COVID 19 Analysis

Anthony Tetreault

2025-04-08

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 calculate statistics per 100,000 people.

```
# Import New York Times COVID-19 data
# Import Population Estimates from US Census Bureau
# Use curl to retrieve each file and save in data folder for project
# foreach($year in 2020, 2021, 2022) { curl -L "https://raw.githubusercontent.com/nytimes/covid-19-data/
us_counties_2020 <- read_csv("data/us-counties-2020.csv")
```

```
## 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("data/us-counties-2021.csv")
```

```
## 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("data/us-counties-2022.csv")
```

```
## 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("data/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.
```

```
# This data does not contain estimates for 2022
us_pop_est_22 <- read_csv("data/co-est2024-alldata.csv",
                          col_type = list(
                              STATE = col_double(),
                              COUNTY = col_double()
                          ))
# File retrieved from "https://www2.census.gov/programs-surveys/popest/datasets/2020-2024/counties/tota
# Contains population estimates from 2020-2024.
# Extract 2022 population estimates
us_pop_est_22 <- us_pop_est_22 %>%
    select(STNAME, CTYNAME, STATE, COUNTY, POPESTIMATE2022) %>% # Grab only necessary columns
    pivot_longer(POPESTIMATE2022, names_to = "Year", names_prefix = "POPESTIMATE",
                 values_to = "Estimate") %>%    # Pivot the year variable out of the column title and e
    mutate(Year = as.double(Year)) %>%          # Cast Year as double for join
    filter(STNAME != CTYNAME) %>%               # Filter out state estimates
    # Add fips to us_pop_est_22 before joining to us_population_estimates
    left_join(us_population_estimates %>%
                filter(Year == max(Year)) %>% # Filter by one year to prevent duplicates
                select(fips, STNAME, CTYNAME, STATE, COUNTY),
                by = join_by(STNAME, CTYNAME, STATE, COUNTY))
# and join with us_population_estimates
us_population_estimates <- us_population_estimates %>%
    union(us_pop_est_22) %>% # Add us_pop_est_22
    arrange(STNAME, CTYNAME) # Arrange variables for easier viewing
```

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 and tidy the 2020, 2021, and 2022 COVID data sets.

us_counties_202122 <- bind_rows(us_counties_2020, us_counties_2021, us_counties_2022) %>% # Combine all
    filter(state != "Puerto Rico" & cases > 0 & deaths > 0)  # Remove stats from Puerto Rico and days w

max_date <- max(us_counties_202122$date) # Find max date
us_total_cases <- us_counties_202122 %>% # Group by date and sum cases to that date from first case
    group_by(date) %>%
    summarize(total_cases = sum(cases, na.rm = TRUE))
us_total_deaths <- us_counties_202122 %>% # Group by date and sum deaths to that date from first case
    group_by(date) %>%
    summarize(total_deaths = sum(deaths, na.rm = TRUE))
us_totals <- full_join(us_total_deaths, us_total_cases, by = join_by("date" == "date"))
us_totals
```

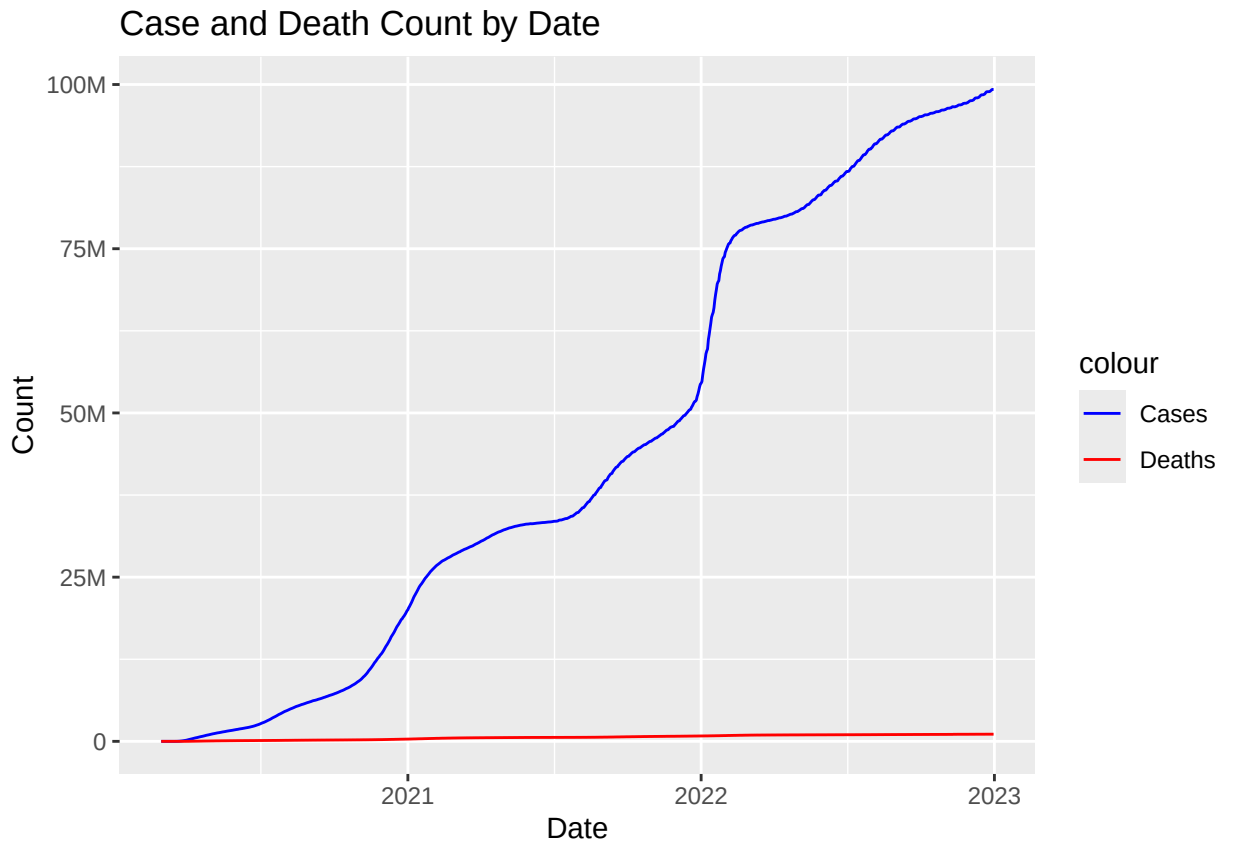
```
## # A tibble: 1,037 x 3
##    date       total_deaths total_cases
##    <date>            <dbl>       <dbl>
##  1 2020-02-29            1           4
##  2 2020-03-01            3          11
##  3 2020-03-02            6          15
##  4 2020-03-03           10          22
##  5 2020-03-04           12          35
##  6 2020-03-05           12          54
##  7 2020-03-06           15          66
##  8 2020-03-07           19          80
##  9 2020-03-08           22         102
## 10 2020-03-09           26         171
## # i 1,027 more rows
```

- To start, we had to combine all three of the NYT datasets together using bind_rows(). As their schema was identical, we had no need to adjust anything before combining. After combining, we cleaned the data to remove all US territory data (Puerto Rico) and return only those dates with at least one case and one death that day. While this adds a small amount of error to our numbers, at the scale of the data, the effects are negligible. As of December 31, 2022, there were 99256606 total cases with a total death count of 1091628

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?

```r
ggplot(us_totals, aes(x = date)) +
    geom_line(aes(y = total_cases, color = "Cases")) +      # Cases layer
    geom_line(aes(y = total_deaths, color = "Deaths")) +    # Deaths layer
    scale_color_manual(values = c("blue", "red"), labels = c("Cases", "Deaths")) + # Adjust scale by
    labs(title = "Case and Death Count by Date", x = "Date", y = "Count") + # Add labels
    scale_x_date() + # Scale x-axis by date
    scale_y_continuous(labels = label_number(scale_cut = cut_short_scale())) # Re-scale y-axis
```



- In order to get these two scaled lines on the same graph, we had to do a few things to coerce the scale and visibility of the levels. We used both color and scaling, ensuring lines are different colors and changing the y-axis from a scientific notation to something more readable. This graph could definitely be misinterpreted, due to the scaling needed to capture the full case count, especially when we know that deaths were under reported or misrepresented as other conditions. Where the full case count after ~ three years is ~ 100M (99256606), the death count was just over 1 M (1091628) on 2022-12-31. This makes it look much less deadly than it actually was, relative to other infectious diseases.

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.

```r
# Create a new table, based on the table from Question 1, and calculate the number of new deaths and ca

us_totals_daily_weekly <- us_totals %>%
    mutate(
        delta_deaths_1 = (total_deaths - lag(total_deaths, n = 1)),
        delta_cases_1 = (total_cases - lag(total_cases, n = 1)),
        delta_deaths_7 = round(slide_dbl(delta_deaths_1, ~mean(.x), .before = 6), 1),
        delta_cases_7 = round(slide_dbl(delta_cases_1, ~mean(.x), .before = 6), 1)
    )
us_totals_daily_weekly
```

```
## # A tibble: 1,037 x 7
##    date       total_deaths total_cases delta_deaths_1 delta_cases_1
##    <date>            <dbl>       <dbl>          <dbl>         <dbl>
##  1 2020-02-29            1           4             NA            NA
##  2 2020-03-01            3          11              2             7
##  3 2020-03-02            6          15              3             4
##  4 2020-03-03           10          22              4             7
##  5 2020-03-04           12          35              2            13
##  6 2020-03-05           12          54              0            19
##  7 2020-03-06           15          66              3            12
##  8 2020-03-07           19          80              4            14
##  9 2020-03-08           22         102              3            22
## 10 2020-03-09           26         171              4            69
## # i 1,027 more rows
## # i 2 more variables: delta_deaths_7 <dbl>, delta_cases_7 <dbl>
```

```r
max_new_cases_date <- us_totals_daily_weekly %>%
    filter(delta_cases_1 == max(delta_cases_1, na.rm = TRUE)) %>%
    select(date, delta_cases_1)
max_new_cases_date
```

```
## # A tibble: 1 x 2
##   date       delta_cases_1
##   <date>             <dbl>
## 1 2022-01-10       1425956
```

```r
max_new_deaths_date <- us_totals_daily_weekly %>%
    filter(delta_deaths_1 == max(delta_deaths_1, na.rm = TRUE)) %>%
    select(date, delta_deaths_1)
max_new_deaths_date
```

```
## # A tibble: 1 x 2
##   date       delta_deaths_1
##   <date>              <dbl>
## 1 2022-11-11          13179
```

- We begin by using lag to subtract the day before's totals from the current day's total, leaving us with the daily totals for all counties. Then we must calculate our seven-day rolling averages using slide_dbl() from the slider package. This slide function will act as such: slide_dble(take_this_column, ~calculate_this_stat(.dfplaceholder), .before = how_many_cells_before_to_perform_calc_on). Take

delta_deaths_7, for example. We are going to "slide" (or shift the "window of aggregation," if we will,) the mean of delta_deaths_1 across the 7 days before (6 days before + day of.) We can then filter by max(cases) and max(deaths) to find the date with the highest new cases/deaths. January 2022 had the highest instances of new cases (there are multiple days where new cases exceed 1 million.) November 11th of 2022 had the highest single deaths per day at 13,179.

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.

```
pop_by_yr <- us_population_estimates %>%
    group_by(Year) %>%
    summarize(pop_estimate = sum(Estimate))

# Add population estimate (estimate) to us_totals_daily_weekly
us_totals_daily_weekly <- us_totals_daily_weekly %>%
    mutate(estimate = case_when(
       grepl("2020", date) ~ pop_by_yr$pop_estimate[1],
       grepl("2021", date) ~ pop_by_yr$pop_estimate[2],
       grepl("2022", date) ~ pop_by_yr$pop_estimate[3]))

us_totals_per_100k <- us_totals_daily_weekly %>%
    mutate(
        total_deaths = ((total_deaths/estimate)*100000),
        total_cases = ((total_cases/estimate)*100000),
        delta_deaths_1 = ((delta_deaths_1/estimate)*100000),
        delta_cases_1 = ((delta_cases_1/estimate)*100000),
        delta_deaths_7 = ((delta_deaths_7/estimate)*100000),
        delta_cases_7 = ((delta_cases_7/estimate)*100000)
    ) %>%
    select(-estimate)
us_totals_per_100k
```

```
## # A tibble: 1,037 x 7
##    date       total_deaths total_cases delta_deaths_1 delta_cases_1
##    <date>            <dbl>       <dbl>          <dbl>         <dbl>
##  1 2020-02-29     0.000302     0.00121      NA             NA
##  2 2020-03-01     0.000905     0.00332       0.000603       0.00211
##  3 2020-03-02     0.00181      0.00452       0.000905       0.00121
##  4 2020-03-03     0.00302      0.00664       0.00121        0.00211
##  5 2020-03-04     0.00362      0.0106        0.000603       0.00392
##  6 2020-03-05     0.00362      0.0163        0              0.00573
##  7 2020-03-06     0.00452      0.0199        0.000905       0.00362
##  8 2020-03-07     0.00573      0.0241        0.00121        0.00422
##  9 2020-03-08     0.00664      0.0308        0.000905       0.00664
## 10 2020-03-09     0.00784      0.0516        0.00121        0.0208
## # i 1,027 more rows
## # i 2 more variables: delta_deaths_7 <dbl>, delta_cases_7 <dbl>
```

- To get a population estimate for each year, we took our combined us_population_estimates data (that now includes 2022 data, as well,) group it by year, then sum all estimates for all counties. This gives us a

tibble with Year and pop_estimate which we can then use to add a column to us_totals_daily_weekly from Q3 so we can figure out proportional cases. We then create us_totals_per_100k by mutating all columns in the us_totals_daily_weekly to divide by the population estimate for that year and multiply by 100000 to get deaths and cases totals, single_day_totals, and seven_day_rolling_avgs. This can provide some context and show virality based on a standardized population estimate. This means when we compare this data to other countries (in the third portion,) we will have a better picture to compare values with because we have already adjusted for the variance in population across countries and areas.

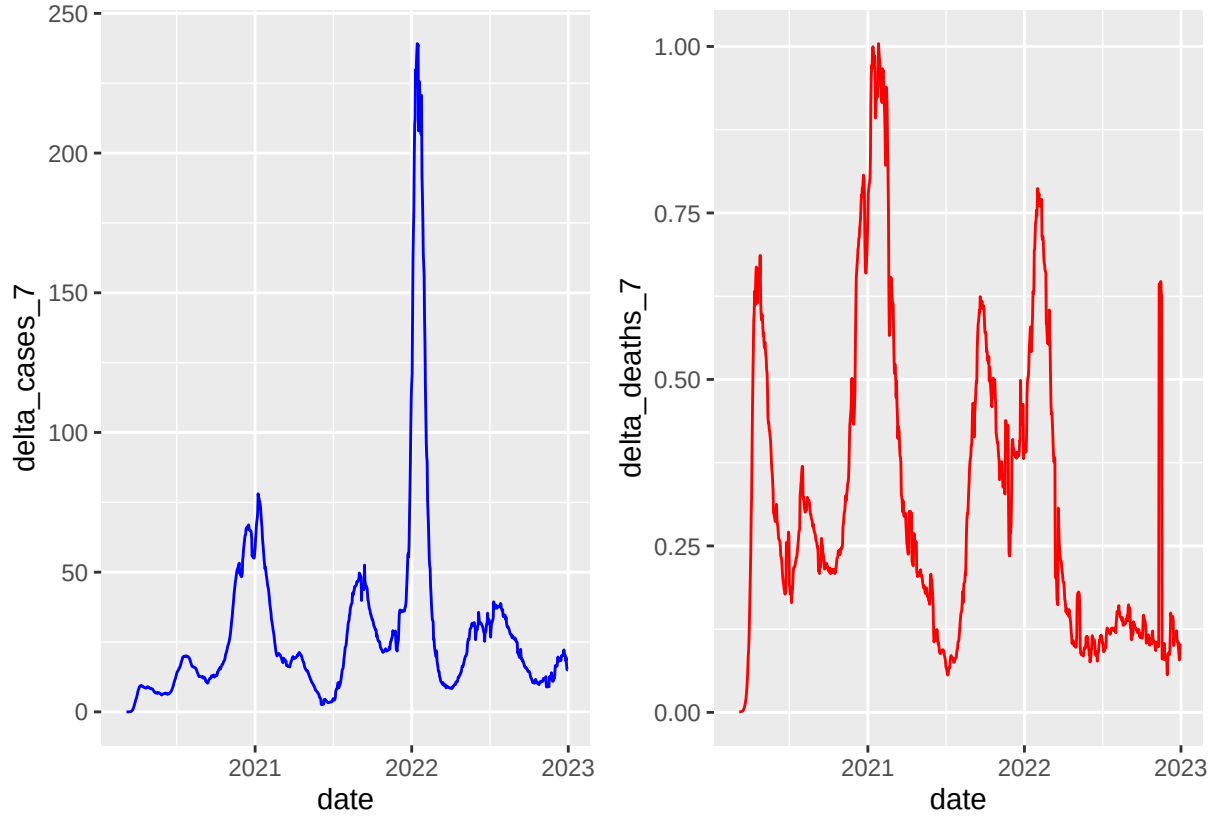
Question 5

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

```
# p1q5-response
# Cases
us_cases_per_100k <- us_totals_per_100k %>%
    ggplot(aes(x=date, y=delta_cases_7)) +
    geom_line(color = "blue")

# Deaths
us_deaths_per_100k <- us_totals_per_100k %>%
    ggplot(aes(x=date, y=delta_deaths_7)) +
    geom_line(color = "red")

us_cases_per_100k + us_deaths_per_100k
```



- We used ggplot to plot out cases_per_100k totals and deaths_per_100k totals on line graphs and then we combined them to create a two column grid. This is helpful to really get a good image of the two separate trends. Plotting them on the same graph, as seen in the previous graphical example, led to some complications in actually seeing the trend of deaths, as the number of deaths were about 1% of the number of cases, so the scale did not allow for actual interpretation of information. By graphing them individually and combining with the patchwork pacakage, we are able to get a better image of both of these trends. The deaths trend follows pretty closely along with the cases shape, with the exception of a spike in deaths during the middle chunk of November 2022, with a multi-day peak from 0.09 to ~ 0.7 delta_deaths_7. These graphs also show that Winters were the major time for spreading events, with Summer having a smaller, but noticeable bumps, as well. For Winters, it makes sense that people were spending more time indoors and, thus around each other in confined spaces, precautions taken or not. Summer is a similar situation, increased consociality, but with a less pronounced effect as many activities in the Summer can be done in areas more conducive to precautions working (specifically outdoor activities.)

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.

```
# p2q1-response
# Determine the top 10 states in terms of total deaths and cases between March 15, 2020, and December 3

total_cases_byst <- us_counties_202122 %>%
    group_by(date, state) %>%
    summarize(total_cases = sum(cases, na.rm = TRUE))

total_deaths_byst <- us_counties_202122 %>%
    group_by(date, state) %>%
    summarize(total_deaths = sum(deaths, na.rm = TRUE))

us_totals_byst <- full_join(
        total_cases_byst, total_deaths_byst,
        by = join_by(state, date)
    ) %>%
    select(state, date, total_deaths, total_cases)
max_date <- as.Date("2021-12-31")
top_10_states <- us_totals_byst %>%
    group_by(state) %>%
    filter(date == max_date) %>%
    arrange(desc(total_deaths), desc(total_cases)) %>%
    head(10)
state_pops <- us_population_estimates %>%
    group_by(STNAME, Year) %>%
    summarize(popest = sum(Estimate))
top_10_pop_21 <- state_pops %>%
    filter(Year == 2021) %>%
    arrange(desc(popest)) %>%
```

```
    head(10) %>% view()
top_10_pop_21
```

```
## # A tibble: 10 x 3
## # Groups:   STNAME [10]
##    STNAME          Year   popest
##    <chr>          <dbl>    <dbl>
##  1 California      2021 39237836
##  2 Texas           2021 29527941
##  3 Florida         2021 21781128
##  4 New York        2021 19835913
##  5 Pennsylvania    2021 12964056
##  6 Illinois        2021 12671469
##  7 Ohio            2021 11780017
##  8 Georgia         2021 10799566
##  9 North Carolina  2021 10551162
## 10 Michigan        2021 10050811
```

```
top_10_states
```

```
## # A tibble: 10 x 4
## # Groups:   state [10]
##    state        date       total_deaths total_cases
##    <chr>        <date>            <dbl>       <dbl>
##  1 California   2021-12-31        76709     5515271
##  2 Texas        2021-12-31        76062     4574832
##  3 Florida      2021-12-31        62504     4166392
##  4 New York     2021-12-31        58993     3473970
##  5 Pennsylvania 2021-12-31        36705     2036424
##  6 Georgia      2021-12-31        30283     1798497
##  7 Illinois     2021-12-31        29850     2154058
##  8 Ohio         2021-12-31        29440     2016095
##  9 New Jersey   2021-12-31        29037     1561259
## 10 Michigan     2021-12-31        28984     1706355
```

- We begin by calculating total cases by state (total_cases_byst) and total deaths by state (total_deaths_byst), first grouping by both date and state and then summing cases and deaths by state over each date. Then we join those two together on state and date and using select to reorder our columns for an easier to view end output. Max date is calculated and then used in our top_10_states calculation. To get that we first group by state and then filter our totals by the max_date (2021-12-31), arranging our totals in descending order to get highest first, and then take only ten rows from the head to get our tops. Most of our top states make sense as they correlate with 9 of the states that contain the top 10 largest populations (North Carolina being the missing value, and their deaths/cases were 14th, so close to.) The two states that made movement relative to their populations levels were Georgia and New Jersey. A possible explanation for this is warmer weather in Georgia compared to other places, coupled with a more lax public health climate in the South, in general, could have caused their increases. When you compare to North Carolina, which has the highest concentration of Graduate and Post-graduate degrees in the nation in "The Research Trianble," including a large portion of MD's at the public and private universities within that region of Raleigh-Durham-Chapel Hill, which could explain their lower levels compared to their population and other states around them. For New Jersey, I would argue that it's most likely due to its proximity to New York City and the more urbanized nature of the population distribution of the state (I would argue Pennsylvania probably has a similar explanation,) although, a county-level analysis would need to be done to confirm that.

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?

```
# Determine the top 10 states in terms of deaths and cases per 100,000 people between March 15, 2020, a
max_date1 <- as.Date("2021-12-31") # For 2021 answer
max_date2 <- as.Date("2022-12-31") # For latest date in dataset

st_totals <- us_totals_byst %>%
    mutate(Year = year(date)) %>% # Mutate a year column from the date to match state_pops
    full_join(state_pops, join_by(state == STNAME, Year == Year)) %>% # Join on state and Year to get p
    select(-Year) # Drop the Year column

st_totals_per_100k <- st_totals %>%
    mutate(
        total_deaths = ((total_deaths/popest)*100000),
        total_cases = ((total_cases/popest)*100000),
    ) %>%
    rename(deaths_per_100k = total_deaths, cases_per_100k = total_cases) %>%
    select(-popest)

st_totals_2021 <- st_totals_per_100k %>% # This is what the question was asking for, but I got pop est
    filter(date == max_date1) %>%
    arrange(desc(deaths_per_100k), desc(cases_per_100k)) %>%
    head(10)
st_totals_2021
```

```
## # A tibble: 10 x 4
## # Groups:   date [1]
##    state         date       deaths_per_100k cases_per_100k
##    <chr>         <date>               <dbl>          <dbl>
##  1 Mississippi   2021-12-31            354.         18432.
##  2 Arizona       2021-12-31            333.         18986.
##  3 Alabama       2021-12-31            326.         17790.
##  4 Louisiana     2021-12-31            324.         17908.
##  5 New Jersey    2021-12-31            313.         16847.
##  6 Arkansas      2021-12-31            302.         18488.
##  7 West Virginia 2021-12-31            299.         18405.
##  8 New York      2021-12-31            297.         17514.
##  9 Tennessee     2021-12-31            296.         19783.
## 10 Massachusetts 2021-12-31            290.         16263.
```

```
st_totals_2022 <- st_totals_per_100k %>% # This is the answer for the lastest date in the dataset provi
    filter(date == max_date2) %>%
    arrange(desc(deaths_per_100k), desc(cases_per_100k)) %>%
    head(10)
st_totals_2022
```

```
## # A tibble: 10 x 4
## # Groups:   date [1]
```

```
##    state         date       deaths_per_100k cases_per_100k
##    <chr>         <date>               <dbl>          <dbl>
##  1 Mississippi   2022-12-31            446.          32467.
##  2 West Virginia 2022-12-31            439.          35213.
##  3 Arizona       2022-12-31            436.          32237.
##  4 Alabama       2022-12-31            418.          30908.
##  5 New Mexico    2022-12-31            418.          31156.
##  6 Arkansas      2022-12-31            417.          31263.
##  7 Michigan      2022-12-31            406.          29644.
##  8 Tennessee     2022-12-31            400.          33530.
##  9 Louisiana     2022-12-31            400.          32935.
## 10 Oklahoma      2022-12-31            399.          31044.
```

- First, we establish two separate max_date variables to align with the question's explicit intent (max_date1 gives us up to December 31st, 2021) and with the implied intention of the question (max_date2 gives us the latest date in the data, December 31st, 2022.) I added 2022 county population estimates to the us_population_estimates variable in order to have population estimates for all dates in the data. With that, we will mutate our us_totals_byst to add a Year column based on the year of the date (using lubridate,) and join our state populations by state name and the Year, dropping the Year after the join with a select. This gives us total deaths and total cases and a population estimate for states by date. We then mutate the total_deaths and total_cases to divide by the state population estimate for that year and multiply by 100,000 to get estimates per 100,000 people in the state. Total_deaths and total_cases are renamed to deaths_per_100k and cases_per_100k and the population estimate column is dropped. I then filtered the state totals per 100k by a max_date1 and max_date2 to get total deaths per 100k and total cases per 100k for each state. Arrange those both descending cases/deaths per 100k and there we have it.

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.

```
# Select a state and then filter by state and date range your data from Question 1. Calculate the seven

OR_totals_comb <- st_totals %>% # Calculate Oregon totals for all variables
    group_by(state) %>% # Group by state
    mutate( # Perform calculations
        delta_deaths_1 = (total_deaths - lag(total_deaths)),
        delta_cases_1 = (total_cases - lag(total_cases)),
        delta_deaths_7 = round(slide_dbl(delta_deaths_1, ~mean(.x), .before = 6), 1),
        delta_cases_7 = round(slide_dbl(delta_cases_1, ~mean(.x), .before = 6), 1)
    ) %>%
    filter(state == "Oregon") # Pull out only Oregon
OR_totals_per_100k <- OR_totals_comb %>% # Calculate Oregon totals per 100k using popest
    mutate( # Perform calculations
        deaths_per_100k = (delta_deaths_1/popest)*100000,
        cases_per_100k = (delta_cases_1/popest)*100000,
        deaths_7_day = (delta_deaths_7/popest)*100000,
        cases_7_day = (delta_cases_7/popest)*100000
    ) %>%
    select(state, date, deaths_per_100k, cases_per_100k, deaths_7_day, cases_7_day)
OR_cases_per_100k <- OR_totals_per_100k %>% # Plot Seven-day avg Cases per 100k
    ggplot(aes(x=date, y=cases_7_day)) +
    geom_line(color = "blue") +
    ylim(NA, 250) +
```

```
    labs( # Adjust labels
        title = "Seven-Day Average Cases Per 100k",
        x = "Date",
        y = "Seven-Day Average"
    )

OR_deaths_per_100k <- OR_totals_per_100k %>%
    ggplot(aes(x=date, y=deaths_7_day)) +
    geom_line(color = "red") +
    ylim(NA, 1.0) +
    scale_y_continuous(breaks = breaks_extended(4))
```

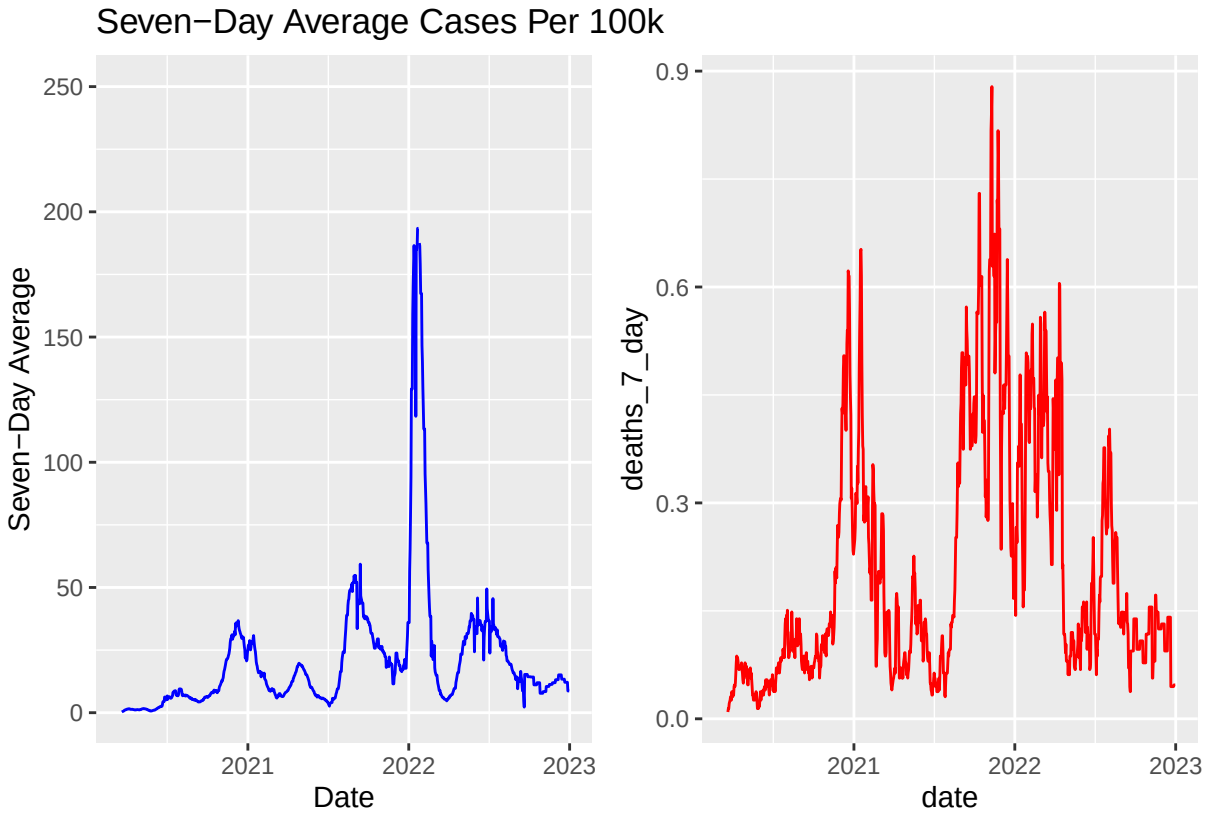
```
## Scale for y is already present.
## Adding another scale for y, which will replace the existing scale.
```

```
    labs( # Adjust labels
        title = "Seven-Day Average Deaths Per 100k",
        x = "Date",
        y = "Seven-Day Average"
    )
```

```
## $x
## [1] "Date"
##
## $y
## [1] "Seven-Day Average"
##
## $title
## [1] "Seven-Day Average Deaths Per 100k"
##
## attr(,"class")
## [1] "labels"
```

```
OR_cases_per_100k + OR_deaths_per_100k
```

```
## Warning: Removed 7 rows containing missing values or values outside the scale range
## (`geom_line()`).
## Removed 7 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



- To begin, we group the state totals data, group it all by state, then make a column for and calculate a daily and weekly rolling average for both deaths and cases with a mutate. From this, we filter by the stae we are looking for, Oregon. These totals are then adjusted by population estimate to summarize deaths, cases, and our rolling averages per 100k people. The final select statement is used to organize the data for viewing. Now that we have our data sorted out, the totals are graphed against the date for both cases and deaths per 100k. For the both of the graphs, the y-axis needed some re-tooling to add more breaks compared to the default production. We then added labels to both of them with labs. These graphs were combined using the patchwork library's graphical concatenation (+).
- As for our graphs themselves, they track pretty well with each other, although, cases have a more pronounced rise during the summer months compared to the deaths data, although that would need to be analyzed a little more, as that might be accounted for by the sheer magnitude of the variance in range.

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

```
# Using the same state as Question 2, filter your state and date range from the combined data set from

max_date <- as.Date("2022-12-31")

OR_counties_202122 <- us_counties_202122 %>%
    filter(state == "Oregon") %>% # Grab only Oregon counties
    select(-state)

OR_cty_pop_est <- us_population_estimates %>% # Drop state code and county code columns
```

```
    filter(STNAME == "Oregon") %>%  # Grab only Oregon counties
    select(-STNAME, -STATE, -COUNTY) %>%
    mutate(CTYNAME = str_extract(CTYNAME, "(\\w+)"))

OR_cty_deaths_total <- OR_counties_202122 %>%
    group_by(date, county, fips) %>%
    summarize(total_deaths = sum(deaths, na.rm = TRUE)) %>%
    mutate(Year = year(date),
           fips = as.double(fips)
    )

OR_cty_cases_total <- OR_counties_202122 %>%
    group_by(date, county, fips) %>%
    summarize(total_cases = sum(cases, na.rm = TRUE)) %>%
    mutate(Year = year(date),
           fips = as.double(fips)
    )

OR_county_totals <- OR_cty_deaths_total %>%
    full_join(OR_cty_cases_total, join_by(date == date, county == county, fips == fips, Year == Year)) %
    full_join(OR_cty_pop_est, join_by(Year == Year, county == CTYNAME, fips == fips)) %>%
    select(-Year)

OR_county_per_100k <- OR_county_totals %>%
    group_by(county) %>%
    mutate( # Perform calculations
        total_deaths = (total_deaths/Estimate)*100000,
        total_cases = (total_cases/Estimate)*100000
    ) %>%
    select(-Estimate)

county_top_5_deaths <- OR_county_per_100k %>%
    filter(date == max_date) %>%
    arrange(desc(total_deaths)) %>%
    head(5)
county_top_5_deaths
```

```
## # A tibble: 5 x 5
## # Groups:   county [5]
##   date       county     fips total_deaths total_cases
##   <date>     <chr>     <dbl>        <dbl>       <dbl>
## 1 2022-12-31 Harney    41025         544.      25753.
## 2 2022-12-31 Josephine 41033         445.      23326.
## 3 2022-12-31 Jefferson 41031         407.      35171.
## 4 2022-12-31 Douglas   41019         403.      22747.
## 5 2022-12-31 Malheur   41045         389.      29701.
```

```
county_top_5_cases <- OR_county_per_100k %>%
    filter(date == max_date) %>%
    arrange(desc(total_cases)) %>%
    head(5)
county_top_5_cases
```

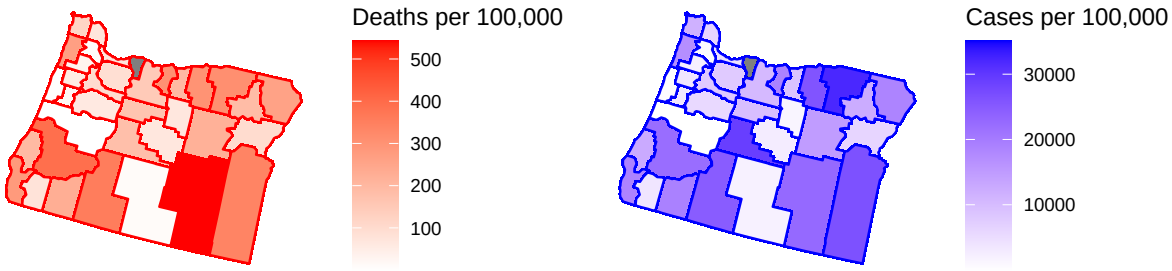
```
## # A tibble: 5 x 5
## # Groups:   county [5]
##   date       county     fips total_deaths total_cases
##   <date>     <chr>     <dbl>        <dbl>       <dbl>
## 1 2022-12-31 Jefferson 41031         407.      35171.
## 2 2022-12-31 Umatilla  41059         311.      32478.
## 3 2022-12-31 Grant     41023         360.      30605.
## 4 2022-12-31 Malheur   41045         389.      29701.
## 5 2022-12-31 Crook     41013         360.      29349.
```

- With our us_counties_202122 data, we filter by the state of Oregon and drop the state column (as they will all be duplicates.) We also need to filter the us_population_estimates data and select columns to match schema of OR_counties_202122, including matching county names (CTYNAME) to county. We can then perform calculations of both case and death totals for all counties and then join those two tibbles and the county population estimates for Oregon. From there we can calculate totals per 100k by grouping by counties, divinging by county population estimates, and multiplying by 100,000. With this data we can filter by the maximum date (December 31st, 2022,) arrange by descending death and case totals, and trim the table to be the top 5 output.
- What I find to be extremely interesting is that the two largest counties, with the most population, c("Multnomah", "Washington"), c(41051, 41067), c(2022, 2022), c(795960, 600229), were not only not in the top 5 counties, but were 30th (Multnomah) and 34th (Washington) for deaths per 100k and 22nd (Multnomah) and 26th (Washington) for cases per 100k. Jefferson county, with the most cases per 100k and third most deaths per 100k, is in the lower 30% of population size. Harney County, with the most deaths per 100k, is the 32nd lowest population (in lowest 15%.)

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

```
OR_county_deaths_plot <- plot_usmap(
    regions = "county", include="OR",
    data = OR_county_per_100k,
    values = "total_deaths", color = "red"
    ) +
    scale_fill_continuous(
        low = "white", high = "red",
        name = "Deaths per 100,000"
    ) +
    theme(legend.position = "right")

OR_county_cases_plot <- plot_usmap(
    regions = "county", include="OR",
    data = OR_county_per_100k,
    values = "total_cases", color = "blue"
    ) +
    scale_fill_continuous(
        low = "white", high = "blue",
        name = "Cases per 100,000"
    ) +
    theme(legend.position = "right")
OR_county_deaths_plot + OR_county_cases_plot
```



- These graphs show that population centers were actually hit way less hard than more rural areas. There are some possible explanations for that, especially in the state of Oregon, where our rural/urban divide is extremely stark. Our rural areas are mostly conservative politically and focus a lot on "freedoms" or positive rights of action, whereas our urban areas are some of the most liberal politically in the country. Multnomah County, where Portland is, had some of the most restrictive lock downs in the country and maintained mask mandates and gathering limitations for significantly longer than a lot of other places.

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.

```
max_date <- as.Date("2022-12-31")

FL_totals_comb <- st_totals %>% # Calculate Oregon totals for all variables
    group_by(state) %>% # Group by state
    mutate( # Perform calculations
        delta_deaths_1 = (total_deaths - lag(total_deaths)),
        delta_cases_1 = (total_cases - lag(total_cases)),
        delta_deaths_7 = round(slide_dbl(delta_deaths_1, ~mean(.x), .before = 6), 1),
        delta_cases_7 = round(slide_dbl(delta_cases_1, ~mean(.x), .before = 6), 1)
    ) %>%
    filter(state == "Florida") # Pull out only Oregon
FL_totals_per_100k <- FL_totals_comb %>% # Calculate Oregon totals per 100k using popest
    mutate( # Perform calculations
        deaths_per_100k = (delta_deaths_1/popest)*100000,
```

```
        cases_per_100k = (delta_cases_1/popest)*100000,
        deaths_7_day = (delta_deaths_7/popest)*100000,
        cases_7_day = (delta_cases_7/popest)*100000
    ) %>%
    select(state, date, deaths_per_100k, cases_per_100k, deaths_7_day, cases_7_day)
FL_totals_per_100k
```

```
## # A tibble: 1,031 x 6
## # Groups:   state [1]
##    state   date       deaths_per_100k cases_per_100k deaths_7_day cases_7_day
##    <chr>   <date>               <dbl>          <dbl>        <dbl>       <dbl>
##  1 Florida 2020-03-06        NA             NA          NA           NA
##  2 Florida 2020-03-07         0              0.00464    NA           NA
##  3 Florida 2020-03-08         0              0          NA           NA
##  4 Florida 2020-03-09         0              0          NA           NA
##  5 Florida 2020-03-10         0              0          NA           NA
##  6 Florida 2020-03-11         0              0          NA           NA
##  7 Florida 2020-03-12         0              0.00927    NA           NA
##  8 Florida 2020-03-13         0              0           0            0.00185
##  9 Florida 2020-03-14         0.00464        0           0.000464     0.00139
## 10 Florida 2020-03-15         0              0.00464     0.000464     0.00185
## # i 1,021 more rows
```

```
CT_totals_comb <- st_totals %>% # Calculate Connecticut totals for all variables
    group_by(state) %>% # Group by state
    mutate( # Perform calculations
        delta_deaths_1 = (total_deaths - lag(total_deaths)),
        delta_cases_1 = (total_cases - lag(total_cases)),
        delta_deaths_7 = round(slide_dbl(delta_deaths_1, ~mean(.x), .before = 6), 1),
        delta_cases_7 = round(slide_dbl(delta_cases_1, ~mean(.x), .before = 6), 1)
    ) %>%
    filter(state == "Connecticut") # Pull out only Connecticut
CT_totals_per_100k <- CT_totals_comb %>% # Calculate Connecticut totals per 100k using popest
    mutate( # Perform calculations
        deaths_per_100k = (delta_deaths_1/popest)*100000,
        cases_per_100k = (delta_cases_1/popest)*100000,
        deaths_7_day = (delta_deaths_7/popest)*100000,
        cases_7_day = (delta_cases_7/popest)*100000
    ) %>%
    select(state, date, deaths_per_100k, cases_per_100k, deaths_7_day, cases_7_day)
CT_totals_per_100k
```

```
## # A tibble: 1,019 x 6
## # Groups:   state [1]
##    state      date       deaths_per_100k cases_per_100k deaths_7_day cases_7_day
##    <chr>      <date>               <dbl>          <dbl>        <dbl>       <dbl>
##  1 Connectic~ 2020-03-18         NA             NA          NA           NA
##  2 Connectic~ 2020-03-19          0.0833         0.917      NA           NA
##  3 Connectic~ 2020-03-20          0              0.556      NA           NA
##  4 Connectic~ 2020-03-21          0.0278         0.639      NA           NA
##  5 Connectic~ 2020-03-22          0.0833         3.64       NA           NA
##  6 Connectic~ 2020-03-23          0.0556         1.97       NA           NA
```

```
##  7 Connectic~ 2020-03-24          0.0556          4.00       NA            NA
##  8 Connectic~ 2020-03-25          0.194           9.03        0.0722        2.96
##  9 Connectic~ 2020-03-26          0.0556          3.78        0.0667        3.37
## 10 Connectic~ 2020-03-27          0.167           7.47        0.0917        4.36
## # i 1,009 more rows
```

```
KY_totals_comb <- st_totals %>% # Calculate Kentucky totals for all variables
    group_by(state) %>% # Group by state
    mutate( # Perform calculations
        delta_deaths_1 = (total_deaths - lag(total_deaths)),
        delta_cases_1 = (total_cases - lag(total_cases)),
        delta_deaths_7 = round(slide_dbl(delta_deaths_1, ~mean(.x), .before = 6), 1),
        delta_cases_7 = round(slide_dbl(delta_cases_1, ~mean(.x), .before = 6), 1)
    ) %>%
    filter(state == "Kentucky") # Pull out only Kentucky
KY_totals_per_100k <- KY_totals_comb %>% # Calculate Kentucky totals per 100k using popest
    mutate( # Perform calculations
        deaths_per_100k = (delta_deaths_1/popest)*100000,
        cases_per_100k = (delta_cases_1/popest)*100000,
        deaths_7_day = (delta_deaths_7/popest)*100000,
        cases_7_day = (delta_cases_7/popest)*100000
    ) %>%
    select(state, date, deaths_per_100k, cases_per_100k, deaths_7_day, cases_7_day)
KY_totals_per_100k
```

```
## # A tibble: 1,021 x 6
## # Groups:   state [1]
##    state    date       deaths_per_100k cases_per_100k deaths_7_day cases_7_day
##    <chr>    <date>               <dbl>          <dbl>        <dbl>       <dbl>
##  1 Kentucky 2020-03-16          NA             NA           NA           NA
##  2 Kentucky 2020-03-17           0              0           NA           NA
##  3 Kentucky 2020-03-18           0              0.0222      NA           NA
##  4 Kentucky 2020-03-19           0.0222         0.311       NA           NA
##  5 Kentucky 2020-03-20           0              0.200       NA           NA
##  6 Kentucky 2020-03-21           0.0222         0.0666      NA           NA
##  7 Kentucky 2020-03-22           0              0.0222      NA           NA
##  8 Kentucky 2020-03-23           0              0.133        0.00666      0.109
##  9 Kentucky 2020-03-24           0              0.200        0.00666      0.135
## 10 Kentucky 2020-03-25           0.0444         0.0666       0.0133       0.142
## # i 1,011 more rows
```

- To accomplish this, I simply used the code written for the Oregon totals per 100k and adjusted it to find the totals for the states I chose and their respective populatin estimates. I chose Florida, Connecticut, and Kentucky to analyze.

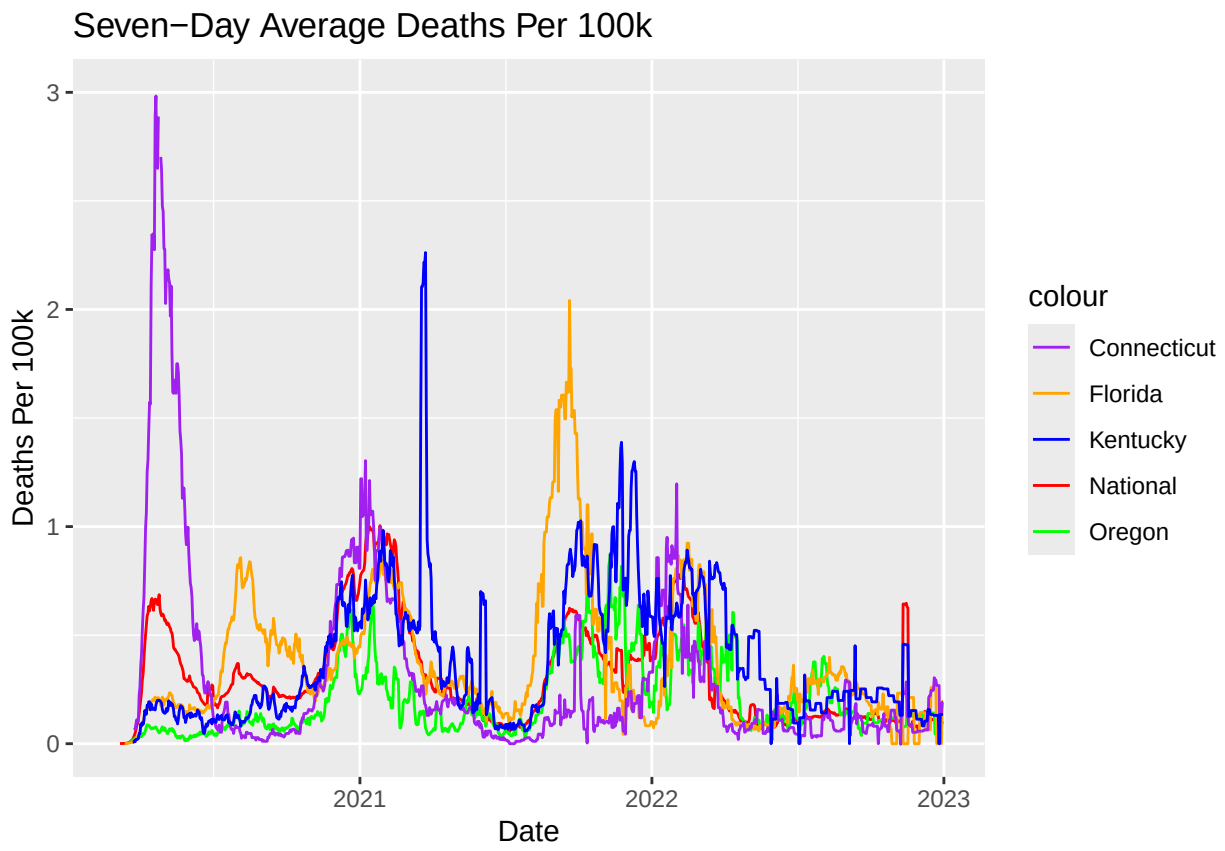
Question 7 Create a visualization comparing the seven-day averages for new deaths and cases per 100,000 people for the four states you selected.

```
states_deaths_per_100k <- ggplot() +
    geom_line(data = us_totals_per_100k, mapping = aes(x=date, y = delta_deaths_7, color = "National"),
    geom_line(data = OR_totals_per_100k, mapping = aes(x=date, y = deaths_7_day, color = "Oregon"), na.
    geom_line(data = FL_totals_per_100k, mapping = aes(x=date, y = deaths_7_day, color = "Florida"), na
```

