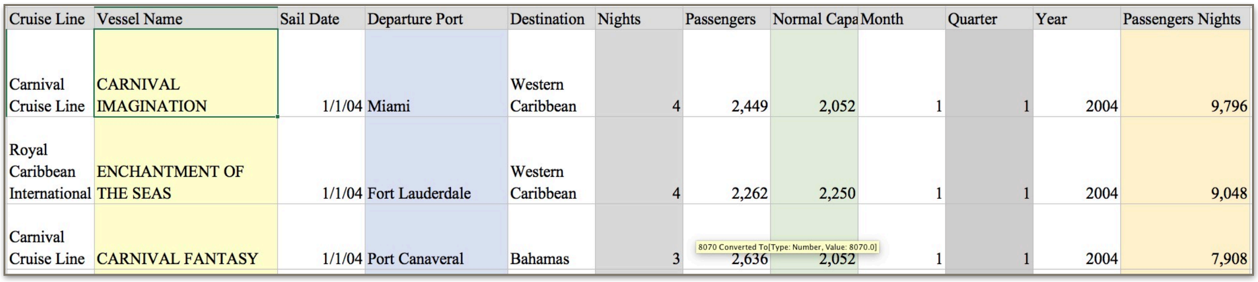
Assignment 6: Matlab Input/Output (Solution)

Date Due: March 24, 2017 Instructor: Trani

Problem 1

Data containing cruise ship information has been collected from the Bureau of Transportation Statistics (BTS). The data is contained in an Excel spreadsheet shown below. Data includes cruise line names, vessel names, days of sailing, etc.



Cruise Line	Vessel Name	Sail Date	Departure Port	Destination	Nights	Passengers	Normal Capa	Month	Quarter	Year	Passengers Nights
Carnival Cruise Line	CARNIVAL IMAGINATION	1/1/04	Miami	Western Caribbean	4	2,449	2,052	1	1	2004	9,796
Royal Caribbean International	ENCHANTMENT OF THE SEAS	1/1/04	Fort Lauderdale	Western Caribbean	4	2,262	2,250	1	1	2004	9,048
Carnival Cruise Line	CARNIVAL FANTASY	1/1/04	Port Canaveral	Bahamas	3	2,636	2,052	1	1	2004	7,908

8070 Converted To[Type: Number, Value: 8070.0]

Task 1:

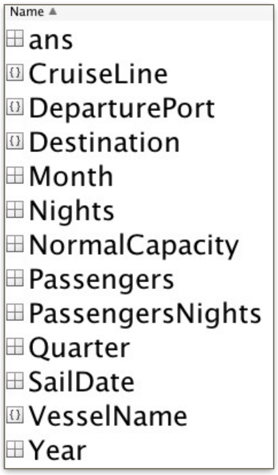
Use the Matlab import Wizard to read the cruise data and save the data into individual cell arrays containing each one of the columns shown in the table above. Generate the code to read the data in Matlab and save it into a Matlab script. Show me the code generated as a screen capture.

Auto generated code in Matlab

```
% Import data from spreadsheet
% Script for importing data from the following spreadsheet:
%
%    Workbook: /Users/atrani/University work/courses/cee3804/Datasets/cruiseShipData.xlsx
%    Worksheet: Sheet1
%
%% Import the data
[~, ~, raw] = xlsread('/Users/atrani/University work/courses/cee3804/Datasets/cruiseShipData.xlsx','Sheet1');
raw = raw(2:end,:);
raw(cellfun(@(x) ~isempty(x) && isnumeric(x) && isnan(x),raw)) = {''};
cellVectors = raw(:,[1,2,4,5]);
raw = raw(:,[3,6,7,8,9,10,11,12]);

%% Replace non-numeric cells with NaN
R = cellfun(@(x) ~isnumeric(x) && ~islogical(x),raw); % Find non-numeric cells
raw(R) = {NaN}; % Replace non-numeric cells

%% Create output variable
```

```
data = reshape([raw{:}],size(raw));

%% Allocate imported array to column variable names
CruiseLine = cellVectors(:,1);
VesselName = cellVectors(:,2);
SailDate = data(:,1);
DeparturePort = cellVectors(:,3);
Destination = cellVectors(:,4);
Nights = data(:,2);
Passengers = data(:,3);
NormalCapacity = data(:,4);
Month = data(:,5);
Quarter = data(:,6);
Year = data(:,7);
PassengersNights = data(:,8);

%% Clear temporary variables
clearvars data raw cellVectors R;
```

Produces:



Name ▲
ans
CruiseLine
DeparturePort
Destination
Month
Nights
NormalCapacity
Passengers
PassengersNights
Quarter
SailDate
VesselName
Year

Task 2:

