Chương 3: Dữ liệu và một số thao tác tiền xử lý dữ liệu

3.1 Phép toán, hàm tính toán và những tính toán đơn giản

round(x, digits = k) : Tính số làm tròn đến k chữ số thập phân

(Bảng đầy đủ trang 24)

3.2 Ma trận và phép toán trên ma trận

matrix(data, nrow, ncol)

data: vector phần tử
nrow: số dòng
ncol: số cột

3.3 Vector dữ liệu

3.3.1 Nhập dữ liệu

3.3.2 Biên tập dữ liệu

3.3.3 Truy cập dữ liệu

Lệnh	Cộng dụng
length(x)	Số phần tử
x[-i]	Đưa ra mọi phần tử trừ phần tử thứ i
x[i:j]	Đưa ra phần tử từ vị trí thứ i đến vị trí thứ j
x[c(i,j,k)]	Đưa ra phần tử ở vị trí i, j , k
x[x>a]	Đưa ra phần tử lớn hơn a
sum(x > a)	Tính số phần tử lớn hơn a
which(x == a)	Đưa ra những vị trí có giá trị bằng a
subset(x, subset)	Đưa ra tập con với điều kiện trong subset

3.3.4 Loại bỏ dữ liệu trống

> x = **na.omit**(x)

3.3.5 Chuyển dữ liệu số thành dữ liệu định danh

> x = c(2, 3, 0, 5, 1, 0, 7, 0, 2, 7)

>pl = x;

>pl[x < 2] = “it”

>pl[x >= 2 & x <= 5] = “bt”

>pl[x>5] = “nhieu”

3.3.6 Biến đổi vector định danh thành vector nhân tố

>pl = **factor**(pl, level = c(“it”, “bt”, “nhieu”))

3.4 Bảng dữ liệu

3.4.1 Nhập dữ liệu dạng bảng

> dulieu = data.frame(x, pl)

3.4.2 Biên tập dữ liệu

> dulieu = **edit**(dulieu)

> #Hoặc

> **fix**(dulieu)

3.4.3 Truy cập dữ liệu

A[i, j] **A[c(i, j, k),]** (Bảng đầy đủ dễ nhớ trang 37)

3.4.4 Loại bỏ dữ liệu trống

> dulieu = **na.omit**(dulieu)

3.5 Lưu dữ liệu

save(dulieu, file = “DuLieu.rda”)

3.6 Tạo dữ liệu

3.6.1 Tạo dãy số cách đều hàm seq

seq(from, to, by)

seq(length, from, by)

seq(length, from, to)

3.6.2 Tạo dãy lặp lại bằng hàm rep

rep(x, time)

x: vector lặp

time: số lần lặp / vector chỉ số lần lặp của mỗi phần tử trong x

3.6.3 Tạo dãy thứ bậc bằng hàm gl

gl(n, k ,length, labels)

n: Số chỉ số bậc

k: Số lần lặp lại của mỗi bậc

lengh: Chiều dài kết quả

labels: vector nhãn gán

3.7 Đọc dữ liệu có sẵn

3.8 Chọn mẫu ngẫu nhiên

sample(x, size, replace , prob)

x: vector gồn những thành phần chọn ngẫu nhiện

size: số phần tử lấy ra

replace: TRUE/ FALSE (Có hoàn lại / Không hoàn lại)

prop: vector cho biết xác suất được chọn của những phần tử trong x

Chương 4: Tóm tắt dữ liệu

- **Hàm tính tần số, tần suất**

Hàm	*Công dụng*
table(x)	Tính tần số của các phần tử trong x
prop.table(table(x), margin)	Tính tần suất của các phần tử trong x
cumsum(table(x))	Tần số tích lũy
cumsum(prop.table(table(x), margin))	Tần suất tích lũy

- **Hàm phân tổ dữ liệu**

cut(x, breaks, labels, right, include.lowest, dig.lab)

x: vector dữ liệu dạng số

breaks: vector số (ít nhất 2) gồm các điểm chia hoặc là một số nguyên dương (>= 2) chỉ số tổ

labels: nhãn của các tổ (mặc định NULL), các nhãn được xây dựng dưới dạng (a, b]

right: dạng logic, right = TRUE thì tổ có dạng (a, b] hoặc ngược lại [a, b), mặc định là right = TRUE

include.lowest: dạng logic, bằng TRUE thì tổ đầu chứa giá trị nhỏ nhất của các điểm chia (khi right = TRUE) hoặc tổ cuối chứa giá trị lớn nhất của các điểm chia (khi right = FALSE), mặc định bằng FALSE.

dig.lab: số nguyên dương chỉ số chữ số trong điểm chia (trong trường hợp không gán nhãn cho các khoảng chia) mặc định bằng 3.

Chương 6: Biến ngẫu nhiên

Hàm	**Công dụng**
mean(x)	Tính trung bình
median(x)	Tính trung vị
quantile(x)	Tính tứ phân vị
quantile(x, 0.8)	Tính phân vị thứ 80
sd(x)	Tính độ lệch chuẩn
var(x)	Tính phương sai

var(x) = sd(x)^2

Giá trị tới hạn Zα của phân phối chuẩn hóa

z(α) = qnorm(1- α)

Giá trị tới hạn t(n, **α)** của phân phối Student

t(n, α) = qt(1-α, n)

Chương 7: Khoảng tin cậy cho tham số một tổng thể

7.1 Khoảng tin cậy cho <u>trung bình một tổng thể</u>

Khoảng tin cậy cho trung bình μ

- Phân phối chuẩn, phương sai σ đã biết, khoảng tin cậy 100(1-**α**)% cho μ là :

  mean(x) – z(**α/2**). σ/ sqrt(n) < μ < mean(x) + z(**α/2**). σ/ sqrt(n)

- Phân phối chuẩn, phương sai σ chưa biết, khoảng tin cậy 100(1-**α**)% cho μ là :

  mean(x) – t(n-1,**α/2**). S(x)/ sqrt(n) < μ < mean(x) + t(n-1,**α/2**). S(x)/ sqrt(n)

  S(x): Phương sai mẫu

- **Khoảng tin cậy cho trung bình tổng thể**
  - Phương sai đã biết (sigma.x: Độ lệch chuẩn tổng thể)
    - Dữ liệu sơ cấp: **z.test**(x, sigma.x, conf.level)
    - Dữ liệu thứ cấp: **zsum.test**(mean.x, sigma.x, n.x, conf.level)
  - Phương sai chưa biết (Biết s.x là độ lệch chuẩn của mẫu)
    - Dữ liệu sơ cấp: **t.test**(x, conf.level)
    - Dữ liệu thứ cấp: **tsum.test**(mean.x, s.x, n.x, conf.level)

7.2 Khoảng tin cây cho tỉ lệ tổng thể

prop.test(x, n, conf.level, correct)

x: số lần thành công

n: Số lần thử

conf.level: số thuộc [0,1] chỉ số tin cậy của khoảng ước lượng

correct: tham số logic, có hay không sự điều chỉnh liên tục Yate (True)

Chương 8: Kiểm định tham số

Bước làm bài kiểm định tham số

Bước 1: Thiết lập Ho H1

Bước 2: Xác định hàm kiểm định

Bước 3: Thực hiện kiểm định

Bước 4: Đọc kết quả và đưa ra kết luận

Kết luận: So sánh p-value với mức ý nghĩa α

- **p-value < α : Bác bỏ Ho**
- **p-value >= α : Không bác bỏ Ho**

8.1 Kiểm định giả thuyết về tham số một tổng thể

8.1.1 Kiểm định giả thuyết về trung bình *một* tổng thể

- **Gía trị thống kê**
  - Phương sai đã biết: z = (mean(x) – mu_o)/ sigma.x / sqrt(n)
  - Phương sai chưa biết: t = (mean(x) – mu_o)/ s.x / sqrt(n)

Ho	Bác bỏ Ho	
mu >= mu_o	z > zα	t > t(n-1,α)
mu <= mu_o	z < zα	t < t(n-1,α)
mu == mu_o	|z| > z(α/2)	|t| > t(n-1,α/2)

Chú ý: library(BSDA)

Kiểm định trung bình một tổng thể

- Biết phương sai tổng thể
  - Dữ liệu sơ cấp : **z.test**(x, sigma.x, mu, alt)
  - Dữ liệu thứ cấp : **zsum.test**(mean.x, sigma.x, n.x, mu, alt)
- Không biết phương sai tổng thể
  - Dữ liệu sơ cấp: **t.test**(x, mu, alt)
  - Dữ liệu thứ cấp: **tsum.test**(mean.x, s.x, n.x, mu, alt)

8.1.2 Kiểm định giả thuyết về tỉ lệ một tổng thể

prop.test(x, n, p, alt, correct)

x: số lần thành công

n: số lần thử

p: xác suất thành công

correct: có sự điều chỉnh liên tục Yate không, mặc định TRUE

8.2 Kiểm định giả thuyết về tham số hai tổng thể

Kiểm định trung bình hai tổng thể

- Trọn mẫu **độc lập**
  - Đã biết phương sai
    - Dữ liệu sơ cấp: **z.test**(x, y, sigma.x, sigma.y, mu, alt)
    - Dữ liệu thứ cấp: **zsum.test**(mean.x, sigma.x, n.x, mean.y, sigma.y, n.y, mu, alt)
  - Chưa biết phương sai
    - Có giả thuyết phương sai bằng nhau
      - Dữ liệu sơ cấp: **t.test**(x, y, mu, alt, var.equal = TRUE)
      - Dữ liệu thứ cấp: **tsum.test**(mean.x, s.x, n.x, mean.y, s.y, n.y, mu,alt,var.equal=TRUE)
    - Không có giả thuyết phương sai bằng nhau:
      - Thay "**var.equal = FALSE**"
- Trọng mẫu **ghép cặp**
  - Dữ liệu sơ cấp: **t.test**(x, y, mu, alt, paired = TRUE)
  - Dữ liệu thứ cấp: **tsum.test**(mean.x, s.x, n.x, mean.y, s.y, n.y, mu, alt, paired =TRUE)

Kiểm định giả thuyết về tỉ lệ hai tổng thể

prop.test(x, n, alt, correct)

x: vector chỉ số lần thành công trong mỗi mẫu

n: vector chỉ soosl ần thử trong mỗi mẫu

alt, correct

Kiểm định giả thuyết về phương sai hai tổng thể

var.test(x, y, ratio, alt)

x: vector chỉ số lần thành công trong mỗi mẫu

n: vector chỉ soosl ần thử trong mỗi mẫu

ratio: tỉ số được giả thuyết của phương sai 2 tổng thể (mặc định là 1)

alt

Chương 9: Phân tích phương sai

9.1 Phân tích phương sai một nhân tố

oneway.test(formula, data, var.equal)

formula: Công thức dạng y ~ g, y là vector dữ liệu biến định lượng, g là vector dữ liệu biến định tính

data: Bảng dữ liệu 2 biến y, g

var.equal: Phương sai có bằng nhau hau không

(ví dụ trang 179)

Hàm khác phân tích phương sai cho kết quả chi tiết hơn anova

anova (lm(y ~ g), data) (anova = aov)

Chú thích:

Df: bậc tự di ở tử và mẫu của thống kê F

Sum Sq: Tổng các bình phương

Mean Sq: Bình phương trung bình

F value: tỷ số F

Pr (>F): p-value

9.2 Phân tích sâu

TukeyHSD(**aov**(y~ g, data), which, ordered, conf.level)

y,g: Như oneway.test

which: vector xâu lý tự gồm tên những biến nguyên nhân

ordered: tham số logic điều chỉnh thứ tự các cặp trung bình được so sánh

conf.level: độ tin cậy

Chú thích:

diff: : chênh lệch giữa các trung bình mẫu

lwr, npr: cận dưới và cận trên của khoảng tin cậy cho số hiệu từng cặp trung bình

p adj: p-value

(Ví dụ trang 181)

Chương 10 Kiểm định phi tham số

Kiểm định phi tham số

- **Kiểm định trung vị một tổng thể**
  - **wilcox.test**(x, mu, alt, exact, correct, conf.int, conf.level)
- K**iểm định trung vị hai tổng thể**
  - **wilcox.test**(x, y, mu, alt, exact, paired, correct, conf.int, conf.level)

Trong đó:

x:	vector dữ liệu mẫu thứ nhất
y:	vector dữ liệu mẫu thứ hai
mu, alt	
exact:	tham số dạng logic xét xem p-value có được tính chính xác không
paired:	tham số logic, có ghép cặp hay không
correct:	tham số dạng logic xét xem khi tính p-value bằng cách xấp xỉ phân phối chuẩn cáp áp dụng điều chỉnh liên tục không
conf.int:	tham số dạng logic xét xem có đưa ra khoảng tin cậy cho trung vị tổng thể không
conf.level:	độ tin cậy của khoảng ước lượng

Kiểm định trung vị nhiều tổng thể

Kruskal.test(x, g)

x:	vector dữ liệu
g:	vector thứ bậc phân loại các phần tử trong x

Chương 11 Kiểm định khi-bình phương

11.1 Kiểm chứng tính độc lập

chisq.test(A)

A: ma trận dữ liệu

11.2 Kiểm chứng mức phù hợp của một phân phối

chisq.test(x, p)

x: vector dữ liệu

p: vector cùng chiều dài với x chỉ xác suất của phân phối cần kiểm chứng