SAMPLE SUPER STORE EDA

Problem Statement:

As a business manager, try to find out the weak areas where you can work to make more profit in Sample Superstore dataset using EDA.

Objective:

- Performed 'Exploratory Data Analysis' on dataset `SampleSuperstore` .
- What all business problems you can derive by exploring the data?

- **Dataset:** https://bit.ly/3i4rbWl

Followed steps:

1. Reading the data
2. Understanding data (normal routine check)
3. Data quality check and missing values
4. Checking for outliers
5. Data visualization and analysis
6. Conclusion
7. Recommendations

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
ls
```

```
 Volume in drive C is OS
 Volume Serial Number is B8CB-B37E

 Directory of C:\Users\DELL\Untitled Folder

24-06-2023  22:04    <DIR>          .
24-06-2023  22:04    <DIR>          ..
24-06-2023  18:28    <DIR>          .ipynb_checkpoints
24-06-2023  18:28         1,113,007 SampleSuperstore.csv
24-06-2023  22:04            27,450 Untitled.ipynb
               2 File(s)      1,140,457 bytes
               3 Dir(s)  113,958,772,736 bytes free
```

In [3]:
```
# reading the data
retail= pd.read_csv('SampleSuperstore.csv')
```

In [4]:
```
retail.head(5)
```

Out[4]:

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Dis
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9400	3	
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14.6200	2	
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5775	5	
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.3680	2	

In [5]:
```
# checking column names
retail.columns
```

Out[5]:
```
Index(['Ship Mode', 'Segment', 'Country', 'City', 'State', 'Postal Code',
       'Region', 'Category', 'Sub-Category', 'Sales', 'Quantity', 'Discount',
       'Profit'],
      dtype='object')
```

In [6]:
```
# checking data dimension
retail.shape
```

Out[6]:
```
(9994, 13)
```

In [7]:
```
# checking column information
retail.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
```

INFERENCE :

None of the column as any null values

In [8]:
```
# checking numerical summary of the data set
retail.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

INFERENCE

1) Average sales is only 229 Rs

2) min of average sales is 0.444000 which is generating negative profite we should stop selling this product as in long term it would impact companies revenue.

3) We can see a large difference between minimum and maximum sales as well as profit

4) we need to check if the we are too dependent on certain products sales which is generating this imbalance

DATA INSPECTION:

In [9]:
```
# checking country column values
retail.Country.unique()
```

Out[9]:
```
array(['United States'], dtype=object)
```

INFERENCE:

From this we can conclude that this data is only for one country so we can drop this column

Transformation 1

In [10]:
```
retail.drop('Country',axis=1,inplace=True)
```

In [11]:
```
retail.head(5)
```

Out[11]:

	Ship Mode	Segment	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount
0	Second Class	Consumer	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	0.00

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

even postal code isnt of much help so we can drop it too

Transformation 2

In [12]:
```
retail.drop('Postal Code',axis=1,inplace=True)
```

In [13]:
```
retail.head(5)
```

Out[13]:

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

`Inferences:`

- Data is biased towards `United States` so dropping column `Country` won't affect further analysis.
- Dropped column `Postal Code` because postal code is something which is regional so, we can draw same insights using column `Region` , `City` and State.

In [14]:
```
# cross-check data dimension
retail.shape
```

Out[14]:
```
(9994, 11)
```

In [15]:
```
retail.head()
```

Out[15]:

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

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

In [18]:
```
retail['Category'].unique()
```

Out[18]:
```
array(['Furniture', 'Office Supplies', 'Technology'], dtype=object)
```

In [19]:
```
retail['Region'].unique()
```

Out[19]:
```
array(['South', 'West', 'Central', 'East'], dtype=object)
```

In [23]:
```
retail['City'].nunique()
```

Out[23]:
```
531
```

In [24]:
```
retail['Sub-Category'].nunique()
```

Out[24]:
```
17
```

There are 531 cities accross 4 region, 3 categories & 17 products distributed accross 2 sector

Transformation 3

We need to add price column

which is sales * quantity

In [26]:
```
retail['Price']=0
```

In [29]:
```
retail['Price']= retail['Sales']*retail['Quantity']
```

In [32]:
```
retail.head(5)
```

Out[32]:

	Ship Mode	Segment	City	State	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
0	Second Class	Consumer	Henderson	Kentucky	South	Furniture	Bookcases	261.9600	2	0.00	83.8272

	Ship Mode	Segment	City	State	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
1	Second Class	Consumer	Henderson	Kentucky	South	Furniture	Chairs	731.9400	3	0.00	658.7460
2	Second Class	Corporate	Los Angeles	California	West	Office Supplies	Labels	14.6200	2	0.00	13.7428
3	Standard Class	Consumer	Fort Lauderdale	Florida	South	Furniture	Tables	957.5775	5	0.45	-1915.1550
4	Standard Class	Consumer	Fort Lauderdale	Florida	South	Office Supplies	Storage	22.3680	2	0.20	5.0328

Transformation 4 & 5

We are multiplying quantity col with discount and profit as they were discount n profit only for one item so we are doing this

In [31]:
```
retail['Profit']= retail['Profit']*retail['Quantity']
```

In [33]:
```
retail['Discount']= retail['Discount']*retail['Quantity']
```

In [34]:
```
retail.head(5)
```

Out[34]:

	Ship Mode	Segment	City	State	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
0	Second Class	Consumer	Henderson	Kentucky	South	Furniture	Bookcases	261.9600	2	0.00	83.8272
1	Second Class	Consumer	Henderson	Kentucky	South	Furniture	Chairs	731.9400	3	0.00	658.7460
2	Second Class	Corporate	Los Angeles	California	West	Office Supplies	Labels	14.6200	2	0.00	13.7428
3	Standard Class	Consumer	Fort Lauderdale	Florida	South	Furniture	Tables	957.5775	5	2.25	-1915.1550
4	Standard Class	Consumer	Fort Lauderdale	Florida	South	Office Supplies	Storage	22.3680	2	0.40	5.0328

In [ ]:
```
retail[]
```

In [36]:
```
retail['Profit'].round(1)
```

Out[36]:
```
0          83.8
1         658.7
2          13.7
3       -1915.2
4           5.0
          ...
9989       12.3
9990       31.3
9991       38.8
```

```
9992      53.3
9993     145.9
Name: Profit, Length: 9994, dtype: float64
```

Transformation 6

We are rounding off the price & profit column

In [37]:
```
retail['Profit']=retail['Profit'].round(1)
```

In [38]:
```
retail['Price']=retail['Price'].round(1)
```

Transformation 7

We no longer need the sales column so I am dropping it

In [40]:
```
retail.drop('Sales',axis=1,inplace=True)
```

In [41]:
```
retail.head()
```

Out[41]:

	Ship Mode	Segment	City	State	Region	Category	Sub-Category	Quantity	Discount	Profit	Price
0	Second Class	Consumer	Henderson	Kentucky	South	Furniture	Bookcases	2	0.00	83.8	523.9
1	Second Class	Consumer	Henderson	Kentucky	South	Furniture	Chairs	3	0.00	658.7	2195.8
2	Second Class	Corporate	Los Angeles	California	West	Office Supplies	Labels	2	0.00	13.7	29.2
3	Standard Class	Consumer	Fort Lauderdale	Florida	South	Furniture	Tables	5	2.25	-1915.2	4787.9
4	Standard Class	Consumer	Fort Lauderdale	Florida	South	Office Supplies	Storage	2	0.40	5.0	44.7

In [42]:
```
retail['Ship Mode'].nunique()
```

Out[42]:
```
4
```

In [43]:
```
retail['Ship Mode'].unique()
```

Out[43]:
```
array(['Second Class', 'Standard Class', 'First Class', 'Same Day'],
      dtype=object)
```

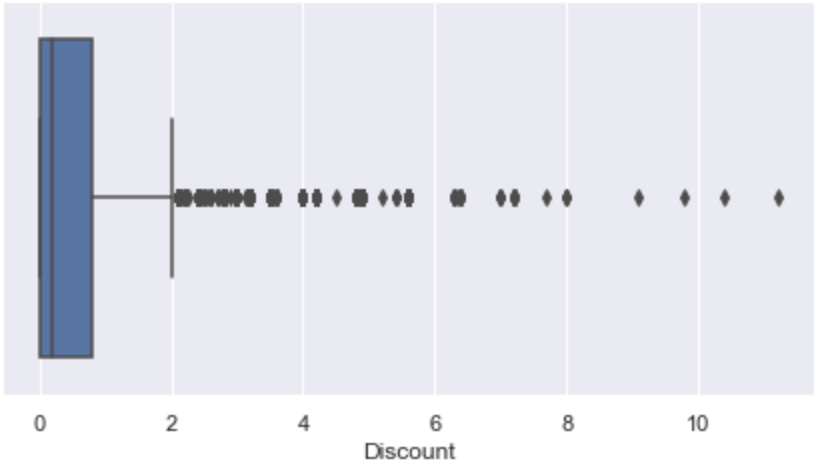
there are 4 ship mode

CHECKING OUTLINIES

In [50]:
```
# plot box plot  for numerical columns to check data outliners

plt.figure(figsize=[16,8])
```

```
sns.set_theme()
plt.subplot(2,2,1)
sns.boxplot(retail['Discount'])
plt.xlable('Discount',fontdict={'color': 'black','fontsize': 14})

plt.subplot(2,2,2)
sns.boxplot(retail['Quantity'])
plt.xlabel('Quantity', fontdict={'color':'red','fontsize':14})

plt.subplot(2,2,3)
sns.boxplot(retail['Price'])
plt.xlabel('Price', fontdict={'color':'red','fontsize':14})

plt.subplot(2,2,4)
sns.boxplot(retail['Profit'])
plt.xlabel('Profit', fontdict={'color':'red','fontsize':14})

plt.show()
```

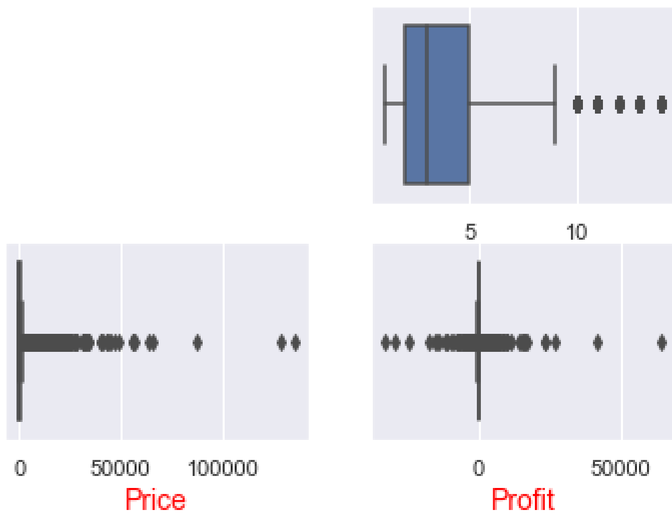
```
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass t
he following variable as a keyword arg: x. From version 0.12, the only valid positional ar
gument will be `data`, and passing other arguments without an explicit keyword will result
in an error or misinterpretation.
  warnings.warn(
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_10864/955804051.py in <module>
      5 plt.subplot(2,2,1)
      6 sns.boxplot(retail['Discount'])
----> 7 plt.xlable('Discount',fontdict={'color': 'black','fontsize': 14})
      8
      9 plt.subplot(2,2,2)

AttributeError: module 'matplotlib.pyplot' has no attribute 'xlable'
```



In [51]:

```
plt.subplot(2,2,2)
sns.boxplot(retail['Quantity'])
plt.xlabel('Quantity', fontdict={'color':'red','fontsize':14})

plt.subplot(2,2,3)
sns.boxplot(retail['Price'])
plt.xlabel('Price', fontdict={'color':'red','fontsize':14})

plt.subplot(2,2,4)
sns.boxplot(retail['Profit'])
plt.xlabel('Profit', fontdict={'color':'red','fontsize':14})

plt.show()
```

```
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass t
he following variable as a keyword arg: x. From version 0.12, the only valid positional ar
gument will be `data`, and passing other arguments without an explicit keyword will result
in an error or misinterpretation.
  warnings.warn(
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass t
he following variable as a keyword arg: x. From version 0.12, the only valid positional ar
gument will be `data`, and passing other arguments without an explicit keyword will result
in an error or misinterpretation.
  warnings.warn(
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass t
he following variable as a keyword arg: x. From version 0.12, the only valid positional ar
gument will be `data`, and passing other arguments without an explicit keyword will result
in an error or misinterpretation.
  warnings.warn(
```



INFERENCE

There is too much variation seen in data

1) The spread of **Price & Discount is right tailed**

2) **we can observe lot of items are generating revenue in negative**

In [52]:

```
# checking numerical summary of the data set

retail.describe()
```

Out[52]:

	Quantity	Discount	Profit	Price
count	9994.000000	9994.000000	9994.000000	9994.000000
mean	3.789574	0.595903	143.128957	1149.495898
std	2.225110	0.988741	1388.956107	3898.666296
min	1.000000	0.000000	-32999.900000	0.400000
25%	2.000000	0.000000	3.200000	48.700000
50%	3.000000	0.200000	26.400000	183.700000
75%	5.000000	0.800000	114.700000	763.175000
max	14.000000	11.200000	64302.800000	135830.900000

INFERENCE

1) we can see that we are selling some products on negative profits we need to stop selling them .

DATA VISULIZATION

In [53]:

```
plt.figure(figsize=[15,12], dpi=60, facecolor='w', edgecolor='k')
sns.set_theme()

plt.subplot(3,2,1)
retail['Ship Mode'].value_counts(normalize=True).plot.barh()
plt.xlabel('Ship Mode', fontdict={'color':'red', 'size':14})

plt.subplot(3,2,2)
retail['Segment'].value_counts(normalize=True).plot.barh()
plt.xlabel('Segment', fontdict={'color':'red', 'size':14})

plt.subplot(3,2,3)
retail['Region'].value_counts(normalize=True).plot.barh()
plt.xlabel('Region', fontdict={'color':'red', 'size':14})

plt.subplot(3,2,4)
retail['Category'].value_counts(normalize=True).plot.barh()
plt.xlabel('Category', fontdict={'color':'red', 'size':14})

plt.subplot(3,2,5)
retail['Sub-Category'].value_counts(normalize=True).plot.barh()
plt.xlabel('Sub-Category', fontdict={'color':'red', 'size':14})

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

Inferences:

1. Ship Mode: `Standard Class` is having higher percentage while `Same Day` is having lower percentage of transaction.

2. Segment: `Consumer` are more in numbers than `Corporate` and `Home Office` .

3. Region: `East` & `West` region is having large distribution.

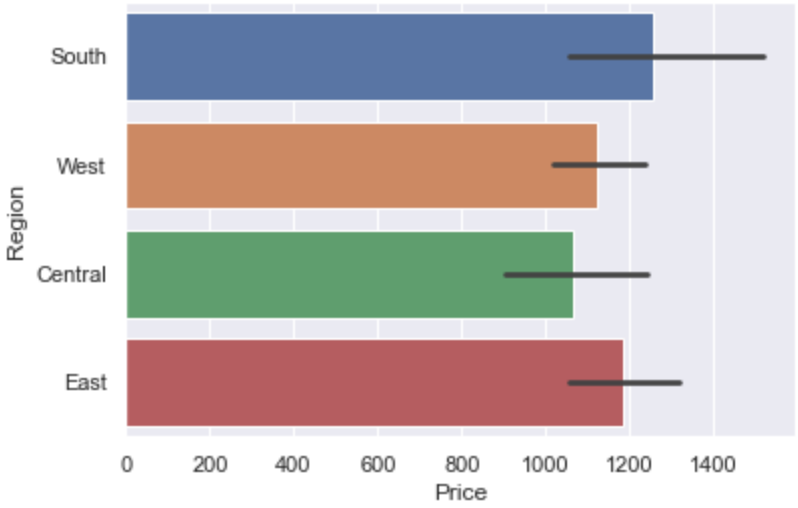
4. Category: `Office Supplies` are larger in proportion.

5. Sub-Category: `Paper` and `Binders` are having major distribution/transctions and copiers is having lesser transactions.

In [57]:
```
sns.barplot('Price','Region', data=retail)
```

```
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass t
he 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 resu
lt in an error or misinterpretation.
  warnings.warn(
```

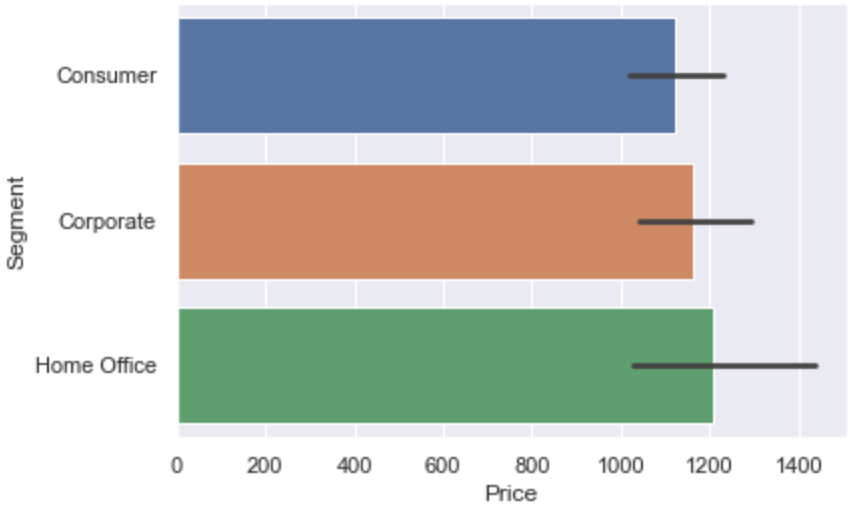
Out[57]:
```
<AxesSubplot:xlabel='Price', ylabel='Region'>
```



In [58]:
```
sns.barplot('Price','Segment', data=retail)
```

```
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass t
he 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 resu
lt in an error or misinterpretation.
  warnings.warn(
```

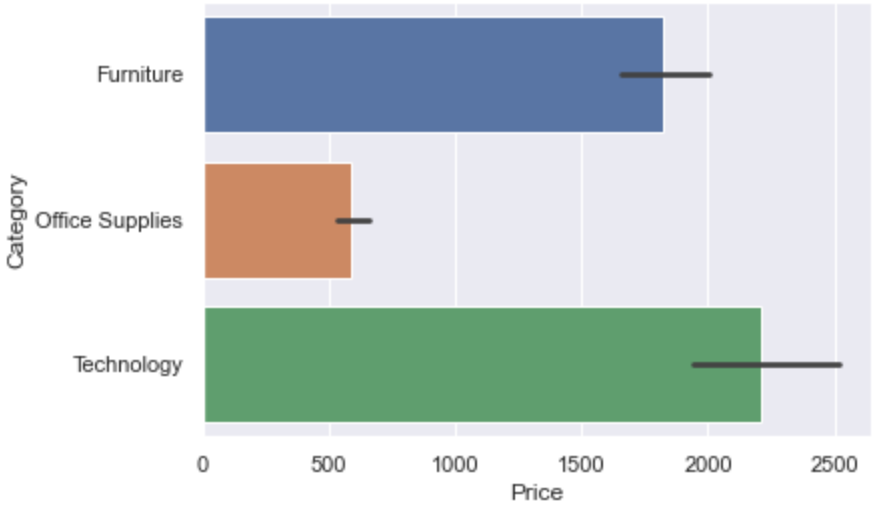
Out[58]:
```
<AxesSubplot:xlabel='Price', ylabel='Segment'>
```



In [59]:
```
sns.barplot('Price','Category', data=retail)
```

```
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass t
he 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 resu
lt in an error or misinterpretation.
  warnings.warn(
```

Out[59]:
```
<AxesSubplot:xlabel='Price', ylabel='Category'>
```



INFERENCE:

from above three graphs we can see that

1) Home office is generating higest revenue

2) east region is generating higest revenue where as central region is generating lowest revenue

3) technology sectory is generating higest revenue

In [60]:
```
sns.heatmap(retail.corr(),annot=True)
```

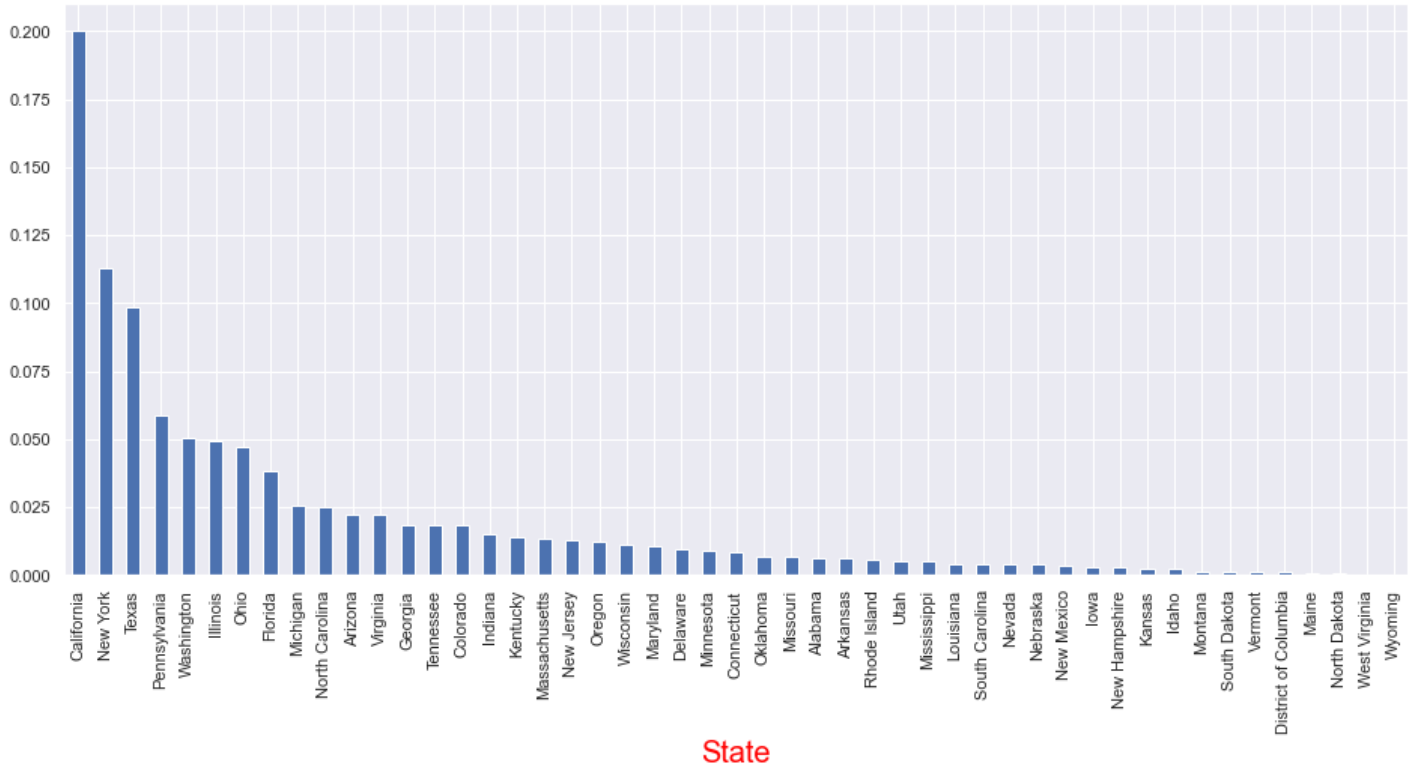
Out[60]:
```
<AxesSubplot:>
```

INFERENCE

1) We can see Discount and Profit are negatively correleated ie more the discount less will the profit we get.

In [61]:
```
# plot the bar graph of percentage distribution of various States

plt.figure(figsize=[16,7])
sns.set_theme()

retail['State'].value_counts(normalize=True).plot.bar()
plt.xlabel('State', fontdict={'color':'red', 'size':20})
```

```
plt.show()
```

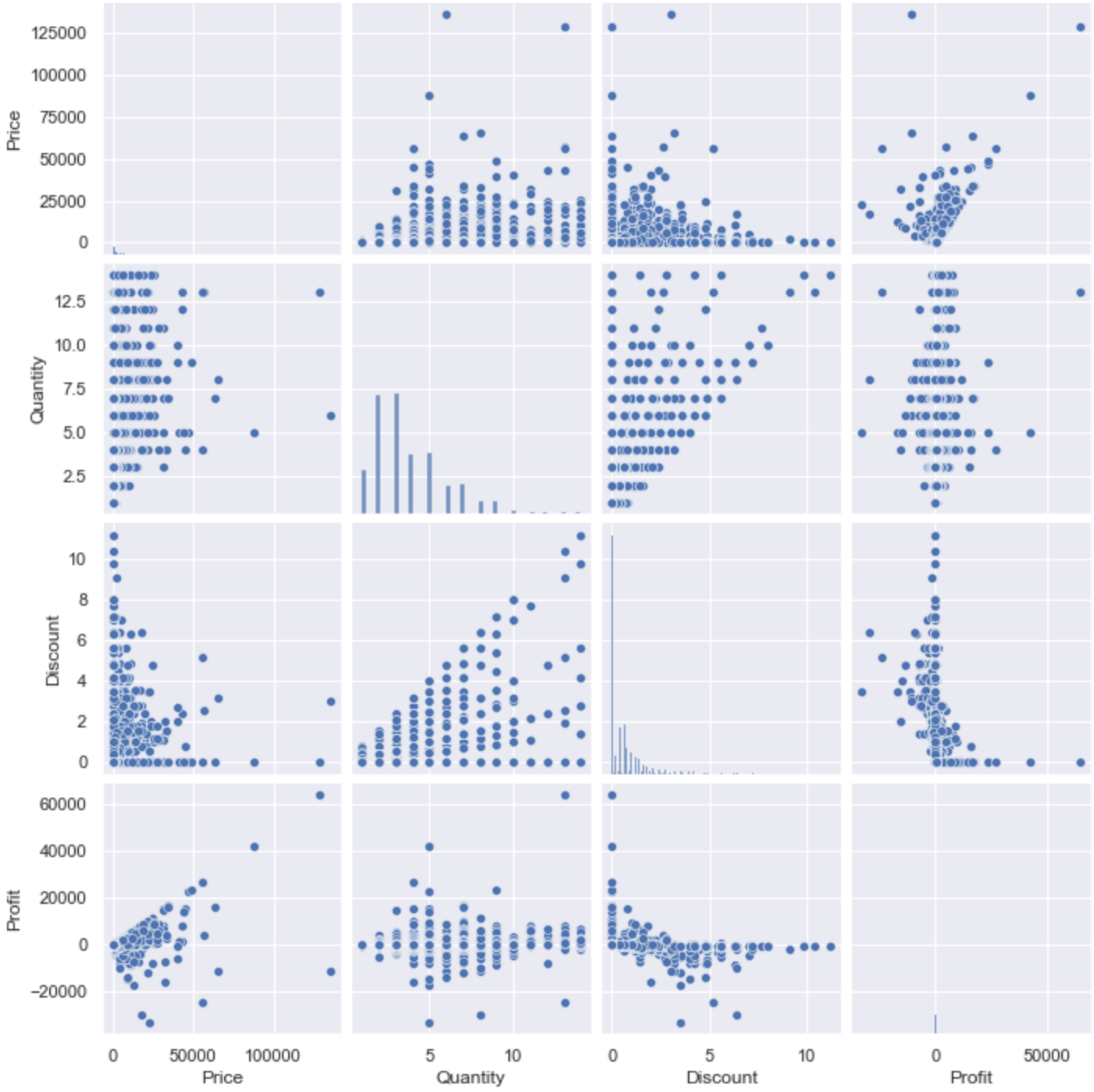


**`Inference:

`** `California` is having higher transaction whereas `New York` and `Texas` is moderate one .

we can stop delevering supplies in Vermont, columbia,maine,north dakota, west virginia and wyoming in order to reduce our expenditure on transportation.

In [63]:
```
# plot the pair plot of Sales, Quantity, Discount and Profit in retail dataframe.
sns.pairplot(retail[['Price','Quantity','Discount','Profit']])
plt.show()
```



Inference:

- We cannot see any patterns for `Quantity` over `Profit` and `Sales`. But we have highest `Sales` and `Profit` at `Quantity` equal to 5.
- `Discount` and `Profit` is highly correlated in negative direction i.e. these are inversely propotional so, if discount for a particular sale increases then this will decrease the profit.
- We can also observe that highest `Sales` are at 50% discount which seems to be an outlier.

In [64]:

```
# trend of profit across various discount
plt.figure(figsize=(10,5))
plt.title('Discount vs Profit\n', fontdict={'color':'red','size':15})
sns.lineplot(x='Discount',y='Profit', data=retail , color='r')
plt.show()
```

Inference:

- For minimum discount, Profit was good but as discount increases, profit goes down.
- Here we can see discount above 23% leads to loss.
- We can observe at `50% discount` , there is the lowest profit or highest loss.

In [66]:
```
# trend of Sales across various discount
plt.figure(figsize=(10,5))
plt.title('Discount vs Price\n', fontdict={'color':'red','size':15})
sns.lineplot(x='Discount',y='Price', data=retail , color='r')
plt.show()
```

Inference:

- We can see up and down trends for sales over discount. There peak sale is at `50% discount` and beyond `50% discount` , sales decreases gradually, might be customer doubted product quality having discount more than 50% -60%.

In [67]:
```
# groupby the Ship Mode to find the mean of the profit and sales seperatly.

retail.groupby('Ship Mode')['Profit','Price'].mean()
```

```
C:\Users\DELL\AppData\Local\Temp/ipykernel_10864/566549940.py:3: FutureWarning: Indexing w
ith multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list
instead.
  retail.groupby('Ship Mode')['Profit','Price'].mean()
```

Out[67]:

Ship Mode	Profit	Price
First Class	149.418270	1056.320286
Same Day	139.192081	1082.384715
Second Class	136.724010	1222.195938
Standard Class	143.953753	1155.920811

In [68]:
```
# groupby the Segment to find the mean of the profit and sales respec.

retail.groupby('Segment')['Profit','Price'].mean().round(2)
```

```
C:\Users\DELL\AppData\Local\Temp/ipykernel_10864/752954476.py:3: FutureWarning: Indexing w
ith multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list
instead.
  retail.groupby('Segment')['Profit','Price'].mean().round(2)
```

Out[68]:

Segment	Profit	Price
Consumer	128.80	1121.05
Corporate	155.45	1164.27
Home Office	163.96	1207.30

In [69]:
```
# groupby the Segment to find the mean of the profit and sales separately.

retail.groupby('Segment')['Profit','Price'].sum()
```

```
C:\Users\DELL\AppData\Local\Temp/ipykernel_10864/126016541.py:3: FutureWarning: Indexing w
ith multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list
instead.
  retail.groupby('Segment')['Profit','Price'].sum()
```

Out[69]:

Segment	Profit	Price
Consumer	668625.1	5819345.9
Corporate	469458.1	3516108.7
Home Office	292347.6	2152607.4

In [71]:
```
# plot the bar plot of profit for various Segments.

plt.figure(figsize=[15,5])
sns.set_theme()
plt.subplot(1,2,1)
sns.barplot(retail['Segment'], retail.Profit)
plt.title('Segment vs Avg Profit\n', fontdict={'color':'red','size':15})
plt.xlabel('Segment', fontdict={'color':'red', 'size':14})
plt.ylabel('Profit\n', fontdict={'color':'red', 'size':14})

# plot the bar plot of sales for various Segments.

plt.subplot(1,2,2)
sns.barplot(retail['Segment'], retail.Price)
plt.title('Segment vs Avg Price \n', fontdict={'color':'red','size':15})
plt.xlabel('Segment', fontdict={'color':'red', 'size':14})
plt.ylabel('Price', fontdict={'color':'red', 'size':14})

plt.show()
```

```
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass t
he 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 resu
lt in an error or misinterpretation.
  warnings.warn(
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass t
he 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 resu
lt in an error or misinterpretation.
  warnings.warn(
```

Inferences:

- `Home Office` segment is making highest average profit as well as average sales.
- `Consumer` segment is having less average profit as well as sales.

In [72]:
```
# groupby the State to find the mean of the profit and sales respeec.
retail.groupby('State')['Profit','Price'].mean()
```

```
C:\Users\DELL\AppData\Local\Temp/ipykernel_10864/3578171154.py:2: FutureWarning: Indexing
with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a lis
t instead.
  retail.groupby('State')['Profit','Price'].mean()
```

Out[72]:

	Profit	Price

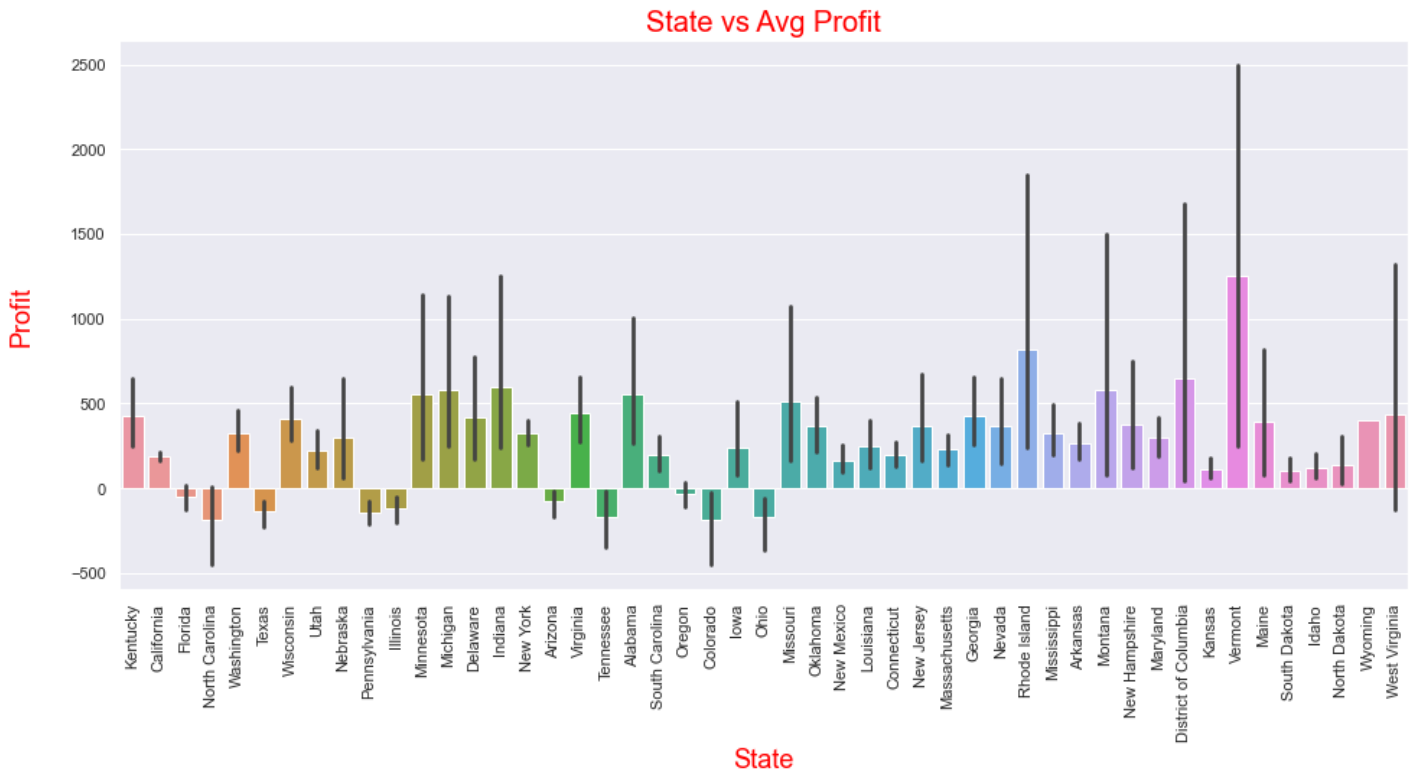
State	Profit	Price
State		
Alabama	554.619672	1941.391803
Arizona	-79.023214	758.941071
Arkansas	269.395000	831.591667
California	189.389805	1150.033733
Colorado	-185.824176	799.954396
Connecticut	195.156098	718.874390
Delaware	418.327083	1204.781250
District of Columbia	651.930000	1572.460000
Florida	-45.622193	1232.613577
Georgia	426.980435	1316.663043
Idaho	122.928571	669.309524
Illinois	-117.311179	742.041260
Indiana	597.891275	1792.200000
Iowa	236.310000	875.190000
Kansas	114.366667	373.091667
Kentucky	423.801439	1418.709353
Louisiana	247.811905	904.940476
Maine	393.125000	953.012500
Maryland	297.176190	1040.224762
Massachusetts	228.137778	972.105926
Michigan	576.274118	1609.221176
Minnesota	553.473034	1516.614607
Mississippi	323.967925	1118.437736
Missouri	516.057576	1697.648485
Montana	576.360000	1609.993333
Nebraska	297.757895	1050.673684
Nevada	364.587179	2226.474359
New Hampshire	372.918519	1560.762963
New Jersey	367.900000	1407.316154
New Mexico	166.645946	652.037838
New York	325.372872	1383.930496
North Carolina	-183.682329	1135.656627
North Dakota	133.471429	526.814286
Ohio	-171.756290	772.991258
Oklahoma	365.737879	1486.507576

State	Profit	Price
Oregon	-34.434677	637.169355
Pennsylvania	-141.121295	1025.767632
Rhode Island	816.066071	2311.333929
South Carolina	193.750000	1014.926190
South Dakota	101.708333	314.433333
Tennessee	-172.756831	866.967213
Texas	-137.103249	846.790254
Utah	223.384906	971.141509
Vermont	1251.054545	4965.990909
Virginia	442.341964	1680.595089
Washington	327.673320	1368.299802
West Virginia	433.150000	1506.800000
Wisconsin	414.309091	1559.476364
Wyoming	400.800000	6412.500000

In [73]:

```
# plot the bar plot of Profit across various States

plt.figure(figsize=[16,7])
sns.set_theme()
sns.barplot(retail['State'], retail.Profit)
plt.title('State vs Avg Profit', fontdict={'color':'red','size':20})
plt.xticks(rotation=90)
plt.xlabel('State', fontdict={'color':'red', 'size':18})
plt.ylabel('Profit\n', fontdict={'color':'red', 'size':18})
plt.show()
```

```
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass t
he 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 resu
lt in an error or misinterpretation.
  warnings.warn(
```



Inferences:

- `Vermont` state is making highest average profit whereas `California` , `New Mexico` , `Kansas` , `South Dakota` , `North Dakota` are some of the state which are having very less profit.
- `Florida` , `North Carolina` , `Texas` , `Pennsylvania` , `Illinois` , `Arizona` , `Tennessee` , `Oregon` , `Colorado` , `Ohio` are the states that are in loss which is represented as negative profit in the above bar graph.

Inferences:

- There is no change in overall profit for states like `Florida` , `North Carolina` , `Texas` , `Pennsylvania` , `Illinois` , `Arizona` , `Tennessee` , `Oregon` , `Colorado` , `Ohio` making loss i.e. negative profit. We need to focus on such states.
- We saw average profit of `Vermont` state was highest than all the other states but the overall profit is less, same for the states `Wyoming` , `West Virginia` etc. Reason behind this might be highest average profit in these states but less total transactions. So, we need to focus on overall sales of such states.
- **Region vs Profit and Region vs Sales**

In [75]:
```
#groupby the Region to find the mean of the profit and Sales respectively.

retail.groupby('Region')['Profit','Price'].mean()
```

```
C:\Users\DELL\AppData\Local\Temp/ipykernel_10864/1809954981.py:3: FutureWarning: Indexing
with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a lis
t instead.
  retail.groupby('Region')['Profit','Price'].mean()
```

Out[75]:

Region	Profit	Price
Central	94.536031	1065.791520

Region	Profit	Price
East	158.197683	1186.560709
South	140.397593	1257.824938
West	166.354199	1122.456135

- **Category vs Profit and Category vs Sales**

In [77]:
```
#groupby the Category to find the mean of the profit and sales.

retail.groupby('Category')['Profit','Price'].mean()
```

```
C:\Users\DELL\AppData\Local\Temp/ipykernel_10864/3912693404.py:3: FutureWarning: Indexing
with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a lis
t instead.
  retail.groupby('Category')['Profit','Price'].mean()
```

Out[77]:

Category	Profit	Price
Furniture	51.669401	1819.525083
Office Supplies	101.103883	588.878510
Technology	385.266919	2209.132323

- **Sub-Category vs Profit and Sub-Category vs Sales**

In [79]:
```
# groupby the Sub-Category to find the mean of the profit and sales respectively

retail.groupby('Sub-Category')['Profit','Price','Discount'].mean()
```

```
C:\Users\DELL\AppData\Local\Temp/ipykernel_10864/2227223841.py:3: FutureWarning: Indexing
with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a lis
t instead.
  retail.groupby('Sub-Category')['Profit','Price','Discount'].mean()
```

Out[79]:

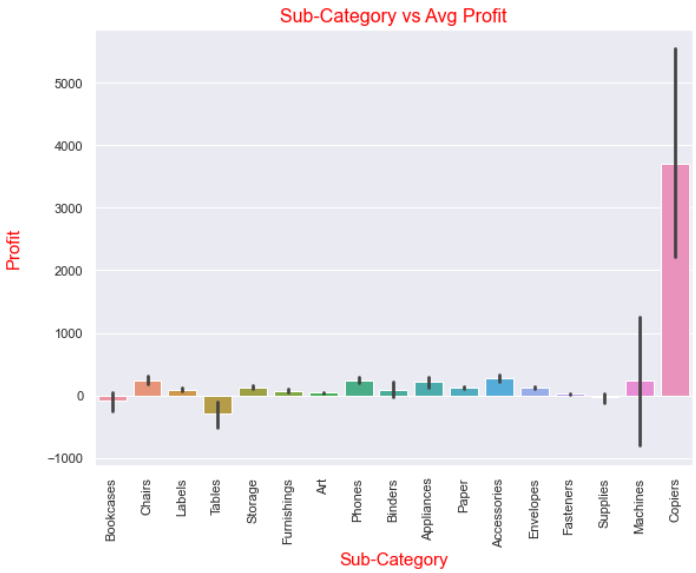
Sub-Category	Profit	Price	Discount
Accessories	276.813419	1134.096645	0.294452
Appliances	209.814592	1174.609657	0.594635
Art	41.092462	170.880653	0.275377
Binders	90.241497	665.433618	1.483848
Bookcases	-91.330702	2622.041228	0.856140
Chairs	245.487520	2694.904538	0.637439
Copiers	3699.220588	9672.380882	0.544118
Envelopes	121.724803	292.045276	0.281890
Fasteners	23.789862	78.603687	0.377880
Furnishings	75.195925	474.521212	0.501985

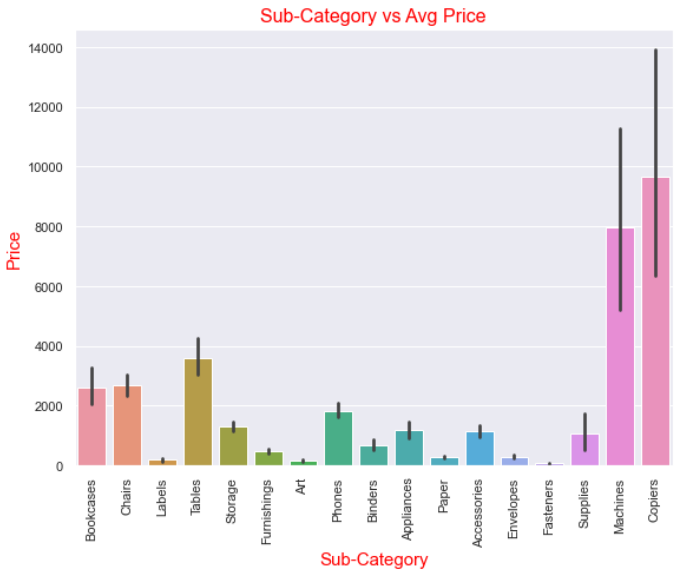
Sub-Category	Profit	Price	Discount
Labels	86.167857	194.051923	0.250000
Machines	230.998261	7954.696522	1.083478
Paper	122.633066	284.234964	0.284526
Phones	246.283240	1832.205287	0.585152
Storage	131.134752	1299.324350	0.263357
Supplies	-26.641579	1059.135263	0.260000
Tables	-291.580251	3587.819436	0.983542

In [80]:

```
# plot the bar plot of Profit for different product Sub-Category
plt.figure(figsize=[20,15])

plt.subplot(2,2,1)
sns.barplot(retail['Sub-Category'], retail.Profit)
plt.title('Sub-Category vs Avg Profit', fontdict={'color':'red','size':16})
plt.xlabel('Sub-Category', fontdict={'color':'red', 'size':15})
plt.ylabel('Profit\n', fontdict={'color':'red', 'size':15})
plt.xticks(rotation=90)

# plot the bar plot of Sales for different product Sub-Category

plt.subplot(2,2,2)
sns.barplot(retail['Sub-Category'], retail.Price)
plt.title('Sub-Category vs Avg Price', fontdict={'color':'red','size':16})
plt.xlabel('Sub-Category', fontdict={'color':'red', 'size':15})
plt.ylabel('Price', fontdict={'color':'red', 'size':15})
plt.xticks(rotation=90)

plt.show()
```

```
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass t
he 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 resu
lt in an error or misinterpretation.
  warnings.warn(
C:\Users\DELL\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass t
he 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 resu
lt in an error or misinterpretation.
  warnings.warn(
```





Inferences:

- `Copiers` are making highest profit as well as sales so market is doing good in this product sub-category.
- `Bookcases` , `Labels` , `Tables` , `Furnishings` , `Art` , `Binders` , `Fastners` , `Supplies` are certain product categories which are either having very less average profit or completely in loss.
- Whereas `Labels` , `Art` , `Fasteners` , `Paper` are the certain categories that are having less profit and sales.

- **City wise Profit exploration**

As per the problem statement we need to find the areas making less profit. So, as we analyzed, Sub-Categories `Bookcases` , `Tables` , `Machine` , `Labels` , `Art` , `Fasteners` , `Paper` , `Supplies` , `Envelopes` are either making less profit or loss. Lets explore the data city wise in detail to know which cities are making loss or no profit for these sub-categories.

Exploring Sub-Category vs Cities where it is sold

In [81]:
```
# import widget library
from ipywidgets import interact, interactive, fixed, interact_manual,widgets
```

In [82]:
```
Category=['Bookcases', 'Tables', 'Machines', 'Labels', 'Art', 'Fasteners', 'Paper', 'Suppl

# create widgets

type_list = ['Both','Profit','Loss']
target_List = ['Profit','Discount','Sales']
hue_list = ['None','Segment','Ship Mode']

Sub_category = widgets.Dropdown(
    options=Category,
    layout={'width': 'max-content'},
    description='Sub-Category',
    disabled=False
);
type = widgets.Dropdown(
    options=type_list,
    layout={'width': 'max-content'},
    description='Profit Type',
    disabled=False
);
```

```
target = widgets.Dropdown(
    options=target_List,
    layout={'width': 'max-content'},
    description='Target Vars',
    disabled=False
);
hue = widgets.RadioButtons(
    options=hue_list,
    value='None',
    layout={'width': 'max-content'},
    description='Hue Variable:',
    disabled=False
)

layout = widgets.HBox([Sub_category,target,type,hue])

# define function based on multiple filters

def cityplot():
    hue_var = None
    n=100
    if type.value == 'Loss':
        Y = retail[(retail['Sub-Category']==Sub_category.value)].groupby(by='City')['Profi
        Y = Y[Y['Profit']<0].sort_values(by='Profit')['City'].reset_index(drop = True)
        data = retail[(retail['Sub-Category']==Sub_category.value) & (retail['City'].isin
        n=len(Y)
        if hue.value == 'None':
            hue_var = None
        else:
            hue_var = retail[(retail['Sub-Category']==Sub_category.value) & (retail['City'
    elif type.value == 'Profit':
        Y = retail[(retail['Sub-Category']==Sub_category.value)].groupby(by='City')['Profi
        Y = Y[Y['Profit']>=0].sort_values(by='Profit')['City'].reset_index(drop = True)
        data = retail[(retail['Sub-Category']==Sub_category.value) & (retail['City'].isin
        n=len(Y)
        if hue.value == 'None':
            hue_var = None
        else:
            hue_var = retail[(retail['Sub-Category']==Sub_category.value) & (retail['City'
    else:
        data = retail[retail['Sub-Category']==Sub_category.value]
        if hue.value == 'None':
            hue_var = None
        else:
            hue_var = retail[(retail['Sub-Category']==Sub_category.value)][hue.value]


    if target.value == 'Discount':
        est = np.mean
    else:
        est = sum
    if len(data[target.value]) >0 :
        plt.figure(figsize=(20,n))
        sns.barplot(y='City', x=target.value,hue = hue_var,estimator=est,data=data)
        plt.title(f'For Sub-Category {Sub_category.value} City VS {target.value}\n', fonto
        plt.xlabel(f'{target.value}', fontdict={'color':'red', 'size':15})
        plt.ylabel(f'City\n', fontdict={'color':'red', 'size':15})
        plt.xticks(rotation=90)
        plt.show()
    else:
        print(f'There is no loss for {Sub_category.value}')
display(layout)

im=widgets.interact_manual(cityplot)
im.widget.children[0].description = 'Show Plot'
display(im)
```

```
<function __main__.cityplot()>
```

Inferences: :

- If we visualize city-wise profit for Sub-Category `Bookcases` . For cities like `Philadelphia` , `Houston` , `Colorado Spring` there is high loss so we need to focus in these cities. Along with if we see cities like `Burlington` , `San Francisco` are making more profit so we can increase stock in these cities rather than cities which are making loss.
- If we visualize city-wise discount for Sub-Category `Bookcases` , most of the cities are providing high discount. This might be one reason for loss or less profit.
- If we visualize city-wise profit for Sub-Category `Tables` , cities like `Settle` are making highest profit even at 0 discount. Hence, we can increase stock in this area for business profit.