

# Linguagem R

## Introdução e Primeiros Passos

Tiago Matos  
(98) 98812-9741  
jersitonmattos@hotmail.com



# Sumário

1. Introdução ao R;
2. Download do R e Rstudio + Rstudio Cloud;
3. Introdução ao Ambiente;
4. Introdução à Linguagem;
5. Aplicações

# Introdução ao R

# O que é a Linguagem R?

- R é uma linguagem de programação multi-paradigma voltada à manipulação, análise e visualização de dados, abaixo temos os principais paradigmas abrangidos pela linguagem:
  - orientada a objetos;
  - programação funcional;
  - linguagem dinâmica;
  - fracamente tipada.
- Criada em 1993 por Ross Ihaka e Robert Gentleman no Departamento de Estatística da Universidade de Auckland, Nova Zelândia;
- A linguagem R é largamente usada entre estatísticos e analistas de dados para desenvolver software de estatística e análise de dados. Pesquisas e levantamentos com profissionais da área mostram que a popularidade do R aumentou substancialmente nos últimos anos.

# Por que aprender R?

- Para trabalhar com dados e realizar análises estatísticas complexas;
- Vantagens da Linguagem R:
  - comunidade e recursos;
  - distribuição de ideias;
  - Integração com outras tecnologias.

# Download do R e Rstudio + Rstudio Cloud

# Onde encontrar a Linguagem R?

- A linguagem R pode ser baixada e instalada gratuitamente a partir do site oficial do projeto R (<https://www.r-project.org/>). O site fornece links para download da versão mais recente do R para diferentes sistemas operacionais, como Windows, macOS e Linux.
- É altamente recomendável usar um ambiente de desenvolvimento integrado (IDE) para facilitar o desenvolvimento em R. Um dos IDEs mais populares para R é o RStudio, que pode ser baixado gratuitamente em seu site oficial (<https://www.rstudio.com/>).
- No RStudio Cloud, é possível criar, editar e executar projetos R diretamente no seu navegador, sem a necessidade de configurar um ambiente local. Ele oferece uma experiência semelhante ao RStudio desktop, com a vantagem de permitir que você acesse seus projetos de qualquer lugar, desde que tenha uma conexão à internet. Acesse o site do RStudio Cloud em <https://rstudio.cloud/>.

# Download do R

1. Acesso: <https://cran.r-project.org/>

The Comprehensive R Archive Network

### Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux \(Debian, Fedora/Redhat, Ubuntu\)](#)
- [Download R for macOS](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

### Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2023-04-21, Already Tomorrow) [R-4.3.0.tar.gz](#), read [what's new](#) in the latest version.
- Sources of [R alpha and beta releases](#) (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are [available here](#). Please read about [new features and bug fixes](#) before filing corresponding feature requests or bug reports.
- Source code of older versions of R is [available here](#).
- Contributed extension [packages](#)

### Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.



# Download do R

2. Clique em: “instalar R pela primeira vez” e depois em “Download”.

Subdirectories:

<a href="#">base</a>	Binaries for base distribution. This is what you want to <a href="#">install R for the first time</a> .
<a href="#">contrib</a>	Binaries of contributed CRAN packages (for R $\geq$ 3.4.x).
<a href="#">old contrib</a>	Binaries of contributed CRAN packages for outdated versions of R (for R $<$ 3.4.x).
<a href="#">Rtools</a>	Tools to build R and R packages. This is what you want to build your own packages on Windows, or to build R itself.

Please do not submit binaries to CRAN. Package developers might want to contact Uwe Ligges directly in case of questions / suggestions related to Windows binaries.

You may also want to read the [R FAQ](#) and [R for Windows FAQ](#).

Note: CRAN does some checks on these binaries for viruses, but cannot give guarantees. Use the normal precautions with downloaded executables.

# Download do R Studio

1. Acesse: <https://posit.co/download/rstudio-desktop/> e escolha a versão compatível com o seu sistema operacional:

OS	Download	Size	SHA-256
Windows 10/11	<a href="#">RSTUDIO-2023.03.1-446.EXE</a> ⬇	208.07 MB	<a href="#">8DCB7103</a>
macOS 11+	<a href="#">RSTUDIO-2023.03.1-446.DMG</a> ⬇	381.89 MB	<a href="#">1AB19F28</a>
Ubuntu 18/Debian 10	<a href="#">RSTUDIO-2023.03.1-446-AMD64.DEB</a> ⬇	137.83 MB	<a href="#">2E803082</a>
Ubuntu 20/Debian 11	<a href="#">RSTUDIO-2023.03.1-446-AMD64.DEB</a> ⬇	137.83 MB	<a href="#">2E803082</a>

# Introdução ao Ambiente

RStudio

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Untitled1 x

Source on Save

Run Source

## Área de Scripts

1. Autocomplete;
2. Verificação de erros;
3. Destaque de sintaxe.

1:1 (Top Level) R Script

Console Terminal x Jobs x

R 4.1.1 · ~/r-scripts/

R version 4.1.1 (2021-08-10) -- "Kick Things"  
Copyright (c) 2021 The R Foundation for Statistical Computing  
Platform: x86\_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.

> |

## Console

1. Execução de códigos;
2. Visualização de erros;
3. Tarefas em segundo plano.

Project: (None)

Environment History Connections Tutorial

Import Dataset 135 MiB

R Global Environment

Environment is empty

## Ambiente de Trabalho

1. Visualizar objetos e variáveis;
2. Interagir com os objetos e variáveis;
3. Gerenciar conexões de banco de dados.

Files Plots Packages Help Viewer

Zoom Export

## Visualização de Plots

1. Mostrar gráficos, páginas da web, aplicativos *Shiny*;
2. Seção de ajuda;
3. Gerenciamento de pacotes;
4. Gerenciamento de arquivos.

# Introdução à Linguagem

# Tipos de Dados

- Dados são átomos de informação.
- Os tipos mais comuns de dados em R são:
  - Vetores (vector): usado para armazenar uma sequência de valores do mesmo tipo;
  - Listas (list): usadas para armazenar diferentes tipos de dados em uma única estrutura. Uma lista pode conter vetores, matrizes, data frames ou até mesmo outras listas;
  - Dataframes (data.frame): usado para armazenar dados tabulares em formato de tabela, onde as colunas podem ser de diferentes tipos de dados.
- Outros tipos:
  - Números, Inteiros, Caracteres, Lógicos, Complexos, Matrizes, POSIXct, Fatores, Datas e os tipos únicos criados pela comunidade.

# Tipos de Dados

```
# vetor
```

```
x ← c(1, 2, 3, 4, 5, 6)
```

```
# lista
```

```
l ← list(  
  "um vetor de números" = x,  
  "um vetor de letras" = c("A", "B", "C")  
)
```

```
# data.frames
```

```
df ← data.frame(  
  numeros = x,  
  letras = c("A", "B", "C", "D", "E", "F")  
)
```

```
Error in data.frame(numeros = x,  
letras = c("A", "B", "C", "D", "E")) :  
arguments imply differing number of  
rows: 6, 5
```

# Operadores

- Operadores são símbolos ou palavras-chave especiais que são usados para realizar operações em variáveis, valores e objetos.

Aritiméticos	Comparação	Lógicos	Atribuição	Sequência
+: soma dois valores.	==: verifica se dois valores são iguais.	!: negação lógica, inverte o valor lógico de uma expressão.	<- ou =: atribui um valor a uma variável.	":": cria uma sequência de números de um valor inicial até um valor final.
-: subtrai um valor de outro.	!=: verifica se dois valores são diferentes.	& ou &&: operador AND lógico, verifica se duas expressões são verdadeiras.	<<-: atribuição global, usada dentro de uma função para atribuir um valor a uma variável fora do escopo da função.	seq(): cria uma sequência de números com um valor inicial, um valor final e um incremento especificado.
*: multiplica dois valores.	<: verifica se um valor é menor que outro.	ou   : operador OR lógico, verifica se pelo menos uma das expressões é verdadeira.		
/: divide um valor por outro.	>: verifica se um valor é maior que outro.			
^ ou **: eleva um valor à potência de outro.	<=: verifica se um valor é menor ou igual a outro.			
%%: retorna o resto da divisão entre dois valores.	>=: verifica se um valor é maior ou igual a outro.			
%/%: retorna a parte inteira do resultado da divisão entre dois valores.				



# Pacotes

- Para instalar um pacote, você pode usar a função `install.packages()` seguida pelo nome do pacote;
- Depois de instalar um pacote, você precisa carregá-lo no ambiente R antes de poder usar suas funções e recursos. Você pode fazer isso usando a função `library()` ou `require()`
- Depois de carregar um pacote, você pode usar suas funções e recursos no seu código R



```
install.packages("nome do pacote")
```



```
library(nomedopacote)
```

```
# ou
```

```
require(nomedopacote)
```

# Pacotes

- Os pacotes são coleções de funções, dados e documentação que são desenvolvidos por programadores e cientistas de dados da comunidade R.
- Em <https://cran.r-project.org/web/views/> é possível acessar uma lista de pacotes por área em que podem ser aplicados.

<a href="#">Agriculture</a>	Agricultural Science
<a href="#">Bayesian</a>	Bayesian Inference
<a href="#">CausalInference</a>	Causal Inference
<a href="#">ChemPhys</a>	Chemometrics and Computational Physics
<a href="#">ClinicalTrials</a>	Clinical Trial Design, Monitoring, and Analysis
<a href="#">Cluster</a>	Cluster Analysis & Finite Mixture Models
<a href="#">Databases</a>	Databases with R
<a href="#">DifferentialEquations</a>	Differential Equations
<a href="#">Distributions</a>	Probability Distributions
<a href="#">Econometrics</a>	Econometrics
<a href="#">Environmetrics</a>	Analysis of Ecological and Environmental Data
<a href="#">Epidemiology</a>	Epidemiology
<a href="#">ExperimentalDesign</a>	Design of Experiments (DoE) & Analysis of Experimental Data
<a href="#">ExtremeValue</a>	Extreme Value Analysis
<a href="#">Finance</a>	Empirical Finance
<a href="#">FunctionalData</a>	Functional Data Analysis
<a href="#">GraphicalModels</a>	Graphical Models
<a href="#">HighPerformanceComputing</a>	High-Performance and Parallel Computing with R
<a href="#">Hydrology</a>	Hydrological Data and Modeling
<a href="#">MachineLearning</a>	Machine Learning & Statistical Learning
<a href="#">MedicalImaging</a>	Medical Image Analysis
<a href="#">MetaAnalysis</a>	Meta-Analysis
<a href="#">MissingData</a>	Missing Data
<a href="#">MixedModels</a>	Mixed, Multilevel, and Hierarchical Models in R
<a href="#">ModelDeployment</a>	Model Deployment with R
<a href="#">NaturalLanguageProcessing</a>	Natural Language Processing
<a href="#">NumericalMathematics</a>	Numerical Mathematics
<a href="#">OfficialStatistics</a>	Official Statistics & Survey Statistics
<a href="#">Omics</a>	Genomics, Proteomics, Metabolomics, Transcriptomics, and Other Omics
<a href="#">Optimization</a>	Optimization and Mathematical Programming
<a href="#">Pharmacokinetics</a>	Analysis of Pharmacokinetic Data
<a href="#">Phylogenetics</a>	Phylogenetics

# Funções

- Funções são **métodos** presentes em pacotes (objetos) que existem para realizar tarefas;
- O pacote **dplyr** fornece funções para manipulação eficiente de dados, como filtragem, seleção, agregação e transformação.
- Operadores de Encadeamento: “|>” e “%>%”

```
require(dplyr)

df > filter(
  numeros > 3
) # linhas em que a coluna "numeros" é maior do que "3"

df > filter(
  letras != "A"
) # linhas em que a coluna "letras" é diferente de "A"

df > mutate(
  numeros = numeros ** 2
) # elevando a coluna "numeros" ao quadrado

df > select(letras) # selecionando apenas a coluna letras

df >
  mutate(
    letras = paste0(letras, '0')
    numeros = numeros ** 4
  ) >
  rename(
    "Letras com zero ao lado" = "letras"
    "Números elevados à quarta potência" = "numeros"
  ) # um exemplo de como NÃO nomear variáveis!
```

# Aplicações utilizando o macropacote tidyverse

# Relação entre a Mortalidade Infantil e a Taxa de Analfabetismo no Maranhão

1. Coletando Dados – Portal DataIMESC;
2. Descobrindo outliers (anomalias);
3. Limpeza de dados;
4. Teste de hipótese – T Student;
5. Análise de regressão linear com duas variáveis.

# Carregando os Dados

- Todos os dados, scripts e a apresentação estarão disponíveis no link do repositório no Github:  
[https://github.com/tiagolofi/curso\\_r\\_corecon](https://github.com/tiagolofi/curso_r_corecon)
- Para carregar dados no Rstudio existem uma gama extensa de métodos, nesta exposição será utilizada a função `read_excel` do pacote `readxl`.

# Construindo um Boxplot

- Um gráfico de boxplot é uma ferramenta gráfica utilizada para ilustrar um conjunto de dados. Por meio dele, é possível visualizar a distribuição de dados com base em cinco estatísticas:
  1. o mínimo;
  2. o primeiro quartil (Q1);
  3. a mediana;
  4. o terceiro quartil e (Q3);
  5. o máximo
- Para carregar dados no Rstudio existem uma gama extensa de métodos, nesta exposição será utilizada a função `read_excel` do pacote `readxl`.

# Teste T de Student? O que é isso?

- É um teste estatístico utilizado para realizar comparações de médias.
- O Teste pode apresentar três versões, em resumo:
  - Comparar a média do melhor tempo entre carros de F1 de diferentes construtoras (amostras independentes);
  - Comparar a média do melhor tempo entre carros da mesma construtora (amostra pareada);
  - Comparar a média do melhor tempo de todos os carros com um valor arbitrário: “eu acho que a média é 1m22s” (uma amostra).



# Regressão Linear

- Investigar a relação entre duas variáveis, tornando possível entender e prever o comportamento de uma determinada variável de interesse.
- A análise da regressão linear com duas variáveis visa estimar esses coeficientes e avaliar a significância estatística da relação entre as variáveis utilizando o Método de Mínimos Quadrados Ordinários (MQO) elaborado pelo célebre matemático Johann Carl Friedrich **Gauss**
- **Algebricamente, uma equação resultante do uso do MQO tem o seguinte formato:**

$$y = a + bx + e$$

onde, y é a variável dependente (a que nos interessa ser explicada);

a, é o intercepto ao eixo y;

b, é o parâmetro “beta”, que reflete a relação entre x e y;

x, é a variável independente (a que explica) e;

e, o termo de erro (mostra que aceitamos de forma humilde a limitação humana e podemos estar errados)

# Dicas de como iniciar uma análise – Tabela de Relacionamentos

		Variável "Y" - Mortalidade Infantil
Variáveis "X"	O que é	é a proporção de óbitos a cada mil nascidos vivos, por município, pode ser interpretada como um percentual ou uma probabilidade
Taxa de Escolaridade da Mãe	Percentual de mães com ensino médio	Relação esperada = negativa
Gasto com atenção primária em saúde em % do PIB	Despesa Empenhada na subfunção 301 / PIB do município	Relação esperada = negativa
Taxa de Pobreza	Percentual de pessoas abaixo da linha da pobreza (R\$ 168/mês)	Relação esperada = positiva
Taxa de sucesso nas visitas dos agentes de saúde	Visitas realizadas com sucesso / Total de Visitas	Relação esperada = negativa

# Bons Estudos!