NAME: KRITHICK BALAJI RAMESH

ROLL_NO:RA2111003011318

SRM EMAIL: kr5623@srmist.edu.in

PERSONAL EMAIL ID: krithickbalaji2@gmail.com

CONTACT NUMBER: +91 6385516155

PROJECT NAME: Exploratory Data Analysis on Superstore’s dataset

Step 1: Importing the libraries and data preprocessing

```
#Importing libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
import scipy.stats as stats
```

Step 2: Importing the data set

```
#Importing the data set
df=pd.read_csv('/content/SampleSuperstore.csv')
```

```
#Checking the shape of the imported data set
df.shape
```

```
(9994, 13)
```

```
#Displaying the features name
df.columns
```

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

```
#Printing the first five rows of the data set
df.head()
```

	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

```
#Printing the last five rows of the data set
df.tail()
```

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit

```
#Basic information of the data
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
```

```
#statistical measure of given data set
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

```
#Checking the NULL values of the given data set
df.isnull().sum()
```

```
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
```

```
#Checking the number of unique values
df.nunique()
```

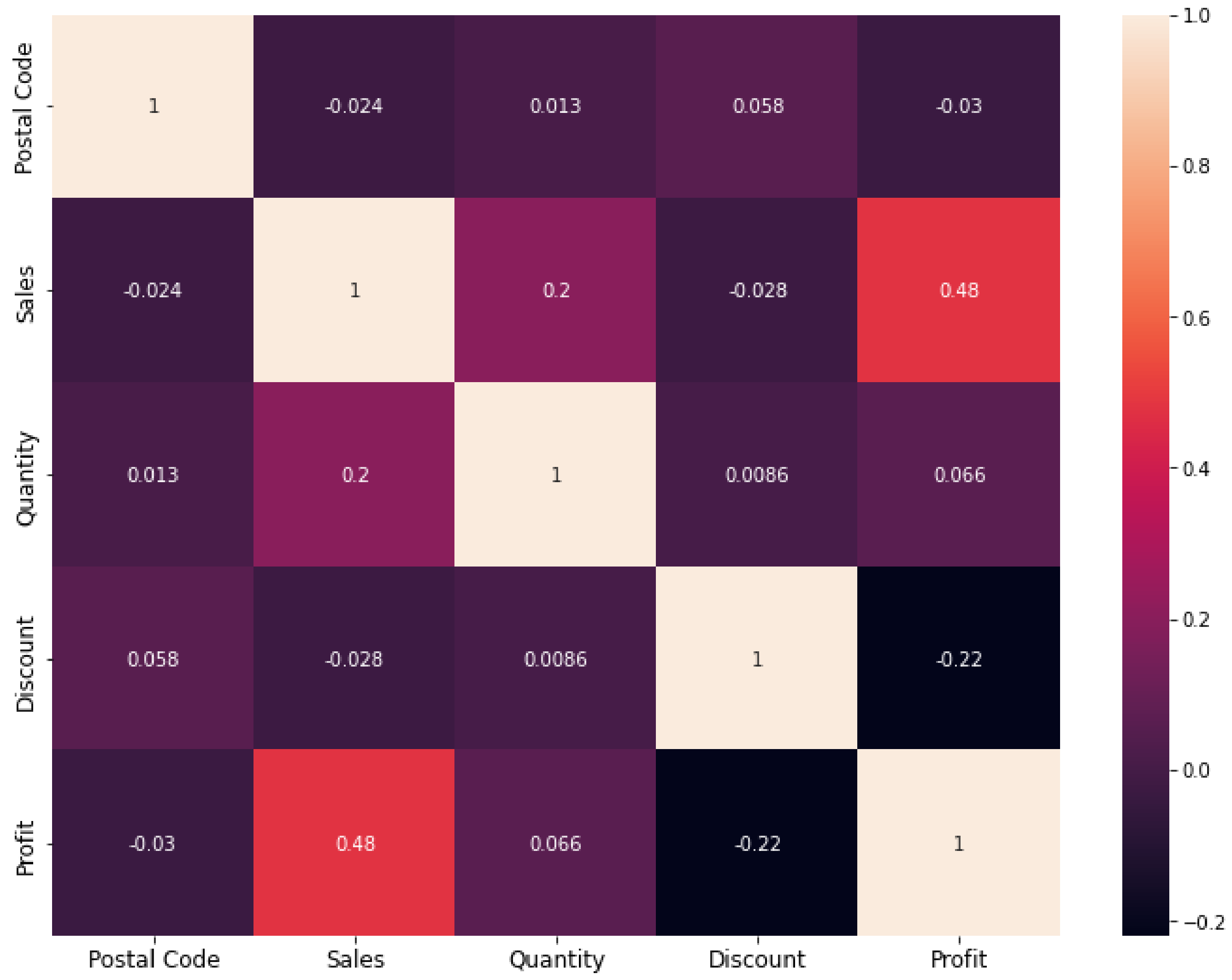
```
Ship Mode          4
Segment            3
Country            1
City             531
State             49
Postal Code      631
Region             4
Category           3
Sub-Category      17
Sales           5825
Quantity          14
Discount          12
Profit          7287
dtype: int64
```

Step 2: Data visualisation using correlation matrix

```
correlation = df.corr()
correlation
```

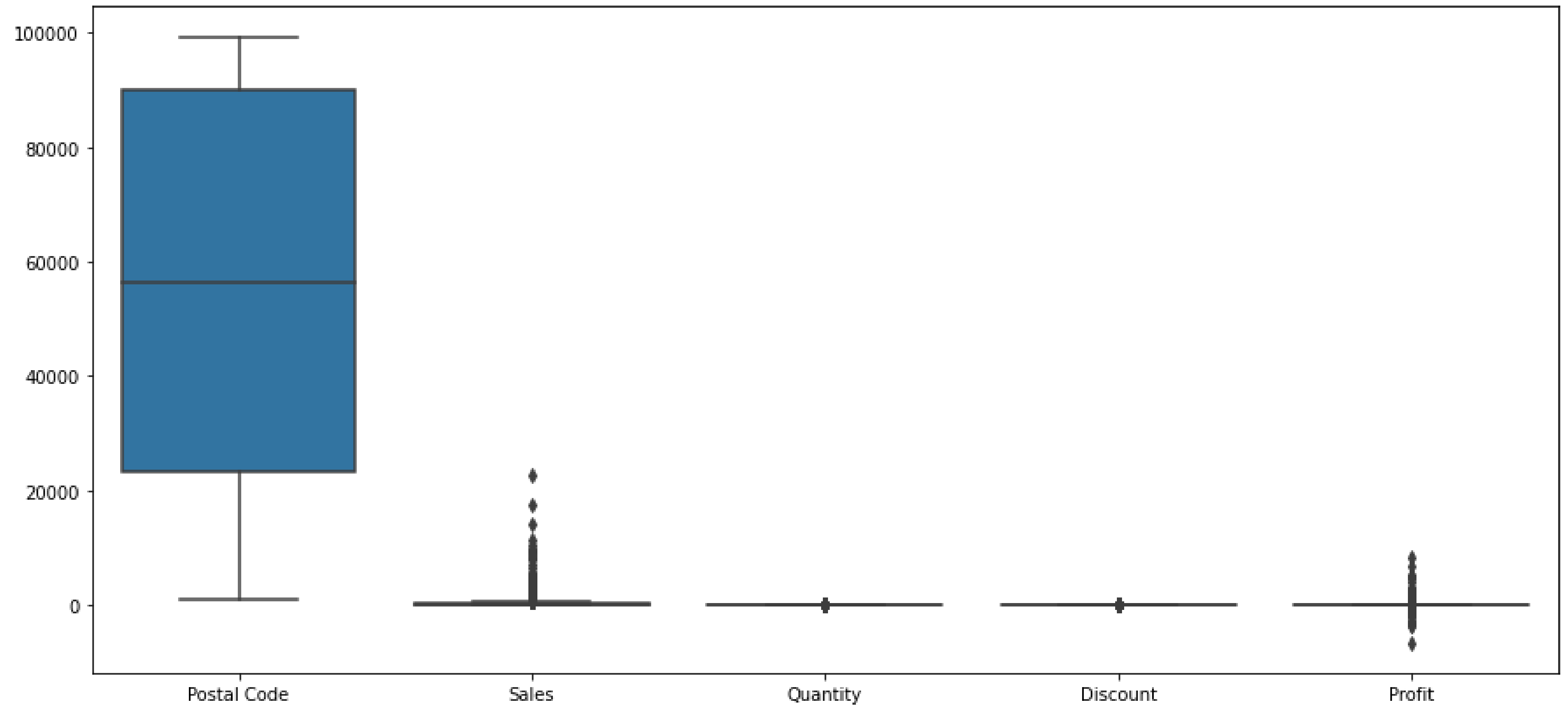
	Postal Code	Sales	Quantity	Discount	Profit
Postal Code	1.000000	-0.023854	0.012761	0.058443	-0.029961
Sales	-0.023854	1.000000	0.200795	-0.028190	0.479064
Quantity	0.012761	0.200795	1.000000	0.008623	0.066253
Discount	0.058443	-0.028190	0.008623	1.000000	-0.219487
Profit	-0.029961	0.479064	0.066253	-0.219487	1.000000

```
#Plotting the heat map by using correlation matrix
plt.figure(figsize=(12,9))
sns.heatmap(correlation,annot=True)
plt.xticks(fontsize=12)
plt.yticks(fontsize=12)
plt.show()
```



```
#we check the outliers of every features using boxplot
plt.figure(figsize=(15,7))
sns.boxplot(data=df)
```

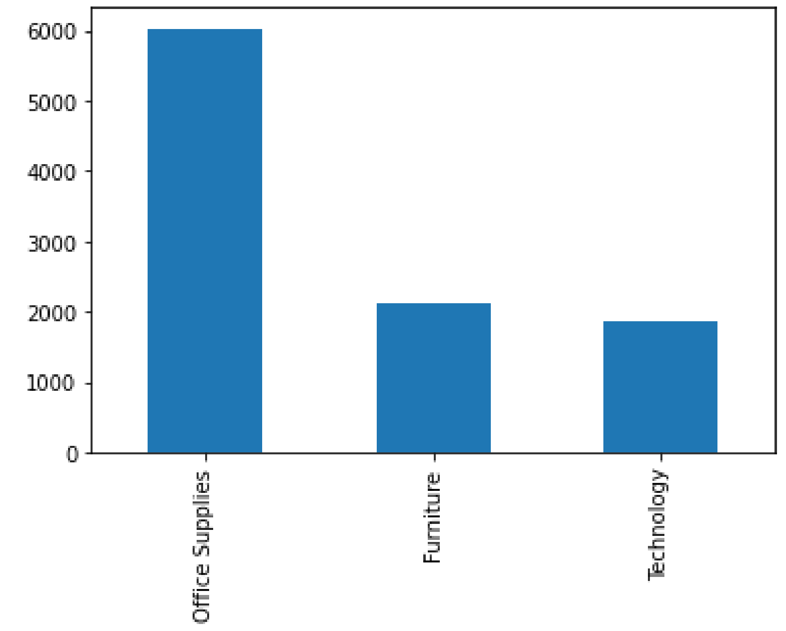
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f60b90076d0>
```



