



**Cursos Integrados
em Vigilância em Saúde**

Curso —

**Análise de dados para a vigilância
em saúde – Curso Básico**

Módulo 1 - Introdução à análise de dados com R – Parte I

UNIVERSIDADE FEDERAL DE SANTA CATARINA

Reitor Irineu Manoel de Souza

Vice-Reitora Joana Célia dos Passos

Pró-Reitora de Pós-graduação Werner Kraus

Pró-Reitor de Pesquisa e Inovação Jacques Mick

Pró-Reitor de Extensão Olga Regina Zigelli Garcia

CENTRO DE CIÊNCIAS DA SAÚDE

Diretor Fabrício de Souza Neves

Vice-Diretora Ricardo de Souza Magini

DEPARTAMENTO DE SAÚDE PÚBLICA

Chefe do Departamento Rodrigo Otávio Moretti Pires

Subchefe do Departamento Sheila Rúbia Lindner

Coordenadora do Curso Alexandra Crispim Boing

INSTITUTO TODOS PELA SAÚDE (ITPS)

Diretor presidente Jorge Kalil (Professor titular da Faculdade de Medicina da Universidade de São Paulo; Diretor do Laboratório de Imunologia do Incor)

ASSOCIAÇÃO BRASILEIRA DE SAÚDE COLETIVA (ABRASCO)

Presidente Rosana Teresa Onocko Campos

EQUIPE DE PRODUÇÃO

Denis de Oliveira Rodrigues

Kamila de Oliveira Belo

Marcelo Eduardo Borges

Oswaldo Gonçalves Cruz

Alexandra Crispim Boing

Antonio Fernando Boing



Módulo 1 - Introdução à análise de dados com R – Parte I

Curso

Análise de dados para a vigilância
em saúde - Curso Básico

Dados Internacionais de Catalogação-na-Publicação (CIP)

- I61 Introdução à análise de dados com R – Parte I / Marcelo Eduardo Borges, Oswaldo Gonçalves Cruz, Denis de Oliveira Rodrigues, Kamila de Oliveira Belo. – Santa Catarina ; São Paulo ; Rio de Janeiro : UFSC ; ITPS ; Abrasco; 2022. 45p. (Análise de dados para a vigilância em saúde - Curso Básico; Módulo 1).

Publicação Online
DOI: 10.52582/curso-analise-dados-vigilancia-modulo1

1. Vigilância em saúde 2. Análise de dados 3. Software R I. Título

Sumário

1. Introdução ao R	08
2. Instalando o R e o Rstudio	10
2.1 R e Rstudio	10
2.2 Instalação do R e do R Studio.....	11
2.3 Como baixar e instalar o R?	12
2.4 Como baixar e instalar o Rstudio?.....	15
3. Escrevendo no painel console	18
4. Escrevendo um Script	20
5. Instalando pacotes e utilizando-os para suas análises	22
5.1 Pacotes: o que são ?.....	22
5.2 Instalando e carregando pacotes	23
5.3 Consultando pacotes carregados e sua versão.....	26
5.4 Atualização dos pacotes	28
6. Boas práticas de organização	29
6.1 Orientações gerais.....	30
6.2 Diretório de trabalho	36
6.3 Utilização da ferramenta “Projetos”.....	40

Introdução ao software R

Nos últimos anos, a sociedade vem acumulando importantes desafios em saúde que evidenciam e reforçam a importância da vigilância em saúde e de suas ações de investigação e nas medidas de prevenção e controle de doenças. As respostas sanitárias baseadas em informações qualificadas e oportunas são essenciais para o controle das doenças, para a comunicação transparente com a sociedade e para se obter melhores desfechos em saúde. Ao mesmo tempo, sabemos como é agitada a rotina de trabalho na vigilância e como é complexo o gerenciamento de todos os dados em saúde. Assim, pensando em ajudar na produção da informação em saúde de forma qualificada e ágil este curso foi estruturado para você.

Por exemplo, você:

- Já imaginou ter todas as suas rotinas de análise de vigilância em saúde prontas com um clique?
- Já se perguntou como pegar informações diretamente do Datasus e transformá-las em um gráfico, tabela ou série histórica de forma automatizada? Isso mesmo que você leu: automatizada!
- Pensou em produzir diversos informes epidemiológicos de todos os agravos de notificação compulsória em menos de 2 minutos?
- Imaginou ter os informes epidemiológicos de todos agravos e notificação compulsória de forma instantânea online?



Isso parece revolucionário, não é mesmo? Não só parece, como é! Você conseguirá em pouco tempo qualificar, agilizar e facilitar seu trabalho. Para isso nós utilizaremos a linguagem de programação R. Sim, você precisará dedicar um pouco de tempo para instalar o programa e aprender comandos básicos. Mas a recompensa a esse esforço será imensurável, poupando-lhe muitas e muitas horas de trabalho a partir de agora e lhe habilitando a produzir análises muito interessantes em sua rotina de trabalho.

Ao final deste módulo, você será capaz de:

1. Conceituar a linguagem R .
2. Apontar as vantagens de uma linguagem de programação para análise de dados.
3. Instalar o R e o RStudio .
4. Executar comandos básicos na interface gráfica RStudio .
5. Criar e salvar *scripts*.
6. Instalar pacotes.
7. Compreender e praticar boas práticas de organização de arquivos para análise de dados.

Neste curso iremos te ensinar a realizar análise de dados para vigilância em saúde utilizando o R . Ensinaremos que uma vez dominada a linguagem, você poderá operacionalizar análises de uma forma muito mais completa e sem retrabalhos.

Aproveite o curso!

1. Introdução ao R

O que é o R?

R é uma **linguagem de programação** que permite trabalhar com dados, realizar análises, gerar gráficos, entre diversas outras aplicações.

O seu uso vem apresentando um crescimento considerável nos últimos anos, sendo uma das linguagens de análises estatísticas mais utilizadas no mundo.

Por ser uma linguagem de programação, ela permite que você possa escrever um **roteiro de instruções** (chamado de *script*), que serão interpretadas e executadas por um computador.

Esta linguagem de programação funciona por meio de uma sequência de **comandos escritos** (ou *códigos*), pelos quais você consegue realizar uma diversidade de tarefas.

Por que R?

Completamente gratuito e acessível

- É uma linguagem de programação *gratuita* e de livre distribuição (não são cobradas taxas de licença).
- Pode ser utilizada em todos os sistemas operacionais: Windows, Linux, Mac OS e Unix.
- Várias extensões para análises de dados são desenvolvidas diariamente e disponibilizadas gratuitamente.

Colaboração e compartilhamento

- Possui uma ampla e crescente comunidade de compartilhamento de códigos e colaboração;
- Na epidemiologia o R é muito utilizado, permitindo desde análises simples a mais robustas que reduzem o tempo de resposta a situações que exigem ações rápidas de saúde pública.
- É amplamente utilizado em universidades, empresas e instituições públicas do mundo inteiro.
- Usuários do mundo todo podem elaborar seus próprios códigos para atenderem a demandas específicas e compartilhá-los *online* com outros usuários.
- Vários pesquisadores e autores de livros na área de estatística, epidemiologia, ecologia, demografia, geografia disponibilizam seu material na internet, inclusive a equipe de desenvolvimento da linguagem.

Automatização de tarefas e reproduzibilidade

- Os *scripts* feitos no R podem ser reproduzidos facilmente em diferentes computadores.
- Tarefas realizadas rotineiramente podem ser automatizadas, economizando o tempo empenhado em tarefas repetitivas e diminuindo o número de cliques que, frequentemente, são necessários para repetir uma tarefa em outros softwares.
- Permite criar relatórios em diversos formatos - como PDF, HTML, Word e LaTeX de forma automatizada e mais ágil.
- Permite criar dashboards, sites e gráficos interativos de forma automatizada.
- O R permite uma integração com diversas plataformas, programas, base de dados e até mesmo outras linguagens de programação.
- Bases de dados muito grandes não podem ser abertas em softwares convencionais. No R você poderá abri-las e analisá-las.

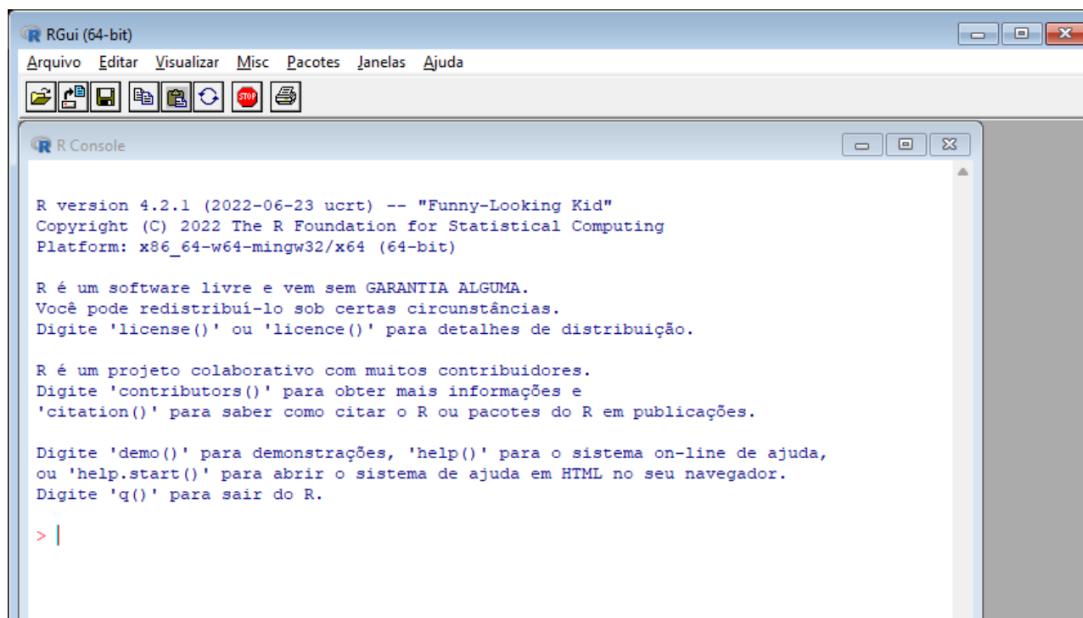
2. Instalando o R e o RStudio

2.1 R e RStudio

Para utilizar o R em seu computador, iremos fazer o *download* de dois programas: o R e o RStudio.

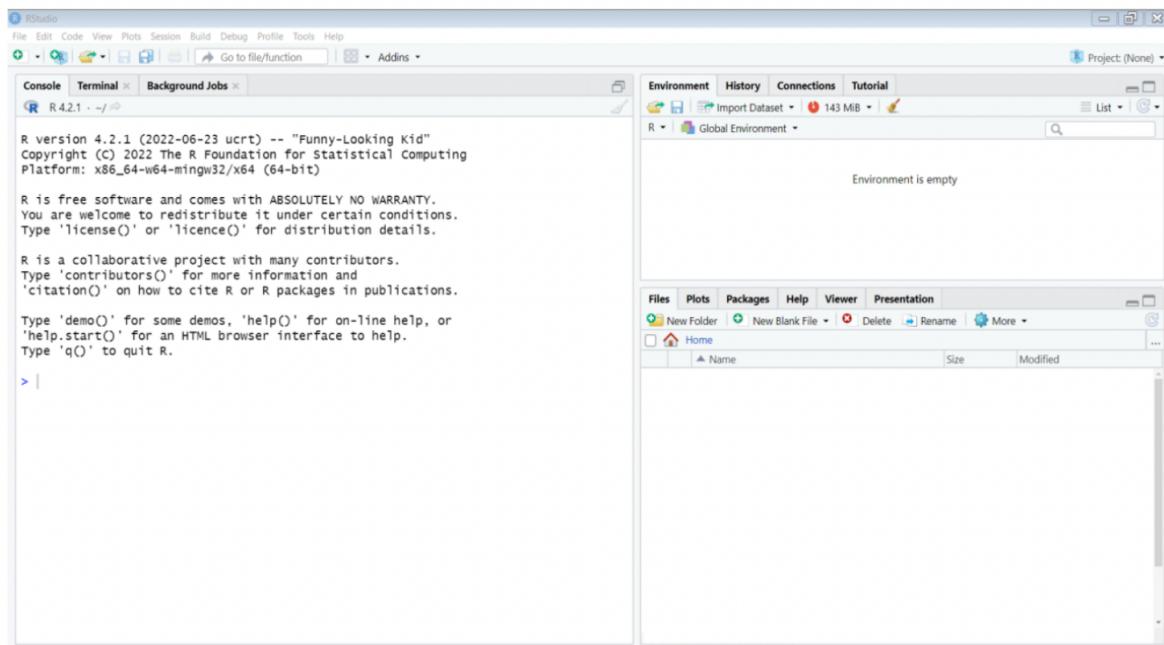
O software R irá configurar o ambiente em sua máquina para que seu computador possa interpretar e executar esta linguagem. Você perceberá que ele se apresenta como uma simples tela com uma linha para inserir comandos, conforme apresentado na Figura 1.

Figura 1: Tela inicial do software R.



Já o programa RStudio apresentará diversas funcionalidades gráficas que irão permitir que você utilize e execute códigos com muita facilidade em uma interface amigável. Ele é um ambiente de desenvolvimento integrado para o R - ou IDE, referente à sigla em inglês *Integrated Development Environment*. Possui diversas funcionalidades que melhoram ainda mais o uso para as análises dos dados da vigilância e o aprendizado do R, com botões, ferramentas, painéis de visualização simultânea e melhor visualização dos arquivos que estão sendo utilizados. Mas ele não funciona sozinho, o RStudio precisa do R. Na Figura 2 temos um exemplo da tela do RStudio.

Figura 2: Tela inicial da IDE RStudio.



2.2 Instalação do R e do Rstudio

Nosso primeiro passo será fazer o *download* e a instalação dos programas R e RStudio em seu computador Windows. Caso você utilize uma máquina com sistemas operacionais Linux ou Mac OS, o mesmo processo pode ser realizado de forma bastante semelhante.



ATENÇÃO

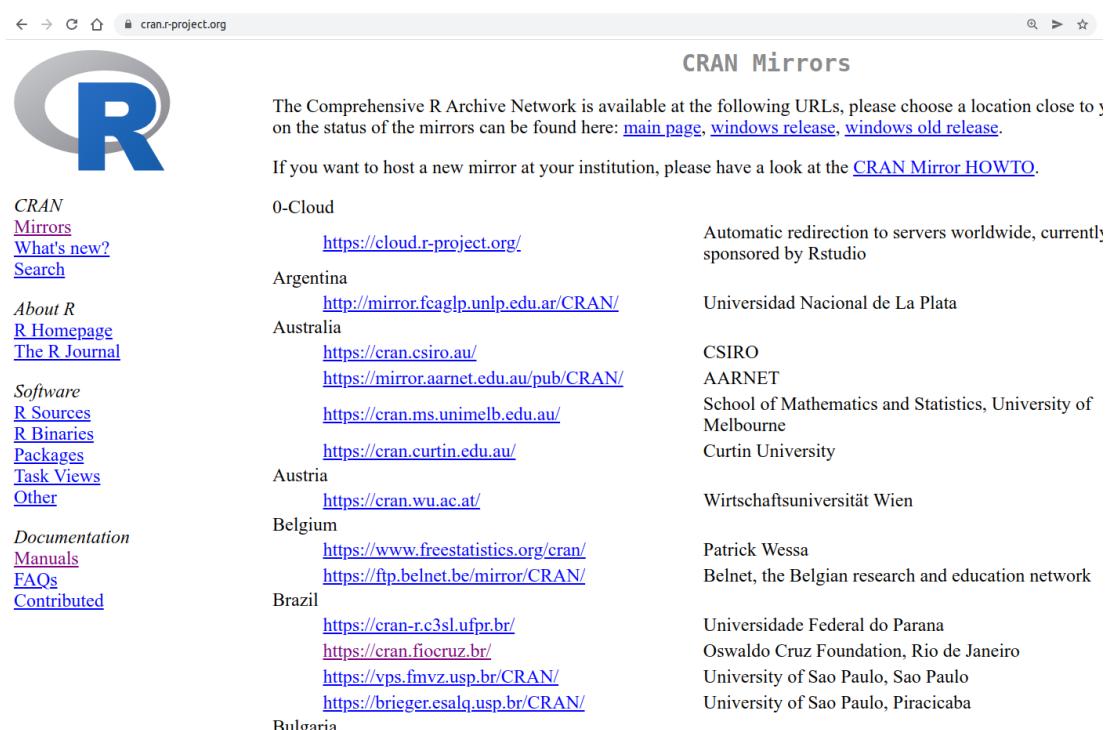
Você pode encontrar dificuldades para instalação se a rede do seu local de trabalho for bloqueada para sites externos. Neste caso, converse com a equipe de suporte em informática e solicite o desbloqueio para o uso do R e RStudio!

2.3 Como baixar e instalar o R?

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 em diversos locais no mundo. Clique em: <https://cran.r-project.org/mirrors.html> e procure na lista que fica à esquerda o país Brasil.

Sugerimos que para este curso você clique no link referente ao repositório da Fiocruz: <https://cran.fiocruz.br/>. Você também encontrará diversos repositórios, conforme a Figura 3, e poderá selecionar o mais próximo do seu local de trabalho ou moradia, se desejar.

Figura 3: Tela de acesso ao repositório CRAN (*The Comprehensive R Archive Network*).



The screenshot shows the 'CRAN Mirrors' page from the CRAN website. At the top, there's a search bar and navigation links. Below the header, there's a large 'R' logo. The main content area is titled 'CRAN Mirrors' and contains the following text:

The Comprehensive R Archive Network is available at the following URLs, please choose a location close to you. The status of the mirrors can be found here: [main page](#), [windows release](#), [windows old release](#).

If you want to host a new mirror at your institution, please have a look at the [CRAN Mirror HOWTO](#).

O-Cloud	https://cloud.r-project.org/	Automatic redirection to servers worldwide, currently sponsored by Rstudio
Argentina	http://mirror.fcaglp.unlp.edu.ar/CRAN/	Universidad Nacional de La Plata
Australia	https://cran.csiro.au/ https://mirror.aarnet.edu.au/pub/CRAN/ https://cran.ms.unimelb.edu.au/ https://cran.curtin.edu.au/	CSIRO AARNET School of Mathematics and Statistics, University of Melbourne Curtin University
Austria	https://cran.wu.ac.at/	Wirtschaftsuniversität Wien
Belgium	https://www.freestatistics.org/cran/ https://ftp.belnet.be/mirror/CRAN/	Patrick Wessa Belnet, the Belgian research and education network
Brazil	https://cran-r.c3sl.ufpr.br/ https://cran.fiocruz.br/ https://vps.fmvz.usp.br/CRAN/ https://brieger.esalq.usp.br/CRAN/	Universidade Federal do Paraná Oswaldo Cruz Foundation, Rio de Janeiro University of São Paulo, São Paulo University of São Paulo, Piracicaba
Bulgaria		

Agora você deverá clicar no link **Download R for Windows** (Figura 4). Caso você utilize outro sistema operacional, é só escolher o link correspondente. Ao clicar, você será redirecionado para outra página na qual deve clicar em **Install R for the first time**.



Figura 4: Tela para download do software R.

The screenshot shows a web browser displaying the CRAN (Comprehensive R Archive Network) homepage at cran.fiocruz.br. The page title is "The Comprehensive R Archive Network". On the left, there's a large R logo and a sidebar with links for CRAN, Mirrors, What's new?, Search, About R, R Homepage, The R Journal, Software, R Sources, R Binaries, Packages, Task Views, Other, Documentation, Manuals, FAQs, and Contributed. The main content area is titled "Download and Install R" and contains instructions for precompiled binary distributions for Windows and Mac users, with links to download R for Linux, macOS, and Windows. Below this, another section titled "Source Code for all Platforms" provides instructions for Windows and Mac users regarding source code compilation and links to various R releases and documentation.

Ao clicar em **Install R for the first time**, abrirá a tela para escolha da versão do R que estará disponível. Clique no *download* no topo da página (Figura 5) - no exemplo abaixo é a versão R 4.2.1 para Windows.

Após o clique será aberta uma janela para a escolha da pasta para salvar o arquivo ou será encaminhado o arquivo executável direto para sua pasta de *download* padrão.



Figura 5: Tela para escolha da versão do software R.

The screenshot shows a web browser displaying the CRAN Windows binary distribution page for R 4.2.1. The URL in the address bar is <https://cran.fiocruz.br>. The main content area is titled "R-4.2.1 for Windows". At the top, there is a prominent red rectangular box highlighting the "Download R-4.2.1 for Windows (79 megabytes, 64 bit)" link. Below this, there are links to "README on the Windows binary distribution" and "New features in this version". A note states that this build requires UCRT, which is part of Windows since Windows 10 and Windows Server 2016. On older systems, UCRT has to be installed manually from [here](#). Another note says if you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the [md5sum](#) of the .exe to the [fingerprint](#) on the master server. There is a "Frequently asked questions" section with links to "Does R run under my version of Windows?" and "How do I update packages in my previous version of R?". A note below says to see the [R FAQ](#) for general information about R and the [R Windows FAQ](#) for Windows-specific information. A "Other builds" section lists patches to this release, the development version, and previous releases. A note to webmasters says a stable link will redirect to the current Windows binary release at [<CRAN MIRROR>/bin/windows/base/release.html](#). At the bottom, it says "Last change: 2022-06-23".

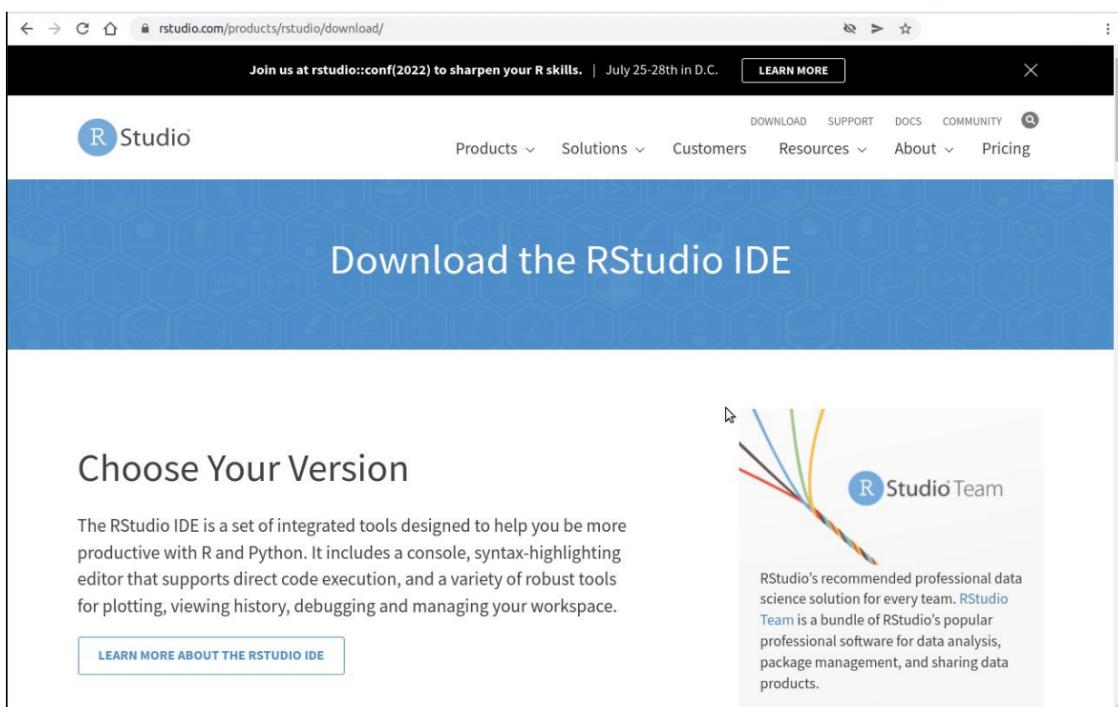
Após o fim do *download*, clique duas vezes no arquivo que foi baixado, e siga as instruções para completar a instalação.

Pronto, agora que você já baixou e instalou o R, precisaremos instalar o software RStudio que nos permitirá utilizar os códigos durante a aula com maior praticidade.

2.4 Como baixar e instalar o Rstudio?

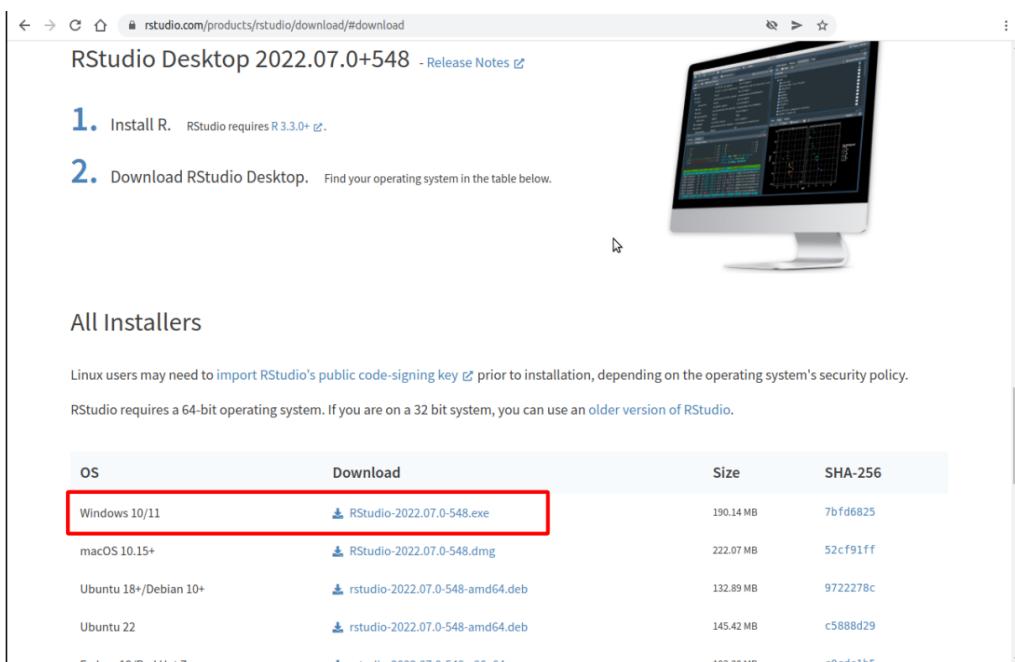
Agora que já baixamos o R, vamos baixar o RStudio. Para fazer o download do programa RStudio, acesse o link: <https://www.rstudio.com/products/rstudio/download/>. Feito isso, você verá a tela conforme a Figura 6.

Figura 6: Plataforma para download da IDE RStudio.



O RStudio possui versões pagas com licença comercial, mas aqui iremos utilizar a de uso livre. Role a página até visualizar o link da versão **RStudio Desktop**, e a licença de código aberto (FREE), conforme Figura 7. Clique no botão de download referente a esta versão. Você será redirecionado para a versão mais atual. Veja na imagem abaixo o link para versões do Windows:

Figura 7: Tela para download da IDE RStudio.



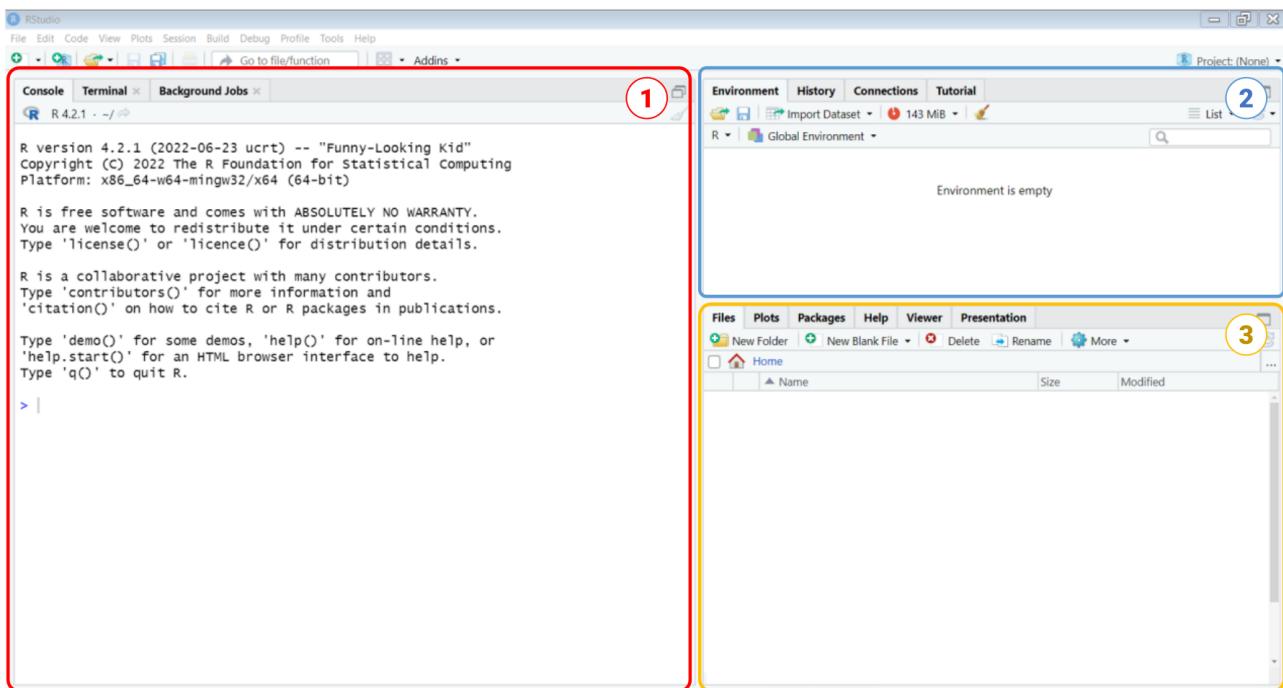
Ao clicar em *download* para **Windows 10/11** você verá aberta uma janela para a escolha da pasta para salvar o arquivo ou será encaminhado o arquivo executável direto para sua pasta de *download* padrão.

Clique no arquivo executável, aguarde a conclusão da instalação e clique sobre “Terminar”. Assim que o *download* for finalizado, abra o arquivo de execução e siga as instruções para completar a instalação.

Para verificar se a instalação foi concluída com sucesso clique no menu **Iniciar** do seu computador e digite **RStudio**, ou acesse por meio do menu iniciar ou por meio da pasta de instalação.

Ao abrir o **RStudio**, veja que é uma interface similar à apresentada na Figura 8 abaixo. Como a grande maioria dos programas, ele possui um menu superior principal e, logo abaixo, um menu de ícones. Além disso, a interface possui três principais painéis que numeramos para facilitar a sua compreensão:

Figura 8: Tela inicial da IDE RStudio .



1 - Painel Console: neste painel é onde os comandos escritos por você serão executados e onde serão mostrados os resultados. Tudo que você solicitar ao R, será interpretado no console, como funções para executar alguma ação.

2 - Painel Environment: neste painel serão guardados os objetos que você criar e importar para o R. Mais à frente, veremos mais detalhes deste painel. Neste painel tem outro muito importante agregado: o painel **History**, que mostra todo o histórico de comandos executados.

3 - Painel Files: neste painel são mostrados os arquivos presentes na pasta de trabalho que você está usando. É possível navegar entre as pastas. Neste painel há outros agregados igualmente importantes:

- painel **Plots**, onde os gráficos são mostrados;
- painel **Packages**, onde são mostrados complementos ao R instalados (chamados de pacotes e os veremos mais à frente);
- painel **Help**, com uma ampla estrutura de ajuda e documentação.

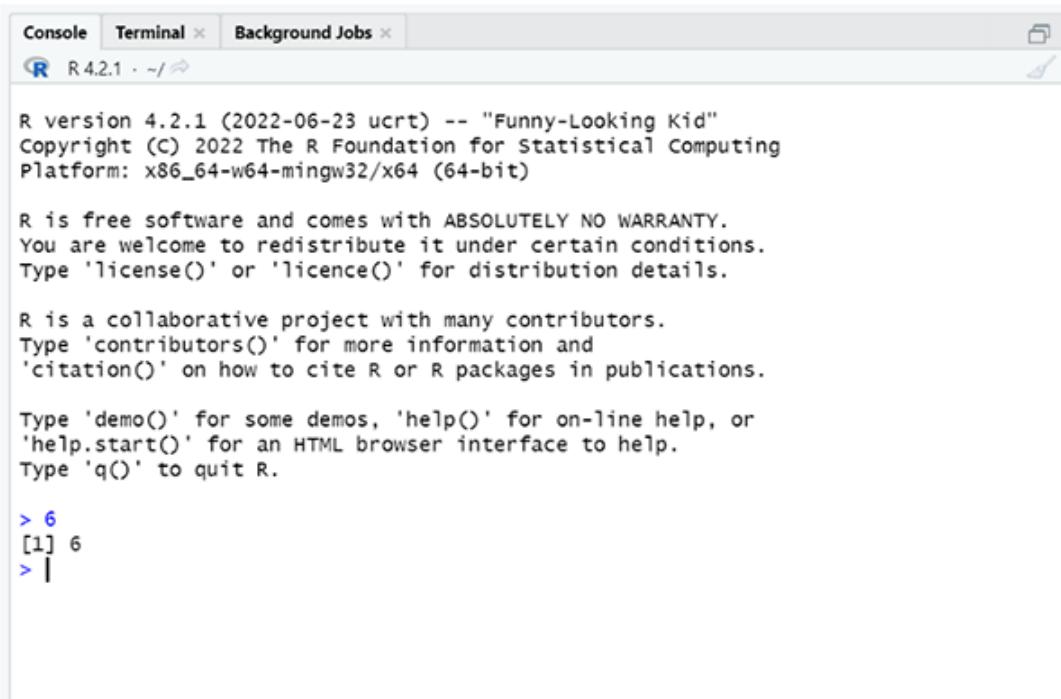
3. Escrevendo no painel console

Tenha em mente: embora pareça muito difícil iniciar uma análise de dados da vigilância com uma linguagem de programação como o R, à medida que você utilizar as ferramentas do curso em sua prática você perceberá que o R segue uma lógica intuitiva e tornará o seu dia a dia mais simples.

Como apresentado anteriormente, o **painel Console** é onde todos os comandos são executados e os resultados são visualizados. No início da linha de comando, quase sempre você encontrará o símbolo de maior que (>). Esse símbolo indica que a linha pode receber algum comando.

Os códigos podem ser escritos e executados diretamente no **Console**. Para isso, basta apenas digitar algum comando e executá-lo apertando a tecla **Enter**. Veja o que acontece quando digitamos um número qualquer, no próprio console (Figura 9).

Figura 9: Tela inicial do console no RStudio .



```
R version 4.2.1 (2022-06-23 ucrt) -- "Funny-Looking Kid"
Copyright (C) 2022 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.

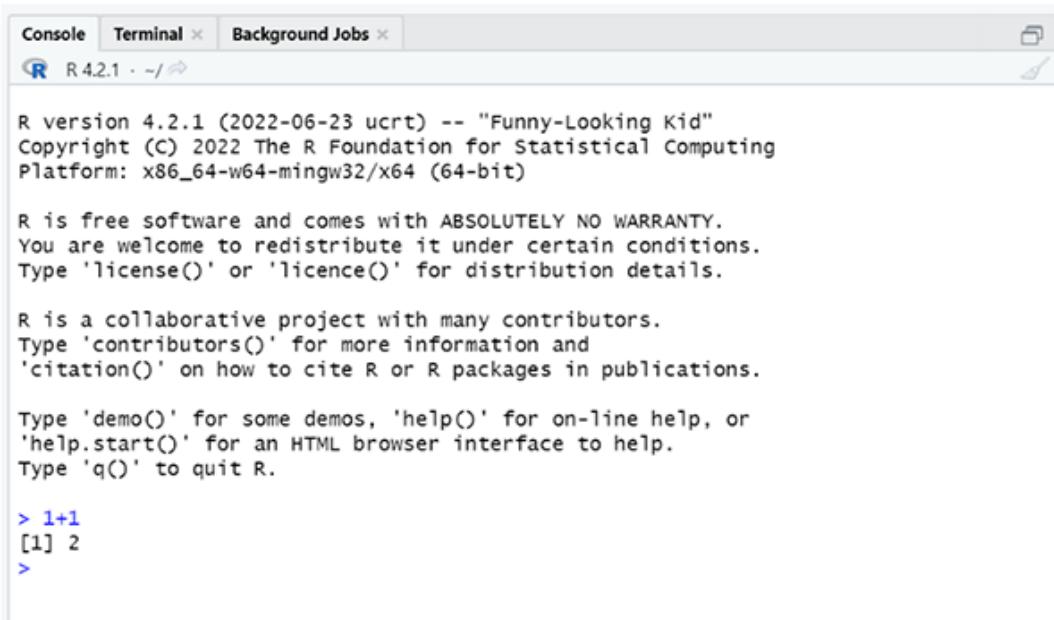
R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

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.

> 6
[1] 6
> |
```

Agora, experimente digitar um comando de soma simples, algo como **1 + 1**, conforme a Figura 10. Perceba que o resultado é mostrado logo abaixo o comando, precedido de [1], que significa a primeira linha do resultado (Figura 10).

Figura 10: Tela inicial do console no RStudio com comandos digitados.



```
Console Terminal × Background Jobs ×
R 4.2.1 - ~/ ◇
R version 4.2.1 (2022-06-23 ucrt) -- "Funny-Looking Kid"
Copyright (C) 2022 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.

> 1+1
[1] 2
>
```

No **R**, comando é toda ação que precisa ser realizada. Essa ação é efetuada por meio de funções que acionamos por meio dos comandos. Veremos mais sobre funções no decorrer do Curso.

4. Escrevendo um Script

Ao usar o painel Console é necessário entender que ele não guarda os comandos executados. Por isso há a estrutura de um *script* que armazena os comandos em um arquivo.

O *script* é um texto escrito em formas de comandos, também chamados de códigos, que o R interpreta como sendo uma ação a ser executada. O *script* é, portanto, um roteiro para a execução de uma rotina no trabalho.

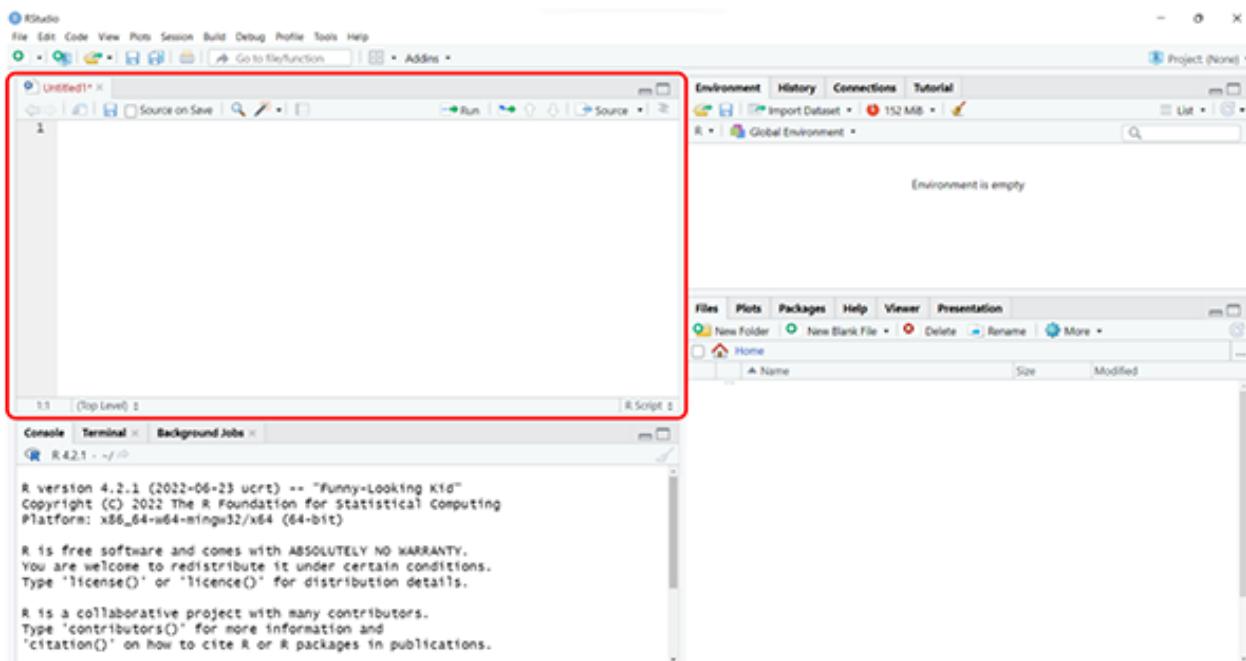
Essa ferramenta de *script* é muito útil para guardar os comandos necessários para uma análise que precisa ser repetida toda semana, por exemplo. Ou ainda um gráfico que precisa ser visualizado todos os dias, usando dados novos que vão sendo recebidos. Ou seja, usando *script*, as tarefas rotineiras ficam menos desgastantes.

No RStudio um novo *script* é criado clicando no Menu superior e seguindo o seguinte caminho:

- File.
- New File.
- R Script.

Um novo painel é adicionado no RStudio, deslocando o painel Console para baixo, destacando o *script* no canto esquerdo superior, conforme a Figura 11.

Figura 11: Tela inicial do *script* para armazenamento dos comandos escritos.



Caso você queira salvar um *script*, basta retornar ao menu superior e seguir clicando em:

- *File*;
- *Save*.

Como todo programa de computador, será aberta uma janela para a escolha da pasta onde o arquivo do *script* será salvo. A extensão salva é **.R**. Mais à frente no curso, vamos aprender exemplos e comandos que podem ser organizados em *scripts*, os chamados códigos.

5. Instalando pacotes e utilizando-os para suas análises

5.1 Pacotes: o que são?

Pacotes (*packages*, em inglês) são bibliotecas contendo funções e dados para o uso no **R**. Muitas vezes eles têm como fim atender finalidades específicas.

Ao instalar o **R** pela primeira vez, diversos pacotes básicos serão instalados e carregados de forma automática na primeira sessão. Durante este curso, iremos instalar diversos pacotes com finalidades específicas para análise de dados epidemiológicos.

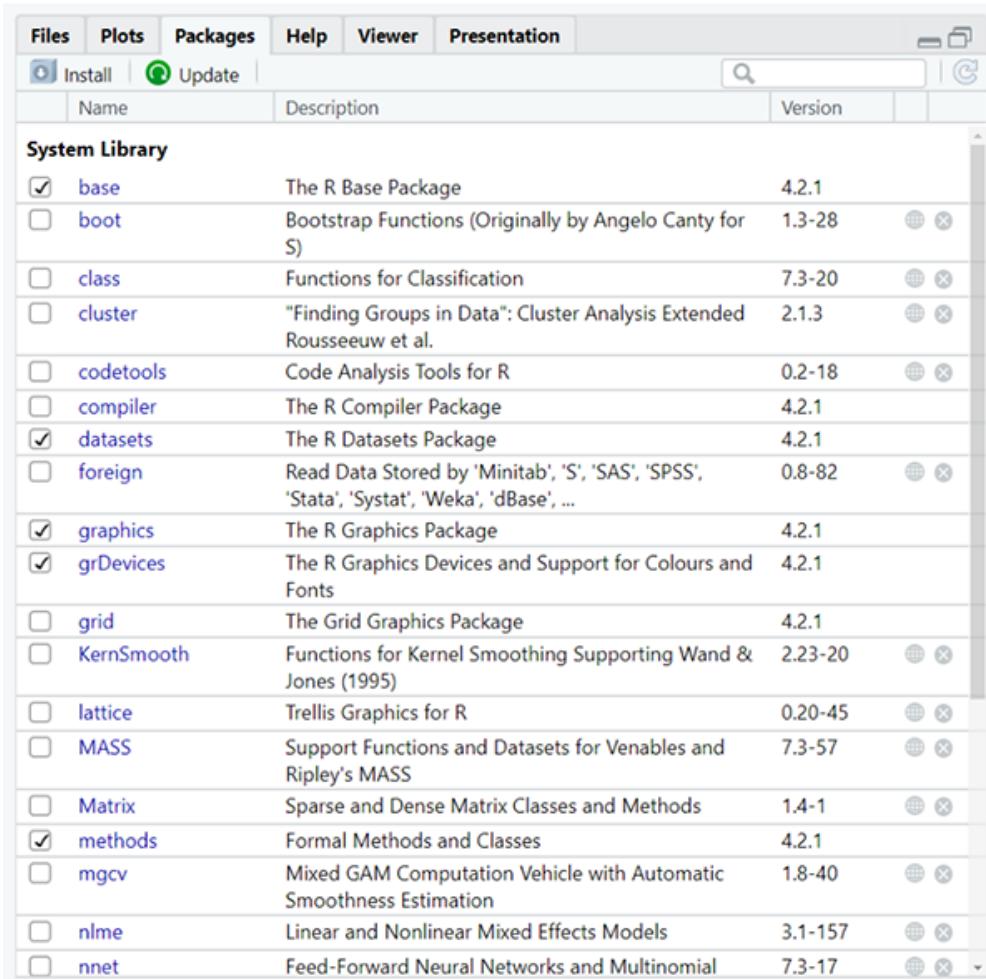
O uso destes pacotes é um dos motivos do grande sucesso e flexibilidade do **R**, visto que usuários do mundo todo trabalham atualmente para desenvolver novos pacotes que atendam necessidades específicas de seus usuários. Em especial, nos últimos anos foram desenvolvidos diversos pacotes que contribuem consideravelmente para o trabalho com dados de saúde pública e vigilância epidemiológica.

5.2 Instalando e carregando pacotes

Pacotes novos podem ser instalados e carregados no `R` de diferentes maneiras: via `RStudio` ou via comandos previamente descritos em um *script*. Além disso, a instalação pode ocorrer por meio de repositórios *online* ou arquivos locais. Neste curso, iremos aprender como instalar pacotes a partir de repositórios *online*.

Para instalar e carregar pacotes pelo `RStudio`, verifique a tela no canto *inferior direito* e escolha a aba *Packages*. Esta aba irá apresentar uma lista com todos os pacotes disponíveis e instalados em sua máquina, assim como uma breve descrição do pacote e sua versão atual (Figura 12).

Figura 12: Tela Packages contendo a lista com todos os pacotes disponíveis e instalados.

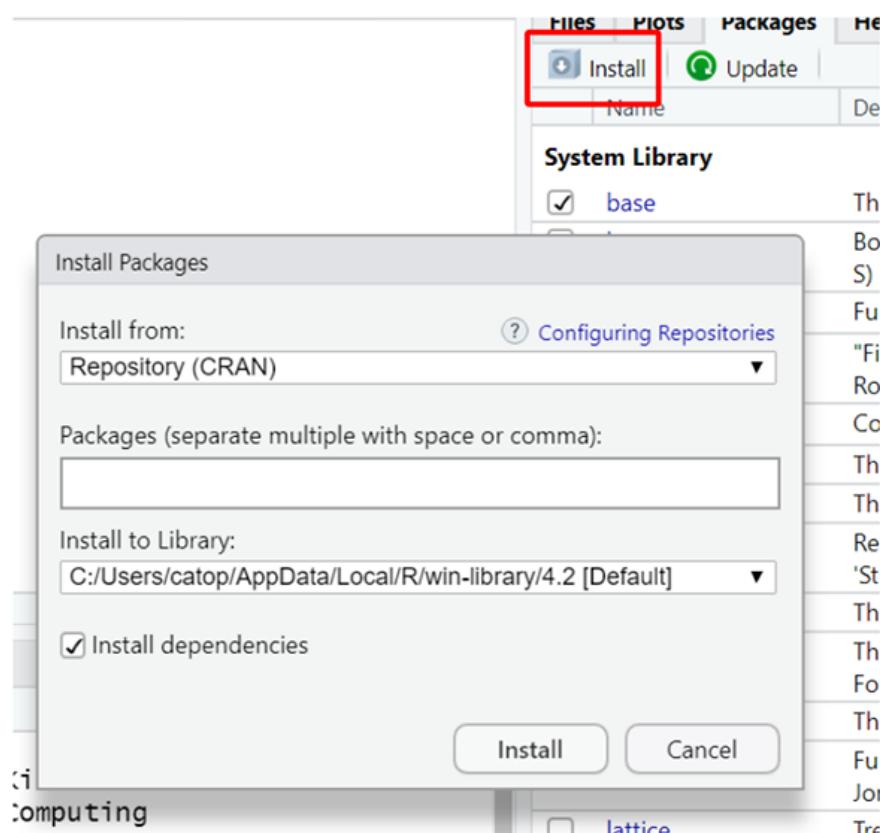


The screenshot shows the RStudio interface with the 'Packages' tab selected in the top menu bar. Below the menu, there are two buttons: 'Install' and 'Update'. A search bar is located to the right of the buttons. The main area displays a table with columns for Name, Description, and Version. The table lists various R packages, many of which have checkboxes to their left, indicating they can be selected for installation. The 'System Library' section is visible on the left side of the table.

	Name	Description	Version
System Library			
<input checked="" type="checkbox"/>	base	The R Base Package	4.2.1
<input type="checkbox"/>	boot	Bootstrap Functions (Originally by Angelo Canty for S)	1.3-28
<input type="checkbox"/>	class	Functions for Classification	7.3-20
<input type="checkbox"/>	cluster	"Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al.	2.1.3
<input type="checkbox"/>	codetools	Code Analysis Tools for R	0.2-18
<input type="checkbox"/>	compiler	The R Compiler Package	4.2.1
<input checked="" type="checkbox"/>	datasets	The R Datasets Package	4.2.1
<input type="checkbox"/>	foreign	Read Data Stored by 'Minitab', 'S', 'SAS', 'SPSS', 'Stata', 'Systat', 'Weka', 'dBase', ...	0.8-82
<input checked="" type="checkbox"/>	graphics	The R Graphics Package	4.2.1
<input checked="" type="checkbox"/>	grDevices	The R Graphics Devices and Support for Colours and Fonts	4.2.1
<input type="checkbox"/>	grid	The Grid Graphics Package	4.2.1
<input type="checkbox"/>	KernSmooth	Functions for Kernel Smoothing Supporting Wand & Jones (1995)	2.23-20
<input type="checkbox"/>	lattice	Trellis Graphics for R	0.20-45
<input type="checkbox"/>	MASS	Support Functions and Datasets for Venables and Ripley's MASS	7.3-57
<input type="checkbox"/>	Matrix	Sparse and Dense Matrix Classes and Methods	1.4-1
<input checked="" type="checkbox"/>	methods	Formal Methods and Classes	4.2.1
<input type="checkbox"/>	mgcv	Mixed GAM Computation Vehicle with Automatic Smoothness Estimation	1.8-40
<input type="checkbox"/>	nlme	Linear and Nonlinear Mixed Effects Models	3.1-157
<input type="checkbox"/>	nnet	Feed-Forward Neural Networks and Multinomial	7.3-17

Para carregar um pacote basta selecionar a caixa ao lado esquerdo do nome do pacote. A instalação de um novo pacote é feita pela seleção do botão “*Install*”. Ao clicar neste botão, o **RStudio** irá abrir uma pequena caixa. Caso queira instalar a partir de um repositório *online*, na primeira caixa *Install from* mantenha a seleção *Repository (CRAN)* e na caixa abaixo digite o nome do pacote a ser instalado, conforme exemplificado na Figura 13.

Figura 13: Tela de instalação de um novo pacote.



A forma mais comum de instalação de um pacote é pela linha de comando, utilizando a função `install.packages()`. Para isto, você deve utilizar como argumento o nome do pacote entre aspas. Veja o *script* abaixo, com o comando utilizado para instalar o pacote `epitools` para análises epidemiológicas:

```
#Instalando o pacote epitools para análises epidemiológicas
install.packages("epitools")
```

Após instalar um pacote, para utilizá-lo em uma análise você deve carregá-lo em sua sessão atual. Para isso, digite em seu `RStudio` a função `library()`, conforme abaixo:

```
#Carregando o pacote epitools instalado anteriormente
library("epitools")
```

ATENÇÃO



Ao fechar o `RStudio`, ele iniciará uma nova sessão em `R` e os pacotes carregados anteriormente não estarão mais carregados. Logo, deve-se sempre carregar os pacotes necessários para o seu trabalho, logo no início do seu `script!!!` Os comandos de acionamento de pacotes devem estar presentes nos seus `scripts`.

5.3 Consultando pacotes carregados e sua versão

Para verificar quais pacotes estão carregados em sua área de trabalho, é possível digitar o comando `sessionInfo()` no painel Console.

O comando `sessionInfo()` é utilizado para que seja possível visualizar no painel Console, uma série de informações sobre a sua sessão atual:

- a versão de `R` utilizada (`R version`);
- a Plataforma (`Platform`);
- o seu sistema operacional (`Running under`);
- os pacotes padrão carregados ao abrir o `R` (após a linha `attached base packages:`);
- os pacotes carregados exclusivamente por outro pacote após a linha `loaded via a namespace (and not attached)`.

Abaixo, vamos mostrar (1) as linhas de comando (códigos) necessárias digitadas diretamente no Console, e (2) seus resultados (*outputs*) após a execução do comando. Esse recurso será muito utilizado neste curso. Veja abaixo:

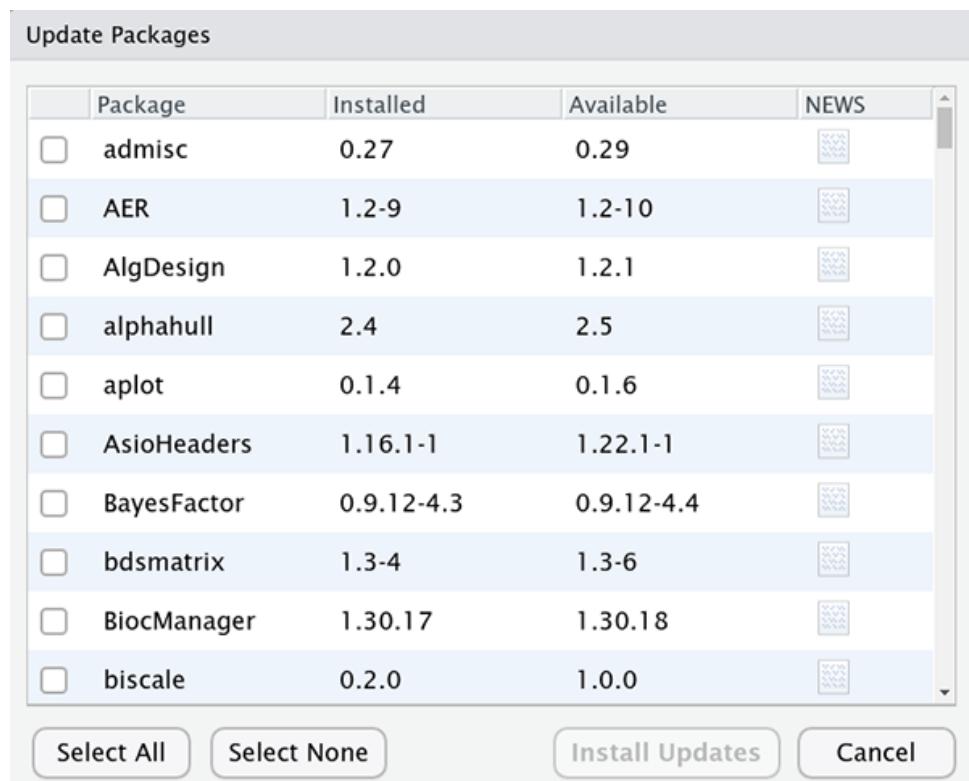
```
sessionInfo()
```

```
#> R version 4.2.1 (2022-06-23 ucrt)
#> Platform: x86_64-w64-mingw32/x64 (64-bit)
#> Running under: Windows 10 x64 (build 19044)
#>
#> Matrix products: default
#>
#> locale:
#> [1] LC_COLLATE=Portuguese_Brazil.utf8  LC_CTYPE=Portuguese_Brazil.utf8
#> [3] LC_MONETARY=Portuguese_Brazil.utf8 LC_NUMERIC=C
#> [5] LC_TIME=Portuguese_Brazil.utf8
#>
#> attached base packages:
#> [1] stats      graphics   grDevices  utils      datasets   methods    base
#>
#> loaded via a namespace (and not attached):
#> [1] digest_0.6.29   R6_2.5.1        jsonlite_1.8.0  magrittr_2.0.3
#> [5] evaluate_0.15   stringi_1.7.8   cachem_1.0.6   rlang_1.0.4
#> [9] cli_3.3.0       rstudioapi_0.13  jquerylib_0.1.4 bslib_0.4.0
#> [13] rmarkdown_2.14   tools_4.2.1     stringr_1.4.0   xfun_0.31
#> [17] yaml_2.3.5     fastmap_1.1.0   compiler_4.2.1  htmltools_0.5.3
#> [21] knitr_1.39     sass_0.4.2
```

5.4 Atualização dos pacotes

Algumas vezes o R pode emitir avisos *warning* sinalizando que seu pacote está desatualizado para uso. Quando isso ocorrer você pode atualizar as versões de pacotes clicando no botão *Update* na aba *Packages* (Figura 14), apresentada anteriormente. Uma janela similar à imagem abaixo será aberta.

Figura 14: Tela de atualização da versão de um pacote (*Update Packages*).



Ainda nesta tela clique no botão *Select All* (selecionar todos) e, em seguida, clique em *Install Updates* (em português Instalar Atualizações) para instalar todas as atualizações.

6. Boas práticas de organização

Agora que você já baixou o programa no seu computador e aprendeu que os *scripts* funcionam como roteiros de comandos que executarão rotinas de trabalho, vamos seguir pensando que você tem uma sequência de comandos para análise dos dados da vigilância. Assim, não há a necessidade de gastar mais tempo refazendo uma atividade sempre que precisar repeti-la.

Contudo, para que o R execute seus códigos sem erro, ou que você consiga compreender as etapas de trabalho que estão sendo executadas e ganhar tempo na rotina de análise de dados da vigilância em saúde, a organização no desenvolvimento desses *scripts* é fundamental.

Nos próximos itens descreveremos dicas valiosas para um iniciante em análise de dados para vigilância em saúde utilizando a linguagem de programação R !!!

6.1 Orientações gerais

Agora vamos passar algumas orientações mais gerais de como podemos organizar o nosso trabalho quando utilizamos uma linguagem de programação no dia a dia. Falaremos aqui sobre como organizar, formatar e salvar seu *script* de maneira a tornar o trabalho mais intuitivo e simples.

Vamos lá aprender como podemos fazer:

a) Organização do script

Quase sempre precisamos que outros profissionais da vigilância executem as rotinas de trabalho, seja por trabalhar com parcerias, por necessidade de somar esforços para agir contra um agravo, ou porque você necessitou tirar férias ou se ausentar por qualquer motivo. Nestes casos pode ser necessário compartilhar seu código de análise com outros colegas e ele deverá estar legível e de fácil entendimento.

Para estas situações, ou várias outras que podem surgir, seguem algumas sugestões importantes:

Utilizar sempre nomes simples, sem acentos, sem espaços entre as palavras e escritos de forma correta.

- Manter sempre as ações a serem executadas listadas no *script* por ordem de execução.
- Enumerar os passos do seu *script*.
- Eliminar do *script* comandos errados ou que não funcionam.
- Deixar espaços entre os comandos. Pular linha sempre ajuda!
- Comentar o *script*: O uso do símbolo *hashtag* (#) gera uma linha que o R entende como um texto simples, não um comando. Utilize para enumerar os passos, explicar um comando, colocar título ou observações. Ao comentar seu *script*, ele fica mais compreensível e mais fácil de ser seguido como um roteiro.

Nas Figura abaixo, exemplificamos as sugestões acima. Compare as duas situações apresentadas. Na Figura 15 temos um *script* **sem** organização adequada e, na Figura 16, temos um *script* seguindo boas práticas.



Figura 15: Método inadequado para a construção de scripts do R.



```
Untitled1* 
1
✖ 2 Tirar a média de uma coluna
3 mean(bd$variavel)
4 bd<-read.csv('basexxx.csv')
5 Hist(bdx)
✖ 6 Tirar a média de uma coluna2
7 mean(bd$variavel2)
8 bd<-read.csv('basexxx.csv')
9 Hist(bdx)
✖ 10 fazer o gráfico para o coordenador|
```

Figura 16: Método adequado para a construção de scripts do R.



```
script_gerar_grafico_e_media.R 
1 # Script para gerar grafico e media
2 # Observacao: Enviar grafico para o Coordenador
3
4 # 1 - Importar a base
5 base_SINASC <- read.csv(file = 'base_dados_SINASC_140620222.csv')
6
7 # 2 - Verificar media
8 mean(base_SINASC$IDADEMAE)
9
10 # 3 - Gerar o grafico histograma da Idade
11 hist(base_SINASC$IDADEMAE)
```

b) Formatação do script

A **indentação** é a organização ou formatação do seu código adicionando recuos (sejam com espaços ou tabulações) no código. Trata-se de uma rotina muito usada em várias linguagens de programação. A prática de indentar o código - utilizar espaços em branco para organização dos comandos - gera uma estrutura mais agradável para a leitura, com uma lógica mais simples de entendimento.

Observe na Figura 17 um exemplo de código escrito com organização, fácil de ser lido e reproduzido:

Figura 17: Método adequado de uso de scripts do R.



```
script_calcular_indicador.R
1 # Script para calcular indicador de risco
2
3 # 1 - Numero de casos
4 casos <- 150
5
6 # 2 - Populacao
7 populacao <- 75852
8
9 # 3 - Metodo de calculo (por 100 mil habitantes)
10 taxa <- casos / populacao * 100000
11
12 # 4 - Verificar o risco
13 ifelse(taxa > 300,
        "Em periodo epidêmico",
        "Fora do periodo epidêmico")
```

Observe na Figura 18 um exemplo de código escrito sem a preocupação com os códigos e o que eles fazem. Perceba apenas a organização e os espaços entre os comandos, entre as palavras e entre os símbolos (=, +, -, <-):

Figura 18: Método inadequado para uso de scripts do R.



```
Untitled1*  
Run Source  
1 casos <- 150  
2 populacao <- 75852  
3 taxa <- casos/populacao * 100000  
4 ifelse(taxa > 300, "Em período epidêmico", "Fora do período epidêmico")
```



Para indentar um código, você pode utilizar um atalho no RStudio : selecione as teclas **CTRL**, **SHIFT** e **A** ao mesmo tempo.

c) Salvar o trabalho

O `Rstudio` salva de forma automática todas as ações do código executadas, ou seja, ele salva por padrão todos os comandos rodados **enquanto ele está aberto**. No primeiro momento esse recurso pode parecer suficiente para garantir o armazenamento do seu trabalho, porém, é um grande concentrador de problemas.

Ao salvar de modo automático a sua área de trabalho, o `RStudio` faz com que você perca a noção do conteúdo. Ou seja, se ontem você estava empenhado em uma análise sobre incidência de dengue no município, ao reabrir o programa hoje para analisar os dados de Covid-19 terá que limpar a área de trabalho, pois ele estará com vários objetos referentes à dengue.

Outra desvantagem de utilizar esta ferramenta é a dependência dela, pois sempre precisará resgatar seu passado sem se preocupar em salvar da maneira correta os arquivos de sua análise. Dessa forma, ao menor sinal de erro no programa, toda a sua análise pode simplesmente ser apagada.

Um outro ponto muito importante para quem utiliza computadores que não são super-computadores é que, como a ferramenta automática é um concentrador de comandos e arquivos, muitos gigabytes são necessários para manter isso armazenado. Isso tornará seu computador mais lento sempre que reabrir o programa.

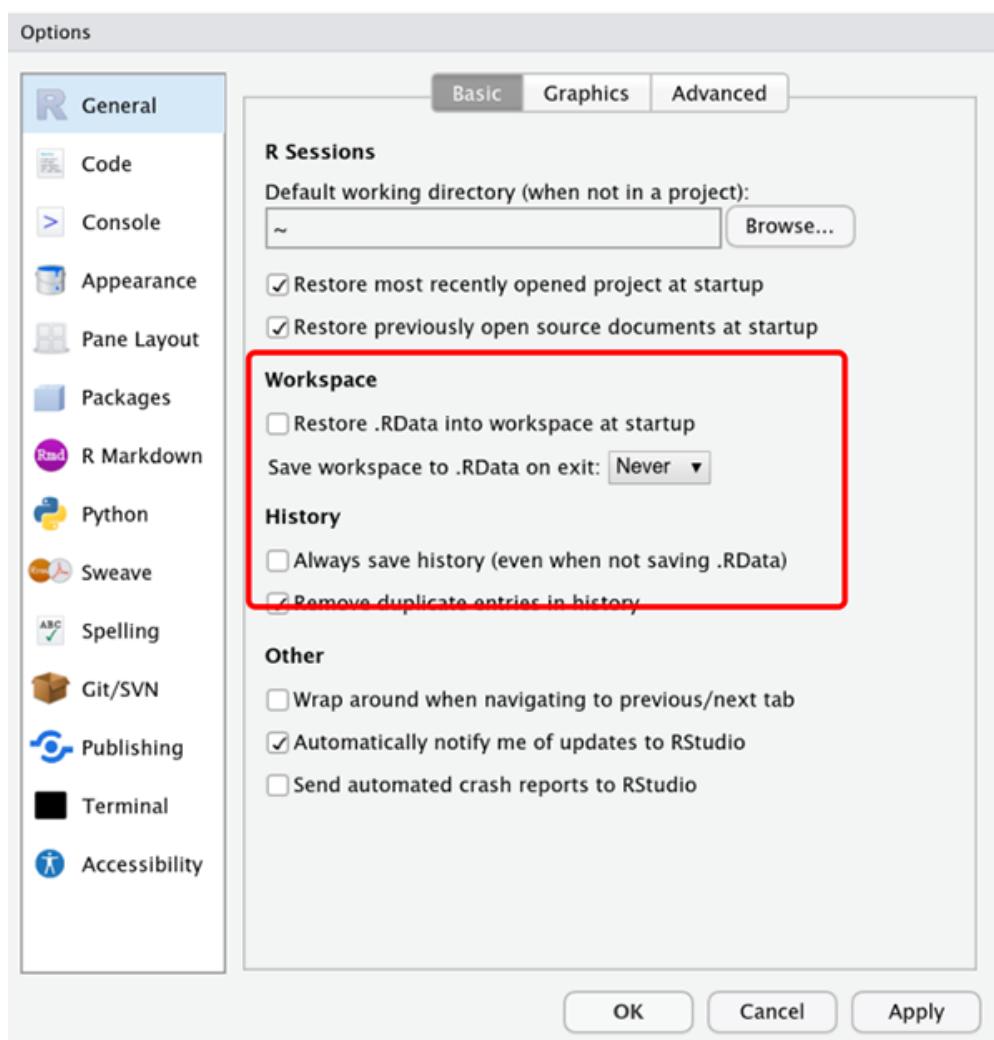
De fato, quando você usa um fluxo de trabalho como será sugerido no próximo item, não há necessidade de acúmulo infundável de histórico de comandos. Essa prática é desaconselhada pela maioria dos analistas e epidemiologistas que utilizam o `RStudio`.

Para desativar esta funcionalidade você deve seguir os seguintes itens no menu principal:

- Tools .
- Global options .
- Aba General .

Uma janela similar à Figura 19 será visualizada:

Figura 19: Tela para desativação do histórico de comandos no RStudio.



Ao abrir a aba **General**, **desmarque** os seguintes itens, para desativação do histórico de comandos no **RStudio**:

- 'Restore .RData into workspace at startup';
- 'Always save history (even when not saving .RData) '.

Além disso, você deve escolher **Never** no item 'Save workspace to .RData on exit'.

6.2 Diretório de trabalho

Um diretório de trabalho é uma pasta onde o R vai registrar a localização padrão para seus arquivos. É nele que devem ficar as bases de dados, os scripts e os relatórios que são necessários numa análise. Esse também será o local onde serão salvos todos os materiais que forem criados durante sua análise.

Não há regras para definição de um diretório de trabalho padrão, podendo ser uma pasta existente ou uma nova criada por você especialmente para receber os arquivos de uma análise. É uma boa prática ter um diretório diferente para cada análise que for realizada, seguindo as dicas citadas anteriormente.

Para verificar qual o diretório de trabalho que o R definiu como padrão inicial, basta digitar o comando `getwd()` no console e pressionar Enter. Ao executar esse comando, o R mostrará no console o caminho da pasta onde estão seus arquivos. Observe como funciona este comando, conforme exemplo da Figura 20.

Figura 20: Tela do painel Console com comando `getwd()` no RStudio.



The screenshot shows the RStudio interface with the 'Console' tab selected. The R version information and license details are displayed at the top. In the console window, the command `> getwd()` is entered, followed by its output: `[1] "C:/Users/PC/Documentos/ANALISE_DADOS_VIG_R"`.

```
R version 4.2.1 (2022-06-23) -- "Funny-Looking Kid"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)

R é um software livre e vem sem GARANTIA ALGUMA.
Você pode redistribui-lo sob certas circunstâncias.
Digite 'license()' ou 'licence()' para detalhes de distribuição.

R é um projeto colaborativo com muitos contribuidores.
Digite 'contributors()' para obter mais informações e
'citation()' para saber como citar o R ou pacotes do R em publicações.

Digite 'demo()' para demonstrações, 'help()' para o sistema on-line de ajuda,
ou 'help.start()' para abrir o sistema de ajuda em HTML no seu navegador.
Digite 'q()' para sair do R.

> getwd()
[1] "C:/Users/PC/Documentos/ANALISE_DADOS_VIG_R"
>
```

Além da escrita legível e objetiva dos *scripts*, também devemos organizar os arquivos em diretórios ou pastas no computador. É muito comum esquecer de definir o diretório e, durante sua análise, encontrar erros por indicar caminhos errados ao R.



ATENÇÃO

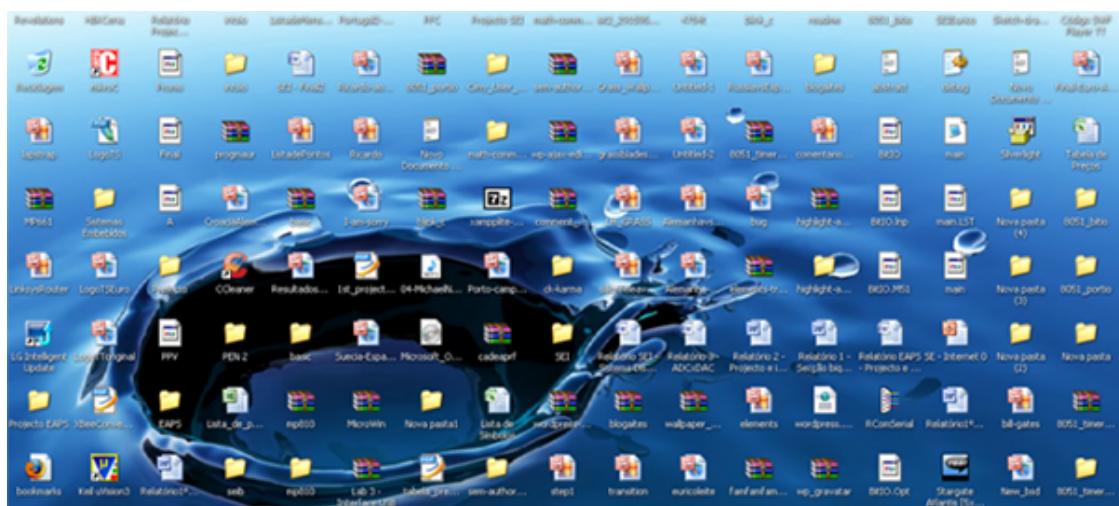
Lembre-se que o R segue as instruções que você escreveu. Assim, se você errar ele errará também!!

Esta organização é de grande importância, pois há várias situações em que há a necessidade de localizar um arquivo rapidamente:

- O secretário de saúde liga para o setor de vigilância epidemiológica dizendo que precisa do relatório imediatamente.
- A imprensa está aguardando um gráfico sobre a situação de uma doença.
- Um colega precisa reproduzir o relatório atualizado.
- Ao final do mês vários relatórios devem ser produzidos para a prestação de contas da equipe de gestão.

Agora observe a Figura 21 abaixo e imagine você encontrando um arquivo nesta tela de computador:

Figura 21: Imagem de uma área de trabalho.



Habituarse a organizar pastas de trabajo con **nomes más intuitivos** e **nomes de archivos que facilitem la identificación del contenido** son importantes acciones que deben seguirse por usuarios iniciantes y experimentados de cualquier lenguaje de programación para análisis de datos.

Confira na Figura 22 exemplos de estrutura que você não deve seguir e na Figura 23 sugestões de diretórios organizados.

Figura 22: Imagem de uma área de trabalho inadequada.

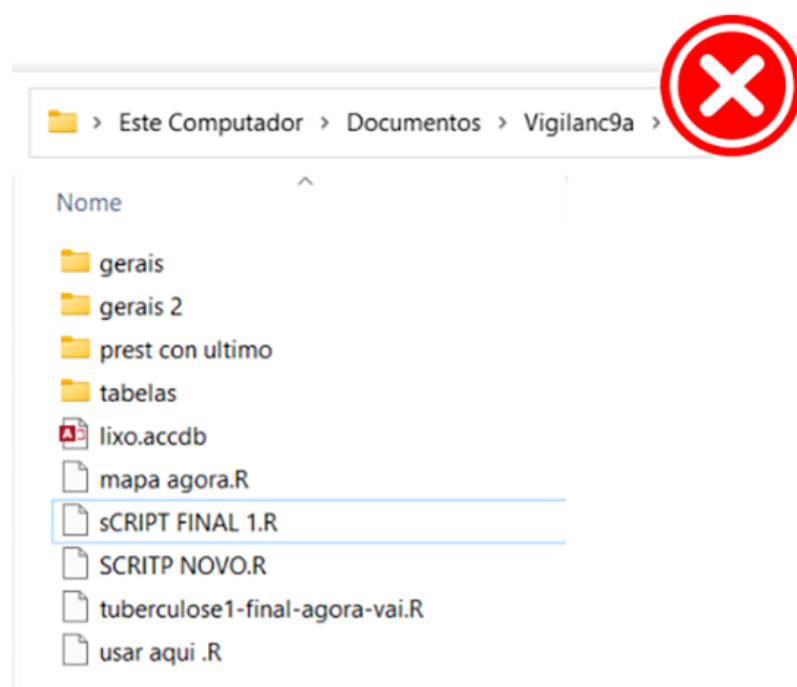
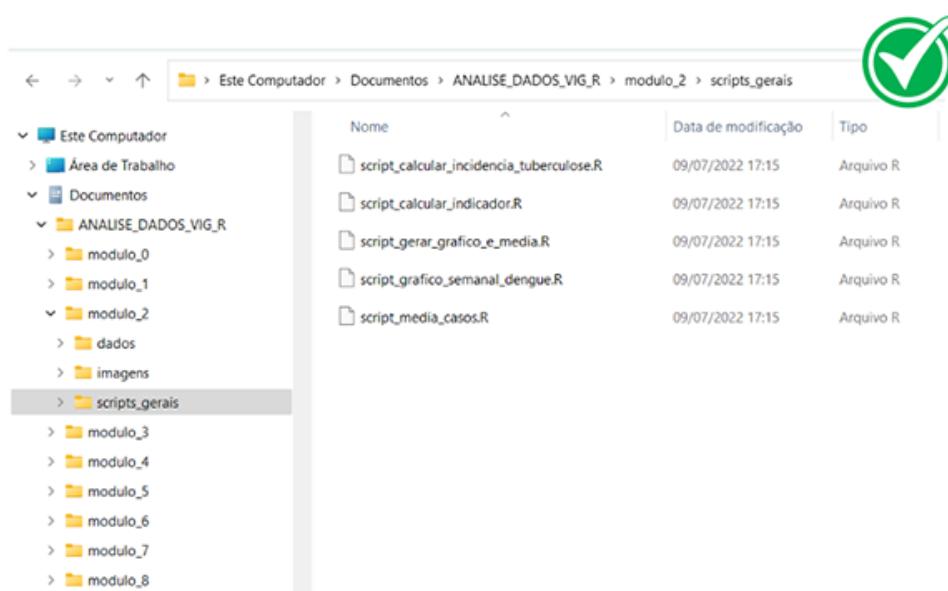


Figura 23: Imagem de uma área de trabalho adequada.



6.3 Utilização da ferramenta “Projetos”

O **RStudio** possui uma funcionalidade que torna a etapa de organização do diretório de trabalho e gerenciamento de arquivos ou *scripts* muito mais fácil: a ferramenta de **Projeto**, arquivo de extensão do tipo **.Rproj**.

Esta ferramenta permite que o diretório seja definido pela mesma localização do projeto. Desse forma, sempre que precisar trocar de computador ou compartilhar uma análise, o **R** saberá de forma automática qual é a localização dos seus arquivos do projeto. Assim, não será necessário digitar longos caminhos e nem se preocupar com uma possível mudança de pasta. Tudo estará organizado em uma mesma pasta!

Utilizar **Projeto** (**.Rproj**) garantirá:

- organização do seus diretórios;
- facilidade para importar e exportar bases de dados no **R**;
- facilidade para análises de dados compartilhadas com outras pessoas no dia a dia.



É uma boa prática utilizar a ferramenta de **Projetos** (**.Rproj**) sempre que realizar análises. Crie subpastas separadas, dentro do diretório do projeto, para base de dados, para figuras, para *scripts*, para relatórios, etc. Esta organização será valiosa e aumentará a sua produtividade.

a) Como criar projetos (.Rproj)

Para criar você necessitará:

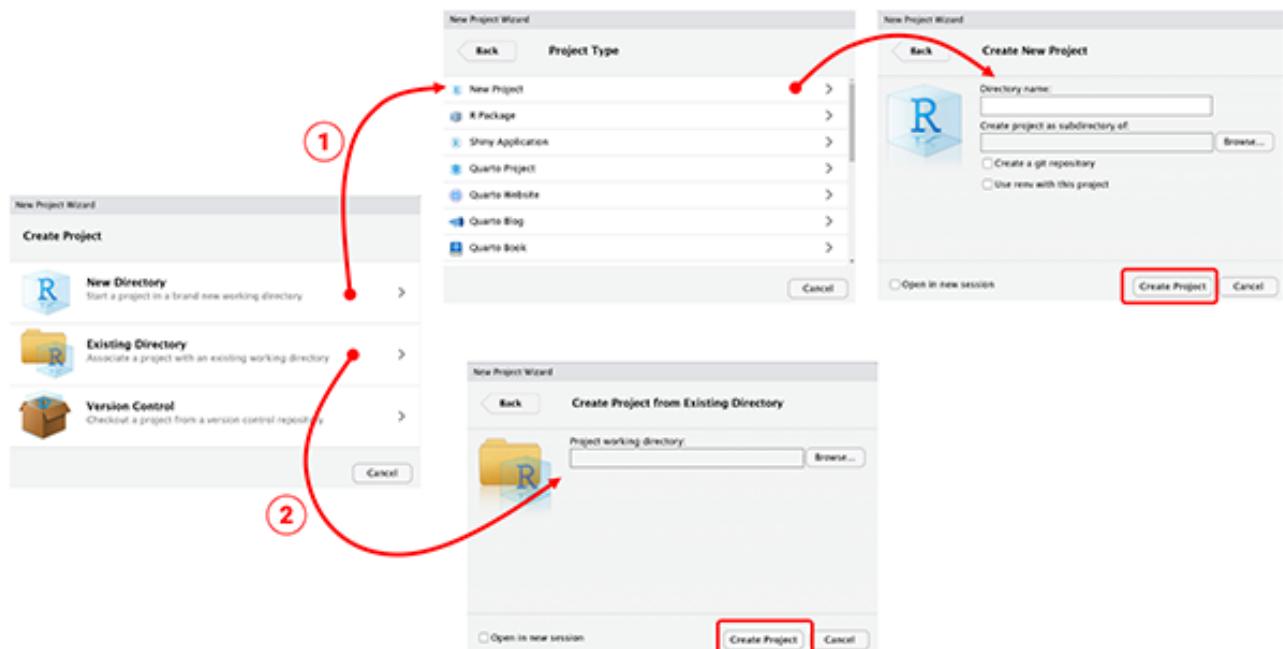
- Ir até o menu e clicar em *File*;
- Clicar em *New Project*.

Uma janela será retornada para escolha de criação de um projeto em:

1. uma nova pasta *New Directory*, seguindo para *New Project* e definição do nome **ou**;
2. a partir de uma pasta já existente *Existing Directory*, seguindo para a seleção da pasta.

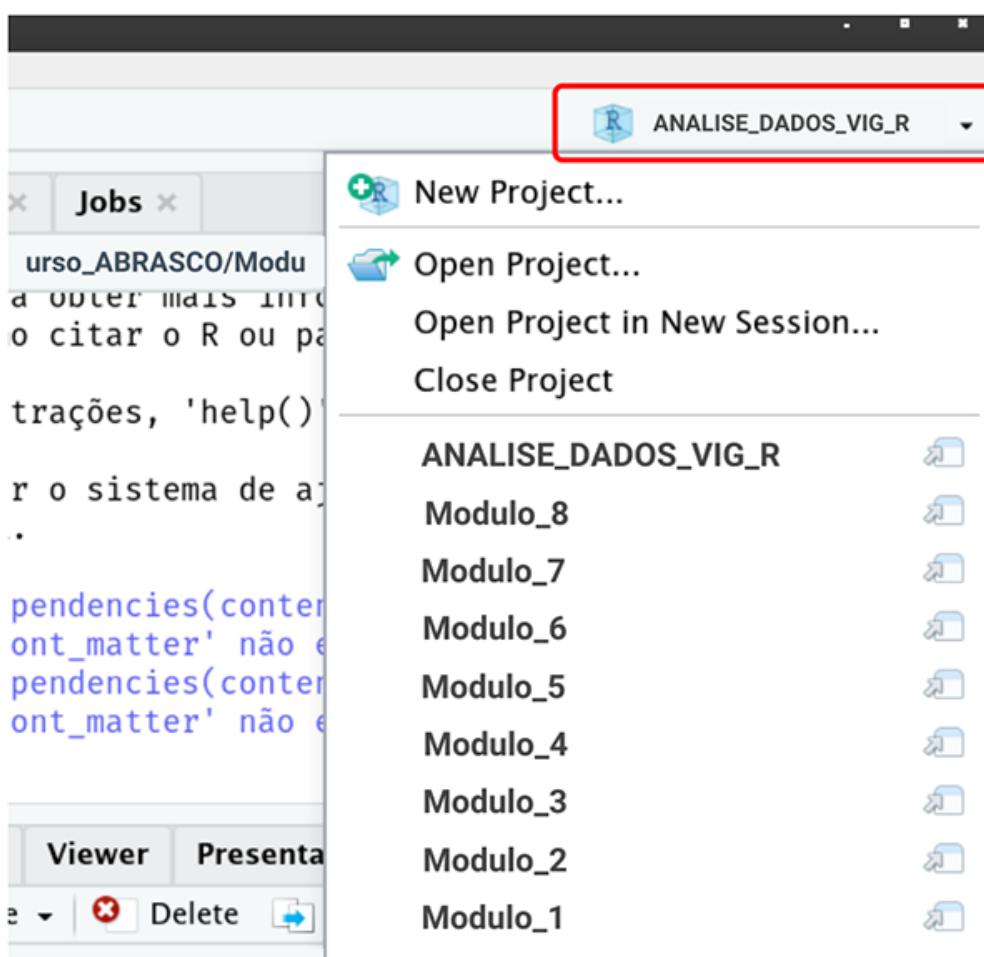
A Figura 24 abaixo mostra a sequência de telas para criação por ambas as opções:

Figura 24: Criando uma nova pasta de armazenamento (*NewDirectory*) e um novo projeto (*New Project*) com RStudio.



Após criar o projeto, um arquivo de extensão **.Rproj** será criado dentro da pasta. Este será o atalho para abrir o projeto no **RStudio**. Ao clicar no **Projeto** (**.Rproj**) todas as pastas do diretório do projeto serão carregadas automaticamente para **RStudio** e o nome do projeto será exibido na barra de ferramentas Projetos (localizada no canto direito da barra de ferramentas principal), indicando que o nome do projeto está ativo. Veja na Figura 25.

Figura 25: Tela de visualização dos projetos (.Rproj).



Você também pode utilizar o pacote `usethis` para criação de projetos diretamente na linha de comando! Basta digitar o comando abaixo no console, definindo qual será a pasta para o projeto. Caso a pasta já exista, será criada toda a estrutura para o projeto. Caso contrário, a pasta é criada e segue a mesma função.

Observe os comandos abaixo para instalar e carregar o pacote `usethis`:

```
#instalando pacote usethis para criar um projeto na linha de comando
install.packages("usethis")

# carregando o pacote usethis para utilizá-lo
library("usethis")
```

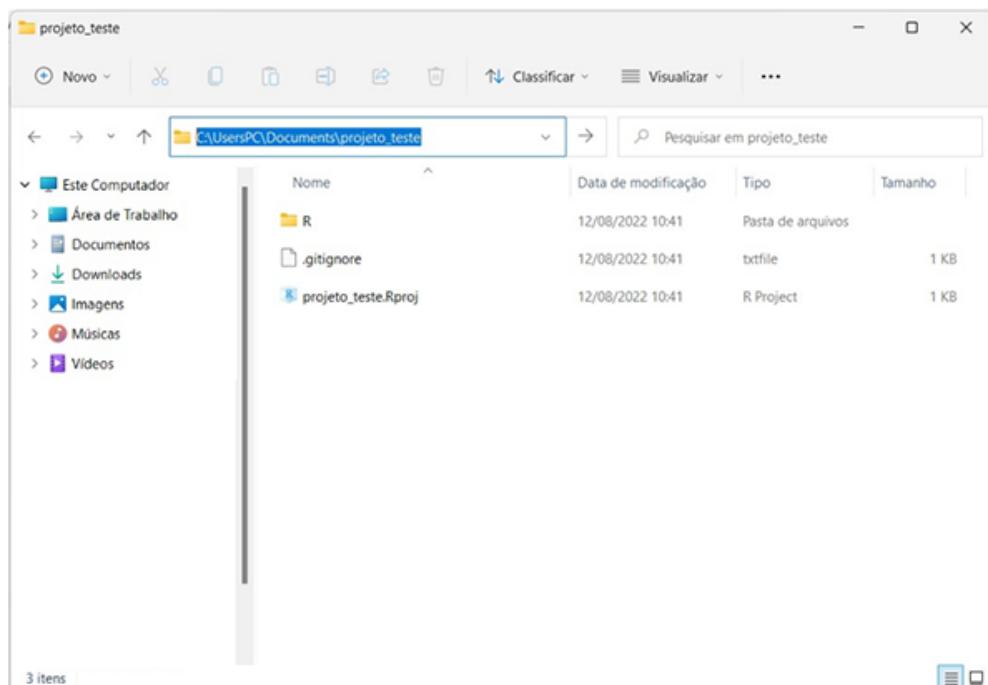
A função `create_project()` presente no pacote `usethis`, será necessária para indicar o local no seu computador em que o arquivo do tipo (`.Rproj`) deverá ser criado. Este caminho é importante para apontar onde estarão os arquivos do seu Projeto em seu computador. Ao criar um projeto (`.Rproj`), clicando sob ele você poderá reabrir o `RStudio` já com todo os seus arquivos necessários para suas análises.

```
#instalando pacote usethis para criar um projeto na linha de comando
create_project(path = "C:/nome_do_caminho_da_pasta")
```

Veja o exemplo abaixo, nele estamos criando um projeto na pasta “`C:/Users/PC/Desktop/Documents/projeto_teste`”, replique-o e observe o *output* no painel Console do seu RStudio:

```
library(usethis)
create_project(path = "C:/Users/PC/Desktop/Documents/projeto_teste/")
✓ Setting active project to 'C:/Users/PC/Desktop/Documents/projeto_teste'
✓ Creating 'R/'
✓ Writing 'projeto_teste.Rproj'
✓ Adding '.Rproj.user' to '.gitignore'
✓ Opening 'C:/Users/PC/Desktop/Documents/projeto_teste' in new RStudio session
✓ Setting active project to ''
```

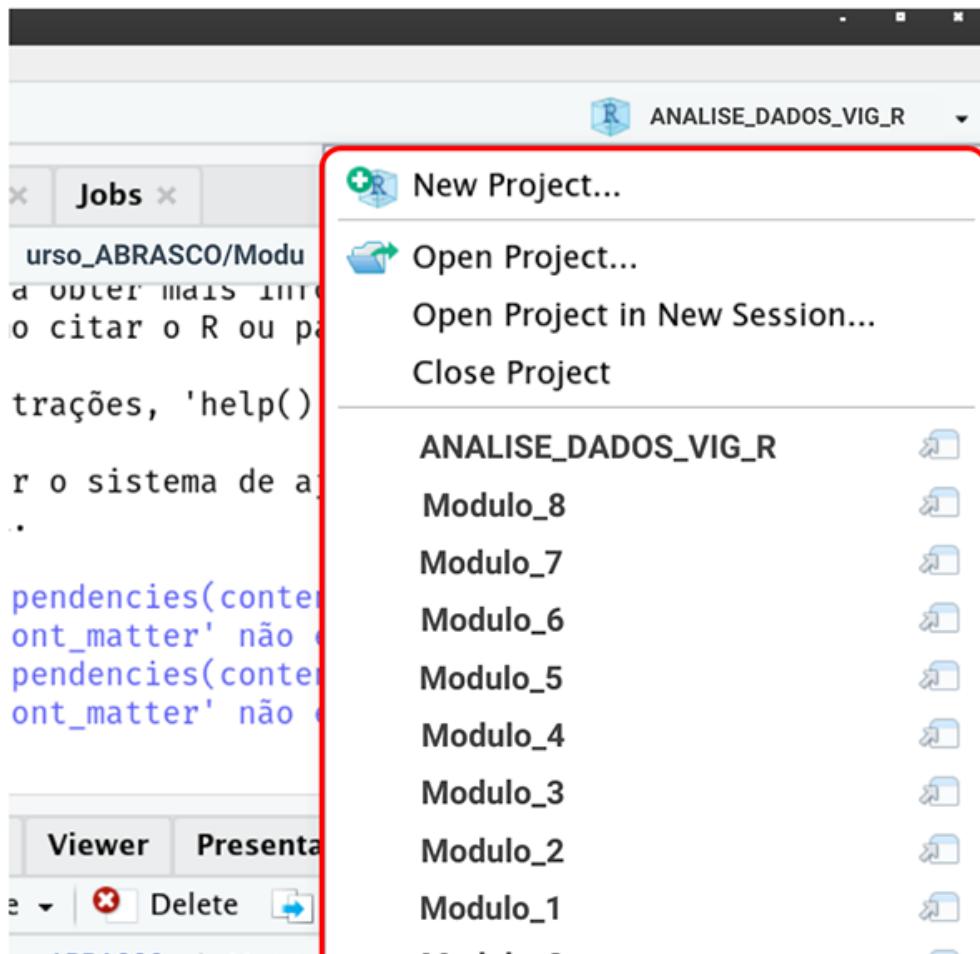
Figura 26: Tela com projeto criado viausethis.



b) Selecionando Projetos (.Rproj)

É possível abrir um **Projeto** (.Rproj) já existente. Outra maneira de selecionar **Projeto** (.Rproj) é utilizar a opção Projetos localizada no canto direito da barra de ferramentas principal. Perceba que na Figura 27 é apresentada uma lista com os projetos recentes utilizados. Basta clicar sob o projeto que deseja tornar ativo para que ele seja aberto. Lembre sempre de salvar seu projeto. Mas, caso esqueça, o **RStudio** irá confirmar o fechamento do projeto somente após você salvar os arquivos.

Figura 27: Tela para seleção do projeto (.Rproj).



Agora que você chegou até aqui já é capaz de baixar o programa que fará a análise de dados da vigilância no seu computador. Vamos em frente!



Próximo módulo

Viu? Até que foi tranquilo. A partir de agora você já está pronto para avançar na escrita de scripts, iniciando os primeiros passos na programação com R e aplicando em análise de dados na vigilância em saúde.

Acesse os demais módulos deste curso!

