General Instructions

Due Date: Sunday, July 23rd by 11:59pm (submit via Canvas)

Assignment Summary Instructions:
This assignment has one problem, summarized below. You will use MATLAB as a tool to solve the problem for the given test cases, ensuring that your code is flexible for any additional test cases that might be used to evaluate it.

- Seismic Data (Application of Civil, Geotechnical, Electrical, and Petroleum Engineering)

Canvas Submission Instructions:
After completing this assignment in MATLAB, to receive credit, you must submit your code in Canvas MATLAB Grader. The following components must be submitted as part of the MA5 Module:

- MA5: SortColumns Function
- MA5: Main Script

To submit your script, copy and paste your code into the submission window, making sure to remove any housekeeping commands. You may submit to Canvas MATLAB Grader as many times as you want before the deadline, without any penalty. The highest score attained before the deadline will be graded. All components are due before the due date. Credit for each component will be awarded based upon the percentage of successfully completed assessments.

Explanation of P-Code:
Under the Additional Resources folder accompanying this prompt, you will find a file named **SortColumns_Solution.p**. This is a working solution to the Projection function required to complete this problem. This file is unopenable and the contents can't be read in a text editor, but it can be called like a typical .m function file in MATLAB. If you get stuck and are unable to successfully develop your own Projection function, you are encouraged to instead copy FindZero_Solution.p to your working folder and rename it **SortColumns.p**. Now, when your main script calls the Projection function, it will automatically call the .p file and you will now have an opportunity to successfully finish developing your main script.

Proficiency Time: Times are included with the Background and Task sections. These times are the estimated amount of time it should take you to **redo** an assignment once you are fully proficient in material that it covers. To practice, reread the background in the given Comprehension Time and attempt to complete the problem in the given Proficiency Time.

Academic Honesty Reminder

The work you submit for this assignment should be your work alone. You are encouraged to support one another through collaboration in brainstorming approaches to the problem and troubleshooting. In this capacity, you are permitted to view other students' solutions, however, copying of another student's work is strongly discouraged.

This assignment will be checked for similarity using a MATLAB code. The similarity code will check each submission for likeness between other student submissions, past student submissions, the solution manual, and online resources and postings. If your submission is flagged for an unreasonably high level of similarity, it will be reviewed by the ENGI 1331 faculty, and action will be taken by faculty if deemed appropriate.

NOTE: Since this is an automated system for all sections, if any of your work is not your own, you will be caught. Changing variable names, adding comments, or spacing will not trick the similarity algorithm.

Background: Comprehension Time: 3 – 5 min

Though often overlooked in states like Texas, Seismic Engineering is an interdisciplinary branch of civil engineering that analyzes how seismic wave propagations (aka, earthquakes) affect man-made structures. Seismic design is a factor in almost every building design in the world and understanding of how seismic waves travel is critical in ensuring they are safe.

You are a civil engineer working with seismic data which represents the velocity of seismic waves in a layer of earth. To better analyze this data, you are required to organize and perform some data validations on it. You are provided the .csv file **SeismicData.csv** which contains the data you must analyze. Each cell value in this data represents a velocity in the Earth's crust, measured in m/s. Each column of data represents a set of measurements taken at a certain location in the crust.

Tasks: Proficiency Time: 55 – 80 min

TASK 1: (7 – 12 min)
Load in the data file **SeismicData.csv.** There should only be positive values in this data. Check each value in the data, and if a value of zero or less is found prompt the user for a value to replace that measurement. Continue prompting the user for an input until a positive number is entered. In the statement prompting the user, include the index (row and column values) of the value that must be replaced. Once all the values in the data are positive, output the number of locations that the user changed to positive to the command window.

TASK 2: (7 – 12 min)
Depending on the region of the crust, the maximum seismic wave velocity possible is between 13,500 and 15,000 [m/s]. Prompt the user to define the maximum limit. Check that the user has entered a value in the acceptable range and continue prompting the user until an appropriate value is entered.

Check the dataset for any values greater than the velocity limit. If a value over the velocity limit is found, replace that measurement with the limit. Store the indices of the replaced values in a matrix, with the row values stored in column 1 and the column values stored in column 2. Export this matrix as **SeismicData_changes.csv**. Output the number of measurements that were replaced with the velocity limit to the command window.

