



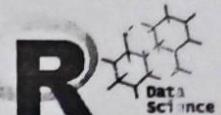
R for Data Science

Bài 5: *Function*

Phòng LT & Mạng

https://csc.edu.vn/lap-trinh-va-cSDL/R-Programming-Language-for-Data-Science_190

2020



Giới thiệu



- ❑ Function (phương thức) là tập hợp các dòng code được viết để thực hiện một chức năng nào đó.
- ❑ Mục đích xây dựng function là để có thể tái sử dụng khi cần.
- ❑ R cung cấp rất nhiều function xây dựng sẵn (*built-in function*), và người dùng cũng có thể tự xây dựng function (*user-defined function*)



Nội dung

1. Built-in function
2. User-defined function
3. Gọi sử dụng function



Built-in function

❑ Function phải có arguments (gồm:
variable, parameter, option)

❑ Chỉ cần gọi sử dụng khi cần

❑ Ví dụ: `v <- c(1:100)
summary(v)`

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.00	25.75	50.50	50.50	75.25	100.00

`print(sum(v))`

[1] 5050



Built-in function

- Là các phương thức được cung cấp sẵn (từ các packages chứa phương thức của R)

- Chỉ cần gọi sử dụng khi cần

```
v <- c(1:100)
summary(v)
```

□ Ví dụ:

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.00	25.75	50.50	50.50	75.25	100.00

```
print(sum(v))
```

```
[1] 5050
```

R programming language for Data Science

5

Built-in function

□ Packages

- Là các module dùng cho việc phân tích

- Có khoảng 6000 packages trong R

- Cài đặt trực tiếp bằng:

```
install.packages("tên_packages")
```

- Ví dụ: install.packages("tidyverse")

Built-in function

□ Một số packages phổ biến

- Base: R base package
- Ggplot2: data visualization
- Foreign: đọc dữ liệu từ SPSS, Stata...
- Tidyverse: chuẩn bị dữ liệu cho phân tích
- Rms: regression models
- Cluster: cluster analysis

□ Xem danh sách các package đã cài: gõ library()



R programming language for Data Science

7

Built-in function



□ Xem thông tin về built-in function, gõ help(function_name) ở màn hình console

- Ví dụ: help(mean)

R: Arithmetic Mean ▾ Find in Topic
mean {base}

Arithmetic Mean

Description

Generic function for the (trimmed) arithmetic mean.

Usage

`mean(x, ...)`

Default S3 method:

`mean(x, trim = 0, na.rm = FALSE, ...)`



R programming language for Data Science

8

Built-in function

□ Xem ví dụ của built-in function, gõ example(function_name) ở màn hình console

- Ví dụ:

```
> example(sum)

sum<- ## Pass a vector to sum, and it will add the elements together.
sum> sum(1:5)
[1] 15

sum<- ## Pass several numbers to sum, and it also adds the elements.
sum> sum(1, 2, 3, 4, 5)
[1] 15

sum<- ## In fact, you can pass vectors into several arguments, and everything gets added.
sum> sum(1:2, 3:5)
[1] 15

sum<- ## If there are missing values, the sum is unknown, i.e., also missing, ....
sum> sum(1:5, NA)
[1] NA

sum<- ## ... unless we exclude missing values explicitly:
sum> sum(1:5, NA, na.rm = TRUE)
[1] 15
```

R programming language for Data Science

9

Nội dung

1. Built-in function

2. User-defined function

3. Gọi sử dụng function

- Là những phương thức do người dùng tự xây dựng để giải quyết một công việc nào đó
- Trong User-defined function có thể gọi các Built-in function



User-defined function

□ Cấu trúc

```
function_name <- function(arg_1, arg_2, ...) {  
  function body  
  return(value)  
}
```

- **function_name**: tên của function
- **arg_1, arg_2...**: các tham số truyền vào cho function. Tham số là tùy chọn, có thể có hoặc không. Ta cũng có thể đặt giá trị mặc định (default value) cho function.
- **function body**: các câu lệnh trong function
- **return(value)**: giá trị trả về của function



User-defined function

- Ví dụ: Xây dựng phương thức tính tổng các số chẵn từ start đến end

- In kết quả trực tiếp

```
sum_even.function <- function(start, end){
  sum_even <- 0
  for(i in start:end){
    if(i%%2 ==0){
      sum_even <- sum_even + i
    }
  }
  print(sum_even)
}
```

User-defined function

- Trả kết quả về

```
sum_even.function <- function(start, end){
  sum_even <- 0
  for(i in start:end){
    if(i%%2 ==0){
      sum_even <- sum_even + i
    }
  }
  return(sum_even)
}
```

Nội dung

1. Built-in function
2. User-defined function
3. Gọi sử dụng function



Gọi sử dụng function

□ Công thức

`function_name(giá_trị_1, giá_trị_2,...)`

● Ví dụ:

```
print(paste("Sum of even numbers is",sum_even.function(1,10)))  
[1] "Sum of even numbers is 30"
```

● Ghi chú: khi muốn sử dụng giá trị trả về của function thì cần khai báo biến để nhận giá trị này.

▪ Ví dụ: `sum_even = sum_even.function(1, 10)`



Gọi sử dụng function

- Chú ý:

- Có thể gọi function kèm tên tham số = value với thứ tự tham số có thể thay đổi nhưng số lượng tham số không thay đổi

- Ví dụ:

```
print(paste("Sum of even numbers is",sum_even.function(start = 1, end = 10)))
[1] "Sum of even numbers is 30"

print(paste("Sum of even numbers is",sum_even.function(end = 10, start = 1)))
[1] "Sum of even numbers is 30"
```



Gọi sử dụng function

- Chú ý:

- Có thể gán giá trị mặc định cho tham số khi xây dựng function

- Ví dụ:

```
drink.function <- function(price, type ="Tea"){
  print(paste("With ", price, ", you can drink", type))
}
```

```
drink.function(10000)
drink.function(10000, "Coffee")

[1] "With 10000 , you can drink Tea"
[1] "With 10000 , you can drink Coffee"
```



Gọi sử dụng function

- Chú ý:

- Để thuận tiện trong việc tái sử dụng, ta viết các function vào 1 script. Sau đó, gọi script đó vào nơi cần dùng:

```
source("ten_duong_dan_chua_functions")
```

- Ví dụ:

```
source("~/R/chuong6/ex_6_functions.R")
```





Chapter 5: Functions

Exercise 1: Tính BMI và đánh giá

- Chương trình cho phép người dùng nhập vào cân nặng và chiều cao.
- Chương trình tính và in ra chỉ số BMI và đánh giá. Với
 - Cách tính: $BMI = \text{Cân nặng} / (\text{Chiều cao} * \text{Chiều cao})$
 - Bảng đánh giá BMI: Gầy: <18.5 , Bình thường: $18.5 - 24.99$, Thừa cân: ≥ 25

Yêu cầu:

- Xây dựng function tính BMI, đánh giá
- Gọi sử dụng các function đã viết, in kết quả

Exercise 2: Giải PT bậc 1, PT bậc 2

Yêu cầu 1: Giải PT bậc 1: $ax + b = 0$: Nhập vào a, b => Giải PT và in kết quả

Qui tắc xử lý :

- Nếu $a = 0, b < 0$: phương trình vô nghiệm
- Nếu $a = 0, b = 0$: phương trình vô số nghiệm
- Nếu $a \neq 0$: phương trình có nghiệm $x = -b/a$

Yêu cầu 2: Giải PT bậc 2: $ax^2 + bx + c = 0$: Nhập vào a, b, c => Giải PT và in kết quả

Quy tắc xử lý:

- Nếu $a = 0$: Giải phương trình bậc 1: $bx+c = 0$
- Nếu $a \neq 0$: Tính $\Delta = b^2 - 4ac$
 - Nếu $\Delta < 0$ thì phương trình vô nghiệm.
 - Nếu $\Delta = 0$ thì phương trình (2) có nghiệm kép : $x_1 = x_2 = x_0 = -b/2a$
 - Nếu $\Delta > 0$ thì phương trình (2) có 2 nghiệm phân biệt : $x_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a}$

Exercise 3: Xây dựng function

- Hãy xây dựng các function: tinh_tien_dien.function, kiem_tra_SNT.function, tinh_cuoc_xe.function, ... dựa trên các bài tập đã làm
- Gọi sử dụng các function này => In kết quả



Gợi ý:

Exercise 1: Tính BMI và đánh giá

```
In [1]: # tinh bmi
bmi.function <- function(height, weight){
  bmi <- weight/(height * height)
  # print(bmi)
  return(bmi)
}

In [2]: danh_gia_bmi.function <- function(bmi){
  result <-
  if(bmi < 18.5){
    result <- "Gay"
  }else if(bmi<25){
    result <- "Binh thuong"
  }else{
    result <- "Thua can"
  }
  return(result)
}

In [3]: height <- as.numeric(readline(prompt = "Input height (m): "))
weight <- as.numeric(readline(prompt = "Input weight (kg): "))

bmi <- bmi.function(height, weight)
danh_gia <-danh_gia_bmi.function(bmi)
print(paste("BMI: ",bmi))
print(paste("Ket qua: ",danh_gia))

Input height (m): 1.58
Input weight (kg): 55
[1] "BMI: 22.0317256849864"
[1] "Ket qua: Binh thuong"
```

Exercise 2

Giải phương trình bậc 1



```
In [4]: # giao PT bac 1
ptb1.function <- function(a,b){
  result <- ""
  if(a==0 & b!=0){
    result <- "PT vo nghiem"
  }else if(a==0 & b==0){
    result <- "PT vo so nghiem"
  }else{
    result <- paste("Nghiem:", -b/a)
  }
  return(result)
}
```

```
In [5]: a <- as.numeric(readline(prompt = "Input a: "))
b <- as.numeric(readline(prompt = "Input b: "))

result = ptb1.function(a,b)
print(paste(a,"x +",b,"= 0 => ",result))

Input a: 5
Input b: 7
[1] "5 x + 7 = 0 =>  Nghiem: -1.4"
```

Giải phương trình bậc 2

```
In [6]: # giao PT bac 2
ptb2.function <- function(a,b,c){
  result <- ""
  # khi a == 0 => giao PTB1
  if(a==0){
    result <- ptb1.function(b,c)
  }else{
    # giao PTB2
    delta <- b*b - 4*a*c
    if(delta<0){
      result <- "Phuong trinh vo nghiem"
    } else if(delta==0){
      nghiem <- -b/(2*a)
      result <- paste("x1 = x2 = x0 =", nghiem)
    } else{
      x1 <- (-b + sqrt(delta))/(2*a)
      x2 <- (-b - sqrt(delta))/(2*a)
      result <- paste("x1 =", x1, ", x2 =", x2)
    }
  }
  return(result)
}
```

```
In [7]: a <- as.numeric(readline(prompt = "Input a: "))
b <- as.numeric(readline(prompt = "Input b: "))
c <- as.numeric(readline(prompt = "Input c: "))
result = ptb2.function(a,b,c)
print(paste(a,"x^2 +",b,"x +",c, "= 0 => ",result))

Input a: -3
Input b: 2
Input c: 4
[1] "-3 x^2 + 2 x + 4 = 0 => x1 = -0.86851709182133 , x2 = 1.535183758488"
```

Exercise 3: Các bạn tự thực hiện

```
In [8]: # Tao thu vien function de tai su dung khi can
source("ex_functions.R")

In [9]: # Tinh tien dien
so_kw <- as.integer(readline(prompt = "So kw tieu thu: "))

So kw tieu thu: 200

In [11]: tien_dien = tinh_tien_dien.function(200)
print(paste("Tien dien:", tien_dien, "VND"))

[1] "Tien dien: 329450 VND"

In [12]: # Kiem tra so nguyen to
n <- as.integer(readline(prompt = "Nhap n: "))

Nhap n: 7

In [13]: test = kt_soNT.function(n)
print(paste(n, if (test) "is Prime" else "is not Prime"))

[1] "7 is Prime"

In [ ]:
```