Programação em R

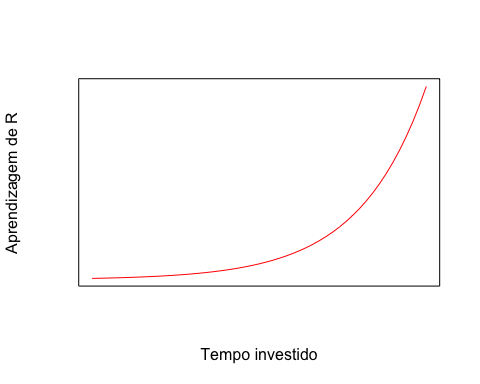
Tutorial 1: relembrando o básico (pontos importantes)

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 de… **objetos** (obviamente) 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(15,7,20,4,12)  
objeto\_x

## [1] 15 7 20 4 12

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 usadas numa análise de dados são vetores (variáveis singulares) e os (conjuntos de dados) data.frame ou data.table. 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

Vetores (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] 150 70 200 40 120

print(objeto\_x + objeto\_x)

## [1] 30 14 40 8 24

## Modelos estatísticos

### objetos com um conjunto de variáveis

Modelos lineares produzidos por funções em R contém diferentes informações que podem ser armazenados num único objeto. Objetos provenientes de modelos lineares (ex: lme), contém dentre outras informações, valores de p, coeficientes dos preditores, fórmula do modelo, parâmetros da fórmula usada, etc. Tais objetos 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"

## Indexação

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

Em vetores (1 dimensão), é possível acessar os valores usando index ou a ordem dos elementos (1:length(x)).

variavel # objeto

## [1] 15 7 20 4 12

variavel[1] # primeiro elemento do objeto

## [1] 15

variavel[5] # último elemento do objeto

## [1] 12

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

dados[,]

## X1 X2  
## 1 0.4536806 controle  
## 2 -1.0587902 controle  
## 3 -1.4361511 controle  
## 4 -0.2317647 controle  
## 5 -0.1721807 controle  
## 6 -0.7616180 controle  
## 7 0.5438237 controle  
## 8 -2.1762361 controle  
## 9 1.1188320 controle  
## 10 -0.4171862 controle  
## 11 0.6523894 controle  
## 12 -1.6973639 controle  
## 13 -0.5918125 controle  
## 14 0.7577777 controle  
## 15 -1.5020551 controle  
## 16 0.8817215 controle  
## 17 0.3800739 controle  
## 18 -1.0544358 controle  
## 19 -2.1990625 controle  
## 20 0.7649665 controle

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

## [1] 0.4536806