

Scilab 簡介 2

2011/09/29

Chih-Han Lin 林致翰

`clin@ltl.iams.sinica.edu.tw`

or `r99245002@ntu.edu.tw`

常用函式

`zeros(m[,n,1...])` ex. `zeros(1,1)`

製造 m (乘 n 乘 $1\dots$) 的零陣列 (所有陣列元素為 0)

`ones(m[,n,1...])` ex. `ones(1,1)`

製造 m (乘 n 乘 $1\dots$) 的單位陣列 (所有陣列元素為 1)

`rand(m[,n,1...])` ex. `rand(1,1)`

製造 m (乘 n 乘 $1\dots$) 的亂數陣列 (所有陣列元素為 0~1 之間的亂數)

```
-->zeros(1,5)
```

```
ans =
```

```
0.    0.    0.    0.    0.
```

```
-->ones(5,5)
```

```
ans =
```

```
1.    1.    1.    1.    1.  
1.    1.    1.    1.    1.  
1.    1.    1.    1.    1.  
1.    1.    1.    1.    1.  
1.    1.    1.    1.    1.
```

```
-->rand(5,1)
```

```
ans =
```

```
0.2113249
```

```
0.7560439
```

```
0.0002211
```

```
0.3303271
```

```
0.6653811
```

```
-->rand(2,2,2)
```

```
ans =
```

```
(:, :, 1)
```

```
0.6283918
```

```
0.6857310
```

```
0.8497452
```

```
0.8782165
```

```
(:, :, 2)
```

```
0.0683740
```

```
0.6623569
```

```
0.5608486
```

```
0.7263507
```

迴圈練習

ex. 如何利用程式擲骰子？

1. 製造介於 0~6 的亂數 $\rightarrow 6 * \text{rand}()$ 或 $6 * \text{rand}(1, 1)$

1. 將 0~1 之間的的數歸類為數字面 1

1~2 之間的的數歸類為數字面 2，依此類推

可以使用無條件進位函數 `ceil`

ex. $\rightarrow \text{ceil}(1.2)$ `ans=2`

`ceil(1.5)` `ans=2`

`ceil(2.8)` `ans=3`

備註：無條件捨去：`floor()`，四捨五入：`round()`

```
for i=1:20
    s(1,i)=ceil(6*rand(1,1))
end
disp(s)
```

執行二十次

依序將產生的數字擺入 s 陣列中

印出儲存結果的陣列

執行 scinote 寫好的程式

```
->exec('C:\Users\clin\Documents\randomtest.sce', -1)
```

```
column 1 to 11
2.    1.    5.    1.    6.    4.    2.    4.    6.    5.    1.
column 12 to 20
5.    3.    2.    6.    6.    2.    6.    5.    6.
```

```
--> s = ceil(6 * rand(1, 20))
```

假如迴圈只是用來重複執行一些與迴圈變數(在這個例子中是 *i*) 無關的指令，這些指令如果可以處理陣列變數，那麼可以用陣列宣告來取代 `for` 迴圈，不僅執行速度較快也比較簡潔

```
s =
```

```
column 1 to 11
2.    1.    5.    1.    3.    2.    2.    2.    5.    1.    6.
column 12 to 20
3.    2.    1.    6.    6.    3.    5.    1.    1.
```

if 迴圈

```
if (邏輯變數) then  
迴圈程式碼  
end
```

假如 邏輯變數 == 1，執行 迴圈程式碼

假如 邏輯變數 == 0，執行 迴圈程式碼

通常 邏輯變數 是一個 邏輯判斷式 ex. $x > 1$, $a == b$, $w < 4$

如果該變數是一個數值，則只有該數值為零時不執行迴圈程式碼
(視為邏輯真值) 其他的狀況都會執行迴圈程式碼

```
if (邏輯變數) then  
迴圈程式碼 (1)  
else  
迴圈程式碼 (2)  
end
```

假如 邏輯變數 == 1，執行 迴圈程式碼 (1)

假如 邏輯變數 == 0，執行 迴圈程式碼 (2)


```
s=0.3  
if s>1 then  
disp('s>1')  
end
```



s > 1

```
s=1  
if s>1 then  
disp('s>1')  
end
```



無結果顯示

```
s=0.5  
if s>1 then  
    disp('s>1')  
else  
    disp('s<=1')  
end
```



s > 1

ex. 寫一個可以判定使用者數值是否為 3 的變數的程式

```
number=input('give me a number');  
  
if (number/3)-ceil(number/3)==0 then  
    disp('your number is divisible by 3')  
else  
    disp('your number is not divisible by 3')  
end
```

假如可以整除，則 `number/3` 會得到整數，與無條件捨去的結果相同

`modulo(m, n)` 輸出 m 除以 n 的餘數

```
--> modulo(5, 3)      ans = 2  
--> modulo(13, 3)     ans = 1  
--> modulo(18, 3)     ans = 0
```

```
number=input('give me a number');
```

```
if modulo(number, 3) == 0 then  
    disp('your number is divisible by 3')  
else  
    disp('your number is not divisible by 3')  
end
```

加強版 if: elseif 迴圈

```
if 邏輯變數(1) then  
迴圈程式碼(1)  
elseif 邏輯變數(2) then  
迴圈程式碼(2)  
elseif 邏輯變數(3) then  
迴圈程式碼(3)  
else  
迴圈程式碼(4)  
end
```

若 邏輯變數(1) 為真，
執行 迴圈程式碼(1)
否則判定 邏輯變數(2)
若 邏輯變數(2) 為真，
執行 迴圈程式碼(2)
否則判定 邏輯變數(3)
若 邏輯變數(3) 為真，
執行 迴圈程式碼(3)
否則執行 迴圈程式碼(4)
結束迴圈

可不加

ex. 改寫可以判定使用者數值是否為 3 的變數的程式
告訴使用者該數 -1 或 -2 才能被 3 整除

```
number=input('give me a number');  
  
if modulo(number,3)==0 then  
    disp('your number is divisible by 3')  
elseif modulo(number,3)==1 then  
    disp('your number -1 is divisible by 3')  
else  
    disp('your number -2 is divisible by 3')  
end
```

select 迴圈 (效果類似 elseif)

```
select 變數
case 數值(1) then 迴圈程式碼(1)
case 數值(2) then 迴圈程式碼(2)
.....
case 數值(n) then 迴圈程式碼(n)
else
迴圈程式碼(else)
end
```

若 變數等於數值(1)，執行 迴圈程式碼(1)

若 變數等於數值(2)，執行 迴圈程式碼(2)，其餘累推

若 變數不等於任何 case 後的提示數值

執行 迴圈程式碼(else)，結束迴圈

ex. 改寫可以判定使用者數值是否為 3 的變數的程式
告訴使用者該數 -1 或 -2 才能被 3 整除
若該數不為整數，告訴使用者這不是整數

```
number=input('give me a number');  
  
select modulo(number,3)  
  case 0 then  
    disp('your number is divisible by 3')  
  case 1 then  
    disp('your number -1 is divisible by 3')  
  case 2 then  
    disp('your number -2 is divisible by 3')  
  else  
    disp('your number is not an integer')  
end
```

printf

printf 顧名思義即 Scilab 模擬 C language 列印輸出的函數，用法與 C language 的相同，範例中的 `< %+specifier >` 可以看成不同資料型態的變數，`%s %d` 依序對應到後頭以 comma 分開的變數 `name` 與 `age-1`；不同的 specifier 代表不同的 data type，比方說 `s` 代表字串，`d` 代表十進位制的數值變數，詳細的用法請自行查找 Scilab 的說明。

```
-->name='clin'; age=21;
```

```
printf('You are %s. Next year you will be %d. ', name, age+1)
```



```
You are clin. Next year you will be 22.
```

規則可以參考 <http://www.cplusplus.com/reference/cstdio/printf/>

常用的有 `%f`: Decimal floating point, `%s`: String of characters,

`%d`: Signed decimal integer

`\n` 換行, `%%`印出百分比符號, `\t` 跳格

ex. 利用亂數模擬擲骰 10000 次，並統計各數字出現的頻率

```
count=zeros(1,6);

for i=1:10000
    select ceil(6*rand())
        case 1 then count(1,1)=count(1,1)+1;
        case 2 then count(1,2)=count(1,2)+1;
        case 3 then count(1,3)=count(1,3)+1;
        case 4 then count(1,4)=count(1,4)+1;
        case 5 then count(1,5)=count(1,5)+1;
        case 6 then count(1,6)=count(1,6)+1;
    end
end

for k=1:6
printf('probability of result %i is %f \n'...,
k,count(1,k)/1000)
end
```

連三個句點在 scilab 中代表
接續下一行的意思

輸出結果

```
probability of result 1 is 0.161700  
probability of result 2 is 0.164100  
probability of result 3 is 0.168200  
probability of result 4 is 0.170400  
probability of result 5 is 0.170000  
probability of result 6 is 0.165600
```

`size(array, n)` 找出陣列變數 `array` 第 `n` 維的元素數目

```
a = 1.    2.    3.    -->size(a)    ans= 2. 3.
     4.    5.    6.    -->size(a,1)   ans= 2.
a 是 2x3 的矩陣      -->size(a,2)   ans= 3.
```

`find(array, 條件式)` 找出 `array` 中符合條件式的元素座標

```
    a(1)      a(4)      a(6)      a(7)
      ↓        ↓        ↓        ↓
a=[1, 3, 4, 2, 4, 2, 1, 5, 3]
--> find(a<3)
ans= 1.      4.      6.      7.
a2=a(find(a<3))
--> a2= 1.      2.      2.      1.
```

ex. 利用亂數模擬擲骰 10000 次，並統計各數字出現的頻率
(搭配 `find` 與 `size` 指令)

```
count=zeros(1,6);  
  
result=ceil(6*rand(1,10000));  
  
for k=1:6  
    temp=find(result==k);  
    count(1,k)=size(temp,2);  
    printf('probability of result %i is %f \n'...,  
        k,count(1,k)/10000)  
end
```

字串

```
--> var1='it'           var1= it
--> var2='is'           var2= is
--> var3='here'        var3= here
```

```
-->var1+var2+var3
```

```
ans = itishere
```

```
-->var1+' '+var2+' '+var3
```

```
ans = it is here
```

```
-->var3+' '+var1+' '+var2
```

```
ans = here it is
```

```
-->ascii('abcd')
```

```
ans = 97.      98.      99.      100.
```

String 相關指令 (說明手冊 Scilab help-> Strings 分項)

string : conversion to string

eval : evaluation of a matrix of strings

strchr : Find the first occurrence of a character in a string

grep : find matches of a string in a vector of strings

strcmp : compare character strings

.....

```
--> a=[ 'what is your name?' , 'how old are you ?' ] 
```

```
a = !what is your name? how old are you ? !
```

```
-->name = input( a(1),'s' ) 
```

```
what is your name?  clin  name = clin
```

```
--> age = input( a(2) ) 
```

```
how old are you ?  21  age=21.
```

檔案存取

```
--> x=1:0.5:3            x = 1.    1.5    2    2.5    3.  
--> y=0:-0.5:-2         y = 0.    -0.5   -1   -1.5   -2.  
--> save('dataXY.dat',x,y)       --> clear 
```

```
-->listvarinfile('data.dat') 
```

Name	Type	Size	Bytes
x	constant	1 by 5	56
y	constant	1 by 5	56


```
--> load('data.dat', 'y') 
--> x       !--error 4 未定義變數
--> y       y = 0.  -0.5  -1  -1.5  -2.
```

```
--> clear       --> load('data.dat') 
--> x       x = 1.  1.5  2  2.5  3.
--> y       y = 0.  -0.5  -1  -1.5  -2.
```

檔案存取 (txt 檔)

開啟

檔案名稱

若檔案存在則開啟該檔案
若不存在則創造該檔案

```
--> Id1=file('open','demotext.txt','unknown')
```

Id1 = 1.

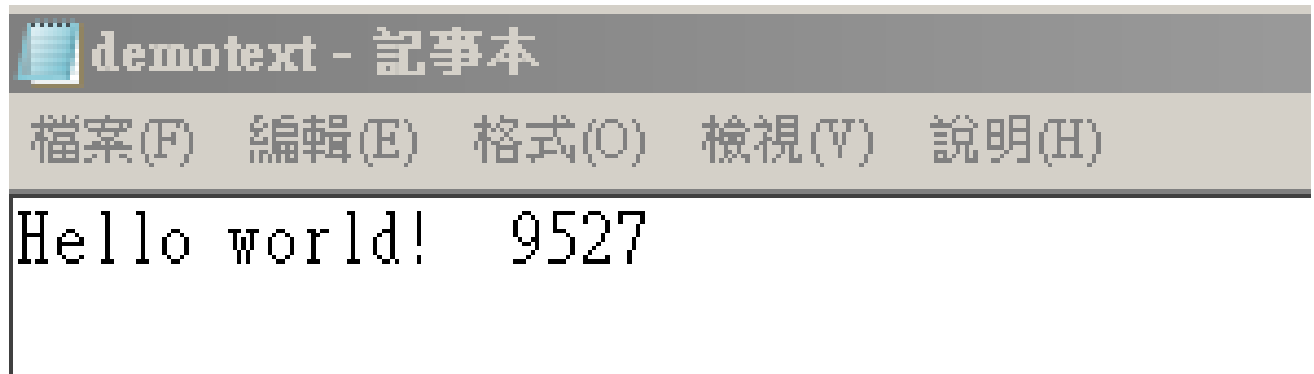
對 Scilab 而言，file 會賦予處理的檔案一個標號
所有相關的指令只要輸入指定的編號就能辨別檔案對象

```
--> fprintf(Id1,'Hello world! %d', 9527)
```

```
--> file('close',Id1)
```

fprintf 的用法與 printf 相同
只不過是將內容寫到指定的檔案中

處理完檔案要記得
關閉它。



顯示結果

使用 `mgetl` 可以讀取 `txt` 檔案內容

找出當前的工作目錄路徑

```
--> pwd          ans = E:\scilabworkdic
-->mgetl('E:\scilabworkdic\demotext.txt')

ans = Hello world!  9527  取出內容
```

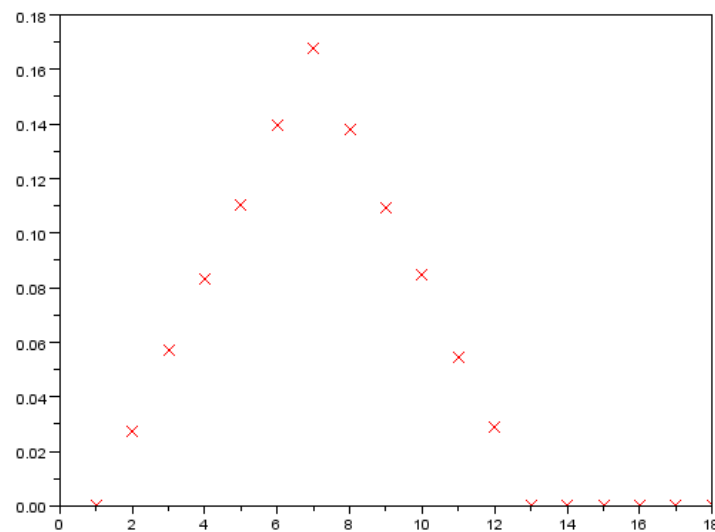
相關且常用的指令還有 `mgetstr`, `mopen`, `mclose`, `mput`, `mputl` 等等，有關 `file` 以及 `write`, `read` 的用法請參考講義 `LNEP_programming.pdf`

Q: 假如需要的文字檔非常大，你如何利用已有的指令按照你設定的規則取出你要得資訊？

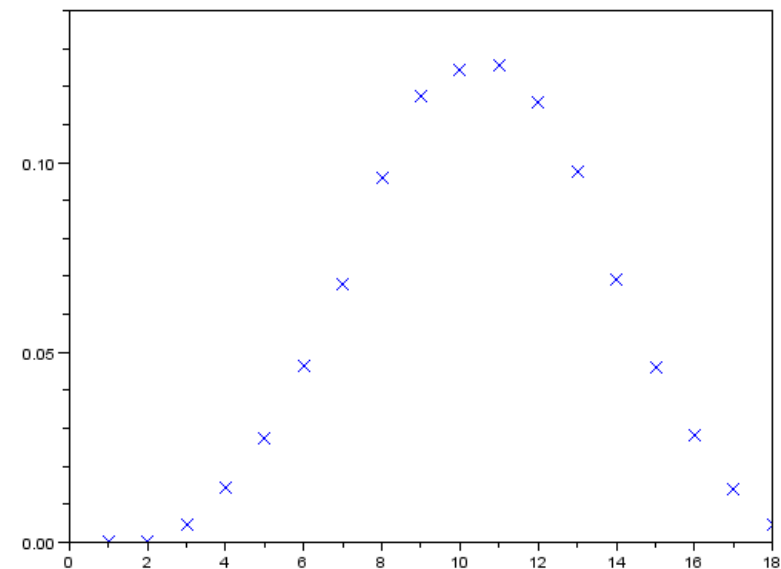
實作內容

請模擬擲兩顆骰子（皆為六面）100000次，計算骰子兩面點數之和（1,2,3,4,5,..., 11,12）出現的機率，並繪圖表示

請模擬擲三顆骰子（皆為六面）100000次，計算骰子三面點數之和（1,2,3,4,5,..., 17,18）出現的機率，並繪圖表示



兩顆骰子



三顆骰子

1. 寫一個程式 `WriteMyText.sce`，具備以下功能：
 - (a) 要求使用者輸入一段文字
 - (b) 接著要求使用者輸入檔名 ex. `mytext`
 - (c) 最後使用者輸入的文字會被儲存到 `mytest.txt` 中

2. 寫一個程式 `ReadMyText.sce`，具備以下功能：
 - (a) 要求使用者輸入檔名 ex. `mytext`
 - (b) `mytest.txt` 的內容會被輸出至命令列中 (螢幕上)

檔案也可以是 `.dat`，文字儲存到變數 `contents` 中

3. 寫一個程式 `MergeMyText.sce`，具備以下功能：

- (a) 要求使用者依序輸入兩個檔名 `ex. mytext1` 與 `ex. mytext1`
- (b) 要求使用者輸入一個新檔名 `ex. mytext_new`
- (c) `mytext1.txt` 與 `mytext2.txt` 的內容會合成一份新的文件 `mytext_new.txt`，且 `mytext2.txt` 的內容會被排在 `mytext1.txt` 之後。

助教如何測試你的程式是否正常工作：

使用 `WriteMyText` 寫入文字 `Here is the contents in mytext1.dat.` ; 存入檔案 `mytext1.dat`，寫入文字 `Here is the contents in mytext2.dat.` ; 存入檔案 `mytext2.dat`，使用 `MergeMyText` 合併檔案 `mytext1.dat` 與 `mytext2.dat`，並將結果存入 `mytext3.dat`。最後使用 `ReadMyText` 來讀取 `mytext3.dat`，螢幕上應該顯示

`Here is the contents in mytext1.dat Here. is the contents in mytext2.dat.`