THERE ARE NO OUTLIERS PRESENT AS SUCH

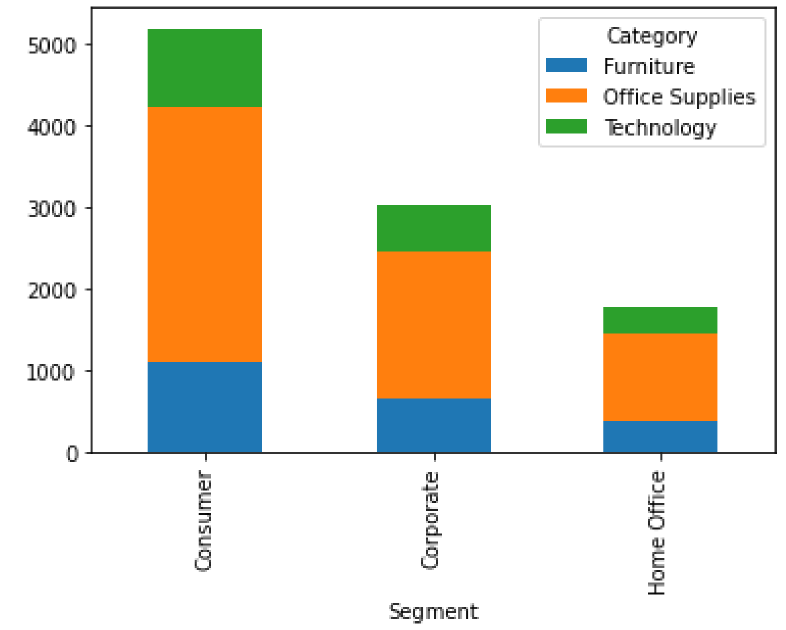
```
df['Category'].value_counts().plot(kind='bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f60b8f21f10>
```



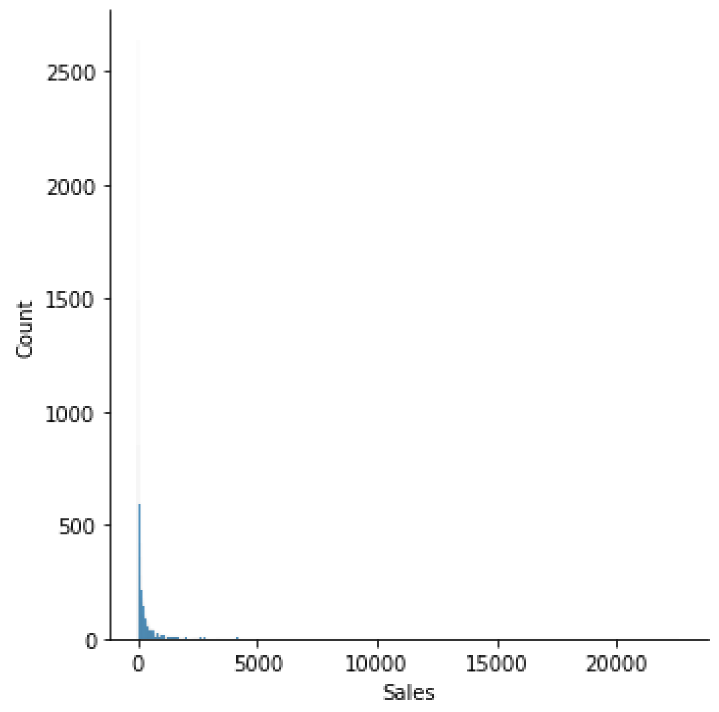
```
#To form a graph showing different categories under each segment
pd.crosstab(df['Segment'],df['Category']).plot(kind = 'bar',stacked=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f60b8ee4090>
```



