Super Store Sales Analysis

Analyse sales data of Super Store mart and identify opportunities to boost business growth

Fig – 1.
[Shoprite - Nigeria | Lower Prices You Can Trust]

Introduction

Super Store is a small retail business located in the United States. They sell Furniture, Office Supplies and Technology products and their customers are the mass Consumer, Corporate and Home Offices.The data set contains sales, profit and geographical information of Super Store.

Our task is to analyse the sales data and identify weak areas and opportunities for Super Store to boost business growth.

After collecting our data we can process the data through using Python Programming and we would be recommending some business advantages to the stores and others who are doing similar business.

Business Questions

1. Which Category is Best Selling and Most Profitable?
2. What are the Best Selling and Most Profitable Sub-Category?
3. Which is the Top Selling Sub-Category?
4. Which Customer Segment is Most Profitable?
5. Which is the Preferred Ship Mode?
6. Which Region is the Most Profitable?
7. Which City has the Highest Number of Sales?

Preparing the Environment

We will import the required libraries and read in the data set and we are using Python in Ananconda Enviroment

The Dataset is with this folder or you can get it from Kaggle.com
Dataset name – (SampleSuperstore.csv)

Pandas — Data manipulation
Matplotlib and Seaborn — Data visualisation

```
# Import libraries and alias for easy reading
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
%matplotlib inline

# Read in data in CSV format
superstore = pd.read_csv('../Data Analytics/The Sparks
Internship/Super Store/SampleSuperstore.csv')
```

Fig – 2.

Data Exploration

Let's have a look at the data using df.head() and df.tail() function.

```
# Preview first 5 rows of data set
superstore.head()
```

Fig – 3

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

Preview first 5 rows of Super Store data set

Fig – 4

```
# Preview last 5 rows of data set
superstore.tail()
```

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

Preview last 5 rows

Fig – 5

You can easily tell that there is a mix of categorical, geographical and numerical variables.

Each row represents an order of an item and the corresponding quantity, sales, discount and profit. There is also the mode of shipment, customer segment and geographical aspects.

Next, we will use the df.shape() and df.info() to get more information.

```
# Shape of data set
superstore.shape

(9994, 13)

# Summarised information of data set
superstore.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
```

Fig – 6

There are 9,994 rows including header and 13 columns. Data type is correct and matches the corresponding values.

Data Cleaning

Before we start with the analysis, we must first clean the data or "scrub the dirt".

For this analysis, we will look at the more common issues such as missing and duplicate data.

Handling Missing Values

Here, we will find out whether there is Null value.

```
# Find the number of null values for all columns
superstore.isnull().sum()


Ship Mode       0
Segment         0
Country         0
City            0
State           0
Postal Code     0
Region          0
Category        0
Sub-Category    0
Sales           0
Quantity        0
Discount        0
Profit          0
dtype: int64
```

Fig – 7

The result shows that the data set does not have any null value.

Duplicate Data

Then, we will find out whether there is duplicate data.

```
# Find the number of duplicate data
superstore.duplicated().sum()


17
```

Fig – 8

Result shows that there are 17 duplicated rows. Let’s run another function to view the duplicated data.

```
# Show the duplicated rows
superstore[superstore.duplicated(keep = 'last')]
```

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
568	Standard Class	Corporate	United States	Seattle	Washington	98105	West	Office Supplies	Paper	19.440	3	0.0	9.3312
591	Standard Class	Consumer	United States	Salem	Oregon	97301	West	Office Supplies	Paper	10.368	2	0.2	3.6288
935	Standard Class	Home Office	United States	Philadelphia	Pennsylvania	19120	East	Office Supplies	Paper	15.552	3	0.2	5.4432
1186	Standard Class	Corporate	United States	Seattle	Washington	98103	West	Office Supplies	Paper	25.920	4	0.0	12.4416
1479	Standard Class	Consumer	United States	San Francisco	California	94122	West	Office Supplies	Paper	25.920	4	0.0	12.4416
2803	Standard Class	Consumer	United States	San Francisco	California	94122	West	Office Supplies	Paper	12.840	3	0.0	5.7780
2807	Second Class	Consumer	United States	Seattle	Washington	98115	West	Office Supplies	Paper	12.960	2	0.0	6.2208
2836	Standard Class	Consumer	United States	Los Angeles	California	90036	West	Office Supplies	Paper	19.440	3	0.0	9.3312
3127	Standard Class	Consumer	United States	New York City	New York	10011	East	Office Supplies	Paper	49.120	4	0.0	23.0864
3405	Standard Class	Home Office	United States	Columbus	Ohio	43229	East	Furniture	Chairs	281.372	2	0.3	-12.0588
3412	Standard Class	Corporate	United States	San Francisco	California	94122	West	Office Supplies	Art	11.760	4	0.0	3.1752
5372	Standard Class	Corporate	United States	Houston	Texas	77041	Central	Office Supplies	Paper	15.552	3	0.2	5.4432
5493	Same Day	Home Office	United States	San Francisco	California	94122	West	Office Supplies	Labels	41.400	4	0.0	19.8720
6245	Standard Class	Home Office	United States	Seattle	Washington	98105	West	Furniture	Furnishings	22.140	3	0.0	6.4206
6409	First Class	Consumer	United States	Houston	Texas	77041	Central	Office Supplies	Paper	47.952	3	0.2	16.1838
8457	Second Class	Corporate	United States	Chicago	Illinois	60653	Central	Office Supplies	Binders	3.564	3	0.8	-6.2370
8533	Standard Class	Consumer	United States	Detroit	Michigan	48227	Central	Furniture	Chairs	389.970	3	0.0	35.0973

Preview the 17 duplicated rows

Fig – 9

We will then run the drop_duplicates() function to drop the duplicated rows.

```
# Drop the duplicated rows
superstore.drop_duplicates(inplace = True)

# Find the no. of rows and columns
superstore.shape

(9977, 13)
```

Fig – 10

Alright! We can confirm that the duplicated rows have been dropped as the rows were 9993 before and now it's reduced to 9977 rows.

Calculated Field

Now, we will create a calculated field for Profit Margin. The formula is (Profit / Sales) * 100.

```
superstore['Profit Margin %'] = (superstore.Profit / superstore.Sales)
* 100
superstore.head(5)
```

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit	Profit Margin %
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	0.00	41.9136	16.00
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9400	3	0.00	219.5820	30.00
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14.6200	2	0.00	6.8714	47.00
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5775	5	0.45	-383.0310	-40.00
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.3680	2	0.20	2.5164	11.25

The new calculated field 'Profit Margin %' is added to the last column

Fig – 11

Now that the data set has been scrubbed, we can proceed with some statistics analysis!

Descriptive Statistics

Here, we will do a descriptive statistical analysis. We use df.describe() and assign 'include = 'all' to ensure that categorical features are also included in the output.

```python
# Get descriptive statistics summary
superstore.describe(include = "all")
```

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit	Prof Mar
count	9977	9977	9977	9977	9977	9977.000000	9977	9977	9977	9977.000000	9977.000000	9977.000000	9977.00000	997
unique	4	3	1	531	49	NaN	4	3	17	NaN	NaN	NaN	NaN	NaN
top	Standard Class	Consumer	United States	New York City	California	NaN	West	Office Supplies	Binders	NaN	NaN	NaN	NaN	NaN
freq	5955	5183	9977	914	1996	NaN	3193	6012	1522	NaN	NaN	NaN	NaN	NaN
mean	NaN	NaN	NaN	NaN	NaN	55154.964117	NaN	NaN	NaN	230.148902	3.790719	0.156278	28.69013	12.0
std	NaN	NaN	NaN	NaN	NaN	32058.266816	NaN	NaN	NaN	623.721409	2.226657	0.206455	234.45784	46.6
min	NaN	NaN	NaN	NaN	NaN	1040.000000	NaN	NaN	NaN	0.444000	1.000000	0.000000	-6599.97800	-275
25%	NaN	NaN	NaN	NaN	NaN	23223.000000	NaN	NaN	NaN	17.300000	2.000000	0.000000	1.72620	7.50
50%	NaN	NaN	NaN	NaN	NaN	55901.000000	NaN	NaN	NaN	54.816000	3.000000	0.200000	8.67100	27.0
75%	NaN	NaN	NaN	NaN	NaN	90008.000000	NaN	NaN	NaN	209.970000	5.000000	0.200000	29.37200	36.2
max	NaN	NaN	NaN	NaN	NaN	99301.000000	NaN	NaN	NaN	22638.480000	14.000000	0.800000	8399.97600	50.0

Whoops, not able to screenshot Profit Margin, but it's here on the farthest right.

Fig – 12

You will see 'NaN' in some of the categorical columns and that's perfectly fine. Categorical values are not meant to have calculations performed on them so, we can ignore those.

What we want to focus is the unique count and frequency of the categorical features such as

There are 4 ship modes and 3 customer segments. Nearly half of the orders are from Consumer segment using Standard Class shipment.
The store carries 3 category of items with 17 sub-category. 60% of orders are for Office Supplies and at least 15% purchases are for Binders.
While for numerical parameters, it's interesting to note that the

75% of orders makes at least 36% profit margin.
Loss-making orders can go up to 275% losses. We must place emphasis on these loss-making sales to cut them off.
We will perform some analysis below to test these observations.

Exploratory Data Analysis(EDA)

1. Which Category is Best Selling and Most Profitable?

Best Selling means looking at the Sales and Most Profitable is referring to the Profit. We will group the Sales, Profit and also Quantity by Category and run the df.group().sum and pd.DataFrame() functions.

```
# Group sales, profit and quantity by category
category_analysis = pd.DataFrame(superstore.groupby(['Category'])
[['Sales', 'Profit', 'Quantity']].sum())
category_analysis
```

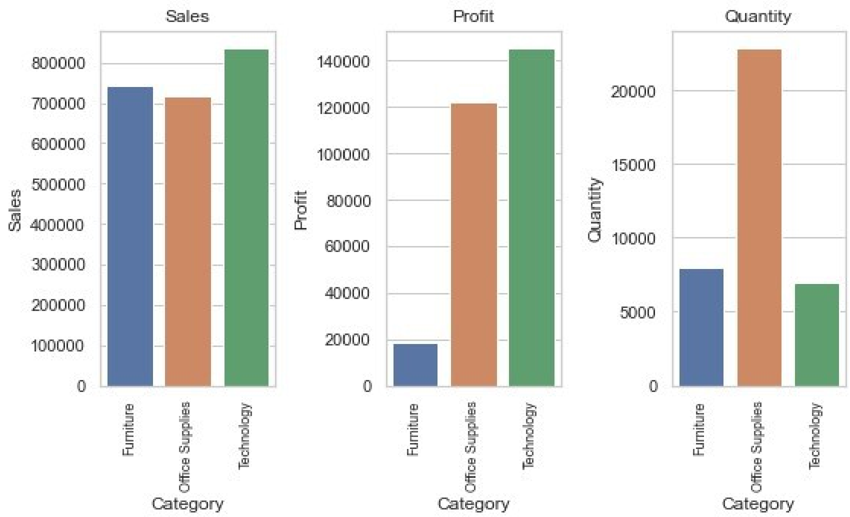
	Sales	Profit	Quantity
Category			
Furniture	741306.3133	18421.8137	8020
Office Supplies	718735.2440	122364.6608	22861
Technology	836154.0330	145454.9481	6939

Group by Sales, Profit and Quantity by Category

Fig – 13

```
# Set for grouped plots - figure with a 2x2 grid of Axes
sns.set_theme(style="whitegrid")
figure, axis = plt.subplots(1, 3, figsize=(8, 5))


# Plot barplots
cat1 = sns.barplot(x = category_analysis.index, y =
category_analysis.Sales, ax=axis[0])
cat2 = sns.barplot(x = category_analysis.index, y =
category_analysis.Profit, ax=axis[1])
cat3 = sns.barplot(x = category_analysis.index, y =
category_analysis.Quantity, ax=axis[2])


# Set titles
cat1.set(title = 'Sales')
cat2.set(title = 'Profit')
cat3.set(title = 'Quantity')


# Rotate axis for x-axis
plt.setp(cat1.get_xticklabels(), rotation = 'vertical', size = 9)
plt.setp(cat2.get_xticklabels(), rotation = 'vertical', size = 9)
plt.setp(cat3.get_xticklabels(), rotation = 'vertical', size = 9)


# Set spacing between subplots
figure.tight_layout()
```

Fig – 14



Bar plots of Sales, Profit and Quantity by Category

Fig – 15

Our observations are:

All 3 categories — Furniture, Office Supplies and Technology make similar amount of sales.

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.

Although Furniture makes similar sales as Technology, it 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.

2. What are the Best Selling and Most Profitable Sub-Category?

```
# Group by sub-category
subcat_analysis = pd.DataFrame(superstore.groupby(['Sub-Category'])
[['Sales', 'Profit']].sum())

# Sort by descending order according to sales
subcat_sales = pd.DataFrame(subcat_analysis.sort_values('Sales',
ascending = False))
subcat_sales
```

Fig – 16

	Sales	Profit
Sub-Category		
Phones	330007.0540	44515.7306
Chairs	327777.7610	26567.1278
Storage	223843.6080	21278.8264
Tables	206965.5320	-17725.4811
Binders	203409.1690	30228.0003
Machines	189238.6310	3384.7569
Accessories	167380.3180	41936.6357
Copiers	149528.0300	55617.8249
Bookcases	114879.9963	-3472.5560
Appliances	107532.1610	18138.0054
Furnishings	91683.0240	13052.7230
Paper	78224.1420	33944.2395
Supplies	46673.5380	-1189.0995
Art	27107.0320	6524.6118
Envelopes	16476.4020	6964.1767
Labels	12444.9120	5526.3820
Fasteners	3024.2800	949.5182

Data frame of Sales and Profit by the Sub-Category in descending order

Fig – 17

```
# Sort by descending order according to profit
subcat_profit = pd.DataFrame(subcat_analysis.sort_values('Profit',
ascending = False))
subcat_profit
```

Fig – 18

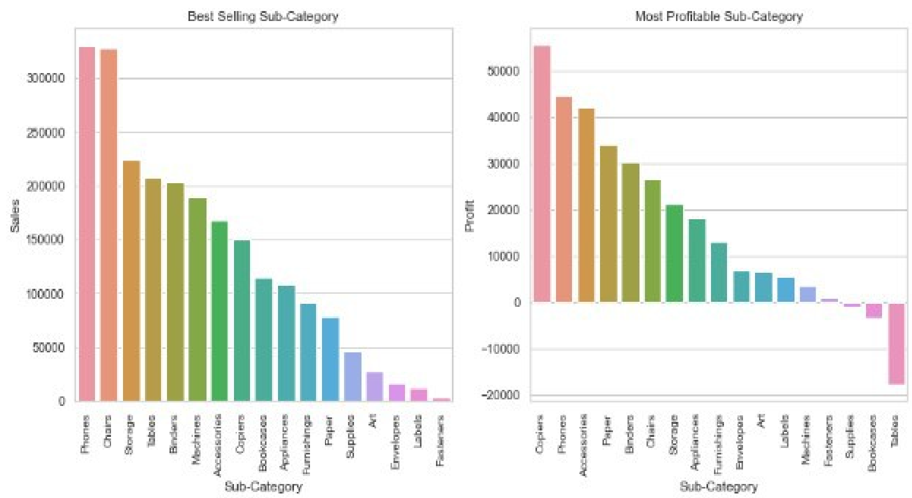
	Sales	Profit
Sub-Category		
Copiers	149528.0300	55617.8249
Phones	330007.0540	44515.7306
Accessories	167380.3180	41936.6357
Paper	78224.1420	33944.2395
Binders	203409.1690	30228.0003
Chairs	327777.7610	26567.1278
Storage	223843.6080	21278.8264
Appliances	107532.1610	18138.0054
Furnishings	91683.0240	13052.7230
Envelopes	16476.4020	6964.1767
Art	27107.0320	6524.6118
Labels	12444.9120	5526.3820
Machines	189238.6310	3384.7569
Fasteners	3024.2800	949.5182
Supplies	46673.5380	-1189.0995
Bookcases	114879.9963	-3472.5560
Tables	206965.5320	-17725.4811

Data frame of Sales and Profit by the Sub-Category in ascending order

Fig – 19

```
# Plot Bar Plots

sns.set_theme(style="whitegrid")

# Set for grouped plots - figure with a 1x2 grid of Axes
figure, axis = plt.subplots(1, 2, figsize=(12, 6))

# Plot Bar Plot for Best Selling Sub-Category
subcat1 = sns.barplot(data = subcat_sales, x = subcat_sales.index, y =
subcat_sales.Sales, ax=axis[0])
subcat1.set(title="Best Selling Sub-Category")
subcat1.set_xticklabels(subcat1.get_xticklabels(),rotation =
"vertical", size = 10)

# Plot Bar Plot for Most Profitable Sub-Category
subcat2 = sns.barplot(data = subcat_profit, x = subcat_profit.index, y
= subcat_profit.Profit, ax=axis[1])
subcat2.set(title = "Most Profitable Sub-Category")
subcat2.set_xticklabels(subcat2.get_xticklabels(),rotation =
"vertical", size = 10)

# Set spacing between subplots
figure.tight_layout()

plt.show()
```

Fig – 20



Bar plots showing the (L) Best Selling and (R) Most Profitable for Sub-Category

Fig – 21

Let's analyse the bar plots:

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.

3. **Which is the Top Selling Sub-Category?**

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

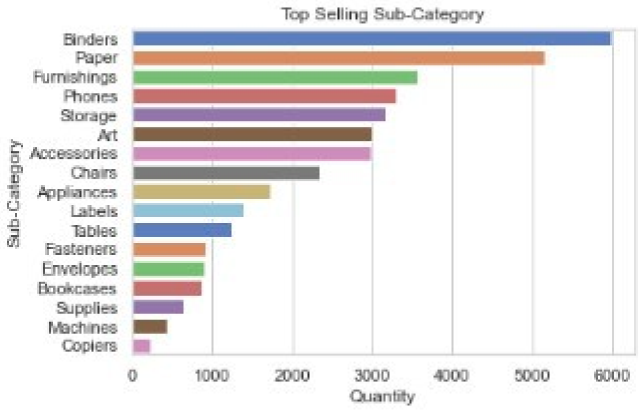
Fig – 22

	Quantity
Sub-Category	
Binders	5971
Paper	5144
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

Quantity sold for each Sub-Category

Fig – 23

```
# Plot Bar Plot for Top Selling Sub-Category
sns.set_theme(style="whitegrid")
sns.barplot(data = subcat_quantity, y = subcat_quantity.index, x =
subcat_quantity.Quantity, palette = "muted")
plt.title("Top Selling Sub-Category")
plt.show()
```



Bar plot showing Top Selling for each Sub-Category

Fig – 24

Here, we can deduce that:

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.

4. **Which Customer Segment is Most Profitable?**

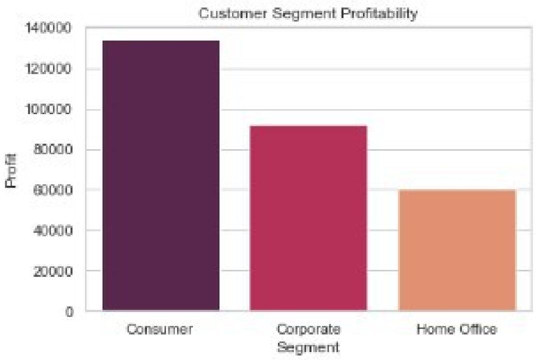
```
segment_analysis = pd.DataFrame(superstore.groupby(['Segment'])
[['Profit']].sum())
segment_analysis
```

	Profit
Segment	
Consumer	134007.4413
Corporate	91954.9798
Home Office	60279.0015

Total Profit by Customer Segment

Fig – 25

```
# Plot Bar Plot
sns.set_theme(style="whitegrid")
sns.barplot(data = segment_analysis, x = segment_analysis.index, y =
segment_analysis.Profit, palette = "rocket")
plt.title("Customer Segment Profitability")
plt.show()
```



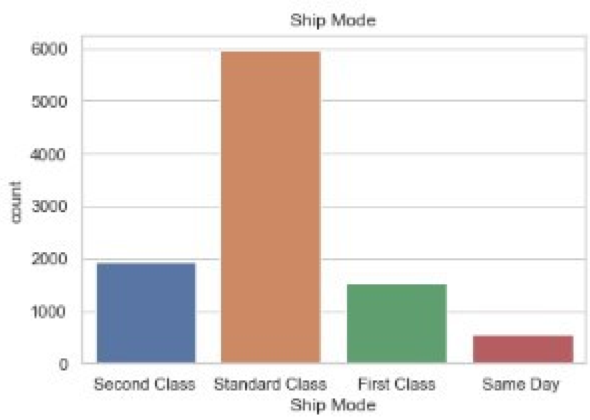
Plot showing Total Profit for each customer Segment

Fig – 26

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.

5. Which is the Preferred Ship Mode?

```
# Plot shipment mode
sns.set_theme(style="whitegrid")
sns.countplot(superstore['Ship Mode'])
plt.title("Ship Mode")
plt.show()
```



Plot showing no. of orders using different Ship Mode

Fig – 27

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.

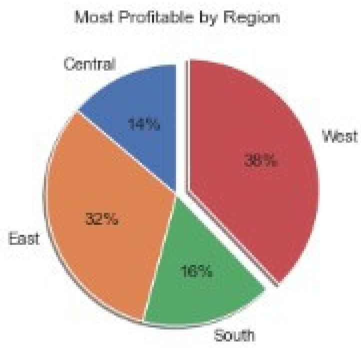
6. Which Region is the Most Profitable?

```
region_analysis = pd.DataFrame(superstore.groupby(['Region'])
['Profit'].sum().reset_index())
region_analysis
```

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

Fig – 28

```
# Plot Pie Chart
explode = [0, 0, 0, 0.1]

plt.pie(region_analysis.Profit, labels = region_analysis.Region,
startangle = 90, autopct = "%1.0f%%", explode = explode, shadow =
True)
plt.title("Most Profitable by Region")

plt.show()
```



Pie chart showing percentage of sales by Region

East and West region are most profitable.

Fig – 29

7. Which City has the Highest Number of Sales?

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

	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
Abilene	1.392	2

Bottom 10 cities with lowest sales

Fig – 31

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

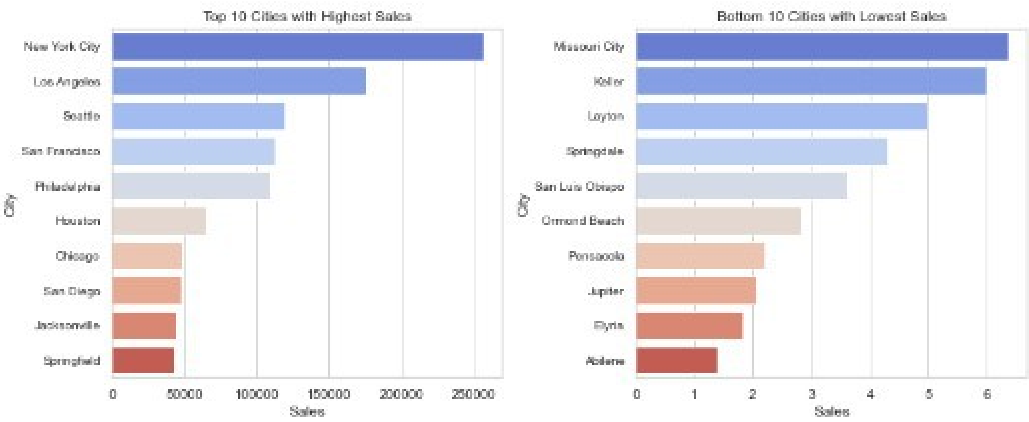
	Sales	Quantity
City		
New York City	256319.0410	3413
Los Angeles	175831.9010	2876
Seattle	119460.2820	1578
San Francisco	112577.1720	1920
Philadelphia	109061.4610	1978
Houston	64441.2564	1460
Chicago	48535.9770	1129
San Diego	47521.0290	670
Jacksonville	44713.1830	429
Springfield	43054.3420	649

Top 10 cities with highest sales

Fig – 30

```
# Set for grouped plots - figure with a 1x2 grid of Axes
figure, axis = plt.subplots(1, 2, figsize=(12, 5))

sns.set_theme(style="whitegrid")

top10c = sns.barplot(data = top10, y = top10.index, x = top10.Sales,
palette = "coolwarm", 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()
```

Fig – 31



L: Top 10 cities with highest sales | R: Bottom 10 cities with lowest sales

Fig – 32

There is a huge disparity between the cities with highest sales and lowest sales. Marketing strategy has to target the top 10 cities.

<u>Strategic Recommendation</u>

Here, we revisit our business questions and share with you our high-level business recommendations.

<u>Business Questions</u>

Which Category is Best Selling and Most Profitable?

What are the Best Selling and Most Profitable Sub-Category?

Which is the Top Selling Sub-Category?

Which Customer Segment is Most Profitable?

Which is the Preferred Ship Mode?

Which Region is the Most Profitable?

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.

Note:
As for thos who are not Programmer . a) You can employ anyone to help you to install Python in your system(either PC or Laptop) and as for programmer who can program in Python , its straight forward.

1—www.python.org(You can download python)

2- Install those Libraries that necessary

3 – Install Anaconda(www.anaconda.com)

4 – Follows the program as its stated.

- If there is error Google and find solution, as at the time I sent this pdf no error whatsoever.

Happy coding.

Happy coding