Final Project 2: Reproducible Report on COVID19 Data"

TSP

11/30/2022

#Peer-graded Assignement: NYPD Shooting Incident Data Report

Assignement Tasks: Import, tidy and analyze the COVID19 dataset from the Johns Hopkins github site. This is the same dataset I used in class. Feel free to repeat and reuse what I did if you want to. Be sure your project is reproducible and contains some visualization and analysis that is unique to your project. You may use the data to do any analysis that is of interest to you. You should include at least two visualizations and one model. Be sure to identify any bias possible in the data and in your analysis.

#Step 1: Install packages and enable the package required for data analysis

#Step 2: Import data from COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) # at Johns Hopkins University

```
url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_cov
file_names <- c("time_series_covid19_confirmed_global.csv",
                "time_series_covid19_deaths_global.csv",
                "time_series_covid19_confirmed_US.csv",
                "time_series_covid19_deaths_US.csv")
urls <- str_c(url_in,file_names)
global_cases <- read_csv(urls[1])
```

```
## Rows: 289 Columns: 1048
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1046): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

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

```
## Rows: 289 Columns: 1048
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1046): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

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

```
## Rows: 3342 Columns: 1055
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1049): UID, code3, FIPS, Lat, Long_, 1/22/20, 1/23/20, 1/24/20, 1/25/20...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

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

```
## Rows: 3342 Columns: 1056
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1050): 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.
```

#Step 3: Clean and Tidy data

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

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

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

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

```
#Check ddescriptive statitic and information
summary(global)
```

```
##  Province_State     Country_Region          date                  cases
##  Length:301716      Length:301716      Min.   :2020-01-22   Min.   :       0
##  Class :character   Class :character   1st Qu.:2020-10-08   1st Qu.:     508
##  Mode  :character   Mode  :character   Median :2021-06-26   Median :   11566
##                                        Mean   :2021-06-26   Mean   :  832293
##                                        3rd Qu.:2022-03-14   3rd Qu.:  192872
##                                        Max.   :2022-11-30   Max.   :98788140
##      deaths
##  Min.   :      0
##  1st Qu.:      3
##  Median :    125
##  Mean   :  12428
##  3rd Qu.:   2654
##  Max.   :1080444
```

```
# Remove the zero case
global <- global %>% filter(cases > 0)
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:278414      Length:278414      Min.   :2020-01-22   Min.   :       1
##  Class :character   Class :character   1st Qu.:2020-11-16   1st Qu.:    1025
##  Mode  :character   Mode  :character   Median :2021-07-27   Median :   16700
##                                        Mean   :2021-07-23   Mean   :  901952
##                                        3rd Qu.:2022-04-01   3rd Qu.:  236296
##                                        Max.   :2022-11-30   Max.   :98788140
##      deaths
##  Min.   :      0
##  1st Qu.:      7
##  Median :    185
##  Mean   :  13468
##  3rd Qu.:   3204
##  Max.   :1080444
```

```
# Check maximum case whether it is correct or not
global <- global %>% filter(cases > 28000000)
global
```

```
## # A tibble: 1,855 x 5
##    Province_State Country_Region date          cases deaths
##    <chr>          <chr>          <date>        <dbl>  <dbl>
##  1 <NA>           Brazil         2022-02-18 28072238 643340
##  2 <NA>           Brazil         2022-02-19 28177367 644195
##  3 <NA>           Brazil         2022-02-20 28218180 644592
##  4 <NA>           Brazil         2022-02-21 28258458 644918
##  5 <NA>           Brazil         2022-02-22 28361951 645735
##  6 <NA>           Brazil         2022-02-23 28493336 646714
##  7 <NA>           Brazil         2022-02-24 28589235 647703
##  8 <NA>           Brazil         2022-02-25 28679671 648496
##  9 <NA>           Brazil         2022-02-26 28749552 649184
## 10 <NA>           Brazil         2022-02-27 28776794 649437
## # ... with 1,845 more rows
```

