Author : Shruti Jaiswal

Technical Task 3 : Exploratory Data Analysis - Retail

Dataset Link : https://bit.ly/3i4rbWl (https://bit.ly/3i4rbWl)

In this task, we will perform 'Exploratory Data Analysis' on the dataset 'SampleSuperstore'. We try to explore the data and find out the weak areas where we can work to make more profit.

In [115]:
```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from plotnine import *
import warnings
warnings.filterwarnings('ignore')
```

Step 1 : Understanding the dataset

In [116]:
```
data=pd.read_csv("C:/Users/shrut/Desktop/SampleSuperstore.csv")
```

In [117]:
```
data.head()     #listing the first five rows of the dataset
```

Out[117]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sa
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.96
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.94
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14.62
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.57
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.36

In [118]:
```
data.tail()     #listing the last five rows of the dataset
```

Out[118]:

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

In [119]:
```
data.shape       #dimensionality of the dataset
```

Out[119]:
```
(9994, 13)
```

In [120]:
```
data.describe()      #gives the statistical data
```

Out[120]:

	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 [121]:
```
data.info()      #Returns the concise summary of the dataset
```

```
<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 [122]:
```
data.columns       #list the names of all the columns in the dataset
```

Out[122]:
```
Index(['Ship Mode', 'Segment', 'Country', 'City', 'State', 'Postal Code',
       'Region', 'Category', 'Sub-Category', 'Sales', 'Quantity', 'Discount',
       'Profit'],
      dtype='object')
```

In [123]:
```
data.nunique()       #shows the number of unique values in each column
```

Out[123]:
```
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
```

In [124]:
```
data['Ship Mode'].unique()
```

Out[124]:
```
array(['Second Class', 'Standard Class', 'First Class', 'Same Day'],
      dtype=object)
```

Step 2 : Cleaning the data

In [125]:
```
data.isnull().sum()      #checks the missing values
```

Out[125]:
```
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 [126]:
```
data.duplicated().sum()      #checks the duplicated data
```

Out[126]:
```
17
```

In [127]:
```
data.drop_duplicates()
```

Out[127]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage
...	...	...	...	...	...	...	...	...	...
9989	Second Class	Consumer	United States	Miami	Florida	33180	South	Furniture	Furnishings
9990	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Furniture	Furnishings
9991	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Technology	Phones
9992	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Office Supplies	Paper
9993	Second Class	Consumer	United States	Westminster	California	92683	West	Office Supplies	Appliances

9977 rows × 13 columns

In [128]:
```
store = data.drop(['Postal Code'],axis=1)       #dropping the irrelevant column
store.head()
```

Out[128]:

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

Step 3 : Relationship analysis and Data Visualization

In [129]:
```
correlation = store.corr()       #correlation between variables
store.corr()
```

Out[129]:

