EXPLORATORY DATA ANALYSIS: SAMPLE SUPERSTORE

DEPT DS/CSBS, The Sports Foundation

Name :- Anmol Malviya

Designation:- Data Science & Business Analytics Intern

Linkedin:

You can all Import extra Important libraries like Pandas,Matplotlib and Seaborn.

```
%matplotlib inline
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style('whitegrid')
sns.set_palette('husl')
```

After Importing the libraries we will import our sample store dataset which is in .csv format, we will name it variable as "superstore".

```
superstore = pd.read_csv('../input/sample-superstore.csv')
```

Reading the dataset

```
superstore.head()
```

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.96	2
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.94	3
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14.62	2
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.577	5
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.368	2

```
superstore.dtypes
```

```
Ship Mode       object
Segment         object
Country         object
City            object
State           object
Postal Code      int64
Region          object
Category        object
Sub-Category    object
Sales          float64
Quantity         int64
Discount       float64
Profit         float64
dtype: object
```

```
superstore.shape
```

```
(9994, 13)
```

```
superstore.describe()
```

	Postal Code	Sales	Quantity	Discount	Profit
count	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000
mean	55190.379428	229.858001	3.789574	0.156203	28.656896
std	32063.693350	623.245101	2.225110	0.206226	234.260108
min	1040.000000	0.444000	1.000000	0.000000	-6599.978000
25%	23223.000000	17.280000	1.000000	0.000000	1.728750
50%	56430.500000	54.490000	3.000000	0.200000	8.666500
75%	90008.000000	210.605000	5.000000	0.200000	29.364000
max	99301.000000	22638.480000	14.000000	0.800000	8399.976000

```
superstore.describe(include='O')
```

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category
count	9994	9994	9994	9994	9994	9994	9994	9994	9994
unique	4	3	1	531	49	631	4	3	17
top	Standard Class	Consumer	United States	New York City	California	10035	West	Office Supplies	Binders
freq	5968	5191	9994	915	1946	263	3203	6026	1523

```
superstore['Country'].unique()
```

```
array(['United States'], dtype=object)
```

```
superstore['Ship Mode'].unique()
```

```
array(['Second Class', 'Standard Class', 'First Class', 'Same Day'],
      dtype=object)
```

```
superstore['Segment'].unique()
```

```
array(['Consumer', 'Corporate', 'Home Office'], dtype=object)
```

```
superstore['Category'].unique()
```

```
array(['Furniture', 'Office Supplies', 'Technology'], dtype=object)
```

```
superstore['Sub-Category'].unique()
```

```
array(['Bookcases', 'Chairs', 'Labels', 'Tables', 'Storage', 'Furnishings',
       'Art', 'Phones', 'Binders', 'Accessories', 'Machines', 'Copiers',
       'Bookcases', 'Chairs', 'Storage', 'Furnishings', 'Paper', 'Binders',
       'Art', 'Phones', 'Accessories', 'Machines', 'Copiers', 'Paper',
       'Bookcases', 'Chairs', 'Labels', 'Tables', 'Storage', 'Furnishings',
       'Art', 'Phones', 'Binders', 'Accessories', 'Machines', 'Copiers',
       'Paper', 'Bookcases', 'Chairs', 'Labels', 'Tables', 'Storage',
       'Furnishings', 'Art', 'Phones', 'Binders', 'Accessories',
       'Machines', 'Copiers', 'Paper', 'Bookcases', 'Chairs', 'Labels',
       'Tables', 'Storage', 'Furnishings', 'Art', 'Phones', 'Binders',
       'Accessories', 'Machines', 'Copiers', 'Paper', 'Bookcases',
       'Chairs', 'Labels', 'Tables', 'Storage', 'Furnishings', 'Art',
       'Phones', 'Binders', 'Accessories', 'Machines', 'Copiers',
       'Paper'], dtype=object)
```

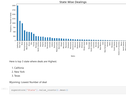
Univariate Analysis

```
# plotting single column in seaborn
```

```
superstore['Ship Mode'].value_counts
```

```
<bound method IndexOpsMixin.value_counts of 0    Second Class
1    Second Class
2    Second Class
3    Standard Class
4    Standard Class
5    Standard Class
6    Standard Class
7    Standard Class
8    Standard Class
9    Standard Class
10   Standard Class
11   Standard Class
12   Standard Class
13   Standard Class
14   Standard Class
15   Standard Class
```

```
State Wise Dealings
```



Here top 3 state where deals are Highest

1. California
2. New York
3. Texas

Wyoming lowest Number of deal

```
[In]: [sales['City'].value_counts().nlargest(3)]
```

```
[Out]: [City
California    1021
New York      1021
Texas         1021
Name: count, dtype: int64]
```

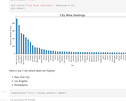
```
[In]: sales.shape
```

```
[Out]: (9994, 21)
```

Earlier number of deal per state is 256

6) Deal Analysis: City wise

```
[In]: plt.figure(figsize=(15,5))
sns.countplot(data=sales,x='City',order=sales['City'].value_counts().index[:3])
plt.xticks(rotation=90)
plt.title('City Wise Dealings')
plt.show()
```



Here top 3 city where deals are Highest

1. New York City
2. Los Angeles
3. Philadelphia

```
[In]: [sales['City'].value_counts().nsmallest(3)]
```

```
[Out]: [City
Murfreesboro    1
Chicopee        1
Westminster     1
Name: count, dtype: int64]
```

Earlier number of deal per city is 15

State wise analysis of Profit, Discount, Sale

```
[In]: [sales['State'].value_counts().nsmallest(3)]
```

```
[Out]: [State
North Dakota    56
West Virginia   60
Montana         61
Maine           62
Idaho           63
Wyoming         77
South Dakota    84
Vermont         85
New Hampshire   86
Name: count, dtype: int64]
```

```
[In]: [sales[sales.State.isin(['North Dakota', 'West Virginia', 'Montana', 'Maine', 'Idaho', 'Wyoming', 'South Dakota', 'Vermont', 'New Hampshire'])].shape]
```

```
[Out]: [(1000, 21)]
```

(1) Sale wise Profit analysis

```
[In]: df_statewisesales=sales.groupby('State').sum()[['Sales','Discount','Profit']]
```

```
[In]: plt.figure(figsize=(15,5))
sns.barplot(data=df_statewisesales.reset_index(),x='State',y='Sales')
plt.xticks(rotation=90)
plt.title('State wise Sales Analysis')
plt.show()
```

Wyoming lowest Sale

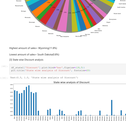
Sales analysis of Wyoming

```
[In]: df_statewisesales[df_statewisesales.Sales==df_statewisesales.Sales.min()]
```

```
[Out]:
       Sales  Discount  Profit
State
Wyoming  1603     16.1   -24.1
```

(2) Sale wise Profit analysis

```
[In]: plt.figure(figsize=(15,5))
sns.barplot(data=df_statewisesales.reset_index(),x='State',y='Profit')
plt.xticks(rotation=90)
plt.title('State wise Profit Analysis')
plt.show()
```



Highest amount of sale - Wyoming!! Why

Lowest amount of sale - South Dakota!!Why

(3) Sale wise Discount analysis

```
[In]: plt.figure(figsize=(15,5))
sns.barplot(data=df_statewisesales.reset_index(),x='State',y='Discount')
plt.xticks(rotation=90)
plt.title('State wise Discount Analysis')
plt.show()
```

Wyoming lowest Sale

Conclusion

- Profit is only large that of sale but there are some areas where profit could be increased
- Profit and Sale are proportional to each other
- Sale is more in Large Segment
- Sale category is Copier High Profit Sales
- Sub category: Binders, Machines and Sale tables have High Discounts
- Category: Machines sales and Profit relation is Negatively
- Category: Furniture profit relation is Furniture
- Region: Wyoming Sale
- State: Ohio, Lowest Profit
- Negative Sales Ohio High Profit & sales
- Here is top 3 City where deals are Highest
  1. New York City
  2. Los Angeles
  3. Philadelphia
- Sales and Profit are Moderately Correlated
- Sales and Discount are Moderately Correlated
- Discount and Profit are Negatively Correlated
- Here is top 3 state where deals are Highest
  1. California
  2. New York
  3. Texas
- Wyoming Lowest Number of deal,Highest amount of sales~Wyoming!!Why
- Lowest amount of sales: South Dakota!!Why
- Lastly I want to thank Pranav Dastey, V - The Spark Foundation and internships for giving me this wonderful opportunity.