US COVID_19 Cases and Deaths State Comparison

Hannah Bravo De Rueda

November 2023

US State Comparison Building on the previous project, analyzing the national averages of COVID-19 statistics in the US, this project will now dive into the COVID-19 statistics at the state level. 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.

Data Import & Wrangling The first task is to determine the top 10 states in terms of total deaths and cases between March 15, 2020, and December 31, 2021. Before we can determine the top ten states, we need to import the data, combine the three years of data, and remove the records for Puerto Rico.

```
# Import New York Times COVID-19 data
us_counties_2020 <-
  read_csv(
    "https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counties-2020.csv")
```

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

```
us_counties_2021 <-
  read_csv(
    "https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counties-2021.csv")
```

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

```r
us_counties_2022 <-
  read_csv(
    "https://raw.githubusercontent.com/nytimes/covid-19-data/master/us-counties-2022.csv")
```

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

```r
# Combine the 2020, 2021, and 2022 COVID data sets.
(us_counties <- us_counties_2020 %>%
  bind_rows(us_counties_2021) %>%
  bind_rows(us_counties_2022))
```

```
## # A tibble: 3,258,152 x 6
##    date       county      state      fips  cases deaths
##    <date>     <chr>       <chr>      <chr> <dbl>  <dbl>
##  1 2020-01-21 Snohomish   Washington 53061     1      0
##  2 2020-01-22 Snohomish   Washington 53061     1      0
##  3 2020-01-23 Snohomish   Washington 53061     1      0
##  4 2020-01-24 Cook        Illinois   17031     1      0
##  5 2020-01-24 Snohomish   Washington 53061     1      0
##  6 2020-01-25 Orange      California 06059     1      0
##  7 2020-01-25 Cook        Illinois   17031     1      0
##  8 2020-01-25 Snohomish   Washington 53061     1      0
##  9 2020-01-26 Maricopa    Arizona    04013     1      0
## 10 2020-01-26 Los Angeles California 06037     1      0
## # i 3,258,142 more rows
```

```r
# Now, remove Puerto Rico and other US territories
(us_counties <- us_counties %>%
  filter(date >= "2020-03-15",
         state != "Puerto Rico",
         state != "Virgin Islands",
         state != "Northern Mariana Islands",
         state != "Guam",
         state != "American Samoa"))
```

```
## # A tibble: 3,171,661 x 6
##    date       county     state   fips  cases deaths
##    <date>     <chr>      <chr>   <chr> <dbl>  <dbl>
##  1 2020-03-15 Baldwin    Alabama 01003     1      0
##  2 2020-03-15 Elmore     Alabama 01051     1      0
##  3 2020-03-15 Jefferson  Alabama 01073    13      0
##  4 2020-03-15 Lee        Alabama 01081     1      0
##  5 2020-03-15 Limestone  Alabama 01083     1      0
##  6 2020-03-15 Montgomery Alabama 01101     1      0
```

```
##  7 2020-03-15 Shelby     Alabama 01117     2      0
##  8 2020-03-15 Tuscaloosa Alabama 01125     3      0
##  9 2020-03-15 Anchorage  Alaska  02020     1      0
## 10 2020-03-15 Graham     Arizona 04009     1      0
## # i 3,171,651 more rows
```

```
us_counties %>%
  filter(date == "2022-12-31") %>%
  group_by(state)
```

```
## # A tibble: 3,168 x 6
## # Groups:   state [51]
##    date       county   state   fips  cases deaths
##    <date>     <chr>    <chr>   <chr> <dbl>  <dbl>
##  1 2022-12-31 Autauga  Alabama 01001 18961    230
##  2 2022-12-31 Baldwin  Alabama 01003 67496    719
##  3 2022-12-31 Barbour  Alabama 01005  7027    111
##  4 2022-12-31 Bibb     Alabama 01007  7692    108
##  5 2022-12-31 Blount   Alabama 01009 17731    260
##  6 2022-12-31 Bullock  Alabama 01011  2886     54
##  7 2022-12-31 Butler   Alabama 01013  6185    130
##  8 2022-12-31 Calhoun  Alabama 01015 39458    665
##  9 2022-12-31 Chambers Alabama 01017 10311    174
## 10 2022-12-31 Cherokee Alabama 01019  6456    133
## # i 3,158 more rows
```

```
# Determine the top 10 states in terms of total deaths and cases between March 15, 2020,
# and December 31, 2021. To do this, transform your combined COVID-19 data to summarize
# total deaths and cases by state up to December 31, 2021.

state_totals <- us_counties %>%
  filter(date == "2021-12-31") %>%
  select(date, state, cases, deaths) %>%
  group_by(state) %>%
  summarise(total_cases = sum(cases), total_deaths = sum(deaths)) %>%
  arrange(desc(total_cases))
state_totals
```

```
## # A tibble: 51 x 3
##    state          total_cases total_deaths
##    <chr>                <dbl>        <dbl>
##  1 California         5515613        76709
##  2 Texas              4574881        76062
##  3 Florida            4166392        62504
##  4 New York           3473970        58993
##  5 Illinois           2154058        31017
##  6 Pennsylvania       2036424        36705
##  7 Ohio               2016095        29447
##  8 Georgia            1798497        30283
##  9 Michigan           1706355        28984
## 10 North Carolina     1685504        19436
## # i 41 more rows
```

I imported three data sets for COVID-19 cases and deaths in the US across 2020, 2021, and 2022 published by New York Times. I combined the three data sets, and filtered out records that are non-sovereign US territories, to focus exclusively on the 50 states. Once I had a combined data set for all 50 states, including the District of Columbia, across each year of the pandemic, I aggregated the data across county to get a total number for each state, as of December 31, 2021. From there, I was able to determine the 10 states in the US with the highest number of COVID-19 cases and deaths. California tops the list at #1, then Texas, Florida, and New York with the 4th highest number of cases. It's no surprise that we see California, Texas, Florida, and New York holding the top 4 spots, considering they're the states with the largest populations in the US. So naturally, we see higher numbers for COVID-19 cases and deaths, compared to states with smaller populations. It would be more interesting to determine the states with the highest number of COVID-19 cases and deaths proportionate to the state's population, by calculating cases and deaths per 100,000 people.

Top 10 States Impacted 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.

```
# Determine the top 10 states for deaths and cases per 100,000 people between March 15, 2020,
# and December 31, 2021. You should first tidy and transform the population estimates to include
# population totals by state. Use your relational data verbs (e.g. full_join()) to join the
# population estimates with the cases and death statistics using the state name as a key.
# Finally, mutate your table to calculate deaths and cases per 100,000 people and summarize by state.

# Import Population Estimates from US Census Bureau
us_population_estimates <- read_csv("https://raw.githubusercontent.com/HannahBravo/US-COVID-19-Statisti
```

```
## 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.
```

```
# Calculate the population estimates for each state by finding the average across 2020 and 2021
(state_pop_est <- us_population_estimates %>%
  group_by(STNAME) %>%
  summarise(st_est = round(sum(Estimate)/2, 0)))
```

```
## # A tibble: 51 x 2
##    STNAME                 st_est
##    <chr>                   <dbl>
##  1 Alabama               5032340
##  2 Alaska                 732557
##  3 Arizona               7227151
##  4 Arkansas              3019062
##  5 California           39368787
##  6 Colorado              5798188
##  7 Connecticut           3602928
##  8 Delaware               997635
##  9 District of Columbia   680072
## 10 Florida              21675530
## # i 41 more rows
```

```r
# Now join the state population estimates to the state cases and deaths table, and calculate
# the top 10 states with the highest cases and deaths per 100,000 people
state_totals %>%
  left_join(state_pop_est, by = join_by(state == STNAME)) %>%
  mutate(cases_per = round(total_cases/st_est*100000, 1),
         deaths_per = round(total_deaths/st_est*100000, 1)) %>%
  arrange(desc(cases_per))
```

```
## # A tibble: 51 x 6
##    state        total_cases total_deaths   st_est cases_per deaths_per
##    <chr>              <dbl>        <dbl>    <dbl>     <dbl>      <dbl>
##  1 North Dakota      174220         2057   776955    22423.       265.
##  2 Alaska            156130          954   732557    21313        130.
##  3 Rhode Island      231096         3066  1095920    21087.       280.
##  4 South Dakota      179204         2486   891238    20107.       279.
##  5 Wyoming           115638         1526   578035    20005.       264
##  6 Tennessee        1379917        20640  6947668    19862.       297.
##  7 Utah              637144         3787  3309830    19250.       114.
##  8 Florida          4166392        62504 21675530    19222.       288.
##  9 Kentucky          864599        12149  4506676    19185.       270.
## 10 Arizona          1381488        24229  7227151    19115.       335.
## # i 41 more rows
```

