COVID-19 Data Analysis

Anonymous

2025-04-20

Objective

I will use the following data sets from Johns Hopkins University to analyze data pertaining to the COVID-19 pandemic. My objective is to answer the questions below, is well as using an ARIMA model to predict COVID-19 deaths in the US during the first quarter of 2023. 1. Which US state's population was most affected by the COVID-19 virus? 2. How did the United State's mortality rate compare to the rest of the world?

Data Overview

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

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

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

uid_lookup_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/U
```

Now I will read in the data and take an initial look at it.

```
#read in JHU csv files
us_cases <- read_csv(urls[1])
global_cases <- read_csv(urls[2])
us_deaths <- read_csv(urls[3])
global_deaths <- read_csv(urls[4])
global_population <- read_csv(uid_lookup_url)

#view data
us_cases
```

```
## # 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>, ...
```

```
global_cases
```

```
## # 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>, ...
```

```
us_deaths
```

```
## # 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>, ...
```

```
global_deaths
```

```
## # 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>, ...
```

```
global_population
```

```
## # 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>
```

The COVID data sets contain variables such as province/state, country/region, along with other identifiers that do not initially appear to helpful to my analysis. Each date has a corresponding column, which I will pivot into rows to make more time-series friendly. The population data has one row per city/county, represented by the 'Combined_Key' column.

Tidy and Transform Data

After taking an initial look at the data, it is evident that some tidying and transformation needs to be done. To start, I will tidy up the global data by pivoting the case and death data and removing some unnecessary columns. I will also filter out all rows where cases are not more than zero, and ensure there is a date column in the correct format.

```
#pivot and remove unnecessary 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))

#pivot and remove unnecessary 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))

#join both data sets
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)`
```

```
#view new data set
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
```

The same transformations will be applied to the US data.

```
#pivot and remove unnecessary columns
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_))

#pivot and remove unnecessary columns
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_))

#join both data sets
us <- us_cases %>%
  full_join(us_deaths) %>%
  filter(cases > 0) %>%
  rename(County = "Admin2")

#view new data set
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>
```

In order to truly compare the US data to the global data, I need to create a mutual column and bring

populations into the global data set. The column will be called "Combined_Key", which mimics the same column in the US data.

```
#create a final global data set by performing a left join to only bring in relevant population data
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)

#create combined_key by combining province_state and country_region
global <- global %>%
  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 my analysis.

```
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>
```

```
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
```

```
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.

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

#Aggregate cases/deaths
```

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

  #bring in population data
  left_join(state_pop, by = "Province_State") %>%
  filter(Population > 0) %>%
  filter(!is.na(Population)) %>%

  #Create new variables
  mutate(deaths_per_mill = deaths * 1000000 / Population,
         cases_per_mill = cases * 1000000 / Population,
         mortality_rate = deaths/ cases) %>%
  select(Province_State, date, cases, cases_per_mill, deaths, deaths_per_mill, mortality_rate, Populati

#one row per state with accurate totals
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>
```

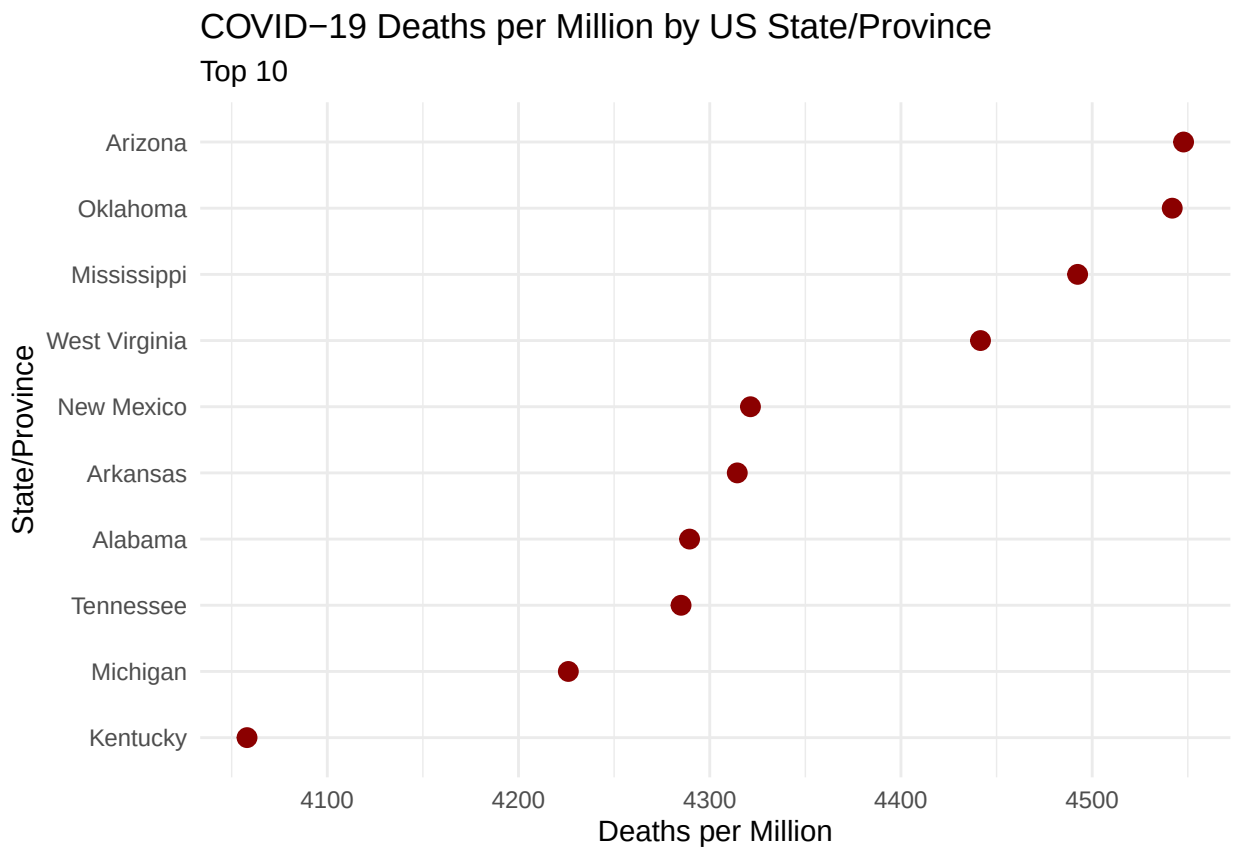
```r
us_states_ovr
```

```
## # 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.

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

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/Province",
       subtitle = "Top 10",
       x = "Deaths per Million",
       y = "State/Province") +
  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.

```
#Get total US population
us_pop <- us %>%
  distinct(Country_Region, Province_State, County, .keep_all = TRUE) %>%
  group_by(Country_Region) %>%
  summarize(Population = sum(Population))

#Aggregate cases/deaths
us_totals <- us %>%
  group_by(Country_Region, date) %>%
  summarize(cases = sum(cases),
            deaths = sum(deaths)) %>%
  ungroup() %>%

  #bring in population data
  left_join(us_pop, by = "Country_Region") %>%
  filter(Population > 0) %>%
  filter(!is.na(Population)) %>%

  #Create new variables
  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

#one row per state with accurate totals
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>
```

```
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.

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

#Aggregate cases/deaths globally
global_totals <- global %>%
  group_by(Country_Region, date) %>%
  summarize(cases = sum(cases),
            deaths = sum(deaths)) %>%
  ungroup() %>%

  #bring in population data
  left_join(global_pop, by = "Country_Region") %>%
  filter(Population > 0) %>%
  filter(!is.na(Population)) %>%

  #Create new variables
  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

#one row per country with accurate totals
global_ovr <- global_totals %>%
  group_by(Country_Region) %>%
  filter(date == max(date),
         #Filter out North Korea
         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
```

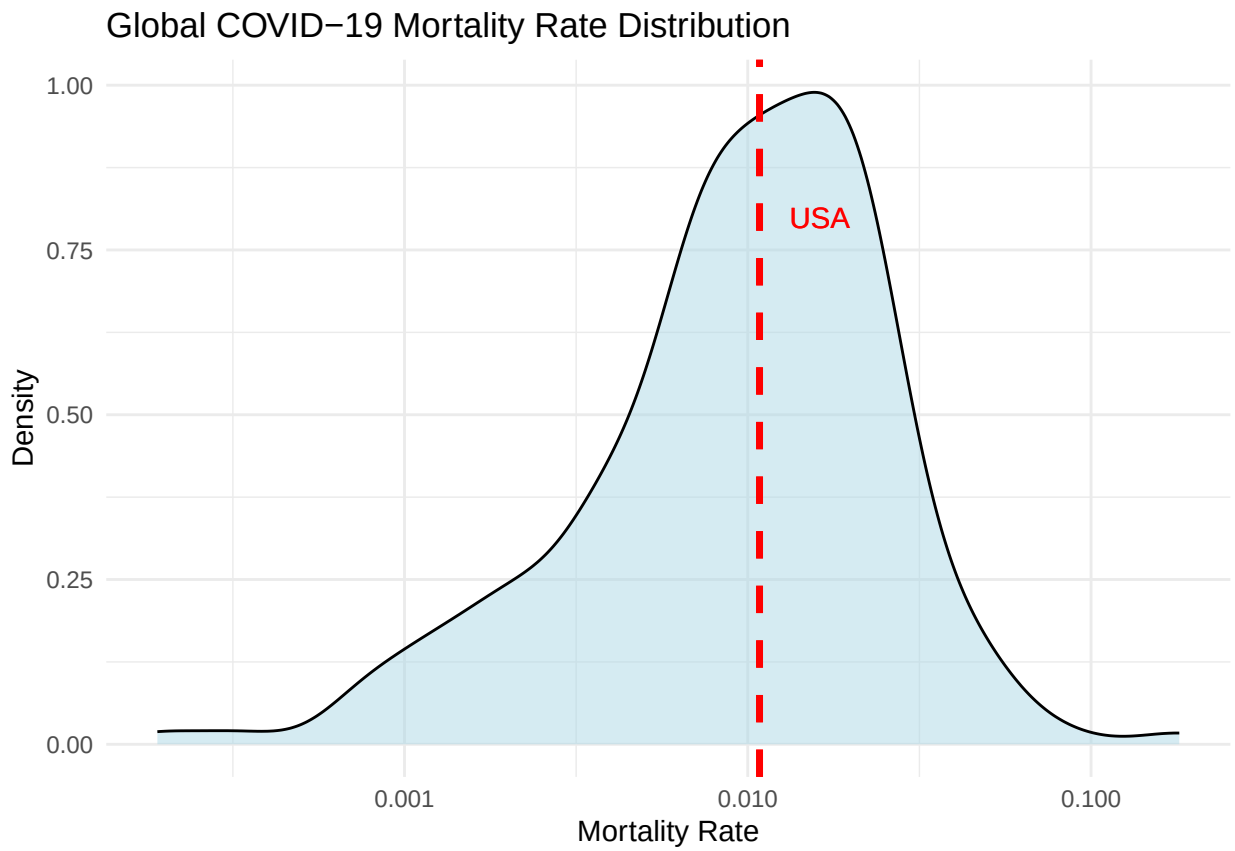
```
##  6 Afghanistan    2020-02-29     5          0.128      0              0
##  7 Afghanistan    2020-03-01     5          0.128      0              0
##  8 Afghanistan    2020-03-02     5          0.128      0              0
##  9 Afghanistan    2020-03-03     5          0.128      0              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>
```

```
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.

```
#append US summary
merged_data <- bind_rows(global_ovr, us_ovr)

#Density plot
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.

```
#Filter out days with 0 deaths
model_data <- us_totals %>%
  filter(deaths > 0) %>%
  select(date, deaths)

# Split data into training (2020-2022) and testing (2023)
train_data <- model_data %>% filter(date < as.Date("2023-01-01"))
test_data <- model_data %>% filter(date >= as.Date("2023-01-01"))


# Convert training data to time series object
ts_train <- ts(train_data$deaths, start = c(2020, 1), frequency = 365)

# Convert testing data to time series object
ts_test <- ts(test_data$deaths, start = c(2023, 1), frequency = 365)

# Apply 2nd Differencing on training data for stationarity
```

```r
diff_train <- diff(diff(ts_train))

# p-value is greater than .05, stationarity achieved
adf.test(diff_train)
```

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

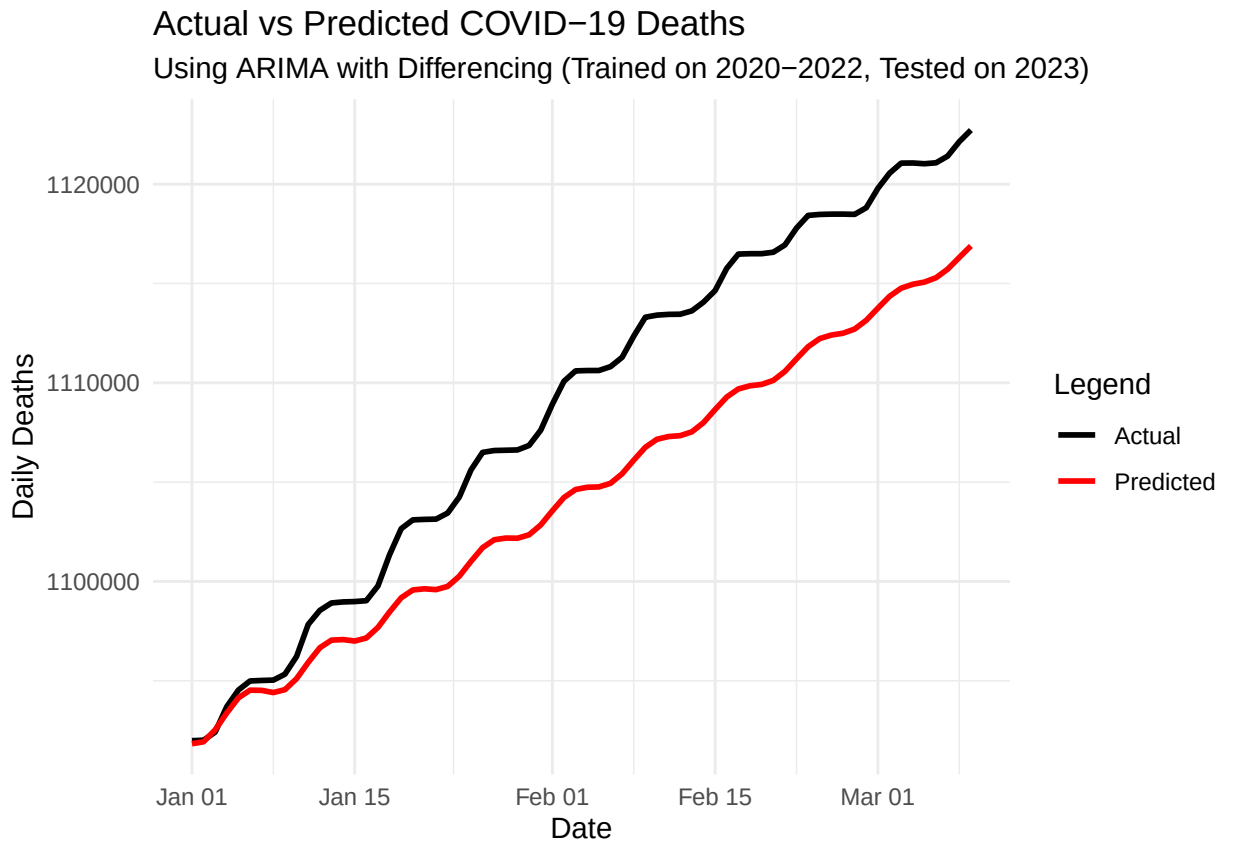
```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff_train
## Dickey-Fuller = -9.8415, Lag order = 10, p-value = 0.01
## alternative hypothesis: stationary
```

```r
# Fit ARIMA model to training data only
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
# Forecast for the length of the testing set
forecasted <- forecast(arima_model, h = length(ts_test))

# Reverse differencing for testing period
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))

# Create results data frame for predictions
predicted_dates <- seq(
  from = as.Date("2023-01-01"),
  by = "day",
  length.out = length(original_scale_predictions)
)

#Actual deaths for 2020-2022
actual_deaths <- model_data %>%
  filter(date <= as.Date("2022-12-31"))
```

```r
#Combine actual and predicted deaths
comparison <- bind_rows(
  train_data %>% filter(date >= as.Date("2023-01-01")),  # Keep actual deaths for 2023
  data.frame(date = predicted_dates, deaths = test_data$deaths, predicted_deaths = original_scale_predi
  filter(year(date) == 2023)  # Filter to include only 2023 data

#Plot predicted vs actual deaths
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",
    subtitle = "Using ARIMA with Differencing (Trained on 2020-2022, Tested on 2023)",
    x = "Date",
    y = "Daily Deaths",
    color = "Legend") +
  theme_minimal()
```



Conclusion

By using the data sets from Johns Hopkins University, I was able to complete all my objectives. However, it is important to discuss potential biases in my analysis. There are many factors that influence COVID-19 cases, deaths, and by association, mortality rates. the data sets provided do not account for variables such

as government policy, vaccine rates, or the time between diagnosis and death (lag). In the United States, these variables would be different across both states and cities. Globally, many countries had very strict COVID-19 policies, while many countries did not have much policy at all. When interpreting the results of my analysis, it is important to remember that the data does not account for these types of variables, making my findings more exploratory than factual.