Reproducible Report on COVID19 Data

Christopher Chery

2025-06-19

Project Objective

Utilizing Johns Hopkins University datasets, I will analyze COVID-19 pandemic data to address the following questions: which US state's population was most affected by the virus and how did the United States' mortality rate compare globally? Additionally, I will employ an ARIMA model to forecast US COVID-19 deaths for the first quarter of 2023.

Data Overview

First, I will import the necessary libraries and import the COVID19 and population data from the five JHU csv files.

```r
library("tidyverse")
library("dplyr")
library("lubridate")
library("forecast")
library("tseries")
```

```
## 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.
## 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.
## 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.
## 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.
## 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.

## # A tibble: 3,342 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
##  7 84001013 US    USA     840  1013 Butler   Alabama        US              31.8
##  8 84001015 US    USA     840  1015 Calhoun  Alabama        US              33.8
##  9 84001017 US    USA     840  1017 Chambers Alabama        US              32.9
## 10 84001019 US    USA     840  1019 Cherokee Alabama        US              34.2
## # i 3,332 more rows
## # 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>, ...

## # A tibble: 289 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
##  7 <NA>              Antigua and Bar~  17.1 -61.8          0         0         0
##  8 <NA>              Argentina        -38.4 -63.6          0         0         0
##  9 <NA>              Armenia           40.1  45.0          0         0         0
## 10 Australian Capit~ Australia        -35.5 149.           0         0         0
## # i 279 more rows
```

```
## # 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>, ...

## # A tibble: 3,342 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
##  7 84001013 US    USA     840  1013 Butler   Alabama        US              31.8
##  8 84001015 US    USA     840  1015 Calhoun  Alabama        US              33.8
##  9 84001017 US    USA     840  1017 Chambers Alabama        US              32.9
## 10 84001019 US    USA     840  1019 Cherokee Alabama        US              34.2
## # i 3,332 more rows
## # 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>, ...

## # A tibble: 289 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
##  7 <NA>              Antigua and Bar~  17.1 -61.8          0         0         0
##  8 <NA>              Argentina        -38.4 -63.6          0         0         0
##  9 <NA>              Armenia           40.1  45.0          0         0         0
## 10 Australian Capit~ Australia        -35.5 149.           0         0         0
## # i 279 more rows
## # 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>, ...

## # A tibble: 4,321 x 12
##      UID iso2  iso3  code3 FIPS  Admin2 Province_State Country_Region        Lat
##    <dbl> <chr> <chr> <dbl> <chr> <chr>  <chr>          <chr>               <dbl>
##  1     4 AF    AFG       4 <NA>  <NA>   <NA>           Afghanistan          33.9
```

```
## 2      8 AL    ALB       8 <NA>  <NA>   <NA>          Albania             41.2
## 3     10 AQ    ATA      10 <NA>  <NA>   <NA>          Antarctica         -71.9
## 4     12 DZ    DZA      12 <NA>  <NA>   <NA>          Algeria             28.0
## 5     20 AD    AND      20 <NA>  <NA>   <NA>          Andorra             42.5
## 6     24 AO    AGO      24 <NA>  <NA>   <NA>          Angola             -11.2
## 7     28 AG    ATG      28 <NA>  <NA>   <NA>          Antigua and Barbuda  17.1
## 8     32 AR    ARG      32 <NA>  <NA>   <NA>          Argentina          -38.4
## 9     51 AM    ARM      51 <NA>  <NA>   <NA>          Armenia             40.1
## 10    40 AT    AUT      40 <NA>  <NA>   <NA>          Austria             47.5
## # i 4,311 more rows
## # i 3 more variables: Long_ <dbl>, Combined_Key <chr>, Population <dbl>
```

Tidy and Transfrom Data

```r
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))

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

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

```r
global
```

```
## # A tibble: 306,827 x 5
##    Province_State Country_Region date       cases deaths
##    <chr>          <chr>          <date>     <dbl>  <dbl>
##  1 <NA>           Afghanistan    2020-02-24     5      0
##  2 <NA>           Afghanistan    2020-02-25     5      0
##  3 <NA>           Afghanistan    2020-02-26     5      0
##  4 <NA>           Afghanistan    2020-02-27     5      0
##  5 <NA>           Afghanistan    2020-02-28     5      0
```

```
##   6 <NA>           Afghanistan    2020-02-29     5      0
##   7 <NA>           Afghanistan    2020-03-01     5      0
##   8 <NA>           Afghanistan    2020-03-02     5      0
##   9 <NA>           Afghanistan    2020-03-03     5      0
##  10 <NA>           Afghanistan    2020-03-04     5      0
## # i 306,817 more rows
```

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

us <- us_cases %>%
  full_join(us_deaths) %>%
  filter(cases > 0) %>%
  rename(County = "Admin2")

us
```

```
## # A tibble: 3,474,292 x 8
##    County Province_State Country_Region Combined_Key date       cases Population
##    <chr>  <chr>          <chr>          <chr>        <date>     <dbl>      <dbl>
##  1 Autau~ Alabama        US             Autauga, Al~ 2020-03-24     1      55869
##  2 Autau~ Alabama        US             Autauga, Al~ 2020-03-25     5      55869
##  3 Autau~ Alabama        US             Autauga, Al~ 2020-03-26     6      55869
##  4 Autau~ Alabama        US             Autauga, Al~ 2020-03-27     6      55869
##  5 Autau~ Alabama        US             Autauga, Al~ 2020-03-28     6      55869
##  6 Autau~ Alabama        US             Autauga, Al~ 2020-03-29     6      55869
##  7 Autau~ Alabama        US             Autauga, Al~ 2020-03-30     8      55869
##  8 Autau~ Alabama        US             Autauga, Al~ 2020-03-31     8      55869
##  9 Autau~ Alabama        US             Autauga, Al~ 2020-04-01    10      55869
## 10 Autau~ Alabama        US             Autauga, Al~ 2020-04-02    12      55869
## # i 3,474,282 more rows
## # i 1 more variable: deaths <dbl>
```

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


global <- global %>%
```

```r
  unite("Combined_Key",
        c(Province_State, Country_Region),
        sep = ", ",
        na.rm = TRUE,
        remove = FALSE)
```

Now that the tidying and transformations are complete, these final data sets can be used for analysis.

```r
us
```

```
## # A tibble: 3,474,292 x 8
##    County Province_State Country_Region Combined_Key date       cases Population
##    <chr>  <chr>          <chr>          <chr>        <date>     <dbl>      <dbl>
##  1 Autau~ Alabama        US             Autauga, Al~ 2020-03-24     1      55869
##  2 Autau~ Alabama        US             Autauga, Al~ 2020-03-25     5      55869
##  3 Autau~ Alabama        US             Autauga, Al~ 2020-03-26     6      55869
##  4 Autau~ Alabama        US             Autauga, Al~ 2020-03-27     6      55869
##  5 Autau~ Alabama        US             Autauga, Al~ 2020-03-28     6      55869
##  6 Autau~ Alabama        US             Autauga, Al~ 2020-03-29     6      55869
##  7 Autau~ Alabama        US             Autauga, Al~ 2020-03-30     8      55869
##  8 Autau~ Alabama        US             Autauga, Al~ 2020-03-31     8      55869
##  9 Autau~ Alabama        US             Autauga, Al~ 2020-04-01    10      55869
## 10 Autau~ Alabama        US             Autauga, Al~ 2020-04-02    12      55869
## # i 3,474,282 more rows
## # i 1 more variable: deaths <dbl>
```

```r
summary(us)
```

```
##     County          Province_State     Country_Region     Combined_Key
##  Length:3474292     Length:3474292     Length:3474292     Length:3474292
##  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.   :      1   Min.   :       0   Min.   :    0.0
##  1st Qu.:2020-12-27   1st Qu.:    687   1st Qu.:   10953   1st Qu.:   10.0
##  Median :2021-09-20   Median :   2849   Median :   26248   Median :   47.0
##  Mean   :2021-09-19   Mean   :  15489   Mean   :  104502   Mean   :  205.1
##  3rd Qu.:2022-06-15   3rd Qu.:   9345   3rd Qu.:   68098   3rd Qu.:  137.0
##  Max.   :2023-03-09   Max.   :3710586   Max.   :10039107   Max.   :35545.0
```

```r
global
```

```
## # A tibble: 306,827 x 7
##    Combined_Key Province_State Country_Region date       cases deaths Population
##    <chr>        <chr>          <chr>          <date>     <dbl>  <dbl>      <dbl>
##  1 Afghanistan  <NA>           Afghanistan    2020-02-24     5      0   38928341
##  2 Afghanistan  <NA>           Afghanistan    2020-02-25     5      0   38928341
##  3 Afghanistan  <NA>           Afghanistan    2020-02-26     5      0   38928341
##  4 Afghanistan  <NA>           Afghanistan    2020-02-27     5      0   38928341
```

```
##   5 Afghanistan  <NA>          Afghanistan    2020-02-28     5       0    38928341
##   6 Afghanistan  <NA>          Afghanistan    2020-02-29     5       0    38928341
##   7 Afghanistan  <NA>          Afghanistan    2020-03-01     5       0    38928341
##   8 Afghanistan  <NA>          Afghanistan    2020-03-02     5       0    38928341
##   9 Afghanistan  <NA>          Afghanistan    2020-03-03     5       0    38928341
##  10 Afghanistan  <NA>          Afghanistan    2020-03-04     5       0    38928341
## # i 306,817 more rows
```

```
summary(global)
```

```
##  Combined_Key       Province_State     Country_Region          date
##  Length:306827      Length:306827      Length:306827      Min.   :2020-01-22
##  Class :character   Class :character   Class :character   1st Qu.:2020-12-12
##  Mode  :character   Mode  :character   Mode  :character   Median :2021-09-16
##                                                           Mean   :2021-09-11
##                                                           3rd Qu.:2022-06-15
##                                                           Max.   :2023-03-09
##
##      cases                deaths          Population
##  Min.   :        1   Min.   :      0   Min.   :6.700e+01
##  1st Qu.:     1316   1st Qu.:      7   1st Qu.:7.866e+05
##  Median :    20365   Median :    214   Median :6.948e+06
##  Mean   :  1032863   Mean   :  14405   Mean   :2.890e+07
##  3rd Qu.:   271281   3rd Qu.:   3665   3rd Qu.:2.914e+07
##  Max.   :103802702   Max.   :1123836   Max.   :1.380e+09
##                                        NA's   :6729
```

Exploratory Data Analysis

Objective #1

For my first objective of determining which US state was most affected by COVID-19, I will summarize cases, deaths, and population by each state and again by the total United States. I will also create variables for cases per million, deaths per million, and mortality rate.

```
state_pop <- us %>%
  distinct(Province_State, County, .keep_all = TRUE) %>%
  group_by(Province_State) %>%
  summarize(Population = sum(Population))

us_by_state <- us %>%
  group_by(Country_Region, Province_State, date) %>%
  summarize(cases = sum(cases),
            deaths = sum(deaths)) %>%
  ungroup() %>%

  left_join(state_pop, by = "Province_State") %>%
  filter(Population > 0) %>%
  filter(!is.na(Population)) %>%

  mutate(deaths_per_mill = deaths * 1000000 / Population,
         cases_per_mill = cases * 1000000 / Population,
```

```r
         mortality_rate = deaths/ cases) %>%
  select(Province_State, date, cases, cases_per_mill, deaths, deaths_per_mill, mortality_rate, Populati

us_states_ovr <- us_by_state %>%
  group_by(Province_State) %>%
  filter(date == max(date)) %>%
  ungroup() %>%
  select(Province_State, cases, cases_per_mill, deaths, deaths_per_mill, mortality_rate, Population)

us_by_state
```

```
## # A tibble: 61,039 x 8
##    Province_State date       cases cases_per_mill deaths deaths_per_mill
##    <chr>          <date>     <dbl>          <dbl>  <dbl>           <dbl>
##  1 Alabama        2020-03-11     3          0.612      0               0
##  2 Alabama        2020-03-12     4          0.816      0               0
##  3 Alabama        2020-03-13     8          1.63       0               0
##  4 Alabama        2020-03-14    15          3.06       0               0
##  5 Alabama        2020-03-15    28          5.71       0               0
##  6 Alabama        2020-03-16    36          7.34       0               0
##  7 Alabama        2020-03-17    51         10.4        0               0
##  8 Alabama        2020-03-18    61         12.4        0               0
##  9 Alabama        2020-03-19    88         17.9        0               0
## 10 Alabama        2020-03-20   115         23.5        0               0
## # i 61,029 more rows
## # i 2 more variables: mortality_rate <dbl>, Population <dbl>
```

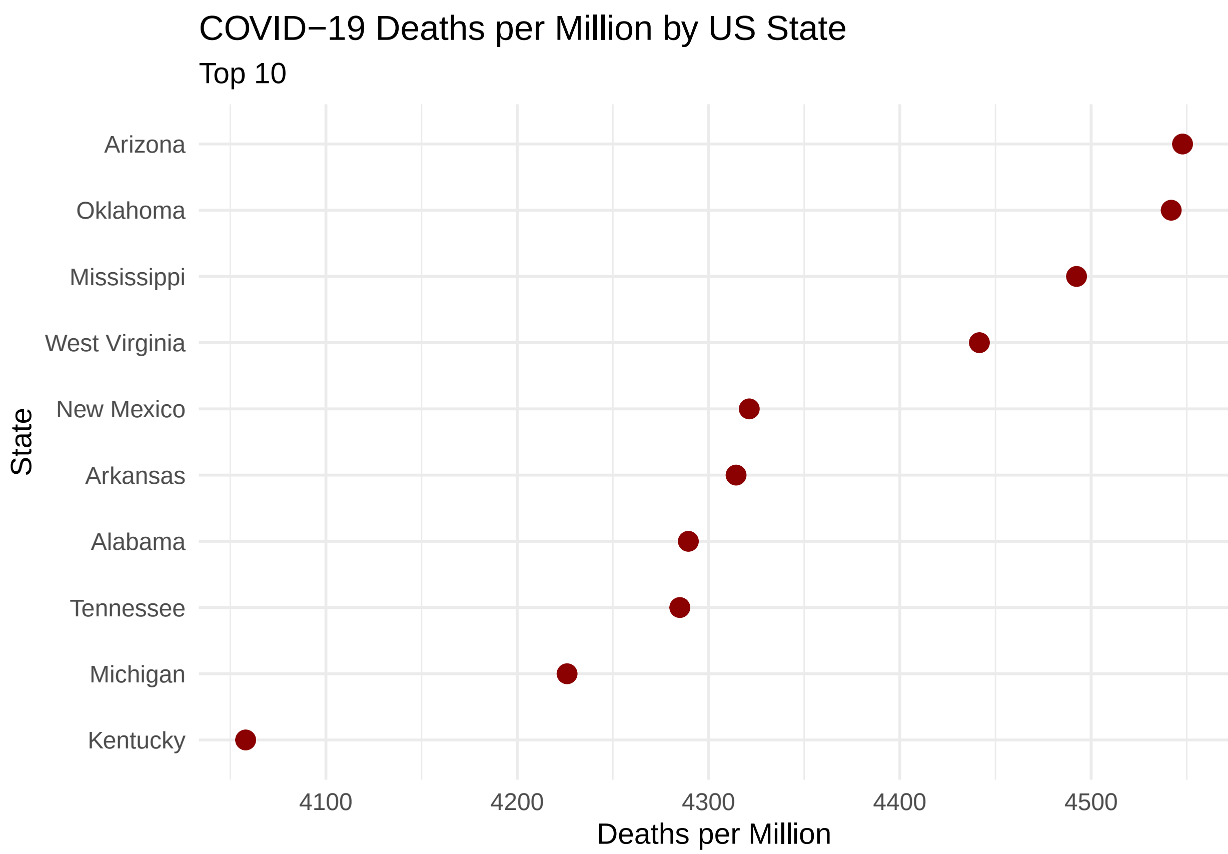
```r
us_states_ovr
```

```
## # A tibble: 56 x 7
##    Province_State     cases cases_per_mill deaths deaths_per_mill mortality_rate
##    <chr>              <dbl>          <dbl>  <dbl>           <dbl>          <dbl>
##  1 Alabama           1.64e6        335401.  21032           4289.        0.0128
##  2 Alaska            3.08e5        422134.   1486           2039.        0.00483
##  3 American Samoa    8.32e3        149530.     34            611.        0.00409
##  4 Arizona           2.44e6        335707.  33102           4548.        0.0135
##  5 Arkansas          1.01e6        333648.  13020           4314.        0.0129
##  6 California        1.21e7        306986. 101159           2560.        0.00834
##  7 Colorado          1.76e6        306387.  14181           2463.        0.00804
##  8 Connecticut       9.77e5        273935.  12220           3427.        0.0125
##  9 Delaware          3.31e5        339706.   3324           3414.        0.0100
## 10 District of Colu~ 1.78e5        252136.   1432           2029.        0.00805
## # i 46 more rows
## # i 1 more variable: Population <dbl>
```

Now I will plot my Death per Million variable to identify the top 10 states that were most affected by the COVID-19 deaths.

```r
top_10_states <- us_states_ovr %>%
  arrange(desc(deaths_per_mill)) %>%
  head(10)
```

```r
ggplot(top_10_states, aes(x = deaths_per_mill, y = reorder(Province_State, deaths_per_mill))) +
  geom_point(color = "darkred", size = 3) +
  labs(title = "COVID-19 Deaths per Million by US State",
       subtitle = "Top 10",
       x = "Deaths per Million",
       y = "State") +
  theme_minimal()
```



The plot shows that relative to population, Arizona was the state most affected by COVID-19 deaths.

Objective #2

For my second objective of determining how the US's mortality rate compares to the rest of the world, I will now perform the same summarizations and create the same variables, but instead grouping on a national level. I will have 2 data-frames, one containing time-series data and another with a cumulative total.

```r
us_pop <- us %>%
  distinct(Country_Region, Province_State, County, .keep_all = TRUE) %>%
  group_by(Country_Region) %>%
  summarize(Population = sum(Population))

us_totals <- us %>%
  group_by(Country_Region, date) %>%
  summarize(cases = sum(cases),
            deaths = sum(deaths)) %>%
```

```r
  ungroup() %>%

  left_join(us_pop, by = "Country_Region") %>%
  filter(Population > 0) %>%
  filter(!is.na(Population)) %>%

  mutate(deaths_per_mill = deaths * 1000000 / Population,
         cases_per_mill = cases * 1000000 / Population,
         mortality_rate = deaths/ cases) %>%
  select(Country_Region, date, cases, cases_per_mill, deaths, deaths_per_mill, mortality_rate, Populatio

us_ovr <- us_totals %>%
  group_by(Country_Region) %>%
  filter(date == max(date)) %>%
  ungroup() %>%
  select(Country_Region, cases, cases_per_mill, deaths, deaths_per_mill, mortality_rate, Population)

us_totals
```

```
## # A tibble: 1,143 x 8
##    Country_Region date       cases cases_per_mill deaths deaths_per_mill
##    <chr>          <date>     <dbl>          <dbl>  <dbl>           <dbl>
##  1 US             2020-01-22     1        0.00301      0               0
##  2 US             2020-01-23     1        0.00301      0               0
##  3 US             2020-01-24     2        0.00602      0               0
##  4 US             2020-01-25     2        0.00602      0               0
##  5 US             2020-01-26     5        0.0150       0               0
##  6 US             2020-01-27     5        0.0150       0               0
##  7 US             2020-01-28     5        0.0150       0               0
##  8 US             2020-01-29     6        0.0180       0               0
##  9 US             2020-01-30     6        0.0180       0               0
## 10 US             2020-01-31     8        0.0241       0               0
## # i 1,133 more rows
## # i 2 more variables: mortality_rate <dbl>, Population <dbl>
```

```r
us_ovr
```

```
## # A tibble: 1 x 7
##   Country_Region     cases cases_per_mill  deaths deaths_per_mill mortality_rate
##   <chr>              <dbl>          <dbl>   <dbl>           <dbl>          <dbl>
## 1 US             103802702        312263. 1122724           3377.         0.0108
## # i 1 more variable: Population <dbl>
```

The same data-frames will now be built using the global data.

```r
global_pop <- global %>%
  distinct(Country_Region, Province_State, .keep_all = TRUE) %>%
  group_by(Country_Region) %>%
  summarize(Population = sum(Population))

global_totals <- global %>%
  group_by(Country_Region, date) %>%
```

```r
  summarize(cases = sum(cases),
            deaths = sum(deaths)) %>%
  ungroup() %>%

  left_join(global_pop, by = "Country_Region") %>%
  filter(Population > 0) %>%
  filter(!is.na(Population)) %>%

  mutate(deaths_per_mill = deaths * 1000000 / Population,
         cases_per_mill = cases * 1000000 / Population,
         mortality_rate = deaths/ cases) %>%
  select(Country_Region, date, cases, cases_per_mill, deaths, deaths_per_mill, mortality_rate, Populati

global_ovr <- global_totals %>%
  group_by(Country_Region) %>%
  filter(date == max(date),
         cases > 1) %>%
  ungroup() %>%
  select(Country_Region, cases, cases_per_mill, deaths, deaths_per_mill, mortality_rate, Population)

global_totals
```

```
## # A tibble: 208,133 x 8
##    Country_Region date       cases cases_per_mill deaths deaths_per_mill
##    <chr>          <date>     <dbl>          <dbl>  <dbl>           <dbl>
##  1 Afghanistan    2020-02-24     5          0.128      0               0
##  2 Afghanistan    2020-02-25     5          0.128      0               0
##  3 Afghanistan    2020-02-26     5          0.128      0               0
##  4 Afghanistan    2020-02-27     5          0.128      0               0
##  5 Afghanistan    2020-02-28     5          0.128      0               0
##  6 Afghanistan    2020-02-29     5          0.128      0               0
##  7 Afghanistan    2020-03-01     5          0.128      0               0
##  8 Afghanistan    2020-03-02     5          0.128      0               0
##  9 Afghanistan    2020-03-03     5          0.128      0               0
## 10 Afghanistan    2020-03-04     5          0.128      0               0
## # i 208,123 more rows
## # i 2 more variables: mortality_rate <dbl>, Population <dbl>
```

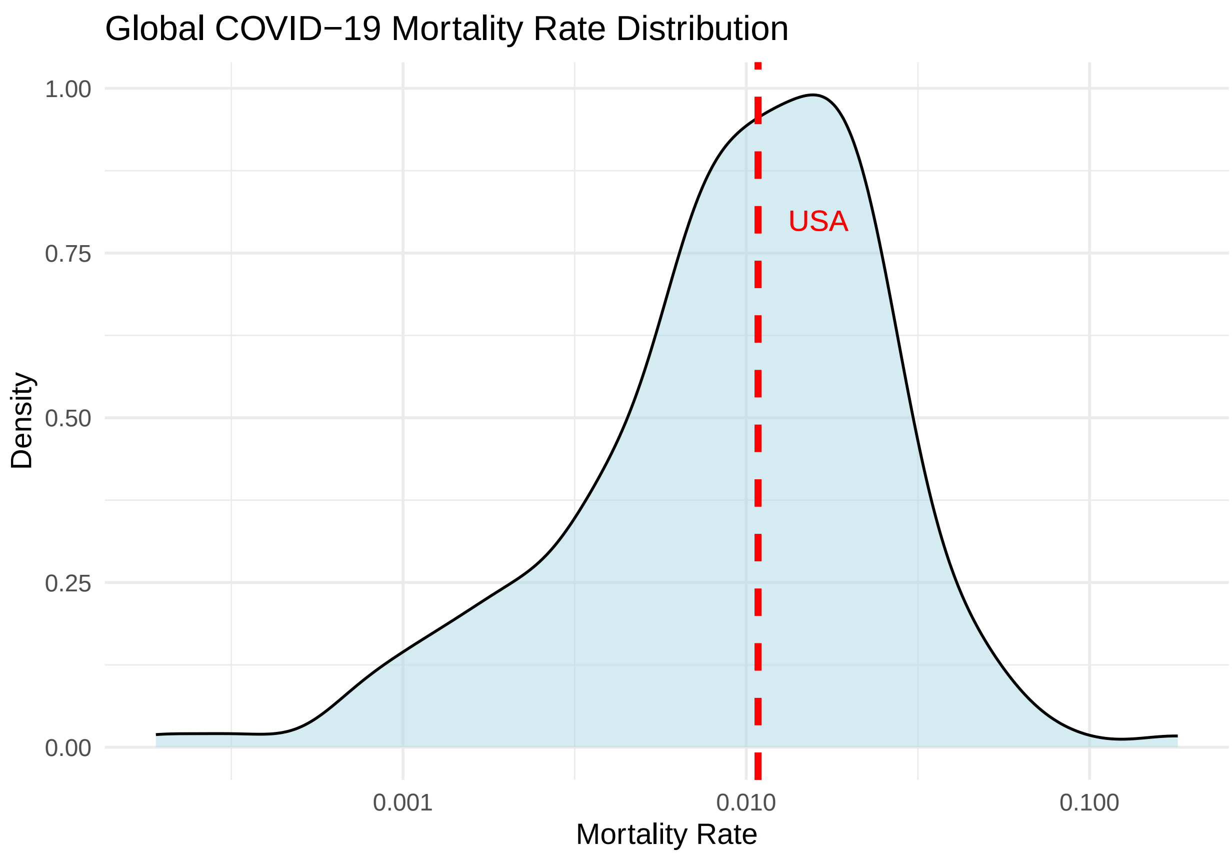
```r
global_ovr
```

```
## # A tibble: 193 x 7
##    Country_Region     cases cases_per_mill deaths deaths_per_mill mortality_rate
##    <chr>              <dbl>          <dbl>  <dbl>           <dbl>          <dbl>
##  1 Afghanistan       2.09e5          5380.   7896            203.        0.0377
##  2 Albania           3.34e5        116220.   3598           1250.        0.0108
##  3 Algeria           2.71e5          6191.   6881            157.        0.0253
##  4 Andorra           4.79e4        619815.    165           2136.        0.00345
##  5 Angola            1.05e5          3204.   1933             58.8       0.0184
##  6 Antigua and Barb~ 9.11e3         92987.    146           1491.        0.0160
##  7 Argentina         1.00e7        222254. 130472           2887.        0.0130
##  8 Armenia           4.47e5        150953.   8727           2945.        0.0195
##  9 Australia         1.14e7        447745.  19574            769.        0.00172
```

```
## 10 Austria           5.96e6       661879.  21970          2439.         0.00369
## # i 183 more rows
## # i 1 more variable: Population <dbl>
```

Now that my data-frames are complete, I will merge them together so that the data can be plotted. Since there is a large number of different countries in this data, I will be using a density plot to compare the global COVID-19 mortality rates.

```r
merged_data <- bind_rows(global_ovr, us_ovr)

ggplot(merged_data, aes(x = mortality_rate)) +
  geom_density(fill = "lightblue", alpha = 0.5) +
  geom_vline(data = subset(merged_data, Country_Region == "US"),
             aes(xintercept = mortality_rate),
             color = "red", size = 1.2, linetype = "dashed") +
  annotate("text",
           x = subset(merged_data, Country_Region == "US")$mortality_rate,
           y = Inf,
           label = "USA",
           vjust = 8,
           hjust = -.5,
           color = "red") +
  labs(title = "Global COVID-19 Mortality Rate Distribution",
       x = "Mortality Rate",
       y = "Density") +
  scale_x_log10() +
  theme_minimal()
```



The density plot shows that the US has a COVID-19 mortality rate slightly above 1%, which appears to be in line with the global average rate.

Objective 3

For my third and final objective, I will feed the 'US Totals' data-frame into an ARIMA model to predict COVID-19 deaths during the first quarter of 2023. The model will be trained using the data from 2020-2022, and the predicted deaths will be compared to the actual deaths for the first quarter of 2023.

```
model_data <- us_totals %>%
  filter(deaths > 0) %>%
  select(date, deaths)

train_data <- model_data %>% filter(date < as.Date("2023-01-01"))
test_data <- model_data %>% filter(date >= as.Date("2023-01-01"))

ts_train <- ts(train_data$deaths, start = c(2020, 1), frequency = 365)

ts_test <- ts(test_data$deaths, start = c(2023, 1), frequency = 365)

diff_train <- diff(diff(ts_train))

adf.test(diff_train)
```

```
## Warning in adf.test(diff_train): p-value smaller than printed p-value
```

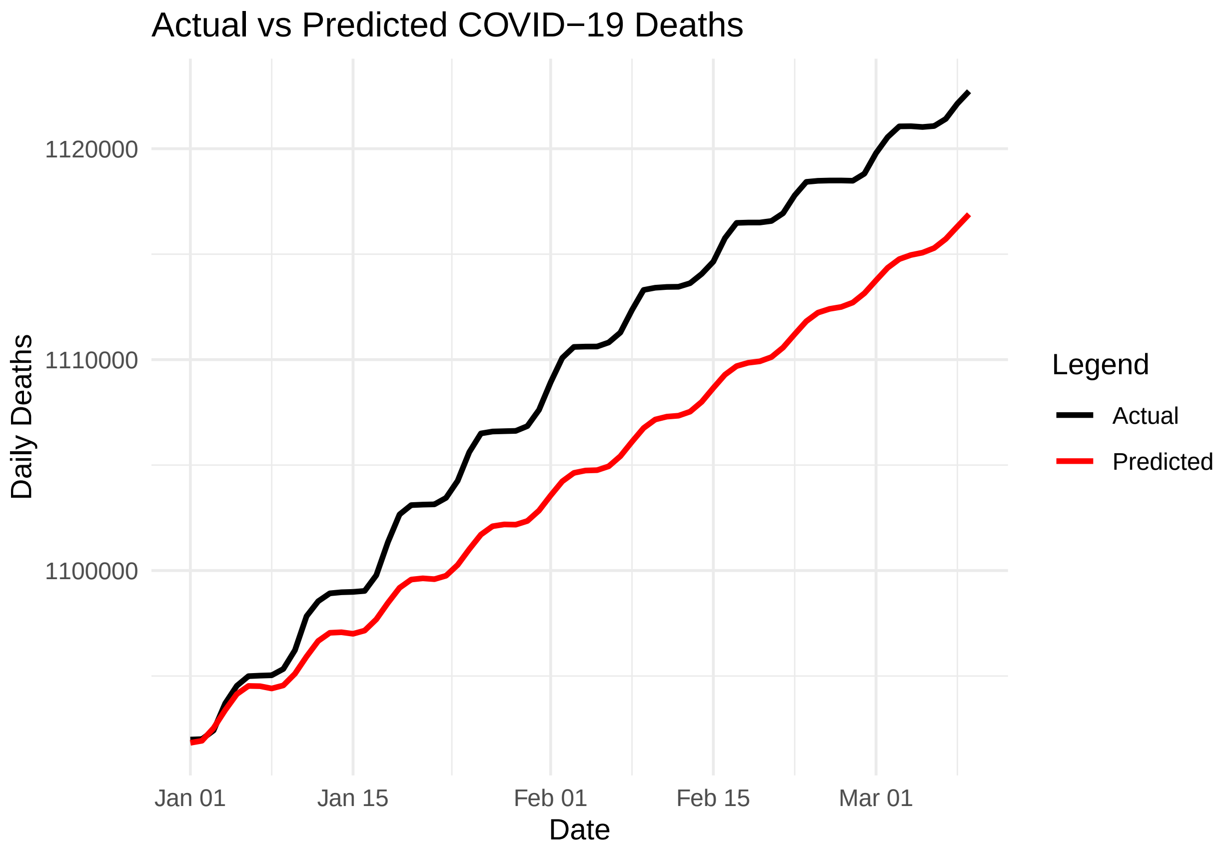
```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff_train
## Dickey-Fuller = -9.8415, Lag order = 10, p-value = 0.01
## alternative hypothesis: stationary
```

```r
arima_model <- auto.arima(diff_train)
summary(arima_model)
```

```
## Series: diff_train
## ARIMA(4,0,2) with zero mean
##
## Coefficients:
##          ar1      ar2      ar3      ar4      ma1     ma2
##       0.4359  -0.4011  -0.2703  -0.4048  -1.2359  0.7379
## s.e.  0.0325   0.0322   0.0307   0.0314   0.0210  0.0314
##
## sigma^2 = 193792:  log likelihood = -7768.06
## AIC=15550.13   AICc=15550.23   BIC=15584.72
##
## Training set error measures:
##                    ME     RMSE      MAE MPE MAPE      MASE       ACF1
## Training set 1.149794 438.9405 279.3877 Inf  Inf 0.4229312 -0.0868795
```

```r
forecasted <- forecast(arima_model, h = length(ts_test))

forecasted_differences <- as.numeric(forecasted$mean)
first_cumsum <- cumsum(forecasted_differences) + as.numeric(tail(diff(ts_train), n = 1))
original_scale_predictions <- cumsum(first_cumsum) + as.numeric(tail(ts_train, n = 1))

predicted_dates <- seq(
  from = as.Date("2023-01-01"),
  by = "day",
  length.out = length(original_scale_predictions)
)

actual_deaths <- model_data %>%
  filter(date <= as.Date("2022-12-31"))

comparison <- bind_rows(
  train_data %>% filter(date >= as.Date("2023-01-01")),
  data.frame(date = predicted_dates, deaths = test_data$deaths, predicted_deaths = original_scale_predi
  filter(year(date) == 2023)

ggplot(comparison, aes(x = date)) +
  geom_line(aes(y = deaths, color = "Actual"), size = 1, na.rm = TRUE) +
  geom_line(aes(y = predicted_deaths, color = "Predicted"), size = 1, na.rm = TRUE) +
  scale_color_manual(values = c("Actual" = "black", "Predicted" = "red")) +
  labs(
    title = "Actual vs Predicted COVID-19 Deaths",
    x = "Date",
    y = "Daily Deaths",
```

```r
    color = "Legend") +
  theme_minimal()
```



Conclusion

Utilizing data sets from Johns Hopkins University, I successfully achieved all research objectives. However, it's crucial to acknowledge potential biases within the analysis. Numerous factors influence COVID-19 cases, deaths, and associated mortality rates. The provided data sets don't account for critical variables like government policy, vaccination rates, or the lag time between diagnosis and death. Within the United States, these variables varied significantly across states and cities. Globally, some countries implemented stringent COVID-19 policies, while others adopted a more relaxed approach. Therefore, when interpreting the results of this analysis, it's essential to remember that the data's inability to account for these variables renders the findings more exploratory than definitively factual.

R Session Info

```
## R version 4.3.3 (2024-02-29)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS 15.5
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib;  LAPACK v
```

```
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: America/New_York
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] tseries_0.10-58 forecast_8.24.0 lubridate_1.9.4 forcats_1.0.0
##  [5] stringr_1.5.1   dplyr_1.1.4     purrr_1.0.2     readr_2.1.5
##  [9] tidyr_1.3.1     tibble_3.2.1    ggplot2_3.5.1   tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] utf8_1.2.4        generics_0.1.4    stringi_1.8.3     lattice_0.22-5
##  [5] hms_1.1.3         digest_0.6.35     magrittr_2.0.3    evaluate_0.23
##  [9] grid_4.3.3        timechange_0.3.0  fastmap_1.2.0     nnet_7.3-19
## [13] fansi_1.0.6       scales_1.3.0      cli_3.6.2         crayon_1.5.2
## [17] rlang_1.1.3       bit64_4.6.0-1     munsell_0.5.1     withr_3.0.0
## [21] yaml_2.3.8        tools_4.3.3       parallel_4.3.3    tzdb_0.5.0
## [25] colorspace_2.1-0  curl_6.3.0        vctrs_0.6.5       R6_2.5.1
## [29] zoo_1.8-14        lifecycle_1.0.4   bit_4.6.0         vroom_1.6.5
## [33] urca_1.3-4        pkgconfig_2.0.3   pillar_1.9.0      gtable_0.3.5
## [37] quantmod_0.4.28   glue_1.7.0        Rcpp_1.0.14       highr_0.10
## [41] xfun_0.52         lmtest_0.9-40     tidyselect_1.2.1  rstudioapi_0.16.0
## [45] knitr_1.45        farver_2.1.1      nlme_3.1-164      htmltools_0.5.8.1
## [49] labeling_0.4.3    xts_0.14.1        rmarkdown_2.29    timeDate_4041.110
## [53] fracdiff_1.5-3    compiler_4.3.3    quadprog_1.5-8    TTR_0.24.4
```