Bài tập R tự làm
Phần III: Phân tích thống kê

Nguyễn Văn Tuấn

8 Phân tích mô tả

Phần lớn các phân tích dữ liệu đều bắt đầu từ phân tích mô tả. Trong phân tích mô tả, như tên gọi, chúng ta quan tâm đến ước tính một số thông số phổ biến như trung bình, trung vị, phương sai, độ lệch chuẩn, v.v. Các ước số này giúp chúng ta "hiểu" thêm về dữ liệu và có những cảm nhận ban đầu về dữ liệu phân tích.

8. 1 Mô tả dữ liệu dùng hàm sẵn có trong R

Trong phần này, các bạn sẽ làm việc với file "`fev.xlsx`". Trước hết, các bạn hãy export dữ liệu này sang dạng .csv, và dùng read.csv để đọc vào R:

```
fev=read.csv("~/Google Drive/TDT Projects/Workshop 12-
2015/Datafiles/fev.csv", header=T)

attach(fev)

head(fev)
```

Trong R, có hàm `summary` có thể dùng để xem qua các chỉ số phân tích thống kê mô tả:

```
summary(fev)
```

8. 2 Mô tả dữ liệu dùng hàm `describe` và `describeBy` trong `psych`

Hoặc dùng hàm `describe` trong package `psych` cũng có thể cung cấp nhiều thông tin quan trọng:

```
library(psych)
describe(fev)
describe(fev, skew=F)
describe(fev, skew=F, range=F)
```

Mô tả theo nhóm với hàm `describeBy:`

```
describeBy(fev, group=Gender)
describeBy(fev, group=Gender, range=F, skew=F)
```

8. 3 Mô tả dữ liệu dùng hàm `stat.desc` trong package `astecs`.

Package "pastecs" cũng có thể sử dụng để mô tả dữ liệu đơn giản.

```
library(pastecs)
stat.desc(fev)

options(digits=2)
stat.desc(fev)

stat.desc(fev, basic=F)
stat.desc(fev, desc=F)
```

Một cách khác nữa là dùng hàm `tabular` trong package `tables` (đã đề cập trong phần II) để mô tả dữ liệu và sắp xếp trong những bảng kết quả có chất lượng tốt hơn các hàm trên đây.

8. 4 Ước tính khoảng tin cậy 95% của một số trung vị

Không có phương pháp toán học để ước tính khoảng tin cậy 95% của trung vị, nên chúng ta phải dùng phương pháp bootstrap như sau:

```
x = rnorm(n=100, mean=25, sd=5)
B = 10000
xbar = rep(0, B)

for (i in 1:B) {
  bs.x = sample(x, length(x), replace=T)
  xbar[i] = median(bs.x)
}

quantile(xbar, c(0.025, 0.5, 0.975))
hist(xbar, col="blue", border="yellow")
```

9 Kiểm định giả thuyết: biến liên tục

Kiểm định giả thuyết khoa học thường được thể hiện dưới dạng so sánh và tương quan. So sánh có thể giữa hai nhóm hay nhiều hơn hai nhóm. Trong trường hợp một nhóm, kiểm định giả thuyết khoa học có thể liên quan đến việc phân tích mối tương giữa hai hay nhiều hơn 2 biến số. Trong phần này, các bạn sẽ thực hành một số phương pháp phân tích thống kê phổ biến qua việc sử dụng các hàm R thông thường. Các bạn cũng sẽ học thêm một số phương pháp hiện đại và "nâng cao" như bootstrap và phương pháp Bayes.

9.1 Kiểm định t cho một số trung bình

1. Kiểm định t cho một mẫu. Dữ liệu dưới đây là trọng lượng của 10 con chuột (tính bằng gram). Hãy thử kiểm định giả thuyết rằng trọng lượng trung bình là 350 g.

   ```
   wt = c(355, 533, 630, 218, 513, 551, 560, 431, 434, 432, 393)

   t.test(wt, mu=350)
   ```

   Diễn giải kết quả của phân tích trên.

2. Nghiên cứu trước - sau (before - after study). Kiểm định t cũng có thể áp dụng để kiểm định giả thuyết của các nghiên cứu mà đối tượng được đo lường 2 lần, thường là trước và sau can thiệp. Dữ liệu dưới đây so sánh điểm thi của 15 học sinh trước và sau khi dự một lớp học về kĩ năng học.

```
before = c(18, 21, 16, 22, 19, 24, 17, 21, 23, 18, 14, 16, 16,
19, 18)

after = c(22, 25, 17, 24, 16, 29, 20, 23, 19, 20, 15, 15, 18,
26, 18)
```

Hãy dùng phương pháp kiểm định t để đánh giá xem lớp học có tác động đến điểm học.

```
diff = after-before
t.test(diff, mu=0)
```

Viết một vài dòng để diễn giải kết quả của phân tích trên.

3. Phương pháp phân tích boostrap cho nghiên cứu trước - sau.

```
# nhập dữ liệu

before = c(18, 21, 16, 22, 19, 24, 17, 21, 23, 18, 14, 16, 16, 19, 18)
after = c(22, 25, 17, 24, 16, 29, 20, 23, 19, 20, 15, 15, 18, 26, 18)
dat = data.frame(before, after)


# Bắt đầu lấy mẫu ngẫu nhiên
fd <- function(data, i) {
       d = dat[i, ]
       diff = d$after - d$before
       }

# gọi package boot
library(boot)

bdiff = boot(dat, fd, R=10000)
boot.ci(bdiff)
plot(bdiff)
```

4. Phương pháp hoán vị. Dữ liệu từ các nghiên cứu trước sau cũng có thể phân tích bằng phương pháp hoán vị (permutation) qua package coin như sau:

```
before = c(18, 21, 16, 22, 19, 24, 17, 21, 23, 18, 14, 16, 16, 19, 18)
after = c(22, 25, 17, 24, 16, 29, 20, 23, 19, 20, 15, 15, 18, 26, 18)

library(coin)
perm.test(before, after, paired=T, exact=T, alternative="greater")
```

9.2 Kiểm định t cho 2 số trung bình

1. Dữ liệu dưới đây được thu thập từ hai nhóm A và B. Giả thuyết vô hiệu là hai nhóm có trung bình giống nhau. Bạn hãy dùng hàm `t.test()` trong R để kiểm định giả thuyết đó. Trước hết nhập dữ liệu hai nhóm:

```
A = c(79.98, 80.04, 80.02, 80.04, 80.03, 80.03, 80.004,
79.97, 80.05, 80.03, 80.02, 80.00, 80.02)

B = c(80.02, 79.94, 79.98, 79.97, 79.97, 80.03,  79.95,
79.97)
```

Bước kế tiếp là vẽ biểu đồ hộp:

```
boxplot(A, B, col="blue")
boxplot(A, B, col="blue", border="red", names=c("A", "B"))
```

Dùng hàm `t.test`, và diễn giải kết quả phân tích:

```
t.test(A, B)
```

Kiểm tra xem phương sai của hai nhóm tương đương nhau?

```
var.test(A, B)
```

2. Một cách khác để dùng hàm `t.test` là tạo dữ liệu thành 2 cột. Cột 1 là biến chỉ nhóm, cột hai là biến số liệu cho phân tích:

```
A = c(79.98, 80.04, 80.02, 80.04, 80.03, 80.03, 80.004,
79.97, 80.05, 80.03, 80.02, 80.00, 80.02)

B = c(80.02, 79.94, 79.98, 79.97, 79.97, 80.03,  79.95,
79.97)

# tạo ra biến nhóm
nA = length(A)
nB = length(B)
Group = c(rep("A", nA), rep("B", nB))
# Tạo ra biến WT
WT = c(A, B)
# Tạo ra một dataset
dat = data.frame(Group, WT)
```

Dùng hàm `t.test`

```
t.test(WT ~ Group, data=dat)
```

Kiểm tra xem kết quả có giống như câu hỏi 1.

Kiểm tra biến số có tuân theo luật phân bố chuẩn, và diễn giải kết quả:

```
hist(WT)
qqnorm(WT); qqline(WT)

shapiro.test(WT)
```

3. **Dùng phương pháp bootstrap**. Nếu bạn cảm thấy chưa "thoải mái" với kết quả phân tích trên vì biến số có vẻ không tuân theo luật phân bố chuẩn, thì

một phương pháp kiểm định khác có tên là bootstrap có thể giúp cho việc kiểm định giả thuyết tốt hơn.

```
k = 10000 # lấy mẫu 10 ngàn lần
bsampleA = replicate(k, sample(A, replace = T))
bsampleB = replicate(k, sample(B, replace = T))
# mỗi lần tính toán khác biệt giữa 2 nhóm
diff = apply(bsampleA,2,mean) - apply(bsampleB,2,mean)
# báo cáo kết quả
quantile(diff, c(0.025, 0.50, 0.0975))
hist(diff, col="blue", border="white")
```

Diễn giải kết quả phân tích, và so sánh với phương pháp t.test.

4. **Phương pháp bootstrap** thay thế cho kiểm định t. Trong điều kiện biến phân tích không tuân theo luật phân bố chuẩn, hoặc vi phạm các giả định thống kê, có thể dùng phương pháp bootstrap để thay thế kiểm định t. Các mã dưới đây kiểm định sự khác biệt giữa 2 nhóm (treat và control) bằng phương pháp bootstrap:

```
treat = c(0.05, 0.15, 0.35, 0.25, 0.20, 0.05, 0.10, 0.05,
            0.30, 0.05, 0.25)

control = c(0, 0.15, 0, 0.05, 0, 0, 0.05, 0.10)

n1 = length(treat)
n2 = length(control)

B = 10000
difference = numeric(B)
no.effect = 0

for (i in 1:B) {
   bs.treat <- sample(treat, n1, replace=T)
   bs.control <- sample(control, n2, replace=T)
   difference[i] = mean(bs.treat) - mean(bs.control)
   if (difference[i] < 0) no.effect = no.effect+1
  }

hist(difference, col="blue", border="white")
quantile(diff, c(0.025, 0.5, 0.975))
```

Diễn giải kết quả phân tích. Chứng cứ có thuyết phục cho giả thuyết hai nhóm thật sự khác nhau?

5. **Phương pháp phi tham số (non-parametric)** như Wilcoxon's rank test cũng có thể dùng để kiểm định so sánh 2 nhóm với biến số liên tục. Trong R có hàm `wilcox.test` như sau:

```
treat = c(0.05, 0.15, 0.35, 0.25, 0.20, 0.05, 0.10, 0.05,
```

```
              0.30, 0.05, 0.25)

control = c(0, 0.15, 0, 0.05, 0, 0, 0.05, 0.10)

wilcox.test(treat, control)
```

Diễn giải kết quả phân tích. Chứng cứ có thuyết phục cho giả thuyết hai nhóm thật sự khác nhau?

6. **Phương pháp hoán vị (permutation test)** cũng có thể dùng để kiểm định so sánh 2 nhóm với biến số liên tục. Package `exactRankTests` hoặc `coin` có thể dùng cho mục tiêu này:

```
library(exactRankTests); library(coin)

treat = c(0.05, 0.15, 0.35, 0.25, 0.20, 0.05, 0.10, 0.05,
              0.30, 0.05, 0.25)

control = c(0, 0.15, 0, 0.05, 0, 0, 0.05, 0.10)

perm.test(treat, control, exact=T)
```

7. **Kiểm định t dựa vào dữ liệu tóm tắt.** Các phân tích trên đây dựa vào giả định có dữ liệu gốc của từng đối tượng nghiên cứu. Tuy nhiên, có nhiều trường hợp dữ liệu chỉ tồn tại dưới dạng tóm lược (như số đối tượng, trung bình, và độ lệch chuẩn). Trong trường hợp đó, có thể dùng package BSDA để thực hiện kiểm định t:

```
library(BSDA)
tsum.test(mean.x=0.16, s.x=0.11, n.x=11, mean.y=0.044,
s.y=0.056, n.y=8)
```

10 Kiểm định giả thuyết: biến phân loại

10.1 Mô tả một tỉ lệ

1. Ước tính khoảng tin cậy 95% của một tỉ lệ bằng phương pháp chính xác (exact method).

   Dùng package binom để ước tính khoảng tin cậy 95% của tỉ lệ:

```
library(binom);
x = c(34, 12);
n = c(5964, 5566)
binom.confint(x, n, conf.level=0.95, methods="all")
```

Diễn giải và so sánh các kết quả phân tích.

10.2 So sánh 2 tỉ lệ

1. **So sánh hai tỉ lệ, phương pháp kiểm định nhị phân (binomial test)**. Dữ liệu so sánh hai tỉ lệ (hoặc 2 xác suất) thường xuất hiện dưới dạng bảng số liệu 2 dòng và 2 cột (2 x 2 table). Số liệu dưới đây là kết quả từ một nghiên cứu về estrogen và ung thư vú. Nghiên cứu có 2 nhóm: nhóm 5964 nữ được điều trị bằng thay thế estrogen, và nhóm 5566 nữ không dùng thuốc. Kết quả sau 5 năm nghiên cứu, nhóm estrogen có 34 người mắc bệnh ung thư vú, nhóm chứng có 12 người mắc bệnh ung thư vú:

Nhóm	N	Ung thư	Không ung thư
Estrogen	5964	34	5930
Chứng	5566	12	5554

Có thể nhập dữ liệu dưới dạng ma trận như sau:

```
Group = c("Estrogen", "Estrogen", "Control", "Control")
Cancer = c("Yes", "No", "Yes", "No")
NN = c(34, 5930, 12, 5554)
dat = data.frame(Group, Cancer, NN)
dat
```

hoặc nhập dữ liệu như là một ma trận:

```
dat1 = matrix(c(34, 5930, 12, 5554), ncol=2, byrow=T)
colnames(dat1) = c("Cancer", "Not cancer")
rownames(dat1) = c("Estrogen", "Control")
dat1
```

Dùng hàm R để tính toán tỉ lệ ung thư và khoảng tin cậy 95% của từng nhóm.

```
prop.test(34, 5964)
prop.test(12, 5566)
```

Dùng hàm R để kiểm định giả thuyết vô hiệu là hai nhóm có nguy cơ ung thư như nhau:

```
prop.test(as.table(dat1))
```

2. **Relative risk: So sánh hai tỉ lệ, phương pháp bootstrap**. Một phương pháp so sánh 2 tỉ lệ là dùng relative risk - tỉ số nguy cơ. Gọi nguy cơ mắc bệnh của hai nhóm là p_1 và p_2, RR được định nghĩa là RR = p_1 / p_2. Phương pháp bootstrap (tái chọn mẫu) có thể áp dụng để tính khoảng tin cậy 95% của RR. Thử các lệnh sau đây và diễn giải kết quả:

```
n = c(5964, 5566)
disease = c(34, 12);
```

```
estrogen = c(rep(1, disease[1]), rep(0, n[1]-disease[1]));
control = c(rep(1, disease[2]), rep(0, n[2]-disease[2]));
B = 10000;
bs1 = rep(NA, B);
bs2 = rep(NA, B);
for (i in 1:B) {
 resample1 = sample(estrogen, n[1], replace=T)
 dis1 = sum(resample1)
 resample2 = sample(control, n[2], replace=T)
 dis2 = sum(resample2)
 bs1[i] = dis1/n[1]
 bs2[i] = dis2/n[2]
}
RR = bs1/bs2;
quantile(RR, c(0.025, 0.50, 0.975))
```

3. **So sánh hai tỉ lệ, phương pháp bootstrap**. Phương pháp phân tích trong câu 2 trên là tính tỉ số nguy cơ (relative risk hay RR). Hãy sửa lại mã trên để tính khác biệt giữa 2 tỉ lệ. (Gợi ý: sửa dòng lệnh **`RR = bs1/bs2`** trong chương trình trên).

4. **So sánh hai tỉ lệ, phương pháp Bayes**. Trong mã dưới đây, các bạn so sánh hai tỉ lệ bằng phương pháp Bayes, với thông tin tiền định là phân bố Beta.

```
x1 = 34;  n1 = 5964; alpha1 = 1; beta1 = 1;
x2 = 12;  n2 = 5566; alpha2 = 1; beta2 = 1;
p1 = rbeta(10000, x1 + alpha1, n1 - x1 + beta1);
p2 = rbeta(10000, x2 + alpha2, n2 - x2 + beta2);
rd = p2 - p1;
plot(density(rd));
hist(rd, col="blue", border="yellow")
quantile(rd, c(.025, 0.5, 0.975))
```

11 Phân tích phương sai (ANOVA)

11.1 Phân tích phương sai 1 yếu tố (one-way ANOVA)

Bệnh vẩy nến (psoriasis) là một bệnh do rối loạn hệ miễn dịch làm ảnh hưởng đến da. Các nhà khoa học thực hiện một nghiên cứu trên 37 bệnh nhân để xem mối liên quan giữa gene IGFL4 và bệnh. Dữ liệu `psoriasis.xlsx` gồm có 3 biến:

`type`: loại da (psne, psor, và healthy);
`intensity`: là mức độ hoạt động của gen, biến liên tục;
`typeNum`: là mã số của type (với giá trị 1, 2, 3).

Câu hỏi nghiên cứu là intensity có khác biệt giữa các loại da hay không. Nói cách khác, có mối liên quan giữa loại da và kích hoạt của gen IGFL4?

1. Đọc dữ liệu vào R và gọi là `psoriasis`. Dùng srtipchart hay boxplot để có một cái nhìn chung về dữ liệu:

```
stripchart(intensity ~ type)
boxplot(intensity ~ type)
```

2. Tính trung bình theo nhóm da:

```
mean(intensity[type=="healthy"])
mean(intensity[type=="psne"])
mean(intensity[type=="psor"])
```

3. Phân tích ANOVA:

```
model = lm(intensity ~ type, data=psoriasis)
summary(model)
```

Diễn giải kết quả phân tích. Chú ý đến các giá trị Estimate, Std Error, t-value và trị số P.

4. Có thể thử mô hình sau:

```
model2 = lm(intensity ~ type –1, data=psoriasis)
summary(model2)
```

Chú ý Estimates bây giờ có phần khác hơn so với `model`.

4. So sánh giữa các nhóm (post hoc comparison)

```
anova = aov(intensity ~ type, data=psoriasis)
TukeyHSD(anova)
plot(TukeyHSD(anova))
```

Giải thích kết quả phân tích post-hoc.

11.2 Phân tích phi tham số tương đương với ANOVA là Kruskal Wallis test cũng có thể triển khai trong R:

```
kruskal.test(intensity ~ type)
```

12 Hồi qui tuyến tính

12.1 Mô hình hồi qui tuyến tính đơn giản

Trong phần này, các bạn sẽ phân tích dataset cherry.csv. Do đó, trước hết cần phải đọc dữ liệu này vào R.

1. Ước tính tham số

```
model1 = lm(volume ~ diameter, data=cherry)
model1

# vẽ biểu đồ tán xạ và thêm đường biểu diễn
plot(volume ~ diameter, pch=16)
abline(model1)
```

2. Cần thêm thôn tin về mô hình

```
summary(model1)
```

Bạn hãy chú ý và hiểu cho được các thông số trong Estimate, Std. Error, t-value.

3. Thử lệnh `confint(model)` để có thêm thông tin về khoảng tin cậy 95% của các tham số.

4. Kiểm tra (validation) mô hình. Tìm hiểu các fitted values, raw residuals, standardized residuals từ mô hình. Giải thích ý nghĩa của các thông tin này.

```
plot(fitted(model1), residuals(model1))
plot(fitted(model1), rstandard(model1))
qqnorm(rstandard(model1))
abline(0,1)
```

Kiểm tra xem mô hình có thích hợp hay không?

Một cách khác là vẽ tất cả biểu đồ trong một cửa sổ:

```
par(mfrow=c(2,2))
plot(model1)
```

5. Hoán chuyển dữ liệu: Phân tích dữ liệu với biến log(volume) là biến phụ thuộc, và log(diameter) là biến độc lập. Xem mô hình mới có thích hợp cho dữ liệu.

6. Mô hình hồi qui đa biến: Phân tích mô hình hồi qui đa biến với log(volume) là biến phụ thuộc, và hai biến độc lập là log(diameter) và log(height):

```
model2 = lm(log(volume) ~ log(diameter) +
log(height), data=cherry)

summary(model2)
```

So sánh hai mô hình model1 và model2:

```
anova(model2, model1)
```

Giải thích ý nghĩa của output trên.

12.2 Mô hình hồi qui tuyến tính đơn giản, tiên lượng: dữ liệu cats

Trong phần này, các bạn sẽ phân tích dataset cats trong package MASS.

```
library(MASS)
data(cats)
cats
```

gồm 144 số liệu về cân nặng (`Bwt`), trọng lượng tim (`Hwt`) của 144 con mèo đực và cái (`Sex`).

1. Vẽ biểu đồ tương quan (scatter plot) với trục hoành là `Bwt` và trục tung là `Hwt`. Bạn hãy xem mối tương quan có phải tuyến tích hay không?

2. Phân tích mô hình hồi qui tuyến tính *cho các mèo đực*, với biến phụ thuộc là `Hwt` và biến độc lập là `Bwt`. Gọi đối tượng là **`linreg`**. Vẽ đường biểu diễn tuyến tính (dùng hàm `abline`).

3. Tìm hệ số tương quan của đường biểu diễn (fitted line). Tìm mức độ khác biệt về trọng lượng quả tim nếu hai con mèo khác nhau về cân nặng 1 kg. Tìm khoảng tin cậy 95% của độ khác biệt.

4. Dùng kĩ thuật validation để đánh giá sự thích hợp của mô hình hồi qui tuyến tính.

5. Dùng ước số để tìm giá trị kì vọng (hay tiên lượng) của trọng lượng quả tim cho một con mèo nặng 3 kg.

```
new.obs = data.frame(Bwt=3)
predict(linreg, new.obs)
predict(linreg, new.obs, interval="predict")
```

12.3 Mô hình hồi qui tuyến tính đa biến: toxicity of dissolution

Dữ liệu từ sự phân huỷ của 24 hoá chất được thu thập để nghiên cứu về mối liên quan giữa độ độc của hoá chất và vài yếu tố khác. Dữ liệu lser.xlsx có các biến sau đây:

`tox`: độ độc của hoá chất
`base`: khả năng hấp thu hydrogen ion
`colour`: khả năng đổi màu
`acid`: khả năng giải thoát hydrogen ion

1. Lưu trữ dữ liệu dưới dạng csv, đọc vào R, và tạo ra một R dataset gọi là `lser`. Dùng lệnh `plot(lser)` để xem dữ liệu.

2. Dùng lm để mô hình hoá dữ liệu bằng mô hình hồi qui tuyến tính với `tox` là biến phụ thuộc, và các biến `base`, `colour`, và `acid` là biến độc lập. Giải thích và diễn giải ý nghĩa của kết quả phân tích.

3. Các biến tiên lượng đều có ý nghĩa thống kê? Các bạn có thể loại bỏ từng biến một (nếu không có ý nghĩa thống kê) cho đến khi có mô hình sau cùng.

4. Tính nồng độ độc hại và khoảng tin cậy 95% cho một hoá chất với base=0.60, acid=0.95, và colour=0.52.

12.4 Mô hình hồi qui phi tuyến tính (non-linear regression)

Trong một thí nghiệm hoá học, các nhà khoa học đo lường ezyme activity của một hoá chất dưới nhiều nồng độ của một chất ức chế (inhibitor). Dữ liệu `inhib.xlsx` sau đây gồm các biến:

S: nồng độ của chất nền (substrate)
I: nồng độ của chất ức chế
R: Tỉ suất phản ứng

1. Vẽ biểu đồ tương quan giữa S (trục hoành) và R (trục tung). Tại sao biểu đồ này không mang tính minh hoạ? Giải thích.

```
grp = c(rep(1, times=12), rep(2, times=12), rep(3,
times=12))

plot(S, R, col=grp)

plot(S, R, pch=grp)
```

Khi không có ức chế, mối liên quan giữa S và R thường được mô tả bằng mô hình ay qui luật Michaelis-Menten:

$$R \approx \frac{V_{\max}.S}{K+S} \quad (1)$$

trong đó Vmax và K là hai tham số cần phải ước tính từ dữ liệu thí nghiệm. Phương pháp ước tính thường là least squares. Trong R, có hàm `nls` có thể dùng để ước tính tham số.

2. Tạo ra một dataset và gọi là dat0, bao gồm dữ liệu với I = 0. Sau đó, thử các lệnh sau đây:

```
mm0 = nls(R ~ Vmax*S / (K+S), start=list(Vmax=3,
K=100), data=dat0)

summary(mm0)
```

Cố gắng diễn giải ý nghĩa của output.

3. Thẩm định dao động dư:

```
plot(fitted(mm0), residuals(mm0))
```

Mô hình có phù hợp với dữ liệu?

