

原住民族資料分析線上訓練工作坊：R的基礎 與應用

第二週

Kacing 廖彥傑

英國艾賽克斯大學博士候選人

結構 (Data Structure)

Table 5-1. R object-mode mapping

| Object | Example | Mode |
|-----------------------------|---|-----------|
| Number | 3.1415 | Numeric |
| Vector of numbers | <code>c(2.7.182, 3.1415)</code> | Numeric |
| Character string | "Moe" | Character |
| Vector of character strings | <code>c("Moe", "Larry", "Curly")</code> | Character |
| Factor | <code>factor(c("NY", "CA", "IL"))</code> | Numeric |
| List | <code>list("Moe", "Larry", "Curly")</code> | List |
| Data frame | <code>data.frame(x=1:3, y=c("NY", "CA", "IL"))</code> | List |
| Function | <code>print</code> | Function |

參考書目：R Cookbook Proven Recipes for Data Analysis, Statistics, and Graphics by J. D. Long, Paul Teetor

向量 (Vectors)

建立一個向量，命名(`names()`) 它。

```
v <- c(20, 20, NA)
names(v) <- c("Kacing", "Labak", "Truku")
print(v)
```

```
#> Kacing Labak Truku
#>      20      20      NA
```

提取名稱為

```
v[["Kacing"]]
```

```
#> [1] 20
```

```
v["Kacing"]
```

```
#> Kacing
#>      20
```

列表 (Lists)

用 `list()` 建立 `list` 物件

```
a.list <- list(x = 1:6,  
             y = "a",  
             z = c(TRUE, FALSE))
```

提取它

```
a.list$x
```

```
#> [1] 1 2 3 4 5 6
```

另個方式提取它

```
a.list[["x"]]
```

```
#> [1] 1 2 3 4 5 6
```

列表 (Lists)

比較一下 `a.list[["x"]]` 與 `a.list["x"]`

```
a.list[["x"]]
```

```
#> [1] 1 2 3 4 5 6
```

```
a.list["x"]
```

```
#> $x
```

```
#> [1] 1 2 3 4 5 6
```

用 `c()` 提取，挑取特定的 **vector**

```
a.list[c(1,3)]
```

```
#> $x
```

```
#> [1] 1 2 3 4 5 6
```

```
#>
```

```
#> $z
```

```
#> [1] TRUE FALSE
```

矩陣 (matrix)

`ncol = 1` 的matrix

```
one.col.matrix <- matrix(1:6, ncol = 1)
```

`ncol = 2` 的matrix

```
two.col.matrix <- matrix(1:6, ncol = 2)
```

構面

```
dim(two.col.matrix)
```

```
#> [1] 3 2
```

矩陣 (matrix)

建立4個col的matrix

```
A <- matrix(1:20, ncol = 4)
```

matrix的遞遺性

```
A + 1
```

```
#>      [,1] [,2] [,3] [,4]
#> [1,]    2    7   12   17
#> [2,]    3    8   13   18
#> [3,]    4    9   14   19
#> [4,]    5   10   15   20
#> [5,]    6   11   16   21
```

factor (類化)

```
my.vector <- c(1, 1, 0, 0, 0, 1)
my.factor <- factor(x = my.vector,
                   levels = c(1, 0),
                   labels = c("treated", "control"))
```

用 `levels()` 看剩幾種類別

```
levels(my.factor)
```

```
#> [1] "treated" "control"
```

內建函數 (build-in function)

用 `str()` 查看 `a.list` 內容

```
str(a.list)
```

```
#> List of 3  
#> $ x: int [1:6] 1 2 3 4 5 6  
#> $ y: chr "a"  
#> $ z: logi [1:2] TRUE FALSE
```

用 `append()` 增加更多 `list` 於之前的 `list`

```
another.list <- append(a.list, list(yy = 1:10, zz = letters[5:1]))
```

用 `str()` 查看

```
str(another.list)
```

```
#> List of 5  
#> $ x : int [1:6] 1 2 3 4 5 6  
#> $ y : chr "a"  
#> $ z : logi [1:2] TRUE FALSE  
#> $ yy: int [1:10] 1 2 3 4 5 6 7 8 9 10  
#> $ zz: chr [1:5] "e" "d" "c" "b" ...
```

內建函數 (build-in function)

刪除某個list 中的vector

```
a.list$y <- NULL
```

鳥巢list (nested list)

```
nested.list <- list(A = list("a", "aa", "aaa"),  
                   B = list("b", "bb"))
```

用 `str()` 查看

```
str(nested.list, max.level = 1)
```

```
#> List of 2  
#> $ A:List of 3  
#> $ B:List of 2
```

內建函數 (build-in function)

用 `is.list()`，判斷 `nested.list` 是否屬於 `list` 性質

```
is.list(nested.list)
```

```
#> [1] TRUE
```

用 `unlist()` 去 `list` 特質

```
c.vec <- unlist(nested.list)
```

用 `mode()`，判斷 `nested.list` 是否屬於 `list` 性質

```
mode(nested.list)
```

```
#> [1] "list"
```

```
names(nested.list)
```

```
#> [1] "A" "B"
```

```
mode(c.vec)
```

```
#> [1] "character"
```

練習：提取鳥巢序列(nested list)

首先，建立一個鳥巢list，嘗試提換別的數值。

```
nested.list <- list(A = list("a", "aa", "aaa"),  
                  B = list("b", "bb"))
```

```
str(nested.list)
```

```
#> List of 2  
#> $ A:List of 3  
#> ..$ : chr "a"  
#> ..$ : chr "aa"  
#> ..$ : chr "aaa"  
#> $ B:List of 2  
#> ..$ : chr "b"  
#> ..$ : chr "bb"
```

用以下範例自己跑看看

```
nested.list[1]  
nested.list[[1]][2]  
nested.list[[1]][[2]]  
nested.list[2]  
nested.list[2][[1]]
```

資料架構 (Dataframe)

創建一個資料框架

```
a.df <- data.frame(x = 1:6,  
                  y = "a",  
                  z = c(TRUE, FALSE))
```

用 `is.data.frame()` 判斷 **a.df** 是否屬於資料框架

```
is.data.frame(a.df)
```

```
#> [1] TRUE
```

用 `is.list()` 判斷 **a.df** 是否屬於 **list** 序列

```
is.list(a.df)
```

```
#> [1] TRUE
```

資料架構 (Dataframe)

`length()` 看有多少變數

```
length(a.df)
```

```
#> [1] 3
```

`colnames()` 看變數名稱

```
colnames(a.df)
```

```
#> [1] "x" "y" "z"
```

提取變數裡的資料

```
a.df$x
```

```
#> [1] 1 2 3 4 5 6
```

另一個方式，提取變數裡的資料

```
a.df[["x"]]
```

```
#> [1] 1 2 3 4 5 6
```

資料架構 (Dataframe)

選取第一個column

```
a.df[ , 1]
```

```
#> [1] 1 2 3 4 5 6
```

```
a.df[[1]]
```

```
#> [1] 1 2 3 4 5 6
```

選取第一個row

```
a.df[1, ]
```

| x | y | z |
|---|---|------|
| 1 | a | TRUE |

資料架構 (Dataframe)

選取1至2 row

```
a.df[1:2, ]
```

| x | y | z |
|---|---|-------|
| 1 | a | TRUE |
| 2 | a | FALSE |

取帶a, 1 特為99

```
a.df[1, 1] <- 99
```

把a, 1 所有值變成-99

```
a.df[ , 1] <- -99
```

資料架構 (Dataframe)

用 `subset()` 取中x大於3

```
subset(a.df, x > 3)
```

| | | |
|---|---|---|
| x | y | z |
|---|---|---|

```
subset(a.df, x > 5)[ , -3]
```

| | |
|---|---|
| x | y |
|---|---|

```
subset(a.df, x > 3)$x
```

```
#> numeric(0)
```

資料架構 (Dataframe)

用 `which()` 來找出特定值

```
which(colnames(a.df) == "y")
```

```
#> [1] 2
```

不要選取用 `-`，選區用 `+`

```
a.df[, -which(colnames(a.df) == "y")]
```

| x | z |
|-----|-------|
| -99 | TRUE |
| -99 | FALSE |
| -99 | TRUE |
| -99 | FALSE |
| -99 | TRUE |
| -99 | FALSE |