

# Superstore Sales Data Analysis

August 19, 2023

```
[4]: #importing packages
```

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

```
[5]: # importing dataset
```

```
df = pd.read_csv(r"/Users/scipio/Downloads/Sales_Dataset_Project.csv")

#converting 'Order Date' column to datatiem format
df['Order Date'] = pd.to_datetime(df['Order Date'])

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 9800 entries, 0 to 9799
```

```
Data columns (total 18 columns):
```

#	Column	Non-Null Count	Dtype
0	Row ID	9800 non-null	int64
1	Order ID	9800 non-null	object
2	Order Date	9800 non-null	datetime64[ns]
3	Ship Date	9800 non-null	object
4	Ship Mode	9800 non-null	object
5	Customer ID	9800 non-null	object
6	Customer Name	9800 non-null	object
7	Segment	9800 non-null	object
8	Country	9800 non-null	object
9	City	9800 non-null	object
10	State	9800 non-null	object
11	Postal Code	9789 non-null	float64
12	Region	9800 non-null	object
13	Product ID	9800 non-null	object
14	Category	9800 non-null	object
15	Sub-Category	9800 non-null	object
16	Product Name	9800 non-null	object
17	Sales	9800 non-null	float64

```
dtypes: datetime64[ns](1), float64(2), int64(1), object(14)
memory usage: 1.3+ MB
```

```
/var/folders/3k/bzmghyyj1j51lkx1mc36njw0000gn/T/ipykernel_29522/4161371504.py:6
: UserWarning: Parsing dates in DD/MM/YYYY format when dayfirst=False (the
default) was specified. This may lead to inconsistently parsed dates! Specify a
format to ensure consistent parsing.
    df['Order Date'] = pd.to_datetime(df['Order Date'])
```

## 1 Objective

Data Analysis of the sales data of a global superstore. The analysis will be guided by the following questions:

1. What was the most profitable region, state, and city in the dataset?
2. What was the most profitable category and sub category in the dataset?
3. What is the most profitable product in the dataset?
4. What was the most popular shipping method in the dataset?
5. What was the most profitable year in the dataset?

### 1.1 Analysis

#### 1.1.1 1. What was the most profitable region, state, and city in the dataset?

```
[6]: #Region Sales Total
region_sales_totals = round(df.groupby('Region')['Sales'].sum(),2)

#sorting results
print(region_sales_totals.sort_values(ascending = False))
```

```
Region
West      710219.68
East      669518.73
Central   492646.91
South     389151.46
Name: Sales, dtype: float64
```

```
[7]: #Percentage Calculation
Region_Total_Sales_Pct = round(df.groupby('Region')['Sales'].sum()/df['Sales'].
    ↪sum(),2)

#Sorting Values
Region_Total_Sales_Pct.sort_values(ascending = False).head()
```

```
[7]: Region
West      0.31
East      0.30
Central   0.22
South     0.17
```

Name: Sales, dtype: float64

```
[8]: #Most profitable state in West Region

#filtering for West region
West_State_Sales_Total = round(df[df['Region'] == 'West'].
    ↳groupby('State')['Sales'].sum(),2)

#sorting results
West_State_Sales_Total.sort_values(ascending = False).head()
```

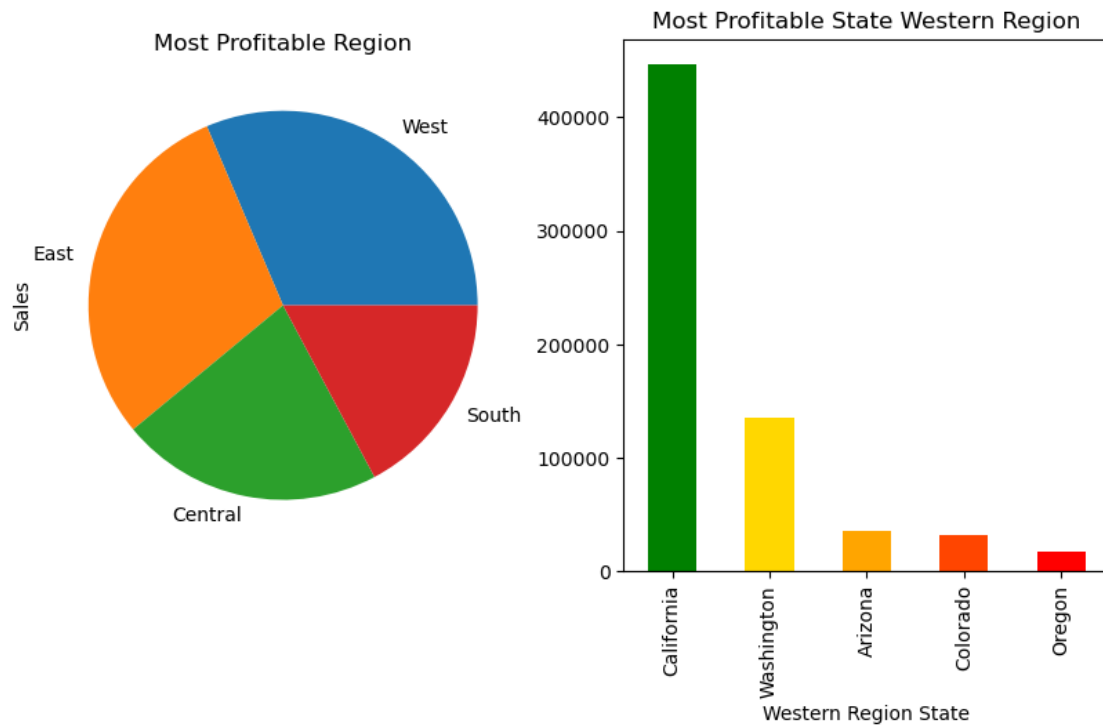
```
[8]: State
California    446306.46
Washington    135206.85
Arizona       35272.66
Colorado      31841.60
Oregon        17284.46
Name: Sales, dtype: float64
```

```
[9]: # Creating Subplots
fig,axs = plt.subplots(nrows=1,ncols=2, figsize = (10,5))

#Color List
colors_5 = ['Green','Gold','Orange', 'OrangeRed', 'Red']
colors_4 = ['Green','Gold','Orange','Red']

#Subplots
region_sales_totals.sort_values(ascending = False).head(5).plot(kind = 'pie',
    ↳ax = axs[0], title = 'Most Profitable Region')
West_State_Sales_Total.sort_values(ascending = False).head(5).plot(kind =
    ↳'bar', ax = axs[1], title = 'Most Profitable State Western Region', xlabel =
    ↳'Western Region State', color=colors_5)
```

```
[9]: <Axes: title={'center': 'Most Profitable State Western Region'}, xlabel='Western
Region State'>
```



The West region was the most profitable region in the dataset, totaling 710,219.68 USD in sales, 31% of total sales in the dataset. California was the most profitable state in the West region, totaling 446,306.46 USD in sales, accounting for 63% of the Western region's total sales.

```
[10]: # Most profitable state
State_Total_Sales = round(df.groupby('State')['Sales'].sum(),2)

#Sorting Results
State_Total_Sales.sort_values(ascending = False).head(5)
```

```
[10]: State
California      446306.46
New York        306361.15
Texas           168572.53
Washington      135206.85
Pennsylvania    116276.65
Name: Sales, dtype: float64
```

```
[11]: #State Total Sales Percentage
State_Sales_Total_Pct = round(df.groupby('State')['Sales'].sum()/df['Sales'].
    ↪sum(),2)

#Sorting Percentages
State_Sales_Total_Pct.sort_values(ascending = False).head(5)
```

```
[11]: State
      California      0.20
      New York       0.14
      Texas          0.07
      Washington     0.06
      Pennsylvania   0.05
      Name: Sales, dtype: float64
```

```
[12]: #Most profitable city in California
      Most_Profitable_City_Cali = round(df[df['State']=='California'].
      ↪groupby('City')['Sales'].sum(),2)

      # sorting values
      Most_Profitable_City_Cali.sort_values(ascending = False).head()
```

```
[12]: City
      Los Angeles      173420.18
      San Francisco    109041.12
      San Diego        47521.03
      Fresno           7888.53
      Sacramento       7311.28
      Name: Sales, dtype: float64
```

```
[13]: #California City Sales Percentage
      California_City_Sales_Pct = round(df[df['State']=='California'].
      ↪groupby('City')['Sales'].sum()/df[df['State']=='California']['Sales'].
      ↪sum(),2)

      #Sorting Values
      California_City_Sales_Pct.sort_values(ascending = False).head(5)
```

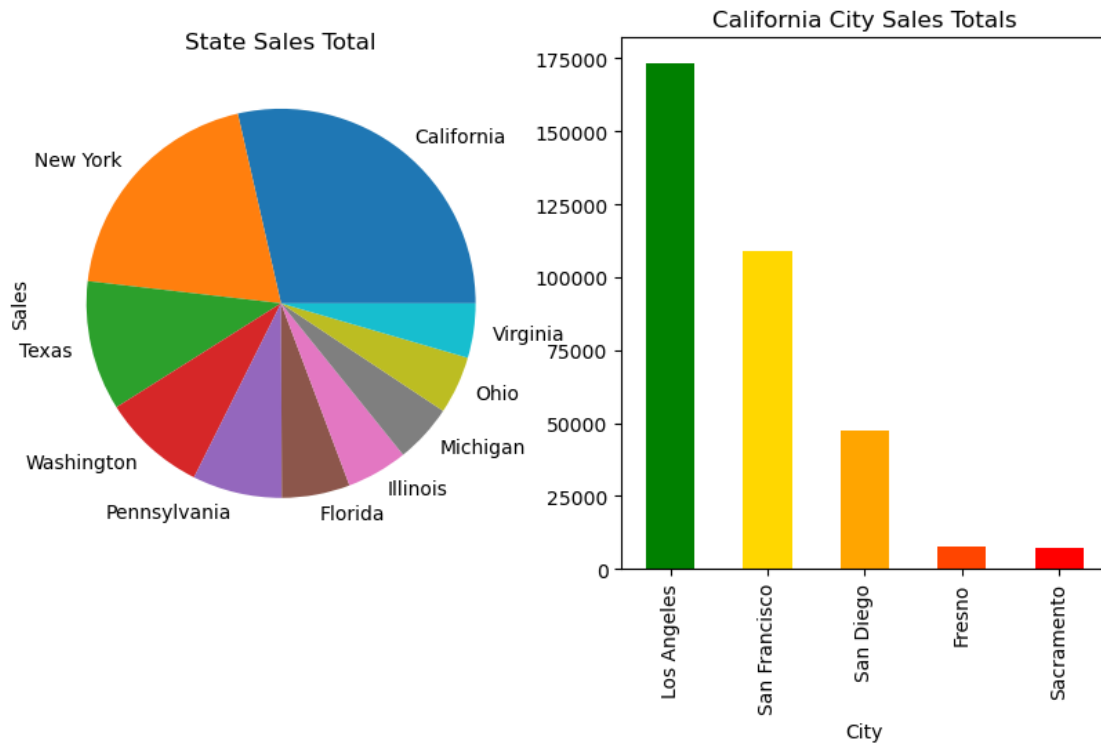
```
[13]: City
      Los Angeles      0.39
      San Francisco    0.24
      San Diego        0.11
      Fresno           0.02
      Sacramento       0.02
      Name: Sales, dtype: float64
```

```
[14]: fig,axs = plt.subplots(nrows=1, ncols=2, figsize = (10,5))

      colors_5 = ['Green','Gold','Orange', 'OrangeRed', 'Red']

      State_Total_Sales.sort_values(ascending = False).head(10).plot(kind = 'pie',
      ↪title = 'State Sales Total', ax = axs[0])
      Most_Profitable_City_Cali.sort_values(ascending = False).head().plot(kind =
      ↪'bar', color = colors_5, title = 'California City Sales Totals')
```

```
[14]: <Axes: title={'center': 'California City Sales Totals'}, xlabel='City'>
```



California was the most profitable state in the dataset, totaling 446,306.46 USD in sales, accounting for 20% of the total sales. Los Angeles was the most profitable city in California, totaling 173,420.18 USD in sales, accounting for 39% of the California sales total.

```
[15]: # Most profitable city in the dataset
City_Sales_Totals = round(df.groupby('City')['Sales'].sum(),2)

#sorting values
City_Sales_Totals.sort_values(ascending = False).head()
```

```
[15]: City
New York City    252462.55
Los Angeles     173420.18
Seattle          116106.32
San Francisco    109041.12
Philadelphia     108841.75
Name: Sales, dtype: float64
```

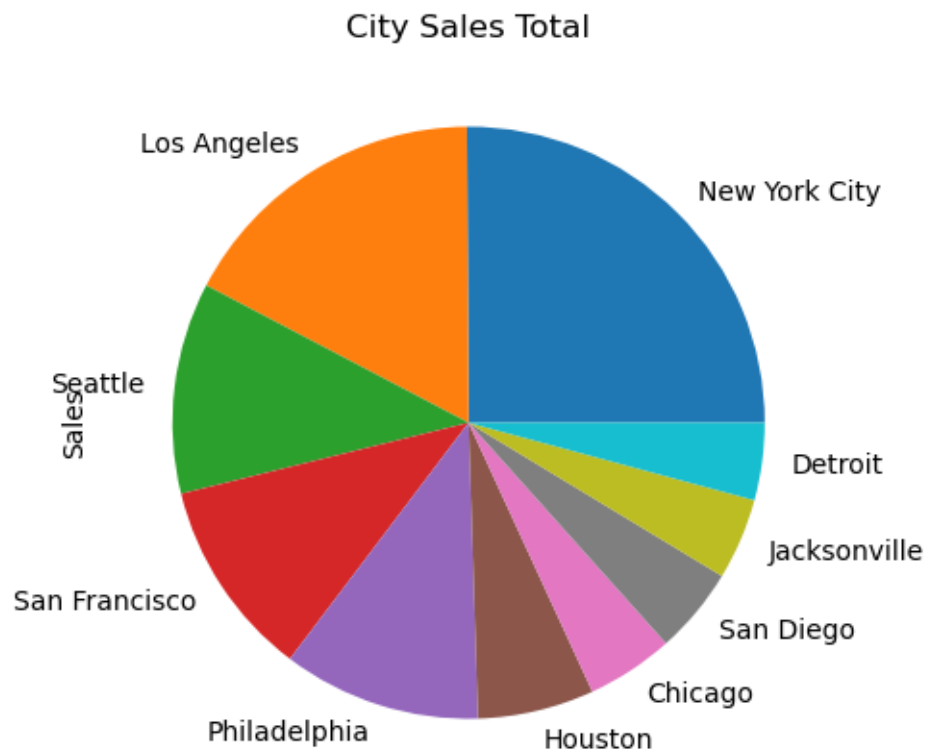
```
[16]: #City Total Sales Percentage
City_Total_Sales_Pct = round(df.groupby('City')['Sales'].sum()/df['Sales'].
    ↪sum(),2)
```

```
#Sorting Values
City_Total_Sales_Pct.sort_values(ascending = False).head()
```

```
[16]: City
New York City    0.11
Los Angeles      0.08
Seattle          0.05
San Francisco    0.05
Philadelphia     0.05
Name: Sales, dtype: float64
```

```
[17]: City_Sales_Totals.sort_values(ascending = False).head(10).plot(kind = 'pie',
↳ title = 'City Sales Total', figsize = (10,5))
```

```
[17]: <Axes: title={'center': 'City Sales Total'}, ylabel='Sales'>
```



New York City was the most profitable city in the dataset, totaling 252,462.55 USD in sales, accounting for 11% of total sales.

### 1.1.2 2. What was the most profitable category and sub category in the dataset?

```
[18]: # Category Total Sales
Category_Total_Sales = round(df.groupby('Category')['Sales'].sum(),2)

#Sorting
Category_Total_Sales.sort_values(ascending = False).head()
```

```
[18]: Category
Technology      827455.87
Furniture       728658.58
Office Supplies 705422.33
Name: Sales, dtype: float64
```

```
[19]: #Category Total Sales Percentage
Category_Total_Sales_Pct = round(df.groupby('Category')['Sales'].sum()/
    ↪df['Sales'].sum(),2)

#Sorting
Category_Total_Sales_Pct.sort_values(ascending = False).head()
```

```
[19]: Category
Technology      0.37
Furniture       0.32
Office Supplies 0.31
Name: Sales, dtype: float64
```

```
[20]: #Most Profitable Product in Technology
Tech_Category_Profitable_Product = round(df[df['Category']== 'Technology'].
    ↪groupby('Product Name')['Sales'].sum(),2)

#Sorting
Tech_Category_Profitable_Product.sort_values(ascending = False).head()
```

```
[20]: Product Name
Canon imageCLASS 2200 Advanced Copier      61599.82
Cisco TelePresence System EX90 Videoconferencing Unit 22638.48
Hewlett Packard LaserJet 3310 Copier        18839.69
HP Designjet T520 Inkjet Large Format Printer - 24" Color 18374.90
Lexmark MX611dhe Monochrome Laser Printer  16829.90
Name: Sales, dtype: float64
```

```
[21]: #Most Profitable Product in Technology Percentage
Tech_Category_Profitable_Product_Pct = round(df[df['Category']== 'Technology'].
    ↪groupby('Product Name')['Sales'].sum()/df[df['Category']==
    ↪'Technology']['Sales'].sum(),2)

#Sorting
```



```
Tech_Category_Profitable_Product_Pct.sort_values(ascending = False).head()
```

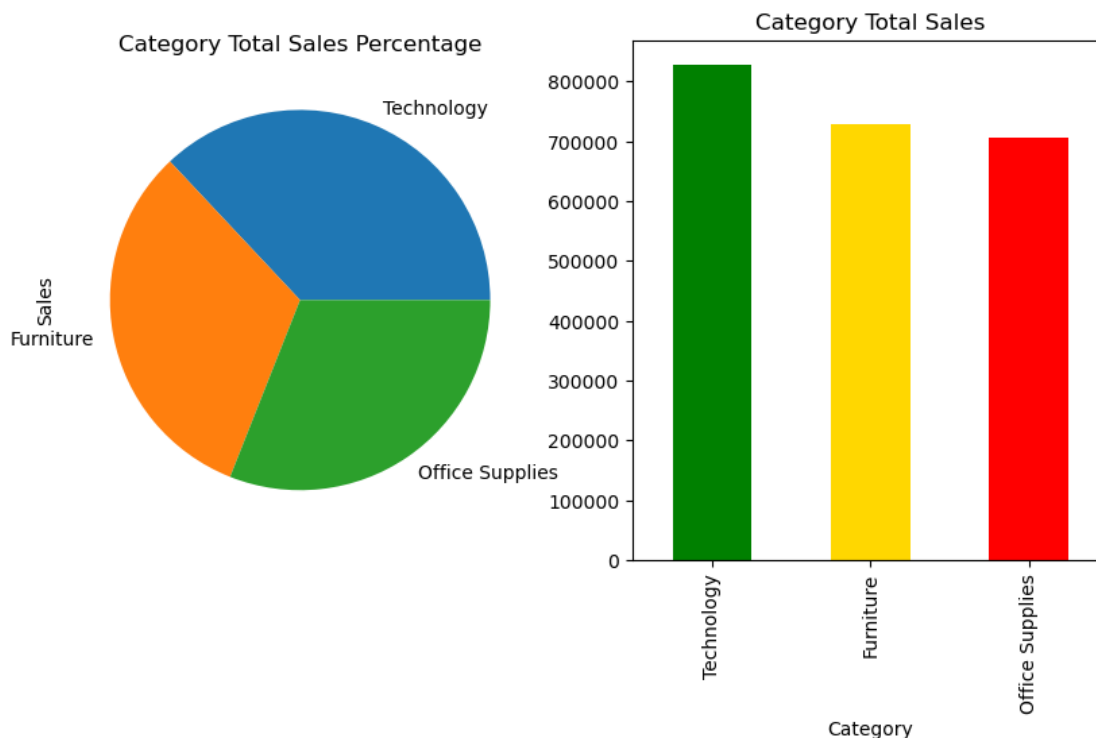
```
[21]: Product Name
      Canon imageCLASS 2200 Advanced Copier        0.07
      Cisco TelePresence System EX90 Videoconferencing Unit  0.03
      3D Systems Cube Printer, 2nd Generation, Magenta    0.02
      Samsung Galaxy Mega 6.3                        0.02
      Lexmark MX611dhe Monochrome Laser Printer         0.02
      Name: Sales, dtype: float64
```

```
[22]: #Subplots
fig,axs = plt.subplots(nrows = 1, ncols =2, figsize = (10,5))

colors_3= ['Green','Gold','Red']

Category_Total_Sales.sort_values(ascending = False).plot(kind = 'bar', ax =_
↳axs[1], title = 'Category Total Sales', color = colors_3)
Category_Total_Sales_Pct.sort_values(ascending = False).plot(kind = 'pie', ax =_
↳axs[0], title= 'Category Total Sales Percentage')
```

```
[22]: <Axes: title={'center': 'Category Total Sales Percentage'}, ylabel='Sales'>
```



Technology was the most profitable category, totaling 827,455.87 USD in sales, accounting for 37% of total sales. The *Canon imageCLASS 2200 Advanced Copier* was the most profitable product

in the Technology category totaling 61,599.82 USD in sales, accounting for 7% of the Technology category sales.

```
[23]: #Sub Category Total Sales
Sub_Category_Sales = round(df.groupby('Sub-Category')['Sales'].sum(),2)

#Sorting
Sub_Category_Sales.sort_values(ascending = False).head()
```

```
[23]: Sub-Category
Phones      327782.45
Chairs      322822.73
Storage     219343.39
Tables      202810.63
Binders     200028.78
Name: Sales, dtype: float64
```

```
[24]: #Sub Category Total Sales Percentage
Sub_Category_Sales_Pct = round(df.groupby('Sub-Category')['Sales'].sum()/
    ↪df['Sales'].sum(),3)

#Sorting
Sub_Category_Sales_Pct.sort_values(ascending = False)
```

```
[24]: Sub-Category
Phones      0.145
Chairs      0.143
Storage     0.097
Tables      0.090
Binders     0.088
Machines    0.084
Accessories 0.073
Copiers     0.065
Bookcases   0.050
Appliances  0.046
Furnishings 0.039
Paper       0.034
Supplies    0.021
Art         0.012
Envelopes   0.007
Labels      0.005
Fasteners   0.001
Name: Sales, dtype: float64
```

```
[25]: # Most profitable product in sub-category phones
Sub_Cat_Phones_Sales_Total = round(df[df['Sub-Category']=='Phones'].
    ↪groupby('Product Name')['Sales'].sum(),2)
```

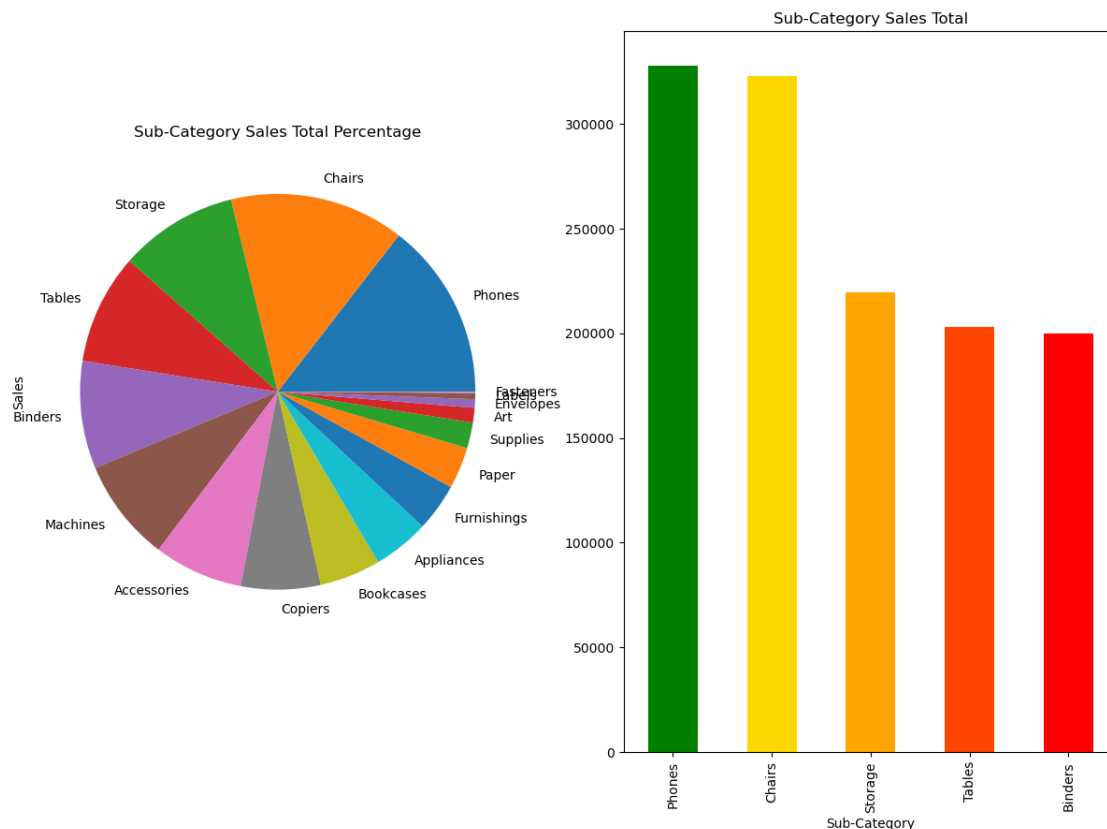
```
#Sorting
```

```
Sub_Cat_Phones_Sales_Total.sort_values(ascending = False).head()
```

```
[25]: Product Name
Samsung Galaxy Mega 6.3      13943.67
Apple iPhone 5               12996.60
Wilson Electronics DB Pro Signal Booster    8878.40
Mitel MiVoice 5330e IP Phone    7699.72
Samsung Galaxy S III - 16GB - pebble blue (T-Mobile)  7139.80
Name: Sales, dtype: float64
```

```
[26]: fig,axs = plt.subplots(nrows = 1, ncols =2, figsize = (15,10))
Sub_Category_Sales_Pct.sort_values(ascending = False).plot(kind = 'pie',ax =_
    ↪axs[0], title = 'Sub-Category Sales Total Percentage')
Sub_Category_Sales.sort_values(ascending = False).head().plot(kind = 'bar', ax_
    ↪= axs[1], color = colors_5, title = 'Sub-Category Sales Total' )
```

```
[26]: <Axes: title={'center': 'Sub-Category Sales Total'}, xlabel='Sub-Category'>
```



Phones was the most profitable Sub-Category in the dataset, totaling 327,782.45 USD in sales, accounting for nearly 15% of the Phones Sub-Category sales totals. The Samsung Galaxy Mega

6.3 was the most popular product in the Sub-Category Phones, totaling 13,943.67 USD in sales.

### 1.1.3 3. What is the most profitable product in the dataset?

```
[27]: # Product total sales
Product_Sales_Total = round(df.groupby('Product Name')['Sales'].sum(),2)

#Sorting
Product_Sales_Total.sort_values(ascending = False).head(1)
```

```
[27]: Product Name
Canon imageCLASS 2200 Advanced Copier    61599.82
Name: Sales, dtype: float64
```

The Canon imageCLASS 2200 Advanced Copier was the most profitable product in the dataset, totaling 61,599.82 USD in sales.

### 1.1.4 4. What was the most popular shipping method in the dataset?

```
[28]: #Ship Mode Totals
Ship_Mode_Totals = df.groupby('Ship Mode')['Ship Mode'].count()

#sorting
Ship_Mode_Totals.sort_values(ascending = False)
```

```
[28]: Ship Mode
Standard Class    5859
Second Class     1902
First Class      1501
Same Day         538
Name: Ship Mode, dtype: int64
```

```
[29]: # Ship Mode Totals Percentage
Ship_Mode_Totals_Percentage = round(df.groupby('Ship Mode')['Ship Mode'].
    ↪count()/df['Ship Mode'].count(),2)

#Sorting
Ship_Mode_Totals_Percentage.sort_values(ascending = False)
```

```
[29]: Ship Mode
Standard Class    0.60
Second Class     0.19
First Class      0.15
Same Day         0.05
Name: Ship Mode, dtype: float64
```

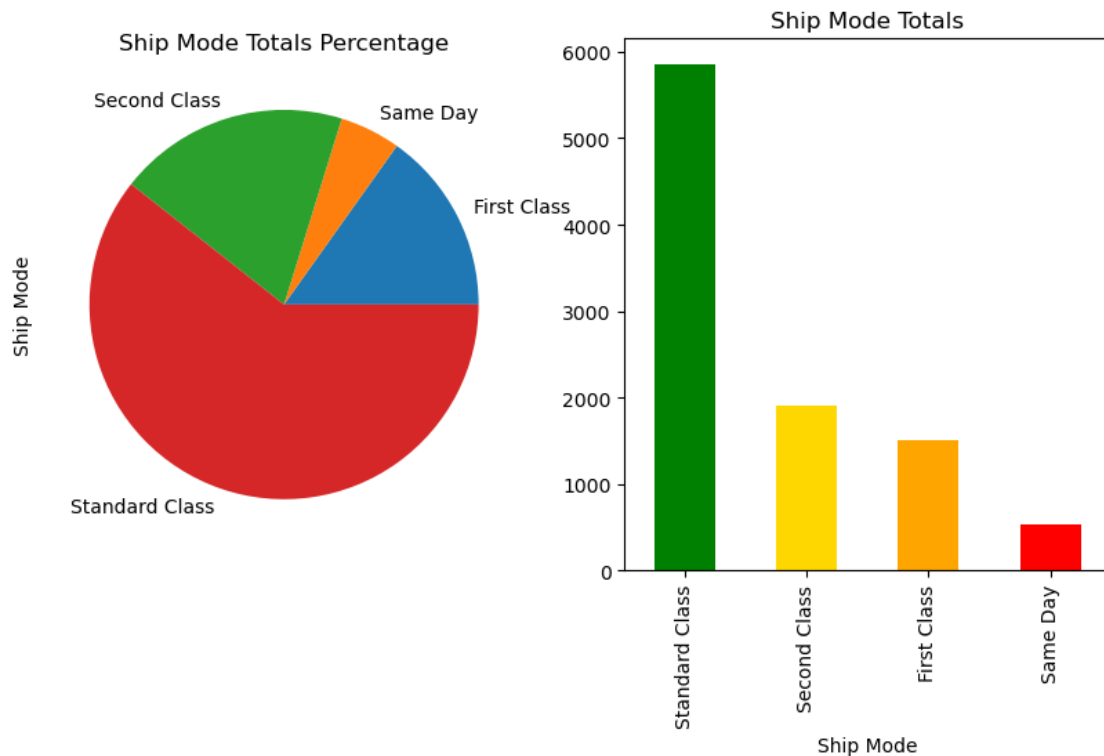
```
[31]: # subplot
fig,axs = plt.subplots(nrows = 1, ncols =2,figsize = (10,5))
```

```

Ship_Mode_Totals.sort_values(ascending = False).plot(kind = 'bar', title = 'Ship Mode Totals', ax = axs[1], color = colors_4)
Ship_Mode_Totals_Percentage.plot(kind = 'pie', title = 'Ship Mode Totals Percentage', ax = axs[0])

```

[31]: <Axes: title={'center': 'Ship Mode Totals Percentage'}, ylabel='Ship Mode'>



The most popular shipping method was Standard Class Shipping. 5859, 60%, of orders were sent to customers via Standard Class Shipping.

### 1.1.5 5. What was the most profitable year in the dataset?

```

[42]: #Creating a Year column
df['Year'] = df['Order Date'].dt.year

# Year Total Sales
Year_Total_Sales = round(df.groupby('Year')['Sales'].sum(), 2)

#Sorting
Year_Total_Sales.sort_values(ascending = False)

```

```
[42]: Year
      2018    722052.02
      2017    600192.55
      2015    479856.21
      2016    459436.01
      Name: Sales, dtype: float64
```

```
[73]: #Year Total Sale Percentages
Year_Total_Sales_Percentage = round(df.groupby('Year')['Sales'].sum()/
    ↪df['Sales'].sum(),2)

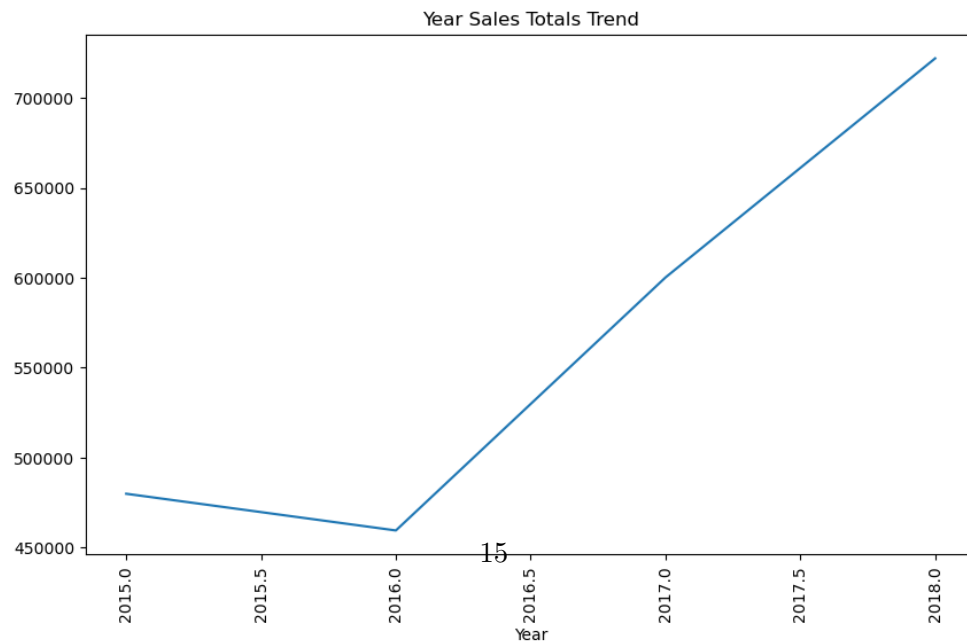
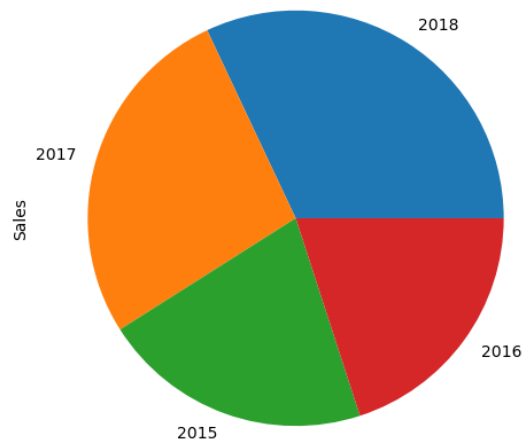
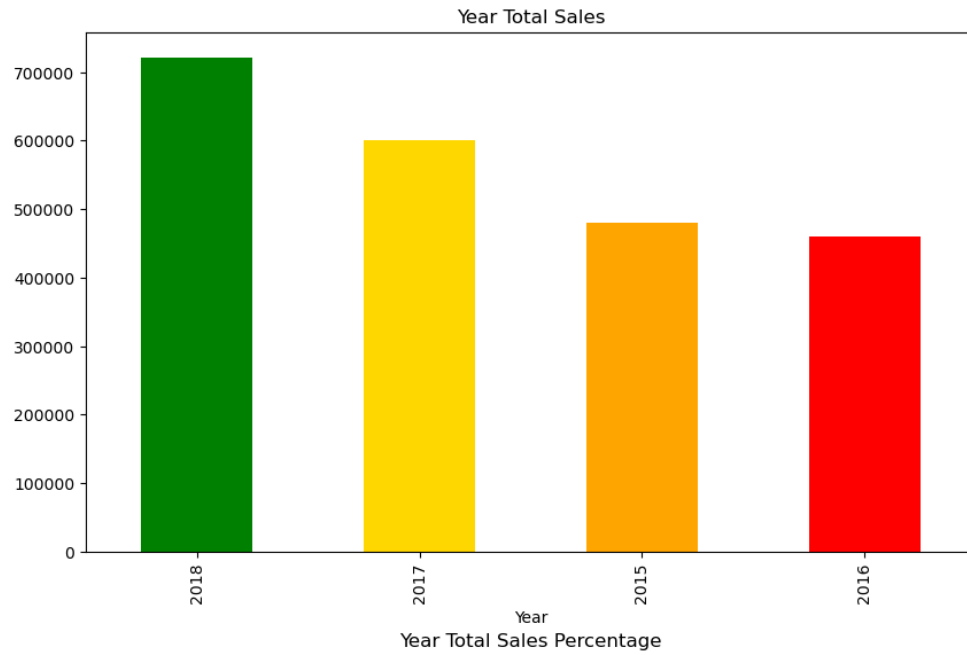
#Sorting
Year_Total_Sales_Percentage.sort_values(ascending = False)
```

```
[73]: Year
      2018     0.32
      2017     0.27
      2015     0.21
      2016     0.20
      Name: Sales, dtype: float64
```

```
[75]: #Subplots
fig, axs = plt.subplots(nrows = 3, figsize = (10,20))

Year_Total_Sales.sort_values(ascending = False).plot(kind = 'bar', title =
    ↪'Year Total Sales', color = colors_4, ax = axs [0])
Year_Total_Sales_Percentage.sort_values(ascending = False).plot(kind = 'pie',
    ↪ax = axs[1], title = 'Year Total Sales Percentage')
df.groupby('Year')['Sales'].sum().plot(kind = 'line', title = 'Year Sales
    ↪Totals Trend', ax = axs[2], rot = 90)
```

```
[75]: <Axes: title={'center': 'Year Sales Totals Trend'}, xlabel='Year'>
```



2018 was the most profitable year in the dataset, totaling 722,052.02 USD in sales, accounting for nearly a third of total sales. Additionally, there was a positive trend in total sales in the dataset. On average there was a 60,548.95 USD, 12%, year over year (YoY) increase in sales.

#### **1.1.6 Conclusion**

Overall there was a positive trend in sales YoY with an average increase of 60,548.95 USD, 12%, YoY in total sales. 2018 was the most profitable year of sales in the dataset totaling 722,052.02 USD in sales. Additionally, the Western region was the most profitable region in the dataset, totaling 710,219.68 USD in sales, accounting for 31% of total sales. California was the most profitable state while New York City was the most profitable city in the dataset. The Technology category was the most profitable category in the dataset, accounting for 37% of total sales. Phones were the most profitable subcategory, totaling 327,782.45 USD in sales. The Canon imageCLASS 2200 Advanced Copier was the most profitable product in the dataset, totaling 61,599.82 USD in sales. Lastly, Standard Class Shipping was the most popular method of shipping orders, 60% of all orders were shipped using Standard Class Shipping.