COVID 19 Analysis

Anna

27/04/23

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, 2022, and 2023 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

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

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

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

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

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

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

```
us_counties_2023 <- read_csv("https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counties
```

```
## Rows: 267009 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 The first task is to combine and tidy the 2020, 2021, 2022, and 2023 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, 2022, 2023 COVID data sets.

us_counties_all <- rbind(us_counties_2020,
                         us_counties_2021,
                         us_counties_2022,
                         us_counties_2023) %>%
    # remove rows for Puerto Rico and rows before March 15, 2022
    filter(state != "Puerto Rico" & date >= "2020-03-15") %>%
```

```
    # group by date and calculate total cases and deaths by date
    group_by(date) %>%
    summarize(
      total_cases = sum(cases),
      total_deaths = sum(deaths)
      )

# view tidy dataset
us_counties_all
```

```
## # A tibble: 1,104 x 3
##    date       total_cases total_deaths
##    <date>           <dbl>        <dbl>
##  1 2020-03-15        3595           68
##  2 2020-03-16        4502           91
##  3 2020-03-17        5901          117
##  4 2020-03-18        8345          162
##  5 2020-03-19       12387          212
##  6 2020-03-20       17998          277
##  7 2020-03-21       24507          359
##  8 2020-03-22       33050          457
##  9 2020-03-23       43474          577
## 10 2020-03-24       53899          783
## # i 1,094 more rows
```

```
# save last date of dataset
max_date <- max(us_counties_all$date)

# filter the last row
last_row <- us_counties_all %>%
    filter(date == max_date)

# total number of cases and deaths for the last date
us_total_cases <- last_row$total_cases
us_total_deaths <- last_row$total_deaths
```

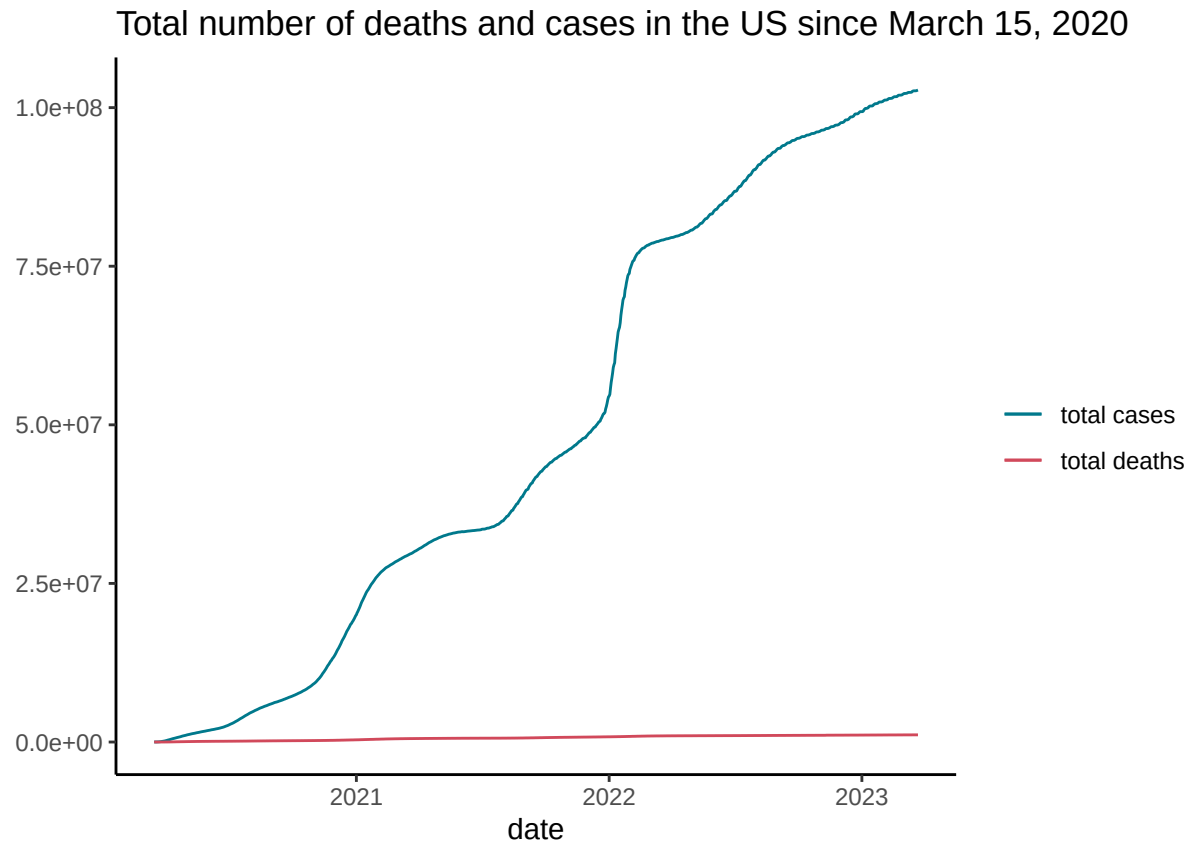
– Communicate your methodology, results, and interpretation here –

I combined four data sets on global COVID-19 cases, deaths, hospitalizations, and recoveries using the row-binding function. After removing any rows before March 15, 2020, and those related to Puerto Rico, I grouped the remaining data by date and calculated the total number of cases and deaths reported on each date.

As of March 23, 2023, the total number of cases was 102,770,844 and the total number of deaths was 1,129,496.

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
# Create a visualization for the total number of US cases and deaths since March 15, 2020.
theme_set(theme_classic())

ggplot(us_counties_all, aes(x = date)) +
    # create 2 line graphs for total cases and total deaths
    geom_line(aes(y = total_cases, col = "total cases")) +
    geom_line(aes(y = total_deaths, col = "total deaths")) +
    # add title and remove y-axis label
    labs(title = "Total number of deaths and cases in the US since March 15, 2020",
         y = "") +
    scale_color_manual(name = "",
                       values = c("total cases"="#00798c", "total deaths"="#d1495b"))  # line color
```



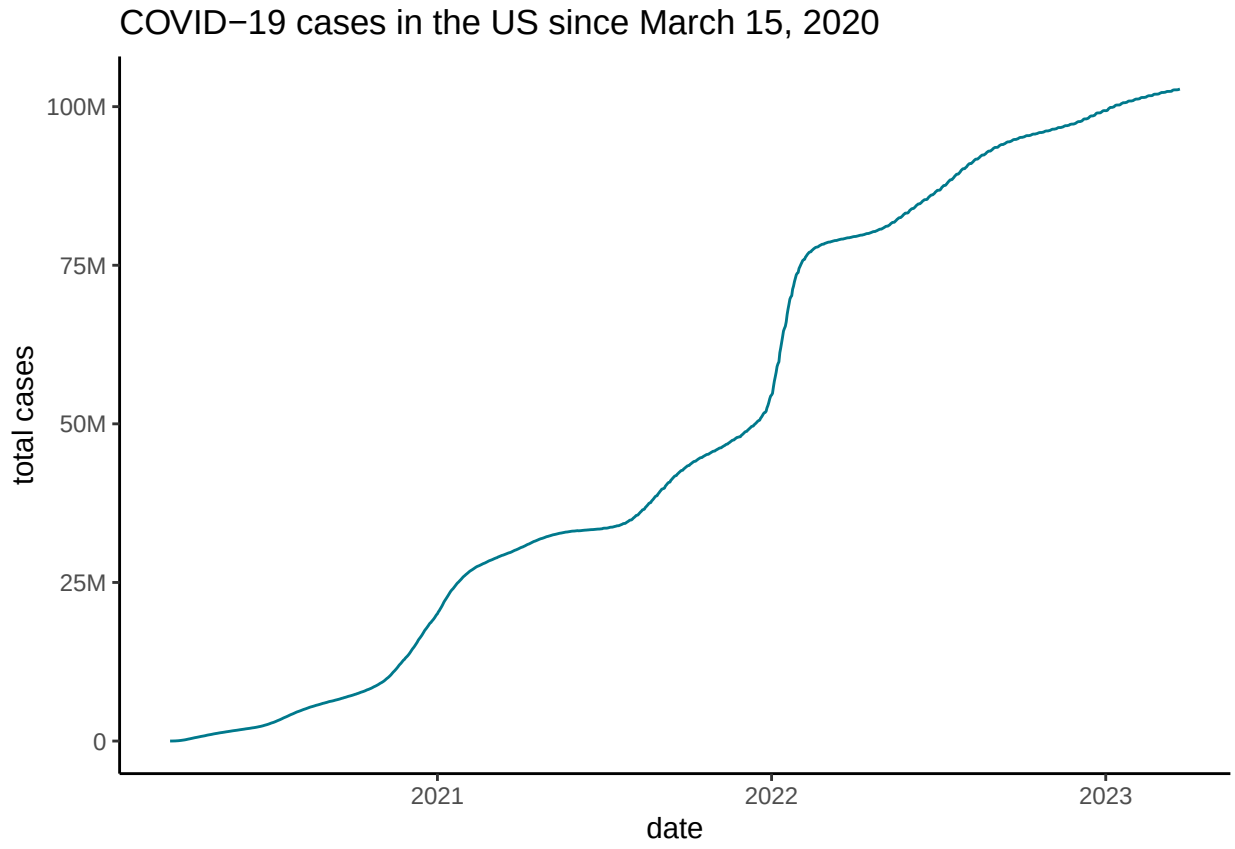
– Communicate your methodology, results, and interpretation here –

Based on the geom_line plot that I created, we can see that there was a rapid growth in cases at the end of 2020 and the beginning of 2022. While changes in the total number of cases can be easily interpreted over time, the significant disparity between the number of cases and deaths makes it difficult to do the same for the total number of deaths. Therefore, a more effective solution could be to create two separate plots for cases and deaths to better visualize the trends in each.

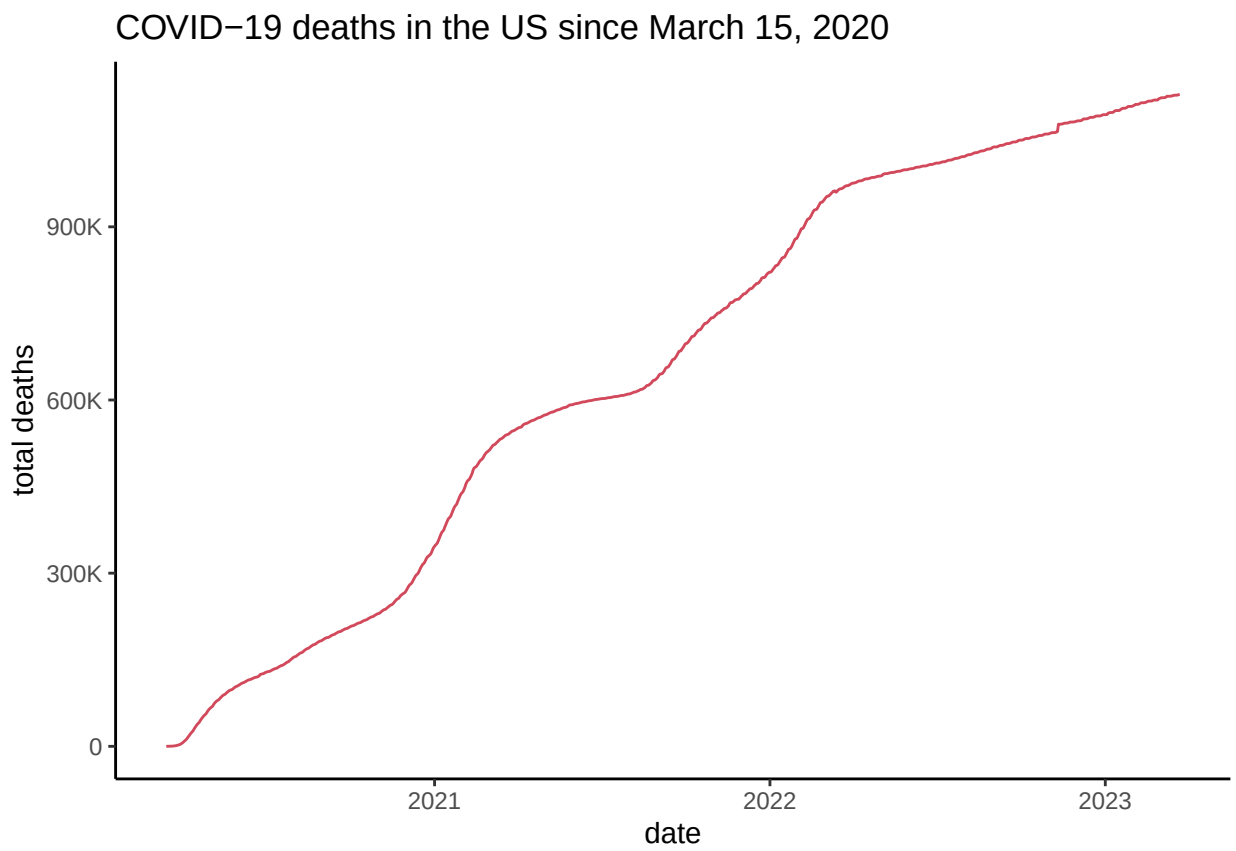
```r
# Create a plot for the total number of US cases since March 15, 2020.

