The Sparks Foundation

GRIP JULY2021 Batch

Data Science and Business Analytics

Task3- Perform 'Expolatory Data Analysis' on dataset "SampleStore'

Author- Tanushree Gaur

Import all libraries

In [29]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

Importing and reeading data

In [30]:
```
df=pd.read_csv('C:\\Users\\tanu.Umeshchand\\Downloads\\SampleSuperstore.csv')
df.head()
```

Out[30]:

	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 [31]:
```
df.tail()
```

Out[31]:

	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 [10]:
```
df.shape
```

Out[10]:
```
(9994, 13)
```

In [32]:
```
df1=df.drop('Postal Code',axis=1)
df1.shape
```

Out[32]:
```
(9994, 12)
```

Checking Null values

In [33]:
```
df1.isnull().sum()
```

Out[33]:
```
Ship Mode       0
Segment         0
Country         0
City            0
State           0
Region          0
Category        0
Sub-Category    0
Sales           0
Quantity        0
Discount        0
Profit          0
dtype: int64
```

Exploring dataset

In [34]:
```
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 [35]:
```
df.duplicated().sum()
```

Out[35]:
```
17
```

Droping duplicates

In [38]:
```
df.drop_duplicates()
```

Out[38]:

	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9989	Second Class	Consumer	United States	Miami	Florida	33180	South	Furniture	Furnishings	25.2480	3	0.20	4.1028
9990	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Furniture	Furnishings	91.9600	2	0.00	15.6332
9991	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Technology	Phones	258.5760	2	0.20	19.3932
9992	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Office Supplies	Paper	29.6000	4	0.00	13.3200
9993	Second Class	Consumer	United States	Westminster	California	92683	West	Office Supplies	Appliances	243.1600	2	0.00	72.9480

9977 rows × 13 columns

In [40]:
```
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
```

Check for nullvalues

In [41]:
```
df1.isna().sum()
```

Out[41]:
```
Ship Mode       0
Segment         0
Country         0
City            0
State           0
Region          0
Category        0
Sub-Category    0
Sales           0
Quantity        0
Discount        0
Profit          0
dtype: int64
```

No null value, find the correlation

In [42]:
```
df1.corr()
```

Out[42]:

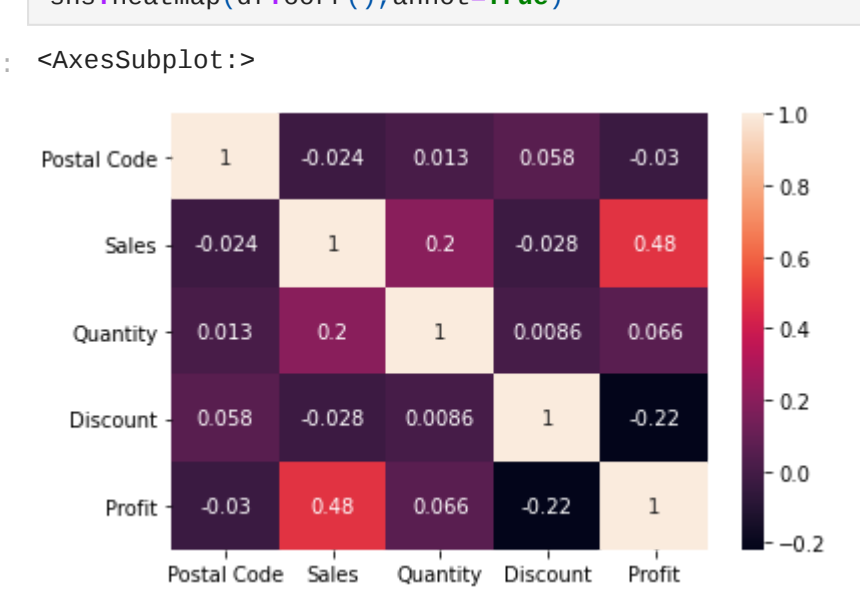
	Sales	Quantity	Discount	Profit
Sales	1.000000	0.200795	-0.028190	0.479064
Quantity	0.200795	1.000000	0.008623	0.066253
