GRIP: The Spark Foundation

Data Science And Bussiness Analytics Intern

Author: Shalaka Patil

Task 3 : Exploratory Data Analysis- Retail

Performe 'Exploratory Data Analysis' on dataset 'Samplesuperstore' As a Bussiness Manager try to find out the weak areas where you can work to make more profit. What all bussiness problems you can derive by Explorating the data ? I used python to perform EDA on this Dataset

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("C:/Users/DELL/Downloads/SampleSuperstore.csv")
```

Basic Data Insights

In [3]:
```
df.sample(5)
```

Out[3]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
6717	Standard Class	Home Office	United States	Hot Springs	Arkansas	71901	South	Office Supplies	Paper	25.920	4	0.0	12.7008
2561	Standard Class	Consumer	United States	Nashville	Tennessee	37211	South	Office Supplies	Labels	9.216	4	0.2	3.3408
1559	Standard Class	Corporate	United States	Seattle	Washington	98103	West	Office Supplies	Binders	35.352	9	0.2	12.8151
7737	First Class	Consumer	United States	Charlotte	North Carolina	28205	South	Office Supplies	Art	12.672	9	0.2	1.4256
496	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Office Supplies	Binders	119.616	8	0.2	40.3704

In [4]:
```
df.head()
```

Out[4]:

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

Out[5]:

	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 [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()
```

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

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

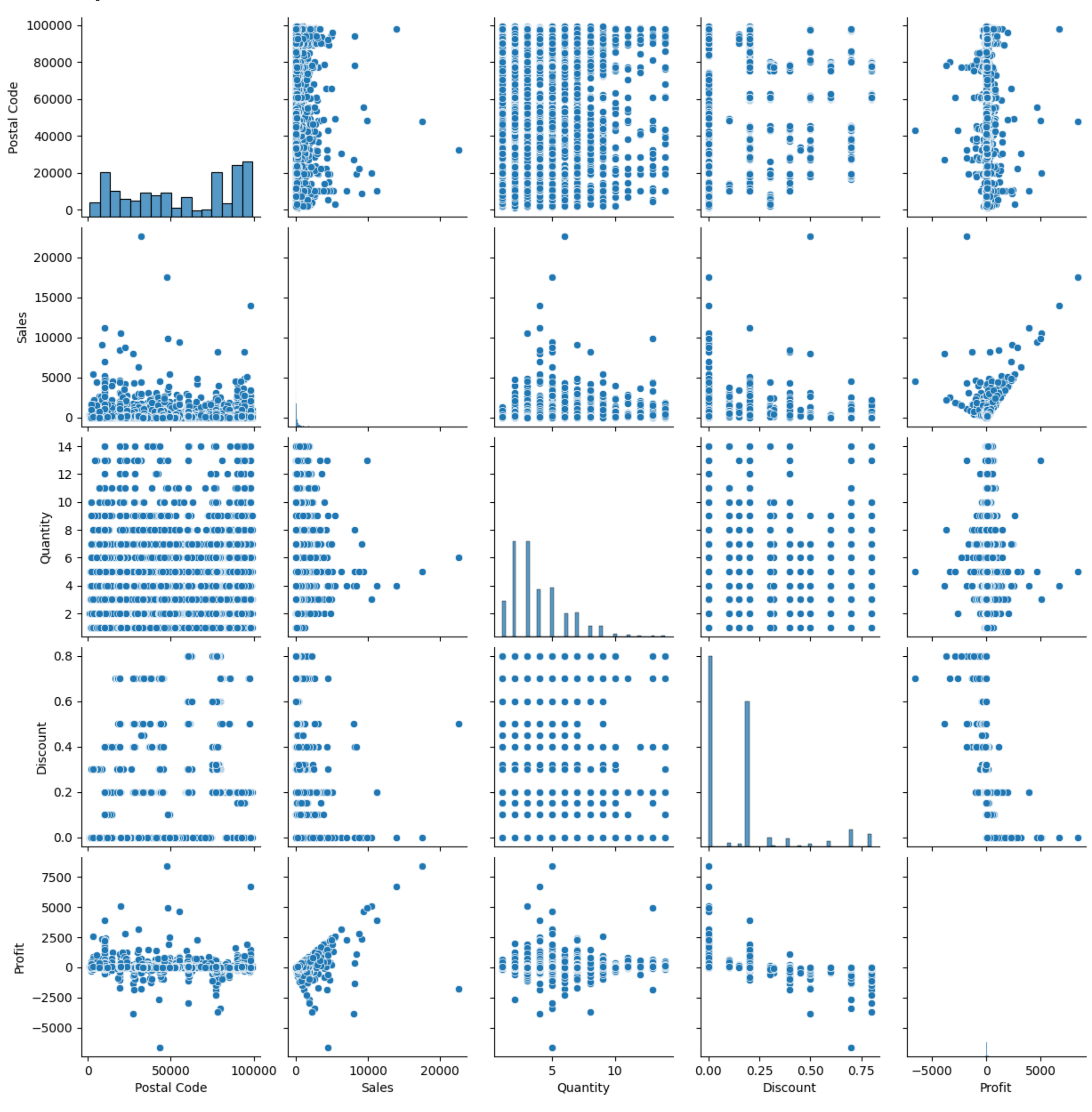
In [10]:
```
df.isnull().sum()
```

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
```

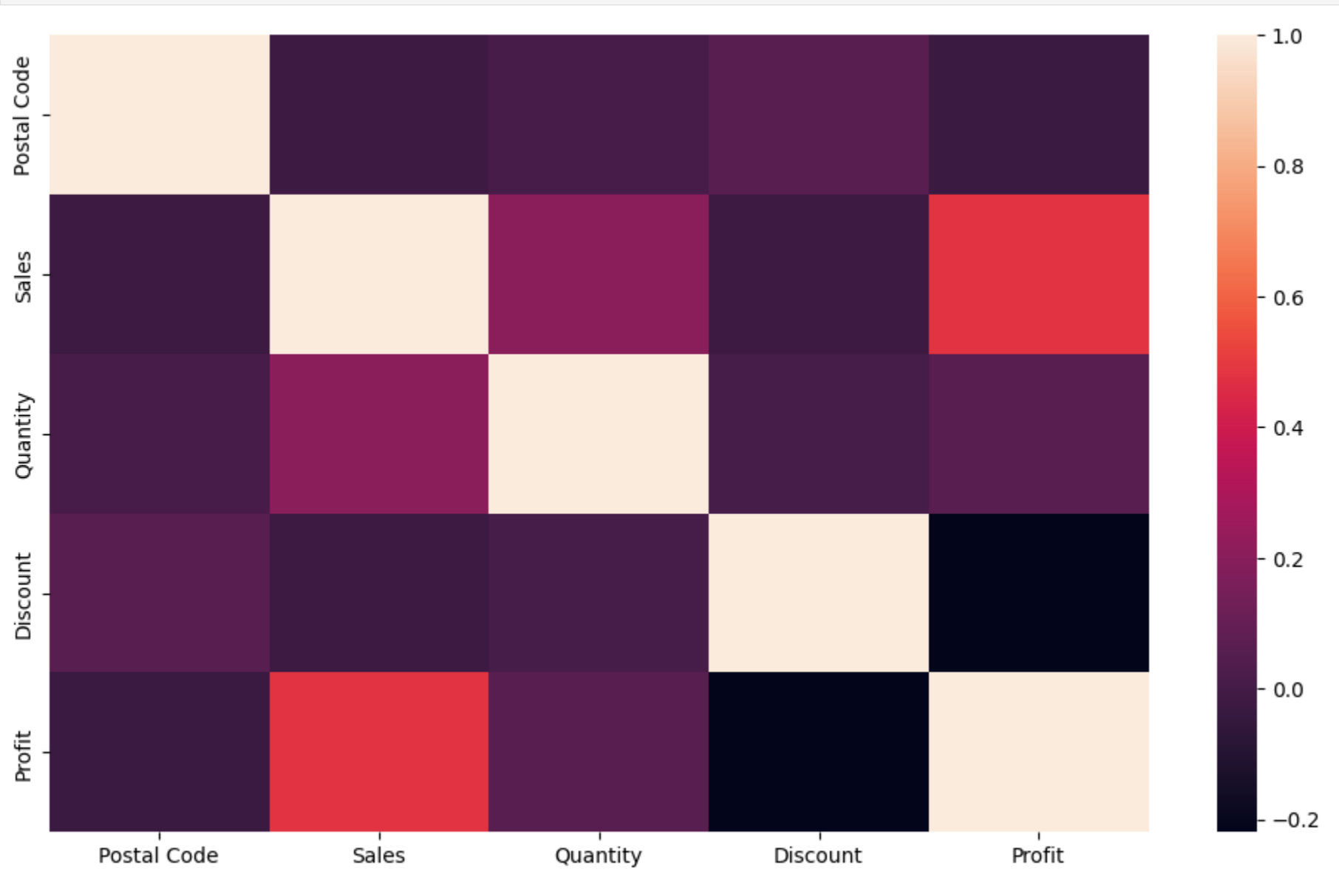
Data Visualization

In [11]:
```
sns.pairplot(df)
```

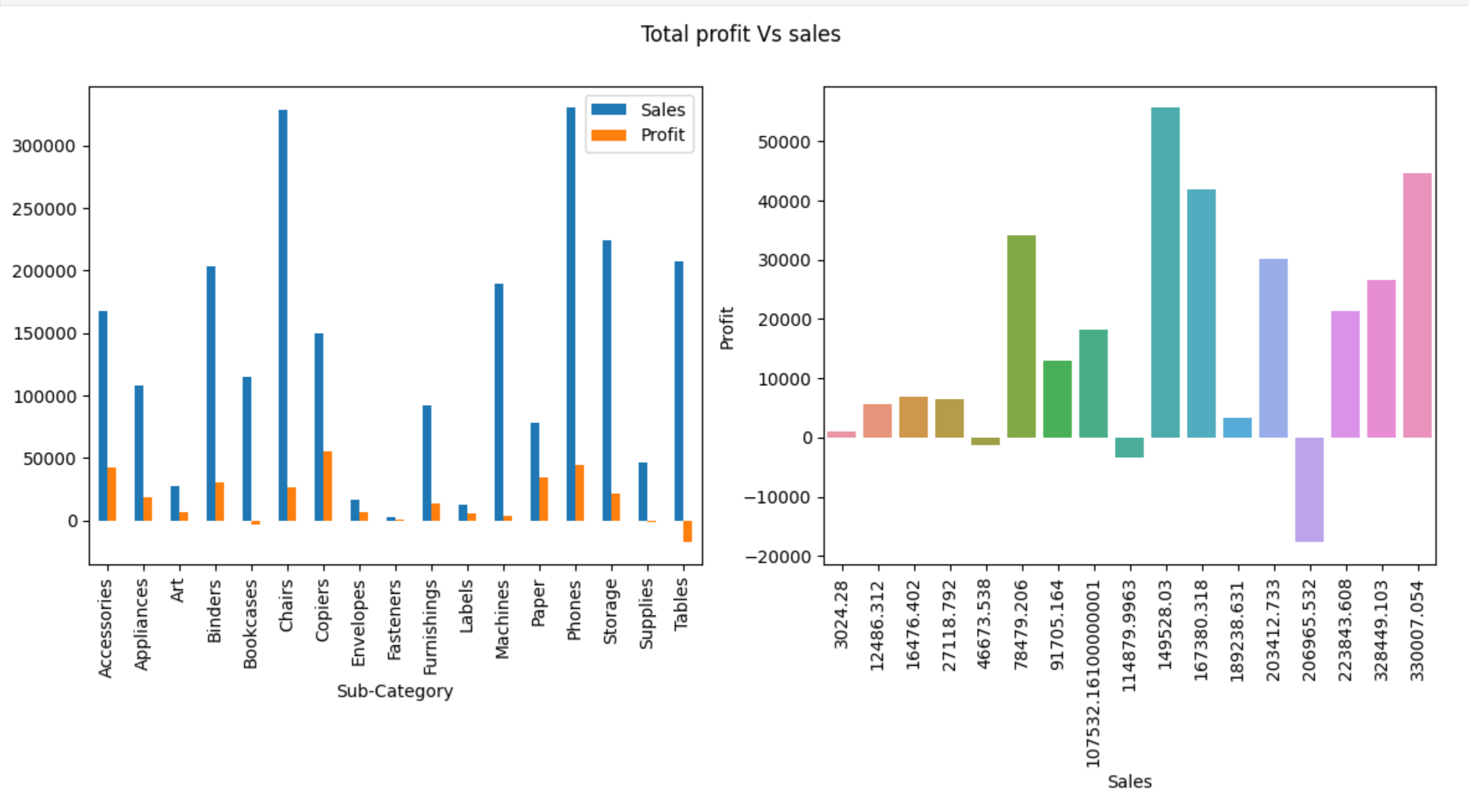
Out[11]:
```
<seaborn.axisgrid.PairGrid at 0x1f4c4a6b370>
```



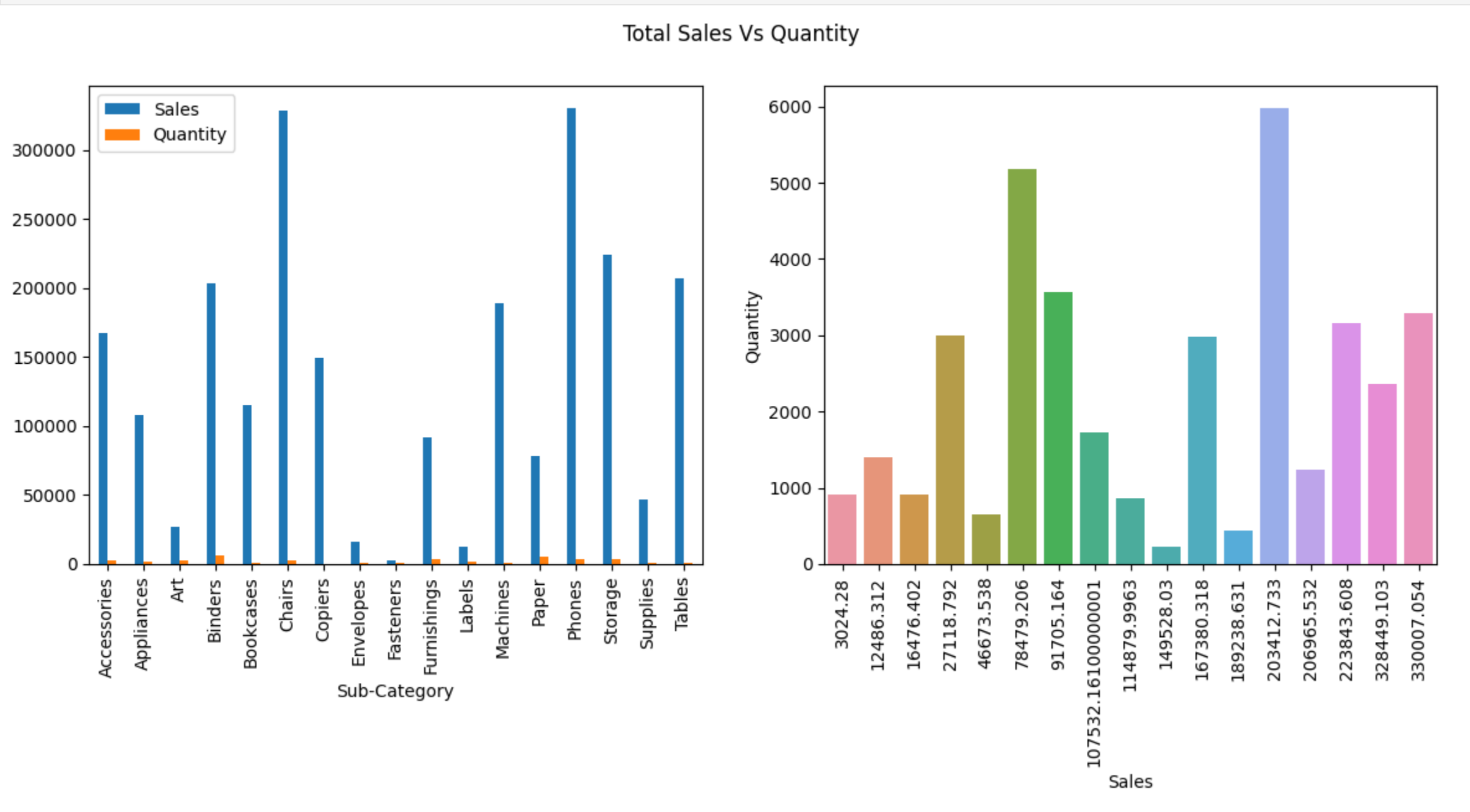
In [12]:
```
fig,axes = plt.subplots(1,1,figsize=(12,7))
sns.heatmap(df.corr())
plt.show()
```



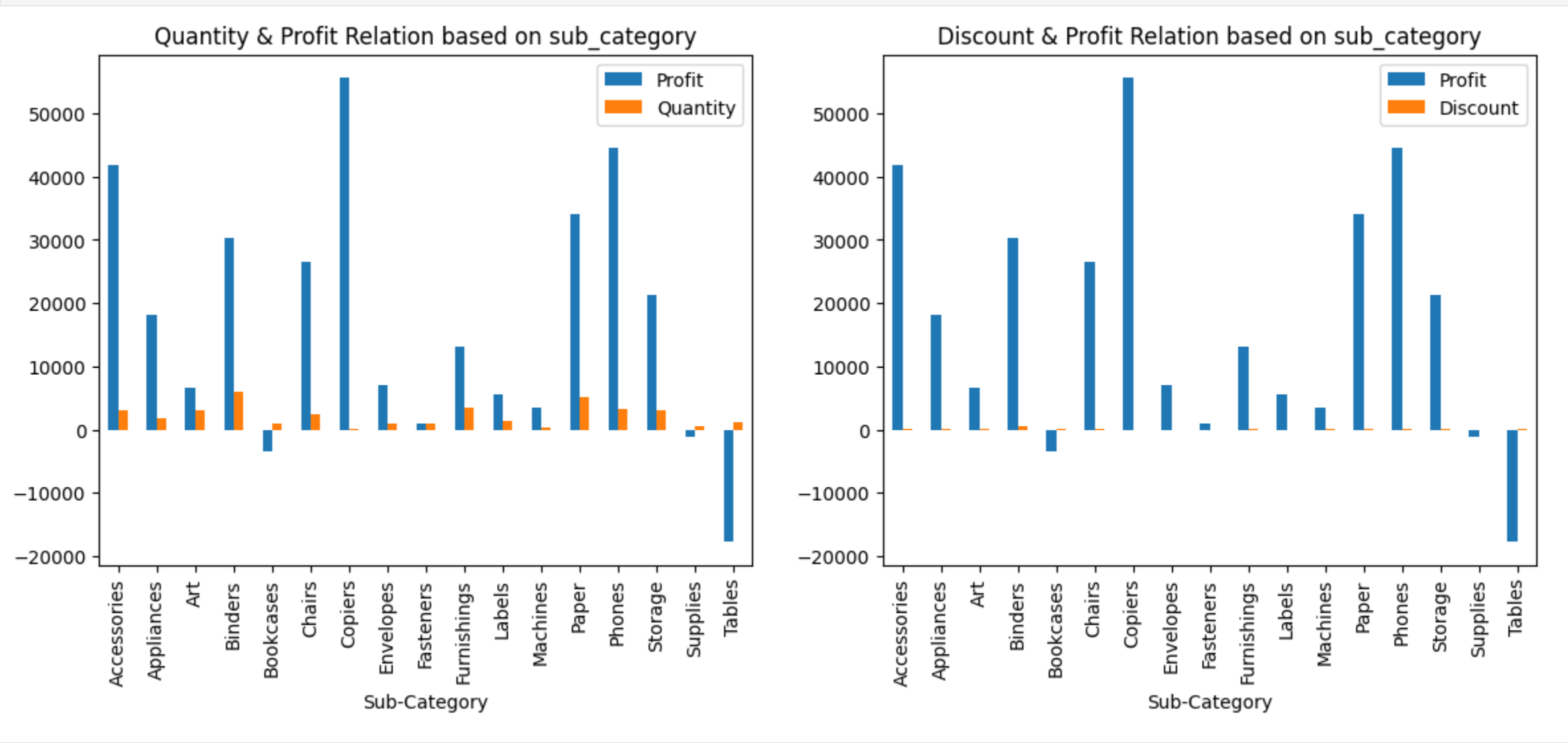
