week9example

December 6, 2018

1 Week 9 - Numpy and Matplotlib

This week will be very hands on, as such, there are no lecture slides!

```
In [1]: # We'll start by importing the packages
        import numpy as np
        import matplotlib.pyplot as plt
```

```
In [6]: # The central object in numpy is an array
        # These are similar to lists, but more powerful

        # Create a simple array
        my_array = np.array([1,2,3,4,5,6,7,8])
        # Or, like with range()
        my_array2 = np.arange(1,9)
```

```
In [19]: # Numpy arrays allow us to perform element-wise operations
         np.sum(my_array - my_array2)
```

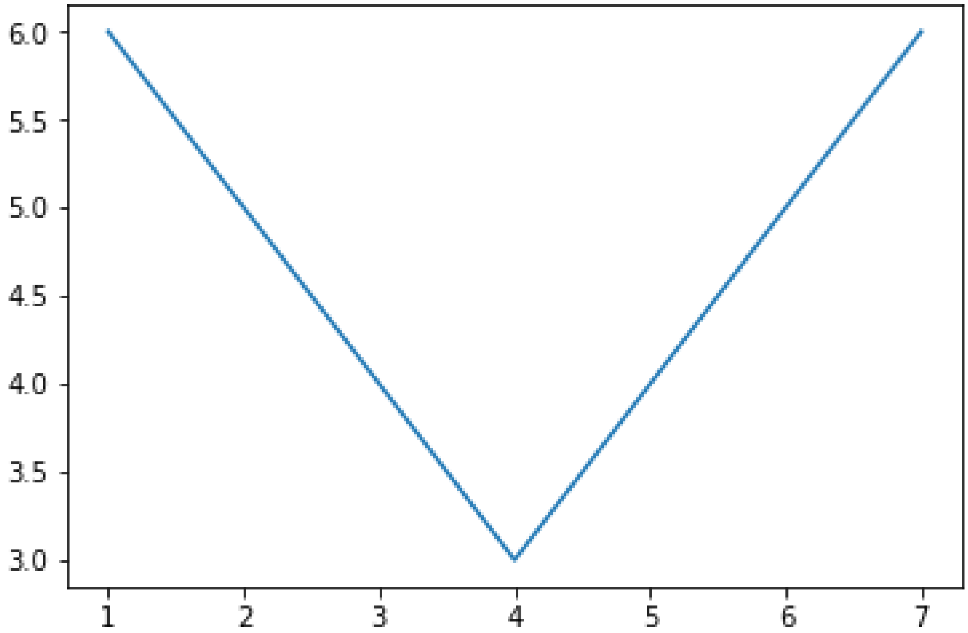
```
In [25]: # Loading data through numpy is easy to
         # There's a few ways, but we'll use genfromtext
         my_data = np.genfromtxt("data/exampleData.csv", delimiter=",",names=True,dtype=None)
         # Delimiter tells numpy how columns are seperated
         # Names tells numpy that the top row contains column names
         # dtype=None warns numpy that it needs to figure out the kinds of data itself
```

```
In [18]: for row in my_data:
             print("My name is: " + str(row[0])[2:-1])
         # The b' before the names indicates the datatype numpy has chosen
         # When converting to string, we want to trim this away
         # (There are better, but more complicated, ways)
```

```
My name is: alex
My name is: john
My name is: frank
My name is: bob
```

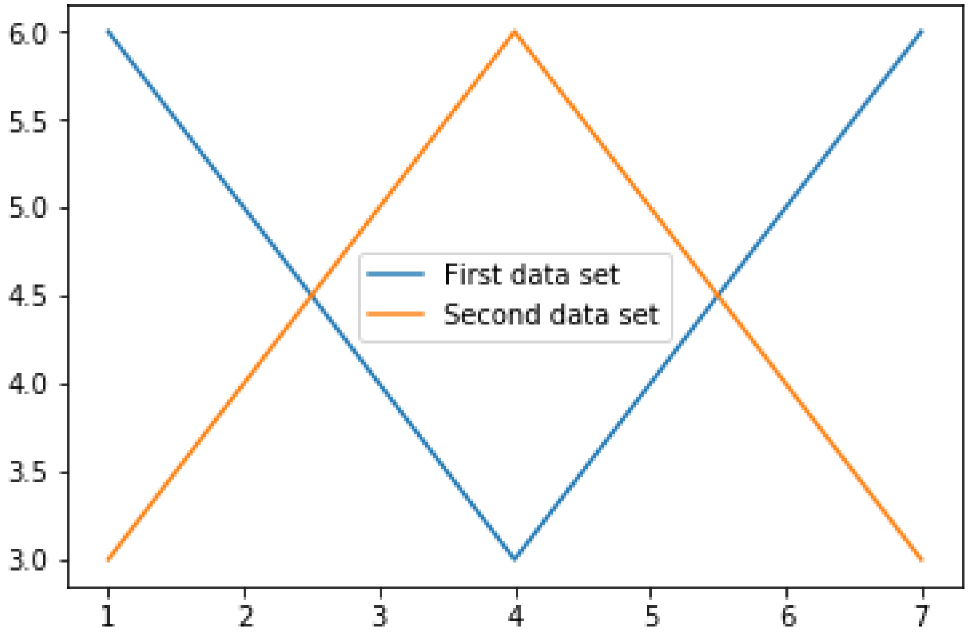
```
In [20]: # Plotting with matplotlib is straightforward
         x_data = [1,2,3,4,5,6,7]
         y_data = [6,5,4,3,4,5,6]

         plt.plot(x_data, y_data) # Create the plot
         plt.show() # Tell matplotlib to show it to us
```



```
In [22]: # There are lots of options to make plots look nicer
         y2_data = [3,4,5,6,5,4,3]

         # We can also plot multiple data sets on the same canvas
         plt.plot(x_data, y_data, label="First data set")
         plt.plot(x_data, y2_data, label="Second data set")
         plt.legend()
         plt.show()
```



```
In [24]: # You can see a much fancier plot which I made in the repository
```