

# THỐNG KÊ MÁY TÍNH

(Computational Statistics)

Trường Đại học Nha Trang

Khoa Công nghệ thông tin

Bộ môn Hệ thống thông tin

Giảng viên: TS.Nguyễn Khắc Cường

# CHƯƠNG 1

## GIỚI THIỆU NGÔN NGỮ R

# Giới thiệu ngôn ngữ R

- Các đối tượng cơ bản
  - Đại lượng vô hướng (scalar)
    - Các đối tượng đơn (Tạo lập bởi lệnh gán)

```
> x <-7  
> y<-x*2+3  
> x  
[1] 7  
> y  
[1] 17  
> mode(x)  
[1] "numeric"
```

- Lũy thừa cơ số e (dùng biểu diễn giá trị lớn)

```
> N<-3.5e23  
> N  
[1] 3.5e+23
```

# Giới thiệu ngôn ngữ R

- Các đối tượng cơ bản
  - Đại lượng vô hướng (scalar)
    - Biểu diễn số vô hạn

```
> x<-5/0  
> x  
[1] Inf
```

- Thông báo đại lượng không phải là số (Not a Number)

```
> x<-5/0  
> x  
[1] Inf  
> x-x  
[1] NaN
```

# Giới thiệu ngôn ngữ R

- Các đối tượng cơ bản

- Đại lượng vô hướng (scalar)

- Chuỗi

```
> x<-"Thống kê máy tính"
> x
[1] "Thống kê máy tính"
.
```

- Vector

- Mảng các phần tử đơn và cùng kiểu
- Dùng hàm c() *c:concatenation*

```
> x<-c(1,2,3,4,5)
> x
[1] 1 2 3 4 5
>
> y<-c(hs1="Lan", hs2="Hue")
> y
  hs1  hs2
"Lan" "Hue"
```

# Giới thiệu ngôn ngữ R

- Các đối tượng cơ bản
  - Vector
    - Dùng hàm `seq()`    *seq:sequence*

```
> x<-seq(2,8)
> x
[1] 2 3 4 5 6 7 8
> x<-seq(2,8,2)
> x
[1] 2 4 6 8
> x<-seq(2,8,3)
> x
[1] 2 5 8
```

# Giới thiệu ngôn ngữ R

- Các đối tượng cơ bản
  - Vector
    - Dùng hàm rep()

```
> x<-rep(2,5)
> x
[1] 2 2 2 2 2
>
> x<-rep(2:3,5)
> x
[1] 2 3 2 3 2 3 2 3 2 3
>
> x<-rep(c(2,4,6),3)
> x
[1] 2 4 6 2 4 6 2 4 6
> x<-rep(2:4,c(6,4,2))
> x
[1] 2 2 2 2 2 2 3 3 3 3 4 4
```

# Giới thiệu ngôn ngữ R

- Các đối tượng cơ bản
  - Matrix
    - Mảng 2 chiều
    - Cú pháp: `matrix(<dữ liệu>, nrow=<số hàng>, ncol=<số cột>, byrow=<FALSE|TRUE>, dimnames= <NULL|tên hàng và cột>)`

```
> matrix(1:12,3,4)
      [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
>
> matrix(1:12,3,4, byrow=T)
      [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12
```



# Giới thiệu ngôn ngữ R

- Các đối tượng cơ bản

- Matrix

```
> matrix(1:12,3,4, byrow=T, dimnames=list(c("A","B","C"), c("I","J","H","K")))  
  I  J  H  K  
A 1  2  3  4  
B 5  6  7  8  
C 9 10 11 12
```

- Qui định rõ dimension (size) của matrix

```
> x<-c(2,4,6,8,9,0,1,7)  
> dim(x)<-c(2,4)  
> x  
      [,1] [,2] [,3] [,4]  
[1,]    2    6    9    1  
[2,]    4    8    0    7
```

# Giới thiệu ngôn ngữ R

- Các đối tượng cơ bản
  - Matrix
    - Qui định rõ vị trí điền giá trị các phần tử, dùng hàm cbind(), rbind()

```
> cbind(A=1:4, B=5:8, D=9:12)
      A B  D
[1,] 1 5  9
[2,] 2 6 10
[3,] 3 7 11
[4,] 4 8 12
>
> rbind(A=1:4, B=5:8, D=9:12)
      [,1] [,2] [,3] [,4]
A         1     2     3     4
B         5     6     7     8
D         9    10    11    12
```

