Exploratory Data Analysis - Retail (Task 3)

Perform 'Exploratory Data Analysis' on datasheet 'Sample superstore'

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

What all buisness problems you can derive by exploring the data?

I used Python to perform EDA on this datasheet.

Performing 'Exploratory Data Analysis' on datasheet ' SampleSuperstore'

In [7]:
```
# importing necessary modules
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
%matplotlib inline
```

In [8]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [9]:
```
sns.color_palette("tab10")
```

Out[9]:

In [10]:
```
# Loading data
data = pd.read_csv("C:\\Users\\abhi9\\Downloads\\Sample store.csv")
```

In [11]:
```
# first five rows
data.head()
```

Out[11]:

	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 [12]:
```
# last five rows
data.tail()
```

Out[12]:

	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 [13]:
```
data.shape
```

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

In [15]:
```
# statistical overview of the data
data.describe()
```

Out[15]:

	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 [16]:
```
#columns inside the datasheet
data.columns
```

Out[16]:
```
Index(['Ship Mode', 'Segment', 'Country', 'City', 'State', 'Postal Code',
       'Region', 'Category', 'Sub-Category', 'Sales', 'Quantity', 'Discount',
       'Profit'],
      dtype='object')
```

In [17]:
```
# overall info about data
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 [19]:
```
data['Country'].value_counts()
```

Out[19]:
```
United States    9994
Name: Country, dtype: int64
```

In [20]:
```
# Calculating Cost
data['Cost'] = data['Sales'] - data['Profit']

# Calculating Profit %
data['Profit%'] = (data['Profit']/data['Cost'])*100
```

In [21]:
```
data.head()
```

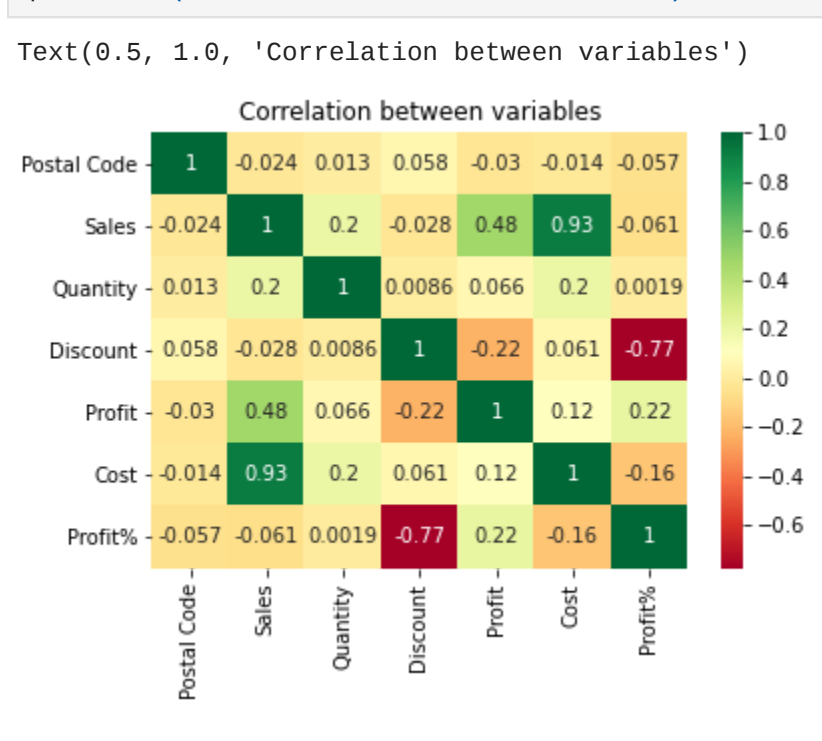
Out[21]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit	Cost	Profit%
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	0.00	41.9136	220.0464	19.047619
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9400	3	0.00	219.5820	512.3580	42.857143
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14.6200	2	0.00	6.8714	7.7486	88.679245
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5775	5	0.45	-383.0310	1340.6085	-28.571429
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.3680	2	0.20	2.5164	19.8516	12.676056

Multivariate Visualizations

In [22]:
```
# correlation marix and heatmap
datacorr = data.corr()
sns.heatmap(datacorr, annot=True , cmap='RdYlGn')
plt.title('Correlation between variables')
```

Out[22]:
```
Text(0.5, 1.0, 'Correlation between variables')
```



Finding modes of shipping products and observing which is preferred mode of shipping?

In [23]:
```
shipmodetypes = data.groupby('Ship Mode')
for i, df in shipmodetypes:
    print(i)
```

