**ENGR 102 [Skipping Lab 9 – MidTerm Renumbering]**

**Lab # 10B**

These labs are meant to familiarize you with two of the most commonly used engineering packages in Python namely 1) numpy and 2) matplotlib.

This individual assignment is meant to follow on to the team lab work. Be sure you have completed the team lab work, first. You are to individually create two programs and turn in the two programs.

**Activity #1: Matrix multiply and plot.**

1. You will create a program that repeatedly multiplies a matrix by a point and plots the data to the screen.

In this case, we will have a 2D point, (x,y). The point can be represented as a vector: . We can also define a 2x2 matrix, . Computing the product of *M* with *v* gives us a new point *v’*: . We can then use *v’* as the new point, multiply by matrix *M* again, and get another point, i.e. . This can go on indefinitely, creating a long sequence of points.

Create a program that uses numpy to create a matrix and a point. Specifically, you should begin with the point (1, 0), and the matrix: . Then, repeatedly multiply the matrix by the point to get a new point. You should repeat this between 150 and 250 times.

Note: the purpose of this part of the assignment is to get practice with numpy, so **you should use numpy for your operations**, even if you find it easier to perform this computation a different way.

1. Then, make your program **plot the data points, using matplotlib**. Be sure to **label the axes, and include a title**. Your **title should give a brief description of the shape that the points “trace” out**.

**Rubric: [35 points]**

[8 points] – Numpy used to represent matrix and points

[8 points] – Matrix multiplication is iterated on points via loop

[5 points] – Points are stored in a reasonable list structure as they are generated

[8 points] – Points are plotted in a reasonable graph that shows their progression

[6 points] – Graph has labeled axes and indicates that the points form a spiral

**Activity #2: Plotting multiple data points**

One of the most common ways that data can be stored to be loaded in a spreadsheet or other similar table is with a CSV (Comma Separated Value) file. These often are given a .csv extension. A csv file is a way of representing a table in a file. Each line represents a row of the table, and the cells in each column are separated by commas. CSV files can usually be read into spreadsheet programs (such as Excel), and most spreadsheets can output their data in a CSV format (sometimes called “comma delimited” format). You are going to practice writing and reading CSV data directly.

On the class website eCampus is a CSV file containing weather data from Coulter Field (in Bryan) for 3 years (**1 day is missing for some reason!**); the data was taken from Weather Underground (wunderground.com). There are different versions of the file for Windows and Mac; the only difference is whether the end of a line contains just a new-line character, or both a new-line and a carriage return (you don’t need to worry about the difference). Open the file in any text browser and you should see what it is. Note that the first line of the file contains the column headers explaining what each column is.

Download the file to your system, and write a program that will read in the file and plot the data in a set of graphs. **Using matplotlib**, you are to create the following **4 graphs**.

1. Create a **line graph** that shows both average temperature and pressure plotted over the period of time. Both should be plotted on the same graph, with **date on the x-axis**, and **different y axes** for the two different measurements.
2. Create a **histogram** showing amounts of precipitation. The x axis should cover a reasonable range of precipitation levels, and the y axis should show number of days that had precipitation in the specific range.
3. Create a **scatterplot** indicating relationships between average temperature and average dew point (one on each axis).
4. Create a **bar chart**, with one bar per month, showing the average temperature along with error bars indicating the high and low temperatures from that month.
   1. Note: You will want to create new lists of data, but you may find it useful to use the max/min/sum functions on lists.

Be sure to **include legends, labels, title, etc. on all charts**, and be sure your **scales/ranges and such are set appropriately to display all the data**.

**Rubric: [65 points]**

[8 points] – Weather data file is read in.

[10 points] – Average temperature and pressure are plotted on same graph

[10 points] – Histogram created with amount of precipitation

[10 points] – Scatterplot created to show relation between temperature and dew point

[5 points] – High/average/low temperature calculated per month

[10 points] – Bar chart created plotting high/average/low temperature

[12 points] – All charts are fully labeled with title, legends, axis labels, etc.