

CSC 215 – Project 2

April 4, 2018

Due:

- Friday, May 4, 11:59pm
- MATLAB code file(s) and a screen capture of the plot, submitted via Canvas. There may be multiple code files to upload if you write functions for any portion of the assignment.

Objective

- Programming in MATLAB
- Processing data
- Working with large sets of data

Goals

- Download weather data
- Extract certain weather station data.
- Plot multiple sets of average temperature from each weather station

Steps

- Download all the 2016 and 2017 weather data from <https://www.ncdc.noaa.gov/orders/qclcd/> using the websave() function in MATLAB. For example, you will find the January 2016 data in a file called QCLCD201601.zip. QCLCD stands for Quality Controlled Local Climate Data.
- Unzip each monthly file using the unzip() function in MATLAB.
 - Each zip file contains several text files. We only need the daily file. For the January 2016 date, that file would be “201601daily.txt”
 - The station file identifies each reporting weather station and assigns each a unique WBAN (Weather Bureau Army Navy) id.
 - The daily file contains weather data for each day of the month from all the weather stations. We’ll be collecting the average temperature data from the “Tavg” column.
- For this project, we’ll be collecting data from only a few of the 2000+ reporting weather stations. The ids and the station names we’ll use for this project are below.
 - 12960 George Bush Airport, Houston Texas
 - 94728 Central Park, New York City, New York
 - 03017 Denver International Airport, Colorado
 - 12836 Key West International Airport, FL
 - 11641 San Juan, Puerto Rico
 - 23234 San Francisco International Airport, California
- There are multiple ways to process the daily weather files in MATLAB. Here is my suggestion:
 - Use fopen() and fclose() to open and close the various daily files.
 - Use fgets() to read one line of data from the daily weather data files.

- Use the `textscan()` function and comma as the delimiter to split up the data you just read.
- `textscan()` returns data as a cell. A cell can contain different types of data. `cell2mat()` can convert cell data to a matrix. If you need to convert string data to a number, use `str2num()`.
 - `strData = fgets (fileID);`
 - `cellData = textscan(strData, '%s', 'Delimiter', ',');`
 - `wbanID = cell2mat(cellData{1}(1)); % first field in the data`
 - `day = cell2mat(cellData{1}(2)); % second field in the data`
- If any data is missing (or non-numeric), you can set the temperature to -1
- Plot the temperatures for the specified weather stations with a legend.

Rubric

- **6 points** Download one of the zip files.
- **6 points** Use a loop to download all the zip files.
- **6 points** Unzip one of the downloaded zip files.
- **6 points** Use a loop to unzip all the downloaded files.
- **6 points** Only download and unzip the file if the daily text data files don't already exist on your computer.
- **20 points** Open all the data files and read each file until the end of file.
- **20 points** Extract only the weather stations' data we care about into MATLAB variables.
- **20 points** Plot all the weather stations' data with a legend.
- **10 points** Comment the major logic in the code (download, unzip, extract, plot).

Hints

- When specifying path names in Windows, remember that the `\` character starts an escape character such as `'\n'` (newline) and `'\t'` (tab). To specify the actual `\` character, it needs to be `\\` in the path name.
- If you don't have a lot of disk space on your computer, use a MATLAB command to delete all the data files except for the daily file after unzipping the data.
- You can use `sprintf()` to format data and save it in a string as opposed to `fprintf()` which displays it to the command windows.
- If you use `%02d` with `sprintf()`, it will result in a 2 digit, zero filled number. This can be useful to determine the month in the file name of the data file (for example: 01, 02, 03 through 12).