

plotting_beginner_fun

May 28, 2024

0.0.1 Exercise 1: Basic Line Plot

Objective: Create a basic line plot to visualize the trend of a dataset.

Instructions:

1. Generate a list of numbers from 1 to 100 for the x-axis using `range()` or `numpy.arange()`.
2. Create a corresponding list of values for the y-axis by applying a mathematical function (e.g., $y = x^2$) to the x-axis values.
3. Use `matplotlib.pyplot` to plot the x-axis and y-axis data.
4. Add a title to the plot as “Basic Line Plot”.
5. Label the x-axis as “X-Values” and the y-axis as “Y-Values”.
6. Display the plot using `plt.show()`.

0.0.2 Exercise 2: Bar Chart

Objective: Create a bar chart to compare different categories of data.

Instructions:

1. Create a list of categories (e.g., ‘Category A’, ‘Category B’, ‘Category C’).
2. Assign a list of values corresponding to each category.
3. Plot a bar chart using `plt.bar()` with the categories and their values.
4. Add a title to the chart as “Category Value Comparison”.
5. Label the x-axis with the category names and the y-axis as “Values”.
6. Display the plot.

0.0.3 Exercise 3: Scatter Plot

Objective: Create a scatter plot to examine the relationship between two variables.

Instructions:

1. Generate two lists of random numbers of the same length for the x-axis and y-axis using `numpy.random.rand()` or `numpy.random.randint()`.
2. Plot a scatter plot using `plt.scatter()` with the generated data.
3. Add a title to the plot as “Scatter Plot of Random Data”.
4. Label the x-axis as “Variable X” and the y-axis as “Variable Y”.
5. Display the plot.

0.0.4 Exercise 4: Multiple Subplots

Objective: Create a figure with multiple subplots showing different types of plots.

Instructions:

1. Create a figure using `plt.figure()` and add multiple subplots using `plt.subplot()`.
2. In the first subplot, plot a sine wave using `numpy.sin()`.
3. In the second subplot, plot a cosine wave using `numpy.cos()`.
4. In the third subplot, plot a line plot using a linear function.
5. In the fourth subplot, plot a scatter plot with random data.
6. Add titles to each subplot indicating the type of plot.
7. Label the axes appropriately for each subplot.
8. Display the figure with all subplots.

These exercises will help students practice creating different types of plots and become familiar with the basic functionalities of Matplotlib.