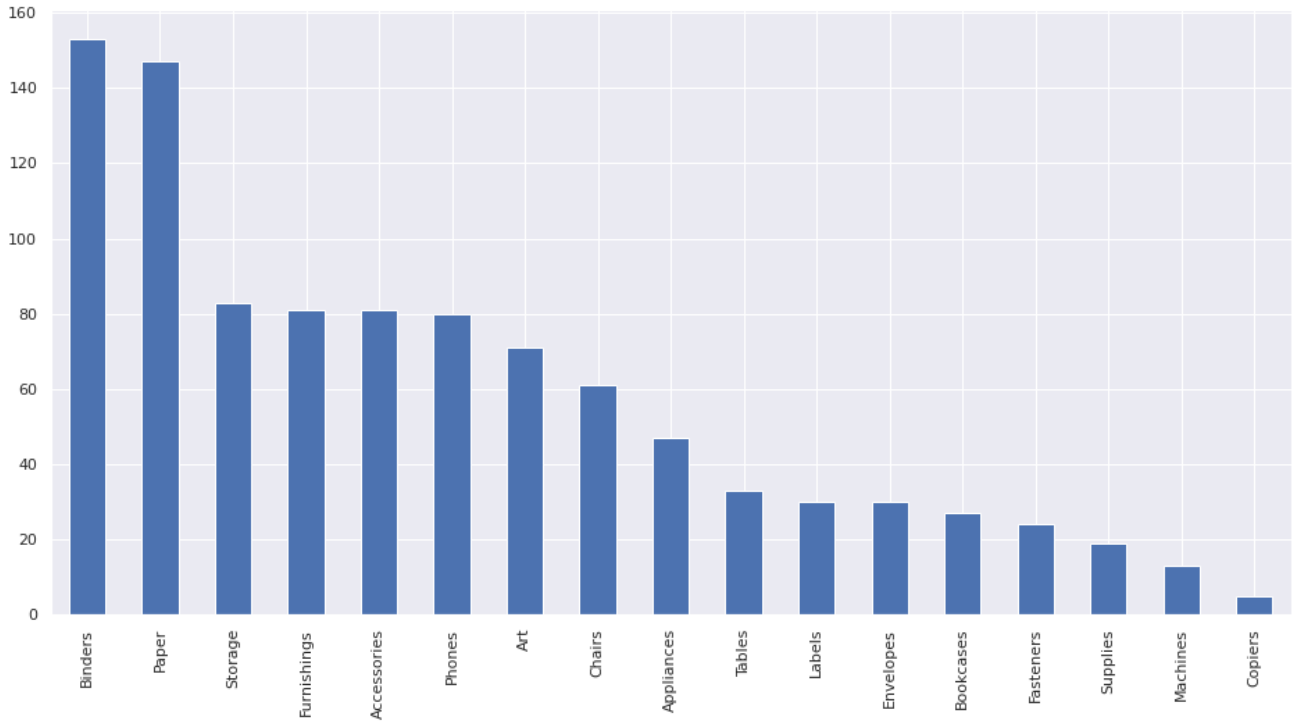



```
        Ship Mode    Segment        Country  ... Quantity Discount   Profit
2    Second Class  Corporate  United States  ...        2      0.0   6.8714
5  Standard Class   Consumer  United States  ...        7      0.0  14.1694
6  Standard Class   Consumer  United States  ...        4      0.0   1.9656
```

```
d_ny=data[data["State"]=="New York"]
plt.figure(figsize=(16,8))
d_ny['Sub-Category'].value_counts().plot(kind='bar')
plt.show()
print(d_ny.head())
```




```
d_texas=data[data["State"]=="Texas"]
plt.figure(figsize=(16,8))
d_texas['Sub-Category'].value_counts().plot(kind='bar')
plt.show()
print(d_texas.head())
```




```
         Ship Mode      Segment        Country  ... Quantity Discount     Profit
14  Standard Class  Home Office  United States  ...        5      0.8 -123.8580
15  Standard Class  Home Office  United States  ...        3      0.8   -3.8160
34    Second Class  Home Office  United States  ...        3      0.2    9.9468
35     First Class    Corporate  United States  ...        7      0.2  123.4737
36     First Class    Corporate  United States  ...        5      0.6 -147.9630

[5 rows x 13 columns]
```

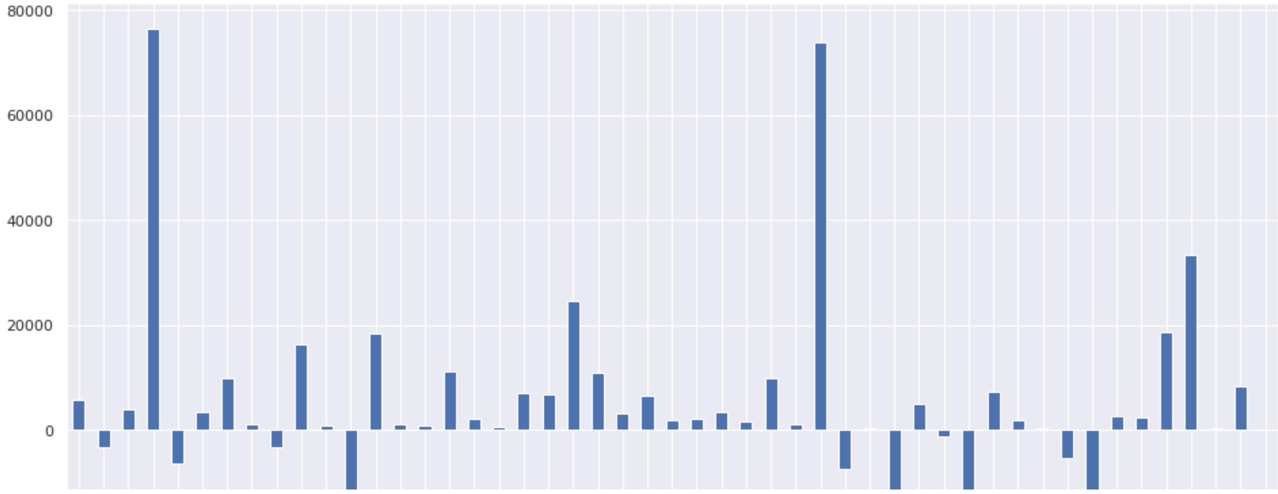
HEAT MAP ,TO UNDERSTAND THE CORELATION BETWEEN DIFFERENT FEATURES

```
fig,axes = plt.subplots(1,1,figsize=(16,8))
sns.heatmap(data.corr(), annot= True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f199ada5510>
```


```
plt.figure(figsize=(16,8))
d_profit=data.groupby('State')['Profit']
d_profit.sum().plot(kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f1996407ed0>
```



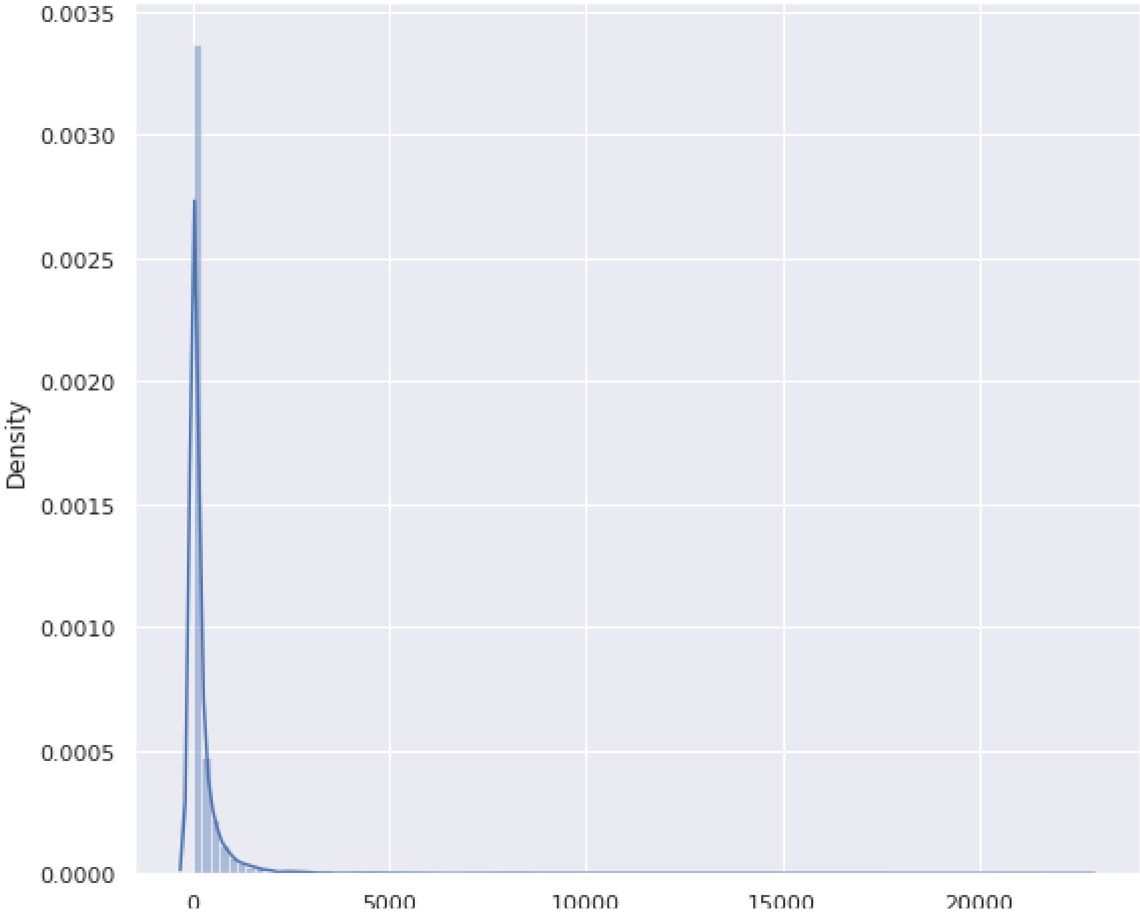
SCATTER PLOT

```
fig, ax = plt.subplots(figsize = (10 , 6))
ax.scatter(data["Sales"] , data["Profit"])
ax.set_xlabel('Sales')
ax.set_ylabel('Profit')
plt.show()
```

DISTRIBUTION PLOT

```
print(data['Sales'].describe())
plt.figure(figsize = (9 , 8))
sns.distplot(data['Sales'], color = 'b', bins = 100, hist_kws = {'alpha': 0.4});
```

```
count     9994.000000
mean       229.858001
std        623.245101
min          0.444000
25%         17.280000
50%         54.490000
75%        209.940000
max      22638.480000
Name: Sales, dtype: float64
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2557: FutureWarning:
  warnings.warn(msg, FutureWarning)
```



✓ 0s completed at 8:11 PM