Plotting(1) - Line Graphs

First, let's cover the basics of inputting X and Y values and their outputs when creating graphs.

A. X and Y are both vectors :

They must have equal lengths.

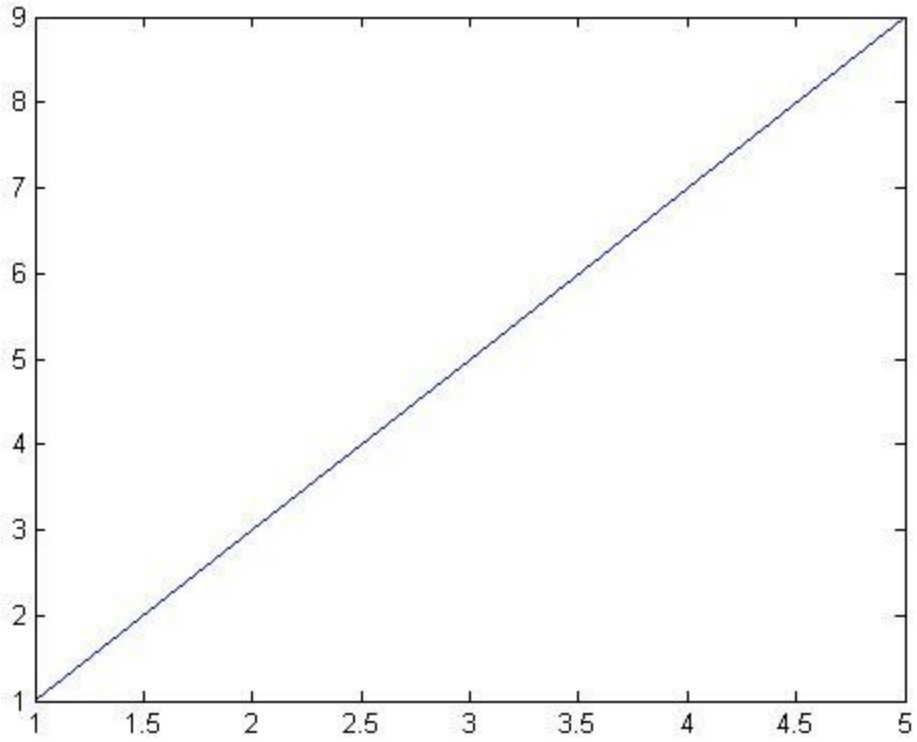
The plot will be X versus Y.

For example:

input:

```
>> plot(1:5, 1:2:9)
```

output:

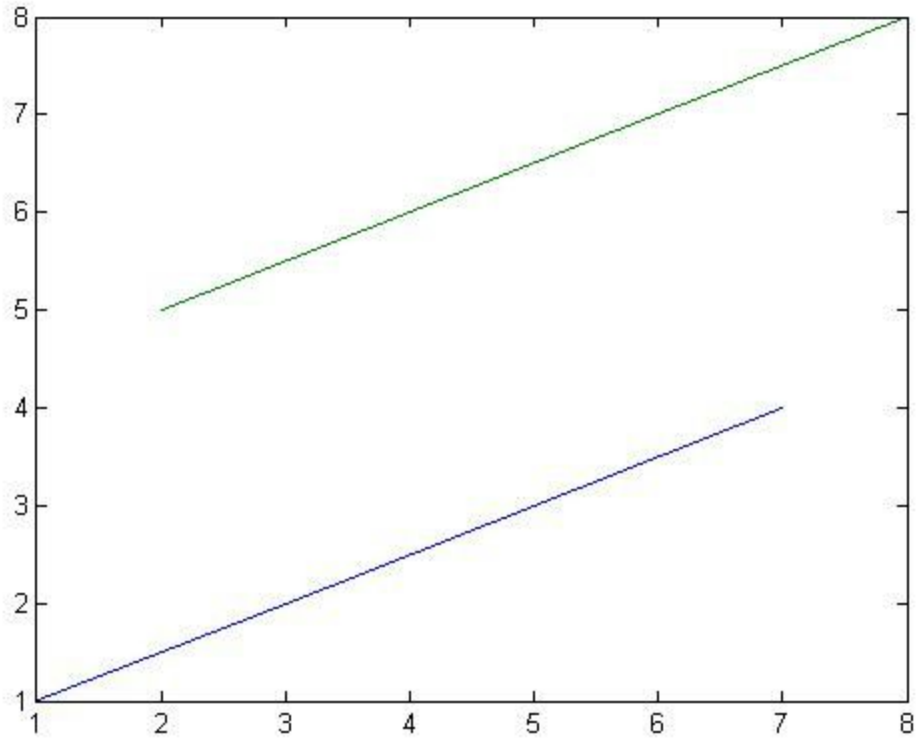


B. X and Y are both matrices:
They must have equal sizes.
The plot will be columns of Y versus columns of X.
For example:
input:

```
>> d = [1,5;2,6;3,7;4,8]
>> x = [1, 2;3,4;5,6;7,8]
>> plot(x, d)
```

output:

```
d =
   1   5
   2   6
   3   7
   4   8
x =
   1   2
   3   4
   5   6
   7   8
```



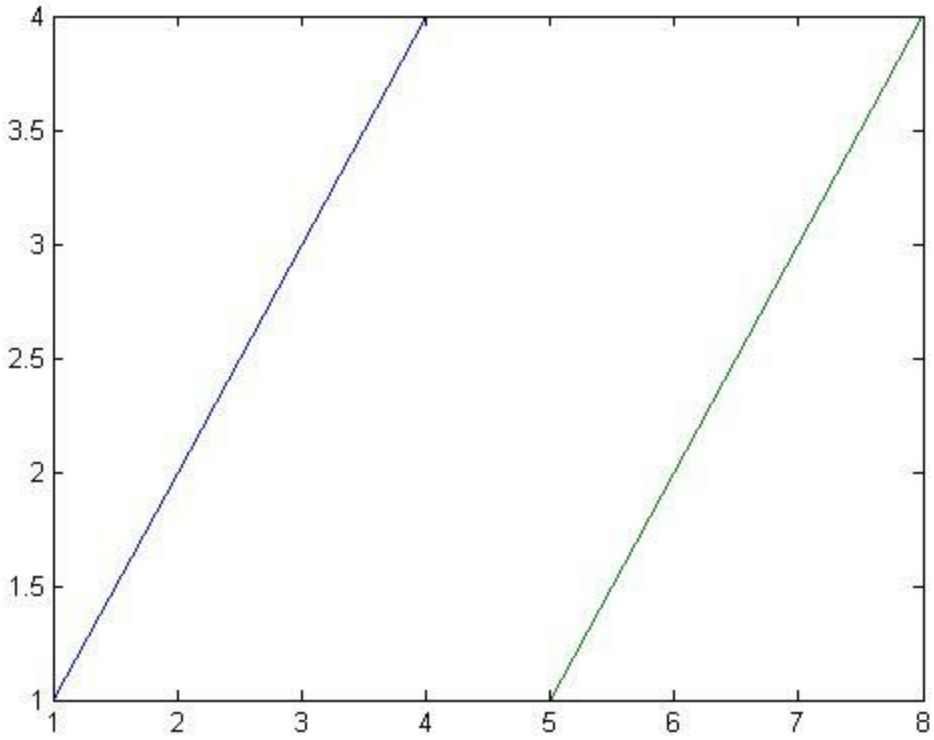
Since there are two colons in both X and Y, there are two lines.
This is equivalent to graphing d(:,1) vs x(:,1) and d(:, 2) vs x(:, 2) on the same graph.

C. X and Y are a vector and a matrix (order does not matter)
One dimension of the matrix must match the length of the vector.
For example:
input:

```
>> d = [1,5;2,6;3,7;4,8]
>> plot(d, 1:4)
```

output:

```
d =
   1   5
   2   6
   3   7
   4   8
```



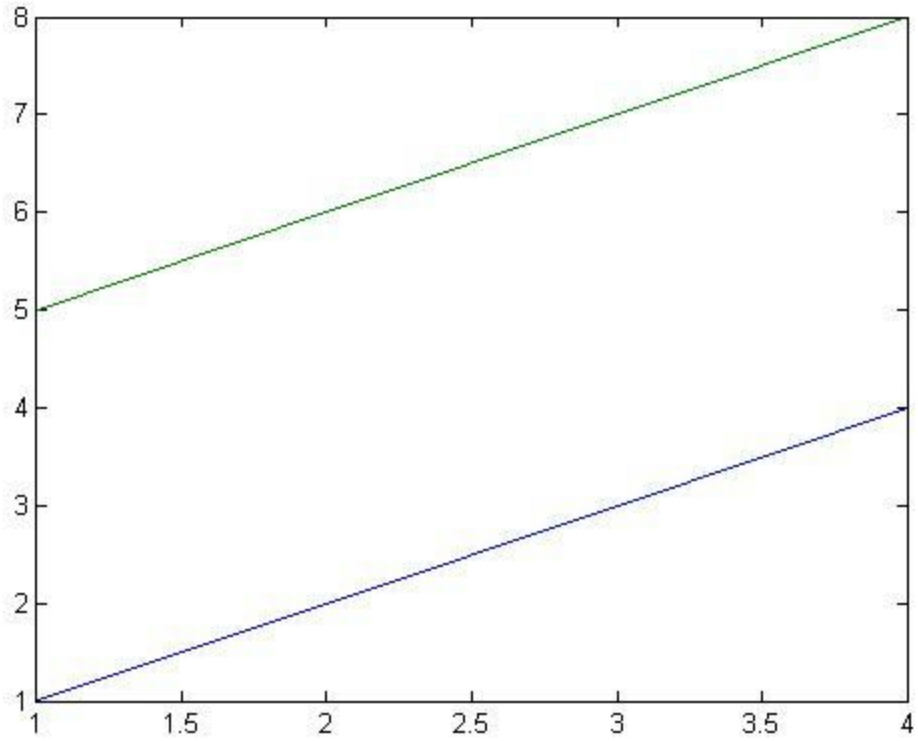
Since the number of rows in the matrix is the same as the length of the vector, the graph is column vs vector.
input:

```
>> d = [1,2,3,4;5,6,7,8]
>> plot(d, 1:4)
```

output:

```
d =
   1   2   3   4
```

5 6 7 8



With d still being the x axis, the dimensions of d are changed to 2 by 4. This time, the number of columns in d is the same as the length of the vector, so this graph is rows vs vector.

D. X and Y are a scalar and a scalar/vector

The graph will only be points.

This will not be visible without inputting other arguments into the plot() parameter.

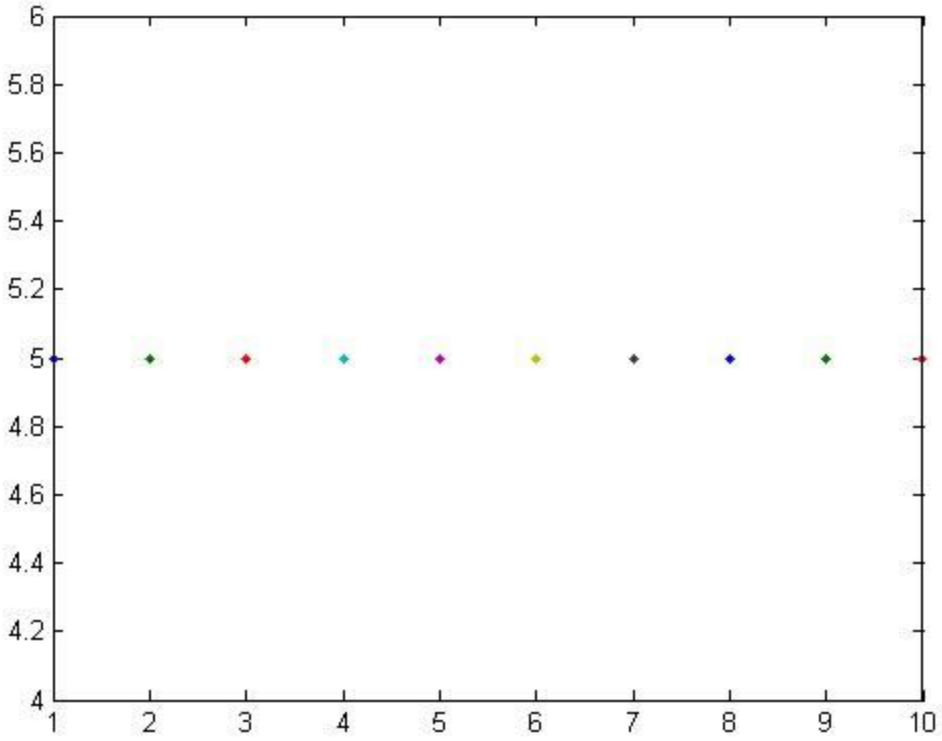
For examples:

input:

```
>> x = 1:10
>> plot(x, 5, '.')
```

output:

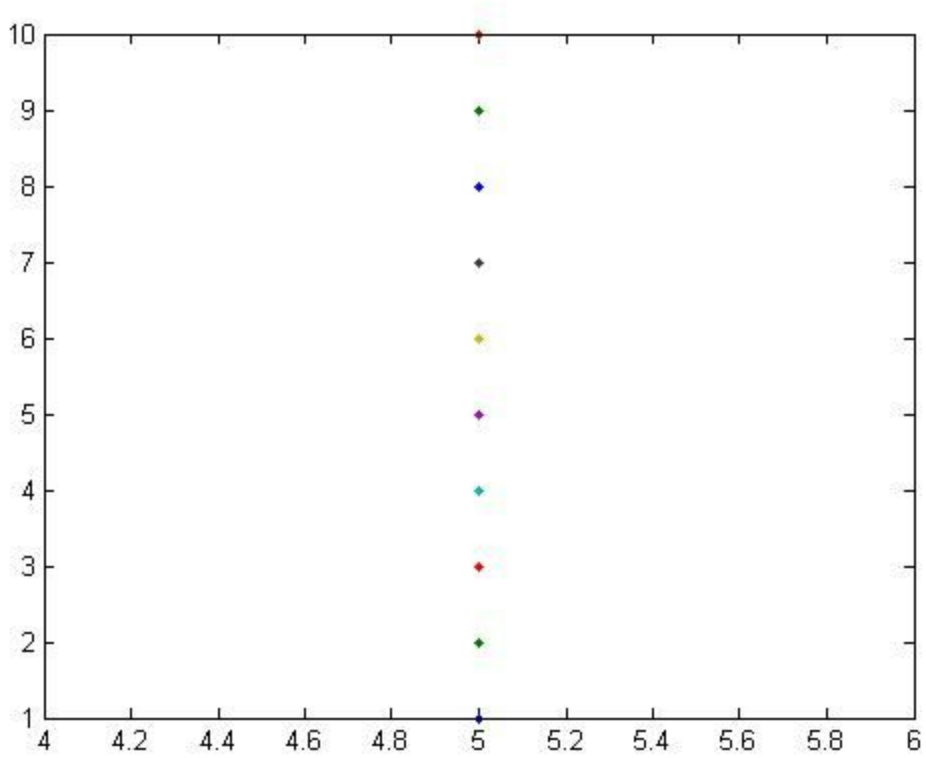
```
x =
     1     2     3     4     5     6     7     8     9    10
```



input:

```
>> x = 1:10
>> plot(5, x, '.')
```

output: x = 1 2 3 4 5 6 7 8 9 10



These are much of the possible combinations of X and Y. Now we can talk about implementing other arguments.

There are a multitude of different arguments to use:

A. plot(x, y)

A graph that has 1 line with the default style.

B. plot(x, y, LineSpec)

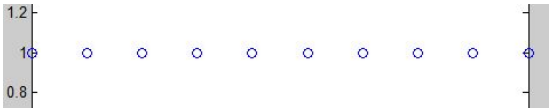
A graph that has 1 line with a customized style.

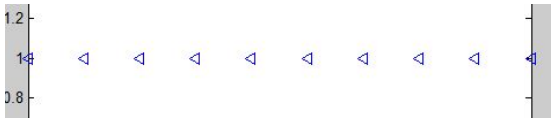
There are a few things that can be customized: Line style, Marker and Color.

The input format should be '(lineStyle)(Marker)(Color)'. Here are the charts that contain all the possible inputs and their effects:

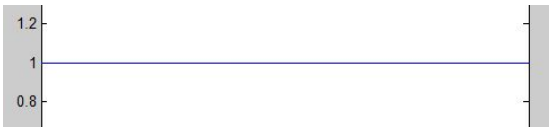
Line Style	Effect
-	
--	

-.	
:	
none	No line

Marker	Effect
o	
+	
*	
.	
x	
s	
d	
^	

v	
>	
<	
p	
h	
none	No marker

Color	Effect
y	
m	
c	
r	

g	
b	
w	
k	

Besides these, there are more options called Name-Value pair arguments. They can be placed anywhere, as long as the line they are customizing is in front of them. The syntax is ‘(Setting)’,(Value). Some common ones are listed below:

Setting	Value restrictions
‘LineWidth’	default is 0.5; must be a positive value
‘MarkerSize’	default is 6; also affects the width of the marker edge; must be a positive value
‘MarkerEdgeColor’	must be a color value (see the chart above)
‘MarkerFaceColor’	must be a color value (see the chart above)

For example:

plot(x, y, ‘--og’, ‘LineWidth’, 4, ‘MarkerEdgeColor’, ‘g’, x2, y2, ‘>’, ‘MarkerFaceColor’, ‘b’)

For more Name-Value Pair Arguments, visit :

http://www.mathworks.com/help/matlab/ref/chartline-properties.html

C. plot(X1, Y1, X2, Y2, …. , Xn, Yn)

Plots n lines on the same graph with default line style and marker but different colors.

D. plot(X1, Y1, LineSpec1, X2, Y2, LineSpec2 …. , Xn, Yn, LineSpecn)

Plots n lines on the same graph with customized style for each line.

If you wish to graph multiples lines on the same graph without doing everything at once, type in hold on to “hold on” to the current graph. When you are finished, type hold off to “let go” of the graph.