```
First Class
Same Day
Second Class
Standard Class
```

In [25]:
```
data.groupby('Ship Mode').groups
```

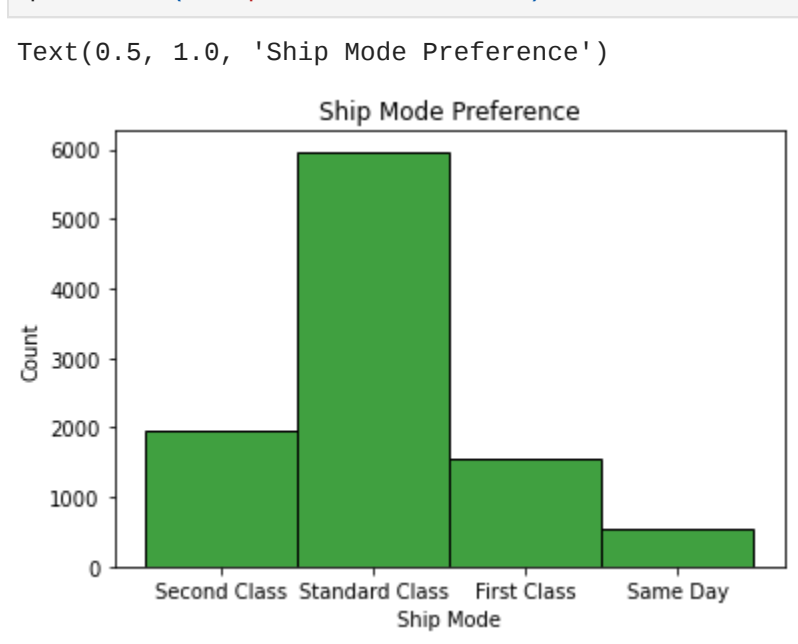
Out[25]:
```
{'First Class': [35, 36, 44, 45, 55, 56, 57, 58, 59, 60, 61, 69, 75, 76, 77, 79, 80, 84, 97, 119, 120, 121, 122, 123, 130, 131, 132, 151, 152, 153, 154, 155, 160, 189, 190, 191, 19
2, 193, 201, 202, 219, 220, 221, 222, 223, 224, 252, 253, 271, 272, 273, 274, 275, 293, 294, 295, 296, 297, 308, 316, 326, 327, 328, 329, 330, 349, 350, 351, 352, 353, 359, 360, 37
6, 381, 382, 388, 402, 421, 427, 433, 484, 485, 486, 487, 510, 511, 512, 522, 523, 524, 540, 541, 546, 547, 552, 563, 564, 565, 598, 610, ...], 'Same Day': [366, 367, 368, 369, 65
7, 658, 664, 683, 684, 746, 747, 792, 813, 814, 882, 959, 987, 1001, 1002, 1003, 1086, 1135, 1146, 1147, 1148, 1149, 1150, 1166, 1193, 1194, 1195, 1196, 1234, 1235, 1236, 1237, 127
3, 1274, 1275, 1276, 1355, 1356, 1382, 1383, 1384, 1385, 1386, 1387, 1388, 1389, 1390, 1391, 1392, 1437, 1438, 1459, 1467, 1473, 1508, 1562, 1563, 1564, 1568, 1593, 1630, 1631, 163
2, 1633, 1634, 1650, 1651, 1704, 1728, 1729, 1815, 1816, 1830, 1831, 1847, 1848, 1862, 1865, 1880, 1881, 1882, 1979, 1980, 2011, 2012, 2102, 2106, 2107, 2108, 2109, 2110, 2111, 211
2, 2113, 2114, 2115, ...], 'Second Class': [0, 1, 2, 17, 18, 19, 20, 23, 25, 26, 34, 46, 71, 78, 85, 88, 92, 93, 94, 96, 102, 113, 114, 115, 116, 124, 128, 129, 140, 157, 161, 176,
177, 178, 180, 181, 182, 183, 184, 203, 211, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 258, 259, 260, 262, 263, 270, 280, 281, 286, 287, 288, 289, 290, 291,
292, 304, 309, 310, 311, 312, 325, 331, 332, 333, 334, 335, 336, 339, 340, 341, 342, 343, 383, 391, 392, 393, 395, 396, 398, 399, 400, 401, 424, 425, 426, 436, ...], 'Standard Clas
s': [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 21, 22, 24, 27, 28, 29, 30, 31, 32, 33, 37, 38, 39, 40, 41, 42, 43, 47, 48, 49, 50, 51, 52, 53, 54, 62, 63, 64, 65, 66, 67, 6
8, 70, 72, 73, 74, 81, 82, 83, 86, 87, 89, 90, 91, 95, 98, 99, 100, 101, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 117, 118, 125, 126, 127, 133, 134, 135, 136, 137, 138, 13
9, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 156, 158, 159, 162, 163, ...]}
```

In [26]:
```
data['Ship Mode'].value_counts()
```

Out[26]:
```
Standard Class    5968
Second Class      1945
First Class       1538
Same Day           543
Name: Ship Mode, dtype: int64
```

In [28]:
```
sns.histplot(x = data['Ship Mode'], color = 'g')
plt.title('Ship Mode Preference')
```

Out[28]:
```
Text(0.5, 1.0, 'Ship Mode Preference')
```



Customer Segments

In [30]:
```
segmenttypes = data.groupby('Segment')
for i,df in segmenttypes:
      print(i)
```

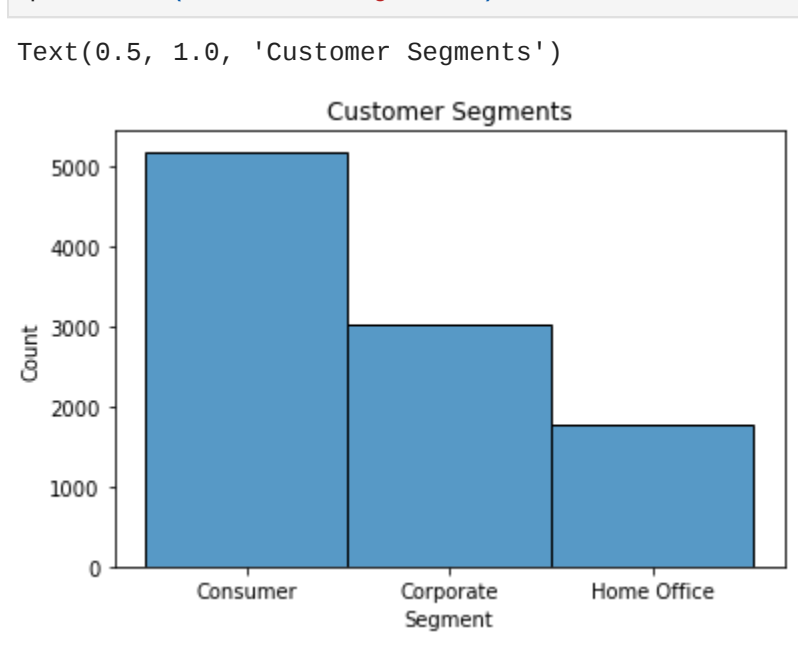
```
Consumer
Corporate
Home Office
```

In [31]:
```
data['Segment'].value_counts()
```

Out[31]:
```
Consumer       5191
Corporate      3020
Home Office    1783
Name: Segment, dtype: int64
```

In [32]:
```
sns.histplot(x = data['Segment'])
plt.title('Customer Segments')
```

Out[32]:
```
Text(0.5, 1.0, 'Customer Segments')
```



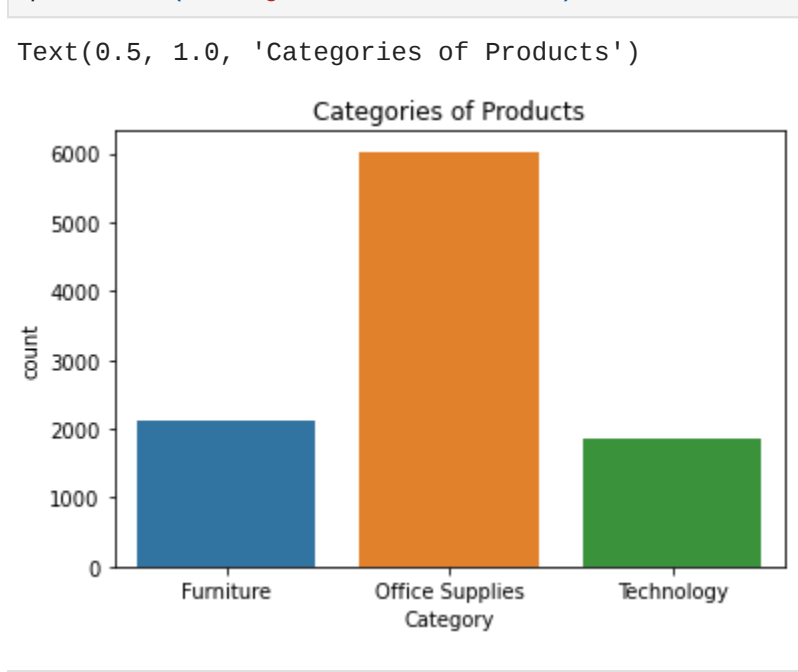
Category wise Analysis

In [33]:
```
cat = data.groupby('Category')
for i,df in cat:
    print(i)
```

