Computing Covid Mortality rates using Facebook Data

Lindsay Katz

2023-06-16

Document Summary

Here is an overview of everything I did in this document:

First, I re-computed published quarterly Covid death rates using CDC Covid death count data and Census population data. This involved:

- Aggregating the population data into correct age groups
- Performing linear interpolation to obtain monthly population estimates
- Merging the deaths and population data
- Creating a quarter variable, and aggregating the monthly-level data to the quarterly-level
- Computing the age-specific quarterly death rates for each region
- Performing the age-adjustment using 2000 US standard population projection data from the years 2000 and 2021
  - Note: I tried two standard populations for the year 2000 - one from a table referenced in the CDC's technical report (grouped by age), and one from the Census website which is by single year of age, where I also got the 2021 data
- Assessing which standard population distribution produced death rates closest to those published
  - It was the 2000 national population projections for the year 2021, with an average absolute difference of 8.31 (compared to 22.1 and 21.9 from the 2000 standard pop. table and census data, respectively)
  - I plotted these all as well to provide a visual of the death rate estimate differences by state and quarter

Next, I prepared the deaths and population data to be analyzed with the Facebook data. This involved:

- Aggregating the population data into the age groups which match those in the deaths data
  - Note: the Covid deaths data had a few different overlapping age group options to choose from, so I picked the ones closest to those in the Facebook data
- Performing linear interpolation to obtain monthly population estimates
- Merging the deaths and population data
- Adding the smooth Facebook traveler rate data to the merged deaths and population dataframe

With the resulting dataframe, I computed age-specific mortality rates with and without the traveler adjustment. I created a PDF with plots of these age-specific death rate data for each region and sex, and looked at the average relative and absolute differences between the Covid death rates with and without the travel adjustment.

Re-computing published Covid death rates

Before performing our own analysis, we want to try to re-compute the published quarterly death rates using the Covid deaths and population data we intend to use for our analysis, to validate that the data we're using produce results that generally match those published. These published estimates are age-adjusted, and are not disaggregated by sex. We will follow the same computational method as outlined by the CDC, which uses "age-specific death rates to the U.S. standard population (relative age distribution of year 2000 projected population of the United States)".

In their technical note, the CDC cited this report which has the 2000 U.S. standard population with the following age groups: < 1 year, 1-4, 5-14, 15-24, 25-34, 35-44, 45-54, 55-64, 65-74, 75-84, and 85+. However, it's a bit unclear whether they used these data to perform the age-adjustment, or if they alternatively used the 2000 National population projections for the year 2021 which is available on the census website. The 2000 national population projection datasets available from the census bureau exist for 1999-2100, and are available by single year of age, which would be beneficial for our analysis using Facebook data later on which has unique age groupings. For this reason, we will perform the age-adjustment with three possible options for the standard population structure:

1. 2000 U.S. standard population data from Table V of the report cited by the CDC (with age groupings)
2. 2000 National Population Projections data from the Census bureau for the year 2000 (single year of age)
3. 2000 National Population Projections data from the Census bureau for the year 2021 (single year of age)

Read in cleaned Census population and CDC covid death data

```
library(tidyverse)
```

```
# clean pop data
pop_df <- read_csv("../data/census_pop_all_ages.csv", show_col_types = F) %>%
  select(age_gp:july1_2021)

# clean covid deaths data
deaths_df <- read_csv("../data/covid19_deaths_all_ages.csv", show_col_types = F) %>%
  select(region:last_col())
```

Aggregate population data

To begin, let's aggregate the population data to match the age groups from the standard population table mentioned earlier: < 1 year, 1-4, 5-14, 15-24, 25-34, 35-44, 45-54, 55-64, 65-74, 75-84, and 85+. These age groups are readily available in the Covid deaths dataset, so we can work with it as-is.

```
# create age groupings to match those of standard pop projection
pop_df_agg <- pop_df %>% filter(age_gp!="Total") %>%
  # remove + from 85+ so we can convert to numeric and group ages
  mutate(age_gp = str_remove(age_gp, "\\+")) %>%
  rename(age = age_gp) %>%
  mutate(age = as.numeric(age)) %>%
  mutate(age_gp = case_when(age <= 0 ~ "<1",
                          age > 0 & age <= 4 ~ "1-4",
                          age > 4 & age <= 14 ~ "5-14",
                          age > 14 & age <= 24 ~ "15-24",
                          age > 24 & age <= 34 ~ "25-34",
                          age > 34 & age <= 44 ~ "35-44",
                          age > 44 & age <= 54 ~ "45-54",
                          age > 54 & age <= 64 ~ "55-64",
                          age > 64 & age <= 74 ~ "65-74",
                          age > 74 & age <= 84 ~ "75-84",
                          age >= 85 ~ "85+")) %>%
  select(-age) %>%
  mutate(age_gp = factor(age_gp, level=c("<1", "1-4","5-14","15-24","25-34","35-44","45-54","55-64", "65
  arrange(region, age_gp) %>%
  select(region, sex, age_gp, july1_2020, july1_2021) %>%
  group_by(sex, region, age_gp) %>%
  summarise(across(july1_2020:july1_2021, sum)) %>%
  ungroup()
```

Perform linear interpolation to get monthly population estimates

Now we will obtain monthly population estimates using linear interpolation. To do so, we first compute the annual population change, and divide that by 12 so we have an estimate for monthly change.

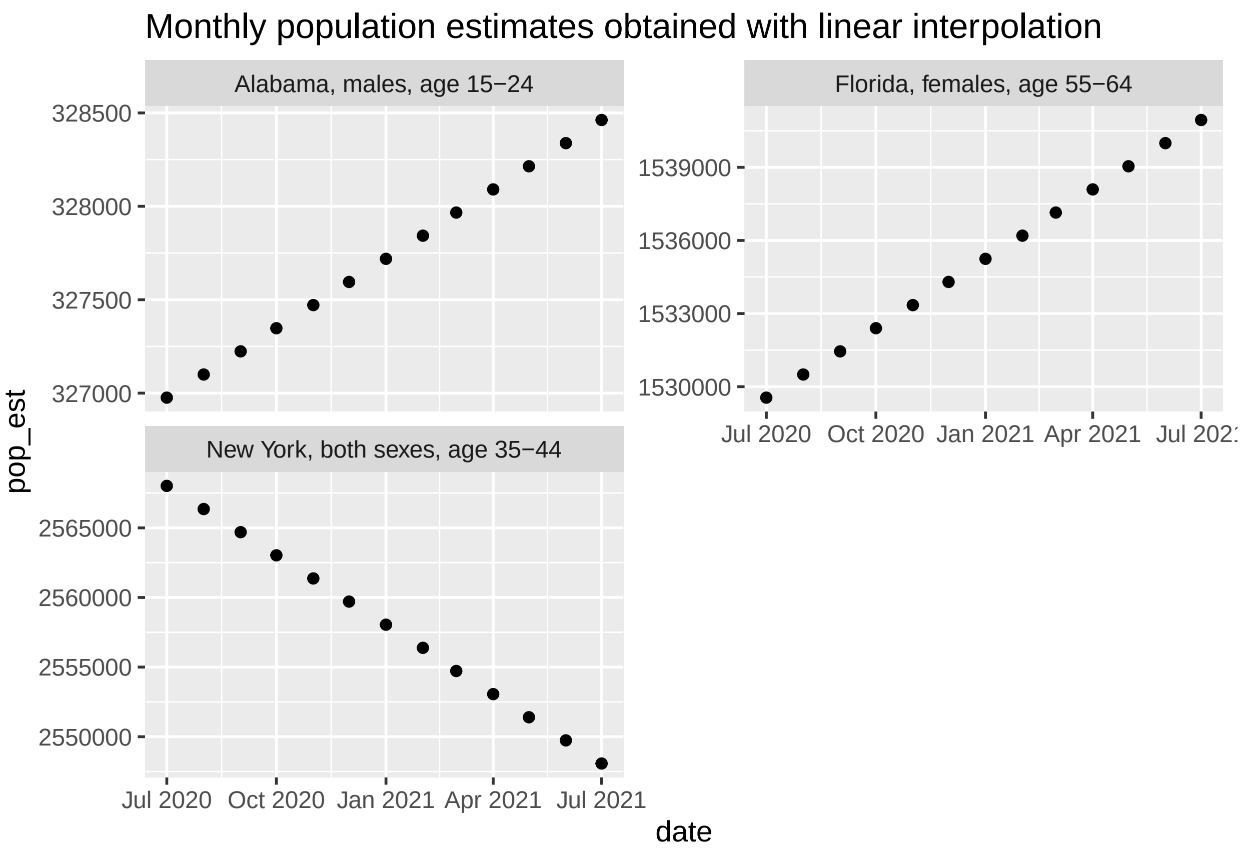
```
# compute annual population change, divide by 12 so we can add to each month for linear interpolation
pop_df_agg <- pop_df_agg %>% mutate(pop_diff = july1_2021-july1_2020,
                                    pop_diff_12 = pop_diff/12)
```

Now, we can use the monthly change value to perform linear interpolation, providing us with a population estimate for each month. We obtain the monthly population estimate for August 2020 by taking the population estimate we have from July 2020, and adding the monthly change value. We continue to do this for each consecutive month, using the previous month's estimate plus the monthly change value to obtain that month's population estimate, until we reach July 2021 for which we already have an estimate from the Census data.

```
# perform linear interpolation
# adding an additional 12th of the annual pop difference to the previous month's estimate
# pivot longer for plotting purposes, convert date column from string to date class
pop_df_agg <- pop_df_agg %>%
  mutate(aug1_2020 = july1_2020 + pop_diff_12,
         sept1_2020 = aug1_2020 + pop_diff_12,
         oct1_2020 = sept1_2020 + pop_diff_12,
         nov1_2020 = oct1_2020 + pop_diff_12,
         dec1_2020 = nov1_2020 + pop_diff_12,
         jan1_2021 = dec1_2020 + pop_diff_12,
         feb1_2021 = jan1_2021 + pop_diff_12,
         march1_2021 = feb1_2021 + pop_diff_12,
         april1_2021 = march1_2021 + pop_diff_12,
         may1_2021 = april1_2021 + pop_diff_12,
         june1_2021 = may1_2021 + pop_diff_12) %>%
  select(age_gp, sex, region, july1_2020, aug1_2020:june1_2021, july1_2021) %>%
  pivot_longer(july1_2020:july1_2021, names_to = "date", values_to = "pop_est") %>%
  mutate(date = lubridate::mdy(date))
```

Check that a few of these look correct (linear) when plotted. All look good.

```
# choose 3 sub-pops and plot monthly pop estimates
pop_df_agg %>% filter((region=="Alabama" & sex=="male" & age_gp=="15-24")|
                        (region=="New York" & sex=="total" & age_gp=="35-44")|
                        (region=="Florida" & sex=="female" & age_gp=="55-64")) %>%
  ggplot(aes(x=date, y=pop_est))+
  geom_point()+
  facet_wrap(.~region, scales="free_y", ncol=2,
             labeller = as_labeller(c("Alabama" = "Alabama, males, age 15-24",
                                      "Florida" = "Florida, females, age 55-64",
                                      "New York" = "New York, both sexes, age 35-44")))+
  ggtitle("Monthly population estimates obtained with linear interpolation")
```



