TASK 3 : Exploratory data analysis - retail

problem statement: perform 'exploratory data analysis on dataset 'SampleSuperstore'.This task is about exploratory data analysis -retail.where the task focuses on a business manager who will try to find out weak areas where he can work to make profit

In [61]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')
```

In [62]:
```
df=pd.read_csv(r'C:\Users\modil\Downloads\SampleSuperstore.csv')
```

EXPLORING DATA AND BASIC INSIGHT

In [63]:
```
df.head()
```

Out[63]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	0.00	41.9136
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9400	3	0.00	219.5820
2	Second Class	Consumer	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14.6200	2	0.00	6.8714
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5775	5	0.45	-383.0310
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.3680	2	0.20	2.5164

In [64]:
```
df.tail()
```

Out[64]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
9989	Second Class	Consumer	United States	Miami	Florida	33180	South	Furniture	Furnishings	25.248	3	0.2	4.1028
9990	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Furniture	Furnishings	91.960	2	0.0	15.6332
9991	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Technology	Phones	258.576	2	0.2	19.3932
9992	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Office Supplies	Paper	29.600	4	0.0	13.3200
9993	Second Class	Consumer	United States	Westminster	California	92683	West	Office Supplies	Appliances	243.160	2	0.0	72.9480

In [65]:
```
df.shape
```

Out[65]:
```
(9994, 13)
```

In [66]:
```
df.describe()
```

Out[66]:

	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 [67]:
```
df.isnull().sum()
```

Out[67]:
```
Ship Mode       0
Segment         0
Country         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 [68]:
```
df.info()
```

Out[68]:
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 13 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Ship Mode     9994 non-null   object 
 1   Segment       9994 non-null   object 
 2   Country       9994 non-null   object 
 3   City          9994 non-null   object 
 4   State         9994 non-null   object 
 5   Postal Code   9994 non-null   int64  
 6   Region        9994 non-null   object 
 7   Category      9994 non-null   object 
 8   Sub-Category  9994 non-null   object 
 9   Sales         9994 non-null   float64
 10  Quantity      9994 non-null   int64  
 11  Discount      9994 non-null   float64
 12  Profit        9994 non-null   float64
dtypes: float64(3), int64(2), object(8)
memory usage: 1015.1+ KB
```

In [69]:
```
df.columns
```

Out[69]:
```
Index(['Ship Mode', 'Segment', 'Country', 'City', 'State', 'Postal Code',
       'Region', 'Category', 'Sub-Category', 'Sales', 'Quantity', 'Discount',
       'Profit'],
      dtype='object')
```

In [70]:
```
df.duplicated().sum()
```

Out[70]:
```
17
```

In [71]:
```
df.nunique()
```

Out[71]:
```
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
```

In [72]:
```
df.drop('Postal Code',axis=1,inplace=True)
```

In [73]:
```
df.head()
```

Out[73]:

	Ship Mode	Segment	Country	City	State	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
0	Second Class	Consumer	United States	Henderson	Kentucky	South	Furniture	Bookcases	261.9600	2	0.00	41.9136
1	Second Class	Consumer	United States	Henderson	Kentucky	South	Furniture	Chairs	731.9400	3	0.00	219.5820
2	Second Class	Consumer	United States	Los Angeles	California	West	Office Supplies	Labels	14.6200	2	0.00	6.8714
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	South	Furniture	Tables	957.5775	5	0.45	-383.0310
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	South	Office Supplies	Storage	22.3680	2	0.20	2.5164

In [74]:
```
df['Ship Mode'].value_counts()
```

Out[74]:
```
Standard Class    5968
Second Class      1945
First Class       1538
Same Day           543
Name: Ship Mode, dtype: int64
```

In [75]:
```
df['Segment'].value_counts()
```

Out[75]:
```
Consumer      5191
Corporate     3020
Home Office   1783
Name: Segment, dtype: int64
```

In [76]:
```
df['Region'].value_counts()
```

Out[76]:
```
West     3203
East     2848
Central  2323
South    1620
Name: Region, dtype: int64
```

In [77]:
```
df['Category'].value_counts()
```

Out[77]:
```
Office Supplies    6026
Furniture          2121
Technology         1847
Name: Category, dtype: int64
```

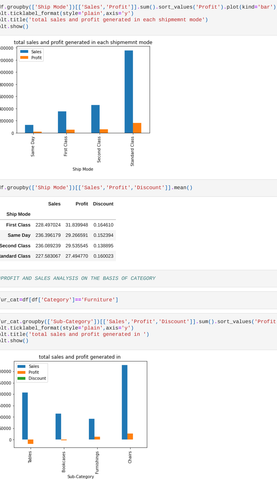
In [78]:
```
df['Sub-Category'].value_counts()
```

Out[78]:
```
Binders        1523
Paper          1370
Furnishings     957
Phones          889
Storage         846
Art             796
Accessories     775
Chairs          617
Appliances      466
Labels          364
Tables          319
Envelopes       254
Bookcases       228
Fasteners       217
Supplies        190
Machines        115
Copiers          68
Name: Sub-Category, dtype: int64
```

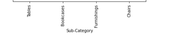
VISUALIZATION

In [79]:
```
sns.pairplot(df)
```

Out[79]:
```
<seaborn.axisgrid.PairGrid at 0x1e15ec03910>
```



In [80]:
```
plt.figure(figsize=(5,6))
sns.heatmap(df.corr(),annot=True)
plt.show()
```



There is a positive correlation between sales and profit(sales increases profit increases) There is a positive correlation between quantity and profit(quality increases profit increases) There is a negative correlation between profit and discount(discount increases profit decreases) There is a negative correlation between sales and discount(sales increases discount decreases) There is nearly no correlation between quantity and discount

In [ ]:
```
#PROFIT AND SALES ANALYSIS ON THE SHIPMENT MODE
```

In [81]:
```
plt.figure
sns.countplot(data=df,x='Ship Mode')
plt.title('number of sales in each shipmement mode')
plt.show()
```


In [82]:
```
df.groupby(['Ship Mode'])[['Sales','Profit']].sum().sort_values('Profit').plot(kind='bar')
plt.ticklabel_format(style='plain',axis='y')
plt.title('total sales and profit generated in each shipmement mode')
plt.show()
```


In [83]:
```
df.groupby(['Ship Mode'])[['Sales','Profit','Discount']].mean()
```

Out[83]:

Ship Mode	Sales	Profit	Discount
First Class	228.497024	31.839948	0.164410
Same Day	236.396179	29.266591	0.152394
Second Class	236.089239	29.535545	0.138895
Standard Class	227.583067	27.494770	0.160023

In [ ]:
```
#PROFIT AND SALES ANALYSIS ON THE BASIS OF CATEGORY
```

In [84]:
```
fur_cat=df[df['Category']=='Furniture']
```

In [85]:
```
fur_cat.groupby(['Sub-Category'])[['Sales','Profit','Discount']].sum().sort_values('Profit').plot(kind='bar')
plt.ticklabel_format(style='plain',axis='y')
plt.title('total sales and profit generated in ')
plt.show()
```