PCC-DS 391 Lab Assignment 5

Topic: Python Numpy

1. Write a program that uses numpy to add and multiply two 1D arrays and store the result in a third array.

```
import numpy as np
# Define two 1D arrays
array1 = np.array([1, 2, 3, 4, 5])
array2 = np.array([5, 4, 3, 2, 1])

# Add the two arrays
result_array1 = np.add(array1, array2)
# Multiply the two arrays element-wise
result_array2 = np.multiply(array1, array2)

# Display the result
print("Array 1:", array1)
print("Array 2:", array2)
print("Result Array (Array 1 + Array 2):", result_array1)
print("Result Array (Array 1 * Array 2):", result_array2)
```

2. Write a program using numpy package to find the lcm and hcf of two numbers.

```
import numpy as np
# Input two numbers
num1 = 12
num2 = 18

# Find HCF (GCD)
hcf = np.gcd(num1, num2)

# Find LCM
lcm = abs(num1 * num1) // hcf

# Display the results
print("Number 1:", num1)
print("Number 2:", num2)
print("HCF (GCD):", hcf)
print("LCM:", lcm)
```

3. Write a program using numpy to create an array of size 6 where the elements of array are random numbers between 10 to 30.

```
import numpy as np
# Create an array of size 6 with random integers between 10 and 30
random_array = np.random.randint(10, 31, size=6)

# Display the array
```

```
print("Random Array:", random_array)
```

4. Write a program to sort a NumPy array in ascending order and descending order.

```
import numpy as np
arr = np.array([10, 2, 8, 4])
sorted_arr_asc = np.sort(arr)  # ascending order
print("Ascending order", sorted_arr_asc)
sorted_arr_dsc = np.sort(arr)[::-1]  # descending order
print("Descending order", sorted_arr_dsc)
```

5. Write a program that finds the index of a specific element in a NumPy array, index of the maximum element in a NumPy array and index of the minimum element in a NumPy array.

```
import numpy as np
arr = np.array([10, 2, 8, 4])
index = np.where(arr == 8)
max_index = np.argmax(arr)
min_index = np.argmin(arr)
print("Index of 8", index)
print("Index of max element", max_index)
print("Index of min element", min_index)
```

6. Write a program that finds the indices of elements in a NumPy array that satisfy multiple conditions (e.g., elements greater than 2 and less than 5).

```
import numpy as np
arr = np.array([1, 7, 3, 4])
indices = np.where((arr > 2) & (arr < 5))
print("Indices are", indices)
```

7. Write a program to display line graph on a standard dataset.

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

# Load a standard dataset (example: flights dataset from seaborn)
flights = sns.load_dataset("flights")

# Display the first few rows of the dataset
print(flights.head())

# Create a line plot showing the number of passengers over time (per month and year)
plt.figure(figsize=(10, 6))
sns.lineplot(data=flights, x="year", y="passengers", hue="month", palette="tab10")

# Add title and labels
plt.title("Number of Airline Passengers Over Time")
```

```
plt.xlabel("Year")
plt.ylabel("Number of Passengers")

# Display the graph
plt.show()
```

Explanation:

- **Dataset**: The flights dataset from seaborn contains information about the number of passengers flying per month from 1949 to 1960.
- **Line Plot**: sns.lineplot() is used to create a line graph. It plots the number of passengers on the y-axis and the year on the x-axis. The hue parameter distinguishes the months with different colors.
- **Graph Display**: The graph shows how the number of airline passengers changes over time, with one line for each month.

8. Write a program to display scatterplot on a standard dataset.

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

# Load the standard "iris" dataset from seaborn
iris = sns.load_dataset("iris")

# Display the first few rows of the dataset
print(iris.head())

# Create a scatter plot showing sepal length vs. sepal width
plt.figure(figsize=(10, 6))
sns.scatterplot(data=iris, x="sepal_length", y="sepal_width", hue="species", palette="deep", s=100)

# Add title and labels
plt.title("Sepal Length vs. Sepal Width (Iris Dataset)")
plt.xlabel("Sepal Length (cm)")
plt.ylabel("Sepal Width (cm)")

# Display the scatter plot
plt.show()
```

Explanation:

- **Dataset**: The iris dataset is a popular dataset that contains measurements of iris flowers, including sepal length, sepal width, petal length, petal width, and species.
- **Scatter Plot**: sns.scatterplot() is used to create the scatter plot. The x and y axes represent the sepal length and sepal width, respectively. The hue parameter colors the points based on the species of the flower.
- **Customization**: The palette sets the color scheme, and s=100 makes the points larger for better visibility.

PCC-DS 391 Lab Assignment 5

9. Write a program to display bar graphs on a standard dataset.

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

# Load the standard "tips" dataset from seaborn
tips = sns.load_dataset("tips")

# Display the first few rows of the dataset
print(tips.head())

# Create a bar plot showing the total bill for each day
plt.figure(figsize=(10, 6))
sns.barplot(data=tips, x="day", y="total_bill", estimator=sum, palette="muted")

# Add title and labels
plt.title("Total Bill per Day (Tips Dataset)")
plt.xlabel("Day")
plt.ylabel("Total Bill (Sum)")

# Display the bar plot
plt.show()
```

Explanation:

- **Dataset**: The tips dataset from seaborn contains information about the total bill, tips, gender, day, time, and more, collected from a restaurant.
- **Bar Plot**: sns.barplot() creates the bar graph. The x axis shows the days of the week, and the y axis shows the total bill, which is summed for each day using the estimator=sum argument.
- **Customization**: The palette argument sets the color scheme of the bars. You can adjust it as needed.

10. Write a program to display histogram on a standard dataset.

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

# Load the standard "titanic" dataset from seaborn
titanic = sns.load_dataset("titanic")

# Display the first few rows of the dataset
print(titanic.head())

# Create a histogram showing the distribution of passengers' ages
plt.figure(figsize=(10, 6))
sns.histplot(data=titanic, x="age", bins=30, kde=True, color="blue")

# Add title and labels
plt.title("Age Distribution of Titanic Passengers")
```

```
plt.xlabel("Age")
plt.ylabel("Frequency")

# Display the histogram
plt.show()
```

Explanation:

- **Dataset**: The titanic dataset from seaborn contains information about Titanic passengers, including their age, class, fare, gender, and survival status.
- **Histogram**: sns.histplot() creates a histogram to display the distribution of the "age" column. The bins=30 argument specifies the number of bins, and kde=True adds a Kernel Density Estimate (a smooth curve) over the histogram for a better visualization of the age distribution.
- **Customization**: The color argument sets the color of the histogram bars.