Check if total sex data are equal to sum of male and female data

Before moving on, I'm interested to see whether the sum of population counts for males and females is close to the population count where sex is "Total". The code chunk below looks at the difference in these values by age group and state, and shows that there are no cases where the two values have an absolute difference greater than 0.0001. This means that there is no real difference if we aggregate sex ourselves when we compute quarterly Covid death rate estimates, or if we use the aggregated numbers to do so (i.e. data when sex = "Total").

```
# check whether the numbers in pop_df for sex="total" are much different than the sum of male and femal
# this code returns zero
pop_df_agg %>%
  mutate(total_flag = ifelse(sex=="total", 1, 0)) %>%
  group_by(total_flag, age_gp, region) %>%
  summarise(tot_pop = sum(pop_est)) %>%
  ungroup() %>%
  group_by(age_gp, region) %>%
  pivot_wider(names_from = total_flag, values_from = tot_pop) %>%
  mutate(diff = `1`-`0`) %>%
  filter(abs(diff) > 0.0001) %>%
  nrow()
```

```
## [1] 0
```

Now do the same for Covid deaths data. I do this based on the sum of deaths classified as: Covid, Pneumonia and Covid, and Pneumonia or Influenza or Covid. In contrast to what we saw for the population data, there are 2852 of rows out of 11271 where there is a notable difference between the sum of male and female counts and the sex-aggregated counts. On average, the sex-aggregated death count is higher by approximately 13.2 deaths. For this reason, I'll use the sex-aggregated level data when computing the quarterly estimates.

```
# check whether the numbers in deaths_df for sex="total" are much different than the sum of male and fe
# for 2852 rows (~25% of dataset), they are
# in part may be due to censoring of death counts
deaths_df %>%
  mutate(total_flag = as.factor(ifelse(sex=="total", 1, 0))) %>%
  select(total_flag, region:date, covid_deaths, pneumonia_covid_deaths, pneumonia_influenza_covid_death
  mutate(across(covid_deaths:pneumonia_influenza_covid_deaths, ~ ifelse(is.na(.), 0, .))) %>%
  mutate(deaths = covid_deaths+pneumonia_covid_deaths+pneumonia_influenza_covid_deaths) %>%
  group_by(total_flag, age_gp, region, date) %>%
  summarise(tot_deaths = sum(deaths)) %>%
  ungroup() %>%
  pivot_wider(names_from = total_flag, values_from = tot_deaths) %>%
  mutate(diff = `1`-`0`) %>%
  filter(abs(diff) > 0.0001) %>%
  summarise(n_row = n(),
            avg_diff = mean(diff))
```

```
## # A tibble: 1 x 2
##   n_row avg_diff
##   <int>    <dbl>
## 1  2852     13.2
```

Merge population and deaths data

Merge the population data and deaths data on age group, date, region, and sex. Before doing so, re-code the "Under 1 year" age group in the deaths data to "<1", to match the population data.

```
# recode under 1 year to <1 to match pop data
deaths_df <- deaths_df %>% mutate(age_gp = ifelse(age_gp=="Under 1 year", "<1", age_gp))

# merge population and deaths data
covid_df <- left_join(pop_df_agg, deaths_df, by=c("age_gp", "date", "region", "sex"))

# display first few rows
head(covid_df)
```

```
## # A tibble: 6 x 11
##   age_gp sex   region date       pop_est total~1 covid~2 pneum~3 pneum~4 influ~5
##   <chr>  <chr> <chr>  <date>       <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 <1     fema~ Alaba~ 2020-07-01  27565       20       0       0       0       0
## 2 <1     fema~ Alaba~ 2020-08-01  27531.      18       0       0       0       0
## 3 <1     fema~ Alaba~ 2020-09-01  27498.      16      NA      NA      NA       0
## 4 <1     fema~ Alaba~ 2020-10-01  27464.      13       0       0       0       0
## 5 <1     fema~ Alaba~ 2020-11-01  27431.      13       0       0       0       0
## 6 <1     fema~ Alaba~ 2020-12-01  27397.      NA       0       0       0       0
## # ... with 1 more variable: pneumonia_influenza_covid_deaths <dbl>, and
## #   abbreviated variable names 1: total_deaths, 2: covid_deaths,
## #   3: pneumonia_deaths, 4: pneumonia_covid_deaths, 5: influenza_deaths
```

Compute quarterly death rate estimates and compare them to CDC's rates

Using our clean population and Covid deaths data, let's now compute quarterly death rate estimates by state to ensure our numbers are similar to those published on the CDC website using the following method:

- Categorize months into quarters
- Create total deaths variable for death rate computation equal to the sum of Covid deaths, Covid *and* Pneumonia deaths, and Covid *or* Influenza *or* Pneumonia deaths
- For each age group, region, and quarter grouping, divide the total death count by the population estimate for the second month in that quarter to obtain the age-specific mortality rate
  - Note: we use the population estimate from the second month of that quarter in accordance with the technical note on the CDC website
  - Note: we multiply this death rate by 100,000 as published death rates we will compare to are per 100,000 population
- Obtain the age-adjusted death rate using the direct standardization approach as outlined in the CDC technical note (more on this later).

```
# create quarter variable (ordered factor)
quarterly_df <- covid_df %>%
  mutate(quarter = case_when(as.character(date) %in% c("2020-07-01", "2020-08-01", "2020-09-01") ~ "Q3_2
                             as.character(date) %in% c("2020-10-01", "2020-11-01", "2020-12-01") ~ "Q4_2
                             as.character(date) %in% c("2021-01-01", "2021-02-01", "2021-03-01") ~ "Q1_2
                             as.character(date) %in% c("2021-04-01", "2021-05-01", "2021-06-01") ~ "Q2_2
  mutate(quarter = factor(quarter, c("Q3_2020", "Q4_2020", "Q1_2021", "Q2_2021"), ordered = TRUE)) %>%
  # remove 2021-07-01 data b/c we don't have data from other months in that quarter
  filter(!is.na(quarter))
```

```
# create total deaths variable that we will use to compute age-specific death rates
# just using sex=total b/c numbers are pretty different for the covid deaths data
quarterly_df <- quarterly_df %>% filter(sex=="total") %>%
  mutate(across(total_deaths:pneumonia_influenza_covid_deaths, ~ ifelse(is.na(.x), 0, .x))) %>%
  mutate(deaths = covid_deaths + pneumonia_covid_deaths + pneumonia_influenza_covid_deaths) %>%
  select(region, quarter, date, age_gp, pop_est, deaths)
```

As mentioned, for the quarterly estimates, the CDC uses the population from the second month of the quarter. So let's use that as our denominator for age-specific death rate, and add up deaths from the all 3 months of the quarter as the numerator.

```
# obtain quarterly age-specific death rates per 100,000 population
quarterly_df <- quarterly_df %>% group_by(region, age_gp, quarter) %>%
  summarise(deaths = sum(deaths),
            pop_est = pop_est[date==median(date)]) %>%
  ungroup() %>%
  mutate(death_rate = 100000*(deaths/pop_est))
```

Now, let's add the projected 2000 U.S. standard population to the data (recall we're trying with 3 different ones), and compute standardized deaths by multiplying the age-specific death rate by the standard population for each age/sex/state group. I will be doing this with all 3 standard population datasets to see which produces the closest values to those published. To clean the Census data, I referred to their file layout documentation.

Note that the data for the year 2000 from the report cited by the CDC has slightly different standard population values than those on the census website. This finding aligns with the statement in this report: "published sums of the single years of age do not necessarily add to the published 5-year totals".

```
# read in population projections data for 2000
pop_proj_2000 <- read_table("../data/2000_pop_proj.txt", col_names = FALSE) %>%
  rename(series = X1,
         year = X2,
         age = X3,
         total_pop = X4,
         male_pop = X5,
         female_pop = X6) %>%
  select(year:female_pop)

# read in population projections data for 2021
pop_proj_2021 <- read_table("../data/2021_pop_proj.txt", col_names = FALSE) %>%
  rename(series = X1,
         year = X2,
         age = X3,
         total_pop = X4,
         male_pop = X5,
         female_pop = X6) %>%
  select(year:female_pop) %>%
  filter(year==2021)
```

```
# sanity check that the sum of all ages is equal to the total ages row
# 2000
stopifnot(setdiff(pop_proj_2000 %>% filter(age!=999) %>% summarise(across(total_pop:female_pop, ~ sum(.
                  pop_proj_2000 %>% filter(age==999) %>% select(total_pop:female_pop)) %>% nrow() == 0)
# 2021
stopifnot(setdiff(pop_proj_2021 %>% filter(age!=999) %>% summarise(across(total_pop:female_pop, ~ sum(.
                  pop_proj_2021 %>% filter(age==999) %>% select(total_pop:female_pop)) %>% nrow() == 0)

# sanity check that the sum of male and female equals the total pop values
# 2000
stopifnot(pop_proj_2000 %>% mutate(male_plus_female = male_pop+female_pop) %>%
            filter(total_pop!=male_plus_female) %>% nrow() == 0)
# 2021
stopifnot(pop_proj_2021 %>% mutate(male_plus_female = male_pop+female_pop) %>%
            filter(total_pop!=male_plus_female) %>% nrow() == 0)
```

```
# aggregate pop projections to match age groups and create new dataframe with these aggregates for both
std_pop_proj <- pop_proj_2000 %>% filter(age!=999) %>%
    mutate(age_gp = case_when(age <= 0 ~ "<1",
                            age > 0 & age <= 4 ~ "1-4",
                            age > 4 & age <= 14 ~ "5-14",
                            age > 14 & age <= 24 ~ "15-24",
                            age > 24 & age <= 34 ~ "25-34",
                            age > 34 & age <= 44 ~ "35-44",
                            age > 44 & age <= 54 ~ "45-54",
                            age > 54 & age <= 64 ~ "55-64",
                            age > 64 & age <= 74 ~ "65-74",
                            age > 74 & age <= 84 ~ "75-84",
                            age >= 85 ~ "85+")) %>%
  group_by(age_gp) %>%
  summarise(census_pop_2000 = sum(total_pop)) %>%
  left_join(., pop_proj_2021 %>% filter(age!=999) %>%
    mutate(age_gp = case_when(age <= 0 ~ "<1",
                            age > 0 & age <= 4 ~ "1-4",
                            age > 4 & age <= 14 ~ "5-14",
                            age > 14 & age <= 24 ~ "15-24",
                            age > 24 & age <= 34 ~ "25-34",
                            age > 34 & age <= 44 ~ "35-44",
                            age > 44 & age <= 54 ~ "45-54",
                            age > 54 & age <= 64 ~ "55-64",
                            age > 64 & age <= 74 ~ "65-74",
                            age > 74 & age <= 84 ~ "75-84",
                            age >= 85 ~ "85+")) %>%
  group_by(age_gp) %>%
  summarise(census_pop_2021 = sum(total_pop)),
  by = "age_gp")

# add in values from Table V (pg 78) of CDC report https://www.cdc.gov/nchs/data/nvsr/nvsr69/nvsr69-13-
std_pop_proj <- left_join(std_pop_proj,
                          tibble(age_gp = c("<1", "1-4", "5-14", "15-24", "25-34", "35-44",
                                            "45-54", "55-64", "65-74", "75-84", "85+"),
                                 cdc_pop_2000 = c(3794901, 15191619, 39976619, 38076743, 37233437, 4465
                                                  37030152, 23961506, 18135514, 12314793, 4259173)),
                          by = "age_gp")

# code age group variable as ordered factor
std_pop_proj <- std_pop_proj %>%
  mutate(age_gp = factor(age_gp,
                         c("<1", "1-4", "5-14", "15-24", "25-34", "35-44", "45-54", "55-64", "65-74", "
                         ordered = TRUE)) %>%
  arrange(age_gp)

# display standard pop proj table
std_pop_proj
```

```
## # A tibble: 11 x 4
##    age_gp census_pop_2000 census_pop_2021 cdc_pop_2000
##    <ord>            <dbl>           <dbl>        <dbl>
##  1 <1             3788840         4473879      3794901
##  2 1-4           15076601        17602806     15191619
##  3 5-14          39689314        42964863     39976619
##  4 15-24         38414789        42414868     38076743
##  5 25-34         37440476        42889212     37233437
##  6 35-44         44893767        41311458     44659185
##  7 45-54         37166202        38449295     37030152
##  8 55-64         24001096        41915474     23961506
##  9 65-74         18188857        32667055     18135514
## 10 75-84         12334552        15919804     12314793
## 11 85+            4311884         6858887      4259173
```

```
# add the 3 standard population variables to quarterly_df, and compute standardized deaths for each
quarterly_df <- left_join(quarterly_df, std_pop_proj, by="age_gp") %>%
  select(quarter, region, age_gp, death_rate, cdc_pop_2000, census_pop_2000, census_pop_2021) %>%
  mutate(cdc_deaths_2000 = death_rate*cdc_pop_2000,
         census_deaths_2000 = death_rate*census_pop_2000,
         census_deaths_2021 = death_rate*census_pop_2021)
```

```
# sanity check that there is only one distinct standard population value per data source per age group
stopifnot(quarterly_df %>%
            select(age_gp, cdc_pop_2000, census_pop_2000, census_pop_2021) %>%
            group_by(age_gp) %>%
            distinct() %>%
            summarise(n=n()) %>%
            filter(n!=1) %>% nrow() == 0)
```

