Task:Perform 'Exploratory Data Analysis' on the provided dataset 'SampleSuperstore'

In [2]:
```
#Import required libraries
import io
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [3]:
```
#Load Datasets
df = pd.read_csv(r'Downloads\SampleSuperstore.csv')
df.head()
```

Out[3]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Su
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14.6
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.3

In [4]:
```
#To check shape of data
df.shape
```

Out[4]: (9994, 13)

In [5]:
```
#To remove Country coloumn
df1 = df.drop(['Country'], axis=1)
```

In [6]:
```
#After removing Country
df1.head()
```

Out[6]:

	Ship Mode	Segment	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quan
0	Second Class	Consumer	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	
1	Second Class	Consumer	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9400	
2	Second Class	Corporate	Los Angeles	California	90036	West	Office Supplies	Labels	14.6200	
3	Standard Class	Consumer	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5775	
4	Standard Class	Consumer	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.3680	

In [7]:
```
#To check null values
df1.isnull().sum()
```

Out[7]:
```
Ship Mode      0
Segment        0
City           0
State          0
Postal Code    0
Region         0
Category       0
Sub-Category   0
Sales          0
Quantity       0
Discount       0
Profit         0
dtype: int64
```

In [8]:
```
#Information about data
df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   Ship Mode     9994 non-null   object
 1   Segment       9994 non-null   object
 2   City          9994 non-null   object
 3   State         9994 non-null   object
 4   Postal Code   9994 non-null   int64
 5   Region        9994 non-null   object
 6   Category      9994 non-null   object
 7   Sub-Category  9994 non-null   object
 8   Sales         9994 non-null   float64
 9   Quantity      9994 non-null   int64
 10  Discount      9994 non-null   float64
 11  Profit        9994 non-null   float64
dtypes: float64(3), int64(2), object(7)
memory usage: 937.1+ KB
```

In [9]:
```
#To describe data
df1.describe ()
```

Out[9]:

	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

In [10]:
```
#To check Ship mode types
df1['Ship Mode'].value_counts()
```

Out[10]:
```
Standard Class    5968
Second Class      1945
First Class       1538
Same Day           543
Name: Ship Mode, dtype: int64
```

In [11]:
```
#To make pie chart of ship mode
df1['Ship Mode'].value_counts().plot.pie(autopct="%1.1f%%")
```

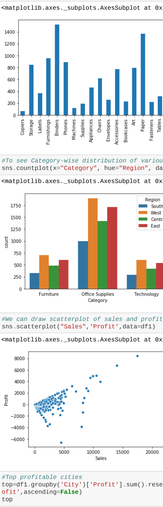
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x185e1e70688>

In [12]:
```
#To see sub-category wise distribution
df1['Sub-Category'].value_counts(sort=False)
```

Out[12]:
```
Copiers          68
Storage         846
Labels          364
Furnishings     957
Binders        1523
Phones          889
Machines        115
Supplies        190
Appliances      466
Chairs          617
Envelopes       254
Accessories     775
Bookcases       228
Art             796
Paper          1370
Fasteners       217
Tables          319
Name: Sub-Category, dtype: int64
```

In [13]:
```
df1['Sub-Category'].value_counts(sort=False).plot(kind="bar")
```

Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x185e41c4288>

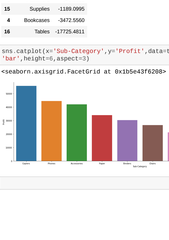


In [14]:
```
#To see Category-wise distribution of various regions
sns.countplot(x="Category", hue="Region", data=df1)
```

Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x185e4267b88>

In [15]:
```
#We can draw scatterplot of sales and profit
sns.scatterplot("Sales",'Profit',data=df1)
```

Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x185e432ba08>



In [16]:
```
#Top profitable cities
top=df1.groupby('City')['Profit'].sum().reset_index().sort_values(by='Profit',ascending=False)
top
```

Out[16]:

	City	Profit
329	New York City	62036.9837
266	Los Angeles	30440.7579
452	Seattle	29156.0967
438	San Francisco	17507.3854
123	Detroit	13181.7908
...	...	...
80	Chicago	-6654.5688
241	Lancaster	-7239.0684
434	San Antonio	-7299.0502
207	Houston	-10153.5485
374	Philadelphia	-13837.7674

531 rows × 2 columns

In [17]:
```
sns.catplot('City','Profit',data=top.head(10),kind='bar',height=6,aspect=3)
plt.show()
```


In [18]:
```
#Top Profitable Sub-Category
top_category = df1.groupby('Sub-Category')['Profit'].sum().reset_index()
.sort_values(by='Profit',ascending=False)
top_category
```

Out[18]:

	Sub-Category	Profit
6	Copiers	55617.8249
13	Phones	44515.7306
0	Accessories	41936.6357
12	Paper	34053.5693
3	Binders	30221.7633
5	Chairs	26590.1663
14	Storage	21278.8264
1	Appliances	18138.0054
9	Furnishings	13059.1436
7	Envelopes	6964.1767
2	Art	6527.7870
10	Labels	5546.2540
11	Machines	3384.7569
8	Fasteners	949.5182
15	Supplies	-1189.0995
4	Bookcases	-3472.5560
16	Tables	-17725.4811

In [20]:
```
sns.catplot(x='Sub-Category',y='Profit',data=top_category.head(10),kind='bar',height=6,aspect=3)
```

Out[20]: <seaborn.axisgrid.FacetGrid at 0x1b5e43f6208>

In [ ]:
```
```