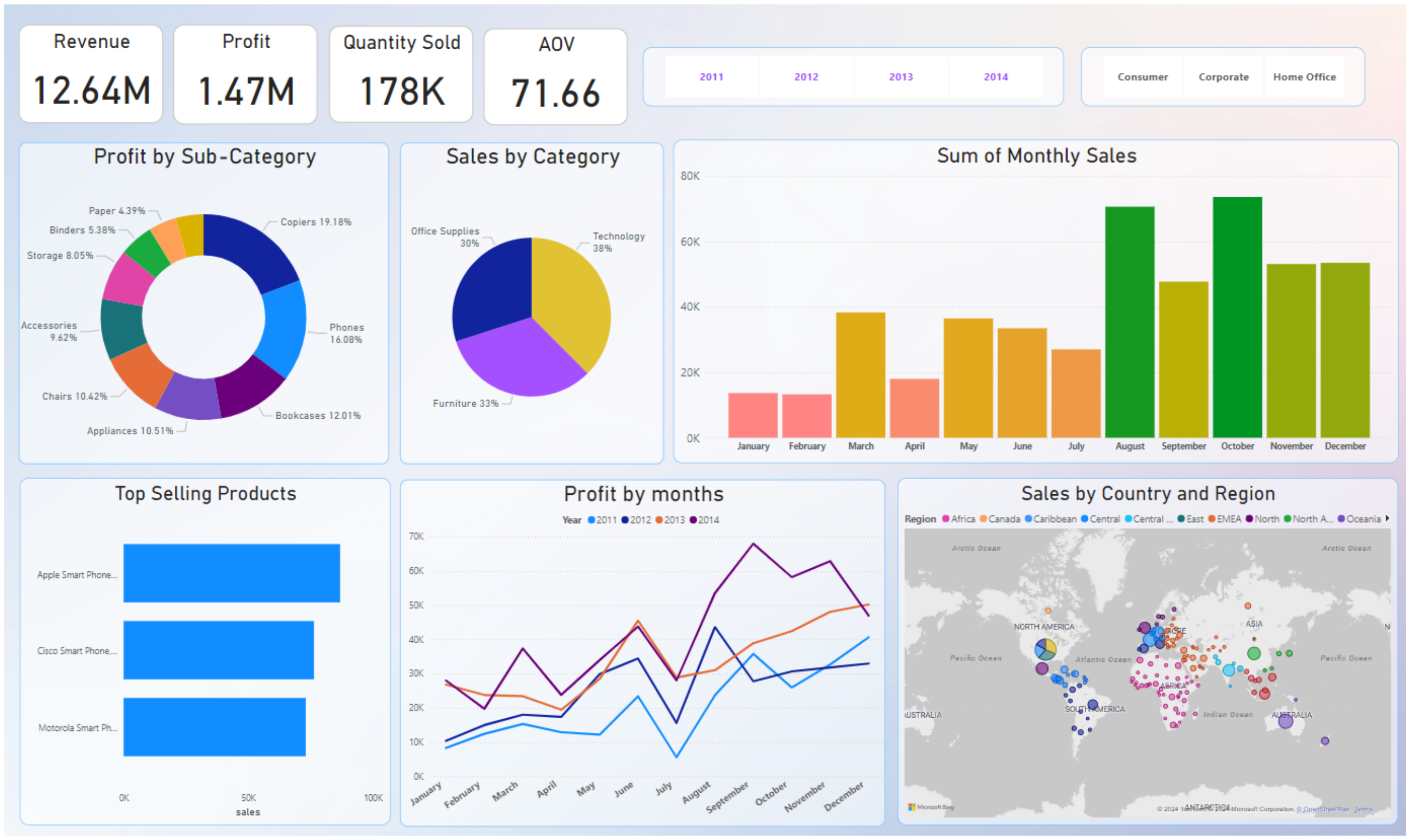
NAME : SATWIK K (NapQueen) ANARX DOCUMENTATION k.satwik10@gmail.com

Question1

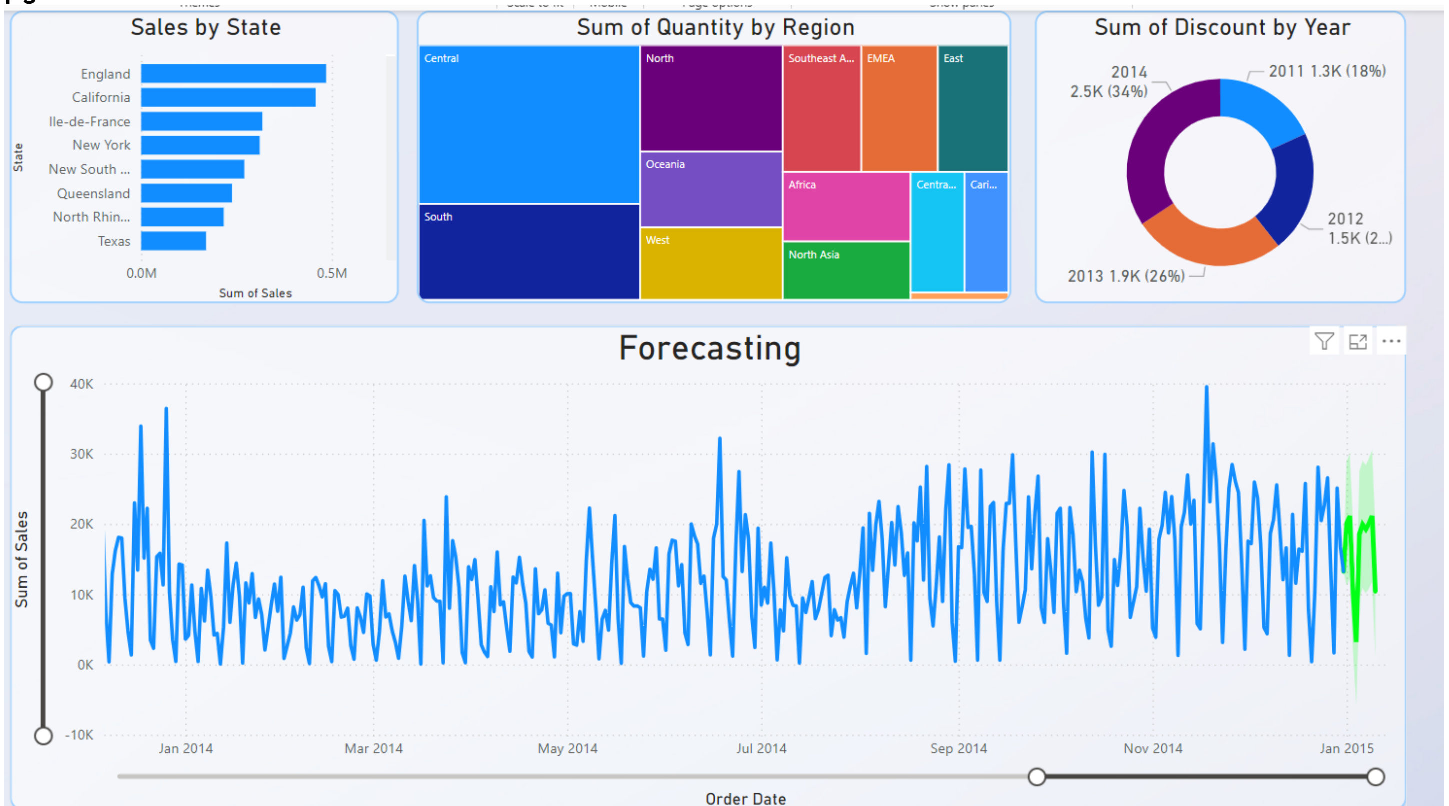
Use the provided sales data in Excel format and construct a dashboard in PowerBI

DASHBOARD

pg-1



pg-2



If any problem opening the files please us this link
https://drive.google.com/drive/folders/17VF3xbpNuMzjbQAKPlRXwJFBKaZV7R-P?usp=drive_link

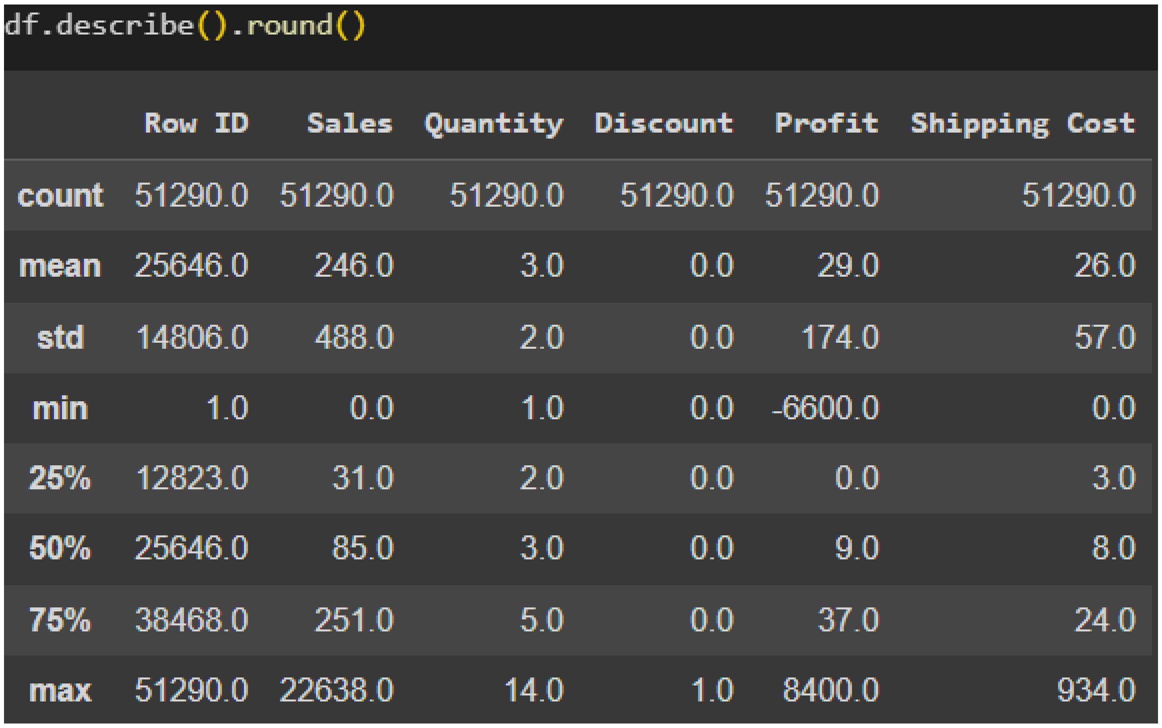
Steps

- Preprocess the data : I removed the pin code column because it had too many missing numbers and I added an extra column called AOV(Average order value) to find the average order value of our customer
- Simple KPI cards to show revenue, profits, quantity sold and AOV
- Used a Slicer to organize the data on y-o-y basis and Segment basis
- Used line chart with date order to get monthly profits over the years
- Created a simple Forecasting module which can forecast the next possible price upto 15days with 95% confidence
- Used map to Legend = Region and bubble size for sales for country to get which country and region we have the most sales in we can use the map and look at it
- Used a treemap to check the quantity based on each region
- Sales Growth Rate: Show the sales growth rate on a quarterly or yearly basis to understand the business's growth trends.

PYTHON DOCUMENTATION

OBJECTIVE

- Overall sales trend
- Sales in Countries
- Top 10 products by sales
- Total Sales by Segment
- Most Selling Products
- Most preferred Ship Mode
- Most Profitable Category and Sub-Category

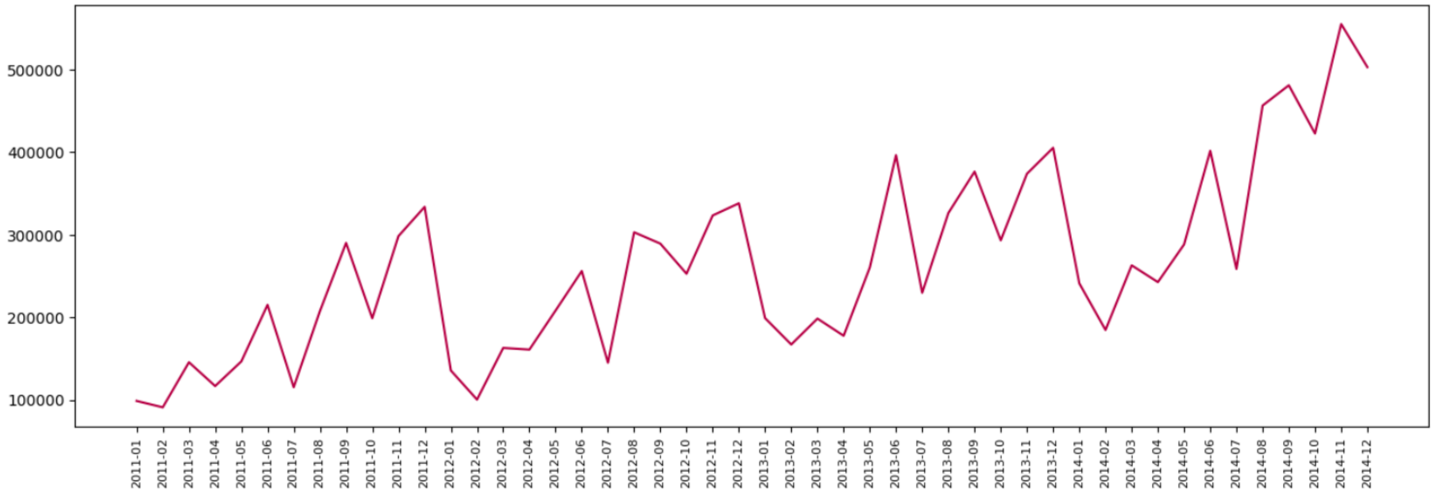


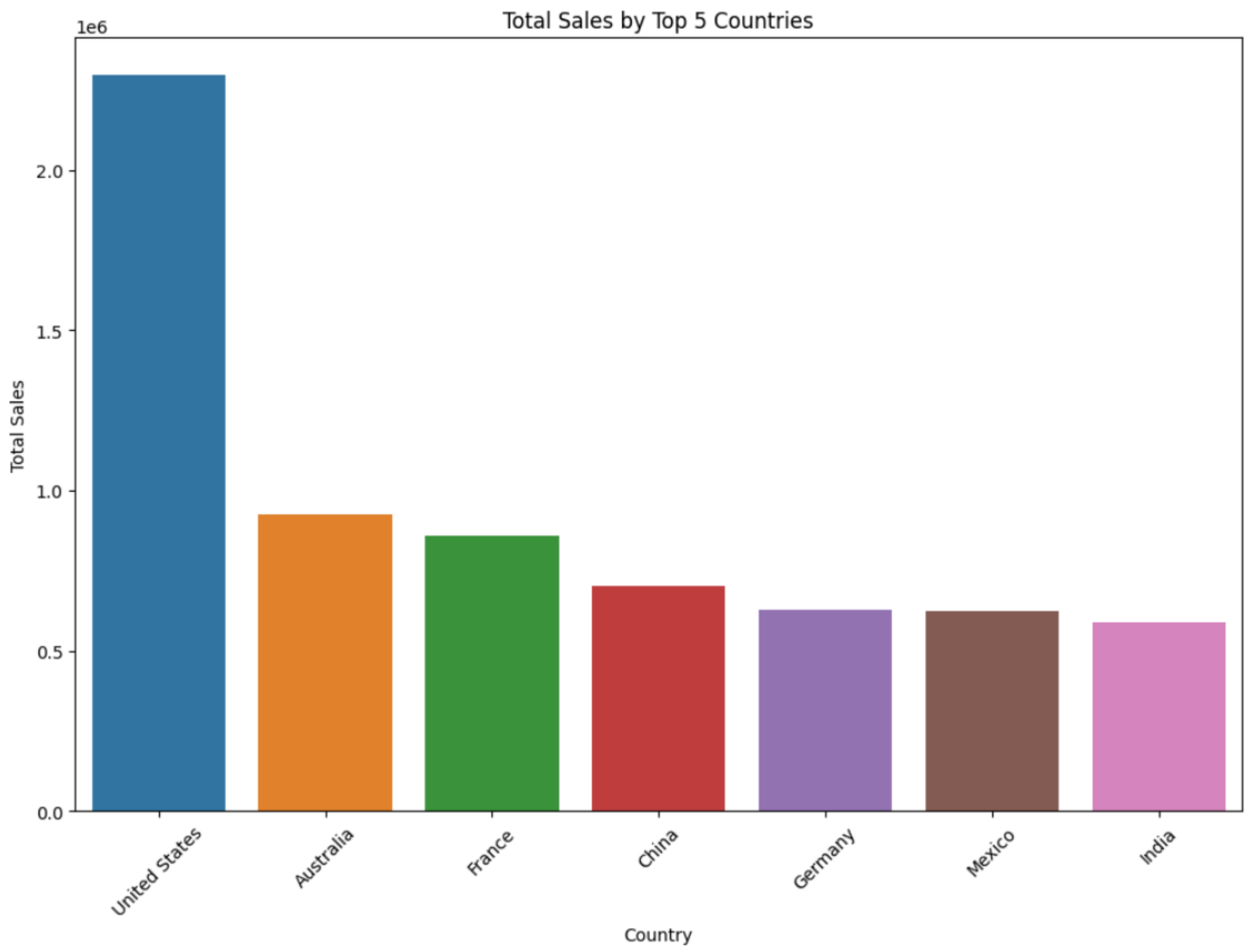
```
df.describe().round()
```

	Row ID	Sales	Quantity	Discount	Profit	Shipping Cost
count	51290.0	51290.0	51290.0	51290.0	51290.0	51290.0
mean	25646.0	246.0	3.0	0.0	29.0	26.0
std	14806.0	488.0	2.0	0.0	174.0	57.0
min	1.0	0.0	1.0	0.0	-6600.0	0.0
25%	12823.0	31.0	2.0	0.0	0.0	3.0
50%	25646.0	85.0	3.0	0.0	9.0	8.0
75%	38468.0	251.0	5.0	0.0	37.0	24.0
max	51290.0	22638.0	14.0	1.0	8400.0	934.0

vis

overall sales trend




Total Sales by Top 5 Countries
1e6
Total Sales
0.0
0.5
1.0
1.5
2.0
United States
Australia
France
China
Germany
Mexico
India
Country

dzvg503uh

August 2, 2024

NapQueen(ANARX) GLOBAL SUPERSTORE ANALYSIS

OBJECTIVE

Overall sales trend

Sales in Countries

Top 10 products by sales

Total Sales by Segment

Most Selling Products

Most preferred Ship Mode

Most Profitable Category and Sub-Category

IMPORTING REQUIRED LIBRARIES

[3]:
```
# Data Manipulation
import pandas as pd

# Data Visualisation
import matplotlib.pyplot as plt
%matplotlib inline

import seaborn as sns
```

IMPORTING THE DATASET

[4]:
```
# Importing dataset
df = pd.read_csv('Global-Superstore - Global-Superstore.csv.csv')
```

DATA AUDIT

You can't make your data work for you until you know what data you're talking about.

To get a quick idea of what the data looks like, we can call the head function on the data frame. By default, this returns the top five rows, but it can take in a parameter of how many rows to return.

[5]:
```
# First five rows of the dataset
df.head()
```

```
[5]:    Row ID           Order ID  Order Date   Ship Date      Ship Mode Customer ID  \
     0   32298    CA-2012-124891   7/31/2012   7/31/2012       Same Day    RH-19495
     1   26341     IN-2013-77878    2/5/2013    2/7/2013   Second Class    JR-16210
     2   25330     IN-2013-71249  10/17/2013  10/18/2013    First Class    CR-12730
     3   13524  ES-2013-1579342   1/28/2013   1/30/2013    First Class    KM-16375
     4   47221      SG-2013-4320   11/5/2013   11/6/2013       Same Day     RH-9495

          Customer Name      Segment           City            State  …  \
     0       Rick Hansen     Consumer  New York City         New York  …
     1     Justin Ritter    Corporate     Wollongong  New South Wales  …
     2      Craig Reiter     Consumer       Brisbane       Queensland  …
     3  Katherine Murray  Home Office         Berlin           Berlin  …
     4       Rick Hansen     Consumer          Dakar            Dakar  …

             Product ID    Category Sub-Category  \
     0   TEC-AC-10003033  Technology  Accessories
     1   FUR-CH-10003950   Furniture       Chairs
     2   TEC-PH-10004664  Technology       Phones
     3   TEC-PH-10004583  Technology       Phones
     4  TEC-SHA-10000501  Technology      Copiers

                                            Product Name     Sales Quantity  \
     0  Plantronics CS510 - Over-the-Head monaural Wir…  2309.650        7
     1          Novimex Executive Leather Armchair, Black  3709.395        9
     2                  Nokia Smart Phone, with Caller ID  5175.171        9
     3                     Motorola Smart Phone, Cordless  2892.510        5
     4                       Sharp Wireless Fax, High-Speed  2832.960        8

       Discount    Profit  Shipping Cost  Order Priority
     0      0.0  762.1845         933.57        Critical
     1      0.1 -288.7650         923.63        Critical
     2      0.1  919.9710         915.49          Medium
     3      0.1  -96.5400         910.16          Medium
     4      0.0  311.5200         903.04        Critical

     [5 rows x 24 columns]
```

```
[6]: # Last five rows of the dataset
     df.tail()
```

```
[6]:        Row ID         Order ID Order Date  Ship Date       Ship Mode  \
     51285   29002   IN-2014-62366  6/19/2014  6/19/2014        Same Day
     51286   35398  US-2014-102288  6/20/2014  6/24/2014  Standard Class
     51287   40470  US-2013-155768  12/2/2013  12/2/2013        Same Day
     51288    9596  MX-2012-140767  2/18/2012  2/22/2012  Standard Class
     51289    6147  MX-2012-134460  5/22/2012  5/26/2012    Second Class
```

```
      Customer ID     Customer Name      Segment      City       State  …  \
51285    KE-16420  Katrina Edelman    Corporate      Kure   Hiroshima  …
51286    ZC-21910  Zuschuss Carroll     Consumer   Houston       Texas  …
51287    LB-16795    Laurel Beltran  Home Office    Oxnard  California  …
51288    RB-19795        Ross Baird  Home Office  Valinhos   S<o Paulo  …
51289    MC-18100     Mick Crebagga     Consumer  Tipitapa     Managua  …

            Product ID         Category Sub-Category  \
51285  OFF-FA-10000746  Office Supplies    Fasteners
51286  OFF-AP-10002906  Office Supplies   Appliances
51287  OFF-EN-10001219  Office Supplies    Envelopes
51288  OFF-BI-10000806  Office Supplies      Binders
51289  OFF-PA-10004155  Office Supplies        Paper

                                           Product Name   Sales Quantity  \
51285                        Advantus Thumb Tacks, 12 Pack  65.100        5
51286  Hoover Replacement Belt for Commercial Guardsm…  0.444        1
51287       #10- 4 1/8" x 9 1/2" Security-Tint Envelopes  22.920        3
51288                             Acco Index Tab, Economy  13.440        2
51289            Eaton Computer Printout Paper, 8.5 x 11  61.380        3

      Discount   Profit  Shipping Cost  Order Priority
51285      0.0   4.5000           0.01          Medium
51286      0.8  -1.1100           0.01          Medium
51287      0.0  11.2308           0.01            High
51288      0.0   2.4000           0.00          Medium
51289      0.0   1.8000           0.00            High

[5 rows x 24 columns]
```

[7]:
```python
# Shape of the dataset
df.shape
```

[7]:
```
(51290, 24)
```

[8]:
```python
# Columns present in the dataset
df.columns
```

[8]:
```
Index(['Row ID', 'Order ID', 'Order Date', 'Ship Date', 'Ship Mode',
       'Customer ID', 'Customer Name', 'Segment', 'City', 'State', 'Country',
       'Postal Code', 'Market', 'Region', 'Product ID', 'Category',
       'Sub-Category', 'Product Name', 'Sales', 'Quantity', 'Discount',
       'Profit', 'Shipping Cost', 'Order Priority'],
      dtype='object')
```

[9]:
```python
# A concise summary of the dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51290 entries, 0 to 51289
Data columns (total 24 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   Row ID          51290 non-null  int64  
 1   Order ID        51290 non-null  object 
 2   Order Date      51290 non-null  object 
 3   Ship Date       51290 non-null  object 
 4   Ship Mode       51290 non-null  object 
 5   Customer ID     51290 non-null  object 
 6   Customer Name   51290 non-null  object 
 7   Segment         51290 non-null  object 
 8   City            51290 non-null  object 
 9   State           51290 non-null  object 
 10  Country         51290 non-null  object 
 11  Postal Code     9994 non-null   float64
 12  Market          51290 non-null  object 
 13  Region          51290 non-null  object 
 14  Product ID      51290 non-null  object 
 15  Category        51290 non-null  object 
 16  Sub-Category    51290 non-null  object 
 17  Product Name    51290 non-null  object 
 18  Sales           51290 non-null  float64
 19  Quantity        51290 non-null  int64  
 20  Discount        51290 non-null  float64
 21  Profit          51290 non-null  float64
 22  Shipping Cost   51290 non-null  float64
 23  Order Priority  51290 non-null  object 
dtypes: float64(5), int64(2), object(17)
memory usage: 9.4+ MB
```

Now we can do further analysis on our data to answer our questions. Before that, we should see if there are any missing values in our data set.To check if there are any missing values in the entire data set we use the isnull function, then see if there are any values.

[10]:
```
# Checking missing values
df.isna().sum()
```

[10]:
```
Row ID            0
Order ID          0
Order Date        0
Ship Date         0
Ship Mode         0
Customer ID       0
Customer Name     0
Segment           0
City              0
```

```
State                 0
Country               0
Postal Code       41296
Market                0
Region                0
Product ID            0
Category              0
Sub-Category          0
Product Name          0
Sales                 0
Quantity              0
Discount              0
Profit                0
Shipping Cost         0
Order Priority        0
dtype: int64
```

Postal code has many missing values since we have city address we dont require postal code and we cant fill postal codes with other vales so drop postal code column

[12]:
```
# Drop the 'Postal Code' column
df = df.drop(columns=['Postal Code'])
```

Next, we can look at some descriptive statistics of the data frame with the describe method.

This shows some descriptive statistics on the data set. Notice, it only shows the statistics on the numerical columns. From here you can see the following statistics:

- Row count, which aligns to what the shape attribute showed us.
- The mean, or average.
- The standard deviation, or how spread out the data is.
- The minimum and maximum value of each column
- The number of items that fall within the first, second, and third percentiles.

[13]:
```
# Generating descriptive statistics summary
df.describe().round()
```

[13]:
```
        Row ID    Sales  Quantity  Discount   Profit  Shipping Cost
count  51290.0  51290.0   51290.0   51290.0  51290.0        51290.0
mean   25646.0    246.0       3.0       0.0     29.0           26.0
std    14806.0    488.0       2.0       0.0    174.0           57.0
min        1.0      0.0       1.0       0.0  -6600.0            0.0
25%    12823.0     31.0       2.0       0.0      0.0            3.0
50%    25646.0     85.0       3.0       0.0      9.0            8.0
75%    38468.0    251.0       5.0       0.0     37.0           24.0
max    51290.0  22638.0      14.0       1.0   8400.0          934.0
```

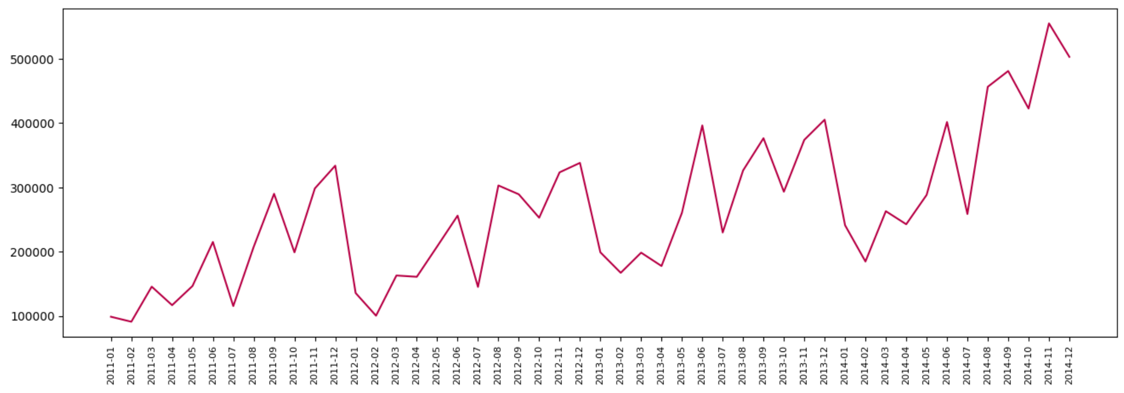
EXPLORATORY DATA ANALYSIS

- WHAT IS THE OVERALL SALES TREND?

[16]:
```python
# Getting month year from order_date
df['Order Date'] = pd.to_datetime(df['Order Date'])
df['month_year'] = df['Order Date'].apply(lambda x: x.strftime('%Y-%m'))
```

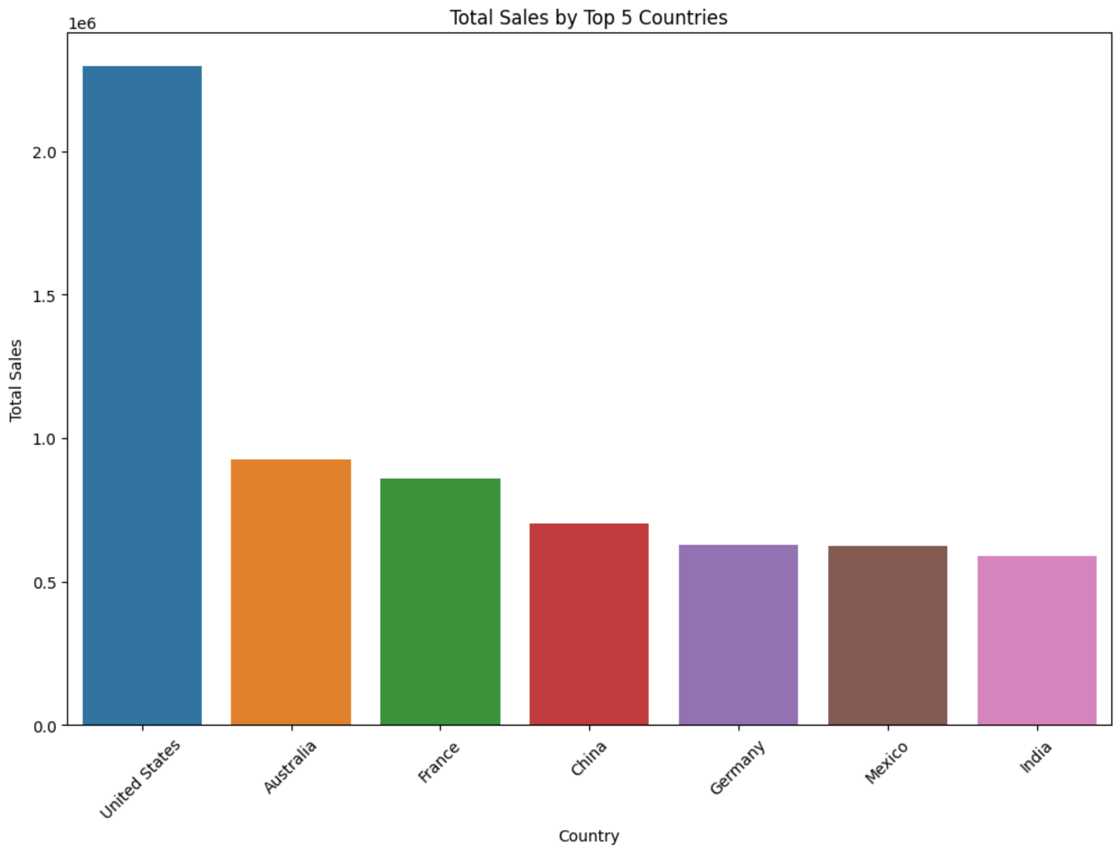
[20]:
```python
# Step 3: Group by 'month_year' and sum 'Sales'
df_temp = df.groupby('month_year')['Sales'].sum().reset_index()
```

[22]:
```python
# Setting the figure size
plt.figure(figsize=(16, 5))
plt.plot(df_temp['month_year'], df_temp['Sales'], color='#b80045')
plt.xticks(rotation='vertical', size=8)
plt.show()
```



- What are the sales in Countries?

[37]:
```python
# Group by 'Country' and sum 'Sales'
country_sales = df.groupby('Country')['Sales'].sum().reset_index()

# Sort the dataframe in descending order and select the top 5 countries
top_countries = country_sales.sort_values(by='Sales', ascending=False).head(7)

# Plotting the data
plt.figure(figsize=(12, 8))
sns.barplot(x='Country', y='Sales', data=top_countries, estimator=sum)
plt.title('Total Sales by Top 5 Countries')
plt.xlabel('Country')
plt.ylabel('Total Sales')
plt.xticks(rotation=45)
plt.show()
```



- WHICH ARE THE TOP 10 PRODUCTS BY SALES?

[24]:
```
# Grouping products by sales
prod_sales = df.groupby('Product Name')['Sales'].sum().reset_index()

# Sorting the dataframe in descending order
prod_sales.sort_values(by=['Sales'], inplace=True, ascending=False)

# Top 10 products by sales
prod_sales[:10]
```

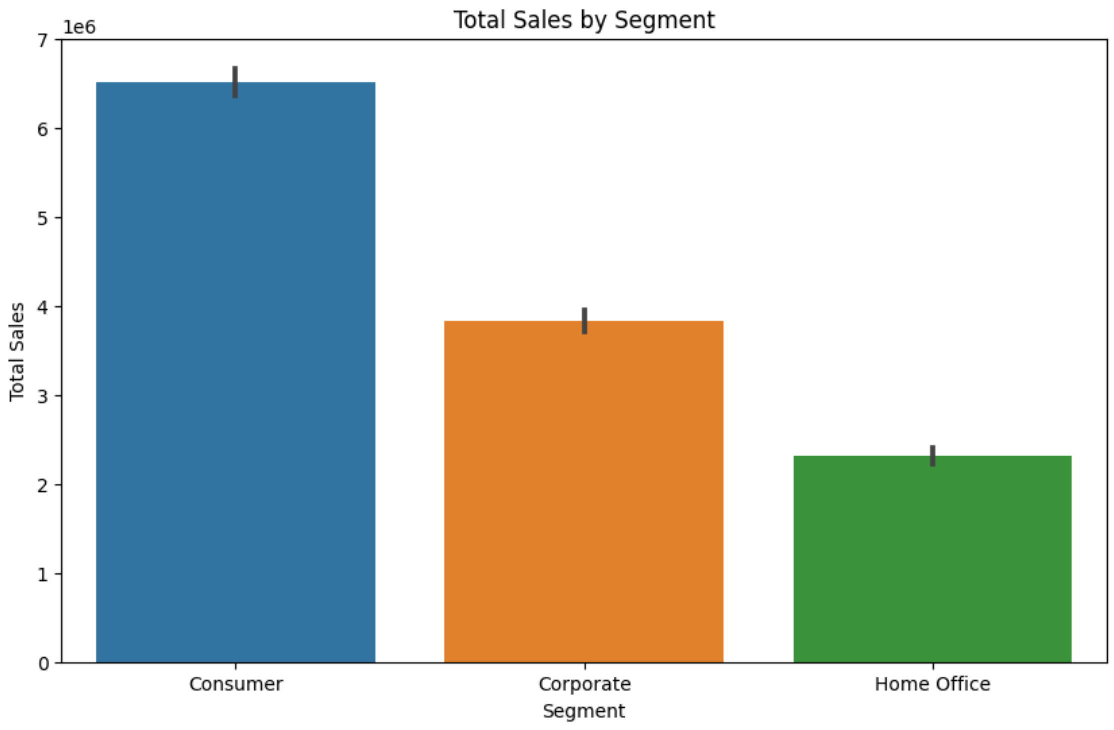
[24]:
```
                                        Product Name       Sales
310                         Apple Smart Phone, Full Size  86935.7786
970                         Cisco Smart Phone, Full Size  76441.5306
2415                     Motorola Smart Phone, Full Size  73156.3030
2501                        Nokia Smart Phone, Full Size  71904.5555
866                   Canon imageCLASS 2200 Advanced Copier  61599.8240
1837             Hon Executive Leather Armchair, Adjustable  58193.4841
2631   Office Star Executive Leather Armchair, Adjust…  50661.6840
1714   Harbour Creations Executive Leather Armchair, …  50121.5160
2988                         Samsung Smart Phone, Cordless  48653.4600
```

```
2502                    Nokia Smart Phone, with Caller ID  47877.7857
```

-

Total Sales by Segment

[38]:
```
plt.figure(figsize=(10, 6))
sns.barplot(x='Segment', y='Sales', data=df, estimator=sum)
plt.title('Total Sales by Segment')
plt.xlabel('Segment')
plt.ylabel('Total Sales')
plt.show()
```



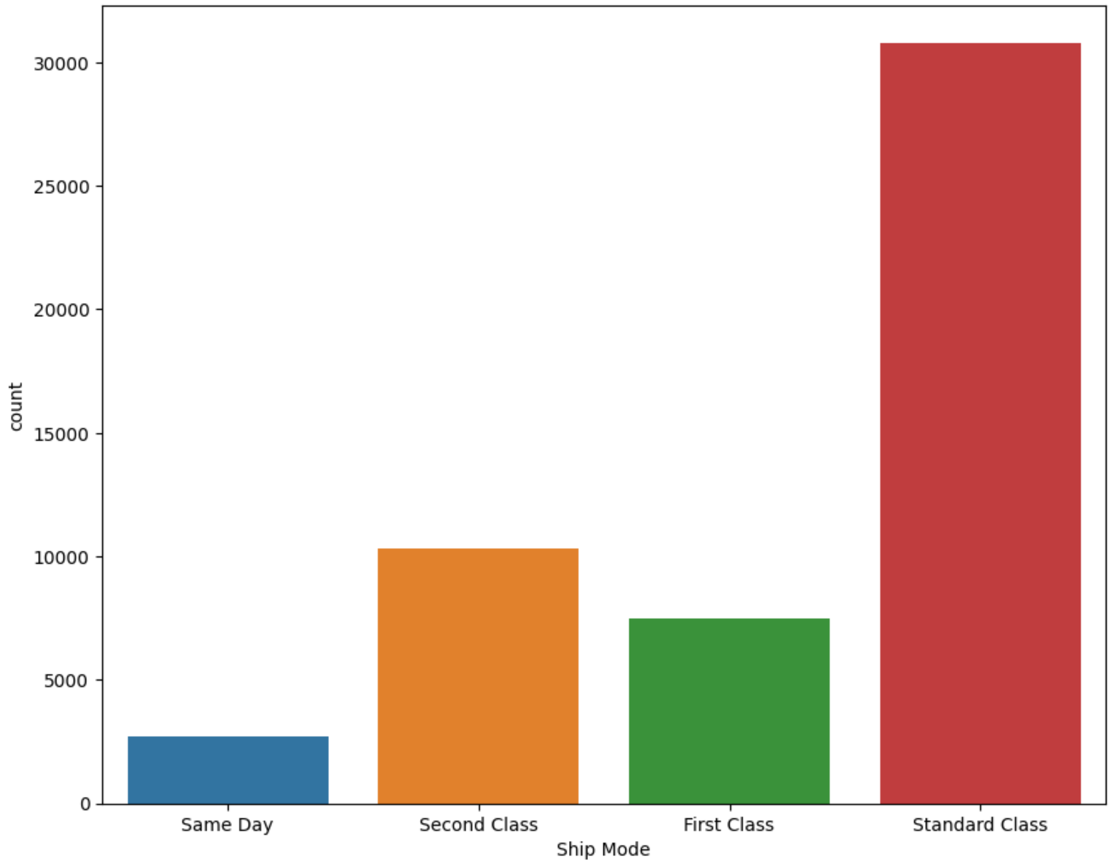
- WHICH ARE THE MOST SELLING PRODUCTS?

[26]:
```
# Grouping products by Quantity
best_selling_prods = pd.DataFrame(df.groupby('Product Name')['Quantity'].sum().
 ↪reset_index())

# Sorting the dataframe in descending order
best_selling_prods.sort_values(by=['Quantity'], inplace=True, ascending=False)

# Most selling products
best_selling_prods[:10]
```

[26]:
```
                                  Product Name  Quantity
3275                                   Staples       876
894                     Cardinal Index Tab, Clear       337
1210                Eldon File Cart, Single Width       321
2840               Rogers File Cart, Single Width       262
3070   Sanford Pencil Sharpener, Water Color       259
3335   Stockwell Paper Clips, Assorted Sizes       253
446                        Avery Index Tab, Clear       252
1981                       Ibico Index Tab, Clear       251
3179                Smead File Cart, Single Width       250
3266   Stanley Pencil Sharpener, Water Color       242
```

- WHAT IS THE MOST PREFERRED SHIP MODE?

[27]:
```python
# Setting the figure size
plt.figure(figsize=(10, 8))

# countplot: Show the counts of observations in each categorical bin using bars
sns.countplot(x='Ship Mode', data=df)

# Display the figure
plt.show()
```



- WHICH ARE THE MOST PROFITABLE CATEGORY AND SUB-CATEGORY?

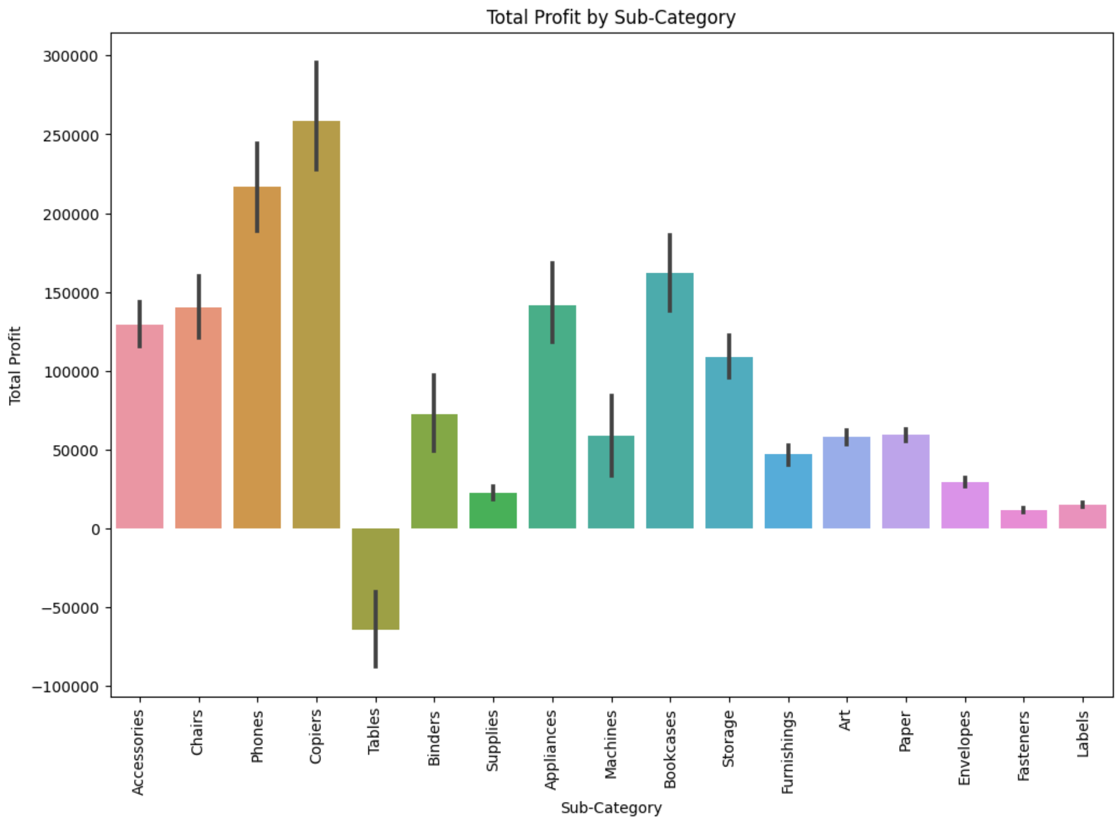
[31]:
```
# Grouping products by Category and Sub-Category
cat_subcat = pd.DataFrame(df.groupby(['Category', 'Sub-Category'])['Profit'].
 ↪sum())

# Sorting the values
cat_subcat.sort_values(['Category','Profit'], ascending=False)
```

[31]:
```
                                      Profit
Category        Sub-Category
Technology      Copiers         258567.54818
                Phones          216717.00580
                Accessories     129626.30620
                Machines         58867.87300
Office Supplies Appliances      141680.58940
                Storage         108461.48980
                Binders          72449.84600
                Paper            59207.68270
                Art              57953.91090
                Envelopes        29601.11630
                Supplies         22583.26310
                Labels           15010.51200
                Fasteners        11525.42410
Furniture       Bookcases       161924.41950
                Chairs          140396.26750
                Furnishings      46967.42550
                Tables          -64083.38870
```

-

Graphical values

[39]:
```
# Visualization 4: Profit by Sub-Category
plt.figure(figsize=(12, 8))
sns.barplot(x='Sub-Category', y='Profit', data=df, estimator=sum)
plt.title('Total Profit by Sub-Category')
plt.xlabel('Sub-Category')
plt.ylabel('Total Profit')
plt.xticks(rotation=90)
plt.show()
```



here we can see that tablets are loss making for us so its better for the company to stop selling tablets as we are losing money in those