Author - AYUSH CHHOKER

DATA SCIENCE AND BUSINESS ANALYTICS INTERN

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.

Importing Libraries

In [1]:

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

In [6]:

```
df = pd.read_csv('D:\samstore.csv')  #Loading dataset
```

In [5]:

```
df.head()    #display top 5 rows
```

Out[5]:

	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 [4]:

```
df.tail()     #bottom 5 rows
```

Out[4]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Cat
9989	Second Class	Consumer	United States	Miami	Florida	33180	South	Furniture	Furnis
9990	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Furniture	Furnis
9991	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Technology	Pl
9992	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Office Supplies	I
9993	Second Class	Consumer	United States	Westminster	California	92683	West	Office Supplies	Appli

In [7]:

```
df.shape
```

Out[7]:

```
(9994, 13)
```

In [8]:

```
df.describe()       #display summary
```

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

In [9]:

```
df.isnull().sum()       #checking null values
```

Out[9]:

```
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 [10]:

```
df.info()           #information about dataset
```

```
<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 [11]:

```
df.columns
```

Out[11]:

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

In [12]:

```
df.duplicated().sum()
```

Out[12]:

```
17
```

In [13]:

```
df.nunique()
```

Out[13]:

```
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 [14]:

```
df['Postal Code'] = df['Postal Code'].astype('object')
```

In [15]:

```
df.drop_duplicates(subset=None,keep='first',inplace=True)
df.duplicated().sum()
```

Out[15]:

```
0
```

In [16]:

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

Out[16]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c0fb5b7640>
```

In [17]:

```
df = df.drop(['Postal Code'],axis = 1)    #dropping postal code columns
```

In [18]:

```
sns.pairplot(df, hue = 'Ship Mode')
```

Out[18]:

```
<seaborn.axisgrid.PairGrid at 0x1c0fb581b20>
```

In [19]:

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

Out[19]:

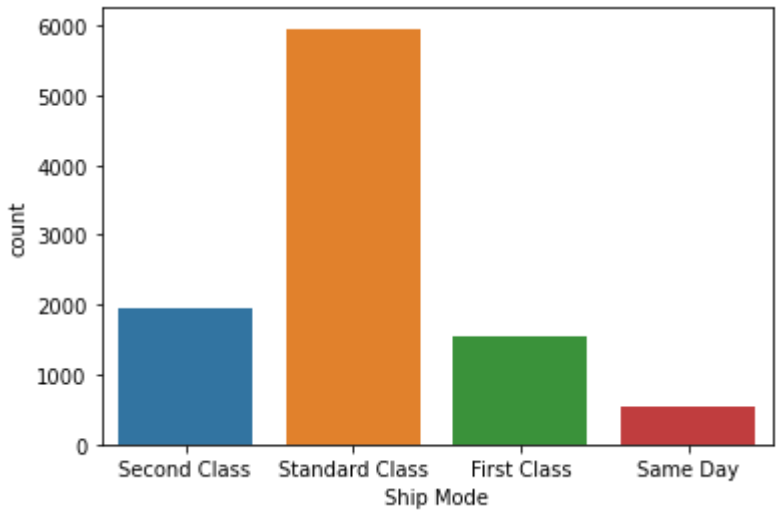
```
Standard Class    5955
Second Class      1943
First Class       1537
Same Day           542
Name: Ship Mode, dtype: int64
```

In [20]:

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

Out[20]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c0fc1eee50>
```



In [21]:

```
df['Segment'].value_counts()        #valuecounts for segment
```

Out[21]:

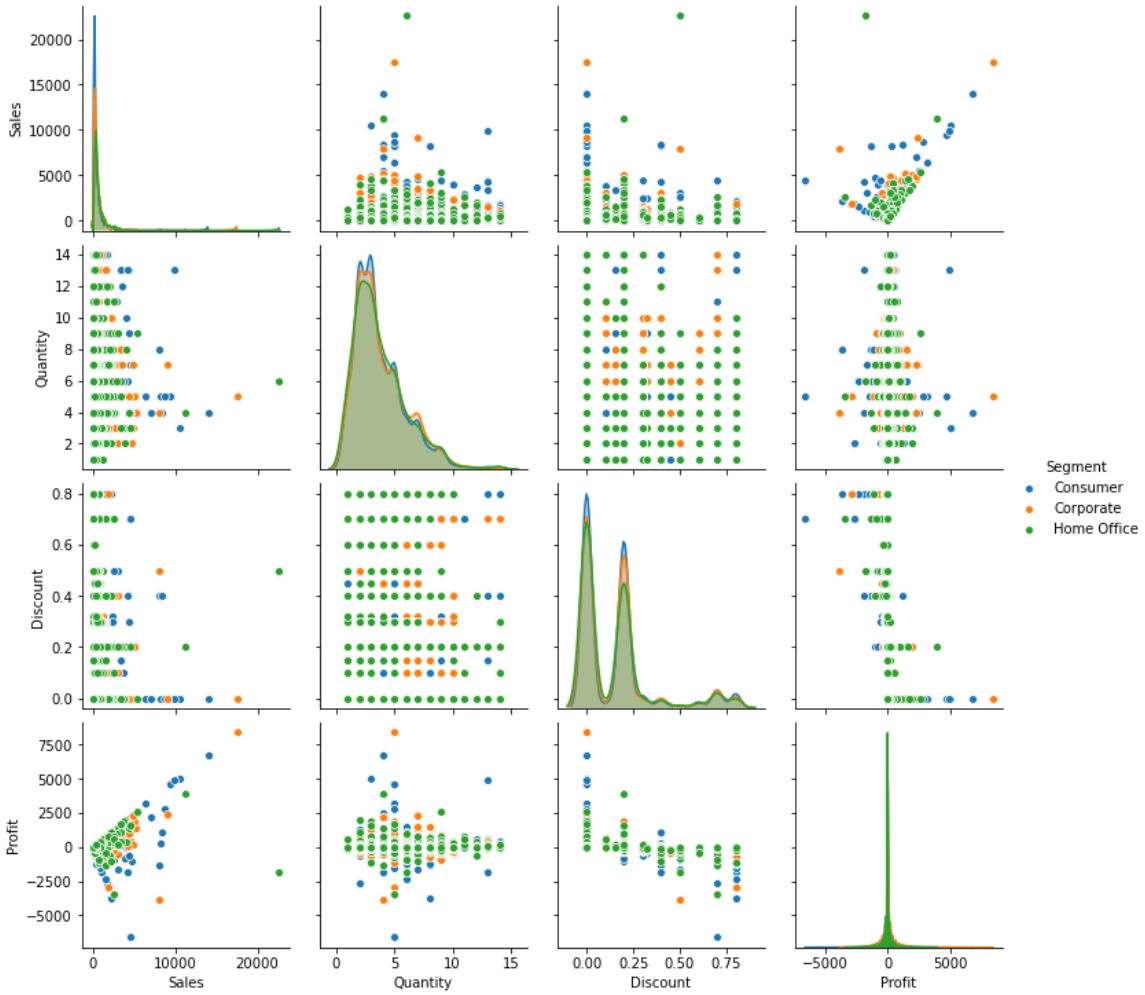
```
Consumer       5183
Corporate      3015
Home Office    1779
Name: Segment, dtype: int64
```

In [22]:

```
sns.pairplot(df,hue = 'Segment')      #plotting pair plot
```

Out[22]:

```
<seaborn.axisgrid.PairGrid at 0x1c0fc1ee2b0>
```

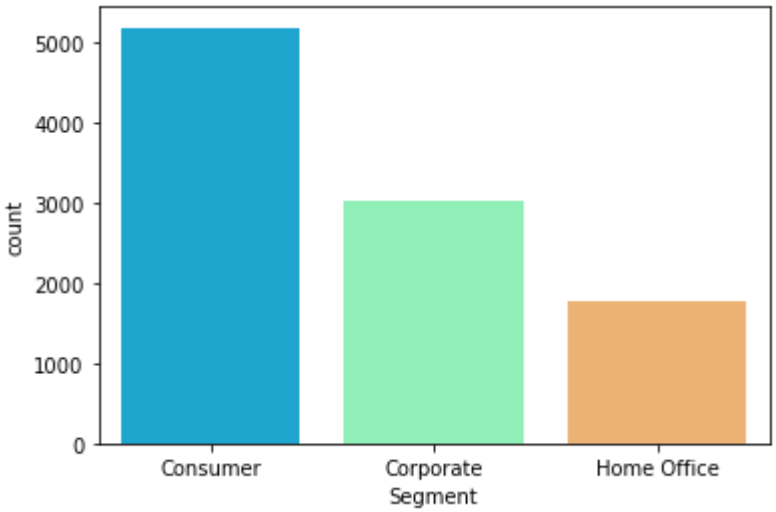


In [23]:

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

Out[23]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c0fc08f9a0>
```



In [24]:

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

Out[24]:

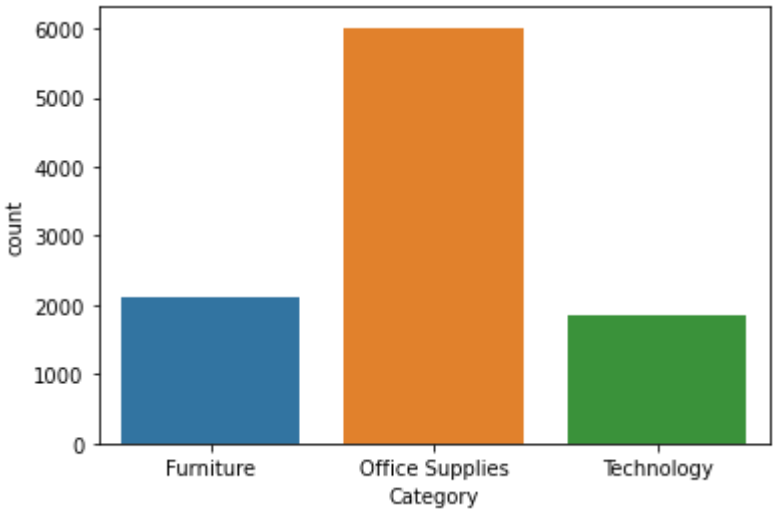
```
Office Supplies    6012
Furniture          2118
Technology         1847
Name: Category, dtype: int64
```

In [25]:

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

Out[25]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x1c0fc0cbe20>
```

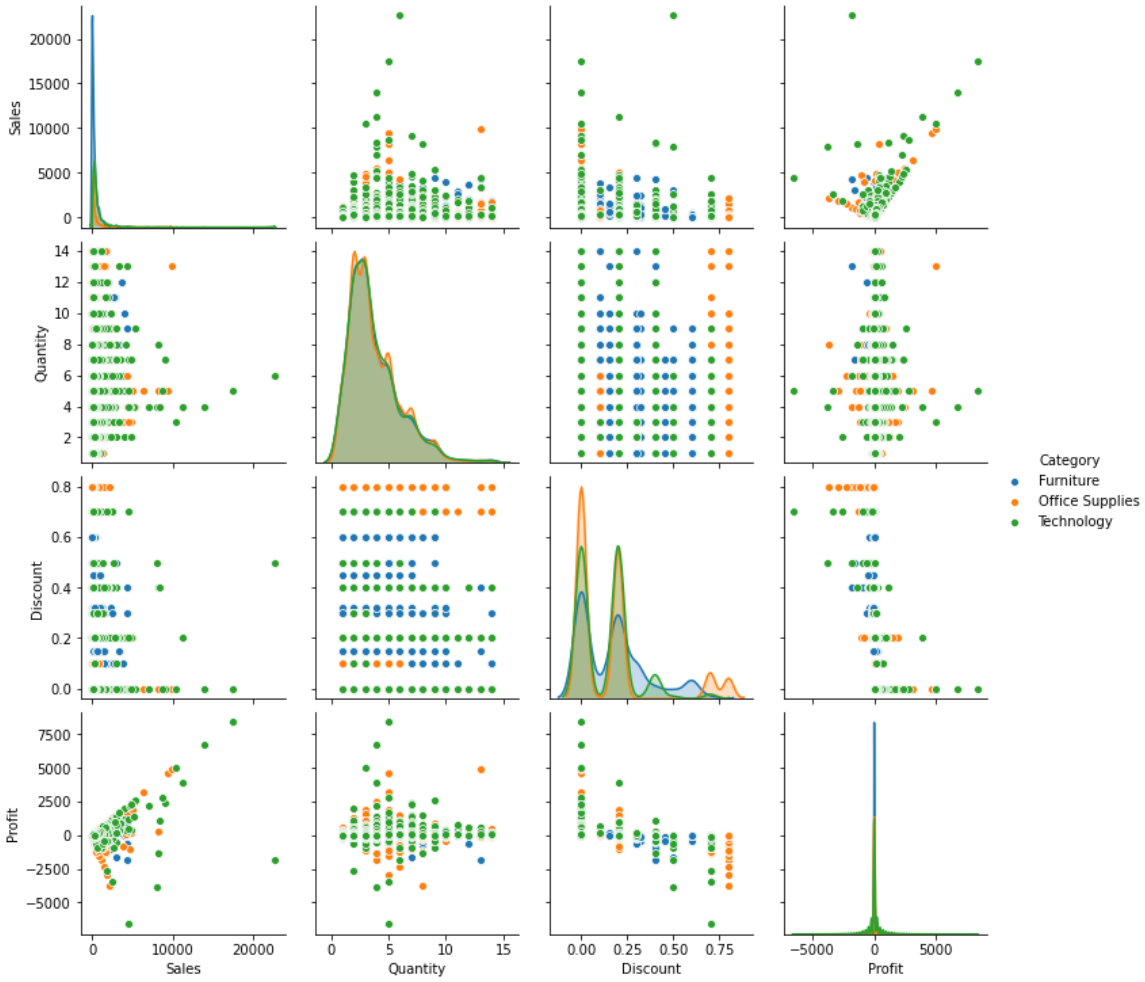


In [26]:

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

Out[26]:

```
<seaborn.axisgrid.PairGrid at 0x1c0fc0f8f40>
```



In [27]:

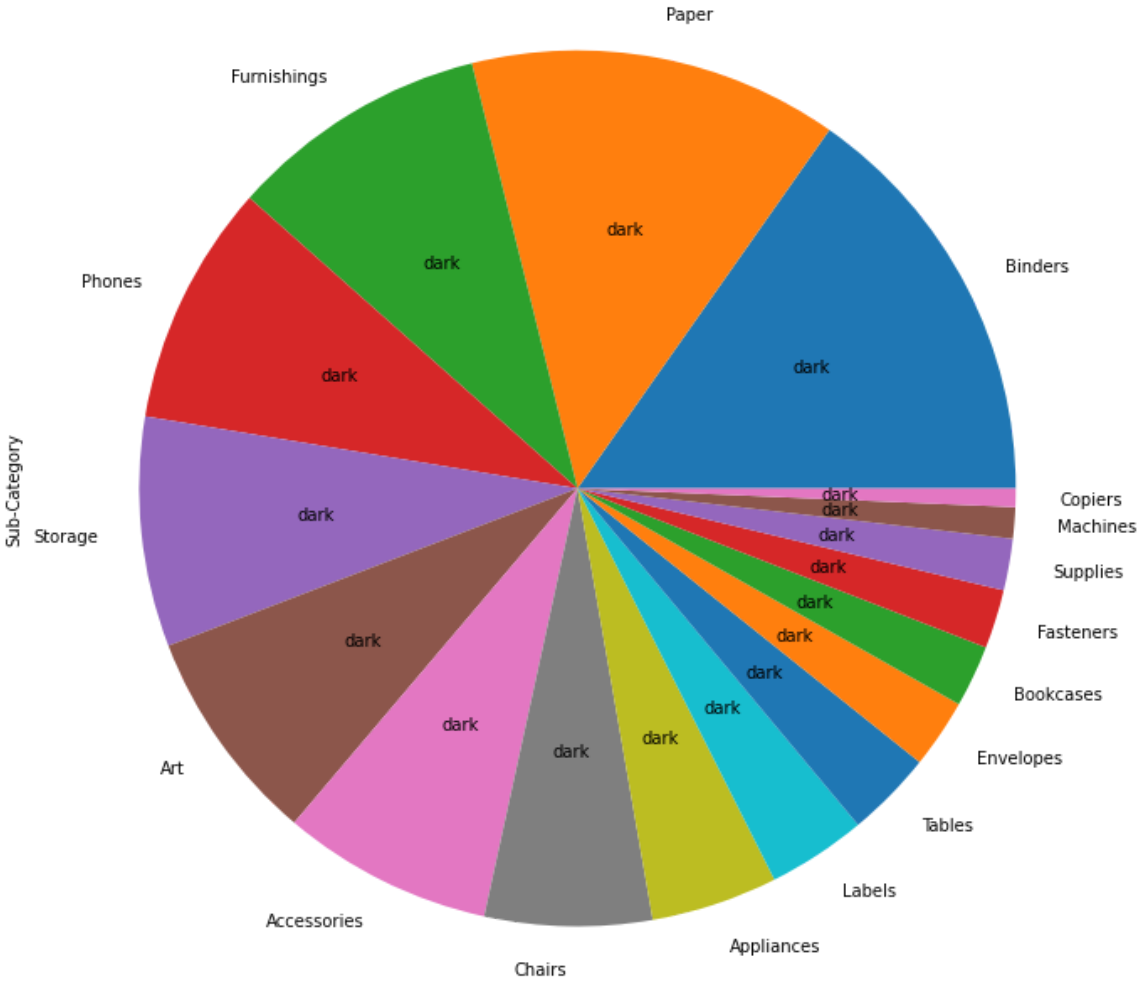
```
df['Sub-Category'].value_counts()
```

Out[27]:

```
Binders        1522
Paper          1359
Furnishings     956
Phones          889
Storage         846
Art             795
Accessories     775
Chairs          615
Appliances      466
Labels          363
Tables          319
Envelopes       254
Bookcases       228
Fasteners       217
Supplies        190
Machines        115
Copiers          68
Name: Sub-Category, dtype: int64
```

In [28]:

```
plt.figure(figsize=(15,12))
df['Sub-Category'].value_counts().plot.pie(autopct='dark')
plt.show()
```


Paper
Furnishings
Phones
Sub-Category
Storage
Art
Accessories
Chairs
Appliances
Labels
Tables
Envelopes
Bookcases
Fasteners
Supplies
Machines
Copiers
Binders
dark

Observation 1

Maximum are from Binders, Paper, furnishings, Phones, storage, art, accessories and minimum from copiers, machines, suppliers

In [29]:

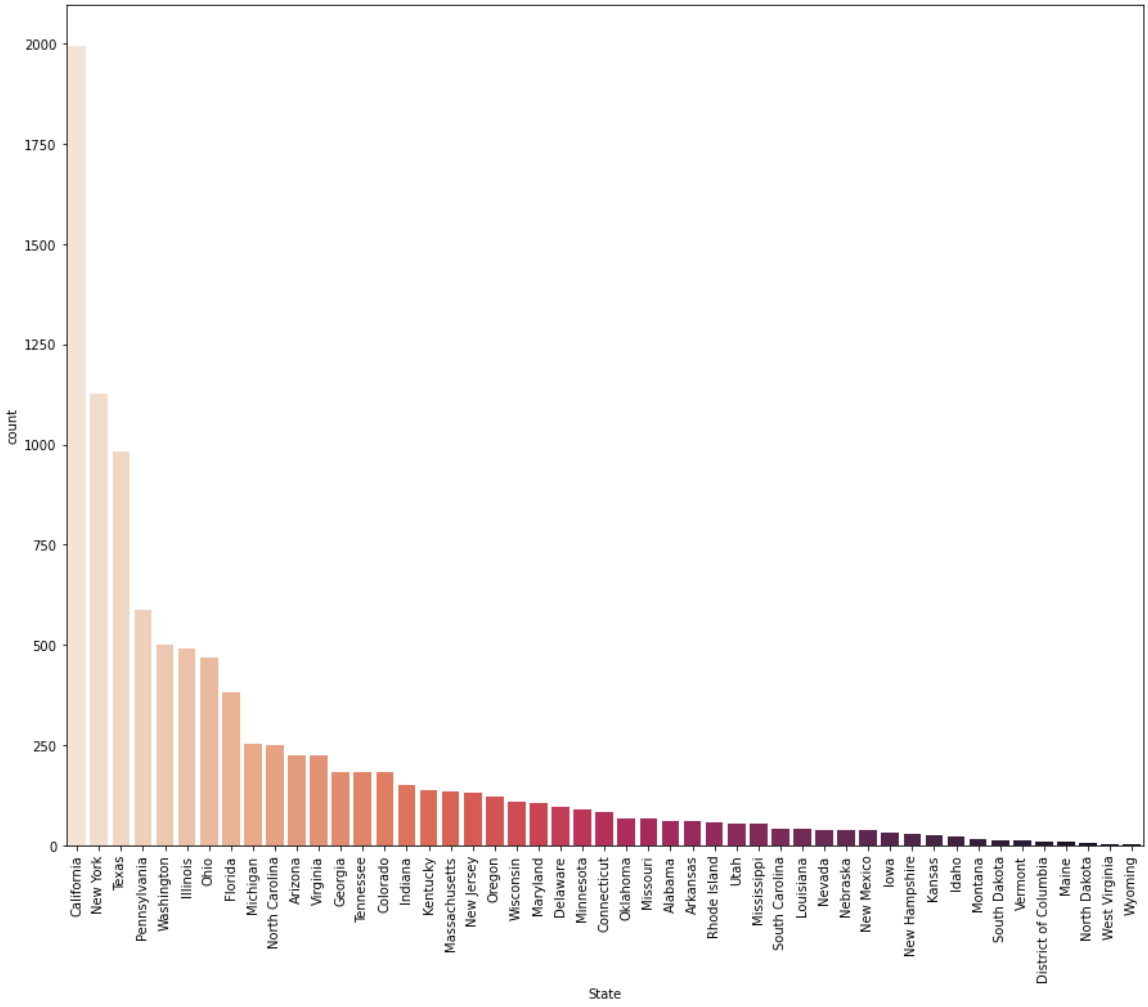
```
df['State'].value_counts()
```

Out[29]:

```
California              1996
New York                1127
Texas                    983
Pennsylvania             586
Washington               502
Illinois                 491
Ohio                     468
Florida                  383
Michigan                 254
North Carolina           249
Arizona                  224
Virginia                 224
Georgia                  184
Tennessee                183
Colorado                 182
Indiana                  149
Kentucky                 139
Massachusetts            135
New Jersey               130
Oregon                   123
Wisconsin                110
Maryland                 105
Delaware                  96
Minnesota                 89
Connecticut               82
Oklahoma                  66
Missouri                  66
Alabama                   61
Arkansas                  60
Rhode Island              56
Utah                      53
Mississippi               53
South Carolina            42
Louisiana                 42
Nevada                    39
Nebraska                  38
New Mexico                37
Iowa                      30
New Hampshire             27
Kansas                    24
Idaho                     21
Montana                   15
South Dakota              12
Vermont                   11
District of Columbia      10
Maine                      8
North Dakota               7
West Virginia              4
Wyoming                    1
Name: State, dtype: int64
```

In [30]:

```
plt.figure(figsize=(15,12))
sns.countplot(x='State',data=df,palette='rocket_r',order=df['State'].value_counts().ind
ex)
plt.xticks(rotation=90)
plt.show()
```

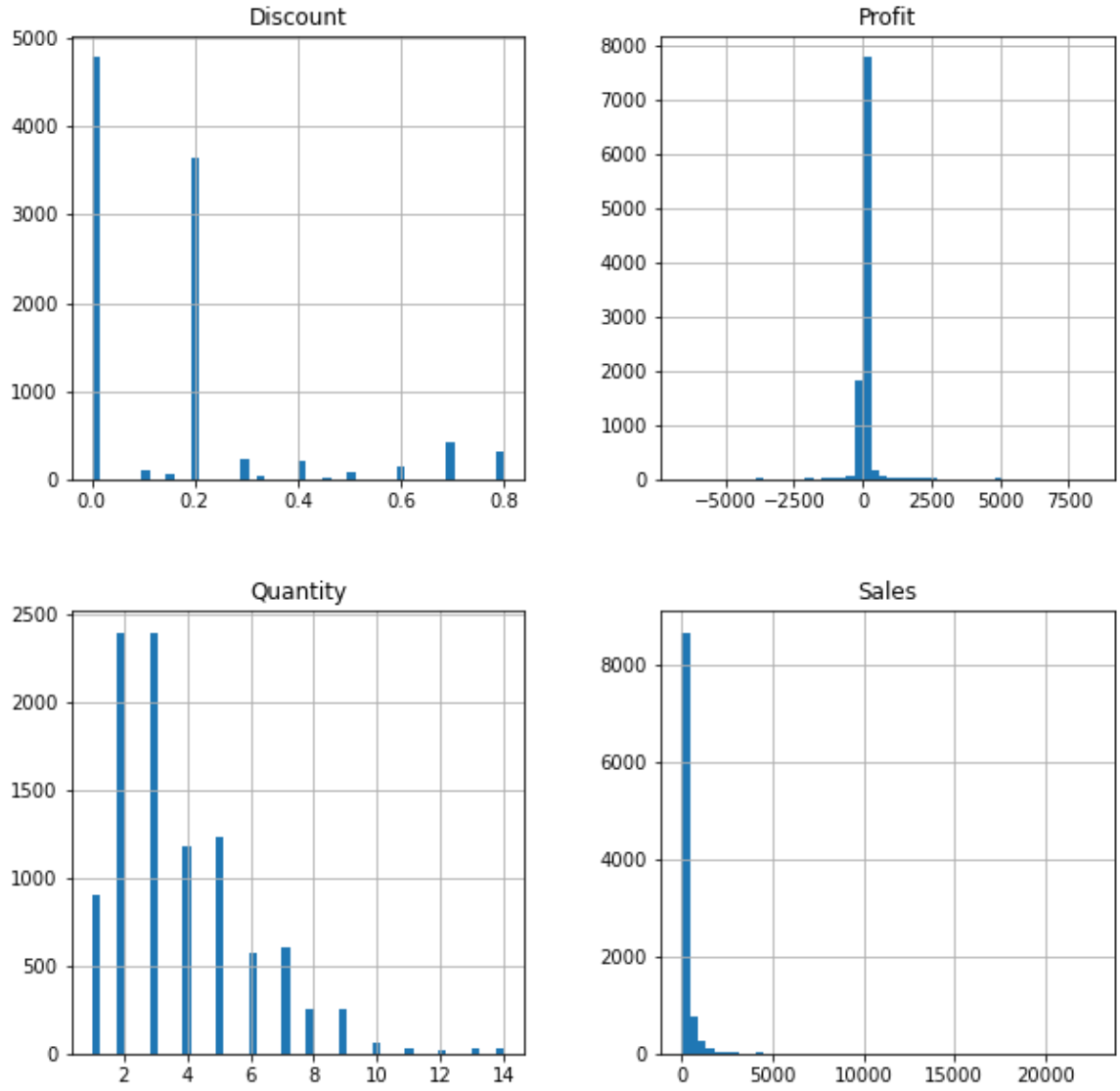


observation 2

Highest number of buyers are from California and New York

In [31]:

```
df.hist(figsize=(10,10),bins=50)
plt.show()
```

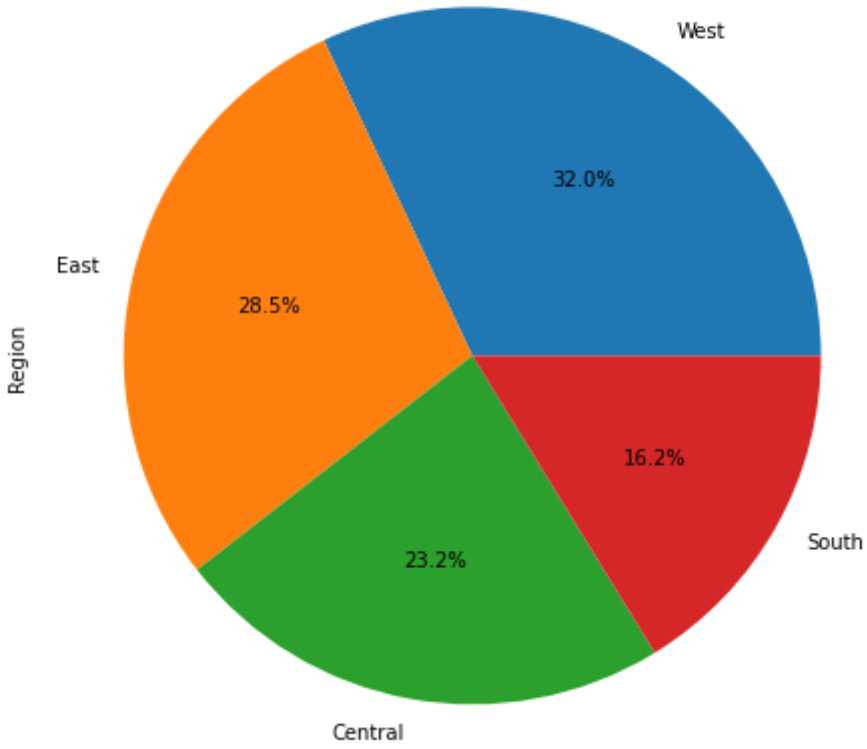


observation 3

1. Most customers tends to buy quantity of 2 and 3
2. Discount give maximum is 0 to 20 percent

In [32]:

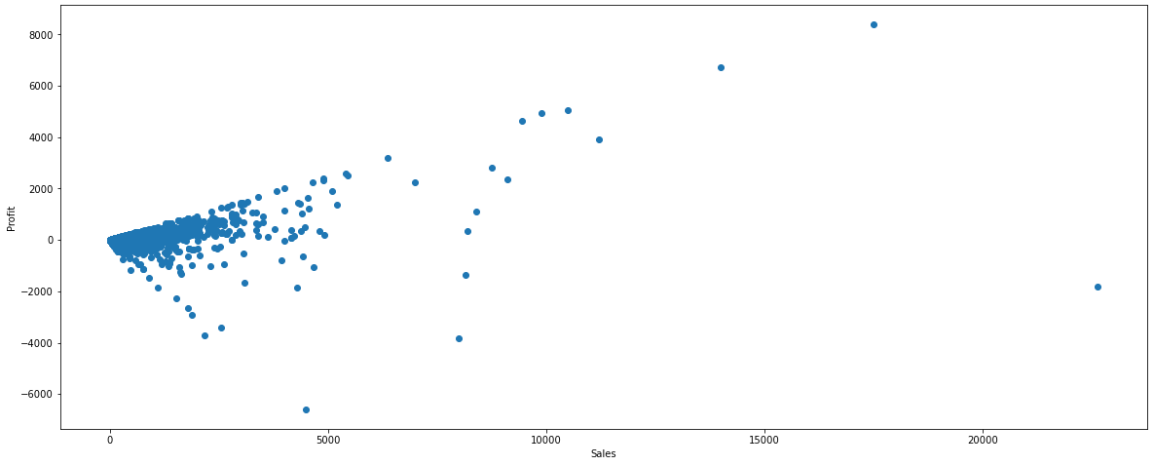
```
plt.figure(figsize=(10,8))
df['Region'].value_counts().plot.pie(autopct = '%1.1f%%')
plt.show()
```



profit vs discount

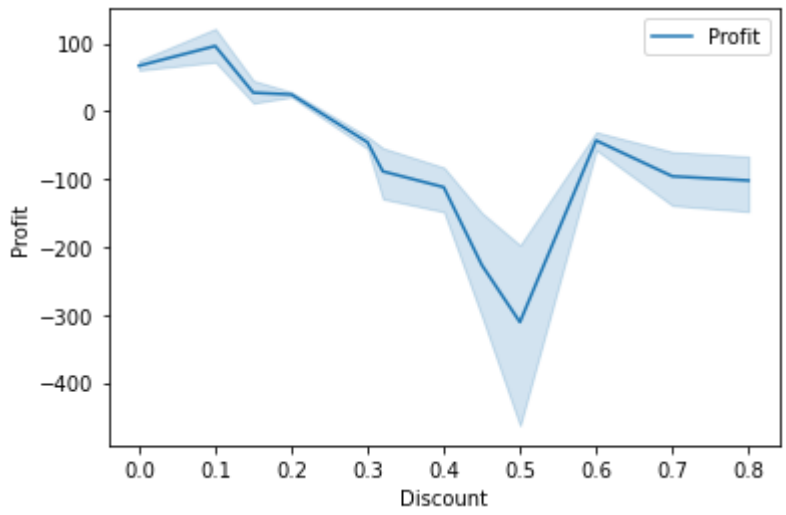
In [33]:

```
fig,ax=plt.subplots(figsize=(20,8))
ax.scatter(df['Sales'],df['Profit'])
ax.set_xlabel('Sales')
ax.set_ylabel('Profit')
plt.show()
```



In [34]:

```
sns.lineplot(x='Discount',y='Profit',label='Profit',data=df)
plt.legend()
plt.show()
```



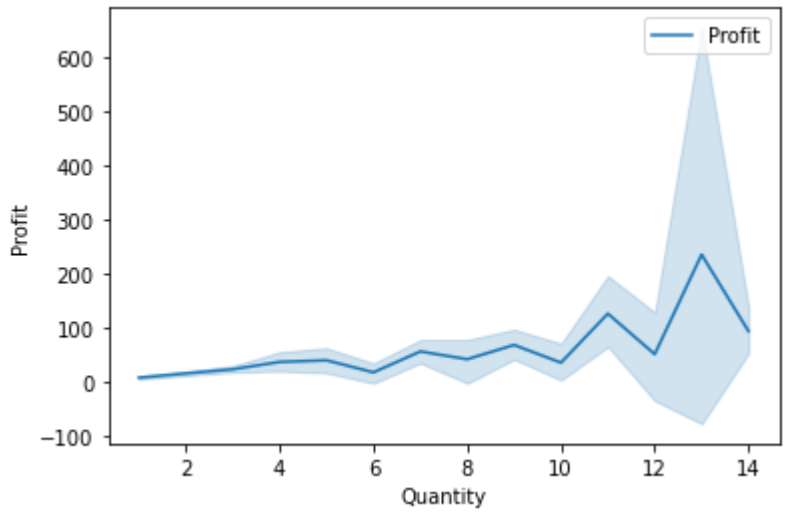
observation 4

No correlation between profit and discount

Profit vs Quantity

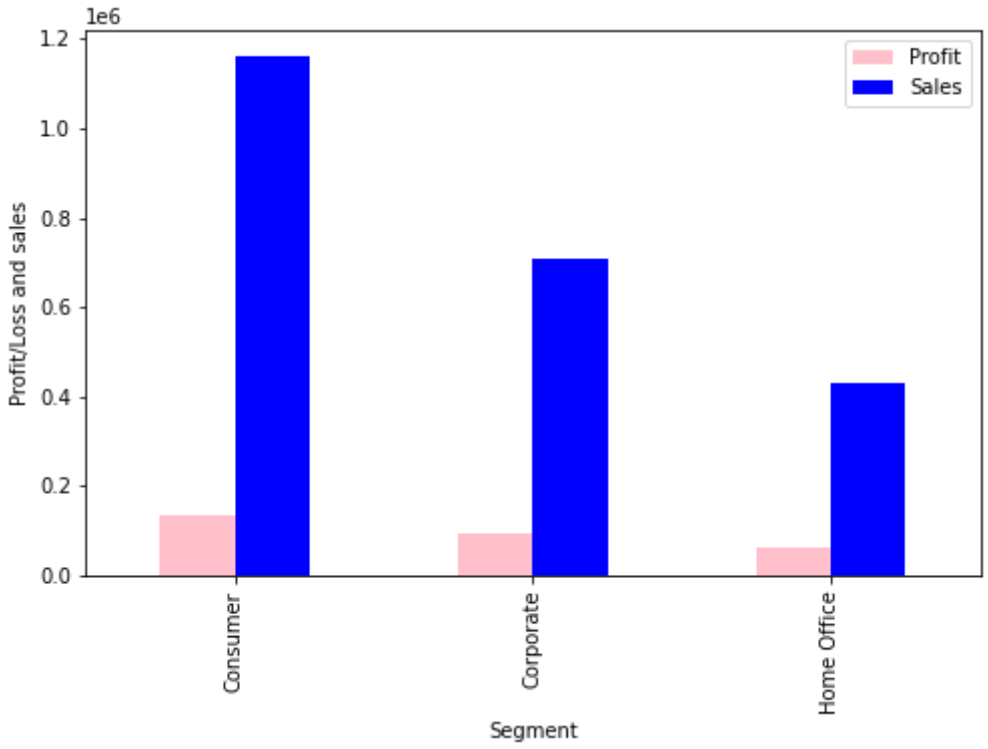
In [35]:

```
sns.lineplot(x='Quantity',y='Profit',label='Profit',data=df)
plt.legend()
plt.show()
```



In [36]:

```
df.groupby('Segment')[['Profit','Sales']].sum().plot.bar(color=['pink','blue'],figsize=
(8,5))
plt.ylabel('Profit/Loss and sales')
plt.show()
```

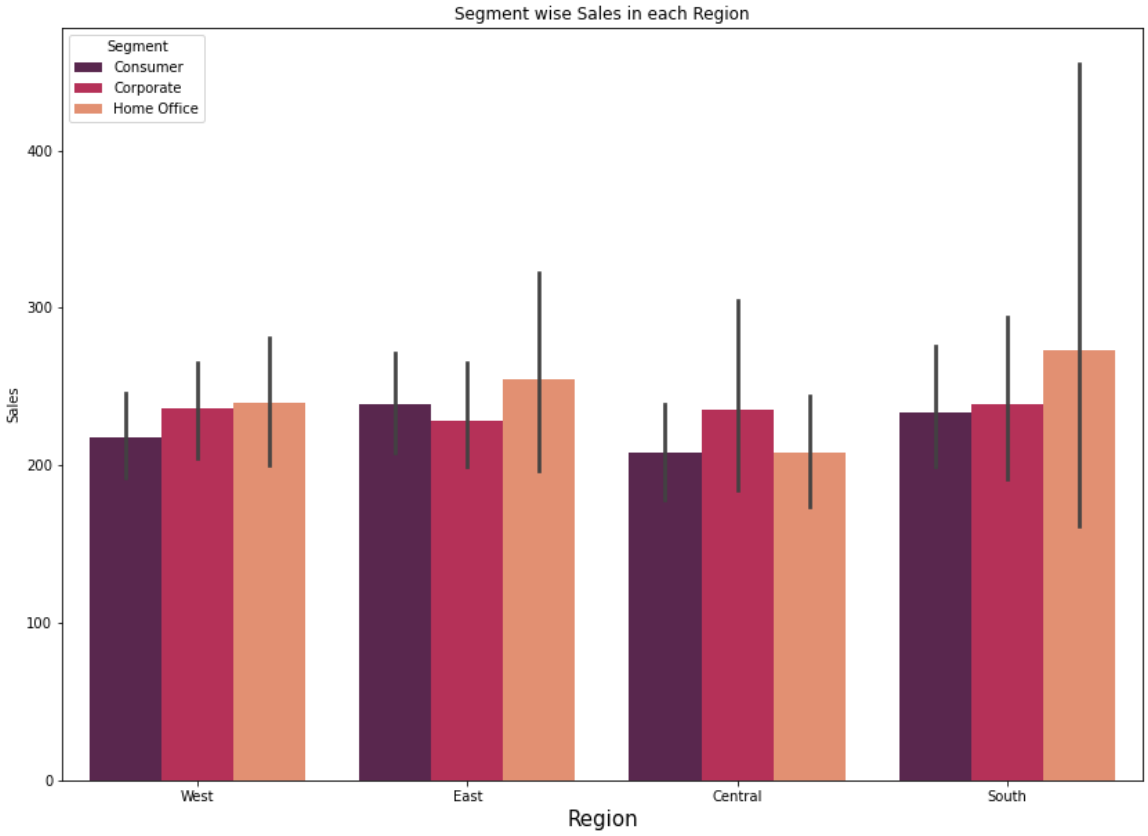


observation 5

Profit and sales are maximum in consumer segment and minimum in Home Office segment

In [39]:

