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
```

```
## Parsed with column specification:
## cols(
##   date = col_date(format = ""),
##   county = col_character(),
##   state = col_character(),
##   fips = col_character(),
##   cases = col_double(),
##   deaths = col_double()
## )
```

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

```
## Parsed with column specification:
## cols(
##   date = col_date(format = ""),
##   county = col_character(),
##   state = col_character(),
##   fips = col_character(),
##   cases = col_double(),
##   deaths = col_double()
## )
```

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

```
## Parsed with column specification:
## cols(
##   date = col_date(format = ""),
##   county = col_character(),
```

```
##   state = col_character(),
##   fips = col_character(),
##   cases = col_double(),
##   deaths = col_double()
## )
```

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

```
## Parsed with column specification:
## cols(
##   STNAME = col_character(),
##   CTYNAME = col_character(),
##   fips = col_double(),
##   STATE = col_double(),
##   COUNTY = col_double(),
##   Year = col_double(),
##   Estimate = col_double()
## )
```

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.
#
## YOUR CODE HERE ##

# combining all 3 datasets to form one df using rbind
combined_df <- rbind(us_counties_2020, us_counties_2021, us_counties_2022)

# performing the required tidying and other data preprocessing of data needed
df1 <- combined_df %>%
  filter(state != 'Puerto Rico', date >= '2020-03-15') %>% # filtering out the rows with Puerto Rico as
  arrange(date) %>% # arranging as per date in ascending order
  select(date, cases, deaths) %>% # selecting the only required columns
  group_by(date) %>% # grouping by date to sum up total deaths and cases
  summarise(total_deaths = sum(deaths),
            total_cases = sum(cases))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
# creating the 3 variables for max date, total deaths and total cases and using the tail function to ge
max_date <- tail(df1, n = 1)$date # replace the quotes with your code to find the most recent date in t
us_total_cases <- tail(df1, n = 1)$total_cases
us_total_deaths <- tail(df1, n = 1)$total_deaths
df1
```

```
## # A tibble: 1,022 x 3
##    date       total_deaths total_cases
##    <date>            <dbl>       <dbl>
##  1 2020-03-15           68        3595
```

```
##  2 2020-03-16           91        4502
##  3 2020-03-17          117        5901
##  4 2020-03-18          162        8345
##  5 2020-03-19          212       12387
##  6 2020-03-20          277       17998
##  7 2020-03-21          359       24507
##  8 2020-03-22          457       33050
##  9 2020-03-23          577       43474
## 10 2020-03-24          783       53899
## # ... with 1,012 more rows
```

```
# 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 –

Using the rbind function, it is possible to combine the three datasets as required. The US territory 'Puerto Rico' can be filtered away first and the dataset can be arranged and necessary columns was selected. The dateset was grouped by the date columns since we need the total deaths and cases for every date. And since we already arranged the data in ascending order previous, the last row after grouping ans summing up the deaths and cases columns will give you the required answers for the total deaths and cases as on '2022-12-31' As of December 31, 2022, the total number of deaths in the US was 1.094296×10^6 arising out of 9.9374764×10^7 total cases! Assuming each person that tested positive for covid, did so only once - we can estimate that the mortality rate was 0.0110118

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?

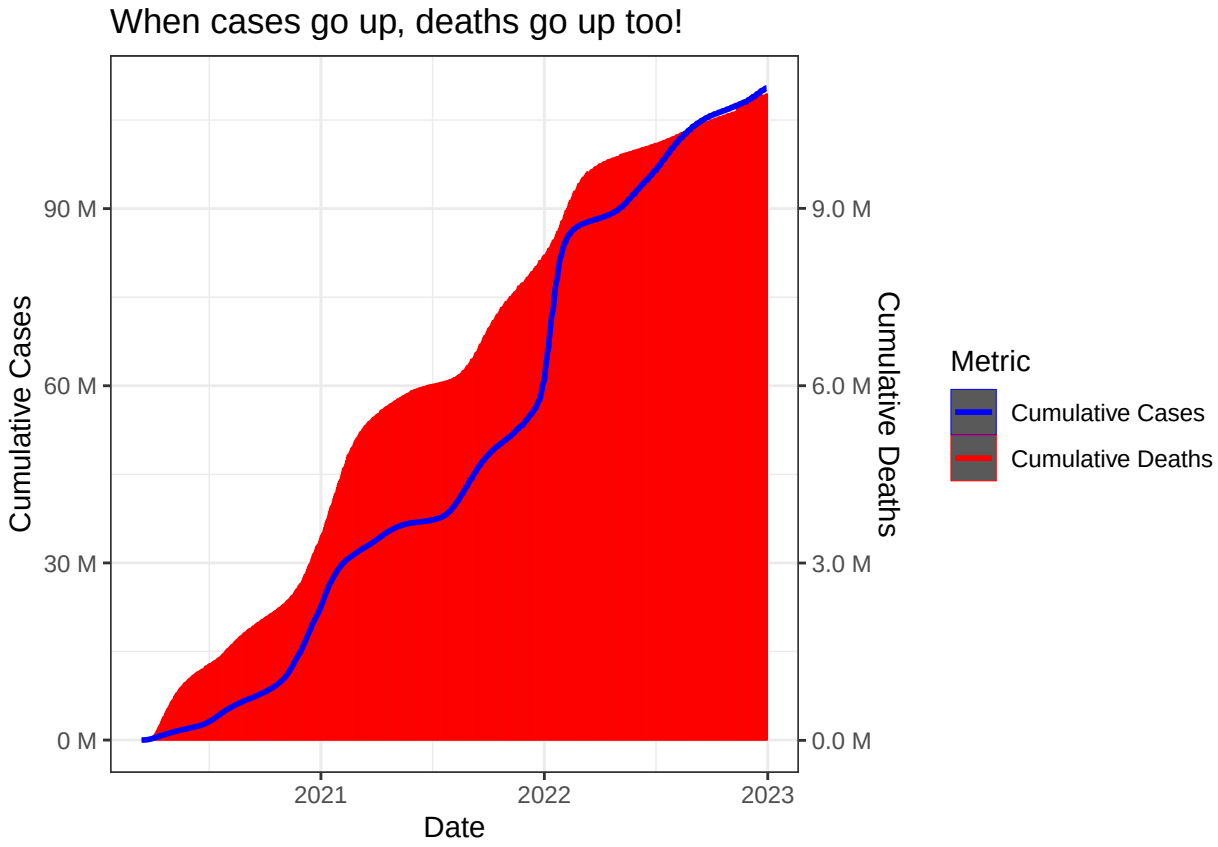
```
# Create a visualization for the total number of US cases and deaths since March 15, 2020.
#
## YOUR CODE HERE ##

library(scales)
```

```
##
## Attaching package: 'scales'
```

```
## The following object is masked from 'package:purrr':
##
```

```
##     discard
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
df1$total_deaths <- as.numeric(df1$total_deaths)
df1$total_cases <- as.numeric(df1$total_cases)

coeff <- 90

ggplot(df1, aes(x = date)) +
  geom_bar(aes(y = total_deaths, color = "Cumulative Deaths"), size = 0.1, stat = "identity", alpha = 1
  geom_line(aes(y = total_cases/coeff, color = "Cumulative Cases"), size = 1, position = position_nudge
  labs(x = "Date", color = "Metric") +
  scale_color_manual(values = c("Cumulative Deaths" = "red", "Cumulative Cases" = "blue")) +
  theme_bw() +
  scale_y_continuous(
    name = "Cumulative Cases",
    labels = unit_format(unit = "M", scale = 100e-6),
    sec.axis = sec_axis(~.*10, name = "Cumulative Deaths", labels = unit_format(unit = "M", scale = 1e-
  ) +
ggtitle("When cases go up, deaths go up too!")
```



– Communicate your methodology, results, and interpretation here –

The total number of US cases and deaths is visualized using line and bar plots. Mainly, making use of a secondary y-axis to have both cases and deaths represented in a single chart. Based on the values of total deaths and total cases, it is apparent that the cases is around 99 million, and deaths is just over 1 million.

This makes cases a 100 times multiple of deaths approximately. Hence I'm taking a coefficient of 100 to get the scales to be similar to get the visualization better. A bar plot is used for deaths which is in red in color and a line plot in blue for total cases. The left y-axis indicates the cumulative cases and the right y-axis which is the secondary axis indicates the cumulative deaths. It is apparent that when the total number of cases go up, the number of deaths go up too.

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 ca
#
# Hint: Look at the documentation for lag() when computing the number of new deaths and cases and the s
#
#
## YOUR CODE HERE ##

# loading the zoo library needed to calculate the rolling averages
library(zoo)
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
# calculating the daily changes in cases and deaths and the 7 day rolling averages for both
df2 <- df1 %>%
  mutate(delta_deaths_1 = c(NA, diff(total_deaths)),
         delta_cases_1 = c(NA, diff(total_cases)),
         delta_deaths_7 = rollapply(delta_deaths_1, width = 7, FUN = mean, align = "right", fill = NA),
         delta_cases_7 = rollapply(delta_cases_1, width = 7, FUN = mean, align = "right", fill = NA))

# viewing the new df
df2
```

```
## # A tibble: 1,022 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
## # ... with 1,012 more rows, and 2 more variables: delta_deaths_7 <dbl>,
## #   delta_cases_7 <dbl>
```

```r
# getting the dates which had the maximum occurrences of cases and deaths
max_new_cases_date <- df2[which.max(df2$delta_cases_1), ]$date
max_new_deaths_date <- df2[which.max(df2$delta_deaths_1), ]$date
min_new_cases_date <- df2[which.min(df2$delta_cases_1), ]$date
min_new_deaths_date <- df2[which.min(df2$delta_deaths_1), ]$date
```

```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
```

– Communicate your methodology, results, and interpretation here –

Used the function rollapply to get the 7 day rolling averages and diff function to the get the daily changes in cases and deaths. The date that had the maximum no of cases is - 2022-01-10 The date that had the maximum no of deaths is - 2022-11-11 The date that had the minimum no of cases is - 2021-06-04 The date that had the minimum no of deaths is - 2022-03-14

```r
# Create a new table, based on the table from Question 3, and calculate the number of new deaths and ca

# Hint: To calculate per 100,000 people, first tidy the population estimates data and calculate the US
#
# Hint: look at the help documentation for grepl() and case_when() to divide the averages by the US pop
# 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"))
#

## YOUR CODE HERE ##

# calculating the total population of the US
pop_df <- us_population_estimates %>%
  group_by(Year) %>%
  summarise(total_pop = sum(Estimate))
```

Question 4

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
# creating separate variables to hold these population numbers
pop_2020 <- pop_df$total_pop[1]
pop_2021 <- pop_df$total_pop[2]

# using the grepl and case_when functions to get the number of cases and deaths per 100,000 people
# since the us_population_estimates did not have population data for the year 2022, any dates in the ye
df3 <- df2 %>%
  filter(date < "2022-01-01") %>%
  mutate(total_deaths = case_when(grepl("2020", date) ~ (total_deaths/pop_2020)*100000,
                                  grepl("2021", date) ~ (total_deaths/pop_2021)*100000),
         total_cases = case_when(grepl("2020", date) ~ (total_cases/pop_2020)*100000,
                                 grepl("2021", date) ~ (total_cases/pop_2021)*100000),
         delta_deaths_1 = case_when(grepl("2020", date) ~ (delta_deaths_1/pop_2020)*100000,
                                    grepl("2021", date) ~ (delta_deaths_1/pop_2021)*100000),
         delta_cases_1 = case_when(grepl("2020", date) ~ (delta_cases_1/pop_2020)*100000,
                                   grepl("2021", date) ~ (delta_cases_1/pop_2021)*100000),
         delta_deaths_7 = case_when(grepl("2020", date) ~ (delta_deaths_7/pop_2020)*100000,
                                    grepl("2021", date) ~ (delta_deaths_7/pop_2021)*100000),
         delta_cases_7 = case_when(grepl("2020", date) ~ (delta_cases_7/pop_2020)*100000,
                                   grepl("2021", date) ~ (delta_cases_7/pop_2021)*100000))

# viewing the end data
df3
```

```
## # A tibble: 657 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
## # ... with 647 more rows, and 2 more variables: delta_deaths_7 <dbl>,
## #   delta_cases_7 <dbl>
```

```
tail(df3)
```

```
## # A tibble: 6 x 7
##   date       total_deaths total_cases delta_deaths_1 delta_cases_1
##   <date>            <dbl>       <dbl>          <dbl>         <dbl>
## 1 2021-12-26         245.      15661.         0.0238          55.2
## 2 2021-12-27         245.      15825.         0.525          163.
## 3 2021-12-28         246.      15939.         0.752          114.
## 4 2021-12-29         247.      16084.         0.632          145.
## 5 2021-12-30         247.      16256.         0.424          172.
## 6 2021-12-31         247.      16386.         0.358          130.
## # ... with 2 more variables: delta_deaths_7 <dbl>, delta_cases_7 <dbl>
```

```
summary(df3)
```

```
##       date             total_deaths        total_cases
##  Min.   :2020-03-15   Min.   :  0.02051   Min.   :    1.084
##  1st Qu.:2020-08-26   1st Qu.: 54.05684   1st Qu.: 1751.967
##  Median :2021-02-06   Median :138.65281   Median : 8083.863
##  Mean   :2021-02-06   Mean   :123.53855   Mean   : 6825.658
##  3rd Qu.:2021-07-20   3rd Qu.:182.75397   3rd Qu.:10263.390
##  Max.   :2021-12-31   Max.   :247.37767   Max.   :16385.805
##
##  delta_deaths_1     delta_cases_1     delta_deaths_7    delta_cases_7
##  Min.   :0.006938   Min.   : -7.253   Min.   :0.01676   Min.   :  1.269
##  1st Qu.:0.152941   1st Qu.:  9.478   1st Qu.:0.21448   1st Qu.: 10.357
##  Median :0.298464   Median : 17.260   Median :0.31422   Median : 19.250
##  Mean   :0.377256   Mean   : 24.988   Mean   :0.37845   Mean   : 24.567
##  3rd Qu.:0.529543   3rd Qu.: 35.364   3rd Qu.:0.49988   3rd Qu.: 35.711
##  Max.   :1.644502   Max.   :171.960   Max.   :1.00630   Max.   :113.149
##  NA's   :1          NA's   :1         NA's   :7         NA's   :7
```

```
#  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
```

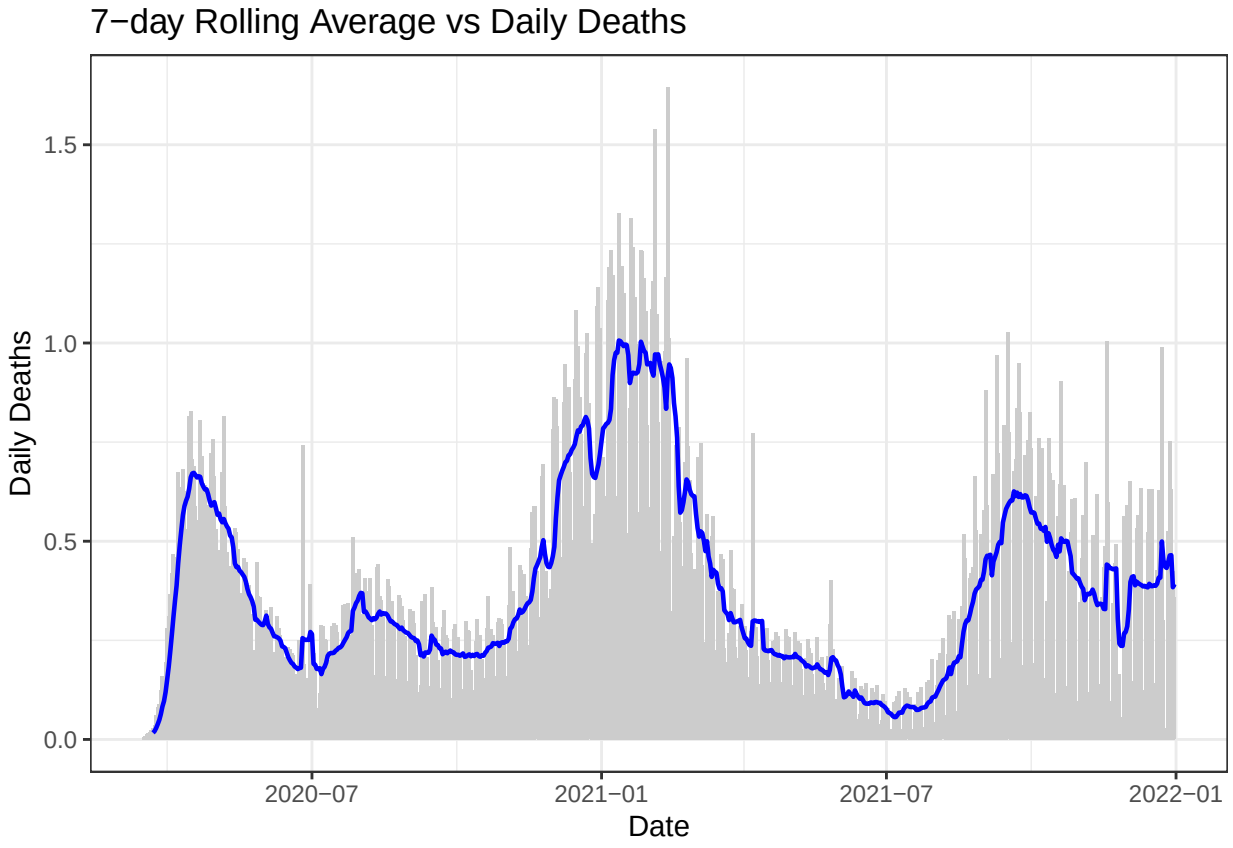
– Communicate your methodology, results, and interpretation here –

When comparing the beginning of 2020 with end of 2021, the number of cases increased from 1 to 16000 roughly per 100,000 people. And the number of deaths also increased from 0.02 to 247 per 100,000 people during the same time period.

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

# visualization for deaths
ggplot(df3, aes(x = date)) +
  geom_col(aes(y = delta_deaths_1), fill = "gray80", width = 2, stat = "identity", alpha = 1) +  # Raw
  geom_line(aes(y = delta_deaths_7), color = "blue", size = 0.8) +  # Rolling average as line
  labs(x = "Date", y = "Daily Deaths", title = "7-day Rolling Average vs Daily Deaths") +
  theme_bw()
```

Question 5

```
## Warning: Ignoring unknown parameters: stat

## Warning: Removed 1 rows containing missing values (position_stack).

## Warning: position_stack requires non-overlapping x intervals

## Warning: Removed 7 row(s) containing missing values (geom_path).
```



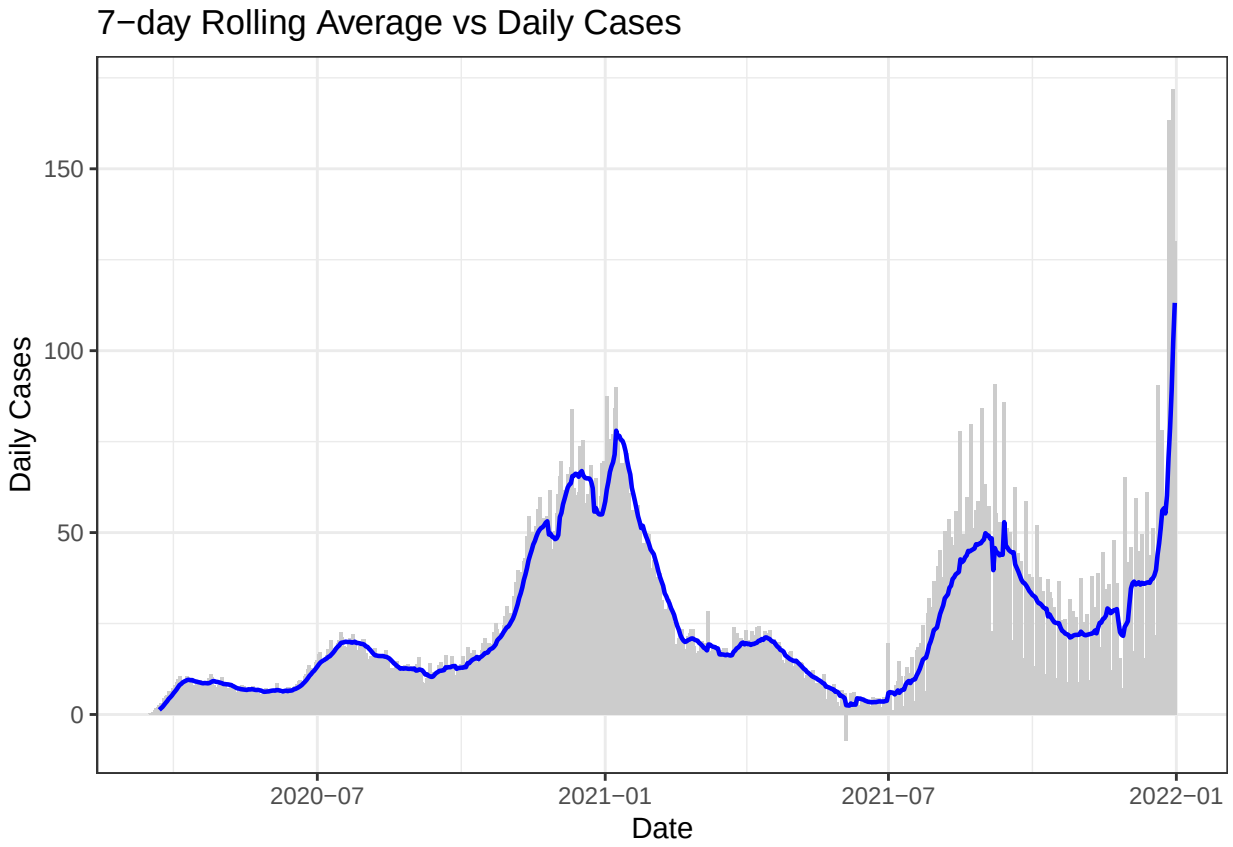
```r
# visualization for cases
ggplot(df3, aes(x = date)) +
  geom_col(aes(y = delta_cases_1), fill = "gray80", width = 2, stat = "identity", alpha = 1) +  # Raw s
  geom_line(aes(y = delta_cases_7), color = "blue", size = 0.8) +  # Rolling average as line
  labs(x = "Date", y = "Daily Cases", title = "7-day Rolling Average vs Daily Cases") +
  theme_bw()
```

```
## Warning: Ignoring unknown parameters: stat
```

```
## Warning: Removed 1 rows containing missing values (position_stack).
```

```
## Warning: position_stack requires non-overlapping x intervals
```

```
## Warning: Removed 7 row(s) containing missing values (geom_path).
```



– Communicate your methodology, results, and interpretation here –

To get a better view of the 7 day rolling averages that is indicated by the blue line, it is compared with the actual raw deaths and cases that occurred on a per day basis that is shown by the gray column bars. There are two visualizations for cases and deaths here. It looks like the maximum daily deaths occurred towards the end of 2020 and the beginning of 2021. And wrt cases, daily occurrences peaked towards the end of 2020 and a similar trend occurred in the month of Aug-sept and had maximum cases towards the beginning of 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

# creating a df grouped by state and date to find the total sum of deaths and cases
# appropriate filters have been added as per the question
state_df <- combined_df %>%
  filter(state != 'Puerto Rico', date >= "2020-03-15", date <= "2021-12-31") %>%
  group_by(state, date) %>%
  summarise(total_deaths = sum(deaths),
            total_cases = sum(cases))
```

```
## `summarise()` regrouping output by 'state' (override with `.groups` argument)
```

```
# using the above df to sort it based on total cases in descending order and obtaining the top 10 state
state_df <- state_df %>%
  arrange(desc(total_cases)) %>%
  distinct(state, .keep_all = TRUE)

state_df
```

```
## # A tibble: 55 x 4
## # Groups:   state [55]
##    state          date       total_deaths total_cases
##    <chr>          <date>            <dbl>       <dbl>
##  1 California     2021-12-31        76709     5515613
##  2 Texas          2021-12-31        76062     4574881
##  3 Florida        2021-12-31        62504     4166392
##  4 New York       2021-12-31        58993     3473970
##  5 Illinois       2021-12-30        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       2021-12-29        28984     1706355
## 10 North Carolina 2021-12-31        19436     1685504
## # ... with 45 more rows
```

```
# 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
```

– Communicate your methodology, results, and interpretation here –

It looks like California was the state that was affected the most in terms of total cases and deaths during the duration of March 2020 to December 2021. Followed by Texas and Florida taking the 2nd and 3rd position in this ranking. It seems a bit expected as well since these states are also very highly populated and the chances of a person testing positive in these states are much higher than the other states.

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

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

```
# Determine the top 10 states in terms of deaths and cases per 100,000 people between March 15, 2020, a


# obtaining the year from the overall date
state_df <- state_df %>%
    mutate(year = year(date))

# grouping the population data by state and year to find the sum of total population for each state and
state_pop_df <- us_population_estimates %>%
    group_by(STNAME, Year) %>%
    summarise(total_pop = sum(Estimate)) %>%
    arrange(STNAME, Year)
```

```
## `summarise()` regrouping output by 'STNAME' (override with `.groups` argument)
```

```
# finding the deaths and cases per 100,000 people for each state using its own respective population
state_per_100k <- state_df %>%
  full_join(state_pop_df, by = c("state" = "STNAME", "year" = "Year")) %>%
  mutate(deaths_per_100k = (total_deaths/total_pop)*100000,
         cases_per_100k = (total_cases/total_pop)*100000) %>%
  select(state, date, deaths_per_100k, cases_per_100k) %>%
  arrange(desc(cases_per_100k)) %>%
  head(n = 10)

# viewing the data
state_per_100k
```

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

```
# 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
```

– Communicate your methodology, results, and interpretation here –

It looks like the list from Question 1 is very different to the above list when calculated per 100,000 people. States such as California and Texas which were at the top of the list in Q1 is not even in the top 10 when calculating per 100,000. This goes to show how population plays a role in determining how affected a place is. The list in Q2 shows how states have been affected relatively when their respective population is considered. States such as North Dakota and Alaska may not have had high occurrences of deaths and cases overall, but considering its population, it starts to rank high. The second method seems to be a better indicator of how responsive the state has been to in its efforts to curb the spread of the disease.

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

# selecting the state of colorado and obtaining sum of total deaths and cases
colorado_df <- combined_df %>%
  filter(state == 'Colorado', date >= "2020-03-15", date <= "2021-12-31") %>%
  group_by(state, date) %>%
  summarise(total_deaths = sum(deaths),
            total_cases = sum(cases)) %>%
  mutate(year = year(date))
```

```
## `summarise()` regrouping output by 'state' (override with `.groups` argument)
```

```
# using tha above information and after performing a join with its population data - calculating the ro
rolling_colorado_df <- colorado_df %>%
    left_join(state_pop_df, by = c("state" = "STNAME", "year" = "Year")) %>%
    mutate(deaths_per_100k = (total_deaths/total_pop)*100000,
           cases_per_100k = (total_cases/total_pop)*100000,
           deaths_7_day = (rollapply(c(NA, diff(total_deaths)), width = 7, FUN = mean, align = "right",
           cases_7_day = (rollapply(c(NA, diff(total_cases)), width = 7, FUN = mean, align = "right", fi

# viewing the data
rolling_colorado_df
```

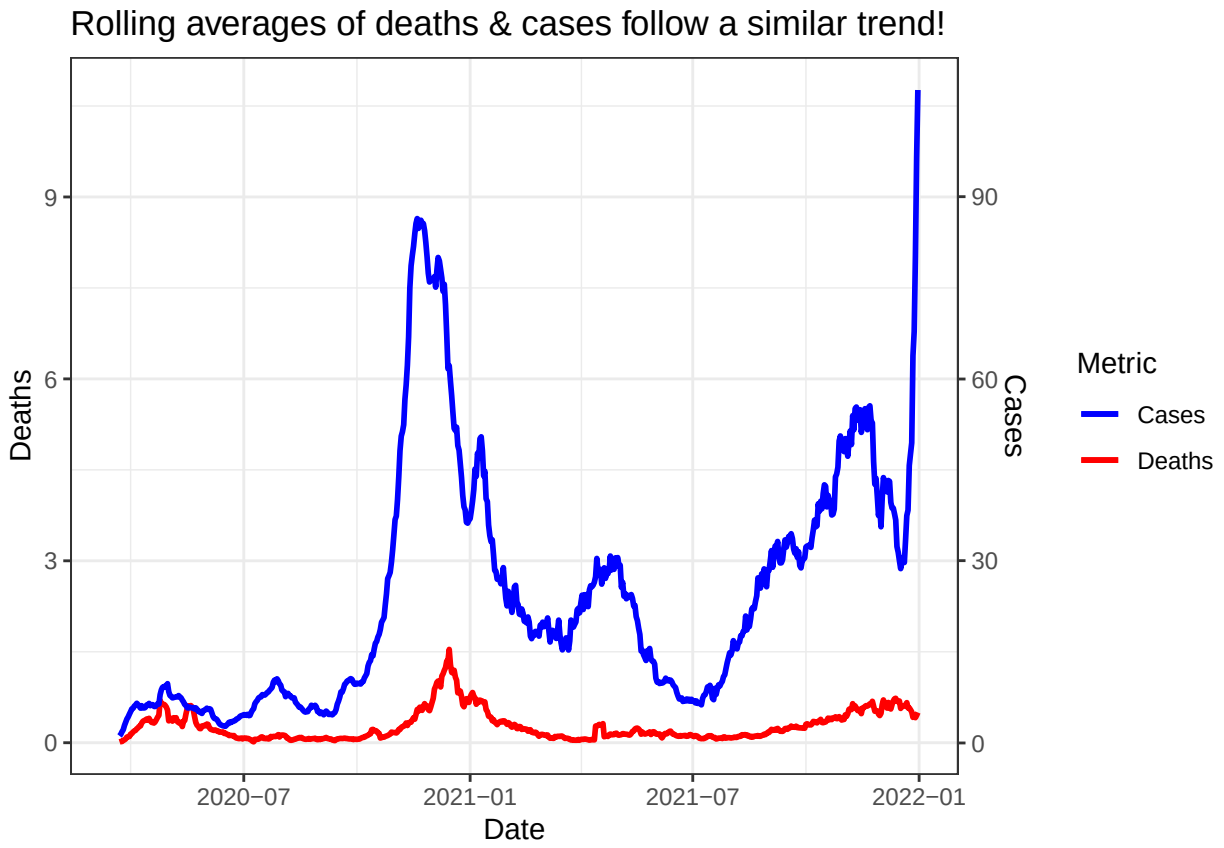
```
## # A tibble: 657 x 10
## # Groups:   state [1]
##    state date       total_deaths total_cases  year total_pop deaths_per_100k
```

```
##    <chr> <date>            <dbl>       <dbl> <dbl>     <dbl>           <dbl>
##  1 Colo~ 2020-03-15            2         136  2020   5784308          0.0346
##  2 Colo~ 2020-03-16            2         161  2020   5784308          0.0346
##  3 Colo~ 2020-03-17            3         183  2020   5784308          0.0519
##  4 Colo~ 2020-03-18            3         216  2020   5784308          0.0519
##  5 Colo~ 2020-03-19            5         278  2020   5784308          0.0864
##  6 Colo~ 2020-03-20            5         364  2020   5784308          0.0864
##  7 Colo~ 2020-03-21            6         475  2020   5784308          0.104
##  8 Colo~ 2020-03-22            7         591  2020   5784308          0.121
##  9 Colo~ 2020-03-23           10         721  2020   5784308          0.173
## 10 Colo~ 2020-03-24           11         912  2020   5784308          0.190
## # ... with 647 more rows, and 3 more variables: cases_per_100k <dbl>,
## #   deaths_7_day <dbl>, cases_7_day <dbl>
```

```r
# setting a coefficient to visualize the graphs and scales better
coeff <- 10

# plotting a line graph of rolling averages of cases vs deaths
ggplot(rolling_colorado_df, aes(x = date)) +
  geom_line(aes(y = deaths_7_day, color = "Deaths"), size = 1) +  # Raw scores as bar chart
  geom_line(aes(y = cases_7_day/coeff, color = "Cases"), size = 1) +  # Rolling average as line
  labs(x = "Date", color = "Metric") +
  scale_color_manual(values = c("Deaths" = "red", "Cases" = "blue")) +
  theme_bw() +
  scale_y_continuous(
        name = "Deaths",
        #labels = unit_format(unit = "M", scale = 100e-6),
        sec.axis = sec_axis(~.*10, name = "Cases")  # Adjust the transformation
    ) +
    ggtitle("Rolling averages of deaths & cases follow a similar trend!")
```

```
## Warning: Removed 7 row(s) containing missing values (geom_path).
```

```
## Warning: Removed 7 row(s) containing missing values (geom_path).
```




```
# 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
```

– Communicate your methodology, results, and interpretation here –

After calculating the rolling averages for deaths and cases for the state of Colorado, the same as been plotted onto a line graph. They seem to follow a similar pattern as expected - when cases go up, deaths also go up. Colorado seems to not have been affected as badly as the other states in the list as per Q2. Relatively, based on it's population, it was not in the top 10.

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

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

# getting the total deaths and cases arranged by deaths in descending order
county_d_df <- combined_df %>%
  filter(state == "Colorado", date >= "2020-03-15", date <= "2021-12-20") %>%
  group_by(county, date) %>%
  summarise(total_deaths = sum(deaths),
            total_cases = sum(cases)) %>%
  arrange(desc(total_deaths), desc(total_cases)) %>%
  distinct(county, .keep_all = TRUE) %>%
  mutate(year = year(date))
```

```
## `summarise()` regrouping output by 'county' (override with `.groups` argument)
```

```r
# using another df to get the fips column
county_fips_df <- combined_df %>%
  filter(state == "Colorado", date >= "2020-03-15", date <= "2021-12-20") %>%
  select(county, fips) %>%
  distinct(county, .keep_all = TRUE)

# joining the above two tables to get all the required columns arranged by deaths
county_d_df <- county_d_df %>%
  left_join(county_fips_df, by = "county") %>%
  select(county, date, fips, everything())

# getting the total deaths and cases arranged by cases in descending order
county_c_df <- combined_df %>%
  filter(state == "Colorado", date >= "2020-03-15", date <= "2021-12-20") %>%
  group_by(county, date) %>%
  summarise(total_deaths = sum(deaths),
            total_cases = sum(cases)) %>%
  arrange(desc(total_cases), desc(total_deaths)) %>%
  distinct(county, .keep_all = TRUE) %>%
  mutate(year = year(date))
```

```
## `summarise()` regrouping output by 'county' (override with `.groups` argument)
```

```r
# joining the above two tables to get all the required columns arranged by cases
county_c_df <- county_c_df %>%
  left_join(county_fips_df, by = "county") %>%
  select(county, date, fips, everything())

# getting the population details for each of the counties of Colorado
county_pop_df <- us_population_estimates %>%
  filter(STNAME == "Colorado") %>%
  separate(CTYNAME, into = c("county"), sep = " County")
```

```
## Warning: Expected 1 pieces. Additional pieces discarded in 128 rows [1, 2, 3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```r
# calculating the deaths and cases per 100k for the df that is arranged by deaths
county_d_df <- county_d_df %>%
  left_join(county_pop_df, by = c("county", "year" = "Year")) %>%
  select(county, date, fips.x, total_deaths, total_cases, year, Estimate) %>%
  rename("fips" = "fips.x") %>%
  mutate(deaths_per_100k = (total_deaths/Estimate)*100000,
```

```
         cases_per_100k = (total_cases/Estimate)*100000)

# calculating the deaths and cases per 100k for the df that is arranged by cases
county_c_df <- county_c_df %>%
  left_join(county_pop_df, by = c("county", "year" = "Year")) %>%
  select(county, date, fips.x, total_deaths, total_cases, year, Estimate) %>%
  rename("fips" = "fips.x") %>%
  mutate(deaths_per_100k = (total_deaths/Estimate)*100000,
         cases_per_100k = (total_cases/Estimate)*100000)

# viewing the data
county_d_df
```

```
## # A tibble: 65 x 9
## # Groups:   county [65]
##    county date       fips  total_deaths total_cases  year Estimate
##    <chr>  <date>     <chr>        <dbl>       <dbl> <dbl>    <dbl>
##  1 El Pa~ 2021-12-20 08041         1355      119772  2021   737867
##  2 Denver 2021-12-20 08031         1065      106747  2021   711463
##  3 Jeffe~ 2021-12-20 08059         1061       76732  2021   579581
##  4 Adams  2021-12-20 08001         1057       90476  2021   522140
##  5 Arapa~ 2021-12-20 08005         1046       95769  2021   654900
##  6 Pueblo 2021-12-20 08101          643       30739  2021   169622
##  7 Weld   2021-12-20 08123          569       55599  2021   340036
##  8 Mesa   2021-12-20 08077          445       29542  2021   157335
##  9 Larim~ 2021-12-20 08069          393       47444  2021   362533
## 10 Dougl~ 2021-12-20 08035          361       48740  2021   368990
## # ... with 55 more rows, and 2 more variables: deaths_per_100k <dbl>,
## #   cases_per_100k <dbl>
```

```
county_c_df
```

```
## # A tibble: 65 x 9
## # Groups:   county [65]
##    county date       fips  total_deaths total_cases  year Estimate
##    <chr>  <date>     <chr>        <dbl>       <dbl> <dbl>    <dbl>
##  1 El Pa~ 2021-12-20 08041         1355      119772  2021   737867
##  2 Denver 2021-12-20 08031         1065      106747  2021   711463
##  3 Arapa~ 2021-12-20 08005         1046       95769  2021   654900
##  4 Adams  2021-12-20 08001         1057       90476  2021   522140
##  5 Jeffe~ 2021-12-20 08059         1061       76732  2021   579581
##  6 Weld   2021-12-20 08123          569       55599  2021   340036
##  7 Dougl~ 2021-12-20 08035          361       48740  2021   368990
##  8 Larim~ 2021-12-20 08069          393       47444  2021   362533
##  9 Bould~ 2021-12-20 08013          323       36754  2021   329543
## 10 Pueblo 2021-12-20 08101          643       30739  2021   169622
## # ... with 55 more rows, and 2 more variables: deaths_per_100k <dbl>,
## #   cases_per_100k <dbl>
```

```
# 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
```

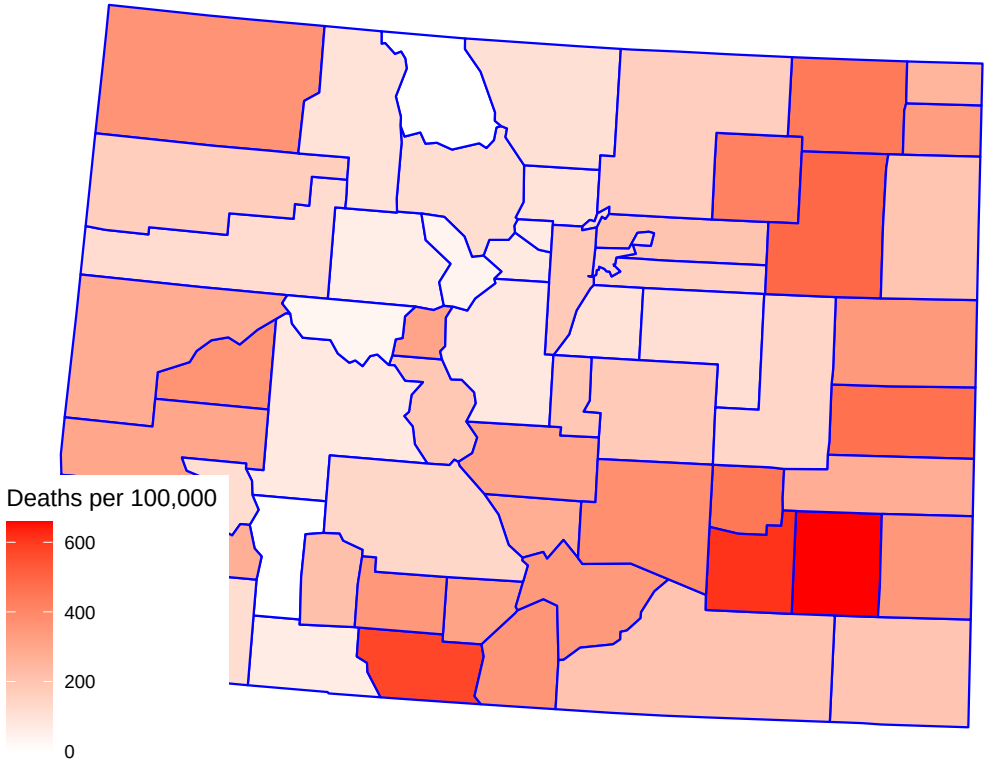
– Communicate your methodology, results, and interpretation here –

To get the required tables county wise, I have grouped the combined_df using county and date after adding the necessary filters. For some reason, I could not get the distinct values of county after arranging them when grouping by fips as well, hence I have extracted the fips in a separate df and performed a left join to add the fips column in the end result as well. Then I have preprocessed the us population dataset for the counties of Colorado and used the population to calculate the deaths and cases per 100,000 people.

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

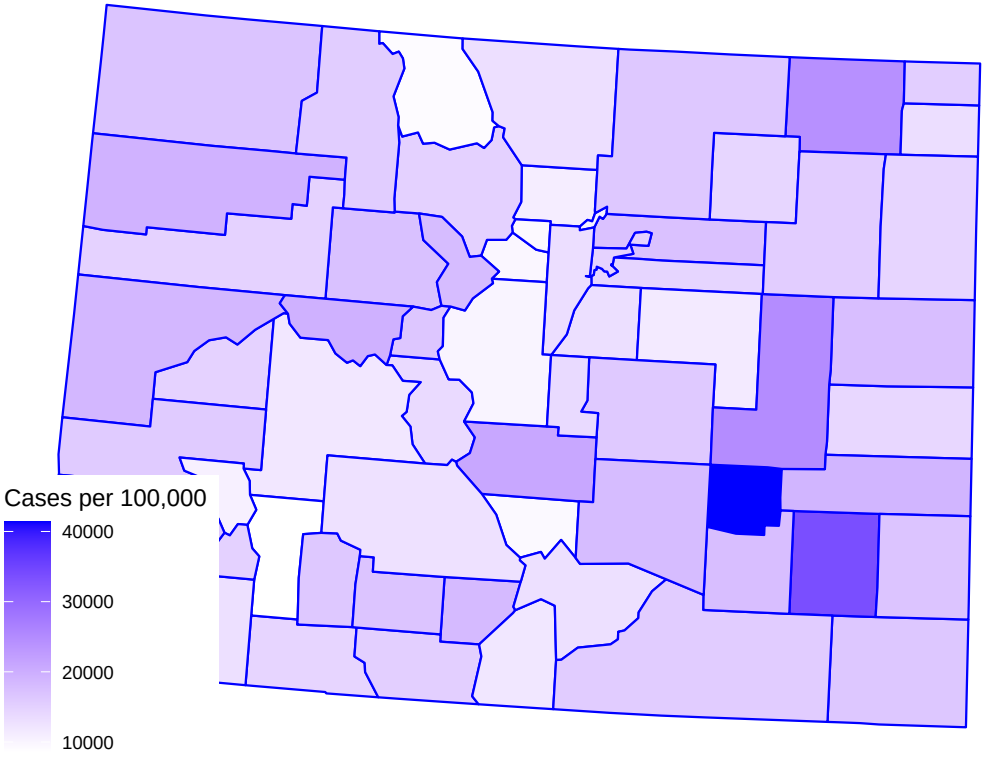
```
# plotting the deaths_per_100k for each county of Colorado using maps
plot_usmap(regions = "county", include="CO", data = county_d_df, values = "deaths_per_100k", color = "bl
  scale_fill_continuous(low = "white", high = "red", name = "Deaths per 100,000") +
  ggtitle("Deaths per 100,000 in Colorado Counties")
```

Deaths per 100,000 in Colorado Counties




```
# plotting the cases_per_100k for each county of Colorado using maps
plot_usmap(regions = "county", include="CO", data = county_c_df, values = "cases_per_100k", color = "blu
  scale_fill_continuous(low = "white", high = "blue", name = "Cases per 100,000") +
  ggtitle("Cases per 100,000 in Colorado Counties")
```

Cases per 100,000 in Colorado Counties




```
# 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")
```

– Communicate your methodology, results, and interpretation here –

It looks like the County Oteri, though its low population, was severely affected both by terms of the total cases and deaths that occurred here. Other counties such as Crowley and Bent were also highly affected given their low population when taking into consideration of the metric per 100,000 people.

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.

```
# selecting the state of California and obtaining sum of total deaths and cases
california_df <- combined_df %>%
  filter(state == 'California', date >= "2020-03-15", date <= "2021-12-31") %>%
  group_by(state, date) %>%
  summarise(total_deaths = sum(deaths),
            total_cases = sum(cases)) %>%
```

```
  mutate(year = year(date))
```

```
## `summarise()` regrouping output by 'state' (override with `.groups` argument)
```

```
# using tha above information and after performing a join with its population data - calculating the ro
rolling_california_df <- california_df %>%
    left_join(state_pop_df, by = c("state" = "STNAME", "year" = "Year")) %>%
    mutate(deaths_per_100k = (total_deaths/total_pop)*100000,
           cases_per_100k = (total_cases/total_pop)*100000,
           deaths_7_day = (rollapply(c(NA, diff(total_deaths)), width = 7, FUN = mean, align = "right",
           cases_7_day = (rollapply(c(NA, diff(total_cases)), width = 7, FUN = mean, align = "right", fi

# selecting the state of New York and obtaining sum of total deaths and cases
newyork_df <- combined_df %>%
  filter(state == 'New York', date >= "2020-03-15", date <= "2021-12-31") %>%
  group_by(state, date) %>%
  summarise(total_deaths = sum(deaths),
            total_cases = sum(cases)) %>%
  mutate(year = year(date))
```

```
## `summarise()` regrouping output by 'state' (override with `.groups` argument)
```

```
# using tha above information and after performing a join with its population data - calculating the ro
rolling_newyork_df <- newyork_df %>%
    left_join(state_pop_df, by = c("state" = "STNAME", "year" = "Year")) %>%
    mutate(deaths_per_100k = (total_deaths/total_pop)*100000,
           cases_per_100k = (total_cases/total_pop)*100000,
           deaths_7_day = (rollapply(c(NA, diff(total_deaths)), width = 7, FUN = mean, align = "right",
           cases_7_day = (rollapply(c(NA, diff(total_cases)), width = 7, FUN = mean, align = "right", fi

# selecting the state of Minnesota and obtaining sum of total deaths and cases
minnesota_df <- combined_df %>%
  filter(state == 'Minnesota', date >= "2020-03-15", date <= "2021-12-31") %>%
  group_by(state, date) %>%
  summarise(total_deaths = sum(deaths),
            total_cases = sum(cases)) %>%
  mutate(year = year(date))
```

```
## `summarise()` regrouping output by 'state' (override with `.groups` argument)
```

```
# using tha above information and after performing a join with its population data - calculating the ro
rolling_minnesota_df <- minnesota_df %>%
    left_join(state_pop_df, by = c("state" = "STNAME", "year" = "Year")) %>%
    mutate(deaths_per_100k = (total_deaths/total_pop)*100000,
           cases_per_100k = (total_cases/total_pop)*100000,
           deaths_7_day = (rollapply(c(NA, diff(total_deaths)), width = 7, FUN = mean, align = "right",
           cases_7_day = (rollapply(c(NA, diff(total_cases)), width = 7, FUN = mean, align = "right", fi

# viewing the data
rolling_colorado_df
```

```
## # A tibble: 657 x 10
## # Groups:   state [1]
##    state date       total_deaths total_cases  year total_pop deaths_per_100k
##    <chr> <date>            <dbl>       <dbl> <dbl>     <dbl>           <dbl>
##  1 Colo~ 2020-03-15            2         136  2020   5784308          0.0346
```

```
##  2 Colo~ 2020-03-16            2         161  2020   5784308          0.0346
##  3 Colo~ 2020-03-17            3         183  2020   5784308          0.0519
##  4 Colo~ 2020-03-18            3         216  2020   5784308          0.0519
##  5 Colo~ 2020-03-19            5         278  2020   5784308          0.0864
##  6 Colo~ 2020-03-20            5         364  2020   5784308          0.0864
##  7 Colo~ 2020-03-21            6         475  2020   5784308          0.104
##  8 Colo~ 2020-03-22            7         591  2020   5784308          0.121
##  9 Colo~ 2020-03-23           10         721  2020   5784308          0.173
## 10 Colo~ 2020-03-24           11         912  2020   5784308          0.190
## # ... with 647 more rows, and 3 more variables: cases_per_100k <dbl>,
## #   deaths_7_day <dbl>, cases_7_day <dbl>
```

```
rolling_california_df
```

```
## # A tibble: 657 x 10
## # Groups:   state [1]
##    state date       total_deaths total_cases  year total_pop deaths_per_100k
##    <chr> <date>            <dbl>       <dbl> <dbl>     <dbl>           <dbl>
##  1 Cali~ 2020-03-15            6         478  2020  39499738          0.0152
##  2 Cali~ 2020-03-16           11         588  2020  39499738          0.0278
##  3 Cali~ 2020-03-17           14         732  2020  39499738          0.0354
##  4 Cali~ 2020-03-18           17         893  2020  39499738          0.0430
##  5 Cali~ 2020-03-19           19        1067  2020  39499738          0.0481
##  6 Cali~ 2020-03-20           24        1283  2020  39499738          0.0608
##  7 Cali~ 2020-03-21           28        1544  2020  39499738          0.0709
##  8 Cali~ 2020-03-22           35        1851  2020  39499738          0.0886
##  9 Cali~ 2020-03-23           39        2240  2020  39499738          0.0987
## 10 Cali~ 2020-03-24           52        2644  2020  39499738          0.132
## # ... with 647 more rows, and 3 more variables: cases_per_100k <dbl>,
## #   deaths_7_day <dbl>, cases_7_day <dbl>
```

```
rolling_newyork_df
```

```
## # A tibble: 657 x 10
## # Groups:   state [1]
##    state date       total_deaths total_cases  year total_pop deaths_per_100k
##    <chr> <date>            <dbl>       <dbl> <dbl>     <dbl>           <dbl>
##  1 New ~ 2020-03-15            6         732  2020  20154933          0.0298
##  2 New ~ 2020-03-16           10         950  2020  20154933          0.0496
##  3 New ~ 2020-03-17           18        1375  2020  20154933          0.0893
##  4 New ~ 2020-03-18           32        2387  2020  20154933          0.159
##  5 New ~ 2020-03-19           39        4161  2020  20154933          0.194
##  6 New ~ 2020-03-20           68        7113  2020  20154933          0.337
##  7 New ~ 2020-03-21           95       10371  2020  20154933          0.471
##  8 New ~ 2020-03-22          142       15188  2020  20154933          0.705
##  9 New ~ 2020-03-23          183       20899  2020  20154933          0.908
## 10 New ~ 2020-03-24          264       25704  2020  20154933          1.31
## # ... with 647 more rows, and 3 more variables: cases_per_100k <dbl>,
## #   deaths_7_day <dbl>, cases_7_day <dbl>
```

```
rolling_minnesota_df
```

```
## # A tibble: 657 x 10
## # Groups:   state [1]
##    state date       total_deaths total_cases  year total_pop deaths_per_100k
##    <chr> <date>            <dbl>       <dbl> <dbl>     <dbl>           <dbl>
```

```
##  1 Minn~ 2020-03-15            0          35  2020   5707165          0
##  2 Minn~ 2020-03-16            0          54  2020   5707165          0
##  3 Minn~ 2020-03-17            0          60  2020   5707165          0
##  4 Minn~ 2020-03-18            0          77  2020   5707165          0
##  5 Minn~ 2020-03-19            0          89  2020   5707165          0
##  6 Minn~ 2020-03-20            0         115  2020   5707165          0
##  7 Minn~ 2020-03-21            1         138  2020   5707165          0.0175
##  8 Minn~ 2020-03-22            1         171  2020   5707165          0.0175
##  9 Minn~ 2020-03-23            1         235  2020   5707165          0.0175
## 10 Minn~ 2020-03-24            1         264  2020   5707165          0.0175
## # ... with 647 more rows, and 3 more variables: cases_per_100k <dbl>,
## #   deaths_7_day <dbl>, cases_7_day <dbl>
```

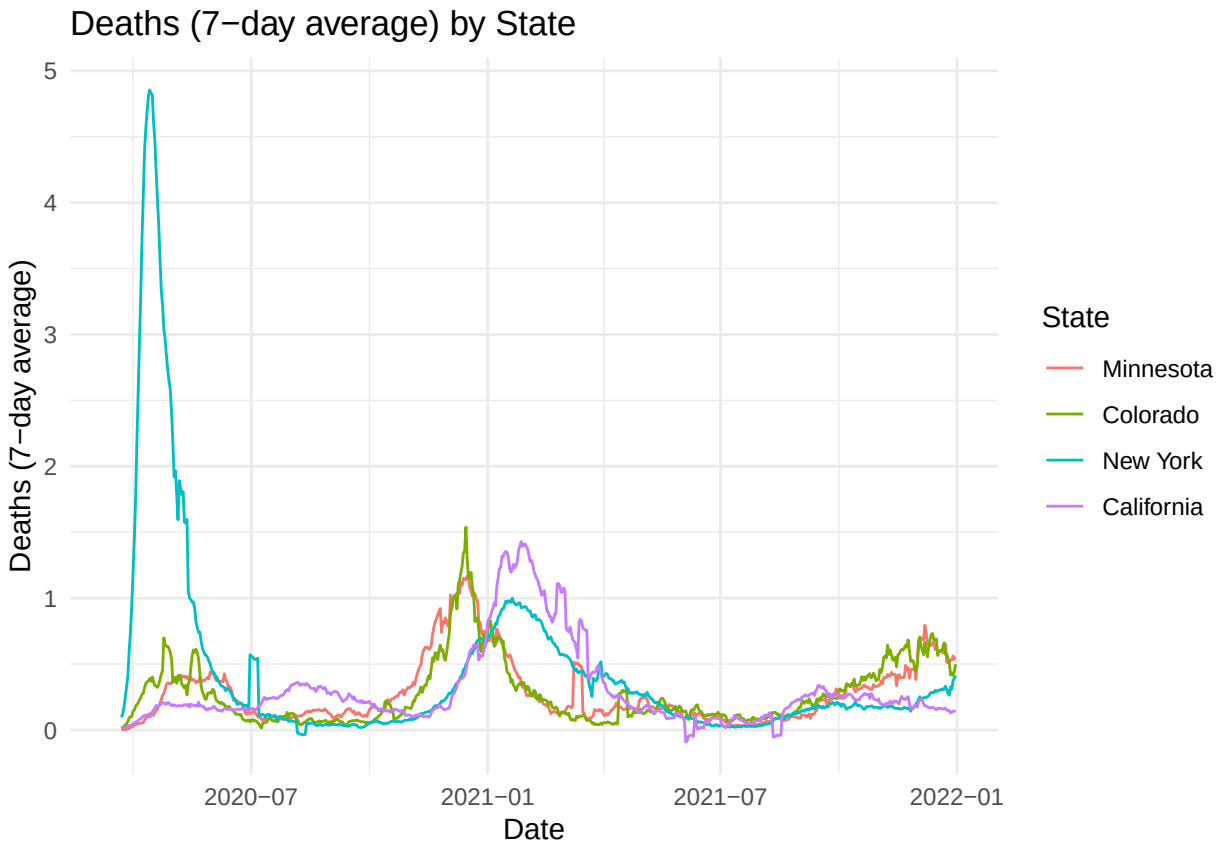
– Communicate your methodology, results, and interpretation here –

Following the same method that was followed for the state of Colorado in Question 3. The other three states that are included are California, New York and Minnesota. Picked California and New York for their huge population and Colorado and Minnesota for their relatively lesser population. This is done to see how population plays a role in rolling 7 day average metrics for deaths and cases.

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.

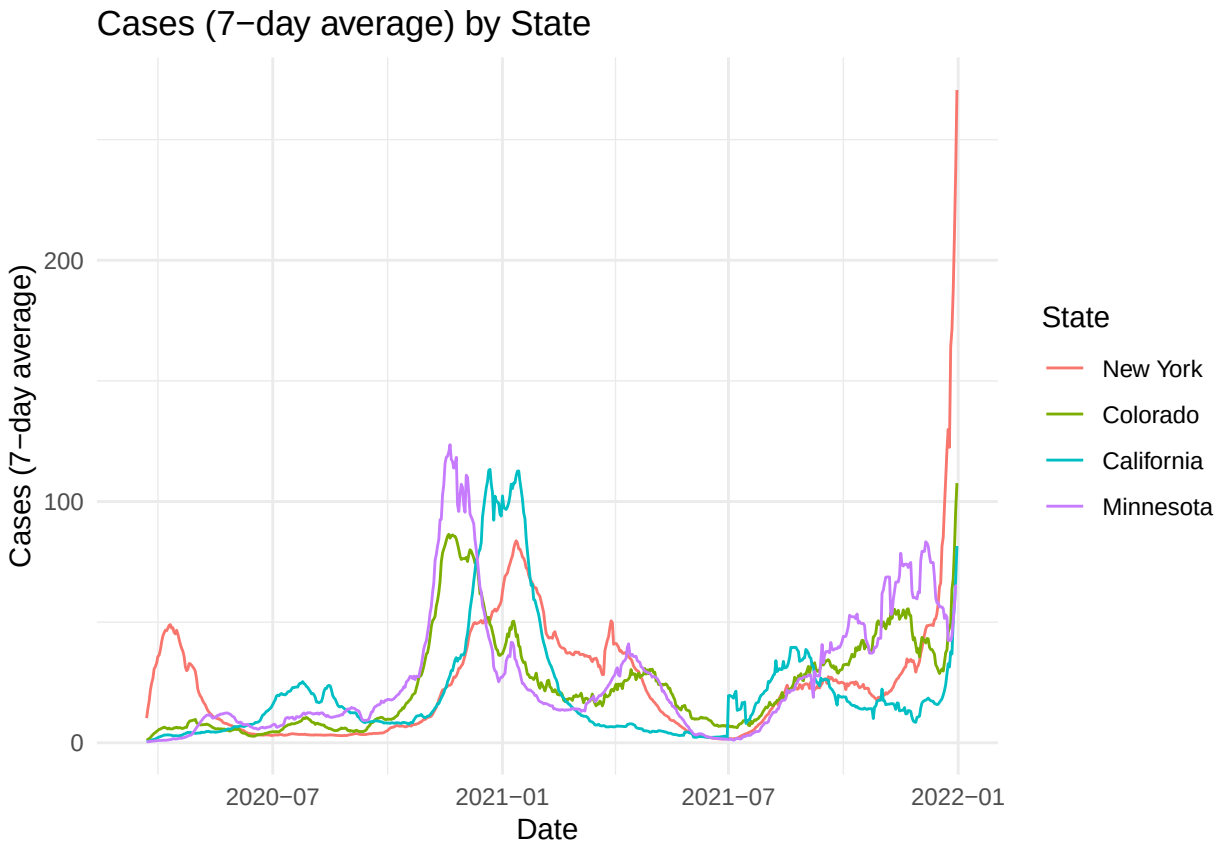
```
# combining the previous dataframes for each state into one df
combined_rolling_state_df <- rbind(rolling_colorado_df, rolling_california_df, rolling_newyork_df, rolli

# visualizing the rolling 7 days average for deaths for each state
ggplot(combined_rolling_state_df, aes(x = date, y = deaths_7_day, color = fct_reorder2(state, date, dea
  geom_line() +
  labs(x = "Date", y = "Deaths (7-day average)", color = "State") +
  ggtitle("Deaths (7-day average) by State") +
  theme_minimal()
```

```
## Warning: Removed 28 row(s) containing missing values (geom_path).
```




```
# visualizing the rolling 7 days average for cases for each state
ggplot(combined_rolling_state_df, aes(x = date, y = cases_7_day, color = fct_reorder2(state, date, cases
  geom_line() +
  labs(x = "Date", y = "Cases (7-day average)", color = "State") +
  ggtitle("Cases (7-day average) by State") +
  theme_minimal()
```

```
## Warning: Removed 28 row(s) containing missing values (geom_path).
```



– Communicate your methodology, results, and interpretation here –

It looks like at the beginning, New York had the highest number of deaths per 100,000 people when compared to the other 3 states and Minnesota was leading at the end. In terms of cases per 100,000 people, New York was leading at the beginning and at the end. This is mostly due to the high population that is present in this state. fct_reorder2 is used so that it becomes easier to visualize which state is at the top and also align the labels accordingly.

```
# 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

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   `Province/State` = col_character(),
##   `Country/Region` = col_character()
## )

## See spec(...) for full column specifications.
```

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

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
```

```
##   `Province/State` = col_character(),
##   `Country/Region` = col_character()
## )
## See spec(...) for full column specifications.
```

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

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   iso2 = col_character(),
##   iso3 = col_character(),
##   Admin2 = col_character(),
##   Province_State = col_character(),
##   Country_Region = col_character(),
##   Combined_Key = col_character()
## )
## See spec(...) for full column specifications.
```

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

```
## Parsed with column specification:
## cols(
##   .default = col_double(),
##   iso2 = col_character(),
##   iso3 = col_character(),
##   Admin2 = col_character(),
##   Province_State = col_character(),
##   Country_Region = col_character(),
##   Combined_Key = col_character()
## )
## See spec(...) for full column specifications.
```

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

```
## Parsed with column specification:
## cols(
##   `Country Name` = col_character(),
##   `Country Code` = col_character(),
##   `Series Name` = col_character(),
##   `Series Code` = col_character(),
##   `2020 [YR2020]` = col_character(),
##   `2021 [YR2021]` = col_character()
## )
```

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

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

# obtaining the cases data from csse data
csse_colorado_c_df <- csse_us_cases %>%
  pivot_longer(cols = -c(1:11), names_to = "year", values_to = "cases") %>%
  filter(Province_State == "Colorado") %>%
  mutate(date = as.Date(year, format = "%m/%d/%y")) %>%
```

```
  select(FIPS, Admin2, Province_State, date, cases) %>%
  rename("fips" = "FIPS", "county" = "Admin2", "state" = "Province_State") %>%
  filter(date >= "2020-03-15", date <= "2021-12-31")

# obtaining the deaths data from csse data
csse_colorado_d_df <- csse_us_deaths %>%
  pivot_longer(cols = -c(1:11), names_to = "year", values_to = "deaths") %>%
  filter(Province_State == "Colorado") %>%
  mutate(date = as.Date(year, format = "%m/%d/%y")) %>%
  select(FIPS, Admin2, Province_State, date, deaths) %>%
  rename("fips" = "FIPS", "county" = "Admin2", "state" = "Province_State") %>%
  filter(date >= "2020-03-15", date <= "2021-12-31")

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

# performing a join to combine the above two datasets
csse_colorado_df <- csse_colorado_c_df %>%
  left_join(csse_colorado_d_df)
```

```
## Joining, by = c("fips", "county", "state", "date")
```

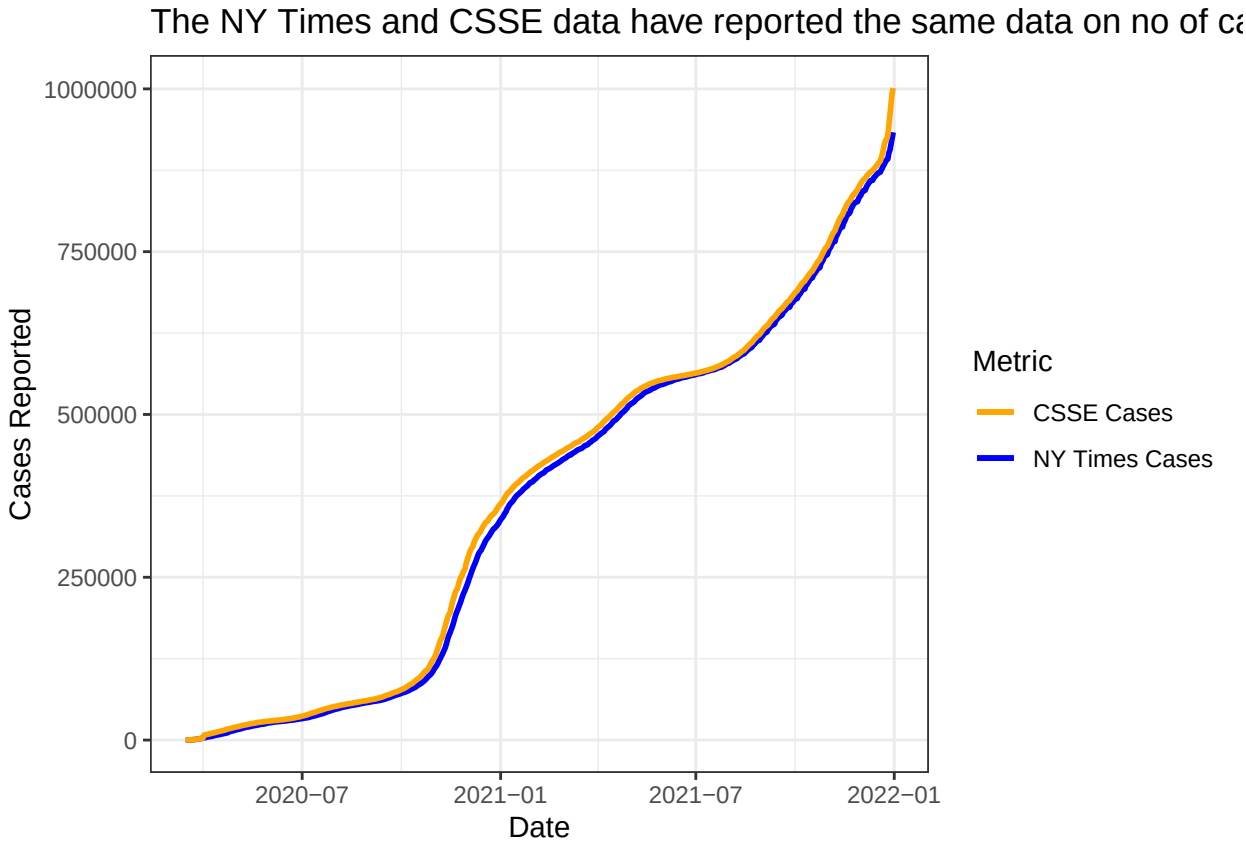
```
# summing up cases and deaths on a state and date level
csse_colorado_total_df <- csse_colorado_df %>%
  group_by(state, date) %>%
  summarise(CSSE_cases = sum(cases),
            CSSE_deaths = sum(deaths))
```

```
## `summarise()` regrouping output by 'state' (override with `.groups` argument)
```

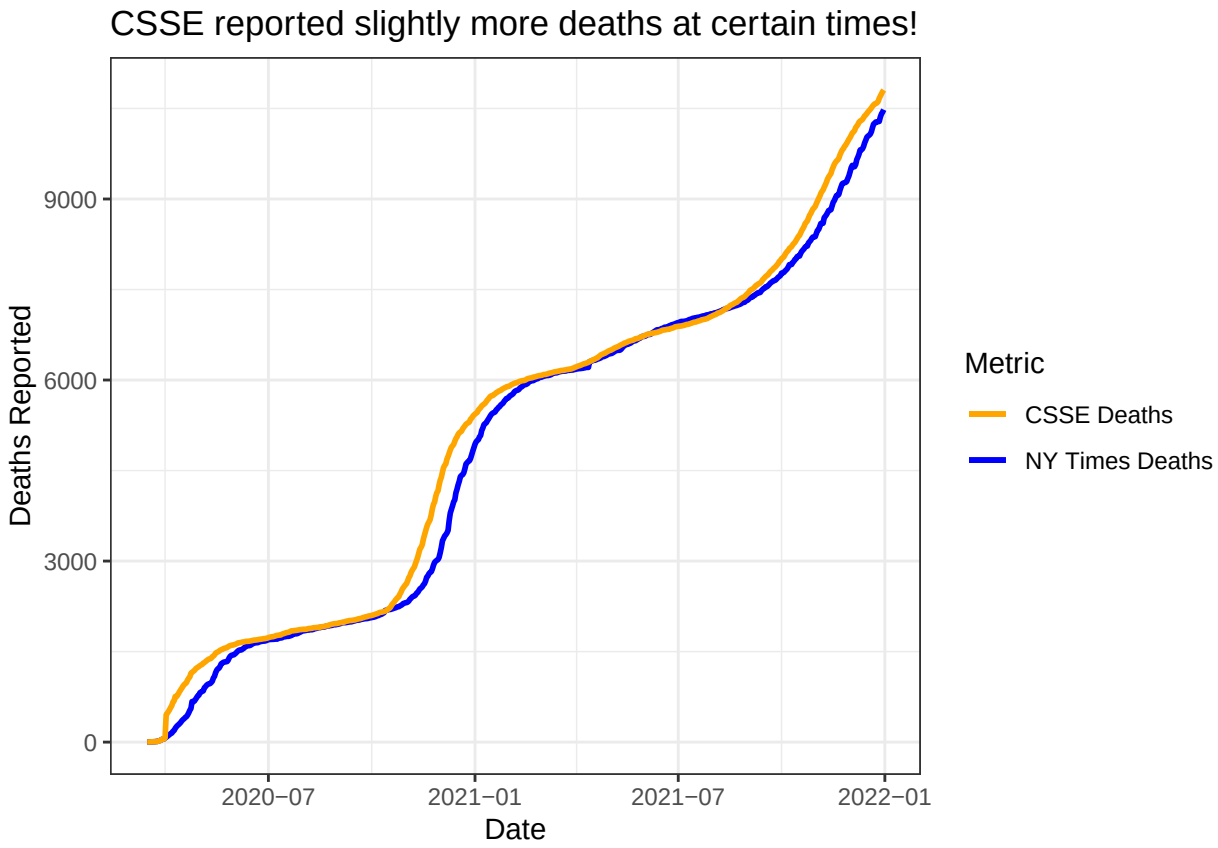
```
# renaming columns for convenience
colorado_df <- colorado_df %>%
  rename("NY_Times_cases" = "total_cases", "NY_Times_deaths" = "total_deaths")

final_colorado_df <- colorado_df %>%
  full_join(csse_colorado_total_df)
```

```
## Joining, by = c("state", "date")
```

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

# plotting a line graph for comparing the NY Times vs CSSE cases data
ggplot(final_colorado_df, aes(x = date)) +
  geom_line(aes(y = NY_Times_cases, color = "NY Times Cases"), size = 1) +
  geom_line(aes(y = CSSE_cases, color = "CSSE Cases"), size = 1) +
  labs(x = "Date", y = "Cases Reported", color = "Metric") +
  scale_color_manual(values = c("CSSE Cases" = "orange", "NY Times Cases" = "blue")) +
  theme_bw() +
  ggtitle("The NY Times and CSSE data have reported the same data on no of cases!")
```




```
# plotting a line graph for comparing the NY Times vs CSSE deaths data
ggplot(final_colorado_df, aes(x = date)) +
  geom_line(aes(y = NY_Times_deaths, color = "NY Times Deaths"), size = 1) +
  geom_line(aes(y = CSSE_deaths, color = "CSSE Deaths"), size = 1) +
  labs(x = "Date", y = "Deaths Reported", color = "Metric") +
  scale_color_manual(values = c("CSSE Deaths" = "orange", "NY Times Deaths" = "blue")) +
  theme_bw() +
  ggtitle("CSSE reported slightly more deaths at certain times!")
```




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

– Communicate your methodology, results, and interpretation here –

When comparing the NY Times and CSSE data, it looks like both reported more or less the same numbers wrt the number of cases during this duration. But when it comes to the deaths, CSSE has reported slightly more than NY Times for the Colorado State, especially during the beginning, end of 2020 and towards the end of 2021. These were also the moments when the number of deaths and cases were increasing according to the data reported by both of them. During such times, it can be expected that the data given by both may not be similar due to some underestimation and overestimation that can easily occur.

Question 2 Now that you have verified the data reported from the CSSE and NY Times are similar, combine the global and US CSSE data sets and identify the top 10 countries in terms of deaths and cases per

100,000 people between March 15, 2020, and December 31, 2021.

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

# tidying up the global cases dataset
global_cases_df <- csse_global_cases %>%
  pivot_longer(cols = -c(1:4), names_to = "year", values_to = "cases") %>%
  mutate(date = as.Date(year, format = "%m/%d/%y")) %>%
  rename("country" = "Country/Region") %>%
  filter(date >= "2020-03-15", date <= "2021-12-31") %>%
  mutate(year = year(date))

# tidying up the global deaths dataset
global_deaths_df <- csse_global_deaths %>%
  pivot_longer(cols = -c(1:4), names_to = "year", values_to = "deaths") %>%
  mutate(date = as.Date(year, format = "%m/%d/%y")) %>%
  rename("country" = "Country/Region") %>%
  filter(date >= "2020-03-15", date <= "2021-12-31") %>%
  mutate(year = year(date))

# tidying up the global population dataset
global_pop_df <- globabl_population_estimates %>%
  pivot_longer(cols = -c(1:4), names_to = "year", values_to = "population") %>%
  separate(year, into = c("year"), sep = " ") %>%
  rename("country" = "Country Name") %>%
  mutate(year = as.numeric(year),
         population = as.numeric(population))
```

```
## Warning: Expected 1 pieces. Additional pieces discarded in 534 rows [1, 2, 3, 4,
## 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, ...].
```

```
## Warning in mask$eval_all_mutate(dots[[i]]): NAs introduced by coercion
```

```
# performing the joins needed to combine above three datasets and caluclating the cases and deaths per
global_combined_df <- global_cases_df %>%
  full_join(global_deaths_df) %>%
  inner_join(global_pop_df) %>%
  select(country, Lat, Long, year, date, cases, deaths, population) %>%
  mutate(deaths_per_100k = (deaths/population)*100000,
         cases_per_100k = (cases/population)*100000)
```

```
## Joining, by = c("Province/State", "country", "Lat", "Long", "year", "date")
```

```
## Joining, by = c("country", "year")
```

```
# viewing the top 10 countries with highest deaths per 100,000 people
top_10_deaths_global_df <- global_combined_df %>%
  arrange(desc(deaths_per_100k)) %>%
  distinct(country, .keep_all = TRUE) %>%
  head(n = 10)

# viewing the top 10 countries with highest cases per 100,000 people
top_10_cases_global_df <- global_combined_df %>%
  arrange(desc(cases_per_100k)) %>%
  distinct(country, .keep_all = TRUE) %>%
```

```
  head(n = 10)

top_10_deaths_global_df
```

```
## # A tibble: 10 x 10
##    country   Lat  Long  year date        cases deaths population deaths_per_100k
##    <chr>   <dbl> <dbl> <dbl> <date>      <dbl>  <dbl>      <dbl>            <dbl>
##  1 Peru    -9.19 -75.0  2021 2021-12-31 2.30e6 202690   33359000             608.
##  2 Bulgar~ 42.7   25.5  2021 2021-12-31 7.47e5  30955    6882000             450.
##  3 Bosnia~ 43.9   17.7  2021 2021-12-31 2.91e5  13442    3263000             412.
##  4 Hungary 47.2   19.5  2021 2021-12-31 1.26e6  39186    9721000             403.
##  5 Moldova 47.4   28.4  2021 2021-12-31 3.76e5  10275    2614000             393.
##  6 Monten~ 42.7   19.4  2021 2021-12-31 1.70e5   2411     621000             388.
##  7 North ~ 41.6   21.7  2021 2021-12-31 2.25e5   7960    2072000             384.
##  8 Georgia 42.3   43.4  2021 2021-12-31 9.35e5  13800    3712000             372.
##  9 Croatia 45.1   15.2  2021 2021-12-31 7.15e5  12538    4025000             312.
## 10 Romania 45.9   25.0  2021 2021-12-31 1.81e6  58752   19156000             307.
## # ... with 1 more variable: cases_per_100k <dbl>
```

```
top_10_cases_global_df
```

```
## # A tibble: 10 x 10
##    country   Lat   Long  year date        cases deaths population
##    <chr>   <dbl>  <dbl> <dbl> <date>      <dbl>  <dbl>      <dbl>
##  1 Andorra 42.5    1.52  2021 2021-12-31 2.37e4    140      77000
##  2 Monten~ 42.7   19.4   2021 2021-12-31 1.70e5   2411     621000
##  3 Georgia 42.3   43.4   2021 2021-12-31 9.35e5  13800    3712000
##  4 Seyche~ -4.68  55.5   2021 2021-12-30 2.48e4    134      99000
##  5 San Ma~ 43.9   12.5   2021 2021-12-31 8.20e3    100      34000
##  6 Sloven~ 46.2   15.0   2021 2021-12-31 4.64e5   5589    2101000
##  7 Mongol~ 46.9  104.    2021 2021-12-31 6.93e5   1986    3329000
##  8 United~ 55.4   -3.44  2021 2021-12-31 1.29e7 177397   67503000
##  9 Lithua~ 55.2   23.9   2021 2021-12-31 5.24e5   7397    2766000
## 10 Serbia  44.0   21.0   2021 2021-12-31 1.30e6  12714    6863000
## # ... with 2 more variables: deaths_per_100k <dbl>, cases_per_100k <dbl>
```

– Communicate your methodology, results, and interpretation here –

When it comes to the highest deaths per 100,000 people, it looks like a lot of European countries suffered the most. A lot of these countries like Bulgaria, Hungary, Moldova, Croatia are on the top 10. This is the same case when it comes to cases per 100,000 people too. Countries in the top 10 include United Kingdom, Serbia, San Marino. It is also surprising to see Seychelles - which is a remote island off the coast of East Africa.

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.

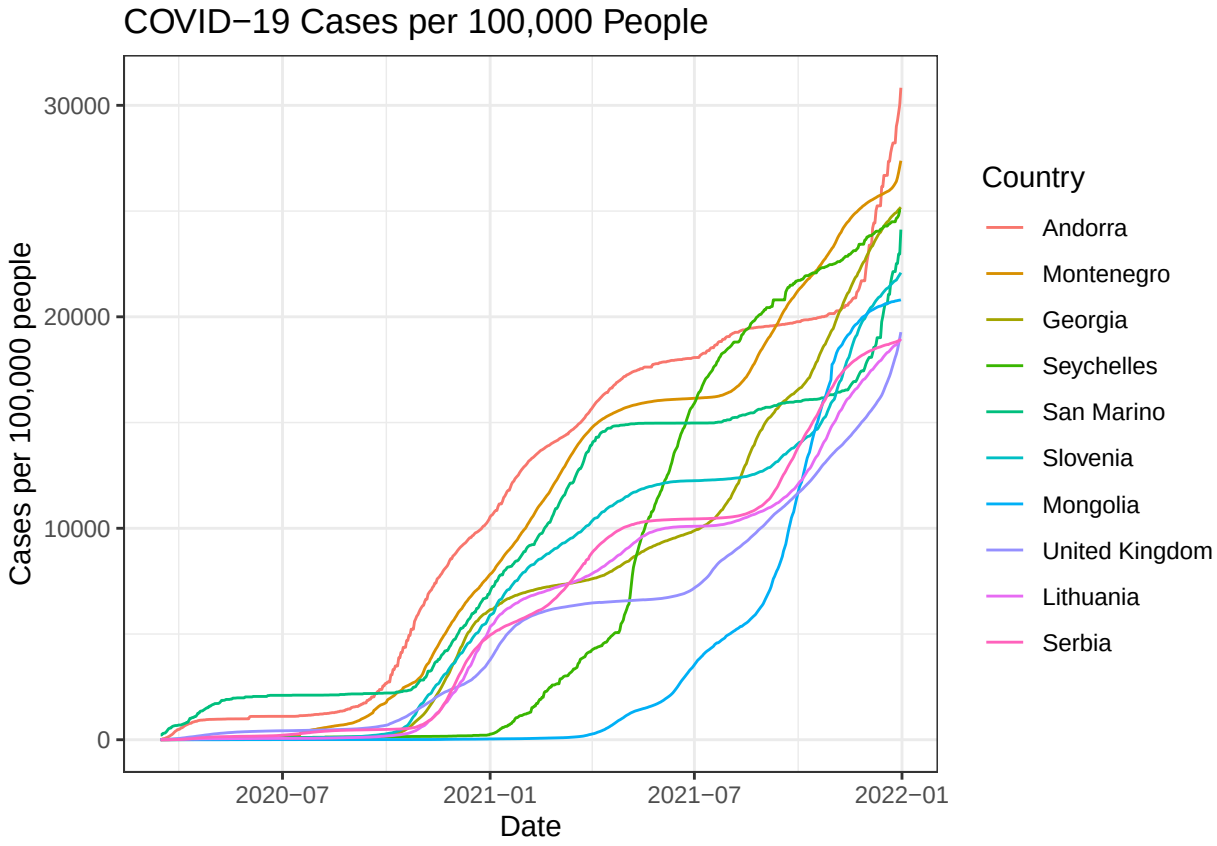
```
# Creating a separate dataset for visualizing cases per 100,000 people
top_10_cases_visual_df <- global_combined_df %>%
    filter(country %in% top_10_cases_global_df$country) %>%
    group_by(country, date) %>%
    summarise(cases_per_100k = sum(cases_per_100k))
```

```
## `summarise()` regrouping output by 'country' (override with `.groups` argument)
```

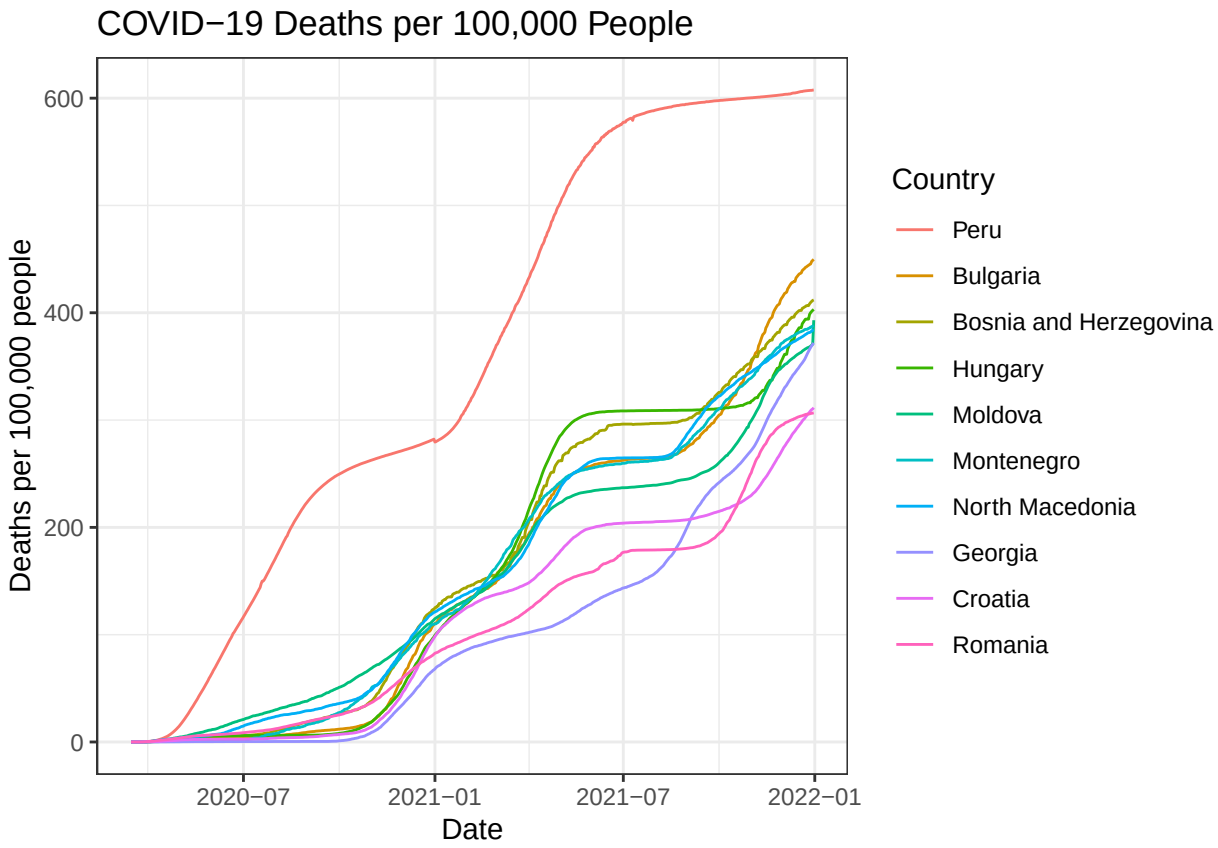
```r
# Creating a separate dataset for visualizing deaths per 100,000 people
top_10_deaths_visual_df <- global_combined_df %>%
    filter(country %in% top_10_deaths_global_df$country) %>%
    group_by(country, date) %>%
    summarise(deaths_per_100k = sum(deaths_per_100k))
```

```
## `summarise()` regrouping output by 'country' (override with `.groups` argument)
```

```r
# visualizing cases for the top 10 countries with highest cases per 100,000 people
ggplot(top_10_cases_visual_df, aes(x = date, y = cases_per_100k, color = fct_reorder2(country, date, cas
  geom_line(size = 0.5) +
  labs(x = "Date", y = "Cases per 100,000 people", color = "Country") +
  ggtitle("COVID-19 Cases per 100,000 People") +
  theme_bw()
```




```r
# visualizing deaths for the top 10 countries with highest deaths per 100,000 people
ggplot(top_10_deaths_visual_df, aes(x = date, y = deaths_per_100k, color = fct_reorder2(country, date, c
  geom_line(size = 0.5) +
  labs(x = "Date", y = "Deaths per 100,000 people", color = "Country") +
  ggtitle("COVID-19 Deaths per 100,000 People") +
  theme_bw()
```

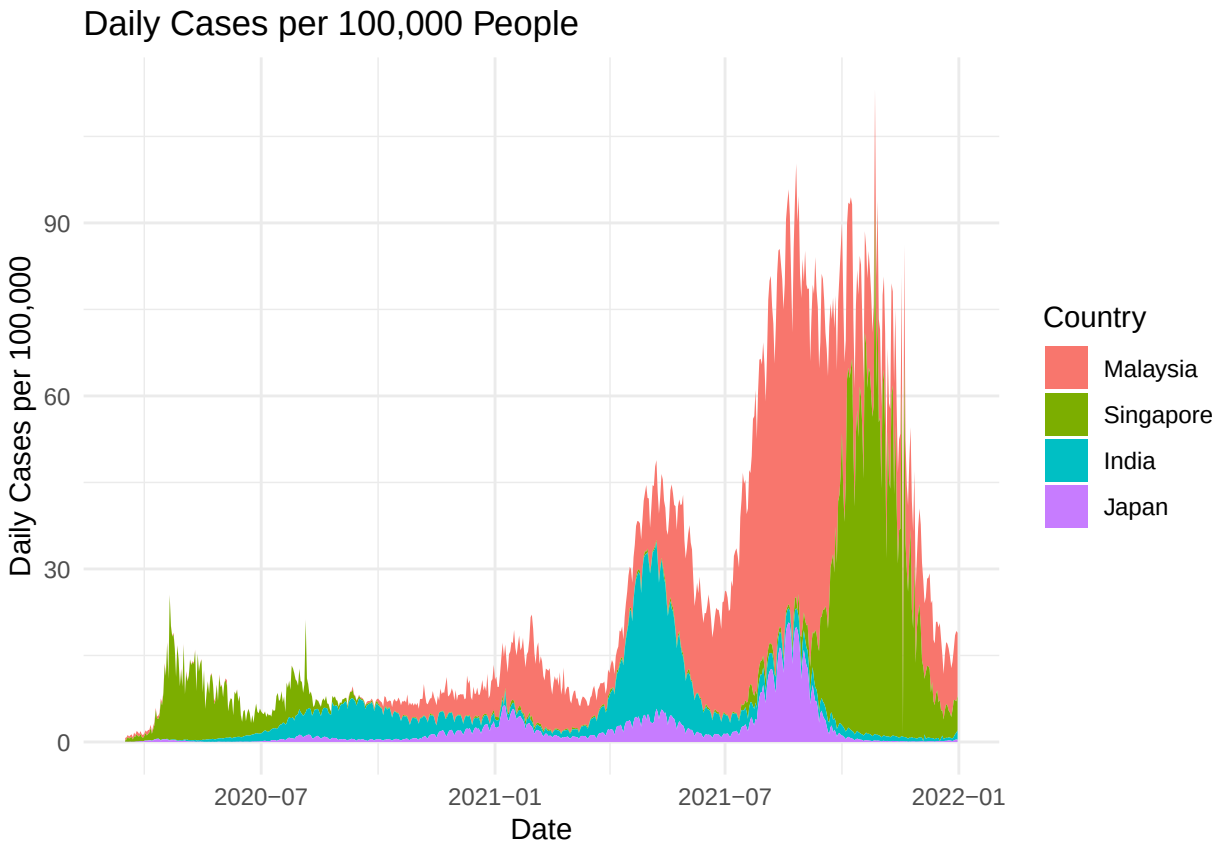


– Communicate your methodology, results, and interpretation here –

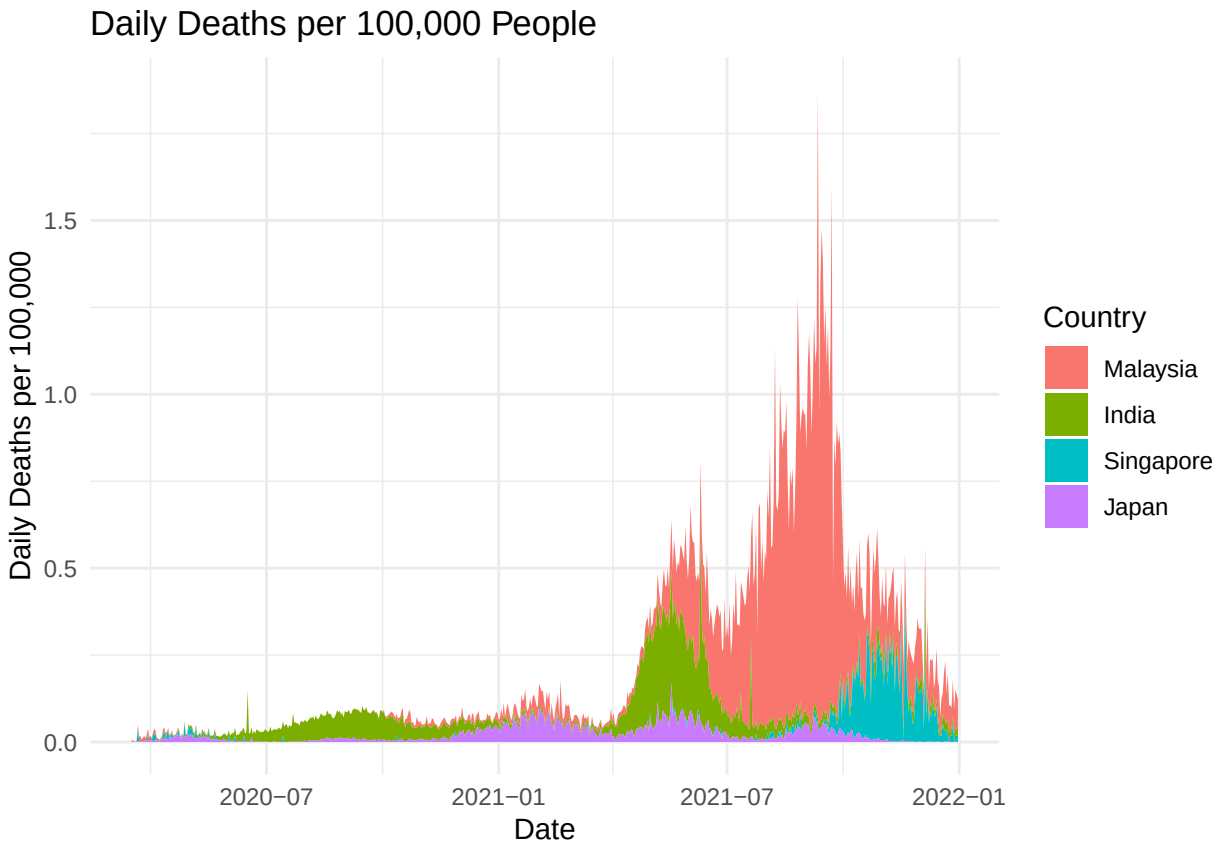
In terms of cases, Andorra leads the list. It ranked consistently at the top throughout the entire duration of 2 years as well. But when it comes to death per 100,000 people, Peru is ranked not just at the top but has a significant difference when compared to the other countries. This could have been due to several factors such as non-speedy distribution of vaccines and overcrowding of certain areas with a lot of homes.

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.

```
asia_df <- global_combined_df %>%
  mutate(daily_cases = c(NA, diff(cases)),
         daily_deaths = c(NA, diff(deaths)),
         daily_cases_per_100k = (daily_cases/population)*100000,
         daily_deaths_per_100k = (daily_deaths/population)*100000) %>%
  filter(country %in% c("India", "Singapore", "Japan", "Malaysia"), daily_cases_per_100k >= 0, daily_dea

ggplot(asia_df, aes(x = date, y = daily_cases_per_100k, fill = fct_reorder2(country, date, daily_cases_
  geom_area() +
  labs(x = "Date", y = "Daily Cases per 100,000", fill = "Country") +
  ggtitle("Daily Cases per 100,000 People") +
  theme_minimal()
```




```
ggplot(asia_df, aes(x = date, y = daily_deaths_per_100k, fill = fct_reorder2(country, date, daily_deaths
  geom_area() +
  labs(x = "Date", y = "Daily Deaths per 100,000", fill = "Country") +
  ggtitle("Daily Deaths per 100,000 People") +
  theme_minimal()
```



– Communicate your methodology, results, and interpretation here –

I've used an area chart to indicate which country has the highest daily deaths and cases per 100,000 people. Countries from Asia - India, Malaysia, Singapore and Japan are considered. When it comes to daily cases, it looks like Malaysia is at the top throughout mostly. Especially towards the very end. This could be the case that this country has a low population and a high case rate for its population. Even for deaths, Malaysia is at the top with India having some significant numbers closer to Aug of 2021.