

## MATPLOTLIB ASSIGNMENT:

### 1. Scatter Plot

To visualize the relationship between  $x$  and  $y$ :

```
python
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
y = [2, 4, 5, 7, 6, 8, 9, 10, 12, 13]

plt.scatter(x, y)
plt.xlabel('x')
plt.ylabel('y')
plt.title('Scatter Plot of x vs y')
plt.show()
```

### Line Plot

To visualize the trend of values:

```
python

plt.plot(x, y, marker='o')
plt.xlabel('x')
plt.ylabel('y')
plt.title('Line Plot of x vs y')
plt.show()
```

### 3. Bar Chart

To represent the frequency of each item in categories:

```
python

categories = ['A', 'B', 'C', 'D', 'E']
values = [25, 40, 30, 35, 20]

plt.bar(categories, values)
plt.xlabel('Categories')
plt.ylabel('Values')
plt.title('Bar Chart of Categories')
plt.show()
```

### 4. Histogram

To visualize the distribution of values in data:

```
python

data = [1, 2, 2, 3, 3, 3, 4, 4, 4, 4, 5, 5, 5, 5, 5]
```

```
plt.hist(data, bins=5)
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Histogram of Data')
plt.show()
```

## 5. Pie Chart

To represent the percentage distribution of different sections:

python

```
sections = ['Section A', 'Section B', 'Section C', 'Section D']
sizes = [25, 30, 15, 30]

plt.pie(sizes, labels=sections, autopct='%1.1f%%')
plt.title('Pie Chart of Sections')
plt.show()
```

### *SEABORN ASSIGNMENT:1. Scatter Plot with Synthetic Dataset*

python

```
import numpy as np

x = np.random.rand(50)
y = np.random.rand(50)

plt.scatter(x, y)
plt.xlabel('x')
plt.ylabel('y')
plt.title('Scatter Plot with Synthetic Dataset')
plt.show()
```

### *2. Distribution of Random Numbers*

python

```
data = np.random.randn(1000)

plt.hist(data, bins=30)
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.title('Histogram of Random Numbers')
plt.show()
```

### *3. Comparing Categories Based on Numerical Values*

python

```
categories = ['A', 'B', 'C', 'D', 'E']
values = [25, 40, 30, 35, 20]

plt.bar(categories, values)
plt.xlabel('Categories')
plt.ylabel('Values')
plt.title('Bar Chart Comparing Categories')
plt.show()
```

### *4. Distribution Across Categories*

python

```
import seaborn as sns
```

```

data = np.random.randint(1, 100, size=(100, 3))
categories = np.random.choice(['A', 'B', 'C'], size=100)
df = pd.DataFrame(data, columns=['Value1', 'Value2', 'Value3'])
df['Category'] = categories

sns.boxplot(x='Category', y='Value1', data=df)
plt.title('Box Plot of Value1 Across Categories')
plt.show()

```

## 5. Correlation Matrix Heatmap

python

```

import seaborn as sns

data = np.random.rand(10, 12)
df = pd.DataFrame(data)

corr = df.corr()
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.title('Heatmap of Correlation Matrix')

```

**PLOTLY ASSIGNMENT: (Use Plotly for the visualization of the given questions)**

### 1. 3D Scatter Plot using Matplotlib:

python

```

import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
import pandas as pd

# Generate the data
np.random.seed(30)
data = {
    'X': np.random.uniform(-10, 10, 300),
    'Y': np.random.uniform(-10, 10, 300),
    'Z': np.random.uniform(-10, 10, 300)
}
df = pd.DataFrame(data)

# Create a 3D scatter plot
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.scatter(df['X'], df['Y'], df['Z'])

ax.set_xlabel('X Label')
ax.set_ylabel('Y Label')
ax.set_zlabel('Z Label')
ax.set_title('3D Scatter Plot')
plt.show()

```

### 2. Violin Plot using Seaborn:

python

```

import seaborn as sns

```

```
# Generate the data
np.random.seed(15)
data = {
    'Grade': np.random.choice(['A', 'B', 'C', 'D', 'F'], 200),
    'Score': np.random.randint(50, 100, 200)
}
df = pd.DataFrame(data)

# Create a violin plot
sns.violinplot(x='Grade', y='Score', data=df)
plt.title('Violin Plot of Scores by Grade')
plt.show()
```

### 3. Heatmap using Seaborn:

python

```
# Generate the data
np.random.seed(20)
data = {
    'Month': np.random.choice(['Jan', 'Feb', 'Mar', 'Apr', 'May'], 100),
    'Day': np.random.choice(range(1, 31), 100),
    'Sales': np.random.randint(1000, 5000, 100)
}
df = pd.DataFrame(data)

# Pivot the data for heatmap
pivot_table = pd.pivot_table(df, values='Sales', index='Day',
                              columns='Month', aggfunc=np.sum, fill_value=0)

# Create a heatmap
sns.heatmap(pivot_table, annot=True, fmt='d', cmap='YlGnBu')
plt.title('Heatmap of Sales')
plt.show()
```

## Using Plotly

### 4. 3D Surface Plot:

python

```
import plotly.graph_objs as go
import numpy as np

x = np.linspace(-5, 5, 100)
y = np.linspace(-5, 5, 100)
x, y = np.meshgrid(x, y)
z = np.sin(np.sqrt(x**2 + y**2))

data = go.Surface(z=z, x=x, y=y)
layout = go.Layout(title='3D Surface Plot', autosize=False, width=800,
                    height=800)

fig = go.Figure(data=[data], layout=layout)
fig.show()
```

### 5. Bubble Chart:

python

```
import plotly.express as px

# Generate the data
np.random.seed(25)
data = {
    'Country': ['USA', 'Canada', 'UK', 'Germany', 'France'],
    'Population': np.random.randint(100, 1000, 5),
    'GDP': np.random.randint(500, 2000, 5)
}
df = pd.DataFrame(data)

# Create a bubble chart
fig = px.scatter(df, x='GDP', y='Population', size='Population',
color='Country', hover_name='Country',
                  title='Bubble Chart of Country Population vs GDP',
size_max=60)
fig.show()
```

## Using Bokeh

### 1. Sine Wave Plot:

python

```
from bokeh.plotting import figure, show
from bokeh.io import output_notebook
import numpy as np

output_notebook()

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

p = figure(title="Sine Wave", x_axis_label='X', y_axis_label='Y')
p.line(x, y, legend_label="Sine", line_width=2)

show(p)
```

### 2. Scatter Plot:

python

```
from bokeh.plotting import figure, show
from bokeh.io import output_notebook
import numpy as np

output_notebook()

np.random.seed(30)
x = np.random.rand(100)
y = np.random.rand(100)
sizes = np.random.randint(10, 100, 100)
colors = np.random.randint(1, 256, 100)

p = figure(title="Scatter Plot", x_axis_label='X', y_axis_label='Y')
p.scatter(x, y, size=sizes, color=colors, alpha=0.6)
```

```
show(p)
```

### 3. Bar Chart:

python

```
from bokeh.plotting import figure, show
from bokeh.io import output_notebook

output_notebook()

fruits = ['Apples', 'Oranges', 'Berries', 'Bananas', 'Grapes']
counts = [10, 20, 30, 40, 50]

p = figure(x_range=fruits, title="Fruit Counts", toolbar_location=None,
tools="")
p.vbar(x=fruits, top=counts, width=0.9)

show(p)
```

### 4. Histogram:

python

```
from bokeh.plotting import figure, show
from bokeh.io import output_notebook
import numpy as np

output_notebook()

data = np.random.randn(1000)
hist, edges = np.histogram(data, bins=50)

p = figure(title="Histogram", x_axis_label='Value',
y_axis_label='Frequency')
p.quad(top=hist, bottom=0, left=edges[:-1], right=edges[1:],
fill_color="navy", line_color="white", alpha=0.5)

show(p)
```

### 5. Heatmap:

python

```
from bokeh.plotting import figure, show
from bokeh.io import output_notebook
from bokeh.transform import linear_cmap
from bokeh.models import ColorBar, BasicTicker, LinearColorMapper
import pandas as pd

output_notebook()

data = pd.DataFrame({
    'x': np.random.randint(0, 10, 100),
    'y': np.random.randint(0, 10, 100),
    'value': np.random.random(100)
})
```

```

mapper = LinearColorMapper(palette="Viridis256", low=data['value'].min(),
high=data['value'].max())

p = figure(title="Heatmap", x_range=(0, 10), y_range=(0, 10),
toolbar_location=None, tools="")
p.rect(x="x", y="y", width=1, height=1, source=data, line_color=None,
fill_color=linear_cmap('value', 'Viridis256', data['value'].min(),
data['value'].max()))

color_bar = ColorBar(color_mapper=mapper,
ticker=BasicTicker(desired_num_ticks=10), label_standoff=12,
border_line_color=None, location=(0,0))
p.add_layout(color_bar, 'right')

```

**BOKEH ASSIGNMENT: (Use Bokeh for the visualization of the given questions)**

## **Bokeh Plot Displaying a Sine Wave**

python

```

from bokeh.plotting import figure, show
from bokeh.io import output_notebook
import numpy as np

output_notebook()

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

p = figure(title="Sine Wave", x_axis_label='X', y_axis_label='Y')
p.line(x, y, legend_label="Sine", line_width=2)

show(p)

```

## **2. Bokeh Scatter Plot with Random Values**

python

```

from bokeh.plotting import figure, show
from bokeh.io import output_notebook
import numpy as np

output_notebook()

np.random.seed(30)
x = np.random.rand(100)
y = np.random.rand(100)
sizes = np.random.randint(10, 100, 100)
colors = np.random.randint(1, 256, 100)

p = figure(title="Scatter Plot", x_axis_label='X', y_axis_label='Y')
p.scatter(x, y, size=sizes, color=[f"#{i:02x}{i:02x}{i:02x}" for i in
colors], alpha=0.6)

show(p)

```

## **3. Bokeh Bar Chart for Fruit Counts**

python

```
from bokeh.plotting import figure, show
from bokeh.io import output_notebook

output_notebook()

fruits = ['Apples', 'Oranges', 'Bananas', 'Pears']
counts = [20, 25, 30, 35]

p = figure(x_range=fruits, title="Fruit Counts", toolbar_location=None,
tools="")
p.vbar(x=fruits, top=counts, width=0.9)

show(p)
```

## 4. Bokeh Histogram to Visualize Data Distribution

python

```
from bokeh.plotting import figure, show
from bokeh.io import output_notebook
import numpy as np

output_notebook()

data_hist = np.random.randn(1000)
hist, edges = np.histogram(data_hist, bins=30)

p = figure(title="Histogram", x_axis_label='Value',
y_axis_label='Frequency')
p.quad(top=hist, bottom=0, left=edges[:-1], right=edges[1:],
fill_color="navy", line_color="white", alpha=0.5)

show(p)
```

## 5. Bokeh Heatmap

python

```
from bokeh.plotting import figure, show
from bokeh.io import output_notebook
from bokeh.transform import linear_cmap
from bokeh.models import ColorBar, BasicTicker, LinearColorMapper
import pandas as pd
import numpy as np

output_notebook()

data_heatmap = np.random.rand(10, 10)
x = np.linspace(0, 1, 10)
y = np.linspace(0, 1, 10)
xx, yy = np.meshgrid(x, y)

data = pd.DataFrame(data_heatmap, columns=x, index=y)
data = data.stack().reset_index()
data.columns = ['x', 'y', 'value']

mapper = LinearColorMapper(palette="Viridis256", low=data['value'].min(),
high=data['value'].max())
```



```

p = figure(title="Heatmap", x_range=(0, 1), y_range=(0, 1),
toolbar_location=None, tools="")
p.rect(x="x", y="y", width=0.1, height=0.1, source=data, line_color=None,
fill_color=linear_cmap('value', 'Viridis256', data['value'].min(),
data['value'].max()))

color_bar = ColorBar(color_mapper=mapper,
ticker=BasicTicker(desired_num_ticks=10), label_standoff=12,
border_line_color=None, location=(0,0))
p.add_layout(color_bar, 'right')

show(p)

```

## Extra: 3D Data Heatmap

python

```

from bokeh.plotting import figure, show
from bokeh.io import output_notebook
from bokeh.models import ColumnDataSource
import numpy as np
import pandas as pd

output_notebook()

x = np.linspace(-5, 5, 100)
y = np.linspace(-5, 5, 100)
x, y = np.meshgrid(x, y)
z = np.sin(np.sqrt(x**2 + y**2))

data = {
    'X': x.flatten(),
    'Y': y.flatten(),
    'Z': z.flatten()
}
df = pd.DataFrame(data)

source = ColumnDataSource(df)

p = figure(title="3D Data Heatmap", tools="hover", tooltips="X: @X, Y: @Y,
Z: @Z")
p.image(image=[df.pivot('Y', 'X', 'Z').values], x=-5, y=-5, dw=10, dh=10,
palette="Spectral11")

```