

Introdução à Linguagem R - Workshop

Encontro 1

Davi Moreira

14 de Outubro, 2018

Sumário

1 Estrutura	2
1.1 Principais objetivos	2
1.2 Estrutura dos encontros	2
1.3 Sobre o professor:	2
1.4 Objetivos do primeiro encontro	3
2 R, o RStudio e o GitHub	3
2.1 O que é o R?	3
2.2 O que é o RStudio?	3
2.3 O ambiente do RStudio	4
2.4 O que é versionamento e o GitHub?	4
3 Como o R funciona e onde obter ajuda	7
3.1 Criando um projeto com versionamento	7
3.2 Onde obter ajuda	10
4 Estruturas de dados no R	11
4.1 Operadores aritméticos, operadores lógicos e de comparação	11
4.2 Estruturas de dados	11
4.3 Exemplo de uso dos operadores	15
5 Importação e exportação de dados	16
5.1 Tipos de arquivos de dados	16
5.2 Importação de dados	17
5.3 Exportação de dados	18
6 Atividade Prática - Git Commit / Push / Pull	19
7 Atividade Prática - R	19
8 Links úteis para o próximo encontro	19

1 Estrutura

1. INTRODUÇÃO, IMPORTAÇÃO E EXPORTAÇÃO DE DADOS

- Apresentação do ambiente R e do RStudio;
- Versionamento: GitHub;
- Onde obter ajuda;
- Estruturas de dados no R (variáveis, vetores, matrizes, listas, data frame);
- Operadores matemáticos e operadores lógicos;
- Tipos de arquivos de dados;
- Importação e exportação de dados;

2. PRÉ-PROCESSAMENTO DE DADOS

- Tidyverse tools: Pacote dplyr
- Filtro, ordenação, agregação, sumarização, merging, reshaping, criação de novas variáveis;
- Manipulação de strings e datas;

3. REQUISITANDO DADOS DA WEB

- Extrairando conteúdo da página: rvest, xml2, css e xpath;
- Baixando arquivos da web: pdf, csv, excel;

4. ESTATÍSTICA BÁSICA, VISUALIZAÇÃO DE DADOS E REPORTING

- Medidas de tendência central e dispersão;
- Medidas de associação: o coeficiente de correlação de Pearson; - Testes de médias e proporções;
- Regressão linear;
- Tidyverse tools: Pacote ggplot2 - gráficos de colunas;
- boxplot;
- gráficos de linha;
- faceting;
- formatação;
- reporting (R Markdown)

1.1 Principais objetivos

Até o final o aluno deverá ser capaz de utilizar a linguagem R para, em nível básico:

- Importar dados
- Processá-los para análise
- Elaborar relatórios

1.2 Estrutura dos encontros

- A partir do segundo encontro, 15 minutos serão reservados para dúvidas e revisão do conteúdo do encontro prévio.
- A cada encontro, incluindo o último, será oferecida uma atividade prática.
- Não haverá avaliação.

1.3 Sobre o professor:

Prof. Dr. Davi Moreira

Web-Page: <https://www.davimoreira.com>

Office Hours Agenda: <https://www.wejoinin.com/sheets/jjdlkk>

1.4 Objetivos do primeiro encontro

Até o final do encontro o aluno deverá ser capaz de:

- Criar projeto no RStudio, garantindo seu versionamento;
- Importar base de dados;
- Obter informações relevantes sobre os dados importados.