Next, we obtain the standard death rate by dividing the sum of the standardized deaths across all age groups by the sum of the standard population across all age groups.

```
# compute standardized death rate using 3 standard pop options
quarterly_age_adj <- quarterly_df %>%
  group_by(quarter, region) %>%
  summarise(death_rate_cdc_2000 = sum(cdc_deaths_2000)/sum(cdc_pop_2000),
            death_rate_census_2000 = sum(census_deaths_2000)/sum(census_pop_2000),
            death_rate_census_2021 = sum(census_deaths_2021)/sum(census_pop_2021)) %>%
  ungroup()

# display first few rows
head(quarterly_age_adj)
```

```
## # A tibble: 6 x 5
##   quarter region     death_rate_cdc_2000 death_rate_census_2000 death_rate_cen~1
##   <ord>   <chr>                    <dbl>                  <dbl>            <dbl>
## 1 Q3_2020 Alabama                 101.                   102.             128.
## 2 Q3_2020 Alaska                    1.13                   1.13             1.71
## 3 Q3_2020 Arizona                 103.                   104.             128.
## 4 Q3_2020 Arkansas                 84.6                   84.8            107.
## 5 Q3_2020 California               64.7                   64.9             81.1
## 6 Q3_2020 Colorado                 21.9                   22.0             27.6
## # ... with abbreviated variable name 1: death_rate_census_2021
```

Now let's read in the quarterly Covid death rates published by the CDC, and clean the data to match ours for easier comparison.

```
# read in the Quarterly Provisional Death Rates for COVID-19 from the CDC to compare our numbers to
### NOTE - these data are not on our repo but can be downloaded from https://www.cdc.gov/nchs/pressroom

quarterly_cdc <- read_csv("~/Desktop/RA/fb-migration/data/cdc-deaths-data/cdc_quarterly_covid_deaths.cs

# rename variables to match ours, add full state name column
quarterly_cdc <- quarterly_cdc %>% rename(quarter = Quarters,
                                          state_ab = STATE,
                                          death_rate_official = RATE) %>%
  mutate(region = state.name[match(state_ab, state.abb)],
         quarter = str_replace(quarter, " ", "_")) %>%
  select(region, quarter, death_rate_official) %>%
  filter(quarter %in% c("Q3_2020", "Q4_2020", "Q1_2021", "Q2_2021")) %>%
  mutate(quarter = factor(quarter, c("Q3_2020", "Q4_2020", "Q1_2021", "Q2_2021"), ordered = TRUE))
```

Create a dataframe comparing our quarterly estimates and the CDCs. Discrepancies may be partially due to the numerator computation which involves a choice in terms of death categorization. It's unclear based on the CDCs website exactly which deaths are used to compute their estimates.

```
# create dataframe with CDCs published quarterly estimates and the ones we computed
quarterly_compare <- left_join(quarterly_age_adj, quarterly_cdc, by=c("region", "quarter")) %>%
  pivot_longer(death_rate_cdc_2000:death_rate_census_2021, names_to = "std_pop_source", values_to = "dea
  select(quarter, region, std_pop_source, death_rate_est, death_rate_official) %>%
  mutate(diff = death_rate_official - death_rate_est)
```

Compare the differences from the official death rates across standard population data choice. First, let's look at the difference between the two standard population distributions for the year 2000. The difference in resulting quarterly death rate estimates are minimal, with the biggest observed absolute difference equal to ~1.15.

```
# look at avg absolute difference between the two 2000 standard pop data sources - quite minimal
quarterly_compare %>% filter(std_pop_source!="death_rate_census_2021") %>%
  select(quarter, region, std_pop_source, death_rate_est) %>%
  pivot_wider(names_from = std_pop_source, values_from = death_rate_est) %>%
  mutate(diff = death_rate_cdc_2000-death_rate_census_2000) %>%
  arrange(desc(abs(diff))) %>%
  slice(1:5)
```

```
## # A tibble: 5 x 5
##   quarter region       death_rate_cdc_2000 death_rate_census_2000   diff
##   <ord>   <chr>                      <dbl>                  <dbl>  <dbl>
## 1 Q4_2020 North Dakota                324.                   325. -1.15
## 2 Q4_2020 South Dakota                304.                   305. -1.10
## 3 Q4_2020 Iowa                        196.                   197. -0.760
## 4 Q4_2020 Indiana                     200.                   201. -0.707
## 5 Q4_2020 Kansas                      190.                   191. -0.643
```

Evidently, the rates produced using the 2021 census standard population projection data have the smallest average absolute difference from the official estimates.

```
# average overall absolute difference from official rates - 2021 census std pop projection is smallest
quarterly_compare %>% group_by(std_pop_source) %>%
  summarise(avg_abs_diff = mean(abs(diff), na.rm=T))
```

```
## # A tibble: 3 x 2
##   std_pop_source         avg_abs_diff
##   <chr>                         <dbl>
## 1 death_rate_cdc_2000           22.1
## 2 death_rate_census_2000        21.9
## 3 death_rate_census_2021         8.31
```

Further, I had a look at which standard population data source produced the rate with the smallest mean absolute difference from the official estimates by region. The 2021 standard population data did so in 49 out of 50 states.

```
# look at the std pop source that produced the smallest average absolute difference from the official d
# the 2021 census data did so for 49 of 50 regions we considered
quarterly_compare %>% group_by(std_pop_source, region) %>%
  summarise(avg_abs_diff = mean(abs(diff), na.rm=T)) %>%
  group_by(region) %>%
  filter(avg_abs_diff==min(avg_abs_diff)) %>%
  group_by(std_pop_source) %>%
  summarise(n_min_diff=n())
```

```
## # A tibble: 2 x 2
##   std_pop_source         n_min_diff
##   <chr>                       <int>
## 1 death_rate_census_2000          1
## 2 death_rate_census_2021         49
```

Overall, the 2000 National Population Projections for the year 2021 from the Census website produced quarterly death rate estimates which matched the most closely with those published by the CDC. We will use that as the standard population for our analysis. 57.8% of our quarterly death rate estimates produced with the 2021 standard population are larger than those reported by the CDC.

```
# summary of rows where CDC est is larger or smaller than ours
quarterly_compare %>%
  filter(std_pop_source=="death_rate_census_2021") %>%
  select(quarter, region, diff) %>%
  mutate(cdc_larger = ifelse(diff > 0, 1, 0)) %>%
  filter(!is.na(cdc_larger)) %>%
  group_by(cdc_larger) %>%
  summarise(n=n()) %>%
  ungroup() %>%
  mutate(pct_rows = 100*(n/sum(n)))
```

```
## # A tibble: 2 x 3
##   cdc_larger     n pct_rows
##        <dbl> <int>    <dbl>
## 1          0   115     57.8
## 2          1    84     42.2
```

Prepare data for analysis with Facebook travel rates

Using our tidy dataset with population estimates and Covid death counts, we can add in our cleaned monthly Facebook travelers dataset and compute population adjustments.

```
# read in clean monthly travel rate data, so we can recompute death rates with our fb data
monthly_travel <- read_csv("../data/clean-fb-data/rates_age_monthly.csv", show_col_types = F)
```

Let's clean up this dataset for merging with our Covid deaths and population estimates dataset.

- Drop unnecessary variables
- Drop Canadian data
- Re-code sex to match how it is coded in our merged dataset
- Re-code the age groups to match those in our merged dataset (main difference: 13-29 to 18-29, hand-waving on this)
- Drop data from 01 June 2020 as it is not part of our analysis

```
# clean monthly travel data to prepare for merge with covid_df
travel_df <- monthly_travel %>%
  filter(!(region %in% c("Alberta", "British Columbia", "Ontario", "Quebec", "Prince Edward Island", "Nu
                         "Northwest Territories", "Newfoundland and Labrador", "Yukon", "Saskatchewan",
                         "Manitoba","New Brunswick", "Nova Scotia", "Canada"))) %>%
  select(age_gp, sex, region, date, monthly_trav, monthly_pop, loess_rate) %>%
  mutate(sex = ifelse(sex=="F", "female", "male"),
         age_gp = case_when(age_gp == "13-29" ~ "18-29",
                            age_gp == "30-50" ~ "30-49",
                            age_gp == "50-65" ~ "50-64")) %>%
  filter(date!="2020-06-01")
```

Let's now create a dataframe with death and population estimates that are aggregated to match the Facebook age groups. We are limited by the age groups present in the deaths data. The ones that match the Facebook data best are 18-29, 30-39, 40-49, and 50-64 so we will go with those.

```
# filter deaths data for correct age groups,
# remove aggregated sex level data, and
# compute total deaths variable to use in our death rate calculations
deaths_df_new <- deaths_df %>%
  filter(age_gp %in% c("18-29", "30-39", "40-49", "50-64")) %>%
  mutate(across(total_deaths:pneumonia_influenza_covid_deaths, ~ ifelse(is.na(.x), 0, .x))) %>%
  mutate(deaths = covid_deaths + pneumonia_covid_deaths + pneumonia_influenza_covid_deaths) %>%
  select(region, age_gp, sex, date, deaths)
```

```r
# combine 30-39 and 40-49 to match fb data
deaths_df_new <- deaths_df_new %>%
  mutate(age_gp = ifelse(age_gp=="30-39" | age_gp=="40-49", "30-49", age_gp)) %>%
  group_by(region, age_gp, sex, date) %>%
  summarise(deaths = sum(deaths)) %>%
  ungroup()
```

```r
# create age groups and aggregate population estimates
pop_df_new <- pop_df %>%
  filter(!str_detect(age_gp, "\\+|[:alpha:]")) %>%
  mutate(age_gp = as.numeric(age_gp)) %>%
  rename(age = age_gp) %>%
  mutate(age_gp = case_when(age > 17 & age <= 29 ~ "18-29",
                            age > 29 & age <= 49 ~ "30-49",
                            age > 49 & age <= 64 ~ "50-64")) %>%
  filter(!is.na(age_gp)) %>%
  select(-age) %>%
  group_by(sex, region, age_gp) %>%
  summarise(across(july1_2020:july1_2021, sum)) %>%
  ungroup()
```

Now we perform linear interpolation just as we did before.

```r
# perform linear interpolation to get monthly population estimates by state, sex, and age group
pop_df_new <- pop_df_new %>%
  mutate(pop_diff = july1_2021-july1_2020,
         pop_diff_12 = pop_diff/12) %>%
  mutate(aug1_2020 = july1_2020 + pop_diff_12,
         sept1_2020 = aug1_2020 + pop_diff_12,
         oct1_2020 = sept1_2020 + pop_diff_12,
         nov1_2020 = oct1_2020 + pop_diff_12,
         dec1_2020 = nov1_2020 + pop_diff_12,
         jan1_2021 = dec1_2020 + pop_diff_12,
         feb1_2021 = jan1_2021 + pop_diff_12,
         march1_2021 = feb1_2021 + pop_diff_12,
         april1_2021 = march1_2021 + pop_diff_12,
         may1_2021 = april1_2021 + pop_diff_12,
         june1_2021 = may1_2021 + pop_diff_12) %>%
  select(age_gp, sex, region, july1_2020, aug1_2020:june1_2021, july1_2021) %>%
  pivot_longer(july1_2020:july1_2021, names_to = "date", values_to = "pop_est") %>%
  mutate(date = lubridate::mdy(date))
```

Combine the cleaned deaths and population estimates data.

```r
# merge deaths and population estimates data
covid_df_new <- left_join(pop_df_new, deaths_df_new, by=c("region", "age_gp", "sex", "date"))
```

Now, merge the Facebook traveller data with the `covid_df_new` dataframe we created above.

```r
# merge fb travel data with covid death/pop data, select variables of interest
fb_analysis_df <- left_join(covid_df_new, travel_df, by=c("region", "age_gp", "sex", "date")) %>%
  select(region, age_gp, sex, date, loess_rate, deaths, pop_est)

# display first few rows
head(fb_analysis_df)
```

```
## # A tibble: 6 x 7
##   region  age_gp sex    date       loess_rate deaths pop_est
##   <chr>   <chr>  <chr>  <date>          <dbl>  <dbl>   <dbl>
## 1 Alabama 18-29  female 2020-07-01     0.0253      0 393656
## 2 Alabama 18-29  female 2020-08-01     0.0245      0 393441.
## 3 Alabama 18-29  female 2020-09-01     0.0231      0 393227.
## 4 Alabama 18-29  female 2020-10-01     0.0215      0 393012.
## 5 Alabama 18-29  female 2020-11-01     0.0216      0 392798.
## 6 Alabama 18-29  female 2020-12-01     0.0225      0 392583.
```

Compute age-specific mortality rates

Using our `fb_analysis_df` data, we can obtain age-specific mortality rates with a traveller adjustment to the denominator (i.e population estimate).

```
# compute age specific mortality rates with and without traveller adjustment
fb_age_rates <- fb_analysis_df %>%
  mutate(pop_est_adj = (pop_est*(1+loess_rate)),
         death_rate_adj = 100000*(deaths/pop_est_adj),
         death_rate = 100000*(deaths/pop_est)) %>%
  # death rate adj is NA for sex=total b/c travel rate is not aggregated by sex
  filter(!is.na(death_rate_adj))

# export to CSV
# write_csv(fb_age_rates, "../data/age_specific_death_rates.csv")

# select variables of interest
fb_age_rates <- fb_age_rates %>% select(region, age_gp, sex, date, death_rate, death_rate_adj)
```

The average relative difference from the travel adjustment is approximately -2.4%, and the average absolute difference is approximately -0.21.

```
# compute avg relative and absolute difference
fb_age_rates %>%
  mutate(relative_diff = 100*(death_rate_adj-death_rate)/death_rate,
         abs_diff = death_rate_adj-death_rate) %>%
  summarise(avg_relative_diff = mean(relative_diff, na.rm = T),
            avg_abs_diff = mean(abs_diff, na.rm = T)) %>%
  mutate(across(avg_relative_diff:avg_abs_diff, ~ round(., 4)))
```

```
## # A tibble: 1 x 2
##   avg_relative_diff avg_abs_diff
##               <dbl>        <dbl>
## 1             -2.43       -0.211
```

Intuitively we expect to see these decreases in death rate with the traveler adjustment, because we're making the denominator bigger in the process. The magnitude of these changes vary by region. Nevada has the average relative difference with the greatest magnitude of -3.8%. New York has the average relative difference with the smallest magnitude of -1.4%.

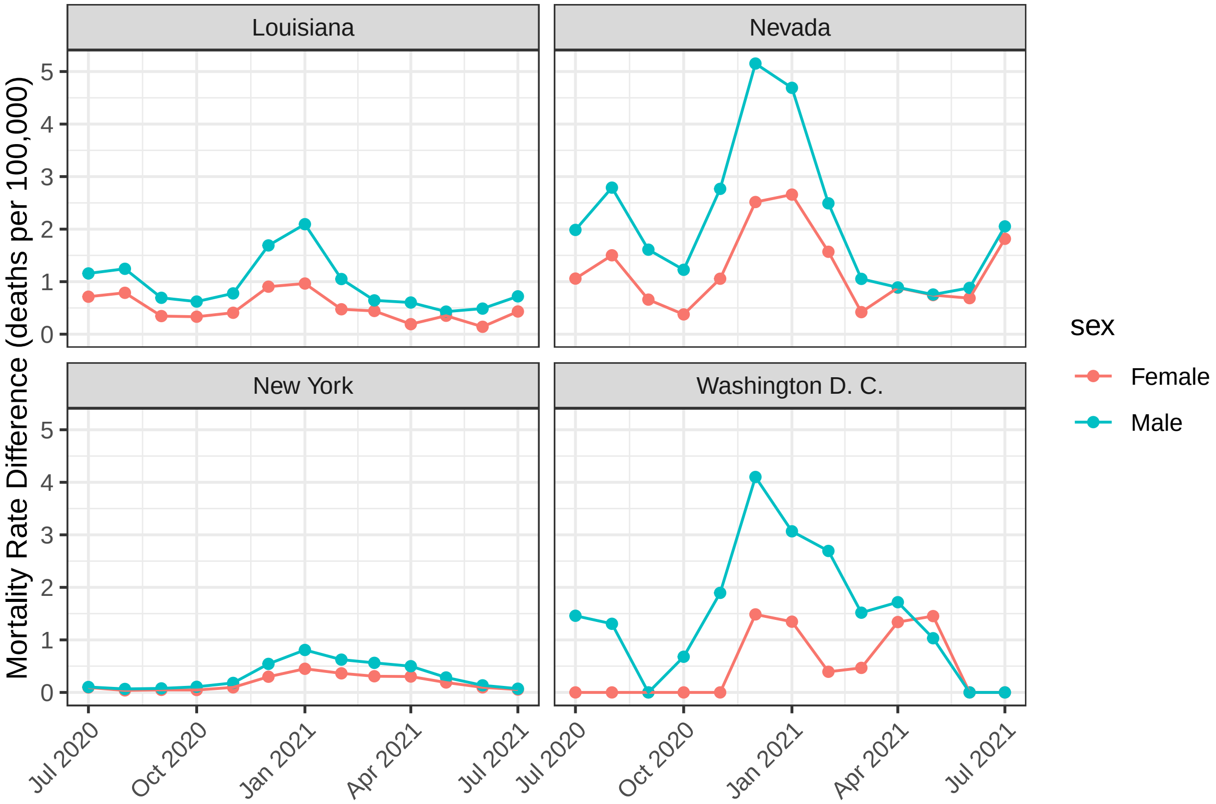
```
# look at average relative and absolute difference with the adjustment, by region
fb_age_rates %>%
  mutate(relative_diff = 100*(death_rate_adj-death_rate)/death_rate,
         abs_diff = death_rate_adj-death_rate) %>%
  group_by(region) %>%
  summarise(avg_relative_diff = mean(relative_diff, na.rm = T),
            avg_abs_diff = mean(abs_diff, na.rm = T)) %>%
  mutate(across(avg_relative_diff:avg_abs_diff, ~ round(., 4)))
```

```
## # A tibble: 51 x 3
##    region      avg_relative_diff avg_abs_diff
##    <chr>                   <dbl>        <dbl>
##  1 Alabama                 -2.49      -0.347
##  2 Alaska                  -2.44      -0.0049
##  3 Arizona                 -2.82      -0.574
##  4 Arkansas                -2.54      -0.349
##  5 California              -1.65      -0.263
##  6 Colorado                -2.65      -0.160
##  7 Connecticut             -1.59      -0.0708
##  8 Delaware                -3.30      -0.120
##  9 Florida                 -2.93      -0.360
## 10 Georgia                 -3.25      -0.398
## # ... with 41 more rows
```

```
# same as above but just look at max and min average relative diff
fb_age_rates %>%
  mutate(relative_diff = 100*(death_rate_adj-death_rate)/death_rate,
         abs_diff = death_rate_adj-death_rate) %>%
  group_by(region) %>%
  summarise(avg_relative_diff = mean(relative_diff, na.rm = T),
            avg_abs_diff = mean(abs_diff, na.rm = T)) %>%
  mutate(across(avg_relative_diff:avg_abs_diff, ~ round(., 4))) %>%
  filter(avg_relative_diff== min(avg_relative_diff, na.rm = T) |
           avg_relative_diff == max(avg_relative_diff, na.rm = T))
```

```
## # A tibble: 2 x 3
##   region   avg_relative_diff avg_abs_diff
##   <chr>                <dbl>        <dbl>
## 1 Nevada               -3.81      -0.645
## 2 New York             -1.40      -0.0904
```

```
# plot differences for 50-64 group for a few regions, by sex
fb_age_rates |>
  filter(age_gp=="50-64", region=="Louisiana"|region=="Washington D. C."|region=="New York"|region=="Ne
  mutate(diff = (death_rate - death_rate_adj),
         sex = factor(sex, levels = c("female", "male"), labels = c("Female", "Male"))) |>
  ggplot(aes(date, diff, col = sex)) +
  geom_point()+
  geom_line()+
  facet_wrap(~region)+
  labs(y = "Mortality Rate Difference (deaths per 100,000)", x = "")+
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust=1))
```


Louisiana
Nevada
New York
Washington D. C.
Mortality Rate Difference (deaths per 100,000)
0
1
2
3
4
5
Jul 2020
Oct 2020
Jan 2021
Apr 2021
Jul 2021
sex
Female
Male