

Introdução ao programa R

Ronald Targino, DEMA-UFC

Notas de aula

3. Importação e exportação de arquivos de dados

3.1. Importação

```
# Funções para importação (leitura) de arquivos
# planilhas
read.csv( file, header = TRUE, sep = ",", dec = ".", ...)
# planilhas
read.csv2( file, header = TRUE, sep = ";", dec = ".", ...)
# textos e planilhas
read.table( file, header = FALSE, sep = "", dec = ".", ...)

# Argumentos das funções
# file: nome do arquivo; ou caminho e nome do arquivo
# header: informa a presença (TRUE) ou ausência (FALSE) de nome de colunas(variáveis)
# sep: caracter usado como separador de registros (nas linhas)
# dec: caracter usado como ponto decimal

# verificar diretório de trabalho; os arquivos de dados devem estar nesse diretório
getwd()

# Salve os quatro arquivos de dados (dados1.csv, dados2.txt, dados3.txt, dados4.txt)
# no seu diretório de trabalho. Abra-os e verifique os caracteres usados
# como separador de registros e ponto decimal.

# Faça a leitura dos quatro arquivos de dados:
rm(list = ls()) # remover todos os objetos

d1 = read.table("dados1.csv", header = TRUE, sep = ",")
d1

d2 = read.table("dados2.txt", header = TRUE, sep = ";", dec = ".")
d2

d3 = read.table("dados3.txt", header = TRUE, dec = ",")
d3

d4 = read.table("dados4.txt", header = TRUE)
d4

# Observações:
# a função read.table tem como padrão (empregado caso não seja informado): sep = ""
# (espaço em branco ou tabulação) e dec = "." (ponto decimal).
# caso os arquivos não estejam no diretório de trabalho, informe o caminho. Exemplo:
# d5 = read.table("C:/Users/Ronald/dados4.txt", header = TRUE)
```

```

# Explorar o objeto d4
d4 = read.table("dados4.txt", header = TRUE)
d4

##      ident estudante sexo altura
## 1         1        ana   f    170
## 2         2        bela   f    160
## 3         3        carla   f    150
## 4         4        denis   m    155
## 5         5        elena   f    160
## 6         6        fabio   m    158
## 7         7         gabi   f    172
## 8         8         jorge   m    179
## 9         9         luis   m    168
## 10        10        maria  f    165

is(d4) # é um data frame!

## [1] "data.frame" "list"          "oldClass"     "vector"

str(d4) # estrutura do data frame d4

## 'data.frame':   10 obs. of  4 variables:
## $ ident      : int  1 2 3 4 5 6 7 8 9 10
## $ estudante: Factor w/ 10 levels "ana","bela","carla",...: 1 2 3 4 5 6 7 8 9 10
## $ sexo       : Factor w/ 2 levels "f","m": 1 1 1 2 1 2 1 2 2 1
## $ altura     : int  170 160 150 155 160 158 172 179 168 165

attach(d4) # cada coluna (variável) de d4 é um vetor que pode ser acessado pelo nome da variável
sexo

## [1] f f f m f m f m m f
## Levels: f m

table(sexo) # distribuição de frequência da variável sexo

## sexo
## f m
## 6 4

length(sexo) # número de observações na variável sexo

## [1] 10

summary(altura) # mínimo, 1o. quartil, mediana, média, 3o. quartil, máximo

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 150.0   158.5   162.5   163.7   169.5   179.0

mean(altura) # altura média

## [1] 163.7

median(altura) # altura mediana

## [1] 162.5

min(altura); max(altura) # menor e maior alturas

## [1] 150

```

```
## [1] 179
var(altura); sd(altura) # variância e desvio padrão da altura

## [1] 76.23333
## [1] 8.73117
altura[estudante == 'luis'] # altura do estudante luis

## [1] 168
names(d4) # variáveis do data frame d4

## [1] "ident"      "estudante" "sexo"      "altura"
which(altura < 155) # identifica as posições dos registros(alunos) de altura inferior a 155

## [1] 3
which(altura >= 165) # identifica as posições dos registros(alunos) de altura maior ou igual a 165

## [1] 1 7 8 9 10
# data frame M1 contendo apenas estudantes do sexo feminino
M1 = subset(d4, sexo == 'f')
M1

##      ident estudante sexo altura
## 1         1      ana    f     170
## 2         2      bela    f     160
## 3         3      carla    f     150
## 5         5      elena    f     160
## 7         7       gabi    f     172
## 10        10     maria    f     165

is(M1)

## [1] "data.frame" "list"      "oldClass"   "vector"
# data frame M2 contendo apenas estudantes do sexo feminino e altura maior que 165
M2 = subset(d4, (sexo == 'f' & altura > 165))
M2

##      ident estudante sexo altura
## 1         1      ana    f     170
## 7         7       gabi    f     172
# data frame M3 contendo apenas as variáveis sexo e altura
M3 = subset(d4, select = c(sexo, altura))
M3

##      sexo altura
## 1      f     170
## 2      f     160
## 3      f     150
## 4      m     155
## 5      f     160
## 6      m     158
## 7      f     172
## 8      m     179
## 9      m     168
```

```
## 10    f    165
```

```
# data frame M4 contendo apenas as variáveis identificação e altura dos  
# estudantes do sexo masculino
```

```
M4 = subset(d4, sexo == 'm', select = c(ident,altura))
```

```
M4
```

```
##   ident altura  
## 4      4    155  
## 6      6    158  
## 8      8    179  
## 9      9    168
```

```
# acrescentando as notas dos estudantes no data frame d4
```

```
nota1 = c(10, 9, 5, 7, 4, 6, 5, 10, 9, 3)
```

```
nota2 = c(9, 10, 6, 6, 5, 5, 4, 10, 8, 6)
```

```
detach(d4)
```

```
d4 = cbind(d4, nota1, nota2)
```

```
rm(nota1, nota2)
```

```
attach(d4)
```

```
d4
```

```
##   ident estudante sexo altura nota1 nota2  
## 1      1      ana   f    170     10     9  
## 2      2      bela   f    160     9     10  
## 3      3      carla   f    150     5     6  
## 4      4      denis   m    155     7     6  
## 5      5      elena   f    160     4     5  
## 6      6      fabio   m    158     6     5  
## 7      7      gabi   f    172     5     4  
## 8      8      jorge   m    179    10    10  
## 9      9      luis   m    168     9     8  
## 10    10      maria   f    165     3     6
```

```
d4
```

```
##   ident estudante sexo altura nota1 nota2  
## 1      1      ana   f    170     10     9  
## 2      2      bela   f    160     9     10  
## 3      3      carla   f    150     5     6  
## 4      4      denis   m    155     7     6  
## 5      5      elena   f    160     4     5  
## 6      6      fabio   m    158     6     5  
## 7      7      gabi   f    172     5     4  
## 8      8      jorge   m    179    10    10  
## 9      9      luis   m    168     9     8  
## 10    10      maria   f    165     3     6
```

```
# data frame M5 contendo apenas os estudantes com nota1 maior que 8 e nota2 maior que 7
```

```
M5 = subset(d4, (nota1 > 8 & nota2 >7))
```

```
M5
```

```
##   ident estudante sexo altura nota1 nota2  
## 1      1      ana   f    170     10     9  
## 2      2      bela   f    160     9     10  
## 8      8      jorge   m    179    10    10  
## 9      9      luis   m    168     9     8
```

```
# data frame M6: eliminando as variáveis sexo e altura
M6 = subset(d4, (nota1 > 8 & nota2 > 7), select = -c(sexo, altura))
M6
```

```
##   ident estudante nota1 nota2
## 1     1      ana    10     9
## 2     2      bela     9    10
## 8     8     jorge    10    10
## 9     9      luis     9     8
```

```
# data frame M7: idêntico ao M6
M7 = subset(M5, select = -c(sexo, altura))
M7
```

```
##   ident estudante nota1 nota2
## 1     1      ana    10     9
## 2     2      bela     9    10
## 8     8     jorge    10    10
## 9     9      luis     9     8
```

```
# Outros exemplos
M8 = subset(d4, nota2 > nota1, select = -c(sexo, altura))
M8
```

```
##   ident estudante nota1 nota2
## 2     2      bela     9    10
## 3     3     carla     5     6
## 5     5     elena     4     5
## 10    10    maria     3     6
```

```
M9 = subset(d4, (nota1 > 8 | nota2 > 8), select = -c(sexo, altura))
M9
```

```
##   ident estudante nota1 nota2
## 1     1      ana    10     9
## 2     2      bela     9    10
## 8     8     jorge    10    10
## 9     9      luis     9     8
```

```
# Operadores Lógicos
# & e &&: E
# | e ||: OU
```

```
a = 0; b = 1; d = 5; e = 10
a > b
```

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

```
d > e
```

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

```
(a > b) && (d > e)
```

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

```
a < b
```

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

```

d > e
## [1] FALSE
(a < b) && (d > e)
## [1] FALSE
a > b; d < e
## [1] FALSE
## [1] TRUE
(a > b) && (d < e)
## [1] FALSE
a < b; d < e
## [1] TRUE
## [1] TRUE
(a < b) && (d < e)
## [1] TRUE
a > b; d > e
## [1] FALSE
## [1] FALSE
(a > b) || (d > e)
## [1] FALSE
a < b; d > e
## [1] TRUE
## [1] FALSE
(a < b) || (d > e)
## [1] TRUE
a > b; d < e
## [1] FALSE
## [1] TRUE
(a > b) || (d < e)
## [1] TRUE
a < b; d < e
## [1] TRUE
## [1] TRUE
(a < b) || (d < e)
## [1] TRUE

```

```

a = c(10,9,8,7)
b = c(9,10,8,6)
a > 8 & b > 9 # todos os elementos do vetor são comparados

## [1] FALSE TRUE FALSE FALSE
a > 8 | b > 9 # todos os elementos do vetor são comparados

## [1] TRUE TRUE FALSE FALSE
a > 8 && b > 9 # apenas o primeiro elemento do vetor é comparado

## [1] FALSE
a > 8 || b > 9 # apenas o primeiro elemento do vetor é comparado

## [1] TRUE
a = c(7,9,8,7)
b = c(5,10,8,6)
a > 8 & b > 9 # todos os elementos do vetor são comparados

## [1] FALSE TRUE FALSE FALSE
a > 8 | b > 9 # todos os elementos do vetor são comparados

## [1] FALSE TRUE FALSE FALSE
a > 8 && b > 9 # apenas o primeiro elemento do vetor é comparado

## [1] FALSE
a > 8 || b > 9 # apenas o primeiro elemento do vetor é comparado

## [1] FALSE

```

3.2. Exportação

```

# Caso não seja informado o caminho, os arquivos serão salvos no diretório de trabalho
write.table(d4, file = "dados4a.csv", sep = ";", dec = ".", row.names = FALSE)
write.table(d4, file = "dados4b.txt", sep = " ", dec = ",", row.names = FALSE)

# write.csv usa "." para ponto decimal e "," para separar registros
# write.csv2 usa "," para ponto decimal e ";" para separar registros

# Exercício: Pesquisar como importar e exportar arquivos excel (extensões xls exlsx).

```