



Métodos Computacionais em R

Claudio Neto, Dr.

Universidade Federal do Recôncavo da Bahia
Cruz das Almas, BA, BRA

Novembro, 8, 2021

Agradecimentos

- Esses slides são fortemente baseados no trabalho da Dra. Silvia Canelón (@spcanelon), "Tour of the Tidyverse", bem como nos slides da Dra. Sara Mortara e da Dra. Andrea Sánchez-Tapia (Instituto Hub).
- Para isso usei `xaringan`  do Dr. Yihui Xie e `xaringanExtra`  do Garrick Aden-Buie, e adaptado do R-Ladies `xaringan` theme designed by Alison Hill
- Utilizaremos várias referências, todas disponíveis nesse link [GitHub](#)

apresentação

- **Claudio Carneiro da Cruz Neto** Economista (UESC), MSc. Desenvolvimento Regional e Meio Ambiente (UESC), Dr. em Economia (UNB). Sou usuário do R desde 2018, e me considero apaixonado por essa linguagem de programação. Amo trabalhar estudando a interface entre Economia e Meio Ambiente, por isso a maioria dos meus trabalhos está direcionado para essa linha de pesquisa.
- Website, Twitter, Github
- **Grupo de Estudos em Economia e Meio Ambiente** Se quiser saber mais, clique aqui

sobre nosso curso

Ambientação ao R e ao RStudio

- Vantagens de usarmos programação para analisar dados
- O que é e por que usar o RStudio
- Escrevendo e rodando códigos

Introdução à programação em R

- Objetos e vetores
- Classes: números, caracteres, lógicos e data frames
- Operadores de seleção; utilizar e criar funções no R
- Operadores lógicos e aritméticos
- Loopings de programação: for e while
- O operador pipe (%>%)

sobre nosso curso - continuação

Programação no dia a dia

- Exemplos de scripts reais
- Usando e instalando pacotes
- Estruturação de projetos
- Boas práticas de programação em R

Importação de bases para dentro do R

- Importando arquivos de texto: .csv e .txt
- Importando arquivos excel: .xls e .xlsx

sobre nosso curso - continuação

Manipulação de bases de dados (o pacote tidyverse)

- Seleccionando colunas
- Filtrando e ordenando linhas
- Modificando e criando colunas
- Criando sumarizações
- Juntando duas bases

Gráficos (o pacote ggplot2)

- Gráficos de pontos (dispersão)
- Gráficos de barras
- Boxplots e histogramas
- Customizando gráficos

sobre nosso curso - continuação

Construindo artigos e relatórios em R Markdown

- Introdução ao R Markdown
- Construindo relatórios em HTML, PDF e Word

sobre hoje

1. conhecendo a linguagem R e fluxo de trabalho
2. introdução ao R
3. manipulação de dados
4. criando figuras básicas

códigos de apoio

todos os códigos disponíveis em:

um bestiário de funções úteis

antes de começar

- procure a janela de chat do zoom, e não hesite em fazer perguntas
- tem um botão verde para contar se tudo está certo. Periodicamente vamos perguntar se todo mundo está acompanhando, por favor use este botão ou responda no chat
- pode abrir sua câmera :) procure manter seu microfone silenciado.

1. conhecendo a linguagem R e fluxo de trabalho

por que R?

- linguagem de **código aberto, livre & sem custo**: um dos pilares da ciência aberta
- **baseada em scripts**: essencial para reprodutibilidade, facilidade para ampliar a escala, transparência, correção/robustez
- acessível (em comparação a outras linguagens de programação)
- muito comum na Biologia, Ciência de Dados e outras áreas. A escolha para quem precisa fazer análises estatísticas
- **interdisciplinar e modular**: muitos pacotes são escritos por especialistas
- uma filosofia de passar facilmente do uso à programação

por que R?

- ferramentas para **comunicar os resultados**: manuscritos, apresentações, *apps*
- comunicação com outras linguagens de programação (ex. python e **reticulate**)
- ótima capacidade gráfica
- **suporte oficial**: documentação, vignettes, listas de email
- **suporte não oficial**: uma comunidade ativa: listas de email, grupos de usuáries (useR), capítulos de **R-Ladies+**, comunidade de **RStudio**, Stack Overflow,

R tem uma estrutura modular: pacotes

- a instalação de base de **R** inclui pacotes mantidos pelo **R Core Development Team**
- repositório oficial de pacotes: **CRAN** (The Comprehensive **R** Archive Network <https://cran.r-project.org/>) or Bioconductor, GitHub, rOpenSci.org
- pacotes fora da instalação de base precisam ser **instalados** e **carregados**
- os pacotes são coleções de **funções**
- as funções têm **opções**: "parâmetros", "argumentos"
- vamos manter a convenção **nome_da_funcao()** e **pacote::nome_da_funcao()** quando for preciso especificar o pacote ao qual a função pertence.

Então vamos instalar o R

- A instalação padrão do R é feita a partir do CRAN, uma rede de servidores espalhada pelo mundo que armazena versões idênticas e atualizadas de códigos e documentações para o R.
- Para instalar o R no Windows, siga os seguintes passos:
- Acesse o CRAN: <https://www.r-project.org/>
- No menu à esquerda, encontre a opção Download e clique em CRAN.
- Escolha a opção de servidor (mirror) mais próxima de você.

Então vamos instalar o R.

- Clique na opção base. Na nova página, clique em **Download R x.x.x for Windows*, sendo x.x.x o número da versão que será baixada. Em qualquer problema, tente escolher outro servidor no passo 3.
- Feito o download, clique duas vezes no arquivo baixado e siga as instruções para instalação.
- Para realizar o Download **R** 4.1.0 para Windows, clique no link: (<https://cran.r-project.org/bin/windows/base/R-4.1.0-win.exe>)
- pacotes fora da instalação de base precisam ser **instalados** e **carregados**

Então vamos instalar o R Studio.

- Agora vamos instalar a versão open source do RStudio, a IDE que utilizaremos para escrever e executar códigos em R.
- Para instalar o RStudio no Windows, siga os seguintes passos:
- Acesse a página de downloads da RStudio:
<https://rstudio.com/products/rstudio/download/#download>
- Se você tiver acesso administrador, baixe a versão referente ao seu sistema operacional que está na lista de All Installers.
- Se você não tiver acesso de administrador, baixe a versão referente ao seu sistema operacional que está na lista de Zip/Tarballs.
- Clique duas vezes no arquivo baixado e siga as instruções para instalação.

