Graph Plotting in MATLAB

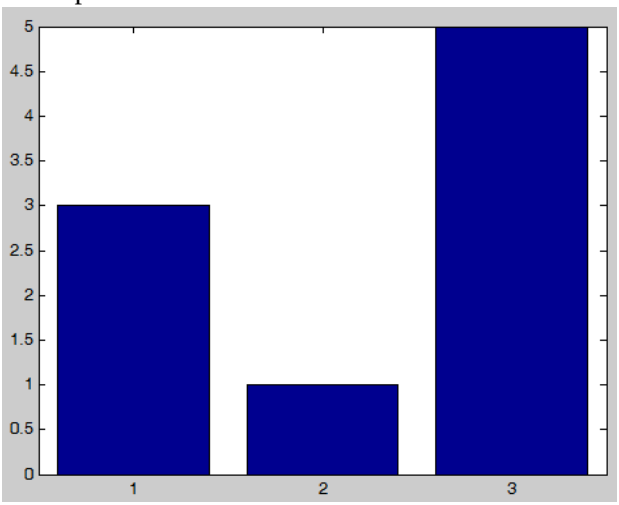
I. Creating/Closing a Figure window

```
figure                  % Opens a new figure window
figure('name', 'abc') % Name the newly opened figure window "abc"
close                   % Closes the current figure
close all               % Closes all figures
```

II. Bar charts

The `bar()` function is a very handy way to plot bar charts in MATLAB. Similar to many other built-in functions, there are many optional arguments that you can pass into the function.

```
y = [3 1 5];
bar(y); % Gives you this graph ->
```



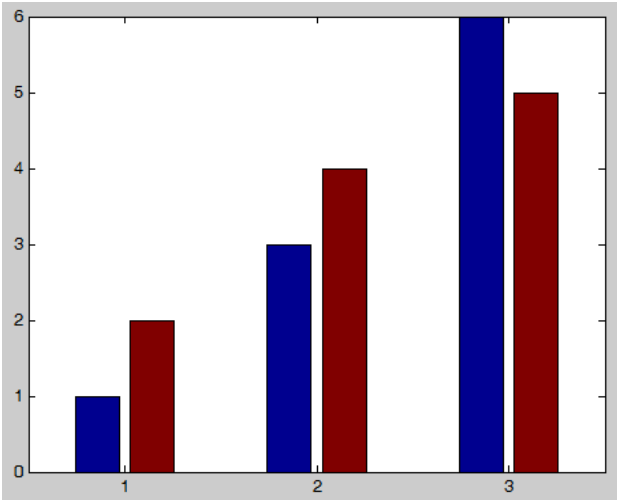
You can also specify the location of each bar by inserting one more argument before.

```
x = [7 8 9];
bar(x,y);
```

This plots the bars of lengths y at locations x, i.e., the 3-unit bar at x=7, 1-unit bar at x = 8, and 5-unit bar at x=9 (like the graph on the right, but [1 2 3] will be changed to [7 8 9], respectively).

Sometimes, we may want to group multiple bars together for comparison (especially human vs model performance under different conditions). To do this, feed in a matrix to `bar()`. It takes each **row** as a group of bars. The number of **columns** determines the number of bars in each group. Suppose you've got [1 3 6] from humans and [2 4 5] from the model, under three different conditions. You can plot your data by doing this:

```
y = [1 2; 3 4; 6 5];
bar(y);
```



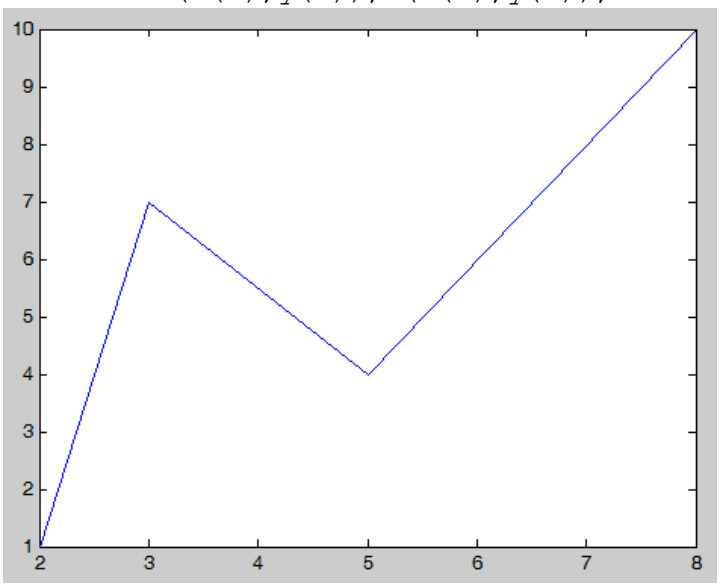
Alternatively, `y` can be defined in the following (with a transpose), which is easier for us (perceptually) to retrieve the data in each condition:

```
y = [1 3 6; 2 4 5]';
```

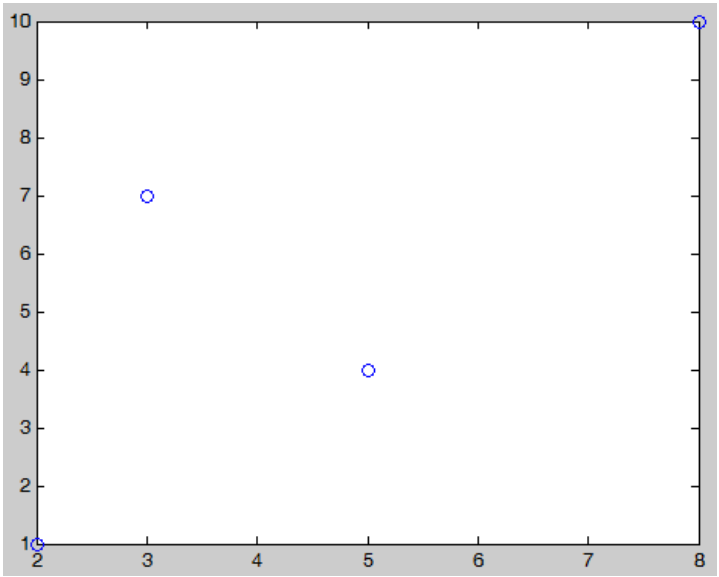
III. Plotting lines / dots

`Plot()` is a very useful function for plotting. Below are some illustrations.

```
y = [1 7 4 10];          % Define y (the values you want to plot as bars)
x = [2 3 5 8];           % Define x (the "independent variable")
plot(x,y);               % Plots lines that link the points (x(1),y(1)),
                         %  (x(2),y(2)), (x(3),y(3)), etc.
```




```
plot(x,y,'o');           % Plots discrete points (x(1),y(1)), (x(2),y(2)),
                         %  (x(3),y(3)), etc., using the circles as
                         %  markers (o is the small letter o for onion)
```

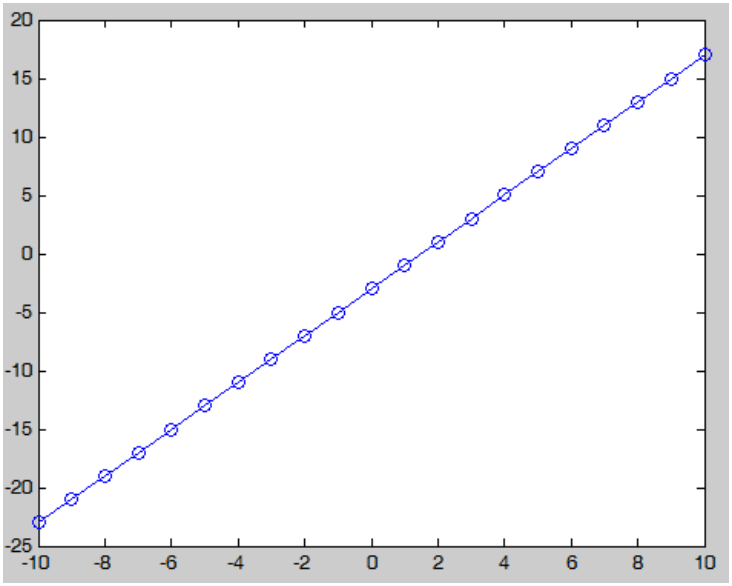



```
plot(x,y,'-o');          % -o draws the line and plots the o at the same
                         %  time; - is the minus sign
```

IV. Plotting techniques

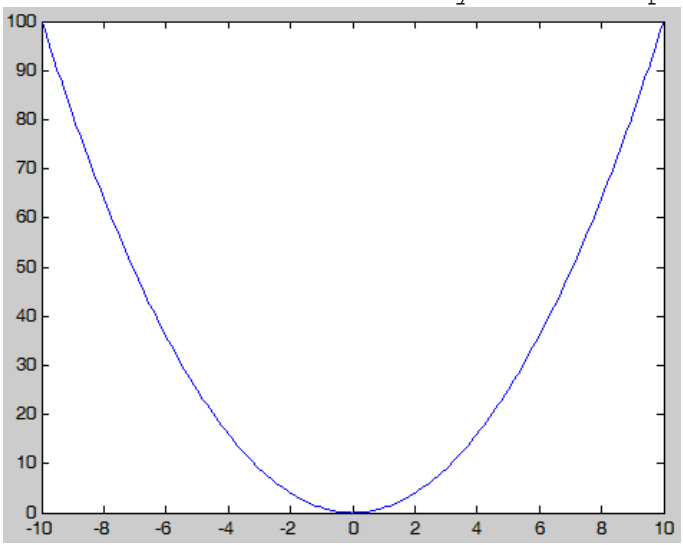
You can use `plot()` to plot more complicate math functions. Suppose you want to plot the line for f(x) = 2x – 3, from x = -10 to x = 10. It's very straightforward if you use `plot()`:

```
x = -10:10;             % Define the range of x
y = 2*x-3;              % Evaluate where the y coordinates are based on x
plot(x,y,'-o');
```



You can plot (visually) smooth curves if your `x` vector has small step size.
Let's try plotting the function $f(x) = x^2$, within x= [-10, 10].

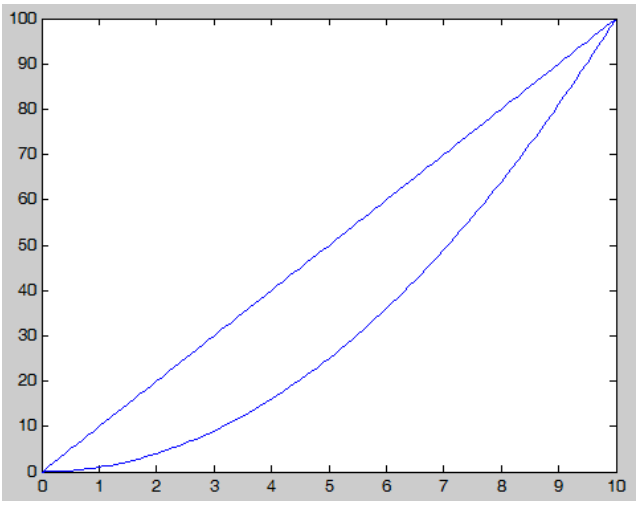
```
x = -10:0.1:10;         % Define the range of x, step size = 0.1
y = x.^2;               % Note the element-by-element operation .^
```



Other useful plotting commands

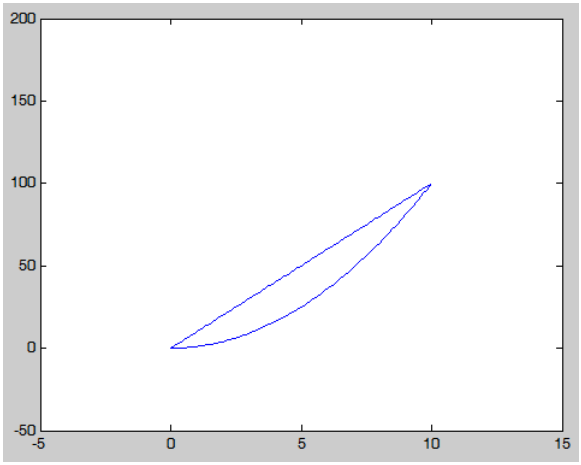
`hold on` activates "same-figure plotting". After this command, graphs will be plotted on the same figure window.

```
x = 0:0.1:10;
xten = x*10;
xsq = x.^2;
plot(x, xten);
hold on;
plot(x, xsq);
```



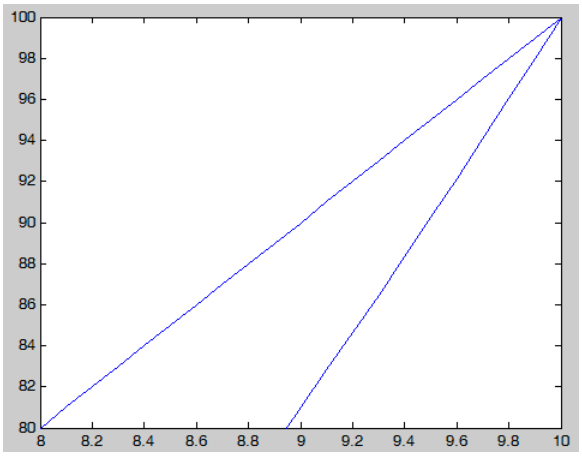
`xlim` and `ylim` set the ranges of the x- and y-axes, respectively. Alternatively, you may also use the `function axis([xmin xmax ymin ymax])`, where the arguments (ordered in a vector) are the limts of the x- and y- axes.

```
% Zoom out
xlim([-5 15]);
ylim([-50 200]);
```




```
% Alternatively
axis([-5 15 -50 200]);
```

```
% Zoom in
xlim([8 10]);
ylim([80 100]);
```




```
% Alternatively
axis([8 10 80 100]);
```

[NOTE]

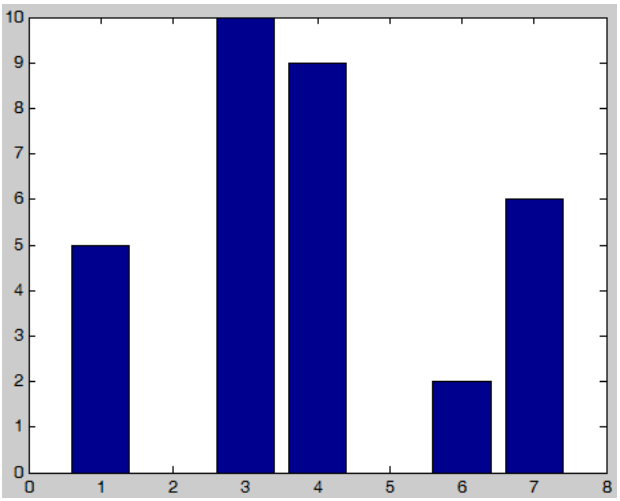
There are many more useful and interesting functions/commands in MATLAB for plotting (e.g., changing line colors, plotting 3D surfaces, etc.). EXPLORE the HELP to find out more!

In-Class Exercise

1. Generate **q1.m** to plot the following figures

   Replicate the following plots.

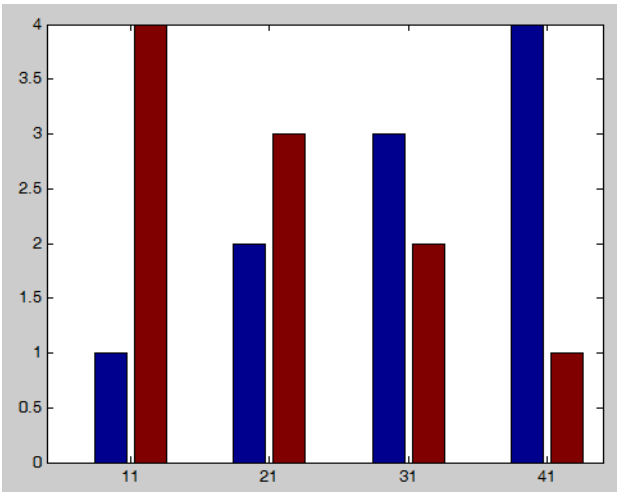
   a) 5 bar, at x locations [3 1 7 6 4], heights [10 5 6 2 9].



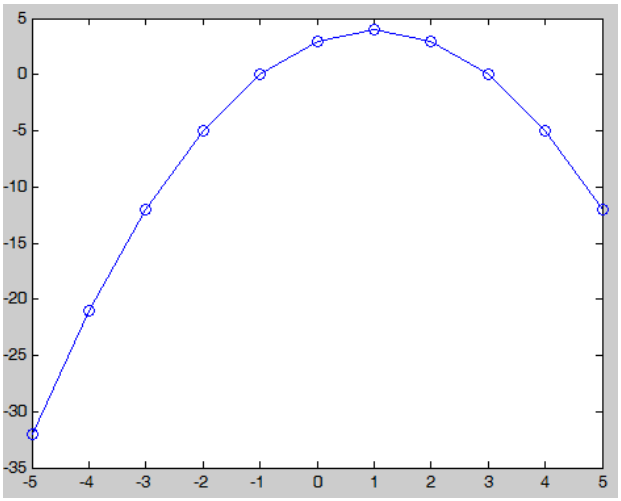
   b)
   Define y as a 2x4 matrix:
   ```
   y = [1 2 3 4; 4 3 2 1];
   ```
   Take the transpose of y (y becomes 4x2):
   ```
   y = y';
   ```

   Define x as a 1x4 vector:
   ```
   x = [11 21 31 41];
   ```

   Plot the 4 pairs of bar using
   ```
   bar(x, y);
   ```



   c) The function $f(x) = -x^2 + 2x + 3$, from -5 to 5, with step size 1.
   Plot the markers ("o") and the line altogether.



2. Generate **q2.m** file for this problem.
   Recall the Body-Mass Index (BMI), a value to reflect how underweight/overweight a person is. Given a person's height ***h*** (in inches) and weight ***w*** (in pounds), the formula for BMI is given by:

$$BMI = \frac{704.5w}{h^2}$$

   a. A person is 70-inch tall. Assuming his height remains constant, compute the BMI values for this person for the weight range between 100 pounds and 200 pounds, with step size of 10 pounds. Plot a line to describe how BMI changes as weight varies within the above range.

   b. The "Normal" range for BMI is said to be from 18.5 to 25. Let's take 23 to be the "ideal point".
      i. Rearrange the formula to express *w* in terms of *h* and BMI.
      ii. Suppose you're working at a health consultant firm. You'd like to advise your client what his/her "ideal weight" should be by referring to a professional graph, so that, if the person tells you his/her height, you have a handy graph to refer to and get a rough idea of how heavy he/she should be.
      With the information given, together with your common sense (e.g., a reasonable range of human heights), plot a graph to serve this purpose.