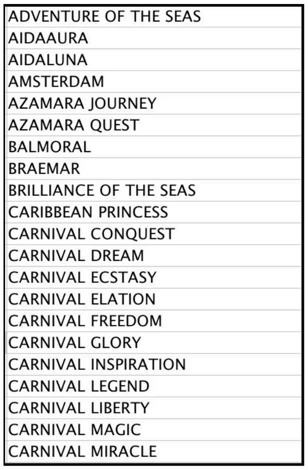
Improve the script developed in task 1 to do the following analyses:

a) Find the indices of the cruises (or instances) with more than 7 nights using the Matlab FIND function.

b) Find the vessel names belonging to the set of cruises that travel more than 7 nights. This means list all the vessel names associated with the indices found in item (a)

```
% Problem # 2
% Finds indices with more than 7 nights of cruise length
indices_moreThan_7nights                =find(Nights>7);
% Finds instances of vessels that cruise for 7 nights or more
vessels7NightsOrMore                     =  VesselName( indices_moreThan_7nights);
% Finds the unique vessels that cruise
uniqueVessels7Nights                     = unique(vessels7NightsOrMore);
% Number of unique vessels cruising for more than 7 days
noUniqueVessels = length(uniqueVessels7Nights );
```

There are **144 distinct ships** that perform cruises of more than 7 days. A sample of the ships found are:



ADVENTURE OF THE SEAS
AIDAAURA
AIDALUNA
AMSTERDAM
AZAMARA JOURNEY
AZAMARA QUEST
BALMORAL
BRAEMAR
BRILLIANCE OF THE SEAS
CARIBBEAN PRINCESS
CARNIVAL CONQUEST
CARNIVAL DREAM
CARNIVAL ECSTASY
CARNIVAL ELATION
CARNIVAL FREEDOM
CARNIVAL GLORY
CARNIVAL INSPIRATION
CARNIVAL LEGEND
CARNIVAL LIBERTY
CARNIVAL MAGIC
CARNIVAL MIRACLE

Task 3:

Improve your script created in Task 2 to find the names of the cruise ships (vessels) whose passenger carrying capacity is less than 1,000 passengers. Again, use the FIND command as needed.

```
indices_Capacity_LT_1000                =find(NormalCapacity<1000);
% Finds instances of vessels that cruise for 7 nights or more
vessels7_Capacity_LT_1000               =  VesselName(indices_Capacity_LT_1000);
% Finds the unique vessels that cruise
uniqueVessels_Cap_LT_1000               = unique(vessels7_Capacity_LT_1000 );
% Number of unique vessels cruiseing for more than 7 days
noUniqueVessels_Cap_LT_1000             = length(uniqueVessels_Cap_LT_1000);
```

There are 28 ships with a capacity below 1,000 . The list is presented in the table below.

Ship Name	Ship Name
AZAMARA JOURNEY'	SEABOURN PRIDE'
AZAMARA QUEST'	SEABOURN SOJOURN'
BRAEMAR'	SEABOURN SPIRIT'

Ship Name	Ship Name
COSTA ALLEGRA'	SEADREAM I'
CRYSTAL HARMONY'	SEADREAM II'
CRYSTAL SYMPHONY'	SEVEN SEAS MARINER'
OCEAN PRINCESS'	SEVEN SEAS NAVIGATOR'
PACIFIC PRINCESS'	SEVEN SEAS VOYAGER'
PRINSENDAM'	SILVER CLOUD'
RADISSON DIAMOND'	SILVER SHADOW'
REGATTA'	SILVER WHISPER'
ROYAL PRINCESS'	SILVER WIND'
SAGA RUBY'	WIND SPIRIT'

Task 4:

Find the average (using the MEAN Matlab command) passenger capacity of the vessels found in Task 3.

```
% Capacity of the cruises found in the analysis
meanCapacityOfCruises_LT_1000 = mean(NormalCapacity(indices_Capacity_LT_1000));
```

```
Average cruise capacity of cruises with capacity < 1000 passenger = 479
```

Task 5:

Find the number of times the ship Celebrity Galaxy sailed in the period of the database.

```
positionOfCelebrityGalaxy        =strcmp('CELEBRITY GALAXY',VesselName);    % generates values of 0 or 1
% Finds the indices that are "1" that match the string "CELEBRITY GALAXY'
indicesOfCelebrityGalaxy         = find(positionOfCelebrityGalaxy);              % Finds indices of 1 = string match
numberOfCruises_CGalaxy          = length(indicesOfCelebrityGalaxy);             % Counts the number of matches found
disp(' ')
disp(['Number of Cruise by the Celebrity Galaxy = ',num2str(numberOfCruises_CGalaxy ) ])
```

Problem 2

A formula to estimate the noise generated by rail vehicles is,

Leq = SELref + 10 log(Ncars) + 20 log(S/52) + 10 log (V) – 36

where:

Leq = equivalent noise level (decibels - dBA)

SEL ref = reference sound exposure level (decibels - dBA)

Ncars = number of cars in the train

S = train speed (mph)

V = hourly average train volume (trains per hour)

Write a Matlab script to calculate the value of Leq given the values of S (speed), SELref (sound exposure level), Ncars (train cars), and hourly train volume (V). Assume the value of SELref to be 76 dBA for this train.

Task 2:

Test the Matlab script created in Task 1 to calculate the value of Leq. In your calculations assume a train has 4 cars, the hourly train volume is 20 trains/hr and the train travels at 55 mph. Display the result in the command window and write the following statement: " The Value of Equivalent Noise Level is: yyyyy (db)". Where yyy is the result of the Matlab script.

Task 3:

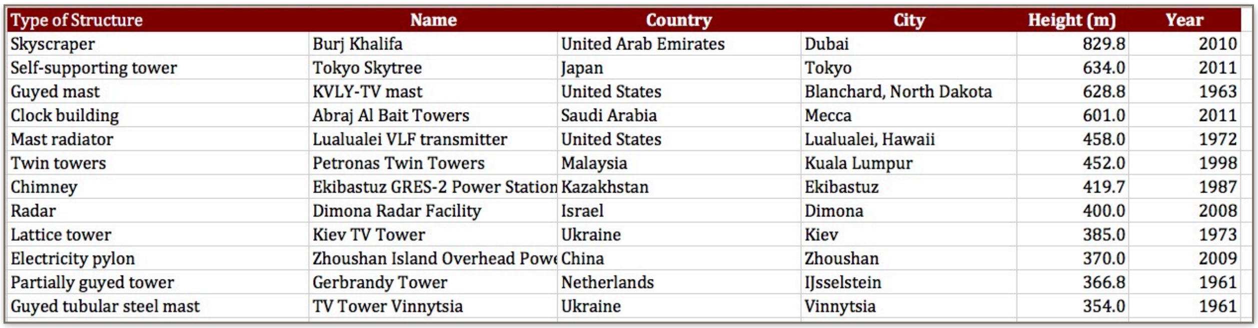
Create a new Matlab script to test the function created in Task 1 but now for an array of speed values ranging from 20-70 mph at intervals of 0.1 mph. In the new Matlab script do a plot of speed vs Leq.

Task 4:

Repeat Task 3 for Ncars = 6 and plot in the same graph as the answers for Task 3.

Problem 3

Data containing 50 famous building structures in the World has been collected from Wikipedia. The data is contained in an Excel spreadsheet shown below. Data includes the type of structure, the name of the building, country, city and height. Show all your work as screen captures.



Type of Structure	Name	Country	City	Height (m)	Year
Skyscraper	Burj Khalifa	United Arab Emirates	Dubai	829.8	2010
Self-supporting tower	Tokyo Skytree	Japan	Tokyo	634.0	2011
Guyed mast	KVLY-TV mast	United States	Blanchard, North Dakota	628.8	1963
Clock building	Abraj Al Bait Towers	Saudi Arabia	Mecca	601.0	2011
Mast radiator	Lualualei VLF transmitter	United States	Lualualei, Hawaii	458.0	1972
Twin towers	Petronas Twin Towers	Malaysia	Kuala Lumpur	452.0	1998
Chimney	Ekibastuz GRES-2 Power Station	Kazakhstan	Ekibastuz	419.7	1987
Radar	Dimona Radar Facility	Israel	Dimona	400.0	2008
Lattice tower	Kiev TV Tower	Ukraine	Kiev	385.0	1973
Electricity pylon	Zhoushan Island Overhead Powe	China	Zhoushan	370.0	2009
Partially guyed tower	Gerbrandy Tower	Netherlands	IJsselstein	366.8	1961
Guyed tubular steel mast	TV Tower Vinnytsia	Ukraine	Vinnytsia	354.0	1961

Task 1:

Use the Matlab XLSREAD function to read the data. Show me your script as a screen capture.

Task 2:

Improve the script developed in Task 1 to do the following analyses:

a) Find number of structures built in the United States.

b) Find the names of the building structures belonging to the set in item (a).

Task 3:

Improve your script created in Task 2 to find the names of the buildings that are less than 20 years old.

Task 4:

Find the average height of Masonry buildings in the database.

Task 5:

Add Matlab code to the script above to export the structure name, country, city and height (i that order) to a comma delimited file.

Problem 4

Retrieve the Autobahn data contained in Week 1 of the syllabus for this class. Save the data into a text file. A sample of the data is shown below.

```
% Traffic Flow Data
%
% Autobahn data
%
% Column 1 = Density (vehicle/km-lane)
% Column 2 = Speed (km/hr)
0.08    160.00
0.08    152.00
0.00    0.00
```

Task 1

Use the Matlab LOAD function to read the data into a script. This script will contain other calculations shown in the next tasks. Parse the data and save two arrays called Density and Speed.

```
% Traffic Flow Data
%
% Autobahn data
load autobahnData.m
%
% Column 1 = Density (veh/km-lane)
% Column 2 = Speed (km/hr)
% 0.08    160.00
% 0.08    152.00
% 0.00    0.00
```

Task 2

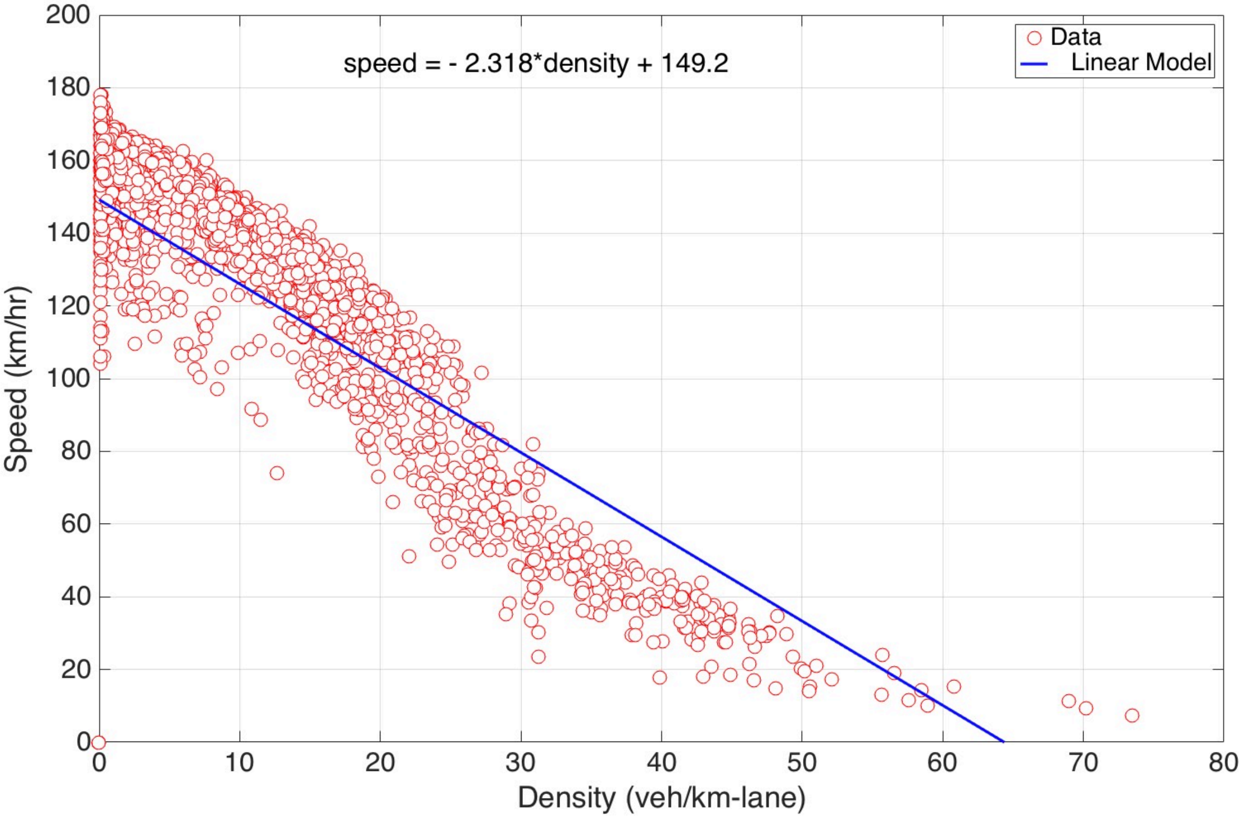
Plot the values of Speed vs Density (in x-axis). Your plot should have large markers (^) and red dots for each point in the data. Also change the font size of the axes labels to 24.

```
% Define a few variables
density        = autobahnData(:,1);
speed          = autobahnData(:,2);
% Make a plot
plot(density,speed,'or')
xlabel('Density (veh/km-lane)','fontsize',24)
ylabel('Speed (km/hr)','fontsize',24)
grid
```

Task 3

Use the Basic Curve Fitting tool in the plot done in Task 2 to estimate the best linear curve fitting possible between speed and density. Comment.

The Speed- Density relationship is shown below. The linear fit is not a good approximation beyond density values of 25 vehicles per lane-km.



Task 4

Estimate the maximum speed recorded in the Autobahn.

Maximum speed is 178 km/hr

max(speed)

Task 5

Select the observations in the data whose speed is above 100 km/kr. select the corresponding values of Density for the same set. Plot two sets of density vs speed data points. Red for speeds values above 100 km/hr. Green for cases where speed is less than 100 km/hr.