Homework_1

January 25, 2026

0.1 Preliminary Loading

Note: Due to the storm, my wifi at home was running rather slowly. My computer was not able to properly run some of the longer lines of code from the data cleaning portion. I did my best to make sure I understood all the commands and what exactly I was doing at every step, but I will be at office hours this week to make sure my approach was correct, and that I better understand next week's work

[1]:
```
!pip -q install rpy2
%load_ext rpy2.ipython
```

```
[notice] A new release of pip is
available: 24.2 -> 25.3
[notice] To update, run:
pip install --upgrade pip
```

[ ]:
```
%%R
library(tidyverse)
# Month list ------------------------------------------------------------------
y <- 2018
monthlist <- sprintf("%02d", 1:12)

  # Readers (quiet & typed) ---------------------------------------------------
  read_contract <- function(path) {
    read_csv(
      path,
      skip = 1,
      col_names = c(
        "contractid","planid","org_type","plan_type","partd","snp","eghp",
        "org_name","org_marketing_name","plan_name","parent_org","contract_date"
      ),
      col_types = cols(
        contractid = col_character(),
        planid     = col_double(),
        org_type   = col_character(),
        plan_type  = col_character(),
        partd      = col_character(),
        snp        = col_character(),
        eghp       = col_character(),
```

```
      org_name   = col_character(),
      org_marketing_name = col_character(),
      plan_name  = col_character(),
      parent_org = col_character(),
      contract_date = col_character()
    ),
    show_col_types = FALSE,
    progress = FALSE
  )
}

read_enroll <- function(path) {
  read_csv(
    path,
    skip = 1,
    col_names =␣
↪c("contractid","planid","ssa","fips","state","county","enrollment"),
    col_types = cols(
      contractid = col_character(),
      planid     = col_double(),
      ssa        = col_double(),
      fips       = col_double(),
      state      = col_character(),
      county     = col_character(),
      enrollment = col_double()
    ),
    na = "*",
    show_col_types = FALSE,
    progress = FALSE
  )
}

load_month <- function(m, y) {
  c_path <- paste0("../ma-data/ma/enrollment/Extracted Data/
↪CPSC_Contract_Info_", y, "_", m, ".csv")
  e_path <- paste0("../ma-data/ma/enrollment/Extracted Data/
↪CPSC_Enrollment_Info_", y, "_", m, ".csv")

  contract.info <- read_contract(c_path) %>%
    distinct(contractid, planid, .keep_all = TRUE)

  enroll.info <- read_enroll(e_path)

  contract.info %>%
    left_join(enroll.info, by = c("contractid","planid")) %>%
    mutate(month = as.integer(m), year = y)
}
```

```
# Read all months, then tidy once ---------------------------------------

plan.year <- map_dfr(monthlist, ~ load_month(.x, y)) %>%
  arrange(contractid, planid, state, county, month) %>%
  group_by(state, county) %>%
  fill(fips, .direction = "downup") %>%
  ungroup() %>%
  group_by(contractid, planid) %>%
  fill(plan_type, partd, snp, eghp, plan_name, .direction = "downup") %>%
  ungroup() %>%
  group_by(contractid) %>%
  fill(org_type, org_name, org_marketing_name, parent_org, .direction =␣
↪"downup") %>%
  ungroup()


# Collapse to yearly panel ----------------------------------------------
final.plans <- plan.year %>%
  group_by(contractid, planid, fips, year) %>%
  arrange(month, .by_group = TRUE) %>%
  summarize(
    n_nonmiss        = sum(!is.na(enrollment)),
    avg_enrollment   = ifelse(n_nonmiss > 0, mean(enrollment, na.rm = TRUE),␣
↪NA_real_),
    sd_enrollment    = ifelse(n_nonmiss > 1, sd(enrollment, na.rm = TRUE),␣
↪NA_real_),
    min_enrollment   = ifelse(n_nonmiss > 0, min(enrollment, na.rm = TRUE),␣
↪NA_real_),
    max_enrollment   = ifelse(n_nonmiss > 0, max(enrollment, na.rm = TRUE),␣
↪NA_real_),
    first_enrollment = ifelse(n_nonmiss > 0, first(na.omit(enrollment)),␣
↪NA_real_),
    last_enrollment  = ifelse(n_nonmiss > 0,  last(na.omit(enrollment)),␣
↪NA_real_),
    state            = last(state),
    county           = last(county),
    org_type         = last(org_type),
    plan_type        = last(plan_type),
    partd            = last(partd),
    snp              = last(snp),
    eghp             = last(eghp),
    org_name         = last(org_name),
    org_marketing_name = last(org_marketing_name),
    plan_name        = last(plan_name),
    parent_org       = last(parent_org),
```

```
      contract_date    = last(contract_date),
      year             = last(year),
      .groups = "drop"
    )
```

```
  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   3.4.4       tibble    3.2.1
 lubridate 1.9.3       tidyr     1.3.0
 purrr     1.0.2
  Conflicts                         tidyverse_conflicts()
 dplyr::filter() masks stats::filter()
 dplyr::lag()    masks stats::lag()
 Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors
```

[ ]:
```
%%R
library(tidyverse)


# Month list ------------------------------------------------------------------
y <- 2018
monthlist <- sprintf("%02d", 1:12)

  # Readers (quiet & typed) ---------------------------------------------------
  read_contract <- function(path) {
    read_csv(
      path,
      skip = 1,
      col_names = c(
        "contractid","planid","org_type","plan_type","partd","snp","eghp",
        "org_name","org_marketing_name","plan_name","parent_org","contract_date"
      ),
      col_types = cols(
        contractid = col_character(),
        planid     = col_double(),
        org_type   = col_character(),
        plan_type  = col_character(),
        partd      = col_character(),
        snp        = col_character(),
        eghp       = col_character(),
        org_name   = col_character(),
        org_marketing_name = col_character(),
        plan_name  = col_character(),
        parent_org = col_character(),
        contract_date = col_character()
```

```
    ),
    show_col_types = FALSE,
    progress = FALSE
  )
}

read_enroll <- function(path) {
  read_csv(
    path,
    skip = 1,
    col_names =␣
↪c("contractid","planid","ssa","fips","state","county","enrollment"),
    col_types = cols(
      contractid = col_character(),
      planid     = col_double(),
      ssa        = col_double(),
      fips       = col_double(),
      state      = col_character(),
      county     = col_character(),
      enrollment = col_double()
    ),
    na = "*",
    show_col_types = FALSE,
    progress = FALSE
  )
}

# One-month loader ------------------------------------------------------
load_month <- function(m, y) {
  c_path <- paste0("../ma-data/ma/enrollment/Extracted Data/
↪CPSC_Contract_Info_", y, "_", m, ".csv")
  e_path <- paste0("../ma-data/ma/enrollment/Extracted Data/
↪CPSC_Enrollment_Info_", y, "_", m, ".csv")

  contract.info <- read_contract(c_path) %>%
    distinct(contractid, planid, .keep_all = TRUE)

  enroll.info <- read_enroll(e_path)

  contract.info %>%
    left_join(enroll.info, by = c("contractid","planid")) %>%
    mutate(month = as.integer(m), year = y)
}

# Read all months, then tidy once ---------------------------------------

plan.year <- map_dfr(monthlist, ~ load_month(.x, y)) %>%
```

```
  arrange(contractid, planid, state, county, month) %>%
  group_by(state, county) %>%
  fill(fips, .direction = "downup") %>%
  ungroup() %>%
  group_by(contractid, planid) %>%
  fill(plan_type, partd, snp, eghp, plan_name, .direction = "downup") %>%
  ungroup() %>%
  group_by(contractid) %>%
  fill(org_type, org_name, org_marketing_name, parent_org, .direction =␣
↪"downup") %>%
  ungroup()


# Collapse to yearly panel -----------------------------------------------
final.plans <- plan.year %>%
  group_by(contractid, planid, fips, year) %>%
  arrange(month, .by_group = TRUE) %>%
  summarize(
    n_nonmiss        = sum(!is.na(enrollment)),
    avg_enrollment   = ifelse(n_nonmiss > 0, mean(enrollment, na.rm = TRUE),␣
↪NA_real_),
    sd_enrollment    = ifelse(n_nonmiss > 1, sd(enrollment, na.rm = TRUE),␣
↪NA_real_),
    min_enrollment   = ifelse(n_nonmiss > 0, min(enrollment, na.rm = TRUE),␣
↪NA_real_),
    max_enrollment   = ifelse(n_nonmiss > 0, max(enrollment, na.rm = TRUE),␣
↪NA_real_),
    first_enrollment = ifelse(n_nonmiss > 0, first(na.omit(enrollment)),␣
↪NA_real_),
    last_enrollment  = ifelse(n_nonmiss > 0,  last(na.omit(enrollment)),␣
↪NA_real_),
    state            = last(state),
    county           = last(county),
    org_type         = last(org_type),
    plan_type        = last(plan_type),
    partd            = last(partd),
    snp              = last(snp),
    eghp             = last(eghp),
    org_name         = last(org_name),
    org_marketing_name = last(org_marketing_name),
    plan_name        = last(plan_name),
    parent_org       = last(parent_org),
    contract_date    = last(contract_date),
    year             = last(year),
    .groups = "drop"
  )
```

0.1.1 Question Attempts

[ ]:
```
%%R
library(tidyverse)
# Question 1: Table for plan types
table1 <- final_plans %>%
  group_by(plan_type) %>%
  summarize(plan_count = n_distinct(contractid, planid)) %>%
  rename(`2018` = plan_count)

print(table1)
```

[ ]:
```
%%R
library(tidyverse)
# Question 2: Remove SNP, eghp, and 800 series plans
final_plans_clean <- final_plans %>%
  filter(snp == "No",
         eghp == "No",
         planid < 800)

table1_new <- final_plans_clean %>%
  group_by(plan_type) %>%
  summarize(plan_count = n_distinct(contractid, planid)) %>%
  rename(`2018` = plan_count)

print(table1_new)
```

[ ]:
```
%%R
library(tidyverse)
# Question 3: Inner merge with Service Area and calculate average enrollment
final_data <- final_plans_clean %>%
  inner_join(final_service_area, by = c("contractid", "fips"))

table1_average <- final_data %>%
  group_by(plan_type) %>%
  summarize(avg_enrollment = mean(avg_enrollment, na.rm = TRUE))

print(table1_average)
```