AUTHOR : K.H.L.B.GAYATHRI

DATA SCIENCE & BUSINESS ANALYTICS TASKS.

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 more profit.

DATASET : SAMPLESUPERSTORE.CSV (https://bit.ly/3i4rbWl (https://bit.ly/3i4rbWl))

Importing Libraries

In [1]:

```
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [3]:

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

In [13]:

```
df=pd.read_csv("SampleSuperstore.csv")
df.head()
```

Out[13]:

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

In [12]:

```
1 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 [14]:

```
1 df.isnull().sum()
```

Out[14]:

```
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 [15]:

```
1 df.columns
```

Out[15]:

```
Index(['Ship Mode', 'Segment', 'Country', 'City', 'State', 'Postal Code',
       'Region', 'Category', 'Sub-Category', 'Sales', 'Quantity', 'Discoun
t',
       'Profit'],
      dtype='object')
```

In [16]:

```
1 df.shape
```

Out[16]:

```
(9994, 13)
```

In [17]:

```
1 df.nunique()
```

Out[17]:

```
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
```

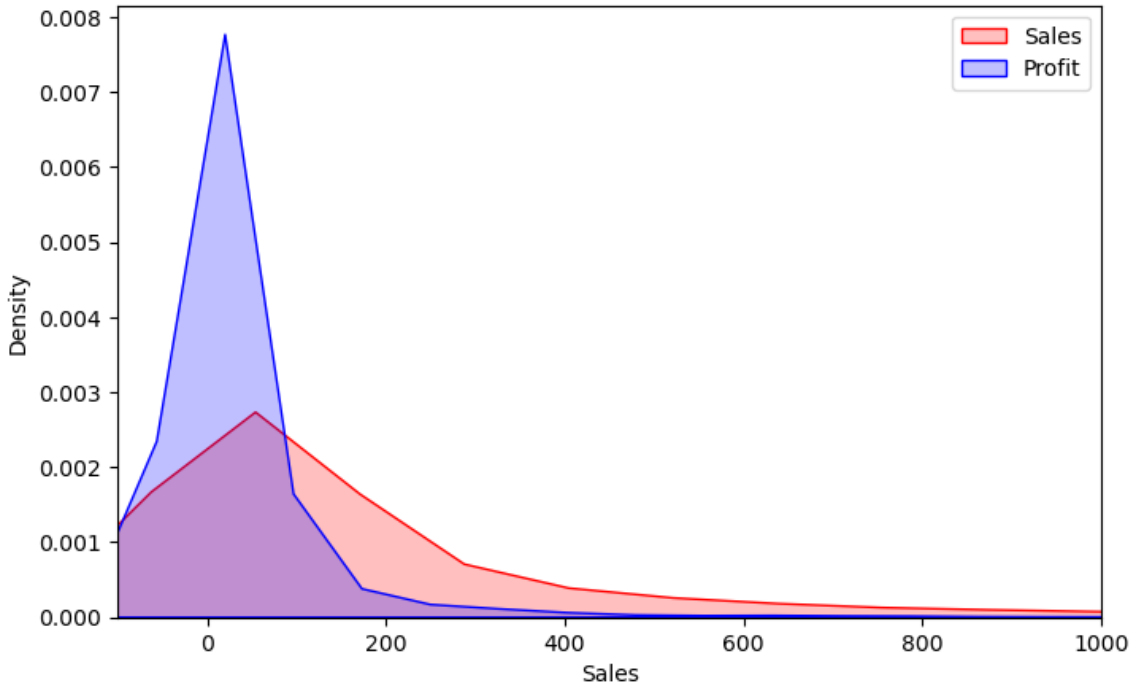
Exploratory Data Analysis

In [35]:

```
plt.figure(figsize=(8,5))
sns.kdeplot(df['Sales'],color='red',label='Sales',shade=True)
sns.kdeplot(df['Profit'],color='Blue',label='Profit',shade=True)
plt.xlim([-100,1000])
plt.legend()
```

Out[35]:

```
<matplotlib.legend.Legend at 0x164320a2fa0>
```



Analysis using Pairplot of each column

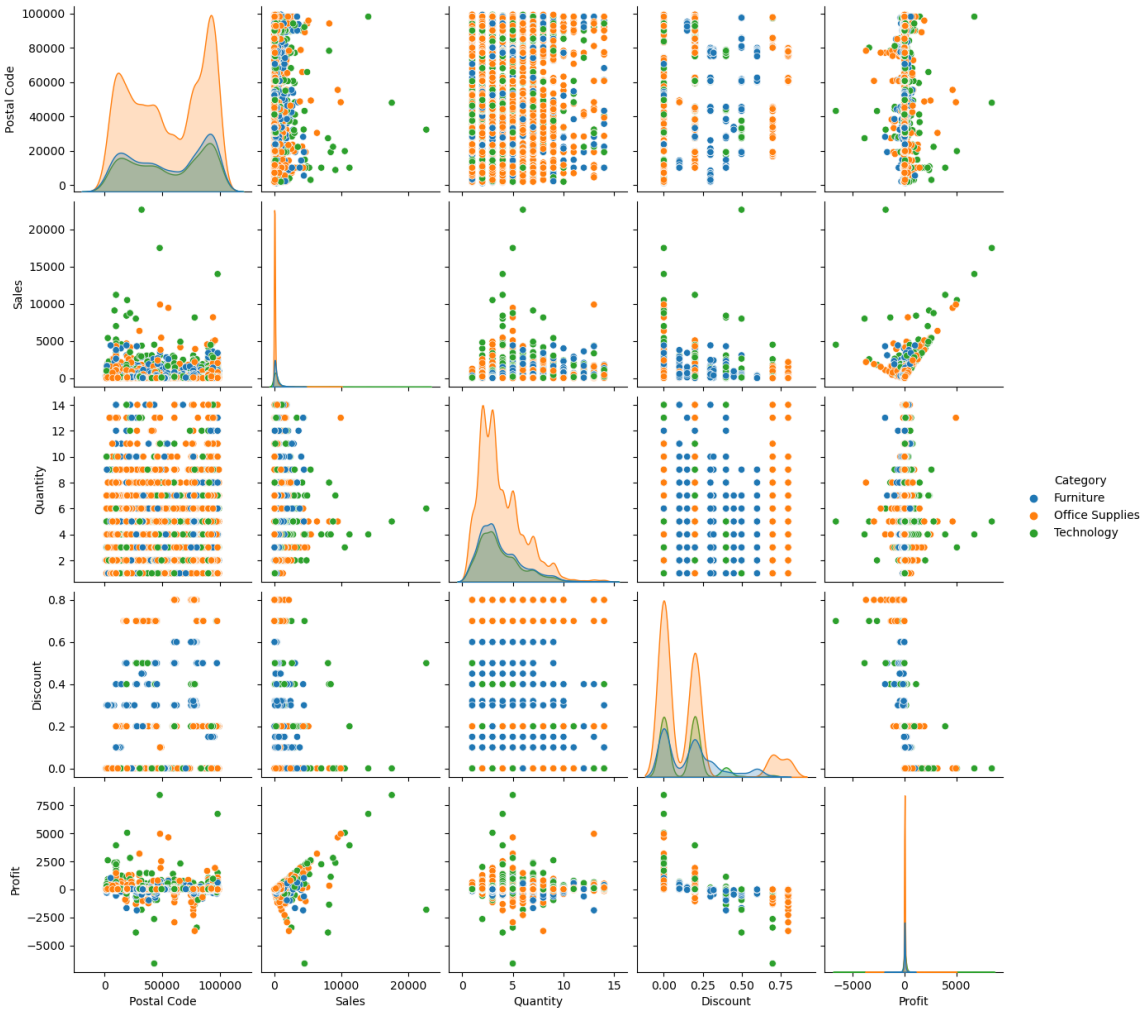
[1] Based on the Catagory

In [40]:

```
1 sns.pairplot(df,hue='Category')
```

Out[40]:

```
<seaborn.axisgrid.PairGrid at 0x1643e08eeb0>
```



[2] Based on Region

In [42]:

```
1 sns.pairplot(df,hue='Region')
```

Out[42]:

```
<seaborn.axisgrid.PairGrid at 0x16438a12e20>
```

[3] Based on the segment

In [43]:

```
1 sns.pairplot(df,hue='Segment')
```

Out[43]:

```
<seaborn.axisgrid.PairGrid at 0x1644779cb20>
```

In [44]:

