Week 7 Tutorial 6

The purpose of this program is to demonstrate how to add error bars to a plot. This will also utilize the `polyval` function for getting best fit values.

```
% Always clear workspace variables before a new tutorial or program.
clear
clc
close all  % closes all figure windows
```

Edit the code below and update the variable named **`name`** with your name for this tutorial in the code below.

```
name="";
fprintf("Output for Tutorial_07_6 run by %s.\n\n", name)
```

Figure 1

Here, we'll run through a quick example of using polyfit and polyval, unlike before where we computed the values for the best fit, we can defer to the polyval function to do that for us.

```
% Open a figure 1 window


% Given Data
xData = [1.0 1.5 3 4 6 8 10];
yData = [1.2 1.4 2.6 4.0 4.6 4.8 6.5];
```

Recall, polyfit determines the coefficients that will best fit the data.

```
% polyfit finds the coefficients of a polynomial that will best fit the data.
% The number 1 represents the order of the polynomial (1 = linear fit).
fitCoefficients = polyfit(xData,yData,1);
```

Now, we'll generate some x values to perform the best fit function on, but we'll simply pass the coefficients, along with the x data, into the polyval to get our y values.

```
xFit = 0: 0.1: 10;  % values to evaluate the fit function at
yFit = polyval(fitCoefficients,xFit); % computes yFit at each xFit value
```

And now, we plot the data, just as before.

```
% plot the fit equation and data on the same graph with blue circles for
% the data points and a red line for the fit equation
plot(xData,yData,'ob',xFit,yFit,'r')
title('Linear Fit of Data')
xlabel('X')
ylabel('Y')
legend('Data Points','Fit Equation','Location','southeast')
```

Figure 2

Now, we'll run through an example plotting error bars. Error bars are helpful in seeing just how far our data is from the best fit line. The `errorBar()` function works just like the `plot()` function however, it accepts a third parameter for the error values. In this example, we'll manually create some error values.

```
% Open a figure 1 window


% plotting data with error bars and a fit equation on the same graph
% vector of error bars for yData
yErrors = [0.3 0.3 0.6 0.5 0.4 0.5 0.6];
errorbar(xData,yData,yErrors,'bo') % blue circles with vertical error bars
```

Recall, we can use the `hold` function to plot multiple datasets on the same plot (**errorbar()** was the first technical plot function). This is useful if you prefer to break up your code, performing functions, plotting data, performing more functions, plotting more data, rather than performing all of the calculations up front and putting them all in one `plot()` function.

IMPORTANT: Just like resetting `format` or turning `diary` off, do not forget to call `hold off` to turn off the hold function. Doing so may result is undesired outcomes in future scripts.

```
hold on
% This plot line will be placed on the same plot as the previous data
plot(xFit,yFit,'r')

% We still only need to call this once, since it's for a single plot
title('Linear Fit of Data with Error Bars')
xlabel('X')
ylabel('Y')
legend('Data Points w/ Error Bars','Fit Equation','Location','southeast')
hold off
```

Example Output:

Run this tutorial from the **Command Window** and ensure your output matches the following.

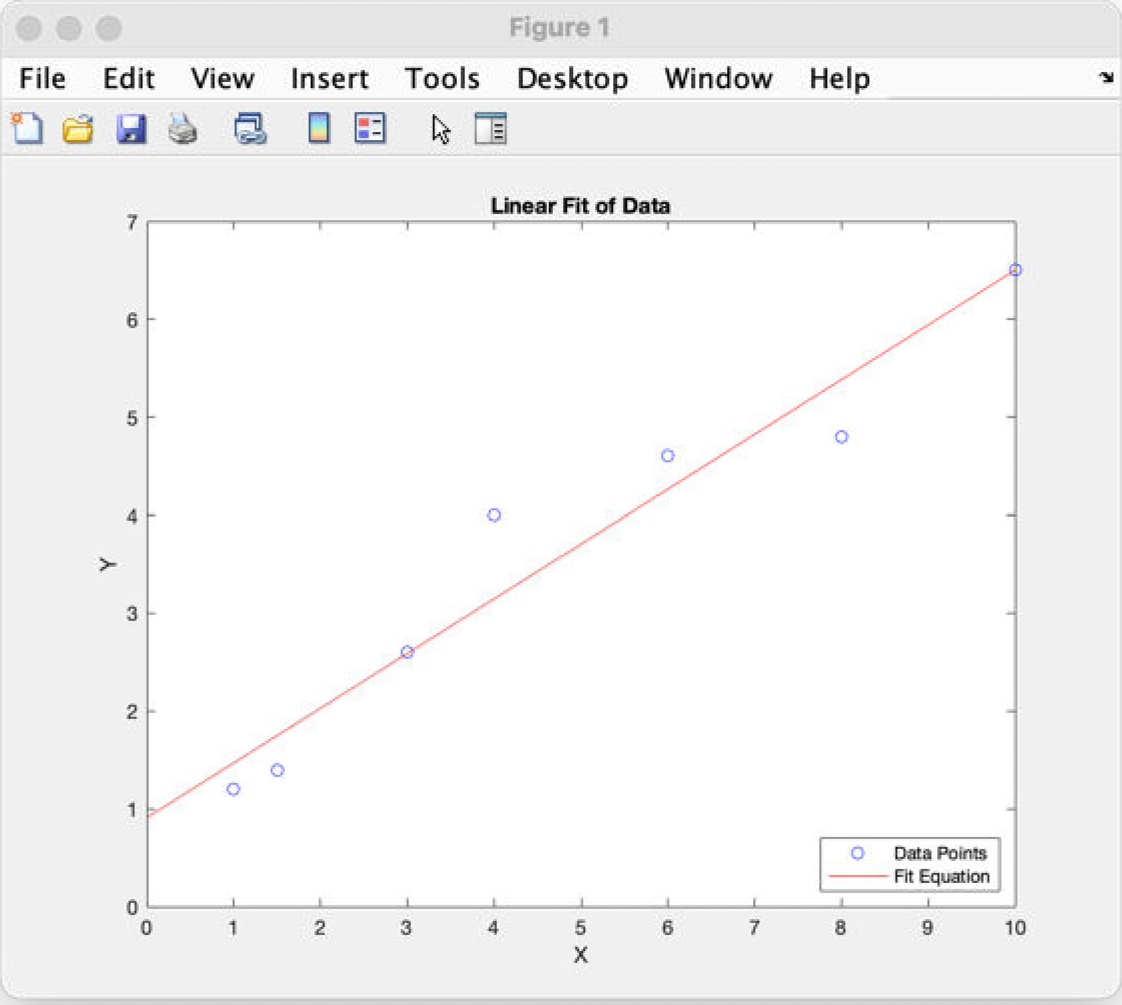
Figure 1
File
Edit
View
Insert
Tools
Desktop
Window
Help
Linear Fit of Data
Y
X
0
1
2
3
4
5
6
7
0
1
2
3
4
5
6
7
8
9
10
Data Points
Fit Equation

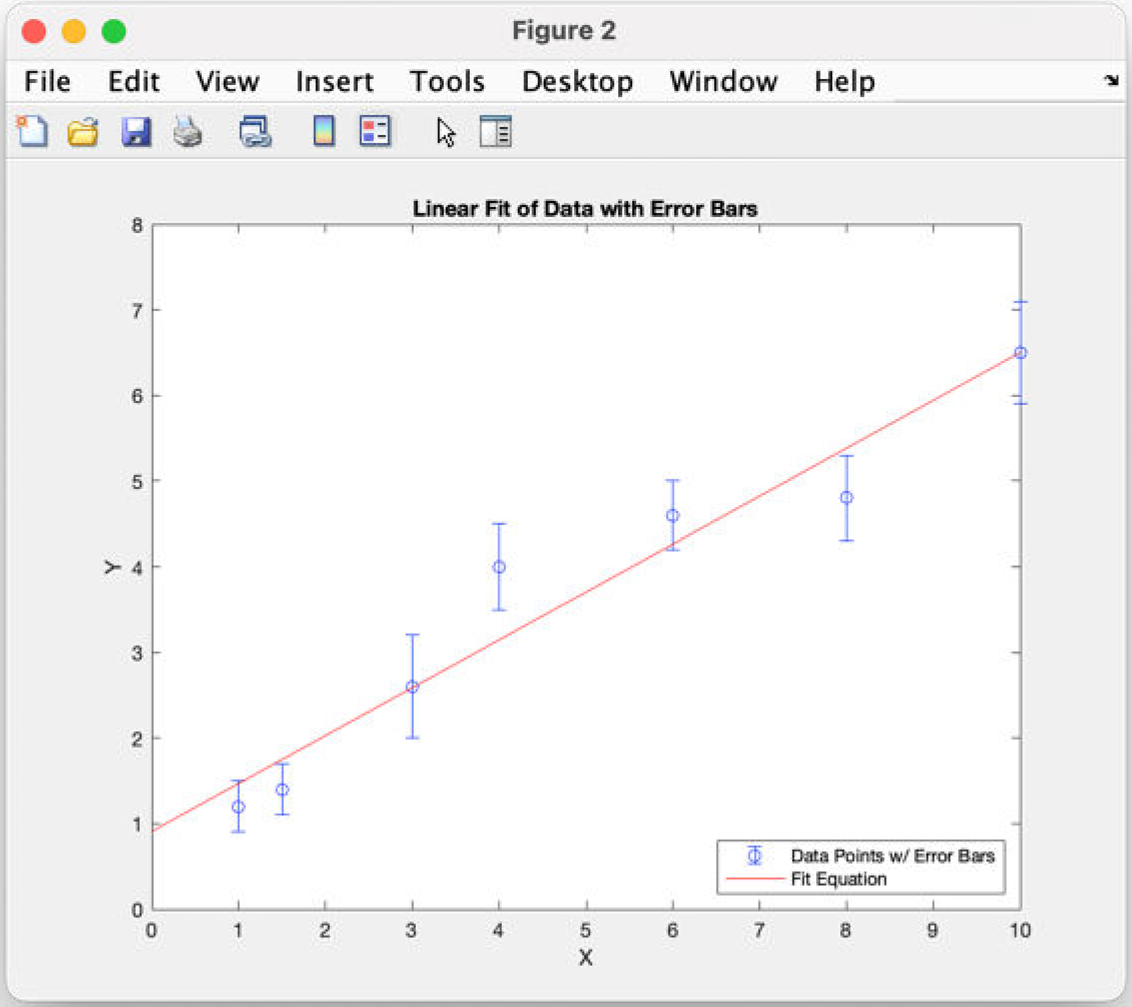
Figure 2
File
Edit
View
Insert
Tools
Desktop
Window
Help
Linear Fit of Data with Error Bars
0
1
2
3
4
5
6
7
8
Y
0
1
2
3
4
5
6
7
8
9
10
X
Data Points w/ Error Bars
Fit Equation