```
Furniture
Office Supplies
Technology
```

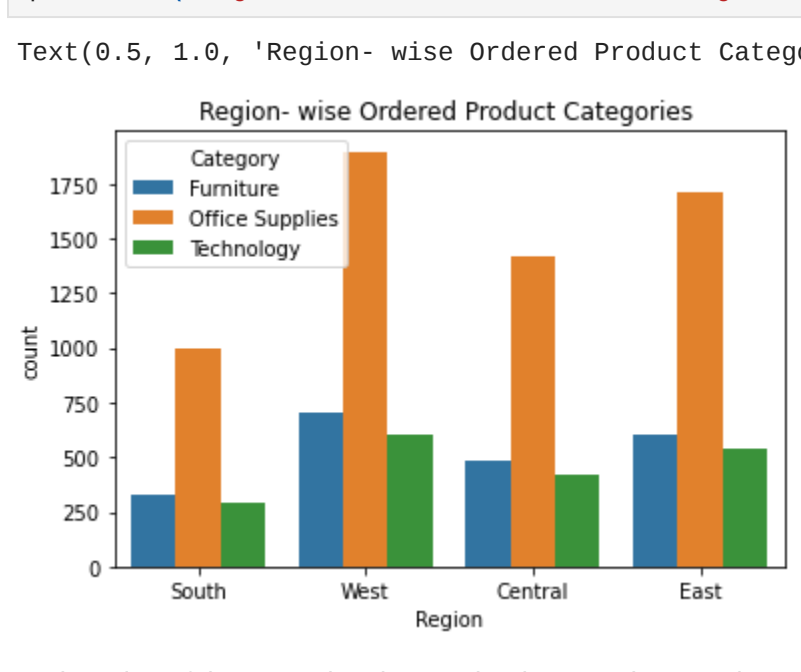
In [34]:
```
sns.countplot(x=data['Category'])
plt.title('Categories of Products')
```

Out[34]:
```
Text(0.5, 1.0, 'Categories of Products')
```



In [35]:
```
sns.countplot(x = data['Region'], hue = data['Category'])
plt.title('Region- wise Ordered Product Categories ')
```

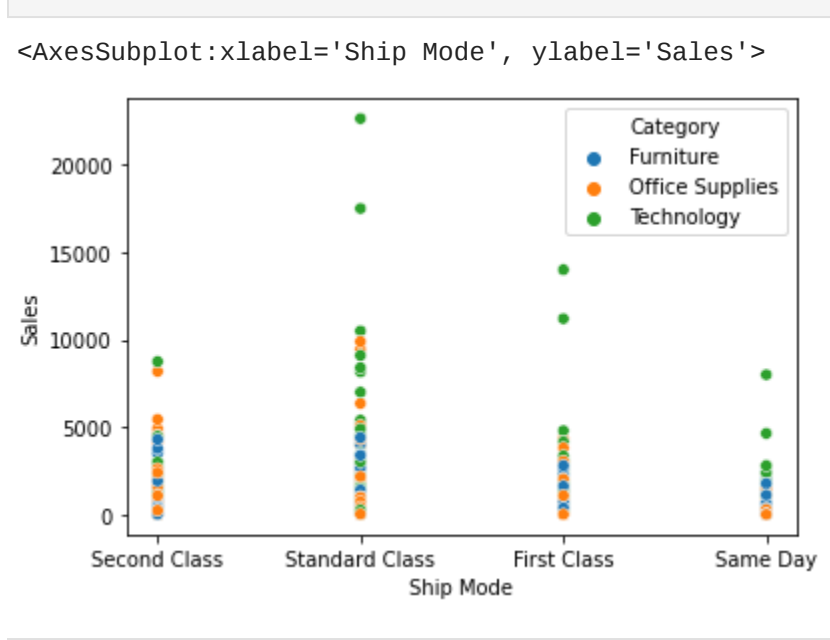
Out[35]:
```
Text(0.5, 1.0, 'Region- wise Ordered Product Categories ')
```



