# Name:Pranil Rego

# Task 3- Exploratory Data Analysis - Retail

### To Perform 'Exploratory Data Analysis' on dataset "SampleSuperstore" To detect the weak areas where more work is necessary to make profit.

# Importing the Libraries

In [29]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

# Reading the Data

In [13]:
```
df = pd.read_csv(r'C:\Users\Pranil Rego\Downloads\SampleSuperstore.csv')
df.head()
```

Out[13]:

| | Ship Mode | Segment | Country | City | State | Postal Code | Region | Category | Sub-Category | Sales | Q |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Second Class | Consumer | United States | Henderson | Kentucky | 42420 | South | Furniture | Bookcases | 261.9600 | |
| 1 | Second Class | Consumer | United States | Henderson | Kentucky | 42420 | South | Furniture | Chairs | 731.9400 | |
| 2 | Second Class | Corporate | United States | Los Angeles | California | 90036 | West | Office Supplies | Labels | 14.6200 | |
| 3 | Standard Class | Consumer | United States | Fort Lauderdale | Florida | 33311 | South | Furniture | Tables | 957.5775 | |
| 4 | Standard Class | Consumer | United States | Fort Lauderdale | Florida | 33311 | South | Office Supplies | Storage | 22.3680 | |

# Preprocessing Part-1

In [14]:
```
#Check the shape
df.shape
```

Out[14]:
```
(9994, 13)
```

In [15]:
```
#check for the missing value
df.isnull().sum()
```

Out[15]:
```
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 [20]:
```
#count of each category under shipment mode
df["Ship Mode"].value_counts()
```

Out[20]:
```
Standard Class    5968
Second Class      1945
First Class       1538
Same Day           543
Name: Ship Mode, dtype: int64
```

In [21]:
```
df["Country"].nunique()
```

Out[21]:
```
1
```

In [22]:
```
## dropping columns which dont affect profits much.
df = df.drop(["Country","Postal Code"],axis=1)
```

In [23]:
```
#checking for duplicate values
df.duplicated().sum()
```

Out[23]:
```
50
```

In [24]:
```
#dropping the duplicate values
df.drop_duplicates(inplace=True)
df
```

Out[24]:

| | Ship Mode | Segment | City | State | Region | Category | Sub-Category | Sales | Quantity | Dis |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | Second Class | Consumer | Henderson | Kentucky | South | Furniture | Bookcases | 261.9600 | 2 | |