Discount	-0.028190	0.008623	1.000000	-0.219487
Profit	0.479064	0.066253	-0.219487	1.000000

Correlation is done by 5 columns as other columns contains non-numeric values

Visualising the dataset

In [45]:
```
#visualizing the correlation
import seaborn as sns
sns.heatmap(df.corr(),annot=True)
```

Out[45]:
```
<AxesSubplot:>
```

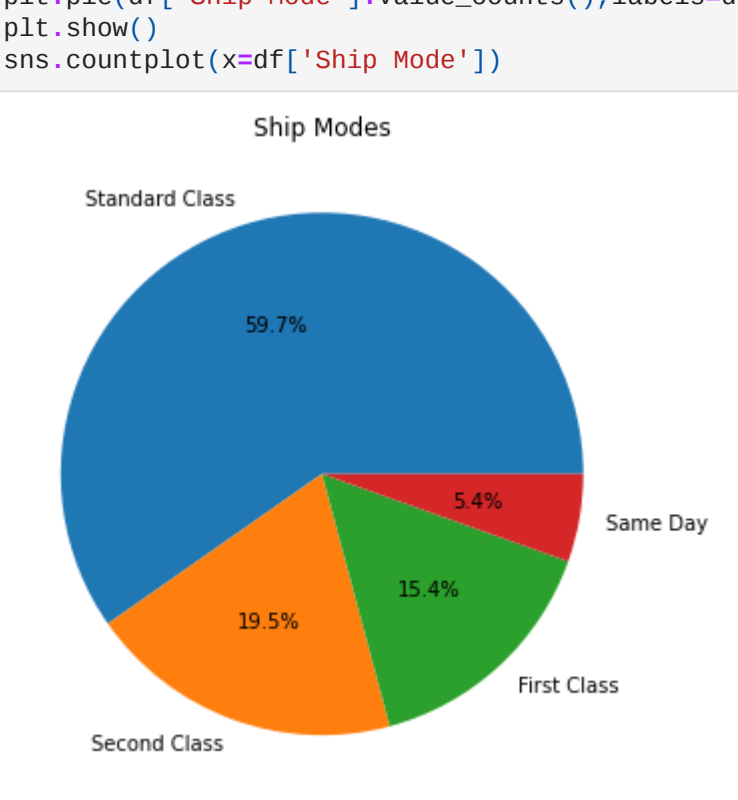


1 represents a strong positive correlation and -0.2 for negative correlation

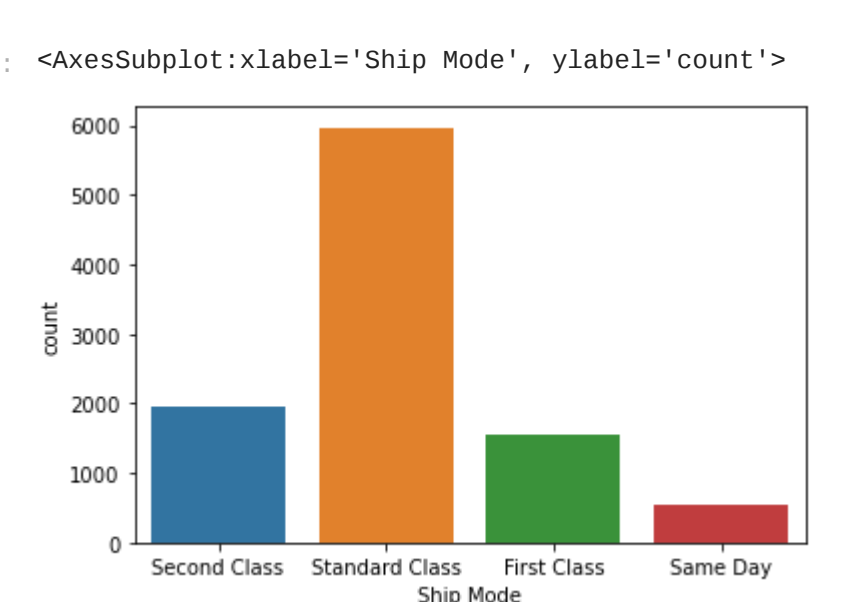
In [50]:
```
df1['Ship Mode'].value_counts()
```

Out[50]:
```
Standard Class    5968
Second Class      1945
First Class       1538
Same Day           543
Name: Ship Mode, dtype: int64
```

In [56]:
```
plt.figure(figsize=(6,6))
plt.title('Ship Modes')
plt.pie(df['Ship Mode'].value_counts(),labels=df['Ship Mode'].value_counts().index,autopct='%1.1f%%')
