THE SPARKS FOUNDATION

Data Science and Buisness Analytics Tasks

NAME : M.NIVETHITHAA

TASK 3 - Exploratory Data Analysis - Retail

DATA AUDIT

Importing Libraries

In [1]:
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import folium
```

Displaying Raw Dataset

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

Out[2]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Prof
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	0.00	41.913
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9400	3	0.00	219.582
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14.6200	2	0.00	6.871
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5775	5	0.45	-383.031
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.3680	2	0.20	2.516
...	...	...	...	...	...	...	...	...	...	...	...	...	.
9989	Second Class	Consumer	United States	Miami	Florida	33180	South	Furniture	Furnishings	25.2480	3	0.20	4.102
9990	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Furniture	Furnishings	91.9600	2	0.00	15.633
9991	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Technology	Phones	258.5760	2	0.20	19.393
9992	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Office Supplies	Paper	29.6000	4	0.00	13.320
9993	Second Class	Consumer	United States	Westminster	California	92683	West	Office Supplies	Appliances	243.1600	2	0.00	72.948

9994 rows × 13 columns

First five rows of the dataset

In [3]:
```
data.head(5)
```

Out[3]:

	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

Last five rows of the dataset

In [4]:
```
data.tail(5)
```

Out[4]:

	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

Shape of the dataset

In [5]:
```
data.shape
```

Out[5]:
```
(9994, 13)
```

Columns present in the dataset

In [6]:
```
data.columns
```

Out[6]:
```
Index(['Ship Mode', 'Segment', 'Country', 'City', 'State', 'Postal Code',
       'Region', 'Category', 'Sub-Category', 'Sales', 'Quantity', 'Discount',
       'Profit'],
      dtype='object')
```

Summary of the dataset

In [7]:
```
data.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 [8]:
```
data.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

Checking Datatypes

