Aggregation

Ricky Truong

August 2025

Background

Delete everything in environment

{r, message = FALSE, warning = FALSE} # rm(list = ls()) #

Load libraries

```
library(arrow)
library(sf)
library(tidyverse)
```

Load data for crosswalk (between block group and ZCTA)

```
crosswalk <- read_parquet("nhgis__block2zcta__2010.parquet")
```

Load data for PM2.5 (at ZCTA level)

```
# Create path to the .parquet files
pm25_path <- "dataverse_files"

# List all .parquet files for years 2000-2020
file_list <- list.files(path = pm25_path,
                        pattern = "^pm25__ushap__zcta_yearly__20\\d{2}\\.parquet$",
                        full.names = TRUE)

# Read and combine all files into data frame
pm25 <- file_list %>%
  lapply(read_parquet) %>%
  bind_rows()
```

Load data for CO2 (at census-block-group level)

```
# Create path to the .gdb directory
co2_path <- "CMS_DARTE_V2_1735/data/DARTE_v2.gdb"

# List the available layers (feature classes) inside the .gdb
st_layers(dsn = co2_path)
```

```
## Driver: OpenFileGDB
## Available layers:
##                             layer_name geometry_type features fields   crs_name
## 1 DARTE_v2_blockgroup_kgco2_1980_2017 Multi Polygon   220333     43 Vulcan_LCC
```

```
# Load the specific layer
co2 <- st_read(dsn = co2_path, layer = "DARTE_v2_blockgroup_kgco2_1980_2017")
```

```
## Reading layer 'DARTE_v2_blockgroup_kgco2_1980_2017' from data source
##   '/n/home10/rtruong/CMS_DARTE_V2_1735/data/DARTE_v2.gdb' using driver 'OpenFileGDB'
## Simple feature collection with 220333 features and 43 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: -7034829 ymin: -1957575 xmax: 3488418 ymax: 4595604
## Projected CRS: Vulcan_LCC
```

Data wrangling for `crosswalk`

Add separate variables for `block` and `block_group`

```
crosswalk <- crosswalk %>%
  rename(fips_code = block) %>%
  mutate(block_group = substring(fips_code, 12, 12),
         block = substring(fips_code, 13, 15))
```

Calculate weights

```
crosswalk <- crosswalk %>%
  group_by(state, county, tract, block_group) %>%
  mutate(block_group_population = sum(total_pop),
         block_fraction = total_pop / block_group_population)
```

Add separate variable for `fips_code_no_block` (to combine with `co2`)

```
crosswalk <- crosswalk %>%
  mutate(fips_code_no_block = substring(fips_code, 1, 12))
```

Data wrangling for `co2`

Convert `co2` to tidy format

```
co2 <- co2 %>%
  pivot_longer(cols = starts_with("kgco2_"),
               names_to = "year",
               names_prefix = "kgco2_",
               values_to = "value") %>%
  mutate(year = as.integer(year))
```

Previewing

Preview `crosswalk`

```
head(crosswalk)
```

```
## # A tibble: 6 x 11
## # Groups:   state, county, tract, block_group [1]
##   fips_code       zcta  total_pop state county tract  block_group block
##   <chr>           <chr>     <int> <chr> <chr>  <chr>  <chr>       <chr>
## 1 010010201001000 36067        61 01    001    020100 1           000
## 2 010010201001001 36067         0 01    001    020100 1           001
## 3 010010201001002 36067         0 01    001    020100 1           002
## 4 010010201001003 36067        75 01    001    020100 1           003
## 5 010010201001004 36067         0 01    001    020100 1           004
## 6 010010201001005 36067         1 01    001    020100 1           005
## # i 3 more variables: block_group_population <int>, block_fraction <dbl>,
## #   fips_code_no_block <chr>
```

Preview `pm25`

```
head(pm25)
```

```
## # A tibble: 6 x 3
##    year  PM25 zcta
##   <int> <dbl> <chr>
## 1  2000 11.0  01040
## 2  2000  9.97 01050
## 3  2000 10.3  01053
## 4  2000 11.0  01056
## 5  2000 10.8  01057
## 6  2000 10.8  01060
```

Preview co2

```
head(co2)
```

```
## Simple feature collection with 6 features and 7 fields
## Geometry type: MULTIPOLYGON
## Dimension:     XY
## Bounding box:  xmin: 1088553 ymin: -744852.4 xmax: 1091589 ymax: -740659.4
## Projected CRS: Vulcan_LCC
## # A tibble: 6 x 8
##   GEOID     bg_area_m2 geo_num Shape_Length Shape_Area                    Shape
##   <chr>          <dbl>   <dbl>        <dbl>      <dbl>        <MULTIPOLYGON [m]>
## 1 01081041~   6974538. 1.08e10       12959.   6974538. (((1088813 -744604.1, 10~
## 2 01081041~   6974538. 1.08e10       12959.   6974538. (((1088813 -744604.1, 10~
## 3 01081041~   6974538. 1.08e10       12959.   6974538. (((1088813 -744604.1, 10~
## 4 01081041~   6974538. 1.08e10       12959.   6974538. (((1088813 -744604.1, 10~
## 5 01081041~   6974538. 1.08e10       12959.   6974538. (((1088813 -744604.1, 10~
## 6 01081041~   6974538. 1.08e10       12959.   6974538. (((1088813 -744604.1, 10~
## # i 2 more variables: year <int>, value <dbl>
```

Apply crosswalk

Search crosswalk for value in pm25 as a sanity check

```
crosswalk %>%
  filter(zcta == "01040")
```

```
## # A tibble: 944 x 11
## # Groups:   state, county, tract, block_group [37]
##    fips_code       zcta  total_pop state county tract  block_group block
##    <chr>           <chr>     <int> <chr> <chr>  <chr>  <chr>       <chr>
##  1 250138114001000 01040         0 25    013    811400 1           000
##  2 250138114001001 01040         0 25    013    811400 1           001
```

```
##  3 250138114001002 01040         0 25    013    811400 1           002
##  4 250138114001003 01040         0 25    013    811400 1           003
##  5 250138114001004 01040         0 25    013    811400 1           004
##  6 250138114001005 01040         0 25    013    811400 1           005
##  7 250138114001006 01040         0 25    013    811400 1           006
##  8 250138114001007 01040         0 25    013    811400 1           007
##  9 250138114001008 01040         0 25    013    811400 1           008
## 10 250138114001009 01040         0 25    013    811400 1           009
## # i 934 more rows
## # i 3 more variables: block_group_population <int>, block_fraction <dbl>,
## #   fips_code_no_block <chr>
```

Search crosswalk for value in co2 as a sanity check

```
crosswalk %>%
  filter(fips_code_no_block == "010810416001")
```

```
## # A tibble: 36 x 11
## # Groups:   state, county, tract, block_group [1]
##    fips_code       zcta  total_pop state county tract  block_group block
##    <chr>           <chr>     <int> <chr> <chr>  <chr>  <chr>       <chr>
##  1 010810416001000 36801       898 01    081    041600 1           000
##  2 010810416001001 36801         0 01    081    041600 1           001
##  3 010810416001002 36801         0 01    081    041600 1           002
##  4 010810416001003 36801       210 01    081    041600 1           003
##  5 010810416001004 36801        12 01    081    041600 1           004
##  6 010810416001005 36801         0 01    081    041600 1           005
##  7 010810416001006 36801         0 01    081    041600 1           006
##  8 010810416001007 36801       335 01    081    041600 1           007
##  9 010810416001008 36801       165 01    081    041600 1           008
## 10 010810416001009 36801       169 01    081    041600 1           009
## # i 26 more rows
## # i 3 more variables: block_group_population <int>, block_fraction <dbl>,
## #   fips_code_no_block <chr>
```

Combine crosswalk and co2

```
# Select for only necessary variables to avoid large data sets
crosswalk <- crosswalk %>%
  ungroup() %>%
  select(fips_code, zcta, block_fraction, fips_code_no_block)

co2 <- co2 %>%
  ungroup() %>%
```

```
  st_drop_geometry() %>%
  as.data.frame() %>%
  select(GEOID, year, value)

# Combine via inner_join()
df <- inner_join(crosswalk, co2, join_by("fips_code_no_block" == "GEOID"))
```

```
## Warning in inner_join(crosswalk, co2, join_by("fips_code_no_block" == "GEOID")): Detected an
## i Row 1 of `x` matches multiple rows in `y`.
## i Row 83563 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to silence this warning.
```

```
# Wrangle df
df <- df %>%
  rename(block_group_value = value) %>%
  mutate(block_value = block_group_value * block_fraction) %>%
  select(-c(fips_code_no_block, block_fraction))

# Preview df
head(df, 100)
```

```
## # A tibble: 100 x 5
##    fips_code       zcta   year block_group_value block_value
##    <chr>           <chr> <int>             <dbl>       <dbl>
##  1 010010201001000 36067  1980          2514107.     219714.
##  2 010010201001000 36067  1981          2487247.     217367.
##  3 010010201001000 36067  1982          2456484.     214678.
##  4 010010201001000 36067  1983          2546810.     222572.
##  5 010010201001000 36067  1984          2664648.     232870.
##  6 010010201001000 36067  1985          2721388.     237829.
##  7 010010201001000 36067  1986          2834398.     247705.
##  8 010010201001000 36067  1987          2875839.     251327.
##  9 010010201001000 36067  1988          2919320.     255127.
## 10 010010201001000 36067  1989          2929569.     256022.
## # i 90 more rows
```

Double check output (again, as a sanity check)

```
# For block group of 010010201001000 in 1980, the value should be 2514107
co2 %>%
  filter(GEOID == "010010201001", year == 1980)
```

```
##          GEOID year   value
## 1 010010201001 1980 2514107
```

Applying pm25

Combine pm25 and df

```
# Rename block_value to CO2 and aggregate from block-year to ZCTA-year
df_zcta_year <- df %>%
  rename(CO2 = block_value) %>%
  group_by(zcta, year) %>%
  summarise(CO2 = sum(CO2, na.rm = TRUE), .groups = "drop")

# Combine via inner_join()
aggregated <- inner_join(pm25, df_zcta_year, join_by("zcta", "year"))
```

Double check extreme values

Preview aggregated

```
# Preview aggregated with no organization
head(aggregated, 100)
```

```
## # A tibble: 100 x 4
##     year  PM25 zcta         CO2
##    <int> <dbl> <chr>      <dbl>
##  1  2000 11.0  01040 175364484.
##  2  2000  9.97 01050  10033335.
##  3  2000 10.3  01053   2347437.
##  4  2000 11.0  01056 142651609.
##  5  2000 10.8  01057  22816411.
##  6  2000 10.8  01060 143065898.
##  7  2000 10.4  01062  18294790.
##  8  2000 10.2  01066    744847.
##  9  2000 10.6  01069 122951078.
## 10  2000  8.93 01070   4338842.
## # i 90 more rows
```

```
# Preview aggregated with organization
aggregated %>%
  arrange(zcta) %>%
  select(year, zcta, PM25, CO2) %>%
  head(100)
```

```
## # A tibble: 100 x 4
##     year zcta   PM25   CO2
```

```
##    <int> <chr> <dbl> <dbl>
##  1  2000 00601    NA     0
##  2  2001 00601    NA     0
##  3  2002 00601    NA     0
##  4  2003 00601    NA     0
##  5  2004 00601    NA     0
##  6  2005 00601    NA     0
##  7  2006 00601    NA     0
##  8  2007 00601    NA     0
##  9  2008 00601    NA     0
## 10  2009 00601    NA     0
## # i 90 more rows
```

```
# Filter for rows with missing values for PM25
missing_pm25 <- aggregated %>%
  filter(is.na(PM25))

missing_pm25
```

```
## # A tibble: 8,262 x 4
##     year  PM25 zcta    CO2
##    <int> <dbl> <chr> <dbl>
##  1  2000    NA 00601     0
##  2  2000    NA 00602     0
##  3  2000    NA 00603     0
##  4  2000    NA 00606     0
##  5  2000    NA 00610     0
##  6  2000    NA 00612     0
##  7  2000    NA 00616     0
##  8  2000    NA 00617     0
##  9  2000    NA 00622     0
## 10  2000    NA 00623     0
## # i 8,252 more rows
```

```
# Filter for rows with missing values for CO2
missing_co2 <- aggregated %>%
  filter(is.na(CO2))

missing_co2
```

```
## # A tibble: 0 x 4
## # i 4 variables: year <int>, PM25 <dbl>, zcta <chr>, CO2 <dbl>
```

```
# Filter for rows with zero for PM25
zero_pm25 <- aggregated %>%
  filter(PM25 == 0)
```

```
zero_pm25
```

```
## # A tibble: 0 x 4
## # i 4 variables: year <int>, PM25 <dbl>, zcta <chr>, CO2 <dbl>
```

```
# Filter for rows with zero for CO2
zero_co2 <- aggregated %>%
  filter(CO2 == 0)

zero_co2
```

```
## # A tibble: 10,746 x 4
##     year  PM25 zcta    CO2
##    <int> <dbl> <chr> <dbl>
##  1  2000    NA 00601     0
##  2  2000    NA 00602     0
##  3  2000    NA 00603     0
##  4  2000    NA 00606     0
##  5  2000    NA 00610     0
##  6  2000    NA 00612     0
##  7  2000    NA 00616     0
##  8  2000    NA 00617     0
##  9  2000    NA 00622     0
## 10  2000    NA 00623     0
## # i 10,736 more rows
```

Search pm25 for missing value in aggregated as a sanity check

```
# For ZCTA of 00601 in 2000, the value should be NA
pm25 %>%
  filter(zcta == "00601", year == 2000)
```

```
## # A tibble: 1 x 3
##    year  PM25 zcta
##   <int> <dbl> <chr>
## 1  2000    NA 00601
```

Search df for zero value in aggregated as a sanity check

```
# For ZCTA of 00601 in 2000, the values should be 0
df %>%
  filter(zcta == "00601", year == 2000)
```

```
## # A tibble: 1,177 x 5
##    fips_code       zcta   year block_group_value block_value
##    <chr>           <chr> <int>              <dbl>       <dbl>
##  1 720019563001000 00601  2000                  0           0
##  2 720019563001001 00601  2000                  0           0
##  3 720019563001002 00601  2000                  0           0
##  4 720019563001003 00601  2000                  0           0
##  5 720019563001004 00601  2000                  0           0
##  6 720019563001005 00601  2000                  0           0
##  7 720019563001006 00601  2000                  0           0
##  8 720019563001007 00601  2000                  0           0
##  9 720019563001008 00601  2000                  0           0
## 10 720019563001009 00601  2000                  0           0
## # i 1,167 more rows
```

Exporting files

Export aggregated

```
write.csv(aggregated, "~/aggregated.csv", row.names = FALSE)
```