

Lab – 21

Student Name: Navneet P

Student ID : AF0411619

Topic: Data Visualization

About Matplotlib: Matplotlib is a powerful plotting library in Python used for creating static, animated, and interactive visualizations. Matplotlib's primary purpose is to provide users with the tools and functionality to represent data graphically, making it easier to analyze and understand. It was originally developed by John D. Hunter in 2003 and is now maintained by a large community of developers.

Installation: pip install matplotlib

Version: import matplotlib matplotlib.__version__ =====> (3.9.2)

Functions/submodules used in this Assignment:

1. **Pyplot:** module in Matplotlib for creating static, animated, and interactive visualizations.
2. **Matplotlib.pyplot.plot():** Creates a plot line graph
3. **Matplotlib.pyplot.bar():** Creates a bar chart
4. **Matplotlib.pyplot.subplots():** Creates multiple subplots in a single figure for complex layouts.
5. **Set():** A direct way of adding labels and title in just a single function call
6. **Matplotlib.pyplot.show():** Adds a label to the x-axis of the plot.
7. **Matplotlib.pyplot.xlabel():** Adds a label to the x-axis of the plot.
8. **Matplotlib.pyplot.ylabel():** Adds a label to the y-axis of the plot.
9. **Matplotlib.pyplot.title():** Adds a title to the plot.
10. **Grid():** Adds grid lines to the plot for easier reading of data points.
11. **Matplotlib.pyplot.bar_label():** Adds labels to the bars in a bar chart to display values on each bar.
12. **Pandas.read_excel():** reads the path specified excel workbook

Example 1: Create a basic graph using list data and the matplotlib library.

Solution:

```
Example 1

# Importing the matplotlib.pyplot module for plotting
import matplotlib.pyplot as plt

# Data for the x and y axes
x = [1, 2, 3, 4, 5] # x-coordinates
y = [10, 20, 40, 50, 70] # y-coordinates

# Plotting the x and y values as a line graph
plt.plot(x, y)

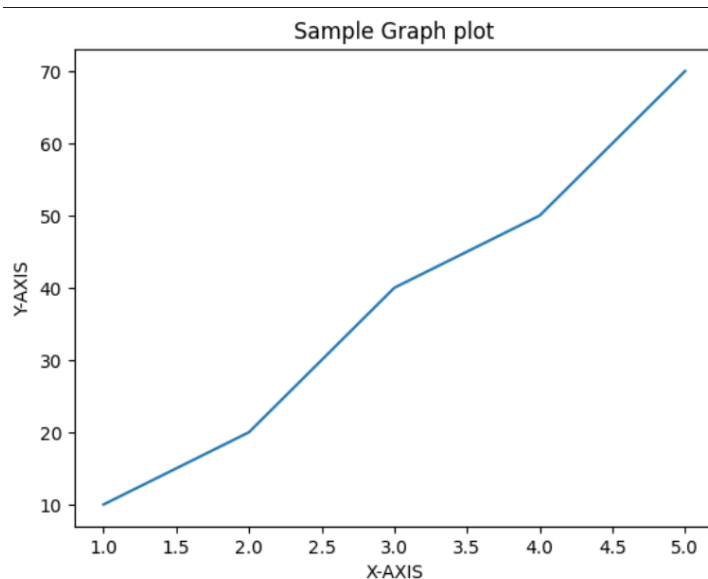
# Adding a label to the x-axis
plt.xlabel("X-AXIS")

# Adding a label to the y-axis
plt.ylabel("Y-AXIS")

# Adding a title to the plot
plt.title("Sample Graph plot")

# Displaying the plot
plt.show()
```

Output:



Example 2: Create a simple sine wave based graph using matplotlib.

Solution:

```
Example 2

# Importing the necessary modules for plotting and numerical operations
import matplotlib.pyplot as plt
import numpy as np

# Creating an array of time values from 0 to 2 with a step of 0.01
t = np.arange(0.0, 2.0, 0.01)

# Creating the sine wave data for the voltage values
s = 1 + np.sin(2 * np.pi * t)

# Creating a figure and an axes object using subplots (for more complex layouts)
fig, ax = plt.subplots()

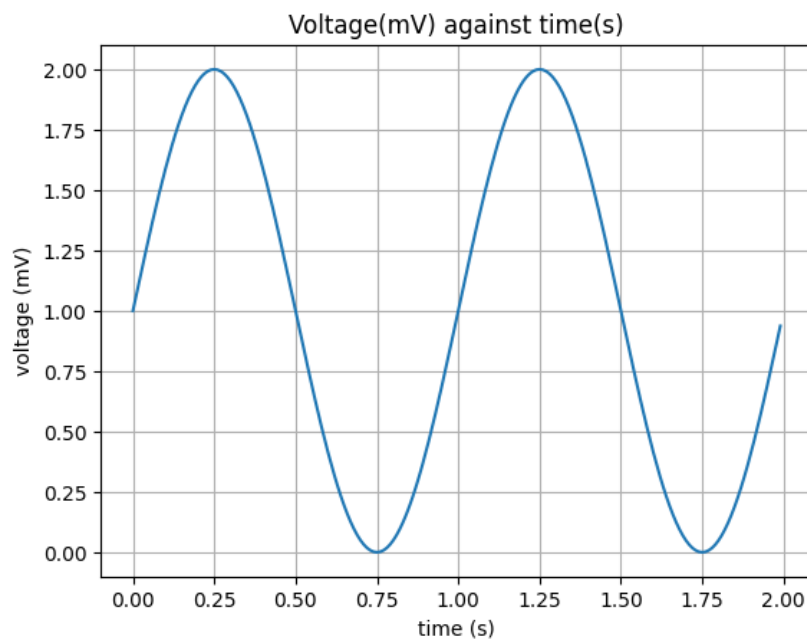
# Plotting the time (t) against the sine wave (s) on the axes
ax.plot(t, s)

# Setting the labels for the x and y axes and the title of the graph
ax.set(xlabel='time (s)', ylabel='voltage (mV)',
       title='Voltage(mV) against time(s)')

# Adding a grid to the plot for better readability
ax.grid()

# Displaying the plot
plt.show()
```

Output:



Q1. Create sample line chart using matplotlib

Solution:

```
Question 1

# Importing the necessary modules for plotting and numerical operations
import matplotlib.pyplot as plt
import numpy as np

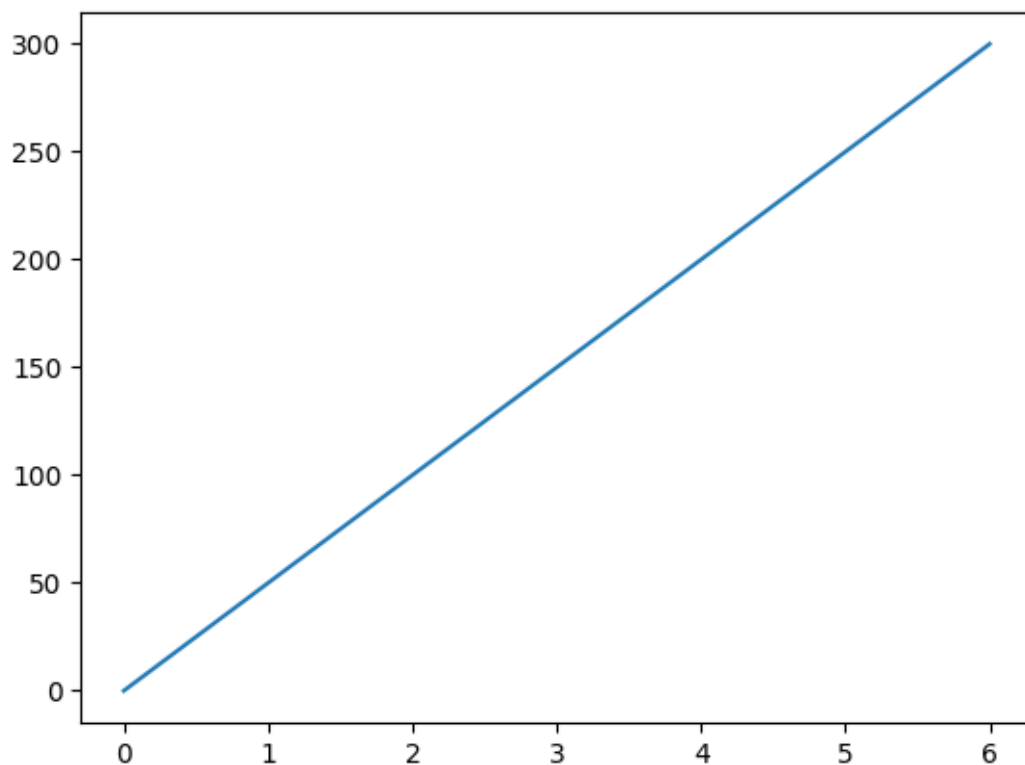
# Creating an array for x-axis values (from 0 to 6)
x = np.array([0, 6])

# Creating an array for y-axis values (from 0 to 300)
y = np.array([0, 300])

# Plotting the x and y values as a line graph
plt.plot(x, y)

# Displaying the plot
plt.show()
```

Output:



Q2. Create Bar Chart using Sample data of employee salary report with different bar colors, data labels.

Solution:

Approach would be to first create a sample dataset preferably in excel with different columns such as Empid, Empname and their salary. In this instance, we'll be formulating a bar chart for just 5 people in the organisation for observational purposes.

Below is the dataset used for the bar chart. We'll be using pandas library to import the raw data into the code and work with it.

Sample Dataset:

Empid	Empname	salary
100	Niko	30000
101	Derby	40000
102	Troy	50000
103	Fraser	30000
104	Raju	30000

The read_excel() functions returns a dictionary with the column names as keys and the data in the form of lists as its values. We then store these lists by using their specified keys (Empid, Empname, Salary) into a numpy array and convert them into x and y values for the graph. For the actual graph we would use the matplotlib library and a bunch of its functions.

Below is the code doing the very task mentioned above:

Code:

```
Question 2

# Importing the necessary libraries for data manipulation and plotting
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Reading data from an Excel file into a pandas DataFrame
data = pd.read_excel(r'C:\Python Programs\matplotlib\sales_data.xlsx')

# Defining a list of colors to use in the bar plot
colors = ["red", "blue", "green", "yellow", "orange"]

# Extracting the 'Empid' column from the DataFrame
empid = data['Empid']
# Converting the 'Empid' column into a NumPy array (though not used in the plot)
emparray = np.array(empid)

# Extracting the 'Empname' column for employee names
empname = data['Empname']
# Converting the employee names into a NumPy array for plotting
empnamearray = np.array(empname)

# Extracting the 'salary' column for employee salaries
salary = data['salary']
# Converting the salary data into a NumPy array for plotting
empsalaryarray = np.array(salary)

# Creating a bar plot with employee names on the x-axis and salaries on the y-axis
barplot = plt.bar(empnamearray, empsalaryarray, color=colors)

# Adding labels to the bars, showing the salary in the center of each bar
plt.bar_label(barplot, labels=empsalaryarray, label_type="center")

# Setting the x-axis label
plt.xlabel('Names')

# Setting the y-axis label
plt.ylabel('Salary')

# Setting the title of the plot
plt.title("Employee statistics")

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

Output:

