COVID 19 Analysis

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-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(
  "https://raw.githubusercontent.com/nytimes/covid-19-data/master/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(
  "https://raw.githubusercontent.com/nytimes/covid-19-data/master/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("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 and tidy the 2020, 2021, and 2022 COVID data sets.
# Hint: Review the rbind() documentation to combine the three data sets.
us_counties_combined <- rbind(us_counties_2020, us_counties_2021, us_counties_2022)

# Drop rows pertaining to Puerto Rico
us_total <- us_counties_combined %>%
  filter(state != "Puerto Rico") %>%
  # Keep rows dated 2020-03-15 and later
  filter(date >= ymd("2020-03-15")) %>%
  # Calculate cumulative total deaths and cases on each day
  group_by(date) %>%
  summarize(total_deaths = sum(deaths), total_cases = sum(cases))

head(us_total, n = 10)
```

```
## # A tibble: 10 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
```

```
# Sort by date to get most recent
max_date <- arrange(us_total, desc(date)) %>%
  select(date) %>%
  slice_head(n = 1) %>%
  deframe() %>%
  ymd()

max_date
```

```
## [1] "2022-12-31"
```

```
# Cumulative total cases as of max_date
us_total_cases <- arrange(us_total, desc(date)) %>%
  select(total_cases) %>%
  slice_head(n = 1) %>%
  deframe()

us_total_cases
```

```
## [1] 99374764
```

```
# Cumulative total cases as of max_date
us_total_deaths <- arrange(us_total, desc(date)) %>%
  select(total_deaths) %>%
  slice_head(n = 1) %>%
  deframe()

us_total_deaths
```

```
## [1] 1094296
```

```
# Your output should look similar to the following tibble:
#
#   A tibble: 657 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
```

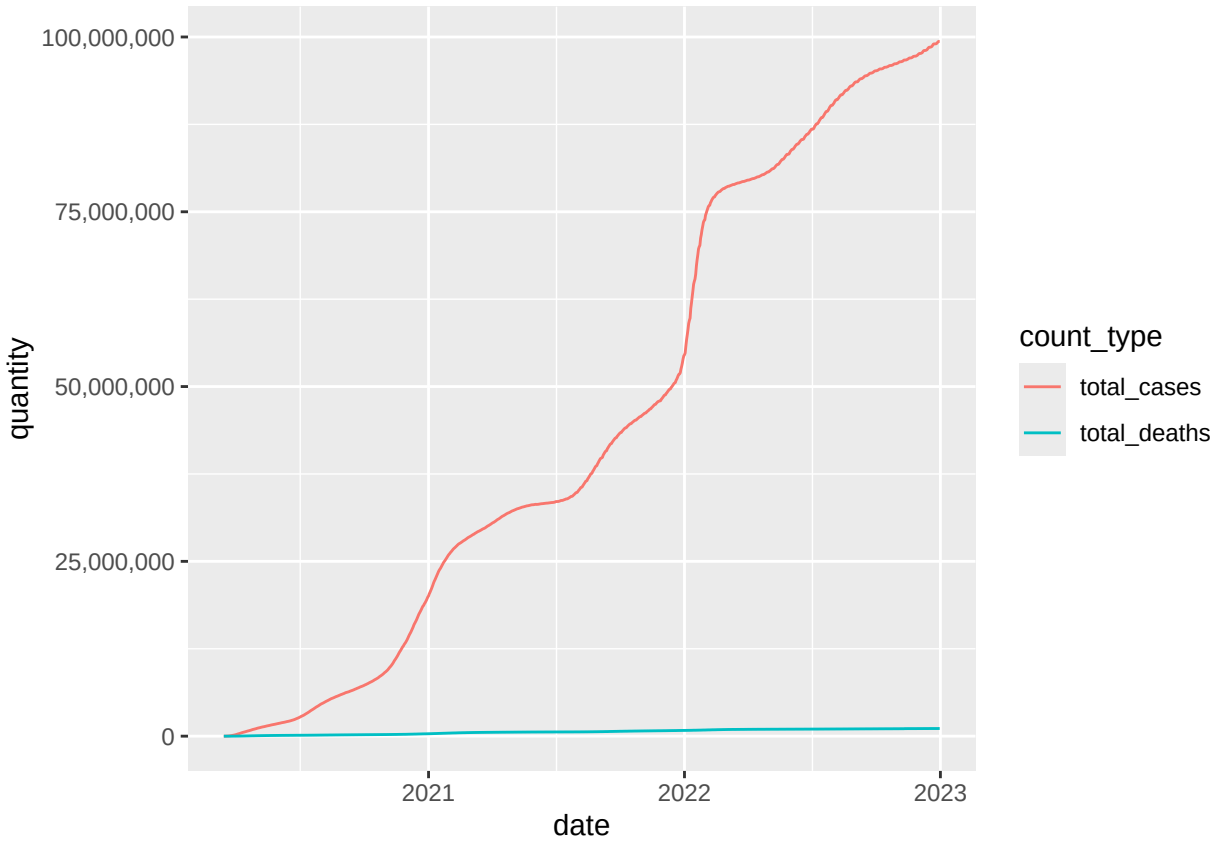
```
#   6 2020-03-20          277         17998
#   7 2020-03-21          359         24507
#   8 2020-03-22          457         33050
#   9 2020-03-23          577         43474
#  10 2020-03-24          783         53899
# ... with 647 more rows
#
```

– Communicate your methodology, results, and interpretation here –

As of December 31, 2022, the most recent date available in the New York Times Covid-19 data, there was a cumulative total of 9.9374764×10^7 Covid-19 cases and 1.094296×10^6 Covid-19 deaths across all US counties excluding the territory of Puerto Rico.

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?

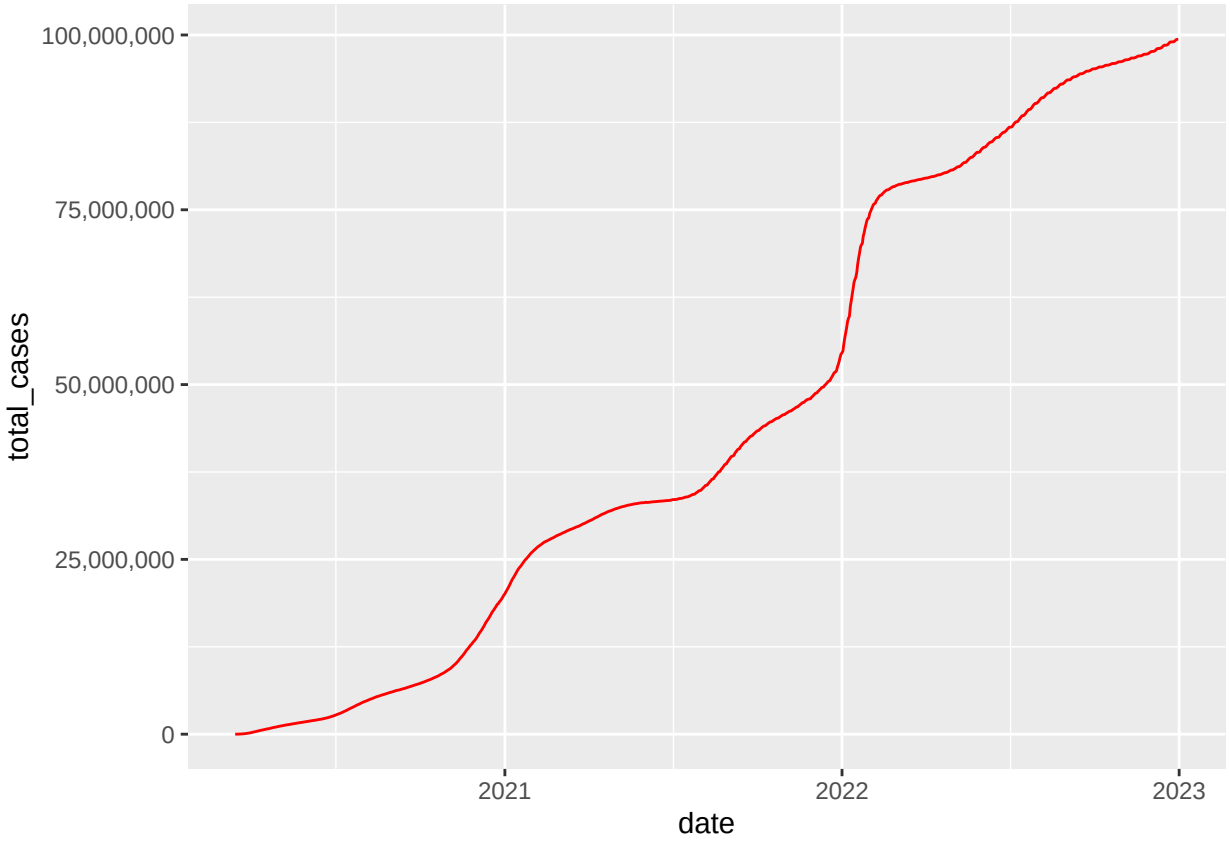
```
# Create a visualization for the total number of US cases and deaths since March 15, 2020.

# Pivot us_total to combine total_deaths and total_cases into a single column,
# then create a new column to indicate if the amount pertains to deaths or cases,
# so as to plot two line graphs, one for deaths and one for cases,
# on the same plot

us_total %>% pivot_longer(cols = c(total_deaths, total_cases),
                          names_to = "count_type",
                          values_to = "quantity"
                          ) %>%
  ggplot(mapping = aes(x = date, y = quantity, color = count_type)) +
  scale_y_continuous(labels = scales::label_comma()) +
  geom_line()
```



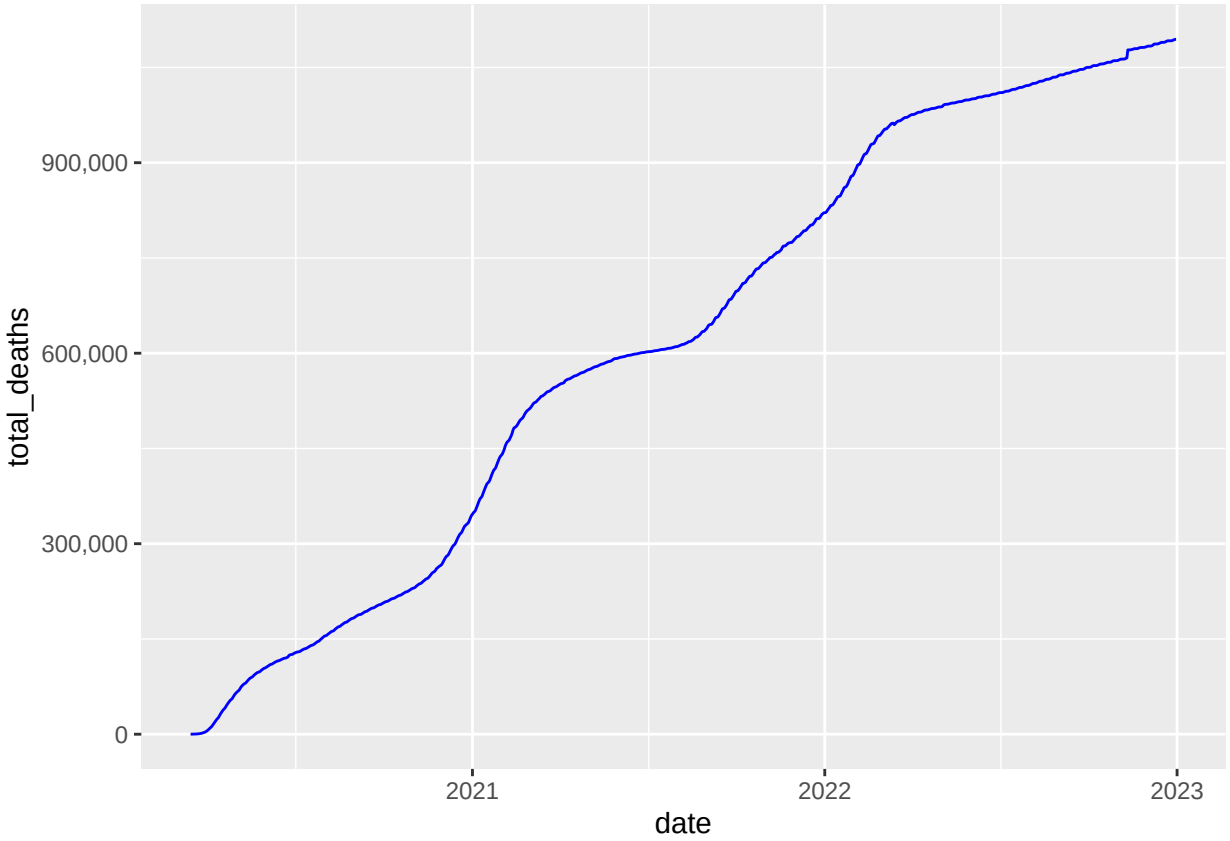
A line plot is most suitable for showing how the numerical variables `total_deaths` and `total_cases` change over time. To generate the above line plot, the `us_total` table was manipulated further by merging all counts into a single column and then adding a new column to indicate whether the count pertains to deaths or cases. The result is a red line plot showing the change in `total_cases` over time and a blue line plot showing the change in `total_deaths` over time. The line plots show that the cumulative total number of Covid-19 cases in the US have been increasing steadily since March 15, 2020.

At first glance, the cumulative total number of Covid-19 deaths over time appears constant at a level near to zero. However, this is likely because `total_deaths` is generally of a much smaller scale than `total_cases`. To examine the change in `total_deaths` over time, it may be better to plot the two variables on separate line plots:

```
# Line plot of total_cases over time
ggplot(us_total, mapping = aes(x = date, y = total_cases)) +
  geom_line(color = "red") +
  scale_y_continuous(labels = scales::label_comma())
```




```
# Line plot of total_deaths over time
ggplot(us_total, mapping = aes(x = date, y = total_deaths)) +
  geom_line(color = "blue") +
  scale_y_continuous(labels = scales::label_comma())
```



In the above plot, it appears more clearly that the cumulative total number of Covid-19 deaths in the US had actually increased steadily from March 15, 2020 to December 31, 2022.

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.

```
# Create a new table, based on the table from Question 1, and
# calculate the number of new deaths and cases each day and
# a seven day average of new deaths and cases.
#
# Hint: Look at the documentation for lag() when computing the number of
# new deaths and cases and the seven-day averages.
#
# Compute new variables for daily new deaths and cases by subtracting
# 1-day lagged values from total_deaths and total_cases
us_covid_delta <- us_total %>% mutate(
  "delta_deaths_1" = total_deaths - lag(total_deaths, n = 1),
  "delta_cases_1" = total_cases - lag(total_cases, n = 1),
  # Compute new variables for seven-day averages of new deaths and cases
  "delta_deaths_7" = frollmean(delta_deaths_1, n = 7),
  "delta_cases_7" = frollmean(delta_cases_1, n = 7)
  )
```

```r
head(us_covid_delta, n = 10)
```

```
## # A tibble: 10 x 7
##    date       total_deaths total_cases delta_deaths_1 delta_cases_1
##    <date>            <dbl>       <dbl>          <dbl>         <dbl>
##  1 2020-03-15           68        3595             NA            NA
##  2 2020-03-16           91        4502             23           907
##  3 2020-03-17          117        5901             26          1399
##  4 2020-03-18          162        8345             45          2444
##  5 2020-03-19          212       12387             50          4042
##  6 2020-03-20          277       17998             65          5611
##  7 2020-03-21          359       24507             82          6509
##  8 2020-03-22          457       33050             98          8543
##  9 2020-03-23          577       43474            120         10424
## 10 2020-03-24          783       53899            206         10425
## # i 2 more variables: delta_deaths_7 <dbl>, delta_cases_7 <dbl>
```

```r
# Date with largest number of new cases
max_new_cases_date <- us_covid_delta %>%
  arrange(desc(delta_cases_1)) %>%
  slice_head(n = 1) %>%
  select(date) %>%
  deframe() %>%
  ymd()

max_new_cases_date
```

```
## [1] "2022-01-10"
```

```r
# Date with largest number of new deaths
max_new_deaths_date <- us_covid_delta %>%
  arrange(desc(delta_deaths_1)) %>%
  slice_head(n = 1) %>%
  select(date) %>%
  deframe() %>%
  ymd()

max_new_deaths_date
```

```
## [1] "2022-11-11"
```

```r
#  Your output should look similar to the following tibble:
#
#  date
#  total_deaths     >  the cumulative number of deaths up to and including the associated date
#  total_cases      >  the cumulative number of cases up to and including the associated date
#  delta_deaths_1   >  the number of new deaths since the previous day
#  delta_cases_1    >  the number of new cases since the previous day
#  delta_deaths_7   >  the average number of deaths in a seven-day period
#  delta_cases_7    >  the average number of cases in a seven-day period
#==
```

```
#  A tibble: 813 x 7
#    date           total_deaths   total_cases   delta_deaths_1    delta_cases_1  delta_deaths_7  delta
#    <date>            <dbl>          <dbl>         <dbl>             <dbl>        <dbl>           <
#  1 2020-03-15           68           3600            0                 0          NA
#  2 2020-03-16           91           4507           23               907          NA
#  3 2020-03-17          117           5906           26              1399          NA
#  4 2020-03-18          162           8350           45              2444          NA
#  5 2020-03-19          212          12393           50              4043          NA
#  6 2020-03-20          277          18012           65              5619          NA
#  7 2020-03-21          360          24528           83              6516          NA
#  8 2020-03-22          458          33073           98              8545        55.7              4
#  9 2020-03-23          579          43505          121             10432        69.7              5
# 10 2020-03-24          785          53938          206             10433        95.4              6
# ... with 803 more rows
```

Computing the number of new deaths and cases for each date shows that January 10, 2022 had the highest number of new daily cases and November 11, 2022 had the highest number of new daily deaths across all US counties, excluding Puerto Rico, from March 15, 2020 to December 31, 2022.

```
# 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.

# Hint: To calculate per 100,000 people, first tidy the population estimates
# data and calculate the US population in 2020 and 2021. Then, you will need to
# divide each statistic by the estimated population and then multiply by 100,000.
#
# Hint: look at the help documentation for grepl() and case_when() to divide
# the averages by the US population for each year.
# For example, take the simple tibble, t_new:
#
#     x     y
#   <int> <chr>
#     1     a
#     2     b
#     3     a
#     4     b
#     5     a
#     6     b
#
#
# To add a column, z, that is dependent on the value in y, you could:
#
# t_new %>%
#   mutate(z = case_when(grepl("a", y) ~ "not b",
#                        grepl("b", y) ~ "not a"))
#

# Tidy the US population estimates data to find total population per year
us_population_by_year <- us_population_estimates %>%
  group_by(Year) %>%
```

```
  summarize("total_population" = sum(Estimate))

# US population data is only available for 2020 and 2021
us_population_estimates %>% count(Year)
```

Question 4

```
## # A tibble: 2 x 2
##    Year     n
##   <dbl> <int>
## 1  2020  3143
## 2  2021  3143
```

```
# Get 2020 and 2021 populations
us_population_2020 <- us_population_by_year %>%
  filter(Year == 2020) %>%
  select(total_population) %>%
  deframe()
us_population_2021 <- us_population_by_year %>%
  filter(Year == 2021) %>%
  select(total_population) %>%
  deframe()

# Create new table showing new deaths and cases per 100,000 people each day
# and a seven-day average of new deaths and cases per 100,000 people
# Keep existing variables but modify values to a per 100,000 basis

us_covid_delta_per_100k <- us_covid_delta %>%
  mutate(
    # If the year is 2020, get the result of each variable divided by
    # us_population_2020 then multiplied by 100,000
    total_deaths = case_when(
      grepl("2020", date) ~ (total_deaths / us_population_2020 * 100000),
      # Else if the year is 2021, get the result of each variable divided by
      # us_population_2021 then multiplied by 100,000
      grepl("2021", date) ~ (total_deaths / us_population_2021 * 100000)
      ),
    total_cases = case_when(
      grepl("2020", date) ~ (total_cases / us_population_2020 * 100000),
      grepl("2021", date) ~ (total_cases / us_population_2021 * 100000)
      ),
    delta_deaths_1 = case_when(
      grepl("2020", date) ~ (delta_deaths_1 / us_population_2020 * 100000),
      grepl("2021", date) ~ (delta_deaths_1 / us_population_2021 * 100000)
      ),
    delta_cases_1 = case_when(
      grepl("2020", date) ~ (delta_cases_1 / us_population_2020 * 100000),
      grepl("2021", date) ~ (delta_cases_1 / us_population_2021 * 100000)
      ),
    delta_deaths_7 = case_when(
      grepl("2020", date) ~ (delta_deaths_7 / us_population_2020 * 100000),
      grepl("2021", date) ~ (delta_deaths_7 / us_population_2021 * 100000)
      ),
```

```
    delta_cases_7 = case_when(
      grepl("2020", date) ~ (delta_cases_7 / us_population_2020 * 100000),
      grepl("2021", date) ~ (delta_cases_7 / us_population_2021 * 100000)
      )
    )

head(us_covid_delta_per_100k, n = 10)
```

```
## # A tibble: 10 x 7
##    date       total_deaths total_cases delta_deaths_1 delta_cases_1
##    <date>            <dbl>       <dbl>          <dbl>         <dbl>
##  1 2020-03-15       0.0205        1.08       NA              NA
##  2 2020-03-16       0.0275        1.36        0.00694         0.274
##  3 2020-03-17       0.0353        1.78        0.00784         0.422
##  4 2020-03-18       0.0489        2.52        0.0136          0.737
##  5 2020-03-19       0.0640        3.74        0.0151          1.22
##  6 2020-03-20       0.0836        5.43        0.0196          1.69
##  7 2020-03-21       0.108         7.39        0.0247          1.96
##  8 2020-03-22       0.138         9.97        0.0296          2.58
##  9 2020-03-23       0.174        13.1         0.0362          3.14
## 10 2020-03-24       0.236        16.3         0.0621          3.14
## # i 2 more variables: delta_deaths_7 <dbl>, delta_cases_7 <dbl>
```

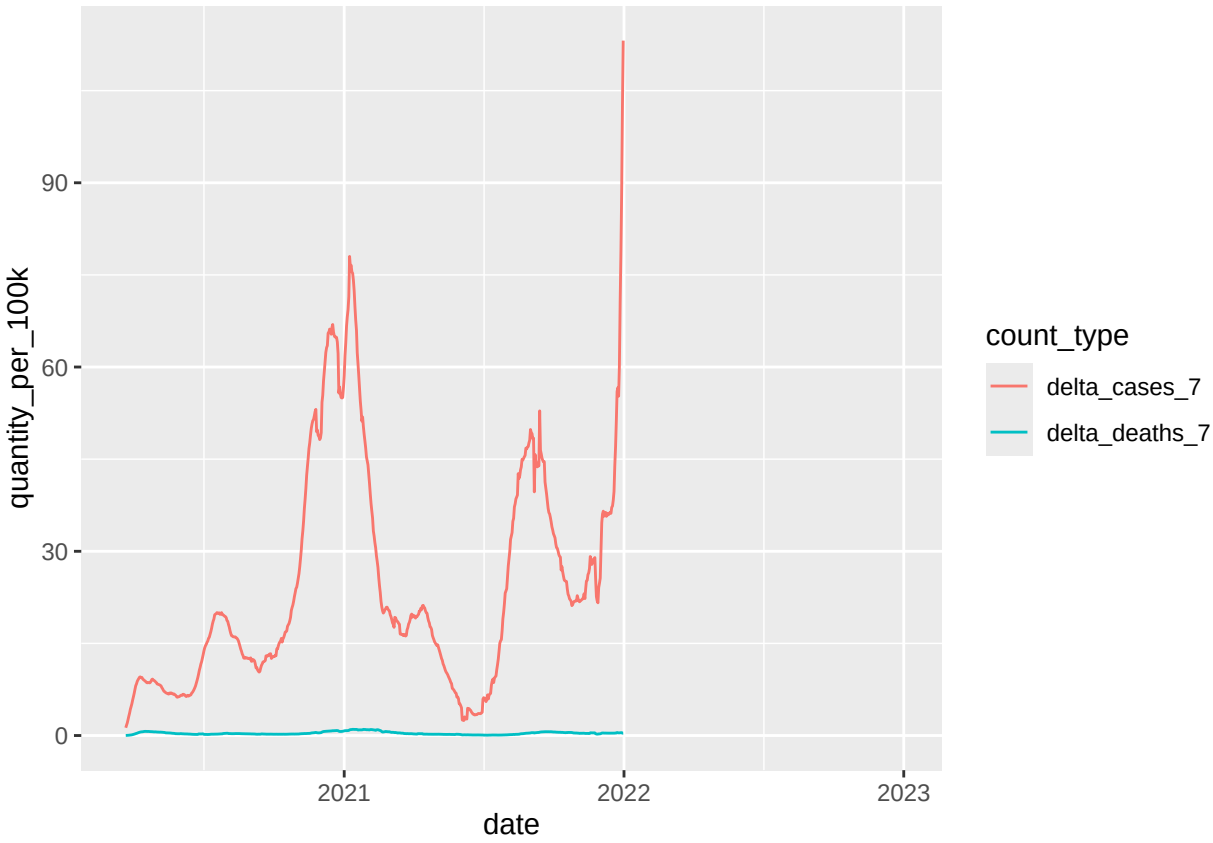
```
#  Your output should look similar to the following tibble:
#
#  date
#  total_deaths     >  the cumulative number of deaths up to and including the associated date
#  total_cases      >  the cumulative number of cases up to and including the associated date
#  delta_deaths_1   >  the number of new deaths since the previous day
#  delta_cases_1    >  the number of new cases since the previous day
#  delta_deaths_7   >  the average number of deaths in a seven-day period
#  delta_cases_7    >  the average number of cases in a seven-day period
#==
#  A tibble: 657 x 7
#       date         total_deaths   total_cases   delta_deaths_1   delta_cases_1 delta_deaths_7 delta_c
#      <date>            <dbl>         <dbl>          <dbl>           <dbl>           <dbl>         <db
#   1 2020-03-15         0.0205        1.08               0               0              NA           N
#   2 2020-03-16         0.0275        1.36         0.00694           0.274              NA           N
#   3 2020-03-17         0.0353        1.78         0.00784           0.422              NA           N
#   4 2020-03-18         0.0489        2.52          0.0136           0.737              NA           N
#   5 2020-03-19         0.0640        3.74          0.0151            1.22              NA           N
#   6 2020-03-20         0.0836        5.43          0.0196            1.69              NA           N
#   7 2020-03-21         0.108         7.39          0.0247            1.96              NA           N
#   8 2020-03-22         0.138         9.97          0.0296            2.58          0.0168          1.2
#   9 2020-03-23         0.174        13.1           0.0362            3.14          0.0209          1.6
#  10 2020-03-24         0.236        16.3           0.0621            3.14          0.0287          2.0
```

The above `us_covid_delta_per_100k` table shows the cumulative total deaths and cases, daily new deaths and cases, and the seven-day averages of daily new deaths and cases, all adjusted to a basis of per 100,000 people from March 15, 2020 to December 31, 2021. Values from 2022 are not available because the US population estimates data from the US Census Bureau only shows data from 2020 and 2021, and not from 2022.

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

# Pivot us_covid_delta_per_100k to combine delta_deaths_7 and delta_cases_7
# into a single column, then create a new column to indicate if the amount
# pertains to deaths or cases, so as to plot two line graphs,
# one for deaths and one for cases on the same plot

us_covid_delta_per_100k %>%
  select(date, delta_deaths_7, delta_cases_7) %>%
  pivot_longer(
    cols = c(delta_deaths_7, delta_cases_7),
    names_to = "count_type",
    values_to = "quantity_per_100k"
    ) %>%
  ggplot(mapping = aes(x = date, y = quantity_per_100k, color = count_type)) +
  geom_line()
```

Question 5

```
## Warning: Removed 744 rows containing missing values or values outside the scale range
## (`geom_line()`).
```

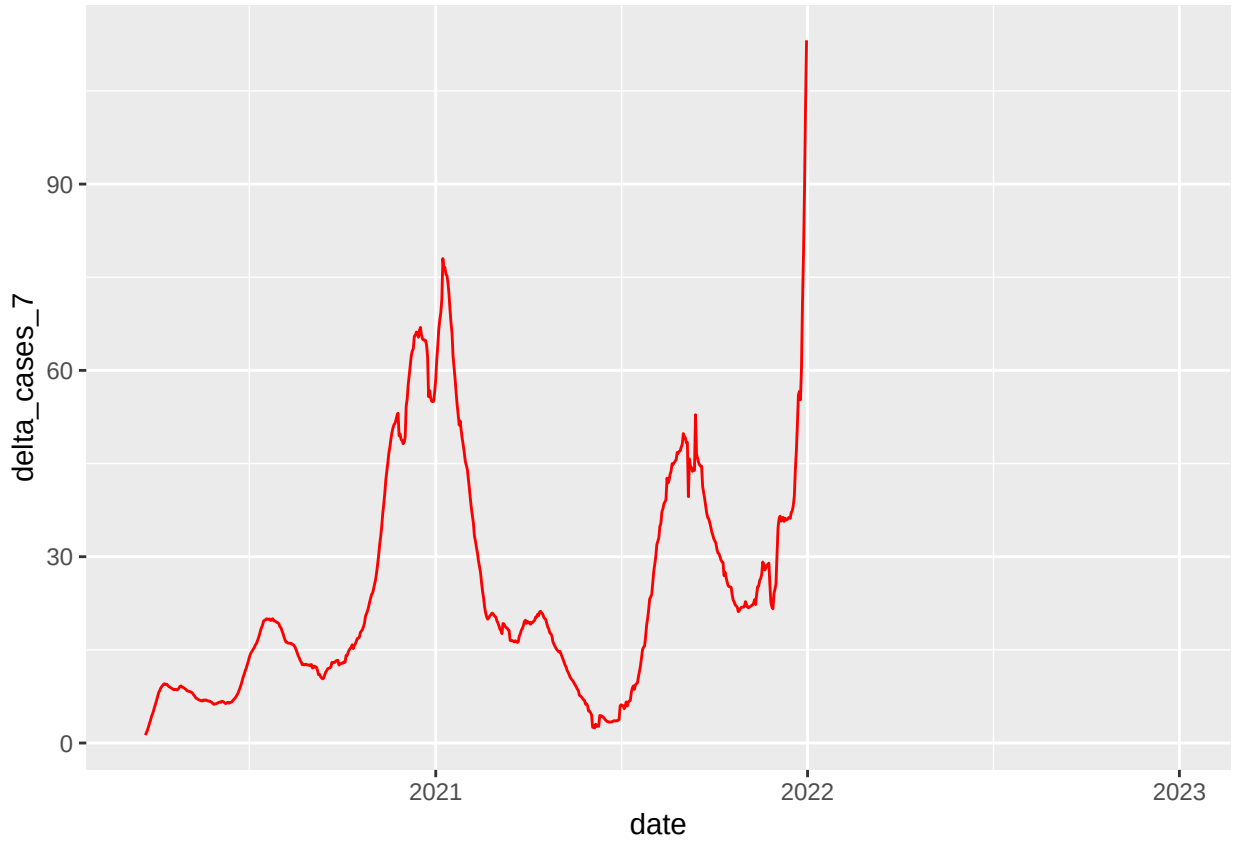


A line plot is most suitable for showing how the numerical variables representing the seven-day average cases and deaths per 100,000 people had changed over time. To generate the above line plot, the `us_covid_delta_per_100k` table was manipulated further by merging all counts into a single column and then adding a new column to indicate whether the count pertains to deaths or cases. The result is a red line plot showing the change in `delta_deaths_7` over time and a blue line plot showing the change in `delta_cases_7` over time. The line plots show that the seven-day average of Covid-19 cases per 100,000 people in the US increased from March 15, 2020 to about January 2021, then it fell until about May 2021, before increasing again until the end of 2021. As mentioned above, 2022 values are not included because of missing corresponding population data.

There doesn't seem to be any significant change in the seven-day average deaths per 100,000 people in the US. This could be because of this value being on a much smaller scale than what is shown in the plot. We can examine further by plotting the seven-day average of cases and deaths per 100,000 people on separate line plots:

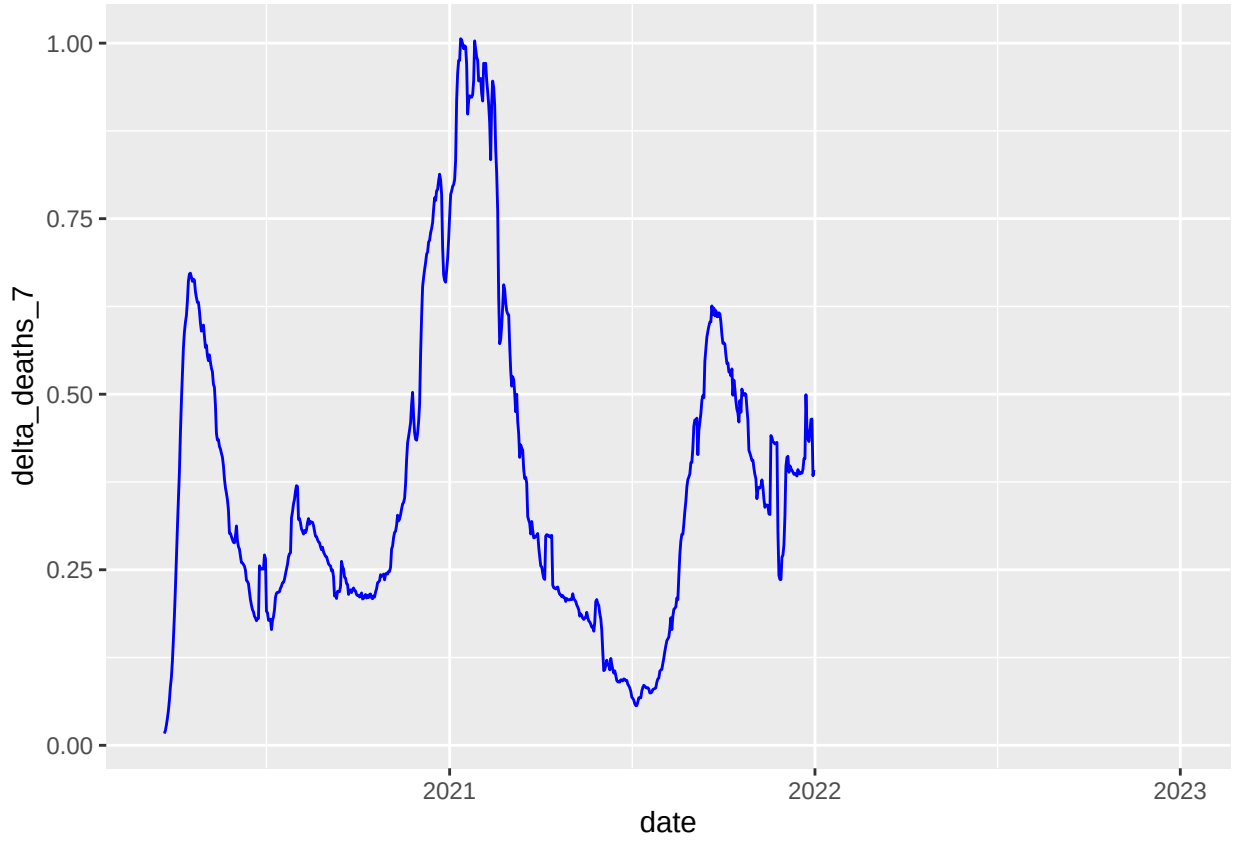
```
# Line plot of delta_cases_7 per 100k over time
ggplot(us_covid_delta_per_100k, mapping = aes(x = date, y = delta_cases_7)) +
  geom_line(color = "red")
```

```
## Warning: Removed 372 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```
# Line plot of delta_deaths_7 per 100k over time
ggplot(us_covid_delta_per_100k, mapping = aes(x = date, y = delta_deaths_7)) +
  geom_line(color = "blue")
```

```
## Warning: Removed 372 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



From the plot of `delta_deaths_7` over time, it can be seen that the seven-day average of deaths per 100,000 people in the US fluctuated between several highs and lows from March 15, 2020 to December 31, 2021, hitting its peak at about January 2021.

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

states_covid <- us_counties_combined %>%
  # Get data as of 2021-12-31
  filter(date == "2021-12-31") %>%
  group_by(state, date) %>%
  summarize(total_deaths = sum(deaths), total_cases = sum(cases)) %>%
  arrange(desc(total_cases))
```

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

```
head(states_covid, n = 10)
```

```
## # A tibble: 10 x 4
## # Groups:   state [10]
##    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
```

```
# Your transformed data should look similar to the following tibble:
#
# A tibble: 51 x 4
#    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 41 more rows
```

Using the `us_countied_combined` dataset, I filtered out rows for `2021-12-31`, then grouped deaths and cases by the `state` and `date` fields, then calculated the sum of deaths and cases for each state, then sorted the table by `total_cases` in descending order. On December 31, 2021, the 10 states with the highest number of `total_cases` in descending order are California, Texas, Florida, New York, Illinois, Pennsylvania, Ohio, Georgia, Michigan, and North Carolina.

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
```

```r
# Get state populations per year
states_population <- us_population_estimates %>%
  group_by(STNAME, Year) %>%
  summarize(state_population = sum(Estimate)) %>%
  rename("state" = STNAME) %>%
  # Filter by 2021 since the question looks at stats as of 2021-12-31
  filter(Year == 2021)
```

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

```r
# Join states_covid to states_population to add population data
states_covid %>% left_join(states_population, by= join_by(state)) %>%
  # Calculate deaths_per_100k and cases_per_100k by dividing by state population
  mutate(
    deaths_per_100k = total_deaths / state_population * 100000,
    cases_per_100k = total_cases / state_population * 100000
         ) %>%
  arrange(desc(cases_per_100k)) %>%
  select(state, date, deaths_per_100k, cases_per_100k)
```

```
## # A tibble: 56 x 4
## # Groups:   state [56]
##    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.
## # i 46 more rows
```

```r
# Your transformed data should look similar to the following tibble:
#
# A tibble: 51 x 4
#    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 41 more rows
```

To get population data for each state as `states_population`, I took the `us_population_estimates` dataset and grouped rows by `state` and `year` and obtained the total population in each state for each year. I filtered rows for year 2021 since the focus is on statistics as of `2021-12-31`. I joined the `states_covid` table to the `states_population` table and then created the variables `deaths_per_100k` and `cases_per_100k` by dividing each state's deaths and cases by their respective 2021 populations.

I did expect the above list to be different from the previous one that shows the total deaths and cases for each state, because the previous list did not consider the differences in state population sizes and therefore showed that the states with the higher numbers of deaths and cases were mostly the states with the highest populations. However, Covid-19 affected different parts of the country to different extents at a given point in time, irrespective of the population in that area. A spike usually started from one area and then gradually spreaded to others, so I would expect that at a single point in time, some states had a greater proportion of deaths and cases than others. The per 100,000 dataset seems to validate this by showing that it is not necessarily the states with the highest populations that were the most badly affected by Covid-19 as of December 31, 2021.

Reporting cases and deaths on a per 100,000 people basis is more informative because it shows figures on a proportional basis and prevents differences in population sizes from skewing the results when comparing between states.

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.

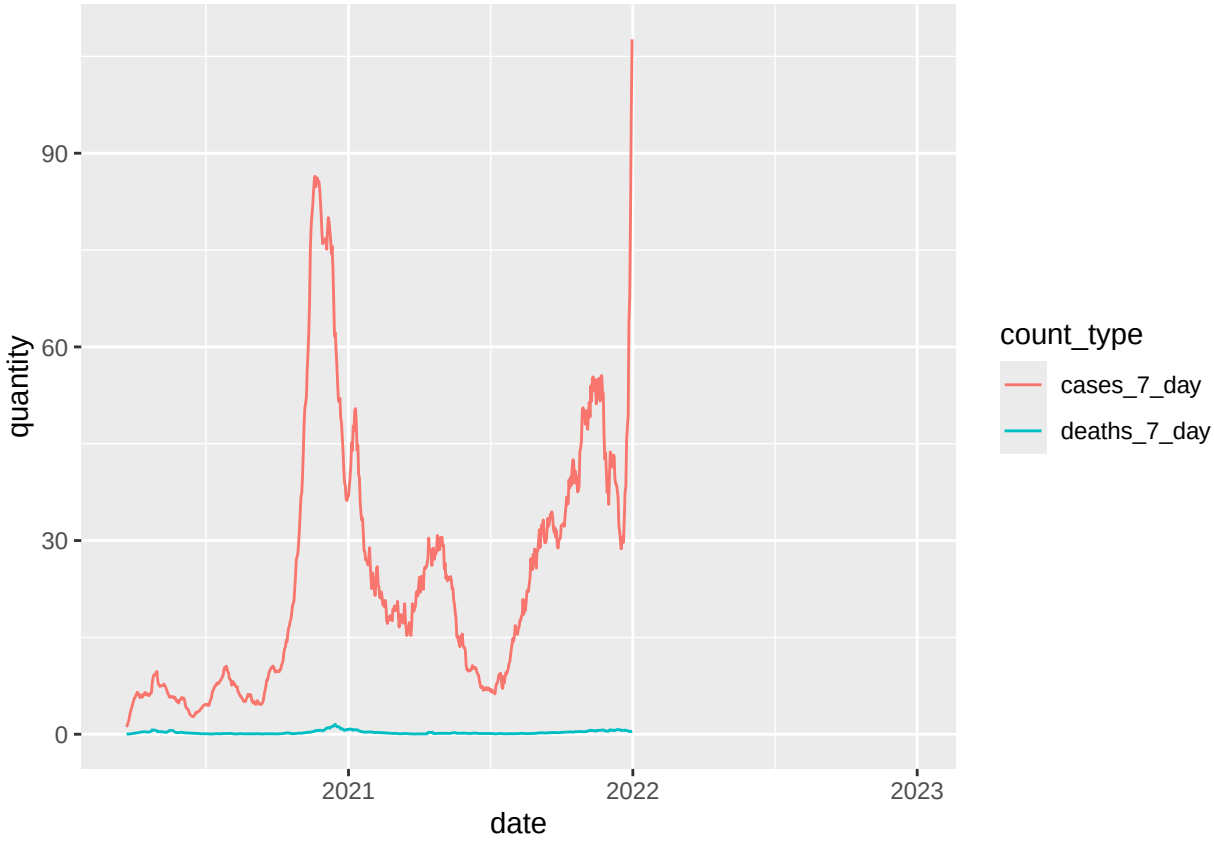
```
# Select a state and then filter by state and date range your data from Question 1. Calculate the seven

# First, get Colorado's population
co_population <- us_population_estimates %>%
  filter(STNAME == "Colorado") %>%
  group_by(Year) %>%
  summarize(total_population = sum(Estimate))
# Get 2020 and 2021 populations
co_population_2020 <- co_population %>%
  filter(Year == 2020) %>%
  select(total_population) %>%
  deframe()
co_population_2021 <- co_population %>%
  filter(Year == 2021) %>%
  select(total_population) %>%
  deframe()
# Then, calculate total deaths and cases in CO
co_covid_delta <- us_counties_combined %>%
  filter(state == "Colorado", date >= "2020-03-15") %>%
  group_by(date) %>%
  summarize(total_deaths = sum(deaths), total_cases = sum(cases)) %>%
  mutate(
    # Add CO population for corresponding year
    population = case_when(
      grepl("2020", date) ~ co_population_2020,
      grepl("2021", date) ~ co_population_2021
      ),
    # Calculate daily new deaths and cases
    delta_deaths_1 = total_deaths - lag(total_deaths, n = 1),
    delta_cases_1 = total_cases - lag(total_cases, n = 1),
    # Calculate seven-day averages of cases and deaths per 100,000
```

```
    deaths_7_day = frollmean(delta_deaths_1, n = 7) / population * 100000,
    cases_7_day = frollmean(delta_cases_1, n = 7) / population * 100000,
    # Caculate total deaths and cases per 100,000
    deaths_per_100k = total_deaths / population * 100000,
    cases_per_100k = total_cases / population * 100000
    ) %>%
  select(date, total_deaths, total_cases, population, deaths_per_100k, cases_per_100k, deaths_7_day, cas

head(co_covid_delta, n = 10)
```

```
## # A tibble: 10 x 8
##    date       total_deaths total_cases population deaths_per_100k cases_per_100k
##    <date>            <dbl>       <dbl>      <dbl>           <dbl>           <dbl>
##  1 2020-03-15            2         136    5784308          0.0346            2.35
##  2 2020-03-16            2         161    5784308          0.0346            2.78
##  3 2020-03-17            3         183    5784308          0.0519            3.16
##  4 2020-03-18            3         216    5784308          0.0519            3.73
##  5 2020-03-19            5         278    5784308          0.0864            4.81
##  6 2020-03-20            5         364    5784308          0.0864            6.29
##  7 2020-03-21            6         475    5784308          0.104             8.21
##  8 2020-03-22            7         591    5784308          0.121            10.2
##  9 2020-03-23           10         721    5784308          0.173            12.5
## 10 2020-03-24           11         912    5784308          0.190            15.8
## # i 2 more variables: deaths_7_day <dbl>, cases_7_day <dbl>
```

```
# Create line plot showing seven-day averages of new deaths and cases over time
co_covid_delta %>%
  # Merge deaths_7_day and cases_7_day into same column
  # to plot on same line plot
  pivot_longer(
    cols = c(deaths_7_day, cases_7_day),
    names_to = "count_type",
    values_to = "quantity"
  ) %>%
  ggplot(mapping = aes(x = date, y = quantity, color = count_type)) +
  geom_line()
```

```
## Warning: Removed 744 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```
# Your transformed data should look similar to the following tibble:
#
# A tibble: 656 × 9
#     state    date       total_deaths total_cases population deaths_per_100k cases_per_100k deaths_7_d
#     <chr>    <date>            <dbl>       <dbl>      <dbl>           <dbl>          <dbl>        <dbl>
#  1 Colorado 2020-03-15            2         136    5784308          0.0346           2.35           NA
#  2 Colorado 2020-03-16            2         161    5784308          0.0346           2.78           NA
#  3 Colorado 2020-03-17            3         183    5784308          0.0519           3.16           NA
#  4 Colorado 2020-03-18            3         216    5784308          0.0519           3.73           NA
#  5 Colorado 2020-03-19            5         278    5784308          0.0864           4.81           NA
#  6 Colorado 2020-03-20            5         364    5784308          0.0864           6.29           NA
#  7 Colorado 2020-03-21            6         475    5784308          0.104            8.21           NA
#  8 Colorado 2020-03-22            7         591    5784308          0.121           10.2          0.0123
#  9 Colorado 2020-03-23           10         721    5784308          0.173           12.5          0.0198
# 10 Colorado 2020-03-24           11         912    5784308          0.190           15.8          0.0198
# ... with 646 more rows
```

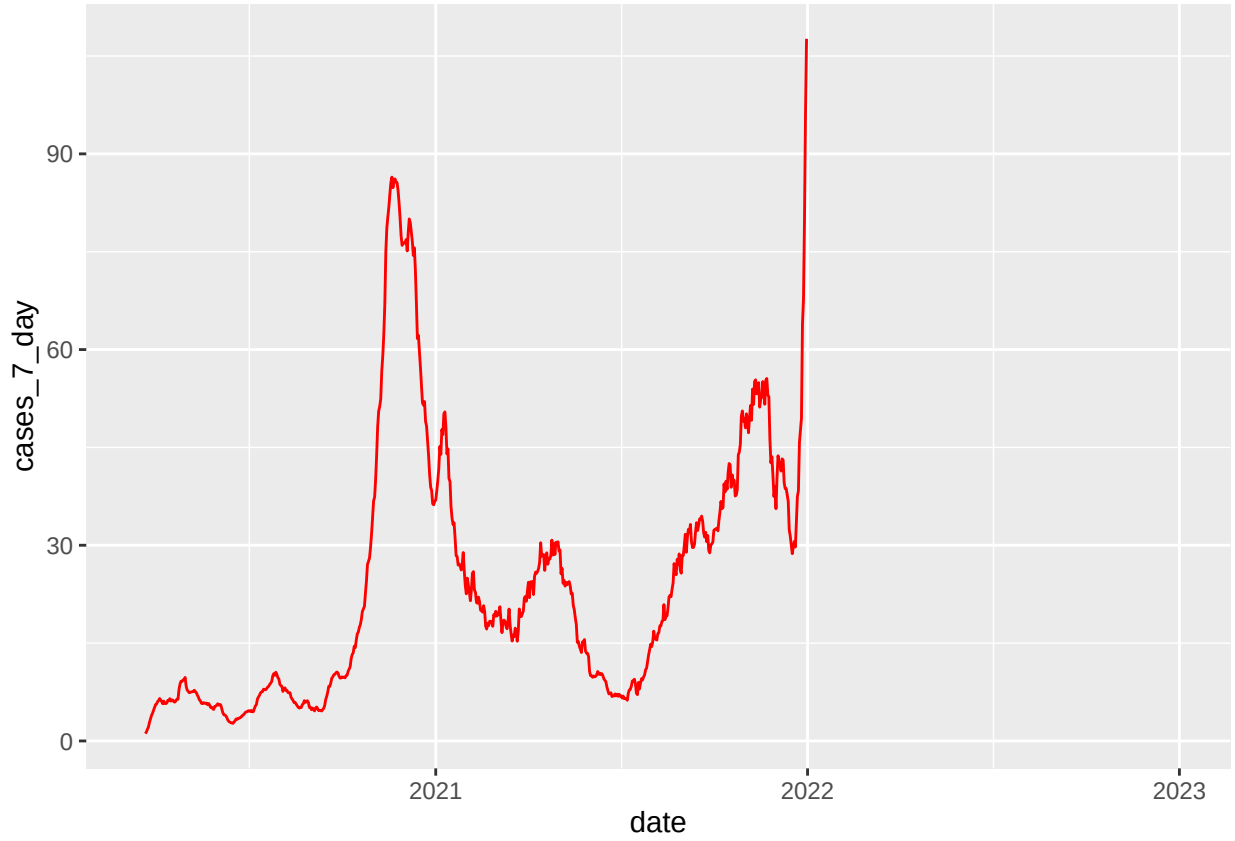
From the above line plot, the seven-day average of new daily cases in Colorado had increased from March 15, 2020 to the end of 2020, then decreased to the middle of 2021, before increasing again towards the end of 2021. Values in 2022 are not included because population data for 2022 is not available in `us_population_estimates`.

The seven-day average of new daily deaths per 100,000 people in Colorado seems to be near zero throughout the same time period. While the value is small, any significant change to this value cannot be seen clearly from the above plot because this value is on a much smaller scale than the seven-day average of new daily cases per 100,000 people. We can examine further by plotting the seven-day average of cases and deaths per

100,000 people on separate line plots:

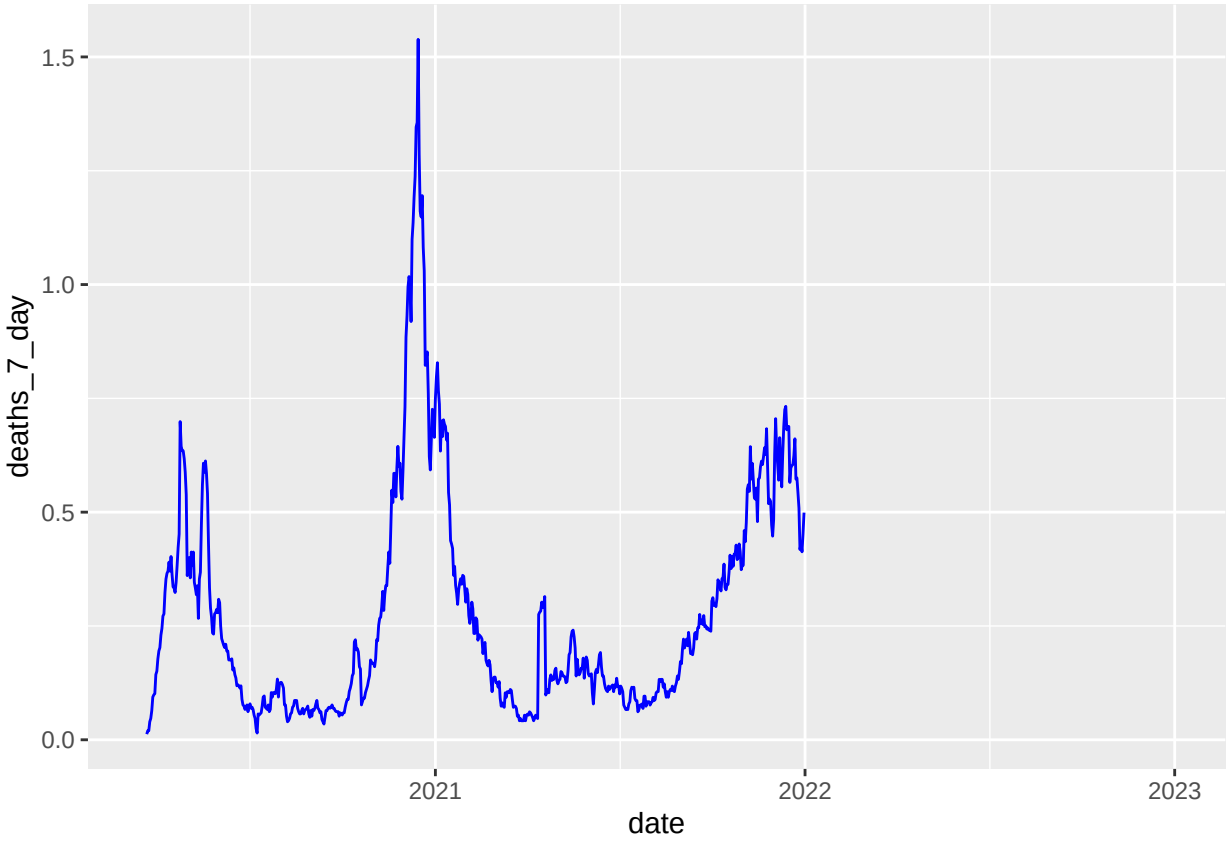
```
# Line plot of cases_7_day per 100k over time
ggplot(co_covid_delta, mapping = aes(x = date, y = cases_7_day)) +
  geom_line(color = "red")
```

```
## Warning: Removed 372 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```
# Line plot of deaths_7_day per 100k over time
ggplot(co_covid_delta, mapping = aes(x = date, y = deaths_7_day)) +
  geom_line(color = "blue")
```

```
## Warning: Removed 372 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



Although the seven-day average of new daily deaths per 100,000 in Colorado is indeed small from March 15, 2020 to December 31, 2021, there are significant trends which actually mirror those observed from the seven-day average of new daily cases per 100,000 in Colorado.

