

Assignment 6: Matlab Practice and Plots

Solution Set

Instructor: Trani

Problem 1

A civil engineer sends you a “comma separated” data file containing aircraft characteristics used in airport design. The first few lines of the file look as follows:

```
Adam Aircraft Industries,A500,7000 ,98,44.00,9.58
Aero Spacelines,B-377,170000 ,123,156.25,48.50
Aeronca,11AC Chief,1250 ,64,36.00,6.00
Aeronca,15AC Sedan,2050 ,67,37.50,10.33
Aerospatiale,ATR-42-320,36817 ,124,80.58,24.92
```

The six columns of each record are explained below. The file name is “aircraft_characteristics_noheader_csv.csv”.

Column 1 = manufacturer name

Column 2 = aircraft name

Column 3 = aircraft takeoff weight (lb)

Column 4 = aircraft approach speed (nautical miles per hour)

Column 5 = aircraft wingspan (feet)

Column 6 = aircraft tail height (feet)

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.

```
% A script to automate the analysis of aircraft data
% Programmer: T. Trani (March 23, 2012)

clear
clc
% Read the data. The format of the data read is six columns seperated by commas as follows:

% Adam Aircraft Industries,A500,7000 ,98,44.00,9.58
% Aero Spacelines,B-377,170000 ,123,156.25,48.50
% Aeronca,11AC Chief,1250 ,64,36.00,6.00
% Aeronca,15AC Sedan,2050 ,67,37.50,10.33
% Aerospatiale,ATR-42-320,36817 ,124,80.58,24.92

fid = fopen('aircraft_characteristics_noheader_csv.csv'); % open the file (Matlab assigns a file ID)
readData = textscan(fid, '%s %s %f %f %f', 'delimiter', ','); % read six columns with two strings and two floating point (numbers)
fclose(fid) % closes the file that we opened in the previous line
```

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.

```
% Define new variables before doing the analysis  
manufacturer = readData{1};  
name = readData{2};  
takeoffWeight = readData{3};  
approachSpeed = readData{4};  
wingspan = readData{5};  
tailHeight = readData{6};
```

Task 3

Modify the Matlab script created in Task 3 to find the aircraft whose takeoff mass is greater or equal than 255,000 lb. The Federal Aviation Administration calls this aircraft as “heavy”. Create a variable that contains the names of these aircraft and count the number of aircraft in this group. Find the average takeoff weight of this group of aircraft.

```
% Task 3 – find aircraft that are heavy (takeoffweight > 255000 lb)  
  
indicesForHeavyAircraft = find(takeoffWeight > 255000);    % finds indices for heavy aircraft  
heavyAircraft = name(indicesForHeavyAircraft);              % saves the names of heavy aircraft  
noHeavyAircraft = length(heavyAircraft);                   % Number of heavy aircraft  
  
disp(['Number of Heavy Aircraft are: ',num2str(noHeavyAircraft) ])
```

Task 4

Modify the Matlab script created in Task 3 to plot the aircraft takeoff weight vs. the aircraft tail height. Comment on the relationship observed.

```
% Task 4 – Plot takeoff weight vs. tail height  
  
figure  
plot(takeoffWeight,tailHeight,'or')  
xlabel('Takeoff Weight (lb)', 'fontsize',20)  
ylabel('Tail Height (ft)', 'fontsize',20)  
grid
```

The plot produced is shown in Figure 1.

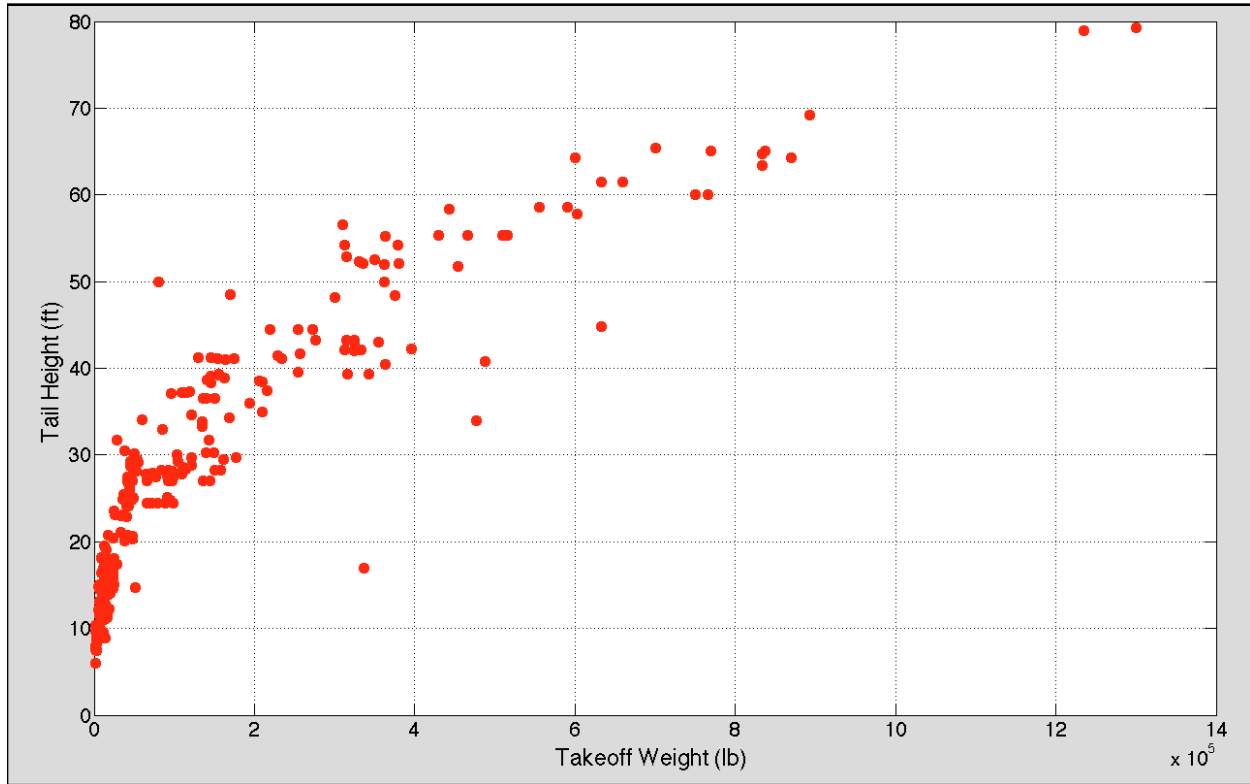


Figure 1. Aircraft Takeoff Weight vs. Aircraft Tail Height.

The relationship is clearly nonlinear and is driven (not a known fact to you) by the designer's desire to provide a large enough tail volume to control the aircraft in flight for a range of flight conditions. At the same time, a desired to reduce the tail size to reduce aerodynamic drag.

Problem 2

A high-speed train (shown in Figure 2) accelerates according to the following formula:

$$a(t) = \frac{dV}{dt} = k_1 - k_2V$$

where $a(t) = \frac{dV}{dt}$ is the rate of change of velocity over time (m/s-s) - acceleration, V is the train speed in m/s and k_1, k_2 are two constants that relate the acceleration of the train as a function of speed. For a High-Speed train the values of k_1 is 2.1 (m/s-s) and $k_2 = 0.023$ (1/second).



Figure 2. High-Speed Train (Amtrak Acela Train).

Task 1

Create a Matlab script to calculate the acceleration of the train as a function of speed. In this script save two vectors containing the values of speed and acceleration.

```
% Constant acceleration model
%
% Equation modeled
% dV/dt = k1 - k2 * V

% Programmer: T. Trani
% March 23, 2012

k1 = 2.1;    % maximum acceleration (m/s-s)
k2 = 0.023; % units are 1/second

% Solve the deceleration equations of motion

trainSpeed = 0:1:90;           % vector with speeds (m/s)
trainAcceleration = k1 - k2* trainSpeed; % acceleration of train (calculated) (m/s-s)

figure
plot(trainSpeed,trainAcceleration,'o-')
xlabel('Train Speed (m/s)','fontsize',20)
ylabel('Train Acceleration (m/s-s)','fontsize',20)
grid
```

Task 2

Plot the speed of the train (x-axis) and the acceleration (y-axis). Label your axes and headings accordingly. Comment on the trends observed.

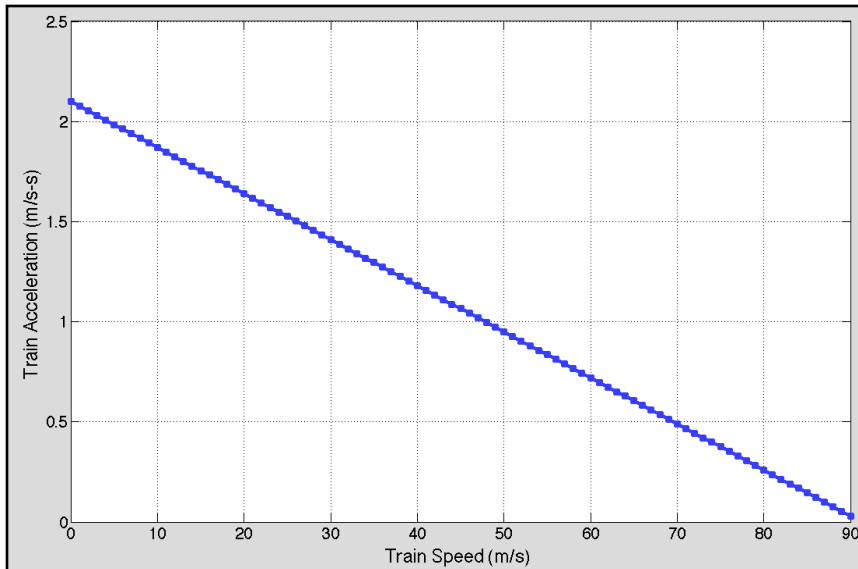


Figure 2. Train Acceleration vs. Speed.

Task 3

Change the attributes of the plot created in Task 2 as follows:

- Line depicting acceleration profile should be blue
- Line width should be 3
- Font size of the X and Y labels should be 24
- Font size of the X and Y numbers should be 20

Problem 3

Task 1

Create a Matlab script to:

a) Load the Autobahn data file in Matlab (see week 1 of the syllabus to retrieve the Autobahn file). The data contains two columns as shown below.

```
% Traffic Flow Data
%
% Autobahn data
%
% Column 1 = Density (veh/km-lane)
% Column 2 = Speed (km/hr)
0.08 160.00
0.08 152.00
0.00 0.00
0.00 0.00
0.07 162.00
```

```
% Script to load Autobahn data to make simple plots
% Programmer: T. Trani
% Date: 03/24/12

% Load the data from a text file

% Data format:

% Column 1 = Density (veh/km-lane)
% Column 2 = Speed (km/hr)

load Autobahn_data.txt
```

Create two distinct variables containing the numerical values of the two columns of the data file loaded in part (a). The first vector should contain all elements of column 1 and should be labeled “density”. The second vector contains the values of speed and should be labeled “speed”.

```
% Setup variables to manipulate later

density = Autobahn_data(:,1);
speed = Autobahn_data(:,2);

% Make a plot of speed vs. density

plot(density,speed,'+')
xlabel('Speed (km/hr)','fontsize',20)
ylabel('Density (veh/km-la)')
grid
```

Task 2

Enhance the script of Task 1. Using the “hist” command in Matlab, plot a histogram of the speeds observed on the Autobahn. Label your axes and headings accordingly. Comment on the shape of the speed distribution of the data. Is there a central tendency?

```
% Task 3 – create a histogram with 20 bins
```

```
hist(speed,20)  
xlabel('Speed (km/hr)', 'fontsize', 20)  
ylabel('Frequency', 'fontsize', 20)  
grid
```

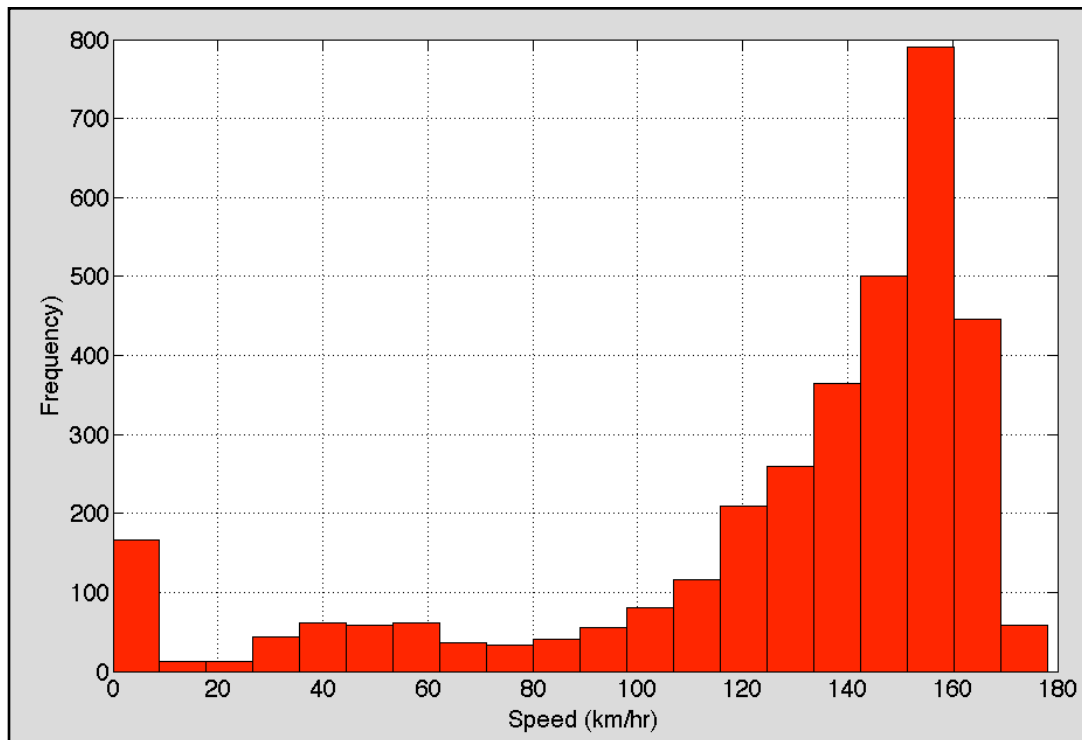


Figure 3. Histogram for Autobahn Speeds.

The tendency of for speeds to be concentrated on the high end of the scale with a mean of

Task 3

Enhance the script of Task 2 to calculate the mean and standard deviation of speeds observed in the Autobahn using the Matlab commands “mean” and “std”, respectively. Output the values of the mean and standard deviation to the command window using the Matlab “disp” command. Comment on the results obtained.

```

meanSpeed = mean(speed);
stdSpeed = std(speed);

clc
disp(['The mean of the speed is ', num2str(meanSpeed) , ' km/hr'])
disp(['The standard deviation of the speed is ', num2str(stdSpeed) , ' km/hr' ])

```

The mean of the speed is 127.6 km/hr
The standard deviation of the speed is 44.0126 km/hr

Task 4

Enhance the script of Task 3. Plot the speed (y-axis) values vs. the density of the traffic flow (x-axis). After the plot is created and with the plot window open, use the “Basic Fitting” feature of Matlab found under the “Tools” pull-down menu to find a regression equation (try first and second order polynomials) that relates density and speed. The “Basic Fitting” menu is shown in the figure below.

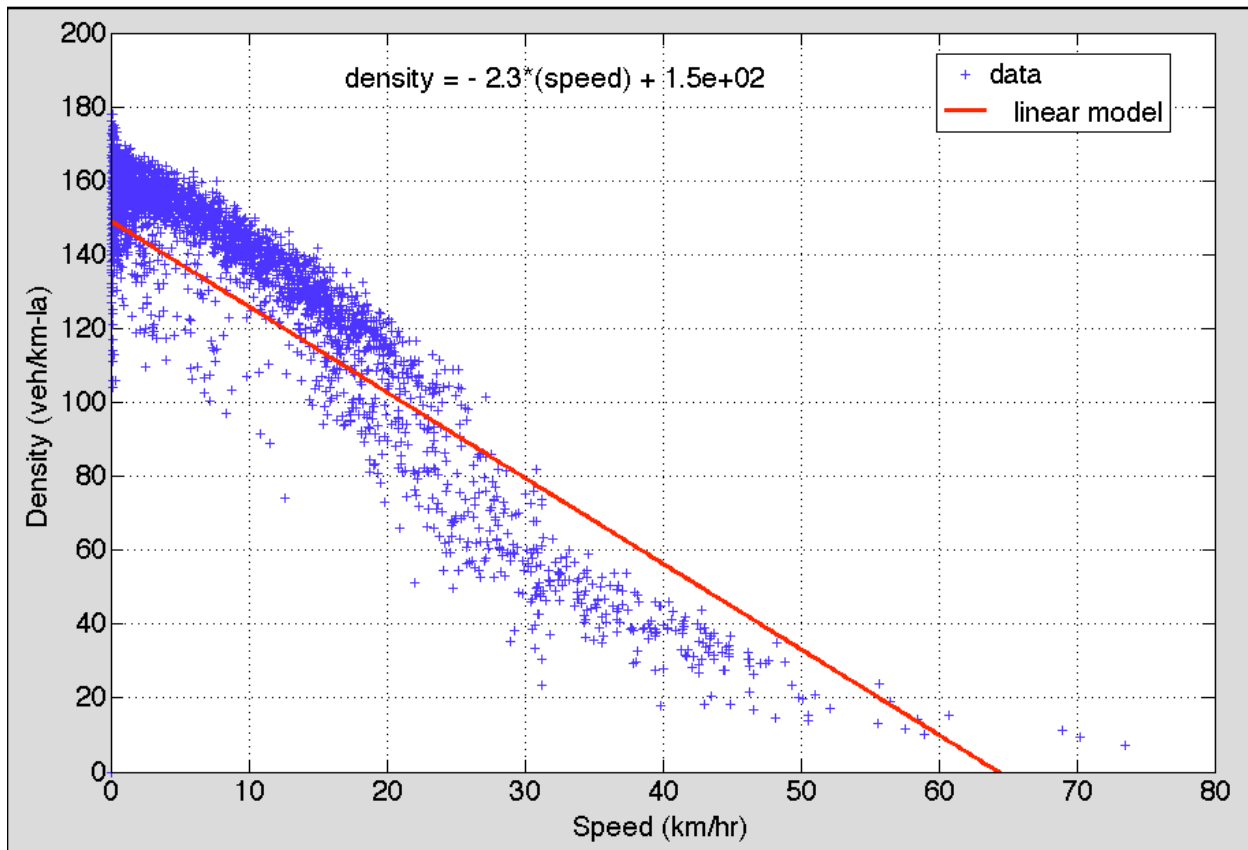


Figure 4. Density vs. Speed Relationship with Linear Model (First-order Polynomial).

Problem 4

The ACME company makes \$1600 for every metric ton of concrete mix of type A delivered. The company makes \$1735 for every metric ton of concrete mix of type B delivered. After formulating this optimization problem to maximize the revenue for the ACME company we found the following first Table (see Table 1) in the Simplex Method (see Homework 3 solution for more information if needed).

Let $x_1 =$ be the amount produced of type A
 $x_2 =$ be the amount produced of type B

$$Z - 1600x_1 - 1735x_2 = 0$$

$$-3/4x_1 + x_2 + x_3 = 500$$

$$2/3x_1 + x_2 + x_4 = 1,067$$

$$3x_1 + x_2 + x_5 = 2700$$

Table 1. Initial Table of the Problem. Current Solution is: $x_1, x_2, x_3, x_4, x_5 = [0 \ 0 \ 500 \ 1067 \ 2700]$. Basic variables are x_3, x_4, x_5 . Non-basic variables (i.e., those that are zero in the solution) are x_1, x_2 .

Basic Variable	Z	x_1	x_2	x_3	x_4	x_5	RHS
	1	-1600	-1735	0	0	0	0
x_3	0	-3/4	1	1	0	0	500
x_4	0	2/3	1	0	1	0	1067
x_5	0	3	1	0	0	1	2700

Task 1

Write down a matrix A in Matlab containing the numerical values found in the first Table to solve the optimization problem. Include the values of the RHS in the matrix A.

Task 2

Use Matlab and perform row operations to solve this optimization problem. In other words, using the matrix created in Task 1 and the rules of the Simplex Method, perform row iterations on the matrix A to reach an optimal solution. For every new table found, write down the numerical values of the modified matrix A. You can verify the solution posted for Assignment 3.

Step 1: select column with largest negative value in row 1 (i.e., Z row)

`[n,rowPivot] = min(A(1,:))`

produces

n = -1735 and rowPivot = 3

rowPivot is the position of the pivot column in matrix A

Step 2: calculate the ratios of the RHS values and the elements of the pivot column

`ratioTest=A(2:n,m)./A(2:n,rowPivot)`

produces

ratioTest =

500
1067
2700

Step 3: select the minimum of the ratios computed and save its index (i.e., row)

[n,ratioMin] = min(ratioTest)

produces

n = 500 and ratioMin = 1. This implies the first constraint equation is the pivot row (ratioMin).

Step 4: perform a row operation using constraint (1) (row for x_3) to eliminate the coefficient in the z-row for x_2 . This can be done by multiplying row for x_3 by 1735 and adding it to row (1). Let nRow1 be the modified row 1 of matrix A.

>> format shorte

>> nRow1=A(2,:)*1735+A(1,:)

produces a new row 1 (n1) as follows:

nRow1 = [1.0000e+00 -2.9012e+03 0 1.7350e+03 0 0 8.6750e+05];

This new row can be inserted in the second table of the Simplex Method as follows:

Basic Variable	Z	x_1	x_2	x_3	x_4	x_5	RHS
	1	-2.90E+03	0	1.74E+03	0	0	8.68E+05
x_2	0	-3/4	1	1	0	0	500
x_4	0	2/3	1	0	1	0	1067
x_5	0	3	1	0	0	1	2700

Step 5: perform a row operations using constraint (1) (row for x_3) to eliminate the coefficients in the pivot column (column for x_2) for rows x_4 and x_5 . This can be done by multiplying the first constraint row by (-1) and adding it to rows 3 and 4 of the modified matrix A.

Let the new row 3 new called nRow3.

nRow3=A(2,)*(-1)+A(3,:);

nRow3 = [0 1.4167e+00 0 -1.0000e+00 1.0000e+00 0 5.6700e+02];

Let the new row 4 new called nRow4.

$nRow4 = A(2,:) * (-1) + A(4,:);$

$nRow4 = [0 \ 3.7500e+00 \ 0 \ -1.0000e+00 \ 0 \ 1.0000e+00 \ 2.2000e+03];$

The new modified matrix A becomes:

A <4x7 double>								
	1	2	3	4	5	6	7	
1	1	-2.90...	0	1735	0	0	867500	
2	0	-0.7500	1	1	0	0	500	
3	0	1.4167	0	-1	1	0	567	
4	0	3.7500	0	-1	0	1	2200	

This matrix is written in Table form as:

Basic Variable	Z	x_1	x_2	x_3	x_4	x_5	RHS
	1	-2.90E+03	0	1.74E+03	0	0	8.68E+05
x_2	0	-3/4	1	1	0	0	500
x_4	0	1.4167	0	-1	1	0	567
x_5	0	3.75	0	-1	0	1	2200

This table is not optimal because the solution for the objective function Z can be improved by introducing variable x_1 into the solution. Steps 1-5 can be repeated once again to arrive to an optimal solution in the next table.