Phân tích đánh giá hiệu năng hệ thống

Lập trình R cơ bản

Nguyễn Hải Châu

Trường Đại học Công nghệ
Đại học Quốc gia Hà Nội

Giới thiệu R

The R Project for Statistical Computing

- R là một phần mềm miễn phí chuyên về tính toán thống kê và đồ thị
- R có thể chạy trên Linux, MacOS, Windows và nhiều hệ điều hành khác
- Tải miễn phí R tại: `https://www.r-project.org/`

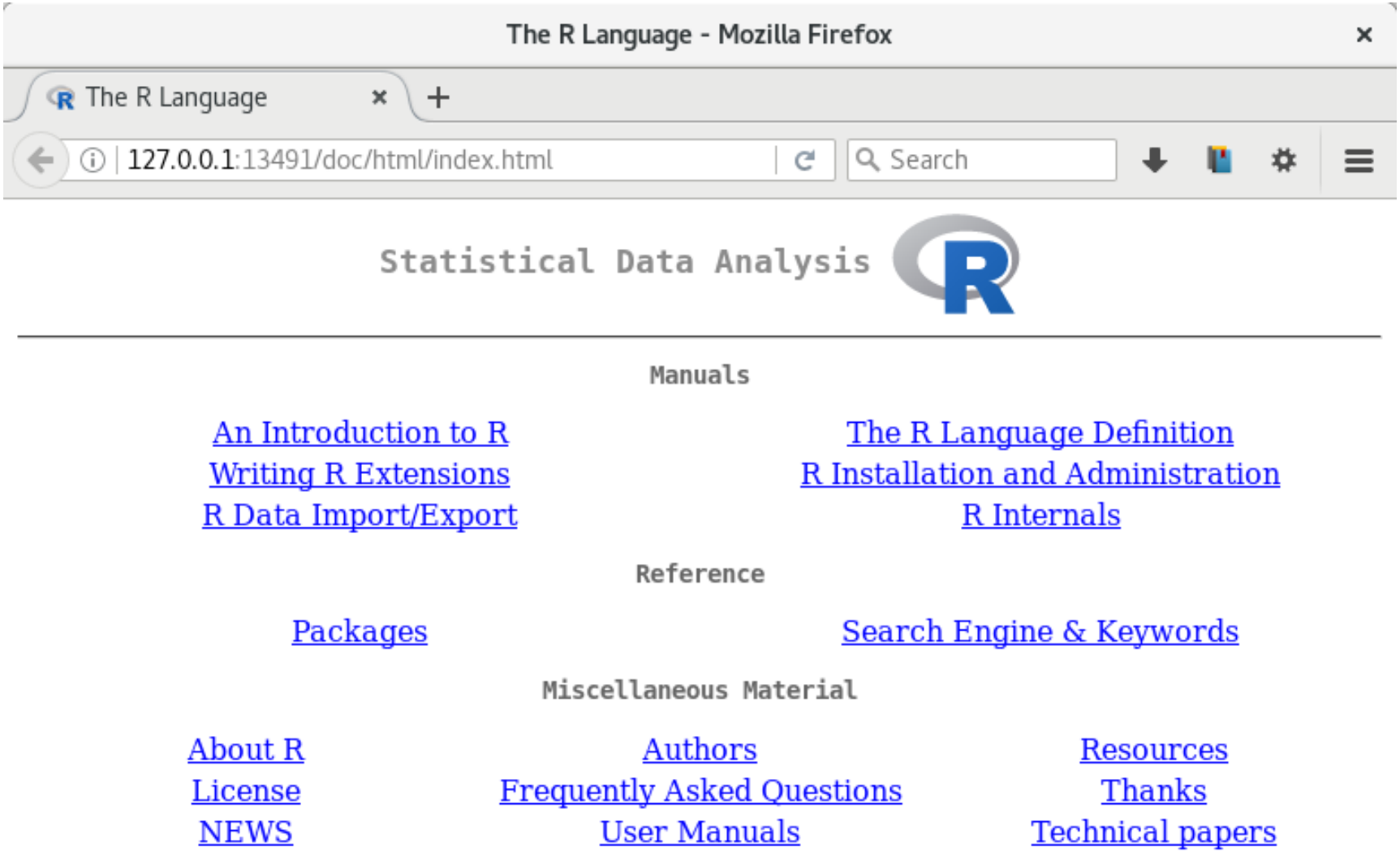
R cơ bản

Khởi động R và xem hướng dẫn

- Trên Linux: gõ lệnh R, Windows: nháy đúp vào biểu tượng của R
- Sau khi dấu nhắc "`>`" của R hiện ra, gõ lệnh `help.start()`

```
[chau@dino arduino]$ R

R version 3.4.1 (2017-06-30) -- "Single Candle"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-redhat-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> help.start()
starting httpd help server ... done
If the browser launched by '/usr/bin/xdg-open' is already running, it is *not* restarted, and you
itch to its window.
Otherwise, be patient ...
>
```

Màn hình hướng dẫn



Các phép toán

Lưu ý các dòng có hai dấu '##' ở đầu dòng là kết quả thực hiện lệnh

```
5-1+10          # Cộng trừ

## [1] 14

7*10/2          # Nhân chia

## [1] 35

5.2^2.2         # Lũy thừa, hoặc 5.2**2.2

## [1] 37.60169

pi              # Số Pi

## [1] 3.141593

exp(1)          # Số e

## [1] 2.718282
```

Phép gán

```
x <- 5
x
```

```
## [1] 5
```

```
6 -> y
y
```

```
## [1] 6
```

```
z <- pi
z
```

```
## [1] 3.141593
```

Gọi hàm

```
print(x)          # In giá trị của x, print() là hàm có sẵn

## [1] 5

ls()              # Xem các đối tượng trong bộ nhớ

##  [1] "a"         "addition" "ar"       "b"         "ch"        "dd"
##  [7] "df"        "ft"       "ft1"      "ft2"       "i"         "int"
## [13] "j"         "l"        "m"        "myiris"    "mylist"    "s"
## [19] "simple"    "sp"       "str"      "tt"        "v"         "x"
## [25] "y"         "z"

rm(x)             # Xóa x, giải phóng bộ nhớ
ls()

##  [1] "a"         "addition" "ar"       "b"         "ch"        "dd"
##  [7] "df"        "ft"       "ft1"      "ft2"       "i"         "int"
## [13] "j"         "l"        "m"        "myiris"    "mylist"    "s"
## [19] "simple"    "sp"       "str"      "tt"        "v"         "y"
## [25] "z"
```

Mode

Các đơn vị dữ liệu cơ bản của R được lưu trữ với cùng dung lượng bộ nhớ. Các kiểu đơn vị lưu trữ này được gọi là *mode*. R có các mode:

- LOGICAL: có hai giá trị `TRUE`, `FALSE`, có thể viết tắt `T`, `F`
- `character`
- `numeric`
- `complex`
- `raw`

Mode

```
mode(x <- integer())

## [1] "numeric"

mode(x <- double())

## [1] "numeric"

mode(x <- TRUE)

## [1] "logical"

mode(x <- 'a')

## [1] "character"

mode(x <- factor('a'))

## [1] "numeric"
```

Vector: Chỉ số, độ dài, tổng

```
x <- 0:9
x
```

```
##  [1] 0 1 2 3 4 5 6 7 8 9
```

```
x[5]              # Chỉ số vector bắt đầu từ 1
```

```
## [1] 4
```

```
x <- c(1,3,2,6,7,-1,-8)
x[3]
```

```
## [1] 2
```

```
length(x)         # Độ dài vector
```

```
## [1] 7
```

```
sum(x)            # Tính tổng
```

```
## [1] 10
```

Vector: các phép toán số học

```
x <- 1.2: 6.4
x

## [1] 1.2 2.2 3.2 4.2 5.2 6.2

x*2

## [1]  2.4  4.4  6.4  8.4 10.4 12.4

x/2

## [1] 0.6 1.1 1.6 2.1 2.6 3.1

x-1

## [1] 0.2 1.2 2.2 3.2 4.2 5.2
```

Vector: tập con

```
x <- 15:30
(c(x[5], x[9]))

## [1] 19 23

(x[c(5,9)])

## [1] 19 23

(x[-c(1, 6:11, 15)])

## [1] 16 17 18 19 26 27 28 30
```

Vector: giá trị null

```
(x <- c(10, 20, NA, 4, NA, 2))

## [1] 10 20 NA  4 NA  2

mean(x)

## [1] NA

(i <- is.na(x))

## [1] FALSE FALSE  TRUE FALSE  TRUE FALSE

(j <- !i)

## [1]  TRUE  TRUE FALSE  TRUE FALSE  TRUE

y <- x[!i]
mean(y)

## [1] 9

mean(x, na.rm=TRUE)

## [1] 9
```

Chia hai vector

```
(x <- seq(2, 10, by=2)) # Gán, đồng thời in giá trị của x
```

```
## [1]  2  4  6  8 10
```

```
(y <- 1:5)
```

```
## [1] 1 2 3 4 5
```

```
x/y
```

```
## [1] 2 2 2 2 2
```

Chia hai vector

```
(x <- seq(2, 10, by=1.5)) # Gán, đồng thời in giá trị của x
```

```
## [1] 2.0 3.5 5.0 6.5 8.0 9.5
```

```
(y <- 1:5)
```

```
## [1] 1 2 3 4 5
```

```
x/y
```

```
## Warning in x/y: longer object length is not a multiple of shorter object length
```

```
## [1] 2.000000 1.750000 1.666667 1.625000 1.600000 9.500000
```

```
(z <- 1:2)
```

```
## [1] 1 2
```

```
x/z
```

```
## [1] 2.00 1.75 5.00 3.25 8.00 4.75
```

Vector ký tự

```
(s <- c('Hà Nội', "Huế", 'Sài Gòn'))
```

```
## [1] "Hà Nội"  "Huế"     "Sài Gòn"
```

```
paste(s[1], s[2], s[3])
```

```
## [1] "Hà Nội Huế Sài Gòn"
```

```
paste(s[1], s[2], s[3], sep="-")
```

```
## [1] "Hà Nội-Huế-Sài Gòn"
```

```
paste0(s[1], s[2], s[3])
```

```
## [1] "Hà NộiHuếSài Gòn"
```

```
paste0(s[1], s[2], s[3], sep='-')
```

```
## [1] "Hà NộiHuếSài Gòn-"
```

Chỉ số ký tự của vector

```r
x <- c(160, NA, 175, NA, 180)
y <- c(50, 65, 65, 80, 70)
names(x) <- c("Hà Nội", "Hải Phòng", "Nam Định", "Đà Nẵng", "Nha Trang")
cbind(x, y) # Đặt hai vector cạnh nhau theo dạng cột
```

```
##             x  y
## Hà Nội    160 50
## Hải Phòng  NA 65
## Nam Định  175 65
## Đà Nẵng    NA 80
## Nha Trang 180 70
```

```r
names(x)
```

```
## [1] "Hà Nội"    "Hải Phòng" "Nam Định"  "Đà Nẵng"   "Nha Trang"
```

```r
names(y)
```

```
## NULL
```

```r
x['Hà Nội', 'Nha Trang']  # Ok: x được đặt tên
```

```
##    Hà Nội Nha Trang
##       160       180
```

```r
y['Hà Nội', 'Nha Trang']  # Fail: y không được đặt tên
```

```
## [1] NA NA
```

Toán tử logic

```
5 > 4; 5 < 4; 5 == 4; 5 != 4
```

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

```
!(5 == 5)
```

```
## [1] FALSE
```

```
!(5 == 4)
```

```
## [1] TRUE
```

```
5 == 4 & 5 == 5
```

```
## [1] FALSE
```

```
5 == 4 | 5 == 5
```

```
## [1] TRUE
```

```
xor(TRUE, FALSE);xor(TRUE, TRUE)
```

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

Một số giá trị đặc biệt

- `NA`: Giá trị null (không xác định)
- `NaN`: Not a number
- `Inf`: Infinity (vô cùng)

```
(x <- c(NA, 1 / 0, 0/0, -1/0, -Inf, Inf - Inf, Inf, 5))

## [1]   NA  Inf  NaN -Inf -Inf  NaN  Inf    5

is.na(x)

## [1]  TRUE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE

is.nan(x)

## [1] FALSE FALSE  TRUE FALSE FALSE  TRUE FALSE FALSE

x == Inf

## [1]    NA  TRUE    NA FALSE FALSE    NA  TRUE FALSE
```

Đối tượng

```r
x <- 2                    # Vector độ dài 1
x <- vector()             # Vector độ dài 0
x <- matrix()             # Ma trận 1 hàng 1 cột
x <- 'ĐHCN'               # Vector ký tự độ dài 1
x <- c('ĐHCN', 'ĐHQG')          # Vector ký tự độ dài 2
x <- function() {}        # Hàm không có tham số
```

Cấu trúc điều khiển

Rẽ nhánh

```
if (test) { # Test là biểu thức logic
   Thực hiện thao tác A
} else { # if có thể lồng nhau
   Thực hiện thao tác B
}
```

Vòng lặp `for`

```
x <- seq(10, 50, by=10)
x

## [1] 10 20 30 40 50

l <- length(x)
for (i in 1:l)
   if (x[i] < 30) {
      x[i] <- x[i] * 2
   } else {
      x[i] <- x[i] + 1
   }
x

## [1] 20 40 31 41 51
```

Vòng lặp vector hóa `sapply`

Tốc độ thực hiện của `sapply` nhanh hơn của `for`

```
x <- seq(10, 50, by=10)
x

## [1] 10 20 30 40 50

x <- sapply(x, function(j) {
   if (j < 30) {
      j*2
   } else {
      j+1
   }
})
x

## [1] 20 40 31 41 51
```

Hàm

Khai báo và sử dụng hàm:

```
# Khai báo
function.name <- function(arguments) {
   body and return values
}
# Sử dụng
function.name(arguments)
```

Có thể khai báo hàm trên một hoặc nhiều dòng:

```
simple <- function() {1}
simple()

## [1] 1

simple <- function() {
   1
}
simple()

## [1] 1
```

Hàm

Khai báo và sử dụng hàm:

```
addition <- function(x) {x+1}
addition(5)

## [1] 6

addition()

## Error in addition(): argument "x" is missing, with no default

addition <- function(x=0) {x+1} # Tham số vào ngầm định là 0
addition(5)

## [1] 6

addition()

## [1] 1
```

Sử dụng các thư viện

Nạp/gỡ một thư viện

- R có nhiều thư viện core và cộng đồng đóng góp
- Thư viện được cộng đồng đóng góp được liệt kê ở `https://cran.r-project.org/web/packages/`

```
library(epiR)    # Nạp thư viện

## Package epiR 0.9-87 is loaded
## Type help(epi.about) for summary information
##

detach('package:epiR', unload=TRUE)      # Gỡ thư viện đã nạp
```

Cài đặt/xóa thư viện

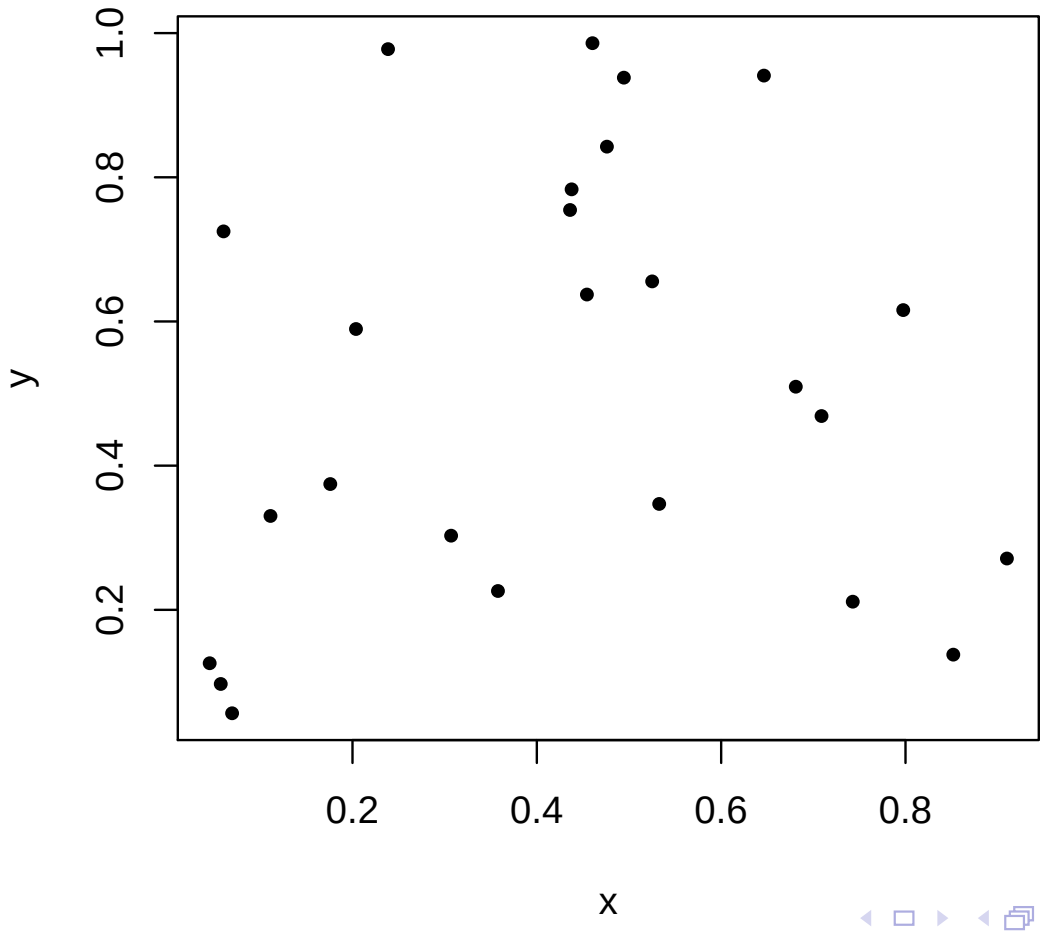
- Trên Linux: Chỉ có `root` được cài đặt/xóa thư viện trên R của hệ thống. Người sử dụng thông thường chỉ được cài đặt/xóa thư viện trên bản R của cá nhân
- Khi cài đặt thư viện, cần chỉ ra địa chỉ lưu trữ, nếu không R sẽ tự động đưa ra menu chọn

```
install.packages('ggplot2')      # R sẽ đưa ra menu chọn địa chỉ lưu trữ
remove.packages('ggplot2')       # Xóa thư viện vừa cài đặt
# Chỉ ra địa chỉ lưu trữ
install.packages('ggplot2', repo='http://cran.asm.ac.jp')
```

Đồ thị

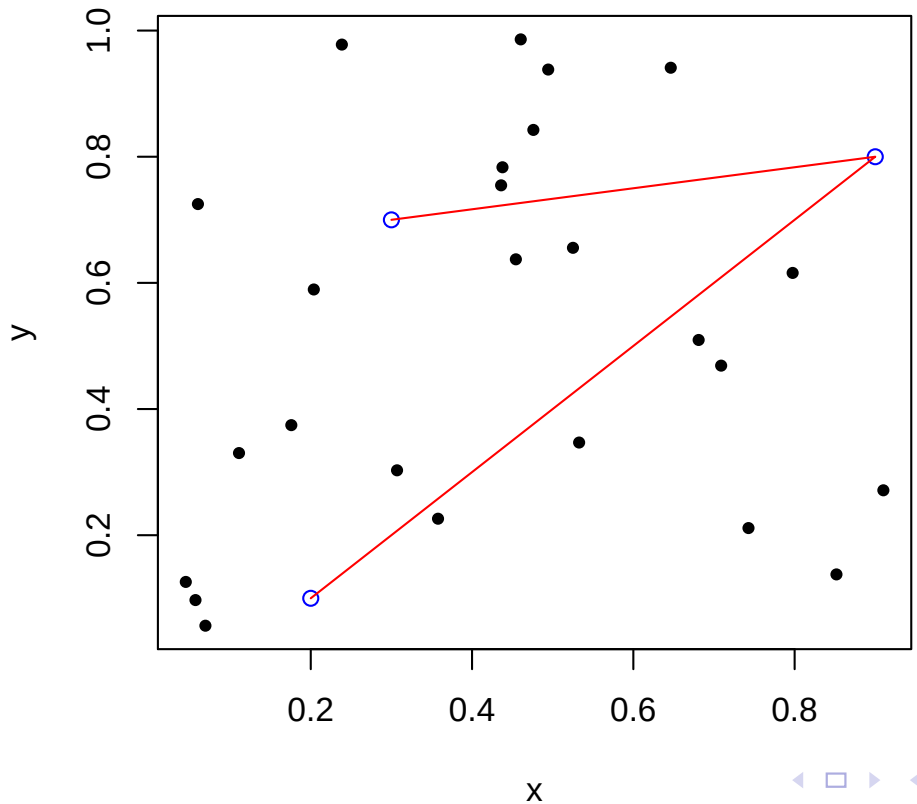
Các hàm đồ thị bậc cao

```
x <- runif(25) ; y <- runif(25) ; plot(x, y, pch=20) # Scatter plot
```



Các hàm đồ thị bậc thấp

```r
plot(x, y, pch=20) # Scatter plot
a <- c(0.2, 0.9, 0.3); b <- c(0.1, 0.8, 0.7)
points(a, b, col='blue')
lines(a, b, col='red')
```



Ghi đồ thị ra file raster

```
png('myplot.png', width=640, height=640) # Sử dụng hàm jpeg() cho định dạng JPEG
plot(x, y, pch=20) # Scatter plot
a <- c(0.2, 0.9, 0.3); b <- c(0.1, 0.8, 0.7)
points(a, b, col='blue')
lines(a, b, col='red')
dev.off()

## pdf
##   2

system('ls -l myplot.png', intern=TRUE) # Lệnh shell trên Linux

## [1] "-rw-rw-r-- 1 chau chau 13749 Sep  4 06:56 myplot.png"
```

Ghi đồ thị ra file vector

```
setEPS()
postscript('myplot.eps')
plot(x, y, pch=20) # Scatter plot
a <- c(0.2, 0.9, 0.3); b <- c(0.1, 0.8, 0.7)
points(a, b, col='blue')
lines(a, b, col='red')
dev.off()

## pdf
##   2

system('ls -l myplot.eps', intern=TRUE) # Lệnh shell trên Linux

## [1] "-rw-rw-r-- 1 chau chau 4895 Sep  4 06:56 myplot.eps"
```

Các kiểu dữ liệu trong R

Kiểu dữ liệu

- `factor`: Dữ liệu phân loại, có thể có thứ tự hoặc không (liên hệ kiểu `enum` trong C)
- `numeric, integer`: Kiểu số thực, nguyên
- `character`: Kiểu ký tự
- `Date`: Kiểu ngày tháng
- Để xem kiểu của một đối tượng, dùng hàm `class`
- Để chuyển đổi kiểu, dùng hàm
  tt as.x, trong đó `x` trùng với các kiểu tương ứng. Ví dụ:
  `as.integer, as.factor`

Kiểu dữ liệu factor

```
ft <- c('Thấp', 'Trung bình', 'Cao')
class(ft)       # Xem kiểu của ft

## [1] "character"

ft1 <- factor(ft)
ft2 <- factor(ft, order=TRUE)
ft1

## [1] Thấp       Trung bình Cao
## Levels: Cao Thấp Trung bình

ft2

## [1] Thấp       Trung bình Cao
## Levels: Cao < Thấp < Trung bình

ft2 <- factor(ft2, levels(ft2)[c(2,3,1)])
ft2

## [1] Thấp       Trung bình Cao
## Levels: Thấp < Trung bình < Cao
```

Kiểu dữ liệu `factor`

```
class(ft1)

## [1] "factor"

class(ft2)

## [1] "ordered" "factor"

is.ordered(ft1)

## [1] FALSE

is.ordered(ft2)

## [1] TRUE
```

Kiểu dữ liệu numeric, integer

```
z <- c(1.1, 2.2, 3.4)
i <- c(1, 2, 3)
class(z)

## [1] "numeric"

class(y)

## [1] "numeric"

i <- as.integer(i)
class(i)

## [1] "integer"
```

Kiểu dữ liệu character

```
str <- 'ABCDEF'
nchar(str)          # Độ dài string

## [1] 6

substr(str, 2, 3) # Lấy chuỗi con từ vị trí thứ 2 đến vị trí thứ 3

## [1] "BC"

ar <- c('abcd', 'ef', 'x')
nchar(ar)

## [1] 4 2 1

substr(ar, 2, 3)

## [1] "bc" "f"  ""
```

Kiểu dữ liệu Date

```
dd <- Sys.Date()
tt <- Sys.time()
class(dd)

## [1] "Date"

dd

## [1] "2017-09-04"

class(tt)

## [1] "POSIXct" "POSIXt"

tt

## [1] "2017-09-04 06:56:59 +07"
```

Kiểu dữ liệu `matrix`, `array`

- Vector: mảng (array) 1 chiều
- `matrix`: `array` hai chiều
- `array` có thể có $k > 2$ chiều

```
(v <- c(1,2,3))

## [1] 1 2 3

(m <- matrix(c(1,2,3,4,5,6), nrow=2, ncol=3, byrow=T))

##      [,1] [,2] [,3]
## [1,]    1    2    3
## [2,]    4    5    6

(m <- matrix(c(1,2,3,4,5,6), nrow=2, ncol=3, byrow=F))

##      [,1] [,2] [,3]
## [1,]    1    3    5
## [2,]    2    4    6

m[1,2]

## [1] 3
```

Kiểu dữ liệu matrix, array

```
v <- 1 : 24
(a <- array(v, dim = c(3, 5, 2)))
```

```
## , , 1
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    4    7   10   13
## [2,]    2    5    8   11   14
## [3,]    3    6    9   12   15
##
## , , 2
##
##      [,1] [,2] [,3] [,4] [,5]
## [1,]   16   19   22    1    4
## [2,]   17   20   23    2    5
## [3,]   18   21   24    3    6
```

```
a[2,3,1]
```

```
## [1] 8
```

```
a[3,4,2]
```

```
## [1] 3
```

Kiểu dữ liệu `list`

- Kiểu dữ liệu này có thể chứa bất kỳ đối tượng nào
- Có thể truy cập đến các đối tượng trong list qua chỉ số hoặc tên

```
ch <- letters[1:5]
int <- as.integer(1:7)
m <- matrix(runif(10), nrow=2, byrow=T)
(mylist <- list(ch, int, m))

## [[1]]
## [1] "a" "b" "c" "d" "e"
##
## [[2]]
## [1] 1 2 3 4 5 6 7
##
## [[3]]
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0.1957738 0.4242057 0.5632126 0.3628713 0.8350565
## [2,] 0.9336440 0.4088234 0.7965890 0.6108089 0.8841672

mylist[[1]]

## [1] "a" "b" "c" "d" "e"
```

Kiểu dữ liệu `list`

```r
ch <- letters[1:5]
int <- as.integer(1:7)
m <- matrix(runif(10), nrow=2, byrow=T)
(mylist <- list(letter=ch, int=int, mat=m))
```

```
## $letter
## [1] "a" "b" "c" "d" "e"
##
## $int
## [1] 1 2 3 4 5 6 7
##
## $mat
##           [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0.8350613 0.9397688 0.8035827 0.6942886 0.4209943
## [2,] 0.7192759 0.6079472 0.7527440 0.8535885 0.6045554
```

```r
mylist$letter
```

```
## [1] "a" "b" "c" "d" "e"
```

```r
mylist[[1]]
```

```
## [1] "a" "b" "c" "d" "e"
```

Kiểu dữ liệu data.frame

- data.frame có thể hiểu như một bảng trong SQL, mỗi cột của data.frame chứa các phần tử dữ liệu cùng kiểu
- Các hàng chứa các phần tử thuộc các kiểu dữ liệu khác nhau

```
# Dữ liệu iris của Fisher ở dạng data frame
head(iris)

##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa

str(iris)

## 'data.frame': 150 obs. of  5 variables:
##  $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
##  $ Sepal.Width : num  3.5 3 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
##  $ Petal.Length: num  1.4 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
##  $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.1 ...
##  $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
```

Truy cập đến các phần tử dữ liệu của `data.frame`

```
iris[2,3]

## [1] 1.4

iris[3,2]

## [1] 3.2

iris$Sepal.Width[3] # Tương đương với iris[3,2]

## [1] 3.2
```

Vào/ra dữ liệu với data.frame

```
write.csv(iris, 'iris.csv', row.names=F)
system('ls -l iris.csv; head -3 iris.csv', intern=T)

## [1] "-rw-rw-r-- 1 chau chau 4026 Sep  4 06:56 iris.csv"
## [2] "\"Sepal.Length\",\"Sepal.Width\",\"Petal.Length\",\"Petal.Width\",\"Spec
## [3] "5.1,3.5,1.4,0.2,\"setosa\""
## [4] "4.9,3,1.4,0.2,\"setosa\""

myiris <- read.csv('iris.csv', sep=',', header=T)
head(myiris)

##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa
## 4          4.6         3.1          1.5         0.2  setosa
## 5          5.0         3.6          1.4         0.2  setosa
## 6          5.4         3.9          1.7         0.4  setosa
```

Thao tác dữ liệu với `data.frame`

- Có nhiều thư viện thao tác dữ liệu với `data.frame`. Tham khảo `dplyr`, `tidyr`, `data.table`
- Thư viện tổng hợp: `tidyverse`

Các hàm thống kê mô tả và đồ thị

Trung bình, trung vị, phương sai, độ lệch chuẩn

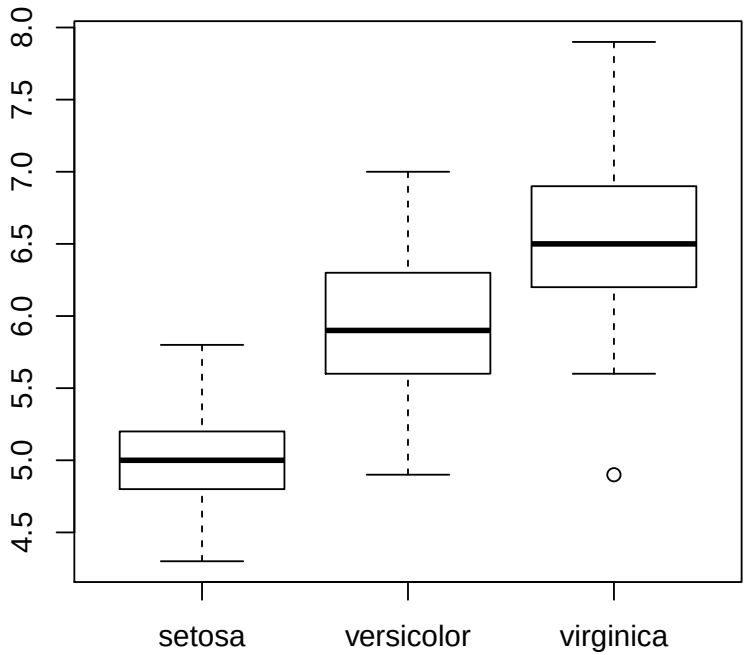
```
mean(x)          # Trung bình

## [1] 0.4312446

median(x)        # Trung vị

## [1] 0.4543482

var(x)           # Phương sai

## [1] 0.0702071

sd(x)            # Độ lệch chuẩn

## [1] 0.2649662
```

Biểu đồ barplot

```
arth <- read.csv('arth.csv')

## Warning in file(file, "rt"): cannot open file 'arth.csv': No such file or directory
## Error in file(file, "rt"): cannot open the connection

barplot(table(arth$Improved)) # Hiệu quả chữa bệnh

## Error in table(arth$Improved): object 'arth' not found

plot(table(arth$Improved, arth$Sex)) # Hiệu quả chữa bệnh chia theo giới tính

## Error in table(arth$Improved, arth$Sex): object 'arth' not found
```

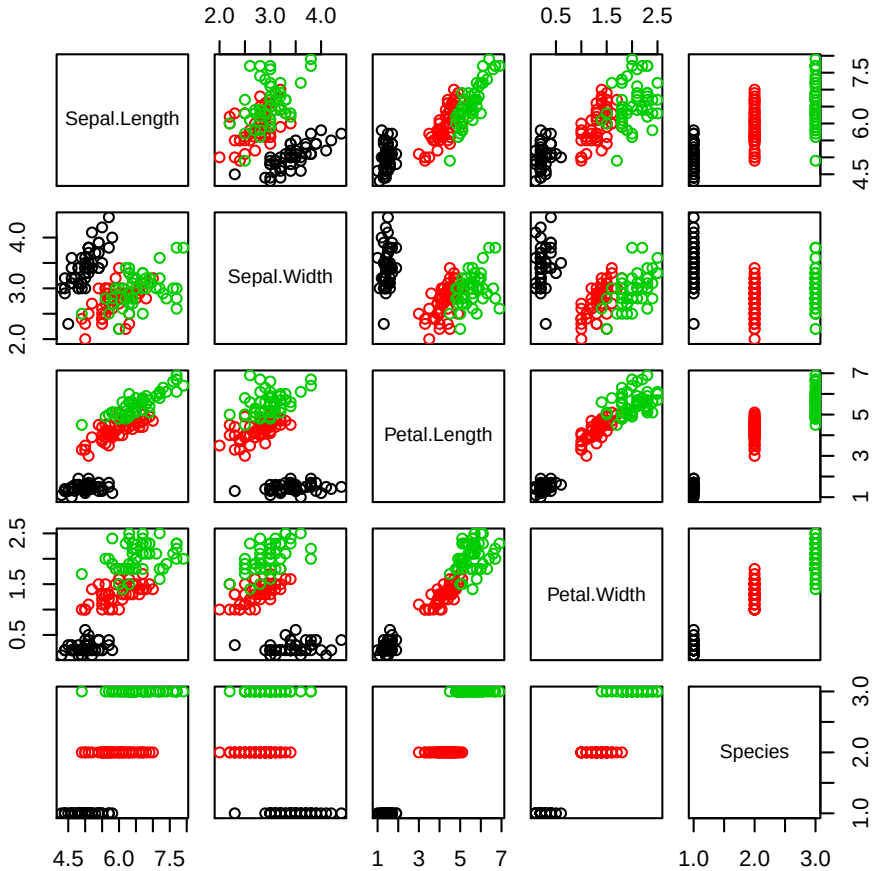
Biểu đồ boxplot

```
head(iris, 3)

##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
## 2          4.9         3.0          1.4         0.2  setosa
## 3          4.7         3.2          1.3         0.2  setosa

boxplot(data=iris, Sepal.Length~Species)
```



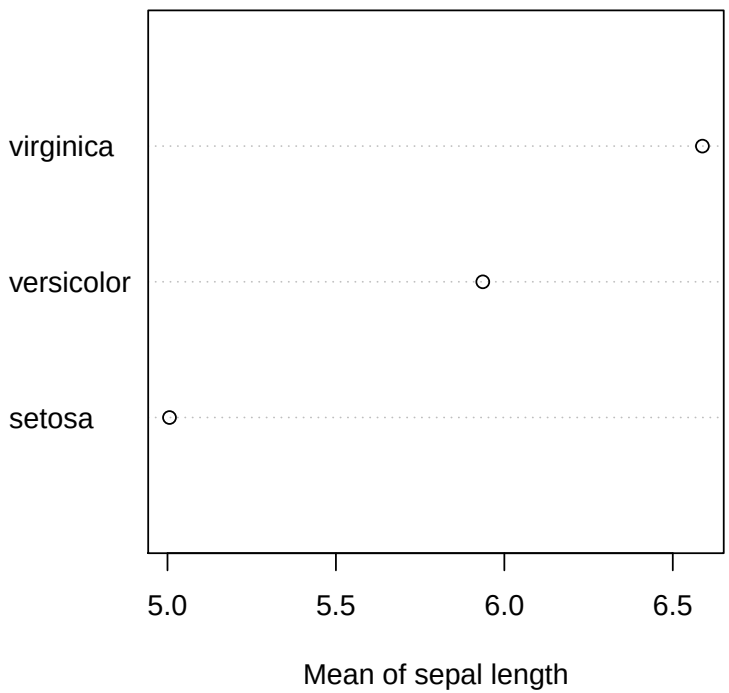
Biểu đồ scatterplot

```
plot(iris, col=iris$Species)
```



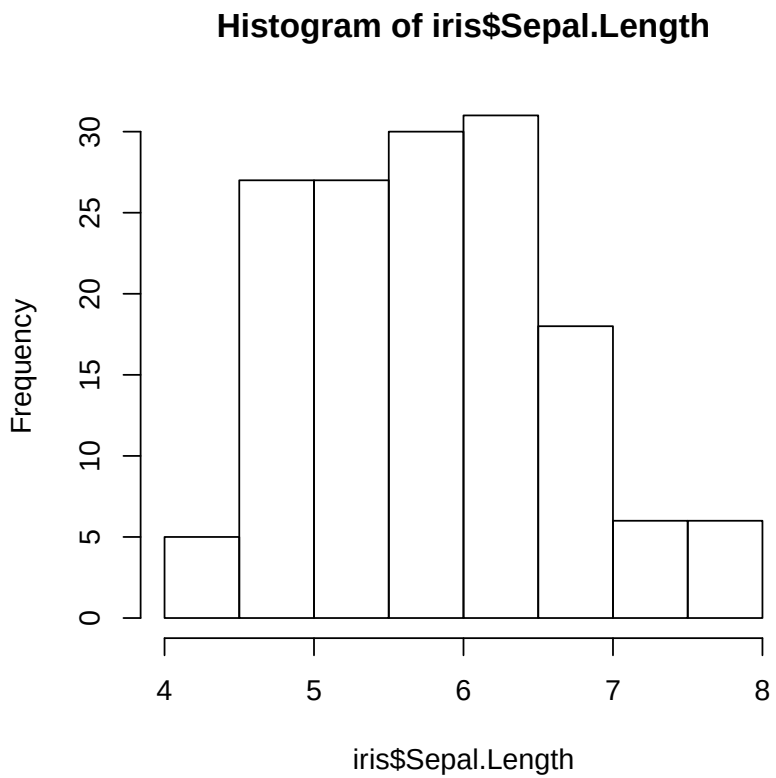
Biểu đồ dotchart

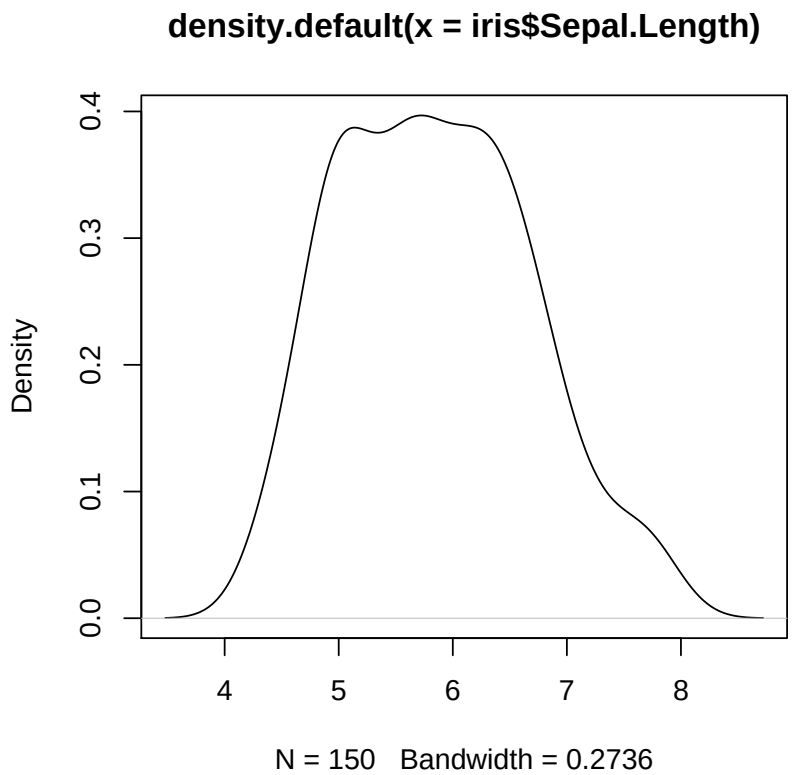
```
library(tidyverse)
df <- iris %>% group_by(Species) %>% summarise(sepalmean = mean(Sepal.Length))
sp <- df$sepalmean
names(sp) <- df$Species
dotchart(sp, xlab='Mean of sepal length')
```



Biểu đồ histogram và density

```
hist(iris$Sepal.Length, breaks=10)
plot(density(iris$Sepal.Length))
```





Biểu đồ densityplot với thư viện lattice

```
library(lattice)
densityplot(x=iris$Sepal.Length, xlab='Sepal Length')
densityplot(x=iris$Sepal.Length, group=iris$Species, xlab='Sepal Length')
```