2 R, o RStudio e o GitHub

2.1 O que é o R?

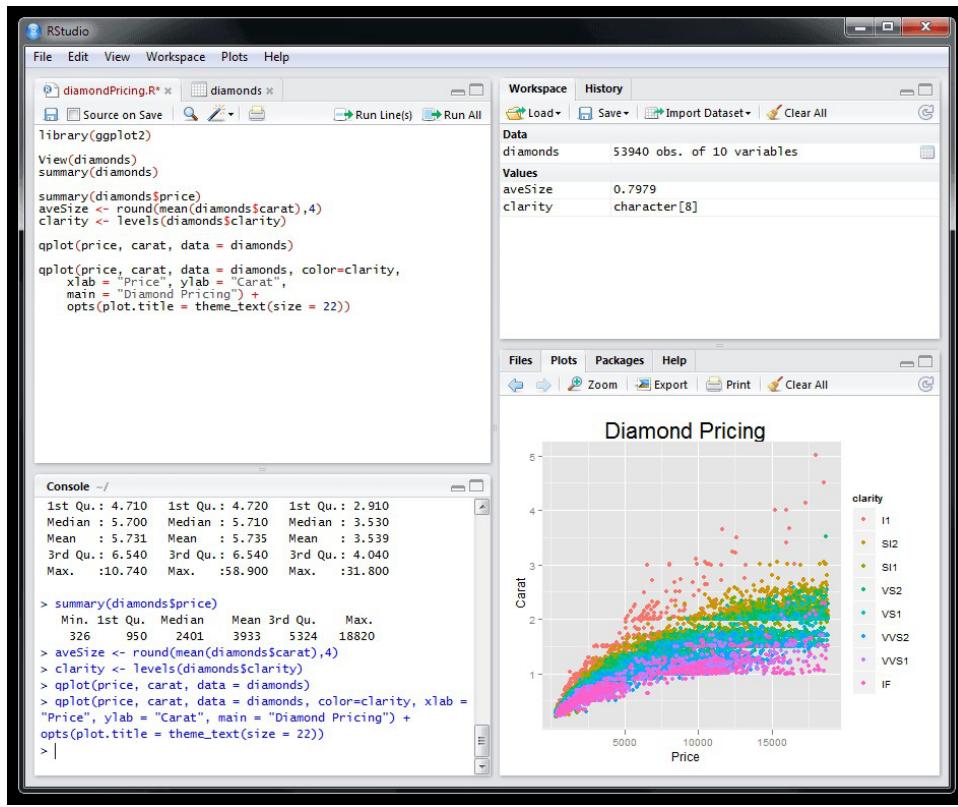
R é uma linguagem de programação e também um ambiente de desenvolvimento integrado para cálculos estatísticos e gráficos (Wikipédia).



2.2 O que é o RStudio?

O RStudio é um software livre de ambiente de desenvolvimento integrado (IDE) para R (Wikipédia RStudio). IDE, do inglês *Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado, é um programa de computador que reúne características e ferramentas de apoio ao desenvolvimento de software com o objetivo de agilizar este processo (Wikipédia IDE).

2.3 O ambiente do RStudio



2.4 O que é versionamento e o GitHub?

Um sistema de controle de versões (ou versionamento - [Wiki](#)), VCS (do inglês *version control system*) ou ainda SCM (do inglês *source code management*) na função prática da Ciência da Computação e da Engenharia de Software, é um software que tem a finalidade de gerenciar diferentes versões no desenvolvimento de um documento qualquer.

O [GitHub](#) é uma plataforma de hospedagem de código para controle de versão e colaboração. Ele permite que você e outros trabalhem juntos em projetos de qualquer lugar. Para mais detalhes, veja [este Tutorial](#).

2.4.1 Fluxo de versionamento:

Base de dados inicial

History

1st Commit



```
"name","year","time","lat","long"  
"Allison",1995,1995-06-03 00:00:00,17.4,-84.3  
"Allison",1995,1995-06-03 00:00:00,17.4,-84.3  
"Allison",1995,1995-06-03 12:00:00,19.3,-85.9  
"Allison",1995,1995-06-03 12:00:00,19.3,-85.9  
"Allison",1995,1995-06-04 00:00:00,22.4,-86.8  
"Allison",1995,1995-06-04 00:00:00,23.3,-86.3  
"Allison",1995,1995-06-04 00:00:00,23.3,-86.3  
"Allison",1995,1995-06-04 12:00:00,24.2,-86.2  
"Allison",1995,1995-06-04 18:00:00,26.2,-86.2  
"Allison",1995,1995-06-05 00:00:00,27.6,-86.1  
"Allison",1995,1995-06-05 00:00:00,28.5,-85.6  
"Allison",1995,1995-06-05 12:00:00,29.6,-84.7  
"Allison",1995,1995-06-05 18:00:00,30.7,-83.8  
"Allison",1995,1995-06-06 00:00:00,31.8,-82.8  
"Allison",1995,1995-06-06 00:00:00,32.7,-81.5  
"Allison",1995,1995-06-06 12:00:00,33.6,-80  
"Allison",1995,1995-06-06 18:00:00,34.5,-79.1  
"Allison",1995,1995-06-07 00:00:00,35.3,-78.9  
"Allison",1995,1995-06-07 00:00:00,37.1,-78.6  
"Allison",1995,1995-06-07 12:00:00,38.1,-78.1  
"Allison",1995,1995-06-07 18:00:00,39.1,-78.2  
"Allison",1995,1995-06-08 00:00:00,41.4,-77.7  
"Allison",1995,1995-06-08 06:00:00,42.4,-76  
"Allison",1995,1995-06-08 12:00:00,43.8,-83.7
```

Project

Pré-processamento

History

1st Commit

2nd Commit

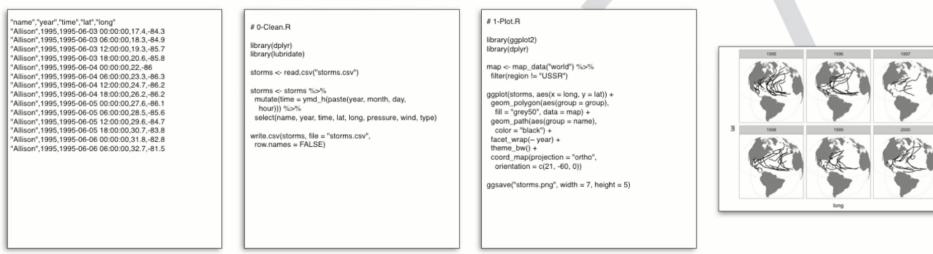
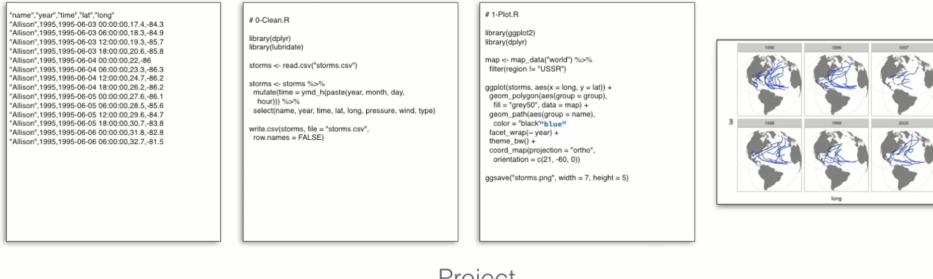


```
"name","year","time","lat","long"  
"Allison",1995,1995-06-03 00:00:00,17.4,-84.3  
"Allison",1995,1995-06-03 00:00:00,17.4,-84.3  
"Allison",1995,1995-06-03 12:00:00,19.3,-85.9  
"Allison",1995,1995-06-03 12:00:00,19.3,-85.9  
"Allison",1995,1995-06-04 00:00:00,22.4,-86.8  
"Allison",1995,1995-06-04 00:00:00,23.3,-86.3  
"Allison",1995,1995-06-04 00:00:00,23.3,-86.3  
"Allison",1995,1995-06-04 12:00:00,24.2,-86.2  
"Allison",1995,1995-06-04 18:00:00,26.2,-86.2  
"Allison",1995,1995-06-05 00:00:00,27.6,-86.1  
"Allison",1995,1995-06-05 00:00:00,28.5,-85.6  
"Allison",1995,1995-06-05 12:00:00,29.6,-84.7  
"Allison",1995,1995-06-05 18:00:00,30.7,-83.8  
"Allison",1995,1995-06-06 00:00:00,31.8,-82.8  
"Allison",1995,1995-06-06 00:00:00,32.7,-81.5
```

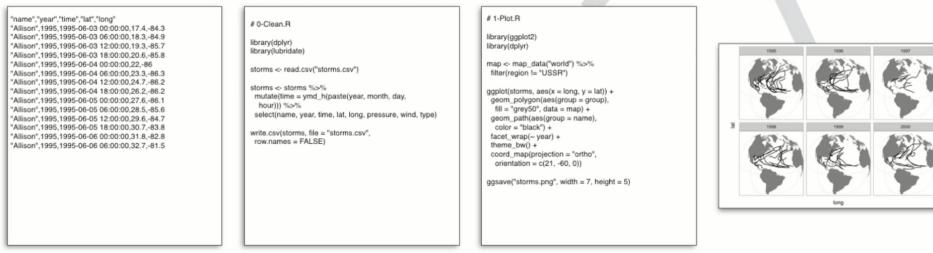
```
# 0-Clean.R  
library(dplyr)  
library(lubridate)  
  
storms <- read.csv("storms.csv")  
  
storms <- storms %>%  
  mutate(time = ymd_hms(paste(year, month, day,  
    hour)))  
  select(-name, -year, time, lat, long, pressure, wind, type)  
  
write.csv(storms, file = "storms.csv",  
  row.names = FALSE)
```

Project

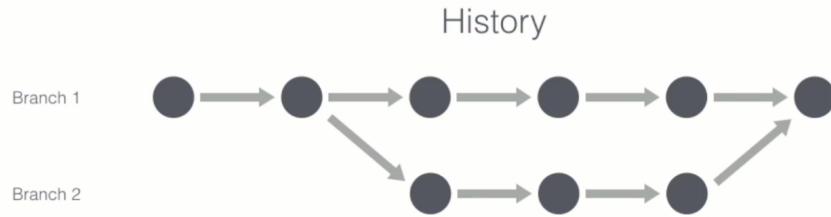
Recuperação de versões



Versões paralelas



Versões paralelas



Visão global



3 Como o R funciona e onde obter ajuda

Como em qualquer tarefa de programação, para que a linguagem R e o ambiente RStudio possam ser utilizados de forma adequada com o objetivo de ser obter a máxima eficiência dessas ferramentas, é necessário ter muita, mas muita organização. Só assim é possível garantir replicabilidade da análise, checagem e sua validação. Felizmente, o RStudio foi programado para nos ajudar nessa tarefa. Por isso, vamos começar por aprender a criar um projeto em seu ambiente.

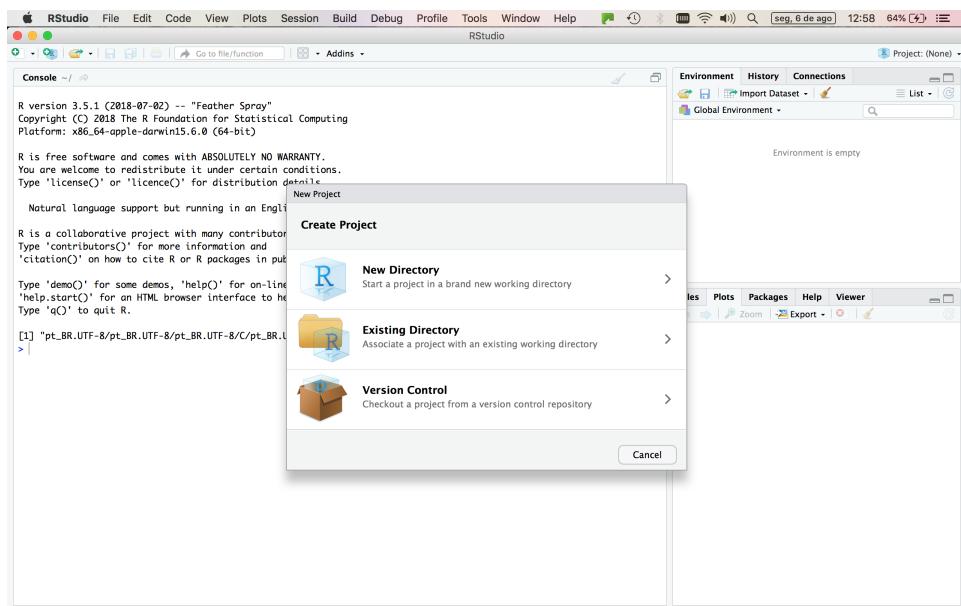
3.1 Criando um projeto com versionamento

Para criar um ambiente de projeto, basta seguir os 5 passos seguir. Para mais informações, ver: - [Webinar RStudio¹](#).

- Veja especialmente [esse Webinar](#).

Passo 1 - Novo Projeto:

¹Para acesso a outros Webinars, acessar: <https://www.rstudio.com/resources/webinars/>



Passo 2 - Diretório:

New Project

Create New Project

Back

Directory name:

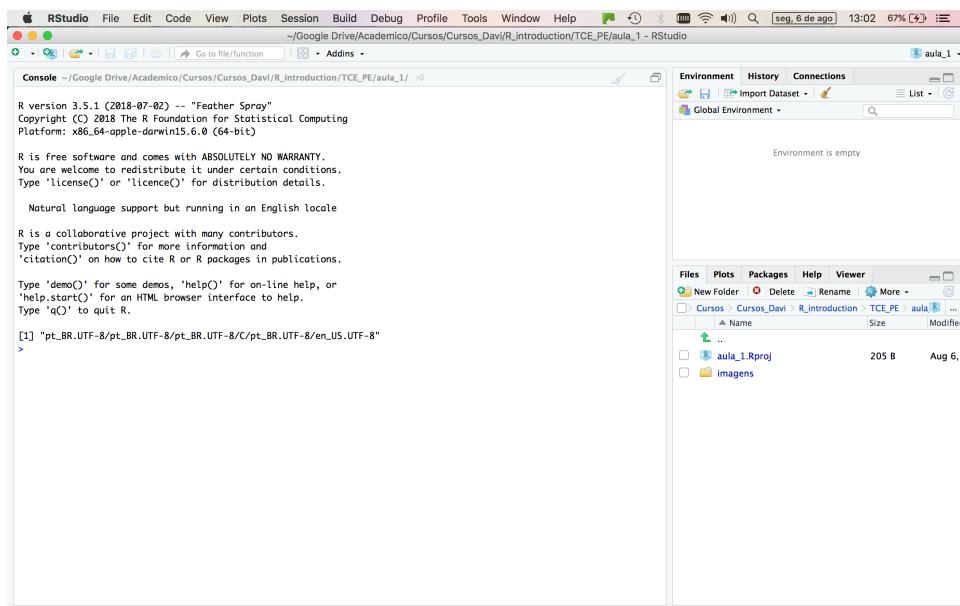
Create project as subdirectory of:

Create a git repository

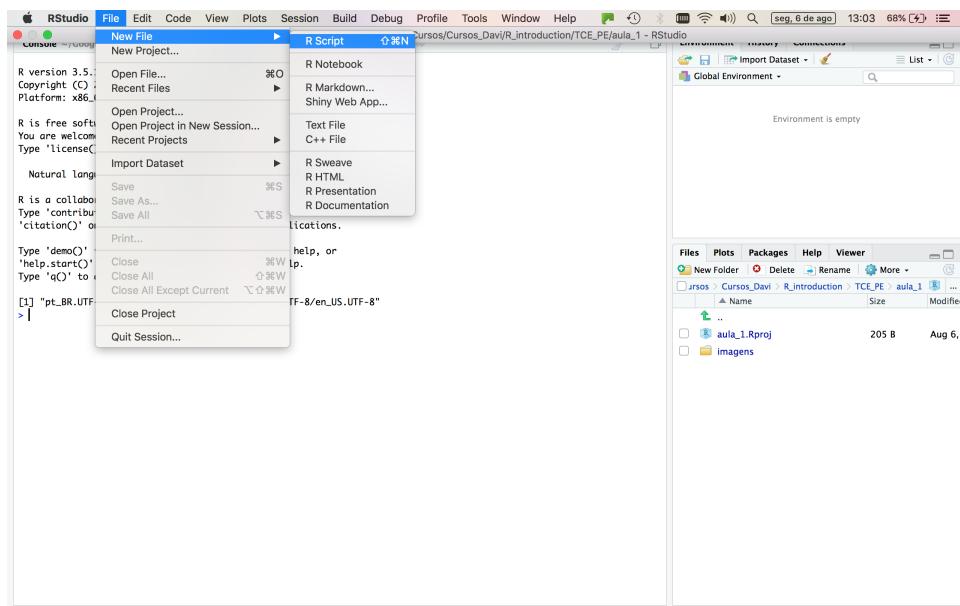
Use packrat with this project

Open in new session

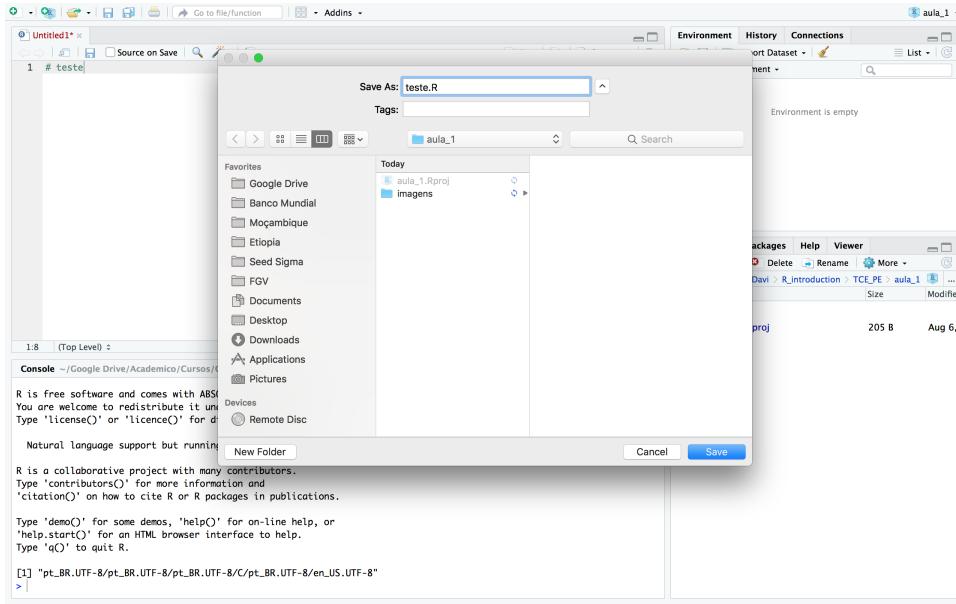
Passo 3 - Ambiente criado:



Passo 4 - Novo Script:



Passo 5 - Salvando Script:



Pronto, agora o ambiente está preparado para início das atividades.

3.2 Onde obter ajuda

O R é um software livre, o RStudio é um software livre, suas potencialidades resultam da contribuição de inúmeros profissionais ao redor do globo. Por esta razão, quase tudo que se deseja fazer ou conhecer a respeito dessas duas ferramentas está gratuitamente disponível na web. A regra é colaborar!

3.2.1 Fóruns

- [Stackoverflow - EN](#)
- [Stackoverflow - PT](#)

3.2.2 Documentação e tutoriais

- Ajuda das funções, por exemplo: `?sum`, `?head`, `?summary`.
- Documentação dos Pacotes - ex: `ggplot2`
- Vinhetas dos Pacotes - ex: `ggplot2`
- Cheat Sheets

3.2.3 Guia de estilo

- [RStudio's R Style Guide](#)
- [Google's R Style Guide](#)

4 Estruturas de dados no R

4.1 Operadores aritiméticos, operadores lógicos e de comparação

Tabela 1: Operadores matemáticos

Sinal	Operacao
+	Soma
-	Subtração
/	Divisão
*	Multiplicação
^	Potência

Tabela 2: Operadores lógicos e de comparação

Sinal	Operacao
==	Igual
!=	Diferente
<	Menor
>	Maior
<=	Menor igual
>=	Maior igual
&&	E
	Ou
#	Comentário
%in%	Está presente em

4.2 Estruturas de dados

4.2.1 Variáveis e Vetores

```
x <- 1  
x
```

```
[1] 1  
x <- 2  
x
```

```
[1] 2  
x <- c(1, 2)  
x
```

```
[1] 1 2  
is.vector(x)
```

```
[1] TRUE  
length(x)
```

```
[1] 2
```

```

x <- c(1:10)
x

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

x <- seq(1, 50, 5)
x

[1] 1 6 11 16 21 26 31 36 41 46

class(x)

[1] "numeric"

is.numeric(x)

[1] TRUE

x <- c("TCE", "PE", "Recife")
x

[1] "TCE" "PE" "Recife"

x[2]

[1] "PE"

is.numeric(x)

[1] FALSE

class(x)

[1] "character"

is.character(x)

[1] TRUE

x <- c(1, 2, "Recife")
class(x)

[1] "character"

```

4.2.2 Matrices

```

m1 <- matrix( c(1,11,2,22,3,33),
              nrow=3,
              ncol=2,
              byrow=TRUE)
kable(m1)

```

1	11
2	22
3	33

```
class(m1)
```

```
[1] "matrix"
```

```

dim(m1)

[1] 3 2

m1 [2,2]

[1] 22

v1 <- c(1,2,3)
v2 <- c(4,5,6)

m2 <- rbind(v1, v2)
kable(m2)

```

v1	1	2	3
v2	4	5	6

```

dim(m2)

[1] 2 3

```

4.2.3 Listas

```

a <- c(1, 2, 3)
b <- c("a", "b", "c", "d")
c <- c(TRUE, FALSE, TRUE, TRUE)
lista1 <- list(a, b, c)
lista1

[[1]] [1] 1 2 3
[[2]] [1] "a" "b" "c" "d"
[[3]] [1] TRUE FALSE TRUE TRUE
lista1[1]

[[1]] [1] 1 2 3
class(lista1[1])

[1] "list"
lista1[[1]]

[1] 1 2 3
class(lista1[[1]])

[1] "numeric"
lista1[[2]][1] <- "j"
lista1[[2]]

[1] "j" "b" "c" "d"

```

4.2.4 Data Frames

```
bd <- data.frame(id = c(1:5), nivel = c("baixo", "medio", "alto", "medio", "alto"),
                  n.alunos = c(500, 200, 100, 200, 100))
kable(bd)
```

	id	nivel	n.alunos
	1	baixo	500
	2	medio	200
	3	alto	100
	4	medio	200
	5	alto	100

```
dim(bd)
```

```
[1] 5 3
```

```
str(bd)
```

	variable	classe	first_values
	id	integer	1, 2, 3, 4, 5
	nivel	integer	baixo, medio, alto, medio, alto
	n.alunos	double	500, 200, 100, 200, 100

```
head(bd)
```

	id	nivel	n.alunos
	1	baixo	500
	2	medio	200
	3	alto	100
	4	medio	200
	5	alto	100

```
summary(bd)
```

	id	nivel	n.alunos
	Min. :1	alto :2	Min. :100
	1st Qu.:2	baixo:1	1st Qu.:100
	Median :3	medio:2	Median :200
	Mean :3	NA	Mean :220
	3rd Qu.:4	NA	3rd Qu.:200
	Max. :5	NA	Max. :500

```
bd$nivel
```

x
baixo
medio

x
alto
medio
alto

```
class(bd$nivel)
```

[1] "factor"

```
table(bd$nivel)
```

Var1	Freq
alto	2
baixo	1
medio	2

4.2.5 Atividade prática:

```
# O R possui algumas bases de dados para teste. Uma delas é "mtcars".
head(mtcars)

# Com a base de dados mtcars, obtenha:
# Uma descrição dos tipos de variáveis da base
# Um resumo descritivo da base
# O número de dimensões da base
# Imprima a terceira coluna
# Imprima a segunda linha
# O quarto elemento presente na variável "cyl"
```

4.3 Exemplo de uso dos operadores

```
# Soma valores
1+1
```

[1] 2

```
# Soma variáveis
x <- 1
y <- 2
x + y
```

[1] 3

```
x * y
```

[1] 2

```
# Soma vetores
x <- c(1:3)
y <- c(1:3)
x + y
```

```
[1] 2 4 6
sum(x)

[1] 6
sum(x) == sum(y)

[1] TRUE
sum(x + y)

[1] 12
x + 1

[1] 2 3 4
# Comparação
1 == 1

[1] TRUE
x == y

[1] TRUE TRUE TRUE
x != y

[1] FALSE FALSE FALSE
# Comparação
x <- c(1:10) # Atribuindo valores de 1 a 10 a x
x # verificando x

[1] 1 2 3 4 5 6 7 8 9 10
x[(x>8) | (x<5)] # Verificando em x quais elementos são maiores que 8 ou menores que 5

[1] 1 2 3 4 9 10
# Comparação
y <- c(1,2,3) # Atribuindo valores de 1 a 3 a y
x %in% y # Verificando quais elementos de x também pertencem a y

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

5 Importação e exportação de dados

Para uma rápida introdução ao tópico, veja ^ (Outras “Folhas de dicas” podem ser encontradas em: <https://www.rstudio.com/resources/cheatsheets/>): Cheat Sheet: Importação de dados

Neste curso, vamos usar diferentes bases de dados para os exemplos. Porém, a principal será a base de [Microdados Censo Escolar de 2016](#).

5.1 Tipos de arquivos de dados

O R é uma linguagem muito maleável. Praticamente aceita e opera todas as estruturas de dados disponíveis sem grandes problemas. Para que se tenha sucesso na importação ou exportação dos dados, basta que se faça o uso do pacote correto, podendo esse ser encontrado através de consultas na internet e nos links apresentados

no material e nos encontros. De todo modo, para análise de dados em geral é recomendável que se trabalhe com dados em arquivos de extenção consideradas simples, são elas: .txt, .csv, .RData.

5.2 Importação de dados

Os [Microdados Censo Escolar de 2016](#) referem-se a uma série de bases de dados que podem ser carregadas separadamente no R, sendo essa uma de suas principais características. Reconhecendo a importância e potencialidade da linguagem o diretório de disponibilização da base de dados do INEP traz consigo o arquivo “LEIA-ME”, no qual apresenta a melhor maneira de carregar os dados de acordo com o software estatístico que será utilizado. É com base nesse arquivo que faremos a importação dos dados.

5.2.1 Definindo diretório:

Para que o R encontre arquivos solicitados para importação ou para que possa exportar arquivos, é necessário que seja indicado ao programa o diretório correto. Como estamos trabalhando com um ambiente de projeto, o R automaticamente terá como diretório de referência o diretório no qual foi salvo o projeto.

Para fazer a verificação de qual diretório está sendo utilizado, basta usar a função `getwd()`.

```
getwd()
```

```
[1] "/Users/davi/Google Drive/Academico/Cursos/Cursos _ Davi/R _ introduction/UFPE _ 2018/curso _ r _ ufpe"
```

Como este não é o diretório direto no qual estão os dados de interesse, é possível definir novo diretório a partir da função `setwd()`.

```
setwd("./dados/")
```

5.2.2 Abertura de bases pequenas (ESCOLAS e TURMAS):

Com o diretório definido, vamos seguir as orientações do arquivo “LEIA-ME” disponibilizado pelo INEP.

```
setwd("./dados/")
turmas <- read.csv2("TURMAS.csv", sep = "|") # Carregando base de dados
```

Verificando aspectos estruturais da base de dados:

```
dim(turmas) # verificando dimensões da base de dados
names(turmas)[1:10] # verificando nomes das colunas na base de dados
head(turmas[, 1:5]) # verificando as primeiras 6 linhas da base de dados
```

5.2.3 Abertura de bases maiores (MATRÍCULAS e DOCENTES):

Como o arquivo “LEIA-ME” bem indica, o software R, como padrão, trabalha com as bases de dados utilizando a memória RAM do computador. O carregamento de bases muito grandes utilizando a leitura tradicional (como `read.table` ou `read.csv`) pode sobrecarregar o computador, ou mesmo resultar em erro por falta de memória. Dessa forma, para a leitura das bases de MATRÍCULAS e DOCENTES, que possuem mais de 10 milhões de linhas (Brasil), faz-se necessário o uso de pacotes adicionais.

O INEP, portanto, sugere o uso do pacote `ffbase` para trabalhar com essas bases, tendo em vista que o mesmo faz uso do disco rígido ao invés da memória RAM. O pacote `ffbase` armazena a base de dados no R como um objeto da classe `ffdf` - diferentemente da leitura tradicional, que gera um objeto da classe `data.frame`.

O objeto `ffdf` também permite a aplicação de algumas funções – não todas – que são utilizadas com objetos da classe `data.frame` (por exemplo, `table`, `merge` e `transform`). Para aplicação de filtros nas bases `ffdf`, recomendamos o uso da função `ffwhich` (veja a ajuda da função para maiores informações: `?ffwhich`). Além disso, para concatenar as bases de cada região (as bases Docentes e Matrículas estão separadas por região), uma abaixo da outra, é necessário utilizar a função `ffdfappend` (para mais informações: `?ffdfappend`). Informações adicionais estão disponíveis na [ajuda do pacote](#).

```
install.packages("ffbase", dependencies = TRUE) # instalando o pacote
require(ffbase) # carregando o pacote
# definindo diretório
setwd("./dados/")
# carregando base de dados
docentes_ne <- read.csv2.ffdf(file = "DOCENTES_NORDESTE.csv", sep = "|", first.rows=100000)
# verificando estrutura da base de dados
dim(docentes_ne)
docentes_ne[1:5,]
names(docentes_ne)
table.ff(docentes_ne$CO_UF_END)
```

Pronto, agora temos em nosso ambiente de trabalho duas bases de dados: `turmas` e `docentes_ne`.

5.3 Exportação de dados

5.3.1 Arquivos em formato `ffdf`

Em função do tamanho dos arquivos do Microdados do Censo Escolar, o INEP recomenda que as bases de dados muito grandes sejam salvas já no formato padrão do pacote `ffbase`, o formato `ffdf`.

```
# definindo diretório
setwd("./dados/")
save.ffdf(docentes_ne, dir = "./docentes_ne", overwrite = TRUE)
```

Para carregar arquivos no formato `ffdf` pode-se usar a função `load.ffdf`.

```
# definindo diretório
setwd("./dados/")
load.ffdf(dir="./docentes_ne")
```

5.3.2 Arquivos em formato `.RData`

No caso de uma base de dados menor, podemos salvá-la em outros formatos. Um formato muito utilizado é o `.RData`. Utilizando a base `turmas` a seguir faremos um filtro selecionando somente turmas do Estado de PE e salvaremos o resultado do filtro em formato `.RData`.

```
# selecionando linhas da base nas quais CO_UF == 26
turmas_pe <- subset(turmas, turmas$CO_UF == "26")
# comparando as bases
dim(turmas)
dim(turmas_pe)
# definindo diretório
setwd("./dados/")
# salvando nova base
save(turmas_pe, file ="turmas_pe.RData")
```

Para carregar o arquivo salvo, basta:

```
# definindo diretório
setwd("./dados/")
load("turmas_pe.RData") # Carregando base de dados
dim(turmas_pe) # verificando dimensões da base de dados
names(turmas_pe) # nomes das colunas da base de dados
head(turmas_pe) # início da base de dados
```

6 Atividade Prática - Git Commit / Push / Pull

- Assista ao Webinar [Managing Part 2](#). Faça o versionamento do seu projeto e crie um repositório público. Veja como fazer o `commit` e o `push` para o repositório. Veja também como importar (`pull`) o repositório.

7 Atividade Prática - R

- Inicie o RStudio, abra um novo R Script declare duas variáveis (`x` e `y`), atribua valores numéricos a elas e o resultado de sua soma à variável `z`.
- No mesmo R script, abra a base de dados de docentes do nordeste disponíveis no microdados do censo atribuindo-a ao objeto `bd` e obtenha a média de idade (`NU_IDADE`) com a função `mean` conforme o exemplo abaixo.

```
mean(cars$dist)
```

```
## [1] 42.98
```

8 Links úteis para o próximo encontro

- Cheat Sheet: Transformação de dados
- Cheat Sheet: Tidyverse