lab-05

AUTHOR
Andre Gala-Garza

Disclaimer: Generative AI was used to assist with templating and writing code in this assignment; however, this code was checked manually and edited by hand to ensure accuracy.

Source: OpenAI. (2026). *ChatGPT (GPT-5.2 Thinking)* [Large language model]. https://chatgpt.com/.

1. Load packages

```
library("data.table")
library("dplyr")
```

```
Attaching package: 'dplyr'

The following objects are masked from 'package:data.table':

    between, first, last

The following objects are masked from 'package:stats':

    filter, lag

The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```
library("dtplyr")
library("tidyr")
```

2. Load population dataset

```
url <- 'https://raw.githubusercontent.com/dmcable/BIOSTAT620W26/main/data/covid/population.csv'
if (!file.exists("population.csv"))
  download.file(
    url = url,
    destfile = "population.csv",
    method   = "libcurl",
    timeout  = 60
    )
population <- read.csv('population.csv')
```

```
dim(population)
```

```
[1] 53  5
```

```
head(population)
```

```
  X       V1       V2           V3    V4
1 1 POP_2020 POP_2021         NAME state
2 2  3962031  3986639     Oklahoma    40
3 3  1961455  1963692     Nebraska    31
4 4  1451911  1441553       Hawaii    15
5 5   887099   895376 South Dakota    46
6 6  6920119  6975218    Tennessee    47
```

3. Examine and clean population matrix

```
# Remove redundant index column and state column
population$X <- NULL
population$V4 <- NULL
head(population)
```

```
        V1       V2           V3
1 POP_2020 POP_2021         NAME
2  3962031  3986639     Oklahoma
3  1961455  1963692     Nebraska
4  1451911  1441553       Hawaii
5   887099   895376 South Dakota
6  6920119  6975218    Tennessee
```

```
# Make the first row the header
library("janitor")
```

```
Attaching package: 'janitor'

The following objects are masked from 'package:stats':

    chisq.test, fisher.test
```

```
population <- population %>%
  row_to_names(row_number = 1)
head(population)
```

```
  POP_2020 POP_2021         NAME
2  3962031  3986639     Oklahoma
3  1961455  1963692     Nebraska
4  1451911  1441553       Hawaii
```

```
5   887099   895376 South Dakota
6  6920119  6975218    Tennessee
7  3114071  3143991       Nevada
```

```
# Give columns better names and types
# Make data tidy (make "year" and "pop" columns)
colnames(population) <- c("2020", "2021", "state_name")
population <- population %>%
  pivot_longer(
    cols = c("2020", "2021"),
    names_to = "year",
    values_to = "pop"
  ) %>%
  relocate(year, pop)

population$year <- as.integer(population$year)
population$pop <- as.integer(population$pop)
population <- as.data.frame(population)
rownames(population) <- 1:nrow(population)

head(population)
```

```
  year     pop state_name
1 2020 3962031   Oklahoma
2 2021 3986639   Oklahoma
3 2020 1961455   Nebraska
4 2021 1963692   Nebraska
5 2020 1451911     Hawaii
6 2021 1441553     Hawaii
```

```
# Create column for state abbreviations
state_lookup <- setNames(state.abb, state.name)
additional_states <- c("District of Columbia" = "DC",
                       "Puerto Rico" = "PR")
state_lookup <- c(state_lookup, additional_states)

population$state <- state_lookup[population$state_name]

head(population)
```

```
  year     pop state_name state
1 2020 3962031   Oklahoma    OK
2 2021 3986639   Oklahoma    OK
3 2020 1961455   Nebraska    NE
4 2021 1963692   Nebraska    NE
5 2020 1451911     Hawaii    HI
6 2021 1441553     Hawaii    HI
```

4. Make a regions dataframe

```
library(jsonlite)
url <- "https://github.com/datasciencelabs/2024/raw/refs/heads/main/data/regions.json"
regions <- fromJSON(url) # use fromJSON to read as a data.frame

dim(regions)
```

```
[1] 10  3
```

```
regions
```

```
   region                                        region_name
1       1                                        New England
2       2 New York and New Jersey, Puerto Rico, Virgin Islands
3       3                                       Mid-Atlantic
4       4                                          Southeast
5       5                                            Midwest
6       6                                      South Central
7       7                                     Central Plains
8       8                                    Mountain States
9       9                                            Pacific
10     10                                  Pacific Northwest

states
1
Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island, Vermont
2
New Jersey, New York, Puerto Rico, Virgin Islands
3
Delaware, District of Columbia, Maryland, Pennsylvania, Virginia, West Virginia
4                                                                                Alabama,
Florida, Georgia, Kentucky, Mississippi, North Carolina, South Carolina, Tennessee
5
Illinois, Indiana, Michigan, Minnesota, Ohio, Wisconsin
6
Arkansas, Louisiana, New Mexico, Oklahoma, Texas
7
Iowa, Kansas, Missouri, Nebraska
8
Colorado, Montana, North Dakota, South Dakota, Utah, Wyoming
9  Arizona, California, Hawaii, Nevada, American Samoa, Commonwealth of the Northern Mariana
Islands, Federated States of Micronesia, Guam, Marshall Islands, Republic of Palau
10
Alaska, Idaho, Oregon, Washington
```

```
# Separate the lists of states to their own rows
# Make the "region" column a factor
regions <- unnest(regions, c(region, states), keep_empty = FALSE)
```

```
regions <- rename(regions, state_name = states)
regions$region <- as.factor(regions$region)
head(regions)
```

```
# A tibble: 6 × 3
  region region_name state_name
  <fct>  <chr>       <chr>
1 1      New England Connecticut
2 1      New England Maine
3 1      New England Massachusetts
4 1      New England New Hampshire
5 1      New England Rhode Island
6 1      New England Vermont
```

```
# Rename value in region_name with a long name
unique(regions$region_name)
```

```
 [1] "New England"
 [2] "New York and New Jersey, Puerto Rico, Virgin Islands"
 [3] "Mid-Atlantic"
 [4] "Southeast"
 [5] "Midwest"
 [6] "South Central"
 [7] "Central Plains"
 [8] "Mountain States"
 [9] "Pacific"
[10] "Pacific Northwest"
```

```
regions$region_name[regions$region_name == "New York and New Jersey, Puerto Rico, Virgin Islands"]
regions %>% filter(region_name == "NY, NJ, PR, VI")
```

```
# A tibble: 4 × 3
  region region_name    state_name
  <fct>  <chr>          <chr>
1 2      NY, NJ, PR, VI New Jersey
2 2      NY, NJ, PR, VI New York
3 2      NY, NJ, PR, VI Puerto Rico
4 2      NY, NJ, PR, VI Virgin Islands
```

5. Merge regions dataframe with population dataframe

```
population <- left_join(population, regions, by = "state_name")
head(population)
```

```
  year     pop state_name state region    region_name
1 2020 3962031   Oklahoma    OK      6  South Central
```

```
2 2021 3986639   Oklahoma    OK      6  South Central
3 2020 1961455   Nebraska    NE      7 Central Plains
4 2021 1963692   Nebraska    NE      7 Central Plains
5 2020 1451911     Hawaii    HI      9        Pacific
6 2021 1441553     Hawaii    HI      9        Pacific
```

6. Download state-level SARS-COV2 data

```
library(httr2)
api <- "https://data.cdc.gov/resource/pwn4-m3yp.json"
request <- request(paste0(api,paste0("?$limit=10000000000")))
response <- request |> req_perform() |> resp_body_string()
cases_raw <- fromJSON(response)
head(cases_raw)
```

```
            date_updated state              start_date                end_date
1 2023-02-23T00:00:00.000    AZ 2023-02-16T00:00:00.000 2023-02-22T00:00:00.000
2 2022-12-22T00:00:00.000    LA 2022-12-15T00:00:00.000 2022-12-21T00:00:00.000
3 2023-02-23T00:00:00.000    GA 2023-02-16T00:00:00.000 2023-02-22T00:00:00.000
4 2023-03-30T00:00:00.000    LA 2023-03-23T00:00:00.000 2023-03-29T00:00:00.000
5 2023-02-02T00:00:00.000    LA 2023-01-26T00:00:00.000 2023-02-01T00:00:00.000
6 2023-03-23T00:00:00.000    LA 2023-03-16T00:00:00.000 2023-03-22T00:00:00.000
  tot_cases new_cases tot_deaths new_deaths new_historic_cases
1 2434631.0    3716.0    33042.0       39.0              23150
2 1507707.0    4041.0    18345.0       21.0              21397
3 3061141.0    5298.0    42324.0       88.0               6800
4 1588259.0    2203.0    18858.0       23.0               5347
5 1548508.0    5725.0    18572.0       47.0               4507
6 1580709.0    1961.0    18835.0       35.0               2239
  new_historic_deaths
1                   0
2                   0
3                   0
4                   0
5                   0
6                   0
```

```
dim(cases_raw)
```

```
[1] 10380    10
```

The string "?$limit=10000000000" is an argument to the request named "limit" that specifies how many results to obtain from the JSON endpoint. By setting this limit to an extremely large number, the maximum possible number of rows from the dataset will be obtained; however, if we remove the limit argument, it is set to the default number of results on one page, which is 1,000. We can see that we obtained 10,380 records by setting the limit argument, which is actually more records than if we did not set a limit.

7. Wrangle the cases dataset

```
colnames(cases_raw)
```

```
 [1] "date_updated"        "state"               "start_date"
 [4] "end_date"            "tot_cases"           "new_cases"
 [7] "tot_deaths"          "new_deaths"          "new_historic_cases"
[10] "new_historic_deaths"
```

```
# Select only the necessary columns
cases_clean <- cases_raw %>%
  select("state", "end_date", "tot_cases")
names(cases_clean) <- c("state", "date", "cases")
head(cases_clean)
```

```
  state                   date     cases
1    AZ 2023-02-22T00:00:00.000 2434631.0
2    LA 2022-12-21T00:00:00.000 1507707.0
3    GA 2023-02-22T00:00:00.000 3061141.0
4    LA 2023-03-29T00:00:00.000 1588259.0
5    LA 2023-02-01T00:00:00.000 1548508.0
6    LA 2023-03-22T00:00:00.000 1580709.0
```

```
# Give columns better types
library(lubridate)
```

```
Attaching package: 'lubridate'

The following objects are masked from 'package:data.table':

    hour, isoweek, mday, minute, month, quarter, second, wday, week,
    yday, year

The following objects are masked from 'package:base':

    date, intersect, setdiff, union
```

```
cases_clean$date <- as.Date(ymd_hms(cases_clean$date))
cases_clean$cases <- as.numeric(cases_clean$cases)
head(cases_clean)
```

```
  state       date   cases
1    AZ 2023-02-22 2434631
2    LA 2022-12-21 1507707
3    GA 2023-02-22 3061141
4    LA 2023-03-29 1588259
```

```
5    LA 2023-02-01 1548508
6    LA 2023-03-22 1580709
```

8. Additional data wrangling

```
# Merge population and cases dataframes
cases_clean$year <- year(cases_clean$date)
population <- left_join(population, cases_clean, by = c("state", "year"))
head(population)
```

```
  year     pop state_name state region   region_name       date cases
1 2020 3962031   Oklahoma    OK      6 South Central 2020-01-22     0
2 2020 3962031   Oklahoma    OK      6 South Central 2020-01-29     0
3 2020 3962031   Oklahoma    OK      6 South Central 2020-02-05     0
4 2020 3962031   Oklahoma    OK      6 South Central 2020-02-12     0
5 2020 3962031   Oklahoma    OK      6 South Central 2020-02-19     0
6 2020 3962031   Oklahoma    OK      6 South Central 2020-02-26     0
```

```
tail(population)
```

```
     year    pop state_name state region       region_name       date  cases
5299 2021 732673     Alaska    AK     10 Pacific Northwest 2021-11-24 144792
5300 2021 732673     Alaska    AK     10 Pacific Northwest 2021-12-01 146508
5301 2021 732673     Alaska    AK     10 Pacific Northwest 2021-12-08 148018
5302 2021 732673     Alaska    AK     10 Pacific Northwest 2021-12-15 149237
5303 2021 732673     Alaska    AK     10 Pacific Northwest 2021-12-22 150283
5304 2021 732673     Alaska    AK     10 Pacific Northwest 2021-12-29 152363
```

```
# Sort population dataframe by state and by date within each state
population <- population[order(population$state, population$date), ]
head(population)
```

```
     year    pop state_name state region       region_name       date cases
5252 2020 732441     Alaska    AK     10 Pacific Northwest 2020-01-22     0
5235 2020 732441     Alaska    AK     10 Pacific Northwest 2020-01-29     0
5236 2020 732441     Alaska    AK     10 Pacific Northwest 2020-02-05     0
5237 2020 732441     Alaska    AK     10 Pacific Northwest 2020-02-12     0
5238 2020 732441     Alaska    AK     10 Pacific Northwest 2020-02-19     0
5239 2020 732441     Alaska    AK     10 Pacific Northwest 2020-02-26     0
```

```
tail(population)
```

```
     year    pop state_name state region     region_name       date  cases
1933 2021 578803    Wyoming    WY      8 Mountain States 2021-11-24 110047
1934 2021 578803    Wyoming    WY      8 Mountain States 2021-12-01 111160
1935 2021 578803    Wyoming    WY      8 Mountain States 2021-12-08 112348
1936 2021 578803    Wyoming    WY      8 Mountain States 2021-12-15 113159
```

```
1937 2021 578803    Wyoming    WY      8 Mountain States 2021-12-22 113902
1938 2021 578803    Wyoming    WY      8 Mountain States 2021-12-29 115215
```