Python Programming

Lab:- 21(Data Visualization)

Student Id:- AF0417098

Student Name:- Ankush

Matplotlib:- **It is a widely-used Python library for creating static, animated, and interactive visualizations in Python. It provides a flexible framework for creating a wide range of plots and charts.**

Here are some key points about data visualization using Matplotlib in Python:-

1. **Basic Plotting Functions:-**

- **Line Plot**: Use **plt.plot()** to create line plots, which are useful for showing trends over time.
- **Bar Plot**: Use **plt.bar()** to create bar charts for categorical data comparison.
- **Histogram**: Use **plt.hist()** to visualize the distribution of numerical data.
- **Scatter Plot**: Use **plt.scatter()** to show the relationship between two numerical variables.

2. **Customization Options:-**

- **Labels and Titles:** Use **plt.xlabel(), plt.ylabel(),** and **plt.title()** to add axis labels and titles to your plots.
- **Colors and Styles:** Customize the color and style of lines and markers using parameters like color, linestyle, and marker**.**
- **Legends:** Use **plt.legend()** to add legends, helping viewers understand the elements represented in the plot.

3. **Subplots:-**

- Use plt.subplot() to create multiple plots in a single figure, allowing for better comparison of different datasets or visualizations.

4. **Data Annotation:-**

- You can annotate points on your plots using plt.text() or plt.annotate() which helps highlight specific data points or trends.

Assignment Questions:-

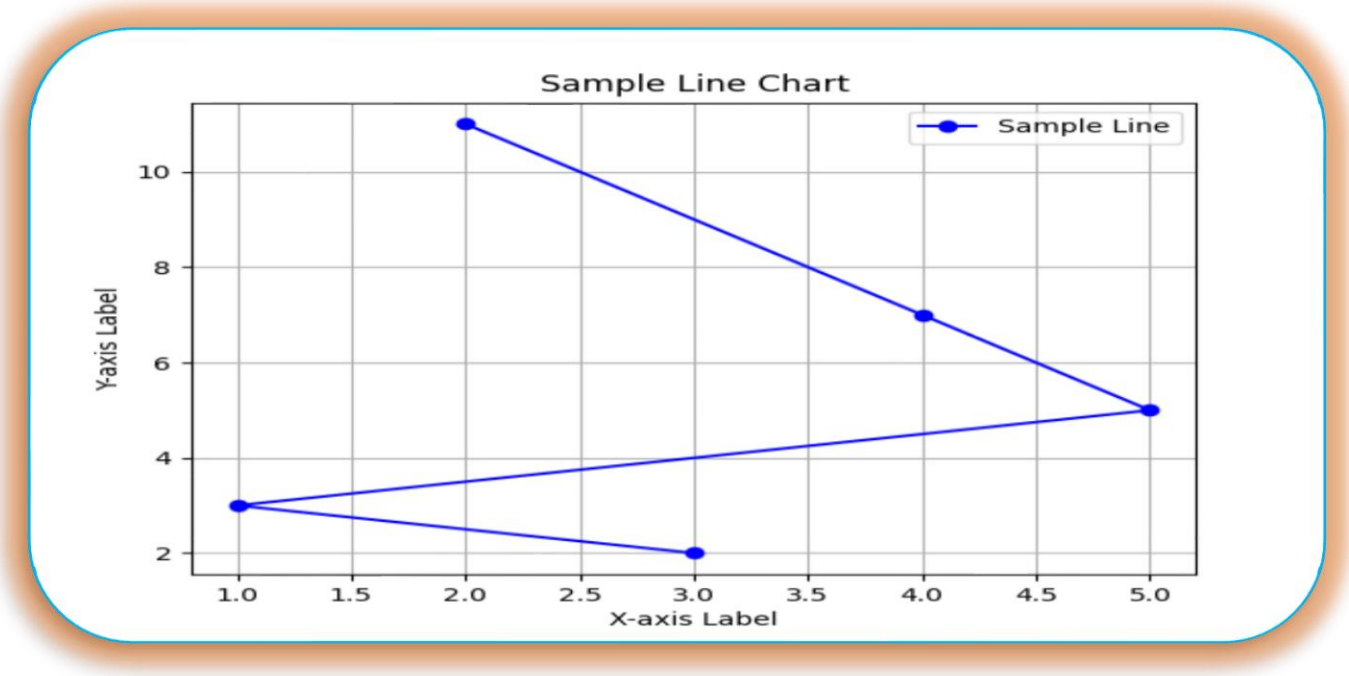
Ques1:- Create sample line chart using matplotlib.

Program:-

lab21.py > ...

```python
import matplotlib.pyplot as plt
import pandas as pd

# Sample data
x = [3, 1, 5, 4, 2]  # X-axis values
y = [2, 3, 5, 7, 11]  # Y-axis values

# Create the line chart
plt.plot(x, y, marker='o', linestyle='-', color='b', label='Sample Line')

# Adding title and labels
plt.title('Sample Line Chart')
plt.xlabel('X-axis Label')

# Adding a legend
plt.legend()

# Display the chart
plt.grid(True)
plt.show()

# x and y: Lists representing the data points for axes.
# plt.plot(...): Plots the data as a line chart, won the data points.
# plt.title(...): Adds a title to the chart.
# plt.xlabel(...) and plt.ylabel(...): Labels for axes.
# plt.legend(): Displays a legend for the plotted line.
# plt.grid(True): Adds a grid to the chartreadability.
# plt.show(): Displays the chart.
# plt.ylabel('Y-axis Label')
```

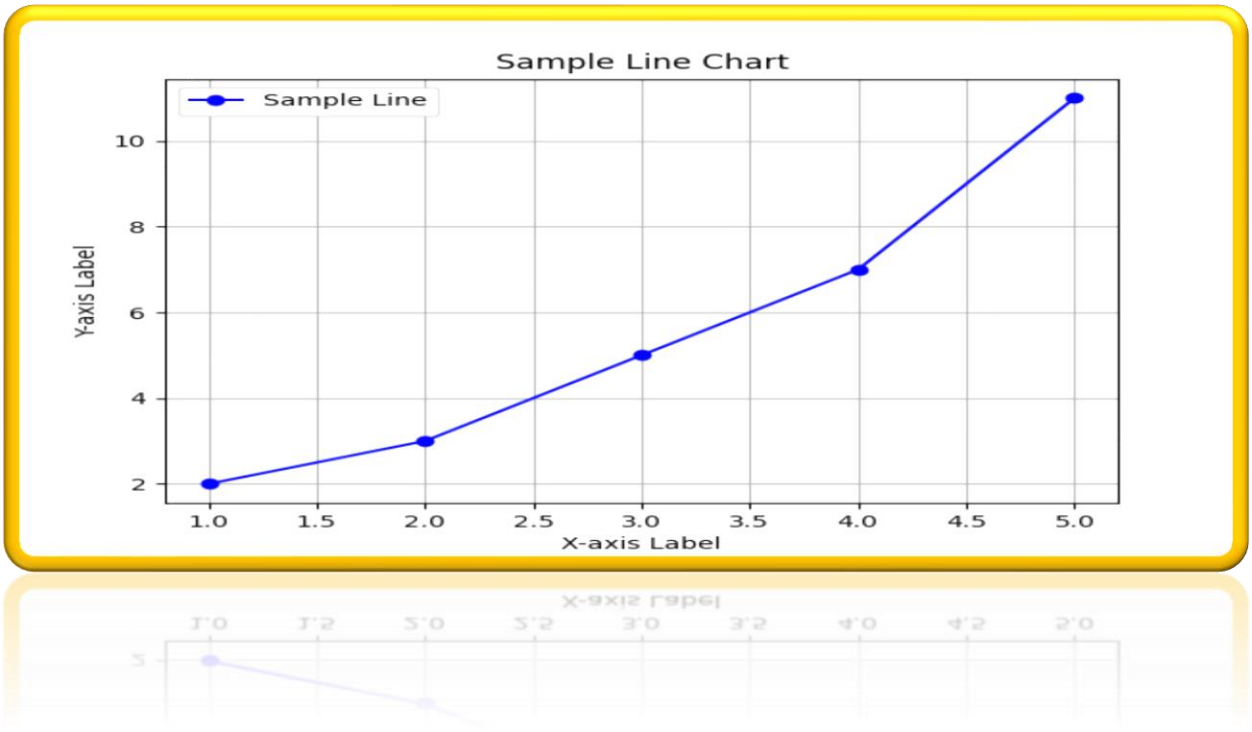
Output:-



#Another one example of this question:-

```python
import matplotlib.pyplot as plt
import pandas as pd

# Sample data
x = [1, 2, 3, 4, 5]  # X-axis values
y = [2, 3, 5, 7, 11]  # Y-axis values

# Create the line chart
plt.plot(x, y, marker='o', linestyle='-', color='b', label='Sample Line')

# Adding title and labels
plt.title('Sample Line Chart')
plt.xlabel('X-axis Label')

# Adding a legend
plt.legend()

# Display the chart
plt.grid(True)
plt.show()

# x and y: Lists representing the data points for axes.
# plt.plot(...): Plots the data as a line chart, won the data points.
# plt.title(...): Adds a title to the chart.
# plt.xlabel(...) and plt.ylabel(...): Labels for axes.
# plt.legend(): Displays a legend for the plotted line.
# plt.grid(True): Adds a grid to the chartreadability.
# plt.show(): Displays the chart.
# plt.ylabel('Y-axis Label')
```

Output:-


Sample Line Chart
Sample Line
Y-axis Label
X-axis Label
10
8
6
4
2
1.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

Ques 2.

Create Bar Chart using Sample data of employee salary report with different bar colors, data labels. #file name is Book2.xlsx

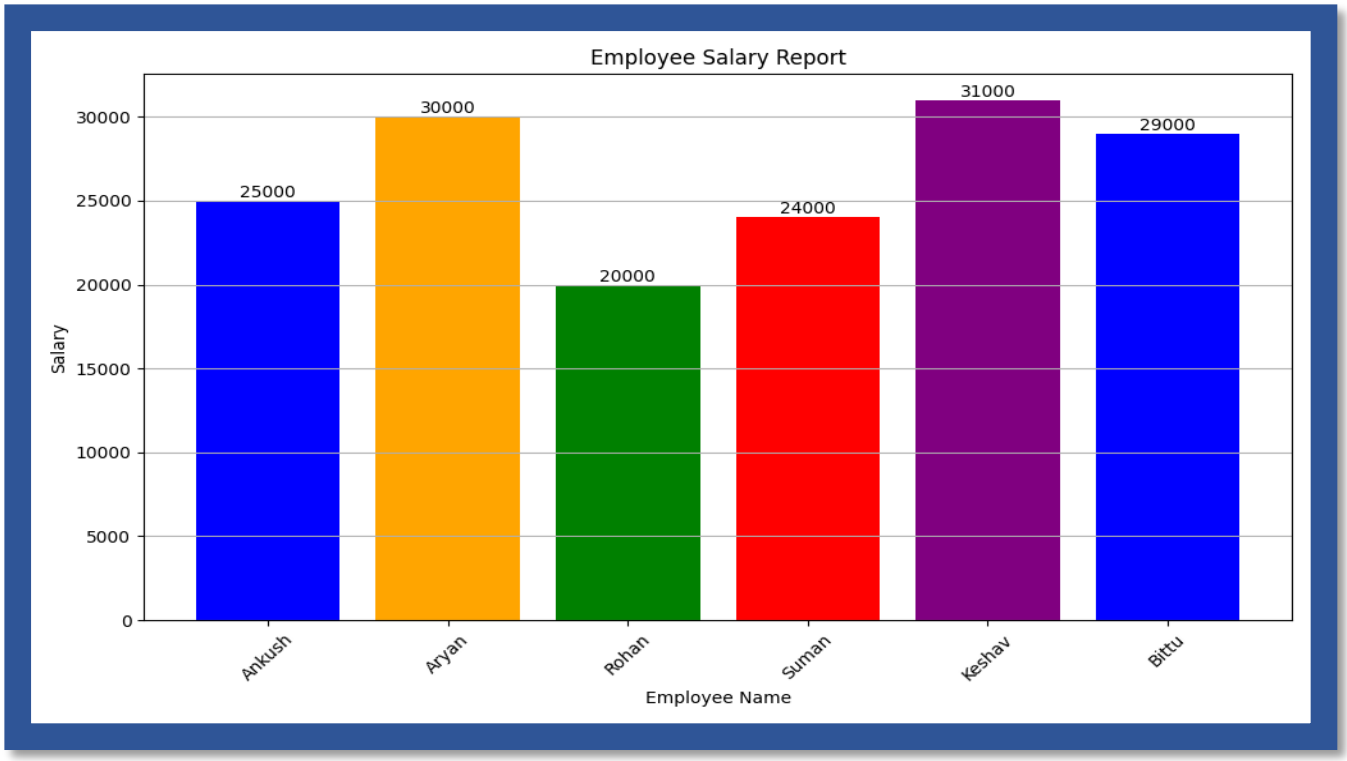
Program:-

lab21.py ● Book2.xlsx

lab21.py > ...

```
30
31  import matplotlib.pyplot as plt
32  import pandas as pd
33
34  # Load the data from the Excel file
35  data = pd.read_excel('Book2.xlsx')
36
37  # Extract employee names and salaries
38  employee_names = data['EmpName']
39  salaries = data['Salary']
40
41  # Create a bar chart
42  plt.figure(figsize=(10, 6))
43  bars = plt.bar(employee_names, salaries, color=['blue', 'orange', 'green', 'red', 'purple'])
44
45  # Add data labels directly above each bar
46  plt.bar_label(bars)
47
48  # Adding title and labels
49  plt.title('Employee Salary Report')
50  plt.xlabel('Employee Name')
51  plt.ylabel('Salary')
52  plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
53
54  # Display the chart
55  plt.grid(axis='y')
56  plt.tight_layout()  # Adjust layout to make room for labels
57  plt.show()
58
59
```

Output:-


Employee Salary Report
Salary
Employee Name
0
5000
10000
15000
20000
25000
30000
25000
30000
20000
24000
31000
29000
Ankush
Aryan
Rohan
Suman
Keshav
Bittu