In order to determine the states with the highest number of COVID-19 cases and deaths proportionate to their overall population, we need to weight each state's total number of cases and deaths per 100,000 people. I first imported the population estimate for each state in the US, for 2020 and 2021, tidied it up by averaging the estimates across 2020 and 2021, and then grouped them by state. With each states averaged population estimate, I recalculated the COVID-19 numbers for each state by dividing by the population estimate, and multiplied that result by 100,000. The new values are the COVID-19 statistics for each state, per 100,000 people. Giving us a better picture of COVID-19's impact on each state, relative to the size of their overall population.

I arranged the table to display the results with the highest 'total_cases' to lowest, grouped by state. We now see that North Dakota tops the list at #1, Alaska in second, Rhode Island, South Dakota, and Wyoming securing the top five for the states with the highest number of cases and deaths per 100,000 people. At the bottom of the list, we see Hawaii at #51, then Oregon, Vermont, Maine, and Washington rounding out the bottom five, for the states with the least number of COVID-19 cases and deaths per 100,000 people.

This normalized list shows us which states were hit the hardest by COVID-19, despite population size. Looking further into why North Dakota, and Alaska were impacted the most, compared to Hawaii and Oregon, gives us a better chance for narrowing in on the why. For instance, it's interesting that Alaska and Hawaii, both remote islands, feature on opposite ends of the list. Why was Alaska more exposed to COVID-19 than Hawaii? Did public policy and economic factors contribute more to the outcome than geographic location?

North Dakota Since North Dakota was impacted the most by COVID-19, per 100,000 people, I will calculate the seven-day averages for new cases and deaths per 100,000 people. Once I have calculated the averages, I will create a visualization using ggplot2 to represent the data.

```r
# Filter the above table for North Dakota, and calculate new cases/deaths per 100,000 people
# and the 7-day rolling average between 3-15-2020 & 12-31-2021

# Filter previous table for North Dakota, and then sum cases and deaths by date
```

```r
(nd_totals <- us_counties %>%
  filter(state == "North Dakota" & date <= "2021-12-31") %>%
  group_by(date, state) %>%
  summarize(total_cases = sum(cases), total_deaths = sum(deaths)))
```

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

```
## # A tibble: 657 x 4
## # Groups:   date [657]
##    date       state        total_cases total_deaths
##    <date>     <chr>              <dbl>        <dbl>
##  1 2020-03-15 North Dakota           1            0
##  2 2020-03-16 North Dakota           1            0
##  3 2020-03-17 North Dakota           5            0
##  4 2020-03-18 North Dakota           7            0
##  5 2020-03-19 North Dakota          19            0
##  6 2020-03-20 North Dakota          27            0
##  7 2020-03-21 North Dakota          28            0
##  8 2020-03-22 North Dakota          30            0
##  9 2020-03-23 North Dakota          32            0
## 10 2020-03-24 North Dakota          37            0
## # i 647 more rows
```

```r
# Calculate cases/deaths per 100,000 people
(nd_totals <- nd_totals %>%
  left_join(state_pop_est, by = join_by(state == STNAME)) %>%
  mutate(cases_per = round(total_cases/st_est*100000, 2),
         deaths_per = round(total_deaths/st_est*100000, 4)))
```

```
## # A tibble: 657 x 7
## # Groups:   date [657]
##    date       state        total_cases total_deaths st_est cases_per deaths_per
##    <date>     <chr>              <dbl>        <dbl>  <dbl>     <dbl>      <dbl>
##  1 2020-03-15 North Dakota           1            0 776955      0.13          0
##  2 2020-03-16 North Dakota           1            0 776955      0.13          0
##  3 2020-03-17 North Dakota           5            0 776955      0.64          0
##  4 2020-03-18 North Dakota           7            0 776955      0.9           0
##  5 2020-03-19 North Dakota          19            0 776955      2.45          0
##  6 2020-03-20 North Dakota          27            0 776955      3.48          0
##  7 2020-03-21 North Dakota          28            0 776955      3.6           0
##  8 2020-03-22 North Dakota          30            0 776955      3.86          0
##  9 2020-03-23 North Dakota          32            0 776955      4.12          0
## 10 2020-03-24 North Dakota          37            0 776955      4.76          0
## # i 647 more rows
```

```r
# Calculate NEW cases per 100,000 people
(nd_totals <- add_column(nd_totals,
           new_cases_per = nd_totals$cases_per - lag(nd_totals$cases_per, n = 1, default = 0)))
```

```
## # A tibble: 657 x 8
```

```
## # Groups:   date [657]
##    date       state        total_cases total_deaths st_est cases_per deaths_per
##    <date>     <chr>              <dbl>        <dbl>  <dbl>     <dbl>      <dbl>
##  1 2020-03-15 North Dakota           1            0 776955      0.13          0
##  2 2020-03-16 North Dakota           1            0 776955      0.13          0
##  3 2020-03-17 North Dakota           5            0 776955      0.64          0
##  4 2020-03-18 North Dakota           7            0 776955      0.9           0
##  5 2020-03-19 North Dakota          19            0 776955      2.45          0
##  6 2020-03-20 North Dakota          27            0 776955      3.48          0
##  7 2020-03-21 North Dakota          28            0 776955      3.6           0
##  8 2020-03-22 North Dakota          30            0 776955      3.86          0
##  9 2020-03-23 North Dakota          32            0 776955      4.12          0
## 10 2020-03-24 North Dakota          37            0 776955      4.76          0
## # i 647 more rows
## # i 1 more variable: new_cases_per <dbl>
```

```r
# Calculate NEW deaths per 100,000 people
(nd_totals <- add_column(nd_totals,
           new_deaths_per = nd_totals$deaths_per - lag(nd_totals$deaths_per, n = 1, default = 0)))
```

```
## # A tibble: 657 x 9
## # Groups:   date [657]
##    date       state        total_cases total_deaths st_est cases_per deaths_per
##    <date>     <chr>              <dbl>        <dbl>  <dbl>     <dbl>      <dbl>
##  1 2020-03-15 North Dakota           1            0 776955      0.13          0
##  2 2020-03-16 North Dakota           1            0 776955      0.13          0
##  3 2020-03-17 North Dakota           5            0 776955      0.64          0
##  4 2020-03-18 North Dakota           7            0 776955      0.9           0
##  5 2020-03-19 North Dakota          19            0 776955      2.45          0
##  6 2020-03-20 North Dakota          27            0 776955      3.48          0
##  7 2020-03-21 North Dakota          28            0 776955      3.6           0
##  8 2020-03-22 North Dakota          30            0 776955      3.86          0
##  9 2020-03-23 North Dakota          32            0 776955      4.12          0
## 10 2020-03-24 North Dakota          37            0 776955      4.76          0
## # i 647 more rows
## # i 2 more variables: new_cases_per <dbl>, new_deaths_per <dbl>
```

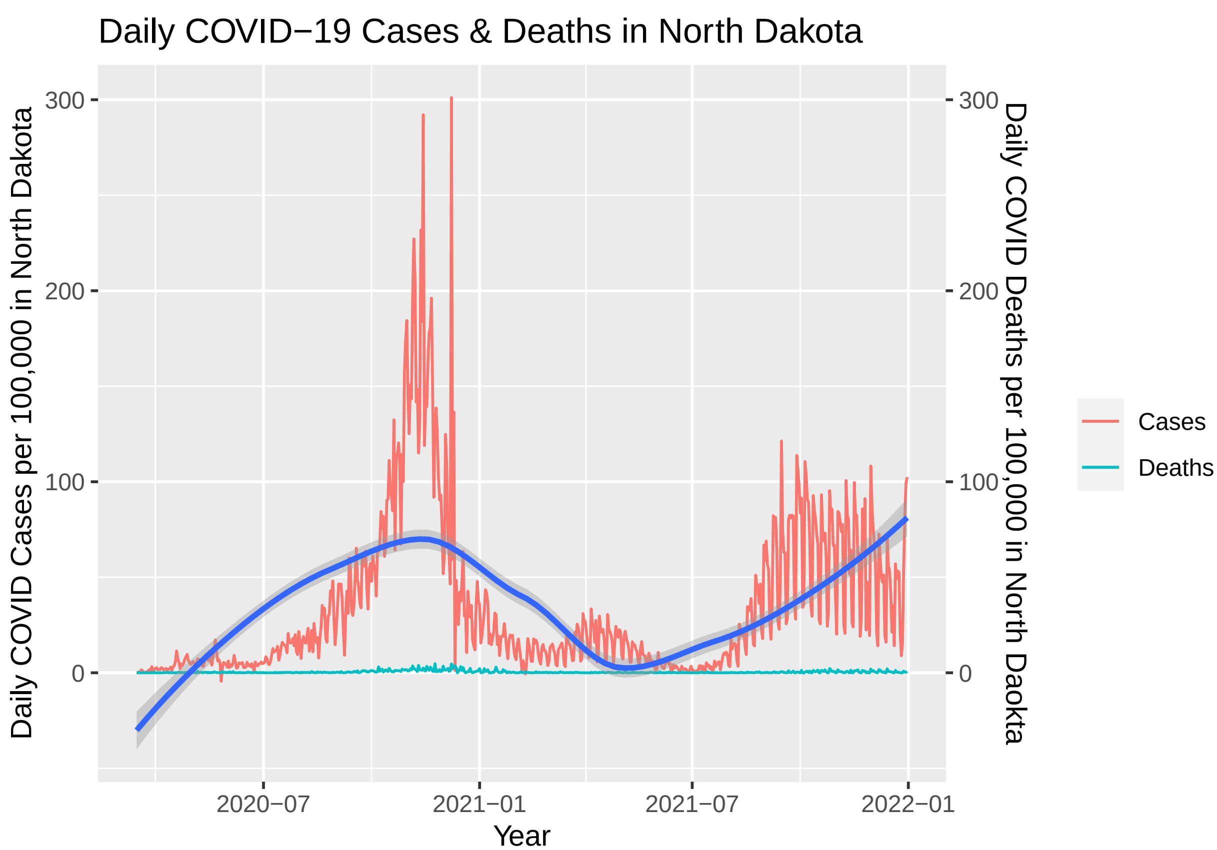
```r
# Calculate the 7-day rolling average for cases/deaths per 100,000 people
(nd_wkly_avg <- nd_totals %>%
  ungroup() %>%
  mutate(wkly_avg_cases = round(lag((lead(nd_totals$new_cases_per, n= 7) -
                                  new_cases_per)/7, n = 7), 2),
        wkly_avg_deaths = round(lag((lead(nd_totals$new_deaths_per, n= 7) -
                                   new_deaths_per)/7, n = 7), 3)))
```

```
## # A tibble: 657 x 11
##    date       state        total_cases total_deaths st_est cases_per deaths_per
##    <date>     <chr>              <dbl>        <dbl>  <dbl>     <dbl>      <dbl>
##  1 2020-03-15 North Dakota           1            0 776955      0.13          0
##  2 2020-03-16 North Dakota           1            0 776955      0.13          0
##  3 2020-03-17 North Dakota           5            0 776955      0.64          0
##  4 2020-03-18 North Dakota           7            0 776955      0.9           0
##  5 2020-03-19 North Dakota          19            0 776955      2.45          0
```

```
##   6 2020-03-20 North Dakota          27             0 776955       3.48           0
##   7 2020-03-21 North Dakota          28             0 776955       3.6            0
##   8 2020-03-22 North Dakota          30             0 776955       3.86           0
##   9 2020-03-23 North Dakota          32             0 776955       4.12           0
## 10 2020-03-24 North Dakota           37             0 776955       4.76           0
## # i 647 more rows
## # i 4 more variables: new_cases_per <dbl>, new_deaths_per <dbl>,
## #   wkly_avg_cases <dbl>, wkly_avg_deaths <dbl>
```

```
# Create a visualization representing the data for North Dakota
nd_totals %>%
  ggplot(aes(x = date)) +
  geom_line(aes(y = new_cases_per, color = "Cases")) +
  geom_line(aes(y = new_deaths_per, color = "Deaths")) +
  geom_smooth(aes(y = new_cases_per)) +
  scale_y_continuous(
    name = "Daily COVID Cases per 100,000 in North Dakota",
    sec.axis = sec_axis(~., name = "Daily COVID Deaths per 100,000 in North Daokta")) +
  labs(x = "Year", title = "Daily COVID-19 Cases & Deaths in North Dakota", color = "")
```