4. Vẽ biểu đồ tương quan cho dữ liệu mà I=0. Thử các lệnh sau đây:

```
f = function(S) 2.98 * S / (35.802 + S)
plot(f, from=0, to=620, add=T)
```

Lệnh đầu tiên định nghĩa hàm số tiên lượng, còn lệnh 2 là thêm một đường biểu diễn vào biểu đồ tương quan.

Bây giờ, bạn muốn phân tích cho toàn bộ dữ liệu. Hãy xem xét hàm số sau đây:

$$R \approx \frac{V_{\max} S}{K_1\left(1+1/K_2\right)+S} \qquad (2)$$

Ở đây, $V_{\max}$, K_1, K_2 là các tham số cần phải ước tính từ dữ liệu thực tế.

5. Dùng nls để phân tích dữ liệu với mô hình trên. Chẳng hạn như các bạn có thể dùng V_{max} = 3, K_1 = 100, K_2 = 25 như là các giá trị khởi đầu (starting values). Ước số của các tham số là gì?

6. Vẽ biểu đồ tương quan như câu hỏi 1 một lần nữa, với các màu khác nhau cho các nồng độ của I. Thêm 3 đường biểu diễn vào biểu đồ, mỗi đường biểu diễn thể hiện một nồng độ ức chế.

7. Một cách khác, bạn có thể dùng phương trình Michaelis-Menten cho mỗi ức chế I, nhưng với các giá trị Vmax và K khác nhau. Mô hình này có thể ước tính bằng lệnh:

```
mm2 = nls(R ~ Vmax[grp]*S / (K[grp]+S),
start=list(Vmax=c(3,3,3), K=c(100,100,100)),
data=inhib)
```

8. Mô hình 2 thật ra là "nested" trong mô hình 1, Nếu gọi `mm1` là mô hình trong câu hỏi 5, thử so sánh:

```
anova(mm1, mm2)
```

Mô hình 2 phù hợp với dữ liệu hơn mô hình 1?

13 Hồi qui logistic và mô hình khác

13.1 Mô hình hồi qui logistic

Trong mô hình hồi qui logistic, biến phụ thuộc (y) là biến nhị phân (bivary variable) chỉ có 2 giá trị. Nói cách khác, y = 1 hoặc 0. Gọi p là xác suất x = 1, mô hình hồi qui logistic phát biểu rằng p liên quan với biến tiên lượng x theo hàm số:

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

Bài tập này sẽ được thực hành trên dữ liệu `coalworkers1.xlsx`. Số liệu thu thập từ một nhóm công nhân hầm mỏ để tìm hiểu mối liên quan giữa thời gian phơi nhiễm (`exposure`) và nguy cơ mắc bệnh:

exposure	normal	diseased
5.8	98	0
15	51	3
21.5	34	9
27.5	35	13
33.5	32	19
39.5	23	15
46	12	16
51.5	4	7

```
exposure = c(5.8, 15, 21.5, 27.5, 33.5, 39.5, 46, 51.5)
normal = c(98, 51, 34, 35, 32, 23, 12, 4)
diseased = c(0, 3, 9, 13, 19, 15, 16, 7)
coalworker1 = data.frame(exposure, normal, diseased)
```

1. Đọc dữ liệu vào R, và gọi tên dataset là `coalworker1`. Thử vài lệnh sau đây (và hiểu ý nghĩa):

   ```
   total = normal + diseased
   prob = diseased / total
   logodds = log(prob / (1-prob))
   plot(logodds ~ exposure, pch=16)
   ```

2. Phân tích dữ liệu bằng mô hình hồi qui logistic:

   ```
   status = matrix(c(diseased, normal), ncol=2)
   status
   logreg = glm(status ~ log(exposure), family=binomial)
   ```

   Trong lệnh trên `family=binomial` báo cho R biết rằng biến status là biến nhị phân.

3. Thử các lệnh sau đây:

```
summary(logreg)
abline(logreg)
```

4. Đọc dữ liệu `coalworkers2.xlsx` vào R và gọi dataset là coalworker2. Dùng hàm sau đây để phân tích mô hình logistic:

```
logreg2 = glm(y ~ log(exposures), data=coalworkers2,
family=binomial)

summary(logreg2)
```

Chú ý các bạn có cùng ước số với mô hình `logreg`.

5. Nếu một công nhân bị phơi nhiễm 30 năm, xác suất mà công nhân này mắc bệnh là bao nhiêu?

6. Phải phơi nhiễm bao nhiêu năm để có xác suất mắc bệnh là 50%?

13.2 Mô hình hồi qui logistic dùng package rms

Trong R có package "rms", rất quan trọng trong việc mô hình hoá các mối tương quan. Package có thể cung cấp nhiều thông số quan trọng của mô hình hồi qui logistic.

Trong phần này, các bạn sẽ làm việc với dữ liệu `parasite`, với biến infection là biến độc lập.

```
par = read.csv("~/Google Drive/TDT Projects/Workshop 12-
2015/Datafiles/Parasite.csv", header=T)

# mã hoá infection thành biến binary có giá trị 0/1

par$infected[par$infection == "present"] <- 1
par$infected[par$infection == "absent"] <- 0

# gọi package rms
library(rms)

logit = lrm(infected ~ sex + age + weight, x=T, y=T,
data=par)

logit
```

13.3 Đường cong ROC (ROC curve)

1. Mô hình hồi qui logistic thường được dùng để xây dựng mô hình tiên lượng. Một trong những chỉ số quan trọng của mô hình tiên lượng là diện tích dưới đường cong ROC. Trong R có một số hàm và lệnh giúp các bạn xây dựng một biểu đồ ROC qua package `Epi`, `ROCR` và `pROC`. Trong phần này, các bạn sẽ học cách xây dựng biểu đồ ROC từ dữ liệu thực tế:

```
# đọc dữ liệu vào R
```

```
par = read.csv("~/Google Drive/TDT Projects/Workshop 12-
2015/Datafiles/Parasite.csv", header=T)

# mã hoá infection thành biến nhị phân có giá trị 0/1

par$infected[par$infection == "present"] <- 1
par$infected[par$infection == "absent"] <- 0

# áp dụng mô hình hồi qui logistic

model1 = glm(infected ~ sex + age + weight, data=par,
family=binomial)

model2 = glm(infected ~ sex + age + I(age^2) +
I((weight>=12)*(weight-12)), data=par, family=binomial)

# Tính giá trị tiên lượng cho mô hình 1 và 2

par$pred1 = predict(model1, type="response")
par$pred2 = predict(model2, type="response")

# Dùng Epi, ROC dùng hồi qui logistic

library(Epi)
ROC(form = infected ~ sex+age+weight, plot="ROC")

ROC(form = infected ~ sex + age + I(age^2) +
I((weight>=12)*(weight-12)), plot="ROC")


# Dùng pROC, một package khác để xây dựng ROC

library(pROC)

rocobj1 = roc(par$infected, par$pred1)
rocobj2 = roc(par$infected, par$pred2)

auc(rocobj1); auc(rocobj2);
ci.auc(rocobj1); ci.auc(rocobj2)

plot.roc(par$infected, par$pred1, col="blue")
plot.roc(par$infected, par$pred1, col="red", add=T)

# smooth

plot.roc(smooth(rocobj1), col="blue")
plot.roc(smooth(rocobj2), col="red", add=T)
```

2. Các bạn hãy xây dựng mô hình hồi qui logistic để tiên lượng khả năng mang thai của một nghiên cứu về kĩ thuật IVF. Kết quả mang thai thể hiện qua biến số y (có giá trị 1 hoặc 0). Các biến tiên lượng là `age, bmi, cyclelen, fsh, smoking, antfoll, ovolume`.

```
# Đọc dữ liệu
```

```
ivf = read.csv("~/Google Drive/TDT Projects/Workshop 12-
2015/Datafiles/IVF.csv", header=T)

head(ivf)

# Mô hình hồi qui logistic

model1 = glm(y ~ age + bmi + fsh + antfoll + ovolume,
family=binomial, data=ivf)

model2 = glm(y ~ age + bmi + fsh + antfoll + ovolume +
smoking, family=binomial, data=ivf)

ivf$predicted1 = predict(model1, type="response")
ivf$predicted2 = predict(model2, type="response")

library(Epi); library(ROCR); library(pROC)

rocobj1 = roc(ivf$y, ivf$predicted1)
rocobj2 = roc(ivf$y, ivf$predicted2)

auc(rocobj1); auc(rocobj2);

ci.auc(rocobj1); ci.auc(rocobj2)

plot.roc(ivf$y, ivf$predicted1, col="blue")
plot.roc(ivf$y, ivf$predicted2, col="red", add=T)

# smooth

plot.roc(smooth(rocobj1), col="blue")
plot.roc(smooth(rocobj2), col="red", add=T)
```

13.4 Mô hình hồi qui Poisson

Ví dụ dưới đây là dữ liệu về số giải thưởng cho các học sinh trung học thuộc một trường trung học Los Angeles. Số liệu được thu thập cho 200 học sinh. Dữ liệu có 3 biến chính là: num_awards, prog, và math. Biến num_awards là biến phụ thuộc, và chúng ta muốn tiên lượng từ hai biến prog và math. Biến prog có 3 giá trị thể hiện cho 1=general, 2=academic, và 3=vocational.

Trước hết, đọc dữ liệu vào R:

```
aw = read.csv("~/Google Drive/TDT Projects/Workshop 12-
2015/Datafiles/poisson_sim.csv", head=T)

attach(aw)
head(aw)
hist(num_awards, col="blue", "border="white")
```

Áp dụng mô hình hồi qui Poisson:

```
mod = glm(num_awards ~ as.factor(prog) + math,
family="poisson", data=aw)
```

```
summary(mod)
```

Xem thử mô hình có tiên lượng kết quả giải thưởng tốt?

```
aw$pred = predict(mod, type="response")
aw = aw[with(aw, order(prog, math)),  ]

library(ggplot2)

p = ggplot(aw, aes(x=math, y=pred, colour=as.factor(prog)))

p = p + geom_point(aes(y = num_awards), alpha=0.5,
position=position_jitter(h=0.2))

p = p + geom_line(size=1) + labs("Math score", y="Expected
number of awards")
```

14 Phân tích đa biến (multivariate analysis)

14.1 Principal Component Analysis (PCA)

Các bạn sẽ làm việc với nghiên cứu về các loại cua trong dữ liệu `crabs` thuộc package MASS. Trong nghiên cứu này, các nhà khoa học đo lường chiều dài (CL), bề rộng (CW) và kích thước đầu (FL) của mỗi con cua. Có tất cả 200 con cua, 100 con màu xanh và 100 con màu da cam. Các nhà khoa học muốn dùng các biến CW, CL và FL để phân biệt hoặc tiên lượng màu cua.

1. Các bạn sẽ dùng đơn vị log.

   ```
   library(MASS)
   head(crabs)
   logcrabs = log(crabs[, 4:8])
   head(logcrabs)

   group = crabs$sex : crabs$sp
   group
   plot(logcrabs, col=group)
   ```

   Chú ý rằng 4 màu đen/đỏ/xanh lá cây/xanh dương thể hiện 4 loại cua female-blue / female-orange / male-blue / male-orange.

2. Dùng hàm `princomp` để phân tích yếu tố:

   ```
   pca = princomp(logcrabs, cor=T)
   pca
   summary(pca)

   plot(pca)
   loadings(pca)
   pca$scores
   ```

3. Thử một số lệnh sau đây:

```
ScoreData = data.frame(pca$scores)
plot(ScoreData)
plot(ScoreData, col=group)
plot(ScoreData[,1:3], col=group)
plot(logcrabs[,1], ScoreData[,1], col=group)
```

15 Xử lí dữ liệu trống (missing data)

R có package mice (multivariate imputation by chained equations) rất có ích trong việc xử lí các dữ liệu không đầy đủ hay nói chung là "missing". Cách sử dụng mice có thể minh hoạ qua các hàm sau đây.

Chúng ta có một dữ liệu trong file "test with missing data.csv" có một số số liệu không đầy đủ. Dữ liệu có 3 cột: Girth, Height, và Volume. Biến Volume được xem là biến phụ thuộc. Mục đích là tìm hiểu ảnh hưởng của Girth và Height đến Volume.

Để phân tích dữ liệu này, chúng ta cần cài đặt 2 packages: VIM và mice. Sau đó, các bạn có thể đọc dữ liệu vào R như sau:

```
dat = read.csv("~/Google Drive/TDT Projects/Workshop 12-
2015/Datafiles/test with missing data.csv", header=T)

attach(dat); head(dat)
```

Tìm hiểu xem xu hướng missing data là gì. Hàm sau đây sẽ báo cáo số lượng missing data cho từng biến trong dữ liệu:

```
md.pattern(dat)
```

Dùng margin plot trong VIM để xem các biến có missing data:

```
marginplot(dat[c(2,3)], col=c("blue", "red"))
```

Dùng mice để "điền vào" (impute) các giá trị trống. Option m=5 có nghĩa là tìm 5 giá trị khả dĩ cho mỗi giá trị trống:

```
dat2 = mice(dat, m=5)

cdat = complete(dat2, inc=TRUE)
cdat
```

Áp dụng mô hình hồi qui tuyến tính cho dữ liệu cdat (tức dữ liệu "hoàn chỉnh", không có missing data):

```
model1 = with(cdat, lm(Volume ~ Girth + Height))
summary(model1)
```

So sánh kết quả với mô hình có missing data:

```
model2 = lm(Volume ~ Girth + Height, data=dat)
summary(model2)
```

Bổ sung

```
source("http://pcwww.liv.ac.uk/~william/R/crosstab.r")

crosstab(Survey, row.vars = "Age", col.vars = "Sex", type = "f")

# Row percentages
crosstab(Survey, row.vars = "Age", col.vars = "Sex", type = "r")

# Column percentages
crosstab(Survey, row.vars = "Age", col.vars = "Sex", type = "c")

# Joint percentages (sums to 100 within final two table dimensions)
crosstab(Survey, row.vars = c("Age", "Sex"), col.vars = "Health", type =
"j")

# Total percentages (sums to 100 across entire table)
crosstab(Survey, row.vars = c("Age", "Sex"), col.vars = "Health", type =
"t")


# All margins
crosstab(Survey, row.vars = c("Age", "Sex"), col.vars = "Health", type =
"f")

# Calculate proportions rather than percentages
crosstab(Survey, row.vars = "Age", col.vars = "Sex", type = "t", percentages
= FALSE)

# Round output to 1 decimal place.
crosstab(Survey, row.vars = "Age", col.vars = "Sex", type = "t", percentages
= FALSE, dec.places = 1)

crosstab(Survey, row.vars = "Age")
crosstab(Survey, row.vars = "Age", col.vars = "Sex", type = c("f", "c"),
addmargins = FALSE)
crosstab(Survey, row.vars = "Age", col.vars = "Sex", type = c("f", "t"),
style = "wide", addmargins = FALSE)
crosstab(Survey, row.vars = "Age", col.vars = "Sex", type = c("f", "t"),
style = "long", addmargins = FALSE)
```