south region of the US orders less technology products and more office supplies. west orders more than any other region

In [36]:
```
sns.scatterplot(x = data['Ship Mode'], y = data['Sales'], hue = data['Category'])
```

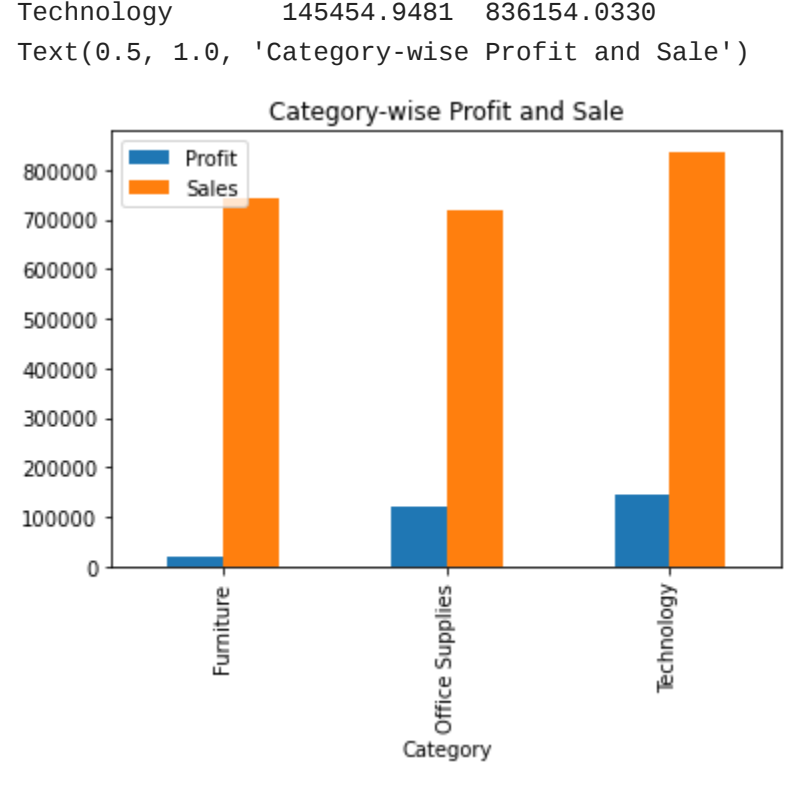
Out[36]:
```
<AxesSubplot:xlabel='Ship Mode', ylabel='Sales'>
```



In [38]:
```
ds = data.groupby('Category')['Profit','Sales'].agg('sum')
print(ds)
ds.plot.bar()
plt.legend(loc = 'upper left')
plt.title('Category-wise Profit and Sale')
```

