

# 向量

---

向量 ( Vector ) 是由相同基本類型的元素所構成的數列，是R語言中的最常用的物件，同時也可以作為R語言中最基本的資料登錄方式。

## I. 建立向量

---

建立向量物件的函數主要有 `seq()`、`c()`、`rep()` 等，稍後分別介紹。

### 1. 規則向量：`seq()` 函數

1. `seq()` 函數可用于建立一個規則型的數值向量物件，它的使用格式如下：

```
seq(from, to, by = width, length = numbers)
```

#### 2. 公差為正負1的等差數列

- 指令為 `a:b`
  - 若a比b小，數列遞增，公差為1
  - 若a比b大，數列遞減，公差為-1
  - a、b不一定要是整數，也可以是小數
  - 數列的最後一項不一定剛好是終止值

### Exercise 1

使用 `seq()` 函數建立規則型的數值向量物件。

```
seq(1, 9)                # seq(from = 1, to = 9)
seq(1, 9, by = 2)         # seq(from = 1, to = 9, by = 2)
seq(1.5, 4.5, by = 0.5)   # seq(from = 1.5, to = 4.5, by
= 0.5)
seq(1, 9, length = 5)     # 若未給by，則數列等距離切割

x <- 1:7
x <- 7:1
x <- 1.5:7.8
x <- 7.8:1.5
x <- 4.3:-3.5
```

## 2. 不規則向量：函數 `c()`

`c()` 函數中的 `c` 是 concatenate 或 combine 的縮寫，用於連接向量物件元素。

- 把一堆數字連接起來，成為一個數列。
- 可以用在純量與純量、純量與向量，以及向量與向量之間的連結。

### Exercise 2

使 `c()` 函數拼接輸出一個數值向量物件。

```
x <- c(1, 3, 7, 4, 9)
x
```

## 3. 重複向量： `rep()` 函數

如果向量物件內某些元素是重複的，則可以使用 `rep()` 函數建立這種類型的向量物件，它的使用格式如下。

```
rep(x, times, each, length)
```

### Exercise 3

使用 `rep()` 函數建立向量物件重複元素的應用。

```
rep(5, 5)
rep(5, times = 5)

rep(1:5, 3)
rep(1:5, times = 3)
rep(1:5, each = 3)
rep(1:5, length = 3)
rep(1:5, times = c(5,4,3,2,1)) # 設定每個元素複製的次數

rep(1:3, times = 3, each = 2)
rep(1:3, times = 3, length = 8)
rep(1:3, each = 2, length = 8)
```

## II. 向量的數學運算

- 向量物件的數學運算函數有：常見運算函數、計算元素積函數、累積運算函數、差值運算函數、排序函數、計算向量物件長度函數和基本統計函數等。

把向量代入函數中，相當於把一整批數值代入函數運算，傳回的也是整批運算結果，即結果向量。

### 1.常見運算：和與平均、最大與最小

- `sum()`: 計算所有向量物件元素和的函數
- `mean()`: 計算所有向量物件元素平均值的函數
- `max()`: 計算所有向量物件元素最大值的函數
- `min()`: 計算所有向量物件元素最小值的函數

### Exercise 4

計算向量物件元素的總和、最大值、最小值和平均值示例。

```
x <- c(2, 6, 8, 12, 10)
sum(x)
max(x)
min(x)
mean(x)
```

## 2. 向量元素積

- `prod()`: 階乘，計算所有向量物件元素積的函數

### Exercise 5

用 `prod()` 函數計算5的階乘和10的階乘示例。

```
prod(1:5)
prod(1:10)
```

## 3. 累積運算函數: 加與乘、最大與最小

- `cumsum()`: 所有向量元素累加和
- `cumprod()`: 所有向量元素累積積
- `cummax()`: 從起點到該元素位置 ( 累積分量 ) 的最大值
- `cummin()`: 從起點到該元素位置 ( 累積分量 ) 的最小值

### Exercise 6

計算某向量物件元素的累積和、累積積、累積最大值、累積最小值示例。

```
x <- c(10, 5, 9, 15, 7, 11)
cumsum(x)
cumprod(x)
cummax(x)
cummin(x)
```

## 4. 差值運算函數

- `diff()`: 計算各元素與下一個元素的差

### Exercise 7

用差值運算函數計算上面向量的差值。

```
x <- c(10, 5, 9, 15, 7, 11)
diff(x)
```

## 5. 排序函數

- `sort()`：把資料按照由小到大的順序排列。
  - `sort(x, decreasing = FALSE)`
- `rank()`：排名。
  - 返回的向量物件元素是原向量物件按從小到大排序後所得向量物件的位置。
- `rev()`：把原始資料的順序倒過來。
- `order()`：位置指標。
  - 把`order()`後的結果代入原向量的位置指標中，效果相當於`sort()`。

### Exercise 8

分別用`sort()`、`rank()`、`rev()`函數對向量元素進行正排、逆排、正排中元素排位和顛倒排序示例。

```
x <- c(10, 5, 9, 15, 7, 11)
sort(x)
sort(x, decreasing = TRUE)
rank(x)
rev(x)

order(x) # sort後此排序向量 ( 5 7 9 10 11 15 ) 的位置指標
x[order(x)]
x[order(x, decreasing = TRUE)]
```

## 6. 計算向量物件長度的函數

- `length()`：計算向量物件長度，即向量物件元素的個數。

### Exercise 9

用計算向量物件長度的函數計算物件元素的個數示例。

```
x <- c(10, 5, 9, 15, 7, 11)
length(x)
```

## 7. 基本統計函數

- `var()`：用於計算樣本的變異數（或稱方差）。
- `sd()`：用於計算樣本的標準差。

### Example 10

基本統計函數示例。

```
sd(c(11, 15, 18))  
var(14:16)
```

### Example 11

對向量的綜合運算示例。

```
x <- 1:5  
y <- 6:10  
z <- c(x, y, 11, 12)  
z  
z <- (x+y)  
z  
min(z)  
range(z)  
sum(z)  
var(z)
```

## 8. 向量的數學運算

- 數學運算子和內置數學運算函數，在向量中一樣可以運算。
  - 數學運算子有加(+)、減(-)、乘(\*)、除(/)、次方(^)、商數(%%)和餘數(%%)七種。
  - 數學運算函數有絕對值(abs)、開根號(sqrt)、自然對數(log)、自然指數(exp)等。
- 當兩個長度不同的向量做數學運算時，較短向量會自動循環延伸到相同長度進行運算，稱廣播運算。

## Example 12

對向量的綜合運算示例。

```
x <- c(1, 3, 5, 7)
y <- c(2, 4, 6, 8)

x + y
x - y
x * y
x / y
x ^ y
y %/% x
y %% x

x <- 1:2 # # 長度是倍數
y <- 1:6
x + y
x - y

x <- 1:2 # 長度非倍數
y <- 1:5
x + y
x - y

sqrt(x)
```