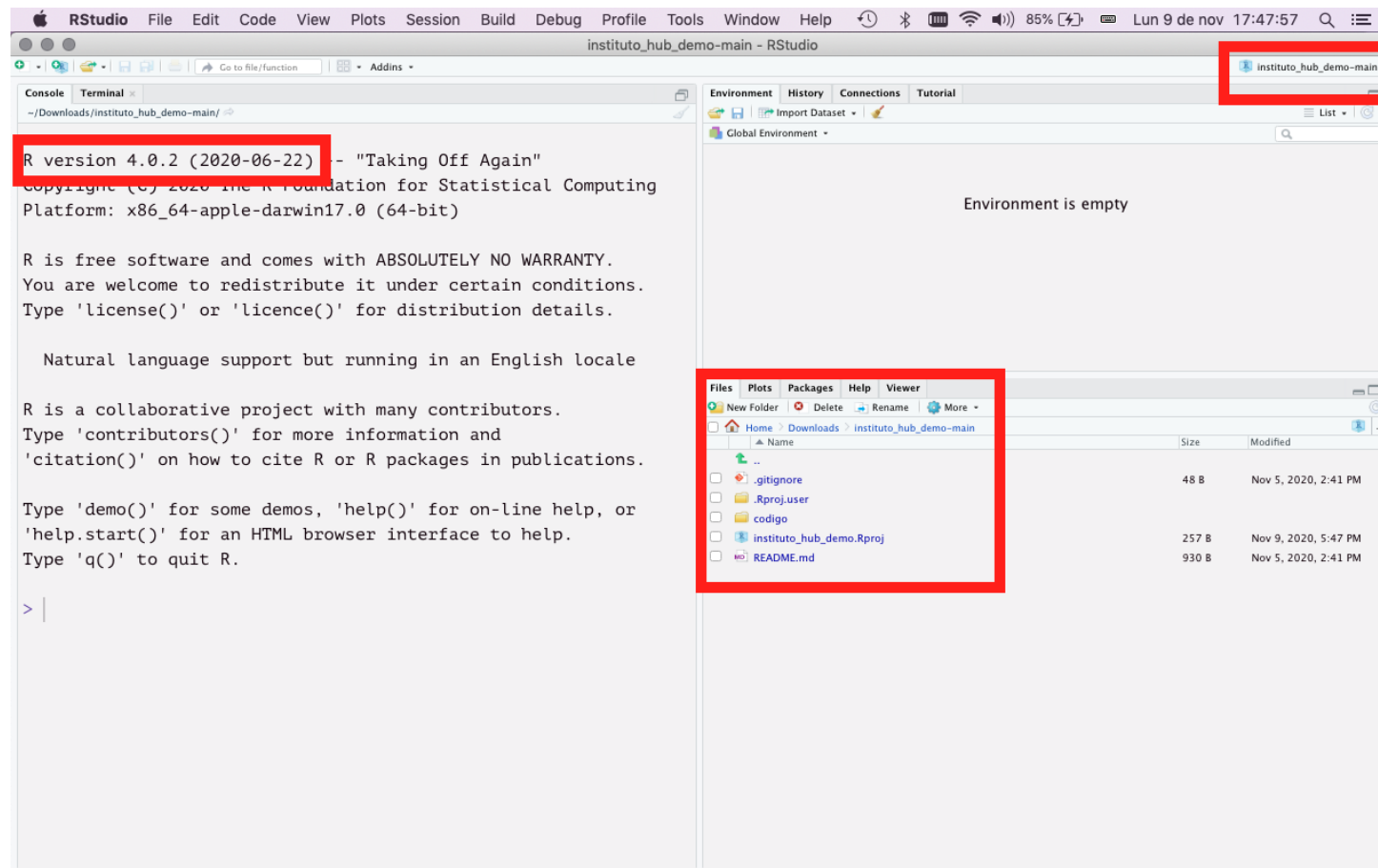
Pense no R assim



E agora pense no R Studio



Visão do RStudio



- tela separada em três painéis. criar um **script** abre o quarto painel.
- podemos executar código no console mas vamos usar o **script** para guardar os passos.
- escrever no script e clicar **control + enter** para mandar o conteúdo para o console.
- outras abas: help, environment, plots, files...

configuração e organização de projetos

pasta de trabalho

- R tem que saber **em que pasta você está trabalhando**, para poder buscar os arquivos que você quer ler, e saber onde guardar os resultados de suas análises: **working directory**: pasta de trabalho
- para saber onde você está: **getwd()**
- você pode mudar esta localização usando **setwd()** mas não é recomendado (erros!)
Jenny Bryan
- melhor: em vez de abrir RStudio e trocar a pasta, abrir um script de R (extensão **.R**) ou um projeto de **RStudio project** (extensão **.Rproj**) **já na pasta onde você vai trabalhar**.
(Igual ao que você faria abrindo um arquivo)

organização de projetos e Boas Práticas

- cada projeto de análise de dados deveria estar em **uma única pasta**
- **subpastas** para organizar o conteúdo.
- em vez de usar **caminhos absolutos** é melhor usar **caminhos relativos**
- evite **C://usuario/seu_nome/suas_pastas/seu_projeto** -> caminho absoluto
- **.** "aqui" (o resultado de **getwd()**)
- **./subpastas** (ex. **/figs**)
- o nível acima **..**
- os **dados originais** não devem ser modificados. qualquer modificação deve ser salva à parte, junto com os scripts que fizeram a modificação.