| **1** | Second Class | Consumer | Henderson | Kentucky | South | Furniture | Chairs | 731.9400 | 3 | |
| **2** | Second Class | Corporate | Los Angeles | California | West | Office Supplies | Labels | 14.6200 | 2 | |
| **3** | Standard Class | Consumer | Fort Lauderdale | Florida | South | Furniture | Tables | 957.5775 | 5 | |
| **4** | Standard Class | Consumer | Fort Lauderdale | Florida | South | Office Supplies | Storage | 22.3680 | 2 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **9989** | Second Class | Consumer | Miami | Florida | South | Furniture | Furnishings | 25.2480 | 3 | |
| **9990** | Standard Class | Consumer | Costa Mesa | California | West | Furniture | Furnishings | 91.9600 | 2 | |
| **9991** | Standard Class | Consumer | Costa Mesa | California | West | Technology | Phones | 258.5760 | 2 | |
| **9992** | Standard Class | Consumer | Costa Mesa | California | West | Office Supplies | Paper | 29.6000 | 4 | |
| **9993** | Second Class | Consumer | Westminster | California | West | Office Supplies | Appliances | 243.1600 | 2 | |

9944 rows × 12 columns

In [28]:
```
import seaborn as sns
```

In [30]:
```
sns.distplot(df["Quantity"])
plt.show()
```

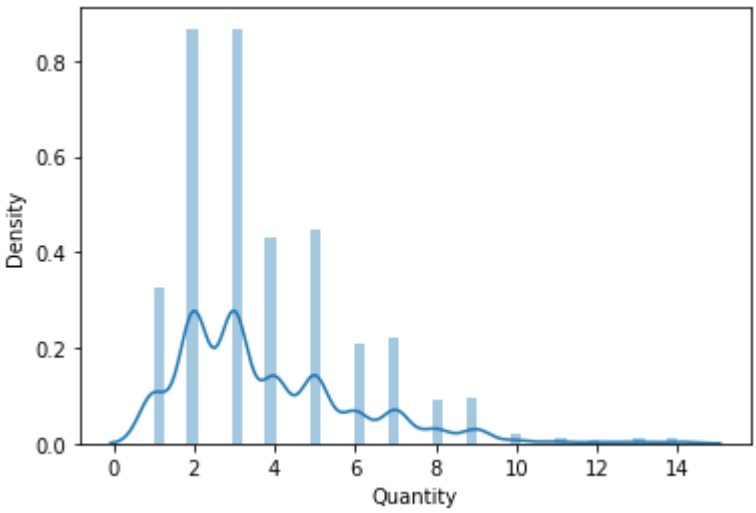
```
C:\Users\Pranil Rego\AppData\Local\Temp\ipykernel_23928\903518770.py:1: UserWarnin
g:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.gith
ub.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(df["Quantity"])
```



In [31]:
```
sns.distplot(df["Discount"])
plt.show()
```

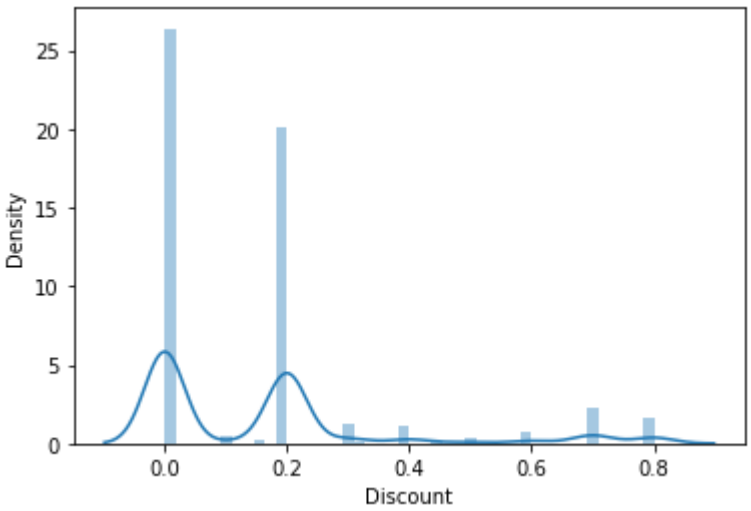
```
C:\Users\Pranil Rego\AppData\Local\Temp\ipykernel_23928\3783034909.py:1: UserWarnin
g:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.gith
ub.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(df["Discount"])
```



In [32]:
```
sns.distplot(df["Profit"])
plt.show()
```

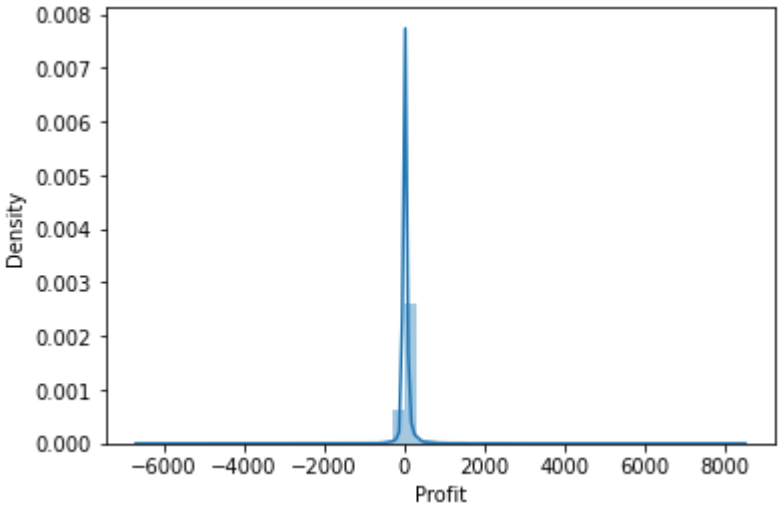
```
C:\Users\Pranil Rego\AppData\Local\Temp\ipykernel_23928\861365785.py:1: UserWarnin
g:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751 (https://gist.gith
ub.com/mwaskom/de44147ed2974457ad6372750bbe5751)

  sns.distplot(df["Profit"])
```




In [33]:
```
corr = df.corr()
corr
```

Out[33]:

| | Sales | Quantity | Discount | Profit |
|---|---|---|---|---|
| **Sales** | 1.000000 | 0.200469 | -0.028625 | 0.479078 |
| **Quantity** | 0.200469 | 1.000000 | 0.008307 | 0.066089 |
| **Discount** | -0.028625 | 0.008307 | 1.000000 | -0.219939 |
| **Profit** | 0.479078 | 0.066089 | -0.219939 | 1.000000 |

In [34]:
```
sns.heatmap(corr, annot = True)
```

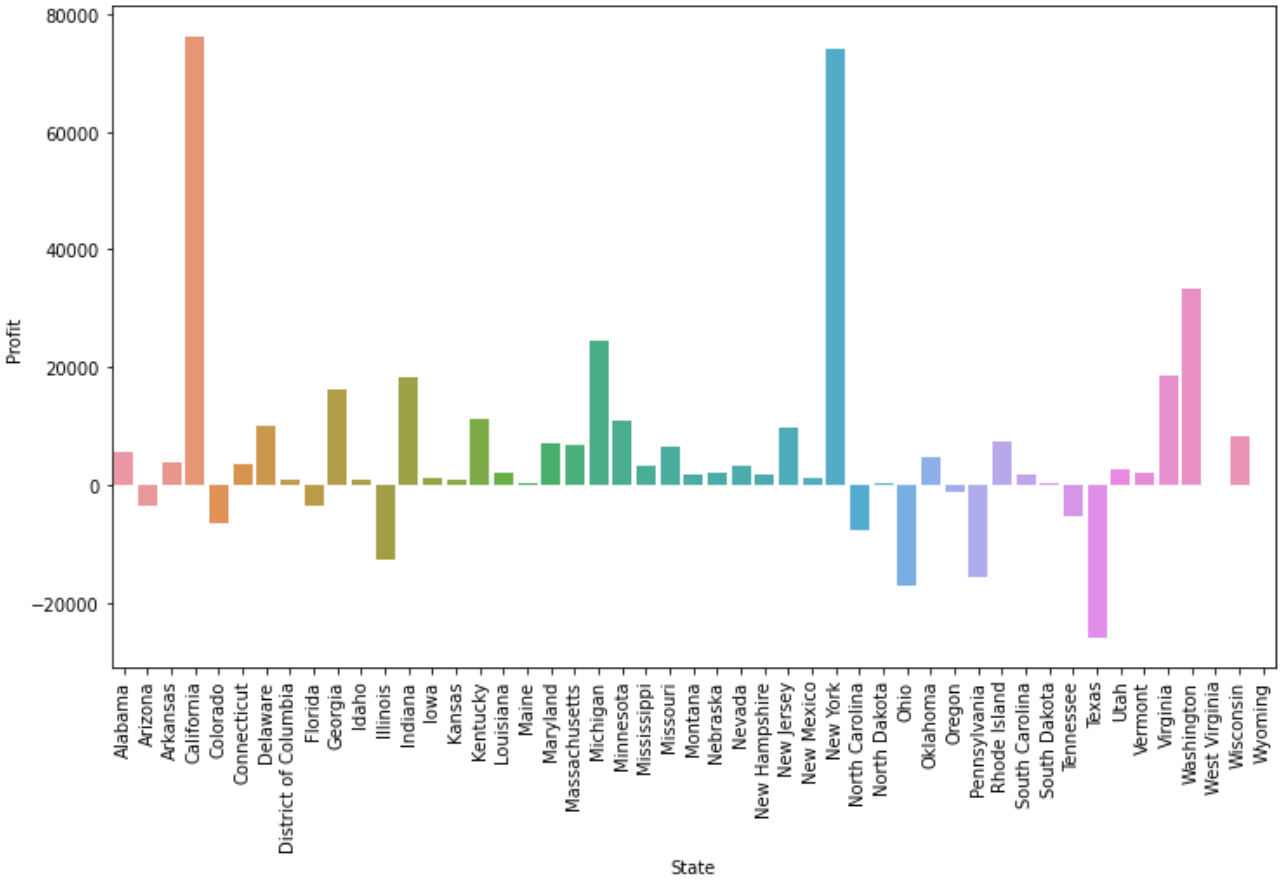
Out[34]: <AxesSubplot:>


In [35]:
```
sales_df = df.groupby("State")["Profit"].sum()
sales_df = sales_df.reset_index()
sales_df.head()
```

Out[35]:

| | State | Profit |
|---|---|---|
| 0 | Alabama | 5786.8253 |
| 1 | Arizona | -3427.9246 |
| 2 | Arkansas | 4008.6871 |
| 3 | California | 76215.9705 |
| 4 | Colorado | -6527.8579 |

In [36]:
```
plt.figure(figsize=(12,7))
sns.barplot(x= sales_df["State"] , y = sales_df["Profit"])
plt.ylabel("Profit")
plt.xlabel("State")
plt.xticks(rotation = "vertical")
plt.show()
```




**We can infer from the above barplot that the states - California and New York are having the highest profit while Ohio, Pennsylvania and Texas are having the highest losses or least profits(negative)**

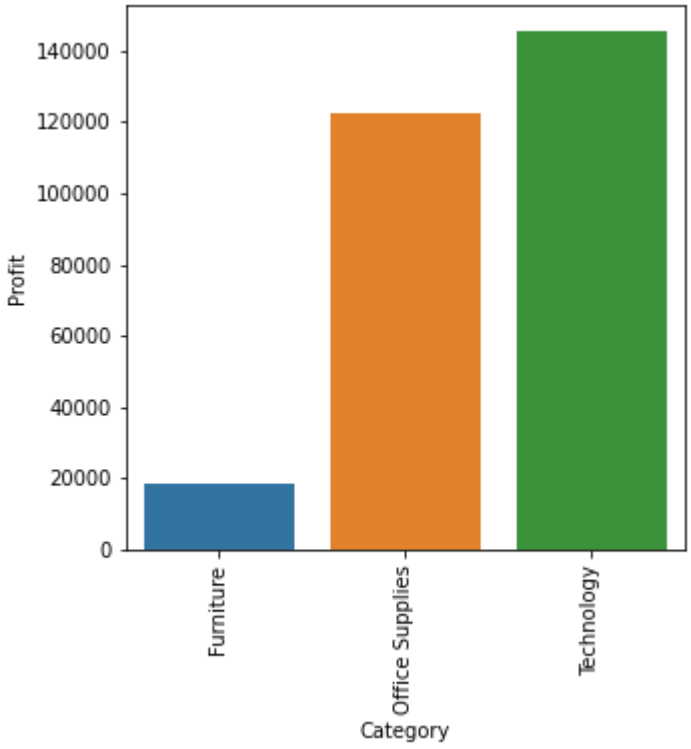
# Category vs Profit

In [37]:
```
category_df = df.groupby("Category")["Profit"].sum()
category_df = category_df.to_frame().reset_index()
category_df
```

Out[37]:

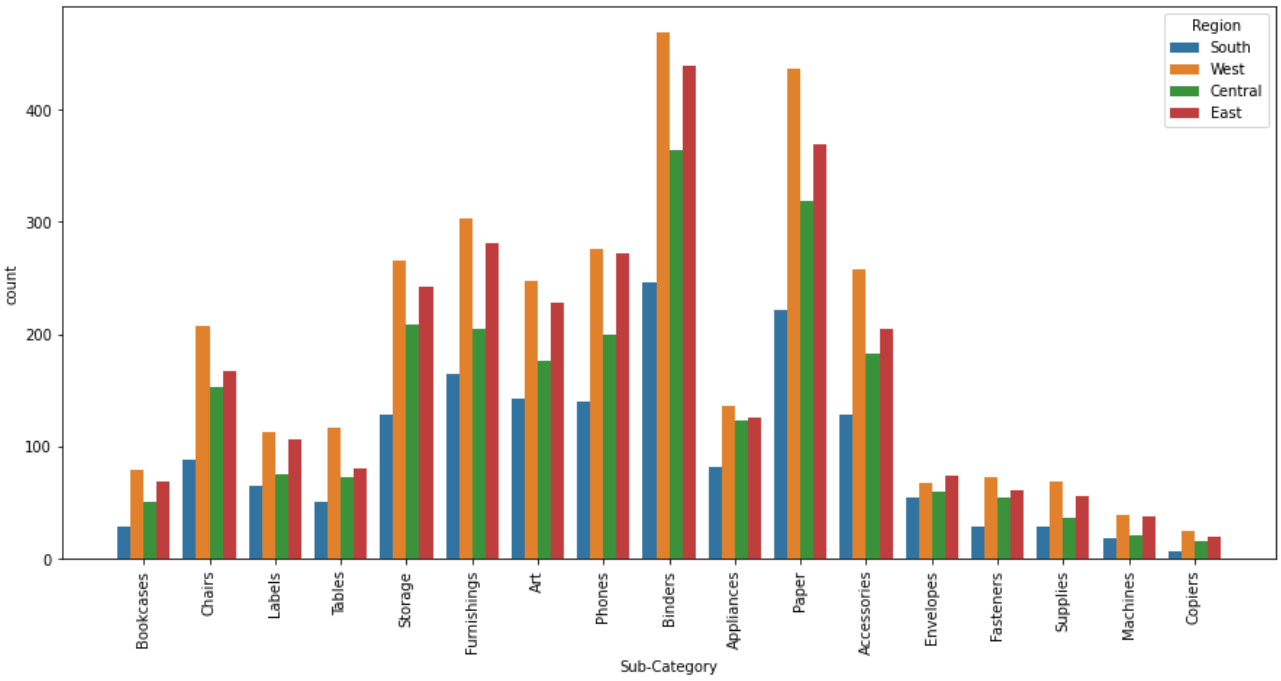
| | Category | Profit |
|---|---|---|
| **0** | Furniture | 18484.9459 |
| **1** | Office Supplies | 122196.0726 |
| **2** | Technology | 145416.5394 |

In [38]:
```
plt.figure(figsize = (5,5))
sns.barplot(x = category_df["Category"], y = category_df["Profit"])
plt.ylabel("Profit")
plt.xlabel("Category")
plt.xticks(rotation="vertical")
plt.show()
```




In [41]:
```
plt.figure(figsize=(15,7))
sns.countplot(x="Sub-Category", hue= "Region", data=df)
plt.xticks(rotation="vertical")
plt.plot()
```

Out[41]: []




Copiers, Machines and Supplies are the least sold products overall . South accounts for the least sales in any of the sub-categories.

# Sales per State

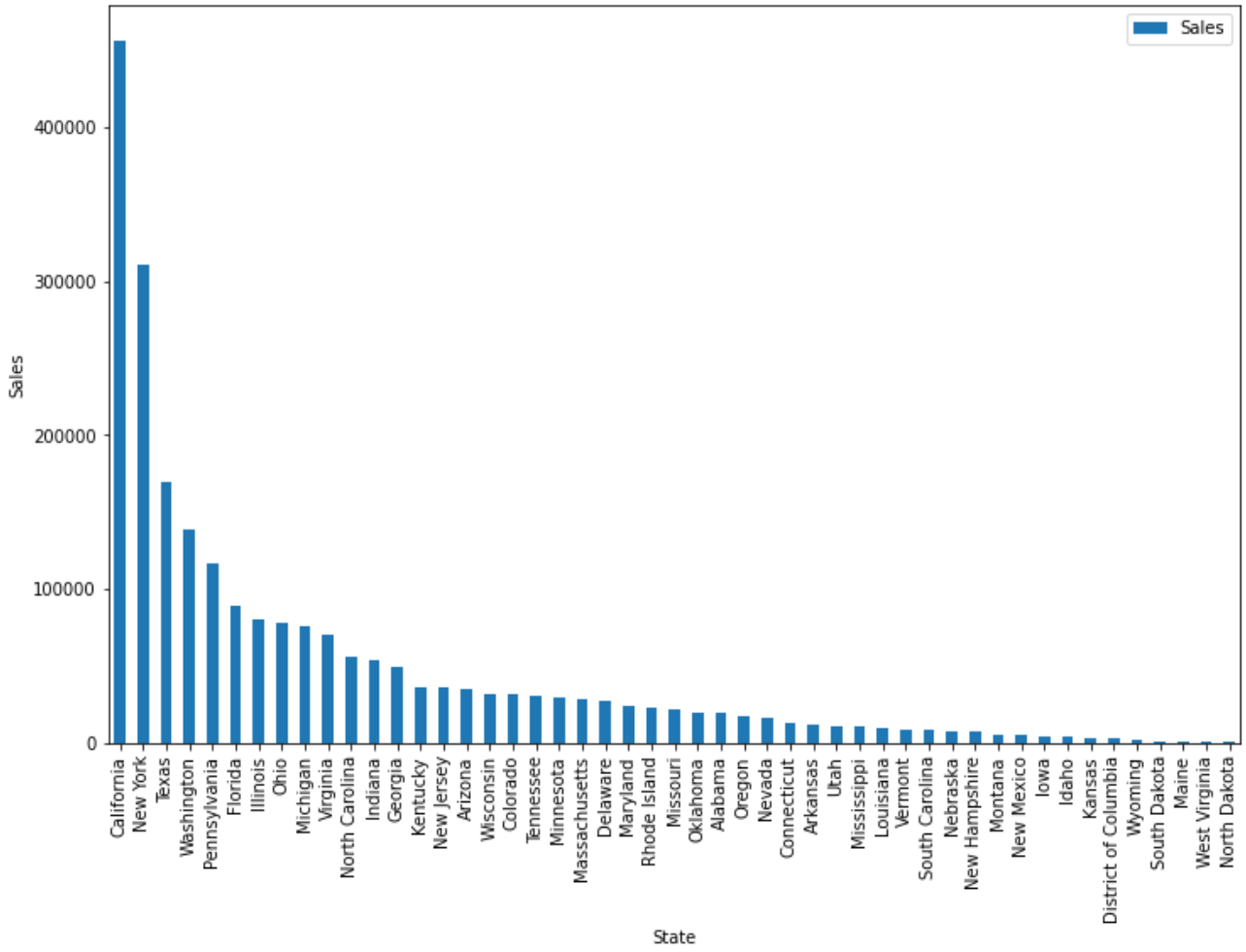
In [42]:
```
df_state = df.groupby("State")["Sales"].sum().sort_values(ascending = False)
df_state = df_state.to_frame().reset_index()
df_state.head()
```

Out[42]:

| | State | Sales |
|---|---|---|
| **0** | California | 456629.9285 |
| **1** | New York | 310349.2150 |
| **2** | Texas | 170101.1278 |
| **3** | Washington | 138560.8100 |
| **4** | Pennsylvania | 116383.0100 |

In [43]:
```
df_state.plot(kind = "bar" , x = "State" ,  y = "Sales" , figsize = (12,8))
plt.ylabel("Sales")
```

Out[43]: Text(0, 0.5, 'Sales')

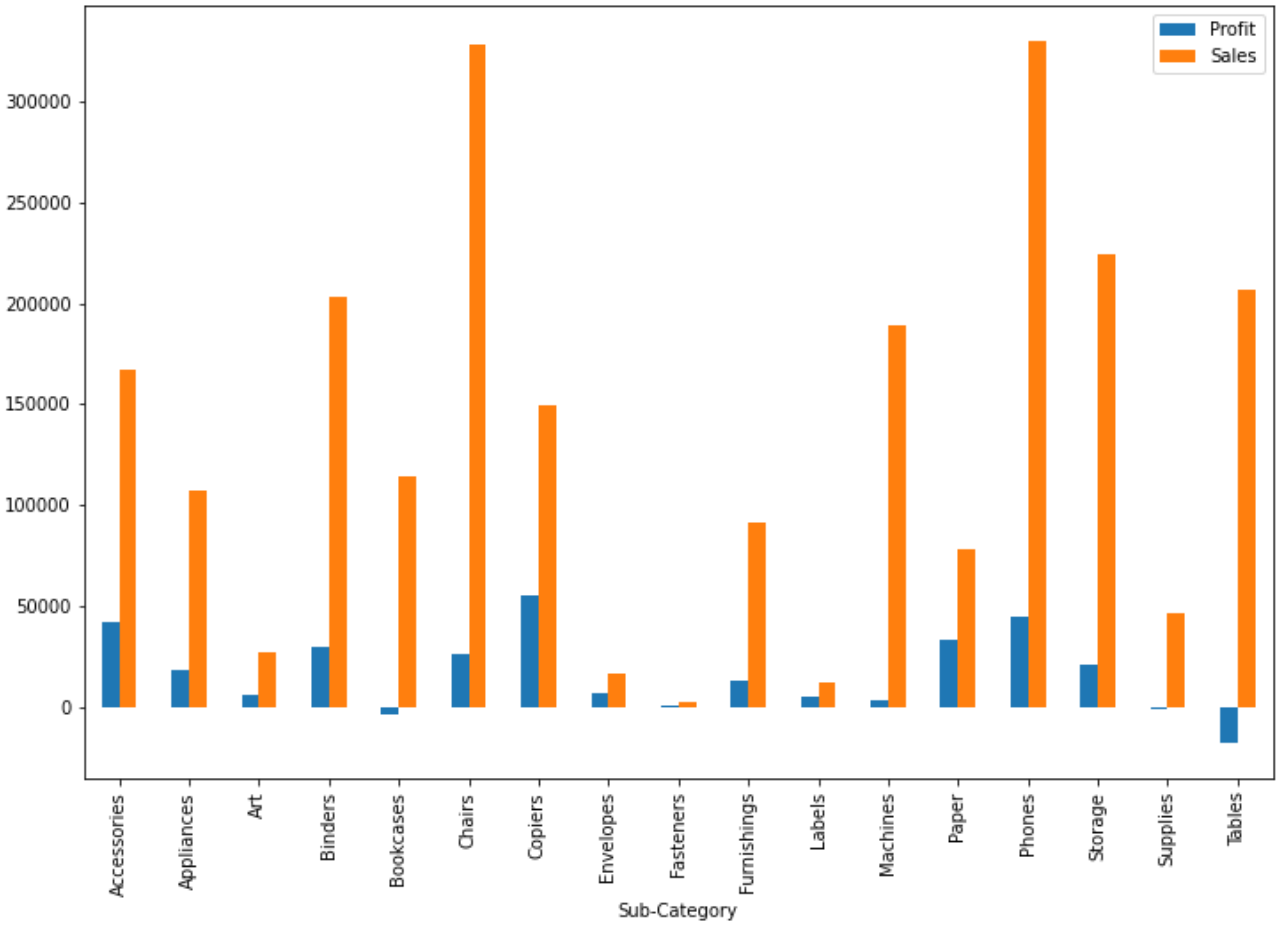


# Highest sales- California, New York and Texas

# Profit and sales for sub-categories

In [44]:
```
sub_df = df.groupby("Sub-Category")["Profit" , "Sales"].sum()
sub_df.plot(kind = "bar"  ,figsize = (12 , 8))
plt.show()
```

```
C:\Users\Pranil Rego\AppData\Local\Temp\ipykernel_23928\3316674198.py:1: FutureWarn
ing: Indexing with multiple keys (implicitly converted to a tuple of keys) will be
deprecated, use a list instead.
  sub_df = df.groupby("Sub-Category")["Profit" , "Sales"].sum()
```




Conclusion :

● We can conclude that our sales are higher as compared to our profit.

● Tables, Bookcases and Supplies are responsible for maximum losses( in negative)

Weak Areas :

● Though Copiers are the least sold products it makes most of the profit so, we must look for ways to improve the sales of the Copiers.

● Tables should either be removed from the market or major changes should be made to tables in order to not incur losses in future.

● We should try to improve our sales in North Dakota, South Dakota , West Virginia and Columbia using new techniques.

● Our sales at Illinois, Ohio, Texas and Pennsylvania are making losses so we must concentrate on the loss making issues in this region.

● So , in order to improve our sales and profit we must pay special attention to our losses and strengthen our weak areas as mentioned above.

In [ ]: