Week 1 Exercises

- Due date: Tues, Sept 16 by 11:59pm EST. If you need extra time, just let me know. Remember these assignments are optional, but I strongly recommend completing them for practice.
- Complete all exercises in a single script named "lastname_firstname_week1_hw.m". Each question should be in a separate section using %% and an appropriate header. You don't need to make separate sections for sub-questions (e.g. 1a-e). If you're not sure how to use a function, use the help function in MATLAB (`help fxname`)
- **REMEMBER GOOD CODING PRACTICES**: start each section with the problem number, use comments to explain what you're doing, give your variables descriptive names, and use whitespace.
- If you get stuck or have questions, please don't hesitate to email me at mlnguyen@princeton.edu. I'll provide written feedback on assignment by end of the week.

In-class exercise from June 12

1. [From in-class exercise. Can skip if you already did it.] Create a 3x2 matrix A of 3s. Create a second 3x2 matrix B with the numbers 1-6. Add the two matrices together. Multiple the two matrices by elements.
    a. Try multiplying matrix A by matrix B. Why didn't it work? Use transpose to modify one of the matrices and try again.
    b. Create a 2x2 matrix of ones, a 2x2 matrix of twos, and a 2x2 matrix of threes. Concatenate the three matrices in order vertically and then horizontally using square brackets and commas/semicolons. Repeat using the horzcat and vertcat functions.
    c. Concatenate the matrices from (3) in a 2x2x3 matrix

2. Look up the documentation for the functions rand() and randi().
    a. Use the appropriate function to generate a 3x4 matrix of random integers that vary from 0-9. Display the matrix
    b. Display all the values in the odd numbered rows
    c. Replace the value at index 10 with 100. Replace all the values in the last column with 20s.
    d. Use the appropriate function and matrix operations to create a 1x4 vector of 99s. Concatenate it to the bottom of the matrix from (c). You resulting matrix should now be 4x4, with the last row 99s.
    e. Use the appropriate function to create a 4x4 matrix of random numbers ranging from 0-1 inclusive. Concatenate this matrix with the matric from (d) to create a 4x4x2 matrix. Print this final matrix. Print the size of this matrix.

3. In a visual learning study, five subjects were trained to detect the orientation (left or right) of a Gabor patch (examples) presented in the periphery of the visual field. The subjects' accuracy was measured before and after the training. The pre- and post-

training scores for each subject are as follows: sub1, 0.51, .72; sub2, 0.59, .83; sub3, 0.048, 0.51; sub4, 0.58, 0.69; sub5, 0.49, 0.65.

a. Create an array containing the pre-test scores for all the subjects (in order from sub1 to sub5). Create a second array with the post-test scores.
b. Calculate the improvement in accuracy from pre- to post-test.
c. Concatenate the pre and post-test score arrays into a 2x5 matrix. The first row should be pre-test scores and the second row post-test scores. Each column is a subject. Get the dimensions of the new matrix.
d. Look up the documentation for the mean() function. Calculate the mean pre- and post-test score across all the subjects. Calculate the mean of the pre- and post-test score for each subject.

4. In this last exercise, we will load and display slices from an MRI anatomical scan. Download the file, MNI152_T1_2mm_brain.mat from the class GitHub repo. Save it in your homework directory. This is a standardized anatomical brain created by the Montreal Neurological Institute (MNI) by averaging the anatomies of 152 subjects. "T1" refers to the image type, and "2mm" is the image spatial resolution. ".mat" indicates that it is a MATLAB data file.
    a. Load the file into your workspace in MATLAB by typing:

        ```
        load('MNI152_T1_2mm_brain.mat').
        ```

        If you get an error, make sure you've spelled it correctly. Also make sure that your working directory (type: pwd) is the same as the directory that you downloaded this file to.
    b. In the command line, type:

        ```
        whos
        ```

        You should see a new variable called "nifti_img." This is the brain image you just loaded. NIFTI is the most common file type for MRI data.
    c. Print the size of nifty_img. How many dimensions does nifty_img have and what are their values?
        i. The first dimension, x, goes from left to right in the brain image
        ii. The second dimension, y, goes posterior to anterior
        iii. The third dimension, z, goes inferior to superior
        iv. If you don't know these terms, check out this link
    d. Visualize a slice from each anatomical plane: coronal, sagittal, and axial (or horizontal). If you don't know these terms, check out this link. First, select a slice in any dimension (any slice) and then save it to a variable. The slice should be 2-dimensional. Check and print the size.
    e. Use this code to visualize the slice:

        ```
        figure; colormap gray
        ```

```
imagesc(flipud(your_slice_var'))
```

Repeat this for each coronal, sagittal, and axial slices. Try a bunch of different indexes to take a look at the brain anatomy. Once you are satisfied exploring, choose slices from the center of the brain for the HW assignment.

5. Rate the difficulty of this assignment from 1-5 where 1 = very easy, 5 = very hard
6. Rate the usefulness for this assignment for learning from 1-5 where 1 = not at all useful, 5 = very useful
7. Approximately how long did this assignment take you?