```
plt.figure(figsize=(14,10))
plt.title('Segment wise Sales in each Region')
sns.barplot(x='Region',y='Sales',data=df,hue='Segment',order=df['Region'].value_counts
().index,palette='rocket')
plt.xlabel('Region',fontsize=15)
plt.show()
```

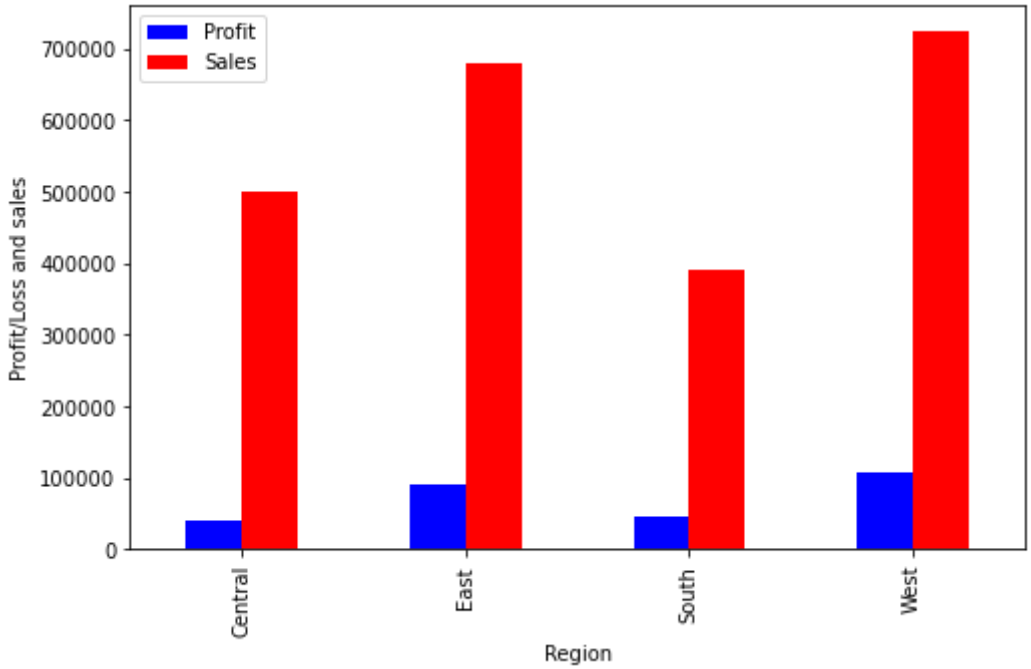


observation 6

Segment wise sales are almost same in every region

In [40]:

```
df.groupby('Region')[['Profit','Sales']].sum().plot.bar(color=['blue','red'],figsize=(8
,5))
plt.ylabel('Profit/Loss and sales')
plt.show()
```

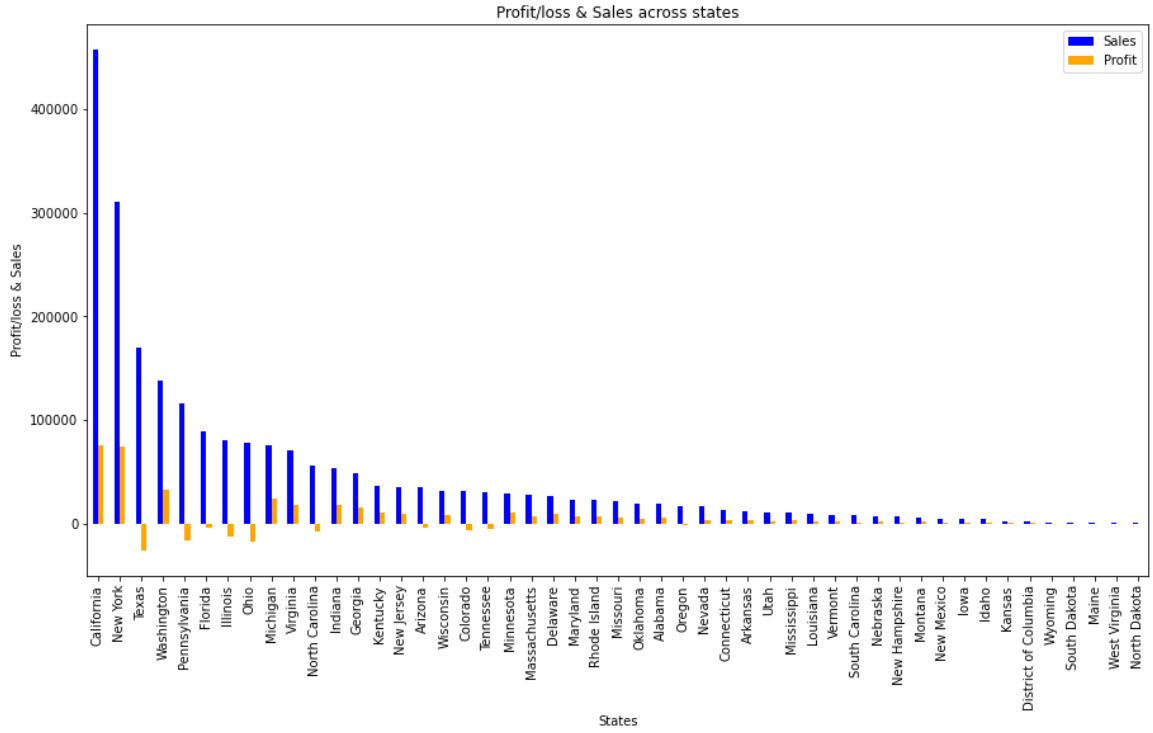


Observation 7

Profit and sales are maximum in west region and minimum in south region

In [41]:

```
ps = df.groupby('State')[['Sales','Profit']].sum().sort_values(by='Sales',ascending=False)
ps[:].plot.bar(color=['blue','orange'],figsize=(15,8))
plt.title('Profit/loss & Sales across states')
plt.xlabel('States')
plt.ylabel('Profit/loss & Sales')
plt.show()
```



observation 8

high profit is for california, new york loss is for texas, pennsylvania, Ohio

In [42]:

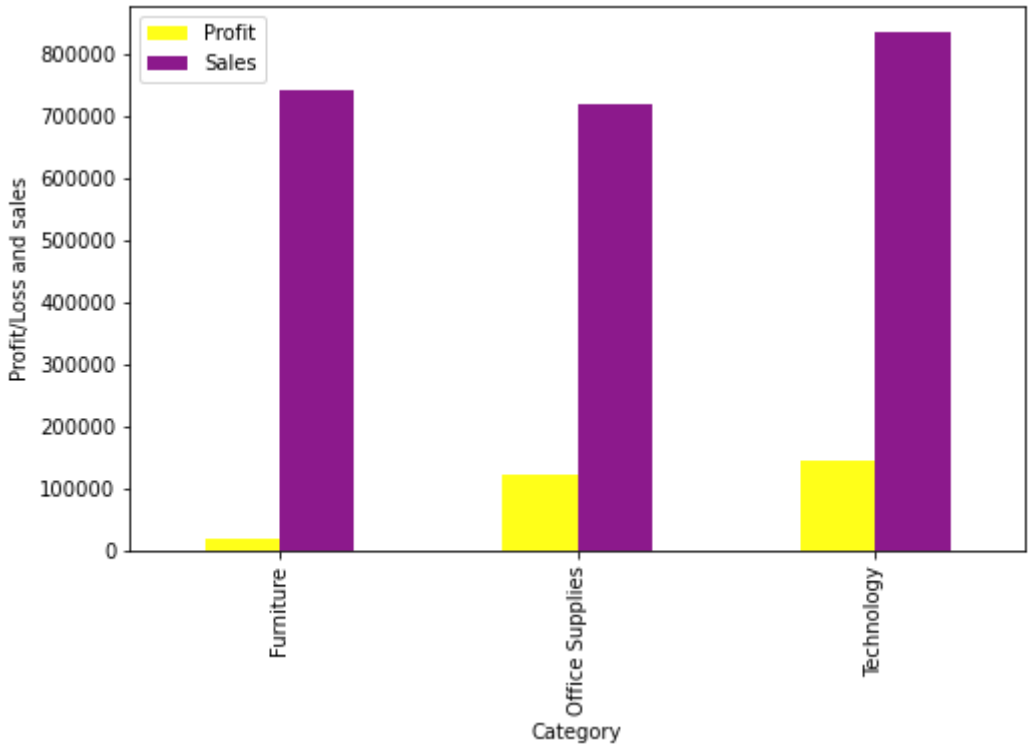
```
t_states = df['State'].value_counts().nlargest(10)
t_states
```

Out[42]:

```
California        1996
New York          1127
Texas              983
Pennsylvania       586
Washington         502
Illinois           491
Ohio               468
Florida            383
Michigan           254
North Carolina     249
Name: State, dtype: int64
```

In [43]:

```
df.groupby('Category')[['Profit','Sales']].sum().plot.bar(color=['yellow','purple'],alp
ha=0.9,figsize=(8,5))
plt.ylabel('Profit/Loss and sales')
plt.show()
```



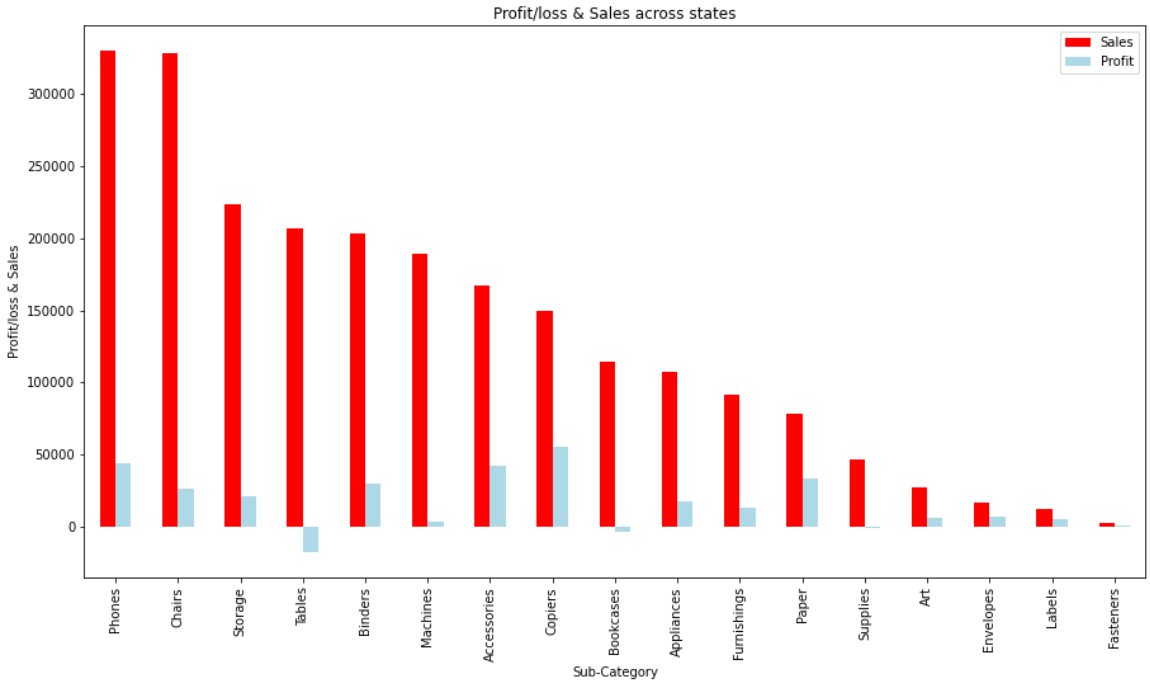
Observation 9

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

1. Technology and Office Supplies have high profit.
2. urniture have less profit

In [44]:

```
ps = df.groupby('Sub-Category')[['Sales','Profit']].sum().sort_values(by='Sales',ascend
ing=False)
ps[:].plot.bar(color=['red','lightblue'],figsize=(15,8))
plt.title('Profit/loss & Sales across states')
plt.xlabel('Sub-Category')
plt.ylabel('Profit/loss & Sales')
plt.show()
```



observation 10

1. Phones sub-category have high sales.
2. chairs have high sales but less profit compared to phones
3. Tables and Bookmarks sub-categories facing huge loss

In [ ]: