Group06.Project01.Final

June 10, 2024

1 Project-Superstore Sales Analysis and Performance

Domain: Sales

Organisation:Vigor Council

Group : Priyanshu, Saraansh, Vishal

Start Date:21.03.2024 Expected End Date:31.03.2024

[1]:
```
#importing python librarires
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

[2]:
```
#importing excel file
df=pd.read_csv("Sample - Superstore.csv")
```

[3]:
```
#showing the dataset
df
```

[3]:

	Row ID	Order ID	Order Date	Ship Date	Ship Mode	\
0	1	CA-2016-152156	11/8/2016	11/11/2016	Second Class	
1	2	CA-2016-152156	11/8/2016	11/11/2016	Second Class	
2	3	CA-2016-138688	6/12/2016	6/16/2016	Second Class	
3	4	US-2015-108966	10/11/2015	10/18/2015	Standard Class	
4	5	US-2015-108966	10/11/2015	10/18/2015	Standard Class	
…	…	…	…	…	…	
9989	9990	CA-2014-110422	1/21/2014	1/23/2014	Second Class	
9990	9991	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	
9991	9992	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	
9992	9993	CA-2017-121258	2/26/2017	3/3/2017	Standard Class	
9993	9994	CA-2017-119914	5/4/2017	5/9/2017	Second Class	

	Customer ID	Customer Name	Segment	Country	City	\
0	CG-12520	Claire Gute	Consumer	United States	Henderson	
1	CG-12520	Claire Gute	Consumer	United States	Henderson	
2	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	

```
3       SO-20335    Sean O'Donnell   Consumer  United States  Fort Lauderdale
4       SO-20335    Sean O'Donnell   Consumer  United States  Fort Lauderdale
…          …               …         …            …               …
9989    TB-21400  Tom Boeckenhauer   Consumer  United States           Miami
9990    DB-13060       Dave Brooks   Consumer  United States       Costa Mesa
9991    DB-13060       Dave Brooks   Consumer  United States       Costa Mesa
9992    DB-13060       Dave Brooks   Consumer  United States       Costa Mesa
9993    CC-12220      Chris Cortes   Consumer  United States      Westminster

      … Postal Code  Region       Product ID         Category Sub-Category  \
0     …       42420   South  FUR-BO-10001798        Furniture    Bookcases
1     …       42420   South  FUR-CH-10000454        Furniture       Chairs
2     …       90036    West  OFF-LA-10000240  Office Supplies       Labels
3     …       33311   South  FUR-TA-10000577        Furniture       Tables
4     …       33311   South  OFF-ST-10000760  Office Supplies      Storage
…   …         …     …              …              …           …
9989  …       33180   South  FUR-FU-10001889        Furniture  Furnishings
9990  …       92627    West  FUR-FU-10000747        Furniture  Furnishings
9991  …       92627    West  TEC-PH-10003645       Technology       Phones
9992  …       92627    West  OFF-PA-10004041  Office Supplies        Paper
9993  …       92683    West  OFF-AP-10002684  Office Supplies   Appliances

                                              Product Name     Sales  Quantity  \
0                     Bush Somerset Collection Bookcase  261.9600         2
1     Hon Deluxe Fabric Upholstered Stacking Chairs,…  731.9400         3
2     Self-Adhesive Address Labels for Typewriters b…   14.6200         2
3         Bretford CR4500 Series Slim Rectangular Table  957.5775         5
4                           Eldon Fold 'N Roll Cart System   22.3680         2
…                                                      …       …       …
9989                              Ultra Door Pull Handle   25.2480         3
9990  Tenex B1-RE Series Chair Mats for Low Pile Car…   91.9600         2
9991                                Aastra 57i VoIP phone  258.5760         2
9992  It's Hot Message Books with Stickers, 2 3/4" x 5"   29.6000         4
9993  Acco 7-Outlet Masterpiece Power Center, Wihtou…  243.1600         2

      Discount    Profit
0         0.00   41.9136
1         0.00  219.5820
2         0.00    6.8714
3         0.45 -383.0310
4         0.20    2.5164
…          …       …
9989      0.20    4.1028
9990      0.00   15.6332
9991      0.20   19.3932
9992      0.00   13.3200
9993      0.00   72.9480
```

```
[9994 rows x 21 columns]
```

[4]:
```
#checking whether the data has null values
df.isnull().sum()
```

[4]:
```
Row ID           0
Order ID         0
Order Date       0
Ship Date        0
Ship Mode        0
Customer ID      0
Customer Name    0
Segment          0
Country          0
City             0
State            0
Postal Code      0
Region           0
Product ID       0
Category         0
Sub-Category     0
Product Name     0
Sales            0
Quantity         0
Discount         0
Profit           0
dtype: int64
```

[5]:
```
#statistical values
df[["Sales","Quantity","Profit"]].describe()
```

[5]:

	Sales	Quantity	Profit
count	9994.000000	9994.000000	9994.000000
mean	229.858001	3.789574	28.656896
std	623.245101	2.225110	234.260108
min	0.444000	1.000000	-6599.978000
25%	17.280000	2.000000	1.728750
50%	54.490000	3.000000	8.666500
75%	209.940000	5.000000	29.364000
max	22638.480000	14.000000	8399.976000

[6]:
```
df.describe(include="object").T
```

[6]:

	count	unique	top	freq
Order ID	9994	5009	CA-2017-100111	14
Order Date	9994	1237	9/5/2016	38

```
Ship Date      9994   1334       12/16/2015    35
Ship Mode      9994      4   Standard Class  5968
Customer ID    9994    793         WB-21850    37
Customer Name  9994    793    William Brown    37
Segment        9994      3         Consumer  5191
Country        9994      1    United States  9994
City           9994    531    New York City   915
State          9994     49       California  2001
Region         9994      4             West  3203
Product ID     9994   1862  OFF-PA-10001970    19
Category       9994      3  Office Supplies  6026
Sub-Category   9994     17          Binders  1523
Product Name   9994   1850  Staple envelope    48
```

[7]:
```
# checking data type
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Row ID         9994 non-null   int64
 1   Order ID       9994 non-null   object
 2   Order Date     9994 non-null   object
 3   Ship Date      9994 non-null   object
 4   Ship Mode      9994 non-null   object
 5   Customer ID    9994 non-null   object
 6   Customer Name  9994 non-null   object
 7   Segment        9994 non-null   object
 8   Country        9994 non-null   object
 9   City           9994 non-null   object
 10  State          9994 non-null   object
 11  Postal Code    9994 non-null   int64
 12  Region         9994 non-null   object
 13  Product ID     9994 non-null   object
 14  Category       9994 non-null   object
 15  Sub-Category   9994 non-null   object
 16  Product Name   9994 non-null   object
 17  Sales          9994 non-null   float64
 18  Quantity       9994 non-null   int64
 19  Discount       9994 non-null   float64
 20  Profit         9994 non-null   float64
dtypes: float64(3), int64(3), object(15)
memory usage: 1.6+ MB
```

[8]:
```
#changing data type
df['Order Date'] = pd.to_datetime(df['Order Date'], format='mixed')
df['Ship Date'] = pd.to_datetime(df['Ship Date'], format='mixed')
```

[9]:
```
#with changed data type
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 21 columns):
 #   Column         Non-Null Count  Dtype         
---  ------         --------------  -----         
 0   Row ID         9994 non-null   int64         
 1   Order ID       9994 non-null   object        
 2   Order Date     9994 non-null   datetime64[ns]
 3   Ship Date      9994 non-null   datetime64[ns]
 4   Ship Mode      9994 non-null   object        
 5   Customer ID    9994 non-null   object        
 6   Customer Name  9994 non-null   object        
 7   Segment        9994 non-null   object        
 8   Country        9994 non-null   object        
 9   City           9994 non-null   object        
 10  State          9994 non-null   object        
 11  Postal Code    9994 non-null   int64         
 12  Region         9994 non-null   object        
 13  Product ID     9994 non-null   object        
 14  Category       9994 non-null   object        
 15  Sub-Category   9994 non-null   object        
 16  Product Name   9994 non-null   object        
 17  Sales          9994 non-null   float64       
 18  Quantity       9994 non-null   int64         
 19  Discount       9994 non-null   float64       
 20  Profit         9994 non-null   float64       
dtypes: datetime64[ns](2), float64(3), int64(3), object(13)
memory usage: 1.6+ MB
```

[10]:
```
#check number of rows and columns
df.shape
```

[10]:
```
(9994, 21)
```

[11]:
```
def showlabels(ax):
    for data in ax.containers: ax.bar_label(data)
```

1.1 2 Data Analysis

1.2 2.1 Sales Analysis

1.2.1 1) Total Sales

To know the total sales for united states in this dataset

[12]:
```
us_sales=df["Sales"].sum().astype(int)
print(f"The Total Sales for United States in this dataset is:{us_sales}")
```

```
The Total Sales for United States in this dataset is:2297200
```

1.2.2 2) Sales Year on Year

Comparing increase in sales year by year

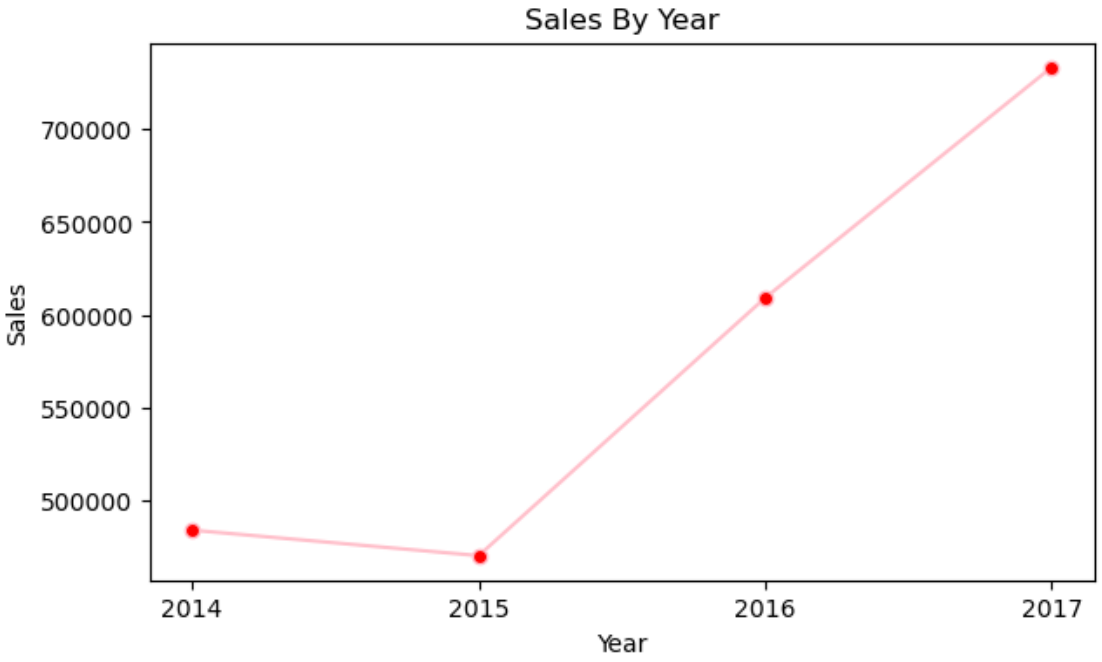
[13]:
```
sales_by_year=df.groupby(df["Order Date"].dt.year)["Sales"].sum()
print(sales_by_year)
```

```
Order Date
2014    484247.4981
2015    470532.5090
2016    609205.5980
2017    733215.2552
Name: Sales, dtype: float64
```

[14]:
```
plt.figure(figsize=(7,4))
line=plt.plot(sales_by_year.
  ↪index,sales_by_year,marker="o",color='pink',mfc='red')
plt.title("Sales By Year")
plt.xlabel("Year")
plt.ylabel("Sales")
plt.xticks(sales_by_year.index)


plt.show
```

[14]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



1.2.3 3) Sales Quarter on Quarter

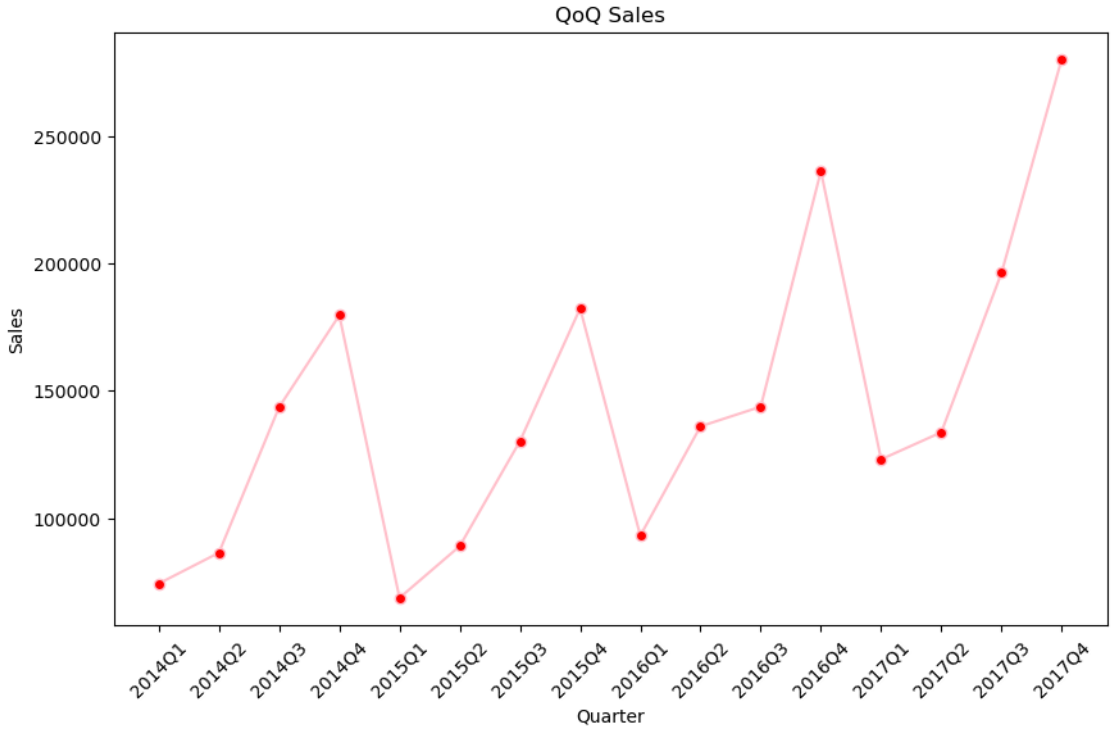
Comparing the sales on quarterly basis

[15]:
```
df['Quarter']=df['Order Date'].dt.to_period("Q")
quarterly_sales=df.groupby("Quarter")["Sales"].sum().astype(int)
print(quarterly_sales)
```

```
Quarter
2014Q1     74447
2014Q2     86538
2014Q3    143633
2014Q4    179627
2015Q1     68851
2015Q2     89124
2015Q3    130259
2015Q4    182297
2016Q1     93237
2016Q2    136082
2016Q3    143787
2016Q4    236098
2017Q1    123144
2017Q2    133764
2017Q3    196251
2017Q4    280054
Freq: Q-DEC, Name: Sales, dtype: int32
```

[16]:
```
df_quarterly = df.groupby(df["Order Date"].dt.to_period("Q"))["Sales"].sum().
  ↪reset_index()

plt.figure(figsize=(10,6))
plt.plot(df_quarterly["Order Date"].astype(str),␣
  ↪df_quarterly["Sales"],marker="o",color='pink',mfc='red')
plt.title("QoQ Sales")
plt.xlabel("Quarter")
plt.ylabel("Sales")
plt.xticks(rotation=45)
plt.show()
```



1.2.4 4) Sales by Region

Comparing which region had the most sales

[17]:
```
Count_of_region=df['Region'].value_counts()
High_sales=Count_of_region.idxmax()
print("Highest sale is in region:",High_sales)
```

```
Highest sale is in region: West
```

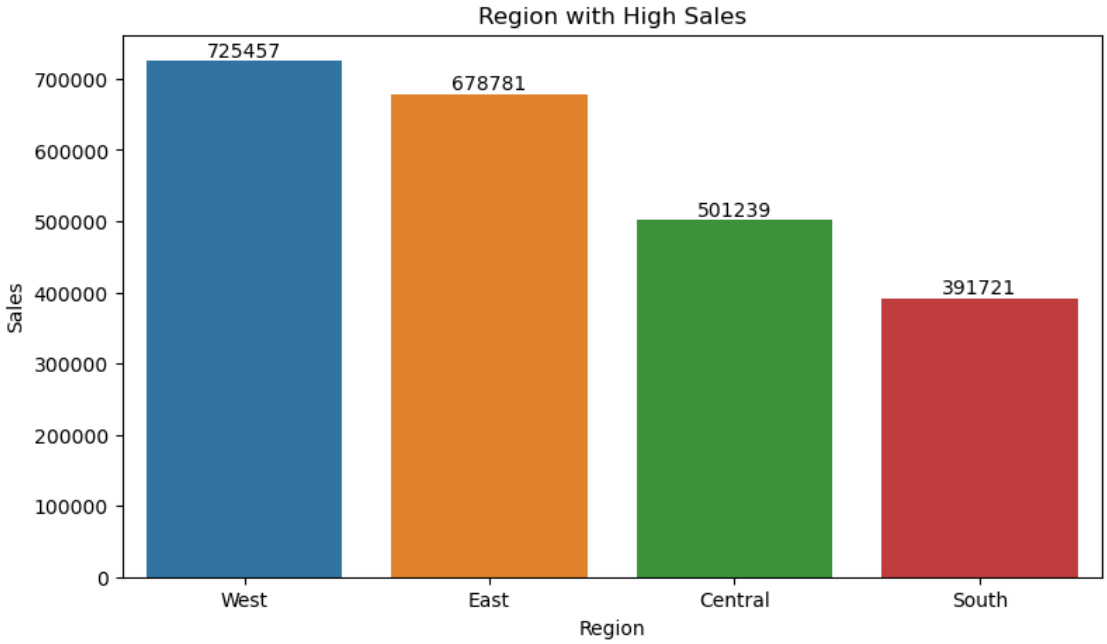
[18]:
```
ount_of_region=df['Region'].value_counts()
High_sales=Count_of_region.idxmin()
print("Lowestt sale is in region:",High_sales)
```

```
Lowestt sale is in region: South
```

[19]:
```
p =df.groupby(["Region"])["Sales"].sum().astype(int)
x =p.sort_values(ascending= False).reset_index()
print(x)
```

```
    Region   Sales
0     West  725457
1     East  678781
2  Central  501239
3    South  391721
```

[20]:
```
plt.figure(figsize=(9,5))
ax=sns.barplot(x="Region",y="Sales", data = x)
showlabels(ax)
plt.title("Region with High Sales")
plt.show()
```



1.2.5 5) Sales by City

Comparing which city had the Most Sales and which city had the Least Sale

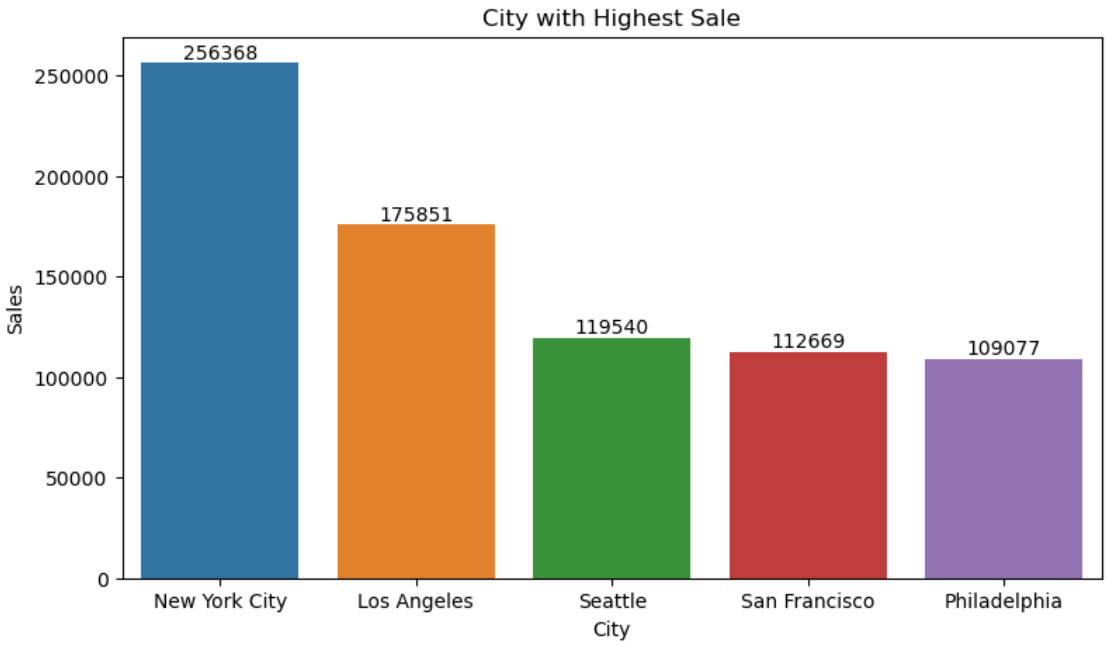
[21]:
```
p =df.groupby(["City"])["Sales"].sum().astype(int)
x =p.sort_values(ascending= False).head(5).reset_index()
print(x)
```

```
           City   Sales
0  New York City  256368
```

```
1    Los Angeles  175851
2        Seattle  119540
3  San Francisco  112669
4   Philadelphia  109077
```

[22]:
```
plt.figure(figsize=(9,5))
ax=sns.barplot(x="City",y="Sales", data = x)
showlabels(ax)
plt.title("City with Highest Sale")
```

[22]:
```
Text(0.5, 1.0, 'City with Highest Sale')
```

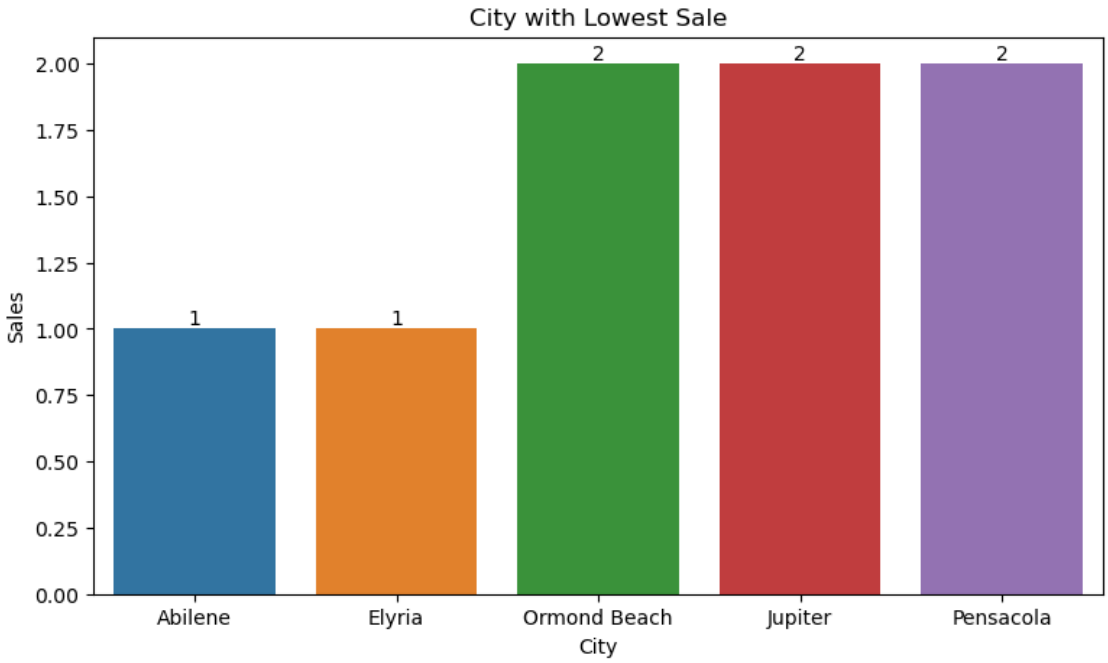


[23]:
```
p =df.groupby(["City"])["Sales"].sum().astype(int)
x =p.sort_values(ascending= True).head(5).reset_index()
print(x)
```

```
           City  Sales
0       Abilene      1
1        Elyria      1
2  Ormond Beach      2
3       Jupiter      2
4     Pensacola      2
```

[24]:
```
plt.figure(figsize=(9,5))
ax=sns.barplot(x="City",y="Sales", data = x)
showlabels(ax)
plt.title("City with Lowest Sale")
```

[24]: Text(0.5, 1.0, 'City with Lowest Sale')



1.3 2.2 Profit Analysis

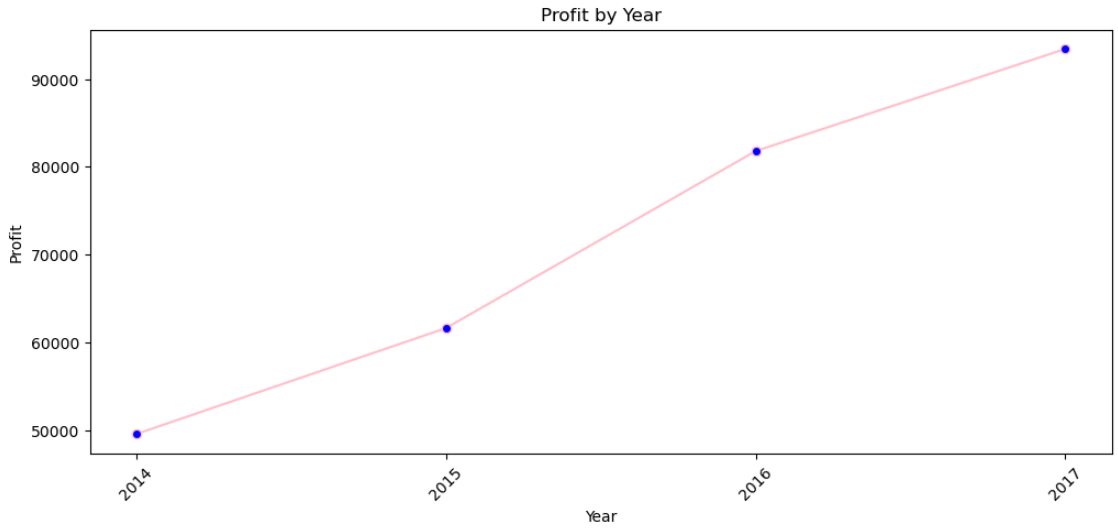
1.3.1 1)Profit Year On Year

Comparing Profits for each year

[25]:
```
profit_by_year=df.groupby(df["Order Date"].dt.year)["Profit"].sum()
print(profit_by_year)
```

```
Order Date
2014    49543.9741
2015    61618.6037
2016    81795.1743
2017    93439.2696
Name: Profit, dtype: float64
```

[26]:
```
plt.figure(figsize=(12,5))
line=plt.plot(profit_by_year.index, profit_by_year.values,␣
 ↪marker="o",color="pink",mfc="blue")
plt.title("Profit by Year")
plt.xlabel("Year")
plt.ylabel("Profit")
plt.xticks(profit_by_year.index,rotation=45)
plt.show()
```



1.3.2 2) Region with Highest profit

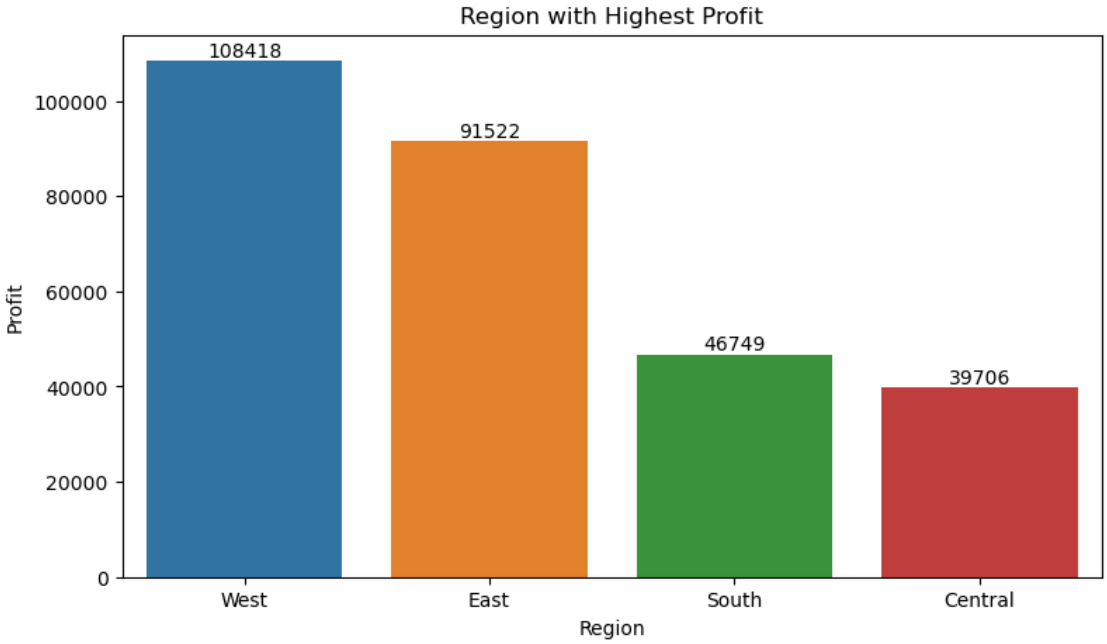
Comparing which region has the most profit

[27]:
```
p =df.groupby(["Region"])["Profit"].sum().astype(int)
x =p.sort_values(ascending= False).reset_index()
print(x)
```

```
    Region  Profit
0     West  108418
1     East   91522
2    South   46749
3  Central   39706
```

[28]:
```
plt.figure(figsize=(9,5))
ax=sns.barplot(x="Region",y="Profit", data = x)
showlabels(ax)
plt.title("Region with Highest Profit")
```

[28]:
```
Text(0.5, 1.0, 'Region with Highest Profit')
```



1.4 3) Profit - City Wise

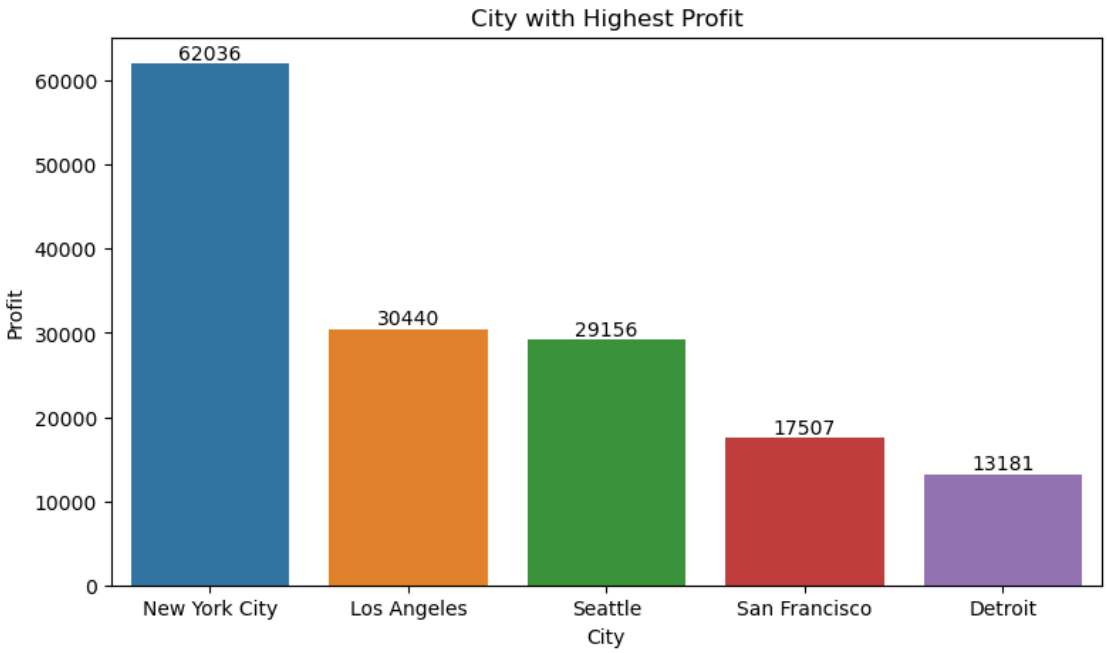
Top 5 City with Highest Profit

[29]:
```
p =df.groupby(["City"])["Profit"].sum().astype(int)
x =p.sort_values(ascending= False).head(5).reset_index()
print(x)
```

```
            City  Profit
0  New York City   62036
1    Los Angeles   30440
2        Seattle   29156
3  San Francisco   17507
4        Detroit   13181
```

[30]:
```
plt.figure(figsize=(9,5))
ax=sns.barplot(x="City",y="Profit", data = x)
showlabels(ax)
plt.title("City with Highest Profit")
```

[30]: Text(0.5, 1.0, 'City with Highest Profit')



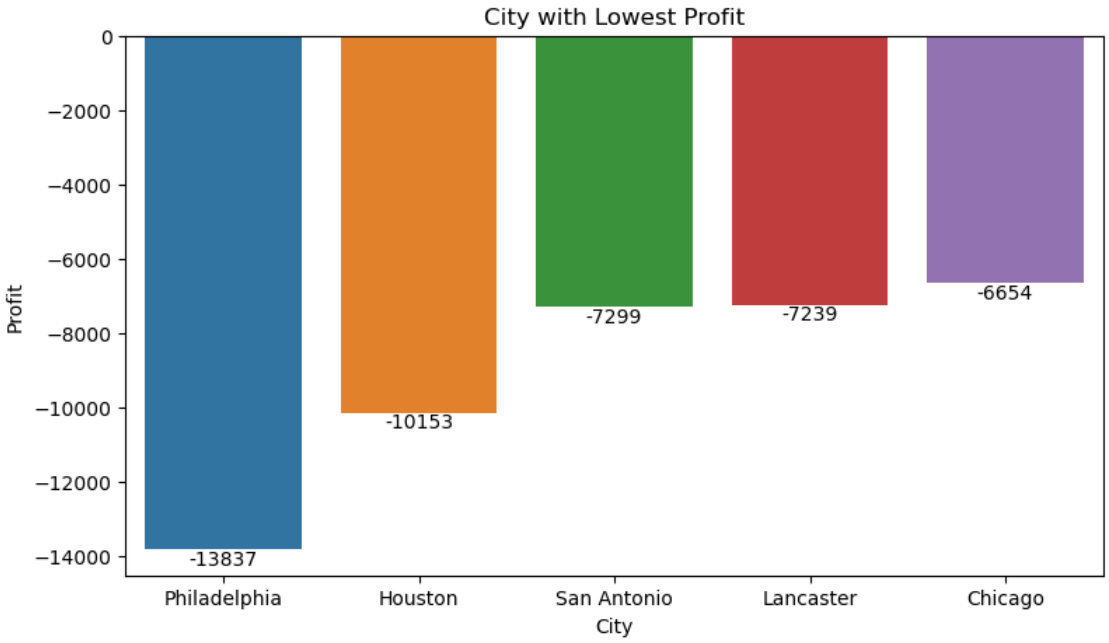
Top 5 City with Lowest Profit

[31]:
```
p =df.groupby(["City"])["Profit"].sum().astype(int)
x =p.sort_values(ascending= True).head(5).reset_index()
print(x)
```

```
          City  Profit
0  Philadelphia  -13837
1       Houston  -10153
2   San Antonio   -7299
3     Lancaster   -7239
4       Chicago   -6654
```

[32]:
```
plt.figure(figsize=(9,5))
ax=sns.barplot(x="City",y="Profit", data = x)
showlabels(ax)
plt.title("City with Lowest Profit")
```

[32]: Text(0.5, 1.0, 'City with Lowest Profit')



1.5 2.3 Customer Analysis

1.5.1 1) Unique Customers

How many Unique Customer are there in this data?

[33]:
```
unique_customers=df['Customer Name'].nunique()
print("Number of unique customers:", unique_customers)
```

```
Number of unique customers: 793
```

1.5.2 2) Average Purchase Order Value

Checking what is the Average value for Order by the Customer

[34]:
```
x=df["Sales"].mean().astype(int)
print(f"{x} is the Average Purchase Order Value per Customer")
```

```
229 is the Average Purchase Order Value per Customer
```

1.5.3 3) Top Customer

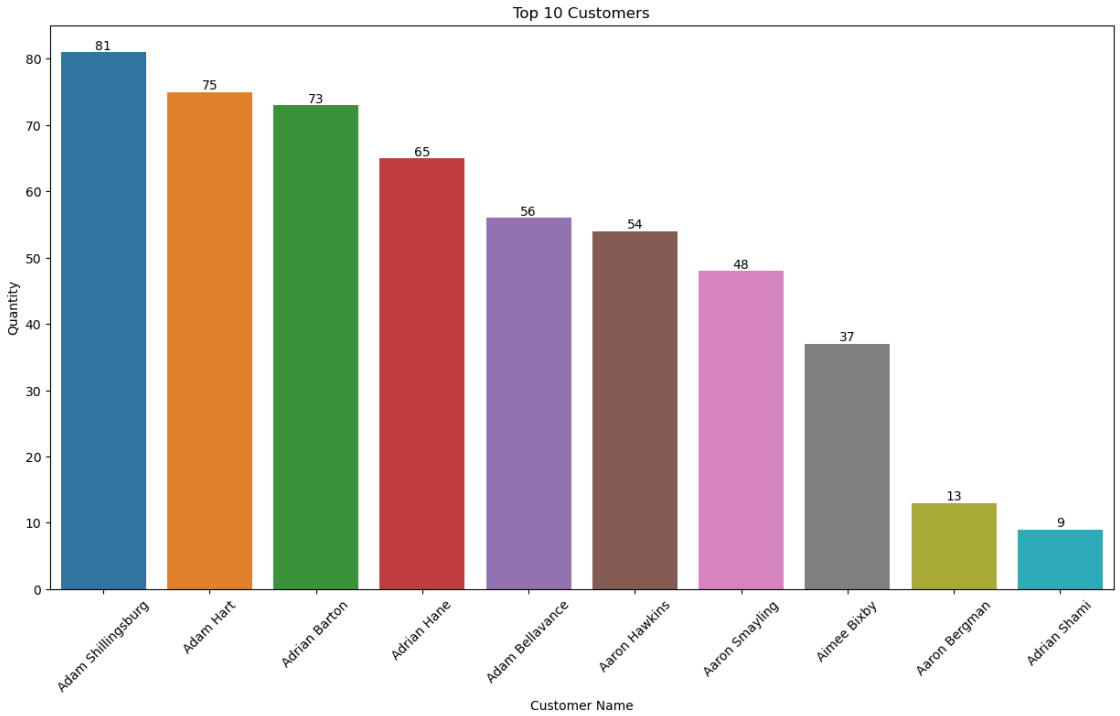
Comparing which customer purchased in most quantity

[35]:
```
x=df.groupby("Customer Name")["Quantity"].sum().head(10).sort_values(ascending=␣
↪False).reset_index()
print(x)
```

```
        Customer Name  Quantity
0  Adam Shillingsburg        81
```

```
1           Adam Hart        75
2       Adrian Barton        73
3         Adrian Hane        65
4     Adam Bellavance        56
5       Aaron Hawkins        54
6      Aaron Smayling        48
7         Aimee Bixby        37
8       Aaron Bergman        13
9        Adrian Shami         9
```

[36]:
```
plt.figure(figsize=(15,8))
ax=sns.barplot(x="Customer Name", y="Quantity", data=x)
plt.title("Top 10 Customers")
showlabels(ax)
plt.xticks(rotation=45)
plt.show()
```



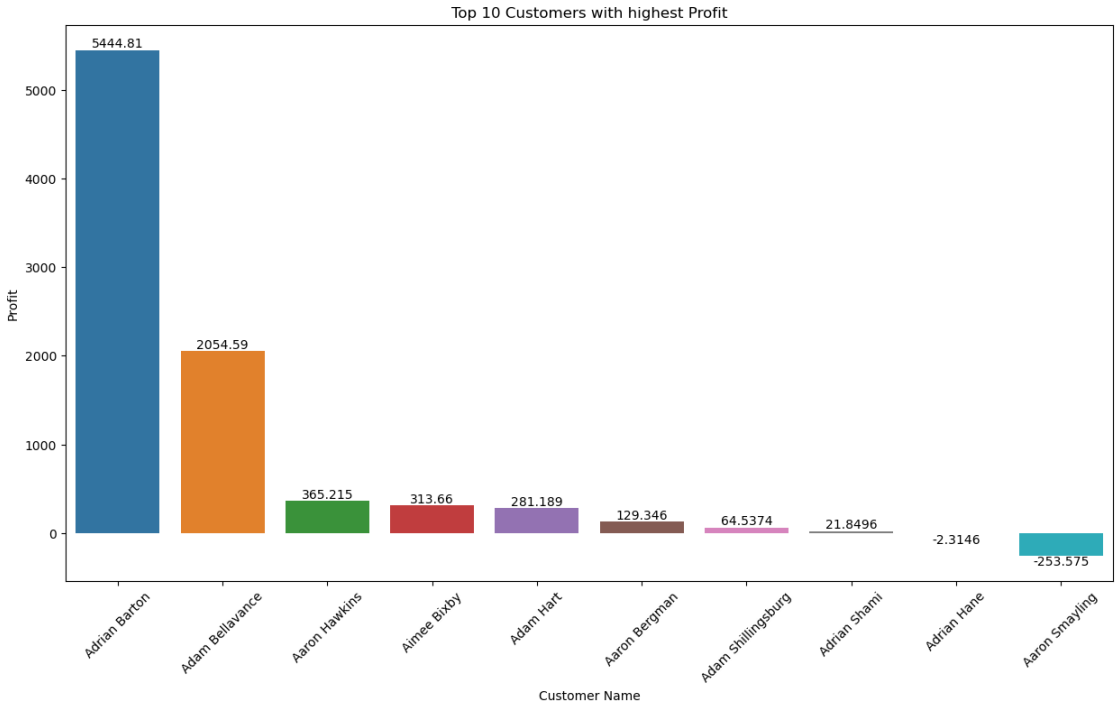
1.5.4 4) Most Profitable Customer

[37]:
```
x=df.groupby("Customer Name")["Profit"].sum().head(10).sort_values(ascending=␣
↪False).reset_index()
print(x)
```

```
      Customer Name     Profit
0       Adrian Barton  5444.8055
```

```
1     Adam Bellavance  2054.5885
2       Aaron Hawkins   365.2152
3         Aimee Bixby   313.6597
4           Adam Hart   281.1890
5       Aaron Bergman   129.3465
6  Adam Shillingsburg    64.5374
7        Adrian Shami    21.8496
8         Adrian Hane    -2.3146
9      Aaron Smayling  -253.5746
```

[38]:
```
plt.figure(figsize=(15,8))
ax=sns.barplot(x="Customer Name", y="Profit", data=x)
plt.title("Top 10 Customers with highest Profit")
showlabels(ax)
plt.xticks(rotation=45)
plt.show()
```



1.5.5 5) Average Quantity Ordered Per Customer

[39]:
```
x=df["Quantity"].mean().astype(int)
print(f"The Average Quantity Ordered per Customer is:{x}")
```

```
The Average Quantity Ordered per Customer is:3
```

1.6 2.4 Category Analysis

1.6.1 1) Sales - Category Wise

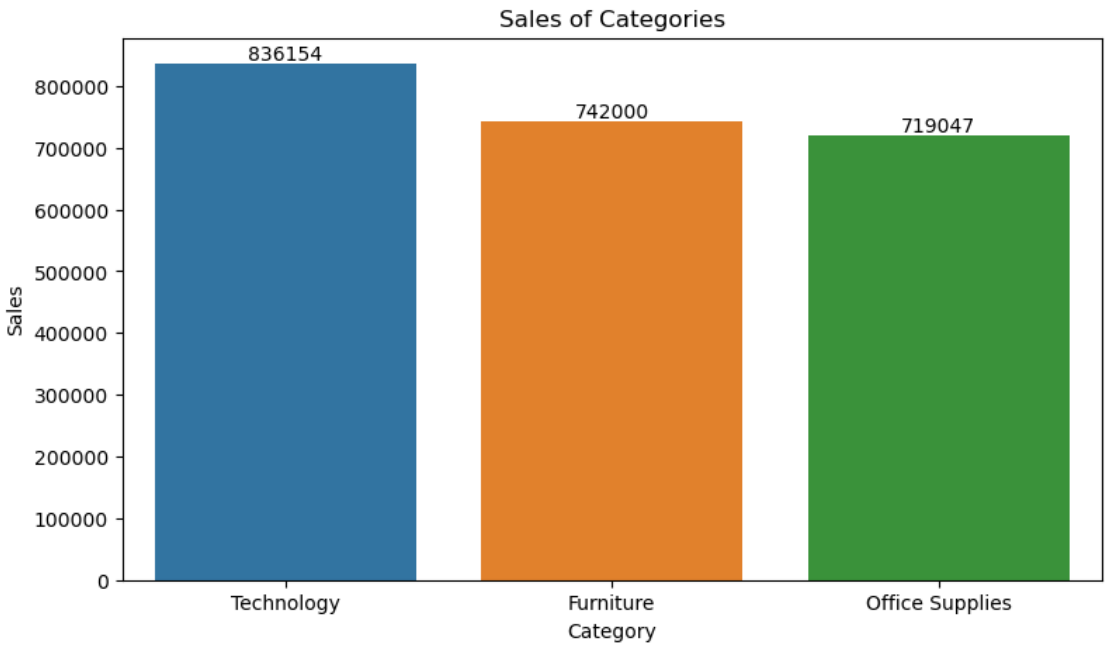
Comparing which category had the most Sales

[40]:
```
x=df.groupby("Category")["Sales"].sum().sort_values(ascending=False).
  ↪reset_index()
print(x)
```

```
          Category        Sales
0       Technology  836154.0330
1        Furniture  741999.7953
2  Office Supplies  719047.0320
```

[41]:
```
plt.figure(figsize=(9,5))
ax=sns.barplot(x="Category",y="Sales", data = x)
showlabels(ax)
plt.title("Sales of Categories")
```

[41]: Text(0.5, 1.0, 'Sales of Categories')



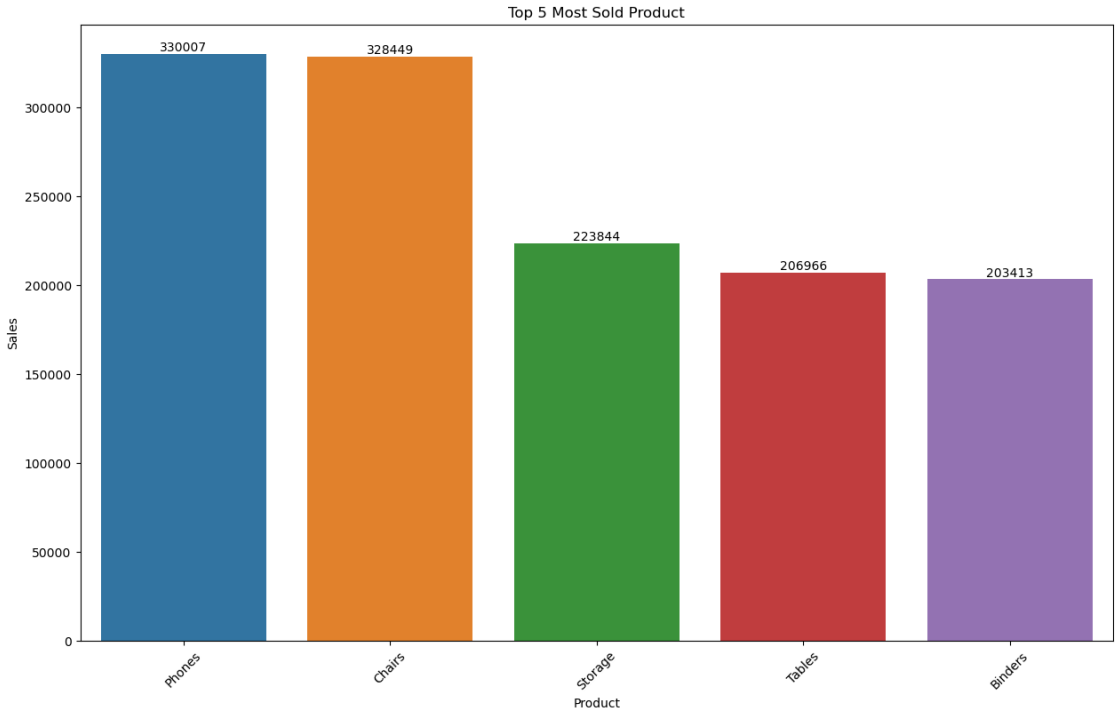
1.6.2 2) Sales - Product Wise

Comparing which product had the Most Sales and which Product had the Least Sales

[42]:
```
x=df.groupby("Sub-Category")["Sales"].sum().sort_values(ascending=False).
  ↪reset_index().head(5)
print(x)
```

```
  Sub-Category       Sales
0       Phones  330007.054
1       Chairs  328449.103
2      Storage  223843.608
3       Tables  206965.532
4      Binders  203412.733
```

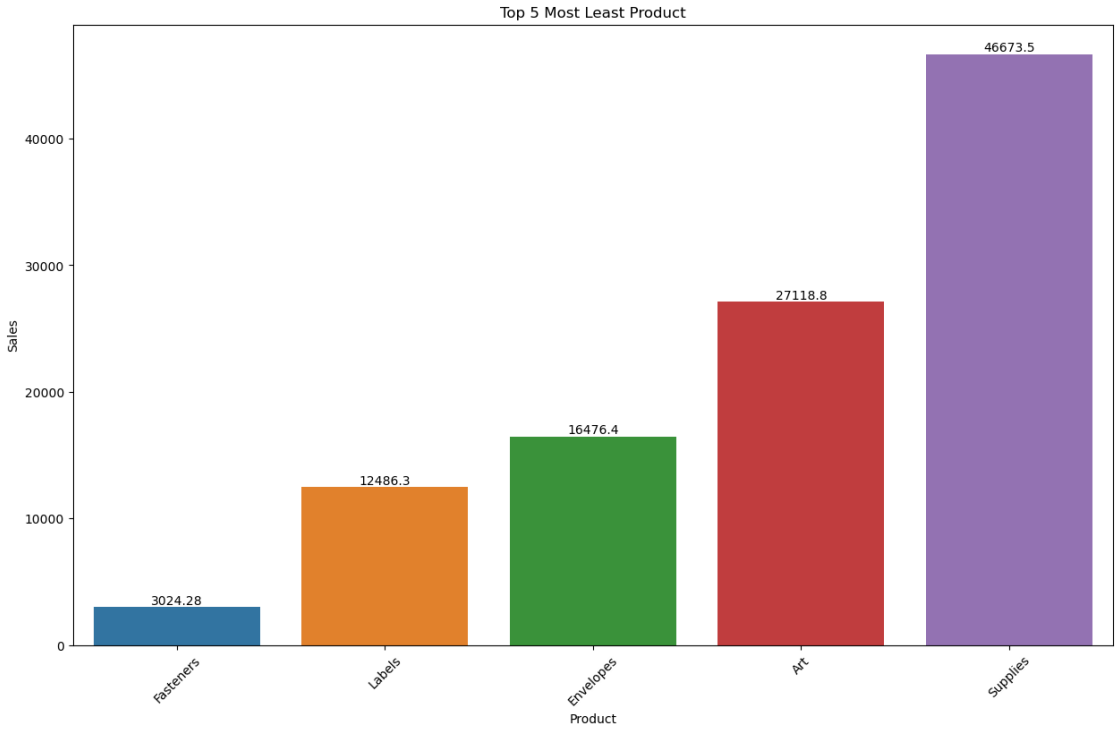
[43]:
```
plt.figure(figsize=(15,9))
ax=sns.barplot(x="Sub-Category",y="Sales", data = x)
plt.title("Top 5 Most Sold Product")
plt.xticks(rotation=45)
plt.xlabel("Product")
showlabels(ax)
```



[44]:
```
x=df.groupby("Sub-Category")["Sales"].sum().sort_values(ascending=True).
  ↪reset_index().head(5)
print(x)
```

```
  Sub-Category      Sales
0    Fasteners   3024.280
1       Labels  12486.312
2    Envelopes  16476.402
3          Art  27118.792
4     Supplies  46673.538
```

[45]:
```
plt.figure(figsize=(15,9))
ax=sns.barplot(x="Sub-Category",y="Sales", data = x)
plt.title("Top 5 Most Least Product")
plt.xticks(rotation=45)
plt.xlabel("Product")
showlabels(ax)
```



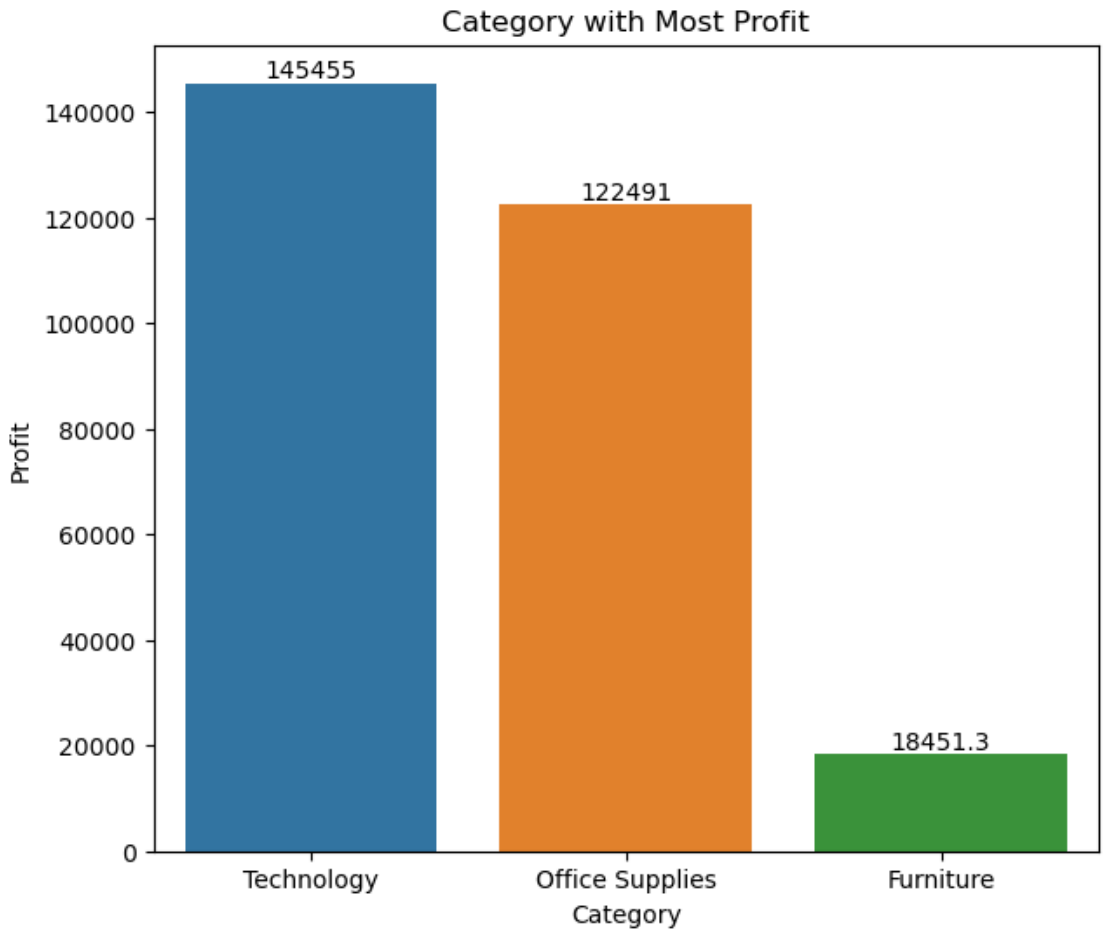
1.6.3 3) Profit - Category Wise

Comparing which category had the most profit

[46]:
```
cw=df.groupby("Category")["Profit"].sum().sort_values(ascending=False).
↪reset_index()
print(cw)
```

```
          Category       Profit
0       Technology  145454.9481
1  Office Supplies  122490.8008
2        Furniture   18451.2728
```

[47]:
```
plt.figure(figsize=(7,6))
ax=sns.barplot(x="Category",y="Profit", data=cw)
showlabels(ax)
plt.title("Category with Most Profit")
plt.show()
```



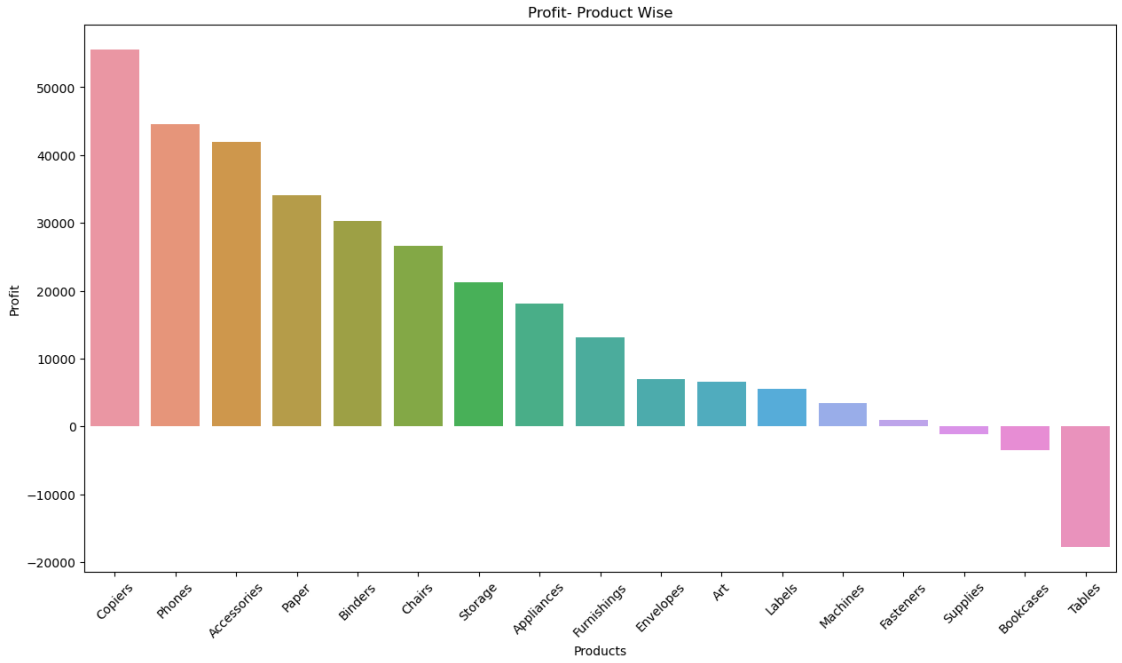
1.6.4 4) Profit - Product Wise

[48]:
```
x=df.groupby("Sub-Category")["Profit"].sum().sort_values(ascending=False).
  ↪reset_index()
print(x)
```

```
   Sub-Category      Profit
0       Copiers  55617.8249
1        Phones  44515.7306
2   Accessories  41936.6357
3         Paper  34053.5693
4       Binders  30221.7633
5        Chairs  26590.1663
6       Storage  21278.8264
7    Appliances  18138.0054
8   Furnishings  13059.1436
9     Envelopes   6964.1767
10          Art   6527.7870
```

```
11       Labels   5546.2540
12     Machines   3384.7569
13    Fasteners    949.5182
14     Supplies  -1189.0995
15    Bookcases  -3472.5560
16       Tables -17725.4811
```

[49]:
```
plt.figure(figsize=(15,8))
ax=sns.barplot(x="Sub-Category",y="Profit", data=x)
plt.title("Profit- Product Wise")
plt.xlabel("Products")
plt.xticks(rotation=45)
plt.show()
showlabels(ax)
```



1.7 2.5 Shipment Analysis

1.7.1 2.5.1 Most Preferred/Opted Shipment Mode

[50]:
```
x=df["Ship Mode"].value_counts().idxmax()
print(f"{x} is the most preferred shipment mode.")
```

```
Standard Class is the most preferred shipment mode.
```

1.8 3. Summary

1.8.1 1)Sales Summary

Region West
Achieved the highest sales, totaling `8,725,457 units`

Top Sales Cities `New York`: Contributed 256,368 units in sales.
`California`: Followed closely with 175,851 units in sales.

Lowest Sales Cities
`Abilene and Elyria`: Both cities had only 1 sale each.

1.8.2 2)Profit Summary

Year-on-Year Profit
The Superstore's profit is increasing.

Region West
Led in profit, totaling 108,414 units.

Top Profit Cities
`New York`: Generated 62,036 units in profit.
`Los Angeles`: Followed with 30,440 units in profit.

Lowest Profit Cities
`Philadelphia`: Recorded a negative profit of 13,837 units.
`Houston`: Also had a negative profit of 10,153 units.

1.8.3 3)Customer Summary

Unique Customers
There are `793` unique customers.

Average Purchase Order Value
Each customer's average purchase order value is `229` units.

Top 10 Customers (Based on Quantity of Purchases)
Adam Shillingsburg: Made 81 purchases.
Adam Hart: Followed with 75 purchases.

1.8.4 4)Category Summary

Sales by Product
`Supplies`: Lead in sales with 46,671.5 units sold.
`Ink`: Second-highest selling product with 27,118.8 units sold.
`Envelopes and Labels`: Comparable sales figures, with 16,716.4 and 12,488.3 units sold respectively.

`Beverages`: Despite having the least sales at 3,024.28 units, they still maintain a significant market presence.

Top-Selling Products
`Phones`: Sold 330,007 units.
`Chairs`: Followed closely with 328,449 units.

Most Profitable Products
`Copiers`: Generated 55,617 units in profit.
`Phones`: Followed closely in profitability.

1.9 4. Conclusion

The Superstore has demonstrated a strong performance with increasing year-on-year profits, led by the West region and cities like New York and California. However, it's crucial to address the losses in cities like Philadelphia, Houston, and San Antonio.

The customer base is robust with 793 unique customers, but there's potential for growth by encouraging lower-volume customers to increase their purchasing activity. This could boost overall sales and average purchase order value.

In terms of products, Supplies lead the sales, and Phones and Chairs are the top-selling items. However, there's room for growth in the sales of Beverages, Labels, and Envelopes. The Superstore should also focus on increasing the sales of Labels and Fasteners, and reducing the production cost for Bookcases and Tables, which are currently making a loss.

The Technology category has the most sales and profit, indicating a strong market presence. However, the Furniture category, which has the least sales and profit, needs more focus.

In conclusion, while the Superstore is on a positive trajectory, strategic efforts should be made to boost sales in underperforming areas, reduce costs for less profitable products, and encourage increased purchasing activity among lower-volume customers. A deeper analysis of customer behavior and preferences could provide valuable insights to inform these strategies.