

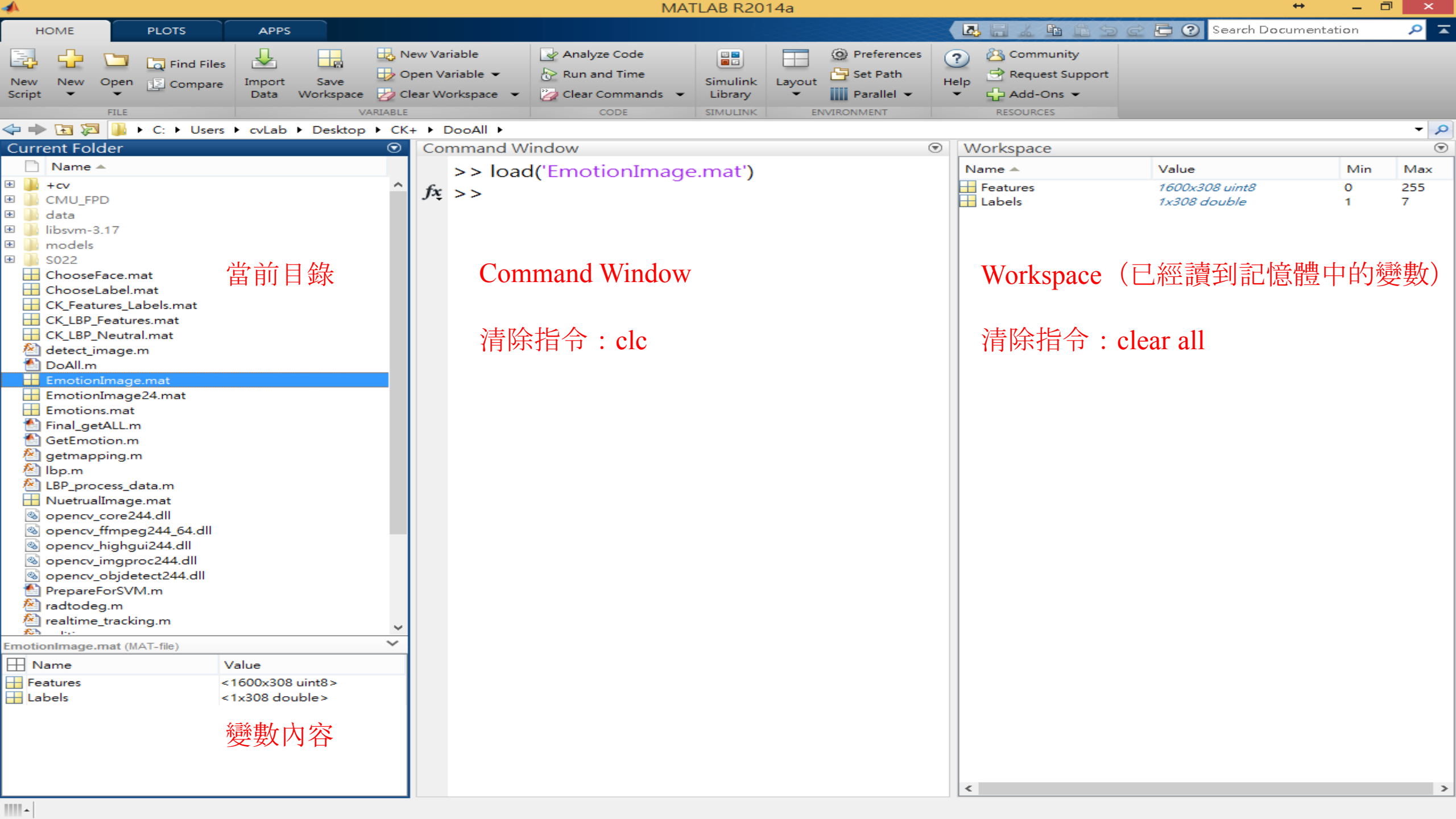
Matlab Introduction

Outline

- 安裝、簡介
- 向量、矩陣
- 迴圈、函數
- 字串、顯示
- 影像處理

安裝Matlab

校務資訊系統→計通中心相關服務→校園授權軟體下載系統→
中文→其他類→M6_MATLAB_**R201XA**_for_Windows



當前目錄

Command Window

Workspace (已經讀到記憶體中的變數)

清除指令：clc

清除指令：clear all

變數內容

Commandline

- 任何未指派結果的數學計算會將結果儲存在 **ans** 變數
- 指令最後加上 ; 運算結果就不會顯示於 **commandline**
- 變數第一字元必須是英文，後面可以接數字或是底線
- 變數最多只能有**31**個字母，大小寫有分別
- 句尾未加;的指令，會將結果顯示在**commandline**

Ex :

```
>> 32*32
```

```
ans =
```

```
1024
```

```
>> X = 1024/32
```

```
X =
```

```
32
```

```
>> X = X*X
```

```
X =
```

```
1024
```

變數儲存

- 若想要將變數儲存，可以使用**save**指令

指令	功能
<code>save</code>	將workspace中所有變數儲存為matlab.mat
<code>save filename</code>	將workspace中所有變數儲存為filename.mat
<code>save filename var1 var2</code>	將指定變數var1, var2儲存為filename.mat
<code>save('filename','var1','var2')</code>	功能同上

- 也可以直接在workspace上用右鍵選擇變數進行**save**操作

向量

- Matlab中的變數可以用來儲存向量(Vector)或者矩陣(Matrix)以進行各種運算

Ex :

```
>> t = [3 7 11 5]

t =

     3     7    11     5
```

以空白隔開元素，宣告一個Vector

```
>> t(6) = 10

t =

     3     7    11     5     0    10
```

在指定的位置新增一個元素，向量長度增長，中間的位置自動補上0

```
>> t(4) = []

t =

     3     7    11     0    10
```

利用空集合[]來刪除元素，向量長度會減少

矩陣

```
>> A = [1 2 3 4;5 6 7 8;9 10 11 12];  
>> A  
  
A =  
  
    1     2     3     4  
    5     6     7     8  
    9    10    11    12
```

```
>> A'  
  
ans =  
  
     1     5     9  
     2     6    10  
     3     7    11  
     4     8    12
```

宣告矩陣時，是以row-major的形式，
每個row以;隔開，直接在
commandline輸入變數名稱會直接印
出矩陣內容

在矩陣或向量變數後面加上'
就能得到其轉置(transpose)結果

矩陣

```
>> B = A(2, 1:3)
```

```
B =
```

```
5  6  7
```

利用:可以進行區間操作，若:兩邊不
加上數字則表示整個row或column

```
>> A = [A B']
```

```
A =
```

```
1  2  3  4  5  
5  6  7  8  6  
9 10 11 12  7
```

宣告或指派時，矩陣或向量的元素
也可以是矩陣或向量

矩陣

```
>> A = [ 1 2 3 ] + 5  
A =  
    6  7  8
```

矩陣和一般純量加減

```
>> A = [ 12 34 56 20 ]  
>> B = [ 1 3 2 4 ]  
>> C = A + B  
C =  
    13 37 58 24
```

相加或相減的矩陣必須具有相同的維度

矩陣

```
>> A = [123 , 442];    >> C = A/3
>> B = 2*A            C =
B =                    41.0000  147.3333
    246   884
```

純量的乘除

```
>> A = [ 1; 2 ];
>> B = [ 3 4 5 ];
>> C = A * B;
C =
    3 4 5
    6 8 10
```

矩陣相乘需要滿足 M by N x N by K

矩陣

```
>> A = [ 1 2 3 4 ]; B = [ 2 2 2 2 ];  
>> C = A .* B;  
>> C =  
  
    2  4  6  8
```

點積或點除，矩陣大小必須相同

```
>> A = magic(3); B = A^2;  
B = 91 67 67  
    67 91 67  
    67 67 91
```

方陣可以用^來進行次方運算

矩陣

- Matlab內建了許多數學函數，例如以下

函數	功能
$Y = \text{abs}(x)$	取 x 的絕對值
$Y = \sin(x)$	取 x 的正弦值
$Y = \exp(x)$	自然指數
$Y = \log(x)$	自然對數
$Y = \min(x)$	極小值
$Y = \max(x)$	極大值
$Y = \text{mean}(x)$	平均值
$Y = \text{sum}(x)$	總和
$Y = \text{sort}(x)$	排序

```
>> A = [ 1 2 3 4;5 6 7 8];  
>> sum(A)  
  
ans =  
  
    6    8   10   12
```

這些函數不僅限於向量(一維)，用在矩陣時預設為對**column**進行操作

```
>> A = [ 1 2 3 4;5 6 7 8];  
>> sum(A,2)  
  
ans =  
  
    10  
    26
```

在函數後面加上,2可以讓函數變成對**row**進行操作

迴圈

- Matlab提供重複迴圈(Loops)及條件判斷(Conditions)等流程控制指令

For :

```
a=0;  
for i = 1:10  
    a = a+i;  
end
```

While :

```
b=10;  
while b>0  
    b=b-1;  
end
```

Switch :

```
a = 0;  
switch b  
    case {1,3,5}  
        a = a+1;  
    case {2,4,6}  
        a = a*2;  
    otherwise  
        a = a -2;  
end
```

If / else :

```
if A > 10  
    A = A/2;  
else  
    A = A+1;  
end
```

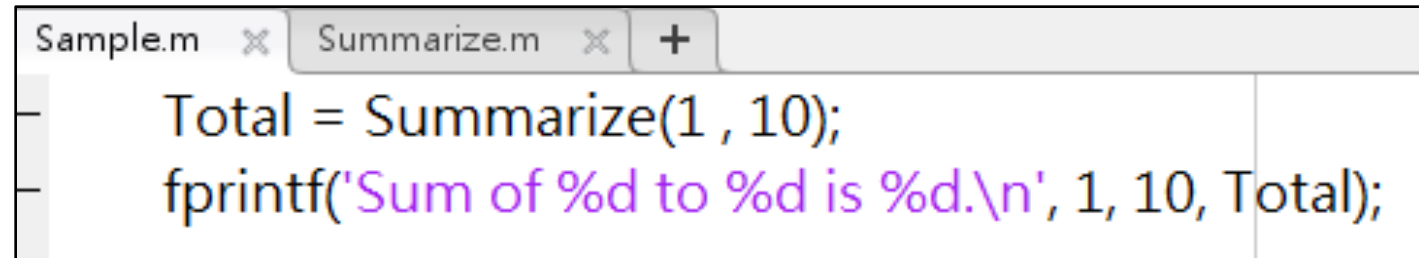
函數

- 若要一次執行大量的Matlab指令，可以將指令寫成一個副檔名為m的script
- 建立.m檔時，若選擇function，Matlab會內建預設的格式

```
function [ output_args ] = FunctionSample( input_args )  
%FUNCTIONSAMPLE Summary of this function goes here  
% Detailed explanation goes here  
  
end
```

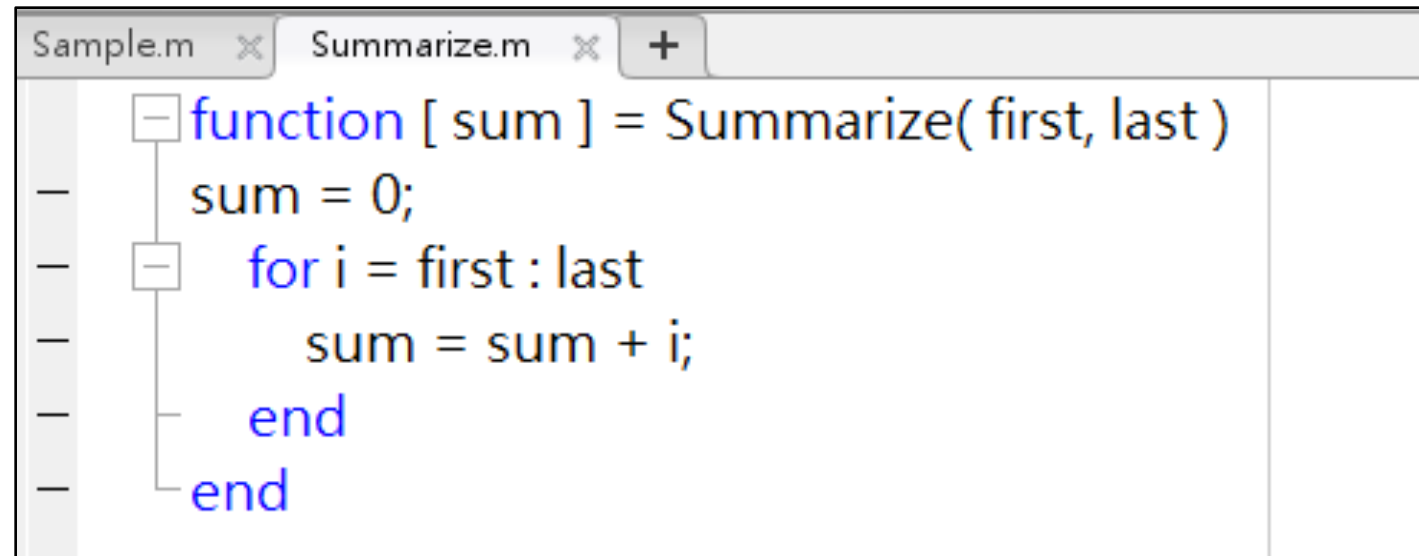
函數

```
>> Sample  
Sum of 1 to 10 is 55.
```



A screenshot of a MATLAB editor window with two tabs: 'Sample.m' and 'Summarize.m'. The 'Sample.m' tab is active. The code in the editor is:

```
Total = Summarize(1, 10);  
fprintf('Sum of %d to %d is %d.\n', 1, 10, Total);
```



A screenshot of a MATLAB editor window with two tabs: 'Sample.m' and 'Summarize.m'. The 'Summarize.m' tab is active. The code in the editor is:

```
function [ sum ] = Summarize( first, last )  
    sum = 0;  
    for i = first : last  
        sum = sum + i;  
    end  
end
```


函數

- 不論要執行的是Script還是Function，都必須在目前目錄下，Matlab才找得到，否則就必須使用addpath來讓Matlab知道要去哪裡找這些檔案。

```
addpath(genpath('c:/matlab/myfiles'))
```

- 使用genpath可以一次將所有子目錄也加入。

補充說明

- 在Matlab裡面，有很多方法可以避免使用迴圈，使你的程式更加快速

`>> A = randi([1 100],1,100);` 利用randi產生一個1~100隨機排列的row vector

如果想要知道特定的數字出現在那些位置，比較以下兩種方式：

```
A = randi([1 100],1,100);
X = 65;
for i = 1:size(A,2)
    if A(i) == X
        index = [index i];
    end
end
```

使用for迴圈的方法會跑很慢，如果你在迴圈執行時，有不斷地改變變數的size就會更慢。

`[value,index] = find(A==X);`

這樣就快很多

- 在Matlab裡面，`A==X`這個指令的結果是一個logical vector而find會找出所有不為零的數字的位置，多利用這兩項性質不僅能讓程式寫起來精簡，執行起來也會更快。

字串

- Matlab中字串是以char vector保存，字串的操作可參考以下幾個例子：

```
>> str1 = 'Cat';  
>> str2 = 'Meow';  
>> str3 = [str1, str2]
```

```
str3 =
```

```
CatMeow
```

```
>> str4 = strcat(str2,str3)
```

```
str4 =
```

```
MeowCatMeow
```

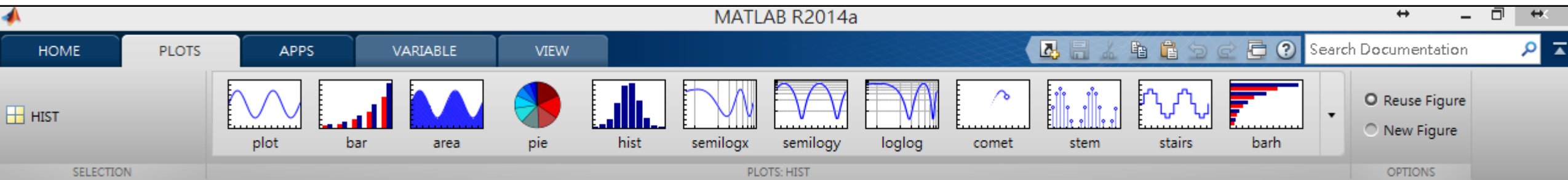
直接宣告一個字串變數

字串的串接可以用向量形式

也可以用**strcat**來串接字串

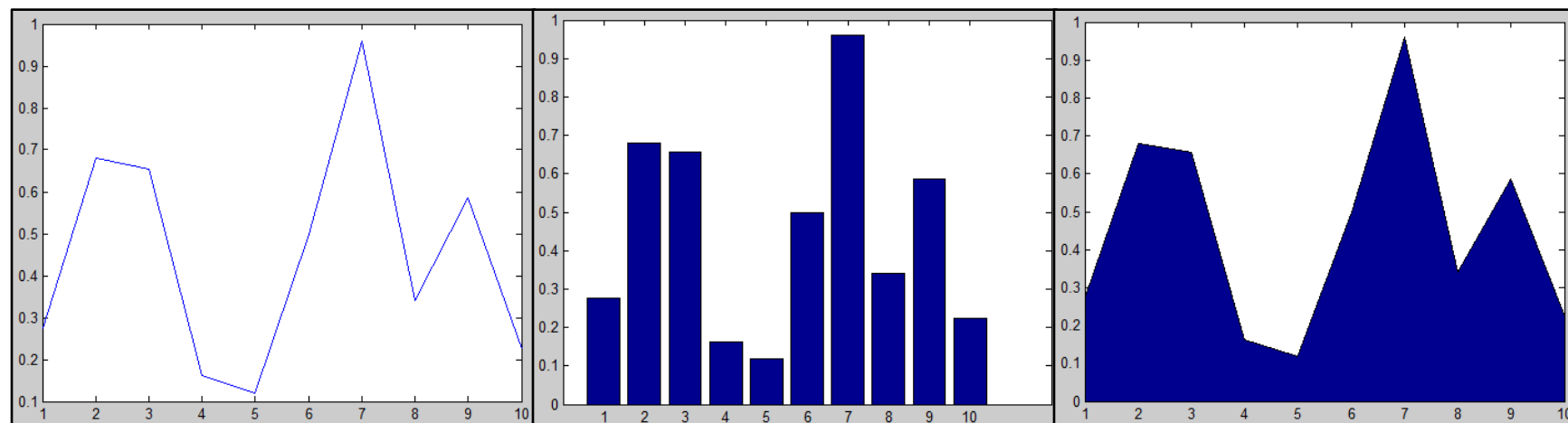
圖表

- Matlab提供很多種圖表function可以直接使用



- 可以直接用滑鼠點選，也可以在commandline輸入指令來顯示圖表

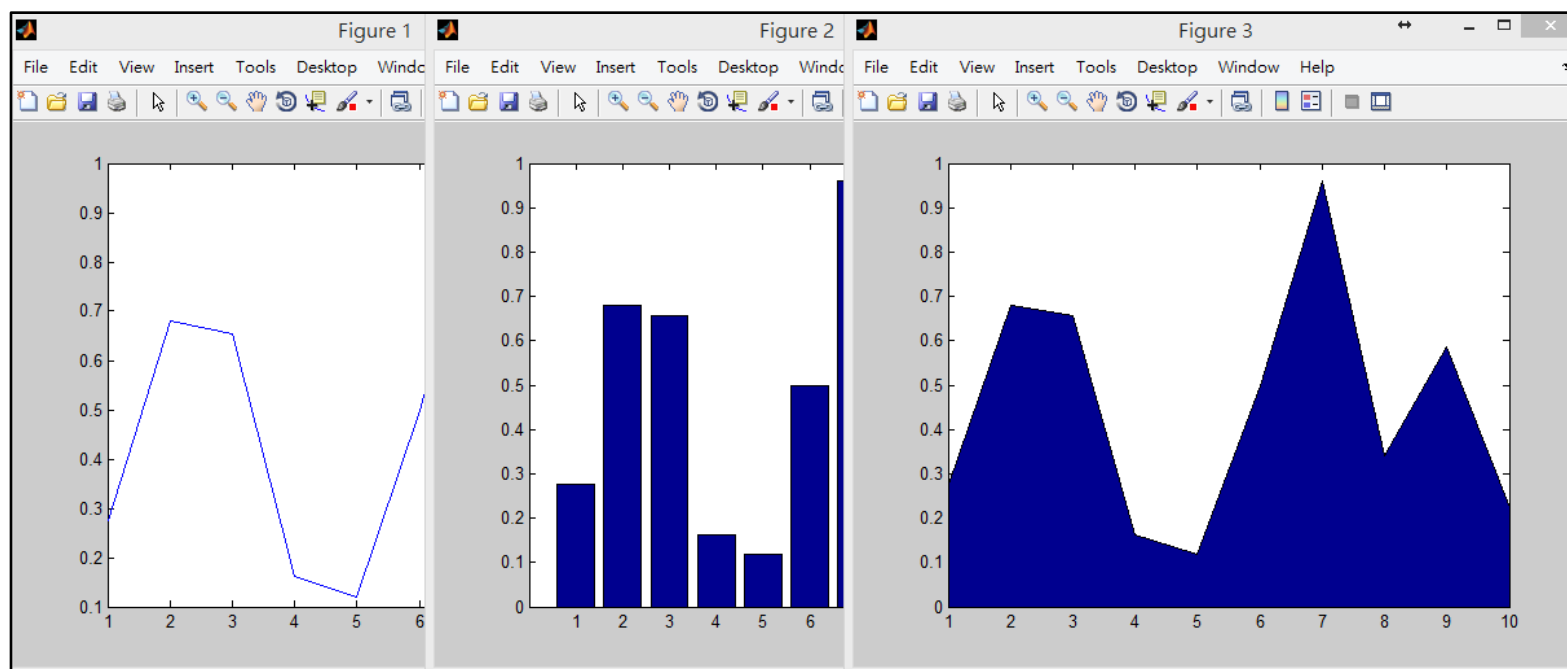
```
>> plot(HIST)
>> bar(HIST)
>> area(HIST)
>> pie(HIST)
>> hist(HIST)
```



圖表

- Matlab預設只會有一個顯示視窗(window)，如果按照上一頁投影片的方式顯示圖表，會不斷覆蓋前一個圖表，這時候可以使用**figure**指令，來告訴Matlab你希望有一個新的**window**來顯示接下來的內容。

```
>> figure  
>> plot(HIST)  
>> figure  
>> bar(HIST)  
>> figure  
>> area(HIST)
```



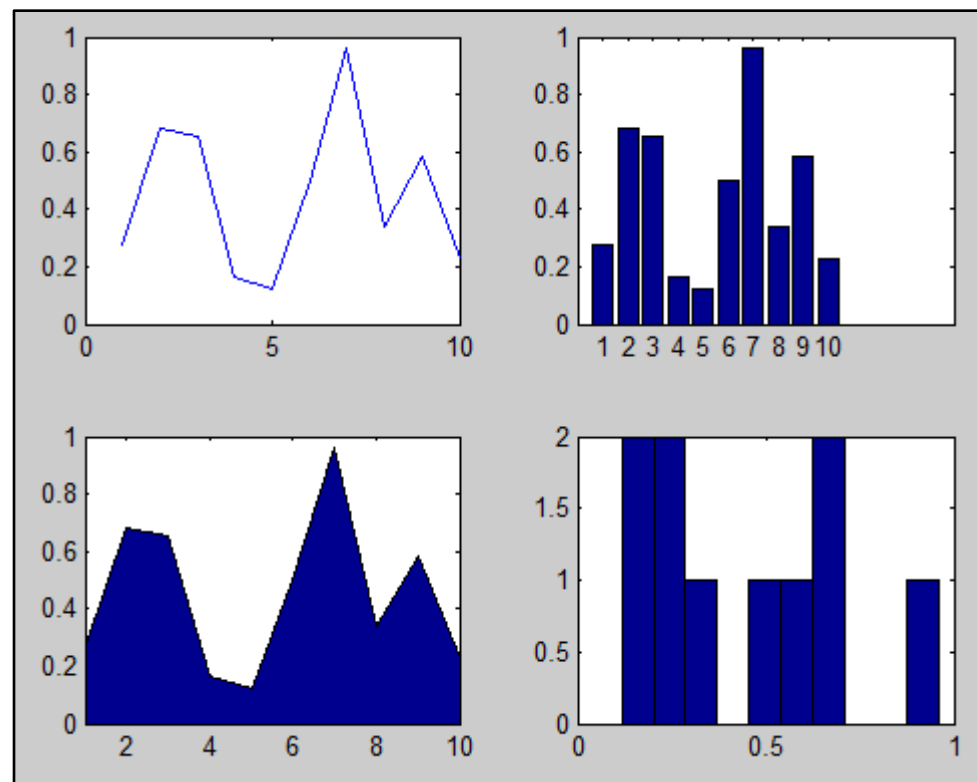
圖表

- 如果你希望所有的圖表能在同一個視窗中顯示，你可以使用subplot

subplot(rows, columns, index),display_something;

圖表的位置順序是row-major

```
subplot(2, 2, 1),plot(HIST);  
subplot(2, 2, 2),bar(HIST);  
subplot(2, 2, 3),area(HIST);  
subplot(2, 2, 4),hist(HIST);
```



影像處理

- Matlab的image processing toolbox中提供了很多和影像處理相關的函數

函數	功能
<code>A = imread('filename.png');</code>	讀取圖片
<code>imshow(Temp);</code>	顯示圖片
<code>G = rgb2gray(A);</code>	將RGB圖轉為灰階圖
<code>R = imrotate(A, angle);</code>	旋轉圖片
<code>S = imresize(A, scale);</code>	調整圖片大小

- 圖片被讀取進來之後會轉換成 **Height x Weight x Colors** 的三維 **uint8 matrix**
圖片可以分成**uint8(0~255)**和**double(0~1)**兩種儲存型態，在操作時需要注意變數型態，若不小心搞錯的時候有可能出現全黑或全白的結果。
可以利用**im2double(A)**以及**im2uint8(A)**轉換變數的型態。

影像處理

```
>> A = imread('Rennes.jpg');  
>> G = rgb2gray(A);  
>> B = im2double(A)+0.5;  
>> subplot(1,3,1),imshow(A);  
>> subplot(1,3,2),imshow(G);  
>> subplot(1,3,3),imshow(B);
```

讀取圖片時要注意路徑以及副檔名

可以直接用純量加法來調整亮度



影像處理

```
>> meanBrightness = mean(mean(G));  
>> G(G>1.5*meanBrightness)=255;  
>> G(G<0.5*meanBrightness)=0;  
>> imshow(G);
```



使用兩次**mean**取得整張灰階圖的平均值
將大於**1.5**倍平均的畫素點設置成白色
將小於**0.5**倍平均的畫素點設置成黑色
得到強化對比的圖片

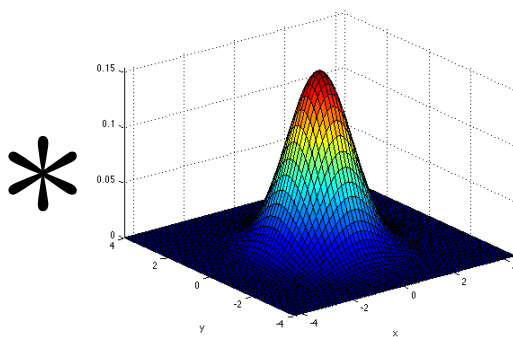


影像處理

- 在影像處理中，很常使用到convolution，Matlab也提供了很多常用的kernel可以呼叫

```
>> gaussian_filter = fspecial('gaussian',[9 9],3);  
>> B = imfilter(A,gaussian_filter);  
>> imshow(B);
```

呼叫一個9x9的Gaussian kernel
使用imfilter對整張圖做convolution



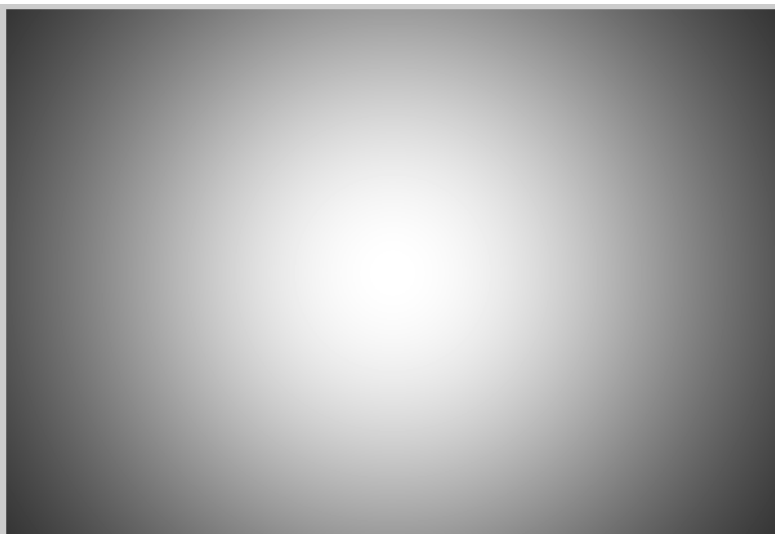
影像處理

- 也可以使用內建的kernel來製作mask處理影像

```
>> [h, w, ~] = size(A);  
>> gMask = fspecial('gaussian',[h w],0.5*min(h,w));  
>> gMask = gMask / max(max(gMask));  
>> imshow(gMask);  
>> imshow(double(A) .* repmat( gMask,1,1,3)/255);
```

使用**size**取得圖片的高和寬，不需要保存的變數可以用~表示
用**fspecial**呼叫一個**gaussian kernel**，第三個變數是**sigma**
Normalize讓其分布為0~1

利用**repmat**讓mask能對應到三個color的維度，並以點乘套用



Reference

- <http://mirlab.org/jang/books/matlabProgramming4beginner/slide/>
MATLAB程式設計《入門篇》 張智星 老師
- <http://mirlab.org/jang/books/matlabProgramming4guru/slide/>
MATLAB程式設計《進階篇》 張智星 老師
- <http://www.mathworks.com/help/matlab/>
MATLAB Documentation