

Project Title:

Retail Revelations: Navigating Superstore Insights

In [1]: *# Importing the Libraries*

```
In [2]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

#visualization
import seaborn as sns
```

In [3]: *# Importing the dataset*

```
In [4]: df = pd.read_csv("Analysis of Super Store - DA.csv")
```

In [5]: *# Printing the dataset (First Five Rows)*

In [6]: df.head()

Out[6]:

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

```
In [44]: # Printing the dataset (Last Five Rows)
df.tail()
```

Out[44]:

	Ship Mode	Segment	Country	City	State	Region	Category	Sub-Category	
9989	Second Class	Consumer	United States	Miami	Florida	South	Furniture	Furnishings	
9990	Standard Class	Consumer	United States	Costa Mesa	California	West	Furniture	Furnishings	
9991	Standard Class	Consumer	United States	Costa Mesa	California	West	Technology	Phones	2
9992	Standard Class	Consumer	United States	Costa Mesa	California	West	Office Supplies	Paper	
9993	Second Class	Consumer	United States	Westminster	California	West	Office Supplies	Appliances	2

```
In [9]: # Shape (Rows, Columns)
```

```
In [10]: df.shape
```

Out[10]: (9994, 13)

```
In [11]: # Statistical summary of the dataset
```

```
In [12]: df.describe()
```

Out[12]:

	Postal Code	Sales	Quantity	Discount	Profit
count	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000
mean	55190.379428	229.858001	3.789574	0.156203	28.656896
std	32063.693350	623.245101	2.225110	0.206452	234.260108
min	1040.000000	0.444000	1.000000	0.000000	-6599.978000
25%	23223.000000	17.280000	2.000000	0.000000	1.728750
50%	56430.500000	54.490000	3.000000	0.200000	8.666500
75%	90008.000000	209.940000	5.000000	0.200000	29.364000
max	99301.000000	22638.480000	14.000000	0.800000	8399.976000

```
In [45]: # Finding whether they contain null values or not
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 12 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   Region          9994 non-null   object  
 6   Category        9994 non-null   object  
 7   Sub-Category    9994 non-null   object  
 8   Sales           9994 non-null   float64  
 9   Quantity        9994 non-null   int64    
10  Discount        9994 non-null   float64  
11  Profit          9994 non-null   float64  
dtypes: float64(3), int64(1), object(8)
memory usage: 937.1+ KB
```

```
In [15]: # deleting the unnecessary column
```

```
In [16]: df.drop(columns = ["Postal Code"], inplace = True)
```

```
In [17]: # finding unique values in given columns
```

```
In [18]: print(df["Ship Mode"].unique())

['Second Class' 'Standard Class' 'First Class' 'Same Day']
```

```
In [19]: print(df["Segment"].unique())

['Consumer' 'Corporate' 'Home Office']
```

```
In [20]: print(df["Country"].unique())

['United States']
```

```
In [21]: print(df["Category"].unique())

['Furniture' 'Office Supplies' 'Technology']
```

```
In [22]: print(df["Region"].unique())

['South' 'West' 'Central' 'East']
```

```
In [46]: # Number of unique users and items in the dataset
df.nunique()
```

```
Out[46]: Ship Mode      4
         Segment      3
         Country      1
         City       531
         State       49
         Region       4
         Category      3
         Sub-Category  17
         Sales      5825
         Quantity     14
         Discount     12
         Profit     6170
         dtype: int64
```

```
In [47]: # Check for duplicates
df.duplicated().sum()
```

```
Out[47]: 63
```

```
In [27]: # check for missing values
df.isnull().sum()
```

```
Out[27]: Ship Mode      0
         Segment      0
         Country      0
         City         0
         State        0
         Region       0
         Category     0
         Sub-Category  0
         Sales        0
         Quantity     0
         Discount     0
         Profit       0
         dtype: int64
```

```
In [28]: df.columns
```

```
Out[28]: Index(['Ship Mode', 'Segment', 'Country', 'City', 'State', 'Region',
                'Category', 'Sub-Category', 'Sales', 'Quantity', 'Discount', 'Profit'],
              dtype='object')
```

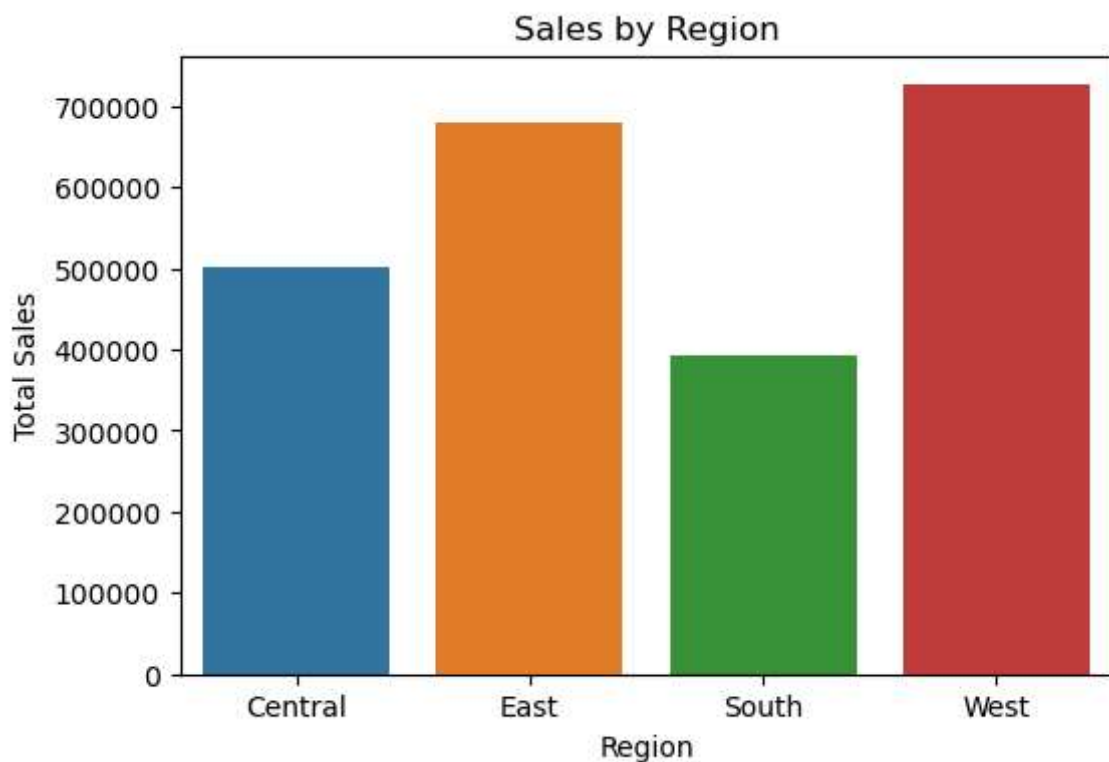
Data Visualization

1) sales analysis based on region (Bar Chart)

Visualize total sales for each region

```
In [29]: #Group data by region and calculate total sales
region_sales = df.groupby("Region")["Sales"].sum().reset_index()
```

```
In [30]: # creating a bar chart
plt.figure(figsize=(6, 4))
sns.barplot(x="Region", y="Sales", data= region_sales)
plt.xlabel("Region")
plt.ylabel("Total Sales")
plt.title("Sales by Region")
#plt.xticks(rotation=45)
plt.show()
```



```
In [31]: #Insights:
```

1. Western region is the top performer with ~700,000 in sales.
2. Southern region follows with ~400,000 in sales.

Improvement Areas:

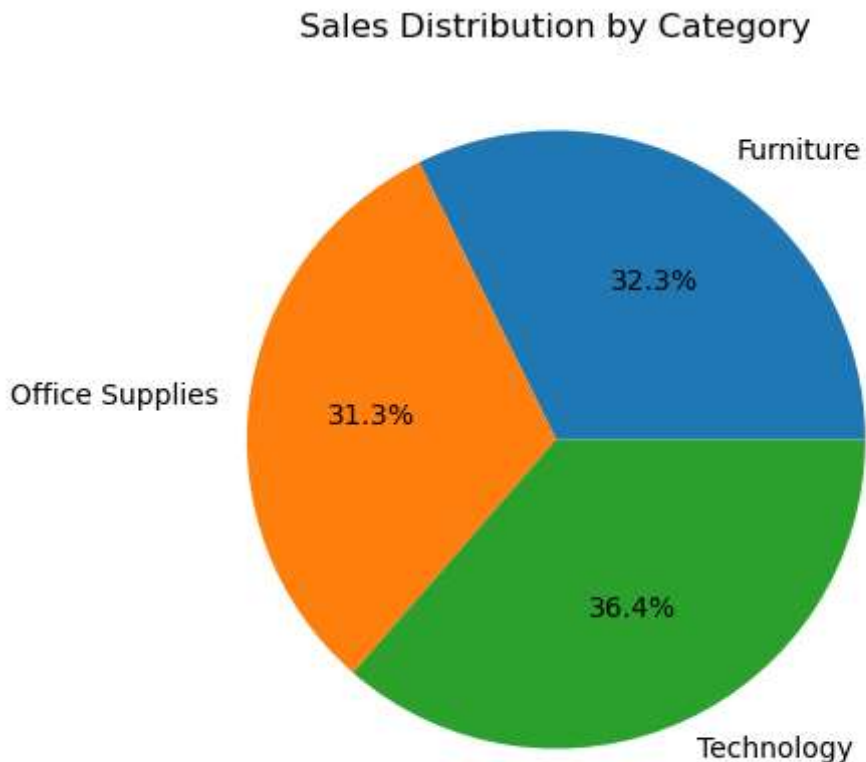
- Analyze reasons behind the sales gap between the Western and Southern regions.
- Explore marketing and sales strategies to bridge the gap and potentially boost sales in other regions.

2) Category-wise Sales (Pie Chart)

Pie chart: show the distribution of sales across different product categories

```
In [43]: #Group data by category and calculate total sales
category_sales = df.groupby("Category")["Sales"].sum().reset_index()
```

```
In [33]: plt.figure(figsize=(5, 5))
plt.pie(category_sales["Sales"], labels=category_sales["Category"],
        autopct="%1.1f%%")
plt.title("Sales Distribution by Category")
plt.show()
```



```
In [34]: #Insights:
```

1. Technology leads with 36.4% of total sales.
2. Furniture (32.3%) and Office Supplies (31.3%) follow.

Improvement Areas:

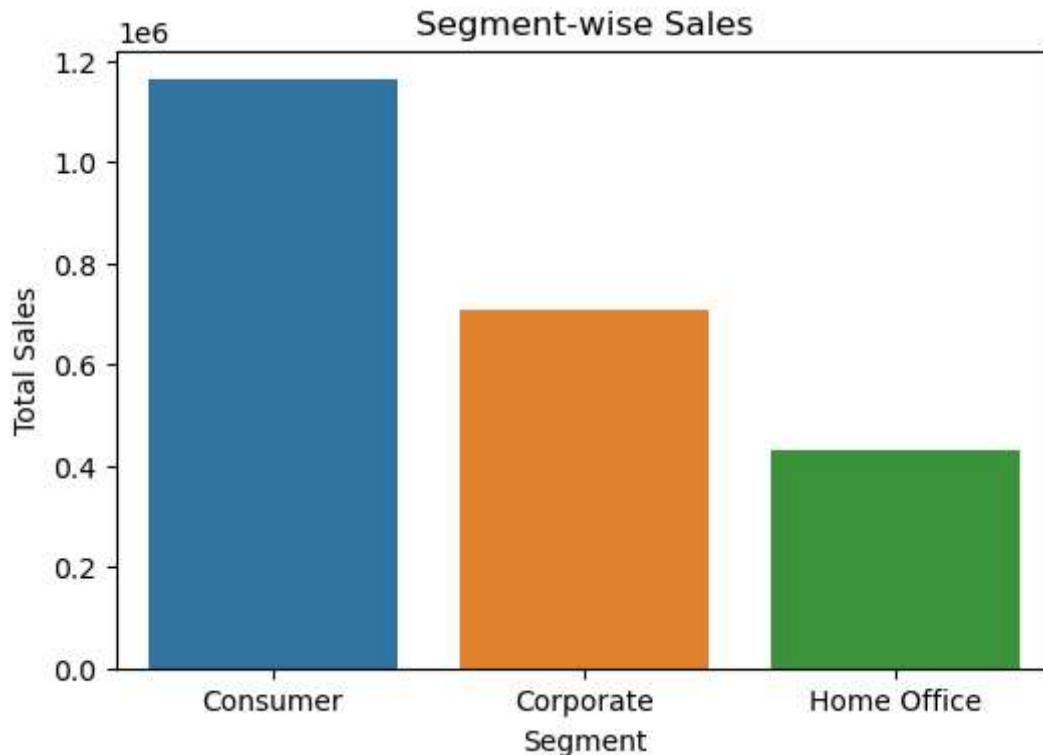
- Consider expanding product offerings within the Technology category based on customer demand and market trends.
- Evaluate the profitability of each category (considering both sales and profit margins) to optimize product focus.

3) Segment-wise Sales (Bar Chart)

Visualize the count of sales for each segment

Group data by segment and calculate sales count

```
In [35]: segment_sales = df.groupby("Segment")["Sales"].sum().reset_index()
plt.figure(figsize=(6, 4))
sns.barplot(x="Segment", y="Sales", data=segment_sales)
plt.xlabel("Segment")
plt.ylabel("Total Sales")
plt.title("Segment-wise Sales")
#plt.xticks(rotation=45)
plt.show()
```



```
In [36]: #Insights:
```

- 1.Consumer segment has the highest sales count
- 2.Corporate and Home Office segments follow with ~0.7 and ~0.4 orders, respectively

Improvement Areas:

- Investigate reasons behind the lower sales in Corporate and Home Office segments.
- Develop targeted marketing campaigns and promotional strategies to attract and retain customers in these segments.

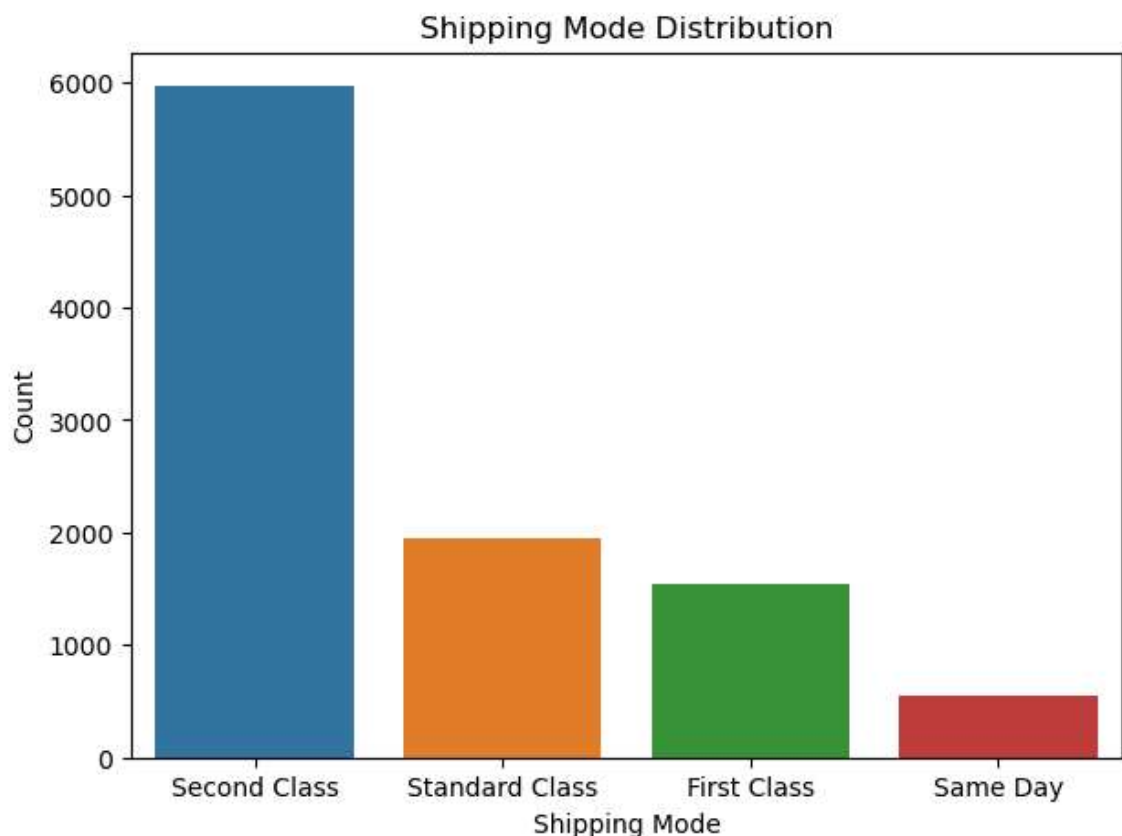
4) Shipping Mode Distribution (Bar Chart)

Display the count of each shipping mode

```
In [37]: # Get unique shipping modes
shipping_modes = df["Ship Mode"].unique()

# Count occurrences of each shipping mode
shipping_mode_counts = df["Ship Mode"].value_counts()

# Create a bar chart
plt.figure(figsize=(7, 5))
sns.barplot(x=shipping_modes, y=shipping_mode_counts)
plt.xlabel("Shipping Mode")
plt.ylabel("Count")
plt.title("Shipping Mode Distribution")
#plt.xticks(rotation=45)
plt.show()
```



```
In [38]: #Insights:
```

1. Second Class Shipping is the most popular option counts (6000)
2. Standard Class counts (2000), First Class counts (1500) and Same Day counts are used less frequently counts(1000)

Improvement Areas:

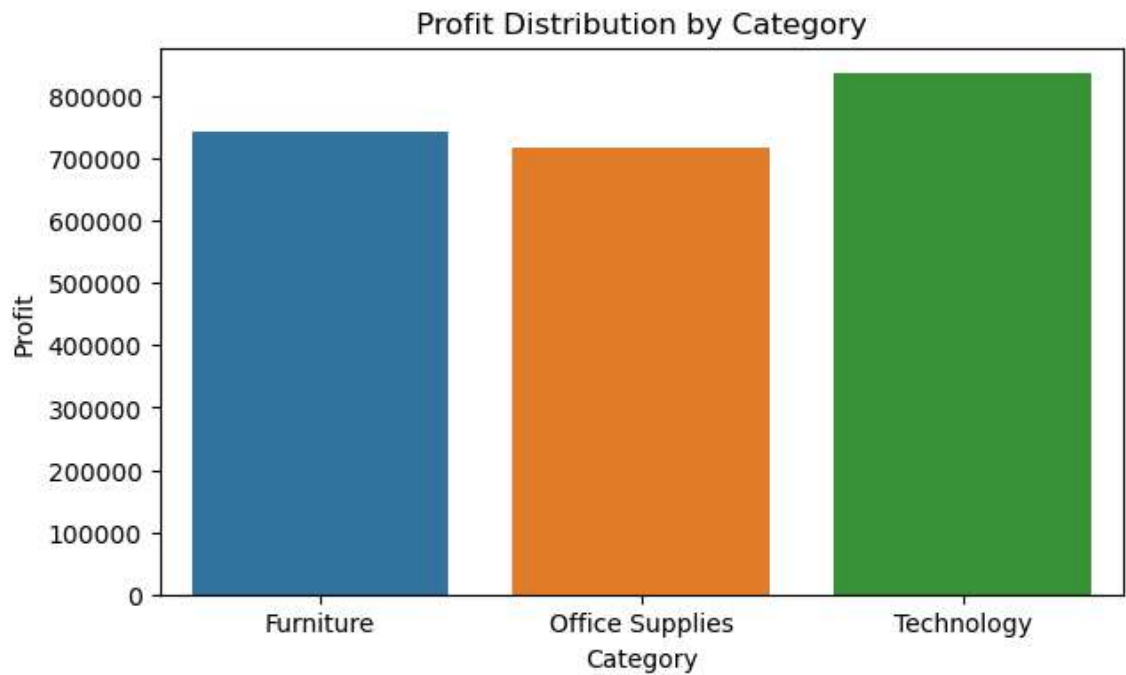
- Analyze customer preferences and needs for different shipping options.
- Explore offering competitive pricing or bundled promotions for alternative shipping modes (Express and Same Day) to encourage their use.

5) Profit Analysis (Bar/Line Chart)

Analyze profit distribution by category (Bar Plot)

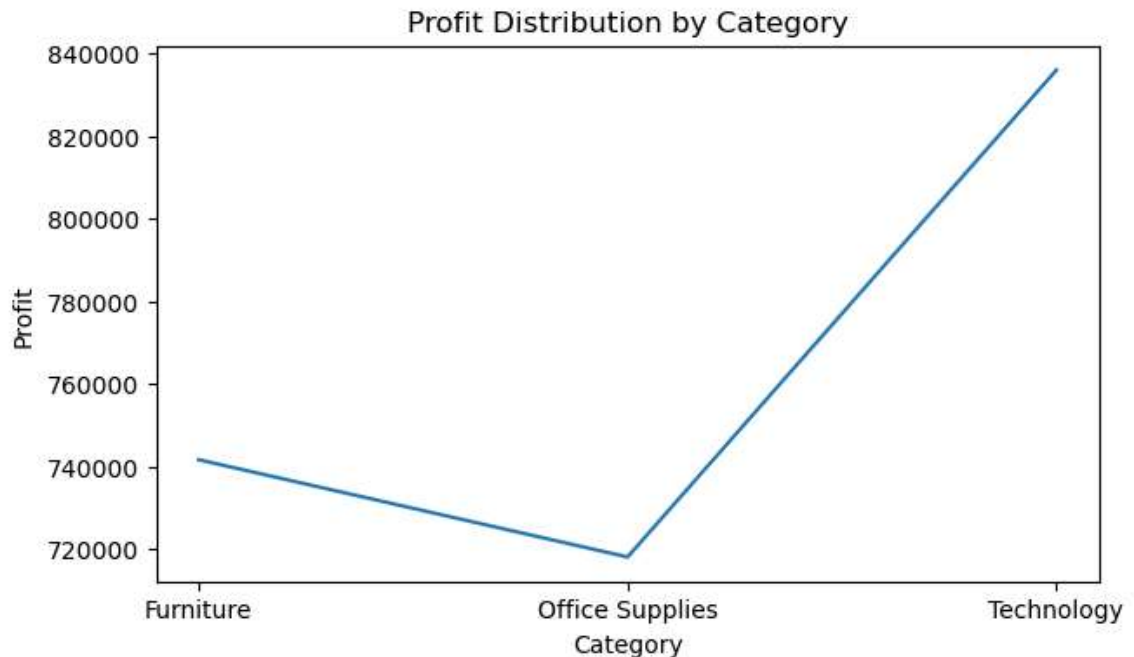

```
In [39]: # Calculate profit by category (assuming "Profit" and "Discount" exist)
df["Profit"] = df["Sales"] - df["Discount"]
category_profit = df.groupby("Category")["Profit"].sum().reset_index()

# Bar chart
plt.figure(figsize=(7, 4))
sns.barplot(x="Category", y="Profit", data=category_profit)
plt.xlabel("Category")
plt.ylabel("Profit")
plt.title("Profit Distribution by Category")
#plt.xticks(rotation=45)
plt.show()
```



Analyze profit distribution by category (Line Plot)

```
In [40]: # Line chart
plt.figure(figsize=(7, 4))
sns.lineplot(x="Category", y="Profit", data=category_profit)
plt.xlabel("Category")
plt.ylabel("Profit")
plt.title("Profit Distribution by Category")
#plt.xticks(rotation=45)
plt.show()
```



```
In [41]: #Insights:
```

1. "Technology" has the highest overall profit.
2. "Furniture" (740,000) and Office Supplies (720,000) follow.

Improvement Areas:

- Investigate cost factors influencing profit margins across categories (e.g., production costs, discounts).
- Optimize pricing strategies based on profitability and market competition to maximize overall profit.

Conclusion:

Analysis reveals the Western region and Technology category as top performers in sales, while the Consumer segment leads in order count. Second Class reigns supreme, with potential for growth in alternative options. Technology leads in profit, but profitability analysis across categories is crucial.

Improvement opportunities lie in:

- . Bridging regional sales gaps
- . Optimizing product offerings and focus
- . Attracting customers in specific segments
- . Exploring alternative shipping strategies
- . Maximizing profit through cost analysis and pricing optimization

. These insights provide a springboard for further exploration and strategic decision-making to enhance overall retail performance.

THANK YOU!

In []: