GRIP - The Sparks Foundation

Task details :

To perform Exploratory Data Analysis(EDA) Dataset used : Sample Superstore dataset Task no.: 3

Domain : Data Science and Business Analytics

Batch : October23

Done by : Ravuvari Nithish

Importing the required libraries

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

Loading the dataset

```
df = pd.read_csv("SampleSuperstore.csv")
```

Using head() and tail() functions from Pandas library

```
df.head()
```

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Profit
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	0.00	41.9136
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9400	3	0.00	219.5820
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14.6200	2	0.00	6.8714
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5775	5	0.45	-383.0310
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.3680	2	0.20	2.5164

```
df.tail()
```

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Prof
9989	Second Class	Consumer	United States	Miami	Florida	33180	South	Furniture	Furnishings	25.248	3	0.2	4.10
9990	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Furniture	Furnishings	91.960	2	0.0	15.63
9991	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Technology	Phones	258.576	2	0.2	19.39
9992	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Office Supplies	Paper	29.600	4	0.0	13.32
9993	Second Class	Consumer	United States	Westminster	California	92683	West	Office Supplies	Appliances	243.160	2	0.0	72.94

Printing a random row

```
df.sample(7)
```

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	P
1868	Standard Class	Home Office	United States	Philadelphia	Pennsylvania	19143	East	Office Supplies	Storage	78.256	2	0.2	-17
4231	Standard Class	Corporate	United States	Dallas	Texas	75220	Central	Office Supplies	Paper	15.552	3	0.2	5
1510	Standard Class	Corporate	United States	Pomona	California	91767	West	Office Supplies	Art	385.600	8	0.0	11
1367	First Class	Corporate	United States	Tucson	Arizona	85705	West	Furniture	Chairs	899.136	4	0.2	-146
9443	Standard Class	Home Office	United States	Jacksonville	Florida	32216	South	Technology	Phones	219.184	2	0.2	19
6521	Second Class	Consumer	United States	Jackson	Michigan	49201	Central	Furniture	Chairs	302.670	3	0.0	72
6307	Second Class	Corporate	United States	Eugene	Oregon	97405	West	Office Supplies	Paper	47.952	3	0.2	16

Checking the missing values

```
df.isnull().sum()
```

```
Ship Mode       0
Segment         0
Country         0
City            0
State           0
Postal Code     0
Region          0
```

```
Category        0
Sub-Category    0
Sales           0
Quantity        0
Discount        0
Profit          0
dtype: int64
```

Finding Total number of null values in a dataset

```
null_values = df.isnull().sum().sum()
print("total number of null values = ", null_values)
```

```
total number of null values =  0
```

Summary of the dataset

```
print(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
None
```

Statistical details of the dataset

```
df.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	0.206452	234.260108

Dimension, Columns and dtype of the dataset

```
df.shape
```

```
(9994, 13)
```

```
df.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
```

```
df.columns
```

```
Index(['Ship Mode', 'Segment', 'Country', 'City', 'State', 'Postal Code',
       'Region', 'Category', 'Sub-Category', 'Sales', 'Quantity', 'Discount',
       'Profit'],
      dtype='object')
```

Checking the dataset for duplicates and dropping the duplicate elements

```
df.duplicated().sum()
```

```
17
```

```
df.drop_duplicates()
```

	Ship Mode	Segment	Country	City	State	Postal Code	Region	Category	Sub-Category	Sales	Quantity	Discount	Pr
0	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Bookcases	261.9600	2	0.00	41
1	Second Class	Consumer	United States	Henderson	Kentucky	42420	South	Furniture	Chairs	731.9400	3	0.00	219
2	Second Class	Corporate	United States	Los Angeles	California	90036	West	Office Supplies	Labels	14.6200	2	0.00	6
3	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Furniture	Tables	957.5775	5	0.45	-383
4	Standard Class	Consumer	United States	Fort Lauderdale	Florida	33311	South	Office Supplies	Storage	22.3680	2	0.20	2
...	...	...	...	...	...	...	...	...	...	...	...	...	
9989	Second Class	Consumer	United States	Miami	Florida	33180	South	Furniture	Furnishings	25.2480	3	0.20	4
9990	Standard Class	Consumer	United States	Costa Mesa	California	92627	West	Furniture	Furnishings	91.9600	2	0.00	15
	Standard		United										
	Standard	Consumer	United	Costa Mesa	California	92627	West	Office	Paper	20.6000	4	0.00	13

Distinct values in the datset

```
df.nunique()
```

```
Ship Mode          4
Segment            3
Country            1
City             531
State             49
Postal Code      631
Region             4
Category           3
Sub-Category      17
Sales           5825
Quantity          14
Discount          12
Profit          7287
dtype: int64
```

Finding the correlation of dataset using corr() method

```
df.corr()
```

```
<ipython-input-16-2f6f6606aa2c>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a futur
  df.corr()
```

	Postal Code	Sales	Quantity	Discount	Profit
Postal Code	1.000000	-0.023854	0.012761	0.058443	-0.029961

Finding the covariance of dataset using cov() method

Quantity	0.012761	0.200795	1.000000	0.008623	0.066253

```
df.cov()
```

```
<ipython-input-17-6f98a29763d5>:1: FutureWarning: The default value of numeric_only in DataFrame.cov is deprecated. In a future version, it will default to False. Select only valid
  df.cov()
```

	Postal Code	Sales	Quantity	Discount	Profit
Postal Code	1.028080e+09	-476682.766590	910.415885	386.870404	-225045.849445
Sales	-4.766828e+05	388434.455308	278.459923	-3.627228	69944.096586
Quantity	9.104159e+02	278.459923	4.951113	0.003961	34.534769
Discount	3.868704e+02	-3.627228	0.003961	0.042622	-10.615173
Profit	-2.250458e+05	69944.096586	34.534769	-10.615173	54877.798055

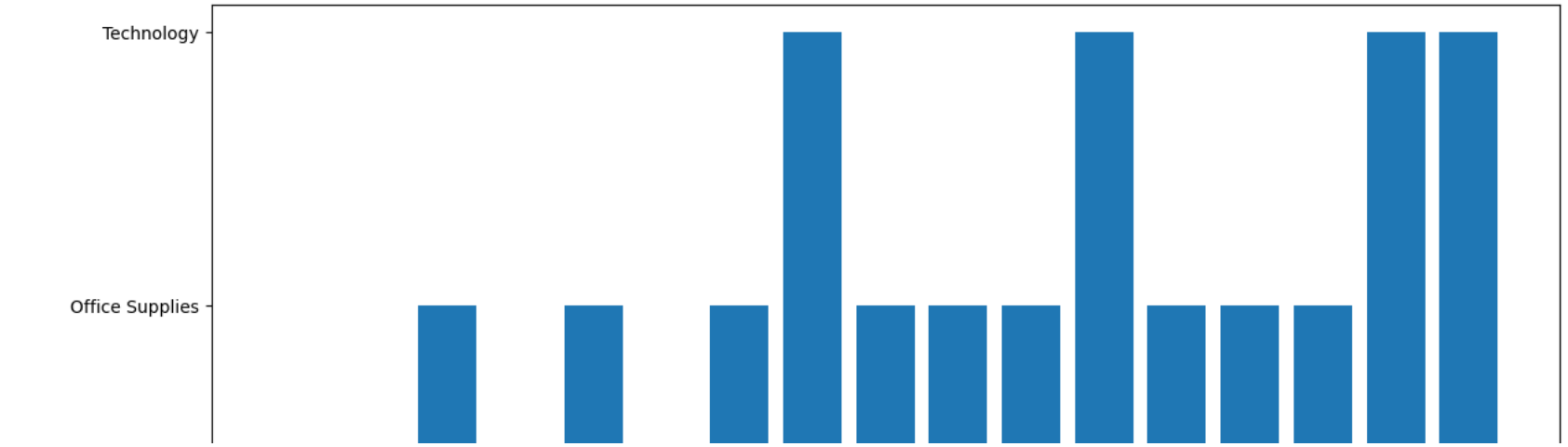
Finding the Series containing counts of unique values

```
df.value_counts()
```

```
Ship Mode       Segment      Country        City           State       Postal Code  Region   Category         Sub-Category  Sales    Quantity  Discount  Profit
Standard Class  Consumer     United States  Salem          Oregon      97301        West     Office Supplies  Paper         10.368   2         0.2        3.6288      2
Second Class    Corporate    United States  Chicago        Illinois    60653        Central  Office Supplies  Binders       3.564    3         0.8       -6.2370      2
Standard Class  Consumer     United States  San Francisco  California  94122        West     Office Supplies  Paper         12.840   3         0.0        5.7780      2
                                            Los Angeles    California  90036        West     Office Supplies  Paper         19.440   3         0.0        9.3312      2
Same Day        Home Office  United States  San Francisco  California  94122        West     Office Supplies  Labels        41.400   4         0.0        19.8720     2
                                                                                                                                                                    ..
Second Class    Corporate    United States  Las Vegas      Nevada      89115        West     Office Supplies  Paper         97.880   2         0.0        48.9400     1
                                            Little Rock    Arkansas    72209        South    Office Supplies  Envelopes     182.940  3         0.0        85.9818     1
                                                                                                             Paper         44.960   2         0.0        20.6816     1
                                                                                                             Storage       62.040   4         0.0        17.3712     1
Standard Class  Home Office  United States  Yuma           Arizona     85364        West     Technology       Machines      599.985  5         0.7       -479.9880    1
Length: 9977, dtype: int64
```

Visualization of the dataset

```
plt.figure(figsize=(14,6))
plt.bar('Sub-Category','Category', data=df)
plt.show()
```




```
print(df['State'].value_counts())
plt.figure(figsize=(15,8))
sns.countplot(x=df['State'])
plt.xticks(rotation=90)
plt.show()
```

```
Oklahoma                66
Missouri                66
Alabama                 61
Arkansas                60
Rhode Island            56
Utah                    53
Mississippi             53
Louisiana               42
South Carolina          42
Nevada                  39
Nebraska                38
New Mexico              37
Iowa                    30
New Hampshire           27
Kansas                  24
Idaho                   21
Montana                 15
South Dakota            12
Vermont                 11
District of Columbia    10
Maine                    8
North Dakota             7
West Virginia            4
Wyoming                  1
Name: State, dtype: int64
```

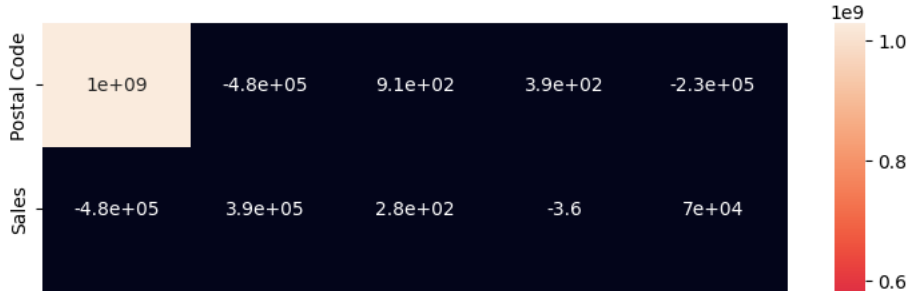
Heatmap of the dataset

```
fig,axes = plt.subplots(1,1,figsize=(10,6))
sns.heatmap(df.corr(), annot= True)
plt.show()
```

```
<ipython-input-21-0c1519aa326a>:2: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version, it will default to False. Select only valid
  sns.heatmap(df.corr(), annot= True)
```

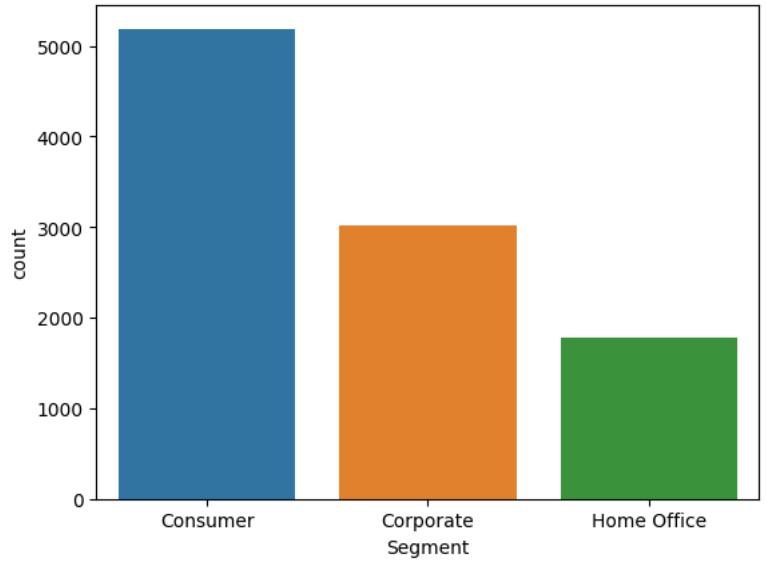

```
fig,axes = plt.subplots(1,1,figsize=(9,6))
sns.heatmap(df.cov(), annot= True)
plt.show()
```

```
<ipython-input-22-d3c100f5e8f3>:2: FutureWarning: The default value of numeric_only in DataFrame.cov is deprecated. In a future version, it will default to False. Select only valid
  sns.heatmap(df.cov(), annot= True)
```



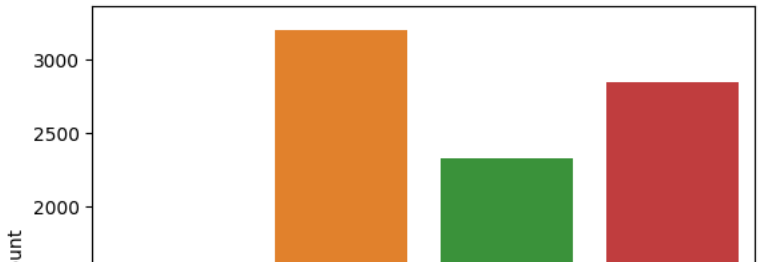

```
sns.countplot(x=df['Segment'])
```

```
<Axes: xlabel='Segment', ylabel='count'>
```



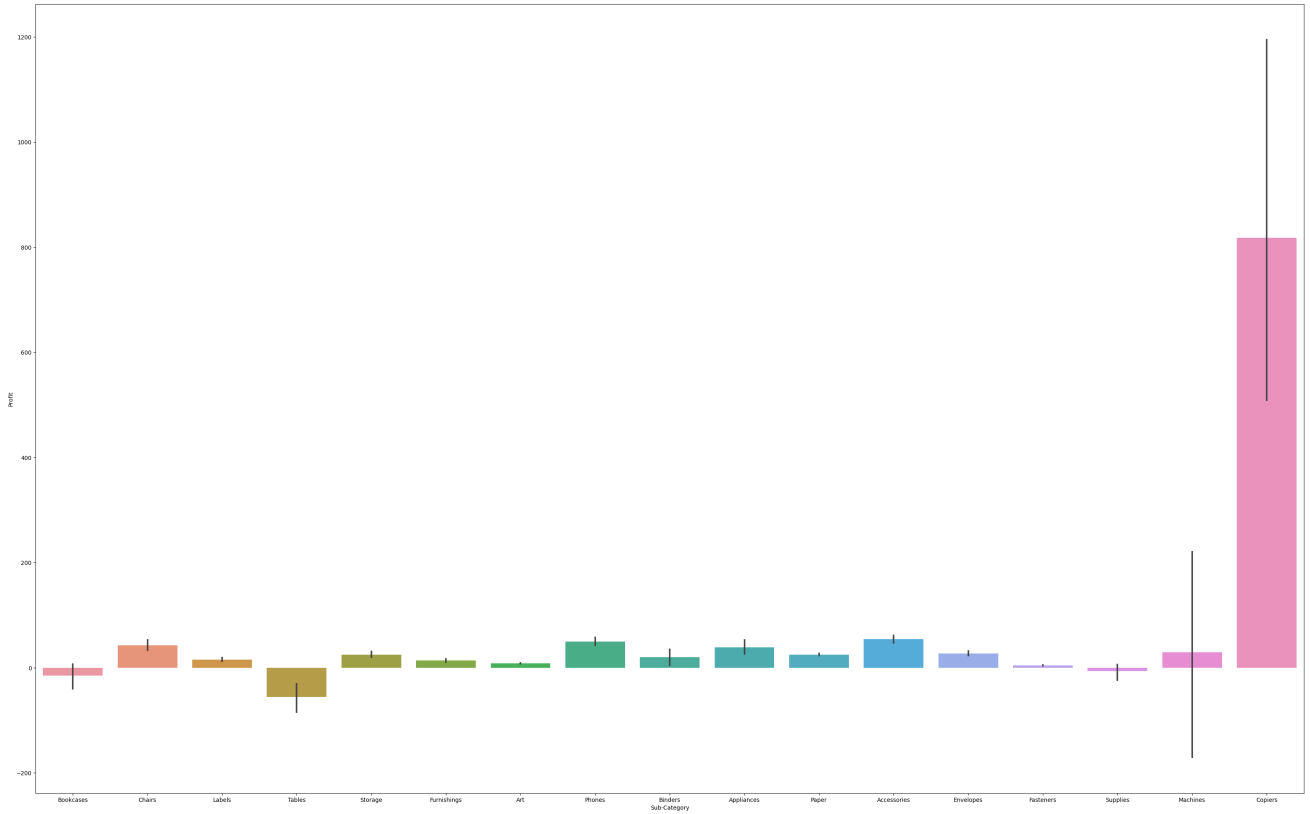

```
sns.countplot(x=df['Region'])
```

```
<Axes: xlabel='Region', ylabel='count'>
```




```
plt.figure(figsize=(40,25))
sns.barplot(x=df['Sub-Category'], y=df['Profit'])
```

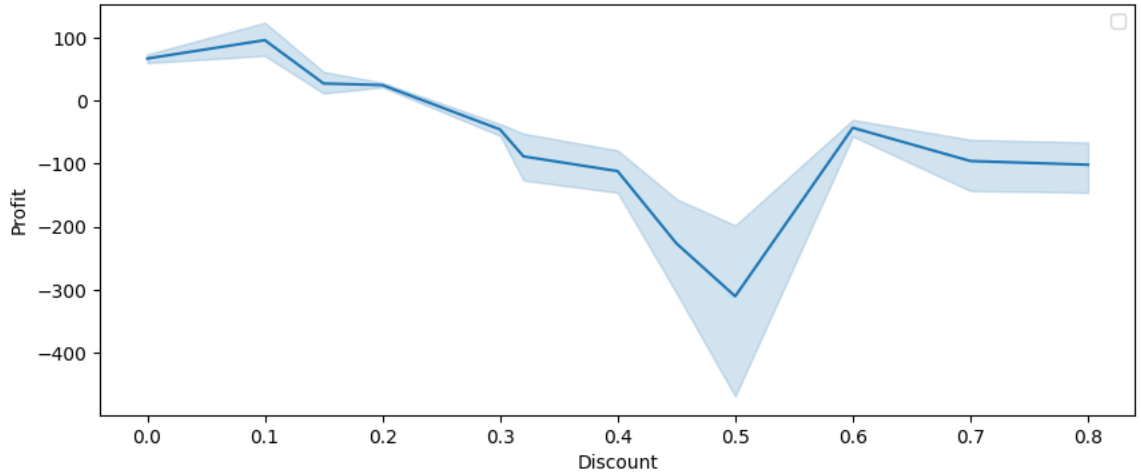
```
<Axes: xlabel='Sub-Category', ylabel='Profit'>
```



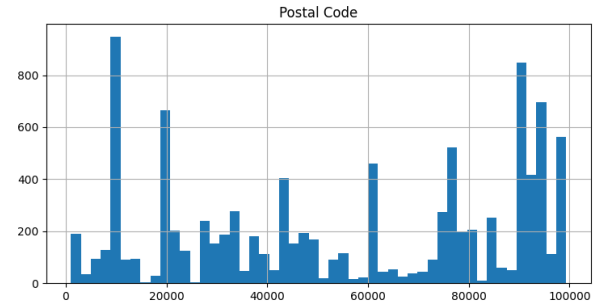

```
plt.figure(figsize = (10,4))
sns.lineplot(x="Discount", y="Profit", data = df)
```

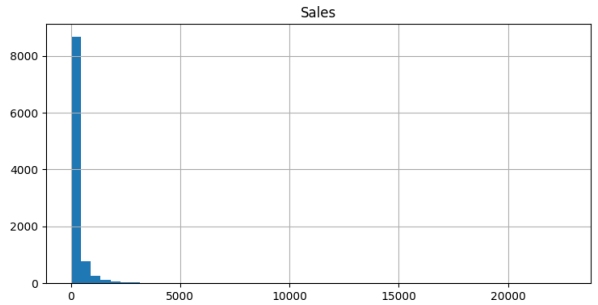
```
plt.legend()
```

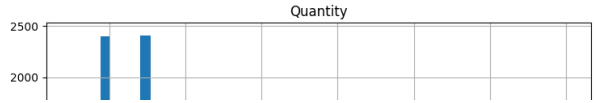
```
WARNING:matplotlib.legend:No artists with labels found to put in legend.  Note that artists whose label start with an underscor
<matplotlib.legend.Legend at 0x7fd5e857bee0>
```

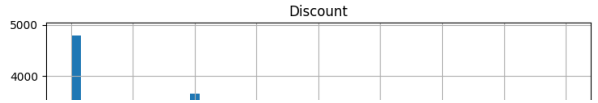



```
df.hist(bins=50 ,figsize=(20,15))
plt.show()
```



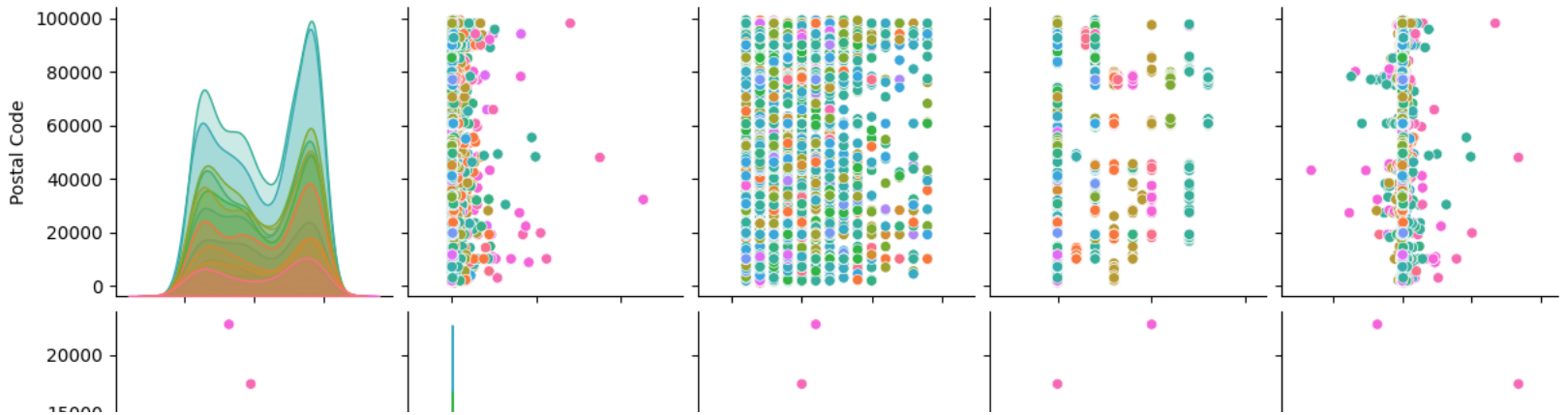







```
figsize=(15,10)
sns.pairplot(df,hue='Sub-Category')
```

```
<seaborn.axisgrid.PairGrid at 0x7fd5e8531de0>
```



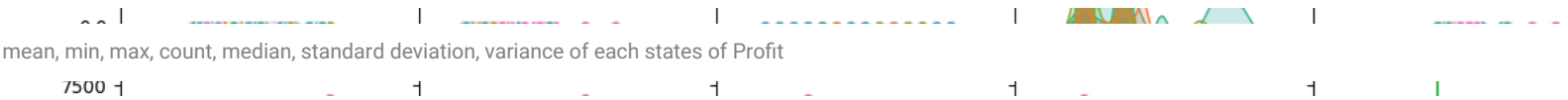
Sum the sales, profit, discount, quantity according to every state of region and also according to sub-categories sales

```
grouped=pd.DataFrame(df.groupby(['Ship Mode','Segment','Category','Sub-Category','State','Region'])['Quantity','Discount','Sales','Profit'].sum().reset_index())
grouped
```

```
<ipython-input-29-457e25d98647>:1: FutureWarning: Indexing with multiple keys (implicitly converted to a tuple of keys) will be deprecated, use a list instead.
  grouped=pd.DataFrame(df.groupby(['Ship Mode','Segment','Category','Sub-Category','State','Region'])['Quantity','Discount','Sales','Profit'].sum().reset_index())
```

	Ship Mode	Segment	Category	Sub-Category	State	Region	Quantity	Discount	Sales	Profit
0	First Class	Consumer	Furniture	Bookcases	Arizona	West	5	0.70	181.470	-320.5970
1	First Class	Consumer	Furniture	Bookcases	California	West	9	0.45	1809.497	243.2526
2	First Class	Consumer	Furniture	Bookcases	Colorado	West	3	0.70	89.991	-152.9847
3	First Class	Consumer	Furniture	Bookcases	Florida	South	3	0.20	314.352	-15.7176
4	First Class	Consumer	Furniture	Bookcases	Georgia	South	5	0.00	354.900	88.7250
...	...	...	...	...	...	...	...	...	...	...
2978	Standard Class	Home Office	Technology	Phones	Texas	Central	12	0.60	808.704	77.9712
2979	Standard Class	Home Office	Technology	Phones	Vermont	East	5	0.00	1294.750	336.6350
2980	Standard Class	Home Office	Technology	Phones	Virginia	South	17	0.00	365.130	58.7384
2981	Standard Class	Home Office	Technology	Phones	Washington	West	17	1.20	1989.448	63.2645
2982	Standard Class	Home Office	Technology	Phones	Wisconsin	Central	1	0.00	125.990	35.2772

2983 rows × 10 columns



Sum, mean, min, max, count, median, standard deviation, variance of each states of Profit

```
df.groupby("State").Profit.agg(["sum","mean","min","max","count","median","std","var"])
```

Georgia	16250.0433	88.315453	0.1134	3177.4750	184	22.24690	283.026694	80104.109456
Idaho	826.7231	39.367767	1.1151	259.5297	21	14.70000	63.027976	3972.525785
Illinois	-12607.8870	-25.625787	-2929.4845	874.9875	492	-1.81440	175.695233	30868.814827
Indiana	18382.9363	123.375411	0.0000	8399.9760	149	18.76700	693.643105	481140.757367
Iowa	1183.8119	39.460397	2.5920	394.2680	30	13.93560	73.763444	5441.045702
Kansas	836.4435	34.851813	1.7280	149.3820	24	11.96880	42.619311	1816.405680

```
x = df.iloc[:, [9, 10, 11, 12]].values
from sklearn.cluster import KMeans
wcss = []

for i in range(1, 11):
    kmeans = KMeans(n_clusters = i, init = 'k-means++',
                    max_iter = 300, n_init = 10, random_state = 0).fit(x)
    wcss.append(kmeans.inertia_)

sns.set_style("whitegrid")
sns.FacetGrid(df, hue ="Sub-Category",height = 6).map(plt.scatter,'Sales','Quantity')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:,1],
            s = 100, c = 'yellow', label = 'Centroids')

plt.legend()
```

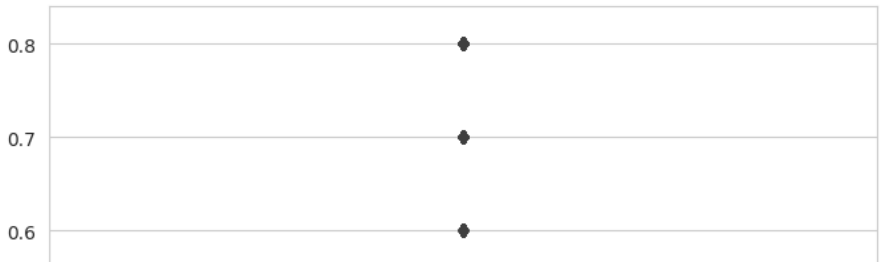
```
<matplotlib.legend.Legend at 0x7fd5e879dfc0>
```


```
sns.pairplot(df)
```

```
<seaborn.axisgrid.PairGrid at 0x7fd5e8ca2980>
```


```
fig, axes = plt.subplots(figsize = (8 , 8))

sns.boxplot(df['Discount'])
```

```
<Axes: >
```




```
fig, axes = plt.subplots(figsize = (10 , 10))

sns.boxplot(df['Profit'])
```

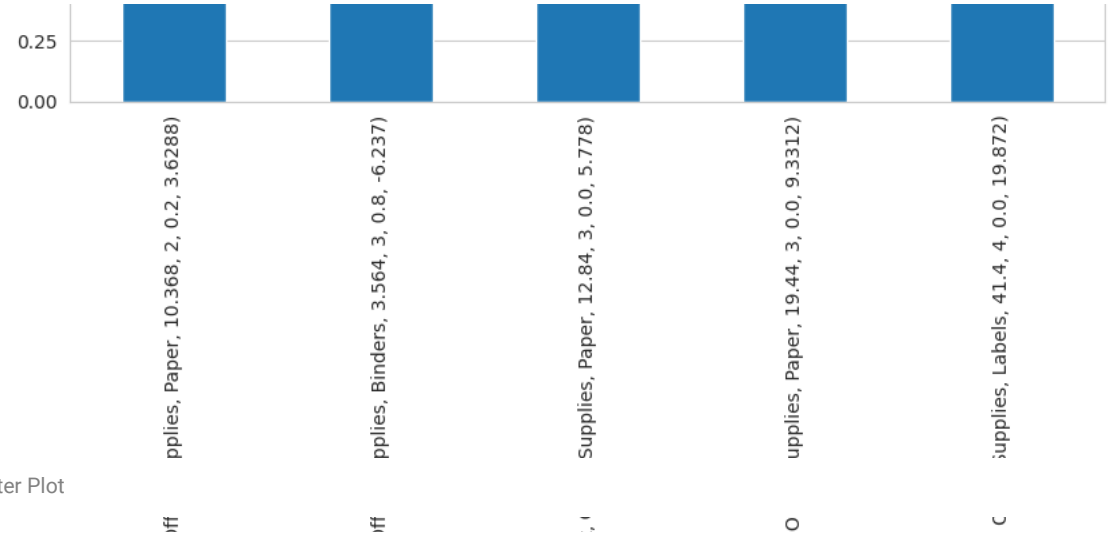
```
<Axes: >
```



```
Q1 = df.quantile(q = 0.25, axis = 0, numeric_only = True, interpolation = 'linear')

Q3 = df.quantile(q = 0.75, axis = 0, numeric_only = True, interpolation = 'linear')

IQR = Q3 - Q1

print(IQR)
```

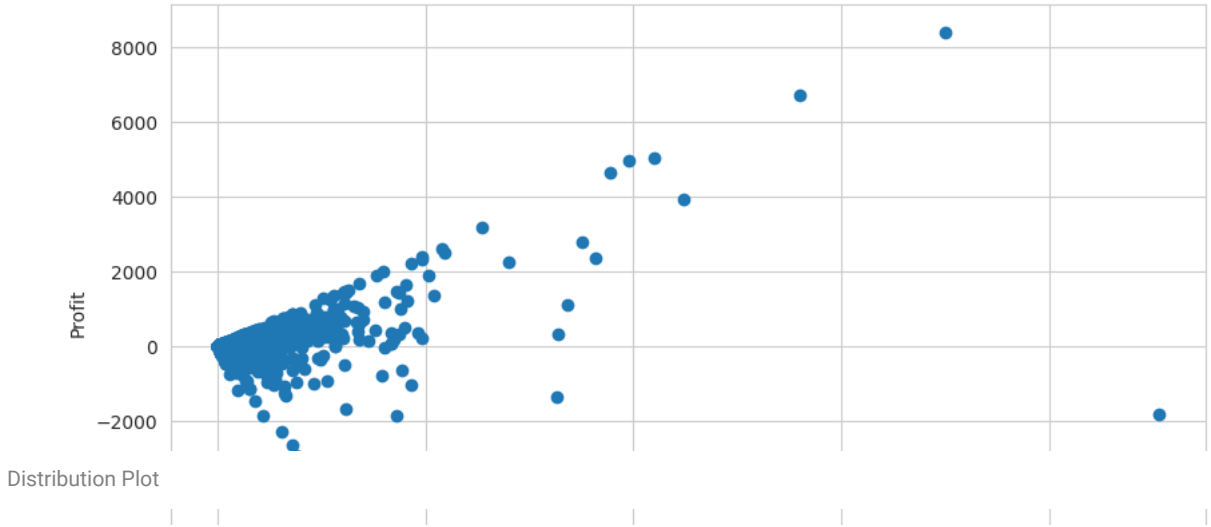
```
Postal Code    66785.00000
Sales            192.66000
Quantity           3.00000
Discount           0.20000
Profit            27.63525
dtype: float64
```

```
df.value_counts().nlargest().plot(kind = 'bar' , figsize = (10 , 5))
```



Scatter Plot

```
fig, ax = plt.subplots(figsize = (10 , 6))
ax.scatter(df["Sales"] , df["Profit"])
ax.set_xlabel('Sales')
ax.set_ylabel('Profit')
plt.show()
```




```
print(df['Sales'].describe())
plt.figure(figsize = (9 , 8))
sns.distplot(df['Sales'], color = 'b', bins = 100, hist_kws = {'alpha': 0.4});
```