```
                      Profit        Sales
Category
Furniture         18451.2728  741999.7953
Office Supplies  122490.8008  719047.0320
Technology       145454.9481  836154.0330
```

Out[38]:
```
Text(0.5, 1.0, 'Category-wise Profit and Sale')
```



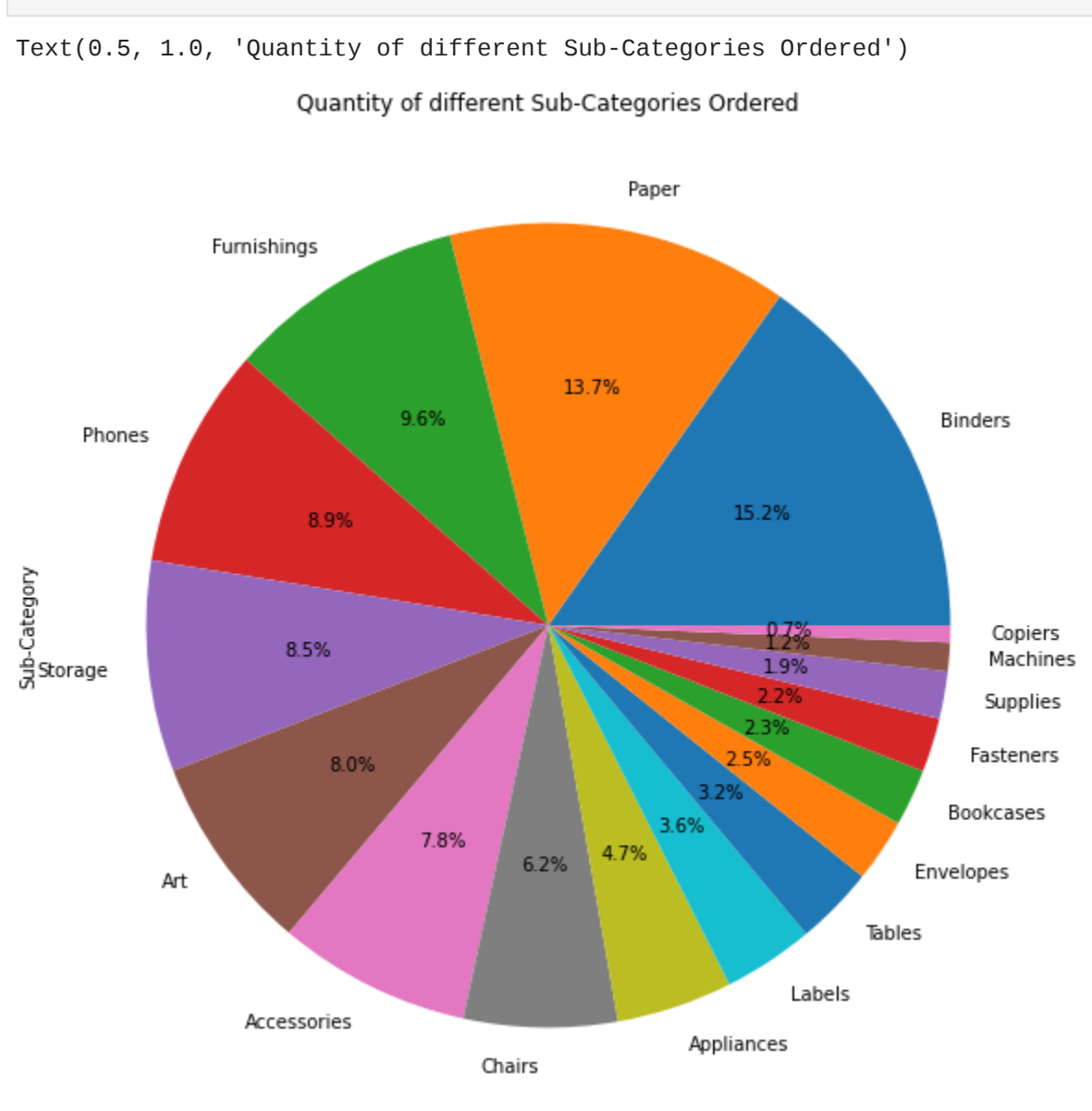
Sub-Categorical-wise Analysis

In [39]:
```
subcatarr = []
subcat = data.groupby('Sub-Category')
for i,df in subcat:
    print(i)
    subcatarr.append(i)
```

