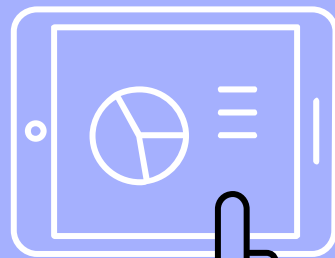
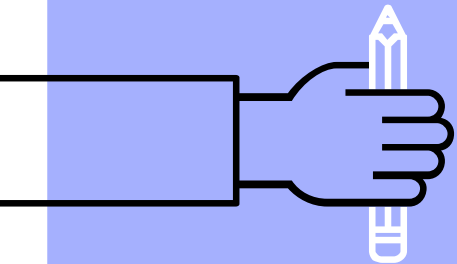


MATLAB Graphics Structure

Data and Image Display

盧家鋒 Chia-Feng Lu, Ph.D.
Department of Biomedical Imaging
and Radiological Sciences, NYCU
alvin4016@nycu.edu.tw



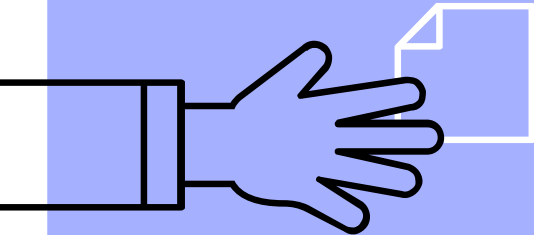
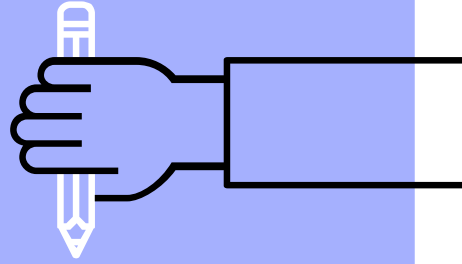
Contents

- ▷ Graphics structure and data display
- ▷ Image display and colormap

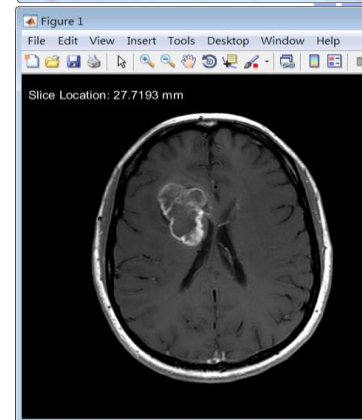
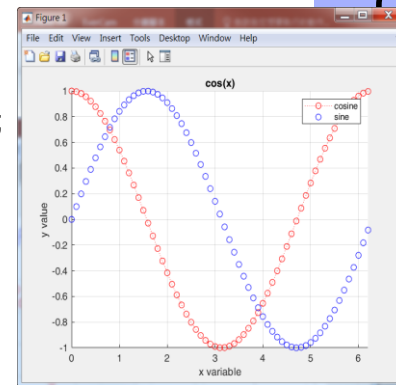
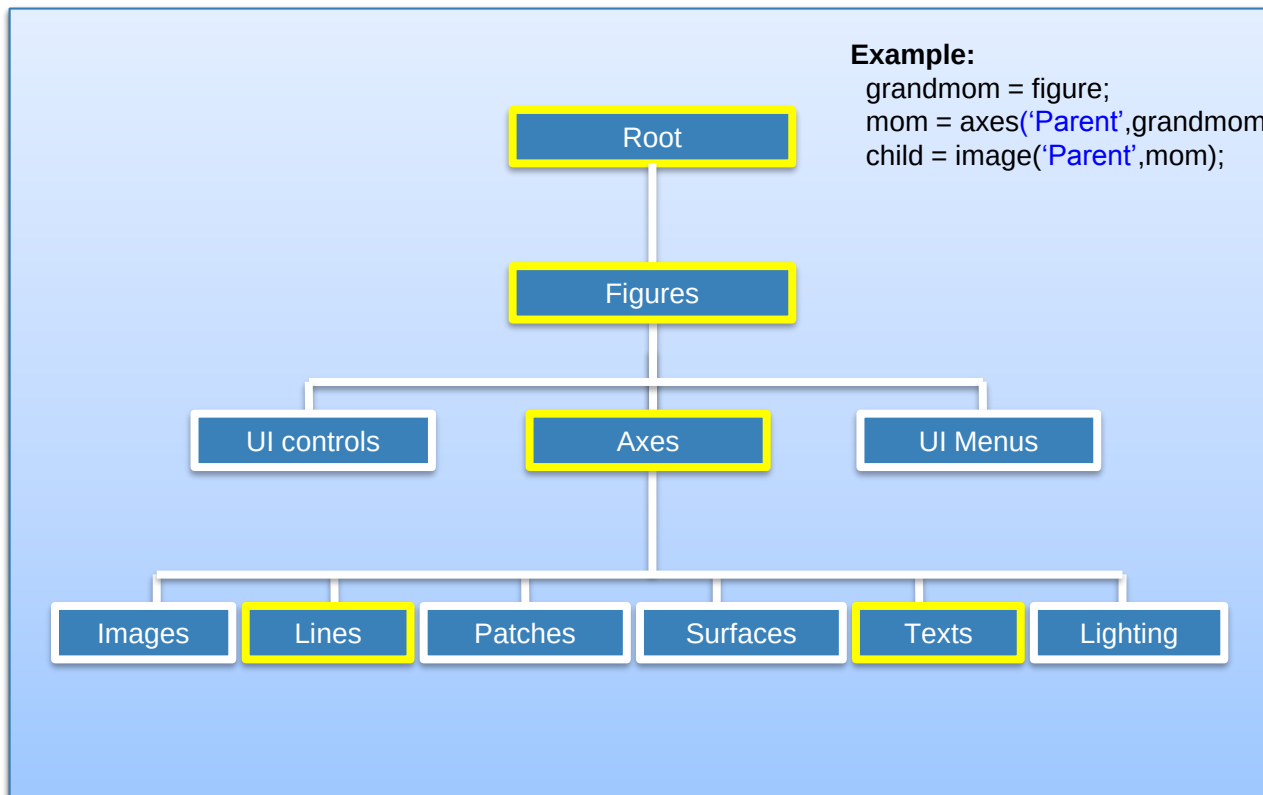
Please download the handout and materials from

http://cflu.lab.nycu.edu.tw/CFLu_course_matlabimage.html

Graphics structure and data display



Hierarchical Relations of Objects



Plot Your First Line!

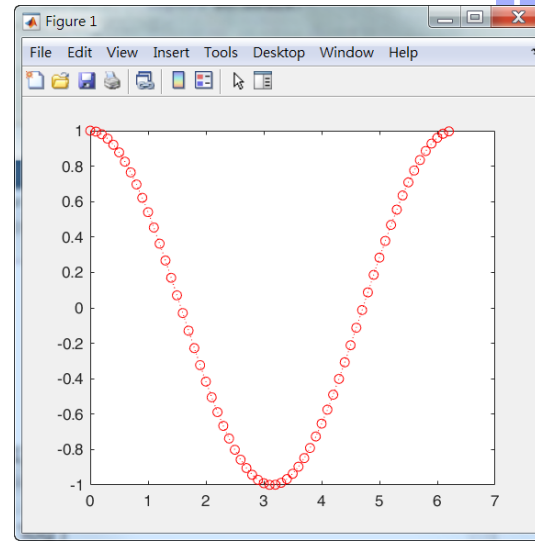
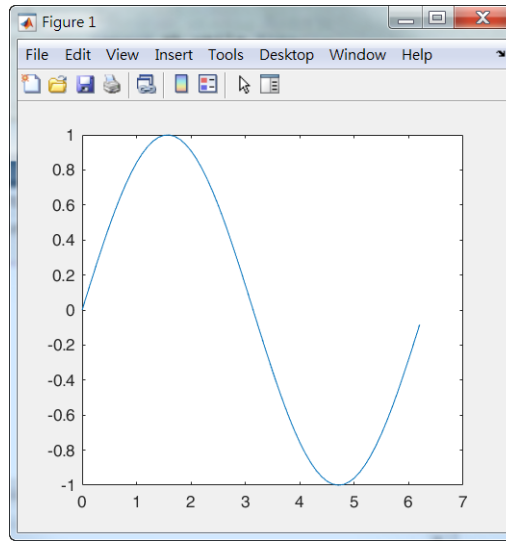
```
x=[0:0.1:2*pi];
```

```
figure,
```

```
plot(x,sin(x))
```

```
plot(x,cos(x),'ro:')
```

help plot

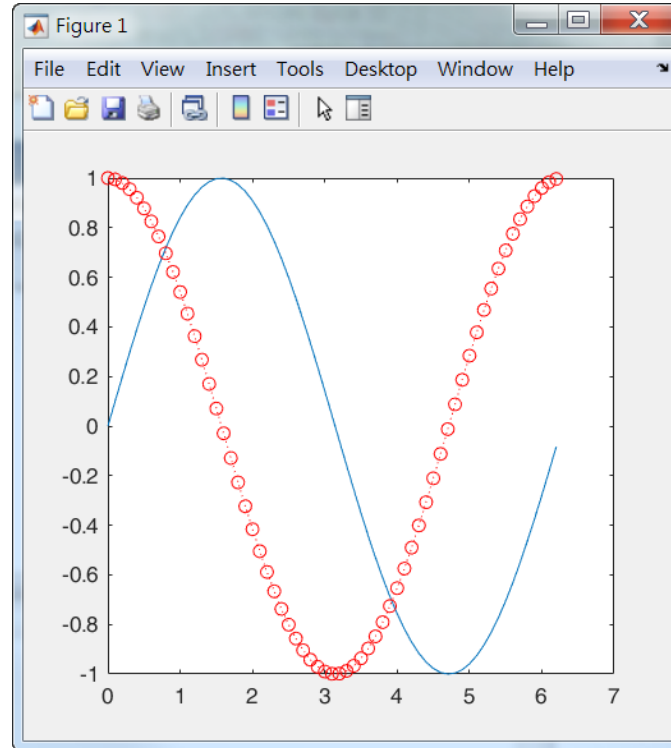


Plot Two Lines Together

```
x=[0:0.1:2*pi];  
figure,  
plot(x,sin(x))  
hold on  
plot(x,cos(x),'ro:')
```

HINT:

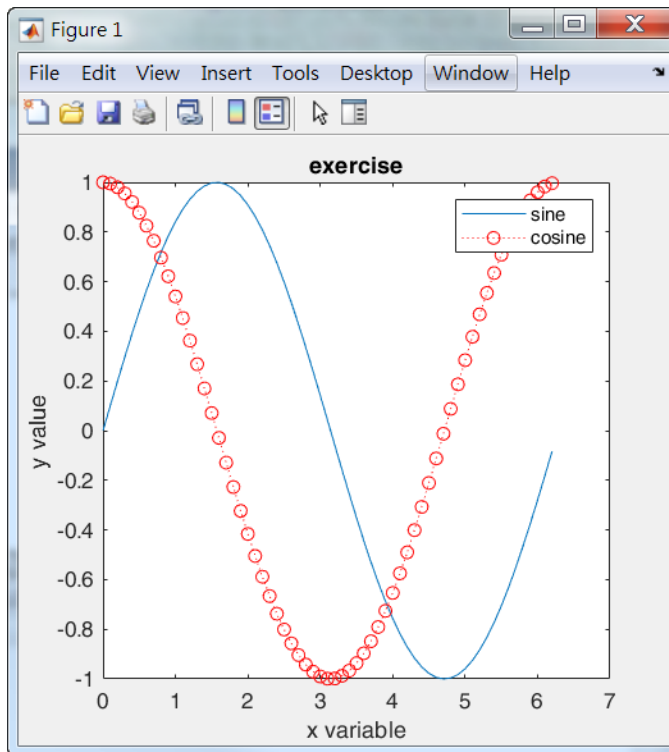
“**hold on**” allows subsequent graphing commands to add into the existing graph without resetting the figure.



Add Information to Your Plot

Following the last slide...

```
legend('sine','cosine')  
xlabel('x variable')  
ylabel('y value')  
title('exercise')
```

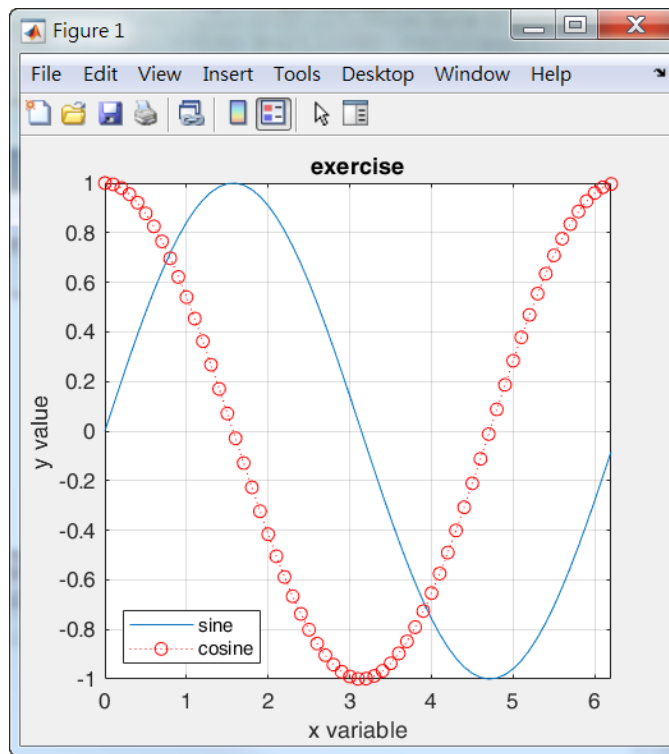


Control Axes Properties

Following the last slide...

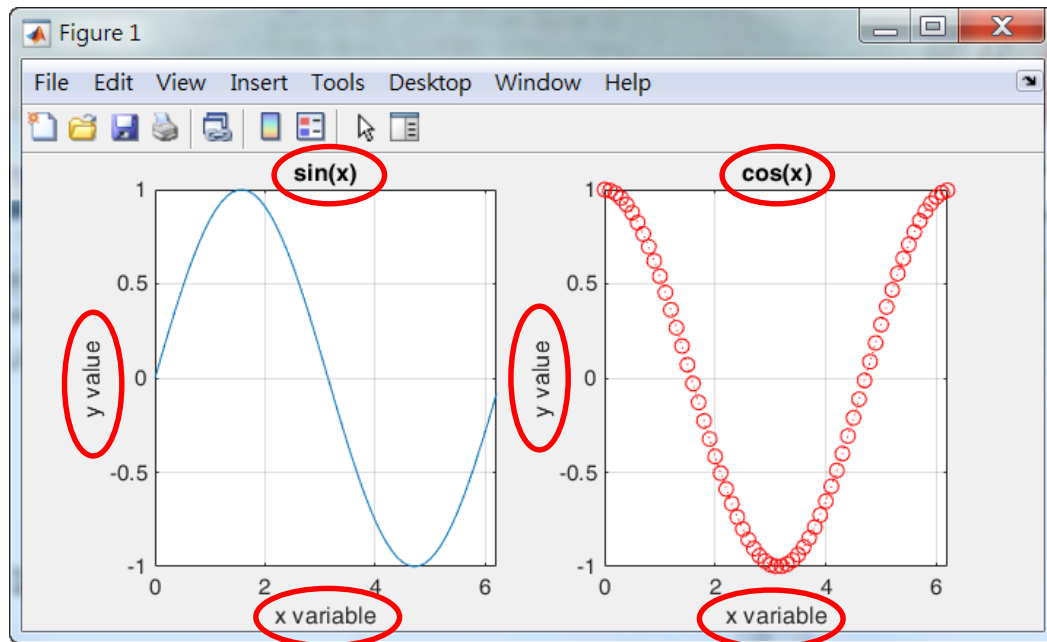
grid on,

`axis([min(x) max(x) -1 1])`



Plot Two Lines Separately

```
x=[0:0.1:2*pi];  
figure,  
subplot(1,2,1),  
plot(x,sin(x))  
subplot(1,2,2),  
plot(x,cos(x),'ro:')
```



Plot vs. Line

- ▶ Actually, **plot** is a kind of **line**!
- ▶ Understand the properties of **line** can help us to fully understand the usage of **plot**.

```
x=[0:0.1:2*pi];  
figure,  
plot(x,cos(x),'ro:');  
figure,  
line('xdata',x,'ydata',cos(x),'color','r','marker','o','linestyle',':');
```

Both commands print out the same results!!

Control Line Properties

<Commad Window>

set(line)

- ▶ Parent
- ▶ Xdata, Ydata, Zdata
- ▶ color
- ▶ Linestyle, Linewidth
- ▶ Marker, MarkerSize
- ▶ MarkerEdgeColor, MarkerFaceColor

```
Color
EraseMode: [ {normal} | background | xor | none ]
LineStyle: [ {-} | -- | : | -. | none ]
LineWidth
Marker: [ + | o | * | . | x | square | diamond | v | ^ | > | < | pentag
MarkerSize
MarkerEdgeColor: [ none | {auto} ] -or- a ColorSpec.
MarkerFaceColor: [ {none} | auto ] -or- a ColorSpec.
XData
YData
ZData

ButtonDownFcn: string -or- function handle -or- cell array
Children
Clipping: [ {on} | off ]
CreateFcn: string -or- function handle -or- cell array
DeleteFcn: string -or- function handle -or- cell array
BusyAction: [ {queue} | cancel ]
HandleVisibility: [ {on} | callback | off ]
HitTest: [ {on} | off ]
Interruptible: [ {on} | off ]
Parent
Selected: [ on | off ]
SelectionHighlight: [ {on} | off ]
Tag
UIContextMenu
UserData
Visible: [ {on} | off ]
```

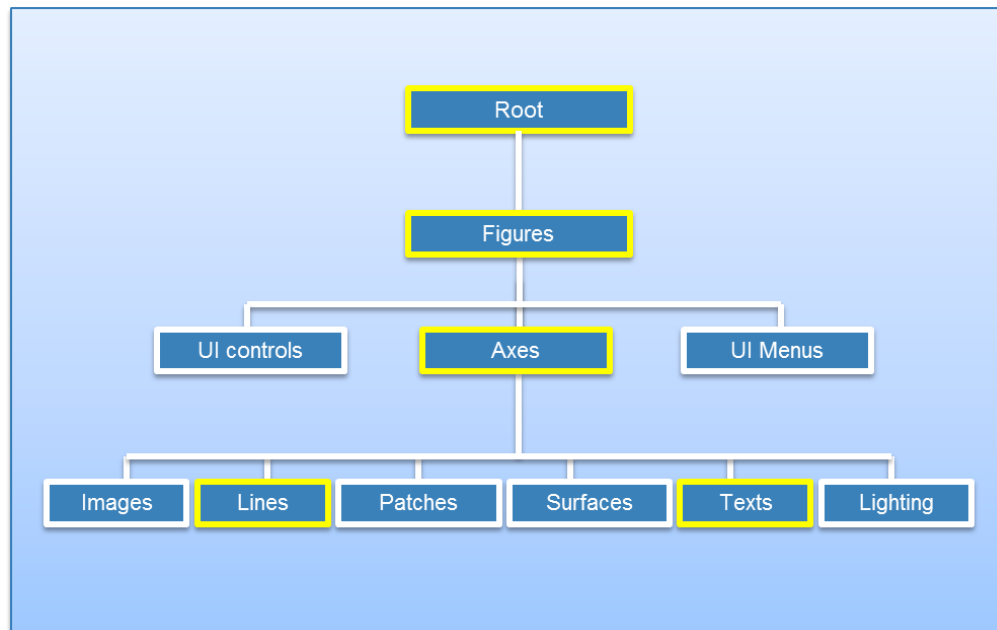
Control Line Properties: **handle**

▷ Please open MImaterials_L11\DemoLine.m

▷ set(figure)

▷ set(axes)

▷ set(line)

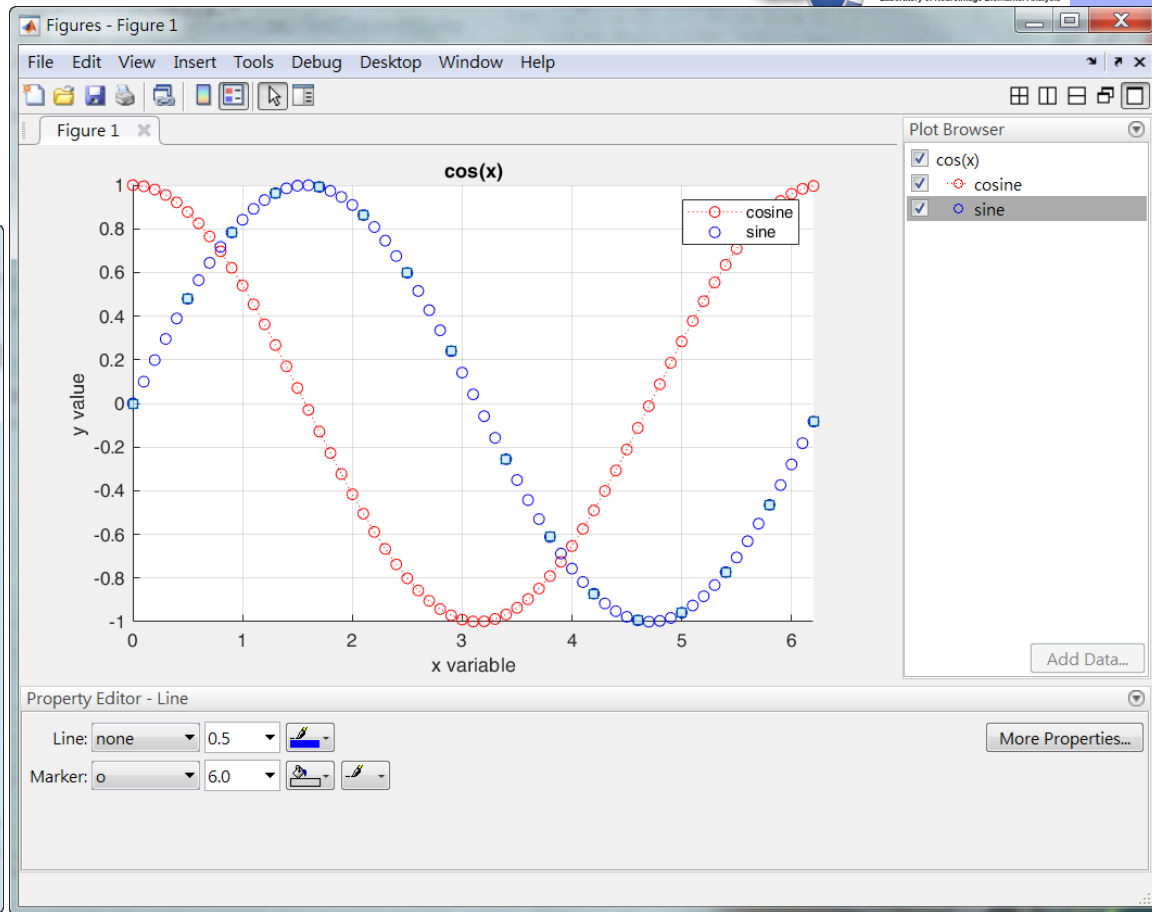
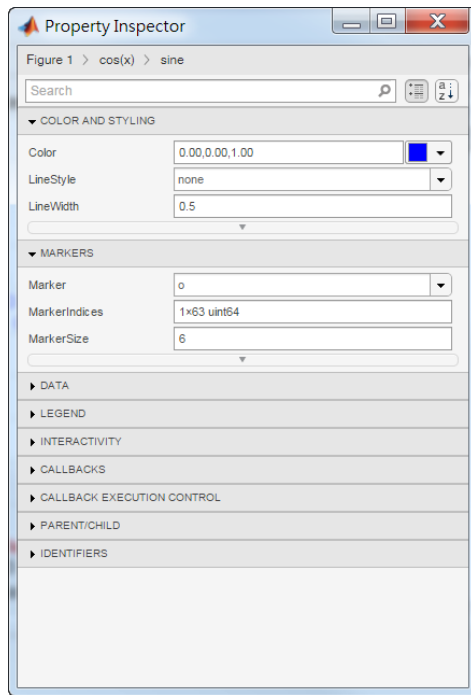


Learn the usage of set

```
5 - x=[0:0.1:2*pi];
6
7 - handle.fig1=figure;
8 - handle.axes1=axes('parent',handle.fig1);
9 - handle.line1=line('parent',handle.axes1);
10 - handle.line2=line('parent',handle.axes1);
11
12 - set(handle.line1,'xdata',x,'ydata',cos(x),'color','r','marker','o','linestyle',':');
13 - set(handle.line2,'xdata',x,'ydata',sin(x),'color','b','marker','o','linestyle','none');
14
15 - legend('cosine','sine')
16 - xlabel('x variable')
17 - ylabel('y value')
18 - title('cos(x)')
19 - grid on,
20 - axis([min(x) max(x) -1 1])
```

propedit

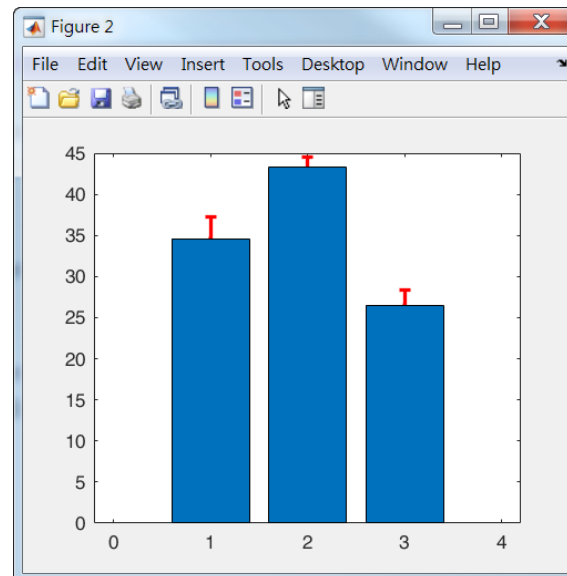
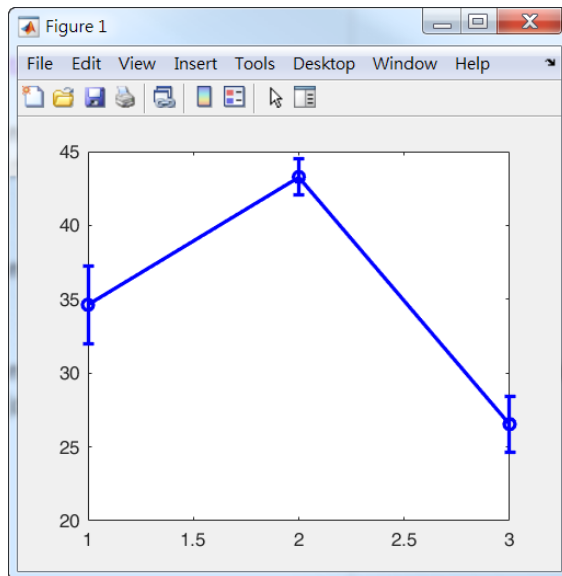
► Property editor



Data variation - errorbar

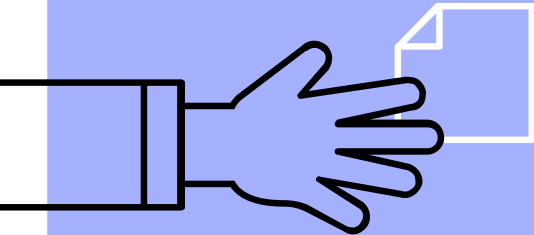
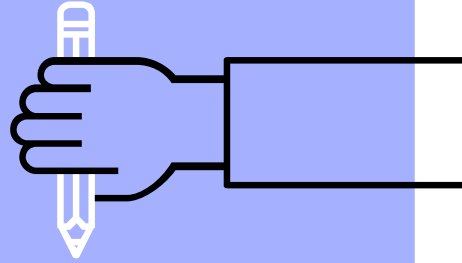
Useful function

- ▷ errorbar
- ▷ bar

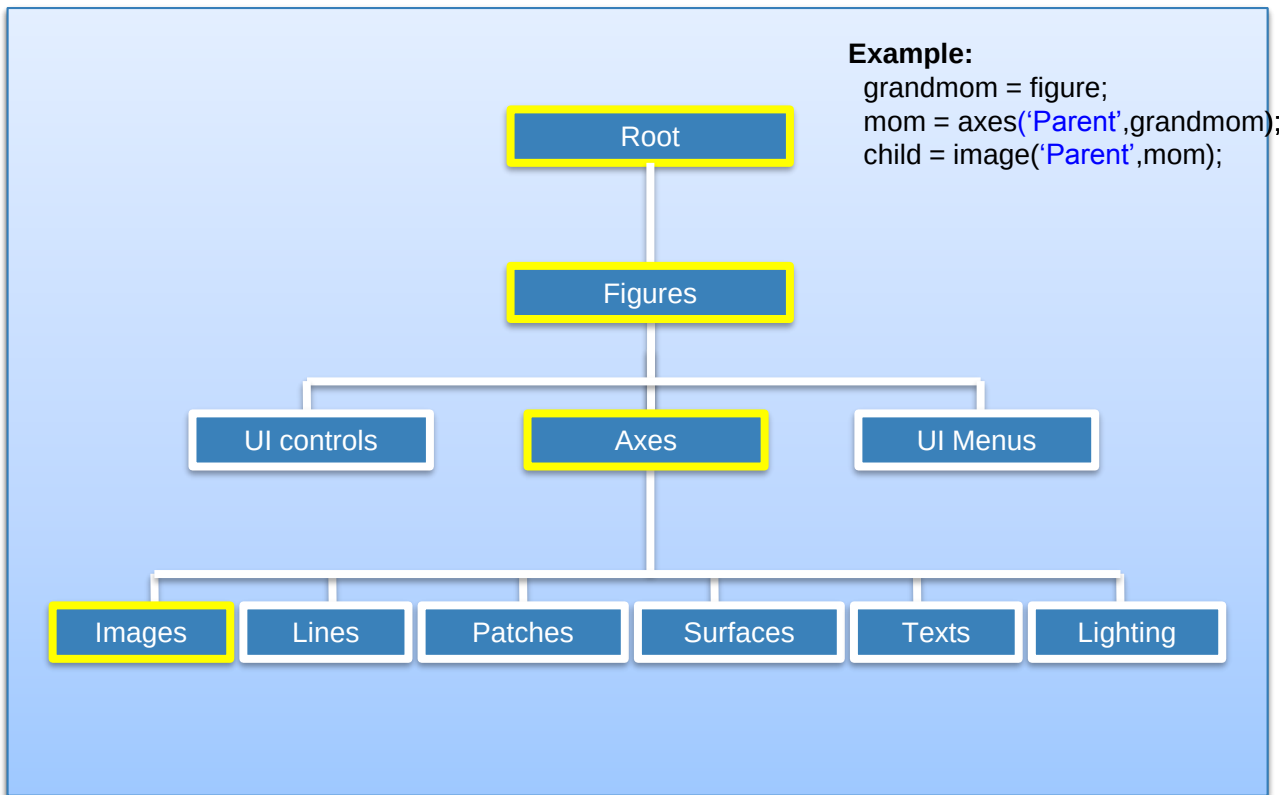


Please open and run MImaterials_L11\DemoErrorBar.m

Image display and colormap



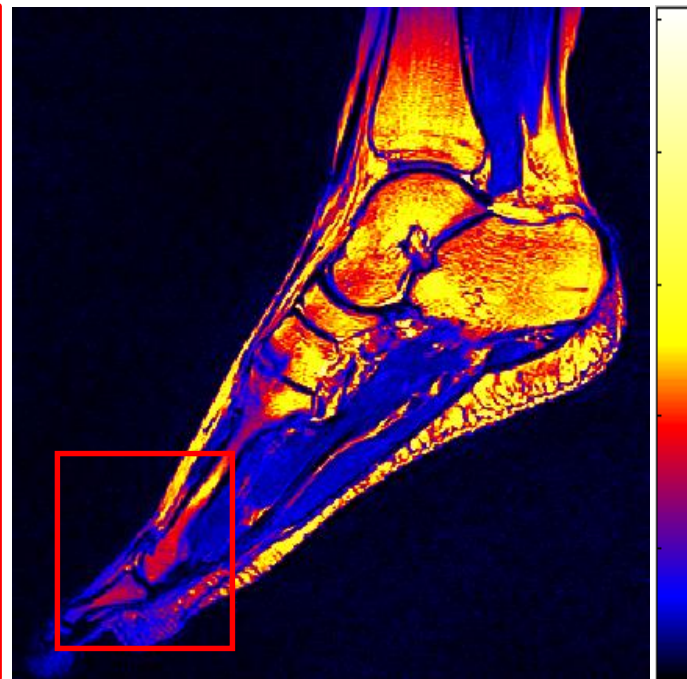
Hierarchical Relations of Objects



Concept of image intensity

▷ Pixel value and gray level

	3	4	5	6	7	8	9	10	11	12	13	14	15	16
13	17	20	18	19	18	17	19	19	19	20	20	19	15	
14	17	19	18	19	17	18	18	18	19	21	20	22	20	
15	17	17	17	19	18	18	20	20	19	22	23	22	116	
16	19	19	21	20	20	20	23	22	22	24	21	56	322	
17	18	21	19	20	19	21	20	22	24	27	24	223	429	
18	19	18	20	18	20	23	21	24	25	14	158	284	376	
19	17	17	18	18	19	20	21	21	22	60	331	214	183	
20	16	16	17	16	18	21	21	26	0	321	419	337	386	
21	17	16	15	17	20	22	22	17	83	582	242	329	654	
22	16	15	17	19	21	20	24	4	339	353	165	344	308	
23	15	17	18	19	19	21	10	135	495	130	262	144	164	
24	16	17	17	18	18	21	11	424	203	299	167	132	157	
25	17	17	18	18	20	3	189	444	300	332	153	150	116	
26	16	17	18	19	20	9	395	383	424	108	191	206	145	
27	18	16	18	17	15	142	370	347	139	95	197	252	126	
28	17	18	19	12	59	145	384	298	97	128	111	146	397	
29	19	19	15	85	205	324	429	178	91	142	201	513	507	
30	18	15	76	236	260	228	158	69	253	366	476	297	25	
31	21	70	180	295	287	136	176	331	326	219	95	2	21	
32	67	129	134	180	218	220	313	199	46	7	11	24	26	
33	76	57	67	84	162	202	176	47	12	20	20	21	21	
34	111	75	17	10	31	35	10	15	20	21	18	19	20	
35	62	26	16	16	13	13	17	20	18	18	18	18	24	



RGB Color Style

- ▶ Each color is composed of 3 numbers as a 1 x 3 vector

Color	RED	GREEN	BLUE
	1	1	1
	0	0	0
	1	0	0
	0	1	0
	0	0	1
	1	1	0

the value must between 0 and 1!!

- ▶ `uisetcolor` (can help you determine the color vector)

Intensity vs. Colormap

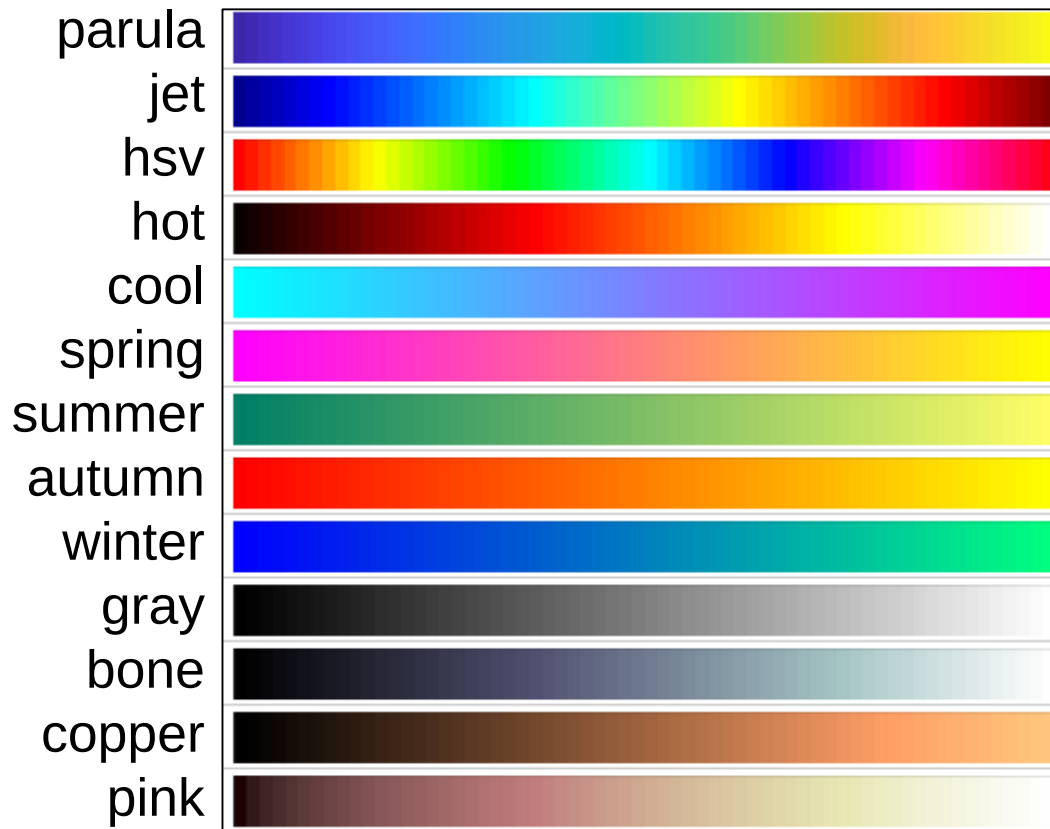
$$\text{cmap} = \begin{bmatrix} \text{R} & \text{G} & \text{B} \\ 0 & 0 & 0; \\ 1 & 0 & 0; \\ 0 & 1 & 0; \\ \vdots & \vdots & \vdots \\ 1 & 1 & 1 \end{bmatrix};$$

3
← corresponding color for intensity of 1
← corresponding color for intensity of 2
n
← corresponding color for intensity of n

Image Display Function

- ▷ **imshow**
 - Display image (with gray colormap in default).
 - Intensity between 0 and 1
- ▷ **image**
 - Display image (with parula colormap in default).
 - More suitable for color image

Predefined colormaps

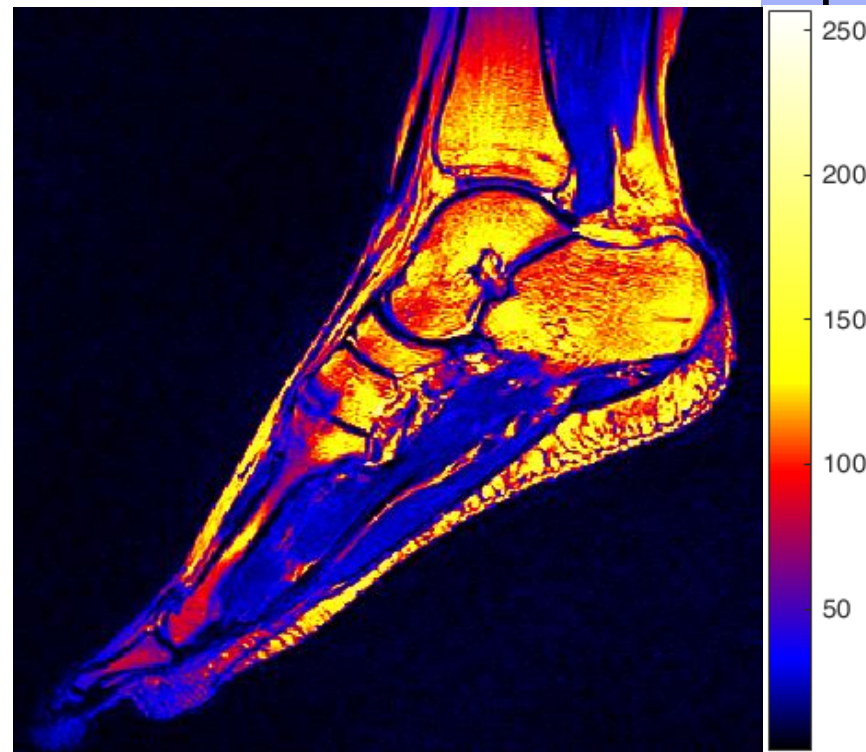
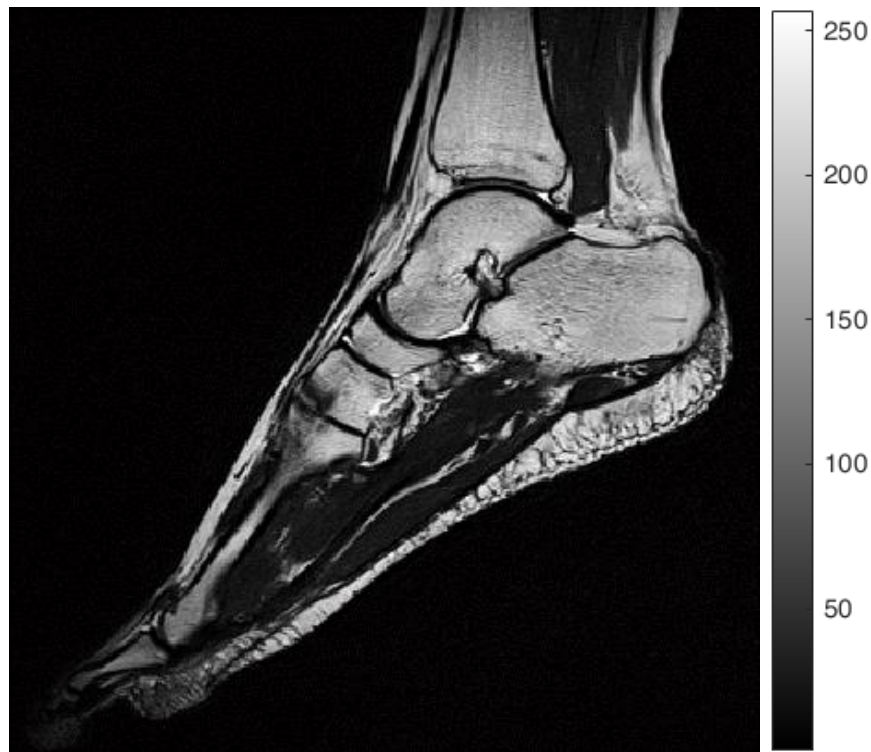


Intensity vs. Color

```
X=[4 5 6; 3 1 2];  
image(X);  
cmap=[1 0 0;0 1 0;0 0 1;1 1 0;0 1 1;1 0 1];  
colormap(cmap)
```



Customized colormap



Please open and run MImaterials_L11\ImageEx11.m

Customized colormap

```

6  %% load image
7  load('footimg.mat')
8
9  img=(img-min(img(:)))/(max(img(:))-min(img(:)));
10 img=img*255;
11
12 %% image colormap
13 figure,
14 image(img), axis off
15 colormap(gray(256)), colorbar
16
17 figure,
18 image(img), axis off
19 cmap=zeros(32,2) linspace(0,1,32)';... % black -> blue
20     linspace(0,1,64)' zeros(64,1) linspace(1,0,64)';... % blue -> red
21     ones(32,1) linspace(0,1,32)' zeros(32,1);... % red -> yellow
22     ones(128,2) linspace(0,1,128)']; % yellow -> white
23 colormap(cmap), colorbar
  
```

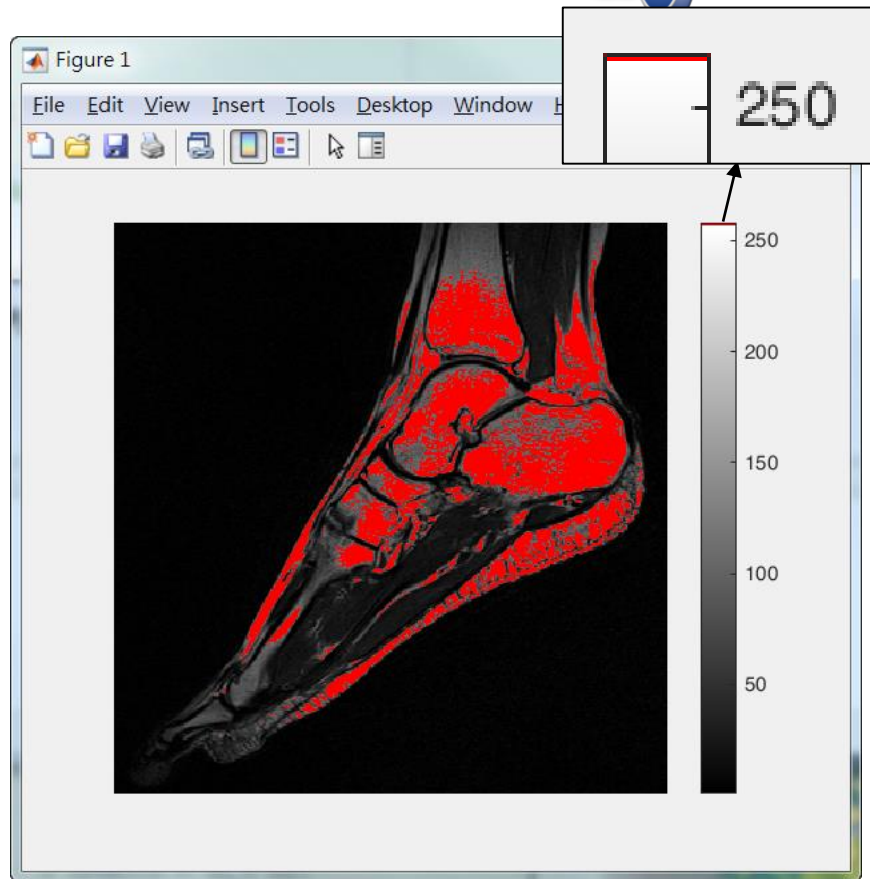
32 64 32 128
 Black → blue → red → yellow → white

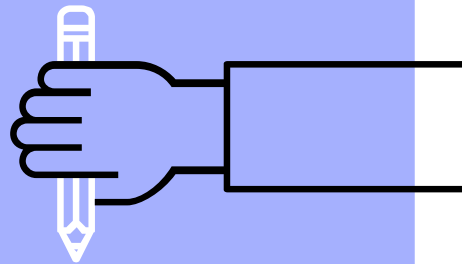


Homework...

- ▶ Please display the foot image with the red-labeled region for the high-intensity area (intensity > 600).
- ▶ 1. **find** the indices of pixels with intensity > 600.
- ▶ 2. rescale img intensity to 1~255. (overall 255 gray levels)
- ▶ 3. assign the intensity of indices (found in step 1) to 256.

`colormap([gray(255);1,0,0])`





THE END

alvin4016@nycu.edu.tw

