Week 7 MATLAB Activities Plots

Contents

Time required: 180 minutes

1. Create a MATLAB script named **Wk07Lastname.m**
2. Save all programs in this script.
3. Include your name and date at the top of the script file as comments.
4. Put a Section Break between each program.

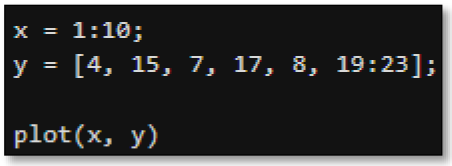
Reading

Matlab A Practical Introduction to Programming and Problem Solving (Stormy Attaway)

Sections 5.1, 5.2

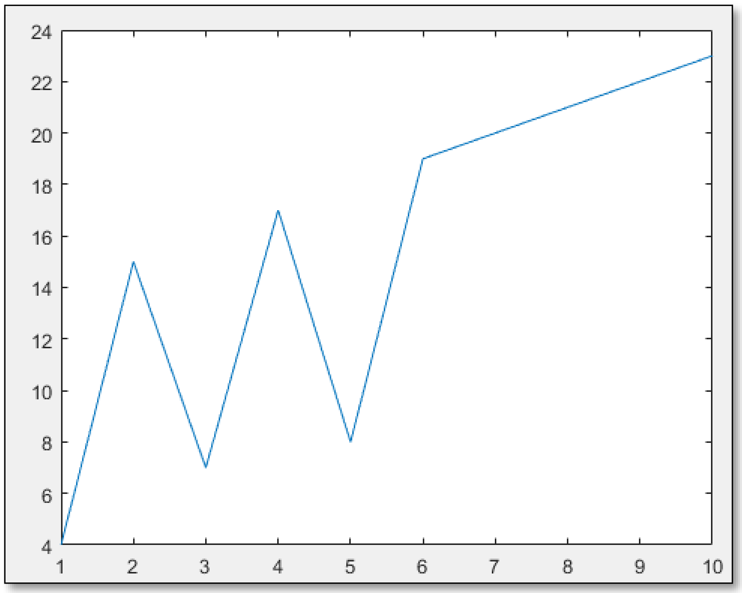
Tutorial 1: Basic Plots

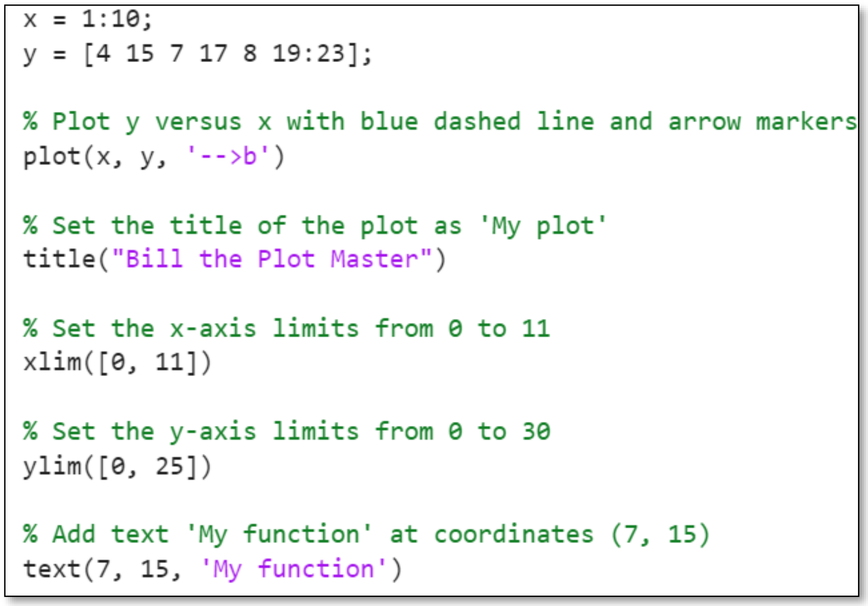
Plots in Matlab are created by simply connecting the dots. This means that you have to provide the actual coordinates of the points - both the x and y coordinates. It is not enough to simply say plot $y = x^2$, you must give exactly which x values you want squared.



```
x = 1:10;
y = [4, 15, 7, 17, 8, 19:23];

plot(x, y)
```

Example run:

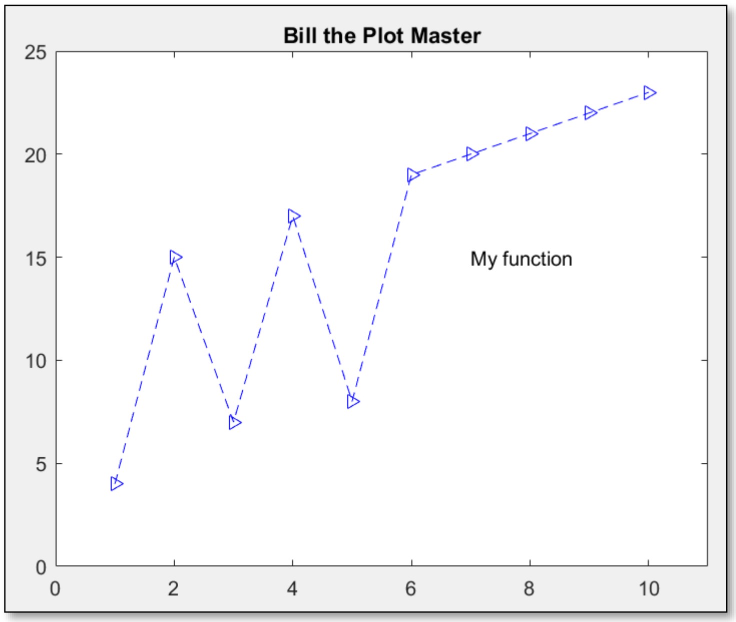




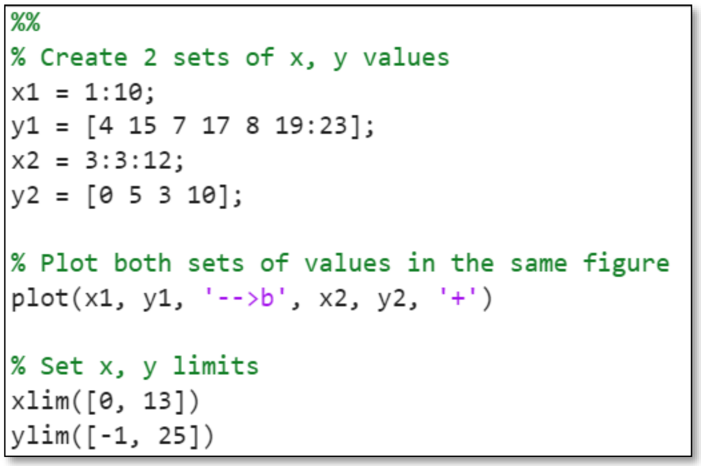
```
x = 1:10;
y = [4 15 7 17 8 19:23];

% Plot y versus x with blue dashed line and arrow markers
plot(x, y, '-->b')

% Set the title of the plot as 'My plot'
title("Bill the Plot Master")

% Set the x-axis limits from 0 to 11
xlim([0, 11])

% Set the y-axis limits from 0 to 30
ylim([0, 25])

% Add text 'My function' at coordinates (7, 15)
text(7, 15, 'My function')
```

In this example, you can see that there is a title, the limits on the graph are now 0 to 11 (for x) and 0 to 30 (for y), and there is text on the screen, starting at the coordinate (7, 5).

The symbols '-->b ' indicate that the lines should be dashed, the points marked with triangles and the lines colored blue (the default color if none is provided).

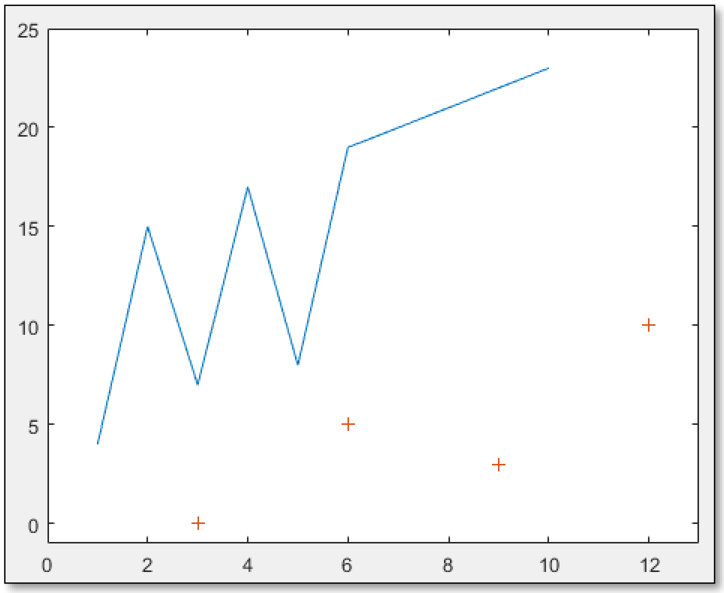


If you want to plot more than one set of data in the same figure window, there are two ways to do this. You can put both in the same plot command or by using the hold command with two plot commands.



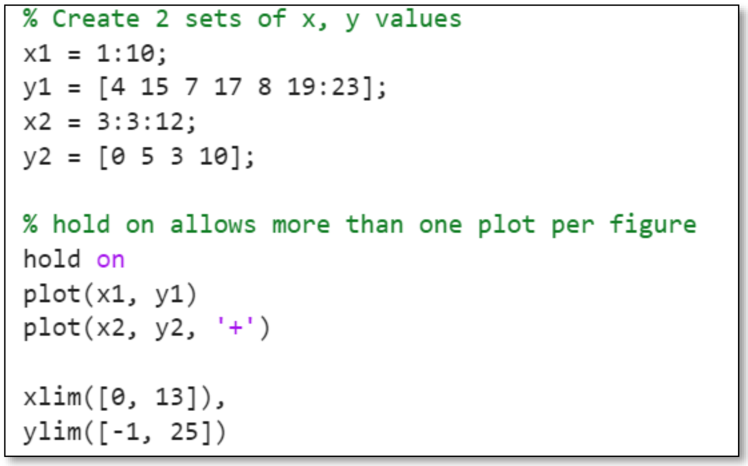
```
%%
% Create 2 sets of x, y values
x1 = 1:10;
y1 = [4 15 7 17 8 19:23];
x2 = 3:3:12;
y2 = [0 5 3 10];

% Plot both sets of values in the same figure
plot(x1, y1, '-->b', x2, y2, '+')

% Set x, y limits
xlim([0, 13])
ylim([-1, 25])
```

Example run:



From this example we can see that if you want to use the same plot command, you simply list the pairings, (with any details following each pair) x1,y1,x2,y2,x3,y3, ... The symbol '+' means that the second plot is points (not lines) marked with plus signs.

We use data from the last example, but use the hold command. This allows more clarity in the code by separating the plot commands.



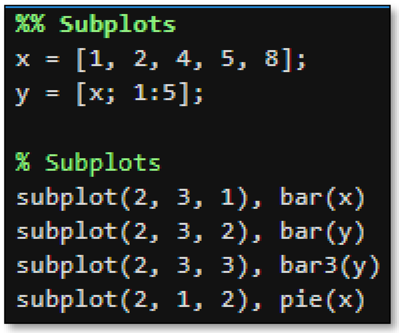
```
% Create 2 sets of x, y values
x1 = 1:10;
y1 = [4 15 7 17 8 19:23];
x2 = 3:3:12;
y2 = [0 5 3 10];

% hold on allows more than one plot per figure
hold on
plot(x1, y1)
plot(x2, y2, '+')

xlim([0, 13]),
ylim([-1, 25])
```

This plot should look exactly like the last one.

In the last example, if you didn't include the hold command, the second plot would simply overwrite the first.

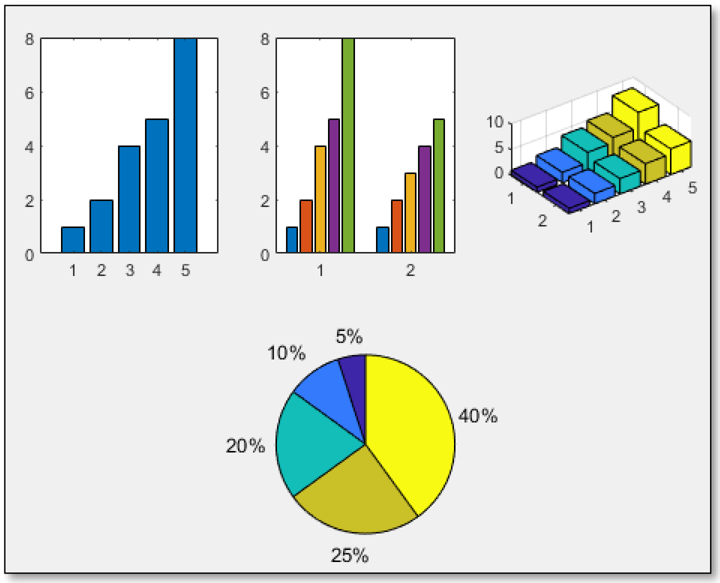
Tutorial 2: Subplots

Here we look at the subplot command as well as bar graphs and pie charts to create multiple graphs on the same figure.



```
%% Subplots
x = [1, 2, 4, 5, 8];
y = [x; 1:5];

% Subplots
subplot(2, 3, 1), bar(x)
subplot(2, 3, 2), bar(y)
subplot(2, 3, 3), bar3(y)
subplot(2, 1, 2), pie(x)
```

Example run:



In this example, the subplot on each line does two things. First, it (invisibly) subdivides the figure into rows and columns and second, it indicates where the plot will be shown (counting across the rows).

subplot(# rows, # columns, location)

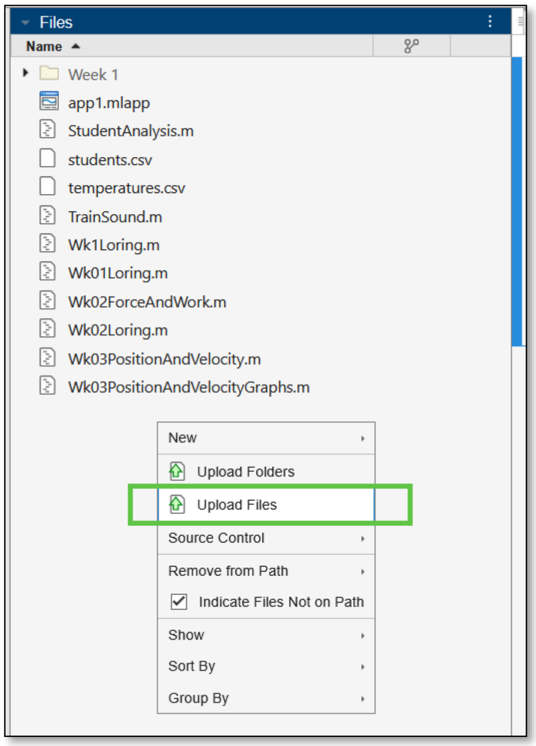
You can see in the example that even though the plot was broken up into 2 rows and 3 columns in the first three subplot lines, the fourth line re-subdivides the plot into 2 rows and 1 column so that the pie plot can take up the entire bottom of the plot. Using subplots does not require the hold command.

Tutorial 3: Plotting Weather Data from a File

Let’s learn some types of plots with data from a csv file. **temperatures.csv** is attached to this assignment.

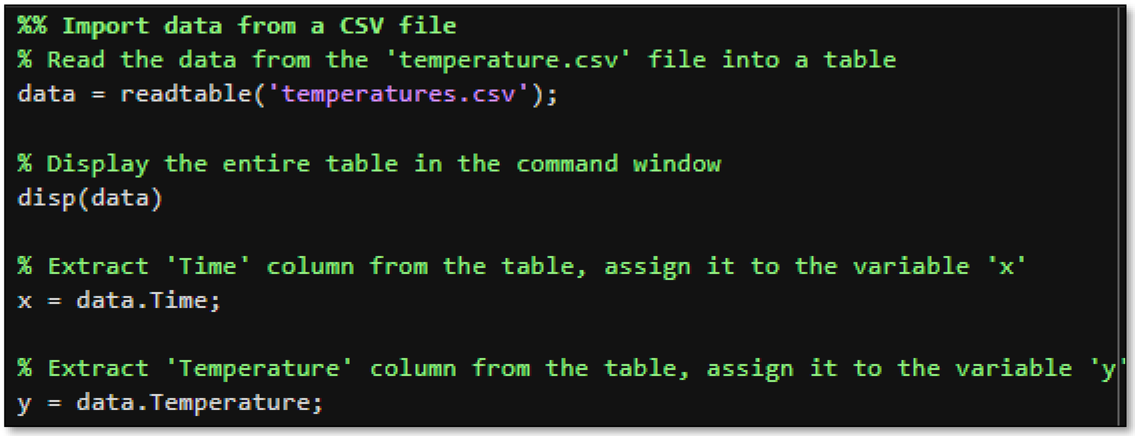
1. Download **temperatures.csv** to your computer.

2. In MATLAB → Right Click the files pane to upload the file.



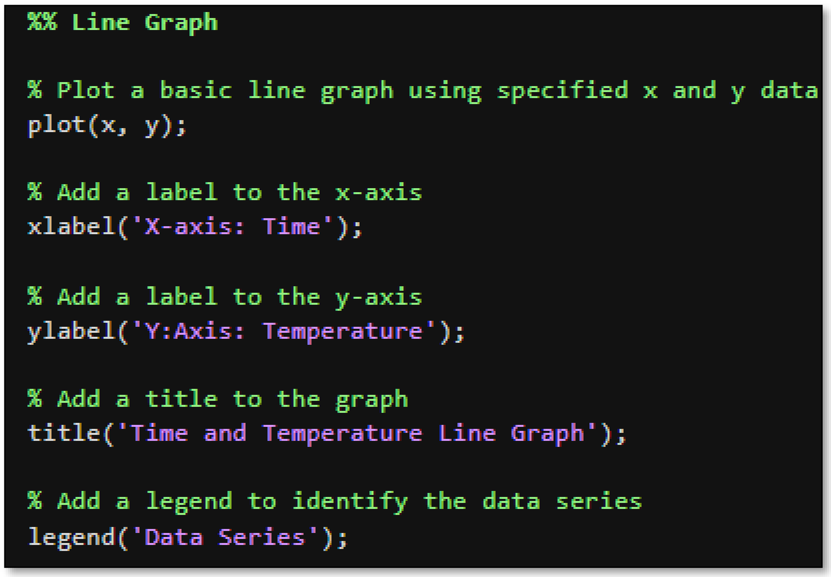
3. Browse to and select **temperatures.csv**

We are now ready to read the data from the file.



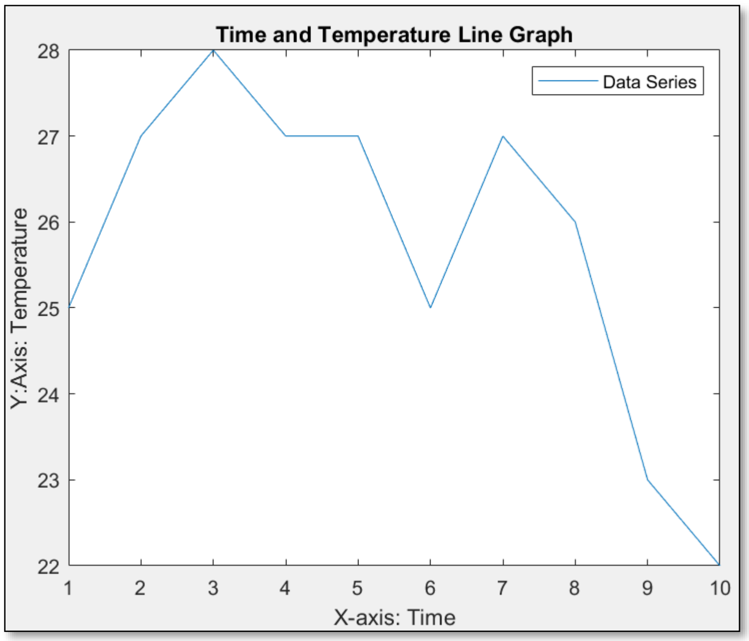
```
%% Import data from a CSV file
% Read the data from the 'temperature.csv' file into a table
data = readtable('temperatures.csv');

% Display the entire table in the command window
disp(data)

% Extract 'Time' column from the table, assign it to the variable 'x'
x = data.Time;

% Extract 'Temperature' column from the table, assign it to the variable 'y'
y = data.Temperature;
```

A line graph.

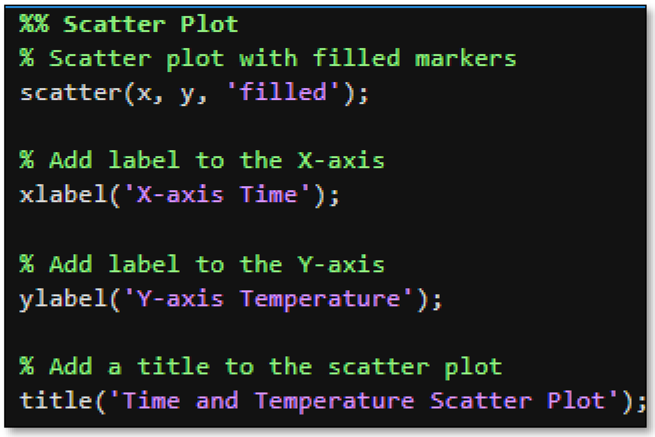


```
%% Line Graph

% Plot a basic line graph using specified x and y data
plot(x, y);

% Add a label to the x-axis
xlabel('X-axis: Time');

% Add a label to the y-axis
ylabel('Y:Axis: Temperature');

% Add a title to the graph
title('Time and Temperature Line Graph');

% Add a legend to identify the data series
legend('Data Series');
```

Example run:

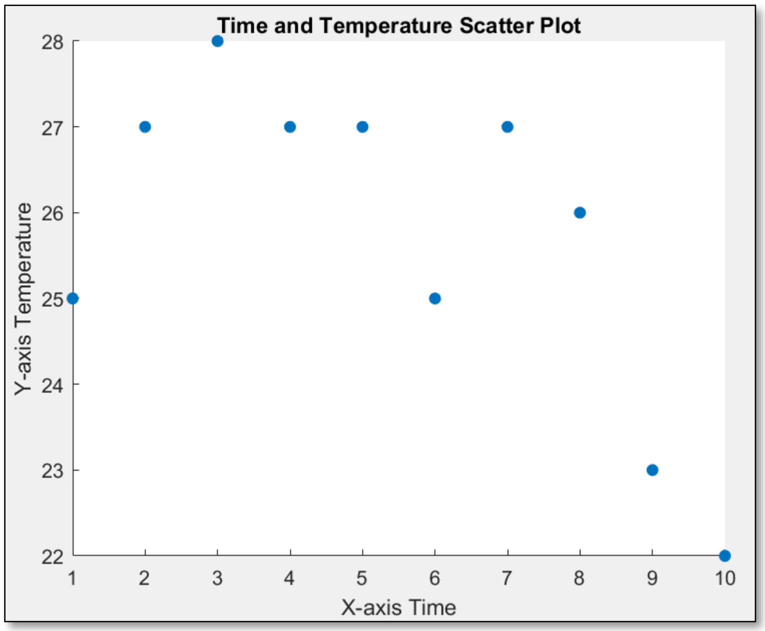


A scatter plot.

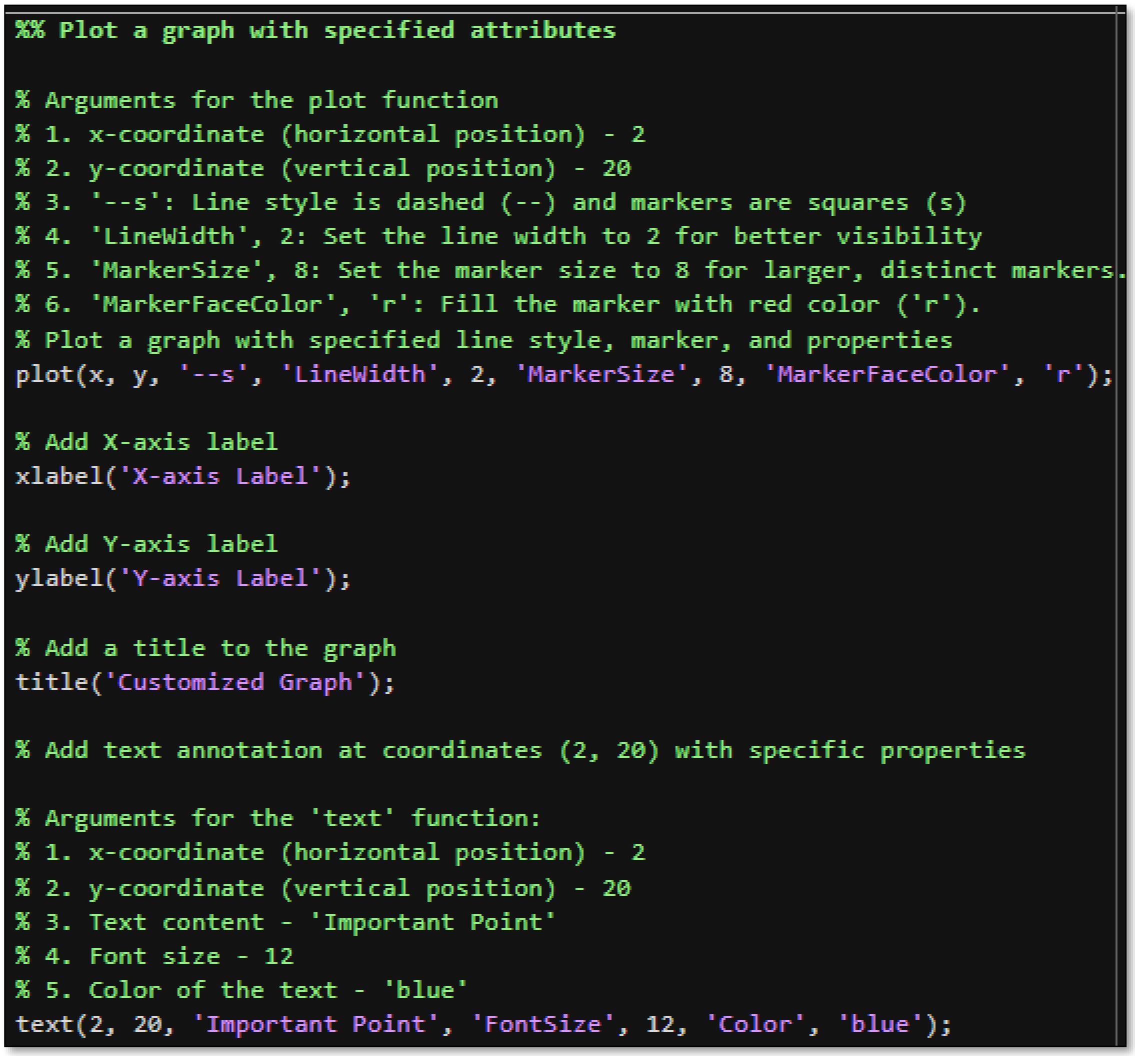


```
%% Scatter Plot
% Scatter plot with filled markers
scatter(x, y, 'filled');

% Add label to the X-axis
xlabel('X-axis Time');

% Add label to the Y-axis
ylabel('Y-axis Temperature');

% Add a title to the scatter plot
title('Time and Temperature Scatter Plot');
```

Example run:

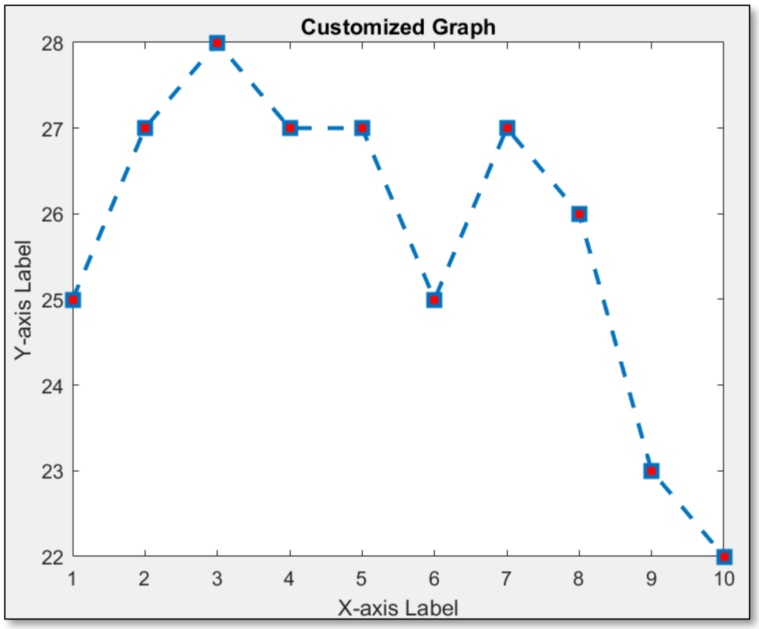


Plot a graph with specified attributes.

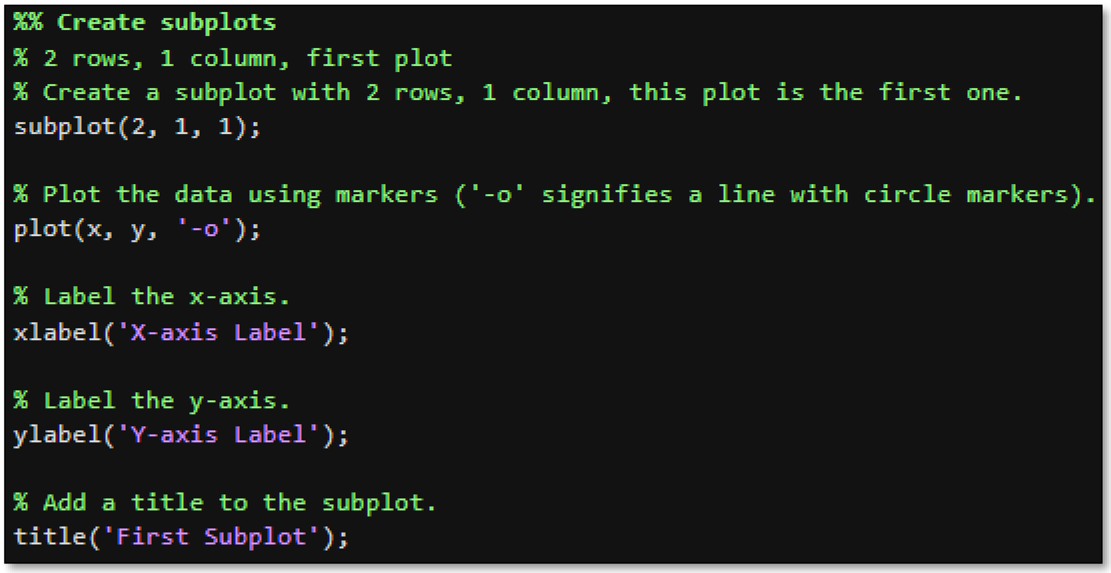


```
%% Plot a graph with specified attributes

% Arguments for the plot function
% 1. x-coordinate (horizontal position) - 2
% 2. y-coordinate (vertical position) - 20
% 3. '--s': Line style is dashed (--) and markers are squares (s)
% 4. 'LineWidth', 2: Set the line width to 2 for better visibility
% 5. 'MarkerSize', 8: Set the marker size to 8 for larger, distinct markers.
% 6. 'MarkerFaceColor', 'r': Fill the marker with red color ('r').
% Plot a graph with specified line style, marker, and properties
plot(x, y, '--s', 'LineWidth', 2, 'MarkerSize', 8, 'MarkerFaceColor', 'r');

% Add X-axis label
xlabel('X-axis Label');

% Add Y-axis label
ylabel('Y-axis Label');

% Add a title to the graph
title('Customized Graph');

% Add text annotation at coordinates (2, 20) with specific properties

% Arguments for the 'text' function:
% 1. x-coordinate (horizontal position) - 2
% 2. y-coordinate (vertical position) - 20
% 3. Text content - 'Important Point'
% 4. Font size - 12
% 5. Color of the text - 'blue'
text(2, 20, 'Important Point', 'FontSize', 12, 'Color', 'blue');
```

Example run:

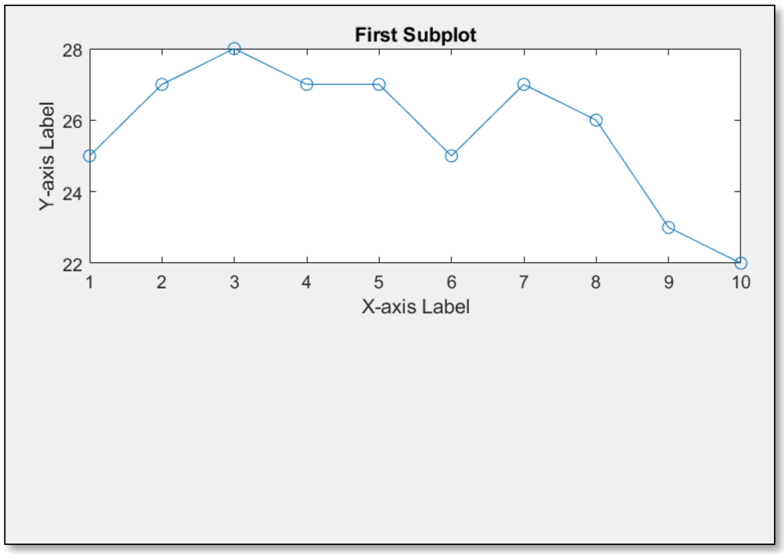


Create 2 subplots.

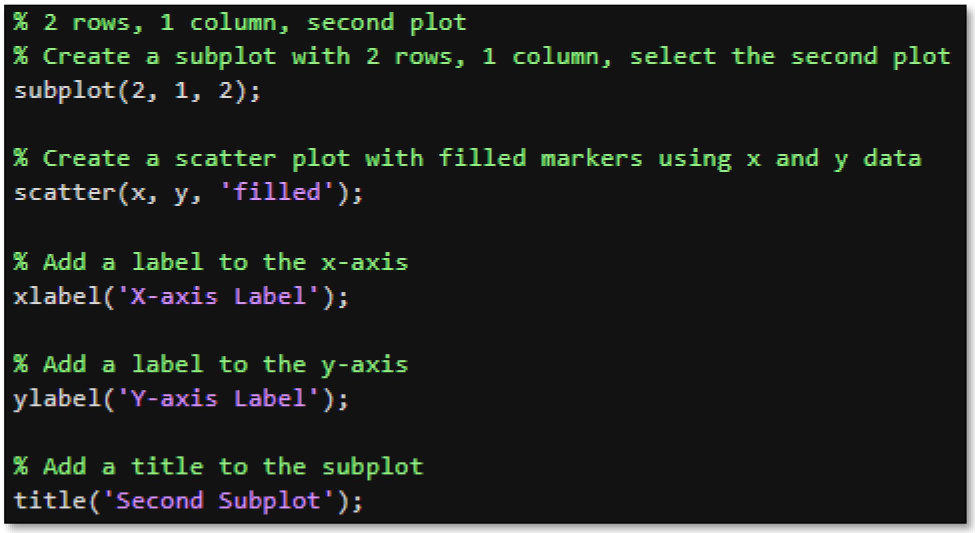


```
%% Create subplots
% 2 rows, 1 column, first plot
% Create a subplot with 2 rows, 1 column, this plot is the first one.
subplot(2, 1, 1);

% Plot the data using markers ('-o' signifies a line with circle markers).
plot(x, y, '-o');

% Label the x-axis.
xlabel('X-axis Label');

% Label the y-axis.
ylabel('Y-axis Label');

% Add a title to the subplot.
title('First Subplot');
```

Example run:

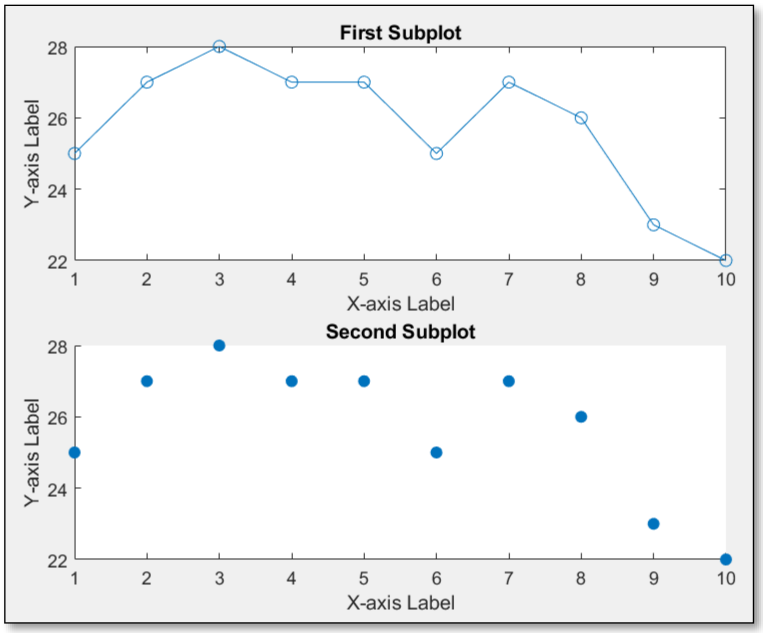


More subplots.

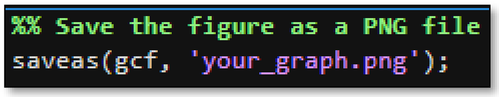


```
% 2 rows, 1 column, second plot
% Create a subplot with 2 rows, 1 column, select the second plot
subplot(2, 1, 2);

% Create a scatter plot with filled markers using x and y data
scatter(x, y, 'filled');

% Add a label to the x-axis
xlabel('X-axis Label');

% Add a label to the y-axis
ylabel('Y-axis Label');

% Add a title to the subplot
title('Second Subplot');
```

Example run:



Save the figure as a png file. You will find this file in your MATLAB file folder

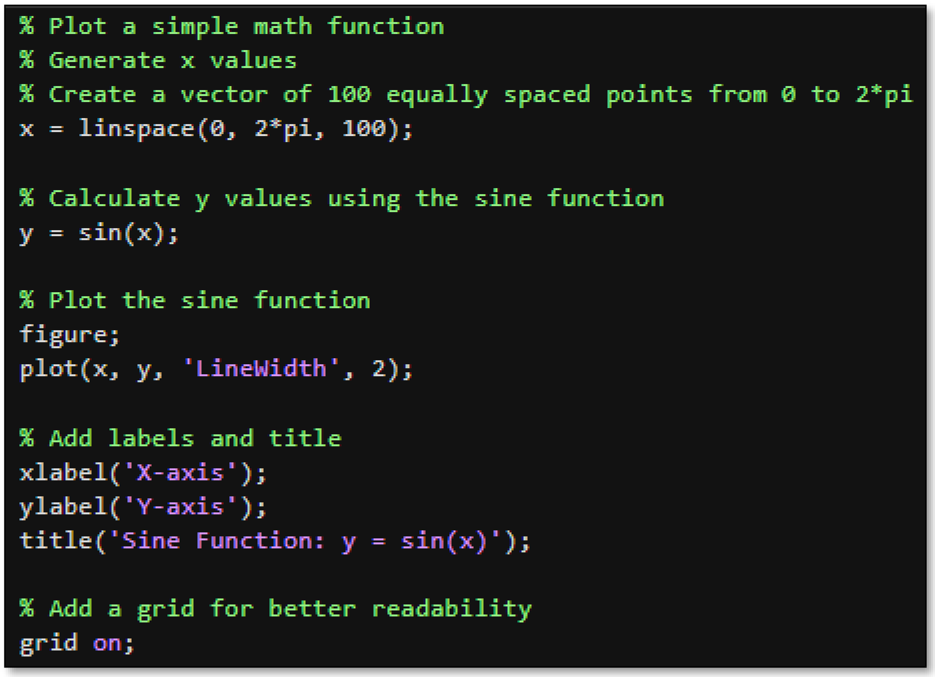


```
%% Save the figure as a PNG file
saveas(gcf, 'your_graph.png');
```

Tutorial 4: Plot a Math Function

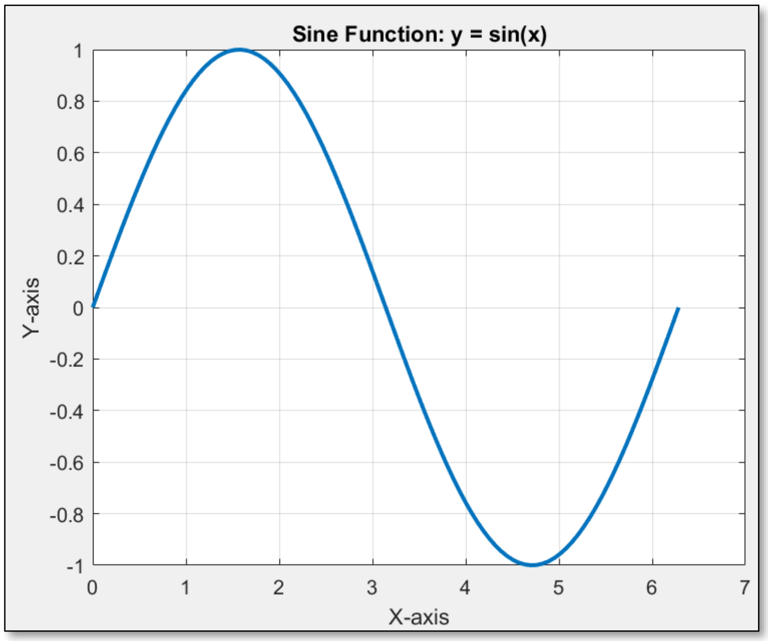
Let's plot a simple mathematical function, the sine function *y=sin(x)*

This code generates x values from 0 to 2pi, calculates the corresponding y values using the sine function. Plot the sine function using the **plot** function.



```
% Plot a simple math function
% Generate x values
% Create a vector of 100 equally spaced points from 0 to 2*pi
x = linspace(0, 2*pi, 100);

% Calculate y values using the sine function
y = sin(x);

% Plot the sine function
figure;
plot(x, y, 'LineWidth', 2);

% Add labels and title
xlabel('X-axis');
ylabel('Y-axis');
title('Sine Function: y = sin(x)');

% Add a grid for better readability
grid on;
```

Example run:



Assignment 1: Quadratic Function Plot

This assignment defines a quadratic function with specific coefficients (a, b, c), generates x values, calculates corresponding y values, and then plots the function using the plot function. The resulting graph represents a quadratic function

Create a MATLAB line plot for a quadratic function.

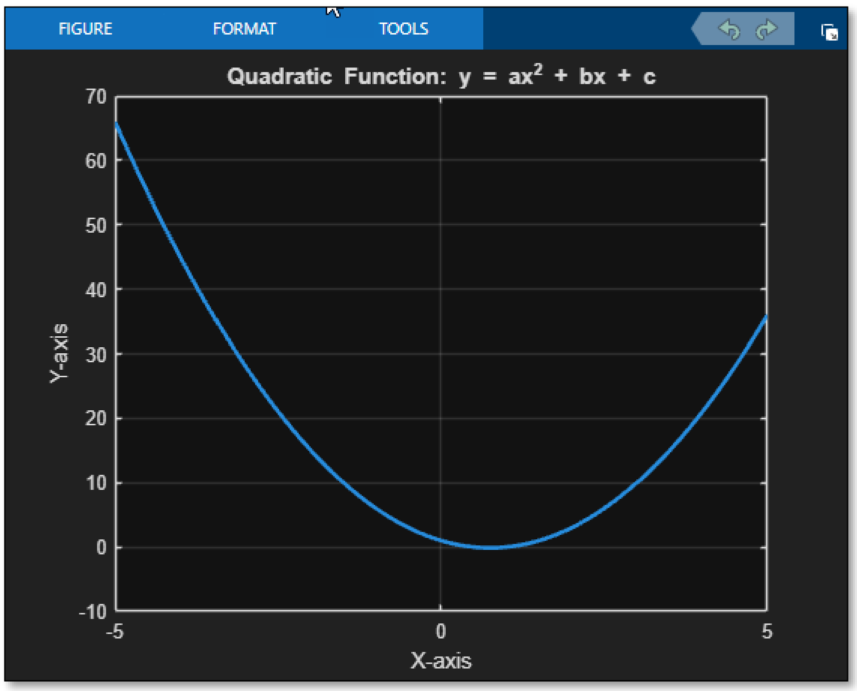
$$y = ax^2 + bx + c$$

Where a = 2, b = -3, and c = 1

TODO:

```
% Define coefficients for the quadratic function

% Generate x values
% Create a vector of 100 equally spaced points from -5 to 5
x = linspace(-5, 5, 100);

% Calculate y values using the quadratic function

% Plot the algebraic function
% If you use this format, x squared is displayed properly
title('Quadratic Function: y = ax^2 + bx + c');

% Add a grid for better readability
grid on;
```

Example run:



Assignment 2: Subplot Equations

For the functions

$$y1 = e^{x/5}$$

$$y2 = sin(x)$$

$$y3 = \sqrt{x}y1$$

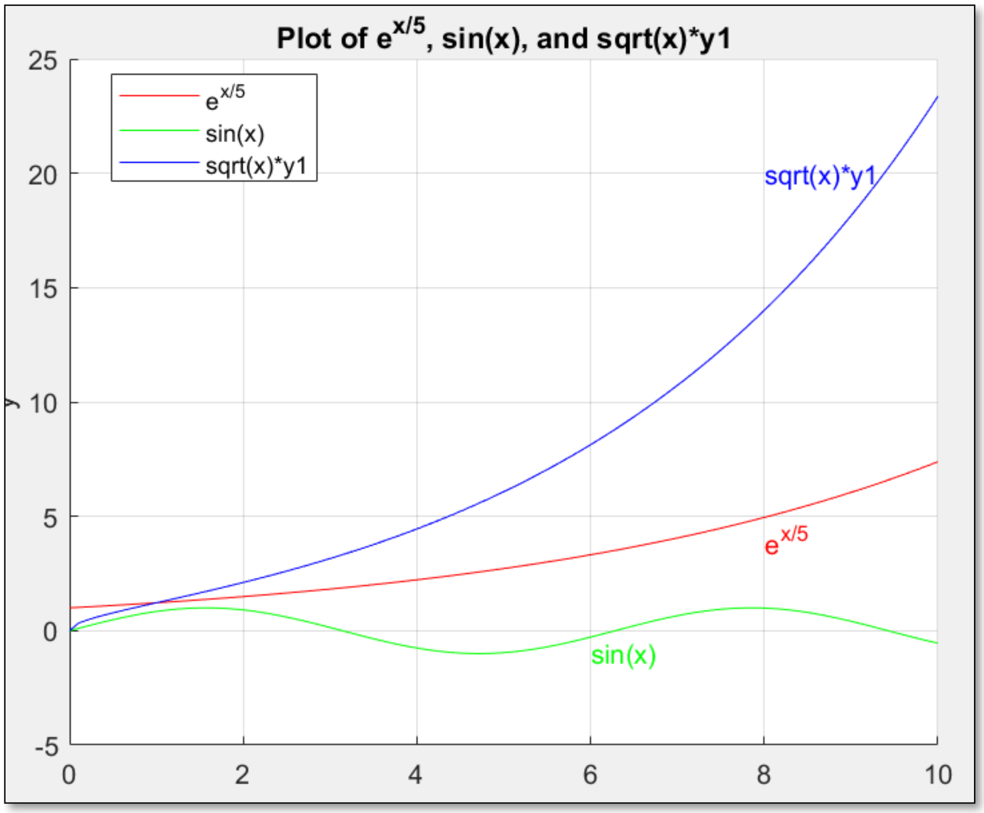
for x between 0 and 10.

Plot all three on the same plot, but with different colors (use linspace with enough points so that the curves look smooth).

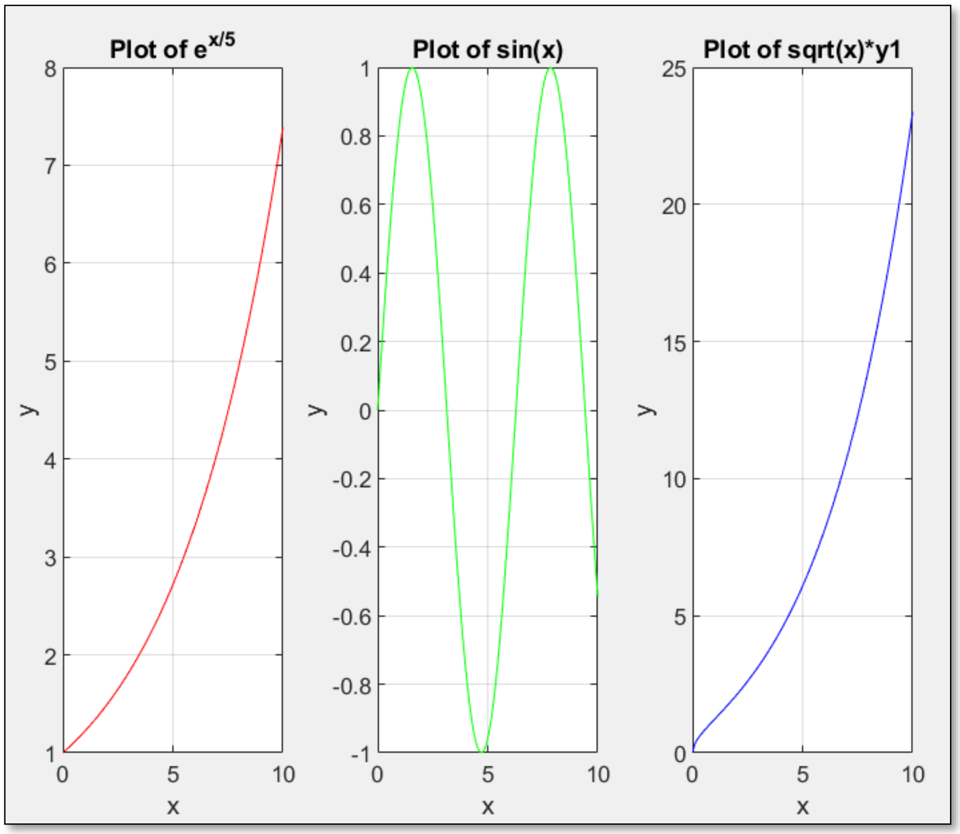
1. Put in a descriptive title and add text inside the window to label the individual graphs.

The following will allow you to display superscript in titles and labels.

```
title('Plot of e^{x/5}, sin(x), and sqrt(x)*y1');
```



2. Plot all three in the same figure window, but with each in their own subplot and each with its own descriptive title.



Assignment Submission

1. Submit properly named and commented script files.
2. Attach a screenshot of the Command Window showing the successful execution of each script.
3. Attach all to the assignment in Blackboard.