

# Análises do SIVEP-Gripe para Variantes Gestantes e Puérperas

Gestantes e puérperas

25/julho/2022

## 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/covid\\_gesta\\_puerp\\_br](https://observatorioobstetrico.shinyapps.io/covid_gesta_puerp_br).

```
#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",
    "writexl",
    "modelsummary"
  )
lapply(packages, loadlibrary)
```

A base de dados SIVEP-Gripe (Sistema de Informação da Vigilância Epidemiológica da Gripe) tem os registros dos casos e óbitos de SRAG (Síndrome Respiratória Aguda Grave). A notificação é compulsória para síndrome gripal (caracterizado por pelo menos dois dos seguintes sinais e sintomas: febre, mesmo que referida, calafrios, dor de garganta, dor de cabeça, tosse, coriza, distúrbios olfatórios ou de paladar) e que tem dispneia / desconforto respiratório ou pressão persistente no peito ou Saturação de O<sub>2</sub> menor que 95% no ar ambiente ou cor azulada dos lábios ou rosto. Indivíduos assintomáticos com confirmação laboratorial por biologia molecular ou exame imunológico para infecção por COVID-19 também são relatados.

Para notificações no Sivep-Gripe, os casos hospitalizados em hospitais públicos e privados e todas as mortes devido a infecções respiratórias agudas graves, independentemente da hospitalização, devem ser considerados.

A vigilância da SRAG no Brasil é desenvolvida pelo Ministério da Saúde (MS), por meio da Secretaria de Vigilância em Saúde (SVS), desde a pandemia de Influenza A (H1N1) em 2009. Mais informações em <https://coronavirus.saude.gov.br/definicao-de-caso-e-notificacao>.

Os dados do ano de 2020, 2021 e 2022 foram obtidos em 26/Janeiro/2022 no site <https://opendatasus.saude.gov.br>. Os dados de 2020, 2021 e 2022 são carregados e combinados abaixo:

```
##### carregando as bases de dados #####
ckanr::ckanr_setup("https://opendatasus.saude.gov.br")

arqs <- ckanr::package_search("srag 2020")$results %>%
  purrr::map("resources") %>%
  purrr::map(purrr::keep, ~ .x$mimetype == "text/csv") %>%
  purrr::map_chr(purrr::pluck, 1, "url")

arqs2 <- ckanr::package_search("srag 2021")$results %>%
  purrr::map("resources") %>%
  purrr::map(purrr::keep, ~ .x$mimetype == "text/csv") %>%
  purrr::map_chr(purrr::pluck, 2, "url")

dados_a <- fread(arqs[1], sep = ";")

dados_b <- fread(arqs[2], sep = ";")

dados_c <- fread(arqs2[1], sep = ";")

#### Concatenar dados 2020, 2021 e 2022 #####
dados_a <- dados_a %>%
  mutate(FATOR_RISC = case_when(FATOR_RISC == 1 ~ "S",
                                FATOR_RISC == 2 ~ "N"))

dados_b <- dados_b %>%
  mutate(FATOR_RISC = case_when(FATOR_RISC == 1 ~ "S",
                                FATOR_RISC == 2 ~ "N"))

dados_c <- dados_c %>%
  mutate(FATOR_RISC = case_when(FATOR_RISC == 1 ~ "S",
                                FATOR_RISC == 2 ~ "N"))

# COD_IDADE de 2022 para character para padronizar com 2020 e 2021

dados_c$COD_IDADE <- as.character(dados_c$COD_IDADE)

dados1 <- dados_a %>%
  full_join(dados_b) %>%
  full_join(dados_c)

#Criar variavel de ano do caso
dados1 <- dados1 %>%
  dplyr::mutate(
    dt_sint = as.Date(DT_SIN_PRI, format = "%d/%m/%Y"),
    dt_nasc = as.Date(DT_NASC, format = "%d/%m/%Y"),
    ano = lubridate::year(dt_sint),
  ) %>%
```

```
filter(
  dt_sint >= as.Date("16-02-2020", format = "%d-%m-%Y") &
  dt_sint <= as.Date("16-07-2022", format = "%d-%m-%Y")
)
```

Há atualmente 3296533 observações na base de dados e são as variáveis:

```
#names(dados1)
```

```
#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)
```

## Filtragem e tratamento dos dados

A variável que indica a classificação é a CLASSI\_FIN, com as seguintes categorias: 1-SRAG por influenza, 2-SRAG por outro vírus respiratório, 3-SRAG por outro agente etiológico, 4-SRAG não especificado e 5-SRAG por COVID-19.

```
#tabela de frequência para a classificação
```

```
questionr::freq(
  dados1$CLASSI_FIN,
  cum = FALSE,
  total = TRUE,
  na.last = FALSE,
  valid = FALSE
) %>%
  kable(caption = "Tabela de frequências para classificação do caso ",
        digits = 2)
```

Table 1: Tabela de frequências para classificação do caso

	n	%
1	22799	0.7
2	41644	1.3
3	11524	0.3
4	956508	29.0
5	2079318	63.1
NA	184740	5.6
Total	3296533	100.0

```
#codificar campo em branco para classificação como 9
dados1$CLASSI_FIN <-
  ifelse(is.na(dados1$CLASSI_FIN) == TRUE, 9, dados1$CLASSI_FIN)
```

Agora vamos filtrar os casos hospitalizados.

```
dados2 <- dados1 %>%
  filter(HOSPITAL == 1)
```

Note também que há 3118036 casos.

Casos de SRAG por COVID-19:

```
dados2 <- dados2 %>%
  filter(CLASSI_FIN == 5)
```

Note também que há 1989953 casos.

**Tipo de diagnóstico:**

```
#Caso diagnosticado por PCR
dados2 <- dados2 %>%
  mutate(pcr_SN = case_when(
    (PCR_SARS2 == 1) |
      (
        str_detect(DS_PCR_OUT, "SARS|COVID|COV|CORONA|CIVID")
      ) ~ "sim",
    TRUE ~ "não"
  ))
```

```
#Identificar se diagnóstico por sorologia
dados2$res_igg <-
  ifelse(is.na(dados2$RES_IGG) == TRUE, 0, dados2$RES_IGG)

dados2$res_igm <-
  ifelse(is.na(dados2$RES_IGM) == TRUE, 0, dados2$RES_IGM)

dados2$res_iga <-
  ifelse(is.na(dados2$RES_IGA) == TRUE, 0, dados2$RES_IGA)
```

```

dados2$sorologia_SN <-
  ifelse(dados2$res_igg == 1 |
        dados2$res_igm == 1 | dados2$res_iga == 1,
        "sim",
        "não")

#Identificar se diagnosticado por antígeno
dados2 <- dados2 %>%
  mutate(antigeno_SN = case_when(
    (AN_SARS2 == 1) | #positivo
    (
      str_detect(DS_AN_OUT, "SARS|COVID|COV|CORONA|CONA")
    ) ~ "sim",
    TRUE ~ "não"
  ))

```

A variável `classi_covid` identifica o tipo de diagnóstico. Essa variável é válida apenas para os casos confirmados de SRAG por COVID-19 (`CLASSI_FIN=5`).

```

#Criação da variável de classificação da covid-19
dados2 <- dados2 %>%
  mutate(
    classi_covid = case_when(
      CLASSI_FIN == 5 & pcr_SN == "sim" ~ "pcr",
      CLASSI_FIN == 5 & pcr_SN == "não" &
        antigeno_SN == "sim" ~ "antigenio",
      CLASSI_FIN == 5 & sorologia_SN == "sim" &
        antigeno_SN == "não" &
        pcr_SN == "não" ~ "sorologia",
      CLASSI_FIN != 5 ~ "não", #não é outro agente etiológico ou não especificado
      TRUE ~ "outro"
    )
  )

```

```

#tabela de frequências para tipo de diagnóstico
questionr::freq(
  dados2$classi_covid,
  cum = FALSE,
  total = TRUE,
  na.last = FALSE,
  valid = FALSE
) %>%
  kable(caption = "Tabela de frequências para o tipo de diagnóstico",
        digits = 2)

```

Table 2: Tabela de frequências para o tipo de diagnóstico

	n	%
antigenio	283440	14.2
outro	362628	18.2
pcr	1223248	61.5
sorologia	120637	6.1

	n	%
Total	1989953	100.0

```
dados2 <- dados2 %>%
  filter(classi_covid == "pcr")
```

Note também que há 1223248 casos.

O próximo passo é identificar as pessoas gestantes. Para isso, vamos analisar a variável `CS_GESTANT`. Essa variável assume os valores: 1-1º Trimestre; 2-2º Trimestre; 3-3º Trimestre; 4-Idade Gestacional Ignorada; 5-Não; 6-Não se aplica; 9-Ignorado.

```
#tabela de frequência para gestação
questionr::freq(
  dados2$CS_GESTANT,
  cum = FALSE,
  total = TRUE,
  na.last = FALSE,
  valid = FALSE
) %>%
  kable(caption = "Tabela de frequências para variável
    sobre gestação", digits = 2)
```

Table 3: Tabela de frequências para variável sobre gestação

	n	%
0	99	0.0
1	894	0.1
2	2567	0.2
3	6242	0.5
4	441	0.0
5	390714	31.9
6	764205	62.5
9	58086	4.7
Total	1223248	100.0

Há 99 casos com `CS_GESTANT=0`, em que a categoria 0 não tem código no dicionário.

Vamos ver se há alguma inconsistência ao analisar essa variável conjuntamente com `CS_SEXO` (F-feminino, M-masculino e I-ignorado).

```
#tabela de frequência para sexo
questionr::freq(
  dados2$CS_SEX0,
  cum = FALSE,
  total = TRUE,
  na.last = FALSE,
  valid = FALSE
) %>%
  kable(caption = "Tabela de frequências para sexo", digits = 2)
```

Table 4: Tabela de frequências para sexo

	n	%
F	543558	44.4
I	133	0.0
M	679557	55.6
Total	1223248	100.0

```
#tabela cruzada para gestação e sexo
table(dados2$CS_GESTANT, dados2$CS_SEXO)
```

```
##
##           F           I           M
## 0         29         40         30
## 1        894          0          0
## 2       2567          0          0
## 3       6242          0          0
## 4        441          0          0
## 5    389154         19    1541
## 6     86251         59 677895
## 9     57980         15         91
```

Veja que há 0 casos de CS\_SEXO=M com CS\_GESTANT=1,2,3 ou 4, como esperado.

A variável indicadora de puerpério é PUERPERA, com categorias 1-sim, 2-não e 9-Ignorado.

```
#tabela de frequencias para puerpério
questionr::freq(
  dados2$PUERPERA,
  cum = FALSE,
  total = TRUE,
  na.last = FALSE,
  valid = FALSE
) %>%
  kable(caption = "Tabela de frequências para variável indicadora de puérpera",
        digits = 2)
```

Table 5: Tabela de frequências para variável indicadora de puérpera

	n	%
1	3260	0.3
2	458878	37.5
9	11631	1.0
NA	749479	61.3
Total	1223248	100.0

```
#tabela cruzada de puerpério e sexo
table(dados2$PUERPERA, dados2$CS_SEXO)
```

```
##
```

```
##           F           I           M
##    1    3217           0           43
##    2 216451          34 242393
##    9   5318           5   6308
```

Veja que há 43 casos de CS\_SEX0=M com PUERPERA = 1 casos de puérpera e sexo masculino.

A próxima seleção é considerar só as pessoas do sexo feminino:

```
# Filtragem dos casos do sexo feminino
dados3 <- dados2 %>%
  filter(CS_SEX0 == "F")
```

Após essa seleção foram selecionados 543558 casos.

A próxima seleção é considerar gestantes e puérperas com idade maior que 10 e menor ou igual a 55 anos.

```
#criando a nossa variável de ano como a diferença entre dt_sint e dt_nasc.
#Nos casos sem dt_nasc, consideramos
#o campo NU_IDADE_N
dados3 <- dados3 %>%
  mutate(
    idade = as.period(interval(start = dt_nasc, end = dt_sint))$year,
    idade_anos = ifelse(is.na(idade), NU_IDADE_N, idade)
  )

# Filtragem dos casos com 55 anos ou menos
dados3 <- dados3 %>%
  filter(idade_anos > 9 & idade_anos <= 49)
```

Após essa seleção foram selecionados 149243 casos.

Agora vamos criar a variável de trimestre gestacional ou puerpério. Veja que para puerpério (**puerp**) são considerados os casos não gestante ou ignorado com PUERPERA = 1.

```
#Criação da variável classi_gesta_puerp para o momento gestacional ou puerpério
dados3 <- dados3 %>%
  mutate(
    classi_gesta_puerp = case_when(
      CS_GESTANT == 1 ~ "1tri",
      CS_GESTANT == 2 ~ "2tri",
      CS_GESTANT == 3 ~ "3tri",
      CS_GESTANT == 4 ~ "IG_ig",
      CS_GESTANT == 5 &
        PUERPERA == 1 ~ "puerp",
      CS_GESTANT == 9 & PUERPERA == 1 ~ "puerp",
      TRUE ~ "não"
    )
  )
```

A última filtragem consiste em selecionar os casos de gestantes ou puérperas.

```
#Seleção só dos casos de gestantes ou puérperas
dados4 <- dados3 %>%
  filter(classi_gesta_puerp != "não")
```



Ficamos com 12364 casos de gestantes e puérperas, distribuídas nos seguintes grupos de trimestre gestacional e puerpério:

```
#tabela de frequência para grupo gestacional
questionr::freq(
  dados4$classi_gesta_puerp,
  cum = FALSE,
  total = TRUE,
  na.last = FALSE,
  valid = FALSE
) %>%
  kable(caption = "Tabela de frequências para variável
    de trimestre gestacional ou puerpério", digits = 2)
```

Table 6: Tabela de frequências para variável de trimestre gestacional ou puerpério

	n	%
1tri	868	7.0
2tri	2517	20.4
3tri	6225	50.3
IG_ig	393	3.2
puerp	2361	19.1
Total	12364	100.0

```
dados5 <- dados4
```

Agora vamos criar a variável “variante”

```
#criando variável chamada variante(original,gama,delta,omicron)
in_gama <- as.Date("01-02-2021",format="%d-%m-%Y")
in_delta <- as.Date("01-08-2021",format="%d-%m-%Y")
in_ommi <- as.Date("01-01-2022",format="%d-%m-%Y")
dados5 <- dados5 %>%
  mutate(variante = case_when(dt_sint < in_gama ~ "original",
                              dt_sint >= in_gama & dt_sint < in_delta ~ "gama",
                              dt_sint >= in_delta & dt_sint < in_ommi ~ "delta",
                              dt_sint >= in_ommi ~ "omicron"))
```

```
dados5 <- dados5 %>%
  mutate(
    grupos = case_when(
      CS_GESTANT == 1 | CS_GESTANT == 2 | CS_GESTANT == 3 | CS_GESTANT == 4 ~ "gestante",
      CS_GESTANT == 5 &
        PUERPERA == 1 ~ "puerpera",
      CS_GESTANT == 9 & PUERPERA == 1 ~ "puerpera",
      TRUE ~ "não"
    )
  )

ctable(dados5$variante,dados5$grupos)
```

```
## Cross-Tabulation, Row Proportions
## variante * grupos
## Data Frame: dados5
##
## -----
##      grupos      gestante      puerpera      Total
## variante
##      delta      576 (83.4%)      115 (16.6%)      691 (100.0%)
##      gama      4603 (81.8%)      1025 (18.2%)      5628 (100.0%)
##      omicron      1020 (78.2%)      284 (21.8%)      1304 (100.0%)
##      original      3804 (80.2%)      937 (19.8%)      4741 (100.0%)
##      Total      10003 (80.9%)      2361 (19.1%)      12364 (100.0%)
## -----
```

No que segue tratamos as variáveis consideradas no Observatório Obstétrico Covid-19.

### Região do Brasil:

```
#Criação da variável de região
regions <- function(state) {
  southeast <- c("SP", "RJ", "ES", "MG")
  south <- c("PR", "SC", "RS")
  central <- c("GO", "MT", "MS", "DF")
  northeast <-
    c("AL", "BA", "CE", "MA", "PB", "PE", "PI", "RN", "SE")
  north <- c("AC", "AP", "AM", "PA", "RO", "RR", "TO")
  out <-
    ifelse(any(state == southeast),
           "southeast",
           ifelse(any(state == south),
                  "south",
                  ifelse(
                    any(state == central),
                    "central",
                    ifelse(any(state == northeast),
                           "northeast", "north")
                  )))
  return(out)
}

dados5$region <- sapply(dados5$SG_UF, regions)
dados5$region <-
  ifelse(is.na(dados5$region) == TRUE, 0, dados5$region)

saveRDS(dados5, "dados_amostragem.rds")
```

```
#tabela de frequências para região
questionr::freq(
  dados5$region,
  cum = FALSE,
  total = TRUE,
  na.last = FALSE,
  valid = FALSE
```

```
) %>%
  kable(caption = "Tabela de frequências para a região do Brasil", digits = 2)
```

Table 7: Tabela de frequências para a região do Brasil

	n	%
central	1590	12.9
north	782	6.3
northeast	2182	17.6
south	2352	19.0
southeast	5458	44.1
Total	12364	100.0

Veja que há 0 casos sem a informação da região do país (codificado como 0).

No que segue, tratamos as variáveis de caracterização, sintomas, comorbidades, desfechos e variáveis de tempo.

## Variáveis de caracterização

```
#Raça
dados5 <- dados5 %>%
  mutate(
    raca = case_when(
      CS_RACA == 1 ~ "branca",
      CS_RACA == 2 ~ "preta",
      CS_RACA == 3 ~ "amarela",
      CS_RACA == 4 ~ "parda",
      CS_RACA == 5 ~ "indigena",
      TRUE ~ NA_character_
    )
  )

#Escolaridade
dados5 <- dados5 %>%
  mutate(
    escol = case_when(
      CS_ESCOL_N == 0 ~ "sem escol",
      CS_ESCOL_N == 1 ~ "fund1",
      CS_ESCOL_N == 2 ~ "fund2",
      CS_ESCOL_N == 3 ~ "medio",
      CS_ESCOL_N == 4 ~ "superior",
      TRUE ~ NA_character_
    )
  )

#Faixa etária
dados5 <- dados5 %>%
  mutate(
    faixa_et = case_when(
```

```

    NU_IDADE_N <= 19 ~ "<20",
    NU_IDADE_N >= 20
    & NU_IDADE_N <= 34 ~ "20-34",
    NU_IDADE_N >= 35 ~ ">=35",
    TRUE ~ NA_character_
  )
)
dados5$faixa_et <-
  factor(dados5$faixa_et, levels = c("<20", "20-34", ">=35"))

#Internação no hospital
dados5 <- dados5 %>%
  mutate(hospital = case_when(HOSPITAL == 1 ~ "sim",
                              HOSPITAL == 2 ~ "não",
                              TRUE ~ NA_character_))

#Histórico de viagem
dados5 <- dados5 %>%
  mutate(hist_viagem = case_when(HISTO_VGM == 1 ~ "sim",
                                  HISTO_VGM == 2 ~ "não",
                                  TRUE ~ NA_character_))

#Síndrome gripal evoluída para SRAG
dados5 <- dados5 %>%
  mutate(sg_para_srag = case_when(SURTO_SG == 1 ~ "sim",
                                    SURTO_SG == 2 ~ "não",
                                    TRUE ~ NA_character_))

#Infecção adquirida no hospital
dados5 <- dados5 %>%
  mutate(inf_inter = case_when(NOSOCOMIAL == 1 ~ "sim",
                                NOSOCOMIAL == 2 ~ "não",
                                TRUE ~ NA_character_))

#Contato com ave ou suíno
dados5 <- dados5 %>%
  mutate(cont_ave_suino = case_when(AVE_SUINO == 1 ~ "sim",
                                      AVE_SUINO == 2 ~ "não",
                                      TRUE ~ NA_character_))

#Vacina para gripe
dados5 <- dados5 %>%
  mutate(vacina = case_when(VACINA == 1 ~ "sim",
                             VACINA == 2 ~ "não",
                             TRUE ~ NA_character_))

dados5 <- dados5 %>%
  mutate(vacina_cov = case_when(VACINA_COV == 1 ~ "sim",
                                 VACINA_COV == 2 ~ "não",
                                 variante == "original" ~ "não",
                                 TRUE ~ NA_character_))

#Antiviral

```

```

dados5 <- dados5 %>%
  mutate(
    antiviral = case_when(
      ANTIVIRAL == 1 ~ "Oseltamivir",
      ANTIVIRAL == 2 ~ "Zanamivir",
      TRUE ~ NA_character_
    )
  )

#Zona de residência
dados5 <- dados5 %>%
  mutate(zona = case_when(CS_ZONA == 1 ~ "urbana",
                           CS_ZONA == 2 ~ "rural",
                           CS_ZONA == 3 ~ "periurbana",
                           TRUE ~ NA_character_))

#Se mudança de município para atendimento
dados5 <- dados5 %>%
  mutate(mudou_muni = case_when((CO_MUN_RES == CO_MU_INTE) &
                                !is.na(CO_MU_INTE) &
                                !is.na(CO_MUN_RES) ~ "não",
                                (CO_MUN_RES != CO_MU_INTE) &
                                !is.na(CO_MU_INTE) &
                                !is.na(CO_MUN_RES) ~ "sim",
                                TRUE ~ NA_character_))

```

## Sintomas

```

#Febre
dados5 <- dados5 %>%
  mutate(febre = case_when(FEBRE == 1 ~ "sim",
                           FEBRE == 2 ~ "não",
                           TRUE ~ NA_character_))

#Tosse
dados5 <- dados5 %>%
  mutate(tosse = case_when(TOSSE == 1 ~ "sim",
                           TOSSE == 2 ~ "não",
                           TRUE ~ NA_character_))

#Garganta
dados5 <- dados5 %>%
  mutate(garganta = case_when(GARGANTA == 1 ~ "sim",
                              GARGANTA == 2 ~ "não",
                              TRUE ~ NA_character_))

#Dispneia
dados5 <- dados5 %>%
  mutate(dispnéia = case_when(DISPNEIA == 1 ~ "sim",
                              DISPNEIA == 2 ~ "não",

```

```

TRUE ~ NA_character_))

#Desconforto respiratório
dados5 <- dados5 %>%
  mutate(desc_resp = case_when(DESC_RESP == 1 ~ "sim",
                                DESC_RESP == 2 ~ "não",
                                TRUE ~ NA_character_))

#Saturação
dados5 <- dados5 %>%
  mutate(saturacao = case_when(SATURACAO == 1 ~ "sim",
                                SATURACAO == 2 ~ "não",
                                TRUE ~ NA_character_))

#Diarreia
dados5 <- dados5 %>%
  mutate(diarreia = case_when(DIARREIA == 1 ~ "sim",
                                DIARREIA == 2 ~ "não",
                                TRUE ~ NA_character_))

#Vômito
dados5 <- dados5 %>%
  mutate(vomito = case_when(VOMITO == 1 ~ "sim",
                              VOMITO == 2 ~ "não",
                              TRUE ~ NA_character_))

#Dor abdominal
dados5 <- dados5 %>%
  mutate(dor_abd = case_when(DOR_ABD == 1 ~ "sim",
                              DOR_ABD == 2 ~ "não",
                              TRUE ~ NA_character_))

#Fadiga
dados5 <- dados5 %>%
  mutate(fadiga = case_when(FADIGA == 1 ~ "sim",
                              FADIGA == 2 ~ "não",
                              TRUE ~ NA_character_))

#Perda olfativa
dados5 <- dados5 %>%
  mutate(perd_olft = case_when(PERD_OLFT == 1 ~ "sim",
                                PERD_OLFT == 2 ~ "não",
                                TRUE ~ NA_character_))

#Perda do paladar
dados5 <- dados5 %>%
  mutate(perd_pala = case_when(PERD_PALA == 1 ~ "sim",
                                PERD_PALA == 2 ~ "não",
                                TRUE ~ NA_character_))

```

## Comorbidades

```
#Cardiopatia
dados5 <- dados5 %>%
  mutate(cardiopati = case_when(CARDIOPATI == 1 ~ "sim",
                                CARDIOPATI == 2 ~ "não",
                                TRUE ~ NA_character_))

#Hematológica
dados5 <- dados5 %>%
  mutate(hematologi = case_when(HEMATOLOGI == 1 ~ "sim",
                                HEMATOLOGI == 2 ~ "não",
                                TRUE ~ NA_character_))

#Hepática
dados5 <- dados5 %>%
  mutate(hepatica = case_when(HEPATICA == 1 ~ "sim",
                              HEPATICA == 2 ~ "não",
                              TRUE ~ NA_character_))

#Asma
dados5 <- dados5 %>%
  mutate(asma = case_when(ASMA == 1 ~ "sim",
                          ASMA == 2 ~ "não",
                          TRUE ~ NA_character_))

#Diabetes
dados5 <- dados5 %>%
  mutate(diabetes = case_when(DIABETES == 1 ~ "sim",
                              DIABETES == 2 ~ "não",
                              TRUE ~ NA_character_))

#Neurológica
dados5 <- dados5 %>%
  mutate(neuro = case_when(NEUROLOGIC == 1 ~ "sim",
                           NEUROLOGIC == 2 ~ "não",
                           TRUE ~ NA_character_))

#Pneumopatia
dados5 <- dados5 %>%
  mutate(pneumopati = case_when(PNEUMOPATI == 1 ~ "sim",
                                 PNEUMOPATI == 2 ~ "não",
                                 TRUE ~ NA_character_))

#Imunossupressão
dados5 <- dados5 %>%
  mutate(imunodepre = case_when(IMUNODEPRE == 1 ~ "sim",
                                 IMUNODEPRE == 2 ~ "não",
                                 TRUE ~ NA_character_))

#Renal
dados5 <- dados5 %>%
  mutate(renal = case_when(RENAL == 1 ~ "sim",
```

```

        RENAL == 2 ~ "não",
        TRUE ~ NA_character_))

#Obesidade
dados5 <- dados5 %>%
  mutate(obesidade = case_when(OBESIDADE == 1 ~ "sim",
                                OBESIDADE == 2 ~ "não",
                                TRUE ~ NA_character_))

```

## Desfechos

```

#UTI
dados5 <- dados5 %>%
  mutate(uti = case_when(UTI == 1 ~ "sim",
                          UTI == 2 ~ "não",
                          TRUE ~ NA_character_))

#Suporte ventilatório
dados5 <- dados5 %>%
  mutate(
    suport_ven = case_when(
      SUPORT_VEN == 1 ~ "invasivo",
      SUPORT_VEN == 2 ~ "não invasivo",
      SUPORT_VEN == 3 ~ "não",
      TRUE ~ NA_character_
    )
  )

dados5$suport_ven <- factor(dados5$suport_ven,
                           levels = c("invasivo", "não invasivo", "não"))

dados5 <- dados5 %>%
  mutate(intubacao_SN = case_when(SUPORT_VEN == 1 ~ "sim",
                                   SUPORT_VEN == 2 ~ "não",
                                   SUPORT_VEN == 3 ~ "não",
                                   TRUE ~ NA_character_))

#Evolução
dados5 <-
  dados5 %>% mutate(
    evolucao = case_when(
      EVOLUCAO == 1 ~ "Cura",
      EVOLUCAO == 2 ~ "Óbito",
      EVOLUCAO == 3 ~ "Óbito",
      TRUE ~ NA_character_
    )
  )

#Classificação final
dados5 <-
  dados5 %>% mutate(
    classi_fin1 = case_when(
      CLASSI_FIN == 5 ~ "COVID-19",
      CLASSI_FIN == 1 ~ "Influenza",

```



```

CLASSI_FIN == 2 ~ "Outro vírus",
CLASSI_FIN == 3 ~ "Outro agente",
TRUE ~ NA_character_
)
)

```

```

#tabela de frequência para UTI
questionr::freq(
  dados5$uti,
  cum = FALSE,
  total = TRUE,
  na.last = FALSE,
  valid = FALSE
) %>%
  kable(caption = "Tabela de frequências para UTI",
        digits = 2)

```

Table 8: Tabela de frequências para UTI

	n	%
não	7963	64.4
sim	3619	29.3
NA	782	6.3
Total	12364	100.0

```

#tabela de frequência para suporte ventilatório
questionr::freq(
  dados5$suport_ven,
  cum = FALSE,
  total = TRUE,
  na.last = FALSE,
  valid = FALSE
) %>%
  kable(caption = "Tabela de frequências para
    suporte ventilatório",
        digits = 2)

```

Table 9: Tabela de frequências para suporte ventilatório

	n	%
invasivo	1698	13.7
não invasivo	4035	32.6
não	5467	44.2
NA	1164	9.4
Total	12364	100.0

```

#tabela de frequência para evolução
questionr::freq(
  dados5$evolucao,

```

```

cum = FALSE,
total = TRUE,
na.last = FALSE,
valid = FALSE
) %>%
kable(caption = "Tabela de frequências para evolução",
      digits = 2)

```

Table 10: Tabela de frequências para evolução

	n	%
Cura	10292	83.2
Obito	1252	10.1
NA	820	6.6
Total	12364	100.0

```

#tabela de frequência para classificação final
questionr::freq(
  dados5$classi_fin1,
  cum = FALSE,
  total = TRUE,
  na.last = FALSE,
  valid = FALSE
) %>%
kable(caption = "Tabela de frequências para classificação final",
      digits = 2)

```

Table 11: Tabela de frequências para classificação final

	n	%
COVID-19	12364	100
Total	12364	100

## Variáveis de tempo

Vamos criar as variáveis de tempo: tempo entre primeiros sintomas e internação, tempo entre primeiros sintomas e notificação, tempo de permanência na UTI e tempo entre primeiros sintomas e evolução. Para isso, vamos primeiro tratar as variáveis de data no que segue.

```

##### datas
#dt_notific - Data do preenchimento da ficha de notificação
#dt_sint - Data de 1ºs sintomas (deve ser menor que dt_notific)
#dt_interna - Data da internação por SRAG
#dt_pcr - Data do Resultado RT-PCR/outro método por Biologia Molecular
#dt_entuti - Data da entrada na UTI
#dt_saiduti - Data da saída da UTI
#dt_evoluca - Data da alta ou óbito
#dt_digita - Preenchido automaticamente pelo sistema com a data da digitação da ficha.
##Não é a data de preenchimento da ficha manualmente e sim a data em que é digitado no sistema.

```

```

dados5 <- dados5 %>%
  mutate(dt_notific = as.Date(DT_NOTIFIC, format = "%d/%m/%Y"),
         dt_sint = as.Date(DT_SIN_PRI, format = "%d/%m/%Y"),
         dt_interna = as.Date(DT_INTERNA, format = "%d/%m/%Y"),
         dt_pcr = as.Date(DT_PCR, format = "%d/%m/%Y"),
         dt_entuti = as.Date(DT_ENTUTI, format = "%d/%m/%Y"),
         dt_saiduti = as.Date(DT_SAIDUTI, format = "%d/%m/%Y"),
         dt_evoluca = as.Date(DT_EVOLUCA, format = "%d/%m/%Y"),
         dt_encerra = as.Date(DT_ENCERRA, format = "%d/%m/%Y"),
         dt_digita = as.Date(DT_DIGITA, format = "%d/%m/%Y"))

hoje <- Sys.Date() #data de hoje

#Arrumando as datas de internação inconsistentes.
#Quando for maior que a data de hoje e menor que a data dos primeiros sintomas é NA.
dados5 <- dados5 %>%
  mutate(dt_interna = case_when((dt_interna <= hoje &
                                dt_interna >= dt_sint) ~ dt_interna,
                                TRUE ~ NA_Date_
  ))

#Criando as variáveis de tempo a partir da diferença entre as datas.
# tempo_sintomas_hosp: tempo entre primeiros sintomas e internação.
# tempo_sintomas_notific: tempo entre primeiros sintomas e notificação.
# tempo_uti: tempo de permanência na UTI.
# tempo_sint_evolucao: tempo entre primeiros sintomas e evolução
dados5 <- dados5 %>%
  mutate(
    tempo_sintomas_hosp = as.numeric(dt_interna - dt_sint),
    tempo_sintomas_notific = as.numeric(dt_notific - dt_sint),
    tempo_uti = as.numeric(dt_saiduti - dt_entuti),
    tempo_sint_evolucao = as.numeric(dt_evoluca - dt_sint)
  )

```

## Exportando as bases de dados

```

dados6 <- dados5 %>%
  select(
    SEM_PRI,
    idade_anos,
    SG_UF,
    ID_MN_RESI,
    CO_MUN_RES,
    CO_MU_INTE,
    CLASSI_FIN,
    classi_fin1,
    DT_SIN_PRI,
    DT_EVOLUCA,
    dt_evoluca,
    dt_sint,

```

```
ano,  
classi_gesta_puerp,  
classi_covid,  
region,  
raca,  
escol,  
mudou_muni,  
zona,  
faixa_et,  
hospital,  
hist_viagem,  
sg_para_srag,  
inf_inter,  
cont_ave_suino,  
vacina,  
vacina_cov,  
antiviral,  
febre,  
tosse,  
garganta,  
dispneia,  
desc_resp,  
saturacao,  
diarreia,  
vomito,  
dor_abd,  
fadiga,  
perd_olft,  
perd_pala,  
cardiopati,  
hematologi,  
hepatica,  
asma,  
diabetes,  
neuro,  
pneumopati,  
imunodepre,  
renal,  
obesidade,  
uti,  
suport_ven,  
evolucao,  
dt_evoluca,  
tempo_sintomas_hosp,  
tempo_sintomas_notific,  
tempo_uti,  
tempo_sint_evolucao,  
DOSE_1_COV,  
DOSE_2_COV,  
grupos,  
variante  
)
```

*#Exportando*

```
# write_xlsx(dados6, "dados6.xlsx")  
saveRDS(dados6, "dados6.rds")
```