**MODULE-2**

import numpy as np,pandas as pd,matplotlib.pyplot as plt

from datetime import datetime

import scipy.stats

import seaborn as sns

**## importing a csv file into**

x=pd.read\_csv('Sales.csv')

**## creating pandas dataframe**

df=pd.DataFrame({'S.no':[1,2,3,4,5],'Date':['24-11-2005','6-9-2006','3-12-2007','6-8-2008','10-10-2009'],'India\_income':[12.85,20.00,22.50,26.70,28.20], 'China\_income':[15.5,17.2,19.0,20.8,23.0],'Tokyo\_income':[10.0,11.8,13.5,10.2,9.0],'Kenya\_income':[8.0,9.7,11.8,13.0,13.9]})

print(df)

**## converting date column to datetime**

updated\_time= datetime.datetime.strftime()

df['Date']=pd.to\_datetime(df['Date'])

df.info()

**## creating stats from the above dataframe**

c=df[['India\_income','China\_income','Tokyo\_income','Kenya\_income']].max() ##max of all income columns

print('Maximum:',c)

d=df[['India\_income','China\_income','Tokyo\_income','Kenya\_income']].min() ##min of all income columns

print('Minimum:',d)

#e=b.max(axis=1)

f=df[['India\_income','China\_income','Tokyo\_income','Kenya\_income']].mean() ## mean

print('Mean:',f)

g=df[['India\_income','China\_income','Tokyo\_income','Kenya\_income']].std() ## std deviation

print('Std deviation:',g)

h=df[['India\_income','China\_income','Tokyo\_income','Kenya\_income']].var() ## variance

print('Variance:',h)

i=df[['India\_income','China\_income','Tokyo\_income','Kenya\_income']].median() ##median

print('Median:',i)

j=df[['India\_income','China\_income','Tokyo\_income','Kenya\_income']].mode() ## mode

print('Mode:',j)

k=df[['India\_income','China\_income','Tokyo\_income','Kenya\_income']].quantile(0.1) ##10th percentile

print('10th percentile:',k)

l=df[['India\_income','China\_income','Tokyo\_income','Kenya\_income']].quantile(0.2) ##20th percentile

print('20th percentile:',l)

m=df[['India\_income','China\_income','Tokyo\_income','Kenya\_income']].quantile(0.25) ##1st quartile

print('1st quartile:',m)

n=df[['India\_income','China\_income','Tokyo\_income','Kenya\_income']].quantile(0.5) ##2nd quartile

print('2nd quartile:',n)

o=df[['India\_income','China\_income','Tokyo\_income','Kenya\_income']].quantile(0.75) ##3rd quartile

print('3rd quartile:',o)

r=o-m #intequartile range(IQR) (difference between 1st and 3rd quartiles)

print('IQR:',r)

#n=b[['India\_income','China\_income','Tokyo\_income','Kenya\_income']].percentile(10,interpolation='midpoint')

#print(n)

p=df[['India\_income','China\_income','Tokyo\_income','Kenya\_income']].skew() ##finding skewness

print('Skewness:',p)

q=df[['India\_income','China\_income','Tokyo\_income','Kenya\_income']].kurt() ##finding kurtosis

print('Kurtosis:',q)

**##pearson correlation matrix and spearman rank correlation matrix**

import numpy as np

import scipy.stats

import matplotlib.pyplot as plt

pc=df.corr(method='pearson')

pc

sc=df.corr(method='spearman')

sc

**##Discuss the relationships between the fields of df\_stats. For example, 2nd**

**quartile and the median are the same.**

Quartiles in measurements are values that separate your information into quarters.

Quartiles are classified into three types i.e. first, second, third quartiles.

* The first quartile (Q1): is defined as the middle number between the smallest number and the median of the given data set.
* The second quartile (Q2) :median of the given data set.
* The third quartile (Q3):is the middle number between the median and the largest value of the data set.
* Inter Quartile Relation(IQR):It is the difference between first and third quartiles

**Note:** The above discussion clearly shows that second quartile and median are same in the given or considered dataset.

**##Discuss how the columns of df\_stats are useful in data analysis.**

Used to express mean, median and mode of the dataset.

Also used to show the minimum,maximum values and other statistical discussions.

* When dealing with large volumes of datasets, statistics are very useful in the technical analysis and interpretation.
* Mean, median and mode metrics provide the information on data values at the center of dataset.
* Percentiles and quartiles explains about the spread of the data around that center.
* Standard deviation and variance describes how the data values are varying/ differing from the mean (like far or close to the mean).

**##Analyse data based on your discussion and explain the results. What are the notable**

**features of the dataset?**

In the considered dataset, we can observe the incomes of respective country for the respective year

For instance, in case of 'India\_income':

1. max income in the year 2009 is 28.2
2. min income in the year 2005 is 12.85
3. mean income is 12.85
4. median income is 22.05
5. standard deviation about 6.092824 is deviated from its mean value between 2005 to 2009
6. variance is 37.1225
7. 10th percentile is 15.71
8. first quartile is 20.0
9. second quartile is 22.05
10. third quartile is 26.7
11. India\_income is negatively skewed by -0.836460 between 2005 to 2009 year
12. kurtosis value is 0.235830

**Note:** From the results, it is observed that mean and the second quartile are same.

**##Create a Pearson correlation matrix (it is a square matrix) between all the possible fields.**

**What are the conclusions you make?**

Correlation is a statistical technique that can show the relation between the two sets of variables or between the features.

This can be classified as positive(0 to +1),negative(0 to -1) and no correlation(0) depends on the value of correlation coefficient ranges.

From the above results, Pearson correlation coefficient is negatively and positively correlated between +1 and -1.

**##Create a Spearman’s Rank correlation matrix (it is a square matrix) between all the**

**possible fields. What are the conclusions you make?**

The Spearman's rank correlation matrix is same as the concept of the Pearson correlation matrix which depend on the correlation coefficient.

where it additionally works with the rank values of themselves along with the values of datasets.

From the above results, Spearman’s Rank correlation is negatively and positively correlated between +1 and -1.

**##Create a seaborne pairplot for df\_product\_a. What are the conclusions you can make**

**using the analysis sofar.**

**## seaborne pairplot**

z=sns.pairplot(data=df[['Date','India\_income','China\_income','Tokyo\_income','Kenya\_income']],hue='Date')

* Built in themes for styling matplotlib graphics
* Visualizing univariate and bivariate data
* Fitting in and visualizing linear regression models
* Plotting statistical time series data
* Seaborne works well with NumPy and Pandas data structures
* It comes with built in themes for styling Matplotlib graphics

Seaborne results shows the plot between all incomes for different considered years (i.e. India\_income vs China\_income for all the years)

Hence these results explains the comparison between two countries incomes for different years.

**##Using Plotly, draw weekly and monthly time-series graphs of the numeric fields. Explain the results**

‘Plotly’ is another library in python to visualize the data in various designs like bar graphs, pie plots, etc.

The bar graph between the year and each country's income is shown graphically. (i.e. year vs India\_income, year vs China\_income, year vs Tokyo\_income and year vs Kenya\_income).

**#BAR PLOT**

x=[12.85,20,22.5,26.7,28.2]

y=[15.5,17.2,19,20.8,23]

plt.bar(x,y,width=0.5,color=['red','green'])

plt.xlabel('India\_income')

plt.ylabel('China\_income')

#b.plt.bar()

plt.show()

**#BAR PLOT**

x=[2005,2006,2007,2008,2009]

y=[12.85,20,22.5,26.7,28.2]

plt.bar(x,y,width=0.5,color=['red','green'])

plt.xlabel('India\_income')

plt.ylabel('China\_income')

#b.plt.bar()

plt.show()

**#BAR PLOT**

import numpy as np, pandas as pd, matplotlib.pyplot as plt

x1=[1,2,3,4,5]

y1=[3,4,5,6,7]

x2=[7,8,9,11,13]

y2=[12,13,14,15,16]

plt.plot(x1,y1,label='line1')

plt.plot(x2,y2,label='line2')

plt.xlabel('speed')

plt.ylabel('distance')

plt.title('Journey')

plt.legend()

plt.show()

**#SCATTER PLOT**

import numpy as np,pandas as pd,matplotlib.pyplot as plt

x=[1,2,3,4,5,6,7,8,9,10]

y=[11,12,13,14,15,16,17,18,19,20]

plt.scatter(x,y,label='stars',color='red',s=20)

plt.show()

**##Compare and contrast the prices of each type, each location and {location and type} combination. Visualize the results using suitable plots.**

**#Prices of radio item in different years using Bar plot**

import numpy as np, matplotlib.pyplot as plt

df=pd.read\_csv('Income\_a.csv')

tick\_label=(2005,2006,2007,2008,2009,2010,2011,2012)

plt.bar(df['RADIO\_ITEM'],df['RADIO\_PRICES'],tick\_label=tick\_label,width=0.5,color=['red','green'])

plt.xlabel("RADIO\_ITEM")

plt.ylabel("RADIO\_PRICES")

plt.show()

**#Prices of tv item in different years using Bar plot**

import numpy as np, matplotlib.pyplot as plt

df=pd.read\_csv('Income\_a.csv')

tick\_label=(2005,2006,2007,2008,2009,2010,2011,2012)

plt.bar(df['TV\_ITEM'],df['TV\_PRICES'],tick\_label=tick\_label,width=0.5,color=['red','green'])

plt.xlabel("TV\_ITEM")

plt.ylabel("TV\_PRICES")

plt.show()

**#Prices of tv item in different years using Line plot**

df=pd.read\_csv('Income\_a.csv')

plt.plot(df['TV\_ITEM'],df['TV\_PRICES'],color='green')

plt.xlabel("TV\_ITEM")

plt.ylabel("TV**\_**PRICES")

plt.show()

**#Prices of radio item in different years using Line plot**

df=pd.read\_csv('Income\_a.csv')

plt.plot(df['RADIO\_ITEM'],df['RADIO\_PRICES'],color='green')

plt.xlabel("RADIO\_ITEM")

plt.ylabel("RADIO\_PRICES")

plt.show()

**#Prices of tv item in different locations using Bar plot**

df=pd.read\_csv('Income\_a.csv')

tick\_label=('Footscray','Brooklyn','Braybrook','Napier st','Kingsville','Kings st','Falcon st','Para st')

plt.bar(df['TV\_ITEM'],df['TV\_PRICES'],tick\_label=tick\_label,width=0.5,color=['red','green'])

plt.xlabel("TV\_ITEM")

plt.ylabel("TV\_PRICES")

plt.show()

**#Number of Tv detected items in different locations using Bar plot**

df=pd.read\_csv('Income\_a.csv')

tick\_label=('Footscray','Brooklyn','Braybrook','Napier st','Kingsville','Kings st','Falcon st','Para st')

plt.bar(df['TV\_ITEM'],df['TVDETECTED\_ITEMS'],tick\_label=tick\_label,width=0.5,color=['red','green'])

plt.xlabel("TV\_ITEM")

plt.ylabel("TVDETECTED\_ITEMS")

plt.show()

**#Number of Radio detected items in different locations using Bar plot**

df=pd.read\_csv('Income\_a.csv')

tick\_label=('Footscray','Brooklyn','Braybrook','Napier st','Kingsville','Kings st','Falcon st','Para st')

plt.bar(df['RADIO\_ITEM'],df['RADIODETECTED\_ITEMS'],tick\_label=tick\_label,width=0.5,color=['red','green'])

plt.xlabel("RADIO\_ITEM")

plt.ylabel("RADIODETECTED\_ITEMS")

plt.show()

**#Number of Radio detected items in different locations using line plot**

df=pd.read\_csv('Income\_a.csv')

plt.plot(df['RADIO\_ITEM'],df['RADIODETECTED\_ITEMS'],color='green')

plt.xlabel("RADIO\_ITEM")

plt.ylabel("RADIODETECTED\_ITEMS")

plt.show()

**##Visualize data on a folium map. The locations should have markers with a color range based on the mean values of bags\_t. Tooltips should show the total values of bags\_t and total values of bag\_t for each type. When markers are clicked, the average values of all numeric fields should be shown.**

import folium

map=folium.Map(location=[-37.813629,144.963058],zoom\_start=15)

folium.Marker([-37.813629,144.963058], popup='Melbourne',icon=folium.Icon(color='Red')).add\_to(map)

map

**#Adding polyline locations**

import folium

map=folium.Map(location=[-37.813629,144.963058],zoom\_start=15)

folium.Marker([-37.813629,144.963058],popup='Melbourne').add\_to(map)

folium.Marker([-33.868820,151.209290],popup='Sydney').add\_to(map)

folium.PolyLine(locations=[(-37.813629,144.963058),(-33.868820,151.209290)],line\_opacity=1.0).add\_to(map)

map

**#circular marker**

import folium

map=folium.Map(location=[-37.813629,144.963058],zoom\_start=15)

folium.CircleMarker([-37.813629,144.963058],popup='Melbourne').add\_to(map)

map

**#Adding 5 city names as circle marker**

import numpy as np

import pandas as pd

import folium

df=pd.read\_csv('Income\_a.csv')

df.head()

import folium

import numpy as np

import pandas as pd

df=pd.read\_csv('Income\_a.csv')

df.head()

lat=-37.813629

lang=144.963058

folium.Map(location=[lat,lang],zoom\_start=12)

colors={'ES':'red','FS':'blue',

}

melbourne\_circle=folium.Map(location=[lat,lang],zoom\_start=12)

for lat,lng,num in zip(df.X,df.Y,range(1,df.shape[0])):

popup=folium.Popup(df['CITY\_NAME'][num],parse\_html=True)

folium.CircleMarker(

[lat,lng],

radius=6,

color=colors[df.iloc[num,-1]],

fill=True,

fill\_color=colors[df.iloc[num,-1]],

fill\_opacity=0.7,

popup=popup

).add\_to(melbourne\_circle)

melbourne\_circle

**#Clustering the map showing Average values**

import folium

import numpy as np

import pandas as pd

df=pd.read\_csv('Income\_a.csv')

df.head()

lat=-37.813629

lang=144.963058

folium.Map(location=[lat,lang],zoom\_start=12)

colors={'FS':22.5,'ES':19.0,'FS':10.2,'ES':11.8,'ES':23,

}

melbourne\_cluster=folium.Map(location=[lat,lang],zoom\_start=12)

from folium import plugins

cluster=plugins.MarkerCluster().add\_to(melbourne\_cluster)

melbourne\_Marker=folium.Map(location=[lat,lang],zoom\_start=12)

for lat,lng,num in zip(df.Y,df.X,range(1,df.shape[0])):

popup = folium.Popup(df['CITY\_NAME'][num], parse\_html=True)

folium.Marker(

[lat,lng],

popup=popup,

icon=folium.Icon(color=colors[df.iloc[num,-1]])

).add\_to(melbourne\_cluster)

melbourne\_cluster

**#Map showing Average values as circle marker**

import folium

import numpy as np

import pandas as pd

df=pd.read\_csv('Income\_a.csv')

df.head()

lat=-37.813629

lang=144.963058

folium.Map(location=[lat,lang],zoom\_start=12)

colors={'ES':[22.5,19.0,10.2,11.8],'FS':[22.5,19.0,10.2,11.8] }

melbourne\_circle=folium.Map(location=[lat,lang],zoom\_start=12)

for lat,lng,num in zip(df.X,df.Y,range(1,df.shape[0])):

popup=folium.Popup(df['AVG\_INCOME'][num],parse\_html=True)

folium.CircleMarker(

[lat,lng],

radius=15,

color=colors[df.iloc[num,-1]],

fill=True,

fill\_color=colors[df.iloc[num,-1]],

fill\_opacity=0.7,

popup=popup

).add\_to(melbourne\_circle)

melbourne\_circle