TASK 3: (20 – 26 min)
Typically, seismic velocity data is presented such that the velocities for each location are in ascending order. Create a user-defined function named **SortColumns.m** that can sort all the columns of a given matrix in ascending order. This function will have one input (the matrix with columns to be sorted) and one output (the matrix with its columns sorted in ascending order). Use this function to sort each column of the seismic velocity dataset in ascending order. Export the sorted velocity matrix as **SeismicData_sorted.csv.**
For Task 3, you cannot use the following functions: sort(), sortrows(), max(), min(), find(), or equivalent variations.

TASK 4: (12 – 17 min)
Due to variations in the Earth's crust, normalizing the data can be useful or necessary to make accurate comparisons between measurements taken at different locations. Prompt the user to choose a location (column) to normalize. The normalization equation is given as

$$v_n = \frac{v - v_{min}}{v_{max} - v_{min}} \quad \text{Eq. 1}$$

where a normalized velocity measurement v_n is found with the original velocity measurement v and the minimum and maximum velocity measurements for that location, v_{min} and v_{max} respectively. Output a formatted table of the original values and the normalized values (see sample output).
For Task 4, you cannot use the following functions: functions listed for Task 3, cell2table(), or equivalent variations.

<u>TASK 5: (3 – 5 min)</u>
Ask the user if they would like to normalize a different column of the sorted velocity data. If yes, repeat Task 4, starting with the user input to choose a location to normalize. If no, the program should continue to Task 6.

<u>TASK 6: (6 – 8 min)</u>
Plot the set of normalized velocity data for the final location in two ways: with the measurement number on the x-axis and the normalized velocities on the y-axis, and with the normalized velocities on the x-axis and the original velocities on the y-axis. Your plots should contain the following, and be presented as two figures on the same window (subplotted):

- X-axis label
- Y-axis label
- Title
- Gridlines

Sample Output

<u>Sample output is one continuous run of the code, broken up to make it easier to view</u>

Command Window

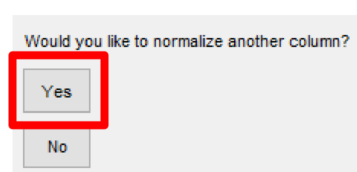
```
Please enter a positive value at (6, 6): -5
Please enter a positive value at (6, 6): 0
Please enter a positive value at (6, 6): 1000
Please enter a positive value at (7, 1): 2000
Please enter a positive value at (10, 1): 3000
Please enter a positive value at (10, 9): 4000
Please enter a positive value at (12, 3): 100000

There were 5 values changed from negative/zero to positive.

Enter a velocity limit between 13,500 and 15,000 [m/s]: 13000
Enter a velocity limit between 13,500 and 15,000 [m/s]: 14500
There were 11 values over the velocity limit.

Which location's data would you like to normalize? 2
Original    Normalized
   87        0.0000
  795        0.0548
 2824        0.2117
 2912        0.2186
 4518        0.3428
 5083        0.3865
 5608        0.4271
 6872        0.5249
 7339        0.5610
 7782        0.5953
 8327        0.6375
10106        0.7751
11585        0.8895
11978        0.9199
13013        1.0000

Which location's data would you like to normalize? 5
```



Command Window

```
Which location's data would you like to normalize? 5
Original     Normalized
 1494        0.0000
 2448        0.0734
 2588        0.0841
 2859        0.1050
 2969        0.1134
 3716        0.1708
 3901        0.1851
 3918        0.1864
 4470        0.2288
 6029        0.3487
 6146        0.3577
 7944        0.4959
10305        0.6775
11728        0.7869
14500        1.0000
```

Would you like to normalize another column?

Yes

No

```
Which location's data would you like to normalize? 10
Original     Normalized
  534        0.0000
 1511        0.0700
 1958        0.1020
 5414        0.3494
 6123        0.4002
 6507        0.4277
 7378        0.4900
 7678        0.5115
 7753        0.5169
 8984        0.6050
 9437        0.6375
13209        0.9076
13253        0.9107
14500        1.0000
14500        1.0000
```

Would you like to normalize another column?

Yes

No

SeismicData_changes.csv and part of SeismicData_sorted.csv:

	1	2
1	1	7
2	1	10
3	5	6
4	6	8
5	7	8
6	7	10
7	9	3
8	10	5
9	10	8
10	12	1
11	12	3
12		

	1	2	3
1	604	87	297
2	671	795	523
3	2000	2824	1786
4	2513	2912	2343
5	3000	4518	5151
6	3533	5083	6190
7	4002	5608	6837
8	4031	6872	6908
9	6984	7339	7330
10	8667	7782	9631
11	8974	8327	10393
12	10881	10106	11210

Required plots: