2
Giới thiệu ngôn ngữ R

2.1 R là gì ?

Nói một cách ngắn gọn, R là một phần mềm sử dụng cho phân tích thống kê và đồ thị. Thật ra, về bản chất, R là ngôn ngữ máy tính đa năng, có thể sử dụng cho nhiều mục tiêu khác nhau, từ tính toán đơn giản, toán học giải trí (recreational mathematics), tính toán ma trận (matrix), đến các phân tích thống kê phức tạp. Vì là một ngôn ngữ, cho nên người ta có thể sử dụng R để phát triển thành các phần mềm chuyên môn cho một vấn đề tính toán cá biệt.

Hai người sáng tạo ra R là hai nhà thống kê học tên là Ross Ihaka và Robert Gentleman. Kể từ khi R ra đời, rất nhiều nhà nghiên cứu thống kê và toán học trên thế giới ủng hộ và tham gia vào việc phát triển R. Chủ trương của những người sáng tạo ra R là theo định hướng mở rộng (Open Access). Cũng một phần vì chủ trương này mà R hoàn toàn miễn phí. Bất cứ ai ở bất cứ nơi nào trên thế giới đều có thể truy nhập và tải toàn bộ mã nguồn của R về máy tính của mình để sử dụng. Cho đến nay, chỉ qua chưa đầy 5 năm phát triển, càng ngày càng có nhiều các nhà thống kê học, toán học, nghiên cứu trong mọi lĩnh vực đã chuyển sang sử dụng R để phân tích dữ liệu khoa học. Trên toàn cầu, đã có một mạng lưới gần một triệu người sử dụng R, và con số này đang tăng theo cấp số nhân. Có thể nói trong vòng 10 năm nữa, chúng ta sẽ không cần đến các phần mềm thống kê đắt tiến như SAS, SPSS hay Stata (các phần mềm này rất đắt tiền, có thể lên đến 100.000 USD một năm) để phân tích thống kê nữa, vì tất cả các phân tích đó có thể tiến hành bằng R.

Vì thế, những ai làm nghiên cứu khoa học, nhất là ở các nước còn nghèo khó như nước ta, cần phải học cách sử dụng R cho phân tích thống kê và đồ thị. Bài viết ngắn này sẽ hướng dẫn bạn đọc cách sử dụng R. Tôi giả định rằng bạn đọc không biết gì về R, nhưng tôi kì vọng bạn đọc biết qua về cách sử dụng máy tính.

2.2 Tải R xuống và cài đặt vào máy tính

Để sử dụng R, việc đầu tiên là chúng ta phải cài đặt R trong máy tính của mình. Để làm việc này, ta phải truy nhập vào mạng và vào website có tên là "Comprehensive R Archive Network" (CRAN) sau đây:

http://cran.R-project.org.

Tài liệu cần tải về, tùy theo phiên bản, nhưng thường có tên bắt đầu bằng mẫu tự R và số phiên bản (version). Chẳng hạn như phiên bản tôi sử dụng vào cuối năm 2005 là 2.2.1, nên tên của tài liệu cần tải là:

`R-2.2.1-win32.zip`

Tài liệu này khoảng 26 MB, và địa chỉ cụ thể để tải là:

http://cran.r-project.org/bin/windows/base/R-2.2.1-win32.exe

Tại website này, chúng ta có thể tìm thấy rất nhiều tài liệu chỉ dẫn cách sử dụng R, đủ trình độ, từ sơ đẳng đến cao cấp. Nếu chưa quen với tiếng Anh, tài liệu này của tôi có thể cung cấp những thông tin cần thiết để sử dụng mà không cần phải đọc các tài liệu khác.

Khi đã tải R xuống máy tính, bước kế tiếp là cài đặt (set-up) vào máy tính. Để làm việc này, chúng ta chỉ đơn giản nhấn chuột vào tài liệu trên và làm theo hướng dẫn cách cài đặt trên màn hình. Đây là một bước rất đơn giản, chỉ cần 1 phút là việc cài đặt R có thể hoàn tất.

2.3 Package cho các phân tích đặc biệt

R cung cấp cho chúng ta một "ngôn ngữ" máy tính và một số *function* để làm các phân tích căn bản và đơn giản. Nếu muốn làm những phân tích phức tạp hơn, chúng ta cần phải tải về máy tính một số *package* khác. Package là một phần mềm nhỏ được các nhà thống kê phát triển để giải quyết một vấn đề cụ thể, và có thể chạy trong hệ thống R. Chẳng hạn như để phân tích hồi qui tuyến tính, R có function `lm` để sử dụng cho mục đích này, nhưng để làm các phân tích sâu hơn và phức tạp hơn, chúng ta cần đến các package như **lme4**. Các package này cần phải được tải về máy tính và cài đặt.

Địa chỉ để tải các package vẫn là: http://cran.r-project.org, rồi bấm vào phần "Packages" xuất hiện bên trái của mục lục trang web. Một số package cần tải về máy tính để sử dụng cho các ví dụ trong sách này là:

Tên package	**Chức năng**
`Trellis`	Dùng để vẽ đồ thị và làm cho đồ thị đẹp hơn
`lattice`	Dùng để vẽ đồ thị và làm cho đồ thị đẹp hơn
`Hmisc`	Một số phương pháp mô hình dữ liệu của F. Harrell
`Design`	Một số mô hình thiết kế nghiên cứu của F. Harrell
`Epi`	Dùng cho các phân tích dịch tễ học
`epitools`	Một package khác chuyên cho các phân tích dịch tễ học
`foreign`	Dùng để nhập dữ liệu từ các phần mềm khác như SPSS, Stata, SAS, v.v…
`Rmeta`	Dùng cho phân tích tổng hợp (meta-analysis)
`meta`	Một package khác cho phân tích tổng hợp
`survival`	Chuyên dùng cho phân tích theo mô hình Cox (Cox's proportional hazard model)

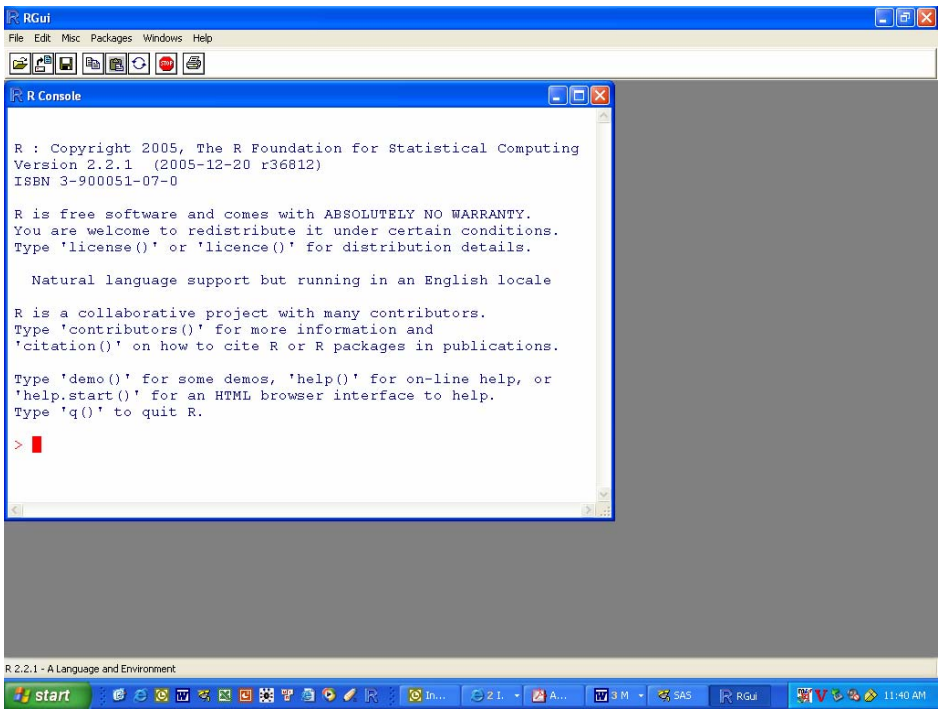
splines	Package cho survival vận hành
Zelig	Package dùng cho các phân tích thống kê trong lĩnh vực xã hội học
genetics	Package dùng cho phân tích số liệu di truyền học
BMA	Bayesian Model Average
leaps	Package dùng cho BMA

2.4 Khởi động và ngưng chạy R

Sau khi hoàn tất việc cài đặt, một *icon*

R 2.2.1.lnk

sẽ xuất hiện trên *desktop* của máy tính. Đến đây thì chúng ta đã sẵn sàng sử dụng R. Có thể nhấp chuột vào icon này và chúng ta sẽ có một *window* như sau:



R thường được sử dụng dưới dạng "*command line*", có nghĩa là chúng ta phải trực tiếp gõ lệnh vào cái *prompt* màu đỏ trên. Các lệnh phải tuân thủ nghiêm ngặt theo "văn phạm" và ngôn ngữ của R. Có thể nói toàn bộ bài viết này là nhằm hướng dẫn bạn đọc hiểu và viết theo ngôn ngữ của R. Một trong những văn phạm này là R phân biệt giữa Library và library. Nói cách khác, R phân biệt lệnh viết bằng chữ hoa hay chữ thường. Một văn phạm khác nữa là khi có hai chữ rời nhau, R thường dùng dấu chấm để

thay vào khoảng trống, chẳng hạn như `data.frame, t.test, read.table,` v.v… Điều này rất quan trọng, nếu không để ý sẽ làm mất thì giờ của người sử dụng.

Nếu lệnh gõ ra đúng "văn phạm" thì R sẽ cho chúng ta một cái prompt khác hay cho ra kết quả nào đó (tùy theo lệnh); nếu lệnh không đúng văn phạm thì R sẽ cho ra một thông báo ngắn là không đúng hay không hiểu. Ví dụ, nếu chúng ta gõ:

```
> x <- rnorm(20)
>
```

thì R sẽ hiểu và làm theo lệnh đó, rồi cho chúng ta một prompt khác: `>`. Nhưng nếu chúng ta gõ:

```
> R is great
```

R sẽ không "đồng ý" với lệnh này, vì ngôn ngữ này không có trong thư viện của R, một thông báo sau đây sẽ xuất hiện:

```
Error: syntax error
>
```

Khi muốn rời khỏi R, chúng ta có thể đơn giản nhấn nút chéo (x) bên góc trái của window, hay gõ lệnh `q()`.

2.5 "Văn phạm" ngôn ngữ R

"Văn phạm" chung của R là một lệnh (command) hay function (tôi sẽ thỉnh thoảng đề cập đến là "hàm"). Mà đã là hàm thì phải có thông số; cho nên theo sau hàm là những thông số mà chúng ta phải cung cấp. Chẳng hạn như:

```
> reg <- lm(y ~ x)
```

thì `reg` là một object, còn `lm` là một hàm, và `y ~ x` là thông số của hàm. Hay:

```
> setwd("c:/works/stats")
```

thì `setwd` là một hàm, còn `"c:/works/stats"` là thông số của hàm.

Để biết một hàm cần có những thông số nào, chúng ta dùng lệnh `args(x)`, (args viết tắt chữ `arguments`) mà trong đó x là một hàm chúng ta cần biết:

```
> 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, ...)
```

```
NULL
```

R là một ngôn ngữ "đối tượng" (object oriented language). Điều này có nghĩa là các dữ liệu trong R được chứa trong object. Định hướng này cũng có vài ảnh hưởng đến cách viết của R. Chẳng hạn như thay vì viết `x = 5` như thông thường chúng ta vẫn viết, thì R yêu cầu viết là `x == 5`.

Đối với R, `x = 5` tương đương với `x <- 5`. Cách viết sau (dùng kí hiệu <-) được khuyến khích hơn là cách viết trước (=). Chẳng hạn như:

```
> x <- rnorm(10)
```

có nghĩa là mô phỏng 10 số liệu và chứa trong object `x`. Chúng ta cũng có thể viết `x = rnorm(10)`.

Một số kí hiệu hay dùng trong R là:

`x == 5`	x bằng 5
`x != 5`	x không bằng 5
`y < x`	y nhỏ hơn x
`x > y`	x lớn hơn y
`z <= 7`	z nhỏ hơn hoặc bằng 7
`p >= 1`	p lớn hơn hoặc bằng 1
`is.na(x)`	Có phải x là biến số missing
`A & B`	A và B (AND)
`A | B`	A hoặc B (OR)
`!`	Không là (NOT)

Với R, tất cả các câu chữ hay lệnh sau kí hiệu # đều không có hiệu ứng, vì # là kí hiệu dành cho người sử dụng thêm vào các ghi chú, ví dụ:

```
> # lệnh sau đây sẽ mô phỏng 10 giá trị normal
> x <- rnorm(10)
```

2.6 Cách đặt tên trong R

Đặt tên một đối tượng (object) hay một biến số (variable) trong R khá linh hoạt, vì R không có nhiều giới hạn như các phần mềm khác. Tên một object phải được viết liền nhau (tức không được cách rồi bằng một khoảng trống). Chẳng hạn như R chấp nhận `myobject` nhưng không chấp nhận `my object`.

```
> myobject <- rnorm(10)
> my object <- rnorm(10)
Error: syntax error in "my object"
```

Nhưng đôi khi tên `myobject` khó đọc, cho nên chúng ta nên tác rời bằng "." Như `my.object`.

```
> my.object <- rnorm(10)
```

Một điều quan trọng cần lưu ý là R phân biết mẫu tự viết hoa và viết thường. Cho nên `My.object` khác với `my.object`. Ví dụ:

```
> My.object.u <- 15
> my.object.L <- 5
> My.object.u + my.object.L
[1] 20
```

Một vài điều cần lưu ý khi đặt tên trong R là:

- Không nên đặt tên một biến số hay variable bằng kí hiệu "_" (underscore) như `my_object` hay `my-object`.

- Không nên đặt tên một object giống như một biến số trong một dữ liệu. Ví dụ, nếu chúng ta có một `data.frame` (dữ liệu hay dataset) với biến số `age` trong đó, thì không nên có một object trùng tên `age`, tức là không nên viết: `age <- age`. Tuy nhiên, nếu data.frame tên là `data` thì chúng ta có thể đề cập đến biến số `age` với một kí tự `$` như sau: `data$age`. (Tức là biến số `age` trong data.frame `data`), và trong trường hợp đó, `age <- data$age` có thể chấp nhận được.

2.7 Hỗ trợ trong R

Ngoài lệnh `args()` R còn cung cấp lệnh `help()` để người sử dụng có thể hiểu "văn phạm" của từng hàm. Chẳng hạn như muốn biết hàm `lm` có những thông số (arguments) nào, chúng ta chỉ đơn giản lệnh:

```
> help(lm)
```

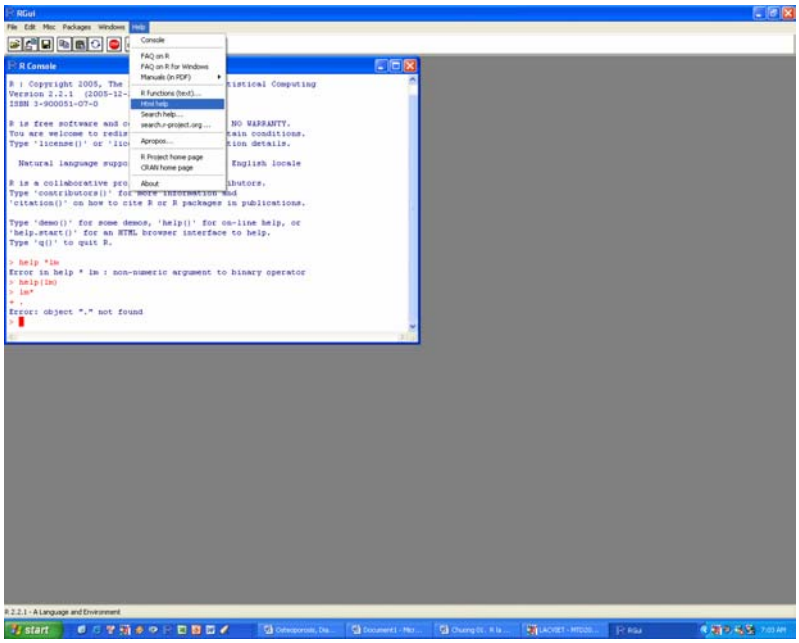
hay

```
> ?lm
```

Một cửa sổ sẽ hiện ra bên phải của màn hình chỉ rõ cách sử dụng ra sao và thậm chí có cả ví dụ. Bạn đọc có thể đơn giản copy và dán ví dụ vào R để xem cách vận hành.

Trước khi sử dụng R, ngoài sách này nếu cần bạn đọc có thể đọc qua phần chỉ dẫn có sẵn trong R bằng cách chọn mục `help` và sau đó chọn `Html help` như hình dưới

đây để biết thêm chi tiết. Bạn đọc cũng có thể copy và dán các lệnh trong mục này vào R để xem cho biết cách vận hành của R.



Thay vì chọn mục trên, bạn đọc cũng có thể đơn giản lệnh:

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

và một cửa sổ sẽ xuất hiện chỉ dẫn toàn bộ hệ thống R.

Hàm `apropos` cũng rất có ích vì nó cung cấp cho chúng ta tất cả các hàm trong R bắt đầu bằng kí tự mà chúng ta muốn tìm. Chẳng hạn như chúng ta muốn biết hàm nào trong R có kí tự "`lm`" thì chỉ đơn giản lệnh:

```
> apropos(lm)
```

Và R sẽ báo cáo các hàm với kí tự `lm` như sau có sẵn trong R:

```
 [1] ".__C__anova.glm"      ".__C__anova.glm.null" ".__C__glm"
 [4] ".__C__glm.null"       ".__C__lm"             ".__C__mlm"
 [7] "anova.glm"            "anova.glmlist"        "anova.lm"
[10] "anova.lmlist"         "anova.mlm"            "anovalist.lm"
[13] "contr.helmert"        "glm"                  "glm.control"
[16] "glm.fit"              "glm.fit.null"         "hatvalues.lm"
[19] "KalmanForecast"       "KalmanLike"           "KalmanRun"
[22] "KalmanSmooth"         "lm"                   "lm.fit"
[25] "lm.fit.null"          "lm.influence"         "lm.wfit"
[28]   "lm.wfit.null"                           "model.frame.glm"
"model.frame.lm"
```

```
[31] "model.matrix.lm"      "nlm"                  "nlminb"
[34] "plot.lm"              "plot.mlm"             "predict.glm"
[37] "predict.lm"           "predict.mlm"          "print.glm"
[40] "print.lm"             "residuals.glm"        "residuals.lm"
[43] "rstandard.glm"        "rstandard.lm"         "rstudent.glm"
[46] "rstudent.lm"          "summary.glm"          "summary.lm"
[49] "summary.mlm"          "kappa.lm"
```

2.8 Môi trường vận hành

Dữ liệu phải được chứa trong một khu vực (directory) của máy tính. Trước khi sử dụng R, có lẽ cách hay nhất là tạo ra một directory để chứa dữ liệu, chẳng hạn như `c:\works\stats`. Để R biết dữ liệu nằm ở đâu, chúng ta sử dụng lệnh `setwd` (set working directory) như sau:

```
> setwd("c:/works/stats")
```

Lệnh trên báo cho R biết là dữ liệu sẽ chứa trong directory có tên là c:\works\stats. Chú ý rằng, R dùng forward slash "/" chứ không phải backward slash "\" như trong hệ thống Windows.

Để biết hiện nay, R đang "làm việc" ở directory nào, chúng ta chỉ cần lệnh:

```
> getwd()
[1] "C:/Program Files/R/R-2.2.1"
```

Cái prompt mặc định của R là ">". Nhưng nếu chúng ta muốn có một prompt khác theo cá tính cá nhân, chúng ta có thể thay thế dễ dàng:

```
> options(prompt="R> ")
R>
```

Hay:

```
> options(prompt="Tuan> ")
Tuan>
```

Màn ảnh R mặc định là 80 characters, nhưng nếu chúng ta muốn màn ảnh rộng hơn, thì chỉ cần ra lệnh:

```
> options(width=100)
```

Hay muốn R trình bày các số liệu ở dạng 3 số thập phân:

```
> options(scipen=3)
```

Các lựa chọn và thay đổi này có thể dùng lệnh `options()`. Để biết các thông số hiện tại của R là gì, chúng ta chỉ cần lệnh:

```
> options()
```

Tìm hiểu ngày tháng:

```
> Sys.Date()
[1] "2006-03-31"
```

Nếu bạn đọc cần thêm thông tin, một số tài liệu trên mạng (viết bằng tiếng Anh) cũng rất có ích. Các tài liệu này có thể tải xuống máy miễn phí:

R for beginners (của Emmanuel Paradis):
http://cran.r-project.org/doc/contrib/rdebuts_en.pdf

Using R for data analysis and graphics (của John Maindonald):
http://cran.r-project.org/doc/contrib/usingR.pdf