Retail Superstore

A retail sale occurs when a business sells a product or service to an individual consumer for his or her own use. The transaction itself can occur through a number of different sales channels, such as online, in a brick-and-mortar storefront, through direct sales, or direct mail. The aspect of the sale that qualifies it as a retail transaction is that the end user is the buyer.

Problem Statement

1.As a business manager, try to find out the weak areas where you can work to make more profit.

2.What all business problems you can derive by exploring the data?

'Exploratory Data Analysis' on dataset 'SampleSuperstore'

In [1]:
```
import numpy as np # for numeric data
import pandas as pd # for dataframe and analysis
import matplotlib.pyplot as plt # for plots
import seaborn as sns #for Advance plots
import warnings
```

In [2]:
```
df = pd.read_csv("SampleSuperstore.csv")
df.head()
```

Out[2]:

	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	Corporate	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 [3]:
```
df.shape
```

Out[3]:
```
(9994, 13)
```

In [4]:
```
#There are total 13 attributes about orders at superstore.
```

In [5]:
```
df.columns
```

Out[5]:
```
Index(['Ship Mode', 'Segment', 'Country', 'City', 'State', 'Postal Code',
       'Region', 'Category', 'Sub-Category', 'Sales', 'Quantity', 'Discount',
       'Profit'],
      dtype='object')
```

In [6]:
```
df.info()
```

```
<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 [7]:
```
df.isna().sum()
```

Out[7]:
```
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 [8]:
```
df.describe()
```

Out[8]:

	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

OBSERVATIONS

Retail Superstore is making average profit of 28.65 overall on each order.

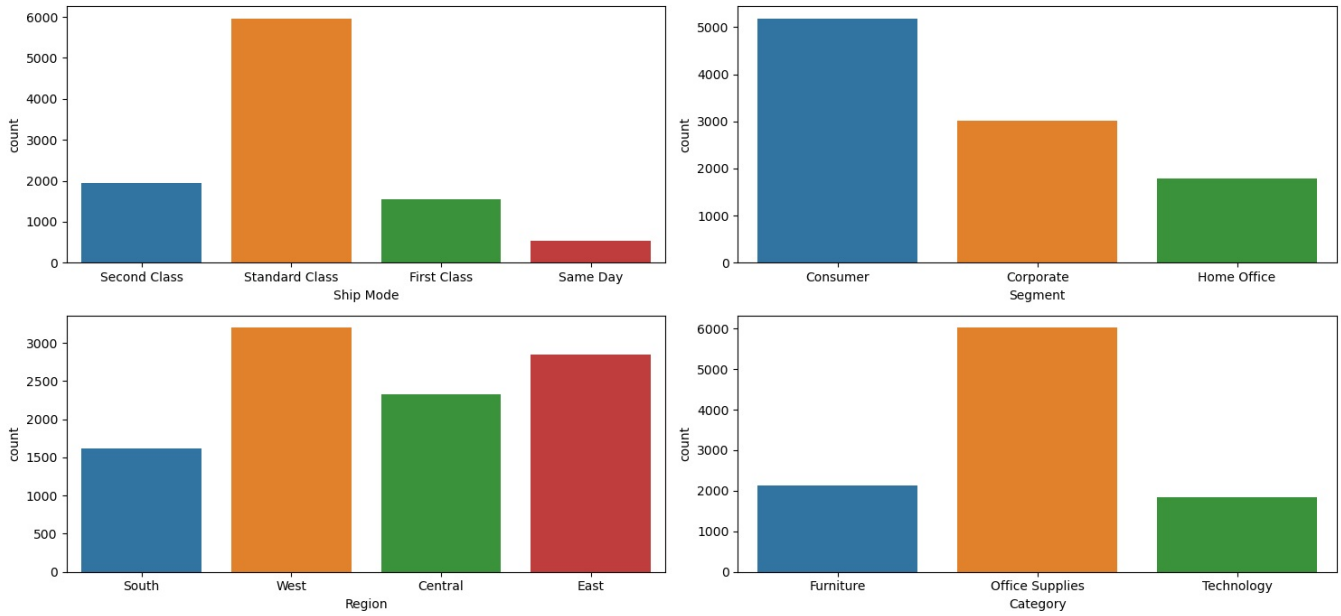
Maximum loss incured is of around 6599.97 and profit of around 8399.97.