In [9]:
```
data.dtypes
```

Out[9]:
```
Ship Mode        object
Segment          object
Country          object
City             object
State            object
Postal Code       int64
Region           object
Category         object
Sub-Category     object
Sales           float64
Quantity          int64
Discount        float64
Profit          float64
dtype: object
```

Checking missing values

In [10]:
```
data.isna().sum()
```

Out[10]:
```
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
```

No Missing values

Checking Duplicates

In [11]:
```
data.duplicated().sum()
```

Out[11]:
```
17
```

Removing duplicates

In [12]:
```
data.drop_duplicates(keep=False,inplace=True)
data
```

Out[12]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Prof
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	0.00	41.913
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9400	3	0.00	219.582
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14.6200	2	0.00	6.871
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5775	5	0.45	-383.031
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.3680	2	0.20	2.516
...	...	...	...	...	...	...	...	...	...	...	...	...	.
9989	Second Class	Consumer	United States	Miami	Florida	33180	South	Furniture	Furnishings	25.2480	3	0.20	4.102
9990	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Furniture	Furnishings	91.9600	2	0.00	15.633
9991	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Technology	Phones	258.5760	2	0.20	19.393
9992	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Office Supplies	Paper	29.6000	4	0.00	13.320
9993	Second Class	Consumer	United States	Westminster	California	92683	West	Office Supplies	Appliances	243.1600	2	0.00	72.948

9960 rows × 13 columns

In [13]:
```
data.duplicated().sum()
```

Out[13]:
```
0
```

No duplicates

CORRELATION

In [14]:
```
data.corr()
```

Out[14]:

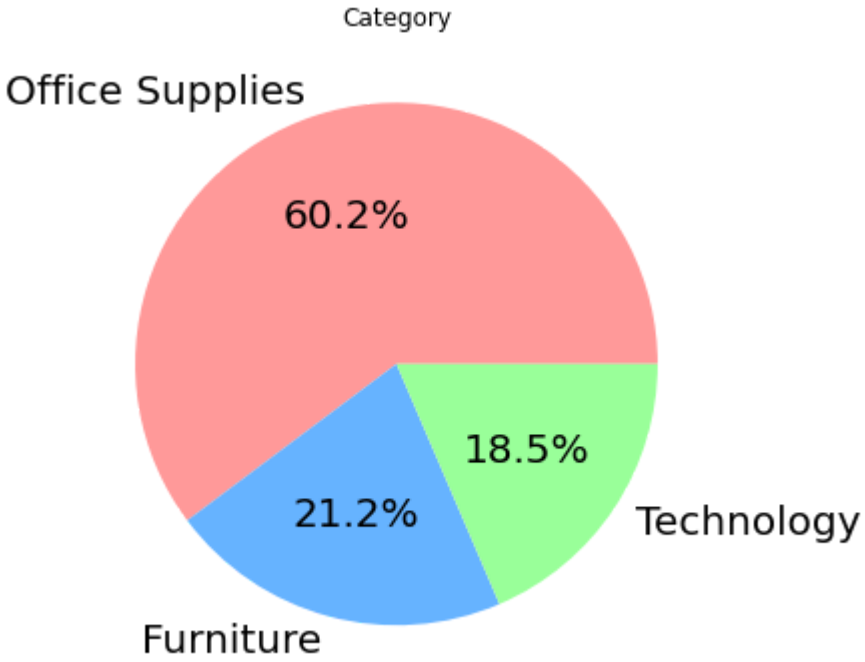
	Postal Code	Sales	Quantity	Discount	Profit
Postal Code	1.000000	-0.023096	0.013461	0.060012	-0.029823
Sales	-0.023096	1.000000	0.200649	-0.028433	0.479070
Quantity	0.013461	0.200649	1.000000	0.008734	0.066168
Discount	0.060012	-0.028433	0.008734	1.000000	-0.219837
