

Super Store Sales Analysis

- Data

Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	\
0	1	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520
1	2	CA-2016-152156	11/8/2016	11/11/2016	Second Class	CG-12520
2	3	CA-2016-138688	6/12/2016	6/16/2016	Second Class	DV-13045
3	4	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335
4	5	US-2015-108966	10/11/2015	10/18/2015	Standard Class	SO-20335

	Customer Name	Segment	Country	City	...	\
0	Claire Gite	Consumer	United States	Henderson	...	
1	Claire Gite	Consumer	United States	Henderson	...	
2	Darrin Van Huff	Corporate	United States	Los Angeles	...	
3	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	
4	Sean O'Donnell	Consumer	United States	Fort Lauderdale	...	

	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	

	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	

	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

[5 rows x 21 columns]

- **Data Info**

```
#Fetch information about dataset columns
data.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
```

- **Data Transformation**

- Fixed date format of order_date and ship_date into dd-MM-yyyy.
- Fetch Month and year from order_date.
- Calculate tenor between ship_date and order_date

```
#Order date and shipment date datatype is object convert into date/datetime
data["Order Date"] = pd.to_datetime(data["Order Date"],format='mixed')
data["Ship Date"] = pd.to_datetime(data["Ship Date"],format='mixed')
print('Datatype of Order date',data["Order Date"].dtype)
print('Datatype of Ship date',data["Ship Date"].dtype)
print('\n')
print(data["Order Date"].head())
print('\n')
print(data["Ship Date"].head())
```

```
Datatype of Order date datetime64[ns]
Datatype of Ship date datetime64[ns]
```

```
0   2016-11-08
1   2016-11-08
2   2016-06-12
3   2015-10-11
4   2015-10-11
Name: Order Date, dtype: datetime64[ns]
```

```
0   2016-11-11
1   2016-11-11
2   2016-06-16
3   2015-10-18
4   2015-10-18
Name: Ship Date, dtype: datetime64[ns]
```

```

#Convert Order date and ship date into dd-mm-yyyy
data["Tenor"] = data["Ship Date"] - data["Order Date"]
data["Order Month"] = data["Order Date"].dt.strftime('%b')
data["Order Year"] = data["Order Date"].dt.strftime('%Y')
data["Order Date"] = data["Order Date"].dt.strftime('%d-%m-%Y')
data["Ship Date"] = data["Ship Date"].dt.strftime('%d-%m-%Y')
print('\n')
print(data["Order Date"].head())
print('\n')
print(data["Ship Date"].head())
print('\n')
print(data["Order Month"].head())
print('\n')
print(data["Order Year"].head())
print('\n')
print(data["Tenor"].head())

```

```

0    08-11-2016
1    08-11-2016
2    12-06-2016
3    11-10-2015
4    11-10-2015
Name: Order Date, dtype: object

```

```

0    11-11-2016
1    11-11-2016
2    16-06-2016
3    18-10-2015
4    18-10-2015
Name: Ship Date, dtype: object

```

```

0    Nov
1    Nov
2    Jun
3    Oct
4    Oct
Name: Order Month, dtype: object

```

```

0    2016
1    2016
2    2016
3    2015
4    2015
Name: Order Year, dtype: object

```

```

0    3 days
1    3 days
2    4 days
3    7 days
4    7 days
Name: Tenor, dtype: timedelta64[ns]

```

- **Data Analysis and Data Visualization**

- **Overview**

```
#Overview of Sales
print('Overview of Sales :::')
print('Total Number of Orders: ',data["Order ID"].nunique())
print('Total Quantity Sold: ',data["Quantity"].sum())
print('Total Sales: ',data["Sales"].sum().round(2))
print('Customer Aquasition: ',data["Customer ID"].nunique())
agv_days = data["Tenor"].mean()
print('Average delivery days to ship the order',agv_days)
```

```
Overview of Sales :::
Total Number of Orders:  5009
Total Quantity Sold:  37873
Total Sales:  2297200.86
Customer Aquasition:  793
Average delivery days to ship the order 3 days 22:59:46.311787072
```

- **Total Sales by Year**

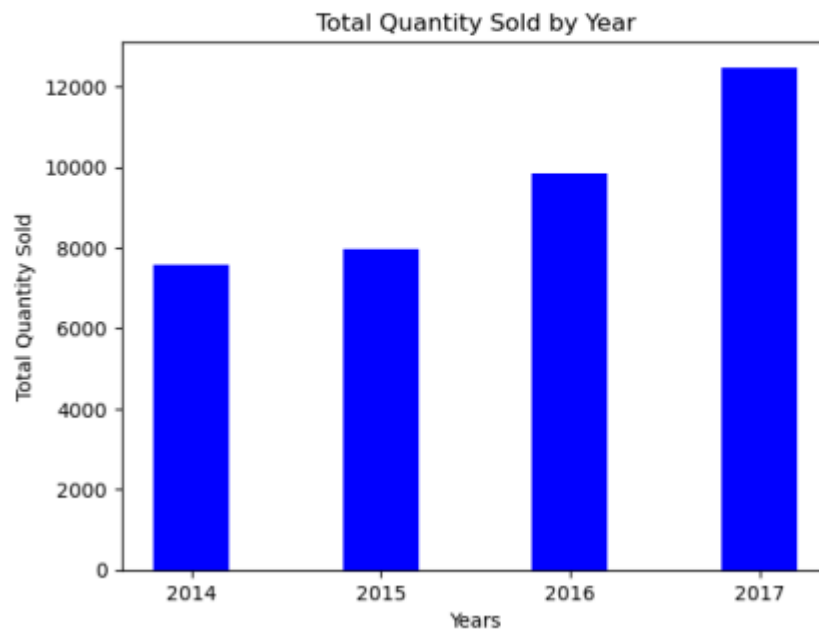
```
#Total Sales By Year
year_data = data[["Order Year","Quantity","Sales"]].copy()
year_data.head()
```

	Order Year	Quantity	Sales
0	2016	2	261.9600
1	2016	3	731.9400
2	2016	2	14.6200
3	2015	5	957.5775
4	2015	2	22.3680

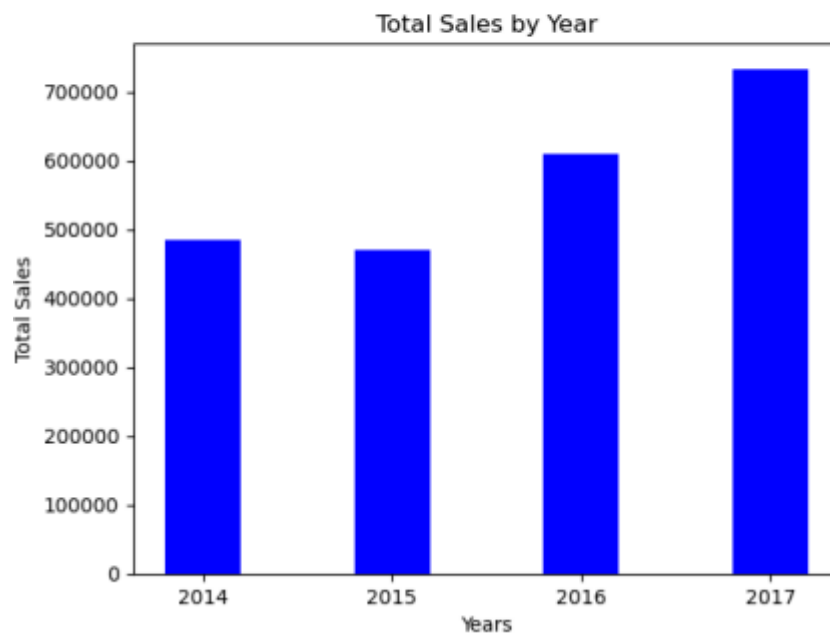
```
year_data = year_data.groupby("Order Year").sum().round(2)
year_data
```

	Order Year	Quantity	Sales
	2014	7581	484247.50
	2015	7979	470532.51
	2016	9837	609205.60
	2017	12476	733215.26

```
#Total Quantity Sold By Year
plt.bar(year_data["Order Year"],year_data["Quantity"],width = 0.4,color = "b")
plt.xlabel("Years")
plt.ylabel("Total Quantity Sold")
plt.title("Total Quantity Sold by Year")
plt.show()
```



```
# Total Sales by Year
plt.bar(year_data["Order Year"],year_data["Sales"],width = 0.4,color = "b")
plt.xlabel("Years")
plt.ylabel("Total Sales")
plt.title("Total Sales by Year")
plt.show()
```

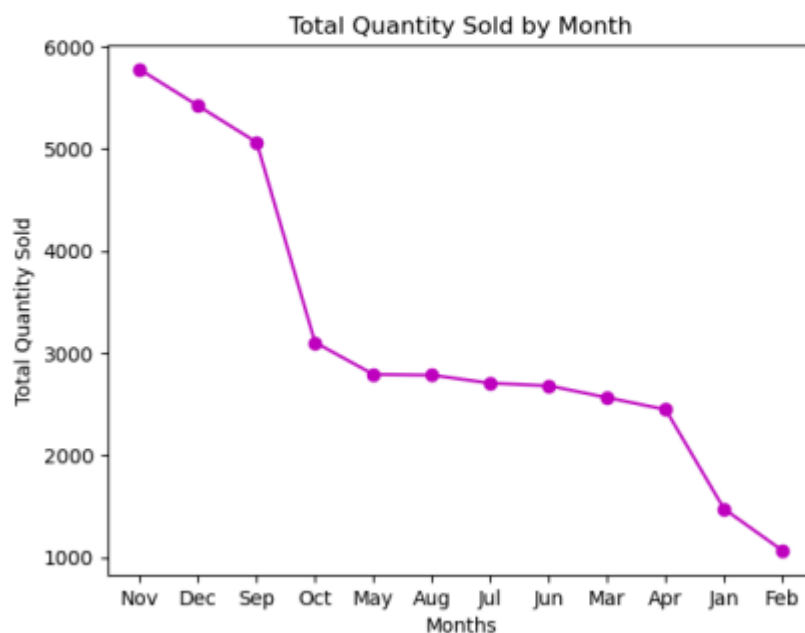


- Total Sales by Month

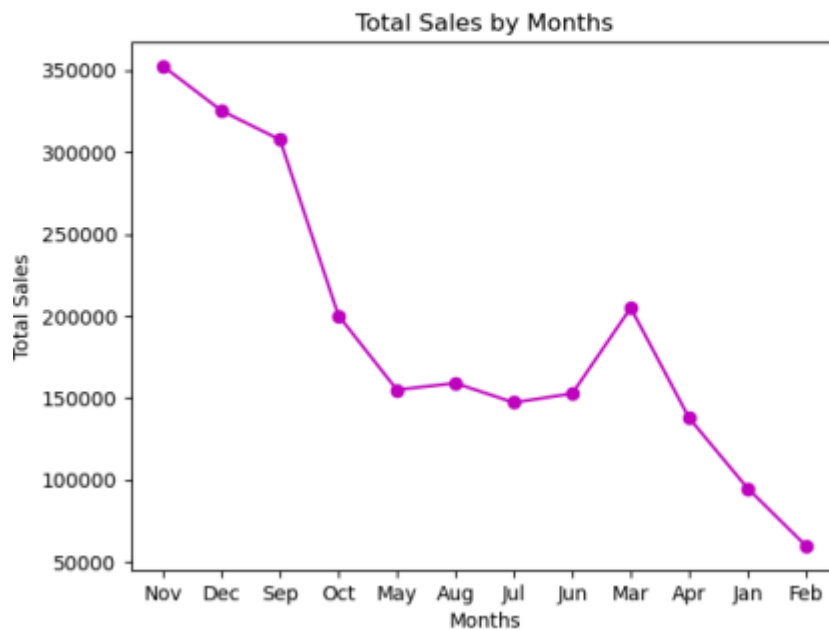
```
month_data = month_data.groupby("Order Month").sum().round(2)
month_data = month_data.sort_values(['Quantity', 'Sales'], ascending=[False, False])
month_data
```

	Quantity	Sales
Order Month		
Nov	5775	352461.07
Dec	5419	325293.50
Sep	5062	307649.95
Oct	3104	200322.98
May	2791	155028.81
Aug	2784	159044.06
Jul	2705	147238.10
Jun	2680	152718.68
Mar	2564	205005.49
Apr	2447	137762.13
Jan	1475	94924.84
Feb	1067	59751.25

```
#Total Quantity Sold by months
plt.plot(month_data["Order Month"], month_data["Quantity"], color = "m", marker = "o")
plt.xlabel("Months")
plt.ylabel("Total Quantity Sold")
plt.title("Total Quantity Sold by Month")
plt.show()
```



```
#Total Sales by months
plt.plot(month_data["Order Month"],month_data["Sales"],color = "m",marker = "o")
plt.xlabel("Months")
plt.ylabel("Total Sales")
plt.title("Total Sales by Months")
plt.show()
```



- Total Sales by Year and Month

```
: year_month_data = year_month_data.groupby(["Order Year","Order Month"]).sum().round(2)
year_month_data = year_month_data.sort_values(['Order Year','Quantity','Sales'],ascending=[True,False,False])
year_month_data
```

		Quantity	Sales
Order Year	Order Month		
2014	Nov	1219	78628.72
	Dec	1079	69545.62
	Sep	1000	81777.35
	Aug	609	27909.47
	Mar	585	55691.01
	Oct	573	31453.39
	Jul	550	33946.39
	Apr	536	28295.34
	Jun	521	34595.13
	May	466	23648.29
	Jan	284	14236.90
	Feb	159	4519.89

2015	Nov	1310	75972.56
	Dec	1203	74919.52
	Sep	1086	64595.92
	Oct	631	31404.92
	Aug	598	36898.33
	May	575	30131.69
	Jul	557	28765.32
	Apr	543	34195.21
	Mar	515	38726.25
	Jun	486	24797.29
	Feb	239	11951.41
	Jan	236	18174.08
2016	Dec	1414	96999.04
	Nov	1406	79411.97
	Sep	1316	73410.02
	May	863	56987.73
	Oct	767	59687.74
	Jul	758	39261.96
	Jun	742	40344.53
	Aug	693	31115.37
	Apr	635	38750.04
	Mar	579	51715.88
	Jan	358	18542.49
	Feb	306	22978.82
2017	Nov	1840	118447.82
	Dec	1723	83829.32
	Sep	1660	87866.65
	Oct	1133	77776.92
	Jun	931	52981.73
	May	887	44261.11
	Mar	885	58872.35
	Aug	884	63120.89
	Jul	840	45264.42
	Apr	733	36521.54
	Jan	597	43971.37
	Feb	363	20301.13

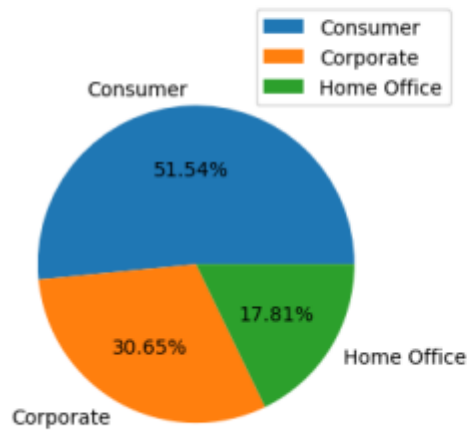
- Total Sales by Segment

```
segment_data = segment_data.groupby("Segment").sum().round(2)
segment_data
```

	Quantity	Sales
Segment		
Consumer	19521	1161401.34
Corporate	11608	706146.37
Home Office	6744	429653.15

```
#Total Qunatity Sold by Segments in percentage
plt.pie(segment_data["Quantity"], labels = segment_data["Segment"], autopct = "%0.2f%%", radius = 0.75)
plt.legend()
plt.title("Total Quantity Sold by Segment")
plt.show()
```

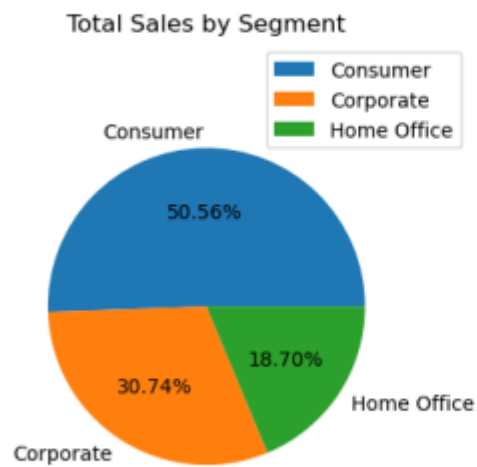
Total Quantity Sold by Segment



```

: #Total Sales by Segments in percentage
plt.pie(segment_data["Sales"], labels = segment_data["Segment"], autopct = "%0.2f%%", radius = 0.75)
plt.legend()
plt.title("Total Sales by Segment")
plt.show()

```



- Top Sales by Category

```

Category_data = Category_data.groupby("Category").sum().round(2)
Category_data = Category_data.sort_values(['Quantity', 'Sales'], ascending=[False, False])
Category_data

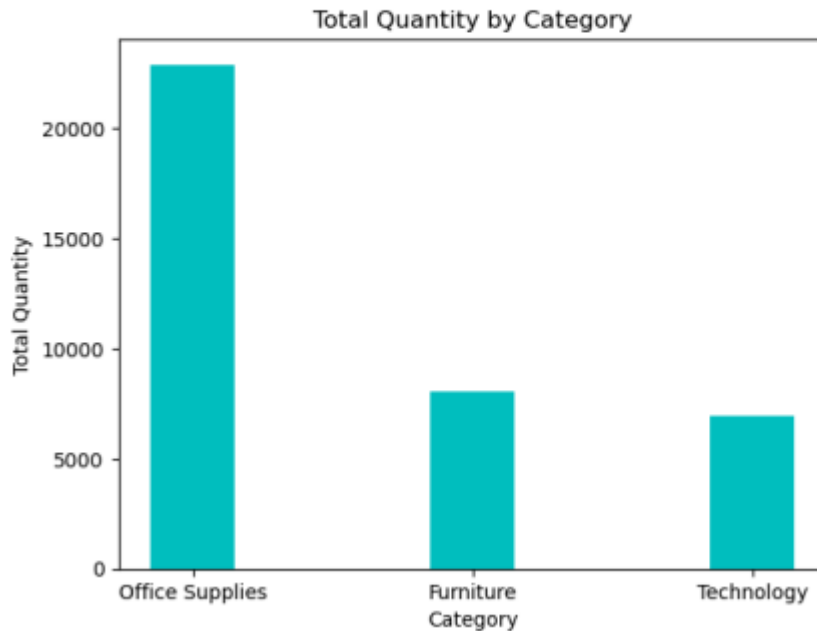
```

	Quantity	Sales
Category		
Office Supplies	22906	719047.03
Furniture	8028	741999.80
Technology	6939	836154.03

```

#Total Qunatity Sold by Category
plt.bar(Category_data["Category"],Category_data["Quantity"],width = 0.3,color = "c")
plt.xlabel("Category")
plt.ylabel("Total Quantity")
plt.title("Total Quantity by Category")
plt.show()

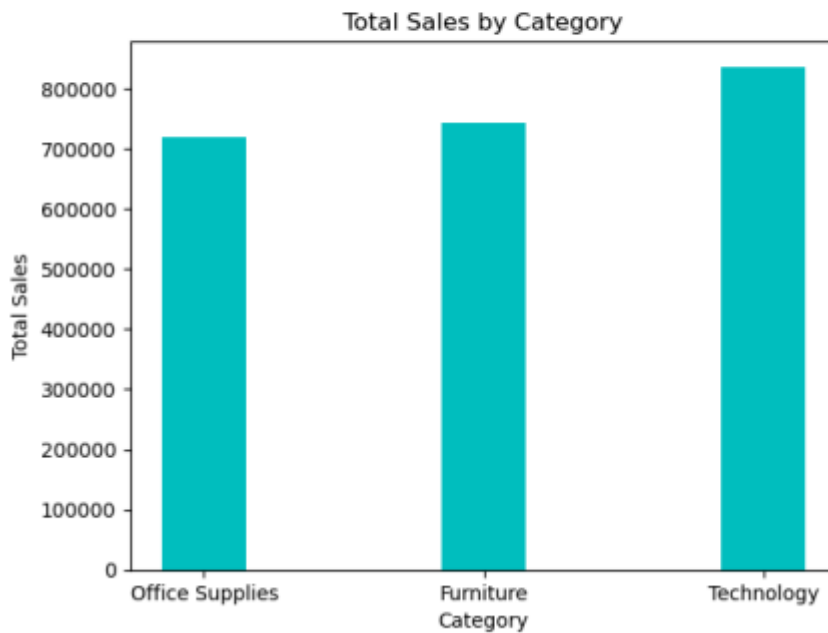
```



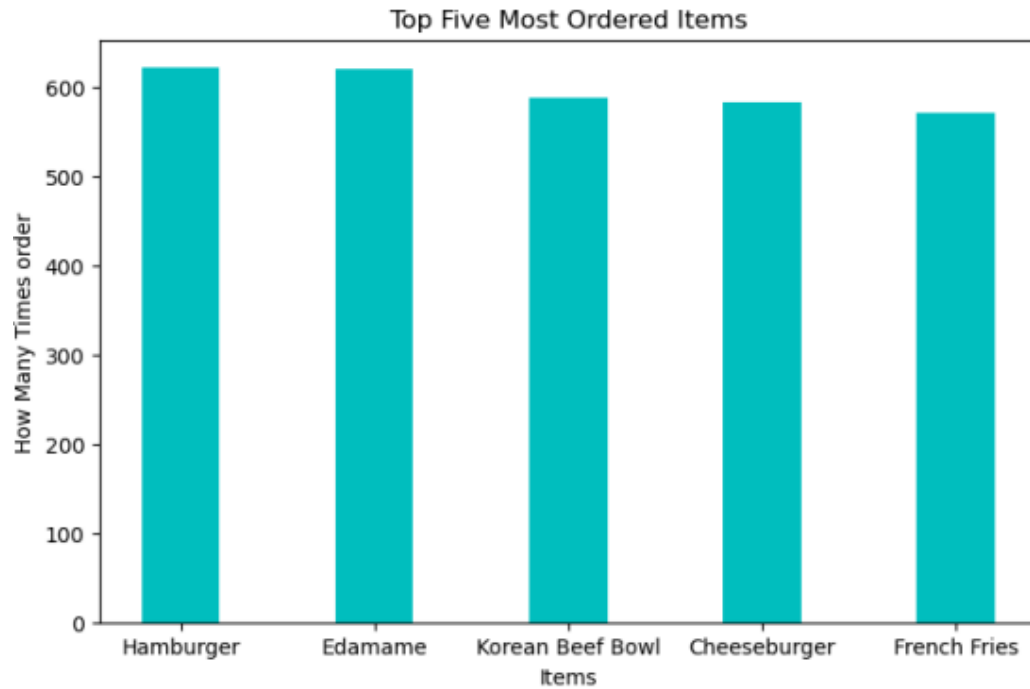
```

#Total Sales by Category
plt.bar(Category_data["Category"],Category_data["Sales"],width = 0.3,color = "c")
plt.xlabel("Category")
plt.ylabel("Total Sales")
plt.title("Total Sales by Category")
plt.show()

```



```
plt.figure(figsize=(8,5))
plt.bar(most_order_data["items"],most_order_data["How Many Times order"],width = 0.4,color = "c")
plt.xlabel("Items")
plt.ylabel("How Many Times order")
plt.title("Top Five Most Ordered Items")
plt.show()
```



- **Top Five States who have done the great sale**

```
#Total Quantity sold by State
plt.figure(figsize=(6,5))
plt.plot(state_data["State"],state_data["Quantity"],color = "c",marker = "s",linestyle = ":")
plt.xlabel("State")
plt.ylabel("Total Quantity Sold")
plt.title("Total Quantity Sold by State")
plt.show()
```

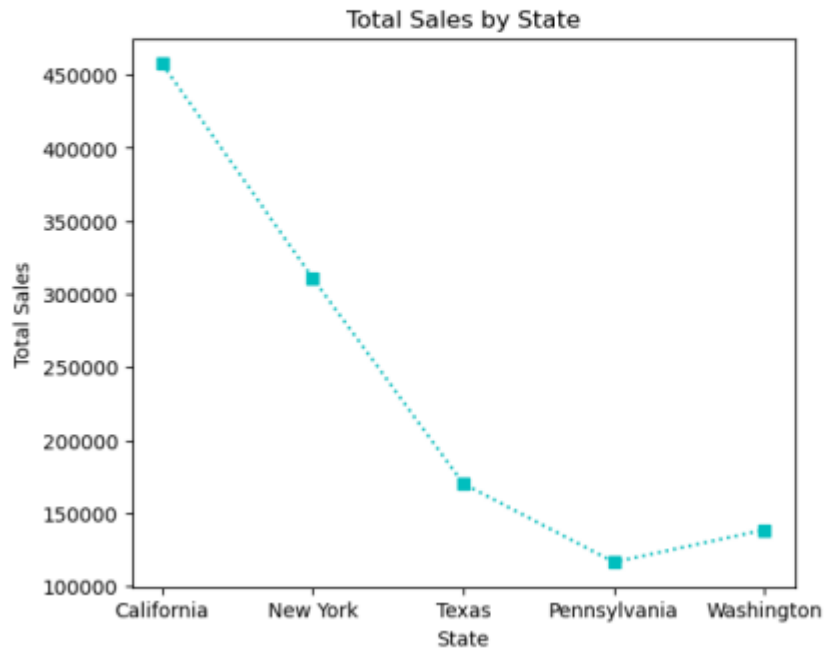


```
#Total Sales by State
```

```

: #Total Sales by State
plt.figure(figsize=(6,5))
plt.plot(state_data["State"],state_data["Sales"],color = "c",marker = "s",linestyle = ":")
plt.xlabel("State")
plt.ylabel("Total Sales")
plt.title("Total Sales by State")
plt.show()

```



- Total Sales by Region

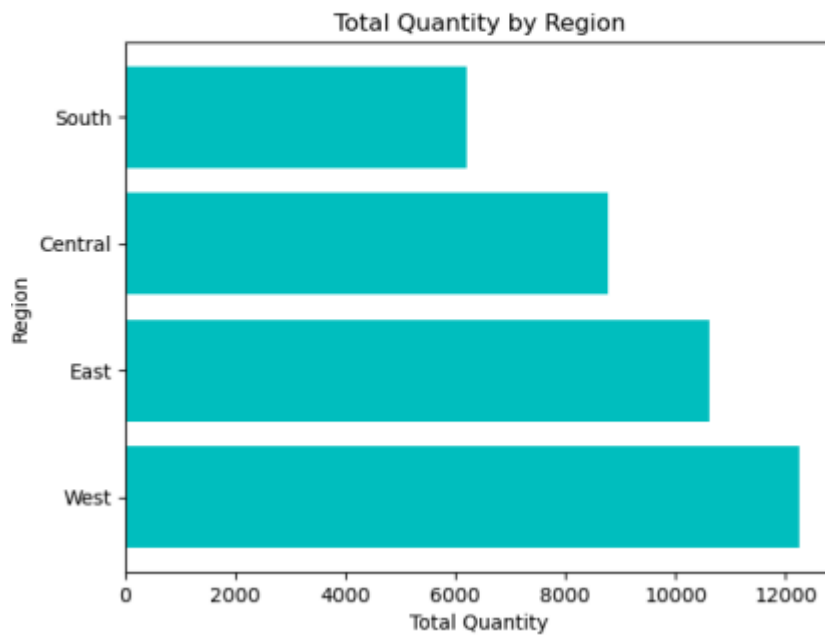
```

region_data = region_data.groupby(["Region"]).sum().round(2)
region_data = region_data.sort_values(["Quantity", "Sales"],ascending=[False, False])
region_data

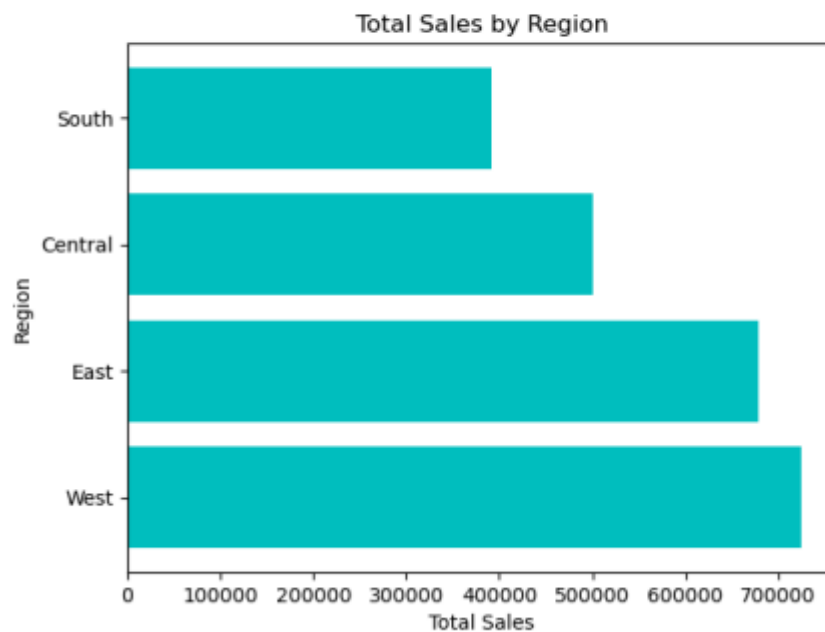
```

	Quantity	Sales
Region		
West	12266	725457.82
East	10618	678781.24
Central	8780	501239.89
South	6209	391721.90

```
#Total Quantity Sold by region
plt.barh(region_data["Region"],region_data["Quantity"],color = "c")
plt.ylabel("Region")
plt.xlabel("Total Quantity")
plt.title("Total Quantity by Region")
plt.show()
```



```
#Total sales by region
plt.barh(region_data["Region"],region_data["Sales"],color = "c")
plt.ylabel("Region")
plt.xlabel("Total Sales")
plt.title("Total Sales by Region")
plt.show()
```



Key Findings & Conclusion

This project has provided an in-depth analysis of sales data from a major superstore chain, focusing on the period from 2014 to 2017. The analysis aimed to uncover key insights that would help the superstore enhance its profitability and market positioning in the face of increasing market demands and competition.

Key Findings:

1. **Sales Trends:** The data reveals a consistent increase in sales over the years, indicating a positive growth trajectory for the superstore. This upward trend suggests that the superstore's strategies are effective in driving sales, although continued vigilance and adaptation to market changes are crucial for sustaining this growth.
2. **Seasonal Patterns:** Sales performance shows significant seasonal variations. The months of November and December are the strongest for sales, likely due to holiday shopping and end-of-year promotions. Conversely, January and February exhibit lower sales figures, possibly due to post-holiday lulls. Understanding these seasonal trends can help the superstore optimize inventory and marketing strategies throughout the year.
3. **Sales by Segment:** The consumer segment is the largest contributor to sales, followed by corporate and home office segments. This distribution highlights the importance of focusing on consumer-oriented strategies while also tailoring offerings to meet the needs of corporate clients and home office users.
4. **Product Categories:** Office supplies lead in sales, with furniture and technology following. This suggests that while office supplies are a staple for customers, there is also substantial demand for furniture and technology. Strategic emphasis on these categories can help balance the product mix and capitalize on market opportunities.
5. **Geographic Distribution:** The top-performing states for sales are California, New York, Texas, Pennsylvania, and Washington. These states represent key markets

with significant revenue potential. Tailoring regional strategies and promotions for these high-performing states can drive further growth.

6. **Shipping Preferences:** The most preferred shipping methods are Standard Class, Second Class, First Class, and Same Day. Standard Class and Second Class are the most popular, indicating that cost-effective shipping options are valued by customers. Offering efficient shipping options while managing costs effectively can enhance customer satisfaction and retention.

Recommendations:

- **Enhance Seasonal Promotions:** Capitalize on the strong sales months **November and December** with targeted marketing campaigns and promotions to maximize revenue. Develop strategies to boost sales during slower months **January and February**, such as special discounts or bundled offers.
- **Focus on Consumer Segment:** Given its significant contribution to sales, continue to prioritize consumer-focused products and marketing strategies. Develop tailored promotions and loyalty programs to further engage this segment.
- **Optimize Product Offering:** Maintain a strong inventory of office supplies while also expanding the range of furniture and technology products. Analyze sales data to adjust product offerings based on evolving customer preferences.
- **Regional Strategy:** Strengthen market presence in the top-performing states by localizing marketing efforts and exploring regional partnerships. Utilize data to address specific needs and preferences of customers in these states.
- **Shipping Strategy:** Continue to offer preferred shipping methods while exploring ways to enhance delivery efficiency. Consider introducing additional shipping options to meet varying customer needs and improve satisfaction.

By leveraging these insights and recommendations, the superstore can refine its business strategies to drive continued growth, optimize operations, and enhance its competitive position in the retail market.