In [1]:

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

In [2]:

```
df = pd.read_csv('SampleSuperstore.csv')
df.sample(10)
```

Out[2]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
90	Standard Class	Corporate	United States	Los Angeles	California	90036	West	Technology	Phones	73.584	2	0.2	8.2782
6291	First Class	Consumer	United States	Salem	Virginia	24153	South	Furniture	Chairs	701.960	2	0.0	168.4704
9884	Standard Class	Corporate	United States	Los Angeles	California	90008	West	Technology	Accessories	62.310	3	0.0	22.4316
8347	Standard Class	Corporate	United States	Lawrence	Massachusetts	1841	East	Furniture	Furnishings	9.480	1	0.0	3.7920
7093	Standard Class	Consumer	United States	Henderson	Kentucky	42420	South	Office Supplies	Paper	51.550	5	0.0	24.2285
3172	Standard Class	Consumer	United States	Chicago	Illinois	60623	Central	Furniture	Chairs	528.430	5	0.3	0.0000
8825	First Class	Consumer	United States	Philadelphia	Pennsylvania	19134	East	Office Supplies	Labels	6.912	3	0.2	2.5056
8486	Standard Class	Home Office	United States	Fresno	California	93727	West	Office Supplies	Paper	110.960	2	0.0	53.2608
6300	Standard Class	Consumer	United States	Chicago	Illinois	60623	Central	Office Supplies	Binders	24.588	3	0.8	-38.1114
3661	Second Class	Home Office	United States	Jackson	Michigan	49201	Central	Technology	Phones	377.970	3	0.0	94.4925

In [3]:

```
df.shape
```

```
Out[3]: (9994, 13)
```

```
In [4]: 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 [5]: df.describe()
```

Out[5]:

	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 [6]:

```
x  = df[df.duplicated()]
x
```

Out[6]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
950	Standard Class	Home Office	United States	Philadelphia	Pennsylvania	19120	East	Office Supplies	Paper	15.552	3	0.2	5.4432
3406	Standard Class	Home Office	United States	Columbus	Ohio	43229	East	Furniture	Chairs	281.372	2	0.3	-12.0588
3670	Standard Class	Consumer	United States	Salem	Oregon	97301	West	Office Supplies	Paper	10.368	2	0.2	3.6288
4117	Standard Class	Consumer	United States	Los Angeles	California	90036	West	Office Supplies	Paper	19.440	3	0.0	9.3312
4553	Standard Class	Consumer	United States	San Francisco	California	94122	West	Office Supplies	Paper	12.840	3	0.0	5.7780
5905	Same Day	Home Office	United States	San Francisco	California	94122	West	Office Supplies	Labels	41.400	4	0.0	19.8720
6146	Standard Class	Corporate	United States	San Francisco	California	94122	West	Office Supplies	Art	11.760	4	0.0	3.1752
6334	Standard Class	Consumer	United States	New York City	New York	10011	East	Office Supplies	Paper	49.120	4	0.0	23.0864
6357	Standard Class	Corporate	United States	Seattle	Washington	98103	West	Office Supplies	Paper	25.920	4	0.0	12.4416
7608	Standard Class	Consumer	United States	San Francisco	California	94122	West	Office Supplies	Paper	25.920	4	0.0	12.4416
7735	Standard Class	Corporate	United States	Seattle	Washington	98105	West	Office Supplies	Paper	19.440	3	0.0	9.3312
7759	Standard Class	Corporate	United States	Houston	Texas	77041	Central	Office Supplies	Paper	15.552	3	0.2	5.4432
8032	First Class	Consumer	United States	Houston	Texas	77041	Central	Office Supplies	Paper	47.952	3	0.2	16.1838
8095	Second Class	Consumer	United States	Seattle	Washington	98115	West	Office Supplies	Paper	12.960	2	0.0	6.2208

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
9262	Standard Class	Consumer	United States	Detroit	Michigan	48227	Central	Furniture	Chairs	389.970	3	0.0	35.0973
9363	Standard Class	Home Office	United States	Seattle	Washington	98105	West	Furniture	Furnishings	22.140	3	0.0	6.4206
9477	Second Class	Corporate	United States	Chicago	Illinois	60653	Central	Office Supplies	Binders	3.564	3	0.8	-6.2370

In [7]:

```python
df.drop_duplicates(inplace=True)
```

In [8]:

```python
df.shape
```

Out[8]:

```
(9977, 13)
```

In [9]:

```python
df[df.duplicated()]
```

Out[9]:

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

In [10]:

```python
# Droping the Country Column as the dataset is only for US
df.drop('Country', axis=1 ,inplace=True)
df
```

Out[10]:

	Ship Mode	Segment	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
0	Second Class	Consumer	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	0.00	41.9136
1	Second Class	Consumer	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9400	3	0.00	219.5820
2	Second Class	Corporate	Los Angeles	California	90036	West	Office Supplies	Labels	14.6200	2	0.00	6.8714
3	Standard Class	Consumer	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5775	5	0.45	-383.0310
4	Standard Class	Consumer	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.3680	2	0.20	2.5164
...	...	...	...	...	...	...	...	...	...	...	...	...
9989	Second Class	Consumer	Miami	Florida	33180	South	Furniture	Furnishings	25.2480	3	0.20	4.1028

	Ship Mode	Segment	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
9990	Standard Class	Consumer	Costa Mesa	California	92627	West	Furniture	Furnishings	91.9600	2	0.00	15.6332
9991	Standard Class	Consumer	Costa Mesa	California	92627	West	Technology	Phones	258.5760	2	0.20	19.3932
9992	Standard Class	Consumer	Costa Mesa	California	92627	West	Office Supplies	Paper	29.6000	4	0.00	13.3200
9993	Second Class	Consumer	Westminster	California	92683	West	Office Supplies	Appliances	243.1600	2	0.00	72.9480

9977 rows × 12 columns

In [11]:

```
df.head()
```

Out[11]:

	Ship Mode	Segment	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
0	Second Class	Consumer	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	0.00	41.9136
1	Second Class	Consumer	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9400	3	0.00	219.5820
2	Second Class	Corporate	Los Angeles	California	90036	West	Office Supplies	Labels	14.6200	2	0.00	6.8714
3	Standard Class	Consumer	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5775	5	0.45	-383.0310
4	Standard Class	Consumer	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.3680	2	0.20	2.5164

In [168…

```
# Get descriptive statistics summary

df.describe(include = "all").T
```

Out[168…

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Ship Mode	9977	4	Standard Class	5955	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Segment	9977	3	Consumer	5183	NaN	NaN	NaN	NaN	NaN	NaN	NaN
City	9977	531	New York City	914	NaN	NaN	NaN	NaN	NaN	NaN	NaN
State	9977	49	California	1996	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Postal Code	9977.0	NaN	NaN	NaN	55154.964117	32058.266816	1040.0	23223.0	55901.0	90008.0	99301.0
Region	9977	4	West	3193	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Category	9977	3	Office Supplies	6012	NaN	NaN	NaN	NaN	NaN	NaN	NaN

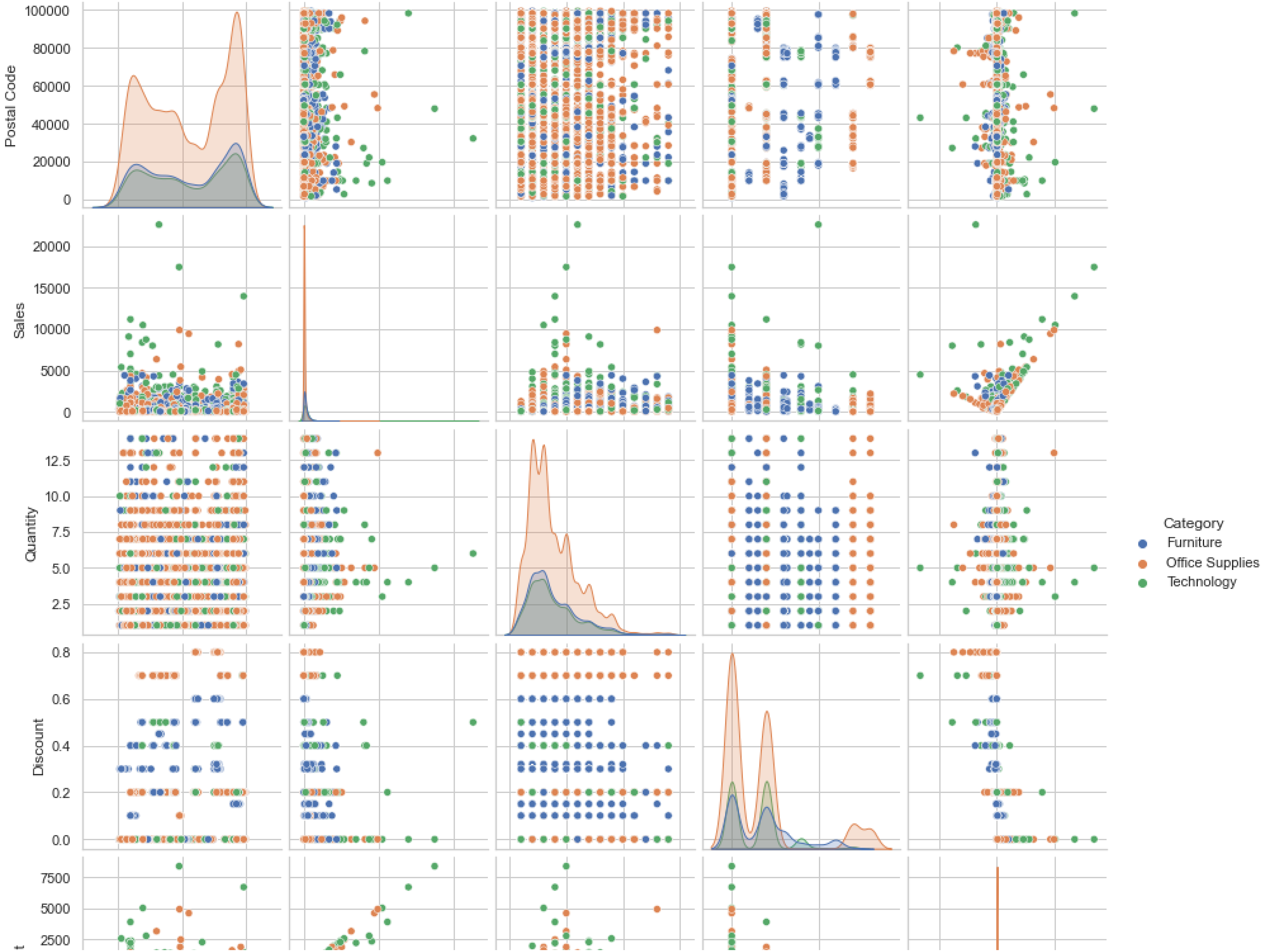
	count	unique	top	freq	mean	std	min	25%	50%	75%	max
Sub-Category	9977	17	Binders	1522	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Sales	9977.0	NaN	NaN	NaN	230.148902	623.721409	0.444	17.3	54.816	209.97	22638.48
Quantity	9977.0	NaN	NaN	NaN	3.790719	2.226657	1.0	2.0	3.0	5.0	14.0
Discount	9977.0	NaN	NaN	NaN	0.156278	0.206455	0.0	0.0	0.2	0.2	0.8
Profit	9977.0	NaN	NaN	NaN	28.69013	234.45784	-6599.978	1.7262	8.671	29.372	8399.976

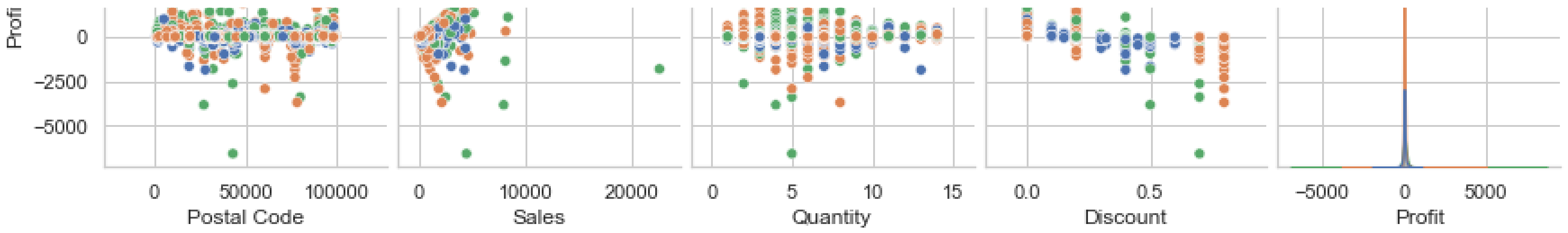
In [192…

```
sns.pairplot(df , hue='Category' , data= df)
```

Out[192…

```
<seaborn.axisgrid.PairGrid at 0x1ceda3b54c0>
```


Postal Code
100000
80000
60000
40000
20000
0
Sales
20000
15000
10000
5000
0
Quantity
12.5
10.0
7.5
5.0
2.5
Discount
0.8
0.6
0.4
0.2
0.0
t
7500
5000
2500
Category
● Furniture
● Office Supplies
● Technology

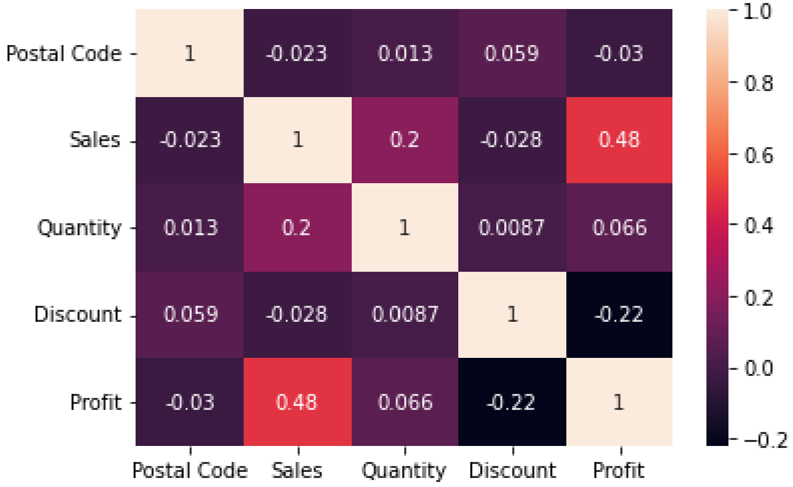


In [44]:

```
sns.heatmap(df.corr(),  annot=True)
```

Out[44]:

```
<AxesSubplot:>
```



In [128…

```
df2 = pd.DataFrame(df.groupby(['Sub-Category'])[['Sales', 'Profit']].sum())
# df2
df2_1 = df2.sort_values('Sales', ascending=False)
df2_2 = df2_1.drop('Profit' , axis=1)
df2_2
```

Out[128…

	Sales
Sub-Category	
Phones	330007.0540
Chairs	327777.7610
Storage	223843.6080

	Sales
Sub-Category	
Tables	206965.5320
Binders	203409.1690
Machines	189238.6310
Accessories	167380.3180
Copiers	149528.0300
Bookcases	114879.9963
Appliances	107532.1610
Furnishings	91683.0240
Paper	78224.1420
Supplies	46673.5380
Art	27107.0320
Envelopes	16476.4020
Labels	12444.9120
Fasteners	3024.2800

In [131…

```
df3_1 = df2.sort_values('Profit' , ascending = False)
df3_2 = df3_1.drop('Sales' , axis=1)
df3_2
```

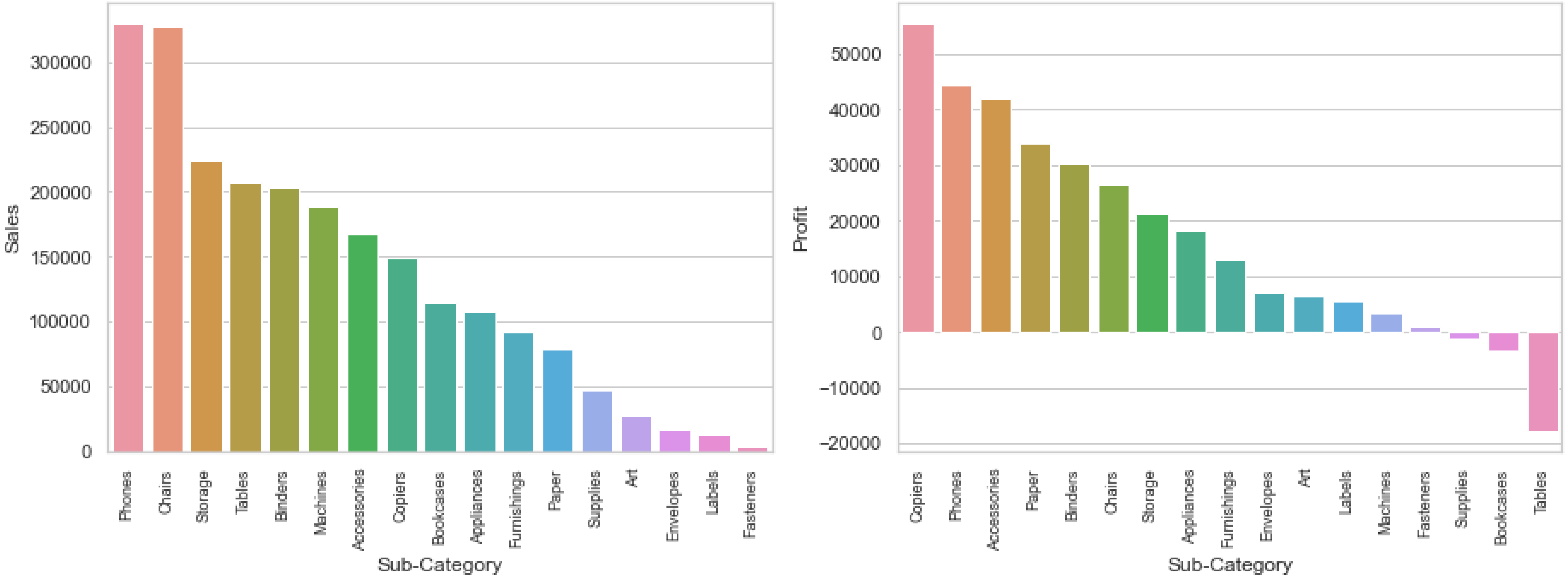
Out[131…

	Profit
Sub-Category	
Copiers	55617.8249
Phones	44515.7306
Accessories	41936.6357
Paper	33944.2395
Binders	30228.0003

	Profit
Sub-Category	
Chairs	26567.1278
Storage	21278.8264
Appliances	18138.0054
Furnishings	13052.7230
Envelopes	6964.1767
Art	6524.6118
Labels	5526.3820
Machines	3384.7569
Fasteners	949.5182
Supplies	-1189.0995
Bookcases	-3472.5560
Tables	-17725.4811

In [141…

```
sns.set_theme(style="whitegrid")
figure, axis = plt.subplots(1, 2, figsize=(13, 5))
sales_plot = sns.barplot(x=df2_2.index , y=df2_2['Sales'] ,  ax=axis[0])
profit_plot = sns.barplot(x=df3_2.index , y=df3_2['Profit'] ,  ax=axis[1])

plt.setp(sales_plot.get_xticklabels(), rotation = 'vertical', size = 9)
plt.setp(profit_plot.get_xticklabels(), rotation = 'vertical', size = 9)
figure.tight_layout()
```



Observations:

Phones and Chairs are Top 2 best selling sub-category.

Copiers produces most profit, followed by Phones, Accessories, Papers and Binders. The marketing strategy has to focus on marketing these products.

On the other end of the spectrum, Machines, Fasteners, Supplies, Bookcases and Tables make close to zero margin to losses. These are products that Super Store can consider dropping from the product catalogue or increase the sale price and profit margin or bargain for a lower price from the supplier.

In [144…

```
df4 = pd.DataFrame(df.groupby(['Sub-Category'])[['Quantity']].sum().sort_values('Quantity',ascending=False))

df4
```

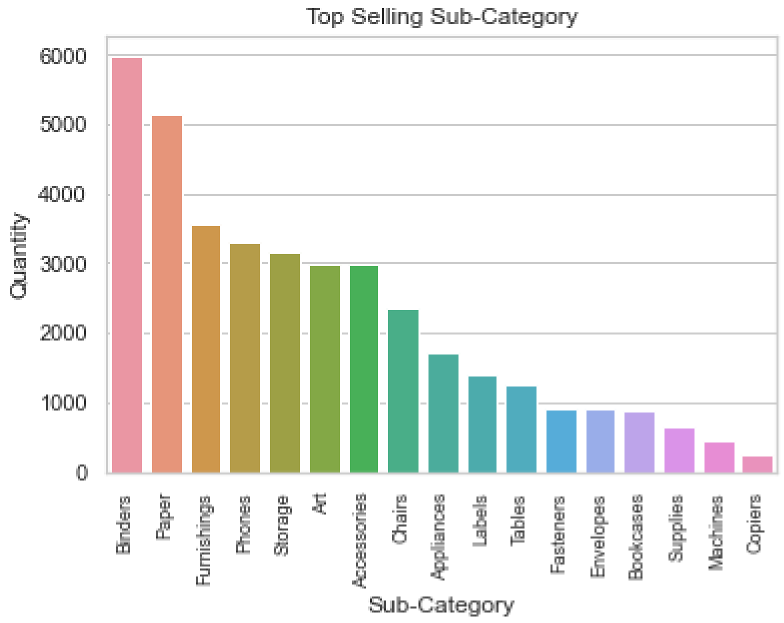
Out[144…

	Quantity
Sub-Category	
Binders	5971
Paper	5144

	Quantity
Sub-Category	
Furnishings	3560
Phones	3289
Storage	3158
Art	2996
Accessories	2976
Chairs	2351
Appliances	1729
Labels	1396
Tables	1241
Fasteners	914
Envelopes	906
Bookcases	868
Supplies	647
Machines	440
Copiers	234

In [149…

```
df4_1 = sns.barplot(x=df4.index , y=df4['Quantity'])
plt.setp(df4_1.get_xticklabels(), rotation = 'vertical', size = 9)
plt.title("Top Selling Sub-Category")
figure.tight_layout()
```



Observation:

Super Store should ensure inventory are always well-stocked for the top selling sub-category such as Binders, Paper, Furnishings and Phones.

Despite being most profitable, Copiers sell the least only 234, but as it is a relatively expensive office equipment that is usually used for few years, it is understandable that it sells the least among all.

In [188…

```
State =df.State
max_sales_city = pd.DataFrame(State.value_counts())[:10]
max_sales_city
```

Out[188…

	State
California	1996
New York	1127
Texas	983
Pennsylvania	586
Washington	502

	State
Illinois	491
Ohio	468
Florida	383
Michigan	254
North Carolina	249

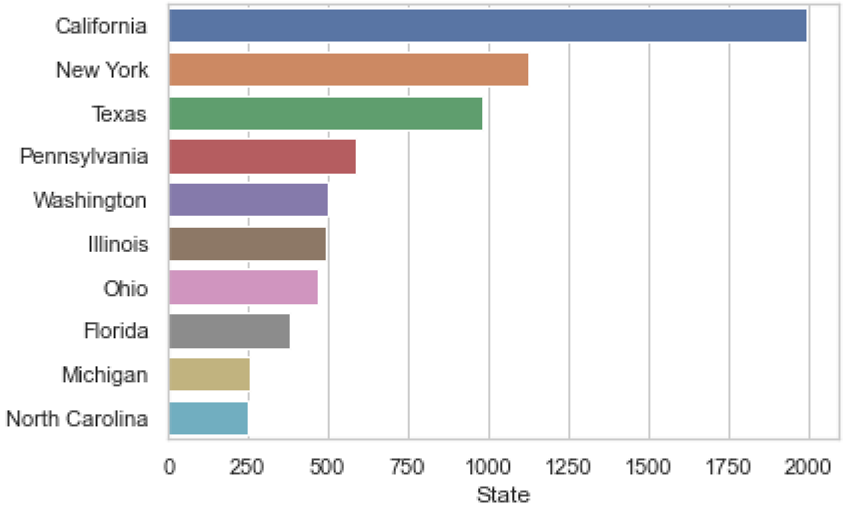
In [191…

```
sns.barplot(max_sales_city['State'] , max_sales_city.index)
```

```
c:\users\gaura\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[191…

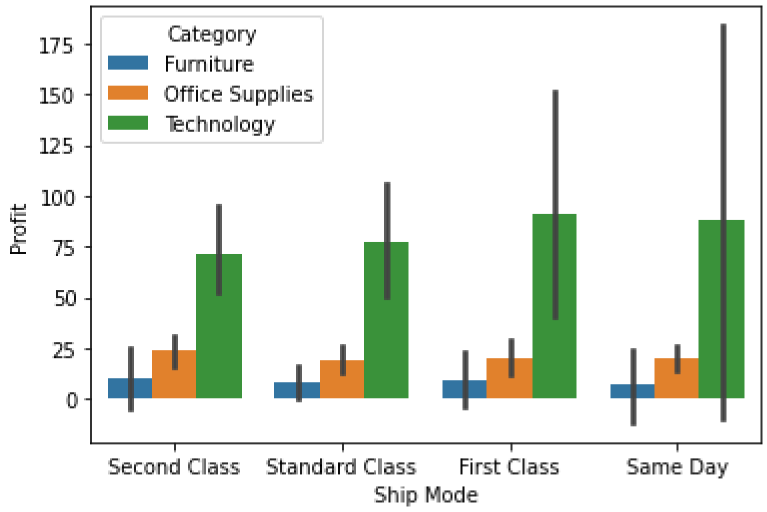
```
<AxesSubplot:xlabel='State'>
```



In [43]:

```
# shipmode sales
sns.barplot(df["Ship Mode"] , df['Profit'] , data=df , hue='Category')
```

```
c:\users\gaura\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variables as keyword args: x, y. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
Out[43]: <AxesSubplot:xlabel='Ship Mode', ylabel='Profit'>
```



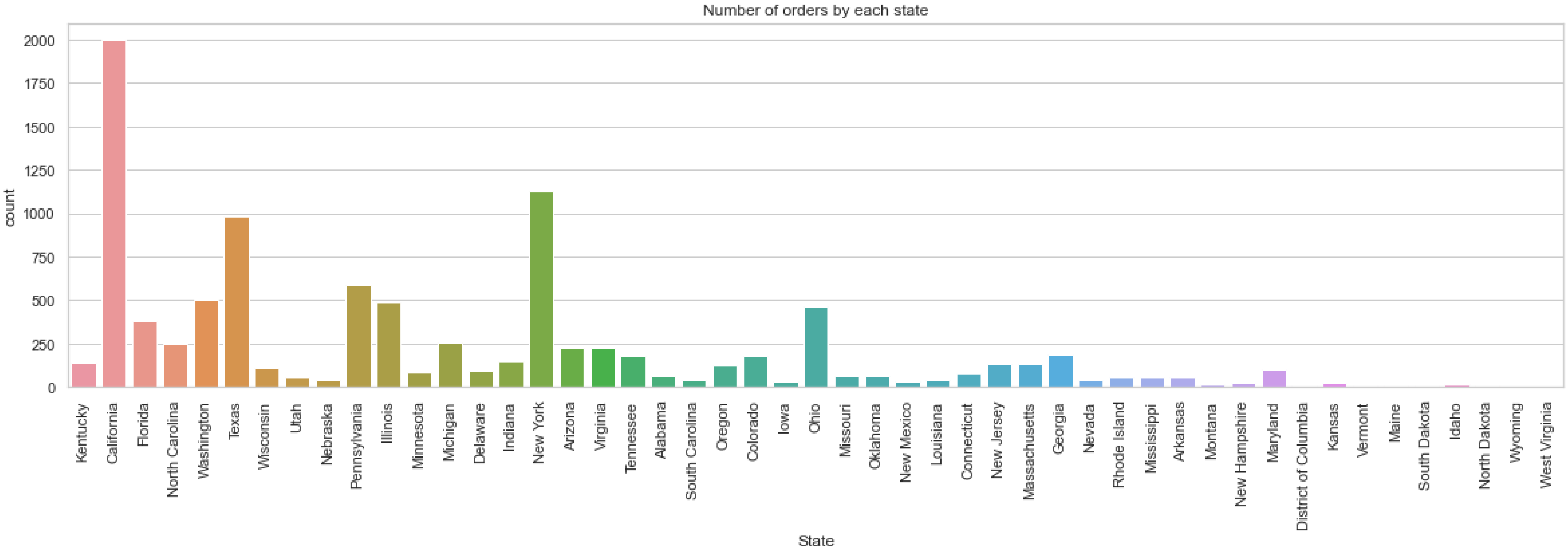
Highly Correlated Columns are

1) Quantity and Sales

2) Profit and Sales

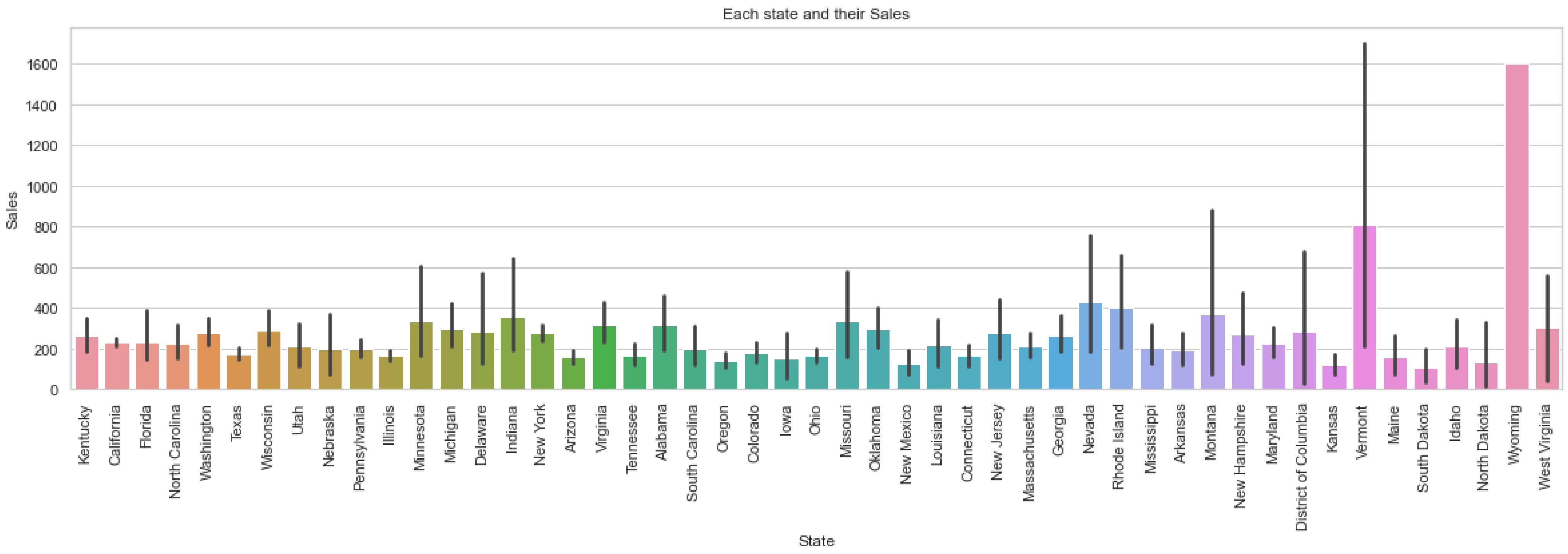
```
In [142…

plt.figure(figsize=(20,5))
plt.title("Number of orders by each state")
sns.countplot(df['State'],label="Count")
plt.xticks(rotation=90)
figure.tight_layout()
```

```
c:\users\gaura\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.1
2, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misi
nterpretation.
  warnings.warn(
```



In [143…

```
# State abd Sales

plt.figure(figsize=(20,5))
plt.title("Each state and their Sales")
sns.barplot(x = df['State'] , y = df["Sales"])
plt.xticks(rotation=90)
figure.tight_layout()
```



In [62]:

```
df1 = pd.DataFrame(df.groupby(['Category'])[['Sales', 'Profit', 'Quantity']].sum())

df1
```

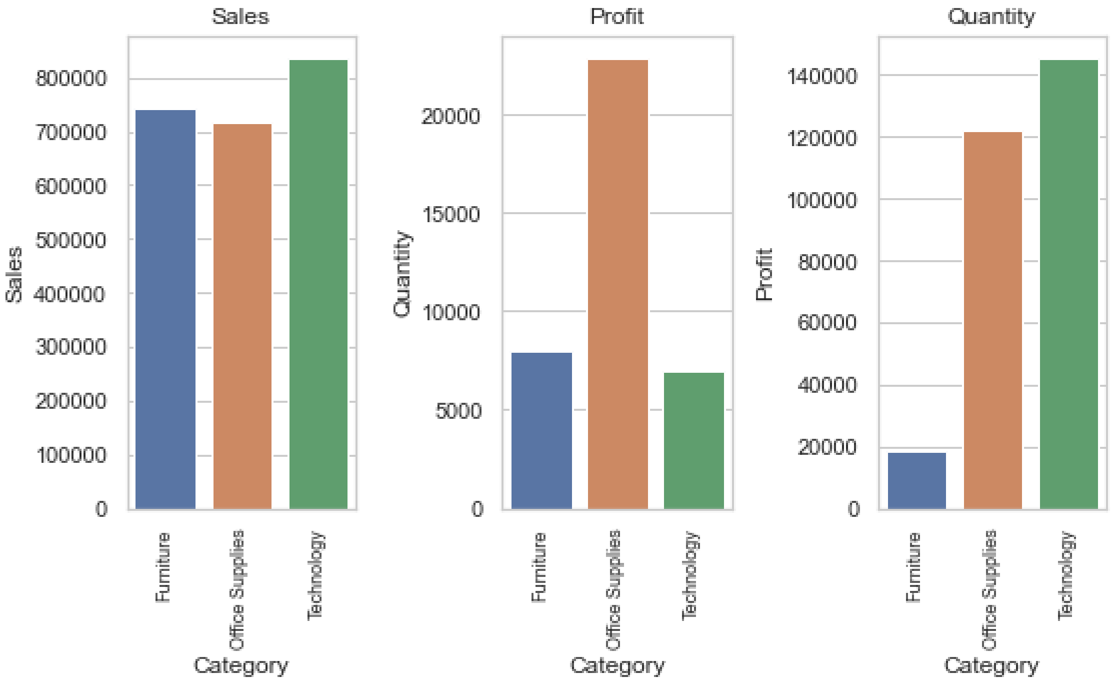
Out[62]:

	Sales	Profit	Quantity
Category			
Furniture	741306.3133	18421.8137	8020
Office Supplies	718735.2440	122364.6608	22861
Technology	836154.0330	145454.9481	6939

In [89]:

```
sns.set_theme(style="whitegrid")

figure, axis = plt.subplots(1, 3, figsize=(8, 5))
category_plot1 =sns.barplot(x=df1.index , y= df1['Sales']  , ax=axis[0])
category_plot2 =sns.barplot(x=df1.index , y= df1['Quantity']  , ax=axis[1])
category_plot3 =sns.barplot(x=df1.index, y= df1['Profit']  , ax=axis[2])
```

```
category_plot1.set(title = 'Sales')
category_plot2.set(title = 'Profit')
category_plot3.set(title = 'Quantity')
# Rotate axis for x-axis
plt.setp(category_plot1.get_xticklabels(), rotation = 'vertical', size = 9)
plt.setp(category_plot2.get_xticklabels(), rotation = 'vertical', size = 9)
plt.setp(category_plot3.get_xticklabels(), rotation = 'vertical', size = 9)
# Set spacing between subplots
figure.tight_layout()
```



observations

All 3 categories — Furniture and Office Supplies were make similar amount of sales but Technology amount of sales was way far

Technology is Best Selling and it’s good to know that this category is the Most Profitable too. Only minimal quantity is sold as these products are usually one-off purchases that can last at least 4–5 years.

Furniture is the least profitable and quantity sold are at a minimum too.

Office Supplies sells the most in terms of quantity as it is relatively cheap product.

D. Which Customer Segment is Most Profitable?

In [165…

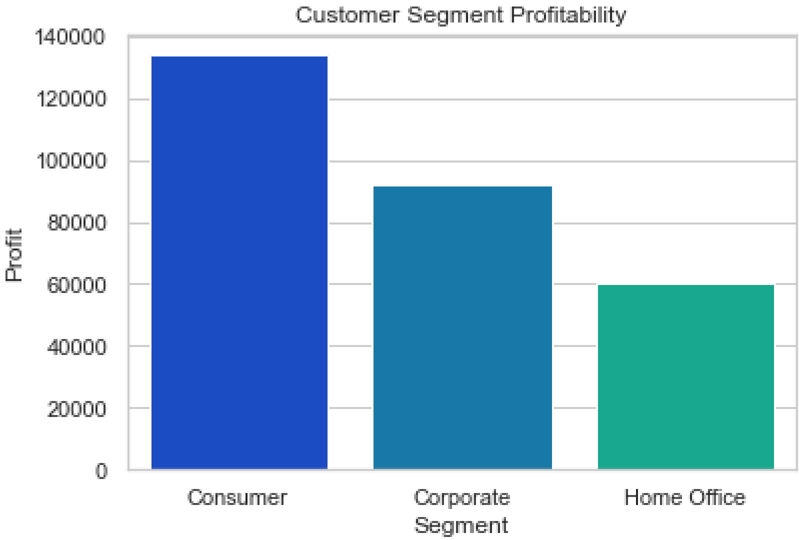
```
df4 = pd.DataFrame(df.groupby(['Segment'])[['Profit']].sum())

df4
```

Out[165…

	Profit
Segment	
Consumer	134007.4413
Corporate	91954.9798
Home Office	60279.0015

In [166…

```
sns.set_theme(style="whitegrid")
sns.barplot(data = df4, x = df4.index, y = df4.Profit, palette = "winter")
plt.title("Customer Segment Profitability")
plt.show()
```



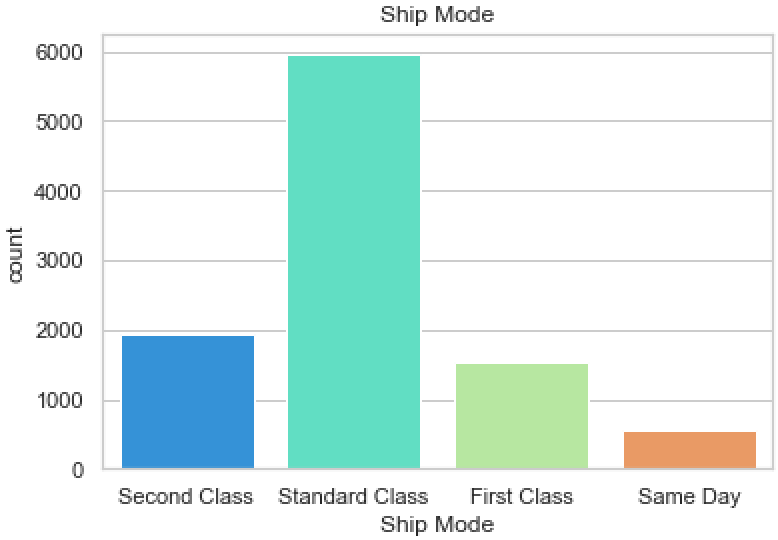
Consumer segment is most profitable, followed by Corporate Segment and Home Office. Hence, marketing strategy has to target or place more focus on retaining Consumer and Corporate Segment customers.

E. Which is the Preferred Ship Mode?

In [161…

```
# Plot shipment mode
sns.set_theme(style="whitegrid")
sns.countplot(df['Ship Mode'], palette = "rainbow")

plt.title("Ship Mode")

plt.show()
```

```
c:\users\gaura\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.1
2, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misi
nterpretation.
  warnings.warn(
```



By a landslide, Standard Class is the preferred method of shipment and perhaps the cheapest one too. The other modes are not popular among the customers and may be too costly.

F. Which Region is the Most Profitable?

In [159…

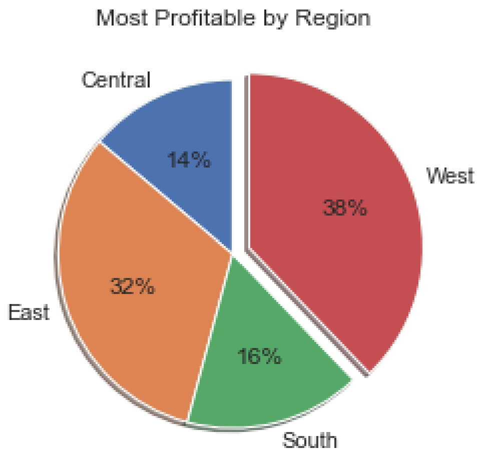
```
df5 = pd.DataFrame(df.groupby(['Region'])['Profit'].sum().reset_index())
df5
```

Out[159…

	Region	Profit
0	Central	39655.8752
1	East	91506.3092
2	South	46749.4303
3	West	108329.8079

In [160…

```
explode = [0, 0, 0, 0.1]
plt.pie(df5.Profit, labels = df5.Region, startangle = 90, autopct = "%1.0f%%", explode = explode, shadow = True)

plt.title("Most Profitable by Region")

plt.show()
```



East and West region are most profitable

G. Which City has the Highest Number of Sales?

In [153…

```
city_sales_df = pd.DataFrame(df.groupby(['City'])['Sales', 'Quantity'].sum().sort_values('Sales',ascending = False))
top10 = city_sales_df[:10]
top10
```

```
C:\Users\gaura\AppData\Local\Temp/ipykernel_12948/2369898876.py:1: FutureWarning: Indexing with multiple keys (implicitly converted to a tupl
e of keys) will be deprecated, use a list instead.
```

```
city_sales_df = pd.DataFrame(df.groupby(['City'])['Sales', 'Quantity'].sum().sort_values('Sales',ascending = False))
```

Out[153…

```
                    Sales  Quantity
        City
New York City  256319.0410     3413
  Los Angeles  175831.9010     2876
      Seattle  119460.2820     1578
San Francisco  112577.1720     1920
 Philadelphia  109061.4610     1978
      Houston   64441.2564     1460
      Chicago   48535.9770     1129
    San Diego   47521.0290      670
 Jacksonville   44713.1830      429
  Springfield   43054.3420      649
```

In [154…

```
bottom10 = city_sales_df[-10:]
bottom10
```

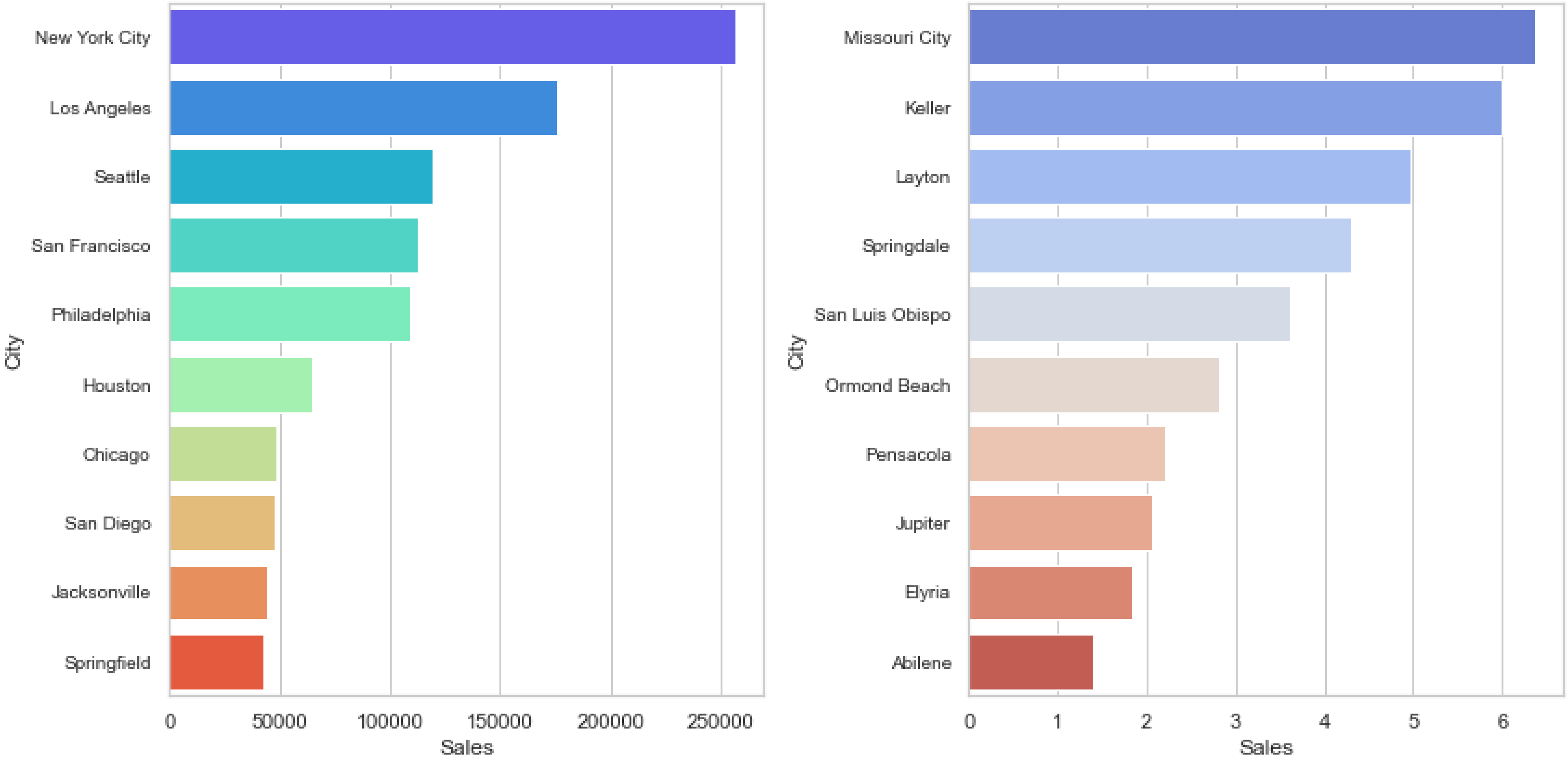
Out[154…

```
                 Sales  Quantity
           City
  Missouri City  6.370         7
         Keller  6.000         2
         Layton  4.960         4
     Springdale  4.300         2
San Luis Obispo  3.620         2
   Ormond Beach  2.808         3
      Pensacola  2.214         3
        Jupiter  2.064         1
         Elyria  1.824         1
```

	Sales	Quantity
City		
Abilene	1.392	2

In [155…

```
figure, axis = plt.subplots(1,2, figsize=(12, 6))

sns.set_theme(style="whitegrid")


top10c = sns.barplot(data = top10, y = top10.index, x = top10.Sales, palette = "rainbow", ax = axis[0])
#top10c.set(Title = "Top 10 Cities with Highest Sales")
top10c.set_yticklabels(top10c.get_yticklabels(),size = 10)

# Plot Bar Plot for Best Selling Sub-Category
bottom10c = sns.barplot(data = bottom10, y = bottom10.index, x = bottom10.Sales, palette = "coolwarm", ax=axis[1])
#bottom10c.set(Title = "Bottom 10 Cities with Lowest Sales")
bottom10c.set_yticklabels(bottom10c.get_yticklabels(),size = 10)

# Set spacing between subplots

figure.tight_layout()
plt.show()
```



Recommendations : -

Focus on Technology sub-category and Phones and Chairs as they are highest selling and most profitable. Bundle them with the less profitable products such as Bookcases, Table and Chairs to offset the losses.

Selling Bookcases and Tables result in huge losses, so Super Store has to consider to bundle them together with High Selling or Profitable sub-category such as Chairs, Copiers, Phones and Office Supplies products.

For Home Offices customers, these people might be busy with work and less likely to spend time selecting individual products, so create a Home Office package with products used for offices such as table, chairs, phone, copiers, storage, label, fasteners, bookcases.

For loss-making products like Supplies, Bookcases, Tables, consider to either drop these from the catalogue or change suppliers and bargain for cheaper price.

Consumer and Corporate Segment make up more than 70% of customerbase. Target them, especially customers from the East and West region in the Top 10 cities with Highest Sales by introducing special promotions and bundles for mass Consumer and Home Offices and send promotional emails or flyers.