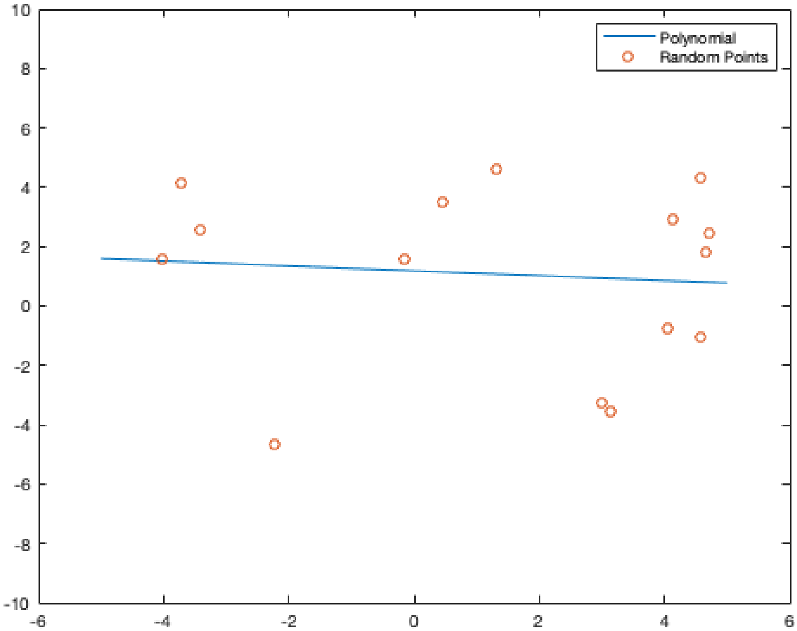
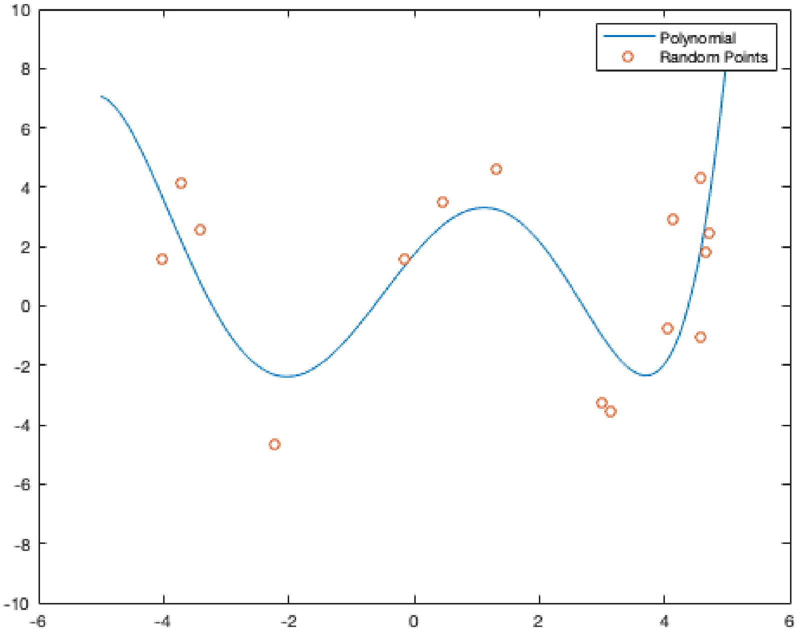
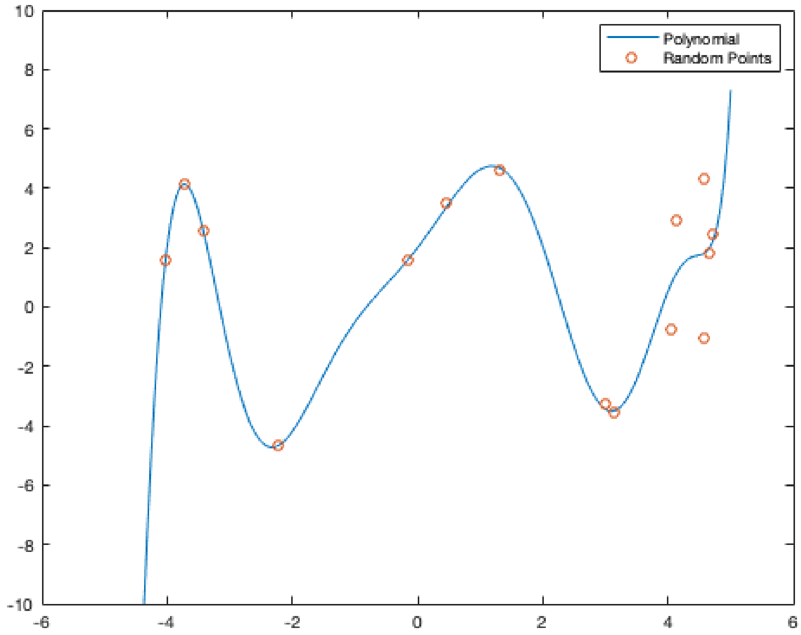
```
clear all
close all

% Generating random data points. Only do this once then record the
% set the values

nmbrPts = 15; % Number of points you want

%use the two following lines to get random values for x and yd
%x = -5 + 10 * rand(nmbrPts, 1); % Random x values in range [-5, 5]
%y = -5 + 10 * rand(nmbrPts, 1);

% Setting values from one random itteration
x =[3.1472;4.0579;-3.7301;4.1338;1.3236;-4.0246;-2.2150;0.4688;4.5751;4.6489;-3.4239;4.7059;4.5717;-0.1462;3.0028];
y =[-3.5811;-0.7824;4.1574;2.9221;4.5949;1.5574;-4.6429;3.4913;4.3399;1.7874;2.5774;2.4313;-1.0777;1.5548;-3.2881];
points=[x y];

% Setting which degrees to test for
PolyDegrees = [1, 5, 10];

% Setting up the error
errorVCTR = zeros(1, numel(PolyDegrees));

% for loop to test each itteration
for i = 1:numel(PolyDegrees)
    degree = PolyDegrees(i);
    B = ones(nmbrPts, degree + 1);

    % for loop to set the A matrix in Ax=b
    for j = 2:(degree + 1)
        B(:, j) = x.^(j - 1);
    end

    % Finding the unique x vector in Ax=b using ((A'*A)^-1)*A'*b (y=b)
    xANS =inv(B'*B)*B'*y;


    % Compute least squares error
    yBstFit = B * xANS;
    errorVCTR(i) = norm(y-yBstFit);


    % Using step size of 0.01 to generate the points for polynomial
    polypointsx = [];
    polypointsy = [];
    for i=-5:0.01:5
        polypointsx = [polypointsx i];
        for j = 2:(degree + 1)
            TempPtsPoly(j) = i.^(j - 1);
        end
        polypointsy = [polypointsy xANS(1)+sum(sum(TempPtsPoly*xANS))];
    end

    % Ploting polynomial
    figure
    plot(polypointsx,polypointsy)
    ylim([-10 10]) % Setting limit so you can see the points
    hold on
    xlim([-6 6])
    scatter(x,y) % Plotting the random points
    legend('Polynomial', 'Random Points')
end

% Determining the best fitting polynomial
[min_error, best_fit_index] = min(errorVCTR); % Finds which polynomial has least error and which degree it is
```

```
best_degree = PolyDegrees(best_fit_index);
for i=1:length(PolyDegrees)
    fprintf('Error of degree %d is %.4f.\n', PolyDegrees(i), errorVCTR(i)); % Display all errors
end
fprintf('The best fitting polynomial has degree %d with least squares error %.4f.\n', best_degree, min_error);
```

```
Error of degree 1 is 11.2825.
Error of degree 5 is 8.1113.
Error of degree 10 is 4.5404.
The best fitting polynomial has degree 10 with least squares error 4.5404.
```







Published with MATLAB® R2023b