Eric Wan - ezw23@drexel.edu - MatLab Exam Version 5

Table of Contents

Problem 1 - A

```
C = 7; % set C
D = -7; % set D
E = 0; % set E
F = -2; % set F

% part i
A1 = [C 0 0 D -3; 2 0 1 2 -2; 5 0 0 7 3; 7 1 0 5 -1; 2 E 1 2 F]; % set
 A1
B = [15; -7; 5; 4; -7]; % set B
augA1B = [A1, B] % combine and form augmented matrix

% part ii
rrefA1B = rref(augA1B) % rref of A1B
%{
solution is consistent as there is no row of 0s in the RREF
 coefficeient
matrix A1 with a nonzero in the "solution" b matrix as seen in row 5
 with
the entire row being 0
%}

% part iii
free = rrefA1B(:,5) % displaying free variable column from rref
% column 5, or X5, is the free variable

% part iv
A1^(-1) % computing inverse
% The coefficient matrix is not invertible
```

```
augA1B =

     7     0     0    -7    -3    15
     2     0     1     2    -2    -7
     5     0     0     7     3     5
     7     1     0     5    -1     4
     2     0     1     2    -2    -7
```

```
rrefA1B =

    1.0000         0         0         0         0    1.6667
         0    1.0000         0         0   -3.1429   -5.2857
         0         0    1.0000         0   -2.8571   -9.3810
         0         0         0    1.0000    0.4286   -0.4762
         0         0         0         0         0         0


free =

         0
   -3.1429
   -2.8571
    0.4286
         0

Warning: Matrix is singular to working precision.

ans =

   Inf   Inf   Inf   Inf   Inf
   Inf   Inf   Inf   Inf   Inf
   Inf   Inf   Inf   Inf   Inf
   Inf   Inf   Inf   Inf   Inf
   Inf   Inf   Inf   Inf   Inf
```

Problem 1 - B

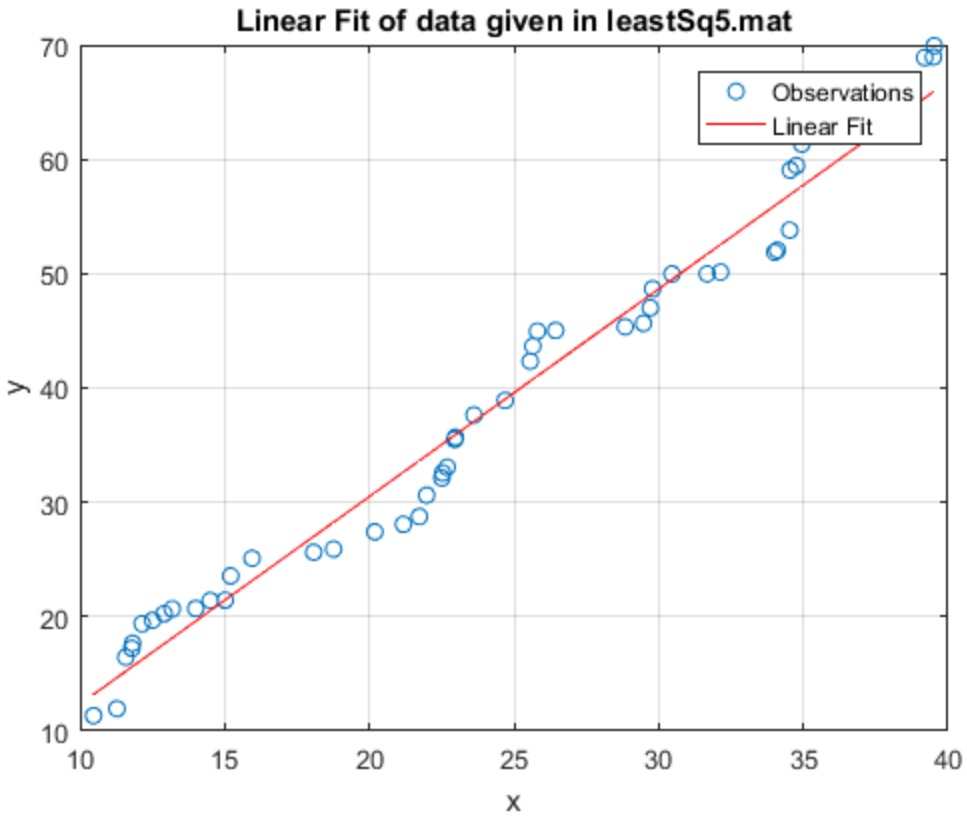
```
C = 5; % set C
D = 2; % set D
E = 3; % set E
F = 4; % set F
A2 = [C 0 0 D -3; 2 0 1 2 -2; 5 0 0 7 3; 7 1 0 5 -1; 2 E 1 2 F];

% part i
inv(A2) % computing inverse

% part ii
I = eye(5); % setting 5x5 identity matrix
A2I = [A2, I]; % setting AI augmented matrix
rref(A2I) % rref A2I

% part iii
gA2 = A2;
gA2(1,:) = gA2(1,:) - gA2(3,:);
gA2(4,:) = gA2(4,:) - gA2(3,:);
gA2(4,:) = gA2(4,:) - gA2(2,:);
gA2(3,:) = gA2(3,:) - 2 * gA2(2,:);
gA2(2,:) = gA2(2,:) - 2 * gA2(3,:);
```

```
gA2(5,:) = gA2(5,:) - 2 * gA2(3,:);
gA2(5,:) = gA2(5,:) - 3 * gA2(4,:);
gA2(5,:) = gA2(5,:) / 8;
gA2(2,:) = gA2(2,:) - 5 * gA2(5,:);
gA2(3,:) = gA2(3,:) + 2 * gA2(5,:);
gA2(3,:) = gA2(3,:) + gA2(1,:);
gA2(1,:) = gA2(1,:) / (-5);
gA2(5,:) = gA2(5,:) - gA2(1,:);
gA2(2,:) = gA2(2,:) + 9 * gA2(1,:);
gA2(2,:) = gA2(2,:) / (-2.7);
gA2(1,:) = gA2(1,:) - 1.2 * gA2(2,:);
gA2(5,:) = gA2(5,:) + 1.7 * gA2(2,:);
gA2(4,:) = gA2(4,:) + gA2(5,:);
gA2(4,:) = gA2(4,:) + 4 * gA2(1,:);
gA2(4,:) = gA2(4,:) + 2 * gA2(2,:);
sol = [gA2(3,:); gA2(4,:); gA2(5,:); gA2(1,:); gA2(2,:)]


ans =

    1.0000   -0.2500    0.2500   -0.7500    0.2500
   -1.3333    0.1296   -0.6111    1.3889   -0.1296
    1.3333    0.4815    0.4444   -1.5556    0.5185
   -1.0000    0.2778   -0.1667    0.8333   -0.2778
    0.6667   -0.2315    0.3056   -0.6944    0.2315


ans =

  Columns 1 through 7

    1.0000         0         0         0         0    1.0000   -0.2500
         0    1.0000         0         0         0   -1.3333    0.1296
         0         0    1.0000         0         0    1.3333    0.4815
         0         0         0    1.0000         0   -1.0000    0.2778
         0         0         0         0    1.0000    0.6667   -0.2315

  Columns 8 through 10

    0.2500   -0.7500    0.2500
   -0.6111    1.3889   -0.1296
    0.4444   -1.5556    0.5185
   -0.1667    0.8333   -0.2778
    0.3056   -0.6944    0.2315


sol =

    1.0000         0         0         0         0
         0    1.0000         0         0         0
         0         0    1.0000         0    0.0000
         0         0         0    1.0000   -0.0000
         0         0         0         0    1.0000
```

Problem 2

part i

```
load('leastSq5.mat');

% part ii
type linefit.m

% part iii
[N, XT, D, YT, beta_est, Y_est] = linefit(X, Y); % running linefit
 funct

% part iv
% done in the linefit function

% part v
plot(X, Y, 'o'), hold on, grid on % plotting data points of
 pts_setA(1)
plot(X, Y_est, 'r'); % plotting line of best fit
legend('Observations','Linear Fit'); % labeling legend
xlabel('x'); % labeling x axis
ylabel('y'); % labeling y axis
title('Linear Fit of data given in leastSq5.mat');

% part vi
err = YT - Y_est; % calculating error of each Y value
RMSEL = (err'*err/N)^0.5 % calculating RMS error


function[N, XT, D, YT, beta_est, Y_est] = linefit(X, Y)
N = length(X);
XT = X';
D = [ones(N,1), XT];
YT = Y';
beta_est = (D'*D)^-1*(D'*YT);
Y_est = D*beta_est;
end

RMSEL =

    2.4916
```



Published with MATLAB® R2016a