Statistics and Data Analysis in MATLAB
Kendrick Kay, kendrick.kay@wustl.edu

Homework 1: MATLAB Basics

To complete this assignment, prepare a MATLAB script called `homework1.m` along with any necessary accompanying function .m files. Then, run the MATLAB `publish` command (e.g. `publish('homework1.m')`) to run the script and generate HTML output showing the results. Turn in a print-out of the HTML output (e.g. from your web browser) and also a print-out of any function .m files that you write.

Hint: In your script file, place `%%` on a line by itself at each point where you want the HTML output to show figures and command-window text. Please note that your code should be commented (where necessary), including documentation of any functions that you write.

Problem 0. Download the .mat file at http://kendrickkay.net/psych5007/Homework1.mat (you will need this file to complete some of the problems below).

Problem 1. Generate two sets of 1,000 random numbers that are drawn from the normal distribution (Hint: `randn`). Create a scatter plot of the first set of numbers against the second set of numbers. Label the two axes as "Condition A" and "Condition B". Show the resulting figure.

Problem 2. The `data1` variable in Homework1.mat is a matrix of dimensions 2 x 20. Sum the numbers in each row and then compute the ratio between the two sums (i.e. sum of row 1 divided by sum of row 2). Store the answer in a variable called `result` and echo the value of this variable to the screen.

Problem 3. The `timeA`, `valsA`, `timeB`, `valsB` variables in Homework1.mat represent time-series data for two signals, A and B. The time points at which A was measured are given in `timeA`, and the corresponding measurements are given in `valsA`. (The same goes for B, `timeB`, and `valsB`.) Create a figure in which both signals are plotted as lines. Use a red line for A and a green line for B. Label the two axes and add a title and legend to the figure. Show the resulting figure.

Problem 4. Create a variable containing the matrix shown to the right. Then, append a new row to the matrix that consists of all 1s. Then, change the top-right element of the matrix to 0. Finally, extract the last column of the matrix and reshape it to be a row vector. Store this vector in a variable called `result` and echo the value of this variable to the screen.

```
3.0000   -1.0000    1.5000
2.0000         0         0
1.0000    1.0000    1.0000
```

Problem 5. Write a function called `checkpositive.m` that accepts a 2D matrix as input and outputs a row vector. The row vector should contain as many elements as there are rows in the 2D matrix, and each element should indicate whether the corresponding row in the 2D matrix contains all positive numbers (use 0 to indicate no and 1 to indicate yes). The function should also report the results to the command window; the report should look something like this:

```
Row 1: no
Row 2: yes
...etc.
```

Finally, in your `homework1.m` script, issue the following commands:

```
result = checkpositive([4 1 1; -1 0 0; 0 0 0; .1 .2 .1]);
result
%%
```