```
1 df.corr()
```

Out[44]:

	Postal Code	Sales	Quantity	Discount	Profit
Postal Code	1.000000	-0.023854	0.012761	0.058443	-0.029961
Sales	-0.023854	1.000000	0.200795	-0.028190	0.479064
Quantity	0.012761	0.200795	1.000000	0.008623	0.066253
Discount	0.058443	-0.028190	0.008623	1.000000	-0.219487
Profit	-0.029961	0.479064	0.066253	-0.219487	1.000000

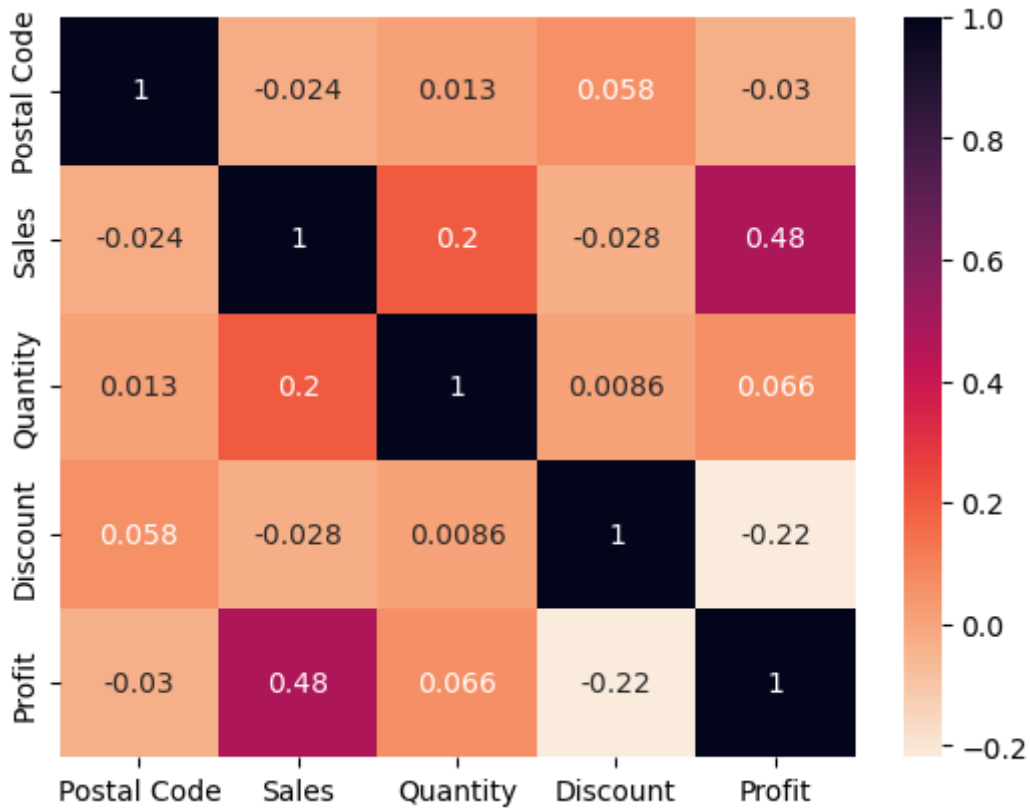
Heatmap for Correlation

In [45]:

```
1 sns.heatmap(df.corr(),cmap='rocket_r',annot=True)
```

Out[45]:

<AxesSubplot:>



From above Heatmap:

i)Sales and Profit are Moderately Correlated.
ii)Discount and Profit are Negatively Correlated.
iii)Quantity and Profit are less Moderately Correlated.

Count plot of each column

In [4]:

```
1 df['Ship Mode'].value_counts()
```

Out[4]:

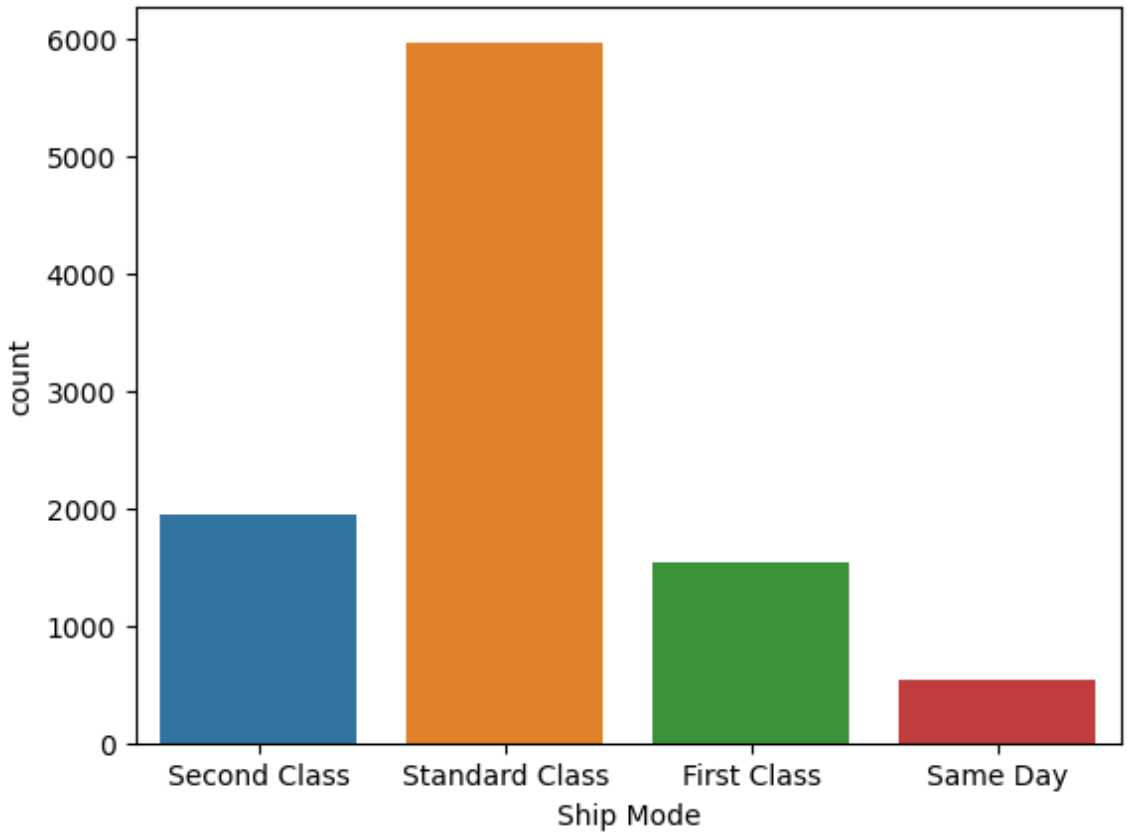
```
Standard Class    5968
Second Class      1945
First Class       1538
Same Day           543
Name: Ship Mode, dtype: int64
```

In [5]:

```
1 sns.countplot(x=df['Ship Mode'])
```

Out[5]:

```
<AxesSubplot:xlabel='Ship Mode', ylabel='count'>
```



In [6]:

```
1 df['Segment'].value_counts()
```

Out[6]:

```
Consumer       5191
Corporate      3020
Home Office    1783
Name: Segment, dtype: int64
```

In [ ]:

```
1
```

In [ ]:

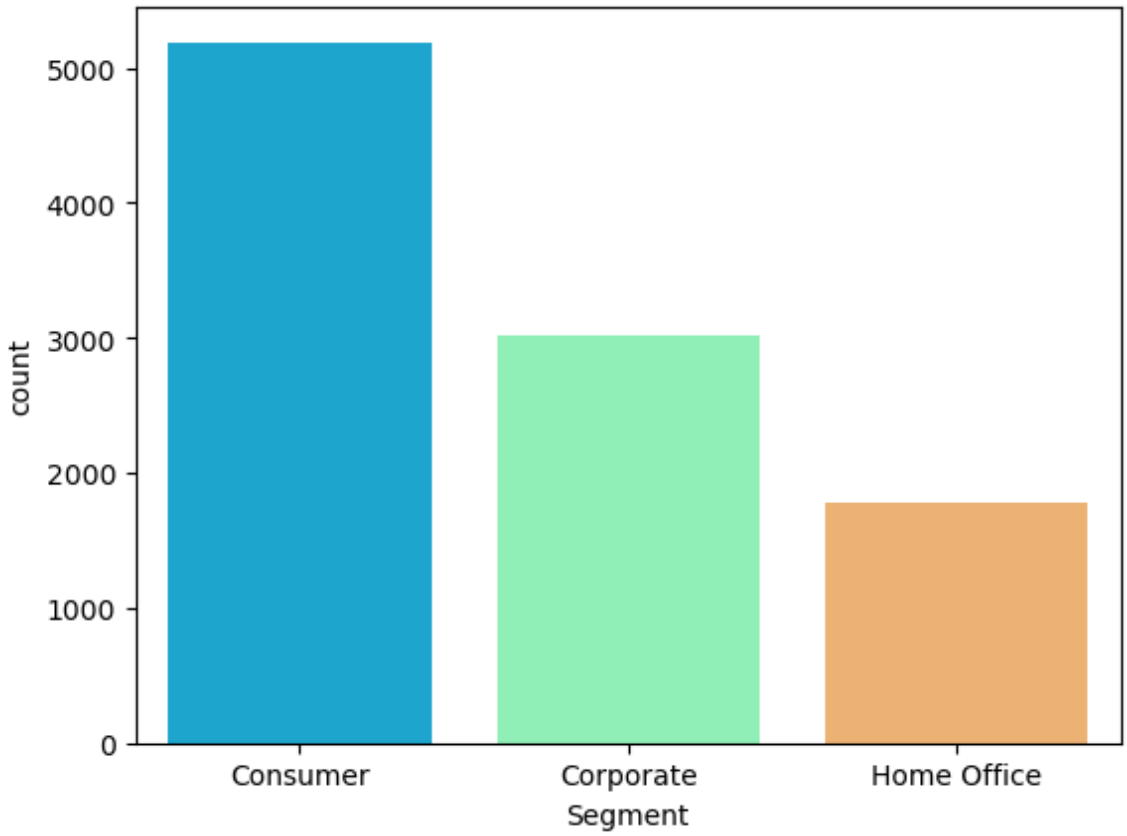
```
1
```

In [9]:

```
1 sns.countplot(x = 'Segment',data = df, palette = 'rainbow')
```

Out[9]:

```
<AxesSubplot:xlabel='Segment', ylabel='count'>
```



In [10]:

```
1 df['Category'].value_counts()
```

Out[10]:

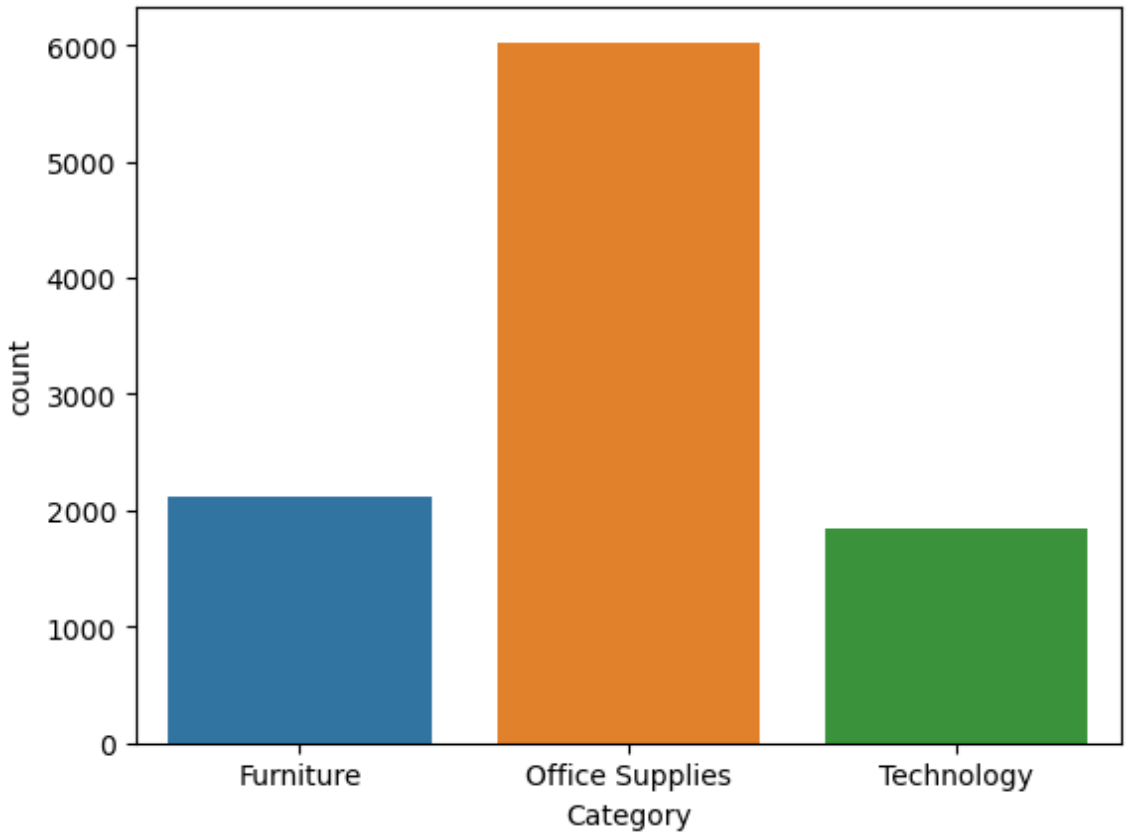
```
Office Supplies    6026
Furniture          2121
Technology         1847
Name: Category, dtype: int64
```

In [11]:

```
sns.countplot(x='Category',data=df,palette='tab10')
```

Out[11]:

```
<AxesSubplot:xlabel='Category', ylabel='count'>
```



In [67]:

```
fig, axs = plt.subplots(ncols=2, nrows = 2, figsize = (10,10))
sns.distplot(df['Sales'], color = 'red',  ax = axs[0][0])
sns.distplot(df['Profit'], color = 'green',  ax = axs[0][1])
sns.distplot(df['Quantity'], color = 'orange',  ax = axs[1][0])
sns.distplot(df['Discount'], color = 'blue',  ax = axs[1][1])
axs[0][0].set_title('Sales Distribution', fontsize = 20)
axs[0][1].set_title('Profit Distribution', fontsize = 20)
axs[1][0].set_title('Quantity distribution', fontsize = 20)
axs[1][1].set_title('Discount Distribution', fontsize = 20)
plt.show()
```

. . .

Statewise Deal Analysis

In [68]:

```
df['Country'].value_counts()
```

Out[68]:

```
United States    9994
Name: Country, dtype: int64
```

In [69]:

```
df1 = df['State'].value_counts()
df1.head(10)
```

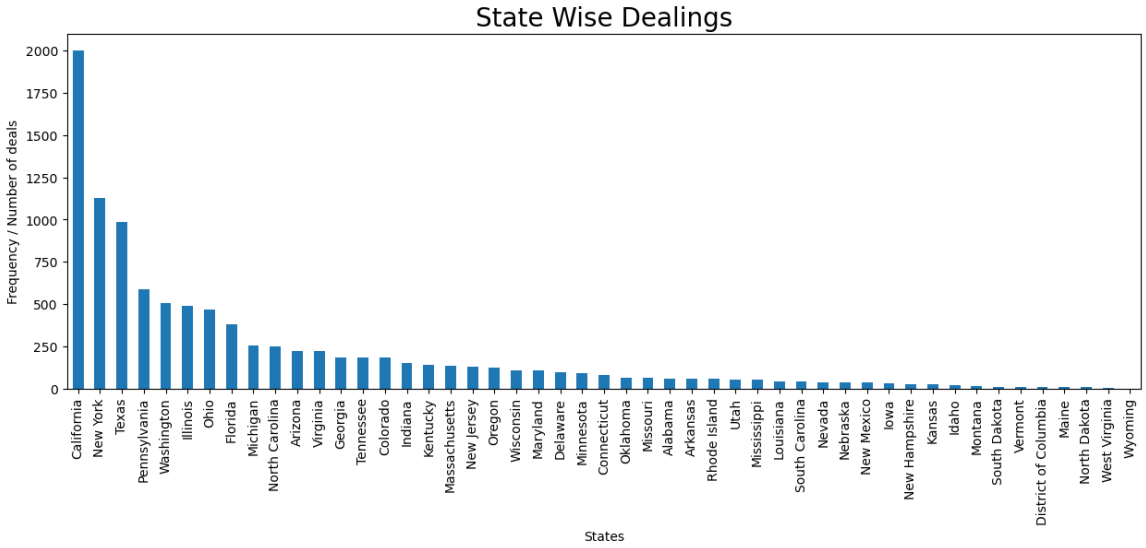
Out[69]:

