Numpy

## Introduction to Numpy

* Numpy arrays usually come in two forms: Vectors (1-d arrays) and matrices (2-d arrays).
* 1 d matrices can only have one row or one column of data.
* To start using numpy, install the library with command prompt (conda install numpy or pip install numpy)
* Import the numpy library hereafter referred to as ‘np’

# Numpy Arrays

## Creating Numpy Arrays

* A list can be cast as a 1d array using ‘array’ on a python object {np.array(objectname)}
* A list of lists (nested list) such as [[1,2,3], [4,5,6], [7,8,9]] can be cast as a 2d array using ‘array’ on a python object {np.array(objectname)}
* A more efficient method of generating an array is to use np.arange. it works like range in python but returns the values as an array {np.arange(start, stop, step)}
* You can generate a 2d array of 0s with np.zeroes((a,b)) where a and b are number of rows and columns.
* You can generate a 1d array of 0s with np.zeroes(a) where a is number of zeroes for the array
* You can generate a 2d array of 1s with np.ones((a,b)) where a and b are number of rows and columns.
* You can generate a 1d array of 1s with np.ones(a) where a is number of columns for a 2d array
* You can use np.linspace(start, stop, no) to generate a 1d no of evenly spaced numbers between a range.
* You can create an identity matrix with np.eye(a). a is the number of columns and rows (which are the same for an identity matrix). It creates a matrix of that number of rows by columns with a diagonal of 1s and os elsewhere.

N:B: What is an identity matrix?

## Numpy Random libraries

* You can generate an array of random numbers between 0 and 1 by using the np.random.rand(a) or np.random.rand(a,b) for 1 and 2 dimensional arrays respectively.
* You can generate an array of random numbers between 0 and 1 from a normal or Gaussian distribution by using the np.random.randn(a) or np.random.randn(a,b) for 1 and 2 dimensional arrays respectively.
* You can generate an array of random integers by using the np.random.randint(a,b) or np.random.randint(a,b,c) for a single integer or a group of integers respectively where a=start, b = stop, c = no of integers

## Attributes and methods of an array

* You can use the reshape attribute to reshape an array with arrayname.reshape(a,b) where a and b are the rows and columns of the new array respectively.
* You can check the highest and lowest values in the array using arrayname.max() and arrayname.min() respectively.
* You can check the index location of the highest and lowest values in the array using arrayname.argmax() and arrayname.argmin() respectively.
* You can check the shape of an array using the arrayname.shape attribute
* You can check the datatype of an array using the arrayname.dtype attribute

## Numpy indexing

1 d array

* You can call using index just similar to a python list
* You can call with arraynname[index], arraynname[start:stop], arraynname[start:stop:step] to call an element in an array respectively
* You can broadcast (change a number of elements in an array at once to a single element) an array by using arrayname[:] = new value. A broadcast changes the value permanently. \*\*\*

Note: the new value being broadcast can be a list, number, tuple, (as long as they are completely made up of integers) but not a dictionary

* You can copy an array by using arrayname.copy() and reassigning it.

2 d array

* There are 2 ways to grab an element from a 2d array: single and double bracket formats respectively.
* Double bracket involves using the index of the rows and columns to grab a value using two brackets e.g. arrayname[row index][column index].
* Single bracket involves using the index of the rows and columns to grab a value using a single bracket e.g. arrayname[row index, column index].
* You can use slice notations to grab a section of the array in 2d form with array\_name[start and end of rows, start and end of columns] e.g., array\_name[1:4, 3:7]

Conditional selection

* You can grab only the sections of an array that meet a Boolean condition using array\_name[array\_name </>/== a] which returns only the members of the array that meet the given condition

## NumPy operations

* For an array with array, you can operate an array with itself or another using arithmetic symbols such as array +/-/\*/\*\*array.
* For an array with a scalar, you can operate an array with itself or another using arithmetic symbols such as array +/-/\*/\*\*scalar.
* Numpy will usually not throw up errors when performing math operations but will instead give a warning along with other results, meaning, the code doesn’t break because of a single erroneous operation especially when having 0 as an element in the array.

## Universal array functions

* Universal array functions allow one to perform a function and broadcast the result across the array.
* They are called using np.functionname(arrayname)
* Examples include np.sin/cos/tan/log, np.exp/sqrt, etc.

Note: you can perform computations such as standard deviation, square root, etc