On an average superstore is doing sales of around 229 per order.

Max discount offered by superstore ia around 80%.

Univariate Analysis

In [11]:
```
fig, axis = plt.subplots(nrows = 2 , ncols = 2 , figsize = (15 , 7))
sns.countplot(data = df , x = df["Ship Mode"] , ax = axis[0 , 0])
sns.countplot(data = df , x = df["Segment"] , ax = axis[0 , 1])
sns.countplot(data = df , x = df["Region"] , ax = axis[1, 0])
sns.countplot(data = df , x = df["Category"] , ax = axis[1, 1])
plt.tight_layout()
plt.show()
```



OBSERVATIONS

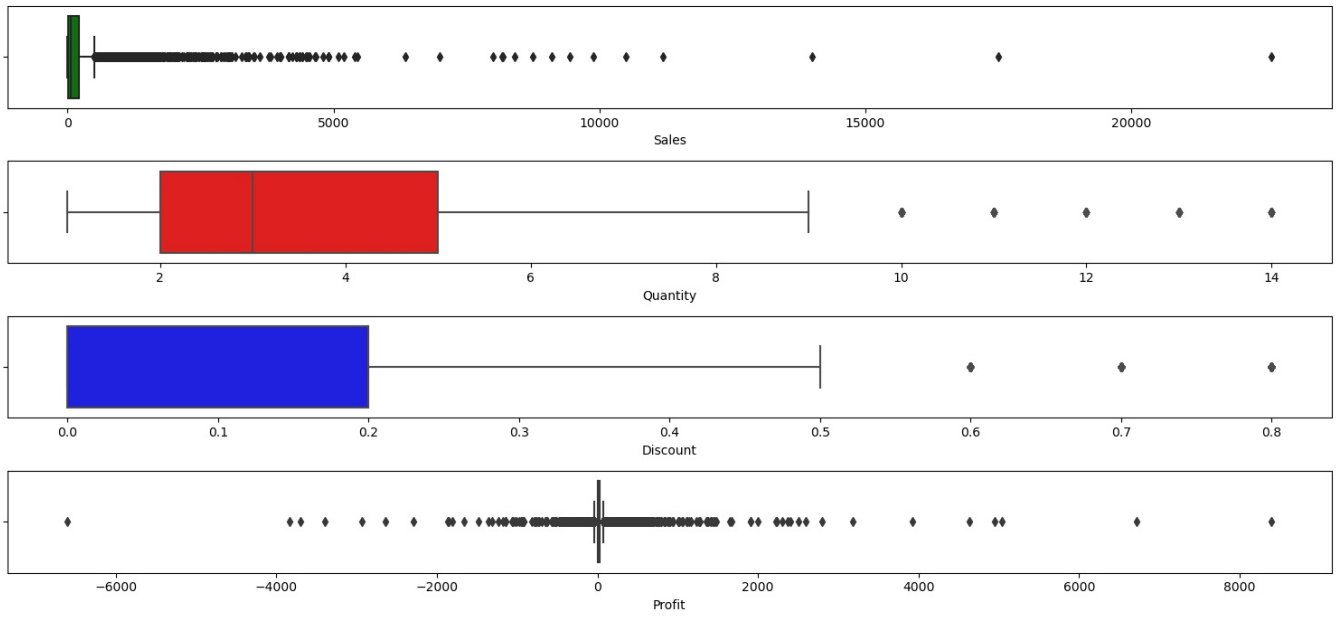
Most of the orders have ship mode as Standard Class and least for the Same Day.

Majority of the orders are Consumer segment while least for Home Office.

Highest number of orders are from West and East Region while least from South.

Majorily the category of order are of Office supplies as compared to other categories

In [13]:
```
fig, axis = plt.subplots(nrows = 4 , ncols = 1 , figsize = (15 , 7))
sns.boxplot(data = df , x = df["Sales"] , ax = axis[0],color = "g")
sns.boxplot(data = df , x = df["Quantity"] , ax = axis[1],color = "r")
sns.boxplot(data = df , x = df["Discount"] , ax = axis[2],color = "b")
sns.boxplot(data = df , x = df["Profit"] , ax = axis[3],color = "y")
plt.tight_layout()
plt.show()
```



OBSERVATIONS

There are many outliers in Sales attribute and all of them are in positive side i.e higher side.

Majority of the Quantity lies between 3 to 9 orders with some outliers on higher side.

Discount majorily ranges from 0 to 50% with 60%,70% and 80% as an outlier.

Profit has highest number of outliers and seems to incur losses too.

Bi-Variate Analysis

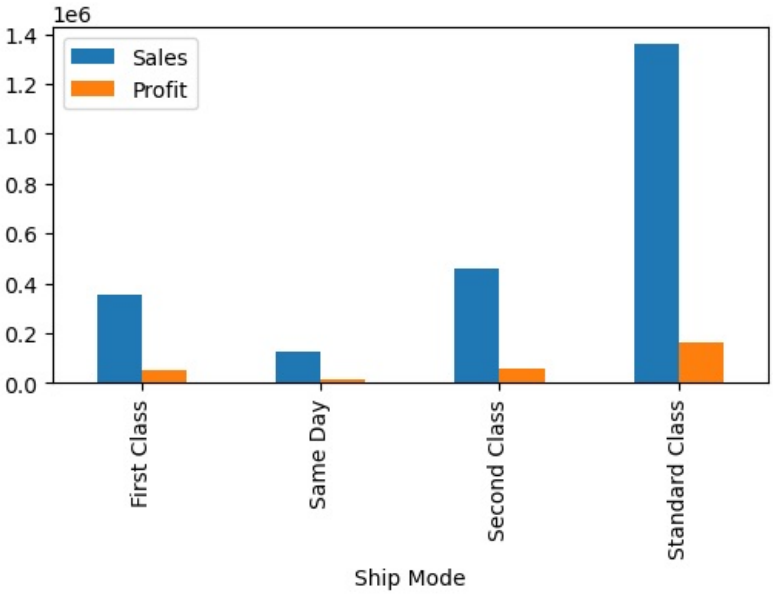
In [16]:
```
df.head()
```

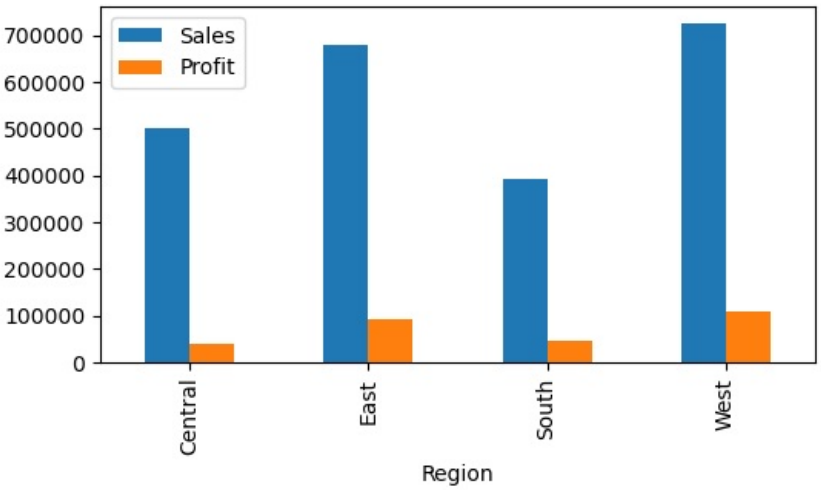
Out[16]:

