# EDA\_SampleSuperStore

March 17, 2023

### 1 Task-03

The Exploratory Data Analysis of Super Market Data.

By:- Ukant Jadia GitHub Profile Link: https://github.com/ukantjadia/TSF-GRIP-Tasks

## 2 Setting Up Environment

#### 2.1 Loading some key Libraries

```
## Import key Projects

## for data handling
import pandas as pd
import numpy as np
from skimpy import skim
from scipy.stats import skew
from scipy.stats import kurtosis as krt

# for data visualization
import matplotlib.pyplot as plt
import seaborn as sns
from IPython.display import display

## For warnings
import warnings
warnings.filterwarnings("ignore", category=FutureWarning)
```

## 2.2 Collecting the Data

```
[3]: ## Reading the data and its Information

df = pd.read_csv('SampleSuperstore.csv')

df.head()
```

```
[3]:
            Ship Mode
                         Segment
                                        Country
                                                                        State
                                                             City
         Second Class
                        Consumer United States
                                                       Henderson
                                                                    Kentucky
         Second Class
    1
                        Consumer United States
                                                       Henderson
                                                                     Kentucky
    2
         Second Class Corporate United States
                                                      Los Angeles California
```

3 4	Standard Class			tates Fort La		Florida Florida	
4	Standard Class	Consume	er onroed bo	tates rolt La	uderdare	rioriua	L
	Postal Code Re	gion	Category	Sub-Category	Sales	Quantity	\
0	42420 S	outh	Furniture	Bookcases	261.9600	2	
1	42420 S	outh	Furniture	Chairs	731.9400	3	
2	90036	West Off:	ice Supplies	Labels	14.6200	2	
3	33311 S	outh	Furniture	Tables	957.5775	5	
4	33311 S	outh Off:	ice Supplies	Storage	22.3680	2	
	Discount Pr	ofit					
0	0.00 41.	9136					
1	0.00 219.	5820					
2	0.00 6.	8714					
3	0.45 -383.	0310					
4	0.20 2.	5164					

## 2.3 Analysising Data

## 2.3.1 Understanding the data

## [3]: df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 13 columns):

#	Column	Non-Null Count	Dtype
0	Ship Mode	9994 non-null	object
1	Segment	9994 non-null	object
2	Country	9994 non-null	object
3	City	9994 non-null	object
4	State	9994 non-null	object
5	Postal Code	9994 non-null	int64
6	Region	9994 non-null	object
7	Category	9994 non-null	object
8	Sub-Category	9994 non-null	object
9	Sales	9994 non-null	float64
10	Quantity	9994 non-null	int64
11	Discount	9994 non-null	float64
12	Profit	9994 non-null	float64
	(-)		(-)

dtypes: float64(3), int64(2), object(8)

memory usage: 1015.1+ KB

### Observation

- we could see that the features and its data type, along with Null count
- We have zero null-count in our dataset

#### • DataType table below

Data Type	Features				
int64(Continuous)	Postal Code, Quantity				
float64(Continuous)	Sales, Discount, Profit				
object(Categorical)	Ship Mode, Segment, Country, City, State,				
	Region, Category, Sub-Category				

#### 2.3.2 Handling the Data

```
[12]: ## Data Cleaning/ Wrangling
      ## 1. Unique values in features with object datatype
      obj_cols = df.select_dtypes(include=['object']).columns
      num_cols = df.select_dtypes(include=np.number).columns.tolist()
      unique_values = lambda :[ print("Unique count in ",obj_cols[i]," is_u
      ,len(df[obj_cols[i]].unique()), and values: ",df[obj_cols[i]].unique(),".

¬\n") for i in range(len(obj_cols))]
      unique_values()
      print("Unique values in City is :" , len(df['City'].unique()))
      print("Unique values in State is :" , len(df['State'].unique()))
     Unique count in Ship Mode is 4 and values: ['Second Class' 'Standard Class'
     'First Class' 'Same Day'] .
     Unique count in Segment is 3 and values: ['Consumer' 'Corporate' 'Home
     Office'] .
     Unique count in Country is 1 and values: ['United States'] .
     Unique count in City is 531 and values: ['Henderson' 'Los Angeles' 'Fort
     Lauderdale' 'Concord' 'Seattle'
      'Fort Worth' 'Madison' 'West Jordan' 'San Francisco' 'Fremont'
      'Philadelphia' 'Orem' 'Houston' 'Richardson' 'Naperville' 'Melbourne'
      'Eagan' 'Westland' 'Dover' 'New Albany' 'New York City' 'Troy' 'Chicago'
      'Gilbert' 'Springfield' 'Jackson' 'Memphis' 'Decatur' 'Durham' 'Columbia'
      'Rochester' 'Minneapolis' 'Portland' 'Saint Paul' 'Aurora' 'Charlotte'
      'Orland Park' 'Urbandale' 'Columbus' 'Bristol' 'Wilmington' 'Bloomington'
      'Phoenix' 'Roseville' 'Independence' 'Pasadena' 'Newark' 'Franklin'
      'Scottsdale' 'San Jose' 'Edmond' 'Carlsbad' 'San Antonio' 'Monroe'
      'Fairfield' 'Grand Prairie' 'Redlands' 'Hamilton' 'Westfield' 'Akron'
      'Denver' 'Dallas' 'Whittier' 'Saginaw' 'Medina' 'Dublin' 'Detroit'
      'Tampa' 'Santa Clara' 'Lakeville' 'San Diego' 'Brentwood' 'Chapel Hill'
      'Morristown' 'Cincinnati' 'Inglewood' 'Tamarac' 'Colorado Springs'
      'Belleville' 'Taylor' 'Lakewood' 'Arlington' 'Arvada' 'Hackensack'
```

'Saint Petersburg' 'Long Beach' 'Hesperia' 'Murfreesboro' 'Layton' 'Austin' 'Lowell' 'Manchester' 'Harlingen' 'Tucson' 'Quincy' 'Pembroke Pines' 'Des Moines' 'Peoria' 'Las Vegas' 'Warwick' 'Miami' 'Huntington Beach' 'Richmond' 'Louisville' 'Lawrence' 'Canton' 'New Rochelle' 'Gastonia' 'Jacksonville' 'Auburn' 'Norman' 'Park Ridge' 'Amarillo' 'Lindenhurst' 'Huntsville' 'Fayetteville' 'Costa Mesa' 'Parker' 'Atlanta' 'Gladstone' 'Great Falls' 'Lakeland' 'Montgomery' 'Mesa' 'Green Bay' 'Anaheim' 'Marysville' 'Salem' 'Laredo' 'Grove City' 'Dearborn' 'Warner Robins' 'Vallejo' 'Mission Viejo' 'Rochester Hills' 'Plainfield' 'Sierra Vista' 'Vancouver' 'Cleveland' 'Tyler' 'Burlington' 'Waynesboro' 'Chester' 'Cary' 'Palm Coast' 'Mount Vernon' 'Hialeah' 'Oceanside' 'Evanston' 'Trenton' 'Cottage Grove' 'Bossier City' 'Lancaster' 'Asheville' 'Lake Elsinore' 'Omaha' 'Edmonds' 'Santa Ana' 'Milwaukee' 'Florence' 'Lorain' 'Linden' 'Salinas' 'New Brunswick' 'Garland' 'Norwich' 'Alexandria' 'Toledo' 'Farmington' 'Riverside' 'Torrance' 'Round Rock' 'Boca Raton' 'Virginia Beach' 'Murrieta' 'Olympia' 'Washington' 'Jefferson City' 'Saint Peters' 'Rockford' 'Brownsville' 'Yonkers' 'Oakland' 'Clinton' 'Encinitas' 'Roswell' 'Jonesboro' 'Antioch' 'Homestead' 'La Porte' 'Lansing' 'Cuyahoga Falls' 'Reno' 'Harrisonburg' 'Escondido' 'Royal Oak' 'Rockville' 'Coral Springs' 'Buffalo' 'Boynton Beach' 'Gulfport' 'Fresno' 'Greenville' 'Macon' 'Cedar Rapids' 'Providence' 'Pueblo' 'Deltona' 'Murray' 'Middletown' 'Freeport' 'Pico Rivera' 'Provo' 'Pleasant Grove' 'Smyrna' 'Parma' 'Mobile' 'New Bedford' 'Irving' 'Vineland' 'Glendale' 'Niagara Falls' 'Thomasville' 'Westminster' 'Coppell' 'Pomona' 'North Las Vegas' 'Allentown' 'Tempe' 'Laguna Niguel' 'Bridgeton' 'Everett' 'Watertown' 'Appleton' 'Bellevue' 'Allen' 'El Paso' 'Grapevine' 'Carrollton' 'Kent' 'Lafayette' 'Tigard' 'Skokie' 'Plano' 'Suffolk' 'Indianapolis' 'Bayonne' 'Greensboro' 'Baltimore' 'Kenosha' 'Olathe' 'Tulsa' 'Redmond' 'Raleigh' 'Muskogee' 'Meriden' 'Bowling Green' 'South Bend' 'Spokane' 'Keller' 'Port Orange' 'Medford' 'Charlottesville' 'Missoula' 'Apopka' 'Reading' 'Broomfield' 'Paterson' 'Oklahoma City' 'Chesapeake' 'Lubbock' 'Johnson City' 'San Bernardino' 'Leominster' 'Bozeman' 'Perth Amboy' 'Ontario' 'Rancho Cucamonga' 'Moorhead' 'Mesquite' 'Stockton' 'Ormond Beach' 'Sunnyvale' 'York' 'College Station' 'Saint Louis' 'Manteca' 'San Angelo' 'Salt Lake City' 'Knoxville' 'Little Rock' 'Lincoln Park' 'Marion' 'Littleton' 'Bangor' 'Southaven' 'New Castle' 'Midland' 'Sioux Falls' 'Fort Collins' 'Clarksville' 'Sacramento' 'Thousand Oaks' 'Malden' 'Holyoke' 'Albuquerque' 'Sparks' 'Coachella' 'Elmhurst' 'Passaic' 'North Charleston' 'Newport News' 'Jamestown' 'Mishawaka' 'La Quinta' 'Tallahassee' 'Nashville' 'Bellingham' 'Woodstock' 'Haltom City' 'Wheeling' 'Summerville' 'Hot Springs' 'Englewood' 'Las Cruces' 'Hoover' 'Frisco' 'Vacaville' 'Waukesha' 'Bakersfield' 'Pompano Beach' 'Corpus Christi' 'Redondo Beach' 'Orlando' 'Orange' 'Lake Charles' 'Highland Park' 'Hempstead' 'Noblesville' 'Apple Valley' 'Mount Pleasant' 'Sterling Heights' 'Eau Claire' 'Pharr' 'Billings' 'Gresham' 'Chattanooga' 'Meridian' 'Bolingbrook' 'Maple Grove' 'Woodland' 'Missouri City' 'Pearland' 'San Mateo' 'Grand Rapids'

'Visalia' 'Overland Park' 'Temecula' 'Yucaipa' 'Revere' 'Conroe' 'Tinley Park' 'Dubuque' 'Dearborn Heights' 'Santa Fe' 'Hickory' 'Carol Stream' 'Saint Cloud' 'North Miami' 'Plantation' 'Port Saint Lucie' 'Rock Hill' 'Odessa' 'West Allis' 'Chula Vista' 'Manhattan' 'Altoona' 'Thornton' 'Champaign' 'Texarkana' 'Edinburg' 'Baytown' 'Greenwood' 'Woonsocket' 'Superior' 'Bedford' 'Covington' 'Broken Arrow' 'Miramar' 'Hollywood' 'Deer Park' 'Wichita' 'Mcallen' 'Iowa City' 'Boise' 'Cranston' 'Port Arthur' 'Citrus Heights' 'The Colony' 'Daytona Beach' 'Bullhead City' 'Portage' 'Fargo' 'Elkhart' 'San Gabriel' 'Margate' 'Sandy Springs' 'Mentor' 'Lawton' 'Hampton' 'Rome' 'La Crosse' 'Lewiston' 'Hattiesburg' 'Danville' 'Logan' 'Waterbury' 'Athens' 'Avondale' 'Marietta' 'Yuma' 'Wausau' 'Pasco' 'Oak Park' 'Pensacola' 'League City' 'Gaithersburg' 'Lehi' 'Tuscaloosa' 'Moreno Valley' 'Georgetown' 'Loveland' 'Chandler' 'Helena' 'Kirkwood' 'Waco' 'Frankfort' 'Bethlehem' 'Grand Island' 'Woodbury' 'Rogers' 'Clovis' 'Jupiter' 'Santa Barbara' 'Cedar Hill' 'Norfolk' 'Draper' 'Ann Arbor' 'La Mesa' 'Pocatello' 'Holland' 'Milford' 'Buffalo Grove' 'Lake Forest' 'Redding' 'Chico' 'Utica' 'Conway' 'Cheyenne' 'Owensboro' 'Caldwell' 'Kenner' 'Nashua' 'Bartlett' 'Redwood City' 'Lebanon' 'Santa Maria' 'Des Plaines' 'Longview' 'Hendersonville' 'Waterloo' 'Cambridge' 'Palatine' 'Beverly' 'Eugene' 'Oxnard' 'Renton' 'Glenview' 'Delray Beach' 'Commerce City' 'Texas City' 'Wilson' 'Rio Rancho' 'Goldsboro' 'Montebello' 'El Cajon' 'Beaumont' 'West Palm Beach' 'Abilene' 'Normal' 'Saint Charles' 'Camarillo' 'Hillsboro' 'Burbank' 'Modesto' 'Garden City' 'Atlantic City' 'Longmont' 'Davis' 'Morgan Hill' 'Clifton' 'Sheboygan' 'East Point' 'Rapid City' 'Andover' 'Kissimmee' 'Shelton' 'Danbury' 'Sanford' 'San Marcos' 'Greeley' 'Mansfield' 'Elyria' 'Twin Falls' 'Coral Gables' 'Romeoville' 'Marlborough' 'Laurel' 'Bryan' 'Pine Bluff' 'Aberdeen' 'Hagerstown' 'East Orange' 'Arlington Heights' 'Oswego' 'Coon Rapids' 'San Clemente' 'San Luis Obispo' 'Springdale' 'Lodi' 'Mason'] .

Unique count in State is 49 and values: ['Kentucky' 'California' 'Florida' 'North Carolina' 'Washington' 'Texas'

```
'Wisconsin' 'Utah' 'Nebraska' 'Pennsylvania' 'Illinois' 'Minnesota'
'Michigan' 'Delaware' 'Indiana' 'New York' 'Arizona' 'Virginia'
'Tennessee' 'Alabama' 'South Carolina' 'Oregon' 'Colorado' 'Iowa' 'Ohio'
'Missouri' 'Oklahoma' 'New Mexico' 'Louisiana' 'Connecticut' 'New Jersey'
'Massachusetts' 'Georgia' 'Nevada' 'Rhode Island' 'Mississippi'
'Arkansas' 'Montana' 'New Hampshire' 'Maryland' 'District of Columbia'
'Kansas' 'Vermont' 'Maine' 'South Dakota' 'Idaho' 'North Dakota'
'Wyoming' 'West Virginia'] .
```

Unique count in Region is 4 and values: ['South' 'West' 'Central' 'East'] .

Unique count in Category is 3 and values: ['Furniture' 'Office Supplies' 'Technology'] .

```
Unique count in Sub-Category is 17 and values: ['Bookcases' 'Chairs' 'Labels' 'Tables' 'Storage' 'Furnishings' 'Art' 'Phones' 'Binders' 'Appliances' 'Paper' 'Accessories' 'Envelopes' 'Fasteners' 'Supplies' 'Machines' 'Copiers'] .

Unique values in City is: 531
Unique values in State is: 49
```

#### Observation

- As Store Manager
  - We have orders from 1 country, 531 cites, with 631 postal code and 49 states with all 4 region, with 3 main category and 17 sub-category to 3 main segments placed with 4 different ship modes.

#### 2.3.3 Stastical Analysis of Data

```
[10]: ## Summary of Statics

## Using skimpy for quick view of summary
skim(df)

df.describe().round(2).T
```

df.describe().round(2).T												
<b>\( \)</b>	Dat	s a Summary	kimpy sumn	nary Data Types						Ш		
datafra ⇔	me		Values	Colu	mn Type	e Cour	nt					ш
Number	of :	rows	9994	obje	ct	8						Ш
	of	columns	12	floa	t64	3						ш
<b>→</b>		in	t64	1								
$\hookrightarrow$					n	umber						Ш
⊶hist		missing	complete		mean	sd	p0	p25	p75	p100	ш	
$\hookrightarrow$			rate									Ш
Sales		0		1	230	620	0.44	17	210	23000		Ш

Quantity	0	1	3.8	2.2	1	2	5	14	Ц
Discount	0	1	0.16	0.21	0	0	0.2	0.8	Ц
Profit	0	1	29	230	-6600	1.7	29	8400	Ц

End

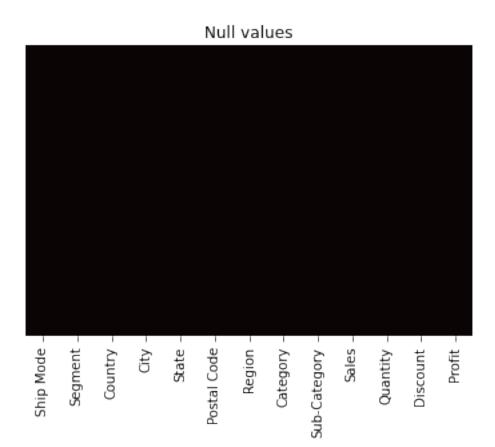
[10]:		count	mean	std	min	25%	50%	75%	max
	Sales	9994.0	229.86	623.25	0.44	17.28	54.49	209.94	22638.48
	Quantity	9994.0	3.79	2.23	1.00	2.00	3.00	5.00	14.00
	Discount	9994.0	0.16	0.21	0.00	0.00	0.20	0.20	0.80
	Profit	9994.0	28.66	234.26	-6599.98	1.73	8.67	29.36	8399.98

## Observation From Stastical Analysis

- From above summary
  - Sales and Profit contains more outiler than the outiler of Discount and Quantity.
  - There is no null values are present in given data.

## 2.3.4 Checking For Null Values

```
[4]: ## Checking Null vales in dataset
sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='mako')
plt.title('Null values')
plt.show()
print('Null Values in Dataset\n',df.isnull().sum()/(len(df))*100)
```



Null Values in	n Dataset
Ship Mode	0.0
Segment	0.0
Country	0.0
City	0.0
State	0.0
Postal Code	0.0
Region	0.0
Category	0.0
Sub-Category	0.0
Sales	0.0
Quantity	0.0
Discount	0.0
Profit	0.0
dtype: float64	4

### Observation

• From above heatmap we can see that there is zero null values in dataset.

#### 2.3.5 Dropping some Attribute

```
[7]: ## Dropping the Postal Code attribute

df = df.drop(['Postal Code'],axis=1)

df.sample(5)
```

```
[7]:
                Ship Mode
                               Segment
                                              Country
                                                                           State \
                                                                City
          Standard Class
                           Home Office
     6497
                                        United States
                                                          Louisville
                                                                        Kentucky
     2835 Standard Class
                              Consumer
                                        United States
                                                         Los Angeles
                                                                      California
     2680 Standard Class Home Office United States
                                                             Jackson
                                                                        Michigan
     2927 Standard Class
                              Consumer United States
                                                       Apple Valley
                                                                       Minnesota
     5316
              First Class Home Office United States
                                                           Knoxville
                                                                       Tennessee
                                                                      Discount \
            Region
                           Category Sub-Category
                                                     Sales
                                                            Quantity
             South
                                                                           0.0
     6497
                          Furniture Furnishings
                                                    20.940
                                                                   3
     2835
              West
                    Office Supplies
                                      Appliances
                                                    10.890
                                                                   1
                                                                           0.0
     2680
          Central
                         Technology
                                     Accessories
                                                  619.950
                                                                   5
                                                                           0.0
     2927
          Central
                    Office Supplies
                                             Art
                                                     8.640
                                                                   3
                                                                           0.0
     5316
             South
                          Furniture Furnishings
                                                    11.808
                                                                   2
                                                                           0.2
             Profit
             6.0726
     6497
     2835
             2.8314
     2680
         111.5910
             2.5056
     2927
     5316
             1.3284
```

### 2.3.6 Calculating The Skewness and Kurtosis in Data

```
[5]:
        Features
                               Kurtosis
                   Skewness
     0
        Quantity
                   1.278353
                               1.990293
     1
           Sales
                  12.970805
                             305.158427
       Discount
                   1.684042
                               2.407740
          Profit
                   7.560297 396.989229
```

#### Observation

- As basic stats we know
  - if skewness =  $0 \rightarrow Normal ditribution$
  - if skewness > 0 -> Left tail distribution
  - if skewness < 0 -> Right tail distribution

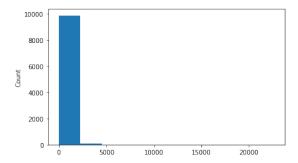
and

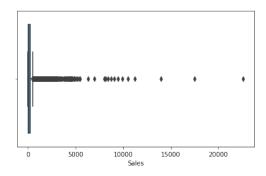
- if Kurtosis = 3 -> Normal distribution
- if kurtosis < 3 -> then it's called playkurtic(heavy-tailed)
- if kurtosis > 3 -> it's called leptokurtic and it tries to produce more outlier rather than normal distribution.
- So, in comparison to other continuour variable Sales have Huge Outlier and Heavly Tailed.
- Quatity is also contain the hugre outlier.

### 2.3.7 Univariate Analysis

```
[15]: ## Drawing the comparison plots between numberical data
for col in num_cols:
    print("Column is ",col)
    print('Skew : ',round(df[col].skew(),2))
    plt.figure(figsize=(15,4))
    plt.subplot(1,2,1)
    df[col].hist(grid=False)
    plt.ylabel('Count')
    plt.subplot(1,2,2)
    sns.boxplot(x=df[col])
    plt.show()
```

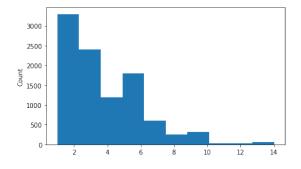
Column is Sales Skew: 12.97

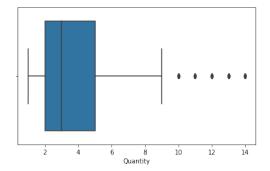




Column is Quantity

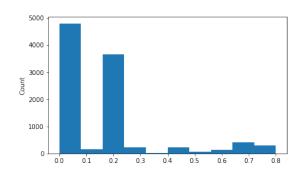
Skew : 1.28

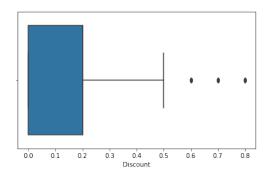




Column is Discount

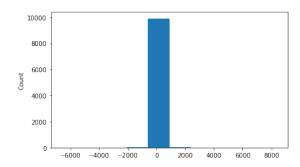
Skew : 1.68

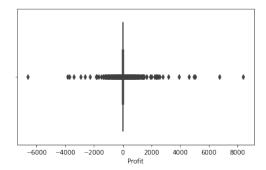




Column is Profit

Skew : 7.56





## Observation of Univariate Analysis

• Sales, Discount, Quantity are highly right skwed.

### 2.3.8 Skewness with Distribution plot

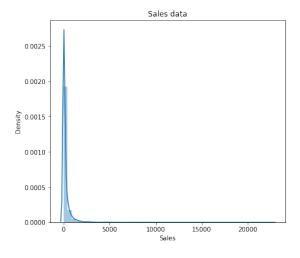
```
[6]: ## Univariate Analysis
## For understanding single variable
## Now we are doing the univariate analysis of our continous data

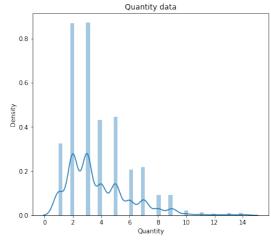
plt.figure(figsize=(10,10))
fig, ax = plt.subplots(1,2,figsize=(15,6))
fig, ax1 = plt.subplots(1,2,figsize=(15,6))

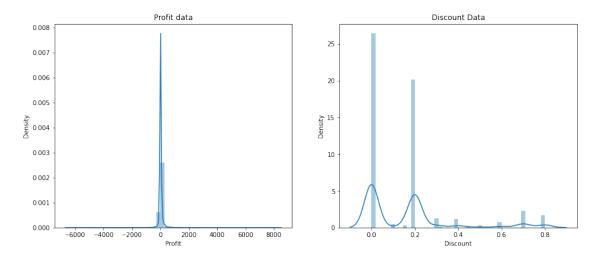
sns.distplot(df['Sales'],ax=ax[0])
ax[0].title.set_text("Sales data")
sns.distplot(df['Quantity'],ax=ax[1])
ax[1].title.set_text("Quantity data")
sns.distplot(df['Profit'],ax=ax1[0])
ax1[0].title.set_text("Profit data")
sns.distplot(df['Discount'],ax=ax1[1])
ax1[1].title.set_text("Discount Data")

plt.show()
```

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



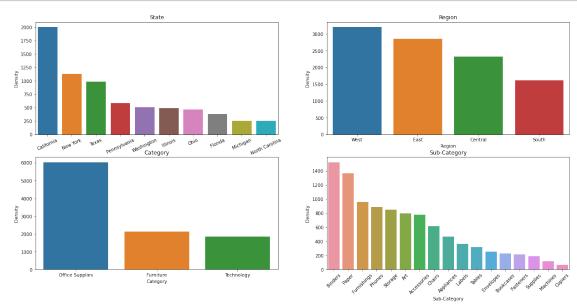




Observation From above visualization we can say that, - We are having high sales of value less than 5000. - The average of Quantity is 3-4. It means in most of the order we have 3 Quantity of product. -Profit is not very high for most of the product, and we can say Profit and Discount is related to each other.(the peak in both graph at low values only). - We are having 1.0-1.5 as our average of discount data.

#### 2.3.9 Univariate Analysis of Categorical Data

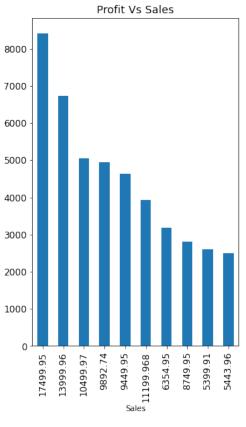
```
[7]: ## Univariate analysis
     ## Doing univariate analysis for categorical variable
     fig, ax = plt.subplots(2,2,figsize=(21,10))
     # Region
     sns.barplot(x = df['Region'].value counts().index,y = df['Region'].
      ⇔value_counts(),data=df,ax=ax[0,1])
     ax[0,1].set_xlabel('Region')
     ax[0,1].set_ylabel('Density')
     ax[0,1].set_title('Region')
     # Quantity
     sns.barplot(x = df['State'].value_counts().head(10).index,y = df['State'].
      \Rightarrowvalue_counts().head(10),ax=ax[0,0])
     ax[0,0].set_xlabel('State')
     ax[0,0].set_ylabel('Density')
     ax[0,0].tick_params(axis='x', labelrotation=25)
     ax[0,0].set_title('State')
     # Discount
     sns.barplot(x = df['Sub-Category'].value_counts().index,y = df['Sub-Category'].
      →value_counts(),ax=ax[1,1])
     ax[1,1].set_xlabel('Sub-Category')
     ax[1,1].set ylabel('Density')
```

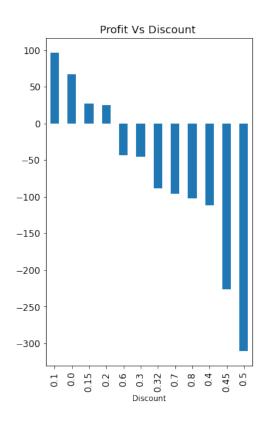


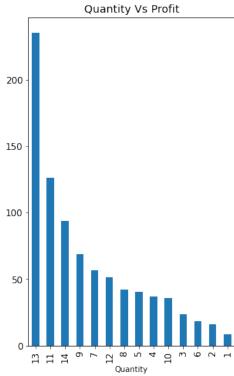
#### Observation of univariate analysis of categorical data

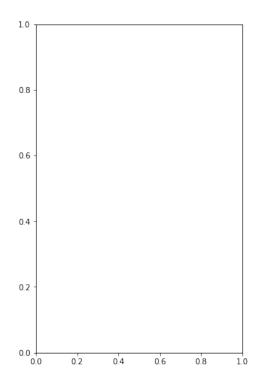
- The major orders are of office supplies
- region is west, city is california
- so, we need to focus more on the major regions and product

#### 2.3.10 Doing bivariate analysis





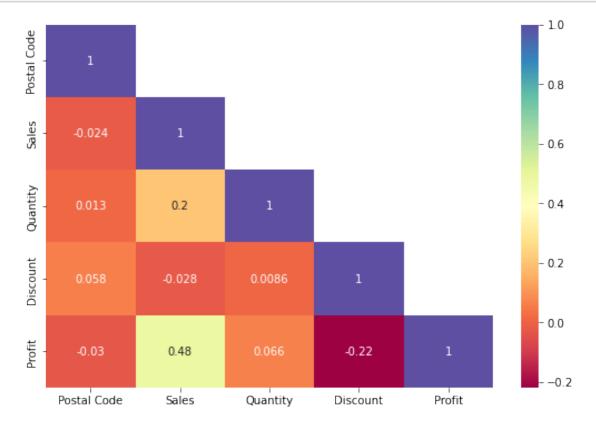




### 2.3.11 Multivariate Analysis

```
[]: # Corr in half traingle
    corr_df = df.corr()
    np.tril(np.ones(corr_df.shape)).astype(np.bool)
    corr_df_lower = corr_df.where(np.tril(np.ones(corr_df.shape)).astype(np.bool))
```

```
[]: ## Correlation Heatmap for relation ship among features
plt.figure(figsize=(9,6))
color=sns.color_palette('Spectral',as_cmap=True)
sns.heatmap(corr_df_lower,annot=True,cmap=color)
plt.show()
```



#### Observation

- Postal Code is just a geological code, So there is no relation of it with any other feature
- From above graph we have following observation:
  - Discount <-> Profit and Sales <-> Discount both having the negative relation to each other.
  - Quantity <-> Profit has the Positive relation, it means the more quantity in order, the more profit we have. and this is the similar case between the Sales <-> Profit and Quatity <-> Discount also.

 Sales <-> Quantity also has the Positive relation, means the more ordered quantity, the higher rate of selling.

## 2.4 Conclusion

- For more sell
  - add more discount on products quantity rage from 3-5.
  - add more discount on the office and furniture category.
  - Focus more on the West, East region and California state.
- To reduce the loss
  - keep the discount on product up to 2.00 only

## 3 Thank You

3.1 For notebook pdf ukantjadia.me/tsf/task-03.pdf