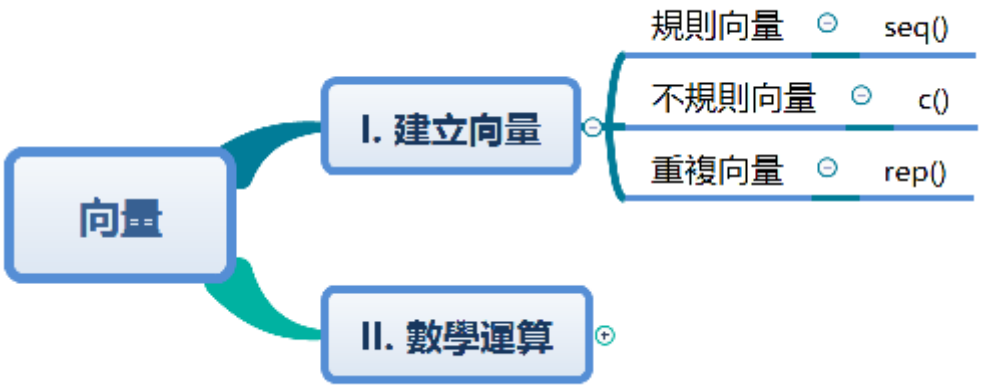
向量與矩陣

Week 4 複習



I. 建立向量

=> 三個常見的建立向量物件的函數

- 不規則向量：`c()`函數
- 規則向量：`seq()`函數
- 重複向量： `rep()`函數

```
# 數值向量
(x <- c(1, 3, 7, 4, 9))
(y1 <- seq(from = 1, to = 9, by = 2))
(y2 <- seq(1, 9, length = 5))
(y3 <- 1:9)
(z1 <- rep(3, times = 5))
(z2 <- rep(c(1,3), each = 2))
(z3 <- rep(1:3, times = c(1,2,3))
```

=> Question: 請問

- `rep(c(1,3), times = 3, each = 2, length = 8))`是多少？
- 又，`rep(1:3, rep(2:3))`是多少？

~這學期不像上學期，只是讓同學知道寫程式不可怕！

同樣是中學程度的東西，如果我們真要將程式應用在中學數學時，通常是應用題，應用就會有變化。當應用上出問題時，七到八成的問題是同學中小學的數學好不好，兩到三成出在同學對程式的基本語法熟不熟。通常，程式是最後一里路，有問題時，先要問的是「專業知識與普通常識」行不行，真正是程式的問題的通常都可以克服（我們不是資管資工系，用的與教的都不難，若真是程式有問題，通常是不熟，也不願花時間想一下 => 運算思維與程式設計），只是因為程式寫不出來，所以所有的問題都歸給程式，甚至說自己沒有天賦、沒有興趣、用不到等等。

II. 向量的數學運算

=> 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

```
x <- 1:5
sum(x)
max(x)
min(x)
mean(x)
```

Week 5 向量的數學運算

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。

- 如`5! = 5*4*3*2*1`

```
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 <- seq(1,9,2)
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

=> 同學很容易混淆下面各排序函數的含意

- `sort()`: 把資料按照由小到大的順序排列。
  - 預設是sort(x, decreasing = FALSE)
- `rank()`：排名，第一名、第二名...。
  - 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(reverse)。
- `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)   # 5  7  9 10 11 15
sort(x, decreasing = TRUE) # 15 11 10  9  7  5
rank(x)
rev(x)

order(x)  # 由小到大排序後（5  7  9 10 11 15），此向量在原向量的位置指標2 5 3 1 6 4
x[order(x)] # 即x[c(2,5,3,1,6,4)] => 5  7  9 10 11 15
x[order(x, decreasing = TRUE)] # 即x[c(4,6,1,3,5,2)]
```

6. 物件長度

- `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. 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. 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。

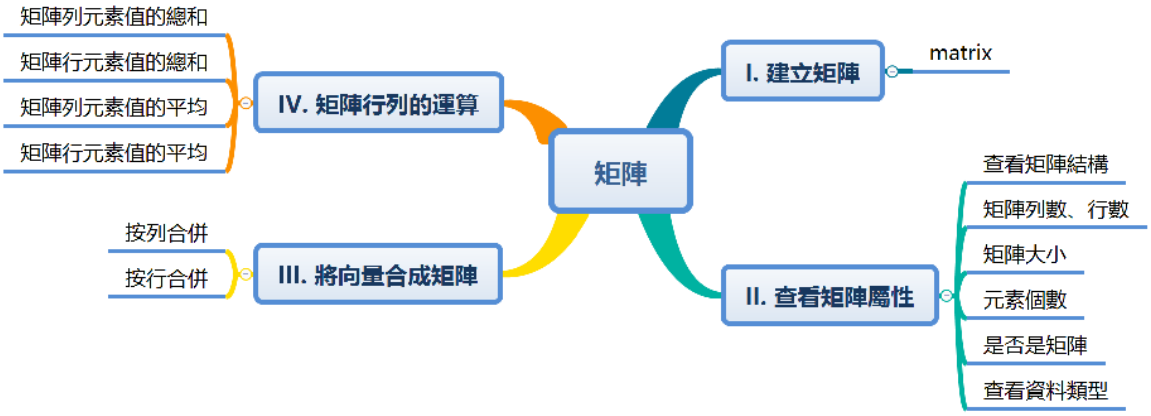
```
x <- c(1, 3, 5, 7)
y <- c(2, 4, 6, 8)

x + y
```

```r
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)
```

考試題庫：請自行練習舊講義上的是非選擇題與課本習題

矩陣

R語言中，除向量之外，矩陣也是資料登錄和計算的最簡單形式。



1. 建立矩陣

1. 若矩陣A有m列n行，則稱矩陣A是一個大小為$m * n$階的矩陣，總共有$m * n$個元素，每個元素以$a_{i,j}$表示。

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \dots & a_{mn} \end{bmatrix} \tag{1}$$

2. 建立矩陣使用`matrix()`函數，其使用格式如下：

```
matrix(data，nrow = m，ncol = n，byrow = FALSE，dimnames = NULL)
```

- data：數據。
- nrow：預計行的數量m。
- ncol：預計列的數量n。
- byrow：邏輯值。
  - 默認是FALSE，即按行給資料，一行一行直向排列呈現資料。
  - 若要改成按列給資料，一列一列橫向排列呈現資料，請加入參數`byrow = TRUE`。
- dimnames：矩陣的屬性。

Exercise 1

建立一個數據為1:12，用4列(nrow = 4)先填行的矩陣(a.matrix)。

$$\begin{bmatrix} 1 & 5 & 9 \\ 2 & 6 & 10 \\ 3 & 7 & 11 \\ 4 & 8 & 12 \end{bmatrix} \tag{2}$$

```
a.matrix <- matrix(1:12, nrow = 4) # A <- matrix(1:12, nrow = 4, ncol = 3, byrow = FALSE)
a.matrix

a.matrix<- matrix(1:10, nrow = 4) # 提供的向量元素不夠用時，R會自動將元素重複循環
```

Exercise 2

建立一個數據為1:12，用4列(nrow = 4)先填列的矩陣(b.matrix)。

運算思維：先寫出來矩陣長什麼樣子，再想指令。

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \end{bmatrix} \tag{3}$$

```
b.matrix <- matrix(1:12, nrow = 4, byrow = TRUE) # 一列一列寫元素
b.matrix
```

2. 查看矩陣屬性

- `str()`：查看矩陣結構
- `nrow()`：矩陣列數
- `ncol()`：矩陣行數
- `dim()`：矩陣的大小，含行數和列數
- `length()`：矩陣元素個數
- `is.matrix()`：是否是矩陣
- `class()`：查詢資料結構

Exercise 3

查看矩陣物件屬性函數應用示例。

```
str(a.matrix)
str(b.matrix)

nrow(a.matrix)
nrow(b.matrix)

ncol(a.matrix)
ncol(b.matrix)

dim(a.matrix)
dim(b.matrix)
```

```
length(a.matrix)
length(b.matrix)

is.matrix(a.matrix)
is.matrix(b.matrix)

class(a.matrix)
class(b.matrix)
```

3. 將向量合成矩陣

- 將向量合成矩陣的函數有`rbind()`和`cbind()`兩個。

1. 長度相同的向量，可以合併成一個矩陣。

- `rbind()函數`：將兩個或多個向量合成矩陣，每個向量各自占一列。
- `cbind()函數`：將兩個或多個向量合成矩陣，每個向量各自占一行。

Exercise 4

用rbind()函數將兩個向量構成矩陣示例。

$$\begin{bmatrix}7 & 11 & 15\end{bmatrix}和\begin{bmatrix}5 & 10 & 9\end{bmatrix}成\begin{bmatrix}7 & 11 & 15\\5 & 10 & 9\end{bmatrix} \tag{4}$$

```
v1 <- c(7, 11, 15)
v2 <- c(5, 10, 9)
a1 <- rbind(v1, v2)
a1
```

Exercise 5

用rbind()函數將矩陣和向量構成新矩陣示例。

$$\begin{bmatrix}7 & 11 & 15\\5 & 10 & 9\end{bmatrix}和\begin{bmatrix}3 & 6 & 12\end{bmatrix}成\begin{bmatrix}7 & 11 & 15\\5 & 10 & 9\\3 & 6 & 12\end{bmatrix} \tag{5}$$

```
v3 <- c(3, 6, 12)
a2 <- rbind(a1, v3)
a2
```

Exercise 6

用cbind()函數將兩個向量構成矩陣示例。

$$\begin{bmatrix} 7 \\ 11 \\ 15 \end{bmatrix} 和 \begin{bmatrix} 5 \\ 10 \\ 9 \end{bmatrix} 成 \begin{bmatrix} 7 & 5 \\ 11 & 10 \\ 15 & 9 \end{bmatrix} \tag{6}$$

```
v1 <- c(7, 11, 15)
v2 <- c(5, 10, 9)
a3 <- cbind(v1, v2)
a3
```

Exercise 7

用cbind()函數將兩個向量和一個矩陣構成新矩陣示例。

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} 和 \begin{bmatrix} 11 \\ 12 \\ 13 \end{bmatrix} 和 \begin{bmatrix} 21 & 24 \\ 22 & 25 \\ 23 & 26 \end{bmatrix} 成 \begin{bmatrix} 1 & 11 & 21 & 24 \\ 2 & 12 & 22 & 25 \\ 3 & 13 & 23 & 26 \end{bmatrix} \tag{7}$$

```
cbind(1:3, 11:13, matrix(21:26, nrow = 3))
```

2. 大小相同的矩陣可合併成新矩陣

- rbind()：按列合併矩陣。
  - 新矩陣列數增多。
  - 被併接矩陣的行數必須相等。
- cbind()：按行合併矩陣。
  - 新矩陣行數增多。
  - 被併接矩陣的列數必須相等。

Exercise 8

用cbind()函數將兩個矩陣構成新矩陣示例。

$$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} 和 \begin{bmatrix} 10 & 13 & 16 \\ 11 & 14 & 17 \\ 12 & 15 & 18 \end{bmatrix} 成 \begin{bmatrix} 1 & 4 & 7 & 10 & 13 & 16 \\ 2 & 5 & 8 & 11 & 14 & 17 \\ 3 & 6 & 9 & 12 & 15 & 18 \end{bmatrix} \tag{8}$$

```r
x1 <- matrix(c(1:9), nrow = 3)
x1

x2 <- matrix(c(10:18), nrow = 3)
x2

cbind(x1, x2)
```

Exercise 9

用rbind()函數將兩個矩陣構成新矩陣示例。

$$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} 和 \begin{bmatrix} 10 & 13 & 16 \\ 11 & 14 & 17 \\ 12 & 15 & 18 \end{bmatrix} 成 \begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \\ 10 & 13 & 16 \\ 11 & 14 & 17 \\ 12 & 15 & 18 \end{bmatrix} \quad (9)$$

```r
rbind(x1, x2)
```

======

4矩陣的運算

1. 矩陣行（Row）和列（Column）的運算函數有四個:

- `rowSums()`: 矩陣列元素值的總和
- `colSums()`: 矩陣行元素值的總和
- `rowMeans()`:矩陣列元素值的平均值
- `colMeans()`:矩陣行元素值的平均值

Exercise 10

計算矩陣的列中元素值的總和及平均值示例。

$$\begin{bmatrix} 7 & 8 & 6 \\ 12 & 8 & 9 \end{bmatrix} 和 \begin{bmatrix} 11 & 9 & 12 \\ 15 & 7 & 12 \end{bmatrix} 成 \begin{bmatrix} 7 & 8 & 6 & 11 & 9 & 12 \\ 12 & 8 & 9 & 15 & 7 & 12 \end{bmatrix} \quad (10)$$

```r
v1 <- c(7, 8, 6, 11, 9, 12)
v2 <- c(12, 8, 9, 15, 7, 12)
a <- rbind(v1, v2)
a

rowSums(a)
rowMeans(a)
```

Exercise 11

計算矩陣的行中元素值的總和及平均值示例。

$$\begin{bmatrix} 7 \\ 8 \\ 12 \end{bmatrix} \text{和} \begin{bmatrix} 8 \\ 9 \\ 15 \end{bmatrix} \text{成} \begin{bmatrix} 7 & 8 \\ 8 & 9 \\ 12 & 15 \end{bmatrix} \tag{11}$$

```r
v1 <- c(7, 8, 12)
v2 <- c(8, 9, 15)
b <- cbind(v1, v2)
colSums(b)
colMeans(b)
```

2. 矩陣的四則運算

- 矩陣的維度大小相同（行數與列數一樣），兩個矩陣可以做加、減運算。

Exercise 12

矩陣加減運算示例。

$$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} +/- \begin{bmatrix} 10 & 13 & 16 \\ 11 & 14 & 17 \\ 12 & 15 & 18 \end{bmatrix} =? \tag{12}$$

Exercise 13

矩陣加減運算示例。

```r
x1 <- matrix(c(1:9), nrow = 3)
x2 <- matrix(c(10:18), nrow = 3)

x1 + x2    # 維度相同的矩陣相加
x1 - x2    # 維度相同的矩陣相減
```

(Optional)補充說明：矩陣相乘、逆矩陣、與行列式

- `%*%`：矩陣相乘，第一個矩陣的行數和第二個矩陣的列數相同。
  - $A_{m*n} * B_{n*m} = C_{m*m}$

Exercise 14

矩陣相乘運算示例。

$$\begin{bmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{bmatrix} * \begin{bmatrix} 10 & 13 & 16 \\ 11 & 14 & 17 \\ 12 & 15 & 18 \end{bmatrix} = \begin{bmatrix} 138 & 174 & 210 \\ 171 & 216 & 261 \\ 204 & 258 & 1312 \end{bmatrix} \quad (13)$$

以C矩陣第一個元素（第一列第一行）為例， 為A矩陣第一列元素和B矩陣第一行元素相成之和。即
$138 = 1 * 10 + 4 * 11 + 7 * 12 = 10 + 44 + 84$

```r
x1 <- matrix(c(1:9), nrow = 3)
x2 <- matrix(c(10:18), nrow = 3)

x1 %*% x2      # 矩陣相乘
```

- 矩陣沒有除法，通常是指求逆矩陣(inverse)。
  - 若$A * A^{-1} = I$，則A^{-1}稱為A矩陣的逆矩陣。
  - 在R語言，可用`solve(A)`函數求算逆矩陣

~逆矩陣的數學推導繁瑣，若高中沒學過同學，待日後有機會再學。

Exercise 15

求算逆矩陣運算示例。

```
a1 <- c(3, 2, 5)
a2 <- c(2, 3, 2)
a3 <- c(5, 2, 4)
(A <- rbind(a1, a2, a3))
X <- solve(A)
X
```

- 矩陣用`det()函數`可以求算對應的行列式(determinant)

Exercise 16

求算矩陣的行列式運算示例。

```
det(A)  # A矩陣的行列式
```

~微積分與線性代數在進階數學和統計中很重要，想要做數據分析者，請學線性代數。