



# Aula 2

Semabio 2024

Carolina Musso

IB/UnB





# Sobre o curso

- Estrutura:
  - Aula 1: Instalação e Intro Rmarkdown
  - Aula 2 (hoje): Rmarkdown, pacote dplyr
  - Aula 3: Gráficos ggplot, Tabelas resumo e testes estatísticos

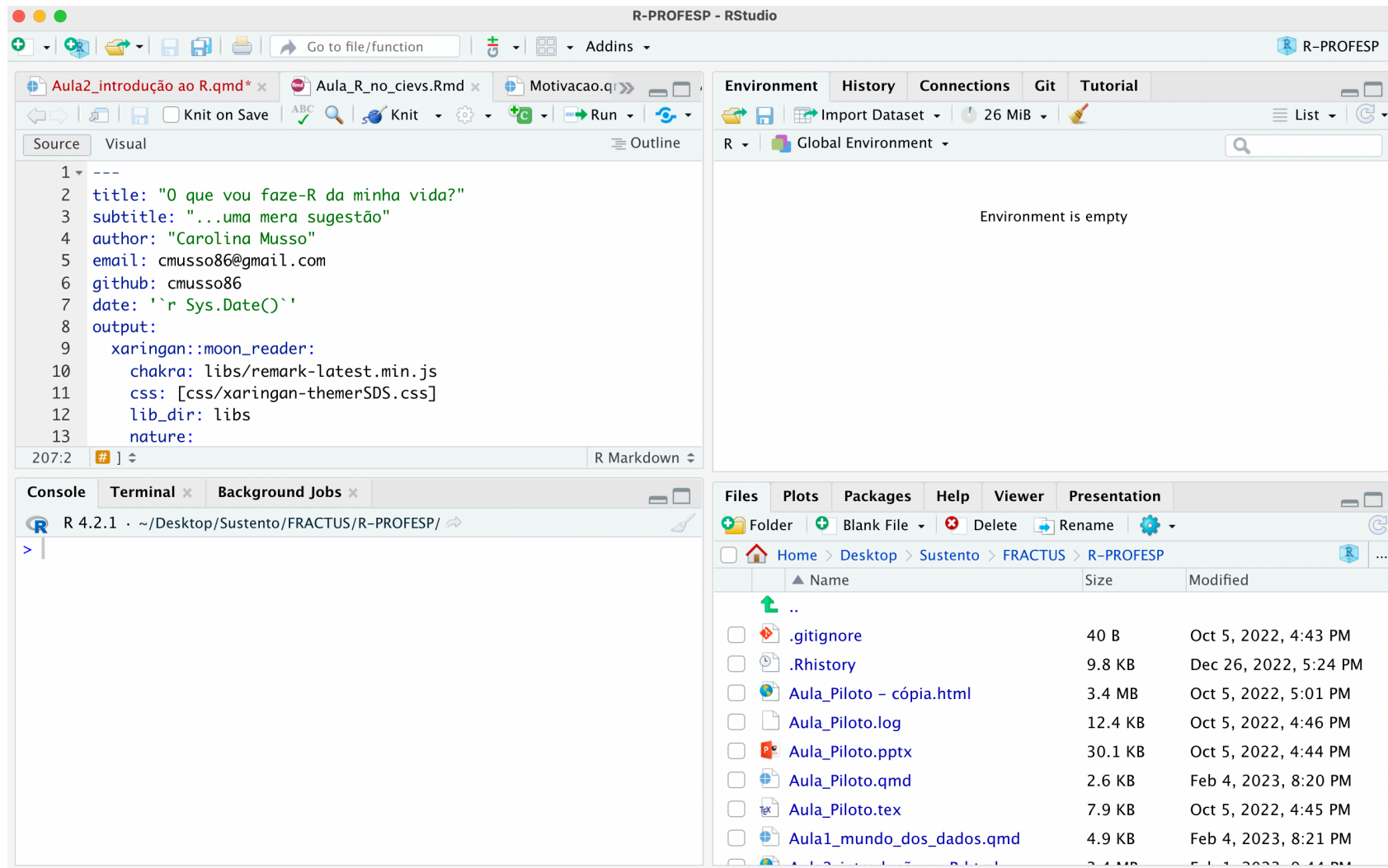


# Quiz





# Os Quatro Paineis do R-Studio







# Rmarkdown



- Rmarkdown

- Quarto



- YiHui



# R de trás-para-frente

- Fizemos o download de um código pronto.
- Analisamos, rodamos, sem “entender”!



# Você conseguiu diferenciar essas partes?

- Cabeçalho
- Texto
- Código

```
1 ---
2 title: "Meu primeiro arquivo automático"
3 author: "Meu nome"
4 date: "`r format(Sys.Date(), '%d/%m/%Y')`"
5 output:
6   word_document: default
7   html_document: default
8 ---
9
10 ```{r sets, echo=F, message=F, warning=F }
11 # COMENTÁRIOS INICIAIS ----
12
13 # tá vendo que eu estou escrevendo em uma parte que tem um fundo cinza? Isso é chamado de "chunk" que
14 # significa pedaço em inglês. Um "chunk" é onde a gente vai escrever os códigos de R mesmo. Depois vamos
15 # voltar a esses assuntos com mais detalhes!
16
17 # PRE-SETS ----
18 rm(list=ls()) # limpa o ambiente
19
20 pasta=paste0("Saidas-Exercicios-", Sys.Date()) #cria uma pasta para as saídas do dia
21 if(!dir.exists(pasta)) dir.create(pasta) #cria uma pasta para as saídas do dia
22
23
24
25
26
27
28
29
30
31 ```{r pacotes, echo=F, message=F, warning=F }
32
33
34
35
36
37 ## Aqui é o subtítulo que você quiser
38
39 Daí, aqui você escreve o que você achar interessante sobre os dados. Pode interpretar, enfim...
40
```



# Cabeçalho

- Metadados
- Linguagem YAML
- É MUITO sensível (qualquer coisa atrapalha)

```
1 |---
2 |title: "Meu primeiro arquivo automático"
3 |author: "Meu nome"
4 |date: "`r format(Sys.Date()), '%d/%m/%Y'`"
5 |output:
6 |  word_document: default
7 |  html_document: default
8 |---
9 |
```





# Texto

- Markdown

`## Aqui é o subtítulo que você quiser`

Daí, aqui você escreve o que você achar interessante sobre os dados.

`## Número de casos acumulados`



# Código

- É o coração do R mesmo
  - Linguagem de programação
  - Como que difere de Excel (apontar e clicar?)

```
```${r pacotes, echo=F, message=F, warning=F }  
# PACOTES ----  
  
# agora vou caggar os pacotes que vamos usar usando uma função especial. Não se preocupe em entender  
isso agora. Apenas rode! Vai instalar os pacotes que não tiver instalados no seu computador. E carregar  
todos.  
  
if (!require(pacman)) install.packages("pacman")  
  pacman::p_load(tidyverse, data.table, flextable, officer)  
```${pre>
```



# O que são esses chunks afinal?

- O “código” propriamente dito
- As partes dos chunks
  - cabeçalho (do chunk)
  - conteúdo
    - código
    - comentários



# Brincando com o código



- Mudar de cor, posição ...
- Prática





# Pacotes



# E os pacotes o que são?

- Extensões do R
- Teremos uma aula só sobre o uso de pacotes
  - CRAN
  - Vamos usar muito o tidyverse
- EpiR-Handbook
- Instalação de pacotes, Curso-R
- Tidyverse, R4DS



# Instalar e Carregar

- Para utilizar um pacote é preciso **instalá-lo** (uma única vez) e depois **carregá-lo** (sempre que for usar)

## Jeito clássico

```
1 # para instalar
2 install.packages("tidyverse") # precisa estar entre aspas
3
4 # para carregar
5 library(tidyverse) #pode estar entre aspas ou não
```



# Pacote Pacman

- Mas nós vamos fazer de outro jeito!
- Vamos usar um pacote que gerencia pacotes (eu disse que tinha pacotes para tudo!)

```
1 if (!require("pacman")) install.packages("pacman")  
2 pacman::p_load("tidyverse")
```

- Verifica a instalação e carrega! : Garantir reprodutibilidade e automatização.
- O pacote é o pacman, a função que faz isso é o p\_load.





# O que são funções

- O pacote é um conjunto de funções... mas o que são funções?
- “Funções em programação é um conjunto de instruções para simplificar uma tarefa repetitiva”
- precisam de **argumentos**.

```
1 sqrt(9)
```

```
[1] 3
```

```
1 round(3.89, digits = 0 )
```

```
[1] 4
```

```
1 round(3.89, 0 )
```

```
[1] 4
```



# O pacote tidyverse





# RBase

```
1 head(cars[cars$speed>7,],3)
```

	speed	dist
5	8	16
6	9	10
7	10	18

## Com o tidyverse

```
1 cars %>% #isso chama pipe  
2   filter(speed>7) %>%  
3   head(3)
```

	speed	dist
1	8	16
2	9	10
3	10	18



Objetos, funções,  
variáveis e pacotes





# Erros comuns

- Erros comuns
- Pedindo Ajuda
- Getting Help
- Palestra Latin-R: Conquistando Errores en R, material



# Google Forms



# Parte 2: Pacote dplyr



# Leituras para aprofundamento

- EpiHandbook, 8. Cleaning data and core functions
- Curso-R, 7. Manipulação
- DataCamp, Introduction to the Tidyverse
- R4DS, 5. Data transformation





# Pacote do tidyverse

- Vamos dar preferência para funções desse pacote
  - do pacote dplyr que faz parte do tidyverse

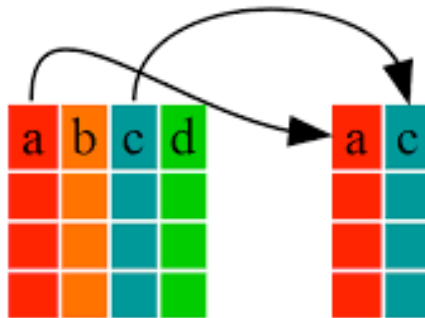




# select()

- Trabalhando com as colunas
- As vezes queremos selecionar apenas algumas colunas (pode ser pesado e confuso manter todas as colunas sempre)

`select(data.frame,a,c)`



`Sepal.Length Sepal.Width Petal.Length Petal.Width Species`

1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa



# Função select()

- Vimos que as funções são intruções: muitas vezes tem cara de verbo
- a função select precisa de 2 argumentos: a base de dados e as colunas que você quer selecionar

```
1 # dplyr
2
3 select(iris, Sepal.Length, Sepal.Width) %>%
4   head(5)
```

	Sepal.Length	Sepal.Width
1	5.1	3.5
2	4.9	3.0
3	4.7	3.2
4	4.6	3.1
5	5.0	3.6

```
1 # dplyr
2 select(iris, starts_with("Sepal")) %>%
3   head(5)
```

---

	Sepal.Length	Sepal.Width
1	5.1	3.5
2	4.9	3.0
3	4.7	3.2
4	4.6	3.1
5	5.0	3.6

```
1 # R base
2
3 iris[,c("Sepal.Length", "Sepal.Width")][1:6,]
```

	Sepal.Length	Sepal.Width
1	5.1	3.5
2	4.9	3.0
3	4.7	3.2
4	4.6	3.1
5	5.0	3.6
6	5.4	3.9





# Função filter()

- Trabalhando com as linhas
- As vezes queremos selecionar apenas algumas linhas: algum grupo específico.
- Raciocínio parecido com a função select(): precisa de 2 argumentos: a base de dados e a *condição sobre alguma coluna* para fazer o filtro.

# Subset Observations (Rows)



```
1 # dplyr
2 iris %>% # outra forma de fazer
3   filter(Species == "setosa") %>%
4   select(starts_with("Petal"), Species) %>%
5   head(5)
```

	Petal.Length	Petal.Width	Species
1	1.4	0.2	setosa
2	1.4	0.2	setosa
3	1.3	0.2	setosa
4	1.5	0.2	setosa
5	1.4	0.2	setosa

```
1 #R base
2 iris[iris$Species=="setosa", ]
```

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
--	--------------	-------------	--------------	-------------	---------


1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa
9	4.4	2.9	1.4	0.2	setosa
10	4.9	3.1	1.5	0.1	setosa
11	5.4	3.7	1.5	0.2	setosa
12	4.8	3.4	1.6	0.2	setosa
13	4.8	3.0	1.4	0.1	setosa
14	4.3	3.0	1.1	0.1	setosa



# mutate()

- criar novas variáveis

storm	wind	pressure	date	
Alberto	110	1007	2000-08-12	
Alex	45	1009	1998-07-30	
Allison	65	1005	1995-06-04	
Ana	40	1013	1997-07-01	
Arlene	50	1010	1999-06-13	
Arthur	45	1010	1996-06-21	



storm	wind	pressure	date	ratio
Alberto	110	1007	2000-08-12	9.15
Alex	45	1009	1998-07-30	22.42
Allison	65	1005	1995-06-04	15.46
Ana	40	1013	1997-07-01	25.32
Arlene	50	1010	1999-06-13	20.20
Arthur	45	1010	1996-06-21	22.44

```
1 # dplyr
2
3 iris %>%
4   mutate(area = Petal.Length*Petal.Width/2) %>%
5   select(area) %>%
6   head(5)
```

area

```
1 0.14  
2 0.14  
3 0.13  
4 0.15  
5 0.14
```

```
1 # R base  
2 iris$area <- iris$Petal.Length*iris$Petal.Width/2
```



# group\_by()

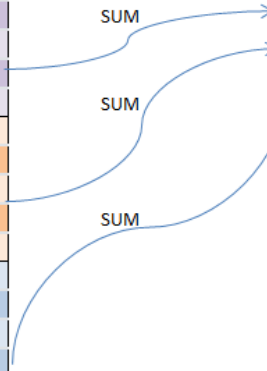
- sempre agrupar e fazer alguma conta!

```
1 # dplyr
2 iris %>%
3   mutate(area = Petal.Length*Petal.Width/2) %>%
4   group_by(Species) %>%
5   summarise(media=mean(area))
```

```
# A tibble: 3 × 2
  Species    media
  <fct>      <dbl>
1 setosa      0.183
2 versicolor  2.86
3 virginica   5.65
```



sepal.length	sepal.width	variety
5.1	3.5	Setosa
4.9	3	Setosa
4.7	3.2	Setosa
4.6	3.1	Setosa
5	3.6	Setosa
7	3.2	Versicolor
6.4	3.2	Versicolor
6.9	3.1	Versicolor
5.5	2.3	Versicolor
6.5	2.8	Versicolor
6.3	3.3	Virginica
5.8	2.7	Virginica
7.1	3	Virginica
6.3	2.9	Virginica
6.5	3	Virginica
7.6	3	Virginica
4.9	2.5	Virginica



variety	sepal.length	sepal.width
Setosa	24.3	16.4
Versicolor	32.3	14.6
Virginica	44.5	20.4



# Desafio RBase



**Até amanhã!**