plt.show()
sns.countplot(x=df['Ship Mode'])
```



Out[56]:
```
<AxesSubplot:xlabel='Ship Mode', ylabel='count'>
```



In ship mode, most of them are preferring the standard class

In [57]:
```
df1['Segment'].value_counts()
```

Out[57]:
```
Consumer       5191
Corporate      3020
Home Office    1783
Name: Segment, dtype: int64
```

In [59]:
```
sns.pairplot(df,hue='Segment')
```

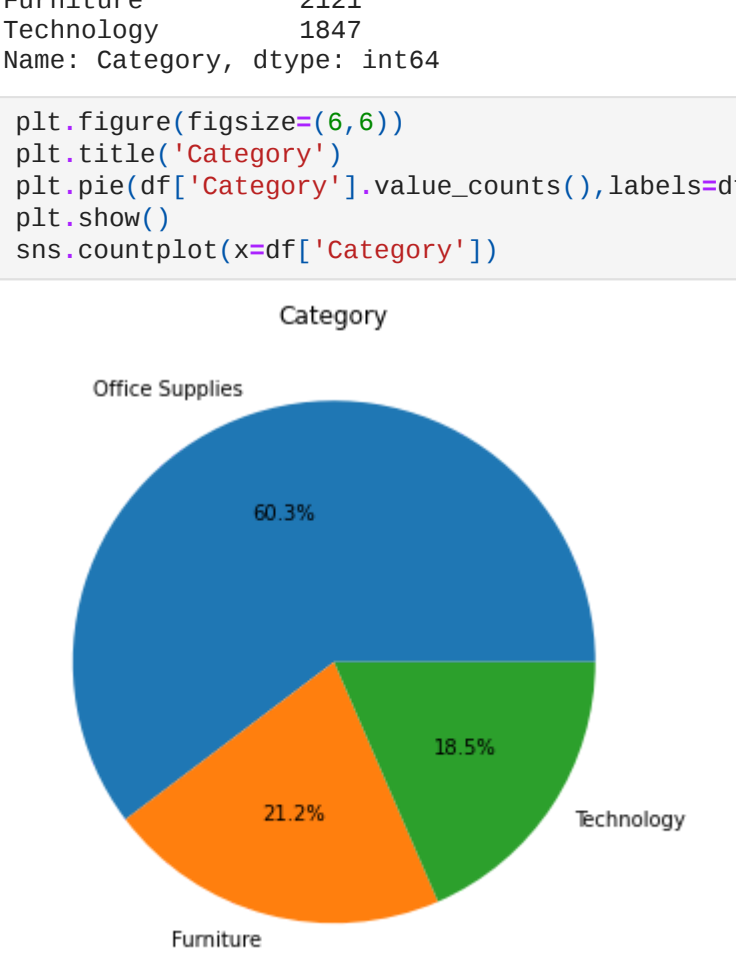
Out[59]:
```
<seaborn.axisgrid.PairGrid at 0xe062240c10>
```

In segment, profit and sales has positive correlation

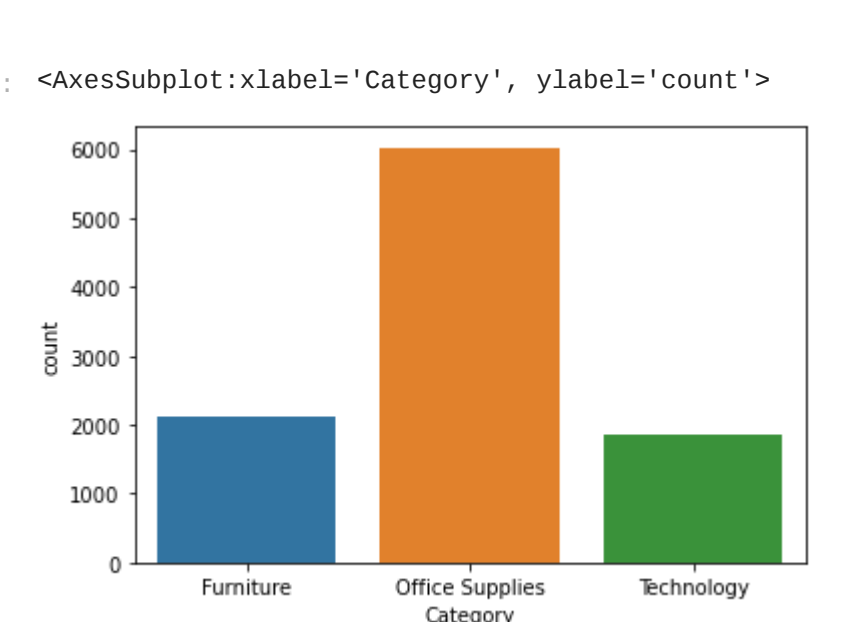
In [60]:
```
df['Category'].value_counts()
```

Out[60]:
```
Office Supplies    6026
Furniture          2121
Technology         1847
Name: Category, dtype: int64
```

In [61]:
```
plt.figure(figsize=(6,6))
plt.title('Category')
plt.pie(df['Category'].value_counts(),labels=df['Category'].value_counts().index,autopct='%1.1f%%')
plt.show()
sns.countplot(x=df['Category'])
```

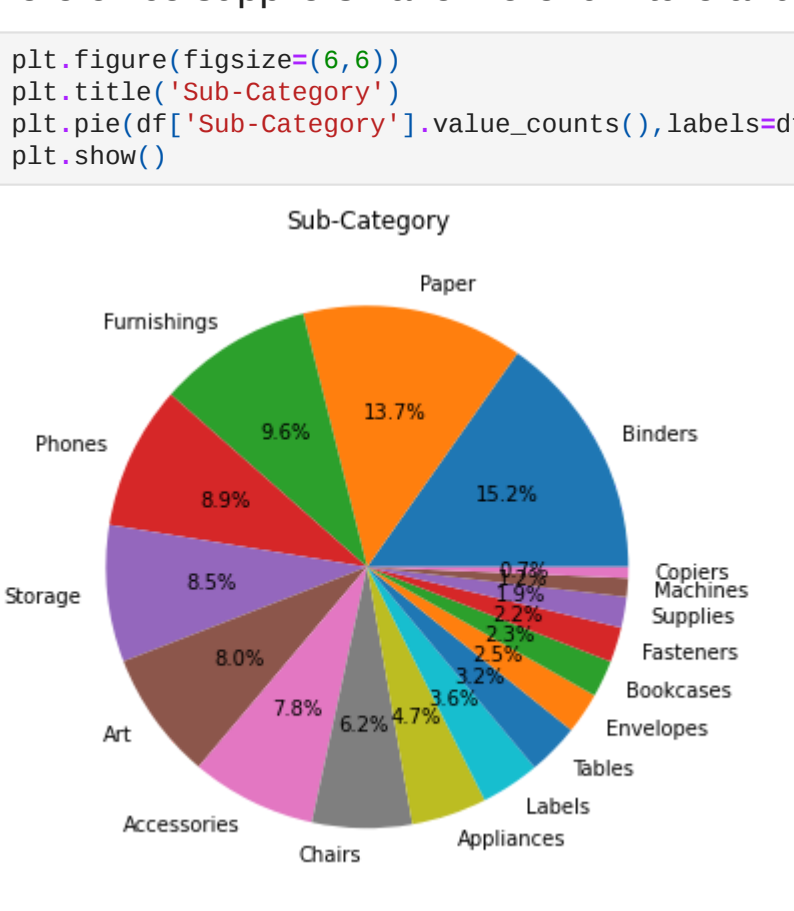


Out[61]:
```
<AxesSubplot:xlabel='Category', ylabel='count'>
```



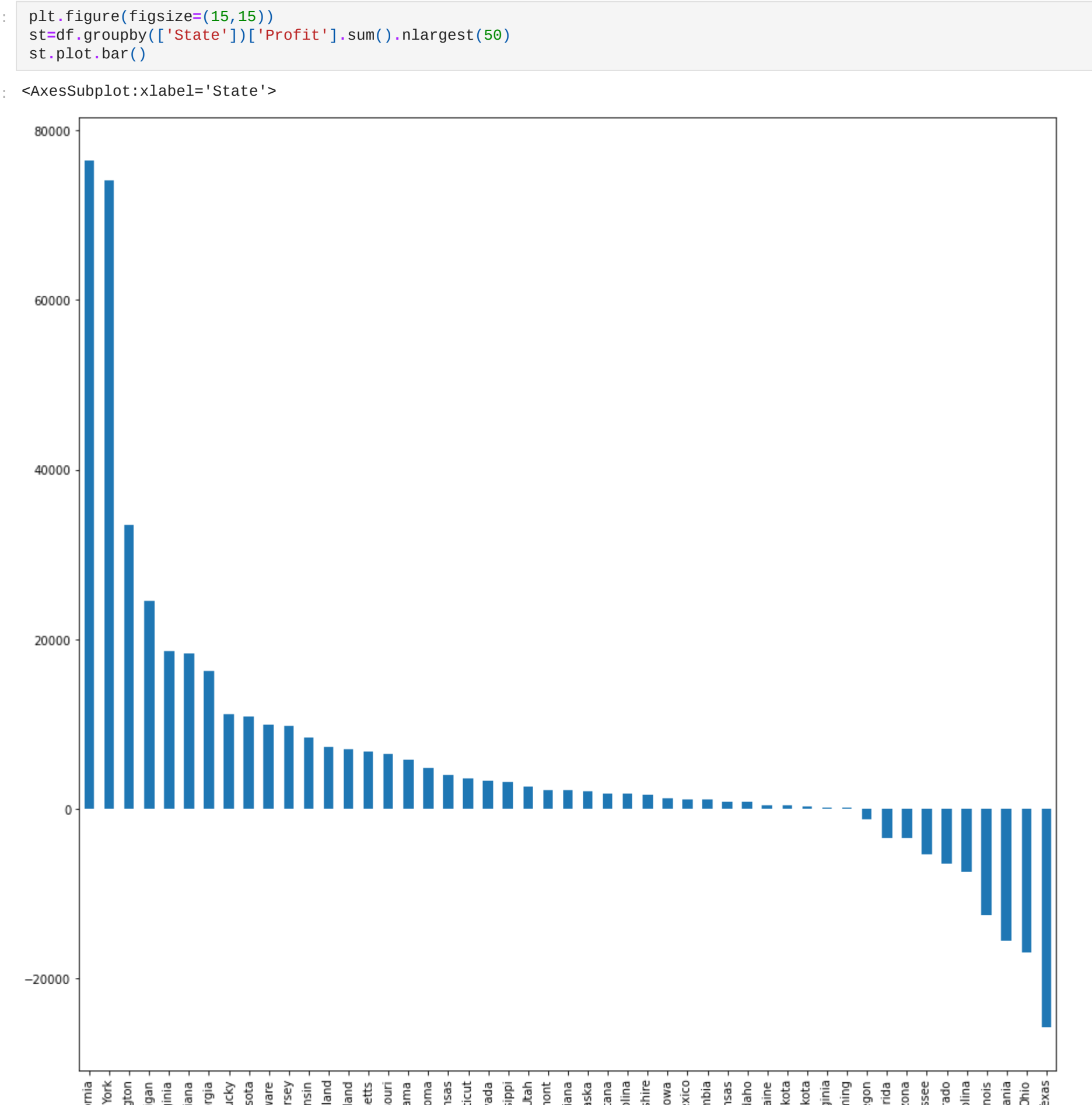
Here office suppliers have more furniture and technology

In [65]:
```
plt.figure(figsize=(6,6))
plt.title('Sub-Category')
plt.pie(df['Sub-Category'].value_counts(),labels=df['Sub-Category'].value_counts().index,autopct='%1.1f%%')
plt.show()
```



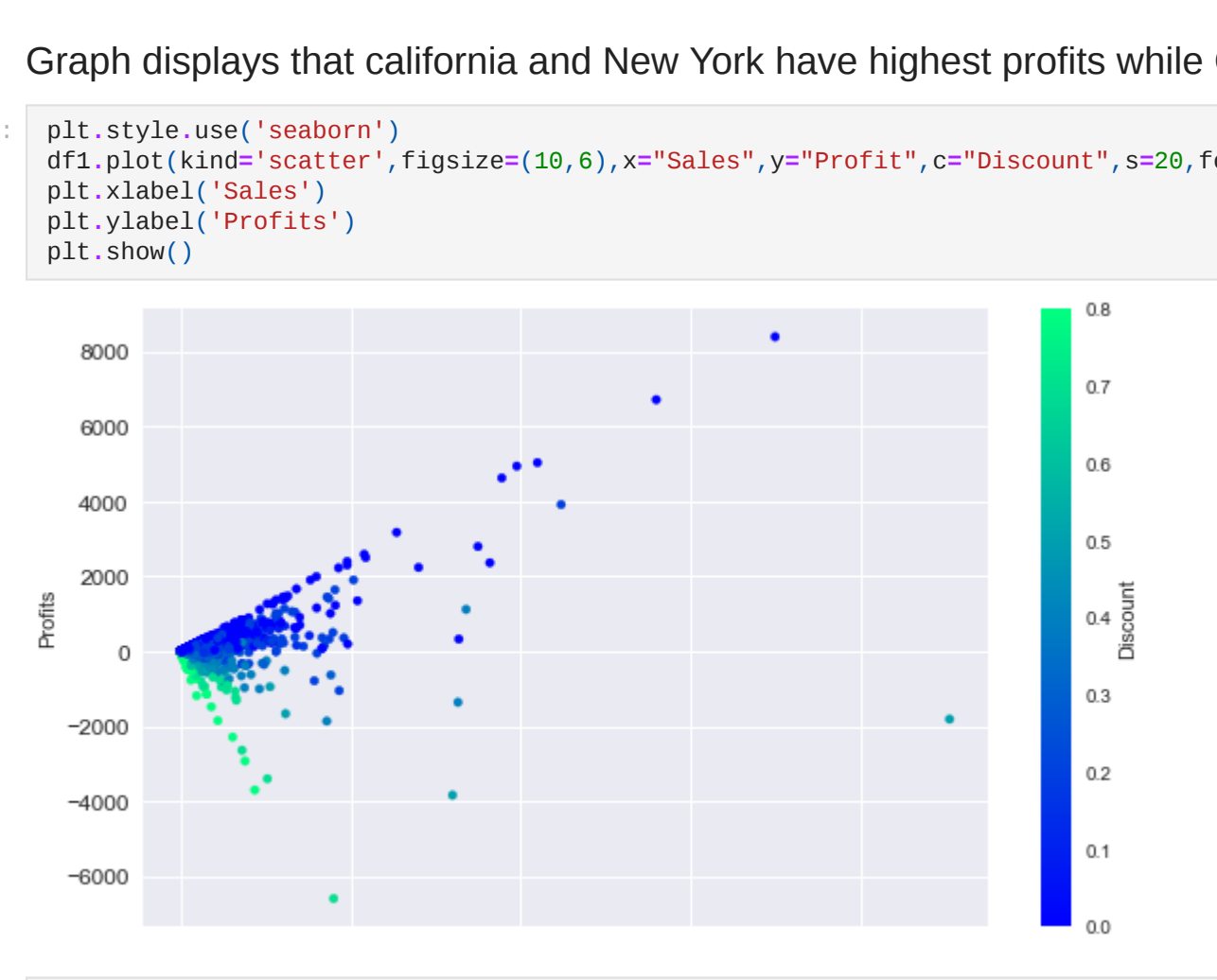
In [67]:
```
plt.figure(figsize=(15,15))
st=df.groupby(['State'])['Profit'].sum().nlargest(50)
st.plot.bar()
```

Out[67]:
```
<AxesSubplot:xlabel='State'>
```

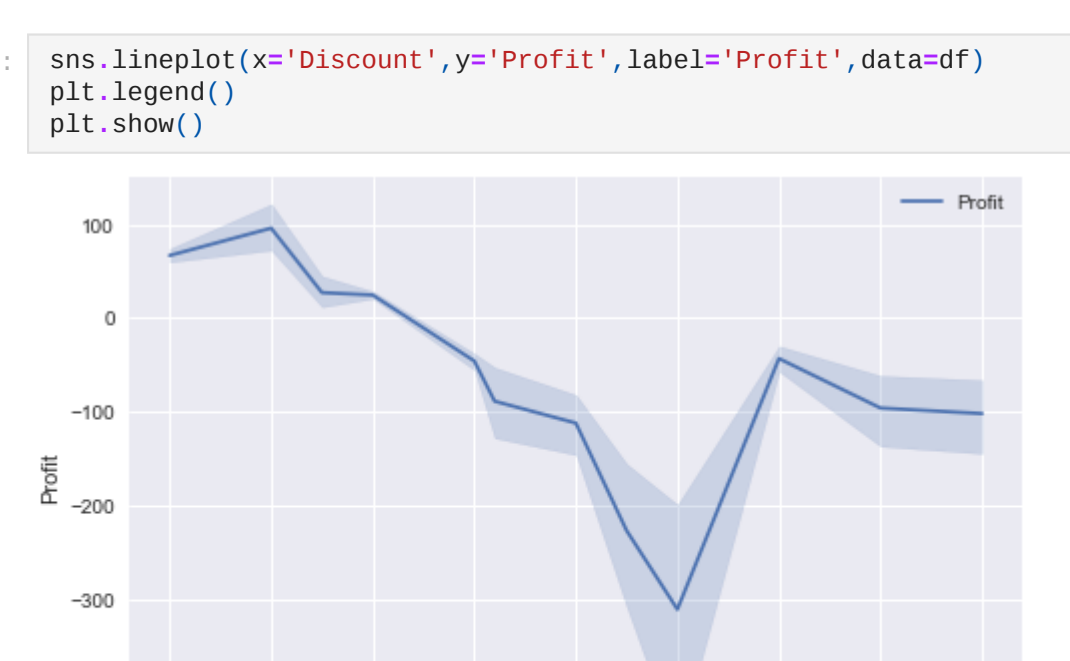


