# Lab 5: Matplotlib

## Exercise 1: Basic Line Plot

1. **Objective**: Practice creating a simple line plot.
2. **Dataset**:

```
x = [0, 1, 2, 3, 4, 5, 6]
y = [0, 1, 4, 9, 16, 25, 36]
```

3. **Instructions**:
   - Use the `x` and `y` lists above.
   - Plot `x` vs `y`.
   - Add labels for the x-axis and y-axis.
   - Add a title to the plot.
   - Show the plot.

## Exercise 2: Customizing Plots

1. **Objective**: Customize the appearance of a plot.
2. **Dataset**: Same as Exercise 1.
3. **Instructions**:
   - Use the same `x` and `y` data as in Exercise 1.
   - Change the color and style of the line (e.g., use a dashed line and green color).
   - Add markers (e.g., circles) to the data points.
   - Add a grid to the plot.

## Exercise 3: Bar Plot

1. **Objective**: Create a bar plot to visualize categories and values.

2. **Dataset**:

   ```
   categories = ['Apples', 'Bananas', 'Cherries', 'Dates', 'E
   lderberries']
   values = [10, 15, 7, 12, 5]
   ```

3. **Instructions**:
   - Use the `categories` and `values` lists above.
   - Plot a bar chart using the data.
   - Add labels for the x-axis and y-axis.
   - Add a title to the plot.

## Exercise 4: Histogram

1. **Objective**: Visualize data distribution using a histogram.
2. **Dataset**:

   ```
   import numpy as np
   data = np.random.normal(0, 1, 500)  # 500 data points from
   a normal distribution
   ```

3. **Instructions**:
   - Use the `data` array generated above.
   - Plot a histogram of the data with at least 20 bins.
   - Customize the appearance (e.g., add an edge color).
   - Add labels for the x-axis and y-axis.
   - Add a title to the plot.

## Exercise 5: Scatter Plot

1. **Objective**: Create a scatter plot to visualize relationships between data.
2. **Dataset**:

```
import numpy as np
x = np.random.rand(50)  # 50 random x-values between 0 and
1
y = np.random.rand(50)  # 50 random y-values between 0 and
1
```

3. **Instructions**:
   - Use the `x` and `y` arrays generated above.
   - Plot a scatter plot using the data.
   - Add labels for the x-axis and y-axis.
   - Add a title to the plot.

## Exercise 6: Subplots

1. **Objective**: Practice creating multiple plots within a single figure using subplots.
2. **Dataset**:
   - **Line Plot**: `x = [1, 2, 3, 4, 5]`, `y = [1, 4, 9, 16, 25]`
   - **Bar Plot**: `categories = ['A', 'B', 'C', 'D', 'E']`, `values = [5, 7, 3, 8, 6]`
   - **Histogram**: `data = np.random.randn(1000)` (1000 random values)
   - **Scatter Plot**: `x_scatter = np.random.rand(50)`, `y_scatter = np.random.rand(50)`
3. **Instructions**:
   - Use the datasets above for each plot type.
   - Create a figure with 2×2 subplots.
   - Plot a different type of plot in each subplot.
   - Add titles to each subplot.

## Exercise 7: Pie Chart

1. **Objective**: Visualize proportions using a pie chart.

2. **Dataset**:

```
categories = ['Marketing', 'Development', 'Sales', 'Suppor
t']
values = [20, 35, 25, 20]
```

3. **Instructions**:
   - Use the `categories` and `values` lists above.
   - Plot a pie chart using the data.
   - Add a title to the chart.
   - Optionally, add a legend.

## Exercise 8: Stacked Bar Plot

1. **Objective**: Create a stacked bar plot to visualize the composition of categories.
2. **Dataset**:

```
categories = ['Group 1', 'Group 2', 'Group 3']
value1 = [5, 7, 3]
value2 = [6, 8, 4]
value3 = [4, 3, 5]
```

3. **Instructions**:
   - Use the `categories`, `value1`, `value2`, and `value3` lists above.
   - Create a stacked bar plot where each category has the three different values stacked.
   - Add labels for the x-axis and y-axis.
   - Add a legend to indicate the different parts of the stack.

## Exercise 9: Box Plot

1. **Objective**: Visualize the spread and distribution of a dataset using a box plot.

2. **Dataset**:

```
import numpy as np
dataset1 = np.random.normal(60, 10, 100)  # 100 data point
s around mean 60
dataset2 = np.random.normal(70, 15, 100)  # 100 data point
s around mean 70
dataset3 = np.random.normal(80, 5, 100)   # 100 data point
s around mean 80
```

3. **Instructions**:
    - Use `dataset1`, `dataset2`, and `dataset3` above.
    - Plot a box plot for each dataset.
    - Add labels and a title.

## Exercise 10: Line Plot with Annotations

1. **Objective**: Practice adding annotations to plots.
2. **Dataset**:

```
x = range(0, 20)
y = [value ** 2 for value in x]
```

3. **Instructions**:
    - Use the `x` and `y` values above.
    - Create a line plot.
    - Annotate the highest and lowest points on the plot.
    - Add labels for the x-axis and y-axis, and a title.

**Submission Instructions:**

For each exercise, create a separate Python file (e.g., ex1.py, ex2.py, etc) containing your code and plot for that specific problem. Upload all the Python files as a single submission once you have completed all the exercises.