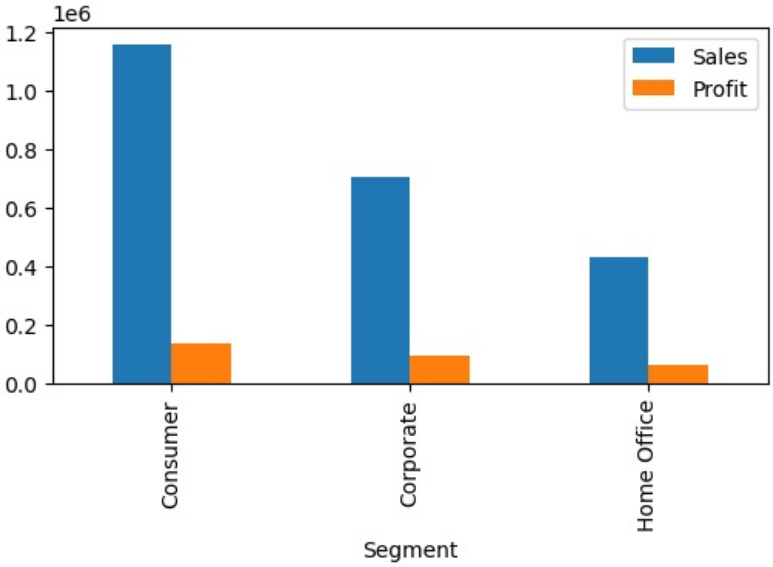
	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	0.00	41.9136
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9400	3	0.00	219.5820
2	Second Class	Corporate	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

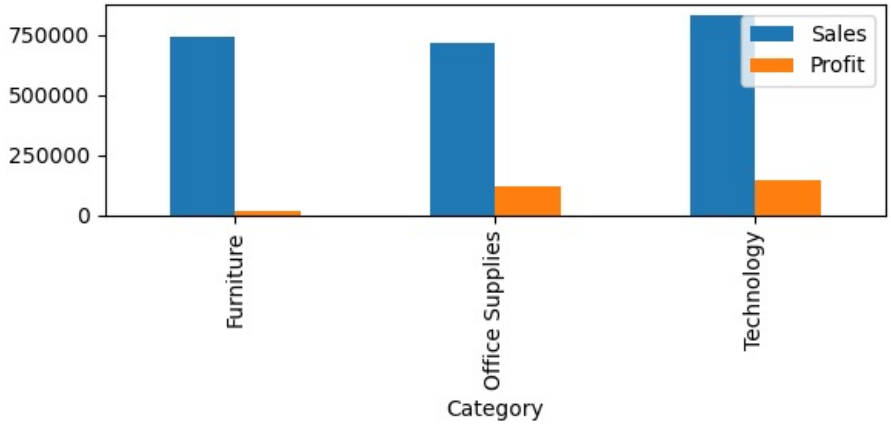
In [17]:
```
pd.DataFrame(df.groupby('Ship Mode').sum()[['Sales','Profit']]).plot(kind='bar',figsize=(6,3))
pd.DataFrame(df.groupby('Region').sum()[['Sales','Profit']]).plot(kind='bar',figsize=(6,3))
pd.DataFrame(df.groupby('Segment').sum()[['Sales','Profit']]).plot(kind='bar',figsize=(6,3))
pd.DataFrame(df.groupby('Category').sum()[['Sales','Profit']]).plot(kind='bar',figsize=(6,3))
plt.tight_layout()
plt.show()
```

```
C:\Users\91988\AppData\Local\Temp\ipykernel_11984\472609435.py:1: FutureWarning: The default value of numeric_o
nly in DataFrameGroupBy.sum is deprecated. In a future version, numeric_only will default to False. Either spec
ify numeric_only or select only columns which should be valid for the function.
  pd.DataFrame(df.groupby('Ship Mode').sum()[['Sales','Profit']]).plot(kind='bar',figsize=(6,3))
C:\Users\91988\AppData\Local\Temp\ipykernel_11984\472609435.py:2: FutureWarning: The default value of numeric_o
nly in DataFrameGroupBy.sum is deprecated. In a future version, numeric_only will default to False. Either spec
ify numeric_only or select only columns which should be valid for the function.
  pd.DataFrame(df.groupby('Region').sum()[['Sales','Profit']]).plot(kind='bar',figsize=(6,3))
C:\Users\91988\AppData\Local\Temp\ipykernel_11984\472609435.py:3: FutureWarning: The default value of numeric_o
nly in DataFrameGroupBy.sum is deprecated. In a future version, numeric_only will default to False. Either spec
ify numeric_only or select only columns which should be valid for the function.
  pd.DataFrame(df.groupby('Segment').sum()[['Sales','Profit']]).plot(kind='bar',figsize=(6,3))
C:\Users\91988\AppData\Local\Temp\ipykernel_11984\472609435.py:4: FutureWarning: The default value of numeric_o
nly in DataFrameGroupBy.sum is deprecated. In a future version, numeric_only will default to False. Either spec
ify numeric_only or select only columns which should be valid for the function.
  pd.DataFrame(df.groupby('Category').sum()[['Sales','Profit']]).plot(kind='bar',figsize=(6,3))
```


