CIEG 675 - MATLAB FOR ENGINEERING ANALYSIS (In Class #1)

- All parts of the in-class problem (a, b, and c) are combined into one coding.

- This is done to ensure that the data of y that is generated by using random numbers stay consistent for all parts.

- Coefficients of the equation of y(x) = p1 + p2*x are to be determined.

- Three methods of solving for these coefficients include: 1) By linear algebra standpoint, 2) By using polyfit function, and 3) By using regress function.

- First table generated in the output is for noise = 0. Second and third table are for noise = 50 and 100 respectively. (Unsure on how to name the table separately)

- p_1, p_2, and p_3 defined in the loop refer to the coefficients (both p1 and p2 in the original y equation) for method 1, 2, and 3 respectively.

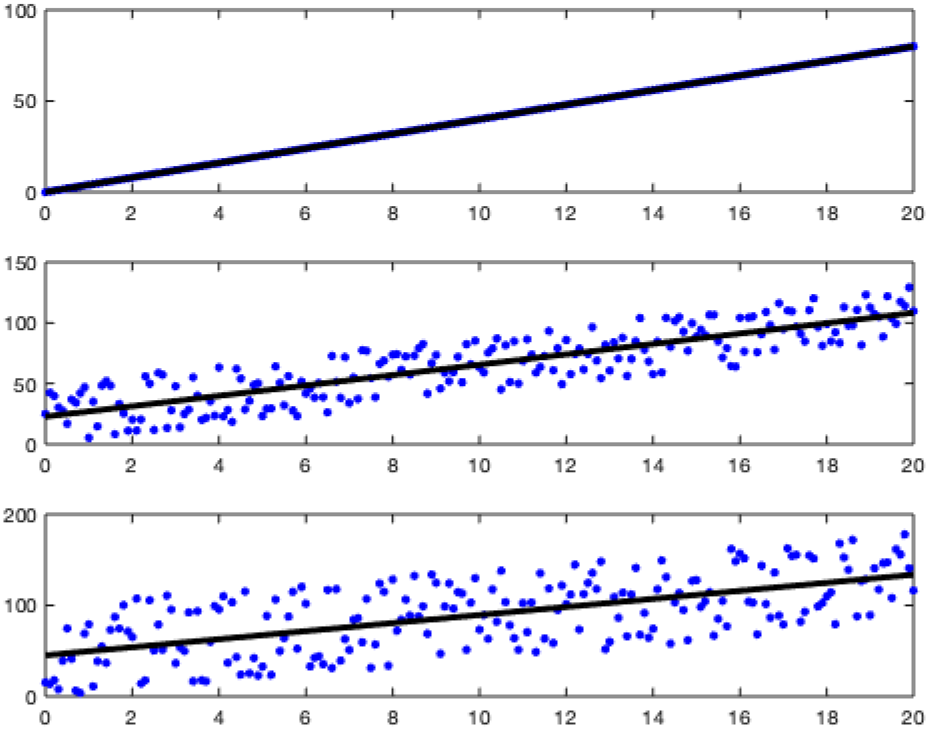
- Both coefficients from all methods are the same, by making sure that the randomly generated data for y stays consistent for all methods.

```
noise = [0,50,100];
% Three noises varied, and placed in a vector for loop purposes.
format shortG

for i = 1:length(noise)
    x = 0:0.1:20;
    y = 4*x + noise(i)*rand(1,length(x));
    % rand outputs uniformly distributed random numbers between 0 & 1.
    x = x(:);y = y(:); % To make x and y into long vectors
    A = [ones(length(x),1),x]; B = y;
    % Place constants and x into their own columns
    p_1 = A\B; % Solve for coefficients by Method 1
    p_2 = polyfit(x,y,1); % Solve for coefficients by Method 2
    p_2(:,[1,2]) = p_2(:,[2,1]);
    % Flip column 1 and 2 in order to match with the coefficients of
 other
    % methods, since polyfit outputs the coefficients of the highest
 degree
    % order of variable first.
    [p_3,pint,r,rint,stats] = regress(y,A);
    % Solve for coefficients by Method 3

    % From this point onwards are just my attempt in creating a table
 from
    % MATLAB and plotting the points to see how it looks like

    p_coeff = zeros(3,3); % Empty array to store all p values to turn
 into table
    p_coeff(:,1) = [1;2;3]; % List of methods
    p_coeff(1,[2,3]) = p_1'; % List of p1 and p2 from Method 1
    p_coeff(2,[2,3]) = p_2; % List of p1 and p2 from Method 2
    p_coeff(3,[2,3]) = p_3'; % List of p1 and p2 from Method 3
    array2table(p_coeff,'VariableNames',{'Methods','p_1','p_2'})

    subplot(3,1,i);
    plot(x,y,'b.','markersize',10)
    hold on
    eqn = p_1(1) + p_1(2).*x;
    % Doesn't matter which method to use, since they are all the same
    plot(x,eqn,'k-','linewidth',3)
end
```

```
ans =

  3×3 table

    Methods       p_1         p_2
    _______    __________    ___

       1       2.4161e-16     4 
       2       9.3621e-16     4 
       3       9.3621e-16     4 
```

```
ans =

  3×3 table

    Methods     p_1        p_2  
    _______    ______    ______

       1       22.805    4.2743
       2       22.805    4.2743
       3       22.805    4.2743
```

```
ans =

  3×3 table

    Methods     p_1        p_2  
    _______    ______    ______

       1       45.202    4.4258
       2       45.202    4.4258
       3       45.202    4.4258
```



Published with MATLAB® R2019b