Aplicações práticas da manipulação de dados em linguagem R

Prof. Dr. Eder Angelo Milani

25/07/2024

O conjunto de dados

Nesta aula, utilizaremos um conjunto de dados real que apresenta uma grande quantidade de observações, para ilustrar como manipular de forma eficiente grandes conjuntos de dados.

Os dados são do Sistema de Informação sobre Mortalidade (SIM), do Ministério da Saúde. Este sistema, criado pelo Ministério da Saúde em 1975, analisa dados de mortalidade no Brasil, o que auxilia nas políticas públicas do país. A fonte de dados do SIM é a Declaração de Óbito (DO) que apresenta diversas variáveis para análise (por exemplo: idade, sexo e causa de óbito).

Os dados são públicos e estão disponíveis no site. O formato da base é em csv. Vamos utilizar o conjunto de dados do ano de 2022. Este conjunto está disponível para download na Plataforma Moodle Ipê, ou vocês podem fazer o download diretamente pelo site.

A seguir, vamos trabalhar alguns comandos para manipulação de conjunto de dados em linguagem R, utilizando também alguns conceitos trabalhados nas disciplinas iniciais do Curso.

```
# vamos verificar diretorio
getwd()
```

```
## [1] "G:/Meu Drive/UFG/Especializacao/Aula manipulacao"
```

```
# se precisar mudar o diretorio usar o comando setwd
# setwd("G:\\Meu Drive\\UFG\\Especializacao\\Aula manipulacao")

dataset <- read.csv("DO22OPEN.csv")

# verificando o tamanho da base
dim(dataset)
```

```
## [1] 1544266       1
```

```
#head(dataset)
```

Acho que algo deu errado, pois não era bem esse o conjunto de dados que estava esperando. Vamos abrir o conjunto de dados em um bloco de notas para ver como é o conjunto de dados.

Note que os valores/atributos estão separados por ";' ', com isso precisamos usar o comando *read.csv2* para fazer a leitura correta do conjunto de dados, ou usar o *read.csv* com *sep=";"*. A seguir também é calculado algumas informações sobre o conjunto de dados.

```r
# agora usando read.csv2
dataset <- read.csv2("DO22OPEN.csv")

# usando read.csv2
#dataset <- read.csv("DO22OPEN.csv", sep = ";")

dim(dataset)
```

```
## [1] 1544266      87
```

```r
#head(dataset)
```

Agora sim! Conjunto de dados carregado. A seguir verificaremos como estão definidas as variáveis de interesse. Note os tipos de variáveis!

```r
# vamos usar o comando str
str(dataset)
```

```
## 'data.frame':    1544266 obs. of  87 variables:
##  $ ORIGEM    : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ TIPOBITO  : int  2 2 2 2 2 2 2 2 2 2 ...
##  $ DTOBITO   : int  21042022 22042022 22042022 23042022 20042022 12042022 19042022 8042022 9042022 2
##  $ HORAOBITO : int  1230 1203 1525 937 1351 1420 1400 435 1115 1400 ...
##  $ NATURAL   : int  835 827 835 835 829 835 835 833 190 829 ...
##  $ CODMUNNATU: int  355030 270770 354850 350750 291400 352230 355030 330455 NA 290690 ...
##  $ DTNASC    : int  29041922 5071948 10101952 25061938 18101956 24111934 27101959 28041941 17021947 1
##  $ IDADE     : int  499 473 469 483 465 487 462 480 475 471 ...
##  $ SEXO      : int  2 2 1 1 2 1 2 1 1 2 ...
##  $ RACACOR   : int  1 1 2 1 1 1 1 1 1 2 ...
##  $ ESTCIV    : int  3 2 4 2 3 2 1 2 2 3 ...
##  $ ESC       : int  4 4 4 5 3 NA 2 2 3 1 ...
##  $ ESC2010   : int  3 2 3 5 2 NA 1 1 2 0 ...
##  $ SERIESCFAL: int  3 8 3 NA NA NA NA NA NA NA ...
##  $ OCUP      : int  999992 999992 516335 261125 999992 999993 512105 783220 782305 999993 ...
##  $ CODMUNRES : int  354850 354850 354850 354850 355100 355100 355100 355100 354330 291560 ...
##  $ LOCOCOR   : int  3 3 2 3 1 1 2 1 1 3 ...
##  $ CODESTAB  : int  NA NA 7872593 NA 2081636 2080354 9714138 2025752 2025752 NA ...
##  $ ESTABDESCR: logi  NA NA NA NA NA NA ...
##  $ CODMUNOCOR: int  354850 354850 354850 354850 354850 354850 354850 354850 354850 291560 ...
##  $ IDADEMAE  : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ ESCMAE    : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ ESCMAE2010: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ SERIESCMAE: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ OCUPMAE   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ QTDFILVIVO: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ QTDFILMORT: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ GRAVIDEZ  : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ SEMAGESTAC: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ GESTACAO  : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ PARTO     : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ OBITOPARTO: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ PESO      : int  NA NA NA NA NA NA NA NA NA NA ...
```

```
##  $ TPMORTEOCO: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ OBITOGRAV : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ OBITOPUERP: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ ASSISTMED : int  2 2 1 1 1 NA NA 1 NA 2 ...
##  $ EXAME     : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ CIRURGIA  : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ NECROPSIA : int  1 1 1 1 2 NA NA 2 NA 2 ...
##  $ LINHAA    : chr  "*I219" "*A419" "*A419" "*I269" ...
##  $ LINHAB    : chr  "*I709*I251" "*J180" "*J690" "*I802" ...
##  $ LINHAC    : chr  "*I10X" "*N390" "*N390" "*I739" ...
##  $ LINHAD    : chr  "*I119" "*I693" "*N133" "*G309" ...
##  $ LINHAII   : chr  "*R54X" "*K573*I709*E149*G319*R268" "*K253*F03X*I10X" "*G319*N390*R268*J180*F172
##  $ CAUSABAS  : chr  "I219" "I693" "N133" "I739" ...
##  $ CB_PRE    : logi  NA NA NA NA NA NA ...
##  $ COMUNSVOIM: int  354850 354850 354850 354850 NA NA NA NA NA NA ...
##  $ DTATESTADO: int  22042022 23042022 23042022 23042022 20042022 12042022 19042022 8042022 9042022 2
##  $ CIRCOBITO : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ ACIDTRAB  : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ FONTE     : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ NUMEROLOTE: int  20220025 20220025 20220025 20220025 20220025 20220025 20220025 20220025 20220025
##  $ TPPOS     : chr  "S" "S" "S" "S" ...
##  $ DTINVESTIG: int  2052022 2052022 2052022 2052022 NA NA NA NA NA NA ...
##  $ CAUSABAS_O: chr  "I219" "I693" "N133" "I739" ...
##  $ DTCADASTRO: int  29042022 29042022 29042022 29042022 29042022 29042022 29042022 2052022 2052022 6
##  $ ATESTANTE : int  4 4 4 4 1 1 1 5 2 5 ...
##  $ STCODIFICA: chr  "S" "S" "S" "S" ...
##  $ CODIFICADO: chr  "S" "S" "S" "S" ...
##  $ VERSAOSIST: chr  "3.2.30" "3.2.30" "3.2.30" "3.2.30" ...
##  $ VERSAOSCB : chr  "3.4" "3.4" "3.4" "3.4" ...
##  $ FONTEINV  : int  5 5 5 5 NA NA NA NA NA NA ...
##  $ DTRECEBIM : int  4052022 4052022 4052022 4052022 4052022 4052022 4052022 4052022 4052022 6052022
##  $ ATESTADO  : chr  "I219/I709 I251/I10/I119*R54" "A419/J180/N390/I693*K573 I709 E149 G319 R268" "A4
##  $ DTRECORIGA: int  4052022 4052022 4052022 4052022 4052022 4052022 4052022 4052022 4052022 6052022
##  $ CAUSAMAT  : chr  "" "" "" "" ...
##  $ ESCMAEAGR1: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ ESCFALAGR1: int  6 4 6 8 11 NA 10 10 11 0 ...
##  $ STDOEPIDEM: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ STDONOVA  : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ DIFDATA   : int  13 12 12 11 14 22 15 26 25 4 ...
##  $ NUDIASOBCO: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ NUDIASOBIN: logi  NA NA NA NA NA NA ...
##  $ DTCADINV  : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ TPOBITOCOR: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ DTCONINV  : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ FONTES    : chr  "" "" "" "" ...
##  $ TPRESGINFO: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ TPNIVELINV: chr  "" "" "" "" ...
##  $ NUDIASINF : logi  NA NA NA NA NA NA ...
##  $ DTCADINF  : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ MORTEPARTO: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ DTCONCASO : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ FONTESINF : logi  NA NA NA NA NA NA ...
##  $ ALTCAUSA  : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ CONTADOR  : int  1 2 3 4 5 6 7 8 9 10 ...
```

Note que são muitas as variáveis, vamos nos concentrar em algumas variáveis, para exemplificar o tratamento dos dados. As variáveis escolhidas estão detalhadas na Tabela 1. O dicionário pode ser acessoado aqui.

Table 1: Variáveis que serão manipuladas

Nome da variável	Descrição	Níveis
TIPOBITO	Tipo do óbitode	1 – óbito fetal 2 - óbito não fetal
DTOBITO	Data do óbito	ddmmaaaa
HORAOBITO	Horário do óbito	hhmm
DTNASC	Data do nascimento	ddmmaaaa
SEXO	Sexo	0 - Ignorado 1 – Masculino 2 – Feminino
ESC	Escolaridade em anos	1 – Nenhuma 2 – De 1 a 3 anos 3 – De 4 a 7 anos 4 – De 8 a 11 anos 5 – 12 anos e mais 9 – Ignorado
ESC2010	Escolaridade 2010	0 – Sem escolaridade 1 – Fundamental I (1^a a 4^a série) 2 – Fundamental II (5^a a 8^a série) 3 – Médio (antigo 2^o Grau) 4 – Superior incompleto 5 – Superior completo 9 – Ignorado
RACACOR	Raça/Cor	1 – Branca 2 – Preta 3 – Amarela 4 – Pardo 5 – Indígina

Agora preciso tomar uma descisão, se vou trabalhar no conjunto de dados, ou vou criar variáveis virtuais e então trabalhar nelas. Ou seja, vou usar *attach* ou não? Aqui eu não vou usar as variáveis virtuais.

A estratégia que vou seguir é filtrar o conjunto de dados nas variáveis que vou trabalhar, mas vou salvar em um outro *data.frame*.

```
#install.packages("dplyr")

# ja faz um bom tempo que instalou o pacote? Se sim, atualize o pacote
# update.packages("dplyr")
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.2.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# selecionando apenas as variaveis de interesse
dados_filtrados <- select(dataset, TIPOBITO, DTOBITO, DTNASC, HORAOBITO, SEXO, ESC, ESC2010, RACACOR)

# verificando a dimensao do conjunto de dados
dim(dados_filtrados)
```

```
## [1] 1544266       8
```

```
# visao geral do banco
glimpse(dados_filtrados) # funcao similar aos str mas do pacote dplyer
```

```
## Rows: 1,544,266
## Columns: 8
## $ TIPOBITO  <int> 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, ~
## $ DTOBITO   <int> 21042022, 22042022, 22042022, 23042022, 20042022, 12042022, ~
## $ DTNASC    <int> 29041922, 5071948, 10101952, 25061938, 18101956, 24111934, 2~
## $ HORAOBITO <int> 1230, 1203, 1525, 937, 1351, 1420, 1400, 435, 1115, 1400, 92~
## $ SEXO      <int> 2, 2, 1, 1, 2, 1, 2, 1, 1, 2, 1, 2, 2, 1, 1, 2, 2, 1, 1, 1, ~
## $ ESC       <int> 4, 4, 4, 5, 3, NA, 2, 2, 3, 1, 4, 9, 9, 2, 3, 4, 3, 3, 4, NA~
## $ ESC2010   <int> 3, 2, 3, 5, 2, NA, 1, 1, 2, 0, 3, 9, 9, 1, 1, 2, 1, 1, 3, NA~
## $ RACACOR   <int> 1, 1, 2, 1, 1, 1, 1, 1, 1, 2, 2, 4, 4, 2, 1, 4, 1, 1, 3, 1, ~
```

```
# como lidar com os NA's?
# excluir todas as linhas que tem pelo menos um NA
dados_filtrados2 <- dados_filtrados %>% na.omit()

# verificando a dimensao do conjunto de dados sem NA
dim(dados_filtrados2)
```

```
## [1] 1369345       8
```

```
# quantas linhas foram excluídas?
dim(dados_filtrados)[1] - dim(dados_filtrados2)[1]
```

```
## [1] 174921
```

```
# vou seguir a analise com o conjunto de dados COM os NA's
```

Agora com o conjunto de dados filtrado, vamos formatar as variáveis de interesse.

```
# modificando a variavel "Tipo de obito"

#head(dados_filtrados$TIPOBITO)

dados_filtrados$TIPOBITO <-  factor(dados_filtrados$TIPOBITO, levels = c(1, 2),
                                    labels = c("Fetal", "Nao fetal"))

is.factor(dados_filtrados$TIPOBITO)
```

```
## [1] TRUE
```

```
table(dados_filtrados$TIPOBITO)
```

```
##
##     Fetal Nao fetal
##         0   1544266
```

```
table(dados_filtrados$TIPOBITO, useNA = "always")
```

```
##
##     Fetal Nao fetal      <NA>
##         0   1544266         0
```

```
# observe que todos os preenchimentos sao iguais a 2,
# logo nao agrega informacao ao conjunto de dados
# vamos excluir essa variavel do conjunto de dados

dados_filtrados$TIPOBITO <- NULL

dim(dados_filtrados)
```

```
## [1] 1544266       7
```

```
# modificando a variavel "Data do obito"

# para trabalhar com datas, vamos utilizar o package lubridata
#install.packages("lubridate")
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.2.3
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
#head(dados_filtrados$DTOBITO)
dados_filtrados$DTOBITO <- dados_filtrados$DTOBITO %>% dmy()

class(dados_filtrados$DTOBITO)
```

```
## [1] "Date"
```

```
#head(dados_filtrados$DTOBITO)

# quantidade de obitos por mes
dados_filtrados$DTOBITO %>% month() %>% table() %>% prop.table()
```

```
## .
##          1          2          3          4          5          6          7
## 0.10757603 0.08675448 0.07862829 0.07575897 0.08497629 0.08786893 0.08907857
##          8          9         10         11         12
## 0.08167570 0.07681384 0.07784022 0.07449559 0.07853310
```

```
# mas quantos Na's tenho na variavel?
dados_filtrados$DTOBITO %>% is.na() %>% sum()
```

```
## [1] 0
```

```
# modificando a variavel "Data do nascimento"

#head(dados_filtrados$DTNASC)
dados_filtrados$DTNASC <- dados_filtrados$DTNASC %>% dmy()

class(dados_filtrados$DTNASC)
```

```
## [1] "Date"
```

```
#head(dados_filtrados$DTNASC)

# quantidade de nascimento por mes
dados_filtrados$DTNASC %>% month() %>% table() %>% prop.table()
```

```
## .
##          1          2          3          4          5          6          7
## 0.08629822 0.07417887 0.08405411 0.08076648 0.08795113 0.08686606 0.08445971
##          8          9         10         11         12
## 0.08601657 0.08532608 0.08408526 0.07929398 0.08070353
```

```
# mas quantos Na's tenho na variavel?
dados_filtrados$DTNASC %>% is.na() %>% sum()
```

```
## [1] 3342
```

```
# criando uma variavel idade no dia do obito

# tentativa 1
idade <- difftime(dados_filtrados$DTOBITO, dados_filtrados$DTNASC, units = "days")
# head(idade)

idade <- round(as.numeric(idade/365))
# head(idade)
# sera que esta certo?? Vamos conferir

dados_filtrados$DTNASC[1]
```

```
## [1] "1922-04-29"
```

```r
dados_filtrados$DTOBITO[1]
```

```
## [1] "2022-04-21"
```

```r
# algo de errado aqui!!! Esquecemos de algo??? Simmmm



intervalo <- interval(dados_filtrados$DTNASC, dados_filtrados$DTOBITO)
# head(intervalo)
# intervalo[1]

idade <- time_length(intervalo, "years")
# head(idade)

dados_filtrados$IDADE <- floor(idade) # pegar o maior interior menor que o numero
#head(dados_filtrados$IDADE)

summary(dados_filtrados$IDADE)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's
## -3171.00    57.00    71.00    66.96    83.00   131.00     3342
```

```r
# Como assim idade de -3171 e de 131 anos???

# qual linha esta o valo -3171
posicao <- which(dados_filtrados$IDADE == -3171)
posicao
```

```
## [1] 1445759
```

```r
dados_filtrados$IDADE[posicao]
```

```
## [1] -3171
```

```r
dados_filtrados$DTNASC[posicao]
```

```
## [1] "5193-08-27"
```

```r
dados_filtrados$DTOBITO[posicao]
```

```
## [1] "2022-11-28"
```

```r
# table(dados_filtrados$IDADE)

# idade de 120, 121, 122, 123 e 131 anos??

# substituir as idades < 0 e idades > 120 por NA's
```

```r
posicao <- which(dados_filtrados$IDADE == -3171 | dados_filtrados$IDADE > 120)

dados_filtrados$IDADE[posicao] <- NA

summary(dados_filtrados$IDADE)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##    0.00   57.00   71.00   66.97   83.00  120.00    3350
```

```r
# agora quero criar uma variavel idade categoria ordinal usando os seguintes intervalos
# 0-9 anos
# 10-19 anos
# 20-39 anos
# 40-49 anos
# 50-59 anos
# >=60 anos


idade_cat <- cut(dados_filtrados$IDADE, breaks = c(0, 10, 20, 40, 50, 60, 121), right = FALSE)
table(idade_cat)
```

```
## idade_cat
##   [0,10)  [10,20)  [20,40)  [40,50)  [50,60) [60,121)
##    41609    18484   115224   100843   167484  1097272
```

```r
dados_filtrados$IDADE_CAT <- cut(dados_filtrados$IDADE, breaks = c(0, 10, 20, 40, 50, 60, 121),
                 right = FALSE, labels= c("0-9", "10-19", "20-39", "40-49", "50-59", ">=60") )

table(dados_filtrados$IDADE_CAT)
```

```
##
##     0-9   10-19   20-39   40-49   50-59    >=60
##   41609   18484  115224  100843  167484 1097272
```

```r
# visualizacao do conjunto de dados apos a criacao das variaveis idade

str(dados_filtrados)
```

```
## 'data.frame':    1544266 obs. of  9 variables:
##  $ DTOBITO  : Date, format: "2022-04-21" "2022-04-22" ...
##  $ DTNASC   : Date, format: "1922-04-29" "1948-07-05" ...
##  $ HORAOBITO: int  1230 1203 1525 937 1351 1420 1400 435 1115 1400 ...
##  $ SEXO     : int  2 2 1 1 2 1 2 1 1 2 ...
##  $ ESC      : int  4 4 4 5 3 NA 2 2 3 1 ...
##  $ ESC2010  : int  3 2 3 5 2 NA 1 1 2 0 ...
##  $ RACACOR  : int  1 1 2 1 1 1 1 1 1 2 ...
##  $ IDADE    : num  99 73 69 83 65 87 62 80 75 71 ...
##  $ IDADE_CAT: Factor w/ 6 levels "0-9","10-19",..: 6 6 6 6 6 6 6 6 6 6 ...
```

```r
# modificando a variavel "Hora do obito"
# head(dados_filtrados$HORAOBITO)

# tentativa de modificar o formato
hm(dados_filtrados$HORAOBITO[1])
```

```
## Warning in .parse_hms(..., order = "HM", quiet = quiet): Some strings failed to
## parse
```

```
## [1] NA
```

```r
# note que nao funcionou pq nao esta no formato HH:MM
# veja um exemplo

hm("10:30")
```

```
## [1] "10H 30M 0S"
```

```r
# vamos usar o pacote stringr para tratar a variavel hora do obito
#install.packages("stringr")
library(stringr)

# padronizar a escrita do numero sempre com 4 casas, mesmo se comecar por 0
dados_filtrados$HORAOBITO <- sprintf('%04d', dados_filtrados$HORAOBITO)
#head(dados_filtrados$HORAOBITO)

# mudar a variavel para string
dados_filtrados$HORAOBITO <- dados_filtrados$HORAOBITO %>% as.character()
#head(dados_filtrados$HORAOBITO)

# incluir o : entre a hora e o minuto
dados_filtrados$HORAOBITO <- paste0(str_sub(dados_filtrados$HORAOBITO, 1, 2),
                                    ":", str_sub(dados_filtrados$HORAOBITO, 3, 4))
#head(dados_filtrados$HORAOBITO)

# se morre mais antes ou depois do almoco?
round(100*prop.table( table( ifelse( dados_filtrados$HORAOBITO < 12, "Antes das 12h", "Depois das 12h")
```

```
##
##  Antes das 12h Depois das 12h
##          50.86          49.14
```

```r
# modificando a variavel SEXO
# head(dados_filtrados$SEXO)
dados_filtrados$SEXO <- factor(dados_filtrados$SEXO, levels = c(0, 1, 2),
                               labels = c("Indefinido", "Masculino", "Feminino"))
head(dados_filtrados$SEXO)
```

```
## [1] Feminino  Feminino  Masculino Masculino Feminino  Masculino
## Levels: Indefinido Masculino Feminino
```

```r
prop.table( table(dados_filtrados$SEXO) )
```

```
##
##   Indefinido    Masculino     Feminino
## 0.0004053706 0.5471337192 0.4524609102
```

```r
# quero saber a quantidade de NA's
table(dados_filtrados$SEXO, useNA = "always")
```

```
##
## Indefinido  Masculino   Feminino       <NA>
##        626     844920     698720          0
```

```r
# modificando a variavel ESC - escolaridade
# head(dados_filtrados$ESC)
dados_filtrados$ESC <-  factor(dados_filtrados$ESC,
                               labels = c("Nenhuma", "de 1 a 3 anos", "de 4 a 7 anos",
                                          "de 8 a 11 anos" , " de 12 anos e mais", "Ignorado"),
                               ordered = T)

#head(dados_filtrados$ESC)
class(dados_filtrados$ESC)
```

```
## [1] "ordered" "factor"
```

```r
# modificando a variavel ESC2010 - escolaridade
#head(dados_filtrados$ESC2010)
dados_filtrados$ESC2010 <-  factor(dados_filtrados$ESC2010,
labels = c("Sem escolaridade", "Fundamental I (1ª a 4ª série)",
           "Fundamental II (5ª a 8ª série)", "Médio (antigo 2º Grau)",
           "Superior incompleto", "Superior completo", "Ignorado"),
ordered = T)

class(dados_filtrados$ESC2010)
```

```
## [1] "ordered" "factor"
```

```r
table(dados_filtrados$ESC2010)
```

```
##
##               Sem escolaridade  Fundamental I (1ª a 4ª série)
##                         244438                         488897
## Fundamental II (5ª a 8ª série)         Médio (antigo 2º Grau)
##                         229927                         215995
##            Superior incompleto              Superior completo
##                          15027                          80487
##                       Ignorado
##                         162899
```

```r
# quero mudar o nome das variaveis de escolaridade

dados_filtrados <- dados_filtrados %>%
                rename("ESCOLARIDADE" = "ESC", "ESCOLARIDADE_GRAUS" = "ESC2010")




# modificando a variavel RACA/COR
# head(dados_filtrados$RACACOR)
dados_filtrados$RACACOR <-  factor(dados_filtrados$RACACOR,
                                levels = c(1, 2, 3, 4, 5),
                                labels = c("Branca", "Preta", "Amarela",
                                           "Parda" , "indigina"))


table(dados_filtrados$RACACOR)
```

```
##
##   Branca    Preta  Amarela    Parda indigina
##   787363   131005     9261   582469     5343
```

```r
round(100*prop.table(table(dados_filtrados$RACACOR)), 2)
```

```
##
##   Branca    Preta  Amarela    Parda indigina
##    51.96     8.64     0.61    38.44     0.35
```

```r
# visualizacao do conjunto de dados apos manipulacao

str(dados_filtrados)
```

```
## 'data.frame':    1544266 obs. of  9 variables:
##  $ DTOBITO           : Date, format: "2022-04-21" "2022-04-22" ...
##  $ DTNASC            : Date, format: "1922-04-29" "1948-07-05" ...
##  $ HORAOBITO         : chr  "12:30" "12:03" "15:25" "09:37" ...
##  $ SEXO              : Factor w/ 3 levels "Indefinido","Masculino",..: 3 3 2 2 3 2 3 2 2 3 ...
##  $ ESCOLARIDADE      : Ord.factor w/ 6 levels "Nenhuma"<"de 1 a 3 anos"<..: 4 4 4 5 3 NA 2 2 3 1 ...
##  $ ESCOLARIDADE_GRAUS: Ord.factor w/ 7 levels "Sem escolaridade"<..: 4 3 4 6 3 NA 2 2 3 1 ...
##  $ RACACOR           : Factor w/ 5 levels "Branca","Preta",..: 1 1 2 1 1 1 1 1 1 2 ...
##  $ IDADE             : num  99 73 69 83 65 87 62 80 75 71 ...
##  $ IDADE_CAT         : Factor w/ 6 levels "0-9","10-19",..: 6 6 6 6 6 6 6 6 6 6 ...
```

```r
# usando a função group_by


df_summary <- dados_filtrados %>%
  group_by(SEXO) %>%
  summarise(
    Minimo = min(IDADE, na.rm = TRUE),
    Media_Idade = round(mean(IDADE, na.rm = TRUE)),
```

```
    Mediana = median(IDADE, na.rm = TRUE),
    Maximo = max(IDADE, na.rm = TRUE),
    Contagem = n()
  )

View(df_summary)


# exportando o conjunto de dados manipulado no formato csv

write.csv2(dados_filtrados, "Dados_SIM_2022.csv", row.names = F)
```