```
California        2001
New York          1128
Texas              985
Pennsylvania       587
Washington         506
Illinois           492
Ohio               469
Florida            383
Michigan           255
North Carolina     249
Name: State, dtype: int64
```

In [70]:

```
df1.plot(kind='bar',figsize=(15,5))
plt.ylabel('Frequency / Number of deals')
plt.xlabel('States')

plt.title('State Wise Dealings', fontsize = 20)
plt.show()
```



Here is top 3 state where deals are Highest. i)Califonia
ii)New York
iii)Texas

Wyoming: Lowest Number of deal

In [71]:

```
df['State'].value_counts().mean()
```

Out[71]:

```
203.9591836734694
```

Average number of deal per state is 204.

City Wise analysis of the dealing

In [76]:

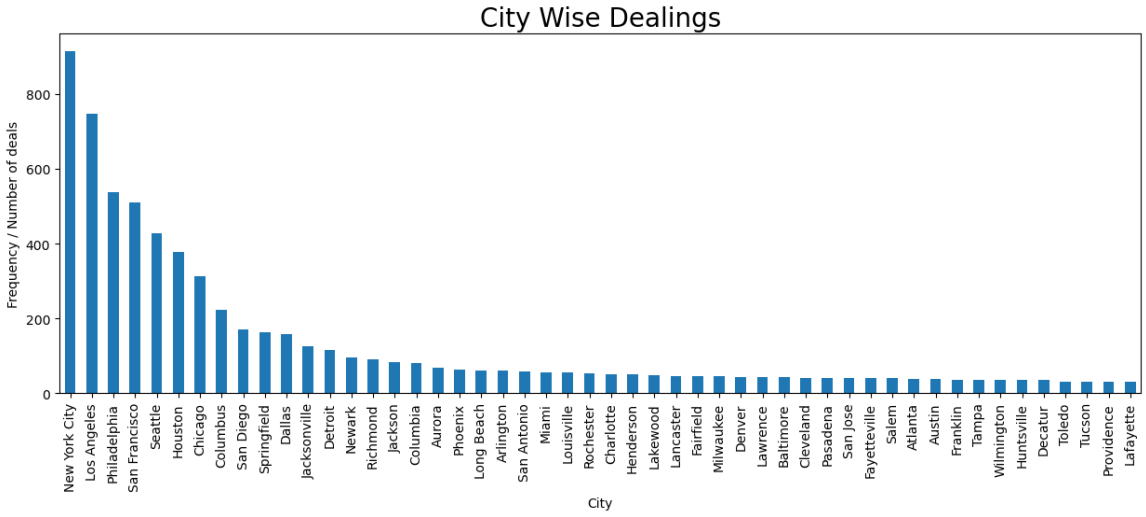
```
df2 = df['City'].value_counts()
df2=df2.head(50)
```

In [77]:

```
df2.plot(kind='bar',figsize=(15,5))
plt.ylabel('Frequency / Number of deals')
plt.xlabel('City')

plt.title('City Wise Dealings', fontsize = 20)
plt.show()

```



Here is top 3 city where deals are Highest.

1. New York City
2. Los Angeles
3. Philadelphia

In [78]:

```
df['City'].value_counts().mean()
```

Out[78]:

```
18.821092278719398
```

Average number of deal per city is 19.

Segment wise analysis of Profit, Discount and sell

In [79]:

```
df['Segment'].value_counts()

```

Out[79]:

```
Consumer       5191
Corporate      3020
Home Office    1783
Name: Segment, dtype: int64
```

In [80]:

```
df_segment= df.groupby(['Segment'])[['Sales', 'Discount', 'Profit']].mean()
df_segment
```

Out[80]:

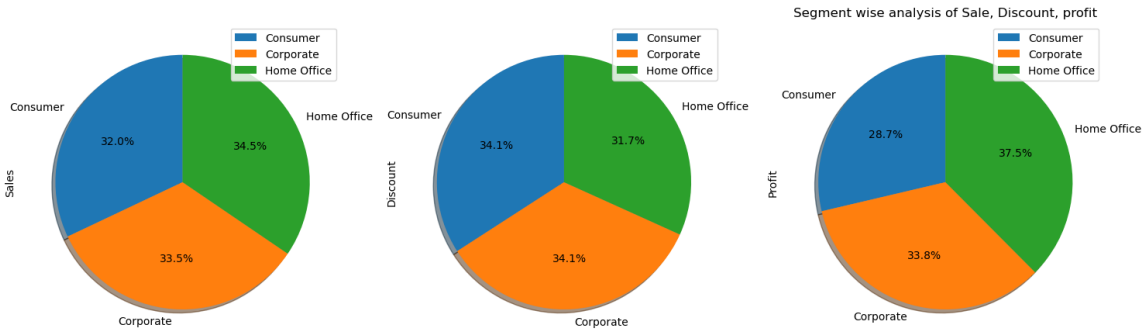
Segment	Sales	Discount	Profit
Consumer	223.733644	0.158141	25.836873
Corporate	233.823300	0.158228	30.456667
Home Office	240.972041	0.147128	33.818664

In [81]:

```
#1. sales 2. Discount 3. Profit
df_segment.plot.pie(subplots=True,
                    autopct='%1.1f%%',
                    figsize=(18, 20),
                    startangle=90,     # start angle 90° (Africa)
                    shadow=True,
                    labels = df_segment.index)
plt.title('Segment wise analysis of Sale, Discount, profit')
```

Out[81]:

```
Text(0.5, 1.0, 'Segment wise analysis of Sale, Discount, profit')
```



Statewise analysis of Profit Discount and sell

In [82]:

```
df['State'].value_counts().head(10)
```

Out[82]:

```
California        2001
New York          1128
Texas              985
Pennsylvania       587
Washington         506
Illinois           492
Ohio               469
Florida            383
Michigan           255
North Carolina     249
Name: State, dtype: int64
```

In [83]:

```
df_state= df.groupby(['State'])[['Sales', 'Discount', 'Profit']].mean()
df_state.head(10)
```

Out[83]:

State	Sales	Discount	Profit
Alabama	319.846557	0.000000	94.865989
Arizona	157.508933	0.303571	-15.303235
Arkansas	194.635500	0.000000	66.811452
California	228.729451	0.072764	38.171608
Colorado	176.418231	0.316484	-35.867351
Connecticut	163.223866	0.007317	42.823071
Delaware	285.948635	0.006250	103.930988
District of Columbia	286.502000	0.000000	105.958930
Florida	233.612815	0.299347	-8.875461
Georgia	266.825217	0.000000	88.315453

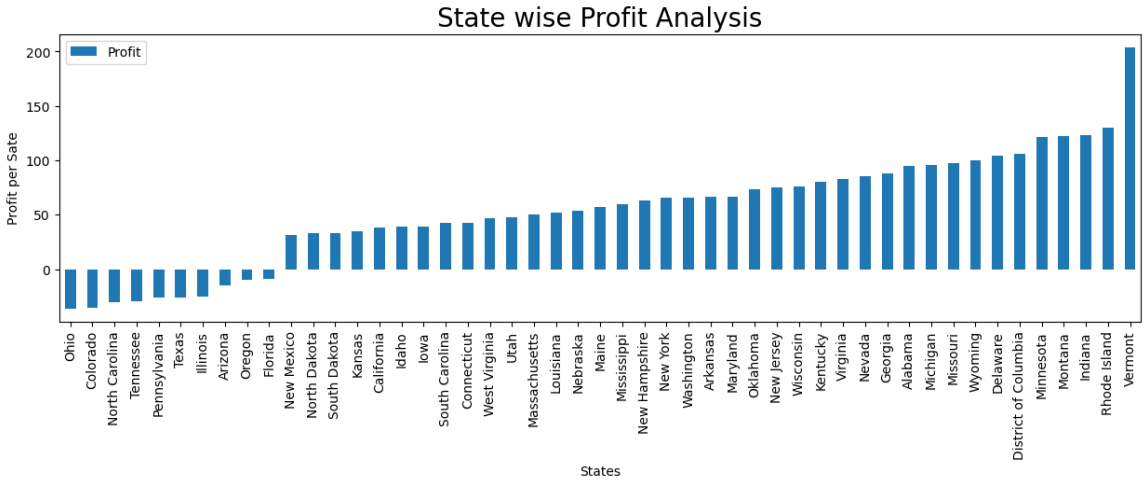
[1] Statewise Profit Analysis

In [86]:

```
df_state1=df_state.sort_values('Profit')

df_state1[['Profit']].plot(kind = 'bar', figsize = (15,4))
plt.title('State wise Profit Analysis', fontsize = 20)
plt.ylabel('Profit per Sate')
plt.xlabel('States')
plt.show()

```



RESULT

Vermont: Highest Profit

Ohio: Lowest Profit

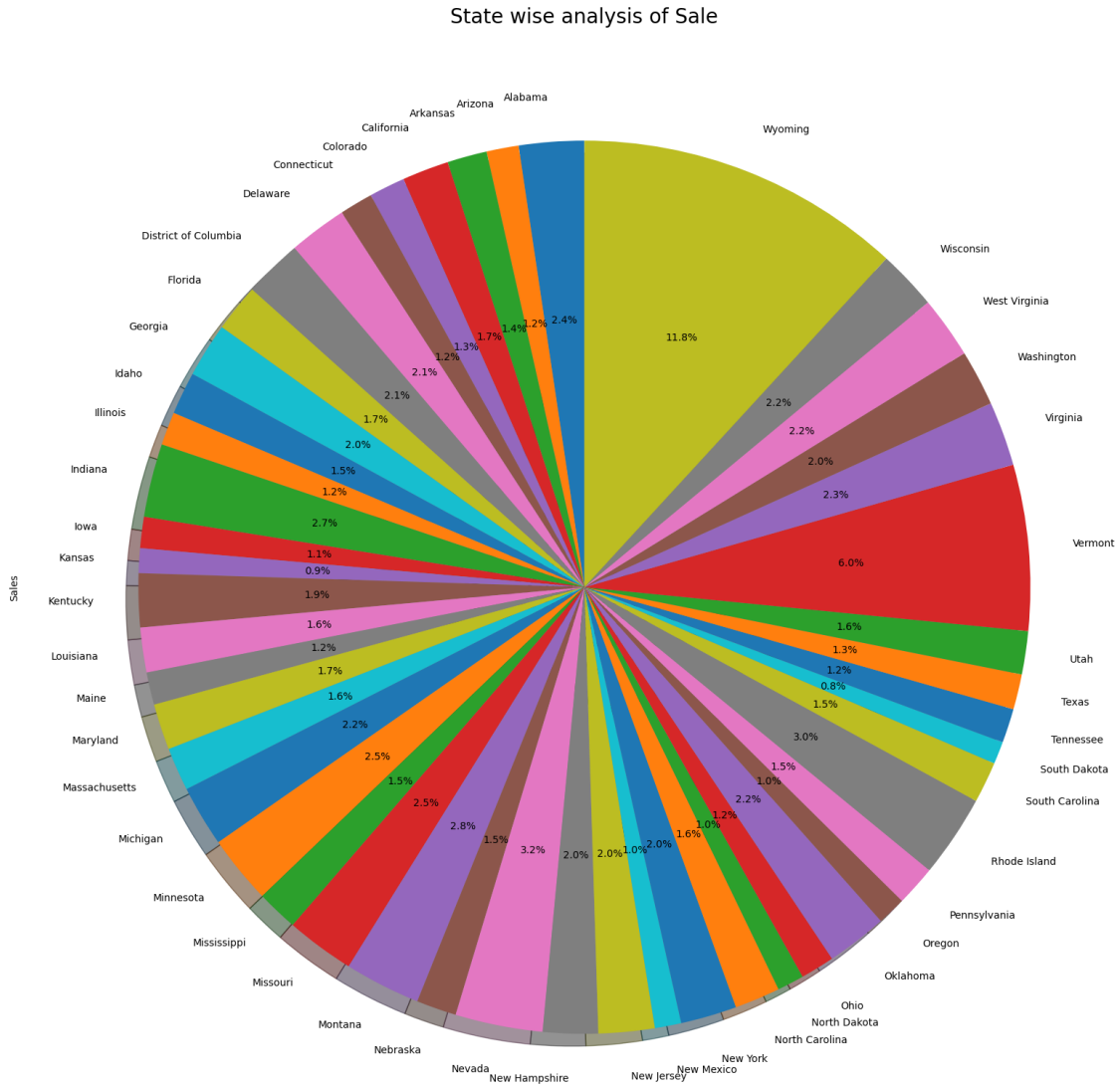
[2] Statewise Sale Analysis

In [89]:

```
df_state['Sales'].plot(kind='pie',
                       figsize = (20,20),
                       autopct='%1.1f%%',
                       startangle=90,     # start angle 90° (Africa)
                       shadow=True)
plt.title('State wise analysis of Sale',fontsize=20)
```

Out[89]:

```
Text(0.5, 1.0, 'State wise analysis of Sale')
```



Highest amount of sales= Wyoming(11.8%)
Lowest amount of sales= South Dakota(0.8%)

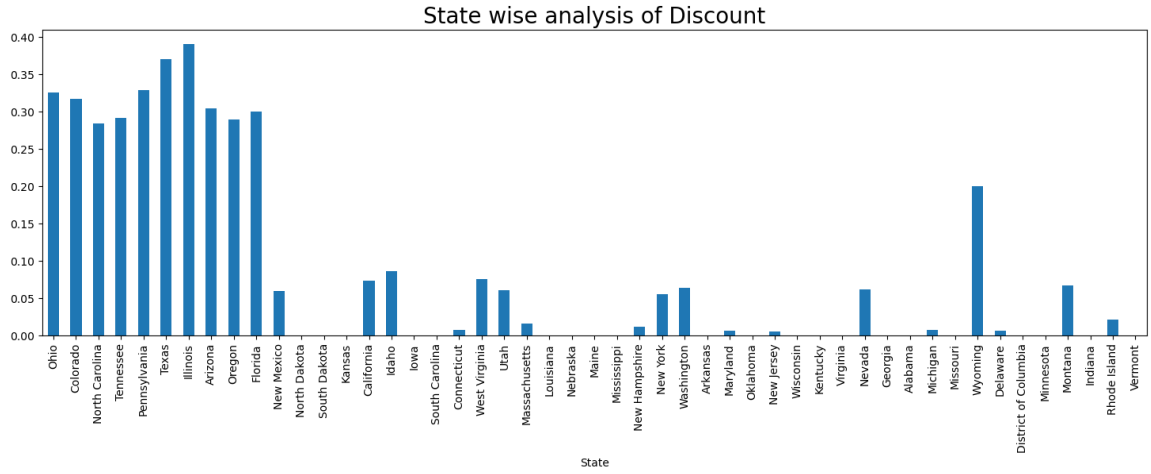
[3] Statewise Discount Analysis

In [95]:

```
df_state1['Discount'].plot(kind='bar',figsize=(18,5))
plt.title('State wise analysis of Discount', fontsize=20)
```

Out[95]:

```
Text(0.5, 1.0, 'State wise analysis of Discount')
```



Citywise Analysis of the Profit

In [96]:

```
df_city= df.groupby(['City'])[['Sales', 'Discount', 'Profit']].mean()
df_city = df_city.sort_values('Profit')
df_city.head()
```

Out[96]:

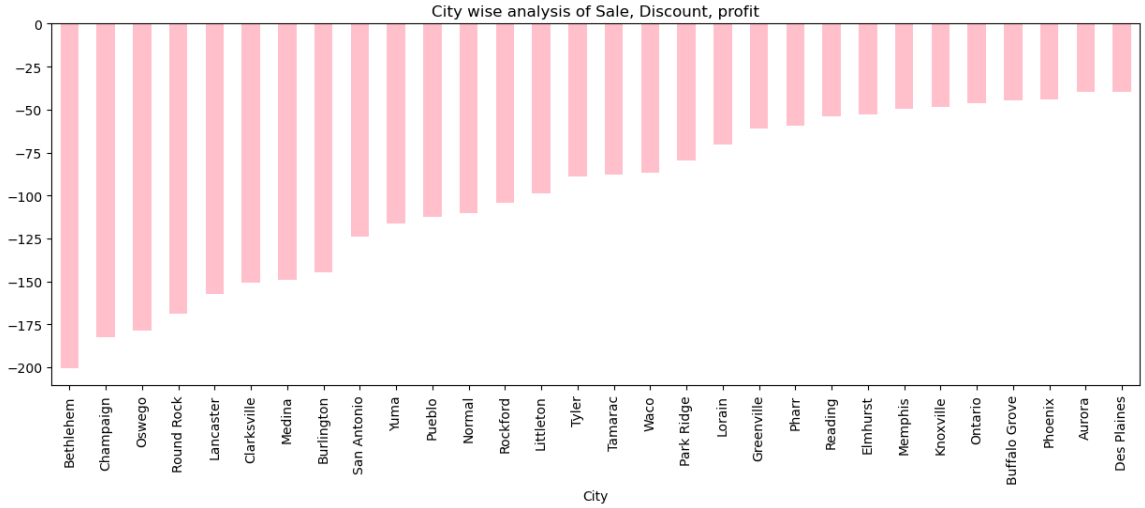
City	Sales	Discount	Profit
Bethlehem	337.926800	0.380000	-200.619160
Champaign	151.960000	0.600000	-182.352000
Oswego	107.326000	0.600000	-178.709200
Round Rock	693.436114	0.274286	-169.061614
Lancaster	215.031826	0.315217	-157.371052

In [97]:

