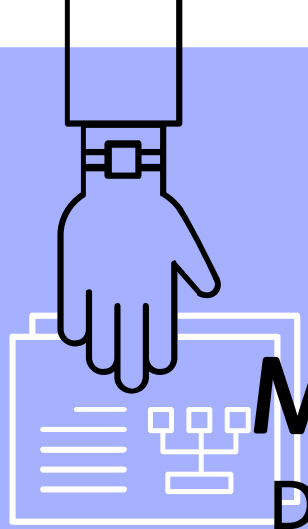
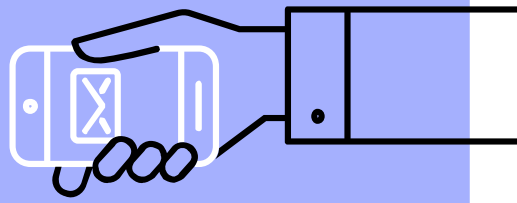
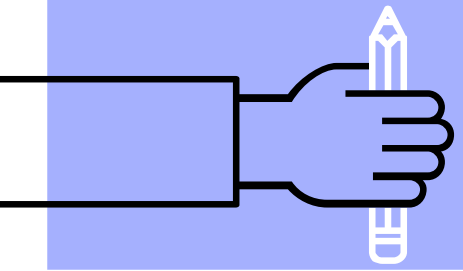


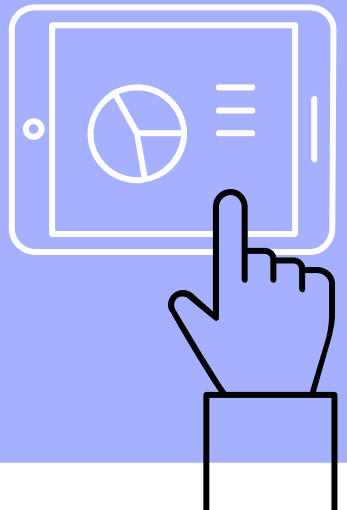


MATLAB Structure Array

DICOM Information and Image Contrast

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





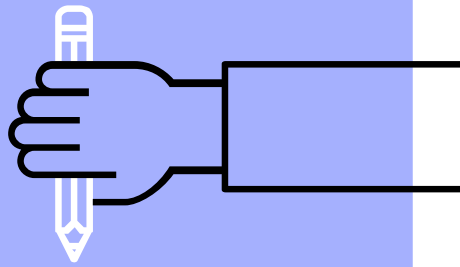
課程內容

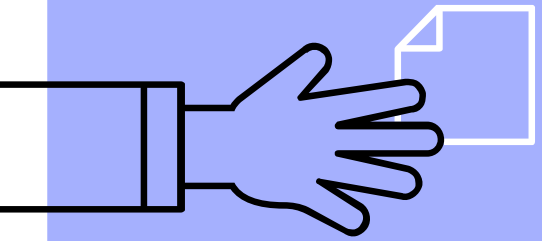
- Structure array
- Contrast adjustment (window center & window width)

Please download the handout and materials from

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

Structure Array





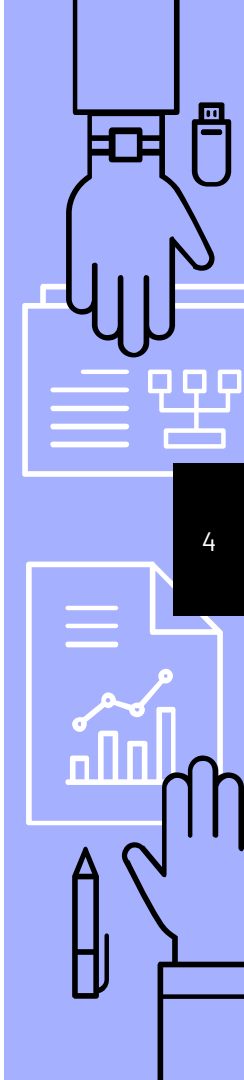
What can we store in an array?

- A=[1:2:30];
- A(9)

- A='This is a test!';
- A(9)

- A=['Alvin'; 'Lu'] % Does it work?
- A=['Alvin'; 'Lu '] % Does it work?

- Can we store numbers and strings in an array?
- A=[25000 'This is a test!'] % Does it work?
- A=[25000; 'This is a test!'] % Does it work?



Strings in an array

- List = ['David '; 'Andy '; 'Jay   '; 'Jolin '; 'Selina'];

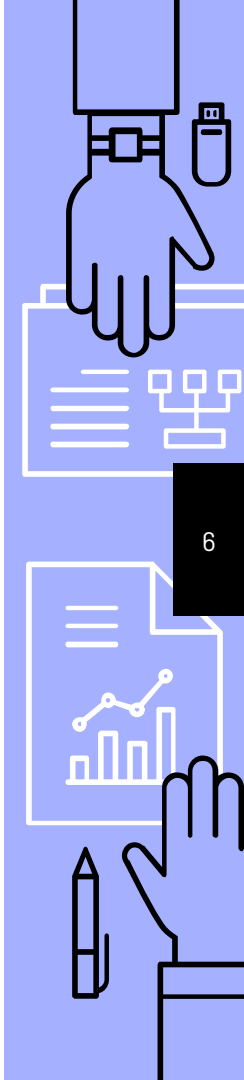
```
% List =
  David
  Andy
  Jay
  Jolin
  Selina
```

	c1	c2	c3	c4	c5	c6
r1	D	a	v	i	d	
r2	A	n	d	y		
r3	J	a	y			
r4	J	o	l	i	n	
r5	S	e	l	i	n	a

More Intuitive Way

When dealing with different length of strings…

- List(1).name = 'David';
- List(2).name = 'Andy';
- List(3).name = 'Jay';
- List(4).name = 'Jolin';
- List(5).name = 'Selina';



Archive Comprehensive Dataset

For example,

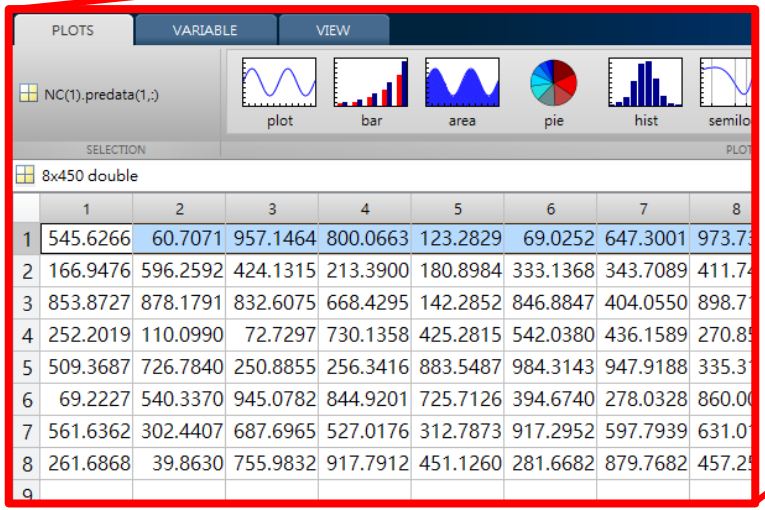
- 30 normal controls vs. 30 patients with stroke

- Name, ID → string
- Age, gender, height, weight → numbers
- Pre-training data (8 x 450 values) → number array
- Post-training data (8 x 450 values) → number array

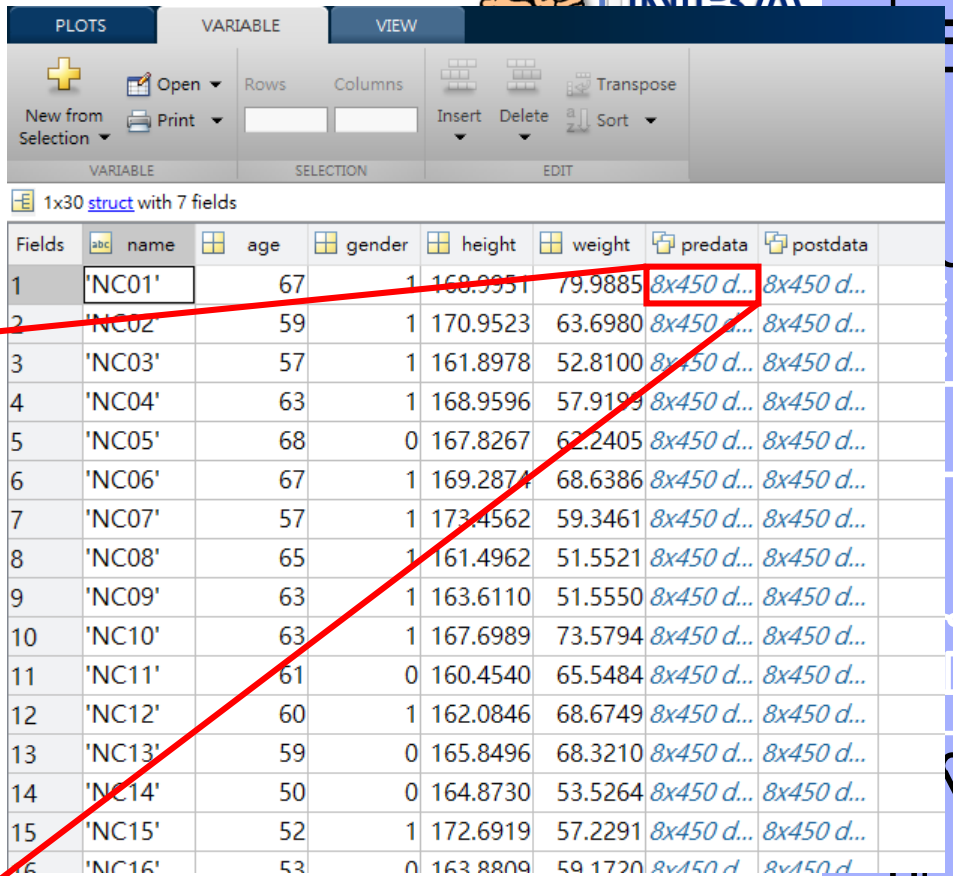
Try to import ExampleStruct.mat file.
(drag and drop to Workspace or use load)

Structure Array

- Store different data formats in a single structure array.
- Numbers, strings, arrays

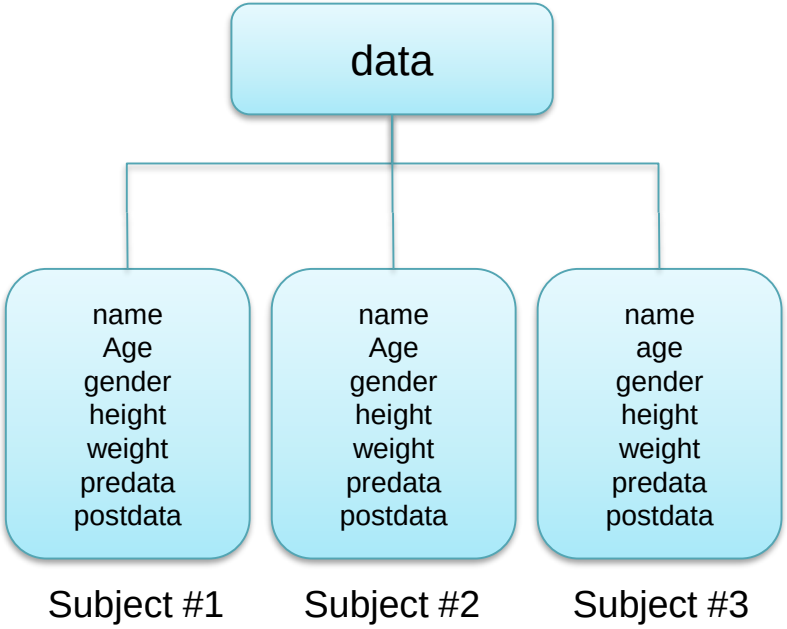


	1	2	3	4	5	6	7	8
1	545.6266	60.7071	957.1464	800.0663	123.2829	69.0252	647.3001	973.7
2	166.9476	596.2592	424.1315	213.3900	180.8984	333.1368	343.7089	411.7
3	853.8727	878.1791	832.6075	668.4295	142.2852	846.8847	404.0550	898.7
4	252.2019	110.0990	72.7297	730.1358	425.2815	542.0380	436.1589	270.8
5	509.3687	726.7840	250.8855	256.3416	883.5487	984.3143	947.9188	335.3
6	69.2227	540.3370	945.0782	844.9201	725.7126	394.6740	278.0328	860.0
7	561.6362	302.4407	687.6965	527.0176	312.7873	917.2952	597.7939	631.0
8	261.6868	39.8630	755.9832	917.7912	451.1260	281.6682	879.7682	457.2
9								



Fields	name	age	gender	height	weight	predata	postdata
1	'NC01'	67	1	168.9951	79.9885	8x450 d...	8x450 d...
2	'NC02'	59	1	170.9523	63.6980	8x450 d...	8x450 d...
3	'NC03'	57	1	161.8978	52.8100	8x450 d...	8x450 d...
4	'NC04'	63	1	168.9596	57.9199	8x450 d...	8x450 d...
5	'NC05'	68	0	167.8267	62.2405	8x450 d...	8x450 d...
6	'NC06'	67	1	169.2874	68.6386	8x450 d...	8x450 d...
7	'NC07'	57	1	173.4562	59.3461	8x450 d...	8x450 d...
8	'NC08'	65	1	161.4962	51.5521	8x450 d...	8x450 d...
9	'NC09'	63	1	163.6110	51.5550	8x450 d...	8x450 d...
10	'NC10'	63	1	167.6989	73.5794	8x450 d...	8x450 d...
11	'NC11'	61	0	160.4540	65.5484	8x450 d...	8x450 d...
12	'NC12'	60	1	162.0846	68.6749	8x450 d...	8x450 d...
13	'NC13'	59	0	165.8496	68.3210	8x450 d...	8x450 d...
14	'NC14'	50	0	164.8730	53.5264	8x450 d...	8x450 d...
15	'NC15'	52	1	172.6919	57.2291	8x450 d...	8x450 d...
16	'NC16'	53	0	163.8809	59.1720	8x450 d...	8x450 d...

Create a Structure Array



a 1x3 structure array

7 **field**s for each subject

Create a Structure array

- data.name='NC01'; % string
- data.age=62; % number
- data.predata=rand(8,450); % number array

1x1 struct with 4 fields

Field	Value	Min	Max
name	'NC01'		
age	62	62	62
predata	*8x450 double*	1.3...	0.9...

Use variable viewer to check the data structure!

Create a Structure array

```
data(1).name='NC01';          % string
data(1).age=62;               % number
data(1).predata=rand(8,450);  % number array
data(2).name='NC02';          % string
data(2).age=54;               % number
data(2).predata=rand(8,450);  % number array
```

Use variable viewer to check the data structure!

Another Way to Create a Structure array

- **struct**
  - Create or convert to structure array
  - S = struct('field1',VALUES1,'field2',VALUES2,...)

- data(3)=struct('name','NC03','age',83,'predata',rand(8,450));

- size(data)

```
ans =
     1     3
```

Index of a structure array

- Use both index and field name to specify data location.

1x2 struct with 3 fields

Fields	name	age	predata
1	'NC01'	62	8x450 d...
2	'NC02'	54	8x450 d...

8x450 double

	1	2	3	4	5	6
1	0.2203	0.3261	0.6257	0.4105	0.6778	0.5
2	0.8901	0.8255	0.2631	0.9692	0.4459	0.7
3	0.3016	0.5502	0.9756	0.6532	0.4971	0.5
4	0.2501	0.6448	0.1614	0.7714	0.1561	0.6
5	0.8966	0.8451	0.7945	0.4152	0.6979	0.4
6	0.1197	0.5950	0.9244	0.1491	0.2047	0.2
7	0.6652	0.3310	0.4663	0.4943	0.6774	0.9
8	0.3275	0.2114	0.6136	0.8487	0.1245	0.9

Navigation in variable viewer…

data(1).predata(6,3)

Index & Matrix Operations

1x2 struct with 3 fields

Fields	name	age	predata
1	'NC01'	62	8x450 d...
2	'NC02'	54	8x450 d...

8x450 double

	1	2	3	4	5	6
1	0.9960	0.1817	0.7442	0.0170	0.2321	0.7
2	0.7362	0.3838	0.9294	0.5174	0.7982	0.7
3	0.0575	0.8371	0.1284	0.5123	0.2378	0.9
4	0.0093	0.7617	0.2210	0.3960	0.5733	0.7
5	0.8309	0.9872	0.3280	0.9207	0.6194	0.5
6	0.3350	0.7110	0.3381	0.9918	0.4537	0.5
7	0.2690	0.3448	0.8673	0.5547	0.7544	0.5
8	0.1083	0.5376	0.6757	0.5708	0.5659	0.9

8x450 double

	1	2	3	4	5	6
1	0.2203	0.3261	0.6257	0.4105	0.6778	0
2	0.8901	0.8255	0.2631	0.9692	0.4459	0
3	0.3016	0.5502	0.9756	0.6532	0.4971	0
4	0.2501	0.6448	0.1614	0.7714	0.1561	0
5	0.8966	0.8451	0.7945	0.4152	0.6979	0
6	0.1197	0.5950	0.9244	0.1491	0.2047	0
7	0.6652	0.3310	0.4663	0.4943	0.6774	0
8	0.3275	0.2114	0.6136	0.8487	0.1245	0

data(1).predata(2:5,3:4)+data(2).predata(4:7,4:5)

data(1).predata(2:5,3:4)-data(2).predata(4:7,4:5)

data(1).predata(2:5,3:4).*data(2).predata(4:7,4:5)

data(1).predata(2:5,3:4)./data(2).predata(4:7,4:5)

Useful Functions

Try it!

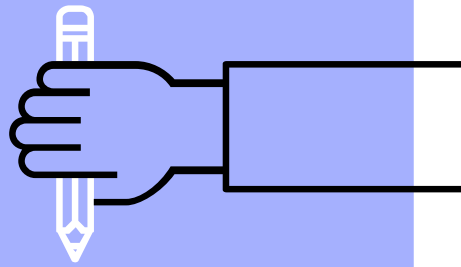
- **isstruct**
  - True for structures
- **fields**
  - Display a list of fields in a structure array
- **isfield**
  - True if field is in structure array
- **rmfield**
  - Remove fields from a structure array

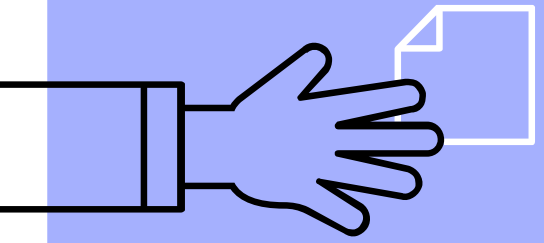
Benefits for Using a Single Array

- Easy to clear, save, and load

- Easy to categorize variables
  - Data % store all dataset-related information
  - handle % store all GUI object handles
  - file % store all file-related information

Contrast Adjustment





Step 1 – Import DICOM Header and Image

ImageEx05.m

```
1   %% Adjust Contrast
2   clear, close all
3
4   % get the header data of a DICOM file
5   info=dicominfo('LungCT.dcm');
6
7   % get the image data of a DICOM file
8   img=dicomread('LungCT.dcm');
```

info

1x1 struct with 125 fields

Field	Value
Filename	'D:\CloudStation\YM\107
FileModDate	'18-十月-2018 07:26:56'
FileSize	527738
Format	'DICOM'
FormatVersion	3
Width	512
Height	512
BitDepth	12
ColorType	'grayscale'

fields(info)

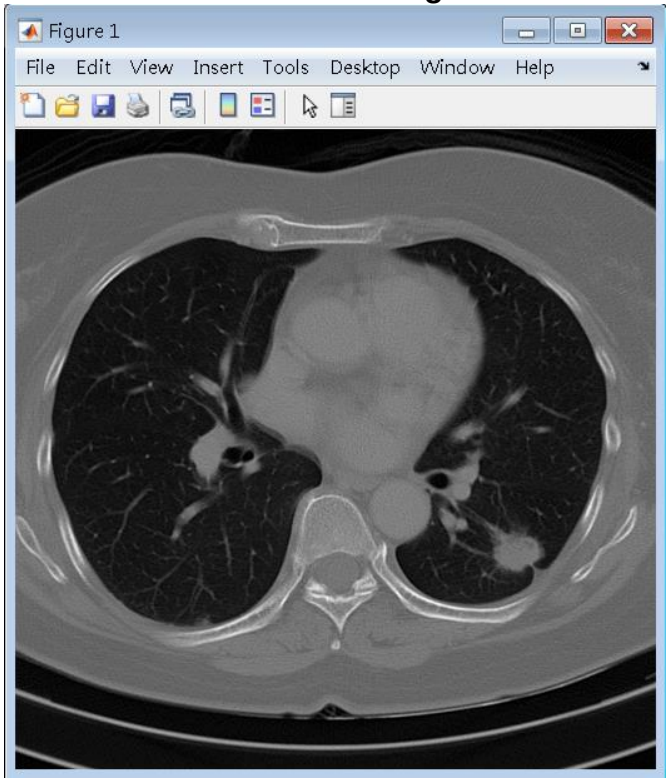
Step 2 – Display Image with Full-range Contrast

ImageEx05.m

```
1   %% Adjust Contrast
2   clear, close all
3
4   % get the header data of a DICOM file
5   info=dicominfo('LungCT.dcm');
6
7   % get the image data of a DICOM file
8   img=dicomread('LungCT.dcm');
9
10  figure, imshow(img,[],'border','tight')
```

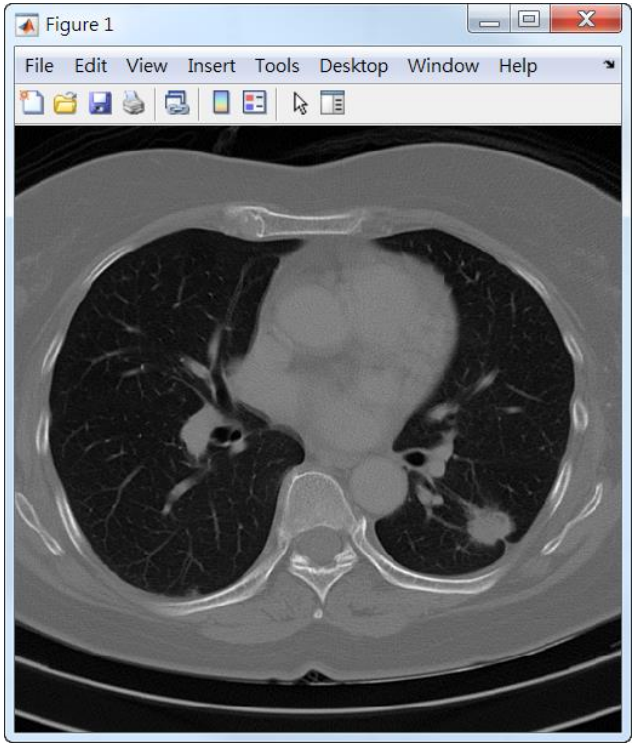
imshow(img,[min(img(:)) max(img(:))],'border','tight')

Chest CT image

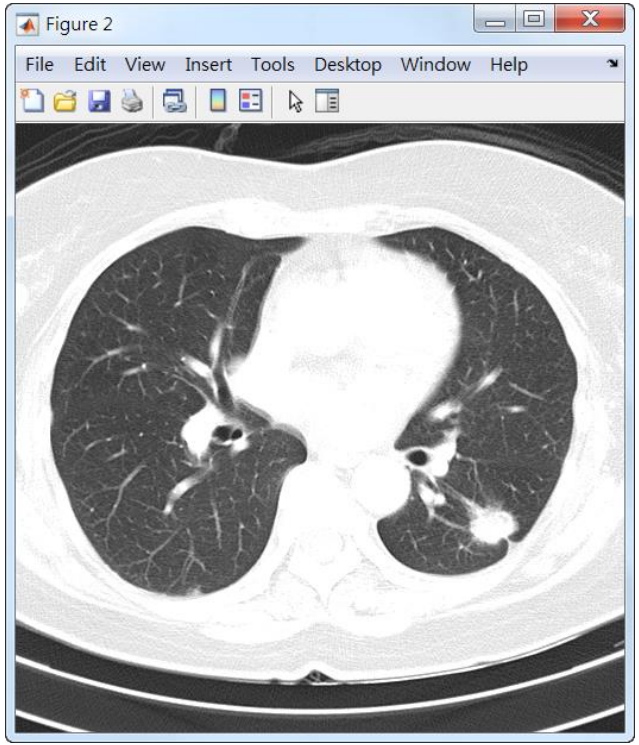


Step 2 – Display Image with Preset Contrast

Full-range window
(Full-dynamic window)



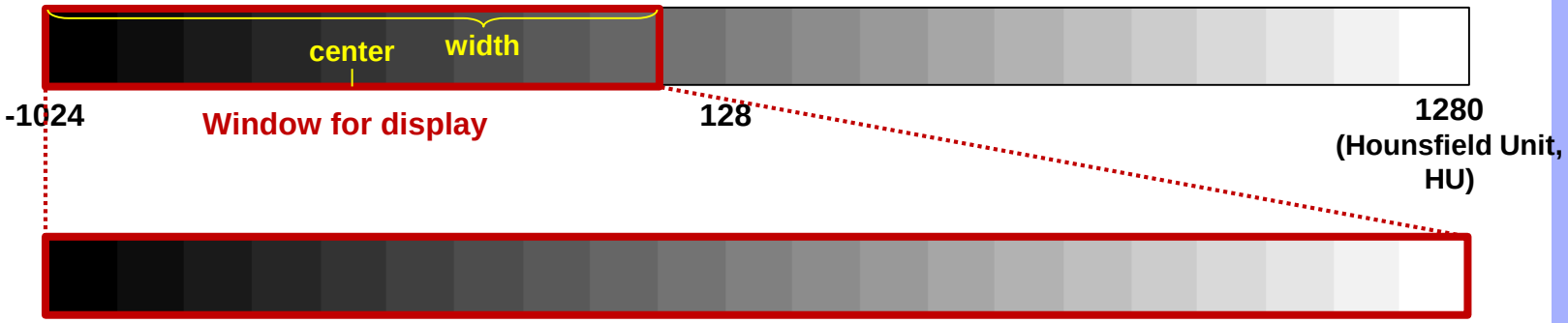
Preset window
(lung window)



Step 2 – Display Image with Preset Contrast

```
% get the image data of a DICOM file
img=dicomread('LungCT.dcm');
```


```
% rearange image intensity
img=double(img)*info.RescaleSlope+info.RescaleIntercept;
```

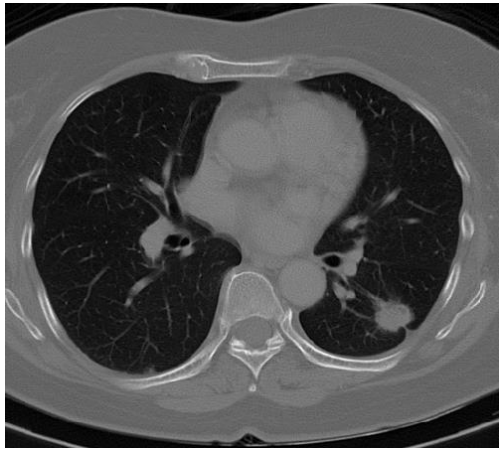


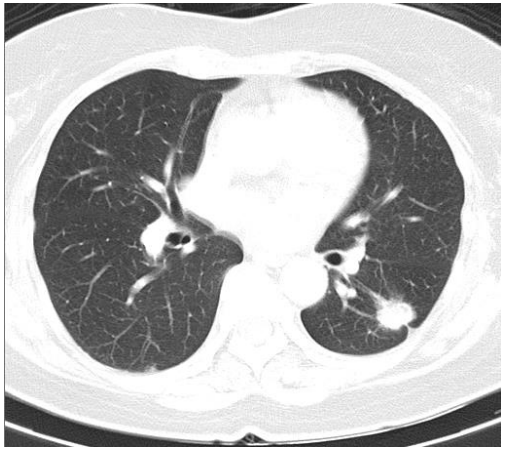
Step 2 – Display Image with Preset Contrast

```
12      %% display image with preset window (lung window)
13      % rearange image intensity
14 -    img=double(img)*info.RescaleSlope+info.RescaleIntercept;
15
16      % determine window center and width
17 -    WC=info.WindowCenter(1); %-600
18 -    WW=info.WindowWidth(1); %1200
19 -    figure, imshow(img,[WC-WW/2,WC+WW/2],'border','tight')
```

Step 3 – Other Windows

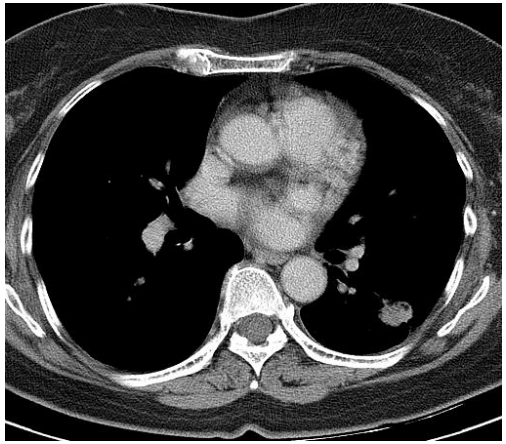
Full-range window
(Full-dynamic window)

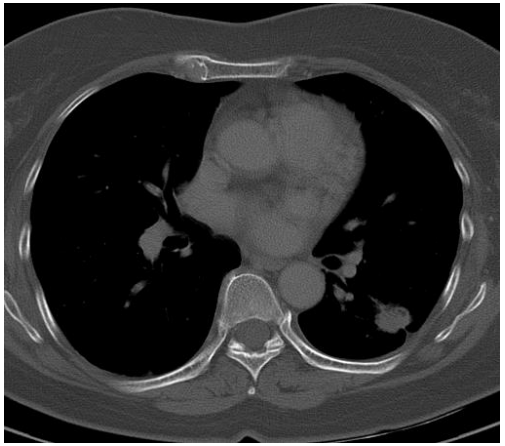




Lung Window
(-600/1200)

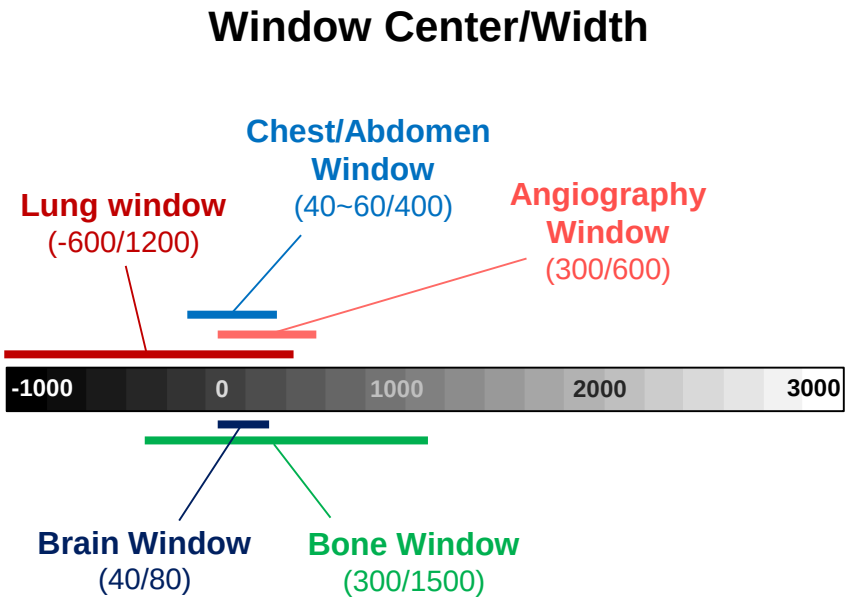
Chest Soft Tissue Window
(40/400)





Bone Window
(300/1500)

Step 3 – Display Image with Other Windows



Substance	Hounsfield Unit (HU)
Air	-1000
Lung	-700
Soft tissue	-300 to -100
Fat	-100 to -50
Water	0
Cerebrospinal fluid (CSF)	15
Blood	+30 to +45
Muscle	+40
Bone	+700 (cancellous bone) to +3000 (dense bone)

Step 3 – Display Image with Other Windows

```
21      %% display image with soft tissue window
22 -    WC=40;
23 -    WW=400;
24 -    figure, imshow(img,[WC-WW/2,WC+WW/2],'border','tight')
25
26      %% display image with bone window
27 -    WC=300;
28 -    WW=1500;
29 -    figure, imshow(img,[WC-WW/2,WC+WW/2],'border','tight')
30
```

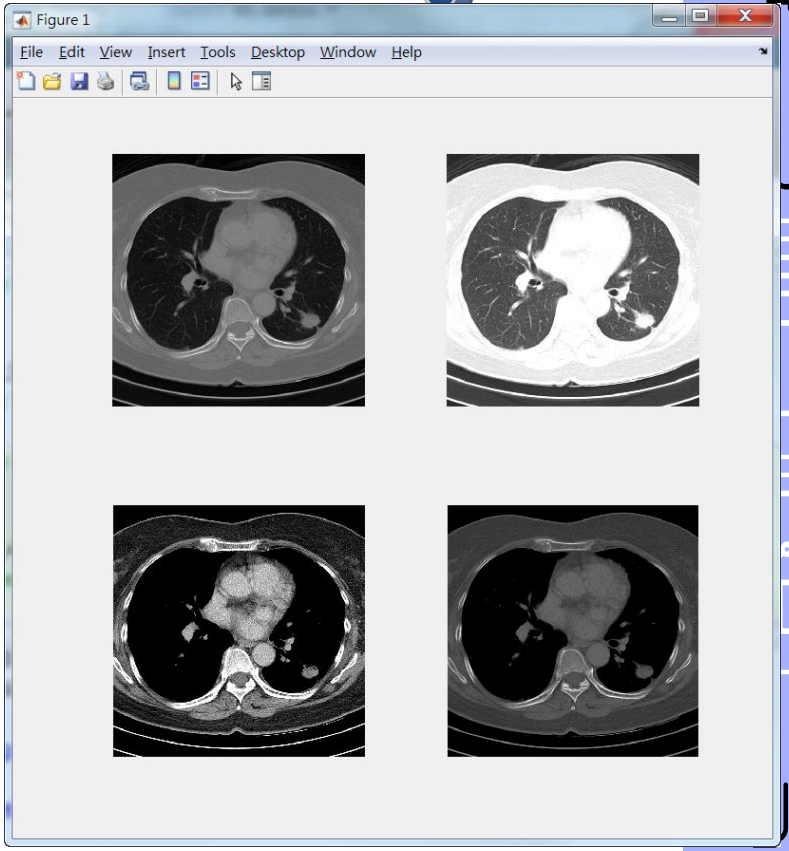
Homework

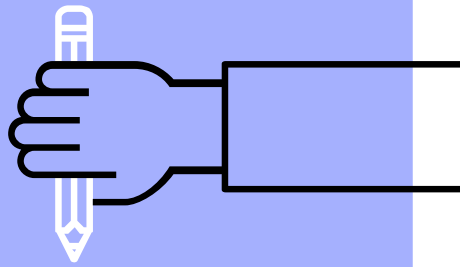
- Try to display all images with different window center/width in one figure.

- **subplot** Create axes in tiled positions.

  subplot(m,n,p), or subplot(mnp), breaks the Figure window into an m-by-n matrix of small axes, selects the p-th axes for the current plot, and returns the axes handle. The axes are counted along the top row of the Figure window, then the second row, etc.

  - Modify from **ImageEx05.m**
  - Replace **figure** by **subplot.**





THE END

alvin4016@nycu.edu.tw