```
#Tidy US data
US_cases
```

```
## # A tibble: 3,342 x 1,055
##         UID iso2  iso3  code3  FIPS Admin2   Provi~1 Count~2   Lat Long_ Combi~3
##       <dbl> <chr> <chr> <dbl> <dbl> <chr>    <chr>   <chr>   <dbl> <dbl> <chr>
##  1 84001001 US    USA     840  1001 Autauga  Alabama US       32.5 -86.6 Autaug~
##  2 84001003 US    USA     840  1003 Baldwin  Alabama US       30.7 -87.7 Baldwi~
##  3 84001005 US    USA     840  1005 Barbour  Alabama US       31.9 -85.4 Barbou~
##  4 84001007 US    USA     840  1007 Bibb     Alabama US       33.0 -87.1 Bibb, ~
##  5 84001009 US    USA     840  1009 Blount   Alabama US       34.0 -86.6 Blount~
##  6 84001011 US    USA     840  1011 Bullock  Alabama US       32.1 -85.7 Bulloc~
##  7 84001013 US    USA     840  1013 Butler   Alabama US       31.8 -86.7 Butler~
##  8 84001015 US    USA     840  1015 Calhoun  Alabama US       33.8 -85.8 Calhou~
##  9 84001017 US    USA     840  1017 Chambers Alabama US       32.9 -85.4 Chambe~
## 10 84001019 US    USA     840  1019 Cherokee Alabama US       34.2 -85.6 Cherok~
## # ... with 3,332 more rows, 1,044 more variables: ‘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>, ...
```

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

US_cases
```

```
## # A tibble: 3,489,048 x 6
##    Admin2  Province_State Country_Region Combined_Key         date       cases
##    <chr>   <chr>          <chr>          <chr>                <date>     <dbl>
##  1 Autauga Alabama        US             Autauga, Alabama, US 2020-01-22     0
##  2 Autauga Alabama        US             Autauga, Alabama, US 2020-01-23     0
##  3 Autauga Alabama        US             Autauga, Alabama, US 2020-01-24     0
##  4 Autauga Alabama        US             Autauga, Alabama, US 2020-01-25     0
##  5 Autauga Alabama        US             Autauga, Alabama, US 2020-01-26     0
##  6 Autauga Alabama        US             Autauga, Alabama, US 2020-01-27     0
##  7 Autauga Alabama        US             Autauga, Alabama, US 2020-01-28     0
##  8 Autauga Alabama        US             Autauga, Alabama, US 2020-01-29     0
##  9 Autauga Alabama        US             Autauga, Alabama, US 2020-01-30     0
## 10 Autauga Alabama        US             Autauga, Alabama, US 2020-01-31     0
## # ... with 3,489,038 more rows
```

```
US_deaths
```

```
## # A tibble: 3,342 x 1,056
##          UID iso2  iso3  code3  FIPS Admin2   Provi~1 Count~2   Lat Long_ Combi~3
##        <dbl> <chr> <chr> <dbl> <dbl> <chr>    <chr>   <chr>   <dbl> <dbl> <chr>
##  1 84001001 US    USA     840  1001 Autauga  Alabama US       32.5 -86.6 Autaug~
##  2 84001003 US    USA     840  1003 Baldwin  Alabama US       30.7 -87.7 Baldwi~
##  3 84001005 US    USA     840  1005 Barbour  Alabama US       31.9 -85.4 Barbou~
##  4 84001007 US    USA     840  1007 Bibb     Alabama US       33.0 -87.1 Bibb, ~
##  5 84001009 US    USA     840  1009 Blount   Alabama US       34.0 -86.6 Blount~
##  6 84001011 US    USA     840  1011 Bullock  Alabama US       32.1 -85.7 Bulloc~
##  7 84001013 US    USA     840  1013 Butler   Alabama US       31.8 -86.7 Butler~
##  8 84001015 US    USA     840  1015 Calhoun  Alabama US       33.8 -85.8 Calhou~
##  9 84001017 US    USA     840  1017 Chambers Alabama US       32.9 -85.4 Chambe~
## 10 84001019 US    USA     840  1019 Cherokee Alabama US       34.2 -85.6 Cherok~
## # ... with 3,332 more rows, 1,045 more variables: 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>, ...
```

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

```
## # A tibble: 3,489,048 x 7
##    Admin2  Province_State Country_Region Combined_Key  Popul~1 date       deaths
##    <chr>   <chr>          <chr>          <chr>           <dbl> <date>      <dbl>
##  1 Autauga Alabama        US             Autauga, Ala~   55869 2020-01-22      0
##  2 Autauga Alabama        US             Autauga, Ala~   55869 2020-01-23      0
##  3 Autauga Alabama        US             Autauga, Ala~   55869 2020-01-24      0
##  4 Autauga Alabama        US             Autauga, Ala~   55869 2020-01-25      0
##  5 Autauga Alabama        US             Autauga, Ala~   55869 2020-01-26      0
##  6 Autauga Alabama        US             Autauga, Ala~   55869 2020-01-27      0
##  7 Autauga Alabama        US             Autauga, Ala~   55869 2020-01-28      0
##  8 Autauga Alabama        US             Autauga, Ala~   55869 2020-01-29      0
##  9 Autauga Alabama        US             Autauga, Ala~   55869 2020-01-30      0
## 10 Autauga Alabama        US             Autauga, Ala~   55869 2020-01-31      0
## # ... with 3,489,038 more rows, and abbreviated variable name 1: Population
```

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

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

```r
US
```

```
## # A tibble: 3,489,048 x 8
##    Admin2  Province_State Country_Region Combi~1 date       cases Popul~2 deaths
##    <chr>   <chr>          <chr>          <chr>   <date>     <dbl>   <dbl>  <dbl>
##  1 Autauga Alabama        US             Autaug~ 2020-01-22     0   55869      0
##  2 Autauga Alabama        US             Autaug~ 2020-01-23     0   55869      0
##  3 Autauga Alabama        US             Autaug~ 2020-01-24     0   55869      0
##  4 Autauga Alabama        US             Autaug~ 2020-01-25     0   55869      0
##  5 Autauga Alabama        US             Autaug~ 2020-01-26     0   55869      0
##  6 Autauga Alabama        US             Autaug~ 2020-01-27     0   55869      0
##  7 Autauga Alabama        US             Autaug~ 2020-01-28     0   55869      0
##  8 Autauga Alabama        US             Autaug~ 2020-01-29     0   55869      0
##  9 Autauga Alabama        US             Autaug~ 2020-01-30     0   55869      0
## 10 Autauga Alabama        US             Autaug~ 2020-01-31     0   55869      0
## # ... with 3,489,038 more rows, and abbreviated variable names 1: Combined_Key,
## #   2: Population
```

```
#Add population to global data with look up table
global <- global %>%
  unite("Combined_Key",
        c(Province_State, Country_Region),
        sep = ", ",
        na.rm = TRUE,
        remove = FALSE)

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

```
## Rows: 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.
```

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

```
## # A tibble: 1,855 x 7
##    Province_State Country_Region date          cases deaths Population Combine~1
##    <chr>          <chr>          <date>        <dbl>  <dbl>      <dbl> <chr>
##  1 <NA>           Brazil         2022-02-18 28072238 643340  212559409 Brazil
##  2 <NA>           Brazil         2022-02-19 28177367 644195  212559409 Brazil
##  3 <NA>           Brazil         2022-02-20 28218180 644592  212559409 Brazil
##  4 <NA>           Brazil         2022-02-21 28258458 644918  212559409 Brazil
##  5 <NA>           Brazil         2022-02-22 28361951 645735  212559409 Brazil
##  6 <NA>           Brazil         2022-02-23 28493336 646714  212559409 Brazil
##  7 <NA>           Brazil         2022-02-24 28589235 647703  212559409 Brazil
##  8 <NA>           Brazil         2022-02-25 28679671 648496  212559409 Brazil
##  9 <NA>           Brazil         2022-02-26 28749552 649184  212559409 Brazil
## 10 <NA>           Brazil         2022-02-27 28776794 649437  212559409 Brazil
## # ... with 1,845 more rows, and abbreviated variable name 1: Combined_Key
```

#Step 4: Visualize

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

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

```
US_by_state
```

```
## # A tibble: 60,552 x 7
##    Province_State Country_Region date       cases deaths deaths_per_mil Popula~1
##    <chr>          <chr>          <date>     <dbl>  <dbl>          <dbl>    <dbl>
##  1 Alabama        US             2020-01-22     0      0              0  4903185
##  2 Alabama        US             2020-01-23     0      0              0  4903185
##  3 Alabama        US             2020-01-24     0      0              0  4903185
##  4 Alabama        US             2020-01-25     0      0              0  4903185
##  5 Alabama        US             2020-01-26     0      0              0  4903185
##  6 Alabama        US             2020-01-27     0      0              0  4903185
##  7 Alabama        US             2020-01-28     0      0              0  4903185
##  8 Alabama        US             2020-01-29     0      0              0  4903185
##  9 Alabama        US             2020-01-30     0      0              0  4903185
## 10 Alabama        US             2020-01-31     0      0              0  4903185
## # ... with 60,542 more rows, and abbreviated variable name 1: Population
```

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

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

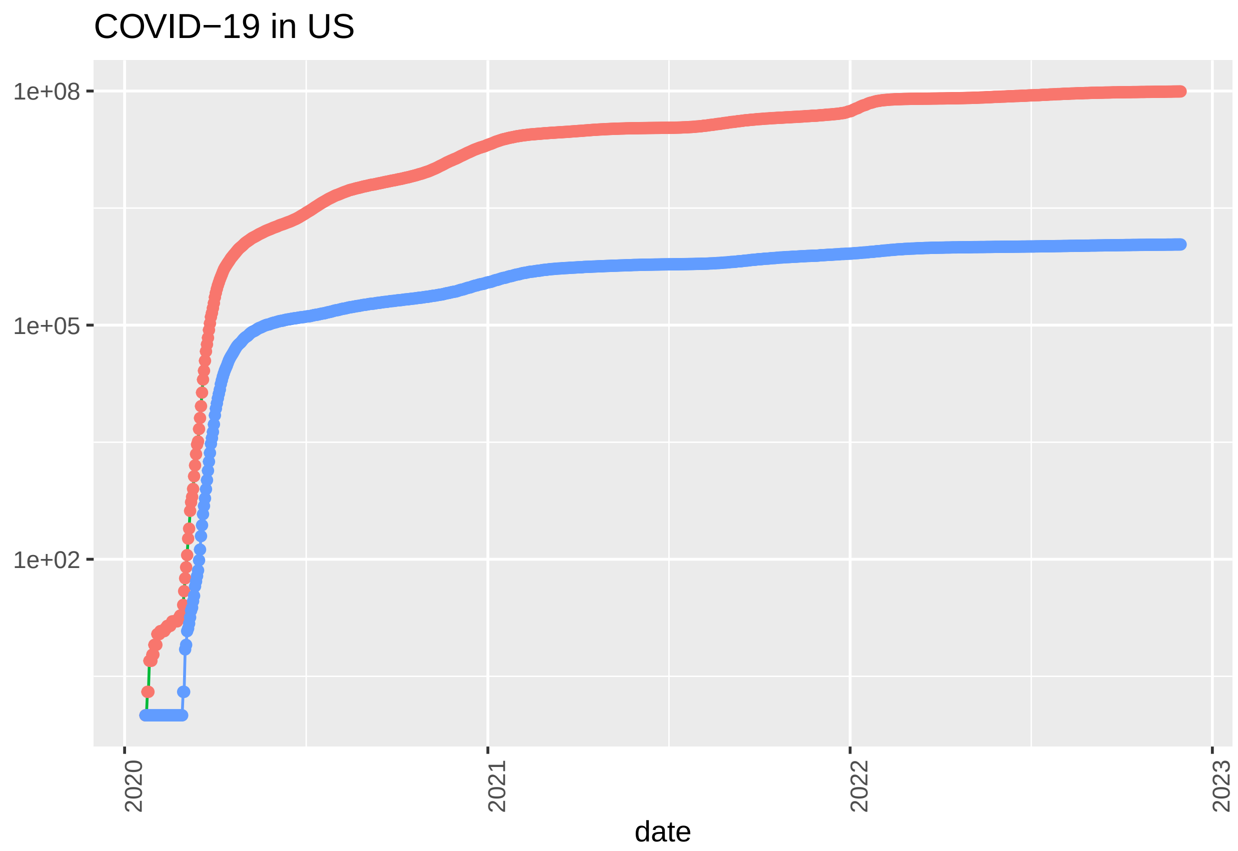
```
US_totals
```

```
## # A tibble: 1,044 x 6
##    Country_Region date       cases deaths deaths_per_mil Population
##    <chr>          <date>     <dbl>  <dbl>          <dbl>      <dbl>
##  1 US             2020-01-22     1      1        0.00300  332875137
##  2 US             2020-01-23     1      1        0.00300  332875137
##  3 US             2020-01-24     2      1        0.00300  332875137
##  4 US             2020-01-25     2      1        0.00300  332875137
##  5 US             2020-01-26     5      1        0.00300  332875137
##  6 US             2020-01-27     5      1        0.00300  332875137
##  7 US             2020-01-28     5      1        0.00300  332875137
##  8 US             2020-01-29     6      1        0.00300  332875137
##  9 US             2020-01-30     6      1        0.00300  332875137
## 10 US             2020-01-31     8      1        0.00300  332875137
## # ... with 1,034 more rows
```

```
tail(US_totals)
```

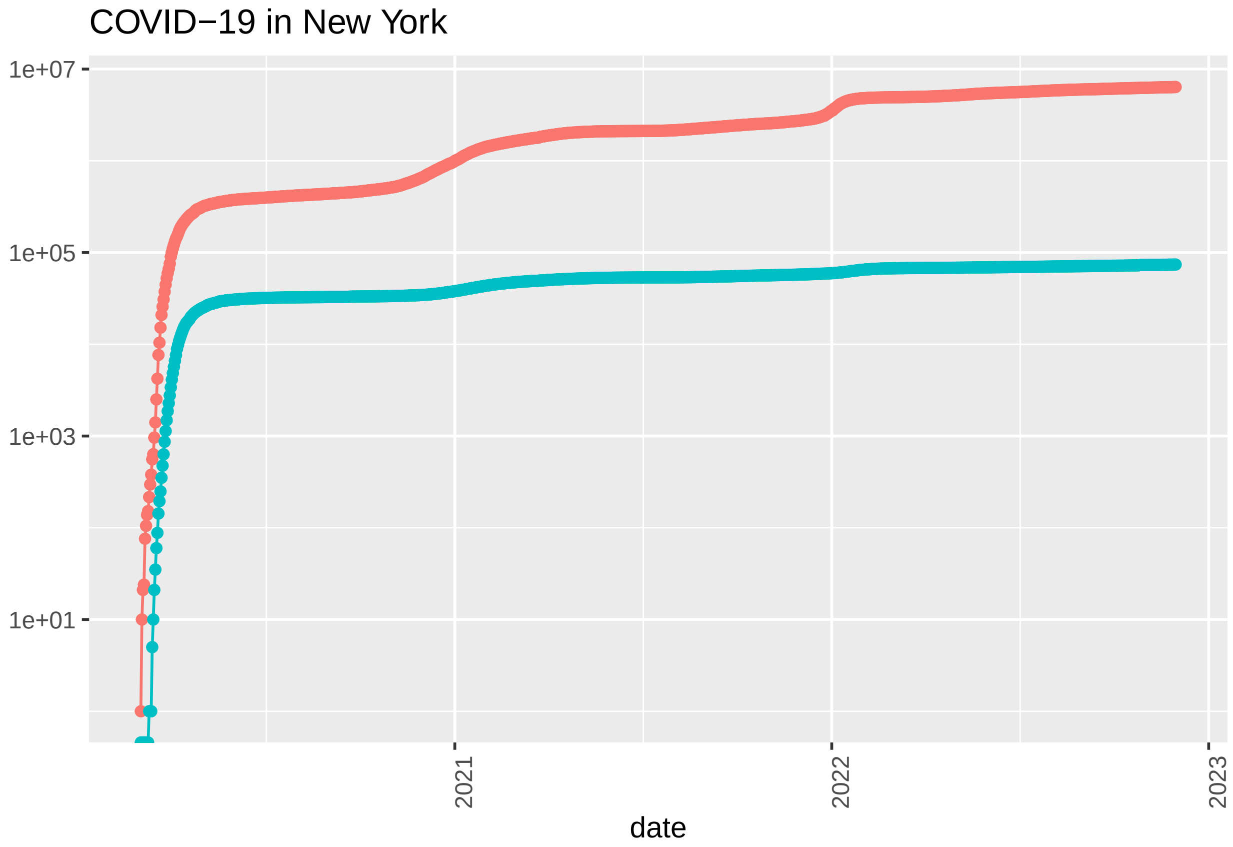
```
## # A tibble: 6 x 6
##   Country_Region date          cases  deaths deaths_per_mil Population
##   <chr>          <date>        <dbl>   <dbl>          <dbl>      <dbl>
## 1 US             2022-11-25 98566003 1079202          3242.  332875137
## 2 US             2022-11-26 98568660 1079204          3242.  332875137
## 3 US             2022-11-27 98573015 1079204          3242.  332875137
## 4 US             2022-11-28 98632732 1079484          3243.  332875137
## 5 US             2022-11-29 98678154 1079877          3244.  332875137
## 6 US             2022-11-30 98788140 1080444          3246.  332875137
```

