

Review: Week 2

1. What does each line in the following snippet do?

```
y = randi(100, 1, 20)
siz = size(y)
z = y.^2
w = y > 50;
flattened_y = y(:)
```
2. Create your own **for** loop that includes an if/else statement. Use the following functions:
 - a. nanmean()
 - b. normrnd()
 - c. disp()
 - d. clock
3. Making arrays: (spoiler: you shouldn't be writing every one of these values out...)
 - a. Create a variable C that is an array of consecutive values from -5 to 5.
 - b. Create a variable D that is an array of values from -10 to 10 in increments of 2 (i.e., [-10, -8, -6, ..., 0, 2, 4, ..., 8, 10]).
4. Indexing arrays: using the variable D that you created in the question 12, how would you do the following?:
 - a. Return the value in the 6th position of array D.
 - b. Return the values in positions 4, 5, and 6 of array D.
 - c. Return the values in positions 1, 3, 4, and 7 of array D.
 - d. Return all values in D that are negative. (Hint: logical operators are your friend)
5. Create your own **while** loop that includes an if/else statement. Use the following functions:
 - a. randi()
 - b. corr()
 - c. disp()
6. How do you get a specific column of data from a table data type? Then, how do you get a specific row entry from that column of data?
7. You have the following matrix of numbers under the variable name "data":
data =

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

 - a. Manually, calculate the result of mean(data, 1) and mean(data, 2).
 - b. What is the result of mean(data(:))?
 - c. Replace each numeric entry with the way you would index the data variable (e.g., data(...)).

Week 3: Creating your own functions

A function is a way to package several different actions that need to happen together and execute all of them in a single line of code. For example, recall the “TA marking” scenario from Worksheet 2 and imagine that there are 2 midterms and 1 final exam to mark.

We could “execute” these lines of code each time the students finish writing their exams:

```
while there is paper left in the stack
    if the student is in group 1
        > grade their exam
        > enter the marks into a database
    else (they are in group 2)
        > put aside for the other TA
    > check the number of papers left in the stack
```

Or, we could bundle these actions into a single function called “mark_exams” where we feed in the papers, and two lists detailing which student is in which group.

```
grades = mark_exams(papers, group_1, group_2)
```

Then, this “mark_exams” function can be called whenever we need to mark something, kind of like an auto-pilot! This would be a useful “mode” to have as it allows you to generalize across the variables instead of “hard coding” in, for example, the number of students or what groups they belong to. Instead, these variables can be inputted as parameters that change.

Week 3 Assignment

Consider the while loop we created last week which only stopped when participants reached a certain score threshold. Now, imagine that we want to test the effect of the threshold on the number of times it must be repeated until that threshold is met.

Part 1

- How would you implement this using a for loop? (Hint: you’ll need to decide on the thresholds you want to test and figure out how to “keep” the number of while loop repetitions that happened for each threshold.)
- How many lines of code are required to run this analysis?

Part 2

In the Command Window, look at the *source code* of the mean function by typing:
open mean

- Look at the first line of the mean code that opens up. Label each component of that first line with what exactly it represents.
- Write out a generalized version of what goes on the first line of any function.

Consider now that we want to turn the memory test thresholding task into a function.

- What are the input(s) and output(s) of our function?
- Based on your inputs and outputs, what should the first line of the function script be?

- e. Open up a new script (this will be your function script). Type in what you came up with for d., and then save the script somewhere in your MATLAB path. What does the name that you're saving the *.m file under need to be?
- f. Paste in the MATLAB code from last week that we used to run the memory test. What happens if you don't change anything from this script? What warnings come up?
- g. What changes do you need to make in the script to use the inputs you specified and output the results?

Part 3

Come up with your own function using a dataset from MATLAB!