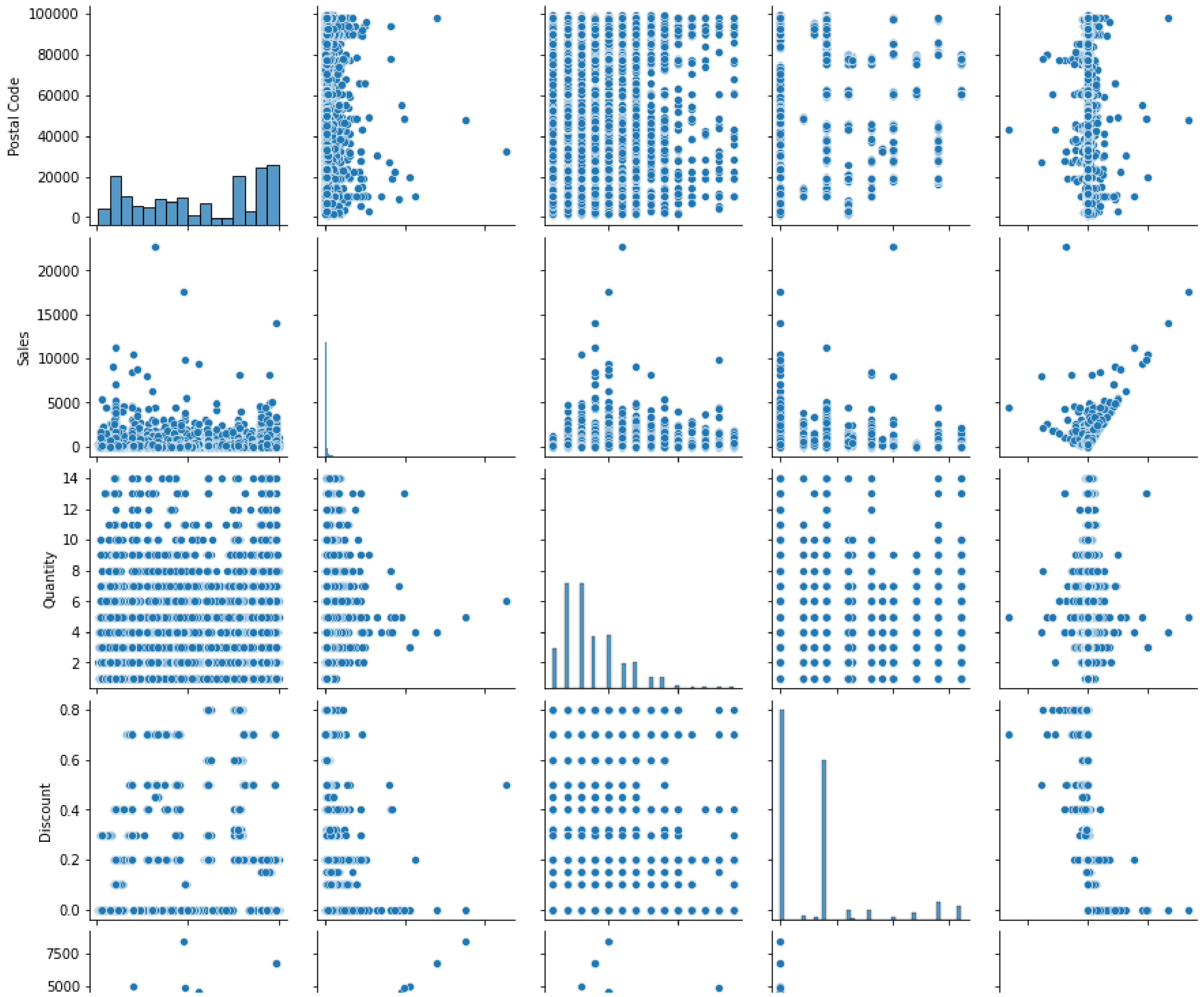
```
sns.displot(df['Sales'])
```

```
<seaborn.axisgrid.FacetGrid at 0x7f60b8e0a350>
```



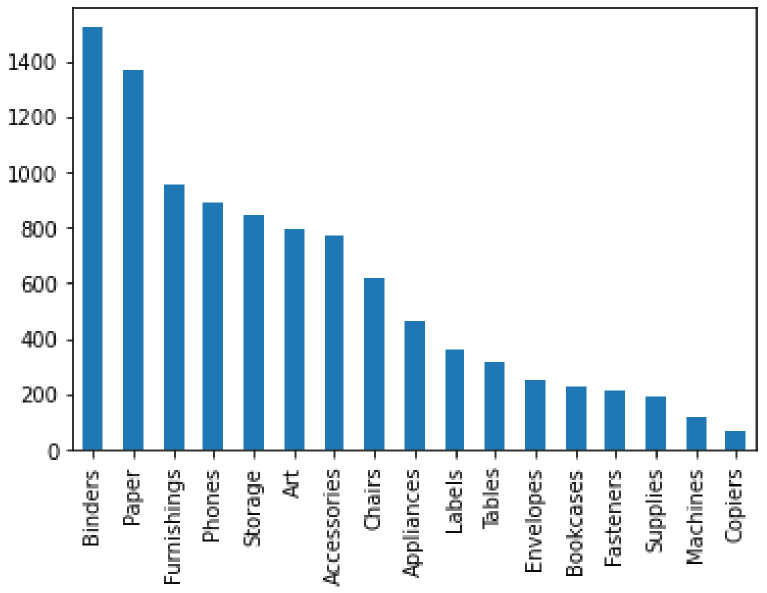
```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7f60b8ceccd0>
```



```
df['Sub-Category'].value_counts().plot(kind = 'bar')
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f60b957b990>
```



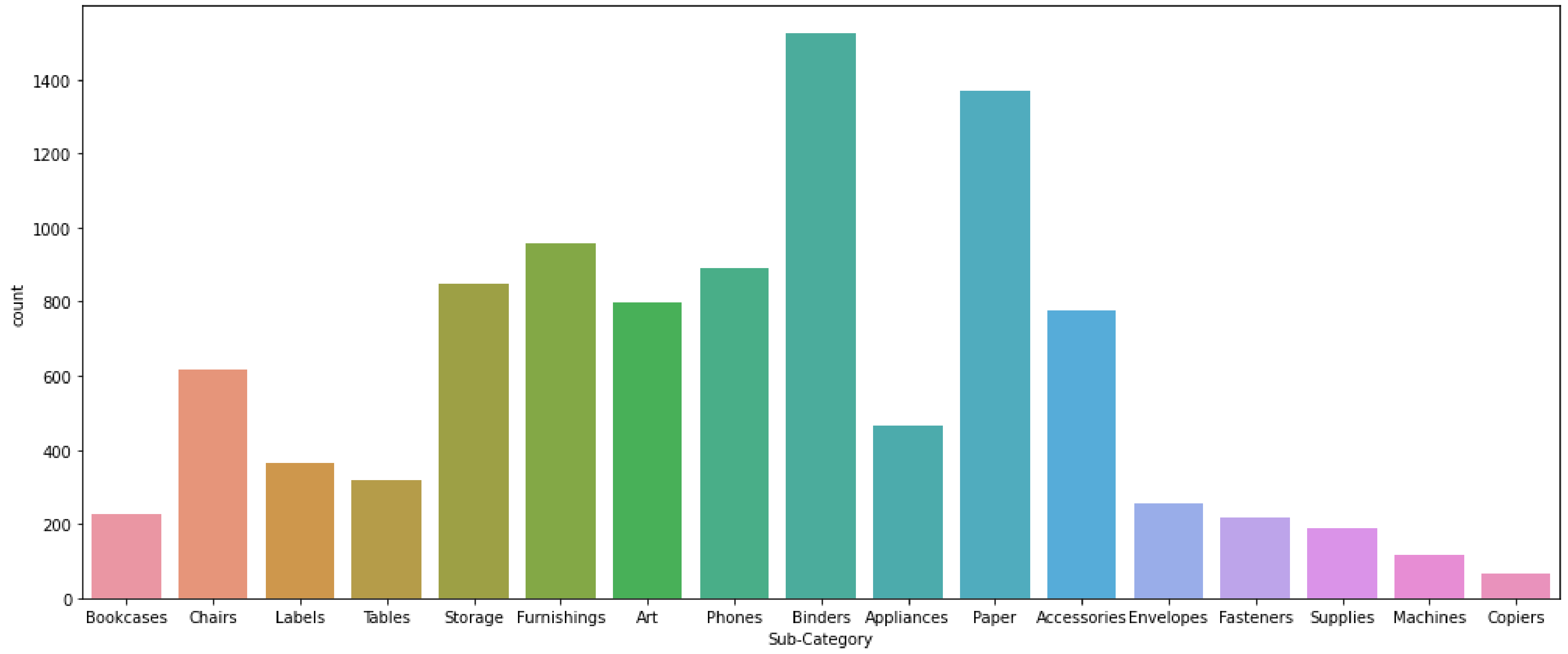
The sub -category is arranged on the basis of most selling products

```
pd.crosstab(df['Region'],df['Category'],df['Profit'],aggfunc='sum').plot(kind="bar",stacked=True)
```

The profit is high when the ship mode is "standard class" and the profit is negligible when the ship mode is "same day".

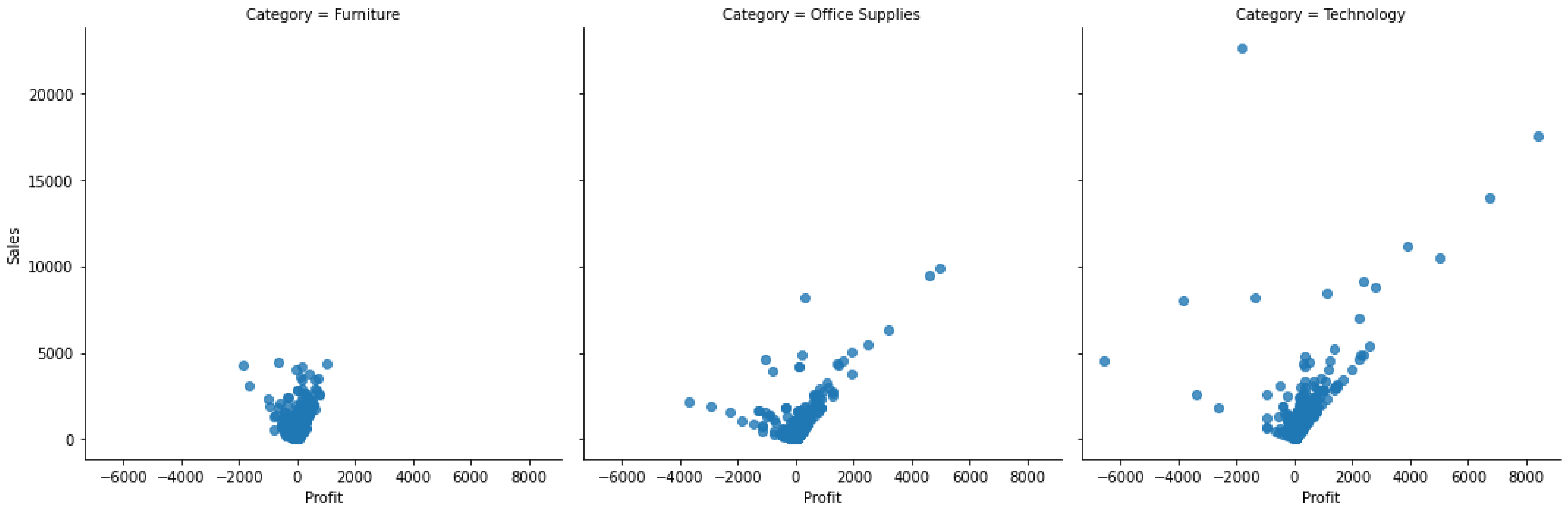
```
plt.figure(figsize=(17,7))
sns.countplot(x=df['Sub-Category'])
print(df['Sub-Category'].value_counts())
```

```
     Binders        1523
     Paper          1370
     Furnishings     957
     Phones          889
     Storage         846
     Art             796
     Accessories     775
     Chairs          617
     Appliances      466
     Labels          364
     Tables          319
     Envelopes       254
     Bookcases       228
     Fasteners       217
     Supplies        190
     Machines        115
     Copiers          68
     Name: Sub-Category, dtype: int64
```



Highest sold sub category is binders and lowest soid sub category is copiers.

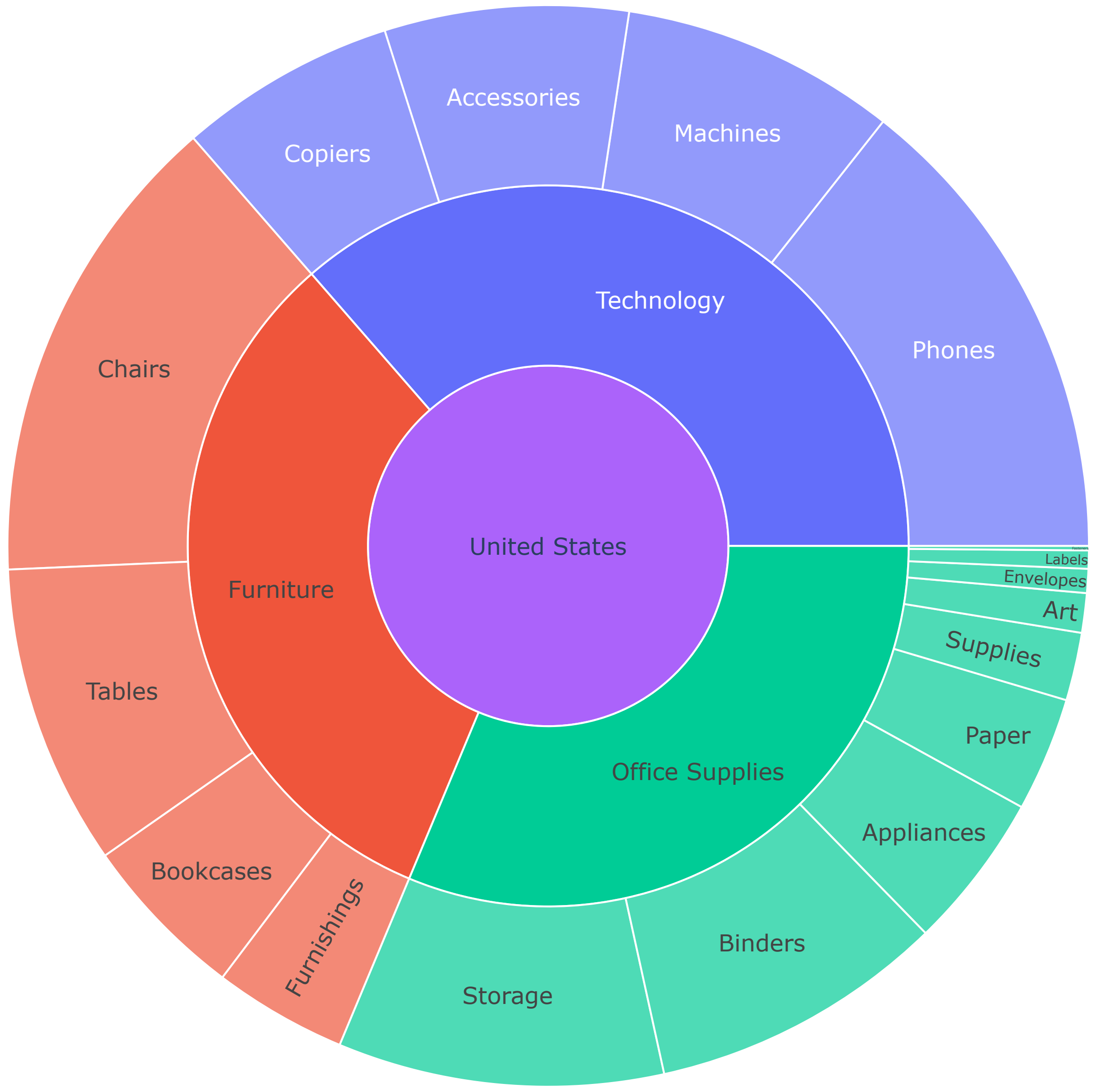
```
sns.lmplot(x='Profit',y='Sales',data=df,fit_reg=False,col="Category")
plt.show()
```



HERE WE OBSERVE THE PROFIT OR THE LOSSES WITH RESPECT TO EACH OF THE SUB CATEGORIES

we observe that table, bookcases and fasteners are in loss whereas the copiers sub category has the highest amount of profit

```
fig=px.sunburst(df,path=['Country','Category','Sub-Category'],values='Sales',color='Category',hover_data=['Ship Mode', 'Segment', 'Country', 'City', 'State', 'Postal Code', 'Region', 'Category', 'Sub-Category', 'Sales', 'Quantity', 'Discount', 'Profit'])
fig.update_layout(height=700)
fig.show()
```



THE FINAL INSIGHTS

1. When the discount is till 3.0 there is a profit.But if the discount increase beyond 0.3 there a loss will be incurred
2. Althrough copiers is the least selling sub-category but has given the most profit out of all the sub category
3. The profit more from the east and westregion of the country