1e6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Sales
Profit
First Class
Same Day
Second Class
Standard Class
Ship Mode


700000
600000
500000
400000
300000
200000
100000
0
Sales
Profit
Central
East
South
West
Region


1e6
1.2
1.0
0.8
0.6
0.4
0.2
0.0
Sales
Profit
Consumer
Corporate
Home Office
Segment



OBSERVATIONS

The profit is far less as compared to the sales that have been made through Standard Class ship mode that infers that some good amount of losses have been incurred.

The sales are lowest from south region.

Profits in central region are not upto the mark as compared to other regions.

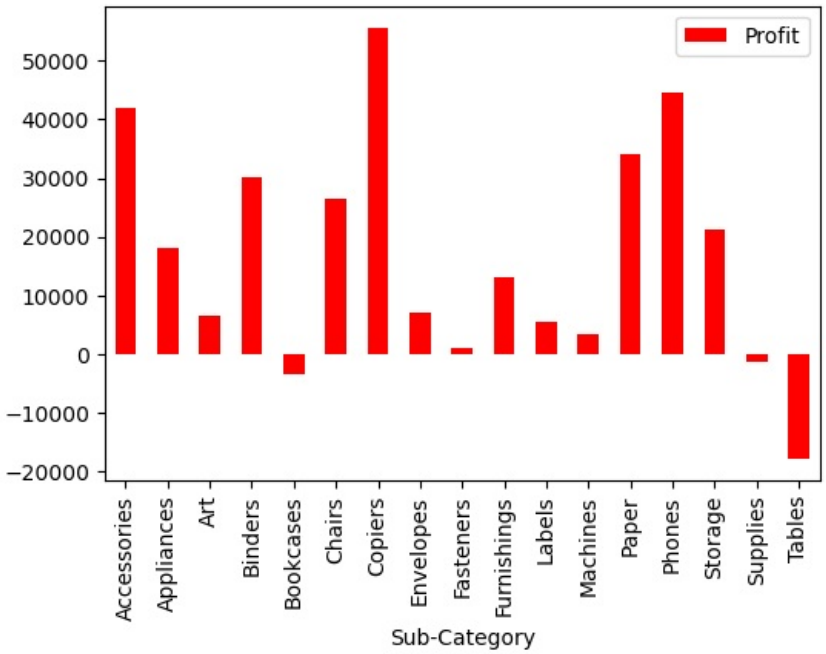
Sales and Profit both are highest for consumer segment and lowest for Home office.

Profit made in furniture category are extremely low as compared to other categories.

