R

1 Dẫn nhập

Trong bài thực hành dưới đây, chúng ta sẽ làm quen với R. Mục tiêu của bài thực hành là giúp sinh viên có kiến thức cơ bản trong việc sử dụng R phục vụ cho các công việc nghiên cứu trong tương lai.

2 Tại sao dùng R?

- R là một chương trình được sử dụng rộng rãi để phân tích thống kê.
- R là phần mềm miễn phí.
- Vẽ được các biểu đồ có thể công bố trên các tập san quốc tế.

3 Cài đặt

Cho HĐH Windows:

- Download và cài đặt R từ `www.r-project.org`

Phiên bản hiện tại của R là 3.2.2. Sau khi cài đặt, R có thể được bắt đầu như bất kỳ ứng dụng khác cho Windows.

4 Văn phạm trong R

R phân biệt lệnh viết bằng chữ hoa hay chữ thường. Ví dụ, R phân biệt giữa `Library` và `library`
Khi có chữ rời nhau, R thường dùng dấu chấm để thay vào khoảng trống, chẳng hạn như `data.frame, t.test, read.table` ...
Nếu gõ đúng văn phạm, thì R sẽ cho chúng ta một cái prompt khác hay cho ra kết quả nào đó tùy theo lệnh. Nếu lệnh không đúng, R sẽ cho ra một thông báo ngắn là không đúng hoặc không hiểu. Ví dụ, nếu chúng ta gõ

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

thì R sẽ hiểu, rồi cho chúng ta một prompt khác: `>`. Nhưng nếu chúng ta gõ

```
> Discrete Structures
```

R sẽ báo lỗi với lệnh này, một thông báo sau đây sẽ xuất hiện:

```
Error: unexpected symbol in "Discrete Structures"
>
```

Văn phạm chung của R là một lệnh (command) hay hàm (function). Theo sau hàm là những thông số mà chúng ta phải cung cấp. Ví dụ:

```
> setwd("D:/DiscreteStructure/Lab1")
```

thì `setwd` là một hàm, còn `"D:/DiscreteStructure/Lab1"` là thông số của hàm.
Để tra cứu thông tin chi tiết của các hàm trong R, ta dùng `help(tenham)` hoặc `?tenham`. Ví dụ, nếu muốn tra cứu thông tin về hàm `plot()`, ta thực hiện như sau

```
> help(plot)
```

hoặc

```
> ?plot
```

R là một ngôn ngữ hướng đối tượng (object oriented language), nghĩa là các dữ liệu trong R được chứa trong object. Dùng ký hiệu `<-` hoặc dấu `=` để gán các dữ liệu chứa trong đối tượng. Ví dụ

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

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

5 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. Chẳng hạn R chấp nhận `myobject` nhưng không chấp nhận `my object`.

```
> myobject <- rnorm(10)
> my object <- rnorm(10)
Error: unexpected symbol in "my object"
```

Đôi khi tên `myobject` khó đọc, nên chúng ta nên tách rời bằng ".". Ví dụ `my.object`
Lưu ý:

- Không nên đặt tên object hay variable bằng ký 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 ký tự $ như sau: `data$age`, và trong trường hợp đó, `age <- data$age` có thể chấp nhận được.

6 Môi trường vận hành trong R

Dữ liệu phải được chứa trong một thư mục (directory) của máy tính, chẳng hạn như `D:\DiscreteStructures\Lab1`
Để R biết dữ liệu nằm ở đâu, ta sử dụng lệnh `setwd` (set working directory) như sau:

```
> setwd("D:/DiscreteStructures/Lab1")
```

Lệnh trên báo cho R biết là dữ liệu sẽ chứa trong directory có tên là `D:\DiscreteStructures\Lab1`.
Chú ý, R dùng forward slash "/" chứ không phải backward slach "\".
Để biết hiện nay R đang làm việc ở directory nào, ta viết như sau:

```
> getwd()
```

7 Các phép toán số học và logic

Phép toán	Mô tả
+	cộng
-	trừ
*	nhân
/	chia
^ hoặc **	lũy thừa
$x\%\%y$	lấy phần dư của (x chia y), $5\%\%2 = 1$
$x\%/\%y$	lấy phần nguyên của (x chia y), $5\%/\%2 = 2$

Bảng 1: Các phép toán số học

Phép toán	Mô tả
<	bé hơn
<=	bé hơn hoặc bằng
>	lớn hơn
>=	lớn hơn hoặc bằng
==	so sánh bằng
!=	so sánh khác
!x	khác x
x | y	x Hoặc y
x & y	x Và y

Bảng 2: Các phép toán logic

Ví dụ 1:

```
> (2-5)*2 + 3 - 2^2
[1] -7
> 2<3
[1] TRUE
```

8 Các hàm số cơ bản

Phép toán	Mô tả
`abs(x)`	giá trị tuyệt đối của x
`sqrt(x)`	căn bậc hai của x
`sqrt(x,y)`	căn bậc y của x
`ceiling(x)`	lấy số nguyên nhỏ nhất lớn hơn x
`floor(x)`	lấy số nguyên lớn nhất nhỏ hơn x
`trunc(x)`	cắt bỏ phần thập phân
`round(x, digits = n)`	làm tròn tới n chữ số sau dấu thập phân
`signif(x, digits = n)`	làm tròn n chữ số sau dấu thập phân
`cos(x), sin(x), tan(x)`	các hàm lượng giác
`acos(x), cosh(x), acosh(x)`	các hàm lượng giác ngược
`log(x)`	lấy logarithm theo cơ số tự nhiên
`log10(x)` hoặc `log2(x)`	lấy logarithm theo cơ số 10 hoặc 2
`exp(x)`	e^x
`factorial(x)`	tính giai thừa

Bảng 3: Các hàm số

Ví dụ 2:

```
> sqrt(9)
[1] 3
> exp(2)
[1] 7.389056
```

9 Tạo và thao tác với vectors trong R

Để tạo một vector trong R ta sử dụng `c()` (viết tắt của chữ *concatenation*). Ví dụ tạo vector $x = (2, 4, -4, 7, 10, 11)$, ta thực hiện

```
> x <- c(2, 4, -4, 7, 10, 11)
```

Các phép toán số học tác động tới vector tương ứng là tác động tới các phần tử trong vector đó. Ví dụ, thực hiện phép nhân và chia với x như trên

```
> x*x
[1] 4 16 16 49 100 121
> x/x
[1] 1 1 1 1 1 1
```

Để tạo vector có dạng chuỗi số, ta dùng ":" hoặc `seq()`. Ví dụ, muốn tạo vector $y = (1, 2, 3, 4, 5)$ thì câu lệnh sẽ là

```
> y <- 1:5
[1] 1 2 3 4 5
```

Hoặc, nếu muốn tạo một chuỗi từ 0 đến 5 với bước nhảy 0.5 ta dùng

```
> y <- seq(0, 2, 0.5)
[1] 0.0 0.5 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
```

Ngoài ra, ta có thể dùng

```
> seq(0, 5, length=5)
[1] 0.00 1.25 2.50 3.75 5.00
```

để tạo một vector gồm 10 phần tử là các số cách đều trong [0, 5].
Một hàm hữu ích khác là `rep()` (viết tắt của từ replicate)cho phép lặp lại các phần tử của vector.

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

Khi muốn truy xuất phần tử có vị trí i của một vector x, ta dùng `x[i]`. Ví dụ, ta muốn truy xuất phần tử thứ 4 của vector `x`, ta viết như sau

```
> x <- c(2, 4, -4, 7, 10, 11)
> x[4]
[1] 7
```

Trong trường hợp, muốn truy xuất các phần tử của vector sao cho chúng thỏa điều kiện nào đó thì sử dụng `x[điều kiện]`, ví dụ truy xuất các phần tử chẵn trong vector `x` ta gõ

```
> x[x %% 2 == 0]
[1]  2  4 -4 10
```

Muốn sắp xếp các phần tử của vector `x` ta dùng hàm `sort()`

```
> sort(x, decreasing=TRUE)
[1] 11 10  7  4  2 -4
> sort(x, decreasing=FALSE)
[1] -4  2  4  7 10 11
```

Muốn biết độ dài của một vector, ta dùng hàm `length()`

```
> length(x)
[1] 6
```

Muốn tính tổng các phần tử của vector `x`, ta dùng hàm `sum()`

```
> sum(x)
[1] 30
```

10 Cách nhập dữ liệu

10.1 Cách nhập trực tiếp bằng hàm `c()`

Chúng ta dùng hàm `c()` để nhập dữ liệu. Mỗi số liệu được cách nhau bằng một dấu phẩy.
Ví dụ 3: chúng ta có dữ liệu về tên và điểm thi đại học của 10 bạn như sau

```
An      25.5
Binh    26.0
Cuong   25.5
Dung    28.5
Hue     27.0
Hung    27.0
Khanh   27.5
Lan     29.0
Mai     28.0
Nhung   28.0
```

Chúng ta nhập dữ liệu trên vào R như sau

```
> name <- c("An", "Binh", "Cuong", "Dung", "Hue", "Hung", "Khanh", "Lan", "Mai", "Nhung")
> grade <- c(25.5, 26.0, 25.5, 28.5, 27.0, 27.0, 27.5, 29.0, 28.0, 28.0)
```

Lệnh thứ nhất cho R biết rằng chúng ta muốn tạo một cột dữ liệu (hay biến số) có tên là `name`, lệnh thứ hai tạo ra một cột khác có tên là `grade`.
Vì `name` và `grade` là hai đối tượng riêng lẻ, nên chúng ta cần phải nhập hai đối tượng này thành một `data.frame` để R có thể xử lý sau này. Để làm việc này, ta cần đến hàm `data.frame`

```
> data <- data.frame(name, grade)
```

Trong lệnh trên, ta cho R biết rằng nhập hai cột `name` và `grade` vào một đối tượng có tên là `data`.
Để kiểm tra xem trong `data` có gì, ta chỉ cần gõ

```
> data
```

Kết quả sẽ là

```
    name grade
1     An  25.5
2   Binh  26.0
3  Cuong  25.5
4   Dung  28.5
5    Hue  27.0
6   Hung  27.0
7  Khanh  27.5
8    Lan  29.0
9    Mai  28.0
10 Nhung  28.0
```

10.2 Nhập dữ liệu từ Excel

Để nhập dữ liệu từ phần mềm Excel, ta cần tiến hành 2 bước

- Bước 1: Lưu dữ liệu dưới dạng "csv"
- Bước 2: Dùng R (lệnh `read.csv`) để nhập dữ liệu.

Ví dụ 4: Chúng ta có một dữ liệu dạng excel có tên `diemthi.xls` được đặt tại thư mục `D:\DS\Lab1`

name	grade
An	25.5
Binh	26.0
Cuong	25.5
Dung	28.5
Hue	27.0
Hung	27.0
Khanh	27.5
Lan	29.0
Mai	28.0
Nhung	28.0

Bảng 4: Dữ liệu từ Excel

Bước 1: Ta lưu lại dưới dạng csv, `diemthi.xls` thành `diemthi.csv`
Bước 2: vào R và ra các lệnh sau đây

```
> setwd("D:/DS/Lab1")
> dt <- read.csv("diemthi.csv", header=TRUE)
```

Lệnh thứ hai ở trên yêu cầu R đọc dữ liệu từ "diemthi.csv", và lưu dữ liệu này trong một object có tên là `dt`.

10.3 Xem thông tin cơ bản về dữ liệu

Giả sử ta đã nhập dữ liệu vào một data.frame có tên là `data`. Để tìm hiểu trong dữ liệu này gồm bao nhiêu cột, bao nhiêu dòng, ta dùng lệnh `dim()` (`dim` viết tắt của từ dimension)

```
> dim(data)
[1] 10  2
```

R cho chúng ta thấy có 10 dòng và 2 cột. Muốn biết những biến số này tên gì, ta dùng lệnh `names()`

```
> names(data)
[1] "name"  "grade"
```

Trong biến `grade`, chúng ta muốn biết có bao nhiêu bạn cùng đạt số điểm, ta dùng lệnh `table()`

```
> table(data$grade)
25.5   26   27 27.5   28 28.5   29
   2    1    2    1    2    1    1
```

Kết quả cho thấy có 2 bạn được 25.5, 1 bạn được 26, ...

11 Xử lý thống kê cơ bản

Để tính các đặc trưng của một biến ngẫu nhiên (mẫu ngẫu nhiên, vector) ta dùng các hàm như sau

Hàm	Mô tả
`mean()`	trung bình mẫu
`var()`	phương sai mẫu
`sd()`	độ lệch chuẩn mẫu
`median()`	trung vị mẫu
`min()`	giá trị nhỏ nhất
`max()`	giá trị lớn nhất
`range()`	miền giá trị
`summary()`	tổng hợp sơ lược về mẫu

Bảng 5: Các hàm thống kê

Tiếp tục ví dụ 3, ta muốn tính giá trị trung bình điểm thi đại học của 10 bạn đó, ta gõ như sau

```
> mean(data$grade)
[1] 27.2
```

12 Nội dung thực hành

Câu 1.

- Tạo một vector có tên là `vector1` gồm 5 phần tử (2, 2, -1, 4, 5, 3, 4, -1, 5, 0).
- Tạo một vector có tên là `vector2` gồm 5 phần tử (3, 0, -2, 4, 1, 2, 4, 3, 5, 1).

- Thực hiện các phép cộng, trừ, nhân, chia, lũy thừa cho hai vector `vector1` và `vector2`
- Tổng các phần tử `vector1` có bé hơn tổng các phần tử `vector2` hay không?
- Truy xuất các phần tử chia 3 dư 1 trong vector `vector2`.
- Gán biến `sub = vector1 - vector2`. Đếm số lượng phần tử bằng 0 của vector `sub`.
- Sắp xếp `vector1` theo thứ tự tăng dần và `vector2` theo thứ tự giảm dần.

Câu 2.
Thống kê về số lượng được "likes" (thích) trên trang cá nhân mạng xã hội facebook của 25 người trong một ngày được cho như sau

name	likes	name	likes	name	likes	name	likes	name	likes
P1	29	P6	20	P11	16	P16	18	P21	13
P2	23	P7	16	P12	21	P17	25	P22	19
P3	25	P8	9	P13	23	P18	19	P23	17
P4	15	P9	17	P14	20	P19	19	P24	29
P5	16	P10	13	P15	16	P20	13	P25	13

Bảng 6: Số lượng lượt được thích trên trang cá nhân facebook của 25 người trong một ngày

- Nhập dữ liệu từ bảng trên vào R với hai cột dữ liệu có tên là `name` và `likes`. Sau đó nhập hai cột này vào một đối tượng có tên là `fb`.
- Hãy tính giá trị trung bình, phương sai, độ lệch chuẩn, trung vị, giá trị lớn nhất, giá trị nhỏ nhất của số lượng được "likes" của 25 người trên.
- Vẽ biểu đồ phân bố (histogram) cho số lượng được "thích" của 25 người đó. (Gợi ý: dùng lệnh `help` để tra cứu cách dùng hàm `hist`)