PSYC 6273 – Computer programming for experimental psychology
Fall 2019

Problem set 1

For the following problems, use the function and variable names suggested, and include sufficient help text and comments. Indicate your name in a comment just after the help text in each file.

Email your solutions to me (rfm@yorku.ca) in a single .zip or .tar file named with your last name in lowercase, e.g., murray.zip.

1. (a) Use MATLAB's random number generators to create the following variables that contain simulated data from 100 trials of a psychological experiment.

`stimnum` is the stimulus number on each trial. This variable is a 100 x 1 matrix of random integers, drawn randomly and with equal probability from the numbers 1, 2, and 3.

`rt` is the subject's reaction time on each trial. It is a 100 x 1 matrix of real numbers. Each number is $0.1 + N_2$, where N is a sample from the normal distribution with mean zero and standard deviation one.

`correct` indicates whether the subject gave the correct response on each trial, encoded as 0 = incorrect, 1 = correct. It is a 100 x 1 matrix of random numbers, equal to 0 with probability 0.2, and equal to 1 with probability 0.8.

(b) Find the mean reaction time on trials where stimulus 1 was shown.

(c) Find the mean reaction time on trials where stimulus 2 was shown and the observer gave the correct response.

(d) Make a more realistic version of the simulated reaction times `rt`: sample the reaction times from $0.1 + N_2$ on trials where the observer gives the incorrect response, and from $0.3 + N_2$ on trials where the observer gives the correct response.

2. Write a function `randnc.m` that accepts two arguments, `m` and `n`, and returns an `m` x `n` matrix of random numbers that are drawn from the standard normal distribution, except that none are more than two standard deviations away from the mean.

Do not use MATLAB's `truncate` function to solve this problem.

Note that although the population mean of the standard normal distribution is zero and the population standard deviation is one, the mean and standard deviation of any given sample do not have to be (and in fact are usually not) zero and one. If you are unclear about what this means, feel free to talk to me.

3. The bubblesort algorithm is a sorting algorithm that works as follows. Suppose you have a 1 x 10 matrix of random numbers that you wish to sort from smallest to largest:

1 7 3 8 5 9 2 0 4 6

Check the first pair of numbers (here, 1 and 7), and see if they are in the right order, i.e., smallest first. If they are in the right order, then leave them as they are, and if they are in the wrong order, then switch them.

Now do the same thing to the second pair of numbers (here, 7 and 3), i.e., if they are in the right order, leave them, and if they are in the wrong order, switch them.

Continue checking and possibly switching each remaining pair of numbers, until you reach the end of the list. After this pass through the list, the list will be closer to being sorted than it was before.

Now repeat all the previous steps, and keep repeating them until you make a full pass through the list without switching any pairs. Once you make a full pass without switching any pairs, the whole list must be sorted, and you are done.

Write a function called bubblesort.m that takes one input argument that is a matrix of numbers. The function uses the bubblesort algorithm and returns one output argument, which is a matrix that contains the numbers sorted from smallest to largest.

Due October 24, 2019