In [19]:
```
pd.DataFrame(df.groupby('Sub-Category').sum()[['Profit']]).plot(kind='bar',color='r',figsize=(6,4))
```

```
C:\Users\91988\AppData\Local\Temp\ipykernel_11984\3730481167.py:1: FutureWarning: The default value of numeric_
only in DataFrameGroupBy.sum is deprecated. In a future version, numeric_only will default to False. Either spe
cify numeric_only or select only columns which should be valid for the function.
  pd.DataFrame(df.groupby('Sub-Category').sum()[['Profit']]).plot(kind='bar',color='r',figsize=(6,4))
```

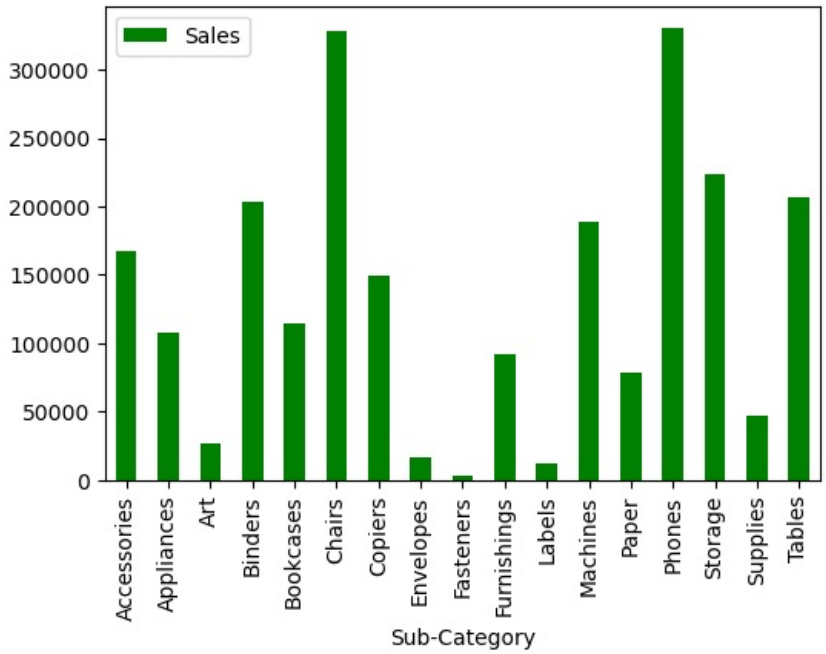
Out[19]:
```
<Axes: xlabel='Sub-Category'>
```



In [20]:
```
pd.DataFrame(df.groupby('Sub-Category').sum()[['Sales']]).plot(kind='bar',color = 'g',figsize=(6,4))
```

```
C:\Users\91988\AppData\Local\Temp\ipykernel_11984\3355148563.py:1: FutureWarning: The default value of numeric_
only in DataFrameGroupBy.sum is deprecated. In a future version, numeric_only will default to False. Either spe
cify numeric_only or select only columns which should be valid for the function.
  pd.DataFrame(df.groupby('Sub-Category').sum()[['Sales']]).plot(kind='bar',color = 'g',figsize=(6,4))
```

Out[20]:
```
<Axes: xlabel='Sub-Category'>
```



OBSERVATIONS

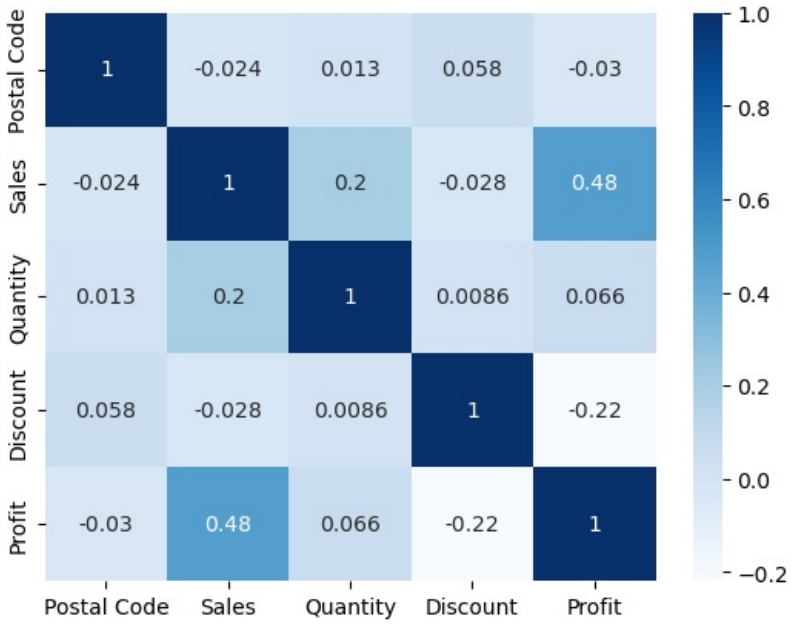
Most profitable sub category is copiers and most loss making is Tables.

