# 醫學統計與 R

林建甫,
Chien-Fu Jeff Lin, MD. PhD. [1]

September 13, 2006

[1] 國立台北大學統計系 (cflin@mail.ntpu.edu.tw). http://web.ntpu.edu.tw/~cflin

# 第 1 章: R 簡介
# 1: Introduction to R

R 系統是由 Ross Ihaka 與 Robert Gentleman 從 S 語言所發展出來, 主要是爲了統計分析與統計繪圖. R 除了資料處理與分析, R 擁有一完整陣列和矩陣的操作運算, 完整圖形工具, 也是一種相當完善的程式設計語言. S 語言在 1980 年代末期, 由 AT&T 實驗室, Rick Becker, John Chambers, 與 Allan Wilks 發展用來進行統計分析與統計繪圖, Insightful 公司將 S 商品化, 並加入許多方便的操作介面, 稱爲 S-PLUS. R 可視爲統計數學軟體, 也是一種程式語言. R 與 S (或 S-PLUS) 語法大多相近, 但是 R 是一個免費 (open-source, GNU General Public License) 的統計分析軟體, 目前由一群跨國際的志工人員組成的 R 核心發展組織 (R core-development team) 所維持, 運作與持續更新發展. R 計畫的網址在 http://www.r-project.org. 在這網址上可獲的更多有關 R 的資訊. R 與 S 都是以物件導向爲主的程式語言, 透過交互作用方式很容易地進行統計分析與統計繪圖, 這與 SAS, SPSS 的方式有所不同.

## 1.1 下載與安裝 R

R 有各種版本, 可以在 Microsoft Window XP, Unix, Linux, Apple Mac OS 等作業性系統運行, 當今的最新版是 R 2.3.1 (2006-06-01), 約 27.3M, (R 時常有更新版本), 下載與安裝簡述如下:

1. 上網至 http://www.r-project.org
2. 按滑鼠點選網頁左邊連結 (Link) 下載區 **Download "CRAN"**
3. 按滑鼠點選網頁 CRAN Mirrors 中的任一鏡像網址, 如 US http://cran.us.r-project.org/ (Pair Networks, Pittsburgh, PA)
4. 按滑鼠點選網頁 Frequently used pages 中的 **Windows** (95 and later)
5. 按滑鼠點選網頁 R for Windows 中的 **base**
6. 按滑鼠右鍵, 點選網頁, R-2.3.1 for Windows 中的 **R-2.3.1-win32.exe**, 儲存至個人檔案夾內
7. 至下載的檔案夾內, 按滑鼠點擊 R-2.3.1-win32.exe 兩次, 即可進行安裝.
8. 可選擇中文或英文進行安裝.

9. 硬碟空間許可下, base 套件全部安裝.
10. 從桌面, 點擊 R-2.3.1 圖像兩次, 開啓程式.
11. 從程式視窗上端, 點選表單–編輯, 最下方–GUI 偏好設定
12. GUI 偏好設定, 可點選 SDI mode, 改變顏色等.

## 1.2 簡單實例

學習 R 最好的方法, 就是要開始使用 R, 初學者要了解 R, 可先進行一些簡單實例的演練.

> **setwd("X://temp//RData")**

改變執行中的檔案夾目錄

1+2

> **help.start()**

發出 超文字連結-網頁流覽器介面 (html one-line help), 線上協助 (需使用網瀏覽器).

> **x <– rnorm(50)**

> **y <– rnorm(x)**

產生 2 維的隨機變數向量 $\underline{\mathbf{x}}$ 和 $\underline{\mathbf{y}}$.

> **plot(x, y)**

對 $\underline{\mathbf{x}}$ 和 $\underline{\mathbf{y}}$ 做圖, 圖案結果會視窗會自動地出現.

> **ls()**

檢查看 R 工作空間中, 有哪些物件

> **rm(x, y)**

移除不再需要的物件

> **x <– 1:25**

產生向量數列 $\underline{\mathbf{x}} = (1, 2, \ldots, 25)$.

> **z <– x^2**

向量 $\underline{\mathbf{z}}$ 爲 $\underline{\mathbf{x}}$ 的變數轉換, $\underline{\mathbf{z}} = \underline{\mathbf{x}}^2$.

> **y <– (2+rnorm(x)) + (4+rnorm(x))*z + (rnorm(x)*sd(z))**

向量 $\underline{\mathbf{y}}$ 爲 $\underline{\mathbf{z}}$ ($\underline{\mathbf{x}}$) 的線性組合加上隨機變數向量

> **plot(x, y)**

$\underline{\mathbf{x}}$ 對 $\underline{\mathbf{y}}$ 做圖

> **simple.data <– data.frame(x=x, y=y, z=z)**

產生一組資料 (one set of data frame), 名叫 simple.data 內含 $x$, $y$, $z$ 三個變數, 從原先三個向量 $\underline{\mathbf{x}}$, $\underline{\mathbf{y}}$, $\underline{\mathbf{z}}$ 合成.

> **fit.lm <– lm(y  x, data=simple.data)**

配適 $y$ 對 $x$ 簡單的線性迴歸.

> **summary(fit.lm)**

摘要配適的結果

> **rm(x, y, z)**
移除原先三個向量物件, **x**, **y**, **z**.

> **attach(simple.data)**
在 R 中, 貼上 simple.data 資料, 可直接使用 x, y, z.

> **fit.lowess <– lowess(x, y)**
配適一各非參數迴歸分析.

> **plot(x, y)**
**x** 對 **y** 做圖

> **lines(x, fit.lowess$y)**
加上非參數迴歸線.

> **abline(coef(fit.lm), col = "red")**
加上簡單線性迴歸線.

> **detach()** 移除 simple.data 資料, 不可再直使用.

> **plot(fitted(fit.lm), resid(fit.lm),**

\+ **xlab="Fitted Values",**

\+ **ylab="Residuals",**

\+ **main="Residuals v.s. Fitted")**
標準線性回歸殘差圖做診斷, 可見配適不好.

> **qqnorm(resid(fit.lm), main="Residuals QQ Plot")**
檢驗殘差的常態分配狀態

> **rm(fit.lm, fit.lowess, simple.data)**
移除不再需要的物件.

# 1.3 R 常用指令

## 1.3.1 物件命名與指令使用

R 與 S 都是以物件導向爲主的程式語言, R 中, 每一樣 "東西", 都叫做 "**物件**", 物件可以是向量 (vector), 矩陣 (matrix), 陣列 (array), 列表 (Lists), 或 資料框架 (data frames) 等. 透過指令, 很容易地對物件進行統計分析與統計繪圖. 須特別注意, 在 R 指令的英文大小寫有差異, s 與 S 是不同的, R 也保留一些物件與指令名稱, 如 c, s, C, T, codeF 等, 這些叫做 "**保留名字 reserved names**

```
FALSE Inf NA NaN NULL TRUE
break else for function if in next repeat while
F T
```

另外一些系統常用的指令名稱, 如

```
c q s t C D F I T diff mean pi range rank var
```

初學者對物件命名時盡量避免定義一個物件, 與現有的物件同名. 所以命名時要避免重覆, 以免後來引起錯亂. 對物件命名時, 物件名稱起始位置須以文字或 “.” (句點), 若物件名稱以 “.” 爲起始, 名稱第二個位置需爲文字, 物件名稱其餘位置, 以文字 (A-Z 或 a-z) , 數字 (0-9), “/”, “.”, 或 “-”, 皆可. 中間不可有空格或 “_” (underscore).

R 基本介面是一個互動式指令視窗, 指令可分成 **運算式 expression** 如 `1+2` 或 **指派運算 (賦值運算) assignment**, 如 `x<-1+2`. 當一個 R 程式需要你輸入指令時, 它會顯示指令提示號, 指令提示符號通常是一個 `>` (大於符號), 完整 **運算式** 指令輸入後的結果, 馬上顯示在指令下方. **指派運算** 同樣會做運算式, 並且把結果 (值) 傳給變數, 但結果不會自動顯示在視窗螢幕上.

指派運算符號通常是 “`<-`”, 一個小於符號和一個短線符號組成, 如 `x<-1+2`, 讀成 `x` “得到” $(1+2)$. 若要重複一個指令, 或是叫回過去的指令, 可以用鍵盤上 “向上” ↑ 箭頭, 調出前面用過, 便可顯示回過去的指令, 再利用鍵盤上<DEL> 更改.

如果一條指令在一行結束的時候, 在語法上還不完整, R 會給出另一個不同的提示符號, 通常是 +, 該提示符號 +, 會出現在第二行, 和隨後的行中, 持續等待輸入指令, 當一指令在語法上是完整的時候, 才執行指令. 不同的完整指令再同一行時, 可用 ; (分號) 隔開, 或是另起一新輸入行. 指令可以放入大括弧內, { }, 放在一起, 構成一個複合運算式 (compound expression). **注釋** (**commands**) 幾乎可以放在任何地方, 任何一行中, 注釋從 # (井號) 開始, 到句子收尾之間的語句就是是注釋.

### 1.3.2 解說與輔助文件

R 有良好的解說文件, 最常使用的線上協助爲啓動網頁流覽器

```
> help.start()
```

若知某一特定函式名稱, 則可直接輸入下列任一種指令

```
> help(mean)
> ?mean
```

注意對於有特殊含義的字元, 可以加上雙引號或者單引號, 即 “字串”; 查詢特殊符號也要用雙引號 (") 括起來,

```
> help("if")
> ?"=="
```

另一協助指令 `help.search()` 可以讓人尋找某一特定主題, 允許你用任何方式搜尋輔助文檔, 如

```
> help.search("linear models")
```

指令 `example()` 可以執行某一特定函式輔助文件中的例子, 如

```
> example(plot)
```

指令 `data()` 可以顯示 R 目前所有的資料組, 以及載入謀特定資料組, 如

```
> data()
> data(Titanic) # 載入 Titanic 這組資料
```

### 1.3.3 顯示物件與移除物件

在 R 中產生和控制的實體稱爲"**物件**", 它們可以是向量, 陣列, 字串, 函式等 R 指令 `object()` 或 `ls()` 可以顯示當前保存在 R 環境中的物件名稱.

```
> object() # 顯示當前保存所有物件
> ls()     # 顯示當前保存所有物件
> ls(x,y)  # 顯示 x 與 y 物件是否存在
```

透過指令 `rm`, 可以刪除物件

```
> rm(x, y)
```

R 工作中產生的所有物件, 可以永久地保存在當前工作目錄下一個文字檔案中, 以便於以後的 R 使用. 在每一次 R 工作結束的時候, 你可以保存所有當前可用的物件. 這些物件會寫入當前工作目錄下, 一個叫 **.RData** 的文字檔案中, 並且所有在這次工作中用過的指令, 都會被保存在工作目錄, 一個叫 **.Rhistory** 的文字檔案中. 當 R 再次在同一工作目錄下啓動時, 這些物件將從這些檔案中, 重新引入使用, 同時, 相關的歷史指令檔案, 也會被引入使用. 使用 R 做統計資料分析, 不同的分析資料計畫, 最好用不同的工作目錄, 在分析資料過程中, 將物件命名爲 `x` 和 `y` 等, 是一件常見的事, 在任一次的分析計畫中, 這樣的命名是有其特定含義的, 但不同分析資料計畫, 在一個工作目錄下進行時, 區別資料內相同物件名稱, 是一件非常困難的事情.

### 1.3.4 中斷執行中的程式

許多時候, 由於程式寫作不當, 造成 R 永無止境的執行運算, 若要中斷執行中的程式, 可以按 **ESC** 鍵, 如

```
> for ( in in 1:100000) print (i) # 請按 ESC 鍵
```

請按 **ESC** 鍵中斷.

### 1.3.5 函式 Functions

R 語言中有許多內部 **函式** (**function**) 物件, 並且可以用在其他的運算式中. 透過函式, 擴展了 R 在程式語言的功能性, 便利性. 大多是函式都作爲 R 系統的一部分提供, 如 `sum()`, `mean()`, `var()` 等等. 這些函數都是用 R 寫成的. 一個函式內通常需輸入 **引數** (**argument**), 以下面的語句形式使用

```
> function.name(arg_1, arg_2, ...)
```

該函式運算的最終結果 (值), 就是函式返回給的物件, 如

```
> x<-c(1, 2, 3, 4, 5) # 函 c() 式返回一個向量
> x
[1] 1 2 3 4 5
> mean(x)             # 函式 mean() 計算平均值後, 返回一個平均值結果的向量
[1] 3
```

### 1.3.6 套件 Packages

有些學者針對特定分析, 寫成專用的 R 函式, 這些學者將特定的統計分析方法許多專用的函式集合成一組"**套件**" (**package**), 如 survival 套件, 專用來進行存活分析. 在 R 中, 由一些標準 (基本) 套件構成 base R, 包含 R 可以進行一些標準統計和繪圖所需的的基本函數, 在任何 R 的安裝版本中, 都會被自動安裝與載入. 另外, 許多不同作者爲 R 貢獻了好幾百個 (非基本) 套件, 若在 R 第一次使用某一特定 (非基本) 套件, 須先連接網際網路, 使用上方表單中 程式套件表單, 自動安裝和更新套件. 若在 R 中, 要使用某一特定 (非基本) 套件, 須先載入此特定套件, 才能使用此特定套件內的函式, 這樣做一是爲了提高效率, 並防止物件名稱的名字互相衝突.

可以使用 `library()`, 指令中沒有參數的指令, 查看當前工作環境中所安裝的套件

```
> library()
```

爲了載入某特定套件, 如 survival 套件, 使用以下的指令

```
> library(survival)
> library("survival")
```

若要取的相關套件的輔助文件, 可用下的指令

```
library(MASS) # 載入 MASS 套件
library(help=MASS) # 一般描述 MASS 套件
help(lda, package="MASS") # lda 是 MASS 套件中的一個函式
help(lda) # 如果已經載入 MASS 套件
```

要查看或使用套件中的內建資料框架 (data.frame), 可用以下的指令

```
> library(stats) # 載入 stats 套件
> data()         # 查看現有資料框架
> data(Puromycin)# 查看 Puromycin 資料框架
> # Alternative way
> data(package="stats")
> data(Puromycin, package="stats")
> Puromycin
   conc rate     state
1  0.02   76   treated
2  0.02   47   treated
3  0.06   97   treated
4  0.06  107   treated
5  0.11  123   treated
.....................
```

### 1.3.7 輸入指令檔案與輸出結果

指令可以用文件編輯軟體先輸入儲存成 ASCII 文字檔, 然後再 R 中叫入執行, 可在視窗左上方表單 **檔案** 點選輸入 R 程式碼, 或用下列指令

```
> source("commands.R")
```

函式 sink() 可以將以執行的指令, 程式碼與結果輸出至檔案儲存, 如下

```
> sink("record.out")
```

### 1.3.8 Rcmdr 套件文字編輯軟體

由 John Fox, McMaster university (英國) 所寫的 **Rcmdr** 套件, 提供如 SPSS 圖形使用者界面 (GUI, Graphics User Interface), 方便資料處裡與常用的統計分析, 可用以下的指令載入, R 會自動安裝與載入其他必要套件. (第一次使用 套件, **Rcmdr** 須先連接網際網路, 安裝和更新套件.)

```
library("Rcmdr")
```

使用 Rcmdr GUI 表單點選, R 會自動產生相關指令, 使用者可以在指令視窗中做修改.

### 1.3.9 文字編輯軟體

R 有簡單的內建文字編輯器, 先從程式視窗上端, 點選表單–編輯, 最下方–GUI 偏好設定, GUI 偏好設定, 可點選 MDI mode 之後, 在視窗上端表單–檔案, 可點選 建立新的指令檔案 或 開啓指令檔案, 就可開啓一 (新) 文字檔案, 容許輸入指令, 當點選並反白所要執行的指令程式碼, 再點選視窗上端表單下, 小的執行圖示–執行程式列–(類似 <=>), 即可執行指令.

用 記事本 (Notepad) 或 Wordpad 等文字編輯軟體, 可先寫入指令程式碼, 然後點選並反白所要執行的指令程式碼, 在文字編輯軟體點選複製, 然後在 R 點選小的執行圖示–貼上–, 即可執行指令.

有許多文字編輯軟體用來支援 R 的使用, 常見如 Xemacs, SciViews R GUI, WinEdt, Tinn-R 等. 對粗學者而言, Tinn-R 是一個小, 但是免費的軟體, 主要用來支援 R 而設計, 可以在下列網址 http://http://www.sciviews.org/Tinn-R/ 下載.