



COMP 1023 Introduction to Python Programming
Exercises on Matplotlib
COMP 1023 Teaching Team

Department of Computer Science & Engineering
The Hong Kong University of Science and Technology, Hong Kong SAR, China



Exercise 1

You are required to create a simple line plot using Matplotlib.

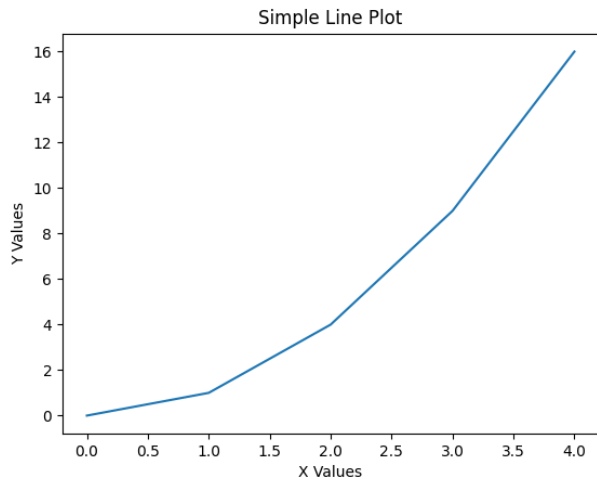
You need to perform the following tasks:

1. Import the Matplotlib library using the alias `plt`.
2. Create two lists:
 - `x = [0, 1, 2, 3, 4]`
 - `y = [0, 1, 4, 9, 16]`
3. Use `plt.plot()` to create a line plot with the given data.
4. Add a title "**Simple Line Plot**" and label the x-axis as "**X Values**" and the y-axis as "**Y Values**".
5. Display the plot using `plt.show()`.

Requirements:

- Use `import matplotlib.pyplot as plt` to import the library.
- Use `plt.plot()` to create the line plot.
- Use `plt.title()`, `plt.xlabel()`, and `plt.ylabel()` to add titles and labels.
- Use `plt.show()` to display the plot.

Expected output:



The following functions will be useful for this exercise:

- `plot()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.plot.html
- `title()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.title.html
- `xlabel()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.xlabel.html
- `ylabel()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.ylabel.html
- `show()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.show.html

Exercise 1 Solutions

```
import matplotlib.pyplot as plt
```

```
# Create the data
```

```
x = [0, 1, 2, 3, 4]
```

```
y = [0, 1, 4, 9, 16]
```

```
# Create the line plot
```

```
plt.plot(x, y)
```

```
# Add title and labels
```

```
plt.title("Simple Line Plot")
```

```
plt.xlabel("X Values")
```

```
plt.ylabel("Y Values")
```

```
# Display the plot
```

```
plt.show()
```

Exercise 2

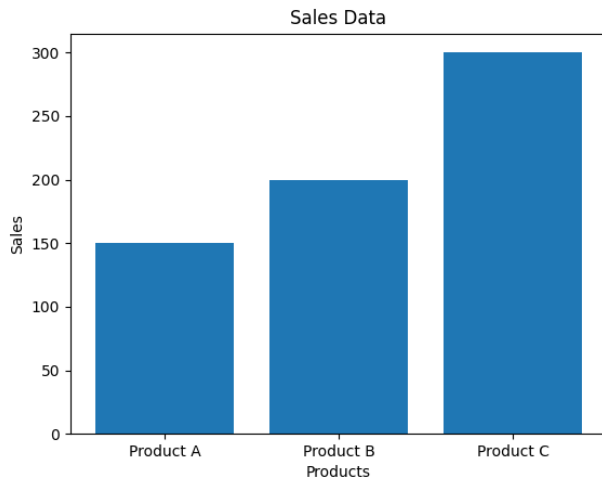
You need to create a bar chart to visualize the sales data of three products. You need to perform the following tasks:

1. Create a list of product names: `products = ['Product A', 'Product B', 'Product C']`.
2. Create a corresponding list of sales figures: `sales = [150, 200, 300]`.
3. Use `plt.bar()` to create a vertical bar chart.
4. Add a title "**Sales Data**" and label the x-axis as "**Products**" and the y-axis as "**Sales**".
5. Display the bar chart using `plt.show()`.

Requirements:

- Use `plt.bar()` to create the bar chart.
- Use `plt.title()`, `plt.xlabel()`, and `plt.ylabel()` to add titles and labels.
- Use `plt.show()` to display the chart.

Expected output:



The following functions will be useful for this exercise:

- `bar()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.bar.html
- `title()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.title.html
- `xlabel()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.xlabel.html
- `ylabel()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.ylabel.html
- `show()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.show.html

Exercise 2 Solutions

```
import matplotlib.pyplot as plt

# Create the data
products = ['Product A', 'Product B', 'Product C']
sales = [150, 200, 300]

# Create the bar chart
plt.bar(products, sales)

# Add title and labels
plt.title("Sales Data")
plt.xlabel("Products")
plt.ylabel("Sales")

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

Exercise 3

You are tasked with creating a scatter plot to visualize the relationship between two variables.

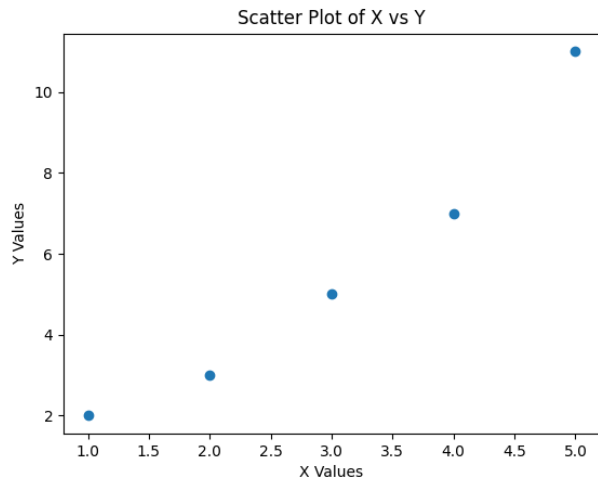
You need to perform the following tasks:

1. Create two lists:
 - `x = [1, 2, 3, 4, 5]`
 - `y = [2, 3, 5, 7, 11]`
2. Use `plt.scatter()` to create a scatter plot with the given data.
3. Add a title "**Scatter Plot of X vs Y**" and label the x-axis as "**X Values**" and the y-axis as "**Y Values**".
4. Display the scatter plot using `plt.show()`.

Requirements:

- Use `plt.scatter()` to create the scatter plot.
- Use `plt.title()`, `plt.xlabel()`, and `plt.ylabel()` to add titles and labels.
- Use `plt.show()` to display the plot.

Expected output:



The following functions will be useful for this exercise:

- `scatter()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.scatter.html
- `title()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.title.html
- `xlabel()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.xlabel.html
- `ylabel()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.ylabel.html
- `show()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.show.html

Exercise 3 Solutions

```
import matplotlib.pyplot as plt

# Create the data
x = [1, 2, 3, 4, 5]
y = [2, 3, 5, 7, 11]

# Create the scatter plot
plt.scatter(x, y)

# Add title and labels
plt.title("Scatter Plot of X vs Y")
plt.xlabel("X Values")
plt.ylabel("Y Values")

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

Exercise 4

You are required to create a histogram to visualize the distribution of a dataset.

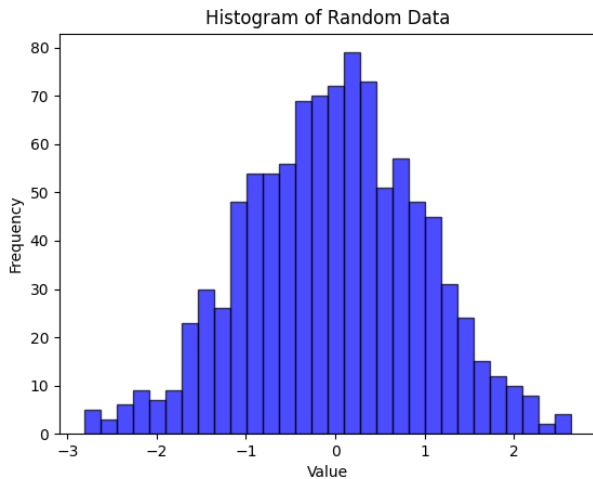
You need to perform the following tasks:

1. Generate a random dataset using NumPy: `data = np.random.randn(1000)`.
2. Use `plt.hist()` to create a histogram of the generated data.
3. Set the number of bins to 30 and customize the color of the bars to blue with some transparency.
4. Add a title "Histogram of Random Data" and label the x-axis as "Value" and the y-axis as "Frequency".
5. Display the histogram using `plt.show()`.

Requirements:

- Use `np.random.randn()` to generate random data.
- Use `plt.hist()` to create the histogram.
- Use `plt.title()`, `plt.xlabel()`, and `plt.ylabel()` to add titles and labels.
- Use `plt.show()` to display the histogram.

Expected output:



The following functions will be useful for this exercise:

- `hist()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.hist.html
- `title()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.title.html
- `xlabel()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.xlabel.html
- `ylabel()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.ylabel.html
- `show()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.show.html

Exercise 4 Solutions

```
import matplotlib.pyplot as plt
import numpy as np

# Generate random data
data = np.random.randn(1000)

# Create the histogram
plt.hist(data, bins=30, color='blue', alpha=0.7, edgecolor='black')

# Add title and labels
plt.title("Histogram of Random Data")
plt.xlabel("Value")
plt.ylabel("Frequency")

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

Exercise 5

You need to create a line plot and save it to a file.

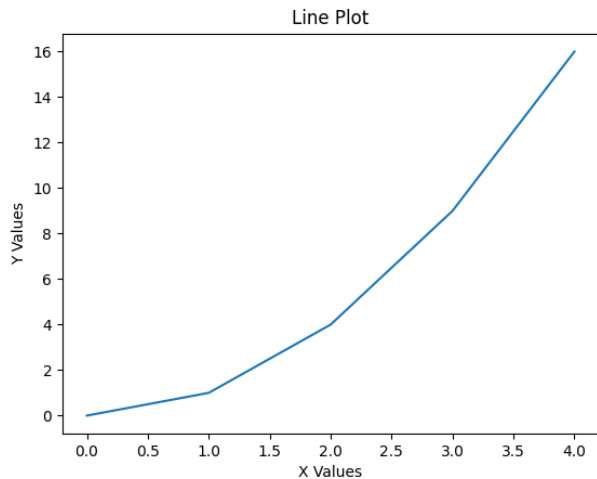
You need to perform the following tasks:

1. Create two lists:
 - `x = [0, 1, 2, 3, 4]`
 - `y = [0, 1, 4, 9, 16]`
2. Use `plt.plot()` to create a line plot with the given data.
3. Add a title **"Line Plot"** and label the x-axis as **"X Values"** and the y-axis as **"Y Values"**.
4. Save the plot to a file named **"line_plot.png"** using `plt.savefig()`.
5. Display the plot using `plt.show()`.

Requirements:

- Use `plt.plot()` to create the line plot.
- Use `plt.title()`, `plt.xlabel()`, and `plt.ylabel()` to add titles and labels.
- Use `plt.savefig()` to save the plot to a file.
- Use `plt.show()` to display the plot.

Expected output:



The following functions will be useful for this exercise:

- `plot()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.plot.html
- `title()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.title.html
- `xlabel()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.xlabel.html
- `ylabel()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.ylabel.html
- `savefig()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.savefig.html
- `show()`: https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.show.html

Exercise 5 Solutions

```
import matplotlib.pyplot as plt
```

```
# Create the data
```

```
x = [0, 1, 2, 3, 4]
```

```
y = [0, 1, 4, 9, 16]
```

```
# Create the line plot
```

```
plt.plot(x, y)
```

```
# Add title and labels
```

```
plt.title("Line Plot")
```

```
plt.xlabel("X Values")
```

```
plt.ylabel("Y Values")
```

```
# Save the plot to a file
```

```
plt.savefig("line_plot.png")
```

```
# Display the plot
```

```
plt.show()
```