o projeto atual

```
./nome_do_projeto/  
├── README.md  
├── codigo  
├── dados  
│   ├── brutos  
│   └── processados  
├── figs  
└── nome_do_projeto.Rproj
```

os projetos de RStudio

- arquivos **opcionais**, com extensão **.Rproj** para abrir a sessão de R no local correto
- um projeto de RStudio pode ser criado clicando no canto superior direito (**Project:**
none > New project)

Introdução a R

criação de objetos

- executar no script ou no console -> não cria nada

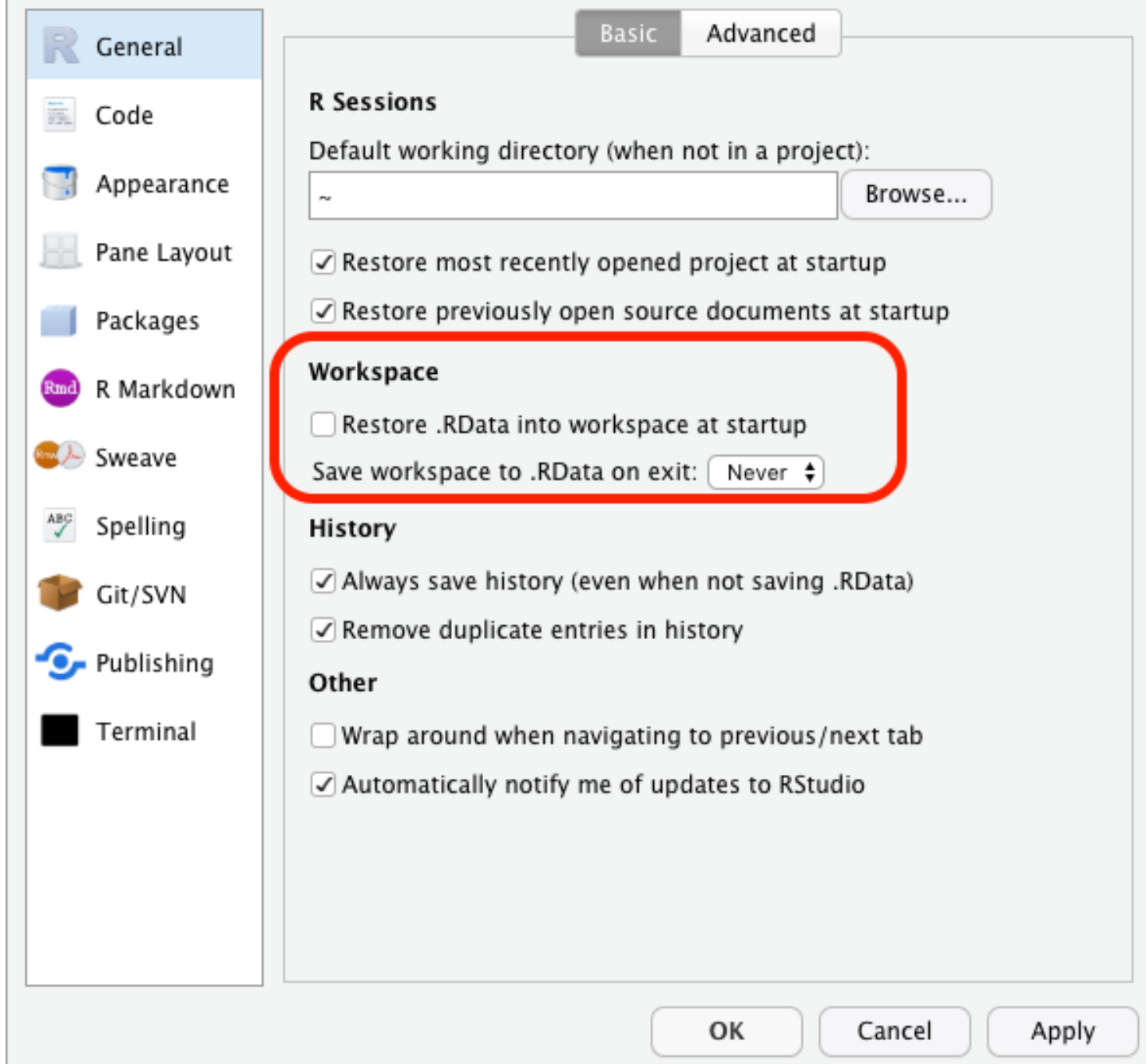
--

- **<-** serve para atribuição a objetos - *ALT + -*
- cria dentro do espaço de trabalho "**workspace**", aba **Environment** de RStudio
- pode explorar o conteúdo com **ls()** (listar)
- quando você cria objetos eles ocupam memória RAM do computador
- o **workspace** pode ser salvo e carregado entre sessões mas não é recomendável fazer isto **#boaspráticas**

criação de objetos

- confiar no **script** para recriar todos os objetos (**reprodutibilidade!**)
- guardar os objetos importantes como **tabelas**, **figuras**, ou objetos individuais de R (**.rda** ou **.rdata**)
- um script deve ler ou criar objetos no início e salvar resultados no final
- é bom criar diferentes **scripts sucessivos** para manter a ordem
- você pode perder controle de como foram criados os objetos
- pode guardar objetos com erros e isto pode interferir no seu trabalho

nas opções gerais



tipos de dados em R

- R entende dados **numéricos** (*numeric*, e *integer* para números inteiros), de **texto** (*character*), valores lógicos (**TRUE/FALSE**), entre outros.

```
populacao <- c(1500, 2000)
estados <- c("Acre", "Alagoas", "Amapá", "Amazonas", "Tocantins")
class(populacao)
```

```
## [1] "numeric"
```

```
class(estados)
```

```
## [1] "character"
```


tipos de dados em R

```
verdadeiro_falso <- c(FALSE, TRUE)
verdadeiro_falso
```

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

```
class(verdadeiro_falso)
```

```
## [1] "logical"
```

```
posicao <- c(1L, 2L)
posicao
```

```
## [1] 1 2
```

```
class(posicao)
```

vetores

- `c()` para criar - "concatenar"

```
str(estados) #estrutura
```

```
## chr [1:5] "Acre" "Alagoas" "Amapá" "Amazonas" ...
```

```
length(estados)
```

```
## [1] 5
```

selecionando elementos de vetores (*subsetting*)

- entre **colchetes** `[]`
- com vetores de **posição** ou vetores **lógicos** (`TRUE/FALSE`)

```
estados <- c("Acre", "Alagoas", "Amapá", "Amazonas", "Tocantins")
estados[1]
estados[c(TRUE, FALSE, TRUE, FALSE)]
estados[1:2]
estados[1:4]
#estados[ 1,3,5] #NAO! a vírgula marca dimensões
estados[c(1,3,5)] #tem que ser um vetor
```

subsetting com cláusulas lógicas

```
casos <- c(150, 200, 400, 500, 500, 600)
casos > 150
casos[casos > 150]
casos >= 150 # superior ou igual
casos < 300
casos == 200
casos != 200
```

operadores lógicos

- igualdade: `==`
- diferença: `!=` (! em geral é negação)
- desigualdades: `<`, `>`, `<=`, `>=`
- união (OR/OU): `|`
- interseção (AND): `&`
- pertencimento: `%in%`
- não pertencimento: `!a %in% b`
- `casos == 200`
- `casos != 200`
- `casos < 300, casos >= 150`
- `casos < 200 | casos > 500`
- `casos > 200 & casos < 500`
- `casos %in% c(100, 200, 300)`
- `!casos %in% c(400)`

O operador lógico cria um **vetor lógico**, a seleção vai entre colchetes:

```
casos[casos > 150]
```

```
## [1] 200 400 500 500 600
```

criando vetores numéricos

```
1:10  
seq(1, 10, 2)  
rep(1:4, 2)  
rep(1:4, each = 2)  
unique(casos)
```

estruturas de dados em R

- **vetor**: lineal, uma dimensão só: `length()`
- **fatores**: vetores que representam **variáveis categóricas** e portanto têm níveis (`levels()`
`factor(estados)`)
- **matrizes**: arranjos de duas dimensões de **dados do mesmo tipo** (`dim()`, `nrow()`,
`ncol()`).
- **listas**: literalmente listas de qualquer outro objeto (inclusive listas)
- **data frames**: arranjos bi-dimensionais de dados de diferentes tipos (i.e., uma coluna numérica, outra com nomes, outra com um fator etc.)

instalando pacotes

```
# Para instalar pacotes desde CRAN  
install.packages("remotes")  
  
# Para instalar coronabr desde GitHub  
remotes::install_github("cccneto/Ibamam")  
  
# Para carregar pacotes  
library(name_pckg)  
  
# Para buscar ajuda  
?name_pckg
```


funções, argumentos e entendendo a ajuda

```
help(funcao)  
?funcao  
??palavra_chave  
args(funcao)
```

```
help("get_corona_br")  
?get_corona_br  
??corona  
args(get_corona_br)
```

- ou selecionar o nome da função e clicar **F1**
- No **help** estão os argumentos na ordem correta, e os valores padrão (por *default*) estão indicados. Se a gente não especificasse a função usaria esses valores.

download dos dados para Amapá

```
#library(coronabr)
```

```
caminho <- "dados/brutos"  
if (!dir.exists(caminho)) {  
  dir.create(caminho)  
}  
get_corona_br(dir = caminho,  
              filename = "01-amapa",  
              uf = "AP")
```

```
amapa <- read.csv("dados/brutos/01-amapa.csv",  
                  stringsAsFactors = FALSE,  
                  encoding = "UTF-8")  
class(amapa)
```

inspecionar objetos data.frame

```
names(amapa)
dim(amapa)
nrow(amapa)
ncol(amapa)
head(amapa) # 6 linhas por padrão
tail(amapa)
rownames(amapa)
length(amapa) # numero de colunas
summary(amapa) # quantis e a média
```

selecionar colunas e filtrar linhas

- entre colchetes também mas **as dimensões separadas por uma vírgula**

`amapa[linhas, colunas]`

```
amapa[1:3, 1:3] # 3 primeiras linhas, e três primeiras colunas
```

```
##      city city_ibge_code      date
## 1 Macapá      1600303 2020-03-20
## 2  <NA>         16 2020-03-20
## 3 Macapá      1600303 2020-03-21
```

```
amapa[4, 13] #quarto elemento da coluna 13
```

```
## [1] 0
```

fontes de ajuda

- Material de estudo em português [link](#)
- [LearnR4Free](#) em inglês e em espanhol
- Disciplina *Projetos de análise de dados usando R* [link](#)
- R Reference Card [PDF](#)
- Base R cheatsheet [PDF](#)
- R Taskviews: listas de pacotes por tema curados por especialistas [link](#)
- Livro [R for Data Science](#)
- Pacote **dados**: os conjuntos de dados de R4DS traduzidos para o português: [GitHub](#)
- RStudio cheatsheet [PDF](#)
- RStudio Data Import Cheatsheet [PDF](#)
- R Bloggers <https://www.r-bloggers.com/>

Onde pedir ajuda?

- Latin-R <http://bit.ly/latinr-slack>
- RLadies+ <https://rladies.org/>
- Slack da comunidade RLadies+ <https://rladies-community-slack.herokuapp.com/>
- YouTube de RLadies+ global <https://www.youtube.com/c/RLadiesGlobal>
- Comunidade de RStudio <https://community.rstudio.com/>
- Configuração das opções globais de RStudio [link](#)
- [Stack OverFlow](#)
- E os coleguinhas de turma!

Proximo titulo da seção