```
#1.Low Profit
df_city['Profit'].head(30).plot(kind='bar',figsize=(15,5),color = 'Pink')
plt.title('City wise analysis of Sale, Discount, profit')
```

Out[97]:

```
Text(0.5, 1.0, 'City wise analysis of Sale, Discount, profit')
```

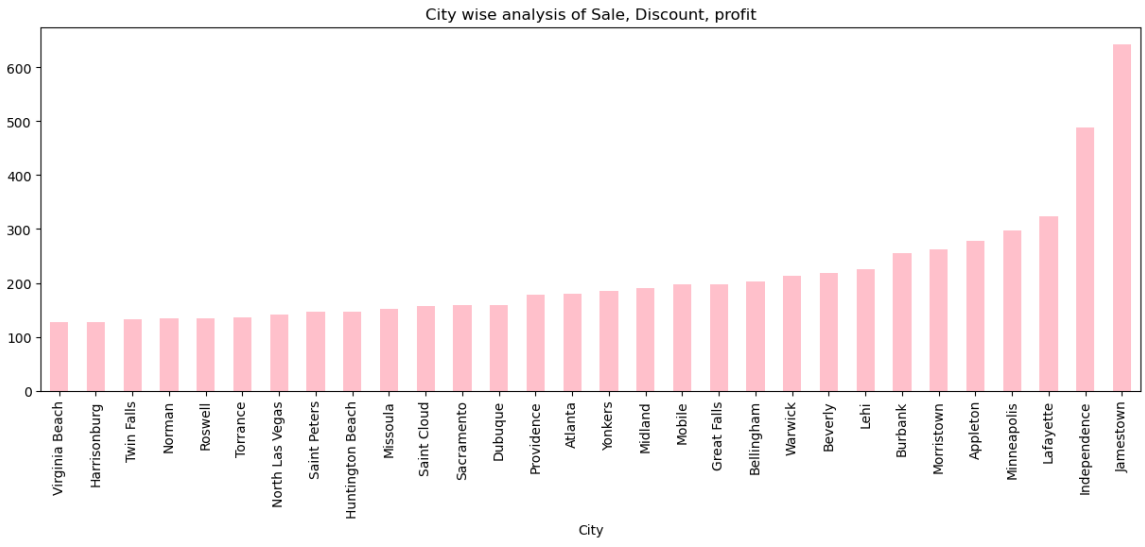


In [98]:

```
#2. High Profit
df_city['Profit'].tail(30).plot(kind='bar',figsize=(15,5),color = 'Pink')
plt.title('City wise analysis of Sale, Discount, profit')
```

Out[98]:

```
Text(0.5, 1.0, 'City wise analysis of Sale, Discount, profit')
```



30 CITIES WHICH HAS PROFIT IN POSITIVE
30 CITIES WHICH HAS PROFIT IN NEGATIVE THE BALANCE IS PRETTY GOOD HERE!

QUANTITY WISE SALES, PROFIT AND DISCOUNT ANALYSIS

In [99]:

```
df_quantity = df.groupby(['Quantity'])[['Sales', 'Discount', 'Profit']].mean()
df_quantity.head(10)
```

Out[99]:

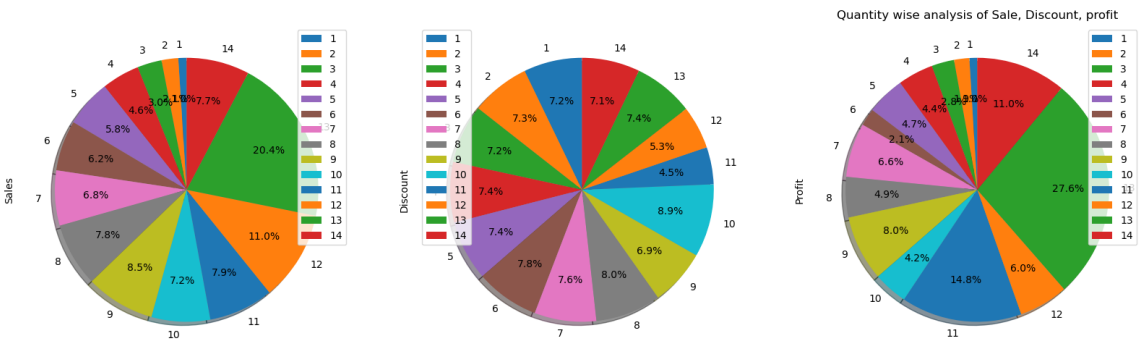
Quantity	Sales	Discount	Profit
1	59.234632	0.152959	8.276396
2	120.354488	0.154858	16.006831
3	175.201578	0.153329	23.667715
4	271.764059	0.157708	37.131310
5	337.936339	0.157146	40.257394
6	362.101960	0.166556	18.051517
7	395.888393	0.161980	56.579163
8	458.210802	0.171595	42.244342
9	498.083683	0.147946	68.557716
10	422.046737	0.190702	35.862404

In [100]:

```
#1. sales 2. Discount 3. Profit
df_quantity.plot.pie(subplots=True,
                     autopct='%1.1f%%',
                     figsize=(20, 20),
                      pctdistance=0.69,
                     startangle=90,     # start angle 90° (Africa)
                     shadow=True,
                     labels = df_quantity.index)
plt.title('Quantity wise analysis of Sale, Discount, profit')
```

Out[100]:

```
Text(0.5, 1.0, 'Quantity wise analysis of Sale, Discount, profit')
```



CATAGORY WISE SALES DISCOUNT AND PROFIT

In [101]:

```
df_category = df.groupby(['Category'])[['Sales', 'Discount', 'Profit']].mean()
df_category
```

Out[101]:

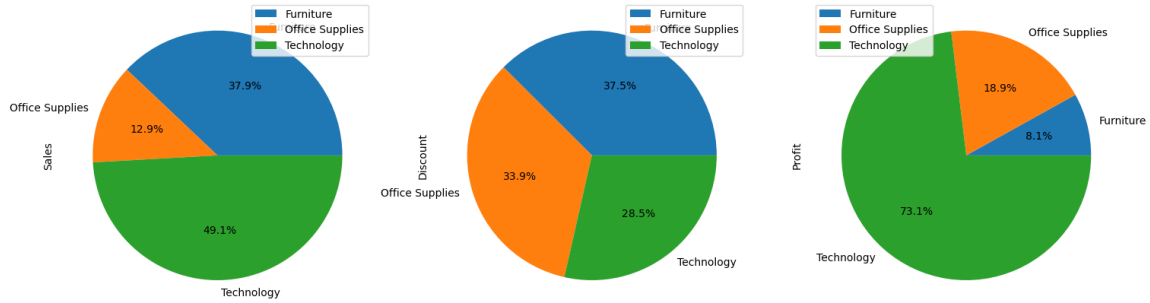
Category	Sales	Discount	Profit
Furniture	349.834887	0.173923	8.699327
Office Supplies	119.324101	0.157285	20.327050
Technology	452.709276	0.132323	78.752002

In [102]:

```
df_category.plot.pie(subplots=True,
                     figsize=(18, 20),
                     autopct='%1.1f%%',
                     labels = df_category.index)
```

Out[102]:

```
array([<AxesSubplot:ylabel='Sales'>, <AxesSubplot:ylabel='Discount'>,
       <AxesSubplot:ylabel='Profit'>], dtype=object)
```



Sub-Category wise Sales, Profit and Discount

In [107]:

