Week 2 Answers

Feel free to copy the code provided into your own MATLAB scripts to check your work! Please remember that although these are answers, there are multiple ways of doing the problem(s). We encourage you to develop your own method of writing and organizing your code! - Lectures - Practice Problems `matlab %Creating Values x_values = linspace(0,100,101); y_values1 = 2.*(x_values).^2 + x_values.^3; y_values2 = x_values.^(3-x_values);`

Notes:\ When creating the values, I used the command called **linspace**. What it does is that it creates a n x 1 matrix of evenly spaced values from an initial value to a final values. By default, it will create a list of 100 values unless specified. The way that it is called is the by the following: MathWorks: linspace `matlab linspace(lower_bound,upper_bound,number_of_values)`

So in this case, I have created 101 values that are evenly spaced from 0 to 100. When you run this, you should see that it goes from 0 to 100 going up by 1. In this particular case, there is another method in which I could have called this: `matlab x_values = lower_bound:step:upper_bound` However, **linspace** is much quicker when it comes to creating decimals. Such as if we attempt to create 36 values between 0 and 8. Try it yourself with both methods!

Question 1

``` matlab %Plotting on SAME graph plot(y_values1,x_values); hold on plot(y_values2,x_values);

%Plot Labels xlabel("Independent") ylabel("Dependent") title('MATLAB Plotting') legend('Y Values 1','Y Values 2')

%Creating Limits for Axis xlim([0,5]) ylim([0,5])

hold off ```

**Notes:**\ Notice here when I wanted to graph another plot on the same graph, I had to call **hold on**. This makes it so that any further modifications I do are going to be placed on the same graph. When finished with the modifications, **hold off** is called so that when I attempt to make any further modifications, the previous graph is not modified.\ Week 2: Plotting Multiple Lines

Another thing that should be noted is how the legend works. The first label is used for the first set of data points and the second label for the second data points and so on.

Week 2: General Plot Customizing

### Question 2

``` matlab %Plotting on DIFFERENT graphs

%Graph 1 plot(y_values1,x_values); hold on %Plot Labels xlabel("Independent") ylabel("Dependent") title('MATLAB Plotting') legend('Y Values 1')

%Creating Limits for Axis xlim([0,5]) ylim([0,5])

hold off

%Graph Problem 2 plot(y_values2,x_values); hold on

%Plot Labels xlabel("Independent") ylabel("Dependent") title('MATLAB Plotting') legend('Y Values 2')

%Creating Limits for Axis xlim([0,5]) ylim([0,5])

hold off ```

**Notes:**\ Above is an example of the **hold on** and **hold off** idea mentioned before.

### Week 2 Funciton

``` matlab function ansW2fun(x_values) y_values1 = 2.*(x_values).^2 + x_values.^3; y_values2 = x_values.^(3-x_values);

%Plotting on SAME graph plot(y_values1,x_values); hold on plot(y_values2,x_values);

%Plot Labels xlabel("Independent") ylabel("Dependent") title('MATLAB Plotting') legend('Y Values 1','Y Values 2')

%Creating Limits for Axis xlim([0,5]) ylim([0,5])

hold off end ``` **Notes:**\ Once again, I've just copied the lines from the previous exercises and have the dependent values evaluated in the function as well. You can see here that the function does not return any values, however it will still display the graphs.
```