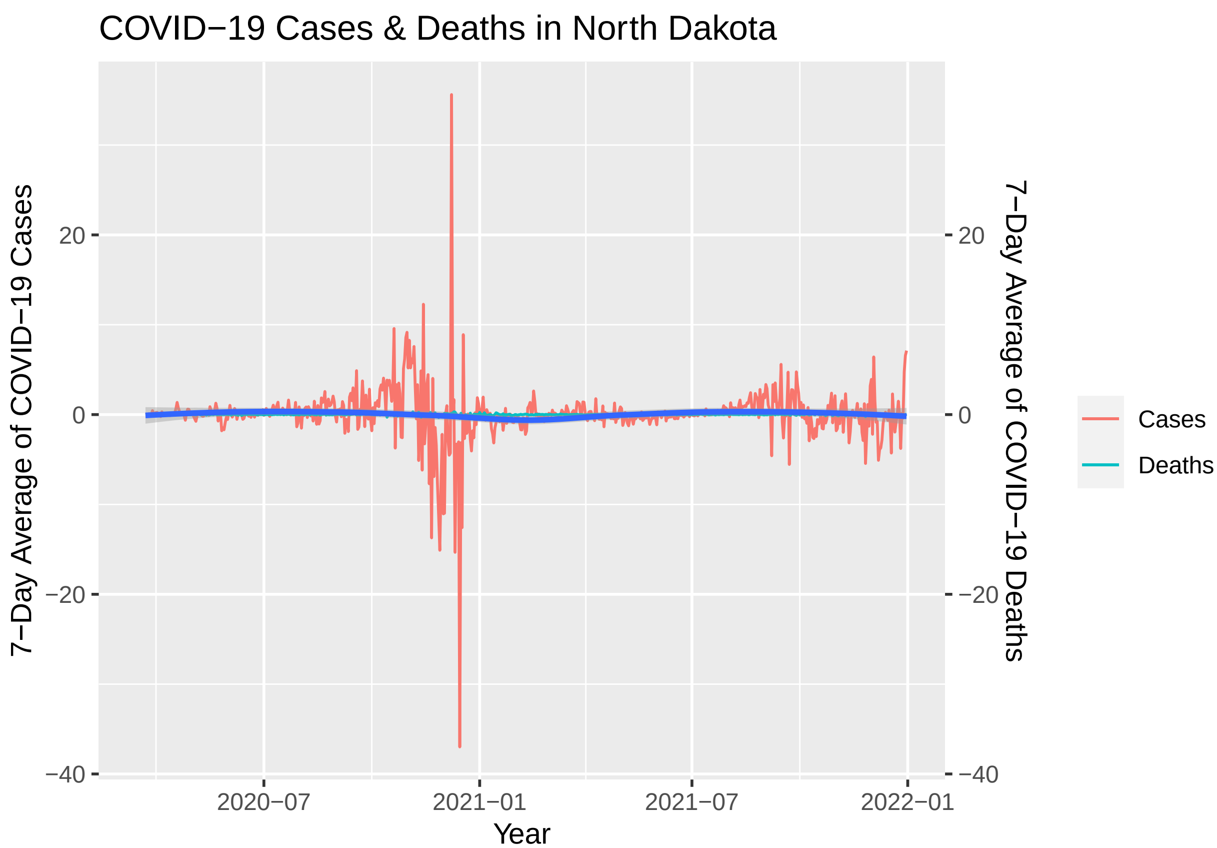
```
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'
```




```
nd_wkly_avg %>%
  ggplot(aes(x = date)) +
  geom_line(aes(y = wkly_avg_cases, color = "Cases")) +
```

```r
  geom_line(aes(y = wkly_avg_deaths, color = "Deaths")) +
  geom_smooth(aes(y = wkly_avg_cases)) +
  scale_y_continuous(
    name = "7-Day Average of COVID-19 Cases",
    sec.axis = sec_axis(~., name = "7-Day Average of COVID-19 Deaths")) +
  labs(x = "Year", title = "COVID-19 Cases & Deaths in North Dakota", color = "")
```

```
## `geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



I chose to take a closer look at North Dakota, since it topped the list, as the state with the most COVID-19 cases and deaths per 100,000 people. I added two columns to the table to calculate the 7-day rolling average of North Dakota's cases and deaths per 100,000 people. I then charted total COVID-19 cases and deaths per 100,000 people, as well as the 7-day rolling average.

The first chart is a time series that shows the cumulative growth of COVID-19 cases and deaths between March 15, 2020 and December 31, 2021. There is a dual axis, with the left axis representing the scale for COVID-19 cases per 100,000 in North Dakota and then the right side axis representing the scale for deaths per 100,000 people. Cases is represented by the red line, and deaths by the light blue line. Both lines follow a similar trend in that they both experience proportionate spikes and plateaus during the same time period, despite their different scales. For instance, there is a significant spike in both cases and deaths in the Fall of 2020. Both cases and deaths start around 0, but cases rises up to over 20,000 cases per 100,000 people and deaths reaches just under 300 deaths per 100,000 people.

The second chart is the 7-day rolling average of both COVID-19 cases and deaths per 100,000 people, between March 15, 2020 and December 31, 2021. There is a dual axis, with the left axis representing the 7-day average of cases per 100,000 in North Dakota and the right axis representing the 7-day average of deaths per 100,000 people. Cases is represented by the red line, and deaths by the light blue line. Both lines follow a similar trend in spikes and dips, despite their different scales. Both cases and deaths reach their highest weekly average in the Fall of 2020 where cases reaches above 150 cases per 100,000 people, and deaths reaches just under 3 deaths per 100,000 people. The next spike that both cases and deaths showed is in the Fall of 2021, with cases getting up to ~75,000 cases per 100,000 people, and deaths reaching just under 1 death per 100,000 people. Those numbers are still only half of what they were in the Fall of 2020.

Both of these charts could be supplemented with event lines denoting important policy change dates for COVID restrictions, vaccine release, and the dates of any enforced mandates.

Top 5 Counties in North Dakota Still analyzing North Dakota, I want to identify the top 5 counties in terms of deaths and cases per 100,000 people.

```r
# Filter North Dakota between 3-15-2020 & 12-31-2021 from the combined data set from the
# previous project to summarize cases and deaths.

# Let's import the county population estimates for North Dakota for 2020 and 2021
nd_county_pop_est <- read_csv("https://raw.githubusercontent.com/HannahBravo/US-COVID-19-Statistics---S
```

```
## New names:
## Rows: 54 Columns: 4
## -- Column specification
## ---------------------------------------------------------- Delimiter: "," chr
## (1): table with row headers in column A and column headers in rows 3 thr... num
## (3): ...2, ...3, ...4
## 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.
## * `` -> `...2`
## * `` -> `...3`
## * `` -> `...4`
```

```r
nd_county_pop_est <- nd_county_pop_est[-1,] %>%
  rename(county =
           "table with row headers in column A and column headers in rows 3 through 4 (leading dots indi
         "2020" = "...2",
         "2021" = "...3",
         "2022" = "...4")

nd_county_pop_est <- nd_county_pop_est[, -4] %>%
    mutate(across("county", str_replace_all, "[.]", ""))
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `across("county", str_replace_all, "[.]", "")`.
## Caused by warning:
## ! The `...` argument of `across()` is deprecated as of dplyr 1.1.0.
## Supply arguments directly to `.fns` through an anonymous function instead.
##
##   # Previously
##   across(a:b, mean, na.rm = TRUE)
##
##   # Now
##   across(a:b, \(x) mean(x, na.rm = TRUE))
```

```r
nd_county_pop_est <- nd_county_pop_est %>%
  mutate(across("county", str_replace_all, " County$", ""))

nd_county_pop_est <- nd_county_pop_est %>%
  mutate(county_est = rowSums(nd_county_pop_est[, -1])/2) %>%
  select(county, county_est)

# Let's join the ND county population estimates to the ND county cases and deaths table
(nd_counties <- us_counties %>%
  filter(state == "North Dakota" & date == "2021-12-31")%>%
  filter(county != "Unknown"))
```

```
## # A tibble: 53 x 6
##    date       county    state        fips  cases deaths
##    <date>     <chr>     <chr>        <chr> <dbl>  <dbl>
##  1 2021-12-31 Adams     North Dakota 38001   472      8
##  2 2021-12-31 Barnes    North Dakota 38003  2229     40
##  3 2021-12-31 Benson    North Dakota 38005  1478     22
##  4 2021-12-31 Billings  North Dakota 38007   120      1
##  5 2021-12-31 Bottineau North Dakota 38009  1186     24
##  6 2021-12-31 Bowman    North Dakota 38011   682      9
##  7 2021-12-31 Burke     North Dakota 38013   360      3
##  8 2021-12-31 Burleigh  North Dakota 38015 25555    281
##  9 2021-12-31 Cass      North Dakota 38017 39829    286
## 10 2021-12-31 Cavalier  North Dakota 38019   597      7
## # i 43 more rows
```

```r
# Top 10 counties in North Dakota with the most cases
(nd_county_cases <- nd_counties %>%
  left_join(nd_county_pop_est, by = join_by(county == county)) %>%
  group_by(county) %>%
  summarise(date, fips, total_county_cases = round(sum(cases)/county_est*100000, 2),
            total_county_deaths = round(sum(deaths)/county_est*100000, 2)) %>%
  arrange(desc(total_county_cases)))
```

```
## # A tibble: 53 x 5
##    county   date       fips  total_county_cases total_county_deaths
##    <chr>    <date>     <chr>              <dbl>               <dbl>
##  1 Rolette  2021-12-31 38079             29579.                297.
##  2 Stark    2021-12-31 38089             27267.                236.
##  3 Eddy     2021-12-31 38027             26066.                259.
##  4 Burleigh 2021-12-31 38015             25870.                284.
##  5 Sioux    2021-12-31 38085             25834.                471.
##  6 Morton   2021-12-31 38059             25619.                392.
##  7 Benson   2021-12-31 38005             25246.                376.
##  8 Walsh    2021-12-31 38099             24692.                324.
##  9 Dickey   2021-12-31 38021             24058.                770.
## 10 Stutsman 2021-12-31 38093             23955.                426.
## # i 43 more rows
```

