**Numpy**

* Installation: Pip3 install numpy
* **Ability to perform operations on entire arrays, as opposed to Python lists on.**

e.g.

import numpy as np

np\_height = np.array(height)

np.weight = np.array(weight)

BMI = np.weight / np\_height\*\*2 -> This **ONLY works for numpy arrays**, not Python lists

**Note**: numpy arrays can only contain a SINGLE type! Different types will be converted into one type.

* Numpy Subsetting

BMI= np.array([21.852, 20.975, 20.75, 24.7, 21.4])

BMI[1]

#this gives 20.975

BMI>23

#this gives an numpy array: array([False, False, False, True, False], dtype=bool)

# Print out BMIs of all baseball players whose BMI is below 21

light = (bmi<21)

print(bmi[light])

* **2D Numpy Array**

e.g.

np\_2d = np.array([[1.73, 1.68, 1.71, 1.89, 1.79],

[69.4, 59.2, 63.6, 88.4, 68.7]])

np\_2d.**shape** ->this gives (2,5) #2 rows, 5 columns

#select the element at the 1st row and 3rd column

np\_2d[0][2] / np\_2d[0,2]

**#select the 2nd and 3rd column**

np\_2d[:, 1:3]

#select the 2nd row

np\_2d[1,:]

* **Basic Statistics (summary statistics)**

np.**mean**(np\_city[:,0])

np.**median**(np\_city[:,0])

np.**std**(np\_city[:,0]) -> standard deviation

np.**corrcoef**(np\_city[:,0], np\_city[:,1]) ->checking correlation coefficient

e.g. generating data

height = np.round(**np.random.normal**(1.75, 0.20, 5000), 2)

weight = np.round(**np.random.normal**(60.32, 15, 5000), 2)

np\_city = np.column\_stack((height, weight))