Phân tích số liệu và biểu đồ bằng

Nguyễn Văn Tuấn
Garvan Institute of Medical Research
Sydney, Australia

Mục lục

Giới thiệu R

Phân tích số liệu và biểu đồ thường được tiến hành bằng các phần mềm thông dụng như SAS, SPSS, *Stata*, *Statistica*, và *S-Plus*. Đây là những phần mềm được các công ti phần mềm phát triển và giới thiệu trên thị trường khoảng ba thập niên qua, và đã được các trường đại học, các trung tâm nghiên cứu và công ti kĩ nghệ trên toàn thế giới sử dụng cho giảng dạy và nghiên cứu. Nhưng vì chi phí để sử dụng các phần mềm này tương đối đắt tiền (có khi lên đến hàng trăm ngàn đô-la mỗi năm), một số trường đại học ở các nước đang phát triển (và ngay cả ở một số nước đã phát triển) không có khả năng tài chính để sử dụng chúng một cách lâu dài. Do đó, các nhà nghiên cứu thống kê trên thế giới đã hợp tác với nhau để phát triển một phần mềm mới, với chủ trương mã nguồn mở, sao cho tất cả các thành viên trong ngành thống kê học và toán học trên thế giới có thể sử dụng một cách thống nhất và **hoàn toàn miễn phí**.

Năm 1996, trong một bài báo quan trọng về tính toán thống kê, hai nhà thống kê học Ross Ihaka và Robert Gentleman [lúc đó] thuộc Trường đại học Auckland, New Zealand phát hoạ một ngôn ngữ mới cho phân tích thống kê mà họ đặt tên là R [1]. Sáng kiến này được rất nhiều nhà thống kê học trên thế giới tán thành và tham gia vào việc phát triển R.

Cho đến nay, qua chưa đầy 10 năm phát triển, càng ngày càng có nhiều 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 hơn một triệu người sử dụng R, và con số này đang tăng rất nhanh. Có thể nói trong vòng 10 năm nữa, vai trò của các phần mềm thống kê thương mại sẽ không còn lớn như trong thời gian qua nữa.

Vậy R là gì? Nói một cách ngắn gọn, R là một phần mềm sử dụng cho phân tích thống kê và vẽ biểu đồ. 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.

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.

1. 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

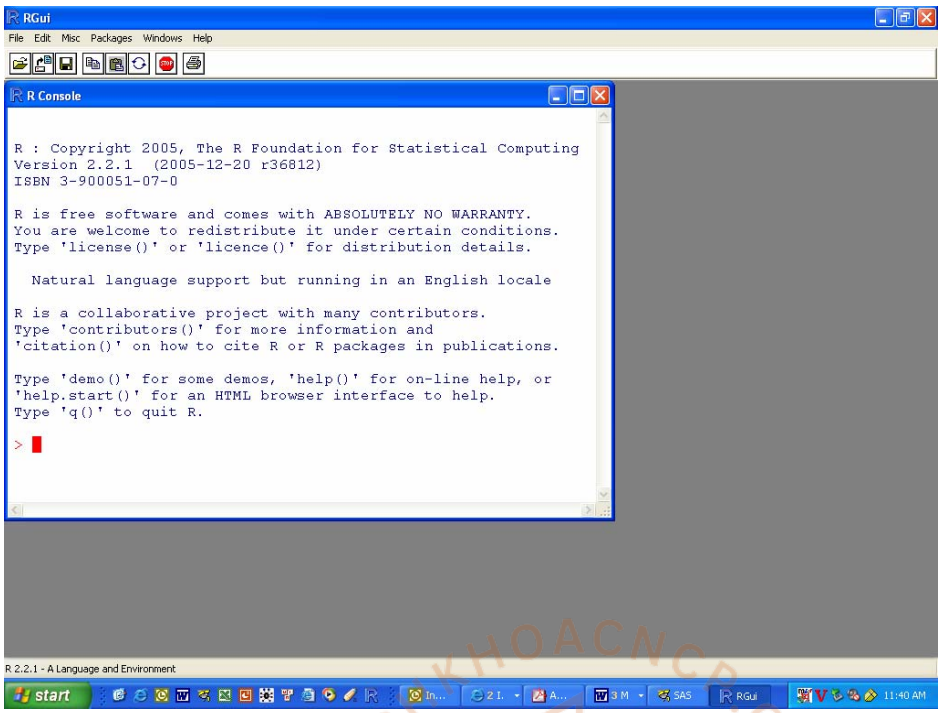
Tại website này, chúng ta có thể tìm thấy rất nhiều tài liệu chỉ dẫn cách sử dụng R, đủ trình độ, từ sơ đẳng đến cao cấp. Nếu chưa quen với tiếng Anh, tài liệu này của tôi có thể cung cấp những thông tin cần thiết để sử dụng mà không cần phải đọc các tài liệu khác.

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.

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:



2. Tải R package và cài đặt vào máy tính

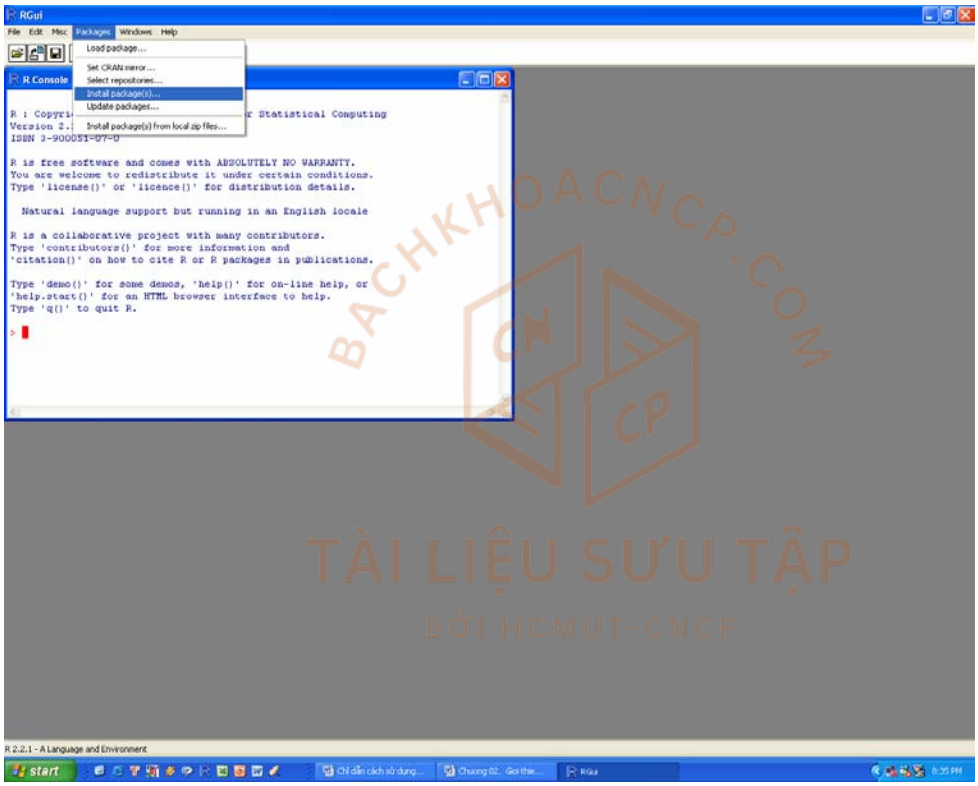
R cung cấp cho chúng ta một "ngôn ngữ" máy tính và một số *function* để làm các phân tích căn bản và đơn giản. Nếu muốn làm những phân tích phức tạp hơn, chúng ta cần phải tải về máy tính một số *package* khác. Package là một phần mềm nhỏ được các nhà thống kê phát triển để giải quyết một vấn đề cụ thể, 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 và cài đặt vào máy tính.

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. Theo tôi, một số package cần tải về máy tính để sử dụng cho các phân tích dịch tễ học 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)
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

Các package này có thể cài đặt trực tuyến bằng cách chọn **Install packages** trong phần **packages** của R như hình dưới đây. Ngoài ra, nếu package đã được tải xuống máy tính cá nhân, việc cài đặt có thể nhanh hơn bằng cách chọn **Install package(s) from local zip file** cũng trong phần **packages** (xem hình dưới đây).



3. "Văn phạm" R

R là một ngôn ngữ tương tác (interactive language), có nghĩa là khi chúng ta ra lệnh, và nếu lệnh theo đúng "văn phạm", R sẽ "đáp" lại bằng một kết quả. Và, sự tương tác tiếp tục cho đến khi chúng ta đạt được yêu cầu. "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. Cú pháp chung của R là như sau:

```
đối tượng <- hàm(thông số 1, thông số 2, …, thông số n)
```

Chẳng hạn như:

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

thì `reg` là một đối tượng (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 trống không (missing value)
`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)
```

3.1 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 rời 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 biệt 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.

3.2 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.

4. Cách nhập dữ liệu vào R

Muốn làm phân tích dữ liệu bằng R, chúng ta phải có sẵn dữ liệu ở dạng mà R có thể hiểu được để xử lí. Dữ liệu mà R hiểu được phải là dữ liệu trong một `data.frame`. Có nhiều cách để nhập số liệu vào một `data.frame` trong R, từ nhập trực tiếp đến nhập từ các nguồn khác nhau. Sau đây là những cách thông dụng nhất:

4.1 Nhập số liệu trực tiếp: `c()`

Ví dụ 1: chúng ta có số liệu về độ tuổi và insulin cho 10 bệnh nhân như sau, và muốn nhập vào R.

```
50      16.5
62      10.8
60      32.3
40      19.3
48      14.2
47      11.3
57      15.5
70      15.8
48      16.2
67      11.2
```

Chúng ta có thể sử dụng function có tên `c` như sau:

```
> age <- c(50,62, 60,40,48,47,57,70,48,67)
> insulin <- c(16.5,10.8,32.3,19.3,14.2,11.3,15.5,15.8,16.2,11.2)
```

Lệnh thứ nhất cho R biết rằng chúng ta muốn tạo ra một cột dữ liệu (từ nay tôi sẽ gọi là *biến số*, tức *variable*) có tên là `age`, và lệnh thứ hai là tạo ra một cột khác có tên là `insulin`. Tất nhiên, chúng ta có thể lấy một tên khác mà mình thích.

Chúng ta dùng function `c` (viết tắt của chữ `concatenation` – có nghĩa là "móc nối vào nhau") để nhập dữ liệu. Chú ý rằng mỗi số liệu cho mỗi bệnh nhân được cách nhau bằng một dấu phẩy.

Kí hiệu `insulin <-` (cũng có thể viết là `insulin =`) có nghĩa là các số liệu theo sau sẽ có nằm trong biến số `insulin`. Chúng ta sẽ gặp kí hiệu này rất nhiều lần trong khi sử dụng R.

R là một ngôn ngữ cấu trúc theo dạng đối tượng (thuật ngữ chuyên môn là "object-oriented language"), vì mỗi cột số liệu hay mỗi một `data.frame` là một đối tượng (object) đối với R. Vì thế, `age` và `insulin` là hai đối tượng riêng lẻ. Bây giờ chúng ta cần phải nhập hai đối tượng này thành một `data.frame` để R có thể xử lí sau này. Để làm việc này chúng ta cần đến function `data.frame`:

```
> tuan <- data.frame(age, insulin)
```

Trong lệnh này, chúng ta muốn cho R biết rằng nhập hai cột (hay hai đối tượng) `age` và `insulin` vào một đối tượng có tên là `tuan`.

Đến đây thì chúng ta đã có một đối tượng hoàn chỉnh để tiến hành phân tích thống kê. Để kiểm tra xem trong `tuan` có gì, chúng ta chỉ cần đơn giản gõ:

```
> tuan
```


Và R sẽ báo cáo:

```
   age insulin
1   50    16.5
2   62    10.8
3   60    32.3
4   40    19.3
5   48    14.2
6   47    11.3
7   57    15.5
8   70    15.8
9   48    16.2
10  67    11.2
```

Nếu chúng ta muốn lưu lại các số liệu này trong một file theo dạng R, chúng ta cần dùng lệnh `save`. Giả dụ như chúng ta muốn lưu số liệu trong directory có tên là "c:\works\insulin", chúng ta cần gõ như sau:

```
> setwd("c:/works/insulin")
> save(tuan, file="tuan.rda")
```

Lệnh đầu tiên (`setwd` – chữ *wd* có nghĩa là *working directory*) cho R biết rằng chúng ta muốn lưu các số liệu trong directory có tên là "c:\works\insulin". Lưu ý rằng thông thường Windows dùng dấu backward slash "/", nhưng trong R chúng ta dùng dấu forward slash "/".

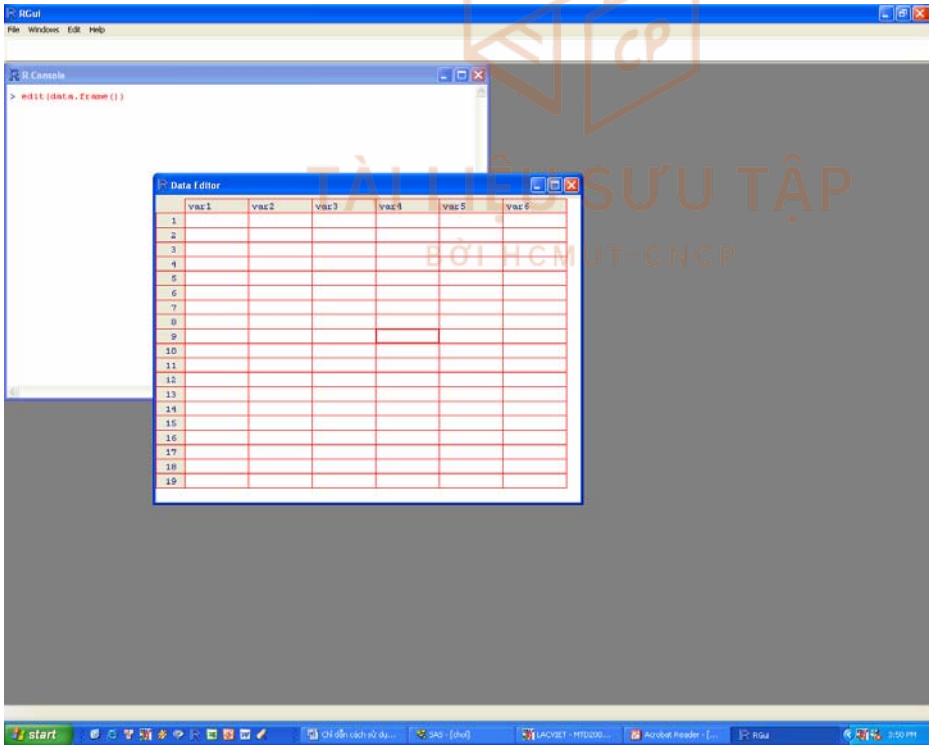
Lệnh thứ hai (`save`) cho R biết rằng các số liệu trong đối tượng `tuan` sẽ lưu trong file có tên là "`tuan.rda`"). Sau khi gõ xong hai lệnh trên, một file có tên `tuan.rda` sẽ có mặt trong directory đó.

4.2 Nhập số liệu trực tiếp: `edit(data.frame())`

Ví dụ 1 (tiếp tục): chúng ta có thể nhập số liệu về độ tuổi và insulin cho 10 bệnh nhân bằng một function rất có ích, đó là: `edit(data.frame())`. Với function này, R sẽ cung cấp cho chúng ta một window mới với một dãy cột và dòng giống như Excel, và chúng ta có thể nhập số liệu trong bảng đó. Ví dụ:

```
> ins <- edit(data.frame())
```

Chúng ta sẽ có một cửa sổ như sau:



Ở đây, R không biết chúng ta có biến số nào, cho nên R liệt kê các biến số `var1`, `var2`, v.v... Nhấp chuột vào cột `var1` và thay đổi bằng cách gõ vào đó `age`. Nhấp chuột vào cột `var2` và thay đổi bằng cách gõ vào đó `insulin`. Sau đó gõ số liệu cho

từng cột. Sau khi xong, bấm nút chéo X ở góc phải của spreadsheet, chúng ta sẽ có một data.frame tên `ins` với hai biến số `age` và `insulin`.

4.3 Nhập số liệu từ một *text file:* `read.table`

Ví dụ 2: Chúng ta thu thập số liệu về độ tuổi và cholesterol từ một nghiên cứu ở 50 bệnh nhân mắc bệnh cao huyết áp. Các số liệu này được lưu trong một text file có tên là `chol.txt` tại directory `c:\works\insulin`. Số liệu này như sau: cột 1 là mã số của bệnh nhân, cột 2 là giới tính, cột 3 là body mass index (bmi), cột 4 là HDL cholesterol (viết tắt là hdl), kế đến là LDL cholesterol, total cholesterol (tc) và triglycerides (tg).

id	sex	age	bmi	hdl	ldl	tc	tg
1	Nam	57	17	5.000	2.0	4.0	1.1
2	Nu	64	18	4.380	3.0	3.5	2.1
3	Nu	60	18	3.360	3.0	4.7	0.8
4	Nam	65	18	5.920	4.0	7.7	1.1
5	Nam	47	18	6.250	2.1	5.0	2.1
6	Nu	65	18	4.150	3.0	4.2	1.5
7	Nam	76	19	0.737	3.0	5.9	2.6
8	Nam	61	19	7.170	3.0	6.1	1.5
9	Nam	59	19	6.942	3.0	5.9	5.4
10	Nu	57	19	5.000	2.0	4.0	1.9
...							
46	Nu	52	24	3.360	2.0	3.7	1.2
47	Nam	64	24	7.170	1.0	6.1	1.9
48	Nam	45	24	7.880	4.0	6.7	3.3
49	Nu	64	25	7.360	4.6	8.1	4.0
50	Nu	62	25	7.750	4.0	6.2	2.5

Chúng ta muốn nhập các dữ liệu này vào R để tiện việc phân tích sau này. Chúng ta sẽ sử dụng lệnh `read.table` như sau:

```
> setwd("c:/works/insulin")
> chol <- read.table("chol.txt", header=TRUE)
```

Lệnh thứ nhất chúng ta muốn đảm bảo R truy nhập đúng directory mà số liệu đang được lưu giữ. Lệnh thứ hai yêu cầu R nhập số liệu từ file có tên là "`chol.txt`" (trong directory c:\works\insulin) và cho vào đối tượng `chol`. Trong lệnh này, `header=TRUE` có nghĩa là yêu cầu R đọc dòng đầu tiên trong file đó như là tên của từng cột dữ kiện.

Chúng ta có thể kiểm tra xem R đã đọc hết các dữ liệu hay chưa bằng cách ra lệnh:

```
> chol
```

Hay

```
> names(chol)
```

R sẽ cho biết có các cột như sau trong dữ liệu (`names` là lệnh hỏi trong dữ liệu có những cột nào và tên gì):

```
[1] "id"  "sex" "age" "bmi" "hdl" "ldl" "tc"  "tg"
```

Bây giờ chúng ta có thể lưu dữ liệu dưới dạng R để xử lí sau này bằng cách ra lệnh:

```
> save(chol, file="chol.rda")
```

4.4 Nhập số liệu từ *Excel:* `read.csv`

Để nhập số liệu từ phần mềm Excel, chúng ta cần tiến hành 2 bước:

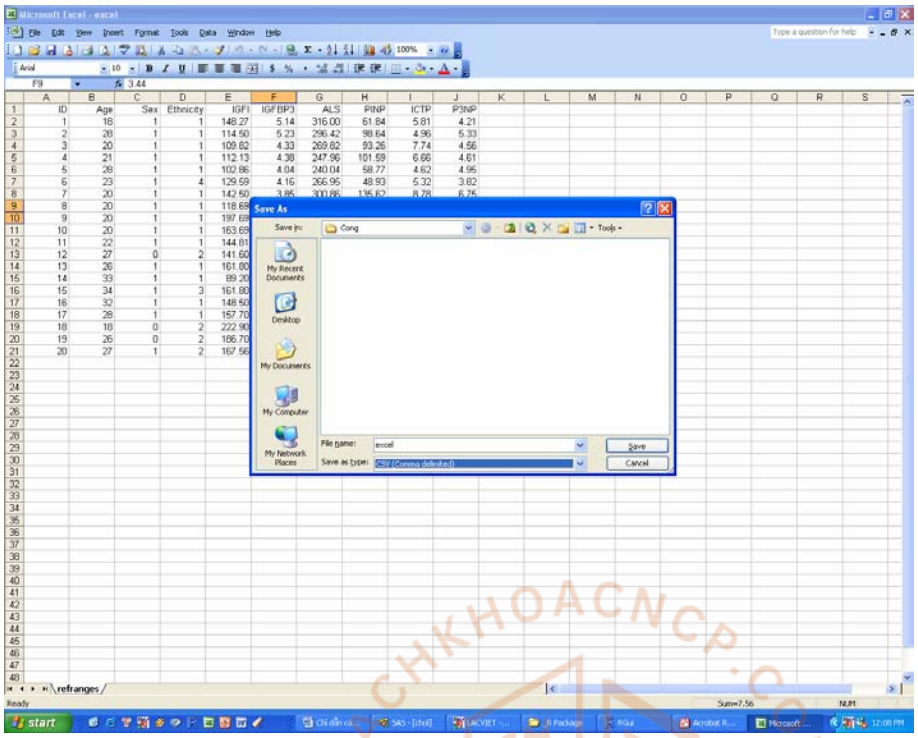
- Bước 1: Dùng lệnh "Save as" trong Excel và lưu số liệu dưới dạng "csv";
- Bước 2: Dùng R (lệnh `read.csv`) để nhập dữ liệu dạng csv.

Ví dụ 3: Một dữ liệu gồm các cột sau đây đang được lưu trong Excel, và chúng ta muốn chuyển vào R để phân tích. Dữ liệu này có tên là `excel.xls`.

ID	Age	Sex	Ethnicity	IGFI	IGFBP3	ALS	PINP	ICTP	P3NP
1	18	1	1	148.27	5.14	316.00	61.84	5.81	4.21
2	28	1	1	114.50	5.23	296.42	98.64	4.96	5.33
3	20	1	1	109.82	4.33	269.82	93.26	7.74	4.56
4	21	1	1	112.13	4.38	247.96	101.59	6.66	4.61
5	28	1	1	102.86	4.04	240.04	58.77	4.62	4.95
6	23	1	4	129.59	4.16	266.95	48.93	5.32	3.82
7	20	1	1	142.50	3.85	300.86	135.62	8.78	6.75
8	20	1	1	118.69	3.44	277.46	79.51	7.19	5.11
9	20	1	1	197.69	4.12	335.23	57.25	6.21	4.44
10	20	1	1	163.69	3.96	306.83	74.03	4.95	4.84
11	22	1	1	144.81	3.63	295.46	68.26	4.54	3.70
12	27	0	2	141.60	3.48	231.20	56.78	4.47	4.07
13	26	1	1	161.80	4.10	244.80	75.75	6.27	5.26
14	33	1	1	89.20	2.82	177.20	48.57	3.58	3.68
15	34	1	3	161.80	3.80	243.60	50.68	3.52	3.35
16	32	1	1	148.50	3.72	234.80	83.98	4.85	3.80
17	28	1	1	157.70	3.98	224.80	60.42	4.89	4.09
18	18	0	2	222.90	3.98	281.40	74.17	6.43	5.84
19	26	0	2	186.70	4.64	340.80	38.05	5.12	5.77
20	27	1	2	167.56	3.56	321.12	30.18	4.78	6.12

Việc đầu tiên là chúng ta cần làm, như nói trên, là vào Excel để lưu dưới dạng csv:

- Vào Excel, chọn `File` → `Save as`
- Chọn `Save as type` "`CSV (Comma delimited)`"



Sau khi xong, chúng ta sẽ có một file với tên "excel.csv" trong directory "c:\works\insulin".

Việc thứ hai là vào R và ra những lệnh sau đây:

```
> setwd("c:/works/insulin")
> gh <- read.csv ("excel.txt", header=TRUE)
```

Lệnh thứ hai `read.csv` yêu cầu R đọc số liệu từ "excel.csv", dùng dòng thứ nhất là tên cột, và lưu các số liệu này trong một object có tên là `gh`.

Bây giờ chúng ta có thể lưu `gh` dưới dạng R để xử lí sau này bằng lệnh sau đây:

```
> save(gh, file="gh.rda")
```

4.5 Nhập số liệu từ một *SPSS:* `read.spss`

Phần mềm thống kê SPSS lưu dữ liệu dưới dạng "sav". Chẳng hạn như nếu chúng ta đã có một dữ liệu có tên là testo.sav trong directory c:\works\insulin, và muốn chuyển dữ liệu này sang dạng R có thể hiểu được, chúng ta cần sử dụng lệnh `read.spss` trong package có tên là `foreign`. Các lệnh sau đây sẽ hoàn tất dễ dàng việc này:

Việc đầu tiên chúng ta cho truy nhập `foreign` bằng lệnh `library:`

```
> library(foreign)
```

Việc thứ hai là lệnh `read.spss`:

```
> setwd("c:/works/insulin")
> testo <- read.spss("testo.sav", to.data.frame=TRUE)
```

Lệnh thứ hai `read.spss` yêu cầu R đọc số liệu từ "testo.sav", và cho vào một `data.frame` có tên là `testo`.

Bây giờ chúng ta có thể lưu `testo` dưới dạng R để xử lí sau này bằng lệnh sau đây:

```
> save(testo, file="testo.rda")
```

4.6 Thông tin về dữ liệu

Giả dụ như chúng ta đã nhập số liệu vào một data.frame có tên là `chol` như trong ví dụ 1. Để tìm hiểu xem trong dữ liệu này có gì, chúng ta có thể nhập vào R như sau:

- Dẫn cho R biết chúng ta muốn xử lí `chol` bằng cách dùng lệnh `attach(arg)` với `arg` là tên của dữ liệu..

```
> attach(chol)
```

- Chúng ta có thể kiểm tra xem chol có phải là một data.frame không bằng lệnh `is.data.frame(arg)` với `arg` là tên của dữ liệu. Ví dụ:

```
> is.data.frame(chol)
[1] TRUE
```

R cho biết `chol` quả là một data.frame.

- Có bao nhiêu cột (hay *variable* = *biến số*) và dòng số liệu (observations) trong dữ liệu này? Chúng ta dùng lệnh `dim(arg)` với `arg` là tên của dữ liệu. (`dim` viết tắt chữ dimension). Ví dụ (kết quả của R trình bày ngay sau khi chúng ta gõ lệnh):

```
> dim(chol)
[1] 50  8
```

- Như vậy, chúng ta có 50 dòng và 8 cột (hay biến số). Vậy những biến số này tên gì? Chúng ta dùng lệnh `names(arg)` với `arg` là tên của dữ liệu. Ví dụ:

```
> names(chol)
[1] "id"  "sex" "age" "bmi" "hdl" "ldl" "tc"  "tg"
```

- Trong biến số `sex`, chúng ta có bao nhiêu nam và nữ? Để trả lời câu hỏi này, chúng ta có thể dùng lệnh `table(arg)` với `arg` là tên của biến số. Ví dụ:

```
> table(sex)
sex
nam Nam  Nu
  1  21  28
```

Kết quả cho thấy dữ liệu này có 21 nam và 28 nữ.

4.7 Tạo dãy số bằng hàm `seq`, `rep` và `gl`

R còn có công dụng tạo ra những dãy số rất tiện cho việc mô phỏng và thiết kế thí nghiệm. Những hàm thông thường cho dãy số là `seq` (sequence), `rep` (repetition) và `gl` (generating levels):

Áp dụng `seq`

- Tạo ra một vector số từ 1 đến 12:

```
> x <- (1:12)
> x
 [1]  1  2  3  4  5  6  7  8  9 10 11 12

> seq(12)
[1]  1  2  3  4  5  6  7  8  9 10 11 12
```

- Tạo ra một vector số từ 12 đến 5:

```
> x <- (12:5)
> x
[1] 12 11 10  9  8  7  6  5

> seq(12,7)
[1] 12 11 10  9  8  7
```

Công thức chung của hàm `seq` là `seq(from, to, by= )` hay `seq(from, to, length.out= ).` Cách sử dụng sẽ được minh hoạ bằng vài ví dụ sau đây:

- Tạo ra một vector số từ 4 đến 6 với khoảng cách bằng 0.25:

```
> seq(4, 6, 0.25)
[1] 4.00 4.25 4.50 4.75 5.00 5.25 5.50 5.75 6.00
```

- Tạo ra một vector 10 số, với số nhỏ nhất là 2 và số lớn nhất là 15

```
> seq(length=10, from=2, to=15)
 [1]  2.000000  3.444444  4.888889  6.333333  7.777778  9.222222
10.666667 12.111111 13.555556 15.000000
```

Áp dụng `rep`

Công thức của hàm `rep` là `rep(x, times, ...)`, trong đó, *x* là một biến số và `times` là số lần lặp lại. Ví dụ:

- Tạo ra số 10, 3 lần:

```
> rep(10, 3)
[1] 10 10 10
```

- Tạo ra số 1 đến 4, 3 lần:

```
> rep(c(1:4), 3)
 [1] 1 2 3 4 1 2 3 4 1 2 3 4
```

- Tạo ra số 1.2, 2.7, 4.8, 5 lần:

```
> rep(c(1.2, 2.7, 4.8), 5)
 [1] 1.2 2.7 4.8 1.2 2.7 4.8 1.2 2.7 4.8 1.2 2.7 4.8 1.2 2.7 4.8
```

- Tạo ra số 1.2, 2.7, 4.8, 5 lần:

```
> rep(c(1.2, 2.7, 4.8), 5)
 [1] 1.2 2.7 4.8 1.2 2.7 4.8 1.2 2.7 4.8 1.2 2.7 4.8 1.2 2.7 4.8
```

Áp dụng `gl`

`gl` được áp dụng để tạo ra một biến thứ bậc (categorical variable), tức biến không để tính toán, mà là đếm. Công thức chung của hàm `gl` là `gl(n, k, length = n*k, labels = 1:n, ordered = FALSE)` và cách sử dụng sẽ được minh hoạ bằng vài ví dụ sau đây:

- Tạo ra biến gồm bậc 1 và 2; mỗi bậc được lặp lại 8 lần:

```
> gl(2, 8)
 [1] 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
Levels: 1 2
```

Hay một biến gồm bậc 1, 2 và 3; mỗi bậc được lặp lại 5 lần:

```
> gl(3, 5)
[1] 1 1 1 1 1 2 2 2 2 2 3 3 3 3 3
Levels: 1 2 3
```

- Tạo ra biến gồm bậc 1 và 2; mỗi bậc được lặp lại 10 lần (do đó length=20):

```
> gl(2, 10, length=20)
[1] 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
Levels: 1 2
```

Hay:

```
> gl(2, 2, length=20)
[1] 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2 1 1 2 2
Levels: 1 2
```

- Cho thêm kí hiệu:

```
> gl(2, 5, label=c("C", "T"))
 [1] C C C C C T T T T T
Levels: C T
```

- Tạo một biến gồm 4 bậc 1, 2, 3, 4. Mỗi bậc lặp lại 2 lần.

```
> rep(1:4, c(2,2,2,2))
[1] 1 1 2 2 3 3 4 4
```

Cũng tương đương với:

```
> rep(1:4, each = 2)
[1] 1 1 2 2 3 3 4 4
```

- Với ngày giờ tháng:

```
> x <- .leap.seconds[1:3]
> rep(x, 2)
[1] "1972-06-30 17:00:00 Pacific Standard Time" "1972-12-31 16:00:00
Pacific Standard Time"
[3] "1973-12-31 16:00:00 Pacific Standard Time" "1972-06-30 17:00:00
Pacific Standard Time"
[5] "1972-12-31 16:00:00 Pacific Standard Time" "1973-12-31 16:00:00
Pacific Standard Time"

> rep(as.POSIXlt(x), rep(2, 3))
[1] "1972-06-30 17:00:00 Pacific Standard Time" "1972-06-30 17:00:00
Pacific Standard Time"
[3] "1972-12-31 16:00:00 Pacific Standard Time" "1972-12-31 16:00:00
Pacific Standard Time"
[5] "1973-12-31 16:00:00 Pacific Standard Time" "1973-12-31 16:00:00
Pacific Standard Time"
```

5. Biên tập số liệu

5.1 Tách rời dữ liệu: `subset`

Chúng ta sẽ quay lại với dữ liệu `chol` trong ví dụ 1. Để tiện việc theo dõi và hiểu "câu chuyện", tôi xin nhắc lại rằng chứng ta đã nhập số liệu vào trong một dữ liệu R có tên là `chol` từ một text file có tên là `chol.txt`:

```
> setwd("c:/works/insulin")
> chol <- read.table("chol.txt", header=TRUE)
> attach(chol)
```

Nếu chúng ta, vì một lí do nào đó, chỉ muốn phân tích riêng cho nam giới, chúng ta có thể tách `chol` ra thành hai data.frame, tạm gọi là `nam` và `nu`. Để làm chuyện này, chúng ta dùng lệnh `subset(data, cond)`, trong đó `data` là data.frame mà chúng ta muốn tách rời, và `cond` là điều kiện. Ví dụ:

```
> nam <- subset(chol, sex=="Nam")
> nu <- subset(chol, sex=="Nu")
```

Sau khi ra hai lệnh này, chúng ta đã có 2 dữ liệu (hai data.frame) mới tên là `nam` và `nu`. Chú ý điều kiện `sex == "Nam"` và `sex == "Nu"` chúng ta dùng == thay vì = để chỉ điều kiện chính xác.

Tất nhiên, chúng ta cũng có thể tách dữ liệu thành nhiều data.frame khác nhau với những điều kiện dựa vào các biến số khác. Chẳng hạn như lệnh sau đây tạo ra một data.frame mới tên là `old` với những bệnh nhân trên 60 tuổi:

```
> old <- subset(chol, age>=60)
> dim(old)
 [1] 25  8
```

Hay một data.frame mới với những bệnh nhân trên 60 tuổi và nam giới:

```
> n60 <- subset(chol, age>=60 & sex=="Nam")
> dim(n60)
[1] 9  8
```

5.2 Chiết số liệu từ một data .frame

Trong `chol` có 8 biến số. Chúng ta có thể chiết dữ liệu `chol` và chỉ giữ lại những biến số cần thiết như mã số (`id`), độ tuổi (`age`) và total cholestrol (`tc`). Để ý từ lệnh `names(chol)` rằng biến số `id` là cột số 1, `age` là cột số 3, và biến số `tc` là cột số 7. Chúng ta có thể dùng lệnh sau đây:

```
> data2 <- chol[, c(1,3,7)]
```

Ở đây, chúng ta lệnh cho R biết rằng chúng ta muốn chọn cột số 1, 3 và 7, và đưa tất cả số liệu của hai cột này vào data.frame mới có tên là `data2`. Chú ý chúng ta sử dụng ngoặc kép vuông [] chứ không phải ngoặc kép vòng (), vì `chol` không phải làm một function. Dấu phẩy phía trước `c`, có nghĩa là chúng ta chọn tất cả các dòng số liệu trong data.frame `chol`.

Nhưng nếu chúng ta chỉ muốn chọn 10 dòng số liệu đầu tiên, thì lệnh sẽ là:

```
> data3 <- chol[1:10, c(1,3,7)]
> print(data3)
   id sex  tc
1   1 Nam 4.0
2   2  Nu 3.5
3   3  Nu 4.7
4   4 Nam 7.7
5   5 Nam 5.0
6   6  Nu 4.2
7   7 Nam 5.9
8   8 Nam 6.1
```

```
9   9 Nam 5.9
10 10  Nu 4.0
```

Chú ý lệnh `print(arg)` đơn giản liệt kê tất cả số liệu trong data.frame `arg`. Thật ra, chúng ta chỉ cần đơn giản gõ `data3`, kết quả cũng giống y như `print(data3)`.

5.3 Nhập hai data.frame thành một: `merge`

Giả dụ như chúng ta có dữ liệu chứa trong hai data.frame. Dữ liệu thứ nhất tên là `d1` gồm 3 cột: `id, sex, tc` như sau:

```
 id sex tc
 1 Nam 4.0
 2  Nu 3.5
 3  Nu 4.7
 4 Nam 7.7
 5 Nam 5.0
 6  Nu 4.2
 7 Nam 5.9
 8 Nam 6.1
 9 Nam 5.9
10  Nu 4.0
```

Dữ liệu thứ hai tên là `d2` gồm 3 cột: `id, sex, tg` như sau:

```
id sex tg
 1 Nam 1.1
 2  Nu 2.1
 3  Nu 0.8
 4 Nam 1.1
 5 Nam 2.1
 6  Nu 1.5
 7 Nam 2.6
 8 Nam 1.5
 9 Nam 5.4
10  Nu 1.9
11  Nu 1.7
```

Hai dữ liệu này có chung hai biến số `id` và `sex`. Nhưng dữ liệu `d1` có 10 dòng, còn dữ liệu `d2` có 11 dòng. Chúng ta có thể nhập hai dữ liệu thành một data.frame bằng cách dùng lệnh `merge` như sau:

```
> d <- merge(d1, d2, by="id", all=TRUE)
> d
   id sex.x  tc sex.y  tg
```

```
1    1    Nam 4.0    Nam 1.1
2    2     Nu 3.5     Nu 2.1
3    3     Nu 4.7     Nu 0.8
4    4    Nam 7.7    Nam 1.1
5    5    Nam 5.0    Nam 2.1
6    6     Nu 4.2     Nu 1.5
7    7    Nam 5.9    Nam 2.6
8    8    Nam 6.1    Nam 1.5
9    9    Nam 5.9    Nam 5.4
10 10     Nu 4.0     Nu 1.9
11 11   <NA>  NA     Nu 1.7
```

Trong lệnh `merge`, chúng ta yêu cầu R nhập 2 dữ liệu `d1` và `d2` thành một và đưa vào data.frame mới tên là `d`, và dùng biến số `id` làm chuẩn. Chúng ta để ý thấy bệnh nhân số 11 không có số liệu cho `tc`, cho nên R cho là `NA` (một dạng "not available").

5.4 Biến đổi số liệu (data coding)

Trong việc xử lí số liệu dịch tễ học, nhiều khi chúng ta cần phải biến đổi số liệu từ biến liên tục sang biến mang tính cách phân loại. Chẳng hạn như trong chẩn đoán loãng xương, những phụ nữ có chỉ số T của mật độ chất khoáng trong xương (bone mineral density hay BMD) bằng hay thấp hơn -2.5 được xem là "loãng xương", những ai có BMD giữa -2.5 và -1.0 là "xốp xương" (osteopenia), và trên -1.0 là "bình thường". Ví dụ, chúng ta có số liệu BMD từ 10 bệnh nhân như sau:

```
-0.92, 0.21, 0.17, -3.21, -1.80, -2.60, -2.00, 1.71, 2.12, -2.11
```

Để nhập các số liệu này vào R chúng ta có thể sử dụng *function* `c` như sau:

```
bmd <- c(-0.92,0.21,0.17,-3.21,-1.80,-2.60,-2.00,1.71,2.12,-2.11)
```

Để phân loại 3 nhóm loãng xương, xốp xương, và bình thường, chúng ta có thể dùng mã số 1, 2 và 3. Nói cách khác, chúng ta muốn tạo nên một biến số khác (hãy gọi là `diagnosis`) gồm 3 giá trị trên dựa vào giá trị của `bmd`. Để làm việc này, chúng ta sử dụng lệnh:

```
# tạm thời cho biến số diagnosis bằng bmd
> diagnosis <- bmd

# biến đổi bmd thành diagnosis
> diagnosis[bmd <= -2.5] <- 1
> diagnosis[bmd > -2.5 & bmd <= 1.0] <- 2
> diagnosis[bmd > -1.0] <- 3

# tạo thành một data frame
> data <- data.frame(bmd, diagnosis)

# liệt kê để kiểm tra xem lệnh có hiệu quả không
> data
```

```
     bmd diagnosis
1  -0.92         3
2   0.21         3
3   0.17         3
4  -3.21         1
5  -1.80         2
6  -2.60         1
7  -2.00         2
8   1.71         3
9   2.12         3
10 -2.11         2
```

5.5 Biến đổi số liệu bằng cách dùng *replace*

Một cách biến đổi số liệu khác là dùng `replace`, dù cách này có vẻ rườm rà chút ít. Tiếp tục ví dụ trên, chúng ta biến đổi từ `bmd` sang `diagnosis` như sau:

```
> diagnosis <- bmd
> diagnosis <- replace(diagnosis, bmd <= -2.5, 1)
> diagnosis <- replace(diagnosis, bmd > -2.5 & bmd <= 1.0, 2)
> diagnosis <- replace(diagnosis, bmd > -1.0, 3)
```

5.6 Biến đổi thành yếu tố (*factor*)

Trong phân tích thống kê, chúng ta phân biệt một biến số mang tính *yếu tố* (factor) và biến số liên tục bình thường. Biến số yếu tố không thể dùng để tính toán như cộng trừ nhân chia, nhưng biến số số học có thể sử dụng để tính toán. Chẳng hạn như trong ví dụ `bmd` và `diagnosis` trên, `diagnosis` là yếu tố vì giá trị trung bình giữa 1 và 2 chẳng có ý nghĩa thực tế gì cả; còn `bmd` là biến số số học.

Nhưng hiện nay, `diagnosis` được xem là một biến số số học. Để biến thành biến số yếu tố, chúng ta cần sử dụng *function* `factor` như sau:

```
> diag <- factor(diagnosis)
> diag
[1] 3 3 3 1 2 1 2 3 3 2
Levels: 1 2 3
```

Chú ý R bây giờ thông báo cho chúng ta biết diag có 3 bậc: 1, 2 và 3. Nếu chúng ta yêu cầu R tính số trung bình của diag, R sẽ không làm theo yêu cầu này, vì đó không phải là một biến số số học:

```
> mean(diag)
[1] NA
Warning message:
argument is not numeric or logical: returning NA in: mean.default(diag)
```

Dĩ nhiên, chúng ta có thể tính giá trị trung bình của `diagnosis`:

```
> mean(diagnosis)
[1] 2.3
```

nhưng kết quả 2.3 này không có ý nghĩa gì trong thực tế cả.

5.7 Phân nhóm số liệu bằng *cut2* (Hmisc)

Trong phân tích thống kê, có khi chúng ta cần phải phân chia một biến số liên tục thành nhiều nhóm dựa vào phân phối của biến số. Chẳng hạn như đối với biến số `bmd` chúng ta có thể "cắt" dãy số thành 3 nhóm tương đương nhau bằng cách dùng function `cut2` (trong thư viện `Hmisc`) như sau:

```
> # nhập thư viện Hmisc để có thể dùng function cut2
> library(Hmisc)
> bmd <- c(-0.92,0.21,0.17,-3.21,-1.80,-2.60,-2.00,1.71,2.12,-2.11)
> # chia biến số bmd thành 2 nhóm và để trong đối tượng group
> group <- cut2(bmd, g=2)
> table(group)
group
[-3.21,-0.92) [-0.92, 2.12]
            5             5
```

Như thấy qua ví dụ trên, `g = 2` có nghĩa là chia thành 2 nhóm (g = group). R tự động chia thành nhóm 1 gồm giá trị `bmd` từ -3.21 đến -0.92, và nhóm 2 từ -0.92 đến 2.12. Mỗi nhóm gồm có 5 số.

Tất nhiên, chúng ta cũng có thể chia thành 3 nhóm bằng lệnh:

```
> group <- cut2(bmd, g=3)
```

Và với lệnh table chúng ta sẽ biết có 3 nhóm, nhóm 1 gồm 4 số, nhóm 2 và 3 mỗi nhóm có 3 số:

```
> table(group)
group
[-3.21,-1.80) [-1.80, 0.21) [ 0.21, 2.12]
            4             3             3
```

6. Sử dụng R cho tính toán đơn giản

Một trong những lợi thế của R là có thể sử dụng như một … máy tính cầm tay. Thật ra, hơn thế nữa, R có thể sử dụng cho các phép tính ma trận và lập chương. Trong chương này tôi chỉ trình bày một số phép tính đơn giản mà học sinh hay sinh viên có thể sử dụng lập tức trong khi đọc những dòng chữ này.

6.1 Tính toán đơn giản

Cộng hai số hay nhiều số với nhau:
```
> 15+2997
[1] 3012
```

Cộng và trừ:
```
> 15+2997-9768
[1] -6756
```

Nhân và chia
```
> -27*12/21
[1] -15.42857
```

Số lũy thừa: $(25 - 5)^3$
```
> (25 - 5)^3
[1] 8000
```

Căn số bậc hai: $\sqrt{10}$
```
> sqrt(10)
[1] 3.162278
```

Số pi (π)
```
> pi
[1] 3.141593
> 2+3*pi
[1] 11.42478
```

Logarit: $\log_e$
```
> log(10)
[1] 2.302585
```

Logarit: $\log_{10}$
```
> log10(100)
[1] 2
```

Số mũ: $e^{2.7689}$
```
> exp(2.7689)
[1] 15.94109

> log10(2+3*pi)
[1] 1.057848
```

Hàm số lượng giác
```
> cos(pi)
[1] -1
```

Vector
```
> x <- c(2,3,1,5,4,6,7,6,8)
> x
[1] 2 3 1 5 4 6 7 6 8

> sum(x)
[1] 42

> x*2
[1]  4  6  2 10  8 12 14 12 16
```

```
> exp(x/10)
[1] 1.221403 1.349859 1.105171 1.648
1.491825 1.822119 2.013753 1.822119
[9] 2.225541

> exp(cos(x/10))
[1] 2.664634 2.599545 2.704736 2.405
2.511954 2.282647 2.148655 2.282647
[9] 2.007132
```

Tính tổng bình phương (sum of squares): $1^2 + 2^2 + 3^2 + 4^2 + 5^2 = ?$
```
> x <- c(1,2,3,4,5)
> sum(x^2)
 [1] 55
```

Tính tổng bình phương điều chỉnh (adjusted sum of squares): $\sum_{i=1}^{n}(x_i - \bar{x})^2 = ?$
```
> x <- c(1,2,3,4,5)
> sum((x-mean(x))^2)
[1] 10
```

Trong công thức trên `mean(x)` là số trung bình của vector `x`.

Tính sai số bình phương (mean square):

Tính phương sai (variance) và độ lệch chuẩn (standard deviation):

$\sum_{i=1}^{n}(x_i-\bar{x})^2/n=?$
`> x <- c(1,2,3,4,5)`
`> sum((x-mean(x))^2)/length(x)`
`[1] 2`
 	
Trong công thức trên, `length(x)` có nghĩa là tổng số phần tử (elements) trong vector `x`.
	Phương sai: $s^2=\sum_{i=1}^{n}(x_i-\bar{x})^2/(n-1)=?$
	`> x <- c(1,2,3,4,5)`
	`> var(x)`
	`[1] 2.5`
	Độ lệch chuẩn: $\sqrt{s^2}$:
	`> sd(x)`
	`[1] 1.581139`

6.2 Sử dụng R cho các phép tính ma trận

Như chúng ta biết ma trận (matrix), nói đơn giản, gồm có dòng (row) và cột (column). Khi viết **A**[*m*, *n*], chúng ta hiểu rằng ma trận **A** có *m* dòng và *n* cột. Trong R, chúng ta cũng có thể thể hiện như thế. Ví dụ: chúng ta muốn tạo một ma trận vuông A gồm 3 dòng và 3 cột, với các phần tử (element) 1, 2, 3, 4, 5, 6, 7, 8, 9, chúng ta viết:

$$A=\begin{pmatrix}1 & 4 & 7\\ 2 & 5 & 8\\ 3 & 6 & 9\end{pmatrix}$$

Và với R:

```
> y <- c(1,2,3,4,5,6,7,8,9)
> A <- matrix(y, nrow=3)
> A
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

Nhưng nếu chúng ta lệnh:

```
> A <- matrix(y, nrow=3, byrow=TRUE)
> A
```

thì kết quả sẽ là:

```
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
```

Tức là một **ma trận chuyển vị (transposed matrix).** Một cách khác để tạo một ma trận hoán vị là dùng `t()`. Ví dụ:

```
> y <- c(1,2,3,4,5,6,7,8,9)
> A <- matrix(y, nrow=3)
> A
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

và B = A' có thể diễn tả bằng R như sau:

```
> B <- t(A)
> B
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
```

Ma trận vô hướng (scalar matrix) là một ma trận vuông (tức số dòng bằng số cột), và tất cả các phần tử ngoài đường chéo (off-diagonal elements) là 0, và phần tử đường chéo là 1. Chúng ta có thể tạo một ma trận như thế bằng R như sau:

```
> # tạo ra mộ ma trận 3 x 3 với tất cả phần tử là 0.
> A <- matrix(0, 3, 3)

> # cho các phần tử đường chéo bằng 1
> diag(A) <- 1
> diag(A)
[1] 1 1 1

> # bây giờ ma trận A sẽ là:
> A
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
```


6.2.1 Chiết phần tử từ ma trận

```
> y <- c(1,2,3,4,5,6,7,8,9)
> A <- matrix(y, nrow=3)
> A
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

```
> # cột 1 của ma trận A
> A[,1]
```

```
[1] 1 4 7

> # cột 3 của ma trận A
> A[3,]
[1] 7 8 9

> # dòng 1 của ma trận A
> A[1,]
[1] 1 2 3

> # dòng 2, cột 3 của ma trận A
> A[2,3]
[1] 6

> # tất cả các dòng của ma trận A, ngoại trừ dòng 2
> A[-2,]
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    3    6    9

> # tất cả các cột của ma trận A, ngoại trừ cột 1
> A[,-1]
     [,1] [,2]
[1,]    4    7
[2,]    5    8
[3,]    6    9

> # xem phần tử nào cao hơn 3.
> A>3
      [,1] [,2] [,3]
[1,] FALSE TRUE TRUE
[2,] FALSE TRUE TRUE
[3,] FALSE TRUE TRUE
```


6.2.2 Tính toán với ma trận

Cộng và trừ hai ma trận. Cho hai ma trận A và B như sau:

```
> A <- matrix(1:12, 3, 4)
> A
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12

> B <- matrix(-1:-12, 3, 4)
> B
     [,1] [,2] [,3] [,4]
[1,]   -1   -4   -7  -10
```

```
[2,]   -2   -5   -8  -11
[3,]   -3   -6   -9  -12
```

Chúng ta có thể cộng A+B:

```
> C <- A+B
> C
     [,1] [,2] [,3] [,4]
[1,]    0    0    0    0
[2,]    0    0    0    0
[3,]    0    0    0    0
```

Hay A-B:

```
> D <- A-B
> D
     [,1] [,2] [,3] [,4]
[1,]    2    8   14   20
[2,]    4   10   16   22
[3,]    6   12   18   24
```

Nhân hai ma trận. Cho hai ma trận:

$$A = \begin{pmatrix} 1 & 4 & 7 \\ 2 & 5 & 8 \\ 3 & 6 & 9 \end{pmatrix} \quad \text{và} \quad B = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

Chúng ta muốn tính *AB*, và có thể triển khai bằng R bằng cách sử dụng %*% như sau:

```
> y <- c(1,2,3,4,5,6,7,8,9)
> A <- matrix(y, nrow=3)
> B <- t(A)
> AB <- A%*%B
> AB
     [,1] [,2] [,3]
[1,]   66   78   90
[2,]   78   93  108
[3,]   90  108  126
```

Hay tính *BA*, và có thể triển khai bằng R bằng cách sử dụng %*% như sau:

```
> BA <- B%*%A
> BA
     [,1] [,2] [,3]
[1,]   14   32   50
[2,]   32   77  122
[3,]   50  122  194
```

Nghịch đảo ma trận và giải hệ phương trình. Ví dụ chúng ta có hệ phương trình sau đây:

$$3x_1 + 4x_2 = 4$$
$$x_1 + 6x_2 = 2$$

Hệ phương trình này có thể viết bằng kí hiệu ma trận: $AX = Y$, trong đó:

$$A = \begin{pmatrix} 3 & 4 \\ 1 & 6 \end{pmatrix}, \quad X = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \quad \text{và} \quad Y = \begin{pmatrix} 4 \\ 2 \end{pmatrix}$$

Nghiệm của hệ phương trình này là: $X = A^{-1}Y$, hay trong R:

```
> A <- matrix(c(3,1,4,6), nrow=2)
> Y <- matrix(c(4,2), nrow=2)
> X <- solve(A)%*%Y
> X
          [,1]
[1,] 1.1428571
[2,] 0.1428571
```

Chúng ta có thể kiểm tra:

```
> 3*X[1,1]+4*X[2,1]
[1] 4
```

Trị số eigen cũng có thể tính toán bằng function `eigen` như sau:

```
> eigen(A)
$values
[1] 7 2

$vectors
           [,1]       [,2]
[1,] -0.7071068 -0.9701425
[2,] -0.7071068  0.2425356
```

Định thức (determinant). Làm sao chúng ta xác định một ma trận có thể đảo nghịch hay không? Ma trận mà định thức bằng 0 là **ma trận suy biến** (**singular matrix**) và không thể đảo nghịch. Để kiểm tra định thức, R dùng lệnh `det()`:

```
> E <- matrix((1:9), 3, 3)
> E
     [,1] [,2] [,3]
[1,]    1    4    7
[2,]    2    5    8
[3,]    3    6    9
```

```
> det(E)
[1] 0
```

Nhưng ma trận F sau đây thì có thể đảo nghịch:

```
> F <- matrix((1:9)^2, 3, 3)
> F
     [,1] [,2] [,3]
[1,]    1   16   49
[2,]    4   25   64
[3,]    9   36   81
> det(F)
[1] -216
```

Và nghịch đảo của ma trận F (F^{-1}) có thể tính bằng function `solve()` như sau:

```
> solve(F)
          [,1]      [,2]       [,3]
[1,]  1.291667 -2.166667  0.9305556
[2,] -1.166667  1.666667 -0.6111111
[3,]  0.375000 -0.500000  0.1805556
```

Ngoài những phép tính đơn giản này, R còn có thể sử dụng cho các phép tính phức tạp khác. Một lợi thế đáng kể của R là phần mềm cung cấp cho người sử dụng tự do tạo ra những phép tính phù hợp cho từng vấn đề cụ thể. R có một package `Matrix` chuyên thiết kế cho tính toán ma trận. Bạn đọc có thể tải package xuống, cài vào máy, và sử dụng, nếu cần. Địa chỉ để tải là: http://cran.au.r-project.org/bin/windows/contrib/r-release/Matrix_0.995-8.zip cùng với tài liệu chỉ dẫn cách sử dụng (dài khoảng 80 trang): http://cran.au.r-project.org/doc/packages/Matrix.pdf.

7. Sử dụng R cho tính toán xác suất

7.1 Phép hoán vị (permutation)

Chúng ta biết rằng 3! = 3.2.1 = 6, và 0!=1. Nói chung, công thức tính hoán vị cho một số n là: $n! = n(n-1)(n-2)(n-3)\times...\times 1$. Trong R cách tính này rất đơn giản với lệnh `prod()` như sau:

- Tìm 3!

```
> prod(3:1)
[1] 6
```

- Tìm 10!

```
> prod(10:1)
[1] 3628800
```

- Tìm 10.9.8.7.6.5.4

```
> prod(10:4)
[1] 604800
```

- Tìm (10.9.8.7.6.5.4) / (40.39.38.37.36)

```
> prod(10:4) / prod(40:36)
[1] 0.007659481
```

7.2 Tổ hợp (combination)

Số lần chọn k người từ n phần tử là: $\binom{n}{k} = \frac{n!}{k!(n-k)!}$. Công thức này cũng có khi viết là C_k^n thay vì $\binom{n}{k}$. Với R, phép tính này rất đơn giản bằng hàm `choose(n, k)`. Sau đây là vài ví dụ minh họa:

- Tìm $\binom{5}{2}$

```
> choose(5, 2)
[1] 10
```

- Tìm xác suất cặp A và B trong số 5 người được đắc cử vào hai chức vụ:

```
> 1/choose(5, 2)
[1] 0.1
```


7.3 Biến số ngẫu nhiên và hàm phân phối

Khi nói đến "phân phối" (hay distribution) là để cập đến các giá trị mà biến số có thể có. Các *hàm phân phối* (distribution function) là hàm nhằm mô tả các biến số đó một cách có hệ thống. "Có hệ thống" ở đây có nghĩa là theo mộ mô hình toán học cụ thể với những thông số cho trước. Trong xác suất thống kê có khá nhiều hàm phân phối, và ở đây chúng ta sẽ xem xét qua một số hàm quan trọng nhất và thông dụng nhất: đó là phân phối nhị phân, phân phối Poisson, và phân phối chuẩn. Trong mỗi luật phân phối, có 4 loại hàm quan trọng mà chúng ta cần biết:

- hàm mật độ xác suất (probability density distribution);
- hàm phân phối tích lũy (cumulative probability distribution);
- hàm định bậc (quantile); và
- hàm mô phỏng (simulation).

R có những hàm sẵn trên có thể ứng dụng cho tính toán xác suất. Tên mỗi hàm được gọi bằng một tiếp đầu ngữ để chỉ loại hàm phân phối, và viết tắt tên của hàm đó. Các tiếp đầu ngữ là `d` (chỉ distribution hay xác suất), `p` (chỉ cumulative probability, xác suất tích lũy), `q` (chỉ định bậc hay quantile), và `r` (chỉ random hay số ngẫu nhiên). Các

tên viết tắt là `norm` (normal, phân phối chuẩn), `binom` (binomial , phân phối nhị phân), `pois` (Poisson, phân phối Poisson), v.v… Bảng sau đây tóm tắt các hàm và thông số cho từng hàm:

Hàm phân phối	Mật độ	Tích lũy	Định bậc	Mô phỏng
Chuẩn	`dnorm(x, mean, sd)`	`pnorm(q, mean, sd)`	`qnorm(p, mean, sd)`	`rnorm(n, mean, sd)`
Nhị phân	`dbinom(k, n, p)`	`pbinom(q, n, p)`	`qbinom (p, n, p)`	`rbinom(k, n, prob)`
Poisson	`dpois(k, lambda)`	`ppois(q, lambda)`	`qpois(p, lambda)`	`rpois(n, lambda)`
Uniform	`dunif(x, min, max)`	`punif(q, min, max)`	`qunif(p, min, max)`	`runif(n, min, max)`
Negative binomial	`dnbinom(x, k, p)`	`pnbinom(q, k, p)`	`qnbinom (p,k,prob)`	`rbinom(n, n, prob)`
Beta	`dbeta(x, shape1, shape2)`	`pbeta(q, shape1, shape2)`	`qbeta(p, shape1, shape2)`	`rbeta(n, shape1, shape2)`
Gamma	`dgamma(x, shape, rate, scale)`	`gamma(q, shape, rate, scale)`	`qgamma(p, shape, rate, scale)`	`rgamma(n, shape, rate, scale)`
Geometric	`dgeom(x, p)`	`pgeom(q, p)`	`qgeom(p, prob)`	`rgeom(n, prob)`
Exponential	`dexp(x, rate)`	`pexp(q, rate)`	`qexp(p, rate)`	`rexp(n, rate)`
Weibull	`dnorm(x, mean, sd)`	`pnorm(q, mean, sd)`	`qnorm(p, mean, sd)`	`rnorm(n, mean, sd)`
Cauchy	`dcauchy(x, location, scale)`	`pcauchy(q, location, scale)`	`qcauchy(p, location, scale)`	`rcauchy(n, location, scale)`
F	`df(x, df1, df2)`	`pf(q, df1, df2)`	`qf(p, df1, df2)`	`rf(n, df1, df2)`
T	`dt(x, df)`	`pt(q, df)`	`qt(p, df)`	`rt(n, df)`
Chi-squared	`dchisq(x, df)`	`pchi(q, df)`	`qchisq(p, df)`	`rchisq(n, df)`

Chú thích: Trong bảng trên, df = degrees of freedome (bậc tự do); prob = probability (xác suất); n = sample size (số lượng mẫu). Các thông số khác có thể tham khảo thêm cho từng luật phân phối. Riêng các luật phân phối F, t, Chi-squared còn có một thông số khác nữa là non-centrality parameter (ncp) được cho số 0. Tuy nhiên người sử dụng có thể cho một thông số khác thích hợp, nếu cần.

7.3.1 Hàm phân phối nhị phân (Binomial distribution)

Như tên gọi, hàm phân phối nhị phân chỉ có hai giá trị: nam / nữ, sống / chết, có / không, v.v… Hàm nhị phân được phát biểu bằng định lí như sau: Nếu một thử nghiệm được tiến hành n lần, mỗi lần cho ra kết quả hoặc là thành công hoặc là thất bại, và gồm xác suất thành công được biết trước là p, thì xác suất có k lần thử nghiệm thành công là: $P(k \mid n, p) = C_k^n p^k (1-p)^{n-k}$, trong đó $k = 0, 1, 2, \ldots, n$. Trong R, có hàm `dbinom(k, n, p)` có thể giúp chúng ta tính công thức $P(k \mid n, p) = C_k^n p^k (1-p)^{n-k}$ một cách nhanh chóng. Trong trường hợp trên, chúng ta chỉ cần đơn giản lệnh:

```
> dbinom(2, 3, 0.60)
[1] 0.432
```

Ví dụ 2: **Hàm nhị phân tích lũy (Cumulative Binomial probability distribution)**. Xác suất thuốc chống loãng xương có hiệu nghiệm là khoảng 70% (tức là $p = 0.70$). Nếu chúng ta điều trị 10 bệnh nhân, xác suất có tối thiểu 8 bệnh nhân với kết quả tích cực là bao nhiêu? Nói cách khác, nếu gọi X là số bệnh nhân được điều trị thành công, chúng ta cần tìm $P(X \geq 8) = ?$ Để trả lời câu hỏi này, chúng ta sử dụng hàm

`pbinom(k, n, p)`. Xin nhắc lại rằng hàm `pbinom(k, n, p)` cho chúng ta $P(X \leq k)$. Do đó, $P(X \geq 8) = 1 - P(X \leq 7)$. Thành ra, đáp số bằng R cho câu hỏi là:

```
> 1-pbinom(7, 10, 0.70)
[1] 0.3827828
```

Ví dụ 3: Mô phỏng hàm nhị phân: Biết rằng trong một quần thể dân số có khoảng 20% người mắc bệnh cao huyết áp; nếu chúng ta tiến hành chọn mẫu 1000 lần, mỗi lần chọn 20 người trong quần thể đó một cách ngẫu nhiên, sự phân phối số bệnh nhân cao huyết áp sẽ như thế nào? Để trả lời câu hỏi này, chúng ta có thể ứng dụng hàm `rbinom (n, k, p)` trong R với những thông số như sau:

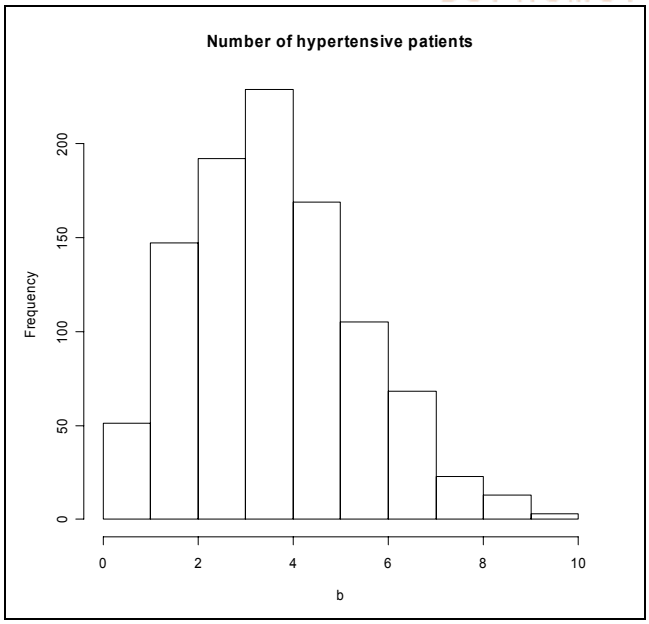
```
> b <- rbinom(1000, 20, 0.20)
```

Trong lệnh trên, kết quả mô phỏng được tạm thời chứa trong đối tượng tên là `b`. Để biết `b` có gì, chúng ta đếm bằng lệnh `table`:

```
> table(b)
b
  0   1   2   3   4   5   6   7   8   9  10
  6  45 147 192 229 169 105  68  23  13   3
```

Dòng số liệu thứ nhất (`0, 5, 6, …, 10`) là số bệnh nhân mắc bệnh cao huyết áp trong số 20 người mà chúng ta chọn. Dòng số liệu thứ hai cho chúng ta biết số lần chọn mẫu trong 1000 lần xảy ra. Do đó, có 6 mẫu không có bệnh nhân cao huyết áp nào, 45 mẫu với chỉ 1 bệnh nhân cao huyết áp, v.v… Có lẽ cách để hiểu là vẽ đồ thị các tần số trên bằng lệnh `hist` như sau:

```
> hist(b, main="Number of hypertensive patients")
```



Biểu đồ 1. Phân phối số bệnh nhân cao huyết áp trong số 20 người được chọn ngẫu nhiên trong một quần thể gồm 20% bệnh nhân cao huyết áp, và chọn mẫu được lặp lại 1000 lần.

Qua biểu đồ trên, chúng ta thấy xác suất có 4 bệnh nhân cao huyết áp (trong mỗi lần chọn mẫu 20 người) là cao nhất (22.9%). Điều này cũng có thể hiểu được, bởi vì tỉ lệ cao huyết áp là 20%, cho nên chúng ta kì vọng rằng trung bình 4 người trong số 20 người được chọn phải là cao huyết áp. Tuy nhiên, điều quan trọng mà biểu đồ trên thể hiện là có khi chúng ta quan sát đến 10 bệnh nhân cao huyết áp dù xác suất cho mẫu này rất thấp (chỉ 3/1000).

7.3.2 Hàm phân phối Poisson (Poisson distribution)

Hàm phân phối Poisson, nói chung, rất giống với hàm nhị phân, ngoại trừ thông số p thường rất nhỏ và n thường rất lớn. Vì thế, hàm Poisson thường được sử dụng để mô tả các biến số rất hiếm xảy ra (như số người mắc ung thư trong một dân số chẳng hạn). Hàm Poisson còn được ứng dụng khá nhiều và thành công trong các nghiên cứu kĩ thuật và thị trường như số lượng khách hàng đến một nhà hàng mỗi giờ.

Ví dụ 4: **Hàm mật độ Poisson (Poisson density probability function)**. Qua theo dõi nhiều tháng, người ta biết được tỉ lệ đánh sai chính tả của một thư kí đánh máy. Tính trung bình cứ khoảng 2.000 chữ thì thư kí đánh sai 1 chữ. Hỏi xác suất mà thư kí đánh sai chính tả 2 chữ, hơn 2 chữ là bao nhiêu?

Vì tần số khá thấp, chúng ta có thể giả định rằng biến số "sai chính tả" (tạm đặt tên là biến số X) là một hàm ngẫu nhiên theo luật phân phối Poisson. Ở đây, chúng ta có tỉ lệ sai chính tả trung bình là 1($\lambda = 1$). Luật phân phối Poisson phát biểu rằng xác suất mà $X = k$, với điều kiện tỉ lệ trung bình λ, :

$$P(X = k \mid \lambda) = \frac{e^{-\lambda}\lambda^{k}}{k!}$$

Do đó, đáp số cho câu hỏi trên là: $P(X = 2 \mid \lambda = 1) = \frac{e^{-2}1^{2}}{2!} = 0.1839$. Đáp số này có thể tính bằng R một cách nhanh chóng hơn bằng hàm `dpois` như sau:

```
> dpois(2, 1)
[1] 0.1839397
```

Chúng ta cũng có thể tính xác suất sai 1 chữ, và xác suất không sai chữ nào:

```
> dpois(1, 1)
[1] 0.3678794

> dpois(0, 1)
```

```
[1] 0.3678794
```

Chú ý trong hàm trên, chúng ta chỉ đơn giản cung cấp thông số $k = 2$ và ($\lambda = 1$. Trên đây là xác suất mà thư kí đánh sai chính tả đúng 2 chữ. Nhưng xác suất mà thư kí đánh sai chính tả hơn 2 chữ (tức 3, 4, 5, … chữ) có thể ước tính bằng:

$$\begin{aligned} P(X > 2) &= P(X = 3) + P(X = 4) + P(X = 5) + \ldots \\ &= 1 - P(X \le 2) \\ &= 1 - 0.3678 - 0.3678 - 0.1839 \\ &= 0.08 \end{aligned}$$

Bằng R, chúng ta có thể tính như sau:

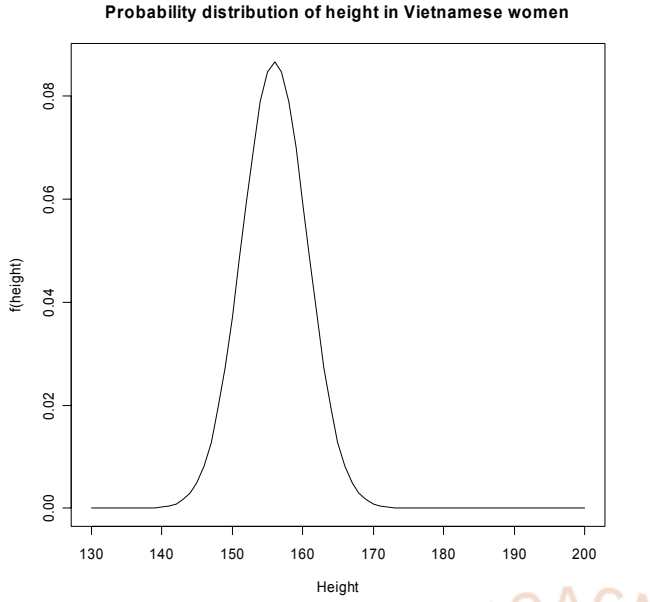
```
# P(X ≤ 2)
> ppois(2, 1)
[1] 0.9196986

# 1-P(X ≤ 2)
> 1-ppois(2, 1)
[1] 0.0803014
```

7.3.3 Hàm phân phối chuẩn (Normal distribution)

Hai luật phân phối mà chúng ta vừa xem xét trên đây thuộc vào nhóm phân phối áp dụng cho các biến số phi liên tục (discrete distributions), mà trong đó biến số có những giá trị theo bậc thứ hay thể loại. Đối với các biến số liên tục, có vài luật phân phối thích hợp khác, mà quan trọng nhất là phân phối chuẩn. Phân phối chuẩn là nền tảng quan trọng nhất của phân tích thống kê. Có thể nói không ngoa rằng hầu hết lí thuyết thống kê được xây dựng trên nền tảng của phân phối chuẩn. Hàm mật độ phân phối chuẩn có hai thông số: trung bình μ và phương sai σ^2 (hay độ lệch chuẩn σ). Gọi X là một biến số (như chiều cao chẳng hạn), hàm mật độ phân phối chuẩn phát biểu rằng xác suất mà $X = x$ là:

$$P\left(X = x \mid \mu, \sigma^2\right) = f\left(x\right) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{\left(x-\mu\right)^2}{2\sigma^2}\right]$$

Ví dụ 5: **Hàm mật độ phân phối chuẩn (Normal density probability function)**. Chiều cao trung bình hiện nay ở phụ nữ Việt Nam là 156 cm, với độ lệch chuẩn là 4.6 cm. Cũng biết rằng chiều cao này tuân theo luật phân phối chuẩn. Với hai thông số μ=156, σ=4.6, chúng ta có thể xây dựng một hàm phân phối chiều cao cho toàn bộ quần thể phụ nữ Việt Nam, và hàm này có hình dạng như sau:



Biểu đồ 2. Phân phối chiều cao ở phụ nữ Việt Nam với trung bình 156 cm và độ lệch chuẩn 4.6 cm. Trụng hoành là chiều cao và trục tung là xác suất cho mỗi chiều cao.

Biểu đồ trên được vẽ bằng hai lệnh sau đây. Lệnh đầu tiên nhằm tạo ra một biến số `height` có giá trị 130, 131, 132, …, 200 cm. Lệnh thứ hai là vẽ biểu đồ với điều kiện trung bình là 156 cm và độ lệch chuẩn là 4.6 cm.

```
> height <- seq(130, 200, 1)
> plot(height, dnorm(height, 156, 4.6),
       type="l",
       ylab="f(height)",
       xlab="Height",
       main="Probability distribution of height in Vietnamese women")
```

Với hai thông số trên (và biểu đồ), chúng ta có thể ước tính xác suất cho bất cứ chiều cao nào. Chẳng hạn như xác suất một phụ nữ Việt Nam có chiều cao 160 cm là:

$$P(X = 160 \mid \mu=156, \sigma=4.6) = \frac{1}{4.6\sqrt{2\times 3.1416}} \exp\left[-\frac{(160-156)^2}{2(4.6)^2}\right]$$
$$= 0.0594$$

Hàm `dnorm(x, mean, sd)` trong R có thể tính toán xác suất này cho chúng ta một cách gọn nhẹ:

```
> dnorm(160, mean=156, sd=4.6)
[1] 0.05942343
```

Hàm xác suất chuẩn tích lũy (cumulative normal probability function). Vì chiều cao là một biến số liên tục, trong thực tế chúng ta ít khi nào muốn tìm xác suất cho một giá trị cụ thể *x*, mà thường tìm xác suất cho một khoảng giá trị *a* đến *b*. Chẳng hạn như chúng ta muốn biết xác suất chiều cao từ 150 đến 160 cm (tức là $P(160 \le X \le 150)$, hay xác suất chiều cao thấp hơn 145 cm, tức $P(X < 145)$. Để tìm đáp số các câu hỏi như thế, chúng ta cần đến hàm xác suất chuẩn tích lũy, được định nghĩa như sau:

$$P(a \le X \le b) = \int_a^b f(x)\,dx$$

Thành ra, $P(160 \le X \le 150)$ chính là diện tích tính từ trục hoành = 150 đến 160 của **biểu đồ 2**. Trong R có hàm `pnorm(x, mean, sd)` dùng để tính xác suất tích lũy cho một phân phối chuẩn rất có ích.

$$\texttt{pnorm(a, mean, sd)} = \int_{-\infty}^{a} f(x)\,dx = P(X \le a \mid \text{mean, sd})$$

Chẳng hạn như xác suất chiều cao phụ nữ Việt Nam bằng hoặc thấp hơn 150 cm là 9.6%:

```
> pnorm(150, 156, 4.6)
[1] 0.0960575
```

Hay xác suất chiều cao phụ nữ Việt Nam bằng hoặc cao hơn 165 cm là:

```
> 1-pnorm(164, 156, 4.6)
[1] 0.04100591
```

Nói cách khác, chỉ có khoảng 4.1% phụ nữ Việt Nam có chiều cao bằng hay cao hơn 165 cm.

Ví dụ 6: Ứng dụng luật phân phối chuẩn: Trong một quần thể, chúng ta biết rằng áp suất máu trung bình là 100 mmHg và độ lệch chuẩn là 13 mmHg, hỏi: có bao nhiêu ngươi trong quần thể này có áp suất máu bằng hoặc cao hơn 120 mmHg? Câu trả lời bằng R là:

```
> 1-pnorm(120, mean=100, sd=13)
[1] 0.0619679
```

Tức khoảng 6.2% người trong quần thể này có áp suất máu bằng hoặc cao hơn 120 mmHg.

7.3.4 Hàm phân phối chuẩn chuẩn hóa (Standardized Normal distribution)

Một biến X tuân theo luật phân phối chuẩn với trung bình bình μ và phương sai σ^2 thường được viết tắt là:

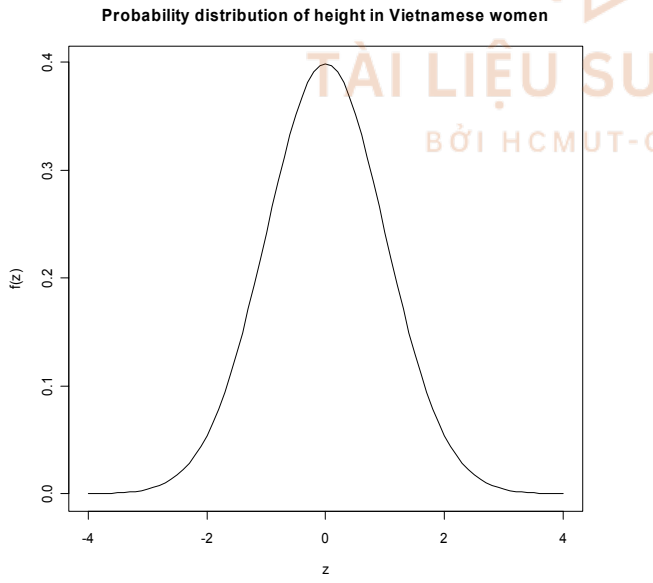
$$X \sim N(\mu, \sigma^2)$$

Ở đây μ và σ^2 tùy thuộc vào đơn vị đo lường của biến số. Chẳng hạn như chiều cao được tính bằng cm (hay m), huyết áp được đo bằng mmHg, tuổi được đo bằng năm, v.v… cho nên đôi khi mô tả một biến số bằng đơn vị gốc rất khó so sánh. Một cách đơn giản hơn là chuẩn hóa (standardized) X sao cho số trung bình là 0 và phương sai là 1. Sau vài thao tác số học, có thể chứng minh dễ dàng rằng, cách biến đổi X để đáp ứng điều kiện trên là:

$$Z = \frac{X - \mu}{\sigma}$$

Nói theo ngôn ngữ toán: nếu $X \sim N(\mu, \sigma^2)$, thì $(X - \mu)/\sigma^2 \sim N(0, 1)$. Như vậy qua công thức trên, Z thực chất là độ khác biệt giữa một số và trung bình tính bằng số độ lệch chuẩn. Nếu $Z = 0$, chúng ta biết rằng X bằng số trung bình μ. Nếu $Z = -1$, chúng ta biết rằng X thấp hơn μ đúng 1 độ lệch chuẩn. Tương tự, $Z = 2.5$, chúng ta biết rằng X cao hơn μ đúng 2.5 độ lệch chuẩn. v.v…

Biểu đồ phân phối chiều cao của phụ nữ Việt Nam có thể mô tả bằng một đơn vị mới, đó là chỉ số z như sau:



Biểu đồ 3. Phân phối chuẩn hóa chiều cao ở phụ nữ Việt Nam.

Biểu đồ trên được vẽ bằng hai lệnh sau đây:

```
> height <- seq(-4, 4, 0.1)
> plot(height, dnorm(height, 0, 1),
       type="l",
       ylab="f(z)",
       xlab="z",
       main="Probability distribution of height in Vietnamese women")
```

Với phân phối chuẩn chuẩn hoá, chúng ta có một tiện lợi là có thể dùng nó để mô tả và so sánh mật độ phân phối của bất cứ biến nào, vì tất cả đều được chuyển sang chỉ số z.

Trong biểu đồ trên, trục tung là xác suất z và trục hoành là biến số z. Chúng ta có thể tính toán xác suất z nhỏ hơn một hằng số (constant) nào đó dê dàng bằng R. Ví dụ, chúng ta muốn tìm $P(z \leq -1.96) = ?$ cho một phân phối mà trung bình là 0 và độ lệch chuẩn là 1.

```
> pnorm(-1.96, mean=0, sd=1)
[1] 0.02499790
```

Hay $P(z \leq 1.96) = ?$

```
> pnorm(1.96, mean=0, sd=1)
[1] 0.9750021
```

Do đó, $P(-1.96 < z < 1.96)$ chính là:

```
> pnorm(1.96) - pnorm(-1.96)
[1] 0.9500042
```

Nói cách khác, xác suất 95% là z nằm giữa -1.96 và 1.96. (Chú ý trong lệnh trên tôi không cung cấp `mean=0, sd=1`, bởi vì trong thực tế, `pnorm` giá trị mặc định (default value) của thông số `mean` là 0 và `sd` là 1).

Ví dụ 5 (tiếp tục). Xin nhắc lại để tiện việc theo dõi, chiều cao trung bình ở phụ nữ Việt Nam là 156 cm và độ lệch chuẩn là 4.6 cm. Do đó, một phụ nữ có chiều cao 170 cm cũng có nghĩa là z = (170 – 156) / 4.6 = 3.04 độ lệch chuẩn, và ti lệ các phụ nữ Việt Nam có chiều cao cao hơn 170 cm là rất thấp, chỉ khoảng 0.1%.

```
> 1-pnorm(3.04)
[1] 0.001182891
```

Tìm định lượng (quantile) của một phân phối chuẩn. Đôi khi chúng ta cần làm một tính toán đảo ngược. Chẳng hạn như chúng ta muốn biết: nếu xác suất Z nhỏ hơn một hằng số z nào đó cho trước bằng p, thì z là bao nhiêu? Diễn tả theo kí hiệu xác suất, chúng ta muốn tìm z trong nếu:

$$P(Z < z) = p$$

Để trả lời câu hỏi này, chúng ta sử dụng hàm `qnorm(p, mean=, sd=)`.

Ví dụ 7: Biết rằng $Z \sim N(0, 1)$ và nếu $P(Z < z) = 0.95$, chúng ta muốn tìm z.

```
> qnorm(0.95, mean=0, sd=1)
[1] 1.644854
```

Hay $P(Z < z) = 0.975$ cho phân phối chuẩn với trung bình 0 và độ lệch chuẩn 1:

```
> qnorm(0.975, mean=0, sd=1)
 [1] 1.959964
```

7.4 Chọn mẫu ngẫu nhiên (random sampling)

Trong xác suất và thống kê, lấy mẫu ngẫu nhiên rất quan trọng, vì nó đảm bảo tính hợp lí của các phương pháp phân tích và suy luận thống kê. Với R, chúng ta có thể lấy mẫu một mẫu ngẫu nhiên bằng cách sử dụng hàm `sample`.

Ví dụ 8: Chúng ta có một quần thể gồm 40 người (mã số 1, 2, 3, …, 40). Nếu chúng ta muốn chọn 5 đối tượng quần thể đó, ai sẽ là người được chọn? Chúng ta có thể dùng lệnh `sample()` để trả lời câu hỏi đó như sau:

```
> sample(1:40, 5)
 [1] 32 26  6 18  9
```

Kết quả trên cho biết đối tượng 32, 26, 8, 18 và 9 được chọn. Mỗi lần ra lệnh này, R sẽ chọn một mẫu khác, chứ không hoàn toàn giống như mẫu trên. Ví dụ:

```
> sample(1:40, 5)
[1]  5 22 35 19  4

> sample(1:40, 5)
[1] 24 26 12  6 22

> sample(1:40, 5)
[1] 22 38 11  6 18
```

v.v…

Trên đây là lệnh để chúng ta chọn mẫu ngẫu nhiên mà không thay thế (random sampling without replacement), tức là mỗi lần chọn mẫu, chúng ta không bỏ lại các mẫu đã chọn vào quần thể.

Nhưng nếu chúng ta muốn chọn mẫu thay thế (tức mỗi lần chọn ra một số đối tượng, chúng ta bỏ vào lại trong quần thể để chọn tiếp lần sau). Ví dụ, chúng ta muốn chọn 10 người từ một quần thể 50 người, bằng cách lấy mẫu với thay thế (random sampling with replacement), chúng ta chỉ cần thêm tham số `replace = TRUE`:

```
> sample(1:50, 10, replace=T)
```

```
 [1] 31 44  6  8 47 50 10 16 29 23
```

Hay ném một đồng xu 10 lần; mỗi lần, dĩ nhiên đồng xu có 2 kết quả H và T; và kết quả 10 lần có thể là:

```
> sample(c("H", "T"), 10, replace=T)
 [1] "H" "T" "H" "H" "H" "T" "H" "H" "T" "T"
```

Cũng có thể tưởng tượng chúng ta có 5 quả banh màu xanh (X) và 5 quả banh màu đỏ (D) trong một bao. Nếu chúng ta chọn 1 quả banh, ghi nhận màu, rồi để lại vào bao; rồi lại chọn 1 quả banh khác, ghi nhận màu, và bỏ vào bao lại. Cứ như thế, chúng ta chọn 20 lần, kết quả có thể là:

```
> sample(c("X", "D"), 20, replace=T)
 [1] "X" "D" "D" "D" "D" "D" "X" "X" "X" "X" "X" "D" "X" "X" "D" "X" "X" "X" "X"
[20] "D"
```

Ngoài ra, chúng ta còn có thể lấy mẫu với một xác suất cho trước. Trong hàm sau đây, chúng ta chọn 10 đối tượng từ dãy số 1 đến 5, nhưng xác suất không bằng nhau:

```
> sample(5, 10, prob=c(0.3, 0.4, 0.1, 0.1, 0.1), replace=T)
[1] 3 1 3 2 2 2 2 2 5 1
```

Đối tượng 1 được chọn 2 lần, đối tượng 2 được chọn 5 lần, đối tượng 3 được chọn 2 lần, v.v… Tuy không hoàn toàn phù hợp với xác suất 0.3, 0.4, 0.1 như cung cấp vì số mẫu còn nhỏ, nhưng cũng không quá xa với kì vọng.

8. Biểu đồ

Trong ngôn ngữ R có rất nhiều cách để thiết kế một biểu đồ gọn và đẹp. Phần lớn những hàm để thiết kế biểu đồ có sẵn trong R, nhưng một số loại biểu đồ tinh vi và phức tạp khác có thể thiết kế bằng các package chuyên dụng như `lattice` hay `trellis` có thể tải từ website của R. Trong chương này tôi sẽ chỉ cách vẽ các biểu đồ thông dụng bằng cách sử dụng các hàm phổ biến trong R.

8.1 Số liệu cho phân tích biểu đồ

Sau khi đã biết qua môi trường và những lựa chọn để thiết kế một biểu đồ, bây giờ chúng ta có thể sử dụng một số hàm thông dụng để vẽ các biểu đồ cho số liệu. Theo tôi, biểu đồ có thể chia thành 2 loại chính: biểu đồ dùng để mô tả một biến số và biểu đồ về mối liên hệ giữa hai hay nhiều biến số. Tất nhiên, biến số có thể là liên tục hay không liên tục, cho nên, trong thực tế, chúng ta có 4 loại biểu đồ. Trong phần sau đây, tôi sẽ điểm qua các loại biểu đồ, từ đơn giản đến phức tạp.

Có lẽ cách tốt nhất để tìm hiểu cách vẽ đồ thị bằng R là bằng một dữ liệu thực tế. Tôi sẽ quay lại **ví dụ 2** (phần 4.2). Trong ví dụ đó, chúng ta có dữ liệu gồm 8 cột (hay

biến số): `id, sex, age, bmi, hdl, ldl, tc,` và `tg.` (Chú ý, `id` là mã số của 50 đối tượng nghiên cứu; `sex` là giới tính (nam hay nữ); `age` là độ tuổi; `bmi` là tỉ số trọng lương; `hdl` là high density cholesterol; `ldl` là low density cholesterol; `tc` là tổng số - total – cholesterol; và `tg` triglycerides). Dữ liệu được chứa trong directory directory `c:\works\insulin` dưới tên `chol.txt`. Trước khi vẽ đồ thị, chúng ta bắt đầu bằng cách nhập dữ liệu này vào R.

```
> setwd("c:/works/stats")
> cong <- read.table("chol.txt", header=TRUE, na.strings=".")
> attach(cong)
```

Hay để tiện việc theo dõi tôi sẽ nhập các dữ liệu đó bằng các lệnh sau đây:

```
sex <- c("Nam", "Nu", "Nu","Nam","Nam", "Nu","Nam","Nam","Nam", "Nu",
         "Nu","Nam", "Nu","Nam","Nam", "Nu", "Nu", "Nu", "Nu", "Nu",
         "Nu", "Nu", "Nu", "Nu","Nam","Nam", "Nu","Nam", "Nu", "Nu",
         "Nu","Nam","Nam", "Nu", "Nu","Nam", "Nu","Nam", "Nu", "Nu",
         "Nam", "Nu","Nam","Nam","Nam", "Nu","Nam","Nam", "Nu", "Nu")

age <- c(57, 64, 60, 65, 47, 65, 76, 61, 59, 57,
         63, 51, 60, 42, 64, 49, 44, 45, 80, 48,
         61, 45, 70, 51, 63, 54, 57, 70, 47, 60,
         60, 50, 60, 55, 74, 48, 46, 49, 69, 72,
         51, 58, 60, 45, 63, 52, 64, 45, 64, 62)

bmi <- c( 17, 18, 18, 18, 18, 18, 19, 19, 19, 19, 20, 20, 20, 20, 20,
          20, 21, 21, 21, 21, 21, 21, 21, 21, 22, 22, 22, 22, 22, 22,
          22, 22, 22, 22, 23, 23, 23, 23, 23, 23, 23, 23, 24, 24, 24,
          24, 24, 24, 25, 25)

hdl <- c(5.000,4.380,3.360,5.920,6.250,4.150,0.737,7.170,6.942,5.000,
         4.217,4.823,3.750,1.904,6.900,0.633,5.530,6.625,5.960,3.800,
         5.375,3.360,5.000,2.608,4.130,5.000,6.235,3.600,5.625,5.360,
         6.580,7.545,6.440,6.170,5.270,3.220,5.400,6.300,9.110,7.750,
         6.200,7.050,6.300,5.450,5.000,3.360,7.170,7.880,7.360,7.750)

ldl <- c(2.0,  3.0,  3.0,  4.0,  2.1,  3.0,  3.0,  3.0,  3.0, 2.0,
         5.0,  1.3,  1.2,  0.7,  4.0,  4.1,  4.3,  4.0,  4.3, 4.0,
         3.1,  3.0,  1.7,  2.0,  2.1,  4.0,  4.1,  4.0,  4.2, 4.2,
         4.4,  4.3,  2.3,  6.0,  3.0,  3.0,  2.6,  4.4,  4.3, 4.0,
         3.0,  4.1,  4.4,  2.8,  3.0,  2.0,  1.0,  4.0,  4.6,  4.0)

tc <-c (4.0, 3.5, 4.7, 7.7, 5.0, 4.2, 5.9, 6.1, 5.9, 4.0,
        6.2, 4.1, 3.0, 4.0, 6.9, 5.7, 5.7, 5.3, 7.1, 3.8,
        4.3, 4.8, 4.0, 3.0, 3.1, 5.3, 5.3, 5.4, 4.5, 5.9,
        5.6, 8.3, 5.8, 7.6, 5.8, 3.1, 5.4, 6.3, 8.2, 6.2,
        6.2, 6.7, 6.3, 6.0, 4.0, 3.7, 6.1, 6.7, 8.1, 6.2)

tg <- c(1.1, 2.1, 0.8, 1.1, 2.1, 1.5, 2.6, 1.5, 5.4, 1.9,
        1.7, 1.0, 1.6, 1.1, 1.5, 1.0, 2.7, 3.9, 3.0, 3.1,
        2.2, 2.7, 1.1, 0.7, 1.0, 1.7, 2.9, 2.5, 6.2, 1.3,
        3.3, 3.0, 1.0, 1.4, 2.5, 0.7, 2.4, 2.4, 1.4, 2.7,
        2.4, 3.3, 2.0, 2.6, 1.8, 1.2, 1.9, 3.3, 4.0, 2.5)

cong <- data.frame(sex, age, bmi, hdl, ldl, tc, tg)
```

8.2 Biểu đồ cho một biến số rời rạc (discrete variable): `barplot`

Biến `sex` trong dữ liệu trên có hai giá trị (`nam` và `nu`), tức là một biến không liên tục. Chúng ta muốn biết tần số của giới tính (bao nhiêu nam và bao nhiêu nữ) và vẽ một biểu đồ đơn giản. Để thực hiện ý định này, trước hết, chúng ta cần dùng hàm table để biết tần số:

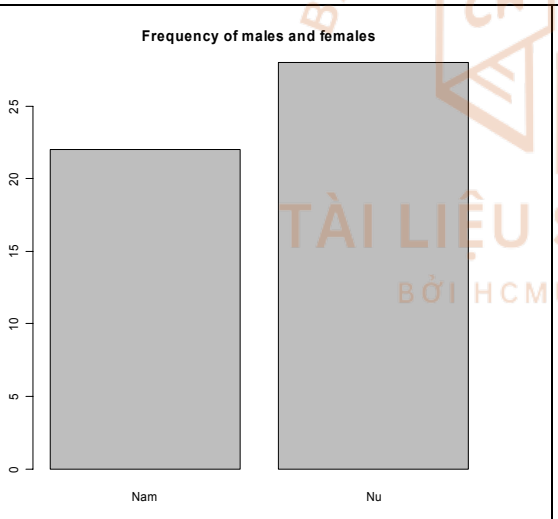
```
> sex.freq <- table(sex)
> sex.freq
sex
Nam  Nu
 22  28
```

Có 22 nam và 28 nữa trong nghiên cứu. Sau đó dùng hàm `barplot` để thể hiện tần số này như sau:

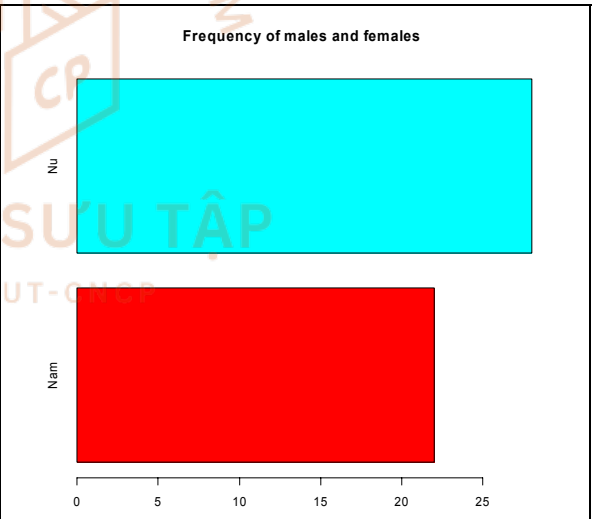
```
> barplot(sex.freq, main="Frequency of males and females")
```

Biểu trên cũng có thể có được bằng một lệnh đơn giản hơn (**Biểu đồ 8a**):

```
> barplot(table(sex), main="Frequency of males and females")
```



Biểu đồ 8a. Tần số giới tính thể hiện bằng cột số.



Biểu đồ 8b. Tần số giới tính thể hiện bằng dòng số.

Thay vì thể hiện tần số nam và nữ bằng 2 cột, chúng ta có thể thể hiện bằng hai dòng bằng thông số `horiz = TRUE`, như sau (xem kết quả trong **Biểu đồ 6b**):

```
> barplot(sex.freq,
        horiz = TRUE,
        col = rainbow(length(sex.freq)),
        main="Frequency of males and females")
```

8.3 Biểu đồ cho hai biến số rời rạc (discrete variable): `barplot`

Age là một biến số liên tục. Chúng ta có thể chia bệnh nhân thành nhiều nhóm dựa vào độ tuổi. Hàm `cut` có chức năng "cắt" một biến liên tục thành nhiều nhóm rời rạc. Chẳng hạn như:

```
> ageg <- cut(age, 3)
> table(ageg)
ageg
  (42,54.7] (54.7,67.3]   (67.3,80]
         19          24           7
```

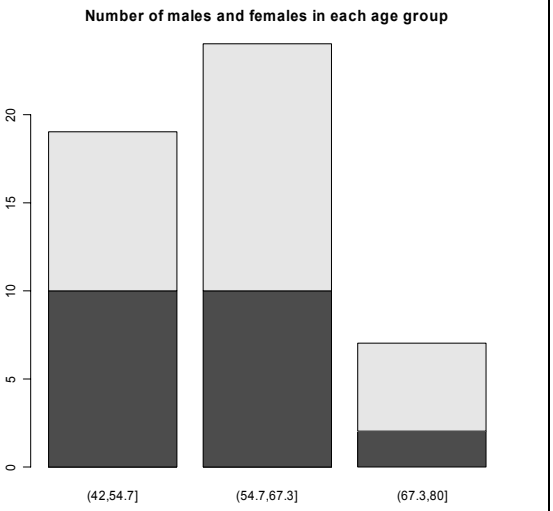
Có hiệu quả chia biến `age` thành 3 nhóm. Tần số của ba nhóm này là: 42 tuổi đến 54.7 tuổi thành nhóm 1, 54.7 đến 67.3 thành nhóm 2, và 67.3 đến 80 tuổi thành nhóm 3. Nhóm 1 có 19 bệnh nhân, nhóm 2 và 3 có 24 và 7 bệnh nhân.

Bây giờ chúng ta muốn biết có bao nhiêu bệnh nhân trong từng độ tuổi và từng giới tính bằng lệnh `table`:

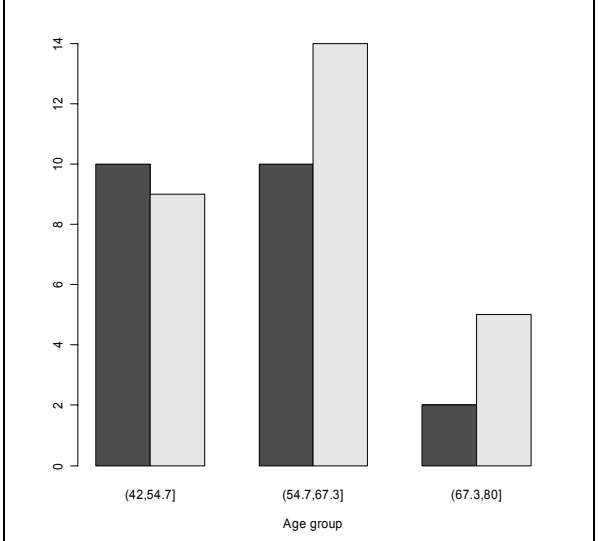
```
> age.sex <- table(sex, ageg)
> age.sex
     ageg
sex   (42,54.7] (54.7,67.3] (67.3,80]
  Nam        10          10         2
  Nu          9          14         5
```

Kết quả trên cho thấy chúng ta có 10 bệnh nhân nam và 9 nữ trong nhóm tuổi thứ nhất, 10 nam và 14 nữa trong nhóm tuổi thứ hai, v.v… Để thể hiện tần số của hai biến này, chúng ta vẫn dùng `barplot`:

```
> barplot(age.sex, main="Number of males and females in each age
   group")
```



Biểu đồ 9a. Tần số giới tính và nhóm tuổi thể hiện bằng cột số.



Biểu đồ 9b. Tần số giới tính và nhóm tuổi thể hiện bằng hai dòng số.

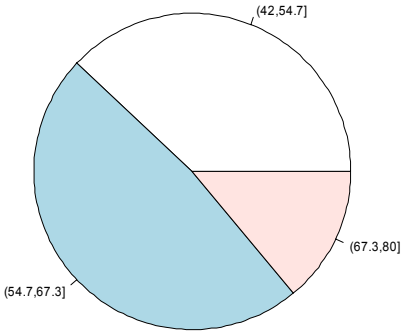
Trong **Biểu đồ 9a**, mỗi cột là cho một độ tuổi, và phần đậm của cột là nữ, và phần màu nhạt là tần số của nam giới. Thay vì thể hiện tần số nam nữ trong một cột, chúng ta cũng có thể thể hiện bằng 2 cột với `beside=T` như sau (**Biểu đồ 9b**):

```
barplot(age.sex, beside=TRUE, xlab="Age group")
```

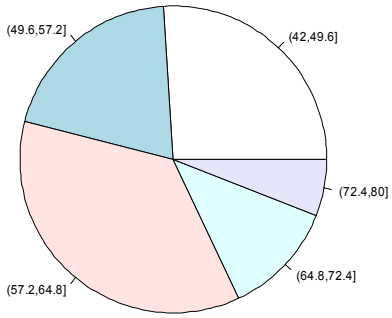
8.4 Biểu đồ hình tròn

Tần số một biến rời rạc cũng có thể thể hiện bằng biểu đồ hình tròn. Ví dụ sau đây vẽ biểu đồ tần số của độ tuổi. **Biểu đồ 10a** là 3 nhóm độ tuổi, và **Biểu đồ 10b** là biểu đồ tần số cho 5 nhóm tuổi:

```
> pie(table(ageg))
pie(table(cut(age,5)))
```



Biểu đồ 10a. Tần số cho 3 nhóm tuổi



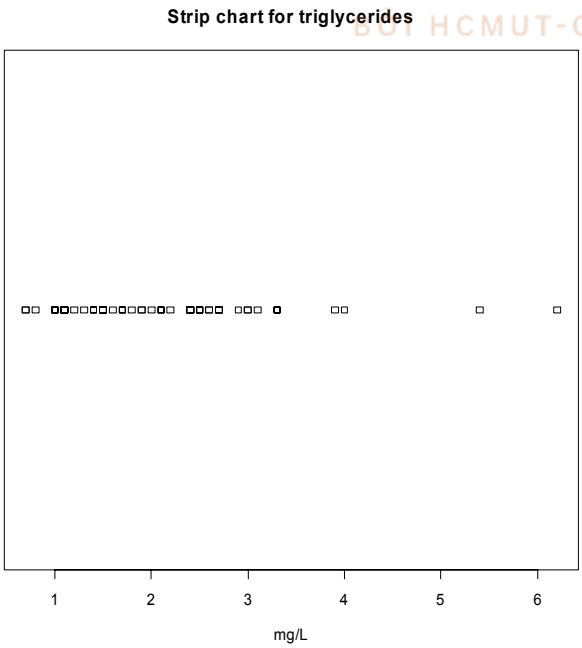
Biểu đồ 10b. Tần số cho 5 nhóm tuổi

8.5 Biểu đồ cho một biến số liên tục: `stripchart` và `hist`

8.5.1 Stripchart

Biểu đồ strip cho chúng ta thấy tính liên tục của một biến số. Chẳng hạn như chúng ta muốn tìm hiểu tính liên tục của triglyceride (`tg`), hàm `stripchart()` sẽ giúp trong mục tiêu này:

```
> stripchart(tg,
             main="Strip chart for triglycerides", xlab="mg/L")
```



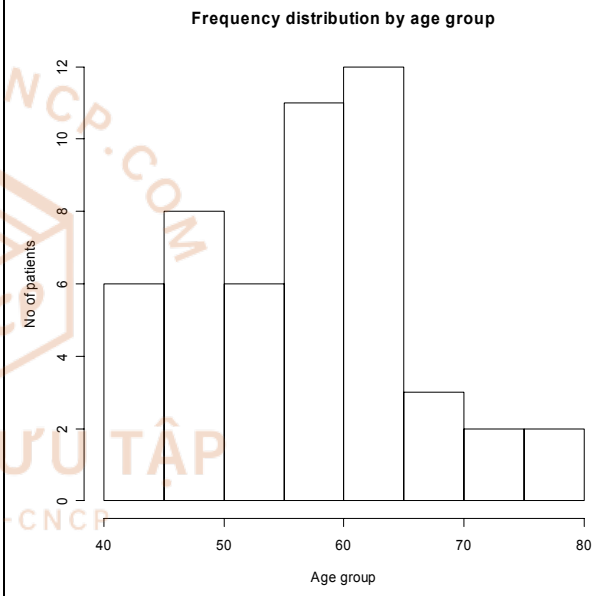
Chúng ta thấy biến số `tg` có sự bất liên tục, nhất là các đối tượng có `tg` cao. Trong khi phần lớn đối tượng có độ `tg` thấp hơn 5, thì có 2 đối tượng với `tg` rất cao (>5).

8.5.2 Histogram

Age là một biến số liên tục. Để vẽ biểu đồ tần số của biến số `age`, chúng ta chỉ đơn giản lệnh `hist(age)`. Như đã đề cập trên, chúng ta có thể cải tiến đồ thị này bằng cách cho thêm tựa đề chính (`main`) và tựa đề của trục hoành (`xlab`) và trục tung (`ylab`):

```
> hist(age)
> hist(age, main="Frequency distribution by age group", xlab="Age
group", ylab="No of patients")
```

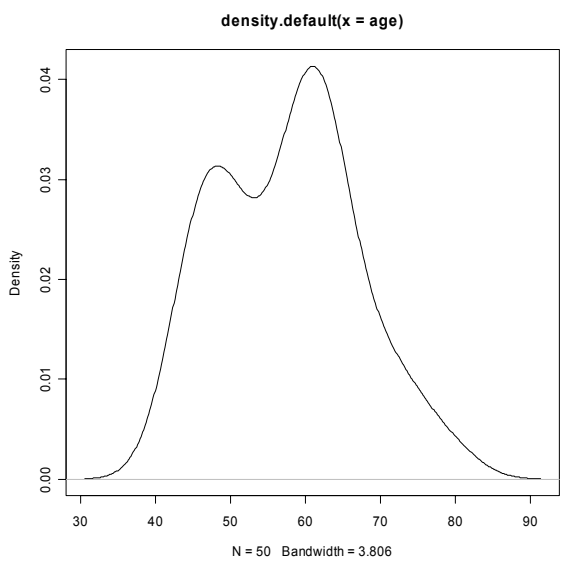
Biểu đồ 11a. Trục tung là số bệnh nhân (đối tượng nghiên cứu) và trục hoành là độ tuổi. Chẳng hạn như tuổi 40 đến 45 có 6 bệnh nhân, từ 70 đến 80 tuổi có 4 bệnh nhân.



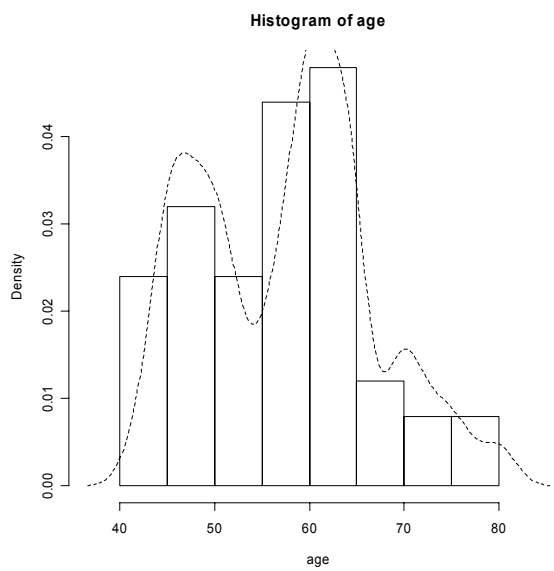
Biểu đồ 11b. Thêm tên biểu đồ và tên của trục trung và trục hoành bằng `xlab` và `ylab`.

Chúng ta cũng có thể biến đổi biểu đồ thành một đồ thị phân phối xác suất bằng hàm `plot(density)` như sau (kết quả trong **Biểu đồ 12a**):

```
> plot(density(age),add=TRUE)
```



Biểu đồ 12a. Xác suất phân phối mật độ cho biến `age` (độ tuổi).



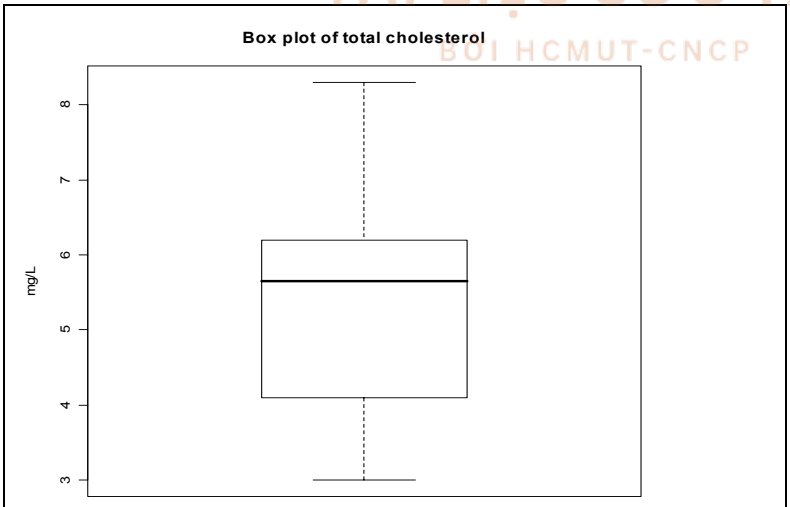
Biểu đồ 12b. Xác suất phân phối mật độ cho biến `age` (độ tuổi) với nhiều interquartile.

Chúng ta có thể vẽ hai đồ thị chồng lên bằng cách dùng hàm interquartile như sau (kết quả xem **Biểu đồ 12b**):

8.6 Biểu đồ hộp (`boxplot`)

Để vẽ biểu đồ hộp của biến số `tc`, chúng ta chỉ đơn giản lệnh:

```
> boxplot(tc, main="Box plot of total cholesterol", ylab="mg/L")
```



Biểu đồ 13. Trong biểu đồ này, chúng ta thấy median (trung vị) khoảng 5.6 mg/L, 25% total cholesterol thấp hơn 4.1, và 75% thấp hơn 6.2. Total cholesterol thấp nhất

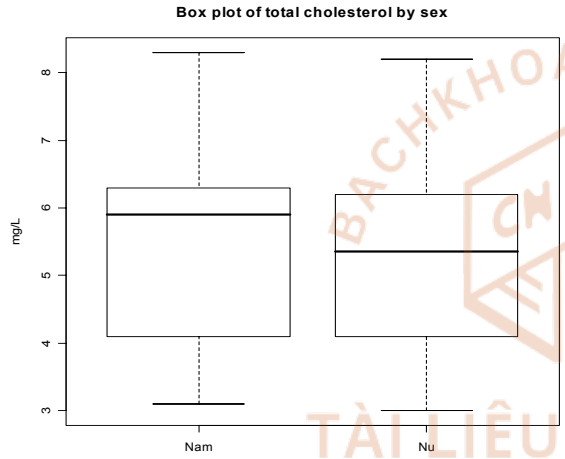
là khoang 3, và cao nhất là trên 8 mg/L.

Trong biểu đồ sau đây, chúng ta so sánh `tc` giữa hai nhóm nam và nữ:

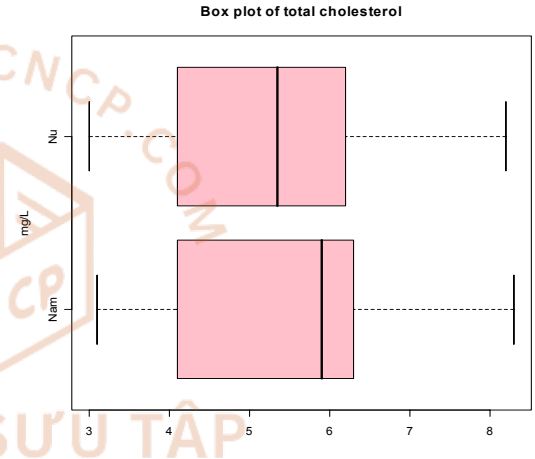
```
> boxplot(tc ~ sex, main="Box plot of total cholestrol by sex",
ylab="mg/L")
```

Kết quả trình bày trong **Biểu đồ 14a**. Chúng ta có thể biến đổ giao diện của đồ thị bằng cách dùng thông số `horizontal=TRUE` và thay đổi màu bằng thông số `col` như sau (**Biểu đồ 14b)**:

```
> boxplot(tc~sex, horizontal=TRUE, main="Box plot of total
cholesterol", ylab="mg/L", col = "pink")
```



Biểu đồ 14a. Trong biểu đồ này, chúng ta thấy trung vị của total cholesterol ở nữ giới thấp hơn nam giới, nhưng độ dao động giữa hai nhóm không khác nhau bao nhiêu.



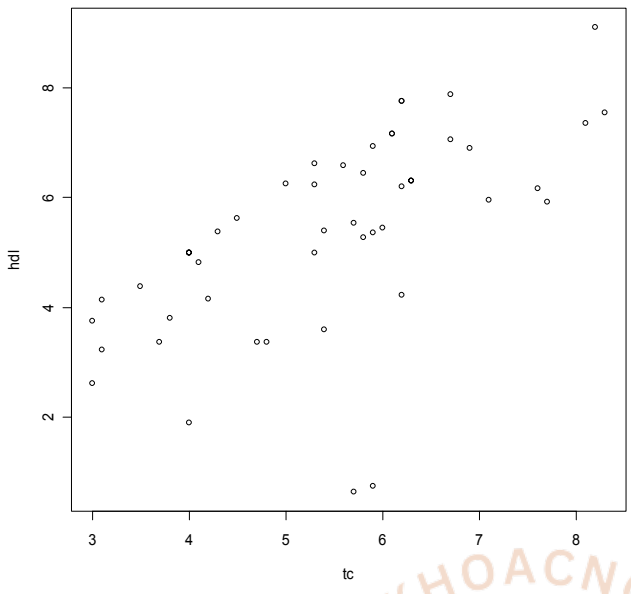
Biểu đồ 14b. Total cholesterol cho từng giới tính, với màu sắc và hình hộp nằm ngang.

8.7 Phân tích biểu đồ cho hai biến liên tục

8.7.1 Biểu đồ tán xạ (scatter plot)

Để tìm hiểu mối liên hệ giữa hai biến, chúng ta dùng biểu đồ tán xạ. Để vẽ biểu đồ tán xạ về mối liên hệ giữa biến số `tc` và `hdl`, chúng ta sử dụng hàm `plot`. Thông số thứ nhất của hàm plot là trục hoành (x-axis) và thông số thứ 2 là trục tung. Để tìm hiểu mối liên hệ giữa `tc` và `hdl` chúng ta đơn giản lệnh:

```
> plot(tc, hdl)
```



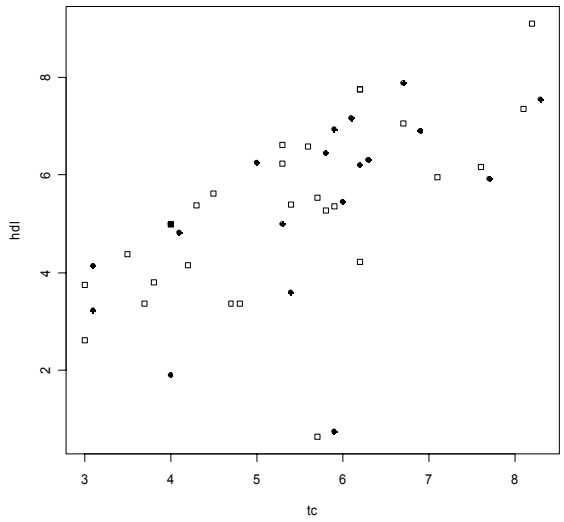
Biểu đồ 15. Mối liên hệ giữa `tc` và `hdl`. Trong biểu đồ này, chúng ta vẽ biến số `hdl` trên trục tung và `tc` trên trục hoành.

Chúng ta muốn phân biệt giới tính (nam và nữ) trong biểu đồ trên. Để vẽ biểu đồ đó, chúng ta phải dùng đến hàm `ifelse`. Trong lệnh sau đây, nếu `sex=="Nam"` thì vẽ kí tự số 16 (ô tròn), nếu không nam thì vẽ kí tự số 22 (tức ô vuông):

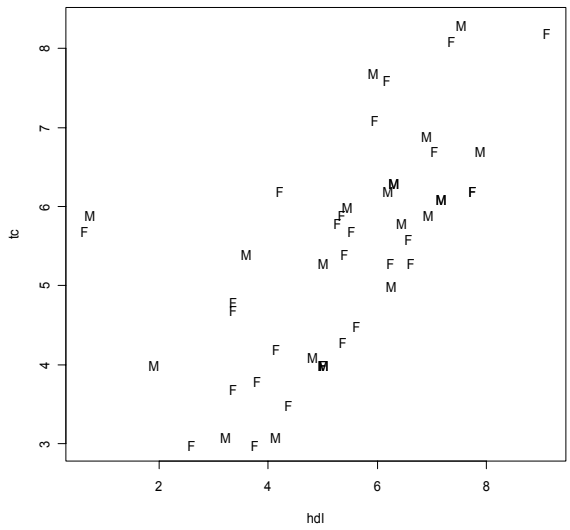
```
> plot(hdl, tc, pch=ifelse(sex=="Nam", 16, 22))
```

Kết quả là **Biểu đồ 16a**. Chúng ta cũng có thể thay kí tư thành "M" (nam) và "F" nữ(xem **Biểu đồ 16b**):

```
> plot(hdl, tc, pch=ifelse(sex=="Nam", "M", "F"))
```



Biểu đồ 16a. Mối liên hệ giữa tc và hdl theo từng giới tính được thể hiện bằng hai kí hiệu dấu.



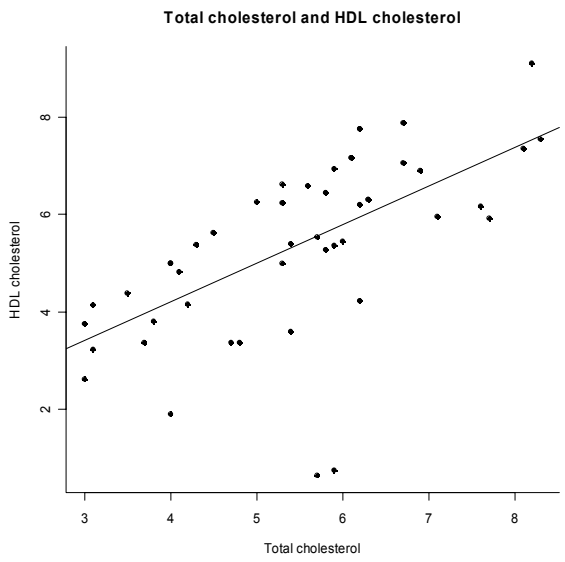
Biểu đồ 16a. Mối liên hệ giữa tc và hdl theo từng giới tính được thể hiện bằng hai kí tự.

Chúng ta cũng có thể vẽ một đường biểu diễn hồi qui tuyến tính (regression line) qua các điểm trên bằng cách tiếp tục ra các lệnh sau đây:

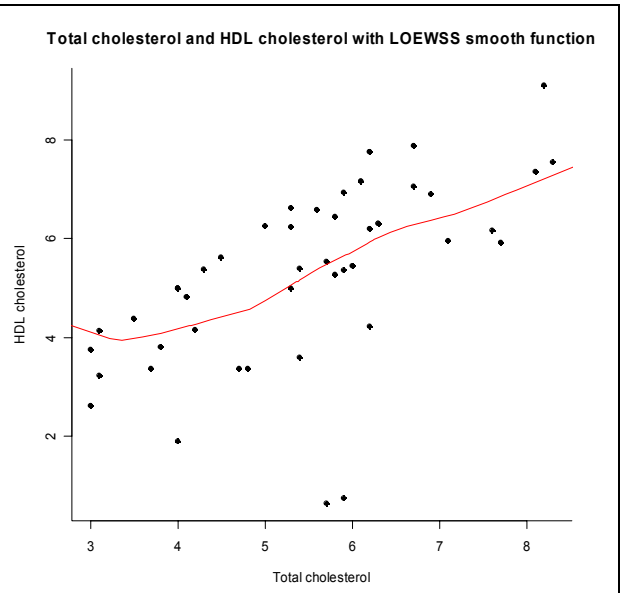
```
> plot(hdl ~ tc, pch=16, main="Total cholesterol and HDL cholesterol",
xlab="Total cholesterol", ylab="HDL cholesterol", bty="l")
> reg <- lm(hdl ~ tc)
> abline(reg)
```

Kết quả là **Biểu đồ 17a** dưới đây. Chúng ta cũng có thể dùng hàm trơn (smooth function) để biểu diễn mối liên hệ giữa hai biến số. Đồ thị sau đây sử dụng `lowess` (một hàm thông thường nhất) trong việc "làm trơn" số liệu `tc` và `hdl` **(Biểu đồ 17b).**

```
> plot(hdl ~ tc, pch=16,
       main="Total cholesterol and HDL cholesterol with LOEWSS smooth
function",
       xlab="Total cholesterol", ylab="HDL cholesterol", bty="l")

> lines(lowess(hdl, tc, f=2/3, iter=3), col="red")
```



Biểu đồ 17a. Trong lệnh trên, `reg<-lm(hdl~tc)` có nghĩa là tìm phương trình liên hệ giữa `hdl` và `tc` bằng linear model (`lm`) và đ8ạt kết quả vào đối tượng `reg`. Lệnh thứ hai `abline(reg)` yêu cầu R vẽ đường thẳng từ phương trình trong `reg`



Biểu đồ 17b. Thay vì dùng abline, chúng ta dùng hàm lowess để thể hiện mối liên hệ giữa `tc` và `hdl`.

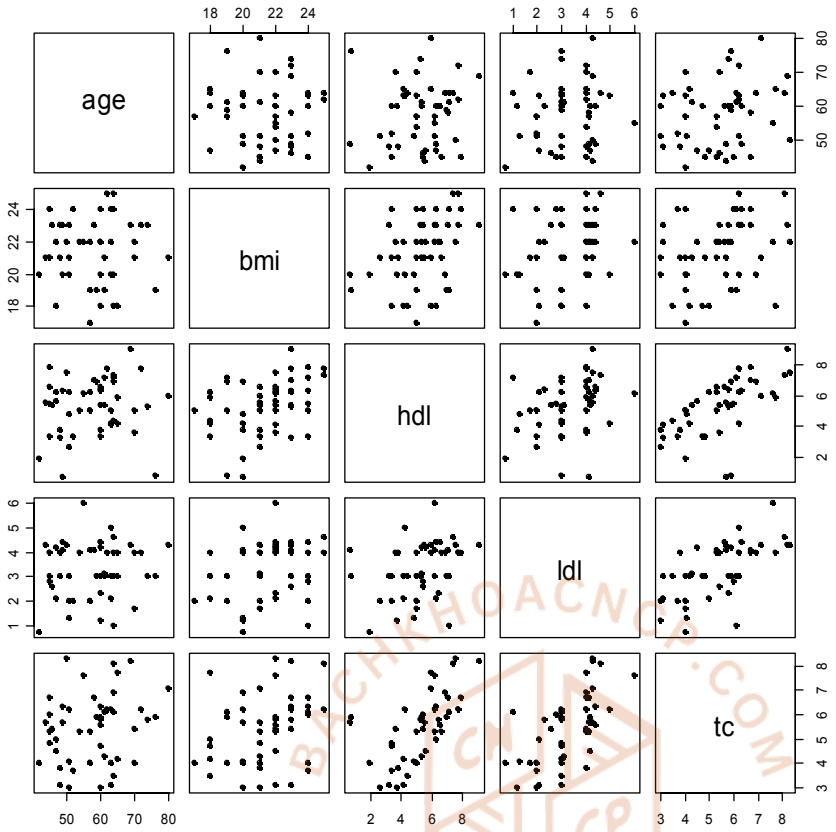
Bạn đọc có thể thí nghiệm với nhiều thông số `f=1/2, f=2/5`, hay thậm chí `f=1/10` sẽ thấy đồ thị biến đổi một cách "thú vị".

8.8 Phân tích Biểu đồ cho nhiều biến: `pairs`

Chúng ta có thể tìm hiểu mối liên hệ giữa các biến số như `age, bmi, hdl, ldl` và `tc` bằng cách dùng lệnh `pairs`. Nhưng trước hết, chúng ta phải đưa các biến số này vào một `data.frame` chỉ gồm những biến số có thể vẽ được, và sau đó sử dụng hàm `pairs` trong R.

```
> lipid <- data.frame(age,bmi,hdl,ldl,tc)
> pairs(lipid, pch=16)
```

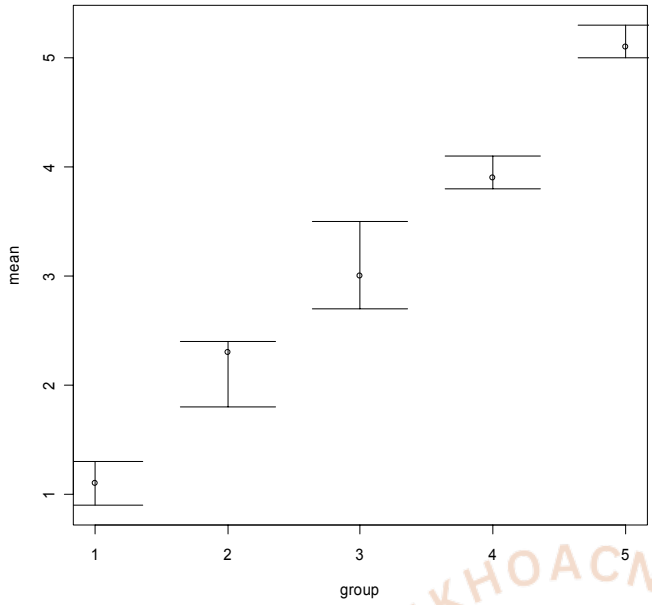
Kết quả sẽ là:



8.9 Biểu đồ với sai số chuẩn (standard error)

Trong biểu đồ sau đây, chúng ta có 5 nhóm (biến số `x` được mô phỏng chứ không phải số liệu thật), và mỗi nhóm có giá trị trung bình `mean`, và độ tin cậy 95% (`lcl` và `ucl`). Thông thường `lcl=mean-1.96*SE` và `ucl = mean+1.96*SE` (SE là sai số chuẩn). Chúng ta muốn vẽ biểu đồ cho 5 nhóm với sai số chuẩn đó. Các lệnh và hàm sau đây sẽ cần thiết:

```
> group <- c(1,2,3,4,5)
> mean <- c(1.1, 2.3, 3.0, 3.9, 5.1)
> lcl <- c(0.9, 1.8, 2.7, 3.8, 5.0)
> ucl <- c(1.3, 2.4, 3.5, 4.1, 5.3)
> plot(group, mean, ylim=range(c(lcl, ucl)))
> arrows(group, ucl, group, lcl, length=0.5, angle=90, code=3)
```



9. Phân tích thống kê mô tả

9.1 Thống kê mô tả (descriptive statistics, `summary`)

Để minh họa cho việc áp dụng R vào thống kê mô tả, tôi sẽ sử dụng một dữ liệu nghiên cứu có tên là `igfdata`. Trong nghiên cứu này, ngoài các chỉ số liên quan đến giới tính, độ tuổi, trọng lượng và chiều cao, chúng tôi đo lường các hormone liên quan đến tình trạng tăng trưởng như `igfi, igfbp3, als,` và các markers liên quan đến sự chuyển hóa của xương `pinp, ictp` và `pinp`. Có 100 đối tượng nghiên cứu. Dữ liệu này được chứa trong directory `c:\works\stats`. Trước hết, chúng ta cần phải nhập dữ liệu vào R với những lệnh sau đây (các câu chữ theo sau dấu # là những chú thích để bạn đọc theo dõi):

```
> options(width=100)
# chuyển directory
> setwd("c:/works/stats")

# đọc dữ liệu vào R
> igfdata <- read.table("igf.txt", header=TRUE, na.strings=".")
> attach(igfdata)

# xem xét các cột số trong dữ liệu
> names(igfdata)
 [1] "id"        "sex"       "age"       "weight"    "height"    "ethnicity"
 [7] "igfi"      "igfbp3"    "als"       "pinp"      "ictp"      "p3np"

> igfdata
     id    sex age weight height ethnicity    igfi  igfbp3     als    pinp    ictp    p3np
```

```
1      1 Female  15     42    162     Asian 189.000 4.00000 323.667 353.970 11.2867  8.3367
2      2   Male  16     44    160 Caucasian 160.000 3.75000 333.750 375.885 10.4300  6.7450
3      3 Female  15     43    157     Asian 146.833 3.43333 248.333 199.507  8.3633 12.5000
4      4 Female  15     42    155     Asian 185.500 3.40000 251.000 483.607 13.3300 14.2767
5      5 Female  16     47    167     Asian 192.333 4.23333 322.000 105.430  7.9233  4.5033
6      6 Female  25     45    160     Asian 110.000 3.50000 284.667  76.487  4.9833  4.9367
7      7 Female  19     45    161     Asian 157.000 3.20000 274.000  75.880  6.3500  5.3200
8      8 Female  18     43    153     Asian 146.000 3.40000 303.000  86.360  7.3700  4.6700
9      9 Female  15     41    149     Asian 197.667 3.56667 308.500 254.803 11.8700  6.8200
10    10 Female  24     45    157   African 148.000 3.40000 273.000  44.720  3.7400  6.1600
...
...
97    97 Female  17     54    168 Caucasian 204.667 4.96667 441.333  64.130  5.1600  4.4367
98    98   Male  18     55    169     Asian 178.667 3.86667 273.000 185.913  7.5267  8.8333
99    99 Female  18     48    151     Asian 237.000 3.46667 324.333 105.127  5.9867  5.6600
100  100   Male  15     54    168     Asian 130.000 2.70000 259.333 325.840 10.2767  6.5933
```

Trên đây chỉ là một phần số liệu trong số 100 đối tượng.

Cho một biến số $x_1, x_2, x_3, ..., x_n$ chúng ta có thể tính toán một số chỉ số thống kê mô tả như sau:

Lí thuyết	Hàm R
Số trung bình: $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$.	`mean(x)`
Phương sai: $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$	`var(x)`
Độ lệch chuẩn: $s = \sqrt{s^2}$	`sd(x)`
Sai số chuẩn (standard error): $SE = \frac{s}{\sqrt{n}}$	Không có
Trị số thấp nhất	`min(x)`
Trị số cao nhất	`max(x)`
Toàn cự (range)	`range(x)`

Ví dụ 9: Để tìm giá trị trung bình của độ tuổi, chúng ta chỉ đơn giản lệnh:

```
> mean(age)
[1] 19.17
```

Hay phương sai và độc lệch chuẩn của tuổi:

```
> var(age)
[1] 15.33444

> sd(age)
[1] 3.915922
```

Tuy nhiên, R có lệnh `summary` có thể cho chúng ta tất cả thông tin thống kê về một biến số:

```
> summary(age)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  13.00   16.00   19.00   19.17   21.25   34.00
```

Nói chung, kết quả này đơn giản và các viết tắt cũng có thể dễ hiểu. Chú ý, trong kết quả trên, có hai chỉ số "`1st Qu`" và "`3rd Qu`" có nghĩa là first quartile (tương đương với vị trí 25%) và third quartile (tương đương với vị trí 75%) của một biến số. First quartile = 16 có nghĩa là 25% đối tượng nghiên cứu có độ tuổi bằng hoặc nhỏ hơn 16 tuổi. Tương tự, Third quartile = 34 có nghĩa là 75% đối tượng có độ tuổi bằng hoặc thấp hơn 34 tuổi. Tất nhiên số trung vị (median) 19 cũng có nghĩa là 50% đối tượng có độ tuổi 19 trở xuống (hay 19 tuổi trở lên).

R không có hàm tính sai số chuẩn, và trong hàm summary, R cũng không cung cấp độ lệch chuẩn. Để có các số này, chúng ta có thể tự viết một hàm đơn giản (hãy gọi là `desc`) như sau:

```
desc <- function(x)
  {
    av <- mean(x)
    sd <- sd(x)
    se <- sd/sqrt(length(x))
    c(MEAN=av, SD=sd, SE=se)
   }
```


Và có thể gọi hàm này để tính bất cứ biến nào chúng ta muốn, như tính biến `als` sau đây:

```
> desc(als)
      MEAN         SD         SE
301.841120  58.987189   5.898719
```

Để có một "quang cảnh" chung về dữ liệu `igfdata` chúng ta chỉ đơn giản lệnh `summary` như sau:

```
> summary(igfdata)
       id              sex          age             weight          height          ethnicity
 Min.   :  1.00   Female:69   Min.   :13.00   Min.   :41.00   Min.   :149.0   African  : 8
 1st Qu.: 25.75   Male  :31   1st Qu.:16.00   1st Qu.:47.00   1st Qu.:157.0   Asian    :60
 Median : 50.50               Median :19.00   Median :50.00   Median :162.0   Caucasian:30
 Mean   : 50.50               Mean   :19.17   Mean   :49.91   Mean   :163.1   Others   : 2
 3rd Qu.: 75.25               3rd Qu.:21.25   3rd Qu.:53.00   3rd Qu.:168.0
 Max.   :100.00               Max.   :34.00   Max.   :60.00   Max.   :196.0

      igfi            igfbp3           als             pinp            ictp
```

```
 Min.   : 85.71   Min.   :2.000   Min.   :192.7   Min.   : 26.74   Min.   : 2.697
 1st Qu.:137.17   1st Qu.:3.292   1st Qu.:256.8   1st Qu.: 68.10   1st Qu.: 4.878
 Median :161.50   Median :3.550   Median :292.5   Median :103.26   Median : 6.338
 Mean   :165.59   Mean   :3.617   Mean   :301.8   Mean   :167.17   Mean   : 7.420
 3rd Qu.:186.46   3rd Qu.:3.875   3rd Qu.:331.2   3rd Qu.:196.45   3rd Qu.: 8.423
 Max.   :427.00   Max.   :5.233   Max.   :471.7   Max.   :742.68   Max.   :21.237

      p3np
 Min.   : 2.343
 1st Qu.: 4.433
 Median : 5.445
 Mean   : 6.341
 3rd Qu.: 7.150
 Max.   :16.303
```

R tính toán tất cả các biến số nào có thể tính toán được! Thành ra, ngay cả cột `id` (tức mã số của đối tượng nghiên cứu) R cũng tính luôn! (và chúng ta biết kết quả của cột `id` chẳng có ý nghĩa thống kê gì). Đối với các biến số mang tính phân loại như `sex` và `ethnicity` (sắc tộc) thì R chỉ báo cáo tần số cho mỗi nhóm.

Kết quả trên cho tất cả đối tượng nghiên cứu. Nếu chúng ta muốn kết quả cho từng nhóm nam và nữ riêng biệt, hàm `by` trong R rất hữu dụng. Trong lệnh sau đây, chúng ta yêu cầu R tóm lược dữ liệu `igfdata` theo `sex`.

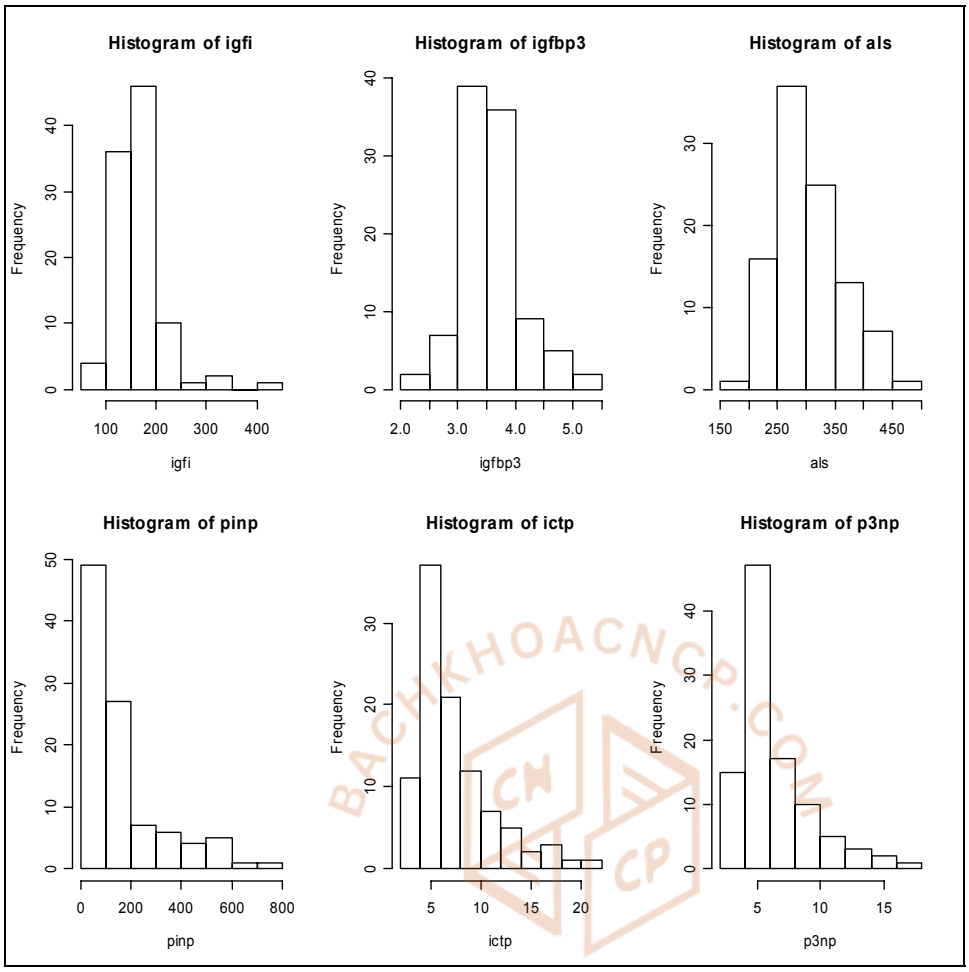
```
> by(igfdata, sex, summary)
```

```
sex: Female
       id           sex          age            weight          height
 Min.   : 1.0   Female:69   Min.   :13.00   Min.   :41.00   Min.   :149.0
 1st Qu.:21.0   Male  : 0   1st Qu.:17.00   1st Qu.:47.00   1st Qu.:156.0
 Median :47.0               Median :19.00   Median :50.00   Median :162.0
 Mean   :48.2               Mean   :19.59   Mean   :49.35   Mean   :161.9
 3rd Qu.:75.0               3rd Qu.:22.00   3rd Qu.:52.00   3rd Qu.:166.0
 Max.   :99.0               Max.   :34.00   Max.   :60.00   Max.   :196.0
     ethnicity       igfi          igfbp3          als
 African  : 4   Min.   : 85.71   Min.   :2.767   Min.   :204.3
 Asian    :43   1st Qu.:136.67   1st Qu.:3.333   1st Qu.:263.8
 Caucasian:22   Median :163.33   Median :3.567   Median :302.7
 Others   : 0   Mean   :167.97   Mean   :3.695   Mean   :311.5
                3rd Qu.:186.17   3rd Qu.:3.933   3rd Qu.:361.7
                Max.   :427.00   Max.   :5.233   Max.   :471.7
      pinp            ictp            p3np
 Min.   : 26.74   Min.   : 2.697   Min.   : 2.343
 1st Qu.: 62.75   1st Qu.: 4.717   1st Qu.: 4.337
 Median : 78.50   Median : 5.537   Median : 5.143
 Mean   :108.74   Mean   : 6.183   Mean   : 5.643
 3rd Qu.:115.26   3rd Qu.: 7.320   3rd Qu.: 6.143
 Max.   :502.05   Max.   :13.633   Max.   :14.420
------------------------------------------------------------
sex: Male
       id            sex          age            weight          height
 Min.   :  2.00   Female: 0   Min.   :14.00   Min.   :44.00   Min.   :155.0
 1st Qu.: 34.50   Male  :31   1st Qu.:15.00   1st Qu.:48.50   1st Qu.:161.5
 Median : 56.00               Median :17.00   Median :51.00   Median :164.0
 Mean   : 55.61               Mean   :18.23   Mean   :51.16   Mean   :165.6
 3rd Qu.: 75.00               3rd Qu.:20.00   3rd Qu.:53.50   3rd Qu.:169.0
 Max.   :100.00               Max.   :27.00   Max.   :59.00   Max.   :191.0
     ethnicity       igfi          igfbp3          als
```

```
 African  : 4   Min.   : 94.67   Min.   :2.000   Min.   :192.7
 Asian    :17   1st Qu.:138.67   1st Qu.:3.183   1st Qu.:249.8
 Caucasian: 8   Median :160.00   Median :3.500   Median :276.0
 Others   : 2   Mean   :160.29   Mean   :3.443   Mean   :280.2
                3rd Qu.:183.00   3rd Qu.:3.775   3rd Qu.:311.3
                Max.   :274.00   Max.   :4.500   Max.   :388.7
      pinp             ictp             p3np
 Min.   : 56.28   Min.   : 3.650   Min.   : 3.390
 1st Qu.:135.07   1st Qu.: 6.900   1st Qu.: 5.375
 Median :245.92   Median : 9.513   Median : 7.140
 Mean   :297.21   Mean   :10.173   Mean   : 7.895
 3rd Qu.:450.38   3rd Qu.:13.517   3rd Qu.:10.010
 Max.   :742.68   Max.   :21.237   Max.   :16.303
```

Để xem qua phân phối của các hormones và chỉ số sinh hóa cùng một lúc, chúng ta có thể vẽ đồ thị cho tất cả 6 biến số. Trước hết, chia màn ảnh thành 6 cửa sổ (với 2 dòng và 3 cột); sau đó lần lượt vẽ:

```
> op <- par(mfrow=c(2,3))
> hist(igfi)
> hist(igfbp3)
> hist(als)
> hist(pinp)
> hist(ictp)
> hist(p3np)
```



9.2 Thống kê mô tả theo từng nhóm

Nếu chúng ta muốn tính trung bình của một biến số như igfi cho mỗi nhóm nam và nữ giới, hàm `tapply` trong R có thể dùng cho việc này:

```
> tapply(igfi, list(sex), mean)
  Female     Male
167.9741 160.2903
```

Trong lệnh trên, `igfi` là biến số chúng ta cần tính, biến số phân nhóm là `sex`, và chỉ số thống kê chúng ta muốn là trung bình (`mean`). Qua kết quả trên, chúng ta thấy số trung bình của `igfi` cho nữ giới (167.97) cao hơn nam giới (160.29).

Nhưng nếu chúng ta muốn tính cho từng giới tính và sắc tộc, chúng ta chỉ cần thêm một biến số trong hàm `list`:

```
> tapply(igfi, list(ethnicity, sex), mean)
            Female     Male
African   145.1252 120.9168
```

```
Asian     165.6589 160.4999
Caucasian 176.6536 169.4790
Others          NA 200.5000
```

Trong kết quả trên, `NA` có nghĩa là "not available", tức không có số liệu cho phụ nữ trong các sắc tộc "others".

9.3 Kiểm định t (`t.test`)

Kiểm định t dựa vào giả thiết phân phối chuẩn. Có hai loại kiểm định t: kiểm định t cho một mẫu (one-sample t-test), và kiểm định t cho hai mẫu (two-sample t-test). Kiểm định t một mẫu nằm trả lời câu hỏi dữ liệu từ một mẫu có phải thật sự bằng một thông số nào đó hay không. Còn kiểm định t hai mẫu thì nhằm trả lời câu hỏi hai mẫu có cùng một luật phân phối, hay cụ thể hơn là hai mẫu có thật sự có cùng trị số trung bình hay không. Tôi sẽ lần lượt minh họa hai kiểm định này qua số liệu `igfdata` trên.

9.3.1 Kiểm định t một mẫu

Ví dụ 10. Qua phân tích trên, chúng ta thấy tuổi trung bình của 100 đối tượng trong nghiên cứu này là 19.17 tuổi. Chẳng hạn như trong quần thể này, trước đây chúng ta biết rằng tuổi trung bình là 30 tuổi. Vấn đề đặt ra là có phải mẫu mà chúng ta có được có đại diện cho quần thể hay không. Nói cách khác, chúng ta muốn biết giá trị trung bình 19.17 có thật sự khác với giá trị trung bình 30 hay không.

Để trả lời câu hỏi này, chúng ta sử dụng kiểm định t. Theo lí thuyết thống kê, kiểm định t được định nghĩa bằng công thức sau đây:

$$t = \frac{\bar{x} - \mu}{s / \sqrt{n}}$$

Trong đó, $\bar{x}$ là giá trị trung bình của mẫu, μ là trung bình theo giả thiết (trong trường hợp này, 30), s là độ lệch chuẩn, và n là số lượng mẫu (100). Nếu giá trị t cao hơn giá trị lí thuyết theo phân phối t ở một tiêu chuẩn có ý nghĩa như 5% chẳng hạn thì chúng ta có lí do để phát biểu khác biệt có ý nghĩa thống kê. Giá trị này cho mẫu 100 có thể tính toán bằng hàm `qt` của R như sau:

```
> qt(0.95, 100)
[1] 1.660234
```

Nhưng có một cách tính toán nhanh gọn hơn để trả lời câu hỏi trên, bằng cách dùng hàm `t.test` như sau:

```
> t.test(age, mu=30)

        One Sample t-test
```

```
data:  age
t = -27.6563, df = 99, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 30
95 percent confidence interval:
 18.39300 19.94700
sample estimates:
mean of x
    19.17
```

Trong lệnh trên `age` là biến số chúng ta cần kiểm định, và `mu=30` là giá trị giả thiết. R trình bày trị số `t = -27.66`, với 99 bậc tự do, và trị số p < 2.2e-16 (tức rất thấp). R cũng cho biết độ tin cậy 95% của age là từ 18.4 tuổi đến 19.9 tuổi (30 tuổi nằm quá ngoài khoảng tin cậy này). Nói cách khác, chúng ta có lí do để phát biểu rằng độ tuổi trung bình trong mẫu này thật sự thấp hơn độ tuổi trung bình của quần thể.

9.3.2 Kiểm định t hai mẫu

Ví dụ 11. Qua phân tích mô tả trên (phần `summary`) chúng ta thấy phụ nữ có độ hormone `igfi` cao hơn nam giới (167.97 và 160.29). Câu hỏi đặt ra là có phải thật sự đó là một khác biệt có hệ thống hay do các yếu tố ngẫu nhiên gây nên. Trả lời câu hỏi này, chúng ta cần xem xét mức độ khác biệt trung bình giữa hai nhóm và độ lệch chuẩn của độ khác biệt.

$$t = \frac{\bar{x}_2 - \bar{x}_1}{SED}$$

Trong đó $\bar{x}_1$ và $\bar{x}_2$ là số trung bình của hai nhóm nam và nữ, và *SED* là độ lệch chuẩn của $(\bar{x}_1 - \bar{x}_2)$. Thực ra, *SED* có thể ước tính bằng công thức:

$$SED = \sqrt{SE_1^2 + SE_2^2}$$

Trong đó SE_1 và SE_2 là sai số chuẩn (standard error) của hai nhóm nam và nữ. Theo lí thuyết xác suất, t tuân theo luật phân phối t với bậc tự do $n_1 + n_2 - 2$, trong đó n_1 và n_2 là số mẫu của hai nhóm. Chúng ta có thể dùng R để trả lời câu hỏi trên bằng hàm `t.test` như sau:

```
> t.test(igfi~ sex)

        Welch Two Sample t-test

data:  igfi by sex
t = 0.8412, df = 88.329, p-value = 0.4025
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -10.46855  25.83627
sample estimates:
mean in group Female   mean in group Male
      167.9741      160.2903
```

R trình bày các giá trị quan trọng trước hết:

```
t = 0.8412, df = 88.329, p-value = 0.4025
```

`df` là bậc tự do. Trị số p = 0.4025 cho thấy mức độ khác biệt giữa hai nhóm nam và nữ không có ý nghĩa thống kê (vì cao hơn 0.05 hay 5%).

```
95 percent confidence interval:
 -10.46855  25.83627
```

là khoảng tin cậy 95% về độ khác biệt giữa hai nhóm. Kết quả tính toán trên cho biết độ `igf` ở nữ giới có thể thấp hơn nam giới 10.5 ng/L hoặc cao hơn nam giới khoảng 25.8 ng/L. Vì độ khác biệt quá lớn và đó là thêm bằng chứng cho thấy không có khác biệt có ý nghĩa thống kê giữa hai nhóm.

Kiểm định trên dựa vào giả thiết hai nhóm nam và nữ có khác phương sai. Nếu chúng ta có lí do để cho rằng hai nhóm có cùng phương sai, chúng ta chỉ thay đổi một thông số trong hàm t với `var.equal=TRUE` như sau:

```
> t.test(igfi~ sex, var.equal=TRUE)

        Two Sample t-test

data:  igfi by sex
t = 0.7071, df = 98, p-value = 0.4812
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -13.88137  29.24909
sample estimates:
mean in group Female   mean in group Male
            167.9741             160.2903
```

Về mặc số, kết quả phân tích trên có khác chút ít so với kết quả phân tích dựa vào giả định hai phương sai khác nhau, nhưng trị số p cũng đi đến một kết luận rằng độ khác biệt giữa hai nhóm không có ý nghĩa thống kê.

9.4 Kiểm định Wilcoxon cho hai mẫu (`wilcox.test`)

Kiểm định t dựa vào giả thiết là phân phối của một biến phải tuân theo luật phân phối chuẩn. Nếu giả định này không đúng, kết quả của kiểm định t có thể không hợp lí (valid). Để kiểm định phân phối của `igfi`, chúng ta có thể dùng hàm `shapiro.test` như sau:

```
> shapiro.test(igfi)

        Shapiro-Wilk normality test
```

```
data:  igfi
W = 0.8528, p-value = 1.504e-08
```

Trị số p nhỏ hơn 0.05 rất nhiều, cho nên chúng ta có thể nói rằng phân phối của `igfi` không tuân theo luật phân phối chuẩn. Trong trường hợp này, việc so sánh giữa hai nhóm có thể dựa vào phương pháp phi tham số (non-parametric) có tên là kiểm định Wilcoxon, vì kiểm định này (không như kiểm định t) không tùy thuộc vào giả định phân phối chuẩn.

```
> wilcox.test(igfi ~ sex)

        Wilcoxon rank sum test with continuity correction

data:  igfi by sex
W = 1125, p-value = 0.6819
alternative hypothesis: true mu is not equal to 0
```

Trị số p = 0.682 cho thấy quả thật độ khác biệt về `igfi` giữa hai nhóm nam và nữ không có ý nghĩa thống kê. Kết luận này cũng không khác với kết quả phân tích bằng kiểm định t.

9.5 Kiểm định t cho các biến số theo cặp (paired t-test, `t.test`)

Kiểm định t vừa trình bày trên là cho các nghiên cứu gồm hai nhóm độc lập nhau (như giữa hai nhóm nam và nữ), nhưng không thể ứng dụng cho các nghiên cứu mà một nhóm đối tượng được theo dõi theo thời gian. Tôi tạm gọi các nghiên cứu này là nghiên cứu theo cặp. Trong các nghiên cứu này, chúng ta cần sử dụng một kiểm định t có tên là paired t-test.

Ví dụ 12. Một nhóm bệnh nhân gồm 10 người được điều trị bằng một thuốc nhằm giảm huyết áp. Huyết áp của bệnh nhân được đo lúc khởi đầu nghiên cứu (lúc chưa điều trị), và sau khi điều khị. Số liệu huyết áp của 10 bệnh nhân như sau:

Trước khi điều trị (x_0)	180, 140, 160, 160, 220, 185, 145, 160, 160, 170
Sau khi điều trị (x_1)	170, 145, 145, 125, 205, 185, 150, 150, 145, 155

Câu hỏi đặt ra là độ biến chuyển huyết áp trên có đủ để kết luận rằng thuốc điều trị có hiệu quả giảm áp huyết. Để trả lời câu hỏi này, chúng ta dùng kiểm định t cho từng cặp như sau:

```
> # nhập dữ kiện
> before <- c(180, 140, 160, 160, 220, 185, 145, 160, 160, 170)
> after  <- c(170, 145, 145, 125, 205, 185, 150, 150, 145, 155)
> bp <- data.frame(before, after)

> # kiểm định t
> t.test(before, after, paired=TRUE)
```

```
        Paired t-test

data:  before and after
t = 2.7924, df = 9, p-value = 0.02097
alternative hypothesis: true difference in means is not equal to
0
95 percent confidence interval:
  1.993901 19.006099
sample estimates:
mean of the differences
                  10.5
```

Kết quả trên cho thấy sau khi điều trị áp suất máu giảm 10.5 mmHg, và khoảng tin cậy 95% là từ 2.0 mmHg đến 19 mmHg, với trị số p = 0.0209. Như vậy, chúng ta có bằng chứng để phát biểu rằng mức độ giảm huyết áp có ý nghĩa thống kê.

Chú ý nếu chúng ta phân tích sai bằng kiểm định thống kê cho hai nhóm độc lập dưới đây thì trị số p = 0.32 cho biết mức độ giảm áp suất không có ý nghĩa thống kê!

```
> t.test(before, after)

        Welch Two Sample t-test

data:  before and after
t = 1.0208, df = 17.998, p-value = 0.3209
alternative hypothesis: true difference in means is not equal to
0
95 percent confidence interval:
 -11.11065  32.11065
sample estimates:
mean of x mean of y
    168.0     157.5
```


9.6 Kiểm định Wilcoxon cho các biến số theo cặp (`wilcox.test`)

Thay vì dùng kiểm định t cho từng cặp, chúng ta cũng có thể sử dụng hàm `wilcox.test` cho cùng mục đích:

```
> wilcox.test(before, after, paired=TRUE)

        Wilcoxon signed rank test with continuity correction

data:  before and after
V = 42, p-value = 0.02291
alternative hypothesis: true mu is not equal to 0
```

Kết quả trên một lần nữa khẳng định rằng độ giảm áp suất máu có ý nghĩa thống kê với trị số (p=0.023) chẳng khác mấy so với kiểm định t cho từng cặp.

9.7 Tần số (frequency)

Hàm `table` trong R có chức năng cho chúng ta biết về tần số của một biến số mang tính phân loại như `sex` và `ethnicity`.

```
> table(sex)
sex
Female   Male
    69     31

> table(ethnicity)
ethnicity
  African     Asian Caucasian    Others
        8        60        30         2
```

Một bảng thống kê 2 chiều:

```
> table(sex, ethnicity)
        ethnicity
sex      African Asian Caucasian Others
  Female       4    43        22      0
  Male         4    17         8      2
```

Chú ý trong các bảng thống kê trên, hàm `table` không cung cấp cho chúng ta số phần trăm. Để tính số phần trăm, chúng ta cần đến hàm `prop.table` và cách sử dụng có thể minh hoạ như sau:

```
# tạo ra một object tên là freq để chứa kết quả tần số
> freq <- table(sex, ethnicity)

# kiểm tra kết quả
> freq
        ethnicity
sex      African Asian Caucasian Others
  Female       4    43        22      0
  Male         4    17         8      2

# dùng hàm margin.table để xem kết quả
> margin.table(freq, 1)
sex
Female   Male
    69     31

> margin.table(freq, 2)
ethnicity
  African     Asian Caucasian    Others
```

```
        8        60        30         2

# tính phần trăm bằng hàm prop.table
> prop.table(freq, 1)
        ethnicity
sex          African      Asian  Caucasian     Others
  Female 0.05797101 0.62318841 0.31884058 0.00000000
  Male   0.12903226 0.54838710 0.25806452 0.06451613
```

Trong bảng thống kê trên, `prop.table` tính tỉ lệ sắc tộc cho từng giới tính. Chẳng hạn như ở nữ giới (female), 5.8% là người Phi châu, 62.3% là người Á châu, 31.8% là người Tây phương da trắng . Tổng cộng là 100%. Tương tự, ớ nam giới tỉ lệ người Phi châu là 12.9%, Á châu là 54.8%, v.v…

```
# tính phần trăm bằng hàm prop.table
> prop.table(freq, 2)
        ethnicity
sex         African     Asian Caucasian    Others
  Female 0.5000000 0.7166667 0.7333333 0.0000000
  Male   0.5000000 0.2833333 0.2666667 1.0000000
```

Trong bảng thống kê trên, `prop.table` tính tỉ lệ giới tính cho từng sắc tộc. Chẳng hạn như trong nhóm người Á châu, 71.7% là nữ và 28.3% là nam.

```
# tính phần trăm cho toàn bộ bảng
> freq/sum(freq)
        ethnicity
sex      African Asian Caucasian Others
  Female    0.04  0.43      0.22   0.00
  Male      0.04  0.17      0.08   0.02
```


9.8 Kiểm định tỉ lệ (proportion test, `prop.test, binom.test`)

Kiểm định một tỉ lệ thường dựa vào giả định phân phối nhị phân (binomial distribution). Với một số mẫu n và tỉ lệ p, và nếu n lớn (tức hơn 50 chẳng hạn), thì phân phối nhị phân có thể tương đương với phân phối chuẩn với số trung bình np và phương sai $np(1-p)$. Gọi x là số biến cố mà chúng ta quan tâm, kiểm định giả thiết $p = \pi$ có thể sử dụng thống kê sau đây:

$$z = \frac{x - n\pi}{\sqrt{n\pi(1-\pi)}}$$

Ở đây, z tuân theo luật phân phối chuẩn với trung bình 0 và phương sai 1. Cũng có thể nói z^2 tuân theo luật phân phối Chi bình phương với bậc tự do bằng 1.

Ví dụ 13. Trong nghiên cứu trên, chúng ta thấy có 69 nữ và 31 nam. Như vậy tỉ lệ nữ là 0.69 (hay 69%). Để kiểm định xem tỉ lệ này có thật sự khác với tỉ lệ 0.5 hay không, chúng ta có thể sử dụng hàm `prop.test(x, n, π)` như sau:

```
> prop.test(69, 100, 0.50)

        1-sample proportions test with continuity correction

data:  69 out of 100, null probability 0.5
X-squared = 13.69, df = 1, p-value = 0.0002156
alternative hypothesis: true p is not equal to 0.5
95 percent confidence interval:
 0.5885509 0.7766330
sample estimates:
   p
0.69
```

Trong kết quả trên, `prop.test` ước tính tỉ lệ nữ giới là 0.69, và khoảng tin cậy 95% là 0.588 đến 0.776. Giá trị Chi bình phương là 13.69, với trị số p = 0.00216. Như vậy, nghiên cứu này có tỉ lệ nữ cao hơn 50%.

Một cách tính chính xác hơn kiểm định tỉ lệ là kiểm định nhị phân `bionom.test(x, n, π)` như sau:

```
> binom.test(69, 100, 0.50)

        Exact binomial test

data:  69 and 100
number of successes = 69, number of trials = 100, p-value = 0.0001831
alternative hypothesis: true probability of success is not equal to 0.5
95 percent confidence interval:
 0.5896854 0.7787112
sample estimates:
probability of success
                  0.69
```

Nói chung, kết quả của kiểm định nhị phân không khác gì so với kiểm định Chi bình phương, với trị số p = 0.00018, chúng ta càng có bằng chứng để kết luận rằng tỉ lệ nữ giới trong nghiên cứu này thật sự cao hơn 50%.

9.9 So sánh hai tỉ lệ (`prop.test, binom.test`)

Phương pháp so sánh hai tỉ lệ có thể khai triển trực tiếp từ lí thuyết kiểm định một tỉ lệ vừa trình bày trên. Cho hai mẫu với số đối tượng n_1 và n_2, và số biến cố là x_1 và x_2. Do đó, chúng ta có thể ước tính hai tỉ lệ p_1 và p_2. Lí thuyết xác suất cho phép chúng ta phát biểu rằng độ khác biệt giữa hai mẫu $d = p_1 - p_2$ tuân theo luật phân phối chuẩn với số trung bình 0 và phương sai bằng:

$$V_d = \left(\frac{1}{n_1} + \frac{1}{n_2}\right) p(1-p)$$

Trong đó:

$$p = \frac{x_1 + x_2}{n_1 + n_2}$$

Thành ra, $z = d/V_d$ tuân theo luật phân phối chuẩn với trung bình 0 và phương sai 1. Nói cách khác, z^2 tuân theo luật phân phối Chi bình phương với bậc tự do bằng 1. Do đó, chúng ta cũng có thể sử dụng `prop.test` để kiểm định hai tỉ lệ.

Ví dụ 14. Một nghiên cứu được tiến hành so sánh hiệu quả của thuốc chống gãy xương. Bệnh nhân được chia thành hai nhóm: nhóm A được điều trị gồm có 100 bệnh nhân, và nhóm B không được điều trị gồm 110 bệnh nhân. Sau thời gian 12 tháng theo dõi, nhóm A có 7 người bị gãy xương, và nhóm B có 20 người gãy xương. Vấn đề đặt ra là tỉ lệ gãy xương trong hai nhóm này bằng nhau (tức thuốc không có hiệu quả)? Để kiểm định xem hai tỉ lệ này có thật sự khác nhau, chúng ta có thể sử dụng hàm `prop.test(x, n, π)` như sau:

```
> fracture <- c(7, 20)
> total <- c(100, 110)
> prop.test(fracture, total)

        2-sample test for equality of proportions with continuity
correction

data:  fracture out of total
X-squared = 4.8901, df = 1, p-value = 0.02701
alternative hypothesis: two.sided
95 percent confidence interval:
 -0.20908963 -0.01454673
sample estimates:
   prop 1    prop 2
0.0700000 0.1818182
```

Kết quả phân tích trên cho thấy tỉ lệ gãy xương trong nhóm 1 là 0.07 và nhóm 2 là 0.18. Phân tích trên còn cho thấy xác suất 95% rằng độ khác biệt giữa hai nhóm có thể 0.01 đến 0.20 (tức 1 đến 20%). Với trị số p = 0.027, chúng ta có thể nói rằng tỉ lệ gãy xương trong nhóm A quả thật thấp hơn nhóm B.

9.10 So sánh nhiều tỉ lệ (`prop.test, chisq.test`)

Kiểm định `prop.test` còn có thể sử dụng để kiểm định nhiều tỉ lệ cùng một lúc. Trong nghiên cứu trên, chúng ta có 4 nhóm sắc tộc và tần số cho từng giới tính như sau:

```
> table(sex, ethnicity)
```

```
        ethnicity
sex      African Asian Caucasian Others
  Female       4    43        22      0
  Male         4    17         8      2
```

Chúng ta muốn biết tỉ lệ nữ giới giữa 4 nhóm sắc tộc có khác nhau hay không, và để trả lời câu hỏi này, chúng ta lại dùng `prop.test` như sau:

```
> female  <- c( 4, 43, 22, 0)
> total <- c(8, 60, 30, 2)
> prop.test(female, total)

        4-sample test for equality of proportions without continuity
        correction

data:  female out of total
X-squared = 6.2646, df = 3, p-value = 0.09942
alternative hypothesis: two.sided
sample estimates:
   prop 1    prop 2    prop 3    prop 4
0.5000000 0.7166667 0.7333333 0.0000000

Warning message:
Chi-squared approximation may be incorrect in: prop.test(female, total)
```

Tuy tỉ lệ nữ giới giữa các nhóm có vẻ khác nhau lớn (73% trong nhóm 3 (người da trắng) so với 50% trong nhóm 1 (Phi châu) và 71.7% trong nhóm Á châu, nhưng kiểm định Chi bình phương cho biết trên phương diện thống kê, các tỉ lệ này không khác nhau, vì trị số p = 0.099.

9.10.1 Kiểm định Chi bình phương (Chi squared test, `chisq.test`)

Thật ra, kiểm định Chi bình phương còn có thể tính toán bằng hàm `chisq.test` như sau:

```
> chisq.test(sex, ethnicity)

        Pearson's Chi-squared test

data:  sex and ethnicity
X-squared = 6.2646, df = 3, p-value = 0.09942

Warning message:
Chi-squared approximation may be incorrect in: chisq.test(sex,
ethnicity)
```

Kết quả này hoàn toàn giống với kết quả từ hàm `prop.test`.

9.10.2 Kiểm định Fisher (Fisher's exact test, `fisher.test`)

Trong kiểm định Chi bình phương trên, chúng ta chú ý cảnh báo:

```
"Warning message:
Chi-squared approximation may be incorrect in: prop.test(female, total)"
```

Vì trong nhóm 4, không có nữ giới cho nên tỉ lệ là 0%. Hơn nữa, trong nhóm này chỉ có 2 đối tượng. Vì số lượng đối tượng quá nhỏ, cho nên các ước tính thống kê có thể không đáng tin cậy. Một phương pháp khác có thể áp dụng cho các nghiên cứu với tần số thấp như trên là kiểm định `fisher` (còn gọi là Fisher's exact test). Bạn đọc có thể tham khảo lí thuyết đằng sau kiểm định fisher để hiểu rõ hơn về logic của phương pháp này, nhưng ở đây, chúng ta chỉ quan tâm đến cách dùng R để tính toán kiểm định này. Chúng ta chỉ đơn giản lệnh:

```
> fisher.test(sex, ethnicity)

        Fisher's Exact Test for Count Data

data:  sex and ethnicity
p-value = 0.1048
alternative hypothesis: two.sided
```

Chú ý trị số p từ kiểm định Fisher là 0.1048, tức rất gần với trị số p của kiểm định Chi bình phương. Cho nên, chúng ta có thêm bằng chứng để khẳng định rằng tỉ lệ nữ giới giữa các sắc tộc không khác nhau một cách đáng kể.

10. Phân tích hồi qui tuyến tính

Ví dụ 15. Để minh họa cho vấn đề, chúng ta thử xem xét nghiên cứu sau đây, mà trong đó nhà nghiên cứu đo lường độ cholestrol trong máu của 18 đối tượng nam. Tỉ trọng cơ thể (body mass index) cũng được ước tính cho mỗi đối tượng bằng công thức tính BMI là lấy trọng lượng (tính bằng kg) chia cho chiều cao bình phương (m^2). Kết quả đo lường như sau:

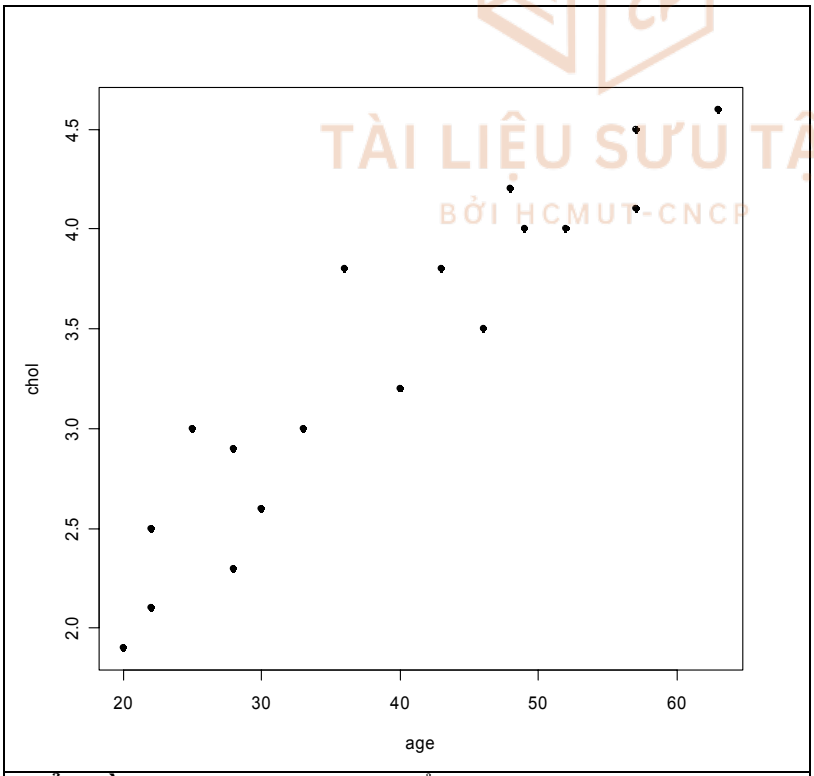
Độ tuổi, tỉ trọng cơ thể và cholesterol

Mã số ID (`id`)	Độ tuổi (`age`)	BMI (`bmi`)	Cholesterol (`chol`)
1	46	25.4	3.5
2	20	20.6	1.9
3	52	26.2	4.0
4	30	22.6	2.6
5	57	25.4	4.5
6	25	23.1	3.0
7	28	22.7	2.9
8	36	24.9	3.8

9	22	19.8	2.1
10	43	25.3	3.8
11	57	23.2	4.1
12	33	21.8	3.0
13	22	20.9	2.5
14	63	26.7	4.6
15	40	26.4	3.2
16	48	21.2	4.2
17	28	21.2	2.3
18	49	22.8	4.0

Nhìn sơ qua số liệu chúng ta thấy người có độ tuổi càng cao độ cholesterol cũng càng cao. Chúng ta thử nhập số liệu này vào R và vẽ một biểu đồ tán xạ như sau:

```
> age <- c(46,20,52,30,57,25,28,36,22,43,57,33,22,63,40,48,28,49)

> bmi <-c(25.4,20.6,26.2,22.6,25.4,23.1,22.7,24.9,19.8,25.3,23.2,
          21.8,20.9,26.7,26.4,21.2,21.2,22.8)

> chol <- c(3.5,1.9,4.0,2.6,4.5,3.0,2.9,3.8,2.1,3.8,4.1,3.0,
            2.5,4.6,3.2, 4.2,2.3,4.0)

> data <- data.frame(age, bmi, chol)
> plot(chol ~ age, pch=16)
```



Biểu đồ 18. Liên hệ giữa độ tuổi và cholesterol.

Biểu đồ 18 trên đây gợi ý cho thấy mối liên hệ giữa độ tuổi (`age`) và cholesterol là một đường thẳng (tuyến tính). Để "đo lường" mối liên hệ này, chúng ta có thể sử dụng hệ số tương quan (coefficient of correlation).

10.1 Hệ số tương quan

Hệ số tương quan (r) là một chỉ số thống kê đo lường mối liên hệ tương quan giữa hai biến số, như giữa độ tuổi (x) và cholesterol (y). Hệ số tương quan có giá trị từ -1 đến 1. Hệ số tương quan bằng 0 (hay gần 0) có nghĩa là hai biến số không có liên hệ gì với nhau; ngược lại nếu hệ số bằng -1 hay 1 có nghĩa là hai biến số có một mối liên hệ tuyệt đối. Nếu giá trị của hệ số tương quan là âm ($r < 0$) có nghĩa là khi x tăng cao thì y giảm (và ngược lại, khi x giảm thì y tăng); nếu giá trị hệ số tương quan là dương ($r > 0$) có nghĩa là khi x tăng cao thì y cũng tăng, và khi x tăng cao thì y cũng giảm theo.

Thực ra có nhiều hệ số tương quan trong thống kê, nhưng ở đây tôi sẽ trình bày 3 hệ số tương quan thông dụng nhất: hệ số tương quan Pearson r, Spearman ρ, và Kendall τ.

10.1.1 Hệ số tương quan Pearson

Cho hai biến số x và y từ n mẫu, hệ số tương quan Pearson được ước tính bằng công thức sau đây: $r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}}$. Trong đó, như định nghĩa phần trên, $\bar{x}$ và $\bar{y}$ là giá trị trung bình của biến số x và y. Để ước tính hệ số tương quan giữa độ tuổi `age` và cholesterol, chúng ta có thể sử dụng hàm `cor(x,y)` như sau:

```
> cor(age, chol)
[1] 0.936726
```

Chúng ta có thể kiểm định giả thiết hệ số tương quan bằng 0 (tức hai biến x và y không có liên hệ). Phương pháp kiểm định này thường dựa vào phép biến đổi Fisher mà R đã có sẵn một hàm `cor.test` để tiến hành việc tính toán.

```
> cor.test(age, chol)

        Pearson's product-moment correlation

data:  age and chol
t = 10.7035, df = 16, p-value = 1.058e-08
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.8350463 0.9765306
sample estimates:
      cor
0.936726
```

10.1.2 Hệ số tương quan Spearman ρ

Hệ số tương quan Pearson chỉ hợp lí nếu biến số *x* và *y* tuân theo luật phân phối chuẩn. Nếu *x* và *y* không tuân theo luật phân phối chuẩn, chúng ta phải sử dụng một hệ số tương quan khác tên là Spearman, một phương pháp phân tích phi tham số. Hệ số này được ước tính bằng cách biến đổi hai biến số *x* và *y* thành thứ bậc (rank), và xem độ tương quan giữa hai dãy số bậc. Do đó, hệ số còn có tên tiếng Anh là Spearman's Rank correlation. R ước tính hệ số tương quan Spearman bằng hàm `cor.test` với thông số `method="spearman"` như sau:

```
> cor.test(age, chol, method="spearman")

        Spearman's rank correlation rho

data:  age and chol
S = 51.1584, p-value = 2.57e-09
alternative hypothesis: true rho is not equal to 0
sample estimates:
     rho
0.947205

Warning message:
Cannot compute exact p-values with ties in: cor.test.default(age,
chol, method = "spearman")
```

10.1.3 Hệ số tương quan Kendall τ

Hệ số tương quan Kendall (cũng là một phương pháp phân tích phi tham số) được ước tính bằng cách tìm các cặp số (*x, y*) "song hành" với nhau. Một cặp (*x, y*) song hành ở đây được định nghĩa là hiệu (độ khác biệt) trên trục hoành có cùng dấu hiệu (dương hay âm) với hiệu trên trục tung. Nếu hai biến số *x* và *y* không có liên hệ với nhau, thì số cặp song hành bằng hay tương đương với số cặp không song hành.

Bởi vì có nhiều cặp phải kiểm định, phương pháp tính toán hệ số tương quan Kendall đòi hỏi thời gian của máy tính khá cao. Tuy nhiên, nếu một dữ liệu dưới 5000 đối tượng thì một máy vi tính có thể tính toán khá dễ dàng. R dùng hàm `cor.test` với thông số `method="kendall"` để ước tính hệ số tương quan Kendall:

```
> cor.test(age, chol, method="kendall")

        Kendall's rank correlation tau

data:  age and chol
z = 4.755, p-value = 1.984e-06
alternative hypothesis: true tau is not equal to 0
sample estimates:
      tau
0.8333333
```

```
Warning message:
Cannot compute exact p-value with ties in: cor.test.default(age,
chol, method = "kendall")
```

10.2 Mô hình của hồi qui tuyến tính đơn giản

Để tiện việc theo dõi và mô tả mô hình, gọi độ tuổi cho cá nhân i là x_i và cholesterol là y_i. Ở đây i = 1, 2, 3, …, 18. Mô hình hồi tuyến tính phát biểu rằng:

$$y_i = \alpha + \beta x_i + \varepsilon_i$$

Nói cách khác, phương trình trên giả định rằng độ cholesterol của một cá nhân bằng một hằng số α cộng với một hệ số β liên quan đến độ tuổi, và một sai số ε_i. Trong phương trình trên, α là *chặn* (intercept, tức giá trị lúc x_i =0), và β là độ dốc (slope hay gradient). Trong thực tế, α và β là hai thông số (paramater, còn gọi là *regression coefficient* hay hệ số hồi qui), và ε_i là một biến số theo luật phân phối chuẩn với trung bình 0 và phương sai σ^2.

Các thông số α, β và σ^2 phải được ước tính từ dữ liệu. Phương pháp để ước tính các thông số này là phương pháp *bình phương nhỏ nhất* (least squares method). Như tên gọi, phương pháp bình phương nhỏ nhất tìm giá trị α, β sao cho $\sum_{i=1}^{n}\left[y_i - (\alpha + \beta x_i)\right]^2$ nhỏ nhất. Sau vài thao tác toán, có thể chứng minh dễ dàng rằng, ước số cho α và β đáp ứng điều kiện đó là:

$$\hat{\beta} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \text{ và } \hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}$$

Ở đây, $\bar{x}$ và $\bar{y}$ là giá trị trung bình của biến số x và y. Chú ý, tôi viết $\hat{\alpha}$ và $\hat{\beta}$ (với dấu mũ phía trên) là để nhắc nhở rằng đây là hai ước số (estimates) của α và β, chứ không phải α và β (chúng ta không biết chính xác α và β, nhưng chỉ có thể ước tính mà thôi). Sau khi đã có ước số $\hat{\alpha}$ và $\hat{\beta}$, chúng ta có thể ước tính độ cholesterol trung bình cho từng độ tuổi như sau:

$$\hat{y}_i = \hat{\alpha} + \hat{\beta} x_i$$

Tất nhiên, $\hat{y}_i$ ở đây chỉ là số trung bình cho độ tuổi x_i, và phần còn lại (tức y_i - $\hat{y}_i$) gọi là *phần dư* (*residual*). Và phương sai của phần dư có thể ước tính như sau: $s^2 = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)}{n-2}$. Ở đây, s^2 chính là ước số của σ^2.

Hàm `lm` (viết tắt từ **l**inear **m**odel) trong R có thể tính toán các giá trị của $\widehat{\alpha}$ và $\widehat{\beta}$, cũng như s^2 một cách nhanh gọn. Chúng ta tiếp tục với ví dụ bằng R như sau:

```
> lm(chol ~ age)

Call:
lm(formula = chol ~ age)

Coefficients:
(Intercept)          age
    1.08922      0.05779
```

Trong lệnh trên, "`chol ~ age`" có nghĩa là mô tả `chol` là một hàm số của `age`. Kết quả tính toán của `lm` cho thấy $\widehat{\alpha} = 1.0892$ và $\widehat{\beta} = 0.05779$. Nói cách khác, với hai thông số này, chúng ta có thể ước tính độ cholesterol cho bất cứ độ tuổi nào trong khoảng tuổi của mẫu bằng phương trình tuyến tính:

$$\hat{y}_i = 1.08922 + 0.05779 \times age$$

Phương trình này có nghĩa là khi độ tuổi tăng 1 năm thì độ cholesterol tăng khoảng 0.058 mmol/L.

Thật ra, hàm `lm` còn cung cấp cho chúng ta nhiều thông tin khác, nhưng chúng ta phải đưa các thông tin này vào một object. Gọi object đó là `reg`, thì lệnh sẽ là:

```
> reg <- lm(chol ~ age)
> summary(reg)

Call:
lm(formula = chol ~ age)

Residuals:
     Min       1Q   Median       3Q      Max
-0.40729 -0.24133 -0.04522  0.17939  0.63040

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.089218   0.221466   4.918 0.000154 ***
age         0.057788   0.005399  10.704 1.06e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3027 on 16 degrees of freedom
Multiple R-Squared: 0.8775,     Adjusted R-squared: 0.8698
F-statistic: 114.6 on 1 and 16 DF,  p-value: 1.058e-08
```

Lệnh thứ hai, `summary(reg)`, yêu cầu R liệt kê các thông tin tính toán trong `reg`. Phần kết quả chia làm 3 phần:

(a) Phần 1 mô tả phần dư (residuals) của mô hình hồi qui:

```
Residuals:
     Min       1Q   Median       3Q      Max
-0.40729 -0.24133 -0.04522  0.17939  0.63040
```

Chúng ta biết rằng trung bình phần dư phải là 0, và ở đây, số trung vị là -0.04, cũng không xa 0 bao nhiêu. Các số quantiles 25% (1Q) và 75% (3Q) cũng khá cân đối chung quan số trung vị, cho thấy phần dư của phương trình này tương đối cân đối.

(b) Phần hai trình bày ước số của $\widehat{\alpha}$ và $\widehat{\beta}$ cùng với sai số chuẩn và giá trị của kiểm định t. Giá trị kiểm định t cho $\widehat{\beta}$ là 10.74 với trị số p = 1.06e-08, cho thấy β không phải bằng 0. Nói cách khác, chúng ta có bằng chứng để cho rằng có một mối liên hệ giữa cholesterol và độ tuổi, và mối liên hệ này có ý nghĩa thống kê.

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.089218   0.221466   4.918 0.000154 ***
age         0.057788   0.005399  10.704 1.06e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(c) Phần ba của kết quả cho chúng ta thông tin về phương sai của phần dư (residual mean square). Ở đây, $s^2 = 0.3027$. Trong kết quả này còn có kiểm định F, cũng chỉ là một kiểm định xem có quả thật β bằng 0, tức có ý nghĩa tương tự như kiểm định t trong phần trên. Nói chung, trong trường hợp phân tích hồi qui tuyến tính đơn giản (với một yếu tố) chúng ta không cần phải quan tâm đến kiểm định F.

```
Residual standard error: 0.3027 on 16 degrees of freedom
Multiple R-Squared: 0.8775,     Adjusted R-squared: 0.8698
F-statistic: 114.6 on 1 and 16 DF,  p-value: 1.058e-08
```

Ngoài ra, phần 3 còn cho chúng ta một thông tin quan trọng, đó là trị số R^2 hay *hệ số xác định bội* (coefficient of determination). Tức là bằng tổng bình phương giữa số ước tính và trung bình chia cho tổng bình phương số quan sát và trung bình. Trị số R^2 trong ví dụ này là 0.8775, có nghĩa là phương trình tuyến tính (với độ tuổi là một yếu tố) giải thích khoảng 88% các khác biệt về độ cholesterol giữa các cá nhân. Tất nhiên trị số R^2 có giá trị từ 0 đến 100% (hay 1). Giá trị R^2 càng cao là một dấu hiệu cho thấy mối liên hệ giữa hai biến số độ tuổi và cholesterol càng chặt chẽ.

Một hệ số cũng cần đề cập ở đây là *hệ số điều chỉnh xác định bội* (mà trong kết quả trên R gọi là "Adjusted R-squared"). Đây là hệ số cho chúng ta biết mức độ cải tiến của phương sai phần dư (residual variance) do yếu tố độ tuổi có mặt trong mô hình tuyến tính. Nói chung, hệ số này không khác mấy so với hệ số xác định bội, và chúng ta cũng không cần chú tâm quá mức.

Giả định của phân tích hồi qui tuyến tính

Tất cả các phân tích trên dựa vào một số giả định quan trọng như sau:

(a) x là một biến số cố định hay fixed, ("cố định" ở đây có nghĩa là không có sai sót ngẫu nhiên trong đo lường);

(b) ε_i phân phối theo luật phân phối chuẩn;

(c) ε_i có giá trị trung bình (mean) là 0;

(d) ε_i có phương sai σ^2 cố định cho tất cả x_i; và

(e) các giá trị liên tục của ε_i không có liên hệ tương quan với nhau (nói cách khác, ε_1 và ε_2 không có liên hệ với nhau).

Nếu các giả định này không được đáp ứng thì phương trình mà chúng ta ước tính có vấn đề hợp lí (validity). Do đó, trước khi trình bày và diễn dịch mô hình trên, chúng ta cần phải kiểm tra xem các giả định trên có đáp ứng được hay không. Trong trường hợp này, giả định (a) không phải là vấn đề, vì độ tuổi không phải là một biến số ngẫu nhiên, và không có sai số khi tính độ tuổi của một cá nhân.

Đối với các giả định (b) đến (e), cách kiểm tra đơn giản nhưng hữu hiệu nhất là bằng cách xem xét mối liên hệ giữa $\hat{y}_i$, x_i, và phần dư e_i ($e_i = y_i - \hat{y}_i$) bằng những đồ thị tán xạ.

Với lệnh `fitted()` chúng ta có thể tính toán $\hat{y}_i$ cho từng cá nhân như sau (ví dụ đối với cá nhân 1, 46 tuổi, độ cholestrol có thể tiên đoán như sau: `1.08922 + 0.05779 x 46 = 3.747`).

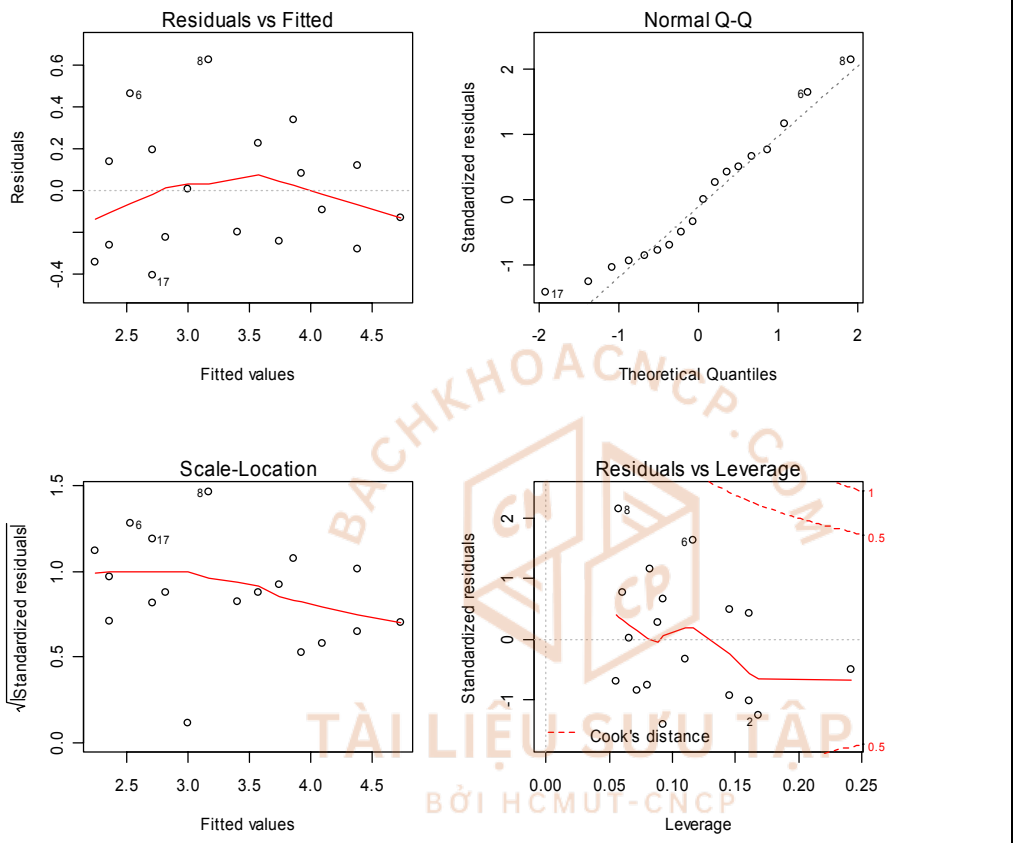

```
> fitted(reg)
       1        2        3        4        5        6        7        8
3.747483 2.244985 4.094214 2.822869 4.383156 2.533927 2.707292 3.169600
       9       10       11       12       13       14       15       16
2.360562 3.574118 4.383156 2.996234 2.360562 4.729886 3.400753 3.863060
      17       18
2.707292 3.920849
```

Với lệnh `resid()` chúng ta có thể tính toán phần dư e_i cho từng cá nhân như sau (với đối tượng 1, e_1 = `3.5 - 3.74748 = -0.24748`):

```
> resid(reg)
           1            2            3            4            5            6
-0.247483426 -0.344985415 -0.094213736 -0.222869265  0.116844338  0.466072660
           7            8            9           10           11           12
 0.192707505  0.630400424 -0.260562185  0.225881729 -0.283155662  0.003765579
          13           14           15           16           17           18
 0.139437815 -0.129885972 -0.200753116  0.336939804 -0.407292495  0.079151419
```

Để kiểm tra các giả định trên, chúng ta có thể vẽ một loạt 4 đồ thị mà tôi sẽ giải thích sau đây:

```
> op <- par(mfrow=c(2,2))          #yêu cầu R dành ra 4 cửa sổ
> plot(reg)                        #vẽ các đồ thị trong reg
```



Biểu đồ 19. Phân tích phần dư để kiểm tra các giả định trong phân tích hồi qui tuyến tính.

(a) Đồ thị bên trái dòng 1 vẽ phần dư e_i và giá trị tiên đoán cholesterol $\hat{y}_i$. Đồ thị này cho thấy các giá trị phần dư tập chung quanh đường y = 0, cho nên giả định (c), hay ε_i có giá trị trung bình 0, là có thể chấp nhận được.

(b) Đồ thị bên phải dòng 1 vẽ giá trị phần dư và giá trị kì vọng dựa vào phân phối chuẩn. Chúng ta thấy các số phần dư tập trung rất gần các giá trị trên đường chuẩn, và do đó, giả định (b), tức ε_i phân phối theo luật phân phối chuẩn, cũng có thể đáp ứng.

(c) Đồ thị bên trái dòng 2 vẽ căn số phần dư chuẩn (standardized residual) và giá trị của $\hat{y}_i$. Đồ thị này cho thấy không có gì khác nhau giữa các số phần dư chuẩn cho các giá trị

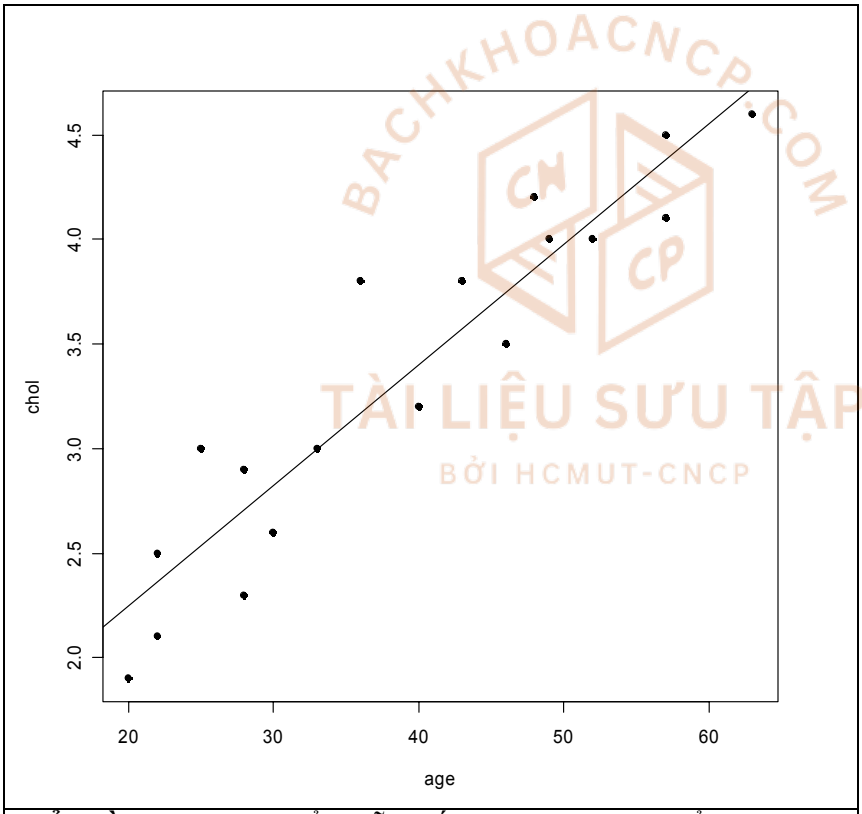
của $\hat{y}_i$, và do đó, giả định (d), tức ε_i có phương sai σ^2 cố định cho tất cả x_i, cũng có thể đáp ứng.

Nói chung qua phân tích phần dư, chúng ta có thể kết luận rằng mô hình hồi qui tuyến tính mô tả mối liên hệ giữa độ tuổi và cholesterol một cách khá đầy đủ và hợp lí.

Mô hình tiên đoán

Sau khi mô hình tiên đoán cholesterol đã được kiểm tra và tính hợp lí đã được thiết lập, chúng ta có thể vẽ đường biểu diễn của mối liên hệ giữa độ tuổi và cholesterol bằng lệnh `abline` như sau (xin nhắc lại object của phân tích là `reg`):

```
> plot(chol ~ age, pch=16)
> abline(reg)
```

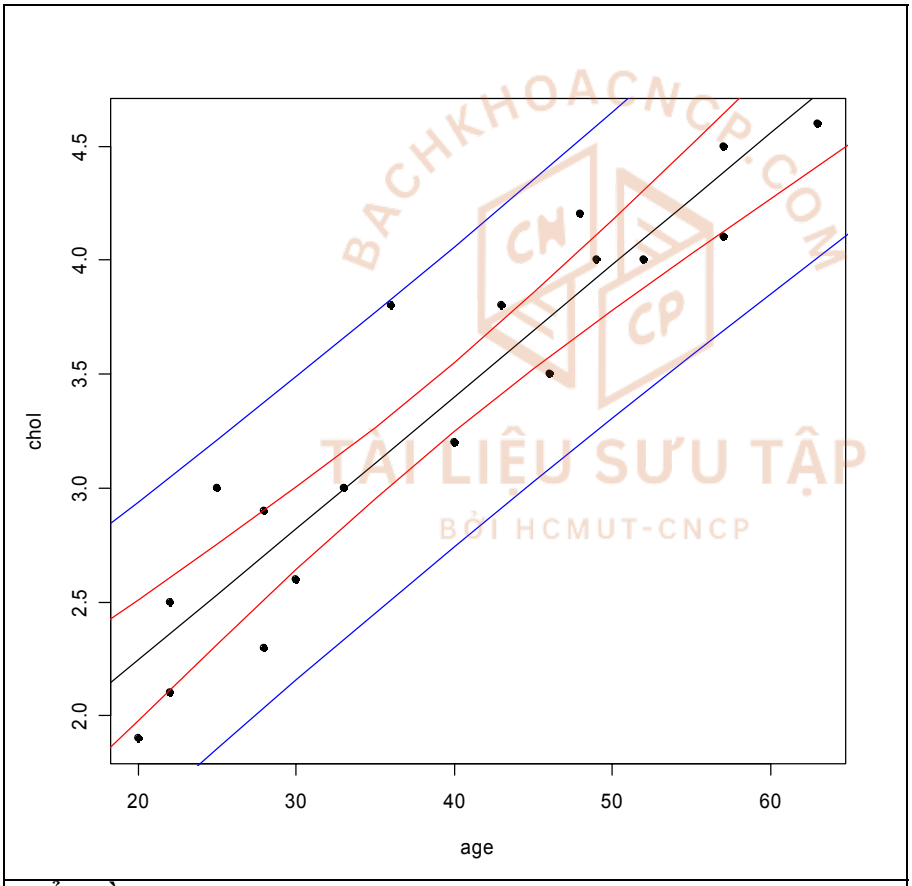


Biểu đồ 20. Đường biểu diễn mối liên hệ giữa độ tuổi (age) và cholesterol.

Nhưng mỗi giá trị $\hat{y}_i$ được tính từ ước số $\widehat{\alpha}$ và $\widehat{\beta}$, mà các ước số này đều có sai số chuẩn, cho nên giá trị tiên đoán $\hat{y}_i$ cũng có sai số. Nói cách khác, $\hat{y}_i$ chỉ là trung bình,

nhưng trong thực tế có thể cao hơn hay thấp hơn tùy theo chọn mẫu. Khoảng tin cậy 95% này có thể ước tính qua R bằng các lệnh sau đây:

```
> reg <- lm(chol ~ age)
> new <- data.frame(age = seq(15, 70, 5))
> pred.w.plim <- predict.lm(reg, new, interval="prediction")
> pred.w.clim <- predict.lm(reg, new, interval="confidence")
> resc <- cbind(pred.w.clim, new)
> resp <- cbind(pred.w.plim, new)
> plot(chol ~ age, pch=16)
> lines(resc$fit ~ resc$age)
> lines(resc$lwr ~ resc$age, col=2)
> lines(resc$upr ~ resc$age, col=2)
> lines(resp$lwr ~ resp$age, col=4)
> lines(resp$upr ~ resp$age, col=4)
```



Biểu đồ 21. Giá trị tiên đoán và khoảng tin cậy 95%.

Biểu đồ trên vẽ giá trị tiên đoán trung bình $\hat{y}_i$ (đường thẳng màu đen), và khoảng tin cậy 95% của giá trị này là đường màu đỏ. Ngoài ra, đường màu xanh là khoảng tin cậy của giá trị tiên đoán cholesterol cho một độ tuổi mới trong quần thể.

10.3 Mô hình hồi qui tuyến tính đa biến (multiple linear regression)

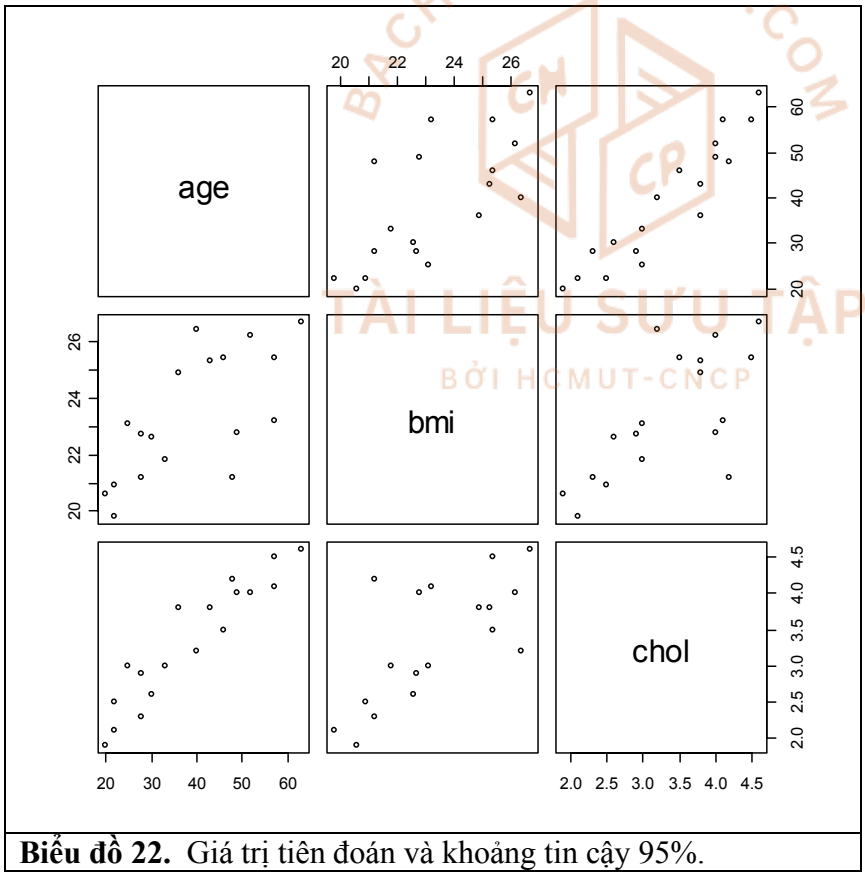
Mô hình được diễn đạt qua phương trình $y_i = \alpha + \beta x_i + \varepsilon_i$ có một yếu tố duy nhất (đó là x), và vì thế thường được gọi là mô hình hồi qui tuyến tính đơn giản (simple linear regression model). Trong thực tế, chúng ta có thể phát triển mô hình này thành nhiều biến, chứ không chỉ giới hạn một biến như trên, chẳng hạn như:

$$y_i = \alpha + \beta_1 x_{1i} + \beta_2 x_{2i} + ... + \beta_k x_{ki} + \varepsilon_i$$

Chú ý trong phương trình trên, chúng ta có nhiều biến x (x_1, x_2, ... *đến* x_k), và mỗi biến có một thông số β_j (j = 1, 2, …, k) cần phải ước tính. Vì thế mô hình này còn được gọi là mô hình hồi qui tuyến tính đa biến.

Ví dụ 16. Chúng ta quay lại nghiên cứu về mối liên hệ giữa độ tuổi, `bmi` và cholesterol. Trong ví dụ, chúng ta chỉ mới xét mối liên hệ giữa độ tuổi và cholesterol, mà chưa xem đến mối liên hệ giữa cả hai yếu tố độ tuổi và `bmi` và cholesterol. Biểu đồ sau đây cho chúng ta thấy mối liên hệ giữa ba biến số này:

```
> pairs(data)
```



Biểu đồ 22. Giá trị tiên đoán và khoảng tin cậy 95%.

Cũng như giữa độ tuổi và cholesterol, mối liên hệ giữa bmi và cholesterol cũng gần tuân theo một đường thẳng. Biểu đồ trên còn cho chúng ta thấy độ tuổi và bmi có liên hệ với

nhau. Thật vậy, phân tích hồi qui tuyến tính đơn giản giữa bmi và cholesterol cho thấy như mối liên hệ này có ý nghĩa thống kê:

```
> summary(lm(chol ~  bmi))

Call:
lm(formula = chol ~ bmi)

Residuals:
    Min      1Q  Median      3Q     Max
-0.9403 -0.3565 -0.1376  0.3040  1.4330

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -2.83187    1.60841  -1.761  0.09739 .
bmi          0.26410    0.06861   3.849  0.00142 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.623 on 16 degrees of freedom
Multiple R-Squared: 0.4808,     Adjusted R-squared: 0.4483
F-statistic: 14.82 on 1 and 16 DF,  p-value: 0.001418
```

BMI giải thích khoảng 48% độ dao động về cholesterol giữa các cá nhân. Nhưng vì BMI cũng có liên hệ với độ tuổi, chúng ta muốn biết nếu hai yếu tố này được phân tích cùng một lúc thì yếu tố nào quan trọng hơn. Để biết ảnh hưởng của cả hai yếu tố age (x_1) và bmi (tạm gọi là x_2) đến cholesterol (y) qua một mô hình hồi qui tuyến tính đa biến, và mô hình đó là:

$$y_i = \alpha + \beta_1 x_{1i} + \beta_2 x_{2i} + \varepsilon_i$$

hay phương trình cũng có thể mô tả bằng kí hiệu ma trận: $\mathbf{Y} = \mathbf{X}\beta + \boldsymbol{\varepsilon}$ mà tôi vừa trình bày trên. Ở đây, **Y** là một vector vector 18 x 1, **X** là một matrix 18 x 2 phần tử, β và một vector 2 x 1, và **ε** là vector gồm 18 x 1 phần tử. Để ước tính hai hệ số hồi qui, β_1 và β_2 chúng ta cũng ứng dụng hàm `lm()` trong R như sau:

```
> mreg <- lm(chol ~ age + bmi)
> summary(mreg)

Call:
lm(formula = chol ~ age + bmi)

Residuals:
    Min      1Q  Median      3Q     Max
-0.3762 -0.2259 -0.0534  0.1698  0.5679

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 0.455458   0.918230   0.496    0.627
```

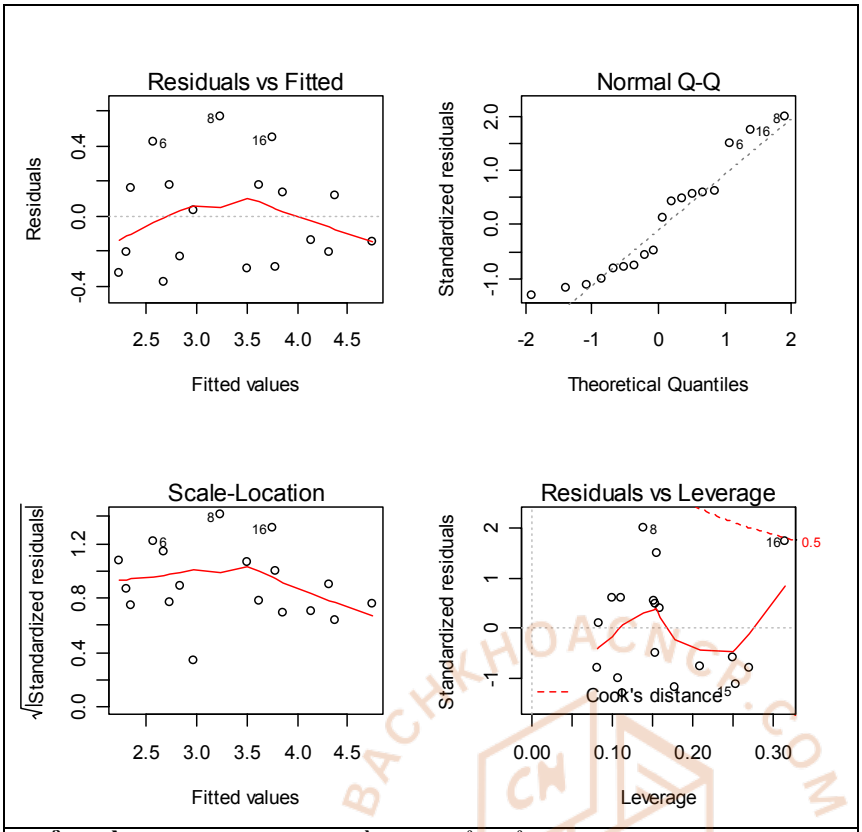
```
age          0.054052   0.007591   7.120 3.50e-06 ***
bmi          0.033364   0.046866   0.712    0.487
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3074 on 15 degrees of freedom
Multiple R-Squared: 0.8815,     Adjusted R-squared: 0.8657
F-statistic: 55.77 on 2 and 15 DF,  p-value: 1.132e-07
```

Kết quả phân tích trên cho thấy ước số $\hat{\alpha} = 0.455$, $\hat{\beta}_1 = 0.054$ và $\hat{\beta}_2 = 0.0333$. Nói cách khác, chúng ta có phương trình ước đoán độ cholesterol dựa vào hai biến số độ tuổi và bmi như sau:

$$\text{Cholesterol} = 0.455 + 0.054(\text{age}) + 0.0333(\text{bmi})$$

Phương trình cho biết khi độ tuổi tăng 1 năm thì cholesterol tăng 0.054 mg/L (ước số này không khác mấy so với 0.0578 trong phương trình chỉ có độ tuổi), và mỗi 1 kg/m^2 tăng BMI thì cholesterol tăng 0.0333 mg/L. Hai yếu tố này "giải thích" khoảng 88.2% (R^2 = 0.8815) độ dao động của cholesterol giữa các cá nhân.

Chúng ta chú ý phương trình với độ tuổi (trong phân tích phần trước) giải thích khoảng 87.7% độ dao động cholesterol giữa các cá nhân. Khi chúng ta thêm yếu tố BMI, hệ số này tăng lên 88.2%, tức chỉ 0.5%. Câu hỏi đặt ra là 0.5% tăng trưởng này có ý nghĩa thống kê hay không. Câu trả lời có thể xem qua kết quả kiểm định yếu tố bmi với trị số p = 0.487. Như vậy, bmi không cung cấp cho chúng thêm thông tin hay tiên đoán cholesterol hơn những gì chúng ta đã có từ độ tuổi. Nói cách khác, khi độ tuổi đã được xem xét, thì ảnh hưởng của bmi không còn ý nghĩa thống kê. Điều này có thể hiểu được, bởi vì qua Biểu đồ 10.5 chúng ta thấy độ tuổi và bmi có một mối liên hệ khá cao. Vì hai biến này có tương quan với nhau, chúng ta không cần cả hai trong phương trình. (Tuy nhiên, ví dụ này chỉ có tính cách minh họa cho việc tiến hành phân tích hồi qui tuyến tính đa biến bằng R, chứ không có ý định mô phỏng dữ liệu theo định hướng sinh học).



Biểu đồ 23. Phân tích phần dư để kiểm tra các giả định trong phân tích hồi qui tuyến tính đa biến.

Tuy BMI không có ý nghĩa thống kê trong trường hợp này, **Biểu đồ 10.6** cho thấy các giả định về mô hình hồi qui tuyến tính có thể đáp ứng.

11. Phân tích phương sai

11.1 Phân tích phương sai đơn giản (one-way analysis of variance - ANOVA)

Ví dụ 17. Bảng dưới đây so sánh độ galactose trong 3 nhóm bệnh nhân: nhóm 1 gồm 9 bệnh nhân với bệnh Crohn; nhóm 2 gồm 11 bệnh nhân với bệnh viêm ruột kết (colitis); và nhóm 3 gồm 20 đối tượng không có bệnh (gọi là nhóm đối chứng). Câu hỏi đặt ra là độ galactose giữa 3 nhóm bệnh nhân có khác nhau hay không?

Độ galactose cho 3 nhóm bệnh nhân Crohn, viêm ruột kết và đối chứng

Nhóm 1: bệnh Crohn	Nhóm 2: bệnh viêm ruột kết	Nhóm 3: đối chứng (control)

1343 1393 1420 1641 1897 2160 2169 2279 2890	1264 1314 1399 1605 2385 2511 2514 2767 2827 2895 3011	1809 2850 1926 2964 2283 2973 2384 3171 2447 3257 2479 3271 2495 3288 2525 3358 2541 3643 2769 3657
n=9 Trung bình: 1910 SD: 516	*n*=11 Trung bình: 2226 SD: 727	*n*=20 Trung bình: 2804 SD: 527

Chú thích: SD là độ lệch chuẩn (standard deviation).

Gọi giá trị trung bình của ba nhóm là μ_1, μ_2, và μ_3, và nói theo ngôn ngữ của kiểm định giả thiết thì giả thiết đảo là:

$$H_o: \mu_1 = \mu_2 = \mu_3$$

Và giả thiết chính là:

$$H_A: \text{có một khác biệt giữa 3 } \mu_j \ (j = 1,2,3)$$

Thoạt đầu có lẽ bạn đọc, sau khi đã học qua phương pháp so sánh hai nhóm bằng kiểm định t, sẽ nghĩ rằng chúng ta cần làm 3 so sánh bằng kiểm định t: giữa nhóm 1 và 2, nhóm 2 và 3, và nhóm 1 và 3. Nhưng phương pháp này không hợp lí, vì có ba phương sai khác nhau. Phương pháp thích hợp cho so sánh là phân tích phương sai. Phân tích phương sai có thể ứng dụng để so sánh nhiều nhóm cùng một lúc (simultaneous comparisons).

Để minh họa cho phương pháp phân tích phương sai, chúng ta phải dùng kí hiệu. Gọi độ galactose của bệnh nhân *i* thuộc nhóm *j* (j = 1, 2, 3) là x_{ij}. Mô hình phân tích phương sai phát biểu rằng:

$$x_{ij} = \mu + \alpha_i + \varepsilon_{ij}$$

Hay cụ thể hơn:

$$x_{i1} = \mu + \alpha_1 + \varepsilon_{i1}$$
$$x_{i2} = \mu + \alpha_2 + \varepsilon_{i2}$$
$$x_{i3} = \mu + \alpha_3 + \varepsilon_{i3}$$

Trước hết, chúng ta cần phải nhập dữ liệu vào R. Bước thứ nhất là báo cho R biết rằng chúng ta có ba nhóm bệnh nhân (1, 2 vả), nhóm 1 gồm 9 người, nhóm 2 có 11 người, và nhóm 3 có 20 người:

```
> group <- c(1,1,1,1,1,1,1,1,1, 2,2,2,2,2,2,2,2,2,2,2,
3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3)
```

Để phân tích phương sai, chúng ta phải định nghĩa biến `group` là một yếu tố - factor.

```
> group <- as.factor(group)
```

Bước kế tiếp, chúng ta nạp số liệu galactose cho từng nhóm như định nghĩa trên (gọi object là `galactose`):

```
> galactose <- c(1343,1393,1420,1641,1897,2160,2169,2279,2890,
1264,1314,1399,1605,2385,2511,2514,2767,2827,2895,3011,
1809,2850,1926,2964,2283,2973,2384,3171,2447,3257,2479,3271,2495,3288,
2525,3358,2541,3643,2769,3657)
```

Đưa hai biến `group` và `galactose` vào một dataframe và gọi là `data`:

```
> data <- data.frame(group, galactose)
> attach(data)
```

Sau khi đã có dử liệu sẵn sàng, chúng ta dùng hàm `lm()` để phân tích phương sai như sau:

```
> analysis <- lm(galactose ~ group)
```

Trong hàm trên chúng ta cho R biết biến `galactose` là một hàm số của `group`. Gọi kết quả phân tích là `analysis`.

Kết quả phân tích phương sai. Bây giờ chúng ta dùng lệnh `anova` để biết kết quả phân tích:

```
> anova(analysis)
Analysis of Variance Table

Response: galactose
          Df   Sum Sq  Mean Sq F value    Pr(>F)
group      2  5683620  2841810  8.6655 0.0008191 ***
Residuals 37 12133923   327944
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Trong kết quả trên, có ba cột: `Df` (degrees of freedom) là bậc tự do; `Sum Sq` là tổng bình phương (sum of squares), `Mean Sq` là trung bình bình phương (mean square); `F value` là giá trị *F*; và `Pr(>F)` là trị số P liên quan đến kiểm định *F*.

11.2 So sánh nhiều nhóm (multiple comparisons) và điều chỉnh trị số p

Cho *k* nhóm, chúng ta có ít nhất là *k(k-1)*/2 so sánh. Ví dụ trên có 3 nhóm, cho nên tổng số so sánh khả dĩ là 3 (giữa nhóm 1 và 2, nhóm 1 và 3, và nhóm 2 và 3). Khi *k*=10, số lần so sánh có thể lên rất cao. Như đã đề cập trong chương 7, khi có nhiều so sánh, trị số p tính toán từ các kiểm định thống kê không còn ý nghĩa ban đầu nữa, bởi vì các kiểm định này có thể cho ra kết quả dương tính giả (tức kết quả với p<0.05 nhưng

trong thực tế không có khác nhau hay ảnh hưởng). Do đó, trong trường hợp có nhiều so sánh, chúng ta cần phải điều chỉnh trị số p sao cho hợp lí.

Có khá nhiều phương pháp điều chỉnh trị số p, và 4 phương pháp thông dụng nhất là: Bonferroni, Scheffé, Holm và Tukey (tên của 4 nhà thống kê học danh tiếng). Phương pháp nào thích hợp nhất? Không có câu trả lời dứt khoát cho câu hỏi này, nhưng hai điểm sau đây có thể giúp bạn đọc quyết định tốt hơn:

(a) Nếu $k < 10$, chúng ta có thể áp dụng bất cứ phương pháp nào để điều chỉnh trị số p. Riêng cá nhân tôi thì thấy phương pháp Tukey thường rất hữu ích trong so sánh.

(b) Nếu $k>10$, phương pháp Bonferroni có thể trở nên rất "bảo thủ". Bảo thủ ở đây có nghĩa là phương pháp này rất ít khi nào tuyên bố một so sánh có ý nghĩa thống kê, dù trong thực tế là có thật! Trong trường hợp này, hai phương pháp Tukey, Holm và Scheffé có thể áp dụng.

Quay lại ví dụ trên, các trị số p trên đây là những trị số chưa được điều chỉnh cho so sánh nhiều lần. Trong chương về trị số p, tôi đã nói các trị số này phóng đại ý nghĩa thống kê, không phản ánh trị số p lúc ban đầu (tức 0.05). Để điều chỉnh cho nhiều so sánh, chúng ta phải sử dụng đến phương pháp điều chỉnh Bonferroni.

Chúng ta có thể dùng lệnh `pairwise.t.test` để có được tất cả các trị số p so sánh giữa ba nhóm như sau:

```
> pairwise.t.test(galactose, group, p.adj="bonferroni")

        Pairwise comparisons using t tests with pooled SD

data:  galactose and group

  1      2
2 0.6805 -
3 0.0012 0.0321

P value adjustment method: bonferroni
```

Kết quả trên cho thấy trị số p giữa nhóm 1 (Crohn) và viêm ruột kết là 0.6805 (tức không có ý nghĩa thống kê); giữa nhóm Crohn và đối chứng là 0.0012 (có ý nghĩa thống kê), và giữa nhóm viêm ruột kết và đối chứng là 0.0321 (tức cũng có ý nghĩa thống kê).

Một phương pháp điều chỉnh trị số p khác có tên là phương pháp Holm:

```
> pairwise.t.test(galactose, group)

        Pairwise comparisons using t tests with pooled SD

data:  galactose and group
```

```
  1      2
2 0.2268 -
3 0.0012 0.0214

P value adjustment method: holm
```

Kết quả này cũng không khác so với phương pháp Bonferroni.

Tất cả các phương pháp so sánh trên sử dụng một sai số chuẩn chung cho cả ba nhóm. Nếu chúng ta muốn sử dụng cho từng nhóm thì lệnh sau đây (`pool.sd=F`) sẽ đáp ứng yêu cầu đó:

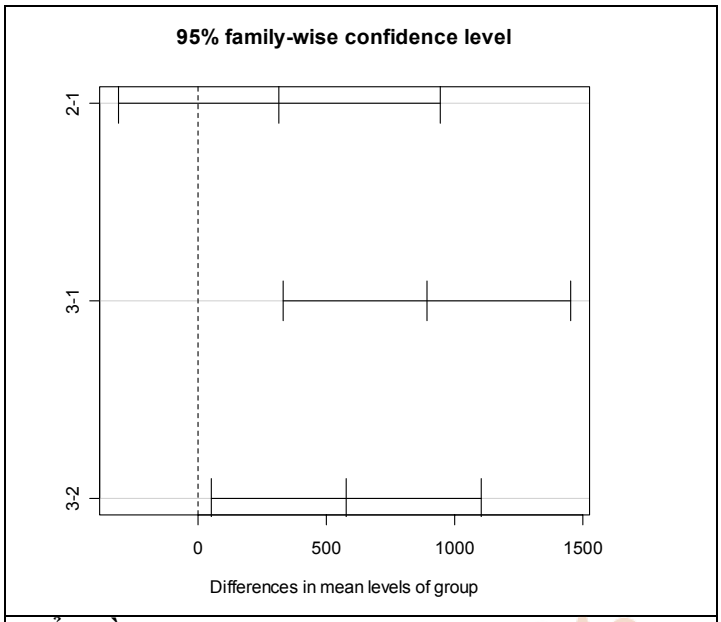
```
> pairwise.t.test(galactose, group, pool.sd=FALSE)

        Pairwise comparisons using t tests with non-pooled SD

data:  galactose and group

  1      2
2 0.2557 -
3 0.0017 0.0544

P value adjustment method: holm
```

Một lần nữa, kết quả này cũng không làm thay đổi kết luận.

Trong các phương pháp trên, chúng ta chỉ biết trị số p so sánh giữa các nhóm, nhưng không biết mức độ khác biệt cũng như khoảng tin cậy 95% giữa các nhóm. Để có những ước số này, chúng ta cần đến một hàm khác có tên là `aov` (viết tắt từ analysis of variance) và hàm `TukeyHSD` (HSD là viết tắt từ Honest Significant Difference, tạm dịch nôm na là "Khác biệt có ý nghĩa thành thật") như sau:

```
> res <- aov(galactose ~ group)
> TukeyHSD (res)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = galactose ~ group)

$group
        diff        lwr      upr     p adj
2-1 316.3232 -312.09857  944.745 0.4439821
3-1 894.2778  333.07916 1455.476 0.0011445
3-2 577.9545   53.11886 1102.790 0.0281768
```

Kết quả trên cho chúng ta thấy nhóm 3 và 1 khác nhau khoảng 894 đơn vị, và khoảng tin cậy 95% từ 333 đến 1455 đơn vị. Tương tự, galactose trong nhóm bệnh nhân viêm ruột kết thấp hơn nhóm đối chứng (nhóm 3) khoảng 578 đơn vị, và khoảng tin cậy 95% từ 53 đến 1103.



Biểu đồ 24. Trung bình hiệu và khoảng tin cậy 95% giữa nhóm 1 và 2, 1 và 3, và 3 và 2. Trục hoành là độ galactose, trục tung là ba so sánh.

11.3 Phân tích bằng phương pháp phi tham số

Phương pháp so sánh nhiều nhóm phi tham số (non-parametric statistics) tương đương với phương pháp phân tích phương sai là Kruskal-Wallis. Cũng như phương pháp Wilcoxon so sánh hai nhóm theo phương pháp phi tham số, phương pháp Kruskal-Wallis cũng biến đổi số liệu thành thứ bậc (ranks) và phân tích độ khác biệt thứ bậc này giữa các nhóm. Hàm `kruskal.test` trong R có thể giúp chúng ta trong kiểm định này:

```
> kruskal.test(galactose ~ group)

        Kruskal-Wallis rank sum test

data:  galactose by group
Kruskal-Wallis chi-squared = 12.1381, df = 2, p-value = 0.002313
```

Trị số p từ kiểm định này khá thấp (p = 0.002313) cho thấy có sự khác biệt giữa ba nhóm như phân tích phương sai qua hàm `lm` trên đây. Tuy nhiên, một bất tiện của kiểm định phi tham số Kruskal-Wallis là phương pháp này không cho chúng ta biết hai nhóm nào khác nhau, mà chỉ cho một trị số p chung. Trong nhiều trường hợp, phân tích phi tham số như kiểm định Kruskal-Wallis thường không có hiệu quả như các phương pháp thống kê tham số (parametric statistics).

11.4 Phân tích phương sai hai chiều (two-way analysis of variance - ANOVA)

Phân tích phương sai đơn giản hay một chiều chỉ có một yếu tố (factor). Nhưng phân tích phương sai hai chiều (two-way ANOVA), như tên gọi, có hai yếu tố. Phương pháp phân tích phương sai hai chiều chỉ đơn giản khai triển từ phương pháp phân tích phương sai đơn giản. Thay vì ước tính phương sai của một yếu tố, phương pháp phân sai hai chiều ước tính phương sai của hai yếu tố.

Ví dụ 18. Trong ví dụ sau đây, để đánh giá hiệu quả của một kĩ thuật sơn mới, các nhà nghiên cứu áp dụng sơn trên 3 loại vật liệu (1, 2 vả 3) trong hai điều kiện (1, 2). Mỗi điều kiện và loại vật liệu, nghiên cứu được lặp lại 3 lần. Độ bền được đo là chỉ số bền bĩ (tạm gọi là `score`). Tổng cộng, có 18 số liệu như sau:

Độ bền bĩ của sơn cho 2 điều kiện và 3 vật liệu

Điều kiện (*i*)	Vật liệu (*j*)		
	1	2	3
1	4.1, 3.9, 4.3	3.1, 2.8, 3.3	3.5, 3.2, 3.6
2	2.7, 3.1, 2.6	1.9, 2.2, 2.3	2.7, 2.3, 2.5

Gọi x_{ij} là `score` của điều kiện i (i = 1, 2) cho vật liệu j (j = 1, 2, 3). (Để đơn giản hóa vấn đề, chúng ta tạm thời bỏ qua k đối tượng). Mô hình phân tích phương sai hai chiều phát biểu rằng:

$$x_{ij} = \mu + \alpha_i + \beta_j + \varepsilon_{ij}$$

μ là số trung bình cho toàn quần thể, các hệ số α_i (ảnh hưởng của điều kiện i)và β_j (ảnh hưởng của vật liệu j) cần phải ước tính từ số liệu thực tế. ε_{ij} được giả định tuân theo luật phân phối chuẩn với trung bình 0 và phương sai σ^2.

Để phân tích bằng R, chúng ta cần phải tổ chức dữ liệu sao cho có 4 biến như sau:

```
Condition   Material    Đối tượng   Score
(điều kiện) (vật liệu)
1           1           1           4.1
1           1           2           3.9
1           1           3           4.3
1           2           4           3.1
1           2           5           2.8
1           2           6           3.3
1           3           7           3.5
1           3           8           3.2
1           3           9           3.6
2           1           10          2.7
2           1           11          3.1
2           1           12          2.6
2           2           13          1.9
2           2           14          2.2
2           2           15          2.3
```

```
2           3           16          2.7
2           3           17          2.3
2           3           18          2.5
```

Chúng ta có thể tạo ra một dãy số bằng cách sử dụng hàm `gl` (generating levels).

```
> condition <- gl(2, 9, 18)
> material  <- gl(3, 3, 18)
```

Và tạo nên 18 mã số (từ 1 đến 18):

```
> id <- 1:18
```

Sau cùng là số liệu cho `score`:

```
> score <- c(4.1,3.9,4.3, 3.1,2.8,3.3, 3.5,3.2,3.6,
           2.7,3.1,2.6, 1.9,2.2,2.3, 2.7,2.3,2.5)
```

Tất cả cho vào một dataframe tên là `data`:

```
> data <- data.frame(condition, material, id, score)
> attach(data)
```

Bây giờ số liệu đã sẵn sàng cho phân tích. Để phân tích phương sai hai chiều, chúng ta vẫn sử dụng lệnh `lm` với các thông số như sau:

```
> twoway <- lm(score ~ condition + material)
> anova(twoway)
Analysis of Variance Table

Response: score
          Df Sum Sq Mean Sq F value    Pr(>F)
condition  1 5.0139  5.0139  95.575 1.235e-07 ***
material   2 2.1811  1.0906  20.788 6.437e-05 ***
Residuals 14 0.7344  0.0525
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Ba nguồn dao động (variation) của `score` được phân tích trong bảng trên. Qua trung bình bình phương (mean square), chúng ta thấy ảnh hưởng của điều kiện có vẻ quan trọng hơn là ảnh hưởng của vật liệu thí nghiệm. Tuy nhiên, cả hai ảnh hưởng đều có ý nghĩa thống kê, vì trị số p rất thấp cho hai yếu tố. Chúng ta yêu cầu R tóm lược các ước số phân tích bằng lệnh `summary`:

```
> summary(twoway)

Call:
lm(formula = score ~ condition + material)

Residuals:
     Min       1Q   Median       3Q      Max
-0.32778 -0.16389  0.03333  0.16111  0.32222
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   3.9778     0.1080  36.841 2.43e-15 ***
condition2   -1.0556     0.1080  -9.776 1.24e-07 ***
material2    -0.8500     0.1322  -6.428 1.58e-05 ***
material3    -0.4833     0.1322  -3.655   0.0026 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.229 on 14 degrees of freedom
Multiple R-Squared: 0.9074,     Adjusted R-squared: 0.8875
F-statistic: 45.72 on 3 and 14 DF,  p-value: 1.761e-07
```

Kết quả trên cho thấy so với điều kiện 1, điều kiện 2 có `score` thấp hơn khoảng 1.056 và sai số chuẩn là 0.108, với trị số p = 1.24e-07, tức có ý nghĩa thống kê. Ngoài ra, so với vật liệu 1, `score` cho vật liệu 2 và 3 cũng thấp hơn đáng kể với độ thấp nhất ghi nhận ở vật liệu 2, và ảnh hưởng của vật liệu thí nghiệm cũng có ý nghĩa thống kê.

Giá trị có tên là "`Residual standard error`" được ước tính từ trung bình bình phương phần dư trong phần (a), tức là $\sqrt{0.0525} = 0.229$, tức là ước số của $\hat{\sigma}$.

Hệ số xác định bội (R^2) cho biết hai yếu tố điều kiện và vật liệu giải thích khoảng 91% độ dao động của toàn bộ mẫu. Hệ số này được tính từ tổng bình phương trong kết quả phần (a) như sau:

$$R^2 = \frac{5.0139 + 2.1811}{5.0139 + 2.1811 + 0.7344} = 0.9074$$

Và sau cùng, hệ số R^2 điều chỉnh phản ánh độ "cải tiến" của mô hình. Để hiểu hệ số này tốt hơn, chúng ta thấy phương sai của toàn bộ mẫu là s^2 = (5.0139 + 2.1811 + 0.7344) / 17 = 0.4644. Sau khi điều chỉnh cho ảnh hưởng của điều kiện và vật liệu, phương sai này còn 0.0525 (tức là residual mean square). Như vậy hai yếu tố này làm giảm phương sai khoảng 0.4644 – 0.0525 = 0.4119. Và hệ số R^2 điều chỉnh là:

$$\text{Adj } R^2 = 0.4119 / 0.4644 = 0.88$$

Tức là sau khi điều chỉnh cho hai yếu tố điều kiện và vật liệu phương sai của score giảm khoảng 88%.

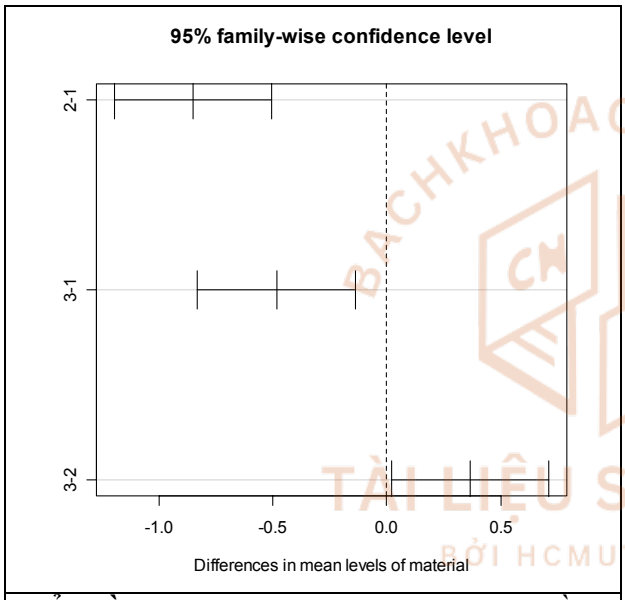
So sánh giữa các nhóm. Chúng ta sẽ ước tính độ khác biệt giữa hai điều kiện và ba vật liệu bằng hàm `TukeyHSD` với `aov`:

```
> res <- aov(score ~ condition+ material+condition)
> TukeyHSD(res)
  Tukey multiple comparisons of means
    95% family-wise confidence level
```

```
Fit: aov(formula = score ~ condition + material + condition)

$condition
         diff       lwr        upr p adj
2-1 -1.055556 -1.287131 -0.8239797 1e-07

$material
          diff         lwr        upr     p adj
2-1 -0.8500000 -1.19610279 -0.5038972 0.0000442
3-1 -0.4833333 -0.82943612 -0.1372305 0.0068648
3-2  0.3666667  0.02056388  0.7127695 0.0374069
```

Biểu đồ sau đây sẽ minh hoạ cho các kết quả trên:

```
> plot(TukeyHSD(res), ordered=TRUE)
There were 16 warnings (use warnings() to see them)
```



Biểu đồ 25. So sánh giữa 3 loại vật liệu bằng phương pháp Tukey.

12. Phân tích hồi qui logistic

Trong các phần trước về phân tích hồi qui tuyến tính và phân tích phương sai, chúng ta tìm mô hình và mối liên hệ giữa một biến phụ thuộc liên tục (continuous dependent variable) và một hay nhiều biến độc lập (independent variable) hoặc là liên tục hoặc là không liên tục. Nhưng trong nhiều trường hợp, biến phụ thuộc không phải là biến liên tục mà là biến mang tính đo lường nhị phân: có/không, mắc bệnh/không mắc bệnh, chết/sống, xảy ra/không xảy ra, v.v…, còn các biến độc lập có thể là liên tục hay không liên tục. Chúng ta cũng muốn tìm hiểu mối liên hệ giữa các biến độc lập và biến phụ thuộc.

Ví dụ 19. Trong một nghiên cứu do tôi tiến hành để tìm hiểu mối liên hệ giữa nguy cơ gãy xương (fracture, viết tắt là `fx`) và mật độ xương cùng một số chỉ số sinh hóa khác, 139 bệnh nhân nam (hay nói đúng hơn là đối tượng nghiên cứu) tuổi từ 60 trở lên. Năm 1990, các số liệu sau đây được thu thập cho mỗi đối tượng: độ tuổi (`age`), tỉ trọng cơ thể (body mass index hay `BMI`), mật độ chất khoáng trong xương (bone mineral density hay `BMD`), chỉ số hủy xương `ICTP`, chỉ số tạo xương `PINP`. Các đối tượng nghiên cứu được theo dõi trong vòng 15 năm. Trong thời gian theo dõi, các bệnh nhân bị gãy xương hay không gãy xương được ghi nhận. Câu hỏi đặt ra ban đầu là có một mối liên hệ gì giữa BMD và nguy cơ gãy xương hay không. Số liệu của nghiên cứu này được trình bày trong phần cuối của chương này, và sẽ trình bày một phần dưới đây để bạn đọc nắm được vấn đề.

Một phần số liệu nghiên cứu về các yếu tố nguy cơ cho gãy xương

id	fx	age	bmi	bmd	ictp	pinp
1	1	79	24.7252	0.818	9.170	37.383
2	1	89	25.9909	0.871	7.561	24.685
3	1	70	25.3934	1.358	5.347	40.620
4	1	88	23.2254	0.714	7.354	56.782
5	1	85	24.6097	0.748	6.760	58.358
6	0	68	25.0762	0.935	4.939	67.123
7	0	70	19.8839	1.040	4.321	26.399
8	0	69	25.0593	1.002	4.212	47.515
9	0	74	25.6544	0.987	5.605	26.132
10	0	79	19.9594	0.863	5.204	60.267
				...		
137	0	64	38.0762	1.086	5.043	32.835
138	1	80	23.3887	0.875	4.086	23.837
139	0	67	25.9455	0.983	4.328	71.334

Ở đây, vì biến phụ thuộc (gãy xương) không được đo lường theo tính liên tục (mà chỉ là *có* hay *không*), cho nên phương pháp phân tích hồi qui tuyến tính để phân tích mối liên hệ giữa biến phụ thuộc và biến độc lập. Một phương pháp phân tích được phát triển tương đối gần đây (vào thập niên 1970s) có tên là logistic regression analysis (hay phân tích hồi qui logistic) có thể áp dụng cho trường hợp trên.

Trong nghiên cứu này, sau 15 năm theo dõi, có 38 bệnh nhân bị gãy xương. Tính theo phần trăm, tỉ lệ gãy xương là 38 / 139 = 0.273 (hay 27.3%).

12.1 Mô hình hồi qui logistic

Cho một tần số biến cố x ghi nhận từ n đối tượng, chúng ta có thể tính xác suất của biến cố đó là:

$$p = \frac{x}{n}$$

p có thể xem là một chỉ số đo lường nguy cơ của một biến cố. Một cách thể hiện nguy cơ khác là *odds* (một danh từ, nếu tôi không lầm, chỉ có trong tiếng Anh – ngay cả tiếng Pháp, Đức, Tây Ban Nha … cũng không có danh từ tương đương với *odds*). Tôi tạm dịch

odds là *khả năng*. Khả năng của một biến cố được định nghĩa đơn giản bằng tỉ số xác suất biến cố xảy ra trên xác suất biến cố không xảy ra:

$$odds = \frac{p}{1-p}$$

Hàm *logit* của *odds* được định nghĩa như sau:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right)$$

Cho một biến độc lập x (x có thể là liên tục hay không liên tục), mô hình hồi qui logistic phát biểu rằng:

$$\text{logit}(p) = \alpha + \beta x$$

Tương tự như mô hình hồi qui tuyến tính, α và β là hai thông số tuyến tính cần phải ước tính từ dữ liệu nghiên cứu. Nhưng ý nghĩa của thông số này, đặc biệt là thông số β, rất khác với ý nghĩa mà ta đã quen với mô hình hồi qui tuyến tính. Để hiểu ý nghĩa của hai thông số này, tôi sẽ quay lại với ví dụ 19.

Vấn đề mà chúng ta muốn biết là mối liên hệ giữa mật độ xương `bmd` và nguy cơ gãy xương (`fx`). Để tiện cho việc minh họa, gọi `bmd` là x, vấn đề mà chúng ta cần biết có thể viết bằng ngôn ngữ mô hình như sau

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right)\alpha + \beta x$$

Nói cách khác:

$$odds(p) = \frac{p}{1-p} = e^{\alpha+\beta x}$$

Nói cách khác, mô hình hồi qui logistic vừa trình bày trên phát biểu rằng mối liên hệ giữa xác suất gãy xương (p) và mật độ xương `bmd` là một mối liên hệ theo hình chữ S. Mô hình trên còn cho thấy xác suất gãy xương p tùy thuộc vào giá trị của x. Thành ra, mô hình trên có thể viết một cách chính xác hơn rằng *khả năng* gãy xương với điều kiện x là:

$$odds(p \mid x) = e^{\alpha+\beta x}$$

Khi $x = x_0$, khả năng gãy xương là: $odds(p \mid x = x_0) = e^{\alpha+\beta x_0}$

Khi $x = x_0 + 1$ (tức tăng 1 đơn vị từ x_0), khả năng gãy xương là:

$$odds(p \mid x = x_0 + 1) = e^{\alpha+\beta(x_0+1)}$$

Và, tỉ số của hai xác suất gãy xương:

$$\frac{odds\left(p \mid x = x_0 + 1\right)}{odds\left(p \mid x = x_0\right)} = \frac{e^{\alpha+\beta(x_0+1)}}{e^{\alpha+\beta x_0}} = e^{\beta}$$

Trong dịch tễ học, e^{β} được gọi là *odds ratio*. *Odds ratio*, như tên gọi là, *tỉ số khả năng* hay *tỉ số khả dĩ*. Nói cách khác, hệ số β trong mô hình hồi qui logistic chính là tỉ số khả dĩ.

Phương pháp để ước tính thông số trong mô hình [3] khá phức tạp (dùng phương pháp maximum likelihood – tức phương pháp *Hợp lí cực đại*) và không nằm trong phạm vi của cuốn sách này, nên tôi sẽ không trình bày ở đây (bạn đọc có thể tham khảo sách giáo khoa để biết thêm, nếu cần thiết). Tuy nhiên, tôi muốn đề cập ngắn gọn là phương pháp hợp lí cực đại cung cấp cho chúng ta một hệ phương trình như sau:

$$\begin{cases} \sum_{i=1}^{n} y_i = \sum_{i=1}^{n} \left(1 + e^{-\left(\hat{\alpha}+\hat{\beta}x_i\right)}\right)^{-1} \\ \sum_{i=1}^{n} x_i y_i = \sum_{i=1}^{n} x_i \left(1 + e^{-\left(\hat{\alpha}+\hat{\beta}x_i\right)}\right) \end{cases}$$

Trong đó, Trong đó, y_i là biến phụ thuộc (gãy xương với giá trị 0 hay 1), và x_i là biến độc lập (mật độ xương), và n là số mẫu. Để tìm ước số $\hat{\alpha}$ và $\hat{\beta}$, một trong những phép tính hay sử dụng là iterative weighted least square hay Newton-Raphson. R sử dụng phép tính Newton-Raphson để tìm hai ước số đó.

Sau khi đã có ước số $\hat{\alpha}$ và $\hat{\beta}$ chúng ta có thể ước tính xác suất p cho bất cứ giá trị nào của x như sau (sau vài thao tác đại số):

$$\hat{p} = \frac{e^{\hat{\alpha}+\hat{\beta}x}}{1 + e^{\hat{\alpha}+\hat{\beta}x}} = \frac{1}{1 + e^{-\left(\hat{\alpha}+\hat{\beta}x\right)}}$$

Chú ý tôi dùng dấu mũ $\hat{p}$ để chỉ số ước tính (predicted value), chứ không phải p là xác suất quan sát. Nếu mô hình mô tả dữ liệu tốt và đầy đủ, độ khác biệt giữa p và $\hat{p}$ nhỏ; nếu mô hình không thích hợp hay không tốt, độ khác biệt đó có thể sẽ cao. Độ khác biệt giữa p và $\hat{p}$ được gọi là *deviance*. Phương pháp tính deviance khá phức tạp, nhưng đó không phải là chủ đề ở đây, cho nên tôi chỉ nói qua khái niệm mà thôi. Khi chúng ta có nhiều mô hình để mô phỏng một hay nhiều mối liên hệ, deviance có thể được sử dụng để đánh giá sự thích hợp của một mô hình, hay để chọn một mô hình "tối ưu".

12.2 Phân tích hồi qui logistic bằng R

Bây giờ, chúng ta quay lại với ví dụ 1, dùng số liệu trong Bảng 12.1 để ước tính hai thông số α và β bằng R. Trước hết chúng ta phải nhập toàn bộ số liệu vào một data

frame, và cho một cái tên, chẳng hạn như `fracture`. Trong trường hợp của tôi, dữ liệu được chứa trong directory c:\works\stats dưới tên `fracture.txt`, do đó, các lệnh sau đây cần thiết để nhập số liệu:

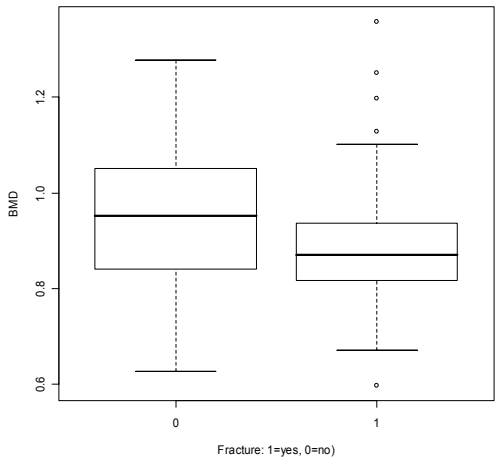
```
# báo cho R biết nơi chứa số liệu

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

# nhập số liệu và cho vào một data frame tên fracture

> fracture <- read.table("fracture.txt", header=TRUE, na.string=".")

# kiểm tra xem có bao nhiêu biến trong dữ liệu fracture

> names(fracture)
 [1] "id"   "fx"   "age"  "bmi"  "bmd"  "ictp" "pinp"

# Chọn những bệnh nhân có đầy đủ số liệu cho phân tích

> fulldata <- na.omit(fracture)
> attach(fulldata)
```

Hai biến mà chúng ta quan tâm trong ví dụ này là: `fx` (gãy xương) và `bmd` (mật độ xương). Chúng ta kiểm tra xem có bao nhiêu bệnh nhân gãy xương:

```
> table(fx)
fx
  0   1
101  38
```

Kế đến, xem mật độ xương trong nhóm gãy xương và không gãy xương ra sao:

```
> tapply(bmd, fx, mean)
        0         1
0.9444851 0.9016667

> boxplot(bmd ~ fx,
          xlab="Fracture: 1=yes, 0=no)",
          ylab="BMD")
```



Kết quả trên cho thấy, bmd trong nhóm bệnh nhân bị gãy xương thấp hơn so với nhóm không bị gãy xương (0.90 và 0.94). Và, kiểm định t sau đây cho thấy mức độ khác biệt này không có ý nghĩa thống kê (p = 0.15).

```
> t.test(bmd~fx)

        Welch Two Sample t-test

data:  bmd by fx
t = 1.4572, df = 53.952, p-value = 0.1508
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.01609226  0.10172922
sample estimates:
mean in group 0 mean in group 1
      0.9444851       0.9016667
```


Để ước tính thông số trong mô hình [4], hàm số glm (viết tắt từ *generalized linear model*) trong R có thể áp dụng, với "cú pháp" như sau:

```
> logistic <- glm(fx ~ bmd, family="binomial")
> summary(logistic)

Call:
glm(formula = fx ~ bmd, family = "binomial")

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0287  -0.8242  -0.7020   1.3780   2.0709

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.063      1.342   0.792    0.428
bmd           -2.270      1.455  -1.560    0.119

(Dispersion parameter for binomial family taken to be 1)
```

```
    Null deviance: 157.81  on 136  degrees of freedom
Residual deviance: 155.27  on 135  degrees of freedom
AIC: 159.27

Number of Fisher Scoring iterations: 4
```

Tôi sẽ lần lượt giải thích các kết quả trên:

(a) Trong lệnh `logistic <- glm(fx ~ bmd, family="binomial")` chúng ta yêu cầu R phân tích theo mô hình `fx` là một hàm số với `bmd` như mô hình [4]. Trong glm có nhiều luật phân phối, mà trong đó phân phối nhị phân (binomial) là một luật phân phối chuẩn cho hồi qui logistic. Do đó, `family="binomial"` cần thiết cho R.

(b) Deviance: phần thứ nhất của kết quả cho biết qua về deviance.

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.0287  -0.8242  -0.7020   1.3780   2.0709
```

Deviance như giải thích trên phản ánh độ khác biệt giữa mô hình và dữ liệu (cũng tương tự như mean square residual trong phân tích hồi qui tuyến tính vậy). Đối với một mô hình đơn lẻ như ví dụ này thì giá trị của deviance không có ý nghĩa gì nhiều.

(c) Phần kế tiếp cung cấp ước số của $\hat{\alpha}$ (mà R đặt tên là `intercept`) và $\hat{\beta}$ (`bmd`) và sai số chuẩn (standard error).

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.063      1.342   0.792    0.428
bmd           -2.270      1.455  -1.560    0.119
```

Qua kết quả này, chúng ta có $\hat{\alpha}$ = 1.063 và $\hat{\beta}$ = -2.27. Ước số $\hat{\beta}$ là số âm cho thấy mối liên hệ giữa nguy cơ gãy xương và `bmd` là mối liên hệ nghịch đảo: xác suất gãy xương tăng khi giá trị của `bmd` giảm. Tuy nhiên, kiểm định z (tính bằng cách lấy ước số chia cho sai số chuẩn) cho chúng ta thấy ảnh hưởng của `bmd` không có ý nghĩa thống kê, vì trị số p = 0.119.

Nhớ rằng tỉ số khả dĩ (odds ratio hay viết tắt là OR) chính là $e^{-2.27}$ = 0.1033. Nói cách khác, khi `bmd` tăng 1 g/cm^2 (đơn vị đo lường của bmd là g/cm^2) thì tỉ số OR giảm 0.9067 hay 90.67%. Nhưng tăng 1 g/cm^2 là mật độ rất cao trong xương và không thực tế. Cho nên một cách tính khác là tính trên độ lệch chuẩn (standard deviation) của `bmd`. Chúng ta sẽ tìm hiểu độ lệch chuẩn của `bmd`:

```
> sd(bmd)
[1] 0.1406543
```

Do đó, OR sẽ tính trên mỗi 0.14 g/cm^2. Và OR cho mỗi độ lệch chuẩn, do đó, là:

$$e^{-2.27*0.1406} = 0.7267$$

Tức là, khi `bmd` tăng một độ lệch chuẩn thì tỉ số khả dĩ gãy xương giảm khoảng 28%. Cũng có thể nói cách khác, là khi `bmd` *giảm* một độ lệch chuẩn thì tỉ số khả dĩ tăng $e^{2.27*0.1406} = 1.376$ hay khoảng 38%.

Một cách khác để biết ảnh hưởng của `bmd` là ước tính xác suất gãy xương qua phương trình:

$$\hat{p} = \frac{e^{1.063-2.27(bmd)}}{1+e^{1.063-2.27(bmd)}}$$

Theo đó, khi `bmd` = 1.00, p = 0.23. Khi `bmd` = 0.86 (tức giảm 1 độ lệch chuẩn), p = 0.291. Tức là, nếu BMD giảm 1 độ lệch chuẩn thì xác suất gãy xương tăng 0.291/0.23 = 1.265 hay 26%5.

(d) Phần cuối của kết quả cung cấp deviance cho hai mô hình: mô hình không có biến độc lập (`null deviance`), và mô hình với biến độc lập, tức là `bmd` trong ví dụ (`residual deviance`).

```
    Null deviance: 157.81  on 136  degrees of freedom
Residual deviance: 155.27  on 135  degrees of freedom
AIC: 159.27
```

Qua hai số này, chúng ta thấy `bmd` ảnh hưởng rất thấp đến việc tiên đoán gãy xương, chỉ làm giảm deviance từ 157.8 xuống còn 155.27, và mức độ giảm này không có ý nghĩa thống kê.

Ngoài ra, R còn cung cấp giá trị của AIC (Akaike Information Criterion) được tính từ deviance và bậc tự do. Tôi sẽ quay lại ý nghĩa của AIC trong phần sắp đến khi so sánh các mô hình.

12.3 Ước tính xác suất bằng R

Xin nhắc lại trong phân tích trên, chúng ta cho các kết quả vào đối tượng `logistic`. Trong đối tượng này có nhiều thông tin có ích, nhưng nếu muốn xem các thông tin này chúng ta phải dùng đến các lệnh như `summary` chẳng hạn. Trong phần này, tôi sẽ trình bày một vài hàm để xem xét các thông tin liên quan đến việc tiên đoán xác suất.

- `predict` dùng để liệt kê các giá trị ước tính (predicted values) của mô hình $\log\left(\frac{p}{1-p}\right) = \alpha + \beta x$ cho từng bệnh nhân.

```
> predict(logistic)
```

```
         1          2          3          4          5          6
```

```
 2.377576584  1.085694014 -2.141117756  1.492824115  0.965379946 -0.941253280
           7            8            9           10           11           12
-1.733686514 -1.675645430 -0.665282957 -0.507046129 -0.941854868 -0.648740461
...
```

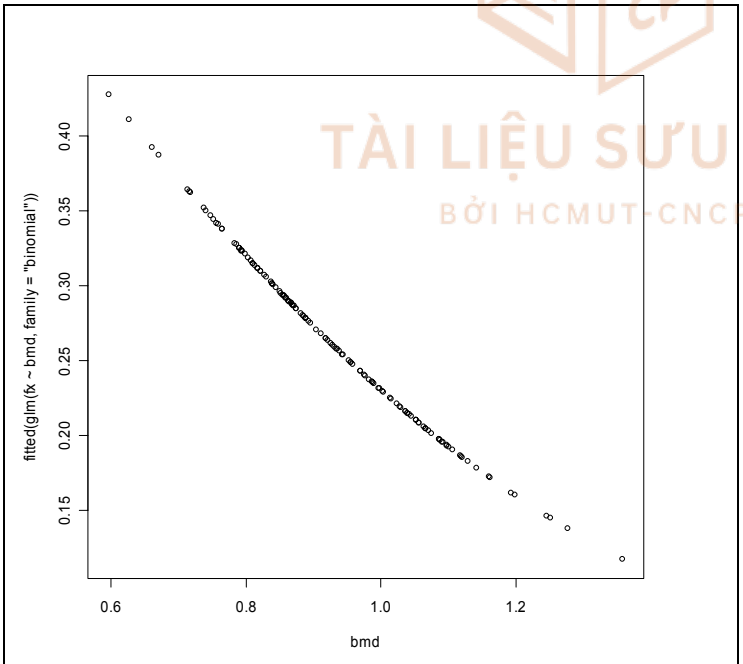
Các số trên là log(p / (1 – p)), tức *log odds*, không có ý nghĩa hực tế bao nhiêu. Chúng ta muốn biết giá trị tiên đoán xác suất p tính từ phương trình $\hat{p} = \frac{e^{1.063-2.27(bmd)}}{1+e^{1.063-2.27(bmd)}}$. Để có giá trị này cho từng bệnh nhân, chúng ta cho thông số `type="response"` vào hàm `predict` như sau:

```
> predict(logistic, type="response")
         1          2          3          4          5          6          7
0.91510135 0.74757001 0.10516416 0.81650178 0.72419767 0.28064726 0.15011664
         8          9         10         11         12         13         14
0.15767295 0.33955387 0.37588624 0.28052582 0.34327343 0.44305196 0.23830776
...
```

Trong kết quả trên (chỉ in một phần) ước tính xác suất gãy xương cho bệnh nhân 1 là 0.915, cho bệnh nhân 2 là 0.747, v.v…

- Chúng ta có thể xem xét các giá trị tiên đoán này với độ `bmd` bằng cách dùng hàm `plot` thông thường:

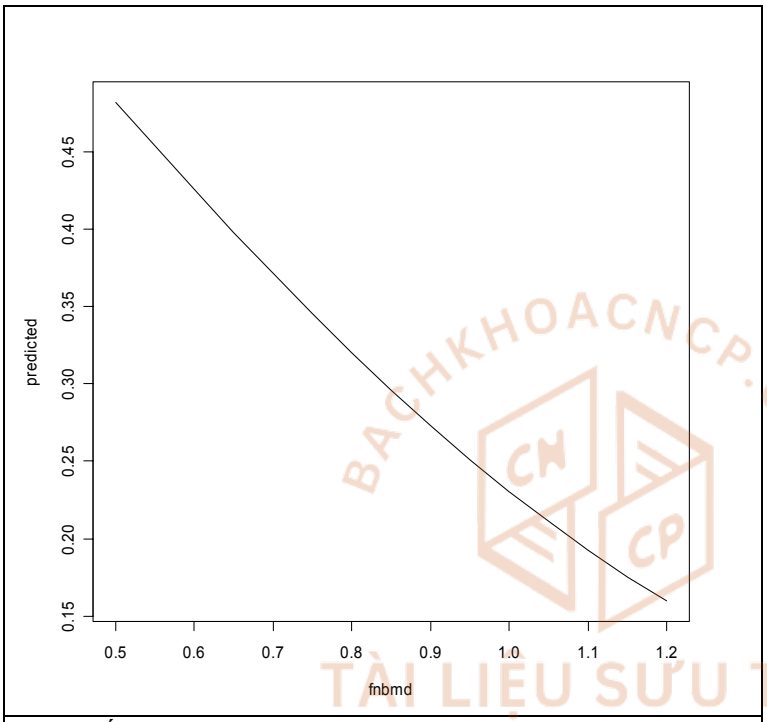
```
> plot(bmd, fitted(glm(fx ~ bmd, family="binomial")))
```



Xác suất tiên đoán gãy xương (trục tung) và độ `bmd` (trục hoành) qua mô hình hồi qui logistic.

Biểu đồ trên có thể cải tiến bằng cách cho các khoảng cách giá trị bmd gần nhau hơn (như 0.50, 0.55, 0.60, …, 1.20 chẳng hạn), và dùng đường biểu diễn thay vì dùng dấu chấm. Các lệnh sau đây sẽ cải tiến biểu đồ.

```
> logistic <- glm(fx ~ bmd, family="binomial")
> fnbmd <- seq(0.5, 1.2, 0.05)  #cho fnbmd từ > 0.50,0.55,0.6,...,1.2
> new.data <- data.frame(bmd = fnbmd) #cho vào một dataframe mới
> predicted <- predict(logistic, new.data, type="response")
> plot(predicted ~ fnbmd, type="l")
```



Xác suất tiên đoán gãy xương (trục tung) và độ bmd (trục hoành) qua mô hình hồi qui logistic.

13. Ước tính cỡ mẫu (sample size estimation)

Một công trình nghiên cứu thường dựa vào một mẫu (sample). Một trong những câu hỏi quan trọng nhất trước khi tiến hành nghiên cứu là cần bao nhiêu mẫu hay bao nhiêu đối tượng cho nghiên cứu. "Đối tượng" ở đây là đơn vị căn bản của một nghiên cứu, là số bệnh nhân, số tình nguyện viên, số mẫu ruộng, cây trồng, thiết bị, v.v… Ước tính số lượng đối tượng cần thiết cho một công trình nghiên cứu đóng vai trò cực kì quan trọng, vì nó có thể là yếu tố quyết định sự thành công hay thất bại của nghiên cứu. Nếu số lượng đối tượng không đủ thì kết luận rút ra từ công trình nghiên cứu không có độ chính xác cao, thậm chí không thể kết luận gì được. Ngược lại, nếu số lượng đối tượng quá nhiều hơn số cần thiết thì tài nguyên, tiền bạc và thời gian sẽ bị hao phí. Do đó, vấn đề then chốt trước khi nghiên cứu là phải ước tính cho được một số đối tượng vừa đủ cho mục tiêu của nghiên cứu. Số lượng đối tượng "vừa đủ" tùy thuộc vào ba yếu tố chính:

- Sai sót mà nhà nghiên cứu chấp nhận, cụ thể là sai sót loại I và II;
- Độ dao động (variability) của đo lường, mà cụ thể là độ lệch chuẩn; và
- Mức độ khác biệt hay ảnh hưởng mà nhà nghiên cứu muốn phát hiện.

Không có số liệu về ba yếu tố này thì không thể nào ước tính cỡ mẫu. Kinh nghiệm của người viết cho thấy rất nhiều người khi tiến hành nghiên cứu thường không có ý niệm gì về các số liệu này, cho nên khi đến tham vấn các chuyên gia về thống kê học, họ chỉ nhận câu trả lời: "không thể tính được"! Trong chương này tôi sẽ bàn qua ba yếu tố trên.

13.1 Khái niệm về "power"

Thống kê học là một phương pháp khoa học có mục đích phát hiện, hay đi tìm những cái có thể gộp chung lại bằng cụm từ "chưa được biết" (unknown). Cái chưa được biết ở đây là những hiện tượng chúng ta không quan sát được, hay quan sát được nhưng không đầy đủ. "Cái chưa biết" có thể là một ẩn số (như chiều cao trung bình ở người Việt Nam, hay trọng lượng một phần tử), hiệu quả của một thuật điều trị, gen có chức năng làm cho cây lá có màu xanh, sở thích của con người, v.v… Chúng ta có thể đo chiều cao, hay tiến hành xét nghiệm để biết hiệu quả của thuốc, nhưng các nghiên cứu như thế chỉ được tiến hành trên một nhóm đối tượng, chứ không phải toàn bộ quần thể của dân số.

Ở mức độ đơn giản nhất, những cái chưa biết này có thể xuất hiện dưới hai hình thức: hoặc là có, hoặc là không. Chẳng hạn như một thuật điều trị có hay không có hiệu quả chống gãy xương, khách hàng thích hay không thích một loại nước giải khát. Bởi vì không ai biết hiện tượng một cách đầy đủ, chúng ta phải đặt ra giả thiết. Giả thiết đơn giản nhất là *giả thiết đảo* (hiện tượng không tồn tại, kí hiệu H-) và *giả thiết chính* (hiện tượng tồn tại, kí hiệu H+).

Chúng ta sử dụng các phương pháp kiểm định thống kê (statistical test) như kiểm định t, F, z, χ^2, v.v… để đánh giá khả năng của giả thiết. Kết quả của một kiểm định thống kê có thể đơn giản chia thành hai giá trị: hoặc là *có ý nghĩa thống kê* (statistical significance), hoặc là *không có ý nghĩa thống kê* (non-significance). Có ý nghĩa thống kê ở đây, như đề cập trong Chương 7, thường dựa vào trị số P: nếu $P < 0.05$, chúng ta phát biểu kết quả có ý nghĩa thống kê; nếu $P > 0.05$ chúng ta nói kết quả không có ý nghĩa thống kê. Cũng có thể xem có ý nghĩa thống kê hay không có ý nghĩa thống kê như là có tín hiệu hay không có tín hiệu. Hãy tạm đặt kí hiệu T+ là kết quả có ý nghĩa thống kê, và T- là kết quả kiểm định không có ý nghĩa thống kê.

Hãy xem xét một ví dụ cụ thể: để biết thuốc risedronate có hiệu quả hay không trong việc điều trị loãng xương, chúng ta tiến hành một nghiên cứu gồm 2 nhóm bệnh nhân (một nhóm được điều trị bằng risedronate và một nhóm chỉ sử dụng giả dược placebo). Chúng ta theo dõi và thu thập số liệu gãy xương, ước tính tỉ lệ gãy xương cho từng nhóm, và so sánh hai tỉ lệ bằng một kiểm định thống kê. Kết quả kiểm định thống kê hoặc là *có ý nghĩa thống kê* ($P<0.05$) hay không có ý nghĩa thống kê ($P>0.05$). Xin nhắc lại rằng chúng ta không biết risedronate thật sự có hiệu nghiệm chống gãy xương

hay không; chúng ta chỉ có thể đặt giả thiết H. Do đó, khi xem xét một giả thiết và kết quả kiểm định thống kê, chúng ta có bốn tình huống:

(a) Giả thuyết H đúng (thuốc risedronate có hiệu nghiệm) và kết quả kiểm định thống kê P<0.05.

(b) Giả thuyết H đúng, nhưng kết quả kiểm định thống kê không có ý nghĩa thống kê;

(c) Giả thuyết H sai (thuốc risedronate không có hiệu nghiệm) nhưng kết quả kiểm định thống kê có ý nghĩa thống kê;

(d) Giả thuyết H sai và kết quả kiểm định thống kê không có ý nghĩa thống kê.

Ở đây, trường hợp (a) và (d) không có vấn đề, vì kết quả kiểm định thống kê nhất quán với thực tế của hiện tượng. Nhưng trong trường hợp (b) và (c), chúng ta phạm sai lầm, vì kết quả kiểm định thống kê không phù hợp với giả thiết. Trong ngôn ngữ thống kê học, chúng ta có vài thuật ngữ:

- xác suất của tình huống (b) xảy ra được gọi là *sai sót loại II* (type II error), và thường kí hiệu bằng β.
- xác suất của tình huống (a) được gọi là *Power*. Nói cách khác, *power* chính là xác suất mà kết quả kiểm định thống cho ra kết quả p<0.05 với điều kiện giả thiết H là thật. Nói cách khác: *power* = $1-\beta$;
- xác suất của tình huống (c) được gọi là *sai sót loại I* (type I error, hay significance level), và thường kí hiệu bằng α. Nói cách khác, α chính là xác suất mà kết quả kiểm định thống cho ra kết quả p<0.05 với điều kiện giả thiết H sai;
- xác suất tình hống (d) không phải là vấn đề cần quan tâm, nên không có thuật ngữ, dù có thể gọi đó là kết quả *âm tính thật* (hay true negative).

Có thể tóm lược 4 tình huống đó trong một Bảng 1 sau đây:

Các tình huống trong việc thử nghiệm một giả thiết khoa học

Kết quả kiểm định thống kê	**Giả thuyết H**	
	Đúng (thuốc có hiệu nghiệm)	**Sai** (thuốc không có hiệu nghiệm)
Có ý nghĩa thống kê (p<0,05)	Dương tính thật (**power**), 1-β= P(s | H+)	Sai sót loại I (**type I error**) α = P(s | H-)
Không có ý nghĩa thống kê (p>0,05)	Sai sót loại II (**type II error**) β = P(ns | H+)	Âm tính thật (**true negative**) 1-α = P(ns | H-)

Chú thích: *s* trong biểu đồ này có nghĩa là significant; *ns* non-significant; *H+* là giả thuyết đúng; và *H-* là giả thuyết sai. Do đó, có thể mô tả 4 tình huống trên bằng ngôn ngữ xác suất có điều kiện như sau: Power $= 1 - \beta = P(s \mid H+)$; $\beta = P(ns \mid H+)$; và $\alpha = P(s \mid H-)$.

13.2 Số liệu để ước tính cỡ mẫu

Như đã đề cập trong phần đầu của chương này, để ước tính số đối tượng cần thiết cho một công trình nghiên cứu, chúng ta cần phải có 3 số liệu: xác suất sai sót loại I và II, độ dao động của đo lường, và độ ảnh hưởng.

- Về xác suất sai sót, thông thường một nghiên cứu chấp nhận sai sót loại I khoảng 1% hay 5% (tức $\alpha = 0.01$ hay 0.05), và xác suất sai sót loại II khoảng $\beta = 0.1$ đến $\beta = 0.2$ (tức power phải từ 0.8 đến 0.9).

- Độ dao động chính là độc lệch chuẩn (standard deviation) của đo lường mà công trình nghiên cứu dựa vào để phân tích. Chẳng hạn như nếu nghiên cứu về cao huyết áp, thì nhà nghiên cứu cần phải có độ lệch chuẩn của áp suất máu. Chúng ta tạm gọi độ dao động là σ.

- Độ ảnh hưởng, nếu là công trình nghiên cứu so sánh hai nhóm, là độ khác biệt trung bình giữa hai nhóm mà nhà nghiên cứu muốn phát hiện. Chẳng hạn như nhà nghiên cứu có thể giả thiết rằng bệnh nhân được điều trị bằng thuốc A có áp suất máu giảm 10 mmHg so với nhóm giả được. Ở đây, 10 mmHg được xem là độ ảnh hưởng. Chúng ta tạm gọi độ ảnh hưởng là Δ.

Một nghiên cứu có thể có một nhóm đối tượng hay hai (và có khi hơn 2) nhóm đối tượng. Và ước tính cỡ mẫu cũng tùy thuộc vào các trường hợp này.

Trong trường hợp một nhóm đối tượng, số lượng đối tượng (n) cần thiết cho nghiên cứu có thể tính toán một cách "thủ công" như sau:

$$n = \frac{C}{(\Delta/\sigma)^2}$$

Trong trường hợp có hai nhóm đối tượng, số lượng đối tượng (n) cần thiết cho nghiên cứu có thể tính toán như sau:

$$n = 2 \times \frac{C}{(\Delta/\sigma)^2}$$

Trong đó, hằng số C được xác định từ xác suất sai sót loại I và II (hay power) như sau:

Hằng số C liên quan đến sai sót loại I và II

α =	β = 0.20 (Power = 0.80)	β = 0.10 (Power = 0.90)	β = 0.05 (Power = 0.95)
0.10	6.15	8.53	10.79
0.05	7.85	10.51	13.00
0.01	13.33	16.74	19.84

13.4 Ước tính cỡ mẫu

13.4.1 Ước tính cỡ mẫu cho một chỉ số trung bình

Ví dụ 20: Chúng ta muốn ước tính chiều cao ở đàn ông người Việt, và chấp nhận sai số trong vòng 1 cm (d = 1) với khoảng tin cậy 0.95 (tức α=0.05) và power = 0.8 (hay β = 0.2). Các nghiên cứu trước cho biết độ lệch chuẩn chiều cao ở người Việt khoảng 4.6 cm. Chúng ta có thể áp dụng công thức [1] để ước tính cỡ mẫu cần thiết cho nghiên cứu:

$$n = \frac{C}{(\Delta/\sigma)^2} = \frac{7.85}{(1/4.6)^2} = 166$$

Nói cách khác, chúng ta cần phải đo chiều cao ở 166 đối tượng để ước tính chiều cao đàn ông Việt với sai số trong vòng 1 cm.

Nếu sai số chấp nhận là 0.5 cm (thay vì 1 cm), số lượng đối tượng cần thiết là: $n = \frac{7.85}{(0.5/4.6)^2} = 664$. Nếu độ sai số mà chúng ta chấp nhận là 0.1 cm thì số lượng đối tượng nghiên cứu lên đến 16610 người! Qua các ước tính này, chúng ta dễ dàng thấy cỡ mẫu tùy thuộc rất lớn vào độ sai số mà chúng ta chấp nhận. Muốn có ước tính càng chính xác, chúng ta cần càng nhiều đối tượng nghiên cứu.

Trong R có hàm `power.t.test` có thể áp dụng để ước tính cỡ mẫu cho ví dụ trên như sau. Chú ý chúng ta cho R biết vấn đề là một nhóm tức `type="one.sample"`:

```
# sai số 1 cm, độc lệch chuẩn 4.6, a=0.05, power=0.8
> power.t.test(delta=1, sd=4.6, sig.level=.05, power=.80,
               type='one.sample')

     One-sample t test power calculation

              n = 168.0131
          delta = 1
             sd = 4.6
      sig.level = 0.05
          power = 0.8
    alternative = two.sided
```

kết quả tính toán từ R là 168, khác với cách tính thủ công 2 đối tượng, vì cố nhiên R sử dụng nhiều số lẻ hơn và chính xác hơn cách tính thủ công. Với sai số 0.5 cm:

```
# sai số 0.5 cm, độc lệch chuẩn 4.6, a=0.05, power=0.8
> power.t.test(delta=0.5, sd=4.6, sig.level=.05, power=.80,
               type='one.sample')

  One-sample t test power calculation

              n = 666.2525
          delta = 0.5
             sd = 4.6
      sig.level = 0.05
          power = 0.8
    alternative = two.sided
```

Ví dụ 21: Một loại thuốc điều trị có khả năng tăng độ alkaline phosphatase ở bệnh nhân loãng xương. Độ lệch chuẩn của alkaline phosphatase là 15 U/l. Một nghiên cứu mới sẽ tiến hành trong một quần thể bệnh nhân ở Việt Nam, và các nhà nghiên cứu muốn biết bao nhiêu bệnh nhân cần tuyển để chứng minh rằng thuốc có thể alkaline phosphatase từ 60 đến 65 U/l sau 3 tháng điều trị, với sai số I α = 0.05 và power = 0.8.

Đây là một loại nghiên cứu "trước – sau" (before-after study); có nghĩa là trước và sau khi điều trị. Ở đây, chúng ta chỉ có một nhóm bệnh nhân, nhưng được đo hai lần (trước khi dùng thuốc và sau khi dùng thuốc). Chỉ tiêu lâm sàng để đánh giá hiệu nghiệm của thuốc là độ thay đổi về alkaline phosphatase. Trong trường hợp này, chúng ta có trị số tăng trung bình là 5 U/l và độ lệch chuẩn là 15 U/l, hay nói theo ngôn ngữ R, `delta=5, sd=15, sig.level=.05, power=.80,` và lệnh:


```
> power.t.test(delta=3, sd=15, sig.level=.05, power=.80,
               type='one.sample')

     One-sample t test power calculation

              n = 198.1513
          delta = 3
             sd = 15
      sig.level = 0.05
          power = 0.8
    alternative = two.sided
```

Như vậy, chúng ta cần phải có 198 bệnh nhân để đạt các mục tiêu trên.

13.4.2 Ước tính cỡ mẫu cho so sánh hai số trung bình

Trong thực tế, rất nhiều nghiên cứu nhằm so sánh hai nhóm với nhau. Cách ước tính cỡ mẫu cho các nghiên cứu này chủ yếu dựa vào công thức [2] như trình bày phần 15.3.1.

Ví dụ 22: Một nghiên cứu được thiết kế để thử nghiệm thuốc alendronate trong việc điều trị loãng xương ở phụ nữ sau thời kì mãn kinh. Có hai nhóm bệnh nhân được tuyển: nhóm 1 là nhóm can thiệp (được điều trị bằng alendronate), và nhóm 2 là nhóm đối chứng (tức không được điều trị). Tiêu chí để đánh giá hiệu quả của thuốc là mật độ xương (bone mineral density – BMD). Số liệu từ nghiên cứu dịch tễ học cho thấy giá trị trung bình của BMD trong phụ nữ sau thời kì mãn kinh là 0.80 g/cm^2, với độ lệch chuẩn là 0.12 g/cm^2. Vấn đề đặt ra là chúng ta cần phải nghiên cứu ở bao nhiêu đối tượng để "chứng minh" rằng sau 12 tháng điều trị BMD của nhóm 1 tăng khoảng 5% so với nhóm 2?

Trong ví dụ trên, tạm gọi trị số trung bình của nhóm 2 là μ_2 và nhóm 1 là μ_1, chúng ta có: μ_1 = 0.8*1.05 = 0.84 g/cm^2 (tức tăng 5% so với nhóm 1), và do đó, Δ = 0.84 – 0.80 = 0.04 g/cm^2. Độ lệch chuẩn là σ = 0.12 g/cm^2. Với power = 0.90 và α = 0.05, cỡ mẫu cần thiết là:

$$n = \frac{2C}{(\Delta/\sigma)^2} = \frac{2\times 10.51}{(0.04/0.12)^2} = 189$$

Và lời giải từ R qua hàm `power.t.test` như sau:

```
> power.t.test(delta=0.04, sd=0.12, sig.level=0.05, power=0.90,
               type="two.sample")

     Two-sample t test power calculation

              n = 190.0991
          delta = 0.04
             sd = 0.12
      sig.level = 0.05
          power = 0.9
    alternative = two.sided

 NOTE: n is number in *each* group
```

Chú ý trong hàm `power.t.test`, ngoài các thông số thông thường như `delta` (độ ảnh hưởng hay khác biệt theo giả thiết), `sd` (độ lệch chuẩn), `sig.level` xác suất sai sót loại I, và `power`, chúng ta còn phải cụ thể chỉ ra rằng đây là nghiên cứu gồm có hai nhóm với thông số `type="two.sample"`.

Kết quả trên cho biết chúng ta cần 190 bệnh nhân **cho mỗi nhóm** (hay 380 bệnh nhân cho công trình nghiên cứu). Trong trường hợp này, power = 0.90 và α = 0.05 có nghĩa là gì ? Trả lời: hai thông số đó có nghĩa là nếu chúng ta tiến hành thật nhiều nghiên cứu (ví dụ 1000) và mỗi nghiên cứu với 380 bệnh nhân, sẽ có 90% (hay 900) nghiên cứu sẽ cho ra kết quả trên với trị số p < 0.05.

13.4.3 Ước tính cỡ mẫu cho phân tích phương sai

Phương pháp ước tính cỡ mẫu cho so sánh giữa hai nhóm cũng có thể khai triển thêm để ước tính cỡ mẫu cho trường hợp so sánh hơn hai nhóm. Trong trường hợp có nhiều nhóm, như đề cập trong Chương 11, phương pháp so sánh là phân tích phương sai. Theo phương pháp này, số trung bình bình phương phần dư (residual mean square, RMS) chính là ước tính của độ dao động của đo lường trong mỗi nhóm, và chỉ số này rất quan trọng trong việc ước tính cỡ mẫu.

Chi tiết về lí thuyết đằng sau cách ước tính cỡ mẫu cho phân tích phương sai khá phức tạp, và không nằm trong phạm vi của chương này. Nhưng nguyên lí chủ yếu vẫn không khác so với lí thuyết so sánh giữa hai nhóm. Gọi số trung bình của k nhóm là μ_1, μ_2, μ_3, . . ., μ_k, chúng ta có thể tính tổng bình phương giữa các nhóm bằng SS $SS = \sum_{i=1}^{k}(\mu_i - \bar{\mu})^2$, trong đó, $\bar{\mu} = \sum_{i=1}^{k}\mu_i / k$. Cho $\lambda = \frac{SS}{(k-1)RMS}$, vấn đề đặt ra là tìm cố lượng cỡ mẫu n sao cho z_β đáp ứng yêu cầu power = 0.80 hay 0.9, mà

$$z_\beta = \frac{1}{\sqrt{(k-1)(1+n\lambda)F + k(n-1)(1+2n\lambda)}} \times \left(\sqrt{k(n-1)\left[2(k-1)(1+n\lambda)^2 - (1|2n\lambda)\right]} - \sqrt{F(k-1)(1+n\lambda)(2k(n-1)-1)}\right)$$

Trong đó F là kiểm định F. (Xem J. Fleiss, "The Design and Analysis of Clinical Experiments", John Wiley & Sons, New York 1986, trang 373).

Ví dụ 23. Để so sánh độ ngọt của một loại nước uống giữa 4 nhóm đối tượng khác nhau về giới tính và độ tuổi (tạm gọi 4 nhóm là A, B, C và D), các nhà nghiên cứu giả thiết rằng độ ngọt trong nhóm A, B. C và D lần lược là 4.5, 3.0, 5.6, và 1.3. Qua xem xét nhiều nghiên cứu trước, các nhà nghiên cứu còn biết rằng RMS về độ ngọt trong mỗi nhóm là khoảng 8.7. Vấn đề đặt ra là bao nhiêu đối tượng cần nghiên cứu để phát hiện sự khác biệt có ý nghĩa thống kê ở mức độ $\alpha = 0.05$ và power = 0.9.

Hàm `power.anova.test` trong R có thể ứng dụng để giải quyết vấn đề. Chúng ta chỉ cần đơn giản cung cấp 4 số trung bình theo giả thiết và số RMS như sau:

```
# trước hết cho 4 số trung bình vào một vector
> groupmeans <- c(4.5, 3.0, 5.6, 1.3)

# sau đó, "gọi" hàm power.anova.test:
> power.anova.test(groups = length(groupmeans),
                   between.var=var(groupmeans),
                   within.var=8.7, power=0.90, sig.level=0.05)

     Balanced one-way analysis of variance power calculation

         groups = 4
```

```
           n = 12.81152
between.var = 3.486667
 within.var = 8.7
  sig.level = 0.05
      power = 0.9

NOTE: n is number in each group
```

Kết quả cho thấy các nhà nghiên cứu cần khoảng 13 đối tượng cho mỗi nhóm (tức 52 đối tượng cho toàn bộ nghiên cứu).

13.4.4 Ước tính cỡ mẫu để ước tính một tỉ lệ

Nhiều nghiên cứu mô tả có mục đích khá đơn giản là ước tính một tỉ lệ. Chẳng hạn như giới y tế thường hay tìm hiểu tỉ lệ một bệnh trong cộng đồng, hay giới thăm dò ý kiến và thị trường thường tìm hiểu tỉ lệ dân số ưa thích một sản phẩm. Trong các trường hợp này, chúng ta không có những đo lường mang tính liên tục, nhưng kết quả chỉ là những giá trị nhị như có / không, thích / không tích, v.v… Và cách ước tính cỡ mẫu cũng khác với ba ví dụ trên đây.

Năm 1991, một cuộc thăm dò ý kiến ở Mĩ cho thấy 45% người được hỏi sẵn sàng khuyến khích con họ nên hiến một quả thận cho những bệnh nhân cần thiết. Khoảng tin cậy 95% của tỉ lệ này là 42% đến 48%, tức một khoảng cách đến 6%! Kết quả này [tương đối] thiếu chính xác, dù số lượng đối tượng tham gia lên đến 1000 người. Tại sao? Để trả lời câu hỏi này, chúng ta thử xem qua một vài lí thuyết về ước tính cỡ mẫu cho một tỉ lệ.

Chúng ta biết qua Chương 6 và 9 rằng nếu $\hat{p}$ được ước tính từ n đối tượng, thì khoảng tin cậy 95% của một tỉ lệ p [trong dân số] là: $\hat{p} \pm 1.96 \times SE(\hat{p})$, trong đó $SE(\hat{p}) = \sqrt{\hat{p}(1-\hat{p})/n}$.

Bây giờ thử lật ngược vấn đề: chúng ta muốn ước tính p sao khoảng rộng $2 \times 1.96 \times SE(\hat{p})$ không quá một hằng số m. Nói cách khác, chúng ta muốn:

$$1.96 \times \sqrt{\hat{p}(1-\hat{p})/n} \le m$$

Chúng ta muốn tìm số lượng đối tượng n để đạt yêu câu trên. Qua cách diễn đạt trên, dễ dàng thấy rằng:

$$n \ge \left(\frac{1.96}{m}\right)^2 \hat{p}(1-\hat{p})$$

Do đó, số lượng cỡ mẫu tùy thuộc vào độ sai số m và tỉ lệ p mà chúng ta muốn ước tính. Độ sai số càng thấp, số lượng cở mẫu càng cao.

Ví dụ 24: Chúng ta muốn ước tính tỉ lệ đàn ông hút thuốc ở Việt Nam, sao cho ước số không cao hơn hay thấp hơn 2% so với tỉ lệ thật trong toàn dân số. Một nghiên cứu trước cho thấy tỉ lệ hút thuốc trong đàn ông người Việt có thể lên đến 70%. Câu hỏi đặt ra là chúng ta cần nghiên cứu trên bao nhiêu đàn ông để đạt yêu cầu trên.

Trong ví dụ này, chúng ta có sai số $m = 0.02$, $\hat{p} = 0.70$, và số lượng cỡ mẫu cần thiết cho nghiên cứu là:

$$n \geq \left(\frac{1.96}{0.02}\right)^2 0.7 \times 0.3$$

Nói cách khác, chúng ta cần nghiên cứu ít nhất là 2017.

Nếu chúng ta muốn giảm sai số từ 2% xuống 1% (tức $m = 0.01$) thì số lượng đối tượng sẽ là 8067! Chỉ cần thêm độ chính xác 1%, số lượng mẫu có thể thêm hơn 6000 người. Do đó, vấn đề ước tính cỡ mẫu phải rất thận trọng, xem xét cân bằng giữa độ chính xác thông tin cần thu thập và chi phí.

R không có hàm cho ước tính cỡ mẫu cho một tỉ lệ, nhưng với công thức trên, bạn đọc có thể viết một hàm để tính rất dễ dàng.

13.4.5 Ước tính cỡ mẫu cho so sánh hai tỉ lệ

Nhiều nghiên cứu mang tính suy luận thường có hai [hay nhiều hơn hai] nhóm để so sánh. Trong phần 15.4.2 chúng ta đã làm quen với phương pháp ước tính cỡ mẫu để so sánh hai số trung bình bằng kiểm định t. Đó là những người cứu mà tiêu chí là những biến số liên tục. Nhưng có nghiên cứu biến số không liên tục mà mang tính nhị phân như tôi vừa bàn trong phần 15.4.3. Để so sánh hai tỉ lệ, phương pháp kiểm định thông dụng nhất là kiểm định nhị phân (binomial test) hay Chi bình phương (χ^2 test). Trong phần này, tôi sẽ bàn qua cách tính cỡ mẫu cho hai loại kiểm định thống kê này.

Gọi hai tỉ lệ [mà chúng ta không biết nhưng muốn tìm hiểu] là p_1 và p_2, và gọi $\Delta = p_1 - p_2$. Giả thiết mà chúng ta muốn kiểm định là $\Delta = 0$. Lí thuyết đằng sau để ước tính cỡ mẫu cho kiểm định giả thiết này khá rườm rà, nhưng có thể tóm gọn bằng công thức sau đây:

$$n = \frac{\left(z_{\alpha/2}\sqrt{2\bar{p}(1-\bar{p})} + z_{\beta}\sqrt{p_1(1-p_1)+p_2(1-p_2)}\right)^2}{\Delta^2}$$

Trong đó, $\bar{p} = (p_1 + p_2)/2$, $z_{\alpha/2}$ là trị số z của phân phối chuẩn cho xác suất $\alpha/2$ (chẳng hạn như khi $\alpha = 0.05$, thì $z_{\alpha/2} = 1.96$; khi $\alpha = 0.01$, thì $z_{\alpha/2} = 2.57$), và z_{β} là trị số z của

phân phối chuẩn cho xác suất β (chẳng hạn như khi β = 0.10, thì z_β = 1.28; khi β = 0.20, thì z_β = 0.84).

Ví dụ 25: Một thử nghiệm lâm sàng đối chứng ngẫu nhiên được thiết kế để đánh giá hiệu quả của một loại thuốc chống gãy xương sống. Hai nhóm bệnh nhân sẽ được tuyển. Nhóm 1 được điều trị bằng thuốc, và nhóm 2 là nhóm đối chứng (không được điều trị). Các nhà nghiên cứu giả thiết rằng tỉ lệ gãy xương trong nhóm 2 là khoảng 10%, và thuốc có thể làm giảm tỉ lệ này xuống khoảng 6%. Nếu các nhà nghiên cứu muốn thử nghiệm giả thiết này với sai sót I là α = 0.01 và power = 0.90, bao nhiêu bệnh nhân cần phải được tuyển mộ cho nghiên cứu?

Ở đây, chúng ta có Δ = 0.10 – 0.06 = 0.04, và $\bar{p}$ = (0.10 + 0.06)/2 = 0.08. Với α = 0.01, $z_{\alpha/2}$ = 2.57 và với power = 0.90, z_β = 1.28. Do đó, số lượng bệnh nhân cần thiết cho mỗi nhóm là:

$$n = \frac{\left(2.57\sqrt{2\times 0.08\times 0.92} + 1.28\sqrt{0.1\times 0.90 + 0.06\times 0.94}\right)^2}{(0.04)^2} = 1361$$

Như vậy, công trình nghiên cứu này cần phải tuyển ít nhất là 2722 bệnh nhân để kiểm định giả thiết trên.

Hàm `power.prop.test` R có thể ứng dụng để tính cỡ mẫu cho trường hợp trên. Hàm `power.prop.test` cần những thông tin như `power`, `sig.level`, `p1`, và `p2`. Trong ví dụ trên, chúng ta có thể viết:

```
> power.prop.test(p1=0.10, p2=0.06, power=0.90, sig.level=0.01)

     Two-sample comparison of proportions power calculation

              n = 1366.430
             p1 = 0.1
             p2 = 0.06
      sig.level = 0.01
          power = 0.9
    alternative = two.sided

 NOTE: n is number in *each* group
```

Chú ý kết quả từ R có phần chính xác hơn (1366 đối tượng cho mỗi nhóm) vì R dùng nhiều số lẽ cho tính toán hơn là tính "thủ công".

Trước khi rời chương này, tôi muốn nhân cơ hội này để nhấn mạnh một lần nữa, ước tính cỡ mẫu cho nghiên cứu là một bước cực kì quan trọng trong việc thiết kế một nghiên cứu cho có ý nghĩa khoa học, vì nó có thể quyết định thành bại của nghiên cứu. Trước khi ước tính cỡ mẫu nhà nghiên cứu cần phải biết trước (hay ít ra là có vài giả thiết *cụ thể*) về vấn đề mình quan tâm. Ước tính cỡ mẫu cần một số thông số như đề cập đến

trong phần đầu của chương, và nếu các thông số này không có thì không thể ước tính được. Trong trường hợp một nghiên cứu hoàn toàn mới, tức chưa ai từng làm trước đó, có thể các thông số về độ ảnh hưởng và độ dao động đo lường sẽ không có, và nhà nghiên cứu cần phải tiến hành một số mô phỏng (simulation) hay một nghiên cứu sơ khởi để có những thông số cần thiết. Cách ước tính cỡ mẫu bằng mô phỏng là một lĩnh vực nghiên cứu khá chuyên sâu, không nằm trong đề tài của sách này, nhưng bạn đọc có thể tìm hiểu thêm phương pháp này trong các sách giáo khoa về thống kê học cấp cao hơn.

Trên đây là vài hướng dẫn nhanh để bạn đọc có thể sử dụng R cho phân tích số liệu và tạo biểu đồ. Bài viết này thực chất là tóm lược từ cuốn *Phân tích số liệu và tạo biểu đồ bằng* R: *hướng dẫn và thực hành,* do Nhà xuất bản Đại học Quốc gia Thành phố Hồ Chí Minh ấn hành vào năm 2006. Chi tiết về lí thuyết và một số phương pháp khác như phân tích sự kiện, xây dựng mô hình thống kê, mô phỏng, lập chương, v.v… có thể tìm trong sách trên.

14. Tài liệu tham khảo

Hiện nay, thư viện sách về R còn tương đối khiêm tốn so với thư viện cho các phần mềm thương mại như SAS và SPSS. Tuy nhiên, trong thời đại tiến bộ phi thường về thông tin internet và toàn cầu hóa như hiện nay, sách in và sách xuất bản trên website không còn là những khác nhau bao xa. Phần lớn chỉ dẫn về cách sử dụng R có thể tìm thấy rải rác đây đó trên các website từ các trường đại học và website cá nhân trên khắp thế giới. Trong phần này tôi chỉ liệt kê một số sách mà bạn đọc, nếu cần tham khảo thêm, nên tìm đọc. Trong quá trình viết cuốn sách mà bạn đọc đang cầm trên tay, tôi cũng tham khảo một số sách và trang web mà tôi sẽ liệt kê sau đây với vài lời nhận xét cá nhân.

Tài liệu tham khảo chính về R là bài báo của hai người sáng tạo ra R: Ihaka R, Gentleman R. R: A language for data analysis and graphics. *Journal of Computational and Graphical Statistics* 1996; 5:299-314.

- "**Data Analysis and Graphics Using R – An Example Approach**" (Nhà xuất bản Cambridge University Press, 2003) của John Maindonald nay đã xuất in lại lần thứ 2 với thêm một tác giả mới John Braun. Đây là cuốn sách rất có ích cho những ai muốn tìm hiểu và học về R. Năm chương đầu của sách viết cho bạn đọc chưa từng biết về R, còn các chương sau thì viết cho các bạn đọc đã biết cách sử dụng R thành thạo.

- "**Introductory Statistics With R**" (Nhà xuất bản Springer, 2004) của Peter Dalgaard là một cuốn sách loại căn bản cho R nhắm vào bạn đọc chưa biết gì về R. Sách tương đối ngắn (chỉ khoảng 200 trang) nhưng khá đắt giá!

- **"Linear Models with R"** (Nhà xuất bản Chapman & Hall/CRC, 2004) của Julian Faraway. Sách hiện có thể tải từ internet xuống miễn phí tại website sau đây: http://www.stat.lsa.umich.edu/~faraway/book/pra.pdf hay http://cran.r-project.org/doc/contrib/Faraway-PRA.pdf. Tài liệu dài 213 trang.

- **"R Graphics (Computer Science and Data Analysis)"** (Nhà xuất bản Chapman & Hall/CRC, 2005) của Paul Murrell. Đây là cuốn sách chuyên về phân tích biểu đồ bằng R. Sách có rất nhiều mã để bạn đọc có thể tự mình thiết kế các biểu đồ phức tạp và … màu mè.

- "**Modern Applied Statistics with S-Plus**" (Nhà xuất bản Springer, 4th Edition, 2003) của W. N. Venables và B. D. Ripley được viết cho ngôn ngữ S-Plus nhưng tất cả các lệnh và mã trong sách này đều có thể áp dụng cho R mà không cần thay đổi. (S-Plus là tiền thân của R, nhưng S-Plus là một phần mềm thương mại, còn R thì hoàn toàn miễn phí!) Đây là cuốn sách có thể nói là cuốn sách tham khảo cho tất cả ai muốn phát triển thêm về R. Hai tác giả cũng là những chuyên gia có thẩm quyền về ngôn ngữ R. Sách dành cho bạn đọc với trình độ cao về máy tính và thống kê học.

Các website quan trọng hay có ích về R

- Rất nhiều tài liệu tham khảo có thể tải từ website chính thức của R sau đây: http://cran.R-project.org/other-docs.html

  Trong đó có một số tài liệu quan trọng như "**An Introduction to R**" của W. N. Venables và B. D. Ripley.
  Địa chỉ internet: http://cran.r-project.org/doc/manuals/R-intro.pdf.

- Vài tài liệu hướng dẫn cách sử dụng R có thể tải (miễn phí) và tham khảo như sau:

  "**R for Beginners**" (57 trang) của Emmanuel Paradis. Tài liệu được soạn cho bạn đọc mới làm quen với R.
  Địa chỉ internet: http://cran.r-project.org/doc/contrib/Paradis-rdebuts_en.pdf.

  "**Using R for Data Analysis and Graphics**: Introduction, Code and Commentary" (35 trang) của John Maindonald là một tóm lược các lệnh và hàm căn bản của R cho phân tích số liệu và biểu đồ. Chủ đề của tài liệu này rất gần với cuốn sách mà bạn đang đọc.
  Địa chỉ internet: http://cran.r-project.org/doc/contrib/usingR.pdf

  "**Statistical Analysis with R – a quick start"** (46 trang) của Oleg Nenadic và Walter Zucchini. Web. Tài liệu hướng dẫn cách ứng dụng R cho phân tích thống kê và biểu đồ.
  Địa chỉ internet: http://www.statoek.wiso.uni-goettingen.de/mitarbeiter/ogi/pub/r_workshop.pdf

  "**A Brief Guide to R for Beginners in Econometrics**" (31 trang) của M. Arai. Tài liệu chủ yếu soạn cho giới phân tích thống kê kinh tế.
  Địa chỉ internet: http://people.su.se/~ma/R_intro

  "**Notes on the use of R for psychology experiments and questionnaires"** (39 trag) của Jonathan Baron và Yuelin Li. Web. Tài liệu được soạn cho giới nghiên cứu tâm lí học và xã hội học. Có ví dụ về log-linear model và một số mô hình phân tích phương sai trong tâm lí học.
  Địa chỉ internet: http://www.psych.upenn.edu/~baron/rpsych/rpsych.html

- StatsRus gồm một sưu tập về các mẹo để sử dụng R hữu hiệu hơn (dài khoảng 80 trang). Địa chỉ internet: http://lark.cc.ukans.edu/pauljohn/R/statsRus.html

- Và sau cùng là một tải liệu "**Hướng dẫn sử dụng R cho phân tích số liệu và biểu đồ**" (khoảng 50 trang – thường xuyên cập nhật hóa) do chính tôi viết bằng tiếng Việt. Website: www.R.ykhoa.net thực chất là tóm lược một số chương chính của cuốn sách này. Trang web này còn có tất cả các dữ liệu (datasets) và các mã sử trong trong sách để bạn đọc có thể tải xuống máy tính cá nhân để sử dụng.

15. Thuật ngữ dùng trong sách

Tiếng Anh	Tiếng Việt
95% confidence interval	Khoảng tin cậy 95%
Akaike Information criterion (AIC)	Tiêu chuẩn thông tin Akaike
Analysis of covariance	Phân tích hiệp biến
Analysis of variance (ANOVA)	Phân tích phương sai
Bar chart	Biểu đồ thanh
Binomial distribution	Phân phối nhị phân
Box plot	Biểu đồ hình hộp
Categorical variable	Biến thứ bậc
Clock chart	Biểu đồ đồng hồ
Coefficient of correlation	Hệ số tương quan
Coefficient of determination	Hệ số xác định bội
Coefficient of heterogeneity	Hệ số bất đồng nhất
Combination	Tổ hợp
Continuous variable	Biến liên tục
Correlation	Tương quan
Covariance	Hợp biến
Cross-over experiment	Thí nghiệm giao chéo
Cumulative probability distribution	Hàm phân phối tích lũy
Degree of freedom	Bậc tự do
Determinant	Định thức
Discrete variable	Biến rời rạc
Dot chart	Biểu đồ điểm
Estimate	Ước số
Estimator	Hàm ước lượng thống kê
Factorial analysis of variance	Phân tích phương sai cho thí nghiệm giai thừa
Fixed effects	Ảnh hưởng bất biến
Frequency	Tần số
Function	Hàm
Heterogeneity	Bất đồng nhất
Histogram	Biểu đồ tần số
Homogeneity	Đồng nhất
Hypothesis test	Kiểm định giả thiết
Inverse matrix	Ma trận nghịch đảo
Latin square experiment	Thí nghiệm hình vuông Latin
Least squares method	Phương pháp bình phương nhỏ nhất
Linear Logistic regression analysis	Phân tích hồi qui tuyến tính logistic
Linear regression analysis	Phân tích hồi qui tuyến tính

Matrix	Ma trận
Maximum likelihood method	Phương pháp hợp lí cực đại
Mean	Số trung bình
Median	Số trung vị
Meta-analysis	Phân tích tổng hợp
Missing value	Giá trị không
Model	Mô hình
Multiple linear regression analysis	Phân tích hồi qui tuyến tính đa biến
Normal distribution	Phân phối chuẩn
Object	Đối tượng
Parameter	Thông số
Permutation	Hoán vị
Pie chart	Biểu đồ hình tròn
Poisson distribution	Phân phối Poisson
Polynomial regression	Hồi qui đa thức
Probability	Xác suất
Probability density distribution	Hàm mật độ xác suất
P-value	Trị số P
Quantile	Hàm định bậc
Random effects	Ảnh hưởng ngẫu nhiên
Random variable	Biến ngẫu nhiên
Relative risk	Tỉ số nguy cơ tương đối
Repeated measure experiment	Thí nghiệm tái đo lường
Residual	Phần dư
Residual mean square	Trung bình bình phương phần dư
Residual sum of squares	Tổng bình phương phần dư
Scalar matrix	Ma trận vô hướng
Scatter plot	Biểu đồ tán xạ
Significance	Có ý nghĩa thống kê
Simulation	Mô phỏng
Standard deviation	Độ lệch chuẩn
Standard error	Sai số chuẩn
Standardized normal distribution	Phân phối chuẩn chuẩn hóa
Survival analysis	Phân tích biến cố
Traposed matrix	Ma trận chuyển vị
Variable	Biến (biến số)
Variance	Phương sai
Weight	Trọng số
Weighted mean	Trung bình trọng số