SALES ANALYSIS USING PYTHON - Jupyter

OBJECTIVE

To find the following:

1.***overall sales trend***

2.***Top*** 10 ***products by sales***.

3.***Most Selling Products***

4.***Most preferred Shipping Mode***

5.***Most Profitable Category and Sub-Category***

THE LIBRARIES USED

1. ***Pandas*** - ***For data manipulation***, ***exploritative data analysis***.
2. ***Matplotlib and Seaborn*** - ***For Data Visualization***
3. %***matplotlib inline*** - ***For inlining to display the output of plotting commands inline within frontends***.

DATASET USED : US SUPERSTORE SALES

In [8]:

```
#importing the libraries
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

In [4]:

```
#Creation of dataframe by using the data-set which is in excel.
#importing the excel file.
df = pd.read_excel("E:\\DATA SCIENCE\\Project\\python\\Data Analysis\\Sales\\superstore
```

Exploratory_DataAnalysis

In [7]:

```
#Let's Display the Top 5 rows of the data-frame.

df.head()
```

Out[7]:

order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market
0	AG-2011-2040	2011-01-01	2011-01-06	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	Africa
1	IN-2011-47883	2011-01-01	2011-01-08	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC
2	HU-2011-1220	2011-01-01	2011-01-05	Second Class	Annie Thurman	Consumer	Budapest	Hungary	EMEA
3	IT-2011-3647632	2011-01-01	2011-01-05	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	EU
4	IN-2011-47883	2011-01-01	2011-01-08	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC

5 rows × 21 columns

The Number of Rows and Columns that are available in this data-set.

In [10]:
```
df.shape
```

Out[10]:
```
(51290, 21)
```

FINDINGS - Rows : 21 Columns : 51,290

Summary Of Sales-Dataset.

1. Number of Non-Null Values in each column.
2. Data-Type of each column.
3. Memory used.

In [12]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51290 entries, 0 to 51289
Data columns (total 21 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   order_id        51290 non-null  object
 1   order_date      51290 non-null  datetime64[ns]
 2   ship_date       51290 non-null  datetime64[ns]
 3   ship_mode       51290 non-null  object
 4   customer_name   51290 non-null  object
 5   segment         51290 non-null  object
 6   state           51290 non-null  object
 7   country         51290 non-null  object
 8   market          51290 non-null  object
 9   region          51290 non-null  object
 10  product_id      51290 non-null  object
 11  category        51290 non-null  object
 12  sub_category    51290 non-null  object
 13  product_name    51290 non-null  object
 14  sales           51290 non-null  float64
 15  quantity        51290 non-null  int64
 16  discount        51290 non-null  float64
 17  profit          51290 non-null  float64
 18  shipping_cost   51290 non-null  float64
 19  order_priority  51290 non-null  object
 20  year            51290 non-null  int64
dtypes: datetime64[ns](2), float64(4), int64(2), object(13)
memory usage: 8.2+ MB
```

Number of Null_Values in each column?

In [13]:

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

Out[13]:

```
order_id          0
order_date        0
ship_date         0
ship_mode         0
customer_name     0
segment           0
state             0
country           0
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
year              0
dtype: int64
```

Findings : There are no null_values in the dataset.

Date Of Entry of first data into data_set?

In [14]:

```
df['order_date'].min()
```

Out[14]:

```
Timestamp('2011-01-01 00:00:00')
```

FINDINGS : On 1st January of 2011 the 1st data was entered.

Date Of Entry of last data into data_set?

In [15]:
```
df['order_date'].max()
```

Out[15]:
```
Timestamp('2014-12-31 00:00:00')
```

FINDINGS : On 31st December 2014 the last data was entered.

Month of Order Date from the data_set.

In [21]:
```
df['month'] = df['order_date'].apply(lambda x: x.strftime('%m'))
#apply is for applyig the condition to each row.
#strftime is to convert date object to string representation.
```

In [22]:
```
df['month']
```

Out[22]:
```
0        01
1        01
2        01
3        01
4        01
         ..
51285    12
51286    12
51287    12
51288    12
51289    12
Name: month, Length: 51290, dtype: object
```

TOP 10 Products Based On sales.

In [59]:
```
#Grouping 'Products' based on 'Sales'.
prod_bySales = pd.DataFrame(df.groupby('product_name').sum(numeric_only=True)['sales'])

#Sorting the sales in descending order.
#Fuction 'sort_values' - To sort the data in asc/desc order) Of Passed Columns.
prod_bySales.sort_values(by=['sales'],ascending=False, inplace = True )

#TOP10 Products
prod_bySales[:10]
```

Out[59]:

	sales
product_name	
Apple Smart Phone, Full Size	86935.7786
Cisco Smart Phone, Full Size	76441.5306
Motorola Smart Phone, Full Size	73156.3030
Nokia Smart Phone, Full Size	71904.5555

product_name	sales
Canon imageCLASS 2200 Advanced Copier	61599.8240
Hon Executive Leather Armchair, Adjustable	58193.4841
Office Star Executive Leather Armchair, Adjustable	50661.6840
Harbour Creations Executive Leather Armchair, Adjustable	50121.5160
Samsung Smart Phone, Cordless	48653.4600
Nokia Smart Phone, with Caller ID	47877.7857

Most Sold Products

In [61]:
```
#grouping the products based on Sold Quantity.
most_sold_products = pd.DataFrame(df.groupby('product_name').sum(numeric_only=True)['qu

#sorting the values in descending order by using the 'sort_values' function and making
most_sold_products.sort_values(by=['quantity'],ascending=False, inplace = True)

#Top10
most_sold_products[:10]
```

Out[61]:

product_name	quantity
Staples	876
Cardinal Index Tab, Clear	337
Eldon File Cart, Single Width	321
Rogers File Cart, Single Width	262
Sanford Pencil Sharpener, Water Color	259
Stockwell Paper Clips, Assorted Sizes	253
Avery Index Tab, Clear	252
Ibico Index Tab, Clear	251
Smead File Cart, Single Width	250
Stanley Pencil Sharpener, Water Color	242

Most Profitable Products .

In [66]:
```
#grouping the products based on Profit.
most_profitable_product = pd.DataFrame(df.groupby('product_name').sum(numeric_only=True

#sorting the values in descending order by using the 'sort_values' function and making
most_profitable_product.sort_values(by=['profit'],ascending=False, inplace = True)
```

```
#Top
most_sold_products[:1]
```

Out[66]:

	profit
product_name	
Canon imageCLASS 2200 Advanced Copier	25199.928

Most Preferred mode of shipment.

In [73]:

```
#grouping the products based on number of quantity.
mst_prfrd_shipmnt = pd.DataFrame(df.groupby('ship_mode').sum(numeric_only=True)['quanti

#sorting the values is descending order.
mst_prfrd_shipmnt.sort_values(by=['quantity'],ascending=False, inplace = True)

mst_prfrd_shipmnt[:5]
```

Out[73]:

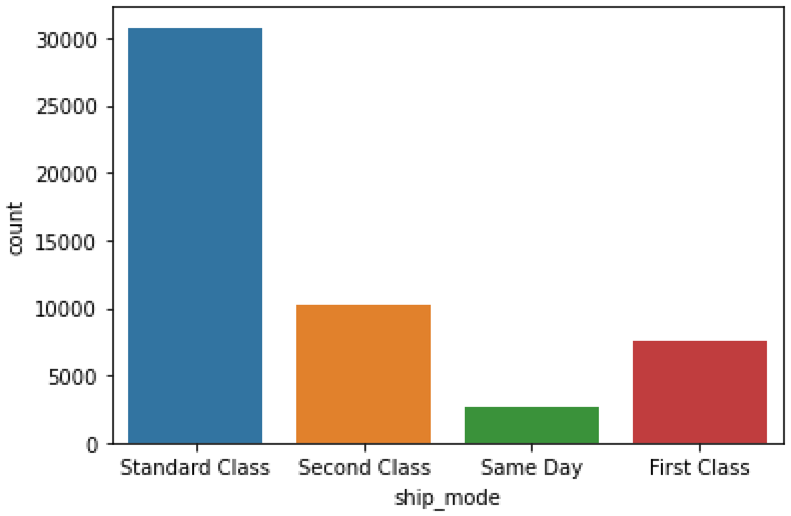
	quantity
ship_mode	
Standard Class	107319
Second Class	35724
First Class	26039
Same Day	9230

Most Preferred mode-of-shipment by Visualization.

we will use seaborn library here to count as well as visualize.

In [84]:

```
sns.countplot(x='ship_mode', data=df) #countplot :counts the no of observation in each
plt.figure(figsize=(10,10)) #figsize: depicts the size of plot.
plt.show() #displaying the plot.
```



```
<Figure size 720x720 with 0 Axes>
```

Statistical Summary of Whole Dataset

In [96]:

```
# describe method gives descriptive statistics of the data frame.Tt only shows the stat

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

Out[96]:

	sales	quantity	discount	profit	shipping_cost	year
count	51290.0	51290.0	51290.0	51290.0	51290.0	51290.0
mean	246.0	3.0	0.0	29.0	26.0	2013.0
std	488.0	2.0	0.0	174.0	57.0	1.0
min	0.0	1.0	0.0	-6600.0	0.0	2011.0
25%	31.0	2.0	0.0	0.0	3.0	2012.0
50%	85.0	3.0	0.0	9.0	8.0	2013.0
75%	251.0	5.0	0.0	37.0	24.0	2014.0
max	22638.0	14.0	1.0	8400.0	934.0	2014.0

In [ ]: