MATLAB
結構矩陣與異值矩陣

盧家鋒 助理教授
alvin4016@ym.edu.tw

請先下載本週上課資料

- http://www.ym.edu.tw/~cflu
- 點選左欄 [課程資料] → [MATLAB圖形使用者介面]
- 下載第4週 [上課資料] materials_L4.zip，檔案大小約6.4MB

- 請先將Current Directory切換至materials_L4資料夾!

CF

本週內容

- 異質矩陣 (cell array)
- 結構矩陣 (structure array)

CF

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?

CF

Strings in an array

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

% List =

David

Andy

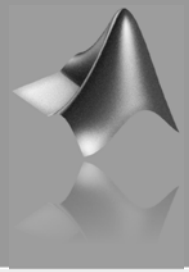
Jay

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

CF



異質矩陣
cell array

CF

Archive Big Dataset

For example,

- 30 normal controls vs. 30 patients with stroke

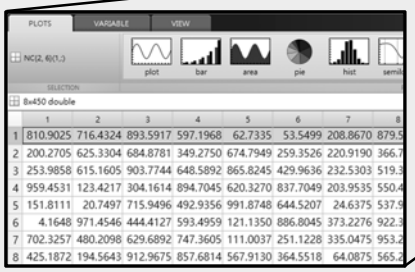
- Name, ID → string
- Age, gender, height, weight → numbers
- Pre-training data (8 x 450 values) → number array
- Post-training data (8 x 450 values) → number array

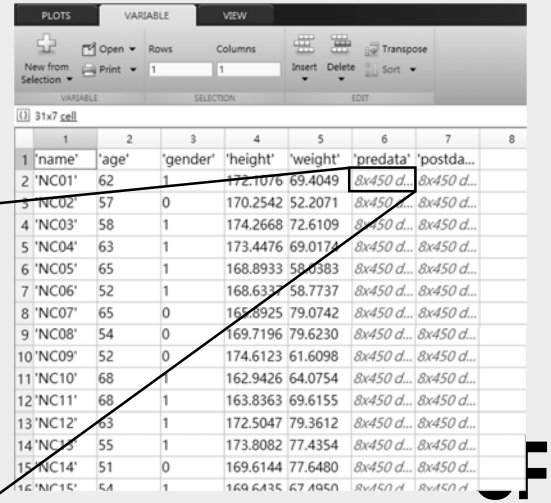
試試看在command window中鍵入
load('ExampleCell.mat')

CF

Cell Array

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





CF

Create a Cell array

- data={ };
- data=cell(4,1);

- data(1)={'NC01'}; % string
- data(2)={62}; % number
- data(3)={rand(8,450)}; % number array
- data(4)={{ 16, '%task onset'}}; % cell array

Use variable viewer to check the data structure!

CF

Create a Cell array – Different Ways

- data=cell(4,1);
  data(1)={'NC01'}; % string
  data(2)={62}; % number
  data(3)={rand(8,450)}; % number array
  data(4)={{ 16, '%task onset'}}; % cell array
- data2=cell(4,1);
  data2{1}='NC01'; % string
  data2{2}=62; % number
  data2{3}=rand(8,450); % number array
  data2{4}={ 16, '%task onset'}; % cell array

CF

More Intuitive Way

When dealing with different length of strings…

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

- List = {'David'; 'Andy'; 'Jay'; 'Jolin'; 'Selina'};

CF

Index of a cell array

- List = {'David'; 'Andy'; 'Jay'; 'Jolin'; 'Selina'};

- Is there any difference between following two commands?
- List(1) → **cell array**
- List{1} → **string**

CF

Useful Functions

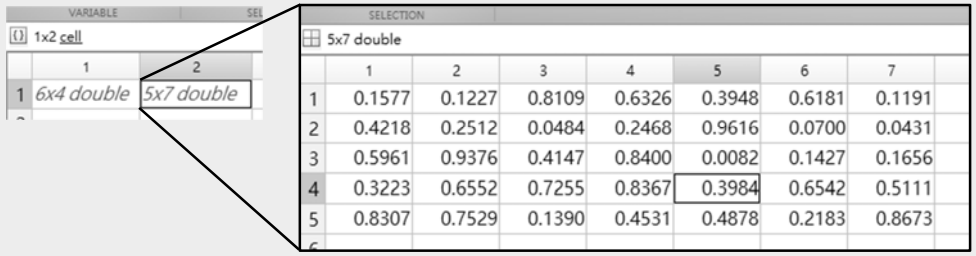
- **cell**
  - Create cell array
- **iscell**
  - True for cell array

- Try it…
  - iscell(List(1))
  - iscell(List{1})

CF

Index of a cell array

- A={rand(6,4),rand(5,7)};

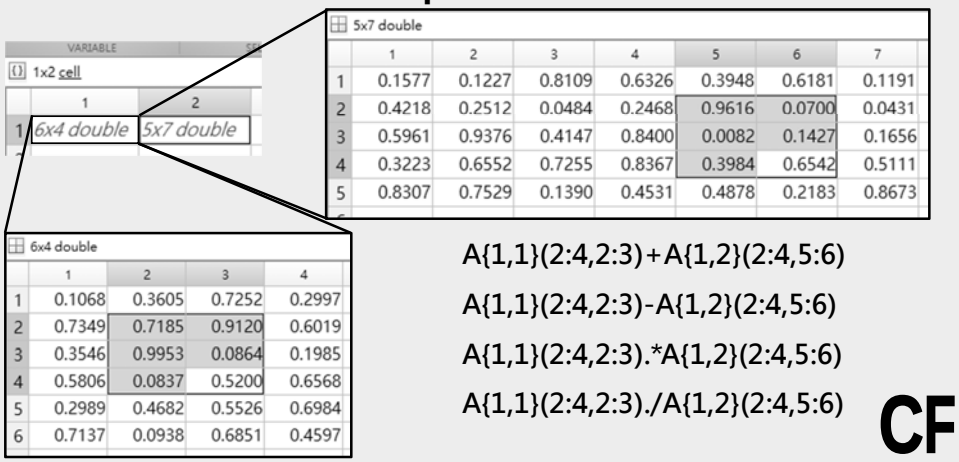


A{1,2}(4,5)

Navigation in variable viewer…

CF

Index & Matrix Operations



A{1,1}(2:4,2:3)+A{1,2}(2:4,5:6)

A{1,1}(2:4,2:3)-A{1,2}(2:4,5:6)

A{1,1}(2:4,2:3).*A{1,2}(2:4,5:6)

A{1,1}(2:4,2:3)./A{1,2}(2:4,5:6)

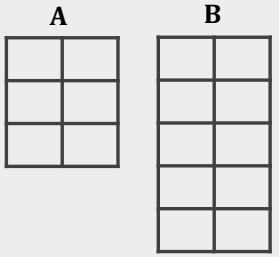
CF

Concatenation of Cell Arrays

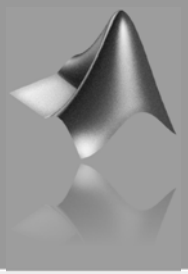
- The same rule as usual…

```
Command Window
>> A=zeros(3,2);
>> B=zeros(5,2);
>> C=[A B];
Error using horzcat
Dimensions of
matrices being
concatenated are not
consistent.

>> C=[A; B];
fx >>
```

```
Command Window
>> A=cell(3,2);
>> B=cell(5,2);
>> C=[A B];
Error using horzcat
Dimensions of
matrices being
concatenated are not
consistent.

>> C=[A; B];
fx >>
```



CF



結構矩陣
structure array

CF

Archive Big Dataset

For example,

- 30 normal controls vs. 30 patients with stroke

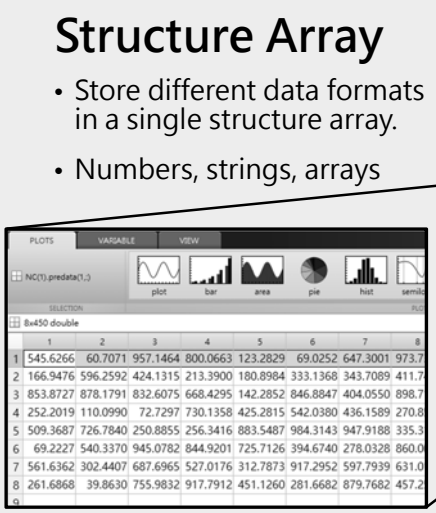
- Name, ID → string
- Age, gender, height, weight → numbers
- Pre-training data (8 x 450 values) → number array
- Post-training data (8 x 450 values) → number array

試試看在command window中鍵入
load('ExampleStruct.mat')

CF

Structure Array

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



1x30 struct with 7 fields

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...

Cell vs. Structure Array

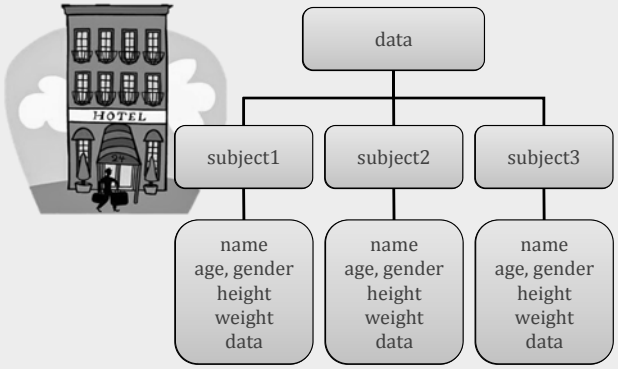
31x7 cell

	1	2	3	4	5	6	7	8
1	'name'	'age'	'gender'	'height'	'weight'	'predata'	'postda...	
2	'NC01'	62	1	172.1076	69.4049	8x450 d...	8x450 d...	
3	'NC02'	57	0	170.2542	52.2071	8x450 d...	8x450 d...	
4	'NC03'	58	1	174.2668	72.6109	8x450 d...	8x450 d...	
5	'NC04'	63	1	173.4476	69.0174	8x450 d...	8x450 d...	
6	'NC05'	65	1	168.8933	58.0383	8x450 d...	8x450 d...	
7	'NC06'	52	1	168.6337	58.7737	8x450 d...	8x450 d...	
8	'NC07'	65	0	165.8925	79.0742	8x450 d...	8x450 d...	
9	'NC08'	54	0	169.7196	79.6230	8x450 d...	8x450 d...	
10	'NC09'	52	0	174.6123	61.6098	8x450 d...	8x450 d...	
11	'NC10'	68	1	162.9426	64.0754	8x450 d...	8x450 d...	
12	'NC11'	68	1	163.8363	69.6155	8x450 d...	8x450 d...	
13	'NC12'	63	1	172.5047	79.3612	8x450 d...	8x450 d...	
14	'NC13'	55	1	173.8082	77.4354	8x450 d...	8x450 d...	
15	'NC14'	51	0	169.6144	77.6480	8x450 d...	8x450 d...	
16	'NC15'	54	1	169.6435	67.4950	8x450 d...	8x450 d...	

1x30 struct with 7 fields

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...

Structure of a Structure Array



a 1x3 structure array

6 **Field**s for each subject

CF

Create a Structure array

- data.name='NC01'; % string
- data.age=62; % number
- data.predata=rand(8,450); % number array
- data.notation={ 16, '%task onset'}; % cell array

1x1 struct with 4 fields

Field	Value	Min	Max
name	'NC01'		
age	62	62	62
predata	*8x450 double*	1.3...	0.9...
notation	*1x2 cell*		

Use variable viewer to check the data structure!

CF

Create a Structure array

- data(1).name='NC01'; % string
  data(1).age=62; % number
  data(1).predata=rand(8,450); % number array
  data(1).notation={ 16, '%task onset'}; % cell array
- data(2).name='NC02'; % string
  data(2).age=54; % number
  data(2).predata=rand(8,450); % number array
  data(2).notation={ 23, '%task onset'}; % cell array

Use variable viewer to check the data structure!

CF

More Intuitive Way

When dealing with different length of strings…

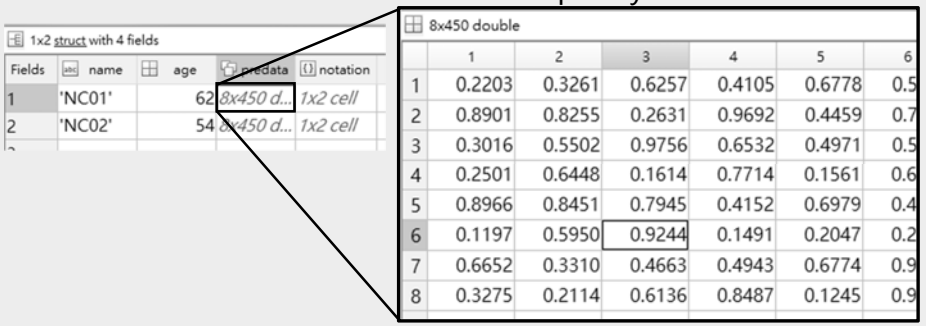
- List = {'David'; 'Andy'; 'Jay'; 'Jolin'; 'Selina'};

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

CF

Index of a struct array

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



1x2 struct with 4 fields

Fields	name	age	predata	notation
1	'NC01'	62	8x450 d...	1x2 cell
2	'NC02'	54	8x450 d...	1x2 cell

8x450 double

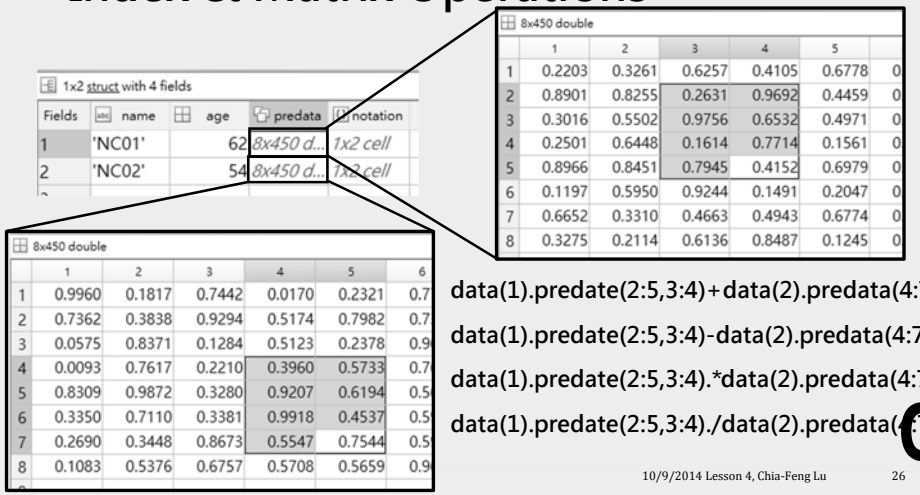
	1	2	3	4	5	6
1	0.2203	0.3261	0.6257	0.4105	0.6778	0.5
2	0.8901	0.8255	0.2631	0.9692	0.4459	0.7
3	0.3016	0.5502	0.9756	0.6532	0.4971	0.5
4	0.2501	0.6448	0.1614	0.7714	0.1561	0.6
5	0.8966	0.8451	0.7945	0.4152	0.6979	0.4
6	0.1197	0.5950	0.9244	0.1491	0.2047	0.2
7	0.6652	0.3310	0.4663	0.4943	0.6774	0.9
8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

data(1).predata(6,3)

Navigation in variable viewer...

CF

Index & Matrix Operations



1x2 struct with 4 fields

Fields	name	age	predata	notation
1	'NC01'	62	8x450 d...	1x2 cell
2	'NC02'	54	8x450 d...	1x2 cell

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

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

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

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

CF

Useful Functions

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

CF

Benefits for Using a Single Array

- Easy to clear, save, and load
- Easy to make it a global variable
  - global data
  - global name age gender height weight predate postdata
- Easy to categorize variables
  - Data % store all dataset-related information
  - Handle % store all GUI object handles
  - File % store all file-related information

CF

THE END

CF

alvin4016@ym.edu.tw