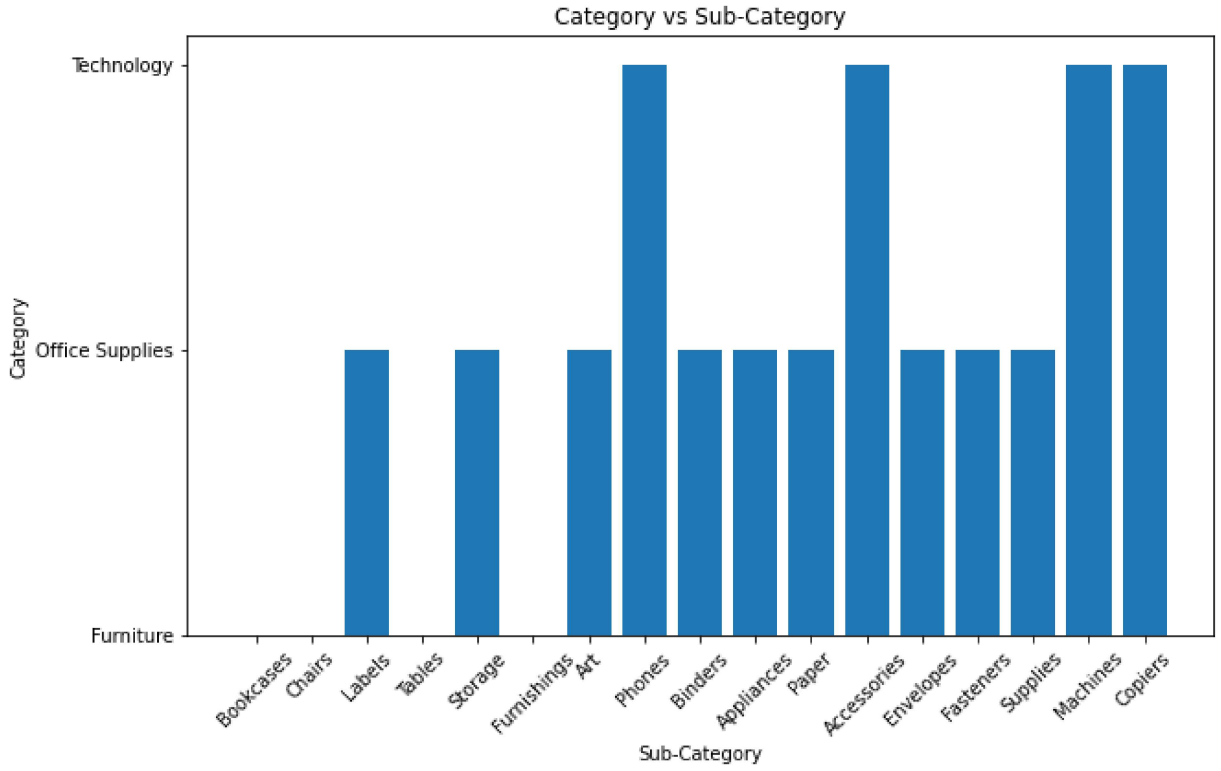
	Sales	Quantity	Discount	Profit
Sales	1.000000	0.200795	-0.028190	0.479064
Quantity	0.200795	1.000000	0.008623	0.066253
Discount	-0.028190	0.008623	1.000000	-0.219487
Profit	0.479064	0.066253	-0.219487	1.000000

In [130]:
```
store.cov()        #covariance of columns
```

Out[130]:

	Sales	Quantity	Discount	Profit
Sales	388434.455308	278.459923	-3.627228	69944.096586
Quantity	278.459923	4.951113	0.003961	34.534769
Discount	-3.627228	0.003961	0.042622	-10.615173
Profit	69944.096586	34.534769	-10.615173	54877.798055

In [131]:
```
plt.figure(figsize=(10,6))
plt.bar('Sub-Category','Category',data=store)
plt.title('Category vs Sub-Category')
plt.xlabel('Sub-Category')
plt.ylabel('Category')
plt.xticks(rotation=45)
plt.show()
```

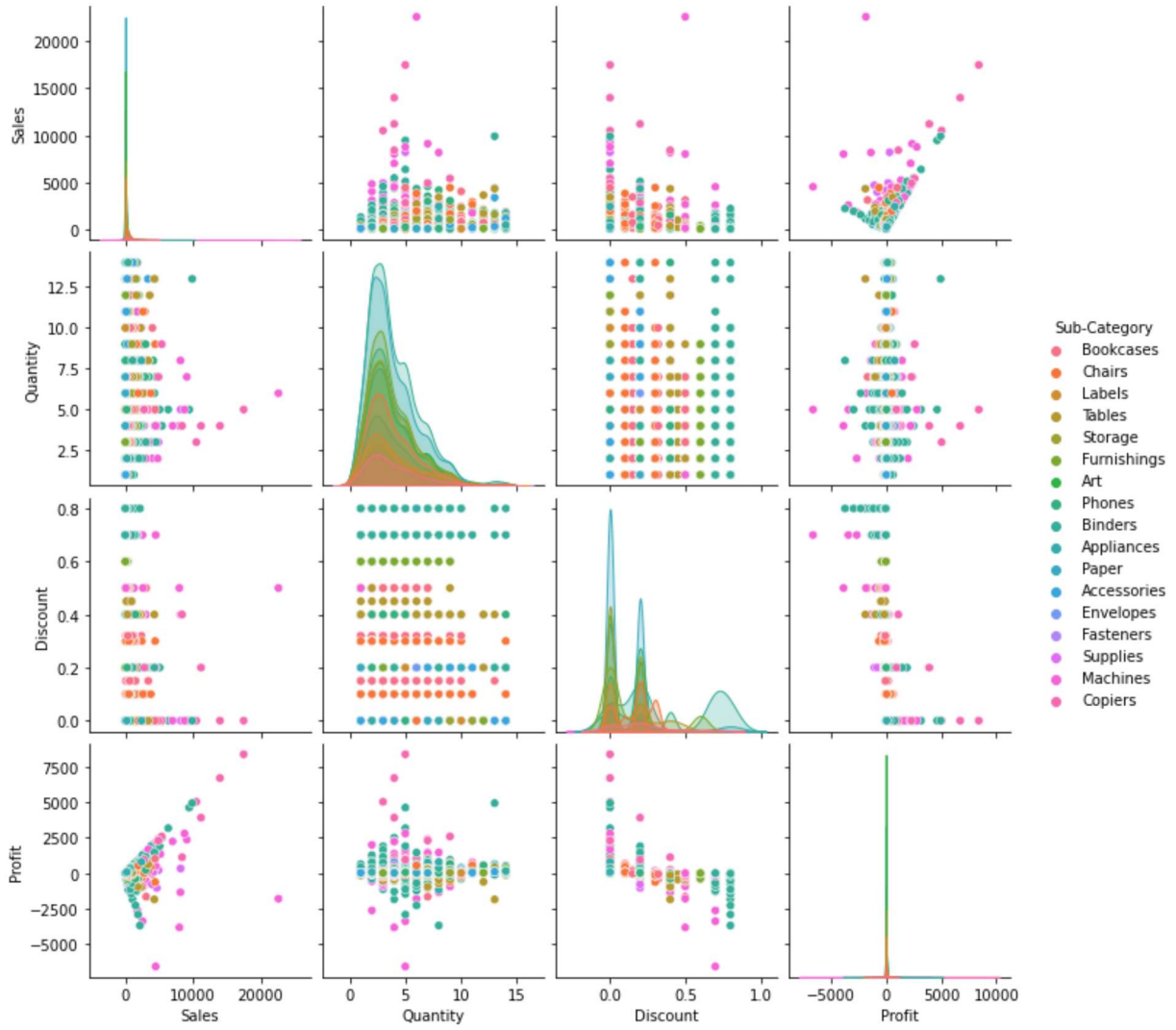


In [132]:
```
sns.heatmap(correlation, xticklabels=correlation.columns, yticklabels=correlation
```

