第 3 章: 資料物件

3: Data Object

R 是以物件導向爲主的程式語言, 在 R 中, 資料或運算指令以具有名稱的 物件 (object), 形式儲存, 資料物件可以是 向量 (vector), 矩陣 (matrix), 陣列 (array), 列表 (Lists), 或 資料框架 (data frames) 等. 在 R 中, 資料分析基本上是 產生資料物件, 對物件命名, 使用函式對物件運算操作. 透過指令, 很容易地對物件進行統計分析與統計繪圖. 上一章討論向量的基本操作, 本章進一步討論 R 的其他資料物件, 包含矩陣 (matrix), 陣列 (array), 列表 (Lists), 或 資料框架 (data frames) 等.

3.1 矩陣物件 Matrix

矩陣 (matrix) 物件由包含相同的元素 (模式, mode) 組成的 2-維度 (2-dimension) 資料物件, 矩陣具有 維度 (dimension) 之屬性, 可以使用函式 `dim()` 檢視. 可以將矩陣視爲一個向量具二維結構, 也可以將 矩陣 視爲一個向量具 2-維度 的陣列 (array).

3.1.1 矩陣函式 matrix()

使用者若要輸入一個簡單的矩陣資料, 列 × 欄 (列 × 行), 或希望以矩陣形式儲

存資料, 可以用 函式 matrix().

```
matrix(data = NA, nrow = 1, ncol = 1, byrow = FALSE, dimnames = NULL)
```

其中引數爲

- nrow = r 設定 “列數” (row numbers).
- ncol = c 設定 “欄數” 或 “行數” (column number).
- byrow = FALSE: 在 R 中的自動設定, 矩陣資料是以欄 (行) 位 (column) 優先填滿. 要改變設定, 可改成 byrow = TRUE.
- dimnames = obj.list 輸入列表設定列位名與欄位名.

設定 列數 nrow = r 或 欄數 ncol = c, r, c 爲正整數. R 內自動設定矩陣物件的元素輸入, 是以一整個 欄位/行位 (column) 優先填滿, 要改變矩陣資料的輸入設定, 可在函式 matrix() 內的引數加入 byrow = T.

使用 函式 dim() 可以回傳具有維度數屬性的資料物件之維度大小. 在 R 中顯示矩陣物件, [m,] 出現在某特定元素左方時, 表示某特定元素在該矩陣物件之第 m 列 (row) 的位置; [, n] 出現在某特定元素上方時, 表示某特定元素在該矩陣物件之第 n 欄 (column) 的位置.

```
> #### matrix()
> ## numeric
> x.mat <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2) # one row first
> x.mat
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
> dim(x.mat)
[1] 2 3
> y.mat <- matrix(c(1, 2, 3, 4, 5, 6), ncol = 2)
> y.mat
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
> dim(y.mat)
[1] 3 2
> z.mat <- matrix(c(1, 2, 3, 4, 5, 6), nrow = 2, byrow = T)
```

```
> z.mat
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
> dim(z.mat)
[1] 2 3
> p.mat <- matrix(c(1, 2, 3, 4, 5, 6), ncol = 2, byrow = T)
> p.mat
     [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
> dim(p.mat)
[1] 3 2
> w.mat <- matrix(c(1:18), nrow = 3)
> w.mat
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    4    7   10   13   16
[2,]    2    5    8   11   14   17
[3,]    3    6    9   12   15   18
> dim(w.mat)
[1] 3 6
> #
> # character
> x.vec <- c("a", "b", "c", "d", "e", "f")
> x.vec
[1] "a" "b" "c" "d" "e" "f"
> dim(x.mat)
[1] 2 3
> y.mat <- matrix(x.vec, nrow = 2, ncol = 3) # byrow = F
> y.mat
     [,1] [,2] [,3]
[1,] "a"  "c"  "e"
[2,] "b"  "d"  "f"
> dim(y.mat)
[1] 2 3
> y.mat <- matrix(x.vec, nrow = 2, ncol = 3, byrow = T)
> y.mat
     [,1] [,2] [,3]
[1,] "a"  "b"  "c"
[2,] "d"  "e"  "f"
> dim(y.mat)
[1] 2 3
> # dim()
> m.vec.mat <- 1:10
> dim(m.vec.mat) <- c(2, 5)
> m.vec.mat
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    3    5    7    9
[2,]    2    4    6    8   10
```

```
> dim(m.vec.mat)
[1] 2 5
```

3.1.2 矩陣的命名

矩陣的命名, 包含 欄位名 (column name) 與 列位名 (row name), 可以使用函式 dimnames() 分別給予矩陣命名. 可以用函式 dimnames() 同時檢視 matrix 之列位名與欄位名. 若要對讀取或命名 矩陣 的 列位名 (row name) 或 欄位名 (column name), 也可以用函式指令 rownames() 與 colnames().

```
> ## matrix dimnames
> x.mat <- matrix(1:6, nrow = 2, ncol = 3)
> dimnames(x.mat) <- list(c("A1", "A2"),
                          c("B1", "B2", "B3"))
> x.mat
   B1 B2 B3
A1  1  3  5
A2  2  4  6
> dim(x.mat)
[1] 2 3
> dimnames(x.mat)
[[1]]
[1] "A1" "A2"

[[2]]
[1] "B1" "B2" "B3"

> rownames(x.mat)
[1] "A1" "A2"
> colnames(x.mat)
[1] "B1" "B2" "B3"
> #
> m.mat <- matrix(c(1, 2, 3, 11, 12, 13),
                  nrow = 2, ncol = 3, byrow = TRUE,
                  dimnames = list(c("row1", "row2"),
                                  c("C1", "C2", "C3")))
> m.mat
     C1 C2 C3
row1  1  2  3
row2 11 12 13
> dim(m.mat)
[1] 2 3
> dimnames(m.mat)
[[1]]
```

```
[1] "row1" "row2"

[[2]]
[1] "C1" "C2" "C3"

> rownames(m.mat)
[1] "row1" "row2"
> colnames(m.mat)
[1] "C1" "C2" "C3"
```

3.1.3 矩陣的下標與索引 Matrix Index

矩陣的 下標 或 索引 (index) 操作, 如同向量的下標與索引操作, 矩陣具有 2-維度下標向量, 個別下標向量可以輸入正整數, 負數, 整數向量, 欄位名等等. 例如, 可以使用 "中括號 []" `matrix.name[i, j]` 可存取矩陣中的第 $[i, j]$ 元素; `matrix.name[i, ]` 可存取矩陣中的第 i 列 (ith row), `matrix.name[ , j]` 可存取矩陣中的第 j 欄 (ith column).

```
> ## matrix index
> x.mat <- matrix(c(1:12), 3, 4)
> x.mat
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
> x.mat[2,3] <- 30
> x.mat
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5   30   11
[3,]    3    6    9   12
> x.mat[2, ]
[1]  2  5 30 11
> x.mat[ ,3]
[1]  7 30  9
> x.mat[c(1,3), c(2,4)]
     [,1] [,2]
[1,]    4   10
[2,]    6   12
> #
> m.mat <- matrix(c(1, 2, 3, 11, 12, 13),
                  nrow = 2, ncol = 3, byrow = TRUE,
                  dimnames = list(c("row1", "row2"),
```

```
                                  c("C1", "C2", "C3")))
> m.mat
     C1 C2 C3
row1  1  2  3
row2 11 12 13
> m.mat[ , c("C1", "C2")]
     C1 C2
row1  1  2
row2 11 12
> m.mat[c("row2"), ]
C1 C2 C3
11 12 13
> m.mat[c("row1"), c("C1", "C3")]
C1 C3
 1  3
```

矩陣下標與索引若僅選取 1 列 或 1 欄, 則會維度縮減產生向量, 若仍要產生矩陣, 則可加入參數 `drop = FALSE`.

```
> ## dimension reduction
> x.mat <- matrix(1:8, 2, 4)
> x.mat[1, ] # reduces to a vector
[1] 1 3 5 7
> x.mat[1, , drop = FALSE] # remains as a matrix
     [,1] [,2] [,3] [,4]
[1,]    1    3    5    7
```

3.1.4 向量與矩陣的合併: rbind() 與 cbind() 函式

在統計分析中, 許多時候必須合併向量或矩陣, 使用函式 `rbind()` 與函式 `cbind()`, 使用者可以分別依照 列 (row) 或 欄 (column) 來合併向量或矩陣.

在 R 中向量並不具有 沒有維度 (no dimension), 例如, 向量可以視爲 $1 \times k$ 的 向量/矩陣, 也可以視爲 $k \times 1$ 的 向量/矩陣, 但是, 當 向量 與 其它 向量/矩陣 進行運算時, 向量 會受到與其進行運算的矩陣物件影響, R 如何處理 向量 與 其它 向量/矩陣 進行運算並沒有清楚的規則, 有時視爲 $1 \times k$ 若任由 R 的內在設定, 則將會有意想不到的運算結果, 因此在進行線性代數相關的計算, 若要避免混淆與錯誤, 初學者可將數學的 1-維度 k-個元素的向量, 重新定義成 R 的 $1 \times k$ 的矩陣 或是 $k \times 1$ 的矩陣, 然後再進行線性代數相關的計算. 同樣, 合併不同 列位數 (row number) 或 欄

位數 (column number) 的向量或矩陣, 因爲使用 recycle 原則必須小心.

```
> ## matrix cbind() and rbind()
> x.vec <- c(1, 2, 3)
> y.vec <- c(8, 9, 10)
> rbind(x.vec, y.vec)  # vector as row vector
      [,1] [,2] [,3]
x.vec    1    2    3
y.vec    8    9   10
> cbind(x.vec, y.vec)  # vector as col vector
     x.vec y.vec
[1,]     1     8
[2,]     2     9
[3,]     3    10
> #
> x.mat <- matrix(c(11:16), 2, 3)
> rbind(x.mat, x.vec) # vector as row vector
      [,1] [,2] [,3]
        11   13   15
        12   14   16
x.vec    1    2    3
> cbind(x.mat, y.vec) # warning
               y.vec
[1,] 11 13 15     8
[2,] 12 14 16     9
Warning message:
In cbind(x.mat, y.vec) :
  number of rows of result is not a multiple of vector length (arg 2)
> #
> x.vec <- c(1, 2)
> y.vec <- c(8, 9)
> rbind(x.vec, y.vec) # vector as row vector
      [,1] [,2]
x.vec    1    2
y.vec    8    9
> cbind(x.vec, y.vec) # vector as col vector
     x.vec y.vec
[1,]     1     8
[2,]     2     9
> #
> x.mat <- matrix(c(11:14), 2, 2)
> z.mat <- rbind(x.mat, x.vec) # vector as row vector
> z.mat
      [,1] [,2]
        11   13
        12   14
x.vec    1    2
> cbind(x.mat, y.vec) # vector as col vector
           y.vec
[1,] 11 13     8
```

```
[2,] 12 14     9
> rbind(z.mat, y.vec) # vector as row vector
      [,1] [,2]
        11   13
        12   14
x.vec    1    2
y.vec    8    9
> cbind(z.mat, y.vec) # warning
            y.vec
      11 13     8
      12 14     9
x.vec  1  2     8
Warning message:
In cbind(z.mat, y.vec) :
  number of rows of result is not a multiple of vector length (arg 2)
```

3.2 陣列物件 Array

陣列 (array) 物件也由包含相同模式 (mode) 的元素組成的 p-維資料物件, 也可以將陣列視爲一個向量具 p-維結構.

3.2.1 陣列函式 array()

陣列可以由包含相同模式 (model) 並具有 維度 (dimension) 之 屬性 (attribute), `dim()`. 可以使用函式 `array()` 產生陣列. R 顯示 3-維陣列物件 $m \times n \times k$, `[m, , ]` 出現在某特定元素之前時, 表示某特定元素在該陣列物件之第 m 列 (row) 的位置; `[ , n, ]` 出現在某特定元素之前時, 表示某特定元素在該陣列物件之第 n 欄 (column) 的位置, 依此類推. [[,,k]] 表示 3-維陣列的第 1, 2-維度之矩陣.

3.2.2 陣列的命名

陣列的命名, 與矩陣類似, 可以使用函式 `dimnames()` 分別給予陣列命名. 可以用函式 `dimnames()` 同時檢視 array 之列位名與欄位名. 若要對讀取或命名 陣列

的第 1 維度 (列位名, row name) 或 第 2 維度 (欄位名, column name), 也可以用函式指令 rownames() 與 colnames().

```
> ## array()
> a.vec <- 1:24
> a.vec
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
[21] 21 22 23 24
> b.array <- array(a.vec, dim = c(4, 3, 2),
                   dimnames = list(c("x1", "x2", "x3", "x4"),
                                   c("y1", "y2", "y3"),
                                   c("z1", "z2")))
> b.array
, , z1
   y1 y2 y3
x1  1  5  9
x2  2  6 10
x3  3  7 11
x4  4  8 12

, , z2
   y1 y2 y3
x1 13 17 21
x2 14 18 22
x3 15 19 23
x4 16 20 24

> mode(b.array)
[1] "numeric"
> dim(b.array)
[1] 4 3 2
> length(b.array)
[1] 24
> dimnames(b.array)
[[1]]
[1] "x1" "x2" "x3" "x4"

[[2]]
[1] "y1" "y2" "y3"

[[3]]
[1] "z1" "z2"
> #
> rownames(b.array)
[1] "x1" "x2" "x3" "x4"
> colnames(b.array)
[1] "y1" "y2" "y3"
```

3.2.3 陣列的下標與索引 Array Index

陣列的 下標 或 索引 (index) 之操作, 與矩陣下標 或 索引 的操作類似, 使用 "中括號" `array.name[i, j, ...]` 可存取陣列中的第 $[i, j, ...]$ 元素;
`array.name[i, , ]` 可存取陣列中的第 i 列 (ith row) 物件,
`array.name[ , j, ]` 可存取矩陣中的第 j 欄 (ith column) 物件等等.

```
> # array index
> a.vec
 [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
[21] 21 22 23 24
> b.arr <- array(a.vec, dim = c(4, 3, 2),
                 dimnames = list(c("x1", "x2", "x3", "x4"),
                                 c("y1", "y2", "y3"),
                                 c("z1", "z2")))
> b.arr
, , z1
   y1 y2 y3
x1  1  5  9
x2  2  6 10
x3  3  7 11
x4  4  8 12

, , z2
   y1 y2 y3
x1 13 17 21
x2 14 18 22
x3 15 19 23
x4 16 20 24

> b.arr[1,2,3]
[1] 15
> b.arr[2,3,4]
[1] 24
> b.arr[2,,]
  i ii iii iv
A 2  8  14 20
B 4 10  16 22
C 6 12  18 24
> b.arr[,2,]
  i ii iii iv
X 3  9  15 21
Y 4 10  16 22
> b.arr[,,2]
  A  B  C
X 7  9 11
```

```
40 Y 8 10 12
```

3.3 列表物件 List

列表 (list) 是一個特殊的 “向量”, 這特殊的向量中的元素是物件. 列表物件 元素的模式 (mode) 是 複雜模式 (complex mode), 列表物件的 結構 (structure) 是 “遞迴型” (recursive).

3.3.1 列表函式 list()

列表物件是由資料物件有順序組成, 列表物中的 “元素”, 稱作 “成份” (component), 是物件本身, 列表物中的成份是有順序的 (order sequence), 成份物件的元素模式, 沒有任合限制, 每一個別成份的物件之原型模式可以不相同. 列表的產生通常先決定每一個成分的物件, 然後再組合成列表. 使用函式 `list()` 將個別成分組成列表. R 許多統計分析的結果常常包含不同物件, 例如, 迴歸模型分析包含 參數估計, 變異數分析, 預測值與殘差, 每一成分的長度與模式不一, 模型分析產生的結果最後多以列表儲存.

```
> ## list()
> ## list w/ component names
> x.num <- c(1, 3, 6)
> y.str <- c("chocolate", "vanilla", "strawberry")
> xy.list <- list(x.num.var = x.num, y.str.var = y.str)
> xy.list
$x.num.var
[1] 1 3 6

$y.str.var
[1] "chocolate"  "vanilla"    "strawberry"

> ## list w/ component names
> x.vec <- 1:4
> y.vec <- c("Male", "Female")
> z.mat <- matrix(1:9, nrow = 3, ncol = 3)
> xyz.list <- list(x.vec, y.vec, z.mat)
```

```
> xyz.list
[[1]]
[1] 1 2 3 4

[[2]]
[1] "Male"   "Female"

[[3]]
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> #
> mode(xyz.list)
[1] "list"
> length(xyz.list)
[1] 3
> dim(xyz.list)
NULL
> names(xyz.list)
NULL
> class(xyz.list)
[1] "list"
> #
> class(xyz.list)
[1] "list"
> # list = data matrix
> id.vec <- c(1, 2, 3, 4)
> age.vec <- c(35, 55, 45, 25)
> sex.vec <- c("Male", "Male", "Female", "Female")
> disease.vec <- c("Yes", "No", "No", "Yes")
> x.list <- list(id = id.vec,
                 age = age.vec,
                 sex = sex.vec,
                 disease = disease.vec)
> x.list
$id
[1] 1 2 3 4

$age
[1] 35 55 45 25

$sex
[1] "Male"   "Male"   "Female" "Female"

$disease
[1] "Yes" "No"  "No"  "Yes"
```

3.3.2 列表的下標與索引 List Index

列表物件的 下標 或 索引 之操作, 與矩陣 或 陣列之操作有所不同, 若一個名字爲 `List.Name` 的列表物件, 要取得 list 其中的 第 “i.number” 成份, 須使用 `List.Name[[3]]`. 注意, 在列表物件的下標操作, `[[i.number]]` 與 `[i.number]` 是不一樣的.

```
> ## list index
> ## list w/o component names
> x.vec <- 1:4
> y.vec <- c("Male", "Female")
> z.mat <- matrix(1:9, nrow = 3, ncol = 3)
> xyz.list <- list(x.vec, y.vec, z.mat)
> xyz.list
[[1]]
[1] 1 2 3 4

[[2]]
[1] "Male"   "Female"

[[3]]
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9

> xyz.list[1]
[[1]]
[1] 1 2 3 4

> xyz.list[[1]]
[1] 1 2 3 4
> xyz.list[2]
[[1]]
[1] "Male"   "Female"
> xyz.list[[2]]
[1] "Male"   "Female"

> xyz.list[[3]]
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> xyz.list[3]
[[1]]
     [,1] [,2] [,3]
```

```
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

若列表中的 成份 (component) 有另外命名爲 comp.name, 可以使用函式 List.Name$comp.na 取得成份名字, 會與使用函式 List.Name[[comp.name]] 取得相同結果. 在 List.Name$comp.nar 或 List.Name[[comp.name]] 加上 “中括號” [i,j] 等, 可以取得 List.Name$comp.name 中的元素. [[i.number]] 可以適用在計算指標, 但 $ 僅能使用在有設定的成分名的列表. 若列表的單一成份內有多維度物件可使用巢狀下標取出資料.

```
> # list w/ component names
> x.vec <- 1:4
> y.vec <- c("Male", "Female")
> z.mat <- matrix(1:9, nrow = 3, ncol = 3)
> xyz.list <- list(class = x.vec, gender = y.vec, score = z.mat)
> xyz.list
$class
[1] 1 2 3 4

$gender
[1] "Male"   "Female"

$score
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9

> xyz.list$class
[1] 1 2 3 4
> xyz.list[["class"]]
[1] 1 2 3 4
> xyz.list[["class"]][2]
[1] 2
>
> xyz.list$gender
[1] "Male"   "Female"
> xyz.list[["gender"]][1]
[1] "Male"
>
> xyz.list$score
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
> xyz.list[["score"]][2,3]
[1] 8
```

3.4 資料框架 Data Frame

一組資料通常包含數字與文字, 在 R 中的 向量 與 矩陣 物件, 只允許相同的變數型式, 若要同時存入數字變數與文字變數, R 可以使用 列表 (list) 與 函式指令 `list()`, 列表資料內可包含不同屬性的變數值.

資料框架 (data frame) 是列表物件的一種特殊情境. 在 資料框架 內的每個變數之觀測值數目 (向量長度) 都相等, 類似矩陣型式, 但是每個變數的變數值不一定是相同的模式, 因此 資料框架 有時又稱爲 資料矩陣 (data matrix), 是一般統計資料分析常用的形式.

3.4.1 資料框架函式 data.frame()

R 可以使用函式 `data.frame()` 將資料儲存成資料框架物件. 輸入時須注意每個變數之觀測值數目 (向量長度) 都必須相等, 若有缺失值, 須先將缺失值符號輸入在向量內.

```
> ## data frame = list
> id.vec <- c(1, 2, 3, 4)
> age.vec <- c(35, 55, 45, 25)
> sex.vec <- c("Male", "Male", "Female", "Female")
> disease.vec <- c("Yes", "No", "No", "Yes")
> x.df <- data.frame(id = id.vec,
                     age = age.vec,
                     sex = sex.vec,
                     disease = disease.vec)
> x.df
  id age    sex disease
1  1  35   Male     Yes
2  2  55   Male      No
3  3  45 Female      No
4  4  25 Female     Yes
> mode(x.df)
[1] "list"
> class(x.df)
[1] "data.frame"

> x.df$age
[1] 35 55 45 25
```

```
> x.df$disease
[1] Yes No  No  Yes
Levels: No Yes
> ## data frame error
> a.num <- c(1, 3)
> b.str <- c("chocolate", "vanilla", "strawberry")
> ab.df <- data.frame(a.num.var = a.num, b.str.var = b.str)
Error in data.frame(a.num.var = a.num, b.str.var = b.str) :
  arguments imply differing number of rows: 2, 3
> ab.df
Error: object 'ab.df' not found
> ## data frame add NA
> a.num <- c(1, 3, NA)
> b.str <- c("chocolate", "vanilla", "strawberry")
> ab.df <- data.frame(a.num.var = a.num, b.str.var = b.str)
> ab.df
  a.num.var  b.str.var
1         1  chocolate
2         3    vanilla
3        NA strawberry
```

3.4.2 資料框架的下標與索引 Data Frame Index

資料框架的 下標 或 索引 (index) 操作, 如同矩陣的下標與索引操作, 可以輸入正整數, 負數, 整數向量, 欄位名等等. 矩陣具有 2-維度下標向量, 例如, 可以使用 "中括號 []" `dataframe.name[i, j]` 可存取資料框中的第 $[i, j]$ 元素; `dataframe.name[i, ]` 可存取資料框中的第 i 列 (ith row), `dataframe.name[ , j]` 可存取資料框中的第 j 欄 (ith column).

資料框架是列表的特例, 因此資料框架的 下標 或 索引 (index) 操作, 也可如同列表的下標與索引操作. 若一個名爲 `data.Name` 的資料框架, 要取的其中的 第 "i.number" 變數 (成份), 須使用 `data.Name[[3]]`. 注意, 在資料框架物件的下標操作, `[[i.number]]` 與 `[i.number]` 是不一樣的.

若資料框架中的變數命名爲 `variable.name`, 可以使用函式 `dataframe.Name$variable.name` 取得變數資料, 會與使用函式 `dataframe.Name[[variable.name]]` 取得相同結果.

在 dataframe.Name$variable.name 或 dataframe.Name[[variable.name]] 加上 “中括號” [i] 等, 可以取得 dataframe.Name$variable.name 中的元素. [[i.number]] 可以適用在計算指標, 但 $ 僅能使用在有設定變數名的資料框架.

```
> ## data frame index
> data(Puromycin)
> Puromycin
    conc rate      state
1   0.02   76    treated
2   0.02   47    treated
3   0.06   97    treated
.......................
21 0.56  144 untreated
22 0.56  158 untreated
23 1.10  160 untreated
> Puromycin$rate
 [1]  76  47  97 107 123 139 159 152 191 201 207 200  67  51  84
[16]  86  98 115 131 124 144 158 160
> Puromycin$state
 [1] treated   treated   treated   treated   treated   treated
 [7] treated   treated   treated   treated   treated   treated
[13] untreated untreated untreated untreated untreated untreated
[19] untreated untreated untreated untreated untreated
Levels: treated untreated
> Puromycin[1]
    conc
1   0.02
2   0.02
3   0.06
.......
21 0.56
22 0.56
23 1.10
> Puromycin[1][[1]]
 [1] 0.02 0.02 0.06 0.06 0.11 0.11 0.22 0.22 0.56 0.56 1.10 1.10
[13] 0.02 0.02 0.06 0.06 0.11 0.11 0.22 0.22 0.56 0.56 1.10
> Puromycin$state[1:3]
[1] treated treated treated
Levels: treated untreated
> Puromycin[1:3, 1:2]
  conc rate
1 0.02   76
2 0.02   47
3 0.06   97
```