# 第 11 天

## 資料的輸入與輸出

來到這裡我們可以鬆一口氣了！怎麼說呢？因為基本功部分已經講完了！假如您跟著我們紮實地學習了前面的內容，接下來您應該會有水到渠成的感覺。在前面的章節中我們一直利用手動的方式建立單純的資料來學習，然而在現實生活中，我們的應用情形比較多是**輸入**一組資料再針對它進行資料的清理或者分析，最後再將結果**輸出**成一個本機路徑中的檔案。

### 內建資料

R 語言有非常豐富的內建資料，這些資料絕大多數都是以資料框（data.frame）為結構儲存，並且在我們開啟 R 語言之後就可以使用，想要知道有哪些內建資料可以使用，只要在命令列（Console）輸入：

data()

就可以在來源（Source）區塊瀏覽一份內建資料集的清單：

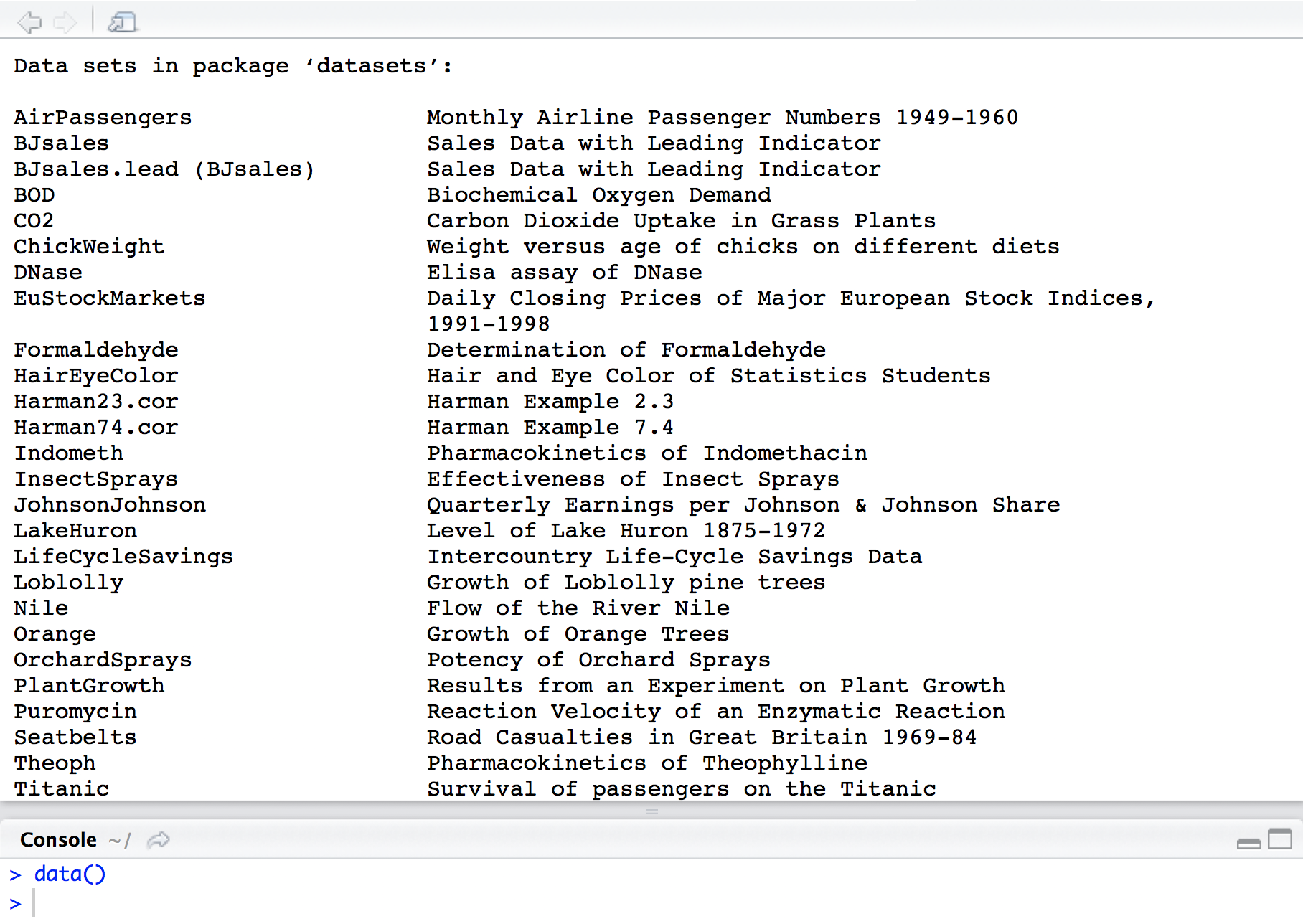


圖 11-1 內建資料集

內建資料集是讓我們可以快速測試函數的好幫手，常常被眾多 R 語言使用者用來測試的內建資料集有 iris、cars、mtcars 與 airquality 等等。如果對這些內建資料集感到好奇，我們可以像查詢函數一般來查詢它們，右下角的查詢區塊就會顯示詳細資料：

?iris # 或者輸入 help(iris)

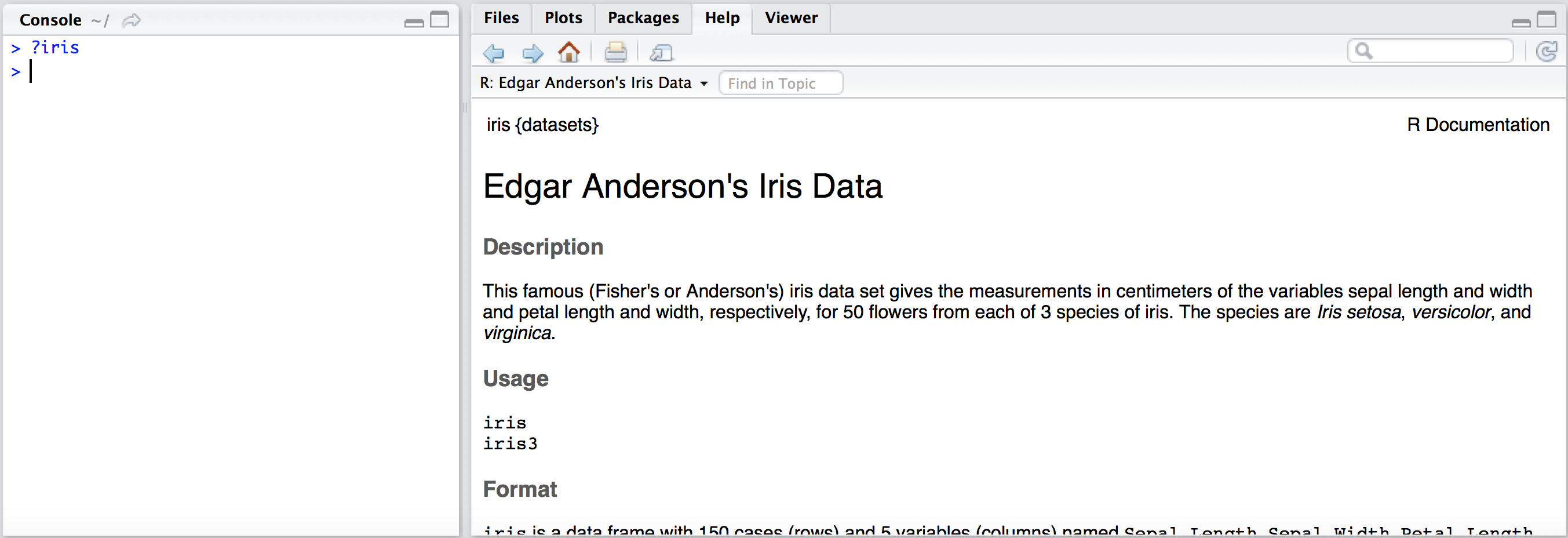


圖 11-2 查詢內建資料集

### 輸入資料

#### read.table() 函數

如果想要輸入一個表格式的資料（tabular data），我們可以使用 read.table() 函數，表格式的資料中每一筆代表一個觀測值，變數以符號區隔（常見有空格、tab 或逗號。）

假設我們的表格式資料（favorite\_bands.txt）是以空格區隔變數：

band lead\_voal formed  
"Beyond" "Wong Ka Kui" 1983  
"Beatles" "John Lennon" 1960

讀入這個資料的寫法是：

file\_path <- "/Users/username/favorite\_bands.txt"  
favorite\_bands <- read.table(file\_path, header = TRUE, stringsAsFactors = FALSE)  
View(favorite\_bands)

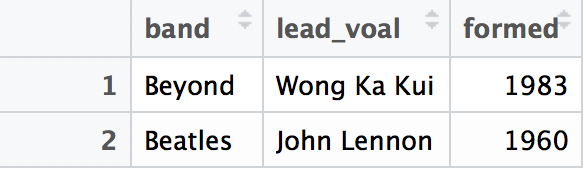


圖 11-3 讀入 .txt 檔案

這裡值得注意的地方有三個：

1. file\_path 參數指定資料的檔案路徑，這個範例是假設我將表格式資料（favorite\_bands.txt）儲存在 /Users/username 的路徑下。如果是 windows 系統的使用者，需要寫成 C:/Users/username/favorite\_bands.txt，注意斜線的方向，windows 系統慣用的 \ 是不被認得的，必須要修改為 /
2. header 參數指定資料的第一筆是否為變數名稱
3. stringsAsFactors 參數指定資料中的文字是否要儲存為因素向量的資料結構

假設我們的表格是資料（favorite\_bands.tsv）是以 tab 區隔變數，副檔名 .tsv 是 tab-separated values 的縮寫：

band lead\_voal formed  
"Beyond" "Wong Ka Kui" 1983  
"Beatles" "John Lennon" 1960

讀入這個資料的寫法也不需要特別修改：

file\_path <- "/Users/username/favorite\_bands.tsv"  
favorite\_bands <- read.table(file\_path, header = TRUE, stringsAsFactors = FALSE)  
View(favorite\_bands)

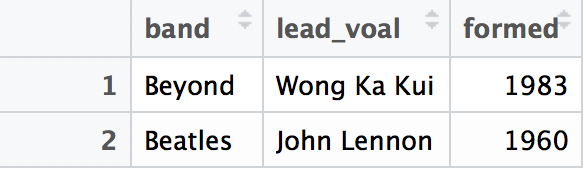


圖 11-4 讀入 .tsv 檔案

假設我們的表格是資料（favorite\_bands.csv）是以逗號區隔變數，副檔名 .csv 是 comma-separated values 的縮寫：

band,lead\_voal,formed  
"Beyond","Wong Ka Kui",1983  
"Beatles","John Lennon",1960

讀入這個資料的寫法需要修改 sep 參數的設定，因為預設 sep 參數只能辨識一個或多個空格，既然我們現在改以逗號區隔變數，就得指定對應的分隔符號：

file\_path <- "/Users/username/favorite\_bands.csv"  
favorite\_bands <- read.table(file\_path, header = TRUE, stringsAsFactors = FALSE, sep = ",")  
View(favorite\_bands)

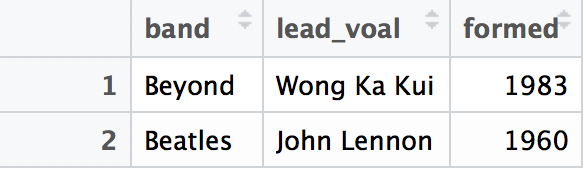


圖 11-5 讀入 .csv 檔案

前面的範例都是從本機路徑讀取檔案，其實 read.table() 函數也可以從網路讀取表格式資料，比如說：

url <- "https://storage.googleapis.com/learn-r-the-easy-way.appspot.com/data\_ch11/favorite\_bands.csv"  
favorite\_bands <- read.table(url, header = TRUE, stringsAsFactors = FALSE, sep = ",")  
View(favorite\_bands)

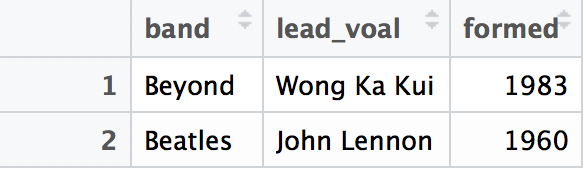


圖 11-6 從雲端讀入 .csv 檔案

我們只是修改了資料所在路徑，從一個本機路徑（file\_path）替換為一個網路位址（url）。[加州大學爾灣分校資料儲存庫](https://archive.ics.uci.edu/ml/index.html)中有為數眾多的資料集，這些資料集多是以表格式資料儲存，我們同樣可以直接使用 read.table() 函數直接將這個儲存庫的資料讀入。舉例來說，我們來讀取一個 [Wine Quality Data Set](https://archive.ics.uci.edu/ml/datasets/Wine+Quality)，它的資料儲存在 <https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/> 這個位址，資料第一筆是變數名稱，區隔符號為冒號：

url <- "https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/winequality-red.csv"  
wine\_quality\_red <- read.table(url, header = TRUE, sep = ":")  
View(wine\_quality\_red)

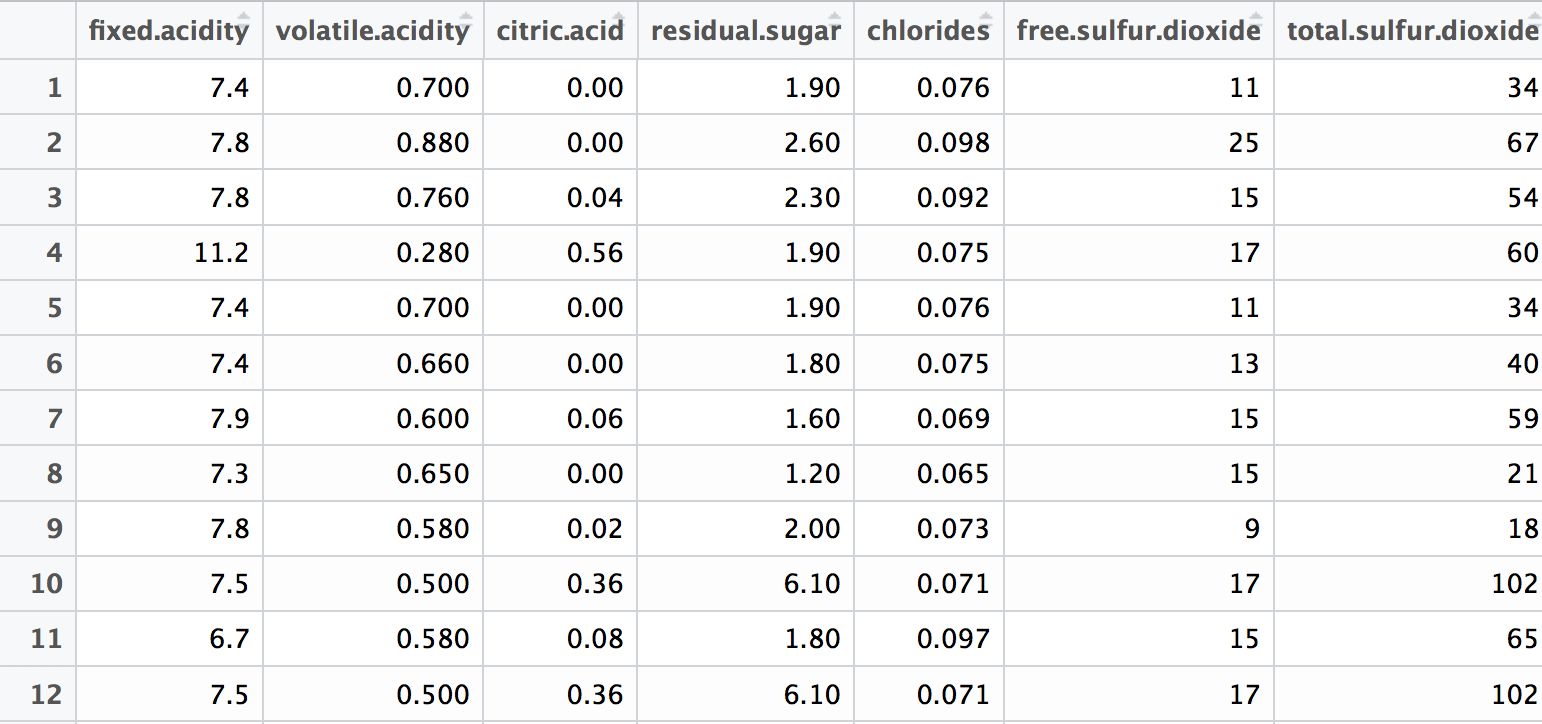


圖 11-7 從 UCI ML Repository 讀入 .csv 檔案

#### readLines() 函數

並非所有我們想讀入的檔案都是以分隔符號簡潔分隔的表格式資料，像是劇本：

Ross: (mortified) Hi.  
Joey: This guy says hello, I wanna kill myself.  
Monica: Are you okay, sweetie?  
Ross: I just feel like someone reached down my throat, grabbed my small intestine, pulled it out of my mouth and tied it around my neck...  
Chandler: Cookie?  
Monica: (explaining to the others) Carol moved her stuff out today.  
Joey: Ohh.  
Monica: (to Ross) Let me get you some coffee.  
Ross: Thanks.  
Phoebe: Ooh! Oh! (She starts to pluck at the air just in front of Ross.)  
Ross: No, no don't! Stop cleansing my aura! No, just leave my aura alone, okay?  
Phoebe: Fine! Be murky!

這時我們可以利用 readLines() 一筆一筆讀入這份資料：

> file\_path <- "/Users/username/friends\_script.txt"  
> friends\_script <- readLines(file\_path)  
> friends\_script  
 [1] "Ross: (mortified) Hi."   
 [2] "Joey: This guy says hello, I wanna kill myself."   
 [3] "Monica: Are you okay, sweetie?"   
 [4] "Ross: I just feel like someone reached down my throat, grabbed my small intestine, pulled it out of my mouth and tied it around my neck..."  
 [5] "Chandler: Cookie?"   
 [6] "Monica: (explaining to the others) Carol moved her stuff out today."   
 [7] "Joey: Ohh."   
 [8] "Monica: (to Ross) Let me get you some coffee."   
 [9] "Ross: Thanks."   
[10] "Phoebe: Ooh! Oh! (She starts to pluck at the air just in front of Ross.)"   
[11] "Ross: No, no don't! Stop cleansing my aura! No, just leave my aura alone, okay?"   
[12] "Phoebe: Fine! Be murky!"

readLines() 函數回傳的物件是一個文字的向量，每個索引值就是原始文字檔中的一筆資訊，假使原始文字檔非常龐大，我們可以加入參數 n 來限定讀入的筆數：

> file\_path <- "/Users/username/friends\_script.txt"  
> friends\_script <- readLines(file\_path, n = 2)  
> friends\_script  
[1] "Ross: (mortified) Hi."   
[2] "Joey: This guy says hello, I wanna kill myself."

#### 常見資料格式的輸入

常見用來儲存資料的格式還有 Excel 試算表與 JSON，我們必須仰賴**套件**來輔助我們輸入這些資料格式：

|  |  |  |
| --- | --- | --- |
| 資料格式 | 套件名稱 | 主要開發者 |
| Excel 試算表 | readxl | Hadley Wickham |
| JSON | jsonlite | Jeroen Ooms |

R 語言使用套件的程序有兩個階段，一個是安裝，另外一個是載入。這兩者的區別就像是買工具書跟查詢工具書，安裝套件就像是將這本工具書買回家裡放置在書櫃；載入套件就像是有需求時將工具書從書櫃取下查詢。我們使用 install.packages() 函數來安裝套件，然後再使用 library() 函數載入套件。

|  |  |  |  |
| --- | --- | --- | --- |
| 動作 | 函數 | 使用頻率 | 比喻 |
| 安裝套件 | install.pacakges() | 同樣版本執行一次 | 買工具書 |
| 載入套件 | library() | 每次使用套件都要執行 | 查詢工具書 |

##### Excel 試算表

假設我們的 Excel 試算表外觀是這樣：

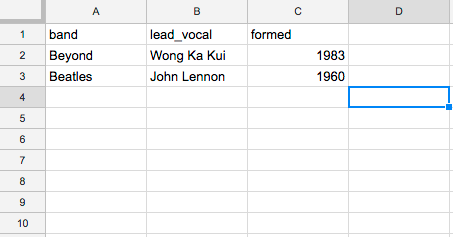


圖 11-8 讀入 .xlsx 檔案

我們可以利用命令列（Console）進行套件的安裝與載入：

# 安裝與載入套件  
install.packages("readxl") # 如果先前已經安裝過就不用執行這行  
library(readxl)

或者利用 RStudio 右下角的 **Packages** 區塊提供的使用者介面來讓我們安裝與載入套件：

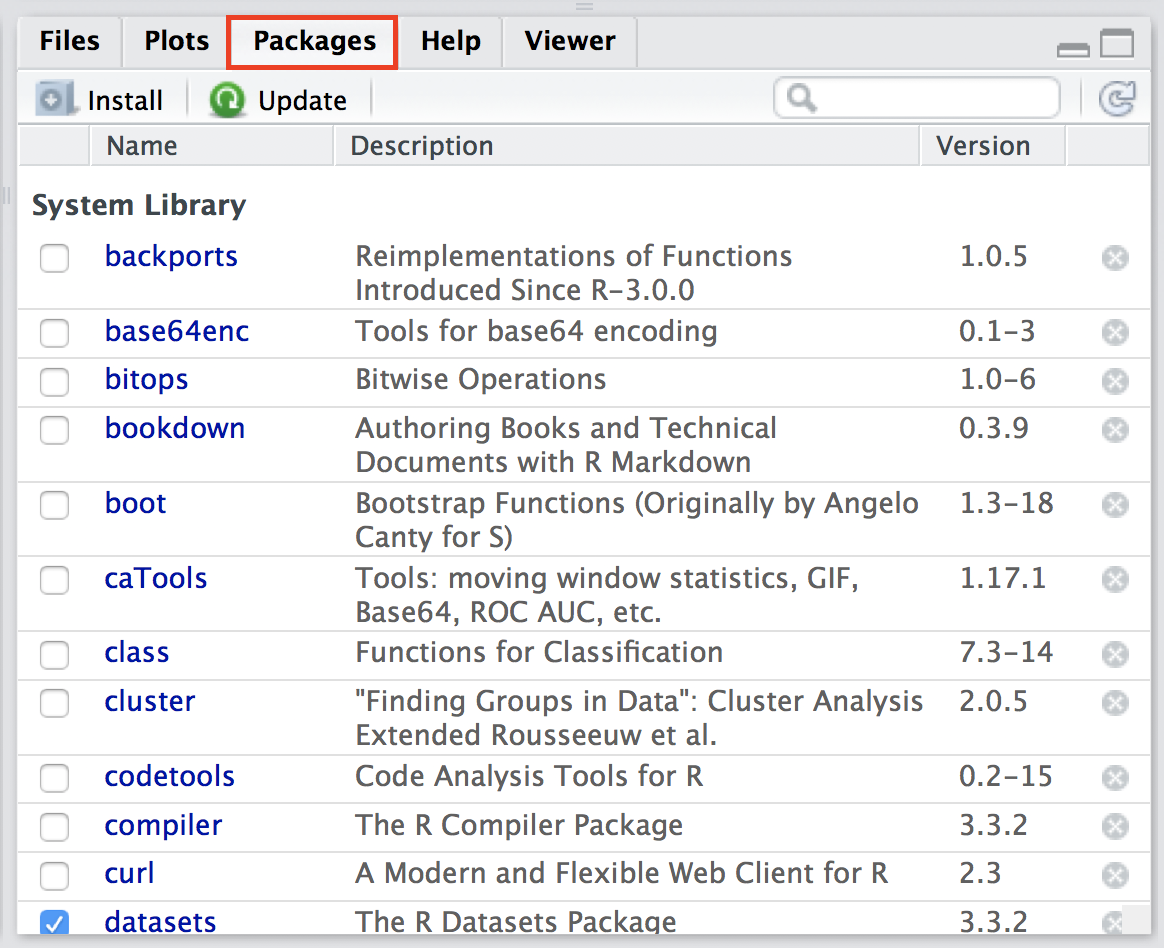


圖 11-9 使用者介面安裝套件

點選 **Install**：

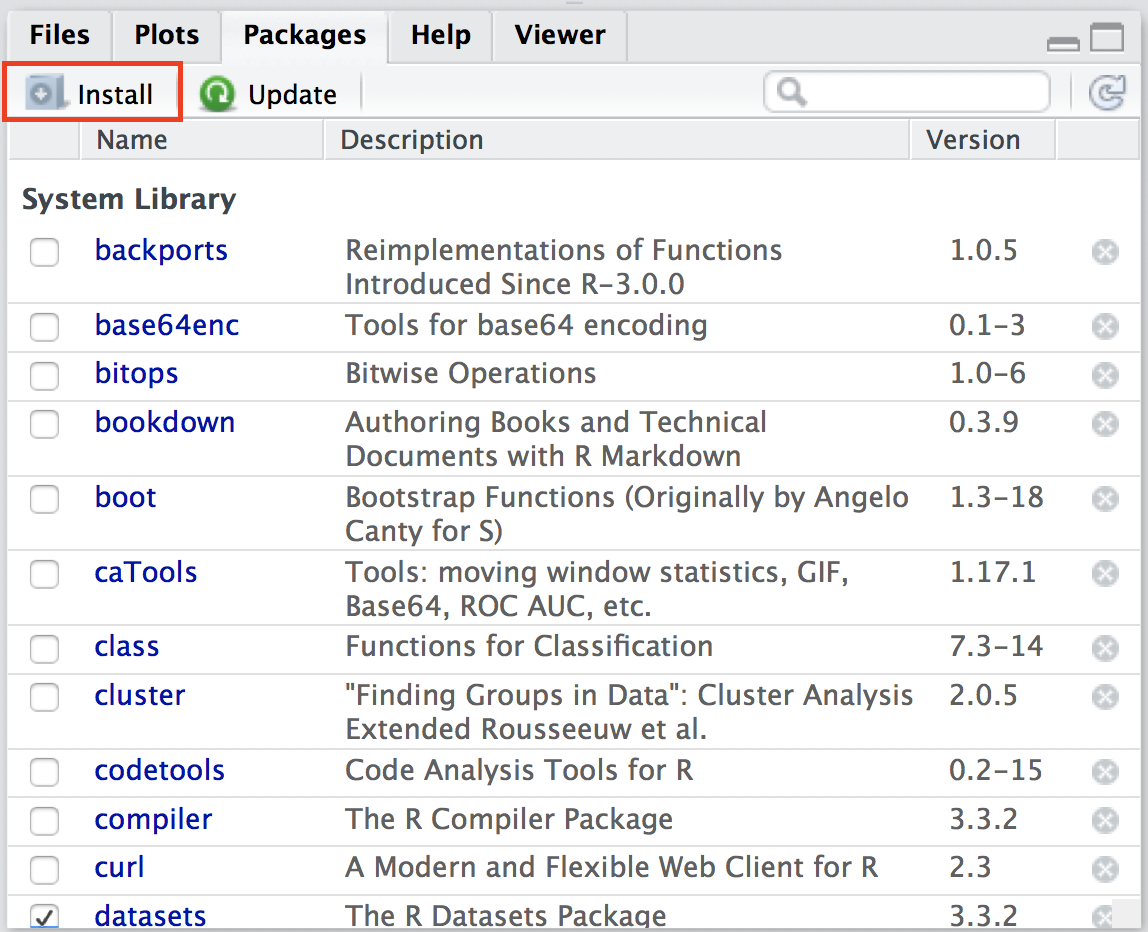


圖 11-10 使用者介面安裝套件（2）

接著輸入要安裝的套件名稱：

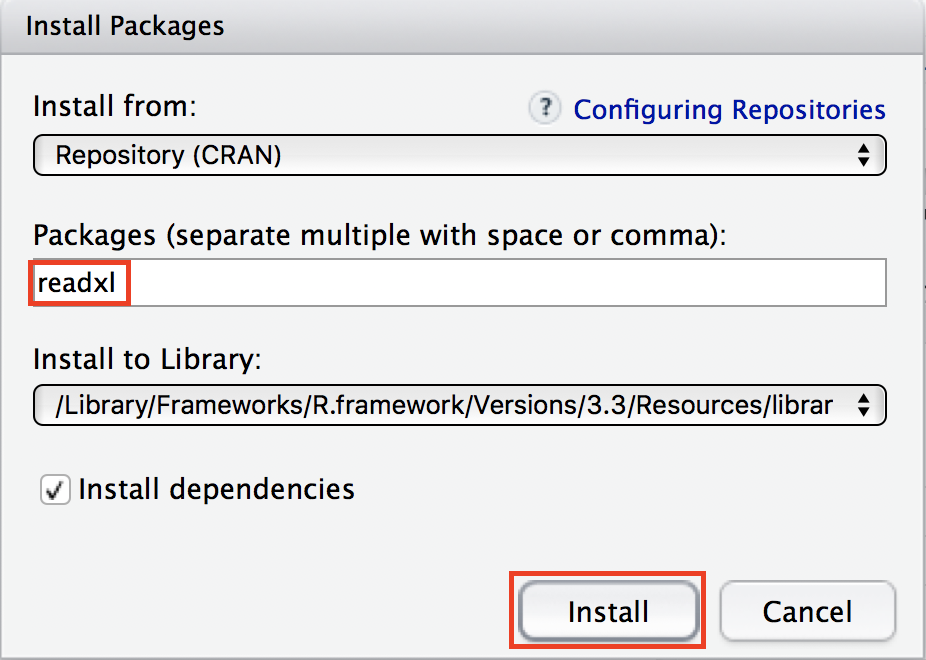


圖 11-11 使用者介面安裝套件（3）

安裝完成之後，把方框勾選起來表示載入這個套件：

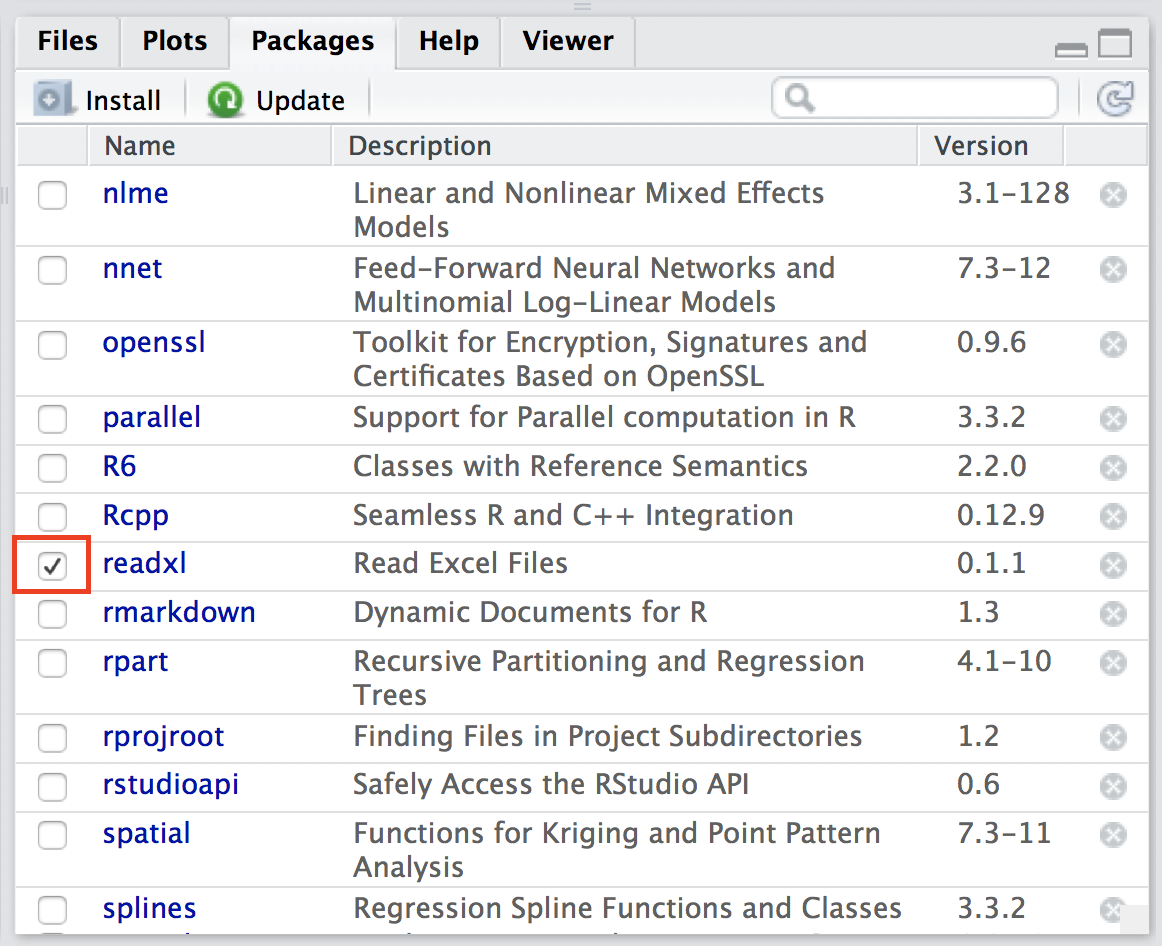


圖 11-12 使用者介面安裝套件（4）

不論是使用指令安裝載入套件，或是以使用者介面安裝載入套件，完成以後都是利用 readxl 套件提供的 read\_excel() 函數讀入 Excel 試算表：

# 讀入  
file\_path <- "/Users/username/favorite\_bands.xlsx"  
favorite\_bands <- read\_excel(file\_path)

##### JSON

JSON 是 JavaScript Object Notation 的縮寫，它的外觀長得像這樣子：

[  
 {"band": "Beyond", "lead\_vocal": "Wong Ka Kui", "formed": 1983},  
 {"band": "Beatles", "lead\_vocal": "John Lennon", "formed": 1960}  
]

我們可以利用 jsonlite 套件提供的 fromJSON() 函數讀入成為資料框：

# 安裝與載入套件  
install.packages("jsonlite") # 如果先前已經安裝過就不用執行這行  
library(jsonlite)  
  
# 讀入  
file\_path <- "/Users/username/favorite\_bands.json"  
favorite\_bands <- fromJSON(file\_path)  
View(favorite\_bands)

如果您想利用使用者介面安裝載入套件，可以模仿前一個 readxl 的範例操作。

### 輸出資料

我們可以將 R 語言的資料框以不同的格式輸出成為純文字檔。

#### write.table() 函數

write.table() 函數預設輸出用空格區隔變數的純文字檔（sep = " "）：

favorite\_bands\_df <- data.frame(band = c("Beyond", "Beatles"), lead\_vocal = c("Wong Ka Kui", "John Lennon"), formed = c(1983, 1960))  
file\_path <- "/Users/username/favorite\_bands.txt"  
write.table(favorite\_bands\_df, file = file\_path, row.names = FALSE)

執行完畢後我們可以在 /Users/username/ 路徑下找到 favorite\_bands.txt 檔案，這裡值得注意的是 file 參數指定的是我們期望輸出檔案的路徑，row.names 參數指定不要將觀測值的索引值輸出，這樣輸出檔案的外觀會比較接近我們平時熟悉的樣子：

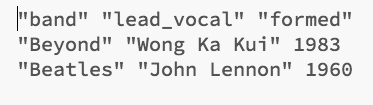


圖 11-13 輸出 .txt

write.table() 函數指定參數 sep = "," 可以輸出逗號區隔的純文字檔：

favorite\_bands\_df <- data.frame(band = c("Beyond", "Beatles"), lead\_vocal = c("Wong Ka Kui", "John Lennon"), formed = c(1983, 1960))  
file\_path <- "/Users/username/favorite\_bands.csv"  
write.table(favorite\_bands\_df, file = file\_path, row.names = FALSE, sep = ",")

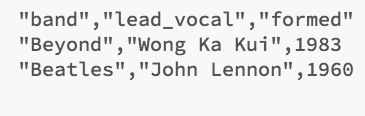


圖 11-14 輸出 .csv

#### toJSON() 函數

我們可以利用 jsonlite 套件提供的 toJSON() 函數將資料框輸出為 JSON：

# 安裝與載入套件  
install.packages("jsonlite") # 如果先前已經安裝過就不用執行這行  
library(jsonlite)  
  
# 輸出  
favorite\_bands\_df <- data.frame(band = c("Beyond", "Beatles"), lead\_vocal = c("Wong Ka Kui", "John Lennon"), formed = c(1983, 1960))  
toJSON(favorite\_bands\_df)

#### writeLines() 函數

toJSON() 函數搭配 writeLines() 函數可以將非表格式資料輸出到本機路徑：

# 安裝與載入套件  
install.packages("jsonlite") # 如果先前已經安裝過就不用執行這行  
library(jsonlite)  
  
# 輸出  
favorite\_bands\_df <- data.frame(band = c("Beyond", "Beatles"), lead\_vocal = c("Wong Ka Kui", "John Lennon"), formed = c(1983, 1960))  
favorite\_bands\_json <- toJSON(favorite\_bands\_df)  
file\_path <- "/Users/username/favorite\_bands.json"  
writeLines(favorite\_bands\_json, con = file\_path)

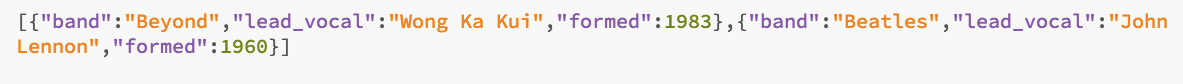


圖 11-15 輸出 .json

### 小結

好啦！第十一天的內容就是這麼多，我們學會了不同格式資料的輸入與輸出，其中在針對 Excel 試算表與 JSON 的資料格式，我們學會如何安裝並且載入套件來協助輸入與輸出。

### 練習題

###### 1. 練習將加州大學爾灣分校資料儲存庫 <http://archive.ics.uci.edu/ml/machine-learning-databases/iris/bezdekIris.data> 的 iris 資料讀入

url <- "http://archive.ics.uci.edu/ml/machine-learning-databases/iris/bezdekIris.data"  
iris\_df <- read.table(\_\_\_, header = \_\_\_, sep = "\_\_\_")

###### 2. 加州大學爾灣分校資料儲存庫的 iris 資料第一筆並不是變數名稱，請根據 <http://archive.ics.uci.edu/ml/machine-learning-databases/iris/iris.names> 這個連結的描述為讀入的資料變數命名

url <- "http://archive.ics.uci.edu/ml/machine-learning-databases/iris/bezdekIris.data"  
iris\_df <- read.table(\_\_\_, header = \_\_\_, sep = "\_\_\_")  
names(iris\_df) <- c("\_\_\_", "\_\_\_", "\_\_\_", "\_\_\_", "\_\_\_")

###### 3. 練習將內建資料 cars 輸出為 cars.csv 至本機路徑，記住要指定參數 row.names = FALSE

file\_path <- "\_\_\_" # 自訂  
write.csv(cars, file = file\_path, row.names = \_\_\_)

### 延伸資訊

* [CRAZY FOR FRIENDS](http://www.livesinabox.com/friends/scripts.shtml)
* [Package readxl](https://cran.r-project.org/web/packages/readxl/readxl.pdf)
* [Package jsonlite](https://cran.r-project.org/web/packages/jsonlite/jsonlite.pdf)
* [HADLEY WICKHAM](http://hadley.nz/)
* [Jeroen Ooms](https://github.com/jeroen)