Out[132]: <AxesSubplot:>

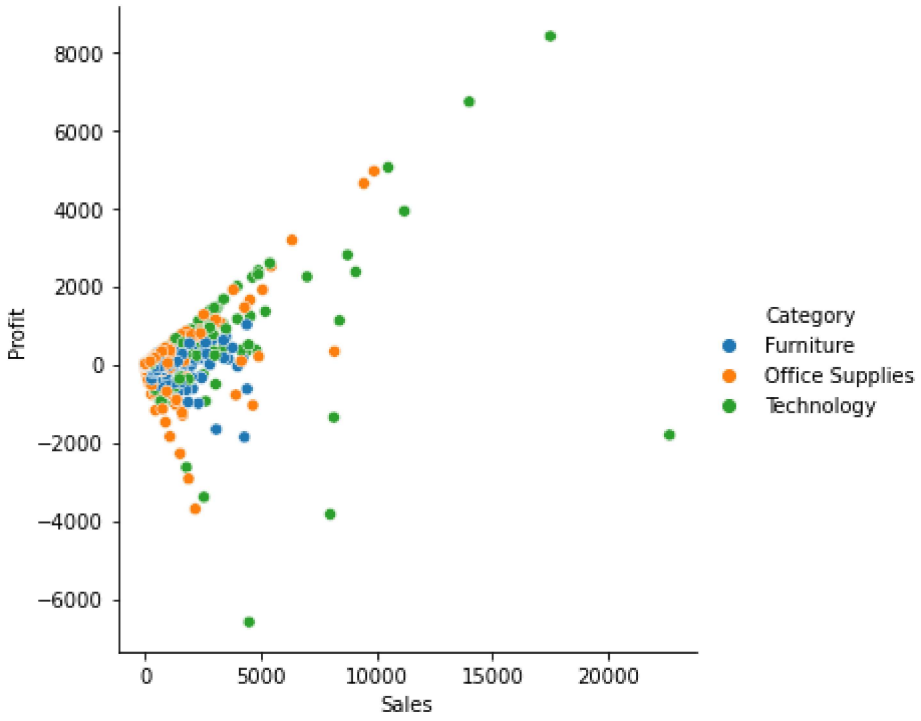
In [133]:
```
figsize=(15,10)
sns.pairplot(store, hue='Sub-Category')
```

