EDA

Jinyu Luo

2024-04-11

```
# States info that were collected in 2017-18
state2017 <- c("Colorado", "Connecticut", "Minnesota", "Montana",
               "New Jersey", "New York", "North Dakota",
               "Pennsylvania", "South Dakota", "Utah", "Washington")

# State abbreviations mapping
state_abbreviations <- c(
  "Alabama"="AL", "Alaska"="AK", "Arizona"="AZ", "Arkansas"="AR", "California"="CA",
  "Colorado"="CO", "Connecticut"="CT", "Delaware"="DE", "Florida"="FL", "Georgia"="GA",
  "Hawaii"="HI", "Idaho"="ID", "Illinois"="IL", "Indiana"="IN", "Iowa"="IA",
  "Kansas"="KS", "Kentucky"="KY", "Louisiana"="LA", "Maine"="ME", "Maryland"="MD",
  "Massachusetts"="MA", "Michigan"="MI", "Minnesota"="MN", "Mississippi"="MS", "Missouri"="MO",
  "Montana"="MT", "Nebraska"="NE", "Nevada"="NV", "New Hampshire"="NH", "New Jersey"="NJ",
  "New Mexico"="NM", "New York"="NY", "North Carolina"="NC", "North Dakota"="ND", "Ohio"="OH",
  "Oklahoma"="OK", "Oregon"="OR", "Pennsylvania"="PA", "Rhode Island"="RI", "South Carolina"="SC",
  "South Dakota"="SD", "Tennessee"="TN", "Texas"="TX", "Utah"="UT", "Vermont"="VT",
  "Virginia"="VA", "Washington"="WA", "West Virginia"="WV", "Wisconsin"="WI", "Wyoming"="WY"
)

vaccine <- read.csv("vaccine.csv") %>%
  mutate(type = ifelse(type == "", "Unknown", type),
         mmr = ifelse(mmr == -1, NA, mmr),
         overall = ifelse(overall == -1, NA, overall),
         per_capita = statespending2016/schagepop2016,
         year = case_when(state %in% state2017 ~ "2017-18", TRUE ~ "2018-19"),
         county = ifelse(county == "", NA, county),
         city = ifelse(city == "", NA, city),
         state_abbr = state_abbreviations[state])

obsID <- 1:nrow(vaccine)
vac <- vaccine %>% mutate(oID = obsID) %>% relocate(oID, .before =state)
duplicate_df <- vac %>%
  group_by(state, county, city, name) %>%
  reframe(oID = oID, type = type, year = year, n= n(), enroll = enroll, mmr = mmr, overall = overall) %
  filter(n > 1) %>%
  group_by(state, county, city, name) %>%
  mutate(record = row_number()) %>% ungroup()

# Calculate average of duplicated data
avg_duplicates <- duplicate_df %>%
  group_by(state, county, city, name) %>%
  mutate(mean_enroll = mean(enroll), mean_mmr = mean(mmr), mean_overall = mean(overall)) %>%
```

```
  select(-c(mmr, overall, enroll, n)) %>%
  rename(mmr = mean_mmr, overall=mean_overall, enroll = mean_enroll)

remove_sid <- avg_duplicates %>% filter(record != 1) %>% pull(oID)

data <- vac %>% filter(!(oID %in% remove_sid)) %>%
  # Add the total number of schools in each state
  left_join(vac %>% group_by(state) %>% summarise(n_school = n_distinct(name)), by = "state")
```

```
data %>%
  group_by(state) %>%
  summarise(spending = unique(per_capita)) %>%
  arrange(spending)
```

```
## # A tibble: 32 x 2
##    state          spending
##    <chr>             <dbl>
##  1 Utah              5885.
##  2 Arizona           5953.
##  3 Idaho             5993.
##  4 Oklahoma          7114.
##  5 North Carolina    7468.
##  6 Tennessee         7554.
##  7 South Dakota      7801.
##  8 Florida           7976.
##  9 Texas             8152.
## 10 Arkansas          8567.
## # i 22 more rows
```

MMR

```
MMR <- data %>% filter(!is.na(mmr)) %>%
  # create a binary variable indicating whether or not the MMR rate
  # reached 95%
  mutate(rate95 = ifelse(mmr >= 95, 1, 0))

nStates <- n_distinct(MMR$state)
MMR <- MMR %>%
  left_join(data.frame(state = unique(MMR$state),
                       stateID = 1:nStates))
```

```
## Joining with `by = join_by(state)`
```

```
MMR %>%
  group_by(state) %>%
  summarise(spending = unique(statespending2016))
```

```
## # A tibble: 21 x 2
##    state         spending
```

```
##    <chr>            <int>
##  1 Arizona        7137123
##  2 Arkansas       4411761
##  3 California    68892072
##  4 Colorado       8060420
##  5 Connecticut    8687640
##  6 Illinois      23933484
##  7 Maine          2290535
##  8 Massachusetts 13998163
##  9 Minnesota      9987116
## 10 Missouri       8996869
## # i 11 more rows
```

Tables

```
MMR %>% group_by(type) %>%
  summarise(nSchools=n(), reached95 = sum(rate95)) %>%
  kable(booktabs = TRUE,
        col.names = c("School Type", "Total Schools (N = 28299)", "Rate Over 95% (N)"),
        caption = "Sample Duplicated School Observations in California") %>%
  kable_styling(latex_options = c("striped", "scale_down"))
```

```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

Table 1: Sample Duplicated School Observations in California

School Type	Total Schools (N = 28299)	Rate Over 95% (N)
BOCES	47	45
Charter	214	55
Kindergarten	1296	826
Nonpublic	18	13
Private	3123	2172
Public	11729	9476
Unknown	11872	7662

```
duplicate_df %>%
  filter(state == "California") %>%
  select(oID, record, name, county, city, type, year, enroll, mmr, overall) %>%
  kable(booktabs = TRUE,
        col.names = c("School ID", "No.Record", "School Name",
                      "County", "City","Type", "Year",
                      "Enrollment","MMR Rate", "Overall Rate"),
        caption = "Sample Duplicated School Observations in California") %>%
  kable_styling(latex_options = c("striped", "scale_down")) %>%
  landscape()
```

```
## Warning in styling_latex_scale(out, table_info, "down"): Longtable cannot be
## resized.
```

Table 2: Sample Duplicated School Observations in California

School ID	No.Record	School Name	County	City	Type	Year	Enrollment	MMR Rate	Overall Rate
5079	1	Stratford	Alameda	Fremont	Private	2018-19	88	98	98
7313	2	Stratford	Alameda	Fremont	Private	2018-19	45	95	95
2454	1	Lincoln Elementary	Fresno	Fresno	Public	2018-19	108	99	98
4232	2	Lincoln Elementary	Fresno	Fresno	Public	2018-19	74	98	98
8181	1	Anneliese Schools	Orange	Laguna Beach	Private	2018-19	28	79	64
8198	2	Anneliese Schools	Orange	Laguna Beach	Private	2018-19	39	77	77
5078	1	Stratford	Santa Clara	Santa Clara	Private	2018-19	161	98	98
7316	2	Stratford	Santa Clara	Santa Clara	Private	2018-19	43	95	95
5085	1	Stratford	Santa Clara	Sunnyvale	Private	2018-19	83	98	98
5086	2	Stratford	Santa Clara	Sunnyvale	Private	2018-19	88	98	98

Stratified Sampling

```r
# Stratified Sampling
# Step 1: Calculate weights for each state based on the number of schools
state_weights <- MMR %>%
  count(state) %>%
  mutate(weight = n / sum(n))

# Step 2: Sample schools within each state, proportional to state weights
# Set a total sample size of N schools
N <- 5000  # Total desired sample size

# Calculate the number of schools to sample from each state, based on weights
state_weights <- state_weights %>%
  mutate(sample_size = round(weight * N))

set.seed(123)

# mmr_samples <- mmr_dat %>%
#   group_by(state) %>%
#   sample_n(N, replace = FALSE) %>%
#   mutate(schoolID = row_number()) %>%
#   relocate(schoolID, .after = stateID) %>%
#   ungroup() %>%
#   arrange(stateID)


mmr_samples <- NULL
for (i in 1:nrow(state_weights)) {
  state <- state_weights[i, ]$state
  size <- state_weights$sample_size[i]
  sample <- MMR %>% filter(state == state) %>%
    sample_n(size, replace = FALSE)
  mmr_samples <- rbind(MMR, sample)
}

mmr_samples <- mmr_samples %>% group_by(state) %>%
  mutate(schoolID = row_number()) %>%
  relocate(schoolID, .after = stateID) %>%
  ungroup() %>%
  arrange(stateID)
```

GEE

```r
# Fit the GEE model
mmr_ex <- geeglm(mmr ~ per_capita + n_school + type,
                 family = gaussian,
                 data = mmr_samples,
                 id = stateID,
                 corstr = "exchangeable")
summary(mmr_ex)$coef
```

```r
# Fit the GEE model
mmr_ar1 <- geeglm(mmr ~ per_capita + n_school + type,
                  family = gaussian,
                  data = mmr_samples,
                  id = stateID,
                  corstr = "ar1")
summary(mmr_ar1)$coef

# Fit the GEE model
mmr_unstr <- geeglm(mmr ~ per_capita + n_school + type,
                  family = gaussian,
                  data = mmr_samples,
                  id = stateID,
                  corstr = "unstructured")
summary(mmr_unstr)$coef
```