### Thực hành thống kê: Tuần 3

# Mô tả dữ liệu

Nội dung chính:

- 1. Nhập, xử lý, xuất dữ liệu
- 2. Tóm tắt dữ liệu
- 3. Mô tả dữ liệu bằng đồ thị

# 1. Nhập, xử lý, xuất dữ liệu

## 1.1 Nhập dữ liệu

### a) Working directory

Trước khi nhập dữ liệu ta nên thiết lập thư mục làm việc. Đây là thư mục chứa những thứ mà ta muốn tương tác với R (như file dữ liệu, code script R, hình ảnh, đồ thị, package,...).

- Thiết lập thư mục làm việc: `setwd()` ; ví dụ ở ổ D, thư mục Works: `setwd('D:/Works')`
- Xem thư mục hiện hành: `getwd()`
- Liệt kê tất cả file trong thư mục làm việc: `list.files()` hoặc `dir()`

```
# Thiết lập thư mục làm việc
setwd('/Users/uyendang/Documents/THTK')
# Xem thư mục hiện hành
getwd()
```

```
## [1] "/Users/uyendang/Documents/THTK"
```

```
# Liệt kê tất cả file trong thư mục làm việc
list.files() # hoặc dir()
```

```
## [1] "BTVN" "Rmd"
```

### b) Workspace

Từ khi mở R (cửa sổ R console xuất hiện) cho đến khi tắt R là một phiên làm việc. Những đối tượng ta tạo ra trong một phiên làm việc được R lưu trong Workspace. Ta có thể lưu lại mọi thứ trong Workspace này để tiếp tục công việc đang làm của ta ở một thời điểm khác.

- Lưu Workspace: `save.image('ten_file.rda')`
- Tải Workspace đã lưu `load('ten_file.rda')`
- Lưu biến đang làm việc, chẳng hạn biến `x` : `save(x, file='ten_file.rda')`
- Khôi phục biến `x` : `load('ten_file.rda')` , `ten_file.rda` là file chứa biến `x` vừa lưu ở trên.
- Xóa 1 biến ra khỏi Workspace: `rm(x)`
- Xóa tất cả: `rm(list=ls())`
- Liệt kê tất cả những biến trong Workspace: `ls()`
- Xem thông tin của biến `x` : `str(x)`
- Xem thông tin của tất cả biến đang làm việc: `ls.str()`

### c) Nhập dữ liệu

- Nhập trực tiếp

```
Data1 <- data.frame(
  Id = c(1,2,3,4,5),
  Ten = c("Hiếu","Thi","Trung","Huyền","Thư"),
  GioiTinh = c("Nam","Nu","Nam","Nu","Nu"),
  ChieuCao = c(1.75,1.66,1.70,1.58,1.64),
  CanNang = c(67,56,62,52,50))
Data1
```

```
##   Id   Ten GioiTinh ChieuCao CanNang
## 1  1  Hiếu      Nam     1.75      67
## 2  2   Thi       Nu     1.66      56
## 3  3 Trung      Nam     1.70      62
## 4  4 Huyền       Nu     1.58      52
## 5  5   Thư       Nu     1.64      50
```

- Nhập từ file .csv: `read.csv()`

```
# Data2 <-read.csv("Data02.csv")
```

- Nhập từ file .txt: `read.table()`

```
# Data3 <- read.table('D:/Đường dẫn/solieu.txt ', header=TRUE, sep= "")
```

## 1.2 Xử lý dữ liệu

- Đưa một data frame vào workspace để xử lý: `attach(dataframe)`
- Tách dữ liệu: `subset( bien_goc, dieu_kien)`
- Nhập 2 dataframe thành một: `merge(frame_1, frame_2, by=)`
- Biến đổi số liệu:
  - Từ biến dạng numeric sang biến phân loại, sử dụng các phép toán logic hoặc dùng lệnh `replace( )`
  - Từ biến dạng numeric sang nhân tố: `factor( )`
- Phân nhóm số liệu, dùng hàm `cut` hoặc `cut2` của thư viện `Hmisc` .

```
# Đưa data frame Data1 vào workspace
attach(Data1)
# Tách dữ liệu thành 2 data.frame theo GioiTinh
nam <- subset(Data1, GioiTinh=='Nam')
nam
```

```
##   Id   Ten GioiTinh ChieuCao CanNang
## 1  1  Hiếu      Nam     1.75      67
## 3  3 Trung      Nam     1.70      62
```

```
nu <- subset(Data1, GioiTinh=='Nu')
nu
```

```
##   Id   Ten GioiTinh ChieuCao CanNang
## 2  2   Thi       Nu     1.66      56
## 4  4 Huyền       Nu     1.58      52
## 5  5   Thư       Nu     1.64      50
```

```
# Tách dữ liệu với GioiTinh là Nữ và cao trên 1.60cm
nucao <- subset(Data1, GioiTinh=='Nu' & ChieuCao >= 1.60)
nucao
```

```
##   Id Ten GioiTinh ChieuCao CanNang
## 2  2 Thi       Nu     1.66      56
## 5  5 Thư       Nu     1.64      50
```

```
# Merge 2 data frame
Data2 = data.frame(
  Id = c(1,2,3,4,5,6,7,8,9),
  Ten = c("Hiếu","Thi","Trung","Huyền","Thư","Nam","Hoàng","Chi","Nguyên"),
  Tuoi = c(18,19,20,16,25,34,15,57,43)
  )
df1 <- merge(Data1, Data2, by="Id")
df1
```

```
##   Id Ten.x GioiTinh ChieuCao CanNang Ten.y Tuoi
## 1  1  Hiếu      Nam     1.75      67  Hiếu   18
## 2  2   Thi       Nu     1.66      56   Thi   19
## 3  3 Trung      Nam     1.70      62 Trung   20
## 4  4 Huyền       Nu     1.58      52 Huyền   16
## 5  5   Thư       Nu     1.64      50   Thư   25
```

```
df2 <- merge(Data1, Data2, by="Id", all=TRUE)
df2
```

```
##   Id Ten.x GioiTinh ChieuCao CanNang  Ten.y Tuoi
## 1  1  Hiếu      Nam     1.75      67   Hiếu   18
## 2  2   Thi       Nu     1.66      56    Thi   19
## 3  3 Trung      Nam     1.70      62  Trung   20
## 4  4 Huyền       Nu     1.58      52  Huyền   16
## 5  5   Thư       Nu     1.64      50    Thư   25
## 6  6  <NA>     <NA>       NA      NA    Nam   34
## 7  7  <NA>     <NA>       NA      NA  Hoàng   15
## 8  8  <NA>     <NA>       NA      NA    Chi   57
## 9  9  <NA>     <NA>       NA      NA Nguyên   43
```

```
# Biến đổi dữ liệu từ biến dạng numeric sang biến phân loại
# < 18 tuổi: thiếu niên (1); 18-35: thanh niên (2), >= 35: trung niên (3)
Nhomtuoi <- Data2$Tuoi
Nhomtuoi[Data2$Tuoi < 18] <- 1
Nhomtuoi[Data2$Tuoi >= 18 & Data2$Tuoi < 35] <- 2
Nhomtuoi[Data2$Tuoi >= 35] <- 3
Data2 <- data.frame(Data2, Nhomtuoi)
Data2
```

```
##   Id    Ten Tuoi Nhomtuoi
## 1  1   Hiếu   18        2
## 2  2    Thi   19        2
## 3  3  Trung   20        2
## 4  4  Huyền   16        1
## 5  5    Thư   25        2
## 6  6    Nam   34        2
## 7  7  Hoàng   15        1
## 8  8    Chi   57        3
## 9  9 Nguyên   43        3
```

```
# Biến đổi dữ liệu từ biến dạng numeric sang biến phân loại: sử dụng replace()
Nhomtuoi2 <- Data2$Tuoi
Nhomtuoi2 <- replace(Nhomtuoi2, Data2$Tuoi < 18, 1)
Nhomtuoi2 <- replace(Nhomtuoi2, Data2$Tuoi >= 18 & Data2$Tuoi <35, 2)
Nhomtuoi2 <- replace(Nhomtuoi2, Data2$Tuoi >= 35, 3)
Data2 <- data.frame(Data2, Nhomtuoi2)
Data2
```

```
##   Id    Ten Tuoi Nhomtuoi Nhomtuoi2
## 1  1   Hiếu   18        2         2
## 2  2    Thi   19        2         2
## 3  3  Trung   20        2         2
## 4  4  Huyền   16        1         1
## 5  5    Thư   25        2         2
## 6  6    Nam   34        2         2
## 7  7  Hoàng   15        1         1
## 8  8    Chi   57        3         3
## 9  9 Nguyên   43        3         3
```

```
mean(Data2$Nhomtuoi2)
```

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

```
# Biến đổi dữ liệu từ biến dạng số sang nhân tố
Data2$Nhomtuoi2 <- factor(Data2$Nhomtuoi2)
Data2$Nhomtuoi2
```

```
## [1] 2 2 2 1 2 2 1 3 3
## Levels: 1 2 3
```

```
mean(Data2$Nhomtuoi2)
```

```
## Warning in mean.default(Data2$Nhomtuoi2): argument is not numeric or logical:
## returning NA
```

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

```
# Phân nhóm số liệu, dùng hàm `cut`
group1 <- cut(Data2$Tuoi, 2)
table(group1)
```

```
## group1
## (15,36] (36,57]
##       7       2
```

```
group2 <- cut(Data2$Tuoi, c(0,18,35,60))
table(group2)
```

```
## group2
##  (0,18] (18,35] (35,60]
##       3       4       2
```

### 1.3. Xuất dữ liệu

a. Định dạng R (.rda): `save()`

```
save(Data2, file= 'data2.rda')
```

b. Định dạng .csv: `write.csv`

```
write.csv(Data2,"mydata.csv")
```

c. Định dạng text (.txt)

```
write.table(Data2, "mydata.txt", sep=",")
```

# 2. Tóm tắt dữ liệu

a. Một số hàm về vec-tơ:

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

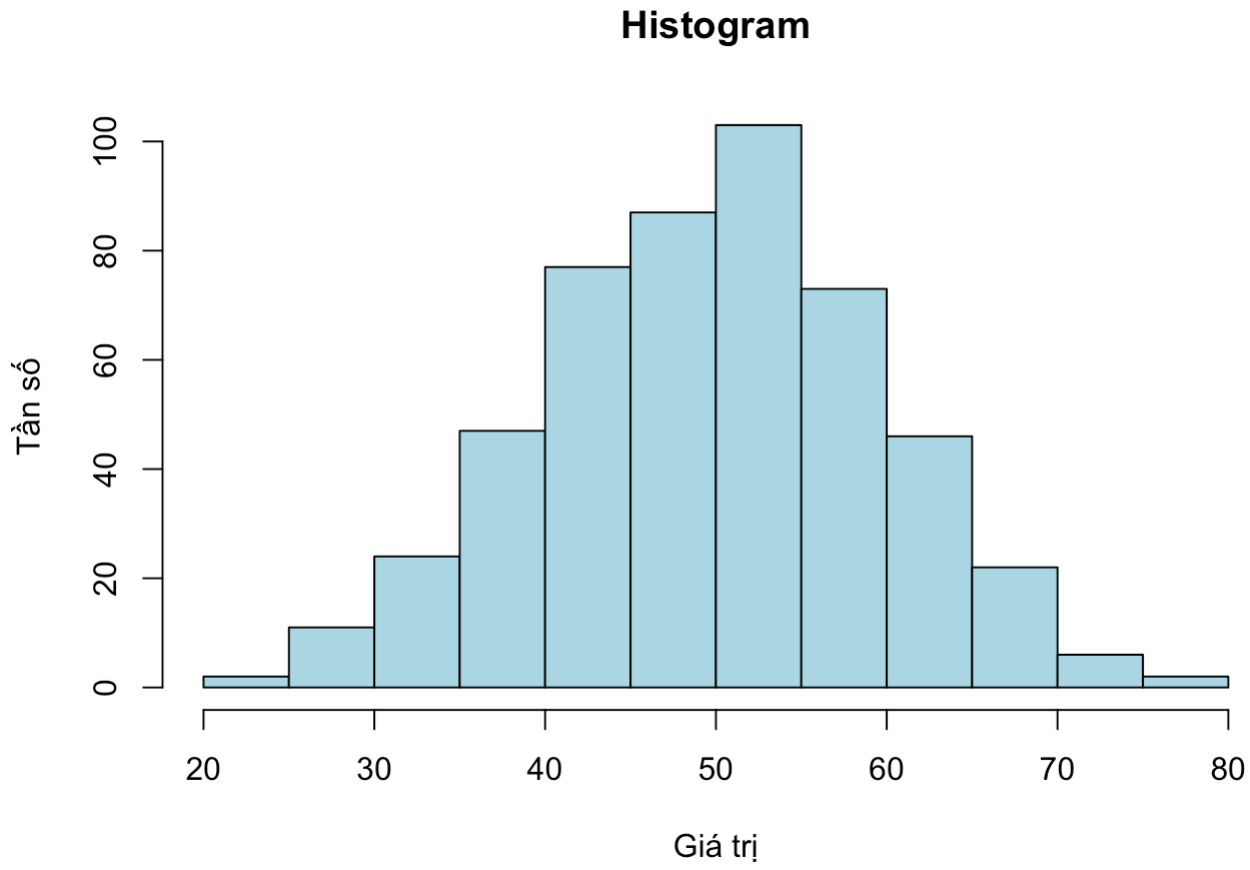
b. Tóm tắt 1 đối tượng (vec-tơ, dataframe)

- `summary(object)` : thông tin chung của object
- `str(object)` : cấu trúc của object

# 3. Mô tả dữ liệu bằng đồ thị

- Histogram: `hist(x, ...)`

```
# Dữ liệu mẫu
data <- rnorm(500, mean = 50, sd = 10)

# Vẽ histogram
hist(data, col = "lightblue", main = "Histogram", xlab = "Giá trị", ylab = "Tần số")
```



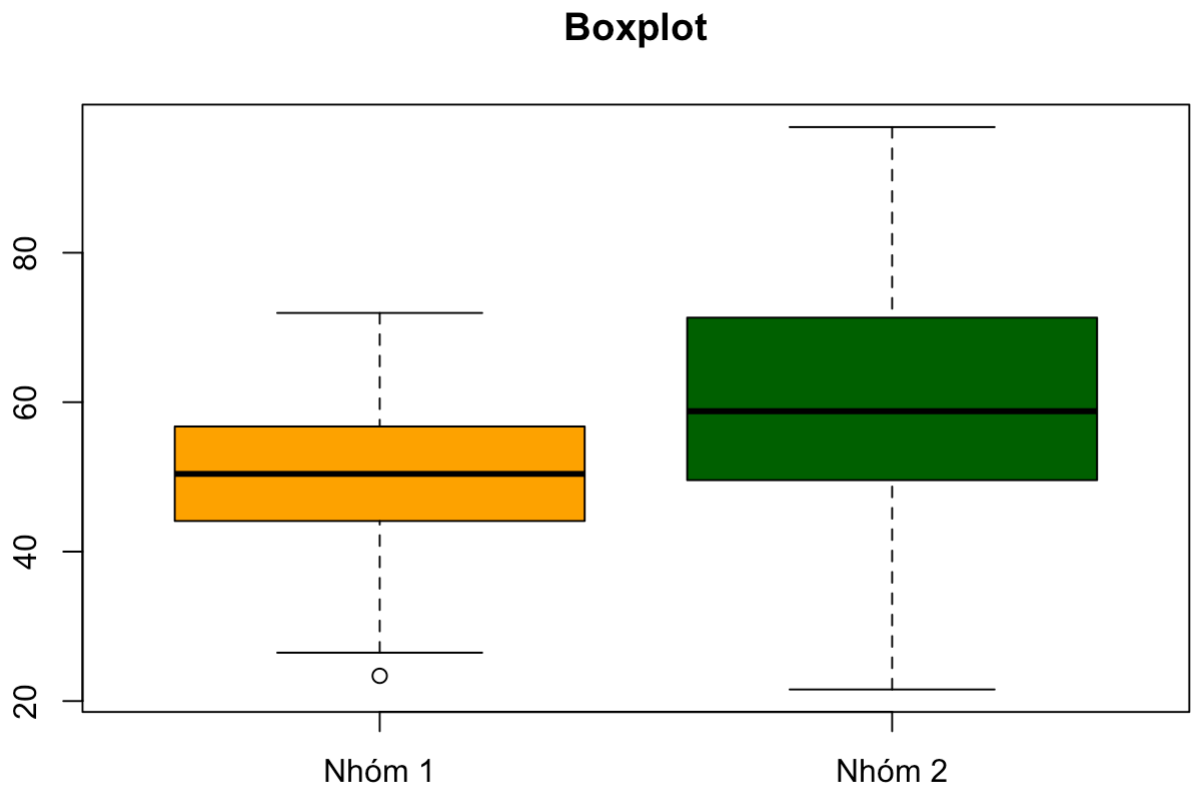

- Stem & Leaf

```
# Dữ liệu mẫu
data <- c(23, 25, 26, 29, 32, 35, 36, 37, 39, 42, 45, 47, 48, 52, 55, 57, 60)

# Vẽ biểu đồ stem-and-leaf
stem(data)
```

```
##
##   The decimal point is 1 digit(s) to the right of the |
##
##   2 | 3569
##   3 | 25679
##   4 | 2578
##   5 | 257
##   6 | 0
```

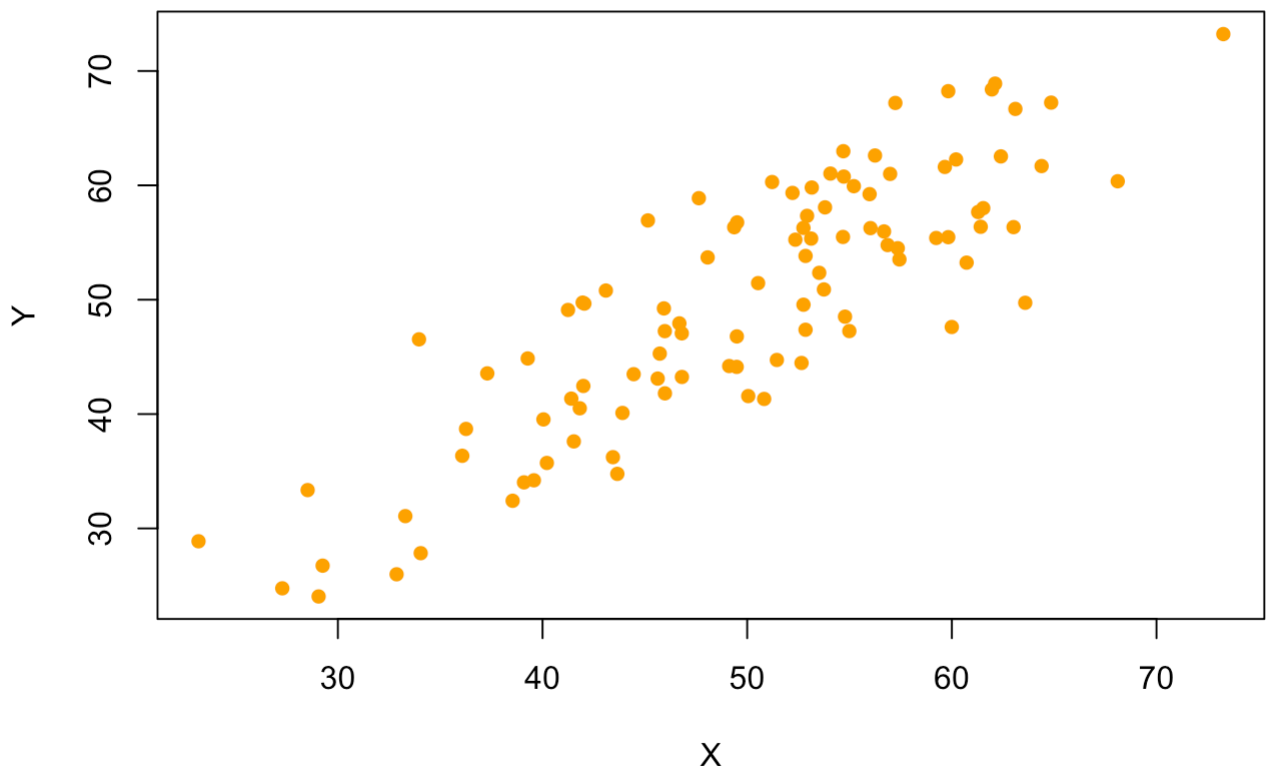
- Boxplot

```
# Dữ liệu mẫu
data1 <- rnorm(100, mean = 50, sd = 10)
data2 <- rnorm(100, mean = 60, sd = 15)

# Vẽ boxplot
boxplot(data1, data2, names = c("Nhóm 1", "Nhóm 2"), col = c("orange", "darkgreen"),
main = "Boxplot")
```

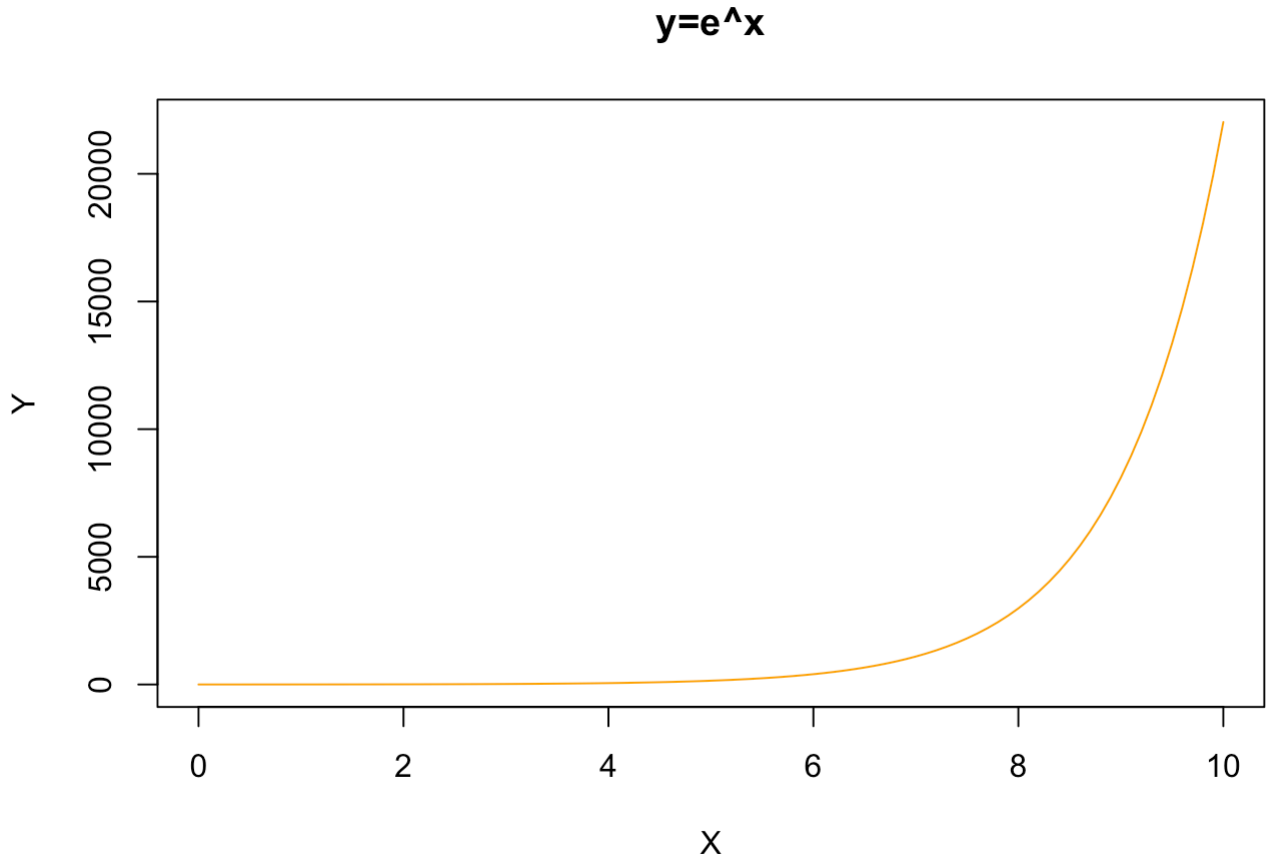



- Scatter plot

```
# Dữ liệu mẫu
x <- rnorm(100, mean = 50, sd = 10)
y <- x + rnorm(100, mean = 0, sd = 5)

# Vẽ biểu đồ phân tán
plot(x, y, col = "orange", pch = 16, main = "Biểu đồ phân tán", xlab = "X", ylab =
"Y")
```

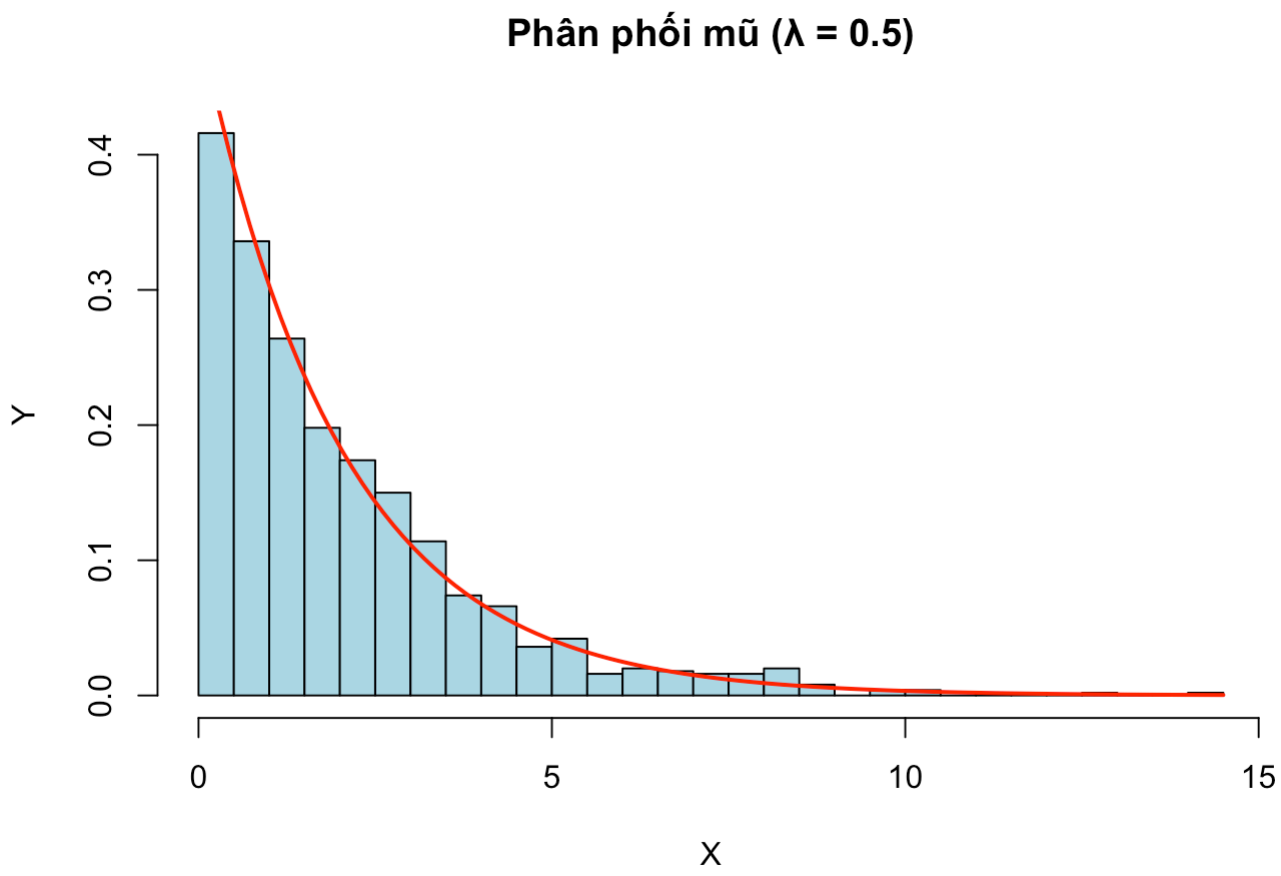



```
# Plot đồ thị hàm số y = e^x
x <- seq(0,10,0.1)
y <- exp(x)
plot(y~x, type='l', col='orange',  xlab = "X", ylab = "Y", main ='y=e^x')
```

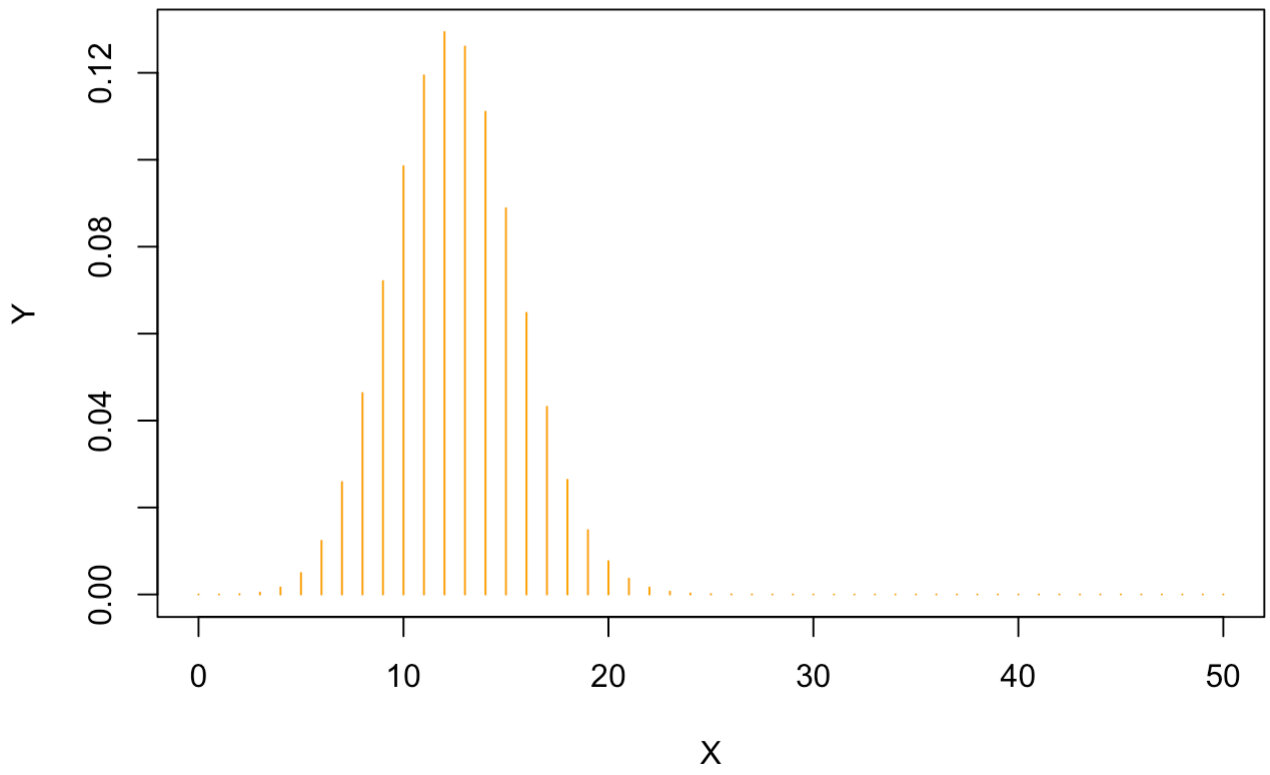



- Đồ thị hàm mật độ

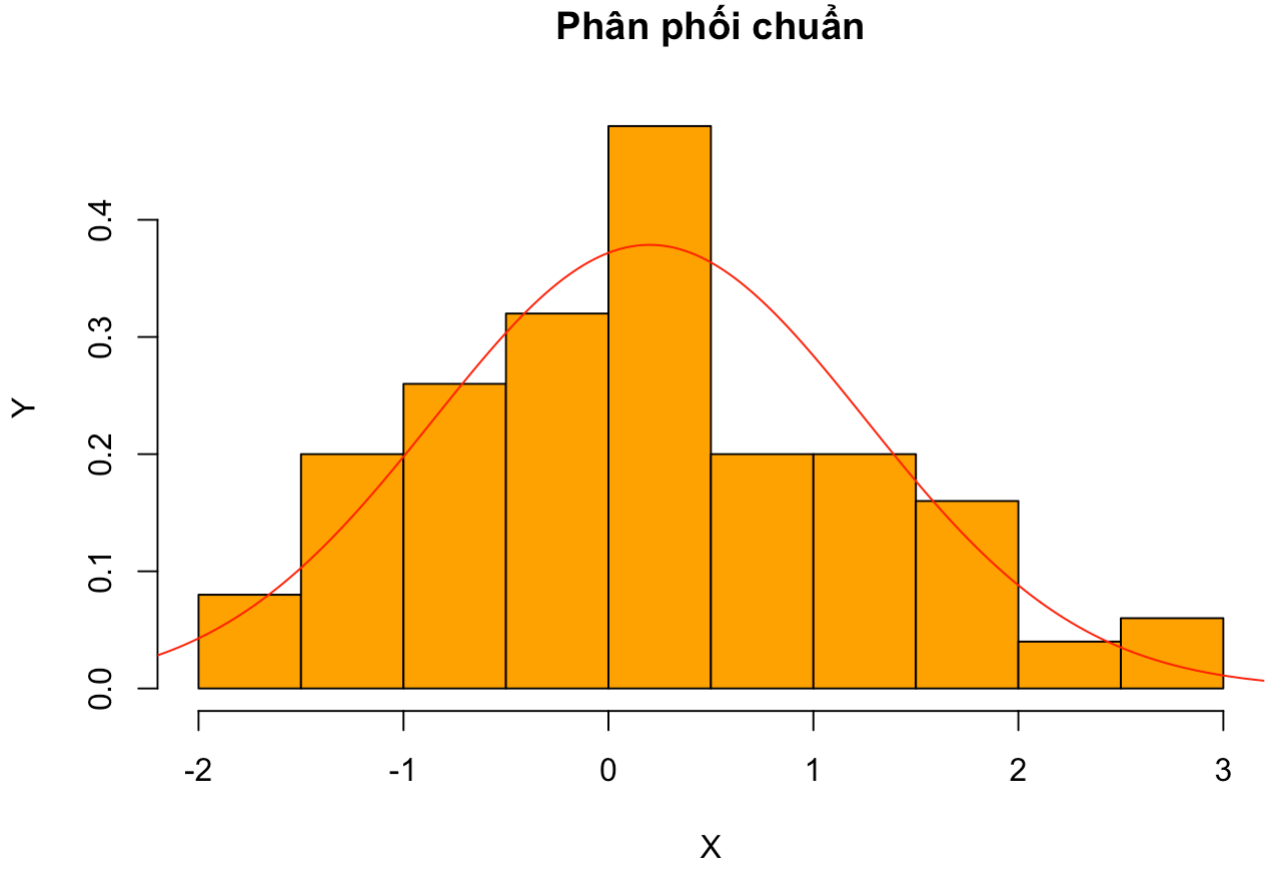
```
# Phân phối mũ
set.seed(123)
exp_data <- rexp(1000, rate = 0.5)  # 1000 giá trị từ Exp(λ = 0.5)

hist(exp_data, prob=T, col = "lightblue", breaks = 30, main = "Phân phối mũ (λ = 0.
5)",  xlab = "X", ylab = "Y")
curve(dexp(x, rate = 0.5), add = TRUE, col = "red", lwd = 2)
```




```r
# Phân phối nhị thức
x <- 0:50
y <- dbinom(x,50,0.25)
plot(x, y, 'h', col='orange', main = "Phân phối nhị thức",  xlab = "X", ylab = "Y")
```




```
# Phân phối chuẩn
sample <- rnorm(100)
hist(sample, prob=T, col='orange',main = "Phân phối chuẩn",  xlab = "X", ylab = "Y")
mu <- mean(sample)
sigma <- sd(sample)
x <- seq(-4,4,length=500)
y <- dnorm(x,mu,sigma)
lines(x,y,col = "red") # hoặc curve(dnorm(x,mu,sigma), add = TRUE, col = "darkgreen",
lwd = 2)
```




# Bài tập

**Bài 1.** Tạo vec-tơ: x =[1,2,5,7,-3,0,5,1,5,6] và y=[2,2,0,-5,7,8,11,9,3,2]

a. Tính x+y, x*y,x-y.
b. Tao z =[Những phần tử chẵn của x], t =[Những phần tử lẻ của y]
c. Trích những phần tử lớn hơn 0 của x và y.
d. Tính trung bình, độ lệch tiêu chuẩn, sai số chuẩn của x và y.
e. Tìm phần tử lớn nhất, bé nhất của x, y.
f. Sắp xếp x tăng dần, y giảm dần.
g. Lưu x và y.

**Bài 2.** Nhập số liệu từ file data01.csv gán vào frame data1. Thực hiện:

a. Tính trung bình, phương sai, trung vị của các biến FPSA và TPSA.
b. Vẽ biểu đồ dạng đường, boxplot cho FPSA và TPSA.
c. Tách những giá trị của biến FPSA có K=0 và K=1.
d. Đọc số liệu từ file data02.csv gán vào frame data2, merge 2 frame này theo biến K.
e. Tạo biến mới tPSA theo yêu cầu sau: Nếu tuổi <= 30, tPSA=0; nếu 30< tuổi <=50, tPSA=1; nếu tuổi>50, tPSA =2. Tạo bảng thống kê cho tPSA.

**Bài 3.**

a. Tạo ngẫu nhiên 100 giá trị có phân phối nhị thức, với n = 60 và xác suất thành công mỗi lần 0.4. Vẽ biểu đồ tổ chức tần số.
b. Tạo ngẫu nhiên 100 giá trị có phân phối Poisson với lambda = 4, vẽ biểu đồ tổ chức tần số.
c. Tạo ngẫu nhiên 100 giá trị có phân phối chuẩn có trung bình là 50 và độ lệch tiêu chuẩn 4. Vẽ hàm phân phối, hàm mật độ.
d. Tạo ngẫu nhiên 100 giá trị có phân phối mũ với lambda=1/25. Vẽ hàm phân phối,hàm mật độ

**Bài 4.** File diesel_engine.dat và diesel_time.csv chứa số liệu về hoạt động của các động cơ chạy bằng dầu diesel. Thực hiện:

a. Đọc số liệu từ hai file này, gán và hai dataframe, đặt tên hai dataframe cùng tên với file.
b. Liệt kê tên các biến có trong hai dataframe vừa nhập.
c. Xác định có bao nhiêu dữ liệu bị khuyết (missing data) trong diesel_engine. Thay thế các giá trị khuyết trong biến speed bằng 1500, biến load bằng 20.
d. Tính: trung bình, phương sai, độ lệch tiêu chuẩn, giá trị lớn nhất, nhỏ nhất của biến alcohol trong dataframe diesel_engine.
e. Ghép hai dataframe diesel_engine và diesel_time lại thành một frame có tên là diesel.
f. Trích giá trị của biến run (số thứ tự các động cơ) mà có thời gian trễ (biến delay) dưới 1.000.
g. Đếm xem có bao nhiêu động cơ có timing bằng 30.
h. Vẽ biểu đồ boxplot cho các biến speed, timing và delay.
i. Vẽ biểu đồ phân tán cho các cặp biến (timing, speed), (temp, press).
j. Chuyển biến load sang biến nhân tố.
k. Chia phạm vi giá trị của biến delay thành 4 đoạn đều nhau và đếm số giá trị nằm trong các đoạn đó. Tạo bảng thống kê và vẽ biểu đồ cột.
l. Chia phạm vi giá trị của biến delay thành 4 đoạn như sau: (0.283, 0.7], (0.7, 0.95], (0.95, 1.2], (1.2, 1.56]. Tạo bảng thống kê và vẽ biểu đồ cột.