

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

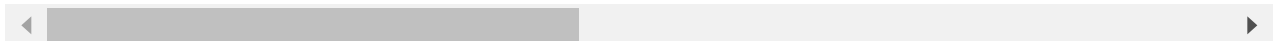
```
In [2]: ss = pd.read_excel(r'C:\Users\bukif\Desktop\Sales-Analysis-master\Sales-Analysis-master
```

```
In [3]: ss.head()
```

```
Out[3]:
```

	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	EL
4	IN-2011-47883	2011-01-01	2011-01-08	Standard Class	Joseph Holt	Consumer	New South Wales	Australia	APAC

5 rows × 21 columns



```
In [4]: ss.shape
```

```
Out[4]: (51290, 21)
```

```
In [5]: ss.columns
```

```
Out[5]: 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 [6]: ss.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 51290 entries, 0 to 51289
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  -
#   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 [7]:

```
ss.describe()
```

Out[7]:

	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

In [8]:

```
ss.isnull().sum()
```

Out[8]:

```

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
```

What is the overall sales trend?

```
In [9]: # Q1. What is the overall sales trend?
        # Q2. Which are the Top 10 products by sales?
        # Q3. Which are the Most Selling Products?
        # Q4. Which is the most preferred Ship Mode?
        # Q5. Which are the Most Profitable Category and Sub-Category?
```

```
In [10]: ss['order_date'].min()
```

```
Out[10]: Timestamp('2011-01-01 00:00:00')
```

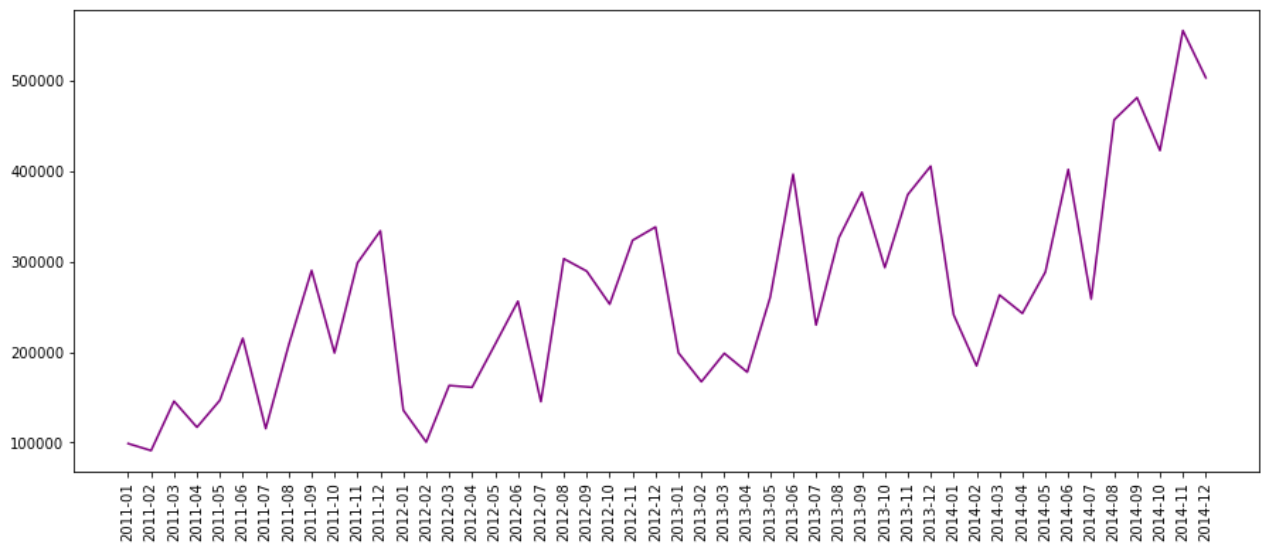
```
In [11]: ss['order_date'].max()
```

```
Out[11]: Timestamp('2014-12-31 00:00:00')
```

```
In [13]: ss['month_year'] = ss['order_date'].apply(lambda x: x.strftime('%Y-%m'))
```

```
In [14]: ss_trend = ss.groupby('month_year').sum()['sales'].reset_index()
```

```
In [15]: plt.figure(figsize=(15,6))
        plt.plot(ss_trend['month_year'], ss_trend['sales'], color='purple')
        plt.xticks(rotation='vertical',size=10)
        plt.show()
```



Which are the Top 10 products by sales?

```
In [16]: prod_sales = pd.DataFrame(ss.groupby('product_name').sum()['sales'])
```

```
In [17]: prod_sales = prod_sales.sort_values('sales', ascending=False)
```

```
In [18]: prod_sales[:10]
```

```
Out[18]:
```

	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	48653.4600
Nokia Smart Phone, with Caller ID	47877.7857

Which are the Most Selling Products?

```
In [19]: most_sell_prod = pd.DataFrame(ss.groupby('product_name').sum()['quantity'])
```

```
In [20]: most_sell_prod= most_sell_prod.sort_values('quantity',ascending=False)
```

```
In [21]: most_sell_prod[:10]
```

```
Out[21]:
```

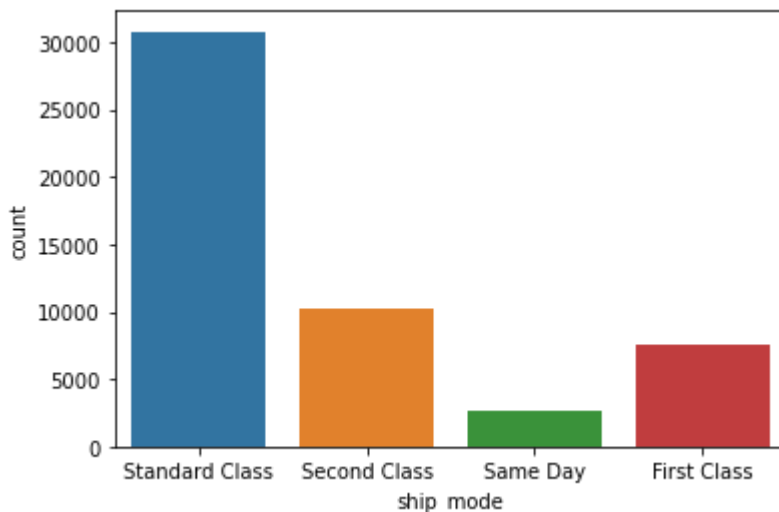
	quantity
Staples	876
Cardinal Index Tab, Clear	337
Eldon File Cart, Single Width	321
Rogers File Cart, Single Width	262
Sanford Pencil Sharpener, Water Color	259
Stockwell Paper Clips, Assorted Sizes	253
Avery Index Tab, Clear	252
Ibico Index Tab, Clear	251
Smead File Cart, Single Width	250
Stanley Pencil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

Which is the most preferred Ship Mode?

```
In [27]: import seaborn as sns
sns.countplot(ss['ship_mode'])
plt.show()
```

C:\ProgramData\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 [28]: cat_subcat_profit = pd.DataFrame(ss.groupby(['category', 'sub_category']).sum()['profit'])
```

```
In [29]: cat_subcat_profit.sort_values(['category', 'profit'], ascending=False)
```

Out[29]:

		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
Furniture	Fasteners	11525.42410
	Bookcases	161924.41950
	Chairs	141973.79750
	Furnishings	46967.42550
	Tables	-64083.38870

```
In [ ]:
```