



Programming with MATLAB

Visualizing data

Dimitris Voudouris, PhD
d.s.voudouris@gmail.com

2D line plots

The most basic plotting function is *plot*

```
plot(95:100)    % this will create an x-y line plot with the values  
                between 95 and 100 along the y-axis and  
                the values 1 to 6 along the x-axis
```

Because we provide only one variable (the vector: 95 96 97 98 99 100), MATLAB places each y-value at the x-value = index position.

In our example, our vector has 6 values, so the x-axis goes from 1 to 6

2D line plots

Assuming that you created the variables as shown in the *onlineScript...*

```
plot(debt_BEL) % will create a line plot with the debt values of Belgium  
               for each of the data points included in this vector
```

Specify options for the line plot:

```
plot(debt_BEL, 'ms- -', 'Linewidth', 4)  
% will plot magenta squares at each data point and connect these  
% squares with a dashed line (- -) that has a thickness of 4 arbitrary units
```

2D line plots

Some useful commands when plotting:

<code>hold on</code>	% will plot new stuff in the same panel
<code>hold off</code>	% if <i>hold on</i> is active, new stuff will be plot in a new figure
<code>figure</code>	% will create a new figure
<code>figure(66)</code>	% will create the new figure 66
<code>xlim([1 5])</code>	% specifies the x-axis values, here from 1 to 5
<code>ylim([76 500])</code>	% specifies the y-axis values, here from 76 to 500
<code>xlabel('demo')</code>	% adds a label along the x-axis, with the text 'demo'
<code>ylabel('demo')</code>	% same for the y-axis
<code>legend('demo')</code>	% adds a legend about color/symbols of the plotted line

Please do look at the details of these functions and make a search (*help*) to figure out what the exercises ask you to do

Histograms

Just use the command *histogram*

```
histogram([1 1 1 1 4 4 5 7 2 2 2])
```

 % will show how many times each number in this vector appears

```
histogram(randi([90 100], 4, 3))
```

% will show how many times each of these random integers
% between 90 and 100 appear in the generated 4 x 3 matrix

```
histogram(debt_BEL, 'Edgecolor', 'k', 'Facecolor', 'g', 'Linewidth', 5)
```

% will show how many times each of the debt values within the
% variable '*debt_BEL*' appears, and will set the resulting bars in
% green color, with thick black outlines

help histogram to see other properties, like setting the number of bins...

Bars

Just use the command *bar*

```
bar([1 1 1 1 4 4 5 7 2 2 2])
```

 % will plot the values of this vector along the x-axis. **Notice the difference with histogram!**

```
bar(randi([90 100], 4, 3))
```

% will plot the values of this matrix in a way that values of row 1 will be
% grouped in x-axis = 1, values of row 2 in x-axis = 2, and so on...

```
bar(debt_GER, 'Facecolor', 'k', 'Edgecolor', 'm', 'Linewidth', 1)
```

% will plot the German debt in black bars with magenta outlines
% and thickness of 1

```
bar([debt_GER', debt_GRE'])
```

% will plot the time-course of German & Greek debts next to each other

Tips

One good way to learn this stuff is to try out the commands.
Reading this and the associated m-file is not enough.

Perform the commands, change the inputs, misbehave with the brackets.
Do you understand why you get the outcome you get? If yes, great.
If not, you should ask me!

Explore for yourself related functions, such as:

help plot

help plot3

help bar3

...

Good luck!