Out[133]: <seaborn.axisgrid.PairGrid at 0x24da39068b0>

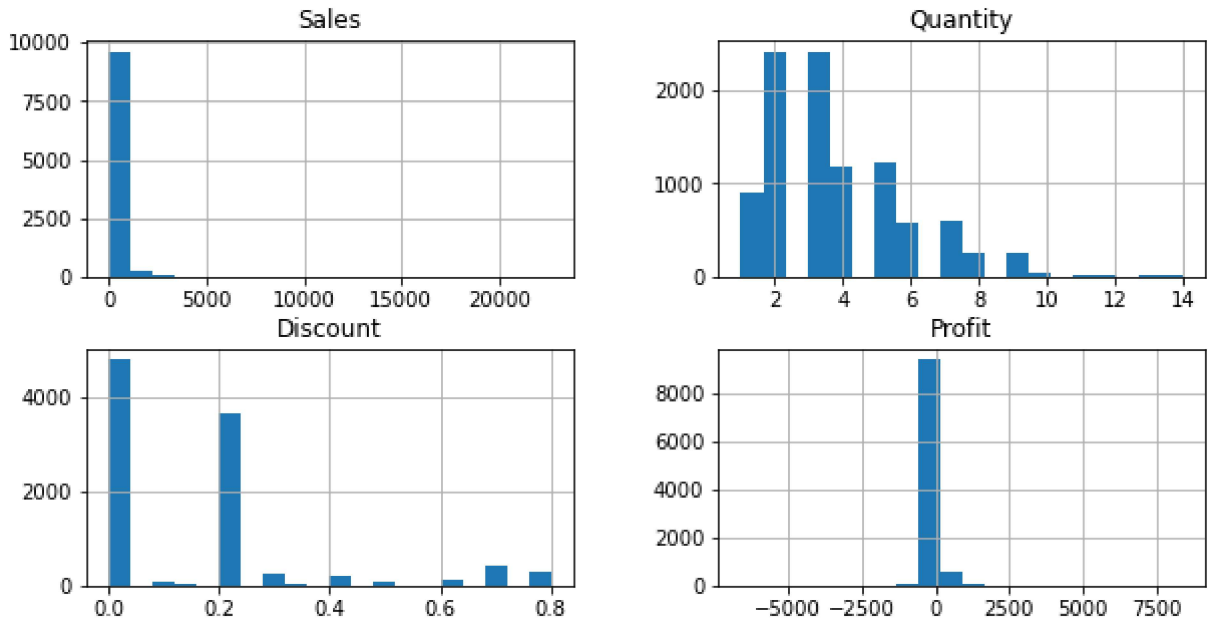


In [134]:
```
sns.relplot(x='Sales', y='Profit', hue='Category', data=store)
```

Out[134]: <seaborn.axisgrid.FacetGrid at 0x24da2fcca00>



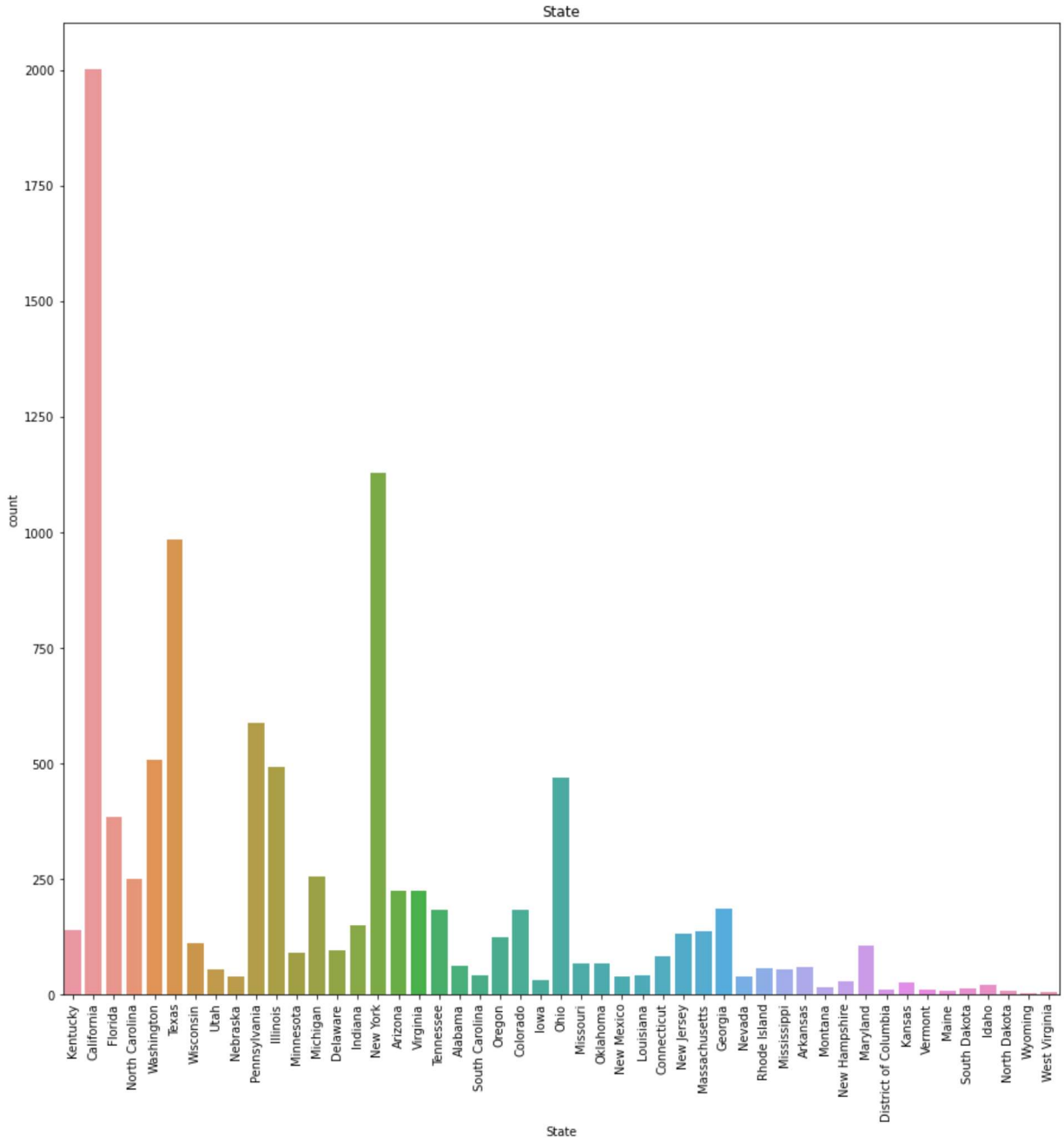
In [135]:
```
store.hist(bins=20,figsize=(10,5))          #plots a histogram
plt.show()
```



