Assignment 7: Matlab Input and Output

Solution Instructor: Trani

Problem 1

The tallest structures in the World are shown in Table 1 as part of an Excel file to be used in this example.

Table 1. Tallest Structures in the World. Source: Wikipedia (2015).

Type of Structure	Name	Country	City	Height (m)	Height (ft)	Year
Skyscraper	Burj Khalifa	United Arab Emirates	Dubai	829.8	2722.0	2010
Self-supporting tower	Tokyo Skytree	Japan	Tokyo	634.0	2080.0	2011
Guyed mast	KVLY-TV mast	United States	Blanchard, North Dakota	628.8	2063.0	1963
Clock building	Abraj Al Bait Towers	Saudi Arabia	Mecca	601.0	1972.0	2011
Mast radiator	Lualualei VLF transmitter	United States	Lualualei, Hawaii	458.0	1503.0	1972
Twin towers	Petronas Twin Towers	Malaysia	Kuala Lumpur	452.0	1482.0	1998
Chimney	Ekibastuz GRES-2 Power Station	Kazakhstan	Ekibastuz	419.7	1377.0	1987
Radar	Dimona Radar Facility	Israel	Dimona	400.0	1312.0	2008

Task 1

Create a Matlab script to read the data. Use the Matlab "xlsread" command in Matlab to accomplish this. The script should also rename the variables according to the names shown in the header of the Excel file. For example, the first column should be converted into a variable called "TypeOfStructure" and so on.

```
% Sample file to import Excel data into Matlab using the
% xlsread command
% A. Trani, March 30, 2015

clear
clc

% File to read looks like this:

% Type of Structure Name    Country City    Height (m)  Height (ft) Year
% Skyscraper    Burj Khalifa    United Arab Emirates    Dubai   829.8   2722.0  2010
% Self-supporting tower Tokyo Skytree   Japan   Tokyo   634.0   2080.0  2011
% Guyed mast    KVLY-TV mast    United States   Blanchard, North Dakota 628.8   2063.0  1963

% Task 1 - Read the complete Excel file

[num,txt,raw] = xlsread('tallest_buildings.xlsx');
```

Task 2

Modify the Matlab script created in Task 2 to find the structures taller than 300 meters. For this solution employ a pointer or index variable as explained in class. In your script create a new variable that contains the names of found structures. Find the average height (in both feet and meters) for this group of tall structures. Write to the command window the answer using the "disp" command in Matlab. In your solution display (i.e., include a screen capture) the names of the structures in the command window so that we know which structures were produced by your code.

```
% Task 2 - Rename variables and  find structures taller than 300 m

noRows = length(num);                          % Detect the number of records

typeOfStructure    = raw(2:noRows,1);     % extracts the type of structure
name               = raw(2:noRows,2);     %  saves the structure name
country            = raw(2:noRows,3);     % saves the country name
height             = num(:,1);                 % saves the height of the structure in meters
yearBuilt          = num(:,3);                 % saves the year built

% Finds structures taller than 300 m

matchHeightGT300       = find(height>300);              % finds indices of structures taller than 300 m
heightGT300            = height(matchHeightGT300);      % height of structures > 300 m
averageHeightGT300     = mean(heightGT300);             % mean of structures found
namesHeightGT300       = name(matchHeightGT300 );       % names of structures > 300 m

clc
disp(['Average height of structures taller than 300 m is  ',num2str(averageHeightGT300), ' meters' ])
disp(' ')
```

Average height of structures taller than 300 m is 448.4437 meters. The following structures were found to be taller than 300 m.

Task 3

Modify the Matlab script created in Task 2 to find the structures build after the year 1990. Create a variable that contains the names of these structures and count the number of aircraft in this group. Find the average height of the structures found.

```
% Task 3 - Find the structures build after the year 1990

matchBuiltAfter1990           = find(yearBuilt>1990);           % finds indices of structures taller than 300 m
heightBuiltAfter1990          = height(matchBuiltAfter1990);    % height of structures > 300 m
averageHeightBuiltAfter1990   = mean(heightBuiltAfter1990);     % mean of structures found
namesBuiltAfter1990           = name(matchBuiltAfter1990);      % names of structures > 300 m

clc
disp(['Average height of structures built after 1990 is  ',num2str(averageHeightBuiltAfter1990), ' meters' ])
disp(' ')
```

```
'Burj Khalifa'
'Tokyo Skytree'
'KVLY-TV mast'
'Abraj Al Bait Towers'
'Lualualei VLF transmitter'
'Petronas Twin Towers'
'Ekibastuz GRES-2 Power Station'
'Dimona Radar Facility'
'Kiev TV Tower'
'Zhoushan Island Overhead Powerline Tie'
'Gerbrandy Tower'
'TV Tower Vinnytsia'
'Millau Viaduct'
'Obninsk Meteorological tower'
'Lakihegy Tower'
'Jinping-I Dam'
```

Average height of structures built after 1990 is 296.1516 meters.

Task 4

Modify the Matlab script created in Task 3 to identify the tall structures in the United States. Create a new variable with the structures built in the US. In your solution display (i.e., include screen capture) the names of the structures in the command window so that we know which structures were produced by your code.

```
% Task 4 - Find the structures in the United States

matchUSStructures = strcmp(country,'United States');     % match structures in the US
indicesUSMatch = find(matchUSStructures);                % get indices of structures built in the US
structuresInUS = name(indicesUSMatch);
```

```
structuresInUS =

   'KVLY-TV mast'
   'Lualualei VLF transmitter'
   'Gateway Arch'
   'ATLAS-I at Kirtland Air Force Base'
   'Anaconda Smelter Stack'
   'San Jacinto Monument'
   'Chicago Temple Building'
   'High Roller'
   'Philadelphia City Hall'
   'Vehicle Assembly Building'
   'Arecibo Telescope'
   'Kingda Ka'
```

Problem 2

This problem deals with a beam supported at both ends and with a single load *W at a known location*. The beam is shown in Figure 1.

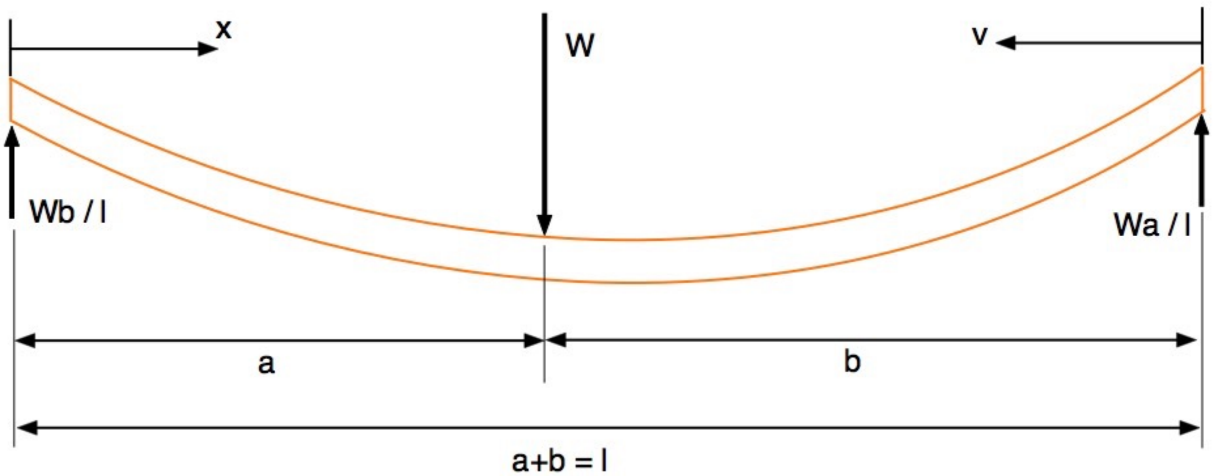


Figure 1. A simple beam supported at both ends with a load W at some known location. Adapted from: http://www.engineersedge.com/beam_bending/beam_bending3.htm.

Nomenclature for beam deflection and stress calculation equations.

W = load (N)
E = Modulus of elasticity (N/m^2)
I = Moment of inertia (m^4)
x = Distance from left side of the beam to the loading point (m)
v = Distance from right hand side beam end point to the loading point (m)
l = beam length (m)
a, b = distances from each beam end point towards the loading point (m)

Let:

$$y_a = \frac{-Wbx}{6EIl}(l^2 - x^2 - b^2)$$

$$y_b = \frac{-Wav}{6EIl}(l^2 - v^2 - a^2)$$

where:

y_a = deflection of the beam from left datum point (section a) in meters
y_b = deflection of the beam from right hand side datum point (section b) in meters

Task 1

Create a Matlab script to estimate the deflections of the beam (y_a and y_b) as a function of known quantities W, E, I, l ,a, and b. Your script should define parameters: W, E, I, l, a and b. The output of the Matlab script consists of the values of beam deflections (y_a and y_b) and their stations (values of x and v).

```
% A. Trani (March 30, 2015)

% W = load at station (N)
% E = Modulus of elasticity (N/m-m)
% I = Moment of inertia (m-m-m-m)
% x = beam station = distance from datum point (wall) to any point on the
%      beam (m)
% l = beam length (m)
% ya, yb = beam deflections at various stations (m)

% Beam properties

W = 6000;                    % Newtons/m
E = 200e9;                    % N/m-m - value for Steel = 200e9
I = 0.001;                    % meters to the fourth power
length_of_beam = 8;          % meters

a = 3;                        % point of application of load W (m)
b = length_of_beam - a;     % complementary distance (m)

% Calculate deflection to the beam at any point in the beam length for part
% a
x = linspace(0,a,100);        % 100 points from left hand side

ya = -W * b * x/ (6*E * I * length_of_beam) .* ( length_of_beam^2 ...
       - x .^2 -b^2);

v = linspace(0,b,100);        % 100 points to the end of the beam

yb = -W * a * v/ (6*E * I * length_of_beam) .* ( length_of_beam^2 ...
       - v .^2 -a^2);
```

Task 2

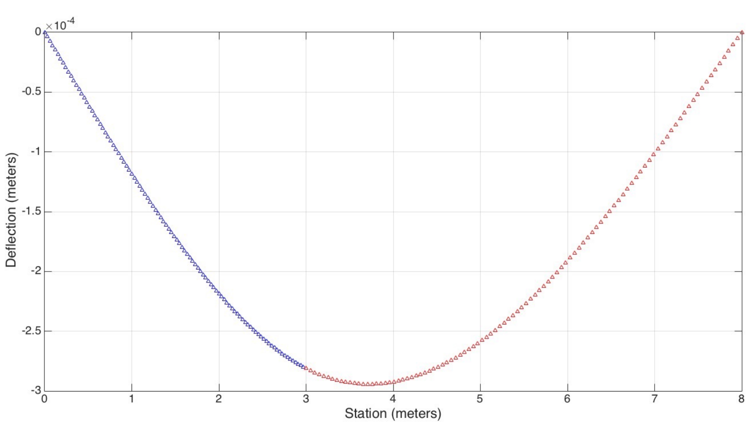
Test your Matlab script using the following values for the beam model parameters. Assume a steel beam is 8 meters long and has the following parameters.

W = 6000 N
E = 200e9 N/m-m
I = 0.001 m^4
l = 8 meters
a = 3 meters
b = 5 meters

Task 3

Plot the results of the beam deflections as a continuous distance (from the left datum of the beam) and verify that the deflections make sense. Note that I added a negative sign to the deflection equations to show the deflections downwards.

```
figure
plot(x,ya,'^b',length_of_beam - v,yb,'^r')
xlabel('Station (meters)','fontsize',24)
ylabel('Deflection (meters)','fontsize',24)
grid
```



Problem 3

The population data file included in this homework contains population data for more than 3000 counties in the US. The data file is found in the Syllabus page (week 2) and names “US Population”. The file data is organized in column format as shown in Table 2. Note that each column with data represents a year.

Table 2. United States County Population Data.

TOTAL POPULATION (IN THOUSANDS)							
NAME	1970	1975	1980	1985	1990	1995	1996
AUTAUGA, AL	24.659	29.619	32.216	32.248	34.353	39.112	40.207
BALDWIN, AL	59.526	67.812	78.949	89.402	98.955	120.896	125.412
BARBOUR, AL	22.642	24.864	24.739	25.001	25.505	27.854	28.298
BIBB, AL	13.823	14.751	15.745	16.157	16.697	18.507	18.885
BLOUNT, AL	27.04	32.138	36.536	37.416	39.473	44.06	45.344
BULLOCK, AL	11.734	11.112	10.599	10.778	11.032	11.431	11.574
BUTLER, AL	21.964	21.195	21.772	22.425	21.886	21.824	21.863
CALHOUN, AL	103.185	108.245	120.025	118.648	116.118	116.79	116.684
CHAMBERS, AL	36.333	37.506	39.228	38.613	36.884	37.179	36.907
CHEROKEE, AL	15.781	18.11	18.888	18.891	19.643	21.871	22.387

Task 1

Use the Matlab import wizard to import the data contained in the file. Create a variable for each column during the import procedure. Save all the variables created and contained in workspace using the “save” command in Matlab (i.e., to a Matlab binary file).

Workspace

Name ▲	Value
NAME	3092x1 cell
Pop1970	3092x1 double
Pop1990	3092x1 double
Pop2010	3092x1 double
VarName10	3092x1 double

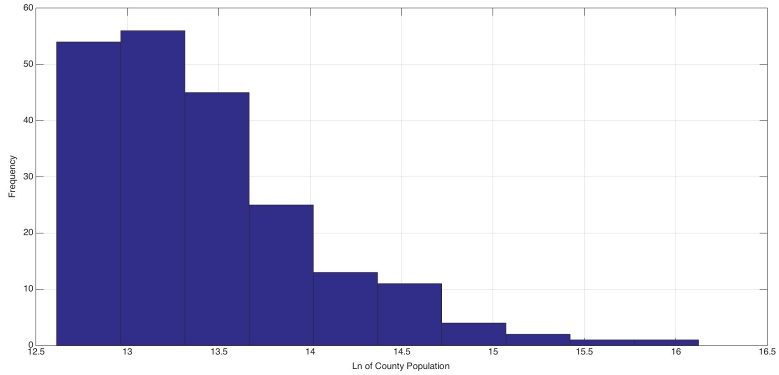
Task 2

Create a script in Matlab that read the data saved in Task 1. Identify counties in the US that had more than 300,000 people in population in the year 2010. Use and index or pointer variable to do this. Create new variable to store the names of the counties that meet the criteria.

```
% Sample file to handle a Population file provided in class
% A. Trani, March 30, 2015

clear
clc

% File contains three variables:
% NAME = county name
% Pop1970 = county population year 1970 (needs to multiply by 1000)
% Pop1990 = county population year 1990 (needs to multiply by 1000)
% Pop2010 = county population year 2010 (needs to multiply by 1000)

load popData

Pop1990 = Pop1990 * 1000;     % data was in thousands
Pop2010 = Pop2010 * 1000;

% Task 2 – Identify counties in the US that had more than 300,000 people in population in the year 2010
% Use and index or pointer variable to do this.
% Create new variable to store the names of the counties that meet the criteria.

matchCountiesGT300K        = find(Pop2010>300000);                   % finds indices of counties with more than 300K people
populationCountiesGT300K   = Pop2010(matchCountiesGT300K);    % saves population of counties with > 300K population
nameCountiesGT300K         = NAME(matchCountiesGT300K);     % saves population of counties with > 300K population
```

Task 3

Add code to the script created in Task 2 to plot a histogram of the distribution of the logarithm (ln) of the county population in the US for counties that had more than 300,000 people in the year 2010. Define a new variable in your script to estimate the natural logarithm of population. A histogram has the frequency of counties found for each bin in the y-axis and the logarithm of the county population in the x-axis.



Task 4

Enhance the script of Task 3. Calculate the rate of growth (in percent) for all US counties between years 1990 and 2010. Create a new variable to identify the counties that experienced a negative growth in the country between 1990 and 2010. Output the names of the counties found using the "xlswrite" command in Matlab. In your answer show a portion (i.e., screen capture) of the Excel file produced.

```
% Task 4 - Find counties with negative growth between 1990 and 2010 and
% output to a file

growthRate = (Pop2010-Pop1990)./Pop1990 * 100;          % growth rate between 2010 and 1990

matchCountiesNegGrowth        = find(growthRate<0);                      % finds indices of counties with more than 300K people
countyGrowthRate              = growthRate(matchCountiesNegGrowth);      % saves population of counties with > 300K population
nameCountiesNegGrowthRate     = NAME(matchCountiesNegGrowth);            % saves population of counties with > 300K population

% Before the cell variable "nameCountiesNegGrowthRate" can be exported it needs to converted into a string variable
% from a cell array. Do this by using the "char" command.

nameCounty = char(nameCountiesNegGrowthRate);

xlswrite('exportCounties.xlsx',nameCounty);    % writes to Excel
```

Problem 4

The airline data file included in this homework contains data for two airlines that are in the process of merging. The comma-separated file is reproduced below.

Table 1. Comma-separated data file with airline data.

```
Airline Name,Manufacturer,Aircraft Type,Engine Type,Age (years),Flight Hours (hrs)
Star Airlines,Boeing,737-800,CFM56-7B26,15,52610
Star Airlines,Boeing,737-800,CFM56-7B26,15,50926
Star Airlines,Boeing,737-800,CFM56-7B26,15,50436
Star Airlines,Boeing,737-800,CFM56-7B26,15,52076
Star Airlines,Boeing,737-800,CFM56-7B26,15,51584
Star Airlines,Boeing,737-800,CFM56-7B26,15,48869
Star Airlines,Boeing,737-800,CFM56-7B26,15,48987
```

Task 1

Create a Matlab script to read the complete data in this file in 6 column format. Use the Matlab "textscan" command with a comma delimiter. An example of the command is shown below:

readData = textscan(fid, '%s %s %s %s %f %f', 'delimiter', ',');

Do not forget to add the lines to open and close the file.

```
% Sample file to import data in Matlab using the
% Textscan command
% A. Trani (March 20, 2015)

clear
clc

% Sample data file

% Airline Name,Manufacturer,Aircraft Type,Engine Type,Age (years),Flight Hours (hrs)
% Star Airlines,Boeing,737-800,CFM56-7B26,15,52610
% Star Airlines,Boeing,737-800,CFM56-7B26,15,50926
% Star Airlines,Boeing,737-800,CFM56-7B26,15,50436
% Star Airlines,Boeing,737-800,CFM56-7B26,15,52076

% Task 1 read the file

fid = fopen('airline_data_csv.csv') ;                          % open the file
readHeader = textscan(fid, '%s', 6, 'delimiter', ',');          % read four columns in first row
readData = textscan(fid, '%s %s %s %s %f %f', 'delimiter', ',');     % read four columns with two strings and two floating point
fclose(fid);
```

Task 2

Enhance the Matlab script created in Task 1 and create 6 variables with the information contained in each column of the file read in Task 1. Label these variables with names that can be used later in the analysis. (see figure below for Task 3).

Task 3

Modify the Matlab script created in Task 3 to find the age of each aircraft in the data of the type "DC-9-83 (MD-83)". Create a variable in your script to also calculate the average age of of this type of aircraft.

```
% Task 2 rename variables

airlineName       = readData{1};     % assign all elements of cell array (column 1) to airlineName
manufacturer      = readData{2};     % assign all elements of column 2 to aircraft manufacturer
aircraftType      = readData{3};     % saves the aircraft type name to a new variable
engineType        = readData{4};
age               = readData{5};
hours             = readData{6};

% Task 3 find the age of  aircraft of the type DC-9-83 (MD-83)

matchNameMD83 = strcmp(aircraftType,'DC-9-83 (MD-83)');      % finds matches to name DC-9-83
indicesOfMD83     = find(matchNameMD83);                      % finds only indices that match

ageOfMD83         = age(indicesOfMD83);                       % finds age of each aircraft that match
averageAgeMD83 = mean (ageOfMD83);                            % find mean of age group
```

Average age of MD83 aircraft is 19.82 years.

Task 4

Modify the Matlab script created in Task 3 to find the aircraft with more than 50,000 flight hours. Find the average age for the aircraft found.

```
% Task 4 Find aircraft with more than 50,000 hrs

matchHoursGT50K          = find(hours>50000);                  % finds matches to aircraft with hours > 50000
aircraftWithHoursGT50K   = aircraftType(matchHoursGT50K);      % finds names of aircraft whose indices match
```