

Thực hành xác suất thống kê

# Bài 1: GIỚI THIỆU VỀ R

# 1. Đôi nét về R

- R là phần mềm sử dụng cho phân tích thống kê và vẽ biểu đồ, nhưng về bản chất, R là một ngôn ngữ máy tính đa năng, có thể sử dụng cho nhiều mục tiêu khác nhau, từ tính toán đơn giản cho đến các phân tích thống kê phức tạp. Ngoài ra, vì R là một ngôn ngữ, 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.
- Một trong những nhược điểm nhưng cũng là ưu điểm là R là môi trường làm việc với những dòng lệnh, do đó, ta có thể thấy được và lưu lại đường như toàn bộ quá trình, từ đó ta có thể sử dụng lại, tìm lỗi, sửa lỗi,...

## 2. Tải và cài đặt R

Để sử dụng R, việc đầu tiên là chúng ta phải cài đặt R vào máy của mình. Để làm được việc này, ta tải phần mềm bằng cách truy cập vào trang web: <http://cran.us.r-project.org>

## 3. Cấu trúc lệnh chung trong R

### 3.1 Cấu trúc lệnh trong R

**Đối tượng**  $\leftarrow$  **hàm** (thông số 1, thông số 2,...)

hay

**Đối tượng** = **hàm** (thông số 1, thông số 2,...)

**Ví dụ:**

```
x <- c()
```

```
x <- sample(5)
```

x ở đây là 1 đối tượng, còn **c()** là 1 hàm nhưng không có thông số, **sample(5)** là 1 hàm với thông số định lượng là 5.

## 3. Cấu trúc lệnh chung trong R

### 3.1 Cấu trúc lệnh trong R

Để biết rõ ràng hơn về hàm nào đó (ý nghĩa, công dụng, khả năng xử lý,...) ta thực hiện

**?tên hàm** hoặc **help(tên hàm)**

Để R biểu diễn các ví dụ của 1 hàm

**example(tên hàm)**

## 3. Cấu trúc lệnh chung trong R

### 3.1 Cấu trúc lệnh trong R

- R còn là một ngôn ngữ “đối tượng”, các dữ liệu trong R được chứa trong object, tức là khi ta khởi tạo 1 biến thì R sẽ tự động tạo ra 1 vùng lưu giá trị của biến.
- **Lưu ý:** Đối tượng ở đây rất đa dạng, có thể là số, véc-tơ, ma trận, chuỗi,....

- **Ví dụ:**

```
> x <- 5
```

```
> y <- 6
```

```
> x + y
```

```
[1] 11
```

```
> z + y
```

```
Error: object 'z' not found
```

- **Ví dụ:**

```
> (z=5) + (t = 7) - (x + y)
```

```
[1] 1
```

### 3. Cấu trúc lệnh chung trong R

#### 3.2 Một số phép toán so sánh hay logic trong R

<code>x == 3</code>	<code>x</code> bằng 3
<code>x != 3</code>	<code>x</code> khác 3
<code>x &lt; y</code>	<code>x</code> nhỏ hơn <code>y</code>
<code>x &gt; y</code>	<code>x</code> lớn hơn <code>y</code>
<code>x &lt;= y</code>	<code>x</code> nhỏ hơn hay bằng <code>y</code>
<code>x &gt;= y</code>	<code>x</code> lớn hơn hay bằng <code>y</code>
<code>z &lt;= 5</code>	<code>z</code> nhỏ hơn hay bằng 5
<code>z &gt;= 5</code>	<code>z</code> lớn hơn hay bằng 5
<code>is.na(x)</code>	có phải <code>x</code> là biến trống không
<code>A &amp; B</code>	<code>A</code> và <code>B</code> (so sánh chân trị (AND) trên mỗi thành phần tương ứng của 2 vectơ <code>A</code> và <code>B</code> )
<code>A &amp;&amp; B</code>	<code>A</code> và <code>B</code> (so sánh chân trị (AND) trên thành phần đầu tiên kể từ trái qua phải của 2 vectơ <code>A</code> và <code>B</code> )
<code>A   B</code>	<code>A</code> và <code>B</code> (so sánh chân trị (OR) trên mỗi thành phần tương ứng của 2 vectơ <code>A</code> và <code>B</code> )
<code>A    B</code>	<code>A</code> và <code>B</code> (so sánh chân trị (AND) trên thành phần đầu tiên kể từ trái qua phải của 2 vectơ <code>A</code> và <code>B</code> )
<code>!A</code>	phủ định <code>A</code> (NOT <code>A</code> )
<code>xor(x, y)</code>	lấy XOR trên mỗi thành phần tương ứng của 2 vectơ <code>x</code> và <code>y</code>



### 3. Cấu trúc lệnh chung trong R

#### 3.2 Một số phép toán so sánh hay logic trong R

- **Ví dụ:**

```
> x <- 5
```

```
> x == 5
```

```
[1] TRUE
```

```
> x == 6
```

```
[1] FALSE
```

```
> !(x == 5)
```

```
[1] FALSE
```

```
> A <- c(1,2,3); B <- c(3,0,4)
```

```
> A
```

```
[1] 1 2 3
```

```
> B
```

```
[1] 3 0 4
```

```
> A&B
```

```
[1] TRUE FALSE TRUE
```

### 3. Cấu trúc lệnh chung trong R

#### 3.2 Một số phép toán so sánh hay logic trong R

- **Lưu ý 1:** Ta có thể gán 1 giá trị mới đề lên 1 biến nào đó đã được định nghĩa.

- **Ví dụ:**

```
> x <- 5
```

```
> y <- 6
```

```
> x + y
```

```
[1] 11
```

```
> x <- 9
```

```
> y <- 4
```

```
> x + y
```

```
[1] 13
```

### 3. Cấu trúc lệnh chung trong R

#### 3.2 Một số phép toán so sánh hay logic trong R

- **Lưu ý 2:** Ta có thể thực hiện cùng 1 lúc nhiều lệnh trên cùng 1 dòng, chỉ cần ngăn cách nhau giữa các lệnh bởi dấu chấm phẩy “;”.

- **Ví dụ:**

```
> x <- 2; y <- 3
```

```
> x
```

```
[1] 2
```

```
> y
```

```
[1] 3
```

```
> x + y
```

```
[1] 5
```

### 3. Cấu trúc lệnh chung trong R

#### 3.2 Một số phép toán so sánh hay logic trong R

- **Lưu ý 3:** Ta có thể sử dụng ký hiệu # để ghi chú.
- **Ví dụ:**  
> #Lệnh sau đây mô tả việc tạo ra 1 dãy số ngẫu nhiên gồm 8 số có giá trị từ 1 tới 12:  
> k = sample(12,8)  
> k  
[1] 9 7 5 3 8 2 1 11

### 3. Cấu trúc lệnh chung trong R

#### 3.3 Một số hàm toán học thường dùng trong R

<code>log(x)</code>	logarit cơ số $e$
<code>log10(x), log(x, n)</code>	logarit cơ số 10, cơ số $n$ của $x$ .
<code>exp(x)</code>	$e^x$
<code>sqrt(x)</code>	căn bậc 2 của $x$
<code>factorial(x)</code>	$x!$
<code>choose(n, k)</code>	tổ hợp $n$ chập $k$
<code>floor(x)</code>	giá trị nguyên $< x$ (sàn của $x$ )
<code>ceiling(x)</code>	giá trị nguyên $> x$ (trần của $x$ )
<code>trunc(x)</code>	làm tròn tới giá trị nguyên gần nhất giữa $x$ và 0
<code>round(x, digits = n)</code>	làm tròn $x$ đến $n$ chữ số
<code>signif(x, digits = n)</code>	hiển thị $x$ dưới dạng dấu chấm thập phân, $n$ tổng chữ số hiển thị
<code>sin(x), cos(x), tan(x)</code>	hàm sin, cos, tan
<code>abs(x)</code>	$ x $
<code>x%/%y</code>	lấy phần nguyên của phép chia $x/y$
<code>x%%y</code>	lấy phần dư của phép chia $x/y$

## 3. Cấu trúc lệnh chung trong R

### 3.3 Một số hàm toán học thường dùng trong R

#### Một số lệnh liên quan

<code>length(x)</code>	chiều dài của $x$ .
<code>x[i]</code>	phần tử thứ $i$ của mảng $x$ .
<code>x[-i]</code>	tất cả các phần tử của $x$ trừ phần tử thứ $i$ ra.
<code>x[1 : 5]</code>	trích $x_1$ cho đến $x_5$ .
<code>x[c(1, 3, 5)]</code>	trích các phần tử thứ 1, 3 và 5.
<code>x[x &gt; 3]</code>	trích tất cả những phần tử lớn hơn 3.
<code>x[x &lt; -2   x &gt; 2]</code>	trích những phần tử $ x  > 2$ .

#### Một số hàm về vectơ: Cho vectơ $x$

<code>max(x), min(x)</code>	giá trị lớn nhất, bé nhất của $x$ .
<code>sum(x)</code>	tổng các giá trị trong $x$ .
<code>mean(x)</code>	trung bình của $x$ .
<code>median(x)</code>	trung vị của $x$ .
<code>range(x)</code>	bằng $\max(x) - \min(x)$ .
<code>var(x)</code>	phương sai của $x$ .
<code>sort(x)</code>	sắp xếp $x$ , mặc định theo thứ tự tăng dần.
<code>order(x)</code>	trả về các vị trí của $x$ khi đã sắp theo thứ tự tăng dần.
<code>quantile(x)</code>	tính các phân vị của $x$ .
<code>cumsum(x)</code>	tổng tích lũy.
<code>cumprod(x)</code>	tích tích lũy.

## 3. Cấu trúc lệnh chung trong R

### 3.3 Cách đặt tên trong R

- Đặt tên 1 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.

- **Ví dụ:**

```
> myobject <- sample(6,4)
```

```
> my object <- sample(6,4)
```

Error: unexpected symbol in "my object"

- Đôi khi để việc đặt tên gợi nhớ dễ hơn, ta có thể tách rời các ký tự ra bởi dấu chấm.

- **Ví dụ:**

```
> my.object <- sample(6,4)
```

## 3. Cấu trúc lệnh chung trong R

### 3.3 Cách đặt tên trong R

- **Lưu ý:** Trong R, có phân biệt mẫu tự viết hoa và mẫu tự viết chữ thường, tức là **My.object** khác với **my.object**.
- **Vài lưu ý khác**
  - Không nên đặt tên biến hay đối tượng mà trong đó có sự xuất hiện của các dấu “\_” (underscore) như my\_object hay my-object.
  - Không nên đặt tên biến hay đối tượng trùng lặp 1 biến trong dữ liệu.



## 4. Bước đầu làm quen với R

### 4.1 Thiết lập thư mục làm việc

- Lệnh thiết lập thư mục làm việc:

`setwd("tên ổ đĩa:/tên thư mục làm việc")`

- Ví dụ:

```
> setwd("E://R_Works")
```

- Hay trên thanh công cụ, ta vào File → Change dir... và ta chọn folder mà ta muốn trở thành môi trường làm việc và các thao tác trên R sẽ được thực hiện và lưu tại đó.
- Lệnh xem môi trường làm việc cho R hiện thời

`getwd()`

- Ví dụ:

```
> getwd()
```

```
[1] "E:/R_Works"
```

## 4. Bước đầu làm quen với R

### 4.1 Thiết lập thư mục làm việc

- Liệt kê các file trong thư mục làm việc

```
> list.files()
```

```
[1] "data"          "Solieu.txt"
```

hay

```
> dir()
```

```
[1] "data"          "Solieu.txt"
```

- Lưu workspace đang làm việc (với tên là tên\_file) để tiện cho việc xem lại, làm việc tiếp tục sau này

```
save.image("tên_file.rda")
```

## 4. Bước đầu làm quen với R

### 4.1 Thiết lập thư mục làm việc

- Khôi phục biến x (thực chất là ta load lại file chứa biến x):

```
load("tên_file.rda")
```

- Xóa 1 biến khỏi bộ nhớ: `rm(x)`
- Xóa tất cả các biến: `rm(list = ls())`
- Liệt kê tất cả các biến hiện hành: `ls()`
- Xem thông tin của 1 biến nào đó: `str()`
- Xem thông tin của tất cả các biến đang làm việc: `ls.str()`

## 4.2 Nhập dữ liệu

### 4.2.1 Nhập dữ liệu trực tiếp

- Ta có số liệu về độ tuổi và huyết áp của 6 người đi khám tại 1 phòng mạch như sau

Age	Bloodpress
45	14
47	13
54	15
50	12
43	11
53	13

## 4.2 Nhập dữ liệu

### 4.2.1 Nhập dữ liệu trực tiếp

**1. Dùng c():** Ta có thể nhập dữ liệu bằng lệnh c() như sau

```
> age <- c(45,47,54,50,43,53)
> bloodpress <- c(14,13,15,12,11,13)
> benhnhan <- data.frame(age,bloodpress)
> benhnhan
```

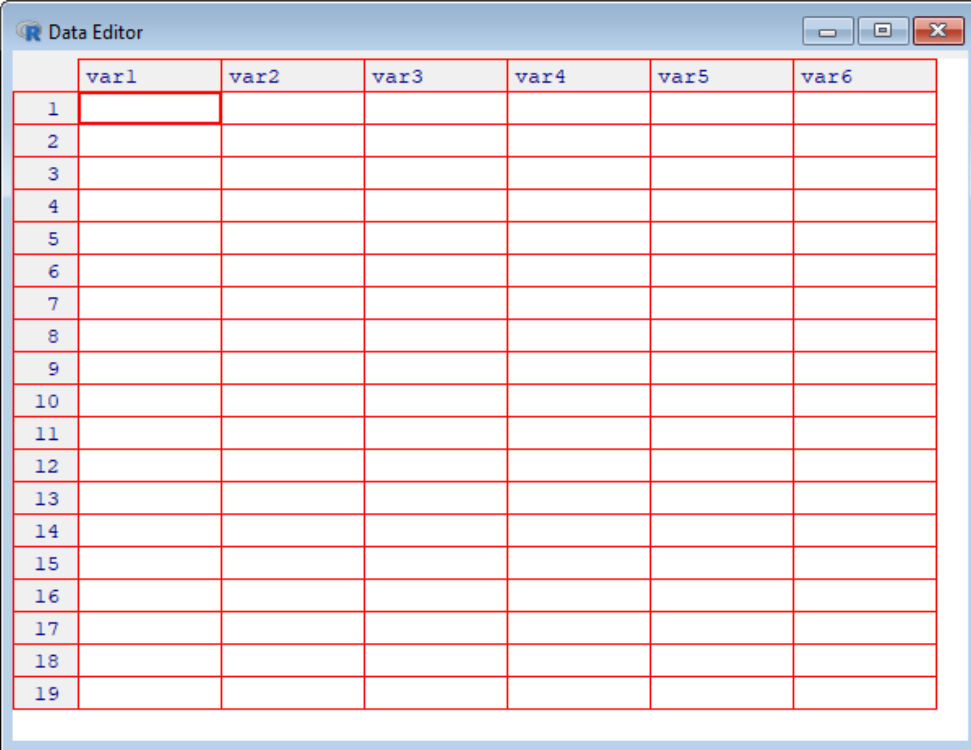
	age	bloodpress
1	45	14
2	47	13
3	54	15
4	50	12
5	43	11
6	53	13

## 4.2 Nhập dữ liệu

### 4.2.1 Nhập dữ liệu trực tiếp

#### 2. Dùng `edit(data.frame())`:

```
> benhnhan1 <- edit(data.frame())
```



The screenshot shows the R Data Editor window titled "Data Editor". It displays a new data frame with 6 columns labeled "var1", "var2", "var3", "var4", "var5", and "var6". There are 19 rows, numbered 1 through 19 on the left. The first cell (row 1, column var1) is highlighted with a red border.

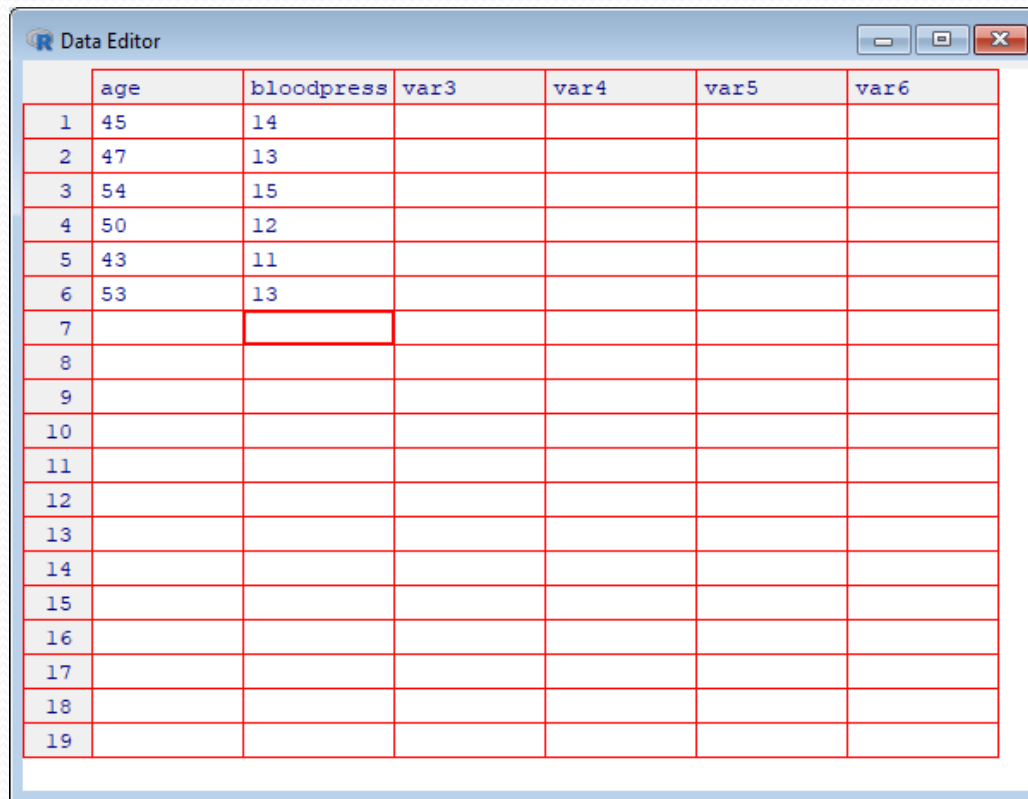
	var1	var2	var3	var4	var5	var6
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						

## 4.2 Nhập dữ liệu

### 4.2.1 Nhập dữ liệu trực tiếp

#### 2. Dùng `edit(data.frame())`:

```
> benhnhan1 <- edit(data.frame())
```



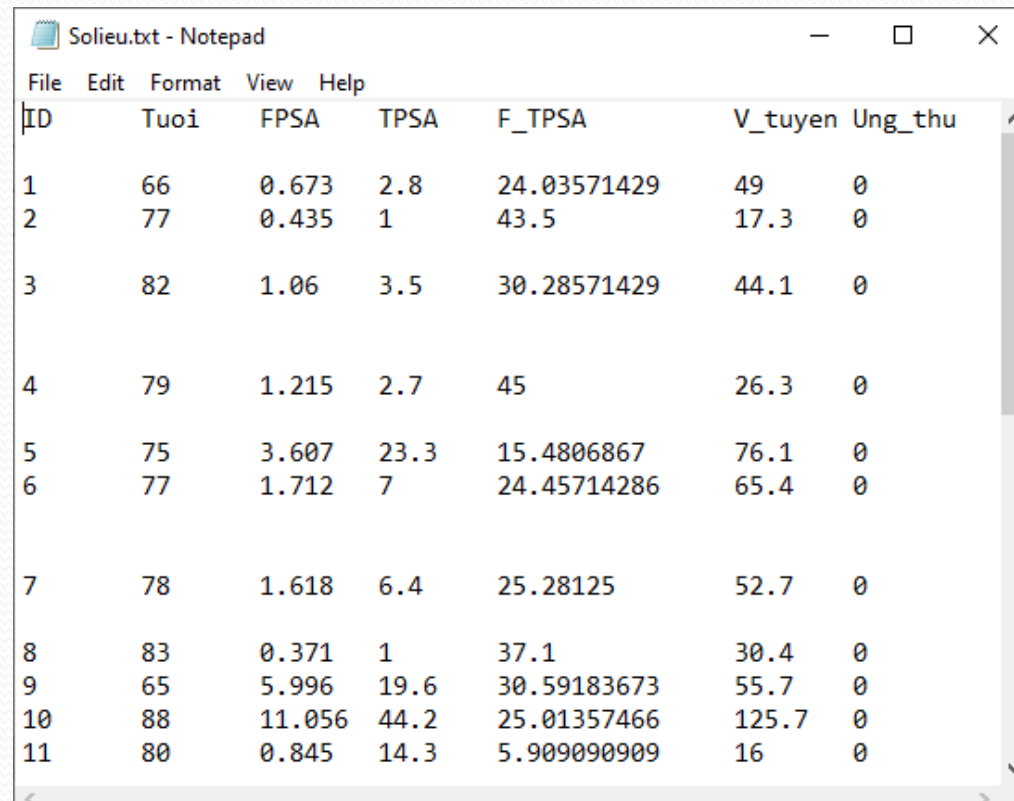
	age	bloodpress	var3	var4	var5	var6
1	45	14				
2	47	13				
3	54	15				
4	50	12				
5	43	11				
6	53	13				
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						
17						
18						
19						

## 4.2 Nhập dữ liệu

### 4.2.2 Nhập dữ liệu từ file

#### 1. Nhập từ file \*.txt: Dùng lệnh `read.table`

Giả sử ta muốn phân tích thống kê số liệu y khoa được lưu ở file Solieu.txt như sau



ID	Tuoi	FPSA	TPSA	F_TPSA	V_tuyen	Ung_thu
1	66	0.673	2.8	24.03571429	49	0
2	77	0.435	1	43.5	17.3	0
3	82	1.06	3.5	30.28571429	44.1	0
4	79	1.215	2.7	45	26.3	0
5	75	3.607	23.3	15.4806867	76.1	0
6	77	1.712	7	24.45714286	65.4	0
7	78	1.618	6.4	25.28125	52.7	0
8	83	0.371	1	37.1	30.4	0
9	65	5.996	19.6	30.59183673	55.7	0
10	88	11.056	44.2	25.01357466	125.7	0
11	80	0.845	14.3	5.909090909	16	0



## 4.2 Nhập dữ liệu

### 4.2.2 Nhập dữ liệu từ file

#### 1. Nhập từ file \*.txt: Dùng lệnh read.table

Ta sẽ thực hiện việc nhập dữ liệu từ file trên như sau

```
> setwd("E:/R_Works")  
> solieu <- read.table("Solieu.txt",header = TRUE)  
> solieu
```

	ID	Tuoi	FPSA	TPSA	F_TPSA	V_tuyen	Ung_thu
1	1	66	0.673	2.8	24.035714	49.0	0
2	2	77	0.435	1.0	43.500000	17.3	0
3	3	82	1.060	3.5	30.285714	44.1	0
4	4	79	1.215	2.7	45.000000	26.3	0
5	5	75	3.607	23.3	15.480687	76.1	0
6	6	77	1.712	7.0	24.457143	65.4	0

...

## 4.2 Nhập dữ liệu

### 4.2.2 Nhập dữ liệu từ file

#### 2. Nhập từ file \*.csv: Dùng lệnh read.csv

- R có thể nhận biết và đọc dữ liệu từ một file Excel với lưu ý là file dữ liệu trong Excel được lưu trữ dưới dạng \*.csv.
- Giả sử ta có một dữ liệu Excel là data01.xls, trước tiên ta sẽ mở file này rồi vào **File** → **Save As**, lưu với dạng \*.csv.

## 4.2 Nhập dữ liệu

### 4.2.2 Nhập dữ liệu từ file

#### 2. Nhập từ file \*.csv: Dùng lệnh read.csv

- Sau khi ta có file “data01.csv”, ta sẽ tiến hành đọc dữ liệu vào R như sau

```
> data <- read.csv("data01.csv", header = TRUE)
```

```
> data
```

	Age	FPSA	TPSA	K
1	66	0.673	2.80	0
2	77	0.435	1.00	0
3	82	1.060	3.50	0
4	79	1.215	2.70	0
5	75	3.607	23.30	0
6	77	1.712	7.00	0

...