

# Aula 2

Renato Rodrigues Silva

# Objetivo da aula

- O objetivo da segunda aula é ensinar ao aluno:
- O layout do RStudio;
- Comandos básico do R.

# Roteiro da aula

- Layout do RStudio;
- Como utilizar o R como calculadora
- Principais objetos no R
  - vector;
  - matrix;
  - arrays;
  - data.frame;
  - lists

# Layout do RStudio

The screenshot displays the RStudio interface with three main panes highlighted by red boxes:

- Source Pane:** Contains an R script named `ggplot2.R` with the following code:

```
1 library(ggplot2)
2 mpg_plot <- ggplot(mpg, aes(x = displ, y = hwy)) +
3   geom_point(aes(colour = class))
4
5 mpg_plot
6
```
- Environment Pane:** Shows the `Global Environment` with a table of objects:

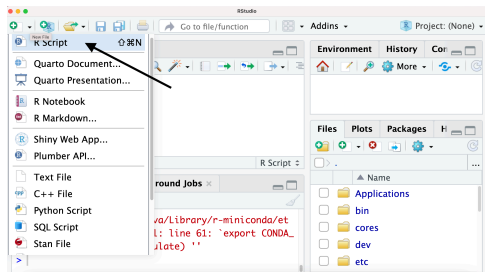
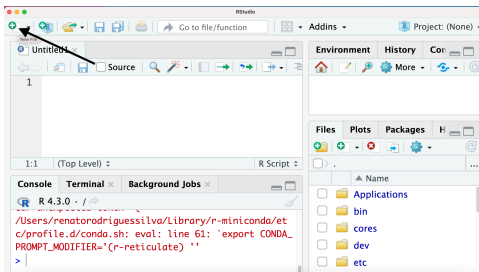
Name	Type	Len...	Size	Value
mpg_plot	gg	9	29.1...	List of 9
- Plots Pane:** Displays a scatter plot of `hwy` vs `displ` colored by `class`. The legend on the right lists the classes: 2seater, compact, midsize, minivan, pickup, subcompact, and suv.

The **Console** pane at the bottom shows the execution of the same code as the Source pane.

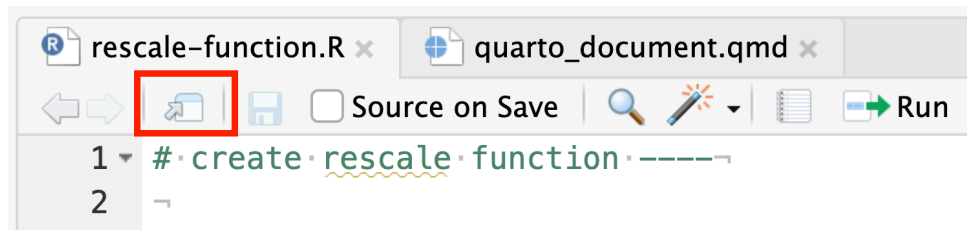
# Painel “Source”

- O painel “source” permite o usuário ver e editar vários arquivos relacionados a códigos fonte de linguagem de programação, tais como: .R, .rmd, .qmd, .py, entre outros.

# Criando novos arquivos de código-fonte



# Destacar a janela



# Painel Console

- O painel console provê uma área para executar o código interativamente.
- Por default, o painel console é o software R, mas é possível ter um console python ou até mesmo o terminal.
- Nesse curso, abordaremos somente o software R.



# Painel “Environment”

- O painel “environment” é composto pelas abas: “environment”, “history”, “connections”, “tutorial” e “presentation”.
- Nesse curso, abordaremos apenas as abas “environment” e “history”

# Aba Environment

- A aba de ambiente exhibe objetos R e Python salvos atualmente.
- Ainda oferece:
  - Carregamento ou salvamento de espaços de trabalho do R,
  - Importação interativa de conjuntos de dados de arquivos de texto, Excel ou SPSS/SAS/Stata.
  - Memória atualmente utilizada pela sessão ativa do R.
  - Ícone de vassoura para remover todos os objetos do ambiente atual.

# Aba Environment

The screenshot displays the RStudio interface. The top-left pane shows a script file named 'Untitled1\*' with the following code:

```
1 a = 1 + 1
2
3 b = 2
```

The bottom-left pane shows the R console with the following output:

```
> a = 1 + 1
>
> b = 2
>
```

The top-right pane shows the Environment pane, which displays the current environment (Global Environment) and the values of the objects created in the script. The Environment pane has tabs for Environment, History, and Connections. The Environment tab is active, showing a memory usage of 70 MiB and a button to 'Remove objetos da memória'. Below this, the 'Values' section shows the following table:

Valores dos objetos	
a	2
b	2

The bottom-right pane shows the Files, Plots, Packages, Help, and View tabs. The Files tab is active, showing a list of files and a button to 'Export'.

# Aba Environment

The screenshot shows the RStudio IDE interface. The top toolbar includes icons for file operations and a search bar. The main editor window displays a script with two lines of code: `a = 1 + 1` and `b = 2`. The Environment pane on the right shows the Global Environment with two objects, `a` and `b`, both having a value of 2. A confirmation dialog titled "Confirm Remove Objects" is overlaid on the interface, asking if the user is sure they want to remove all objects from the environment. The dialog includes a warning icon, the text "Are you sure you want to remove all objects from the environment? This operation cannot be undone.", and a checkbox labeled "Include hidden objects" which is checked. At the bottom of the dialog are "Yes" and "No" buttons.

RStudio

Go to file/function

Addins

Project: (None)

Untitled1\*

Source

```
1 a = 1 + 1
2
3 b = 2
```

3:6 (Top Level)

Environment History Connections

61 MiB

R Global Environment

Values

a	2
b	2

Console Terminal Background

R 4.3.0

```
> a = 1 + 1
>
> b = 2
>
```

Confirm Remove Objects

Are you sure you want to remove all objects from the environment? This operation cannot be undone.

☒ Include hidden objects

Yes No

# Aba Environment

The screenshot displays the RStudio environment with the following components:

- Top Bar:** Includes standard window controls, a toolbar with icons for file operations, a "Go to file/function" search bar, an "Addins" menu, and a "Project: (None)" dropdown.
- Source Editor:** The main workspace for editing R scripts. It shows a file named "Untitled1\*" with the following code:

```
1 a = 1 + 1
2
3 b = 2
```

The code is currently in "Source" view. The status bar at the bottom of this pane indicates "3:6" and "(Top Level)".
- Console:** Located at the bottom left, it shows the execution history of the script. The commands and their outputs are:

```
> a = 1 + 1
>
> b = 2
>
```

The console title bar includes tabs for "Console", "Terminal", and "Background Jobs".
- Environment Panel:** Located on the right side, it has tabs for "Environment", "History", and "Connections". The "Environment" tab is active, showing "R" and "Global Environment" with a memory usage of "61 MiB". The text "Environment is empty" is displayed in the main area.
- Bottom Bar:** Contains tabs for "Files", "Plots", "Packages", "Help", and "View". The "Files" tab is active, showing a toolbar with navigation and zoom controls.

# A aba “History”

- Exibe os comandos que foram executados na sessão atual junto com a funcionalidade de pesquisa.
- Existem botões para carregar/salvar o histórico de comandos em um arquivo,
- Além de enviar o comando selecionado para o console ou inseri-lo no documento de trabalho atual.
- Há um botão de exclusão para remover o histórico selecionado
- Ou há um botão de vassoura para apagar todo o histórico da sessão atual.

# Aba "History"

The screenshot displays the RStudio application window. The top toolbar includes icons for creating a new file, opening a file, saving, and other standard file operations. The main interface is divided into several panes:

- Source Editor:** Contains a script named "Untitled1\*" with the following code:

```
1 a = 1 + 1
2
3 b = 2
```
- Environment Pane:** Shows the current environment with variables `a = 1 + 1` and `b = 2`.
- History Pane:** This pane is active and shows the execution history of the script, displaying the same code as the Source Editor.
- Console:** Shows the R prompt and the output of the executed code:

```
> a = 1 + 1
>
> b = 2
>
```
- Files Pane:** Displays the file explorer, showing the current directory structure with folders like "Applications" and "bin".

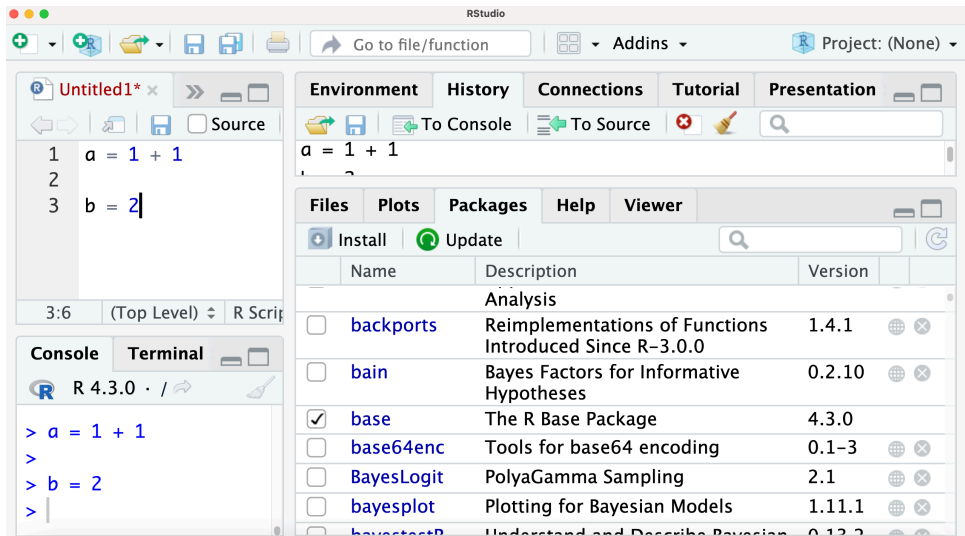
The bottom status bar indicates the R version is 4.3.0 and the current file is "R Script".

# Painel Output

- Esse painel é composto pelas abas: files, plots, packages, help and viewer.
  - Aba Files oferece exploração interativa do projeto atual do R junto com todo o diretório.
  - Aba Plots exibe imagens estáticas geradas pelo código.
  - Aba Packages permite visualizar os pacotes R atualmente instalados e possui uma barra de pesquisa para buscar na biblioteca atual de pacotes.
  - Aba Help é usada para exibir a documentação dos pacotes.
  - Aba Viewer é usada para mostrar o conteúdo de aplicações em Web, como Shiny Apps.



# Aba "Packages"



The screenshot shows the RStudio interface with the 'Packages' pane active. The editor window on the left contains the following R code:

```
1 a = 1 + 1
2
3 b = 2
```

The console at the bottom left shows the execution of the code:

```
> a = 1 + 1
>
> b = 2
>
```

The 'Packages' pane on the right displays a table of installed and available packages. The 'base' package is checked as installed.

	Name	Description	Version	
Analysis				
<input type="checkbox"/>	backports	Reimplementations of Functions Introduced Since R-3.0.0	1.4.1	
<input type="checkbox"/>	bain	Bayes Factors for Informative Hypotheses	0.2.10	
<input checked="" type="checkbox"/>	base	The R Base Package	4.3.0	
<input type="checkbox"/>	base64enc	Tools for base64 encoding	0.1-3	
<input type="checkbox"/>	BayesLogit	PolyaGamma Sampling	2.1	
<input type="checkbox"/>	bayesplot	Plotting for Bayesian Models	1.11.1	
<input type="checkbox"/>	bayestestR	Understand and Describe Bayesian	0.12.2	

# A aba “Help”

The screenshot displays the RStudio application window. The top toolbar includes icons for file operations and a search bar labeled 'Go to file/function'. The main editor area shows a script file named 'Untitled1\*' with the code `a = 1 + 1` on line 1. The console area at the bottom left shows the command `> ?sum` and the cursor on the next line. The right-hand pane is divided into several tabs: 'Environment', 'History', 'Connection', 'Files', 'Plots', 'Packages', and 'Help'. The 'Help' tab is active, displaying the documentation for the `sum` function from the 'base' package. The title of the documentation is 'Sum of Vector Elements', and the section 'Description' is visible at the bottom of the pane.

RStudio

Go to file/function

Project: (None)

Untitled1\* x test.Rhistory x

Source

```
1 a = 1 + 1
2
```

3:6 (Top Level) R Script

Console Terminal x Background Jobs x

R 4.3.0 · /

```
> ?sum
> |
```

Environment History Connection

To Console To Source

```
a = 1 + 1
```

Files Plots Packages Help

R: Sum of Vector Elements Find in Topic

sum {base} R Documentation

## Sum of Vector Elements

### Description

# R como calculadora

O software R é baseado em linhas de comandos. A seguir vamos mostrar como podemos fazer operações básicas de matemática.

```
#Adição
```

```
1 + 1
```

```
[1] 2
```

```
#Subtração
```

```
2 - 1
```

```
[1] 1
```

```
#Multiplicação
```

```
3 * 2
```

```
[1] 6
```

```
#Divisão
```

# R como calculadora

Observe como o R responde em algumas situações

```
#Divisão por zero
```

```
1 / 0
```

```
[1] Inf
```

```
#Raiz quadrada para números negativos
```

```
sqrt(-1)
```

```
[1] NaN
```

```
#Arredondar número para seis casas decimais
```

```
round(pi,6)
```

```
[1] 3.141593
```

# Criar objetos no R

- Os nomes de objetos devem começar com uma letra. Eles podem conter somente letras, bem como podem conter letras, números, `_` e `..`

```
x = 3
```

```
y = 2
```

```
x + y
```

```
[1] 5
```

Conjunto de elementos do mesmo tipo (logical, numeric, integer, double character)

# Formas de declarar vetores

- A forma mais simples de se criar um vetor é usar a função de concatenação "c()".

```
value.num = c(3,4,2,6,20)
```

```
value.num
```

```
[1] 3 4 2 6 20
```

```
value.char = c("koala", "kangaroo")
```

```
value.char
```

```
[1] "koala"      "kangaroo"
```

```
value.logical = c(FALSE, FALSE, TRUE, TRUE)
```

# Formas de declarar vetores

- Segunda maneira de criar vetor no R: usando a função “scan”

```
values = scan(text="
2
3
4
5"
)
```

```
values
```

```
[1] 2 3 4 5
```



# Formas de declarar vetores

- Outra opção usando comando “rep”

```
rep(1,5)
```

```
[1] 1 1 1 1 1
```

```
rep(c(1,2),3)
```

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

```
rep(c(1,6),each=3)
```

```
[1] 1 1 1 6 6 6
```

```
rep(c(1,6),c(3,5))
```

```
[1] 1 1 1 6 6 6 6 6
```

# Formas de declarar vetores

- Outra opção usando comando “seq”

```
seq(from=1,to=5)
```

```
[1] 1 2 3 4 5
```

```
seq(from=1, to=5, by=0.1)
```

```
[1] 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0 2.1 2.2 2.3 2.4  
[20] 2.9 3.0 3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 4.0 4.1 4.2 4.3  
[39] 4.8 4.9 5.0
```

```
seq(from=1, to=5, length=10)
```

```
[1] 1.000000 1.444444 1.888889 2.333333 2.777778 3.222222 3.666667  
[9] 4.555556 5.000000
```

```
rep(seq(from=1, to=5, length=10),each=2)
```

# Formas de declarar vetores

- Outra opção usando comando ":"

```
1:5
```

```
[1] 1 2 3 4 5
```

```
c(1:5,10)
```

```
[1] 1 2 3 4 5 10
```

# Operações com Vetores

```
x = 1:4
```

```
y = 5:8
```

```
x + y
```

```
[1] 6 8 10 12
```

```
2 * x + 1
```

```
[1] 3 5 7 9
```

```
x * y
```

```
[1] 5 12 21 32
```

```
x / y
```

# Matriz

- Conjunto de elementos dispostos em linhas e colunas, em que todos os elementos são do mesmo tipo

```
mat.num = matrix(c(1:16),4,4)
```

```
mat.num
```

	[,1]	[,2]	[,3]	[,4]
[1,]	1	5	9	13
[2,]	2	6	10	14
[3,]	3	7	11	15
[4,]	4	8	12	16

```
mat.char = matrix(LETTERS[1:4],2,2)
```

```
mat.char
```

# Manipulando Matrizes

```
#Criando nomes para as linhas de uma matriz
rownames(mat.num) = c("Sao Paulo", "Americana", "Piracicaba", "Ma

colnames(mat.num) = 1:4

mat.num
```

	1	2	3	4
Sao Paulo	1	5	9	13
Americana	2	6	10	14
Piracicaba	3	7	11	15
Madson	4	8	12	16

# Multiplicação elemento a elemento

```
mat.num2 = diag(seq(10,40,by=10))
```

```
mat.num2
```

	[,1]	[,2]	[,3]	[,4]
[1,]	10	0	0	0
[2,]	0	20	0	0
[3,]	0	0	30	0
[4,]	0	0	0	40

```
mat.num3 = mat.num * mat.num2
```

```
mat.num3
```

	1	2	3	4
Sao Paulo	10	0	0	0

# Acessando elementos das matrizes

```
#Um elemento
```

```
mat.num[1,1]
```

```
[1] 1
```

```
#Linhas
```

```
mat.num[1,]
```

```
1  2  3  4
```

```
1  5  9 13
```

```
#Colunas
```

```
mat.num[,3]
```

Sao Paulo	Americana	Piracicaba	Madson
9	10	11	12

```
#Sub Matrizes
```



# Data.frames

São Similares as matrizes no entanto permite que as colunas tenham diferentes tipos

```
data(iris)
```

```
iris
```

	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

# Manipulação de Data.frames

```
iris$Sepal.Length
```

```
[1] 5.1 4.9 4.7 4.6 5.0 5.4 4.6 5.0 4.4 4.9 5.4 4.8 4.8 4.3 5.8  
[19] 5.7 5.1 5.4 5.1 4.6 5.1 4.8 5.0 5.0 5.2 5.2 4.7 4.8 5.4 5.2  
[37] 5.5 4.9 4.4 5.1 5.0 4.5 4.4 5.0 5.1 4.8 5.1 4.6 5.3 5.0 7.0  
[55] 6.5 5.7 6.3 4.9 6.6 5.2 5.0 5.9 6.0 6.1 5.6 6.7 5.6 5.8 6.2  
[73] 6.3 6.1 6.4 6.6 6.8 6.7 6.0 5.7 5.5 5.5 5.8 6.0 5.4 6.0 6.7  
[91] 5.5 6.1 5.8 5.0 5.6 5.7 5.7 6.2 5.1 5.7 6.3 5.8 7.1 6.3 6.5  
[109] 6.7 7.2 6.5 6.4 6.8 5.7 5.8 6.4 6.5 7.7 7.7 6.0 6.9 5.6 7.7  
[127] 6.2 6.1 6.4 7.2 7.4 7.9 6.4 6.3 6.1 7.7 6.3 6.4 6.0 6.9 6.7  
[145] 6.7 6.7 6.3 6.5 6.2 5.9
```

# Manipulação de Data.frames

```
iris$Renato = TRUE
```

```
iris
```

	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

# List

Generalização dos vetores no sentido que uma lista é uma coleção de objetos

```
A = list(x = 1:4, y = matrix(1:4,2,2), v = list(A=4,B=5))
```

A

\$x

```
[1] 1 2 3 4
```

\$y

	[,1]	[,2]
[1,]	1	3
[2,]	2	4

\$v

\$v\$A

# Manipulação de objetos tipo List

```
A[[1]]
```

```
[1] 1 2 3 4
```

```
A[[3]]
```

```
$A
```

```
[1] 4
```

```
$B
```

```
[1] 5
```

```
A$x
```

```
[1] 1 2 3 4
```

```
A$y
```

# Manipulação de objetos tipo List

```
B = list(s = 1:5, r = 2)
```

```
Q = c(A,B)
```

```
Q
```

```
$x
```

```
[1] 1 2 3 4
```

```
$y
```

	[,1]	[,2]
[1,]	1	3
[2,]	2	4

```
$v
```