- Download numpy: Pip install numpy in terminal (make sure you in correct directory)
- To include numpy in your script: Import numpy as np
- Convert python list to np array: variable = np.array([data], datatype)
  - A list of lists will give us a matrix
  - Second argument can be the data type not required
- np.arange(start, stop, stepsize): used to create 1D array
  - Second argument required, start defaults to 0, step size defaults to 1
- np.linspace(start,stop,num_elts): creates array starting at value start and ending at value stop that are equally spaced
- np.empty((rows,cols)) : instantiate an empty array with uninitialized values of dimensions row x cols
  - Rows and cols default to 1 if not specified
- np.full((rows,cols),value): fills an array of size rows x cols with a given value
  - Specific cases: np.ones((rows,cols)) and np.zeros((rows,cols)) for array with all ones or zeros
- .dtype gives you the data type .shape gives you the dimensions of the array
- Indexing arrays works similarly to python lists
  - Two dimension array index: array[row][col]
    - Array[row] gives us whole row
  - Index from a given element to the end of that row: array[start_idx:]
  - Index from beginning to a given element: array[:end_idx]
    - Previous two commands gives us the data from the range specified
  - Want to index specific values of an array: array[[list of idxs]]
    - Gives you the data in an array from those specified idxs in that order
- Can do all basic arithmetic operations for arrays with same dimensions (add, subtract, multiply, divide, integer divide, raising to a power)
  - If you only want to work across a certain axis we can specify what axis we want to use
    - For 2D axis 0 is rows and axis 1 is cols
    - Ex: np.sum(array,axis=0) will sum across the rows which will output an array of the sums of the columns.
- np.sum(array): sums all elements of an array
- np.mean(array): takes the mean of all elements in array
- np.std(array): takes standard deviation of all elements in array
- Array. T: gives us the transpose of the matrix
- Dot product/Scalar product: np.dot(x1,x2)
- Matrix multiplication: np.matmul(x1,x2)
  - Need same number of cols for first matrix and rows for second matrix
  - Order matters!
- np.argmin(array, axis)/np.argmax(array, axis): returns the idx corresponding to min/max element
  - np.min(array, axis)/np.max(array, axis): returns the min/max value of the given array

    - Specify what axis to split on in order to get an array of the idxs holding the min value of a given row/col
    - Ex: np.argmin(array, axis=1): evaluate across the columns (axis = 1) so it will give us an array of the min values of each row
- np.reshape(array, (rows,cols)): takes an array and changes its shape so it’s more compatible with other arrays you are working with
    - If you list the size of one dimension as -1, NumPy will determine the appropriate dimension