

1



2

Relembrando: Entrada e Saída

- Arquivo "comandos.R"

```
print(10 + 20)  
print("oi mundo")
```

- No Console, pode-se carregar comandos com `source()`:

```
source("comandos.R")
```

- Os comandos são carregados e executados

Relembrando: Entrada e Saída

- Pode-se redirecionar a saída para um arquivo: `sink()`

```
sink("saida.txt")  
print("oi mundo!!")  
sink()
```

- Gera o arquivo `saida.txt`

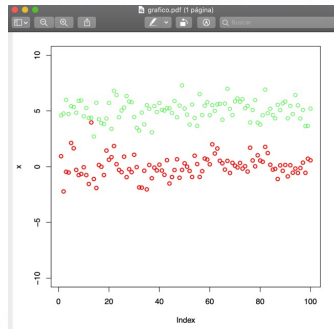


Relembrando: Entrada e Saída

- Outros tipos de saídas, terminados com o comando `dev.off()`

```
pdf("grafico.pdf")
x = rnorm(100)
y = rnorm(100,5,1)
plot(x, lty=2, lwd=2, col="red", ylim=c(-10,10))
points(y, lty=3, col="green")
dev.off()
```

- Gera o arquivo `grafico.pdf`



Prof. Dr. Razer A N R Montañó

SEPT/UFPR

5

5

Relembrando: Entrada e Saída.

- Outros tipos de saídas, terminadas com `dev.off()`

```
png("arquivo.png")
...
dev.off()
```

```
jpeg("arquivo.jpg")
...
dev.off()
```

Prof. Dr. Razer A N R Montañó

SEPT/UFPR

6

6

Leitura/Escreita de Arquivos

- Leitura de vários formatos
- R tem Pacotes para os vários tipos de arquivos e formatos
- Usa o diretório de trabalho para leitura e escrita
- Comandos
 - `setwd()` : seta o diretório de trabalho

```
> setwd("/Users/razer/Downloads")
```

- `getwd()` : obtém o diretório de trabalho

```
> getwd()
[1] "/Users/razer/Downloads"
```

Prof. Dr. Razer A N R Montaña

SEPT/UFR

7

7

Leitura/Escreita de Arquivos

Formato	Pacote	Função
Arquivos		<code>read.table()</code> # arquivos delimitados <code>read.delim()</code> # decimais com . <code>read.delim2()</code> # decimais com ,
CSV		<code>read.csv()</code> # separador , <code>read.csv2()</code> # separador ;
XLS/XLSX	<code>xlsx</code> <code>xlsx</code>	<code>read.xlsx()</code> <code>read.xlsx2()</code>
JSON	<code>rjson</code>	<code>fromJSON(file=<arquivo>)</code>
SPSS	<code>Hmisc</code>	<code>spss.get()</code>
SAS	<code>Hmisc</code>	<code>sasexport.get()</code>
Stata	<code>foreign</code>	<code>read.dta()</code>
Systat	<code>foreign</code>	<code>read.systat()</code>

Prof. Dr. Razer A N R Montaña

SEPT/UFR

8

8

Leitura/Escrita de Arquivos.

Formato	Pacote	Função
Arquivos		<code>write.table()</code> # arquivos delimitados
CSV		<code>write.csv()</code> # separador , <code>write.csv2()</code> # separador ;
XLS/XLSX	xlsx xlsx	<code>write.xlsx()</code> <code>write.xlsx2()</code>
JSON	rjson	<code>toJSON()</code> e <code>write()</code>
SPSS	foreign	<code>write.foreign(data, txt-file, sps-file, package="SPSS")</code>
SAS	foreign	<code>write.foreign(data, txt-file, sps-file, package="SAS")</code>
Stata	foreign	<code>write.dta()</code>
Systat	foreign	<code>write.foreign(..., package="Systat")</code>

Prof. Dr. Razer A N R Montaña

SEPT/UFRP

9

9

Arquivos

<https://www.rdocumentation.org/packages/utils/versions/3.6.2/topics/read.table>

```
read.table(file, header = FALSE, sep = "", quote = "\"",
  dec = ".", numerals = c("allow.loss", "warn.loss", "no.loss"),
  row.names, col.names, as.is = !stringsAsFactors,
  na.strings = "NA", colClasses = NA, nrows = -1,
  skip = 0, check.names = TRUE, fill = !blank.lines.skip,
  strip.white = FALSE, blank.lines.skip = TRUE,
  comment.char = "#",
  allowEscapes = FALSE, flush = FALSE,
  stringsAsFactors = default.stringsAsFactors(),
  fileEncoding = "", encoding = "unknown", text, skipNul = FALSE)

read.csv(file, header = TRUE, sep = ",", quote = "\"",
  dec = ".", fill = TRUE, comment.char = "", ...)

read.csv2(file, header = TRUE, sep = ";", quote = "\"",
  dec = ",", fill = TRUE, comment.char = "", ...)

read.delim(file, header = TRUE, sep = "\t", quote = "\"",
  dec = ".", fill = TRUE, comment.char = "", ...)

read.delim2(file, header = TRUE, sep = "\t", quote = "\"",
  dec = ",", fill = TRUE, comment.char = "", ...)
```

Prof. Dr. Razer A N R Montaña

SEPT/UFRP

10

10

Arquivos

- Para ler um arquivo texto formatado usa-se `read.table`
 - Também tem-se as variantes `read.delim()`, `read.delim2()`, `read.csv()`, `read.csv2()`

```
dados <- read.table("arquivo.dat")
```
- Gera um Data Frame dos dados
- Parâmetros
 - `header`: (T/F) se o arquivo tem cabeçalho
 - `sep`: separador de campos (ex, `"\t"`, `" "`, etc)
 - `quote`: caractere que delimita strings (ex, `"\""`)
 - `dec`: ponto decimal (ex, `"."`, `"."`)
 - Etc.....

Prof. Dr. Razer A N R Montaña

SEPT/UFPR

11

11

Arquivos

- Ex, `arquivo.dat`

```
> dados <- read.table("arquivo.dat")
```

```
> dados
```

```
> dados
```

```
      V1      V2
1  nome sobrenome
2  Razer  Montaña
3  Darth   Vader
4   Luke Skywalker
5 Obi-wan  Kenobi
```

	nome	sobrenome
1	Razer	Montaña
2	Darth	Vader
3	Luke	Skywalker
4	Obi-wan	Kenobi

Prof. Dr. Razer A N R Montaña

SEPT/UFPR

12

12

Arquivos

- Ex, arquivo.dat

```
> dados <- read.table("arquivo.dat", header=T)
```

```
> dados
```

```
> dados
```

```
      nome sobrenome
1   Razer   Montaña
2   Darth   Vader
3    Luke Skywalker
4 Obi-wan   Kenobi
```

	nome	sobrenome
1	Razer	Montaña
2	Darth	Vader
3	Luke	Skywalker
4	Obi-wan	Kenobi

Prof. Dr. Razer A N R Montaña

SEPT/UFPR

13

13

Arquivos

- Variantes

- São *wrappers* para o `read.table()`

- `read.delim(file, header = TRUE, sep = "\t", quote = "\"", dec = ".", fill = TRUE, comment.char = "", ...)`

- Separador TAB

- Strings com "

- Ponto decimal.

- `read.delim2(file, header = TRUE, sep = "\t", quote = "\"", dec = ",", fill = TRUE, comment.char = "", ...)`

- Separador TAB

- Strings com "

- Ponto decimal,

Prof. Dr. Razer A N R Montaña

SEPT/UFPR

14

14

Arquivos

- Para escrever em arquivos, usa-se o `write.table()`

```
write.table(x, file = "", append = FALSE, quote = TRUE, sep = " ",
           eol = "\n", na = "NA", dec = ".", row.names = TRUE,
           col.names = TRUE, qmethod = c("escape", "double"),
           fileEncoding = "")
```

- Parâmetros

- `sep` : separador de campos
- `dec` : ponto decimal
- `row.names` : exportar os nomes de linhas
- `col.names` : exportar os nomes de colunas

Prof. Dr. Razer A N R Montaña

SEPT / UFPR

15

15

Arquivos

- Para escrever em arquivos, usa-se o `write.table()`

```
> df <- data.frame(nome=c("Razer", "Anthom", "Nizer",
"Rojas", "Montaña"), idade=c(25, 30, 18, 15, 20))
```

```
> df
```

	nome	idade
1	Razer	25
2	Anthom	30
3	Nizer	18
4	Rojas	15
5	Montaña	20

```
> write.table(df, "idades.txt")
```

```
idades.txt
1 "nome" "idade"
2 "1" "Razer" 25
3 "2" "Anthom" 30
4 "3" "Nizer" 18
5 "4" "Rojas" 15
6 "5" "Montaña" 20
```

Prof. Dr. Razer A N R Montaña

SEPT / UFPR

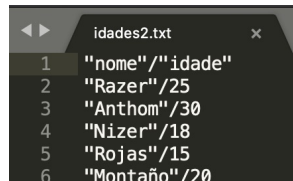
16

16

Arquivos.

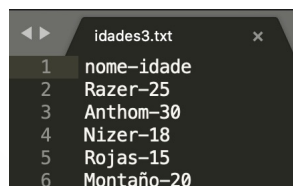
- Outros exemplos

```
> write.table(df, "idades2.txt", row.names=F, sep="/")
```



```
idades2.txt
1 "nome"/"idade"
2 "Razer"/25
3 "Anthom"/30
4 "Nizer"/18
5 "Rojas"/15
6 "Montaño"/20
```

```
> write.table(df, "idades3.txt", quote=F, row.names=F, sep="-")
```



```
idades3.txt
1 nome-idade
2 Razer-25
3 Anthom-30
4 Nizer-18
5 Rojas-15
6 Montaño-20
```

Prof. Dr. Razer A N R Montaño

SEPT/UFPR

17

17

CSV

- Para ler um arquivo CSV usa-se `read.csv()`

```
> dados <- read.csv("/Users/razer/teste.csv")
```

- Gera um Data Frame dos dados

- Parâmetros

- `header = TRUE | FALSE`
- `sep = ","`
- `dec = "."`
- `fileEncoding = "UTF-8"`

- Dado o arquivo `fruitohms.csv`

- <http://www.razer.net.br/datasets/fruitohms.csv>
- Contém a relação entre % de suco e resistência elétrica (ohms) em Kiwis
- Faça download e coloque em uma pasta de seu conhecimento

Prof. Dr. Razer A N R Montaño

SEPT/UFPR

18

18

CSV

- Se quiser trocar o diretório de trabalho, pode-se fazer

```
> setwd("/Users/razer/Downloads")
```

- Leia o arquivo com:

```
> dados <- read.csv("frutohms.csv")
```

- OU

```
> dados <- read.csv("http://www.razer.net.br/datasets/frutohms.csv")
```

- Mostre alguns dados

```
> head(dados)
```

```
> head(dados)
  X juice ohms
1 1   4.0 4860
2 2   5.0 5860
3 3   5.5 6650
4 4   7.5 7050
5 5   8.5 5960
6 6   8.5 6750
```

Prof. Dr. Razer A N R Montaña

SEPT / UFPR

19

19

CSV

- Para escrever em CSV use `write.csv()` ou `write.csv2()`

```
> df <- data.frame(nome=c("Razer", "Anthom", "Nizer", "Rojas",  
"Montaña"), salario=c(50000.00, 85439.33, 123987.654, 18698.10, 73681))
```

```
> write.csv(df, "salario.csv")
```

```
salario.csv
1  """, "nome", "salario"
2  "1", "Razer", 50000
3  "2", "Anthom", 85439.33
4  "3", "Nizer", 123987.654
5  "4", "Rojas", 18698.1
6  "5", "Montaña", 73681
```

```
> write.csv2(df, "salario2.csv")
```

```
salario2.csv
1  """, "nome", "salario"
2  "1", "Razer", 50000
3  "2", "Anthom", 85439,33
4  "3", "Nizer", 123987,654
5  "4", "Rojas", 18698,1
6  "5", "Montaña", 73681
```

Prof. Dr. Razer A N R Montaña

SEPT / UFPR

20

20

CSV.

- Outros exemplos

```
> df <- data.frame(nome=c("Razer", "Anthom", "Nizer", "Rojas",  
"Montaño"), salario=c(50000.00, 85439.33, 123987.654, 18698.10, 73681))  
  
> write.csv2(df, "salario3.csv", row.names=F, quote=F)
```



	nome;salario
1	Razer;50000
2	Anthom;85439,33
3	Nizer;123987,654
4	Rojas;18698,1
5	Montaño;73681



Exercícios.

1. Execute os exercícios apresentados nos slides
2. Carregue o arquivo
http://www.razer.net.br/datasets/Biomassa_REG.csv
3. Carregue o arquivo <http://www.razer.net.br/datasets/fertility.csv>
4. Salve a base de dados IRIS (*data frame iris*) usando os seguintes formatos:
 - a. Separador "***", ponto decimal ".", sem os nomes das linhas e o cabeçalho de colunas, com aspas nos campos string
 - b. CSV com ponto decimal ",", sem os nomes das linhas, com o cabeçalho de colunas e sem aspas nos campos string