```
df_sub_category = df.groupby(['Sub-Category'])[['Sales', 'Discount', 'Profit']].mean
df_sub_category.head(10)
```

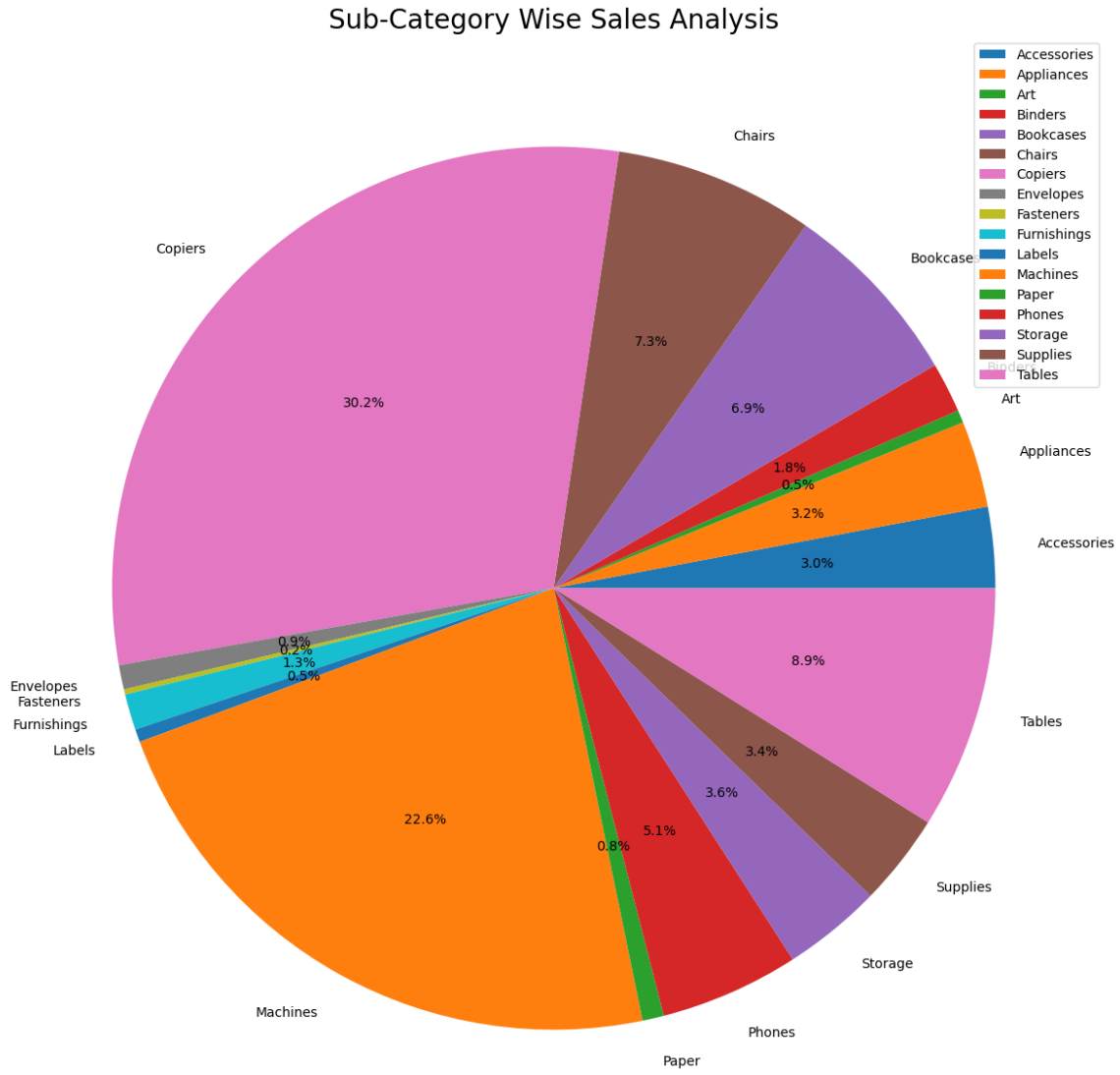
Out[107]:

Sub-Category	Sales	Discount	Profit
Accessories	215.974604	0.078452	54.111788
Appliances	230.755710	0.166524	38.922758
Art	34.068834	0.074874	8.200737
Binders	133.560560	0.372292	19.843574
Bookcases	503.859633	0.211140	-15.230509
Chairs	532.332420	0.170178	43.095894
Copiers	2198.941618	0.161765	817.909190
Envelopes	64.867724	0.080315	27.418019
Fasteners	13.936774	0.082028	4.375660
Furnishings	95.825668	0.138349	13.645918

[1] BASED ON THE SALES

In [108]:

```
plt.figure(figsize = (15,15))
plt.pie(df_sub_category['Sales'], labels = df_sub_category.index, autopct = '%1.1f%%
plt.title('Sub-Category Wise Sales Analysis', fontsize = 20)
plt.legend()
plt.xticks(rotation = 90)
plt.show()
```

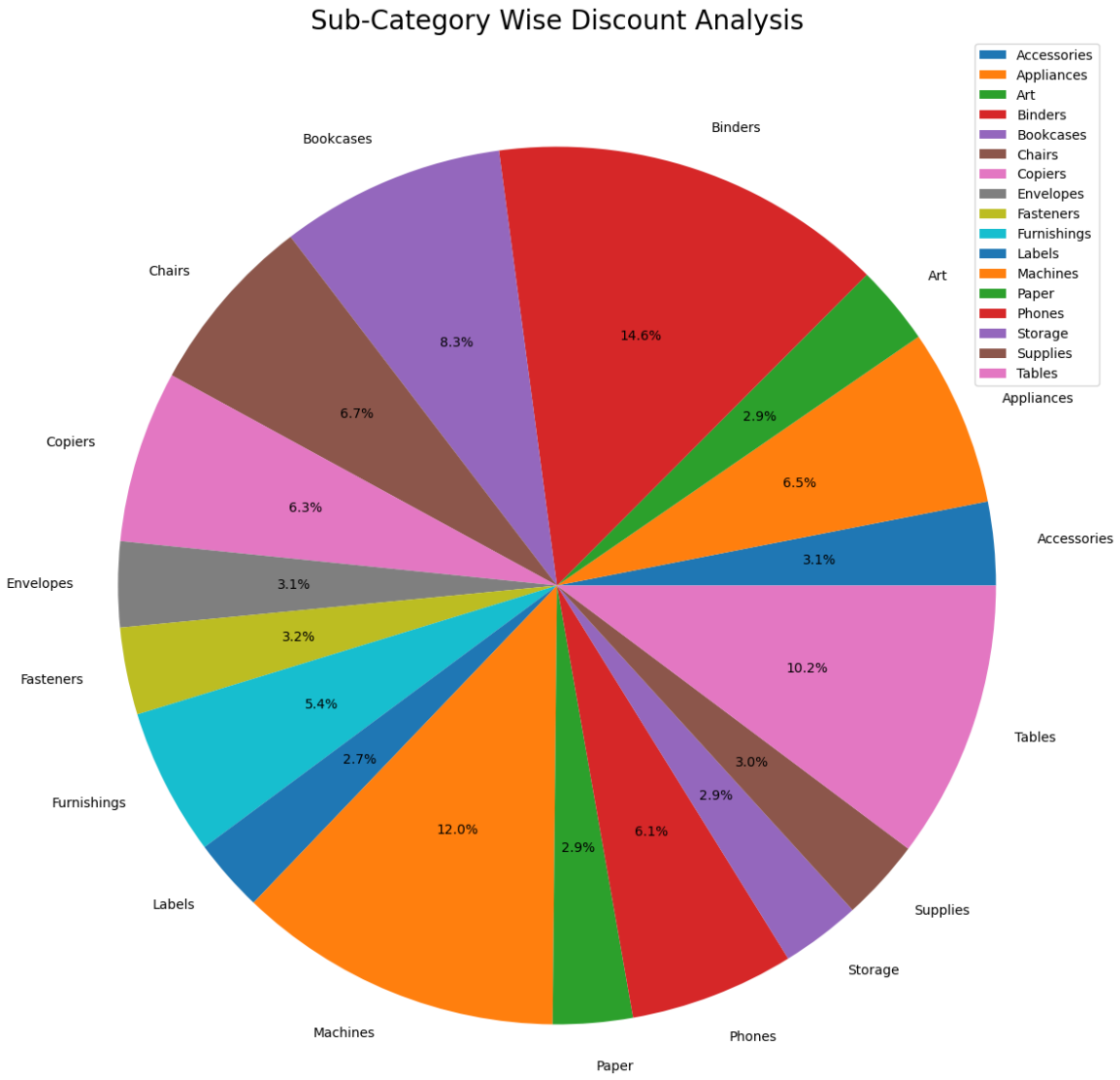


[2] BASED ON THE DISCOUNT

In [109]:

```
plt.figure(figsize = (15,15))
plt.pie(df_sub_category['Discount'], labels = df_sub_category.index, autopct = '%1.1
plt.title('Sub-Category Wise Discount Analysis', fontsize = 20)
plt.legend()
plt.xticks(rotation = 90)
plt.show()

```



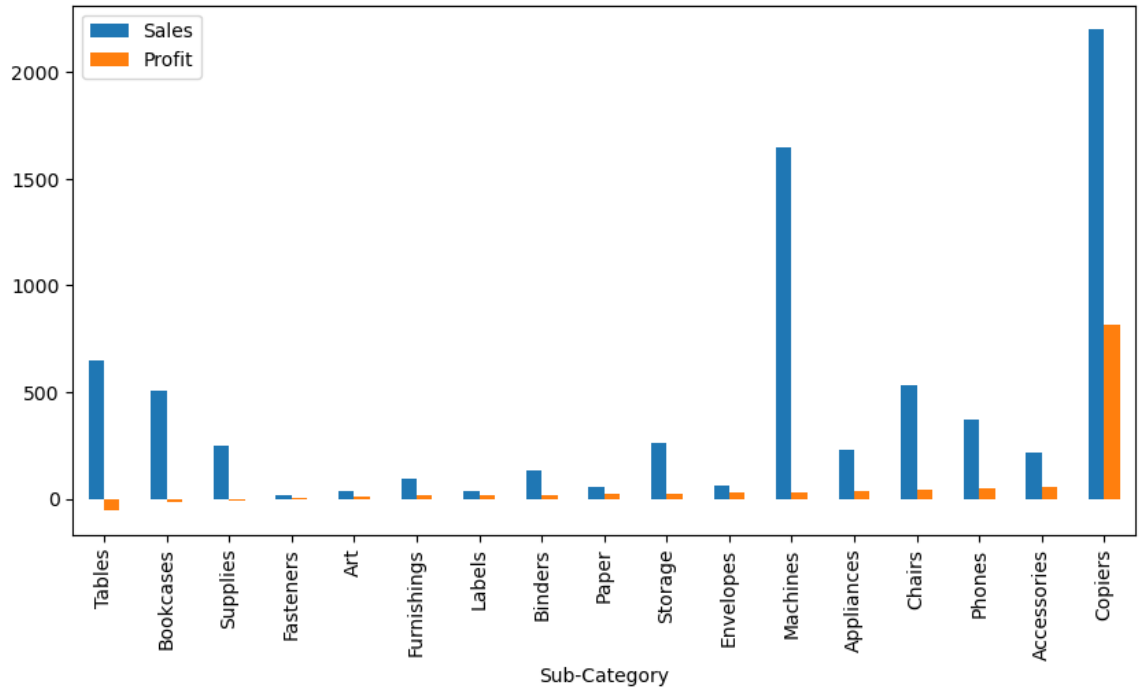
[3] BASED ON THE PROFIT

In [110]:

```
df_sub_category.sort_values('Profit')[['Sales','Profit']].plot(kind='bar',
                                                               figsize= (10,5),
                                                               label=['Avg Sales Pric
```

Out[110]:

<AxesSubplot:xlabel='Sub-Category'>



REGION WISE ANALYSIS

In [111]:

```
df_region = df.groupby(['Region'])[['Sales', 'Discount', 'Profit']].mean()
df_region
```

Out[111]:

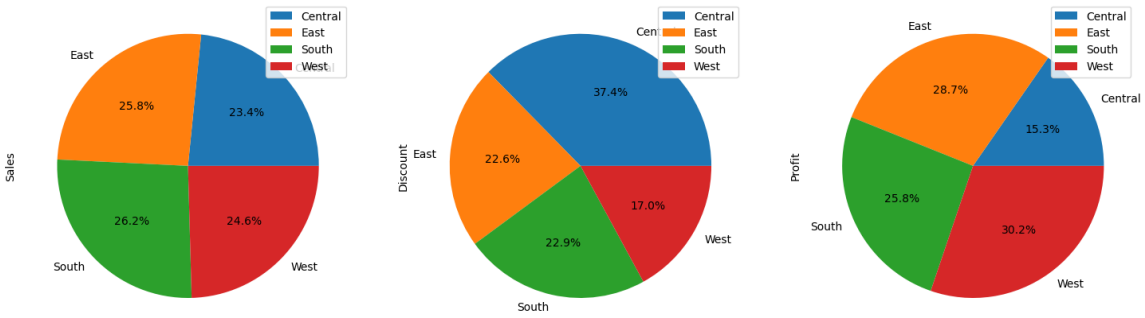
Region	Sales	Discount	Profit
Central	215.772661	0.240353	17.092709
East	238.336110	0.145365	32.135808
South	241.803645	0.147253	28.857673
West	226.493233	0.109335	33.849032

In [112]:

```
df_region.plot.pie(subplots=True,
                   figsize=(18, 20),
                   autopct='%1.1f%%',
                   labels = df_region.index)
```

Out[112]:

```
array([<AxesSubplot:ylabel='Sales'>, <AxesSubplot:ylabel='Discount'>,
       <AxesSubplot:ylabel='Profit'>], dtype=object)
```



SHIP MODE WISE ANALYSIS

In [113]:

```
df['Ship Mode'].value_counts()
```

Out[113]:

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

In [114]:

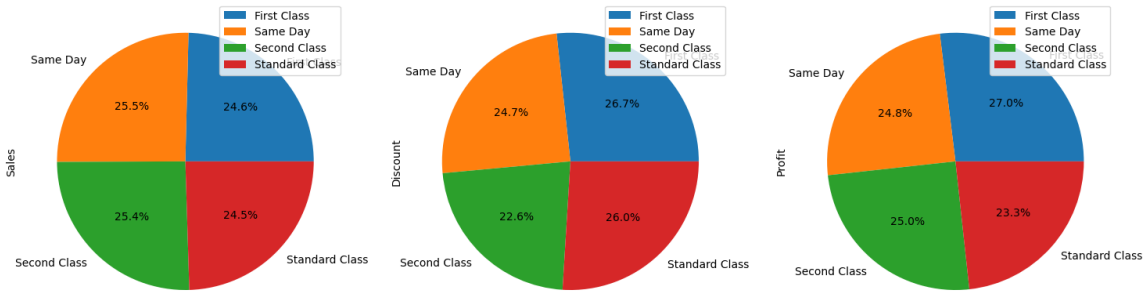
```
df_shipmode = df.groupby(['Ship Mode'])[['Sales', 'Discount', 'Profit']].mean()
```

In [115]:

```
df_shipmode.plot.pie(subplots=True,
                     figsize=(18, 20),
                     autopct='%1.1f%%',
                     labels = df_shipmode.index)
```

Out[115]:

```
array([<AxesSubplot:ylabel='Sales'>, <AxesSubplot:ylabel='Discount'>,
       <AxesSubplot:ylabel='Profit'>], dtype=object)
```



Profit and Discount is high in First Class
Sales is high for Same day ship

RESULT AND CONCLUSION

Profit is more than that of sale but there are some areas where profit could be increased.

Profit and Discount is high in First Class

Sales is high for Same day ship

Sub-category: Copier: High Profit & sales

Sub-category: Binders , Machines and then tables have high Discount.

Category: Maximun sales and Profit obtain in Technology.

Category: Minimun profit obtain in Furniture

State: Vermont: Highest Profit

State: Ohio: Lowest Profit

Segment: Home-office: High Profit & sales

Here is top 3 city where deals are Highest.

New York City

Los Angeles

Philadelphia

Sales and Profit are Moderately Correlated.

Quantity and Profit are less Moderately Correlated.

Discount and Profit are Negatively Correlated

Here is top 3 state where deals are Highest.

Califonia

New York

Texas

Wyoming : Lowest Number of deal,Highest amount of sales= Wyoming(11.8%)

Lowest amount of sales= South Dakota(0.8%)

THANK YOU