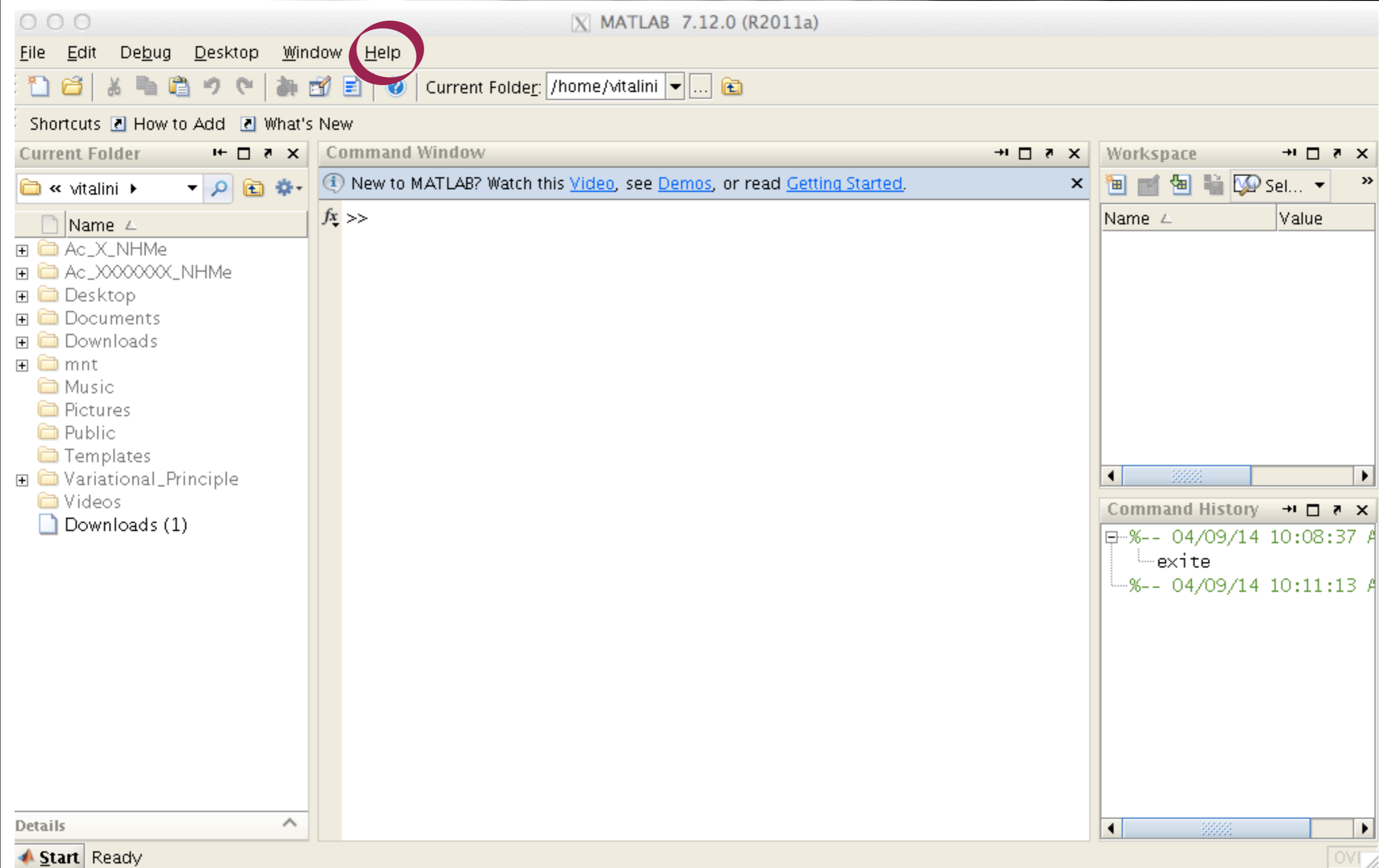


## Introduction to Matlab

- MATLAB is a high-level language and interactive environment for numerical computation, visualization, and programming.
- Enables to analyze data, develop algorithms, and create models and applications.
- Based on C/C++ , Java
- The desktop includes these panels:
  - Current Folder**— Access your files.
  - Command Window**— Enter commands at the command line, indicated by the prompt (>>).
  - Workspace**— Explore data that you create or import from files.



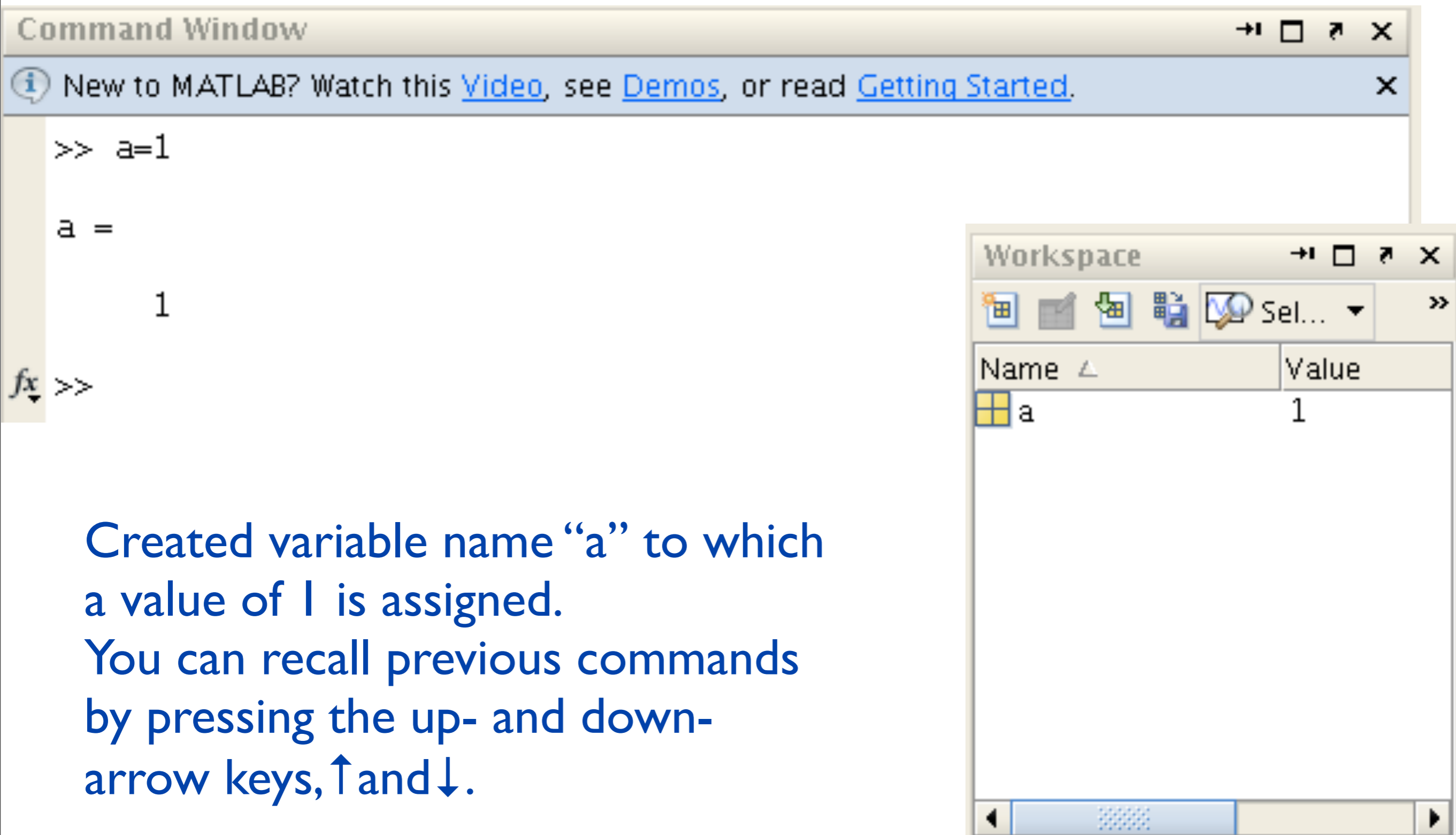
The screenshot displays the MATLAB 7.12.0 (R2011a) environment. The 'Help' menu item is highlighted with a red circle. The Command Window contains the following text:

```
fx >>
```

The Command History window shows the following entries:

```
%-- 04/09/14 10:08:37 A  
└─ exite  
%-- 04/09/14 10:11:13 A
```

## Create Variables

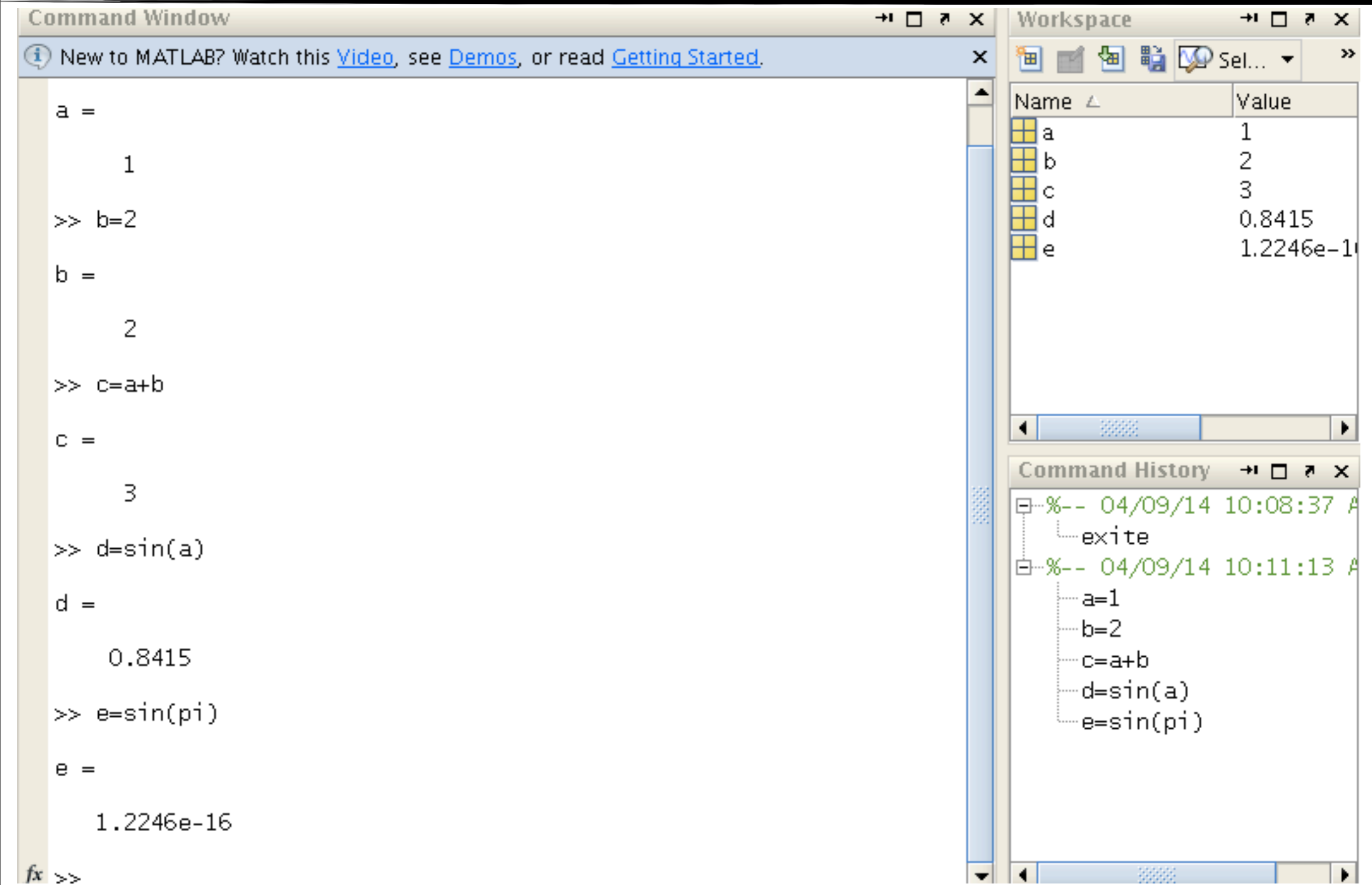


The screenshot displays the MATLAB Command Window and Workspace. The Command Window shows the command `>> a=1` and the output `a = 1`. The Workspace window shows a table with two columns: Name and Value. The table contains one row with the variable name 'a' and the value '1'.

```
>> a=1  
  
a =  
  
    1  
  
fx >>
```

Name	Value
a	1

Created variable name “a” to which a value of 1 is assigned.  
You can recall previous commands by pressing the up- and down-arrow keys, ↑ and ↓.



Command Window

New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).

```
a =  
    1  
>> b=2  
  
b =  
    2  
>> c=a+b  
  
c =  
    3  
>> d=sin(a)  
  
d =  
    0.8415  
>> e=sin(pi)  
  
e =  
    1.2246e-16  
fx >>
```


Workspace

Name	Value
a	1
b	2
c	3
d	0.8415
e	1.2246e-16

Command History

```
%-- 04/09/14 10:08:37 A  
  |-- exite  
%-- 04/09/14 10:11:13 A  
  |-- a=1  
  |-- b=2  
  |-- c=a+b  
  |-- d=sin(a)  
  |-- e=sin(pi)
```

# Strings

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).

```
>> mystring='Hallo World!'
```

```
mystring =
```

```
Hallo World!
```

```
>> mystring1='Hi Everyone!'
```

```
mystring1 =
```

```
Hi Everyone!
```

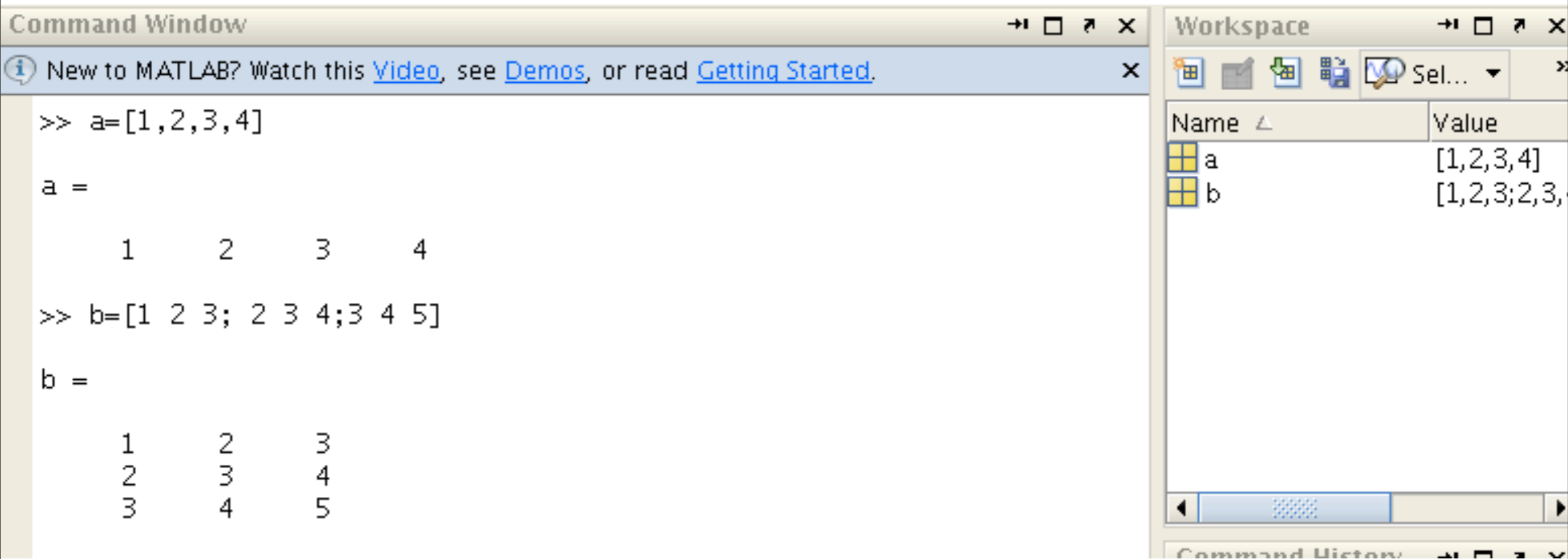
```
>> cat_string=strcat(mystring, ' ', mystring1)
```

```
cat_string =
```

```
Hallo World!Hi Everyone!
```

```
 >>
```

# Matrices and Arrays



The screenshot shows the MATLAB interface with two windows: Command Window and Workspace.

**Command Window:**

```
>> a=[1,2,3,4]

a =

     1     2     3     4

>> b=[1 2 3; 2 3 4;3 4 5]

b =

     1     2     3
     2     3     4
     3     4     5
```

**Workspace:**

Name	Value
a	[1,2,3,4]
b	[1,2,3;2,3,4;3,4,5]

The Command Window also displays a message: "New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#)."

# Accessing Matrix/Array Elements

```
A =  
    16     2     3    13  
     5    11    10     8  
     9     7     6    12  
     4    14    15     1  
  
>> A(:,1)  
  
ans =  
  
    16  
     5  
     9  
     4  
  
>> A(1,:)   
  
ans =  
  
    16     2     3    13  
  
>> A(2,3)  
  
ans =  
  
    10  
  
>>  
  
>> A(1:4,2)  
  
ans =  
  
     2  
    11  
     7  
    14  
  
>> A(2,3:4)  
  
ans =  
  
    10     8  
  
>> A(3:4,1:3)  
  
ans =  
  
     9     7     6  
     4    14    15  
  
fx >>
```




# Matrices and Arrays

```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Gettin
>> c=zeros(5,1)
c =
    0
    0
    0
    0
    0
>> d=zeros(2,5)
d =
    0    0    0    0    0
    0    0    0    0    0
fx >>
```

Initialize arrays or matrices with all null elements.

The first index means rows, the second columns.

# Operations with Matrices and Arrays

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).

```
>> D10=d+10
```

```
D10 =
```

```
    10    10    10    10    10
    10    10    10    10    10
```

```
>> D10(1,1)=D10(1,1)+5
```

```
D10 =
```

```
    15    10    10    10    10
    10    10    10    10    10
```

```
>> E=D10/2
```

```
E =
```

```
    7.5000    5.0000    5.0000    5.0000    5.0000
    5.0000    5.0000    5.0000    5.0000    5.0000
```

```
fx >>
```

# Operations with Matrices and Arrays

## Command Window

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#)

```
>> c=[1 2 3; 4 5 6; 7 8 9]
```

```
c =
```

```
    1    2    3
    4    5    6
    7    8    9
```

```
>> c'
```

```
ans =
```

```
    1    4    7
    2    5    8
    3    6    9
```


```
fx >>
```

# Operations with Matrices and Arrays

$e=c*d$  -- Matrices Multiplication

$E=c.*d$  -- Element wise Multiplication

Command Window

 New to MATLAB? Watch this [Video](#), see

```
>> d=[1 1 1; 2 4 6; 5 5 10]
```

```
d =
```

```
    1    1    1
    2    4    6
    5    5   10
```

```
>> c
```

```
c =
```

```
    1    2    3
    4    5    6
    7    8    9
```

```
>> e=c*d
```

```
e =
```

```
    20    24    43
    44    54    94
    68    84   145
```

```
>> E=c.*d
```

```
E =
```

```
    1    2    3
    8   20   36
   35   40   90
```

## Concatenation

```
>> F=[c,d]
```

```
F =
```

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

## Save Variable

The screenshot displays the MATLAB interface with the following components:

- File Explorer:** Shows the current directory with files like 'Ac\_X\_NHMe', 'Ac\_XXXXXXXX\_NHMe', and 'F.mat' (circled in red).
- Variable Editor - F:** Shows a 3x6 double matrix F with the following values:

	1	2	3	4	5	6	7	8
1	1	2	3	1	1	1		
2	4	5	6	2	4	6		
3	7	8	9	5	5	10		
- Workspace:** Lists variables A, D10, E, F (circled in red), ans, b, c, d, and e.
- Command Window:** Shows the commands:

```
>> save F.mat  
>> load F.mat
```

# For Loop

 New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#).


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

```
>> for i=1:length(F(:,1))  
% looping over all rows of matrix F  
max(F(i,:))  
% printing maximum of each row  
end
```

```
ans =  
  
     3
```

```
ans =  
  
     6
```

```
ans =  
  
    10
```

```
 >>
```

Function that checks length of array.

Function that evaluates maximum of an array.

## While Loop

```

>> i=1
i =
     1

>> F
F =
     1     2     3     1     1     1
     4     5     6     2     4     6
     7     8     9     5     5    10

>> while i<=length(F(:,1))
%loop over rows of F
min(F(i,:))
%minimum of each row
i=i+1;
%increment iterator, the ";" avoids the printout
end

```

```

ans =
     1

ans =
     2

ans =
     5

```

Function that evaluates minimum of an array.

## if statement

```
>> F

F =

     1     2     3     1     1     1
     4     5     6     2     4     6
     7     8     9     5     5    10

>> for i=1:length(F(:,1))
%loop over rows of F
if sum(F(i,:)) > 10
%if the sum of the elements of the ith row
%is bigger than 10
sum(F(i,:))
%print to screen sum of ith row
end
end

ans =

    27

ans =

    44

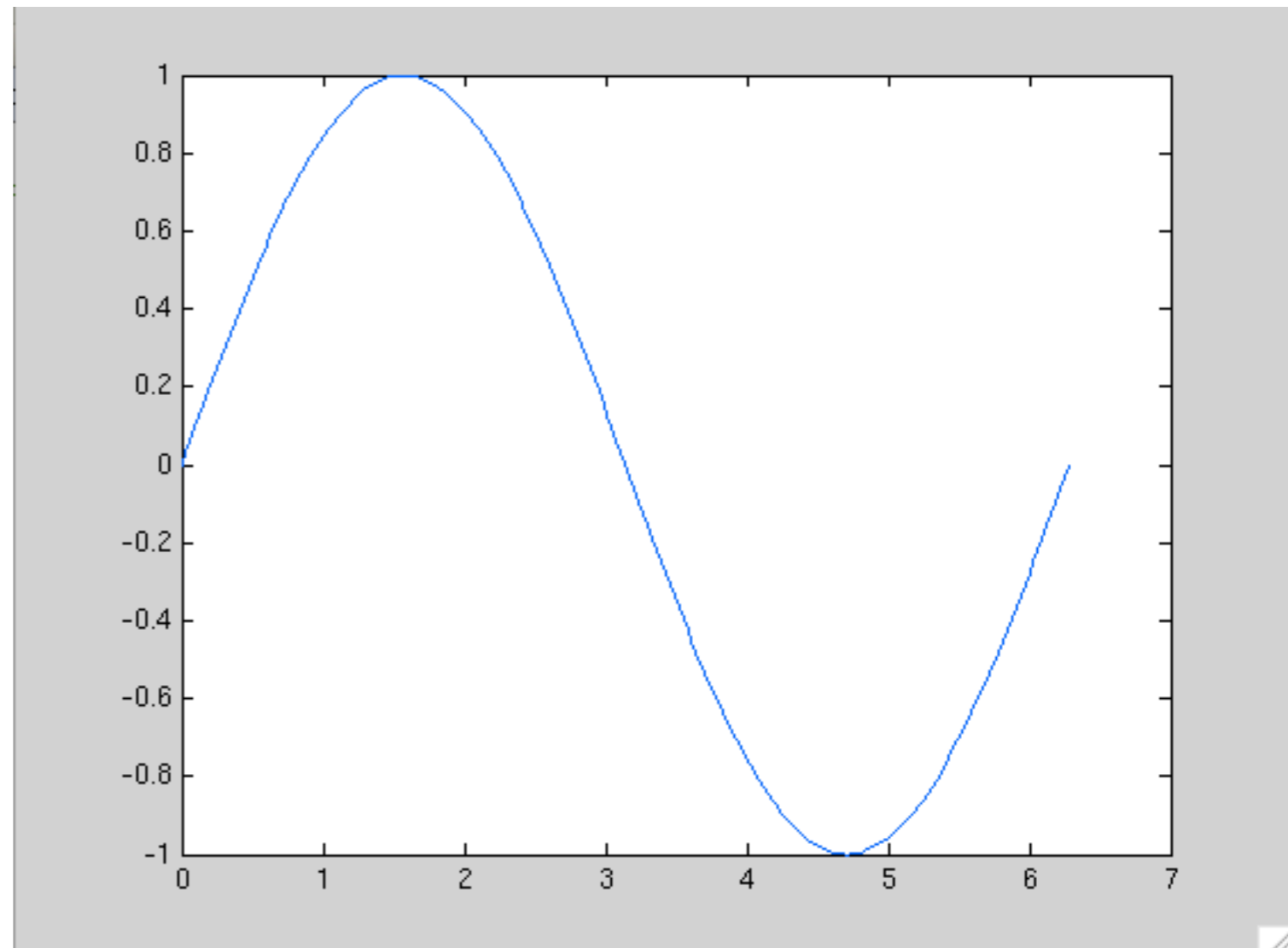
>>
```

Function that evaluates sum of elements of an array.



## Plots: Line plots

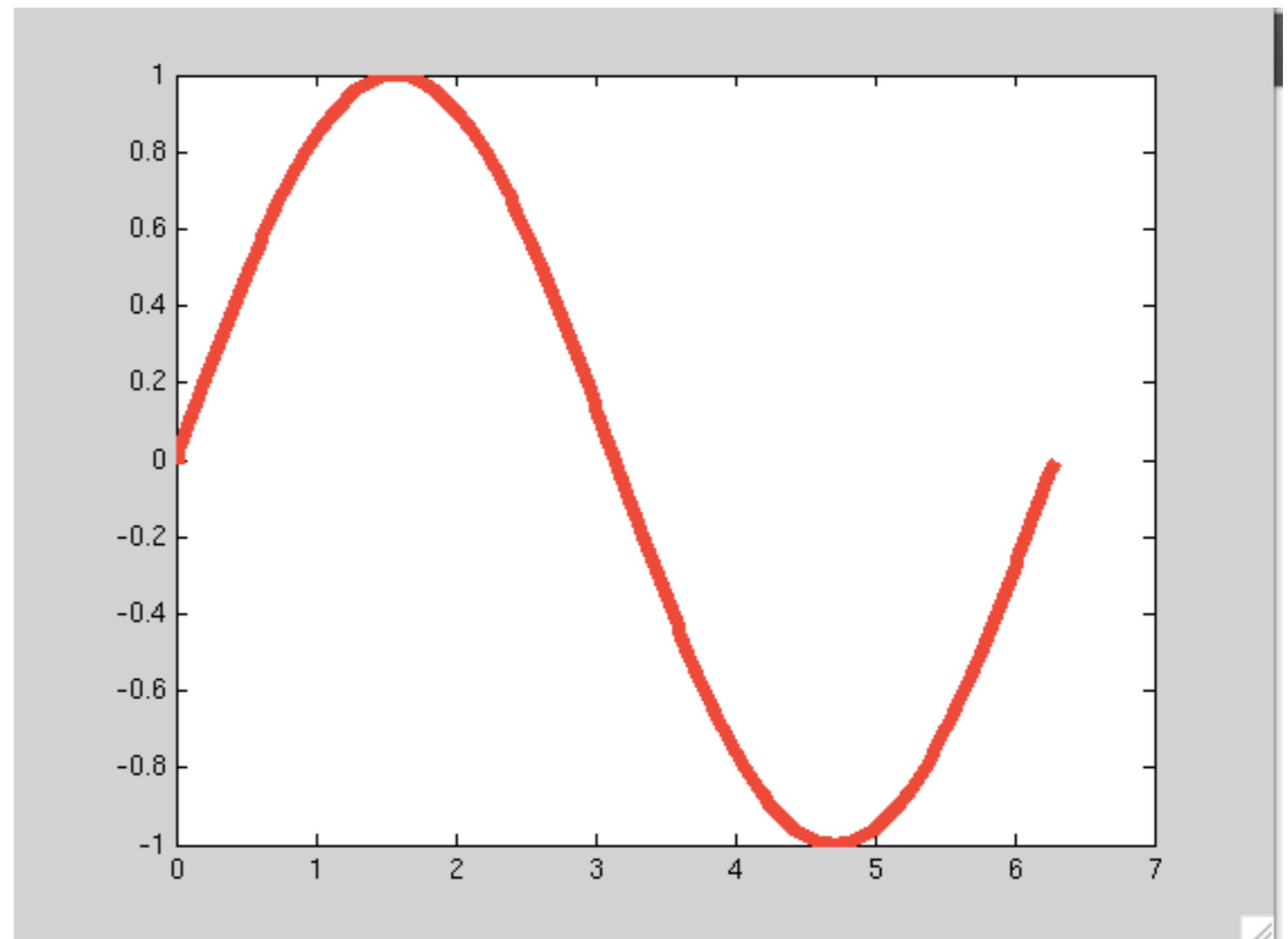
```
>> x=0:pi/100:2*pi;  
>> % x is all the elements between 0 and 2*pi  
>> %taken one every 100  
>> y=sin(x);  
>> % y is an array whose elements are  
>> % the sin of the elements of x  
>> plot (x,y)  
>>
```



## Plots: Line plots

```
>> plot(x,y, '-r', 'Linewidth', 5)  
>>
```

To change color and width of the line

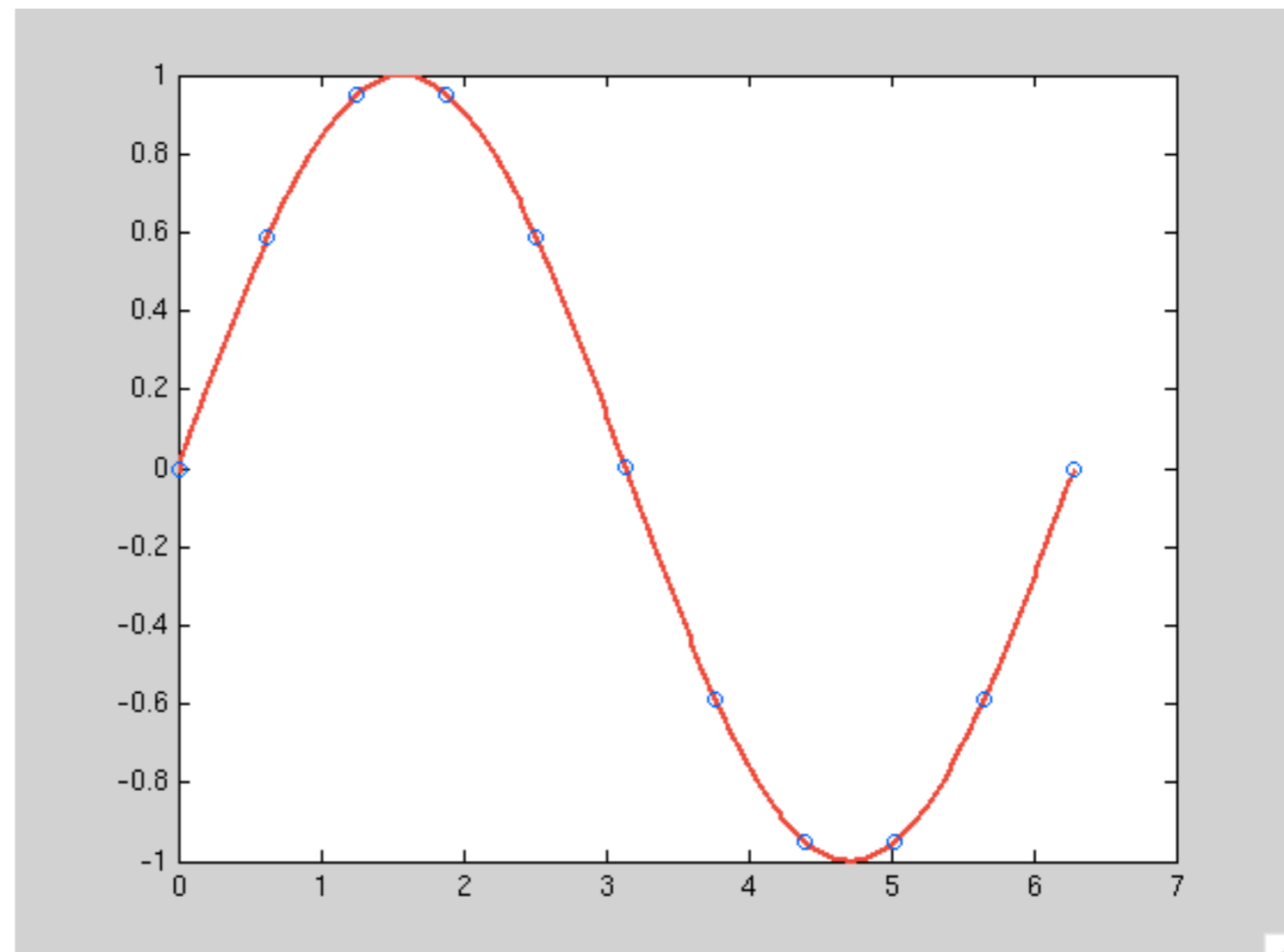


## Plots: Line plots

```
>> plot (x,y, '-r', 'Linewidth', 2)
>> hold on
>> plot (x(1:20:length(x)),y(1:20:length(y)), 'ob', 'MarkerSize', 5)
''
```

Superimpose 2 lines:  
- red solid line  
- dot line only selecting  
one element every 20.

x and y arrays must have  
same length

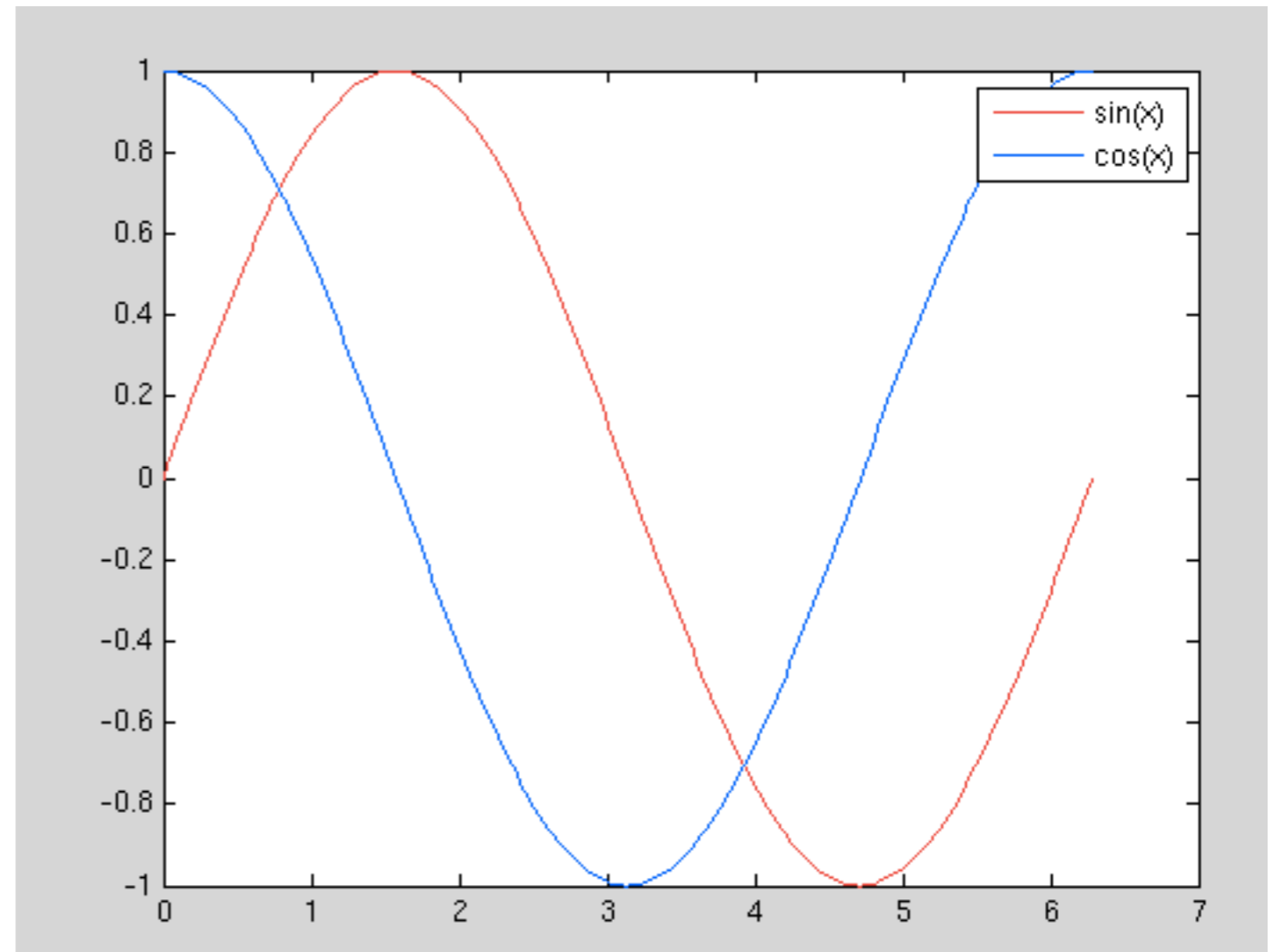


## Plots: Line plots

```
>> z=cos(x);  
>> plot(x,y,'r')  
>> hold on  
>> plot(x,z,'b')  
>> legend('sin(x)', 'cos(x)')
```

plot 2 lines in the same figure

add legend



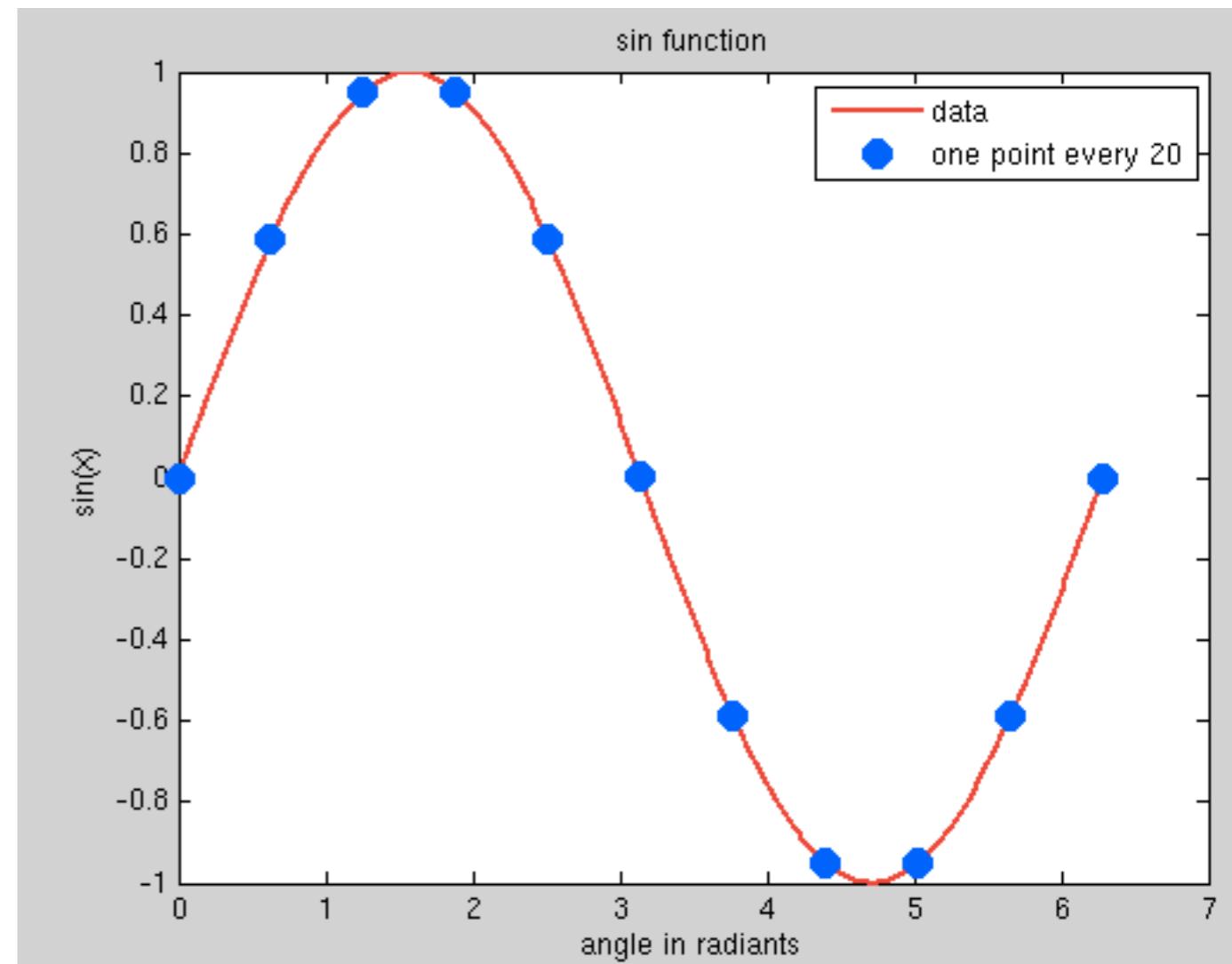
## Plots: Line plots

```
>> plot (x,y, '-r', 'Linewidth', 2)
>> hold on
>> plot (x(1:20:length(x)),y(1:20:length(y)), 'ob', 'MarkerSize', 5)
>> plot (x,y, '-r', 'Linewidth', 2)
>> hold on
>> plot (x(1:20:length(x)),y(1:20:length(y)), 'ob', ...
'MarkerSize', 10, 'MarkerFaceColor','b')
>> legend('data','one point every 20')
>> title('sin function')
>> xlabel('angle in radians')
>> ylabel('sin(x)')
>>
```

Title

axis labels

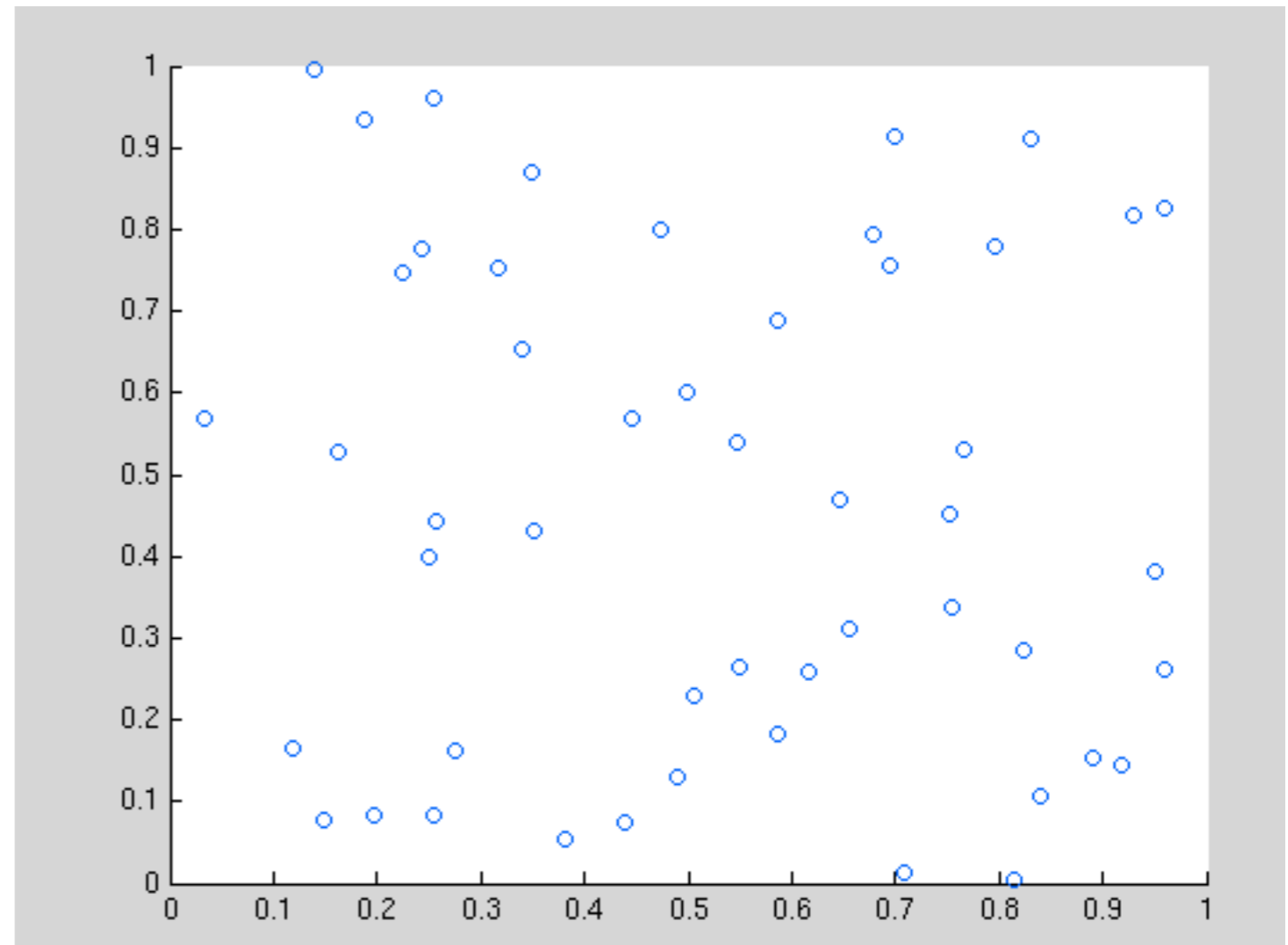
... means continuation



## Plots: Scatter plots

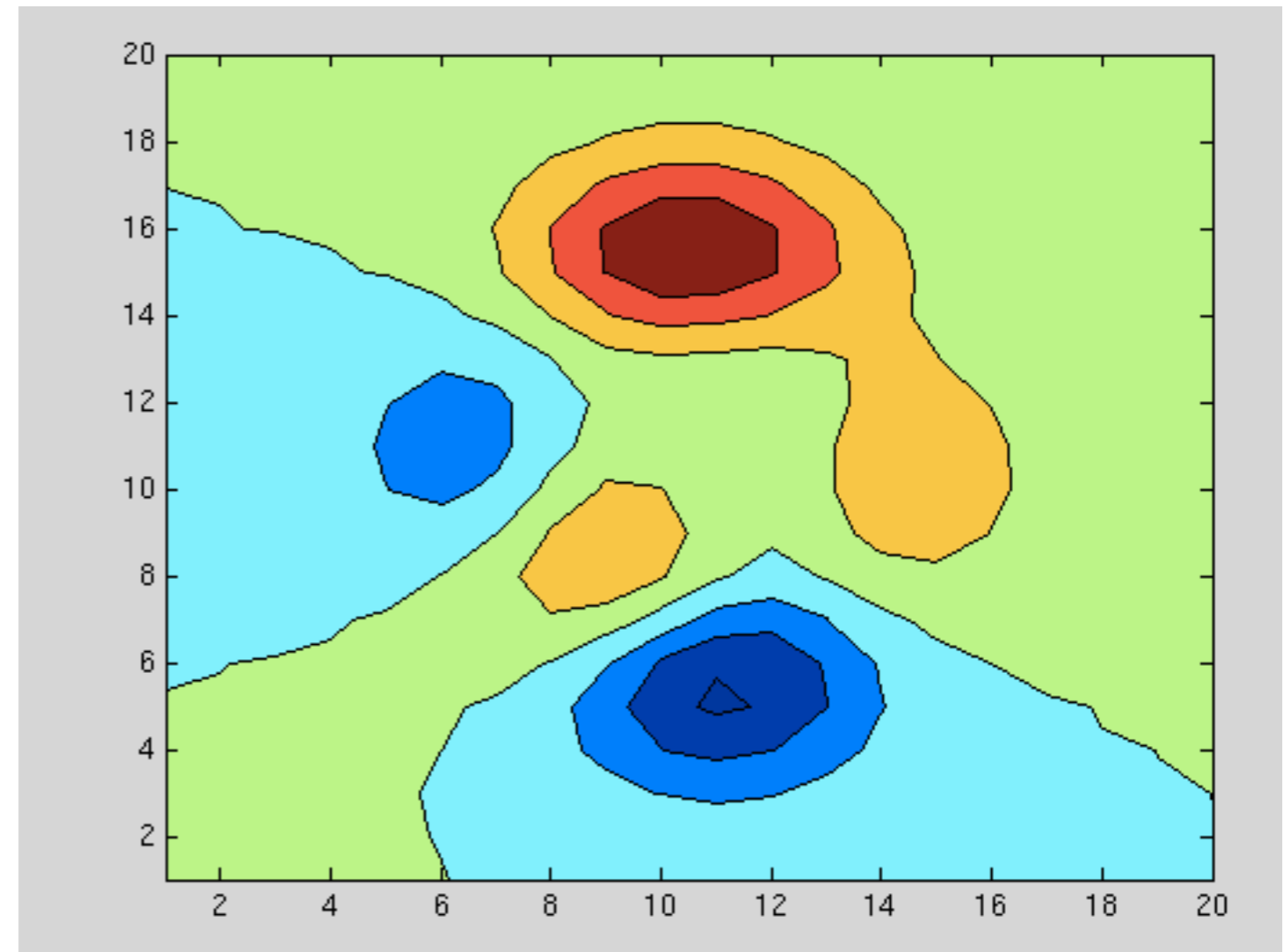
```
>> x=rand(50,1);  
>> y=rand(50,1);  
>> scatter(x,y)
```

rand is a function that selects random numbers (between 0 and 1)



## Plots: 2D plots

```
Z=peaks(20);  
contourf(Z)
```



## Script

The simplest type of MATLAB program is called a script.

A script is a file with a .m extension that contains multiple sequential lines of MATLAB commands and function calls.

You can run a script by typing its name in the command line, by pressing f5 or clicking on the run button.

Go to File -> New script to open the script editor.

Once the script is ready you can save it for later use.

[http://www.mathworks.de/help/pdf\\_doc/matlab/getstart.pdf](http://www.mathworks.de/help/pdf_doc/matlab/getstart.pdf)