

Curso introdução ao R - Atividades

Sollano R. Braga

5 de dezembro de 2016

Contents

| | |
|-------------|---|
| Atividade 1 | 1 |
| Atividade 2 | 2 |
| Atividade 3 | 5 |

Atividade 1

1. Qual a diferença entre vetores e matrizes?

Vetores possuem apenas uma dimensão, matrizes possuem duas (linha e coluna).

2. Qual a diferença entre matrizes e dataframes?

Matrizes só podem ter uma classe de dado. Dataframes podem ter colunas com diferentes classes.

3. Crie um vetor com numeros de 1:10 e salve-o com o nome x. Visualise x no console.

```
x <- c(1:10)
x
```

```
## [1] 1 2 3 4 5 6 7 8 9 10
```

4. Crie um dataframe chamado d, composto por 3 vetores de tamanho 6, sendo um de classe numérica, um de classe character, e um de classe lógica. Visualize-o no console.

```
w <- rnorm(6)
w
```

```
## [1] 0.6910233 1.2505577 -0.9118575 -0.8883542 -1.2591713 1.1256158
```

```
x <- c("a", "b", "c", "d", "e", "f")
x
```

```
## [1] "a" "b" "c" "d" "e" "f"
```

```
y <- c(T, F, T, F, T, F)
y
```

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

```
d <- data.frame(w, x, y)
# ou
```

```
d <- data.frame(w = rnorm(6),
                x = c("a", "b", "c", "d", "e", "f") ,
                y = c(T, F, T, F, T, F) )
d
```

```
##           w x      y
## 1 -0.28283717 a  TRUE
## 2  0.05489298 b FALSE
```

```
## 3 0.23736791 c TRUE
## 4 1.45974560 d FALSE
## 5 0.58835426 e TRUE
## 6 -1.18592292 f FALSE
```

5. Renomeie as variáveis do dataframe criado anteriormente para Numeros, Palavras e Teste.

```
names(d) <- c("Numeros", "Palavras", "Teste")
names(d)
```

```
## [1] "Numeros" "Palavras" "Teste"
```

6. Abra o arquivo dados.xlsx no excel, salve-o como dados.csv, e importe-o no R.

```
# Arquivo - Salvar como - csv - Comma Separated Values

dados <- read.csv2("dados.csv")
```

7. Utilizando o dataframe iris, calcule a média do comprimento das sépalas (Sepal.Length):

```
mean(iris$Sepal.Length)
```

```
## [1] 5.843333
```

Atividade 2

Nesta atividade será utilizado o dataset women, que contém informações de altura e peso de mulheres, em polegadas e libras, respectivamente. Este dado já vem built-in no R, então não é necessário carregá-lo.

1. Crie uma cópia do dataset women, chamada women_mod.

```
women_mod <- women

women_mod
```

```
##      height weight
## 1       58     115
## 2       59     117
## 3       60     120
## 4       61     123
## 5       62     126
## 6       63     129
## 7       64     132
## 8       65     135
## 9       66     139
## 10      67     142
## 11      68     146
## 12      69     150
## 13      70     154
## 14      71     159
## 15      72     164
```

2. Sabendo que uma polegada equivale a 0.0254 metros, crie uma nova variável no dataframe women_mod chamada altura_metros, que converte a variável height para metros, e arredonde-a para uma casa decimal.

```
women_mod$altura_metros <- women_mod$height * 0.0254
women_mod$altura_metros <- round(women_mod$altura_metros , 1)
women_mod
```

```
##      height weight altura_metros
## 1      58    115          1.5
## 2      59    117          1.5
## 3      60    120          1.5
## 4      61    123          1.5
## 5      62    126          1.6
## 6      63    129          1.6
## 7      64    132          1.6
## 8      65    135          1.7
## 9      66    139          1.7
## 10     67    142          1.7
## 11     68    146          1.7
## 12     69    150          1.8
## 13     70    154          1.8
## 14     71    159          1.8
## 15     72    164          1.8
```

3. Sabendo que uma libra equivale a 0.453592 kilos, crie uma nova variável no dataframe `women_mod` chamada `peso_kilos`, que converte a variável `weight` para kilos, e arredonde-a para uma casa decimal.

```
women_mod$peso_kilos <- women_mod$weight * 0.453592
women_mod$peso_kilos <- round(women_mod$peso_kilos, 1)
women_mod
```

```
##      height weight altura_metros peso_kilos
## 1      58    115          1.5      52.2
## 2      59    117          1.5      53.1
## 3      60    120          1.5      54.4
## 4      61    123          1.5      55.8
## 5      62    126          1.6      57.2
## 6      63    129          1.6      58.5
## 7      64    132          1.6      59.9
## 8      65    135          1.7      61.2
## 9      66    139          1.7      63.0
## 10     67    142          1.7      64.4
## 11     68    146          1.7      66.2
## 12     69    150          1.8      68.0
## 13     70    154          1.8      69.9
## 14     71    159          1.8      72.1
## 15     72    164          1.8      74.4
```

4. Delete as variáveis `height` e `weight` do dataframe `women_mod`.

```
women_mod$height <- NULL
women_mod$weight <- NULL

# ou

women_mod[,c("height", "weight")] <- NULL
women_mod
```

```
##      altura_metros peso_kilos
## 1          1.5      52.2
## 2          1.5      53.1
## 3          1.5      54.4
```

```
## 4      1.5      55.8
## 5      1.6      57.2
## 6      1.6      58.5
## 7      1.6      59.9
## 8      1.7      61.2
## 9      1.7      63.0
## 10     1.7      64.4
## 11     1.7      66.2
## 12     1.8      68.0
## 13     1.8      69.9
## 14     1.8      72.1
## 15     1.8      74.4
```

5. Crie um novo objeto chamado `altura_maior_que_1.60`, que contém os dados de mulheres que tem altura maior que 1.60 metros.

```
altura_maior_que_1.60 <- women_mod[women_mod$altura_metros > 1.60, ]
altura_maior_que_1.60
```

```
##      altura_metros peso_kilos
## 8      1.7      61.2
## 9      1.7      63.0
## 10     1.7      64.4
## 11     1.7      66.2
## 12     1.8      68.0
## 13     1.8      69.9
## 14     1.8      72.1
## 15     1.8      74.4
```

6. Crie um novo objeto chamado `altura_e_peso`, que contém os dados de mulheres que tem altura maior do que 1.50 E peso maior ou igual do que 60 kg.

```
altura_e_peso <- women_mod[women_mod$altura_metros > 1.50 & women_mod$peso_kilos >= 60, ]
altura_e_peso
```

```
##      altura_metros peso_kilos
## 8      1.7      61.2
## 9      1.7      63.0
## 10     1.7      64.4
## 11     1.7      66.2
## 12     1.8      68.0
## 13     1.8      69.9
## 14     1.8      72.1
## 15     1.8      74.4
```

7. Crie um novo objeto chamado `altura_ou_peso`, que contém os dados de mulheres que tem altura maior do que 1.50 OU peso menor ou igual do que 60 kg.

```
altura_ou_peso <- women_mod[women_mod$altura_metros > 1.60 | women_mod$peso_kilos <= 60, ]
altura_ou_peso
```

```
##      altura_metros peso_kilos
## 1      1.5      52.2
## 2      1.5      53.1
## 3      1.5      54.4
## 4      1.5      55.8
## 5      1.6      57.2
## 6      1.6      58.5
```

```
## 7      1.6      59.9
## 8      1.7      61.2
## 9      1.7      63.0
## 10     1.7      64.4
## 11     1.7      66.2
## 12     1.8      68.0
## 13     1.8      69.9
## 14     1.8      72.1
## 15     1.8      74.4
```

8. Crie um novo objeto chamado `altura_entre`, que contém os dados de mulheres que tem altura entre 1.50 e 1.60, utilizando o operador.

```
altura_entre <- women_mod[women_mod$altura_metros %in% seq(1.50, 1.60, by=0.01), ]
altura_entre
```

```
##   altura_metros peso_kilos
## 1          1.5      52.2
## 2          1.5      53.1
## 3          1.5      54.4
## 4          1.5      55.8
## 5          1.6      57.2
## 6          1.6      58.5
## 7          1.6      59.9
```

Atividade 3

Será utilizado o dado “dados.csv” nesta atividade:

```
dados <- read.csv2("dados.csv")
head(dados)
```

```
##   CODTALHAO CODPARCELA CODARVORE  CAP ALT1 DESCATEGORIA
## 1      3656         101         101 51.2 21.5      Normal
## 2      3656         101         103 52.2 20.9      Normal
## 3      3656         101         201 22.6  NA Outros Fustes
## 4      3656         101         203 56.0 21.7      Normal
## 5      3656         101         205 45.2 20.5      Normal
## 6      3656         101         302 58.4 21.6      Dominante
```

Calcule o diâmetro a altura do peito.

```
dados$DAP <- dados$CAP / pi
```

ou

```
dados["DAP"] <- dados$CAP / pi
```

```
head(dados)
```

```
##   CODTALHAO CODPARCELA CODARVORE  CAP ALT1 DESCATEGORIA      DAP
## 1      3656         101         101 51.2 21.5      Normal 16.297466
## 2      3656         101         103 52.2 20.9      Normal 16.615776
## 3      3656         101         201 22.6  NA Outros Fustes  7.193803
## 4      3656         101         203 56.0 21.7      Normal 17.825354
## 5      3656         101         205 45.2 20.5      Normal 14.387607
## 6      3656         101         302 58.4 21.6      Dominante 18.589297
```

Sabendo que o código a seguir estima o volume para as árvores medidas:

```
dados$VOL <- exp(-9.7091 + 1.8828 * log(dados$DAP) + 0.9449* log(dados$ALT1) )
```

```
head(dados)
```

```
##   CODTALHAO CODPARCELA CODARVORE  CAP ALT1 DESCATEGORIA      DAP
## 1      3656         101         101 51.2 21.5      Normal 16.297466
## 2      3656         101         103 52.2 20.9      Normal 16.615776
## 3      3656         101         201 22.6  NA Outros Fustes  7.193803
## 4      3656         101         203 56.0 21.7      Normal 17.825354
## 5      3656         101         205 45.2 20.5      Normal 14.387607
## 6      3656         101         302 58.4 21.6    Dominante 18.589297
##           VOL
## 1 0.2111506
## 2 0.2132033
## 3      NA
## 4 0.2521546
## 5 0.1596352
## 6 0.2716972
```

Crie uma função que gere um vetor chamado VOL, que estime o volume, dado as informações de b0, b1, b2, dap e ht.

Estime o volume para o dataframe dados utilizando a função.

```
calcula.vol <- function(dap, ht, b0=-9.7091, b1=1.8828, b2=0.9449){
```

```
  VOL <- exp(b0 + b1*log(dap) + b2*log(ht) )
```

```
  return(VOL)
```

```
}
```

```
dados$VOL <- calcula.vol(dados$DAP, dados$ALT1)
```

```
head(dados)
```

```
##   CODTALHAO CODPARCELA CODARVORE  CAP ALT1 DESCATEGORIA      DAP
## 1      3656         101         101 51.2 21.5      Normal 16.297466
## 2      3656         101         103 52.2 20.9      Normal 16.615776
## 3      3656         101         201 22.6  NA Outros Fustes  7.193803
## 4      3656         101         203 56.0 21.7      Normal 17.825354
## 5      3656         101         205 45.2 20.5      Normal 14.387607
## 6      3656         101         302 58.4 21.6    Dominante 18.589297
##           VOL
## 1 0.2111506
## 2 0.2132033
## 3      NA
## 4 0.2521546
## 5 0.1596352
## 6 0.2716972
```