COVID 19 Analysis – Part 2

Meisam Yousefi

2023-03-23

Required Packages

Part 1 - Basic Exploration of US Data The New York Times (the Times) has aggregated reported COVID-19 data from state and local governments and health departments since 2020 and provides public access through a repository on GitHub. One of the data sets provided by the Times is county-level data for cumulative cases and deaths each day. This will be your primary data set for the first two parts of your analysis.

County-level COVID data from 2020, 2021, and 2022 has been imported below. Each row of data reports the cumulative number of cases and deaths for a specific county each day. A FIPS code, a standard geographic identifier, is also provided which you will use in Part 2 to construct a map visualization at the county level for a state.

Additionally, county-level population estimates reported by the US Census Bureau has been imported as well. You will use these estimates to caluclate statistics per 100,000 people.

```
# Import New York Times COVID-19 data
# Import Population Estimates from US Census Bureau

us_counties_2020 <- read_csv("https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counties
```

```
## Rows: 884737 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): county, state, fips
## dbl  (2): cases, deaths
## date (1): date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
us_counties_2021 <- read_csv("https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counties
```

```
## Rows: 1185373 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): county, state, fips
## dbl  (2): cases, deaths
## date (1): date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
us_counties_2022 <- read_csv("https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counties
```

```
## Rows: 1188042 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (3): county, state, fips
## dbl  (2): cases, deaths
## date (1): date
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
us_population_estimates <- read_csv("fips_population_estimates.csv")
```

```
## Rows: 6286 Columns: 7
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (2): STNAME, CTYNAME
## dbl (5): fips, STATE, COUNTY, Year, Estimate
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

Question 1 Your first task is to combine and tidy the 2020, 2021, and 2022 COVID data sets and find the total deaths and cases for each day since March 15, 2020 (2020-03-15). The data sets provided from the NY Times also includes statistics from Puerto Rico, a US territory. You may remove these observations from the data as they will not be needed for your analysis. Once you have tidied the data, find the total COVID-19 cases and deaths since March 15, 2020. Write a sentence or two after the code block communicating your results. Use inline code to include the `max_date`, `us_total_cases`, and `us_total_deaths` variables. To write inline code use `r`.

```
# Combine the three 2020, 2021, and 2022 COVID data sets using rbind

us_counties_total <- rbind(us_counties_2020, us_counties_2021, us_counties_2022)

# Removing "Puerto Rico" from the states list, and removing dates prior to March 15th 2020

us_counties_total <- us_counties_total %>% dplyr::filter(state != "Puerto Rico",
                                                         date > "2020-03-14")

# Summarizing data for total death and cases, per day

us_combined_total <- us_counties_total %>%
  group_by(date) %>%
  summarise("total_deaths" = sum(deaths),
            "total_cases" = sum(cases))

# Calculating values for the communication part
```

```
max_date <- max(us_combined_total$date) # replace the quotes with your code to find the most recent dat
us_total_cases <- us_combined_total$total_cases[us_combined_total$date == max_date]
us_total_deaths <- us_combined_total$total_deaths[us_combined_total$date == max_date]
```

Answer 1 Displaying the output final table:

```
us_combined_total
```

```
## # A tibble: 1,022 x 3
##    date       total_deaths total_cases
##    <date>            <dbl>       <dbl>
##  1 2020-03-15           68        3595
##  2 2020-03-16           91        4502
##  3 2020-03-17          117        5901
##  4 2020-03-18          162        8345
##  5 2020-03-19          212       12387
##  6 2020-03-20          277       17998
##  7 2020-03-21          359       24507
##  8 2020-03-22          457       33050
##  9 2020-03-23          577       43474
## 10 2020-03-24          783       53899
## # ... with 1,012 more rows
```

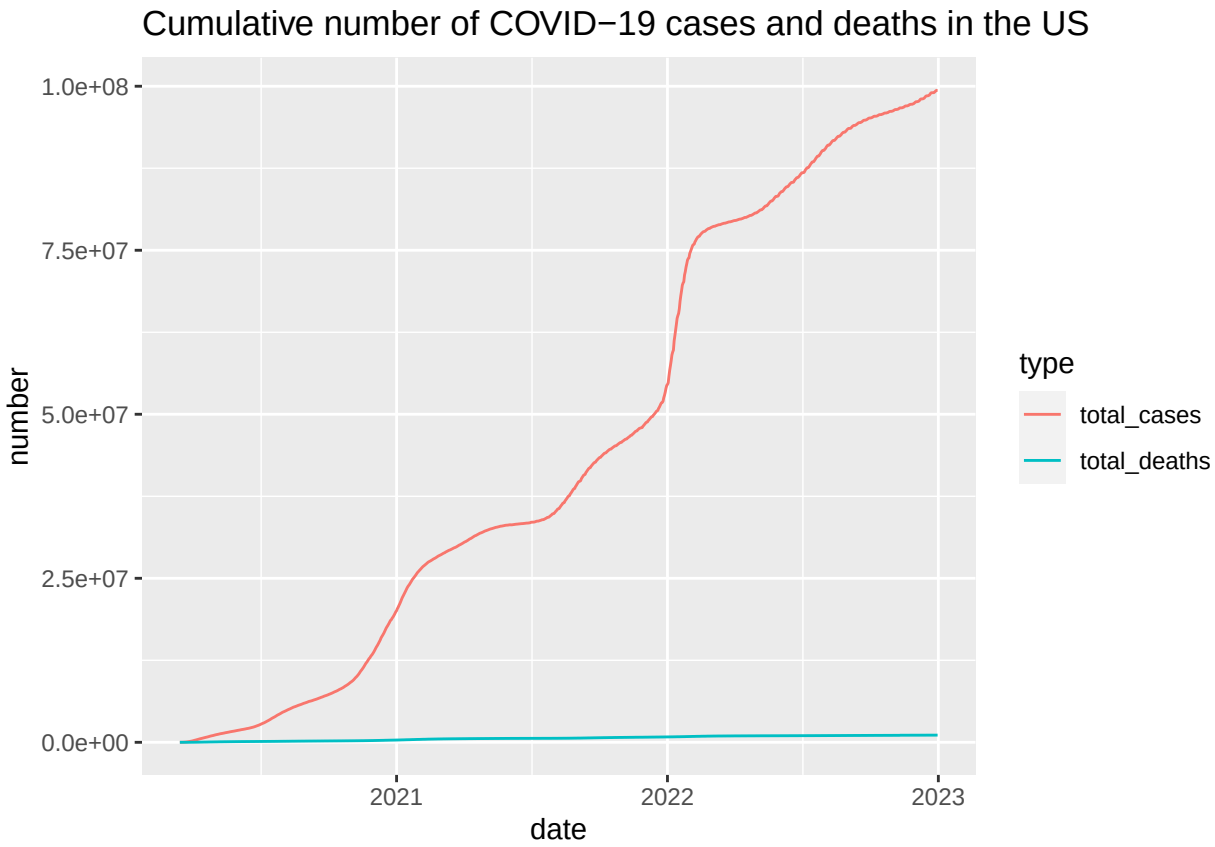
As of December 31, 2022, there has been a cumulative number of 9.9374764×10^7 individuals in the US who were diagnosed with COVID-19, and there has been 1.094296×10^6 deaths reported.

In this analysis we used the data from NYTimes on the daily number of cases and deaths in each county, from the begining of the pandemic until 2022-12-31.

Question 2 Create a visualization for the total number of deaths and cases in the US since March 15, 2020. Before you create your visualization, review the types of plots you can create using the ggplot2 library and think about which plots would be effective in communicating your results. After you have created your visualization, write a few sentences describing your visualization. How could the plot be interpreted? Could it be misleading?

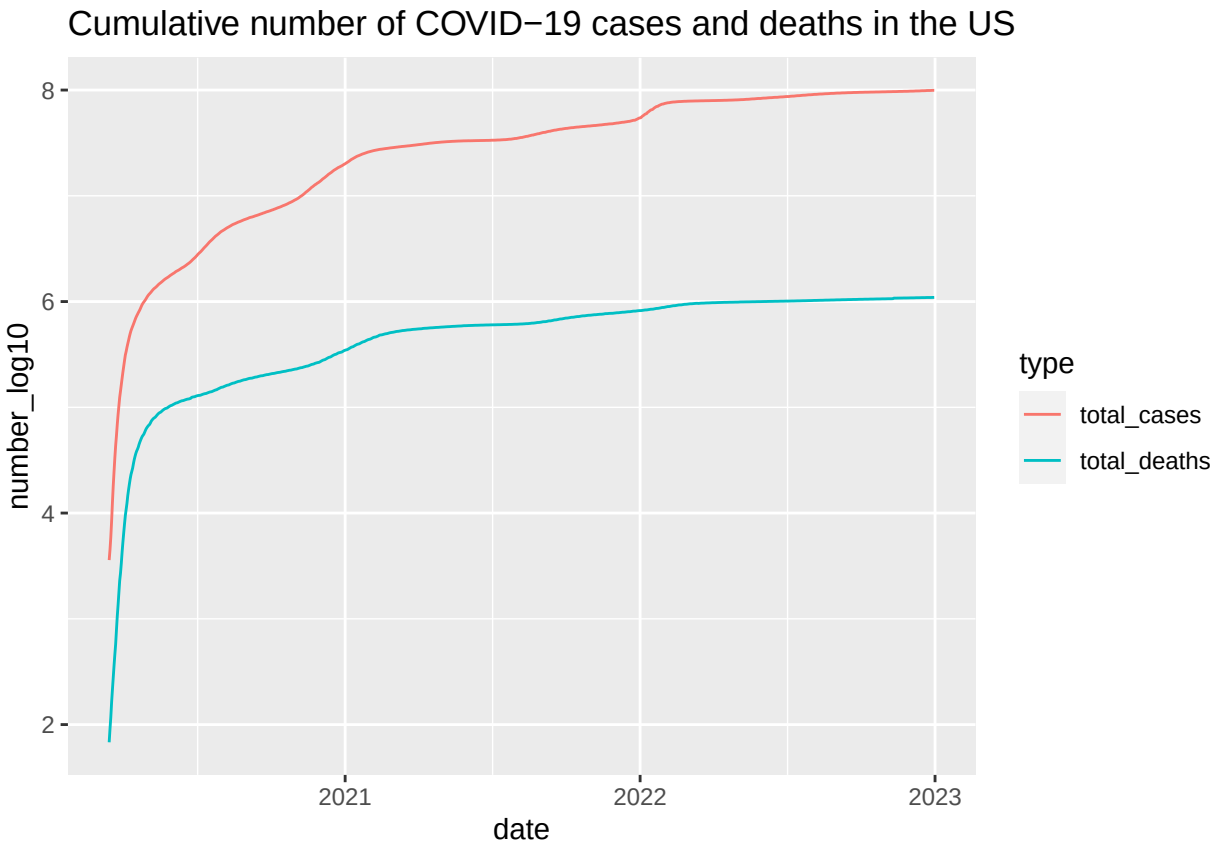
Answer 2 I'll present the data with a simple line-graph, with two separate lines one for the total number of deaths and the other for the total number of cases. To do so we might want to first pivot the table to the long format.

```
us_combined_total %>%
  pivot_longer(cols = -date, names_to = "type", values_to = "number") %>%
  ggplot(aes(x = date, y = number)) + geom_line(aes(color = type)) + labs(title = "Cumulative number of
```



The plot is effective in communicating the message and for the audience to get a grasp of the COVID-19 situation in the states until the end of the 2022, however there is probably one point which might be misleading: Since the actual number of the cases is orders of magnitude higher than the death, the death line seems to be static from this plot so the audience might think that the rate of the cases is getting higher than the deaths. We can overcome this by converting all numbers to log transformed:

```
us_combined_total %>%
  pivot_longer(cols = -date, names_to = "type", values_to = "number") %>%
  mutate(number_log10 = log10(number)) %>%
  ggplot(aes(x = date, y = number_log10)) + geom_line(aes(color = type)) + labs(title = "Cumulative numb
```



Now we can clearly see that the cumulative number of the deaths from COVID-19 is almost always 2 log lower than the total number of the cases, which brings us to an almost 1% chance of death from COVID-19 which was not growing as the pandemic progressed.

Question 3 While it is important to know the total deaths and cases throughout the COVID-19 pandemic, it is also important for local and state health officials to know the the number of new cases and deaths each day to understand how rapidly the virus is spreading. Using the table you created in Question 1, calculate the number of new deaths and cases each day and a seven-day average of new deaths and cases. Once you have organized your data, find the days that saw the largest number of new cases and deaths. Write a sentence or two after the code block communicating your results.

```
# Calculating the number of new deaths and cases each day and a seven day average of new deaths and cas

us_combined_2 <- us_combined_total %>% mutate(
  delta_deaths_1 = total_deaths - lag(total_deaths),
  delta_cases_1 = total_cases - lag(total_cases),
  delta_deaths_7 = zoo::rollmean(delta_deaths_1, k = 7, fill = NA, align = "right"),
  delta_cases_7 = zoo::rollmean(delta_cases_1, k = 7, fill = NA, align = "right")
)

us_combined_2
```

Answer 3

```
## # A tibble: 1,022 x 7
##    date       total_deaths total_cases delta_deaths_1 delta_ca~1 delta~2 delta~3
##    <date>            <dbl>       <dbl>          <dbl>      <dbl>   <dbl>   <dbl>
##  1 2020-03-15           68        3595             NA         NA    NA       NA
##  2 2020-03-16           91        4502             23        907    NA       NA
##  3 2020-03-17          117        5901             26       1399    NA       NA
##  4 2020-03-18          162        8345             45       2444    NA       NA
##  5 2020-03-19          212       12387             50       4042    NA       NA
##  6 2020-03-20          277       17998             65       5611    NA       NA
##  7 2020-03-21          359       24507             82       6509    NA       NA
##  8 2020-03-22          457       33050             98       8543    55.6   4208.
##  9 2020-03-23          577       43474            120      10424    69.4   5567.
## 10 2020-03-24          783       53899            206      10425    95.1   6857.
## # ... with 1,012 more rows, and abbreviated variable names 1: delta_cases_1,
## #   2: delta_deaths_7, 3: delta_cases_7
```

```
# Finding the days with the highest number of new cases and deaths

max_new_cases_date <- us_combined_2$date[us_combined_2$delta_cases_1 == max(us_combined_2$delta_cases_1

max_new_deaths_date <- us_combined_2$date[us_combined_2$delta_deaths_1 == max(us_combined_2$delta_deaths
```

We can see that the pandemic has not been less severe in 2022, as the highest daily number of new confirmed cases belongs to 2022-01-10, and the highest number of deaths happened on NA, 2022-11-11 with 1.2715×10^4 individuals died on that day.

Question 4 Create a new table, based on the table from Question 3, and calculate the number of new deaths and cases per 100,000 people each day and a seven day average of new deaths and cases per 100,000 people.

```
# Calculate the US total population in 2020, 2021 and 2022. Since the "population estimates" were only

us_population_total <- us_population_estimates %>% group_by(Year) %>% summarise(Population = sum(Estima

us_population_total <- rbind(us_population_total, c(2022, 2*(us_population_total$Population[2]) - us_po

# Dividing each statistics by the total population and then multiplying by 100,000

us_combined_3 <- us_combined_2 %>%
  mutate(across(-date, ~ case_when(date < "2021-01-01" ~ (.x*100000)/us_population_total$Population[us_
                                   date >= "2021-01-01" &  date < "2022-01-01" ~ (.x*100000)/us_populati
                                   date >= "2022-01-01" ~ (.x*100000)/us_population_total$Population[us_
  ))

us_combined_3
```

```
## # A tibble: 1,022 x 7
##    date       total_deaths total_cases delta_deaths_1 delta_ca~1 delta~2 delta~3
##    <date>            <dbl>       <dbl>          <dbl>      <dbl>   <dbl>   <dbl>
##  1 2020-03-15       0.0205        1.08       NA           NA     NA        NA
##  2 2020-03-16       0.0275        1.36        0.00694      0.274 NA        NA
##  3 2020-03-17       0.0353        1.78        0.00784      0.422 NA        NA
##  4 2020-03-18       0.0489        2.52        0.0136       0.737 NA        NA
```

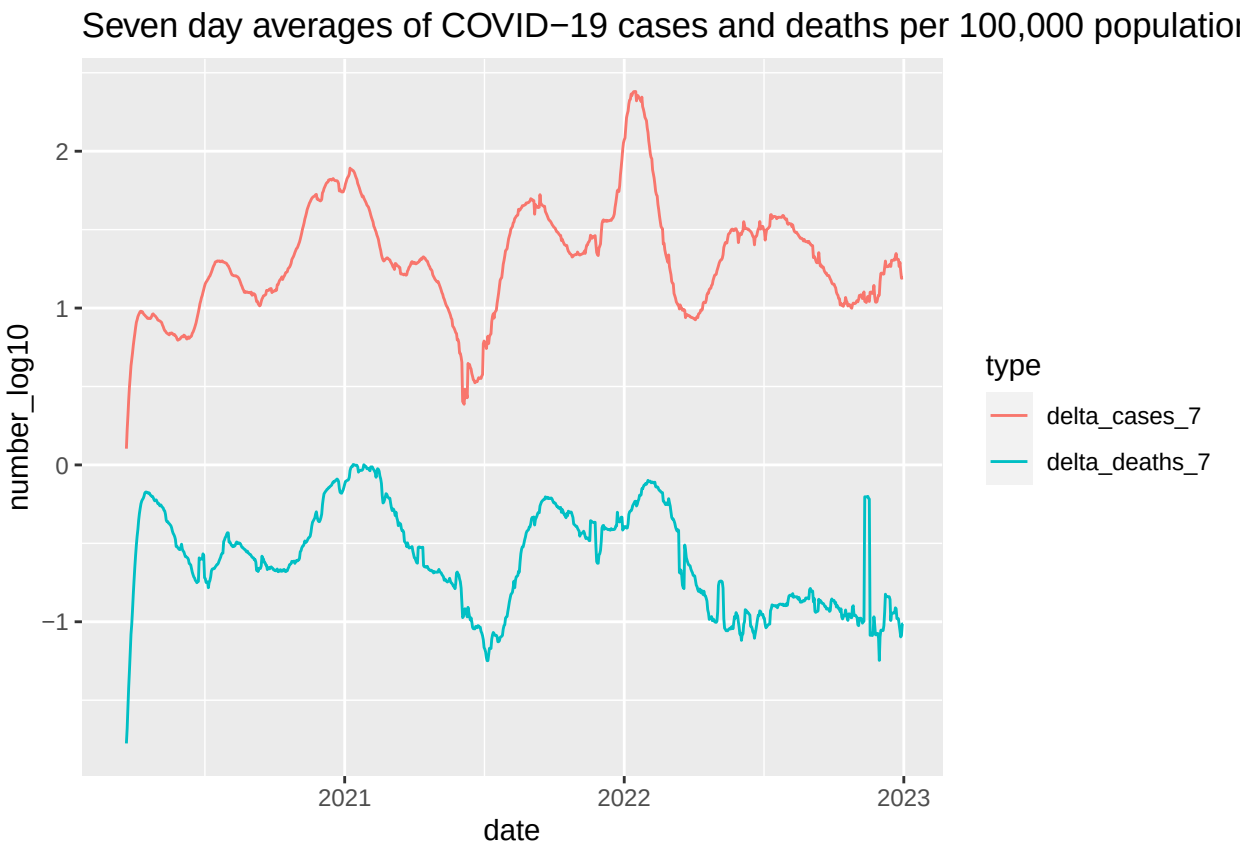
```
##  5 2020-03-19       0.0640        3.74        0.0151       1.22  NA        NA
##  6 2020-03-20       0.0836        5.43        0.0196       1.69  NA        NA
##  7 2020-03-21       0.108         7.39        0.0247       1.96  NA        NA
##  8 2020-03-22       0.138         9.97        0.0296       2.58   0.0168    1.27
##  9 2020-03-23       0.174        13.1         0.0362       3.14   0.0209    1.68
## 10 2020-03-24       0.236        16.3         0.0621       3.14   0.0287    2.07
## # ... with 1,012 more rows, and abbreviated variable names 1: delta_cases_1,
## #   2: delta_deaths_7, 3: delta_cases_7
```

Question 5 Create a visualization to compare the seven-day average cases and deaths per 100,000 people

```
us_combined_3 %>%
  pivot_longer(cols = ends_with("7"), names_to = "type", values_to = "number") %>%
  mutate(number_log10 = log10(number)) %>%
  ggplot(aes(x = date, y = number_log10)) + geom_line(aes(color = type)) + labs(title = "Seven day avera
```

Answer 5

```
## Warning: Removed 14 rows containing missing values (`geom_line()`).
```



By this we can see that again, the pattern of the COVID-19 cases and deaths are similar, with several waves of the pandemic from early 2020 until the end of 2022. However, notably, while the highest weekly average of case diagnosis is in early 2022, we see that the pick deaths belong to the early 2021 which probably shows the impact of vaccination programs on us national COVID-19 burden.

Part 2 - US State Comparison While understanding the trends on a national level can be helpful in understanding how COVID-19 impacted the United States, it is important to remember that the virus arrived in the United States at different times. For the next part of your analysis, you will begin to look at COVID related deaths and cases at the state and county-levels.

Question 1 Your first task in Part 2 is to determine the top 10 states in terms of total deaths and cases between March 15, 2020, and December 31, 2021.

Once you have both lists, briefly describe your methodology and your results.

```
# Determine the top 10 states in terms of total deaths and cases between March 15, 2020, and December 3

us_counties_DEC2021 <- us_counties_total %>%
  filter(date == "2021-12-31") %>%
  group_by(state, date) %>%
  summarise(total_deaths = sum(deaths),
            total_cases = sum(cases)) %>%
  arrange(desc(total_cases))
```

Answer 1

```
## `summarise()` has grouped output by 'state'. You can override using the
## `.groups` argument.
```

```
us_counties_DEC2021
```

```
## # A tibble: 55 x 4
## # Groups:   state [55]
##    state          date       total_deaths total_cases
##    <chr>          <date>            <dbl>       <dbl>
##  1 California     2021-12-31        76709     5515613
##  2 Texas          2021-12-31        76062     4574881
##  3 Florida        2021-12-31        62504     4166392
##  4 New York       2021-12-31        58993     3473970
##  5 Illinois       2021-12-31        31017     2154058
##  6 Pennsylvania   2021-12-31        36705     2036424
##  7 Ohio           2021-12-31        29447     2016095
##  8 Georgia        2021-12-31        30283     1798497
##  9 Michigan       2021-12-31        28984     1706355
## 10 North Carolina 2021-12-31        19436     1685504
## # ... with 45 more rows
```

Since our original data is cumulative sum, to find the total cases and deaths till the end of 2021 is to filter the date "2021-12-31". Then we summarise all the counties by the states.

Question 2 Determine the top 10 states in terms of deaths per 100,000 people and cases per 100,000 people between March 15, 2020, and December 31, 2021.

Once you have both lists, briefly describe your methodology and your results. Do you expect the lists to be different than the one produced in Question 1? Which method, total or per 100,000 people, is a better method for reporting the statistics?