Profit	-0.029823	0.479070	0.066168	-0.219837	1.000000

In [15]:
```
sns.heatmap(data.corr(), cmap="YlGnBu", annot = True)
plt.show()
```

PLOTTING

Pie Chart - Plotting Category

In [16]:
```
plt.figure(figsize = (6,6))
textprops = {"fontsize" : 20}
colors = ['#ff9999','#66b3ff','#99ff99']

plt.title('Category')
plt.pie(data['Category'].value_counts(), labels = data['Category'].value_counts().index, autopct='%1.1f%%', text
plt.show()
```



Result: More products are available in Office Supplies Category

Stack bar - Category and Region wise Sales

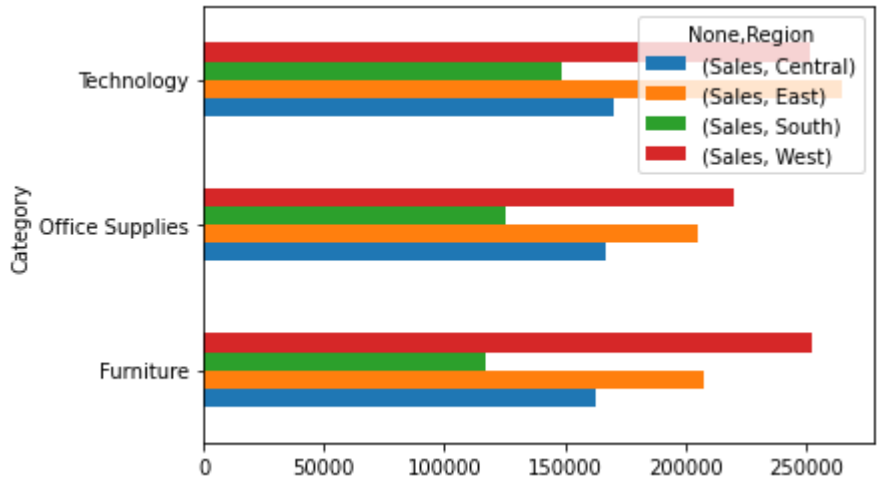
In [17]:
```
data1 = pd.pivot_table(data,index=['Category'], columns=['Region'], values=['Sales'], aggfunc='sum')
data1
```

Out[17]:

	Sales			
Region	**Central**	**East**	**South**	**West**
Category				
Furniture	163017.2238	207728.460	117298.684	252568.4635
Office Supplies	166892.2790	205386.711	125651.313	220493.1530
Technology	170416.3120	264973.981	148771.908	251991.8320

In [18]:
```
data1.plot.barh(stacked=False)
```

Out[18]: <AxesSubplot:ylabel='Category'>



Result:

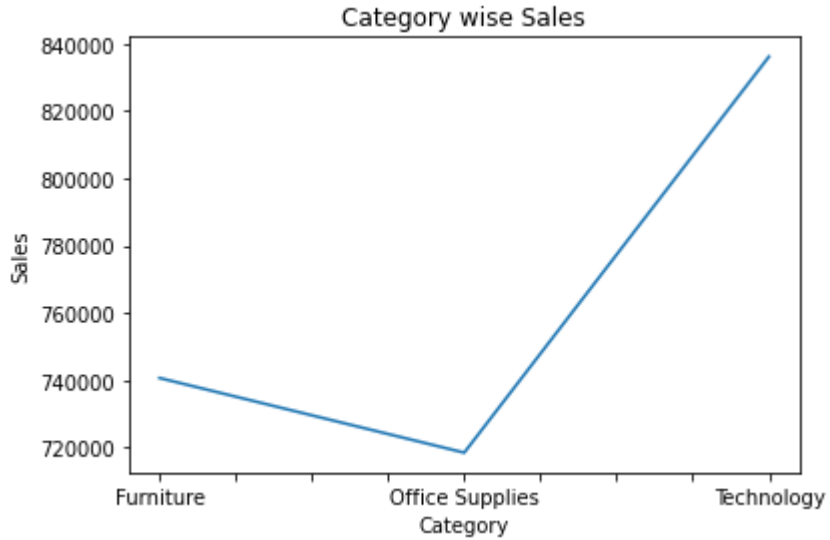
1)In Technology Category, highest sales was done in East Region

2)In Office Supplies Category, highest sales was done in West Region
3)In Furniture Category, highest sales was done in West Region

Line Chart - Category wise Sales

In [19]:
```
var=data.groupby("Category").Sales.sum()
fig=plt.figure()
ax1=fig.add_subplot(1,1,1)
ax1.set_xlabel("Category")
ax1.set_ylabel("Sales")
ax1.set_title("Category wise Sales")
var.plot(kind='line')
```

Out[19]:
```
<AxesSubplot:title={'center':'Category wise Sales'}, xlabel='Category', ylabel='Sales'>
```



Result:Sales was low on Office Supplies Category

DATA EXPLORATION

Weak areas where you can work to make more profit?

In [20]:
```
Meanprofit = data["Profit"].mean()
Meanprofit
```

Out[20]:
```
28.72347625502008
```

In [21]:
```
df = pd.DataFrame(data)
```

In [22]:
```
df1=df[['City', 'Profit']]
df1
```

Out[22]:

	City	Profit
0	Henderson	41.9136
1	Henderson	219.5820
2	Los Angeles	6.8714
3	Fort Lauderdale	-383.0310
4	Fort Lauderdale	2.5164
...	...	...
