Matlab Tutorial
Basics

Gonzalo Vaca-Castano

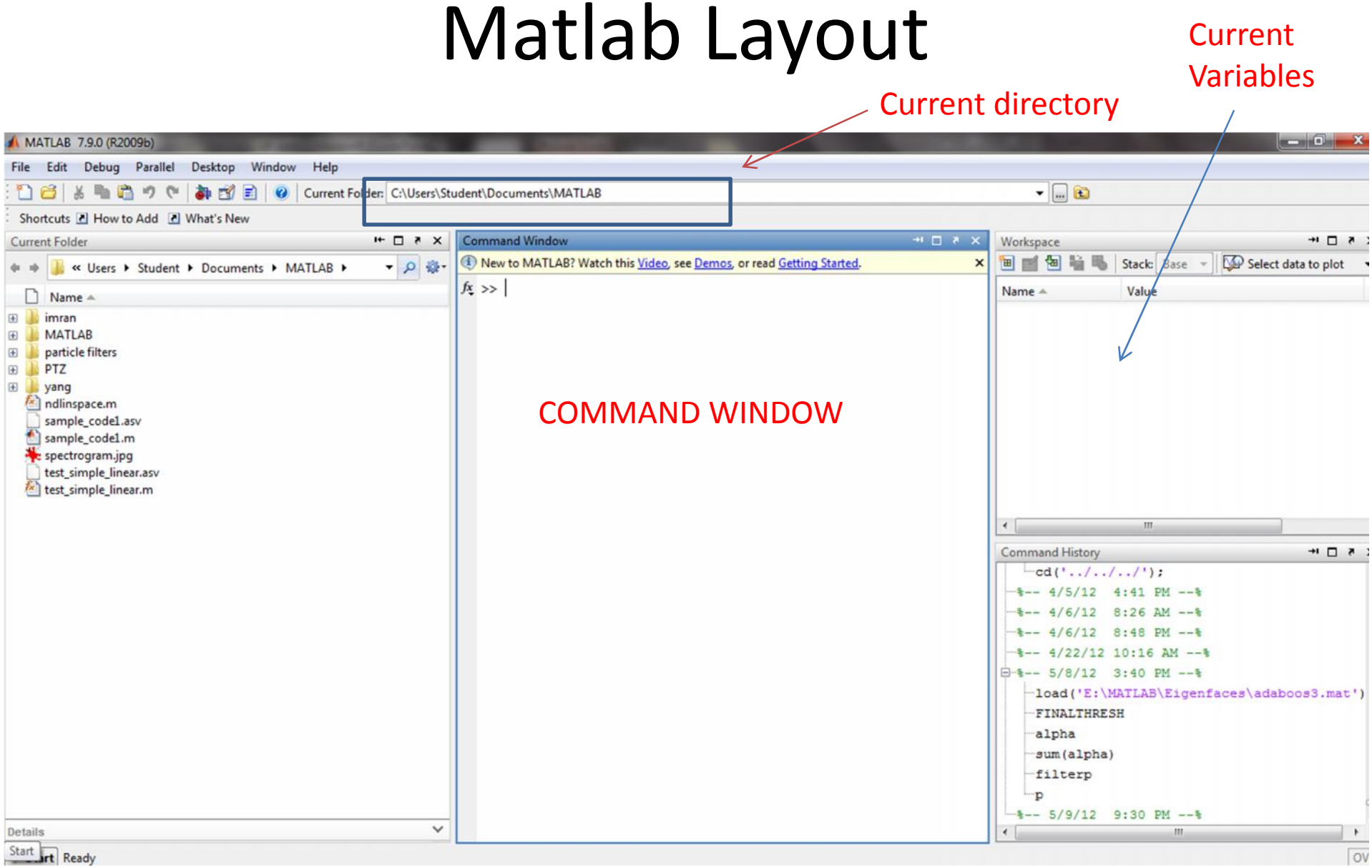
UCF.

2013

Introduction to mathematical programming

- Commonly used coding/computing environment in research: MATLAB (Matrix laboratory)
  - Ideal for numerical computation
  - Easy to learn
  - No compiling hassles
  - Quick testing of ideas!
  - Helpful tools and tricks for engineers/researchers
  - Execution of tasks on command prompt

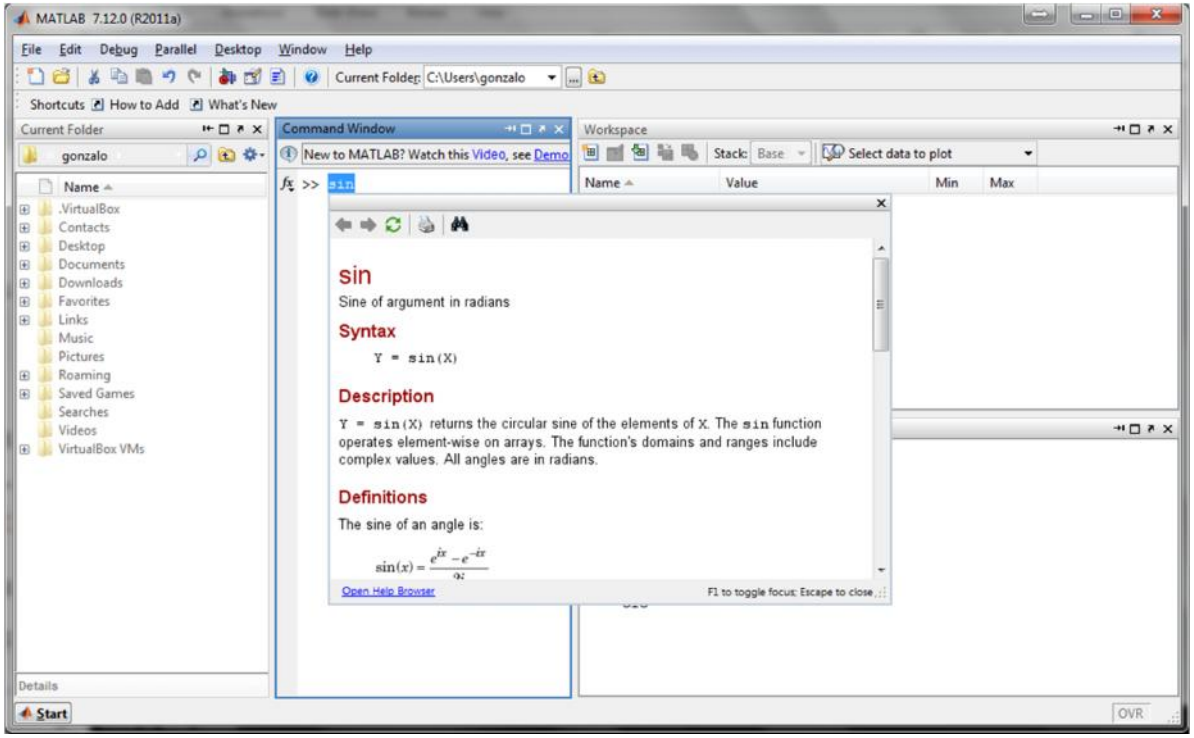
Matlab Layout



Getting Help

>> help ops;

Press F1



Google it !!! . www.mathworks.com usually have an answer to whatever you need

VARIABLES

Variables

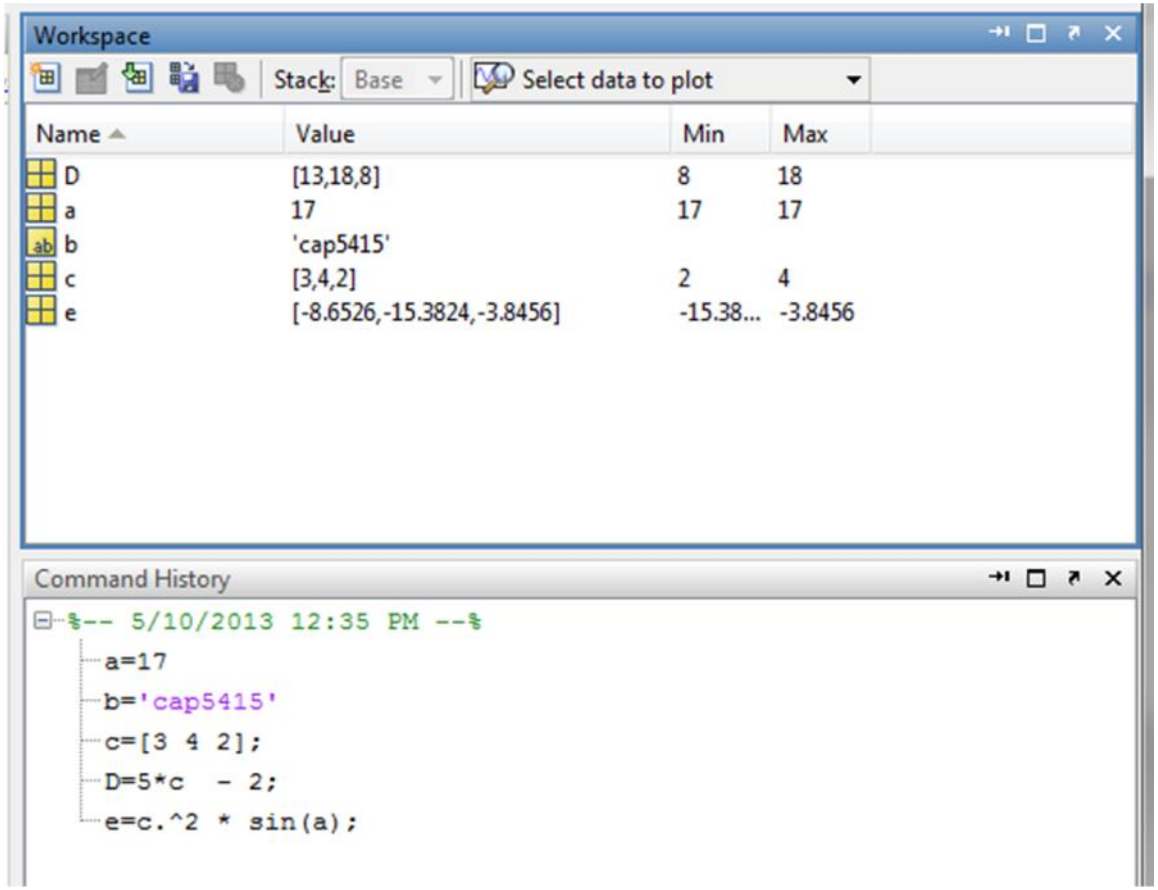
% Variables are defined as the assignment operator.

- a=17
- b='cap5415' ← Output is displayed
- c=[3 4 2]; ← Output is not displayed
- D=5*c - 2;
- e=c.^2 * sin(a);

Variables

% Variables are defined as the assignment operator.

- a=17
- b='cap5415'
- c=[3 4 2];
- D=5*c - 2;
- e=c.^2 * sin(a);



Variables

- Matlab does not require variable declaration.
- Matlab will declare variable automatically (be careful with typos in variable names)

Vectors

- Creating vectors:
  - example: >> x=[1 2 3 4 5]

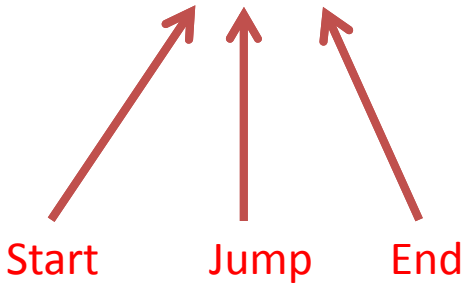
- Values retrieval

```
>> x(1)          %retrieve first element
>> x([1 4])      %retrieve first and fourth element
```

Comment marks in Matlab

Colon Operator (:)

- Creating vectors:
  - example: >>x= [1 2 3 4 5]
- Same as :
  - x=1:5
  - x=1:1:5



MORE EXAMPLES

x = 1:10
x = 1:2:10
x = 10:-2:1
y = 10:-2:1
x = 1:2:10

Colon Operator (:)

- Retrieving vectors:
  - example: >>x= [1 2 3 4 5]
  - a=x(2:4) %retrieve from second to fourth element

Vector Operations

- Dot product: ‘.’
- Sum all elements: sum()
- Sort elements: sort()
- Find histogram of elements: hist()
- Find average of elements: mean()

```
29  % Vector Addition
30  x=[2 5]
31  y=[4 3]
32  v = x+y
33
34  % Scalar  Product
35  a=3
36  w = a*v
37
38  % Operations on Vectors
39  sum(x)
40  max(x)
41  min(x)
42  mean(x)
43  x.^2
44  x .* y
45  x .^ y
46
47  % Inner (dot) product
48  dot(x,y)
49  x=[1 1]
50  y=[-1 1]
51  dot(x,y)
```

Matrices

- Creating a matrix:
  - Example: >> [1 1; 0 1]

- Matrix operations
  - Addition, multiplication, inverse, transpose
  - ‘+’ , ‘*’ , inv() , ‘
- Pinv = pseudo inverse

Defining Matrices

```
52
53  % Matrices 1
54  clc
55  A = zeros(10, 10)
56  A = rand(10, 10)
57  eye(3)
58
59  A=rand(5,5)
60  A(1, 4)
61
62  A=[1 2 3; 4 5 6]
63  B=2*ones(3,4)
```

10 rows, 10 columns zero

3x3 Identity Matrix

A =

0.7577	0.7060	0.8235	0.4387	0.4898
0.7431	0.0318	0.6948	0.3816	0.4456
0.3922	0.2769	0.3171	0.7655	0.6463
0.6555	0.0462	0.9502	0.7952	0.7094
0.1712	0.0971	0.0344	0.1869	0.7547

0.4387

B =

2	2	2	2
2	2	2	2
2	2	2	2

Matrix operation

```
67  % Matrices 2
68  A=[1 2 3; 4 5 6]
69  B=2*ones(2,3)
70  C=A*B'
71
72  A=[7 3 6; 5 1 4;2 9 8];
73  eye(3)*A*eye(3)
74
```

```
A =
     1     2     3
     4     5     6

B =
     2     2     2
     2     2     2

C =
    12    12
    30    30
```

```
ans =

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

Matrix operation

```
74
75  % Matrices 3
76  A=[7 3 6; 5 1 4;2 9 8];
77  C=A'
78
79  (A+C)'
80
81  A'+C'
82
```

C =

7	5	2
3	1	9
6	4	8

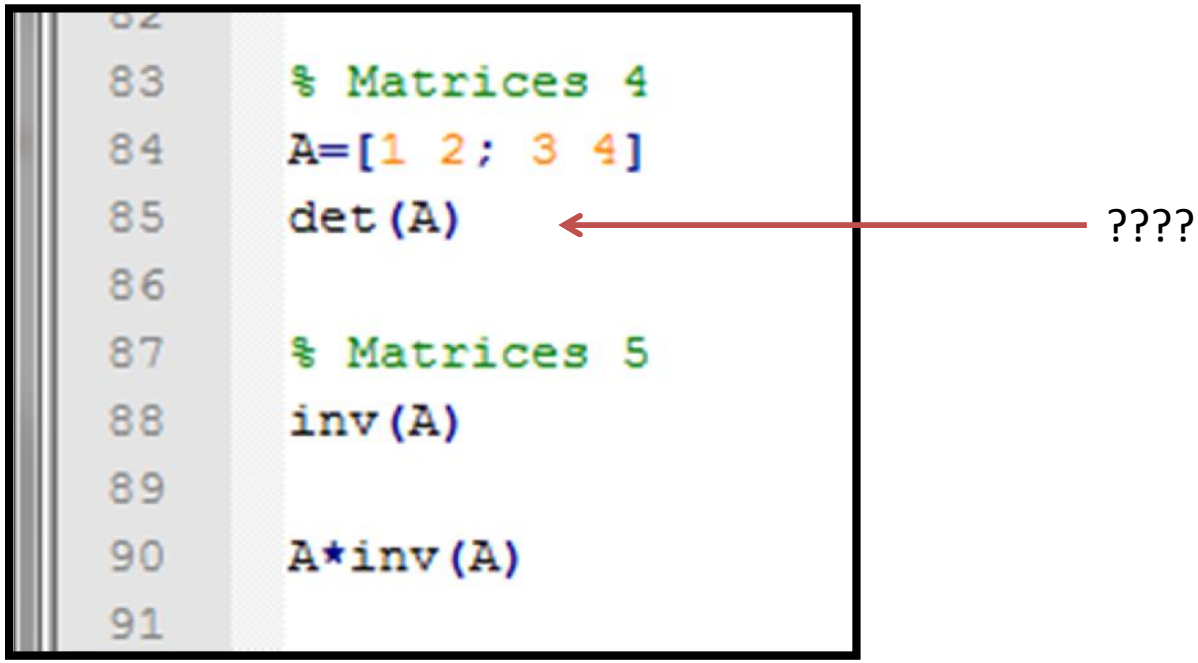
ans =

14	8	8
8	2	13
8	13	16

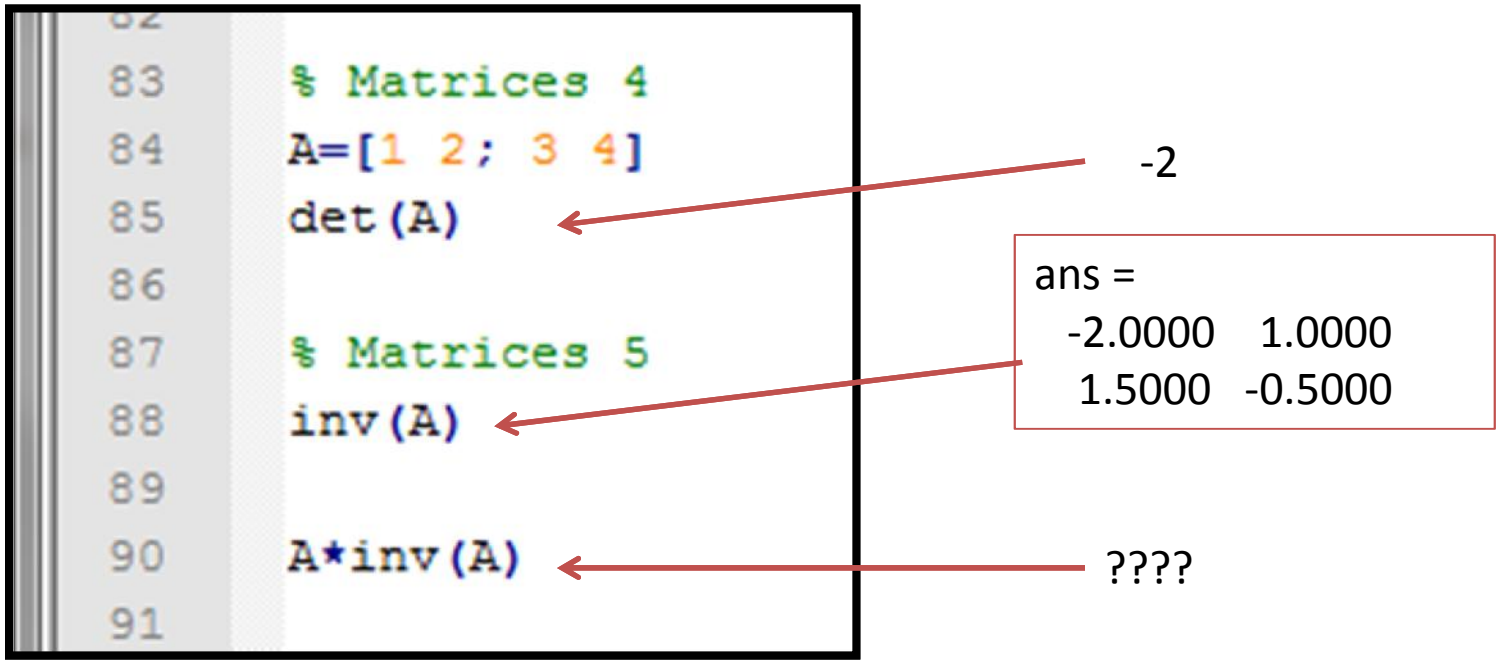
ans =

14	8	8
8	2	13
8	13	16

Matrix operation



Matrix operation



Solving a system of linear equations

- $3x + 2y + z = 0$
- $-\frac{1}{2} * x + \frac{2}{3} * y - z = 1/2$
- $x + -2^{\cdot\cdot} + z = -1$

- $\begin{bmatrix} 3 & 2 & 1 \\ -1/2 & 2/3 & -1 \\ 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 1/2 \\ -1 \end{bmatrix}$

Solving a system of linear equations

- a=[3 2 1; -1/2 2/3 -1; 1 -2 1]
- b=[0;1/2;-1]
- c=inv(a)*b;

- Output: c= [-0.1429; 0.3214; -0.2143]

Indexing 2-D Matrices

```
92
93  % Indexing 2-D Matrices
94  A=102:2:150
95  A=reshape(A,[5 5])
96  whos
97  whos A
98  A(4)
99  A(4)
100 A(4,1)
101 help sub2ind
102 sub2ind([5 5],2,3)
103
104 clear;
105 whos
106 clc
107 close all;
108 memory
```

```
A =
  Columns 1 through 15
   102   104   106   108   110   112   114   116   118   120   122   124   126   128   130

  Columns 16 through 25
   132   134   136   138   140   142   144   146   148   150
```

```
A =
   102   112   122   132   142
   104   114   124   134   144
   106   116   126   136   146
   108   118   128   138   148
   110   120   130   140   150
```

Name	Size	Bytes	Class	Attributes
A	5x5	200	double	
ans	1x1	8	double	

Name	Size	Bytes	Class	Attributes
A	5x5	200	double	

???

Indexing 2-D Matrices

```
92
93  % Indexing 2-D Matrices
94  A=102:2:150
95  A=reshape(A,[5 5])
96  whos
97  whos A
98  A(4)
99  A(4)
100 A(4,1)
101 help sub2ind
102 sub2ind([5 5],2,3)
103
104 clear;
105 whos
106 clc
107 close all;
108 memory
```

108

12

```
A =
   102   112   122   132   142
   104   114   124   134   144
   106   116   126   136   146
   108   118   128   138   148
   110   120   130   140   150
```

Useful if you have a multidimensional Array.

Some Useful functions

```
104    clear;
105    whos
106    clc
107    close all;
108    memory
```

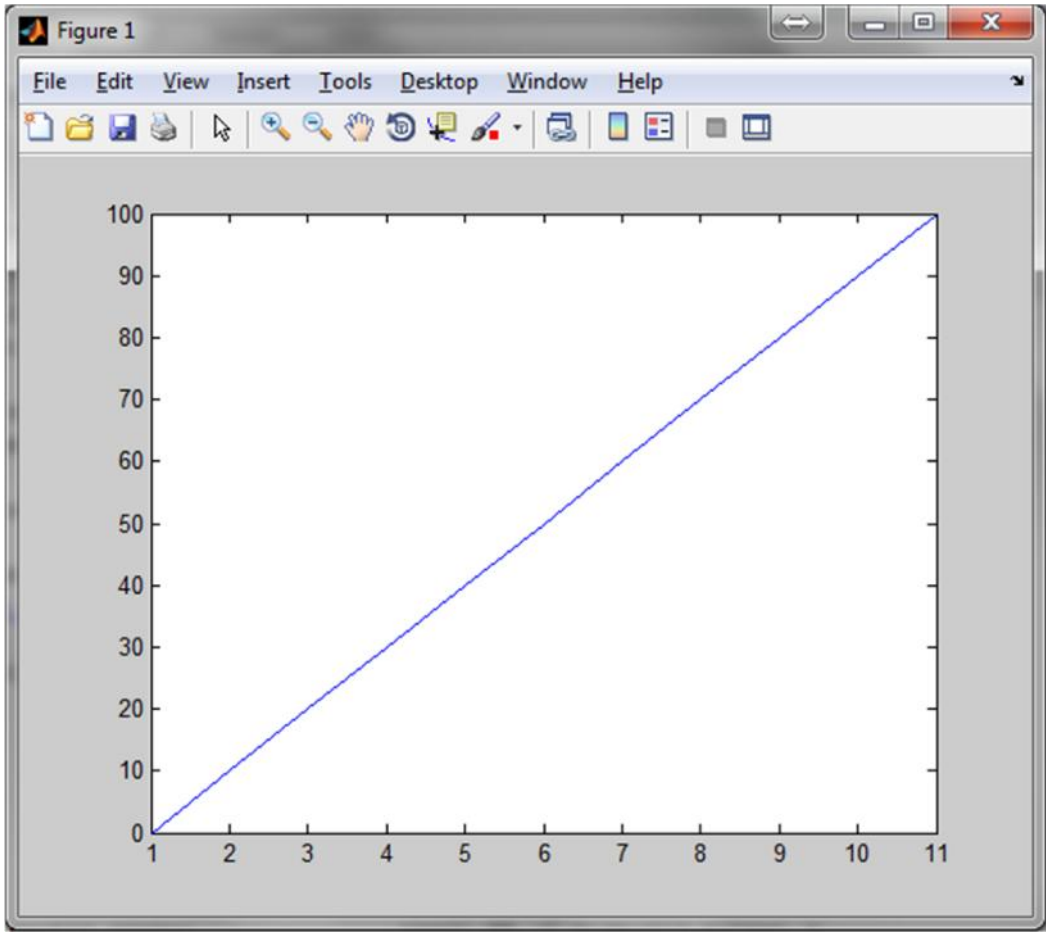
More Useful Function

- Load. Load data from MAT-file into workspace
- Save. Save workspace variables to file
  - save(filename, variables)
- Keyboard
- Pause

PLOT FUNCTIONS

Plot

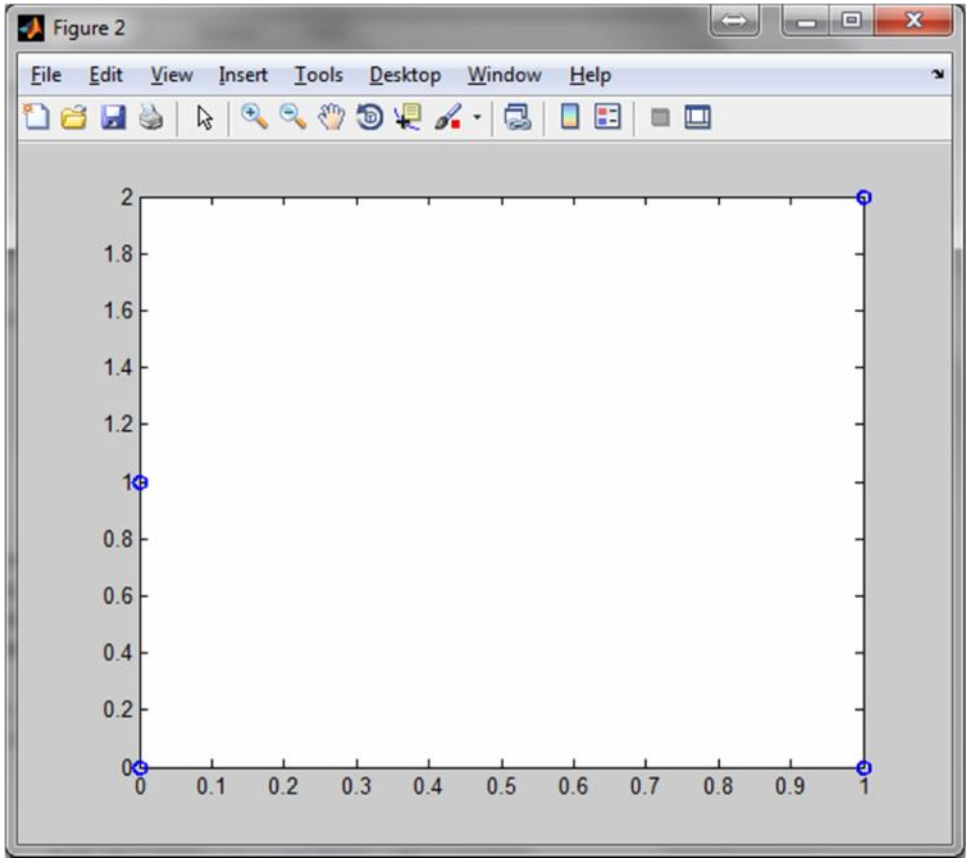
>>plot(0:10:101)



Plot

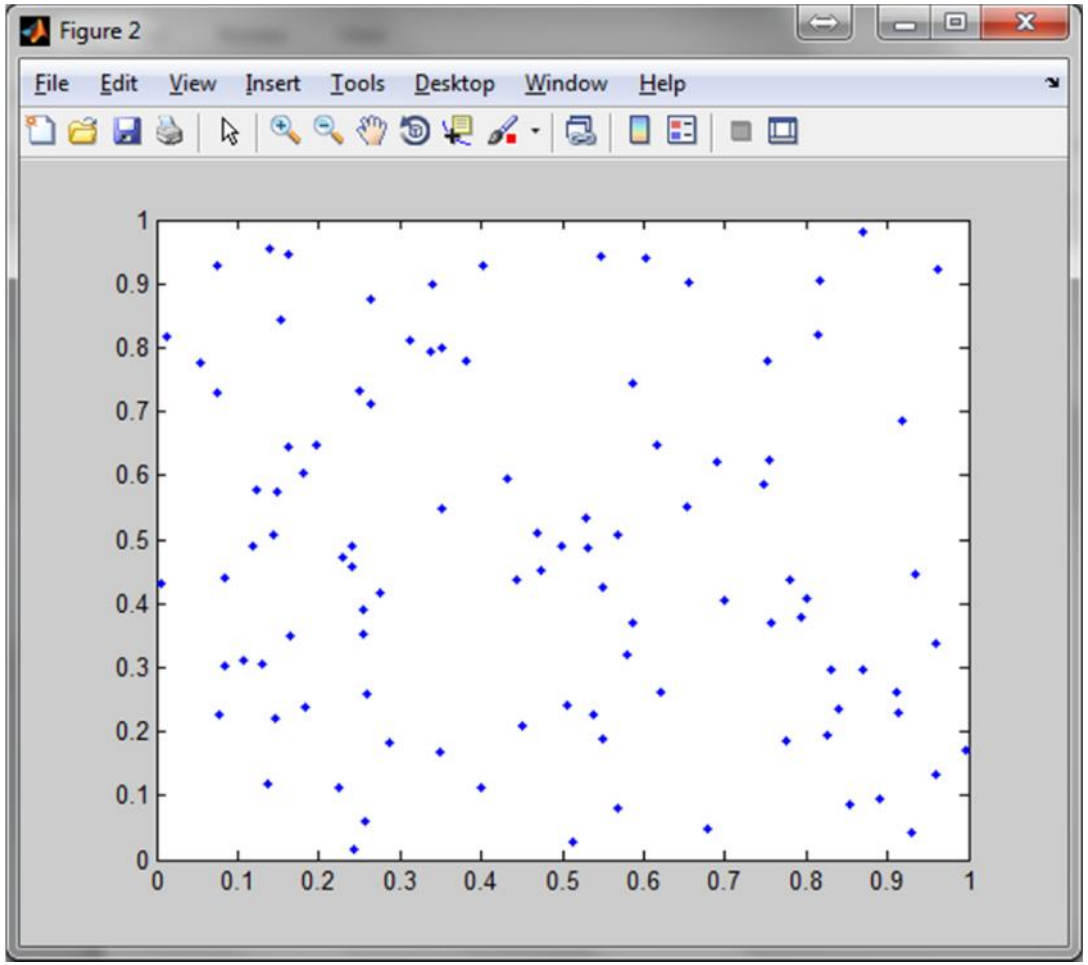
```
>> Q = [0, 0, 1, 1; 0, 1, 0, 2];
>> figure
>> plot(Q(1,:), Q(2,:), 'o', 'LineWidth', 2)
```

Display pattern.
Try ‘x’ for example
Help plot;



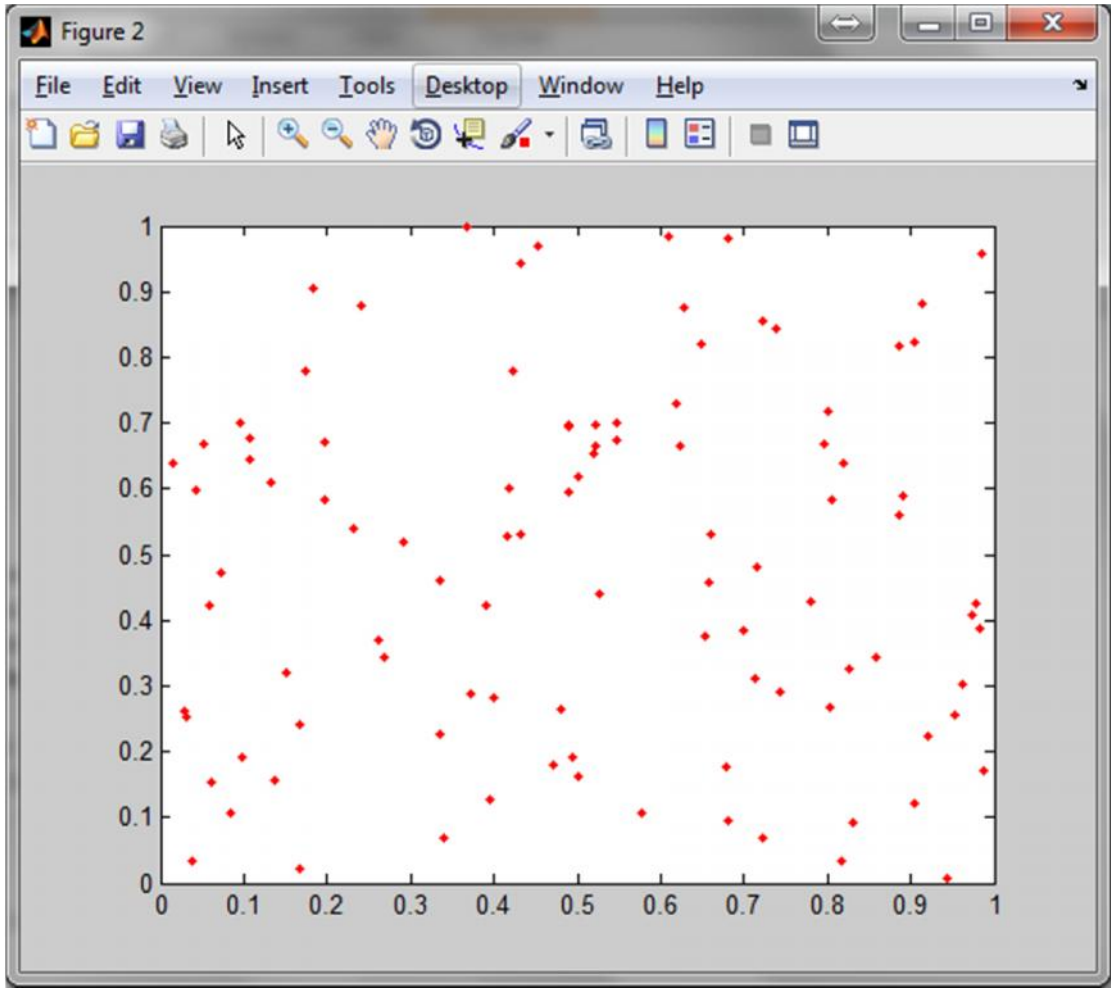
Plot

```
>> plot(rand(100,1), rand(100,1), 'b.');
```



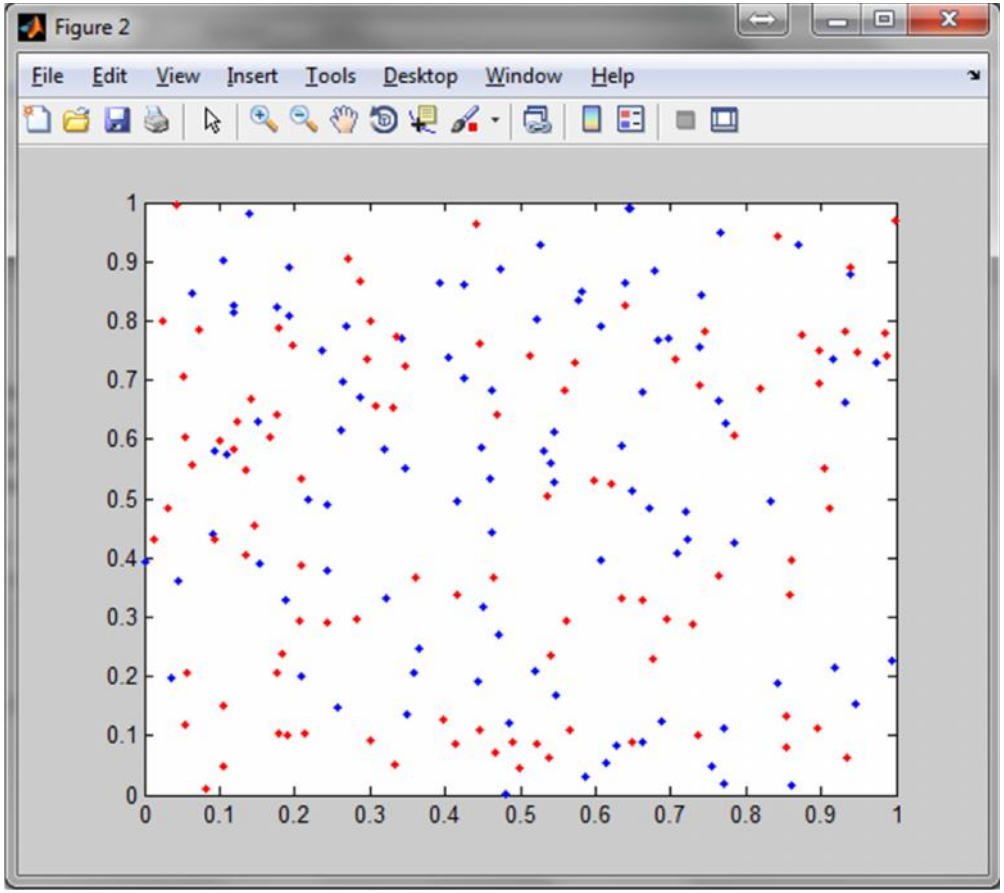
Plot

```
>> plot(rand(100,1), rand(100,1), 'r.');
```



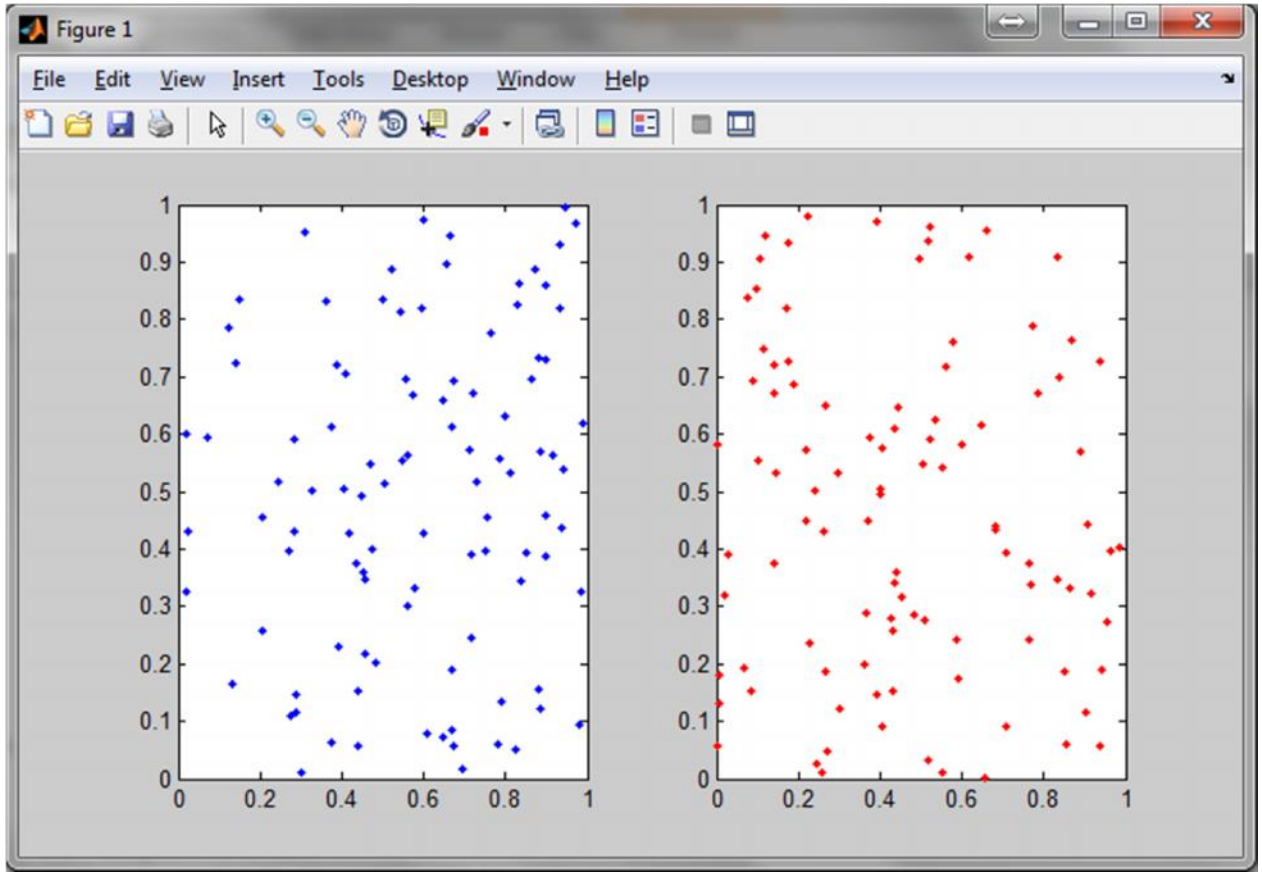
Hold on ; hold off

```
plot(rand(100,1), rand(100,1), 'b.');
hold on
plot(rand(100,1), rand(100,1), 'r.');
hold off
```



Subplot

```
figure(1);
subplot(1,2,1);
plot(rand(100,1), rand(100,1), 'b.');
subplot(1,2,2);
plot(rand(100,1), rand(100,1), 'r.');
```



See also

- xlabel
- ylabel
- axis
- title

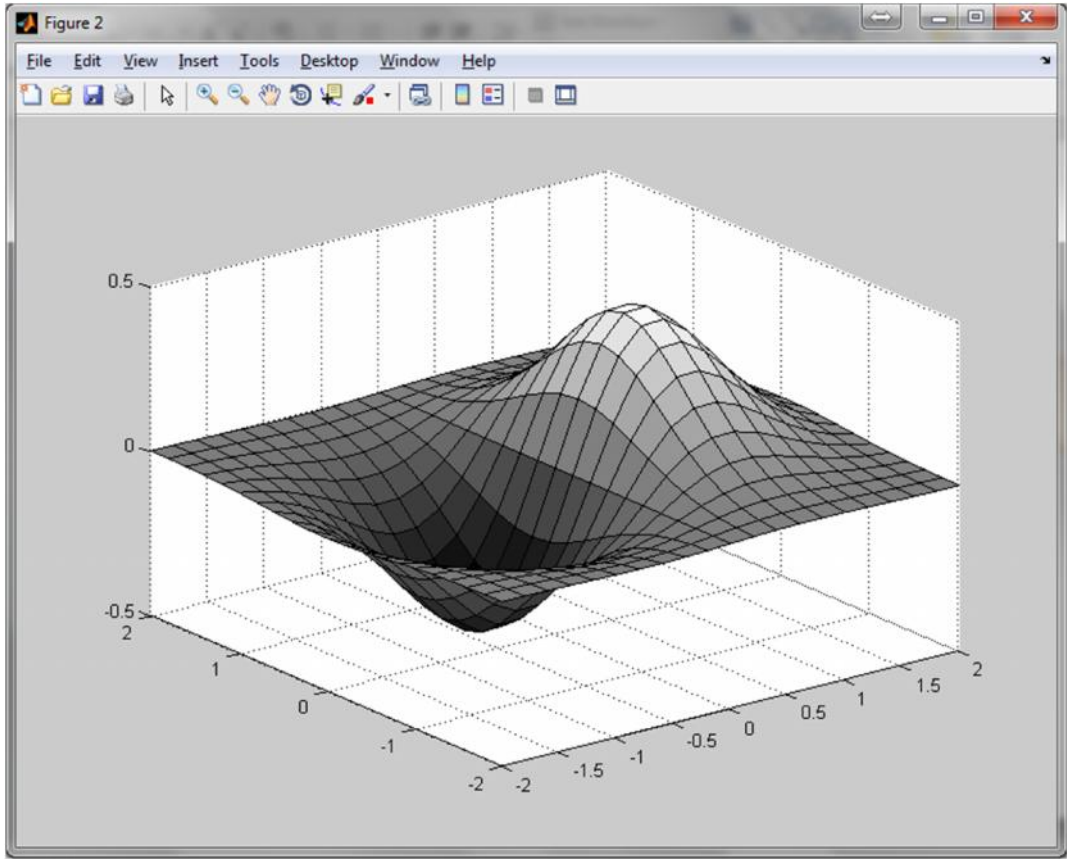
Meshgrid

- [X,Y] = meshgrid(1:3,10:14)

- X =
- 1 2 3
- 1 2 3
- 1 2 3
- 1 2 3
- 1 2 3

- Y =
- 10 10 10
- 11 11 11
- 12 12 12
- 13 13 13
- 14 14 14

```
[X,Y] = meshgrid(-2:.2:2, -2:.2:2);
Z = X .* exp(-X.^2 - Y.^2);
surf(X,Y,Z)
```

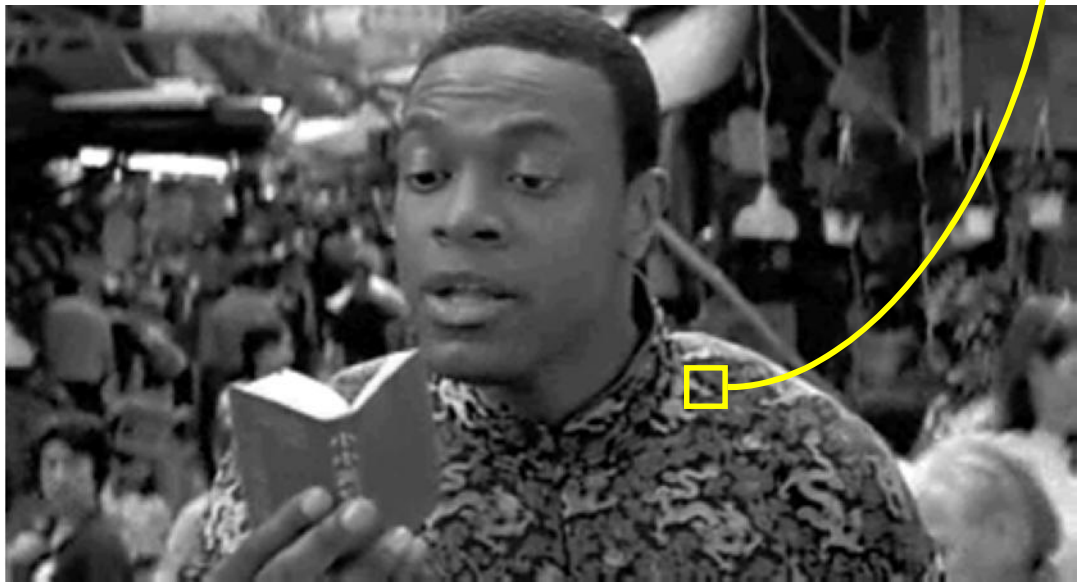


SIMPLE IMAGE

What is an Image?

- 2-D array of numbers (intensity values, gray levels)
- Gray levels 0 (black) to 255 (white)
- Color image is 3 2-D arrays of numbers
  - Red
  - Green
  - Blue

34	23	58	89	105	97	89	83	83	81
97	39	23	67	75	89	89	89	89	81
139	73	26	67	67	58	75	81	81	75
171	147	97	105	64	7	23	58	81	83
56	89	147	155	114	73	48	58	73	81
23	64	115	148	155	114	48	26	48	73
23	56	74	81	73	64	73	81	89	89
73	56	45	62	57	56	73	81	82	82
97	64	81	103	105	97	89	82	82	82
97	81	89	86	89	97	81	78	82	97



Image

```
112  clear all;close all;clc;
113
114  % - imread. Image reading from disk.
115  I1 = imread('groceries.jpg');
116  [rows cols chan]=size(I1);
117  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
118  % - imshow, imagesc. Showing images and matrices in matlab.
119  figure;
120  imshow(I1);
121
122  figure;imshow(I1(:,:,1));
123  figure;imshow(I1(:,:,2));
124  figure;imshow(I1(:,:,3));
125
126  % - rgb2gray. Convert color image to graysca
127  I2 = rgb2gray(I1);
128
129  figure;
130  imshow(I2);
131
132  figure;
133  imagesc(I2)
134
135  imwrite(I2,'groceries_gray.jpg');
136
137  [rows cols chan]=size(I2);
138
139  whos
```

Read Image File

rows =684; cols =912; chan =3

Image (Channel 1)

```
112  clear all;close all;clc;
113
114  % - imread. Image reading from disk.
115  I1 = imread('groceries.jpg');
116  [rows cols chan]=size(I1);
117  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
118  % - imshow, imagesc. Showing images and matrices in matlab.
119  figure;
120  imshow(I1);
121
122  figure;imshow(I1(:,:,1));
123  figure;imshow(I1(:,:,2));
124  figure;imshow(I1(:,:,3));
125
126  % - rgb2gray. Convert color image to grayscale.
127  I2 = rgb2gray(I1);
128
129  figure;
130  imshow(I2);
131
132  figure;
133  imagesc(I2)
134
135  imwrite(I2,'groceries_gray.jpg');
136
137  [rows cols chan]=size(I2);
138
139  whos
```

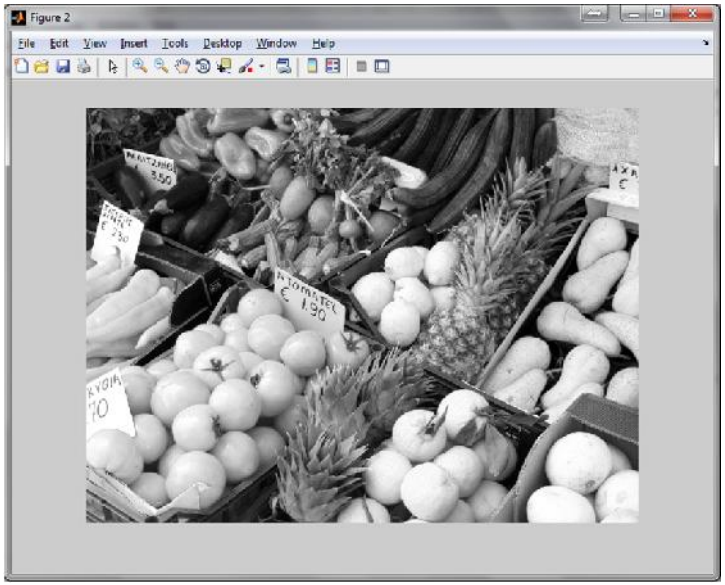
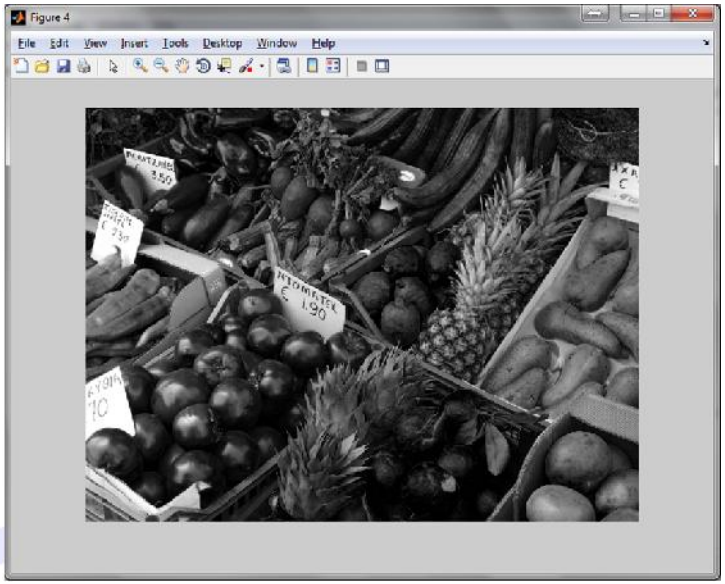


Image (Channel 2)

```
112  clear all;close all;clc;
113
114  % - imread. Image reading from disk.
115  I1 = imread('groceries.jpg');
116  [rows cols chan]=size(I1);
117  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
118  % - imshow, imagesc. Showing images and matrices in matlab.
119  figure;
120  imshow(I1);
121
122  figure;imshow(I1(:,:,1));
123  figure;imshow(I1(:,:,2));
124  figure;imshow(I1(:,:,3));
125
126  % - rgb2gray. Convert color image to grayscale.
127  I2 = rgb2gray(I1);
128
129  figure;
130  imshow(I2);
131
132  figure;
133  imagesc(I2)
134
135  imwrite(I2,'groceries_gray.jpg');
136
137  [rows cols chan]=size(I2);
138
139  whos
```

Image (Channel 3)

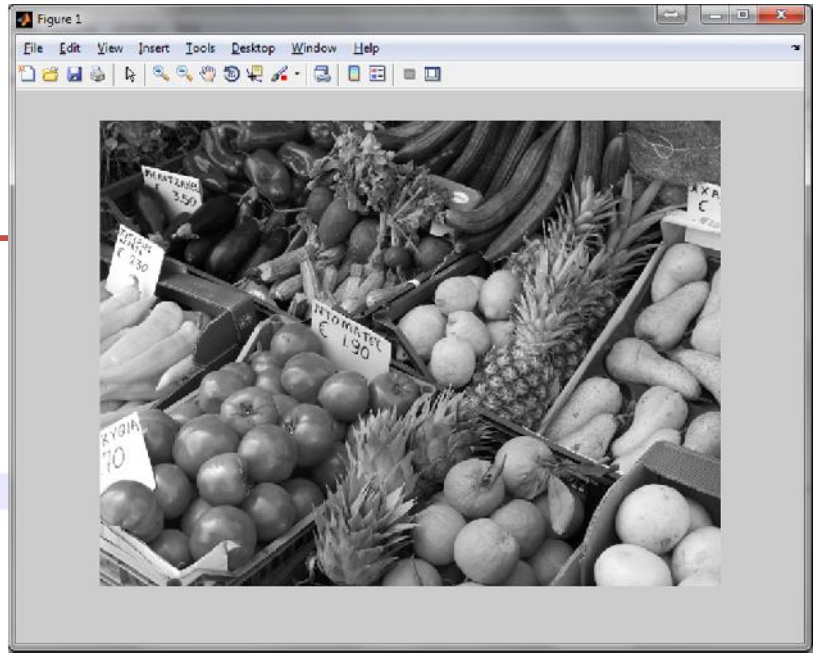
```
112  clear all;close all;clc;
113
114  % - imread. Image reading from disk.
115  I1 = imread('groceries.jpg');
116  [rows cols chan]=size(I1);
117  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
118  % - imshow, imagesc. Showing images and matrices in matlab.
119  figure;
120  imshow(I1);
121
122  figure;imshow(I1(:,:,1));
123  figure;imshow(I1(:,:,2));
124  figure;imshow(I1(:,:,3));
125
126  % - rgb2gray. Convert color image to grayscale.
127  I2 = rgb2gray(I1);
128
129  figure;
130  imshow(I2);
131
132  figure;
133  imagesc(I2)
134
135  imwrite(I2,'groceries_gray.jpg');
136
137  [rows cols chan]=size(I2);
138
139  whos
140
141  close all;
```



Image

```
112  clear all;close all;clc;
113
114  % - imread. Image reading from disk.
115  I1 = imread('groceries.jpg');
116  [rows cols chan]=size(I1);
117  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
118  % - imshow, imagesc. Showing images and matrices in matlab.
119  figure;
120  imshow(I1);
121
122  figure;imshow(I1(:,:,1));
123  figure;imshow(I1(:,:,2));
124  figure;imshow(I1(:,:,3));
125
126  % - rgb2gray. Convert color image to grayscale.
127  I2 = rgb2gray(I1);
128
129  figure;
130  imshow(I2);
131
132  figure;
133  imagesc(I2)
134
135  imwrite(I2,'groceries_gray.jpg');
136
137  [rows cols chan]=size(I2);
138
139  whos
140
```

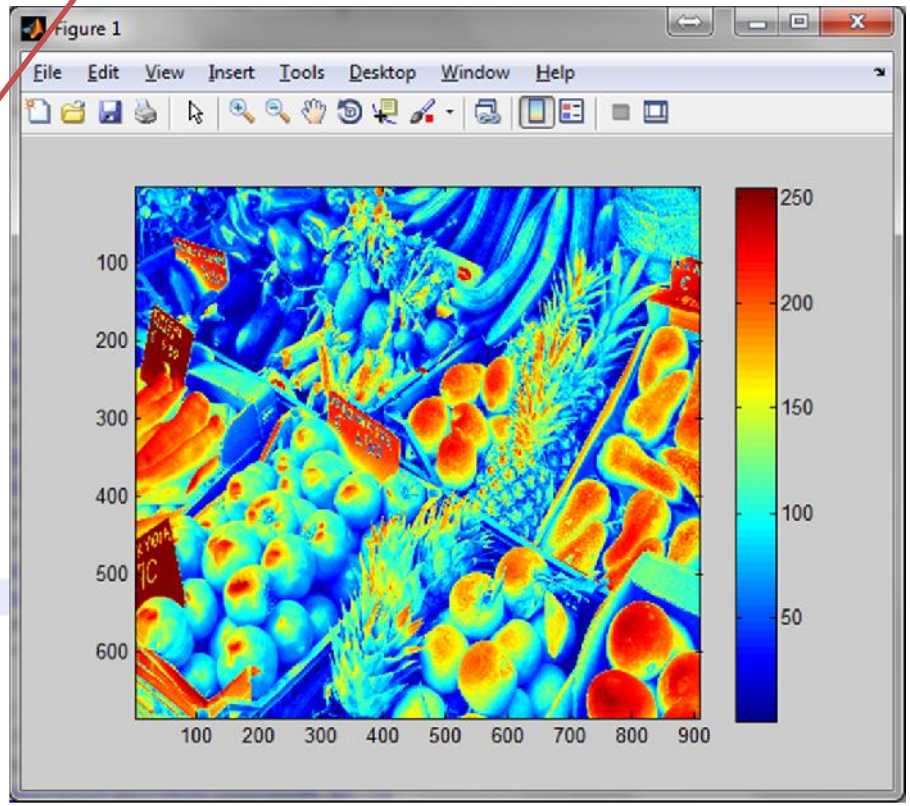
Convert to Gray Scale



Image

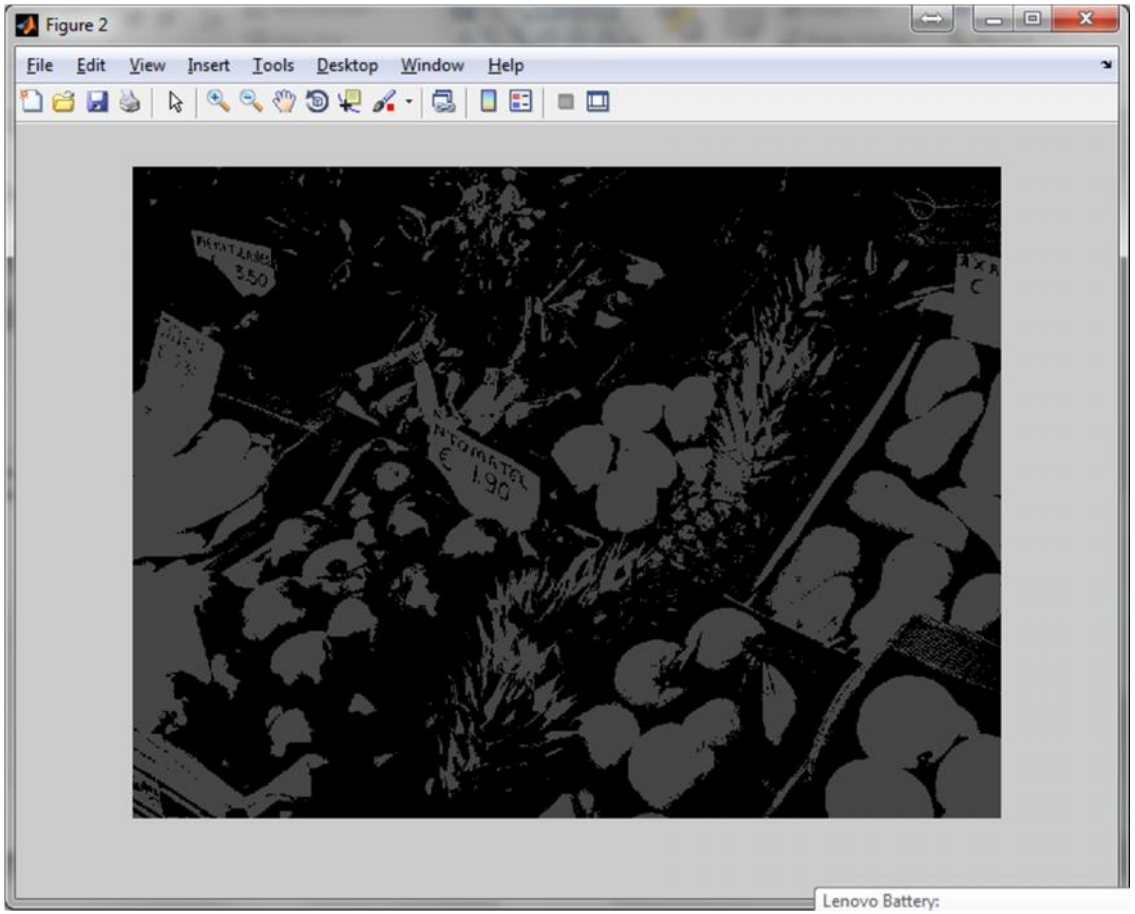
```
112  clear all;close all;clc;
113
114  % - imread. Image reading from disk.
115  I1 = imread('groceries.jpg');
116  [rows cols chan]=size(I1);
117  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
118  % - imshow, imagesc. Showing images and matrices in matlab.
119  figure;
120  imshow(I1);
121
122  figure;imshow(I1(:,:,1));
123  figure;imshow(I1(:,:,2));
124  figure;imshow(I1(:,:,3));
125
126  % - rgb2gray. Convert color image to grayscale.
127  I2 = rgb2gray(I1);
128
129  figure;
130  imshow(I2);
131
132  figure;
133  imagesc(I2)
134
135  imwrite(I2,'groceries_gray.jpg');
136
137  [rows cols chan]=size(I2);
138
139  whos
140
```

Save Image File



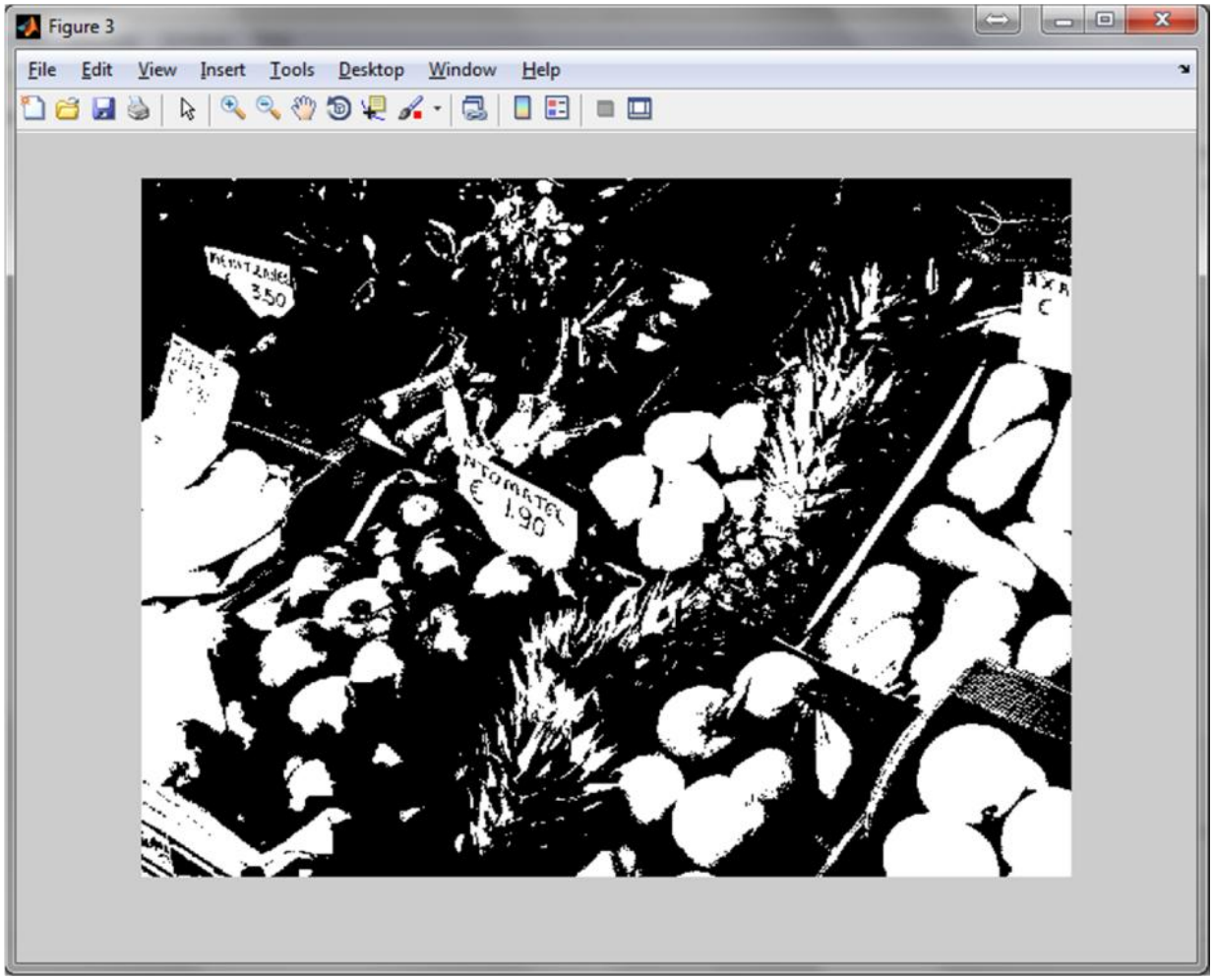
Thresholding, selecting indices, initializing an array

```
thr=128;
I3 = zeros(rows,cols,'uint8');
I3(find(I2>thr))=70;
I3(find(I2<=thr))=0;
figure;
imshow(I3);
```



mat2gray.
Convert arbitrary matrix to scaled, intensity image

```
I4=mat2gray(I3);
figure;
imshow(I4);

unique(I4)
```

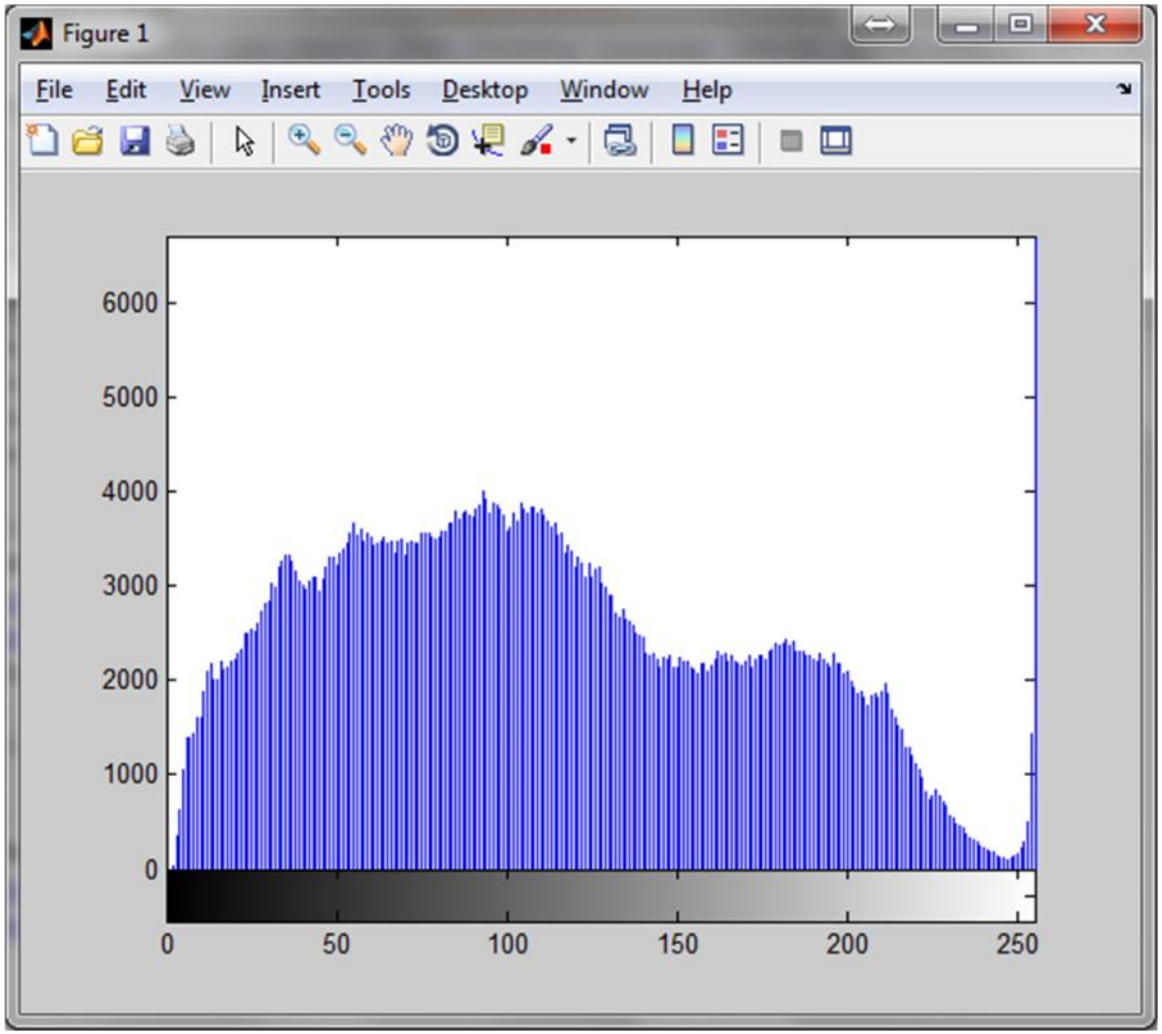



```
ans =

     0
     1
```

hist.

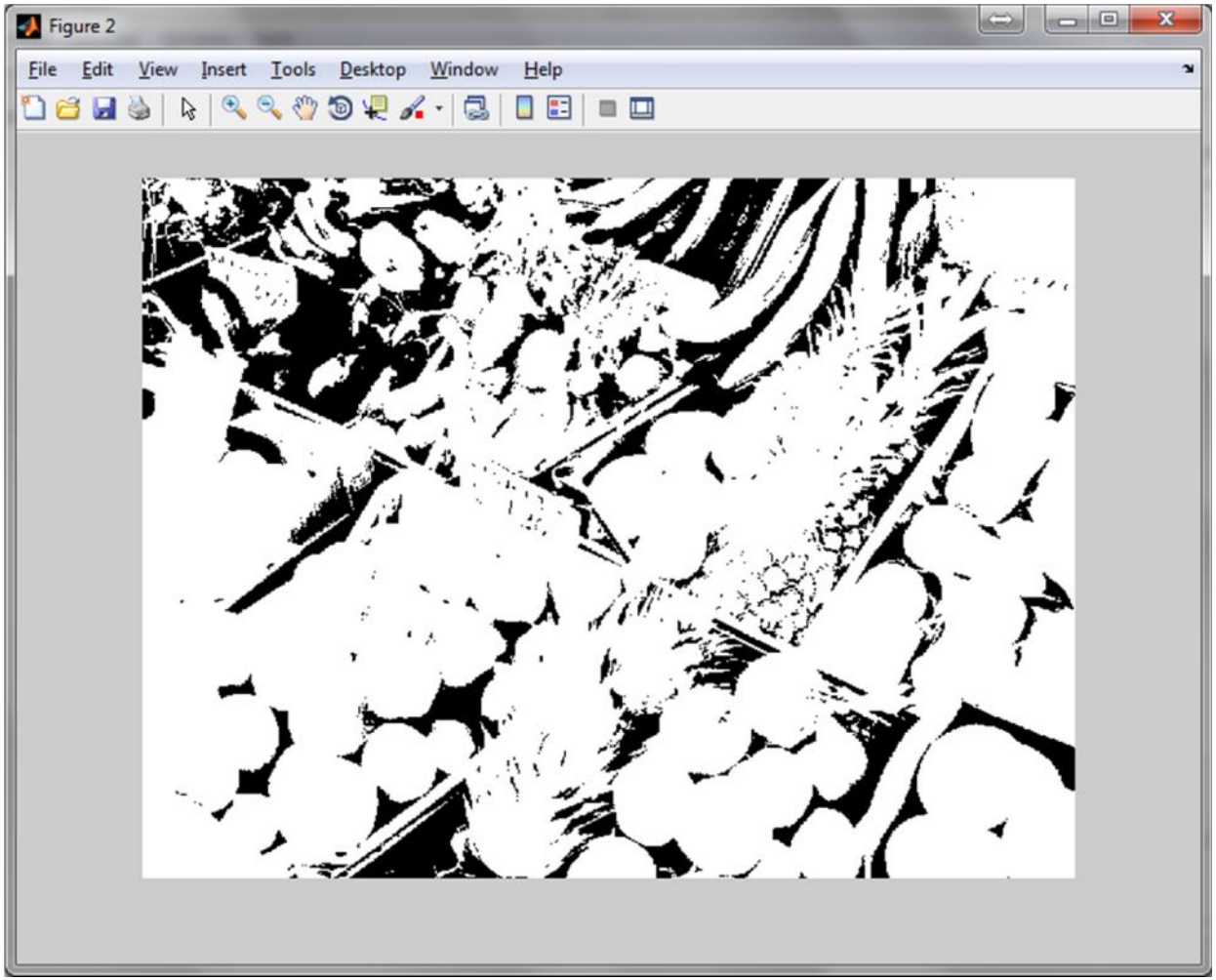
histogram from matrix/Image.

```
figure;
imhist(I2);
```



im2bw.
Threshold gray image

```
I5=im2bw(I2,0.2); % thr 0 to 1
figure;
imshow(I5);
```



MATLAB PROGRAMMING

A little bit on Efficiency

- Matlab is optimized for Matrix Operation
- For-Loops to access data works well in C and other languages, but must me avoided when there is a matrix operation to replace it

Example

- Example of computing distances between all pairs of points using matrices

```
a=uint8(100*rand(1,5))
b=uint8(100*rand(1,5))
diff=zeros(5,5)
for i=1:5
   diff(i,:)=double(b)-double(a(i));
end
```

- No double loops !!!

Example

- Another Matlab Way in 1 line.

```
diff2=repmat(double(b),5,1)-repmat(double(a'),1,5)
```

```
% - repmat. repeat matrices
a=[1 2 3 4]
repmat(a,[3 2])
```

```
ans =
     1     2     3     4     1     2     3     4
     1     2     3     4     1     2     3     4
     1     2     3     4     1     2     3     4
```

Functions

CALL A FUNCTION

```
>> x=testFunc(3,2)
```

testFunc.m

```
function result=testFunc(a,b)

result=(a^2)-b;
```

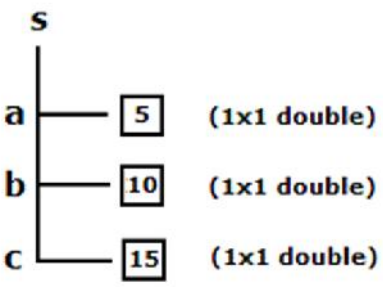
Cells

- Are containers of any kind of data

```
>> mycell = cell(3,4);
>> mycell{1,1}=zeros(3,2)

mycell =

   [3x2 double]    []    []    []
          []    []    []    []
          []    []    []    []

>> mycell{1,2}='test'

mycell =

   [3x2 double]    'test'    []    []
          []       []    []    []
          []       []    []    []
```

Struct. Example 1

```
s.a = 5;
s.b = 10;
s.c = 15;
```

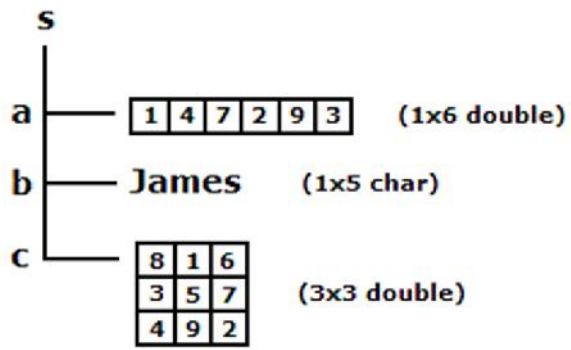


OR . . .

```
s = struct('a', 5, 'b', 10, 'c', 15);
```

Struct. Example 2

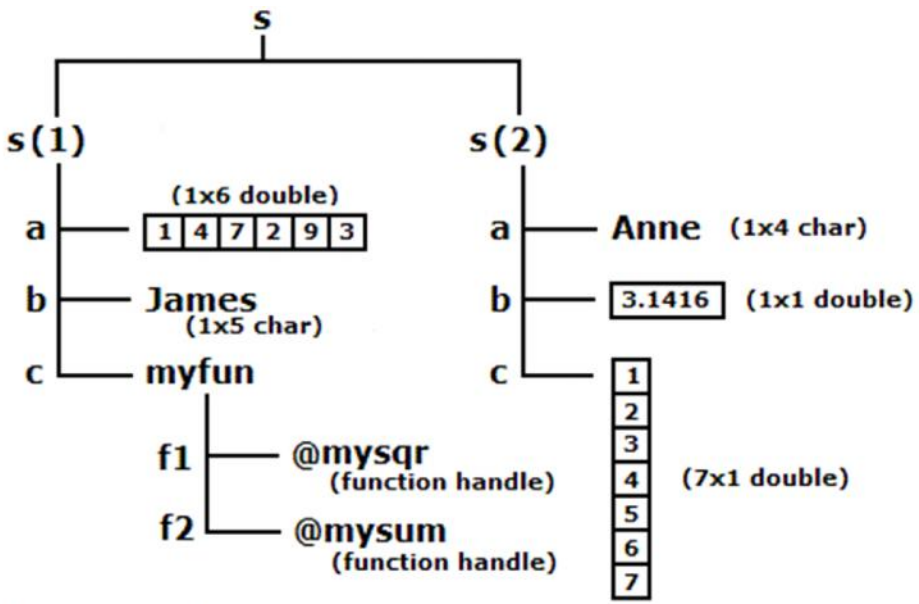
```
s.a = [1 4 7 2 9 3];
s.b = 'James';
s.c = magic(3);

    OR . . .

s = struct('a', [1 4 7 2 9 3], 'b', 'James', 'c', magic(3));
```



Nested Structures

```
s(1).a = [1 4 7 2 9 3];
s(2).a = 'Anne';

s(1).b = 'James';
s(2).b = pi;

myfun.f1 = @mysqr;
myfun.f2 = @mysum;

s(1).c = myfun;
s(2).c = (1:7)';
```

OR . . .




```
myfun = struct('f1', @mysqr, 'f2', @mysum);

s = struct('a', {[1 4 7 2 9 3], 'Anne'},  . . .
           'b', {'James', pi},            . . .
           'c', {myfun, (1:7)'});
```

EXTRA

Showing videos from extracted frames

```
inputPath = 'Sequence';
figure;
for i = 133:183
    sprintf('%04d.jpeg', i)
    imshow(imread([inputPath '\' sprintf('%04d.jpeg', i)]));
    pause(.1);
end
```

- Ffmpeg is a good tool to extract frames from videos

Video. Other way

```
% - Read and display frames directly from AVI
video.
aviinfo('street.avi')
frame25 = aviread('street.avi', 25);
whos
frame25
imshow(frame25.cdata);
```