```
# Determining state wise total population in year 2021
us_counties_estimates <- us_population_estimates %>% group_by(STNAME, Year) %>% summarise(population = s
```

```
## `summarise()` has grouped output by 'STNAME'. You can override using the
## `.groups` argument.
```

```
# Calculating deaths and cases per 100000 individulas per state, and ranking based on most cases per 10
us_counties_2 <- us_counties_DEC2021 %>%
  full_join(us_counties_estimates, by = c('state' = 'STNAME')) %>%
  mutate(deaths_per_100k = 100000*(total_deaths/population),
         cases_per_100k = 100000*(total_cases/population)) %>%
  dplyr::select(state, date, deaths_per_100k, cases_per_100k) %>%
  arrange(desc(cases_per_100k))
```

```
# Output
us_counties_2
```

```
## # A tibble: 55 x 4
## # Groups:   state [55]
##    state        date       deaths_per_100k cases_per_100k
##    <chr>        <date>               <dbl>          <dbl>
##  1 North Dakota 2021-12-31            265.         22482.
##  2 Alaska       2021-12-31            130.         21310.
##  3 Rhode Island 2021-12-31            280.         21093.
##  4 South Dakota 2021-12-31            278.         20014.
##  5 Wyoming      2021-12-31            264.         19979.
##  6 Tennessee    2021-12-31            296.         19783.
##  7 Kentucky     2021-12-31            269.         19173.
##  8 Florida      2021-12-31            287.         19128.
##  9 Utah         2021-12-31            113.         19088.
## 10 Wisconsin    2021-12-31            190.         19008.
## # ... with 45 more rows
```

We see that North Dakota has had the most number of cases per 100,000 population among all states, until the end of 2022

Question 3 Now, select a state and calculate the seven-day averages for new cases and deaths per 100,000 people. Once you have calculated the averages, create a visualization using ggplot2 to represent the data.

```
# Selecting the state of choice
stateName = "Colorado"

# Determining total population in year 2020 and 2021
state_estimates <- us_population_estimates %>%
  filter(STNAME == stateName) %>%
  group_by(Year) %>%
  summarise(population = sum(Estimate))

# Calculating the 7-day average of total death and cases in the state, per 100,000 individuals
state_estimates_2 <- us_counties_total %>%
  filter(state == stateName) %>%
```

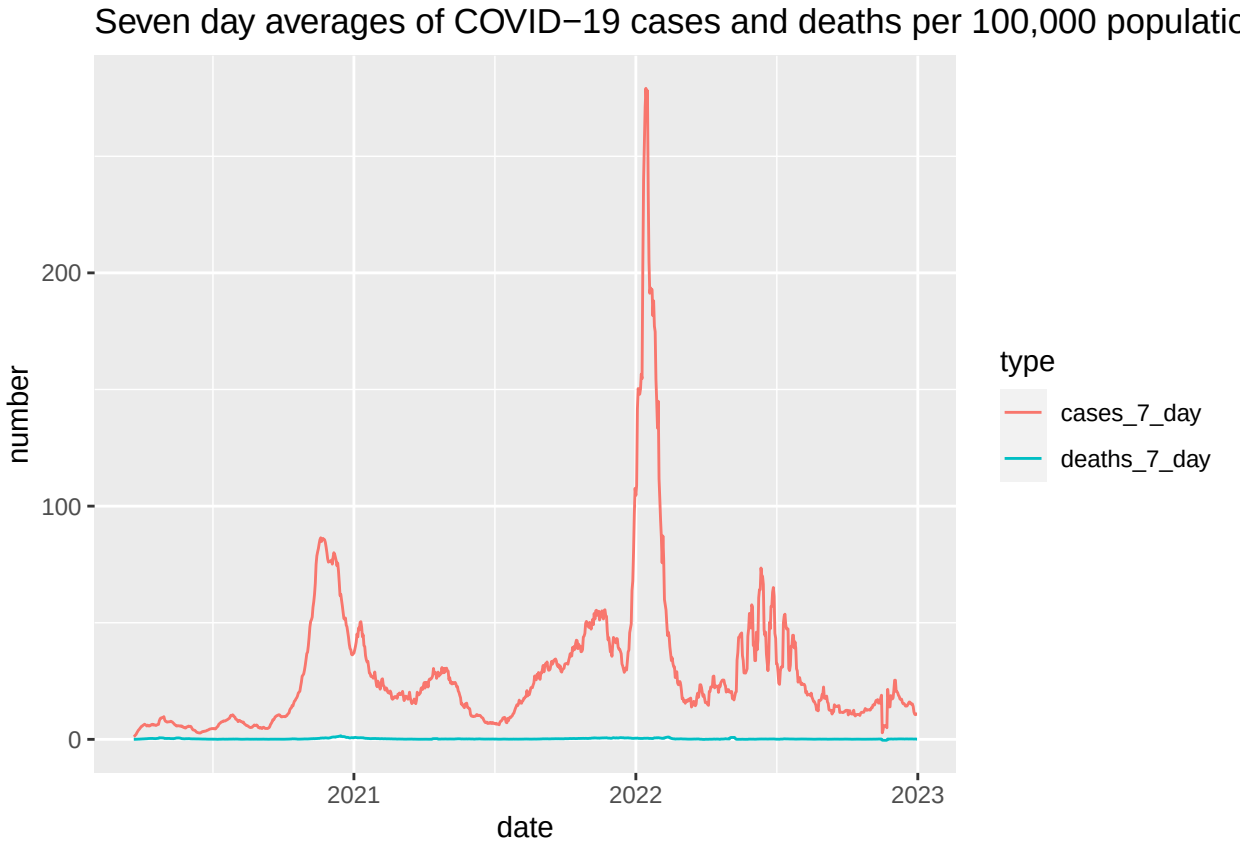
```
  group_by(date) %>%
  summarise("total_deaths" = sum(deaths),
            "total_cases" = sum(cases)) %>%
  mutate(
    population = case_when(date < base::as.Date('2021-01-01') ~ state_estimates$population[state_estima
                           date >= base::as.Date('2021-01-01') ~ state_estimates$population[state_estim
    ) %>%
  mutate(
    deaths_per_100k = 100000*(total_deaths / population),
    cases_per_100k = 100000*(total_cases / population),
    deaths_7_day = 100000*(((total_deaths - lag(total_deaths, 7)) / 7) / population),
    cases_7_day = 100000*(((total_cases - lag(total_cases, 7)) / 7) / population)
  )

state_estimates_2
```

```
## # A tibble: 1,022 x 8
##    date       total_deaths total_cases populat~1 death~2 cases~3 death~4 cases~5
##    <date>            <dbl>       <dbl>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 2020-03-15            2         136   5784308  0.0346    2.35 NA        NA
##  2 2020-03-16            2         161   5784308  0.0346    2.78 NA        NA
##  3 2020-03-17            3         183   5784308  0.0519    3.16 NA        NA
##  4 2020-03-18            3         216   5784308  0.0519    3.73 NA        NA
##  5 2020-03-19            5         278   5784308  0.0864    4.81 NA        NA
##  6 2020-03-20            5         364   5784308  0.0864    6.29 NA        NA
##  7 2020-03-21            6         475   5784308  0.104     8.21 NA        NA
##  8 2020-03-22            7         591   5784308  0.121    10.2   0.0123    1.12
##  9 2020-03-23           10         721   5784308  0.173    12.5   0.0198    1.38
## 10 2020-03-24           11         912   5784308  0.190    15.8   0.0198    1.80
## # ... with 1,012 more rows, and abbreviated variable names 1: population,
## #   2: deaths_per_100k, 3: cases_per_100k, 4: deaths_7_day, 5: cases_7_day
```

```
# Output

state_estimates_2 %>%
  na.omit() %>%
  pivot_longer(cols = ends_with("day"), names_to = "type", values_to = "number") %>%
  ggplot(aes(x = date, y = number)) + geom_line(aes(color = type)) + labs(title = str_c("Seven day avera
```



By changing the `stateName` variable we can repeat the analysis for any desired states.

Question 4 Using the same state, identify the top 5 counties in terms of deaths and cases per 100,000 people.

```
# Determining total population of counties
state_estimates <- us_population_estimates %>%
  filter(STNAME == stateName, Year == 2021) %>% dplyr::select(fips, Estimate)

# Total number of cases and deaths in each county
state_counties <- us_counties_total %>%
  filter(state == stateName, date == base::as.Date('2022-12-31')) %>%
  mutate(fips = as.numeric(fips))

# Merging the two datasets to calculate the per 100,000 number of cases and daths in each county
state_counties_per100k <- state_estimates %>%
  full_join(state_counties, by = c("fips")) %>%
  mutate(
    deaths_per_100k = 100000*(deaths / Estimate),
    cases_per_100k = 100000*(cases / Estimate)) %>%
  dplyr::select(county, date, fips, state, cases, deaths, cases_per_100k, deaths_per_100k)

state_counties_per100k %>% arrange(desc(deaths_per_100k))
```

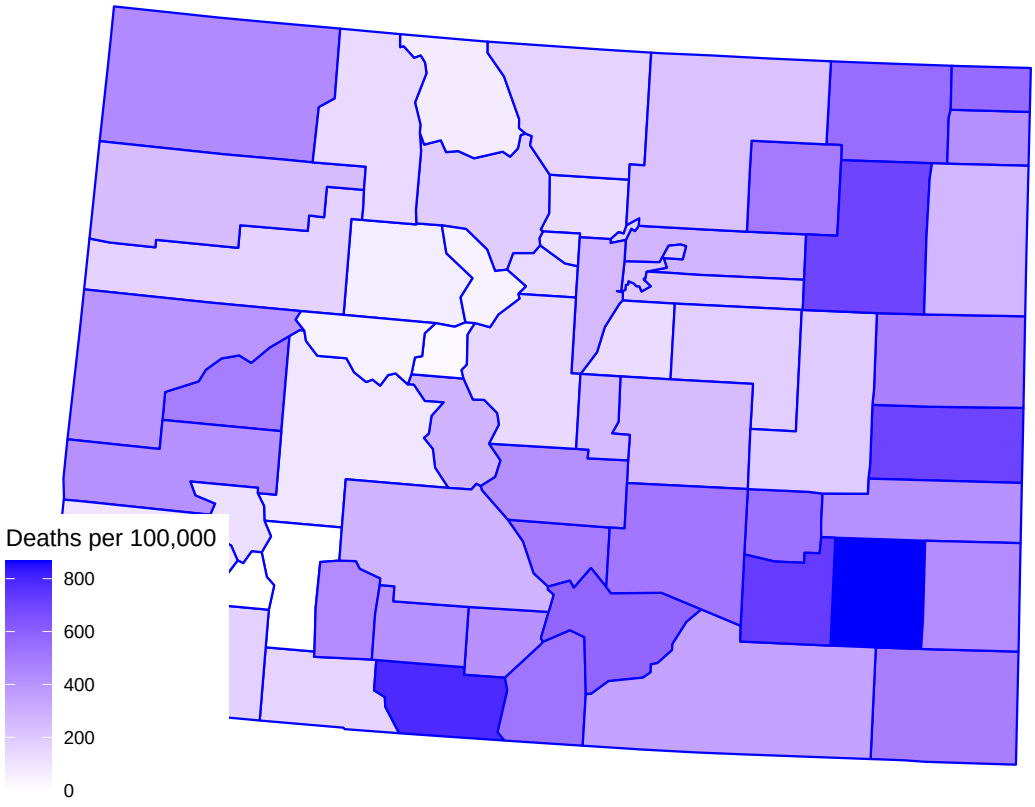
```
## # A tibble: 64 x 8
```

