**MATLAB For Scientists**

**HW4**

**Instructions**

* Complete each question below in an **individual** script or function. Each individual script should be named according to: HW#Q#\_script.m, or HW#Q#\_fun.m where <#> indicates the corresponding HW number or question. These are the scripts that will be considered for grading.
* Submit all files (**including** the provided data, but **excluding** any tables/files that are saved as a result of the HW question) as a compressed .zip folder with the name **<HW#\_studentID.zip>.**
* Each script **must** run entirely through without error. If you did not finish part of a question, comment it out so it runs.
* Suppress all intermediate outputs other than your answer to the question; only the answer should display on the command line.
* Absolutely no hard coding beyond the minimum specified.
* **Everything must be commented. Uncommented codes get zero credit!**

**Problems**

**Question 1 (2 pt)**

Write a script that performs the following actions:

1. Generate a random vector of integers between 0 and 10 of length 5. If at least one of the elements in the random vector is equal to 10, display the sentence: At least 1 number equals 10!.
2. Generate a random vector of integers between 0 and 10 of length 5. If at least one of the elements in the random vector is equal to 10, display the sentence: At least 1 number equals 10!. If none of the numbers are equal to 10, but there is at least one element equal to 5, display the sentence: At least 1 number equals 5!. If none of the elements are equal to 10 or 5, display the sentence: No number equals 10 or 5!.
3. Write a for loop that iterates 20 times such that each loop generates a different random vector of integers between 0 and 10 of length 5. During each loop, display the following sentence: “There are X 10’s generated on loop # Q”, where X is the exact number of 10’s in the vector, and Q is the current loop iteration.
4. Generate one random vector of integers between 0 and 10 of length 12. Loop through all of the numbers to calculate the number of 10’s there are. Every time you reach a number 10, display ‘Number 10 identified’. However, if you reach a number 3, display ‘Number 3 identified’, and break the loop.

**Question 2 (2 pt)**

rem(a,b) is a MATLAB function that returns the remainder of a when divided by b. Write a function that takes in an integer, and displays either ‘This number is even’, or ‘This number is odd’, depending on whether the number is even or odd, using rem. Call this function from a script using an even number and an odd number to show that it works.

**Question 3 (2 pt)**

Write a script that loops through all integers from 0 to 100, and creates a new matrix consisting of all integers that are multiples of 10 (e.g., 10, 20, 30, etc.).

**Question 4 (2 pt)**

A pilot study tested whether a new medication lowers a patient’s average heart rate. For 100 patients, average heart rates were measured before (hr\_before), and 14 days after (hr\_after), treatment. We want to know whether the average heart rates increased or decreased after the treatment for each patient. Using the data stored in HW4Q4\_HRtrial\_single.mat, write a for loop to generate a matrix that stores a 1 if the treatment was effective (i.e., lowered heart rate), a 2 if the treatment had no effect, and 0 if the treatment was detrimental (i.e., increased heart rate). Display the total number of effective treatments, and total number of treatments that had no effect, using a complete sentence.

**Question 5 (plotting is NOT on the exam) (4 pt)**

Using SOL147.mat, SOL147\_rep2.mat, and SOL147\_rep3.mat data (same as last week), write a script that does that following:

* Contains two nested for loops (one for loop inside of a second for loop) that iterates through each drug concentration and day. This results in two iteration variables (e.g., one variable looping through the rows of the matrix and a second looping through the columns) that you can use to index a matrix according to both its row and column.
* Collects a new matrix called max\_survival that stores the maximum survival value for that particular day/concentration across all 3 replicates; to find the maximum value, you must use if statements, not a MATLAB provided function. Note this matrix should be the same size as each individual survival matrix.
* Create a figure that has 4 subplots. On each one, plot survival at a different drug concentration as a function of days (each subplot should have 6 lines: 3 controls, and 3 treatments for each replicate). Plot the controls in black, and each replicate in red. Use a different line style to distinguish between replicates, and keep these line styles consistent for each subplot. Format plots according to best practices (titles, maintaining axis limits, line thickness, font size). The plot should look like this, using the line styles of your choice:

**Question 6 (tables are NOT on the exam) (4 pt)**

HW4Q6\_COVID\_cases.xlsx contains the number of COVID-19 cases reported per state in the US over the previous summer. We are interested in cases in the northeast from the following states: NY, NJ, CT, and MA. Write a script that does the following:

1. Generates 4 tables (one for each state listed above, where each contains only the data from each respective state) and saves them as 4 separate .xlsx files.
2. Calculates the total number of cases reported per state, from **every** state with data in the original excel file, over the time period recorded in this dataset. Note that “NaN” is a MATLAB variable that indicates “non-existent”; these numbers will interfere with “sum” commands, and need to be excluded using either indexing to identify all numbers greater than 0, or the function isnan to logically identify NaN’s.

*Hint: unique(X) will give you all the unique entries in array X.*