John Hopkins COVID-19 Data Set Analysis

2025-07-26

R Packages Utilized in the Analysis

```
library(tidyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

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

```
library(ggplot2)

#Purpose: Efficiently reads and writes data, especially CSVs and text files.
library(readr)

#Provides consistent, simple functions for string (text) manipulation.
library(stringr)
#Purpose: Makes working with dates and times easier and more intuitive.
library(lubridate)
```

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

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

Downloading the Johns Hopkins COVID-19 Data

I will be downloading the Johns Hopkins data sets. First, I set a URL base, which makes up most of the URL. Then I will create a vector of the CSV file names. Next, I will concatenate the base URL with the file names. When we iterate through the concatenated URL, we can assign it to a specific variable. Therefore, we obtained the global cases, global deaths, United States Cases, and United States Deaths into their data frame.

```r
url_base <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_c

# 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"
)

# Construct raw file URLs
urls <- str_c(url_base, file_names)

# Read the data
global_cases <- read_csv(urls[1])
```

```
## Rows: 289 Columns: 1147
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1145): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

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

```
## Rows: 289 Columns: 1147
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1145): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

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

```
## Rows: 3342 Columns: 1154
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1148): UID, code3, FIPS, Lat, Long_, 1/22/20, 1/23/20, 1/24/20, 1/25/20...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
us_deaths <- read_csv(urls[4])
```

```
## Rows: 3342 Columns: 1155
## -- Column specification --------------------------------------------------------
```

```
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1149): UID, code3, FIPS, Lat, Long_, Population, 1/22/20, 1/23/20, 1/24...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(global_cases)
```

```
## # A tibble: 6 x 1,147
##   `Province/State` `Country/Region`   Lat  Long `1/22/20` `1/23/20` `1/24/20`
##   <chr>            <chr>            <dbl> <dbl>     <dbl>     <dbl>     <dbl>
## 1 <NA>             Afghanistan       33.9 67.7          0         0         0
## 2 <NA>             Albania           41.2 20.2          0         0         0
## 3 <NA>             Algeria           28.0  1.66         0         0         0
## 4 <NA>             Andorra           42.5  1.52         0         0         0
## 5 <NA>             Angola           -11.2 17.9          0         0         0
## 6 <NA>             Antarctica       -71.9 23.3          0         0         0
## # i 1,140 more variables: `1/25/20` <dbl>, `1/26/20` <dbl>, `1/27/20` <dbl>,
## #   `1/28/20` <dbl>, `1/29/20` <dbl>, `1/30/20` <dbl>, `1/31/20` <dbl>,
## #   `2/1/20` <dbl>, `2/2/20` <dbl>, `2/3/20` <dbl>, `2/4/20` <dbl>,
## #   `2/5/20` <dbl>, `2/6/20` <dbl>, `2/7/20` <dbl>, `2/8/20` <dbl>,
## #   `2/9/20` <dbl>, `2/10/20` <dbl>, `2/11/20` <dbl>, `2/12/20` <dbl>,
## #   `2/13/20` <dbl>, `2/14/20` <dbl>, `2/15/20` <dbl>, `2/16/20` <dbl>,
## #   `2/17/20` <dbl>, `2/18/20` <dbl>, `2/19/20` <dbl>, `2/20/20` <dbl>, ...
```

```
head(global_deaths)
```

```
## # A tibble: 6 x 1,147
##   `Province/State` `Country/Region`   Lat  Long `1/22/20` `1/23/20` `1/24/20`
##   <chr>            <chr>            <dbl> <dbl>     <dbl>     <dbl>     <dbl>
## 1 <NA>             Afghanistan       33.9 67.7          0         0         0
## 2 <NA>             Albania           41.2 20.2          0         0         0
## 3 <NA>             Algeria           28.0  1.66         0         0         0
## 4 <NA>             Andorra           42.5  1.52         0         0         0
## 5 <NA>             Angola           -11.2 17.9          0         0         0
## 6 <NA>             Antarctica       -71.9 23.3          0         0         0
## # i 1,140 more variables: `1/25/20` <dbl>, `1/26/20` <dbl>, `1/27/20` <dbl>,
## #   `1/28/20` <dbl>, `1/29/20` <dbl>, `1/30/20` <dbl>, `1/31/20` <dbl>,
## #   `2/1/20` <dbl>, `2/2/20` <dbl>, `2/3/20` <dbl>, `2/4/20` <dbl>,
## #   `2/5/20` <dbl>, `2/6/20` <dbl>, `2/7/20` <dbl>, `2/8/20` <dbl>,
## #   `2/9/20` <dbl>, `2/10/20` <dbl>, `2/11/20` <dbl>, `2/12/20` <dbl>,
## #   `2/13/20` <dbl>, `2/14/20` <dbl>, `2/15/20` <dbl>, `2/16/20` <dbl>,
## #   `2/17/20` <dbl>, `2/18/20` <dbl>, `2/19/20` <dbl>, `2/20/20` <dbl>, ...
```

```
head(us_cases)
```

```
## # A tibble: 6 x 1,154
##        UID iso2  iso3  code3  FIPS Admin2  Province_State Country_Region   Lat
##      <dbl> <chr> <chr> <dbl> <dbl> <chr>   <chr>          <chr>          <dbl>
## 1 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
## 2 84001003 US    USA     840  1003 Baldwin Alabama        US              30.7
```

```
## 3 84001005 US    USA     840  1005 Barbour Alabama        US              31.9
## 4 84001007 US    USA     840  1007 Bibb    Alabama        US              33.0
## 5 84001009 US    USA     840  1009 Blount  Alabama        US              34.0
## 6 84001011 US    USA     840  1011 Bullock Alabama        US              32.1
## # i 1,145 more variables: Long_ <dbl>, Combined_Key <chr>, `1/22/20` <dbl>,
## #   `1/23/20` <dbl>, `1/24/20` <dbl>, `1/25/20` <dbl>, `1/26/20` <dbl>,
## #   `1/27/20` <dbl>, `1/28/20` <dbl>, `1/29/20` <dbl>, `1/30/20` <dbl>,
## #   `1/31/20` <dbl>, `2/1/20` <dbl>, `2/2/20` <dbl>, `2/3/20` <dbl>,
## #   `2/4/20` <dbl>, `2/5/20` <dbl>, `2/6/20` <dbl>, `2/7/20` <dbl>,
## #   `2/8/20` <dbl>, `2/9/20` <dbl>, `2/10/20` <dbl>, `2/11/20` <dbl>,
## #   `2/12/20` <dbl>, `2/13/20` <dbl>, `2/14/20` <dbl>, `2/15/20` <dbl>, ...
```

```
head(us_deaths)
```

```
## # A tibble: 6 x 1,155
##        UID iso2  iso3  code3  FIPS Admin2  Province_State Country_Region   Lat
##      <dbl> <chr> <chr> <dbl> <dbl> <chr>   <chr>          <chr>          <dbl>
## 1 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
## 2 84001003 US    USA     840  1003 Baldwin Alabama        US              30.7
## 3 84001005 US    USA     840  1005 Barbour Alabama        US              31.9
## 4 84001007 US    USA     840  1007 Bibb    Alabama        US              33.0
## 5 84001009 US    USA     840  1009 Blount  Alabama        US              34.0
## 6 84001011 US    USA     840  1011 Bullock Alabama        US              32.1
## # i 1,146 more variables: Long_ <dbl>, Combined_Key <chr>, Population <dbl>,
## #   `1/22/20` <dbl>, `1/23/20` <dbl>, `1/24/20` <dbl>, `1/25/20` <dbl>,
## #   `1/26/20` <dbl>, `1/27/20` <dbl>, `1/28/20` <dbl>, `1/29/20` <dbl>,
## #   `1/30/20` <dbl>, `1/31/20` <dbl>, `2/1/20` <dbl>, `2/2/20` <dbl>,
## #   `2/3/20` <dbl>, `2/4/20` <dbl>, `2/5/20` <dbl>, `2/6/20` <dbl>,
## #   `2/7/20` <dbl>, `2/8/20` <dbl>, `2/9/20` <dbl>, `2/10/20` <dbl>,
## #   `2/11/20` <dbl>, `2/12/20` <dbl>, `2/13/20` <dbl>, `2/14/20` <dbl>, ...
```

Transforming Data into Tidy Format

Unfortunately, our data is not in a tidy format because the dates are listed within columns. We will need to put all the data sets into tidy format, and we will do that through the `pivot_longer()` function. This function works by combining the columns into one column; the names_to portion is the new name we assign to the columns being pivoted. The Values_to portion of the function represents the values that were in the previous columns that were pivoted.

```
global_cases_tidy <- global_cases %>%
  pivot_longer(
    cols = matches("^\\d{1,2}/\\d{1,2}/\\d{2}$"),  # columns that are dates like "1/22/20" or "12/1/22"
    names_to = "date",
    values_to = "cases"
  ) %>%
  mutate(date = lubridate::mdy(date))  # convert date strings to Date objects

# You can do the same for other data sets
global_deaths_tidy <- global_deaths %>%
  pivot_longer(cols = matches("^\\d{1,2}/\\d{1,2}/\\d{2}$"), names_to = "date", values_to = "deaths") %
  mutate(date = lubridate::mdy(date))
```

```
us_cases_tidy <- us_cases %>%
  pivot_longer(cols = matches("^\\d{1,2}/\\d{1,2}/\\d{2}$"), names_to = "date", values_to = "cases") %>%
  mutate(date = lubridate::mdy(date))

us_deaths_tidy <- us_deaths %>%
  pivot_longer(cols = matches("^\\d{1,2}/\\d{1,2}/\\d{2}$"), names_to = "date", values_to = "deaths") %>%
  mutate(date = lubridate::mdy(date))
```

Selecting Relevant Columns for Analysis

We won't be using every column in our analysis, so we'll clean the data set by removing unnecessary columns.

```
selected_data_global_cases <- global_cases_tidy %>%
  select(-Long, -Lat)
head(selected_data_global_cases)
```

```
## # A tibble: 6 x 4
##   `Province/State` `Country/Region` date       cases
##   <chr>            <chr>            <date>     <dbl>
## 1 <NA>             Afghanistan      2020-01-22     0
## 2 <NA>             Afghanistan      2020-01-23     0
## 3 <NA>             Afghanistan      2020-01-24     0
## 4 <NA>             Afghanistan      2020-01-25     0
## 5 <NA>             Afghanistan      2020-01-26     0
## 6 <NA>             Afghanistan      2020-01-27     0
```

```
selected_data_global_deaths <- global_deaths_tidy %>%
  select(-Long, -Lat)
head(selected_data_global_deaths)
```

```
## # A tibble: 6 x 4
##   `Province/State` `Country/Region` date       deaths
##   <chr>            <chr>            <date>      <dbl>
## 1 <NA>             Afghanistan      2020-01-22      0
## 2 <NA>             Afghanistan      2020-01-23      0
## 3 <NA>             Afghanistan      2020-01-24      0
## 4 <NA>             Afghanistan      2020-01-25      0
## 5 <NA>             Afghanistan      2020-01-26      0
## 6 <NA>             Afghanistan      2020-01-27      0
```

```
selected_data_us_cases <- us_cases_tidy%>%
  select(-UID, -iso2, -iso3, -code3, -Lat, -Long_, -FIPS)
head(selected_data_us_cases)
```

```
## # A tibble: 6 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
```

```
selected_data_us_deaths <- us_deaths_tidy%>%
  select(-UID, -iso2, -iso3, -code3, -Lat, -Long_, -FIPS)
head(selected_data_us_deaths)
```

```
## # A tibble: 6 x 7
##   Admin2 Province_State Country_Region Combined_Key Population date       deaths
##   <chr>  <chr>          <chr>          <chr>              <dbl> <date>      <dbl>
## 1 Autau~ Alabama        US             Autauga, Al~       55869 2020-01-22      0
## 2 Autau~ Alabama        US             Autauga, Al~       55869 2020-01-23      0
## 3 Autau~ Alabama        US             Autauga, Al~       55869 2020-01-24      0
## 4 Autau~ Alabama        US             Autauga, Al~       55869 2020-01-25      0
## 5 Autau~ Alabama        US             Autauga, Al~       55869 2020-01-26      0
## 6 Autau~ Alabama        US             Autauga, Al~       55869 2020-01-27      0
```

Joining Data Sets

For both the global and the United States data sets, we will want to combine them so that we have one data set that includes both cases and deaths. We want to do this so we can perform specific calculations and analyses.

```
us_combined <- full_join(selected_data_us_cases, selected_data_us_deaths, by = c("Combined_Key", "date"
us_combined_clean <- us_combined %>%
  select(Combined_Key, Province_State, Country_Region, date, cases, deaths, Population)

head(us_combined_clean)
```

```
## # A tibble: 6 x 7
##   Combined_Key  Province_State Country_Region date       cases deaths Population
##   <chr>         <chr>          <chr>          <date>     <dbl>  <dbl>      <dbl>
## 1 Autauga, Ala~ Alabama        US             2020-01-22     0      0      55869
## 2 Autauga, Ala~ Alabama        US             2020-01-23     0      0      55869
## 3 Autauga, Ala~ Alabama        US             2020-01-24     0      0      55869
## 4 Autauga, Ala~ Alabama        US             2020-01-25     0      0      55869
## 5 Autauga, Ala~ Alabama        US             2020-01-26     0      0      55869
## 6 Autauga, Ala~ Alabama        US             2020-01-27     0      0      55869
```

```
global_combine <- full_join(selected_data_global_cases, selected_data_global_deaths, by = c("Province/S

head(global_combine)
```

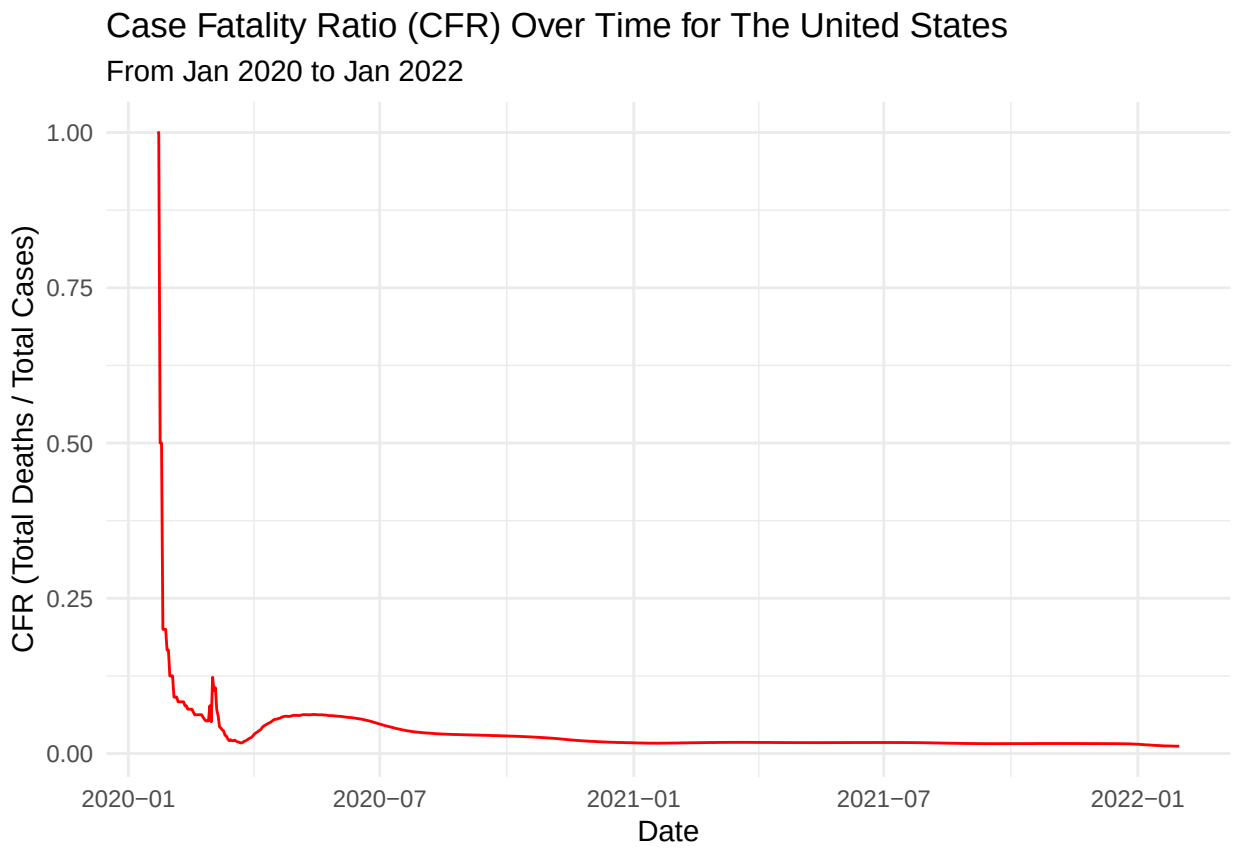
```
## # A tibble: 6 x 5
##   `Province/State` `Country/Region` date       cases deaths
##   <chr>            <chr>            <date>     <dbl>  <dbl>
## 1 <NA>             Afghanistan      2020-01-22     0      0
## 2 <NA>             Afghanistan      2020-01-23     0      0
## 3 <NA>             Afghanistan      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
```

Case Fatality Ratio (CFR) Over Time for the United States

The Case Fatality Ratio (CFR) indicates how deadly a disease is among diagnosed cases. It is calculated as: CFR = (Number of deaths from the disease) / (Number of confirmed cases of the disease). This visualization shows how the CFR for COVID-19 changed over time in the United States between January 2020 and January 2022. We chose this time frame because, after this period, the CFR begins to stabilize and approaches what appears to be an asymptote, suggesting that the fatality rate levels off as testing, treatment, and data collection improve.

From this visualization, we can see that the United States had a very high CFR at the start of the pandemic, but after July 2020, the CFR dropped and remained relatively stable at a very low CFR.

```
us_combined_clean %>%
  filter(date >= as.Date("2020-01-01") & date <= as.Date("2022-01-31")) %>%
  group_by(date) %>%
  summarize(
    total_cases = sum(cases, na.rm = TRUE),
    total_deaths = sum(deaths, na.rm = TRUE),
    .groups = "drop"
  ) %>%
  mutate(cfr = total_deaths / total_cases) %>%
  ggplot(aes(x = date, y = cfr)) +
  geom_line(color = "red") +
  labs(
    title = "Case Fatality Ratio (CFR) Over Time for The United States",
    subtitle = "From Jan 2020 to Jan 2022",
    y = "CFR (Total Deaths / Total Cases)",
    x = "Date"
  ) +
  theme_minimal()
```



Daily New Global Cases and Deaths

With these visuals, we examine daily new cases and deaths at a global level. These visualizations will provide insights into how governments worldwide are managing the virus. From this visualization, a person can track the peaks and valleys in the number of cases and deaths from the disease, which should allow an analyst to see overall trends.

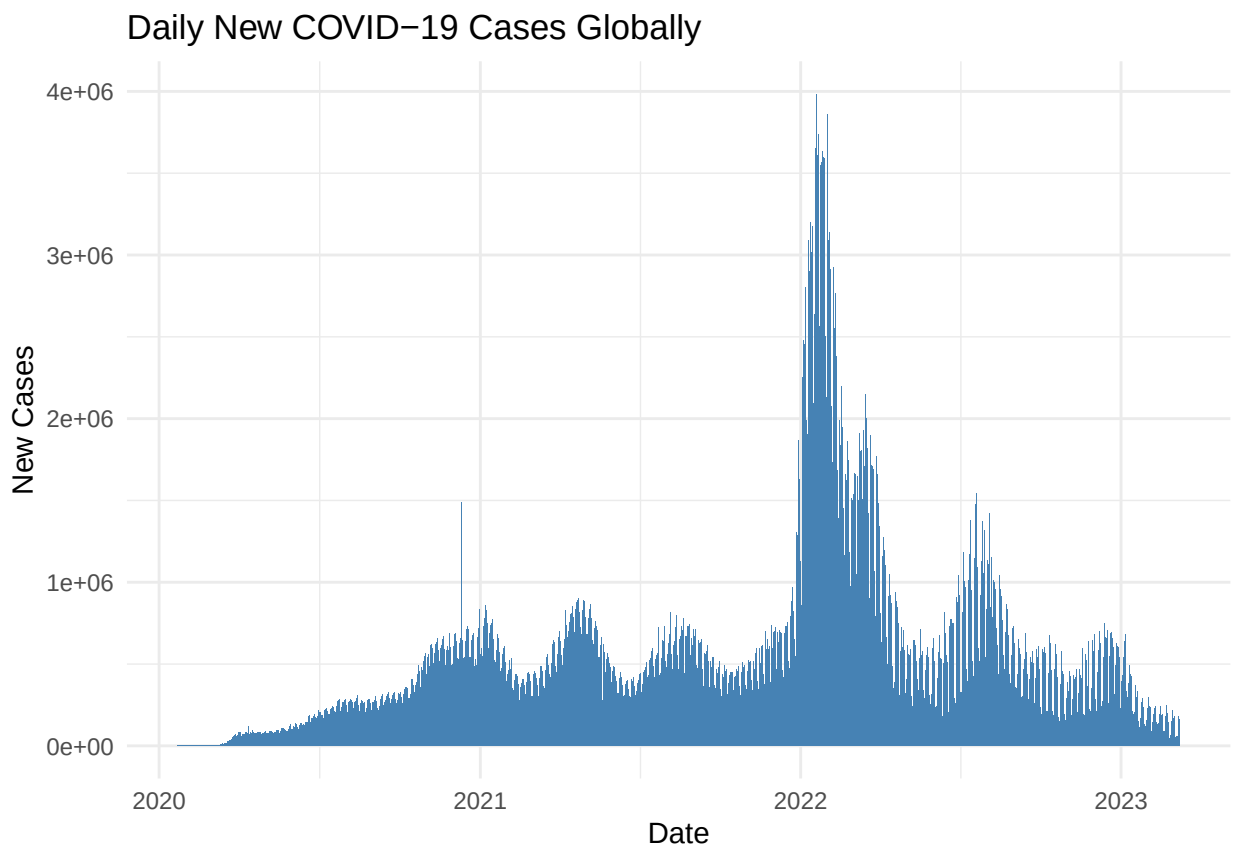
What we can see is that new cases experienced a stark rise in 2022 but then fell off in 2023. While deaths spiked in 2021, they remained at a high level until about halfway through 2022 and then dropped to their lowest point, and have not spiked again.

```
global_combine %>%
 #Ensures that the data is sorted by country and then by date.
 #Sorting is essential before using lag() to correctly calculate daily changes.
 arrange(`Country/Region`, date) %>%

 #Groups data within each country.
 #Calculates daily new cases by subtracting the previous day's cases (cumulative) using lag(cases).
 #Adds a new column called new_cases.
 group_by(`Country/Region`) %>%
 mutate(new_cases = cases - lag(cases)) %>%

 #grouping by date (across all countries).
 #For each date, we sum the new cases from all countries to calculate the total number of new COVID-19
 group_by(date) %>%
```

```r
  #na.rm = TRUE removes any missing values that came from lag() because it was the first documentation
  #.groups = "drop" ensures the result is ungrouped afterward.
  summarise(daily_new_cases = sum(new_cases, na.rm = TRUE), .groups = "drop") %>%
  #Creates a bar chart (geom_col()) where:
  #x = date
  #y = daily_new_cases (total global new cases for that date)
  ggplot(aes(x = date, y = daily_new_cases)) +
  geom_col(fill = "steelblue") +
  labs(
    title = "Daily New COVID-19 Cases Globally",
    y = "New Cases",
    x = "Date"
  ) +
  theme_minimal()
```

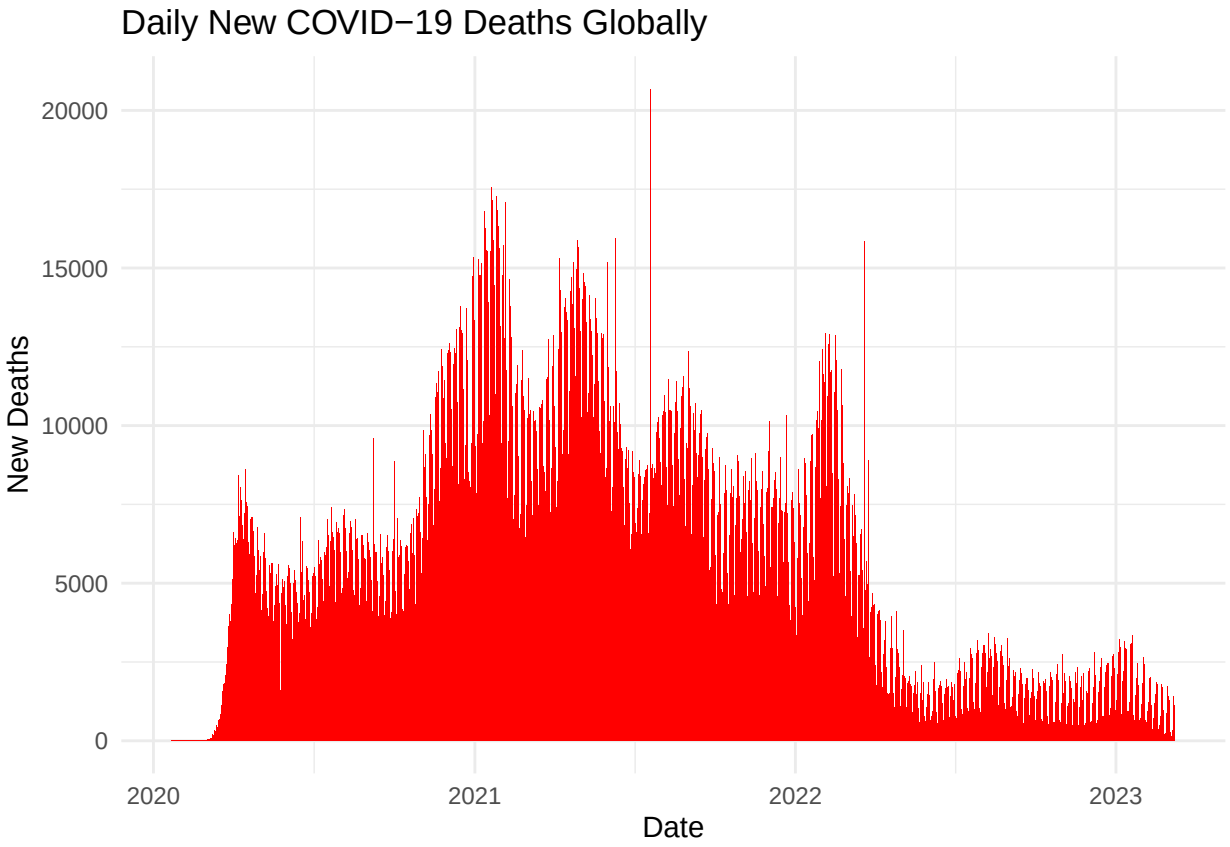



```r
global_combine %>%
  #Ensures that the data is sorted by country and then by date.
  #Sorting is essential before using lag() to correctly calculate daily changes.
  arrange(`Country/Region`, date) %>%

  #Groups data within each country.
  #Calculates daily new deaths by subtracting the previous day's deaths (cumulative) using lag(deaths).
  #Adds a new column called new_deaths.
  group_by(`Country/Region`) %>%
  mutate(new_deaths = deaths - lag(deaths)) %>%
```

```r
#grouping by date (across all countries).
#For each date, we sum the new deaths from all countries to calculate the total number of new COVID-1
group_by(date) %>%
#na.rm = TRUE removes any missing values that came from lag() because it was the first documentation
#.groups = "drop" ensures the result is ungrouped afterward.
summarize(daily_new_deaths = sum(new_deaths, na.rm = TRUE), .groups = "drop") %>%

#Creates a bar chart (geom_col()) where:
#x = date
#y = daily_new_deaths (total global new deaths for that date)
ggplot(aes(x = date, y = daily_new_deaths)) +
geom_col(fill = "red") +
labs(
  title = "Daily New COVID-19 Deaths Globally",
  x = "Date",
  y = "New Deaths"
) +
theme_minimal()
```



Cleaning the Data for the Linear Model

```r
summary_by_state <- us_combined_clean %>%
  #Keeps only rows from the most recent date and drops any counties with missing population data.
```

```r
  filter(date == max(date), !is.na(Population)) %>%
  group_by(Province_State) %>%
  summarize(
    total_case = sum(cases, na.rm = TRUE),
    total_deaths = sum(deaths, na.rm = TRUE),
    total_population = sum(Population, na.rm = TRUE),
    cases_per_100k = (total_case / total_population) * 100000,
    deaths_per_100k = (total_deaths / total_population) * 100000,
    .groups = "drop"
  )

#Ensures that only rows with valid numeric values are included in the model.
#Removes any state that has:
#Missing (NA) values
#Infinite (Inf or -Inf) values
summary_by_state_clean <- summary_by_state %>%
  filter(
    !is.na(cases_per_100k),
    !is.na(deaths_per_100k),
    is.finite(cases_per_100k),
    is.finite(deaths_per_100k)
  )
```

How do COVID case rates per 100k predict death rates per 100k across states?

This linear model attempts to predict the COVID-19 death rate per 100,000 using the case rate per 100,000 as the predictor.

```r
#Fits a linear regression model where:
#Response (Y): deaths_per_100k
#Predictor (X): cases_per_100k
model <- lm(deaths_per_100k ~ cases_per_100k, data = summary_by_state_clean)
summary(model)
```

```
##
## Call:
## lm(formula = deaths_per_100k ~ cases_per_100k, data = summary_by_state_clean)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -233.52  -59.78   14.91   65.35  120.86
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -36.16655   72.48021  -0.499     0.62
## cases_per_100k   0.01133    0.00232   4.881 9.76e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 86.15 on 54 degrees of freedom
## Multiple R-squared:  0.3061, Adjusted R-squared:  0.2933
## F-statistic: 23.82 on 1 and 54 DF,  p-value: 9.763e-06
```

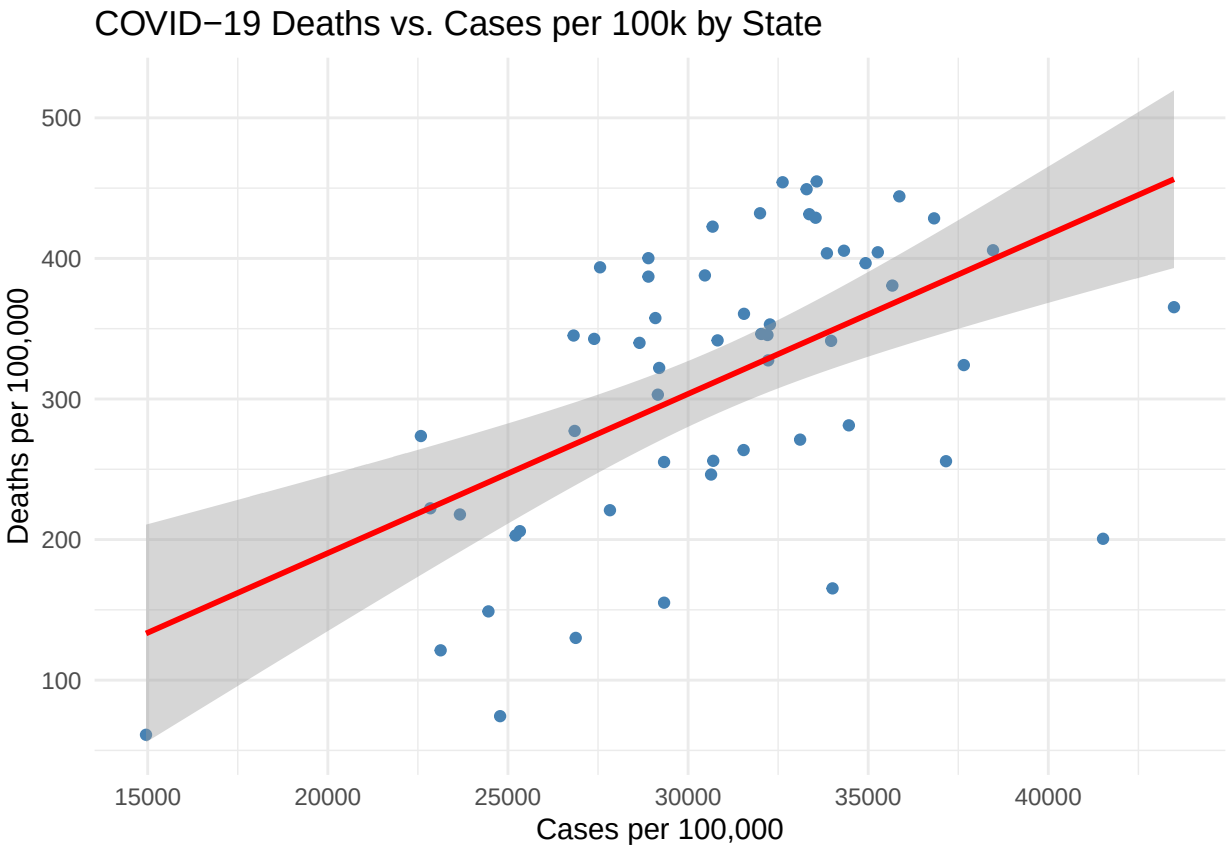
Based on the coefficient predicted, for every one additional case per 100,000 deaths, the rate increased by 0.01133. The coefficient is statistically significant due to the p-value < 0.05, which means that cases per 100,000 are a meaningful predictor of deaths per 100,000. The residual standard error is 86.15, so on average, our predictions are off by 86 deaths per 100k. The Multiple R-squared value indicates the percentage of variation in our dependent variable that the independent variable explains. In our case, cases per 100,000 explains 30.61% of the variation in deaths per 100,000. This means that Cases per 100,000 help explain some of the variation within deaths per 100,000, but not as much as we would like. Ultimately, our model isn't fitting the data very well (we observed this when examining the residual standard error as well). This means there is likely substantial unexplained variation in the model, likely due to other factors such as access to healthcare, financial status, etc.

Scatter Plot with Linear Regression: Cases per 100k as a Predictor of Deaths per 100k

This scatter plot displays the linear regression line with a confidence interval band around the predicted mean. While there is a general upward trend, many data points are widely scattered, indicating that case per 100,000 alone does not strongly predict deaths per 100,000. This suggests that other factors are at play. Some factors that could meaningfully affect deaths per 100,000 are things such as demographics, healthcare access, and public health policies. These factors likely play a significant role in explaining differences in death rates across states.

```
#Scatterplot of each state:
#X-axis: cases_per_100k
#Y-axis: deaths_per_100k
ggplot(summary_by_state_clean, aes(x = cases_per_100k, y = deaths_per_100k)) +
  geom_point(color = "steelblue") +
  #Adds a red regression line (geom_smooth(method = "lm"))
  #Shows the overall trend from the linear model
  #Includes a shaded area for standard error (confidence band) from the se = TRUE
  geom_smooth(method = "lm", se = TRUE, color = "red") +
  labs(
    title = "COVID-19 Deaths vs. Cases per 100k by State",
    x = "Cases per 100,000",
    y = "Deaths per 100,000"
  ) +
  theme_minimal()
```

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



Potential Biases in the Data

One potential bias in the global data set is that we are relying on governments worldwide to report their death and case rates accurately. For various geopolitical reasons, countries may not want to disclose this information. For instance, one reason they may not want to present this information to others is that they don't want their potential enemies to know they are being decimated by a virus, which could invite invaders while they are essentially in a weakened state. Another instance would be that they don't want to let people know cases are worse than what they are reporting, because then their allies will want to slow trade to avoid the potential spread of the virus into their country. These are perfectly plausible reasons for under reporting, which could affect the results we receive in our data set.

As for the United States data set, while there is less concern about receiving false information due to large-scale government corruption, it's not providing nearly enough factors to give an accurate picture of the issue. The United States is not a homogeneous population, unlike some countries. We have extremely varied populations in terms of race, wealth distributions, and age, and all of those factors could influence infection and death rates in counties and states. For instance, in many rural counties in America, people lack access to a nearby hospital because the nearest one is more than 3 hours away. If a person dies in a rural county, is it really because COVID-19 is extremely deadly, or is it because nobody in this area could receive treatment for their more dangerous symptoms? That is something that we can't answer from this data set alone, and it shows how this bias could skew our results.