```
##    county     date        fips state    cases deaths cases_per_100k deaths_per~1
##    <chr>      <date>     <dbl> <chr>    <dbl>  <dbl>          <dbl>        <dbl>
##  1 Bent       2022-12-31  8011 Colorado  2924     50         50773.         868.
##  2 Conejos    2022-12-31  8021 Colorado  2267     60         29782.         788.
##  3 Otero      2022-12-31  8089 Colorado  5233    136         28143.         731.
##  4 Cheyenne   2022-12-31  8017 Colorado   383     12         22437.         703.
##  5 Washington 2022-12-31  8121 Colorado  1224     34         25180.         699.
##  6 Huerfano   2022-12-31  8055 Colorado  1743     40         25188.         578.
##  7 Sedgwick   2022-12-31  8115 Colorado   571     13         24443.         557.
##  8 Logan      2022-12-31  8075 Colorado  8548    119         39782.         554.
##  9 Crowley    2022-12-31  8025 Colorado  3521     32         58566.         532.
## 10 Costilla   2022-12-31  8023 Colorado   934     19         25766.         524.
## # ... with 54 more rows, and abbreviated variable name 1: deaths_per_100k
```

Top county in Colorado in terms of total cases per 100,000 population is "Crowley" Top county in Colorado in terms of total cases per 100,000 population is "Bent"

Question 5 Modify the code below for the map projection to plot county-level deaths and cases per 100,000 people for your state.

```
plot_usmap(regions = "counties", include="CO", data = state_counties_per100k, values = "deaths_per_100k
  scale_fill_continuous(low = "white", high = "blue", name = "Deaths per 100,000")
```



Question 6 Finally, select three other states and calculate the seven-day averages for new deaths and cases per 100,000 people for between March 15, 2020, and December 31, 2021.

```r
# Selecting the state of choice
stateName = "Michigan"

# Determining total population in year 2020 and 2021
state_estimates <- us_population_estimates %>%
  filter(STNAME == stateName) %>%
  group_by(Year) %>%
  summarise(population = sum(Estimate))

# Calculating the 7-day average of total death and cases in the state, per 100,000 individuals
MI_estimates <- us_counties_total %>%
  filter(state == stateName) %>%
  group_by(date) %>%
  summarise("total_deaths" = sum(deaths),
            "total_cases" = sum(cases)) %>%
  mutate(
    population = case_when(date < base::as.Date('2021-01-01') ~ state_estimates$population[state_estima
                           date >= base::as.Date('2021-01-01') ~ state_estimates$population[state_estim
    ) %>%
  mutate(
    deaths_per_100k = 100000*(total_deaths / population),
    cases_per_100k = 100000*(total_cases / population),
    deaths_7_day = 100000*(((total_deaths - lag(total_deaths, 7)) / 7) / population),
    cases_7_day = 100000*(((total_cases - lag(total_cases, 7)) / 7) / population)
  )


# Selecting the state of choice
stateName = "Montana"

# Determining total population in year 2020 and 2021
state_estimates <- us_population_estimates %>%
  filter(STNAME == stateName) %>%
  group_by(Year) %>%
  summarise(population = sum(Estimate))

# Calculating the 7-day average of total death and cases in the state, per 100,000 individuals
MO_estimates <- us_counties_total %>%
  filter(state == stateName) %>%
  group_by(date) %>%
  summarise("total_deaths" = sum(deaths),
            "total_cases" = sum(cases)) %>%
  mutate(
    population = case_when(date < base::as.Date('2021-01-01') ~ state_estimates$population[state_estima
                           date >= base::as.Date('2021-01-01') ~ state_estimates$population[state_estim
    ) %>%
  mutate(
    deaths_per_100k = 100000*(total_deaths / population),
    cases_per_100k = 100000*(total_cases / population),
    deaths_7_day = 100000*(((total_deaths - lag(total_deaths, 7)) / 7) / population),
    cases_7_day = 100000*(((total_cases - lag(total_cases, 7)) / 7) / population)
  )
```

```
# Selecting the state of choice
stateName = "Illinois"

# Determining total population in year 2020 and 2021
state_estimates <- us_population_estimates %>%
  filter(STNAME == stateName) %>%
  group_by(Year) %>%
  summarise(population = sum(Estimate))

# Calculating the 7-day average of total death and cases in the state, per 100,000 individuals
IL_estimates <- us_counties_total %>%
  filter(state == stateName) %>%
  group_by(date) %>%
  summarise("total_deaths" = sum(deaths),
            "total_cases" = sum(cases)) %>%
  mutate(
    population = case_when(date < base::as.Date('2021-01-01') ~ state_estimates$population[state_estima
                           date >= base::as.Date('2021-01-01') ~ state_estimates$population[state_estim
    ) %>%
  mutate(
    deaths_per_100k = 100000*(total_deaths / population),
    cases_per_100k = 100000*(total_cases / population),
    deaths_7_day = 100000*(((total_deaths - lag(total_deaths, 7)) / 7) / population),
    cases_7_day = 100000*(((total_cases - lag(total_cases, 7)) / 7) / population)
  )
```

We calculated the 7-day average of toatl deaths and cases in Illinois (IL), Michigan (MI), and Montana (MO)

```
IL_estimates
```

```
## # A tibble: 1,022 x 8
##    date       total_deaths total_cases populat~1 death~2 cases~3 death~4 cases~5
##    <date>            <dbl>       <dbl>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 2020-03-15            0          94  12785245 0         0.735 NA        NA
##  2 2020-03-16            0         104  12785245 0         0.813 NA        NA
##  3 2020-03-17            1         159  12785245 0.00782   1.24  NA        NA
##  4 2020-03-18            1         286  12785245 0.00782   2.24  NA        NA
##  5 2020-03-19            4         420  12785245 0.0313    3.29  NA        NA
##  6 2020-03-20            5         583  12785245 0.0391    4.56  NA        NA
##  7 2020-03-21            6         751  12785245 0.0469    5.87  NA        NA
##  8 2020-03-22            9        1047  12785245 0.0704    8.19   0.0101    1.06
##  9 2020-03-23           12        1285  12785245 0.0939   10.1    0.0134    1.32
## 10 2020-03-24           16        1535  12785245 0.125    12.0    0.0168    1.54
## # ... with 1,012 more rows, and abbreviated variable names 1: population,
## #   2: deaths_per_100k, 3: cases_per_100k, 4: deaths_7_day, 5: cases_7_day
```

```
MI_estimates
```

```
## # A tibble: 1,022 x 8
##    date       total_deaths total_cases populat~1 death~2 cases~3 death~4 cases~5
##    <date>            <dbl>       <dbl>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 2020-03-15            0          53  10067664 0         0.526 NA        NA
```

```
##  2 2020-03-16            0          54  10067664 0         0.536 NA        NA
##  3 2020-03-17            0          65  10067664 0         0.646 NA        NA
##  4 2020-03-18            1          80  10067664 0.00993   0.795 NA        NA
##  5 2020-03-19            3         334  10067664 0.0298    3.32  NA        NA
##  6 2020-03-20            4         548  10067664 0.0397    5.44  NA        NA
##  7 2020-03-21            6         787  10067664 0.0596    7.82  NA        NA
##  8 2020-03-22            9        1033  10067664 0.0894   10.3    0.0128    1.39
##  9 2020-03-23           16        1324  10067664 0.159    13.2    0.0227    1.80
## 10 2020-03-24           24        1791  10067664 0.238    17.8    0.0341    2.45
## # ... with 1,012 more rows, and abbreviated variable names 1: population,
## #   2: deaths_per_100k, 3: cases_per_100k, 4: deaths_7_day, 5: cases_7_day
```

```
MO_estimates
```

```
## # A tibble: 1,022 x 8
##    date       total_deaths total_cases populat~1 death~2 cases~3 death~4 cases~5
##    <date>            <dbl>       <dbl>     <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1 2020-03-15            0           6   1086193       0   0.552      NA  NA
##  2 2020-03-16            0           8   1086193       0   0.737      NA  NA
##  3 2020-03-17            0           8   1086193       0   0.737      NA  NA
##  4 2020-03-18            0          12   1086193       0   1.10       NA  NA
##  5 2020-03-19            0          19   1086193       0   1.75       NA  NA
##  6 2020-03-20            0          19   1086193       0   1.75       NA  NA
##  7 2020-03-21            0          29   1086193       0   2.67       NA  NA
##  8 2020-03-22            0          34   1086193       0   3.13        0   0.368
##  9 2020-03-23            0          45   1086193       0   4.14        0   0.487
## 10 2020-03-24            0          51   1086193       0   4.70        0   0.566
## # ... with 1,012 more rows, and abbreviated variable names 1: population,
## #   2: deaths_per_100k, 3: cases_per_100k, 4: deaths_7_day, 5: cases_7_day
```

Question 7 Create a visualization comparing the seven-day averages for new deaths and cases per 100,000 people for the four states you selected.

```
CO <- state_estimates_2 %>% dplyr::select(date, deaths_7_day, cases_7_day) %>% rename(CO_deaths = deaths
IL <- IL_estimates %>% dplyr::select(date, deaths_7_day, cases_7_day) %>% rename(IL_deaths = deaths_7_da
MI <- MI_estimates %>% dplyr::select(date, deaths_7_day, cases_7_day) %>% rename(MI_deaths = deaths_7_da
MO <- MO_estimates %>% dplyr::select(date, deaths_7_day, cases_7_day) %>% rename(MO_deaths = deaths_7_da

# Merging data for 4 states
four_states_avg <- reduce(list(CO, IL, MI, MO), inner_join, by = "date") %>% na.omit()

# Tidying up the data frame by pivot longer
four_states_avg <- four_states_avg %>%
  pivot_longer(-date,
               names_to = c("state", ".value"),
               names_sep="_" ) %>%
  pivot_longer(c("deaths", "cases"), names_to = "Var", values_to = "Count") %>%
  mutate(Count = log10(Count))
```

```
## Warning in mask$eval_all_mutate(quo): NaNs produced
```

```
# Plotting
ggplot(four_states_avg) + geom_line(aes(x = date, y = Count, group = interaction(state, Var), color = s
```

We can see that for all of the states we see a more or less similar trend of rise/drops in number of cases and deaths.

END of part2