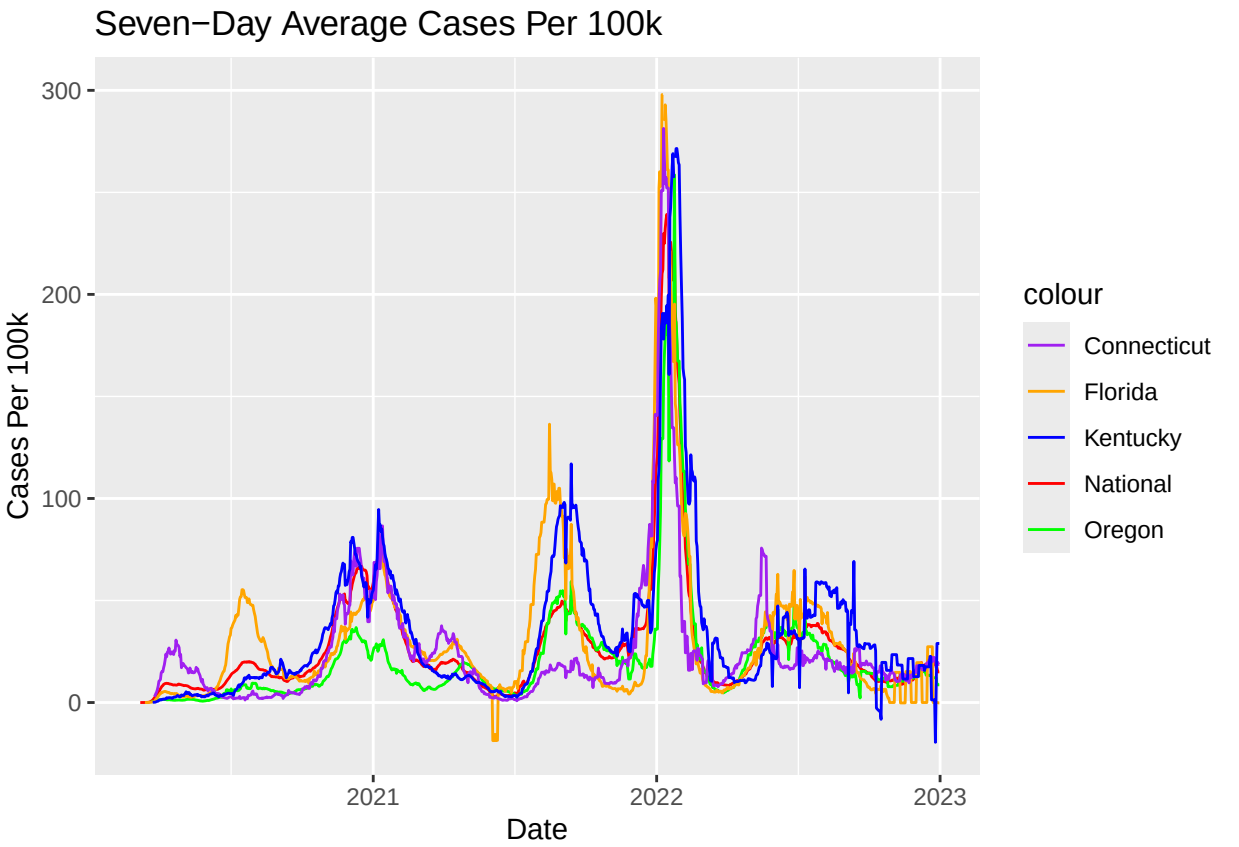
```
    geom_line(data = CT_totals_per_100k, mapping = aes(x=date, y = deaths_7_day, color = "Connecticut")
    geom_line(data = KY_totals_per_100k, mapping = aes(x=date, y = deaths_7_day, color = "Kentucky"), na
    ylim(NA, 3.0) +
    labs(
        title = "Seven-Day Average Deaths Per 100k",
        x = "Date",
        y = "Deaths Per 100k"
    )  +
    scale_color_manual(values = c(
        "National" = "red",
        "Oregon" = "green",
        "Florida" = "orange",
        "Connecticut" = "purple",
        "Kentucky" = "blue"
    ))

states_cases_per_100k <- ggplot() +
    geom_line(data = us_totals_per_100k, mapping = aes(x=date, y = delta_cases_7, color = "National"), r
    geom_line(data = OR_totals_per_100k, mapping = aes(x=date, y = cases_7_day, color = "Oregon"), na.rm
    geom_line(data = FL_totals_per_100k, mapping = aes(x=date, y = cases_7_day, color = "Florida"), na.r
    geom_line(data = CT_totals_per_100k, mapping = aes(x=date, y = cases_7_day, color = "Connecticut"),
    geom_line(data = KY_totals_per_100k, mapping = aes(x=date, y = cases_7_day, color = "Kentucky"), na
    ylim(NA, 300) +
    labs(
        title = "Seven-Day Average Cases Per 100k",
        x = "Date",
        y = "Cases Per 100k"
    ) +
    scale_color_manual(values = c(
        "National" = "red",
        "Oregon" = "green",
        "Florida" = "orange",
        "Connecticut" = "purple",
        "Kentucky" = "blue"
    ))

states_deaths_per_100k
```




```
states_cases_per_100k
```



- I crafted a general outline of ggplot taking in totals_per_100k for each state and plotting date against our cases and deaths, respectively. All of the x-axis' played well (being dates and the exact same across all of them,) but the y-axis gave me some issues. As their values varied greatly, I had to spend some time wrangling the y-axis specifications for each of the states. For the deaths per 100k, I set them all to the same as our highest state Connecticut. For cases, we set our highest value at 300 to accommodate all of the states evaluated. Then I combined them with the patchwork concatenation (+) operator.
- Even adjusted for population, per 100k, we see that Florida had a higher case load than our other states, although Kentucky was a close second. All of our states follow a similar pattern of increasing case and death counts peaking during winter, but also increasing during the Summer. As stated before, I believe that this can explain the increased cases in Florida, as their weather permits more gatherings throughout the year, and with their perpetual Summer could have provided some false sense of safety that being outside, or being forced to be inside together in the A/C, increased the case load. That, combined with a more conservative political climate on a State-level, could have contributed to them being the highest level of case load. Connecticut saw a swift spike in deaths towards the beginning of the pandemic, most likely due to its close proximity and intertwined economic system with its neighbor New York (specifically New York City) – almost half the state is a suburb of the megalopolis. All states also saw an increase specifically in the winter of 2022, into March-April, an increase in case load and an associated spike in deaths, although these death spikes were not the highest in any of the states, even if the case spike was the highest.

```
# Import global COVID-19 statistics aggregated by the Center for Systems Science and Engineering (CSSE)
# Import global population estimates from the World Bank.
```

```
#csse_global_deaths <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_
#csse_global_cases <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_c
#csse_us_deaths <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covi
#csse_us_cases <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_

#globabl_population_estimates <- read_csv("global_population_estimates.csv")
```

Part 3 - Global Comparison

Question 1 Using the state you selected in Part 2 Question 2 compare the daily number of cases and deaths reported from the CSSE and NY Times.

```
# To compare your state data between the two data sets, you will first need to tidy the US CSSE death a
# Hint: Review the documentation for pivot_longer().


# Once you have tidied your data, join the two CSSE US data sets to include cases and deaths in one tab


# Finally, create two visualizations with one plotting the CSSE and NY Times cases and the other plotti
```

```
# Your tidied CSSE data for your selected state should look similar to the following tibble:
#
# A tibble: 43,362 × 6
#     fips  county  state        date       cases  deaths
#     <dbl> <chr>   <chr>       <date>     <dbl>  <dbl>
#  1  8001  Adams   Colorado    2020-03-15     6      0
#  2  8001  Adams   Colorado    2020-03-16     8      0
#  3  8001  Adams   Colorado    2020-03-17    10      0
#  4  8001  Adams   Colorado    2020-03-18    10      0
#  5  8001  Adams   Colorado    2020-03-19    10      0
#  6  8001  Adams   Colorado    2020-03-20    12      0
#  7  8001  Adams   Colorado    2020-03-21    14      0
#  8  8001  Adams   Colorado    2020-03-22    18      0
#  9  8001  Adams   Colorado    2020-03-23    25      0
# 10  8001  Adams   Colorado    2020-03-24    27      0
# ... with 43,352 more rows
```

– Communicate your methodology, results, and interpretation here –

Question 2 Now that you have verified the data reported from the CSSE and NY Times are similar, combine the global and US CSSE data sets and identify the top 10 countries in terms of deaths and cases per 100,000 people between March 15, 2020, and December 31, 2021.

```
# First, combine and tidy the CSSE death and cases data sets. You may wish to keep the two sets separat
# Then, tidy the global population estimates. While tidying your data, remember to include columns that
# You will notice that the population estimates data does not include every country reported in the CSS
```

– Communicate your methodology, results, and interpretation here –

Question 3 Construct a visualization plotting the 10 countries in terms of deaths and cases per 100,000 people between March 15, 2020, and December 31, 2021. In designing your visualization keep the number of data you will be plotting in mind. You may wish to create two separate visualizations, one for deaths and another for cases.

– Communicate your methodology, results, and interpretation here –

Question 4 Finally, select four countries from one continent and create visualizations for the daily number of confirmed cases per 100,000 and the daily number of deaths per 100,000 people between March 15, 2020, and December 31, 2021.

– Communicate your methodology, results, and interpretation here –