# 第 14 天

## 資料處理技巧

初接觸 R 語言的使用者會花很多的時間在處理資料上面，所以我們特別將常見的技巧整理在這裡，我們探討的技巧會針對資料框（Data Frame）這個資料結構，因為這也是多數 R 語言使用者最常面對的資料型態。

### 解構資料框

一個資料框可以被解構為單一變數值、向量與較小的資料框，雖然在[把變數集結起來（2）](chapter6.md)中我們已經有討論過，但為了內容的完整性，在此我們再複習一下相關的技巧。

#### 解構為單一變數值

取出內建資料集 iris 第一筆觀測值的第五個變數（Species）的值：

> iris[1, 5]  
[1] setosa  
Levels: setosa versicolor virginica  
> iris[1, "Species"]  
[1] setosa  
Levels: setosa versicolor virginica

#### 解構為向量

取出內建資料集 iris 的某一個變數：

> iris$Sepal.Length # 亦可以寫作 iris[, "Sepal.Length"]  
 [1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8 4.3 5.8 5.7 5.4 5.1 5.7  
 [20] 5.1 5.4 5.1 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2 5.5 4.9 5.0 5.5 4.9  
 [39] 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1 4.6 5.3 5.0 7.0 6.4 6.9 5.5 6.5 5.7 6.3  
 [58] 4.9 6.6 5.2 5.0 5.9 6.0 6.1 5.6 6.7 5.6 5.8 6.2 5.6 5.9 6.1 6.3 6.1 6.4 6.6  
 [77] 6.8 6.7 6.0 5.7 5.5 5.5 5.8 6.0 5.4 6.0 6.7 6.3 5.6 5.5 5.5 6.1 5.8 5.0 5.6  
 [96] 5.7 5.7 6.2 5.1 5.7 6.3 5.8 7.1 6.3 6.5 7.6 4.9 7.3 6.7 7.2 6.5 6.4 6.8 5.7  
[115] 5.8 6.4 6.5 7.7 7.7 6.0 6.9 5.6 7.7 6.3 6.7 7.2 6.2 6.1 6.4 7.2 7.4 7.9 6.4  
[134] 6.3 6.1 7.7 6.3 6.4 6.0 6.9 6.7 6.9 5.8 6.8 6.7 6.7 6.3 6.5 6.2 5.9

我們可以清楚地發現，取出單一個變數後的資料結構就是**向量**，我們回憶一下使用 data.frame() 函數建立資料框的過程，是否也同樣是輸入數個向量然後結合成為一個資料框呢？

> # 輸入 5 個向量結合成為一個有 5 個變數的資料框  
> team\_name <- c("Chicago Bulls", "Golden State Warriors")  
> wins <- c(72, 73)  
> losses <- c(10, 9)  
> is\_champion <- c(TRUE, FALSE)  
> season <- c("1995-96", "2015-16")  
>   
> great\_nba\_teams <- data.frame(team\_name, wins, losses, is\_champion, season)

#### 解構為較小的資料框

透過指定 [m, n] 中的索引值取出前六個觀測值與其中三個變數：

> iris[1:6, c("Sepal.Length", "Petal.Length", "Species")]  
 Sepal.Length Petal.Length Species  
1 5.1 1.4 setosa  
2 4.9 1.4 setosa  
3 4.7 1.3 setosa  
4 4.6 1.5 setosa  
5 5.0 1.4 setosa  
6 5.4 1.7 setosa

或者利用邏輯值判斷選擇部分的資料框, 這也是實務上較常使用的方式：

> iris[iris$Petal.Length >= 6, c("Sepal.Length", "Petal.Length", "Species")]  
 Sepal.Length Petal.Length Species  
101 6.3 6.0 virginica  
106 7.6 6.6 virginica  
108 7.3 6.3 virginica  
110 7.2 6.1 virginica  
118 7.7 6.7 virginica  
119 7.7 6.9 virginica  
123 7.7 6.7 virginica  
126 7.2 6.0 virginica  
131 7.4 6.1 virginica  
132 7.9 6.4 virginica  
136 7.7 6.1 virginica

假如我們希望使用更多的邏輯值判斷，可以使用 &（and）或者 |（or）這樣的運算子連結判斷條件：

> filter <- (iris$Petal.Length >= 6) & (iris$Sepal.Length >= 7.5)  
> iris[filter, c("Sepal.Length", "Petal.Length", "Species")]  
 Sepal.Length Petal.Length Species  
106 7.6 6.6 virginica  
118 7.7 6.7 virginica  
119 7.7 6.9 virginica  
123 7.7 6.7 virginica  
132 7.9 6.4 virginica  
136 7.7 6.1 virginica

在這裡很多的 R 語言使用者，特別是有其他程式語言編寫經驗的就會問一個問題：& 與 && 或者說 | 與 || 的差別是什麼呢？

在 R 語言中連結兩個單變數的判斷條件，我們使用 && 與 ||：

> 8 > 7 && 8 < 7  
[1] FALSE  
> 8 > 7 || 8 < 7  
[1] TRUE

假如是連結多個變數（例如向量）判斷條件，我們使用 & 與 |：

> c(8, 8) > c(7, 7) & c(8, 8) < c(7, 7)  
[1] FALSE FALSE  
> c(8, 8) > c(7, 7) | c(8, 8) < c(7, 7)  
[1] TRUE TRUE

在連結兩個單變數的判斷條件時若使用 & 與 |，其實不會影響結果：

> 8 > 7 & 8 < 7  
[1] FALSE  
> 8 > 7 | 8 < 7  
[1] TRUE

然而在連結多個變數判斷條件時若使用 && 與 ||，就只會納入第一個索引值的元素做判斷，這就不是我們要的結果：

> c(8, 8) > c(7, 7) && c(8, 8) < c(7, 7)  
[1] FALSE  
> c(8, 8) > c(7, 7) || c(8, 8) < c(7, 7)  
[1] TRUE

假如您是對語法有些潔癖的使用者，兩個寫法的差異解釋如上，這是特別為您準備的；如果您是一個大而化之的使用者，我們建議您一律使用 & 與 | 即可。

### 新增與刪除一個變數

#### 新增一個變數

直接將新的向量指派給既有的資料框：

> team\_name <- c("Chicago Bulls", "Golden State Warriors")  
> wins <- c(72, 73)  
> losses <- c(10, 9)  
> is\_champion <- c(TRUE, FALSE)  
>   
> # 既有的資料框  
> great\_nba\_teams <- data.frame(team\_name, wins, losses, is\_champion)  
>   
> # 新增一個變數  
> season <- c("1995-96", "2015-16")  
> great\_nba\_teams$season <- season  
> great\_nba\_teams  
 team\_name wins losses is\_champion season  
1 Chicago Bulls 72 10 TRUE 1995-96  
2 Golden State Warriors 73 9 FALSE 2015-16

#### 新增一個衍生的變數

直接將計算後的向量指派給既有的資料框：

> team\_name <- c("Chicago Bulls", "Golden State Warriors")  
> wins <- c(72, 73)  
> losses <- c(10, 9)  
> is\_champion <- c(TRUE, FALSE)  
>   
> # 既有的資料框  
> great\_nba\_teams <- data.frame(team\_name, wins, losses, is\_champion)  
>   
> # 新增一個衍生變數  
> great\_nba\_teams$winning\_percentage <- great\_nba\_teams$wins / (great\_nba\_teams$wins + great\_nba\_teams$losses)  
> great\_nba\_teams  
 team\_name wins losses is\_champion winning\_percentage  
1 Chicago Bulls 72 10 TRUE 0.8780488  
2 Golden State Warriors 73 9 FALSE 0.8902439

#### 刪除變數

將欲刪除的變數指派為 NULL：

> team\_name <- c("Chicago Bulls", "Golden State Warriors")  
> wins <- c(72, 73)  
> losses <- c(10, 9)  
> is\_champion <- c(TRUE, FALSE)  
>   
> # 既有的資料框  
> great\_nba\_teams <- data.frame(team\_name, wins, losses, is\_champion)  
>   
> # 刪除變數  
> great\_nba\_teams$is\_champion <- NULL  
> great\_nba\_teams  
 team\_name wins losses  
1 Chicago Bulls 72 10  
2 Golden State Warriors 73 9

### 新增與刪除觀測值

#### 新增觀測值

先將屬於**因素向量**的變數轉換回文字，再使用 rbind() 函數，這是為了防止新增因素向量的層級而造成錯誤：

> team\_name <- c("Chicago Bulls", "Golden State Warriors")  
> wins <- c(72, 73)  
> losses <- c(10, 9)  
> is\_champion <- c(TRUE, FALSE)  
> season <- c("1995-96", "2015-16")  
>   
> # 既有資料框，文字的變數被記錄為因素向量  
> great\_nba\_teams <- data.frame(team\_name, wins, losses, is\_champion, season)  
>   
> # 新增觀測值  
> lakers\_7172 <- c("Los Angeles Lakers", 69, 13, TRUE, "1971-72")  
> great\_nba\_teams <- rbind(great\_nba\_teams, lakers\_7172)  
Warning messages:  
1: In `[<-.factor`(`\*tmp\*`, ri, value = "Los Angeles Lakers") :  
 invalid factor level, NA generated  
2: In `[<-.factor`(`\*tmp\*`, ri, value = "1971-72") :  
 invalid factor level, NA generated

轉換成文字後再進行新增就沒有問題：

> team\_name <- c("Chicago Bulls", "Golden State Warriors")  
> wins <- c(72, 73)  
> losses <- c(10, 9)  
> is\_champion <- c(TRUE, FALSE)  
> season <- c("1995-96", "2015-16")  
>   
> # 既有資料框，選擇不要將文字變數記錄為因素向量  
> great\_nba\_teams <- data.frame(team\_name, wins, losses, is\_champion, season, stringsAsFactors = FALSE)  
>   
> # 新增觀測值  
> lakers\_7172 <- c("Los Angeles Lakers", 69, 13, TRUE, "1971-72")  
> great\_nba\_teams <- rbind(great\_nba\_teams, lakers\_7172)  
> great\_nba\_teams  
 team\_name wins losses is\_champion season  
1 Chicago Bulls 72 10 TRUE 1995-96  
2 Golden State Warriors 73 9 FALSE 2015-16  
3 Los Angeles Lakers 69 13 TRUE 1971-72

#### 刪除觀測值

與解構資料框的方式相同：

> team\_name <- c("Chicago Bulls", "Golden State Warriors")  
> wins <- c(72, 73)  
> losses <- c(10, 9)  
> is\_champion <- c(TRUE, FALSE)  
> season <- c("1995-96", "2015-16")  
>   
> # 既有資料框  
> great\_nba\_teams <- data.frame(team\_name, wins, losses, is\_champion, season)  
>   
> # 刪除觀測值  
> great\_nba\_teams <- great\_nba\_teams[-2, ]  
> great\_nba\_teams  
 team\_name wins losses is\_champion season  
1 Chicago Bulls 72 10 TRUE 1995-96

### 重新命名變數

利用 names() 函數為資料框的變數重新命名：

> team\_name <- c("Chicago Bulls", "Golden State Warriors")  
> wins <- c(72, 73)  
> losses <- c(10, 9)  
> is\_champion <- c(TRUE, FALSE)  
> season <- c("1995-96", "2015-16")  
>   
> # 既有資料框  
> great\_nba\_teams <- data.frame(team\_name, wins, losses, is\_champion, season)  
>  
> # 重新命名 team\_name 為 team  
> names(great\_nba\_teams)[1] <- "team"  
> great\_nba\_teams  
 team wins losses is\_champion season  
1 Chicago Bulls 72 10 TRUE 1995-96  
2 Golden State Warriors 73 9 FALSE 2015-16

這個範例將 team\_name 重新命名為 team。

### 調整變數的位置

調整變數位置為：season、is\_champion、wins、losses 與 team\_name。

> team\_name <- c("Chicago Bulls", "Golden State Warriors")  
> wins <- c(72, 73)  
> losses <- c(10, 9)  
> is\_champion <- c(TRUE, FALSE)  
> season <- c("1995-96", "2015-16")  
>   
> # 既有資料框  
> great\_nba\_teams <- data.frame(team\_name, wins, losses, is\_champion, season)  
  
> # 調整變數的位置  
> great\_nba\_teams <- great\_nba\_teams[, c("season", "is\_champion", "wins", "losses", "team\_name")]  
> great\_nba\_teams  
 season is\_champion wins losses team\_name  
1 1995-96 TRUE 72 10 Chicago Bulls  
2 2015-16 FALSE 73 9 Golden State Warriors

### 對類別變數重新編碼

將 great\_nba\_teams$is\_champion 的邏輯值重新編碼為 "Y" 與 "N"：

> team\_name <- c("Chicago Bulls", "Golden State Warriors")  
> wins <- c(72, 73)  
> losses <- c(10, 9)  
> is\_champion <- c(TRUE, FALSE)  
> season <- c("1995-96", "2015-16")  
>   
> # 既有資料框  
> great\_nba\_teams <- data.frame(team\_name, wins, losses, is\_champion, season)  
>  
> # 重新編碼  
> great\_nba\_teams$is\_champion[great\_nba\_teams$is\_champion == TRUE] <- "Y"  
> great\_nba\_teams$is\_champion[great\_nba\_teams$is\_champion == FALSE] <- "N"  
> great\_nba\_teams  
 team\_name wins losses is\_champion season  
1 Chicago Bulls 72 10 Y 1995-96  
2 Golden State Warriors 73 9 N 2015-16

### 對數值變數重新編碼為類別變數

利用 cut() 函數將 straw\_hat\_df$age 的數值重新編碼為 "20 歲以下"、"超過 20 歲但 30 歲以下" 與 "超過 30 歲"，cut() 函數的參數設定稍微複雜一點，使用者必須要針對不同的類別指定切點 breaks，舉例來說，要將年齡區分為三種類型，那麼我們必須要給定四個年齡切點（就像是植樹或者路燈問題！）：c(0, 20, 30, Inf)，其中 Inf 是 R 語言內建的無限大數值；以及給予三個類型標籤 labels：c("20 歲以下", "超過 20 歲但 30 歲以下", "超過 30 歲")

> # 既有的資料框  
> name <- c("蒙其·D·魯夫", "羅羅亞·索隆", "娜美", "騙人布", "賓什莫克·香吉士", "多尼多尼·喬巴", "妮可·羅賓", "佛朗基", "布魯克")  
> age <- c(19, 21, 20, 19, 21, 17, 30, 36, 90)  
> straw\_hat\_df <- data.frame(name, age)  
>   
> # 將 straw\_hat\_df$age 重新編碼  
> straw\_hat\_df$age\_category <- cut(straw\_hat\_df$age, breaks = c(0, 20, 30, Inf), labels = c("20 歲以下", "超過 20 歲但 30 歲以下", "超過 30 歲"))  
> straw\_hat\_df  
 name age age\_category  
1 蒙其·D·魯夫 19 20 歲以下  
2 羅羅亞·索隆 21 超過 20 歲但 30 歲以下  
3 娜美 20 20 歲以下  
4 騙人布 19 20 歲以下  
5 賓什莫克·香吉士 21 超過 20 歲但 30 歲以下  
6 多尼多尼·喬巴 17 20 歲以下  
7 妮可·羅賓 30 超過 20 歲但 30 歲以下  
8 佛朗基 36 超過 30 歲  
9 布魯克 90 超過 30 歲

### 合併資料框

#### 垂直合併

垂直合併使用 rbind() 函數，**rbind** 是 **row bind** 的縮寫：

> cars\_upper <- cars[1:25, ]  
> cars\_bottom <- cars[26:50, ]  
>   
> cars\_combined <- rbind(cars\_upper, cars\_bottom)

#### 水平合併

單純的水平合併使用 cbind() 函數，**cbind** 是 **column bind** 的縮寫：

> cars\_left <- cars[, 1]  
> cars\_right <- cars[, 2]  
>  
> cars\_combined <- cbind(cars\_left, cars\_right)

如果是要利用某一個變數作為對照的依據，進行類似 Excel 的 **vlookup/hlookup** 或者 SQL 的 **join**，可以使用 merge() 函數：

> # 左邊的資料框  
> name <- c("蒙其·D·魯夫", "羅羅亞·索隆", "娜美", "多尼多尼·喬巴")  
> age <- c(19, 21, 20, 17)  
> left\_df <- data.frame(name, age)  
>   
> # 右邊的資料框  
> name <- c("蒙其·D·魯夫", "多尼多尼·喬巴", "妮可·羅賓")  
> devil\_fruit <- c("橡膠果實", "人人果實", "花花果實")  
> right\_df <- data.frame(name, devil\_fruit)  
>   
> # 合併後的資料框  
> merged\_df <- merge(left\_df, right\_df)  
> merged\_df  
 name age devil\_fruit  
1 多尼多尼·喬巴 17 人人果實  
2 蒙其·D·魯夫 19 橡膠果實

觀察合併後的資料框，我們可以發現只有左邊與右邊都有的兩個觀測值保留在最後的輸出，這是因為 merge() 函數預設是保留左右資料框的交集。假如我們希望保留所有**左邊**資料框的觀測值，我們可以指定參數 all.x = TRUE，那麼右邊資料框對應不到部分，就會以遺漏值 NA 呈現：

> # 左邊的資料框  
> name <- c("蒙其·D·魯夫", "羅羅亞·索隆", "娜美", "多尼多尼·喬巴")  
> age <- c(19, 21, 20, 17)  
> left\_df <- data.frame(name, age)  
>   
> # 右邊的資料框  
> name <- c("蒙其·D·魯夫", "多尼多尼·喬巴", "妮可·羅賓")  
> devil\_fruit <- c("橡膠果實", "人人果實", "花花果實")  
> right\_df <- data.frame(name, devil\_fruit)  
>   
> # 合併後的資料框  
> merged\_df <- merge(left\_df, right\_df, all.x = TRUE)  
> merged\_df  
 name age devil\_fruit  
1 多尼多尼·喬巴 17 人人果實  
2 娜美 20 <NA>  
3 羅羅亞·索隆 21 <NA>  
4 蒙其·D·魯夫 19 橡膠果實

假如我們希望保留所有**右邊**資料框的觀測值，我們可以指定參數 all.y = TRUE，那麼左邊資料框對應不到部分，就會以遺漏值 NA 呈現：

> # 左邊的資料框  
> name <- c("蒙其·D·魯夫", "羅羅亞·索隆", "娜美", "多尼多尼·喬巴")  
> age <- c(19, 21, 20, 17)  
> left\_df <- data.frame(name, age)  
>   
> # 右邊的資料框  
> name <- c("蒙其·D·魯夫", "多尼多尼·喬巴", "妮可·羅賓")  
> devil\_fruit <- c("橡膠果實", "人人果實", "花花果實")  
> right\_df <- data.frame(name, devil\_fruit)  
>   
> # 合併後的資料框  
> merged\_df <- merge(left\_df, right\_df, all.y = TRUE)  
> merged\_df  
 name age devil\_fruit  
1 多尼多尼·喬巴 17 人人果實  
2 蒙其·D·魯夫 19 橡膠果實  
3 妮可·羅賓 NA 花花果實

假如我們希望保留**兩邊**資料框的觀測值，我們可以指定參數 all.x = TRUE 與 all.y = TRUE，對應不到部分一樣會以遺漏值 NA 呈現：

> # 左邊的資料框  
> name <- c("蒙其·D·魯夫", "羅羅亞·索隆", "娜美", "多尼多尼·喬巴")  
> age <- c(19, 21, 20, 17)  
> left\_df <- data.frame(name, age)  
>   
> # 右邊的資料框  
> name <- c("蒙其·D·魯夫", "多尼多尼·喬巴", "妮可·羅賓")  
> devil\_fruit <- c("橡膠果實", "人人果實", "花花果實")  
> right\_df <- data.frame(name, devil\_fruit)  
>   
> # 合併後的資料框  
> merged\_df <- merge(left\_df, right\_df, all.x = TRUE, all.y = TRUE)  
> merged\_df  
 name age devil\_fruit  
1 多尼多尼·喬巴 17 人人果實  
2 娜美 20 <NA>  
3 羅羅亞·索隆 21 <NA>  
4 蒙其·D·魯夫 19 橡膠果實  
5 妮可·羅賓 NA 花花果實

前面的範例中，由於我們左邊與右邊資料框用來作為對照依據的變數都取名為 name，所以 merge() 函數會自動辨認，但是當取名不同的時候，就會出現所謂的**笛卡爾連接（Cartesian Join）**：

> # 左邊的資料框  
> name <- c("蒙其·D·魯夫", "羅羅亞·索隆", "娜美", "多尼多尼·喬巴")  
> age <- c(19, 21, 20, 17)  
> left\_df <- data.frame(name, age)  
>   
> # 右邊的資料框  
> character <- c("蒙其·D·魯夫", "多尼多尼·喬巴", "妮可·羅賓")  
> devil\_fruit <- c("橡膠果實", "人人果實", "花花果實")  
> right\_df <- data.frame(character, devil\_fruit)  
>   
> # 合併後的資料框  
> merged\_df <- merge(left\_df, right\_df)  
> merged\_df  
 name age character devil\_fruit  
1 蒙其·D·魯夫 19 蒙其·D·魯夫 橡膠果實  
2 羅羅亞·索隆 21 蒙其·D·魯夫 橡膠果實  
3 娜美 20 蒙其·D·魯夫 橡膠果實  
4 多尼多尼·喬巴 17 蒙其·D·魯夫 橡膠果實  
5 蒙其·D·魯夫 19 多尼多尼·喬巴 人人果實  
6 羅羅亞·索隆 21 多尼多尼·喬巴 人人果實  
7 娜美 20 多尼多尼·喬巴 人人果實  
8 多尼多尼·喬巴 17 多尼多尼·喬巴 人人果實  
9 蒙其·D·魯夫 19 妮可·羅賓 花花果實  
10 羅羅亞·索隆 21 妮可·羅賓 花花果實  
11 娜美 20 妮可·羅賓 花花果實  
12 多尼多尼·喬巴 17 妮可·羅賓 花花果實

這樣的結果很明顯不是我們想要的，這時只要加入 by.x = "name" 與 by.y = "character" 就可以解決：

> # 左邊的資料框  
> name <- c("蒙其·D·魯夫", "羅羅亞·索隆", "娜美", "多尼多尼·喬巴")  
> age <- c(19, 21, 20, 17)  
> left\_df <- data.frame(name, age)  
>   
> # 右邊的資料框  
> character <- c("蒙其·D·魯夫", "多尼多尼·喬巴", "妮可·羅賓")  
> devil\_fruit <- c("橡膠果實", "人人果實", "花花果實")  
> right\_df <- data.frame(character, devil\_fruit)  
>   
> # 合併後的資料框  
> merged\_df <- merge(left\_df, right\_df, by.x = "name", by.y = "character")  
> merged\_df  
 name age devil\_fruit  
1 多尼多尼·喬巴 17 人人果實  
2 蒙其·D·魯夫 19 橡膠果實

### 小結

好啦！第十四天的內容就是這麼多，我們今天統整的是 R 語言常見的資料處理技巧，部分內容是我們早就嫻熟的技巧，相信您的掌握程度應該相當高！

### 練習題

###### 在 favorite\_bands\_df 新增一個變數 formed\_years，這個變數以系統日期計算樂團成立幾週年。

favorite\_bands\_df <- data.frame(band = c("Beyond", "Beatles"), lead\_vocal = c("Wong Ka Kui", "John Lennon"), formed = c(1983, 1960))

### 延伸資訊

* [R Graphics Cookbook](http://shop.oreilly.com/product/0636920023135.do)