



**Cursos Integrados  
em Vigilância em Saúde**

*Curso* —

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

# **Módulo 2 - Introdução à análise de dados com R - Parte II**

## **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 2 - Introdução à análise de dados com R - Parte II

*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 II / Oswaldo Gonçalves Cruz, Marcelo Eduardo Borges, Kamila de Oliveira Belo, Denis de Oliveira Rodrigues. – Santa Catarina ; São Paulo ; Rio de Janeiro : UFSC ; ITPS ; Abrasco; 2022. 43p. (Análise de dados para a vigilância em saúde – Curso Básico; Módulo 2).

Publicação Online  
[10.52582/curso-analise-dados-vigilancia-modulo2](https://repositorio.ufsc.br/handle/10.52582/curso-analise-dados-vigilancia-modulo2)

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

# Sumário

Conceitos básicos para análise de dados na vigilância em saúde .....	06
<b>1. Analisando seus dados com o R .....</b>	<b>08</b>
1.1 Bancos de dados ou Dataframes.....	11
<b>2. Importando os dados para sua análise .....</b>	<b>13</b>
2.1 Importando arquivos no formato CSV .....	16
2.2 Importando arquivos no formato DBFs .....	22
2.3 Importando arquivos do Microsoft Excel .....	25
2.4 Visualizando os arquivos importados para o R .....	28
<b>3. Tipo de variáveis.....</b>	<b>30</b>
3.1 Variáveis quantitativas.....	32
3.2 Variáveis qualitativas .....	34
<b>4. Como obter ajuda para uso do R? .....</b>	<b>40</b>

## Conceitos básicos para análise de dados na vigilância em saúde

Cada vez mais na realidade dos serviços de vigilância em saúde nos deparamos com o aumento do volume de dados e sua crescente complexidade. Hoje, existe um conjunto de *softwares* disponíveis, que são muito utilizados e apoiam nossas análises, como o Stata, o SPSS e o Epi Info.

Mas, os *softwares* tradicionais de processamento de dados simplesmente não conseguem gerenciar de forma adequada alguns bancos de dados ou mesmo integrá-los, o que requer ferramentas específicas de análise. No seu dia a dia, provavelmente você já deve ter se deparado com momentos em que o Excel, por exemplo, interrompe a leitura de banco de dados por ausência de memória, ou até mesmo essas análises se tornarem extremamente lentas, penosas e repetitivas.

Porém, com o *software R* você será capaz de manipular dados complexos e produzir análises poderosas, já que é uma linguagem de programação voltada à manejo e análise estatística avançada de dados, que pode ser facilmente aplicada por meio de funções e pacotes algumas vezes criados pelo próprio usuário, a partir de necessidades locais, por exemplo.

Na atualidade há um contínuo aumento da coleta automática de grandes bancos de dados na saúde (o famoso *Big Data*) e a Vigilância em Saúde está inserida neste contexto. O profissional de vigilância necessita utilizar seus dados na saúde compreendendo o passo a passo da coleta, organização e interpretação dos dados obtidos. Extrair informações significativas desses bancos de dados requer um esforço e ferramentas adequadas, e saber lidar com os dados de forma estruturada, segura, com precisão e transparência.

Para seguir com este módulo do curso você deve já ter instalado o *software R* e a interface gráfica *RStudio*, que torna o uso e aprendizado do *R* ainda melhor, esses passos estão disponíveis no “Módulo de Introdução à análise de dados com *R* - Parte I”.

Neste módulo 2 serão apresentados alguns conceitos básicos para iniciar a manipulação de **dados secundários com R** utilizando-se dos bancos de dados de diversas fontes!

**Ao final deste módulo, você será capaz de:**

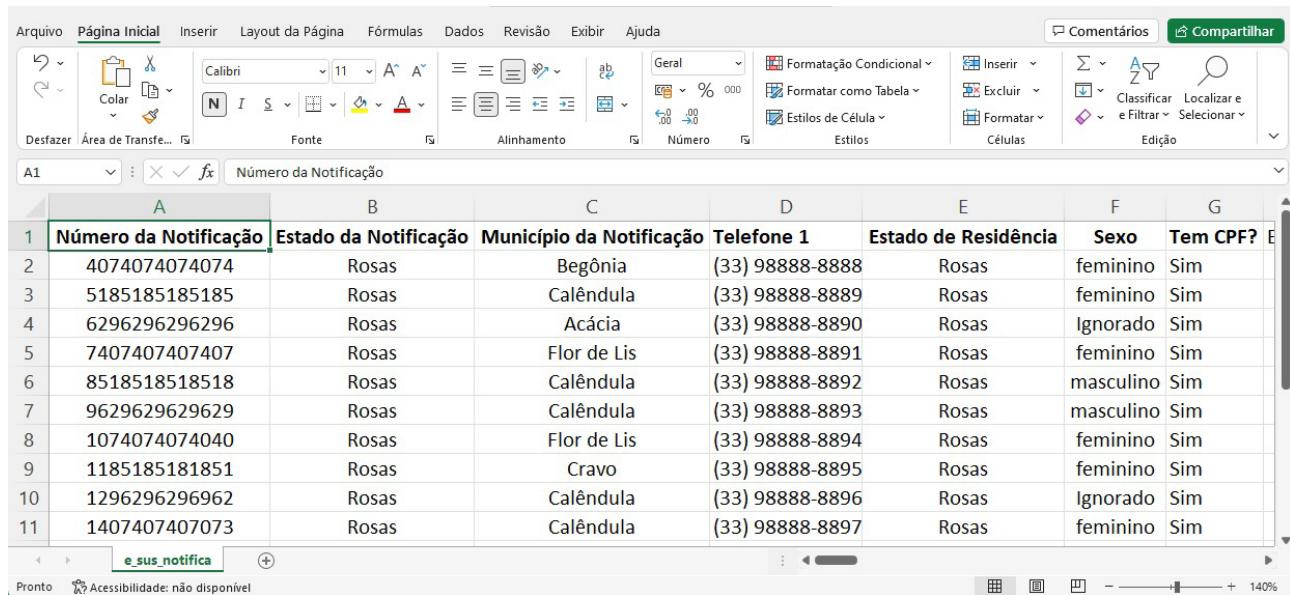
- 1.** conceituar a estrutura de um banco de dados;
- 2.** importar banco de dados para o *RStudio* e criar tabelas para análise de dados;
- 3.** reconhecer e manipular objetos no *R* para analisar dados;
- 4.** conhecer os tipos de variáveis presentes em um banco de dados;
- 5.** buscar ajuda quando necessário para o uso *R*.

## *1. Analisando seus dados com o R*

A análise de dados é uma das mais antigas práticas na vigilância em saúde. Conhecer e acompanhar o estado de saúde da população é uma das atividades mais importantes para elaborar um “diagnóstico de saúde”. Por meio da **análise de situação de saúde** de uma região é possível fazer o conhecimento de perfis, padrões e tendências que fomentem ações de proteção e promoção da saúde, bem como a prevenção e controle de doenças e agravos à saúde da população. A análise de dados envolve, portanto, um processo de descrição e comparação de dados de diferentes fontes, em momentos e locais distintos, buscando apresentar as características de tempo, pessoa e lugar.

Na rotina da vigilância em saúde os dados que subsidiam essas análises são exportados dos Sistemas de Informação em Saúde, como o Sistema de Informação de Mortalidade (SIM); o Sistema de Informação de Nascidos Vivos (SINASC), o Sistema de Informação de Agravos de Notificação (SINAN); o Sistema de Informação do Programa Nacional de Imunização (SIPNI); o Sistema de Informação e-SUS Notifica (e-SUS), dentre outros tantos. A partir desses sistemas, chegamos em tabelas visualizadas no formato da Figura 1.

**Figura 1: Tabela de visualização do e-SUS Notifica  
com seus dados organizados em Colunas e Linhas.**



The screenshot shows a Microsoft Excel spreadsheet titled "e\_sus\_notifica". The table has 11 rows and 8 columns. The columns are labeled: Número da Notificação, Estado da Notificação, Município da Notificação, Telefone 1, Estado de Residência, Sexo, and Tem CPF?. The first row contains the column headers. The data includes various entries such as Rosas, Begônia, (33) 98888-8888, etc. The Excel ribbon at the top shows standard tabs like Arquivo, Página Inicial, and Insertar.

1	Número da Notificação	Estado da Notificação	Município da Notificação	Telefone 1	Estado de Residência	Sexo	Tem CPF?	E
2	4074074074074	Rosas	Begônia	(33) 98888-8888	Rosas	feminino	Sim	
3	5185185185185	Rosas	Calêndula	(33) 98888-8889	Rosas	feminino	Sim	
4	6296296296296	Rosas	Acácia	(33) 98888-8890	Rosas	Ignorado	Sim	
5	7407407407407	Rosas	Flor de Lis	(33) 98888-8891	Rosas	feminino	Sim	
6	8518518518518	Rosas	Calêndula	(33) 98888-8892	Rosas	masculino	Sim	
7	9629629629629	Rosas	Calêndula	(33) 98888-8893	Rosas	masculino	Sim	
8	1074074074040	Rosas	Flor de Lis	(33) 98888-8894	Rosas	feminino	Sim	
9	1185185181851	Rosas	Cravo	(33) 98888-8895	Rosas	feminino	Sim	
10	1296296296962	Rosas	Calêndula	(33) 98888-8896	Rosas	Ignorado	Sim	
11	1407407407073	Rosas	Calêndula	(33) 98888-8897	Rosas	feminino	Sim	

Na Figura 1, vemos uma base fictícia do Sistema de Informação e-SUS Notifica, simulando dados de casos leves de Covid-19 (Notificações de Síndrome Gripal). Abrimos esta tabela utilizando o Microsoft Excel. Este formato de tabela é o que será utilizado neste módulo para introduzir conceitos e alguns elementos essenciais em uma análise de dados com a linguagem R.

Para iniciar, serão necessários a escrita de roteiros (*scripts*) e os comandos no R de maneira a construirmos uma rotina de trabalho automatizada com esses dados, ou seja, vamos construir linhas de códigos que indicarão as ações que precisam ser realizadas.

Estas linhas de código irão conter os seguintes elementos definidos na Figura 2 abaixo:

**Figura 2: Tabela com conceitos utilizados na análise de dados com R .**

Termos	Sinônimos	Descrição
Base de dados	banco de dados, <i>dataframes</i> , tabelas	Estrutura de armazenamento dos dados
Variável	Coluna, <i>column</i> , Cabeçalho	Pode ser entendida como uma caixinha, onde os dados (valores, registros ou observações) são armazenados e que é utilizada durante a execução da análise
Linha	<i>row</i> , Observação, Registro, Valor	São dados de diferentes tipos (números, datas, textos, etc)
Função	Verbo, Comando, Ação	São ordens dadas ao <i>software R</i> , para que ele execute uma ação necessária para sua análise
Argumento	Parâmetros	Criar uma personalização de características para executar um comando (cores, tamanhos, caracteres, etc)
Output	Resultado	São as saídas oriundas de uma ação solicitada, ou seja, tudo que queremos que nosso código no R retorne para nós, podendo ser no formato de textos, tabelas, gráficos, mapas, imagens, etc.

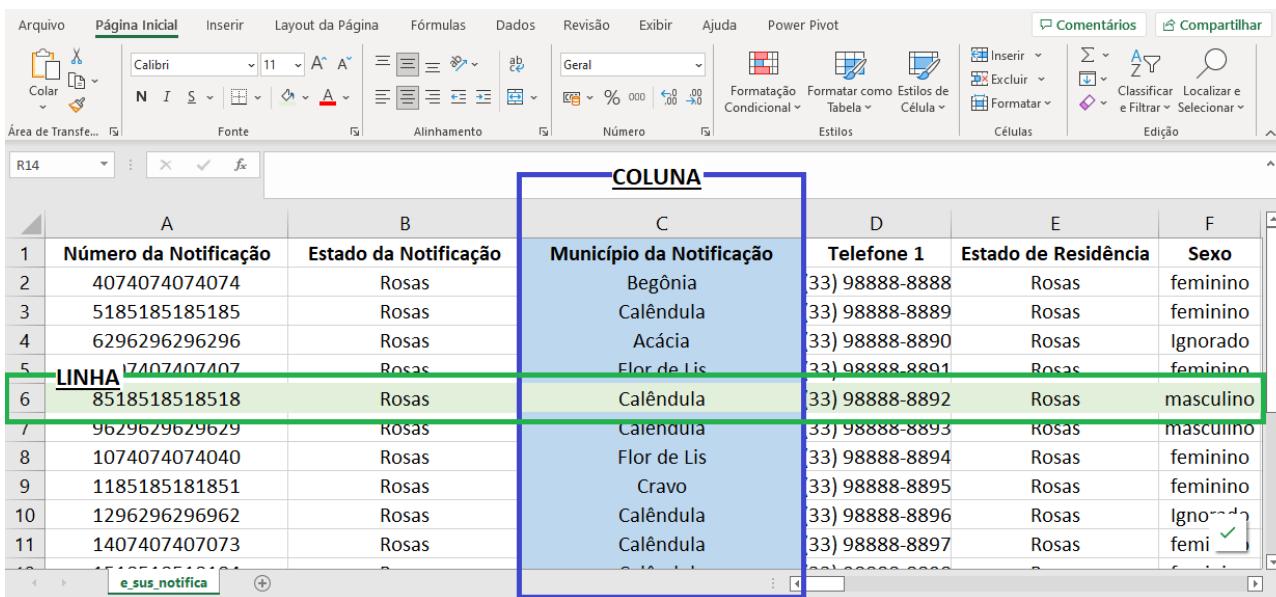
Agora, vamos ver como esses elementos podem ser organizados em um *script*.