In [13]:
```
fig,axes=plt.subplots(1,2,figsize=(14,5))
fig.suptitle("Total profit Vs sales")
sns.barplot(data=df.groupby ('Sub-Category')['Sales','Profit'].agg(sum),x='Sales',y='Profit',ax=axes[1])
df.groupby('Sub-Category')['Sales','Profit'].agg(sum).plot(kind='bar',ax=axes[0])
plt.xticks (rotation=90)
plt.show()
```



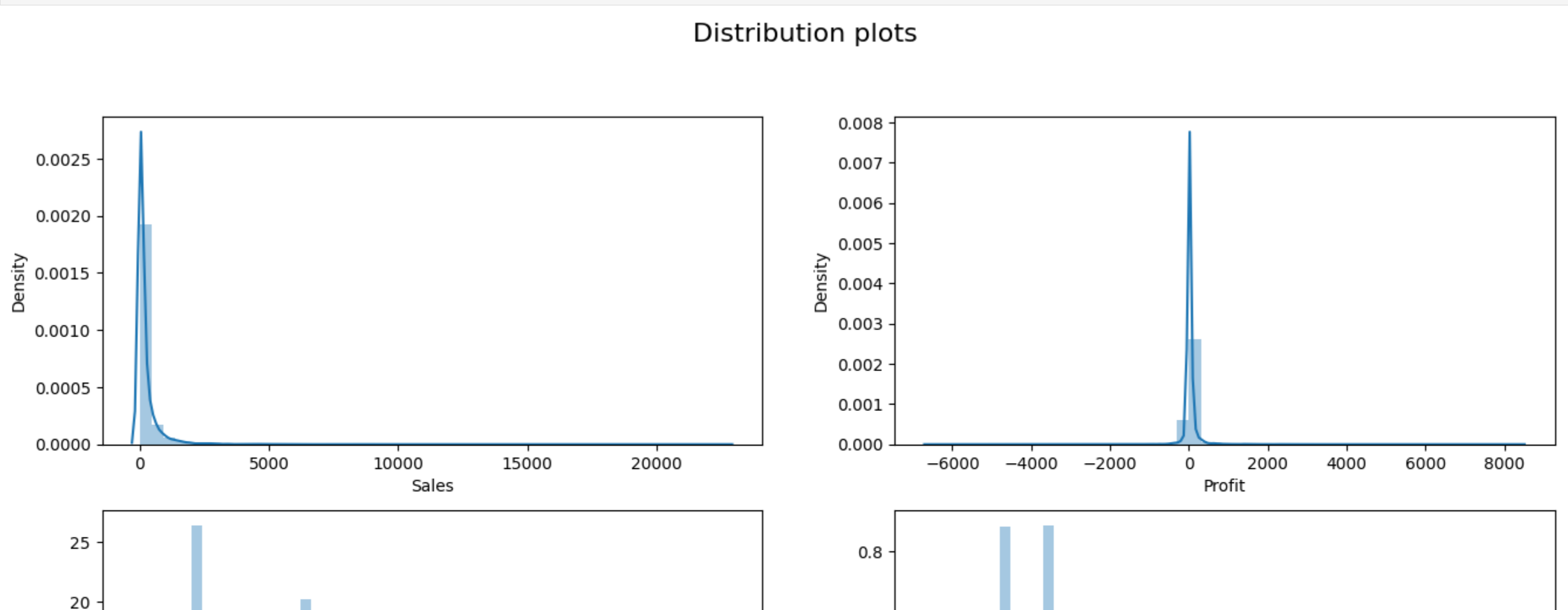
In [14]:
```
fig,axes=plt.subplots(1,2,figsize=(14,5))
fig.suptitle("Total Sales Vs Quantity")
