**Python Libraries for Data Visualization**

**Aim:**

To write a program for python Libraries for Data Visualization(matplotlib, plotly, seaborn, ggplot, altair, bokeh, pygal, geoplotlib)

**Algorithm :**

1. **Set the Objective:** Define the purpose of visualization (e.g., show trends, comparisons, or distributions).
2. **Collect and Clean Data:** Gather the dataset and handle missing, duplicate, or inconsistent values.
3. **Choose Visualization Type:** Select the most suitable chart type based on data and goals.
4. **Prepare the Data:** Filter, aggregate, or transform data to match the chosen visualization.
5. **Generate Initial Plot:** Use a library (e.g., Matplotlib, Seaborn) to create a basic visualization.
6. **Enhance Visualization:** Add titles, axis labels, legends, and colors for clarity and context.
7. **Analyze and Refine:** Check if the visualization conveys the desired insights and make adjustments.
8. **Validate with Stakeholders:** Share with others to confirm that it meets the intended objectives.
9. **Save or Share:** Export the visualization in a suitable format (image, HTML, etc.).
10. **Iterate if Needed:** Incorporate feedback and refine until the visualization is effective.
11. **Program using** **Matplotlib**:

**#Line plot**   
import matplotlib.pyplot as plt

import numpy as np

x = np.linspace(0, 10, 100)

y = np.sin(x)

plt.plot(x, y, label='sin(x)', color='blue')

plt.title("Matplotlib Line Plot")

plt.xlabel("X-axis")

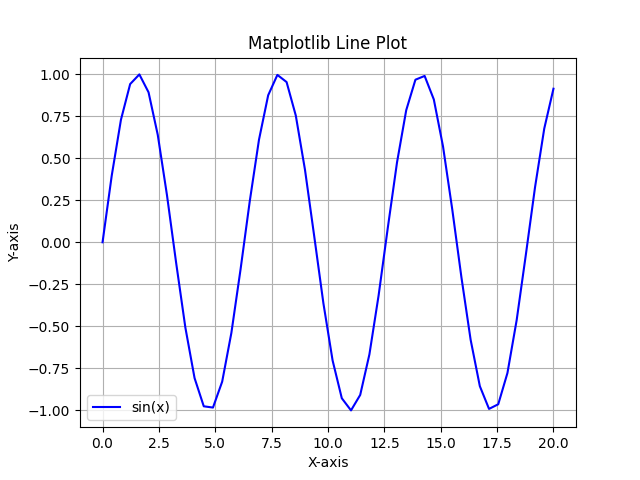
plt.ylabel("Y-axis")

plt.legend()

plt.grid(True)

plt.show()

**Output:**



1. **Program for** **plotly**

**#Interactive Scatter Plot**

import plotly.express as px

import pandas as pd

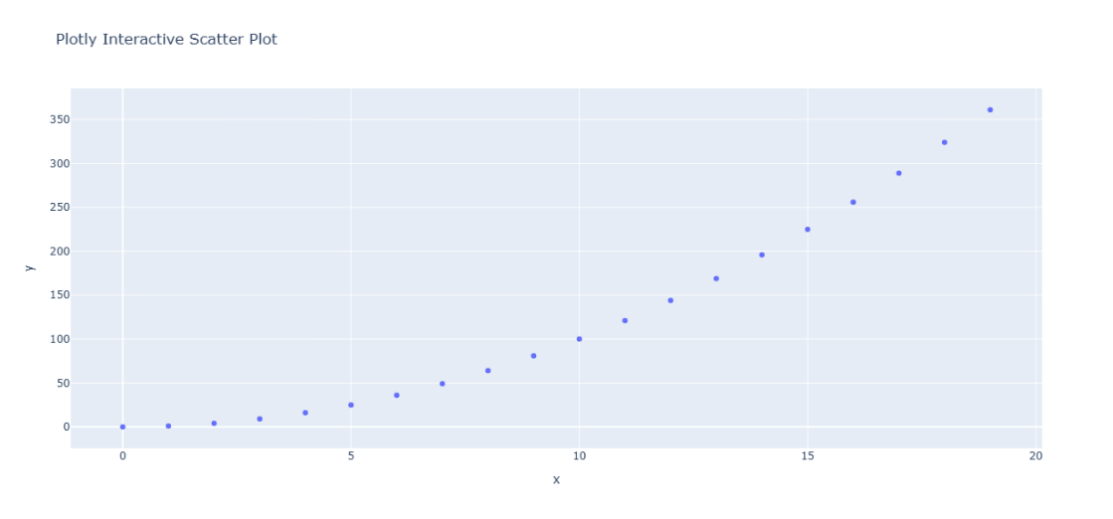
*# Sample data*

df = pd.DataFrame({'x': range(20), 'y': [i\*\*2 for i in range(20)]})

fig = px.scatter(df, x='x', y='y', title="Plotly Interactive Scatter Plot")

fig.show()

**Output:**



1. **Program for Seaborn:**

**#Histogram**

import seaborn as sns

import matplotlib.pyplot as plt  *# Import this to use plt functions*

import numpy as np

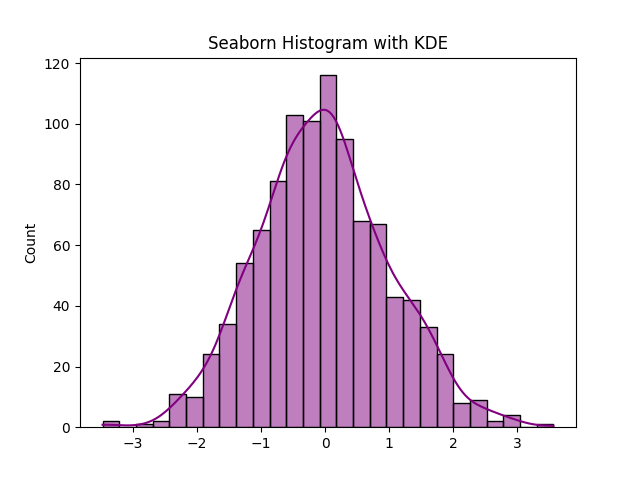
data = np.random.randn(1000)

sns.histplot(data, kde=True, color='purple')

plt.title("Seaborn Histogram with KDE")  *# This uses plt to set the title*

plt.show()

**Output:**

****

1. **Program for ggplot (Using Plotnine)**

**#Bar Chart**

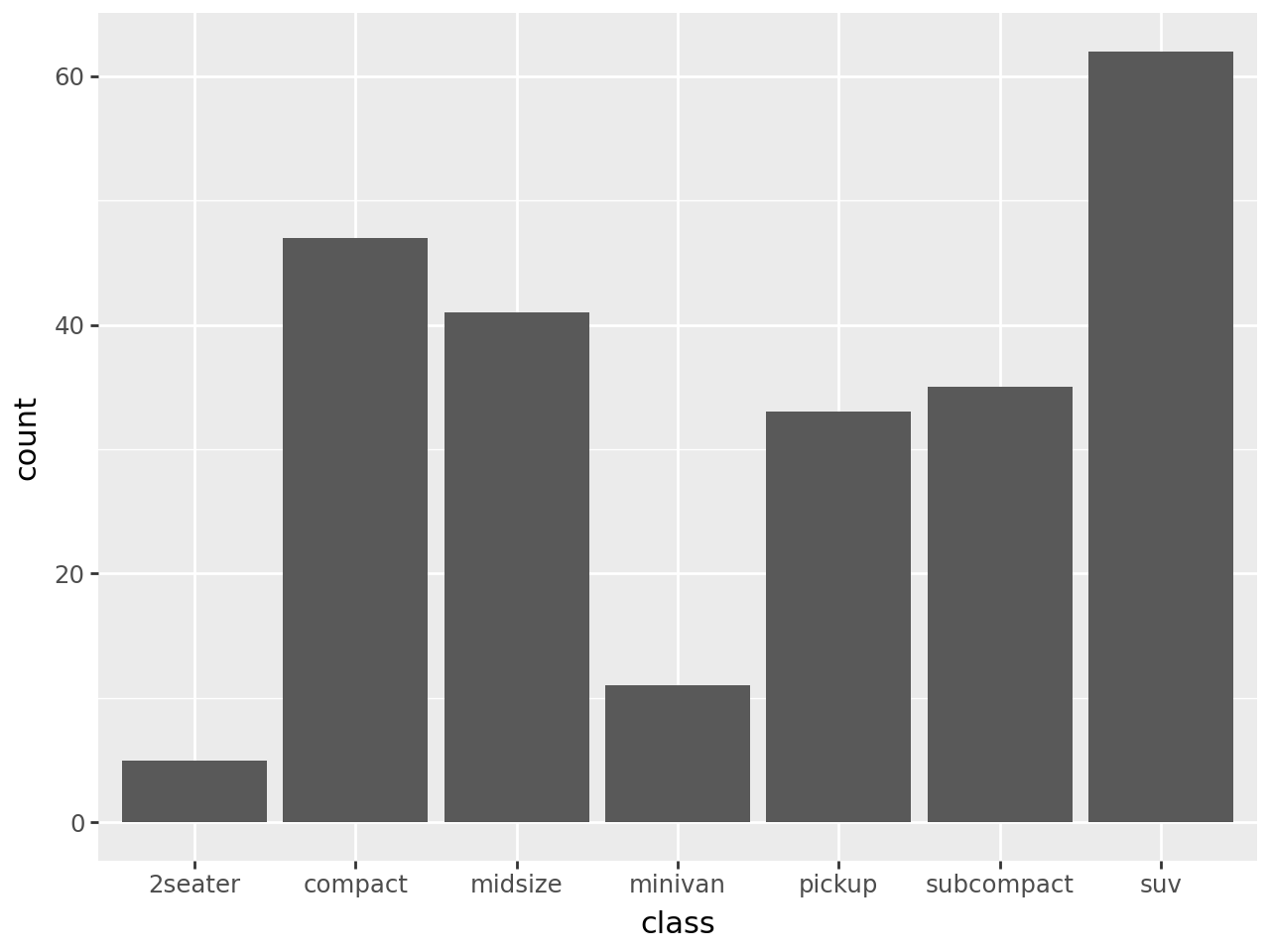
**#use jupyter notebook to get output**

from plotnine.data import mpg

from plotnine import ggplot, aes, geom\_bar

ggplot(mpg) + aes(x="class") + geom\_bar()

**Output:**

****

1. **Program with Altair**

**# Interactive Scatter Plot**

**# use jupyter notebook to get output**

import altair as alt

import pandas as pd

df = pd.DataFrame({'x': range(10), 'y': [i\*\*2 for i in range(10)]})

chart = alt.Chart(df).mark\_circle(size=100).encode(

    x='x',

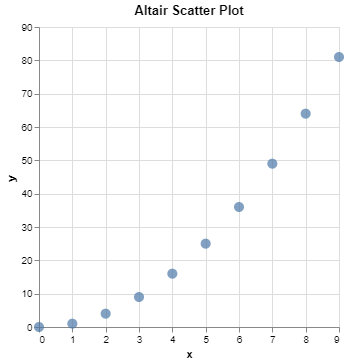
    y='y',

    tooltip=['x', 'y']

).properties(title="Altair Scatter Plot")

chart.show()

**Output:**

****

1. **Program using Bokeh**

**#interactive line plot**

**# use jupyter notebook to get output**

from bokeh.plotting import figure, show

from bokeh.io import output\_notebook

import numpy as np

x = np.linspace(0, 10, 100)

y = np.sin(x)

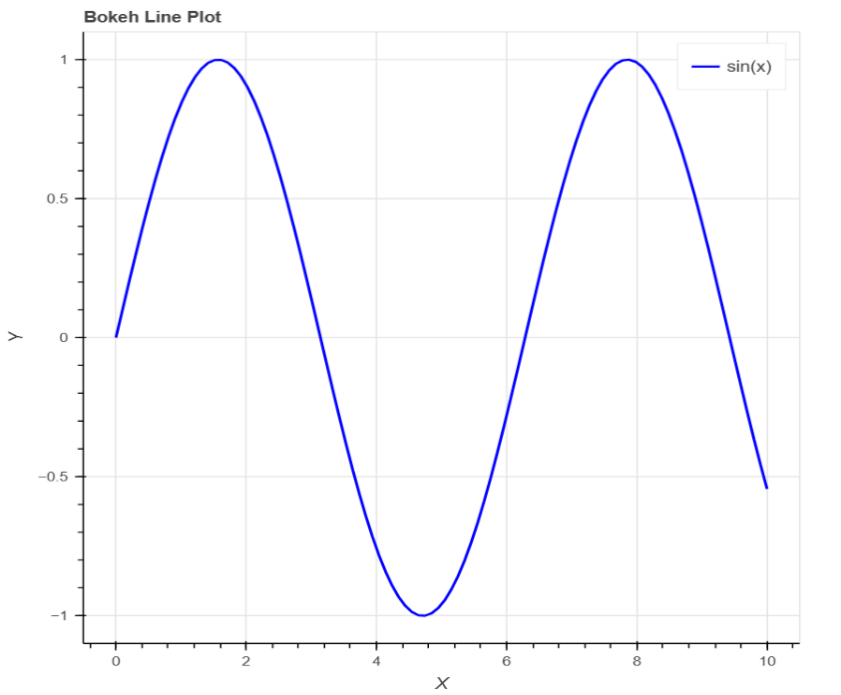
output\_notebook()

p = figure(title="Bokeh Line Plot", x\_axis\_label='X', y\_axis\_label='Y')

p.line(x, y, legend\_label="sin(x)", line\_width=2, color="blue")

show(p)

**Output:**

****

1. **Program using Pygal**

**#Bar chart**

import pygal

bar\_chart = pygal.Bar()

bar\_chart.title = "Pygal Bar Chart"

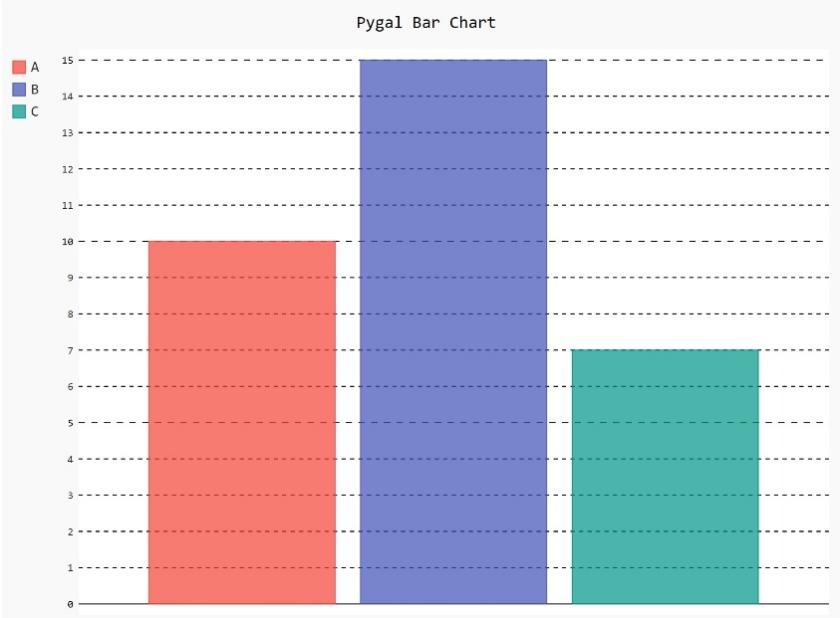
bar\_chart.add('A', 10)

bar\_chart.add('B', 15)

bar\_chart.add('C', 7)

bar\_chart.render\_in\_browser()

Output:

****

1. **Program using Altair:**

**#heatmap**

**#use jupyter notebook to get output**

import altair as alt

import pandas as pd

import numpy as np

# Create a sample dataset

data = pd.DataFrame({

'x': np.repeat(range(10), 10),

'y': list(range(10)) \* 10,

'z': np.random.rand(100)

})

chart = alt.Chart(data).mark\_rect().encode(

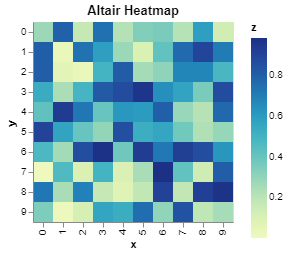
x='x:O',

y='y:O',

color='z:Q'

).properties(title="Altair Heatmap")

chart.show()

**Output:**

**Result:**

Thus the python program to implement python libraries for data visualization is executed successfully.