

1. R的基本運算

R的基本運算有四則運算、數學函數運算和邏輯運算。

1-1 數學運算

數學運算子有加(+)、減(-)、乘(*)、除(/)、次方(^)、商數(%/%)和餘數(%%)七種。

1. 四則運算

- 四則運算的規則
 - 「先乘除後加減」
 - 「有括號先算括號」

Exercise 1

完成下列四則運算

```
4 + 2
4 - 2
4 * 2
4 / 2
```

Exercise 2

完成下列四則運算

```
2 * 3 + 4
2 * (3 + 4)
(2 * 3) + 4
```

2. 算數運算

Exercise 3

完成下列算數運算

```
7/2  
7%%2  
7%/%2
```

- 「次方順位在乘除之前」

Exercise 4

完成下列乘方運算

```
2**3  
2^3  
2**3 * 2^3 #這樣比較清楚
```

1-2 內置函數

可以用R語言中的數學函數完成各種初等數學的運算。

1. 常用數學函數

- `abs()` : 取絕對值
- `sqrt()` : 開根號
- `log()`、`log(x, a)` : 自然對數、一般對數
- `exp()` : 自然指數對數
- `factorial()` : 階乘函數 $n! = n * (n - 1) * \dots * 2 * 1$

Exercise 5

完成下列數學函數運算

```
abs(-5)
sqrt(8)

log(100)
log(100, 10)
exp(0)
exp(1)

factorial(7)
```

2. 四捨五入與近似運算

(1) 四捨五入運算

- `round(x, digits = k)`: 將x四捨五入到小數點第k位，指定想要的小數位數。
- `signif(y, n)`: 將y由左至右取n位有效數字，指定想要的有效位數。
- `options(digits = n)`: 用參數digits設定有效位數，最多可以顯示22位。

Exercise 6

完成下列四捨五入運算

```
round(98.562, digits = 2)
round(98.562, 2)

# 當小數點後的0的位數超過n時，輸出的結果為0：
a = 0.0001234567890 # 13位
round(a, 3)
round(a, 4)
```

Exercise 7

完成下列有效位數的運算

```
# Insert code for Exercise here
```

```
signif(87654.321, digits = 6)
```

```
signif(87654.321, digits = 3)
```

```
signif(12345.678, 6)
```

```
signif(12345.678, 3)
```

```
signif(12345.678, 5)
```

```
options(digits = 15)
```

```
pi
```

(2) 近似函數

- `floor(x)`：由下（左）取最接近整數，得到小於等於x的最近整數。
- `ceiling(x)`：由上（右）取最接近整數，得到大於等於x的最近整數。
- `trunc(x)`：直接取整數，無條件捨去小數。

Exercise 8

完成下列近似函數運算

```
floor(234.56)
```

```
floor(-234.56)
```

```
ceiling(234.56)
```

```
ceiling(-234.56)
```

```
trunc(234.56)
```

```
trunc(-234.56)
```

1-3 邏輯運算

1. 關係（比較）運算

- 大於(>)、大於等於(>=)、小於(<)、小於等於(<=)、等於(=)和不等於(!=)六種。

Exercise 9

完成下列邏輯運算

```
5 > 3
9 < 7
-5 > -3
-9 < (-7)
```

1-4. 無限大

1. 無窮大用 `Inf`，負無窮大用 `-Inf`。
- 正負無窮大是數學運算的結果，如某數除以0，不大用變數賦值的方式指派。

Exercise 10

完成下列無窮大運算

```
5 / 0
10 - (5/0)
```

2. 使用 `is.infinite(x)` 函數可以判斷x是否是正無限大或負無限大，判斷結果用邏輯值 `TRUE` 或 `FALSE` 表示。

Exercise 11

完成下列無窮大的判斷

```
is.infinite(10/0)
is.infinite(10-(10/0))
is.infinite(10-5)
```

3. 使用 `is.finite(x)` 函數可以判斷x是否是正有限大或負有限大，判斷結果用邏輯值 `TRUE` 或 `FALSE` 表示。

Exercise 12

完成下列有限值的判斷

```
is.finite(999)
is.finite(-99999)
is.finite(10/0)
is.finite(10-(10/0))
```

自我檢測

一、判斷題

() 1.有以下兩個命令：

```
x1 <- 9%%5
x2 <- 9%/%2
```

以上兩命令被執行後，x1和x2的值同為4。

() 2.有以下命令：

```
x <- round(1560.998, digits = -2)
```

以上命令被執行後，x的值是1600。

() 3.有以下命令：

```
x <- factorial(3)
```

以上命令被執行後，x的值是8。

() 4.有以下兩個命令：

```
x <- 999/0
is.infinits(x)
```

以上命令被執行的結果是FALSE。

() 5. 有以下兩個命令：

```
x <- NA + 999  
is.na(x)
```

以上命令被執行的結果是TRUE。

二、單選題

() 1 · 以下命令會得出哪種數值結果？

```
36**0.5
```

A.[1]18 B.[1] 6 C.[1]9 D.[1]3

() 2 · 以下命令會得出哪種數值結果？

```
signif(4678.778, 6)
```

A.[1]5678.78 B.[1] 5678.79 C.[1]5678.77 D.[1]5678.778

() 3 · 以下命令會得出哪種數值結果？

```
floor(789.789)
```

A.[1]778.9 B.[1] 789.789 C.[1]789 D.[1]790

() 4 · 以下命令會得出哪種數值結果？

```
round(pi, 2)
```

A.[1]3.1415926 B.[1] pi C.[1]3.14 D.[1]3

() 5 · 以下命令會得出哪種數值結果？

```
x <- Inf/1000
```

A.[1]0 B.[1] Inf C.[1]NA D.NaN

2. R的變數

數值型別的資料可分為常數和變數兩大類。

在程式運行過程中，其值和類型不能被改變的量稱為常數，其值和類型能被改變的量稱為變數。

變數的識別字（即變數名）、變數的值和變數的資料類型稱為變數的三要素。

顯然，變數是R語言的物件之一。

2-1 變數賦值

1. 賦值就是將符號右邊的值指派給符號左邊變數名稱的過程。

- 在R語言中，變數即資料物件，對變數賦值用 `<-`。
 - 可用快速鍵 `Alt + -`（減號）生成。
- 變數賦值後，輸入變數物件的名字，就可以顯示其內容。

例如：

```
x <- 3      # 請用這個賦值
3 -> x
y <- z <- 6
x
y
z
```

此時，變數x的值3,變數y和z的值為6。

2. 可用`ls()`函數查看當前系統中的變數。

```
ls()
```

3. 要刪除記憶體中的物件，可以使用`rm()`函數。

- `rm(x)` 會刪除物件x。
- `rm(x, y)` 會刪除物件x和y。
- `rm(list = ls())` 則會刪除記憶體中的所有物件。

4. 變數命名規則

- 變數名只能包含字母、數位、底線（`_`）和點號（`.`）。
- 開頭不能是數字，必須是英文字母或點號（`.`）。

- 不能用R語言的保留字作變數名；
 - 例如：`function`, `if`, `else`, `while`, `for`, `NA`, `next`, `TRUE` 等
- R語言對英文大小寫字母敏感
 - 例如：`bucket`與`Bucket`、`BUCKET`被視為是三個不同的變數名。
- 不建議用中文命名。

2-2 資料的類型

(1) 常見資料類型

- 數值型 (numeric)：數值型變數還可以再劃分為整數 (integer)、單精確度和雙精度 (double) 三種。
- 邏輯型 (logical)：邏輯型變數的取值只能是TRUE (或T)，或者是FALSE (或F)。
- 字元型 (character)：字元型變數是夾在雙引號" "或單引號' '之間的字串。

(2) 其他資料類型

- 複數型 (complex)：複數型變數具有 $a+bi$ 的形式。
- 原味型 (raw)：原味型變數就是以二進位形式保存的變數。

(3) 特殊資料類型

- `Inf`，其意義為無窮大。
 - 例如， $1/0$ 的結果為 `Inf`
 - 與它相反意義的變數為 `- Inf`，表示負無窮。
- `NaN` (Not a Number)，其意義為不確定。
 - 例如， $0/0$ 的結果為 `NaN`。
- `NA` (Not Available)，其意義為無法得到或缺失時，就給相應的位置賦予 `NA`
 - 與 `NA` 變數的任何運算，其結果均為 `NA`。
- `NULL`，其意義是空的變數。

2-3 資料類型的判別與轉換

1. 資料類型的判別

R語言中有一些針對資料型態或資料結構判斷的指令，這些指令大部分是以`is*`開頭。

例如，`is.numeric()`，`is.character()`，`is.logical()`，`is.complex()`，`is.infinite()`，`is.finite()`，`is.nan()`，`is.na()`，`is.null()`。

2. 資料類型的轉換

當發現資料類型不符合實際情況時，需要對其進行轉換，否則無法做下一步的處理和運算。

在數據的預處理階段，經常會碰到關於資料類型轉換的問題，所使用的函數為`as.*`的形式。

例如，`as.numeric()`，`as.character()`，`as.logical()`，`as.complex()`，`as.null()`。

Exercise 13

變數可以相互轉換示例。

```
## Input Data
x <- 3          # 數字

y <- as.character(x) # 轉文字
y

is.numeric(y)    # 確認是否是文字
```

2-4 R變數的特點

R語言變數有著跟其他程式設計語言不同的兩大特點。

特點一：

在一般程式設計語言中，變數必須遵循“先定義後使用”的原則；但在R語言中，變數在使用時可以不先定義，直接對其賦值。

Exercise 3-3

已知圓的半徑R為5，求圓面積。

```
## Input Data
R <- 5
## Process Data
Area = 3.14 * R^2
## Output Data
Area
```

特點二：

R是動態賦值語言，變數的類型可以隨時改變。

Exercise14

變數的類型可以隨時改變示例。

```
## 動態宣告變數
x <- 5
x
x <- "abc"
```

自我檢測

一、判斷題

- () 1. 在R語言中，變數在使用時可以不先定義，直接對其賦值。
- () 2. 在R語言中，變數的類型不可以隨時改變。
- () 3. 在R語言中，變數的類型除常見的之外，還有複數型和原味型。
- () 4. R語言對英文大小寫字母不敏感，所以bucket與Bucket、BACKET會視為是三個相同的變數名。
- () 5. 在R語言中，負數可以在複數意義下作開方運算。

二、單選題

- () 1. 下列哪一個函數可以在Console視窗列出所有變數資料？
A. ls() B. wm() C. q() D. getwd()

() 2. 下列哪一個是 R 語言不合法的變數名稱？

A. x3 B. x.3 C. .x3 D. 3.x

() 3. 下列哪一個不是 R 語言的等號符號？

A. # B. = C. <- D. ->

() 4. 執行下列命令後，y 的值是下面哪一個？

```
x <- 3  
y <- as.character(x)
```

A. 3 B. -3 C. "3" D. 無法確定

() 5. 若再執行下列命令後，輸出結果是下面哪一個？

```
is.numeric(y)
```

A. [1]3 B. [1]"3" C. [1]TRUE D. [1] FALSE