Kevin Dyer

10/10/21

Cheat Sheet #3

Numpy Arrays

Numpy arrays are grids of values, all the same type, that are the fundamental package for scientific computing within python. They can facilitate mathematical and other types of operations on large numbers of data. Numpy arrays differ from lists in that arrays have nested elements of the same size while lists can contain elements of different types. Arrays are more useful when dealing with longer sequences of data items while lists are preferred for shorter sequences. To make a Numpy array first import the numpy package by typing “import numpy as np” and then one can create a Numpy Array by typing something like “array1=np.array([1,2,3])” or “array2=np.ones([B,4])”.

To index a numpy array one must enter the following: “start:stop:step” repeating for every dimension with a ‘,’ between the values. [rows.columns], so it’s y axis then x axis, the opposite of conventional wisdom. For example, if one were wanting to see the first column they would type “myarray[:,0]” and if they wanted to see a box of data in the middle of a 2x2 they would type “myarray[:2,1,:3]”. Slicing a numpy array is like slicing a list except you can do it in more than one dimension. To slice a numpy array one would need to type the name of their array and then parentheses with the item they want the slice to extend from, colon, and the item they want the slice to extend to. So, it would look something like this: “myarray[from:to]”. Methods are actions associated with an object. Methods for numpy arrays would be written like this: “object.method(arg1…argn)” or “myarray.reshape(512). Attributes are the properties of an object. Key attributes for numpys include “myarray.ndim” which gives the dimensions of the array. Additionally, there is “myarray.shape” which gives the integers representing the size of the array in each dimension.

An example of when I’ve used numpy array in this course would be in my most recent homework assignment, assignment 7. In order to get a function displaying all the data for the month of October I created a numpy array of 31 zeros, “month\_median=np.zeros(31)”, which was then overwritten by the subset of data from my function.