Data_gather

OOBr

2025-11-07

Load data from SIVEP-GRIPE health system to extract deaths data

```r
# Load required packages
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",
    "writexl",
    "janitor",
    "dplyr",
    "forcats",
    "stringr",
    "lubridate",
    "summarytools",
    "magrittr",
    "questionr",
    "knitr",
    "data.table",
    "writexl",
    "modelsummary"
  )
lapply(packages, loadlibrary)


memory.limit(999999)
```

Load SIVEP-Gripe data from OpenDataSUS (CKAN API)

```r
ckanr::ckanr_setup("https://opendatasus.saude.gov.br")

# Configure the CKAN client to use the OpenDataSUS portal.
```

```r
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")

# ----------------------------------------------------------------------
# Note:
# The commented-out section below shows how to read two files (e.g., 2020
# and 2021), recode the FATOR_RISC variable, and concatenate them.
# In the current version, only arqs[2] is used as the data source.
# ----------------------------------------------------------------------

# # Example: read first CSV file
# dados_a <- data.table::fread(arqs[1], sep = ";")
#
# # Example: read second CSV file
# dados_b <- data.table::fread(arqs[2], sep = ";")
#
# # Harmonize risk factor variable (1 = Yes, 2 = No)
# 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"
#   ))
#
# # Merge 2020 and 2021 records
# dadosa <- full_join(dados_a, dados_b)

# In this implementation, load only the second CSV file from the list.
dadosa <- data.table::fread(arqs[2], sep= ";")
```

Preprocessing: create date and year variables

```r
dados <-  dadosa %>%
  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 for confirmed COVID-19 cases
# ======================================================================

# Keep only records classified as COVID-19 according to the final classification.
# (CLASSI_FIN == 5 is the standard code for confirmed COVID-19 in SIVEP-Gripe.)
```

```r
dados1 <- dados %>%
    filter(CLASSI_FIN == 5)

# ----------------------------------------------------------------------
# Historical note (commented code):
# Originally, the analysis could be restricted to cases from the 8th
# epidemiological week of 2020 onwards or to a moving cutoff.
# Currently, we keep all confirmed COVID-19 cases in "dados1".
# ----------------------------------------------------------------------

# # Example of previous restriction by epidemiological week:
# sem <- 43 # current epidemiological week (example)
# # see: SINAN epidemiological calendar for reference
# dados2 <- dados1 %>%
#   filter((ano == 2020) | (ano == 2021 & SEM_PRI <= sem))

# For this version, no additional temporal restriction is applied.
dados2 <- dados1

# ======================================================================
# Restrict to female cases
# ======================================================================

# Keep only cases with female sex (CS_SEXO == "F").
dados3 <- dados2 %>%
  filter(CS_SEXO == "F")


# ======================================================================
# Classify gestational/puerperal status
# ======================================================================

# Create variable "classi_gesta_puerp" to represent gestational trimester
# or puerperium at the time of notification.
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"
    )
  )

# ======================================================================
# Select only pregnant or puerperal women
# ======================================================================
dados4 <- dados3 %>%
  filter(classi_gesta_puerp != "não")
```

```
# ======================================================================
# Construct age variable
# ======================================================================

# Create age in years using the difference between date of birth (dt_nasc)
# and date of symptom onset (dt_sint). When dt_nasc is missing, use
# NU_IDADE_N (numerical age variable from the dataset).
  dados4 <- dados4 %>%
    mutate(
      idade = as.period(interval(start = dt_nasc, end = dt_sint))$year,
      idade_anos = ifelse(is.na(idade), NU_IDADE_N, idade)
    )

# Filter cases to include women aged between 10 and 55 years (inclusive of 10, exclusive >55)
# This restricts to a biologically plausible reproductive age range.
  dados5 <- dados4 %>%
    filter(idade_anos > 9 & idade_anos <= 55)
```

Daily aggregated data for Brazil (all states combined)

```
# Total number of cases by date of symptom onset and epidemiological week.
dados_diarios <- dados5 %>%
  group_by(dt_sint, SEM_PRI) %>%
  summarise(n_casos = n())

# Number of deaths (EVOLUCAO == 2 or 3) by date and week.
# (Standard coding: 2 = death, 3 = death by other causes, depending on version.)
dados_diarios_obito <- dados5 %>%
  filter(EVOLUCAO == 2 | EVOLUCAO == 3) %>%
  group_by(dt_sint, SEM_PRI) %>%
  summarise(n_obitos = n())

# Number of cases with unknown or missing outcome (EVOLUCAO == 9 or NA).
dados_diarios_obito_na <- dados5 %>%
  filter(EVOLUCAO == 9 | is.na(EVOLUCAO)) %>%
  group_by(dt_sint, SEM_PRI) %>%
  summarise(n_obitos_na = n())

# Number of ICU admissions (UTI == 1) by date and week.
dados_diarios_uti <- dados5 %>%
  filter(UTI == 1) %>%
  group_by(dt_sint, SEM_PRI) %>%
  summarise(n_uti = n())

# Number of cases with unknown or missing ICU information (UTI == 9 or NA).
dados_diarios_uti_na <- dados5 %>%
  filter(UTI == 9 | is.na(UTI)) %>%
  group_by(dt_sint, SEM_PRI) %>%
  summarise(n_uti_na = n())

# Sequentially merge (full join) all daily aggregations into a single dataset.
```

```r
dados_conc <- full_join(dados_diarios, dados_diarios_uti , by = c("dt_sint", "SEM_PRI"))

dados_conc1 <- full_join(dados_conc, dados_diarios_uti_na, by = c("dt_sint", "SEM_PRI"))

dados_conc2 <- full_join(dados_conc1, dados_diarios_obito, by = c("dt_sint", "SEM_PRI"))

dados_conc3 <- full_join(dados_conc2, dados_diarios_obito_na, by = c("dt_sint", "SEM_PRI"))

# Replace all missing values with zero, assuming no events reported for that combination.
dados_conc3[is.na(dados_conc3)] <- 0

# Compute percentages:
# - porc_uti: proportion of ICU admissions among cases with known ICU status.
# - porc_obitos: proportion of deaths among cases with known outcome.
dados_conc3 <- dados_conc3 %>%
  mutate(porc_uti = (n_uti/(n_casos-n_uti_na))*100,
         porc_obitos = (n_obitos/(n_casos-n_obitos_na))*100)

# Export national-level aggregated dataset (pregnant and puerperal women).
# Filenames indicate extraction/processing date (01-02-22).
write_csv(dados_conc3, "dados_SIVEP_Gripe_Brasil_01-02-22.csv")
write_xlsx(dados_conc3, "dados_SIVEP_Gripe_Brasil_01-02-22.xlsx")
```

Daily aggregated data by Federative Unit (state-level)

```r
# Total number of cases by date, epidemiological week, and state (SG_UF).
dados_diarios <- dados5 %>%
  group_by(dt_sint, SEM_PRI, SG_UF) %>%
  summarise(n_casos = n())

# Number of deaths by date, week, and state.
dados_diarios_obito <- dados5 %>%
  filter(EVOLUCAO == 2 | EVOLUCAO == 3) %>%
  group_by(dt_sint, SEM_PRI, SG_UF) %>%
  summarise(n_obitos = n())

# Number of cases with unknown/missing outcome by date, week, and state.
dados_diarios_obito_na <- dados5 %>%
  filter(EVOLUCAO == 9 | is.na(EVOLUCAO)) %>%
  group_by(dt_sint, SEM_PRI, SG_UF) %>%
  summarise(n_obitos_na = n())

# Number of ICU admissions by date, week, and state.
dados_diarios_uti <- dados5 %>%
  filter(UTI == 1) %>%
  group_by(dt_sint, SEM_PRI, SG_UF) %>%
  summarise(n_uti = n())

# Number of cases with unknown/missing ICU information by date, week, and state.
dados_diarios_uti_na <- dados5 %>%
  filter(UTI == 9 | is.na(UTI)) %>%
  group_by(dt_sint, SEM_PRI, SG_UF) %>%
  summarise(n_uti_na = n())

# Merge all state-level aggregations.
dados_conc <- full_join(dados_diarios, dados_diarios_uti , by = c("dt_sint", "SEM_PRI", "SG_UF"))

dados_conc1 <- full_join(dados_conc, dados_diarios_uti_na, by = c("dt_sint", "SEM_PRI", "SG_UF"))

dados_conc2 <- full_join(dados_conc1, dados_diarios_obito, by = c("dt_sint", "SEM_PRI", "SG_UF"))

dados_conc3 <- full_join(dados_conc2, dados_diarios_obito_na, by = c("dt_sint", "SEM_PRI", "SG_UF"))

# Replace missing values with zero.
dados_conc3[is.na(dados_conc3)] <- 0

# Compute state-level ICU and mortality proportions among cases with known information.
dados_conc3 <- dados_conc3 %>%
  mutate(porc_uti = (n_uti/(n_casos-n_uti_na))*100,
         porc_obitos = (n_obitos/(n_casos-n_obitos_na))*100)

# Export state-level dataset.
write_csv(dados_conc3, "dados_SIVEP_Gripe_porUF_01-02-22.csv")
write_xlsx(dados_conc3, "dados_SIVEP_Gripe_porUF_01-02-22.xlsx")
```