Graph displays that california and New York have highest profits while Ohio and Texas have least profits

In [70]:
```
plt.style.use('seaborn')
df1.plot(kind='scatter',figsize=(10,6),x="Sales",y="Profit",c="Discount",s=20,fontsize=12,colormap='winter')
plt.xlabel('Sales')
plt.ylabel('Profits')
plt.show()
```

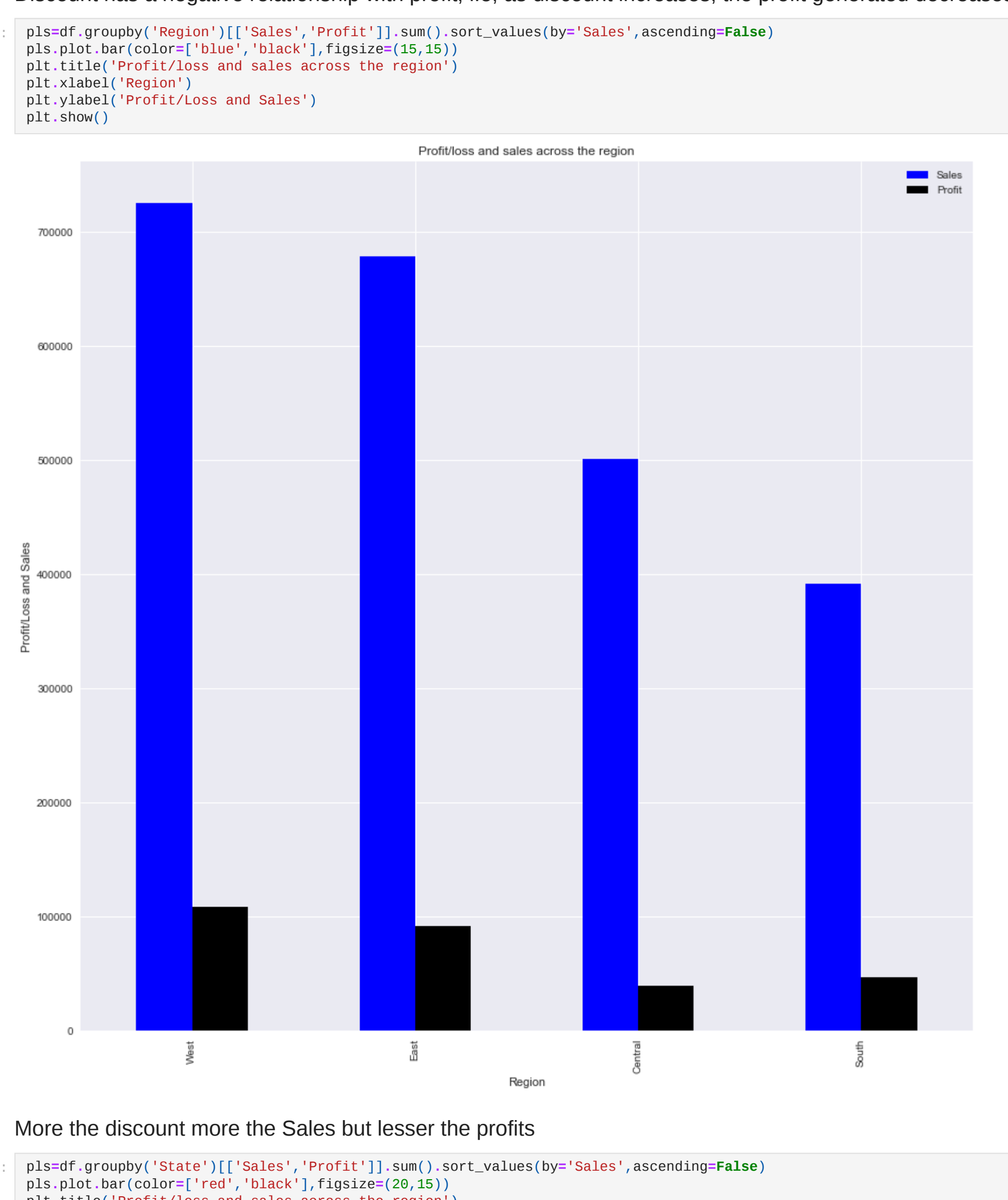


In [71]:
```
sns.lineplot(x='Discount',y='Profit',label='Profit',data=df)
plt.legend()
plt.show()
```



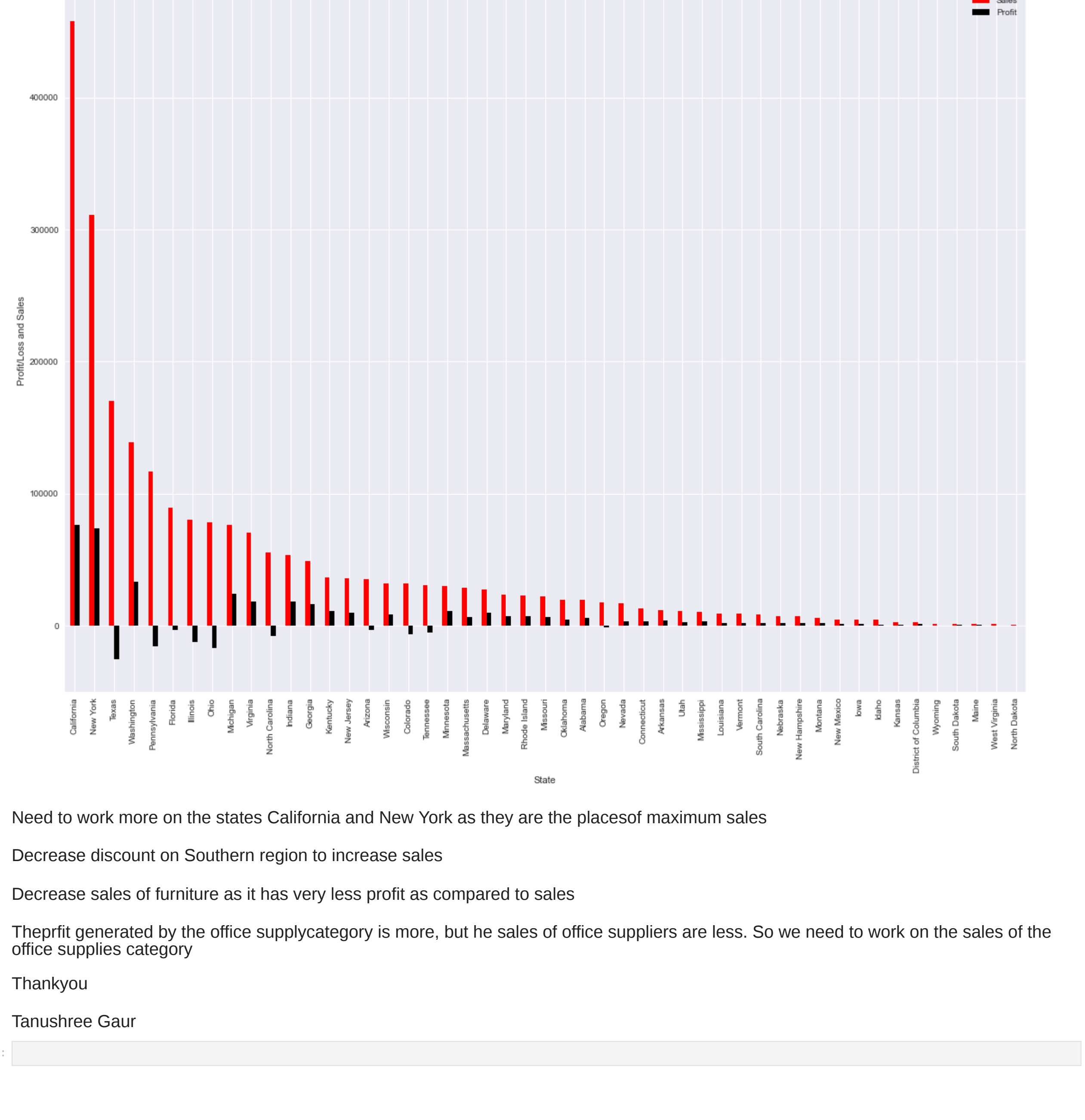
Discount has a negative relationship with profit, i.e, as discount increases, the profit generated decreases

In [72]:
```
pls=df.groupby('Region')[['Sales','Profit']].sum().sort_values(by='Sales',ascending=False)
pls.plot.bar(color=['blue','black'],figsize=(15,15))
plt.title('Profit/loss and sales across the region')
plt.xlabel('Region')
plt.ylabel('Profit/Loss and Sales')
plt.show()
```



More the discount more the Sales but lesser the profits

In [74]:
```
pls=df.groupby('State')[['Sales','Profit']].sum().sort_values(by='Sales',ascending=False)
pls.plot.bar(color=['red','black'],figsize=(20,15))
plt.title('Profit/loss and sales across the region')
plt.xlabel('State')
plt.ylabel('Profit/Loss and Sales')
plt.show()
```



Need to work more on the states California and New York as they are the placesof maximum sales

Decrease discount on Southern region to increase sales

Decrease sales of furniture as it has very less profit as compared to sales

Theprfit generated by the office supplycategory is more, but he sales of office suppliers are less. So we need to work on the sales of the office supplies category

Thankyou

Tanushree Gaur

In [ ]: