Milestone #2

Group 22: Jaemie Anne Abad, Vig Karthik, Kathy LeBert

2023-10-01

#Description of dataset *What is the data source? (1-2 sentences on where the data is coming from, dates included, etc.)*

The major data are from California Office of Environmental Health Hazard Assessment's CalEnviroScreen 4.0 (CES) (published 2021, data averaged from 2017 to 2019). The asthma emergency department visit rates is from California Health and Human Services Open Data Portal (created 2019, updated 2023).

How does the dataset relate to the group problem statement and question?

The group's problem statement and question looks at the correlation between each county's ED visits due to asthma and certain CalEnviroScreen scores. To answer this question, the datasets needed are the rate ED visits and each of the CES measures - which are the ones provided.

```r
# update.packages()

library(data.table)
library(readxl)
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.3     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.3     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.0
## ✔ purrr     1.0.2     
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::between()     masks data.table::between()
## ✖ dplyr::filter()      masks stats::filter()
## ✖ dplyr::first()       masks data.table::first()
## ✖ lubridate::hour()    masks data.table::hour()
## ✖ lubridate::isoweek() masks data.table::isoweek()
## ✖ dplyr::lag()         masks stats::lag()
## ✖ dplyr::last()        masks data.table::last()
## ✖ lubridate::mday()    masks data.table::mday()
## ✖ lubridate::minute()  masks data.table::minute()
## ✖ lubridate::month()   masks data.table::month()
## ✖ lubridate::quarter() masks data.table::quarter()
## ✖ lubridate::second()  masks data.table::second()
## ✖ purrr::transpose()   masks data.table::transpose()
## ✖ lubridate::wday()    masks data.table::wday()
## ✖ lubridate::week()    masks data.table::week()
## ✖ lubridate::yday()    masks data.table::yday()
## ✖ lubridate::year()    masks data.table::year()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become
errors
```

```
#loading the data.table library to read in the csv
#also loaded read_xl to read the xlsx file

data_dictionary <- read_xlsx("calenviroscreen_datadictionary.xlsx")
```

```
## New names:
## • `` -> `...2`
```

```
#using read_xlsx to read in the data dictionary

measures_raw <- fread("calenviroscreen_measures_2021.csv", encoding = "UTF-8")
#using fread() from the data.table package to read the measures csv.
#using fread() so special characters read in properly

scores_raw <- fread("calenviroscreen_scores_demog_2021.csv", encoding = "UTF-8")
#using fread() to read the demographics csv

asthma_raw <- fread("chhs_asthma_ed.csv")
#using fread() to read the asthma csv.

#now that we've read the files, I've went into the csv files and removed
#periods by some abbreviations to avoid '._' string chunks in columns
#this is mostly for aesthetic and typing ease during the project

#the below functions take the raw csvs and convert all columns to lowercase
#they also remove spaces in the column names to convert columns to snakecase

measures <- rename_with(
  measures_raw,             # data frame
  ~ tolower(           # function call (using tolower())
    gsub(" ",          # embedded gsub() checking for empty spaces pattern " "
         "_",          # replace pattern with underscore "_"
         .x,
         fixed = TRUE)
    ))

scores <- rename_with(
  scores_raw,             # data frame
  ~ tolower(           # function call (using tolower())
    gsub(" ",          # embedded gsub() checking for empty spaces pattern " "
         "_",          # replace pattern with underscore "_"
         .x,
         fixed = TRUE)
    ))


asthma <- rename_with(
  asthma_raw,             # data frame
  ~ tolower(           # function call (using tolower())
    gsub(" ",          # embedded gsub() checking for empty spaces pattern " "
         "_",         # replace pattern with underscore "_"
         .x,
         fixed = TRUE)
    ))

str(measures)
```

```
## Classes 'data.table' and 'data.frame':   8035 obs. of  50 variables:
##  $ census_tract              :integer64 6019001100 6077000700 6037204920 6019000700 6019000200
6037542402 6019001000 6037543202 ...
##  $ california_county         : chr  "Fresno" "San Joaquin" "Los Angeles" "Fresno" ...
##  $ ozone                     : num  0.0603 0.0459 0.0479 0.0603 0.0603 ...
##  $ ozone_pctl                : num  82.5 45 53.7 82.5 82.5 ...
##  $ pm2.5                     : num  13.9 11.9 12.3 13.5 13.8 ...
##  $ pm2.5_pctl                : num  97.7 72.6 89.2 95.9 97.5 ...
##  $ diesel_pm                 : num  1.123 0.538 0.781 0.174 1.39 ...
##  $ diesel_pm_pctl            : num  98.7 91.2 96.6 57.1 99.3 ...
##  $ drinking_water            : num  734 390 788 734 734 ...
##  $ drinking_water_pctl       : num  84.4 41.6 92.5 84.4 84.4 ...
##  $ lead                      : num  89.6 77.3 92.6 68.4 75.4 ...
##  $ lead_pctl                 : num  96.5 86.8 98.4 77 85.1 ...
##  $ pesticides                : num  1 63.1 0 44.6 16.6 ...
##  $ pesticides_pctl           : num  42.9 73.7 0 71.6 64.4 ...
##  $ tox_release               : num  4859 520 3683 1630 1975 ...
##  $ tox_release_pctl          : num  92.2 52.4 87.7 74.9 79 ...
##  $ traffic                   : num  1037 856 2523 691 910 ...
##  $ traffic_pctl              : num  60.4 48.3 92.8 35.3 52.3 ...
##  $ cleanup_sites             : num  70.5 61.9 38.8 16.5 10.5 ...
##  $ cleanup_sites_pctl        : num  98.2 97.5 93 77.3 62.4 ...
##  $ groundwater_threats       : num  54.2 78.6 20.5 9.5 28.2 ...
##  $ groundwater_threats_pctl  : num  91.2 95.1 68.9 44.8 78.1 ...
##  $ haz_waste                 : num  3.1 1.27 11.62 2.36 0.35 ...
##  $ haz_waste_pctl            : num  96.3 88.6 99.7 94.1 56.4 ...
##  $ imp_water_bodies          : int  0 13 7 0 0 10 0 7 0 2 ...
##  $ imp_water_bodies_pctl     : num  0 91.9 66.7 0 0 ...
##  $ solid_waste               : num  6 9.25 4.85 5.75 0 5.5 5 1.7 7 9 ...
##  $ solid_waste_pctl          : num  80 89.3 73.1 78.1 0 ...
##  $ pollution_burden          : num  79 73.4 77.7 67.8 66.8 ...
##  $ pollution_burden_score    : num  9.64 8.97 9.48 8.28 8.16 ...
##  $ pollution_burden_pctl     : num  99.9 99.3 99.9 97.4 96.9 ...
##  $ asthma                    : num  129.5 105.9 76.1 139.4 139.1 ...
##  $ asthma_pctl               : num  97.2 94.2 82.8 98.2 98.2 ...
##  $ low_birth_weight          : num  7.8 6.88 7.11 10.65 10.25 ...
##  $ low_birth_weight_pctl     : num  95.6 88.7 90.9 99.8 99.7 ...
##  $ cardiovascular_disease    : num  21.5 20.3 20.9 22.7 22.6 ...
##  $ cardiovascular_disease_pctl: num  92.2 88.1 90.2 94.6 94.4 ...
##  $ education                 : num  44.5 46.4 52.2 41.4 43.6 33.1 44.2 43.3 52.4 48.6 ...
##  $ education_pctl            : num  93.2 94.5 97.4 90.9 92.6 ...
##  $ linguistic_isolation      : num  16 29.7 17.1 15.7 20 NA 21.7 9.7 22.4 24 ...
##  $ linguistic_isolation_pctl : num  79.4 95.5 81.6 78.7 86.6 ...
##  $ poverty                   : num  76 73.2 62.6 65.7 72.7 43.5 79.5 56.8 72.2 78.8 ...
##  $ poverty_pctl              : num  98.9 98.4 93.4 95.4 98.3 ...
##  $ unemployment              : num  12.8 19.8 6.4 15.7 13.7 9.3 15.4 12.6 16.3 14.6 ...
##  $ unemployment_pctl         : num  93.8 99.2 61.5 97.3 95.3 ...
##  $ housing_burden            : num  30.3 31.2 20.3 35.4 32.7 23.7 33.3 29.6 30.8 33.1 ...
##  $ housing_burden_pctl       : num  91 92.3 64 96.4 94.2 ...
##  $ pop_char                  : num  93.2 93.2 83.8 94.6 95.4 ...
##  $ pop_char_score            : num  9.66 9.66 8.69 9.82 9.9 ...
##  $ pop_char_pctl             : num  99.7 99.7 95.8 99.9 99.9 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
str(scores)
```

```
## Classes 'data.table' and 'data.frame':   8035 obs. of  15 variables:
##  $ census_tract                :integer64 6001400100 6001400200 6001400300 6001400400 600140050
0 6001400600 6001400700 6001400800 ...
##  $ ces_4.0_score               : num  4.85 4.88 11.2 12.39 16.73 ...
##  $ ces_4.0_percentile          : num  2.8 2.87 15.94 18.97 29.74 ...
##  $ ces_4.0_percentile_range    : chr  "1-5% (lowest scores)" "1-5% (lowest scores)" "15-20%" "1
5-20%" ...
##  $ total_population            : int  3120 2007 5051 4007 4124 1745 5128 4069 2471 6133 ...
##  $ children_below_10_years_prct: num  7.82 10.46 11.42 9.38 9.12 ...
##  $ pop_10_to_64_years_prct     : num  66.1 66.3 73 78.8 82 ...
##  $ elderly_above_64_years_prct : num  26.06 23.22 15.54 11.83 8.92 ...
##  $ hispanic_prct               : num  3.78 8.67 6.95 12.1 9.46 ...
##  $ white_prct                  : num  74.3 73.5 68 63.7 45.4 ...
##  $ african_american_prct       : num  3.43 2.59 9.09 6.64 21.39 ...
##  $ native_american_prct        : num  0 0.2 0 0.87 0 0.17 0 0.2 2.47 0 ...
##  $ asian_american_prct         : num  12.53 8.52 12.14 10.48 11.34 ...
##  $ other/multiple_prct         : num  5.99 6.53 3.84 6.16 12.37 ...
##  $ county                      : chr  "Alameda County" "Alameda County" "Alameda County" "Alame
da County" ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
str(asthma)
```

```
## Classes 'data.table' and 'data.frame':   4484 obs. of  7 variables:
##  $ county                    : chr  "CALIFORNIA" "ALAMEDA" "ALPINE" "AMADOR" ...
##  $ year                      : int  2015 2015 2015 2015 2015 2015 2015 2015 2015 2015 ...
##  $ strata                    : chr  "Total population" "Total population" "Total population" "T
otal population" ...
##  $ strata_name               : chr  "All ages" "All ages" "All ages" "All ages" ...
##  $ age_group                 : chr  "All ages" "All ages" "All ages" "All ages" ...
##  $ number_of_ed_visits       : int  191904 9939 0 196 1044 185 97 6858 140 592 ...
##  $ age-adjusted_ed_visit_rate: num  50.4 64.3 0 58.4 50.2 48 41.4 65.2 53 36.4 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
#I checked the dataframe structure to see if we needed to change column types.
#The column types seem to be accurately assigned. Based on the column
#descriptors, there seems to be no mismatch of data types. Characters are
#characters, integers are integers, and decimals are numbers.
```

Identify data types for 5+ data elements/columns/variables

Identify 5+ data elements required for your specified scenario. If <5 elements are required to complete the analysis, please choose additional variables of interest in the data set to explore in this milestone.

Utilize functions or resources in RStudio to determine the types of each data element (i.e. character, numeric, factor)

Provide a basic description of the 5+ data elements

Numeric: mean, median, range

Character: unique values/categories

Or any other descriptives that will be useful to the analysis

```
# 1 - CES County
# lists which of CA's 58 counties is specified
class(measures$california_county)
```

```
## [1] "character"
```

```
# 2 - CES Asthma
class(measures$asthma)
```

```
## [1] "numeric"
```

```
summary(measures$asthma)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    4.28   30.06   45.71   51.98   65.80  243.29      11
```

```
class(measures$asthma_pctl)
```

```
## [1] "numeric"
```

```
summary(measures$asthma_pctl)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's 
##   0.01246  25.02804  50.01246  50.02060  75.01246 100.00000        11
```

```
# 3 - CES Score
class(scores$ces_4.0_score)
```

```
## [1] "numeric"
```

```
summary(scores$ces_4.0_score)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   1.034  14.787  25.554  28.324  40.057  93.184     103
```

```
class(scores$ces_4.0_percentile)
```

```
## [1] "numeric"
```

```
summary(scores$ces_4.0_percentile)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.      NA's 
##   0.01261  25.00946  50.00630  50.00630  75.00315 100.00000       103
```

```
# 4 - CES Age
class(scores$children_below_10_years_prct)
```

```
## [1] "numeric"
```

```
summary(scores$children_below_10_years_prct)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    0.00    9.28   11.96   12.06   14.78   51.47      23
```

```
class(scores$pop_10_to_64_years_prct)
```

```
## [1] "numeric"
```

```
summary(scores$pop_10_to_64_years_prct)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    0.00   69.99   73.64   73.13   76.84  100.00      23
```

```
class(scores$elderly_above_64_years_prct)
```

```
## [1] "numeric"
```

```
summary(scores$elderly_above_64_years_prct)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   0.000   9.508  13.340  14.802  18.320 100.000      23
```

```
# 5 - CES Race
class(scores$white_prct)
```

```
## [1] "numeric"
```

```
summary(scores$white_prct)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    0.00   14.73   37.26   38.68   61.00  100.00      23
```

```
class(scores$hispanic_prct)
```

```
## [1] "numeric"
```

```
summary(scores$hispanic_prct)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    0.00   15.67   30.93   38.09   58.17  100.00      23
```

```
class(scores$african_american_prct)
```

```
## [1] "numeric"
```

```
summary(scores$african_american_prct)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##  0.0000  0.8375  2.5700  5.5669  6.6500 84.7100      23
```

```
class(scores$native_american_prct)
```

```
## [1] "numeric"
```

```
summary(scores$native_american_prct)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.     NA's 
##   0.0000   0.0000   0.0000   0.4187   0.3600 100.0000       23
```

```
class(scores$asian_american_prct)
```

```
## [1] "numeric"
```

```
summary(scores$asian_american_prct)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   0.000   3.237   8.435  13.985  18.500  94.550      23
```

```
class(scores$`other/multiple_prct`)
```

```
## [1] "numeric"
```

```
summary(scores$`other/multiple_prct`)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##   0.000   1.397   2.850   3.268   4.700  17.070      23
```

```
# 6 - CHHS Asthma
class(asthma$year)
```

```
## [1] "integer"
```

```
summary(asthma$year)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##    2015    2016    2018    2018    2019    2020 
```

```
class(asthma$`age-adjusted_ed_visit_rate`)
```

```
## [1] "numeric"
```

```
summary(asthma$`age-adjusted_ed_visit_rate`)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    0.00   24.88   40.20   51.40   61.70 3531.00     716 
```

```
# also include county
```

1. measures [chr] California County “california_county” or [int] Census Tract “census_tract”

   Specifies the census tract or the California county that the census tract falls within, used to link the data to a geographic area. (Census tract is interpreted as an integer, but should be changed to characters as these numbers are used for identification, not values.)

2. measures [num] Asthma ED Visit Rate “asthma” or Asthma ED Visits Percentile “asthma_pctl”

   Our public health outcome of interest, asthma emergency department visit rates, which can be influenced by environmental burden, pollution, a history of racial segregation and socioeconomic factors contributing to racial health disparities, and more. More specifically, this variable is the spatially modeled age-adjusted rate of ED visits for asthma per 10,000 (averaged over 2015 to 2017).

3. scores [num] CES 4.0 Score “ces_4.0_score” or CES 4.0 Percentile “ces_4.0_percentile”

   The CalEnviroScreen Score, which uses environmental, health, and socioeconomic information to produce scores by census tract that identify California communities that are most affected by many sources of pollution and are often especially vulnerable to pollution’s effects. The score is calculated as follows: the “pollution burden” (average of exposures and environmental effects) multiplied by “population characteristics” (average of sensitive populations and socioeconomic factors). Each item is usually averaged from 2017 to 2019.

4. scores [num] Age Percents (multiple)

   The percentage of residents in each census tract that fall within the specified age groups (below 10, 10 to 64, 65 and above) according to estimates from the 2019 American Community Survey (ACS).

5. scores [num] Race Percents (multiple)

   The percentage of residents in each census tract that identify as a specified racial/ethnic group (White, Hispanic, African American, Native American, Asian American, Other/Multiple), per 2019 ACS estimates.

6. asthma [int] Year “year”, [num] Asthma ED Visit Rate “age-adjusted_ed_visit_rate”, [chr] County “county”

   Values from the other dataset to be analyzed, CHHS, regarding asthma ED visit rates by county by year (2015 to 2020).

Describe cleaning that each data source may need NOTE: There is no requirement for any data cleaning in this milestone. Please just list out the anticipated data cleaning needed. Examples: *Data elements that need to be converted to a different type*

-May have to convert Census Tract numbers to characters

-Convert CES percentile range to characters

Data elements that need cleaning or re-categorization

-Convert demographic percentages to raw numbers

-Rounding decimal points to just 2 after for most of the CES scores and percentile

Data elements that may be used for future joins (i.e. state, county)

-County CES scores with county ED visit rates due to asthma