## 1.1 Bancos de dados ou Dataframes

Neste curso, utilizaremos dados organizados em duas dimensões, linhas e colunas, formando os chamados bancos de dados ou *dataframes*. Em um `dataframe` cada coluna representa uma variável e cada linha uma observação. Este é um formato comum em quase todos os sistemas de informações de vigilância em saúde.

Vamos retornar à planilha do e-SUS Notifica apresentada, agora destacando um pouco mais a estrutura dessa tabela na Figura 3:

**Figura 3: Planilha aberto com dados e-SUS Notifica organizados em Colunas e Linhas.**



	A	B	C	D	E	F
1	Número da Notificação	Estado da Notificação	Município da Notificação	Telefone 1	Estado de Residência	Sexo
2	4074074074074	Rosas	Begônia	(33) 98888-8888	Rosas	feminino
3	5185185185185	Rosas	Calêndula	(33) 98888-8889	Rosas	feminino
4	6296296296296	Rosas	Acácia	(33) 98888-8890	Rosas	Ignorado
5	7407407407	Rosas	Flor de Lis	(33) 98888-8891	Rosas	feminino
6	LINHA 8518518518518	Rosas	Calêndula	(33) 98888-8892	Rosas	masculino
7	9629629629629	Rosas	Calêndula	(33) 98888-8893	Rosas	masculino
8	1074074074040	Rosas	Flor de Lis	(33) 98888-8894	Rosas	feminino
9	1185185181851	Rosas	Cravo	(33) 98888-8895	Rosas	feminino
10	1296296296962	Rosas	Calêndula	(33) 98888-8896	Rosas	Ignorado
11	1407407407073	Rosas	Calêndula	(33) 98888-8897	Rosas	femi
12	1518518518518518	Rosas	Calêndula	(33) 98888-8898	Rosas	ma

Perceba que foi possível visualizar na Figura 3, na cor azul a variável “Município da Notificação” disposta na coluna (*column*, em inglês) da tabela. As colunas têm algumas características importantes:

- são visualizadas verticalmente na tela;
- cada coluna é única;
- contêm dados do mesmo tipo (texto, número);
- são mencionados pelos nomes, que aparecem na parte superior como títulos, os cabeçalhos.

Já na cor verde, a Figura 3 destaca as Linhas (row, em inglês), também chamadas de observações, valores ou registros. Suas características são:

- são visualizadas horizontalmente na tela;
- podem conter um conjunto de tipos diferentes de dados (números, textos, telefones) para cada registro.

O primeiro passo para iniciar a sua análise de dados é a importação dos bancos de dados. Nas próximas subseções você terá o passo a passo de como importar dados oriundos dos sistemas de informação em saúde do tipo `.csv` e `.dbf`. Também vamos importar arquivos no formato do Microsoft Excel (`.xls` e `.xlsx`), os quais são frequentemente utilizados pelas equipes das vigilâncias.

## 2. Importando os dados para sua análise

No R qualquer estrutura é armazenada na forma de um **objeto**, seja um valor, um conjunto de valores ou até mesmo uma base de dados. Essa linguagem se estrutura, por tanto, no paradigma da **Programação Orientada a Objetos**, ou seja, **tudo será um objeto**.

Para criar e alterar objetos no R é necessário utilizar o símbolo: `<-`, chamado de **operador de atribuição**. Esse operador é formado pelos símbolos “menor que” (`<`) e hífen (`-`). Mas, atenção! O símbolo “igual” (`=`) também pode ser utilizado, mas não recomendamos pois possui muitas outras utilidades e você pode se confundir. Por isso, neste curso, **adotaremos somente o operador `<-` para criar e alterar objetos no R**.

Observe na Figura 4 a etapa de criação de um objeto. Para criá-lo basta definir o nome, inserir o operador de atribuição e, por último, o valor que será atribuído. Veja a Figura 4:

**Figura 4: Etapa de atribuição de dados a um objeto no R.**



nome do objeto

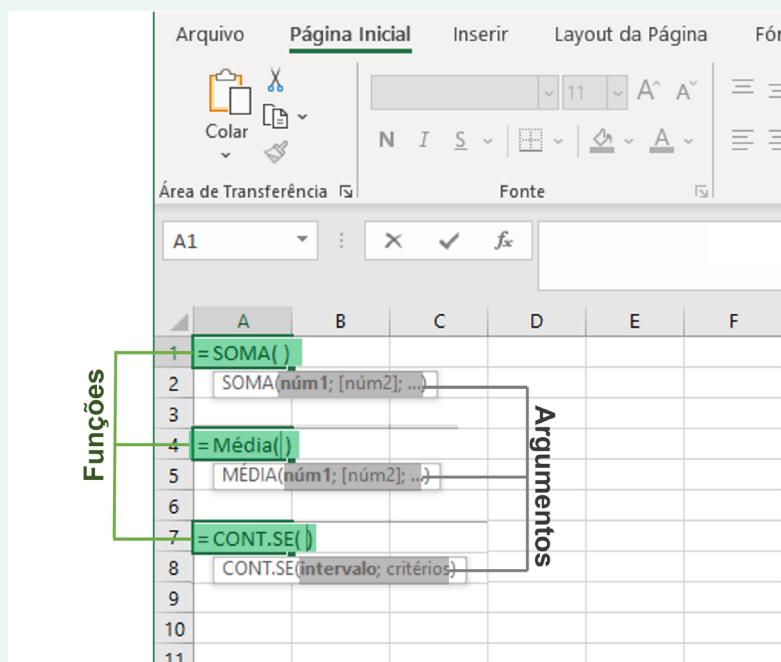
<-

conjunto de valores para  
serem salvos no objeto

A seguir iremos praticar esta etapa de criação de um objeto, aprendendo a importar bancos de dados para analisar dados do Estado de Rosas, um estado fictício criado para este curso. Para as importações ocorrerem, vamos precisar utilizar uma **função**.

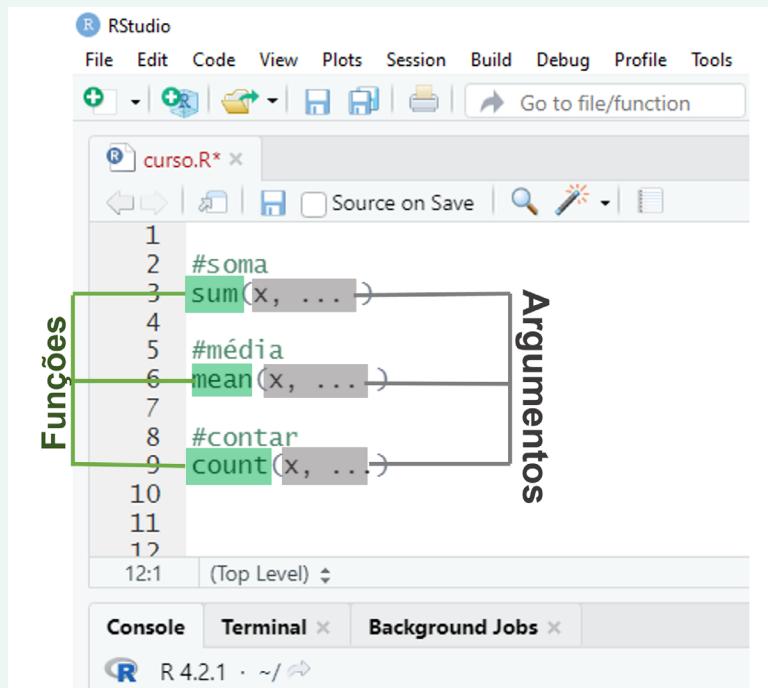
O conceito de **função** no R é semelhante às fórmulas que realizamos no Microsoft Excel. As funções são as ações que você quer que o R execute. Elas exigem, muitas vezes, valores de entrada (que denominamos *input*) e retornam valores de saída (também chamados de *output*). Sempre que escrever uma função, deverá inserir entre parênteses ( ) os chamados *argumentos*. Observe na figura 5 como seria fazer algumas das funções mais comuns no Excel, a função soma, média e cont.se:

**Figura 5: Utilizando uma função no Microsoft Excel.**



Veja que acima temos três exemplos: o cálculo de uma soma, uma média e uma contagem de valores válidos. Todos estão com a função() na cor verde e seus respectivos argumentos na cor cinza. No R escrevemos utilizando esta mesma lógica. Observe na Figura 6 como é escrever uma função com R:

**Figura 6: Utilizando uma função no R.**



The screenshot shows the RStudio interface with a script file named 'curso.R'. The code in the script is:

```

1
2 #soma
3 sum(x, ...)
4
5 #média
6 mean(x, ...)
7
8 #contar
9 count(x, ...)
10
11
12

```

Annotations with arrows point from the left margin to the function names 'sum', 'mean', and 'count', which are highlighted in green. These annotations are labeled 'Funções' (Functions). Another annotation points from the right margin to the argument 'x' in each function call, which is highlighted in grey. This annotation is labeled 'Argumentos' (Arguments).

Percebe como é semelhante? Quase sempre as funções são escritas na língua inglesa; e em geral as funções podem ter um ou mais argumentos, separados por vírgulas (,) ou definidos pelo operador igual (=). Eventualmente, porém, uma função pode não precisar de argumento.

Observe neste módulo como utilizamos as funções, aprendendo a importar banco de dados. Usaremos funções para muitas atividades nos próximos módulos.

## 2.1 Importando arquivos no formato CSV

Para aprender a importação de um arquivo do tipo `.csv` utilizaremos o banco de dados exportado do e-SUS Notifica do Estado de Rosas (Figura 1). Para isso, utilizaremos uma **função** do pacote `readr`. Falaremos mais vezes dele em todo curso. Por enquanto, vamos instalá-lo para, em seguida, importar o arquivo `{e_sus_notifica.csv}`, disponível no menu lateral “Arquivos”, do módulo.



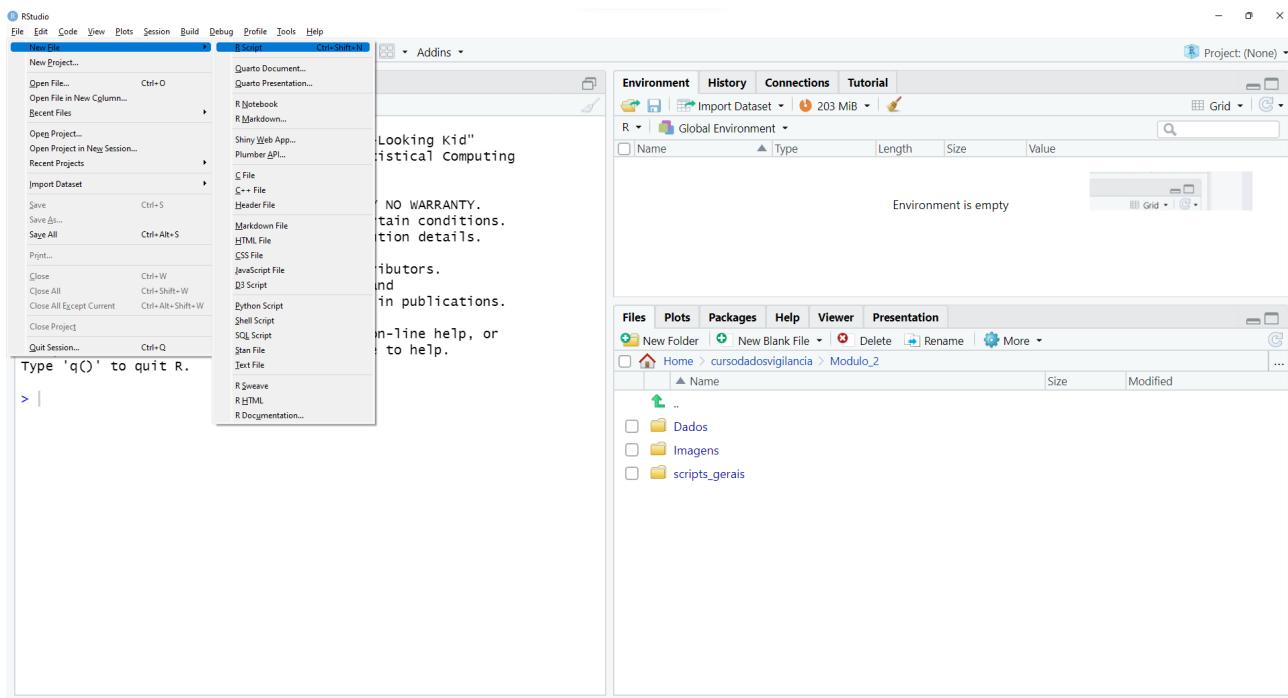
Para encontrar a sua base de dados, de forma rápida, você poderá utilizar o projeto (arquivo no formato `.RProj`) disponível no menu lateral “Arquivos”, do módulo.

Para consultar como criar e/ou utilizar um projeto acesse o Módulo 1.

Acompanhe as telas abaixo, pois vamos fazer um passo a passo!

1. Crie um *script* em branco no `RStudio` seguindo o menu *File, New File e R Script*, repetindo os passos conforme a Figura 7:

**Figura 7: Etapa de criação de um novo script no RStudio .**



**2.** Agora vamos escrever os primeiros comandos para instalar o pacote necessário no corpo do script:

- Digite no RStudio os comandos para instalar o pacote que irá utilizar para importação da base: `if(!require(readr)) install.packages("readr")`
- Em seguida, não esqueça de escrever os comandos para carregar o pacote: `library(readr)`

Veja, como seria construir um código utilizando às boas práticas para o uso de uma linguagem de programação:

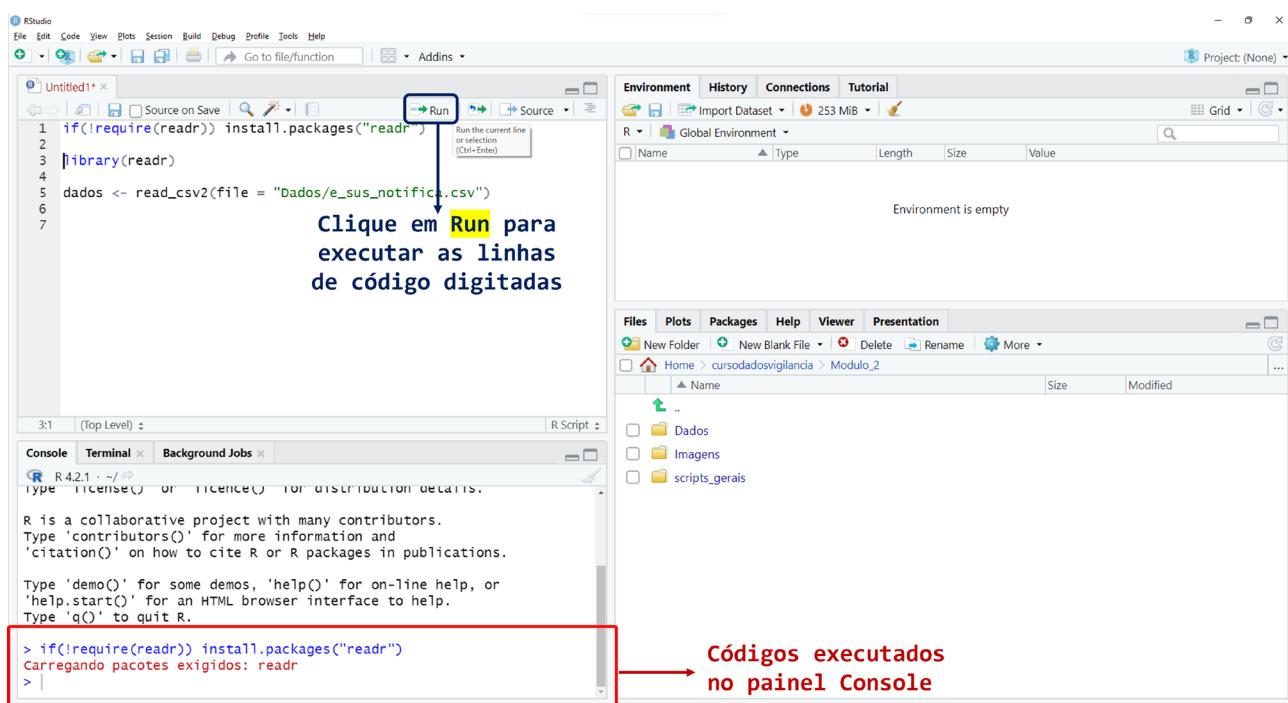
```
# Esta primeira linha verifica se o pacote está instalado.  
# Caso não esteja, o código abaixo irá prosseguir com a instalação do pacote readr  
if(!require(readr)) install.packages("readr")  
  
#Carregando o pacote readr  
library(readr)
```

**3.** Agora que já instalou e carregou os pacotes necessários:

- Clique no botão *Run* (Figura 8) para executar as linhas de código digitadas.
- O **RStudio** é apenas uma interface gráfica para o **R**. Perceba que, ao rodar o código no painel do *script*, os códigos são executados no painel *Console* (Figura 8).

Lembre-se que a primeira linha de código verifica se o pacote está instalado e caso não esteja, será instalado e carregado, conforme a Figura 8.

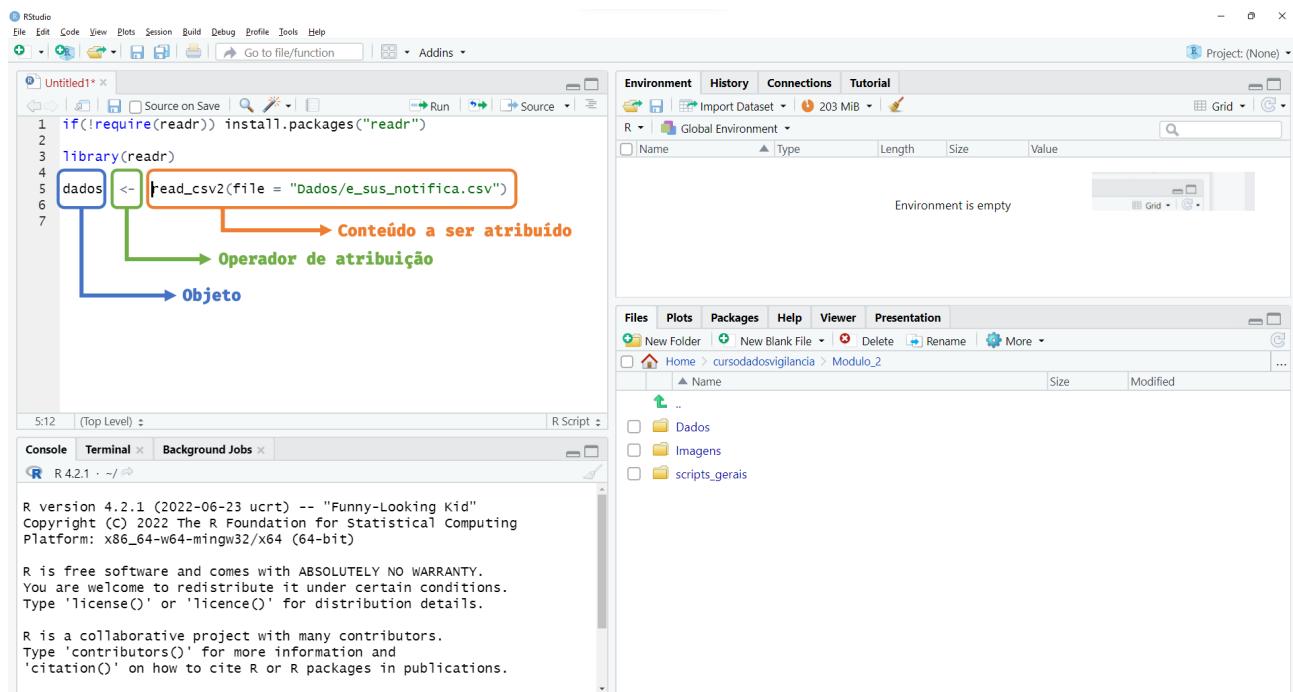
**Figura 8: Etapa de instalação do pacote `readr` no RStudio.**



**4.** Agora, vamos criar um objeto e escolher a função que iremos utilizar para abrir o banco de dados de casos leves de covid-19 (Figura 9):

- Crie no seu **RStudio** o objeto que armazenará o banco de dados: `{dados}`.
- Digite no seu **RStudio** o operador de atribuição `<-`, para sinalizar ao **R** que está armazenando o banco de dados no objeto `{dados}`.
- Digite no seu **RStudio** a função `read_csv2()` do pacote `readr`. Veja na Figura 9 o destaque para a estrutura de uma função para criar um objeto que, neste caso, será uma tabela que armazenando o banco de dados do e-SUS Notifica.
- Também será necessário definir o argumento `file` (arquivo) para que seja possível abrir o banco de dados no **R**. Este argumento solicita que você inclua o local (diretório ou pasta) onde está guardado o arquivo `{e_sus_notifica.csv}` e o nome do arquivo com sua extensão, tudo entre aspas conforme a Figura 9:

**Figura 9: Estrutura de código criando o objeto `dados` no **R**.**



The screenshot shows the RStudio interface with the following code in the script editor:

```

1 if(!require(readr)) install.packages("readr")
2 
3 library(readr)
4 
5 dados <- read_csv2(file = "Dados/e_sus_notifica.csv")
6 
7
  
```

Annotations explain the code structure:

- Conteúdo a ser atribuído**: Points to the variable name `dados`.
- Operador de atribuição**: Points to the assignment operator `<-`.
- Objeto**: Points to the resulting object `dados`.

The RStudio environment pane shows the `dados` object has been created:

Name	Type	Length	Size	Value
dados	Global Environment	1	203 MB	Environment is empty

The file browser pane shows the directory structure:

- Home > cursodadosvigilancia > Modulo\_2
- Dados
- Imagens
- scripts\_gerais

The console pane shows the R version and license information:

```

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.
  
```

Agora, veja os comandos abaixo e valide as linhas de códigos da Figura 9 que você digitou no seu **Rstudio**, e clique em *Run*:

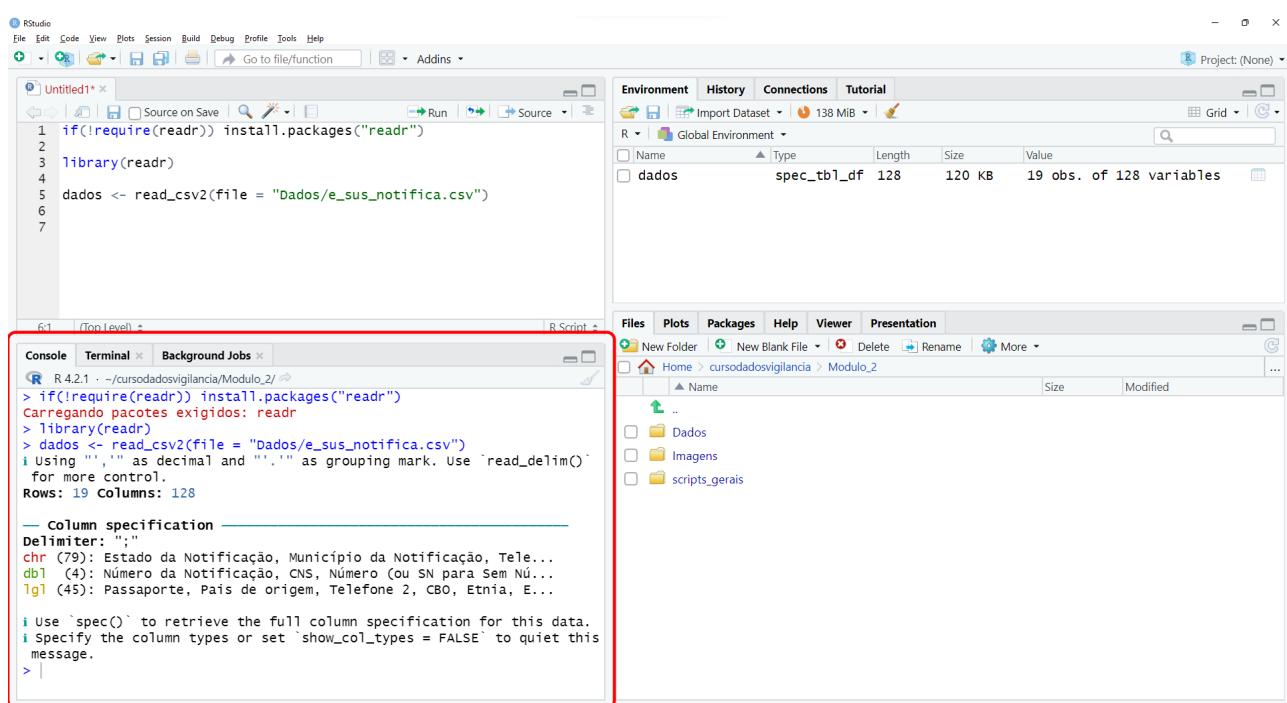
```
# Esta primeira linha verifica se o pacote está instalado.
# Caso não esteja, o código abaixo irá prosseguir com a instalação
if(!require(readr)) install.packages("readr")

# Carregar o pacote readr
library(readr)

# Utilize a função read_csv2 para importar os dados
# do arquivo e_sus_notifica.csv para um objeto chamado "dados"
dados <- read_csv2(file = "Dados/e_sus_notifica.csv")
```

Após clicar em *Run*, a tela do **RStudio** te devolverá resultados dos comandos executados. Observe a Figura 10 onde colocamos em destaque o painel **Console** com os *outputs* de tudo o que acabamos de executar.

**Figura 10: Script de importação de dados do e-SUS Notifica de Rosas, armazenado no objeto `dados` no R.**



The screenshot shows the RStudio interface with the following components visible:

- Code Editor:** Shows the R script code from Figure 10.
- Environment View:** Shows the global environment with a single object named `dados`.
- File Explorer:** Shows the project structure with folders `Dados`, `Imagens`, and `scripts_gerais`.
- Console View:** Contains the output of the R script execution, including the installation of the `readr` package, the loading of the `readr` library, the import of the CSV file into the `dados` object, and the resulting structure of the `dados` object.

A red box highlights the **Console** tab, indicating the focus of the output shown.

Lembre-se que neste curso os blocos em destaque indicam uma simulação de código no ambiente do R.

### **Figura 11: Blocos de código no R.**

① *# Esta primeira linha verifica se o pacote está instalado.  
# Caso não esteja, o código abaixo irá prosseguir com a instalação*  
`if(!require(readr)) install.packages("readr")`

② *#> Carregando pacotes exigidos: readr*



No bloco número 1, de fundo cinza claro, temos os códigos e funções que gostaríamos que você reproduzisse o script no seu RStudio. Perceba que, na primeira e segunda linha desta caixa temos frases iniciando com o símbolo *hashtag* (#). Lembre-se que no R esse símbolo é utilizado para indicar um comentário sobre as etapas de código.

No bloco número 2, temos o **output** da função. É uma reprodução do que será retornado a você quando executar os comandos da caixa número 1. Um output sempre começará com os símbolos hashtag e “maior que”, assim: #>.

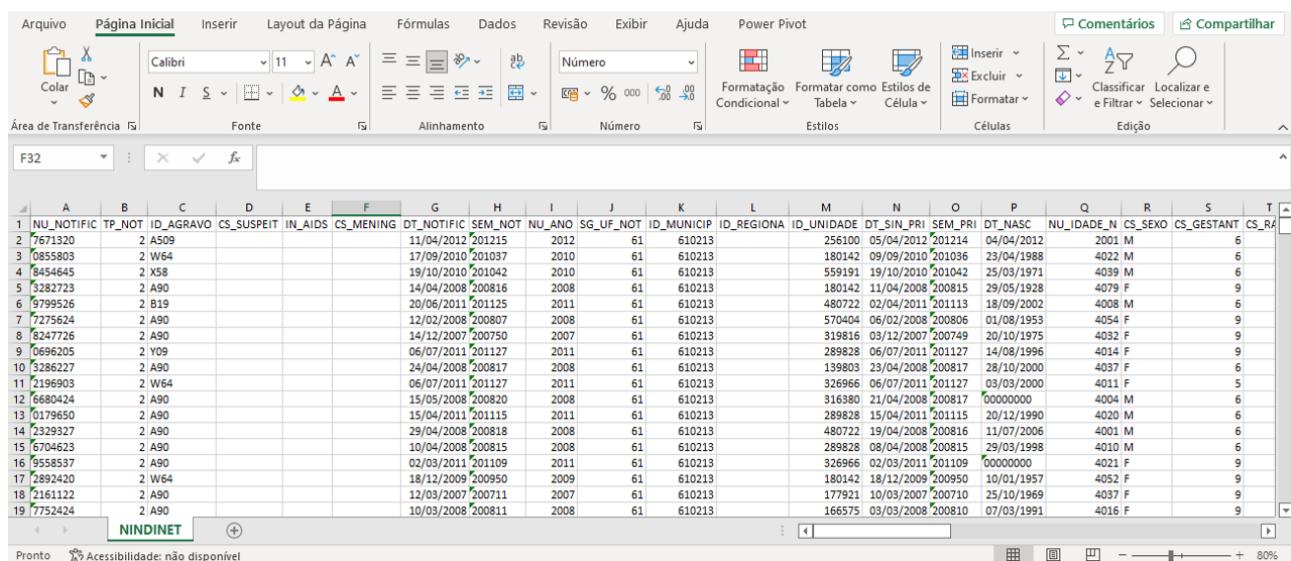
Esta estrutura é apenas para compreensão do curso, ou seja, não é a visualização que você encontrará ao escrever scripts no RStudio e, por isso, acompanhe os prints de telas para apoiá-lo nas dúvidas.

## 2.2 Importando arquivos no formato DBFs

Provavelmente você já esteve em contato com os arquivos do tipo `.dbf`. Eles são muito comuns na análise de dados da vigilância em saúde, pois todos os bancos de dados como Sinan Net, Sinan Online, Sivep Gripe, SIM ou SINASC exportam dados neste tipo de arquivo.

Para aprender a importar um arquivo do tipo `.dbf` iremos utilizar o banco de dados `{NINDINET.dbf}`, exportado do sistema **Sinan Net**, Ficha de Notificação Individual (FIN), do Estado de Rosas. Observe na Figura 12 como é o formato deste banco de dados quando aberto em uma planilha no Microsoft Excel:

**Figura 12: Banco de dados NINDINET.dbf no Microsoft Excel.**



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	
1	NU_NOTIFIC	TP_NOT	ID_AGRAVO	CS_SUSPEIT	IN_AIDS	CS_MENING	DT_NOTIFIC	SEM_NOT	NU_ANO	SG_UF_NOT	ID_MUNICIP	ID_REGIONA	ID_UNIDADE	DT_SIN_PRI	SEM_PRI	DT_NASC	NU_IDADE_N	CS_SEXO	CS_GESTANT	CS_RA
2	7671320	2 A509					11/04/2012	201215	2012	61	610213		256100	05/04/2012	201214	04/04/2012	2001 M		6	
3	0855803	2 W64					17/09/2010	201037	2010	61	610213		180142	09/09/2010	201036	23/04/1988	4022 M		6	
4	6454645	2 X58					19/10/2010	201042	2010	61	610213		559191	19/10/2010	201042	25/03/1971	4039 M		6	
5	3282723	2 A90					14/04/2008	200816	2008	61	610213		180142	11/04/2008	200815	29/05/1928	4079 F		9	
6	6799526	2 B19					20/06/2011	201125	2011	61	610213		480722	02/04/2011	201113	18/09/2002	4008 M		6	
7	2275624	2 A90					12/02/2008	200807	2008	61	610213		570404	06/02/2008	200806	01/08/1953	4054 F		9	
8	6247726	2 A90					14/12/2007	200750	2007	61	610213		319816	03/12/2007	200749	20/10/1975	4032 F		9	
9	6966205	2 Y09					06/07/2011	201127	2011	61	610213		289828	06/07/2011	201127	14/08/1996	4014 F		9	
10	3286227	2 A90					24/04/2008	200817	2008	61	610213		139803	23/04/2008	200817	28/10/2000	4037 F		6	
11	2196903	2 W64					06/07/2011	201127	2011	61	610213		326966	06/07/2011	201127	03/03/2000	4011 F		5	
12	6680424	2 A90					15/05/2008	200820	2008	61	610213		316380	21/04/2008	200817	00000000	4004 M		6	
13	0179650	2 A90					15/04/2011	201115	2011	61	610213		289828	15/04/2011	201115	20/12/1990	4020 M		6	
14	3239327	2 A90					29/04/2008	200818	2008	61	610213		480722	19/04/2008	200816	11/07/2006	4001 M		6	
15	6704623	2 A90					10/04/2008	200815	2008	61	610213		289828	08/04/2008	200815	29/03/1998	4010 M		6	
16	9558537	2 A90					02/03/2011	201109	2011	61	610213		326966	02/03/2011	201109	00000000	4021 F		9	
17	2892420	2 W64					18/12/2009	200950	2009	61	610213		180142	18/12/2009	200950	10/01/1957	4052 F		9	
18	2161122	2 A90					12/03/2007	200711	2007	61	610213		177921	10/03/2007	200710	25/10/1969	4037 F		9	
19	7752424	2 A90					10/03/2008	200811	2008	61	610213		166575	03/03/2008	200810	07/03/1991	4016 F		9	

Para que seja possível manipular esses dados do `{NINDINET.dbf}` no `R`, utilizaremos a função `read.dbf()`, presente no pacote `foreign`. Acompanhe o passo a passo abaixo:

1. Primeiro, certifique-se de que você possui o pacote `foreign` instalado no `R`, e em seguida, faça o carregamento do pacote. Acompanhe o script abaixo e replique-o em seu `RStudio`:

```
# Esta primeira linha verifica se o pacote está instalado.  
# Caso não esteja, irá prosseguir com a instalação do pacote foreign  
if(!require(foreign)) install.packages("foreign")
```

```
#> Carregando pacotes exigidos: foreign
```

```
# Carregue o pacote foreign no RStudio  
library(foreign)
```

2. Agora, você já possui o pacote `foreign` instalado e carregado. Vamos importar o arquivo de nome `{ NINDINET.dbf }` para o ambiente do `R`, armazenando este banco de dados no objeto `{dados_sinan}`. Lembre-se que para este curso, os dados do Estado de Rosas são fictícios.

Observe e replique os comandos do *script* abaixo em seu `RStudio`:

```
# Carregando arquivo NINDINET.dbf no objeto "dados_sinan"  
# Utilizando o argumento "as.is = TRUE" para transformar os dados em caracteres  
dados_sinan <- read.dbf(file = 'Dados/NINDINET.dbf', as.is = TRUE)
```

3. Pronto, já instalamos, carregamos e armazenamos o banco de dados no objeto `{dados_sinan}`. Agora, vamos visualizar as primeiras linhas deste banco de dados no `R` utilizando a função `head()`, digite-a em seu `RStudio` e acompanhe os *outputs*, conforme códigos abaixo:

```
# Com o comando 'head' visualize as primeiras linhas  
# presentes na tabela de dados NINDINET.dbf  
head(dados_sinan)
```

```

#>   NU_NOTIFIC TP_NOT ID_AGRAVO CS_SUSPEIT IN_AIDS CS_MENING DT_NOTIFIC SEM_NOT
#> 1    7671320     2      A509      NA      NA      NA 2012-04-11  201215
#> 2    0855803     2      W64       NA      NA      NA 2010-09-17  201037
#> 3    8454645     2      X58       NA      NA      NA 2010-10-19  201042
#> 4    3282723     2      A90       NA      NA      NA 2008-04-14  200816
#> 5    9799526     2      B19       NA      NA      NA 2011-06-20  201125
#> 6    7275624     2      A90       NA      NA      NA 2008-02-12  200807
#>   NU_ANO SG_UF_NOT ID_MUNICIP ID_REGIONA ID_UNIDADE DT_SIN_PRI SEM_PRI
#> 1    2012      61     610213      NA 256100 2012-04-05  201214
#> 2    2010      61     610213      NA 180142 2010-09-09  201036
#> 3    2010      61     610213      NA 559191 2010-10-19  201042
#> 4    2008      61     610213      NA 180142 2008-04-11  200815
#> 5    2011      61     610213      NA 480722 2011-04-02  201113
#> 6    2008      61     610213      NA 570404 2008-02-06  200806
#>   DT_NASC NU_IDADE_N CS_SEXO CS_GESTANT CS_RACA CS_ESCOL_N SG_UF ID_MN_RESI
#> 1 2012-04-04      2001      M       6      4      10     33 610213
#> 2 1988-04-23      4022      M       6      1     <NA>     33 610213
#> 3 1971-03-25      4039      M       6     NA     <NA>     33 610250
#> 4 1928-05-29      4079      F       9      4      02     33 610213
#> 5 2002-09-18      4008      M       6      4      01     33 610250
#> 6 1953-08-01      4054      F       9      9      09     33 610213
#>   ID_RG_RESI ID_DISTRIT ID_BAIRRO ID_LOGRADO ID_GEO1 ID_GEO2 CS_ZONA ID_PAIS
#> 1      NA      05      020      NA      NA      NA     1     1
#> 2      NA      05      019      NA      NA      NA     1     1
#> 3      NA      05      020      NA      NA      NA     NA     1
#> 4      NA      01      001      NA      NA      NA     1     1
#> 5      NA      01      001      NA      NA      NA     1     1
#> 6      NA      04      014      NA      NA      NA     1     1
#>   NDUPLIC_N IN_VINCULA DT_INVEST ID_OCUPA_N CLASSI_FIN CRITERIO TPAUTOCTO
#> 1      NA      NA     <NA>     <NA>      NA      NA     NA
#> 2      NA      NA     <NA>     <NA>      NA      NA     NA
#> 3      NA      NA 2010-10-19     <NA>      NA      NA     NA
#> 4      NA      NA 2008-04-14     <NA>      5      1     NA
#> 5      NA      NA 2011-06-20 9999991      1      NA     NA
#> 6      NA      NA     <NA>     <NA>      8      NA     NA
#>   COUFINF COPAISINF COMUNINF CODISINF CO_BAINFC NOBAIINF DOENCA_TRA EVOLUCAO
#> 1      NA      0      NA     <NA>      0     <NA>     NA     1
#> 2      NA      0      NA     <NA>      0     <NA>     NA     NA
#> 3      NA      0      NA     <NA>      0     <NA>     NA     NA
#> 4      NA      0      NA     <NA>      0     <NA>     NA     NA
#> 5      NA      0      NA     <NA>      0     <NA>     NA     NA
#> 6      NA      0      NA     <NA>      0     <NA>     NA     NA
#>   DT_OBITO DT_ENCERRA DT_DIGITA DT_TRANSUS DT_TRANSMD DT_TRANSSM DT_TRANSRM
#> 1     <NA>     <NA> 2012-11-09     <NA>     <NA> 2012-11-13     <NA>
#> 2     <NA> 2010-10-16 2010-11-17     <NA>     <NA> 2010-11-23     <NA>
#> 3     <NA> 2010-10-19 2011-03-14     <NA>     <NA> 2011-04-12     <NA>
#> 4     <NA> 2008-06-19 2008-04-24     <NA>     <NA> 2010-11-16     <NA>
#> 5     <NA> 2011-06-20 2011-09-14     <NA>     <NA> 2011-09-19     <NA>
#> 6     <NA> 2008-04-14 2008-02-26     <NA>     <NA> 2010-11-16     <NA>
#>   DT_TRANSRS DT_TRANSSE NU_LOTE_V NU_LOTE_H CS_FLXRET FLXRECEBI MIGRADO_W
#> 1     <NA>     <NA> 2012049      NA      0      2     NA
#> 2     <NA>     <NA> 2010047      NA      0      2     NA
#> 3     <NA>     <NA> 2011015      NA      0      2     NA
#> 4     <NA>     <NA> 2010044      NA      0      2     NA
#> 5     <NA>     <NA> 2011038      NA      1      2     NA
#> 6     <NA>     <NA> 2010043      NA      0      2     NA
#>   CO_USUCAD CO_USUALT
#> 1      NA      NA
#> 2      NA      NA
#> 3      NA      NA
#> 4      NA      NA
#> 5      NA      NA
#> 6      NA      NA

```

Observe no *output* as linhas e colunas deste banco de dados! Foi fácil, não é mesmo? Nesta etapa concluímos o carregamento da base de dados `{NINDINET.dbf}` no `R`. Agora ela já poderá ser utilizada nas análises do Estado de Rosas.



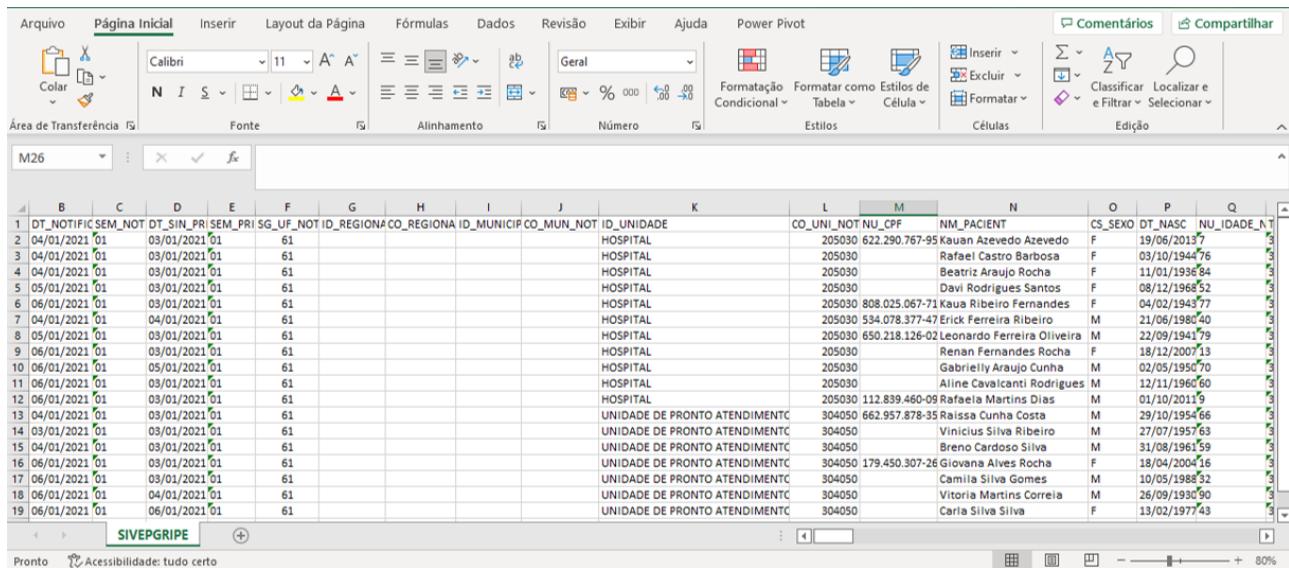
Por padrão, a importação de um arquivo de extensão `.dbf` transforma as variáveis de texto em **fatores** (falaremos mais de fatores adiante no texto). Assim, necessitamos incluir o argumento `as.is = TRUE` na função `read.dbf` para transformar as variáveis do banco de dados em tipo `character` (texto).

## 2.3 Importando arquivos do Microsoft Excel

Comumente utilizamos na vigilância arquivos do Microsoft Excel para manipular alguns bancos de dados que modificamos ou que até mesmo foram criados pelas equipes de vigilância. Você também poderá analisá-los com o `R` com grande praticidade.

Para este exercício você irá importar o arquivo `{sivep_gripe.xlsx}` obtido por meio da exportação do banco de dados do SIVEP Gripe (Sistema de Informação da Vigilância Epidemiológica da Gripe). Esse sistema mantém o registro das notificações de casos hospitalizados por Síndrome Respiratória Aguda Grave (SRAG) e, neste exemplo, vamos utilizar dados fictícios do Estado de Rosas. Veja na Figura 13, a estrutura do arquivo aberto em uma planilha do Excel. É um arquivo que possui uma única planilha chamada `SIVEPGRIPE`.

**Figura 13: Banco de dados {sivep\_gripe.xlsx} no Microsoft Excel.**



B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	
1	DT_NOTIFIC	SEM_NOT	DT_SIN_PRI	SEM_PRI	SG_UF	NOT_ID_REGIONA	CO_REGIONA	ID_MUNICIP	CO_MUN_NOT	ID_UNIDADE	CO_UNI_NOT	NU_CPF	NM_PACIENT	CS_SEXO	DT_NASC	NU_IDADE_NT
2	04/01/2021	01	03/01/2021	01	61					HOSPITAL	205030	622.290.767-95	Kauan Azevedo Azevedo	F	19/06/2013	7
3	04/01/2021	01	03/01/2021	01	61					HOSPITAL	205030	Rafael Castro Barbosa	F	03/10/1947	76	
4	04/01/2021	01	03/01/2021	01	61					HOSPITAL	205030	Beatriz Araujo Rocha	F	11/01/1936	84	
5	05/01/2021	01	03/01/2021	01	61					HOSPITAL	205030	Davi Rodrigues Santos	F	08/12/1965	52	
6	06/01/2021	01	03/01/2021	01	61					HOSPITAL	205030	808.025.067-71	Kaua Ribeiro Fernandes	F	04/02/1943	77
7	04/01/2021	01	04/01/2021	01	61					HOSPITAL	205030	534.078.377-47	Erick Ferreira Ribeiro	M	21/06/1980	40
8	05/01/2021	01	03/01/2021	01	61					HOSPITAL	205030	650.218.126-02	Leonardo Ferreira Oliveira	M	22/09/1947	79
9	06/01/2021	01	03/01/2021	01	61					HOSPITAL	205030	Renan Fernandes Rocha	F	18/12/2007	13	
10	06/01/2021	01	05/01/2021	01	61					HOSPITAL	205030	Gabrielly Araujo Cunha	M	02/05/1950	70	
11	06/01/2021	01	03/01/2021	01	61					HOSPITAL	205030	Aline Cavalcanti Rodrigues	M	12/11/1966	60	
12	06/01/2021	01	03/01/2021	01	61					HOSPITAL	205030	112.839.460-09	Rafaela Martins Dias	M	01/10/2019	
13	04/01/2021	01	05/01/2021	01	61					UNIDADE DE PRONTO ATENDIMENTO	304050	662.957.878-35	Raissa Cunha Costa	M	29/10/1954	66
14	03/01/2021	01	03/01/2021	01	61					UNIDADE DE PRONTO ATENDIMENTO	304050	Vinicius Silva Ribeiro	M	27/07/1957	63	
15	04/01/2021	01	03/01/2021	01	61					UNIDADE DE PRONTO ATENDIMENTO	304050	Breno Cardoso Silva	M	31/08/1961	59	
16	06/01/2021	01	03/01/2021	01	61					UNIDADE DE PRONTO ATENDIMENTO	304050	179.450.307-26	Giovana Alves Rocha	F	18/04/2000	16
17	06/01/2021	01	03/01/2021	01	61					UNIDADE DE PRONTO ATENDIMENTO	304050	Camila Silva Gomes	M	10/05/1983	32	
18	06/01/2021	01	04/01/2021	01	61					UNIDADE DE PRONTO ATENDIMENTO	304050	Vitoria Martins Correia	M	26/09/1959	90	
19	06/01/2021	01	06/01/2021	01	61					UNIDADE DE PRONTO ATENDIMENTO	304050	Carla Silva Silva	F	13/02/1977	43	

Para importar o arquivo {sivep\_gripe.xlsx} para analisá-lo com R, utilizaremos o pacote `readxl` e a sua função `read_excel()`. Vamos lá, como fizemos antes, comece instalando e carregando o pacote `readxl`. Acompanhe o código abaixo e replique os comandos indicados em seu RStudio:

```
# Esta primeira linha verifica se o pacote está instalado.
# Caso não esteja, irá prosseguir com a instalação
if(!require(readxl)) install.packages("readxl")
```

```
#> Carregando pacotes exigidos: readxl
```

```
# Carregue o pacote no RStudio
library(readxl)
```

Com o pacote instalado e carregado, é hora de criar um objeto `{dados_sivep}` e atribuir a ele a função `read_excel()`. Mas atenção, necessitaremos incluir os argumentos `sheet` e `skip` para arquivos neste formato. Veja como fazer:

- o argumento `skip` (pular, em português) é importante para indicar a necessidade de iniciar a importação do arquivo a partir de determinada linha. Por exemplo, se você desejar ler os dados apenas a partir da quarta linha, utilize `skip = 3`.
- o argumento `sheet` é necessário quando queremos indicar para o R qual planilha do arquivo queremos importar, ou seja, neste exemplo devemos definir `sheet = 1` para escolher a primeira planilha. Você pode também substituir o número inteiro “1” pelo nome “SIVEPGRYPE” (Figura 11), escrevendo desta forma o argumento: `sheet = "SIVEPGRYPE"`.

Acompanhe o *script* abaixo e replique estes códigos no seu RStudio:

```
# Importando banco de dados sivep_gripe.xlsx
dados_sivep <- read_excel("Dados/sivep_gripe.xlsx",
                           sheet = "SIVEPGRYPE",
                           skip = 0)

# Agora digite a função `head` para visualizando as primeiras Linhas da tabela
head(dados_sivep)
```

```
#> # A tibble: 6 × 192
#>   NU_NOTIFIC DT_NOTIFIC SEM_NOT DT_SIN...¹ SEM_PRI SG_UF...² ID_RE...³ CO_RE...⁴ ID_MU...⁵
#>   <dbl> <chr> <chr> <chr> <dbl> <lgl> <lgl> <lgl>
#> 1 10039876 04/01/2021 01 03/01/2... 01 61 NA NA NA
#> 2 10042161 04/01/2021 01 03/01/2... 01 61 NA NA NA
#> 3 10043051 04/01/2021 01 03/01/2... 01 61 NA NA NA
#> 4 10045529 05/01/2021 01 03/01/2... 01 61 NA NA NA
#> 5 10046813 06/01/2021 01 03/01/2... 01 61 NA NA NA
#> 6 10047778 04/01/2021 01 04/01/2... 01 61 NA NA NA
#> # ... with 183 more variables: CO_MUN_NOT <lgl>, ID_UNIDADE <chr>,
#> # CO_UNI_NOT <dbl>, NU_CPF <chr>, NM_PACIENT <chr>, CS_SEXO <chr>,
#> # DT_NASC <chr>, NU_IDADE_N <chr>, TP_IDADE <chr>, COD_IDADE <chr>,
#> # CS_GESTANT <chr>, CS_RACA <chr>, CS_ETINIA <lgl>, CS_ESCOL_N <chr>,
#> # NM_MAE_PAC <chr>, NU_CEP <chr>, ID_PAIS <chr>, CO_PAIS <chr>, SG_UF <lgl>,
#> # ID_RG_RESI <lgl>, CO_RG_RESI <chr>, ID_MN_RESI <chr>, CO_MUN_RES <lgl>,
#> # NM_BAIRRO <chr>, NM_LOGRADO <chr>, NU_NUMERO <chr>, NM_COMPLEM <lgl>, ...
```



Por padrão a função `read_excel()` sempre irá identificar, ou seja, lerá a primeira planilha e sua tabela, a partir da primeira linha não-vazia.

Portanto ao chamar a função sem incluir os argumentos `sheet` e `skip`, o R utilizará seu padrão de configuração e lhe retornará como resultado a mesma visualização que apresentamos na Figura 11.

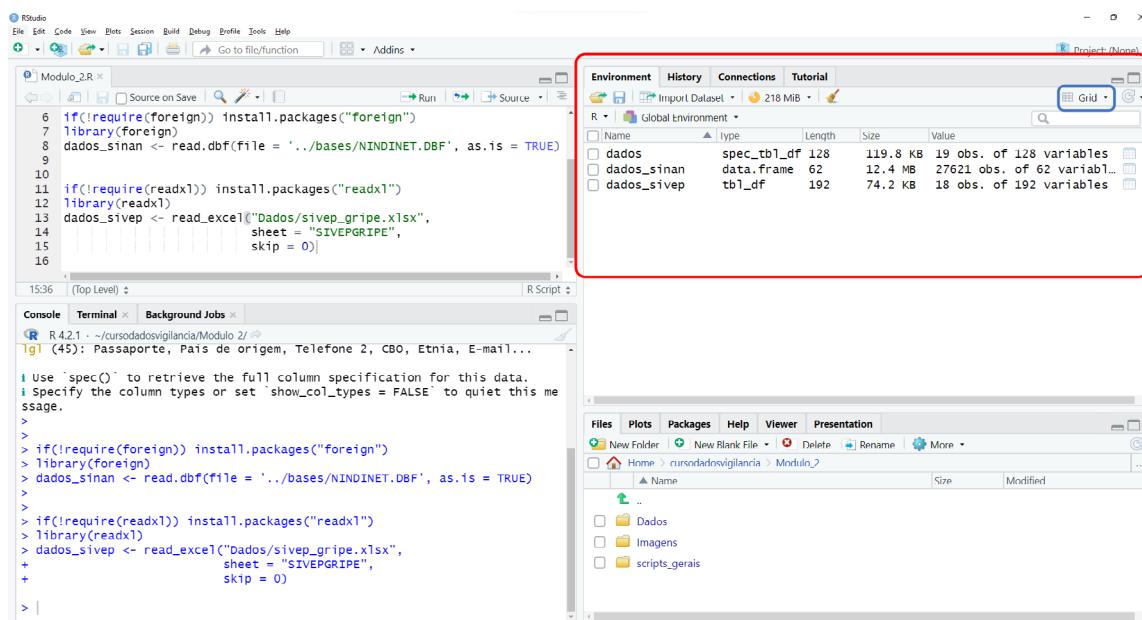
## 2.4 Visualizando os arquivos importados para o R

Mas e se quisermos visualizar as tabelas que armazenamos nos objetos `{dados}`, `{dados_sinan}` ou `{dados_sivep}`?

O R permite que a consulta seja a qualquer momento, isto porque ele armazena cada objeto salvo de uma forma intuitiva e rápida, utilizando o painel `Environment` (Figuras 14 e 15). Ele está localizado, no canto direito superior do RStudio!

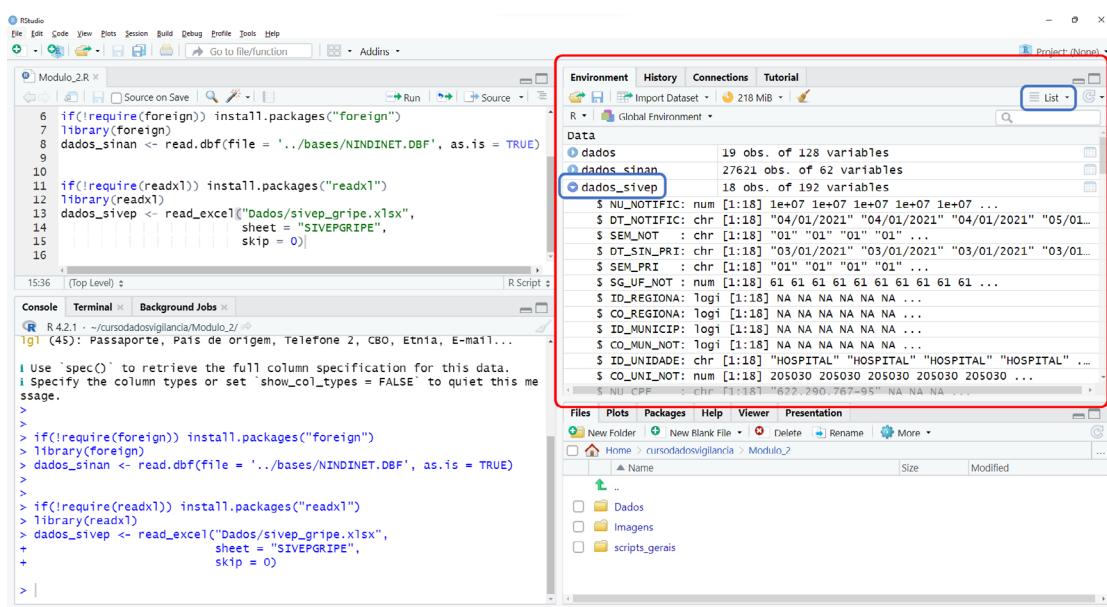
O painel `Environment` possui duas formas de visualização. Em destaque da Figura 14, perceba o botão marcado como `Grid`, ele te permitirá a visualização das tabelas armazenadas identificando os objetos por nome, tipo, comprimento (no caso de banco de dados é número de linhas), tamanho (em bytes) e uma breve sumário. Veja:

**Figura 14: Tela de apresentação do painel Environment no formato Grid para consulta dos objetos criados.**



Já na Figura 15, o modo de visualização é no formato lista (List), onde é possível visualizar as variáveis e o tipo de cada um dos bancos de dados, bastando clicar no botão azul localizado na frente de cada objeto. Veja:

**Figura 15: Tela do painel Environment no formato List para consulta dos tipos de dados dos objetos criados.**

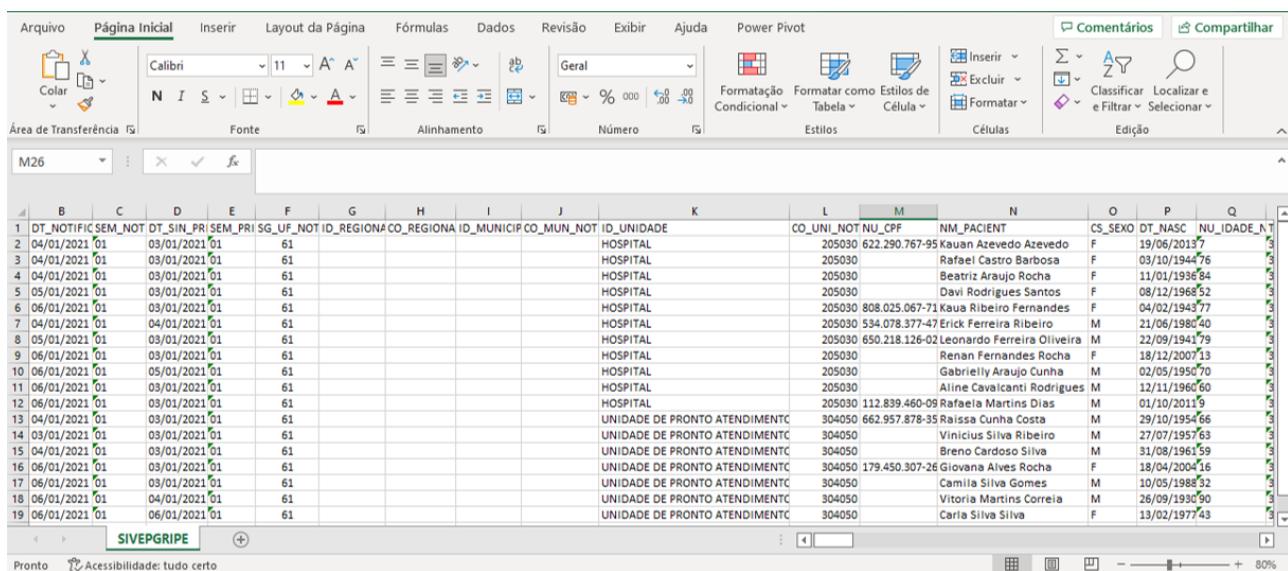


### *3. Tipo de variáveis*

Quais os tipos de dados estão presentes em uma análise? Antes de mais nada, vamos falar sobre o conteúdo presente nas variáveis (colunas). Elas podem conter qualquer característica ou atributo coletado no dia a dia da vigilância, como valores referentes à identificação de uma pessoa como o bairro, sexo ou sua idade, tudo isso pode ser chamado de **variável**.

Observe na Figura 16 o banco de dados exportado do Sivep Gripe com suas variáveis destacadas.

**Figura 16: Tabela do Sivep Gripe com suas variáveis (colunas) para análise.**



	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q		
1	DT_NOTIFIC	SEM_NOT	DT_SIN_PRI	SEM_PRI	SG_UF_NOT	ID_REGIONA	CO_REGIONA	ID_MUNICIP	CO_MUN_NOT	ID_UNIDADE			CO_UNI_NOT	NU_CPF	NM_PACIENT	CS_SEXO	DT_NASC	NU_IDADE_NT
2	04/01/2021	01	03/01/2021	01	61					HOSPITAL	205030	622.290.767-95	Kauan Azevedo Azevedo	F	19/06/2015	5		
3	04/01/2021	01	03/01/2021	01	61					HOSPITAL	205030	Rafael Castro Barbosa	F	03/10/1944	76			
4	04/01/2021	01	03/01/2021	01	61					HOSPITAL	205030	Beatriz Araújo Rocha	F	11/01/1936	84			
5	05/01/2021	01	03/01/2021	01	61					HOSPITAL	205030	Davi Rodrigues Santos	F	08/12/1968	52			
6	06/01/2021	01	03/01/2021	01	61					HOSPITAL	205030	808.025.067-71	Kaua Ribeiro Fernandes	F	04/02/1943	77		
7	04/01/2021	01	04/01/2021	01	61					HOSPITAL	205030	534.078.377-47	Erick Ferreira Ribeiro	M	21/06/1980	40		
8	05/01/2021	01	03/01/2021	01	61					HOSPITAL	205030	650.218.126-02	Leonardo Ferreira Oliveira	M	22/09/1941	79		
9	06/01/2021	01	03/01/2021	01	61					HOSPITAL	205030		Renan Fernandes Rocha	F	18/12/2007	13		
10	06/01/2021	01	05/01/2021	01	61					HOSPITAL	205030		Gabrielly Araújo Cunha	M	02/05/1956	70		
11	06/01/2021	01	03/01/2021	01	61					HOSPITAL	205030		Aline Cavalcanti Rodrigues	M	12/11/1966	60		
12	06/01/2021	01	03/01/2021	01	61					HOSPITAL	205030	112.839.460-09	Rafaela Martins Dias	M	01/10/2019	1		
13	04/01/2021	01	03/01/2021	01	61					UNIDADE DE PRONTO ATENDIMENTO	304050	662.957.878-35	Raissa Cunha Costa	M	29/10/1954	66		
14	03/01/2021	01	03/01/2021	01	61					UNIDADE DE PRONTO ATENDIMENTO	304050		Vinicius Silva Ribeiro	M	27/07/1957	63		
15	04/01/2021	01	03/01/2021	01	61					UNIDADE DE PRONTO ATENDIMENTO	304050		Breno Cardoso Silva	M	31/08/1967	59		
16	06/01/2021	01	03/01/2021	01	61					UNIDADE DE PRONTO ATENDIMENTO	304050	179.450.307-26	Giovana Alves Rocha	F	18/04/2004	16		
17	06/01/2021	01	03/01/2021	01	61					UNIDADE DE PRONTO ATENDIMENTO	304050		Camila Silva Gomes	M	10/05/1988	32		
18	06/01/2021	01	04/01/2021	01	61					UNIDADE DE PRONTO ATENDIMENTO	304050		Vitoria Martins Correia	M	26/09/1990	30		
19	06/01/2021	01	06/01/2021	01	61					UNIDADE DE PRONTO ATENDIMENTO	304050		Carla Silva Silva	F	13/02/1977	43		

Essas variáveis podem ser classificadas a partir do tipo de dados que armazenam (Figura 17), podendo ser **quantitativas** quando armazenam dados de medidas como um número, ou **qualitativas** quando registram as características que não são mensuráveis, como textos. Não se preocupe, ao final deste módulo você estará pronto para analisar qualquer tipo de variável.

**Figura 17: Tabela de classificação dos tipos de dados.**

Classificação	Tipo	Tipo reconhecido pelo R	Exemplo de Variável	Valores
<b>Qualitativa</b>				
Nominais	Texto	<i>character</i>	Município	Rosas
				(0) Sem escolaridade (1) Fundamental 1º ciclo (2) Fundamental 2º ciclo (3) Médio (4) Superior (5) Não se aplica (9) Ignorado
<b>Quantitativas</b>				
Discretas	Números inteiros	<i>numeric</i> ou <i>integer</i>	Número de casos	125
Contínuas	Números decimais	<i>numeric</i> ou <i>double</i>	Taxa de incidência	120.38

### 3.1 Variáveis quantitativas

Os dados contidos em variáveis **quantitativas** são aquelas medidas como um número, com caráter objetivo. Podem ser do tipo **discreta** ou do tipo **contínua**:

- **Discretas:** podem assumir somente valores inteiros obtidos por contagem. São exemplos de variáveis numéricas discretas o número de gestações anteriores constante na Declaração de Nascido Vivo (SINASC), o número de contatos examinados coletado na ficha de Tuberculose e o número de lesões cutâneas dermatológicas apresentadas pelo paciente coletado na ficha de Hanseníase.
- **Contínuas:** admitem qualquer valor numérico em determinado intervalo de variação. São exemplos de variáveis numéricas contínuas o peso ao nascer do recém-nascido, coletado na Declaração de Nascido Vivo (SINASC), resultados de exames de glicemia, aferição de pressão.

De modo geral, as variáveis quantitativas no R são numéricas do tipo inteiro (`integer`) e do tipo decimal (`double`).

#### ATENÇÃO



Ao realizar a etapa de importação de um banco de dados, não necessariamente o tipo da variável estará definido corretamente. **O profissional de vigilância que analisa dados deve sempre avaliar, ou seja, inspecionar o banco de dados que está analisando.**

Para inspecionar o tipo de variável do banco, vamos utilizar a função `sapply()`. Mas para utilizá-la, necessitamos definir os seus principais argumentos da seguinte maneira:

- argumento `X`: incluir qual será o banco de dados, no caso `{dados_sivep}`;
- argumento `FUN`: permite que a gente configure o que queremos visualizar ou saber de cada variável ou coluna do banco de dados. Aqui, escolheremos argumento `FUN = typeof`, isso porque nos permitirá visualizar o tipo (`type`, em inglês) de dado armazenado em cada variável.

Observe e reproduza o *script* abaixo no seu RStudio e verifique se o *output* é similar ao que apresentamos:

```
sapply(X = dados_sivep, FUN = "typeof")
```

```
#> NU_NOTIFIC DT_NOTIFIC SEM_NOT DT_SIN_PRI SEM_PRI SG_UF_NOT
#> "double" "character" "character" "character" "character" "double"
#> ID_REGIONA CO_REGIONA ID_MUNICIP CO_MUN_NOT ID_UNIDADE CO_UNI_NOT
#> "logical" "logical" "logical" "logical" "character" "double"
#> NU_CPF NM_PACIENT CS_SEXO DT_NASC NU_IDADE_N TP_IDADE
#> "character" "character" "character" "character" "character" "character"
#> COD_IDADE CS_GESTANT CS_RACA CS_ETINIA CS_ESCOL_N NM_MAE_PAC
#> "character" "character" "character" "logical" "character" "character"
#> NU_CEP ID_PAIS CO_PAIS SG_UF ID_RG_RESI CO_RG_RESI
#> "character" "character" "character" "logical" "logical" "character"
#> ID_MN_RESI CO_MUN_RES NM_BAIRRO NM_LOGRADO NU_NUMERO NM_COMPLEM
#> "character" "logical" "character" "character" "character" "logical"
#> NU_DDD_TEL NU_TELEFON CS_ZONA SURTO_SG NOSOCOMIAL AVE_SUINO
#> "double" "character" "character" "character" "character" "character"
#> FEBRE TOSSE GARGANTA DISPNEIA DESC_RESP SATURACAO
#> "character" "character" "character" "character" "character" "character"
#> DIARREIA VOMITO OUTRO_SIN OUTRO_DES FATOR_RISC PUERPERA
#> "character" "character" "character" "character" "character" "character"
#> CARDIOPATI HEMATOLOGI SIND_DOWN HEPATICA ASMA DIABETES
#> "character" "character" "character" "character" "character" "character"
#> NEUROLOGIC PNEUMOPATI IMUNODEPRE RENAL OBESIDADE OBES_IMC
#> "character" "character" "character" "character" "character" "logical"
#> OUT_MORBI MORB_DESC VACINA DT_UT_DOSE MAE_VAC DT_VAC_MAE
#> "character" "character" "character" "character" "logical" "logical"
#> M_AMAMENTA DT_DOSEUNI DT_1_DOSE DT_2_DOSE ANTIVIRAL TP_ANTIVIR
#> "logical" "logical" "logical" "logical" "character" "character"
#> OUT_ANTIV DT_ANTIVIR HOSPITAL DT_INTERNA SG_UF_INTE ID_RG_INTE
#> "logical" "character" "character" "double" "logical" "logical"
#> CO_RG_INTE ID_MN_INTE CO_MU_INTE NM_UN_INTE CO_UN_INTE UTI
#> "logical" "logical" "logical" "logical" "logical" "character"
#> DT_ENTUTI DT_SAIDUTI SUPORT_VEN RAIOX_RES RAIOX_OUT DT_RAIOX
#> "character" "character" "character" "character" "character" "character"
#> AMOSTRA DT_COLETA TP_AMOSTRA OUT_AMOST REQUI_GAL PCR_RESUL
#> "character" "character" "character" "logical" "logical" "character"
#> DT_PCR POS_PCRFLU TP_FLU_PCR PCR_FLUASU FLUASU_OUT PCR_FLUBLI
#> "character" "character" "logical" "logical" "logical" "logical"
#> FLUBLI_OUT POS_PCRROUT PCR_VSR PCR_PARA1 PCR_PARA2 PCR_PARA3
#> "logical" "character" "logical" "logical" "logical" "logical"
#> PCR_PARA4 PCR_ADEN0 PCR_METAP PCR_BOCA PCR_RINO PCR_OUTRO
#> "logical" "logical" "logical" "logical" "logical" "logical"
#> DS_PCR_OUT LAB_PCR CO_LAB_PCR CLASSI_FIN CLASSI_OUT CRITERIO
#> "logical" "character" "double" "character" "logical" "character"
#> EVOLUCAO DT_EVOLUCA DT_ENCERRA OBSERVA NOME_PROF REG_PROF
#> "character" "character" "character" "character" "logical" "character"
#> DT_DIGITA HISTO_VGM PAIS_VGM CO_PS_VGM LO_PS_VGM DT_VGM
#> "double" "character" "logical" "logical" "logical" "logical"
#> DT_RT_VGM PCR_SARS2 PAC_COCBO PAC_DSCBO OUT_ANIM DOR_ABD
#> "logical" "character" "logical" "logical" "logical" "character"
#> FADIGA PERD_DLFT PERD_PALA TOMO_RES TOMO_OUT DT_TOMO
#> "character" "character" "character" "double" "character" "character"
#> TP_TES_AN DT_RES_AN RES_AN LAB_AN CO_LAB_AN POS_AN_FLU
#> "double" "logical" "character" "logical" "logical" "logical"
#> TP_FLU_AN POS_AN_OUT AN_SARS2 AN_VSR AN_PARA1 AN_PARA2
#> "logical" "logical" "logical" "logical" "logical" "logical"
#> AN_PARA3 AN_ADEN0 AN_OUTRO DS_AN_OUT TP_AM_SOR SOR_OUT
#> "logical" "logical" "logical" "logical" "double" "logical"
#> DT_CO_SOR TP_SOR OUT_SOR DT_RES RES_IGG RES_IGM
#> "character" "double" "logical" "character" "character" "character"
#> RES_IGA NU_DO POV_CT TP_POV_CT TEM_CPF ESTRANG
#> "logical" "double" "character" "logical" "logical" "logical"
#> NU_CNS VACINA_COV DOSE_1_COV DOSE_2_COV DOSE_REF FAB_COV_1
#> "logical" "logical" "logical" "logical" "logical" "logical"
#> FAB_COV_2 FAB_COVREF LOTE_1_COV LOTE_2_COV LOTE_REF FNT_IN_COV
#> "logical" "logical" "logical" "logical" "logical" "logical"
```

Perceba que ao executar a função `sapply()` no *output* acima é possível visualizar o nome das variáveis e, logo abaixo de cada uma delas, seu tipo (entre aspas). Observe a variável `NU_IDADE_N`, que representa a idade do paciente internado. Ela foi classificada pelo R como `character` (texto), porém sabemos que idade não é um texto, ela apenas está sendo classificada como. As variáveis devem sempre ser classificadas corretamente, pois para rodar as suas análises calculando medidas como frequência, média ou qualquer outro cálculo, será necessário indicar a função adequada vinculando-a ao tipo de variável para que o R execute o código sem erros.

## 3.2 Variáveis qualitativas

Quanto a variáveis **qualitativas**, são aquelas que registram as características particulares e que não são mensuráveis como um número e, geralmente, são definidas como categorias. Podem ser classificadas em **ordinais** e **nominais**:

- **Ordinais:** são aquelas que suas categorias expressam uma ordem ou hierarquia. Exemplos comuns são aquelas variáveis que representam uma resposta de uma avaliação: *muito ruim, ruim, regular, bom* e *excelente*. Obviamente a classificação *excelente* é melhor do que *bom* e assim sucessivamente. Na Vigilância, temos o exemplo da variável grau de incapacidade física no diagnóstico (notificação de Hanseníase do SINAN), a variável escolaridade categorizada, e outras.
- **Nominais:** são as que possuem categorias que não têm nenhuma ordem entre elas. Podem ser dicotômicas (duas categorias) ou ter várias categorias (politômicas). Como exemplos há coluna com dados de sexo (Masculino, Feminino e Ignorado), o resultado de um exame sorológico (Positivo, Negativo, Inconclusivo, Não realizado) ou até mesmo a ocorrência de uma hospitalização (Sim, Não, Ignorado).

É importante que você saiba que as variáveis qualitativas no R podem ser do tipo fator (`factor`) ou somente do tipo texto (`character`). Os fatores são um tipo de objeto no qual as categorias das variáveis são chamadas de `levels` (níveis ou hierarquias) e podem possuir um rótulo para cada categoria, chamado de `label` (nome ou rótulos). Estas categorias podem ser estruturadas no formato de:

- números inteiros (`integer`), ou
- textos (`character`).



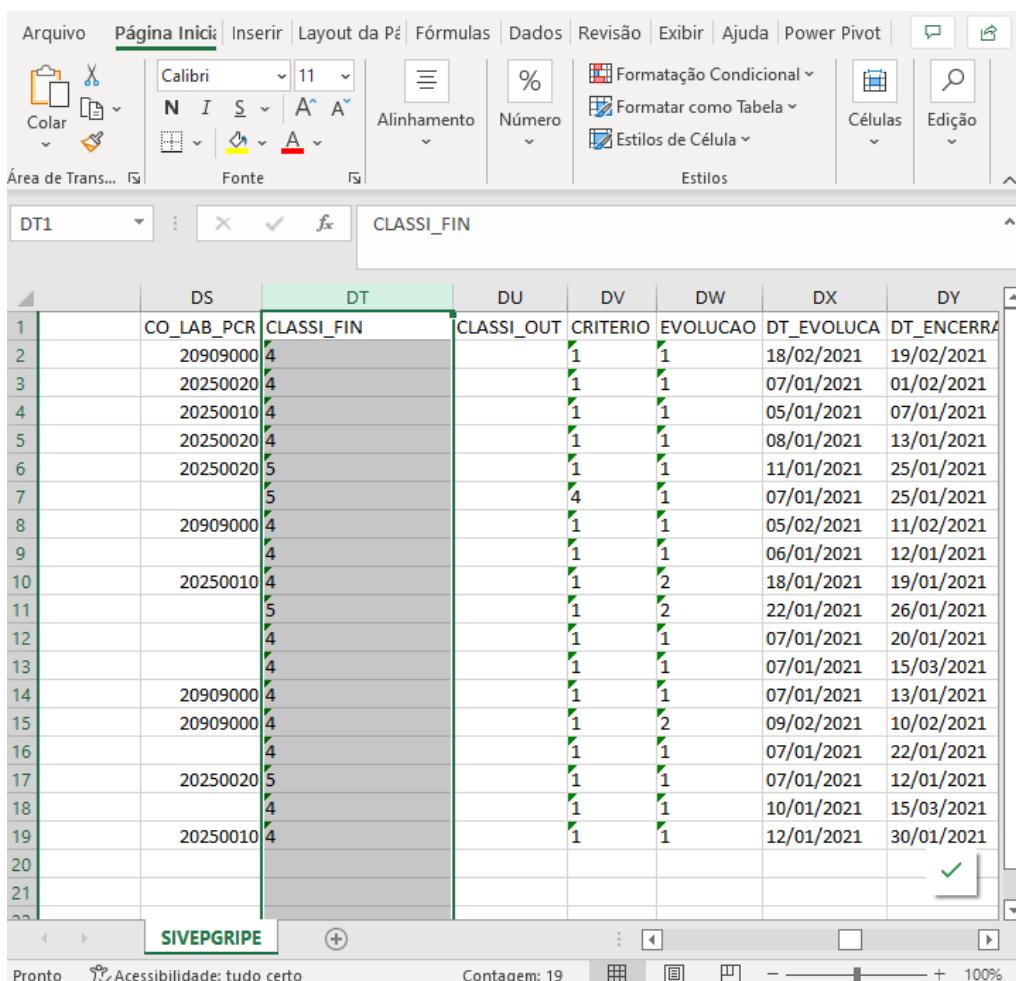
### ATENÇÃO

É importante destacar que para a manipulação de uma variável no R é preciso que ela seja transformada em fator. Para isso utiliza-se a função `factor()`.

Pronto, vamos praticar! Imagine que você necessita analisar a classificação final dos casos de SRAG do Estado de Rosas. Para isto necessitaremos transformar todos os dados contidos na variável `CLASSI_FIN`, do banco de dados `{dados_sivep}`, e para isso utilizaremos o dicionário do banco de dados disponível no menu lateral “Arquivos”, do módulo.

Observe na Figura 18 abaixo como está estruturada a variável `CLASSI_FIN` do banco de dados `{dados_sivep}`:

**Figura 18: Variável `CLASSI_FIN` do Sivep Gripe visualizada pelo Microsoft Excel.**



	DS	DT	CLASSI_FIN	CLASSI_OUT	CRITERIO	EVOLUCAO	DT_EVOLUCA	DT_ENCERRA
1	CO_LAB_PCR							
2	20909000	4			1	1	18/02/2021	19/02/2021
3	20250020	4			1	1	07/01/2021	01/02/2021
4	20250010	4			1	1	05/01/2021	07/01/2021
5	20250020	4			1	1	08/01/2021	13/01/2021
6	20250020	5			1	1	11/01/2021	25/01/2021
7		5			4	1	07/01/2021	25/01/2021
8	20909000	4			1	1	05/02/2021	11/02/2021
9		4			1	1	06/01/2021	12/01/2021
10	20250010	4			1	2	18/01/2021	19/01/2021
11		5			1	2	22/01/2021	26/01/2021
12		4			1	1	07/01/2021	20/01/2021
13		4			1	1	07/01/2021	15/03/2021
14	20909000	4			1	1	07/01/2021	13/01/2021
15	20909000	4			1	2	09/02/2021	10/02/2021
16		4			1	1	07/01/2021	22/01/2021
17	20250020	5			1	1	07/01/2021	12/01/2021
18		4			1	1	10/01/2021	15/03/2021
19	20250010	4			1	1	12/01/2021	30/01/2021
20								
21								

Conforme dicionário de dados, a classificação final do caso (Figura 18) pode ser referida da seguinte forma:

- 1 = SRAG por influenza;
- 2 = SRAG por outro vírus respiratório;
- 3 = SRAG por outro agente etiológico;
- 4 = SRAG não especificado;
- 5 = SRAG por COVID-19.

Observe a Figura 19 com o print de parte do dicionário de dados:

**Figura 19: Dicionário de dados do Sivep Gripe da variável CLASSI\_FIN.**

MINISTÉRIO DA SAÚDE  
SECRETARIA DE VIGILÂNCIA EM SAÚDE

### Dicionário de Dados

#### FICHA DE REGISTRO INDIVIDUAL – CASOS DE SÍNDROME RESPIRATÓRIA AGUDA GRAVE HOSPITALIZADOS

Este documento tem como finalidade descrever as variáveis exportadas para o banco de dados em DBF.

**CAMPO OBRIGATÓRIO** é aquele cuja ausência de dado impossibilita a inclusão do registro no sistema.  
**CAMPO ESSENCIAL** é aquele que, apesar de não ser obrigatório, registra dado necessário à investigação do caso ou ao cálculo de indicador epidemiológico ou operacional.  
**CAMPO INTERNO** é aquele que apesar de não constar na ficha e não aparecer no display da tela, é preenchido automaticamente pelo sistema.  
**CAMPO OPCIONAL** é aquele que só deve ser preenchido caso seja necessário, aparece no display da tela e consta no banco de dados.

Nome do campo	Tipo	Categoria	Descrição	Características	DBF
75-Classificação final do caso	Varchar2(1)	1-SRAG por influenza 2-SRAG por outro vírus respiratório 3-SRAG por outro agente etiológico, qual: 4-SRAG não especificado 5-SRAG por covid-19	Diagnóstico final do caso.  Se tiver resultados divergentes entre as metodologias laboratoriais, priorizar o resultado do RT-PCR.	Campo Obrigatório	CLASSI_FIN

Agora precisaremos criar um fator, utilizando a `factor()` para recodificar, ou categorizar a variável `CLASSI_FIN` do Sivep Gripe. Para isso, precisaremos:

1. Definir um novo objeto para salvar as modificações escrevendo o comando: `dados_sivep$CS_CLASSI_FIN_N`. Onde criaremos a nova variável `CS_CLASSI_FIN_N`, selecionada utilizando cifrão (\$), do banco de dados {`dados_sivep`}.
- Lembre-se de utilizar o sinal de atribuição `<-`, após a expressão `dados_sivep$CS_CLASSI_FIN_N`, pois ele garantirá que todos os comandos solicitados após sua escrita estarão salvos, de forma automática, na nova variável `CS_CLASSI_FIN_N`.
2. E aplicar os três argumentos principais da função `factor()`:
  - `x`: indicando os dados que serão categorizados;
  - `levels`: indicando os valores que serão utilizados como categorias (hierarquia);
  - `labels`: indicando os nomes (rótulos) que vão identificar as categorias.

Observe o *script* abaixo e replique-o em seu `RStudio`.

## ATENÇÃO



O sinal de cifrão `$` indica que estamos realizando uma operação de selecionar uma variável no banco de dados escolhido.

Fique atento à escrita do seu script no `RStudio`, pois qualquer erro ortográfico ou de pontuação pode levar a dificuldades de rodar o seu código, e o `R` te apresentará um aviso (*warning*)!

```
# Recodificando a coluna CLASSI_FIN utilizando a função factor
# e salvando as modificações na nova coluna CS_CLASSI_FIN_N

dados_sivep$CS_CLASSI_FIN_N <- factor(
  x = dados_sivep$CLASSI_FIN,
  levels = c("1", "2", "3", "4", "5"),
  labels = c(
    "SRAG por influenza",
    "SRAG por outro vírus respiratório",
    "SRAG por outro agente etiológico",
    "SRAG não especificado",
    "SRAG por COVID-19"
  )
)
```

Observe que executamos os seguintes argumentos na função `factor ()`:

1. `x = dados_sivep$CLASSI_FIN` indicando o banco de dados `{dados_sivep}` e selecionando com o símbolo `$` (cifrão) a coluna (`CLASSI_FIN`) que será recodificada.
2. `levels = c("1", "2", "3", "4", "5")` indicando quais são os dados que precisam ser transformados.
3. `labels = c("SRAG por influenza", "SRAG por outro vírus respiratório", "SRAG por outro agente etiológico", "SRAG não especificado", "SRAG por COVID-19")` indicando quais são os valores que substituiram os valores citados no `levels` (item 2).

Observe também que no *output* visualizamos as categorias recodificadas corretamente.

Quando transformamos uma variável em `factor`, estamos guardando no `R` cada uma das suas categorias ("1","2","3","4","5") e associando uma identificação para cada uma delas, assim:

- Onde está codificado como "1", receberá o rótulo "SRAG por influenza";
- Onde está codificado como "2", receberá o rótulo "SRAG por outro vírus - respiratório";
- Onde está codificado como "3", receberá o rótulo "SRAG por outro agente etiológico";
- Onde está codificado como "4", receberá o rótulo "SRAG não especificado";
- Onde está codificado como "5", receberá o rótulo "SRAG por COVID-19".

**O número de valores informados no argumento `levels` deve ser o mesmo do número de valores informados no argumento `labels`.**

Perceba que utilizamos a função `c()` no argumento `levels` e `labels`. A função `c()` tem o objetivo de concatenar valores em um conjunto. Como estamos criando um conjunto de textos para os argumentos, os colocamos entre aspas duplas, separando-os por vírgulas.

**Essa é uma estrutura comum nas funções que utilizaremos durante o nosso curso.**

Na linguagem de programação R, para ordenar por ordem hierárquica uma variável ordinal precisamos definir as categorias no argumento `levels` da função `factor` e adicionar o argumento `ordered = TRUE`. Veja no código abaixo como ficaria a recategorização da escolaridade, e replique-o em seu RStudio:



```

dados_sivep$CS_ESCOL_N <- factor(
  x = dados_sivep$CS_ESCOL_N,
  levels = c("0", "1", "2", "3", "4", "5", "9"),
  labels = c(
    "Sem escolaridade",
    "Fundamental 1º ciclo",
    "Fundamental 2º ciclo",
    "Médio",
    "Superior",
    "Não se aplica",
    "Ignorado"
  ),
  ordered = TRUE
)

```

## 4. Como obter ajuda para uso do R?

Muitas vezes, quando estamos escrevendo *scripts* nos deparamos com erros ou uma nova função, principalmente quando estamos aprendendo a analisar dados por meio do software R. Nestes momentos, necessitamos de ajuda para entender algum conceito, a utilizar pacotes, suas funções e a como escrever seus argumentos.

Isso é normal. Todos que trabalham com linguagem de programação acessam mídias sociais, fóruns e páginas da web em busca de informações, técnicas e métodos que ajude a solucionar desafios ou problemas. Algumas das principais formas para se pedir ajuda são:

- a. O Help ou documentação do R [link de acesso: <https://cran.r-project.org/manuals.html>].
- b. Google [link de acesso: <https://www.google.com>].
- c. Fóruns como o Stack Overflow [link de acesso: <https://pt.stackoverflow.com>].
- d. Grupos com outros profissionais de saúde que analisam dados com R.

### a) O Help ou documentação do R

Durante a sua escrita de *script* você pode acessar informações no próprio R. Para acessar essas informações é necessário digitar um ponto de interrogação (?) seguido do nome da função ou comando que quer executar. Ou ainda pode-se, utilizar a função `help()` com o nome da função entre parênteses no script e clicar em *run*. Faça um teste em seu Rstudio e digite a função `help()` para a função de raiz quadrada `sqrt()`:

```
help(sqrt)
```

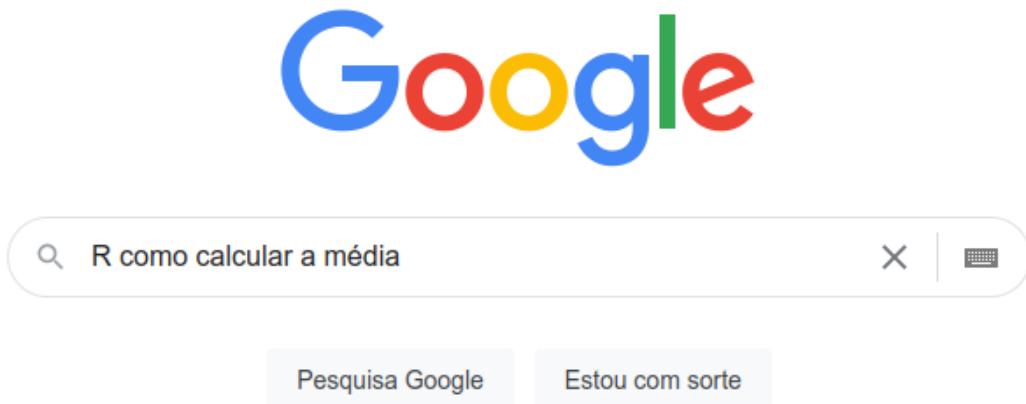
Observe que ao escrever o comando `help(sqrt)`, irá abrir uma janela na aba de ajuda (Help) no RStudio. Com o tempo, você irá se familiarizar com os diferentes termos e conceitos do R e conseguirá utilizar a documentação de cada função com facilidade. Muitas vezes, erros e problemas poderão ser solucionados por um entendimento da documentação no próprio R.

Uma forma muito útil para buscar ajuda ou retirar dúvida do R é digitando os erros que você visualiza no *output* lá nas plataformas de ajuda: *Google* ou *Stackoverflow*. A linguagem R é amplamente difundida pelo mundo. Muitas pessoas a utilizam e, frequentemente, se deparam com erros que você pode se deparar também. Dificilmente, iremos nos deparar com uma dúvida nunca respondida.

### b) O Google

Veja na Figura 20 como buscamos uma dúvida no *Google*. Uma dica seria iniciar colocando a letra R seguida de palavras chaves que remetem à questão:

**Figura 20: Tela de busca do Google.**

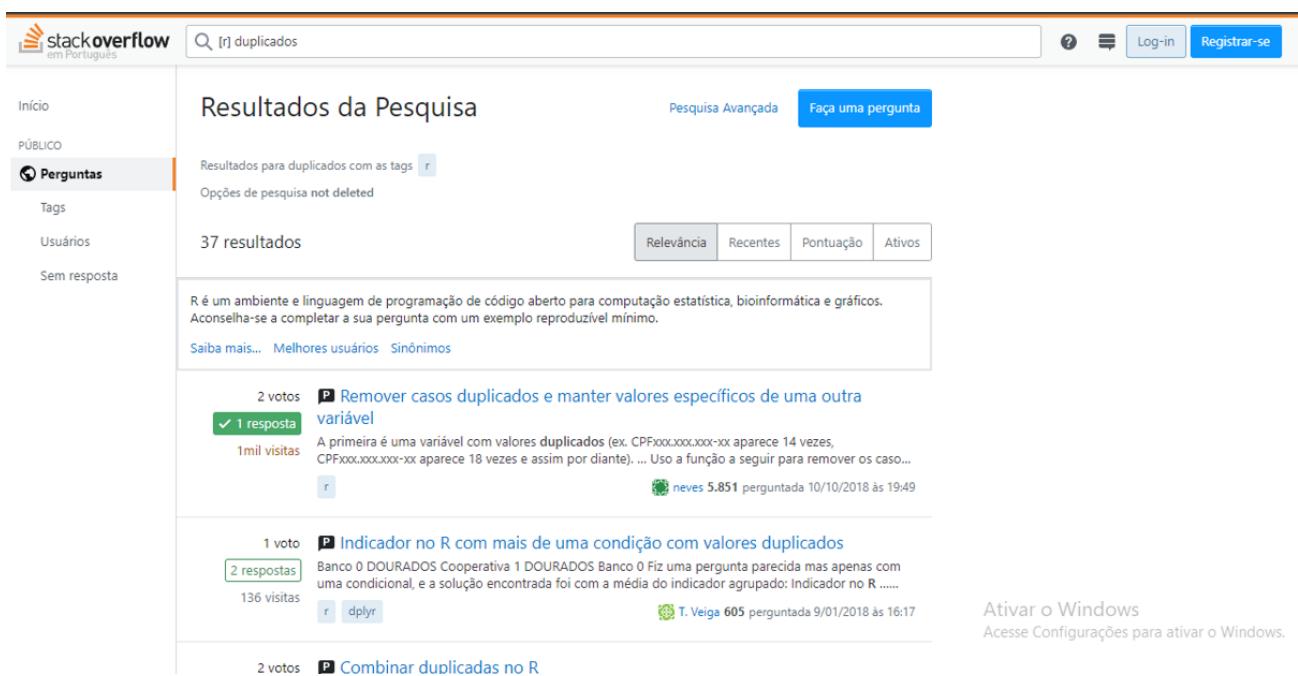


Também poderá colar no “buscador google” literalmente a mensagem de erro emitida pelo R, o que pode te apoiar a encontrar uma solução mais rápida.

### c) O Stackoverflow

Veja na Figura 21 como buscamos uma dúvida no *Stackoverflow*, um site gratuito de perguntas e respostas sobre programação com perguntas e respostas feitas por pessoas que estão escrevendo códigos. Para utilizá-lo, assim como no *Google* você pode iniciar colocando a letra R seguida de palavras chaves ou a própria questão:

**Figura 21: Tela de busca do Stackoverflow.**



The screenshot shows the Stackoverflow homepage with a search bar containing '[r] duplicados'. The results page is titled 'Resultados da Pesquisa' and displays 37 results. The first result is a question titled 'Remover casos duplicados e manter valores específicos de uma outra variável' with one answer. The second result is 'Indicador no R com mais de uma condição com valores duplicados' with two answers. The third result is 'Combinar duplicadas no R' with two votes. The sidebar on the left shows navigation links like 'Início', 'PÚBLICO', and 'Perguntas'.

Você também poderá colar no “buscador” do *Stackoverflow* literalmente a mensagem de erro emitida pelo R, o que pode te apoiar a encontrar uma solução mais rápida.

## Materiais de apoio

Caso queira se aprofundar no conteúdo, recomendamos os seguintes materiais complementares:



- <https://www.rdocumentation.org>
- <https://www.rstudio.com/resources/cheatsheets/>
- <https://livro.curso-r.com/1-instalacao.html>
- <https://appliedepi.org/tutorial/#data-preparation>
- <https://bookdown.org/rdpeng/rprogdatascience/>
- <https://listas.inf.ufpr.br/cgi-bin/mailman/listinfo/r-br>
- <https://cran.r-project.org/other-docs.html#nenglish>
- <https://r4ds.had.co.nz/>



## Próximo módulo

Pronto chegamos ao final do nosso módulo! Agora você já conhece todas as funções básicas para uso da linguagem R. Acesse os demais módulos deste curso para colocar em prática as análises de dados necessárias para estabelecer rotinas de trabalho na vigilância em saúde.

Até ao próximo módulo!

