COMP 116
Programming Assignment #2
Plotting with NC water data
Due Friday, September 17th by 11:59pm

Assignment resources
As in Assignment 1, the data files you will need for this assignment are posted on Sakai. To find them, click on the "Resources" link in the left column. You will then see a list of all the files that are available for download; the images needed for this assignment are in the folder named "Assignment2Resources". You will need just one file: water08.mat. Download this and copy it to your working directory.

Loading data from a .mat file
A MAT-file (.mat) is a MATLAB-readable file used for saving and loading of MATLAB variables. (See http://www.mathworks.com/help/techdoc/ref/save.html and for more info http://www.mathworks.com/help/techdoc/ref/load.html.) In this case, the variables have already been created, so we just need to load them into our workspace. To do this, in MATLAB, change your active directory to the directory where you saved water08.mat, and type `load water08`. The results are shown below:

```
>> load water08
>> whos

  Name            Size              Bytes  Class     Attributes

  depth         276x2                4416  double
  hawgage       365x4               11680  double
  hawrain       365x2                5840  double
  rain          276x2                4416  double
```

(N.B. The `whos` command is used to display info on all the variables in the active workspace.)

From here, we see that 4 variables have been loaded: `depth`, a 276x2 matrix, `hawgage`, a 365x4 matrix, `hawrain`, a 365x2 matrix, and `rain`, a 276x2 matrix.

What data are we working with?
These variables contain data relating to water levels and rainfall in North Carolina. Specifically, we have:

Monthly Jordan and Falls Lake data (from Army Corps of Engineers):

- `depth`: a 276x2 matrix with depth of Jordan and Falls lakes for each month from Jan 1985 to Dec 2007, which is 23 years. Data that is not available is *NaN*.
- `rain`: a 276x2 matrix with total rainfall in inches for each month

Daily Haw river/Jordan Lake data (from USGS):

- `hawgage`: a 365x4 matrix of daily average river or lake height (ft) at Haw River, Bynum, and above & below the Jordan Lake Dam by Moncure . (These sites are listed upstream to downstream, but the gauges are not in that order.)
- `hawrain`: a 365x2 matrix of daily rainfall (in) measured at two rain gauges from 29 Aug 07-28 Aug 08.

What questions are we trying to answer?

These data can be used to answer a lot of interesting scientific questions, but in their current states (that is, big tables full of numbers), it is not easy to find these answers. In this assignment, we're going to use MATLABs analysis and plotting functionality to display these data in ways that are easier to understand. Specifically, you will need to answer the following questions:

Monthly Jordan and Falls Lake data

1. Plot a line graph of depths for both lakes. (Is there an obvious yearly cycle?)
2. The targets for Jordan and Falls lakes are 216ft and 251.5ft, respectively. For how many months was each lake over its target?
3. Plot the rain in August as a line graph over years for both lakes.
4. Compute the average height that Falls Lake is above its target for each month over the 23 years from 1985-2007, and display as bar chart with a bar for each month. Plot the line for 2007 in red on top of this bar chart.

Daily Haw river/Jordan Lake data

5. Determine how many days had more than 1 in of precipitation at the two sites in *hawrain*, and how many days had less than ¼ in.
6. Plot line graphs showing the cumulative amount of rain over the past year at both sites. Which of the two locations (1 or 2) received the most rain?
7. Determine the lowest height for each gauge, and create a matrix or vectors of adjusted heights by subtracting the corresponding lowest heights. Plot these adjust heights as a line graph.
8. Determine the maximum increase and maximum decrease in height from one day to the next for each of the four gauges in *hawgage*.

Useful MATLAB commands

Answering the questions presented above will require the use of several built-in MATLAB functions. You will probably use some (but not necessarily all) of the following functions. Note

that you can find more detailed information on all of these functions in your textbook (hint: the index is your friend) or in the MATLAB online documentation <http://www.mathworks.com/help/index.html>, which is fully searchable.

Array analyzing functions:
• arithmetic (+, -, *, /, .*, ./)
• relational (>, <, >=, <=, ==, !=)
• `mean(A)` - computes the mean (average) of the values in the **vector** A
• `sum(A)` - computes the sum of the values in the **vector** A
• `min(A)` - returns the smallest value in the **vector** A
• `max(A)` - returns the largest value in the **vector** A
• `diff(A)` - generates a new vector containing the differences between adjacent elements of A
• `cumsum(A)` - generates a new vector containing the running total of elements of A

Shape changing functions:
• : (colon) - get a *slice* of a matrix or *stride* through a matrix
http://www.mathworks.com/help/techdoc/ref/colon.html
• ‘ (single-quote) - returns the transpose of the matrix (for example, `A'` is the transpose of `A`)

Plotting functions:
• `bar(...)` - used to create a bar graph
• `plot(...)` - used to create a line or X-Y plot
• `hist(...)` - used to create a histogram plot
• `title(...)` - adds a title to a graph
• `xlabel(...)` - used to add a label on the X-axis of a graph
• `ylabel(...)` - used to add a label on the Y-axis of a graph
• `legend(...)` - used to add a legend to a graph

Note: **ALL** graphs should be given meaningful titles and axis labels. You can use the above functions to do so. A graph that is not appropriately titled and labeled **will not receive full credit.**

Vectors vs. matrices

Note that many of these functions assume the input is a vector, not a matrix. Applying them to a matrix will usually still work, but might not give the result you expect. (Try taking the `mean()` of a matrix and look at the result.)

You can use colon-indexing (a/k/a *slicing*) to pull single rows or columns out of a matrix and operate on them as vectors. For example, `gage2 = hawgages(:,2)` would give you a vector containing only the data from the second gauge.

You can also put vectors together to build matrices using []. For example, if you have column vectors `a` and `b`, you can use them to make a new m-by-2 matrix `M` with `M = [a b]`.

Note that MATLAB looks at **columns** of a data matrix for `plot()`, `mean()`, `sum()`, etc., so transpose(‘) can be useful.

Matrix striding

You can also use the colon operator to *stride* through a matrix. For example, we saw in lecture that `mat = [2:2:20]` creates a vector containing the positive even numbers from up to 20. This syntax `[start:strideLength:end]` can also be used to retrieve values from an array; for example `mat2 = mat(2:2:end)` creates a new matrix containing every other element of `mat` (which happens to be the multiples of 4 up to 20).

Relational operators

MATLAB allows you to test a relation against an entire matrix with a single command. (For folks who have had some previous experience with programming, this means that you don’t need to use a loop to test each element in turn.) For example, if we have

```
mat = (-5:5)

mat =

  -5    -4    -3    -2    -1     0     1     2     3     4     5
```

then we can test if each value is greater than zero with just one command:

```
>> mat > 0

ans =

   0     0     0     0     0     0     1     1     1     1     1
```

So there is a 1 in `ans` if the test was true for that element, and a 0 if it was false. (Think how you could use this in combination with other MATLAB functions to solve some of the problems in this assignment.)

More on plotting

Here is the essential line from the help for `plot`:

> `plot(Y)` plots the columns of Y versus their index if Y is a real number.

You can choose the color, marker, line style, etc. for your plots. As just one of many possibilities, if you wanted to display the vector `lineData` as a red line in a line plot, you would use the command `plot(lineData, 'r')`. For more details, type `help LineSpec`.

`plot()` (and `bar()`, and `hist()` , etc.) erases the screen unless you have said `hold on`. Thus, to put more than one line on the same figure, either put them as columns in a matrix and plot once, or type `hold on` and plot several times. Type `hold off` to cancel a hold, or `clf` to clear the figure.

Please use `title`, `xlabel`, and `ylabel` to add titles and axes labels to each plot. When appropriate, you can use `legend('name1', 'name2', …)` to add a legend.

For example, to plot and label the height for all four gauges:

```
>> plot(hawgage)
>> legend('gauge1', 'gauge2', 'gauge3', 'gauge3')
>> title('Adjusted water height')
>> ylabel('Adjusted height (ft)')
>> xlabel('Day since 28 Aug 07')
```

Data sources

USGS Surface-Water Data for North Carolina, Daily Data http://waterdata.usgs.gov/nc/nwis/sw
US Army Corps of Eng., Wilmington Dist. Water Manag. Unit http://epec.saw.usace.army.mil

Turning in the assignment

As with Assignment 1, Publish your completed script file (by typing `publish Assignment2.m;` (which generates an HTML file and images) or `publish(Assignment2.m, 'pdf');` (which generates a PDF) in the MATLAB command window. Create a folder inside your shared Dropbox folder named Assignment2. Copy the published html directory or pdf file into that Dropbox folder.

Required Components: [100 pts]

The 'Question X' refers to the questions in the "What questions are we trying to answer?" section.

(Note that all code for each of these components should be included in your published script.)

1. Question 1 above: Create a single line plot showing depths for both Jordan and Falls lakes. [10 pts]

2. Question 2: Compute how many months each lake was over its target. (2 numbers, 1 for each lake) [4 pts]

3. Question 3: Create a line plot showing August rain levels. [10 pts]

4. Question 4: Create the combined bar and line plot discussed above. [25 pts]

5. Question 5a: Determine how many days had more than 1" of precipitation at the two sites in `hawrain`. [4 pts]

6. Question 5b: Determine how many days had less than ¼" of precipitation at the two sites in `hawrain`. [4 pts]

7. Question 6a: Plot line graphs showing the cumulative amount of rain over the past year at both sites. [15 pts]

8. Question 6b: Which of the two locations (1 or 2) received the most rain? [5]

9. Question 7: Create a line graph of the adjusted heights. [15 pts]

10. Question 8a: Determine the maximum increase in height from one day to the next for each of the four gauges in *hawgage*. [4 pts]

11. Question 8b: Determine the maximum increase in height from one day to the next for each of the four gauges in *hawgage*. [4 pts]