

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE CIÊNCIAS DA SAÚDE
PROGRAMA DE PÓS-GRADUAÇÃO EM RESIDÊNCIA
MULTIPROFISSIONAL INTEGRADA EM SISTEMA PÚBLICO DE
SAÚDE

Layza de Souza Gonçalves

RStudio na Vigilância em Saúde

Santa Maria, RS
2023

Layza de Souza Gonçalves

Rstudio na Vigilância em Saúde

Monografia de Conclusão de Curso apresentado ao Programa de Pós-Graduação em Residência Multiprofissional Integrada em Sistema Público de Saúde, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do título de **Especialista em Sistema Público de Saúde, Área de Concentração: Vigilância em Saúde.**

Orientadora: Prof.^a Dr.^a Lisiane Bôer Possa

Santa Maria, RS

2023

Layza de Souza Gonçalves

RStudio na Vigilância em Saúde

Monografia de Conclusão de Curso apresentado ao Programa de Pós-Graduação em Residência Multiprofissional Integrada em Sistema Público de Saúde, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do título de **Especialista em Sistema Público de Saúde, Área de Concentração: Vigilância em Saúde.**

Aprovado em 30 de abril de 2023:

**Lisiane Bôer Possa, Dra. (UFSM)
(Presidente/Orientadora)**

Fabiele Feltrin Paniz, Esp. (4ª CRS)

Paula Wesz Keller, Esp. (4ª CRS)

Santa Maria, RS
2023

RESUMO

RStudio na Vigilância em Saúde

AUTOR(A): Layza de Souza Gonçalves

ORIENTADOR(A): Lisiane Bôer Possa

Introdução: A Vigilância em Saúde historicamente se pautava em policiamento e controle de doenças e endemias, com o advento de pensamentos progressistas associados ao Sistema Único de Saúde (SUS) está se reformulou pautada na prevenção, proteção e promoção da saúde. Assim como, as tecnologias que auxiliam nessas novas ideologias, em que a análise situacional de saúde é imprescindível para melhor atender as organizações socio-econômicas-sociais-ambientais que a população está inserida para colaborar nas diferentes redes de saúde, qualificar dados seja da morbimortalidade à endemias e pandemias, estratificar perfis mais propensos a determinadas vacinas e imunobiológicos, por meio principalmente, dos serviços Vigilância Epidemiológica e Imunizações, sendo o software RStudio um programa acessível, intuitivo e claro para esses fins. **Objetivo:** Elaborar mini manual básico para uso do RStudio na Vigilância em Saúde. **Conclusão:** o referido estudo contribuirá com a disponibilidade de informações sistematizadas para o planejamento de ações voltadas a Vigilância em Saúde.

Palavras-chave: RStudio. Vigilância em Saúde. Vigilância Epidemiológica e Imunizações.

ABSTRACT

RStudio in Health Surveillance

AUTHOR: Layza de Souza Gonçalves

ADVISOR: Lisiane Bôer Possa

Introduction: Health Surveillance was historically based on policing and control of diseases and endemic diseases, with the advent of progressive thoughts associated with the Unified Health System (SUS) it has been reformulated based on prevention, protection and health promotion. As well as the technologies that help in these new ideologies, in which the situational analysis of health is essential to better serve the socio-economic-social-environments organizations in which the population is inserted to collaborate in the different health networks, qualify data whether from endemic and pandemic morbidity and mortality, stratify profiles more prone to certain vaccines and immunobiologicals, mainly through the Epidemiological Surveillance and Immunizations services, RStudio software being an accessible, intuitive and clear program for these purposes. **Objective:** To elaborate a basic mini manual for the use of RStudio in Health Surveillance. **Conclusion:** this study will contribute to the availability of systematized information for planning actions aimed at Health Surveillance.

Keywords: RStudio. health surveillance. epidemiological surveillance and immunizations

Sumário

1. INTRODUÇÃO	7
2. JUSTIFICATIVA	8
3. OBJETIVO	9
3.1. OBJETIVO GERAL	9
3.2. OBJETIVO ESPECÍFICO	9
4. REVISÃO DE LITERATURA.....	Erro! Indicador não definido.
4.1. VIGILÂNCIA EM SAÚDE.....	10
4.2. RSTUDIO	12
5. CONCLUSÃO.....	20
6. APÊNDICE	22
6.1. TABELA RESUMO DOS COMANDOS DO RSTUDIO.....	22
7. BIBLIOGRAFIA.....	32

1. INTRODUÇÃO

Vigilância define-se em detectar, analisar e difundir informações sobre saúde-doença de importância socio-demográfica-econômica. A partir da coleta sistemática, qualificação e avaliação de dados desde o nascimento, passando pelo desenvolvimento e ambiente que se está inserido, saúde e adoecimento até a morte de sua população, para que assim se possa disseminar informações e colaborar na construção de conhecimentos e mudanças sociais salutares. Outro ponto importante da Vigilância em Saúde (VS) refere-se a sua transversalidade de atuação, a qual articula e integra ações com o intuito de compreender não só a população, mas ela inserida em seu próprio território, sem deixar de associar-se aos fluxos do sistema de serviço de saúde. Nesse sentido, a VS segue os preceitos básicos do SUS por meio da prevenção, promoção e proteção à saúde ao intervir sobre problemas de saúde por meio de seus sistemas de informação, articulação de ações inter-setoriais diretamente no território ou indiretamente com suporte a quem está na ponta na tomada de decisões, planejamento e avaliação permanente das intervenções realizadas sobre necessidades locais de saúde e assim resultando nas Políticas Públicas. (GUIMARÃES et al., 2017; TEIXEIRA et al., 2018)

De encontro ao exposto, o uso de tecnologias (leves, leves-duras e duras) vem para facilitar esses processos de trabalho. E na Vigilância em Saúde, mais especificamente na Vigilância Epidemiológica e Imunização a apropriação de tecnologias duras (maquinário, programas, softwares, vídeos...) são imprescindíveis e auxiliares. Muitas dessas precisam ser desenvolvidas e/ou melhoradas, porém alguns já estão disponíveis e precisam ser mais divulgadas e difundidas para melhor contribuir nesses serviços. Como exemplo o programa RStudio, o qual dentre suas várias funções (análise de dados e qualificação) vai de encontro a uma Vigilância ativa, dinâmica e acessível (BARRA, 2006). Além disso, após vivência e observações, durante a Residente em Vigilância em Saúde, do *modus operandi* no setor de Epidemiologia a nível Estadual e Municipal levantou-se a necessidade de uma melhor qualificação dos dados e otimização do tempo laboral seja para divulgar dados para os diferentes níveis da rede seja para a população em geral. Logo, questionou-se será que muitas das necessidades poderiam ser supridas com o conhecimento básico desse programa e como usar ele? (WICKHAM, 2019; BARRA, 2006)

2. JUSTIFICATIVA

O grande aumento de dados e complexidade se tornou realidade na rotina dos serviços de Vigilância em Saúde. E há diversidade de softwares disponíveis, para apoiar as análises, como o Stata, o SPSS e o Epi Info. Porém nem todos têm ferramentas suficientes para processar e/ ou integrar diferentes bancos de dados deixando a desejar nas demandas desses serviços. Como a interrupção da leitura de banco de dados memória insuficiente, ou análises são extremamente lentas, trabalhosas e repetitivas. Com isso, é imprescindível o conhecimento e uso de softwares de forma estruturada, segura, com precisão e transparência, programa o qual mais se aproxima dessa definição expressa-se no RStudio e justifica a elaboração deste presente trabalho.

3. OBJETIVO

3.1. OBJETIVO GERAL

Elaborar mini manual básico para uso do RStudio na Vigilância em Saúde

3.2. OBJETIVO ESPECÍFICO

- Produzir uma primeira síntese sistematizada, produzir roteiro, sumarizar, validar o manual com equipe e produzir manual final;
- Otimizar análise dados de interesse da Vigilância Epidemiológica e Imunizações;
- Otimizar os processos de trabalho no setor de Vigilância Epidemiologia e Imunizações de uma Coordenadoria Regional de Saúde.

