

iPBR Assignment 2

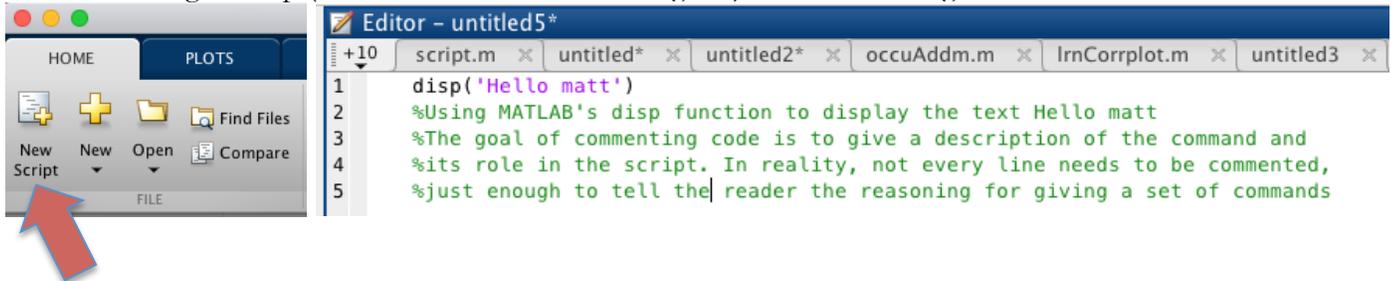
Introduction to MATLAB part I

Due June 28th, 2017 @ 7pm

(Problem sets turned in after this deadline will not receive feedback)

Email finished problem sets and questions to: matthewsmith01@g.harvard.edu

For this assignment please write everything inside the script editor. Save each problem as a separate script with your name and the problem number, ie: mattSmith_P1, mattSmith_P2. As a general tip, it's helpful to outline each logical step (write them out or make a diagram) before executing commands in MATLAB.



Click on the home tab then click the New Script icon to open a new script window. Through this assignment please add comments to your code. Comments are notes that you can write to explain your code to another user. To add a comment type a `%`, followed by whatever text you'd like. An example is shown above. Any text to the right of the `%` will not be executed by MATLAB. Throughout this problem set there are parts that are labeled as Challenge. You should attempt these problems, however, they do not need to be completely solved. The complete answer key will be posted on the 28th.

Problem 1

Create a new script and save it as "yourName_P1". Execute the commands listed below, and use comments to explain what each command is doing.

```
1) M1 = [ 3 5 1; 2 8 4];
2) m1 = [ 80 75 6; 100 34 8];
3) m1 == M1
4) matrix3 = [ 89 4 9; 1 -80 503];
5) matrix3(1,:) = m1;
6) store{2,1} = matrix3
7) store{2,2} = M1
8) store{1,1} = 'Data of matrix3';
9) store{1,2} = 'Data of M1'
10) store{2,2}(1,:)
11) for i = 1 : 0.75 : 75
    save1(cnt) = i;
end
```

Challenge:

```
n = 1;
for i = 1:800
    for k = 800:-1:1
        embedded(i,k) = n;
        n = n+1;
    end
end
```

Download the pSet2data.m file located on the course website under the link to this problem set. Double-click on the downloaded file to open and import the files to MATLAB.

Question 2

Suppose you are a young and talented scientist who is interested in behavioral neurobiology of humans. It's your first day in a new lab and your PI throws some fMRI data at you. They had recorded activity data for 300 seconds at 10Hz (10frames/sec). Your PI suspects that there is a single spike in neural activity somewhere in the data set, and needs you to tell them when this spike occurs. They tell you that spike resulted in a peak at least 10x that of baseline, so simply finding the position of the maximum value in this array will correspond to the time of the spike. Your PI doesn't know how to code, so they have been looking at each time-point in this dataset manually. They wrote a basic algorithm for finding the spike, but need your expertise in coding to implement it in MATLAB.

PI Algorithm:

- i) Set LargestValue equal to 0
- ii) Move to element 1 in the dataset
- iii) Is element 1 greater than LargestValue
- iv) If so, then store position of element 1 as LargestValue
- v) Else, do not change LargestValue
- vi) Move to next element.
- vii) Is element 2 greater than LargestValue
- viii) If so, then store position of element 2 as LargestValue
- ix) Else, do not change LargestValue
- x) Repeat steps ii-v for every element in the array

Your PI seems to think their algorithm is correct, however, it would be much too time consuming to do this manually. They come to you and beg for you to execute this algorithm (or something like it) in MATLAB.

Please refrain from using MATLAB's internal functions for this exercise

- 1) What are the dimensions of p2_vector?
- This variable should have been imported into MATLAB's workspace after you opened the p2_data.mat file.
- 2) Write a for-loop to index through every position in the p2_vector.
- 3) Inside your for-loop write a conditional statement to find the maximum value of the p2_vector.
*hint: one way of accomplishing this is to use your for-loop to search through every position in your vector. As you're proceeding through your for-loop use a conditional statement to compare each value to a variable called LargestValue. If the current value is greater than the LargestValue, then replace the value of LargestValue with the element at the current position.
- 4) Write a conditional statement to find the time-point (index) for when the LargestValue occurs in p2_vector, and save this position as, spikeTime. This will be the time at which the spike occurs.
- 5) Execute the code and take a screenshot of the output:
`figure; plot(p1_vector(spikeTime-25:spikeTime+25))`

Question 3 (Challenge)

This question is a continuation of Question 3 from problem set 1. The details are listed below:

Suppose you are interested in mouse behavior, and you want to know what odors scare mice. When mice are afraid, they stop moving and freeze. To test which odors are fear inducing, you record a movie of the mouse walking around its cage, and randomly release one of ten odors. You record the mouse for 2 hours, and expose the mouse to a randomly selected odor once every 2 minutes (for 30 seconds).

Somebody else's program automatically records the results in two variables. The first variable contains the coordinates of the mouse, recorded every second. The second variable records which odor is being released (stored as the numbers 1-10, with a number 0 if no odor is being released), also recorded every second. Your goal is to create a bar graph that plots the average speed of the mouse when each odor is exposed.

Create a new script titled "yourName_P3". Write a script to convert the two variables you've received into a bar graph.

- a. What type of variable is odorExp?
- b. What data is stored in odorExp?
- c. Identify all time-points that have odor 1 presented.
*hint: one method is to write a for-loop to search through each position of the odor array. In the for-loop you'll provide a conditional statement: if the odor presented == odor 1, then store that time point in an array. Your output should be an array containing all the time points in which odor 1 was presented.
- d. Find the speeds of the mouse when odor 1 is presented. To do this: use the array created in part c to index the speed array (in odorExp) and return only speeds when odor 1 is presented.
- e. Use the mean function to find the average value of the array created in part d. This single value will be the mean speed of the mouse when odor 1 is presented.
The mean function in MATLAB works by typing: mean(yourArray). You're telling MATLAB to use the mean function on the variable, yourArray.
- f. Perform steps C-E for all odor values (0-10) and produce an array that contains the mean speed for each odor, and title it AvgSpeed. This array should have 11 elements, one value for each odor.
- g. Execute the command:
`figure;bar(AvgSpeed)`

Please don't hesitate to send any questions to matthewsmith01@g.harvard.edu.

Attend the review session on Monday @7pm for assistance in person.

Good luck!