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

Importing dataset

Add a comment
Ctrl+Alt+M

```
df=pd.read_csv('Superstore.csv', index_col=0)
df.head(10)
```

RowId	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City	State	Postal Code	Region	Product ID	Category	Sub-Category
1	CA-2021-152156	11/8/2021	11/11/2021	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420.0	South	FUR-BO-10001798	Furniture	Bookcases
2	CA-2021-152156	11/8/2021	11/11/2021	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	42420.0	South	FUR-CH-10000454	Furniture	Chairs
3	CA-2021-138688	6/12/2021	6/16/2021	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California	90036.0	West	OFF-LA-10000240	Office Supplies	Labels
4	US-2020-108966	10/11/2020	10/18/2020	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	33311.0	South	FUR-TA-10000577	Furniture	Tables
5	US-2020-108966	10/11/2020	10/18/2020	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	33311.0	South	OFF-ST-10000760	Office Supplies	Storage
6	CA-2019-115812	6/9/2019	6/14/2019	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	California	90032.0	West	FUR-FU-10001487	Furniture	Furnishings
7	CA-2019-115812	6/9/2019	6/14/2019	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	California	90032.0	West	OFF-AR-10002833	Office Supplies	Art
8	CA-2019-115812	6/9/2019	6/14/2019	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	California	90032.0	West	TEC-PH-10002275	Technology	Phones
9	CA-2019-115812	6/9/2019	6/14/2019	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	California	90032.0	West	OFF-BI-10003910	Office Supplies	Binders

Add a comment
Ctrl+Alt+M

10	CA-2019-115812	6/9/2019	6/14/2019	Standard Class	BH-11710	Brosina Hoffman	Consumer	United States	Los Angeles	California	90032.0	West	OFF-AP-10002892	Office Supplies	Appliances	I

1. Perform Exploratory Data Analysis

Add a comment
Ctrl+Alt+M

a. Check the summary

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 9994 entries, 1 to 9994
Data columns (total 20 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Order ID        9994 non-null   object
 1   Order Date      9994 non-null   object
 2   Ship Date       9994 non-null   object
 3   Ship Mode       9994 non-null   object
 4   Customer ID     9994 non-null   object
 5   Customer Name   9994 non-null   object
 6   Segment         9994 non-null   object
 7   Country/Region  9994 non-null   object
 8   City            9994 non-null   object
 9   State           9994 non-null   object
 10  Postal Code     9983 non-null   float64
 11  Region          9994 non-null   object
 12  Product ID      9994 non-null   object
 13  Category        9994 non-null   object
 14  Sub-Category    9994 non-null   object
 15  Product Name    9994 non-null   object
 16  Sales           9994 non-null   float64
 17  Quantity        9994 non-null   int64
 18  Discount        9994 non-null   float64
 19  Profit          9994 non-null   float64
dtypes: float64(4), int64(1), object(15)
memory usage: 1.6+ MB
```

b. Shape of the dataset

```
df.shape
```

```
(9994, 20)
```

c. Statistical summary

```
df.describe()
```

Add a comment
Ctrl+Alt+M

	Postal Code	Sales	Quantity	Discount	Profit
count	9983.000000	9994.000000	9994.000000	9994.000000	9994.000000
mean	55245.233297	229.858001	3.789574	0.156203	28.656896
std	32038.715955	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%	57103.000000	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

d. Display all numerical and categorical columns

```
numerical_data=df[['Postal Code','Sales','Quantity','Discount','Profit']]
numerical_data.head()
```

RowId	Postal Code	Sales	Quantity	Discount	Profit
1	42420.0	261.9600	2	0.00	41.9136
2	42420.0	731.9400	3	0.00	219.5820
3	90036.0	14.6200	2	0.00	6.8714
4	33311.0	957.5775	5	0.45	-383.0310
5	33311.0	22.3680	2	0.20	2.5164

```
categorical_data=df.drop(['Postal Code','Sales','Quantity','Discount','Profit'], axis=1)
categorical_data.head()
```

RowId	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Country/Region	City	State	Region	Product ID	Category	Sub-Category	Product Name
1	CA-2021-152156	11/8/2021	11/11/2021	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	South	FUR-BO-10001798	Furniture	Bookca	Bush Bookcase
2	CA-2021-152156	11/8/2021	11/11/2021	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	South	FUR-CH-10000454	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs,...
3	CA-2021-138688	6/12/2021	6/16/2021	Second Class	DV-13045	Darrin Van Huff	Corporate	United States	Los Angeles	California	West	OFF-LA-10000240	Office Supplies	Labels	Self-Adhesive Address Labels for Typewriters b...
4	US-2020-108966	10/11/2020	10/18/2020	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	South	FUR-TA-10000577	Furniture	Tables	Bretford CR4500 Series Slim Rectangular Table
5	US-2020-108966	10/11/2020	10/18/2020	Standard Class	SO-20335	Sean O'Donnell	Consumer	United States	Fort Lauderdale	Florida	South	OFF-ST-10000760	Office Supplies	Storage	Eldon Fold 'N Roll Cart System

Add a comment
Ctrl+Alt+M

e. Check for missing values if any

```
df.isnull().sum()
```

```
Order ID          0
Order Date        0
Ship Date         0
Ship Mode         0
Customer ID       0
Customer Name     0
Segment           0
Country/Region    0
City              0
```

```
State              0
Postal Code       11
Region             0
Product ID         0
Category           0
Sub-Category       0
Product Name       0
Sales              0
Quantity           0
Discount           0
Profit             0
dtype: int64
```

Add a comment
Ctrl+Alt+M

Comment on the above given points.

The above dataset has 20 columns and 9993 rows

data types:
float : 4,
int : 1,
object :15

11 Null values in Postal Code Column

2. Solve

a. What are the top 5 most frequently ordered products?

```
df['Product Name'].value_counts()[:5]
```

```
Product Name
Staple envelope            48
Easy-staple paper          46
Staples                    46
Avery Non-Stick Binders    20
Staples in misc. colors    19
Name: count, dtype: int64
```

b. How many unique customers are there, and which customer has placed the most orders?

```
print('Number of Customers:', df['Customer ID'].nunique())
customers=df['Customer Name'].value_counts()
print("Customer with most orders",customers[:1])
```

```
Number of Customers: 793
Customer with most orders Customer Name
William Brown    37
Name: count, dtype: int64
```

Add a comment
Ctrl+Alt+M

c. How do monthly or yearly sales trends look over time?

```
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()
monthly_trends=monthly_sales.groupby(['Year', 'Month'])['Sales'].sum().unstack(level=0)
monthly_trends
# monthly_sales
```

Year	2019	2020	2021	2022
Month				
1	14236.8950	18174.0756	18542.4910	43971.3740
2	4519.8920	11951.4110	22978.8150	20301.1334
3	55691.0090	38726.2520	51715.8750	58872.3528
4	28295.3450	34195.2085	38750.0390	36521.5361
5	23648.2870	30131.6865	56987.7280	44261.1102
6	34595.1276	24797.2920	40344.5340	52981.7257
7	33946.3930	28765.3250	39261.9630	45264.4160
8	27909.4685	36898.3322	31115.3743	63120.8880
9	81777.3508	64595.9180	73410.0249	87866.6520
10	31453.3930	31404.9235	59687.7450	77776.9232
11	78628.7167	75972.5635	79411.9658	118447.8250
12	69545.6205	74919.5212	96999.0430	83829.3188

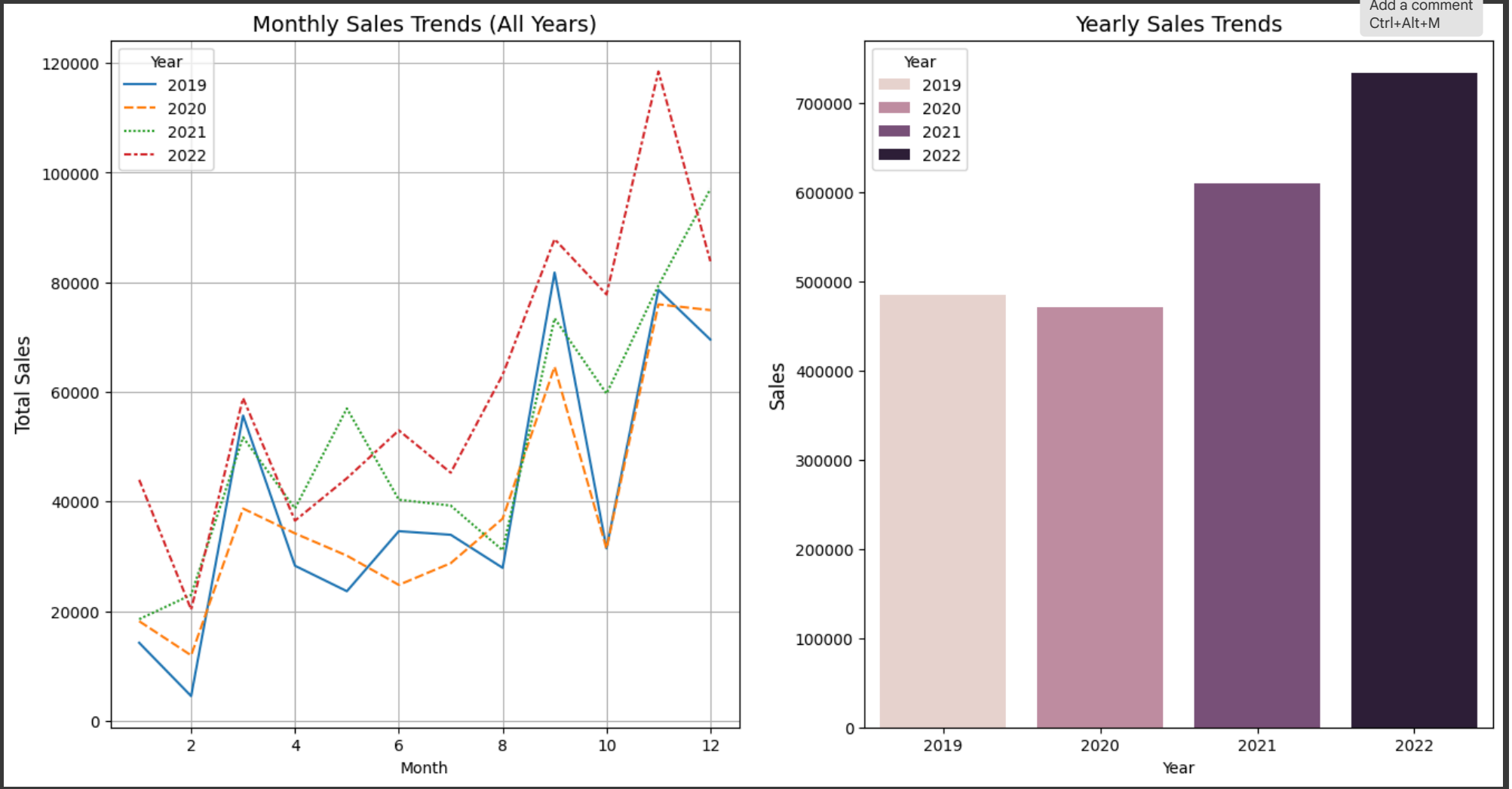
Add a comment
Ctrl+Alt+M

```
yearly_trends = df.groupby('Year')['Sales'].sum()
yearly_trends
```

```
Year
2019    484247.4981
2020    470532.5090
2021    609205.5980
2022    733215.2552
Name: Sales, dtype: float64
```

```
plt.figure(figsize=(16,8))

plt.subplot(1,2,1)
sns.lineplot(monthly_trends)
plt.title('Monthly Sales Trends (All Years)', fontsize=14)
plt.ylabel('Total Sales', fontsize=12)
plt.legend(title='Year', fontsize=10)
plt.grid(True)

plt.subplot(1,2,2)
sns.barplot(x=yearly_trends.index, y=yearly_trends.values, hue=yearly_trends.index)
```

```
plt.title('Yearly Sales Trends', fontsize=14)
plt.ylabel('Total Sales', fontsize=12)
plt.grid()

plt.ylabel('Sales')
plt.grid()
plt.show()
```



Add a comment
Ctrl+Alt+M

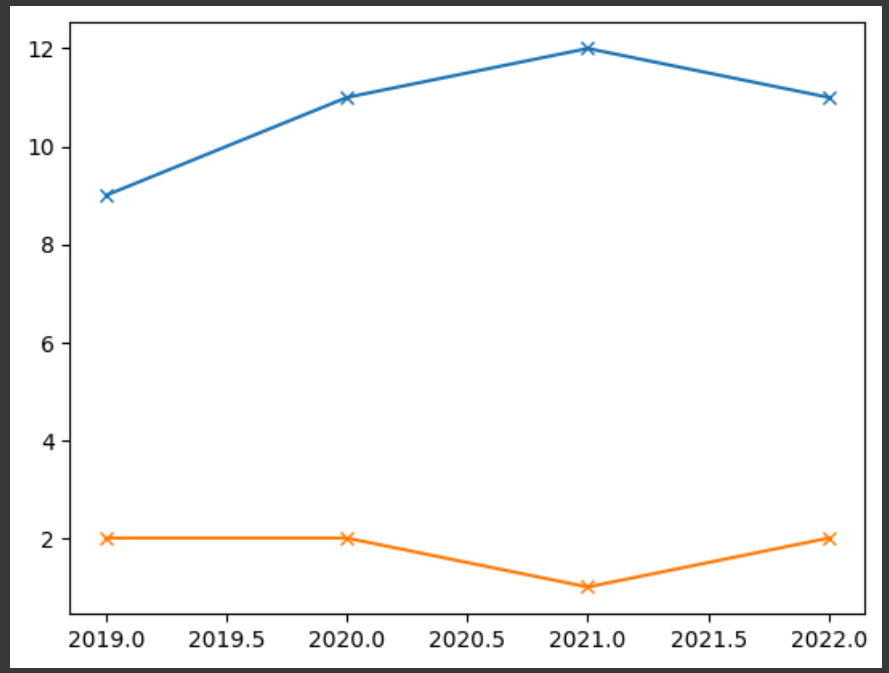
d. What are the peak and low sales months for each year?

```
peak_low_sales = monthly_sales.groupby('Year').apply(lambda group: pd.DataFrame({'Peak Month': [group.loc[group['Sales'].idxmax(), 'Month']],'Low Month': [group.lo
print(peak_low_sales)
```

```
      Peak Month  Low Month
Year
2019           9          2
2020          11          2
2021          12          1
2022          11          2
```

Add a comment
Ctrl+Alt+M

```
plt.plot(peak_low_sales, marker=('x'))
plt.show()
```



3. Category-Based Analysis

a. What are the top-selling categories and sub-categories?

```
catagory=df.groupby(['Category'])['Sales'].sum().sort_values(ascending=False)
```

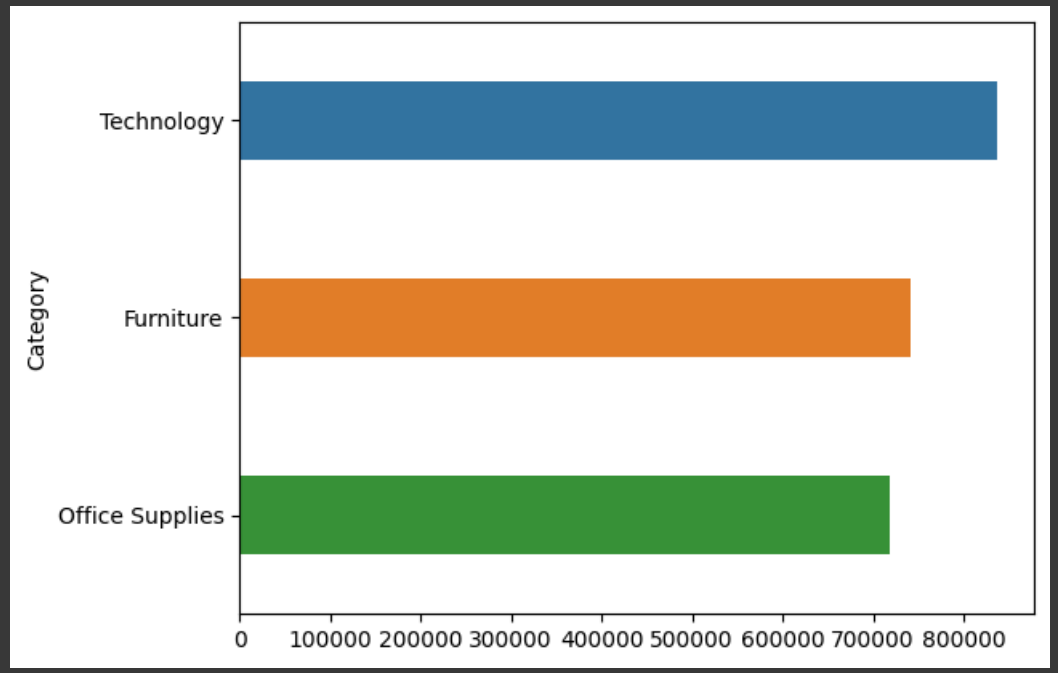
```
sub_category=df.groupby(['Sub-Category'])['Sales'].sum().sort_values
(ascending=False)
```

```
print("Top Selling Categories are : \n")
catagory
```

```
Top Selling Categories are :

Category
Technology         836154.0330
Furniture          741999.7953
Office Supplies    719047.0320
Name: Sales, dtype: float64
```

Add a comment
Ctrl+Alt+M

```
sns.barplot(x=catagory.values, y=catagory.index, hue=catagory.index, width=.4)
plt.xlabel='Sales'
plt.show()
```

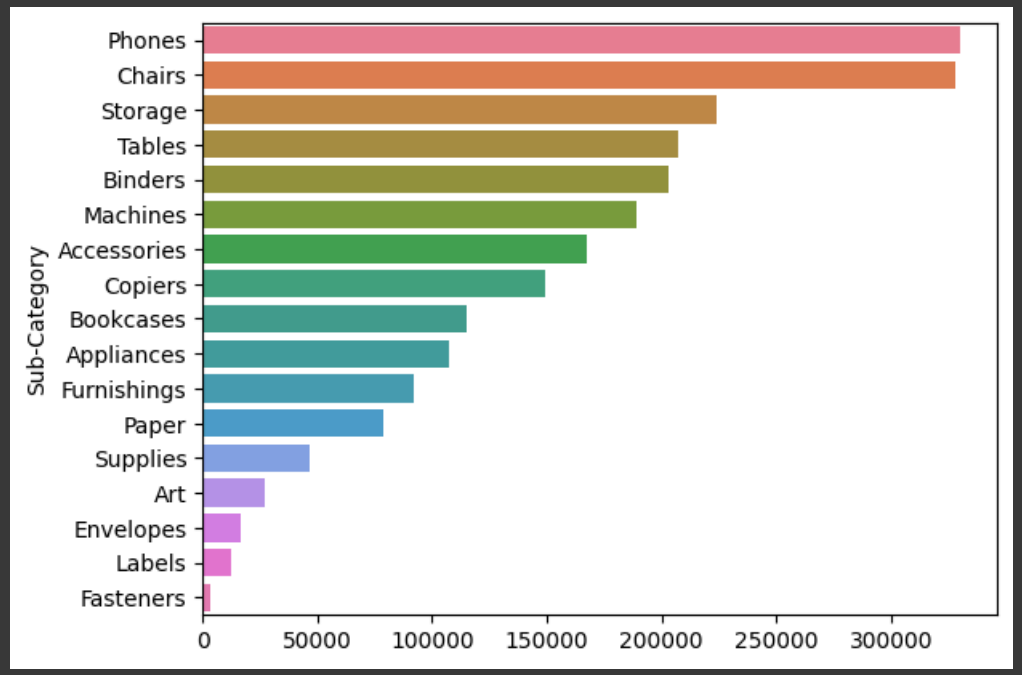



```
print("Top Selling Sub-Categories are : \n")

sub_category.head(5)
```

```
Top Selling Sub-Categories are :

Sub-Category
Phones      330007.054
Chairs      328449.103
Storage     223843.608
Tables      206965.532
Binders     203412.733
Name: Sales, dtype: float64
```

Add a comment
Ctrl+Alt+M

```
sns.barplot(x=sub_category.values, y=sub_category.index, hue=sub_category.index)
plt.xlabel='Sales'
plt.show()
```



b. How does the average order quantity vary across different categories?

```
df.groupby(['Category'])['Quantity'].mean().sort_values()
```

```
Category
Technology        3.756903
Furniture         3.785007
```

```
Office Supplies    3.801195
Name: Quantity, dtype: float64
```

4. Sales Performance by Region

```
df.rename({'Country/Region': 'Region'}, axis=1).head(2)
```

Add a comment
Ctrl+Alt+M

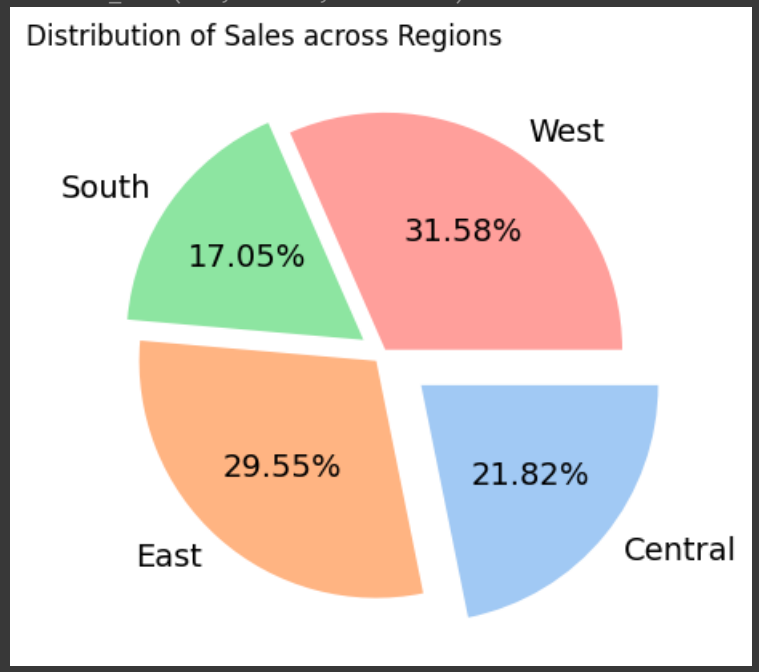
RowId	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Name	Segment	Region	City	State	...	Product ID	Category	Sub-Category	Product Name	Sales	Quantity
1	CA-2021-152156	2021-11-08	11/11/2021	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	...	FUR-BO-10001798	Furniture	Bookcases	Bush Somerset Collection Bookcase	261.96	2
2	CA-2021-152156	2021-11-08	11/11/2021	Second Class	CG-12520	Claire Gute	Consumer	United States	Henderson	Kentucky	...	FUR-CH-10000454	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs,...	731.94	3

2 rows × 22 columns

a. Which region has the highest sales, and how do sales compare across regions?

```
region=df.groupby(['Region'])['Sales'].sum().sort_index()
plt.pie(region, labels=region.index, autopct='%0.2f%%', colors=sns.color_palette('pastel'), explode=[0.21,0.041,0.1, 0.02],  counterclock=False, textprops={'fontsi
sns.barplot(region,width=0.5)
plt.title("Distribution of Sales across Regions")
plt.show()
```

```
c:\Python311\Lib\site-packages\seaborn\categorical.py:379: UserWarning: Attempting to set identical low and high xlims makes transformation singular; automatical
  ax.set_xlim(-.5, n - .5, auto=None)
```



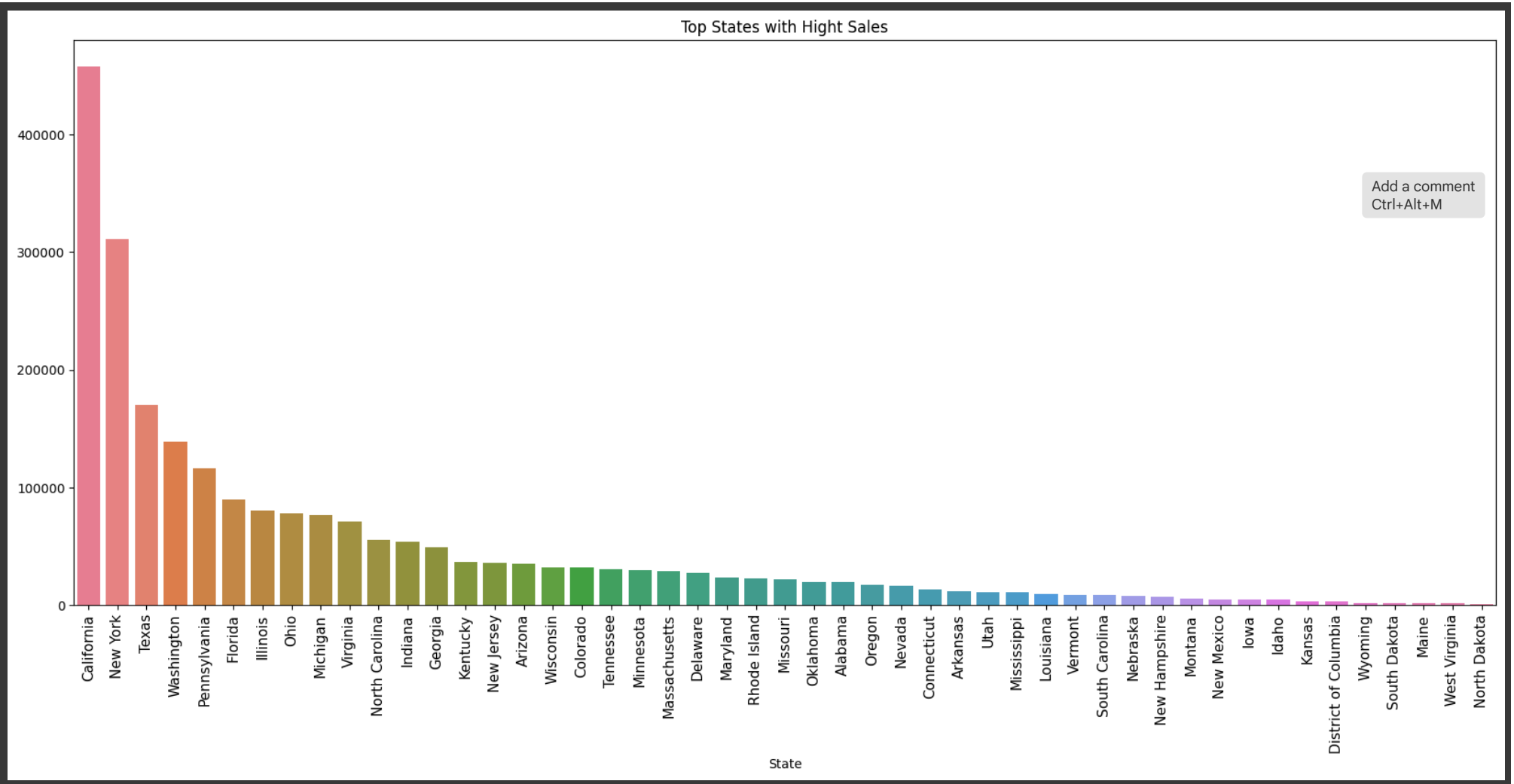
Add a comment
Ctrl+Alt+M

b. What is the distribution of sales across different states and cities?

```
State_sales=df.groupby(['State'])['Sales'].sum().sort_values(ascending=False)
State_sales.head()
```

```
State
California      457687.6315
New York        310876.2710
Texas           170188.0458
Washington      138641.2700
Pennsylvania    116511.9140
Name: Sales, dtype: float64
```

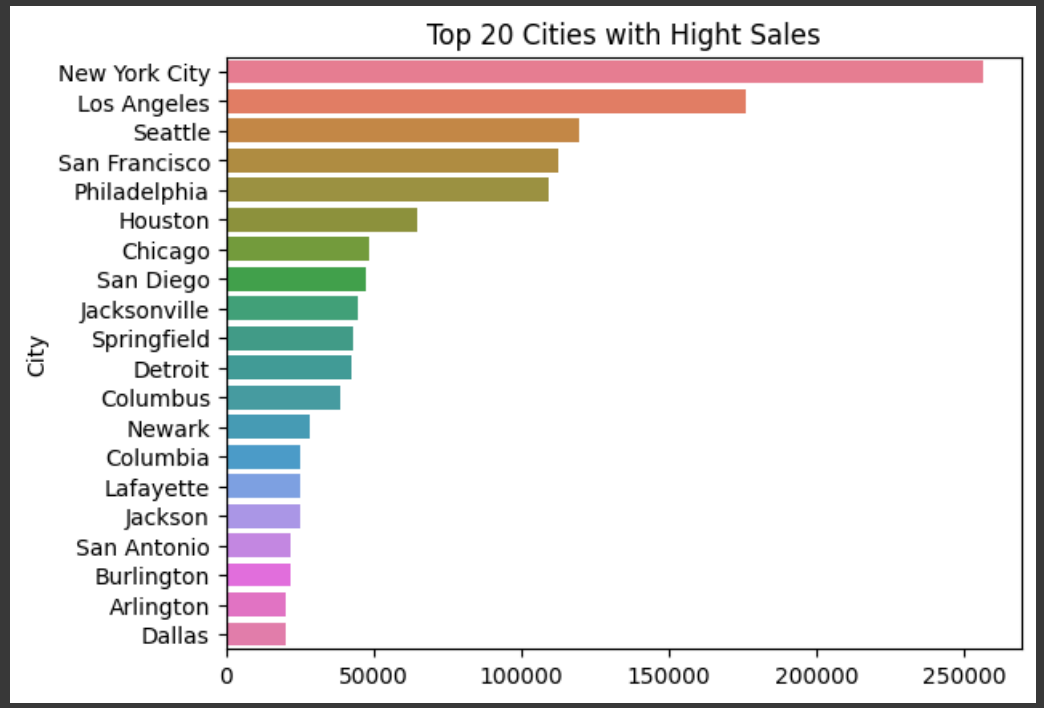
```
plt.figure(figsize=(20,8))
sns.barplot(x=State_sales.index,y=State_sales.values, hue=State_sales.index)
plt.xticks(rotation=90,fontsize=11)
plt.title("Top States with Hight Sales")
plt.show()
```




```
City_sales=df.groupby(['City'])['Sales'].sum().sort_values(ascending=False)[:20]
City_sales.head()
```

```
City
New York City    256368.161
Los Angeles      175851.341
Seattle          119540.742
San Francisco    112669.092
Philadelphia     109077.013
Name: Sales, dtype: float64
```

```
sns.barplot(y=City_sales.index,x=City_sales.values, hue=City_sales.index)
plt.title("Top 20 Cities with Hight Sales")
plt.show()
```



Add a comment
Ctrl+Alt+M

5. Customer Segmentation

a. What is the distribution of customers across segments (e.g., Consumer, Corporate)?

```
customer_segmentation=df.groupby(['Segment'])['Segment'].count()
customer_segmentation
```

```
Segment
Consumer       5191
Corporate      3020
Home Office    1783
Name: Segment, dtype: int64
```