



大數據資料處理實務

李水彬

2023-09-01

Chapter 05-1 邏輯運算

比較運算子

運算元	作用	優先次序	範例(x=6)	結果
==	等於	1	x==5	FALSE
!=	不等於	2	x!=5	TRUE
<	小於	3	x<5	FALSE
>	大於	4	x>5	TRUE
<=	小於或等於	5	x<=5	FALSE
>=	大於或等於	6	x>=5	TRUE

純量比較

- 純量(scalar) 指變數為單一數值。

```
x<-6 # set x to 6
```

```
x==5 # check if x is equal to 5
```

```
## [1] FALSE
```

```
x!=5 # check if x is not equal to 5
```

```
## [1] TRUE
```

純量比較

```
x<5 # check if x is smaller than 5
```

```
## [1] FALSE
```

```
x>5 # check if x is bigger than 5
```

```
## [1] TRUE
```

```
x<=5 # check if x is not bigger than 5
```

```
## [1] FALSE
```

```
x>=5 # check if x is not smaller than 5
```

```
## [1] TRUE
```

向量與純量的比較

- 向量(vector)包含複數個純量的變數，例如

```
x<-c(6,4,5)# x is the vector containing three elmenets 6, 4,5  
x
```

```
## [1] 6 4 5
```

- 比對向量 x 每個值和5是否相同。

```
x==5 # check if x is equal to 5
```

```
## [1] FALSE FALSE TRUE
```

```
x!=5 # check if x is not equal to 5
```

```
## [1] TRUE TRUE FALSE
```

向量與純量的比較

```
x<5 # check if x=(6,4,5) is smaller than 5
```

```
## [1] FALSE TRUE FALSE
```

```
x>5 # check if x is bigger than 5
```

```
## [1] TRUE FALSE FALSE
```

```
x<=5 # check if x is not bigger than 5
```

```
## [1] FALSE TRUE TRUE
```

```
x>=5 # check if x is not smaller than 5
```

```
## [1] TRUE FALSE TRUE
```

向量與向量的比較

```
x<-c(6,4,5)
```

```
cc<-c(5,4,6)
```

```
x==cc #x和cc在向量相同位置的數值是否相同
```

```
## [1] FALSE TRUE FALSE
```


向量與向量的比較

- 向量長度為變數內純量的個數

```
x<-c(6,4,5)
```

```
length(x)#計算x的長度
```

```
## [1] 3
```

```
y<-c(1,2,3,4,5,6)
```

```
length(y)#計算y的長度
```

```
## [1] 6
```

向量與向量的比較

- 兩個向量的長度(length)必須相同

```
x1<-c(3,4,5)
```

```
x2<-c(3,4)
```

```
x1==x2 #x1和x2在向量相同位置的數值是否相同
```

```
## Warning in x1 == x2: 較長的物件長度並非較短物件長度的倍數
```

```
## [1] TRUE TRUE FALSE
```

Warning 警告程式語法可能不合法，但內定規則補正仍可執行。

- x2 比x1少一個值，以循環方式補值在x2 末再插入x2向量。若x1長度是x2的倍數，則可以插入整個x2補滿與x1相同的長度比較。
- 若非倍數，則無法完整插入，就有警告。此例x2末僅僅插入第一個值 3。

向量與向量的比較

```
x1<-c(3,4,5,4,3,6)
```

```
x2<-c(3,4)
```

x1==x2 #x1和x2在向量相同位置的數值是否相同

```
## [1] TRUE TRUE FALSE TRUE TRUE FALSE
```

- 以循環方式補值在x2 末再插入x2向量。
- 程式自動補值的方式如下:

```
c(x2,x2,x2)
```

```
## [1] 3 4 3 4 3 4
```

向量與向量的比較

```
# x<-c(6,4,5);cc<-c(5,4,6)
```

```
x!=cc
```

```
## [1] TRUE FALSE TRUE
```

```
x<cc
```

```
## [1] FALSE FALSE TRUE
```

```
x>cc
```

```
## [1] TRUE FALSE FALSE
```

向量與向量的比較

```
# x<-c(6,4,5);cc<-c(5,4,6)
```

```
x<=cc
```

```
## [1] FALSE TRUE TRUE
```

```
x>=cc
```

```
## [1] TRUE TRUE FALSE
```

課堂練習

- 執行比對x和y是否相同

$x \leftarrow c(5.8, 5.2, 5.7, 5.9)$

$y \leftarrow c(6.2, 4.3, 5.7, 4.9)$

不用程式，寫出以下算式的結果

$x > 5.7$

$y \geq x$

$x - y \geq 0$

$y - x == 0$

邏輯運算子

運算元	作用	優先次序	範例(x=6)	結果
!	反面	1	!(x==5)	TRUE
&	且	2	x>4 & x<7	TRUE
	或	3	x<3 x>5	TRUE

範例 1

`x < -6`

`y < -3`

`x == 6`

`## [1] TRUE`

`y == 6`

`## [1] FALSE`

`!(x == 6)`

`## [1] FALSE`

`!(y == 6)`

`## [1] TRUE`

範例 2

[1] FALSE

[1] TRUE

[1] TRUE

範例 3

```
x<-c(5.8, 5.2, 5.7, 5.9)
```

```
y<-c(6.2, 4.3, 5.7, 4.9)
```

```
!(x==5.7)
```

```
## [1] TRUE TRUE FALSE TRUE
```

```
!(y==5.7)
```

```
## [1] TRUE TRUE FALSE TRUE
```

範例 4

`x==y` #比較運算子

[1] FALSE FALSE TRUE FALSE

`!(x==y)` #邏輯運算

[1] TRUE TRUE FALSE TRUE

`x>y|x<y`

[1] TRUE TRUE FALSE TRUE

課堂練習 8

$x < -c(5.8, 5.2, 5.7, 5.9)$

$y < -c(6.2, 4.3, 5.7, 4.9)$

寫出以下算式的結果

$x == 5.7 \& y == 5.7$

$x > 5.7 \& y < 5.7$

$x < 5.7 | y < 5.7$

計算TRUE的個數

```
x<-c(6, 8, 8, 5, 7, 5, 6, 9, 9, 6)
```

有幾個數字等於8

```
x==8
```

```
## [1] FALSE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
sum(x==8)
```

```
## [1] 2
```

TRUE的位置

- 使用 which() 函數

```
## [1] FALSE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE
```

```
## [1] 2 3
```

課堂練習 9

`x<-c(5, 11, 4, 10, 7, 13, 10, 6, 14, 6, 9, 5, 14, 9, 8, 9, 10, 9, 6, 8)`

1. 有幾個數字等於9
2. 等於9的位置

應用問題

找極值(最大、最小)

```
x<-c(6, 8, 8, 5, 7, 5, 6, 9, 9, 6)
```

```
max(x) # 最大值
```

```
## [1] 9
```

```
min(x) # 最小值
```

```
## [1] 5
```

```
max_x<-max(x)
```

```
which(x==max_x) #method 1
```

```
## [1] 8 9
```

使用 `which.max()` 函數

```
which.max(x)
```

```
## [1] 8
```

- Determines the location, i.e., index of the **(first)** minimum or maximum of a numeric (or logical) vector.
- 不建議使用 `which.max()`

課堂練習 10

$x \leftarrow c(5, 11, 4, 10, 7, 13, 10, 6, 14, 6, 9, 5, 14, 9, 8, 9, 10, 9, 6, 8)$

1. 最大值與最大值的位置
2. 最小值與最小值的位置