Python Pre Work:

**Questions for Mentor:**

* In data camp, sometimes when I try to run my code, it give an error but then I’ll submit and it will say it’s correct. Why? Same with creating scatter plots and line plots. I’ll try to run it and it won’t show the plot, but when I submit it will.

**Python Pre work:**

* NumPy = Numeric Python imported as np
  + Type of array – similar to lists
  + Many abilities like calculating over entire arrays
  + Assumes values are of a single type (int, string, etc). If they are not, they are changed to the same one – called type coercion
  + This is just a new kind of python type
* NumPy multi dimensional arrays
  + Numpy.ndarray = numpy package and n dimensional array
  + Shape is an attribute (**syntax:** np\_array2d.shape)
  + Pulling specific two dimensional indexes can be done like this – np\_2d[0][2] or like this – np\_2d[0, 2]
  + To select all values in a column or row, just put a colon (**syntax:** np\_2d[ : , 1] for all rows and np\_2d[2, : ] for all columns)
* NumPy summary statistics
  + Numpy has several summary statistics functions in it
    - Np.mean etc
    - Np.median
    - Np.corrcoef (correlation coefficient)
    - Np.std
* NumPy can also randomize data
* Matplotlib = data visualization tool in python
* Imported as plt
* Plt.Plot() function takes 2 arguments, x and y axes and makes a line plot
* Plt.Scatter() function takes 2 arguments and makes a scatter plot
* Plt.show() shows the plot
* Line plots are good when there is a time scale along horizontal axis. When trying to assess correlation between two variables, the scatter plot is much better
* Plt.hist = create a histogram. Takes 2 main arguments, x and # of bins
* Plt.Xlabel() and plt.ylabel() functions will label axes
* Customizations have to be before thte show function to show up
* Plt.title() give title
* Plt.yticks() can set the intervals on the axis
* Dictionaries
  + Uses curly brackets
  + Use colon to separate key and value
  + Keys should be immutable
  + Can have dictionary within dictionary (Europe = { ‘spain’: {‘capital’:’madrid’, ‘population’:59.5}, ‘italy’: {‘capital’:’rome’, ‘population’: 60.5}
* When to use list vs dictionary:
  + List – collection of values, order matter, select entire subsets
  + Dictionary – lookup table with unique keys
* Pandas
  + 2 dimensional lists that allow multiple data types
    - Numpy arrays are 2 dimensional lists but only allow one data type
  + Can create the table using dataframe to convert a dictionary
  + Can also import data from csv file for example
  + To pull specific columns out of data frame – use double square brackets instead to keep the DataFrame format
  + Can slice certain rows in a DataFrame using [[1:4]] formatting to pull rows 2-4
  + Loc = location based position in DataFrame
  + iloc = integer based position in DataFrame
* np.logical\_and(), np.logical\_or(), np.logical\_not() are qualifiers that work on numpy arrays
* iterating through dictionaries, use .items():
  + for key, value in dict.items():
* iterating through 2d numpy array to print the items, use np.ndinter():
  + for value in np.ndinter(my\_array):
* in pandas, you have to specify that you want to iterate over the row
  + use iterrows():
* can add a column to a DataFrame a couple different ways (below):
  + for lab, row in brics.iterrows() :
  + brics.loc[lab, "name\_length"] = len(row["country"])
  + brics["name\_length"] = brics["country"].apply(len)
* Random number generator
  + Np.random.rand() creates a pseudo-random number, but each time its run, the same random number is generated
    - As opposed to np.random() where truly random numbers are chosen each time
* Random walk
  + Path of gas molecules