

Practical

1-How do you create a 2D NumPy array and calculate the sum of each row

Ans-

```
import numpy as np

arr = np.array([[1, 2, 3, 4],
                [5, 6, 7, 8],
                [9, 10, 11, 12]])

print("Original 2D NumPy array:")
print(arr)

row_sums = np.sum(arr, axis=1)

print("\nSum of each row:")
print(row_sums)
```

Original 2D NumPy array:

```
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
```

Sum of each row:

```
[10 26 42]
```

2-Write a Pandas script to find the mean of a specific column in a DataFrame

Ans-

```
import pandas as pd

data = {'Name': ['Alice', 'Bob', 'Charlie', 'David'],
        'Age': [24, 27, 22, 32],
        'Score': [85, 92, 78, 88]}
df = pd.DataFrame(data)

print("Original DataFrame:")
print(df)

column_name = 'Age'
mean_age = df[column_name].mean()

print(f"\nMean of the '{column_name}' column: {mean_age}")

column_name_score = 'Score'
mean_score = df[column_name_score].mean()
print(f"Mean of the '{column_name_score}' column: {mean_score}")
```

Original DataFrame:

	Name	Age	Score
0	Alice	24	85
1	Bob	27	92
2	Charlie	22	78
3	David	32	88

Mean of the 'Age' column: 26.25

Mean of the 'Score' column: 85.75

3-Create a scatter plot using Matplotlib

Ans-

```
import matplotlib.pyplot as plt
import numpy as np

np.random.seed(42)
x = np.random.rand(50) * 10
y = np.random.rand(50) * 10
colors = np.random.rand(50)
sizes = np.random.rand(50) * 500 + 50

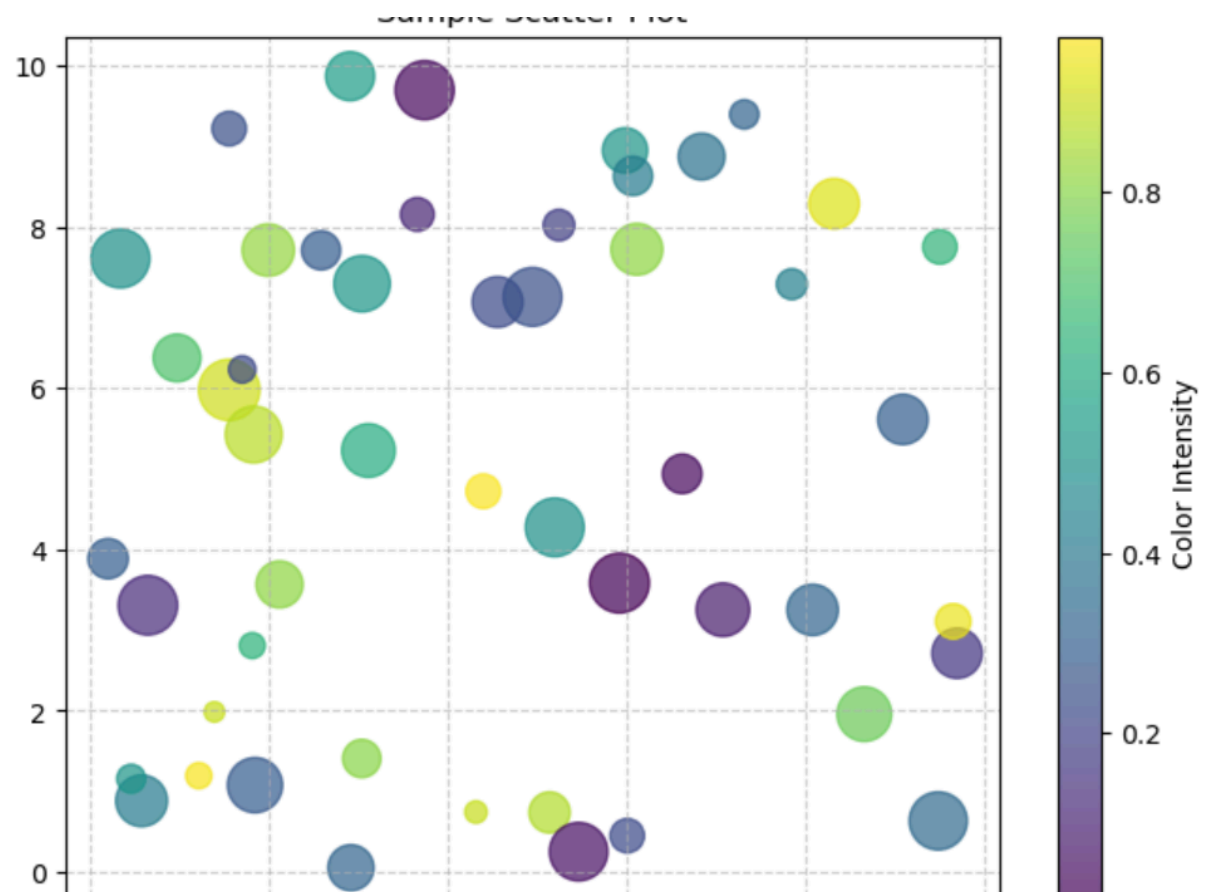
plt.figure(figsize=(8, 6))
plt.scatter(x, y, c=colors, s=sizes, alpha=0.7, cmap='viridis')

plt.xlabel("X-axis Label")
plt.ylabel("Y-axis Label")
plt.title("Sample Scatter Plot")

plt.colorbar(label="Color Intensity")

plt.grid(True, linestyle='--', alpha=0.6)

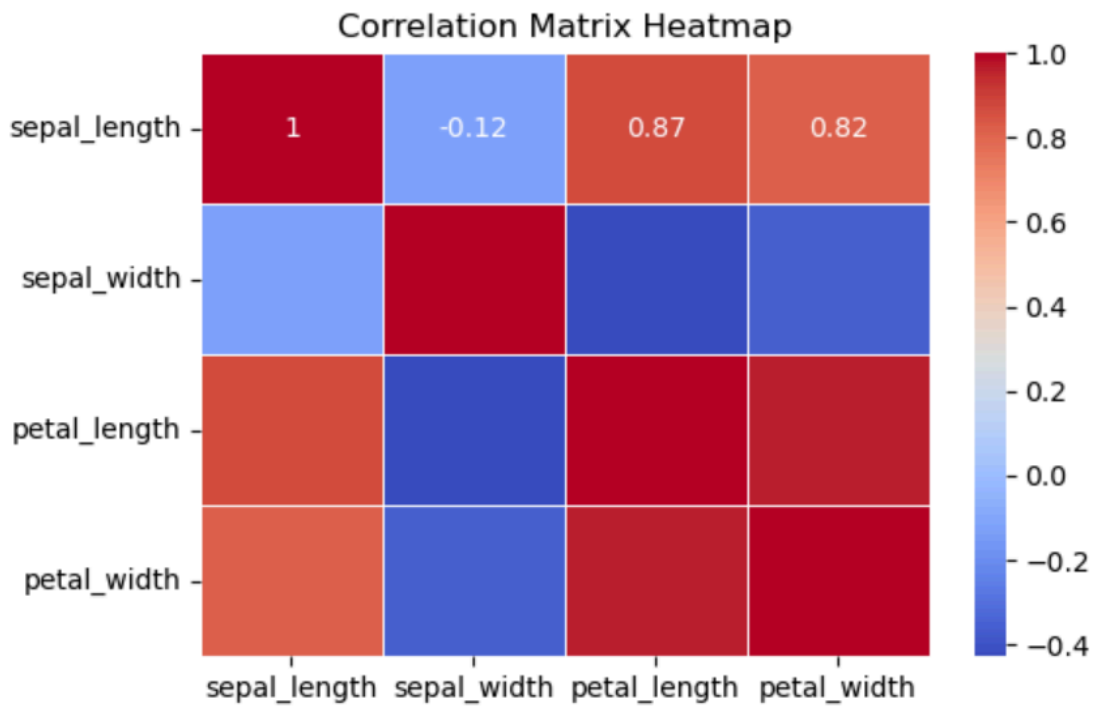
plt.show()
```



4-How do you calculate the correlation matrix using Seaborn and visualize it with a heatmap
Ans-

```
df = sns.load_dataset("iris")
corr_matrix = df.corr(numeric_only=True)

plt.figure(figsize=(8, 6))
sns.heatmap(corr_matrix, annot=True, cmap="coolwarm", linewidths=0.5)
plt.title("Correlation Matrix Heatmap")
plt.show()
```



5-Generate a bar plot using Plotly

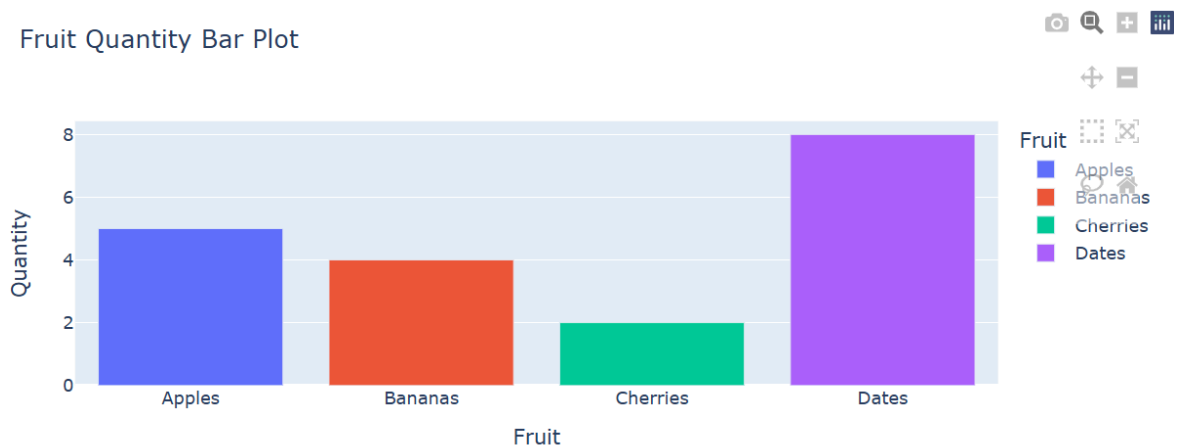
Ans-

```
import plotly.express as px
import pandas as pd

data = {
    'Fruit': ['Apples', 'Bananas', 'Cherries', 'Dates'],
    'Quantity': [5, 4, 2, 8]
}
df = pd.DataFrame(data)

fig = px.bar(df, x='Fruit', y='Quantity', title='Fruit Quantity Bar Plot', color='Fruit')
fig.show()
```

Fruit Quantity Bar Plot



6-Create a DataFrame and add a new column based on an existing column

Ans-

```
import pandas as pd

data = {
    'Name': ['Alice', 'Bob', 'Charlie'],
    'Marks': [85, 42, 77]
}

df = pd.DataFrame(data)

df['Result'] = df['Marks'].apply(lambda x: 'Pass' if x >= 50 else 'Fail')

print(df)
```

	Name	Marks	Result
0	Alice	85	Pass
1	Bob	42	Fail
2	Charlie	77	Pass

7-Write a program to perform element-wise multiplication of two NumPy arrays

Ans-

```
: import numpy as np

array1 = np.array([1, 2, 3, 4])
array2 = np.array([5, 6, 7, 8])

result = array1 * array2

print("Array 1:", array1)
print("Array 2:", array2)
print("Element-wise Multiplication:", result)
```

```
Array 1: [1 2 3 4]
Array 2: [5 6 7 8]
Element-wise Multiplication: [ 5 12 21 32]
```

8-Create a line plot with multiple lines using Matplotlib

Ans-

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4]
y1 = [2, 4, 6, 8]
y2 = [1, 3, 5, 7]
y3 = [5, 7, 4, 6]

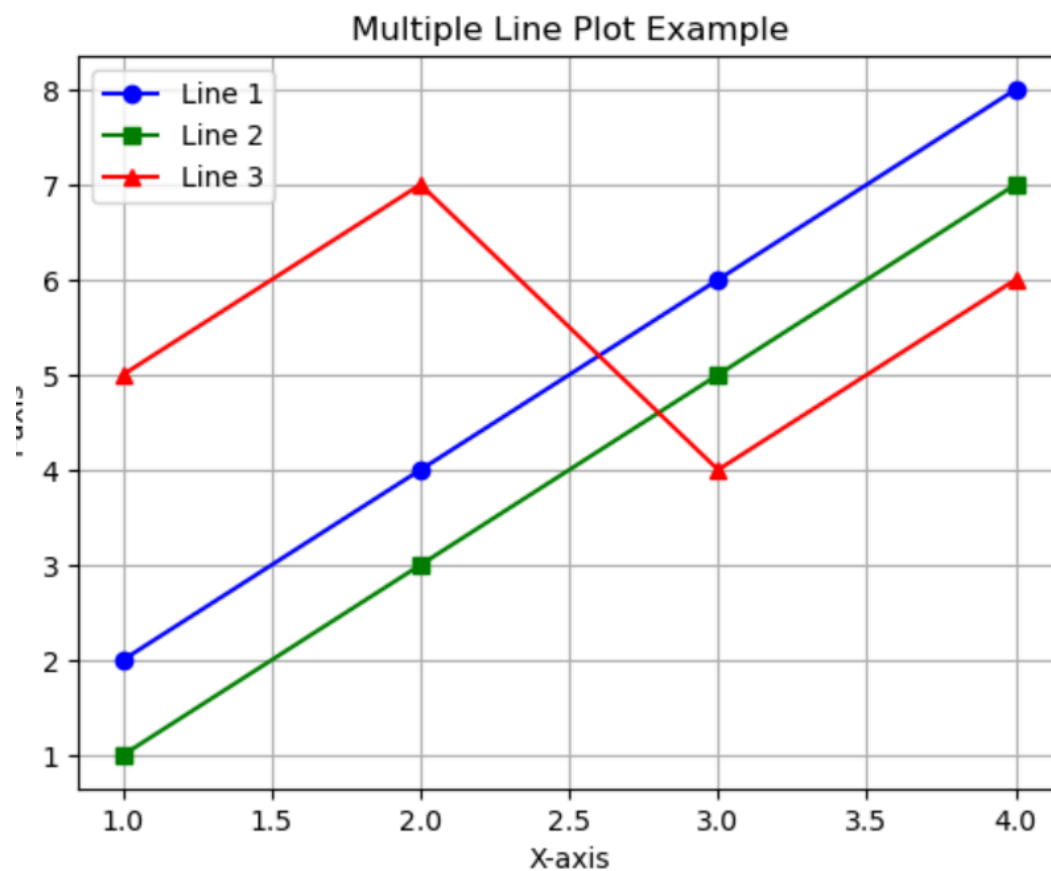
plt.plot(x, y1, label='Line 1', color='blue', marker='o')
plt.plot(x, y2, label='Line 2', color='green', marker='s')
plt.plot(x, y3, label='Line 3', color='red', marker='^')

plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Multiple Line Plot Example')

plt.legend()

plt.grid(True)

plt.show()
```



9- Generate a Pandas DataFrame and filter rows where a column value is greater than a threshold

Ans-

```
import pandas as pd

data = {
    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve'],
    'Score': [85, 42, 90, 67, 76]
}

df = pd.DataFrame(data)

# Define threshold
threshold = 70

filtered_df = df[df['Score'] > threshold]

print("Original DataFrame:")
print(df)

print("\nFiltered DataFrame (Score > 70):")
print(filtered_df)
```

Original DataFrame:

	Name	Score
0	Alice	85
1	Bob	42
2	Charlie	90
3	David	67
4	Eve	76

Filtered DataFrame (Score > 70):

	Name	Score
0	Alice	85
2	Charlie	90
4	Eve	76

10- Perform matrix multiplication using NumPy

Ans-

```
import numpy as np

A = np.array([[1, 2],
              [3, 4]])

B = np.array([[5, 6],
              [7, 8]])

result = np.dot(A, B)

print("Matrix A:")
print(A)

print("\nMatrix B:")
print(B)

print("\nMatrix Multiplication (A × B):")
print(result)
```

Matrix A:

```
[[1 2]
 [3 4]]
```

Matrix B:

```
[[5 6]
 [7 8]]
```

Matrix Multiplication (A × B):

```
[[19 22]
 [43 50]]
```

11-Use Pandas to load a CSV file and display its first 5 rows

Ans-

```
import pandas as pd

file_id = '1p6XFZ3EHZ-XneUFjjwqFucXZTbUZKdvs'
url = f'https://drive.google.com/uc?export=download&id={file_id}'
df = pd.read_csv(url)
print(df.head())
```

	PassengerId	Survived	Pclass	\
0	1	0	3	
1	2	1	1	
2	3	1	3	
3	4	1	1	
4	5	0	3	

	Name	Sex	Age	SibSp	\
0	Braund, Mr. Owen Harris	male	22.0	1	
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	
2	Heikkinen, Miss. Laina	female	26.0	0	
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	
4	Allen, Mr. William Henry	male	35.0	0	

	Parch	Ticket	Fare	Cabin	Embarked
0	0	A/5 21171	7.2500	NaN	S
1	0	PC 17599	71.2833	C85	C
2	0	STON/O2. 3101282	7.9250	NaN	S
3	0	113803	53.1000	C123	S
4	0	373450	8.0500	NaN	S

12-Create a histogram using Seaborn to visualize a distribution

Ans-

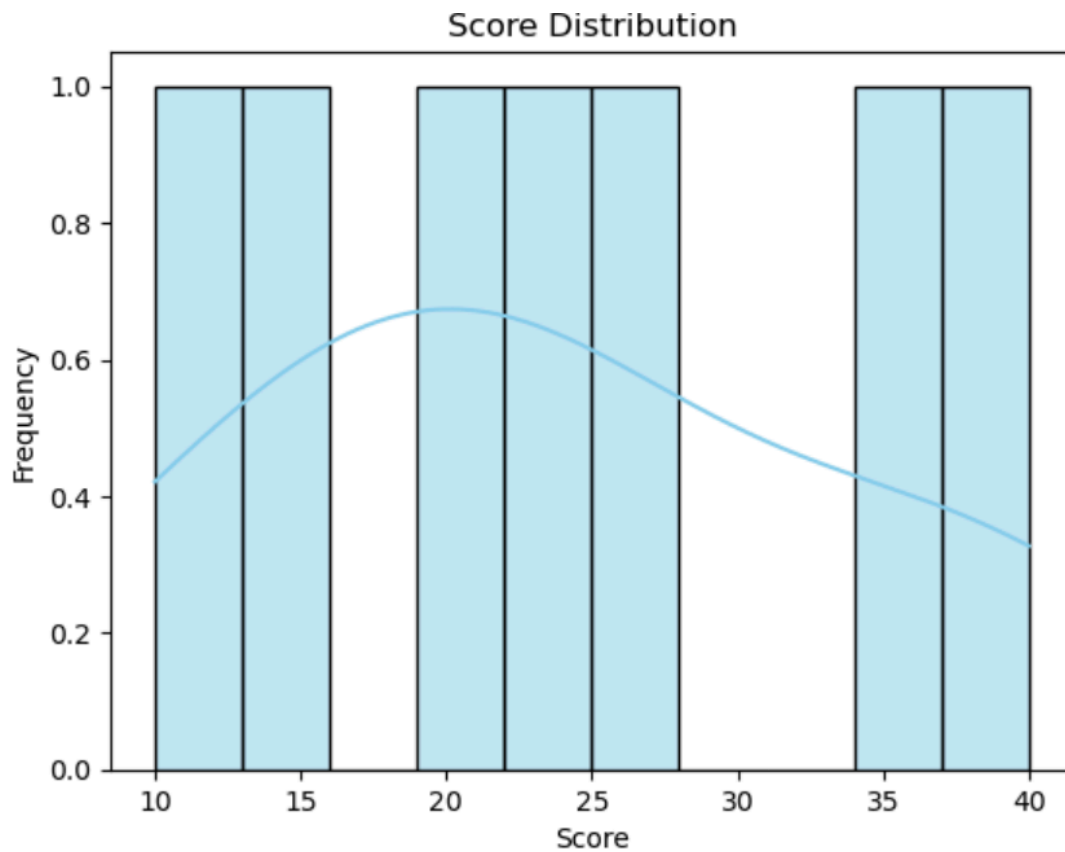
```
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt

data = pd.DataFrame({
    'score': [10,15,20,25,22,35,40]
})

sns.histplot(data['score'], bins=10, kde=True, color='skyblue')

plt.title('Score Distribution')
plt.xlabel('Score')
plt.ylabel('Frequency')

plt.show()
```



13-Create a 3D scatter plot using Plotly.

Ans-

```
import numpy as np
import plotly.graph_objects as go
np.random.seed(42)
x = np.random.rand(100) * 10
y = np.random.rand(100) * 10
z = np.random.rand(100) * 10
fig = go.Figure()

fig.add_trace(go.Scatter3d(
    x=x, y=y, z=z,
    mode='markers',
    marker=dict(size=6, color=z, colorscale='Viridis', opacity=0.8)
))

fig.update_layout(
    title="3D Scatter Plot",
    scene=dict(
        xaxis_title="X Axis",
        yaxis_title="Y Axis",
        zaxis_title="Z Axis"
    )
)
```

3D Scatter Plot

