

# Análises do dados de vacinação

Gestantes e puérperas

07/junho/2023

## Contents

<b>1</b>	<b>Sobre a base de dados e pacotes do R utilizados</b>	<b>1</b>
<b>2</b>	<b>Tratamento dos dados</b>	<b>9</b>
2.1	Variável que indica se gestante ou puérpera . . . . .	9
2.2	Raça . . . . .	10
2.3	Idade . . . . .	12
2.4	UF de aplicação . . . . .	12
2.5	Município de aplicação . . . . .	14
2.6	UF de residência . . . . .	14
2.7	Município de residência . . . . .	17
2.8	Qual vacina aplicada . . . . .	17
2.9	Qual dose . . . . .	19
2.10	Sobre tempo entre aplicação e importação no RNDS . . . . .	23
<b>3</b>	<b>Exclusão dos casos inconsistentes</b>	<b>28</b>
3.1	Dados com status “entered-in-error” . . . . .	28
3.2	ID do paciente que se repete para um mesmo número de dose e para o mesmo estabelecimento de saúde . . . . .	29
3.3	Ano de aplicação da vacina . . . . .	29

## 1 Sobre a base de dados e pacotes do R utilizados

A seguir são carregados os pacotes do R (<https://www.r-project.org>) utilizados para filtragem e tratamento dos dados considerados no dashboard <https://observatorioobstetrico.shinyapps.io/vacinacao-covid19>.

```
#carregar pacotes
loadlibrary <- function(x) {
  if (!require(x, character.only = TRUE)) {
    install.packages(x, dependencies = T)
    if (!require(x, character.only = TRUE))
      stop("Package not found")
  }
}
```

```

packages <-
  c(
    "readr",
    "readxl",
    "janitor",
    "dplyr",
    "forcats",
    "stringr",
    "lubridate",
    "summarytools",
    "magrittr",
    "questionr",
    "knitr",
    "data.table",
    "janitor",
    "modelsummary"
  )
lapply(packages, loadlibrary)

options(scipen=999)

```

```

#funções que vamos usar para as medidas descritivas
media <- function(x)
  mean(x, na.rm = TRUE)
mediana <- function(x)
  median(x, na.rm = TRUE)
DP <- function(x)
  sd(x, na.rm = TRUE)
minimo <- function(x)
  base::min(x, na.rm = TRUE)
maximo <- function(x)
  base::max(x, na.rm = TRUE)
q25 <- function(x)
  stats::quantile(x, p = 0.25, na.rm = TRUE)
q75 <- function(x)
  stats::quantile(x, p = 0.75, na.rm = TRUE)

```

```

IQR <- function(x)
  round(q75(x) - q25(x), 2)
n <- function(x)
  sum(!is.na(x))
faltantes <- function(x)
  round(sum(is.na(x)), digits = 0)

#dados <- read_csv("vacinacao-total-2023-04-04.csv")
dados1 <- read.csv("vacinacao_covid_001801.csv")
dados2 <- read.csv("vacinacao_covid_001901.csv")

dados <- rbind(dados1, dados2)
remove(dados1)
remove(dados2)

```

Dados obtidos em <https://opendatasus.saude.gov.br/dataset/covid-19-vacinacao> na data de 05/junho/2023. A extração dos dados é realizada por meio de consultas à API ElasticSearch disponibilizada em <https://opendatasus.saude.gov.br/dataset/covid-19-vacinacao>.

Além disso, visando diminuir o volume de dados retornado pelas consultas, é realizada uma pré-filtragem dos dados diretamente na API, sendo retornados apenas registros cujo campo `vacina_grupoAtendimento_codigo` seja igual a 1901 (código para puerpera) ou 1801 (código para gestante). Essa etapa de extração foi realizada utilizando a linguagem de programação python.

### 1.0.1 Código para extração

```

import requests
import json
import time
import http.server
import socketserver
import csv

from apscheduler.schedulers.background import BackgroundScheduler
from datetime import datetime

def get_dados_vacinacao(grupoAtendimento = '001801'):
    start_time = time.time()
    print(f'Iniciando exportação de registros do grupo de atendimento {grupoAtendimento}')

```

```

        as {datetime.fromtimestamp(start_time).strftime("%H:%M:%S")}')
total = 0
response = requests.post("https://imunizacao-es.saude.gov.br/_search?scroll=1m",
    json={
        'size': 10000,
        'query': {
            'bool': {
                'filter': [{
                    'term': {
                        'vacina_grupoAtendimento_codigo': grupoAtendimento
                    }
                }]
            }
        }
    },
    auth=('imunizacao_public', 'qlto5t&7r_@+##Tlstigi'),
    headers={'Content-type': 'application/json'})
json_response = json.loads(response.text)
print(f'Total de registros a serem exportados: {json_response["hits"]["total"]["value"]}')
hits = [ dict(sorted(hit['_source'].items())) for hit in json_response['hits']['hits'] ]
csv_header = hits[0].keys()

with open(f'vacinacao_covid_{grupoAtendimento}.csv', 'w', encoding="utf-8") as f:
    writer = csv.DictWriter(f, fieldnames=csv_header)
    writer.writeheader()
    writer.writerows(hits)

total += len(hits)
print(f'Registros exportados: {total}')

while (hits):
    response = requests.post("https://imunizacao-es.saude.gov.br/_search/scroll",
        json={ 'scroll': '1m', 'scroll_id': json.loads(response.text)['_scroll_id'] },
        auth=('imunizacao_public', 'qlto5t&7r_@+##Tlstigi'),
        headers={'Content-type': 'application/json'})
    hits = [ dict(sorted(hit['_source'].items())) for hit in json.loads(response.text)['hits']['hits'] ]

```

```

with open(f'vacinacao_covid_{grupoAtendimento}.csv', 'a', encoding="utf-8") as f:
    writer = csv.DictWriter(f, fieldnames=csv_header)
    writer.writerows(hits)

total += len(hits)
print(f'Registros exportados: {total}')

print(f'Finalizando exportação de registros do grupo de atendimento {grupoAtendimento} as {time.strftime("%H:%M:%S", time.localtime())}')
print(f'Importado {total} de registros em {"{:.2f}".format((time.time() - start_time)/60)} minutos')
return hits

get_dados_vacinacao('001901')
get_dados_vacinacao('001801')

```

*#Criação das variáveis de data, mês e ano da aplicação*

```

dados <- dados %>%
  dplyr::mutate(
    data_nascimento = str_sub(paciente_dataNascimento, start = 1, end = 10),
    data_aplic = str_sub(vacina_dataAplicacao, start = 1, end = 10),
    data_import = str_sub(data_importacao_rnds, start = 1, end = 10),
    dt_nasc = as.Date(data_nascimento),
    dt_aplic = as.Date(data_aplic),
    dt_import = as.Date(data_import),
    ano_aplic = year(dt_aplic),
    mes_aplic = month(dt_aplic)
  )

summary(dados$dt_aplic)

```

```

##           Min.          1st Qu.          Median          Mean          3rd Qu.          Max.
## "1989-01-20" "2021-07-07" "2021-09-13" "2021-11-14" "2022-01-29" "2023-06-05"

```

```
summary(dados$dt_import)
```

```

##           Min.          1st Qu.          Median          Mean          3rd Qu.          Max.
## "1989-01-20" "2021-07-13" "2021-09-20" "2021-08-31" "2022-02-24" "2023-06-05"

```

A data máxima de aplicação é 2023-06-05 e a data máxima de importação é 2023-06-05.

Os dados já estão filtrados vacina\_grupoAtendimento\_codigo igual a 1801 (código de gestante) ou 1901 (código de puerpera).

```
dados$vacina_grupoAtendimento_codigo <- as.character(dados$vacina_grupoAtendimento_codigo)
#tabela de frequência para grupo de atendimento
questionr::freq(
  dados$vacina_grupoAtendimento_codigo,
  cum = FALSE,
  total = TRUE,
  na.last = TRUE,
  valid = TRUE
) %>%
  kable(caption = "Tabela de frequências para grupo de atendimento", digits = 2, format = "markdown")
```

Table 1: Tabela de frequências para grupo de atendimento

	n	%	val%
1801	2246446	77.5	77.5
1901	653754	22.5	22.5
Total	2900200	100.0	100.0

Há atualmente 2900200 observações na base de dados e são as variáveis:

```
names(dados)

## [1] "X.timestamp"
## [2] "X.version"
## [3] "co_condicao_maternal"
## [4] "data_importacao_datalake"
## [5] "data_importacao_rnds"
## [6] "document_id"
## [7] "ds_condicao_maternal"
## [8] "dt_deleted"
## [9] "estabelecimento_municipio_codigo"
## [10] "estabelecimento_municipio_nome"
## [11] "estabelecimento_razaoSocial"
## [12] "estabelecimento_uf"
## [13] "estabelecimento_valor"
```

```
## [14] "estalecimento_noFantasia"
## [15] "id_sistema_origem"
## [16] "paciente_dataNascimento"
## [17] "paciente_endereco_cep"
## [18] "paciente_endereco_coIbgeMunicipio"
## [19] "paciente_endereco_coPais"
## [20] "paciente_endereco_nmMunicipio"
## [21] "paciente_endereco_nmPais"
## [22] "paciente_endereco_uf"
## [23] "paciente_enumSexoBiologico"
## [24] "paciente_id"
## [25] "paciente_idade"
## [26] "paciente_nacionalidade_enumNacionalidade"
## [27] "paciente_racaCor_codigo"
## [28] "paciente_racaCor_valor"
## [29] "sistema_origem"
## [30] "status"
## [31] "vacina_categoria_codigo"
## [32] "vacina_categoria_nome"
## [33] "vacina_codigo"
## [34] "vacina_dataAplicacao"
## [35] "vacina_descricao_dose"
## [36] "vacina_fabricante_nome"
## [37] "vacina_fabricante_referencia"
## [38] "vacina_grupoAtendimento_codigo"
## [39] "vacina_grupoAtendimento_nome"
## [40] "vacina_lote"
## [41] "vacina_nome"
## [42] "vacina_numDose"
## [43] "data_nascimento"
## [44] "data_aplic"
## [45] "data_import"
## [46] "dt_nasc"
## [47] "dt_aplic"
## [48] "dt_import"
## [49] "ano_aplic"
```

```
## [50] "mes_aplic"
```

Veja que há também as variáveis `vacina_grupoAtendimento_nome` e `vacina_categoria_codigo` e `vacina_categoria_nome`.

```
with(
  dados,
  ctable(
    vacina_grupoAtendimento_codigo,
    vacina_grupoAtendimento_nome,
    prop = "n"
  )
)
```

```
## Cross-Tabulation
```

```
## vacina_grupoAtendimento_codigo * vacina_grupoAtendimento_nome
```

```
## Data Frame: dados
```

```
##
```

```
## -----
```

	<code>vacina_grupoAtendimento_nome</code>	Gestante	Puérpera	Total
<code>vacina_grupoAtendimento_codigo</code>				
1801		2246446	0	2246446
1901		0	653754	653754
Total		2246446	653754	2900200

```
## -----
```

```
dados$vacina_categoria_codigo <-
  as.character(dados$vacina_categoria_codigo)
with(dados,
  ctable(
    vacina_grupoAtendimento_codigo,
    vacina_categoria_codigo,
    prop = "n"
  ))
```

```
## Cross-Tabulation
```

```
## vacina_grupoAtendimento_codigo * vacina_categoria_codigo
```

```
## Data Frame: dados
```

```
##
```



```
## -----
##                vacina_categoria_codigo      21      25      Total
##  vacina_grupoAtendimento_codigo
##                1801                2246446      0      2246446
##                1901                0      653754      653754
##                Total                2246446      653754      2900200
## -----
```

```
with(dados,
      ctable(vacina_grupoAtendimento_codigo, vacina_categoria_nome, prop = "n"))
```

```
## Cross-Tabulation
```

```
## vacina_grupoAtendimento_codigo * vacina_categoria_nome
```

```
## Data Frame: dados
```

```
##
```

```
## -----
##                vacina_categoria_nome  Gestantes  Puérperas  Puérperas  Tot
##  vacina_grupoAtendimento_codigo
##                1801                2246446      0          0      22464
##                1901                0      436477      217277      6537
##                Total                2246446      436477      217277      29002
## -----
```

## 2 Tratamento dos dados

### 2.1 Variável que indica se gestante ou puérpera

```
dados <- dados %>%
  mutate(gest_puerp = ifelse(vacina_grupoAtendimento_codigo == "1801",
                             "gesta",
                             "puerp"))
```

```
#tabela de frequência para gest_puerp
```

```
questionr::freq(
  dados$gest_puerp,
  cum = FALSE,
  total = TRUE,
  na.last = TRUE,
```

```

    valid = TRUE
  ) %>%
  kable(caption = "Tabela de frequências para gestante ou puérpera", digits = 2, format = "markdown")

```

Table 2: Tabela de frequências para gestante ou puérpera

	n	%	val%
gesta	2246446	77.5	77.5
puerp	653754	22.5	22.5
Total	2900200	100.0	100.0

## 2.2 Raça

```

#tabela de frequência raca/cor original
questionr::freq(
  dados$paciente_racaCor_codigo,
  cum = FALSE,
  total = TRUE,
  na.last = TRUE,
  valid = TRUE
) %>%
  kable(caption = "Tabela de frequências para raça/cor original", digits = 2, format = "markdown")

```

Table 3: Tabela de frequências para raça/cor original

	n	%	val%
1	1031614	35.6	35.6
2	139861	4.8	4.8
3	847875	29.2	29.2
4	456517	15.7	15.7
5	3727	0.1	0.1
99	420377	14.5	14.5
NA	229	0.0	NA
Total	2900200	100.0	100.0

```
#Criando a variável de raça/cor
```

```
dados <- dados %>%
```

```
  mutate(  
    raca = case_when(  
      paciente_racaCor_codigo == "1" ~ "branca",  
      paciente_racaCor_codigo == "2" ~ "preta",  
      paciente_racaCor_codigo == "3" ~ "parda",  
      paciente_racaCor_codigo == "4" ~ "amarela",  
      paciente_racaCor_codigo == "5" ~ "indigena",  
      TRUE ~ NA_character_  
    )  
  )  
)
```

```
#tabela de frequência raça/cor
```

```
questionr::freq(  
  dados$raca,
```

```
  cum = FALSE,
```

```
  total = TRUE,
```

```
  na.last = TRUE,
```

```
  valid = TRUE
```

```
) %>%
```

```
kable(caption = "Tabela de frequências para raça/cor", digits = 2, format = "markdown")
```

Table 4: Tabela de frequências para raça/cor

	n	%	val%
amarela	456517	15.7	18.4
branca	1031614	35.6	41.6
indigena	3727	0.1	0.2
parda	847875	29.2	34.2
preta	139861	4.8	5.6
NA	420606	14.5	NA
Total	2900200	100.0	100.0

## 2.3 Idade

```
#Criação da variável de idade como diferença entre data de nascimento e de aplicação
dados <- dados %>%
  mutate(
    idade_anos = as.period(interval(start = dt_nasc, end = dt_aplic))$year,
    dif_idade = idade_anos - paciente_idade
  )

#Descritiva da diferença da idade (base de dados) e criada pela diferença de
#datas de nascimento e aplicação
datasummary((dif_idade) ~ 1*(n+faltantes+media+DP+mediana+minimo+maximo),
  data = dados, output = 'markdown')
```

	n	faltantes	media	DP	mediana	minimo	maximo
dif_idade	2899971.00	229.00	0.00	0.00	0.00	0.00	0.00

Podemos observar que há 0 observações com a variável original de idade (`paciente_idade`) diferente da variável criada como a diferença entre data de nascimento e de aplicação da dose.

## 2.4 UF de aplicação

```
#Criando a variável de UF de aplicação
dados <- dados %>%
  mutate(
    aplic_uf = case_when(
      estabelecimento_uf == "" ~ NA_character_,
      estabelecimento_uf == "XX" ~ NA_character_,
      TRUE ~ estabelecimento_uf
    )
  )
```

```
#tabela de frequência para UF aplicação
questionr::freq(
  dados$aplic_uf,
  cum = FALSE,
  total = TRUE,
```

```

na.last = TRUE,
valid = TRUE
) %>%
kable(caption = "Tabela de frequências para UF de aplicação", digits = 2, format = "markdown")

```

Table 6: Tabela de frequências para UF de aplicação

	n	%	val%
AC	7255	0.3	0.3
AL	34106	1.2	1.2
AM	59258	2.0	2.0
AP	6724	0.2	0.2
BA	201770	7.0	7.0
CE	95446	3.3	3.3
DF	44692	1.5	1.5
ES	88449	3.0	3.0
GO	100849	3.5	3.5
MA	83644	2.9	2.9
MG	366333	12.6	12.6
MS	21055	0.7	0.7
MT	60560	2.1	2.1
PA	95372	3.3	3.3
PB	71885	2.5	2.5
PE	202015	7.0	7.0
PI	54056	1.9	1.9
PR	243616	8.4	8.4
RJ	212035	7.3	7.3
RN	67444	2.3	2.3
RO	29110	1.0	1.0
RR	1617	0.1	0.1
RS	164777	5.7	5.7
SC	96384	3.3	3.3
SE	35964	1.2	1.2
SP	440304	15.2	15.2
TO	15480	0.5	0.5
Total	2900200	100.0	100.0

## 2.5 Município de aplicação

```
#Criando a variável de município (código IBGE) da aplicação
dados <- dados %>%
  mutate(
    aplic_muni = ifelse(
      is.na(estabelecimento_municipio_codigo) |
        estabelecimento_municipio_codigo == 999999,
      NA,
      estabelecimento_municipio_codigo
    )
  )
```

## 2.6 UF de residência

```
#Criando a variável de UF da residência
dados <- dados %>%
  mutate(
    resid_uf = case_when(
      paciente_endereco_uf == "" ~ NA_character_,
      paciente_endereco_uf == "XX" ~ NA_character_,
      paciente_endereco_uf == "None" ~ NA_character_,
      TRUE ~ paciente_endereco_uf
    )
  )
```

```
#tabela de frequência para UF residência
questionr::freq(
  dados$resid_uf,
  cum = FALSE,
  total = TRUE,
  na.last = TRUE,
  valid = TRUE
) %>%
  kable(caption = "Tabela de frequências para UF de residência", digits = 2, format = "markdown")
```

Table 7: Tabela de frequências para UF de residência

	n	%	val%
AC	7706	0.3	0.3
AL	36345	1.3	1.3
AM	58477	2.0	2.0
AP	6975	0.2	0.2
BA	207169	7.1	7.2
CE	98049	3.4	3.4
DF	45708	1.6	1.6
ES	86209	3.0	3.0
GO	99068	3.4	3.4
MA	81191	2.8	2.8
MG	365247	12.6	12.6
MS	22501	0.8	0.8
MT	61259	2.1	2.1
PA	96029	3.3	3.3
PB	74552	2.6	2.6
PE	196801	6.8	6.8
PI	53714	1.9	1.9
PR	236360	8.1	8.2
RJ	209599	7.2	7.3
RN	67180	2.3	2.3
RO	29379	1.0	1.0
RR	2339	0.1	0.1
RS	165555	5.7	5.7
SC	100048	3.4	3.5
SE	35707	1.2	1.2
SP	430620	14.8	14.9
TO	16720	0.6	0.6
NA	9693	0.3	NA
Total	2900200	100.0	100.0

