Programação em R

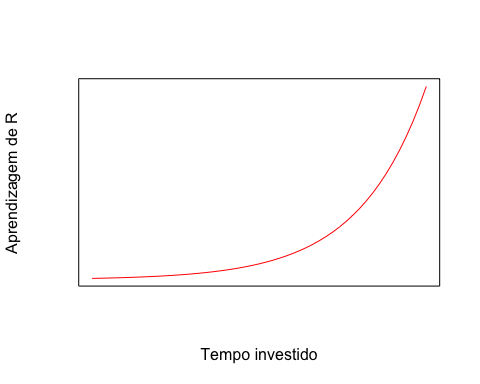
Tutorial 1: alguns pontos importantes a considerar

Samuel Carleial

December 2019

# Introdução

R é uma linguagem de programação muito flexível e de fácil aprendizagem para leigos em computação. Talvez inicialmente a aprendizagem de R para aqueles que começam do zero é muito lenta. Porém, gradualmente se aprende R com mais rapidez e facilidade. Portanto, se diz que a linha de aprendizagem em R é exponencial (veja a figura abaixo).



R é uma linguagem orientada a objetos. Ou seja, nela o foco se dá na criação e manipulação (obviamente) de **objetos** que são carregados na sessão atual do R, consumindo a memória corrente do computador (RAM).

# Exemplo de objeto em R  
# objeto\_x, um vetor composto por números inteiros  
objeto\_x <- c(1,2,3,4,5)  
objeto\_x

## [1] 1 2 3 4 5

Neste tutorial, serão descritos alguns pontos importantes e cruciais da linguagem R, que podem ser de extrema importância na análise e manipulação de dados.

## Classes

Todo e qualquer objeto em R deve pertencer a uma classe, que por sua vez possui diferentes propriedades. As classes mais comuns usada numa análise de dados são os data.frame ou data.table (para conjuntos de dados) ou vetores (para variáveis singulares). Existem em R muitas outras classes e tipos de objetos, como matrizes, arranjos, objetos provenientes de modelos estatísticos, etc.

# Exemplo de diferentes objetos que podem existir em R  
dados <- data.frame("X1" = rnorm(20), "X2" = "controle")  
matriz <- matrix(data = 1:20, ncol = 5)  
variavel <- objeto\_x  
  
paste("dados:", class(dados))

## [1] "dados: data.frame"

paste("matriz:", class(matriz))

## [1] "matriz: matrix"

paste("variavel:", class(variavel))

## [1] "variavel: numeric"

## Vetores

Especificamente, vetores ou variáveis singulares podem assumir diferentes tipos, como números (contínuos ou integrais), caracteres, fatores, datas, etc. Para definir um vetor, usa-se a função c() onde cada elemento do vetor deve ser separado por vírgulas. Aliás, operações usando vetores geralmente são muito fáceis de realizar, como soma, multiplicação, etc. Veja abaixo:

# números contínuos  
class(c(0.4, 1.2, 3.7))

## [1] "numeric"

# números integrais  
class(as.integer(c(objeto\_x)))

## [1] "integer"

# caracteres  
class(c("a", "b", "c"))

## [1] "character"

# fatores  
class(as.factor(c("homem", "mulher", "homem")))

## [1] "factor"

# datas  
class(as.Date(c("2019-10-02", "2019-07-22", "2018-11-02")))

## [1] "Date"

# operação usando um vetor  
print(objeto\_x \* 10)

## [1] 10 20 30 40 50

print(objeto\_x + objeto\_x)

## [1] 2 4 6 8 10

Adicionalmente, modelos lineares podem conter muitas informações diferentes dentro de um mesmo objeto. Portanto, objetos provenientes de modelos lineares (ex: lme), contém dentre outras informações, valores de p, coeficientes dos preditores, formula do modelo, parâmetros da formula usada, etc. Para isso, esses objetos são constituídos de diferentes elementos, que podem ser inspecionados usando diferentes maneiras.

# Exemplo de modelo e sua classe  
library("nlme")  
modelo <- lme(distance ~ age, data = Orthodont)  
class(modelo)

## [1] "lme"

# Estrutura do modelo. O que existe dentro do objeto modelo ?   
names(modelo)

## [1] "modelStruct" "dims" "contrasts" "coefficients"  
## [5] "varFix" "sigma" "apVar" "logLik"   
## [9] "numIter" "groups" "call" "terms"   
## [13] "method" "fitted" "residuals" "fixDF"   
## [17] "na.action" "data"

## Indexamento

Uma característica fundamental em R (mas também em outras linguagens de programação) é poder acessar valores usando indexamento. Ou seja, usar a posição dentro do objeto (index, linha ou coluna) para poder realizar diferentes ações nos dados.

Em vetores (1 dimensão), é possível acessar os valores usando index (1:length(x)).

variavel[1] # primeiro elemento

## [1] 1

variavel[5] # último elemento

## [1] 5

Em tabelas / matrizes, é possível fazer o mesmo através de colunas e linhas.

dados[,]

## X1 X2  
## 1 1.13174842 controle  
## 2 -0.28930671 controle  
## 3 -2.17453295 controle  
## 4 -1.17646740 controle  
## 5 0.73284906 controle  
## 6 -2.32373737 controle  
## 7 -0.54993220 controle  
## 8 -0.40450706 controle  
## 9 -0.55569913 controle  
## 10 -0.04634708 controle  
## 11 0.79983704 controle  
## 12 -0.58079837 controle  
## 13 0.80050413 controle  
## 14 -1.36974344 controle  
## 15 -0.27888671 controle  
## 16 0.36543955 controle  
## 17 0.40058775 controle  
## 18 -0.02679106 controle  
## 19 0.03443341 controle  
## 20 -1.67841139 controle

dados[1,1] # elemento da primeira linha e coluna

## [1] 1.131748