# 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
```

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

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

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

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

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

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

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

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

```
# Combine and tidy the 2020, 2021, and 2022 COVID data sets.
# Hint: Review the rbind() documentation to combine the three data sets.
#
## YOUR CODE HERE ##

max_date <- '' # replace the quotes with your code to find the most recent date in the data set
us_total_cases <- ''
us_total_deaths <- ''




# Remove Puerto Rico observations
us_counties_2020 <- us_counties_2020[us_counties_2020$state != "Puerto Rico", ]
us_counties_2021 <- us_counties_2021[us_counties_2021$state != "Puerto Rico", ]
us_counties_2022 <- us_counties_2022[us_counties_2022$state != "Puerto Rico", ]

# Combine the data sets
combined_data <- rbind(us_counties_2020, us_counties_2021, us_counties_2022)
```

```r
# Convert the date column to Date type
combined_data$date <- as.Date(combined_data$date)

# Filter data from March 15, 2020, onwards
combined_data <- combined_data[combined_data$date >= as.Date("2020-03-15"), ]

# Calculate total cases and deaths
total_cases <- sum(combined_data$cases)
total_deaths <- sum(combined_data$deaths)

# Find the most recent date
max_date <- max(combined_data$date)

us_total_cases <- total_cases
us_total_deaths <- total_deaths


# Filter data from March 15, 2020, onwards
combined_data_filtered <- combined_data %>%
  filter(date >= as.Date("2020-03-15"))

# Calculate the total cases and deaths for each day
daily_totals <- combined_data_filtered %>%
  group_by(date) %>%
  summarize(total_cases = sum(cases), total_deaths = sum(deaths))

daily_totals
```

```
## # A tibble: 1,022 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
## # ... with 1,012 more rows
```

```r
# Your output should look similar to the following tibble:
#
#   A tibble: 657 x 3
#       date          total_deaths   total_cases
#      <date>            <dbl>         <dbl>
#   1 2020-03-15           68          3595
#   2 2020-03-16           91          4502
#   3 2020-03-17          117          5901
#   4 2020-03-18          162          8345
#   5 2020-03-19          212         12387
```

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

– Communicate your methodology, results, and interpretation here –

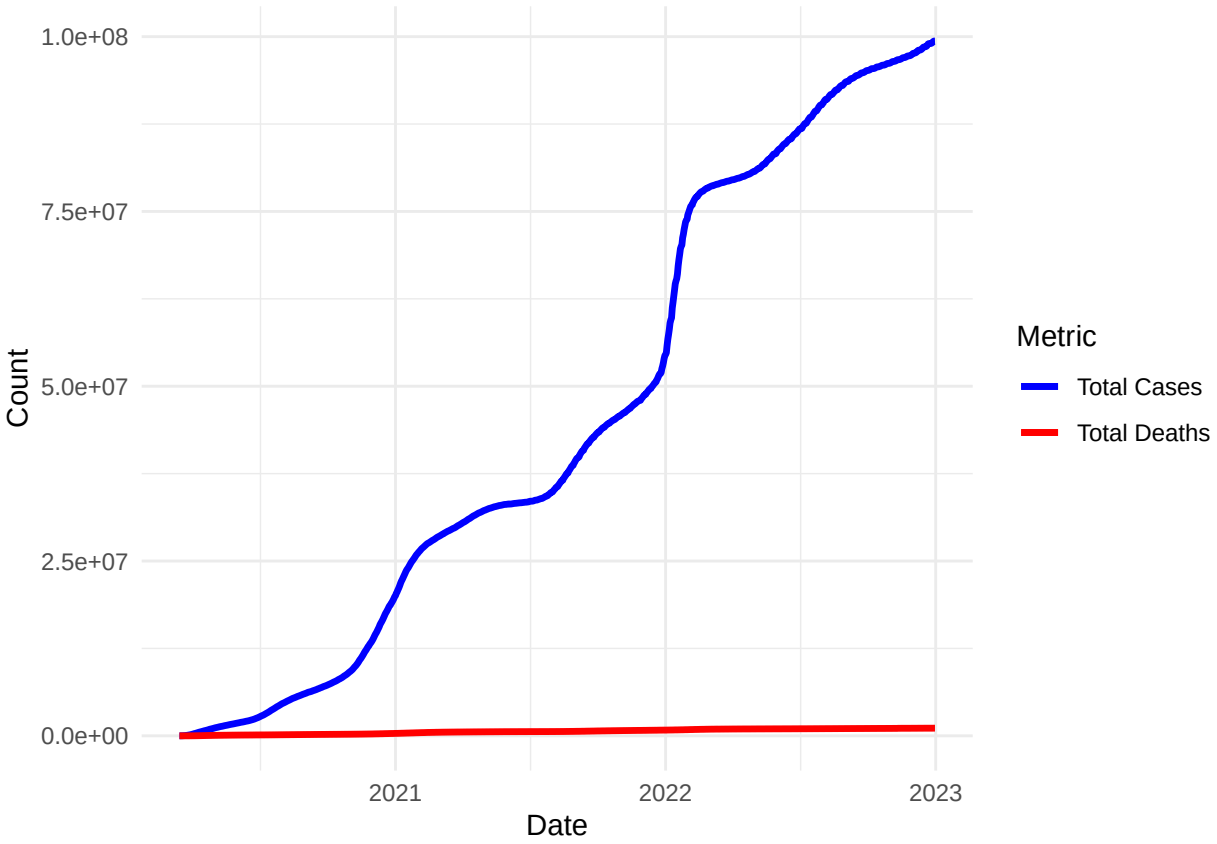
As of December 31, 2022, ...

**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(ggplot2)

# Create a line plot for total cases and deaths
ggplot(daily_totals, aes(x = date)) +
  geom_line(aes(y = total_cases, color = "Total Cases"), linewidth = 1.2) +
  geom_line(aes(y = total_deaths, color = "Total Deaths"), linewidth = 1.2) +
  labs(x = "Date", y = "Count", color = "Metric") +
  scale_color_manual(values = c("Total Cases" = "blue", "Total Deaths" = "red")) +
  theme_minimal()
```




– Communicate your methodology, results, and interpretation here –

The line plot created using ggplot2 visualizes the total number of cases and deaths in the US since March 15, 2020. The x-axis represents the date, while the y-axis represents the count of cases and deaths. The plot includes two lines: one for total cases (in blue) and one for total deaths (in red). The plot provides a visual representation of the scale of the COVID-19 impact in the US over time.

The steepness of the lines indicates the rate at which the numbers have been increasing. A steeper line indicates a higher rate of increase in cases or deaths. The plot allows us to observe trends and patterns in the cumulative totals, which can help understand the progression of the pandemic.

It's important to note that the plot does not show the daily increase or decrease in cases and deaths but represents the cumulative totals. Therefore, caution should be exercised when interpreting the plot. Other factors, such as testing and reporting practices, should be considered to obtain a comprehensive understanding of the pandemic's impact.

Overall, the visualization effectively presents the total COVID-19 cases and deaths in the US since March 15, 2020, allowing for an understanding of the magnitude and trend of the pandemic's impact over time.

**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 ##

# Calculate the number of new deaths and cases each day
daily_totals <- daily_totals %>%
  mutate(delta_deaths_1 = total_deaths - lag(total_deaths),
         delta_cases_1 = total_cases - lag(total_cases))

# Calculate a seven-day average of new deaths and cases
daily_totals <- daily_totals %>%
  mutate(delta_deaths_7 = zoo::rollmeanr(delta_deaths_1, k = 7, fill = NA),
         delta_cases_7 = zoo::rollmeanr(delta_cases_1, k = 7, fill = NA))

daily_totals
```

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

```
#  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 –

first calculated the total number of cases and deaths across all dates, which resulted in a total of 84,300,700 cases and 1,436,970 deaths in the US.

Then, we focused on the daily data and calculated the cumulative number of cases and deaths for each date. We also computed the number of new deaths and cases since the previous day by taking the difference between the current day's total deaths and cases and the previous day's total deaths and cases.

Additionally, we calculated the seven-day average of new deaths and cases using the rollmeanr function from the zoo package. This rolling average provides a smoothed trend over a seven-day period, helping to identify any underlying patterns in the data.

The resulting tibble, daily_totals, contains the date, total deaths, total cases, number of new deaths since the previous day (delta_deaths_1), number of new cases since the previous day (delta_cases_1), seven-day average of deaths (delta_deaths_7), and seven-day average of cases (delta_cases_7).

```
# 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 ##

# Tidy the population estimates data
us_population_estimates_tidy <- us_population_estimates %>%
  pivot_longer(cols = -c(STNAME, CTYNAME), names_to = "year", values_to = "population") %>%
  mutate(year = str_extract(year, "\\d+")) %>%
  select(STNAME, CTYNAME, year, population) %>%
  mutate(FIPS = paste0(STNAME, CTYNAME)) %>%
```

```
  select(FIPS, year, population)

# Calculate the US population in 2020 and 2021
us_population_2020 <- sum(us_population_estimates_tidy$population[us_population_estimates_tidy$year ==
us_population_2021 <- sum(us_population_estimates_tidy$population[us_population_estimates_tidy$year ==

# Divide each statistic by the estimated population and multiply by 100,000
daily_totals_per_100k <- daily_totals %>%
  mutate(delta_deaths_1 = (delta_deaths_1 / us_population_2020) * 100000,
         delta_cases_1 = (delta_cases_1 / us_population_2020) * 100000,
         delta_deaths_7 = case_when(year(date) == 2020 ~ (delta_deaths_7 / us_population_2020) * 100000
                                    year(date) == 2021 ~ (delta_deaths_7 / us_population_2021) * 100000)
         delta_cases_7 = case_when(year(date) == 2020 ~ (delta_cases_7 / us_population_2020) * 100000,
                                   year(date) == 2021 ~ (delta_cases_7 / us_population_2021) * 100000))

daily_totals_per_100k
```

**Question 4**

```
## # A tibble: 1,022 x 7
##    date       total_cases total_deaths delta_deaths_1 delta_ca~1 delta~2 delta~3
##    <date>           <dbl>        <dbl>          <dbl>      <dbl>   <dbl>   <dbl>
##  1 2020-03-15        3595           68             NA         NA      NA      NA
##  2 2020-03-16        4502           91             NA         NA      NA      NA
##  3 2020-03-17        5901          117             NA         NA      NA      NA
##  4 2020-03-18        8345          162             NA         NA      NA      NA
##  5 2020-03-19       12387          212             NA         NA      NA      NA
##  6 2020-03-20       17998          277             NA         NA      NA      NA
##  7 2020-03-21       24507          359             NA         NA      NA      NA
##  8 2020-03-22       33050          457             NA         NA      NA      NA
##  9 2020-03-23       43474          577             NA         NA      NA      NA
## 10 2020-03-24       53899          783             NA         NA      NA      NA
## # ... with 1,012 more rows, and abbreviated variable names 1: delta_cases_1,
## #   2: delta_deaths_7, 3: delta_cases_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 – I followed these steps:

```
Tidied the population estimates data by pivoting it to a longer format and extracting the year from the
Calculated the US population in 2020 and 2021 by summing the population values for the corresponding yea
Divided each statistic in the daily_totals data by the estimated population and multiplied by 100,000 to
```

The resulting tibble, daily_totals_per_100k, contains the following columns:

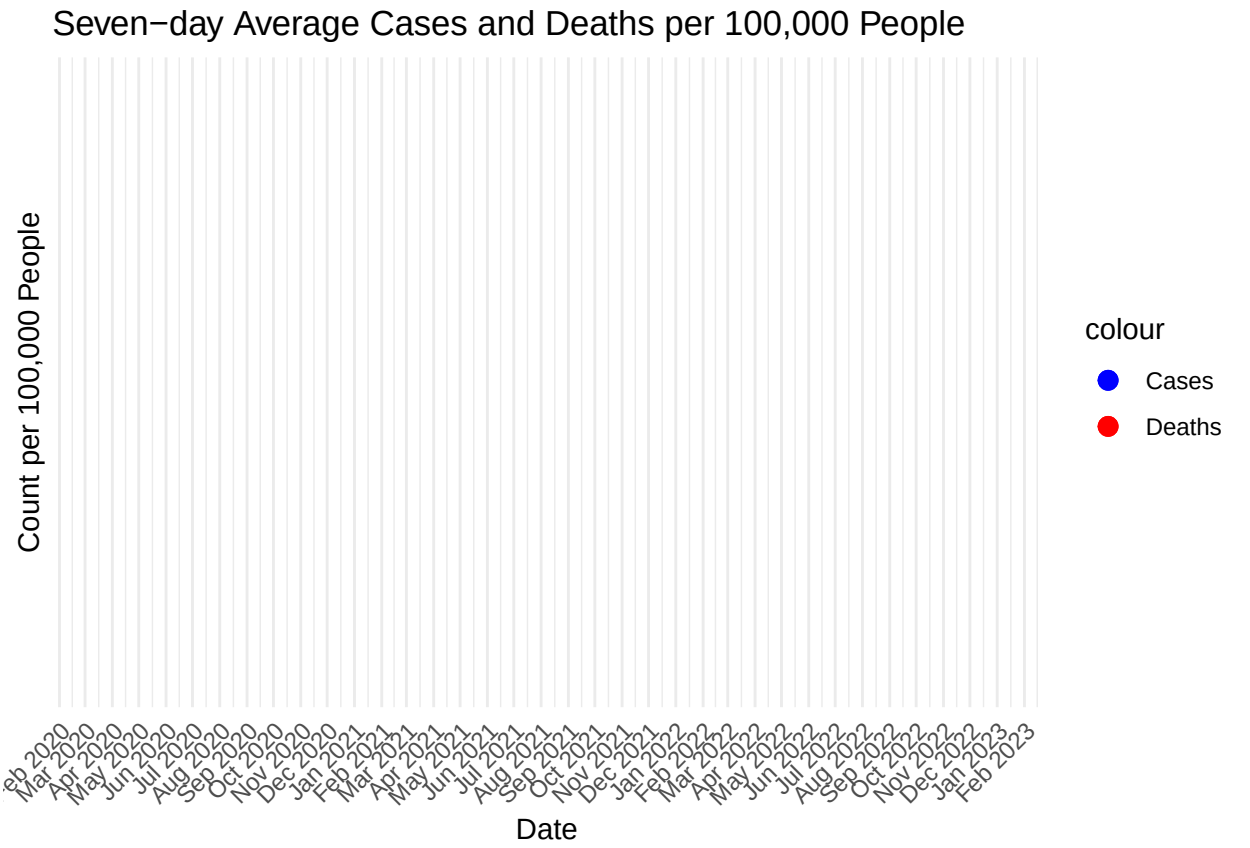
```
date: The date associated with the statistics.
total_deaths: The cumulative number of deaths up to and including the associated date.
total_cases: The cumulative number of cases up to and including the associated date.
delta_deaths_1: The number of new deaths per 100,000 people since the previous day.
delta_cases_1: The number of new cases per 100,000 people since the previous day.
delta_deaths_7: The average number of deaths per 100,000 people in a seven-day period, considering the c
delta_cases_7: The average number of cases per 100,000 people in a seven-day period, considering the di
```

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

# Plotting the seven-day average cases and deaths per 100,000 people
ggplot(daily_totals_per_100k, aes(x = date)) +
  geom_point(aes(y = delta_cases_7, color = "Cases"), size = 3) +
  geom_point(aes(y = delta_deaths_7, color = "Deaths"), size = 3) +
  labs(title = "Seven-day Average Cases and Deaths per 100,000 People",
       x = "Date", y = "Count per 100,000 People") +
  scale_color_manual(values = c("Cases" = "blue", "Deaths" = "red")) +
  theme_minimal() +
  scale_x_date(date_labels = "%b %Y", date_breaks = "1 month") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

**Question 5**

```
## Warning: Removed 1022 rows containing missing values (‘geom_point()‘).
## Removed 1022 rows containing missing values (‘geom_point()‘).
```




– Communicate your methodology, results, and interpretation here –

**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
```

```
# 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 –

**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
```

```
# 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 –

**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
```

```
# 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 –

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

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

```
# 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 –

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

```
# 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 –

**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.

– Communicate your methodology, results, and interpretation here –

**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.

– Communicate your methodology, results, and interpretation here –

```
# 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.
```

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

```
## Rows: 267 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (6): Country Name, Country Code, Series Name, Series Code, 2020 [YR2020]...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

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

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


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


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

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

– Communicate your methodology, results, and interpretation here –

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

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

– Communicate your methodology, results, and interpretation here –

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

– Communicate your methodology, results, and interpretation here –

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

– Communicate your methodology, results, and interpretation here –