

1



2

Pacotes

- O R é formado por três partes básicas
 - **R-base** : o coração do R, com funções principais
 - **Pacotes Recomendados**: instalados junto com o R-base, mas que não são automaticamente carregados
 - Para usar os pacotes recomendados usa-se: `library()`
 - Ex.:

```
> library(nlme)
```

- **Pacotes Contribuídos**: não são instalados junto com o R-base mas estão disponíveis para instalação
- 3 maneiras de instalar pacotes
 - Via CRAN
 - Via Github
 - Via .zip/.tar.gz

Via CRAN

- CRAN: Comprehensive R Archive Network
 - É a rede de distribuição do R
- Comando

```
install.packages("nome_do_pacote")
```

- Exemplo, instalar o H2O, pacote para Machine Learning e BigData:

```
install.packages("h2o")
```

Via Github

- Usa-se o seguinte comando:

```
devtools::install_github("rstudio/shiny")
```

- O pacote `devtools` deve estar instalado (instale via CRAN)
- O pacote Shiny é usado para desenvolver em R para Web
- Informa-se o *username* e o nome do repositório
 - Aqui, o *username* foi "rstudio" e o *repositório* foi "shiny"

Via .zip/.tar.gz

- Deve-se ter o arquivo do pacote
- Usa-se o seguinte comando

```
install.packages("C:/caminho/nome-do-pacote.zip", repos=NULL)
```

- O parâmetro `repos=NULL` indica que é uma instalação local

Uso do Pacote Instalado.

- Após a instalação, para carregar o pacote usa-se:

```
library("h2o")
```

- E já pode ser usado:

```
h2o.init()
```

- Se o pacote já está instalado e só se deseja usar uma função exportada, usa-se o operador `::` (*double colon*)

- Ex

```
devtools::install_github("rstudio/shiny")
```

Console

```

razer — R — 121x35
[razer@Neumann:~]$ r

R version 3.4.0 (2017-04-21) -- "You Stupid Darkness"
Copyright (C) 2017 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin15.6.0 (64-bit)

R é um software livre e vem sem GARANTIA ALGUMA.
Você pode redistribuí-lo sob certas circunstâncias.
Digite 'license()' ou 'licence()' para detalhes de distribuição.

R é um projeto colaborativo com muitos contribuidores.
Digite 'contributors()' para obter mais informações e
'citation()' para saber como citar o R ou pacotes do R em publicações.

Digite 'demo()' para demonstrações, 'help()' para o sistema on-line de ajuda,
ou 'help.start()' para abrir o sistema de ajuda em HTML no seu navegador.
Digite 'q()' para sair do R.

[Área de trabalho anterior carregada]

> 

```

Console

- Ver a versão do R

```
> R.version  
  
platform      _  
arch          x86_64-apple-darwin15.6.0  
arch          x86_64  
os            darwin15.6.0  
system        x86_64, darwin15.6.0  
status  
major         3  
minor         4.0  
year          2017  
month         04  
day           21  
svn rev       72570  
language      R  
version.string R version 3.4.0 (2017-04-21)  
nickname      You Stupid Darkness  
> █
```

Prof. Dr. Razer A N R Montaña

SEPT / UFPR

9

9

Console

- Os comandos são separados por ; ou enter
- É case-sensitive (diferencia maiúsculas e minúsculas)
- Para comentar, inicie com #
- Se um comando não terminar ao dar enter
 - Prompt muda de > para +
- É possível usar um comando novamente através das setas para cima e para baixo

Prof. Dr. Razer A N R Montaña

SEPT / UFPR

10

10

Console

- Efetuar uma operação matemática simples

```
[> 10 + 20
[1] 30
>
```

- Pedir ajuda sobre um comando

```
[> help(print)
>
```

```
print                                package:base                                R Documentation
Print Values
Description:
'print' prints its argument and returns it _invisibly_ (via
'invisible(x)'). It is a generic function which means that new
printing methods can be easily added for new 'class'es.
Usage:
print(x, ...)
## S3 method for class 'factor'
print(x, quote = FALSE, max.levels = NULL,
      width = getOption("width"), ...)
## S3 method for class 'table'
print(x, digits = getOption("digits"), quote = FALSE,
      na.print = "", zero.print = "0", justify = "none", ...)
## S3 method for class 'function'
print(x, useSource = TRUE, ...)
Arguments:
```

Prof. Dr. Razer A N R Montão

SEPT / UFPR

11

11

Console.

- Sair do R: `q()`

```
> q()
Save workspace image? [y/n/c]:
```

- Pergunta se quer salvar o Workspace
 - y – sai e salva o workspace
 - n – sai e não salva o workspace
 - c – cancela a saída

Prof. Dr. Razer A N R Montão

SEPT / UFPR

12

12

Comandos Auxiliares.

Função	Descrição
<code>ls()</code>	Lista de objetos/variáveis definidas
<code>str(x)</code>	Mostra informações/estrutura sobre x
<code>rm(x)</code>	Deleta a variável x
<code>class(x)</code>	Mostra o tipo de x
<code>q()</code>	Sai do R, com opção de salvar workspace e histórico

Prof. Dr. Razer A N R Montão

SEPT/UFPR

13

13

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

Prof. Dr. Razer A N R Montão

SEPT/UFPR

14

14

Entrada e Saída

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

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

- Gera o arquivo `saida.txt`

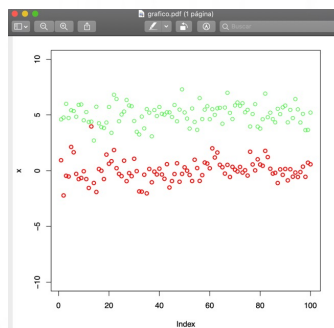


Entrada e Saída

- 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`



Entrada e Saída

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

```
png("arquivo.png")
jpeg("arquivo.jpg")
```

Prof. Dr. Razer A N R Montañó

SEPT/UFPR

17

17

Entrada e Saída

- Comandos para mostrar dados no console

- `print(x)`
 - Retorna seu parâmetro
 - Recebe um parâmetro a ser mostrado
 - Pode ser usado com tipos complexos (listas, etc)
 - Não pula linhas ("`\n`"), pois mostra a sequência de caracteres

```
> print("oi")
[1] "oi"
> print("tchau")
[1] "tchau"
> print("oi\n")
[1] "oi\n"
> x <- list(nome="Razer")
> print(x)
$nome
[1] "Razer"

> a <- print(x)
$nome
[1] "Razer"

> a
$nome
[1] "Razer"

> class(a)
[1] "list"
```

Prof. Dr. Razer A N R Montañó

SEPT/UFPR

18

18

Entrada e Saída

- Comandos para mostrar dados no console
 - `cat(x, y,)`
 - Retorna NULL
 - Pode redirecionar para escrever em arquivos
 - Concatena os parâmetros usando um separador (pode ser definido com parâmetro `sep`)

```
> cat("oi")
oi> cat("oi\n")
oi
> cat("oi", "tchau\n\n")
oi tchau

> a <- cat("oi", "tchau\n\n")
oi tchau

> a
NULL
> x <- list(nome="Razer")
> cat(x)
Error in cat(x) : argumento 1 (tipo 'list') não suportado por 'cat'
```

Prof. Dr. Razer A N R Montão

SEPT/UFPR

19

19

Entrada e Saída.

- Comandos para mostrar dados no console
 - `paste(x, y, ...)`
 - Concatenar Strings
 - Setar o separador usado na concatenação (parâmetro `sep`, default " ")
 - `paste0()`
 - Igual ao `paste()`, mas o separador default é ""

```
> paste("oi")
[1] "oi"
> paste("oi", "tchau")
[1] "oi tchau"
> a <- paste("oi", "tchau")
> a
[1] "oi tchau"
> cat(a)
oi tchau> paste("oi", "tchau", sep="_")
[1] "oi_tchau"
> paste0("oi", "tchau")
[1] "oitchau"
```

Prof. Dr. Razer A N R Montão

SEPT/UFPR

20

20

Comandos Simples

- No console pode-se executar qualquer comando
- Ex, criar um vetor de inteiros de 1 a 30

```
> 1:30
```

- Resulta

```
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
[26] 26 27 28 29 30
```

- Os números entre colchetes indicam o índice do primeiro número de cada linha

```
> seq(1, 200, 3)
[1] 1 4 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55
[20] 58 61 64 67 70 73 76 79 82 85 88 91 94 97 100 103 106 109 112
[39] 115 118 121 124 127 130 133 136 139 142 145 148 151 154 157 160 163 166 169
[58] 172 175 178 181 184 187 190 193 196 199
```

Prof. Dr. Razer A N R Montaña

SEPT/UFPR

21

21

Comandos Simples

- Para colocar o vetor em uma variável

```
> x <- 1:30
```

```
> x
```

- Resulta

```
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
[26] 26 27 28 29 30
```

Prof. Dr. Razer A N R Montaña

SEPT/UFPR

22

22

Comandos Simples

- Pode-se efetuar operações aritméticas

- Ex

```
> 2 * 2 - (4 + 4) / 2
```

- Resulta

```
[1] 0
```

- Para operações aritméticas, funciona a precedência matemática e o uso de parênteses
- Se digitar um comando incompleto, como 5+, é mostrado um "+" indicando que o comando precisa ser finalizado

```
> 5+
```

```
+ 5
```

```
[1] 10
```

Prof. Dr. Razer A N R Montão

SEPT/UFPR

23

23

Comandos Simples..

- Se digitar um comando que o R não reconhece, dá uma mensagem de erro

- Ex

```
> 5 % 5
```

- Resulta

```
Error: unexpected input in "5 % 5"
```

- Ex

```
> coisaesquisita
```

- Resulta

```
Erro: objeto 'coisaesquisita' não encontrado
```

Prof. Dr. Razer A N R Montão

SEPT/UFPR

24

24