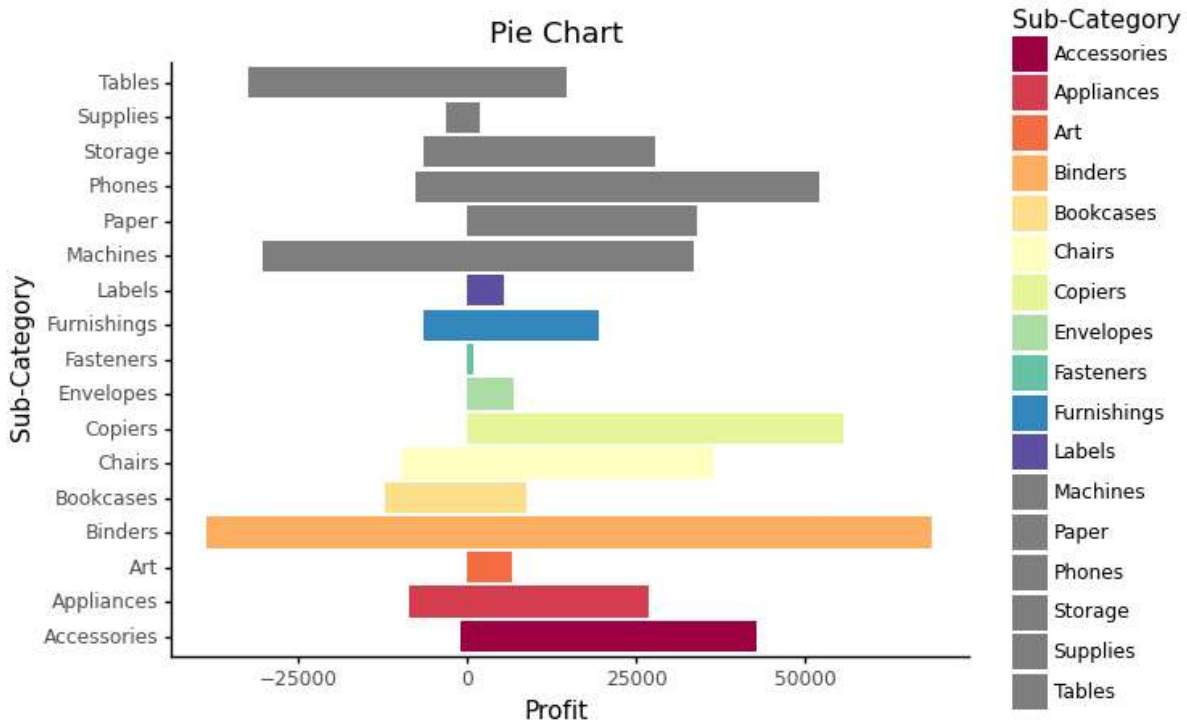
In [136]:
```
store['State'].value_counts()          #counts the total repeatable states
```

Out[136]:
```
California              2001
New York                1128
Texas                    985
Pennsylvania             587
Washington               506
Illinois                 492
Ohio                     469
Florida                  383
Michigan                 255
North Carolina           249
Arizona                  224
Virginia                 224
Georgia                  184
Tennessee                183
Colorado                 182
Indiana                  149
Kentucky                 139
Massachusetts            135
New Jersey               130
Oregon                   124
Wisconsin                110
Maryland                 105
Delaware                  96
Minnesota                 89
Connecticut               82
Missouri                  66
Oklahoma                  66
Alabama                   61
Arkansas                  60
Rhode Island              56
Utah                      53
Mississippi               53
South Carolina            42
Louisiana                 42
Nevada                    39
Nebraska                  38
New Mexico                37
Iowa                      30
New Hampshire             27
Kansas                    24
Idaho                     21
Montana                   15
South Dakota              12
Vermont                   11
District of Columbia      10
Maine                      8
North Dakota               7
West Virginia              4
Wyoming                    1
Name: State, dtype: int64
```

In [137]:
```
plt.figure(figsize=(15,15))
sns.countplot(x=store['State'])
plt.xticks(rotation=90)
plt.title('State')
plt.show()
```



In [138]:
```
Profit_plot = (ggplot(store, aes(x='Sub-Category', y='Profit', fill='Sub-Category
display(Profit_plot)
```

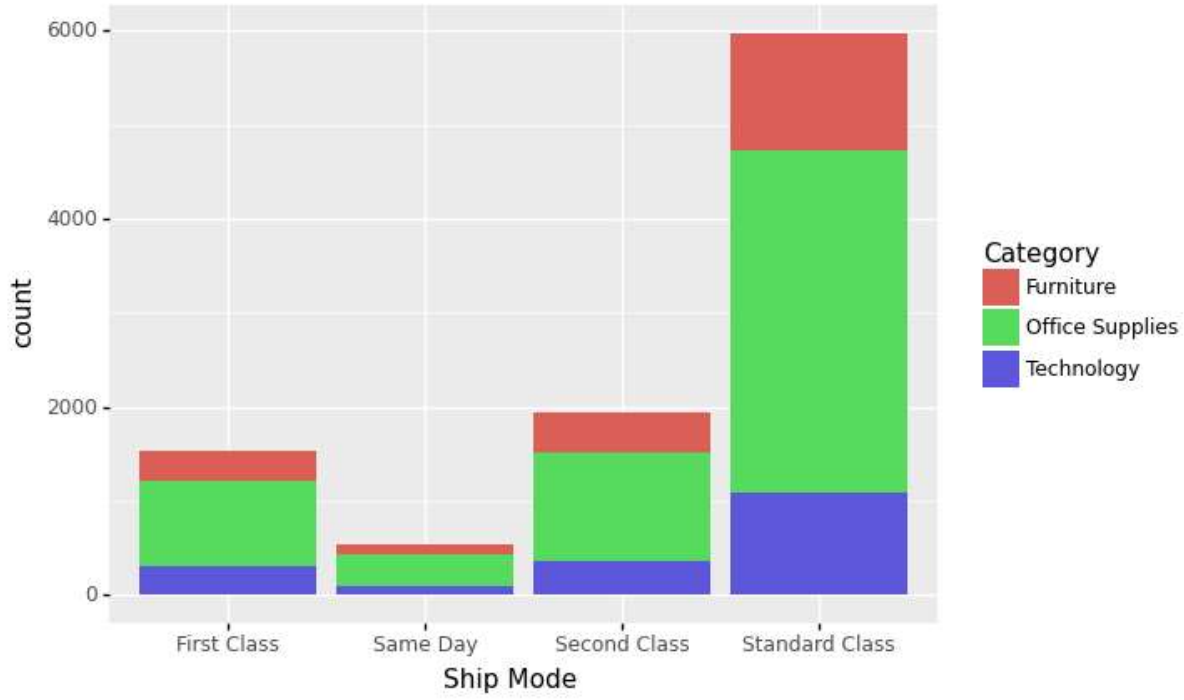



```
<ggplot: (158275119312)>
```

The above pie chart shows the profit and loss of each and every subcategories.

Now we can visualize that "binders" sub-category has suffered the maximum amount of loss and profit at the same time amongst all other categories.Next "Copiers" sub-category has suffered maximum profit with no loss.

