Useful Numpy Commands

You may find the following Numpy functions helpful as you complete the labs in CS355:

np.linspace(start,stop,num) (https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.linspace.html)	Returns a Numpy array of size *num* with evenly spaced values between *start* and *stop*.
arr.shape (https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.ndarray.shape.html)	Returns a tuple of the size of each dimension in a Numpy array.
np.zeros(arr.shape) (https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.zeros.html)	Returns a new Numpy array of zeros of the same shape as *arr*. This is great for making buffer images.
np.array(list) (https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.array.html)	Turns a standard Python list into a Numpy array.
arr[start:stop:step,...,...] (https://docs.scipy.org/doc/numpy-1.13.0/reference/arrays.indexing.html)	Returns the portion of *arr* described by standard Python slice notation. Commas separate slice notatic for each dimension. A single colon returns every element of the specified dimension.
arr + number (https://docs.scipy.org/doc/numpy-1.13.0/user/basics.broadcasting.html)	Performs the element-wise addition of each element of *arr* with *number*. This also works with subtraction(- multiplication(*), division(/) and others.
arr1 + arr2 (https://docs.scipy.org/doc/numpy-1.13.0/user/basics.broadcasting.html)	Performs the element-wise addition between *arr1* wit *arr2*. This also works with subtraction(-), multiplication(*), division(/) and others. **Warning:** If *arr1* and *arr2* are both matrices, * performs a matri multiplication.
np.multiply(arr1,arr2) (https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.multiply.html)	Returns the element-wise multiplication of *arr1* and *arr2.*
np.matmul(arr1, arr2) (https://docs.scipy.org/doc/numpy-dev/reference/generated/numpy.matmul.html)	Returns the matrix multiplication *arr1* and *arr2.*
np.dot(arr1, arr2) (https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.dot.html)	Returns the dot product of *arr1* and *arr2*. **Warning:** Make sure that *arr1* and *arr2* are both row vectors or both column vectors.
np.cross(arr1, arr2) (https://docs.scipy.org/doc/numpy-	Returns the cross product of *arr1* and *arr2*. **Warning:** Make sure

1.13.0/reference/generated/numpy.cross.html)	that *arr1* and *arr2* are both row vectors or both colum vectors.
np.sum(arr, axis=None) (https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.sum.html)	Sums all elements in a Numpy array. If you specify a axis, it will only sum along that axis.
np.atleast_2D(arr) (https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.atleast_2d.html)	Turns a 1D list into a 2D Numpy matrix array.
arr.T (https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.ndarray.T.html)	If *arr* is 2D, it returns the matrix transpose.
np.amax(arr, axis=None) (https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.amax.html)	Returns the maximum value of a Numpy array. If you specify an axis, it will return each maximum value along that axis.
np.argmax(arr, axis=None) (https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.argmax.html)	Returns the index of the maximum value of a Numpy array. If you specify an axis, it will return the index of each maximum value along that axis.
np.clip(arr, min, max) (https://docs.scipy.org/doc/numpy-1.13.0/reference/generated/numpy.clip.html)	Returns an array where each element is between mi and max.