



105-1: EE4052  
計算機程式設計  
Computer Programming

## Unit 07: 邏輯運算與流程控制

連 豐 力

臺大電機系

Sep 2016 - Jan 2017



### 大綱

計算機程式設計 - 2016F  
Unit 07: 邏輯與流程控制  
Feng-Li Lian @ NTU-EE

- 邏輯變數及運算
- 條件分支
- 迴圈

# 簡介

- $x \leftarrow c(-1.2, 3.4, 5.7, -5, 0, 2)$  % 6 個數字
- $y \leftarrow c(-2.2, 4.4, -6.6, 8.8, 0, -3.3)$  % 6 個數字
- 在  $x$  中，哪幾個大於或等於2，共有幾個，以及哪些數字？
- 在  $x$  與  $y$  中，哪幾個滿足  $x_i < y_i$ ，共有幾個，以及哪些數字？

# HW06：邏輯運算與流程控制

- 下表上學期修習 R 程式語言這門課的10位同學成績：

ID	1	2	3	4	5	6	7	8	9	10
Score	92	74	85	60	45	83	66	78	95	55

- 為了要給每一位同學評語，如下所示：
  - 59分(含) 以下： bad
  - 60 - 75分： ok
  - 76 - 90分： good
  - 91分(含)以上： excellent

ID	1	2	3	4	5	6	7	8	9	10
Score	92	74	85	60	45	83	66	78	95	55
Grading	excellent	ok	good	ok	bad	good	ok	good	excellent	bad

## 邏輯變數及運算

5

## 邏輯運算子

計算機程式設計 - 2016F  
Unit 07: 邏輯與流程控制  
Feng-Li Lian @ NTU-EE

### 常用的邏輯運算子：

- `!x` NOT 運算 (非)
- `x & y` AND 運算 (且)
- `x && y` AND 運算  
(但只運算第一個分量)  
結果是一維的邏輯向量，常用於 if
- `x | y` OR 運算 (或)
- `x || y` OR 運算  
(但只運算第一個分量)  
結果是一維的邏輯向量，常用於 if
- `xor(x, y)` Exclusive OR 運算
- `is.na(x)` 是否為遺漏值 (missing value?)

INPUT		OUTPUT	OUTPUT	OUTPUT
A	B	A AND B	A OR B	A XOR B
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

## 將其他類型轉為邏輯類型

計算機程式設計 - 2016F  
Unit 07: 邏輯與流程控制  
Feng-Li Lian @ NTU-EE

- `as.logical()` 將其他類型轉為邏輯類型之物件
- 將數字向量轉為邏輯類型之物件
  - `as.logical( c(0, 1) )`
  - `as.logical( c( -2.2, -1, 0, 1, 2.2 ) )`
- 將文字轉為邏輯類型之物件
  - `as.logical( c("T", "TRUE", "True", "true") )`
  - `as.logical( c("F", "FALSE", "False", "false") )`
  - `as.logical( "handsome" )`

- 7

## 判斷一個物件是否為邏輯類型

計算機程式設計 - 2016F  
Unit 07: 邏輯與流程控制  
Feng-Li Lian @ NTU-EE

- `is.logical()` 判斷一個物件是否為邏輯類型
- `is.logical( 3 < 5 )`
- `is.logical( c(TRUE, FALSE, FALSE, TRUE) )`
- `is.logical( c(-2.2, -1, 0, 1, 2.2) )`
- `is.logical( "handsome" )`

- 8

## 邏輯函數 – any(), all()

計算機程式設計 – 2016F  
Unit 07: 邏輯與流程控制  
Feng-Li Lian @ NTU-EE

- `x <- c(-1.2, 0.5, 1.0, 1.3, 2.4, 5, 6.3)`
- `any(1 < x)`
- `any(x < 5)`
- `any(1 < x & x < 5)` • 是否有**任何數字**介於1到5之間？
- `any(1 < x) & any(x < 5)`
  
- `all(1 < x)`
- `all(x < 5)`
- `all(1 < x & x < 5)` • 是否有**所有數字**介於1到5之間？
- `all(1 < x) & all(x < 5)`

- 9

## 邏輯函數 – which()

計算機程式設計 – 2016F  
Unit 07: 邏輯與流程控制  
Feng-Li Lian @ NTU-EE

- `x <- c(1.2, -3.4, 5.7, -6, 0, 3)`
- `which(x >= 1)` • 哪幾個數字大於等於1？
- `which(x >= 1) & (x <= 4)` • 哪幾個數字大於等於1且小於等於4？
- `which(x >= 6)` • 哪幾個數字大於等於6？
  
- `class(which(x >= 1))`
  
- `x[which(x >= 1)]` • 哪些數字大於等於1？
  
- `length(which(x >= 1))` • 有幾個數字大於等於1？
- `length(which(x >= 6))` • 有幾個數字大於等於6？

- 10

## 條件分支

11



## 關係運算子

計算機程式設計 - 2016F  
Unit 07: 邏輯與流程控制  
Feng-Li Lian @ NTU-EE

- 常用的關係運算子：
  - $x > y$             大於
  - $x \geq y$             大於等於
  - $x < y$             小於
  - $x \leq y$             小於等於
  - $x == y$             等於
  - $x != y$             不等於

- 12

- `3 < 5`
- `3 > 5`
  
- `class( 3 < 5 )`
- `as.integer( 3 < 5 )`
- `as.integer( 3 > 5 )`
  
- `2.5 * ( 3 < 5 )`
- `2.5 * ( 3 > 5 )`

- `x <- c( 1.2, -3.4, 5.7, -6, 0, 3 )`
  
- `x >= 0`
- `as.integer( x >= 0 )`
- `sum( x >= 0 )`
- `table( x >= 0 )`
  
- `x[ x >= 0 ]`

## 關係運算範例 - 數列比較

計算機程式設計 - 2016F  
Unit 07: 邏輯與流程控制  
Feng-Li Lian @ NTU-EE

- `x <- c(1.2, -3.4, 5.7, -6, 0, 3)`
- `y <- c(2.2, -4.4, 6.6, -8.8, 0, 3.3)`
  
- `x < y`
  
- `sum(x < y)`
  
- `table(x < y)`
  
- `x[x < y]`

- 15

## 關係運算範例 - 數值相等比較

計算機程式設計 - 2016F  
Unit 07: 邏輯與流程控制  
Feng-Li Lian @ NTU-EE

- `x <- 0.5 - 0.3`
- `y <- 0.3 - 0.1`
  
- `x == y`
  
- `sprintf("%.20f", x)`
- `sprintf("%.20f", y)`
  
- `all.equal(x, y)`
- `identical(all.equal(x, y), TRUE)`
- `round(x, 10) == round(y, 10)`

- 16



- `a <- 1:10 / 16`
- `a`
- `sprintf("%.20f",a)`
- `a <- 1:10 / 10`
- `a`
- `sprintf("%.20f",a)`

- `if( statement-1 )`  
  `{`  
    `statement-2`  
  `}`
- `if( statement-1 ){`  
  `statement-2`  
  `} else {`  
    `statement-3`  
  `}`

## 條件分支函數範例

計算機程式設計 - 2016F  
Unit 07: 邏輯與流程控制  
Feng-Li Lian @ NTU-EE

- `x <- "handsome"`
- `y <- 2`
- `z <- 2`
  
- `if( x == "handsome" ) { y <- y + 3 }`
- `y`
  
- `if( x == "beautiful" ) { z <- z + 3 }`
- `z`

- 19

## 條件分支函數範例

計算機程式設計 - 2016F  
Unit 07: 邏輯與流程控制  
Feng-Li Lian @ NTU-EE

- `x <- 2.5; y <- 4.7`
  
- `if ( x < y ) { z1 <- x } else { z1 <- y }`
- `z1`
  
- `if ( x > y ) { z2 <- x } else { z2 <- y }`
- `z2`
  
- `min( x, y )`
- `max( x, y )`

- 20

## 條件分支函數 – ifelse()

計算機程式設計 – 2016F  
Unit 07: 邏輯與流程控制  
Feng-Li Lian @ NTU-EE

- `ifelse( logical condition, value.true if TRUE, value.false if FALSE )`
- `x <- -1.5`
- `ifelse( x > 0, x, -x )`
- `abs( x )`

- 21

## 條件分支函數 – switch()

計算機程式設計 – 2016F  
Unit 07: 邏輯與流程控制  
Feng-Li Lian @ NTU-EE

- `switch( expression, statement-1, statement-2, statement-3, ..... )`
- `x <- 3`
- `switch( x, 2 + 2, mean(1:10), 1:5 )`
- `switch( 6, 2 + 2, mean(1:10), 1:5 )`

- 22

## 條件分支函數 – switch()

計算機程式設計 – 2016F  
Unit 07: 邏輯與流程控制  
Feng-Li Lian @ NTU-EE

- `switch( expression, statement-1, statement-2, statement-3, ..... )`
  
- `y <- "fruit"`
- `switch( y, fruit = "banana", vegetable = "broccoli" )`
  
- `y <- "vegetable"`
- `switch( y, fruit = "banana", vegetable = "broccoli" )`
  
- `y <- "meat"`
- `switch( y, fruit = "banana", "Neither" )`

- 23

## 大綱

計算機程式設計 – 2016F  
Unit 07: 邏輯與流程控制  
Feng-Li Lian @ NTU-EE

# 迴圈

## 迴圈指令 – for()

計算機程式設計 – 2016F  
Unit 07: 邏輯與流程控制  
Feng-Li Lian @ NTU-EE

- for( )
- while( )
- repeat( )
  
- next( )
- break( )

- 25

## 迴圈指令 – for()

計算機程式設計 – 2016F  
Unit 07: 邏輯與流程控制  
Feng-Li Lian @ NTU-EE

- `for( index in index.set ) {`  
    statement  
    }  
■  $x(k+1) = 4 x(k) ( 1 - x(k) )$   
■  $k = 1, 2, 3, \dots, x(1) = 0.2$   
  
■ `x <- 0.2`  
■ `for ( k in 2:5 ) {`  
     $x[k] <- 4 * x[k-1] * ( 1 - x[k-1] )$   
    }  
■ `round( x, 4 )`

- 26

## 迴圈指令 – for()

- 給定一個長度為10之向量，計算奇數項之和與偶數項之和的差。
- `x <- 1:10`
- `odd <- seq( from = 1, to = 9, by = 2 )`
- `even <- seq( from = 2, to = 10, by = 2 )`
- `mysum <- mysum.odd <- mysum.even <- 0`
- `for( i in x ) { mysum <- mysum + x[i] }`
- `for( i in odd ) { mysum.odd <- mysum.odd + x[i] }`
- `for( j in even ) { mysum.even <- mysum.even + x[j] }`
- `mysum.odd - mysum.even`
- `sum( x )`
- `sum( x[odd] ) - sum( x[even] )`

- 27

## 迴圈指令 – while()

- `while( statement-1 ) {`  
    `statement-2`  
}
  - `x <- c( 1.2, 3.4, 2.1, 4.3, 3.2, 5.5, 6.7 )`
  - `total <- x[ 1 ]`
  - `count <- 1`
  - `while ( total <= 12 ) {`  
    `count <- count + 1`  
    `total <- total + x[ count ]`  
}
  - `count; total`
- 總數只有12元，可以買幾個？

- 28

## 迴圈指令 – while()

- count; total

- 總數只有12元，可以買幾個？

- y <- cumsum( x )

- sum( y <= 12 )

## 迴圈指令 – repeat()

- repeat { statements

  - if ( statement-1 ) break

- }

- x <- c( 1.2, 3.4, 2.1, 4.3, 3.2, 5.5, 6.7 )

- total <- x[ 1 ]

- count <- 1

- repeat{ count <- count + 1; total <- total + x[ count ]

  - if ( total + x[ count + 1 ] >= 12 ) break

- }

- count; total

## 作業

31

## HW06：邏輯運算與流程控制

計算機程式設計 - 2016F  
Unit 07: 邏輯與流程控制  
Feng-Li Lian @ NTU-EE

On 11/1, 2016

- 下表上學期修習 R 程式語言這門課的 10 位同學成績：

ID	1	2	3	4	5	6	7	8	9	10
Score	92	74	85	60	45	83	66	78	95	55

- 為了要給每一位同學評語，如下所示：
  - 59分(含) 以下： bad
  - 60 - 75分： ok
  - 76 - 90分： good
  - 91分(含)以上： excellent
- 請參考，課程網站上的資料檔案，與範例程式。



# HW06：邏輯運算與流程控制

計算機程式設計 - 2016F  
Unit 07: 邏輯與流程控制  
Feng-Li Lian @ NTU-EE

On 11/1, 2016

- 請寫一個程式，使用下面三種方式中的一種方式，來達到這個目的：
  1. 使用 `if/else` 的設計
  2. 使用 `which()` 的設計
  3. 使用 `switch()` 的設計
- 輸出結果到：`scoreIF`，`scoreWHICH`，`scoreSWITCH`，內容至少包含：

ID	1	2	3	4	5	6	7	8	9	10
Score	92	74	85	60	45	83	66	78	95	55
Grading	excellent	ok	good	ok	bad	good	ok	good	excellent	bad

- 比較一下這三個結果是否一致？
- 在 `.R` 的程式碼中，註解您所加註的程式碼的意義或想法。

- 33

# HW06：邏輯運算與流程控制

計算機程式設計 - 2016F  
Unit 07: 邏輯與流程控制  
Feng-Li Lian @ NTU-EE

On 11/1, 2016

- 繳交下面檔案，檔案名稱：`HW06_學號`
  - 主要指定檔案：`HW06_B01921001.R`
  - 不需要另外繳交報告檔案
- 繳交方式與期限：
  - E-mail 檔案到：[ntucp105f@gmail.com](mailto:ntucp105f@gmail.com)
  - E-mail 主旨：`HW06_B01921001` (就是，作業編號\_您的學號)
  - 繳交期限：`11/2 (Wed), 2016, 11pm` 以前
  - `HW06`，每位繳交自己的程式碼，  
程式碼中需要註解所加註的程式碼的意義或想法。

- 34