

```
In [1]: import pandas as pd
from datetime import datetime, date
# DATA VISUALIZATION
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [2]: df=pd.read_csv("C:\\Users\\Dell\\Desktop\\DataScience\\superstore_sales.csv")
```

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

| | | | | | | | | | | | | | | | | | | |
|---------------------|-----------|-----------------|------------|------------|----------------|-----------------|-------------|-----------------|-----------|--------|---------|----------|-----------------|--------------|----------------------------|----------|----------|-----|
| | 0 | AG-2011-2040 | 01/01/2011 | 01/06/2011 | Standard Class | Toby Braunhardt | Consumer | Constantine | Algeria | Africa | Africa | ... | Office Supplies | Storage | Tenex Lockers, Blue | 408 | 2 | 0.0 |
| | 1 | IN-2011-47883 | 01/01/2011 | 01/08/2011 | Standard Class | Joseph Holt | Consumer | New South Wales | Australia | APAC | Oceania | ... | Office Supplies | Supplies | Acme Trimmer, High Speed | 120 | 3 | 0.1 |
| | 2 | HU-2011-1220 | 01/01/2011 | 01/05/2011 | Second Class | Annie Thurman | Consumer | Budapest | Hungary | EMEA | EMEA | ... | Office Supplies | Storage | Tenex Box, Single Width | 66 | 4 | 0.0 |
| | 3 | IT-2011-3647632 | 01/01/2011 | 01/05/2011 | Second Class | Eugene Moren | Home Office | Stockholm | Sweden | EU | North | ... | Office Supplies | Paper | Enemax Note Cards, Premium | 45 | 3 | 0.5 |
| | 4 | IN-2011-47883 | 01/01/2011 | 01/08/2011 | Standard Class | Joseph Holt | Consumer | New South Wales | Australia | APAC | Oceania | ... | Furniture | Furnishings | Eldon Light Bulb, Duo Pack | 114 | 5 | 0.1 |
| 5 rows × 21 columns | | | | | | | | | | | | | | | | | | |
| In [4]: | df.tail() | | | | | | | | | | | | | | | | | |
| Out[4]: | order_id | order_date | ship_date | ship_mode | customer_name | segment | state | country | market | region | ... | category | sub_category | product_name | sales | quantity | discount | |

5 rows x 21 columns

```
In [4]: df.tail()
```

| | | | | | | | | | | | | | | | | | | |
|---------------------|---------------------|----------------|------------|------------|----------------|------------------|----------|------------------|---------------|--------|---------|-----|-----------------|-----------|---|----|---|--|
| | 51286 | MO-2014-2560 | 12/31/2014 | 01/05/2015 | Standard Class | Liz Preis | Consumer | Souss-Massa-Draâ | Morocco | Africa | Africa | ... | Office Supplies | Binders | Wilson Jones Hole Reinforcements, Clear | 4 | 1 | |
| | 51287 | MX-2014-110527 | 12/31/2014 | 01/02/2015 | Second Class | Charlotte Melton | Consumer | Managua | Nicaragua | LATAM | Central | ... | Office Supplies | Labels | Hon Color Coded Labels, 5000 Label Set | 26 | 3 | |
| | 51288 | MX-2014-114783 | 12/31/2014 | 01/06/2015 | Standard Class | Tamara Dahlien | Consumer | Chihuahua | Mexico | LATAM | North | ... | Office Supplies | Labels | Hon Legal Exhibit Labels, Alphabetical | 7 | 1 | |
| | 51289 | CA-2014-156720 | 12/31/2014 | 01/04/2015 | Standard Class | Jill Matthias | Consumer | Colorado | United States | US | West | ... | Office Supplies | Fasteners | Bagged Rubber Bands | 3 | 3 | |
| 5 rows × 21 columns | | | | | | | | | | | | | | | | | | |
| In [5]: | <pre>df.shape</pre> | | | | | | | | | | | | | | | | | |
| Out[5]: | (51290, 21) | | | | | | | | | | | | | | | | | |

5 rows x 21 columns

```
In [5]: df.shape
```

```
Out[5]: (51290, 21)
```

```
In [6]: df.info
```

```
Out[6]: <bound method DataFrame.info of
0      AG-2011-2040      01/01/2011      01/06/2011      Standard Class
1      IN-2011-47883      01/01/2011      01/08/2011      Standard Class
2      HU-2011-1220      01/01/2011      01/05/2011      Second Class
3      IT-2011-3647632      01/01/2011      01/05/2011      Second Class
4      IN-2011-47883      01/01/2011      01/08/2011      Standard Class
...
51285  CA-2014-115427      12/31/2014      01/04/2015      Standard Class
51286  MO-2014-2560      12/31/2014      01/05/2015      Standard Class
51287  MX-2014-110527      12/31/2014      01/02/2015      Second Class
51288  MX-2014-114783      12/31/2014      01/06/2015      Standard Class
51289  CA-2014-156720      12/31/2014      01/04/2015      Standard Class

...
customer_name      segment      state      country      market \
0      Toby Braunhardt      Consumer      Constantine      Algeria      Africa
1      Joseph Holt      Consumer      New South Wales      Australia      APAC
2      Annie Thurman      Consumer      Budapest      Hungary      EMEA
3      Eugene Moren      Home Office      Stockholm      Sweden      EU
4      Joseph Holt      Consumer      New South Wales      Australia      APAC
...
51285      Erica Bern      Corporate      California      United States      US
51286      Liz Preis      Consumer      Souss-Massa-Draa      Morocco      Africa
51287      Charlotte Melton      Consumer      Managua      Nicaragua      LATAM
51288      Tamara Dahlien      Consumer      Chihuahua      Mexico      LATAM
51289      Jill Matthias      Consumer      Colorado      United States      US

...
region      ...      category      sub_category \
0      Africa      ...      Office Supplies      Storage
1      Oceania      ...      Office Supplies      Supplies
2      EMEA      ...      Office Supplies      Storage
3      North      ...      Office Supplies      Paper
4      Oceania      ...      Furniture      Furnishings
...
51285      West      ...      Office Supplies      Binders
51286      Africa      ...      Office Supplies      Binders
51287      Central      ...      Office Supplies      Labels
51288      North      ...      Office Supplies      Labels
51289      West      ...      Office Supplies      Fasteners

...
product_name      sales      quantity \
0      Tenex Lockers, Blue      408      2
1      Acme Trimmer, High Speed      120      3
2      Tenex Box, Single Width      66      4
3      Enemax Note Cards, Premium      45      3
4      Eldon Light Bulb, Duo Pack      114      5
...
51285      Cardinal Slant-D Ring Binder, Heavy Gauge Vinyl      14      2
51286      Wilson Jones Hole Reinforcements, Clear      4      1
51287      Hon Color Coded Labels, 5000 Label Set      26      3
51288      Hon Legal Exhibit Labels, Alphabetical      7      1
51289      Bagged Rubber Bands      3      3

...
discount      profit      shipping_cost      order_priority      year
0      0.0      166.1490      9.72      Medium      2011
1      0.1      36.0360      8.17      High      2011
2      0.0      29.6400      4.82      High      2011
3      0.5      -26.0550      4.79      Medium      2011
4      0.1      37.7700      4.79      Medium      2011
...
51285      0.2      4.5188      0.89      Medium      2014
51286      0.0      0.4200      0.49      Medium      2014
51287      0.0      12.5600      0.35      Medium      2014
51288      0.0      0.5600      0.20      Medium      2014
51289      0.2      -0.6048      0.17      Medium      2014

[51290 rows x 21 columns]>
```

```
In [8]: df.columns
```

```
Out[8]: Index(['order_id', 'order_date', 'ship_date', 'ship_mode', 'customer_name',
'segment', 'state', 'country', 'market', 'region', 'category', 'sub_category', 'product_name', 'sales', 'quantity',
'discount', 'profit', 'shipping_cost', 'order_priority', 'year'],
dtype='object')
```

```
In [9]: df.describe().round()
```

| Out[9]: | quantity | discount | profit | shipping_cost | year |
|---------|----------|----------|---------|---------------|---------|
| count | 51290.0 | 51290.0 | 51290.0 | 51290.0 | 51290.0 |
| mean | 3.0 | 0.0 | 29.0 | 26.0 | 2013.0 |
| std | 2.0 | 0.0 | 174.0 | 57.0 | 1.0 |
| min | 1.0 | 0.0 | -6600.0 | 0.0 | 2011.0 |
| 25% | 2.0 | 0.0 | 0.0 | 3.0 | 2012.0 |
| 50% | 3.0 | 0.0 | 9.0 | 8.0 | 2013.0 |
| 75% | 5.0 | 0.0 | 37.0 | 24.0 | 2014.0 |
| max | 14.0 | 1.0 | 8400.0 | 934.0 | 2014.0 |

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

```
Out[10]: '01/01/2011'
```

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

```
Out[11]: '9/30/2014'
```

```
In [12]: # Replace comma with empty space for sale attribute
df['sales']=df['sales'].str.replace(',','')

In [13]: # change object data type into integer
df['sales']=df['sales'].astype(int)

In [14]: 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  object
2   ship_date   51290 non-null  object
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  int32
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: float64(3), int32(1), int64(2), object(15)
memory usage: 8.0+ MB
```

```
In [15]: df['sales']
```

| | | |
|----------|--|-----|
| Out[15]: | 0 | 408 |
| | 1 | 120 |
| | 2 | 66 |
| | 3 | 45 |
| | 4 | 114 |
| | ... | |
| | 51285 | 14 |
| | 51286 | 4 |
| | 51287 | 26 |
| | 51288 | 7 |
| | 51289 | 3 |
| | Name: sales, Length: 51290, dtype: int32 | |

```
In [16]: # convert order_date time object into real data & time using panas
df['order_date']=pd.to_datetime(df['order_date'])
```

```
In [17]: #Getting months year from order_date
df['month_year']=df['order_date'].apply(lambda x:x.strftime('%Y-%m'))
```

```
In [18]: df['month_year']
```

| | | |
|----------|--|---------|
| Out[18]: | 0 | 2011-01 |
| | 1 | 2011-01 |
| | 2 | 2011-01 |
| | 3 | 2011-01 |
| | 4 | 2011-01 |
| | ... | |
| | 51285 | 2014-12 |
| | 51286 | 2014-12 |
| | 51287 | 2014-12 |
| | 51288 | 2014-12 |
| | 51289 | 2014-12 |
| | Name: month_year, Length: 51290, dtype: object | |

Q1. What is the overall sales trend?

```
In [36]: # grouping month_year by sale
df_trend=df.groupby('month_year').sum()[['sales']].reset_index()
```

```
In [21]: df_trend
```

| Out[21]: | month_year | sales |
|----------|------------|----------------|
| | 0 | 2011-01 98902 |
| | 1 | 2011-02 91152 |
| | 2 | 2011-03 145726 |
| | 3 | 2011-04 116927 |
| | 4 | 2011-05 146762 |
| | 5 | 2011-06 215214 |
| | 6 | 2011-07 115518 |
| | 7 | 2011-08 207570 |
| | 8 | 2011-09 290230 |
| | 9 | 2011-10 199070 |
| | 10 | 2011-11 298499 |
| | 11 | 2011-12 333941 |
| | 12 | 2012-01 135775 |
| | 13 | 2012-02 100521 |
| | 14 | 2012-03 163092 |
| | 15 | 2012-04 161060 |
| | 16 | 2012-05 208370 |
| | 17 | 2012-06 256181 |
| | 18 | 2012-07 145247 |
| | 19 | 2012-08 303158 |
| | 20 | 2012-09 289390 |
| | 21 | 2012-10 252942 |
| | 22 | 2012-11 323512 |
| | 23 | 2012-12 338245 |
| | 24 | 2013-01 199197 |
| | 25 | 2013-02 167247 |
| | 26 | 2013-03 198591 |
| | 27 | 2013-04 177835 |
| | 28 | 2013-05 260525 |
| | 29 | 2013-06 396513 |
| | 30 | 2013-07 229940 |
| | 31 | 2013-08 326491 |
| | 32 | 2013-09 376626 |
| | 33 | 2013-10 293423 |
| | 34 | 2013-11 373996 |
| | 35 | 2013-12 405476 |
| | 36 | 2014-01 241267 |
| | 37 | 2014-02 184848 |
| | 38 | 2014-03 263110 |
| | 39 | 2014-04 242781 |
| | 40 | 2014-05 288404 |
| | 41 | 2014-06 401843 |
| | 42 | 2014-07 258718 |
| | 43 | 2014-08 456633 |
| | 44 | 2014-09 481186 |
| | 45 | 2014-10 422785 |
| | 46 | 2014-11 555312 |
| | 47 | 2014-12 503154 |

```
In [22]: plt.figure(figsize=(15,6))
plt.plot(df_trend['month_year'],df_trend['sales'],color='#b80045')
plt.xticks(rotation='vertical',size=8)
plt.show()
```



Which are the Top 10 products by sales?

```
In [37]: prod_sales=pd.DataFrame(df.groupby('product_name').sum()[['sales']])
```

```
In [38]: most_selling_product=pd.DataFrame(df.groupby('product_name').sum())
```

```
In [39]: most_selling_product[:10]
```

| Out[39]: | | sales | quantity | discount | profit | shipping_cost | year |
|----------|--|--|----------|----------|--------|---------------|--------------|
| | | product_name | | | | | |
| | | "While you Were Out" Message Book, One Form per Page | 25 | 8 | 0.4 | 10.3880 | 3.10 6042 |
| | | #10 Gummed Flap White Envelopes, 100/Box | 42 | 11 | 0.4 | 16.7678 | 2.09 8052 |
| | | #10 Self-Seal White Envelopes | 109 | 10 | 0.2 | 52.1230 | 11.43 8053 |
| | | #10 White Business Envelopes,4 1/8 x 9 1/2 | 489 | 32 | 0.4 | 223.1408 | 53.03 14088 |
| | | #10- 4 1/8" x 9 1/2" Recycled-Tint Envelopes | 285 | 37 | 0.8 | 115.2806 | 34.95 20124 |
| | | #10- 4 1/8" x 9 1/2" Security-Tint Envelopes | 146 | 21 | 0.6 | 64.8636 | 13.14 16105 |
| | | #10-4 1/8" x 9 1/2" Premium Diagonal Seam Envelopes | 176 | 13 | 0.2 | 63.7470 | 11.13 4026 |
| | | #6 3/4 Gummed Flap White Envelopes | 72 | 9 | 0.8 | 24.9480 | 5.44 8052 |
| | | 1.7 Cubic Foot Compact "Cube" Office Refrigerators | 2706 | 14 | 0.4 | 578.6848 | 209.50 12076 |
| | | 114 Fold Party Design Invitations & White Envelopes, 24 8-1/2" X 11" Cards, 25 Env./Pack | 50 | 7 | 0.2 | 22.7115 | 4.25 4023 |

Q3. Which are the Most selling products?

```
In [40]: most_selling_product=most_selling_product.sort_values('quantity',ascending=False)
```

```
In [41]: most_selling_product[:5]
```

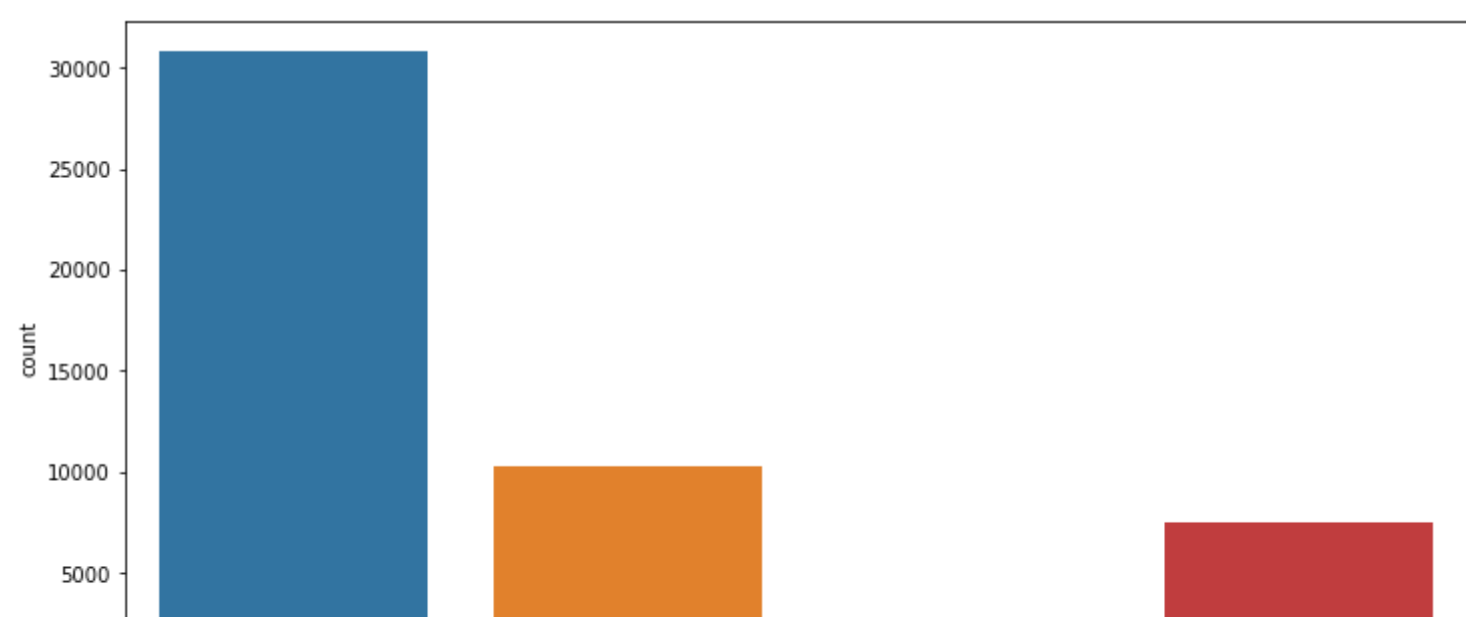
| Out[41]: | | sales | quantity | discount | profit | shipping_cost | year |
|----------|--|---------------------------------------|----------|----------|--------|---------------|----------------|
| | | product_name | | | | | |
| | | Staples | 7009 | 876 | 21.20 | 2611.0694 | 731.43 456868 |
| | | Cardinal Index Tab, Clear | 1922 | 337 | 10.32 | 556.3702 | 208.35 185186 |
| | | Eldon File Cart, Single Width | 34392 | 321 | 8.82 | 4238.8887 | 3970.85 181155 |
| | | Rogers File Cart, Single Width | 29467 | 262 | 16.26 | 2394.8953 | 3136.28 169085 |
| | | Sanford Pencil Sharpener, Water Color | 5581 | 259 | 10.66 | 601.9441 | 570.38 161008 |

Q4. Which is the most preferred Ship Mode?

```
In [42]: plt.figure(figsize=(12,6))
sns.countplot(df['ship_mode'])
plt.show()

C:\Users\Dell\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 misint
pretation.
warnings.warn(

Out[42]: <function matplotlib.pyplot.show(close=None, block=None)>
```



```
In [43]: preferred_ship_mode=pd.DataFrame(df.groupby('ship_mode').sum()[['sales']])
```

```
In [45]: preferred_ship_mode
```

| Out[45]: | | shipping_cost |
|----------|----------------|---------------|
| | ship_mode | |
| | First Class | 308103.25 |
| | Same Day | 115974.06 |
| | Second Class | 314112.62 |
| | Standard Class | 614630.76 |

```
In [44]: preferred_ship_mode=pd.DataFrame(df.groupby('ship_mode').sum()[['shipping_cost']])
```

```
In [46]: preferred_ship_mode
```

| Out[46]: | | shipping_cost |
|----------|----------------|---------------|
| | ship_mode | |
| | First Class | 308103.25 |
| | Same Day | 115974.06 |
| | Second Class | 314112.62 |
| | Standard Class | 614630.76 |

Q5. Which are the Most Profitable Category and Sub-Category?

```
In [35]: cat_sub_profit=pd.DataFrame(df.groupby(['category','sub_category']).sum()[['profit']])
```

```
In [47]: cat_sub_profit
```

| Out[47]: | | profit |
|----------|-----------------|--------------------------|
| | category | sub_category |
| | Furniture | Bookcases 161924.41950 |
| | | Chairs 141973.79750 |
| | | Furnishings 46967.42550 |
| | | Tables -64083.38870 |
| | Office Supplies | Appliances 141680.58940 |
| | | Art 57953.91090 |
| | | Binders 72449.84600 |
| | | Envelopes 29601.11630 |
| | | Fasteners 11525.42130 |
| | | Labels 15010.51200 |
| | | Paper 59207.68270 |
| | | Storage 108461.48980 |
| | | Supplies 22583.26310 |
| | Technology | Accessories 129626.30620 |
| | | Copiers 258567.84318 |
| | | Machines 58867.87300 |
| | | Phones 216717.00580 |

```
In [ ]:
```