9989	Miami	4.1028
9990	Costa Mesa	15.6332
9991	Costa Mesa	19.3932
9992	Costa Mesa	13.3200
9993	Westminster	72.9480

9960 rows × 2 columns

In [23]:
```
df1=df1[(df1['Profit'] < Meanprofit)]
df1.sort_values(by='Profit').head(10)
```

Out[23]:

	City	Profit
7772	Lancaster	-6599.9780
683	Burlington	-3839.9904
9774	San Antonio	-3701.8928
3011	Louisville	-3399.9800
4991	Chicago	-2929.4845
3151	Newark	-2639.9912
5310	Houston	-2287.7820
9639	Concord	-1862.3124
1199	Houston	-1850.9464
2697	Jacksonville	-1811.0784

Result: These are the top 10 weak areas where we should work for profit

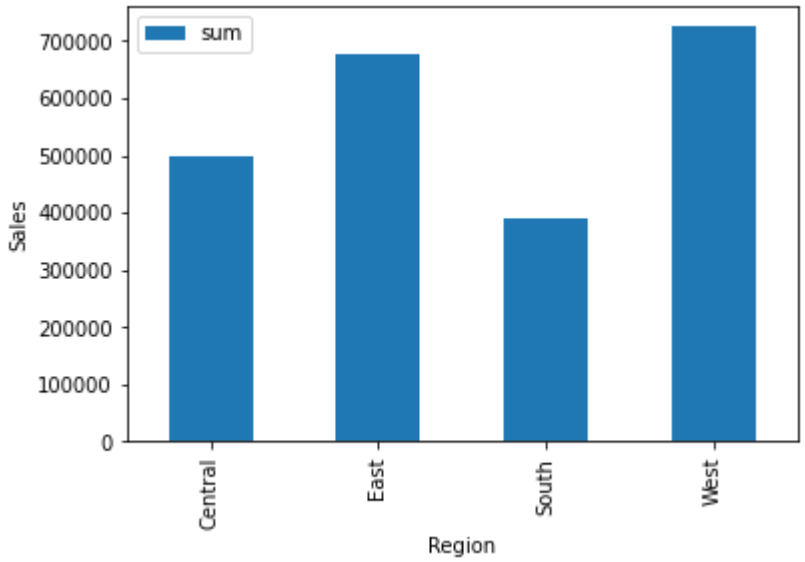
1)What is the region wise sales value?

In [24]:
```
df.groupby("Region").Sales.sum().sort_values(ascending=False)
```

Out[24]:
```
Region
West       725053.4485
East       678089.1520
Central    500325.8148
South      391721.9050
Name: Sales, dtype: float64
```

In [25]:
```
plt.figure(figsize=(10,16))
df.groupby('Region')['Sales'].agg(['sum']).plot.bar()
plt.ylabel('Sales')
plt.show()
```

```
<Figure size 720x1152 with 0 Axes>
```



Result:West Region has Highest Sales

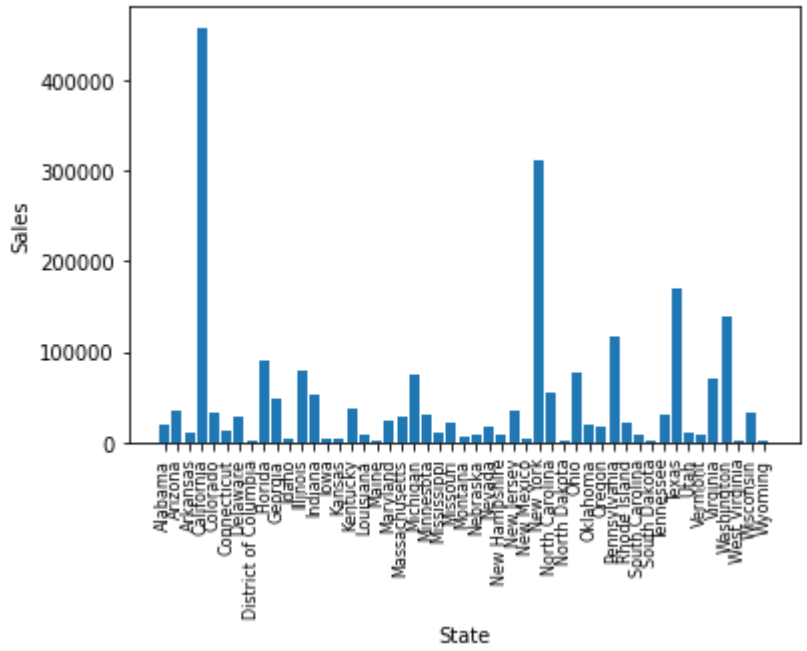
2)Which State is more profitable?

In [26]:
```
new_df=df.groupby(['State']).sum()
new_df.loc[:,'Sales'].sort_values(ascending=False)
```

Out[26]:
```
State
California              457464.9115
New York                310778.0310
Texas                   170061.0378
Washington              138480.3500
Pennsylvania            116480.8100
Florida                  89473.7080
Illinois                 80158.9730
Ohio                     77695.3920
Michigan                 75489.6740
Virginia                 70636.7200
North Carolina           55603.1640
Indiana                  53555.3600
Georgia                  49095.8400
Kentucky                 36591.7500
New Jersey               35764.3120
Arizona                  35282.0010
Wisconsin                32114.6100
Colorado                 32108.1180
Tennessee                30661.8730
Minnesota                29863.1500
Massachusetts            28634.4340
Delaware                 27451.0690
Maryland                 23705.5230
Rhode Island             22627.9560
Missouri                 22205.1500
Oklahoma                 19683.3900
Alabama                  19510.6400
Oregon                   17410.4140
Nevada                   16729.1020
Connecticut              13384.3570
Arkansas                 11678.1300
Utah                     11220.0560
Mississippi              10771.3400
Louisiana                 9217.0300
Vermont                   8929.3700
South Carolina            8481.7100
Nebraska                  7464.9300
New Hampshire             7292.5240
```

```
Montana                     5589.3520
New Mexico                  4783.5220
Iowa                        4579.7600
Idaho                       4382.4860
Kansas                      2914.3100
District of Columbia        2865.0200
Wyoming                     1603.1360
South Dakota                1315.5600
Maine                       1270.5300
West Virginia               1209.8240
North Dakota                 919.9100
Name: Sales, dtype: float64
```

In [27]:
```
keys = [City for City, df in df.groupby(['State'])]
plt.bar(keys,df.groupby(['State']).sum()['Sales'])

plt.ylabel('Sales')
plt.xlabel('State')
plt.xticks(keys, rotation='vertical', size=8)
plt.show()
```



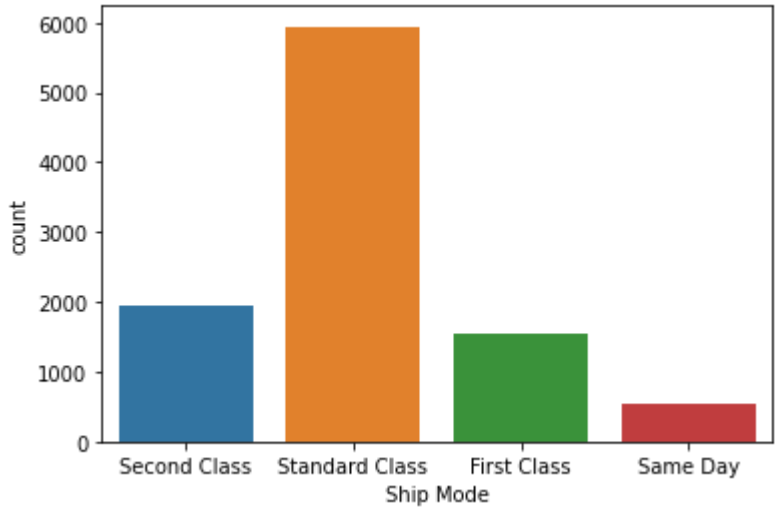
Result: California is more profitable

3)What is the most preferred Shipment Mode?

In [28]:
```
df['Ship Mode'].value_counts()
```

Out[28]:
```
Standard Class    5942
Second Class      1941
First Class       1536
Same Day           541
Name: Ship Mode, dtype: int64
```

In [29]:
```
sns.countplot(x='Ship Mode', data=df)
plt.show()
```



Result: Standard Class is the most preferred Shipment mode

4)Which is the most sold product?

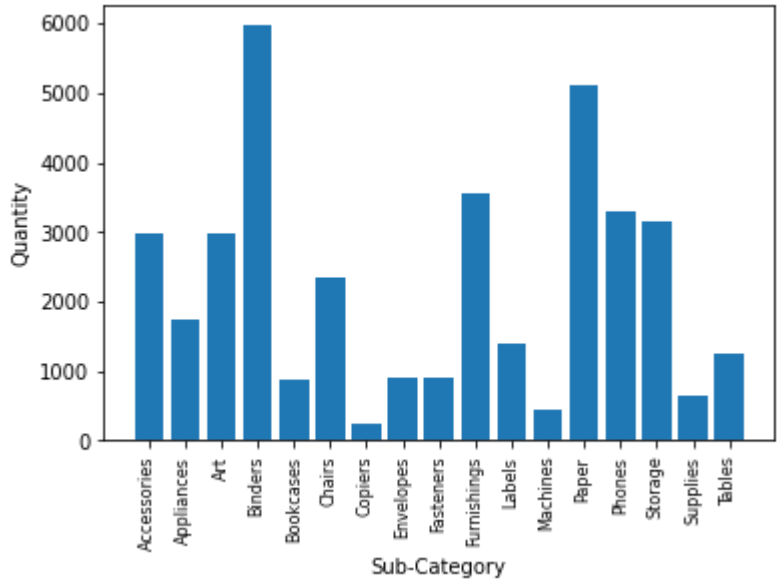
Grouping and Sorting products based on Quantity

In [30]:
```
most_selling_products = pd.DataFrame(df.groupby('Sub-Category').sum()['Quantity'])
most_selling_products.sort_values(by=['Quantity'], inplace=True, ascending=False)
most_selling_products[:10]
```

Out[30]:

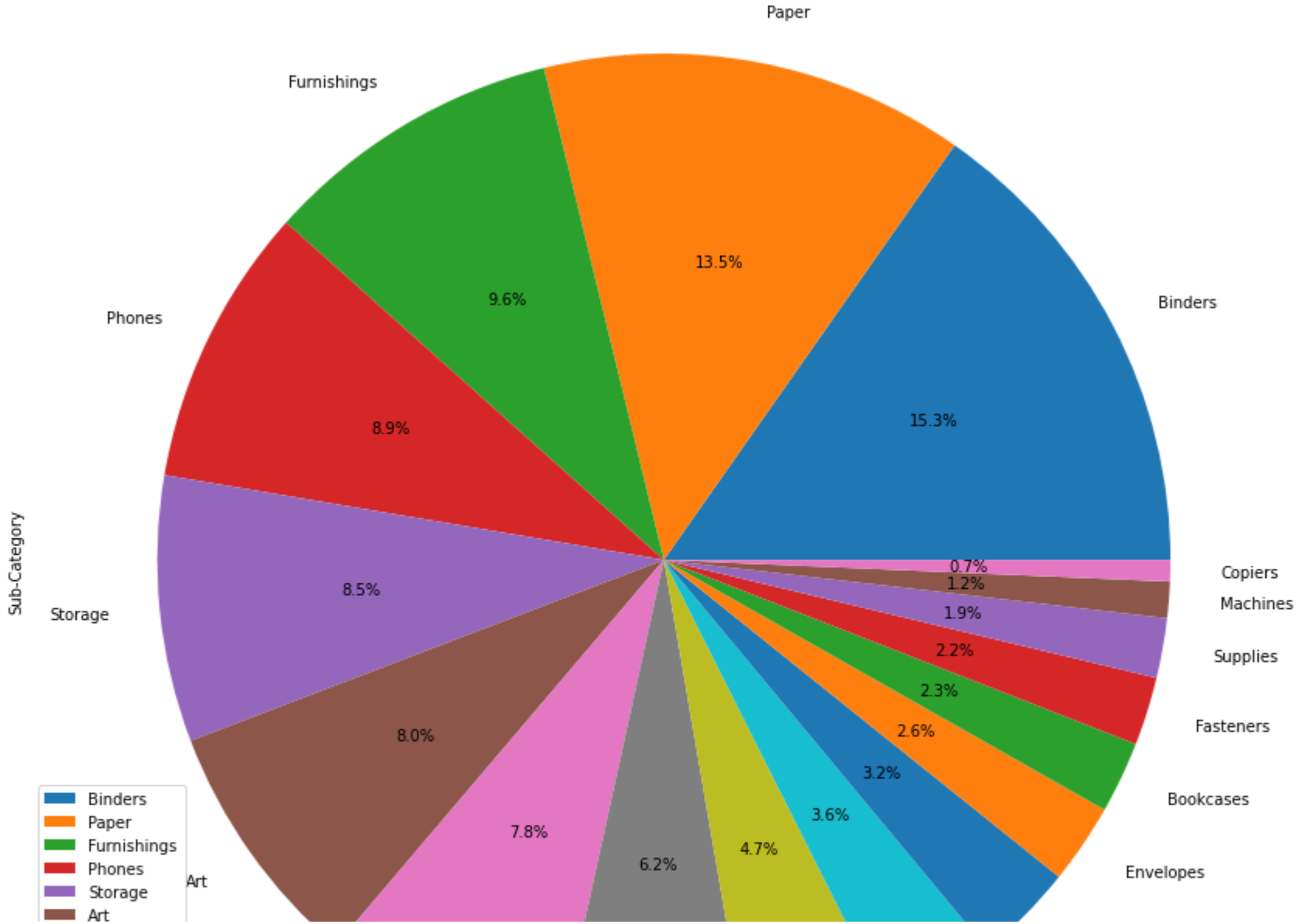
Sub-Category	Quantity
Binders	5968
Paper	5110
Furnishings	3557
Phones	3289
Storage	3158
Art	2992
Accessories	2976
Chairs	2346
Appliances	1729
Labels	1392

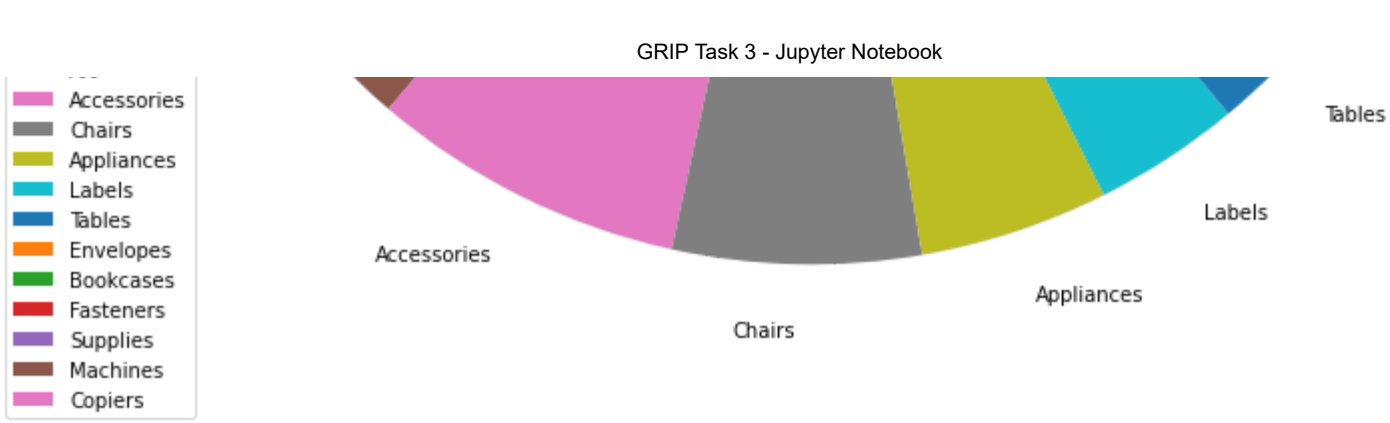
In [31]:
```
product_group = df.groupby('Sub-Category')
quantity = product_group.sum()['Quantity']

keys = [pair for pair, df in product_group]
plt.bar(keys, quantity)
plt.xticks(keys, rotation='vertical', size=8)
plt.ylabel('Quantity')
plt.xlabel('Sub-Category')
plt.show()
```



In [32]:
```
subcategory_chart=df['Sub-Category'].value_counts()
subcategory_chart.plot(kind = 'pie', autopct='%1.1f%%', figsize=(15, 15)).legend()
```

Out[32]: <matplotlib.legend.Legend at 0x19abe6234f0>





Result: Here you can see Binders have sold more with 5968 sales quantity