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

# Get CO counties' populations
co_counties_population <- us_population_estimates %>%
  filter(STNAME == "Colorado") %>%
  select(CTYNAME, Year, Estimate) %>%
  rename(county = CTYNAME, year = Year, population = Estimate) %>%
  # remove the suffix " County" from county to match the county field
  # in the Covid data for doing a join
  mutate(county = str_replace_all(county, "\\sCounty", ""))

head(co_counties_population, n = 10)
```

```
## # A tibble: 10 x 3
##    county     year population
##    <chr>     <dbl>      <dbl>
##  1 Adams      2020     520163
##  2 Adams      2021     522140
```

```
##  3 Alamosa    2020      16362
##  4 Alamosa    2021      16547
##  5 Arapahoe   2020     655112
##  6 Arapahoe   2021     654900
##  7 Archuleta  2020      13424
##  8 Archuleta  2021      13790
##  9 Baca       2020       3486
## 10 Baca       2021       3514
```

```
# Get CO counties' cumulative total deaths and cases per 100,000
# as of most recent date, 2021-12-31
co_counties_covid <- us_counties_combined %>%
  filter(state == "Colorado", (date == "2021-12-31")) %>%
  mutate(year = year(date)) %>%
  left_join(co_counties_population, by = join_by(county, year)) %>%
  mutate(
    total_deaths = deaths / population * 100000,
    total_cases = cases / population * 10000
    ) %>%
  select(county, date, fips, total_deaths, total_cases)

# Top 5 counties in terms of deaths
co_counties_covid %>%
  arrange(desc(total_deaths)) %>%
  slice_head(n = 5)
```

```
## # A tibble: 5 x 5
##   county     date       fips  total_deaths total_cases
##   <chr>      <date>     <chr>        <dbl>       <dbl>
## 1 Bent       2021-12-31 08011         729.       3400.
## 2 Otero      2021-12-31 08089         624.       1811.
## 3 Conejos    2021-12-31 08021         604.       1567.
## 4 Washington 2021-12-31 08121         514.       1588.
## 5 Cheyenne   2021-12-31 08017         469.       1459.
```

```
# Top 5 counties in terms of cases
co_counties_covid %>%
  arrange(desc(total_cases)) %>%
  slice_head(n = 5)
```

```
## # A tibble: 5 x 5
##   county  date       fips  total_deaths total_cases
##   <chr>   <date>     <chr>        <dbl>       <dbl>
## 1 Crowley 2021-12-31 08025        449.        4157.
## 2 Bent    2021-12-31 08011        729.        3400.
## 3 Lincoln 2021-12-31 08073        141.        2525.
## 4 Logan   2021-12-31 08075        451.        2488.
## 5 Pitkin  2021-12-31 08097         34.6       2399.
```

```
# Your transformed data should be similar to the following tibbles:
#
# Arranged by deaths:
```

```
# A tibble: 64 × 4
#    county       date       fips    total_deaths   total_cases
#    <chr>        <date>     <chr>          <dbl>         <dbl>
#  1 El Paso     2021-12-20  08041           1355        119772
#  2 Denver      2021-12-20  08031           1065        106747
#  3 Jefferson   2021-12-20  08059           1061         76732
#  4 Adams       2021-12-20  08001           1057         90476
#  5 Arapahoe    2021-12-20  08005           1046         95769
#  6 Pueblo      2021-12-20  08101            643         30739
#  7 Weld        2021-12-20  08123            569         55599
#  8 Mesa        2021-12-20  08077            445         29542
#  9 Larimer     2021-12-20  08069            393         47444
# 10 Douglas     2021-12-20  08035            361         48740
# ... with 54 more rows
#
#
# Arranged by cases:
# A tibble: 64 × 4
#    county       date       fips    total_deaths   total_cases
#    <chr>        <date>     <chr>          <dbl>         <dbl>
#  1 El Paso     2021-12-20  08041           1355        119772
#  2 Denver      2021-12-20  08031           1065        106747
#  3 Arapahoe    2021-12-20  08005           1046         95769
#  4 Adams       2021-12-20  08001           1057         90476
#  5 Jefferson   2021-12-20  08059           1061         76732
#  6 Weld        2021-12-20  08123            569         55599
#  7 Douglas     2021-12-20  08035            361         48740
#  8 Larimer     2021-12-20  08069            393         47444
#  9 Boulder     2021-12-20  08013            323         36754
# 10 Pueblo      2021-12-20  08101            643         30739
# ... with 54 more rows
```

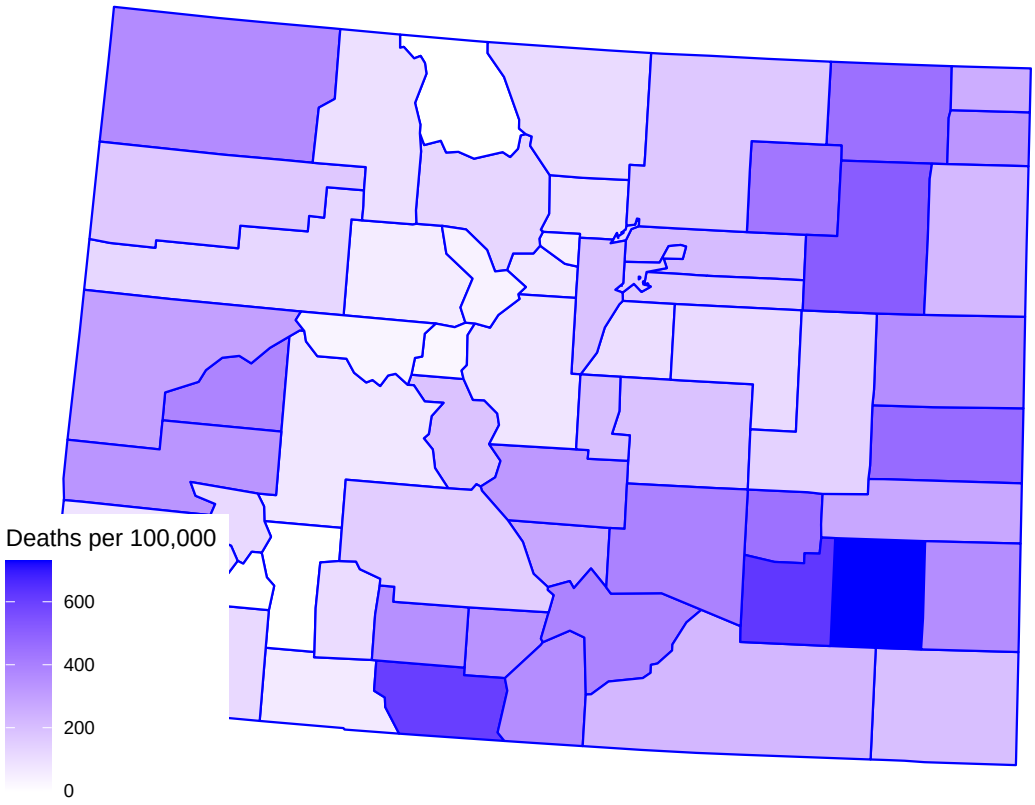
The population for each Colorado county for each year was taken from the `us_population_estimates` data and then joined to the Covid deaths and cases data for Colorado counties by county name and year, after performing the necessary transformations to make the county names the same across both datasets. Then, the cumulative total deaths and cases per 100,000 people as of December 31, 2022 were calculated. The five counties with the highest total deaths per 100,000 people as of December 31, 2022 are Bent, Otero, Conejos, Washington, and Cheyenne. The five counties with the highest total cases per 100,000 people as of December 31, 2022 are Crowley, Bent, Lincoln, Logan, and Pitkin.

Comparing on a per 100,000 basis is useful because it adjusts for the significant differences in population size across different Colorado counties. It should be noted that there are numerous counties with population of less than 100,000, which means that their per 100,000 figures are larger than their actual total deaths and cases. Also, for counties with small populations, a single death or case would be of a bigger proportion of the county's population than for counties with larger populations, meaning that counties with small populations are more likely to show higher per 100,000 deaths and counts.

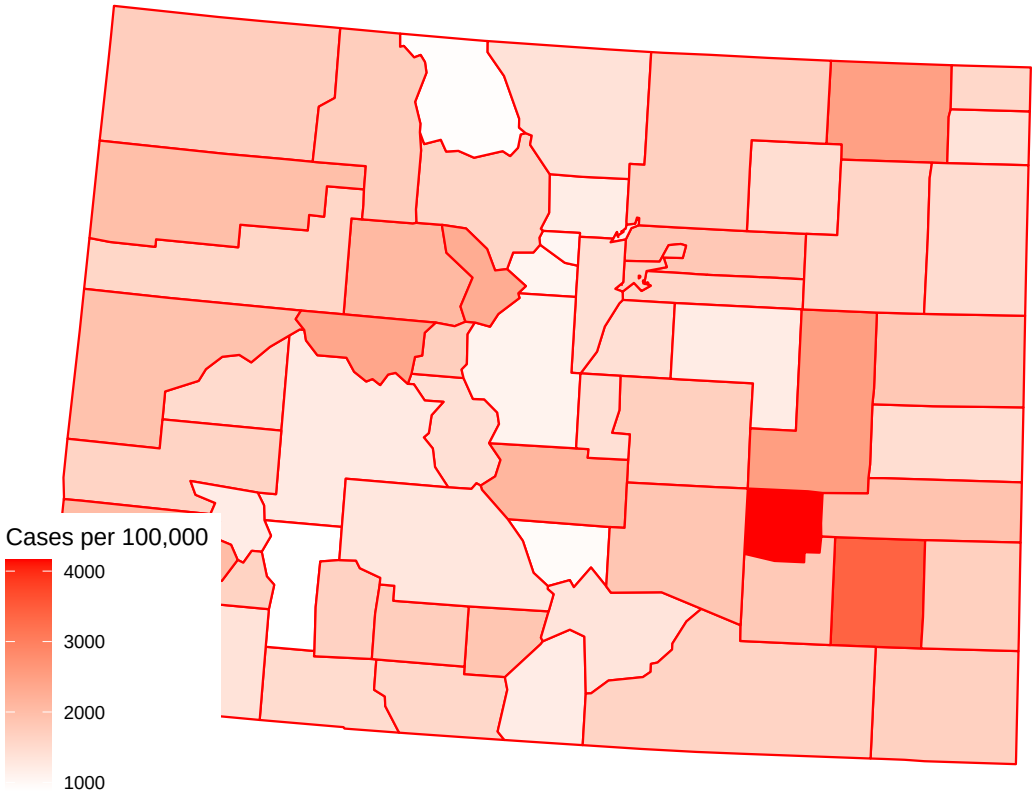
Question 5 Modify the code below for the map projection to plot county-level deaths and cases per 100,000 people for your state.

```
# Create data frame with two columns for fips and total_deaths
co_counties_deaths_plot = co_counties_covid %>%
  select(fips, total_deaths) %>% as.data.frame()
```

```
# Plot map of CO counties' total deaths per 100,000 with a color scale
plot_usmap(regions = "county", include = "CO", data = co_counties_deaths_plot, values = "total_deaths",
  scale_fill_continuous(low = "white", high = "blue", name = "Deaths per 100,000")
```




```
# Create data frame with two columns for fips and total_cases
co_counties_cases_plot = co_counties_covid %>%
  select(fips, total_cases) %>% as.data.frame()

# Plot map of CO counties' total cases per 100,000 with a color scale
plot_usmap(regions = "county", include = "CO", data = co_counties_cases_plot, values = "total_cases", co
  scale_fill_continuous(low = "white", high = "red", name = "Cases per 100,000")
```




```
# Copy and modify the code below for your state.
#
# plot_usmap arguments:
#   regions: can be one of ("states", "state", "counties", "county"). The default is "states"
#   include: The regions to include in the resulting map. If regions is "states"/"state", the value can
#   data: values to plot on the map
#   values: the name of the column that contains the values to be associated with a given region.
#   color: the map outline color.
#
# Reference the plot_usmap documentation for further information using ?plot_usmap

# plot_usmap(regions = "county", include="CO", data = colorado_county, values = "total_deaths", color =
# scale_fill_continuous(low = "white", high = "blue", name = "Deaths per 100,000")
```

Separate plots were made above using the `plot_usmap` function to show the amounts of total deaths and total cases per 100,000 population respectively across the different counties in the state of Colorado. For each plot, a new data frame with just two columns, `fips` and the value to measure, was created to use as the `data` argument for the `plot_usmap` function.

The plots show that on the overall, the eastern counties of Colorado had more total deaths and cases per 100,000 population than other counties, followed by the western counties, and then the counties in the center.

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.

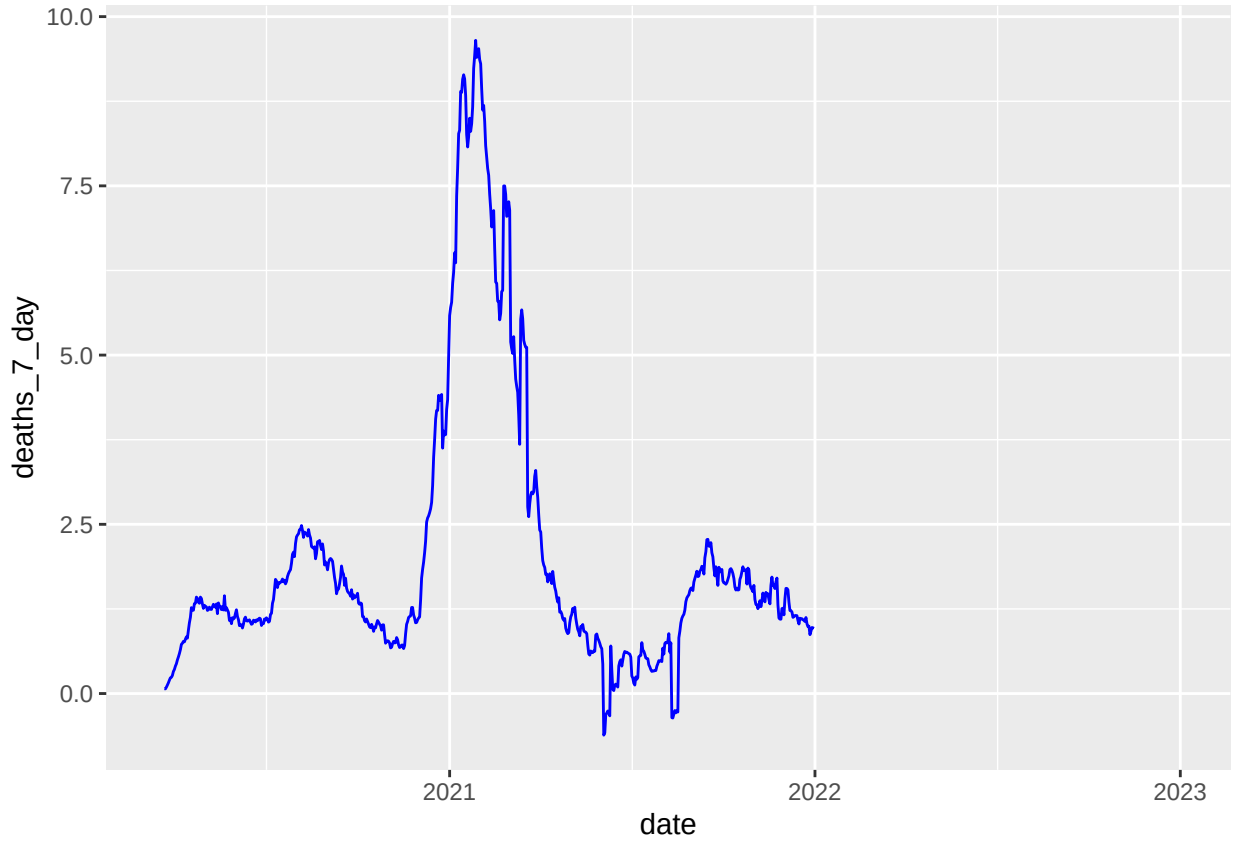
```r
# California

# First, get California's population
ca_population <- us_population_estimates %>%
  filter(STNAME == "California") %>%
  group_by(Year) %>%
  summarize(total_population = sum(Estimate))
# Get 2020 and 2021 populations
ca_population_2020 <- ca_population %>%
  filter(Year == 2020) %>%
  select(total_population) %>%
  deframe()
ca_population_2021 <- ca_population %>%
  filter(Year == 2021) %>%
  select(total_population) %>%
  deframe()
# Then, calculate total deaths and cases in CA
ca_covid_delta <- us_counties_combined %>%
  filter(state == "California", date >= "2020-03-15") %>%
  group_by(date) %>%
  summarize(total_deaths = sum(deaths), total_cases = sum(cases)) %>%
  mutate(
    # Add CO population for corresponding year
    population = case_when(
      grepl("2020", date) ~ co_population_2020,
      grepl("2021", date) ~ co_population_2021
      ),
    # Calculate daily new deaths and cases
    delta_deaths_1 = total_deaths - lag(total_deaths, n = 1),
    delta_cases_1 = total_cases - lag(total_cases, n = 1),
    # Calculate seven-day averages of cases and deaths per 100,000
    deaths_7_day = frollmean(delta_deaths_1, n = 7) / population * 100000,
    cases_7_day = frollmean(delta_cases_1, n = 7) / population * 100000,
    # Caculate total deaths and cases per 100,000
    deaths_per_100k = total_deaths / population * 100000,
    cases_per_100k = total_cases / population * 100000
    ) %>%
  select(
    date,
    total_deaths,
    total_cases,
    population,
    deaths_per_100k,
    cases_per_100k,
    deaths_7_day,
    cases_7_day
    )

head(ca_covid_delta, n = 10)
```

```
## # A tibble: 10 x 8
##    date       total_deaths total_cases population deaths_per_100k cases_per_100k
##    <date>            <dbl>       <dbl>      <dbl>           <dbl>          <dbl>
```

```
##  1 2020-03-15            6         478    5784308           0.104           8.26
##  2 2020-03-16           11         588    5784308           0.190          10.2
##  3 2020-03-17           14         732    5784308           0.242          12.7
##  4 2020-03-18           17         893    5784308           0.294          15.4
##  5 2020-03-19           19        1067    5784308           0.328          18.4
##  6 2020-03-20           24        1283    5784308           0.415          22.2
##  7 2020-03-21           28        1544    5784308           0.484          26.7
##  8 2020-03-22           35        1851    5784308           0.605          32.0
##  9 2020-03-23           39        2240    5784308           0.674          38.7
## 10 2020-03-24           52        2644    5784308           0.899          45.7
## # i 2 more variables: deaths_7_day <dbl>, cases_7_day <dbl>
```

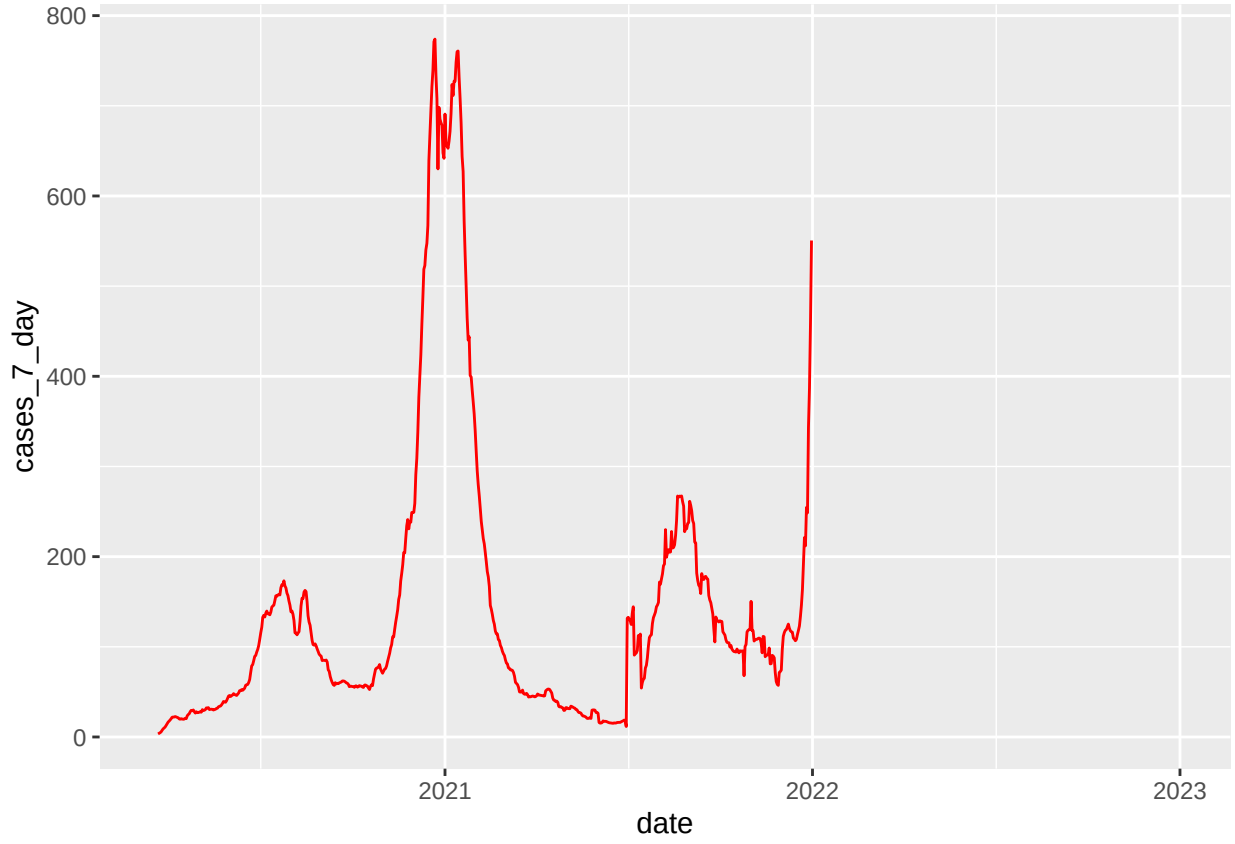
```r
# Create line plot showing CA's seven-day averages of new deaths over time
ca_covid_delta %>%
  ggplot(mapping = aes(x = date, y = deaths_7_day)) +
  geom_line(color = "blue")
```

```
## Warning: Removed 372 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```r
# Create line plot showing CA's seven-day averages of new cases over time
ca_covid_delta %>%
  ggplot(mapping = aes(x = date, y = cases_7_day)) +
  geom_line(color = "red")
```

```
## Warning: Removed 372 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



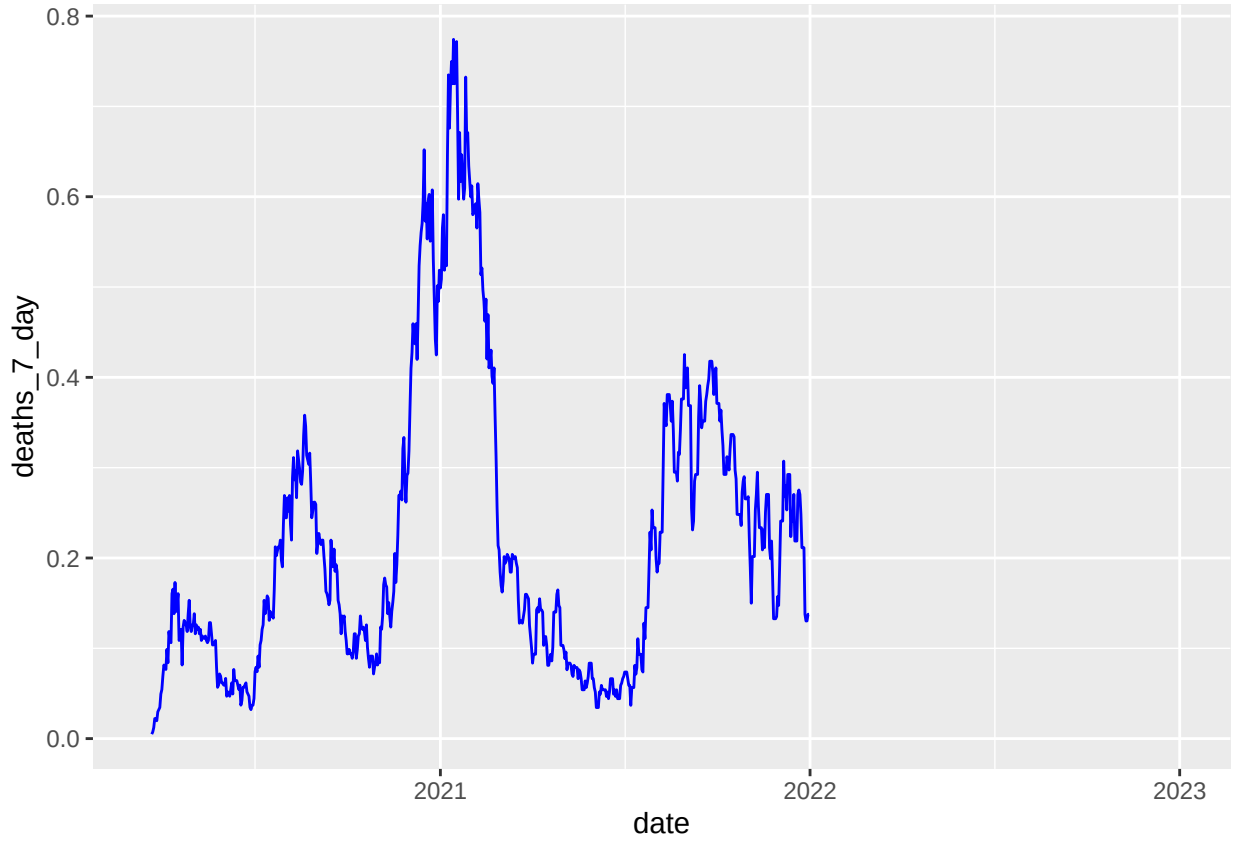

```
# Nevada

# First, get Nevada's population
nv_population <- us_population_estimates %>%
  filter(STNAME == "Nevada") %>%
  group_by(Year) %>%
  summarize(total_population = sum(Estimate))
# Get 2020 and 2021 populations
nv_population_2020 <- nv_population %>%
  filter(Year == 2020) %>%
  select(total_population) %>%
  deframe()
nv_population_2021 <- nv_population %>%
  filter(Year == 2021) %>%
  select(total_population) %>%
  deframe()
# Then, calculate total deaths and cases in NV
nv_covid_delta <- us_counties_combined %>%
  filter(state == "Nevada", date >= "2020-03-15") %>%
  group_by(date) %>%
  summarize(total_deaths = sum(deaths), total_cases = sum(cases)) %>%
  mutate(
    # Add NV population for corresponding year
```

```
    population = case_when(
      grepl("2020", date) ~ co_population_2020,
      grepl("2021", date) ~ co_population_2021
      ),
    # Calculate daily new deaths and cases
    delta_deaths_1 = total_deaths - lag(total_deaths, n = 1),
    delta_cases_1 = total_cases - lag(total_cases, n = 1),
    # Calculate seven-day averages of cases and deaths per 100,000
    deaths_7_day = frollmean(delta_deaths_1, n = 7) / population * 100000,
    cases_7_day = frollmean(delta_cases_1, n = 7) / population * 100000,
    # Caculate total deaths and cases per 100,000
    deaths_per_100k = total_deaths / population * 100000,
    cases_per_100k = total_cases / population * 100000
    ) %>%
  select(
    date,
    total_deaths,
    total_cases,
    population,
    deaths_per_100k,
    cases_per_100k,
    deaths_7_day,
    cases_7_day
    )

head(nv_covid_delta, n = 10)
```

```
## # A tibble: 10 x 8
##    date       total_deaths total_cases population deaths_per_100k cases_per_100k
##    <date>            <dbl>       <dbl>      <dbl>           <dbl>          <dbl>
##  1 2020-03-15            0          26    5784308          0               0.449
##  2 2020-03-16            1          45    5784308          0.0173          0.778
##  3 2020-03-17            1          55    5784308          0.0173          0.951
##  4 2020-03-18            1          82    5784308          0.0173          1.42
##  5 2020-03-19            1          99    5784308          0.0173          1.71
##  6 2020-03-20            2         165    5784308          0.0346          2.85
##  7 2020-03-21            2         165    5784308          0.0346          2.85
##  8 2020-03-22            2         190    5784308          0.0346          3.28
##  9 2020-03-23            4         265    5784308          0.0692          4.58
## 10 2020-03-24            6         315    5784308          0.104           5.45
## # i 2 more variables: deaths_7_day <dbl>, cases_7_day <dbl>
```

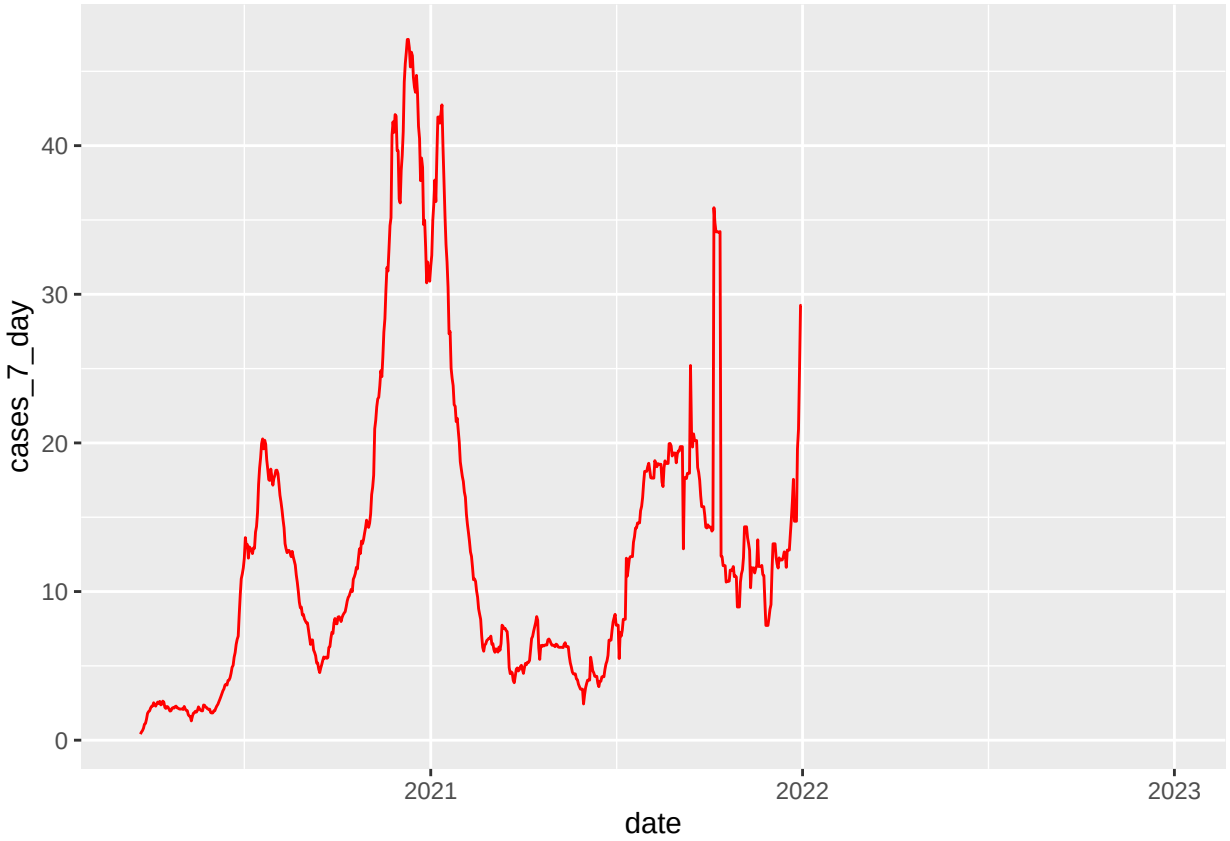
```
# Create line plot showing NV's seven-day averages of new deaths over time
nv_covid_delta %>%
  ggplot(mapping = aes(x = date, y = deaths_7_day)) +
  geom_line(color = "blue")
```

```
## Warning: Removed 372 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```
# Create line plot showing NV's seven-day averages of new cases over time
nv_covid_delta %>%
  ggplot(mapping = aes(x = date, y = cases_7_day)) +
  geom_line(color = "red")
```

```
## Warning: Removed 372 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



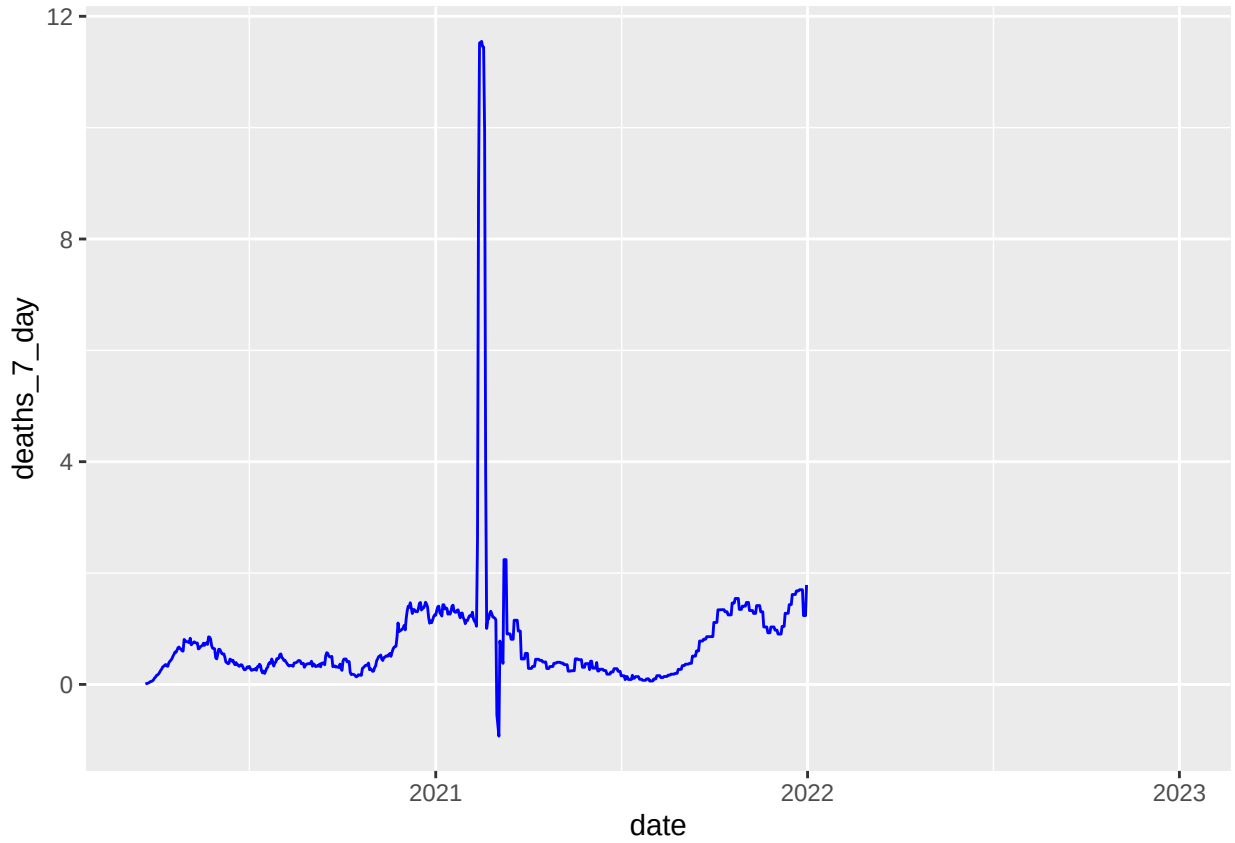

```
# Ohio

# First, get Ohio's population
oh_population <- us_population_estimates %>%
  filter(STNAME == "Ohio") %>%
  group_by(Year) %>%
  summarize(total_population = sum(Estimate))
# Get 2020 and 2021 populations
oh_population_2020 <- oh_population %>%
  filter(Year == 2020) %>%
  select(total_population) %>%
  deframe()
oh_population_2021 <- oh_population %>%
  filter(Year == 2021) %>%
  select(total_population) %>%
  deframe()
# Then, calculate total deaths and cases in OH
oh_covid_delta <- us_counties_combined %>%
  filter(state == "Ohio", date >= "2020-03-15") %>%
  group_by(date) %>%
  summarize(total_deaths = sum(deaths), total_cases = sum(cases)) %>%
  mutate(
    # Add OH population for corresponding year
    population = case_when(
      grepl("2020", date) ~ co_population_2020,
      grepl("2021", date) ~ co_population_2021
```

```
      ),
    # Calculate daily new deaths and cases
    delta_deaths_1 = total_deaths - lag(total_deaths, n = 1),
    delta_cases_1 = total_cases - lag(total_cases, n = 1),
    # Calculate seven-day averages of cases and deaths per 100,000
    deaths_7_day = frollmean(delta_deaths_1, n = 7) / population * 100000,
    cases_7_day = frollmean(delta_cases_1, n = 7) / population * 100000,
    # Caculate total deaths and cases per 100,000
    deaths_per_100k = total_deaths / population * 100000,
    cases_per_100k = total_cases / population * 100000
    ) %>%
  select(
    date,
    total_deaths,
    total_cases,
    population,
    deaths_per_100k,
    cases_per_100k,
    deaths_7_day,
    cases_7_day
    )

head(oh_covid_delta, n = 10)
```

```
## # A tibble: 10 x 8
##    date       total_deaths total_cases population deaths_per_100k cases_per_100k
##    <date>            <dbl>       <dbl>      <dbl>           <dbl>          <dbl>
##  1 2020-03-15            0          37    5784308          0               0.640
##  2 2020-03-16            0          50    5784308          0               0.864
##  3 2020-03-17            0          67    5784308          0               1.16
##  4 2020-03-18            0          90    5784308          0               1.56
##  5 2020-03-19            0         120    5784308          0               2.07
##  6 2020-03-20            1         169    5784308          0.0173          2.92
##  7 2020-03-21            3         247    5784308          0.0519          4.27
##  8 2020-03-22            3         351    5784308          0.0519          6.07
##  9 2020-03-23            6         444    5784308          0.104           7.68
## 10 2020-03-24            8         564    5784308          0.138           9.75
## # i 2 more variables: deaths_7_day <dbl>, cases_7_day <dbl>
```

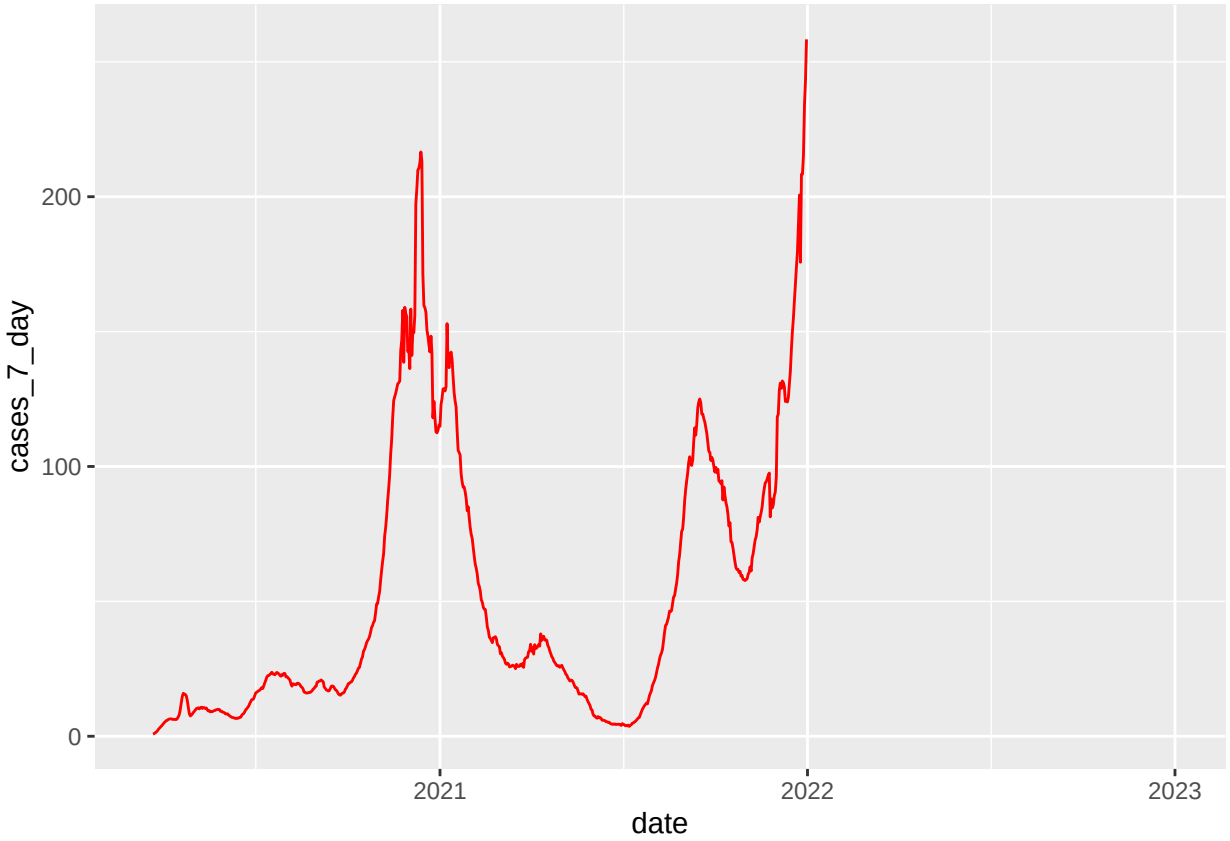
```
# Create line plot showing OH's seven-day averages of new deaths over time
oh_covid_delta %>%
  ggplot(mapping = aes(x = date, y = deaths_7_day)) +
  geom_line(color = "blue")
```

```
## Warning: Removed 372 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```
# Create line plot showing OH's seven-day averages of new cases over time
oh_covid_delta %>%
  ggplot(mapping = aes(x = date, y = cases_7_day)) +
  geom_line(color = "red")
```

```
## Warning: Removed 372 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



In California, the 7-day average of new deaths per 100,000 population rose to a peak of almost 10.0 in early 2021, before falling to about 1.25 by the end of 2021. The 7-day average of new cases per 100,000 population rose to a peak of about 780 at the end of 2020, before falling to about 50 towards the middle of 2021, and then increasing again to about 550 at the end of 2021.

In Nevada, the 7-day average of new deaths per 100,000 population rose to a peak of about 0.78 in early 2021, before falling to about 0.125 by the end of 2021. The 7-day average of new cases per 100,000 population rose to about 47 at the end of 2020, before falling to about 5 in the middle of 2021, and then increasing again to about 30 at the end of 2021.

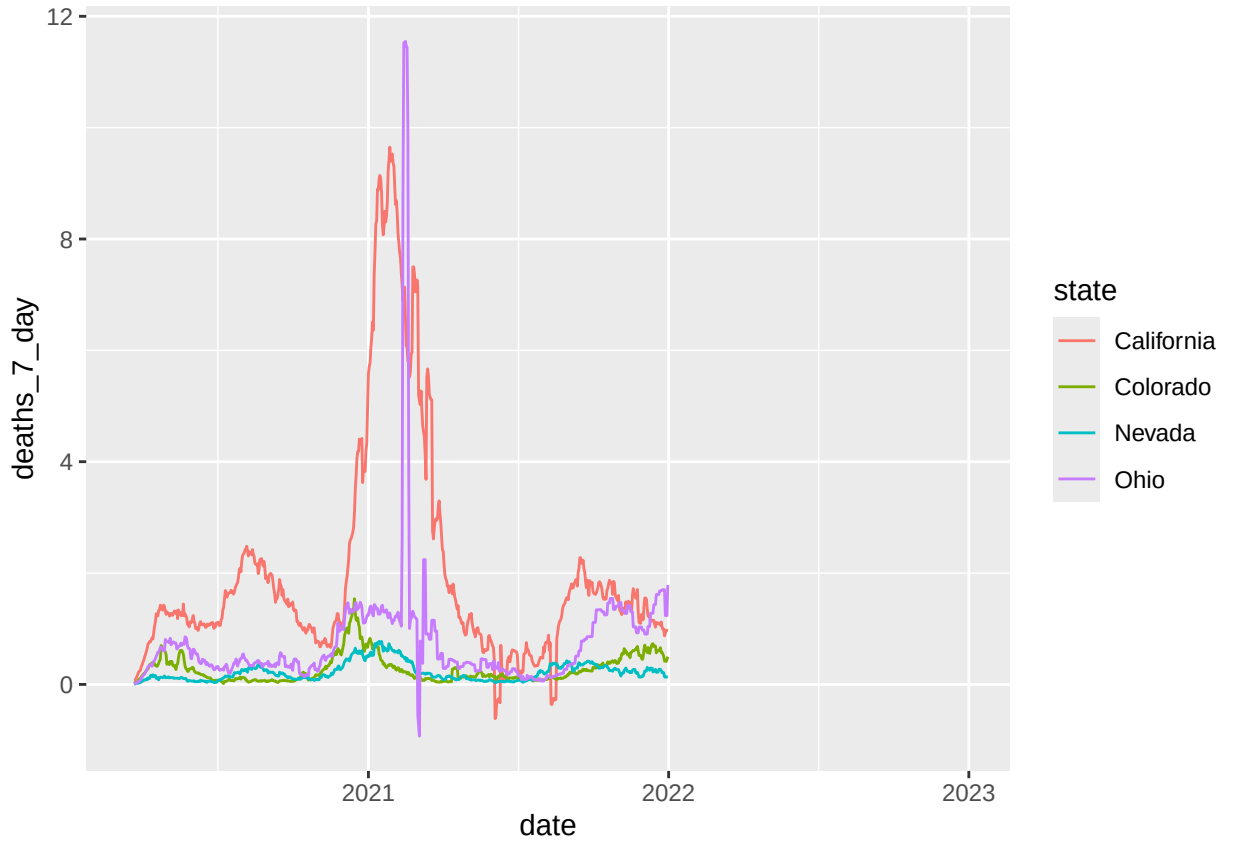
In Ohio, the 7-day average of new deaths per 100,000 population rose to a peak of about 12 in early 2021, before falling to about 2 by the end of 2021. The 7-day average of new cases per 100,000 population rose to about 220 at the end of 2020, before falling to about 40 in the middle of 2021, and then increasing again to about 260 at the end of 2021. There is a slight anomaly in the Ohio data where on 2021-03-02, the number of cumulative total deaths had actually decreased from the previous day, which caused the 7-day average of new deaths to be negative for the following few days.

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.

```
# Prepare a table for each state's 7-day averages of new deaths and cases
# per 100,000
co_covid_compare <- co_covid_delta %>%
  select(date, deaths_7_day, cases_7_day) %>%
  mutate(state = "Colorado")

ca_covid_compare <- ca_covid_delta %>%
```

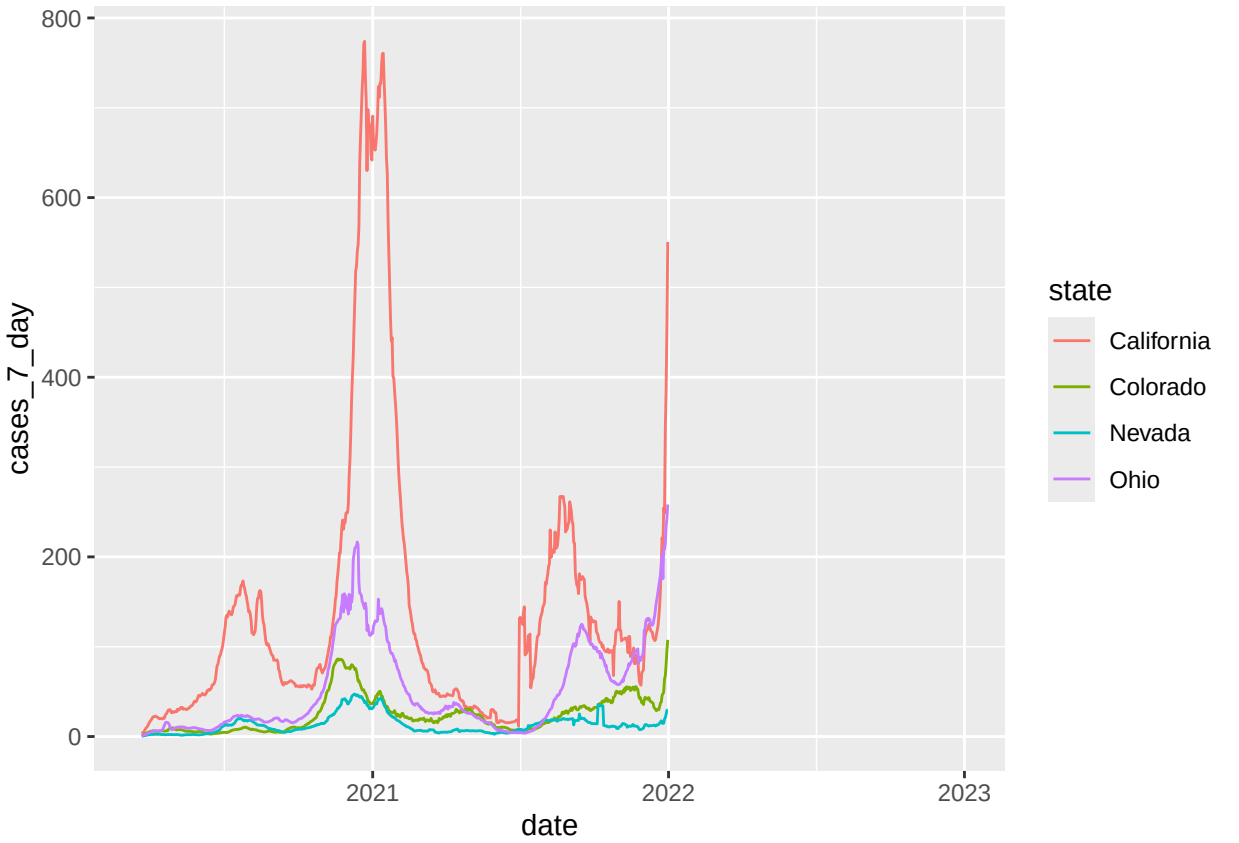
```
  select(date, deaths_7_day, cases_7_day) %>%
  mutate(state = "California")

nv_covid_compare <- nv_covid_delta %>%
  select(date, deaths_7_day, cases_7_day) %>%
  mutate(state = "Nevada")

oh_covid_compare <- oh_covid_delta %>%
  select(date, deaths_7_day, cases_7_day) %>%
  mutate(state = "Ohio")

# Join the four tables by appending one to another
four_states_covid_compare <- co_covid_compare %>%
  rows_append(ca_covid_compare) %>%
  rows_append(nv_covid_compare) %>%
  rows_append(oh_covid_compare)

# Plot each state's 7-day average new deaths per 100,000
four_states_covid_compare %>% ggplot(
  mapping = aes(
    x = date,
    y = deaths_7_day,
    color = state)
  ) +
  geom_line()
```

```
## Warning: Removed 1488 rows containing missing values or values outside the scale range
## (`geom_line()`).
```




```
# Plot each state's 7-day average new cases per 100,000
four_states_covid_compare %>% ggplot(mapping = aes(x = date, y = cases_7_day, color = state)) +
  geom_line()
```

```
## Warning: Removed 1488 rows containing missing values or values outside the scale range
## (`geom_line()`).
```



From the above plots, California generally had the highest 7-day average of new deaths per 100,000 population from March 15, 2020 to December 31, 2021, followed by Ohio, and then Colorado and Nevada at roughly similar levels. As for the 7-day average of new cases per 100,000 population for the same time period, California had the highest amount, followed by Ohio, and then Colorado and Nevada at roughly similar levels. All four states showed similar trends in the 7-day averages of new deaths and cases, which increased to peak levels at around late 2020 to early 2021, and then declined after that. Then, the 7-day average for new cases began increasing again towards the end of 2021.

```
# 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_c
```

Part 3 - Global Comparison

```
## Rows: 289 Columns: 1147
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1145): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## 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.
```

```
csse_global_cases <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_co
```

```
## Rows: 289 Columns: 1147
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1145): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## 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.
```

```
csse_us_deaths <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid
```

```
## Rows: 3342 Columns: 1155
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1149): UID, code3, FIPS, Lat, Long_, Population, 1/22/20, 1/23/20, 1/24...
##
## 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.
```

```
csse_us_cases <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_
```

```
## Rows: 3342 Columns: 1154
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1148): UID, code3, FIPS, Lat, Long_, 1/22/20, 1/23/20, 1/24/20, 1/25/20...
##
## 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.
```

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

```
## Rows: 267 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (6): Country Name, Country Code, Series Name, Series Code, 2020 [YR2020]...
##
## 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.
```

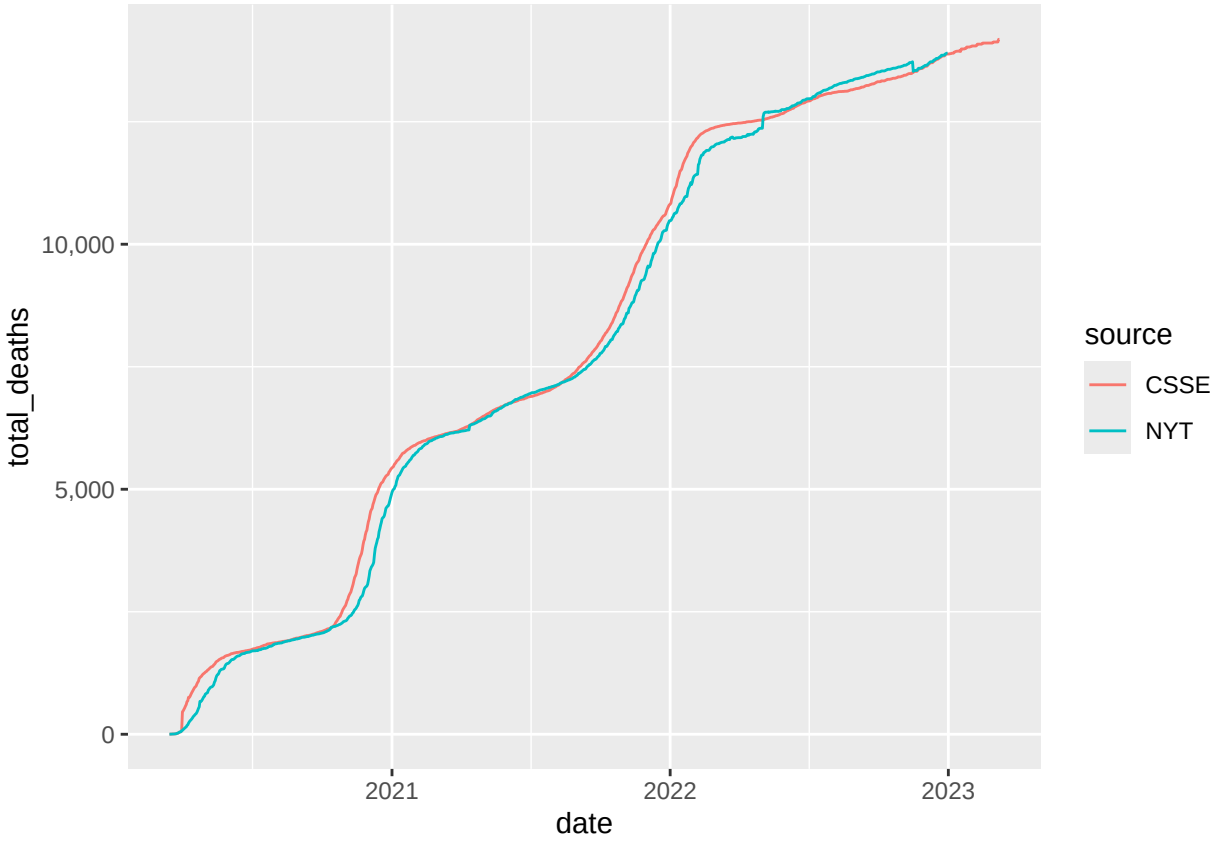
Question 1 Using the state you selected in Part 2 Question 2 compare the daily number of cases and deaths reported from the CSSE and NY Times.

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

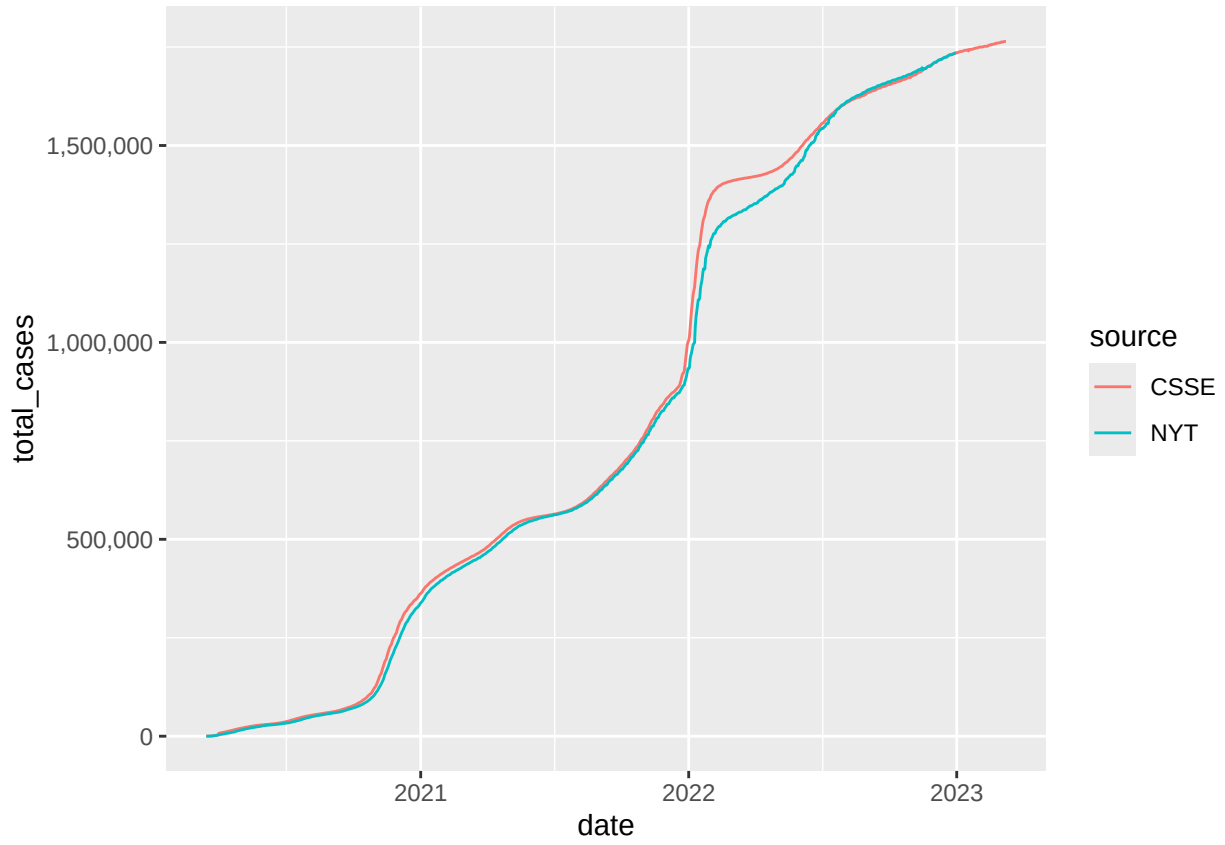
```r
# Use pivot_longer() to shift date values from column names into its own column
# in csse_us_deaths and csse_us_cases

csse_us_deaths_daily <- csse_us_deaths %>% pivot_longer(
  cols = matches("\\d+\\/\\d+/\\d+"),
  names_to = "date",
  values_to = "deaths"
)

csse_us_cases_daily <- csse_us_cases %>% pivot_longer(
  cols = matches("\\d+\\/\\d+/\\d+"),
  names_to = "date",
  values_to = "cases"
)

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

csse_co_daily <- csse_us_deaths_daily %>% full_join(
  csse_us_cases_daily,
  by = join_by(UID, date)
) %>% select(
  FIPS.x,
  Admin2.x,
  Province_State.x,
  date,
  cases,
  deaths
) %>% rename(
  fips = FIPS.x,
  county = Admin2.x,
  state = Province_State.x
) %>% mutate(
  # Parse dates into yyyy-mm-dd
  date = mdy(date)
) %>% filter(
  # Get CSSE US daily deaths and cases for Colorado
  state == "Colorado" & date >= "2020-03-15"
)

head(csse_co_daily)
```

```
## # A tibble: 6 x 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
```

```r
# Finally, create two visualizations
# with one plotting the CSSE and NY Times cases and
# the other plotting the CSEE and NY Times deaths.

# Create new table with CSSE CO total deaths and cases
csse_co_daily_viz <- csse_co_daily %>%
  group_by(date) %>%
  summarize(total_cases = sum(cases), total_deaths = sum(deaths)) %>%
  mutate(source = "CSSE") %>%
  select(date, total_deaths, total_cases, source)

# Create new table with CSSE and NYT data to visualize
csse_nyt_viz <- co_covid_delta %>%
  mutate(source = "NYT") %>%
  select(date, total_deaths, total_cases, source) %>%
  rows_append(csse_co_daily_viz)

# Create line plots comparing total deaths
ggplot(
  csse_nyt_viz,
  mapping = aes(
    x = date,
    y = total_deaths,
    color = source
    )
  ) +
  geom_line() +
  scale_y_continuous(labels = scales::label_comma())
```




```
# Create line plots comparing total cases
ggplot(
  csse_nyt_viz,
  mapping = aes(
    x = date,
    y = total_cases,
    color = source
    )
  ) +
  geom_line() +
  scale_y_continuous(labels = scales::label_comma())
```




```
# 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
```

The CSSE datasets on US deaths and cases were joined together to form a single table, then filtered down to show only data for Colorado, and then finally it was used to create line plots showing Colorado's cumulative total deaths and cases alongside the New York Times data.

The line plots show that the CSSE data shows slightly higher amounts of total deaths and cases than the New York Times data. That said, both data show the same trends and roughly the same volumes of deaths and cases over time.

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 separate.

# CSSE global deaths data

csse_global_deaths_daily <- csse_global_deaths %>% pivot_longer(
  cols = matches("\\d+\\/\\d+\\/\\d+"),
  names_to = "date",
  values_to = "deaths"
) %>% rename(
  country = `Country/Region`
  # Some countries have multiple provinces or states, group by country
  # and get sum of deaths
) %>% group_by(
  date, country
) %>% summarize(
  deaths = sum(deaths)
) %>% mutate(
  date = mdy(date)
)
```

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

```
# Aggregate US deaths data

csse_us_sum_deaths_daily <- csse_us_deaths_daily %>%
  mutate(date = mdy(date)) %>%
  group_by(date) %>%
  summarize(deaths = sum(deaths)) %>%
  mutate(country = "USA")

# Append US deaths to CSSE global deaths

csse_global_deaths_daily <- csse_global_deaths_daily %>%
  rows_append(csse_us_sum_deaths_daily) %>%
  # Extract year to join on global population data
  mutate(year = year(date))

# CSSE global cases data

csse_global_cases_daily <- csse_global_cases %>% pivot_longer(
  cols = matches("\\d+\\/\\d+\\/\\d+"),
  names_to = "date",
  values_to = "cases"
) %>% rename(
  country = `Country/Region`
  # Some countries have multiple provinces or states, group by country
  # and get sum of deaths
) %>% group_by(
```

```
  date, country
) %>% summarize(
  cases = sum(cases)
) %>% mutate(
  date = mdy(date)
)
```

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

```
# Aggregate US cases data

csse_us_sum_cases_daily <- csse_us_cases_daily %>%
  mutate(date = mdy(date)) %>%
  group_by(date) %>%
  summarize(cases = sum(cases)) %>%
  mutate(country = "USA")

# Append US cases to CSSE global cases

csse_global_cases_daily <- csse_global_cases_daily %>%
  rows_append(csse_us_sum_cases_daily) %>%
  # Extract year to join on global population data
  mutate(year = year(date))

# Then, tidy the global population estimates. While tidying your data,
# remember to include columns that you will be able to use
# when joining the COVID-19 data.

global_population_yearly <- global_population_estimates %>% pivot_longer(
  cols = matches("YR\\d{4}"),
  names_to = "year",
  values_to = "population"
) %>%
  # Parse year from Year
  separate_wider_regex(
    cols = year,
    patterns = c(year = "^\\d{4}", ".*")
    ) %>%
  mutate(
    # Convert date from str to int
    year = as.integer(year),
    # Convert population from str to int
    population = as.integer(population)
  )
```

```
## Warning: There were 2 warnings in `mutate()`.
## The first warning was:
## i In argument: `population = as.integer(population)`.
## Caused by warning:
## ! NAs introduced by coercion
## i Run `dplyr::last_dplyr_warnings()` to see the 1 remaining warning.
```

```r
# You will notice that the population estimates data does not include
# every country reported in the CSSE data. When calculating statistics
# per 100,000 people, you will need to filter the CSSE data
# to only include countries that you have population estimates for.

# Inner join CSSE deaths data on population data
csse_global_deaths_per100k <- csse_global_deaths_daily %>%
  inner_join(
    global_population_yearly,
    by = join_by(
      country == `Country Name`,
      year == year
      )
    ) %>%
  # Calculate deaths per 100k population
  mutate(
    deaths_per100k = deaths / population * 100000
  ) %>%
  # Get cumulative total deaths from 2020-03-15 to 2021-12-31
  filter(
    date >= "2020-03-15" & date <= "2021-12-31"
  )

# Get countries with top 10 per 100k deaths as of 2021-12-31
csse_global_deaths_top_10 <- csse_global_deaths_per100k %>%
  filter(date == "2021-12-31") %>%
  # Sort from highest to lowest per 100k deaths
  arrange(desc(deaths_per100k)) %>%
  select(
  country,
  deaths_per100k
) %>%
  head(n = 10)
```

```
## Adding missing grouping variables: ‘date‘
```

```r
csse_global_deaths_top_10
```

```
## # A tibble: 10 x 3
## # Groups:   date [1]
##    date       country                deaths_per100k
##    <date>     <chr>                           <dbl>
##  1 2021-12-31 Peru                             608.
##  2 2021-12-31 Bulgaria                         450.
##  3 2021-12-31 Bosnia and Herzegovina           412.
##  4 2021-12-31 Hungary                          403.
##  5 2021-12-31 Moldova                          393.
##  6 2021-12-31 Montenegro                       388.
##  7 2021-12-31 North Macedonia                  384.
##  8 2021-12-31 Georgia                          372.
##  9 2021-12-31 Croatia                          312.
## 10 2021-12-31 Romania                          307.
```

```r
# Inner join CSSE cases data on population data

csse_global_cases_per100k <- csse_global_cases_daily %>%
  inner_join(
    global_population_yearly,
    by = join_by(
      country == `Country Name`,
      year == year
      )
    ) %>%
  # Calculate cases per 100k population
  mutate(
    cases_per100k = cases / population * 100000
  ) %>%
  # Get cumulative total cases from 2020-03-15 to 2021-12-31
  filter(
    date >= "2020-03-15" & date <= "2021-12-31"
  )

# Get countries with top 10 per 100k cases as of 2021-12-31
csse_global_cases_top_10 <- csse_global_cases_per100k %>%
  filter(date == "2021-12-31") %>%
  # Sort from highest to lowest per 100k deaths
  arrange(desc(cases_per100k)) %>%
  select(
    country,
    cases_per100k
  ) %>%
  head(n = 10)
```

```
## Adding missing grouping variables: ‘date‘
```

```r
csse_global_cases_top_10
```

```
## # A tibble: 10 x 3
## # Groups:   date [1]
##    date       country        cases_per100k
##    <date>     <chr>                  <dbl>
##  1 2021-12-31 Andorra               30831.
##  2 2021-12-31 Montenegro            27381.
##  3 2021-12-31 Georgia               25182.
##  4 2021-12-31 Seychelles            25038.
##  5 2021-12-31 San Marino            24124.
##  6 2021-12-31 Slovenia              22087.
##  7 2021-12-31 Mongolia              20806.
##  8 2021-12-31 United Kingdom        19274.
##  9 2021-12-31 Lithuania             18960.
## 10 2021-12-31 Serbia                18933.
```

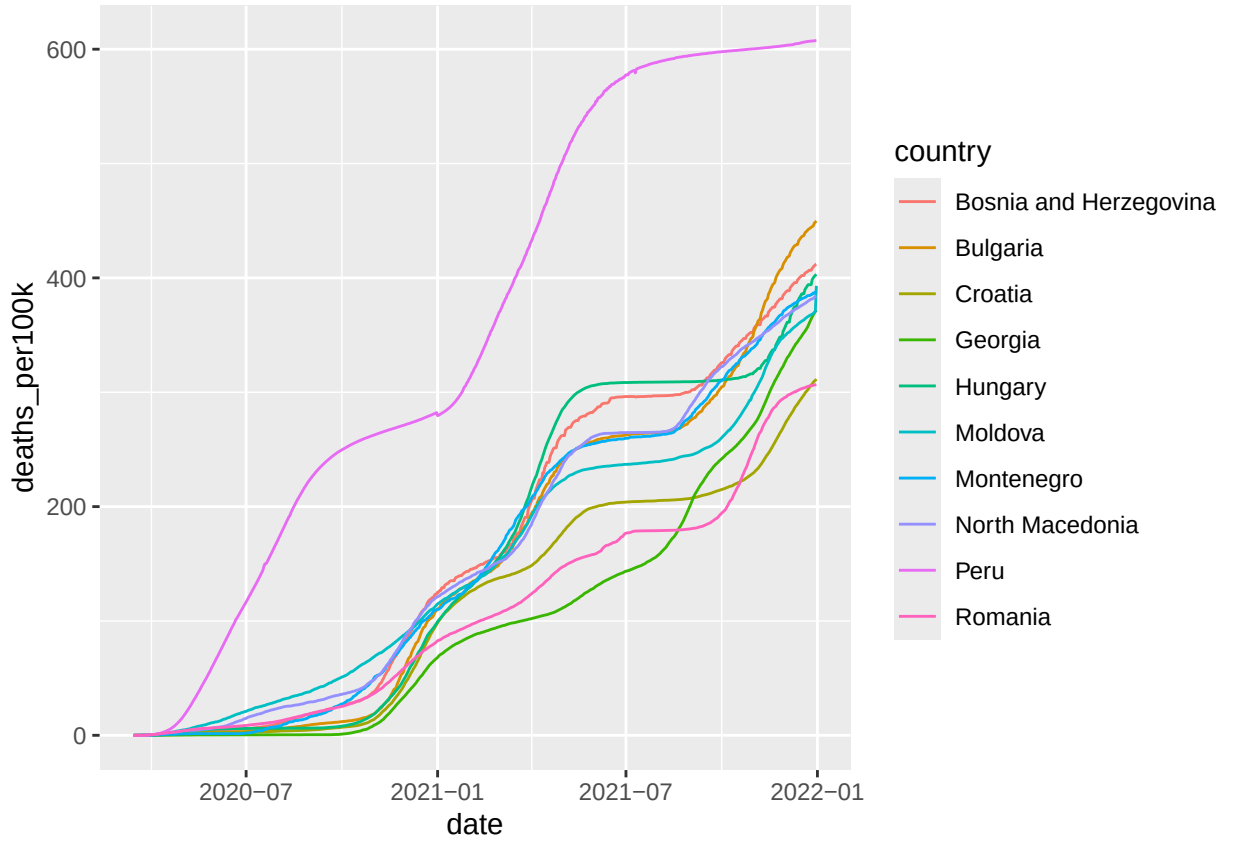
The CSSE global deaths data was tidied, then the CSSE US deaths data was appended to it. Next, the global population estimates data was also tidied. Finally, an inner join was done on the CSSE global deaths and global population data on the fields of country name and year, which subsequently allows for the calculation

of deaths per 100,000 population for dates in 2020 and 2021. Finally, the data was sorted by deaths per 100,000 population in descending order and we see that as of December 31, 2021, the top ten countries in terms of deaths per 100,000 population are Peru, Bulgaria, Bosnia and Herzegovina, Hungary, Moldova, Montenegro, North Macedonia, Georgia, Croatia, and then Romania.

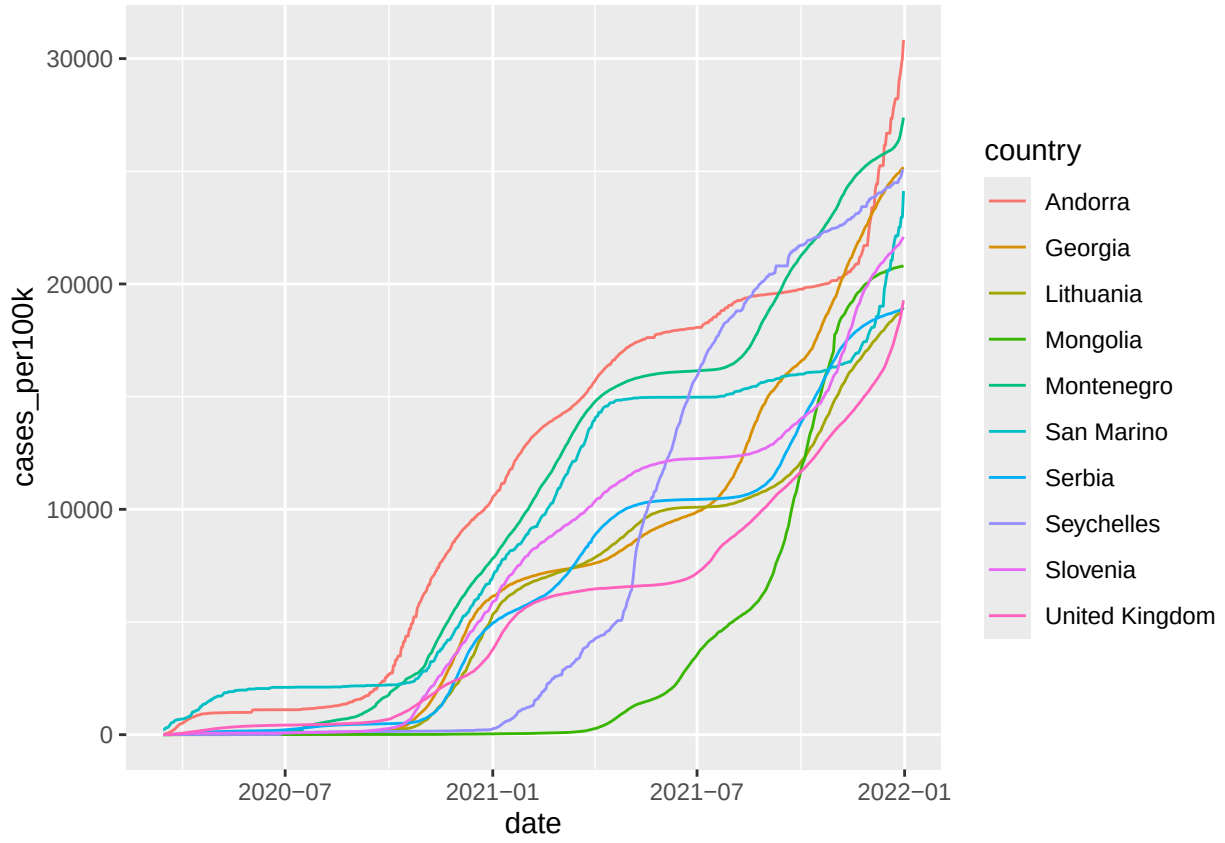
Similarly, the CSSE global cases data was tidied and then the CSSE US cases data was appended to it. Then an inner join was done on the CSSE global cases and global population data on the fields of country name and year, and then the cases per 100,000 population were calculated for each country for dates in 2020 and 2021. After performing the same sort as above, we see that the top ten countries in terms of cases per 100,000 population as of December 31, 2021 are Andorra, Montenegro, Georgia, Seychelles, San Marino, Slovenia, Mongolia, United Kingdom, Lithuania, and then Serbia.

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.

```
# Create line plot showing deaths per 100,000 people for the top 10 countries

# First prepare cases data for plot
csse_global_deaths_per100k %>%
  # Filter by top 10 countries identified in p3q2
  filter(country %in% csse_global_deaths_top_10$country) %>%
  ggplot(mapping = aes(x = date, y = deaths_per100k, color = country)) + geom_line()
```



```r
# Create line plot showing cases per 100,000 people for the top 10 countries

# First prepare cases data for plot
csse_global_cases_per100k %>%
  # Filter by top 10 countries identified in p3q2
  filter(country %in% csse_global_cases_top_10$country) %>%
  ggplot(mapping = aes(x = date, y = cases_per100k, color = country)) + geom_line()
```



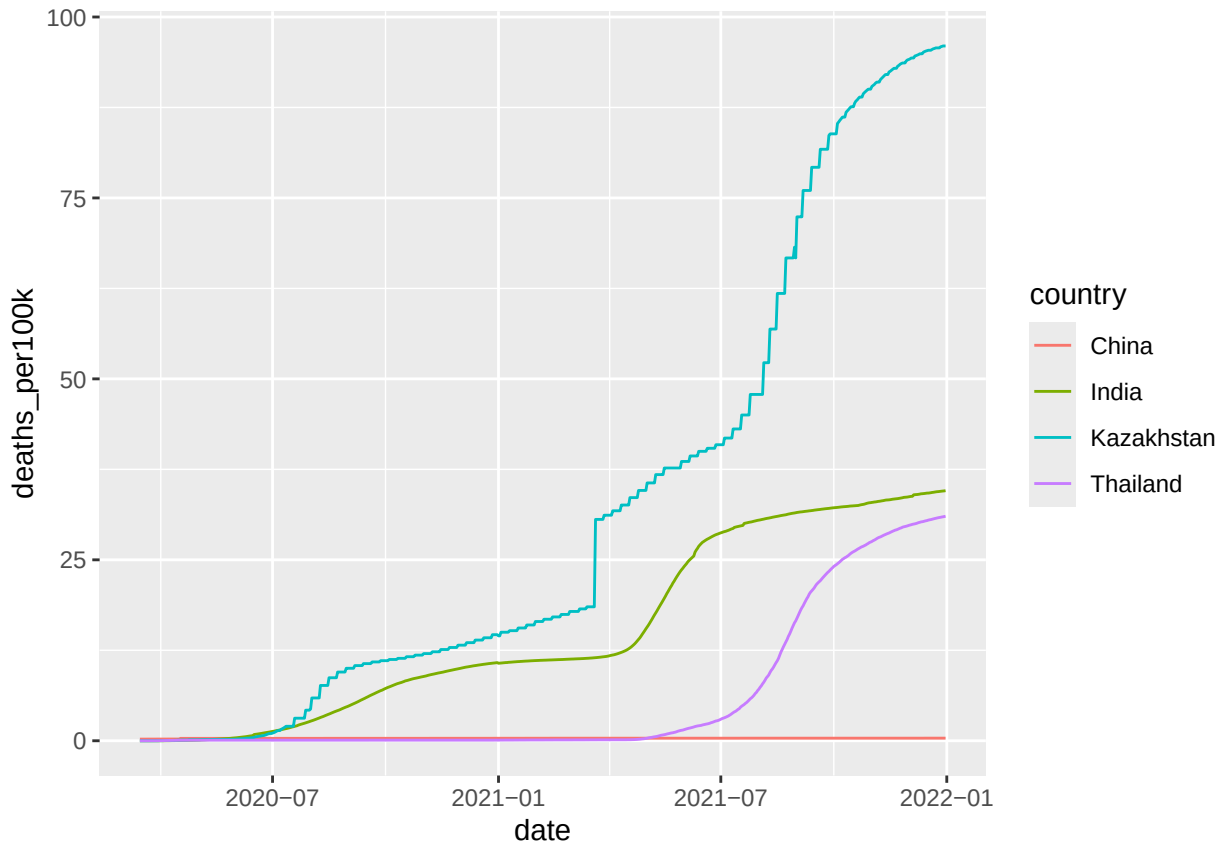
From the above line plots, the ten countries with the highest number of deaths per 100,000 people show similar trends in terms of how the cumulative number of deaths increased over time. Among these countries, the magnitude of deaths per 100,000 people are quite similar across the different points in time from March 15, 2020 to December 31, 2021, except for Peru which showed significantly higher deaths consistently throughout the time period.

As for the ten countries with the highest number of cases per 100,000 people, most show similar trends and magnitude from March 15, 2020 to December 31, 2021, except for Mongolia which showed almost zero cases per 100,000 up until about April 2021, after which the number of cases increased until it reached a similar magnitude to the other countries shown on the plot.

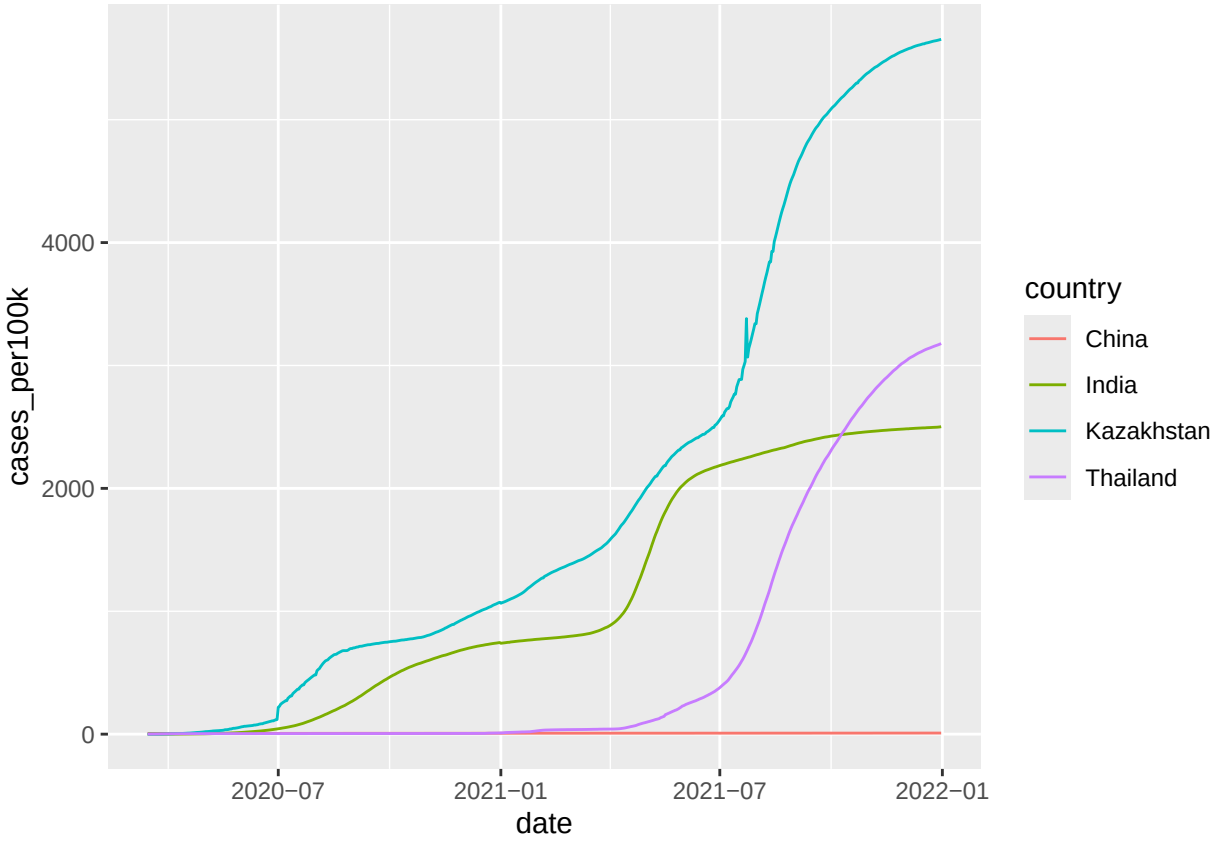
Question 4 Finally, select four countries from one continent and create visualizations for the daily number of confirmed cases per 100,000 and the daily number of deaths per 100,000 people between March 15, 2020, and December 31, 2021.

```r
# From Asia, take China, India, Thailand, Kazakhstan
```

```r
# Create line plot of deaths per 100,000 for each country
csse_global_deaths_per100k %>%
  filter(country %in% c("China", "India", "Thailand", "Kazakhstan")) %>%
  ggplot(mapping = aes(x = date, y = deaths_per100k, color = country)) +
  geom_line()
```




```r
# From Asia, take China, India, Thailand, Kazakhstan

# Create line plot of cases per 100,000 for each country
csse_global_cases_per100k %>%
  filter(country %in% c("China", "India", "Thailand", "Kazakhstan")) %>%
  ggplot(mapping = aes(x = date, y = cases_per100k, color = country)) +
  geom_line()
```



From the above line plots, among the selected four countries from Asia, Kazakhstan reported the highest number of deaths and cases per 100,000 people from March 15, 2020 to December 31, 2021, followed by India, then Thailand, and then China which had reported almost zero deaths and cases per 100,000 people during the same period. Thailand overtook India in terms of number of cases per 100,000 after some time in OCtober 2021.