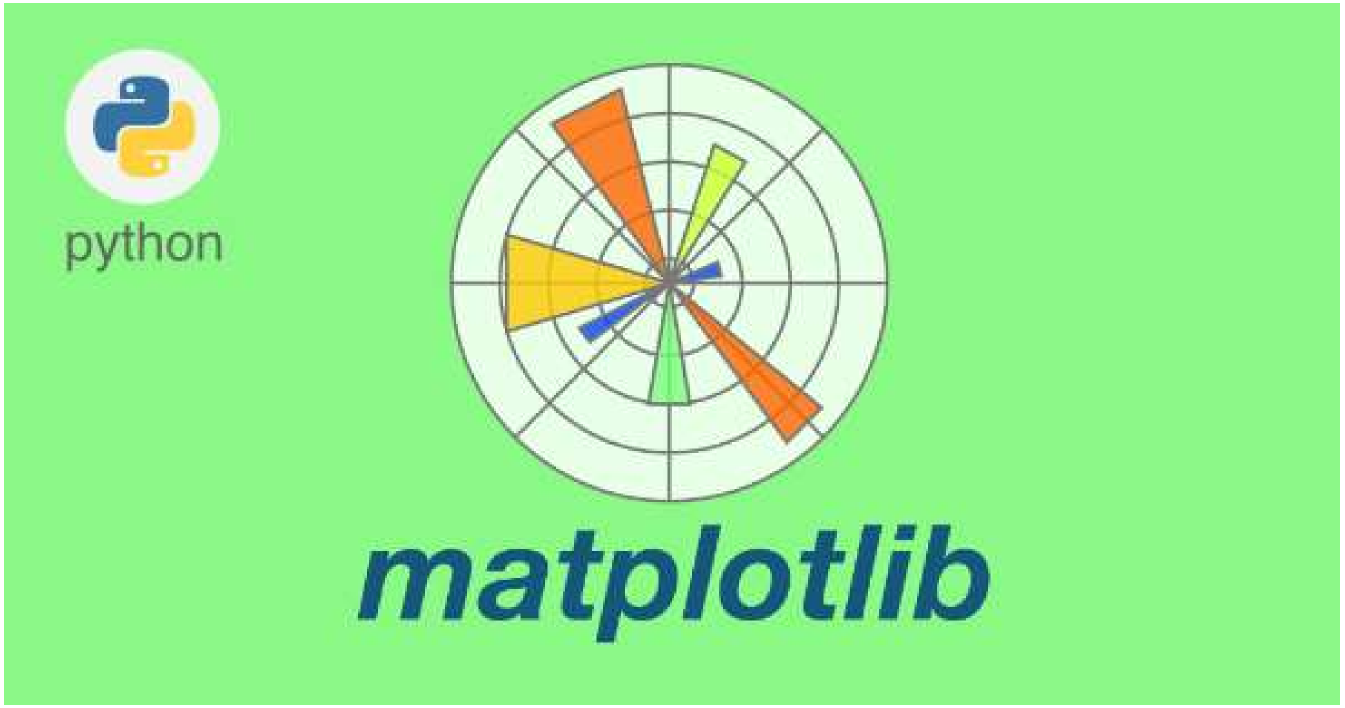
plotting in python

Lecture 5: Introduction to plotting in python



M.Sc. Riyadh Saeed Agid

Salahaddin University – Erbil

riyadh.agid@su.edu.krd

Installation of Matplotlib

If you have Python and PIP already installed on a system, then installation of Matplotlib is very easy.
Install it using this command:

```
C:\Users\Your Name>pip install matplotlib
```

You need to have the following installed on your computer to be able to make nice plots
Python
NumPy
Matplotlib

Also you can check which version you installed by doing :
Information is available in the **sys.version** string in the **sys** module:

```
>>>import sys
>>>print (sys.version)
>>>import numpy
>>>numpy.__version__
```

It should be something like this

‘1.5.1’

```
>>>import matplotlib
>>>matplotlib.__ version__
```

Should give you something like:

‘1.0.1’

Basic plotting

MatPlotLib is a python based plotting package. It is what we are using to make all of the plots in this stage. So, all of the plotting functions below are contained in the **MatPlotLib** package.

Function	Syntax
Open a new plotting window	figure()
Graph of y against x	plot(x, y)
Clear the plotting window	clf()
Graph of array (b) against array (a) where both axes are log scales	loglog(a, b)
Graph of array (b) against array (a) with log scale on the x axis	semilogx(a, b)
Graph of array (b) against array (a) with log scale on the y axis	semilogy(a, b)

Function	Description
Plot	plots y versus x as lines and/or markers
Show	displays a figure
Axis	sets some axis properties
Xlabel	sets the label for the x-axis
Ylabel	sets the label for the y-axis
Title	sets a title for the axes
Subplot	adds a subplot to the current figure
Subplots_adjust	tunes the subplot layout
Legend	places a legend on the axes
Figure	creates a new figure
Savefig	saves the current figure

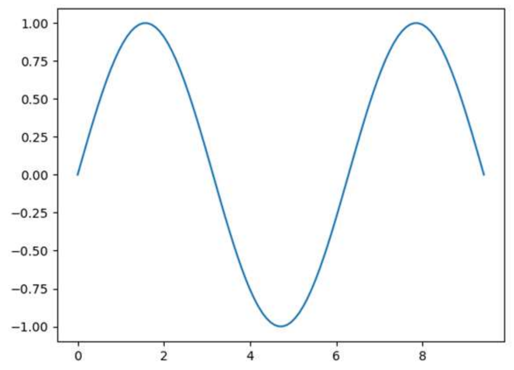
Two plot types which you will find are used very often in python, are line and scatter plots.

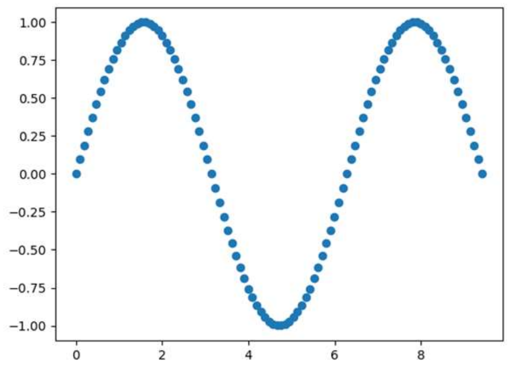
❑ Line and scattered plots

The scatter() function plots one dot for each observation. It needs two arrays of the same length, one for the values of the x-axis, and one for values on the y-axis:

Example

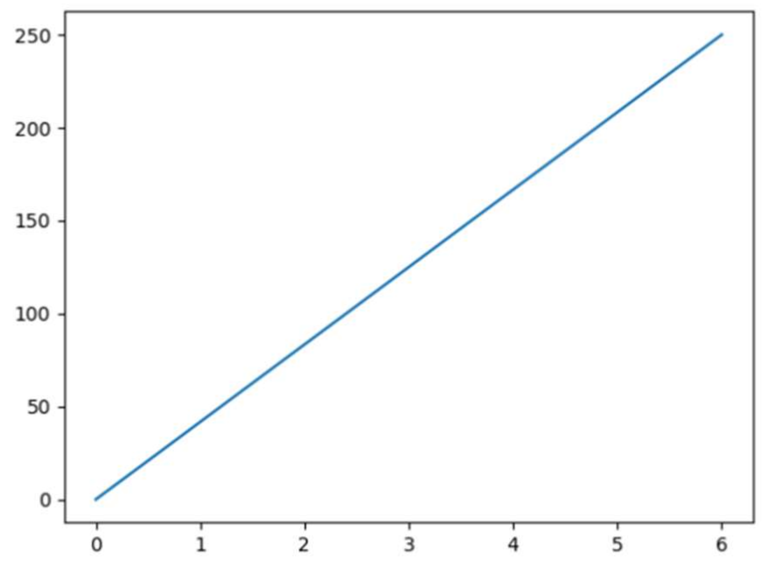
```
>>> import matplotlib.pyplot as plt
>>> from numpy import*
>>> x= linspace(0, 3*pi, 100)
>>>plt.plot(x, sin(x) ) #line plot
>>> plt.show()
>>>plt.plot(x, sin(x), ‘o’) #Scattered
>>>plt.show()
```





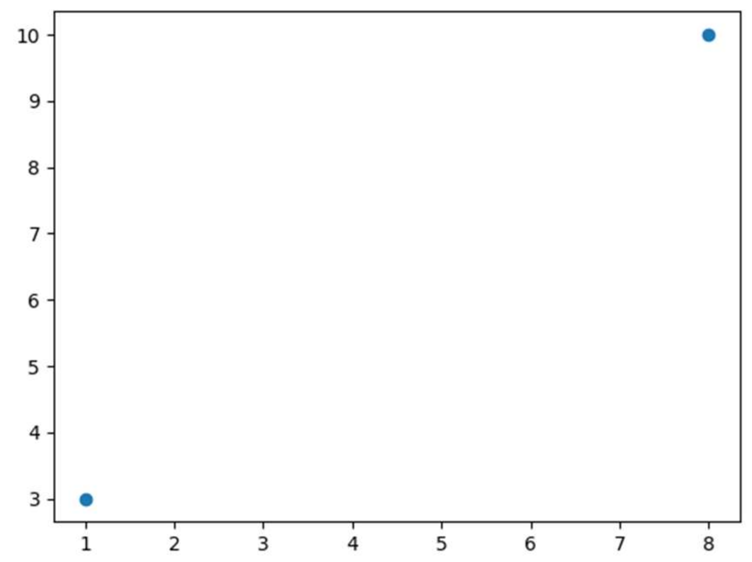
❑ Plotting x and y points

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

x = np.array([0, 6])
y = np.array([0, 250])

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

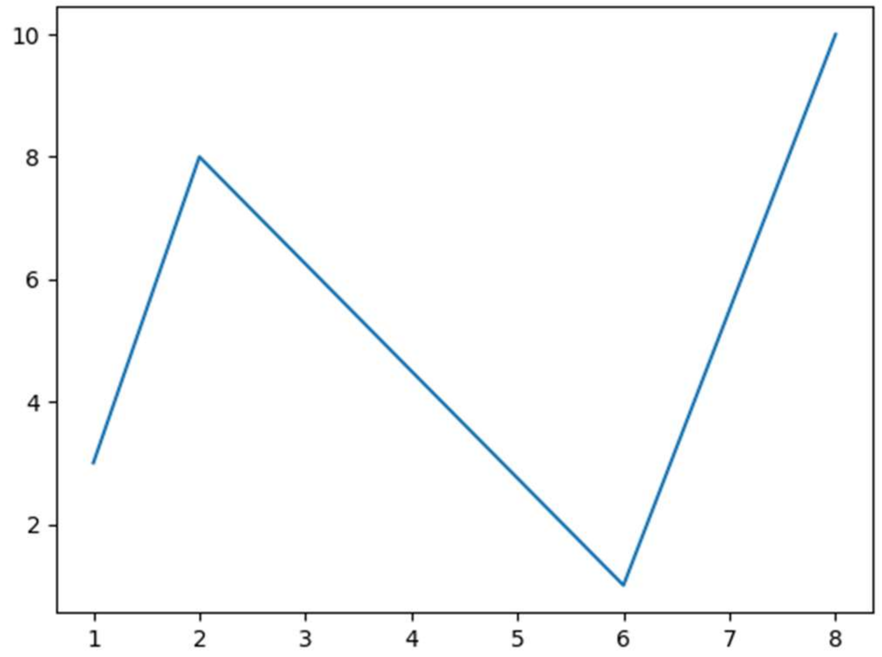


❑ Plotting Without Line

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

xpoints = np.array([1, 8])
ypoints = np.array([3, 10])

plt.plot(xpoints, ypoints, 'o')
plt.show()
```

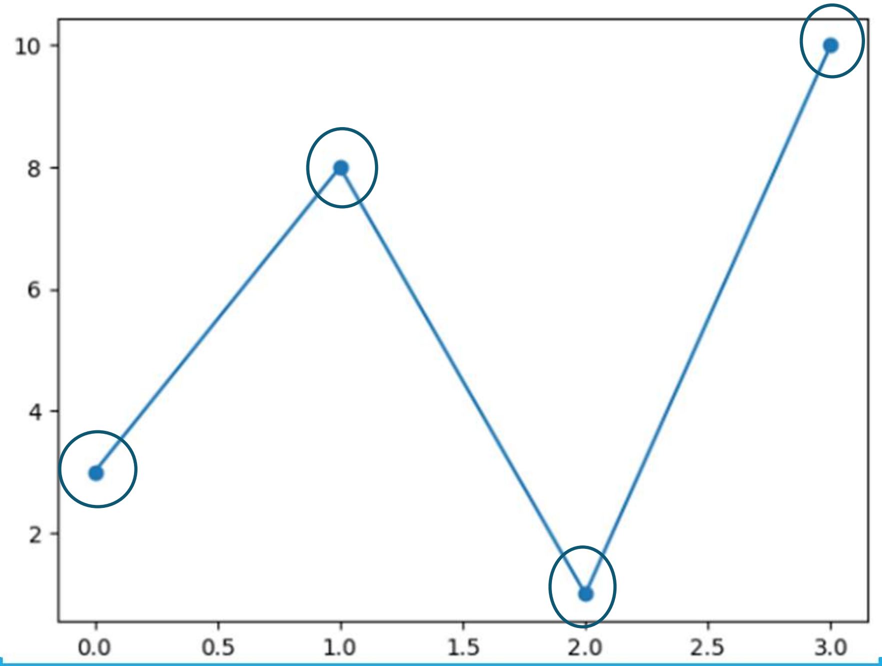


```
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([1, 2, 6, 8])
ypoints = np.array([3, 8, 1, 10])

plt.plot(xpoints, ypoints)
plt.show()
```



❑ Matplotlib Markers

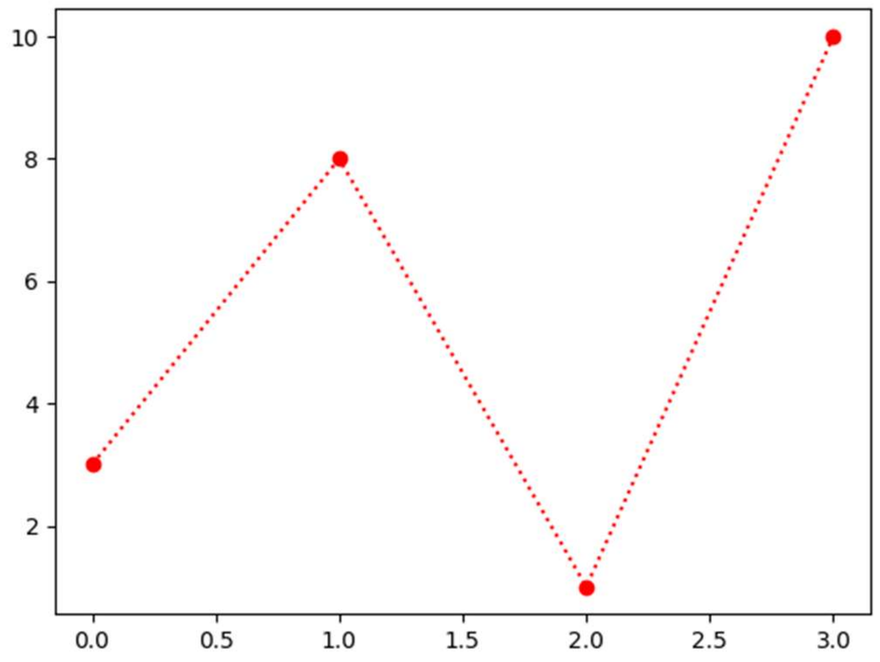
```
import matplotlib.pyplot as plt
import numpy as np

ypoints = np.array([3, 8, 1, 10])

plt.plot(ypoints, marker = 'o')
plt.show()
```



Marker	Description
'o'	Circle
'*'	Star
'.'	Point
','	Pixel
'x'	X
'X'	X (filled)
'+'	Plus
'P'	Plus (filled)
's'	Square
'D'	Diamond
'd'	Diamond (thin)
'p'	Pentagon
'H'	Hexagon
'h'	Hexagon
'v'	Triangle Down
'^'	Triangle Up

❑ *marker|line|color*

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

ypoints = np.array([3, 8, 1, 10])

plt.plot(ypoints, 'o:r')
plt.show()
```



❑ Line Reference

Line Syntax	Description
'-'	Solid line
':'	Dotted line
'--'	Dashed line
'-.'	Dashed/dotted line

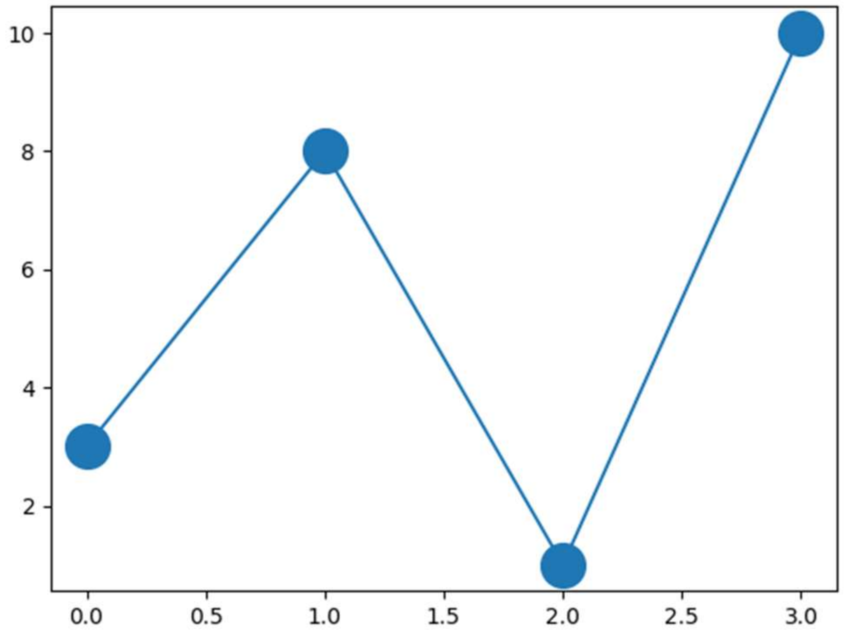
- The earlier plot are by no means complete. It require an axes labels and title. Furthermore, one may also need to specify the scale or limits of the axes, or the width and colour of the lines to make the graph better and easier to read. Also, it is very useful to be able to plot more than one set of data on the same axes and to be able to distinguished between them by using different line, marker styles, and colours. Here are some colour options:

❑ Color Reference

Character	Colour
b	blue
g	green
r	red
c	cyan
m	magenta
y	yellow
k	black

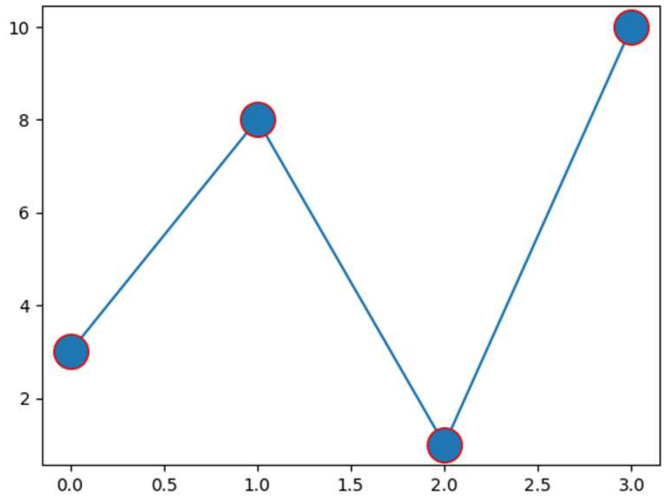
❑ Marker Size

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

ypoints = np.array([3, 8, 1, 10])

plt.plot(ypoints, marker = 'o', ms = 20)
plt.show()
```



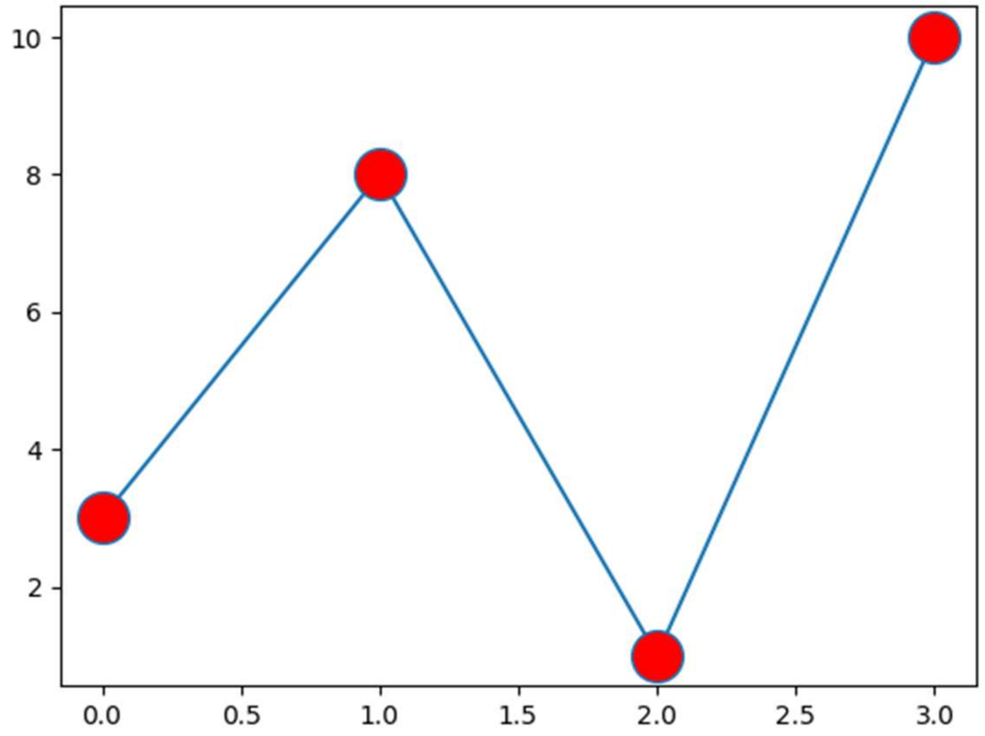
❑ markeredgecolor

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

ypoints = np.array([3, 8, 1, 10])

plt.plot(ypoints, marker = 'o', ms = 20, mec = 'r')
plt.show()
```



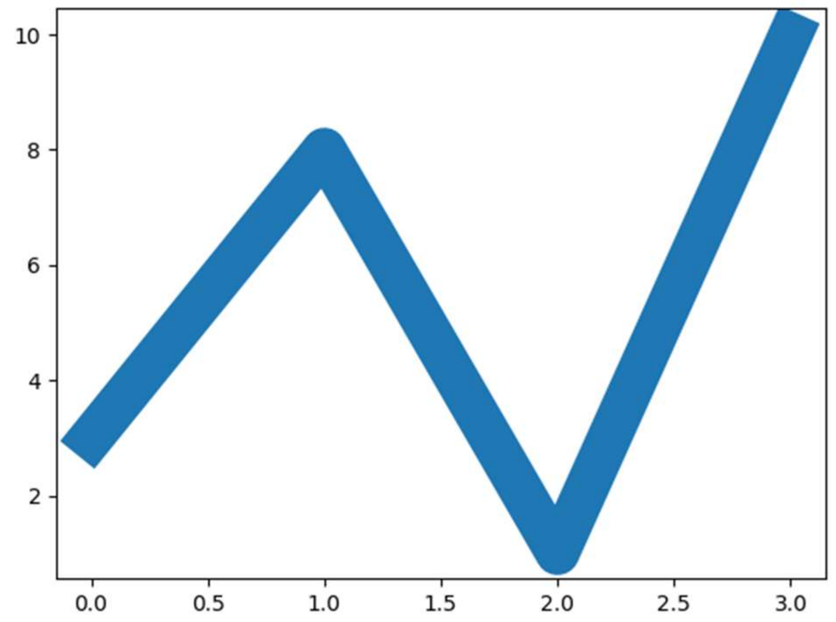
❑ Markerfacecolor

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

ypoints = np.array([3, 8, 1, 10])

plt.plot(ypoints, marker = 'o', ms = 20, mfc = 'r')
plt.show()
```

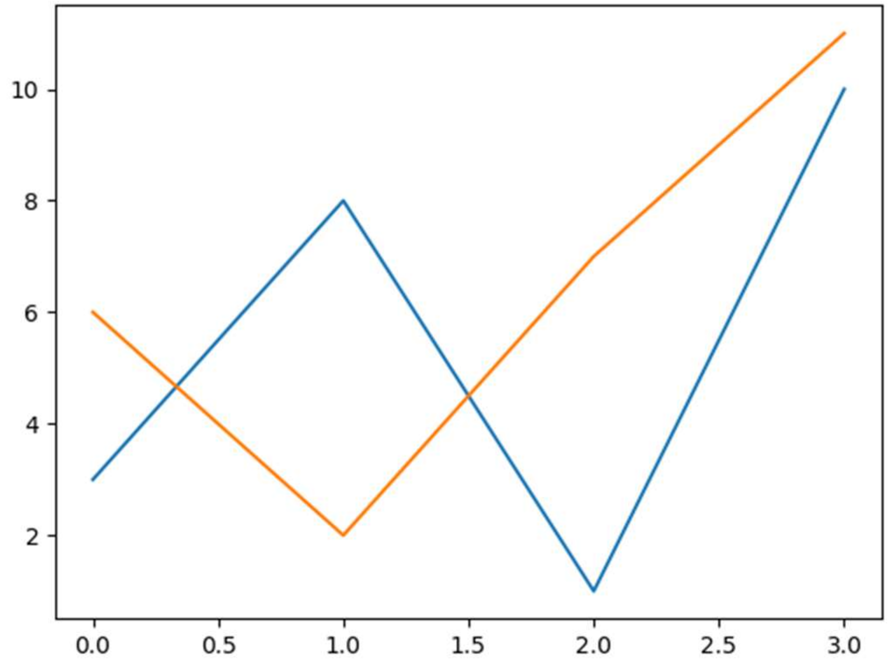


❑ Line Width

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

ypoints = np.array([3, 8, 1, 10])

plt.plot(ypoints, linewidth = '20.5')
plt.show()
```



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

x1 = np.array([0, 1, 2, 3])
y1 = np.array([3, 8, 1, 10])
x2 = np.array([0, 1, 2, 3])
y2 = np.array([6, 2, 7, 11])

plt.plot(x1, y1, x2, y2)
plt.show()
```



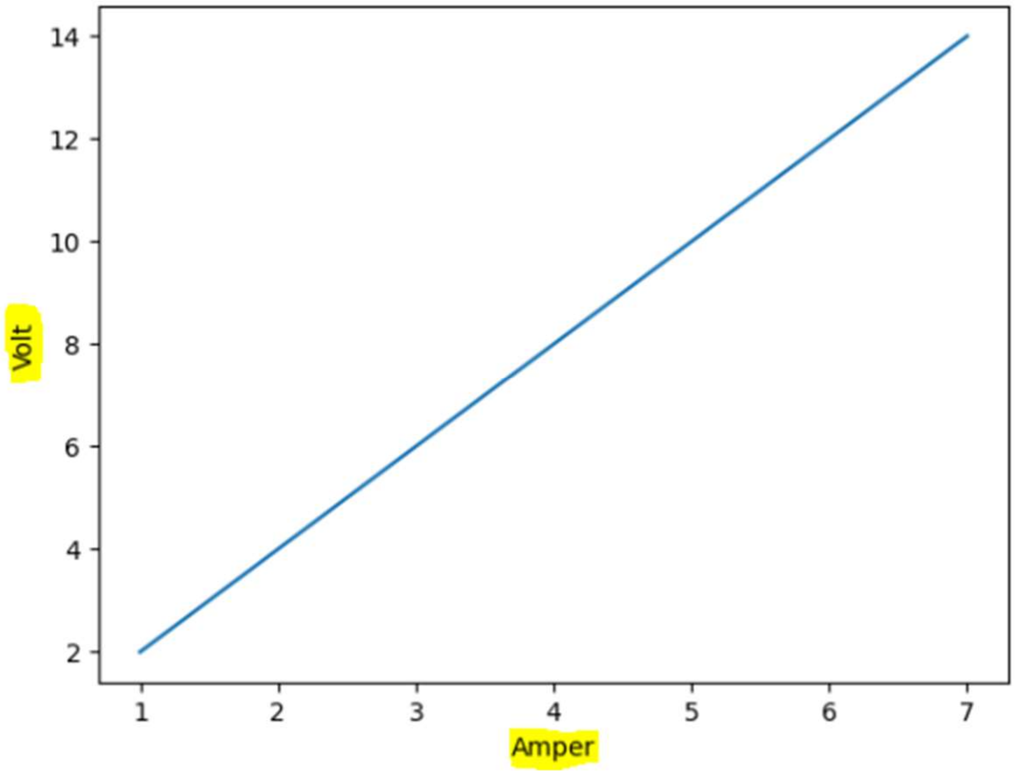
❑ Matplotlib Labels and Title

Create Labels for a Plot

With Pyplot, you can use the xlabel() and ylabel() functions to set a label for the x- and y-axis

Example
Add labels to the x- and y-axis:

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

x = np.array([1, 2, 3, 4, 5, 6, 7])
y = np.array([2, 4, 6, 8, 10, 12, 14])

plt.plot(x, y)

plt.xlabel("Amper")
plt.ylabel("Volt")

plt.show()
```



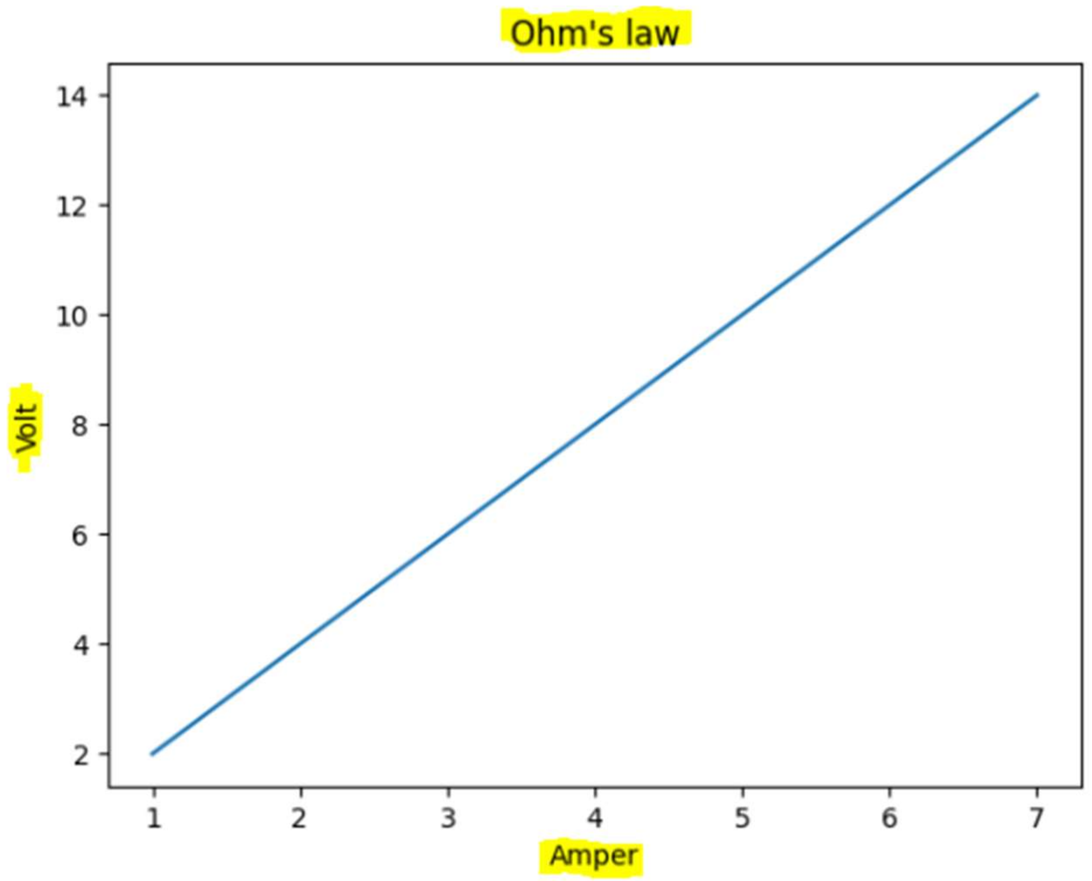
Create a Title for a Plot

With Pyplot, you can use the `title()` function to set a title for the plot.

Example

Add a plot title and labels for the x- and y-axis:

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

x = np.array([1, 2, 3, 4, 5, 6, 7])
y = np.array([2, 4, 6, 8, 10, 12, 14])

plt.plot(x, y)

plt.title("Ohm's law")
plt.xlabel("Amper")
plt.ylabel("Volt")

plt.show()
```



Thank you