

CS150 - Lab Prep 11

Due: Friday Dec. 2, at the beginning of class

For our last (yay!) lab prep you're going to be writing some Python code to get you comfortable with the data that we'll be playing with for the lab this week. The majority of the lab will be in Matlab, however, there will be some data preprocessing in Python (that will be an extension of this lab prep).

1 The data



This week, we're going to be extending our weather aggregator program from Lab 9. Recall that the end result of the program was a file of the form:

```
<date> <hour> <temp>
<date> <hour> <temp>
...
```

where each entry is separated by a tab. For example, if we collected data for a few hours we might get:

```
11-9-2011 14 65
11-9-2011 15 66
```

11-9-2011 16 70
11-9-2011 17 68
11-9-2011 18 59

If you setup this program to run every hour for a particular zip code, you would eventually end up with a file spanning multiple days, with one entry per hour containing the temperature for that date/hour. I've posted two examples of files like this online for you to use for the lab at:

<http://www.cs.middlebury.edu/~dkauchak/classes/cs150/assignments/assign11/>

At this location there are two raw temperature files in the above format:

- `test.raw.txt`: A short file with just a few days of data that will be useful for checking your answers on.
- `wisc.raw.txt`: Real weather data aggregated for roughly 1 year from Wisconsin Rapids, WI obtained from:

<http://www.soils.wisc.edu/asigServlets/asos/SelectHourlyAsos.jsp>

(with a bit of post-processing by myself)

Create a folder for this assignment and download both of these files into that folder (right-click and then select "Save as...").

2 Matlab friendly data

Matlab is not very good at processing text files, so often when working with real data you will manipulate the data with another language (e.g. Python) to get it into a format that's easier for Matlab to handle. The goal of this week's lab will be to plot the temperature data as well as generate some statistics characterizing the weather. We'll do most of this processing in Matlab, however, we'll need to write a Python program to turn the above data into something more Matlab friendly.

To get you on your way to this:

Write a Python function that takes a raw weather filename as a parameter and reads through the file and returns a list containing the last temperature readings for each day.

For example, if you were to run this on `test.raw.txt` you would get:

```
[35, 41, 60]
```

3 The real problem (*OPTIONAL*)

If you'd like to get a bit ahead (but not required for the lab prep), the real problem that we're going to implement in Python is to take a raw temperature file and convert it into a file that Matlab can

read. Specifically, given a raw temperature file, generate a file where each *line* in the file represents a day's worth of temperatures (i.e. 24 entries, one for each hour). For example, for the test data the output would be something like:

```
40 37 35 35 34 36 37 42 45 47 50 52 53 53 52 54 51 50 45 43 41 37 36 35
33 30 29 28 30 30 32 40 41 42 43 43 43 44 45 45 44 42 40 39 40 40 41 41
41 42 42 44 45 50 51 55 65 72 80 86 93 101 102 100 95 80 77 70 67 63 60 60
```

There are three days worth of data. Each line/row is a day and each column is an hour (0-23).

As you read through the file, you can build up a list of the temperatures for a given day. When you reach the end of the day, print out (or better, write to a file) those values as a single line and then reset your list that was storing the temperatures for that day.

If you want to see the output from this processing, I've also included post-processed files in the data directory for this lab:

- `test.matrix.txt` contains the Python output from `test.raw.txt`.
- `wisc.matrix.txt` contains the Python output from the real data, `test.wisc.txt`.