4. METODOLOGIA

4.1. REVISÃO DE LITERATURA

4.1.1. VIGILÂNCIA EM SAÚDE

A historicidade da Vigilância em Saúde remonta medidas pautadas no policiamento, fiscalização e prevenção de epidemias, em que muitas vezes eram impositivas e opressivas contra, principalmente, aos mais vulneráveis. Uma das consequências de tanta opressão sem explicação associado a exclusão social foi a Revolta da Vacina, 1904. Por conseguinte, só em 1968 um sistema de notificação regular para monitoramento da situação epidemiológica instaurou-se, sem, contudo, seguir os ideários ideológico do SUS (promoção, proteção e prevenção da saúde-doença) que começaria seus primeiros passos nas décadas seguintes. Em 1975 o Sistema Nacional de Vigilância Epidemiológica foi implantado e no ano posterior com status mais sociopolítico ganhou força com a Secretaria Nacional de Vigilância Sanitária. Com o advento do SUS, nos anos 90, reestruturou-se a Vigilância com a Secretaria de Vigilância em Saúde, um Sistema Nacional de Vigilância Sanitária e uma área técnica de Vigilância em Saúde Ambiental. No entanto, sua historicidade marcante ainda se fazia muito presente assim como o empenho de se desvencilhar dessas convenções e seguir os ideários do SUS. (GUIMARÃES, 2017).

Nesse caminho em 2004 e 2009 com complemento em 2013, respectivamente, promulgou-se as Portarias GM/MS nº 1.172 para descentralização das ações de Vigilância em Saúde; GM/MS nº 3.252/09 e GM/MS nº 1.378, para o financiamento das ações de Vigilância em Saúde pela União, Estados, Distrito Federal e Municípios e estabelece a Vigilância em Saúde como análise permanente da situação de saúde da população. E finalmente, em 2018, instaura-se a Política Nacional de Vigilância em Saúde (PNVS,) a qual respalda as ações e serviços de saúde da Vigilância em Saúde. A PNVS caracteriza e respalda-se nos princípios do SUS os quais visam universalidade, transversalidade, integralidade e equidade, sendo a administração de responsabilidade do poder público. Além disso, conceitua e disponibiliza ferramentas corroborativas para análise de dados, análise situacional de saúde, políticas de planejamento e saúde, análise epidemiológica dos processos saúde-doença e laboral. Por meio da política houve uma definição divisória para melhor atender as demandas

da vigilância, porém sempre com o intuito de agregar todas nas demandas que surgem, são elas: Vigilância Epidemiológica (VE), a qual inclui as Imunizações; Vigilância Ambiental (VAM), Vigilância em Saúde do Trabalhador (VST) e Vigilância Sanitária (VISA) (BRASIL, 2018).

A VE tem como foco identificar, notificar, investigar e controlar doenças e agravos de interesse de saúde. As suas ações pautam-se na análise e identificação de variáveis para conhecimento e detecção de fatores determinantes e condicionantes da saúde. Inclusive prevenção, detecção e controle de surtos, endemias, epidemias ou pandemias. A VAM atua no âmbito do conhecimento e a detecção de mudanças do meio ambiente que possam interferir na saúde humana e assim adotar medidas de promoção à saúde, prevenção e monitoramento dos fatores de riscos relacionados às doenças ou agravos à saúde. Como por exemplo controle de resíduos e de vetores de transmissão de doenças (insetos e roedores. Já a VST desenvolve de ações que visam a promoção da saúde, prevenção e promoção da morbimortalidade, redução de riscos e/ou agravos e vulnerabilidades no ambiente laboral ou decorrentes dos processos produtivos. A VISA é responsável por medidas para eliminar, diminuir ou prevenir riscos à saúde ao intervir nos problemas sanitários do ambiente (produção e circulação de bens e da prestação de serviços do interesse da saúde). Como por exemplo, fiscalizam serviços de: escolas, hospitais, clubes, academias, parques e centros comerciais (BRASIL, 2018)

Levando-se em conta isto a análise de dados está intrinsecamente interligada as práticas básicas na vigilância em saúde. Ou seja, ter um diagnóstico de saúde populacional ao conhecer e acompanhar o estado de saúde são ações imprescindíveis. E isso é possível mediante análise de situação de saúde de uma região ao conhecer perfis, padrões e tendências e assim fomentar ações de proteção, promoção da saúde, prevenção e controle de doenças e agravos à saúde da população. Esta análise refere-se a um trabalho descritivo e comparativo de dados de diferentes origens, datas e locais, para caracterizar um determinado tempo, pessoa e lugar. Dados os quais estão presentes no cotidiano da VS e 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 vários outros. (WICKHAM,2019; SANTA, CATARINA 2022)

Nessa perspectiva, a Vigilância em Saúde é uma parte essencial de qualquer sistema de saúde. Esta é uma área complexa e que requer que seus profissionais atuantes sejam capacitados de forma a compreender o impacto da detecção e prevenção dos processos de saúde e doença. Além de procurarem interligar as ações entre as vigilâncias e, principalmente, a necessidade de qualificação dos dados obtidos nos trabalhos de campo ou pelos sistemas de informação para uma devolutiva coesa e clara para os diferentes níveis das redes de atenção tal como para a população envolvida. Logo, o uso de ferramentas tecnológicas e a apropriação delas pelos profissionais atuantes nessas áreas são de suma importância, como disse Pinto et al. (2017) e Rocha et al (2017) há necessidade de educação permanente e capacitações constantes para que se possa ter ‘novas perspectivas’ na realidade já existente.

4.1.2. RSTUDIO

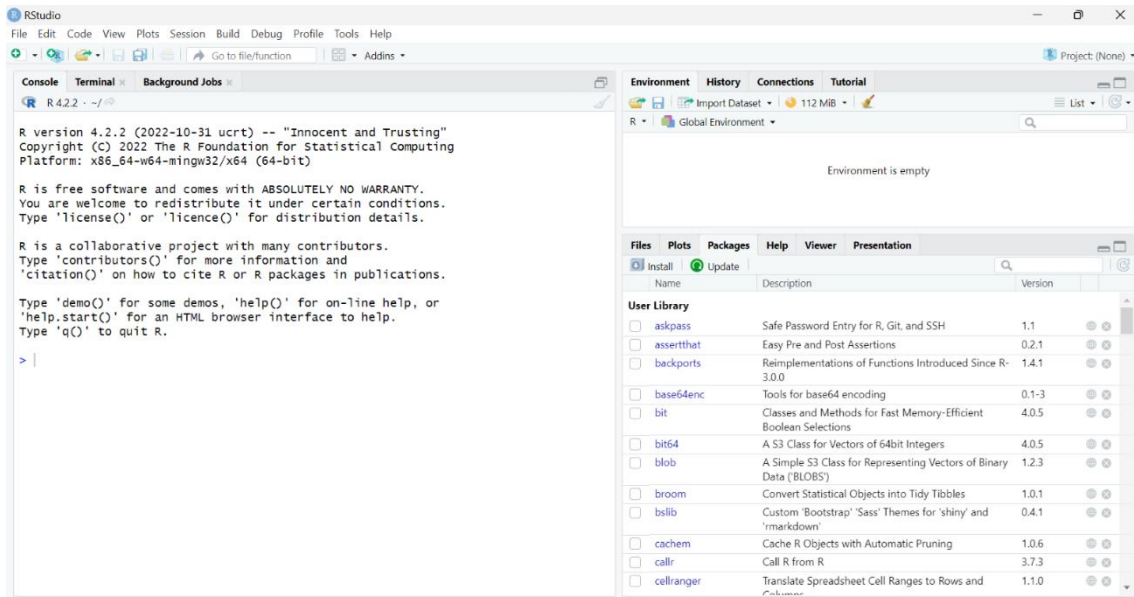
O R é uma linguagem de programação, em que se podem gerar análise de dados, gráficos, planilhas, qualificação de dados dentre outras funções de interesse estatístico, epidemiológico e toda e qualquer área que necessite de suas finalidades. Outro ponto extremamente relevante é sua natureza gratuita de uso. Não apenas por isso, mas tem-se notado o crescimento de sua utilização nos últimos anos, principalmente durante a pandemia, em que a Vigilância em Saúde teve seu papel ímpar, em que se precisou de um processo dinâmico para dados e projeções com o intuito de dirimir danos salutar e sociais, assim como corroborar com a saúde pública. Ao passo que este programa se pode escrever scripts (roteiros pré-definidos e/ou códigos) em que o computador irá reconhecer e executar a tarefa solicitada em minutos. Como por exemplo projeto de pós-graduação da Universidade de Brasília que utilizou a linguagem R para otimizar o encerramento de notificações de casos de dengue (MUSSO, 2022) Além disso, pode ser usado em vários tipos de sistemas operacionais (Windows, Unix, Mac OS, Linux) e também há uma grande comunidade de pesquisadores e simpatizantes do programa que desenvolvem novas extensões para análises de dados e disponibilizam gratuitamente online, como o fórum *Stackoverflow* (<https://pt.stackoverflow.com>) (WICKHAM,2019; SANTA, CATARINA 2022).

Outro ponto relevante diz respeito a reprodutibilidade em diferentes tipos de computadores, ou seja, a não necessidade de um aparelho de última geração para rodar ele. Além disso, pode-se conciliar base de dados (das simples as grandes que não rodam em outros softwares); outras linguagens de programação, plataformas e programas. Assim como tarefas cotidianas podem ser automatizadas, reduzindo o tempo e a repetição delas. Inclusive diferentes formatos de arquivos podem ser gerados (PDF, Word, LaTeX e HTML) para relatórios e dashboards (ferramenta de gestão da informação), sites e gráficos interativos automatizados (WICKHAM,2019; SANTA, CATARINA 2022). Como um trabalho feito em Brasília pelo autor em que foi utilizado a linguagem R para geolocalizar internações cadastradas no Sistema de Informações Hospitalares do SUS (ROCHA et al, 2018).

No R assim como na estatística há uma padronização de conceitos básicos importantes, como variável refere-se a coluna, dentre elas podem existir do tipo quantitativa (elementos mensuráveis). Em que se subdividem em discretas, números inteiros, o R reconhece como “*numeric* ou *integer*” e em contínuas, números decimais, o R reconhece com “*numeric* ou *double*”. E qualitativas (elementos não mensuráveis). Que também se subdividem em nominais, texto, o R lê como “*character*”, exemplo país; e em ordinais, categorias, o R lê como “*factor*”, exemplo sexo, e os “*levels*” são seus diferentes níveis (WICKHAM,2019; SANTA, CATARINA 2022).

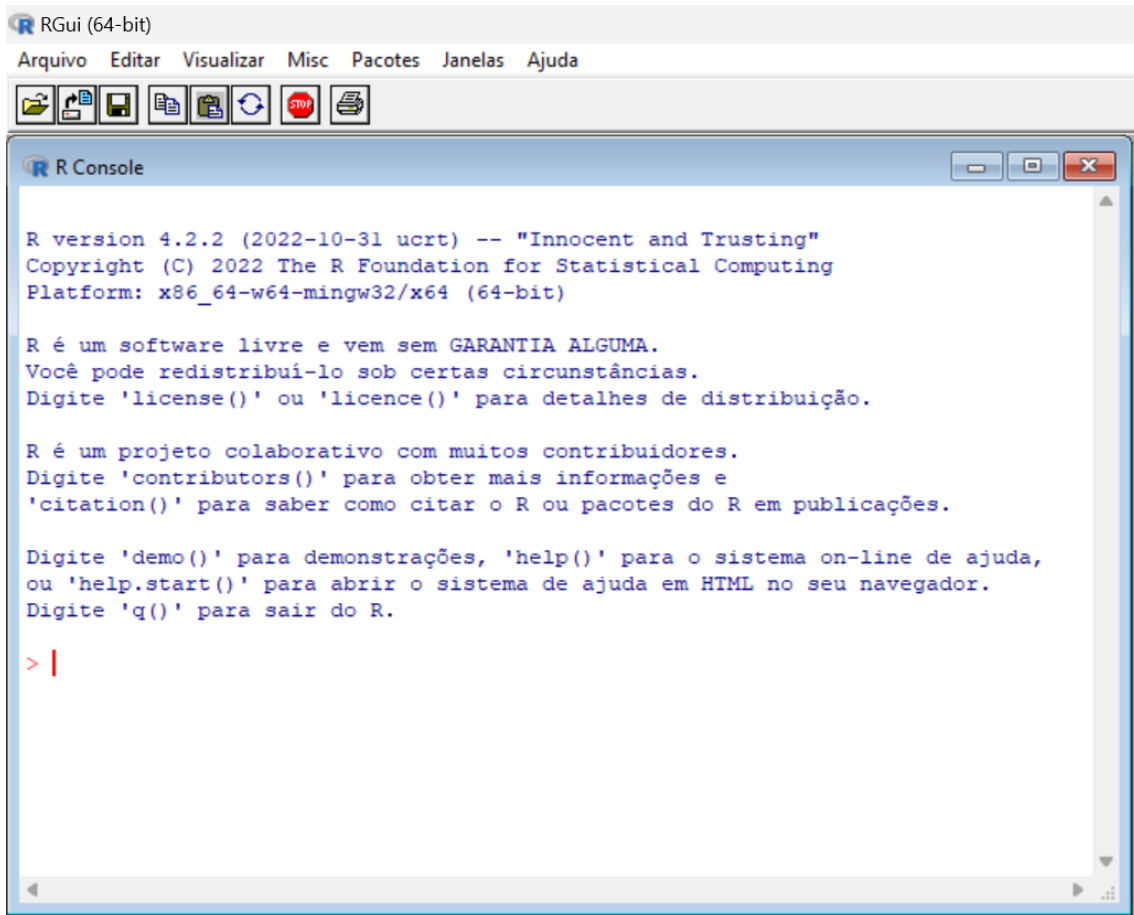
O R basicamente compõe-se por dois programas o “R” e o “RStudio”. O “RStudio” ou IDE (Integrated Development Environment) seriam os outros órgãos em que há diferentes funções e passos para se chegar a um resultado, ou seja, nesse software a diferentes comandos e uma interface interativa de fácil visualização que resultam nos produtos já descritos anteriormente. Enquanto o “R” fazendo-se uma analogia fisiológica pode-se dizer que é o cérebro do programa, é ele vai configurar o computador para interpretar e executar os comandos. Embora pareça trabalhos independentes um depende do outro como uma equipe. Abaixo seguem imagens desses painéis (WICKHAM,2019; SANTA, CATARINA 2022),

Figuria 1: RStudio ou IDE



(Fonte: própria autora)

Figura 2: R



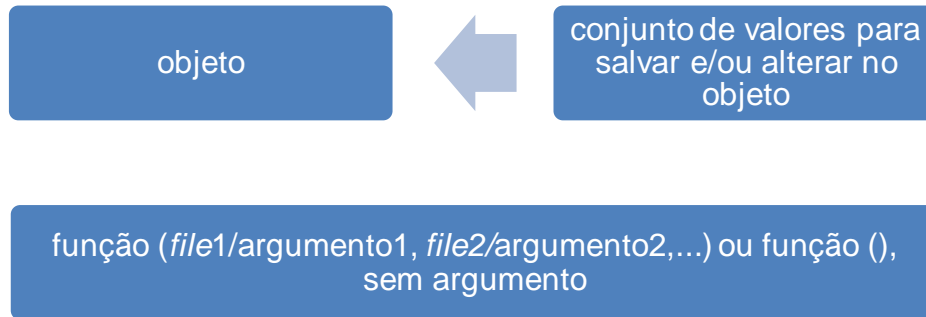
(Fonte: própria autora)

O painel RStudio possui 3 subdivisões, sendo o “Console” como se fosse um controle remoto em que é dado os comandos que executará em uma ação. Já o “Environment” armazena tudo o que é feito no R (criações e importações), inclusive o painel “History”, ou seja, histórico está nele. Enquanto o “Files” apareci os arquivos da pasta que se está utilizando, dentro deste painel há o “Plots” em que aparece os gráficos gerados, o “Packages” são os pacotes que auxiliam os comandos do R e o “Help” para ajuda e documentação (WICKHAM,2019; SANTA, CATARINA 2022).

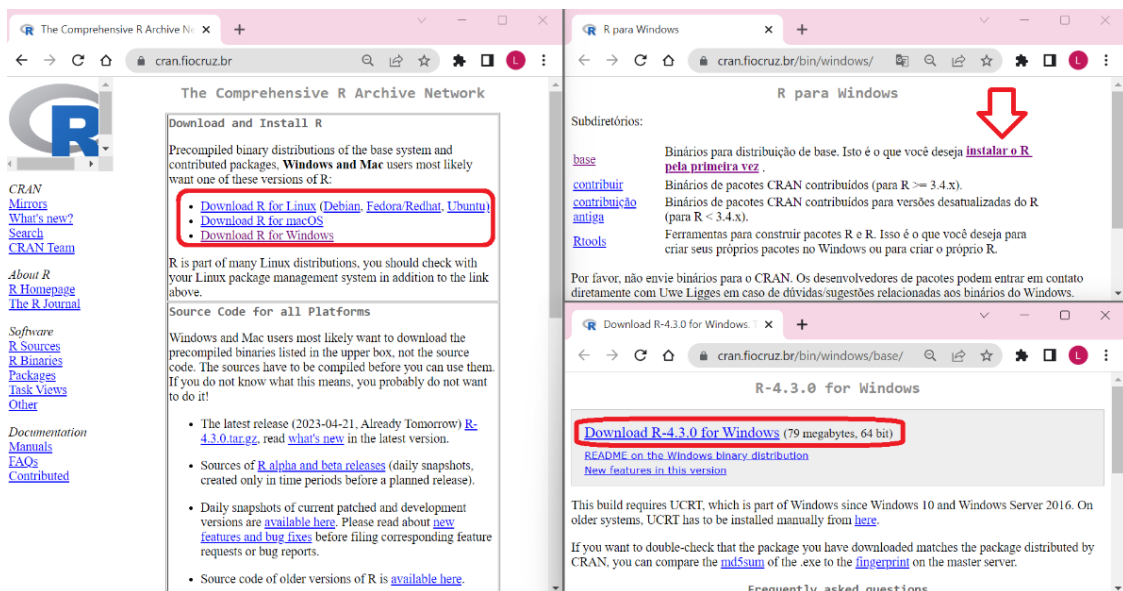
No painel “Console” geralmente terá no início da linha de comando o símbolo de maior que (>), a partir dele escrevem-se os códigos, clica-se em “enter” e será executado uma ação e/ou resultado. Entretanto ele não deixa salva o que é digitado, para isso é preciso um roteiro por meio de códigos pré-estabelecidos para o trabalho diário com instruções individuais em arquivos de modo coeso, bem explicado passo a passo, sem acentuação e caracteres especiais (File -> New File -> R Script) e posteriormente salvá-los (WICKHAM,2019; SANTA, CATARINA 2022).

Em contrapartida, o package já vem com alguns pacotes básicos com o programa R, mas também pode-se instalar outros (online ou arquivos locais) para várias finalidades, as quais corroborarão para análises de dados epidemiológicas. Para os básicos ao clicar na caixa ao lado do nome do pacote irá selecioná-lo e para novos pacotes clica-se em “Install” na caixa “Install from” selecione “Repository (CRAN)” na outra caixa digite o nome do pacote para instalações online. Ou usa-se no painel “Console” o comando “*install.packages*(“ ”)” com o nome do pacote entre as aspas. Após instalado o pacote fica salvo no seu repositório e basta digitar o comando “*library*(“ ”)” com o nome do pacote desejado que ele será executado em sua análise. Toda vez que se fecha o RStudio e o abre deve-se carregar os pacotes que se utilizará. Para pacotes desatualizados (warning) (Package -> Update -> Select All -> Instal Updates) (WICKHAM,2019; SANTA, CATARINA 2022).

Outro ponto importante, para uma vida útil do computador que se usará o R é desabilitar o salvamento automático dos *scripts* para poupar memória dele (*Tools* -> *Global Options* -> *Aba Geral* -> desabilitar: os tópicos do *Workplace* e o primeiro do *History*). Para o R tudo é armazenada na forma de um objeto (valores, conjunto de valores, base de dados etc.). E para se alterar um objeto usa-se <- (“menor que” e “hífen”). Enquanto, para se escrever uma função segue-se os mesmos parâmetros como do Excel (WICKHAM,2019; SANTA, CATARINA 2022).

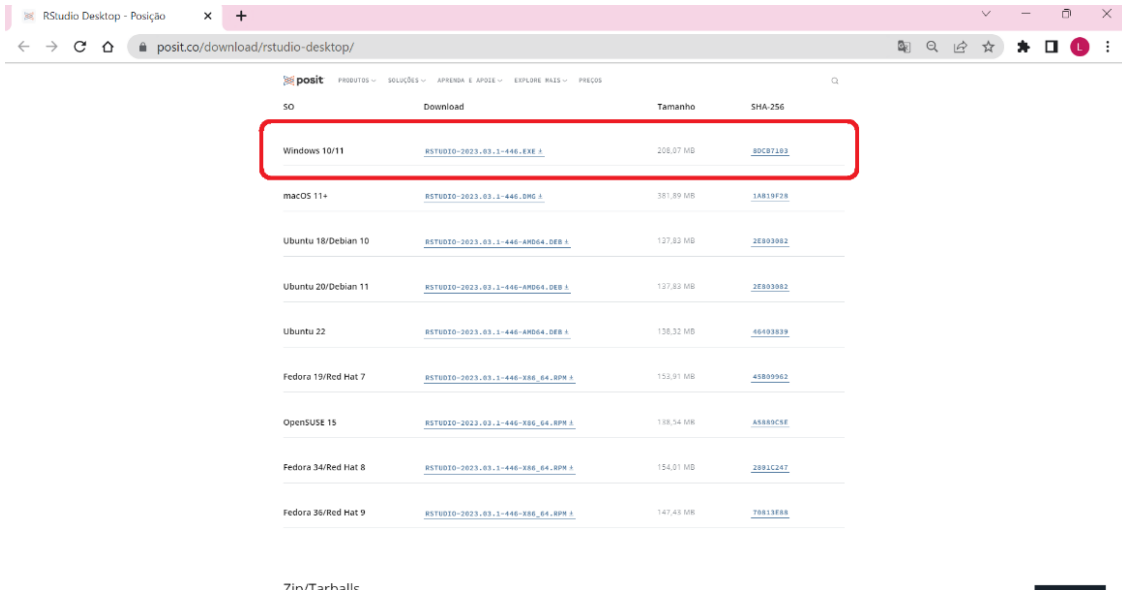


Nessa perspectiva, para instalar este programa inicia-se pelo *software R* a partir do link <https://cran.fiocruz.br/> deve-se escolher o tipo de sistema do computador, conforme figuras abaixo:



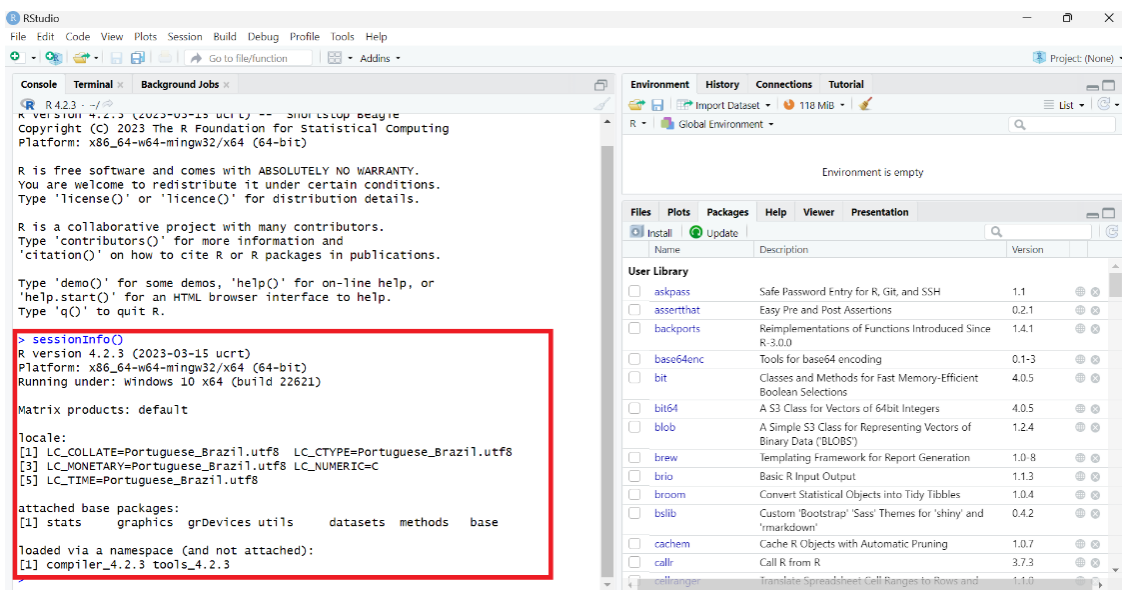
FONTE: print da tela da própria autora

Após deve-se baixar o RStudio por meio do link <https://www.rstudio.com/products/rstudio/download/> e escolhera qual é compatível com sistema operacional do computador, conforme exemplo abaixo:



FONTE: print da tela da própria autora

Por conseguinte, serão apresentados alguns pacotes e funções para o uso do R em análises de dados. Começa-se com pacotes básicos que apenas digitados no console ele será executado



FONTE: print da tela da própria autora

```

Source
Console Terminal Background Jobs
R 4.2.3 - /
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] compiler_4.2.3 tools_4.2.3
> install.packages("epitools")
WARNING: Rtools is required to build R packages but is not currently installed. Pl
ease download and install the appropriate version of Rtools before proceeding:
https://cran.rstudio.com/bin/windows/Rtools/
Installing package into 'C:/Users/layza/AppData/Local/R/win-library/4.2'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/4.2/epitools_0.5-10.1.zi
p'
Content type 'application/zip' length 318567 bytes (311 KB)
downloaded 311 KB

package 'epitools' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
C:\Users\layza\AppData\Local\Temp\RtmpQ53Eu9\downloaded_packages
> library("epitools")

```

FONTE: print da tela da própria autora

Para importar dados a partir de base de dados baixadas usa-se por exemplo um arquivo do tipo .CSV o pacote *readr()* com o comando *read_csv2()*, enquanto que para arquivos tipo .dbf usa-se o pacote *foreign()* com o comando *read.dbf()* e, por fim, para arquivos tipo .xlsx usa-se o pacote *readxl()* e o comando *read.excel()*. Sempre usa-se o comando *getwd()* para saber corretamente onde estão os arquivos, os quais serão importados, para o código digitado no R não dar erro e ter o risco de não localizar o arquivo.

```

Project - RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
C:\R\Project\GRYPE
GRYPE
COD_UNID NM_UNID
1 2376598 A B A
2 2077531 A C CAMARGO CANCER CENTER
3 9735526 A CASA HUMANA MEDICINA CLINICA E PREVENTIVA
4 5185203 A MEDICINA DIAGNOSTICA CAMPO GRANDE
5 2557185 AAHTS
Showing 1 to 6 of 9,839 entries, 2 total columns

Console Terminal Background Jobs
R 4.2.3 - C:/R/Project/
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.

> library(epitools)
> library(useethis)
> library(foreign)
> library(foreign, lib.loc = "C:/Program Files/R/R-4.2.3/libray")
> library(readr)
> library(readxl)
> GRYPE <- read.dbf(file = "C:/R/Project/SENTINELA1907212_00.dbf")

```

FONTE: print da tela da própria autora

Após criar a tabela pode-se editar os dados conforme a demanda e prioridade de informações. As notificações para agilizar o preenchimento muitos dos campos são

números que correspondem a situações e vice-versa, com isso para melhor visualização e entendimento pode-se redefinir ou selecionar esses caracteres e/ou colunas e linhas. Abaixo segue alguns exemplos. É preciso instalar os seguintes pacotes: *foreign*, *readxl*, *readr*, *janitor*, *skimr*, *stringr*, *stringi*, *lubridate*, *summarytools* e *tidyverse* e *descr*.

- Criar tabela com colunas selecionadas:

The screenshot shows the RStudio interface. The Environment pane displays two data objects: 'sivep' (18 obs. of 192 variables) and 'sivep_novo' (18 obs. of 4 variables). The Data viewer shows a table with 5 rows and 4 columns: NU_NOTIFIC, CS_SEXO, CS_RACA, and CS_ESCOL_N. The Console window shows the following R code and output:

```
Warning message:
In library(package, lib.loc = lib.loc, character.only = TRUE, logical.return = TRUE, :
there is no package called 'summarytools'
> library(tidyverse)
— Attaching core tidyverse packages —
✓ dplyr 1.1.2      ✓ purrr 1.0.1
✓ forcats 1.0.0   ✓ tibble 3.2.1
✓ ggplot2 3.4.2   ✓ tidyr 1.3.0
— Conflicts — tidyverse_conflicts() —
✗ dplyr::desc() masks desc::desc()
✗ dplyr::filter() masks stats::filter()
✗ dplyr::lag() masks stats::lag()
✗ tibble::view() masks summarytools::view()
i Use the conflicted package to force all conflicts to become errors
> View(sivep)
> sivep_novo <- select(sivep, NU_NOTIFIC, CS_SEXO, CS_RACA, CS_ESCOL_N)
> View(sivep_novo)
```

FONTE: print da tela da própria autora

- Criar coluna correspondente ao significado da coluna de números da notificação:

The screenshot shows the RStudio interface. The Environment pane displays two data objects: 'sivep' (18 obs. of 192 variables) and 'sivep_novo' (18 obs. of 5 variables). The Data viewer shows a table with 5 rows and 5 columns: NU_NOTIFIC, CS_SEXO, CS_RACA, CS_ESCOL_N, and RACA. The RACA column contains categorical values: BRANCO, BRANCO, BRANCO, AMARELO, and IGNORADO. The Console window shows the following R code and output:

```
Warning message:
there is no package called 'summarytools'
> library(tidyverse)
— Attaching core tidyverse packages —
✓ dplyr 1.1.2      ✓ purrr 1.0.1
✓ forcats 1.0.0   ✓ tibble 3.2.1
✓ ggplot2 3.4.2   ✓ tidyr 1.3.0
— Conflicts — tidyverse_conflicts() —
✗ dplyr::desc() masks desc::desc()
✗ dplyr::filter() masks stats::filter()
✗ dplyr::lag() masks stats::lag()
✗ tibble::view() masks summarytools::view()
i Use the conflicted package to force all conflicts to become errors
> View(sivep)
> sivep_novo <- select(sivep, NU_NOTIFIC, CS_SEXO, CS_RACA, CS_ESCOL_N)
> View(sivep_novo)
> sivep_novo$RACA <- factor(x = sivep_novo$CS_RACA, levels = c("1", "2", "3", "4", "9"),
labels = c("BRANCO", "PARDO", "PRETO", "AMARELO", "IGNORADO"))
> View(sivep_novo)
```

FONTE: print da tela da própria autora

- Excluir coluna:

The screenshot shows the RStudio interface. The console displays the following R code and its output:

```

> View(sivep)
> sivep_novo <- select(sivep, NU_NOTIFIC, CS_SEXO, CS_RACA, CS_ESCOL_N)
> View(sivep_novo)
> sivep_novo$RACA <- factor(x = sivep_novo$CS_RACA, levels = c("1", "2", "3", "4", "9"),
  labels = c("BRANCO", "PARDO", "PRETO", "AMARELO", "IGNORADO"))
> sivep_novo |> select(-CS_SEXO)
# A tibble: 18 x 4
  NU_NOTIFIC CS_RACA CS_ESCOL_N RACA
  <dbl> <chr> <dbl> <fct>
1 10039876 1 9 BRANCO
2 10042161 1 9 BRANCO
3 10043051 1 9 BRANCO
4 10045529 4 NA AMARE...
5 10046813 9 NA IGNOR...
6 10047778 9 9 IGNOR...
7 10048099 1 1 BRANCO
8 10053538 1 9 BRANCO
9 10057136 1 9 BRANCO
10 10059521 4 NA AMARE...
11 10059757 1 9 BRANCO
12 10064472 4 9 AMARE...
13 10078095 1 4 BRANCO
14 10081213 1 3 BRANCO
15 10081687 9 NA IGNOR...
16 10083244 1 NA BRANCO
17 10083404 1 NA BRANCO
18 10084031 1 3 BRANCO

```

The Environment pane on the right shows the following data objects:

- sivep: 18 obs. of 192 variables
- sivep_novo: 18 obs. of 5 variables

The Files pane shows the following files:

- DENGON1907221_00.dbf: 148.1 KB, May 16, 2023, 10:01 PM
- Project.Rproj: 218 B, May 17, 2023, 6:10 PM
- SENTINELA1907212_00.dbf: 653.5 KB, May 16, 2023, 10:40 PM
- sivep_gripe.xlsx: 26.9 KB, May 17, 2023, 5:44 PM

FONTE: print da tela da própria autora

- Inserir nova coluna:

The screenshot shows the RStudio interface. The console displays the following R code and its output:

```

> library(tidyverse)
— Attaching core tidyverse packages — tidyverse 2.0.0 —
✓ forcats 1.0.0 ✓ tibble 3.2.1
✓ ggplot2 3.4.2 ✓ tidyr 1.3.0
✓ purrr 1.0.1
— Conflicts — tidyverse_conflicts() —
x dplyr::desc() masks desc::desc()
x dplyr::filter() masks stats::filter()
x dplyr::lag() masks stats::lag()
x tibble::view() masks summarytools::view()
i Use the conflicted package to force all conflicts to become errors
> GRIPE <- read.dbf(file = "C:/R/Project/MINDINET.dbf", as.is = TRUE)
> gripe_novo <- GRIPE |> select(NU_NOTIFIC, ID_AGRAVO, DT_NOTIFIC,
  DT_DIGITA) |> mutate(TEMPO_DIGITA = DT_DIGITA - DT_NOTIFIC)
> View(gripe_novo)

```

The Environment pane on the right shows the following data objects:

- GRIPE: 27621 obs. of 62 variables
- gripe_novo: 27621 obs. of 5 variables

The Files pane shows the following files:

- DENGON1907221_00.dbf: 148.1 KB, May 16, 2023, 10:01 PM
- Project.Rproj: 218 B, May 17, 2023, 8:30 PM
- SENTINELA1907212_00.dbf: 653.5 KB, May 16, 2023, 10:40 PM
- sivep_gripe.xlsx: 26.9 KB, May 17, 2023, 5:44 PM
- MINDINET.dbf: 12.2 MB, May 17, 2023, 8:30 PM

The Data Viewer shows a table with the following columns: NU_NOTIFIC, ID_AGRAVO, DT_NOTIFIC, DT_DIGITA, and TEMPO_DIGITA. The TEMPO_DIGITA column is highlighted with a red box.

NU_NOTIFIC	ID_AGRAVO	DT_NOTIFIC	DT_DIGITA	TEMPO_DIGITA	
1	7671320	A509	2012-04-11	2012-11-09	212 days
2	0855803	W64	2010-09-17	2010-11-17	61 days
3	8454645	X58	2010-10-19	2011-03-14	146 days
4	3282723	A90	2008-04-14	2008-04-24	10 days
5	9799526	B19	2011-06-20	2011-09-14	86 days

FONTE: print da tela da própria autora

Estes foram alguns exemplos visuais do programa RStudio, logo para uma melhor padronização e otimização fez-se uma tabela genérica (apêndice) com alguns pacotes, suas funções, comandos e exemplos para o uso cotidiano laboral.

5. RESULTADOS

Espera-se que o estudo contribua para apontar a necessidade de qualificação dos dados recebidos pela Vigilância Epidemiológica e Imunizações relacionadas as demandas do serviço. Tendo em vista que a análise de dados é de suma importância por todos os profissionais da Vigilância em Saúde, o que torna primordial a reflexão acerca da criação de mecanismos que favoreçam essa prática.

Por fim, traz-se que, para o enfrentamento desses desafios, sejam planejadas e realizadas ações de educação continuada, para que haja o empoderamento desses profissionais em relação a essas informações e, assim, seja dada a devida visibilidade às ações de vigilância e das demais condições (doenças e agravos) que demandam um olhar diversificado dessa área.

6. APÊNDICE

6.1. TABELA RESUMO DOS COMANDOS DO RSTUDIO

Tabela elaborada pela autora com embasamento em WICKHAM (2019) e SANTA CATARINA (2022)

Pacote	Função	Finalidade do Pacote	Finalidade da Função	Exemplos Script
	<code>install.packages("pacote")</code>		Instalar pacotes online	
	<code>library("pacote")</code>		Carregar pacotes	
	<code>sessionInfo()</code>		Mostra os pacotes ativos e algumas informações resumo do programa <i>RStudio</i>	
	#		Gera linha para comentário e não um comando	
	<code>getwd()</code>		Mostra local padrão que salva documentos/pastas do R	
<code>usethis()</code>	<code>create_project()</code>	Criar "documento" no local padrão do R	Função do pacote <code>usethis()</code> para criar "documento R"	<code>create_project(path = "C:/local padrão/nome_da_pasta/")</code>
	<-		Criar e altera objetos no R	
<code>readr()</code>	<code>read_csv2()</code>	Importar dados .csv	Função para importar dados do tipo .csv	<code>nome_dado_ao_arquivo <- read.csv2(file = "pasta_armazenada/nome_do_documento.csv")</code>
<code>foreign()</code>	<code>as.is = TRUE</code>	Importar dados .dbf	Função para transformar os dados em caracteres/texto	<code>nome_dado_ao_arquivo <- read.dbf(file = "pasta_armazenada/nome_do_documento.dbf", as.is = TRUE)</code>
	<code>head(nome_tabela)</code>		Argumento para visualizar as primeiras linhas da tabela	<code>head(nome_dado_ao_arquivo)</code>

<p><i>readxl()</i></p>	<p><i>skip(nº linha)</i> <i>skip = pular</i></p>	<p>Importar dados .xlsx</p>	<p>Argumento que delimita em que linha inicia a importação</p>	<p><i>nome_dado_ao_arquivo</i> <- <i>read_excel("pasta_armazenada/nome_do_documento.xls", sheet = "NOMEDODOCUMENTO", skip = 0)</i></p>
	<p><i>sheet(planilha)</i></p>		<p>Argumento que delimita qual planilha será importada</p>	
	<p><i>read_excel()</i></p>		<p>Função de ler a primeira planilha do documento e a primeira linha não vazia.</p>	
	<p><i>sapply()</i></p>		<p>Função que mostra o tipo de variável do banco</p>	<p><i>sapply (Z = nome_dado_ao_arquivo, FUN = "typeof")</i> <i>* typeof= tipo</i></p>
	<p><i>fun()</i></p>		<p>Função que configura o que se quer ver das variáveis</p>	
	<p><i>fator()</i></p>		<p>Função de alterar o tipo de variável</p>	<p><i>novo_nome_dado_ao_arquivo\$NOVO_NOME_DA_LINHA_OU_COLUNA</i></p>
	<p><i>\$</i></p>		<p>Selecionar variável no banco de dados</p>	<p><- <i>fator (x= nome_dado_ao_arquivo\$NOME_DA_LINHA_OU_COLUNA,</i></p>
	<p><i>levels</i></p>		<p>Hierarquização valores usados como categorias</p>	<p><i>= levels = c("nº", "nº", "nº"), c("nome_correspondente_ao_nº", "nome_correspondente_ao_nº", "nome_correspondente_ao_nº")</i></p>
	<p><i>labels</i></p>		<p>Nomes das categorias</p>	<p><i>nome_correspondente_ao_nº))</i></p>
	<p><i>C()</i></p>		<p>Função de unir valores em um conjunto</p>	
	<p><i>ordered = TRUE()</i></p>		<p>Função de ordem hierárquica para variável ordinal</p>	<p><i>novo_nome_dado_ao_arquivo\$NOVO_NOME_DA_LINHA_OU_COLUNA</i> <- <i>fator (x= nome_dado_ao_arquivo\$NOME_DA_LINHA_OU_COLUNA, levels = c ("nº", "nº", "nº"), c("nome_correspondente_ao_nº", "nome_correspondente_ao_nº", "nome_correspondente_ao_nº"), ordered = TRUE)</i></p>
<p><i>janitor</i></p>	<p><i>clean_names()</i></p>	<p>Agrupar categorias</p>	<p>Torna variáveis minúsculas e retira caracteres como</p>	<p>• <i>nome_dado_ao_arquivo</i> <- <i>novo_nome_dado_ao_arquivo</i> ></p>

			<p>acento e espaço entre as palavras</p> <p>Visualizar o nome das colunas</p>	<p><i>clean_names(nome_da_linha_ou_coluna)</i></p> <p>* <i>colnames(novo_nome_dado_ao_arquivo)</i></p>
	<i>colnames()</i>			
<i>skimr</i>	<i>n_complete</i>	Verifica o preenchimento de variáveis	Total de células preenchidas	
	<i>n_missing</i>		Número de células em branco	
	<i>complete_rate</i>		Células preenchidas/Total de células	
<i>stringi</i>		Correção de texto		<ul style="list-style-type: none"> • <i>stri_trans_general(nome_dado_ao_arquivo, id = "Latin-ASCII")</i>
<i>lubridate</i>		Organiza datas		<ul style="list-style-type: none"> • <i>as_date("10031995", format = "%d%m%Y")</i> • <i>nome_dado_ao_arquivo\$LINHA_OU_COLUNA_DA_DATA + ou - nº</i> • <i>novo_nome_dado_ao_arquivo <- nome_dado_ao_arquivo > select(LINHA_OU_COLUNA_DA_DATA) > mutate(nova_linha_ou_coluna = LINHA_OU_COLUNA_DA_DATA + nº) novo_nome_dado_ao_arquivo - comparação entre datas</i> • <i>novo_nome_dado_ao_arquivo <- nome_dado_ao_arquivo > select(LINHA_OU_COLUNA_DA_DATA, LINHA_OU_COLUNA_DA_DATA_1) > mutate(DIFERENCA = LINHA_OU_COLUNA_DA_DATA - LINHA_OU_COLUNA_DA_DATA_1) - diferença de dias entre datas</i> • <i>novo_nome_dado_ao_arquivo <- nome_dado_ao_arquivo ></i>

				<pre>mutate(ano = year(LINHA_OU_COLUNA_DA_DATA), mes = month(LINHA_OU_COLUNA_DA_DATA), dia = day(LINHA_OU_COLUNA_DA_DATA), semana_num = wday(LINHA_OU_COLUNA_DA_DATA), semana_nome = wday(LINHA_OU_COLUNA_DA_DATA, label = TRUE), semana_nome_completo = wday(LINHA_OU_COLUNA_DA_DATA, label = TRUE, abbr = FALSE), semana_epidemiologica = epiweek(LINHA_OU_COLUNA_DA_DATA))</pre> <ul style="list-style-type: none"> • <code>seq.Date(from = as.Date("2022-01-01"), to = as.Date("2022-02-01"), by = "day")</code> <p>* <code>floor()</code> = arredonda para valores inteiros</p>
<p>Tidyverse 2 (Meta-pacote para auxiliar nas análises de dados)</p>	ggplot	Pacote que cria tabelas		<ul style="list-style-type: none"> • <code>novo_nome_dado_ao_arquivo > pivot_longer(cols = c("Ano U", "Ano V", "Ano W", "Ano X", "Ano Y"), names_to = "Ano", values_to = "nova_linha_ou_coluna")</code> • <code>novo_nome_dado_ao_arquivo_linha <- novo_nome_dado_ao_arquivo > pivot_longer(cols = c("Ano U", "Ano V", "Ano W", "Ano X", "Ano Y"), names_to = "Ano", values_to = "nova_linha ", names_prefix = "Ano ")</code> • <code>novo_nome_dado_ao_arquivo_coluna <- novo_nome_dado_ao_arquivo_linha > pivot_wider(names_from = "Ano", values_from = "</code>

					<p><i>nova_coluna")</i></p> <ul style="list-style-type: none"> * <i>pivot_longer () = muda coluna e linha de lugar</i> * <i>cols () = colunas que serão mudadas</i> * <i>names_to() = define nome da variável criada e recebe característica que veio do formato largo</i> * <i>values_to () = define a variável dos valores</i> * <i>name_prefix () = exclui caracteres</i> * <i>pivot_wide () = muda linha e coluna de lugar</i> * <i>names_from () = coluna que será mudada</i> * <i>values_from () = coluna que contém os valores que serão mudados</i>
	<i>tibble</i>		Pacote para leitura e gravação de um moderno formato de banco de dados		
	<i>tidyr</i>		Pacote para auxílio para arrumar e transformar banco de dados		
	<i>readr</i>		Pacote para leitura e gravação de arquivos delimitados por vírgulas ou tabulações		
	<i>purrr</i>		Pacote para otimização e desempenho de algoritmos		
	<i>dplyr</i>	<i>glimpse()</i>	Pacote para manipulação	Fornece um sumário dos seus dados	<i>glimpse(novo_nome_dado_ao_arquivo)</i>
		<i>select()</i>	limpeza de algoritmos	Selecionar, excluir e organizar as colunas	<ul style="list-style-type: none"> • <i>nome_dado_ao_arquivo <- novo_nome_dado_ao_arquivo > select(LINHA_OU_COLUNA, LINHA_OU_COLUNA_1, LINHA_OU_COLUNA_2)</i> • <i>nome_dado_ao_arquivo > select(-LINHA_OU_COLUNA > head())</i>
		<i>mutate()</i>		Cria colunas e transforma as	<ul style="list-style-type: none"> • <i>novo_nome_dado_ao_arquivo <- nome_dado_ao_arquivo ></i>

			existentes	<pre>select(LINHA_OU_COLUNA, LINHA_OU_COLUNA_1, LINHA_OU_COLUNA_2) > mutate(NOVA_LINHA_OU_COLU NA = LINHA_OU_COLUNA - LINHA_OU_COLUNA)</pre> <ul style="list-style-type: none"> • <code>novo_nome_dado_ao_arquivo > mutate(NOVA_LINHA_OU_COLU NA = as.numeric(NOVA_LINHA_OU_C OLUNA)) > head()</code> <p>* <code>as.numeric(variável) = muda a classe</code></p>
		<code>group_by()</code>	Cria grupos de registros baseado em critérios em colunas	<ul style="list-style-type: none"> • <code>novo_nome_dado_ao_arquivo > group_by(LINHA_OU_COLUNA) > count(LINHA_OU_COLUNA)</code> <p>* <code>count=contar</code></p>
		<code>filter()</code>	Filtra um conjunto de registros baseado em critério de inclusão ou exclusão	<ul style="list-style-type: none"> • <code>novo_nome_dado_ao_arquivo > filter(LINHA_OU_COLUNA == "X", LINHA_OU_COLUNA > 7) > head()</code> • <code>novo_nome_dado_ao_arquivo > filter(LINHA_OU_COLUNA == "X" LINHA_OU_COLUNA == "Z") > head(10)</code> • <code>novo_nome_dado_ao_arquivo > filter(LINHA_OU_COLUNA %in% c("X", "Y", "Z")) > head(25)</code> <p>* <code> = ou</code></p>
		<code>summarise()</code>	Cria colunas utilizando operações de síntese das linhas	<ul style="list-style-type: none"> • <code>novo_nome_dado_ao_arquivo > group_by(LINHA_OU_COLUNA) > summarise(linha_ou_coluna = n(), linha_ou_coluna_1 = mean(NOVA_LINHA_OU_COLU NA)) > print(n=30)</code> • <code>nome_dado_ao_arquivo > summarise(nova_coluna_total = sum(!is.na(LINHA_OU_COLUNA)), nova_coluna_total_registros = n(), nova_coluna_total_em_branco =</code>

				<pre>sum(is.na(LINHA_OU_COLUNA)) , taxa_completude = (nova_coluna_total/ nova_coluna_total_registros) 100)</pre> <ul style="list-style-type: none"> • nome_dado_ao_arquivo > summarise(nova_coluna_total = n_complete(LINHA_OU COLUNA), nova_coluna_total_em_branco (LINHA_OU COLUNA), taxa_completude = complete_rate(LINHA_OU COLUNA) * 100) <p>* print(n=nº) = visualiza o número de linhas solicitadas * sum() = soma * is.na() = verdadeiro/falso * !is.na() = registros preenchidos</p>
	arrange()	Ordena os valores dos registros conforme colunas definidas em ordem crescente	<ul style="list-style-type: none"> • nome_dado_ao_arquivo > group_by(LINHA_OU_COLUNA) > summarise(nova_coluna_total = n(), nova_coluna_mean(NOVA_LINHA_OU_COLUNA)) > arrange(desc(nova_coluna)) • desc = ordem decrescente 	
	slice()	Seleciona linhas específicas		
	na.if()	Converte itens	<ul style="list-style-type: none"> • nome_dado_ao_arquivo > mutate(LINHA_OU_COLUNA = na_if(LINHA_OU_COLUNA, "9")) > count(LINHA_OU_COLUNA) 	
	case_when()	Recodifica dados	<ul style="list-style-type: none"> • nome_dado_ao_arquivo > mutate(sexo_cat = case_when(CS_SEXO == "M" ~ "Masculino", CS_SEXO == "F" ~ "Feminino", CS_SEXO == "I" ~ "Ignorado", TRUE ~ NA_character_)) > 	

					<p><i>head()</i></p>
	<i>if_else()</i>		Transforma número em texto	<ul style="list-style-type: none"> • <i>basededados3</i> > <i>mutate(idade_anos = if_else(str_sub(NU_IDADE_N, 1, 1) == "4", as.numeric(str_sub(NU_IDADE_N, 2, 4)), 0), fx_etaria = cut(x = idade_anos, breaks = c(0, 10, 20, 60, Inf), labels = c("0-9 anos", "10-19 anos", "20-59 anos", "60 anos e+"), right = FALSE))</i> > <i>select(NU_NOTIFIC, ID_AGRAVO, NU_IDADE_N, idade_anos, fx_etaria)</i> > <i>head()</i> 	
	<i>Left_join()</i>		Uni diferentes tabelas	<ul style="list-style-type: none"> • <i>left_join(x = arquivo_1, y = arquivo_2, by = c("COLUNA_1" = "COLUNA_S"))</i> > <i>head()</i> * <i>class()</i> = ver tipo da variável da coluna * <i>as.integer()</i> = transformar em variável integer, pode-se mudar pelos outros tipos de variáveis – <i>character</i>, <i>numeric</i> ou <i>double</i> 	
<i>stringr</i>		Pacote para manipulação de texto	manipulação de texto	<ul style="list-style-type: none"> • <i>str_to_upper(tabela1)</i> = transforma tudo em letra maiúscula • <i>str_to_lower(tabela1)</i> = transforma tudo em letra maiúscula • <i>str_to_sentence(tabela1)</i> = transforma primeira letra em maiúscula e o resto em minúscula • <i>agravo <- "Tuberculose respiratória, com confirmação bacteriológica e histológica"</i> ↓ <i>str_replace(agravo, pattern = "com", replacement = "sem")</i> ou 	

				<p><i>str_replace_all</i> = para todas as palavras da tabela</p> <ul style="list-style-type: none"> • <i>str_sort(tabela1)</i> = ordem alfabética crescente • <i>str_sort(tabela1, decreasing = TRUE)</i> = ordem alfabética decrescente
	>		Omite o argumento anterior a ele	<pre>novo_arquivo <- arquivo > select(COLUNA_1, COLUNA_2, COLUNA_3, COLUNA_4, COLUNA_5, COLUNA_6)</pre>
<i>tidr()</i>	<i>replace_na()</i>	Substituição de valores	Substitui valores	<ul style="list-style-type: none"> • <i>novo_arquivo > mutate(COLUNA = replace_na(COLUNA, replace = 9)) > count(CS_RACA)</i> • <i>eapv > separate(col = imunobiologico_vacina, into = c("vac_event_1", "vac_event_2", "vac_event_3"), sep = "\\ ") > select("vac_event_1", "vac_event_2", "vac_event_3")</i> <p>* <i>col()</i> = coluna que será separada * <i>into()</i> = nome da nova coluna * <i>sep()</i> = tipo de separador usado na célula * Devido ao ter outras funções usou-se duas barras para o R interpretar diferente</p>
	<i>write()</i>		Exportar tabelas	<ul style="list-style-type: none"> • <i>write_csv(x = base1, file = "C:/local_do_arquivo/nome_do_arquivo)</i> • <i>write_xlsx(x = base2, path = "C:/local_do_arquivo/nome_do_arquivo) - pacote writexl precisa ser instalado antes</i>
	<i>round()</i>		Arredonda números	<pre>round(proporcao * 100, digits = 2)</pre> <p>*<i>digits</i> – determina o número de casas decimais para função <i>round()</i></p>

Alguns operadores lógicos usado para filtrar com o *filter()* (WICKHAM,2019; SANTA, CATARINA 2022):

- == (registro igual a um valor)
- != (registro diferente de um valor)
- > e >= (registro maior e maior igual a um valor)
- < e <= (registro menor e menor igual a um valor)
- %in% (registro de uma variável igual a qualquer elemento de um conjunto, ou seja seleciona os valores)
- ! (registro que não atendem ao critério definido)

7. BIBLIOGRAFIA

MUSSO, Caroline. **Utilização de linguagem r para automatizar a análise e a Comunicação de casos de dengue sem encerramento por região de saúde no distrito federal visando melhoria da qualidade de Dados.** 2022. Pós-graduação Lato Sensu em Ciência de Dados e Machine Learning. Centro Universitário de Brasília, DF, 2022. Disponível em: < <https://repositorio.uniceub.br/jspui/handle/prefix/16246>>.

SANTA CATARINA. UNIVERSIDADE FEDERAL DE SANTA CATARINA. Análise de dados para a vigilância em saúde. 2022. Disponível em: < <https://www.abrasco.org.br/site/analise-de-dados-para-a-vigilancia-em-saude/>>.

WICKHAM, Hadley. R para data science: Importe, arrume, transforme, visualize e modele dados. 1. ed. Rio de Janeiro: Alta Books. 2019.

ROCHA, T.A.H. et al. Geolocalização de internações cadastradas no Sistema de Informações Hospitalares do Sistema Único de Saúde: uma solução baseada no programa estatístico R. **Epidemiol. Serv. Saude**, Brasília, v. 27, n: 4, p. 1-10, 2018. Disponível em: < <https://www.scielo.br/j/ress/a/NLPLHWwpx9VdrR7tRsb6Myz/abstract/?lang=pt#>>.

BRASIL. CONSELHO NACIONAL DE SAÚDE. RESOLUÇÃO Nº 588, DE 12 DE JULHO DE 2018. Brasil. 2018. Disponível em: <<http://conselho.saude.gov.br/resolucoes/2018/Reso588.pdf>>.

TEIXEIRA, Maria Glória et al. Vigilância em Saúde no SUS-construção, efeitos e perspectivas. **Ciência & Saúde Coletiva**, v. 23, p. 1811-1818, 2018.

GUIMARÃES, Raphael Mendonça et al. Os desafios para a formulação, implantação e implementação da Política Nacional de Vigilância em Saúde. **Ciência & Saúde Coletiva**, v. 22, p. 1407-1416, 2017.

PINTO, Daniel Silva et al. Avaliação do conhecimento sobre vigilância em saúde entre os profissionais do Sistema Único de Saúde, Uberlândia, Minas Gerais. **J. Health Biol Sci.**, v. 5, p. 37-43, 2017.

ROCHA, Andrei Fernandes da et al. Opinião. Sobre a Residência Integrada em Saúde com ênfase em Vigilância em Saúde. **Ciência & Saúde Coletiva**, v. 22, p. 3467-3472, 2017.

BARRA, Daniela Couto Carvalho et al. Evolução Histórica e Impacto da Tecnologia na Área da Saúde E da Enfermagem. *Revista Eletrônica de Enfermagem*, v. 08, n. 03, p. 422 - 430, 2006