Chairs and Phones are highest selling sub category while Fasteners,Labels,Envelopes have least sales.

In [22]:
```
sns.heatmap(df.corr(),annot=True,cmap='Blues')
```

```
C:\Users\91988\AppData\Local\Temp\ipykernel_11984\3170444083.py:1: FutureWarning: The default value of numeric_
only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid columns
or specify the value of numeric_only to silence this warning.
  sns.heatmap(df.corr(),annot=True,cmap='Blues')
```

Out[22]:
```
<Axes: >
```



In [23]:
```
#By Heatmap we confirm two obvious logical arguments:

#Discount is negatively correlated with profit.
#Sales has strong positive correlation with profit.
```

What impact discount is making on Sales?

From the heatmap we got a an interesting observation, Discount should have a positive impact on sales but on inferring it seem to not having any great impact on Sales infact it is having negative correlation that is a major problem.

In [25]:
```
df.head()
```

Out[25]:

	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	Corporate	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

Which state are on top and bottom in terms of profit?

In [26]:
```
df_state = pd.DataFrame(df.groupby('State').sum()[["Sales","Profit"]]).reset_index()
df_state.head()
```

```
C:\Users\91988\AppData\Local\Temp\ipykernel_11984\609953856.py:1: FutureWarning: The default value of numeric_o
nly in DataFrameGroupBy.sum is deprecated. In a future version, numeric_only will default to False. Either spec
ify numeric_only or select only columns which should be valid for the function.
  df_state = pd.DataFrame(df.groupby('State').sum()[["Sales","Profit"]]).reset_index()
```

Out[26]:

	State	Sales	Profit
0	Alabama	19510.6400	5786.8253
1	Arizona	35282.0010	-3427.9246
2	Arkansas	11678.1300	4008.6871
3	California	457687.6315	76381.3871
4	Colorado	32108.1180	-6527.8579

In [27]:
```
df_state.columns
```

Out[27]:
```
Index(['State', 'Sales', 'Profit'], dtype='object')
```

In [28]:
```
df_state[df_state['Profit']==max(df_state['Profit'])]
```

Out[28]:

	State	Sales	Profit
3	California	457687.6315	76381.3871

In [29]:
```
df_state[df_state['Profit']==min(df_state['Profit'])]
```

Out[29]:

	State	Sales	Profit
41	Texas	170188.0458	-25729.3563

Which state are top and bottom in terms of sales?

In [30]:
```
df_state[df_state['Sales']==max(df_state['Sales'])]
```

Out[30]:

	State	Sales	Profit
3	California	457687.6315	76381.3871

In [31]:
```
df_state[df_state['Sales']==min(df_state['Sales'])]
```

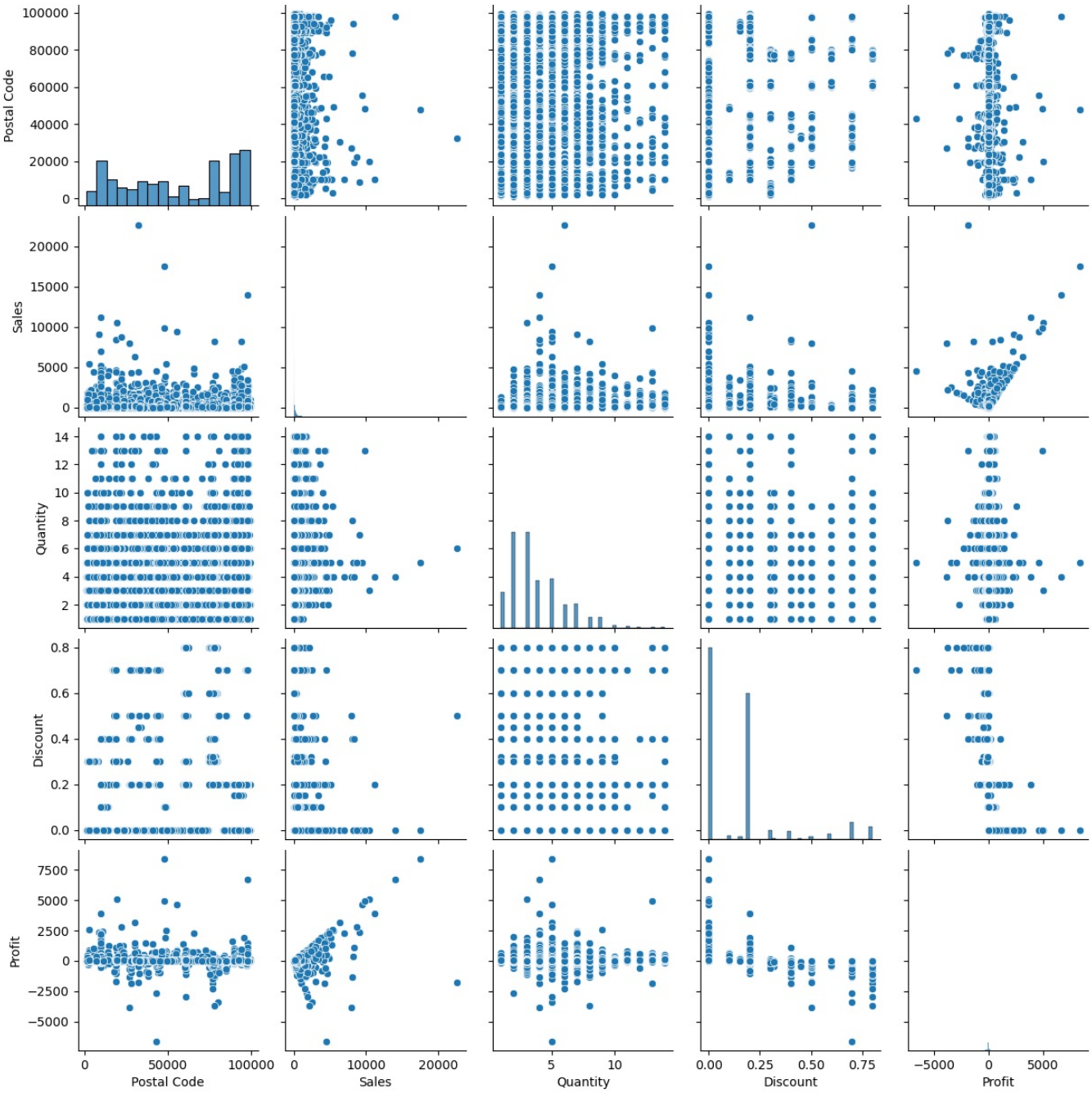
Out[31]:

	State	Sales	Profit
32	North Dakota	919.91	230.1497

State with highest sales is "California" with a sales of 457687. State with lowest sales is "North Dakota" with a loss of 919.

In [34]:
```
plt.figure(figsize=(25,20))
sns.pairplot(df)
plt.show()
```

```
<Figure size 2500x2000 with 0 Axes>
```



RECOMMENDATIONS

1.Superstore should focus more on improving sells in South region,South Dakota state and Texas state as by giving more discount on selected items that are more in demand in these regions and also improve delivery facilities.

2.We should either minimize selling or reduce discount on Furniture category as it is providing least profit especially Tables sub-category which is incurring huge losses.

3.The sales of Fasteners,Envelops,Labels and Art are extremely low, to improve this condition more discounted offers should be provided for a short interval of time to give boost to the sales.

4.The discount is observed to have negatively impacting sales somehow that is a major problem, planning for amount of discount on different products should be restructured such as the in demand products have a moderate discount while heavy discount offers should be provided for less sold products.

In [ ]: