**EXPERIMENT 2**

**Name**

**-**

**SANSKAR AGRAWAL**

**Branch**

**-**

**CSE**

**Semester**

**-**

**5**

**th**

**Subject**

**Name**

**–**

**Machine**

**Learning**

**Lab**

**Subject**

**Code**

**-**

**CSP**

**20**

**-**

**317**

**UID**

**-**

**20**

**BCS**

**5914**

**Section/Group**

**-**

**806**

**B**

**Date**

**of**

**Performance**

**-**

**28/08/2022**

# AIM - Data Visualization

**OBJECTIVE –** To analyze the data for certain trends, patterns may become difficult if the data is in its raw format. To overcome this data visualization comes into play. Data visualization provides a good, organized pictorial representation of the data which makes it easier to understand, observe, analyze.

**Steps Involved-**

Python provides various libraries that come with different features for visualizing data.

All these libraries come with different features and can support various types of graphs. The four main libraries are: Matplotlib, Seaborn, Bokeh, Plotly

**S/W Requirement: -** VS Code or Jupyter Notebook

## INPUT AND OUTPUT –

import pandas as pd

data=pd.read\_csv("/content/tips.csv")

data.head(10)



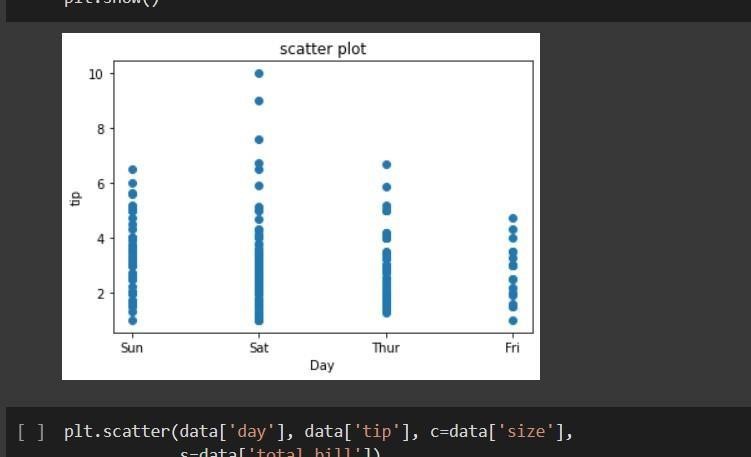
**Scatter Plot :-**

Scatter plots are used to observe relationships between variables and uses dots to represent the relationship between them. The scatter() method in the matplotlib library is used to draw a scatter plot.

pip install matplotlib

import matplotlib.pyplot as plt plt.scatter(data['day'], data['tip']) plt.title('scatterplot') plt.xlabel('Day')

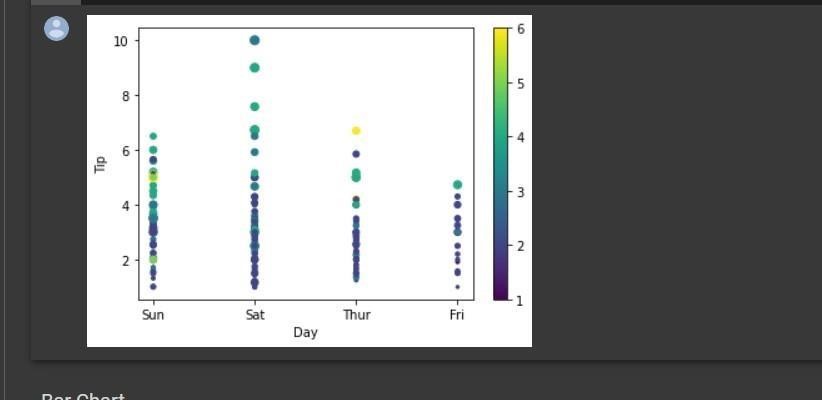
plt.ylabel('tip') plt.show()



plt.scatter(data['day'], data['tip'], c=data['size'], s=data['total\_bill']) plt.xlabel('Day') plt.ylabel('Tip')

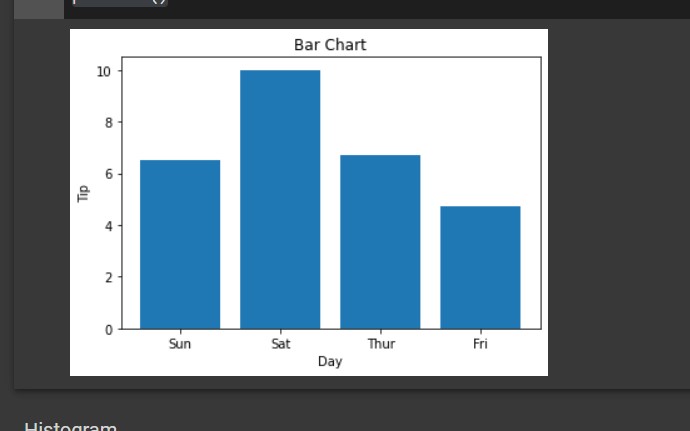
plt.colorbar()

plt.show()



**Bar Chart:--**

A bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent. It can be created using the bar() method.

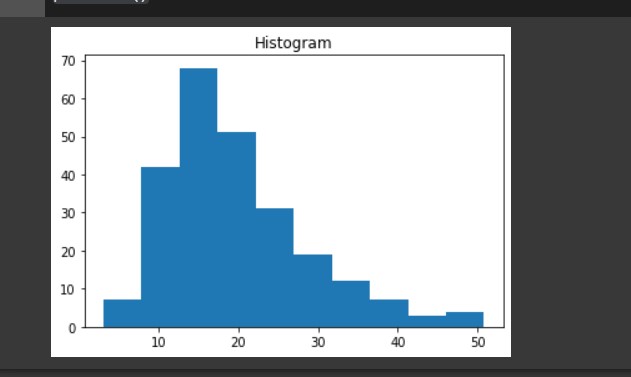
 plt.bar(data['day'], data['tip']) plt.title("Bar Chart") plt.xlabel('Day') plt.ylabel('Tip')

plt.show()

Histogram:-

A histogram is basically used to represent data in the form of some groups. It is a type of bar plot where the X-axis represents the bin ranges while the Y-axis gives information about frequency. The hist() function is used to compute and create a histogram. In histogram, if we pass categorical data then it will automatically compute the frequency of that data i.e. how often each value occurred.

plt.hist(data['total\_bill']) plt.title("Histogram") plt.show()



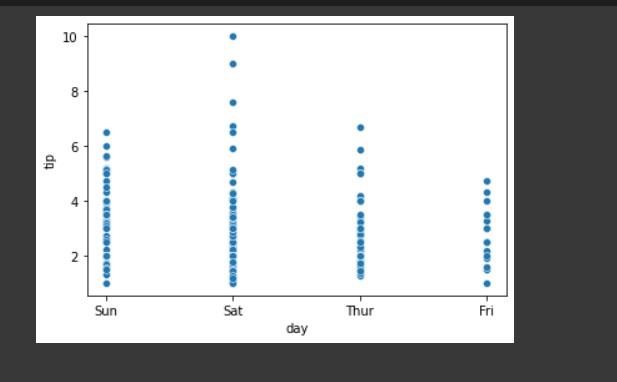
**Seaborn:-**

Seaborn is a high-level interface built on top of the Matplotlib. It provides beautiful design styles and color palettes to make more attractive graphs.

**Scatter Plot :-**

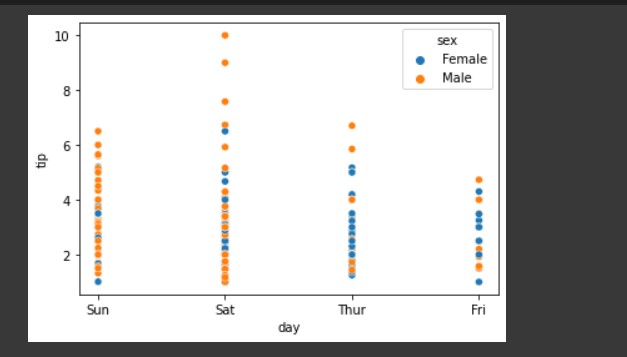
Scatter plot is plotted using the scatterplot() method. This is similar to Matplotlib, but additional argument data is required.

import seaborn as sns import matplotlib.pyplot as plt import pandas as pd sns.scatterplot(x='day', y='tip', data=data,) plt.show()



sns.scatterplot(x='day', y='tip', data=data,hue='sex')

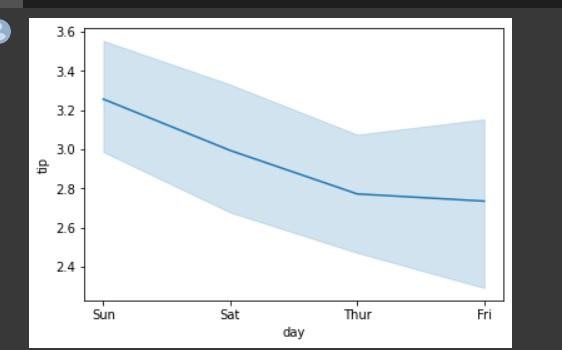
plt.show()



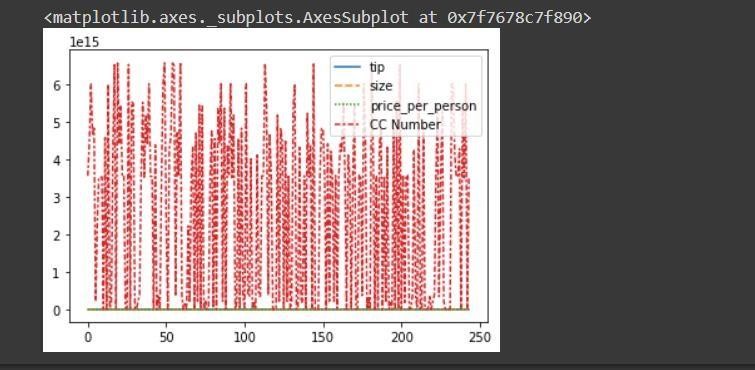
**Line Plot:-**

Line Plot in Seaborn plotted using the lineplot() method. In this, we can pass only the data argument also.

sns.lineplot(x='day', y='tip', data=data) plt.show()



sns.lineplot(data=data.drop(['total\_bill'], axis=1))

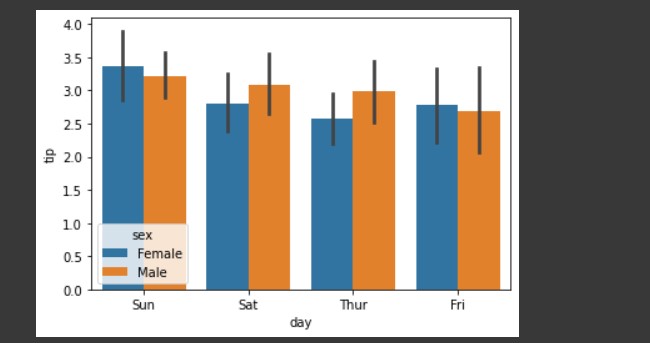


**Bar Plot:-:-**

Bar Plot in Seaborn can be created using the barplot() method.

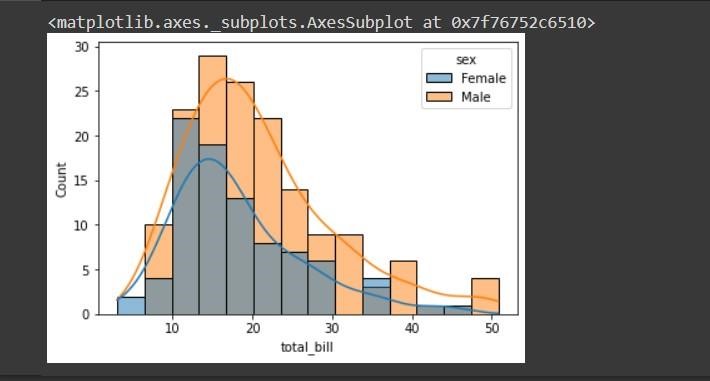
sns.barplot(x='day',y='tip', data=data, hue='sex')

plt.show()



**Histogram:-** The histogram in Seaborn can be plotted using the histplot() function.

sns.histplot(x='total\_bill', data=data, kde=True, hue='sex')



**Bokeh:-**

Bokeh is mainly famous for its interactive charts visualization. Bokeh renders its plots using HTML and JavaScript that uses modern web browsers for presenting elegant, concise construction of novel graphics with high-level interactivity.

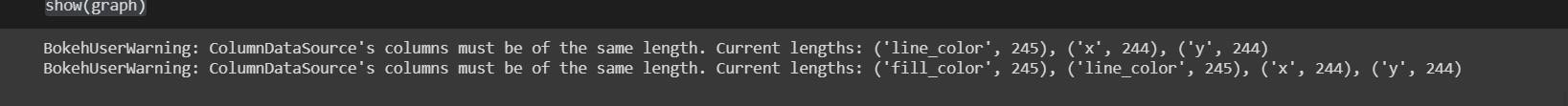
**Scatter Plot:-**

Scatter Plot in Bokeh can be plotted using the scatter() method of the plotting module. Here pass the x and y coordinates respectively.

pip install bokeh from bokeh.plotting import figure, output\_file, show from bokeh.palettes import magma import pandas as pd

graph = figure(title = "Bokeh Scatter Graph") data = pd.read\_csv("tips.csv") color = magma(245)

graph.scatter(data['total\_bill'], data['tip'], color=color) show(graph)



**Line Chart:-**A line plot can be created using the line() method of the plotting module.

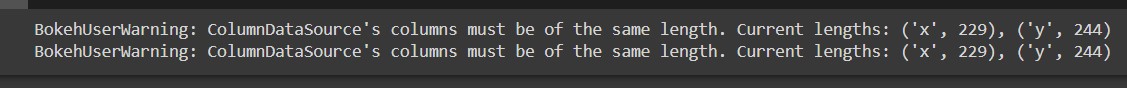
# importing the modules from bokeh.plotting import figure, output\_file, show import pandas as pd

# instantiating the figure object graph = figure(title = "Bokeh Bar Chart")

# reading the database data = pd.read\_csv("tips.csv")

# Count of each unique value of tip column df = data['tip'].value\_counts() df = data['total\_bill'].value\_counts()

# plotting the graph graph.line(df, data['tip']) graph.line(df, data['total\_bill']) # displaying the model show(graph)



**Bar Chart:-**

Bar Chart can be of two types horizontal bars and vertical bars. Each can be created using the hbar() and vbar() functions of the plotting interface respectively.

from bokeh.plotting import figure, output\_file, show import pandas as pd

graph = figure(title = "Bokeh Bar Chart") data = pd.read\_csv("tips.csv")

graph.vbar(data['total\_bill'], top=data['tip']) show(graph)

import plotly.express as px import pandas as pd

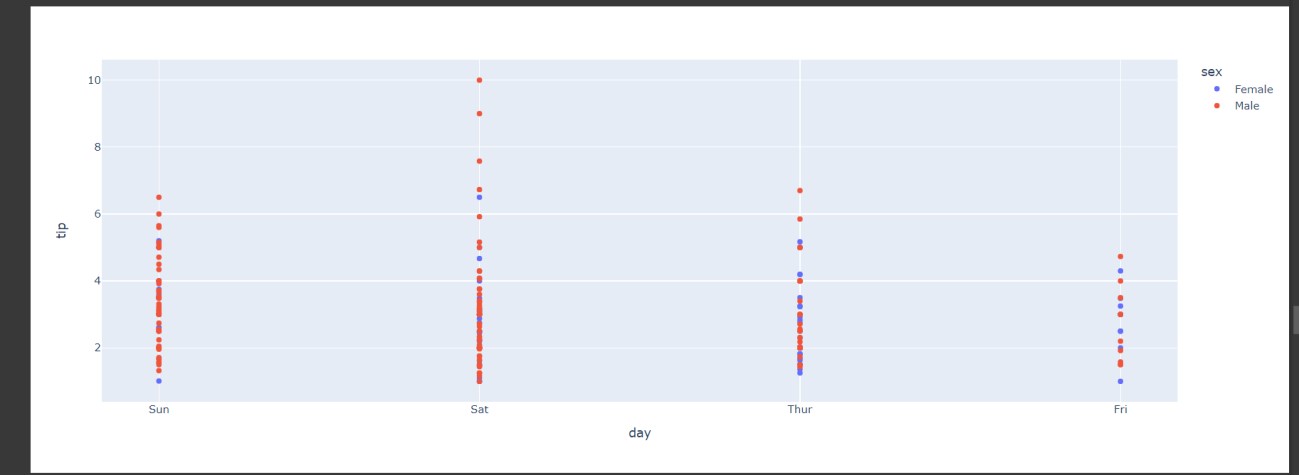
# reading the database

data = pd.read\_csv("tips.csv")

# plotting the scatter chart

fig = px.scatter(data, x="day", y="tip", color='sex')

# showing the plot fig.show()



**Line Chart:-**

Line plot in Plotly is much accessible and illustrious annexation to plotly which manage a variety of types of data and assemble easy-to-style statistic. With px.line each data position isrepresented as a vertex

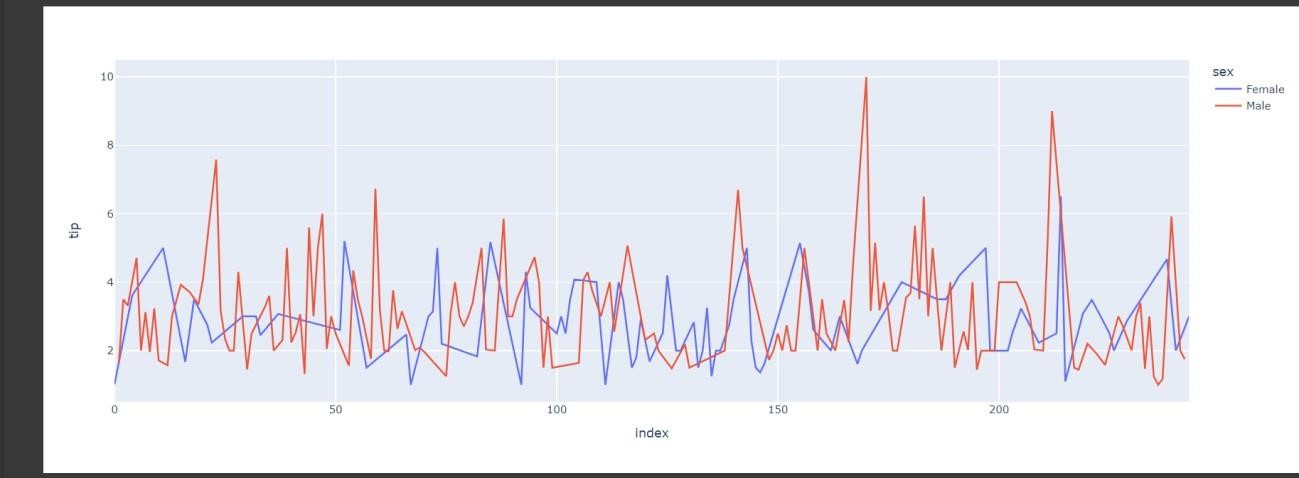
import plotly.express as px

import pandas as pd

# reading the database data = pd.read\_csv("tips.csv")

# plotting the scatter chart fig = px.line(data, y='tip', color='sex')

# showing the plot fig.show()



import plotly.express as px

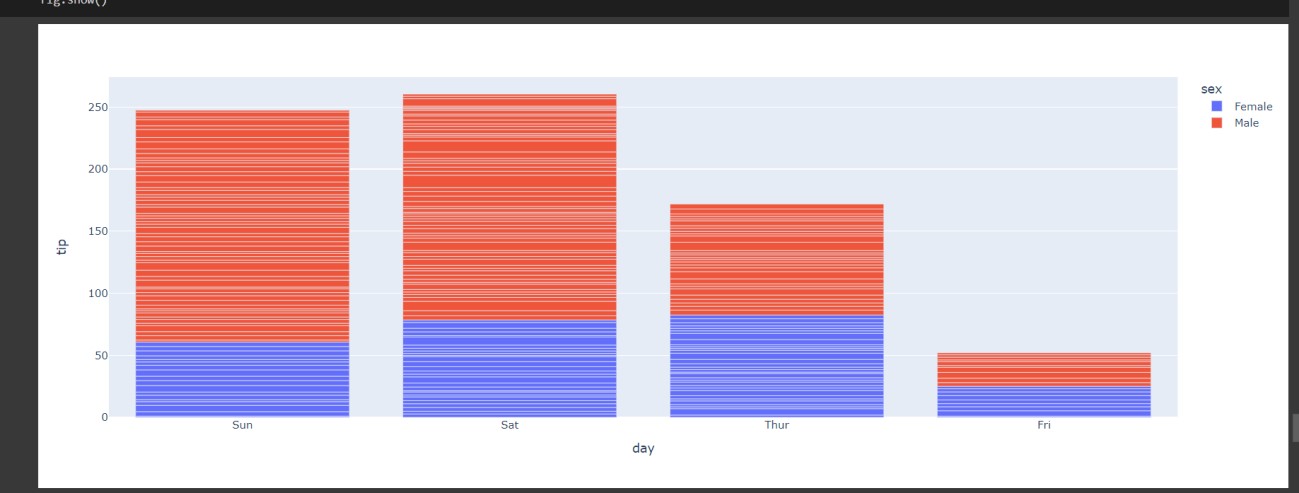
import pandas as pd

# reading the database data = pd.read\_csv("tips.csv")

# plotting the scatter chart

fig = px.bar(data, x='day', y='tip', color='sex')

# showing the plot fig.show()

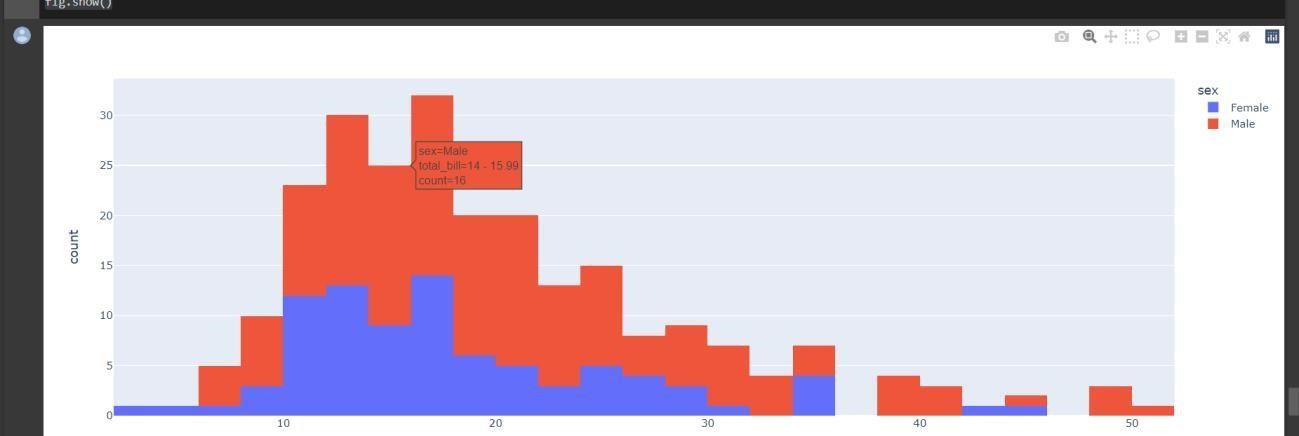


import plotly.express as px import pandas as pd

# reading the database data = pd.read\_csv("tips.csv")

# plotting the scatter chart fig = px.histogram(data, x='total\_bill', color='sex')

# showing the plot fig.show()



**Learning outcomes (What I have learnt) -**

1. How to establish relationship between different columns (variables) in a dataset using Graphs.
2. To draw different kind of graphs and charts by using different kind of libraries.
3. To find Relationship between different variables and map different type of Graphs.
4. Detailed introduction to visualization using matplot , seaborn , plotyl and bokeh libraries.

**Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):**

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. |  |  |  |
| 2. |  |  |  |
| 3. |  |  |  |
|  |  |  |  |