COGS 119 – Fall 2014

Assignment 2 – Due Thu October 23 by 7 pm

Remember to comment your m files using % or Matlab → Text → Comment or using %

help is your friend!

Submit the indicated files via email to cogs119a@gmail.com.

Part 1

Write a small program that interacts with the user using string manipulation and conditional statements.

In a new m-file called YourLastName_hw2part1.m, write a program that does the following:

- Asks the user to enter their first name, saves this as the variable *fname* (Hint: see help **input**).

- Asks the user to enter their last name, saves this as the variable *surname* (Hint: see help **input**).

- Creates a new variable *fullname* that contains the first name and the surname with a space in between. You will need to calculate the number of non-space characters in this string. You may assume there are no spaces within *fname* and *surname*.

- Displays on the screen one of the following (Hint: see help **disp**).

  If the first and last names have no non-space characters in total, then display:

  So you are anonymous, OK.

  If the first and last names have 20 or more non-space characters in total, then display the following message on the screen:

  Whoa, that is a long name!

  If the first and last names have less than 20 non-space characters, then display the following message:

  OK then.

- On a new line, displays on the screen the following message:

  Your first and last names have <N> characters in total.

  Note that N should be the number of non-space characters you calculated to decide which message to display above. Hint: try help **num2str**).

Hint 1: You will need to use at least one conditional statement. See if you can get it working with one conditional statement. Type **help if** in the command window.

Test the program with lots of different inputs to make sure it's behaving correctly. Use semicolons so that the computations that should not be displayed on the screen.

Submit the m file YourLastName_hw2part1.m (e.g., jones_hw2part2.m) when you are done.

Part 2A

Write an m file called YourLastName_RTdata.m (e.g., Jones_RTdata.m) that creates a structure called MyData, which holds the information of subjects in an experiment. The struct should hold the data of a mini experiment that has 2 conditions: Assume that in each trial of the experiment, the subject responds with a key press with their right, or with their left hand, and the experimenter records the reaction times (RT) for 10 trials.

Please pay attention to the following points:

Use a structure called MyData that contains subfields Subject_Number, Test_Time, and Data.

Subject_Number is the index of the subject and will be input by the user (e.g., 1, 101, etc).

Test_Time should contain the current date and time (Hint: Read the help on **now**).

Data is itself a struct that contains the reaction time data and is organized as follows:

Trial_Number (1 to 10)

Trial_Type (1 for Right, 2 for Left for each trial)

RT (reaction time recorded)

You should make up this data as follows:

Trial_Number is obvious. There should be 5 trials expecting 'Right' and 5 expecting 'Left' key presses. The order should be randomized for each subject. Even though the order is random, each time there should be 5 trials of each trial type so the design is balanced. For this exercise, use an array for Trial_Type that contains 1's and 2's (1 for Right, 2 for Left). You will want to create an array that has five 1's and five 2's and randomize the order for each subject. There are multiple ways to do this (Hint: Trial_Type(randperm(numel(Trial_Type))) = Trial_Type; this will work but do you understand why? Read help **numel** and help **randperm**).

RT should contain the reaction time data. Normally, these are generated by the subject. For this homework, you will create an array that has randomly generated numbers between 0 and 1.

MyData should have the following subfields and look something like:

Subject_Number: 202

Test_Time: '14-Oct-2013 14:07:01'

Data: [1x1 struct]

and MyData.Data should have the following subfields and look something like:

Trial_Num: [1 2 3 4 5 6 7 8 9 10]

Trial_Type: [1 1 2 2 2 1 2 2 1 1] ← (this element order of this array is randomized)

RT: [0.2192 0.8721 0.3372 0.1214 0.9214 0.1012 0.4101 0.6561 0.2812 0.7823]

You should save MyData into a file called YourLastName_RTData.mat. (help **save**).

Submit both YourLastName_RTdata.m (commented) and YourLastName_RTData.mat.

Part 2B

Once you have successfully run the above and created the data mat file. Start a new m file called YourLastName_RTAnalyze.m. This file should begin with

clear all;

load 'YourLastName_RTData.mat';

At this point you will have access to MyData, which you created in Part 1. Then your script should calculate and display the following values:

Mean RT for Right hand key presses

Mean RT for Left hand key presses

The Minimum RT and information about it (see box)

The Maximum RT and information about it (see box)

```
> Mean RT for Left Hand was 0.5612

> Mean RT for Right Hand was 0.4911

> Fastest RT was 0.1012, which was on
Trial 6 and it was a Right Hand Trial

> Slowest RT was 0.9214, which was on
Trial 5 and it was a Left Hand Trial
```

For now, just have your script display this information on the matlab window using **disp**. We will soon learn how to write this information in a file. The box above is just an example of the kinds of information you will report so don't worry if your matlab window does not look exactly like this. And of course, you will have different values since your data will be different.

There are several ways of achieving these results. Take a moment to think about how you will do it and plan it on paper if that helps. If you create any temporary variables, arrays, or structures, please explain your reasoning using comments in your m file (using %).

Hint: When you want to operate on a subfield of a structure using a function, you will need to enclose your struct in square brackets. E.g., mean ([mystructure.subfield]), whereas with arrays you can do mean (myarray).

Hint 2: You may need to convert the numbers to strings using **num2str** or similar functions.

Submit YourLastName_RTAnalyze.m