# Giới thiệu ngôn ngữ R

- Các đối tượng cơ bản

- Matrix

- Tạo ma trận đơn vị

```
> x<-matrix(0,4,4)
> diag(x)<-1
> x
```

	[,1]	[,2]	[,3]	[,4]
[1,]	1	0	0	0
[2,]	0	1	0	0
[3,]	0	0	1	0
[4,]	0	0	0	1

- Truy xuất các phần tử ma trận

```
> x<-matrix(1:15, 5, 3)
> x
```

	[,1]	[,2]	[,3]
[1,]	1	6	11
[2,]	2	7	12
[3,]	3	8	13
[4,]	4	9	14
[5,]	5	10	15

```
> x[2,2]
[1] 7
>
> x[,3]
[1] 11 12 13 14 15
>
> x[4,]
[1] 4 9 14
```

# Giới thiệu ngôn ngữ R

- Các đối tượng cơ bản
  - Matrix
    - Nhân 2 ma trận

```
> x<-matrix(1:6, 2:3)
> x
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
>
> y<-matrix(c(4,2,5,1,7,9,4,3,7,8,2,3), 3, 4)
> y
      [,1] [,2] [,3] [,4]
[1,]    4    1    4    8
[2,]    2    7    3    2
[3,]    5    9    7    3
>
> z<-x%*%y
> z
      [,1] [,2] [,3] [,4]
[1,]   35   67   48   29
[2,]   46   84   62   42
```

# Giới thiệu ngôn ngữ R

- Các đối tượng cơ bản
  - Matrix
    - Cộng, trừ 2 ma trận

```
> x<-matrix(1:6,2,3)
> x
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
>
> y<-matrix(7:12,2,3)
> y
      [,1] [,2] [,3]
[1,]    7    9   11
[2,]    8   10   12
>
> z<-x+y
> z
      [,1] [,2] [,3]
[1,]    8   12   16
[2,]   10   14   18
>
> z=y-x
> z
      [,1] [,2] [,3]
[1,]    6    6    6
[2,]    6    6    6
.
```

# Giới thiệu ngôn ngữ R

- Các đối tượng cơ bản
  - Matrix
    - Ma trận chuyển vị: `t(ma trận)`    *t:transpose*

```
> x<-matrix(c(4,2,5,1,7,9,4,3,7,8,2,3),3,4)
> x
      [,1] [,2] [,3] [,4]
[1,]    4    1    4    8
[2,]    2    7    3    2
[3,]    5    9    7    3
>
> y<-t(x)
> y
      [,1] [,2] [,3]
[1,]    4    2    5
[2,]    1    7    9
[3,]    4    3    7
[4,]    8    2    3
```

# Giới thiệu ngôn ngữ R

- Các đối tượng cơ bản

- Matrix

- Tính định thức:  $\det(\text{ma trận})$

*Khai triển Laplace*

Cho ma trận  $A = (a_{ij})_{n \times n}$  khi đó

$$\det(A) = a_{i1}A_{i1} + a_{i2}A_{i2} + \dots + a_{in}A_{in} \quad (i = 1, 2, \dots, n)$$

- Định thức ma trận cấp 3 (quy tắc Sarius):

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix}$$

```
> x<-matrix(c(2,3,5,1,7,8,3,9,2),3,3)
> x
      [,1] [,2] [,3]
[1,]    2    1    3
[2,]    3    7    9
[3,]    5    8    2
>
> y<-det(x)
> y
[1] -110
```

$$|A| = (a_{11}.a_{22}.a_{33} + a_{21}.a_{32}.a_{13} + a_{12}.a_{23}.a_{31}) - (a_{13}.a_{22}.a_{31} + a_{11}.a_{23}.a_{32} + a_{12}.a_{21}.a_{33})$$

# Giới thiệu ngôn ngữ R

- Các đối tượng cơ bản
  - Matrix
    - Ma trận nghịch đảo: `solve(ma trận)`

Cho  $A_{n \times n}$  có  $D = \det(A)$  và  $D_{ij}$  là định thức con của  $D$  bỏ đi hàng  $i$  cột  $j$

Ma trận  $A_{n \times n}$  khả đảo  $\Leftrightarrow \det(A) \neq 0$

$$A^{-1} = \frac{1}{\det(A)} \begin{bmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ A_{n1} & A_{n2} & \cdots & A_{nn} \end{bmatrix}^T \text{ với } A_{ij} = (-1)^{i+j} D_{ij}$$

```
> x<-matrix(c(2,3,5,1,7,8,3,9,2),3,3)
> x
      [,1] [,2] [,3]
[1,]    2    1    3
[2,]    3    7    9
[3,]    5    8    2
>
> y<-solve(x)
> y
      [,1] [,2] [,3]
[1,] 0.5272727 -0.2 0.10909091
[2,] -0.3545455 0.1 0.08181818
[3,] 0.1000000 0.1 -0.10000000
```



# Giới thiệu ngôn ngữ R

- Các đối tượng cơ bản

- Factor

- Lưu dữ liệu
    - Chỉ ra mức độ của từng dữ liệu
    - Cú pháp: `factor(x, levels=sort(unique(x),na.last=TRUE, labels=levels,exclude=NA, order=is.ordered(x))`

```
> x<-factor(1:3)
> x
[1] 1 2 3
Levels: 1 2 3
```

```
>
> x<-factor(1:3,levels=1:5)
> x
[1] 1 2 3
Levels: 1 2 3 4 5
>
> x<-factor(1:3,labels=c("A","B","C"))
> x
[1] A B C
Levels: A B C
```

```
> x<-factor(1:5,exclude=4)
> x
[1] 1    2    3    <NA> 5
Levels: 1 2 3 5
>
> x<-factor(c(2:4),levels=2:5)
> x
[1] 2 3 4
Levels: 2 3 4 5
```

# Giới thiệu ngôn ngữ R

- Các đối tượng cơ bản
  - List
    - Dãy các đối tượng
    - Cú pháp: `list([tên 1=]<đối tượng 1>, [tên 2=]<đối tượng 2> [...],[tên n=]<đối tượng n>))`

```
> x<-list(1:3, c("A","B"), 4:5, last=c(5))
> x
[[1]]
[1] 1 2 3

[[2]]
[1] "A" "B"

[[3]]
[1] 4 5

$last
[1] 5
```

# Giới thiệu ngôn ngữ R

- Các đối tượng cơ bản
  - Data frame
    - Bảng dữ liệu, có vai trò như một quan hệ trong CSDL quan hệ
    - Gồm: một list các vector hay factor
      - Có cùng kích thước
      - Có mối quan hệ với nhau
    - Mỗi hàng ứng với một quan sát (observation)
    - Mỗi cột ứng với một biến (variable) hay thuộc tính (attribute / field)
    - Cú pháp:
      - `data.frame (<vector 1>, <vector 2>, ..., <vector n>)`

# Giới thiệu ngôn ngữ R

- Các đối tượng cơ bản
  - Data frame

```
> x<-1:4
> x
[1] 1 2 3 4
>
```

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

```
> T<-data.frame(x,y)
```

```
Error in data.frame(x, y) :
  arguments imply differing number of rows: 4, 3
```

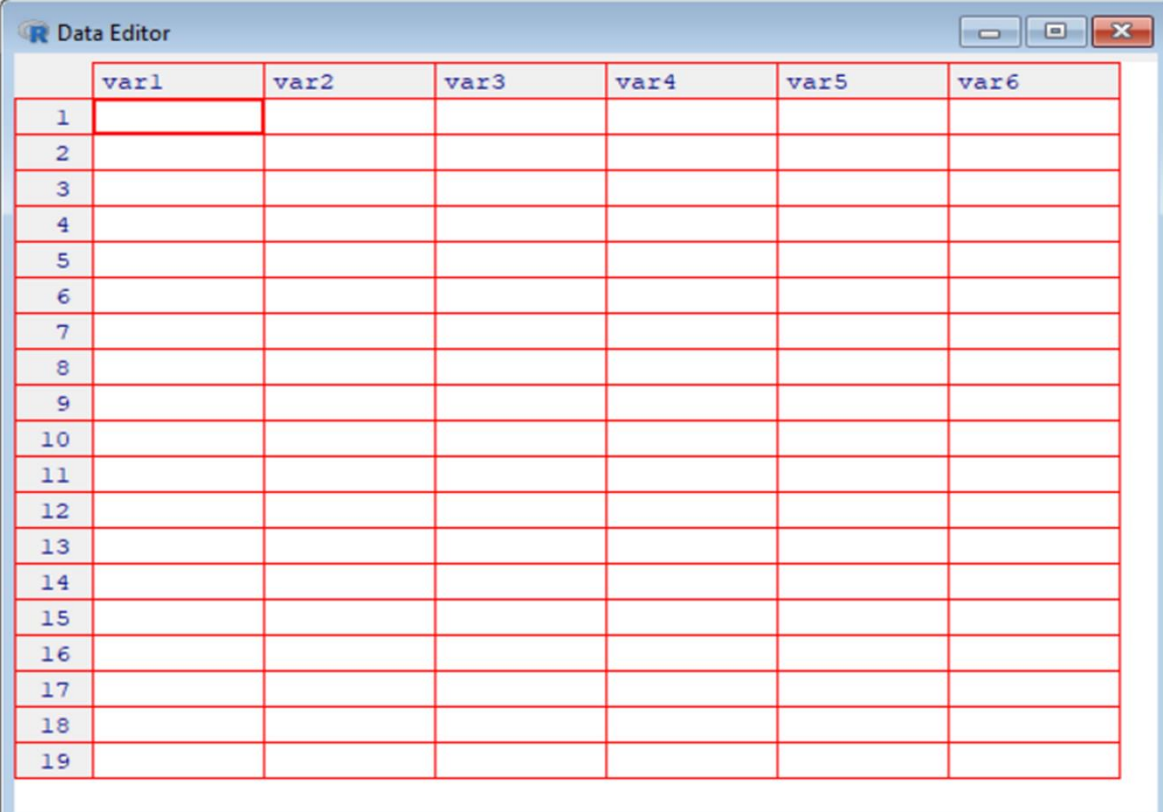
```
> x<-1:4
> x
[1] 1 2 3 4
>
> n<-10
> n
[1] 10
>
> T<-data.frame(x,n)
> T
   x  n
1 1 10
2 2 10
3 3 10
4 4 10
```

```
> x<-1:4
> x
[1] 1 2 3 4
>
> M<-c(10,35)
> M
[1] 10 35
>
> T<-data.frame(x,M)
> T
   x  M
1 1 10
2 2 35
3 3 10
4 4 35
```

# Giới thiệu ngôn ngữ R

- Các đối tượng cơ bản
  - Data frame
    - Tự tạo data frame bằng tay

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



The screenshot shows the R Data Editor window. The title bar reads "R Data Editor". The window contains a table with 6 columns labeled "var1", "var2", "var3", "var4", "var5", and "var6". The rows are numbered 1 through 19 on the left side. The table is currently empty, with all cells highlighted in light pink.

	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						

# Giới thiệu ngôn ngữ R

- Các đối tượng cơ bản

- Data frame

- Tách / Chọn dữ liệu từ data frame = phép chọn (SELECT) trong CSDL QH
    - Cú pháp: subset(<dữ liệu>, <điều kiện>)

```
> x<-1:4
> x
[1] 1 2 3 4
>
> M<-c(12,35,10,18)
> M
[1] 12 35 10 18
>
> T<-data.frame(x,M)
> T
  x  M
1 1 12
2 2 35
3 3 10
4 4 18

> T_sub<-subset(T, M>12)
> T_sub
  x  M
2 2 35
4 4 18
```

# Giới thiệu ngôn ngữ R

- Các đối tượng cơ bản

- Data frame

- Kết nối dữ liệu = phép kết nối tự nhiên trong CSDL QH
    - Cú pháp: `merge(<dữ liệu1>,<dữ liệu2>,by="thuộc tính kết nối", all=TRUE))`

```
> x<-1:4
> x
[1] 1 2 3 4
>
> M<-c(12,35,10,18)
> M
[1] 12 35 10 18
>
> T1<-data.frame(x,M)
> T1
  x  M
1 1 12
2 2 35
3 3 10
4 4 18

> x<-1:4
> x
[1] 1 2 3 4
>
> N<-c(45,23,67,89)
> N
[1] 45 23 67 89
>
> K<-c("Nam", "Nu", "Nu", "Nam")
> K
[1] "Nam" "Nu"  "Nu"  "Nam"
>
> T2<-data.frame(x,N,K)
> T2
  x  N  K
1 1 45 Nam
2 2 23 Nu
3 3 67 Nu
4 4 89 Nam

> KET_NOI<-merge(T1,T2,by="x",all=TRUE)
> KET_NOI
  x  M  N  K
1 1 12 45 Nam
2 2 35 23 Nu
3 3 10 67 Nu
4 4 18 89 Nam
```

# Giới thiệu ngôn ngữ R

- Các phép toán cơ bản

Phép toán					
Số học		So sánh		Logic	
+	<i>cộng</i>	<	<i>nhỏ hơn</i>	!x	<i>NOT(x)</i>
-	<i>trừ</i>	>	<i>lớn hơn</i>	x&y	<i>x AND y</i>
*	<i>nhân</i>	<=	<i>nhỏ hơn hoặc bằng</i>	x&&y	<i>x AND y</i>
/	<i>chia</i>	>=	<i>lớn hơn hoặc bằng</i>	x y	<i>x OR y</i>
^	<i>lũy thừa</i>	==	<i>bằng</i>	x  y	<i>x OR y</i>
%%	<i>modulo</i>	!=	<i>khác</i>	xor(x,y)	<i>x XOR y</i>
%/%	<i>chia lấy phần nguyên</i>				



# Giới thiệu ngôn ngữ R

- Các hàm số học cơ bản

Căn bậc hai: $\sqrt{x}$	<code>sqrt(x)</code>	Hàm <code>cos(x)</code> , <code>sin(x)</code> , <code>tg(x)</code> , <code>arcsin(x)</code> , <code>arccos(x)</code> , <code>arctg(x)</code>	<code>cos(x)</code> , <code>sin(x)</code> , <code>tan(x)</code> , <code>asin(x)</code> , <code>acos(x)</code> , <code>atan(x)</code>
Logarit nepe: <code>ln(x)</code>	<code>log(x)</code>	Hàm tổng $\sum_{i=1}^3 x_i$	<code>X&lt;-c(x1,x2,x3)</code> <code>sum(X)</code>
Logarit cơ số 10: <code>log10(x)</code>	<code>log10(x)</code>	Hàm $e^x$	<code>exp(x)</code>
Logarit cơ số 2: <code>log2(x)</code>	<code>log2(x)</code>	Hàm $ x $	<code>abs(x)</code>

# Giới thiệu ngôn ngữ R

- Chuyển đổi kiểu dữ liệu
  - Cú pháp: `as.<tên kiểu>(<đối tượng>)`
  - Quy tắc:

Chuyển đổi	Hàm	Quy tắc
numeric	as.numeric	FALSE → 0
		TRUE → 1
		"1","2",... → 1,2,...
		"A",... → NA
logical	as.logical	0 → FALSE
		<i>Các số khác</i> → TRUE
		"FALSE","F" → FALSE
		"TRUE","T" → TRUE
		<i>Các ký tự khác</i> → NA
character	as.character	1,2... → "1","2",...
		FALSE → "FALSE"
		TRUE → "TRUE"

# Giới thiệu ngôn ngữ R

- Chuyển đổi kiểu dữ liệu

```
> x<-factor(c(1,10))
> x
[1] 1 10
Levels: 1 10
>
> as.numeric(x)
[1] 1 2
>
>
> x<-factor(c("Male","Female"))
> x
[1] Male   Female
Levels: Female Male
>
> as.numeric(x)
[1] 2 1
```

# Giới thiệu ngôn ngữ R

- Các câu lệnh lập trình cơ bản
  - If..else
    - if <biểu thức logic> <câu lệnh 1> [else <câu lệnh 2>]

```
> x<-c("what","is","truth")
> x
[1] "what"  "is"    "truth"
>
> if ("Hello" %in% x) {
+   print("Found")
+ } else {
+   print("Not found")
+ }
[1] "Not found"
```

# Giới thiệu ngôn ngữ R

- Các câu lệnh lập trình cơ bản

- Switch

- `switch(<biểu thức>, <tr.hợp 1>, <tr.hợp 2>,...)`

- Giá trị của <biểu thức>

- không phải là 1 chuỗi ký tự thì tự động được gán cho <tr.hợp n>

- là 1 chuỗi ký tự thì trường hợp tương ứng được thực hiện

- là số thì trường hợp tương ứng được thực hiện

```
> switch("a", A=1, B=5, a=6)
[1] 6
```

```
> switch(2, A=1, B=5, a=6)
[1] 5
```

# Giới thiệu ngôn ngữ R

- Các câu lệnh lập trình cơ bản

- Repeat

- repeat { <câu lệnh>  
                  if <biểu thức logic> {  
                      break  
                  }  
                  }  
                  }

```
> x<-"Hello"  
> times<-2  
> repeat {  
+   print(x)  
+   times<-times+1  
+   if (times>4) {  
+       break  
+   }  
+ }  
[1] "Hello"  
[1] "Hello"  
[1] "Hello"
```

# Giới thiệu ngôn ngữ R

- Các câu lệnh lập trình cơ bản

- While

- while (<biểu thức logic>) {  
    <câu lệnh>  
}

```
> x<-"Hello"  
> times<-2  
> while (times<5) {  
+   print(x);  
+   times<-times+1;  
+ }  
[1] "Hello"  
[1] "Hello"  
[1] "Hello"
```

# Giới thiệu ngôn ngữ R

- Các câu lệnh lập trình cơ bản

- For

- for (<biến đếm> in <gtrị đầu:gtrị cuối>) { ds câu lệnh }

```
> x<-LETTERS[1:4]
> x
[1] "A" "B" "C" "D"
>
> for (i in x) {
+   print(i)
+ }
[1] "A"
[1] "B"
[1] "C"
[1] "D"
```

```
> x<-2:5
> x
[1] 2 3 4 5
> for (i in x) {
+   print(i)
+ }
[1] 2
[1] 3
[1] 4
[1] 5
```

```
> x<-0
> for (i in 1:10) {
+   x<-x+i
+ }
> print(x)
[1] 55
```



# Giới thiệu ngôn ngữ R

- Các câu lệnh lập trình cơ bản
  - Hàm tự xây dựng
    - <tên hàm> <- function (<ds tham số>){ds câu lệnh}

```
> tong<-function(a,b) {  
+   s<-a+b  
+   print(s)  
+ }  
>  
> tong(2,5)  
[1] 7
```

# Giới thiệu ngôn ngữ R

- Các câu lệnh xuất / nhập dữ liệu
  - Xác định thư mục làm việc
    - Hàm lấy tên thư mục hiện hành: `getwd()`
    - Hàm thiết lập thư mục hiện hành: `setwd(<thư mục>)`

```
> getwd()  
[1] "C:/Users/ADMIN/Documents"
```

```
> setwd("d:/")  
> getwd()  
[1] "d/"
```

# Giới thiệu ngôn ngữ R

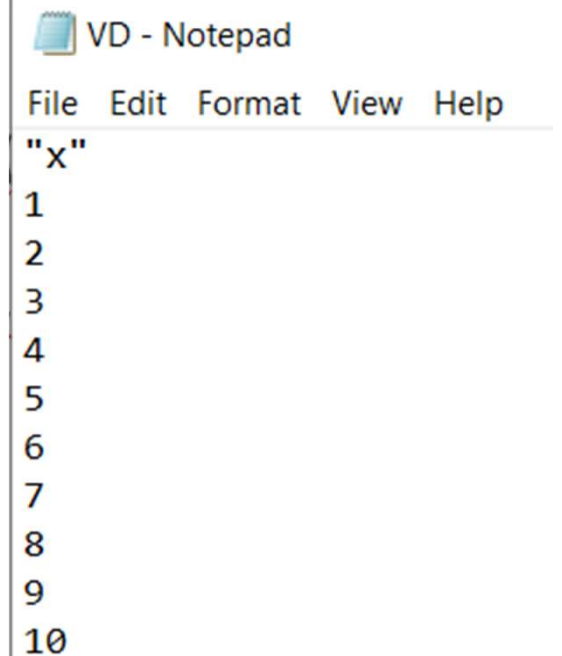
- Các câu lệnh xuất / nhập dữ liệu
  - Hàm lưu dữ liệu vào file \*.Rdata : `save()`
  - Hàm đọc dữ liệu từ file \*.Rdata : `load()`
  - Hàm xóa đối tượng dữ liệu từ R : `rm()`

```
> x<-1:10      #Tạo đối tượng x trong bộ nhớ
> x            #In ra đối tượng x
[1] 1 2 3 4 5 6 7 8 9 10
> save(x, file="VD.Rdata")    #Lưu đối tượng x vào file
> rm(x)          #Xóa đối tượng x trong bộ nhớ
> x              #In ra đối tượng x
Error: object 'x' not found
> load("VD.Rdata")           #Load lại đối tượng x từ file data
> print(x)      #In ra đối tượng x
[1] 1 2 3 4 5 6 7 8 9 10
>
>
> x
[1] 1 2 3 4 5 6 7 8 9 10
```

# Giới thiệu ngôn ngữ R

- Các câu lệnh xuất / nhập dữ liệu
  - Hàm lưu dữ liệu vào file text: `write.table(<"tên file">, row.names=F)`

```
> x<-1:10
> x
[1] 1 2 3 4 5 6 7 8 9 10
>
> write.table(x, file="VD.txt", row.names=F)
```



VD - Notepad

File Edit Format View Help

"x"

1

2

3

4

5

6

7

8

9

10

# Giới thiệu ngôn ngữ R

- Các câu lệnh xuất / nhập dữ liệu
  - Hàm đọc dữ liệu từ file text: `read.table(<"tên file">, header=TRUE)`

```
> a<-read.table("VD.txt", header=T)
```

```
> a
```

	x
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10

```
> a<-read.table("VD.txt", header=F)
```

```
> a
```

	V1
1	x
2	1
3	2
4	3
5	4
6	5
7	6
8	7
9	8
10	9
11	10

# Giới thiệu ngôn ngữ R

- Các câu lệnh xuất / nhập dữ liệu
  - Hàm ghi dữ liệu vào file \*.CSV: `write.csv(<"tên file", row.names=F)`
  - Hàm đọc dữ liệu từ file \*.CSV: `read.csv(<"tên file", header=TRUE)`

```
> x<-1:5
> x
[1] 1 2 3 4 5
>
> y<-(1:5)/10
> y
[1] 0.1 0.2 0.3 0.4 0.5
>
> z<-c("R", "and", "Data Mining", "Example", "Case Studies")
> z
[1] "R"          "and"         "Data Mining" "Example"     "Case Studies"

> dfl<-data.frame(x,y,z)
> dfl
  x  y      z
1 1 0.1     R
2 2 0.2    and
3 3 0.3 Data Mining
4 4 0.4   Example
5 5 0.5 Case Studies
```

# Giới thiệu ngôn ngữ R

- Các câu lệnh xuất / nhập dữ liệu
  - Hàm ghi dữ liệu vào file \*.CSV: `write.csv(<"tên file", row.names=F)`
  - Hàm đọc dữ liệu từ file \*.CSV: `read.csv(<"tên file", header=TRUE)`

```
> names(dfl) <- c("Bien_nguyen", "Bien_thuc", "Bien_ky_tu")
> names(dfl)
[1] "Bien_nguyen" "Bien_thuc"    "Bien_ky_tu"
```

```
> write.csv(dfl, "VD.csv", row.names=F)
```

```
> df2 <- read.csv("VD.csv")
> df2
  Bien_nguyen Bien_thuc Bien_ky_tu
1           1      0.1          R
2           2      0.2         and
3           3      0.3  Data Mining
4           4      0.4    Example
5           5      0.5 Case Studies
```

	A	B	C	D
1	Bien_nguyen	Bien_thuc	Bien_ky_tu	
2	1	0.1	R	
3	2	0.2	and	
4	3	0.3	Data Mining	
5	4	0.4	Example	
6	5	0.5	Case Studies	

Q / A