

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

df=pd.read_csv("/Users/gauravjain/Desktop/PBA/Case 3.csv")
```

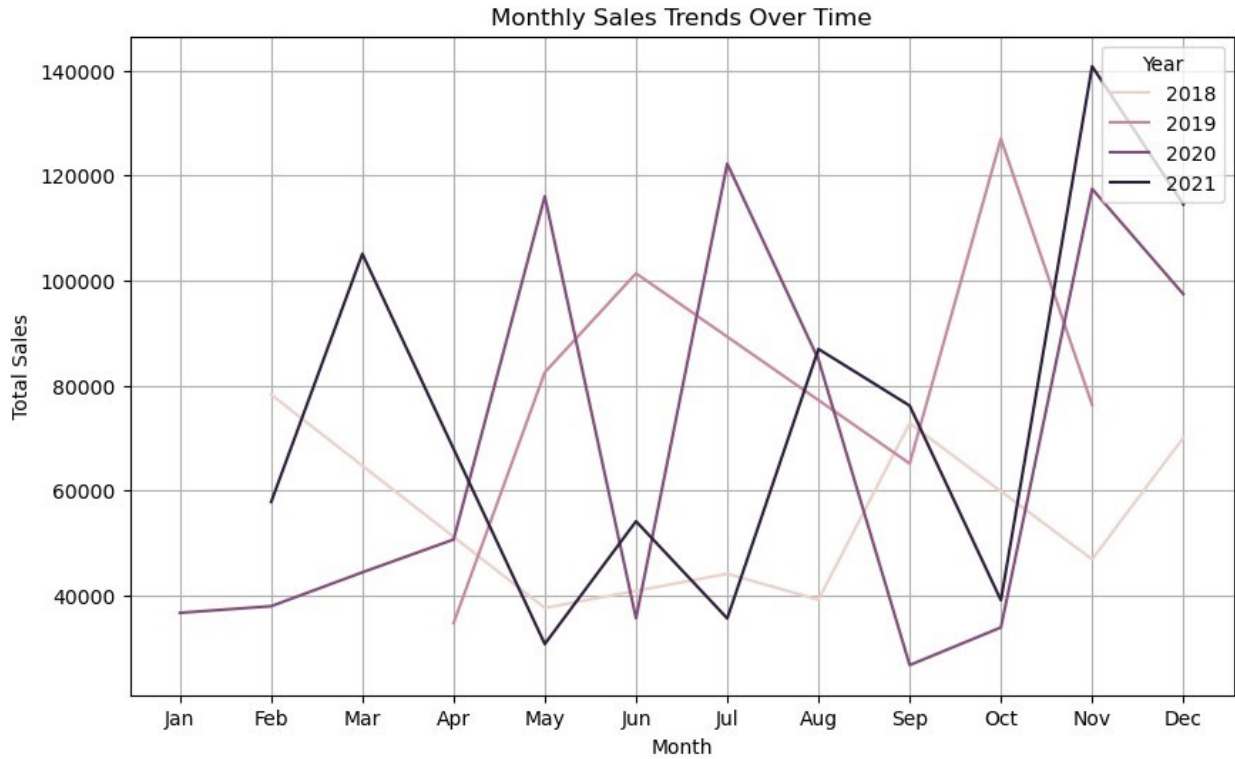
MARKET AND SALES PERFORMANCE ANALYSIS

```
df['Order_Date'] = pd.to_datetime(df['Order_Date'])

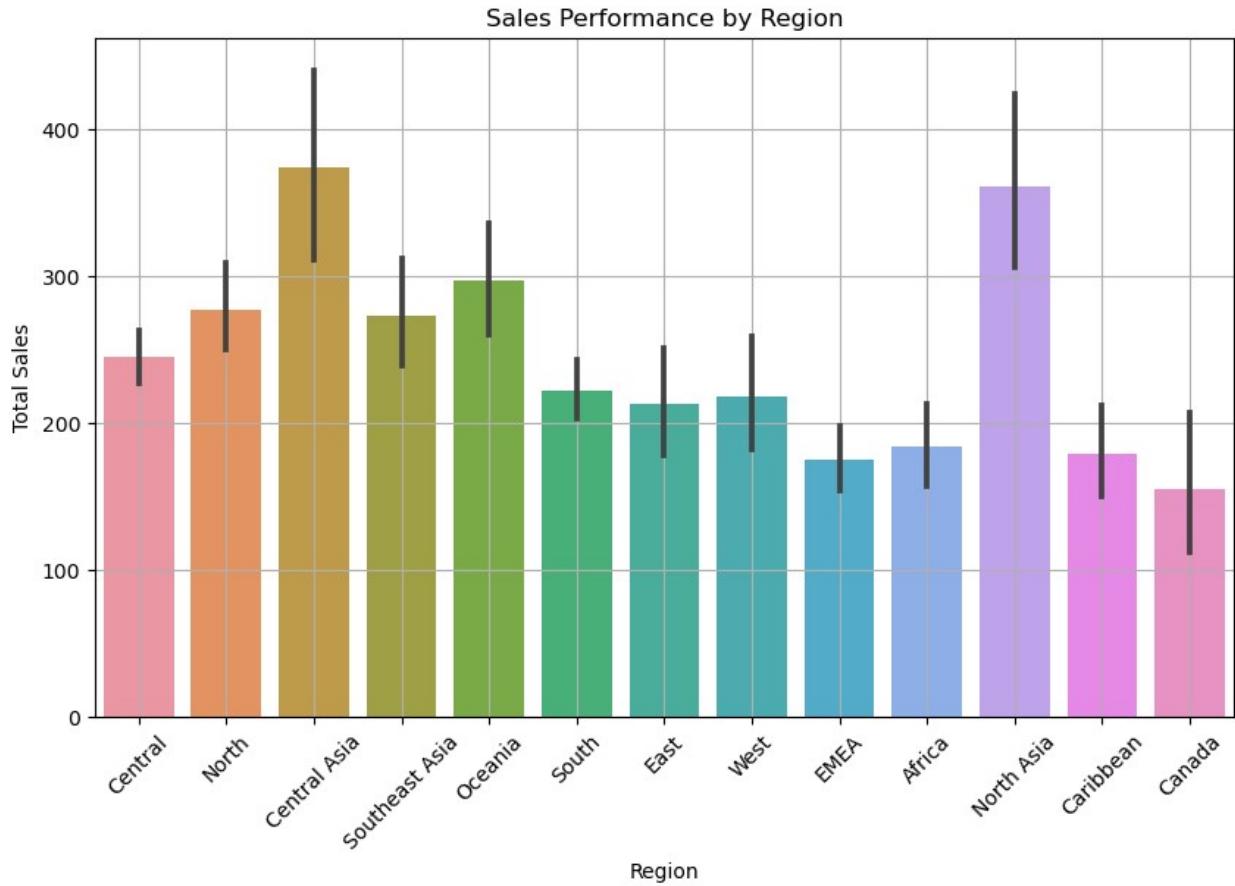
df['Year'] = df['Order_Date'].dt.year
df['Month'] = df['Order_Date'].dt.month

monthly_sales = df.groupby(['Year', 'Month'])
['Sales'].sum().reset_index()

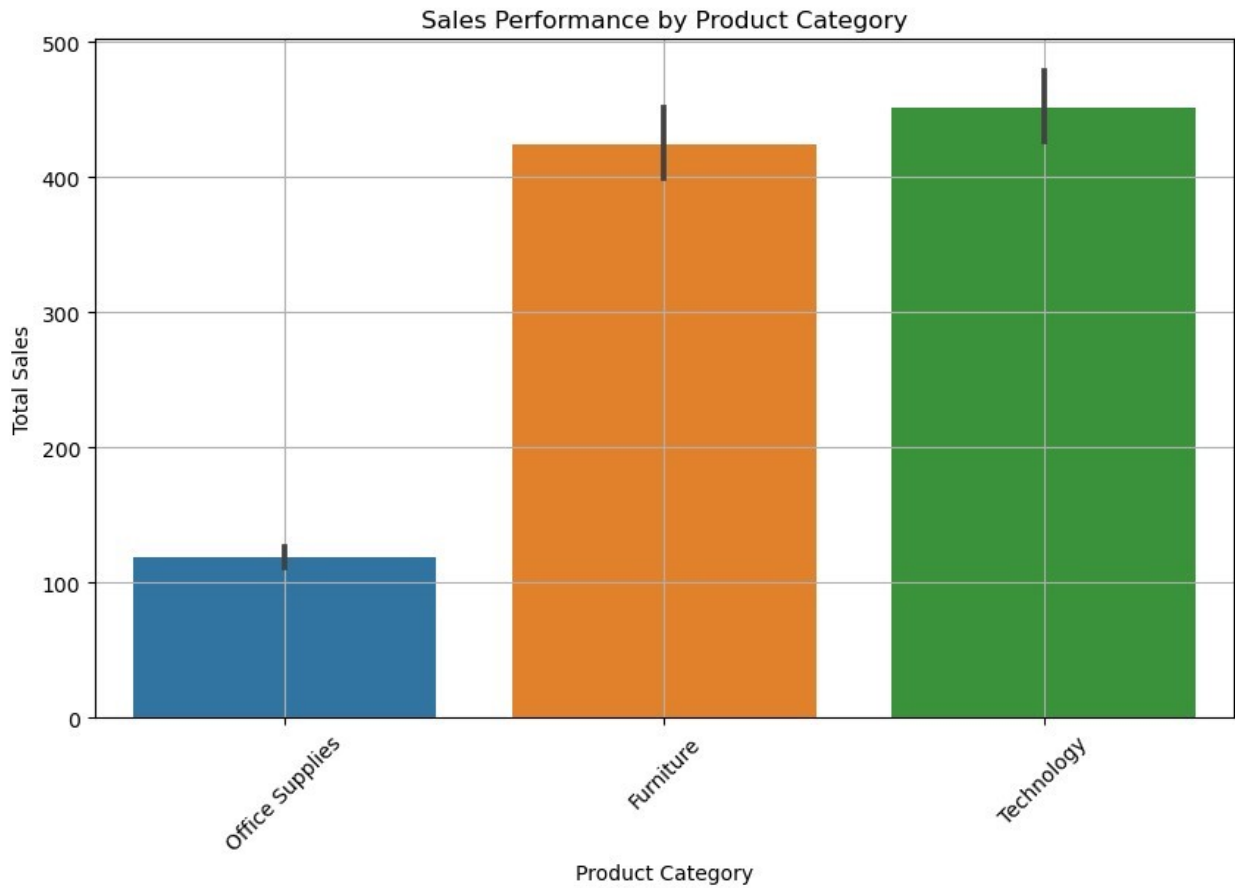
# Plotting sales trends over time
plt.figure(figsize=(10, 6))
sns.lineplot(x='Month', y='Sales', hue='Year', data=monthly_sales)
plt.title('Monthly Sales Trends Over Time')
plt.xlabel('Month')
plt.ylabel('Total Sales')
plt.xticks(range(1, 13), ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec'])
plt.legend(title='Year', loc='upper right')
plt.grid(True)
plt.show()
```



```
# Analyzing sales performance across different regions
plt.figure(figsize=(10, 6))
sns.barplot(x='Region', y='Sales', data=df)
plt.title('Sales Performance by Region')
plt.xlabel('Region') plt.ylabel('Total
Sales') plt.xticks(rotation=45)
plt.grid(True) plt.show()
```



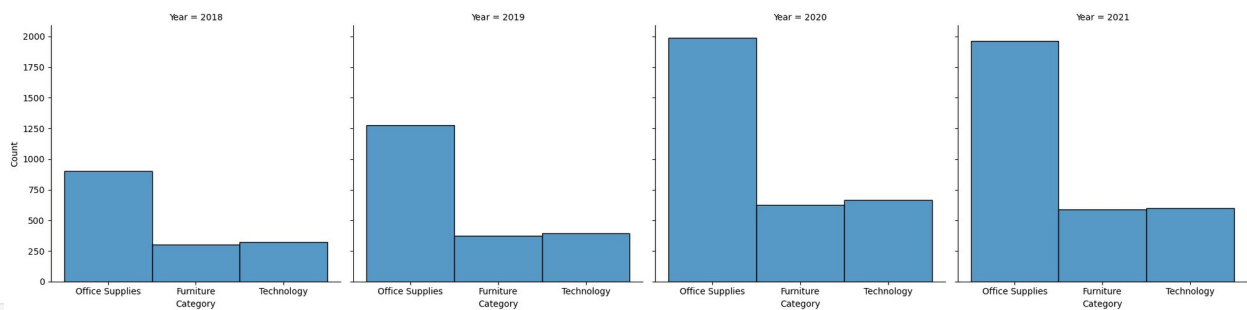
```
# Analyzing sales performance across different product categories
plt.figure(figsize=(10, 6))
sns.barplot(x='Category', y='Sales', data=df)
plt.title('Sales Performance by Product Category')
plt.xlabel('Product Category') plt.ylabel('Total
Sales') plt.xticks(rotation=45) plt.grid(True)
plt.show()
```



```
sns.displot(
    data=df,
    x="Category", col="Year"
)
```

```
/Users/gauravjain/anaconda3/lib/python3.11/site-packages/seaborn/
axisgrid.py:118: UserWarning: The figure layout has changed to tight
self._figure.tight_layout(*args, **kwargs)
```

```
<seaborn.axisgrid.FacetGrid at 0x146d600d0>
```



```
crosstab = pd.crosstab(
    index=df['Region'],
    columns=df['Sub_Category'],
```

```

values=df['Quantity'],
aggfunc='sum'
)

```

```

print(crosstab)

```

Sub_Category Chairs \ Region	Accessories	Appliances	Art	Binders	Bookcases
Africa 88.0	117.0	62.0	257.0	280.0	80.0
Canada 5.0	2.0	7.0	20.0	29.0	8.0
Caribbean 95.0	35.0	38.0	55.0	133.0	71.0
Central 488.0	506.0	240.0	907.0	1001.0	305.0
Central Asia 156.0	89.0	26.0	90.0	173.0	150.0
EMEA 127.0	137.0	78.0	293.0	274.0	75.0
East 125.0	116.0	97.0	171.0	280.0	30.0
North 241.0	174.0	134.0	443.0	469.0	227.0
North Asia 120.0	119.0	37.0	108.0	150.0	101.0
Oceania 218.0	216.0	93.0	121.0	204.0	101.0
South 316.0	266.0	211.0	393.0	651.0	238.0
Southeast Asia 235.0	122.0	50.0	167.0	223.0	107.0
West 170.0	235.0	92.0	196.0	358.0	79.0

Sub_Category Machines \ Region	Copiers	Envelopes	Fasteners	Furnishings	Labels
Africa 96.0	102.0	82.0	131.0	130.0	92.0
Canada 8.0	8.0	5.0	5.0	1.0	19.0
Caribbean 16.0	63.0	85.0	119.0	70.0	90.0
Central 215.0	295.0	327.0	341.0	449.0	334.0

Central Asia	87.0	59.0	81.0	109.0	91.0
36.0					
EMEA	96.0	96.0	83.0	79.0	108.0
136.0					
East	17.0	58.0	41.0	208.0	68.0
50.0					
North	185.0	203.0	185.0	186.0	225.0
99.0					
North Asia	63.0	117.0	95.0	110.0	97.0
69.0					
Oceania	214.0	121.0	139.0	142.0	235.0
72.0					
South	165.0	239.0	210.0	384.0	209.0
114.0					
Southeast Asia	146.0	151.0	143.0	184.0	159.0
118.0					
West	12.0	76.0	35.0	281.0	98.0
22.0					

Sub_Category	Paper	Phones	Storage	Supplies	Tables	Region
Africa	107.0	105.0	261.0	102.0	19.0	
Canada	16.0	7.0	32.0	19.0	NaN	
Caribbean	80.0	78.0	87.0	99.0	7.0	Central
461.0 431.0	769.0	294.0	91.0			
Central Asia	125.0	135.0	110.0	95.0	27.0	EMEA
103.0 125.0	316.0	130.0	23.0			
East	265.0	191.0	171.0	26.0	46.0	
North	158.0	259.0	360.0	151.0	55.0	
North Asia	95.0	104.0	136.0	140.0	53.0	
Oceania	134.0	160.0	189.0	202.0	29.0	South
404.0 285.0	415.0	270.0	88.0			
Southeast Asia	125.0	224.0	167.0	157.0	33.0	
West	349.0	163.0	256.0	53.0	84.0	

Customer behavior and segment analysis

```
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans

rfm_df = df.groupby('Customer_ID').agg({
    'Order_Date': lambda x: (df['Order_Date'].max() - x.max()).days,
    'Order_ID': 'nunique',
    'Sales': 'sum'
}).reset_index()

rfm_df.columns = ['Customer_ID', 'Recency', 'Frequency', 'Monetary']
```

```

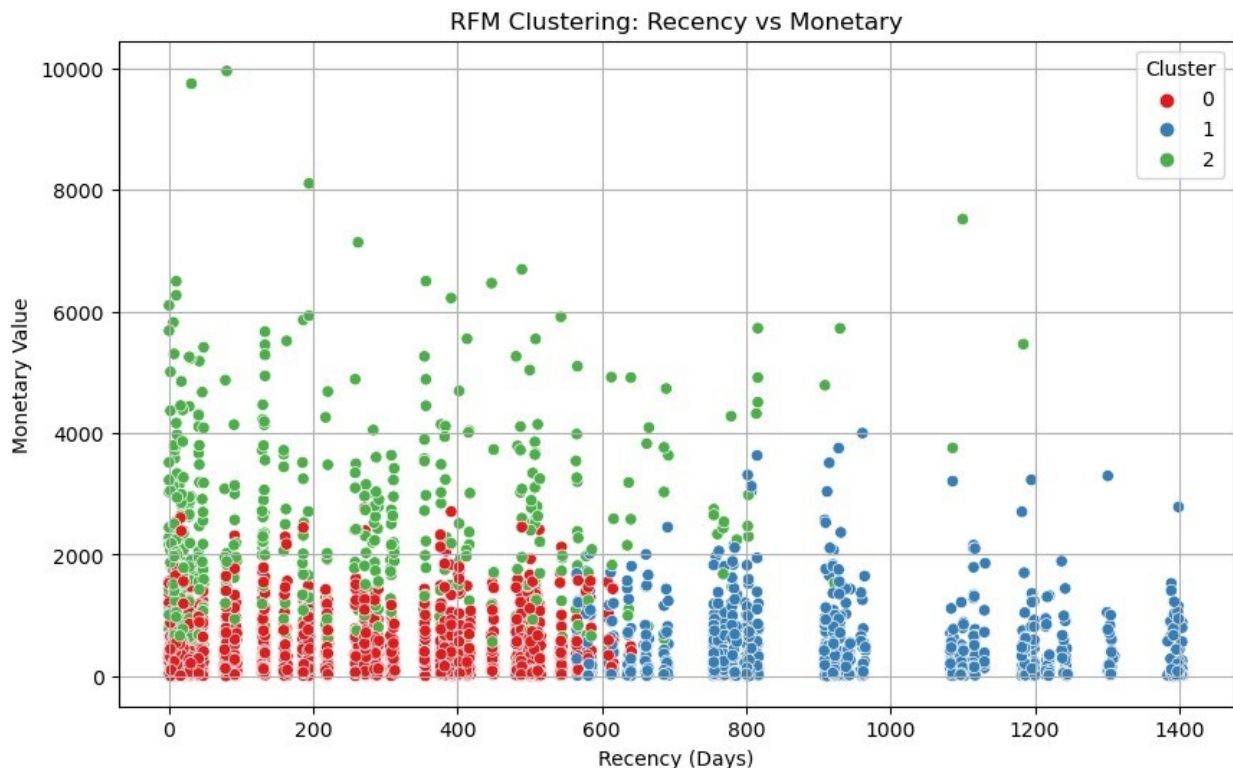
scaler = StandardScaler()
rfm_scaled = scaler.fit_transform(rfm_df[['Recency', 'Frequency',
'Monetary']])

kmeans = KMeans(n_clusters=3, random_state=42)
rfm_df['Cluster'] = kmeans.fit_predict(rfm_scaled)

/Users/gauravjain/anaconda3/lib/python3.11/site-packages/sklearn/
cluster/_kmeans.py:1412: FutureWarning: The default value of `n_init`
will change from 10 to 'auto' in 1.4. Set the value of `n_init`
explicitly to suppress the warning
    super()._check_params_vs_input(X, default_n_init=10)

plt.figure(figsize=(10, 6))
sns.scatterplot(x='Recency', y='Monetary', hue='Cluster', data=rfm_df,
palette='Set1')
plt.title('RFM Clustering: Recency vs Monetary')
plt.xlabel('Recency (Days)')
plt.ylabel('Monetary Value')
plt.legend(title='Cluster', loc='upper right')
plt.grid(True) plt.show()

```

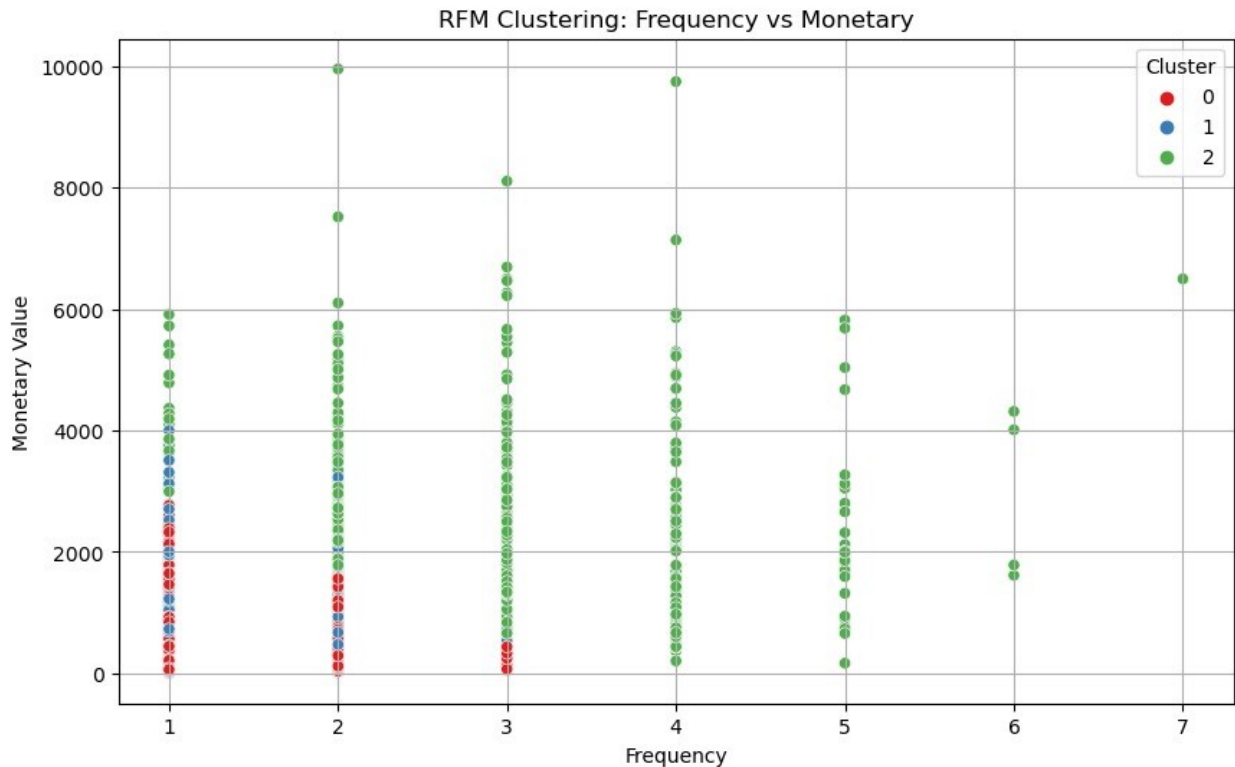


```

plt.figure(figsize=(10, 6))
sns.scatterplot(x='Frequency', y='Monetary', hue='Cluster',
data=rfm_df, palette='Set1')

```

```
plt.title('RFM Clustering: Frequency vs Monetary')
plt.xlabel('Frequency') plt.ylabel('Monetary
Value')
plt.legend(title='Cluster', loc='upper right')
plt.grid(True) plt.show()
```



Product Portfolio Optimization:

```
product_performance = df.groupby('Category').agg({
    'Sales': 'sum',          # Total sales volume
    'Profit': 'sum',         # Total profit
    'Quantity': 'sum'       # Total quantity sold
})

# Additional calculations for profit margin and growth rate
product_performance['Profit_Margin'] = (product_performance['Profit']
/ product_performance['Sales']) * 100
product_performance['Growth_Rate'] = (product_performance['Sales'] -
product_performance['Sales'].shift(1)) /
product_performance['Sales'].shift(1) * 100

product_performance = product_performance.sort_values(by='Sales',
ascending=False)
```



```
print("Product Performance Metrics:")
print(product_performance)
```

```
Product Performance Metrics:
          Sales          Profit  Quantity
Profit_Margin \
Category
Technology      894740.04268  122640.80808      6905      13.706865
Furniture        800346.78560   61554.36390      6844       7.690962
Office Supplies  725932.20580   94100.62570     21092     12.962729
          Growth_Rate
Technology      23.253940
Furniture              NaN
Office Supplies   -9.297792
```

Supply Chain and Shipping Efficiency

```
# Calculate shipping costs per order
df['Shipping_Cost_Per_Order'] = df['Shipping_Cost'] / df['Quantity']

# Grouping by shipping mode and calculating average shipping cost per order
shipping_mode_efficiency = df.groupby('Ship_Mode')
['Shipping_Cost_Per_Order'].mean().sort_values(ascending=False)
df["Ship_Mode"].value_counts()
Ship_Mode
Standard Class      6024
Second Class        2002
First Class         1412
Same Day            561
Name: count, dtype: int64
# Get the frequency count of ship modes ship_mode_counts
= df['Ship_Mode'].value_counts()

# Convert the frequency count to a DataFrame
ship_mode_counts_df = ship_mode_counts.to_frame(name='Count')

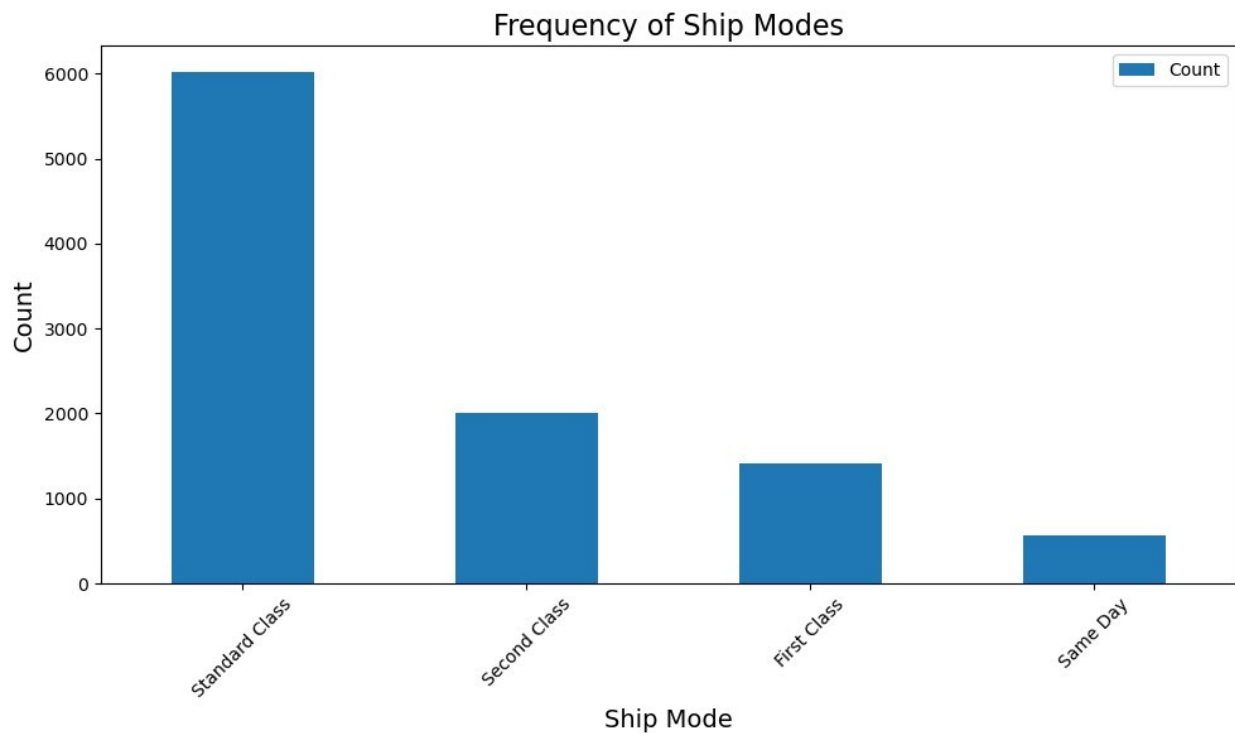
# Plot the bar chart
ship_mode_counts_df.plot(kind='bar', figsize=(10, 6))

# Set the title and axis labels
plt.title('Frequency of Ship Modes', fontsize=16)
```

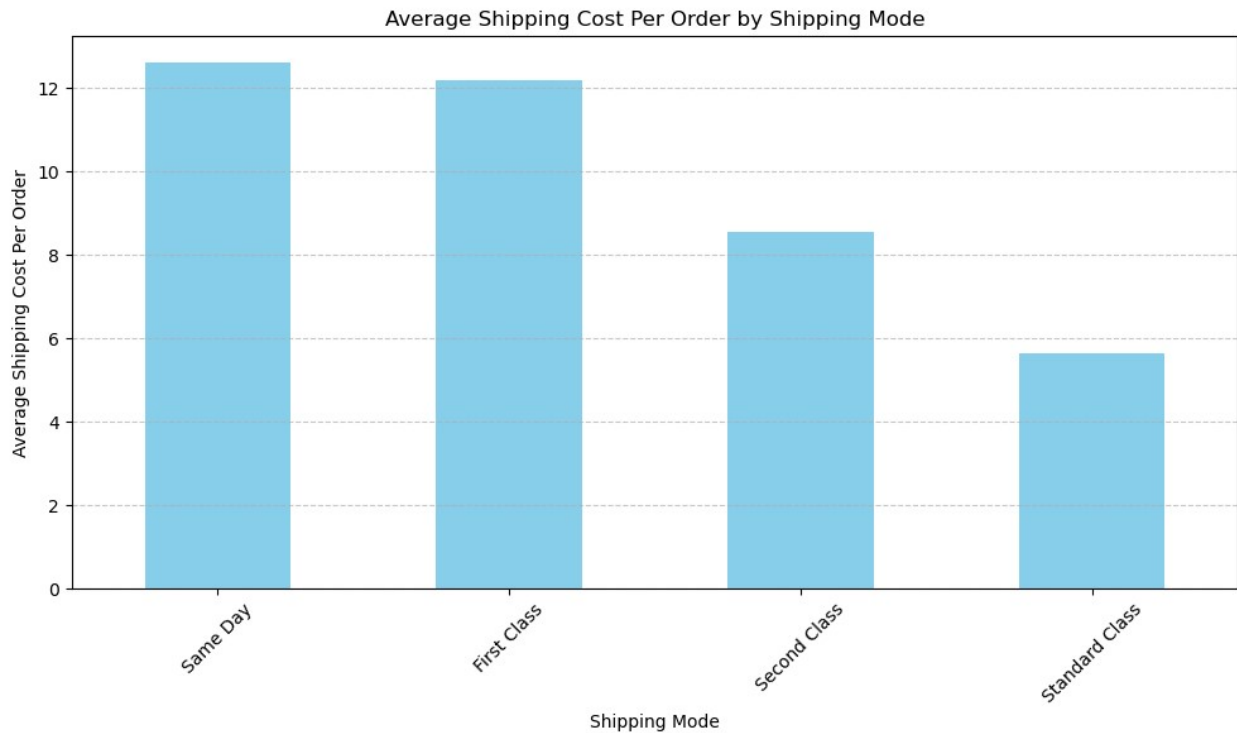
```
plt.xlabel('Ship Mode', fontsize=14)
plt.ylabel('Count', fontsize=14)

# Rotate the x-axis labels for better readability
plt.xticks(rotation=45)

# Show the plot
plt.tight_layout()
plt.show()
```



```
# Visualize shipping costs per order by shipping mode
plt.figure(figsize=(10, 6))
shipping_mode_efficiency.plot(kind='bar', color='skyblue')
plt.title('Average Shipping Cost Per Order by Shipping Mode')
plt.xlabel('Shipping Mode')
plt.ylabel('Average Shipping Cost Per Order')
plt.xticks(rotation=45)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.tight_layout() plt.show()
```



```
df_shippingcost=df.groupby("Sub_Category")
["Shipping_Cost"].mean().round(2)
print(df_shippingcost)
Sub_Category
```

```
Accessories      26.90
Appliances       58.41
Art              8.40
Binders          7.68
Bookcases        68.21
Chairs           44.66
Copiers          70.76
Envelopes        7.90
Fasteners        3.89
Furnishings      13.10
Labels           2.68
Machines         53.74
Paper            7.14
Phones           56.53
Storage          22.46
Supplies         9.65
Tables           93.73
```

```
Name: Shipping_Cost, dtype: float64
```

STRATEGIC RECOMMENDATION Strategic recommendations for competitive advantage can be derived from the analyses conducted on various aspects of Global Superstore's operations. Here's a general framework for proposing strategic initiatives based on the analyses:

1. Product Diversification:
 - Identify underperforming product categories or products with low growth rates.
 - Recommend diversification into new product lines or expansion of existing ones based on market trends and customer preferences.
 - Prioritize product development efforts towards high-demand categories or products with high-profit margins.
2. Targeted Marketing Campaigns:
 - Utilize customer segmentation analysis to tailor marketing strategies for different customer segments.
 - Implement targeted advertising campaigns through various channels, such as social media, email marketing, and personalized recommendations on the website.
 - Focus on promoting top-selling products or newly introduced product lines to drive sales growth.
3. Optimizing Shipping Logistics:
 - Analyze shipping modes and costs to identify inefficiencies and cost-saving opportunities.
 - Optimize the selection of shipping carriers and routes based on cost effectiveness and delivery speed.
 - Implement strategies to minimize shipping delays and improve order fulfillment accuracy.
4. Enhancing Customer Experiences:
 - Leverage customer behavior analysis to personalize the shopping experience and improve customer satisfaction.
 - Implement loyalty programs, discounts, and rewards to incentivize repeat purchases and increase customer retention.
 - Enhance the online shopping platform with user-friendly features, such as product recommendations, easy navigation, and secure payment options.
5. Continuous Improvement:
 - Establish a system for ongoing monitoring and evaluation of key performance indicators (KPIs) related to sales, customer satisfaction, and operational efficiency.
 - Encourage a culture of innovation and continuous improvement within the organization to adapt to changing market dynamics and customer preferences.
 - Foster collaboration between different departments, such as marketing, sales, and logistics, to align strategies and achieve common objectives.

By implementing these strategic initiatives, Global Superstore can gain a competitive advantage in the retail market by improving product offerings, enhancing customer experiences, and optimizing operational efficiency. These recommendations should be periodically reviewed and adjusted based on market feedback and performance metrics to ensure sustained growth and competitiveness.

