12
Phân tích hồi qui logistic

Trong các chương 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 1. 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.

Bảng 12.1. 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:

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

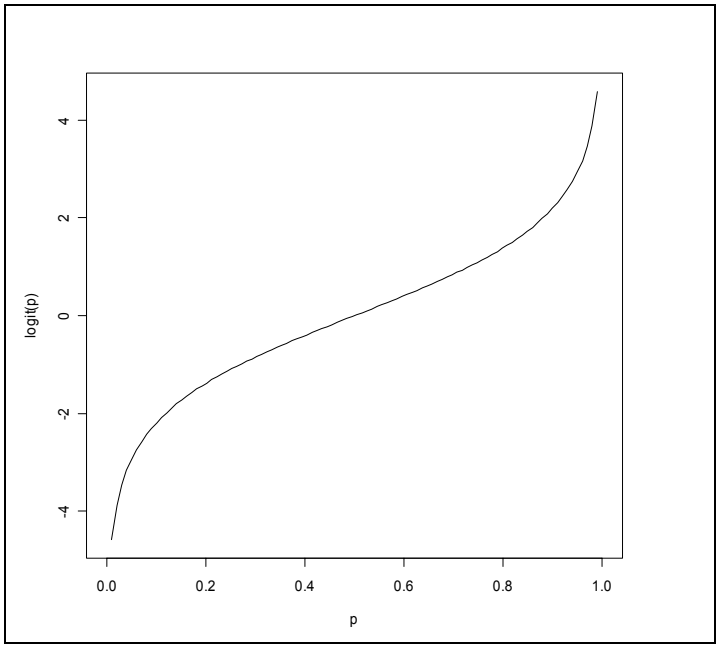
p có thể xem là một chỉ số đo lường nguy cơ của một biến cố. Một cách thể hiện nguy cơ khác là *odds* (một danh từ, nếu tôi không lầm, chỉ có trong tiếng Anh – ngay cả tiếng Pháp, Đức, Tây Ban Nha … cũng không có danh từ tương đương với *odds*). Tôi tạm dịch *odds* là *khả năng*. Khả năng của một biến cố được định nghĩa đơn giản bằng tỉ số xác suất biến cố xảy ra trên xác suất biến cố không xảy ra:

$$odds = \frac{p}{1-p} \qquad [1]$$

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

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) \qquad [2]$$

Mối liên hệ giữa *p* và logit(*p*) là một mối liên hệ liên tục (dĩ nhiên!) và theo dạng như sau:



Biểu đồ 12.1. Mối liên hệ giữa logit(p) và p, cho $1<p<0$.

Chú ý: biểu đồ trên được vẽ bằng các lệnh sau đây:

```
p <- seq(0, 1, length=100)
p <- p[2:(length(p)-1)]
logit <- function(t)
  {
   log(t / (1-t))
  }
plot(logit(p) ~ p, type="l")
```

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

$$\text{logit}(p) = \alpha + \beta x \qquad [3]$$

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 1.

Ví dụ 1 (tiếp theo). 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

$$\text{logit}\left(p\right) = \log\left(\frac{p}{1-p}\right)\alpha + \beta x \qquad [4]$$

Nói cách khác:

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

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:

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

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

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

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

Và, tỉ số của hai xác suất gãy xương:

$$\frac{odds\left(p \mid x = x_0 + 1\right)}{odds\left(p \mid x = x_0\right)} = \frac{e^{\alpha+\beta(x_0+1)}}{e^{\alpha+\beta x_0}} = e^{\beta} \quad [5]$$

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

Ví dụ 1 (tiếp theo). 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:

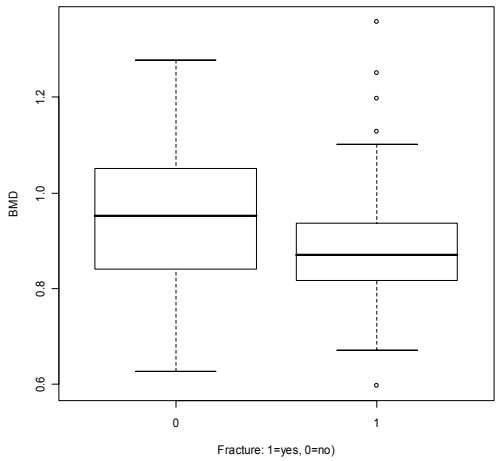
```
# báo cho R biết nơi chứa số liệu

> 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
...
```

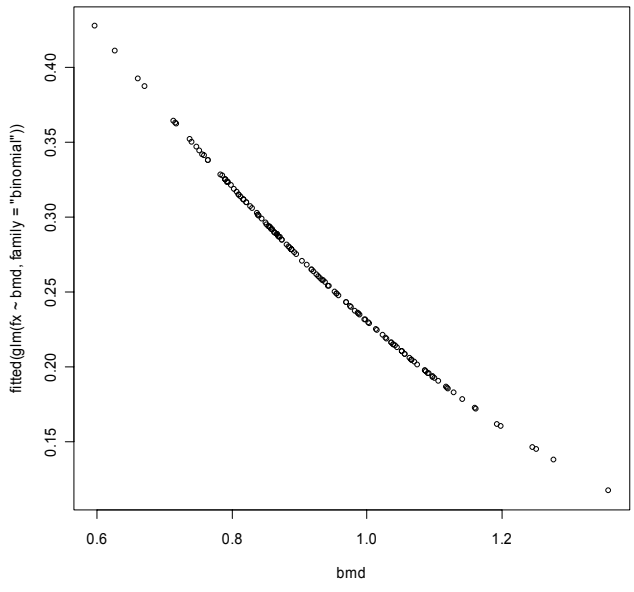
Các số trên là log(p / (1 – p)), tức *log odds*, không có ý nghĩa hực tế bao nhiêu. Chúng ta muốn biết giá trị tiên đoán xác suất p tính từ phương trình $\hat{p} = \frac{e^{1.063-2.27(bmd)}}{1+e^{1.063-2.27(bmd)}}$. Để có giá trị này cho từng bệnh nhân, chúng ta cho thông số `type="response"` vào hàm `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:

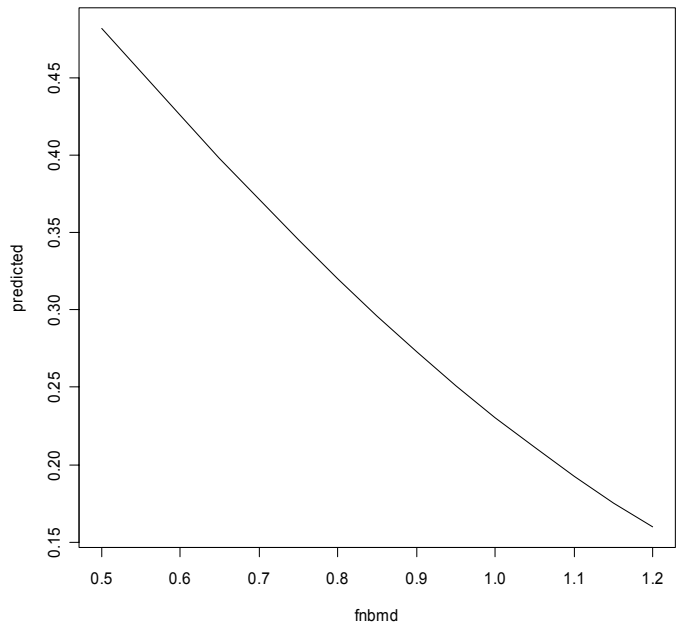
```
> plot(bmd, fitted(glm(fx ~ bmd, family="binomial")))
```



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.

12.4 Phân tích hồi qui logistic từ số liệu giản lược bằng R

Trong quá trình phân tích số liệu vừa trình bày trên đây, chúng ta có số liệu cho từng bệnh nhân và các biến độc lập đều là biến liên tục. Nhưng trong nhiều trường hợp biến độc lập là bậc thứ (và bởi vì biến phụ thuộc chỉ có hai giá trị 0 và 1) cho nên trên lí thuyết chúng ta có thể tóm lược dữ liệu bằng các bảng tần số (frequency table).

Ví dụ 2. Trong một nghiên cứu về ảnh hưởng của thói quen hút thuốc lá, tình trạng béo phì, thở ngáy (trong khi ngủ) đến nguy cơ bệnh cao huyết áp, các nhà nghiên cứu tóm lược số liệu như sau (số liệu trích từ Altman, trang 353):

smoking	obesity	snoring	ntotal	nhyper
0	0	0	60	5
1	0	0	17	2

0	1	0	8	1
1	1	0	2	0
0	0	1	187	35
1	0	1	85	13
0	1	1	51	15
1	1	1	23	8
Tổng số			433	79

Bảng 12.2. Tóm lược số liệu liên quan đến hút thuốc lá (`smoking`), béo phì (`obesity`), ngáy (`snoring`), và cao huyết áp. `ntotal` là tổng số bệnh nhân cho từng nhóm, và `nhyper` là số bệnh nhân trong tổng số bị bệnh cao huyết áp. Các biến số `smoking`, `obesity`, và `snoring` có giá trị `0=no` và `1=yes`.

Trong nghiên cứu có 433 bệnh nhân, và trong số này 79 người (hay 18%) bị bệnh cao huyết áp. Tuy nhiên, tỉ lệ này dao động khá cao theo từng nhóm bệnh nhân. Chẳng hạn như trong nhóm không hút thuốc lá(`smoking=0`), không béo phì (`obesity=0`) và không ngáy (`snoring=0`), tỉ lệ cao huyết áp là 8.3% (5/60). Trong khi đó nhóm với 3 yếu tố nguy cơ trên (`smoking=1`, `obesity=1`, `snoring=0`) thì có hơn 1 phần 3 hay 35% (8/23) bị bệnh cao huyết áp.

Để phân tích mối liên hệ giữa 3 yếu tố nguy cơ đó và bệnh cao huyết áp, trước hết cần phải cho số liệu vào R theo đúng như bảng số liệu trên.

```
> noyes <- c("no", "yes")  #định nghĩa biến noyes có 2 giá trị
> smoking <- gl(2,1,8, noyes) #biến smoking
> obesity <- gl(2,2,8, noyes) #biến obesity
> snoring <- gl(2,4,8, noyes) #biến snoring
> ntotal <- c(60, 17, 8, 2, 187, 85, 51, 23)
> nhyper <- c(5, 2, 1, 0, 35, 13, 15, 8)
> data <- data.frame(smoking, obesity, snoring, ntotal, nhyper)
> data
  smoking obesity snoring ntotal nhyper
1      no      no      no     60      5
2     yes      no      no     17      2
3      no     yes      no      8      1
4     yes     yes      no      2      0
5      no      no     yes    187     35
6     yes      no     yes     85     13
7      no     yes     yes     51     15
8     yes     yes     yes     23      8
```

Bây giờ chúng ta có thể sử dụng hàm `glm` để phân tích số liệu. Trước hết, chúng ta phải tạo thêm một biến số `proportion` như sau:

```
> proportion <- nhyper/ntotal
> logistic <- glm(proportion ~ smoking+obesity+snoring,
                  family="binomial",
                  weight=ntotal)
```

Chú ý trong hàm glm trên, chúng ta mô phỏng `proportion` như là một hàm số của `smoking`, `obesity` và `snoring`, vẫn với phân phối nhị phân (`binomial`), nhưng

có thêm một thông số `weight=ntotal`. Thông số `weight` yêu cầu R sử dụng `ntotal` là một số tóm lược (thay vì một bệnh nhân). Bây giờ, chúng ta có thể xem qua kết quả phân tích:

```
> summary(logistic)

Call:
glm(formula = proportion ~ smoking + obesity + snoring, family = "binomial",
    weights = ntotal)

Deviance Residuals:
       1         2         3         4         5         6         7         8
-0.04344   0.54145  -0.25476  -0.80051   0.19759  -0.46602  -0.21262   0.56231

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.37766    0.38018  -6.254    4e-10 ***
smokingyes  -0.06777    0.27812  -0.244   0.8075
obesityyes   0.69531    0.28509   2.439   0.0147 *
snoringyes   0.87194    0.39757   2.193   0.0283 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 14.1259  on 7  degrees of freedom
Residual deviance:  1.6184  on 4  degrees of freedom
AIC: 34.537

Number of Fisher Scoring iterations: 4
```

Kết quả trên cho thấy biến `smoking` không có ý nghĩa thống kê; cho nên có lẽ chúng ta nên bỏ biến này ra ngoài mô hình và có một mô hình đơn giản hơn:

```
> logistic <- glm(proportion ~ obesity+snoring,
                  family=”binomial”,
                  weight=ntotal)

> summary(logistic)

Call:
glm(formula = proportion ~ obesity + snoring, family = "binomial",
    weights = ntotal)

Deviance Residuals:
       1         2         3         4         5         6         7         8
-0.01247   0.47756  -0.24050  -0.82050   0.30794  -0.62742  -0.14449   0.45770

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -2.3921     0.3757  -6.366 1.94e-10 ***
obesityyes    0.6954     0.2851   2.440   0.0147 *
snoringyes    0.8655     0.3967   2.182   0.0291 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 14.1259  on 7  degrees of freedom
Residual deviance:  1.6781  on 5  degrees of freedom
```

```
AIC: 32.597

Number of Fisher Scoring iterations: 4
```

Phân tích phương sai trên deviance sau đây cũng khẳng định `obesity` và `snoring` là hai biến có ảnh hưởng đến cao huyết áp:

```
> anova(logistic, test="Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: proportion

Terms added sequentially (first to last)


        Df Deviance Resid. Df Resid. Dev P(>|Chi|)
NULL                        7    14.1259
obesity  1   6.8260         6     7.2999    0.0090
snoring  1   5.6218         5     1.6781    0.0177
```

12.5 Phân tích hồi qui logistic đa biến và chọn mô hình

Một trong những vấn đề khó khăn và có khi khá nan giải trong việc phân tích hồi qui logistic đa biến là chọn một mô hình để có thể mô tả đầy đủ dữ liệu. Một nghiên cứu với một biến phụ thuộc y và 3 biến độc lập x_1, x_2 và x_3, chúng ta có thể có những mô hình sau đây để tiên đoán y: $y = f(x_1)$, $y = f(x_2)$, $y = f(x_3)$, $y = f(x_1, x_2)$, $y = f(x_1, x_3)$, $y = f(x_2, x_3)$, và $y = f(x_1, x_2, x_3)$, trong đó f là hàm số. Nói chung với k biến độc lập x_1, x_2, x_3, …, x_k, chúng ta có rất nhiều mô hình (2^k) để tiên đoán y. Trong điều kiện có nhiều mô hình khả dĩ như thế, vấn đề đặt ra là mô hình nào được xem là tối ưu?

Câu hỏi trên đặt ra một câu hỏi cơ bản khác: thế nào là "tối ưu"? Nói một cách ngắn gọn một mô hình tối ưu phải đáp ứng ba tiêu chuẩn sau đây:

- Đơn giản
- Đầy đủ
- Có ý nghĩa thực tế

Tiêu chuẩn đơn giản đòi hỏi mô hình có ít biến số độc lập, vì nếu quá nhiều biến số thì vấn đề diễn dịch sẽ trở nên khó khăn, và có khi thiếu thực tế. Nói một cách ví von, nếu chúng ta bỏ ra 50.000 đồng để mua 500 trang sách tốt hơn là bỏ ra 60.000 ngàn để mua cùng số trang sách. Tương tự, một mô hình với 3 biến độc lập mà có khả năng mô tả dữ liệu tương đương với mô hình với 5 biến độc lập, thì mô hình đầu được chọn. Một mô hình đơn giản là một mô hình … tiết kiệm! (Tiếng Anh gọi là *parsimonious model*).

Tiêu chuẩn đầy đủ ở đây có nghĩa là mô hình đó phải mô tả dữ liệu một cách thỏa đáng, tức phải tiên đoán gần (hay càng gần càng tốt) với giá trị thực tế quan sát của biến

phụ thuộc *y*. Nếu giá trị quan sát của *y* là 10, và nếu có một mô hình tiên đoán là 9 và một mô hình tiên đoán là 6 thì mô hình đầu phải được xem là đầy đủ hơn.

Tiêu chuẩn "có ý nghĩa thực tế", như cách gọi, có nghĩa là mô hình đó phải được yểm trợ bằng lí thuyết hay có ý nghĩa sinh học (nếu là nghiên cứu sinh học), ý nghĩa lâm sàng (nếu là nghiên cứu lâm sàng), v.v... Có thể số điện thoại một cách nào đó có liên quan đến tỉ lệ gãy xương, nhưng tất nhiên một mô hình như thế hoàn toàn vô nghĩa. Đây là một tiêu chuẩn quan trọng, bởi vì nếu một phân tích thống kê dẫn đến một mô hình dù rất có ý nghĩa toán học mà không có ý nghĩa thực tế thì mô hình đó cũng chỉ là một trò chơi con số, trò chơi toán học không hơn không kém, chứ không có giá trị khoa học thật sự.

Tiêu chuẩn thứ ba (có ý nghĩa thực tế) thuộc về lĩnh vực lí thuyết, và tôi sẽ không bàn ở đây. Tôi sẽ bàn qua tiêu chuẩn đơn giản và đầy đủ. Một thước đo quan trọng và có ích để chúng ta quyết định một mô hình đơn giản và đầy đủ là Akaike Information Criterion (AIC) mà chúng ta đã gặp trong phần đầu của chương này. Để hiểu AIC, chúng ta quay lại với ví dụ 1.

Xin nhắc lại trong ví dụ 1, chúng ta muốn tiên đoán gãy xương (biến `fx`) từ các biến độc lập sau đây: độ tuổi (`age`), tỉ số cơ thể (`bmi`), mật độ chất khoáng trong xương (`bmd`), và hai chỉ số hủy xương (`ictp`) và tạo xương (`pinp`).

(a) Chúng ta thử mô hình `fx` là hàm số của độ tuổi:

```
> attach(fulldata)
> summary(glm(fx ~ age, family="binomial", data=fulldata))

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -8.06447    2.72559  -2.959  0.00309 **
age          0.09806    0.03766   2.604  0.00922 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 157.81  on 136  degrees of freedom
Residual deviance: 150.74  on 135  degrees of freedom
AIC: 154.74
```

Chúng ta để ý thấy residual deviance = 150.74, và AIC = 154.74. Thật ra, AIC được ước tính từ công thức:

$$\text{AIC} = \text{Residual Deviance} + 2(\text{số thông số})$$

Trong mô hình trên, chúng ta có 2 thông số (`intercept` và `age`), cho nên AIC = 150.74 + 4 = 154.74.

(b) Mô hình thứ hai mà chúng ta muốn so sánh là `fx` là hàm số của `ictp`:

```
> summary(glm(fx ~ ictp, family="binomial", data=fulldata))
```

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -3.9206     0.7726  -5.074 3.89e-07 ***
ictp          0.6066     0.1527   3.973 7.11e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 157.81  on 136  degrees of freedom
Residual deviance: 139.15  on 135  degrees of freedom
AIC: 143.15
```

Cũng với hai thông số, nhưng mô hình này có giá trị residual deviance (139.15) nhỏ hơn mô hình với độ tuổi (150.74), và do đó AIC cũng thấp hơn (143.15 so với 154.74). Kết quả này cho thấy mô hình với `ictp` mô tả `fx` đầy đủ hơn là mô hình với độ tuổi. So sánh này cho thấy trong hai mô hình này, chúng ta sẽ chọn mô hình với `ictp`.

(c) Bây giờ chúng ta thử xem mô hình với `ictp` và `age`.

```
> summary(glm(fx ~ ictp + age, family="binomial", data=fulldata))

(Intercept) -8.25707    2.91403  -2.834 0.004603 **
ictp         0.55461    0.15665   3.540 0.000399 ***
age          0.06398    0.04067   1.573 0.115701
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 157.81  on 136  degrees of freedom
Residual deviance: 136.61  on 134  degrees of freedom
AIC: 142.61
```

Mô hình này với 3 thông số (`intercept`, `age` và `ictp`), nhưng trị số AIC chỉ giảm xuống 142.61 (so với mô hình với `ictp` là 143.15), một độ giảm rất khiêm tốn, trong khi chúng ta phải "tiêu" thêm một thông số! Chúng ta có thể kết luận rằng `age` không cần thiết trong mô hình này. Thật vậy, trị số p cho age là 0.115, tức không có ý nghĩa thống kê.

Qua ba trường hợp trên, chúng ta có thể rút ra một nhận xét chung: một mô hình đơn giản và đầy đủ phải là mô hình có trị số AIC càng thấp càng tốt *và* các biến độc lập phải có ý nghĩa thống kê. Thành ra, vấn đề đi tìm một mô hình đơn giản và đầy đủ là thật sự đi tìm một (hay nhiều) mô hình với trị số AIC thấp nhất hay gần thấp nhất.

Tất nhiên, chúng ta có thể xem xét nhiều mô hình khác bằng cách thay thế hay tổng hợp các biến số độc lập với nhau. Nhưng một việc làm như thế rất phức tạp, đòi hỏi nhiều thời gian và có khi rườm rà. R có một hàm gọi là `step` có thể giúp chúng ta đi tìm một mô hình đơn giản và đầy đủ. Trong ví dụ trên, cách sử dụng hàm `step` sẽ được viết như sau:

```
> temp <- glm(fx ~ ., family="binomial", data=fulldata)
```

Trong lệnh trên, thông số "`fx ~ .`" có nghĩa là tìm tất cả các biến độc lập (kí hiệu ".") để tiên đoán `fx` trong dataframe `fulldata`. Kết quả cho vào đối tượng `temp`. Để xem kết quả trong temp, chúng ta lệnh `search` như sau:

```
> search <- step(temp)
```

```
> search <- step(temp)
Start:  AIC= 146.09
 fx ~ id + age + bmi + bmd + ictp + pinp

       Df Deviance    AIC
- pinp  1   132.45 144.45
- id    1   132.47 144.47
- age   1   132.63 144.63
- bmi   1   133.41 145.41
- bmd   1   133.87 145.87
<none>      132.09 146.09
- ictp  1   148.90 160.90

Step:  AIC= 144.45
 fx ~ id + age + bmi + bmd + ictp

       Df Deviance    AIC
- id    1   132.81 142.81
- age   1   133.14 143.14
- bmi   1   133.66 143.66
- bmd   1   134.00 144.00
<none>      132.45 144.45
- ictp  1   149.05 159.05

Step:  AIC= 142.81
 fx ~ age + bmi + bmd + ictp

       Df Deviance    AIC
- age   1   133.32 141.32
- bmi   1   133.67 141.67
- bmd   1   134.33 142.33
<none>      132.81 142.81
- ictp  1   149.88 157.88

Step:  AIC= 141.33
 fx ~ bmi + bmd + ictp

       Df Deviance    AIC
- bmi   1   134.34 140.34
<none>      133.32 141.32
- bmd   1   135.65 141.65
- ictp  1   155.18 161.18

Step:  AIC= 140.34
 fx ~ bmd + ictp

       Df Deviance    AIC
<none>      134.34 140.34
- bmd   1   139.15 143.15
- ictp  1   155.27 159.27
```

Trong kết quả trên, R báo cáo cho chúng ta biết từng bước trong quá trình đi tìm mộ mô hình tối ưu. Khởi đầu là mô hình với tất cả 6 biến, và trị số AIC = 146.09. Bước thứ hai

chỉ gồm 5 biến (loại bỏ `pinp`) và AIC = 144.45. Và vân vân. Kết quả có thể tóm lược trong bảng sau đây:

Mô hình	AIC
`fx ~ id + age + bmi + bmd + ictp + pinp`	146.09
`fx ~ id + age + bmi + bmd + ictp`	144.45
`fx ~ age + bmi + bmd + ictp`	142.81
`fx ~ bmi + bmd + ictp`	141.33
`fx ~ bmd + ictp`	140.34

Kết quả 5 bước tìm mô hình, R dừng lại với mô hình gồm 2 biến bmd và ictp vì có giá trị AIC thấp nhất. Thật ra, nếu không muốn in tất cả các bước đi tìm mô hình, chúng ta chỉ cần lệnh `summary` như sau:

```
> summary(search)

Call:
glm(formula = fx ~ bmd + ictp, family = "binomial", data = fulldata)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9126  -0.7317  -0.5559   0.4212   2.1242

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.0651     1.5029  -0.709   0.4785
bmd          -3.4998     1.6638  -2.103   0.0354 *
ictp          0.6876     0.1704   4.036 5.43e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 157.81  on 136  degrees of freedom
Residual deviance: 134.34  on 134  degrees of freedom
AIC: 140.34

Number of Fisher Scoring iterations: 4
```

Kết quả này đơn giản hơn kết quả của hàm `search`, vì `summary` chỉ trình bày mô hình sau cùng. Nói tóm lại, trong phân tích này, chúng ta kết luận rằng `bmd` (mật độ chất khoáng trong xương) và `ictp` (marker về chu trình hủy xương) là hai yếu tố có liên hệ hay ảnh hưởng đến nguy cơ gãy xương.

12.6 Chọn mô hình hồi qui logistic bằng Bayesian Model Average (BMA)

Trong chương 10, tôi đã nói qua cách chọn và xây dựng một mô hình hồi qui tuyến tính bằng ứng dụng phép tính BMA. Chúng ta cũng có thể ứng dụng BMA vào việc xây dựng một mô hình hồi qui logistic.

Tiếp tục ví dụ 1, chúng ta sẽ chuẩn bị dữ liệu cho phân tích BMA bằng cách chọn ra biến phụ thuộc (trong trường hợp này là `fx`) và một ma trận gồm các biến độc lập. Tiếp theo đó, chúng ta sử dụng hàm `bic.glm` để tìm các biến có ảnh hưởng đến `fx`.

```
> attach(fulldata)
> names(fulldata)
[1] "id"   "fx"   "age"  "bmi"  "bmd"  "ictp" "pinp"
```

Chọn cột 3 đến 7 (từ `age` đến `pinp`) làm ma trận biến độc lập
```
> xvars <- fulldata[,3:7]
```

Chọn `fx` làm biến phụ thuộc
```
> y <- fx
```

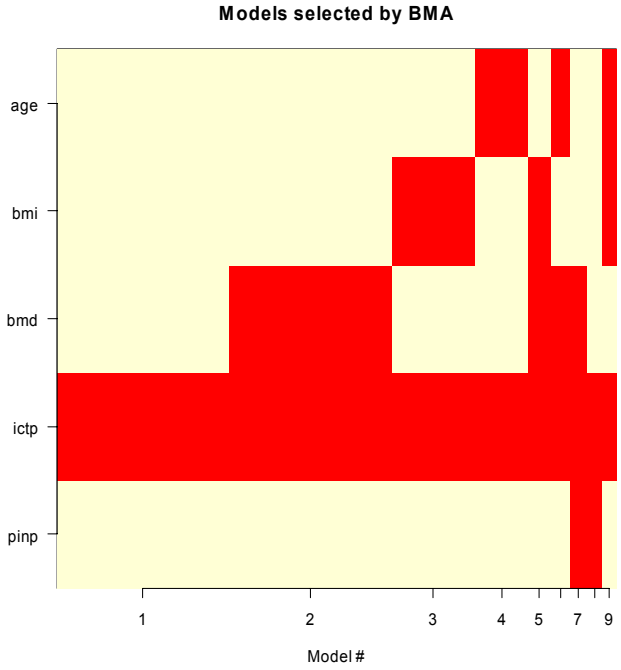
Gọi hàm `bic.glm` với các thông số như sau:
```
> bma.search <- bic.glm(xvars, y, strict=F, OR=20, glm.family="binomial")
```

Tóm lược kết quả phân tích:
```
> summary(bma.search)

Call:
bic.glm.data.frame(x = xvars, y = y, glm.family = "binomial",     strict = F, OR = 20)

  9  models were selected
 Best  5  models (cumulative posterior probability =  0.8836 ):

           p!=0    EV       SD      model 1   model 2  model 3   model 4   model 5
Intercept  100    -2.85012  2.8651    -3.920    -1.065   -1.201    -8.257    -0.072
age         15.3   0.00845  0.0261       .         .        .       0.063       .
bmi         21.7  -0.02302  0.0541       .         .     -0.116       .      -0.070
bmd         39.7  -1.34136  1.9762       .      -3.499      .         .      -2.696
ictp       100.0   0.64575  0.1699     0.606     0.687    0.680     0.554     0.714
pinp         5.7  -0.00037  0.0041       .         .        .         .         .

nVar                                       1         2        2         2         3
BIC                                     -525.044  -524.939 -523.625  -522.672  -521.032
post prob                                  0.307     0.291    0.151     0.094     0.041
```

Kết quả phân tích trên đây cho thấy xác suất mà `ictp` là liên quan đến gãy xương là 100%, trong khi đó, xác suất cho `bmd` chỉ khoảng 40%. Nhưng quan trọng hơn, mô hình "tối ưu" nhất là mô hình với `ictp`, và xác suất cho mô hình này là 0.307. Mô hình tối ưu thứ hai gồm có `ictp` và `bmd` (cũng là mô hình dựa vào tiêu chuẩn AIC như mô tả phần trên), nhưng xác suất cho mô hình này thương đối thấp hơn (0.291). Ba mô hình khác cũng có thể là "ứng viên" để mô tả xác suất gãy xương đầy đủ. Rõ ràng, qua phân tích BMA, chúng ta có nhiều lựa chọn mô hình hơn, và ý thức được sự bất định của một mô hình thống kê.

Biểu đồ sau đây thể hiện kết quả trên. Qua biểu đồ này chúng ta thấy `ictp` là yếu tố có ảnh hưởng đến nguy cơ gãy xương nhất quán nhất. Yếu tố quan trọng thư hai có lẽ là `bmd` hay `bmi`. Các yếu tố như `age` và `pinp` tuy có khả năng ảnh hưởng đến nguy cơ gãy xương, nhưng các yếu tố này không có độ nhất quán cao như các yếu tố vừa kể trên.

```
> imageplot.bma(bma.search)
```



Xây dựng mô hình thống kê là một nghệ thuật toán học. Vì tính nghệ thuật của việc làm, nhà nghiên cứu phải cân nhắc rất nhiều yếu tố để đi đến một mô hình đẹp. Bởi vì mô hình là nhằm mục đích mô tả thực tế, một mô hình đẹp là mô hình mô tả sát với thực tế. Tuy nhiên nếu một mô hình phản ánh 100% thực tế thì đó không còn là "mô hình" nữa, hay quá phức tạp không thể ứng dụng được. Ngược lại một mô hình chỉ mô tả thực tế khoảng 1% thì cũng không thể sử dụng được. Xây dựng mô hình phải làm sao tìm điểm cân bằng cho hai thái cực đó. Đó là một yêu cầu rất cao, cho nên xây dựng mô hình không chỉ tùy thuộc vào các phép tính thống kê, toán học, mà còn phải xem xét đến các yếu tố thực tế để bảo đảm cho sự hữu ích của mô hình. Nói như nhà thống kê học nổi tiếng George Box: "Mô hình nào cũng sai so với thực tế, nhưng trong số các mô hình sai đó, có một vài mô hình có ích".

12.7 Số liệu nghiên cứu về nguy cơ gãy xương trong nam giới trên 60 tuổi

- id: mã số bệnh nhân
- fx: gãy xương hay không (0=không gãy xương, 1=gãy xương)
- age: độ tuổi
- bm: body mass index, tính bằng trọng lượng chia cho chiều cao bình phương
- bmd: (bone mineral density) mật độ chất khoáng trong xương đùi.
- ictp: chỉ số sinh hóa đo lường hoạt tính hủy xương
- pinp: chỉ số sinh hóa đo lường hoạt tính tạo 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
11	1	76	22.5981	0.889	4.704	27.026
12	0	76	26.4236	0.886	5.115	43.256
13	1	62	20.3223	0.889	5.741	51.097
14	0	69	19.3698	0.790	3.880	49.678
15	0	72	24.2215	0.988	5.844	41.672
16	0	67	32.1120	1.119	4.160	60.356
17	0	74	25.3934	1.037	6.728	40.225
18	0	69	23.8895	0.893	4.203	27.334
19	1	78	24.6755	0.850	7.347	38.893
20	0	71	27.1314	0.790	4.476	38.173
21	1	74	23.0518	0.597	4.835	35.141
22	1	76	23.4568	0.889	5.354	27.568
23	1	75	23.5457	0.803	3.773	36.762
24	0	70	23.3234	0.919	3.672	40.093
25	1	69	22.8625	0.870	4.552	29.627
26	0	71	22.0384	0.811	4.286	30.380
27	1	80	24.6914	0.859	5.706	37.529
28	1	79	26.8519	0.867	3.563	43.924
29	0	72	27.1809	0.717	3.760	39.714
30	0	78	23.9512	0.822	3.453	27.294
31	1	80	28.3874	1.004	5.948	33.376
32	0	79	23.5102	0.738	4.193	65.640
33	1	67	19.7232	0.865	4.443	36.252
34	1	84	27.4406	0.808	5.482	33.539
35	0	78	28.6661	0.955	8.815	42.398
36	0	65	23.7812	0.912	4.704	39.254
37	0	70	23.4493	0.857	4.138	75.947
38	0	67	25.5354	0.855	3.727	41.851
39	0	74	24.7409	0.959	3.967	42.293
40	0	73	22.2291	1.036	4.438	40.222
41	0	74	34.4753	1.092	7.271	45.434
42	1	68	32.1929	.	4.269	50.841
43	0	80	23.3355	0.759	4.856	31.114
44	0	78	22.7903	0.757	4.831	73.343
45	1	79	24.6097	0.671	4.870	69.924
46	0	72	27.5802	0.814	3.012	27.088
47	1	67	30.1205	1.101	7.538	35.487
48	0	70	25.8166	0.818	3.564	36.001
49	0	69	30.4218	1.088	3.826	33.833
50	0	67	28.7132	0.934	3.996	56.167

51	0	74	34.5429	0.969	6.762	43.099
52	0	71	24.6097	0.794	4.350	39.023
53	0	67	23.5294	0.830	3.176	36.595
54	0	67	25.6173	1.057	3.738	32.550
55	0	65	25.3086	1.160	3.060	44.757
56	0	66	24.8358	0.811	3.263	26.941
57	0	69	22.3094	0.977	3.106	27.951
58	0	72	26.5285	1.063	6.970	41.188
59	0	75	25.8546	1.091	4.798	36.045
60	0	70	20.6790	0.741	3.908	30.198
61	0	74	28.3675	1.045	4.784	31.339
62	0	71	29.0688	1.066	4.527	24.252
63	0	65	23.9995	0.841	3.089	79.910
64	0	77	22.9819	1.015	4.041	57.147
65	1	67	33.3598	1.129	7.239	67.103
66	0	66	27.1314	1.030	4.096	29.435
67	0	70	24.7676	0.896	4.352	44.291
68	0	70	24.4193	1.106	2.823	37.348
69	0	69	28.2570	0.869	2.974	46.229
70	1	65	23.6614	0.837	2.689	28.738
71	1	75	26.0262	0.921	3.917	29.667
72	0	67	26.5731	1.118	3.832	50.292
73	0	67	24.8591	0.765	7.112	45.778
74	0	73	22.5710	0.752	4.249	39.950
75	1	63	31.8342	1.251	7.303	48.697
76	1	72	24.8016	0.839	3.860	41.055
77	0	73	25.0574	0.662	3.138	36.312
78	0	69	23.9512	0.844	4.069	39.926
79	0	75	23.4586	0.852	4.176	51.394
80	0	65	28.7347	0.795	3.328	27.679
81	0	71	25.3350	0.867	2.349	36.506
82	0	66	28.0899	0.997	4.171	53.094
83	0	66	25.5650	0.827	4.569	25.157
84	0	71	28.7274	1.023	4.111	19.557
85	0	73	32.4074	1.066	5.680	36.995
86	0	64	27.9155	0.874	4.298	43.872
87	0	68	25.5937	0.882	4.056	30.523
88	1	67	28.0428	0.718	9.739	66.974
89	0	66	30.7174	0.856	4.180	34.597
90	0	77	28.3737	1.052	3.737	28.102
91	0	75	28.6990	0.929	3.527	23.008
92	0	67	29.1687	0.953	3.593	16.132
93	0	73	27.4145	0.784	4.332	47.410
94	0	68	29.0688	1.120	6.510	45.674
95	0	70	26.1738	1.040	3.161	36.302
96	0	66	30.1038	1.028	3.930	38.301
97	0	77	24.6559	0.884	3.880	36.560
98	1	71	25.3934	0.943	4.692	69.500
99	0	74	26.4721	1.075	4.561	25.948
100	0	70	29.0253	1.057	3.709	41.322
101	0	78	29.0253	1.098	5.247	23.896
102	0	76	26.2346	1.014	3.958	24.344
103	1	64	26.4915	0.998	4.218	29.390
104	0	67	27.0416	0.905	3.553	23.020
105	0	66	22.7732	0.627	2.333	53.621
106	0	70	30.5241	1.052	5.425	44.352
107	0	66	25.3069	1.086	4.945	64.788
108	1	65	22.3863	0.818	3.786	96.360
109	1	64	34.0136	1.066	5.792	37.473
110	1	70	26.5668	1.198	7.257	28.406
111	1	70	27.6361	0.926	5.746	17.228
112	0	70	25.4017	1.193	2.437	35.432
113	0	68	30.3673	0.938	2.658	32.293
114	0	67	28.0428	0.863	4.246	48.702
115	1	73	27.7778	0.799	3.934	26.709
116	0	71	29.0006	0.969	4.054	22.769
117	1	71	35.2941	0.931	3.631	18.629
118	0	75	29.3658	1.071	4.222	36.555
119	0	76	26.2649	1.161	2.548	24.217
120	0	71	25.6055	0.786	3.832	32.023
121	0	73	29.9136	0.839	4.215	26.507

122	0	64	34.5271	1.042	6.436	53.080
123	0	70	33.4554	0.976	4.541	26.619
124	0	80	29.0688	0.765	3.998	67.388
125	0	67	25.7276	1.277	3.877	22.159
126	0	68	25.6801	1.097	3.782	42.286
127	0	66	25.9701	0.793	2.991	38.673
128	0	64	26.4490	0.989	3.196	31.456
129	1	69	28.6990	0.822	3.565	45.044
130	0	69	25.6173	0.944	6.512	49.557
131	0	67	30.3871	1.245	3.603	46.769
132	0	67	33.6901	1.142	3.666	38.839
133	0	68	28.4005	0.860	2.890	32.140
134	1	59	25.4017	1.172	.	104.579
135	0	66	22.5710	0.956	3.354	36.253
136	1	71	24.4473	0.918	4.633	53.881
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