EECS 1570

Winter 2021

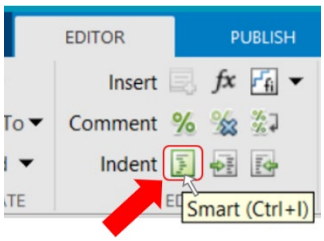
Assignment 4

Due: Monday March 29 by 11:59 p.m.

1. This assignment is worth **6%** of your final grade.
2. Submit your assignment electronically through eClass. Assignments submitted by email will not be accepted.
3. A list of files to submit is found at the end of this assignment. Organize your MATLAB programs as follows with your student information, the question number, and your solution in the indicated locations.

```
% 1570 - Winter 2021 - Assignment 4

% <Last Name>, <First Names>
% <Student Number>
% <York email address>

% Question <x> solution

clear; clc; % begin with a clean start


      <put your solution here>


% *** end ***
```

4. To improve the readability of your MATLAB programs, ensure the statements are properly indented. This is a good habit to develop and is particularly important as your programs get more complex. Here's a quick way to get the proper indentation: (i) Open a script in the MATLAB Editor Window. (ii) Type Ctrl-A to select all the statements in the script. (iii) In the Editor ribbon, click the Smart Indent button.



Questions:

1. (6 points) Write a MATLAB program (**a4q1.m**) to have some fun with random numbers. Begin with a banner message (see below) and then use MATLAB's **randi** function to initialize two integer variables: **lower** in the range 1 to 100, and **upper** in the range 101 to 1000.

   Then, generate 100 random integers in the range **lower** to **upper**. Output the numbers, formatted as shown below.

As well, compute and output the following summary of the numbers:

- The lowest generated number
- The highest generated number
- The mean of the numbers
- The number of odd numbers
- The number of even numbers
- The number of prime numbers
- The prime numbers

To check for prime numbers, use MATLAB's **isprime** function.

End with "Thank you. Bye."

Below is an example run of the program.

```
              FUN WITH RANDOM NUMBERS
================================================
      100 Random Numbers in Range 66 to 467
------------------------------------------------
  395  354  455  279  196  108  311  379  236  102
  173  127  178  242  277  249  417  274  445  322
  450  162  337  182  336  345   93  168  156  334
  405  204  379  337   68  308  221  434   66  251
  236  251  375  195  381  255   80  136  356  256
  127  203  310  143  362  163  434  174  373  141
  181  102  297  340  285  237  325  326  338  321
  445  149  351  160  113  310  246  250  332  375
  206  332  233  404  400  169  312  300  283  415
  172  193  113  443  325  258  323  284  326  284
================================================

Summary
-------
Lowest: 66
Highest: 455
Mean: 269.6
Numbers that are...
   odd: 51
  even: 49
Number of prime numbers: 21
Primes...
     311  379  173  127  277  337  379  337
     251  251  127  163  373  181  149  113
     233  283  193  113  443
Thank you. Bye.
```

Your program should generate output exactly as shown above, adjusted, of course, for the actual numbers generated. Output the prime numbers eight (8) per line, exactly as shown above, adjusted, of course, for the number of prime numbers. Also, output the prime numbers using a single **fprintf** statement and without using a loop statement.

2. (6 points) Do you work on homework with music playing in the background? Let's explore this idea. This question examines whether concentration is affected by the style of music playing in the background. A research project engaged six participants to perform a sorting task with music playing in the background. Each participant was tested on two music conditions: classical music and hip hop music. For each music condition, five sorting trials were performed. The time in seconds was recorded for each trial. Less time means faster sorting, and, arguably, better concentration. The data are shown below.

Participant	Classical Music					Hip Hop Music				
	1	2	3	4	5	1	2	3	4	5
1	54	44	48	42	42	53	51	47	42	40
2	56	48	43	43	30	95	53	55	60	49
3	62	58	54	47	41	84	89	85	64	63
4	46	42	34	29	28	48	41	41	33	33
5	72	44	43	37	43	79	50	40	38	32
6	63	59	42	40	41	78	58	70	55	57

The data are stored in the file **sorting_task.txt** which is available on eClass with this assignment. Write a MATLAB script (**a4q2.m**) that reads the data into a matrix **ST**. Do not output the data. Analyse the data to determine

- the mean sorting time in seconds over all trials in the experiment,
- the mean sorting time for each music condition (classical vs. hip hop),
- which music condition took less time in the sorting task and by what percentage relative to the other condition, and
- whether the difference in sorting time between the two music conditions was statistically significant.

Your MATLAB script should output the results of the above analysis in the command window. The output should look like this:

```
============================================================
Effect of Background Music on Concentration (using a sorting task)
------------------------------------------------------------
Result for sorting time
Grand mean: xxxxx (s)

Results by listening condition
Classical music: xxxxx (s)
Hip hop music: xxxxx (s)

Summary
Sorting time was xxxx% less while listening to xxxx music
Significance test: t(xxx) = xxxxx, p = xxxxxx (xxx)
============================================================
```

Insert values in the output where you see **x**'s above. The values should be computed from the data. Do not type the values into your script. The first three values are times in seconds (with two decimal places). The next computed value is the difference as a percent (with one decimal place) for the music condition that took less time in the sorting task relative to the condition that took more time. On the same line, the next computed value is either "classical" or "hip hop"; i.e., the music condition that took less time overall in the sorting task.

The final line of output text gives the results of a *t*-test to determine if the difference in sorting time between the two music conditions was statistically significant. The four computed values, in order, are

- the degrees of freedom (an integer)
- the *t*-statistic (with two decimal places)
- the probability (with four decimal places), and
- either "significant" (if $p < .05$) or "not significant" (if $p \geq .05$)

For the significance test, review example 2 in lecture 7 (Statistical Functions). There is one minor difference here. You need to use the function **ttest**, rather than **ttest2**, which was used in example 2 in lecture 7.[1]

3. (4 points) An experiment with 16 participants was conducted to compare four soft keyboards for text entry on smart phones. For each keyboard, nine trials were administered. Each trial involved entering a phrase of text. The text entry speed in words per minutes (wpm) was logged for each trial. The following data were collected. Each value is the mean text entry speed (wpm), averaged over 16 participants for the corresponding trial.

Trial	Qwerty	Octopus	TP_Curve	TP_Plus
T1	53.34	24.65	28.69	26.91
T2	50.89	34.99	31.31	30.24
T3	52.29	47.55	31.27	35.74
T4	52.04	54.73	32.35	36.89
T5	57.71	61.59	36.31	40.92
T6	56.64	66.84	36.47	41.83
T7	54.87	62.52	40.09	44.12
T8	52.57	68.90	41.15	44.98
T9	55.47	70.83	40.30	46.27

The above data are in the file **keyboard_comparison.txt** which is available in eClass with this assignment. Additional details on the data are found in **keyboard_comparison_readme.txt**, also available on eClass.

Write a MATLAB program (**a4q3.m**) that uses the **readtable** function to read the data from **keyboard_comparison.txt** into a table **T**. Process the data to compute the mean and standard deviation of the entry speed over the nine trials for each keyboard. Generate output to the Command Window, formatted as follows:

[1] The data here require a *paired-sample t-test* (**ttest**) because the experiment used a within-subjects design: Each participant was exposed to both music conditions. The data in example 2 in lecture 7 used a *two-sample t-test* (**ttest2**) because the experiment used a between-subjects design: Two groups of participants were tested with each participant exposed to only one condition.

```
==============================
Keyboard Comparison Experiment
==============================
Results for Entry Speed (wpm)
------------------------------
Keyboard     Mean       SD
------------------------------
  Qwerty     xxxxx    xxxxx
 Octopus     xxxxx    xxxxx
TP_Curve     xxxxx    xxxxx
 TP_Plus     xxxxx    xxxxx
==============================
```

Where you see **x**'s above, insert the corresponding mean or standard deviation (with two decimal places). Where you see the keyboard names above, extract the keyboard names from **T**. Do not type the names into your script. Also, generate the four data lines in the table above using a single **fprintf** statement with three formatting elements and without using a loop.

What to hand in

Submit your assignment electronically via eClass. Please submit three (3) script files:

a4q1.m (script for Question 1)
a4q2.m (script for Question 2)
a4q3.m (script for Question 3)

*** end ***