Lesson 10.2

Python Scripting 2
Plotting Data with MatPlotLib

Learning Objectives

Upon completion of this lesson, students will be able to

1) Take a set of data, plot it, and fit it to a polynomial
2) Take a set of data and plot it as a histogram

Review: Lists

Building vectors and matrices in Python involves a structure called a List. Lists are enclosed by square brackets and can be made of any data type:

Vector: a = [0,’t’,2,’Y’,4]

Matrix: b = [[1,2,3],[4,5,6]]

Strings are just a list of characters.

c = ‘Python’

We will use Lists to store data to plot

NumPy

NumPy contains a set of numerical routines that support scientific computing in Python. We will be using it in this lesson and it can be included in a script by typing:

```
import numpy as np
```

Then any of the numpy functions can be called by typing

```
np.NumPyFunction
```

matplotlib

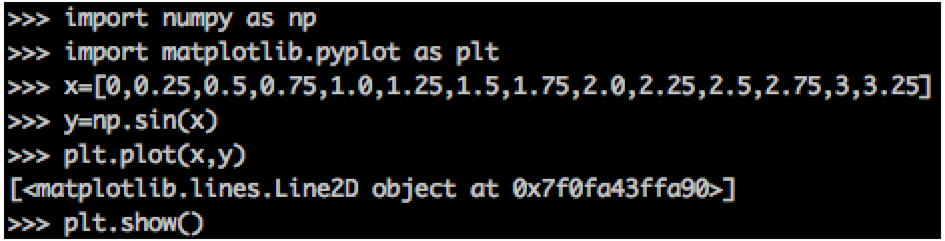
A comprehensive library for creating static, animated, and interactive visualizations in Python

A complete list of functionality and HowTos are found on its website: matplotlib.org

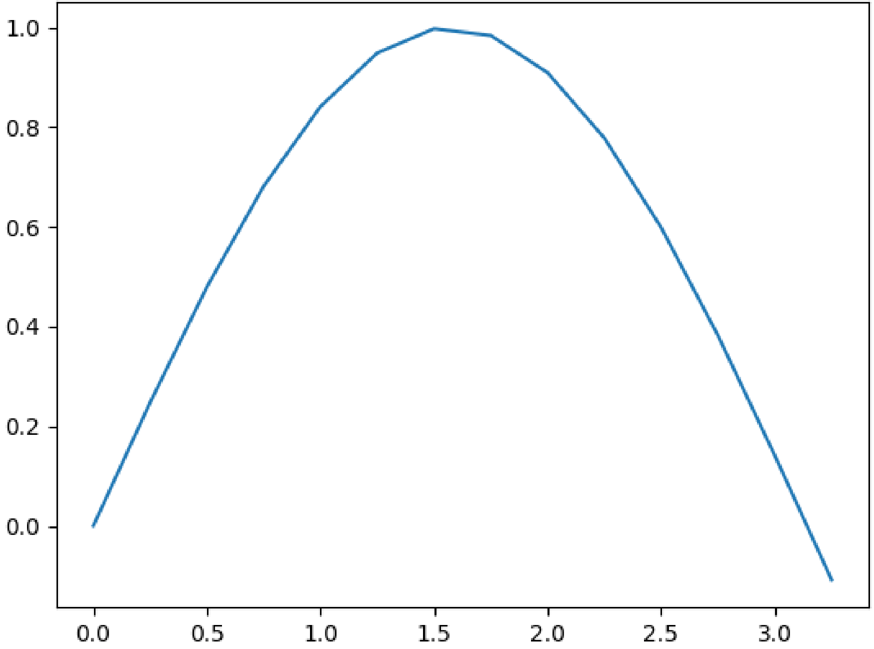
We will examine two functions in particular: plot and hist

matplotlib.pyplot.plot

The plot function included in matplotlib exists in a library called pyplot. A first plot can be made by following this example:



```
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> x=[0,0.25,0.5,0.75,1.0,1.25,1.5,1.75,2.0,2.25,2.5,2.75,3,3.25]
>>> y=np.sin(x)
>>> plt.plot(x,y)
[<matplotlib.lines.Line2D object at 0x7f0fa43ffa90>]
>>> plt.show()
```



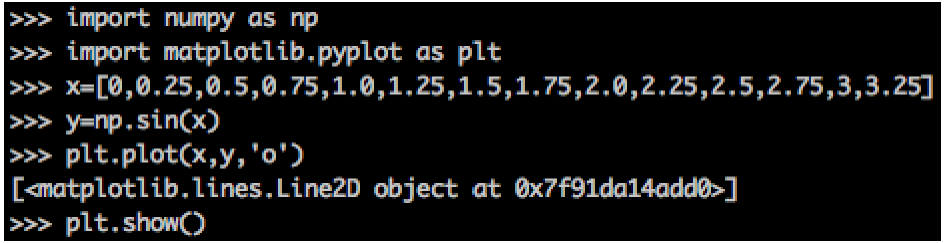
Note

1) that without any other options the plot function connects the data with straight lines.
1) ‘plt.show()’ makes the plot appear

Formatting the Data

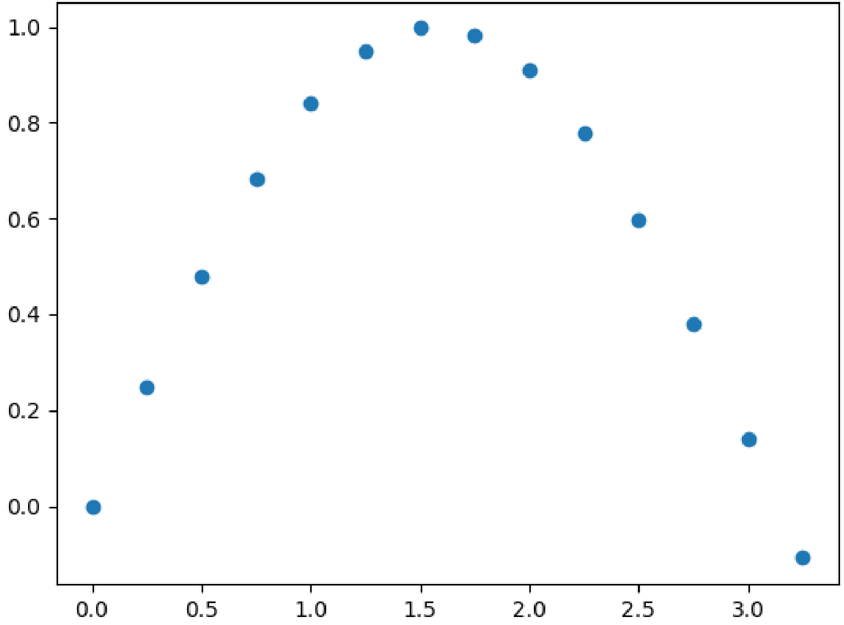
The basic syntax for the plot function is

plot(xdata, ydata, format)



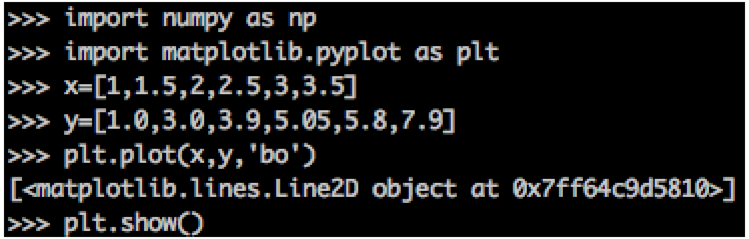
```
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> x=[0,0.25,0.5,0.75,1.0,1.25,1.5,1.75,2.0,2.25,2.5,2.75,3,3.25]
>>> y=np.sin(x)
>>> plt.plot(x,y,'o')
[<matplotlib.lines.Line2D object at 0x7f91da14add0>]
>>> plt.show()
```

‘format’ can be substituted with a string that where the user can change the marker, line and color. See the references for more details

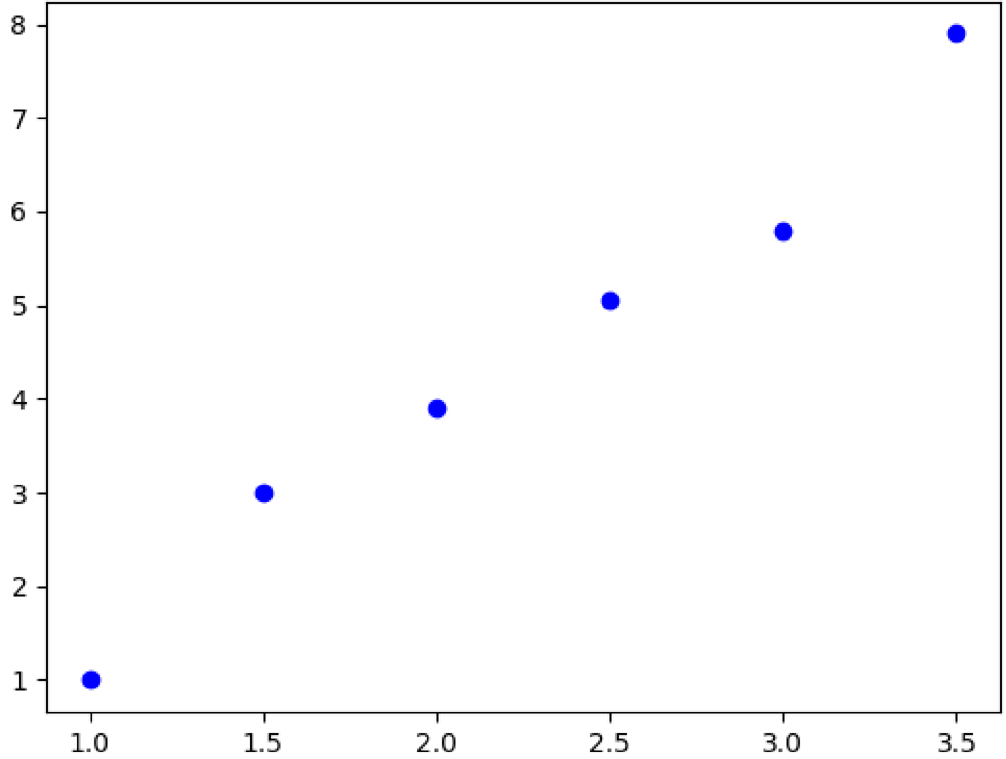


Fitting Data to a Line - Data

We will fit a line to the following data



```
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> x=[1,1.5,2,2.5,3,3.5]
>>> y=[1.0,3.0,3.9,5.05,5.8,7.9]
>>> plt.plot(x,y,'bo')
[<matplotlib.lines.Line2D object at 0x7ff64c9d5810>]
>>> plt.show()
```



numpy.polyfit

The polyfit function included in numpy will fit data to polynomials of degree n using a least squares method. Its basic syntax is:

fit = np.polyfit(x, y, degree)

where

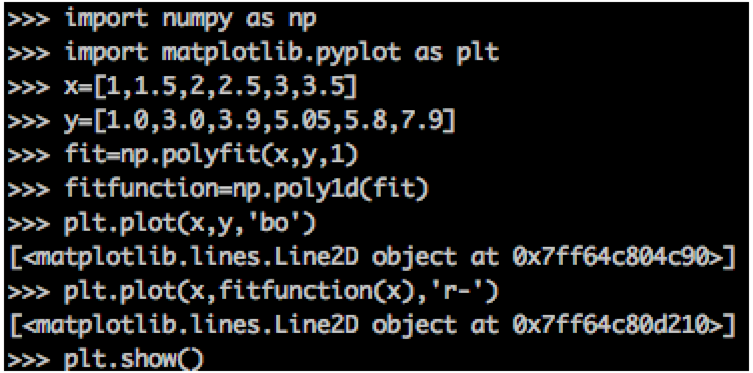
np invokes the numpy library previous imported

x,y are the data

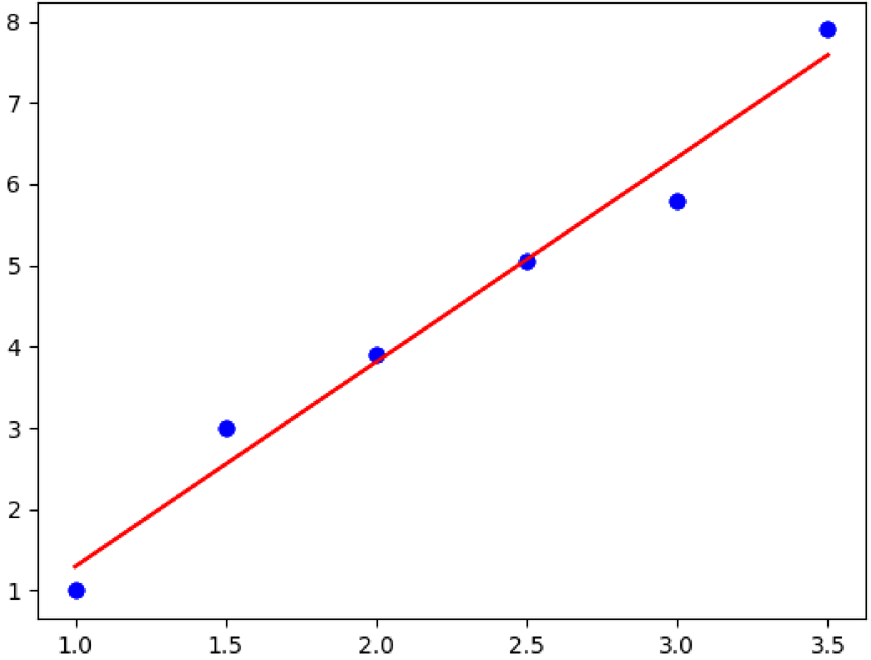
degree is the degree of the polynomial. For a line, this is 1

Fitting Data to a Line - Using polyfit

Here is an example using polyfit. We use another numpy function (poly1d) to convert the output of polyfit so that it can be used as a function with the data.



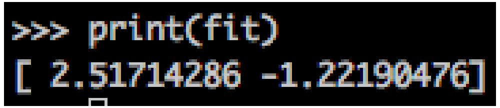
```
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> x=[1,1.5,2,2.5,3,3.5]
>>> y=[1.0,3.0,3.9,5.05,5.8,7.9]
>>> fit=np.polyfit(x,y,1)
>>> fitfunction=np.poly1d(fit)
>>> plt.plot(x,y,'bo')
[<matplotlib.lines.Line2D object at 0x7ff64c804c90>]
>>> plt.plot(x,fitfunction(x),'r-')
[<matplotlib.lines.Line2D object at 0x7ff64c80d210>]
>>> plt.show()
```



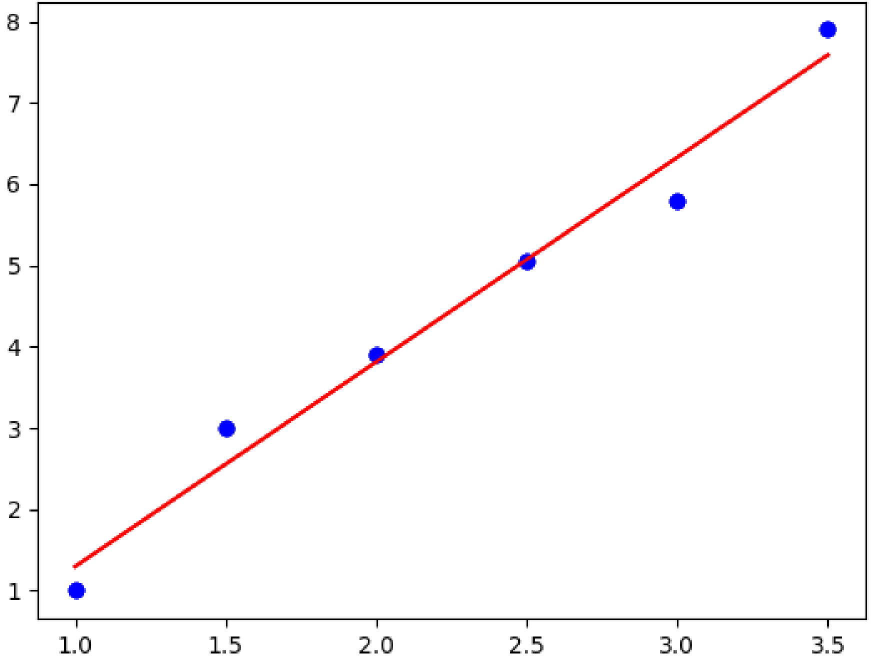
Notice the ‘r-’ in the format of the second plot function which gives the line its color.

polyfit - Slope and y-intercept

If you print the information saved in the fit variable from the output of polyfit



```
>>> print(fit)
[ 2.51714286 -1.22190476]
```



You will get the slope and the y-intercept

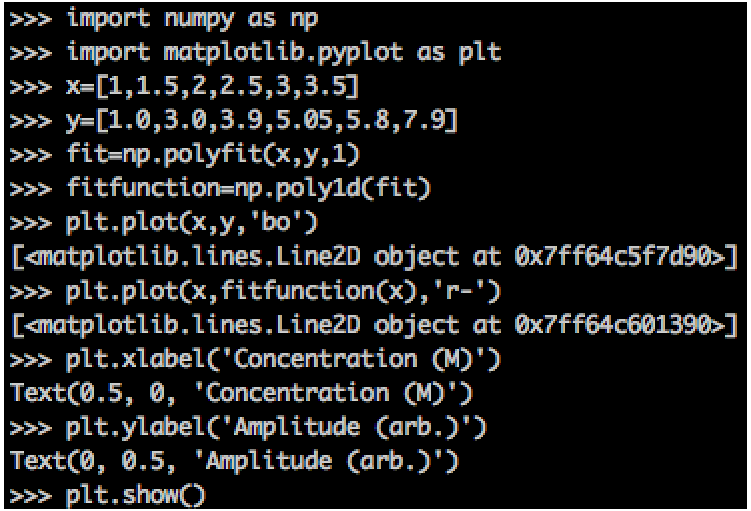
For the fitted line the equation is

$$f(x) = 2.517x - 1.2219$$

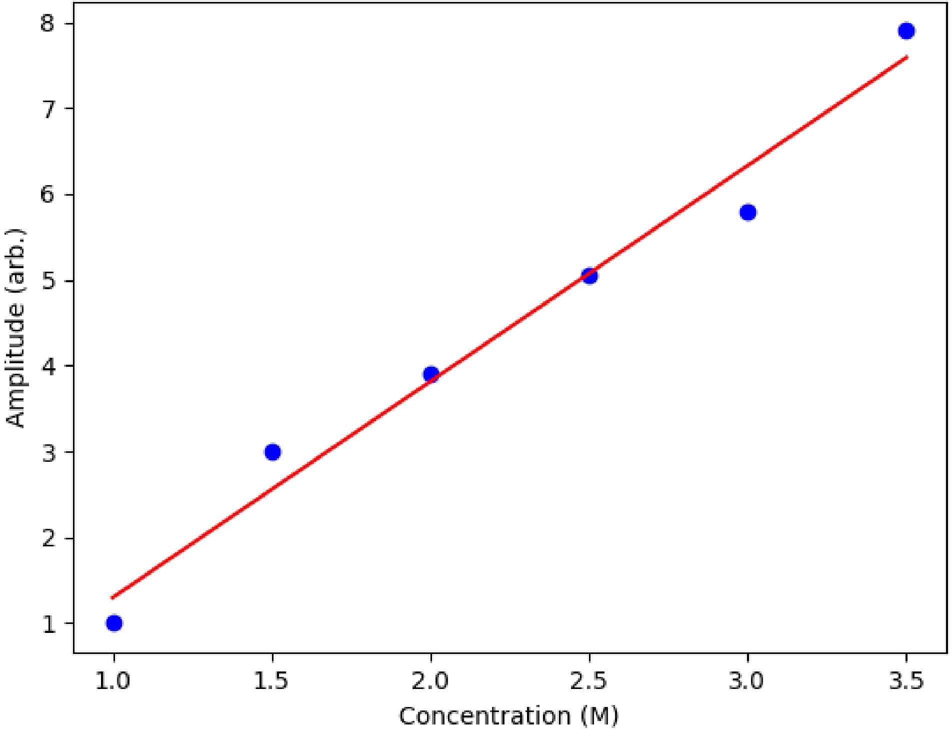
In general, polyfit will output a list of coefficients from the highest to lowest polynomial order term

Adding Axis Labels

Using the plt.xlabel(x-axis label) and plt.ylabel(y-axis label), the axes can be labeled



```
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> x=[1,1.5,2,2.5,3,3.5]
>>> y=[1.0,3.0,3.9,5.05,5.8,7.9]
>>> fit=np.polyfit(x,y,1)
>>> fitfunction=np.poly1d(fit)
>>> plt.plot(x,y,'bo')
[<matplotlib.lines.Line2D object at 0x7ff64c5f7d90>]
>>> plt.plot(x,fitfunction(x),'r-')
[<matplotlib.lines.Line2D object at 0x7ff64c601390>]
>>> plt.xlabel('Concentration (M)')
Text(0.5, 0, 'Concentration (M)')
>>> plt.ylabel('Amplitude (arb.)')
Text(0, 0.5, 'Amplitude (arb.)')
>>> plt.show()
```



Note that the arguments are strings

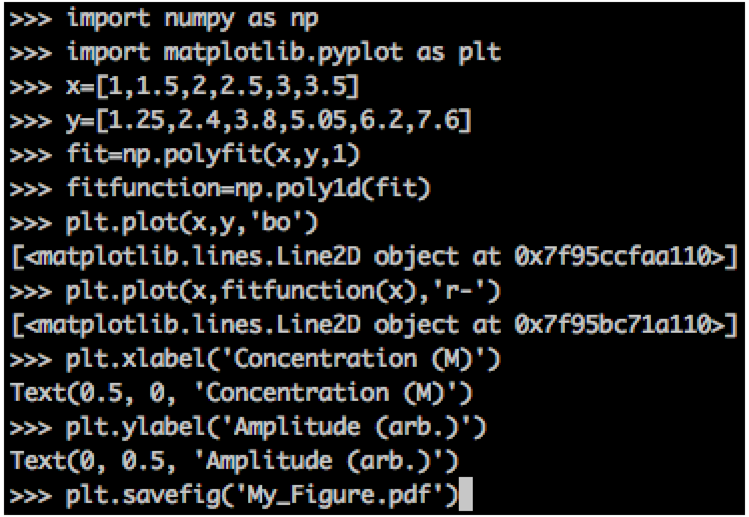
Saving the Plot to a PDF

The savefig command will save the figure to a file.

The type of file is defined by the extension so to save a .pdf add

plt.savefig(‘My_Figure.pdf’)

after defining all the parts of the figure.



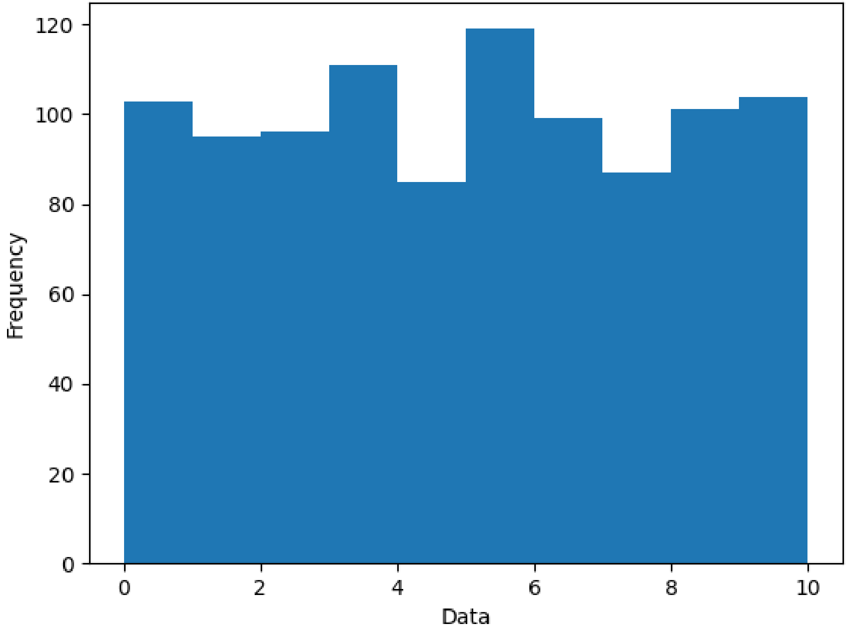
```
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> x=[1,1.5,2,2.5,3,3.5]
>>> y=[1.25,2.4,3.8,5.05,6.2,7.6]
>>> fit=np.polyfit(x,y,1)
>>> fitfunction=np.poly1d(fit)
>>> plt.plot(x,y,'bo')
[<matplotlib.lines.Line2D object at 0x7f95ccfaa110>]
>>> plt.plot(x,fitfunction(x),'r-')
[<matplotlib.lines.Line2D object at 0x7f95bc71a110>]
>>> plt.xlabel('Concentration (M)')
Text(0.5, 0, 'Concentration (M)')
>>> plt.ylabel('Amplitude (arb.)')
Text(0, 0.5, 'Amplitude (arb.)')
>>> plt.savefig('My_Figure.pdf')
```

Histograms

Histograms are plots that illustrates the frequency of a certain group of data

They are typically plotted as vertical bars where

- x-axis label identifies the group
- y-axis label identifies the frequency



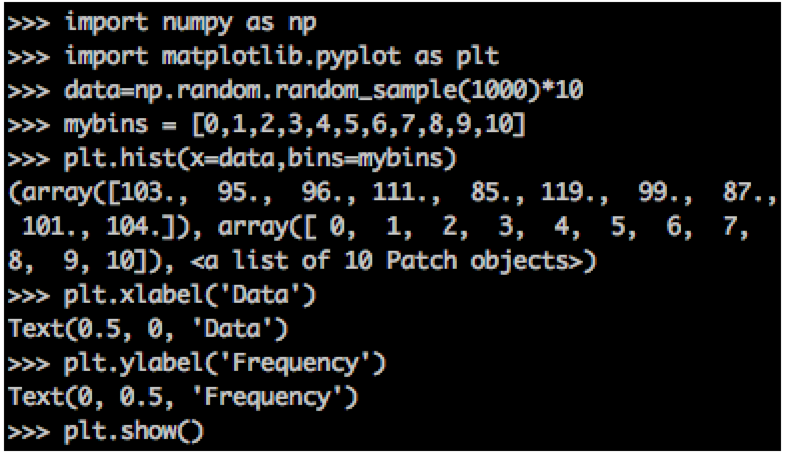
MatPlotLib hist Function

In MatPlotLib, the hist is the histogram function. The basic usage is

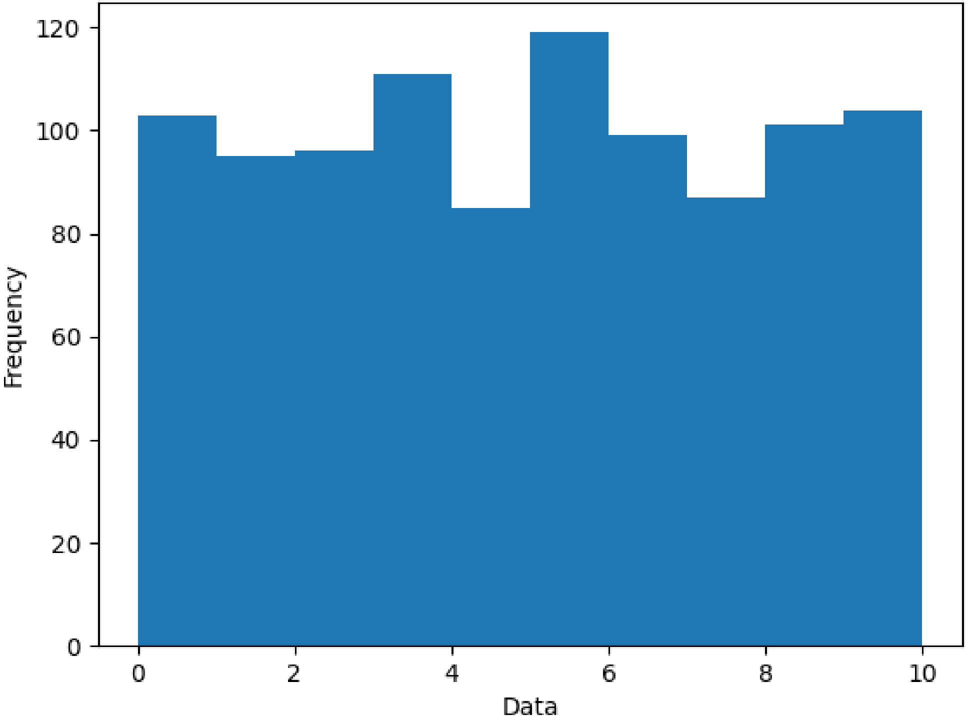
```
import matplotlib.pyplot as plt
data=[list of histogram data]
mybins=[list of group limits]
plt.hist(x=data, bins=mybins)
```

The data is the list of information you want to group
The bins is how you want to group it

hist Example



```
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>> data=np.random.random_sample(1000)*10
>>> mybins = [0,1,2,3,4,5,6,7,8,9,10]
>>> plt.hist(x=data,bins=mybins)
(array([103.,  95.,  96., 111.,  85., 119.,  99.,  87.,
 101., 104.]), array([ 0,  1,  2,  3,  4,  5,  6,  7,
8,  9, 10]), <a list of 10 Patch objects>)
>>> plt.xlabel('Data')
Text(0.5, 0, 'Data')
>>> plt.ylabel('Frequency')
Text(0, 0.5, 'Frequency')
>>> plt.show()
```



-Data comes from random number generator between 0-10

-Bins set to group data by the integer

Summary

- MatPlotLib is a fully featured plotting suite
- The plot function will plot x,y data points in a scatter or join data by lined segments
- The hist function will plot grouped data using bars that indicate the frequency each data point appears