PANDAS SALES ANALYSIS

Objective

Upon initial inspection of data, we can start thinking of some questions about it that it would be answer.

- what is the over all sales trend?
- what are the top 10 products by sales?
- what are the most selling products?
- which is the most preffered ship mode?
- which are the most profitable category and sub category?

importing required libraries

In [3]:
```
# Data Manipulation

import pandas as pd

# Data Visualization

import matplotlib.pyplot as plt
%matplotlib inline

import seaborn as sns
```

IMPORTING THE DATASET

In [6]:
```
df = pd.read_excel("C:\\Users\\naren\\Desktop\\superstore_sales.xlsx")
```

DATA AUDIT

In [7]:
```
# First five rows of the Dataset

df.head()
```

Out[7]:

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market	region	...	category	sub_category	product_name	sales	quantity	discount
0	AG-2011-2040	2011-01-01	2011-01-06	Standard Class	Toby Braunhardt	Consumer	Constantine	Algeria	Africa	Africa	...	Office Supplies	Storage	Tenex Lockers, Blue	408.300	2	0.0
1	IN-2011-47883	2011-01-01	2011-01-08	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC	Oceania	...	Office Supplies	Supplies	Acme Trimmer, High Speed	120.366	3	0.1
2	HU-2011-1220	2011-01-01	2011-01-05	Second Class	Annie Thurman	Consumer	Budapest	Hungary	EMEA	EMEA	...	Office Supplies	Storage	Tenex Box, Single Width	66.120	4	0.0
3	IT-2011-3647632	2011-01-01	2011-01-05	Second Class	Eugene Moren	Home Office	Stockholm	Sweden	EU	North	...	Office Supplies	Paper	Enermax Note Cards, Premium	44.865	3	0.5
4	IN-2011-47883	2011-01-01	2011-01-08	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC	Oceania	...	Furniture	Furnishings	Eldon Light Bulb, Duo Pack	113.670	5	0.1

5 rows × 21 columns

In [8]:
```
# Last five rows of the Dataset

df.tail()
```

Out[8]:

	order_id	order_date	ship_date	ship_mode	customer_name	segment	state	country	market	region	...	category	sub_category	product_name	sales	quantity	discount
51285	CA-2014-115427	2014-12-31	2015-01-04	Standard Class	Erica Bern	Corporate	California	United States	US	West	...	Office Supplies	Binders	Cardinal Slant-D Ring Binder, Heavy Gauge Vinyl	13.904	2	0.2
51286	MO-2014-2060	2014-12-31	2015-01-05	Standard Class	Liz Preis	Consumer	Souss-Massa-Draâ	Morocco	Africa	Africa	...	Office Supplies	Binders	Wilson Jones Hole Reinforcements, Clear	3.990	1	0.0
51287	MX-2014-110527	2014-12-31	2015-01-02	Second Class	Charlotte Melton	Consumer	Managua	Nicaragua	LATAM	Central	...	Office Supplies	Labels	Hon Color Coded Labels, 5000 Label Set	26.400	3	0.0
51288	MX-2014-114783	2014-12-31	2015-01-06	Standard Class	Tamara Dahlen	Consumer	Chihuahua	Mexico	LATAM	North	...	Office Supplies	Labels	Hon Legal Exhibit Labels, Alphabetical	7.120	1	0.0
51289	CA-2014-156720	2014-12-31	2015-01-04	Standard Class	Jill Matthias	Consumer	Colorado	United States	US	West	...	Office Supplies	Fasteners	Bagged Rubber Bands	3.024	3	0.0

5 rows × 21 columns

In [10]:
```
# Shape of dataset
df.shape
```

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

In [11]:
```
# Columns present in the dataset
df.columns
```

Out[11]:
```
Index(['order_id', 'order_date', 'ship_date', 'ship_mode', 'customer_name',
       'segment', 'state', 'country', 'market', 'region', 'product_id',
       'category', 'sub_category', 'product_name', 'sales', 'quantity',
       'discount', 'profit', 'shipping_cost', 'order_priority', 'year'],
      dtype='object')
```

In [12]:
```
# A Concise Summary of the Dataset
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
```

In [13]:
```
# Checking Missing Vlues
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
```

In [14]:
```
# Getting descriptive statistics summary
df.describe()
```

Out[14]:

	sales	quantity	discount	profit	shipping_cost	year
count	51290.000000	51290.000000	51290.000000	51290.000000	51290.000000	51290.000000
mean	246.490581	3.476545	0.142908	28.641740	26.375818	2012.777208
std	487.565361	2.278766	0.212280	174.424113	57.296810	1.098931
min	0.444000	1.000000	0.000000	-6599.978000	0.002000	2011.000000
25%	30.758625	2.000000	0.000000	0.000000	2.610000	2012.000000
50%	85.053000	3.000000	0.000000	9.240000	7.790000	2013.000000
75%	251.053200	5.000000	0.200000	36.810000	24.450000	2014.000000
max	22638.480000	14.000000	0.850000	8399.976000	933.570000	2014.000000

Exploratory Data Analysis

what is the overall sales trend?

In [15]:
```
df["order_date"].min()
```

Out[15]:
```
Timestamp('2011-01-01 00:00:00')
```

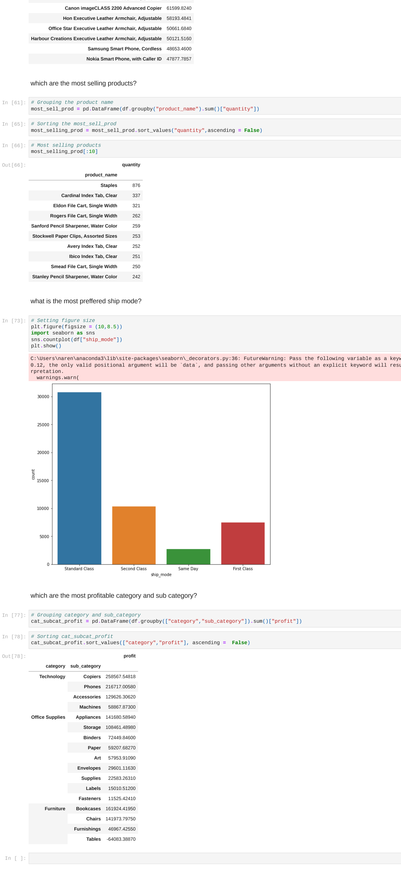
In [16]:
```
df["order_date"].max()
```

Out[16]:
```
Timestamp('2014-12-31 00:00:00')
```

In [25]:
```
# Getting month year from the dataset
df['month_year'] = df['order_date'].apply(lambda x: x.strftime('%Y-%m'))
```

In [27]:
```
# Grouping month year
df_trend = df.groupby('month_year').sum()[["sales"]].reset_index()
```

In [43]:
```
# Setting the figure size
plt.figure(figsize = (15,6))
plt.plot(df_trend['month_year'],df_trend['sales'], color = 'purple')
plt.xticks(rotation = 'vertical', size =8)
plt.show()
```



what are the top 10 products by sales?

In [46]:
```
# Grouping product name column
prod_sales = pd.DataFrame(df.groupby('product_name').sum()["sales"])
```

In [51]:
```
# Sorting prod_sales column
prod_sales = prod_sales.sort_values("sales",ascending = False)
```

In [52]:
```
# Top 10 products by sales
prod_sales[:10]
```

Out[52]:

	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
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	48053.4600
Nokia Smart Phone, with Caller ID	47877.7857

which are the most selling products?

In [61]:
```
# Grouping the product name
most_sell_prod = pd.DataFrame(df.groupby("product_name").sum()["quantity"])
```

In [65]:
```
# Sorting the most_sell_prod
most_selling_prod = most_sell_prod.sort_values("quantity",ascending = False)
```

In [66]:
```
# Most selling products
most_selling_prod[:10]
```

Out[66]:

	quantity
product_name	
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
Eldon Index Tab, Clear	251
Smead File Cart, Single Width	250
Stanley Pencil Sharpener, Water Color	242

what is the most preffered ship mode?

In [73]:
```
# Setting figure size
plt.figure(figsize = (10,8.5))
import seaborn as sns
sns.countplot(df["ship_mode"])
plt.show()
```

```
C:\Users\naren\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```


which are the most profitable category and sub category?

In [77]:
```
# Grouping category and sub_category
cat_subcat_profit = pd.DataFrame(df.groupby(["category","sub_category"]).sum()["profit"])
```

In [78]:
```
# Sorting cat_subcat_profit
cat_subcat_profit.sort_values(["category","profit"], ascending = False)
```

Out[78]:

		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	141973.79700
	Furnishings	46967.42550
	Tables	-64083.38870

In [ ]:
```

```