Johns_Hopkins_Covid19_Data_Project

Conrad Kleykamp

2023-02-13

Introduction

This report will analyze COVID-19 data which was pulled from the Johns Hopkins University Center for Systems Science and Engineering data repository. The data is available on Github and is intended for public use. This analysis will explore trends of COVID cases and deaths across time in the United States. Will there be any significant or notable trends across the years?

Setup

We will first load the necessary packages for this analysis. Afterwards, we will read in the URL and assign variable names to each data set.

```r
# Load tidyverse for future use
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.4.0     v purrr   1.0.1
## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```r
# Load lubridate for future use
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```r
# Read in the URL from Github
url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_cov
```

```r
# Read in file_names
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")
```

```r
# Create vector of the four urls
urls <- str_c(url_in, file_names)
urls
```

```
## [1] "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_
## [2] "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_
## [3] "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_
## [4] "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_
```

```r
# Read in data sets and assign variable names
# This will give us four data sets to analyze
global_cases <- read_csv(urls[1])
```

```
## Rows: 289 Columns: 1122
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1120): 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.
```

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

```
## Rows: 289 Columns: 1122
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1120): 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.
```

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

```
## Rows: 3342 Columns: 1129
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1123): 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: 1130
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1124): 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.
```

Clean, Tidy, and Transform Data

In this step, we will work to clean, tidy, and transform our data sets. This will enable ease of use in our future analyses.

```
# Pivot the global_cases data set, filter out unwanted columns
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_cases
```

```
## # A tibble: 323,102 x 4
##    ‘Province/State‘ ‘Country/Region‘ Date    Cases
##    <chr>            <chr>            <chr>   <dbl>
##  1 <NA>             Afghanistan      1/22/20     0
##  2 <NA>             Afghanistan      1/23/20     0
##  3 <NA>             Afghanistan      1/24/20     0
##  4 <NA>             Afghanistan      1/25/20     0
##  5 <NA>             Afghanistan      1/26/20     0
##  6 <NA>             Afghanistan      1/27/20     0
##  7 <NA>             Afghanistan      1/28/20     0
##  8 <NA>             Afghanistan      1/29/20     0
##  9 <NA>             Afghanistan      1/30/20     0
## 10 <NA>             Afghanistan      1/31/20     0
## # ... with 323,092 more rows
```

```
# Pivot the global_deaths data set, filter out unwanted columns
global_deaths <- global_deaths %>%
  pivot_longer(cols = -c('Province/State','Country/Region', Lat, Long),
               names_to = "Date",
               values_to = "Deaths") %>%
  select(-c(Lat, Long))
global_deaths
```

```
## # A tibble: 323,102 x 4
##    ‘Province/State‘ ‘Country/Region‘ Date    Deaths
##    <chr>            <chr>            <chr>    <dbl>
##  1 <NA>             Afghanistan      1/22/20      0
##  2 <NA>             Afghanistan      1/23/20      0
##  3 <NA>             Afghanistan      1/24/20      0
##  4 <NA>             Afghanistan      1/25/20      0
##  5 <NA>             Afghanistan      1/26/20      0
##  6 <NA>             Afghanistan      1/27/20      0
##  7 <NA>             Afghanistan      1/28/20      0
##  8 <NA>             Afghanistan      1/29/20      0
##  9 <NA>             Afghanistan      1/30/20      0
## 10 <NA>             Afghanistan      1/31/20      0
## # ... with 323,092 more rows
```

```
# Combine global_cases and global_deaths into a single variable
# Rename columns and format to mdy
global <- global_cases %>%
  full_join(global_deaths) %>%
  rename(Country_Region = 'Country/Region',
         Province_State = 'Province/State') %>%
  mutate(Date = mdy(Date))
```

```
## Joining with ‘by = join_by(‘Province/State‘, ‘Country/Region‘, Date)‘
```

```
global
```

```
## # A tibble: 323,102 x 5
##    Province_State Country_Region Date       Cases Deaths
##    <chr>          <chr>          <date>     <dbl>  <dbl>
##  1 <NA>           Afghanistan    2020-01-22     0      0
##  2 <NA>           Afghanistan    2020-01-23     0      0
##  3 <NA>           Afghanistan    2020-01-24     0      0
##  4 <NA>           Afghanistan    2020-01-25     0      0
##  5 <NA>           Afghanistan    2020-01-26     0      0
##  6 <NA>           Afghanistan    2020-01-27     0      0
##  7 <NA>           Afghanistan    2020-01-28     0      0
##  8 <NA>           Afghanistan    2020-01-29     0      0
##  9 <NA>           Afghanistan    2020-01-30     0      0
## 10 <NA>           Afghanistan    2020-01-31     0      0
## # ... with 323,092 more rows
```

```
# Filter out dates with zero cases
global <- global %>%
  filter(Cases > 0)
summary(global)
```

```
##  Province_State     Country_Region          Date                Cases
##  Length:299652      Length:299652      Min.   :2020-01-22   Min.   :        1
##  Class :character   Class :character   1st Qu.:2020-12-06   1st Qu.:     1262
##  Mode  :character   Mode  :character   Median :2021-09-03   Median :    19473
##                                        Mean   :2021-08-29   Mean   :  1001285
##                                        3rd Qu.:2022-05-27   3rd Qu.:   264002
##                                        Max.   :2023-02-12   Max.   :102849008
##      Deaths
##  Min.   :      0
##  1st Qu.:      7
##  Median :    205
##  Mean   :  14176
##  3rd Qu.:   3594
##  Max.   :1114377
```

```
# Pivot the US_cases data set, filter out unwanted columns
# Format to mdy
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_cases
```

```
## # A tibble: 3,736,356 x 6
##    Admin2  Province_State Country_Region Combined_Key         Date       Cases
##    <chr>   <chr>          <chr>          <chr>                <date>     <dbl>
##  1 Autauga Alabama        US             Autauga, Alabama, US 2020-01-22     0
##  2 Autauga Alabama        US             Autauga, Alabama, US 2020-01-23     0
##  3 Autauga Alabama        US             Autauga, Alabama, US 2020-01-24     0
##  4 Autauga Alabama        US             Autauga, Alabama, US 2020-01-25     0
##  5 Autauga Alabama        US             Autauga, Alabama, US 2020-01-26     0
##  6 Autauga Alabama        US             Autauga, Alabama, US 2020-01-27     0
##  7 Autauga Alabama        US             Autauga, Alabama, US 2020-01-28     0
##  8 Autauga Alabama        US             Autauga, Alabama, US 2020-01-29     0
##  9 Autauga Alabama        US             Autauga, Alabama, US 2020-01-30     0
## 10 Autauga Alabama        US             Autauga, Alabama, US 2020-01-31     0
## # ... with 3,736,346 more rows
```

```
# Pivot the US_deaths data set, filter out unwanted columns
# Format to mdy
US_deaths <- US_deaths %>%
  pivot_longer(cols = -(UID:Population),
               names_to = "Date",
               values_to = "Deaths") %>%
  select(Admin2:Deaths) %>%
  mutate(Date = mdy(Date)) %>%
  select(-c(Lat, Long_))
US_deaths
```

```
## # A tibble: 3,736,356 x 7
##    Admin2  Province_State Country_Region Combined_Key  Popul~1 Date       Deaths
##    <chr>   <chr>          <chr>          <chr>           <dbl> <date>      <dbl>
##  1 Autauga Alabama        US             Autauga, Ala~   55869 2020-01-22      0
##  2 Autauga Alabama        US             Autauga, Ala~   55869 2020-01-23      0
##  3 Autauga Alabama        US             Autauga, Ala~   55869 2020-01-24      0
##  4 Autauga Alabama        US             Autauga, Ala~   55869 2020-01-25      0
##  5 Autauga Alabama        US             Autauga, Ala~   55869 2020-01-26      0
##  6 Autauga Alabama        US             Autauga, Ala~   55869 2020-01-27      0
##  7 Autauga Alabama        US             Autauga, Ala~   55869 2020-01-28      0
##  8 Autauga Alabama        US             Autauga, Ala~   55869 2020-01-29      0
##  9 Autauga Alabama        US             Autauga, Ala~   55869 2020-01-30      0
## 10 Autauga Alabama        US             Autauga, Ala~   55869 2020-01-31      0
## # ... with 3,736,346 more rows, and abbreviated variable name 1: Population
```

```
# Combine US_cases and US_deaths into a single variable
US <- US_cases %>%
  full_join(US_deaths)
```

```
## Joining with `by = join_by(Admin2, Province_State, Country_Region,
## Combined_Key, Date)`
```

```
US
```

```
## # A tibble: 3,736,356 x 8
##    Admin2  Province_State Country_Region Combi~1 Date       Cases Popul~2 Deaths
##    <chr>   <chr>          <chr>          <chr>   <date>     <dbl>   <dbl>  <dbl>
##  1 Autauga Alabama        US             Autaug~ 2020-01-22     0   55869      0
##  2 Autauga Alabama        US             Autaug~ 2020-01-23     0   55869      0
##  3 Autauga Alabama        US             Autaug~ 2020-01-24     0   55869      0
##  4 Autauga Alabama        US             Autaug~ 2020-01-25     0   55869      0
##  5 Autauga Alabama        US             Autaug~ 2020-01-26     0   55869      0
##  6 Autauga Alabama        US             Autaug~ 2020-01-27     0   55869      0
##  7 Autauga Alabama        US             Autaug~ 2020-01-28     0   55869      0
##  8 Autauga Alabama        US             Autaug~ 2020-01-29     0   55869      0
##  9 Autauga Alabama        US             Autaug~ 2020-01-30     0   55869      0
## 10 Autauga Alabama        US             Autaug~ 2020-01-31     0   55869      0
## # ... with 3,736,346 more rows, and abbreviated variable names 1: Combined_Key,
## #   2: Population
```

```
# For comparative analysis between countries, we will add the
# population data to the global data set
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/U

uid <- read_csv(uid_lookup_url) %>%
  select(-c(Lat, Long_, Combined_Key, code3, iso2, iso3, Admin2))
```

```
## Rows: 4321 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)
global
```

```
## # A tibble: 299,652 x 7
##    Province_State Country_Region Date       Cases Deaths Population Combined_Key
##    <chr>          <chr>          <date>     <dbl>  <dbl>      <dbl> <chr>
##  1 <NA>           Afghanistan    2020-02-24     5      0   38928341 Afghanistan
##  2 <NA>           Afghanistan    2020-02-25     5      0   38928341 Afghanistan
##  3 <NA>           Afghanistan    2020-02-26     5      0   38928341 Afghanistan
##  4 <NA>           Afghanistan    2020-02-27     5      0   38928341 Afghanistan
##  5 <NA>           Afghanistan    2020-02-28     5      0   38928341 Afghanistan
##  6 <NA>           Afghanistan    2020-02-29     5      0   38928341 Afghanistan
##  7 <NA>           Afghanistan    2020-03-01     5      0   38928341 Afghanistan
##  8 <NA>           Afghanistan    2020-03-02     5      0   38928341 Afghanistan
##  9 <NA>           Afghanistan    2020-03-03     5      0   38928341 Afghanistan
## 10 <NA>           Afghanistan    2020-03-04     5      0   38928341 Afghanistan
## # ... with 299,642 more rows
```

Prepare Data for Analysis

```
# Begin by analyzing US data by state
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: 64,844 x 7
##    Province_State Country_Region Date       Cases Deaths Deaths_per_mill Popul~1
##    <chr>          <chr>          <date>     <dbl>  <dbl>           <dbl>   <dbl>
##  1 Alabama        US             2020-01-22     0      0               0 4903185
##  2 Alabama        US             2020-01-23     0      0               0 4903185
##  3 Alabama        US             2020-01-24     0      0               0 4903185
##  4 Alabama        US             2020-01-25     0      0               0 4903185
##  5 Alabama        US             2020-01-26     0      0               0 4903185
##  6 Alabama        US             2020-01-27     0      0               0 4903185
##  7 Alabama        US             2020-01-28     0      0               0 4903185
##  8 Alabama        US             2020-01-29     0      0               0 4903185
##  9 Alabama        US             2020-01-30     0      0                0 4903185
## 10 Alabama        US             2020-01-31     0      0                0 4903185
## # ... with 64,834 more rows, and abbreviated variable name 1: Population
```

```
# Analyze US data by Country_Region and 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.
```

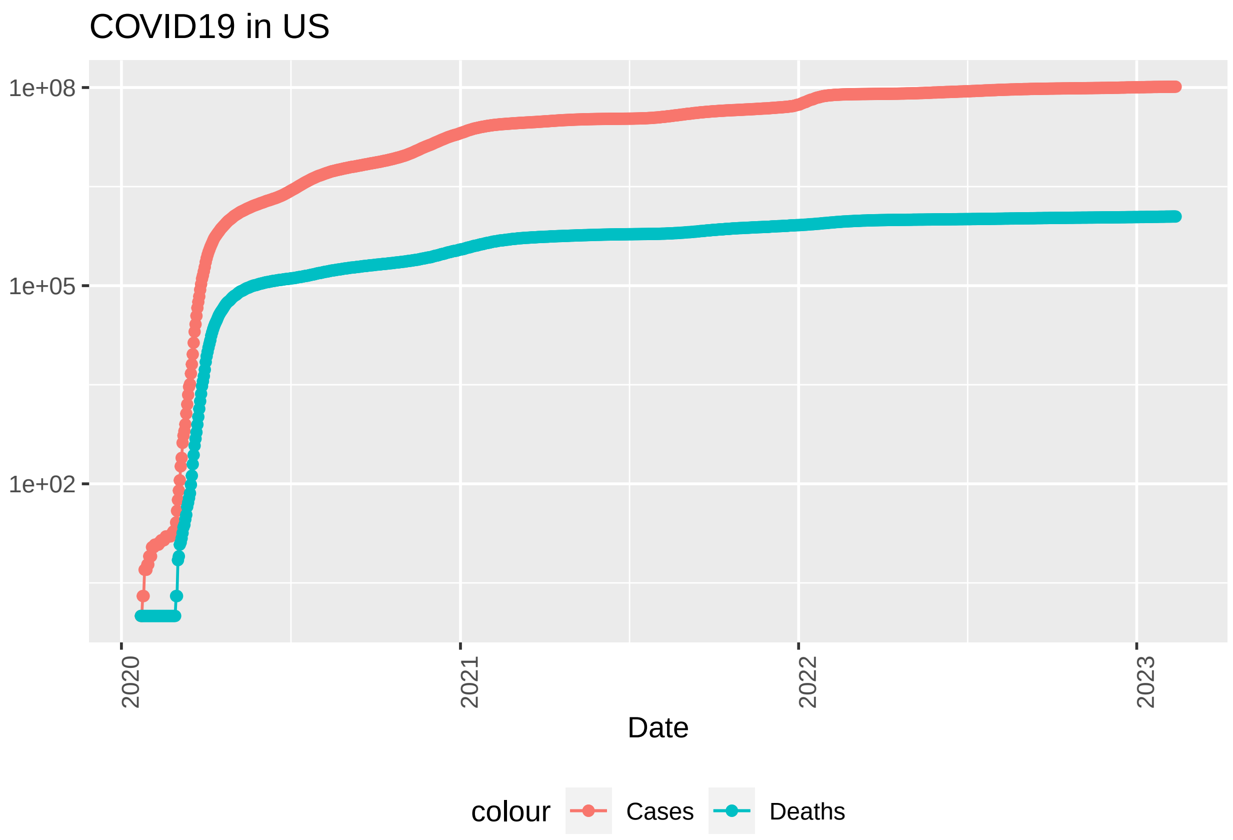
```
US_totals
```

```
## # A tibble: 1,118 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 1,108 more rows
```

Visualize Data

In this section, we will work to create visualizations of the number of cases and deaths in the US and the state of Massachusetts across time.

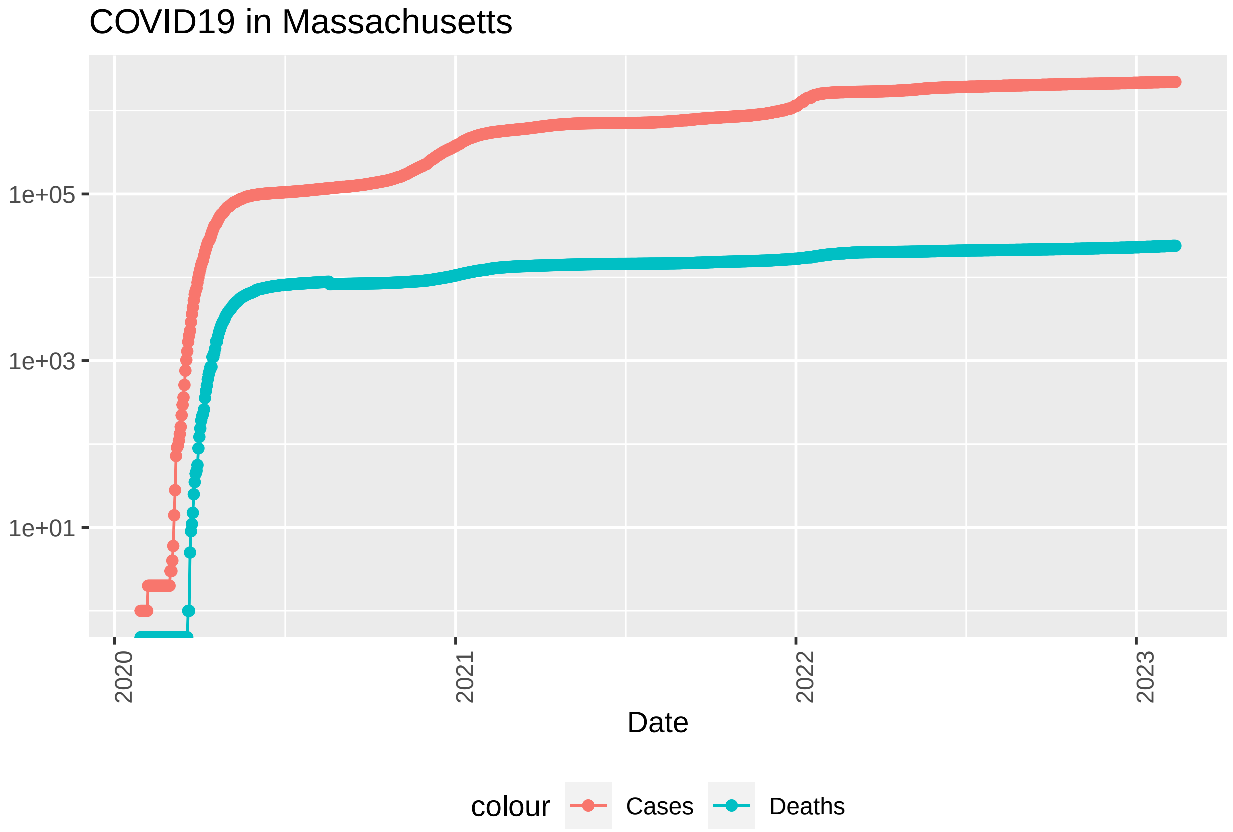
```
# Visualize the total number of cases and deaths in US across
# each year
US_totals %>%
  filter(Cases > 0) %>%
  ggplot(aes(x = Date, y = Cases)) +
  geom_line(aes(color = "Cases")) +
  geom_point(aes(color = "Cases")) +
  geom_line(aes(y = Deaths, color = "Deaths")) +
  geom_point(aes(y = Deaths, color = "Deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90))+
  labs(title = "COVID19 in US", y = NULL)
```



This visualization shows the number of cases (red) and deaths (blue) across time in the US. It is clear that there was a significant increase in the number of both cases and deaths during 2020. This rapid increase began to plateau towards the end of 2020 and onward.

```r
# Visualize total number of cases and deaths in Massachusetts
state <- "Massachusetts"
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 = "COVID19 in Massachusetts", y = NULL)
```

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



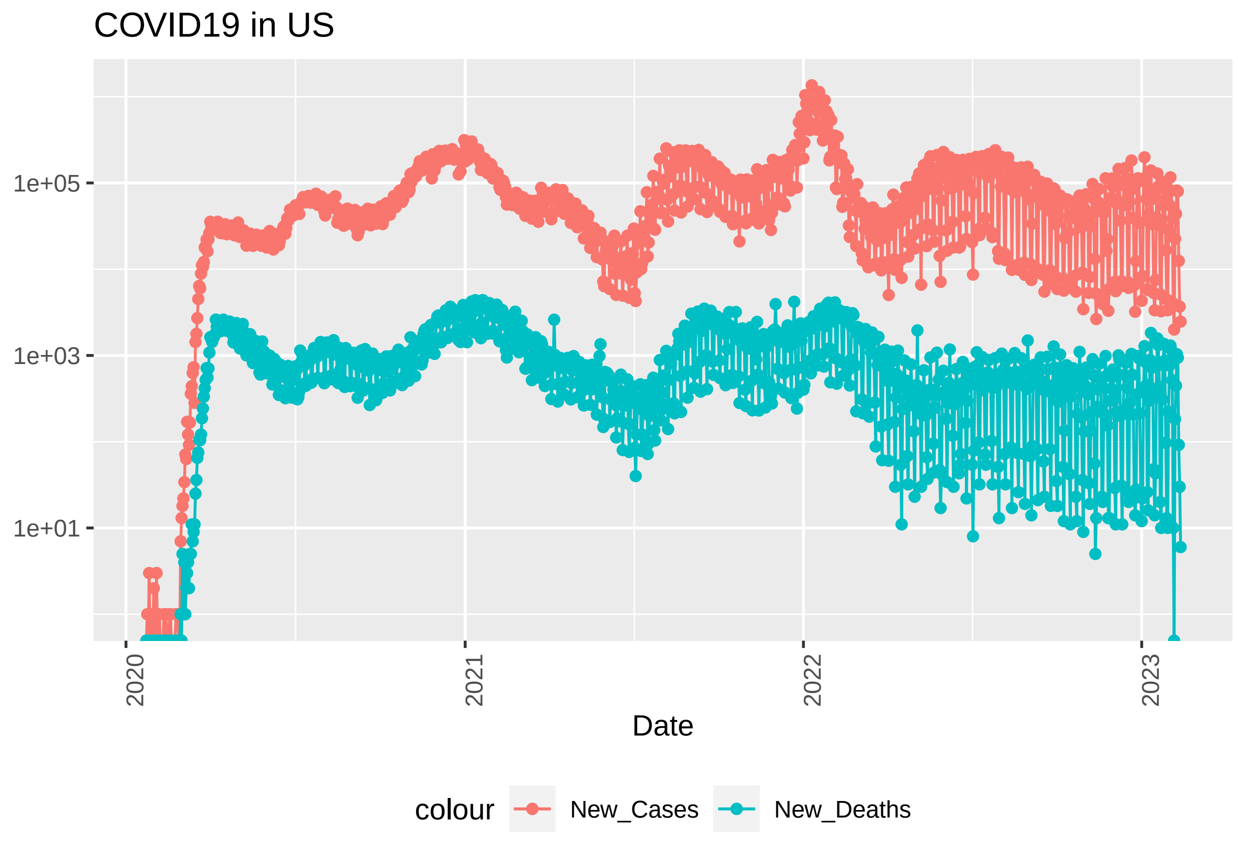
This visualization shows the number of cases (red) and deaths (blue) across time in the state of Massachusetts. The trend appears to be almost identical to the prior visualization. With Covid cases and deaths increasing rapidly but then beginning to plateau at the end of 2020.

In order to gain insight from the plateaus, we will create two new columns representing new cases and new deaths. These new reportings may uncover unique trends.

```
# New US data sets factor in the lag of new cases and deaths
# Two new columns
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")) +
  scale_y_log10() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90))+
  labs(title = "COVID19 in US", y = NULL)
```

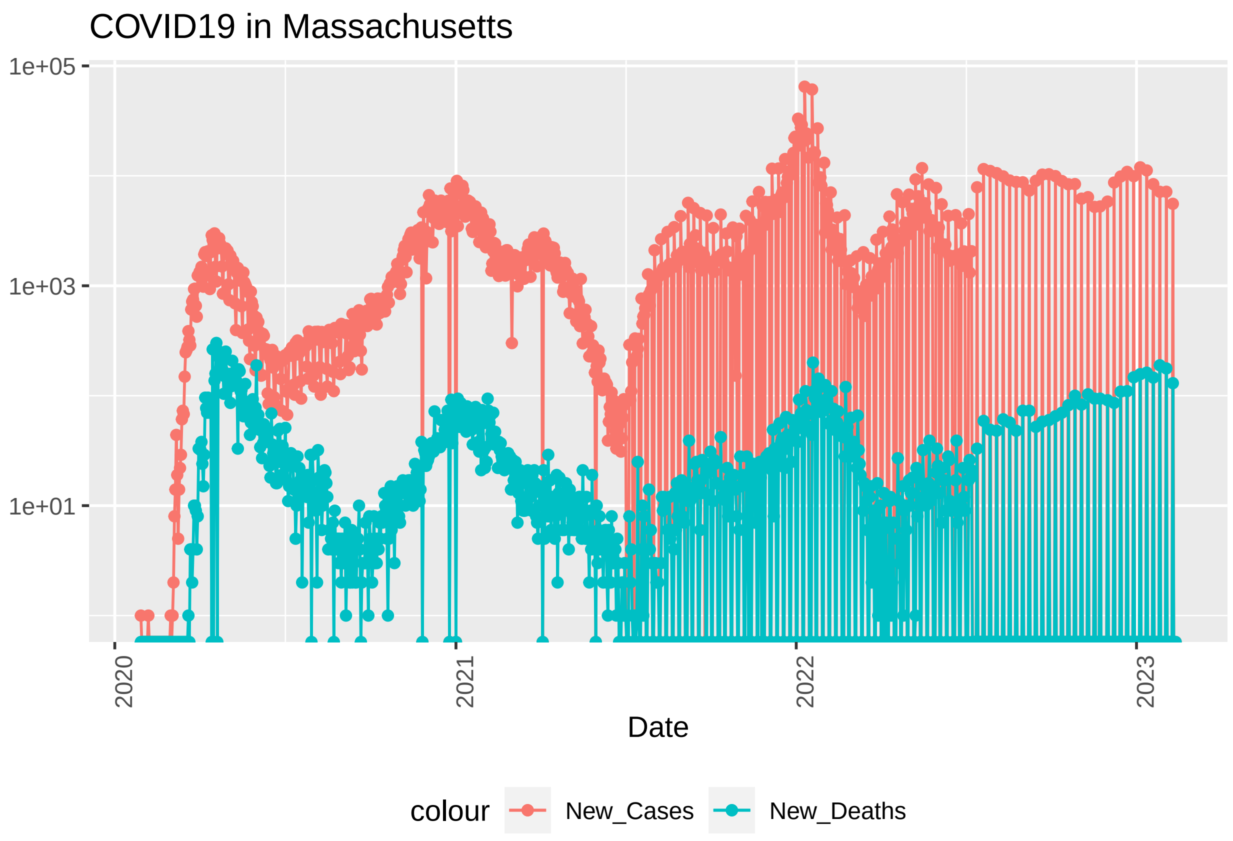
```
## Warning: Transformation introduced infinite values in continuous y-axis
## Transformation introduced infinite values in continuous y-axis

## Warning in self$trans$transform(x): NaNs produced

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

## Warning in self$trans$transform(x): NaNs produced

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

## Warning: Removed 1 row containing missing values (‘geom_line()‘).

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

## Warning: Removed 1 row containing missing values (‘geom_line()‘).

## Warning: Removed 2 rows containing missing values (‘geom_point()‘).
```


COVID19 in US

This visualization shows the number of new cases (red) and new deaths (blue) across time in the US. Similar to the prior set of visualizations, there was a significant increase in cases and deaths during the first half of 2020. However, we can identify new trends here. First, we can see that the number of new cases and new deaths dips significantly halfway through 2021. This may have resulted from the introduction of the vaccine. Despite this, both new cases and new deaths spike at the beginning of 2022 and then begin to decrease.

```r
# Visualize total number of new cases and new deaths in Massachusetts
state <- "Massachusetts"
US_by_state %>%
  filter(Province_State == state) %>%
  filter(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")) +
  scale_y_log10() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90))+
  labs(title = "COVID19 in Massachusetts", y = NULL)
```

```
## Warning in self$trans$transform(x): NaNs produced

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

## Warning in self$trans$transform(x): NaNs produced

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

## Warning in self$trans$transform(x): NaNs produced

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

## Warning in self$trans$transform(x): NaNs produced

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

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

## Warning: Removed 2 rows containing missing values (`geom_point()`).
```



This visualization shows the number of new cases (red) and new deaths (blue) across time in the state of Massachusetts. The number of new cases and new deaths varies significantly in 2020, with a significant peak in the earlier months and a significant decrease halfway through the year. The trends here appear to be much more volatile. One other significant trend here is a massive decrease in both new cases and new deaths halfway through 2021.

Further Analysis

For further analysis, we can compare cases and deaths across each state. This will allow us to identify which states experienced the most or fewest number of cases and deaths.

```r
# Compare cases and deaths across each state
US_state_totals <- US_by_state %>%
  group_by(Province_State) %>%
  summarize(Deaths = max(Deaths), Cases = max(Cases),
            Population = max(Population),
            Cases_per_thou = 1000 * Cases / Population,
            Deaths_per_thou = 1000 * Deaths / Population) %>%
  filter(Cases > 0, Population > 0)
```

```r
# States with smallest number of deaths per thousand
US_state_totals %>%
  slice_min(Deaths_per_thou, n = 10) %>%
  select(Deaths_per_thou, Cases_per_thou, everything())
```

```
## # A tibble: 10 x 6
##    Deaths_per_thou Cases_per_thou Province_State           Deaths  Cases Popul~1
##              <dbl>          <dbl> <chr>                     <dbl>  <dbl>   <dbl>
##  1           0.611           150. American Samoa               34 8.32e3   55641
##  2           0.744           246. Northern Mariana Islands     41 1.36e4   55144
##  3           1.20            230. Virgin Islands              129 2.47e4  107268
##  4           1.27            267. Hawaii                     1805 3.78e5 1415872
##  5           1.44            242. Vermont                     901 1.51e5  623989
##  6           1.53            291. Puerto Rico                5750 1.09e6 3754939
##  7           1.64            338. Utah                       5270 1.08e6 3205958
##  8           1.99            412. Alaska                     1473 3.05e5  740995
##  9           2.02            251. District of Columbia       1425 1.77e5  705749
## 10           2.04            252. Washington                15510 1.92e6 7614893
## # ... with abbreviated variable name 1: Population
```

By using slice_min, we can gain insight into which states have fewest deaths per thousand (population). Here, we can see that the province of American Samoa has only 34 total deaths over the years, resulting in ~0.6 deaths per thousand people.

```
# States with largest number of deaths per thousand
US_state_totals %>%
  slice_max(Deaths_per_thou, n = 10) %>%
  select(Deaths_per_thou, Cases_per_thou, everything())
```

```
## # A tibble: 10 x 6
##    Deaths_per_thou Cases_per_thou Province_State Deaths   Cases Population
##              <dbl>          <dbl> <chr>           <dbl>   <dbl>      <dbl>
##  1            4.52           330. Arizona         32936 2404386    7278717
##  2            4.49           323. Oklahoma        17767 1278295    3956971
##  3            4.45           330. Mississippi     13257  981020    2976149
##  4            4.41           355. West Virginia    7904  637100    1792147
##  5            4.29           318. New Mexico       9001  666445    2096829
##  6            4.28           331. Arkansas        12925  999652    3017804
##  7            4.26           332. Alabama         20892 1627670    4903185
##  8            4.25           364. Tennessee       29056 2487408    6829174
##  9            4.19           304. Michigan        41809 3036304    9986857
## 10            4.04           340. New Jersey      35866 3021244    8882190
```

Conversely, we can use slice_max to determine which states have the highest number of deaths per thousand people. Here, we can see that Arizona has 32936 deaths, resulting in ~4.5 deaths per thousand people.

```
# States with largest number of cases per thousand
US_state_totals %>%
  slice_max(Cases_per_thou, n = 10) %>%
  select(Deaths_per_thou, Cases_per_thou, everything())
```

```
## # A tibble: 10 x 6
##    Deaths_per_thou Cases_per_thou Province_State Deaths   Cases Population
##              <dbl>          <dbl> <chr>           <dbl>   <dbl>      <dbl>
##  1            3.63           432. Rhode Island     3841  457162    1059361
##  2            1.99           412. Alaska           1473  305060     740995
##  3            4.02           380. Kentucky        17939 1698146    4467673
##  4             3.23           373. North Dakota     2458  284627     762062
##  5             2.53           371. Guam              416   60903     164229
##  6             4.25           364. Tennessee       29056 2487408    6829174
##  7             4.41           355. West Virginia    7904  637100    1792147
##  8             3.75           353. South Carolina  19330 1818546    5148714
##  9             3.99           348. Florida         85710 7483857   21477737
## 10             3.95           347. New York        76775 6746006   19453561
```

A quick alteration of our code can yield a view of the states with the highest number of cases per thousand people. One interesting finding is that although Arizona has the highest number of deaths per thousand, it is not on the list of the top 10 states with the most cases per thousand. In fact, Rhode Island has the most cases per thousand (431), but only has ~3.6 deaths per thousand.

Modeling

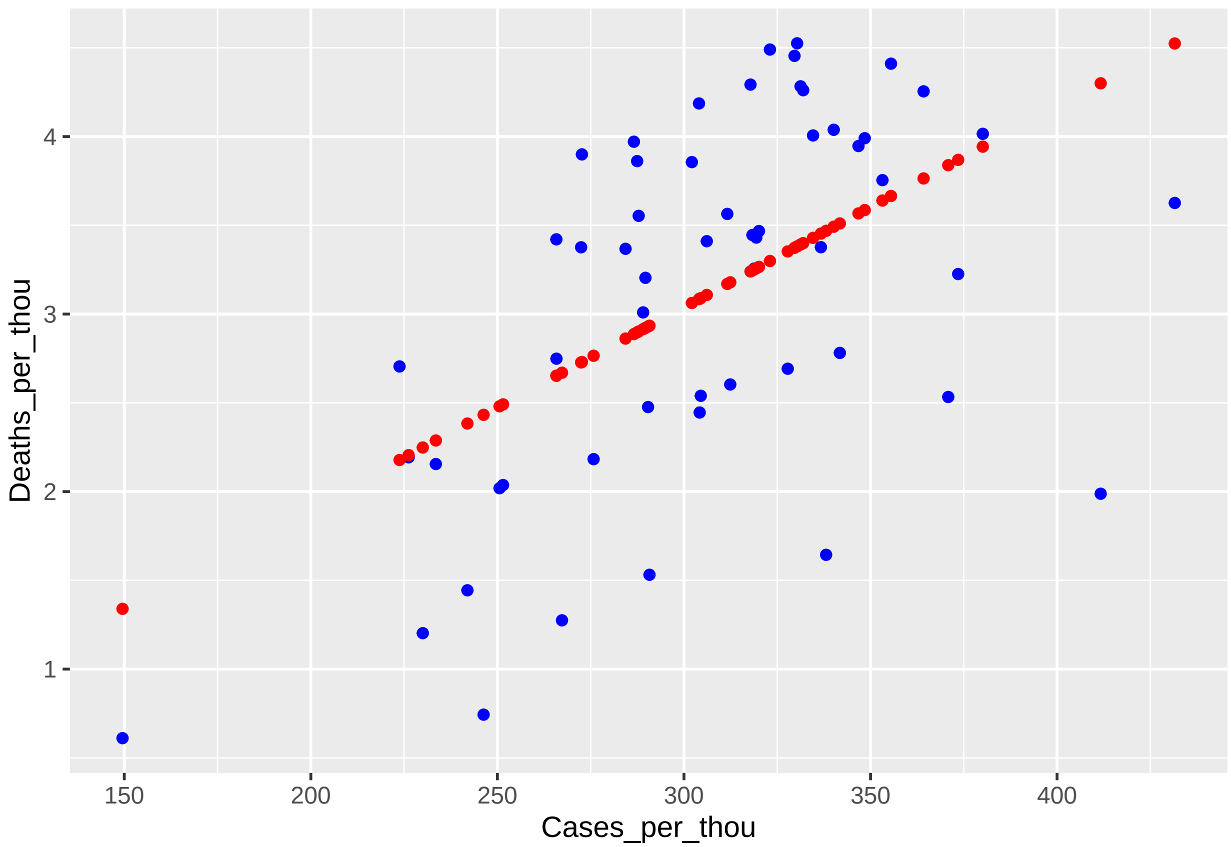
Our following model will attempt to model deaths per thousand as a function of cases per thousand.

```r
# Modeling Deaths_per_thou as a function of Cases_per_thou
mod <- lm(Deaths_per_thou ~ Cases_per_thou, data = US_state_totals)
summary(mod)
```

```
##
## Call:
## lm(formula = Deaths_per_thou ~ Cases_per_thou, data = US_state_totals)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3120 -0.5970  0.1441  0.6494  1.1911
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.348860   0.724793  -0.481    0.632
## Cases_per_thou  0.011292   0.002341   4.824 1.19e-05 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.8597 on 54 degrees of freedom
## Multiple R-squared:  0.3011, Adjusted R-squared:  0.2882
## F-statistic: 23.27 on 1 and 54 DF,  p-value: 1.192e-05
```

```r
# Create a new data set with predictions
US_state_totals_pred <- US_state_totals %>%
  mutate(pred = predict(mod))
```

```r
# Visualize the model
US_state_totals_pred %>% ggplot() +
  geom_point(aes(x = Cases_per_thou, y = Deaths_per_thou),
             color = "blue")+
  geom_point(aes(x = Cases_per_thou, y = pred), color = "red")
```



The model above shows deaths per thousand as a function of cases per thousand. The blue points represent actual data, while the red points represent our predicted values. The predicted values represent a linear line, suggesting a positive linear relationship between cases per thousand and deaths per thousand. The actual values closely adhere to the predicted values from 225 to 375 cases per thousand. However, at higher or lower values, the adherence decreases.

Potential Sources of Bias

1) COVID-19 Reporting Strategies One potential source of bias may be the COVID-19 reporting strategies across each state. I am unaware as to whether or not different states use different reporting and tracking methods. If the methods vary across each state, then the reported data may not accurately reflect the real values of COVID-19 cases and deaths.
2) A Gradual Lax in Reporting During the initial phases of the pandemic, reported cases and deaths increased significantly. However, we found that the number of reported cases and deaths plateaued at the end of 2020. I argue that a gradual lax in reporting over time may have contributed to this plateau in reporting.

Conclusion

Through this brief analysis, we found that both COVID-19 cases and deaths increased significantly through 2020 but began to plateau towards the end of that year. However, through the analysis of new cases and new deaths, we were able to more properly view the volatility of trends. Notably, we found that new cases and new deaths reached a maximum value during the beginning of 2022. Lastly, we modeled the number of COVID-19 deaths per thousand people by the number of cases per thousand. We found a linear relationship between the number of deaths and cases. We also found that our model better predicted actual values when the number of cases per thousand was in the median range.