Podemos observar que há 9693 NA (dados faltantes) para UF de residência. Vamos verificar a distribuição de dados faltantes da UF de residência por UF de aplicação:

```

#criação do indicador de missing para UF de residência
dados <- dados %>%
  mutate(indic_na_uf_resid = ifelse(is.na(resid_uf), "sim", "não"))

with(dados, ctable(aplic_uf, indic_na_uf_resid, prop = "r"))

## Cross-Tabulation, Row Proportions
## aplic_uf * indic_na_uf_resid
## Data Frame: dados
##
## -----
##      indic_na_uf_resid      não      sim      Total
## aplic_uf
## AC      7242 ( 99.8%)      13 (0.2%)      7255 (100.0%)
## AL      34035 ( 99.8%)      71 (0.2%)      34106 (100.0%)
## AM      58947 ( 99.5%)      311 (0.5%)      59258 (100.0%)
## AP       6698 ( 99.6%)       26 (0.4%)       6724 (100.0%)
## BA     201197 ( 99.7%)      573 (0.3%)     201770 (100.0%)
## CE      94813 ( 99.3%)      633 (0.7%)     95446 (100.0%)
## DF     44433 ( 99.4%)      259 (0.6%)     44692 (100.0%)
## ES      88220 ( 99.7%)      229 (0.3%)     88449 (100.0%)
## GO     100538 ( 99.7%)      311 (0.3%)     100849 (100.0%)
## MA      83256 ( 99.5%)      388 (0.5%)     83644 (100.0%)
## MG     365150 ( 99.7%)     1183 (0.3%)    366333 (100.0%)
## MS      21013 ( 99.8%)       42 (0.2%)     21055 (100.0%)
## MT      60448 ( 99.8%)      112 (0.2%)     60560 (100.0%)
## PA      95032 ( 99.6%)      340 (0.4%)     95372 (100.0%)
## PB      71660 ( 99.7%)      225 (0.3%)     71885 (100.0%)
## PE     201168 ( 99.6%)      847 (0.4%)     202015 (100.0%)
## PI      53783 ( 99.5%)      273 (0.5%)     54056 (100.0%)
## PR     242983 ( 99.7%)      633 (0.3%)     243616 (100.0%)
## RJ     211302 ( 99.7%)      733 (0.3%)     212035 (100.0%)
## RN      67291 ( 99.8%)      153 (0.2%)     67444 (100.0%)
## RO      29011 ( 99.7%)       99 (0.3%)     29110 (100.0%)
## RR       1606 ( 99.3%)       11 (0.7%)      1617 (100.0%)
## RS     164367 ( 99.8%)      410 (0.2%)     164777 (100.0%)

```



```
##          SC          96215 ( 99.8%)    169 (0.2%)    96384 (100.0%)
##          SE          35860 ( 99.7%)    104 (0.3%)    35964 (100.0%)
##          SP          438823 ( 99.7%)   1481 (0.3%)   440304 (100.0%)
##          TO          15416 ( 99.6%)     64 (0.4%)    15480 (100.0%)
##      Total          2890507 ( 99.7%)   9693 (0.3%)  2900200 (100.0%)
## -----
```

## 2.7 Município de residência

```
#Criando a variável de município (código IBGE) da residência
dados <- dados %>%
  mutate(
    resid_muni = ifelse(
      is.na(paciente_endereco_coIbgeMunicipio) |
        paciente_endereco_coIbgeMunicipio == 999999,
      NA,
      paciente_endereco_coIbgeMunicipio
    )
  )
```

## 2.8 Qual vacina aplicada

```
#tabela de frequência para vacina_nome
questionr::freq(
  dados$vacina_nome,
  cum = FALSE,
  total = TRUE,
  na.last = TRUE,
  valid = TRUE
) %>%
  kable(caption = "Tabela de frequências para vacina nome (original)", digits = 2, format = "markdown")
```

Table 8: Tabela de frequências para vacina nome (original)

	n	%	val%
COVID-19 ASTRAZENECA - ChAdOx1-S	6003	0.2	0.2
COVID-19 ASTRAZENECA/FIOCRUZ - COVISHIELD	249620	8.6	8.6

	n	%	val%
COVID-19 JANSSEN - Ad26.COV2.S	75758	2.6	2.6
COVID-19 PEDIÁTRICA - PFIZER COMIRNATY	435	0.0	0.0
COVID-19 PFIZER - COMIRNATY	1830585	63.1	63.1
COVID-19 PFIZER - COMIRNATY BIVALENTE	98494	3.4	3.4
COVID-19 PFIZER - COMIRNATY PEDIÁTRICA	156	0.0	0.0
COVID-19 PFIZER - COMIRNATY PEDIÁTRICA MENOR DE 5 ANOS	113	0.0	0.0
COVID-19 SINOVAC - CORONAVAC	2425	0.1	0.1
COVID-19 SINOVAC/BUTANTAN - CORONAVAC	636611	22.0	22.0
Total	2900200	100.0	100.0

```

dados$id <- 1:dim(dados)[1]
#Criando a variável de qual vacina aplicada
dados <- dados %>%
  mutate(
    qual_vacina = case_when(
      vacina_nome == "COVID-19 PFIZER - COMIRNATY BIVALENTE" ~ "Bivalente",
      id %in% str_which(vacina_nome, "AstraZeneca") ~ "AstraZeneca",
      id %in% str_which(vacina_nome, "ASTRA") ~ "AstraZeneca",
      id %in% str_which(vacina_nome, "Coronavac") ~ "Coronavac",
      id %in% str_which(vacina_nome, "CORONAVAC") ~ "Coronavac",
      id %in% str_which(vacina_nome, "JANSSEN") ~ "Janssen",
      id %in% str_which(vacina_nome, "PFIZER") ~ "Pfizer",
      id %in% str_which(vacina_nome, "Pfizer") ~ "Pfizer",
      id %in% str_which(vacina_nome, "Covishield") ~ "Covishield",
      vacina_nome == "" ~ NA_character_,
      vacina_nome == "Pendente Identificação" ~ NA_character_,
      vacina_nome == "Pendente Identificação" ~ NA_character_,
      vacina_nome == "INF3" ~ NA_character_,
      TRUE ~ vacina_nome
    )
  )
# vacina_nome == "Vacina covid-19 - Ad26.COV2.S - Janssen-Cilag" ~ "Janssen",
# vacina_nome == "Covid-19-AstraZeneca" ~ "AstraZeneca",
# vacina_nome == "Covid-19-Coronavac-Sinovac/Butantan" ~ "Coronavac",
# vacina_nome == "Vacina covid-19 - BNT162b2 - BioNTech/Fosun Pharma/Pfizer" ~ "Pfizer",

```

```

# vacina_nome == "Vacina Covid-19 - Covishield" ~ "Covishield",
# vacina_nome == "" ~ NA_character_,
# vacina_nome == "Pendente Identificação" ~ NA_character_,
# vacina_nome == "Pendente Identificação" ~ NA_character_,
# TRUE ~ vacina_nome
)
)

#tabela de frequência para qual_vacina
questionr::freq(
  dados$qual_vacina,
  cum = FALSE,
  total = TRUE,
  na.last = TRUE,
  valid = TRUE
) %>%
  kable(caption = "Tabela de frequências para qual vacina", digits = 2, format = "markdown")

```

Table 9: Tabela de frequências para qual vacina

	n	%	val%
AstraZeneca	255623	8.8	8.8
Bivalente	98494	3.4	3.4
Coronavac	639036	22.0	22.0
Janssen	75758	2.6	2.6
Pfizer	1831289	63.1	63.1
Total	2900200	100.0	100.0

## 2.9 Qual dose

```

#tabela de frequência para vacina_descricao_dose
questionr::freq(
  dados$vacina_descricao_dose,
  cum = FALSE,
  total = TRUE,
  na.last = TRUE,
  valid = TRUE
)

```

```
) %>%
  kable(caption = "Tabela de frequências para qual dose", digits = 2, format = "markdown")
```

Table 10: Tabela de frequências para qual dose

	n	%	val%
	4	0.0	0.0
1ª Dose	1062316	36.6	36.6
1ª Dose Revacinação	1477	0.1	0.1
1º Reforço	3640	0.1	0.1
2ª Dose	978436	33.7	33.7
2ª Dose Revacinação	1861	0.1	0.1
2º Reforço	140295	4.8	4.8
3ª Dose	5488	0.2	0.2
3ª Dose Revacinação	1026	0.0	0.0
3º Reforço	1584	0.1	0.1
4ª Dose	2302	0.1	0.1
4ª Dose Revacinação	1759	0.1	0.1
5ª Dose	1	0.0	0.0
5ª Dose Revacinação	12	0.0	0.0
Dose	2053	0.1	0.1
Dose Adicional	29219	1.0	1.0
Reforço	667893	23.0	23.0
Revacinação	619	0.0	0.0
Única	215	0.0	0.0
Total	2900200	100.0	100.0

Vamos analisar a seguir o cruzamento dos dados de qual vacina aplicada com a descrição da dose.

```
#tabela cruzada de qual dose e qual vacina
with(dados, table(qual_vacina, vacina_descricao_dose))
```

```
##          vacina_descricao_dose
## qual_vacina  1ª Dose 1ª Dose Revacinação 1º Reforço 2ª Dose
## AstraZeneca    3   63880             291         77   42018
## Bivalente      0      7             0         834        1
## Coronavac      0  308624             47         306  294475
```

```
## Janssen      0      49      114      25      46
## Pfizer      1 689756      1025      2398 641896
##          vacina_descricao_dose
## qual_vacina  2ª Dose Revacinação 2º Reforço 3ª Dose 3ª Dose Revacinação
## AstraZeneca      488      43057      251      234
## Bivalente      0      12      11      0
## Coronavac      25      8457      168      35
## Janssen      123      11284      17      136
## Pfizer      1225      77485      5041      621
##          vacina_descricao_dose
## qual_vacina  3º Reforço 4ª Dose 4ª Dose Revacinação 5ª Dose
## AstraZeneca      333      206      491      0
## Bivalente      0      0      0      0
## Coronavac      50      29      38      0
## Janssen      274      122      143      0
## Pfizer      927      1945      1087      1
##          vacina_descricao_dose
## qual_vacina  5ª Dose Revacinação Dose Dose Adicional Reforço Revacinação
## AstraZeneca      1      0      5329 98847      117
## Bivalente      0      0      6 97614      0
## Coronavac      1      0      1141 25610      20
## Janssen      0 2053      2958 58181      41
## Pfizer      10      0      19785 387641      441
##          vacina_descricao_dose
## qual_vacina  Única
## AstraZeneca      0
## Bivalente      9
## Coronavac      10
## Janssen      192
## Pfizer      4
```

Vamos fazer agora a junção de informações que se referem a mesma dose de vacina.

```
dados <- dados %>%
  mutate(
    num_dose = case_when(
      vacina_descricao_dose == "1ª Dose" ~ "1a dose",
```

```

vacina_descricao_dose == "2ª Dose" ~ "2a dose",
vacina_descricao_dose == "3ª Dose" ~ "3a dose",
vacina_descricao_dose == "4ª Dose" ~ "4a dose",
(vacina_descricao_dose == "1º Reforço" |
  vacina_descricao_dose == "Dose Adicional") |
  vacina_descricao_dose == "Reforço" ~ "Dose adicional + 1º reforço + reforço",
vacina_descricao_dose == "Dose" |
  vacina_descricao_dose == "Única" ~ "Dose+Única",
TRUE ~ as.character(vacina_descricao_dose)
)
)

```

*#tabela cruzada de qual dose e qual vacina*

```
with(dados, table(qual_vacina, num_dose))
```

```

##           num_dose
## qual_vacina      1a dose 1ª Dose Revacinação 2a dose 2ª Dose Revacinação
## AstraZeneca      3  63880                    291  42018                    488
## Bivalente        0    7                      0    1                      0
## Coronavac        0 308624                    47  294475                    25
## Janssen          0   49                     114   46                    123
## Pfizer           1 689756                    1025  641896                    1225
##           num_dose
## qual_vacina      2º Reforço 3a dose 3ª Dose Revacinação 3º Reforço 4a dose
## AstraZeneca      43057      251                    234      333      206
## Bivalente         12      11                      0        0        0
## Coronavac        8457      168                    35       50       29
## Janssen          11284      17                     136      274      122
## Pfizer           77485     5041                    621      927     1945
##           num_dose
## qual_vacina      4ª Dose Revacinação 5ª Dose 5ª Dose Revacinação
## AstraZeneca              491      0              1
## Bivalente                0      0              0
## Coronavac                38      0              1
## Janssen                  143      0              0
## Pfizer                   1087      1             10
##           num_dose

```

qual_vacina	Dose adicional + 1º reforço + reforço	Dose+Única	Revacinação
AstraZeneca	104253	0	117
Bivalente	98454	9	0
Coronavac	27057	10	20
Janssen	61164	2245	41
Pfizer	409824	4	441

```
dados_bruto <- dados %>%
  select(paciente_id, resid_muni = paciente_endereco_nmMunicipio,
         aplicacao_muni = estabelecimento_municipio_nome,
         dt_nasc, dt_aplic, dt_import, ano_aplic,
         mes_aplic, aplic_uf, gest_puerp, idade_anos,
         raca, resid_uf, qual_vacina, num_dose,
         sexo = paciente_enumSexoBiologico
  )

saveRDS(dados_bruto, "dados_vac_select_bruto.rds")
```

Há na base de dados 2900200 observações.

## 2.10 Sobre tempo entre aplicação e importação no RNDS

A variável `vacina_dataAplicacao` indica a data de aplicação da vacina e a variável `data_importacao_rnds` não está no dicionário, mas estamos entendendo como a data da importação na RNDS (Rede Nacional de Dados em Saúde). Vamos analisar o tempo entre a data da aplicação e a data da importação no sistema.

```
# tempo_rnds_aplic: tempo entre aplicação e importação no RNDS
dados <- dados %>%
  mutate(
    tempo_rnds_aplic = as.numeric(dt_import - dt_aplic),
  )
```

```
#tempos negativos (data a importação anterior à data da aplicação)
sum(dados$tempo_rnds_aplic<0)
```

```
## [1] 50168
```

Podemos observar que há 50168 casos inconsistentes, em que a data da importação é anterior à data da aplicação.

Segue a distribuição por estado de aplicação com relação ao indicador de tempo negativo:

```

dados <- dados %>%
  mutate(indic_tempo_neg = ifelse(
    is.na(tempo_rnds_aplic),
    NA,
    ifelse(tempo_rnds_aplic < 0, "sim", "não")
  ))

with(dados, ctable(aplic_uf, indic_tempo_neg, prop = "n"))

```

```
## Cross-Tabulation
```

```
## aplic_uf * indic_tempo_neg
```

```
## Data Frame: dados
```

```
##
```

```
## -----
```

	indic_tempo_neg	não	sim	Total
aplic_uf				
AC		7255	0	7255
AL		34106	0	34106
AM		59258	0	59258
AP		6724	0	6724
BA		201770	0	201770
CE		57434	38012	95446
DF		44692	0	44692
ES		88449	0	88449
GO		100849	0	100849
MA		83644	0	83644
MG		356141	10192	366333
MS		21055	0	21055
MT		60237	323	60560
PA		95372	0	95372
PB		71885	0	71885
PE		202015	0	202015
PI		54056	0	54056
PR		242236	1380	243616
RJ		212035	0	212035
RN		67444	0	67444



```
##          RO          29110          0          29110
##          RR          1617          0          1617
##          RS        164517        260        164777
##          SC          96383          1          96384
##          SE          35964          0          35964
##          SP        440304          0        440304
##          TO          15480          0          15480
##      Total        2850032        50168        2900200
## -----
```

E por UF de residência:

```
with(dados, ctable(resid_uf, indic_tempo_neg, prop="n"))
```

```
## Cross-Tabulation
## resid_uf * indic_tempo_neg
## Data Frame: dados
##
## -----
##          indic_tempo_neg      não      sim      Total
## resid_uf
##      AC          7706          0          7706
##      AL        36327          18        36345
##      AM        58457          20        58477
##      AP         6973           2         6975
##      BA       207101          68       207169
##      CE        61429       36620       98049
##      DF        45684          24       45708
##      ES        86185          24       86209
##      GO        99019          49       99068
##      MA        81161          30       81191
##      MG       355243       10004       365247
##      MS        22490          11       22501
##      MT        60921         338       61259
##      PA        95987          42       96029
##      PB        74498          54       74552
##      PE       196713          88       196801
##      PI        53654          60       53714
```

```
##          PR          234965    1395    236360
##          RJ          209423     176    209599
##          RN          67092      88     67180
##          RO          29366      13     29379
##          RR          2337        2     2339
##          RS          165267     288    165555
##          SC          100007      41    100048
##          SE          35696       11     35707
##          SP          430224     396    430620
##          TO          16710       10     16720
##          <NA>          9397      296     9693
##          Total        2850032   50168   2900200
## -----
```

Se filtrarmos apenas os casos com tempo positivo, temos as seguintes medidas descritivas do tempo de atraso de importação por UF de aplicação:

```
dados_tempo_pos <- dados %>%
  filter(tempo_rnds_aplic >= 0)

datasummary((aplic_uf) ~ tempo_rnds_aplic*(n+faltantes+media+DP+mediana+minimo+maximo),
  data = dados_tempo_pos, output = 'markdown')
```

aplic_uf	n	faltantes	media	DP	mediana	minimo	maximo
AC	7255.00	0.00	14.35	26.68	6.00	0.00	459.00
AL	34106.00	0.00	11.29	44.56	0.00	0.00	679.00
AM	59258.00	0.00	20.66	55.92	4.00	0.00	806.00
AP	6724.00	0.00	31.38	54.12	12.00	0.00	645.00
BA	201770.00	0.00	26.92	56.98	6.00	0.00	765.00
CE	57434.00	0.00	80.88	143.07	14.00	0.00	749.00
DF	44692.00	0.00	9.32	34.22	0.00	0.00	815.00
ES	88449.00	0.00	6.66	29.06	1.00	0.00	726.00
GO	100849.00	0.00	8.57	29.51	1.00	0.00	674.00
MA	83644.00	0.00	12.09	34.62	1.00	0.00	736.00
MG	356141.00	0.00	16.85	44.23	2.00	0.00	763.00
MS	21055.00	0.00	14.33	44.33	2.00	0.00	730.00
MT	60237.00	0.00	7.99	37.71	0.00	0.00	3295.00

aplic_uf	n	faltantes	media	DP	mediana	minimo	maximo
PA	95372.00	0.00	21.63	49.24	4.00	0.00	825.00
PB	71885.00	0.00	10.88	36.75	1.00	0.00	741.00
PE	202015.00	0.00	17.54	38.94	4.00	0.00	736.00
PI	54056.00	0.00	13.41	39.29	1.00	0.00	709.00
PR	242236.00	0.00	7.87	36.95	0.00	0.00	783.00
RJ	212035.00	0.00	18.16	46.43	0.00	0.00	852.00
RN	67444.00	0.00	8.99	33.64	1.00	0.00	740.00
RO	29110.00	0.00	7.61	33.61	0.00	0.00	702.00
RR	1617.00	0.00	103.09	92.32	105.00	0.00	398.00
RS	164517.00	0.00	7.74	28.64	0.00	0.00	673.00
SC	96383.00	0.00	11.19	47.52	0.00	0.00	678.00
SE	35964.00	0.00	30.72	67.53	2.00	0.00	685.00
SP	440304.00	0.00	17.52	57.81	0.00	0.00	743.00
TO	15480.00	0.00	6.68	31.37	0.00	0.00	705.00

E por UF de residência:

```
datasummary((resid_uf) ~ tempo_rnds_aplic*(n+faltantes+media+DP+mediana+minimo+maximo),
             data = dados_tempo_pos, output = 'markdown')
```

resid_uf	n	faltantes	media	DP	mediana	minimo	maximo
AC	7706.00	0.00	14.57	29.37	6.00	0.00	459.00
AL	36327.00	0.00	11.79	45.15	0.00	0.00	713.00
AM	58457.00	0.00	20.74	56.07	4.00	0.00	806.00
AP	6973.00	0.00	30.39	54.17	11.00	0.00	645.00
BA	207101.00	0.00	25.98	56.24	5.00	0.00	765.00
CE	61429.00	0.00	74.48	138.31	10.00	0.00	749.00
DF	45684.00	0.00	9.93	35.17	0.00	0.00	815.00
ES	86185.00	0.00	7.15	30.42	1.00	0.00	726.00
GO	99019.00	0.00	9.15	31.42	1.00	0.00	701.00
MA	81161.00	0.00	12.12	35.17	1.00	0.00	736.00
MG	355243.00	0.00	16.85	44.53	2.00	0.00	763.00
MS	22490.00	0.00	14.41	44.83	1.00	0.00	730.00
MT	60921.00	0.00	8.17	38.00	0.00	0.00	3295.00
PA	95987.00	0.00	21.14	48.78	4.00	0.00	825.00

resid_uf	n	faltantes	media	DP	mediana	minimo	maximo
PB	74498.00	0.00	11.19	37.24	1.00	0.00	741.00
PE	196713.00	0.00	17.63	39.78	4.00	0.00	736.00
PI	53654.00	0.00	13.62	39.97	1.00	0.00	709.00
PR	234965.00	0.00	8.12	37.51	0.00	0.00	783.00
RJ	209423.00	0.00	17.98	46.40	0.00	0.00	852.00
RN	67092.00	0.00	9.21	34.43	1.00	0.00	740.00
RO	29366.00	0.00	7.78	33.66	0.00	0.00	702.00
RR	2337.00	0.00	70.68	90.70	10.00	0.00	706.00
RS	165267.00	0.00	7.83	29.00	0.00	0.00	673.00
SC	100007.00	0.00	11.41	47.72	0.00	0.00	707.00
SE	35696.00	0.00	30.51	67.18	2.00	0.00	685.00
SP	430224.00	0.00	17.40	57.09	0.00	0.00	743.00
TO	16710.00	0.00	7.27	34.10	0.00	0.00	716.00

### 3 Exclusão dos casos inconsistentes

#### 3.1 Dados com status “entered-in-error”

Há na base de dados a variável `status` que tem as categorias “entered-in-error” e “final”. Quando `status` é “entered-in-error”, a variável `dt_deleted` informa uma data (caso contrário, o campo é vazio).

```
#tabela de frequência para status
questionr::freq(
  dados$status,
  cum = FALSE,
  total = TRUE,
  na.last = TRUE,
  valid = TRUE
) %>%
  kable(caption = "Tabela de frequências para status", digits = 2, format = "markdown")
```

Table 13: Tabela de frequências para status

	n	%	val%
entered-in-error	60316	2.1	2.1

	n	%	val%
final	2839884	97.9	97.9
Total	2900200	100.0	100.0

Vamos excluir os casos “entered-in-error” por entender que esses casos foram identificados com algum erro e não deveriam ser considerados na base de dados.

```
dados1 <- dados %>%
  filter(status != "entered-in-error")
```

### 3.2 ID do paciente que se repete para um mesmo número de dose e para o mesmo estabelecimento de saúde

Há na bases de dados a coluna `paciente_id`. Vamos agora identificar os casos duplicados com relação à combinação `paciente_id`, data de nascimento, `estabelecimento_valor` (pensando na hipótese de poder repetir o `paciente_id` entre os estabelecimentos) e `vacina_descricao_dose` (número da dose).

```
dados_dup_id <- janitor::get_dupes(dados1, paciente_id, dt_nasc, estabelecimento_valor, vacina_descricao_dose)
dim(dados_dup_id)[1]
```

```
## [1] 44628
```

Podemos observar que há 44628 observações duplicadas (observações que não são únicas). Vamos filtrar então só os casos únicos de combinação de `paciente_id`, data de nascimento, `estabelecimento_valor` e `vacina_descricao_dose` (número da dose).

```
dados2 <-
  dplyr::distinct(dados1,
                  paciente_id,
                  dt_nasc,
                  estabelecimento_valor,
                  vacina_descricao_dose,
                  .keep_all = TRUE)
```

Ficamos agora com 2817245 observações.

### 3.3 Ano de aplicação da vacina

Agora vamos analisar a data da aplicação da vacina.

```
# with(dados2, ctable(mes_aplic, ano_aplic, round.digits = 0))
```

Veja que há 53 casos com data de aplicação em 2020 e 1 em 2019.

No que segue vamos excluir esses casos de vacinação em 2019 e 2020.

```
# vamos excluir os casos de vacinação em 2020 e em 2019
dados3 <- dados2 %>%
  filter(ano_aplic %in% c(2021, 2022, 2023))
```

Ficamos agora com 2817189 observações.

A variável que indica o sexo biológico é `paciente_enumSexoBiologico`. Vamos avaliar a frequência desta variável.

```
#tabela de frequência para sexo
questionr::freq(
  dados3$paciente_enumSexoBiologico,
  cum = FALSE,
  total = TRUE,
  na.last = TRUE,
  valid = TRUE
) %>%
  kable(caption = "Tabela de frequências para variável
    sexo", digits = 2, format = "markdown")
```

Table 14: Tabela de frequências para variável sexo

	n	%	val%
	184	0	0
F	2788898	99	99
M	28107	1	1
Total	2817189	100	100

Note que há 28291 observações que não são sexo biológico feminino.

No que segue vamos excluir esses casos.

```
# vamos excluir os casos de sexo biológico não feminino
dados4 <- dados3 %>%
  filter(paciente_enumSexoBiologico == "F")
```

Ficamos agora com 2788898 observações.

A próxima seleção é avaliar a idade das gestantes.

```
#medidas descritiva da idade
datasummary((idade_anos) ~ 1 * (n + faltantes + media + DP + mediana + minimo +
                                maximo),
            data = dados4,
            output = 'markdown'
)
```

	n	faltantes	media	DP	mediana	minimo	maximo
idade_anos	2788898.00	0.00	28.51	6.71	28.00	0.00	114.00

Além da idade apresentada de forma quantitativa, podemos criar uma idade em categorias: até 9 anos, de 10 até 55 e maior que 55 anos.

```
dados4 <- dados4 %>%
  mutate(
    idade_cat = case_when(
      idade_anos < 10 ~ "< 10 anos",
      idade_anos >= 10 & idade_anos <= 55 ~ "de 10 a 55 anos",
      TRUE ~ "> 55 anos"
    )
  )
```

```
#tabela de frequência para categoria de idade
questionr::freq(
  dados4$idade_cat,
  cum = FALSE,
  total = TRUE,
  na.last = TRUE,
  valid = TRUE
) %>%
  kable(caption = "Tabela de frequências para categoria de idade", digits = 2, format = "markdown")
```

Table 16: Tabela de frequências para categoria de idade

	n	%	val%
< 10 anos	409	0.0	0.0
> 55 anos	4121	0.1	0.1
de 10 a 55 anos	2784368	99.8	99.8
Total	2788898	100.0	100.0

Vamos filtrar de 10 a 55 anos no que segue.

```
# vamos excluir os casos <10 e >55 anos
dados5 <- dados4 %>%
  filter(idade_anos >= 10 & idade_anos <= 55)
```

Com essas seleções ficamos com 2784368 observações.

```
#tabela de frequência para grupo de atendimento
questionr::freq(
  dados5$gest_puerp,
  cum = FALSE,
  total = TRUE,
  na.last = TRUE,
  valid = TRUE
) %>%
  kable(caption = "Tabela de frequências para grupo de atendimento", digits = 2, format = "markdown")
```

Table 17: Tabela de frequências para grupo de atendimento

	n	%	val%
gesta	2155305	77.4	77.4
puerp	629063	22.6	22.6
Total	2784368	100.0	100.0

Por fim, no total, excluímos 115832 observações que identificamos todas as inconsistências anteriormente descritas.

Apresentamos agora a distribuição das vacinas por UF de aplicação:



```
with(dados5, ctable(aplic_uf, num_dose, prop="n"))
```

```
## Cross-Tabulation
```

```
## aplic_uf * num_dose
```

```
## Data Frame: dados5
```

```
##
```

```
## -----
```

	num_dose	1a dose	1ª Dose Revacinação	2a dose	2ª Dose Revacinação	2º Reforço
aplic_uf						
AC	0	2629	11	2454	4	37
AL	0	13246	1	11610	5	90
AM	0	24292	23	16865	18	203
AP	0	2451	6	2051	6	20
BA	3	68552	166	62185	342	154
CE	0	37561	20	32269	42	34
DF	0	15273	20	14865	15	16
ES	0	33292	0	30387	0	39
GO	0	38396	54	34974	43	33
MA	0	32092	29	28691	22	43
MG	0	136553	211	126911	154	79
MS	0	8054	14	7321	7	4
MT	0	24896	59	22052	47	12
PA	0	35881	50	32417	62	41
PB	0	25712	38	23992	46	28
PE	1	79745	134	71378	62	26
PI	0	17280	21	16112	34	55
PR	0	86504	256	81407	399	69
RJ	0	73214	72	66952	107	137
RN	0	24574	0	20049	0	35
RO	0	12041	11	10643	18	8
RR	0	597	0	525	1	3
RS	0	60433	67	55430	179	74
SC	0	37126	42	34766	42	23
SE	0	13675	12	11397	5	23
SP	0	124605	0	124346	0	357
TO	0	6126	14	5583	2	1

```
##      Total          4    1034800          1331    947632          1662    1338
## -----
```

Apresentamos agora a distribuição das vacinas por UF de residência:

```
with(dados5, ctable(resid_uf, num_dose, prop="n"))
```

```
## Cross-Tabulation
```

```
## resid_uf * num_dose
```

```
## Data Frame: dados5
```

```
##
```

```
## -----
```

	num_dose	1a dose	1ª Dose Revacinação	2a dose	2ª Dose Revacinação	2º Reforço
AC	0	2818	11	2575	3	38
AL	0	14034	3	12340	7	104
AM	0	23830	23	16675	18	203
AP	0	2529	6	2148	6	21
BA	3	70062	162	63787	326	1579
CE	0	38370	22	32989	46	361
DF	0	16156	20	15194	15	161
ES	0	32360	1	29748	1	371
GO	0	37294	53	34398	48	331
MA	0	30991	31	27690	21	421
MG	0	135575	205	126213	155	841
MS	0	8566	15	7782	7	51
MT	0	25030	63	22319	44	121
PA	0	36220	49	32503	60	411
PB	0	26566	39	24876	49	291
PE	1	77154	121	69291	60	291
PI	0	17217	19	15993	28	541
PR	0	83870	255	78999	390	691
RJ	0	72489	72	66252	101	1351
RN	0	24465	1	20007	1	341
RO	0	12078	11	10725	18	91
RR	0	908	0	773	1	41
RS	0	60501	64	55707	179	751
SC	0	38639	45	35894	49	241

##	SE	0	13499	12	11311	6	23
##	SP	0	123566	11	122536	13	341
##	TO	0	6655	14	5972	4	2
##	<NA>	0	3358	3	2935	6	4
##	Total	4	1034800	1331	947632	1662	1338
##	-----						

```
dados6 <- dados5 %>%
  select(
    paciente_id,
    dt_nasc,
    dt_aplic,
    dt_import,
    ano_aplic,
    mes_aplic,
    gest_puerp,
    idade_anos,
    raca,
    resid_uf,
    resid_muni,
    paciente_endereco_nmMunicipio,
    qual_vacina,
    num_dose,
    aplic_uf,
    aplic_muni,
    aplicacao_muni = estabelecimento_municipio_nome
  )

saveRDS(dados6, file = "dados_vac_select.rds")
```