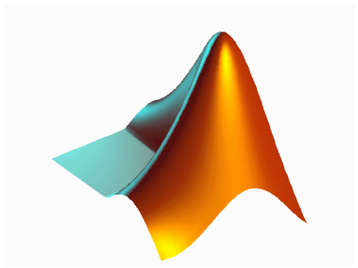


Introduction to *Matlab*

MUS4831, Olivier Lartillot, 12.10.2017

What is *Matlab*?

- A **data-analysis program**:
  - compute, visualize, and analyze your data,
  - with simple command-line commands.
- A **programming language**:
  - high-performance technical computing
  - easier than low-level languages (C, Java, ...)
- A large array of ***toolboxes***: add-on application-specific solutions:
  - signal processing, stats, neural networks, ...
  - *MIDItoolbox*, *MIRtoolbox*, *MoCap Toolbox*.

Basic Data Structures

- Scalars:

```
s = 3.1415
```

3.1415

- Vectors:

```
v = [1, 5, 7]
```

1 5 7

```
w = [1; 5]
```

1
5

→ w′ 1 5

- Matrices:

```
m = [1, 2; 3, 4]
```

1 2
3 4

- Nothing: `[]`

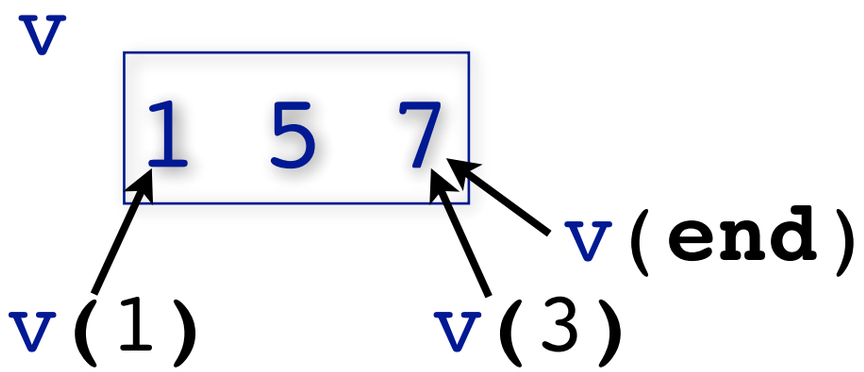
```
[v; v]
```

1 5 7
1 5 7

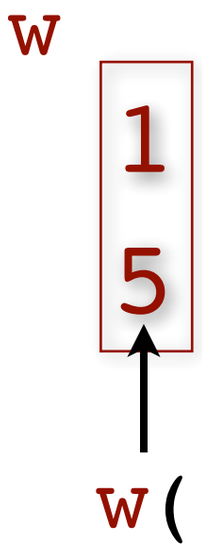
```
[w, m]
```

1 1 2
5 3 4

Accessing elements

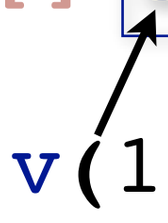



```
length(v) = 3
size(v) = [1 3]
```



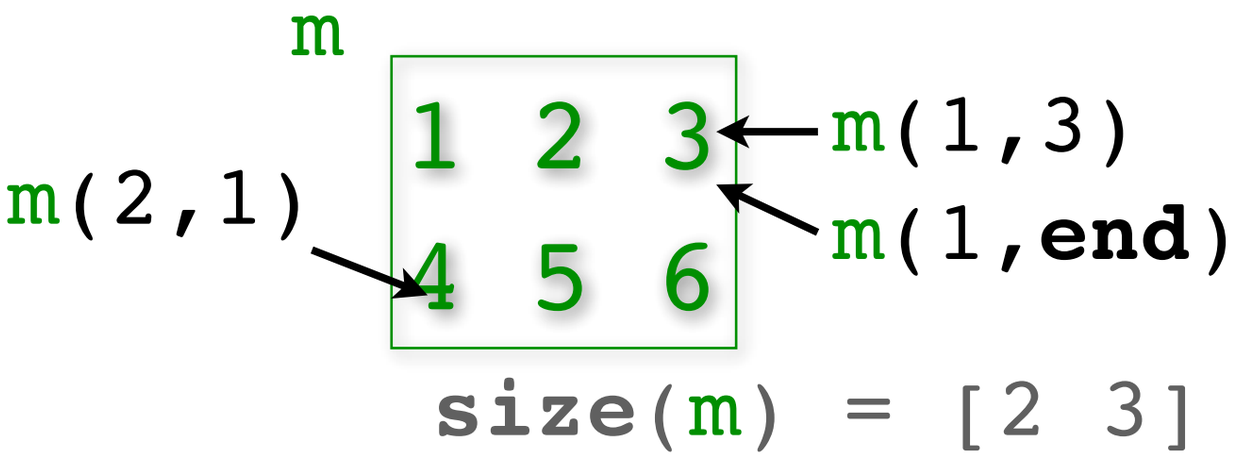

```
length(w) = 2
size(w) = [2 1]
```

```
v(3)=8      1 5 8

v(1)=[]     [] 5 8
```




```
size(v) = [1 2]
```

Accessing elements



m(2,:) → 4 5 6

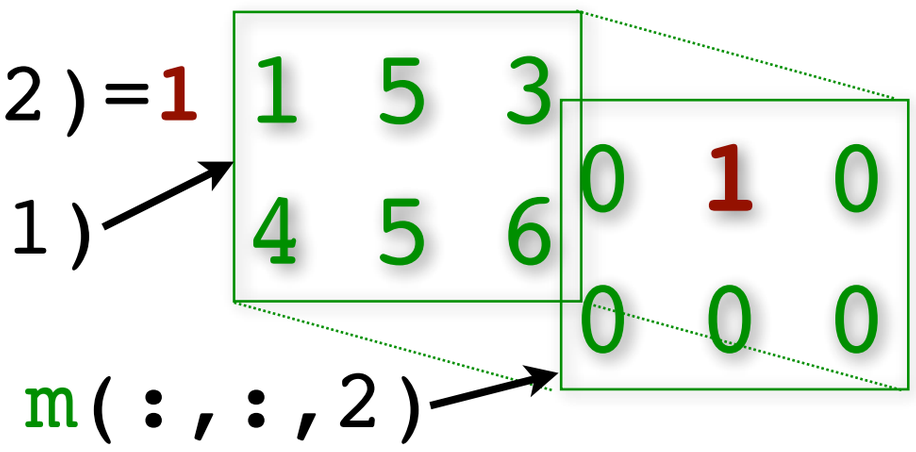
m(:,1) →
1
4

m(1,2)=5 →
1 5 3
4 5 6

More dimensions!

size(m) = [2 3 **2**]

m(1,2,2)=1



Colon operator

```
u = 1:5
```

1	2	3	4	5

```
v = 1:2:9
```

1	3	5	7	9

```
m = [v;v+1;v*2]
```

1	3	5	7	9	1
2	4	**6**	**8**	**10**	2
2	6	**10**	**14**	**18**	3
1	2	3	4	5	

```
m(2:3,3:5)
```

Element-By-Element Operations vs. Matrix Operations

```
[1 2 3]-4 ..... [-3 -2 -1]

[1 2 3]+[4 5 6] ..... [5 7 9]

[1 2 3]*2 ..... [2 4 6]
```

```
2
3
4
```

```
[1 2 3]*[2;3;4] ..... 1 2 3  20

[1 2 3].*[2 3 4] ..... [2 6 12]

[1 2 3].^2 ..... [1 4 9]

[1 2 3]./[2 3 4] ..... [.5 .666 .75]

sin([0 pi/2 pi]) ..... [0 1 0]
```

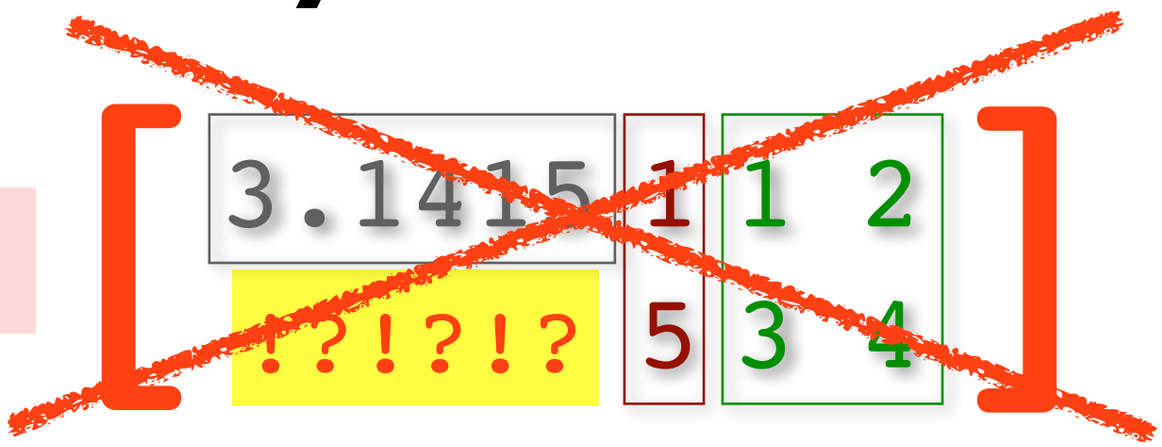
Cell Arrays

s `3.1415`

w
```
1
5
```

m
```
1  2
3  4
```

```
c = [s,w,m]?
```




```
c = {s,w,m}!
```

c { `3.1415` , `1` `5` , `1 2` `3 4` }

c{1} `3.1415`

c(2:3) { `1` `5` , `1 2` `3 4` }

Text

h = ’hello’ w = ’world’

[h,w] ’helloworld’

p = [h,’ ’,w] ’hello world’

{p;”how are you?”} { ’hello world’ ’how are you?” }

Structures

note1

note1.pitch = 'C'
note1.duration = 'quaver'
note1.register = 4

note2

note2.pitch = 'D'
note2.duration = 'crochet'
note2.register = 3

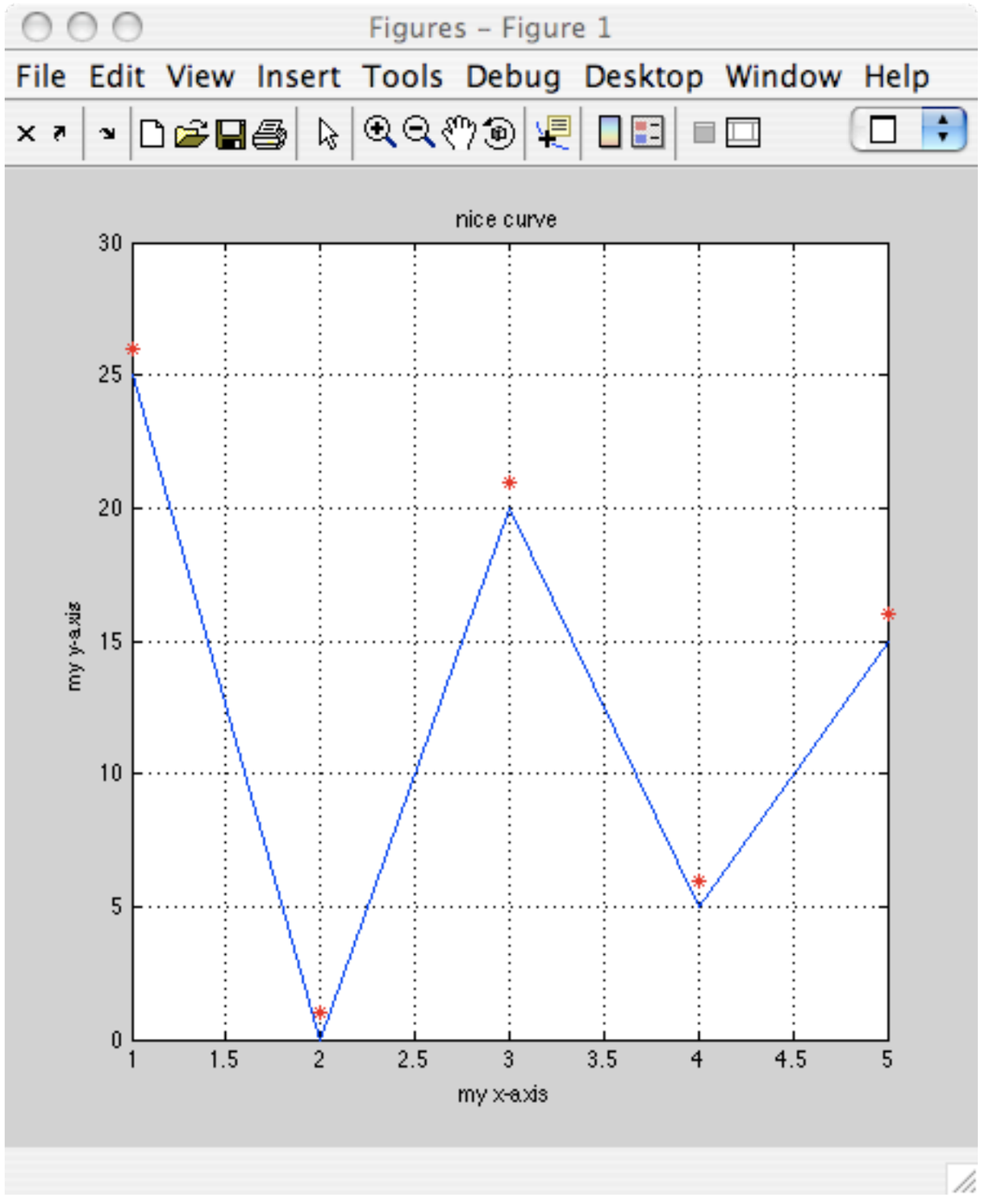
notes = [note1 note2]

notes.pitch 'C' 'D'

[notes.pitch] 'CD'

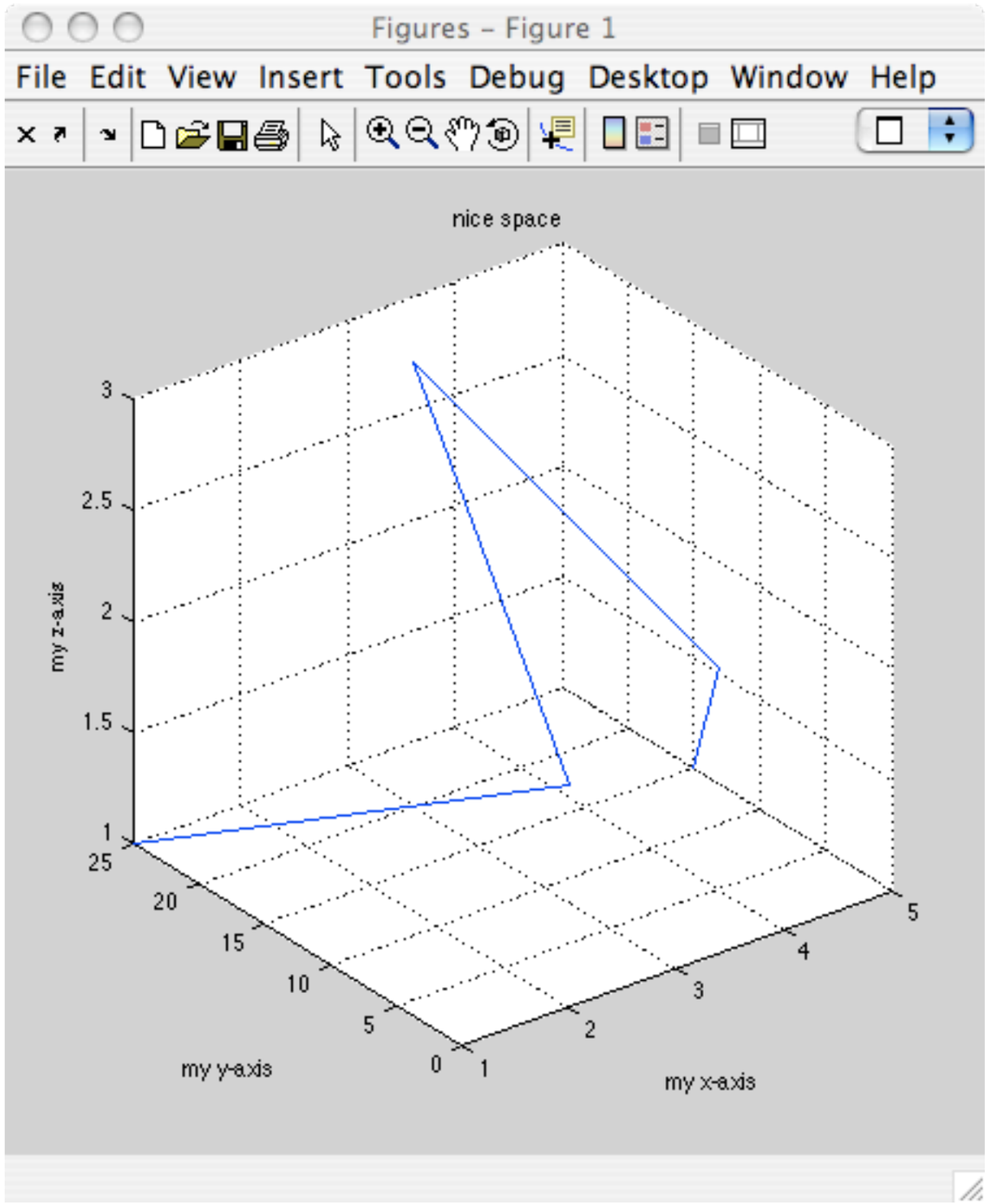
Plots

```
x = 1:5;
y = [25 0 20 5 15];
plot(x, y);
xlabel('my x-axis');
ylabel('my y-axis');
title('nice curve');
grid on
hold on
plot(x, y+1,'r*');
```



3D Plots

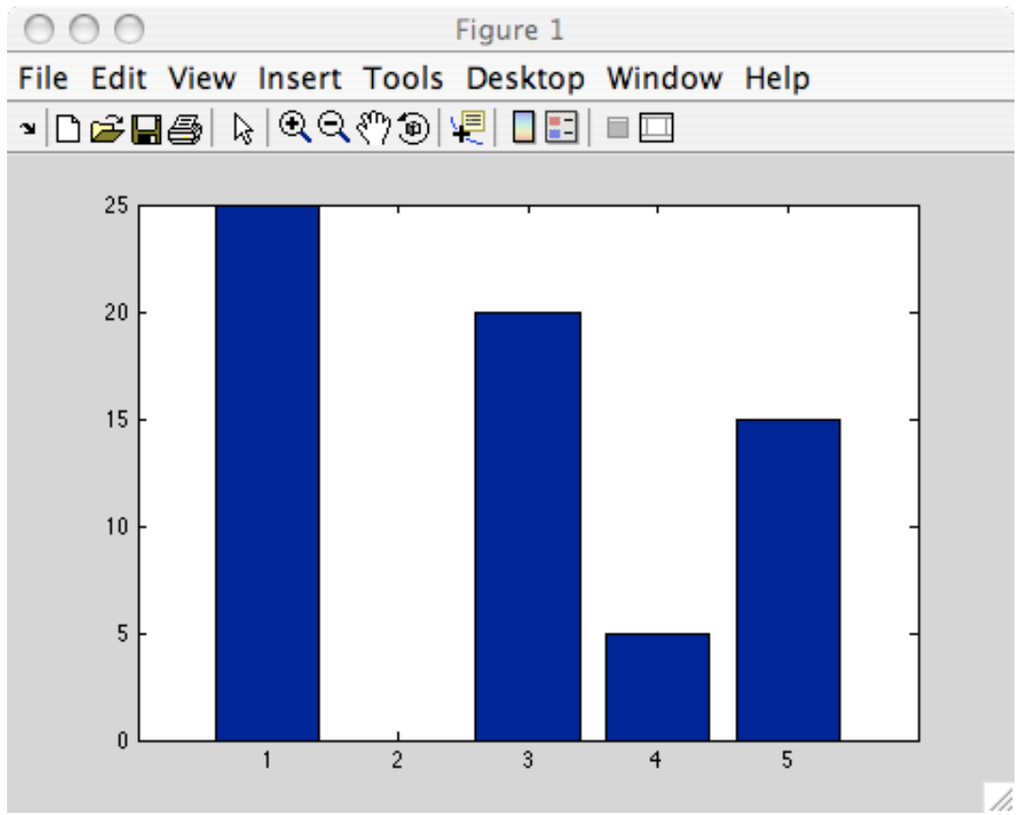
```
x = 1:5;
y = [25 0 20 5 15];
z = [1 2 3 2 1];
plot3(x, y, z);
xlabel('my x-axis');
ylabel('my y-axis');
zlabel('my z-axis');
title('nice space');
```



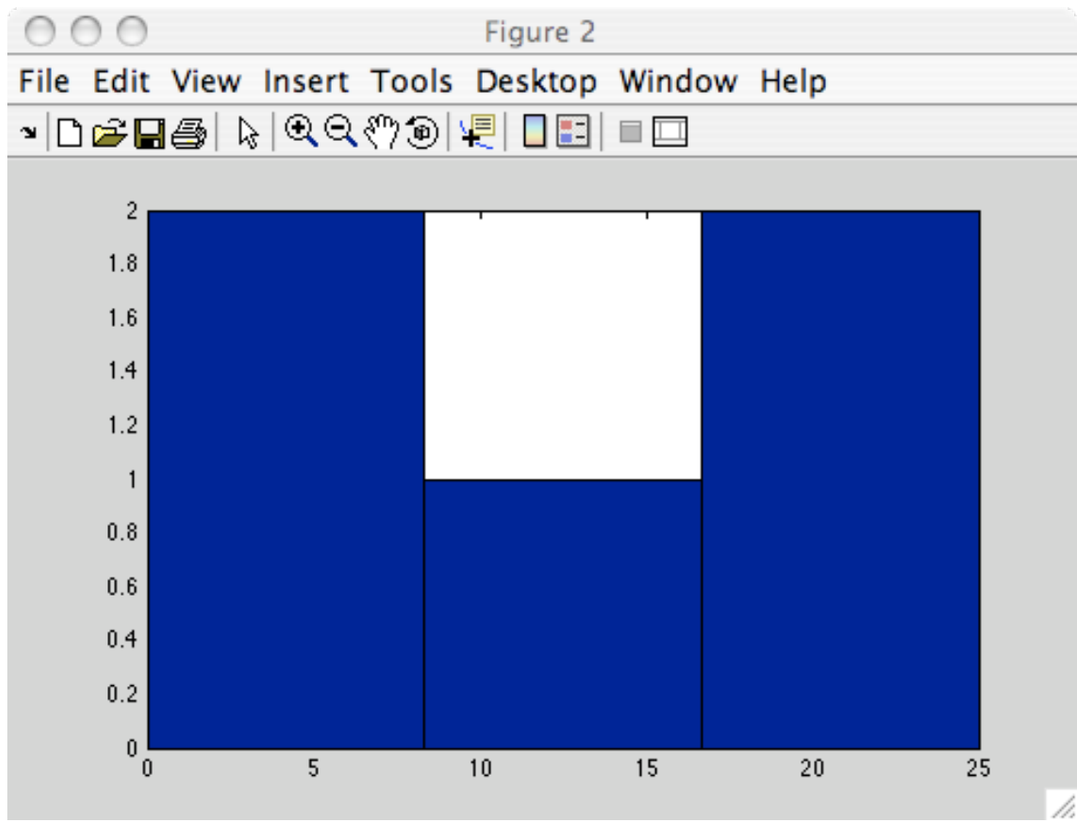
Bars, Histograms

```
x = 1:5;  y = [25 0 20 5 15];
```

```
figure, bar(x,y);
```

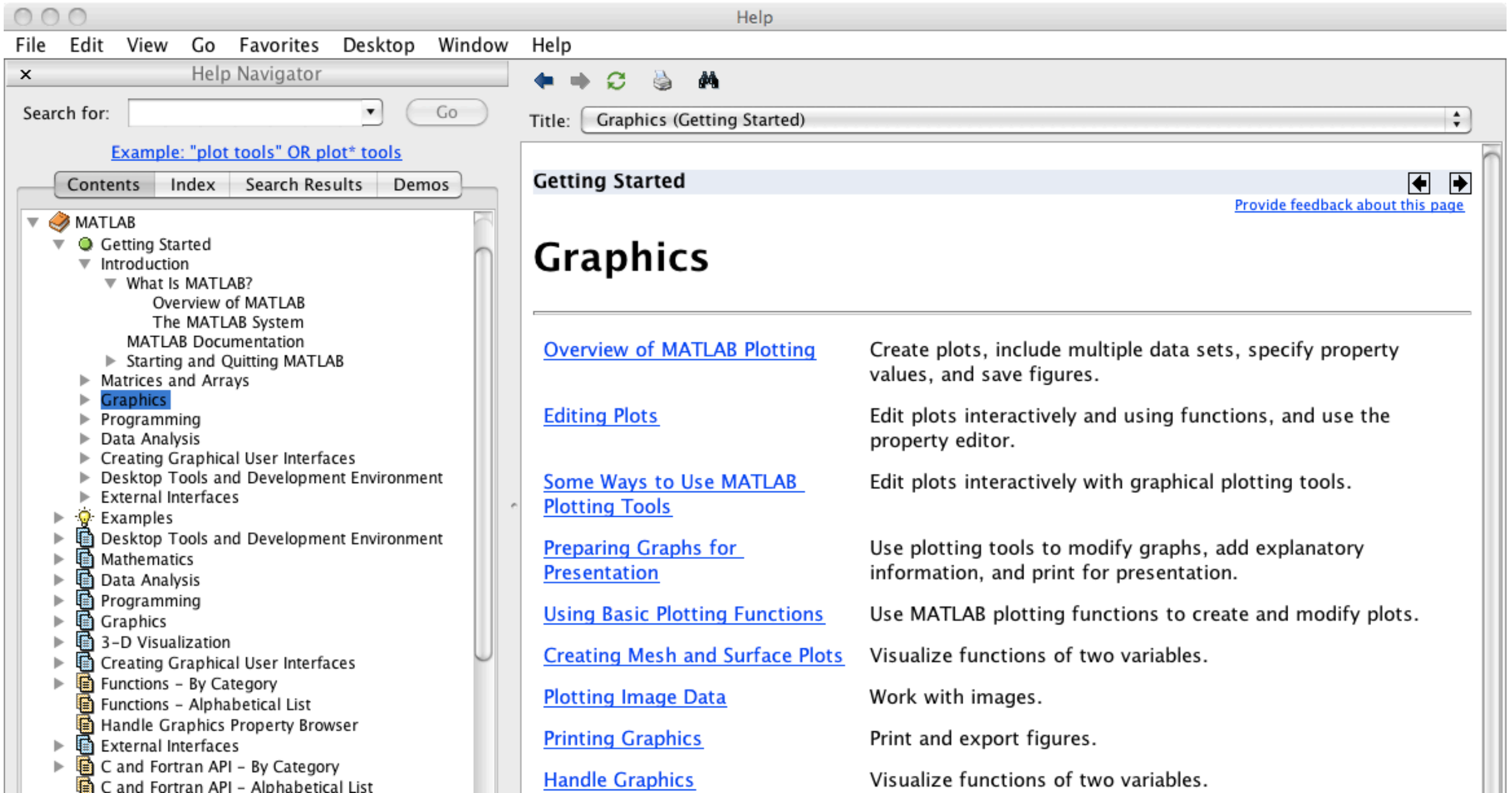



```
figure, hist(y,3);
```



Advanced graphics?

- Check *Matlab* documentation (type '**doc**')



Managing variables

```
>> a = 3;

>> b = 6;

>> who

Your variables are a b

>> save my_file

>> dir

 . .. my_file.mat
```

```
>> clear

>> who

(No variable!)

>> load my_file

>> who

Your variables are a b
```

Scripts

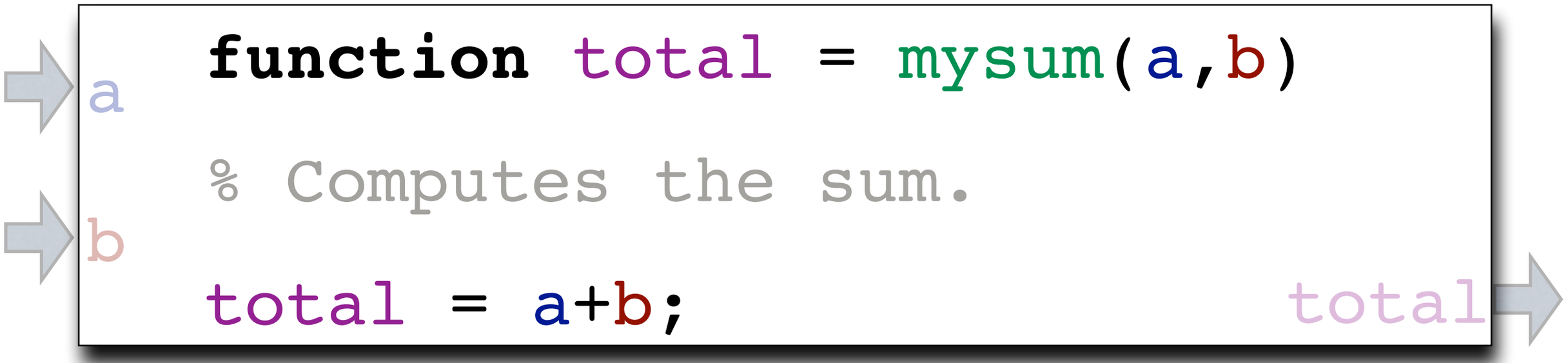
> New > Script

```
% Nice code!

a = 3;

b = 6

total = a+b
```

> Save myscript.m

```
>> myscript

b = 6

total = 9

>> who

Your variables
are a b total
```

Functions




```
>> mysum(3,3)
ans = 6
>> who
Your variable are a b total
```

(internal variables `a` `b` *are defined only inside the function)*

Example of Matlab Commands

- ***sqrt***: square-root
- ***sin***, ***cos***, ***abs***, ***max***, ***min***, ...
- tests: a **==** 1, a **<** 1, **&** (and), **|** (or), **~**(not)
- ***sum***, ***prod***, ***mean***, ***std***, ...
- ***corrcoef***: correlation matrix
- ***rand***: random numbers
- Check *Matlab* documentation: '`doc rand`', etc.

Programming

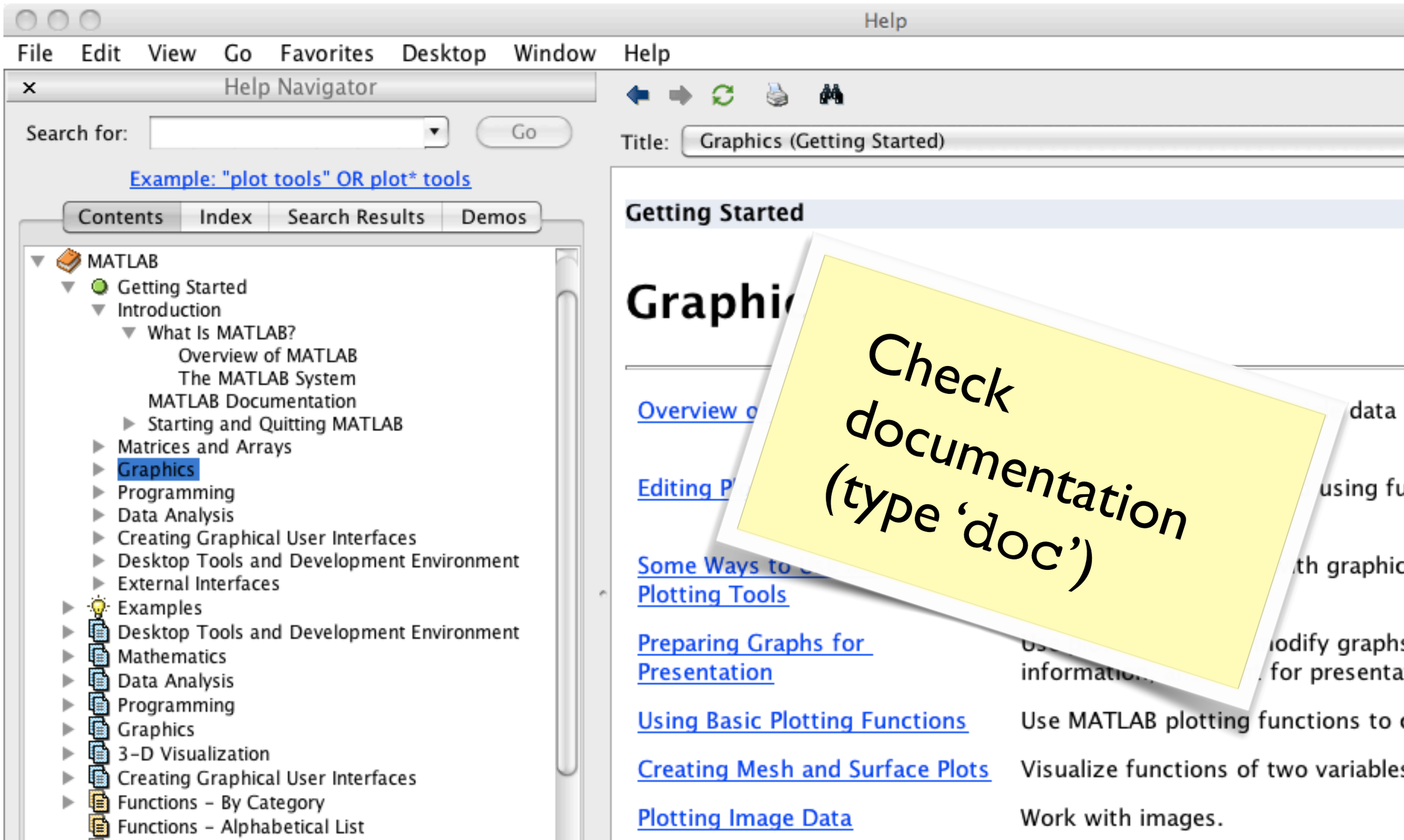
```
for i=1:5

    i*2

end
```

```
ans = 2
ans = 4
ans = 6
ans = 8
ans = 10
```

```
for i=1:5
  if i==3
     disp('three')
  else
     disp('not three')
  end
end
```

```
not three
not three
three
not three
not three
```

More about Matlab?



Exercises

1. Random composer

ChordProgression

12	16	14	12
11	15	13	11
7	11	9	7
4	8	6	4
0	4	2	0

Chord

4	2	0

BassLine

2. Simple synthesizer