

Exploratory Data Analysis - Retail

- Perform 'Exploratory Data Analysis' on dataset 'SampleSuperstore'
- As a business manager, try to find out the weak areas where you can work to make more profit.
- What all business problems you can derive by exploring the data?
- I used Python to perform EDA on this dataset.
- Dataset: <https://bit.ly/3i4rbWI>

By: Varun Gupta

In [1]:

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

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

In [2]:

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

Basic Data Insights

In [3]:

```
df.sample(5)
```

Out[3]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category
332	Second Class	Consumer	United States	Philadelphia	Pennsylvania	19134	East	Office Supplies	Binders
4486	Standard Class	Consumer	United States	Los Angeles	California	90049	West	Office Supplies	Paper
9481	Standard Class	Consumer	United States	Dallas	Texas	75220	Central	Office Supplies	Storage
8833	Standard Class	Home Office	United States	Sandy Springs	Georgia	30328	South	Office Supplies	Storage
7243	Second Class	Consumer	United States	Philadelphia	Pennsylvania	19134	East	Furniture	Chair



In [4]:

```
df.head()
```

Out[4]:

Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category
-----------	---------	---------	------	-------	-------------	--------	----------	--------------

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	26
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	73
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	1
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	95
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	2

In [5]: `df.tail()`

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category
9989	Second Class	Consumer	United States	Miami	Florida	33180	South	Furniture	Furniture
9990	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Furniture	Furniture
9991	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Technology	Phone
9992	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Office Supplies	Pat
9993	Second Class	Consumer	United States	Westminster	California	92683	West	Office Supplies	Applian

In [6]: `df.shape`

Out[6]: (9994, 13)

In [7]: `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 [8]: `df.describe()`

	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

Number of unique values in each column

In [9]: `for i in df.columns:
 print(i,len(df[i].unique()))`

```
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
```

Check for null values

In [11]: `df.isnull().sum()`

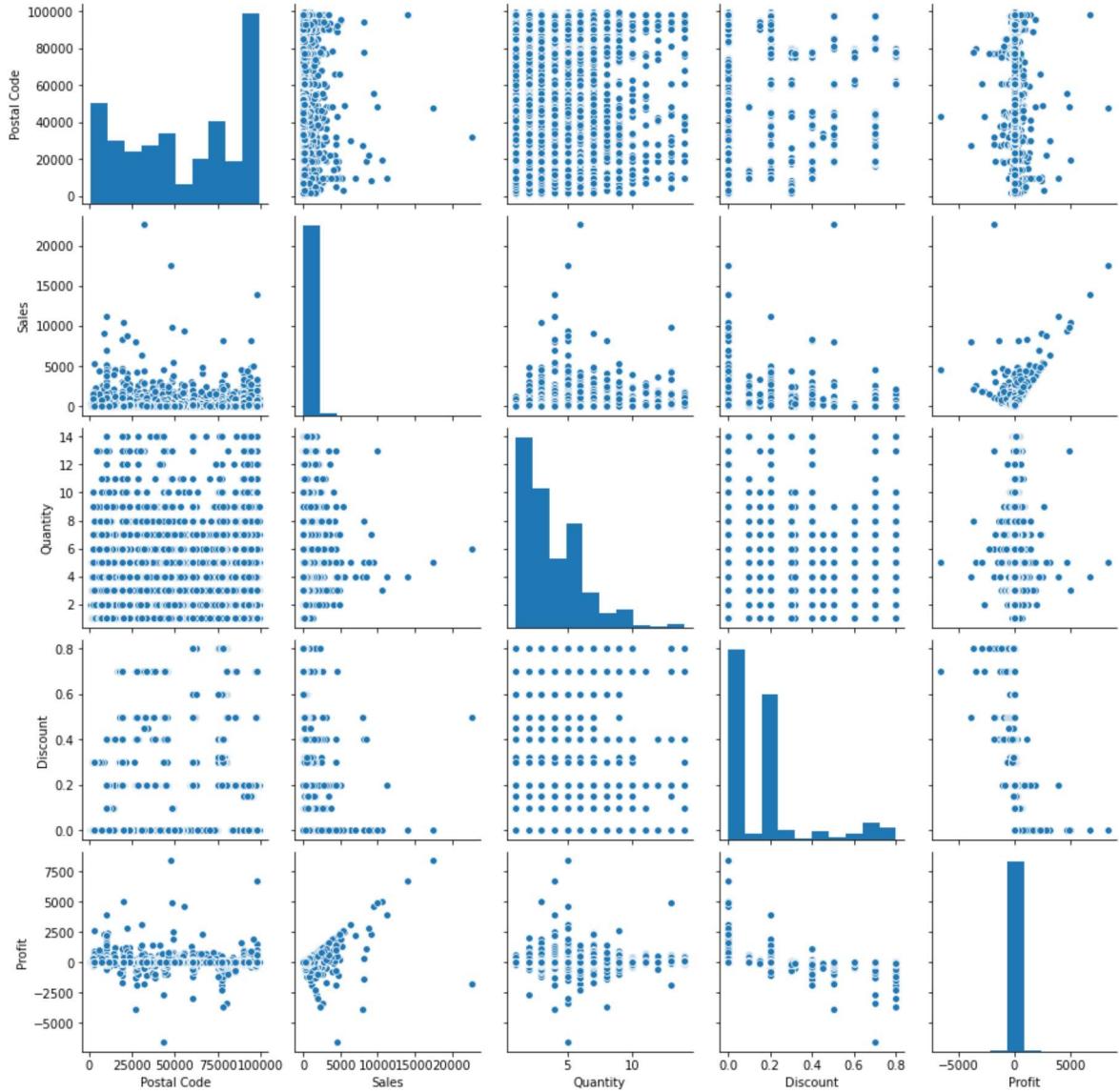
```
Out[11]: Ship Mode      0
          Segment       0
          Country       0
          City          0
          State         0
          Postal Code   0
          Region        0
          Category      0
          Sub-Category   0
          Sales          0
```

```
Quantity      0  
Discount      0  
Profit        0  
dtype: int64
```

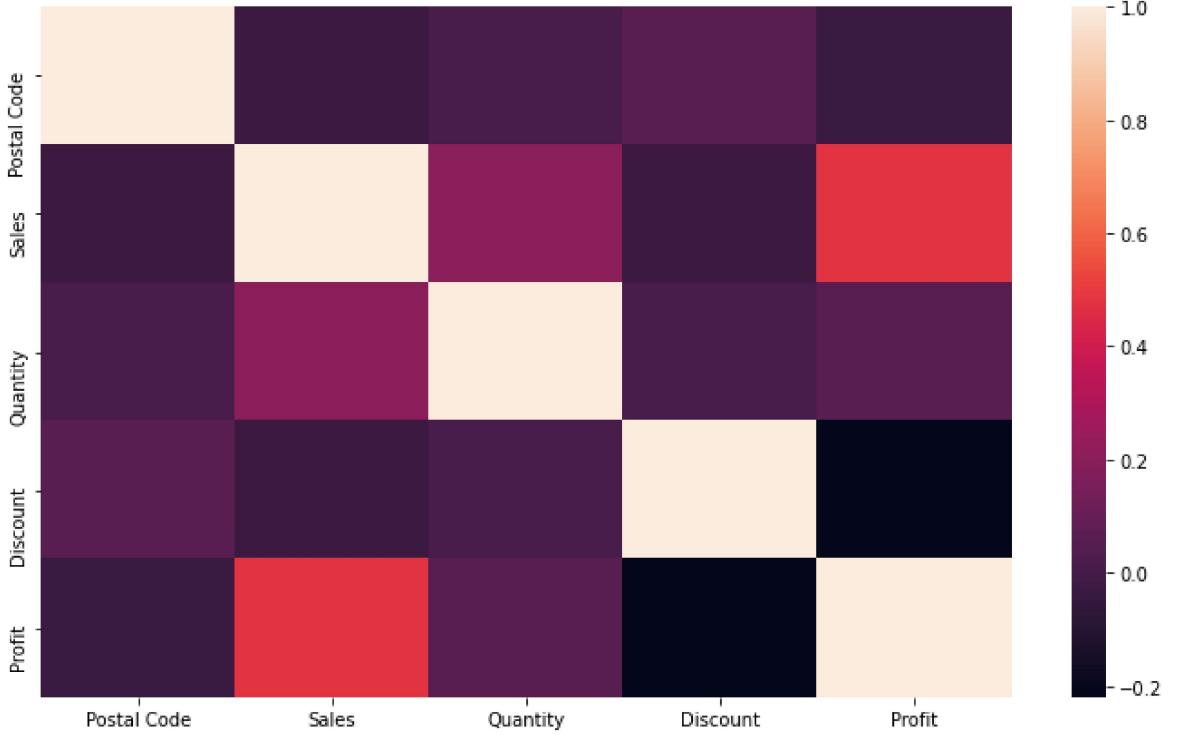
Data Visualization

```
In [13]: sns.pairplot(df)
```

```
Out[13]: <seaborn.axisgrid.PairGrid at 0x9fbeaae9e8>
```



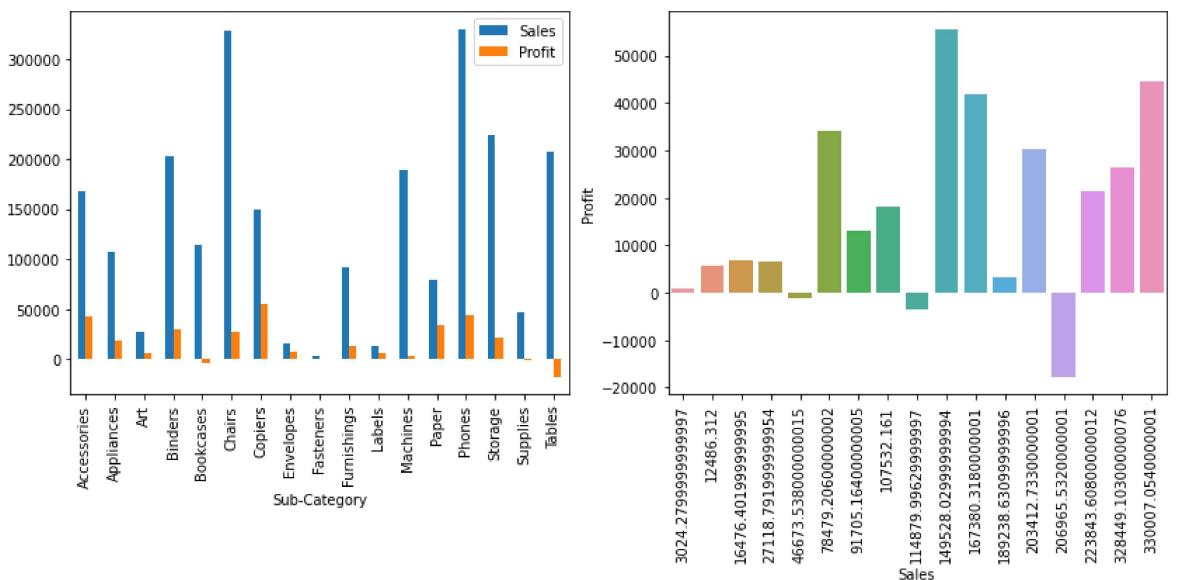
```
In [21]: fig,axes = plt.subplots(1,1,figsize=(12,7))  
sns.heatmap(df.corr())  
plt.show()
```



In [15]:

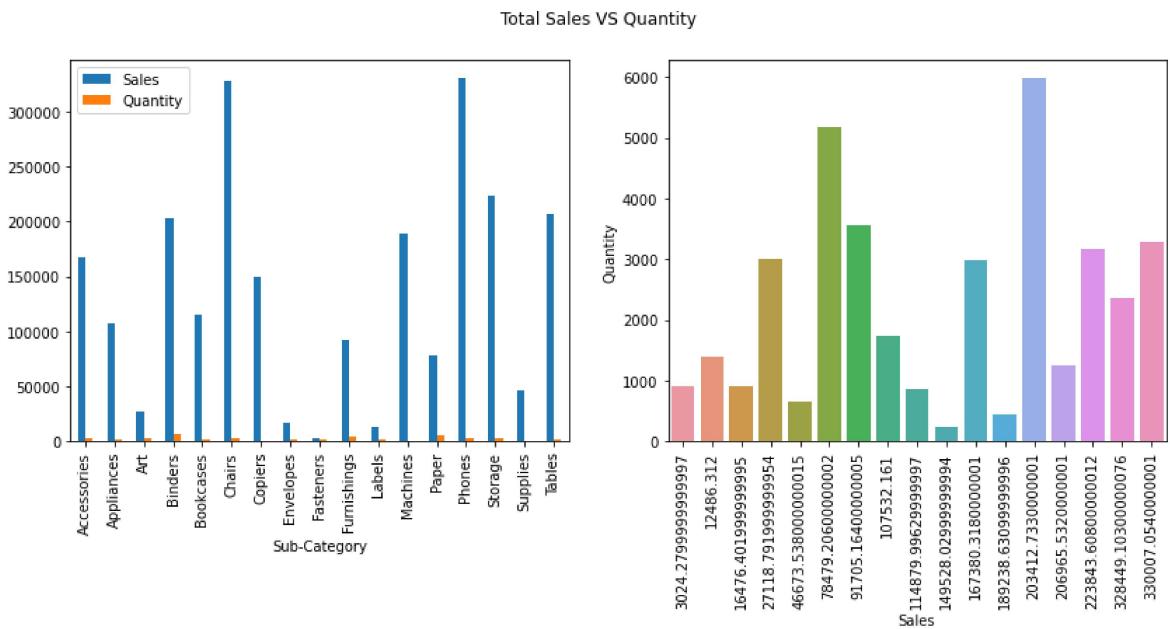
```
fig,axes = plt.subplots(1,2,figsize=(14,5))
fig.suptitle("Total profit VS sales ")
sns.barplot(data=df.groupby('Sub-Category')[['Sales','Profit']].agg(sum),x='Sales',
df.groupby('Sub-Category')[['Sales','Profit']].agg(sum).plot(kind='bar',ax=axes[0])
plt.xticks(rotation=90)
plt.show()
```

Total profit VS sales

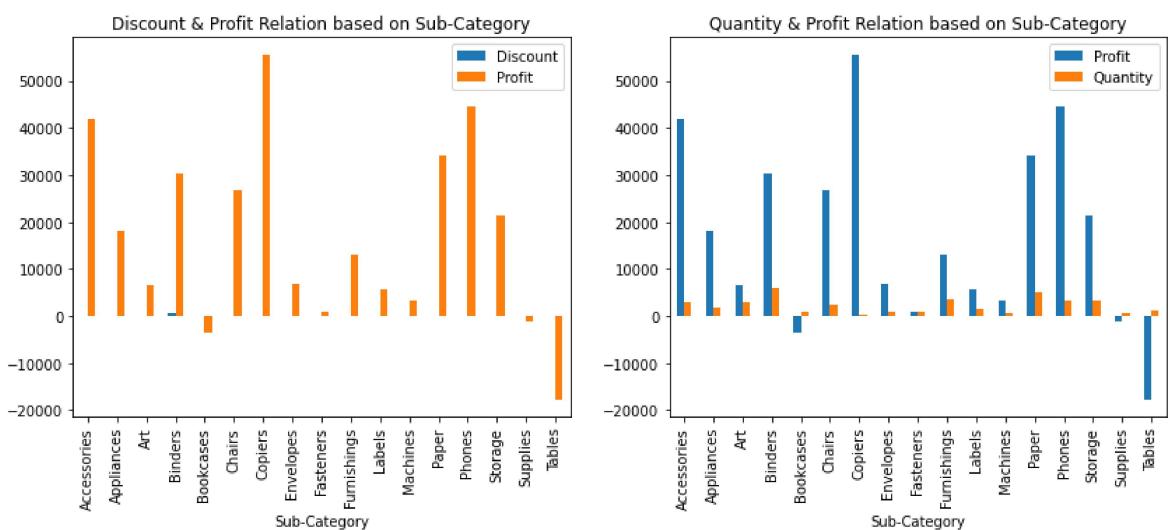


In [16]:

```
fig,axes = plt.subplots(1,2,figsize=(14,5))
fig.suptitle("Total Sales VS Quantity ")
sns.barplot(data=df.groupby('Sub-Category')[['Sales','Quantity']].agg(sum),x='Sales',
df.groupby('Sub-Category')[['Sales','Quantity']].agg(sum).plot(kind='bar',ax=axes[0])
plt.xticks(rotation=90)
plt.show()
```

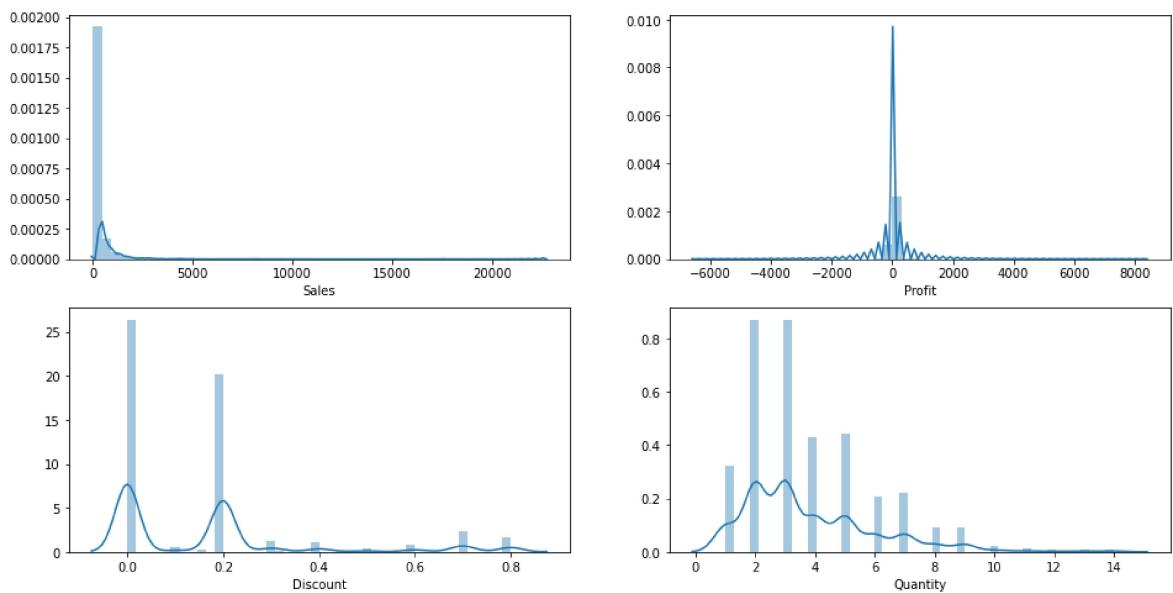


```
In [26]: fig,axes = plt.subplots(1,2,figsize=(14,5))
df.groupby('Sub-Category')['Discount','Profit'].agg(sum).plot(kind='bar',ax=axes[0])
df.groupby('Sub-Category')['Profit','Quantity'].agg(sum).plot(kind='bar',ax=axes[1])
plt.xticks(rotation=90)
plt.show()
```



```
In [18]: fig,axes = plt.subplots(2,2,figsize=(16,8))
fig.suptitle("Distribution plots", fontsize=16)
sns.distplot(df['Sales'],ax=axes[0,0])
sns.distplot(df['Profit'],ax=axes[0,1])
sns.distplot(df['Discount'],ax=axes[1,0])
sns.distplot(df['Quantity'],ax=axes[1,1])
plt.show()
```

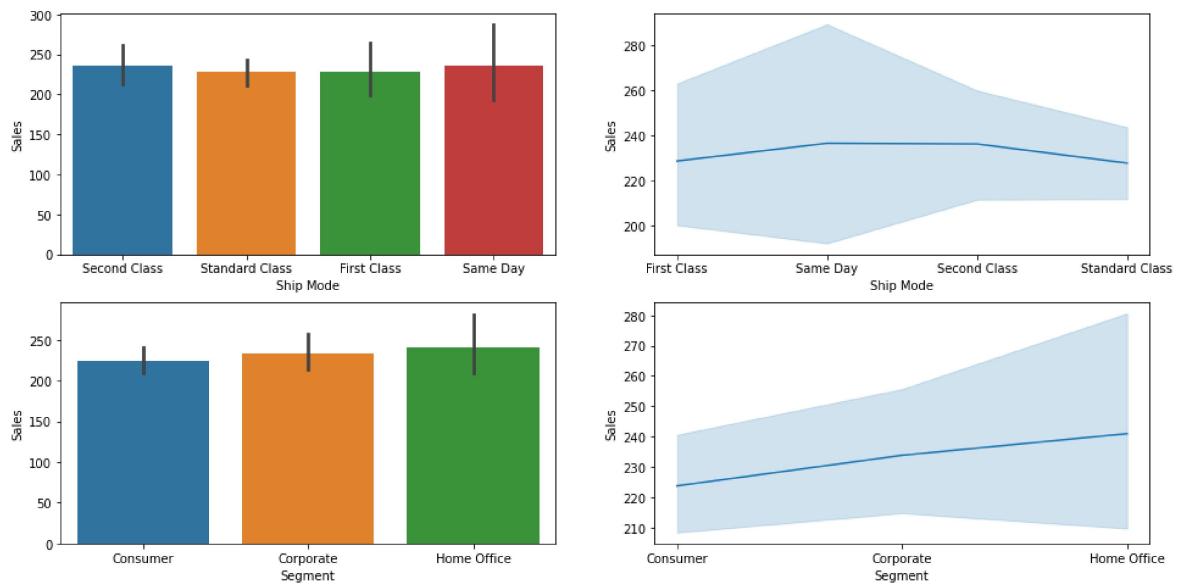
Distribution plots



In [19]:

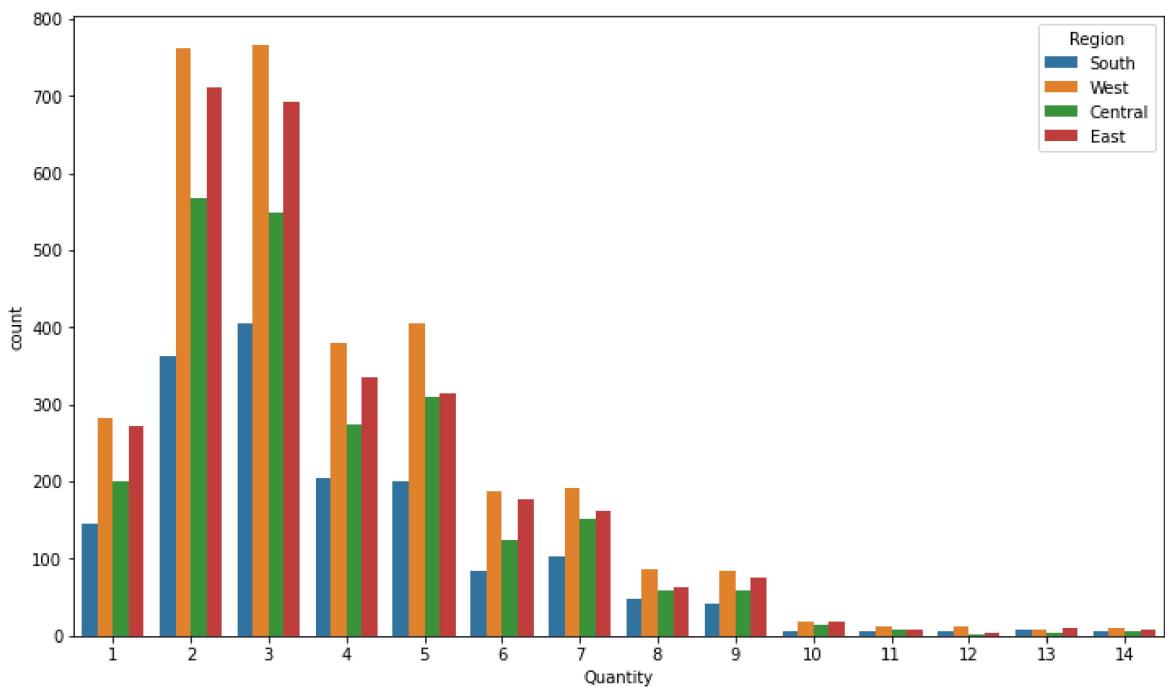
```
fig,axes = plt.subplots(2,2,figsize=(16,8))
fig.suptitle("Sales with different shipping modes and Segments", fontsize=16)
sns.barplot(df['Ship Mode'],df['Sales'],ax=axes[0,0])
sns.lineplot(df['Ship Mode'],df['Sales'],ax=axes[0,1])
sns.barplot(df['Segment'],df['Sales'],ax=axes[1,0])
sns.lineplot(df['Segment'],df['Sales'],ax=axes[1,1])
plt.show()
```

Sales with different shipping modes and Segments



In [20]:

```
fig,ax= plt.subplots(1,1,figsize=(12,7))
sns.countplot(df['Quantity'],hue=df['Region'])
plt.show()
```



Some important Findings

- #### The features Profit and Discounts are highly related.
- #### Over Less quantity of products also the sales were high.
- #### The maximum quantity of product in demand was in range 2-4.
- #### The mode of shipping doesn't affect much to the sales
- #### The Home Office provides highest sales followed by Corporate by a slight variation

Thank You :)

In []: