第 2 章: 輸入與輸出資料之基本方法
2: Basic Importing and Exporting Data

在 R 中, 資料數值 (data values) 以具有名稱的 "**物件**" 形式儲存, 它們可以是向量 (vector), 矩陣 (matrix), 陣列 (array), 列表 (Lists), 或 資料框架 (data frames) 等. 簡單的資料, 可以直接在 R 視窗中輸入, 大型資料, 通常先以資料庫軟體, 試算表軟體等輸入儲存成 R 的外部檔案, 由 R 從外部檔案中讀入, 而不是在 R 中, 用鍵盤輸入.

2.1 資料變數值

資料變數值可能是實際測量數值, 或是屬性的描述, 常會有遺失值, 不同的領域, 不同的軟體, 各有不同的分類. 在此, 主要討論 R 的變數值分類.

2.1.1 變數的分類

在 R 中, 最簡單的資料數值以具有名稱的 向量 (vector) 形式儲存, 單一數值 (scalar), 可視爲僅具有單一元素的向量. 在 R 中資料變數的分類, 與一般基礎統計教科書不盡相同, 在 R 中, 資料中向量變數的數值, 可分成

1. `"numeric"`, 實數向量
2. `"integer"`, 整數向量 (有時需特別指定)
3. `"logical"`, 邏輯變數向量 (true or false), 以 `TRUE` (`T`) 或 `FALSE` (`F`) 呈現, (也可以是 1 (`T`) 與 0 (`F`).
4. `"complex"`, 複數向量
5. `"character"`, 文字或字串向量, 通常輸入時, 在文字或字串兩側加上雙引號.
6. `"list"`, 列表, 是一個由 R (S) 物件所組成的向量.

2.1.2 遺失值 (缺失值) Missing Values

研究資料, 通常會有 **遺失值** (**缺失值**) (**missing value**), 在 R 中, 輸入或輸出遺失值, 通常以 NA 表示, (NA = "not available"), R 還有另外有 NaN = "Not a Number", 以及 NULL 是指物件的長度是 0. 函式 is.na() 可查看向量內那些元素是遺失值.

```
> z<-c(1:2,NA)
[1]  1  2  NA
> is.na(z)
[1] FALSE FALSE  TRUE
> log(z)
[1] 0.0000000 0.6931472   NA
> z/0
[1] Inf Inf  NA
> 0/0
[1] NaN
```

2.2 簡單資料輸入 Entering Simple Data

簡單資料, 通常直接在 R 視窗中輸入, 使用 c(), edit(), scan() 等指令; 也可直接以矩陣 (matrix), 陣列 (array), 列表 (Lists), 形式輸入.

2.2.1 向量與使用 c() 指令 (Vector and c())

簡單資料, 通常使用指派運算 (Assignment), 直接在 R 視窗中輸入, 如

```
> x <- 49.5  # 實數
> x
[1] 49.5
> sqrt(x)
[1] 7.035624
> #
> x<-3 # 整數
> x
[1] 3
> x<-T # 邏輯變數
> y<-F
> x
[1] TRUE
> y
[1] FALSE
> #
> a<-"1" # 文字
> a
```

```
[1] "1"
> a<-1   # 實數
> a
[1] 1
> a<-"character" # 文字
> a
[1] "character"
> z<-"The dog ate my homework"  # 字串
> z
[1] "The dog ate my homework"
> sub("dog","cat",z) # 字串代換
[1] "The cat ate my homework"
> #
> z<-2.3+4.2i # 複數
> z
[1] 2.3+4.2i
```

輸入簡單的向量資料, 可以用 c() 指令 (或 assign()), 如

```
> x <- c(10.4, 5.6, 3.1, 6.4, 21.7)
> x
[1] 10.4  5.6  3.1  6.4 21.7
> assign("x", c(10.4, 5.6, 3.1, 6.4, 21.7))
> x
[1] 10.4  5.6  3.1  6.4 21.7
> c(10.4, 5.6, 3.1, 6.4, 21.7) -> x
> x
[1] 10.4  5.6  3.1  6.4 21.7
> 1/x
[1] 0.09615385 0.17857143 0.32258065 0.15625000 0.04608295
> y <- c(x, 0, x)
> y
 [1] 10.4  5.6  3.1  6.4 21.7  0.0 10.4  5.6  3.1  6.4 21.7
> #
> (flavors<-c("chocolate", "vanilla", "strawberry"))
[1] "chocolate"  "vanilla"    "strawberry"
> flavors
[1] "chocolate"  "vanilla"    "strawberry"
> #
> x<-c(1:5)
> x.temp<-x>2
> x.temp
[1] FALSE FALSE  TRUE  TRUE  TRUE
```

輸入序列向量 (sequence), 可以用 seq() 指令,

```
> x<-c(101:105)
> x
[1] 101 102 103 104 105
```

```
> x<-101:105
> x
[1] 101 102 103 104 105
> (y<-seq(0,5,0.5))
 [1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
> y
 [1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
> help(seq)
```

輸入同一數值重複多次, 可以用 `rep()` 指令,

```
> x<-rep(1,5)
> x
[1] 1 1 1 1 1
> help(rep)
```

2.2.2 使用矩陣與使用 matrix() 指令 (Matrix and matrix())

輸入簡單的矩陣資料, (列 × 行), 或希望以矩陣形式儲存, 可以用 `matrix()`指令, 設定列數 `nrow=2` 或欄 (行) 數 `ncol=2`, R 設定是以欄 (行) 位 (column) 優先填滿, 要改變設定, 可加入 `byrow=T`.

```
> x<-matrix(c(1, 5, 3, 7, 4, 9), nrow=2)
> x
     [,1] [,2] [,3]
[1,]    1    3    4
[2,]    5    7    9
> x<-matrix(c(1, 5, 3, 7, 4, 9), nrow=2, byrow=T)
> x
     [,1] [,2] [,3]
[1,]    1    5    3
[2,]    7    4    9
> y<-matrix(c(1, 5, 3, 7, 4, 9), ncol=2)
> y
     [,1] [,2]
[1,]    1    7
[2,]    5    4
[3,]    3    9
> y<-matrix(c(1, 5, 3, 7, 4, 9), ncol=2, byrow=T)
> y
     [,1] [,2]
[1,]    1    5
[2,]    3    7
[3,]    4    9
> z<-matrix(1:18, nrow=3)
> z
     [,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
```

2.2.3 列表與使用 list() 或 data.frame() 指令

一組資料, 通常包含數字與文字, 向量與矩陣資料, 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, 可以使用列表與 list() 指令, 或資料框架 (data frame) 與 data.frame() 指令, data.frame() 是 list() 的一種特殊情況, data.frame() 內的變數觀測值數目 (向量長度) 都相等, 是一般統計資料分析常用形式.

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

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

> #
> a.num<-c(1,3)
> b.str<-c("chocolate", "vanilla", "strawberry")
> ab.list<-list(a.num.name=a.num, b.str.name=b.str)
> ab.list
$a.num.name
[1] 1 3
$b.str.name
[1] "chocolate"  "vanilla"    "strawberry"

> #####  data.frame()
> x.num<-c(1,3,6)
> y.str<-c("chocolate", "vanilla", "strawberry")
> xy.data<-data.frame(x.num.name=x.num, y.str.name=y.str)
> xy.data
  x.num.name y.str.name
1          1  chocolate
2          3    vanilla
3          6 strawberry
> #
> a.num<-c(1,3)
> b.str<-c("chocolate", "vanilla", "strawberry")
> ab.data<-data.frame(a.num.name=a.num, b.str.name=b.str)
錯誤在 data.frame(a.num.name = a.num, b.str.name = b.str) :
        arguments imply differing number of rows: 2, 3
```

2.2.4 使用 edit() 指令

輸入資料, 可以使用 edit() 指令, 輸入一個新的資料組 x.data,

```
x.data<-edit(data.frame())
```

若要修改 x.data,

```
edit(data.frame(x.data))
```

2.3 輸入外部資料檔案與輸出資料

R 的原先設計假設, 你用其他工具 (如檔編輯器, 資料庫軟體, 或試算表軟體等) 輸入儲存成 R 的外部檔案, 並修改成特定的輸入檔格式, 以使它們符合 R 輸入外部資料檔案的要求. 對統計分析來說, 這是比較容易的.

2.3.1 資料框架 Data Frames

統計計算分析的資料, 通常有一個的基本架構, 在 R 稱作 **資料框架** (**data frame**). 資料框架是類似於在 SAS, STATA 等的 dataset 架構. 資料框架通常類似矩形的列聯表 (cross table), 在資料框架內, 同一變數的數值是在同一欄內 (column). 在 R, 資料框架是統計分析中最基本的資料結構, 許多統計模型分析必須用到資料框架. 資料框架與矩陣類似, 不同的地方在資料框架變數不需要是相同的變數形式 (種類), 實數變數, 文字變數, 邏輯變數等可已放在同一資料框架中, 如表 2.1. 但是矩陣物件只能有相同的變數形式.

1. 每一欄 (column), 都是一個變數
2. 第一列 (row), 可以是變數的 "變數名稱" (variable names)
3. 每一欄的變數形式可以是實數, 文字, 邏輯變數.
4. 第一欄 (column) 有時候是 "**列標籤**" (**row label**)

表 2.1: **DM-TKR** Study Data

No	age	sex	DM	DMyr	preAC	prePC	postAC	postPC	Med	SIDE	PREKS	POSKS	ABS	INFECT
1	67	0	0	10	120	160	140	180	0	0	56	92	1	0
2	67	0	0	11	100	150	150	220	0	1	62	62	0	1
3	72	1	0	4	150	200	120	150	2	0	60	94	1	0
4	82	1	0	8	150	200	160	250	0	1	47	90	1	0
5	73	1	0	3	85	110	140	200	0	0	44	88	0	0
							...							

2.3.2 輸入外部資料檔案至 R 資料框架

統計分析變數, 主要在資料框架 (data frame) 中操作, 使用資料框架函式 `read.table()` 或 `read.csv()` 讀入 / 輸入外部資料檔案, 最容易. (其他函式, 如 `scan()`, 對初學者, 較困難.) 爲了可以直接讀取整個外部檔案進入資料框架, 外部檔案常常要求有特定的格式, 例如

1. 通常是 ASCII 形式的檔案, 多數檔編輯器, 資料庫, 試算表等軟體可以存取.
2. 第一的列 (行) (line, row) 可以有該資料各個變數的 "變數名稱" (variable names) 或是 "**列的名字**" (**column name**).
3. 其餘的列 (line, row), 是各個變數的值.
4. 變數之間, 可以空格分開, 或其他特定符號分隔.
5. 若是變數資料是文字列型, 通常以雙引號包含.
6. 以 "," 分開變數值的 ASCII 形式檔案, 一般成爲 **comma-separated-variable format**, 檔案名通常以 **.csv** 作爲延伸檔名,
7. 第一欄 (column) 有時候是 "**列標籤**" (**row label**) 或是 "**列的名字**" (**row name**)

假設在 R 的工作目錄中, 有一個 ASCII 形式的檔案, 如表 2.2.

表 2.2: **DM-TKR** Study Data in **DMTKRcsv.csv**, (ASCII, CSV Format File)

```
No,age,sex,DM,DMyr,preAC,prePC,postAC,postPC,Med,SIDE,PREKS,POSKS,ABS,INFECT
1,67,0,0,10,120,160,140,180,0,0,56,92,1,0
2,67,0,0,11,100,150,150,220,0,1,62,62,0,1
3,72,1,0,4,150,200,120,150,2,0,60,94,1,0
4,82,1,0,8,150,200,160,250,0,1,47,90,1,0
5,73,1,0,3,85,110,140,200,0,0,44,88,0,0
6,76,0,0,1,120,150,120,200,0,1,52,94,1,0
7,76,0,0,1,120,150,120,200,0,0,48,96,0,0
8,77,0,1,35,200,250,230,300,1,1,42,90,1,0
9,64,0,0,5,130,180,100,150,0,0,40,94,1,0
10,64,0,0,5,130,180,100,150,0,1,45,96,0,0
... ... ...
```

用資料框架函式 `read.table()` 或 `read.csv()` 讀入 / 輸入外部資料檔案 **DMTKRcsv.csv**, 函式中的引數,

1. `header = T` 表示第一列, 是變數的名稱,
2. `row.names = NULL` 表示第一欄無列標籤,
3. `sep = ","` 表示以 "," 分開變數,

4. dec = "." 表示 "." 爲實數的小數點.

```
> DMTKRtable<-read.table("DMTKRcsv.csv",
    header = TRUE, row.names = NULL, sep = ",", dec = ".")
> DMTKRtable
   No age sex DM DMyr preAC prePC postAC postPC Med SIDE PREKS POSKS ABS INFECT
1   1  67   0  0   10   120   160    140    180   0    0    56    92   1      0
2   2  67   0  0   11   100   150    150    220   0    1    62    62   0      1
3   3  72   1  0    4   150   200    120    150   2    0    60    94   1      0
4   4  82   1  0    8   150   200    160    250   0    1    47    90   1      0
5   5  73   1  0    3    85   110    140    200   0    0    44    88   0      0
.........
>###
> DMTKRcsv<-read.csv("DMTKRcsv.csv",
    header = TRUE, sep = ",", dec = ".")
> DMTKRcsv
   No age sex DM DMyr preAC prePC postAC postPC Med SIDE PREKS POSKS ABS INFECT
1   1  67   0  0   10   120   160    140    180   0    0    56    92   1      0
2   2  67   0  0   11   100   150    150    220   0    1    62    62   0      1
3   3  72   1  0    4   150   200    120    150   2    0    60    94   1      0
4   4  82   1  0    8   150   200    160    250   0    1    47    90   1      0
5   5  73   1  0    3    85   110    140    200   0    0    44    88   0      0
.........
```

2.3.3 使用 R 內建資料框架

R 有許多內建資料框架, 另外貢獻套件 (contributed packages) 也有許多內建建資料框架, 可以使用 data() 查看內建資料框架名稱, 或 data(package = "package.name") 查看名稱爲 package.name 套件中的資料框架名稱, 載入資料框架, 可用 data(data.name) 載入 R 內建名稱爲 data.name 資料框架使用, 或 data(data.pack.name, package = "package.name") 載入名稱爲 package.name 套件中, 名稱爲 data.pack.name 資料框架使用.

```
> data()       # 查看內建資料框架名稱
> data(Orange) # 載入 R 內建資料框架 Orange
> help(Orange)
> Orange
   Tree  age circumference
1     1  118            30
2     1  484            58
3     1  664            87
.........................
> #
> library(MASS)
> help(package=MASS)
> data(package="MASS")     # 查看 MASS 套件內建資料框架名稱
> data(VA, package="MASS") # 載入 MASS 套件內建資料框架 VA
> help(VA)
```

```
> VA
    stime status treat age Karn diag.time cell prior
1      72      1     1  69   60         7    1     0
2     411      1     1  64   70         5    1    10
3     228      1     1  38   60         3    1     0
4     126      1     1  63   60         9    1    10
.....................................................
```

2.3.4 使用資料框架內的變數: attach() 與 detach() 指令

當輸入外部資料檔案入資料框架, 或載入內建資料框架, 要使用其中變數, 如 DMTKRtable 中的 age,

```
> DMTKRtable$age
 [1] 67 67 72 82 73 76 76 77 64 64 78 69 73 73 81 81 74 62 75 90 71 71 83
[24] 77 80 80 67 67 72 81 81 74 73 74 63 60 75 67 69 69 70 72 76 74 79 65
[47] 72 69 71 65 67 70 76 74 80 61 56 61 74 66 67 75 54 71 65 70 71 69 74
[70] 76 61 54 61 72 64 65 71 59
```

當固定使用同一資料框架持續進行統計分析時,

1. 可以用 attach(data.frame.name) 載入名稱爲 data.frame.name 資料框架. 這樣就可以直接適用變數名稱來進行分析.
2. 可以用 names(data.frame.name) 取得名稱爲 data.frame.name 資料框架內的變數名稱.

```
> attach(DMTKRtable)
> age
 [1] 67 67 72 82 73 76 76 77 64 64 78 69 73 73 81 81 74 62 75 90 71 71 83
[24] 77 80 80 67 67 72 81 81 74 73 74 63 60 75 67 69 69 70 72 76 74 79 65
[47] 72 69 71 65 67 70 76 74 80 61 56 61 74 66 67 75 54 71 65 70 71 69 74
[70] 76 61 54 61 72 64 65 71 59
> mean(age)  # age 平均值
[1] 70.83333
> names(DMTKRtable)
[1] "No"     "age"    "sex"    "DM"     "DMyr"   "preAC"  "prePC"
[8] "postAC" "postPC" "Med"    "SIDE"   "PREKS"  "POSKS"  "ABS"
[15] "INFECT"
```

當不再使用某一特定資料框架時, 可以用 detach(data.frame.name), 移出 R 工作空間常駐位置.

```
 > detach(DMTKRtable)
> age
錯誤: 找不到目的物件 "age"
> DMTKRtable$age
 [1] 67 67 72 82 73 76 76 77 64 64 78 69 73 73 81 81 74 62 75 90 71 71 83
[24] 77 80 80 67 67 72 81 81 74 73 74 63 60 75 67 69 69 70 72 76 74 79 65
```

```
 [47] 72 69 71 65 67 70 76 74 80 61 56 61 74 66 67 75 54 71 65 70 71 69 74
 [70] 76 61 54 61 72 64 65 71 59
> mean(DMTKRtable$age)
[1] 70.83333
```

2.3.5 輸出 R 資料至外部檔案

統計分析, 常常需儲存資料或分析結果, 輸出至外部檔案, 供其他軟體使用, 主要 使用資料框架函式 `write.table()`, 相對應 `read.table()`, 輸出資料 (資料框架或矩陣) 至外部資料檔案, 最容易. 對大型資料, 可以使用 MASS 套件中的 `write.matrix()` 會較有效率.

```
> write.table(Orange, file = "Orange.csv",
    sep = ",", col.names = NA)
> ?write.table
> #
> library(MASS)
> ?write.matrix
> write.matrix(Orange, file = "OrangeMASS.csv", sep = ",")
```