Initiation to R software Session V

Pierre Michel

Master AMSE 1st year, 2019

Loops and tests

Loops and tests

R has instructions that are similar to that of C language.

Main instructions:

- `for (... in ...) {}`: runs code in {} for some values.
- `while (...) {}`: runs code in {} while a condition is true.
- `if (...) {} else {}`: runs code in {} conditionally.

In most cases, loops can be avoided by **vectorization** and the use of **simpler** and less **time-consuming** functions (e.g `apply()`).

If objects are created in a loop, always **define them first**.

Loops and tests: example 1

Aim: create a vector that equals 0 if x equals 3, 1 elsewhere, then add this vector to x.

```
# Try this example !
# Solution 1: Loop
system.time({
  x = sample(1:100, 10, rep = T)
  y = numeric(length(x))
  for(i in 1:length(x)) {
    if (x[i] == 3) y[i] = 0
    else y[i] = 1
  }
})

# Solution 2: Vectorization
system.time({
  x = sample(1:100, 10, rep = T)
})
```

Loops and tests: example 2

Aim: sum on the columns of a matrix.

```r
# Try this example !
# Solution 1: Loop
system.time({
  M = matrix(sample(1:10, 20, rep = T), 5, 4)
  SM = numeric(length(ncol(M)))
  for (j in 1:ncol(M)) {
    SM[j] = sum(M[,j])
  }
})

# Solution 2: Function
system.time({
  M = matrix(sample(1:10, 20, rep = T), 5, 4)
  SM = apply(M, 2, sum)
})
```

R programming

Write a program in R

A program in R is a **sequence of instructions**, saved in a text file (a script in format `.txt` or `.R`).

A program can be run using the function `source()`.

Useful when you want to perform the same task several times.

Write a program in R: example

Aim: plot the same graphic several times, for 3 different species

```r
# Solution 1:
layout(matrix(1:3,3,1))
data(iris)
data = iris[iris$Species == "setosa",]
plot(data$Sepal.Length, data$Sepal.Width,
          xlab = "Sepal Length", ylab = "Sepal Width")
title("setosa")
data = iris[iris$Species == "versicolor",]
plot(data$Sepal.Length, data$Sepal.Width,
          xlab = "Sepal Length", ylab = "Sepal Width")
title("versicolor")
data = iris[iris$Species == "virginica",]
plot(data$Sepal.Length, data$Sepal.Width,
          xlab = "Sepal Length", ylab = "Sepal Width")
title("virginica")
```

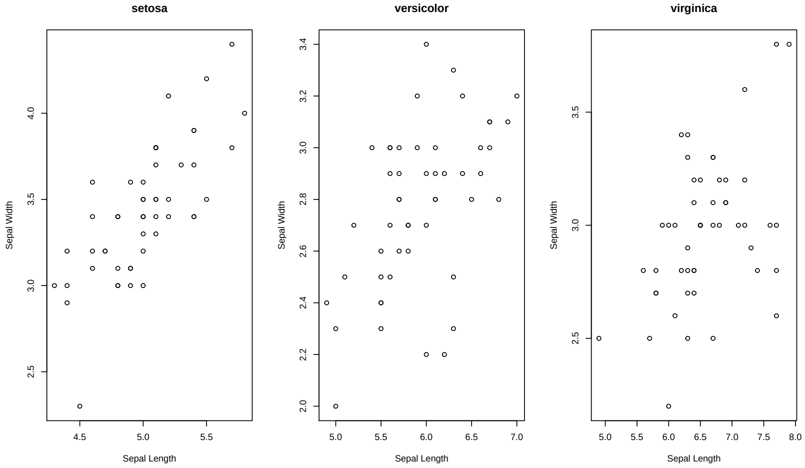
Write a program in R: example

Aim: plot the same graphic several times, for 3 different species

```r
# Solution 2: save the program in file "mySpecies.R"
# The following commands should be saved in a script
layout(matrix(1:3,3,1))
data(iris)
species = unique(iris$Species)
for (i in 1:length(species)) {
  data = iris[ iris$Species == species[i],]
  plot(data$Sepal.Length, data$Sepal.Width,
          xlab = "Sepal Length", ylab = "Sepal Width")
  title(species[i])
}
# Then, run the program saved in a script
source("mySpecies.R")
```

Write a program in R: example

Aim: plot the same graphic several times, for 3 different species



Functions

Functions

To use a program as much as we want, by changing its parameters at will, we can **write a function**. The written functions will have the same properties as those from R packages.

A function is an R program with **variable parameters** given in the `function()` instruction that initiates function writing.

Syntax: `function_name = function(p1, p2, p3, ...) {}`

The instructions of the function are written in `{}`.

Functions

To run, a function must be loaded in memory, this can be done in several ways:

- Type the commands on the keyboard
- Copy/paste them to an R editor
- save the function in a R file and load it with the command `source()`

If you want the function to be loaded when R starts, you can save it in a file with the extension `.Rda`, which will be loaded in memory if located in the working directory (use `getwd()`).

Functions

```r
# Solution 3: write a function saved in "myFun.R"
# The following lines should be saved in the script
myFun = function(specie, data) {
  data = iris[iris$Species == specie,]
  plot(data$Sepal.Length, data$Sepal.Width,
          xlab = "Sepal Length", ylab = "Sepal Width")
  title(specie)
}
# Run the function
layout(matrix(1:3,3,1))
source("myFun.R")
myFun("setosa", iris)
myFun("versicolor", iris)
myFun("virginica", iris)
```

Functions

There are two ways to specify the parameters of a function `fun = function(p1, p2, p3)`:

- by their position: `fun(x,y,z)` is equivalent to `fun(p1=x, p2=y, p3=z)`
- by their name: `fun(p2=y, p1=x, p3=z)`

Default parameters can be defined: `fun = function(p1=1:2, p2=c(T,F), p3=c("1","3"))`

Use `print()` to print the content of an object in a function.

Functions

```r
# Use print
square = function(x) print(x*x)
square(3)
```

```
## [1] 9
```

```r
# Use default parameter
msd = function(x = 1:10) {
  m = mean(x)
  s = sd(x)
  print(c(m,s))
}
msd()
```

```
## [1] 5.50000 3.02765
```

```r
msd(1:100)
```

```
## [1] 50.50000 29.01149
```

Basic functions

Basic mathematical functions

function_name	description
sum(x)	Sum of the elements of x
prod(x)	Product of the elements of x
max(x), min(x)	Maximum, minimum of the elements of
which.max(x)	Returns the index of the maximum of the elements of x
which.min(x)	Returns the index of the minimum of the elements of x
range(x)	Similar to c(min(x), max(x))
mean(x)	Mean of the elements of x
median(x)	Median of the elements of x

Other basic mathematical functions

function_name	description
var(x)	Variance of the elements of x
cov(x)	Covariance matrix if x is a matrix
cor(x)	Correlation matrix of x if x is a matrix or a data.frame
sd(x)	Standard deviation of the elements of x
round(x,n)	Rounds the elements of x to n decimals
rev(x)	Reverse the order of the elements of x
sort(x)	Sort the elements of x in ascending order
rank(x)	Rank the elements of x
scale(x)	Scale (center and reduce) x

. . . and other basic mathematical functions. . .

function_name	description
pmin(x,y,. . .)	A vector whose element i is the minimum of x[i] and y[i]
cumsum(x)	A vector whose element i is the sum of x[i] and y[i]
cumprod()	Similar as cumsum() with product
cummin()	Similar as cumsum() with minimum
cummax()	Similar as cumsum() with maximum
match(x,y)	Vector of same length as x with elements of x that are in y
choose(x,k)	Binomial coefficient
na.omit()	Remove observations with missing values
na.fail()	Returns an error message if x contains at least one NA

. . . and others

function_name	description
unique()	Returns a similar object without duplicates
table()	Returns table of counts of the values of x
subset()	Returns a selection of x based on criteria
sample(x,n)	Random sampling of x of size n, without replacement