```
Accessories
Appliances
Art
Binders
Bookcases
Chairs
Copiers
Envelopes
Fasteners
Furnishings
Labels
Machines
Paper
Phones
Storage
Supplies
Tables
```

In [43]:
```
plt.figure(figsize =(10,10))
data['Sub-Category'].value_counts().plot.pie(autopct="%1.1f%%")
plt.title('Quantity of different Sub-Categories Ordered')
```

Out[43]:
```
Text(0.5, 1.0, 'Quantity of different Sub-Categories Ordered')
```



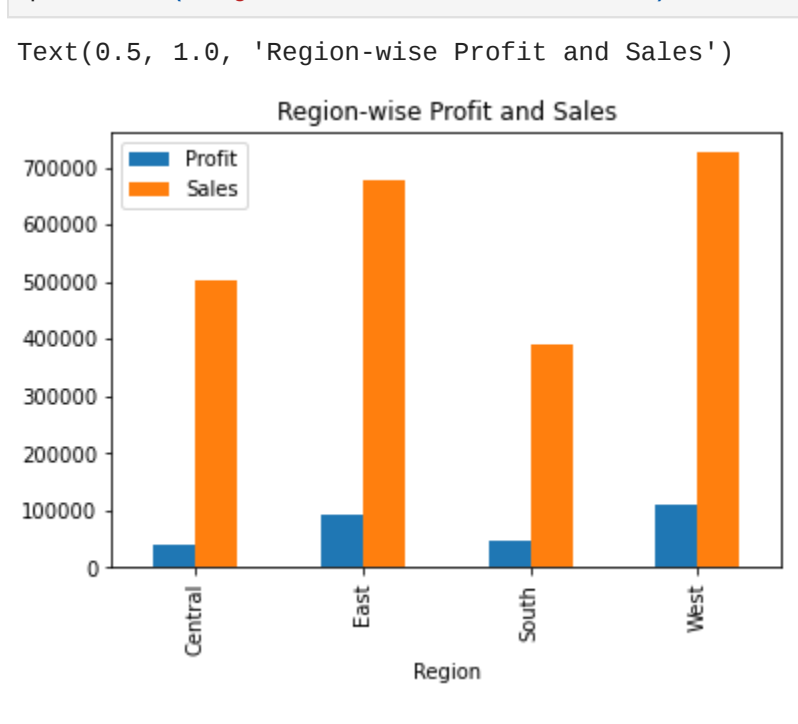
Region wise Analysis

In [44]:
```
regions = data.groupby('Region')
for i,df in regions:
    print(i)
```

```
Central
East
South
West
```

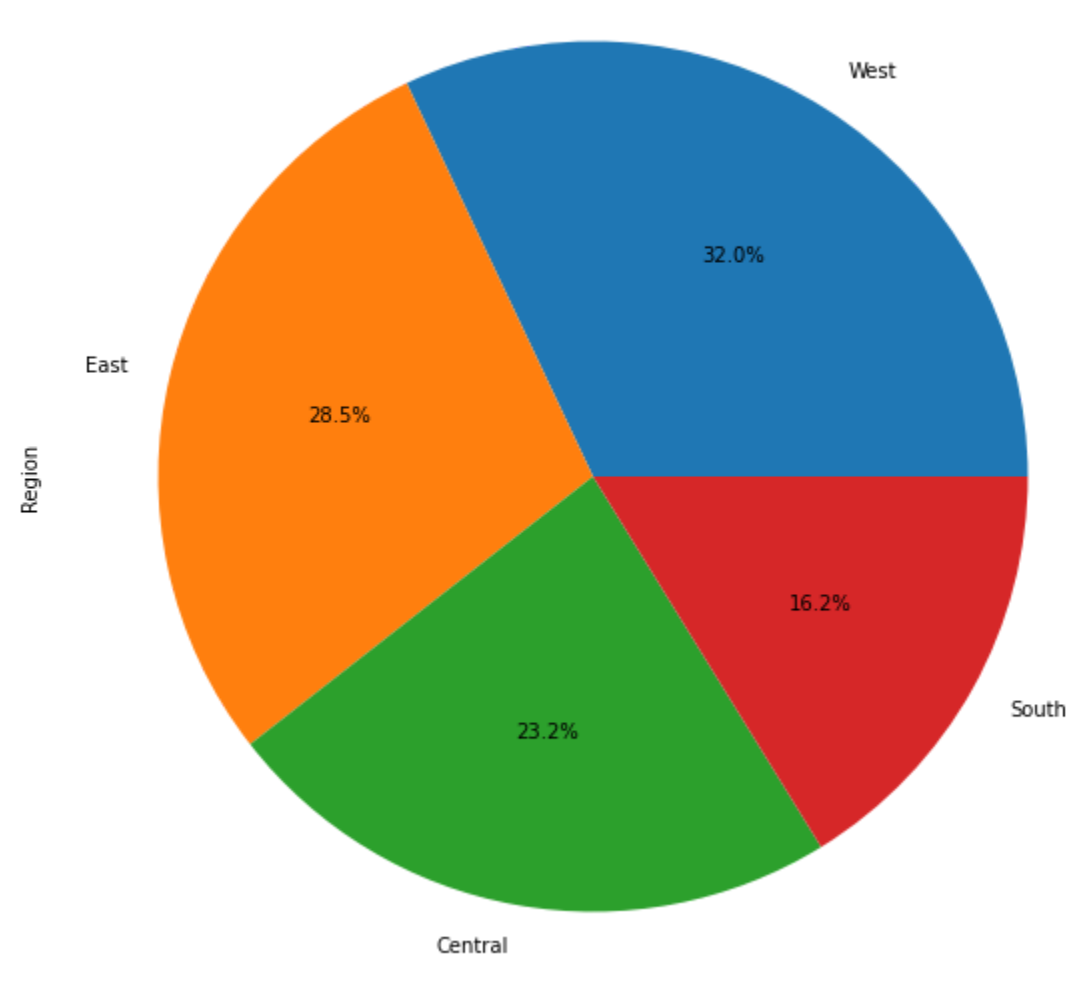
In [46]:
```
rw = data.groupby('Region')['Profit','Sales'].agg('sum')
rw.plot.bar()
plt.legend(loc ='upper left')
plt.title('Region-wise Profit and Sales')
```

Out[46]:
```
Text(0.5, 1.0, 'Region-wise Profit and Sales')
```



In [47]:
```
plt.figure(figsize = (10,10))
data['Region'].value_counts().plot.pie(autopct="%1.1f%%")
```

Out[47]:
```
<AxesSubplot:ylabel='Region'>
```



City wise Analysis

In [71]:
```
city =[]
cities = data.groupby('City')
for i ,df in cities:
    city.append(i)
```

In [72]:
```
len(city)
```

Out[72]:
```
531
```

In [73]:
```
data['City'].value_counts()
```

Out[73]:
```
New York City    915
Los Angeles      747
Philadelphia     537
San Francisco    510
Seattle          428
                ...
Redwood City       1
Lindenhurst        1
Rock Hill          1
Whittier           1
Goldsboro          1
Name: City, Length: 531, dtype: int64
```

In [74]:
```
data['City'].value_counts().min()
```

Out[74]:
```
1
```

In [75]:
```
data['City'].value_counts().max()
```

Out[75]:
```
915
```

In [76]:
```
data[data['City']== 'New York City']
```

Out[76]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit	Cost	Profit%
53	Standard Class	Corporate	United States	New York City	New York	10024	East	Office Supplies	Fasteners	15.260	7	0.0	6.2566	9.0034	69.491525
54	Standard Class	Corporate	United States	New York City	New York	10024	East	Technology	Phones	1029.950	5	0.0	298.6855	731.2645	40.845070
70	Standard Class	Consumer	United States	New York City	New York	10009	East	Office Supplies	Binders	4.616	1	0.2	1.7310	2.8850	60.000000
96	Second Class	Home Office	United States	New York City	New York	10009	East	Furniture	Furnishings	96.530	7	0.0	40.5426	55.9874	72.413793
110	Standard Class	Corporate	United States	New York City	New York	10035	East	Furniture	Furnishings	41.960	2	0.0	10.9096	31.0504	35.135135
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
9926	First Class	Corporate	United States	New York City	New York	10035	East	Technology	Phones	199.980	2	0.0	53.9946	145.9854	36.986301
9927	First Class	Corporate	United States	New York City	New York	10035	East	Office Supplies	Storage	83.920	4	0.0	20.1408	63.7792	31.578947
9938	Standard Class	Corporate	United States	New York City	New York	10009	East	Furniture	Furnishings	60.350	5	0.0	19.9155	40.4345	49.253731
9939	Standard Class	Corporate	United States	New York City	New York	10009	East	Office Supplies	Supplies	35.520	4	0.0	9.9456	25.5744	38.888889
9940	Standard Class	Corporate	United States	New York City	New York	10009	East	Office Supplies	Art	11.200	7	0.0	4.8160	6.3840	75.438596

915 rows × 15 columns

Thank You Very Much