Johns Hopkins COVID-19 Data

Jason Horne

6/13/2022

Background

As part of the response to the global COVID-19 pandemic, research institutions around the world have collected and published data online related to the spread of the virus. Johns Hopkins provides one such set of data, publishing US and global case and death counts to a GitHub repository[1] that is free to access.

We will investigate these data to answer some questions, including:

1. How well do the COVID-19 cases correlate to deaths?
2. Have death rates remained the same through the spread of the Delta and Omicron variants?
3. How has the state of North Carolina fared in the pandemic, compared to the rest of the country?

Load and Standardize

First, we load the data by URL and perform some simple transformations to make them easier to work with.

```
url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covi
file_names <- c("time_series_covid19_confirmed_global.csv",
                "time_series_covid19_deaths_global.csv",
                "time_series_covid19_confirmed_US.csv",
                "time_series_covid19_deaths_US.csv")
urls <- str_c(url_in, file_names)

global_cases <- read_csv(urls[1])
```

```
## Rows: 285 Columns: 882
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (2): Province/State, Country/Region
## dbl (880): 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.
```

```
global_deaths <- read_csv(urls[2])
```

[1] https://github.com/CSSEGISandData/COVID-19

```
## Rows: 285 Columns: 882
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (2): Province/State, Country/Region
## dbl (880): 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.
```

```
US_cases <- read_csv(urls[3])
```

```
## Rows: 3342 Columns: 889
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (883): 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.
```

```
US_deaths <- read_csv(urls[4])
```

```
## Rows: 3342 Columns: 890
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (884): 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.
```

The data come as a single row per political unit (country, state/province, county, etc) with columns for each date.

Let's use pivot_longer to transform the data such that we have a unique row per political unit per date. This will dramatically increase the number of rows in the data, but likewise dramatically decrease the number of columns in the data.

```
global_cases <- global_cases %>%
  pivot_longer(
    cols = -c(`Province/State`, `Country/Region`, Lat, Long),
    names_to = "date",
    values_to = "cases"
  ) %>%
  select(-c (Lat, Long))

global_deaths <- global_deaths %>%
  pivot_longer(
    cols = -c(`Province/State`, `Country/Region`, Lat, Long),
    names_to = "date",
    values_to = "deaths"
  ) %>%
  select(-c (Lat, Long))
```

```
US_cases <- US_cases %>%
  pivot_longer(
    cols = -(UID:Combined_Key),
    names_to = "date",
    values_to = "cases"
  ) %>%
  select(Admin2:cases) %>%
  mutate(date = mdy(date)) %>%
  select(-c (Lat, Long_))

US_deaths <- US_deaths %>%
  pivot_longer(
    cols = -c(UID:Combined_Key, Population),
    names_to = "date",
    values_to = "deaths"
  ) %>%
  select(Admin2:deaths) %>%
  mutate(date = mdy(date)) %>%
  select(-c (Lat, Long_))
```

Note that global and US data are each separated into datasets for the number of cases and the number of deaths. Let's use the join functions to put those together.

After this step, there should be one row per political unit per date, with each row having both the case count and the death count.

```
global <- global_cases %>%
  full_join(global_deaths) %>%
  rename(Country_Region = `Country/Region`,
         Province_State = `Province/State`) %>%
  mutate(date = mdy(date)) %>%
  filter(cases > 0)
```

```
## Joining, by = c("Province/State", "Country/Region", "date")
```

```
US <- US_cases %>%
  full_join(US_deaths)
```

```
## Joining, by = c("Admin2", "Province_State", "Country_Region", "Combined_Key",
## "date")
```

Next, we'll add a Combined_Key field to the global data. Then we'll load a separate dataset with more country-specific information and join it to our global data by state/province and country.

```
global <- global %>%
  unite("Combined_Key",
        c(Province_State, Country_Region),
        sep = ", ",
        na.rm = TRUE,
        remove = FALSE)
```

```
uid_lookup_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/
uid <- read_csv(uid_lookup_url) %>%
  select(-c(Lat, Long_, Combined_Key, code3, iso2, iso3, Admin2))
```

```
## Rows: 4317 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): iso2, iso3, FIPS, Admin2, Province_State, Country_Region, Combined_Key
## dbl (5): UID, code3, Lat, Long_, Population
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
global <- global %>%
  left_join(uid, by = c("Province_State", "Country_Region")) %>%
  select(-c(UID,FIPS)) %>%
  select(Province_State, Country_Region, date, cases, deaths,
         Population, Combined_Key)
```

Summary of Input Data

Let's take a quick look at the data we have so far.

```
summary(US)
```

```
##     Admin2          Province_State     Country_Region     Combined_Key
##  Length:2934276     Length:2934276     Length:2934276     Length:2934276
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##       date                cases            Population           deaths
##  Min.   :2020-01-22   Min.   :  -3073   Min.   :       0   Min.   :  -82
##  1st Qu.:2020-08-28   1st Qu.:    138   1st Qu.:    9917   1st Qu.:    1
##  Median :2021-04-04   Median :   1405   Median :   24892   Median :   24
##  Mean   :2021-04-04   Mean   :   9581   Mean   :   99604   Mean   :  147
##  3rd Qu.:2021-11-10   3rd Qu.:   5357   3rd Qu.:   64979   3rd Qu.:   90
##  Max.   :2022-06-17   Max.   :3057004   Max.   :10039107   Max.   :32250
```

```
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:230618      Length:230618      Min.   :2020-01-22   Min.   :       1
##  Class :character   Class :character   1st Qu.:2020-10-03   1st Qu.:     735
##  Mode  :character   Mode  :character   Median :2021-05-02   Median :   11476
##                                        Mean   :2021-04-28   Mean   :  656210
##                                        3rd Qu.:2021-11-25   3rd Qu.:  162796
##                                        Max.   :2022-06-17   Max.   :86216418
##
```

```
##      deaths          Population        Combined_Key
##  Min.   :      0   Min.   :8.090e+02   Length:230618
##  1st Qu.:      6   1st Qu.:8.696e+05   Class :character
##  Median :    135   Median :7.133e+06   Mode  :character
##  Mean   :  11503   Mean   :2.928e+07
##  3rd Qu.:   2555   3rd Qu.:2.914e+07
##  Max.   :1013358   Max.   :1.380e+09
##                    NA's   :4609
```

We note that there are rows in the US data that have a negative count of cases and deaths. Let's look more closely.

```
US %>% filter(cases < 0)
```

```
## # A tibble: 2 x 8
##   Admin2  Province_State Country_Region Combined_Key date       cases Population
##   <chr>   <chr>          <chr>          <chr>        <date>     <dbl>      <dbl>
## 1 Unassi~ South Carolina US             Unassigned,~ 2022-05-05 -3073          0
## 2 Unassi~ South Carolina US             Unassigned,~ 2022-05-06 -3073          0
## # ... with 1 more variable: deaths <dbl>
```

```
US %>% filter(deaths < 0)
```

```
## # A tibble: 2 x 8
##   Admin2  Province_State Country_Region Combined_Key date       cases Population
##   <chr>   <chr>          <chr>          <chr>        <date>     <dbl>      <dbl>
## 1 Unassi~ South Carolina US             Unassigned,~ 2022-05-05 -3073          0
## 2 Unassi~ South Carolina US             Unassigned,~ 2022-05-06 -3073          0
## # ... with 1 more variable: deaths <dbl>
```

It seems there are two rows from South Carolina with both negative case and death counts. Why could that be?

For now, let's just remove those rows as possible errors, though we should later investigate to see if they're attempting to model a correction in previous numbers, or similar.

```
US <- US %>% filter(cases >= 0)
summary(US)
```

```
##     Admin2          Province_State     Country_Region     Combined_Key
##  Length:2934274     Length:2934274     Length:2934274     Length:2934274
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##       date                cases           Population           deaths
##  Min.   :2020-01-22   Min.   :      0   Min.   :       0   Min.   :    0
##  1st Qu.:2020-08-28   1st Qu.:    138   1st Qu.:    9917   1st Qu.:    1
##  Median :2021-04-04   Median :   1405   Median :   24909   Median :   24
##  Mean   :2021-04-04   Mean   :   9581   Mean   :   99604   Mean   :  147
##  3rd Qu.:2021-11-10   3rd Qu.:   5357   3rd Qu.:   64979   3rd Qu.:   90
##  Max.   :2022-06-17   Max.   :3057004   Max.   :10039107   Max.   :32250
```

Now all of our data have nonnegative case and death counts.

Normalize Counts per Capita

To compare US states with large populations against those with small populations, we need to look at a per capita case and death rate. We will do this by calculating the number of cases and deaths per million.

First, we summarize all of the counties within a state. This US_by_state dataset will have one row per state per date.

```
US_by_state <- US %>%
  group_by(Province_State, Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths),
            Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths * 1000000 / Population) %>%
  select(Province_State, Country_Region, date, cases, deaths,
         deaths_per_mill, Population) %>%
  ungroup()
```

```
## ‘summarise()‘ has grouped output by ’Province_State’, ’Country_Region’. You can
## override using the ‘.groups‘ argument.
```

```
US_by_state
```

```
## # A tibble: 50,924 x 7
##    Province_State Country_Region date       cases deaths deaths_per_mill
##    <chr>          <chr>          <date>     <dbl>  <dbl>           <dbl>
##  1 Alabama        US             2020-01-22     0      0               0
##  2 Alabama        US             2020-01-23     0      0               0
##  3 Alabama        US             2020-01-24     0      0               0
##  4 Alabama        US             2020-01-25     0      0               0
##  5 Alabama        US             2020-01-26     0      0               0
##  6 Alabama        US             2020-01-27     0      0               0
##  7 Alabama        US             2020-01-28     0      0               0
##  8 Alabama        US             2020-01-29     0      0               0
##  9 Alabama        US             2020-01-30     0      0               0
## 10 Alabama        US             2020-01-31     0      0               0
## # ... with 50,914 more rows, and 1 more variable: Population <dbl>
```

Our US_totals will summarize across the states. The resulting dataset will have one row per date, showing the number of total cases and deaths nationwide by that date.

```
US_totals <- US_by_state %>%
  group_by(Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths),
            Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths * 1000000 / Population) %>%
  select(Country_Region, date, cases, deaths,
         deaths_per_mill, Population) %>%
  ungroup()
```

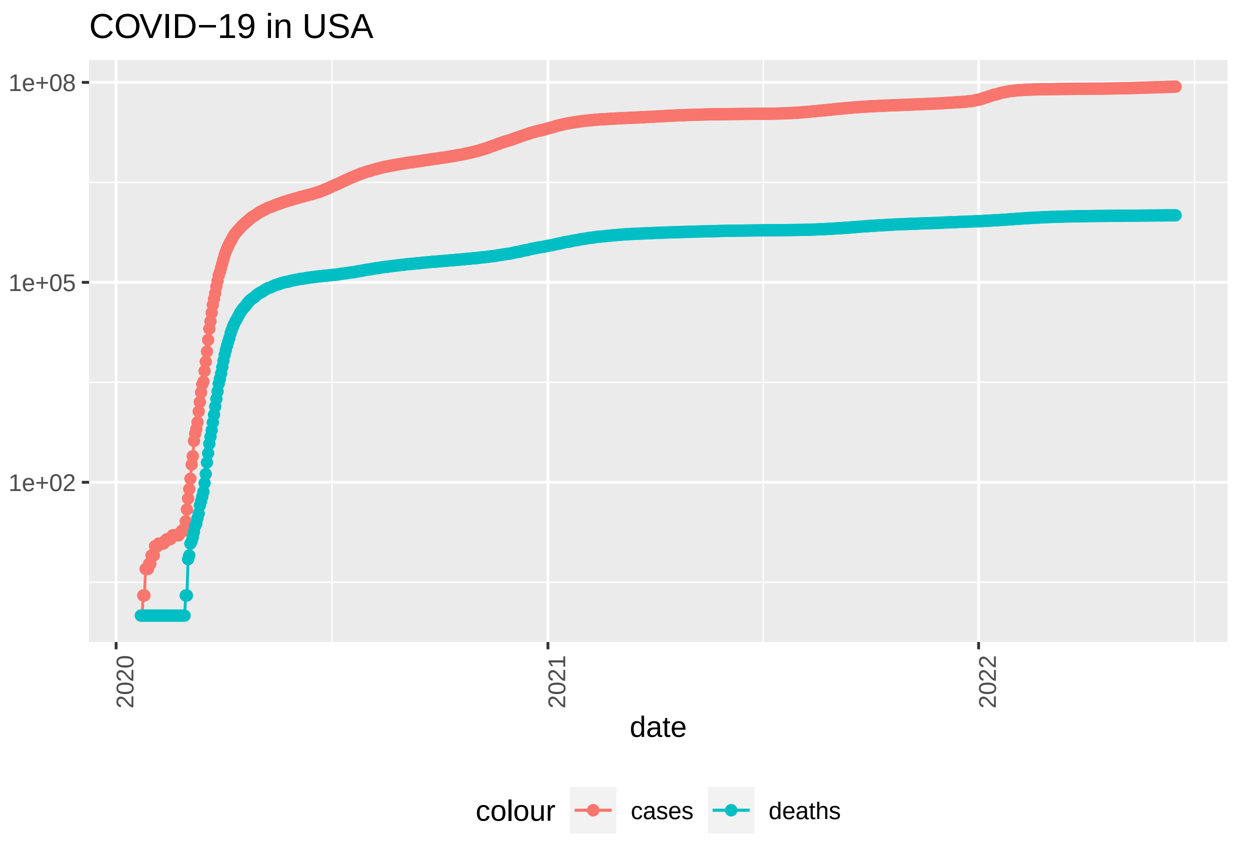
```
## ‘summarise()‘ has grouped output by ’Country_Region’. You can override using
## the ‘.groups‘ argument.
```

```r
US_totals
```

```
## # A tibble: 878 x 6
##    Country_Region date       cases deaths deaths_per_mill Population
##    <chr>          <date>     <dbl>  <dbl>           <dbl>      <dbl>
##  1 US             2020-01-22     1      1         0.00300  332875137
##  2 US             2020-01-23     1      1         0.00300  332875137
##  3 US             2020-01-24     2      1         0.00300  332875137
##  4 US             2020-01-25     2      1         0.00300  332875137
##  5 US             2020-01-26     5      1         0.00300  332875137
##  6 US             2020-01-27     5      1         0.00300  332875137
##  7 US             2020-01-28     5      1         0.00300  332875137
##  8 US             2020-01-29     6      1         0.00300  332875137
##  9 US             2020-01-30     6      1         0.00300  332875137
## 10 US             2020-01-31     8      1         0.00300  332875137
## # ... with 868 more rows
```

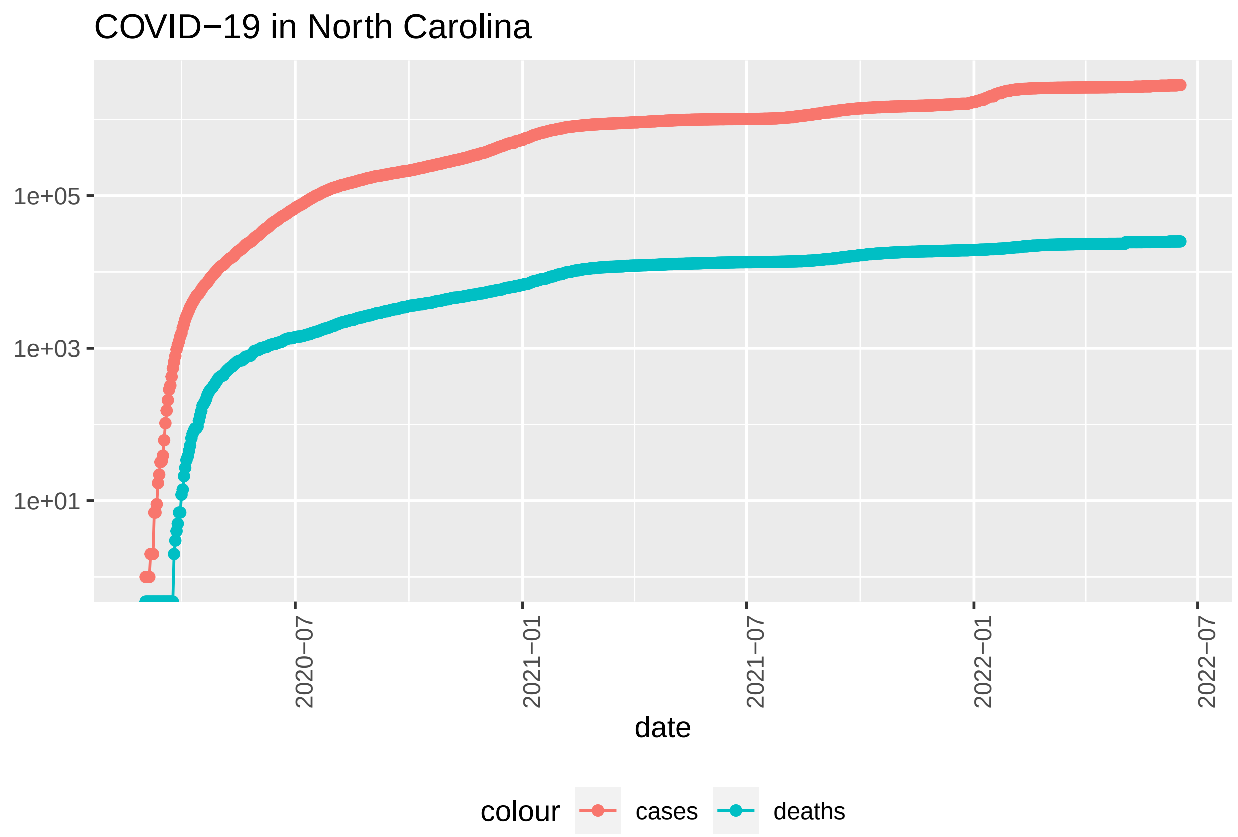
Let's quickly visualize the overall number of cases and deaths within the USA and the state of North Carolina. We'll use a logarithmic scale so early ups and downs aren't obscured by wilder variation later in the pandemic.

```r
US_totals %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = cases)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID-19 in USA", y = NULL)
```



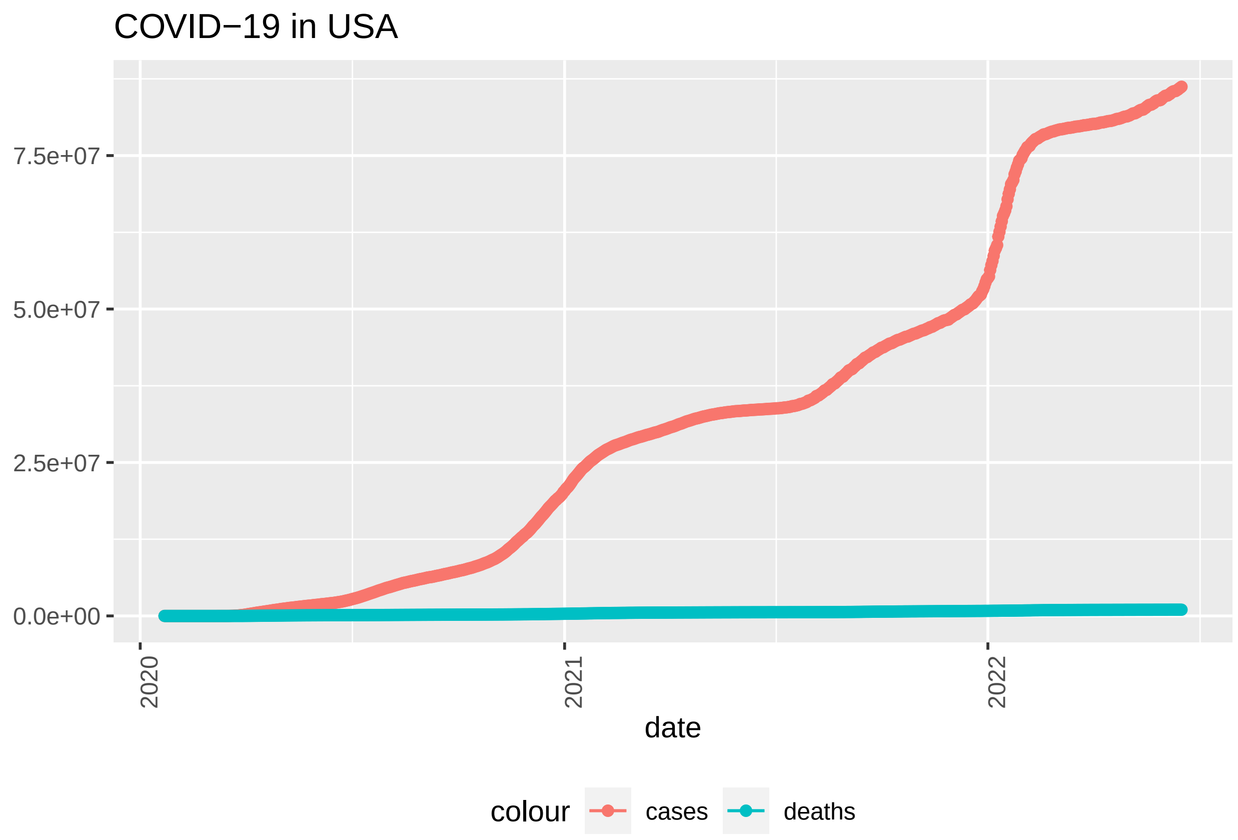
```
state <- "North Carolina"
US_by_state %>%
  filter(Province_State == state) %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = cases)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = str_c("COVID-19 in ", state), y = NULL)
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
## Transformation introduced infinite values in continuous y-axis
```

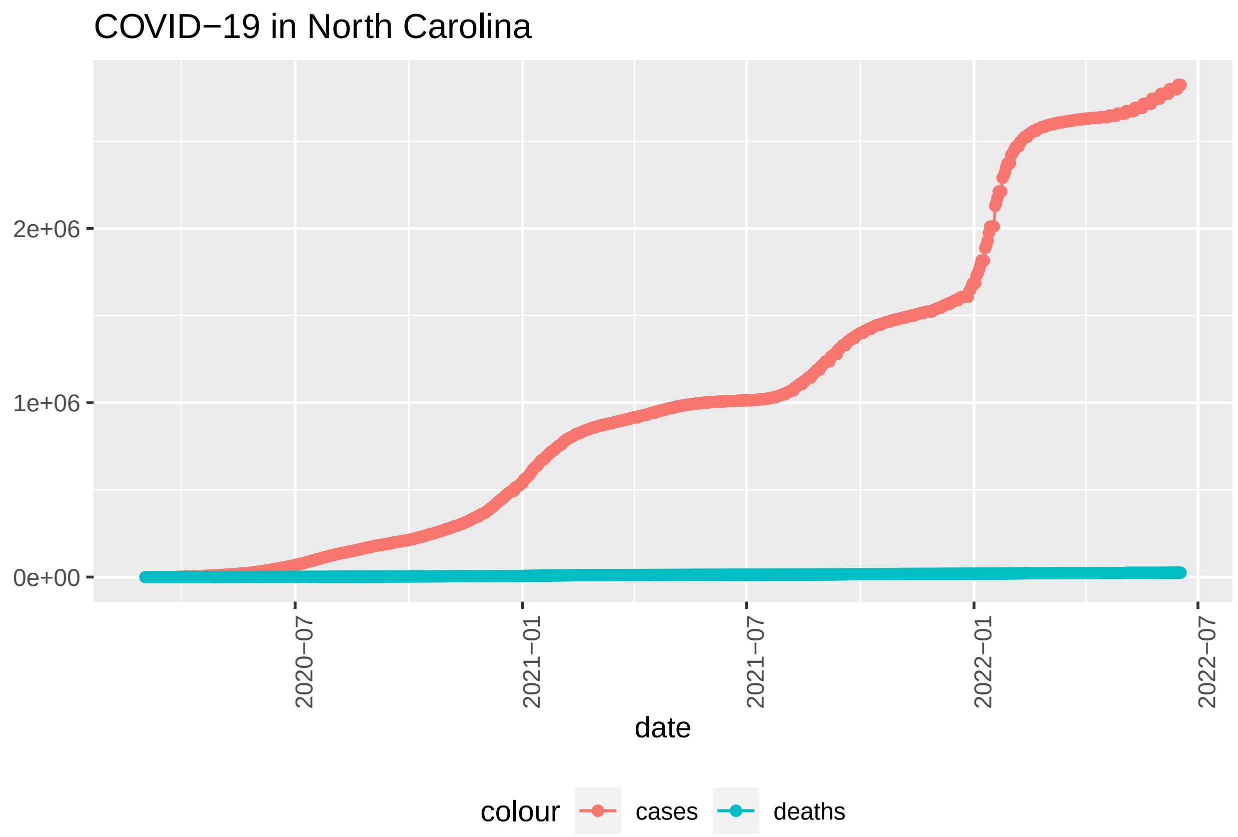


These are interesting, but they really obscure the surges from the Delta and Omicron variants that occurred in late 2021 and early 2022. Let's take off the logarithmic scale and regenerate the graphs.

```
US_totals %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = cases)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) +
  #scale_y_log10() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID-19 in USA", y = NULL)
```



```
state <- "North Carolina"
US_by_state %>%
  filter(Province_State == state) %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = cases)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) +
  #scale_y_log10() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = str_c("COVID-19 in ", state), y = NULL)
```

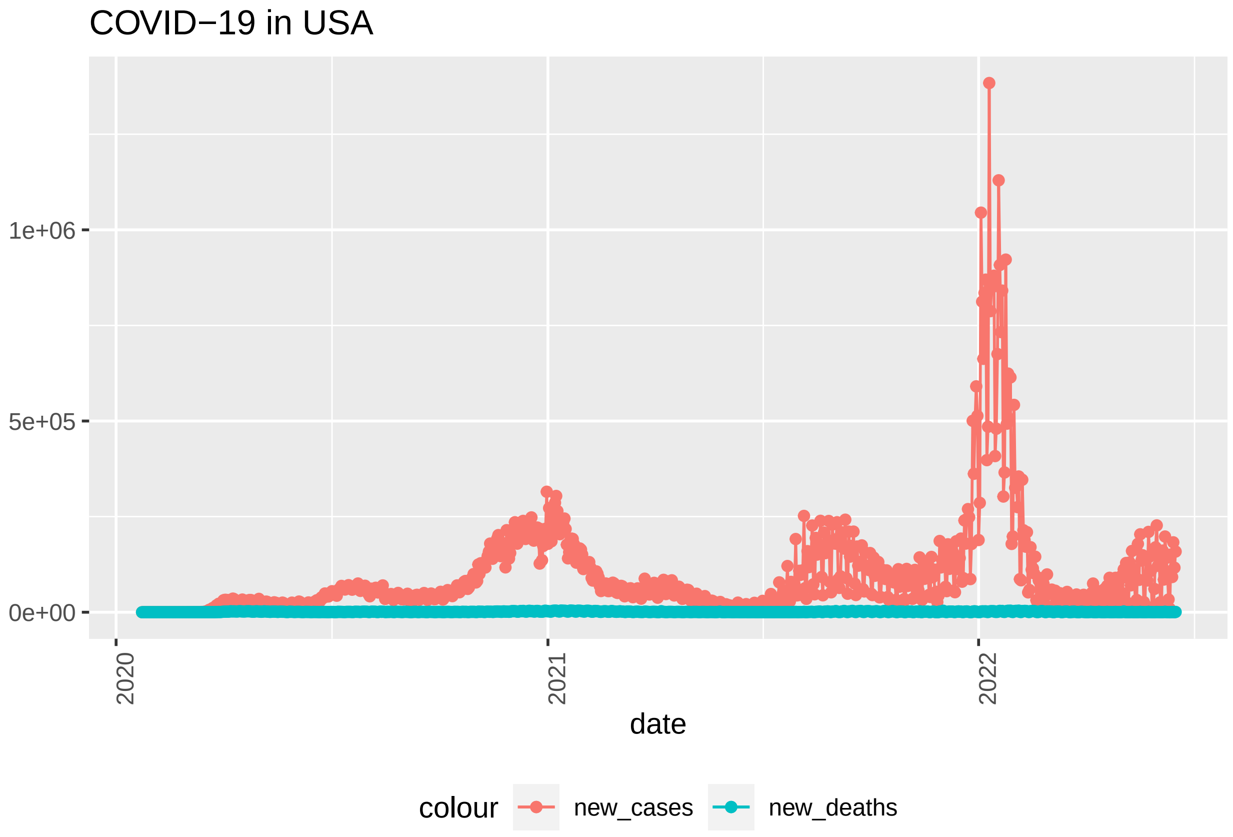


These clearly show the COVID-19 case count in the USA and in North Carolina increasing through early 2021, at which point vaccinations became available. The total case counts started to flatten until the Delta variant became prominent starting in the fall of 2021, then flattened again briefly before an enormous spike from the Omicron variant in the beginning of 2022.

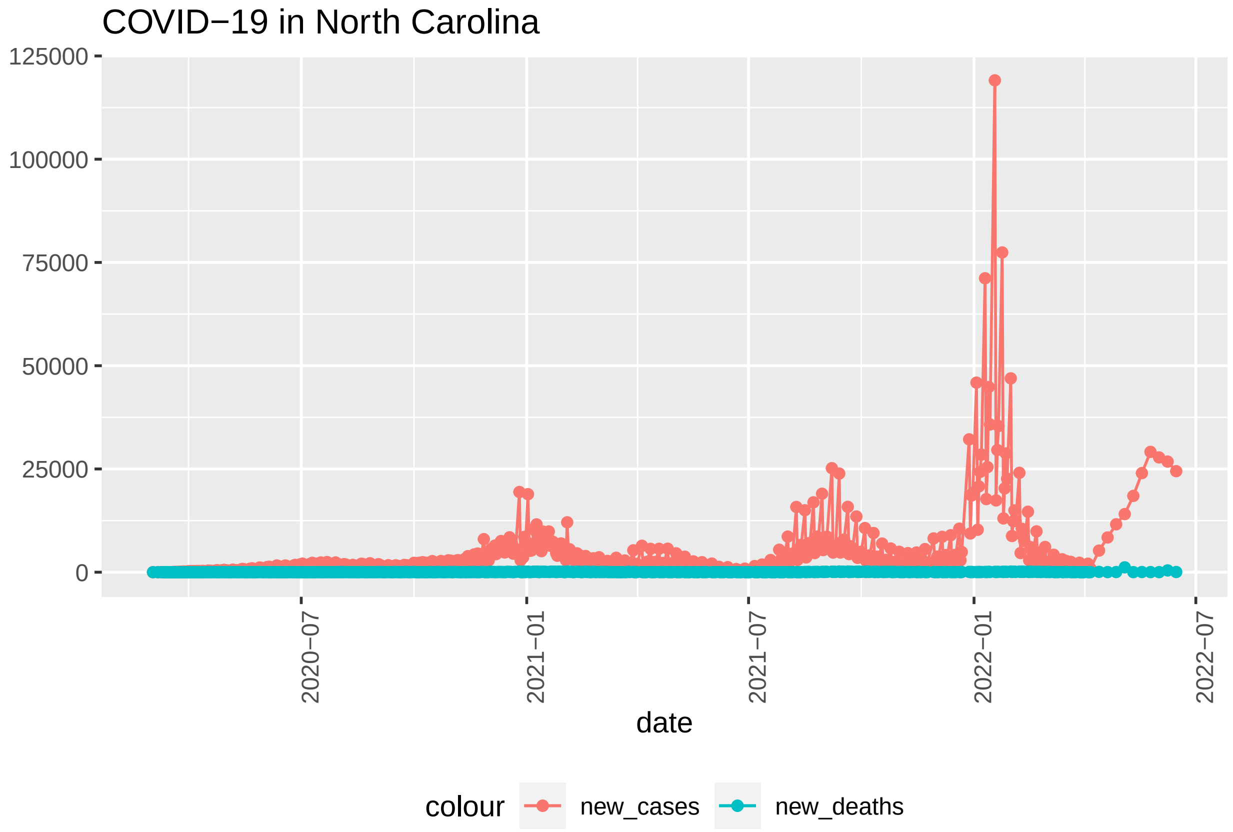
We can visualize this even more clearly if we only look at the incremental number of new cases and deaths per date (rather than the totals of each). To do this, we'll add new_cases and new_deaths columns to the datasets, which we can then plot.

```
US_by_state <- US_by_state %>%
  mutate(new_cases = cases - lag(cases),
         new_deaths = deaths - lag(deaths))
US_totals <- US_totals %>%
  mutate(new_cases = cases - lag(cases),
         new_deaths = deaths - lag(deaths))


US_totals %>%
  ggplot(aes(x = date, y = new_cases)) +
  geom_line(aes(color = "new_cases")) +
  geom_point(aes(color = "new_cases")) +
  geom_line(aes(y = new_deaths, color = "new_deaths")) +
  geom_point(aes(y = new_deaths, color = "new_deaths")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID-19 in USA", y = NULL)
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).

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

## Warning: Removed 1 row(s) containing missing values (geom_path).

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



```
US_by_state %>%
  filter(Province_State == state) %>%
  filter(new_cases > 0) %>%
  ggplot(aes(x = date, y = new_cases)) +
  geom_line(aes(color = "new_cases")) +
  geom_point(aes(color = "new_cases")) +
  geom_line(aes(y = new_deaths, color = "new_deaths")) +
  geom_point(aes(y = new_deaths, color = "new_deaths")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = str_c("COVID-19 in ", state), y = NULL)
```

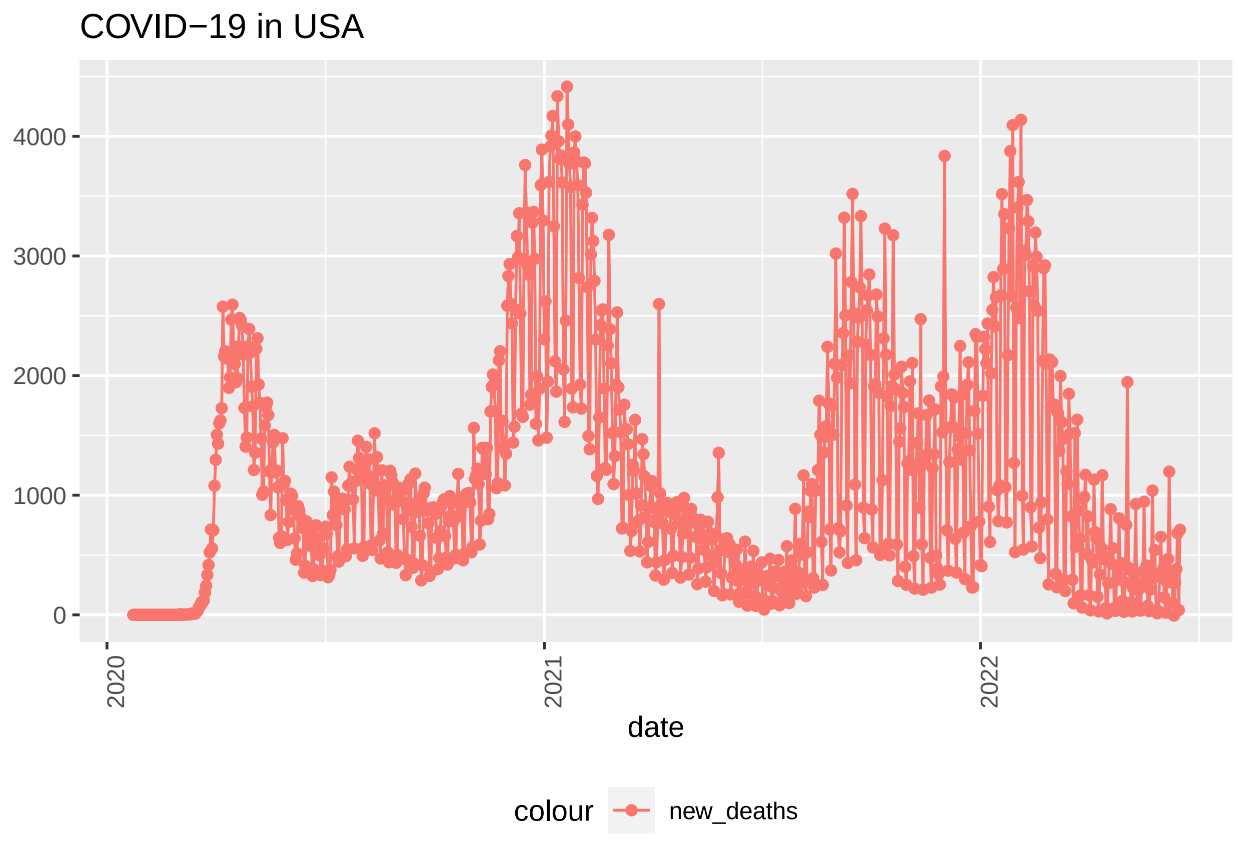


These plots clearly show the spikes for the Delta and Omicron variants, though the death counts are buried at the bottom of the graph since their numbers are so much lower than the case counts. Let's look at just the death numbers to see if we find the same spikes.

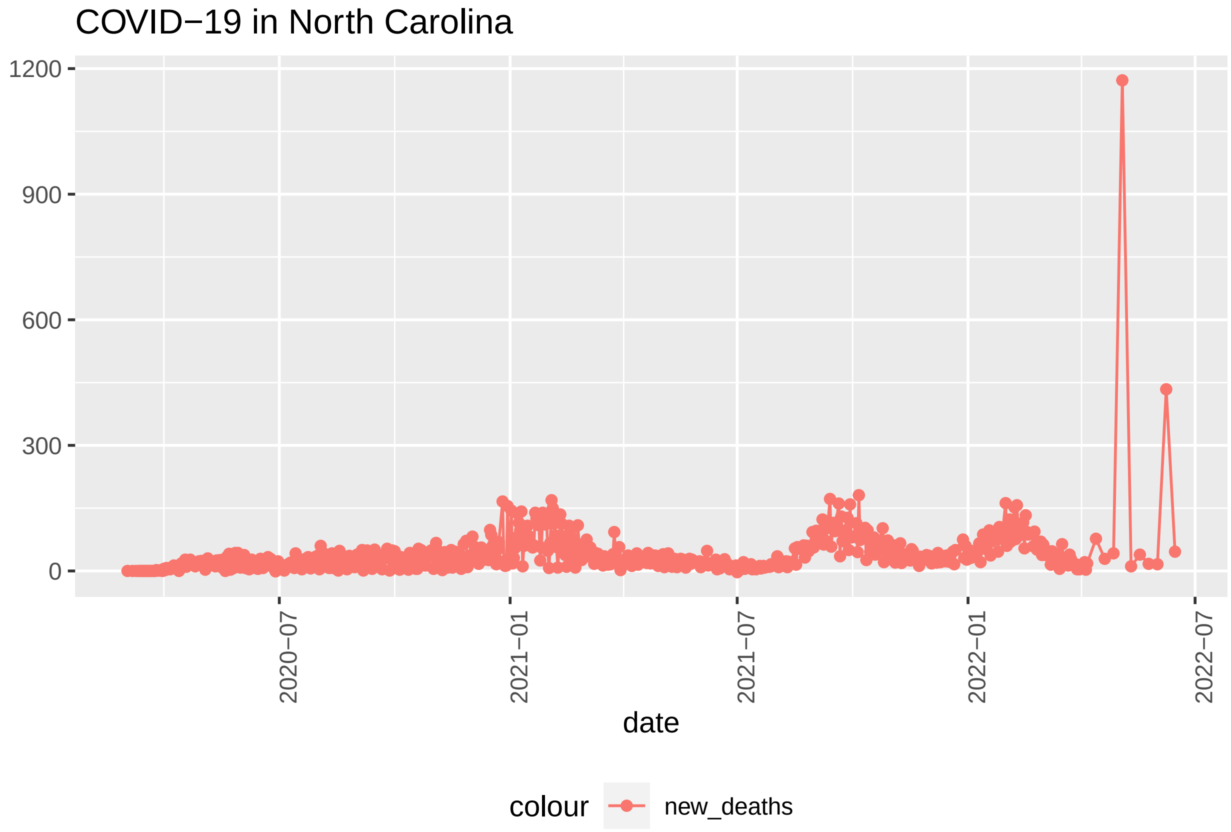
```
US_totals %>%
  ggplot(aes(x = date, y = new_deaths)) +
  geom_line(aes(y = new_deaths, color = "new_deaths")) +
  geom_point(aes(y = new_deaths, color = "new_deaths")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID-19 in USA", y = NULL)
```

```
## Warning: Removed 1 row(s) containing missing values (geom_path).
```

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



```
US_by_state %>%
  filter(Province_State == state) %>%
  filter(new_cases > 0) %>%
  ggplot(aes(x = date, y = new_deaths)) +
  geom_line(aes(y = new_deaths, color = "new_deaths")) +
  geom_point(aes(y = new_deaths, color = "new_deaths")) +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = str_c("COVID-19 in ", state), y = NULL)
```



We can clearly see the spikes for the Delta and Omicron variants in these plots, but the Omicron death counts are roughly the same as those from Delta, and not much larger as the Omicron case count was compared to the Delta case count.

This supports the common assertion that, although Omicron was much more widespread, it was a 'safer' form of the virus that did not result in as many per capita deaths as the earlier variants.

There are a couple of data points for North Carolina in mid-2022 that seem to be outliers. Could these just be due to a lag in reporting earlier deaths? This is an issue for further investigation.

North Carolina vs National

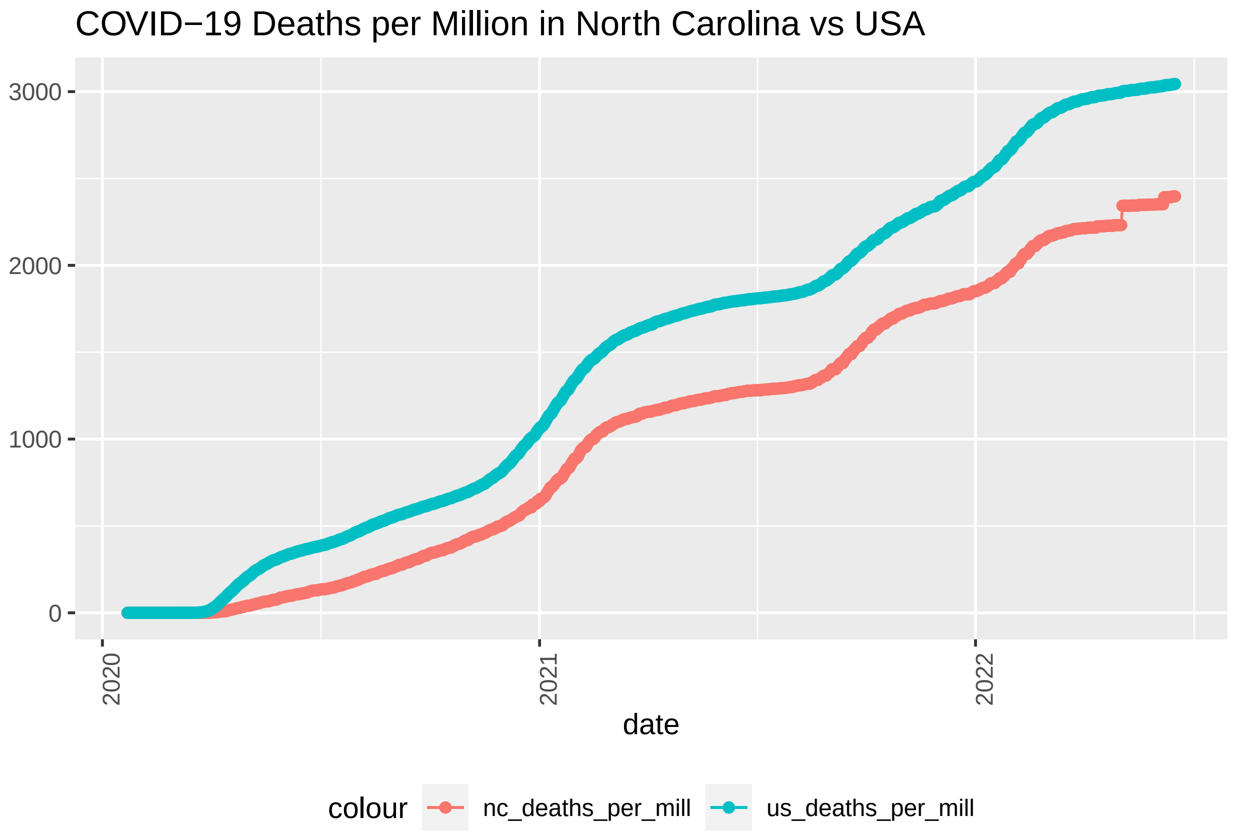
How did the state of North Carolina do in comparison to the whole country?

We'll create a new tibble, joining NC data against the US data by date. We'll rename the columns so we can distinguish the NC values from the US values, and then plot them together.

```
NC_vs_US <- US_by_state %>%
  filter(Province_State == "North Carolina") %>%
  rename(nc_cases = cases, nc_deaths = deaths,
         nc_deaths_per_mill = deaths_per_mill, nc_new_cases = new_cases,
         nc_new_deaths = new_deaths, nc_population = Population) %>%
  select(date:nc_new_deaths) %>%
  full_join(US_totals) %>%
  select(-c(Country_Region)) %>%
  rename(us_cases = cases, us_deaths = deaths,
         us_deaths_per_mill = deaths_per_mill, us_population = Population,
         us_new_cases = new_cases, us_new_deaths = new_deaths)
```

```
## Joining, by = "date"
```

```
NC_vs_US %>%
  ggplot(aes(x = date, y = nc_deaths_per_mill)) +
  geom_line(aes(color = "nc_deaths_per_mill")) +
  geom_point(aes(color = "nc_deaths_per_mill")) +
  geom_line(aes(y = us_deaths_per_mill, color = "us_deaths_per_mill")) +
  geom_point(aes(y = us_deaths_per_mill, color = "us_deaths_per_mill")) +
  #scale_y_log10() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID-19 Deaths per Million in North Carolina vs USA", y = NULL)
```



This seems to indicate that North Carolina has had a lower number of deaths per million residents than the overall nation throughout the pandemic.

Modeling NC vs US

Let's generate a linear model between NC and US deaths per million residents to see how well the NC rate predicts the US rate.

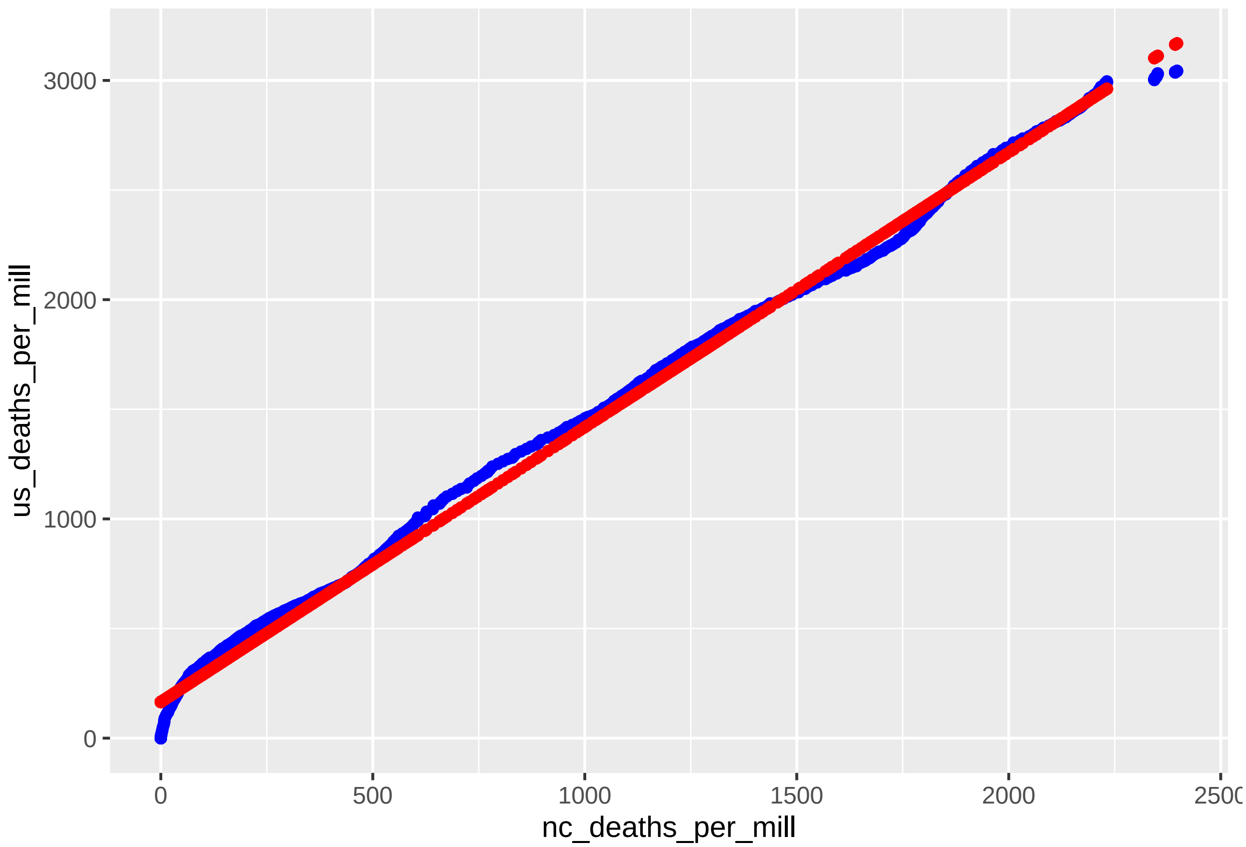
```
mod_nc_us <- lm(us_deaths_per_mill ~ nc_deaths_per_mill, data = NC_vs_US)
summary(mod_nc_us)
```

```
##
## Call:
## lm(formula = us_deaths_per_mill ~ nc_deaths_per_mill, data = NC_vs_US)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -165.28  -26.50   27.10   45.55   93.97
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.653e+02  3.818e+00   43.29   <2e-16 ***
## nc_deaths_per_mill 1.253e+00  2.925e-03  428.44   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 67.74 on 876 degrees of freedom
## Multiple R-squared:  0.9953, Adjusted R-squared:  0.9952
## F-statistic: 1.836e+05 on 1 and 876 DF,  p-value: < 2.2e-16
```

This shows a very high correlation between the two, with a p-value less than 2.2e-16.

Let's plot the model to see the relationship visually.

```
NC_vs_US_w_pred <- NC_vs_US %>% mutate(pred = predict(mod_nc_us))
NC_vs_US_w_pred %>%
  ggplot() +
  geom_point(
    aes(x = nc_deaths_per_mill, y = us_deaths_per_mill), color = "blue") +
  geom_point(aes(x = nc_deaths_per_mill, y = pred), color="red")
```



Indeed, we can see a very strong linear relationship between the number of deaths per million in North Carolina versus in the USA as a whole. The plot winds very tightly back and forth across the linear model.

Conclusion

COVID-19 cases have spiked at times, likely due to appearances of highly-contagious Delta and Omicron variants. The state of North Carolina has experiences highs and lows in parallel with overall USA data, with perhaps a slightly better-than-average death count per million residents. Omicron, per our analysis, does indeed to be a safer variant of the virus as its death counts have not risen in parallel with case counts, as was the case for the original virus and the Delta variant.

Before drawing firm conclusions, we would have to investigate potential sources of bias in the underlying data.

Potential Bias

There are many possible sources of bias in this analysis, including:

1. COVID-19 numbers are reported via agencies within each state, who may have different procedures for collecting the data within that state. This could obscure the actual relationship between COVID-19 cases and deaths across states.
2. We excluded data points with zero or negative numbers of cases and death rates. Why were those there? Were they an attempt to correct errors in data previously reported? This should be investigated.

3. As home testing has become more available, fewer people are testing through health care facilities or laboratories. Positive results from home tests are not necessarily reported to relevant state agencies. Therefore, the actual case numbers are likely higher than are reflected in the data for the time periods when home tests are widely used.
4. Were the “outlier” data points in the NC death counts from spring of 2022 erroneous? These numbers should be confirmed.

Session Information

```
sessionInfo()
```

```
## R version 4.2.0 (2022-04-22)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 20.04.4 LTS
##
## Matrix products: default
## BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
## LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/liblapack.so.3
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.8.0 forcats_0.5.1   stringr_1.4.0   dplyr_1.0.9
##  [5] purrr_0.3.4     readr_2.1.2     tidyr_1.2.0     tibble_3.1.7
##  [9] ggplot2_3.3.6   tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.2 xfun_0.31        haven_2.5.0      colorspace_2.0-3
##  [5] vctrs_0.4.1      generics_0.1.2   htmltools_0.5.2  yaml_2.3.5
##  [9] utf8_1.2.2       rlang_1.0.2      pillar_1.7.0     glue_1.6.2
## [13] withr_2.5.0      DBI_1.1.2        bit64_4.0.5      dbplyr_2.2.0
## [17] modelr_0.1.8     readxl_1.4.0     lifecycle_1.0.1  munsell_0.5.0
## [21] gtable_0.3.0     cellranger_1.1.0 rvest_1.0.2      evaluate_0.15
## [25] labeling_0.4.2   knitr_1.39       tzdb_0.3.0       fastmap_1.1.0
## [29] curl_4.3.2       parallel_4.2.0   fansi_1.0.3      highr_0.9
## [33] broom_0.8.0      backports_1.4.1  scales_1.2.0     vroom_1.5.7
## [37] jsonlite_1.8.0   farver_2.1.0     bit_4.0.4        fs_1.5.2
## [41] hms_1.1.1        digest_0.6.29    stringi_1.7.6    grid_4.2.0
## [45] cli_3.3.0        tools_4.2.0      magrittr_2.0.3   crayon_1.5.1
## [49] pkgconfig_2.0.3  ellipsis_0.3.2   xml2_1.3.3       reprex_2.0.1
## [53] assertthat_0.2.1 rmarkdown_2.14   httr_1.4.3       rstudioapi_0.13
## [57] R6_2.5.1         compiler_4.2.0
```