```r
# Top 10 counties in North Dakota with the most deaths
(nd_county_deaths <- nd_counties %>%
```

```r
  left_join(nd_county_pop_est, by = join_by(county == county)) %>%
  group_by(county) %>%
  summarise(date, fips, total_county_cases = round(sum(cases)/county_est*100000, 2),
            total_county_deaths = round(sum(deaths)/county_est*100000, 2)) %>%
  arrange(desc(total_county_deaths))
```

```
## # A tibble: 53 x 5
##    county    date       fips  total_county_cases total_county_deaths
##    <chr>     <date>     <chr>              <dbl>               <dbl>
##  1 Dickey    2021-12-31 38021             24058.                770.
##  2 Pierce    2021-12-31 38069             21540.                733.
##  3 Renville  2021-12-31 38075             20022.                662.
##  4 Logan     2021-12-31 38047             18717.                583.
##  5 Foster    2021-12-31 38031             21541.                563.
##  6 Kidder    2021-12-31 38043             17018.                548.
##  7 McHenry   2021-12-31 38049             20209.                531.
##  8 Nelson    2021-12-31 38063             20540                 530.
##  9 Towner    2021-12-31 38095             22430.                514.
## 10 Emmons    2021-12-31 38029             18642.                488.
## # i 43 more rows
```

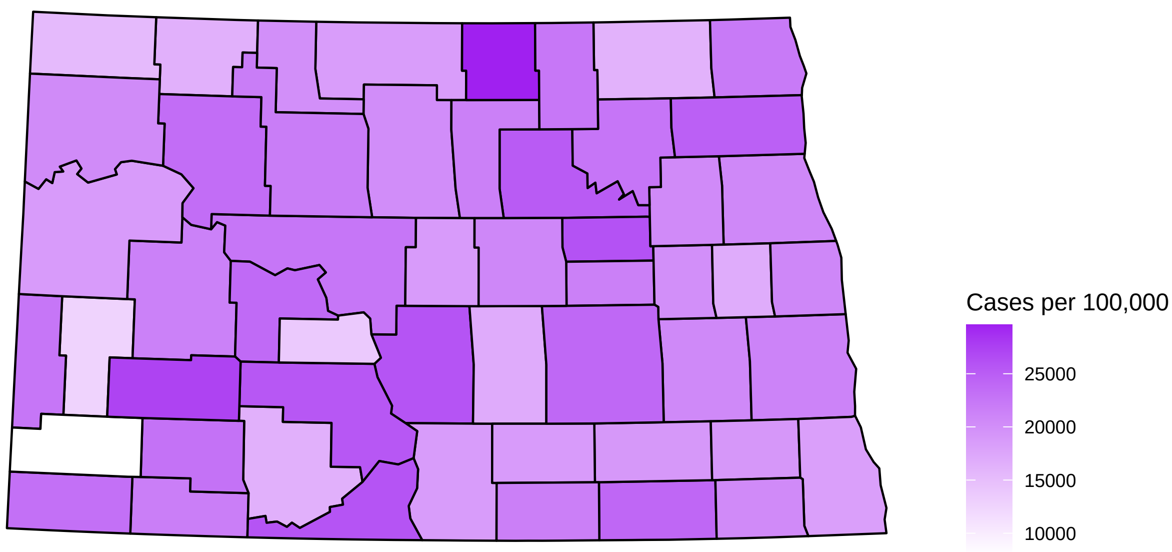
We know that North Dakota as a state overall took the hardest hit from COVID-19, but what about the counties that make up North Dakota? Which ones had the highest number of cases and deaths per 100,000 people? In order to compare the numbers for COVID-19 across counties in North Dakota, I had to import the county population estimates for North Dakota. Once that data was imported and tidied, I calculated the average population estimate for each county across 2020 and 2021. I then used the averaged estimate to calculate the total number of cases and deaths per 100,000 people for each county in North Dakota.

We see Rolette, Stark, and Eddy as the top three counties in North Dakota with the highest COVID-19 cases per 100,000 people. Then Dickey, Pierce, and Renville with the highest number of COVID-19 related deaths per 100,000 people. You would expect the counties that were hit the hardest with COVID-19 cases, would also be the counties hit the hardest with COVID-19 related deaths. However, Dickey county is the only county that appears in the top ten for both cases AND deaths. So why did some counties experience more exposure to the virus, but others experienced more deaths related to the virus? It would be interesting to compare the average age of the population for the counties with the most deaths to those with the most cases.

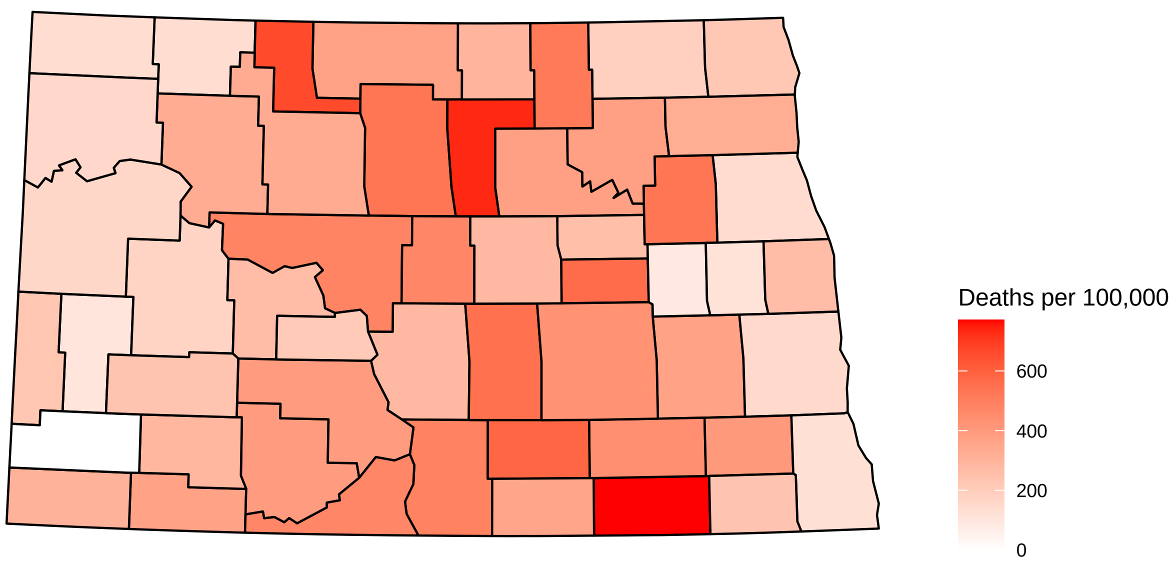
North Dakota County-Level Visualization I will create a map projection to plot county-level deaths and cases per 100,000 people for North Dakota.

```r
# Using 'plot_usmap()' to create a map projection, visualizing the cases and deaths in the
# counties of North Dakota.

# Map projection of COVID cases in North Dakota by county
plot_usmap(regions = "county", include = "ND", data = nd_county_cases,
           values = "total_county_cases", color = "black") +
  scale_fill_continuous(low = "white", high = "purple", name = "Cases per 100,000") +
  theme(legend.position = "right")
```




```r
# Map projection of COVID deaths in North Dakota by county
plot_usmap(regions = "county", include = "ND", data = nd_county_deaths,
           values = "total_county_deaths", color = "black") +
  scale_fill_continuous(low = "white", high = "red", name = "Deaths per 100,000") +
  theme(legend.position = "right")
```



I used the package 'usmap' to visualize a population density map of COVID-19's impact on counties across North Dakota. I created two population density maps, one displaying the impact of COVID-19 cases across counties, and the second shows the impact of COVID-19 related deaths across counties. The regions on the map with the darkest shade of color, are the counties impacted the most by either cases or deaths.

For the map showing cases across North Dakota, we don't see an obvious trend or grouping among the counties hit the hardest or the least. There is one county on the map that looks very close to white, suggesting no instances of COVID-19 cases, which would need to be investigated further. For COVID-19 related deaths across North Dakota, we again don't see an obvious trend or grouping of the counties on either end. However, we do see the same county with almost no instances of COVID-19 deaths.

Again, the counties hit the hardest with the most cases are not the same counties hit the hardest with COVID-19 related deaths. Suggesting that geographic location doesn't seem to play a role in how a county is impacted by COVID-19. Could it instead be a result of how age and economic resources are distributed across counties in North Dakota? It would be interesting to compare these numbers to each counties average age, and SES scores.

Alaska, Oregon, & Hawaii Statistics Finally, I want to look at three other states: Alaska, Oregon, and Hawaii, and calculate the seven-day average for new deaths and cases per 100,000 people between March 15, 2020, and December 31, 2021.

```
# The three other states I am going to pick are Alaska because of it's remoteness and being
# second on the list for most cases per 100,000 people, then Oregon for being second to last
# on the list, and finally Hawaii for it's remoteness and being last on the list of most cases
# per 100,000 people.

# Alaska
```

```
(ak_totals <- us_counties %>%
  filter(state == "Alaska" & date <= "2021-12-31") %>%
  group_by(date, state) %>%
  summarize(total_cases = sum(cases), total_deaths = sum(deaths)))
```

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

```
## # A tibble: 657 x 4
## # Groups:   date [657]
##    date       state  total_cases total_deaths
##    <date>     <chr>        <dbl>        <dbl>
##  1 2020-03-15 Alaska           1            0
##  2 2020-03-16 Alaska           3            0
##  3 2020-03-17 Alaska           6            0
##  4 2020-03-18 Alaska           9            0
##  5 2020-03-19 Alaska          12            0
##  6 2020-03-20 Alaska          14            0
##  7 2020-03-21 Alaska          21            0
##  8 2020-03-22 Alaska          22            0
##  9 2020-03-23 Alaska          36            0
## 10 2020-03-24 Alaska          42            0
## # i 647 more rows
```

```
(ak_totals <- ak_totals %>%
  left_join(state_pop_est, by = join_by(state == STNAME)) %>%
  mutate(cases_per = round(total_cases/st_est*100000, 2),
         deaths_per = round(total_deaths/st_est*100000, 4)))
```

```
## # A tibble: 657 x 7
## # Groups:   date [657]
##    date       state  total_cases total_deaths st_est cases_per deaths_per
##    <date>     <chr>        <dbl>        <dbl>  <dbl>     <dbl>      <dbl>
##  1 2020-03-15 Alaska           1            0 732557      0.14          0
##  2 2020-03-16 Alaska           3            0 732557      0.41          0
##  3 2020-03-17 Alaska           6            0 732557      0.82          0
##  4 2020-03-18 Alaska           9            0 732557      1.23          0
##  5 2020-03-19 Alaska          12            0 732557      1.64          0
##  6 2020-03-20 Alaska          14            0 732557      1.91          0
##  7 2020-03-21 Alaska          21            0 732557      2.87          0
##  8 2020-03-22 Alaska          22            0 732557      3             0
##  9 2020-03-23 Alaska          36            0 732557      4.91          0
## 10 2020-03-24 Alaska          42            0 732557      5.73          0
## # i 647 more rows
```

```
(ak_wkly_avg <- ak_totals %>%
  ungroup() %>%
  mutate(wkly_avg_cases = round(lag((lead(ak_totals$cases_per, n= 7) -
                                     ak_totals$cases_per)/7, n = 7), 2),
         wkly_avg_deaths = round(lag((lead(ak_totals$deaths_per, n= 7) -
                                      ak_totals$deaths_per)/7, n = 7), 3)))
```

```
## # A tibble: 657 x 9
##    date       state  total_cases total_deaths st_est cases_per deaths_per
##    <date>     <chr>        <dbl>        <dbl>  <dbl>     <dbl>      <dbl>
##  1 2020-03-15 Alaska           1            0 732557      0.14          0
##  2 2020-03-16 Alaska           3            0 732557      0.41          0
##  3 2020-03-17 Alaska           6            0 732557      0.82          0
##  4 2020-03-18 Alaska           9            0 732557      1.23          0
##  5 2020-03-19 Alaska          12            0 732557      1.64          0
##  6 2020-03-20 Alaska          14            0 732557      1.91          0
##  7 2020-03-21 Alaska          21            0 732557      2.87          0
##  8 2020-03-22 Alaska          22            0 732557      3             0
##  9 2020-03-23 Alaska          36            0 732557      4.91          0
## 10 2020-03-24 Alaska          42            0 732557      5.73          0
## # i 647 more rows
## # i 2 more variables: wkly_avg_cases <dbl>, wkly_avg_deaths <dbl>
```

```r
# Oregon
(or_totals <- us_counties %>%
  filter(state == "Oregon" & date <= "2021-12-31") %>%
  group_by(date, state) %>%
  summarize(total_cases = sum(cases), total_deaths = sum(deaths)))
```

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

```
## # A tibble: 657 x 4
## # Groups:   date [657]
##    date       state  total_cases total_deaths
##    <date>     <chr>        <dbl>        <dbl>
##  1 2020-03-15 Oregon          39            1
##  2 2020-03-16 Oregon          46            1
##  3 2020-03-17 Oregon          66            2
##  4 2020-03-18 Oregon          74            3
##  5 2020-03-19 Oregon          87            3
##  6 2020-03-20 Oregon         114            3
##  7 2020-03-21 Oregon         137            4
##  8 2020-03-22 Oregon         161            5
##  9 2020-03-23 Oregon         191            5
## 10 2020-03-24 Oregon         209            8
## # i 647 more rows
```

```r
(or_totals <- or_totals %>%
  left_join(state_pop_est, by = join_by(state == STNAME)) %>%
  mutate(cases_per = round(total_cases/st_est*100000, 2),
         deaths_per = round(total_deaths/st_est*100000, 4)))
```

```
## # A tibble: 657 x 7
## # Groups:   date [657]
##    date       state  total_cases total_deaths  st_est cases_per deaths_per
##    <date>     <chr>        <dbl>        <dbl>   <dbl>     <dbl>      <dbl>
##  1 2020-03-15 Oregon          39            1 4243850      0.92     0.0236
##  2 2020-03-16 Oregon          46            1 4243850      1.08     0.0236
```

```
##   3 2020-03-17 Oregon           66            2 4243850      1.56     0.0471
##   4 2020-03-18 Oregon           74            3 4243850      1.74     0.0707
##   5 2020-03-19 Oregon           87            3 4243850      2.05     0.0707
##   6 2020-03-20 Oregon          114            3 4243850      2.69     0.0707
##   7 2020-03-21 Oregon          137            4 4243850      3.23     0.0943
##   8 2020-03-22 Oregon          161            5 4243850      3.79     0.118
##   9 2020-03-23 Oregon          191            5 4243850      4.5      0.118
##  10 2020-03-24 Oregon          209            8 4243850      4.92     0.188
## # i 647 more rows
```

```
(or_wkly_avg <- or_totals %>%
  ungroup() %>%
  mutate(wkly_avg_cases = round(lag((lead(or_totals$cases_per, n= 7) -
                                       or_totals$cases_per)/7, n = 7), 2),
         wkly_avg_deaths = round(lag((lead(or_totals$deaths_per, n= 7) -
                                        or_totals$deaths_per)/7, n = 7), 3)))
```

```
## # A tibble: 657 x 9
##    date       state  total_cases total_deaths  st_est cases_per deaths_per
##    <date>     <chr>        <dbl>        <dbl>   <dbl>     <dbl>      <dbl>
##   1 2020-03-15 Oregon           39            1 4243850      0.92     0.0236
##   2 2020-03-16 Oregon           46            1 4243850      1.08     0.0236
##   3 2020-03-17 Oregon           66            2 4243850      1.56     0.0471
##   4 2020-03-18 Oregon           74            3 4243850      1.74     0.0707
##   5 2020-03-19 Oregon           87            3 4243850      2.05     0.0707
##   6 2020-03-20 Oregon          114            3 4243850      2.69     0.0707
##   7 2020-03-21 Oregon          137            4 4243850      3.23     0.0943
##   8 2020-03-22 Oregon          161            5 4243850      3.79     0.118
##   9 2020-03-23 Oregon          191            5 4243850      4.5      0.118
##  10 2020-03-24 Oregon          209            8 4243850      4.92     0.188
## # i 647 more rows
## # i 2 more variables: wkly_avg_cases <dbl>, wkly_avg_deaths <dbl>
```

```
# Hawaii
(hi_totals <- us_counties %>%
  filter(state == "Hawaii" & date <= "2021-12-31") %>%
  group_by(date, state) %>%
  summarize(total_cases = sum(cases), total_deaths = sum(deaths)))
```

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

```
## # A tibble: 657 x 4
## # Groups:   date [657]
##    date       state  total_cases total_deaths
##    <date>     <chr>        <dbl>        <dbl>
##   1 2020-03-15 Hawaii            7            0
##   2 2020-03-16 Hawaii           10            0
##   3 2020-03-17 Hawaii           14            0
##   4 2020-03-18 Hawaii           16            0
##   5 2020-03-19 Hawaii           26            0
##   6 2020-03-20 Hawaii           37            0
```

```
##   7 2020-03-21 Hawaii          48           0
##   8 2020-03-22 Hawaii          56           0
##   9 2020-03-23 Hawaii          77           0
##  10 2020-03-24 Hawaii          90           0
## # i 647 more rows
```

```r
(hi_totals <- hi_totals %>%
  left_join(state_pop_est, by = join_by(state == STNAME)) %>%
  mutate(cases_per = round(total_cases/st_est*100000, 2),
         deaths_per = round(total_deaths/st_est*100000, 4)))
```

```
## # A tibble: 657 x 7
## # Groups:   date [657]
##    date       state  total_cases total_deaths  st_est cases_per deaths_per
##    <date>     <chr>        <dbl>        <dbl>   <dbl>     <dbl>      <dbl>
##  1 2020-03-15 Hawaii           7            0 1446732      0.48          0
##  2 2020-03-16 Hawaii          10            0 1446732      0.69          0
##  3 2020-03-17 Hawaii          14            0 1446732      0.97          0
##  4 2020-03-18 Hawaii          16            0 1446732      1.11          0
##  5 2020-03-19 Hawaii          26            0 1446732      1.8           0
##  6 2020-03-20 Hawaii          37            0 1446732      2.56          0
##  7 2020-03-21 Hawaii          48            0 1446732      3.32          0
##  8 2020-03-22 Hawaii          56            0 1446732      3.87          0
##  9 2020-03-23 Hawaii          77            0 1446732      5.32          0
## 10 2020-03-24 Hawaii          90            0 1446732      6.22          0
## # i 647 more rows
```

```r
(hi_wkly_avg <- hi_totals %>%
  ungroup() %>%
  mutate(wkly_avg_cases = round(lag((lead(hi_totals$cases_per, n= 7) -
                                      cases_per)/7, n= 7), 2),
         wkly_avg_deaths = round(lag((lead(hi_totals$deaths_per, n= 7) -
                                      deaths_per)/7, n= 7), 3)))
```

```
## # A tibble: 657 x 9
##    date       state  total_cases total_deaths  st_est cases_per deaths_per
##    <date>     <chr>        <dbl>        <dbl>   <dbl>     <dbl>      <dbl>
##  1 2020-03-15 Hawaii           7            0 1446732      0.48          0
##  2 2020-03-16 Hawaii          10            0 1446732      0.69          0
##  3 2020-03-17 Hawaii          14            0 1446732      0.97          0
##  4 2020-03-18 Hawaii          16            0 1446732      1.11          0
##  5 2020-03-19 Hawaii          26            0 1446732      1.8           0
##  6 2020-03-20 Hawaii          37            0 1446732      2.56          0
##  7 2020-03-21 Hawaii          48            0 1446732      3.32          0
##  8 2020-03-22 Hawaii          56            0 1446732      3.87          0
##  9 2020-03-23 Hawaii          77            0 1446732      5.32          0
## 10 2020-03-24 Hawaii          90            0 1446732      6.22          0
## # i 647 more rows
## # i 2 more variables: wkly_avg_cases <dbl>, wkly_avg_deaths <dbl>
```

Breaking down the numbers for North Dakota was interesting, so I went ahead and calculated the same statistics for three other states: Alaska, Hawaii, and Oregon. I chose Alaska because it was the state with the second highest numbers for COVID-19 cases AND deaths per 100,000 people. Oregon and Hawaii I chose because they're on the opposite end of the list, as the two states with the least amount of COVID-19 cases and deaths per 100,000 people.

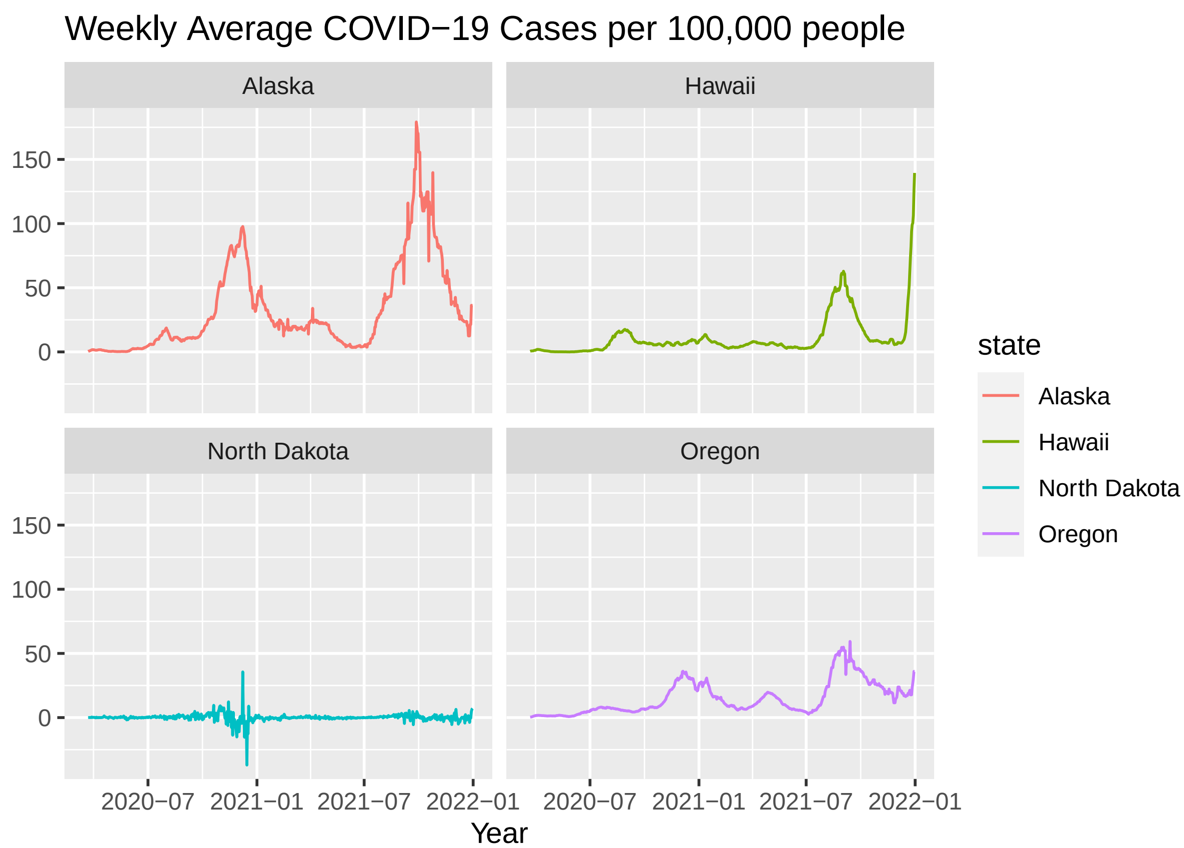
I created a separate table for each state, by filtering the US counties data table to one of the above states, filtered the data again for records between March 15, 2020 and December 31, 2021. I then calculated the cumulative total for COVID-19 cases and deaths in each of the above states, which was then converted to the total per 100,000 people, based off the states population estimate. The last step is to turn the state's totals per 100,000 people into a rolling 7-day average per 100,000 people. Now that we have the numbers for the two states hit the hardest by COVID-19 and the numbers for the two states impacted the least, it would be interesting to vizualize the data for all four states.

Visualizing Averages Across States Now I will create a visualization comparing the seven-day averages for new deaths and cases per 100,000 people for North Dakota, Alaska, Oregon, & Hawaii.

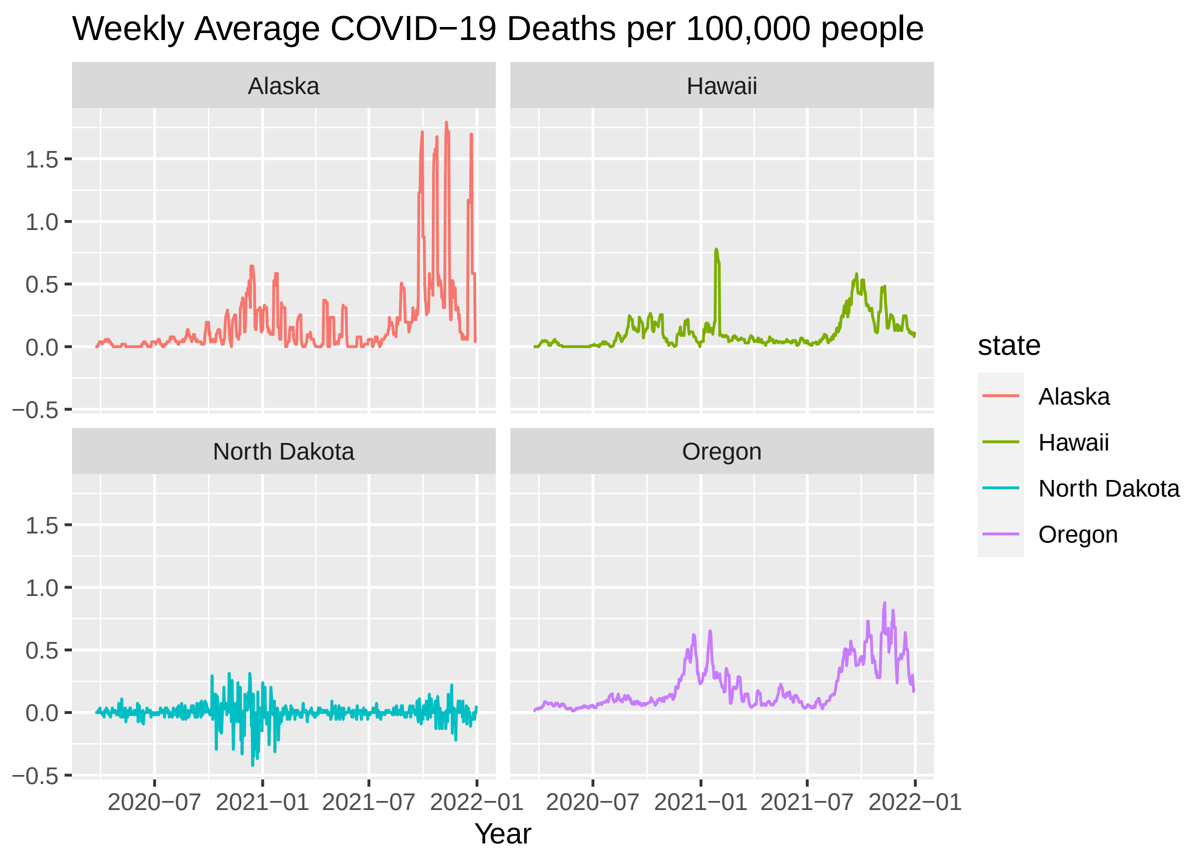
```r
# First let's combine all the weekly average data for the four states into one table to plot.
(st_wkly_avgs <- bind_rows(nd_wkly_avg, ak_wkly_avg, or_wkly_avg, hi_wkly_avg))
```

```
## # A tibble: 2,628 x 11
##    date       state        total_cases total_deaths st_est cases_per deaths_per
##    <date>     <chr>              <dbl>        <dbl>  <dbl>     <dbl>      <dbl>
##  1 2020-03-15 North Dakota           1            0 776955      0.13          0
##  2 2020-03-16 North Dakota           1            0 776955      0.13          0
##  3 2020-03-17 North Dakota           5            0 776955      0.64          0
##  4 2020-03-18 North Dakota           7            0 776955      0.9           0
##  5 2020-03-19 North Dakota          19            0 776955      2.45          0
##  6 2020-03-20 North Dakota          27            0 776955      3.48          0
##  7 2020-03-21 North Dakota          28            0 776955      3.6           0
##  8 2020-03-22 North Dakota          30            0 776955      3.86          0
##  9 2020-03-23 North Dakota          32            0 776955      4.12          0
## 10 2020-03-24 North Dakota          37            0 776955      4.76          0
## # i 2,618 more rows
## # i 4 more variables: new_cases_per <dbl>, new_deaths_per <dbl>,
## #   wkly_avg_cases <dbl>, wkly_avg_deaths <dbl>
```

```r
# Now let's plot the weekly averages for cases and deaths per 100,000 people for the four states.
st_wkly_avgs %>%
  ggplot(aes(x = date)) +
  geom_line(aes(y = wkly_avg_cases, color = state)) +
  facet_wrap(vars(state)) +
  labs(x = "Year", y = "", title = "Weekly Average COVID-19 Cases per 100,000 people")
```




```
st_wkly_avgs %>%
  ggplot(aes(x = date)) +
  geom_line(aes(y = wkly_avg_deaths, color = state)) +
  facet_wrap(vars(state)) +
  labs(x = "Year", y = "", title = "Weekly Average COVID-19 Deaths per 100,000 people")
```



In order to compare the COVID-19 numbers for each of the four states, I first needed to combine each states table of COVID-19 cases and deaths statistics. Once the tables were combined using bind_rows(), I built two visuals to display the time series for each state's 7-day average of cases and then a second visual for deaths. I charted 'date' on the x-axis and the state's 7-day average for either cases or deaths on the y-axis. I chose to differentiate the states by the color of their time series line and by faceting them into individual plots. I found faceting them into individual plots helped simplify it, since it was too busy with all four lines over-layed on one plot. This way, you can see each states unique trend across the two years, and compare inflection points between states.

For instance, we see that for the weekly average of COVID-19 cases per 100,000 people, North Dakota and Alaska had larger spikes than either Oregon or Hawaii. However, Alaska, Oregon, and Hawaii all experienced their highest spike in the fall of 2021; whereas North Dakota experienced it's largest spike in the fall of 2020. Hawaii also shows a significant spike, it's largest yet, at the tail end of the data. It would be interesting to investigate what happened in Hawaii in the winter of 2022.

As far as the weekly average of COVID-19 deaths per 100,000 people goes, North Dakota and Alaska again show higher spikes than Oregon or Hawaii. But North Dakota is the only state who shows a spike in the Fall of 2020. They show fluctuations in their data, but nothing as pronounced North Dakota in the fall of 2020. Alaska's data gets pretty chaotic in the fall of 2021, and while the other four states also show a bump in their data during that time, Alaska shows four very steep spikes and drops, which would also be interesting to look further into, to determine if it wass an error or something to follow.