In [139]:
```
ggplot(store, aes(x='Ship Mode', fill = 'Category')) + geom_bar(stat='count')
```

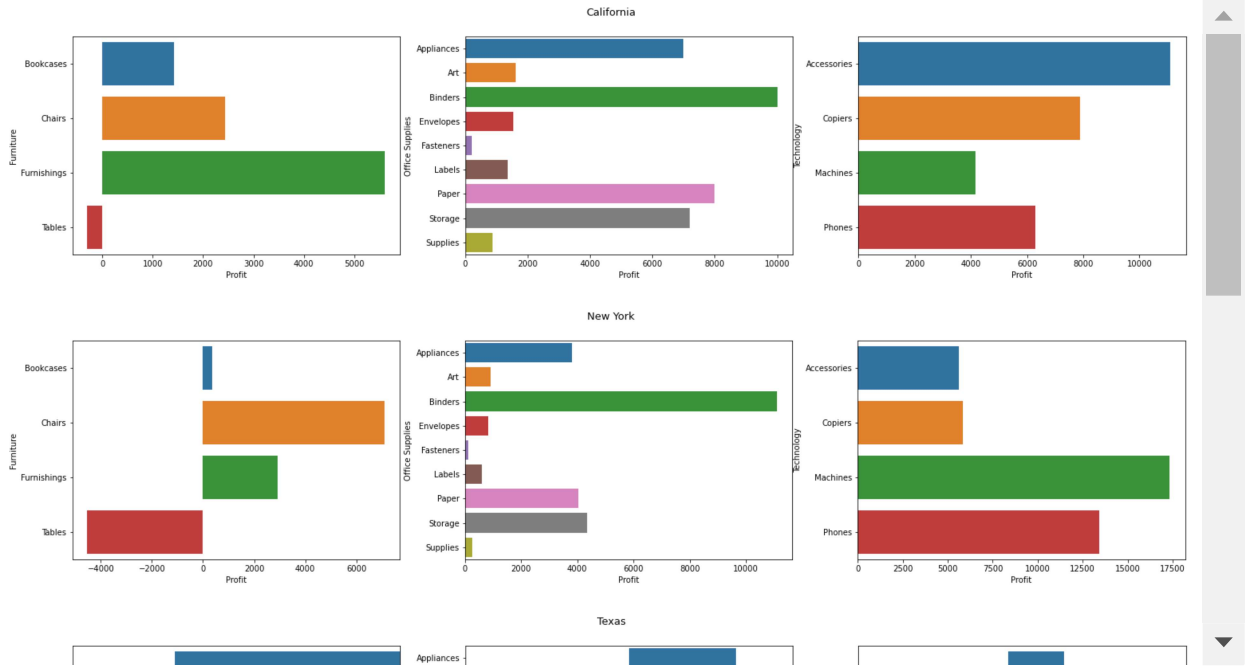


Out[139]: <ggplot: (158261885648)>

In [142]:
```
def state_data_viewer(states):          #plots the turnover generated by different
    product_data=store.groupby(['State'])

    for state in states:
        data=product_data.get_group(state).groupby(['Category'])
        fig,ax = plt.subplots(1,3,figsize=(25,5))
        fig.suptitle(state,fontsize=13)
        ax_index=0
        for cat in ['Furniture','Office Supplies', 'Technology']:
            cat_data=data.get_group(cat).groupby(['Sub-Category']).sum()
            sns.barplot(x=cat_data.Profit, y=cat_data.index, ax=ax[ax_index])
            ax[ax_index].set_ylabel(cat)
            ax_index += 1
        fig.show()
```

In [143]:
```
states = ['California', 'New York', 'Texas', 'Washington', 'Mississippi']
state_data_viewer(states)
```



**From the above data visualization, we can have a good idea of the states and the category where sales and profits are high or less. So as a business manager, one can improve in those states by providing discounts in preferred range so that both the company and the consumer will be in profit. Here, while the Superstore is incurring losses by providing discounts, so one can reduce the discounts in the region where number of consumers are high for a particular kind of category. This will enhance the profit a bit more for the company.

Conclusion:

1)When discount increases,sales increases but profit decreases.In Technology category,we get more profit as compared to other two business.This is because, less discount has been given.

2)The products must be sold with less discount in order to gain some profit.

3)For enhancing the profits, it will be better to minimize supplying Furniture and the items in other categories that result in loss.

In [ ]: