In these problems there are a few functions and capabilities of Python that were not discussed in the notes or shown in the extra practice problems. This is to help students become comfortable with looking at code documentation. It is also to help these problems be more group-based.

1. Create a 3×3 matrix of random numbers. Find the eigenvalues and eigenvectors of this matrix.

2. On bCourses you will find a text file called `lecture_problem_2.txt` that contains a matrix of numbers. Create a new folder on your JupyterHub account and copy it into there. Your Python code for this problem should also go into this folder. Load this data and save it in your code (hint: there is a NumPy function for this!). Create functions to do the following:

   (a) Suppose that each column of the matrix is a vector. Write a function that normalizes each of these vectors and returns the vectors as a matrix. Show that your function works.

   (b) Write a function that takes a matrix of vectors, does the dot product between each pair of vectors, and prints out the results in an understandable way. Use this matrix of normalized vectors you created in part (a) to see if your function works.

3. You will have noticed that there is another text file on bCourses called `lecture_problem_3.csv`. Copy this into the folder that will contain the Python code to solve this problem. We are going to plot the data contained in this matrix. Read in the data contained in the file line by line and then reorganize it, I recommend using lists. Assume that the first column contains the x values and that the other columns are different sets of y values. If you open the text file you'll notice that the first row contains the labels for each data set. You will find the `split` and `strip` python functions useful. You will also need to know that doing something like `float("0.2456")` converts the string `"0.2456"` to a floating point number.

   Create the following plots:

   (a) Line plots with each column of y data on a separate line plot.

   (b) A line plot with all the data on one plot, make sure the lines are distinguishable.

   (c) Create a scatter plot of the first column and your choice of the other columns of data. If there is time, try changing the marker size, type, color, etc.

   Your plots should have axis labels, titles, and if multiple lines are on the same plot, there should be a legend. The titles and axis labels should not be hard-coded (you should not manually type these into your plotting code) you should instead have saved these labels in a useful way when you read in the data for this file.

Math Bootcamp Notes: Preparation for Graduate Physical Chemistry Courses ©2021 by Rachel Clune, Orion Cohen, Avishek Das, Dipti Jasrasaria, Elliot Rossomme is licensed under CC BY-NC 4.0. This license requires that reusers give credit to the creator. It allows reusers to distribute, remix, adapt, and build upon the material in any medium or format, for noncommercial purposes only.