Worksheet 5

Steve Guerrero

Part 3

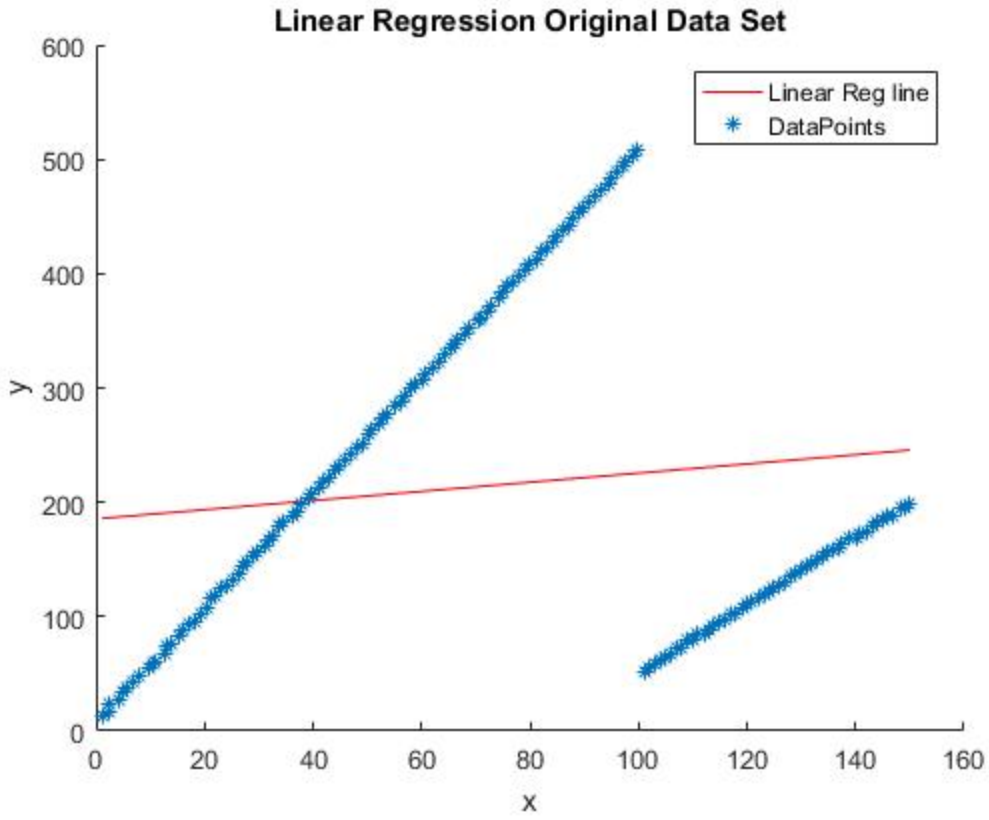
RANSAC algorithm with Linear Regression

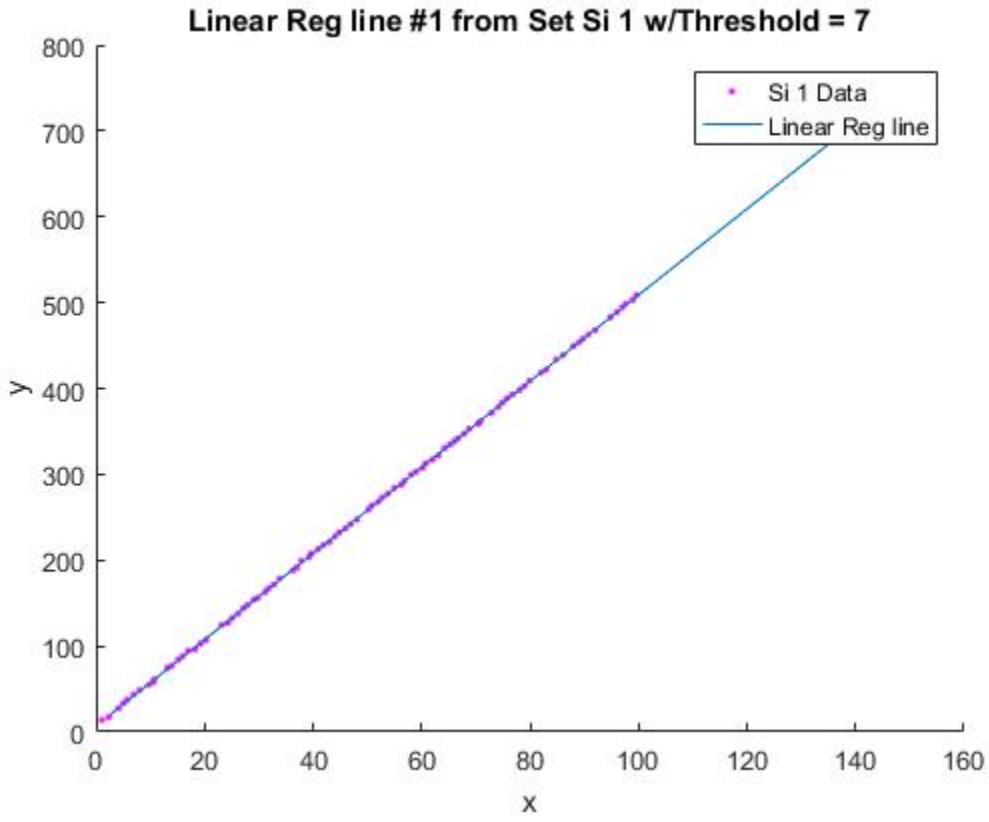
```
clear
close all
clc
%Original Data set
t = load('featureTwo.mat');
[len,~] = size(t.dataPoints(:,2));
orig = len;
% creating matrices from original data set
A(:,1) = t.dataPoints(:,1);
A(:,2) = t.dataPoints(:,2);

% run linear reg the first time
old_stuff = Linear_reg(A(:,1),A(:,2),0);

% get the slope of the line
m = old_stuff(1);
b = old_stuff(2);

% Plotting linear regression line from original data set
yL = ((m*A(:,1)) + b);
figure
hold on
title('Linear Regression Original Data Set')
xlabel('x')
ylabel('y')
plot(A(:,1),yL,'r-', A(:,1),A(:,2), '*');
legend('Linear Reg line', 'DataPoints');
hold off
% Setting threshold for line
threshold = 7;

% initalization
jj = 0;
m1 = ones(16,1);
b1 = ones(16,1);
p = ones(16,1);
d = ones(orig,1);
% find the max distance from points to line and label dp
% find 16 difffrent lines and extract set from best fitting
while(1)
jj = jj + 1;
[len,~] = size(A(:,2));
    for k = 1:16
        % choosing 2 random points from data set
```

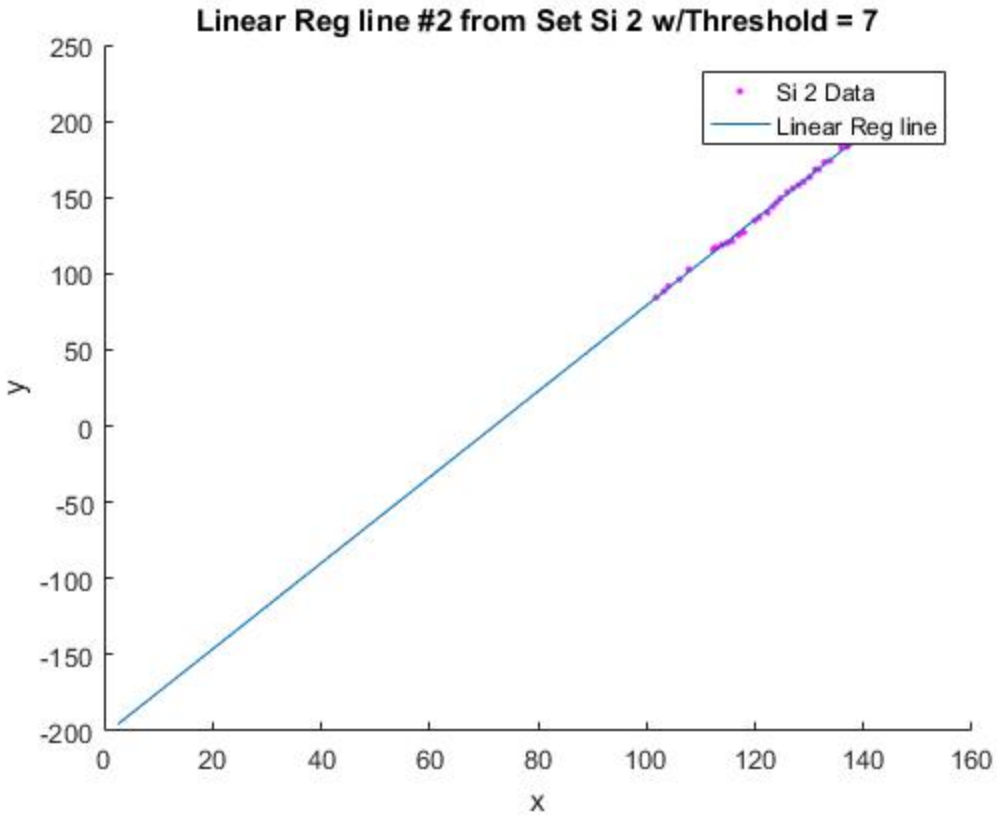
```
        ii = randi(len,1,2);
        while(ii(1) == ii(2))
           ii = randi(len,1,2);
        end

        x1 = A(ii(1),1);
        y1 = A(ii(1),2);
        x2 = A(ii(2),1);
        y2 = A(ii(2),2);

        m1(k) = (y2-y1)/(x2-x1);
        b1(k) = y1 - (m1(k)*x1);


        cnt = 0;
        for i = 1:len
            % Find distances from random line to datapoints

            d(i) = (abs((m1(k)*A(i,1)) + b1(k) - A(i,2)) ^ 2);

            %find points below threshold and place in Si
            if(d(i) < threshold)

                S(k).points(i + cnt) = i;

            else
                cnt = cnt - 1;
            end

        end
        % length f each Si vector
        p(k) = length(S(k).points);

    end

    % from first iteration check for line with most datapoints (best
fit)
    [best_line, ind] = max(p);
    % include the dataset into new Si
    Si = A(S(ind).points(1:best_line),:);
    A1 = setxor(A(:,1),Si(:,1));
    A2 = setxor(A(:,2),Si(:,2));
    A = [A1,A2];
    % clear old Si vector
    for k = 1:16
        S(k).points = [];
    end

    % *running linear regression on new data set Si*
    newer_stuff = Linear_reg(Si(:,1),Si(:,2),jj);
    m1 = newer_stuff(1);
    b1 = newer_stuff(2);
```

```
    y_b = m1 * A1 + b1;


    % Plot the line found from RANSAC and dataset that belongs with
 line
    figure(jj+1)
    hold on
    plot(Si(:,1),Si(:,2),'m.')
    plot(A1,y_b)
    xlabel('x')
    ylabel('y')
    legend(['Si ',num2str(jj),' Data' ],'Linear Reg line')

    title(['Linear Reg line #',num2str(jj),' from Set Si
 ',num2str(jj), ' w/Threshold = ', num2str(threshold)])
    hold off
    stop = length(A1);
    % End of do while loop
    % Ends on condition that less than 25% of points remain (outliers)
    if(stop < (.25*orig))
        break;
    end

end
fprintf('\nNumber of lines found = %i \n' , jj);
fprintf('\nNumber of Outliers from Original Dataset = %i \n' , stop);
```

```
Equation of line original set is: Y = 0.400387X + 185.521744
Equation of line 1 is: Y = 5.018018X + 6.995625
Equation of line 2 is: Y = 2.819357X + -202.762175

Number of lines found = 2

Number of Outliers from Original Dataset = 30
```


Linear Regression Original Data Set
Linear Reg line
DataPoints
y
x
0
100
200
300
400
500
600
0
20
40
60
80
100
120
140
160


Linear Reg line #1 from Set Si 1 w/Threshold = 7
Si 1 Data
Linear Reg line
y
x
0
100
200
300
400
500
600
700
800
0
20
40
60
80
100
120
140
160



Published with MATLAB® R2016a