ggplot(us_counties_all, aes(x = date, y = total_cases)) +
    geom_line(color = "#00798c") +
    labs(title = "COVID-19 cases in the US since March 15, 2020",
```

```
        y = "total cases") +
  # format the values to be in millions
    scale_y_continuous(
        labels = scales::label_number_si()
    )
```

```
## Warning: ‘label_number_si()‘ was deprecated in scales 1.2.0.
## i Please use the ‘scale_cut‘ argument of ‘label_number()‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```




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

ggplot(us_counties_all, aes(x = date, y = total_deaths)) +
    geom_line(color = "#d1495b") +
    labs(title = "COVID-19 deaths in the US since March 15, 2020",
         y = "total deaths") +
# format the values to be in thousands
    scale_y_continuous(
        labels = scales::label_number_si()
    )
```



– Communicate your methodology, results, and interpretation here –

Now, we can observe a notable increase in the number of deaths at the start and end of 2021. This rise in deaths is particularly concerning given the ongoing global pandemic and underscores the need for continued vigilance and preventative measures.

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 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 that will have these columns
#  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   >  the number of new deaths since the previous day
#  delta_cases   >  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
#==

us_counties_new <- us_counties_all %>%
    mutate(delta_cases = total_cases - lag(total_cases, 1),
           delta_deaths = total_deaths - lag(total_deaths, 1),
```

```
          delta_cases_7 = rollmean(delta_cases, k = 7, fill = NA, align = 'right'),
          delta_deaths_7 = rollmean(delta_deaths, k = 7, fill = NA, align = "right"))

# print data set
us_counties_new
```

```
## # A tibble: 1,104 x 7
##    date       total_cases total_deaths delta_cases delta_deaths delta_cases_7
##    <date>           <dbl>        <dbl>       <dbl>        <dbl>         <dbl>
##  1 2020-03-15        3595           68          NA           NA            NA
##  2 2020-03-16        4502           91         907           23            NA
##  3 2020-03-17        5901          117        1399           26            NA
##  4 2020-03-18        8345          162        2444           45            NA
##  5 2020-03-19       12387          212        4042           50            NA
##  6 2020-03-20       17998          277        5611           65            NA
##  7 2020-03-21       24507          359        6509           82            NA
##  8 2020-03-22       33050          457        8543           98         4208.
##  9 2020-03-23       43474          577       10424          120         5567.
## 10 2020-03-24       53899          783       10425          206         6857.
## # i 1,094 more rows
## # i 1 more variable: delta_deaths_7 <dbl>
```

```
## find the days that saw the largest number of new cases and deaths
# find the row where the maximum new cases occurred
max_new_cases <- us_counties_new %>%
    filter(delta_cases == max(us_counties_new$delta_cases, na.rm = TRUE))
# choose date from the row
max_new_cases_date <- max_new_cases$date

# find the row where the maximum new deaths occurred
max_new_deaths <- us_counties_new %>%
    filter(delta_deaths == max(us_counties_new$delta_deaths, na.rm = TRUE))
# choose date from the row
max_new_deaths_date <- max_new_deaths$date
```

– Communicate your methodology, results, and interpretation here –

To subtract the total number of cases and deaths on the previous day from each particular day, I utilized the lag() function, which shifts the time base back by one day. Similarly, I used a similar approach to calculate delta_deaths. To calculate the average numbers of cases and deaths over a seven-day period, I applied the rollmean() function from the zoo package, which calculates a rolling average.

In order to provide more insight into the severity of the situation, it is worth noting that the dates when the maximum number of new cases and deaths occurred were January 10, 2022 and November 11, 2022, respectively.

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

```
# calculate population estimates by year

population_by_year <- us_population_estimates %>%
```

```r
  # group by year
  group_by(Year) %>%
  # get sum of estimated populations
  summarize(population = sum(Estimate))

population_by_year
```

```
## # A tibble: 2 x 2
##    Year population
##   <dbl>      <dbl>
## 1  2020  331501080
## 2  2021  331893745
```

Our data set covers the years 2020-2023, but we only have population data for 2020 and 2021. To address this gap, I plan to use a new data set from `census.gov` that contains monthly population estimates from April 2020 until December 2023. This data set is called "Monthly Population Estimates for the United States: April 1, 2020 to December 1, 2023".

The reason for switching to this new data set is because the population increased in 2022, and it would not be appropriate to use the same population numbers for 2022 and 2023. By using the new data set, we can ensure that our analysis is accurate and up-to-date.

```r
# read excel file from Internet using openxlsx library
us_population_estimates_2 <- read.xlsx("https://www2.census.gov/programs-surveys/popest/tables/2020-202

# calculate estimated population from 2020 until 2023
us_population <- us_population_estimates_2 %>%
  filter(Year.and.Month == ".July 1") %>%    # as for a middle of the year
  # choose only year and population
  transmute(year = c(2020, 2021, 2022, 2023),
         population = Resident.Population)

# save variables for population for each year
population_2020 <- us_population$population[1]
population_2021 <- us_population$population[2]
population_2022 <- us_population$population[3]
population_2023 <- us_population$population[4]

covid_us_per_100k <- us_counties_new %>%
  # mutate columns by dividing each statistics by population
  # of appropriate year and multiply by 100,000
  mutate(
    total_cases = case_when(
      grepl("2020", date) ~ round((total_cases / population_2020) * 100000, 2),
      grepl("2021", date) ~ round((total_cases / population_2021) * 100000, 2),
      grepl("2022", date) ~ round((total_cases / population_2022) * 100000, 2),
      grepl("2023", date) ~ round((total_cases / population_2023) * 100000, 2),
      ),
    total_deaths = case_when(
      grepl("2020", date) ~ round((total_deaths / population_2020) * 100000, 4),
      grepl("2021", date) ~ round((total_deaths / population_2021) * 100000, 4),
      grepl("2022", date) ~ round((total_deaths / population_2022) * 100000, 4),
      grepl("2023", date) ~ round((total_deaths / population_2023) * 100000, 4),
```

```
    ),
  delta_cases = case_when(
    grepl("2020", date) ~ round((delta_cases / population_2020) * 100000, 2),
    grepl("2021", date) ~ round((delta_cases / population_2021) * 100000, 2),
    grepl("2022", date) ~ round((delta_cases / population_2022) * 100000, 2),
    grepl("2023", date) ~ round((delta_cases / population_2023) * 100000, 2),
    ),
  delta_deaths = case_when(
    grepl("2020", date) ~ round((delta_deaths / population_2020) * 100000, 4),
    grepl("2021", date) ~ round((delta_deaths / population_2021) * 100000, 4),
    grepl("2022", date) ~ round((delta_deaths / population_2022) * 100000, 4),
    grepl("2023", date) ~ round((delta_deaths / population_2023) * 100000, 4),
    ),
  delta_cases_7 = case_when(
    grepl("2020", date) ~ round((delta_cases_7 / population_2020) * 100000, 2),
    grepl("2021", date) ~ round((delta_cases_7 / population_2021) * 100000, 2),
    grepl("2022", date) ~ round((delta_cases_7 / population_2022) * 100000, 2),
    grepl("2023", date) ~ round((delta_cases_7 / population_2023) * 100000, 2),
    ),
  delta_deaths_7 = case_when(
    grepl("2020", date) ~ round((delta_deaths_7 / population_2020) * 100000, 4),
    grepl("2021", date) ~ round((delta_deaths_7 / population_2021) * 100000, 4),
    grepl("2022", date) ~ round((delta_deaths_7 / population_2022) * 100000, 4),
    grepl("2023", date) ~ round((delta_deaths_7 / population_2023) * 100000, 4),
    )
  )

covid_us_per_100k
```

```
## # A tibble: 1,104 x 7
##    date       total_cases total_deaths delta_cases delta_deaths delta_cases_7
##    <date>           <dbl>        <dbl>       <dbl>        <dbl>         <dbl>
##  1 2020-03-15        1.08       0.0205       NA         NA              NA
##  2 2020-03-16        1.36       0.0275        0.27       0.0069         NA
##  3 2020-03-17        1.78       0.0353        0.42       0.0078         NA
##  4 2020-03-18        2.52       0.0489        0.74       0.0136         NA
##  5 2020-03-19        3.74       0.0639        1.22       0.0151         NA
##  6 2020-03-20        5.43       0.0836        1.69       0.0196         NA
##  7 2020-03-21        7.39       0.108         1.96       0.0247         NA
##  8 2020-03-22        9.97       0.138         2.58       0.0296          1.27
##  9 2020-03-23       13.1        0.174         3.14       0.0362          1.68
## 10 2020-03-24       16.3        0.236         3.14       0.0621          2.07
## # i 1,094 more rows
## # i 1 more variable: delta_deaths_7 <dbl>
```

– Communicate your methodology, results, and interpretation here –

Initially, I obtained monthly population estimates from the census.gov dataset, covering April 2020 through December 2023. Since our main dataset spans four years, I used this population data to estimate the population for each year.

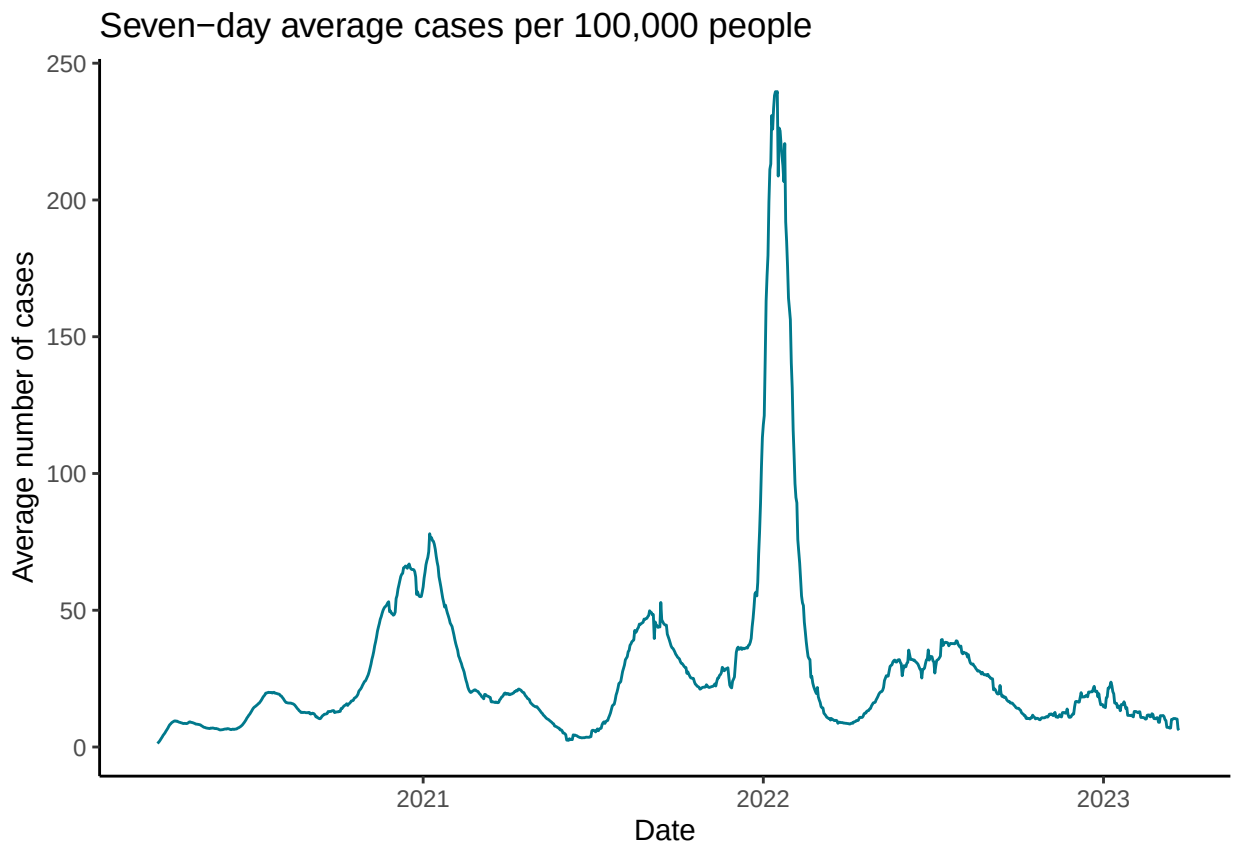
Next, I divided the number of cases and deaths for each day by the estimated population for that year, multiplied the result by 100,000, and rounded to obtain the corresponding numbers of cases and deaths per 100,000 people. To achieve this, I used grepl to identify the column for each year, and then applied the

aforementioned calculations to each column. While I recognize that using a function would be more efficient, I have yet to implement one.

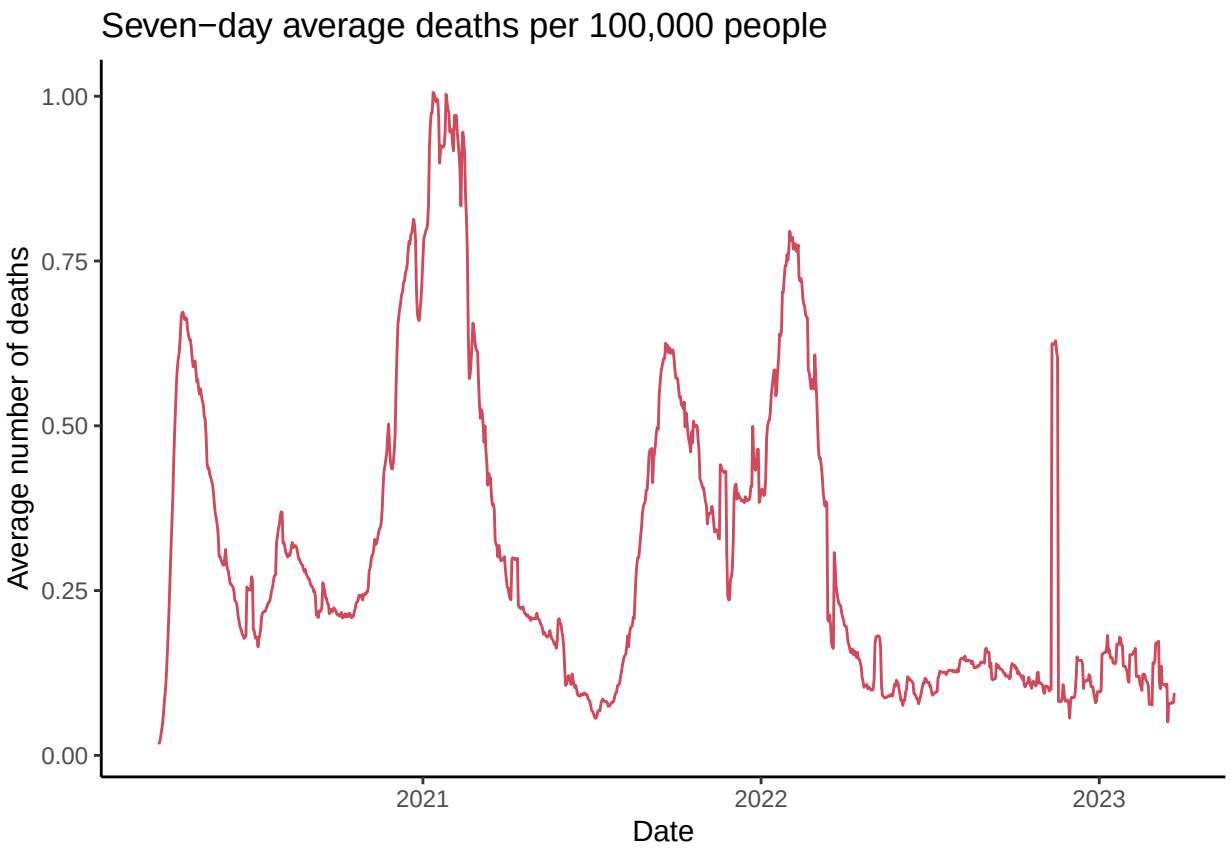
The output of this process is a table that displays the numbers of new cases and deaths per 100,000 people for each day, as well as the seven-day moving averages of these values.

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

```
# Visualization of seven-day average cases per 100,000 people.

covid_us_per_100k %>%
  ggplot(aes(x = date, y = delta_cases_7)) +
  geom_line(color = "#00798c", na.rm = TRUE) +
  ggtitle("Seven-day average cases per 100,000 people") +
  labs(x = "Date", y = "Average number of cases")
```




```
# Visualization of seven-day average deaths per 100,000 people.

covid_us_per_100k %>%
  ggplot(aes(x = date, y = delta_deaths_7)) +
  geom_line(color = "#d1495b", na.rm = TRUE) +
  ggtitle("Seven-day average deaths per 100,000 people") +
  labs(x = "Date", y = "Average number of deaths")
```



– Communicate your methodology, results, and interpretation here –

I used geom_line to compare seven-day averages of cases and deaths. We can see that there are multiple ups and downs in cases but there is one huge spike in the beginning of 2022. In deaths the highest rolling average was in the end of 2020 and the beginning of 2021. As well we can see a spike of deaths in the beginning of 2022 but it's not as distinguishing as in cases. So we can conclude that in the beginning of 2022 COVID variant was very contagious but not as lethal as before.

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.

```
us_states_20_21 <- rbind(us_counties_2020,
                         us_counties_2021) %>%
    select(state,
           date,
           deaths,
           cases) %>%
    # filter out rows for Puerto Rico and dates before March 15, 2020
    filter(state != "Puerto Rico"
```

```
          & date >= "2020-03-15") %>%
    # group by state and date, and calculate total cases and deaths
    group_by(state, date) %>%
    summarise(total_deaths = sum(deaths),
              total_cases = sum(cases)) %>%
    # sort by total_deaths
    arrange(desc(total_deaths))

us_states_total <- us_states_20_21 %>%
    select(state,
           date,
           total_deaths,
           total_cases) %>%
    filter(total_deaths == max(total_deaths),
           total_cases == max(total_cases)) %>%
    arrange(desc(total_deaths))

# drop duplicates
us_states_total <- us_states_total[!duplicated(us_states_total$state), ]

us_states_total %>% head(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 Pennsylvania 2021-12-31        36705     2036424
##  6 Illinois     2021-12-30        31017     2154058
##  7 Georgia      2021-12-31        30283     1798497
##  8 Ohio         2021-12-31        29447     2016095
##  9 New Jersey   2021-12-31        29037     1564253
## 10 Michigan     2021-12-29        28984     1706355
```

– Communicate your methodology, results, and interpretation here –

In the first part, I was analyzing data from 2020 until 2023, but because I don't have state-wise and county-wise data for 2022 and 2023, here I will be looking at data only between March 15, 2020, and December 31, 2021, as it was suggested in the assignment.

In order to determine top 10 states in terms of total deaths and cases between March 15, 2020, and December 31, 2021, I:

- combined necessary data sets
- filtered out Puerto Rico and dates before March 15, 2020
- grouped dataset by state and date
- calculated the total numbers of cases and deaths per each state
- sorted the data set from highest to lowest number of deaths
- then, filtered data set to include only rows with maximum number of cases and deaths
- dropped duplicated rows (that have the same number of cases and deaths but different dates)
- printed the top 10 states

10 states with the highest number of deaths and cases are: California, Texas, Florida, New York, Pennsylvania, Illinois, Georgia, Ohio, New Jersey, Michigan. Most of these states have high population, so it will be interesting to investigate further, to find out if the top states in terms of deaths and cases per 100,000 people are similar to the ones found here.

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?

```
# calculate estimated population from 2020 until 2021
population_by_state <- us_population_estimates %>%
  # group by state
  group_by(STNAME, Year) %>%
  # get sum of estimated population in 2021
  summarize(population = sum(Estimate, na.rm = TRUE)) %>%
  filter(Year == 2021)
```

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

```
# join the population estimates with the cases and death statistics
us_states_population <- left_join(us_states_total, population_by_state, by = join_by(state == STNAME))


# calculate each statistics per 100 thousand people
us_states_per_100k <- us_states_population %>%
  # mutate columns by dividing each statistics by population times 100000
  mutate(
    deaths_per_100k = round((total_deaths / population) * 100000, 0),
    cases_per_100k = round((total_cases / population) * 100000, 0)
  ) %>%
  select(state, date, deaths_per_100k, cases_per_100k) %>%
  arrange(desc(cases_per_100k))

us_states_per_100k
```

```
## # A tibble: 54 x 4
## # Groups:   state [54]
##    state        date       deaths_per_100k cases_per_100k
##    <chr>        <date>               <dbl>          <dbl>
##  1 North Dakota 2021-12-31             265          22482
##  2 Alaska       2021-12-29             130          21310
##  3 Rhode Island 2021-12-30             280          21093
##  4 South Dakota 2021-12-30             278          20014
##  5 Wyoming      2021-12-30             264          19979
##  6 Tennessee    2021-12-30             296          19783
##  7 Kentucky     2021-12-31             269          19173
##  8 Florida      2021-12-31             287          19128
##  9 Utah         2021-12-30             113          19088
## 10 Wisconsin    2021-12-31             190          19008
## # i 44 more rows
```

– Communicate your methodology, results, and interpretation here –

To determine the top 10 states in terms of deaths and cases per 100,000 people between March 15, 2020, and December 31, 2021, I transformed the population estimates to include total population by state in 2021. I used left_join() to join the population estimates with the cases and death statistics using the state name as a key. Then, I calculated deaths and cases per 100,000 people. And, finally, sorted by number of cases.

The result table is different from the one produced in Question 1. Even though we could think that highly populated places would have higher rates of COVID cases, here we see that it's not the case. There must other factors that affect the rates of deaths and cases.

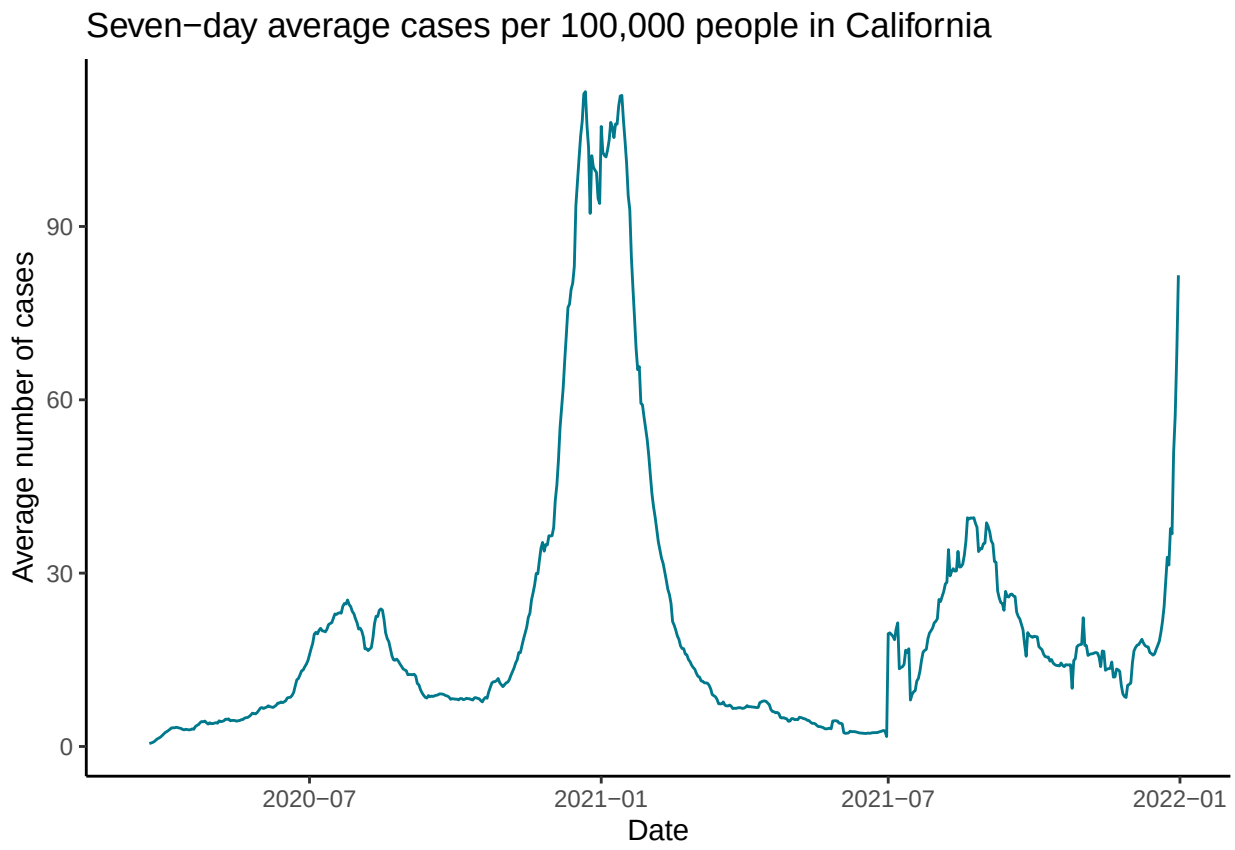
Question 3 Now, select a state and calculate the seven-day averages for new cases and deaths per 100,000 people. Once you have calculated the averages, create a visualization using ggplot2 to represent the data.

```
# count total deaths and cases per day in California
total_by_date_ca <- rbind(us_counties_2020,
                          us_counties_2021) %>%
    # filter California and dates after March 15, 2020
    filter(state == "California"
           & date >= "2020-03-15") %>%
    # group by date and calculate total cases and deaths
    group_by(date) %>%
    summarise(total_deaths = sum(deaths),
              total_cases = sum(cases))

# find population by year in California
population_ca <- us_population_estimates %>%
  filter(STNAME == "California") %>%
  group_by(Year) %>%
  summarize(population = sum(Estimate))

# calculate averages for California
ca_covid_stats <- total_by_date_ca %>%
  # choose necessary columns with transmute
  transmute(
    state = "California",
    date = date,
    total_deaths = total_deaths,
    total_cases = total_cases,
    # find which year from date with grepl and select population estimate from population_ca
    population = case_when(
      grepl("2020", date) ~ population_ca$population[1],
      grepl("2021", date) ~ population_ca$population[2]),
    # calculate deaths and cases per 100000 people
    deaths_per_100k = round(total_deaths / population * 100000, 4),
    cases_per_100k = round(total_cases / population * 100000, 2),
    # calculate rolling averages
    delta_deaths_7 = rollmean(deaths_per_100k - lag(deaths_per_100k, 1), k = 7, fill = NA, align = "rig
    delta_cases_7 = rollmean(cases_per_100k - lag(cases_per_100k, 1), k = 7, fill = NA, align = 'right'

# print result
ca_covid_stats
```

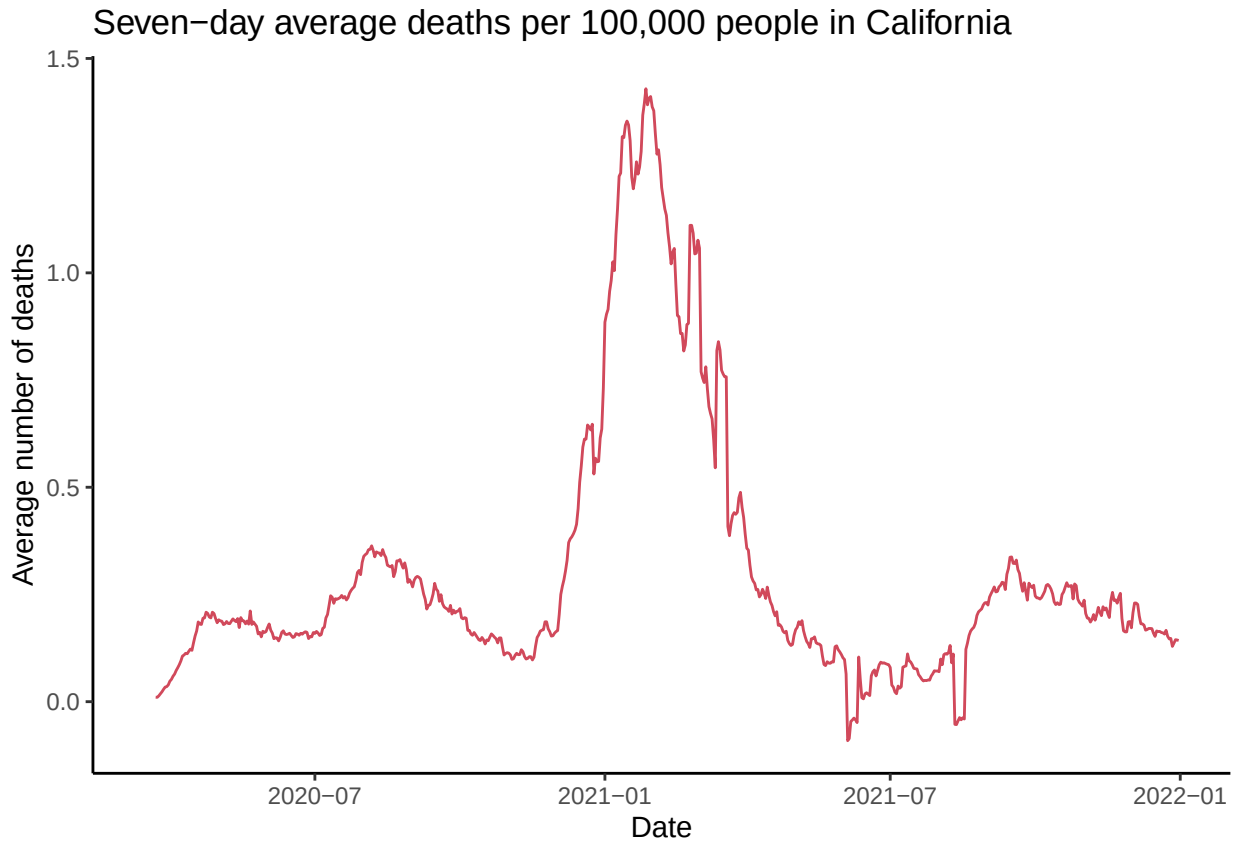
```
## # A tibble: 657 x 9
```

```
##    state      date       total_deaths total_cases population deaths_per_100k
##    <chr>      <date>            <dbl>       <dbl>      <dbl>           <dbl>
##  1 California 2020-03-15            6         478   39499738          0.0152
##  2 California 2020-03-16           11         588   39499738          0.0278
##  3 California 2020-03-17           14         732   39499738          0.0354
##  4 California 2020-03-18           17         893   39499738          0.043
##  5 California 2020-03-19           19        1067   39499738          0.0481
##  6 California 2020-03-20           24        1283   39499738          0.0608
##  7 California 2020-03-21           28        1544   39499738          0.0709
##  8 California 2020-03-22           35        1851   39499738          0.0886
##  9 California 2020-03-23           39        2240   39499738          0.0987
## 10 California 2020-03-24           52        2644   39499738          0.132
## # i 647 more rows
## # i 3 more variables: cases_per_100k <dbl>, delta_deaths_7 <dbl>,
## #   delta_cases_7 <dbl>
```

```r
# Visualization of seven-day average cases per 100,000 people in California.

ca_covid_stats %>%
  ggplot(aes(x = date, y = delta_cases_7)) +
  geom_line(color = "#00798c", na.rm = TRUE) +
  ggtitle("Seven-day average cases per 100,000 people in California") +
  labs(x = "Date", y = "Average number of cases")
```



```r
# Visualization of seven-day average deaths per 100,000 people in California.

ca_covid_stats %>%
  ggplot(aes(x = date, y = delta_deaths_7)) +
  geom_line(color = "#d1495b", na.rm = TRUE) +
  ggtitle("Seven-day average deaths per 100,000 people in California") +
  labs(x = "Date", y = "Average number of deaths")
```



– Communicate your methodology, results, and interpretation here –

To calculate the seven-day averages for new cases and deaths per 100,000 people in California, I counted total deaths and cases per day in California, then I found population in 2020 and 2021. And finally, I put everything in one table, calculating deaths and cases per 100,000 people and the seven-day averages for the new cases and deaths per 100,000 people using rollmean(). Visualization was created using ggplot, there are two different plots for cases and deaths, as they have different scales.

The visualizations reveal a noticeable surge in deaths and cases at the start of 2021, while by the end of that same year, the cases remained high but the deaths did not. This aligns with the conclusion drawn in Part 1, which stated that the COVID variant at the end of 2021 was highly transmissible but less fatal than its earlier counterparts.

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

```r
# find population by county in California
population_county_ca <- us_population_estimates %>%
```

```r
  filter(STNAME == "California") %>%
  group_by(CTYNAME) %>%
  summarize(population = sum(Estimate) / 2) %>%    # divide by 2 to count average population
  transmute(
    # remove last word ("COUNTY") in county column
    county = word(CTYNAME, 1   , -2), # starts with 1st word, ends with -2 (everything except the last w
    population = population)

# calculate deaths and cases per county
counties_ca <- rbind(us_counties_2020, us_counties_2021) %>%
    # filter California and the last day in 2021
    filter(state == "California"
           & date == "2021-12-31") %>%
    # group by county and date, and calculate total cases and deaths
    group_by(county, fips) %>%
    select(state, county, fips, deaths, cases)


# combine 2 data sets
combined_ca <- left_join(population_county_ca, counties_ca, by = join_by(county == county))

# calculate deaths and cases per 100,000 people
counties_combined_ca <- combined_ca %>%
  group_by(county, fips) %>%
  mutate(deaths_per_100k = round(deaths / population * 100000, 1),
         cases_per_100k = round(cases / population * 100000, 0)
  )

# create one data frame sorted by number of deaths
counties_ca_deaths <- counties_combined_ca %>%
    arrange(desc(deaths_per_100k))

# create another data frame sorted by number of cases
counties_ca_cases <- counties_combined_ca %>%
    arrange(desc(cases_per_100k))

# print first 5 rows of both data frames
head(counties_ca_deaths, 5)
```

```
## # A tibble: 5 x 8
## # Groups:   county, fips [5]
##   county     population state fips  deaths  cases deaths_per_100k cases_per_100k
##   <chr>           <dbl> <chr> <chr>  <dbl>  <dbl>           <dbl>          <dbl>
## 1 Imperial      179670  Cali~ 06025    799 4.21e4             445.          23418
## 2 Los Angel~   9909354. Cali~ 06037  27637 1.70e6             279.          17128
## 3 San Berna~   2188725  Cali~ 06071   6051 3.99e5             276.          18231
## 4 Shasta        182016  Cali~ 06089    491 2.70e4             270.          14851
## 5 Stanislaus    552854. Cali~ 06099   1473 9.51e4             266.          17208
```

```r
head(counties_ca_cases, 5)
```

```
## # A tibble: 5 x 8
## # Groups:   county, fips [5]
```

```
##   county     population state fips  deaths  cases deaths_per_100k cases_per_100k
##   <chr>           <dbl> <chr> <chr>  <dbl>  <dbl>           <dbl>          <dbl>
## 1 Lassen          32939 Cali~ 06035     61   8090            185.          24561
## 2 Kings          153035 Cali~ 06031    387  36559            253.          23889
## 3 Imperial       179670 Cali~ 06025    799  42076            445.          23418
## 4 Tulare         475395 Cali~ 06107   1160  89686            244           18866
## 5 San Berna~    2188725 Cali~ 06071   6051 399021            276.          18231
```

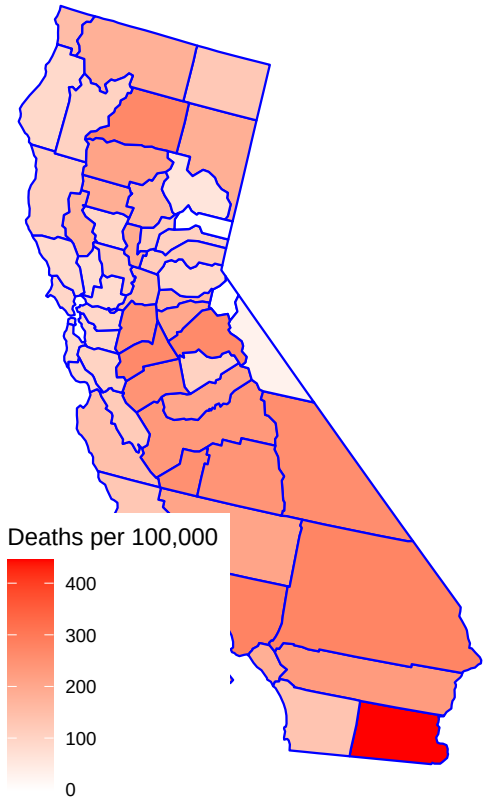
– Communicate your methodology, results, and interpretation here –

Initially, I obtained the population data for each county in California. Then, I created a dataframe that contains information specific to California, starting from March 15, 2020. The data was grouped by county, and the number of deaths and cases for each county were calculated. After combining these 2 data sets, I calculated deaths and cases per 100,00 people. And, finally, I created 2 datasets: one with counties sorted by number of deaths, and the other one sorted by number of cases.

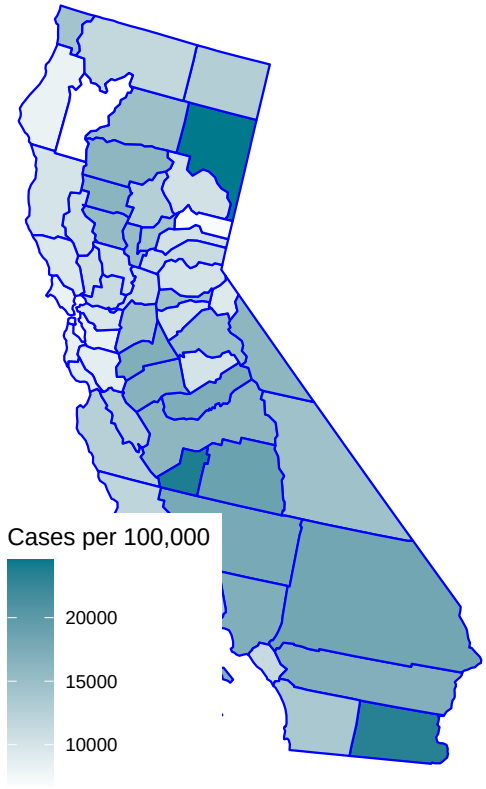
According to the results, top 5 counties with the highest deaths rate are Imperial, Los Angeles, San Bernardino, Shasta and Stanislaus. On the other hand, top 5 counties with the highest number of cases are: Lassen, Kings, Imperial, Tulare and San Bernardino. Interestingly, only two counties, namely Imperial and San Bernardino, are present in both lists.

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

```
plot_usmap(regions = "county", include="CA", data = counties_combined_ca, values = "deaths_per_100k", c
  scale_fill_continuous(low = "white", high = "red", name = "Deaths per 100,000")
```



```r
plot_usmap(regions = "county", include="CA", data = counties_combined_ca, values = "cases_per_100k", col
  scale_fill_continuous(low = "white", high = "#00798c", name = "Cases per 100,000")
```



– Communicate your methodology, results, and interpretation here –

I used `plot_usmap` to visualize deaths and cases rates in different counties of California.

These maps offer a great way to visualize the geographical distribution of COVID-19 cases and deaths in different parts of the state. However, the correlation between areas with the highest cases and those with the highest deaths is not always consistent. This may be due to cases being influenced by population factors such as proximity and travel, while deaths are affected by individual factors such as age, health status, and quality of medical care.

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
# find population by year in 3 states
population_3st <- us_population_estimates %>%
  # group by Year and STNAME
  group_by(Year, STNAME) %>%
  # get sum of estimated populations, for 2020 and 2021
  summarise(population = sum(Estimate, na.rm = TRUE)) %>%
  filter(STNAME == "New York" | STNAME == "Arizona" | STNAME == "Nevada") %>%
  filter(Year == 2021)

# count total deaths and cases per day in 3 states
```

```r
total_by_date_3st <- rbind(us_counties_2020, us_counties_2021) %>%
    # filter 3 states and dates after March 15, 2020
    filter((state == "New York" | state == "Arizona" | state == "Nevada")
           & date >= "2020-03-15") %>%
    # group by state and date and calculate sum of cases and deaths
    group_by(state, date) %>%
    summarise(total_deaths = max(deaths),
              total_cases = max(cases))

# combine 2 data sets
combined_3st <- left_join(total_by_date_3st, population_3st, by = join_by(state == STNAME))

# calculate averages for 3 states
three_states_stats <- combined_3st %>%
  select(!Year) %>%
  mutate(
    # calculate deaths and cases per 100000 people
    deaths_per_100k = round(total_deaths / population * 100000, 4),
    cases_per_100k = round(total_cases / population * 100000, 2),
    # calculate rolling averages
    delta_deaths_7 = rollmean(deaths_per_100k - lag(deaths_per_100k, 1), k = 7, fill = NA, align = "rig
    delta_cases_7 = rollmean(cases_per_100k - lag(cases_per_100k, 1), k = 7, fill = NA, align = 'right'

# print result
three_states_stats
```

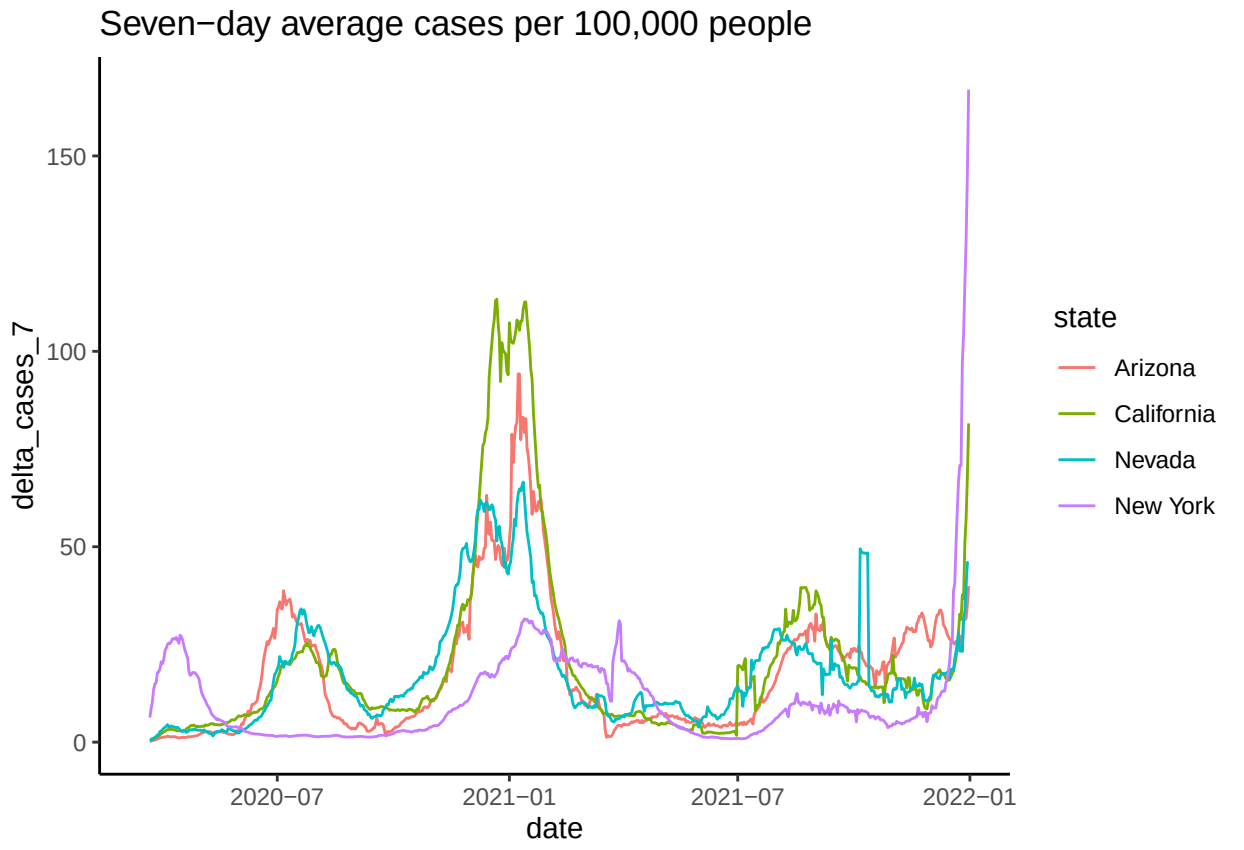
```
## # A tibble: 1,971 x 9
## # Groups:   state [3]
##    state   date       total_deaths total_cases population deaths_per_100k
##    <chr>   <date>            <dbl>       <dbl>      <dbl>           <dbl>
##  1 Arizona 2020-03-15            0           5    7276316          0
##  2 Arizona 2020-03-16            0           8    7276316          0
##  3 Arizona 2020-03-17            0           9    7276316          0
##  4 Arizona 2020-03-18            0          11    7276316          0
##  5 Arizona 2020-03-19            0          22    7276316          0
##  6 Arizona 2020-03-20            1          34    7276316          0.0137
##  7 Arizona 2020-03-21            1          49    7276316          0.0137
##  8 Arizona 2020-03-22            2          81    7276316          0.0275
##  9 Arizona 2020-03-23            2         139    7276316          0.0275
## 10 Arizona 2020-03-24            3         199    7276316          0.0412
## # i 1,961 more rows
## # i 3 more variables: cases_per_100k <dbl>, delta_deaths_7 <dbl>,
## #   delta_cases_7 <dbl>
```

– Communicate your methodology, results, and interpretation here –

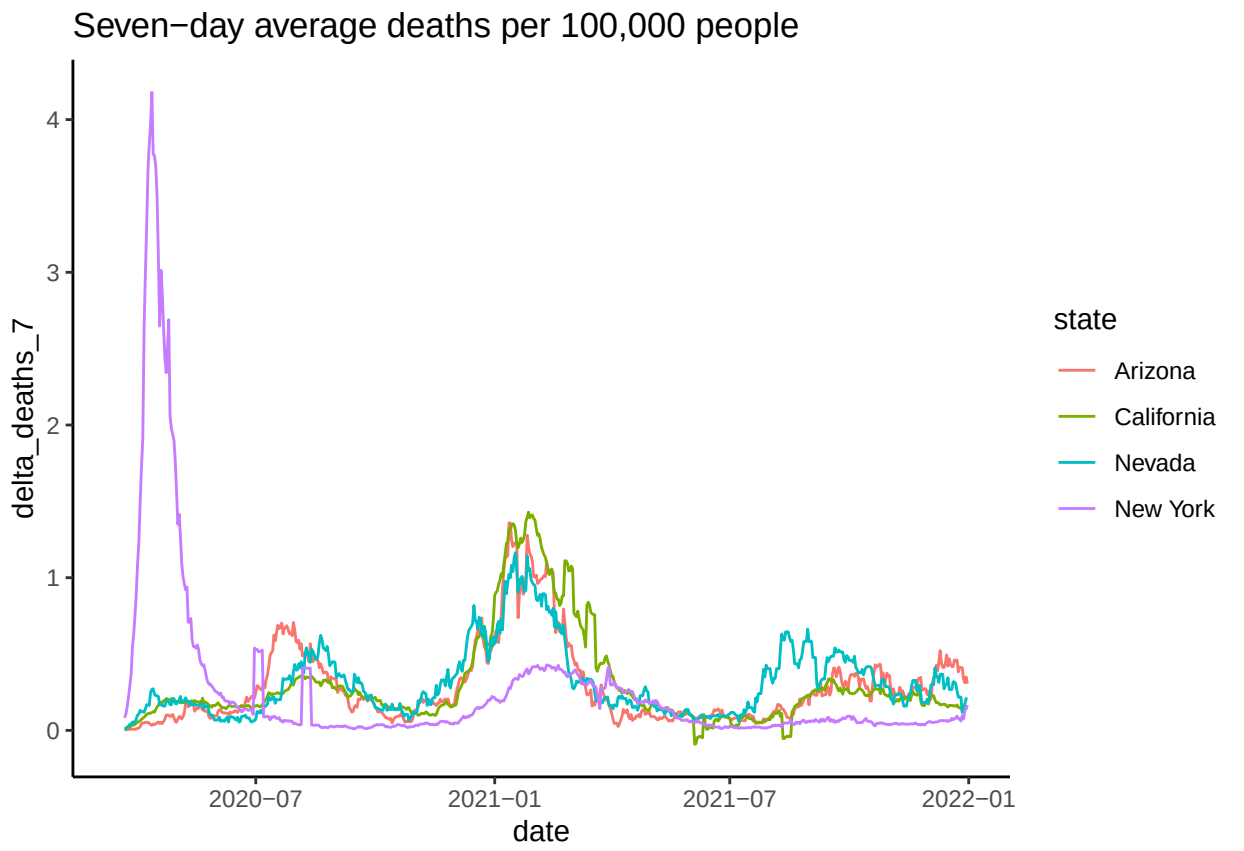
I used similar methodology as in Question 3, where I calculated statistics for California. Here, I found total population in 3 states: New York, Arizona and Nevada. Then, I calculated deaths and cases in these states, first simply by date, then per 100,000 people and, finally, calculated 7-day averages.

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.

```r
# combine 4 states into 1 dataframe
four_states_stats <- rbind(three_states_stats, ca_covid_stats)

# Visualization of seven-day average cases per 100,000 people in 4 states.
four_states_stats %>%
  group_by(state) %>%
  ggplot() +
  geom_line(aes(x = date, y = delta_cases_7, group = state, color = state), na.rm = TRUE) +
  ggtitle("Seven-day average cases per 100,000 people")
```




```r
# Visualization of seven-day average deaths per 100,000 people in 4 states.
four_states_stats %>%
  group_by(state) %>%
  ggplot() +
  geom_line(aes(x = date, y = delta_deaths_7, group = state, color = state), na.rm = TRUE) +
  ggtitle("Seven-day average deaths per 100,000 people")
```



– Communicate your methodology, results, and interpretation here –

I followed a similar methodology as before, but this time I combined data for four states.

The resulting visualization clearly displays the peaks in each of the four states. We observe high peaks in cases at the beginning and end of 2021. At the onset of 2021, California had the highest rate of cases, whereas at the end of 2021, the situation was most severe in New York. Regarding deaths, New York experienced the highest rates at the beginning of the pandemic, while in the beginning of 2021, the death rates were slightly elevated in all four states.

```
# 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.

```
# csse_us_cases and csse_us_deaths have dates as columns columns, so first of all, i will pivot_longer

csse_us_cases_pivoted <- csse_us_cases %>%
  pivot_longer(
```

```
    # choose only necessary dates
     cols = "3/15/20":"12/31/21",
     names_to = "Date",
     values_to = "Cases") %>%
  transmute(
    fips = FIPS,
    county = Admin2,
    state = Province_State,
    date = as.Date(Date, format = "%m/%d/%y"),
    cases = Cases) %>%
  filter(state == "Colorado")

csse_us_deaths_pivoted <- csse_us_deaths %>% pivot_longer(
    cols = "3/15/20":"12/31/21",
    names_to = "Date",
    values_to = "Deaths") %>%
  transmute(
    fips = FIPS,
    county = Admin2,
    state = Province_State,
    date = as.Date(Date, format = "%m/%d/%y"),
    deaths = Deaths) %>%
  filter(state == "Colorado")

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

csse_us <- full_join(csse_us_cases_pivoted,
                     csse_us_deaths_pivoted,
                     by = join_by(fips, county, state, date))

csse_us
```

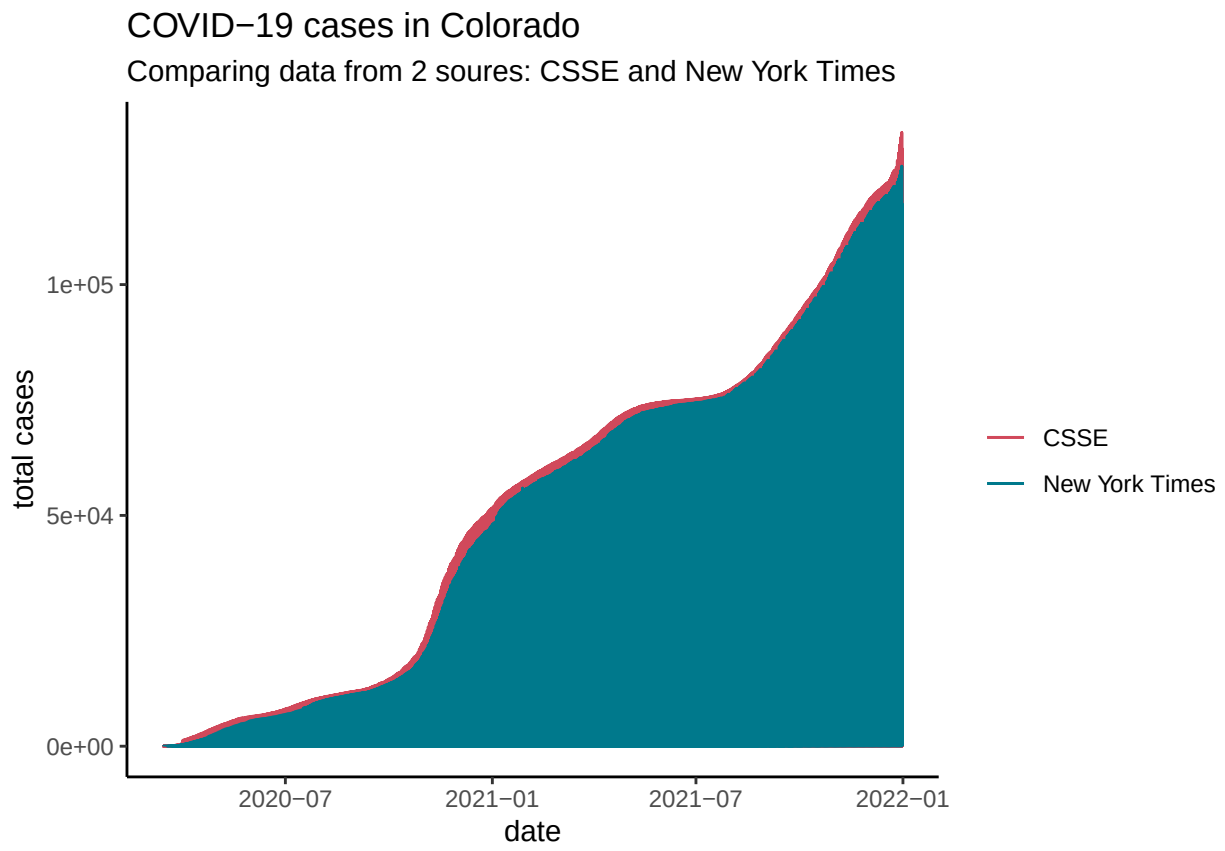
```
## # A tibble: 43,362 x 6
##     fips county state    date       cases deaths
##    <dbl> <chr>  <chr>    <date>     <dbl>  <dbl>
##  1  8001 Adams  Colorado 2020-03-15     6      0
##  2  8001 Adams  Colorado 2020-03-16     8      0
##  3  8001 Adams  Colorado 2020-03-17    10      0
##  4  8001 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
##  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
## # i 43,352 more rows
```

```
# combine CSSE and NYC datasets
# for that, first of all, create unique columns in each dataset

csse_co <- csse_us %>%
  rename(csse_total_cases = cases,
         csse_total_deaths = deaths)
```

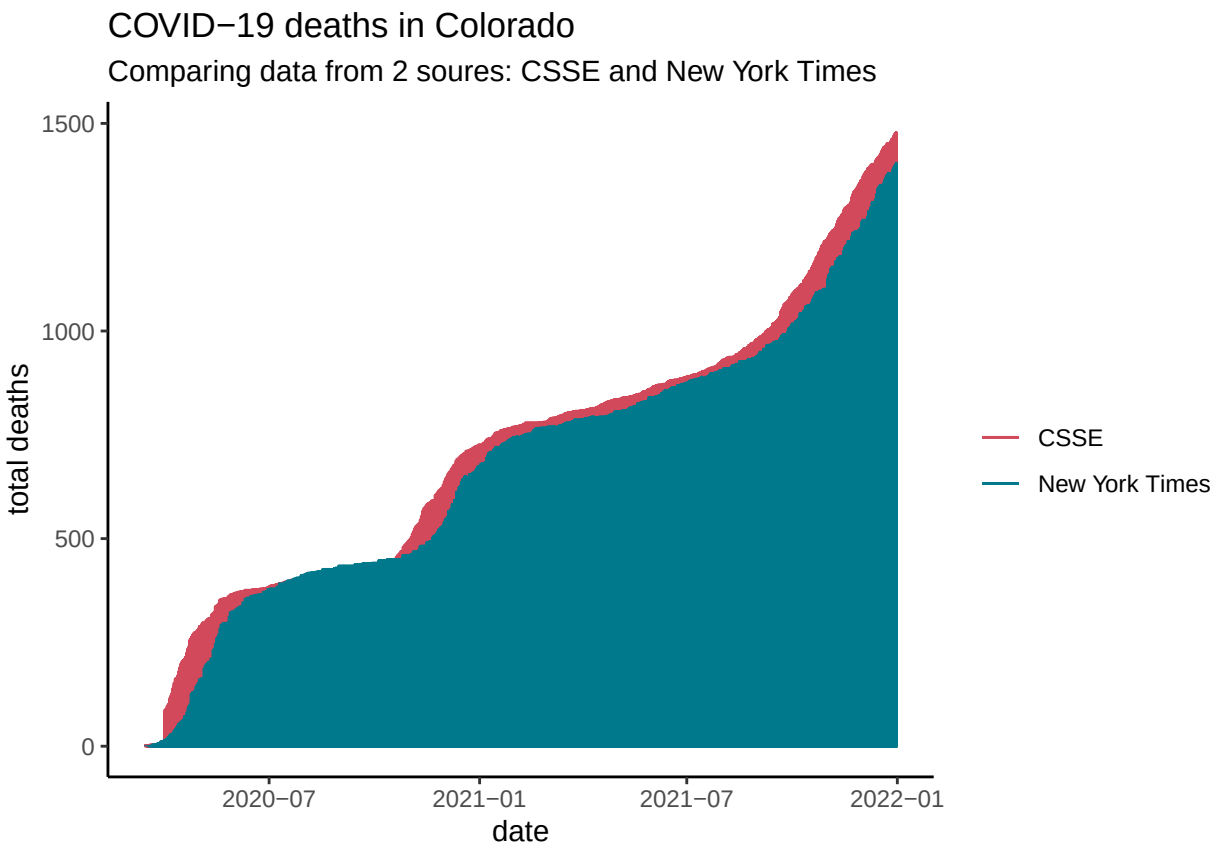
```r
nyt_co <- rbind(us_counties_2020, us_counties_2021) %>%
  filter(state == "Colorado" & date >= "2020-03-15") %>%
  group_by(fips, state, county, date) %>%
  summarise(nyt_total_cases = sum(cases, na.rm = TRUE),
            nyt_total_deaths = sum(deaths, na.rm = TRUE)) %>%
  transform(fips = as.numeric(fips))

csse_nyt <- full_join(
  csse_co, nyt_co, by = join_by(fips, state, county, date)
  )

csse_nyt
```

```
## # A tibble: 43,426 x 8
##     fips county state    date       csse_total_cases csse_total_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
## # i 43,416 more rows
## # i 2 more variables: nyt_total_cases <dbl>, nyt_total_deaths <dbl>
```

```r
csse_nyt %>%
  ggplot(aes(x=date)) +
  geom_line(aes(y = csse_total_cases, col = "CSSE")) +
  geom_line(aes(y = nyt_total_cases, col = "New York Times")) +
  labs(title = "COVID-19 cases in Colorado",
       subtitle = "Comparing data from 2 soures: CSSE and New York Times",
       y = "total cases") +
  scale_color_manual(name = "",
                     values = c("CSSE"="#d1495b", "New York Times"="#00798c"))
```




```
csse_nyt %>%
  ggplot(aes(x=date)) +
  geom_line(aes(y = csse_total_deaths, col = "CSSE")) +
  geom_line(aes(y = nyt_total_deaths, col = "New York Times")) +
  labs(title = "COVID-19 deaths in Colorado",
       subtitle = "Comparing data from 2 soures: CSSE and New York Times",
       y = "total deaths") +
  scale_color_manual(name = "",
                     values = c("CSSE"="#d1495b", "New York Times"="#00798c"))
```



– Communicate your methodology, results, and interpretation here –

I utilized the pivot_longer function to transform the date columns in the csse_us_cases table into row values. The resulting table included state, fips, county, date, and cases, and was filtered for Colorado between March 15, 2020, and December 31, 2021. Through the use of group_by, summarize, and sum functions, I calculated the total number of cases and deaths. I repeated this process for the csse_us_deaths table and subsequently merged the two resulting tables.

Next, I merged the csse and nyt tables based on fips, state, county, and date. Using ggplot, I created two line plots that compared the data from both sources, with the x-axis representing the date, and the y-axis displaying the total cases in the first plot and total deaths in the second.

The findings indicated that the csse and nyt data had similar figures, indicating their reliability. The visualizations provided a clear comparison of the reported cases and deaths, with the number of cases being considerably higher than the number of deaths.

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.

```
# Tidy the CSSE deaths and cases data sets
global_cases <- csse_global_cases %>%
    pivot_longer(cols = "3/15/20":"12/31/21",
                 names_to = "date",
                 values_to = "cases") %>%
    mutate(date = as.Date(date, format = "%m/%d/%y"))  %>%
    group_by(`Country/Region`, date) %>%
```

```
    summarize(total_cases = sum(cases), .groups = "keep") %>%
    mutate(max_cases = max(total_cases)) %>%
    filter(max_cases == total_cases) %>%
    select(`Country/Region`, date, total_cases)

global_deaths <- csse_global_deaths %>%
  pivot_longer(cols = "3/15/20":"12/31/21",
               names_to = "date",
               values_to = "deaths") %>%
  mutate(date = as.Date(date, format = "%m/%d/%y"))  %>%
  group_by(`Country/Region`, date) %>%
  summarize(total_deaths = sum(deaths), .groups = "keep") %>%
  mutate(max_deaths = max(total_deaths)) %>%
  filter(max_deaths == total_deaths) %>%
  select(`Country/Region`, date, total_deaths)

# Tidy the global population estimates
global_population <- global_population_estimates %>%
  filter(!is.na(`Series Code`)) %>%
  pivot_longer(
    cols = c(`2020 [YR2020]`,`2021 [YR2021]`),
    names_to = "year",
    values_to = "population",
    names_pattern = "([0-9]*)",   # pattern for 2020 and 2021
    values_transform = list(population = as.numeric)
)
```

```
## Warning in .f(.x[[i]], ...): NAs introduced by coercion
```

```
## Warning in .Primitive("as.double")(x, ...): NAs introduced by coercion
```

```
# Combine data sets
# Identify top 10 countries in cases per 100k
global_cases <- global_cases %>%
  mutate(year = str_sub(date,1,4))

top_countries_cases <- global_population %>%
    inner_join(global_cases,
               by = join_by("Country Name" == "Country/Region",
                            year == year)) %>%
  mutate(cases_per_100k = round(total_cases / population * 100000, 0)) %>%
  group_by(`Country Name`) %>%
  summarise(total_cases = max(total_cases),
            cases_per_100k = max(cases_per_100k)) %>%
  arrange(desc(cases_per_100k)) %>%
  head(10)

top_countries_cases
```

```
## # A tibble: 10 x 3
##    `Country Name` total_cases cases_per_100k
##    <chr>                <dbl>          <dbl>
```

```
##  1 Andorra              23740          30831
##  2 Montenegro          170034          27381
##  3 Georgia             934741          25182
##  4 Seychelles           24788          25038
##  5 San Marino            8202          24124
##  6 Slovenia            464048          22087
##  7 Mongolia            692621          20806
##  8 United Kingdom    13010853          19274
##  9 Lithuania           524427          18960
## 10 Serbia             1299339          18933
```

```
# Identify top 10 countries in deaths per 100k
global_deaths <- global_deaths %>%
  mutate(year = str_sub(date,1,4))

top_countries_deaths <- global_population %>%
    inner_join(global_deaths,
               by = join_by("Country Name" == "Country/Region",
                            year == year)) %>%
  mutate(deaths_per_100k = round(total_deaths / population * 100000, 0)) %>%
  group_by(`Country Name`) %>%
  summarise(total_deaths = max(total_deaths),
            deaths_per_100k = max(deaths_per_100k)) %>%
  arrange(desc(deaths_per_100k)) %>%
  head(10)

top_countries_deaths
```

```
## # A tibble: 10 x 3
##    `Country Name`         total_deaths deaths_per_100k
##    <chr>                         <dbl>           <dbl>
##  1 Peru                         202690             608
##  2 Bulgaria                      30955             450
##  3 Bosnia and Herzegovina        13442             412
##  4 Hungary                       39186             403
##  5 Moldova                       10275             393
##  6 Montenegro                     2411             388
##  7 North Macedonia                7960             384
##  8 Georgia                       13800             372
##  9 Croatia                       12538             312
## 10 Romania                       58752             307
```

– Communicate your methodology, results, and interpretation here –

To clean the CSSE deaths and cases data sets, I first transformed the data from a wide to a long format using pivot_longer. Then, I added two columns for the date and number of cases or deaths and converted the date column to a date format. After grouping the data by country and date, I calculated the total cases or deaths per country.

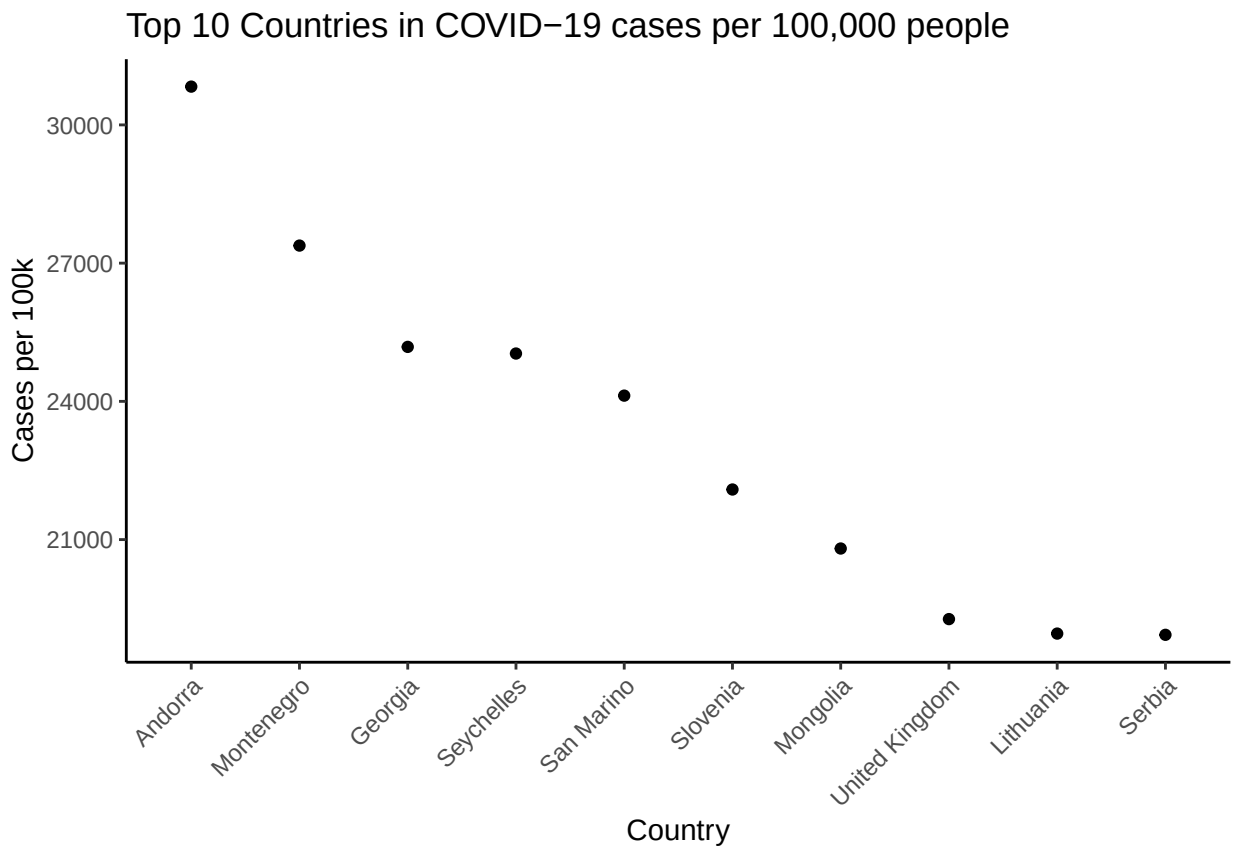
Next, I cleaned the global population estimates dataset by dropping rows without a 'Series Code' and using pivot_longer.

Finally, I combined the CSSE datasets with the global population dataset by country and year. I added a column for cases per 100,000 people and sorted the datasets by this column in descending order.

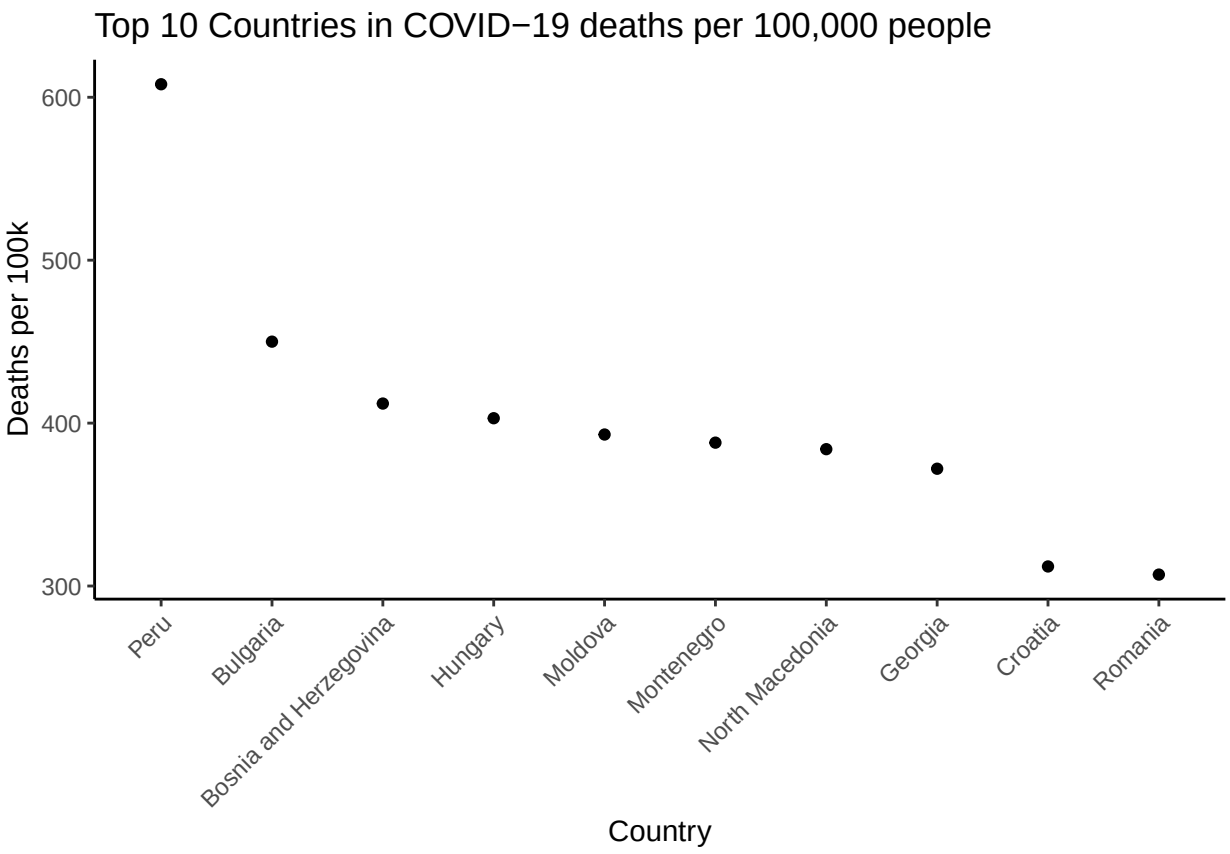
This resulted in two tables with the top 10 countries ranked by cases and deaths per 100,000 people. Although some countries appeared in both tables, the rankings of the top 10 countries varied significantly. Notably, the countries with the highest deaths per 100,000 were not always the same as those with the highest cases per 100,000. This suggests that other factors, such as public health measures and access to personal protective equipment (PPE), may influence the number of deaths caused by COVID-19, regardless of the number of cases.

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.

```
ggplot(top_countries_cases, aes(x = fct_reorder(`Country Name`, desc(cases_per_100k)), y = cases_per_10
    geom_point() +
    labs(title = "Top 10 Countries in COVID-19 cases per 100,000 people",
         y = "Cases per 100k", x = "Country") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))
```




```
ggplot(top_countries_deaths, aes(x = fct_reorder(`Country Name`, desc(deaths_per_100k)), y = deaths_per
    geom_point() +
    labs(title = "Top 10 Countries in COVID-19 deaths per 100,000 people",
         y = "Deaths per 100k", x = "Country") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust=1))
```



– Communicate your methodology, results, and interpretation here –

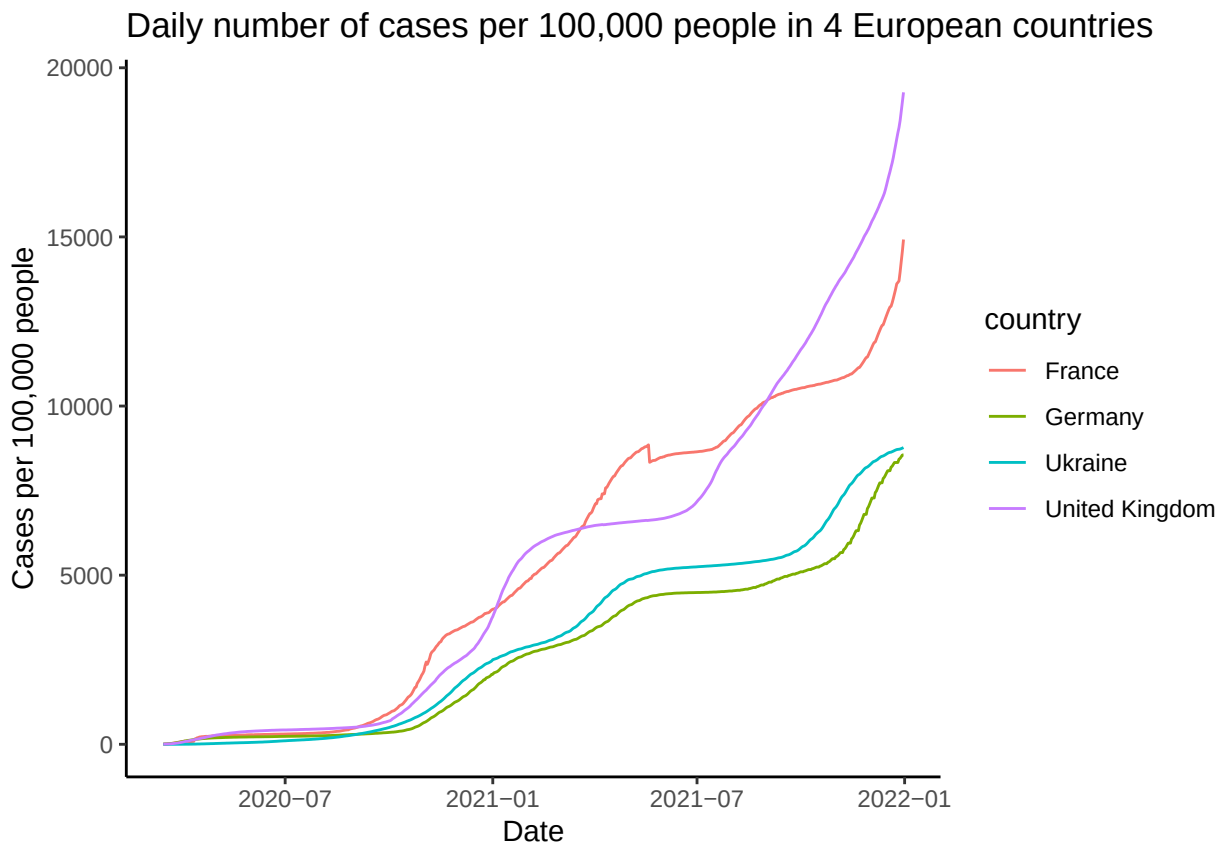
Using ggplot, I generated scatter plots that plotted countries on the x-axis and the total number of cases or deaths on the y-axis.

The results revealed the countries with the highest numbers of COVID-19 cases and deaths. Surprisingly, the countries with the highest number of cases were mostly small countries such as Andorra, Montenegro, Seychelles, and San Marino. On the other hand, Peru had the highest mortality rate per 100k, but it did not rank among the top 10 countries in terms of cases per 100k.

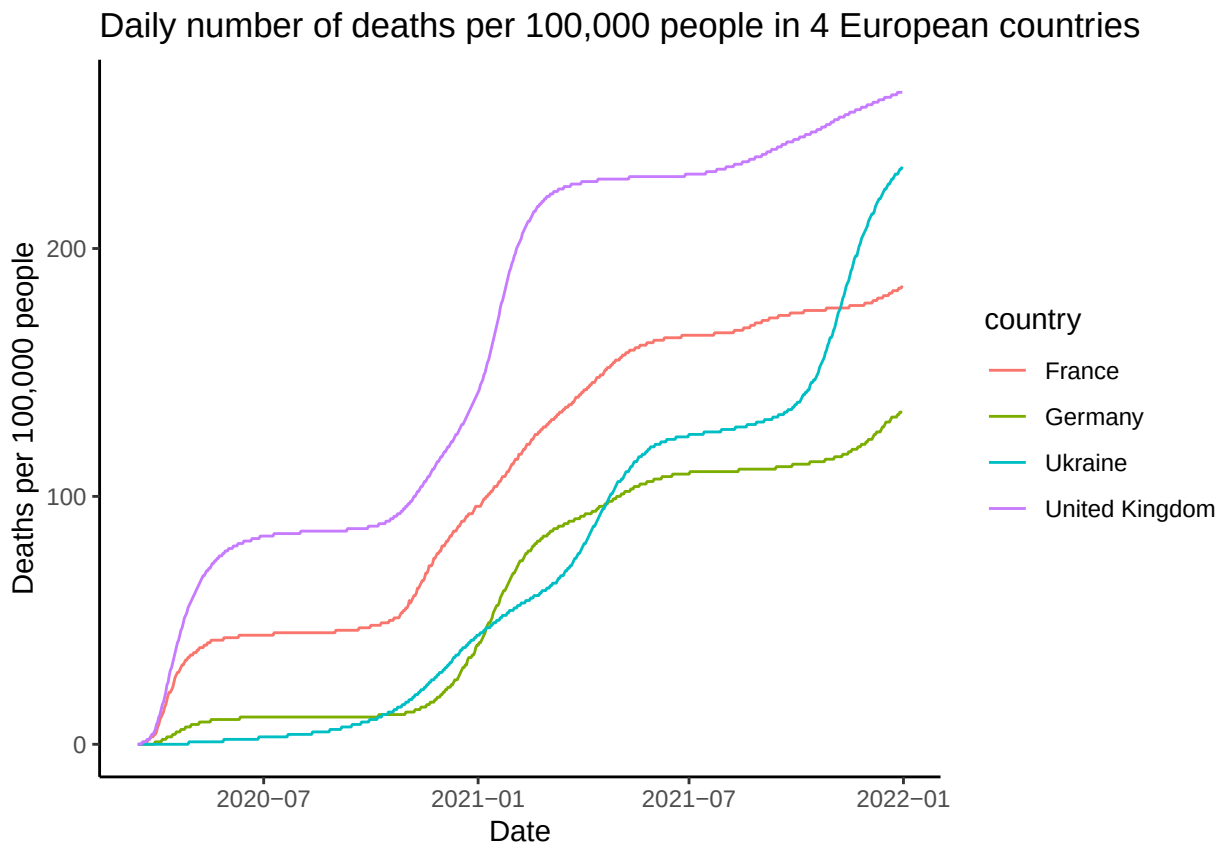
Question 4 Finally, select four countries from one continent and create visualizations for the daily number of confirmed cases per 100,000 and the daily number of deaths per 100,000 people between March 15, 2020, and December 31, 2021.

```
# Find daily numbers of cases and deaths in 4 European countries
europe_4_cases <- global_cases %>%
  rename(country = `Country/Region`) %>%
  filter(country == "France" | country == "United Kingdom" |
           country == "Germany" | country == "Ukraine")

europe_4_deaths <- global_deaths %>%
  rename(country = `Country/Region`) %>%
  filter(country == "France" | country == "United Kingdom" |
           country == "Germany" | country == "Ukraine")


# Filter global population estimates to show only population for 4 chosen countries
```

```r
europe_4_population <- global_population %>%
  rename(country = `Country Name`) %>%
  select(country, year, population) %>%
  filter(country == "France" | country == "United Kingdom" |
           country == "Germany" | country == "Ukraine")


# Combine data sets
# and calculate cases and deaths per 100,000 people
europe_4_cases_per100k <- europe_4_cases %>%
  left_join(europe_4_population,
            by = join_by(country, year)) %>%
  mutate(cases_per_100k = round(total_cases / population * 100000, 0))

europe_4_deaths_per100k <- europe_4_deaths %>%
  left_join(europe_4_population,
            by = join_by(country, year)) %>%
  mutate(deaths_per_100k = round(total_deaths / population * 100000, 0))

# create visualizations
europe_4_cases_per100k %>%
  group_by(country) %>%
  ggplot() +
  geom_line(aes(x = date, y = cases_per_100k, group = country, color = country), na.rm = TRUE) +
  ggtitle("Daily number of cases per 100,000 people in 4 European countries") +
  labs(x = "Date", y = "Cases per 100,000 people")
```




```
europe_4_deaths_per100k %>%
  group_by(country) %>%
  ggplot() +
  geom_line(aes(x = date, y = deaths_per_100k, group = country, color = country), na.rm = TRUE) +
  ggtitle("Daily number of deaths per 100,000 people in 4 European countries")  +
  labs(x = "Date", y = "Deaths per 100,000 people")
```



– Communicate your methodology, results, and interpretation here –

I used datasets created in Question 2 that contain global cases, global deaths and global population. For convenience, I renamed the country column and then I filtered the datasets to show only four chosen countries: France, Germany, Ukraine and United Kingdom. I then combined the population dataset with the cases and deaths data useing the country and date columns. Unlike in Question 2, I did not group by country or search for the total number of cases and deaths per country. Instead, I was interested in tracking changes in cases and deaths over time. I added columns that showed cases and deaths per 100,000 people and created visualizations using ggplot and geom_line.

First visualization revealed that all four countries had similar starting points in 2020, but France and the United Kingdom emerged as the pot two countries, regularly switching positions. Meanwhile, Ukraine and Germany had very similar trends, with Ukraine recording slightly higher numbers of cases throughout the period.

The second visualization showeda greter disparity in death rates between countries, compared to case rates. From March 15, 2020 until the end of 2021, the United Kingdom had the highest number of deaths among the four countries. France held the second position until the end of 2021 when Ukraine took over. While Ukraine and Germany ranked third and fourth, respectively, throughout most of the period, Ukraine's death rate sharply increased towards the end of 2021.