Teaching R as a general programming language

Feng Li

Department of Statistics, Stockholm University

Nov, 2012

Introduction and background

- Intended learning outcomes
  - Know the general programming logic
  - Know the idea of object-oriented programming
  - Use R as a daily tool: a better version of calculator/Excel
- Course book
  - Many books available, most focus on implementing statistical methods/graphics in R.
  - **R Cookbook** by Paul Teetor
- Teaching activities
  - Lectures: Materials available at `http://www.mattiasvillani.com/teaching/programming-in-r/`
  - Labs: R on Solaris machine
    We design questions for the students (some examples later).

Course covered topics

- R workspace:
  `ls(), rm(), getwd(), setwd(), save.image(), load(), source()`
- Data structures and manipulation:
  `vector(), matrix(), array(), data.frame(), list()`
  `read.table(), [ ], [[ ]], $`
- Object-oriented programming:
  `class(), typeof(), function(), return()`
- Loops and `if else` condition
- Vectorization: `*apply()`
- Basic graphical tools: `plot()`, `lines()`, `par()`
- Simple debugging tools: `print(), browser(), traceback()`
- Packages, helps and documentations.
- Introduction to statistical tests and linear regression model with R.
- Good programming style:
  `readability (commenting, naming)`, `extendability`

Sample lab questions
↛ Roots for quadratic equation

7 ♡

1. The roots for the quadratic equation $ax^2 + bx + c = 0$ are of the form

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{and} \quad x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

2. Write a function named `quaroot` to solve the roots of given quadratic equation with `a` ,`b` , `c`, as input arguments. [Hint: you may need the `sqrt()` function]

3. Test your function on the following equations

$$\begin{aligned} x^2 + 4x - 1 &= 0 \\ -2x^2 + 2x &= 0 \\ 3x^2 - 9x + 1 &= 0 \\ x^2 - 4 &= 0 \end{aligned}$$

4. Test your function with the equation $5x^2 + 2x + 1 = 0$. What are the results? Why? [Hint: check $b^2 - 4ac$]?

5. Modify your function and return `NA` if $b^2 - 4ac < 0$.

Sample lab questions
$\looparrowright$ Factorial function (1)

2 ♡

The factorial function is formally defined by

$$n! = \prod_{k=1}^{n} k$$

1. Write a function `myFactorial()` with a `for` loop to calculate the factorial for any positive integer input n.
2. Test your function with the following numbers $n = 1$, 10, 100.
3. Write a similar function `myFactorial2()` using a `while` loop instead and test your function with the following numbers 1, 10, 100 to check if you get the same results as before.
4. Write a third factorial function `myFactorial3()` without loops [Hint: use `prod()` function]
5. Replicate `myFactorial(150)` for 5000 times and check how much time takes to run it. [Hint: use `replicate()` and `system.time()`]
6. Repeat **2.5** with `myFactorial2()` and `myFactorial3()`.
7. For the three ways of calculating the factorial, which one do you prefer. Discuss the pro and cons.

Sample lab questions
↛ Factorial function (2)

8. What will you get if you try `myFactorial(500)` ?
9. Note that
$$\log(n!) = \sum_{k=1}^{n} \log(k).$$
Modify your `myFactorial()` and add an extra argument `log` so that when `log = TRUE` the new function returns the logarithm of the factorial and `log = FALSE` return the usual factorial.
10. What do you obtain if you run `myFactorial(500, log = TRUE)`?
11. Write your final version of the factorial function `myFactorialNew()` without loops and with a log argument.

Sample lab questions
↛ Debug a function

5 ♡

1. Jim wanted to write a function to check if a scalar (single number, not a vector) `a` is present as one the elements of a vector `b`, so he wrote a function `isIn()`

   ```
   isIn <- function(a, b) {
     j <- 1
     while (j <= length(b)) {
       if(a == b[j]) {
         out <- TRUE}
       else {
         out <- FALSE}
       j <- j + 1
     }
     return(out)
   }
   ```

   Test the function with `isIn(3, 1:3)` and `isIn(3, 1:5)`. Does the function do what it is supposed to do?

2. Insert `browser()` in the code at the proper location and check what has happened inside the loop. You may also want to try other debugging tools as well [Hint `?debug` for more information on how to work within the debugging environment].

3. Remove the bug with minimal changes of the code.

4. Jim wanted to extend his function allowing `a` to be a vector as well. His purpose was to check which elements in vector `a` are available in `b`. Here is his yet another attempt.

5

Let $m = 4$ and $n = 6$ and answer the following questions

1. Generate a random vector from the uniform distribution of length $m * n$ and name it as `myVec`.
2. Convert the vector into a $m \times n$ matrix and fill the entries of the matrix **by row** and name it as `X`.
3. Generate a diagonal matrix where the diagonal elements are from the sequence `d <- seq(1,n)` and name it as `D`.
4. Calculate the matrix multiplication `X%*%D` and save it as `outDirect`.
5. Create a $m \times n$ matrix named `D2` where each row in the matrix is filled with the vector `d`.
6. Calculate the element-wise multiplication `X*D2` and save it as `outInDirect`.
7. Compare if the two matrices `outDirect` and `outInDirect` are identical.
8. Now let $m = 400$ and $n = 6000$ and redo **5.1-5.6**. Record the execution (running) time for calculating **5.4** and **5.6**.

Thank you!

- Questions?
- Contact: Feng Li, <feng.li@stat.su.se>.