

Hands-on R - Tipos básicos de objetos no R

Rodrigo Heldt - rodrigoheldt@gmail.com

IMED - HOPEAD - Nov 13, 2017

Tipos básicos de objetos no R

1. Vetor

Vetor é um conjunto de dados unidimensional.

```
## Vetores podem ser criados utilizando a funcao c(), que concatena determinados valores
# Criando um vetor com dados numericos
x <- 2
vetor_numerico <- c(0.5, 0.6, x)
vetor_numerico
```

```
## [1] 0.5 0.6 2.0
```

```
# Criando um vetor com caracteres (strings)
vetor_caracteres <- c("com", "para", "de")
vetor_caracteres
```

```
## [1] "com" "para" "de"
```

```
## Vetores podem ser criados utilizando direto a criacao intervalos entre dois valores
# Criando um vetor com uma sequencia de número
v <- 9:29
v
```

```
## [1] 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
```

```
## Buscar o valor de uma determinada posicao de um vetor
# buscar o valor que esta na terceira posicao do vetor
v[3]
```

```
## [1] 11
```

```
## Identificando que um objeto é da classe vector
is.vector(v)
```

```
## [1] TRUE
```

2. Matriz

Matriz é um conjunto de dados com duas dimensões: linhas e colunas. Todas as colunas precisam ser de apenas uma classe (todas com número, todas com texto, etc.)

```
## Matrizes podem ser criadas utilizando a funcao matrix.
# Por padrao, a funcao matrix organiza o valores coluna por coluna, começando pelo primeiro
# elemento da primeira coluna
m <- matrix(1:6, nrow = 3, ncol = 2)
m
```

```
##      [,1] [,2]
## [1,]    1    4
## [2,]    2    5
## [3,]    3    6

## Buscar o valor de uma determinada posicao de uma matriz
# buscar o valor que esta na segunda linha e terceira coluna
m[2, 1]

## [1] 2

# buscar todos os valores da segunda linha.
m[2, ]

## [1] 2 5

# buscar todos os valores da terceira coluna.
m[, 1]

## [1] 1 2 3

## Utilizar a funcao 'dim' para saber as dimensoes de uma matriz
## (quantidade de linhas e quantidade de colunas)
dim(m)

## [1] 3 2

## Identificando que um objeto é da classe matrix
is.matrix(m)

## [1] TRUE
```

3. Dataframe

Dataframe é um conjunto de dados com duas dimensões: linhas e colunas. Diferenciam-se das matrizes, pois são tabelas nas quais cada coluna representa uma variável que pode ter classe distinta das outras colunas.

```
## Criar data frames utilizando a funcao data.frame, que junta, por coluna, diferentes objetos.
num = 1:4
pais = c("Brasil", "Brasil", "Espanha", "EUA")

d <- data.frame(num, pais)
d

##      num    pais
## 1     1  Brasil
## 2     2  Brasil
## 3     3 Espanha
## 4     4    EUA

## Buscar o valor de uma determinada posicao de um dataframe
# buscar o valor que esta na segunda linha e segunda coluna
d[2, 2]

## [1] Brasil
## Levels: Brasil Espanha EUA
```

```

# buscar todos os valores da segunda linha
d[2, ]

##      num      pais
## 2      2 Brasil

# buscar todos os valores da primeira coluna
d[, 1]

## [1] 1 2 3 4

# buscar todos os valores da coluna chamada "num". Equivalente a 'd[,1]'
d[, "num"]

## [1] 1 2 3 4

# buscar todos os valores da coluna chamada "num" usando $. Equivalente a 'd[,1]'
d$num

## [1] 1 2 3 4

## Utilizar a funcao 'dim' para saber as dimensoes de um dataframe (quantidade de linhas e
## quantidade de colunas)
dim(d)

## [1] 4 2

## verificar as primeiras linhas do data.frame para entender quais sao os dados existentes
head(d)

##      num      pais
## 1      1 Brasil
## 2      2 Brasil
## 3      3 Espanha
## 4      4      EUA

## Apresenta o nome de cada coluna (variavel) do data frame
names(d)

## [1] "num" "pais"

## Identificando que um objeto é da classe data.frame
is.data.frame(d)

## [1] TRUE

```