

General course learning outcomes:

- demonstrate the use of basic programming techniques in the construction of computer programs, including inputting data from a file.
- apply programming techniques to solve problems in engineering.
- complete a team programming assignment that ties together concepts learned in the class.

Activity 1: File Read and Write - to do in lab (team)

✓ Read data from a file

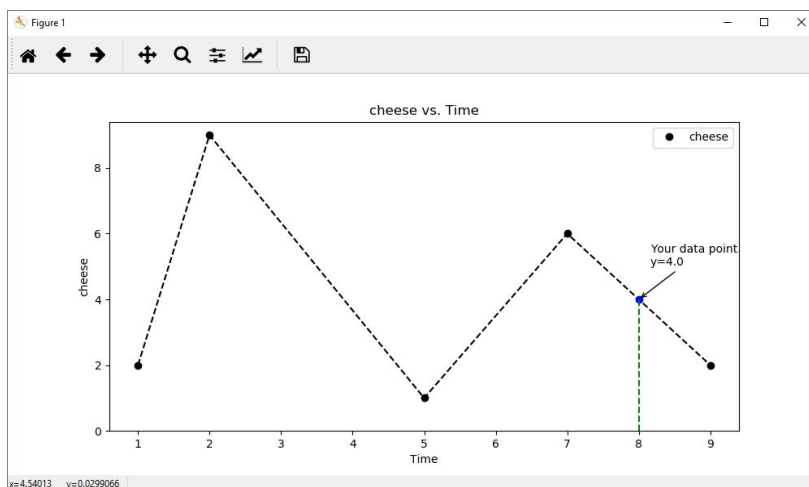
✓ Create a professional plots in Python

Continue working with last week's interpolation-extrapolation program. The second activity included the creation of an external file titled "nailedIt.txt" when the program is executed, based on the data points provided by the user. For this week's activity, we will create a new program that uses this data file.

- (Using the data file)* The program will prompt the user for a data file to read (*for testing, use 'nailedIt.txt', but the user should have the option to select a different file*).
- Ask the user for a value to interpolate or extrapolate using the data from the data file. Read in the data from the file, and perform the necessary interpolation or extrapolation. You may reuse any code from last week's activities to do so.
 - Print out the calculated interpolated/extrapolated value to the user.
 - Using matplotlib, create a scatter plot of the data input from the user and point out the location of the value calculated on the plot (*with annotation*).

For example: your program will ask the user for a time, and then output the y-value at that time, either interpolated or extrapolated from the data read from the file. It will also create and display a scatter plot of the data provided from the file, and indicate on the plot the location of the calculated point with annotated text.

Example plot for Activity 1



(continued, next page)

Activity 2: Weather - to do in lab (team)

☑ *Create Python program to read user input, perform necessary data reformatting, and print the expected output.*

A CSV file containing weather data from Coulter Field in Bryan is available on our eCampus course page. This file contains 3 years of data (minus one day, for some reason!) from Weather Underground (wunderground.com). There are two versions of the file, one for Windows and another for Mac; the only difference is new-line and carriage return formatting (you don't need to worry about this).

Download the file to your system; **do not rename the file** to use. Note that the first line of the file contains the column headers explaining what each column is.

Write a program that will open the .csv file, and give descriptive output of the following information to the PyCharm console:

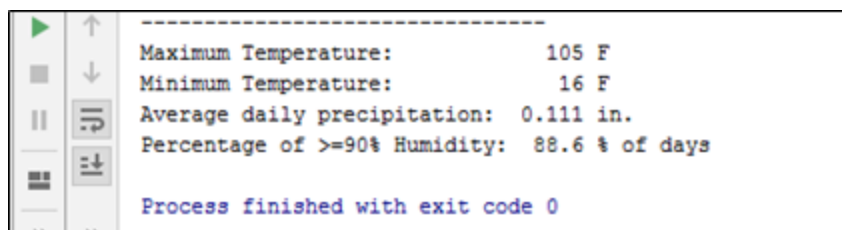
- The maximum and minimum temperature seen over the 3 year period
- The average daily precipitation
- Any two (2) other “interesting” data analysis questions of your choice*

For at least one, use the date information. Here are some ideas, but you can pick whatever you want:

- For some particular day, such as December 25, find the maximum and minimum temperatures reached among the 3 years of data.
- For some particular month, such as July 2015, calculate the average high temperature.
- Calculate how frequently the pressure increases from one day to another vs. how frequently it decreases.
- Calculate the percentage of days when the humidity was above some value, like 90%.
- Calculate the mean and standard deviation of precipitation levels.

The two analysis questions should be different from each other. For example, you should not find the min/max temperature for three different dates, or find the min/max pressure all for December 25.

Example output (note: yours requires two ‘interesting’ data results, this example only contains one):



```
-----  
Maximum Temperature:      105 F  
Minimum Temperature:      16 F  
Average daily precipitation: 0.111 in.  
Percentage of >=90% Humidity: 88.6 % of days  
  
Process finished with exit code 0
```