

Módulo

# Conceitos básicos do software R

Curso: INTRODUÇÃO AO USO  
DO SOFTWARE “R” APLICADO  
À VIGILÂNCIA EM SAÚDE

## MÓDULO 01: CONCEITOS BÁSICOS DO SOFTWARE R

O módulo 1 visa estabelecer uma base sólida para os alunos no uso da linguagem R, preparando-os para explorar e aplicar técnicas mais avançadas de análise de dados nos módulos subsequentes do curso.

Nos últimos anos, em especial após o enfrentamento da Covid-19, os profissionais de saúde entenderam a necessidade de ampliar a sua capacidade de lidar com grandes quantidades de dados e como isso pode ser desafiador. A cada dia, os serviços de saúde produzem uma infinidade de dados e gerenciá-los torna - se cada vez mais complexo se não dispusermos de ferramentas adequadas para o trabalho.

Neste contexto, a linguagem de programação R surge como uma ferramenta poderosa para produção de informações para gestão em saúde com agilidade e qualidade. Com R você pode desde automatizar tarefas repetitivas do cotidiano até conduzir análises estatísticas robustas e visualização de dados em nível profissional.

### O que é R?

R é uma linguagem de programação e ambiente para computação estatística e geração de gráficos. R é de utilização livre! Há uma política forte na comunidade de usuários e desenvolvedores para que o material seja gratuito e de código aberto, o que o torna acessível a qualquer pessoa com interesse em conhecer e utilizar a linguagem.

R fornece uma ampla variedade de técnicas gráficas e estatísticas (modelagem linear e não linear, testes estatísticos clássicos, análise de séries temporais, classificação, agrupamento, aprendizado de máquina) que podem ampliar a capacidade analítica de profissionais de diversas áreas, inclusive da Vigilância em Saúde. É altamente versátil, extensível, e orientado para a comunidade.

### Por que você deveria aprender a linguagem R?

Além da flexibilidade e do custo zero, o que é uma grande vantagem em comparação aos softwares convencionais de análise de dados, o uso da linguagem R pode facilitar o trabalho diário da Vigilância em Saúde reduzindo o número de tarefas repetitivas e cansativas, tornando o trabalho e o tempo dos profissionais mais direcionado ao que realmente é importante, analisar os dados para direcionar ações, intervenções e apoiar a tomada de decisão.

A prática de conduzir análises e gerenciar dados usando uma linguagem de programação (em comparação com o MS Excel ou outra dessas ferramentas baseadas em cliques) melhora a reprodutibilidade, torna a detecção de erros mais fácil, alivia sua carga de trabalho e otimiza recursos.

### Por que fazer este curso?

Este curso faz parte do nível básico do Programa de Fortalecimento da Epidemiologia nos Serviços de Saúde – PROFEPI – e busca apresentar os conceitos iniciais do uso da

linguagem R para a manipulação de dados em epidemiologia, saúde pública e vigilância em saúde. A partir deste conhecimento, o profissional da vigilância em saúde pode desenvolver os conhecimentos necessários para participação nos cursos dos níveis intermediário e avançado do PROFEPI, que são voltados para o uso de métodos analíticos mais complexos (análise de séries temporais, modelos de regressão) com apoio computacional da linguagem R.

## DESTAQUE

*Com a linguagem R você será capaz de tornar as suas tarefas cotidianas de análise de dados muito mais ágeis economizando muitas horas de trabalho.*

*Já imaginou aquele trabalho de busca de inconsistências, aplicação de filtros em diversas colunas busca de duplicidades, geração de gráficos sendo realizado em poucos segundos? Tudo isso é possível com apoio de uma linguagem de programação como R!*

*Certamente, será necessário investir um pouco de tempo na instalação do programa e na familiarização com comandos básicos. Contudo, os benefícios desse esforço serão muitos e transformarão a sua rotina profissional.*

Ao final deste módulo, você será capaz de:

- ◆ Compreender os conceitos básicos da linguagem R.
- ◆ Conhecer as vantagens que uma linguagem de programação pode oferecer para um projeto de análise de dados.
- ◆ Instalar o R e o RStudio .
- ◆ Executar os primeiros comandos básicos com R e R Studio .
- ◆ Criar e salvar seus primeiros códigos.
- ◆ Instalar pacotes.
- ◆ Aplicar boas práticas para um fluxo de trabalho organizado com a linguagem R.

Este é um curso básico, mas depois que você dominar os conceitos iniciais da linguagem R, você será capaz de realizar várias tarefas analíticas e de manipulação de dados de uma forma muito mais completa e sem retrabalhos.

### **Algumas suposições que fazemos sobre o estudante deste curso:**

Para o melhor aproveitamento do conteúdo deste curso assumimos que você, como estudante, sabe como trabalhar com Windows ou Mac. Já deve ter alguma experiência com uso da internet para baixar e instalar softwares e em algum momento já lidou com uma tabela de dados, como uma planilha do MS Excel ou similares. O curso possui vários vídeos de apoio que apresentarão exemplos e tutoriais passo a passo, então se você possui um conhecimento básico do uso do computador você está em condições de aproveitar todo o conteúdo e aprender a linguagem R.

## **Origem do R**

A história da linguagem de programação R, desde suas raízes na linguagem S até sua posição proeminente no cenário da análise de dados, oferece uma visão fascinante da evolução tecnológica ao longo das décadas.

O início data nas décadas de 1970 e 1980, quando a linguagem S começou a ser desenvolvida nos laboratórios Bell por John Chambers e seus colegas. Concebida como uma linguagem voltada para análise estatística e gráficos, S foi influenciada por diversas outras linguagens, como APL, Lisp e Fortran.

No entanto, foi em meados da década de 1990 que uma nova fase começou a tomar forma, os professores Ross Ihaka e Robert Gentleman, da Universidade de Auckland, na Nova Zelândia, deram início a um projeto para desenvolver uma alternativa de código aberto para o software estatístico. Inspirados pela linguagem S, eles expandiram e aprimoraram muitos de seus aspectos para atender às crescentes demandas da análise de dados.

O resultado desse esforço foi o surgimento da linguagem R. Lançada em 1995, a primeira versão do R representou uma implementação completamente nova e reescrita da linguagem S. Sua chegada marcou o início de uma trajetória que o levaria a se tornar uma das ferramentas mais populares para análise de dados e estatísticas.

Apesar das raízes compartilhadas, R e S se diferenciam em vários aspectos significativos. Enquanto R se tornou um projeto de código aberto com uma comunidade de desenvolvedores ativa, S permaneceu como uma linguagem comercial. Além disso, a gratuidade de R para uso e distribuição, juntamente com sua flexibilidade e poder devido à vasta gama de pacotes desenvolvidos pela comunidade, contribuíram para sua crescente popularidade e adoção.

Ao longo das décadas seguintes, o R testemunhou um crescimento exponencial em sua aceitação e uso. Amplamente adotado por acadêmicos, pesquisadores e profissionais de diversas áreas como indústria, saúde, setor financeiro e outros interessados em análise estatística, mineração de dados e visualização, o R se estabeleceu como uma ferramenta indispensável em uma variedade de campos, incluindo ciência de dados, finanças, econometria e bioinformática.

Hoje, o R é reconhecido como uma das principais linguagens para análise de dados, competindo diretamente com outras linguagens de programação como Python e SAS. Sua flexibilidade e vasta coleção de pacotes o tornam uma escolha popular para explorar e analisar dados em uma variedade de contextos e setores.

A história da evolução do R a partir da linguagem S é um testemunho do poder transformador da inovação tecnológica e da colaboração da comunidade de desenvolvedores. Demonstrando como ideias e conceitos podem ser refinados e expandidos ao longo do tempo para atender às crescentes demandas e necessidades da comunidade de análise de dados, essa jornada é um exemplo inspirador do progresso na ciência da computação.

### **O que é o RStudio:**

O RStudio é um ambiente de desenvolvimento integrado (IDE) de código aberto desenvolvido para criar e executar código R. Ele está disponível em versões para os sistemas operacionais Windows, Mac e Linux. Embora não seja necessário usar uma IDE para trabalhar com R, o RStudio facilita muito a vida. Neste curso usaremos sempre a IDE Rstudio para tornar o seu aprendizado mais fácil.

### **Instalando R e RStudio:**

Para iniciar a nossa jornada de aprendizado faremos o *download* de dois softwares (programas) no computador, R e RStudio. O software de R criará o ambiente necessário para interpretar e executar os comandos da linguagem, já o RStudio será a interface que tornará o aprendizado e o trabalho mais fáceis ao longo do curso e também na sua rotina de trabalho com R.

#### **DESTAQUE**

Acesse o vídeo de instalação de R e RStudio, de acordo com o seu sistema operacional:

Vídeo de instalação R para Windows

Vídeo de instalação R para Mac

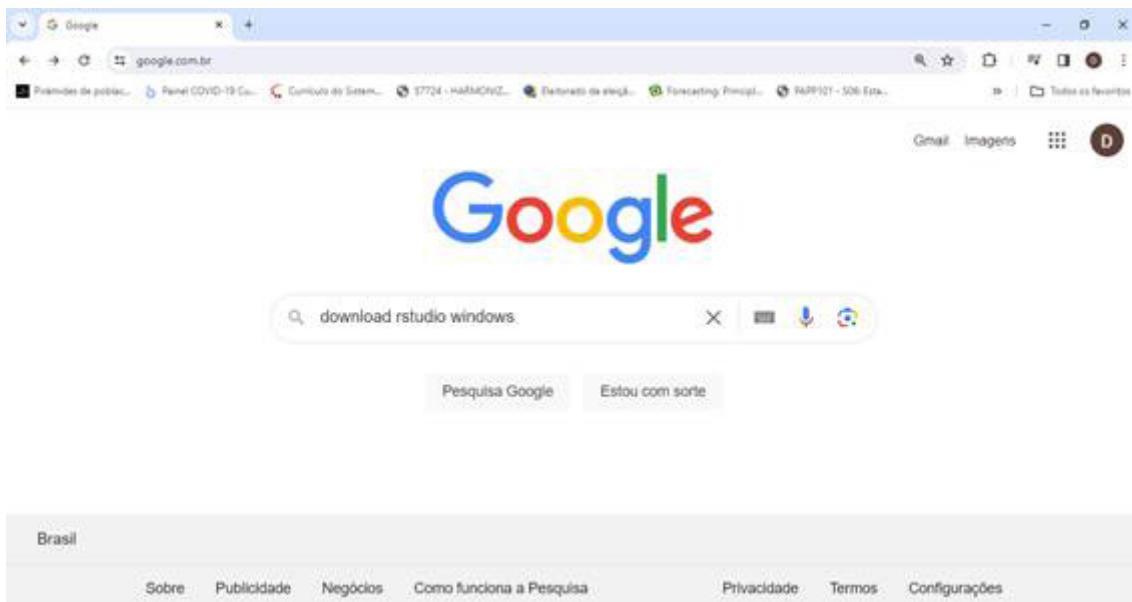
Vídeo de instalação RStudio para Windows

Vídeo de instalação RStudio para Mac

Baixando e instalando o software R e R Studio, tutorial Windows

É possível fazer o download do software R pelo repositório CRAN (The Comprehensive R Archive Network), uma rede de servidores que armazena diversas cópias das versões mais recentes do programa.

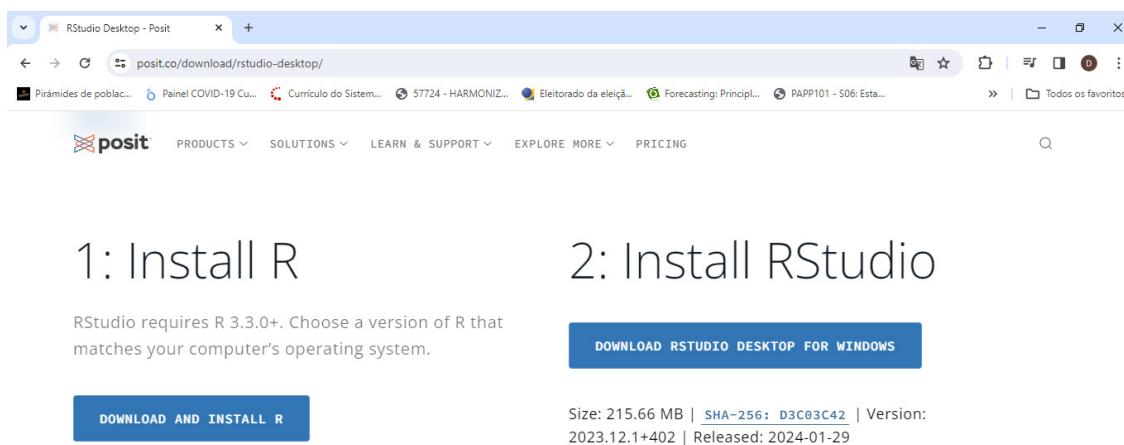
Vamos começar fazendo a busca no Google®, conforme a imagem a seguir:



**Figura 1.** Tela de busca do Google® para download do software RStudio

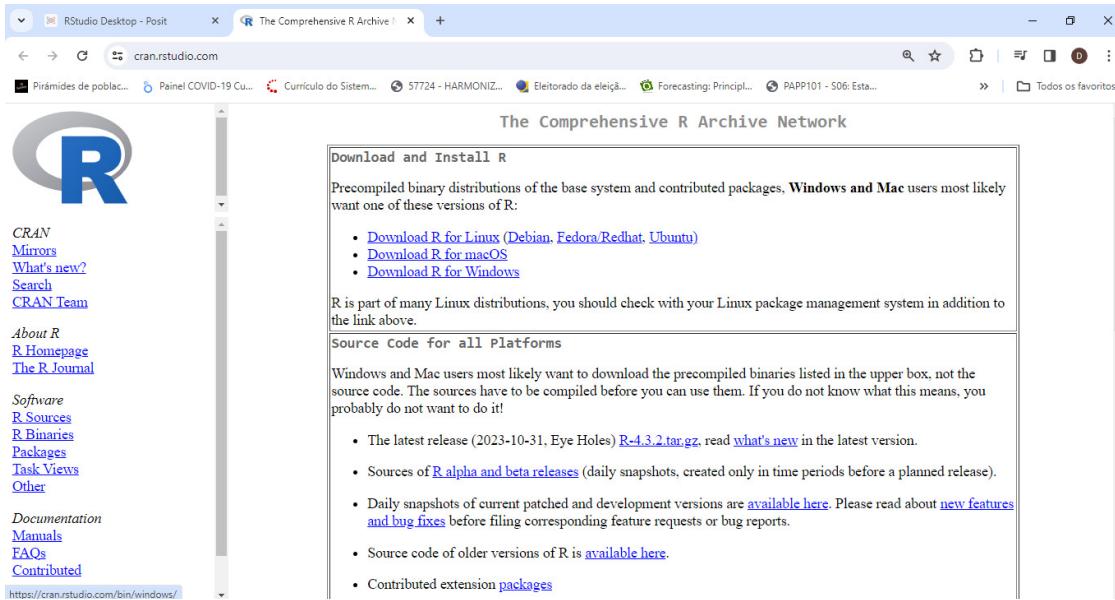
O primeiro resultado da busca deve direcionar para o link: <https://posit.co/download/rstudio-desktop/>. Este link é da página da empresa que criou o RStudio, a Posit, onde você encontrará os links para download de R e RStudio.

Após o carregamento da página role a barra de rolagem e você vai se deparar com a seguinte imagem:



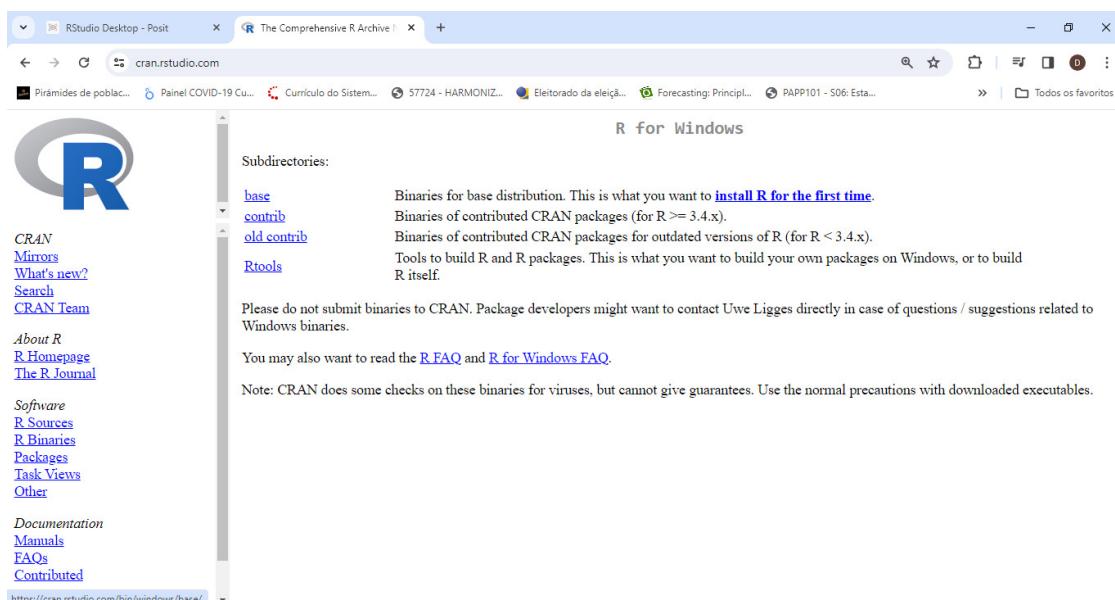
**Figura 2.** Página de entrada para a instalação do R e RStudio

Ao clicar em “DOWNLOAD AND INSTALL R” você será direcionado ao CRAN e logo no topo da página você encontrará o tópico “Download and Install R”, conforme a imagem a seguir, escolha o seu sistema operacional.



**Figura 3.** Página do CRAN com as opções de download do R, de acordo com o sistema operacional

Na próxima etapa clique sobre o link onde está escrito “install R for the fist time”



**Figura 4.** Tela de download do R para Windows

Na próxima página você terá a opção “Download R-4.3.3 for Windows (79 megabytes, 64 bit)”, clique sobre ela e download começará.

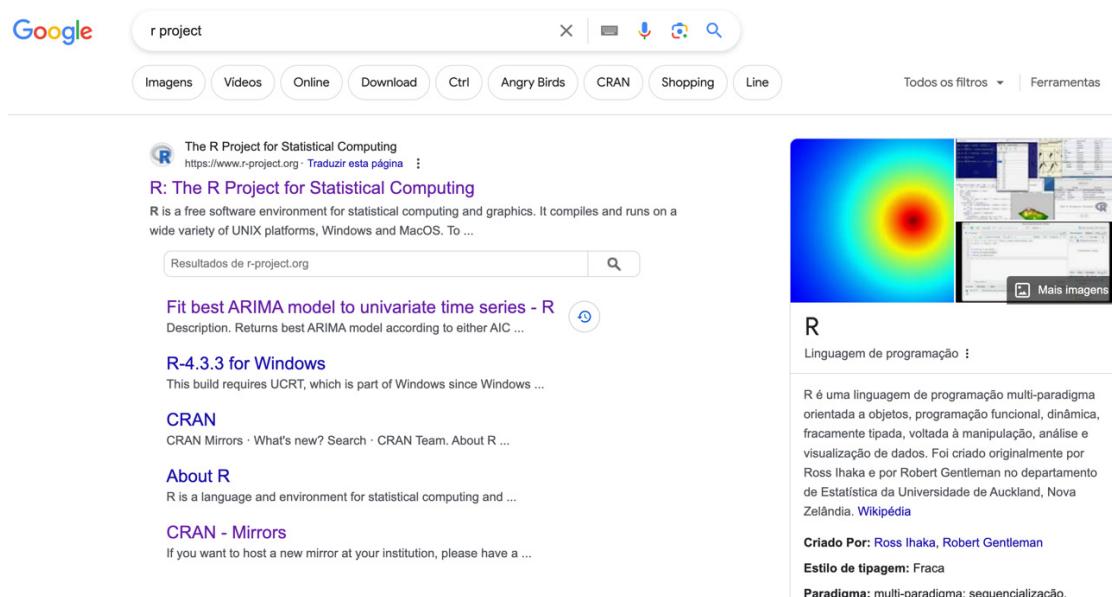
## IMPORTANTE

A versão de R disponível para Windows pressupõe que seu sistema operacional seja de 64 bits. Para confirmar se seu computador possui essa configuração, clique com o botão direito sobre o símbolo do Windows no canto esquerdo da sua tela e selecione a opção “Sistema”. No tópico *Tipo de Sistema* confirme que está escrito “Sistema operacional de 64 bits”.

## Baixando e instalando o software R e R Studio, tutorial MacOS.

Siga as etapas para instalar o R e o RStudio no sistema operacional macOS.

Instalação do R:Acesse o site oficial do R, na página de busca digitando “r project”no console é o primeiro site: <<https://www.r-project.org/>>



**Figura 5.** Tela de busca do Google® para download do software R (r project)

No acesso ao site, no lado esquerdo clique em Download >CRAN;

The screenshot shows the main page of the R Project for Statistical Computing. On the left sidebar, there are links for Home, Download (selected), CRAN, R Project, About R, Logo, Contributors, What's New?, Reporting Bugs, Conferences, Search, Get Involved: Mailing Lists, Get Involved: Contributing, Developer Pages, and R Blog. The 'Download' section is expanded, showing sub-links for Foundation, Board, Members, Donors, and Donate. Below the sidebar, the 'Getting Started' section explains R as a free software environment for statistical computing and graphics, compiled and run on various platforms. It includes a link to download R and choose a CRAN mirror. The 'News' section lists recent releases: R version 4.3.3 (Angel Food Cake) from 2024-02-29, Registration for useR! 2024 (early bird deadline March 31 2024), R version 4.3.2 (Eye Holes) from 2023-10-31, R version 4.2.3 (Shortstop Beagle) from 2023-03-15, and a note about supporting the R Foundation. The 'News via Mastodon' section shows a post from the R\_Contributors account about the R Dev Day @ PLUS event in Salzburg, Austria, on Friday, July 12, 2024.

**Figura 6.** Tela do site do R (r project)

O site abrirá os espelhos de instalação de diversos países, pois o R é de colaboração mundial:

The screenshot shows the 'CRAN Mirrors' page. It starts with a general statement that the Comprehensive R Archive Network is available at various URLs and encourages users to choose a location close to them. It links to the 'main page release'. Below this, it provides instructions for hosting a new mirror and links to the 'CRAN Mirror HOWTO'. The main content is a table listing CRAN mirrors by country, each with its URL and a brief description of the organization behind it. The countries listed include Argentina, Australia, Austria, Belgium, Brazil, Bulgaria, Canada, Chile, China, Colombia, Costa Rica, Czech Republic, Denmark, Finland, France, Germany, Greece, Hungary, India, Ireland, Italy, Japan, Korea, Luxembourg, Mexico, Netherlands, Norway, Poland, Portugal, Russia, Saudi Arabia, Spain, Sweden, Switzerland, Turkey, United Kingdom, and United States.

Country	URL	Description
0-Cloud	<a href="https://cloud.r-project.org/">https://cloud.r-project.org/</a>	Automatic redirection to servers worldwide, currently sponsored by Posit
Argentina	<a href="http://mirror.fcaglp.unlp.edu.ar/CRAN/">http://mirror.fcaglp.unlp.edu.ar/CRAN/</a>	Universidad Nacional de La Plata
Australia	<a href="https://cran.csiro.au/">https://cran.csiro.au/</a> <a href="https://mirror.aarnet.edu.au/pub/CRAN/">https://mirror.aarnet.edu.au/pub/CRAN/</a> <a href="https://cran.ms.unimelb.edu.au/">https://cran.ms.unimelb.edu.au/</a>	CSIRO AARNET School of Mathematics and Statistics, University of Melbourne
Austria	<a href="https://cran.wu.ac.at/">https://cran.wu.ac.at/</a>	Wirtschaftsuniversität Wien
Belgium	<a href="https://www.freestatistics.org/cran/">https://www.freestatistics.org/cran/</a> <a href="https://ftp.belnet.be/mirror/CRAN/">https://ftp.belnet.be/mirror/CRAN/</a>	Patrick Wessa Belnet, the Belgian research and education network
Brazil	<a href="https://cran-r.c3sl.ufpr.br/">https://cran-r.c3sl.ufpr.br/</a> <a href="https://cran.fiocruz.br/">https://cran.fiocruz.br/</a> <a href="https://vps.fmvz.usp.br/CRAN/">https://vps.fmvz.usp.br/CRAN/</a> <a href="https://bruegger.esalq.usp.br/CRAN/">https://bruegger.esalq.usp.br/CRAN/</a>	Universidade Federal do Paraná Oswaldo Cruz Foundation, Rio de Janeiro University of São Paulo, São Paulo University of São Paulo, Piracicaba
Bulgaria	<a href="https://ftp.uni-sofia.bg/CRAN/">https://ftp.uni-sofia.bg/CRAN/</a>	Sofia University
Canada	<a href="https://mirror.rcg.sfu.ca/mirror/CRAN/">https://mirror.rcg.sfu.ca/mirror/CRAN/</a> <a href="https://mugug.ca/mirror/cran/">https://mugug.ca/mirror/cran/</a> <a href="https://mirror.csclub.uwaterloo.ca/CRAN/">https://mirror.csclub.uwaterloo.ca/CRAN/</a>	Simon Fraser University, Burnaby Manitoba Unix User Group University of Waterloo
Chile	<a href="https://cran.dcc.uchile.cl/">https://cran.dcc.uchile.cl/</a>	Departamento de Ciencias de la Computación, Universidad de Chile

**Figura 7.** Tela do site do R para download no espelho CRAN (r project)

No Brasil, vocês podem escolher qualquer espelho de instalação, vejam que há 4 disponíveis: Universidade Federal do Paraná, Oswaldo Cruz Foundation, University of São Paulo (São Paulo) e University of São Paulo (Piracicaba), todos direcionam para a instalação de forma correta.

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 (2024-02-29, Angel Food Cake) [R-4.3.3.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.

**Supporting CRAN**

- CRAN operations most importantly hosting, checking, distributing, and archiving of R add-on packages for various platforms.

**Figura 8.** Tela do site do R para download do R pela primeira vez.

No sistema Mac, escolha “Download R for macOS”.

**ATENÇÃO PARA ESTA ETAPA!**

Verifique se o seu mac possui processador Intel ou chip M1, M2 ou M3 e escolha a versão correta.

This directory contains binaries for the base distribution and of R and packages to run on macOS. R and package binaries for R versions older than 4.0.0 are only from the [CRAN archive](#) so users of such versions should adjust the CRAN mirror setting (<https://cran-archive.r-project.org>) accordingly.

Note: Although we take precautions when assembling binaries, please use the normal precautions with downloaded executables.

R 4.3.3 "Angel Food Cake" released on 2024/02/29

Please check the integrity of the downloaded package by checking the signature:  
pkutil --check-signature R-4.3.3-arm64.pkg  
in the Terminal application. If Apple tools are not available you can check the SHA1 checksum of the downloaded image:  
openssl sha1 R-4.3.3-arm64.pkg

**CHIP M1-3**  
**Escolha essa opção**

For Apple silicon (M1-M3) Macs:  
**R-4.3.3-arm64.pkg**  
SHA1 hash: 376d05f6fb241eeefeb320fb2710067767781efc  
(ca. 92MB, notarized and signed)

For older Intel Macs:  
**R-4.3.3-x86\_64.pkg**  
SHA1 hash: cbff7e657ed67d3a35fd474772680aa3cf6b88d4b  
(ca. 44MB, notarized and signed)

**Processador Intel**  
**Escolha essa opção**

**Latest release:**

**R 4.3.3** binary for macOS 11 (**Big Sur**) and higher, signed and notarized packages.

Contains R 4.3.3 framework, R.app GUI 1.80, Tcl/Tk 8.6.12 X11 libraries and Texinfo 6.8. The latter two components are optional and can be omitted when choosing "custom install", they are only needed if you want to use the `tcltk` R package or build package documentation from sources.

macOS Ventura users: there is a known bug in Ventura preventing installations from some locations without a prompt. If the installation fails, move the downloaded file away from the `Downloads` folder (e.g., to your home or Desktop).

Note: the use of X11 (including `tcltk`) requires [XQuartz](#) (version 2.8.5 or later). Always re-install XQuartz when upgrading your macOS to a new major version.

This release uses Xcode 14.2/14.3 and GNU Fortran 12.2. If you wish to compile R packages which contain Fortran code, you may need to download the corresponding GNU Fortran compiler from <https://mac.R-project.org/tools>. Any external libraries and tools are expected to live in `/opt/R/arm64` (Apple silicon) or `/opt/R/x86_64` (Intel).

[NEWS](#) (for Mac GUI)

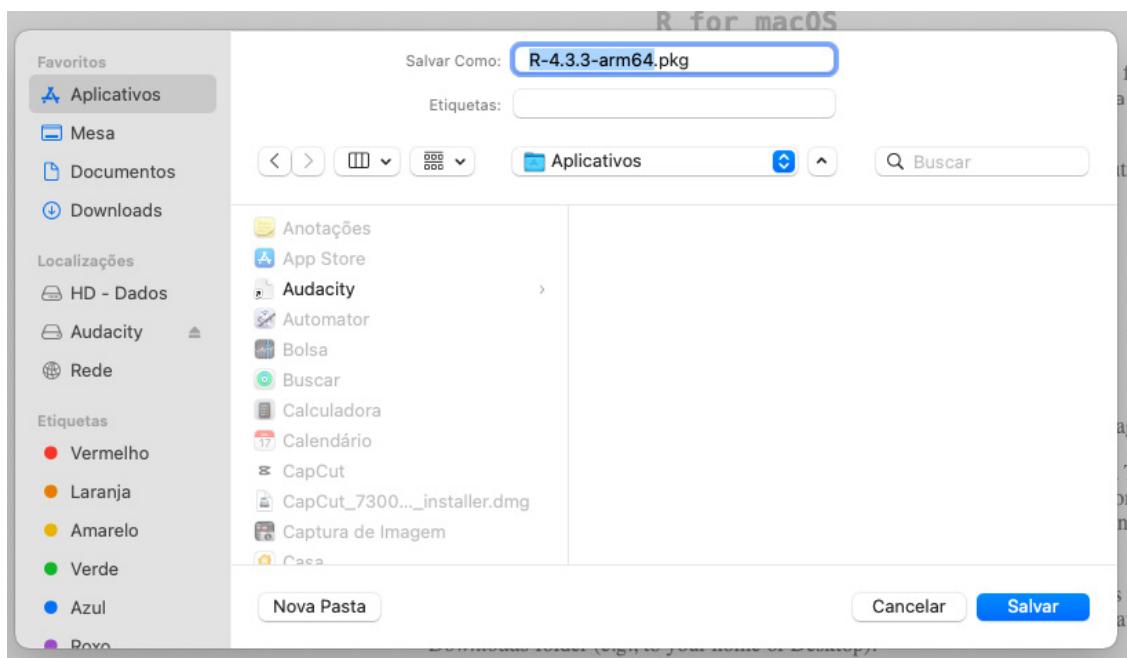
[Mac-GUI-1.80.tar.gz](#)

News features and changes in the R.app Mac GUI

Sources for the R.app GUI 1.80 for macOS. This file is only needed if you want to join the development of

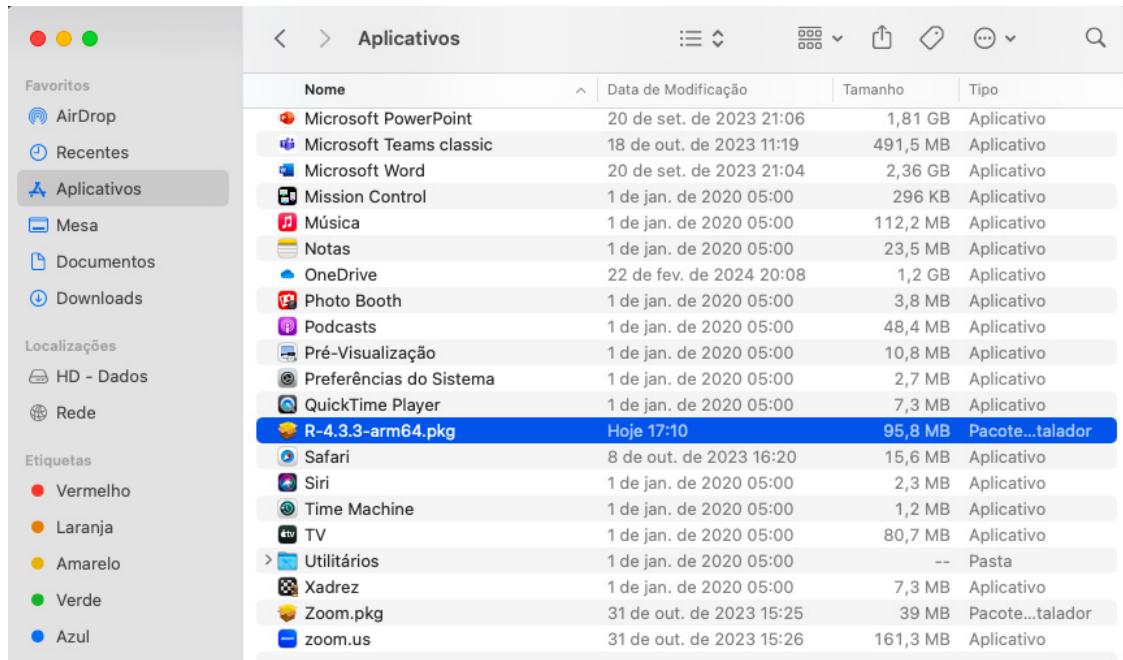
**Figura 9.** Tela do site do R para download do R no sistema operacional MacOS.

Clique no link correto compatível com seu processador para baixar a versão mais recente do R para macOS. Isso irá baixar um arquivo com extensão ".pkg". Salve na pasta aplicativos.



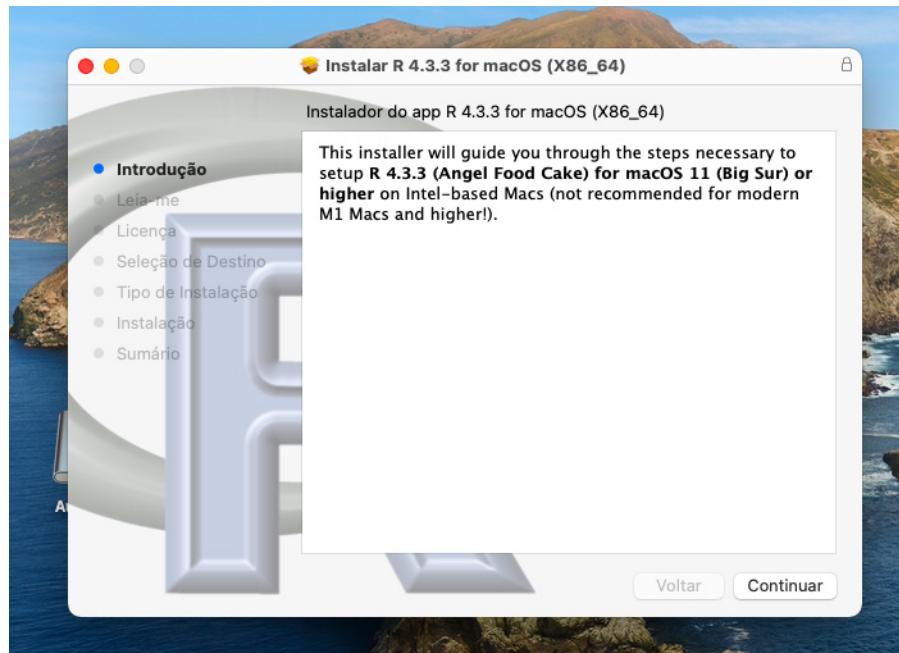
**Figura 10.** Pasta de aplicativos no finder do sistema MacOS.

Após o salvamento do arquivo execute com duplo clique, salvo na pasta aplicativos.



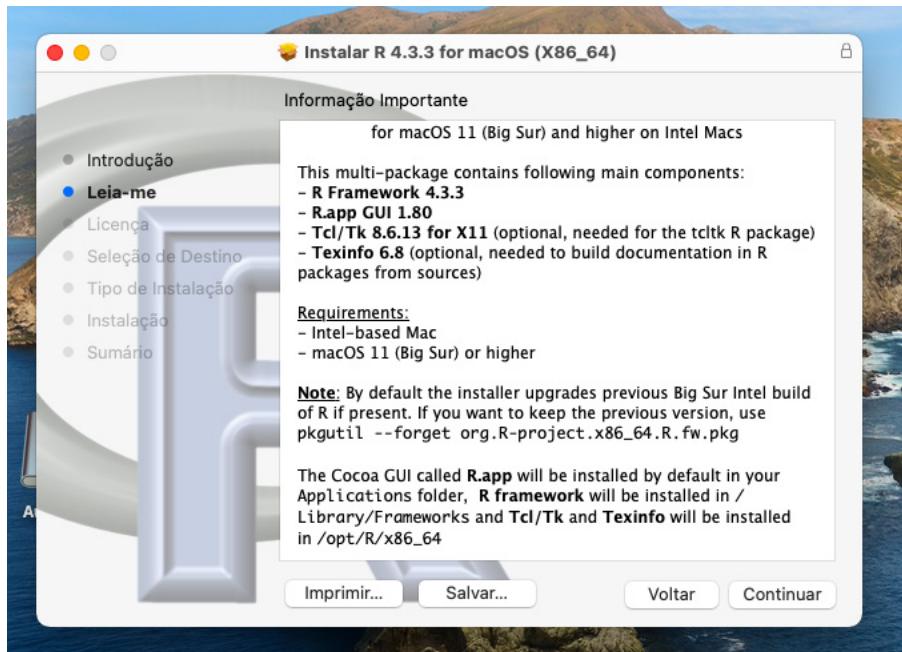
**Figura 11.** Pasta de aplicativos no finder do sistema MacOS, arquivo .pkg para instalação.

O processo de instalação deve iniciar e Siga as instruções do instalador, aceitando os termos do contrato de licença e escolhendo a opção padrão para a instalação.



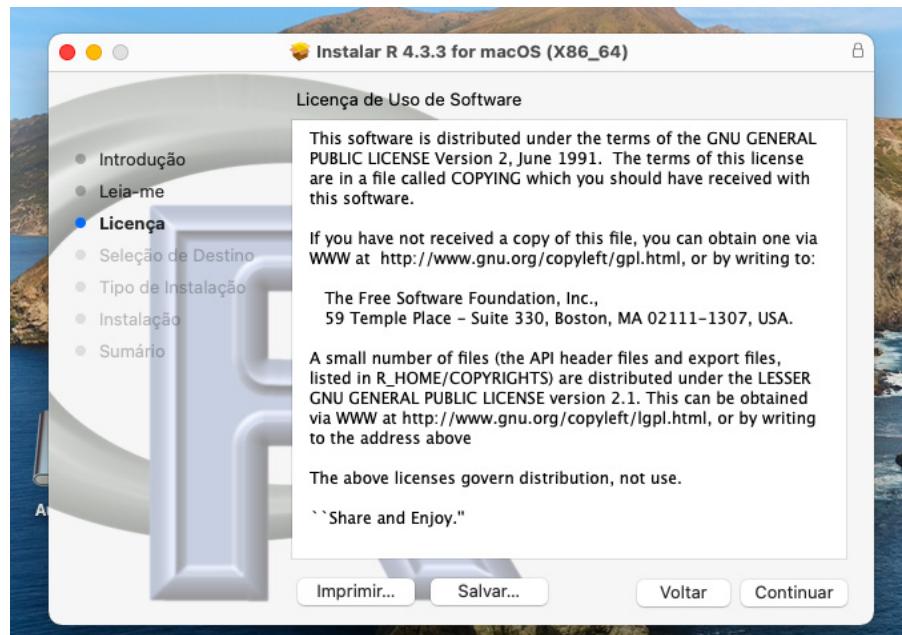
**Figura 12.** Instalação no sistema MacOS.

A instalação será feita usando o padrão, sem clicar em “Continuar”.



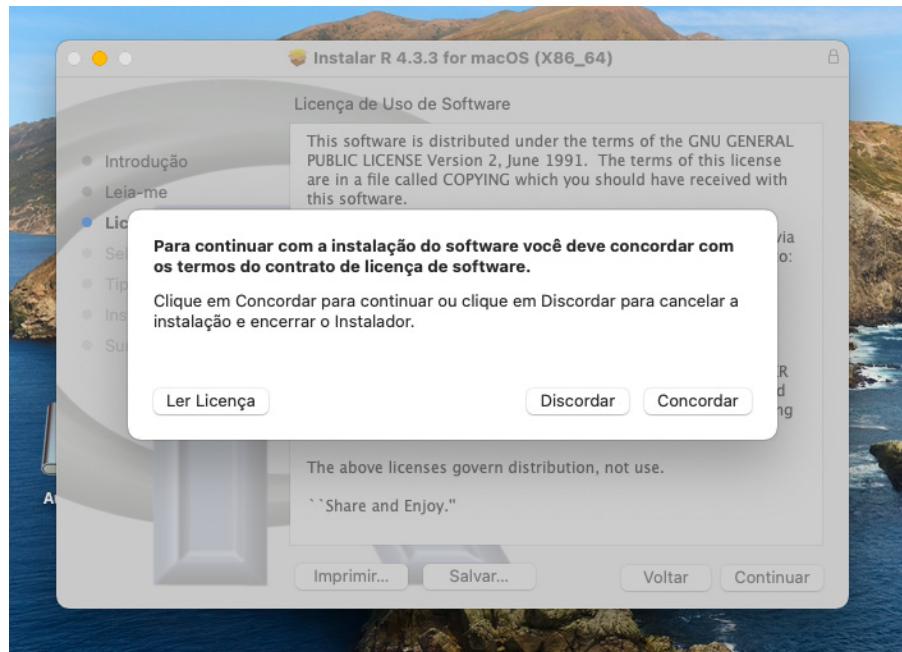
**Figura 13.** Instalação no sistema MacOS.

Na etapa da licença, vai exibir uma janela externa.



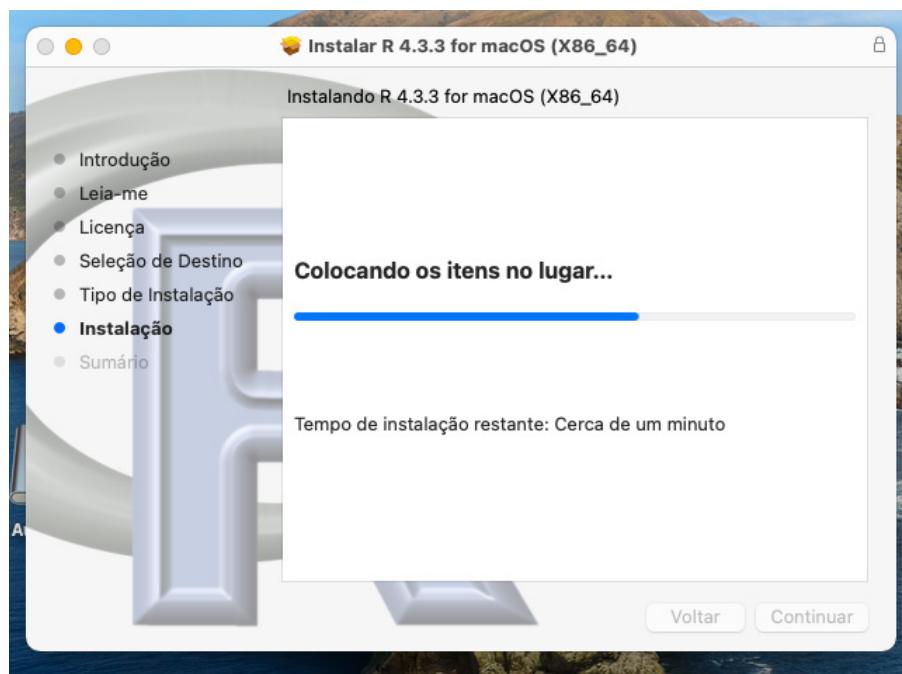
**Figura 14.** Instalação no sistema MacOS.

Na janela externa clique em “Concordar”



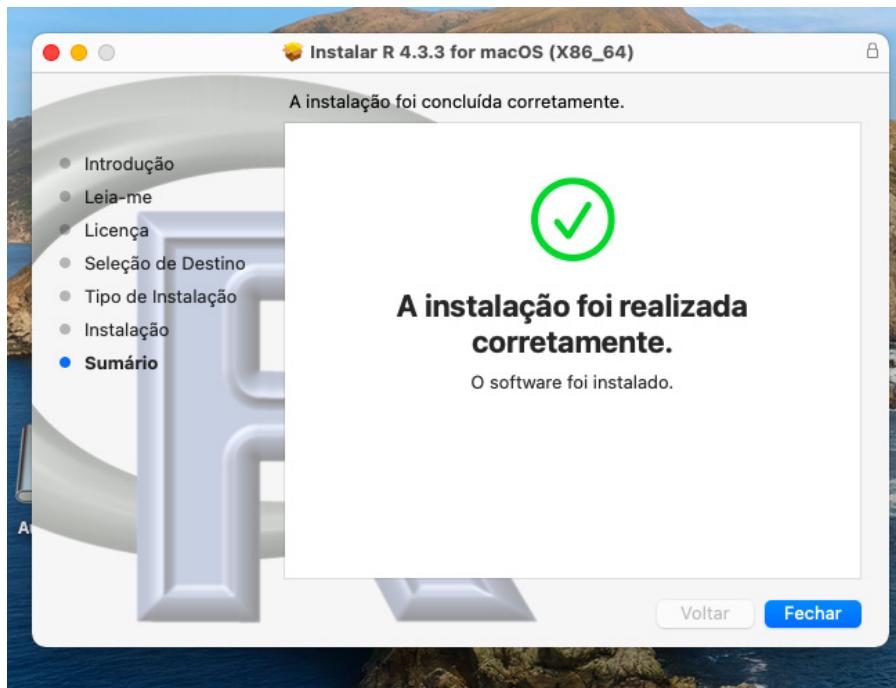
**Figura 15.** Instalação no sistema MacOS.

Depois disso irá proceder com instalação padrão do R



**Figura 16.** Instalação no sistema MacOS.

Por fim, será exibido uma mensagem de sucesso na instalação, caso acuse erro, recomce o processo ou faça a liberação em preferências do sistema.



**Figura 17.** Instalação no sistema MacOS, etapa concluída com sucesso.

Após a instalação R, temos que fazer a instalação do R Studio for MacOS, no site: <https://posit.co/download/rstudio-desktop/>

## 2: Install RStudio

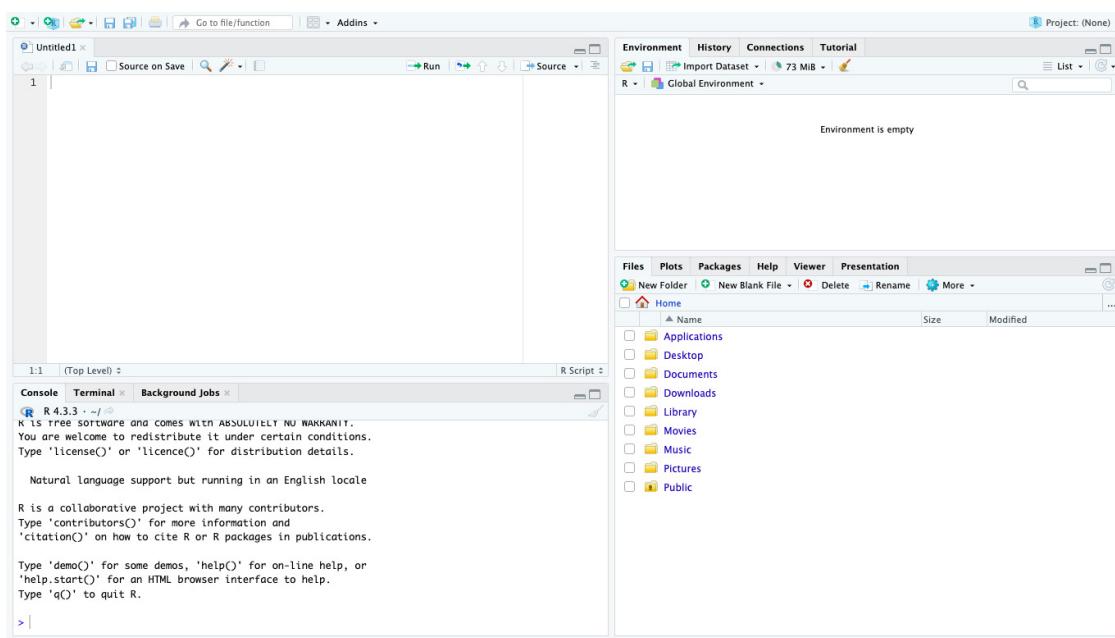
[DOWNLOAD RSTUDIO DESKTOP FOR MACOS 12+](#)

This version of RStudio is only supported on macOS 12 and higher. For earlier macOS environments, please [download a previous version](#).

Size: 382.66 MB | [SHA-256: C8D9185D](#) | Version: 2023.12.1+402 | Released: 2024-01-29

**Figura 18.** Instalação no RStudio no sistema operacional MacOS.

Após o download, clique duas vezes no arquivo baixado para iniciar o instalador do RStudio e siga as instruções do instalador, aceitando os termos do contrato de licença e escolhendo a opção padrão para a instalação.



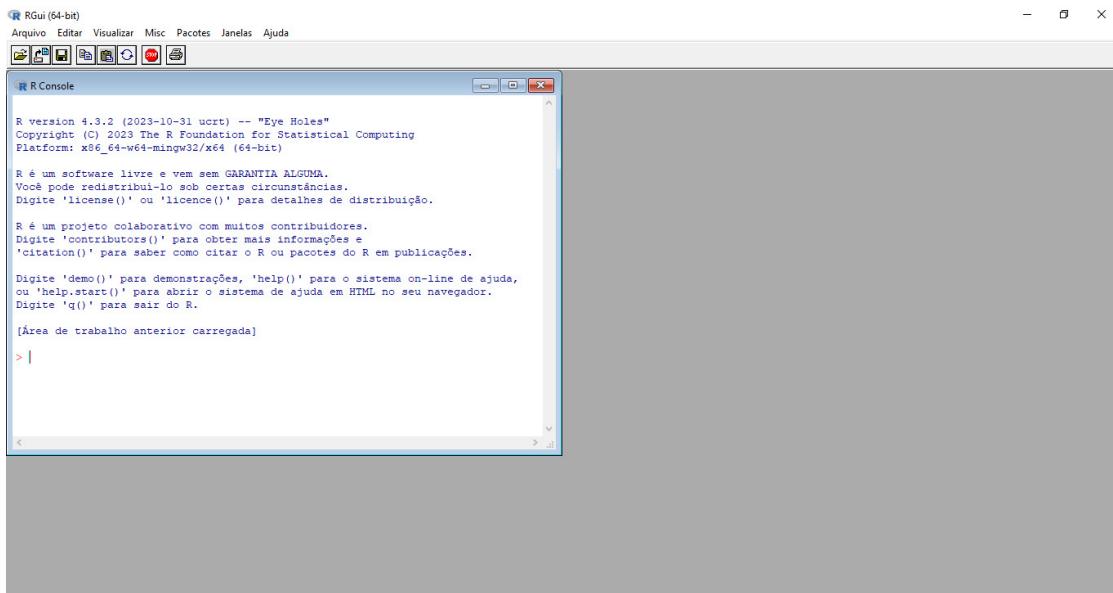
**Figura 19.** Interface do RStudio no sistema operacional MacOS.

Após a instalação bem-sucedida do R e RSTUDIO, você pode abrir o RStudio a partir do *Launchpad* ou da pasta de Aplicativos, e começar a usar o R para suas análises e projetos.

Pronto! Agora vamos conhecer um pouco da interface do R e RStudio!

### Conhecendo a interface do R básico:

Durante o processo de instalação o R vai criar um atalho na sua área de trabalho, se você executou a instalação padrão. Ao clicar no ícone do R ele vai apresentar a interface do R básico, com a aparência mostrada na figura a seguir.



**Figura 20.** Tela de entrada do R básico no Windows

Como já comentamos no início deste módulo, usaremos sempre a interface da IDE RStudio, mas apresentaremos a interface do R básico para que você conheça e também porque faremos uma operação inicial nesta ferramenta, a definição do repositório do CRAN que utilizaremos para acessar os recursos do R no RStudio.

A interface do R básico é muito menos amigável que a do RStudio, então não se assuste ao visualizá-la pela primeira vez. O nosso trabalho todo será realizado na IDE. Aqui nessa etapa faremos a escolha de qual repositório será a nossa fonte para acessar os pacotes que complementam as funções do R básico.

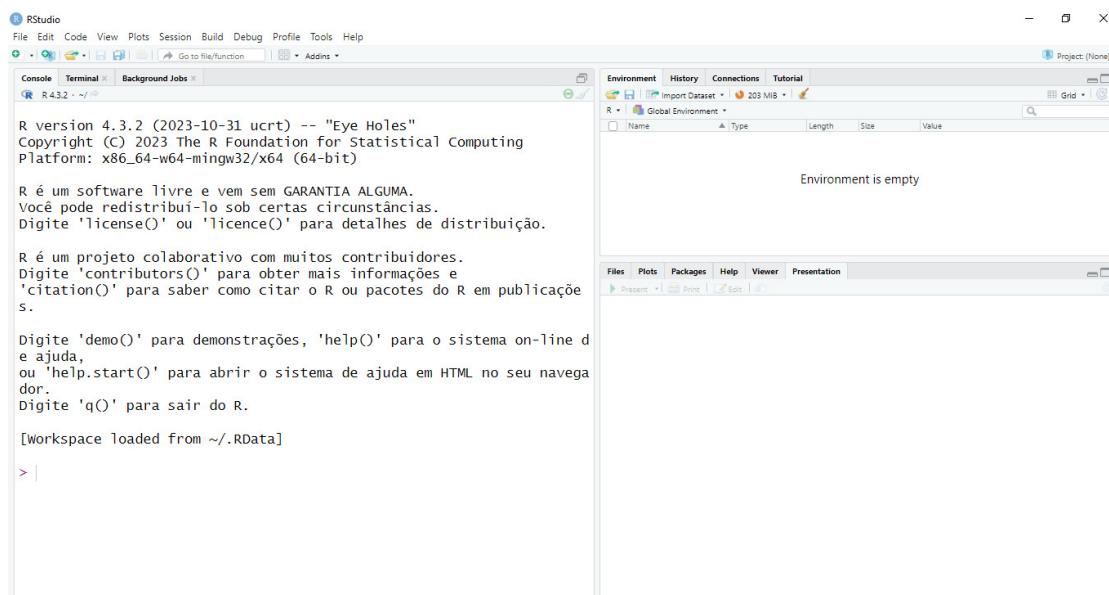
O primeiro passo é clicar na parte superior na barra de menus sobre o menu “pacotes” e escolher a opção “Escolher espelho CRAN...”. Ao clicar nessa opção o R vai mostrar uma lista com vários países onde existem espelho do CRAN, vamos escolher o Brasil e a região que fica mais próxima da sua casa ou instituição de trabalho. Você pode escolher a mais próxima de você entre as opções Paraná (PR), Rio de Janeiro (RJ) ou São Paulo (SP1 e SP2). Pronto, R já sabe onde vai buscar seus recursos quando você chegar na etapa de instalação de pacotes.

No vídeo [apresentação R básico](#), você pode acompanhar um tour rápido pela interface do R, mas a melhor forma de conhecer uma ferramenta nova é explorar por conta própria. Não tenha medo, clique nos menus e veja o que a ferramenta apresenta. Explore!

Encerramos essa breve jornada sobre a interface do R básico, e vamos partir para onde vamos trabalhar ao longo do curso todo, o RStudio. Ao clicar em sair do R ele fará uma pergunta: “Salvar imagem da área de trabalho?”. Pode clicar em Não. Esta função serve para R registrar as operações que você realizou durante o seu trabalho, mas nesta etapa isso não será necessário e voltaremos a falar disso mais adiante no curso.

## Conhecendo a interface do RStudio:

Se você está usando RStudio pela primeira vez, e é o que esperamos neste momento, ele deve apresentar uma interface como a da imagem a seguir:



**Figura 21.** Tela de entrada da IDE RStudio

Assim como na apresentação da interface do R básico, aqui a recomendação é a mesma, explore! No vídeo de [apresentação da IDE RStudio](#), você pode acompanhar uma demonstração do uso dos menus e da sua organização em painéis..

Aqui no e-book focaremos nos quatro principais painéis do RStudio para que você possa iniciar, mesmo que esteja sem acesso ao vídeo neste momento. No primeiro acesso você verá do lado esquerdo superior o painel chamado “Console”, é nesta parte que seu código é executado, aqui está rodando o R básico, afinal é ele quem faz todo o trabalho “sujo” para nós.

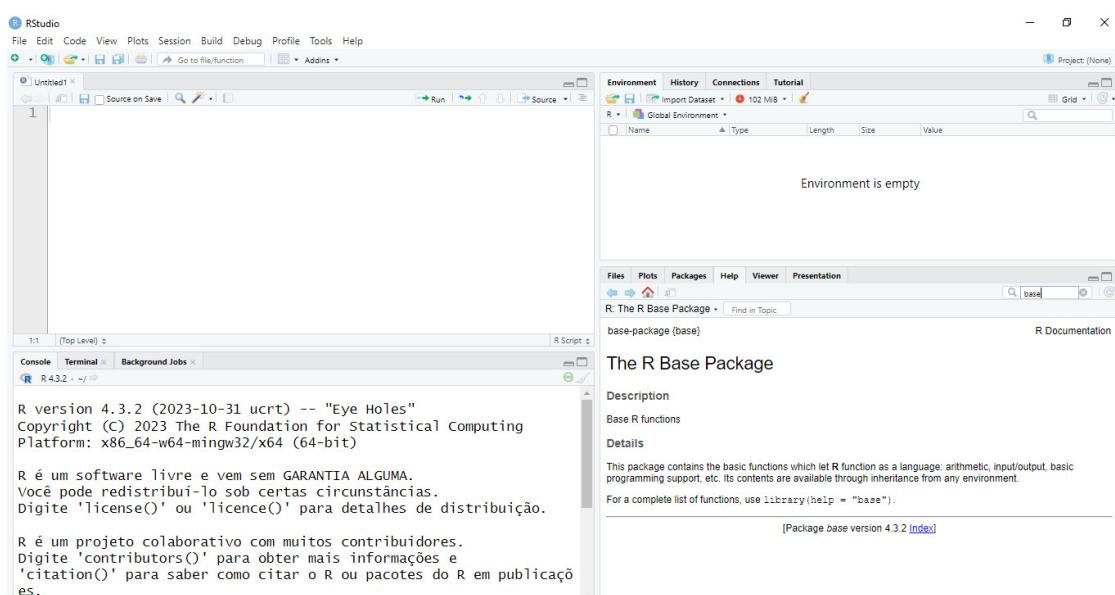
Usando o primeiro menu na parte superior, o menu “File” podemos criar um script, que é o arquivo de código que vamos guardar para termos as nossas tarefas sempre prontas para serem executadas várias vezes. Para isso clique no menu “File” e escolha a opção “New File” e em seguida na primeira opção “R Script”. Ao fazer isso, você verá que o RStudio cria um novo painel na parte superior esquerda como o nome “Untitled1”. É nessa parte que vamos iniciar os nossos códigos nos próximos passos do curso.

Logo do lado direito deste painel você verá um outro bloco chamado “Environment”. Nesta parte estarão os objetos que vamos criar durante os nossos códigos.

## LEMBRETE

Guarde essa informação sobre **objetos**. **R é uma linguagem de programação orientada a objetos, então tudo nela é um objeto.**

O quarto bloco que veremos agora no início começa com a aba de “Help”. Neste bloco está ajuda do R, estão os caminhos dos arquivos, futuros gráficos que vamos criar, essa é a parte mais visual e de apoio do RStudio. A imagem a seguir, ilustra os 4 blocos principais:



**Figura 22.** Painéis principais do RStudio

Assim chegamos a primeira parte do módulo 1, você já instalou o R básico e RStudio, conheceu a sua interface e está pronto para a próxima etapa: instalação dos pacotes.

## Instalação dos pacotes

Os pacotes, ou packages em inglês, representam conjuntos de bibliotecas que incluem funções e dados destinados a ampliar a capacidade do R básico na manipulação e análise de dados. Na etapa anterior, ao realizar a instalação inicial do R, diversos pacotes foram automaticamente instalados, mas para melhorar o nosso fluxo de trabalho precisaremos instalar mais alguns. Não se preocupe, esta tarefa é feita de modo muito simples com auxílio do RStudio.

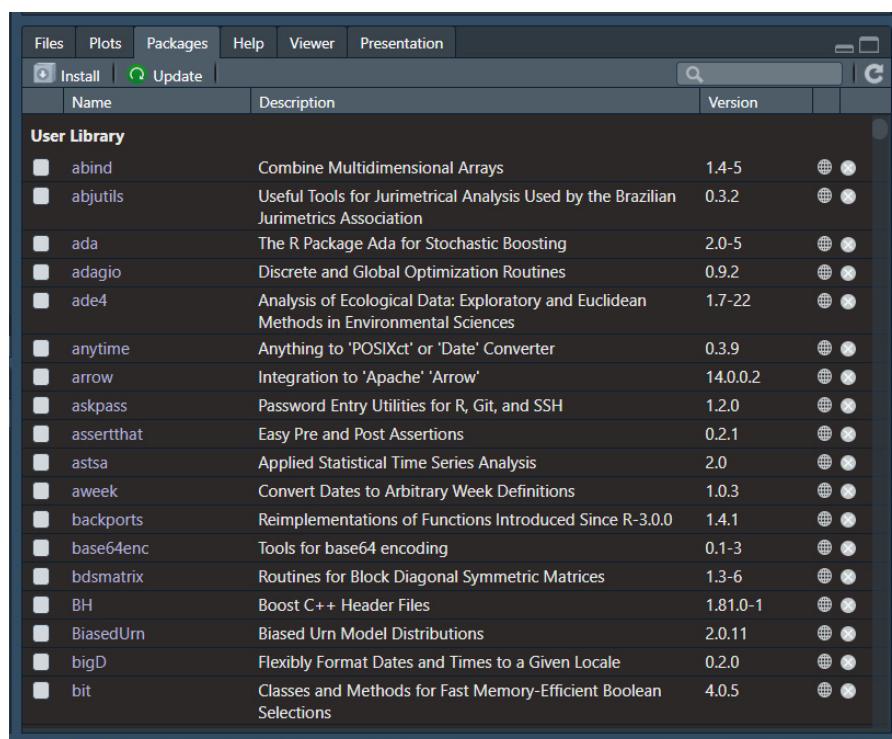
Ao longo deste curso, faremos a instalação de vários pacotes com finalidades específicas voltadas para a análise de dados das diversas áreas da vigilância em saúde. A utilização desses pacotes é um dos fatores-chave para a popularidade e versatilidade do R, uma vez que colaboradores de todo o mundo estão continuamente trabalhando na

criação de novos pacotes que abordem as necessidades específicas para muitos tipos de tarefas em análise de dados.

Nos últimos anos, foram desenvolvidos diversos pacotes que desempenham um papel significativo no manejo de dados relacionados à saúde pública.

Existem diferentes maneiras de instalar pacotes em R. Eles podem ser instalados usando o menu de ferramentas (Tools) do RStudio, pela aba *packages* ou diretamente via código. No vídeo de [instalação de pacotes](#) você pode acompanhar os principais recursos que utilizaremos neste curso.

A seguir, temos um exemplo da aba *packages*, localizada na parte inferior à direita da tela do RStudio mostrando os pacotes que já foram instalados na etapa da instalação inicial e onde estarão também os novos pacotes que instalaremos ao longo do curso.



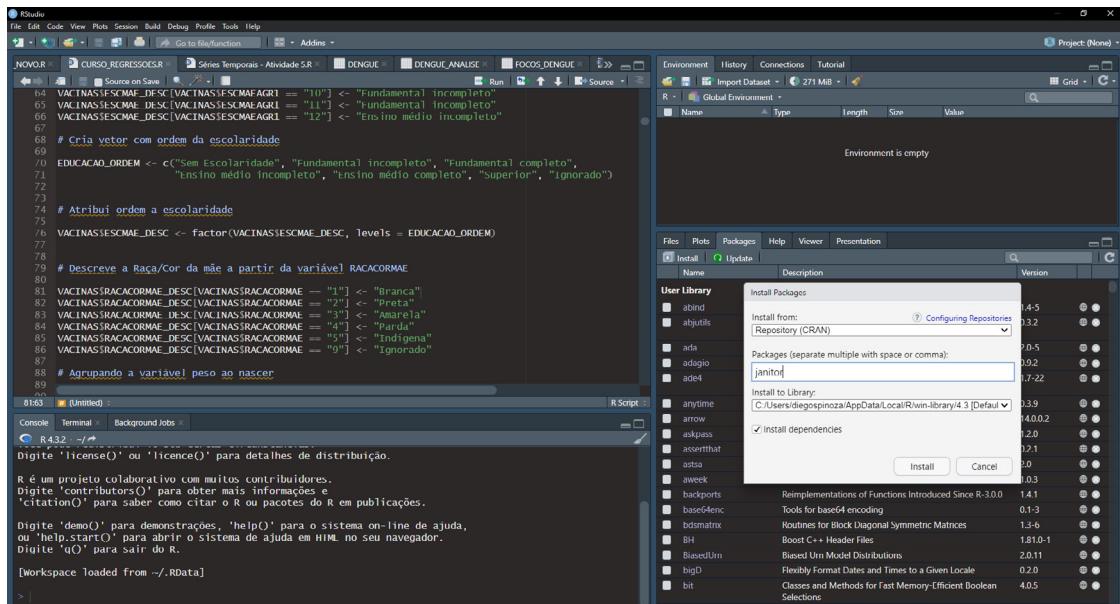
**Figura 23.** Painel de pacotes (packages) do RStudio

## DESTAQUE

Porque a imagem está com fundo preto e no meu RStudio a tela é branca?

No vídeo sobre a [apresentação RStudio](#) tem uma mostra de como você pode personalizar a aparência da sua interface de trabalho, confira! Para algumas pessoas, ou em ambientes muito claros, usar uma tela escura pode ser uma boa opção. Teste aquela que você se adapta melhor!

No e-book faremos um exemplo de instalação usando a aba mostrada na Figura 8 para instalar um pacote muito útil para manipulação de dados, o pacote janitor. Veja na imagem a seguir como isso pode ser feito de maneira simples com RStudio.



**Figura 24.** Instalando um pacote pela aba packages

Ao clicar no botão “Install” o RStudio abrirá uma pequena janela onde você vai escrever o nome do pacote que será instalado. Se você quiser instalar mais de um pacote de uma única vez, basta separar os nomes por uma vírgula ou um espaço.

## AVISO

Nesta parte precisamos fazer um destaque especial. Dentro dos serviços de saúde (Secretarias Municipais, Estaduais, Unidades, Hospitais etc.) você pode se deparar com regras de acesso à internet (como uso de proxy, por exemplo) que podem interferir na rotina de baixar ou instalar programas, não só RStudio, e os pacotes. Consulte a equipe de Tecnologia da sua instituição para entender se essas regras se aplicam ao seu ambiente de trabalho. Não há restrições ao uso, pois ambos são gratuitos, mas as regras de conexão podem dificultar o processo de download de pacotes. No vídeo ([link do vídeo de instalação de pacotes](#)) você tem uma explicação sobre isso e como contornar possíveis erros.

Muito bem! Estamos chegando ao fim de mais uma etapa deste módulo 1, a instalação dos pacotes. Você pode estar se perguntando: Como eu vou saber quais pacotes eu preciso instalar para cada análise que vou fazer?

Essa é uma pergunta recorrente para usuários iniciantes de R e outras linguagens de programação. Neste curso apresentaremos os principais pacotes que devem resolver a maior parte das tarefas de manipulação de dados para que você tenha capacidade de iniciar o uso da linguagem. Com o passar do tempo, e com o uso constante de R, você vai aprender a buscar novas ferramentas que poderão ser úteis no seu cotidiano. Não tenha pressa agora, este é um aprendizado contínuo e mesmo usuários experientes de R não conhecem todos os pacotes disponíveis.

### **Fluxo de trabalho com R:**

Agora que você já tem R e RStudio instalados vamos começar a entender um pouco sobre os scripts que é onde construiremos as nossas análises. Imagine um “roteiro” que guarda seus comandos e os executa automaticamente. Essa é a magia dos scripts! Com essa ferramenta você pode automatizar uma análise de dados que precisa ser feita toda semana, sem perder tempo repetindo os mesmos passos. Você também pode gerar gráficos com dados atualizados automaticamente, sem precisar refazer o processo a cada dia e ainda pode automatizar qualquer tarefa repetitiva no R, liberando tempo para você se concentrar em atividades mais complexas.

O script armazena seus comandos como um “livro de receitas”, permitindo:

- ◆ Reutilização: use o script sempre que precisar realizar a mesma tarefa, sem precisar digitar os comandos novamente.
- ◆ Compartilhamento: compartilhe o script com colegas e colaboradores, facilitando a reprodução da análise ou tarefa.
- ◆ Organização: mantenha seus comandos organizados e documentados em um único arquivo.
- ◆ Com scripts R, você libera seu tempo e transforma seu fluxo de trabalho!
- ◆ Mas para isso algumas “regras” são importantes para garantir que as tarefas possam ser executadas da melhor maneira possível no processo de manipulação e análise de dados.

### **Criando um script:**

No início deste módulo já vimos que para criar um novo script basta acessar o menu “File” e escolher a opção “New File” e em seguida clicar na primeira opção “R Script”.

Para iniciar uma análise o primeiro passo é começar carregando os pacotes que serão utilizados na análise. Cada vez que você acessar o RStudio para iniciar uma análise você precisará carregar os pacotes para que eles estejam disponíveis para o trabalho. Por isso, é recomendável que a primeira parte do seu código seja o carregamento dos pacotes, mesmo antes de carregar as bases de dados. Isso é importante, pois você precisa que os pacotes estejam carregados para poder importar os seus dados.

Outra recomendação é que cada etapa do seu código seja comentada, afinal você pode não lembrar exatamente o que cada etapa está fazendo e isso ajuda você e outras pessoas a compreenderem o que está acontecendo no seu script.

Os comentários podem ser úteis para descrever brevemente o que o código a está fazendo. Com trechos curtos de código talvez não seja necessário deixar um comentário para cada linha de código. Mas à medida que o código que você está escrevendo fica mais complexo, os comentários podem economizar muito tempo para você mesmo (e seus colaboradores) para descobrir o que foi feito.

Uma terceira recomendação importante é que os nomes dos objetos que você criar devem ser descritivos, para que você ao ler um nome possa ter uma ideia clara do que aquele objeto armazena. Por exemplo, se você está criando um objeto que contém uma tabela de pessoas com diabetes do sexo masculino você pode defini-lo como “diabetes\_masculino”. Isso, associado aos comentários, torna o seu código mais fácil de ler e interpretar. Qualquer objeto em R deve começar com uma letra e só recebe letras, números, “\_”, ou um “.”.

## Estrutura da linguagem e termos chave

### O básico do código em R

Até agora trabalhamos apenas com a instalação dos softwares e pacotes e algumas dicas gerais de como trabalhar com R, para que você ganhasse alguma familiaridade com o ambiente que vamos utilizar. A partir de agora passaremos a trabalhar com o código para que o mais rápido possível você seja capaz de gerar seus primeiros scripts e entenda a estrutura da linguagem.

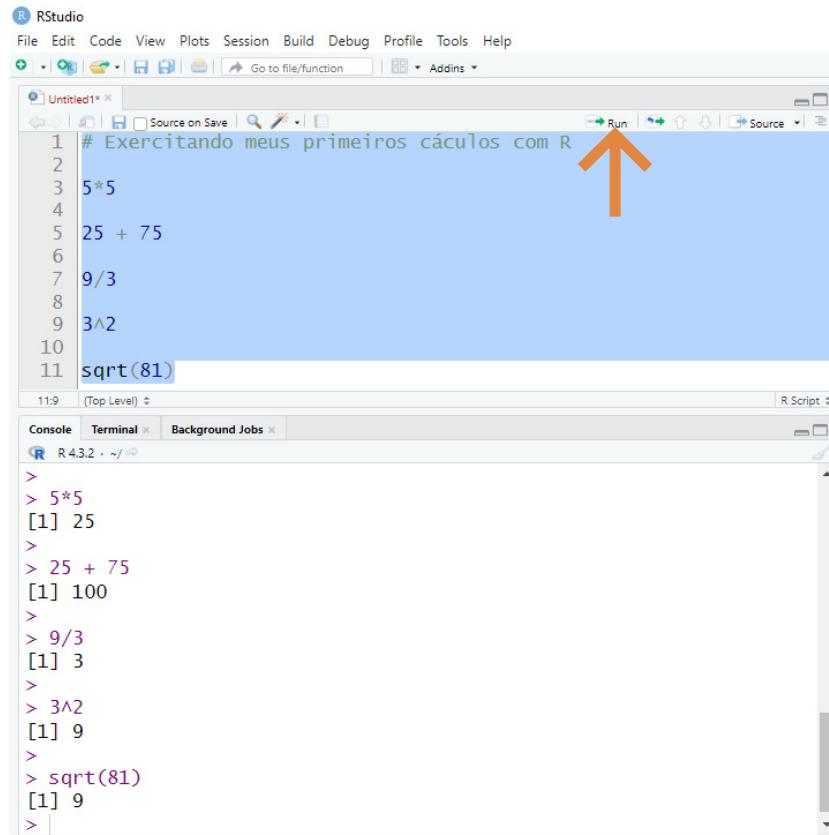
R pode funcionar como uma calculadora, realizando cálculos matemáticos direto no console ou nos seus scripts. Para começar vamos fazer algumas operações básicas para exercitar o uso da ferramenta.

Para começar, crie um novo script (você já aprendeu isso nas etapas anteriores). Agora no painel de scripts vamos fazer algumas operações matemáticas para exercitar, veja no exemplo a seguir:

Código:

```
# Exercitando meus primeiros cálculos com R  
5*5  
25 + 75  
9/3  
3^2  
sqrt(81)
```

No código de exemplo estamos trabalhando com as operações de multiplicação, adição, divisão, potenciação e uma raiz quadrada (com a função `sqrt`). Para que o R realize essas operações você precisa dar um comando de execução. Você pode fazer isso colocando o cursor em cada uma das linhas e clicando no botão “Run” (veja a imagem a seguir), ou ainda selecionando todo o código que você escreveu e usando o mesmo comando para que o R execute todo o código, ou então usando o atalho de teclado Ctrl+Enter.



**Figura 25.** Executando os primeiros cálculos com R usando o comando Run

Vamos continuar com alguns comandos básicos que você pode experimentar para se familiarizar com o ambiente, conforme já mencionamos, o uso da ferramenta é treino e vai se consolidando com o uso:

1. Atribuição de variáveis: Use o operador ‘`<-`’ para atribuir valores a variáveis. Exemplos:

```

x <- 10
y <- "Olá, mundo!"

```

2. Exibição de variáveis: Use a função `print()` ou simplesmente digite o nome da variável para exibir seu valor. Exemplo:

```
print(x)
```

```
y
```

- 3.** Operações matemáticas simples: Experimente realizar algumas operações matemáticas básicas, usando o R como calculadora. Por exemplo:

```
soma <- 5 + 3  
subtracao <- 10 - 2  
multiplicacao <- 4 * 6  
divisao <- 15 / 3  
exponencial<-5**3
```

```
print(soma)  
print(subtracao)  
print(multiplicacao)  
print(divisao)  
print(exponencial)
```

- 4.** Funções demonstrativas: o R possui muitas possibilidades na construção de gráficos, mapas e outros visuais.

```
#Demonstrações da interface gráficas [vá clicando para ir mostrando]
```

```
demo(graphics)
```

```
demo(image)
```

```
demo(persp)
```

- 5.** Tipos de objetos aceitos no ambiente R: número reais, texto, lógico/relacional e número complexos

```
## Tipo de objeto  
x<-c(1,5,3,8)  
y<-c("A","B","A","B","C")  
e<- 2>4  
z<-1i  
mode(x) # informa o tipo do objeto  
mode(y)  
mode(e)  
mode(z)
```

Esses são apenas alguns dos muitos comandos básicos que você pode experimentar no R para começar a se familiarizar com a linguagem e seu ambiente de programação. Conforme você avança, você pode explorar mais funções, estruturas de dados e pacotes para realizar análises mais avançadas e complexas.

Repare que em um dos exemplos usamos o símbolo “#” para inserir um comentário “Exercitando meus primeiros cálculos com R” e que eu pulei uma linha para linha de comando. Isso torna o código mais fácil de ler e permite que você lembre o que está sendo realizado naquela parte do seu script. Observe também que os resultados das operações foram exibidos no painel “Console”. É nesta parte que os códigos são executados, ali temos R básico em ação. Excelente, você já criou seu primeiro código em R!

Agora que você viu que é possível realizar cálculos com R e que já conhece um pouco sobre como criar um script vamos falar sobre a estrutura da linguagem. Entender como as funções do R se comportam vai ajudá-lo a ganhar um pouco de confiança nestes primeiros passos.

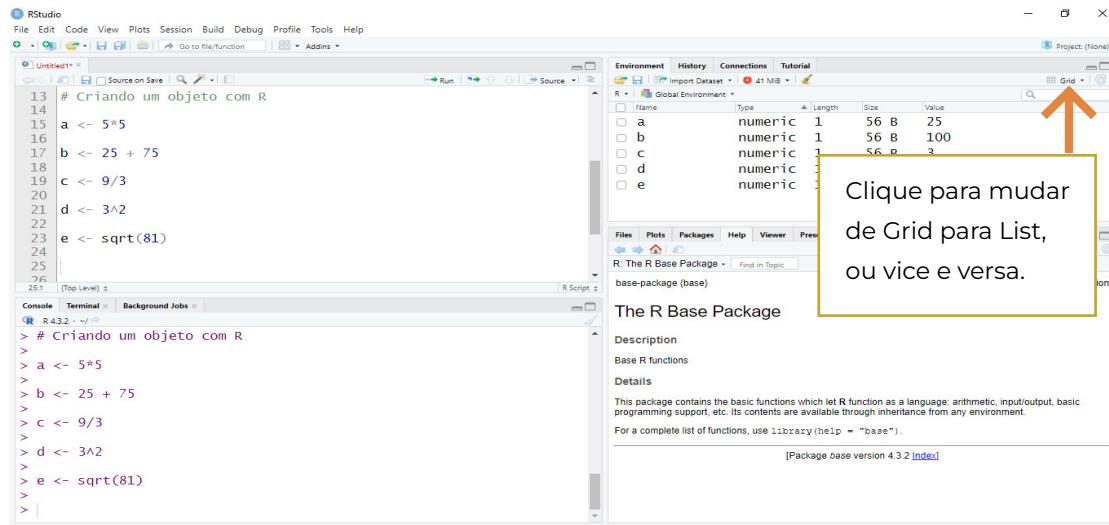
Vamos criar um objeto com R?

```
# Criando um objeto com R
a <- 5*5
b <- 25 + 75
c <- 9/3
d <- 3^2
e <- sqrt(81)
```

Usando as mesmas operações básicas vamos criar os objetos a, b, c, d, e. Para criar um objeto utilizamos o operador de atribuição “<‐” que parece uma seta. Como esse operador não é algo usual de digitarmos usamos o atalho de teclado Alt + -. Faça o teste no seu RStudio, ao usar o atalho de teclado segurando a tecla Alt e teclando o sinal de “‐” a IDE automaticamente cria o operador de atribuição “<‐” e o cerca com espaços. Aprenda esse atalho, pois isso vai tornar a sua capacidade de escrever códigos muito mais rápida e fluida.

Para interpretar o código que você criou leia: “a recebe 5 \* 5” e assim sucessivamente para todos os objetos que você criou. Veja também que agora R criou no painel “Environment” os 5 objetos (a, b, c, d, e) indicando que eles são do tipo numérico, com tamanho (length) 1 e com os valores das operações realizadas.

No seu RStudio ele pode estar exibindo no painel “Environment” apenas os nomes dos objetos e os valores. Para que ele mostre todos esses detalhes clique no botão na parte superior ao lado direito que de estar escrito “List” e mude para “Grid” e veja como ele mostra os detalhes dos objetos como mencionado anteriormente.



**Figura 26.** Criando objetos e usando as funções Grid e List para conhecer o conteúdo dos objetos

## IMPORTANTE

Aqui precisamos consolidar o primeiro ponto essencial da estrutura da linguagem R, o objeto. Lembre-se do significado do objeto “a recebe  $5 * 5$ ”. A estrutura de criação de um objeto segue sempre esse padrão:

**nome\_do\_objeto <- valor**

Um objeto pode receber um valor único, como estamos exercitando nestes primeiros passos, mas pode receber um conjunto de dados completo, como uma tabela do MS Excel, um banco de dados de um sistema de informação, uma lista de valores, um mapa, um gráfico, enfim uma variedade de opções que você vai conhecendo aos poucos.

Gradualmente você está começando a entender um pouco com R se comporta e agora vamos tratar de algo muito importante: chamar uma função em R. As funções realizam as operações necessárias para manipulação dos dados e possuem um padrão, mostrado a seguir:

```
nome_da_funcao(argumento1 = valor1, argumento2 = valor2)
```

Para exercitar esse padrão vamos trabalhar com um conceito clássico da saúde pública: taxa de mortalidade. A taxa de mortalidade é calculada dividindo-se o total de óbitos, em determinado período, pela população calculada para a metade do período. Para isso vamos usar um exemplo.

**Exemplo:**

No Brasil em 2022 ocorreram 63.452 óbitos por Covid-19. Sabendo que a população brasileira em julho de 2022 (metade do ano) era de 203.062.512 habitantes, qual foi a taxa de mortalidade por Covid-19?

Vamos exercitar nossos conhecimentos fazendo esse cálculo no R. Primeiro vamos dividir o número de mortes pela população brasileira.

Código:

```
# Cálculo da taxa de mortalidade por Covid-19 no Brasil  
# Exercitando as operações básicas  
# Cálculo inicial  
63452/203062512  
Agora vamos aplicar a constante 100.000 habitantes
```

Código:

```
# Usando a constante 100.000 habitantes  
(63452/203062512)*100000
```

Nas etapas anteriores executamos as operações básicas que exercitamos no início desta etapa, agora vamos treinar criar objetos no R para fazer a mesma operação. Vamos criar os objetos mortes\_covid19 e populacao\_brasil e vamos atribuir o cálculo a um outro objeto taxa\_mortalidade.

Código:

```
# Usando os objetos  
mortes_covid19 <- 63452  
populacao_brasil <- 203062512  
taxa_mortalidade <- (mortes_covid19/populacao_brasil)*100000  
taxa_mortalidade
```

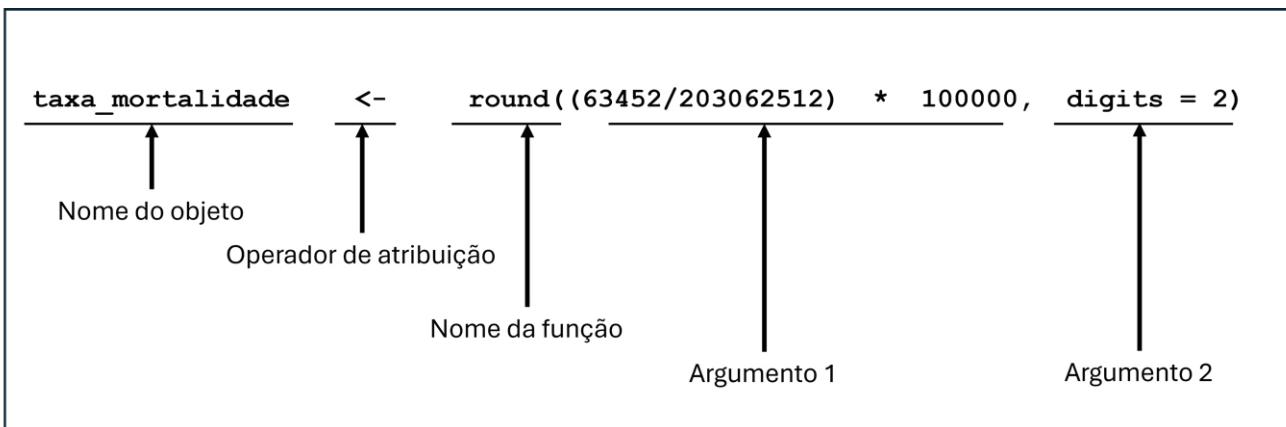
Veja que nessa parte do código usamos o objeto taxa\_mortalidade duas vezes. Quando se trabalha com objetos R apenas guarda o valor do objeto, mas não exibe seu valor para isso, repetimos o nome do objeto para que ao executar R possa exibir o valor no painel “Console”.

Agora vamos exercitar o uso de uma função. Para isso aplicaremos a função do R básico `round()`. Para isso vamos recordar do padrão que uma função R tem, nome da função, argumento 1, argumento 2 e assim sucessivamente. Veja no código:

Código:

```
# Usando uma função para arredondar a taxa para 2 casas decimais
taxa_mortalidade <- round((63452/203062512)*100000, digits = 2)
```

Vamos analisar o código que fizemos a partir da figura a seguir:



**Figura 27.** Estrutura de um código usando uma função na linguagem R

Veja como todos os elementos que aprendemos até agora são usados para executar a tarefa de calcular a taxa de mortalidade. Com essa nova linha de código que fizemos podemos aprender mais uma coisa sobre a linguagem R. Repare que usamos o mesmo nome de objeto, mas dessa vez fizemos uma nova operação de cálculo arredondando as casas decimais. Quando você repete o nome do objeto R simplesmente sobrescreve o novo valor. Então agora só temos um objeto `taxa_mortalidade`, mas agora com o novo valor com apenas duas casas decimais. Por isso, ao criar objetos é importante saber se você quer criar um novo, ou simplesmente substituir o anterior.

Com isso concluímos a primeira etapa de conceitos iniciais de uso da linguagem R. Todo esse código feito nessa etapa está explicado no vídeo (link do vídeo de primeiros passos) passo a passo. Assista e treine no seu próprio computador.

Na próxima parte deste módulo trataremos dos erros mais comuns, para que você possa saber o que fazer se as coisas não correrem bem. Não se preocupe, os erros são frequentes no início e acontecem com todos, sem exceção, mas com a prática você vai perceber que eles podem ser resolvidos e que existe uma infinidade de formas de buscar ajuda.

## Erros e avisos mais comuns sobre a instalação e fluxo de trabalho

O processo de instalação tanto do R básico quanto do RStudio é simples e similar a instalação de outros programas que todos estamos habituados a realizar. Desta forma, a ocorrência de erros é pouco frequente. No entanto, como alertamos no início deste módulo ao tentar instalar esses programas em redes corporativas como de Secretarias estaduais, municipais ou outras instituições de saúde é importante verificar se você possui as permissões de administrador do seu computador, ou se é necessário solicitar que a equipe de tecnologia de informação se encarregue do processo de instalação.

O R básico é um software de código aberto e não há restrições de uso em ambientes de instituições públicas, então vocês não devem ter nenhum receio em instalar devido a licenças. RStudio é uma IDE (ambiente integrado de desenvolvimento), como comentamos no início do módulo, que é licenciada pela empresa Posit Software, PBC. No entanto, o software é de código aberto e pode ser usado também em ambientes de instituições públicas ou privadas, sob os termos da licença pública geral GNU (<https://www.gnu.org/licenses/agpl-3.0.txt>).

O processo de instalação de pacotes também é simples e não apresenta erros de forma habitual. No entanto, se você estiver executando o processo dentro de redes corporativas, como as comentadas anteriormente, você pode se deparar com alguns erros relacionados aos proxys destas redes. Este tipo de erro pode aparecer desta forma:



```
Console Terminal × Background Jobs ×
R 4.3.2 · ~/
Warning in install.packages :
  unable to access index for repository https://cran.rstudio.com/src/contrib:
  não foi possível abrir a URL 'https://cran.rstudio.com/src/contrib/PACKAGES'
```

**Figura 28.** Erro ao baixar pacotes por falha da conexão com a internet.

Nestes casos, entre em contato com a equipe de redes da sua instituição e verifique qual seu perfil de acesso a links de internet para corrigir esse problema. Em redes domésticas (como as da sua casa) você não vai encontrar nenhuma dificuldade em obter os pacotes.

Mesmo que as redes corporativas não permitam o acesso a instalação diretamente via RStudio você ainda pode trabalhar com os pacotes fazendo o download manual diretamente do CRAN ou de um arquivo local, salvo em algum lugar do seu computador. O vídeo de instalação de pacotes (link do vídeo de instalação de pacotes) mostra como fazer essa instalação e oferece algumas dicas adicionais.

Em relação ao fluxo de trabalho com R alguns erros podem aparecer, especialmente neste início, pela falta de familiaridade com uso de uma linguagem de programação. Pela nossa experiência em treinar e acompanhar várias pessoas que estão iniciando

em R, mais da metade dos erros estão relacionados a digitação do código.

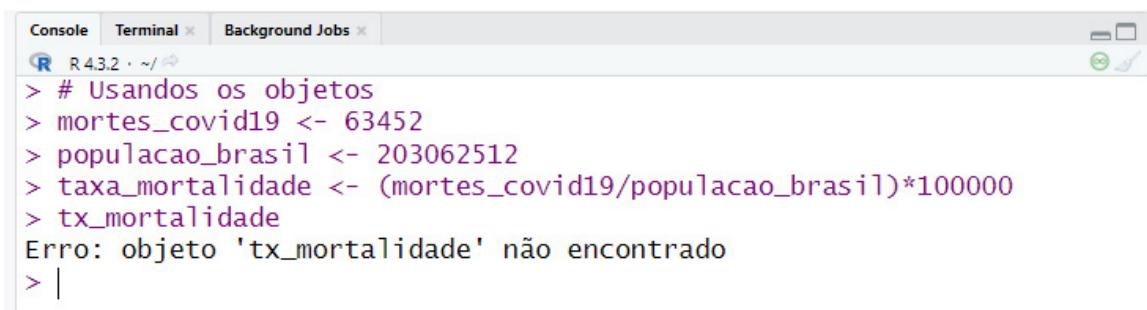
R é extremamente sensível aos erros de digitação. A linguagem propõe uma troca com você, R e RStudio farão todo o trabalho tedioso de manipulação dos dados e cálculos complexos, em contrapartida você precisa ser preciso nas suas instruções.

Vamos ver um exemplo, utilizando o código que escrevemos na etapa anterior.

Código:

```
# Usando os objetos  
mortes_covid19 <- 63452  
populacao_brasil <- 203062512  
taxa_mortalidade <- (mortes_covid19/populacao_brasil)*100000  
tx_mortalidade
```

Se executarmos o código acima R nos devolverá com o erro a seguir:



The screenshot shows the RStudio interface with the 'Console' tab selected. The console window displays the following R code and its execution results:

```
Console Terminal × Background Jobs ×  
R 4.3.2 · ~/  
> # Usando os objetos  
> mortes_covid19 <- 63452  
> populacao_brasil <- 203062512  
> taxa_mortalidade <- (mortes_covid19/populacao_brasil)*100000  
> tx_mortalidade  
Erro: objeto 'tx_mortalidade' não encontrado  
> |
```

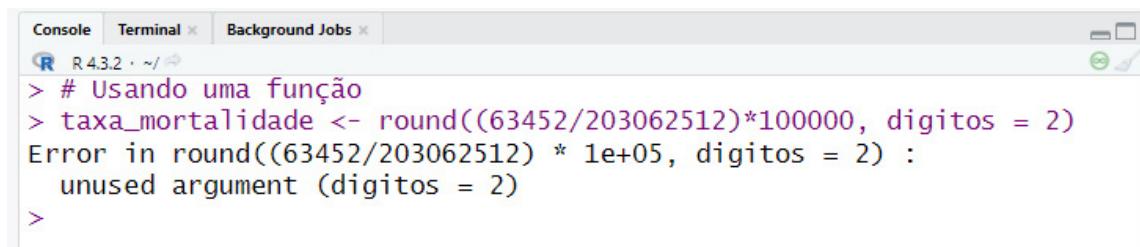
**Figura 29.** Erro de nome de objeto por digitação incorreta

Repare que o objeto taxa\_mortalidade foi escrito como tx\_mortalidade e por isso R não pode encontrá-lo para executar a operação de mostrar o valor calculado. Por isso, neste início é comum que você cometa alguns destes erros de digitação, mas não se desespere, revise seu código, veja os exemplos do e-book, os códigos do curso e os vídeos e provavelmente você vai conseguir identificar e corrigir o problema. O mesmo erro de digitação pode ocorrer nos nomes das funções.

Código:

```
# Usando uma função para arredondar a taxa para 2 casas decimais  
taxa_mortalidade <- round((63452/203062512)*100000, digitos = 2)
```

Veja que ao executar o código acima o argumento digits = 2 foi escrito da forma dígitos = 2 e o R devolveu um erro de “unused argument (dígitos = 2)”. Este erro indica que um dos argumentos da função não foi utilizado.



```
Console Terminal × Background Jobs ×
R 4.3.2 · ~/
> # Usando uma função
> taxa_mortalidade <- round((63452/203062512)*100000, dígitos = 2)
Error in round((63452/203062512) * 1e+05, dígitos = 2) :
  unused argument (dígitos = 2)
>
```

**Figura 30.** Erro de argumento da função por falha na digitação

Um outro erro comum está associado a distinção que o R faz de letras minúsculas e maiúsculas, pois ele é “case sensitive”, ou seja caso os argumentos estejam escritos com letras maiúsculas e minúsculas ele entende de forma diferente como informações distintas, veja o exemplo.

Código:

```
mortes_covid19 <- 63452
populacao_brasil <- 203062512
taxa_mortalidade <- (mortes_covid19/populacao_brasil)*100000
TAXA_MORTALIDADE
```

No R ele mostrará o seguinte erro:



```
R 4.3.3 · ~/
>
> mortes_covid19 <- 63452
>
> populacao_brasil <- 203062512
>
> taxa_mortalidade <- (mortes_covid19/populacao_brasil)*100000
>
> TAXA_MORTALIDADE
Error: object 'TAXA_MORTALIDADE' not found
>
```

**Figura 31.** Erro de objeto não encontrado por uso incorreto de letras maiúsculas

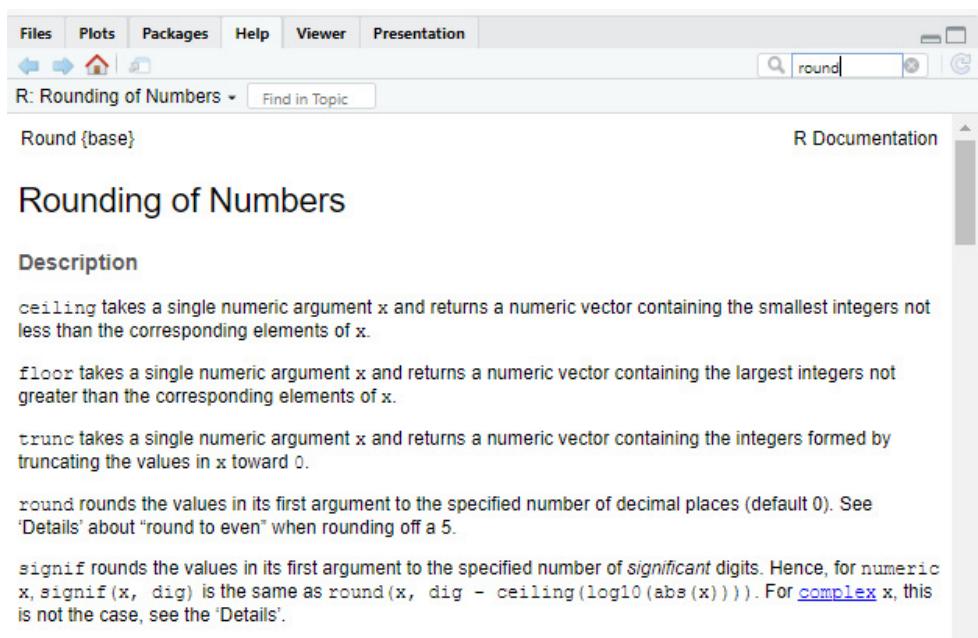
Ou seja, ‘TAXA\_MORTALIDADE’ not found, pois no código ela foi digitada com letra minúscula e ele entende que ‘taxa\_mortalidade’ é diferente de ‘TAXA\_MORTALIDADE’

Nos dias atuais, a popularização dos grandes modelos de inteligência artificial como ChatGPT® ou Gemini® é muito mais fácil pedir ajuda para encontrar erros em códigos R, ou qualquer outra linguagem de programação. Além disso, uma busca no Google® com o erro também pode resolver, além dos fóruns da comunidade R e os tópicos de ajuda.

No vídeo sobre **erros mais comuns**, tem uma explicação detalhada destes erros e como buscar ajuda para resolvê-los, assista e o revisite sempre que tiver dúvidas. A cada módulo teremos uma seção de erros mais comuns para ajudar você a continuar seu desenvolvimento com a linguagem R.

## Tópicos de ajuda e consulta a comunidade de usuários de R

Outra forma de minimizar os erros é buscar compreender como cada função se comporta, quais seus argumentos e qual seu uso. Para isso R tem uma seção de ajuda que fica na aba “Help”. Para exercitarmos vamos ver como funciona uma função que já utilizamos neste módulo, a função “round()”. Para fazer isso digite o nome da função na caixa de pesquisa, como mostrado na figura a seguir.



**Figura 32.** Tela de ajuda do RStudio

Repare que R mostra a qual pacote a função pertence, neste caso {base}, apresenta com detalhes os argumentos da função, seus usos com exemplos, além da referência sobre os cálculos realizados. Muitas das suas dúvidas futuras com R podem ser esclarecidas em uma consulta a essa aba.

Além disso, R possui uma comunidade muito ativa de usuários e desenvolvedores com diversos sites de apoio que você pode buscar para avançar no seu aprendizado. No vídeo sobre **tópicos da ajuda**, você pode conferir algumas dicas e aprender um pouco mais sobre como buscar ajuda na comunidade de R.

Muito bem, chegamos ao fim do módulo 1! Esta primeira parte foi para que você se familiarize com as interfaces do R básico e do RStudio, saiba como instalar os pacotes necessários para análise e comece a compreender um pouco de como criar um script e escrever as suas primeiras linhas de código.

No próximo módulo vamos iniciar as primeiras análises com a importação de bancos de dados para dentro do RStudio e entender um pouco sobre a organização de dados. Até o próximo módulo!

## REFERÊNCIAS

PARADIS, Emmanuel. **R for Beginners. Institut des Sciences de l'Evolution.** Université Montpellier II, 2005.

Schmuller, Joseph. **Análise Estatística com R Para Leigos.** Alta Books.

Wickham, Hadley, Çetinkaya-Rundel, Mine, Grolemund Garrett. **R for Data Science,** 2<sup>a</sup> edição, disponível em <https://r4ds.hadley.nz/>

## FICHA TÉCNICA

EQUIPE DO PROGRAMA DE FORTALECIMENTO DA EPIDEMIOLOGIA NOS SERVIÇOS DE SAÚDE – PROFEPI

### Coordenação Geral

Ethel Leonor Noia Maciel

*Secretaria de Vigilância em Saúde e Ambiente - SVSA/MS*

Guilherme Loureiro Werneck

*Diretor de Ações Estratégicas de Epidemiologia e Vigilância em Saúde e Ambiente - Daevs/SVSA/MS*

Vivian Siqueira Santos Gonçalves

*Coordenadora-Geral de Desenvolvimento da Epidemiologia em Serviços - CGDEP/Daevs/SVSA/MS*

### Equipe Técnica

Amanda Maciel de Quadros - CGDEP/Daevs/SVSA/MS

Fátima Sonally Sousa Gondim - CGDEP/Daevs/SVSA/MS

Maryane Oliveira Campos - CGDEP/Daevs/SVSA/MS

Sarah Yasmin Lucena Gomes - CGDEP/Daevs/SVSA/MS

Barbara Pimentel - CGDEP/Daevs/SVSA/MS

### Docentes

Elisângela Aparecida da Silva Lizzi - Universidade Tecnológica Federal do Paraná (UTFPR)

Diego Spinoza dos Santos - Secretaria Municipal da Saúde de Curitiba (SMS de Curitiba)

### Criação Digital – diagramação e edição de vídeos

Lauro Adolfo Gontijo dos Santos - CGDEP/Daevs/SVSA/MS

Otávio Francisco Batista Martins - CGDEP/Daevs/SVSA/MS

ORGANIZAÇÃO PAN-AMERICANA DE SAÚDE/ ORGANIZAÇÃO MUNDIAL DA SAÚDE (OPAS/OMS)

### Coordenação Geral

Socorro Gross

*Representante da OPAS/OMS no Brasil*

Sebastian Garcia Saiso

*Diretor do Departamento de Evidência e Inteligência para Ação em Saúde da OPAS/OMS em Washington D. C., EUA*

### Equipe Técnica

Juan Cortez-Escalante - OPAS/OMS, Brasil

Mônica Diniz Durães - OPAS/OMS, Brasil