sns.barplot(data=df.groupby ('Sub-Category')['Sales','Quantity'].agg(sum),x='Sales',y='Quantity',ax=axes[1])
df.groupby('Sub-Category')['Sales','Quantity'].agg(sum).plot(kind='bar',ax=axes[0])
plt.xticks (rotation=90)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



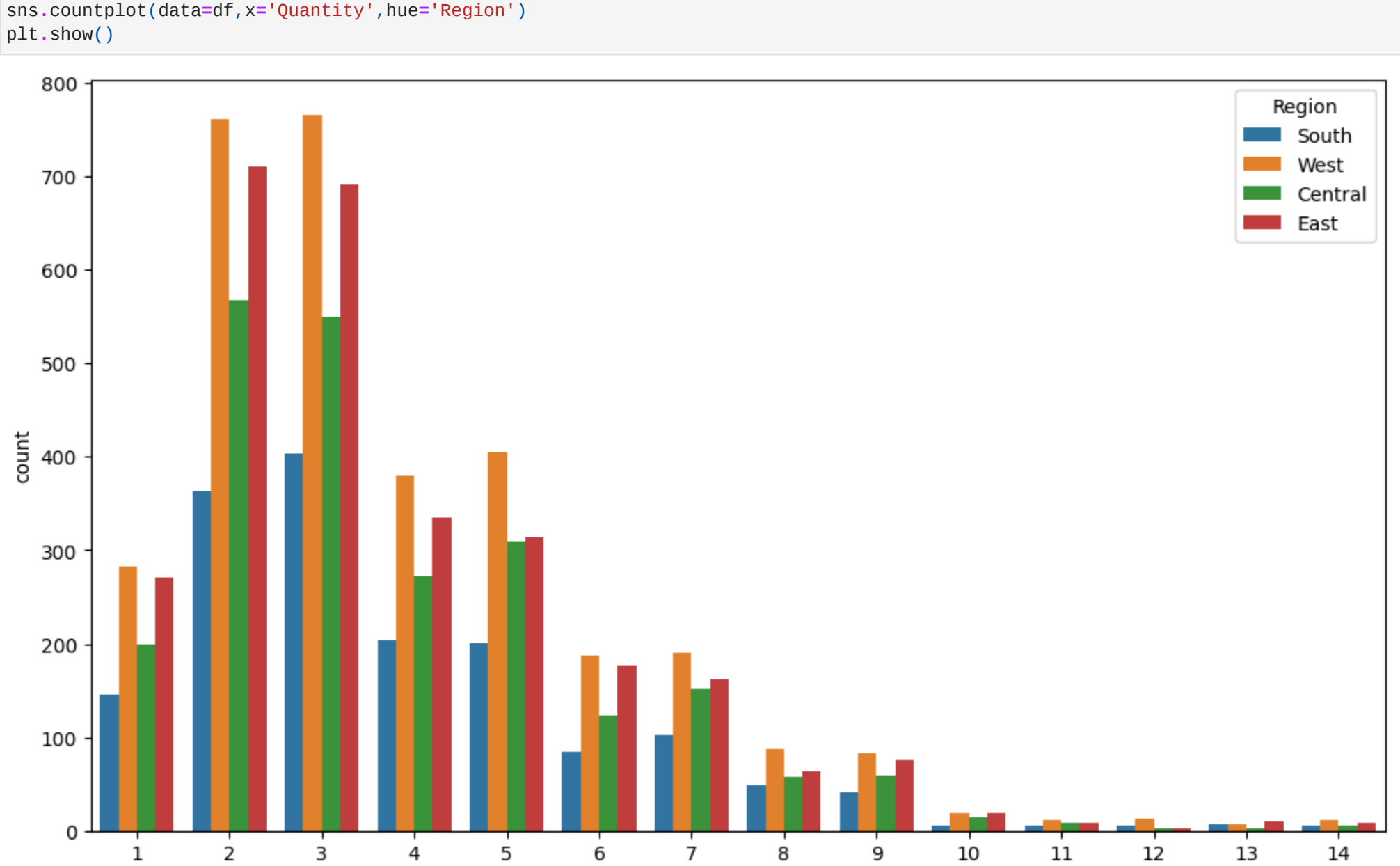
In [15]:
```
fig,axes = plt.subplots(1,2,figsize=(14,5))
df.groupby('Sub-Category')['Profit','Quantity'].agg(sum).plot(kind='bar',ax=axes[0]).set_title('Quantity & Profit Relation based on sub_category')
df.groupby('Sub-Category')['Profit','Discount'].agg(sum).plot(kind='bar',ax=axes[1]).set_title('Discount & Profit Relation based on sub_category')
plt.xticks (rotation=90)
plt.show()
```



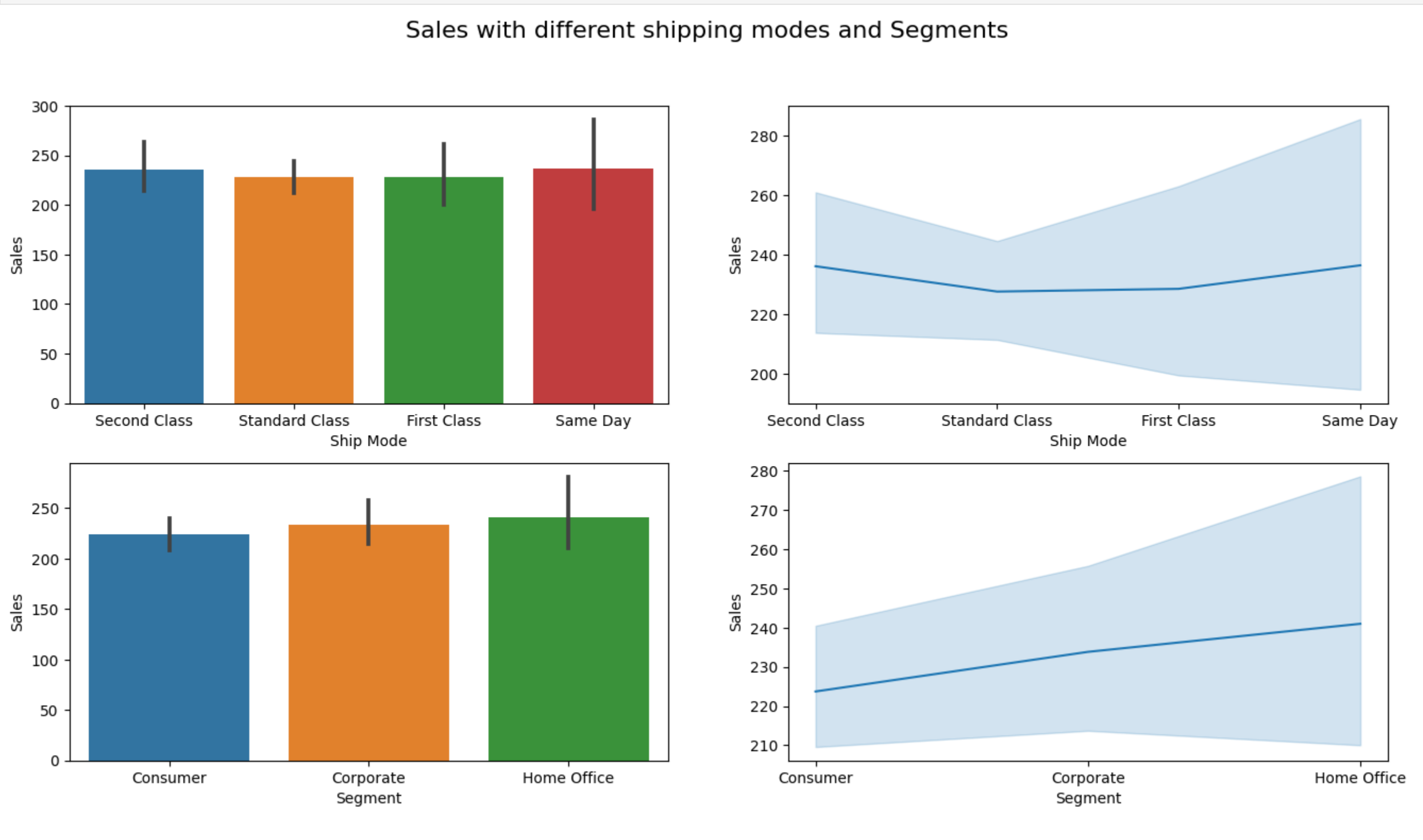
In [16]:
```
fig,axes= plt.subplots(2,2,figsize=(16,8))
fig.suptitle("Distribution plots",fontsize=16)
sns.distplot(df['Sales'],ax=axes[0,0])
sns.distplot(df['Profit'],ax=axes[0,1])
sns.distplot(df['Discount'],ax=axes[1,0])
sns.distplot(df['Quantity'],ax=axes[1,1])
plt.show()
```



In [17]:
```
fig,ax= plt.subplots(1,1,figsize=(12,7))
sns.countplot(data=df,x='Quantity',hue='Region')
plt.show()
```



In [18]:
```
fig,axes = plt.subplots(2,2,figsize=(16,8))
fig.suptitle("Sales with different shipping modes and Segments", fontsize=16)
sns.barplot(data=df,x="Ship Mode",y="Sales",ax=axes[0,0])
sns.lineplot(data=df,x="Ship Mode",y="Sales",ax=axes[0,1])
sns.barplot(data=df,x="Segment",y="Sales",ax=axes[1,0])
sns.lineplot(data=df,x="Segment",y="Sales",ax=axes[1,1])
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.