

Functions and Packages

Ting-Shuo Yo

October 7, 2016

函數與套件

函數 Function

使用 `function()` 宣告一個R物件，即可製作函數。

```
f <- function(<arguments>) {  
    ## Do something interesting  
}
```

在 R 語言中，函數屬於物件的一種，因此有以下特性：

- 可以被當成參數，傳遞給其他函數
- 可以存在另一個函數當中
- 如未特別指定，最後一行的結果會自動被當作回傳值

R 的內建函數

R 內建了豐富的數學和統計函數，例如：

- `sqrt(value)` = 開根號
- `sum(numbers) / mean(numbers)` = 加總/平均數
- `var(numbers) / sd(numbers)` = 變異數 / 標準差
- `cor(A, B) / cov(A,B)` = A 與 B 兩串資料的相關係數 / 共變數
- `prcomp(matrix)` = 主成分分析
- `fa(matrix)` = 因素分析

自訂函數 1/2

回顧剛剛看過的例子：

```
for( x in 88:92){  
  if(x > 90){  
    grade <- "A"  
  } else if(x > 80){  
    grade <- "B"  
  } else {  
    grade <- "C"  
  }  
  print(c(x, grade))  
}
```

```
## [1] "88" "B"  
## [1] "89" "B"  
## [1] "90" "B"  
## [1] "91" "A"  
## [1] "92" "A"
```

自訂函數 2/2

把分數換成等第的功能，我們用了很多次，可以獨立出來寫成函數：

```
grading <- function(x){  
  if(x > 90){  
    grade <- "A"  
  } else if(x > 80){  
    grade <- "B"  
  } else {  
    grade <- "C"  
  }  
  print(c(x, grade))  
}  
  
for(x in 88:92){  
  grading(x)  
}
```

```
## [1] "88" "B"  
## [1] "89" "B"  
## [1] "90" "B"  
## [1] "91" "A"  
## [1] "92" "A"
```


函數的參數 Function Arguments

函數都有參數（`arguments`），而參數可以有預設值（`default`）。

- 正式參數（`formal arguments`）指的是在函數定義時已經明確標出的參數。
- 可以用 `formals` 函數來列出一個函數的正式參數。
- 呼叫函數的時候，不一定會用到所有的正式參數。
- 函數的參數可以 `NULL` 或是有預設值。

參數比對 Argument Matching

R 函數的參數可以用「位置」或「名稱」來比對。以下對 `sd` 函數的呼叫都是一樣的：

```
> mydata <- rnorm(100)
> sd(mydata)
> sd(x = mydata)
> sd(x = mydata, na.rm = FALSE)
> sd(na.rm = FALSE, x = mydata)
> sd(na.rm = FALSE, mydata)
```

雖然參數的先後可以不依照順序，但是建議還是盡量依序排列，以利閱讀。

參數比對 Argument Matching

「位置比對」和「名稱比對」是可以混用的，有名稱的會優先比對，剩下的依照位置順序來比對。以 `lm` 函數為例（linear model 線性模型）：

```
> args(lm)
function (formula, data, subset, weights, na.action,
  method = "qr", model = TRUE, x = FALSE,
  y = FALSE, qr = TRUE, singular.ok = TRUE,
  contrasts = NULL, offset, ...)
```

以下兩種呼叫方式的結果是一樣的：

```
lm(data = mydata, y ~ x, model = FALSE, 1:100)
lm(y ~ x, mydata, 1:100, model = FALSE)
```

函數的定義方式 Defining a Function

```
f <- function(a, b = 1, c = 2, d = NULL) {  
  
}
```

參數可以指定預設值，或是不指定預設值，還可以設為 **NULL**.

The "...” Argument

參數裡的 ... 表示「有一些其他參數」，通常是用來進一步傳給其他函數。

- 當你的函數會呼叫另一個函數，但是又不想把所有的參數寫出來的時候，可以用... 來表示。

```
myplot <- function(x, y, type = "l", ...) {  
  plot(x, y, type = type, ...)  
}
```

- 很多內建函數都有 ... 參數，以便用來做為中間函數。

```
> mean  
function (x, ...)  
UseMethod("mean")
```

The "...” Argument

有時後我們無法事先得知有哪些參數會被使用到，此時 ... 就是必要的參數。

```
> args(paste)
function (... , sep = " ", collapse = NULL)

> args(cat)
function (... , file = "", sep = " ", fill = FALSE,
          labels = NULL, append = FALSE)
```

Arguments Coming After the "...” Argument

在 ... 後面的參數必須明確指定名稱。

```
> args(paste)
function (... , sep = " ", collapse = NULL)
```

```
> paste("a", "b", sep = ":")
[1] "a:b"
```

```
> paste("a", "b", se = ":")
[1] "a b :"
```

R 套件管理

R 的系統設計

- 主要分成兩大部分：
 - R 的基本核心（R-base）
 - 其他
- 各種功能被分散在眾多「套件」之中（packages）
 - ***base R*** 包含核心功能與基本函數
 - 基本安裝也包含 **util, stats, datasets** 等套件
 - 依據應用領域的不同，會有不同的推薦套件
 - 在 **CRAN** 有登記的套件超過 5000 種以上

套件安裝

- 在 Rstudio 透過圖形介面安裝套件
- 透過 console command 安裝

```
install.packages("ggplot2")
```

```
## Installing package into 'C:/Users/Ting-Shuo/Documents/R/win-library/3.3'
```

```
## (as 'lib' is unspecified)
```

```
## package 'ggplot2' successfully unpacked and MD5 sums checked
```

```
##
```

```
## The downloaded binary packages are in
```

```
## D:\TempFiles\TMP\Rtmp40G1T6\downloaded_packages
```


套件版本

- `installed.packages()`指令可以顯示已經安裝的套件

	Package	LibPath	Version
base	"base"	"C:/PROGRA~1/R/R-27~1.1/library"	"2.7.1"
boot	"boot"	"C:/PROGRA~1/R/R-27~1.1/library"	"1.2-33"
class	"class"	"C:/PROGRA~1/R/R-27~1.1/library"	"7.2-42"
cluster	"cluster"	"C:/PROGRA~1/R/R-27~1.1/library"	"1.11.11"
codetools	"codetools"	"C:/PROGRA~1/R/R-27~1.1/library"	"0.2-1"
datasets	"datasets"	"C:/PROGRA~1/R/R-27~1.1/library"	"2.7.1"
foreign	"foreign"	"C:/PROGRA~1/R/R-27~1.1/library"	"0.8-26"

- 升級新版本的 R 有時需要重新安裝套件

常用套件介紹

2015年上半年的資料，最受歡迎的 R 套件：

1. **Rcpp** 提供與 C++ 的整合，提昇計算效率。
2. **ggplot2** 是最受歡迎的進階繪圖套件，背後有 Grammar of Graphics 這個理論基礎。
3. **stringr** 提供了文字資料處理的常用函數。
4. **plyr** 為資料的分割、合併與會整提供了豐富的工具。
5. **digest** 提供為資料加密的功能。

我想用特定的分析方法分析我的資料

例如，我想用階層線性回歸（r hierarchical linear model）

- [Google](#) 是你的好幫手
 - Hierarchical linear models and lmer | R-bloggers
 - Hierarchical Linear Model | R Tutorial
 - Hierarchical Linear Regression - R Tutorial Series
- 通常你會在這些地方找到答案：
 - [CRAN](#) (The Comprehensive R Archive Network)
 - [R-blogger](#)
 - [Stackoverflow](#)

套件新聞

R 現在已經是個很大的社群，隨時有新的套件和應用發佈。

[R Journal](#) 是個線上期刊，半年一期，有很多基於特定套件做某種分析的文章，可以理解套件的特性與使用情境。例如：

- [Maps, Coordinate Reference Systems and Visualising Geographic Data with mapmisc](#) by Patrick Brown

[R-bloggers](#) 則是比較新聞性的網站，除了社群動態之外，也有很多的教學文章，例如：

- [Get air quality data for the United Kingdom using the rdefra package](#) by Claudia Vitolo