```
#Perform data visualization - Visualization_1
# Total Covid case by time
US_totals %>%
  filter(cases > 0)%>%
  ggplot(aes(x = date, y=cases)) +
  geom_line(aes(color = "Cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) +
  scale_y_log10() +
  theme(legend.position = " bottom",
        axis.text.x = element_text(angle = 90))+
  labs(title = "COVID-19 in US", y = NULL)
```



```r
#Perform data visualization - Visualization_2
# New Yosk Covid cases by time
state <- "New York"
US_by_state %>%
  filter(Province_State == state) %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y=cases)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases"))+
  geom_line(aes(y = deaths, color = "deaths"))+
  geom_point(aes(y = deaths, color = "deaths"))+
  scale_y_log10() +
  theme(legend.position = " bottom",
        axis.text.x = element_text(angle = 90))+
  labs(title = str_c("COVID-19 in ",state), y = NULL)
```

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



#Step 4: Analyzing

```
#Transform data - add new_cases and new deaths columns
US_by_state <- US_by_state %>%
  mutate(new_cases = cases - lag(cases),
         new_deaths = deaths - lag(deaths))
US_totals <- US_totals %>%
  mutate(new_cases = cases - lag(cases),
         new_deaths = deaths - lag(deaths))
#checking data
tail(US_by_state)
```

```
## # A tibble: 6 x 9
##   Province_St~1 Count~2 date        cases deaths death~3 Popul~4 new_c~5 new_d~6
##   <chr>         <chr>   <date>      <dbl>  <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
## 1 Wyoming       US      2022-11-25 180426   1931   3336.  578759       0       0
## 2 Wyoming       US      2022-11-26 180426   1931   3336.  578759       0       0
## 3 Wyoming       US      2022-11-27 180426   1931   3336.  578759       0       0
## 4 Wyoming       US      2022-11-28 180426   1931   3336.  578759       0       0
## 5 Wyoming       US      2022-11-29 180925   1938   3349.  578759     499       7
## 6 Wyoming       US      2022-11-30 180925   1938   3349.  578759       0       0
## # ... with abbreviated variable names 1: Province_State, 2: Country_Region,
## #   3: deaths_per_mil, 4: Population, 5: new_cases, 6: new_deaths
```

```
tail(US_totals)
```

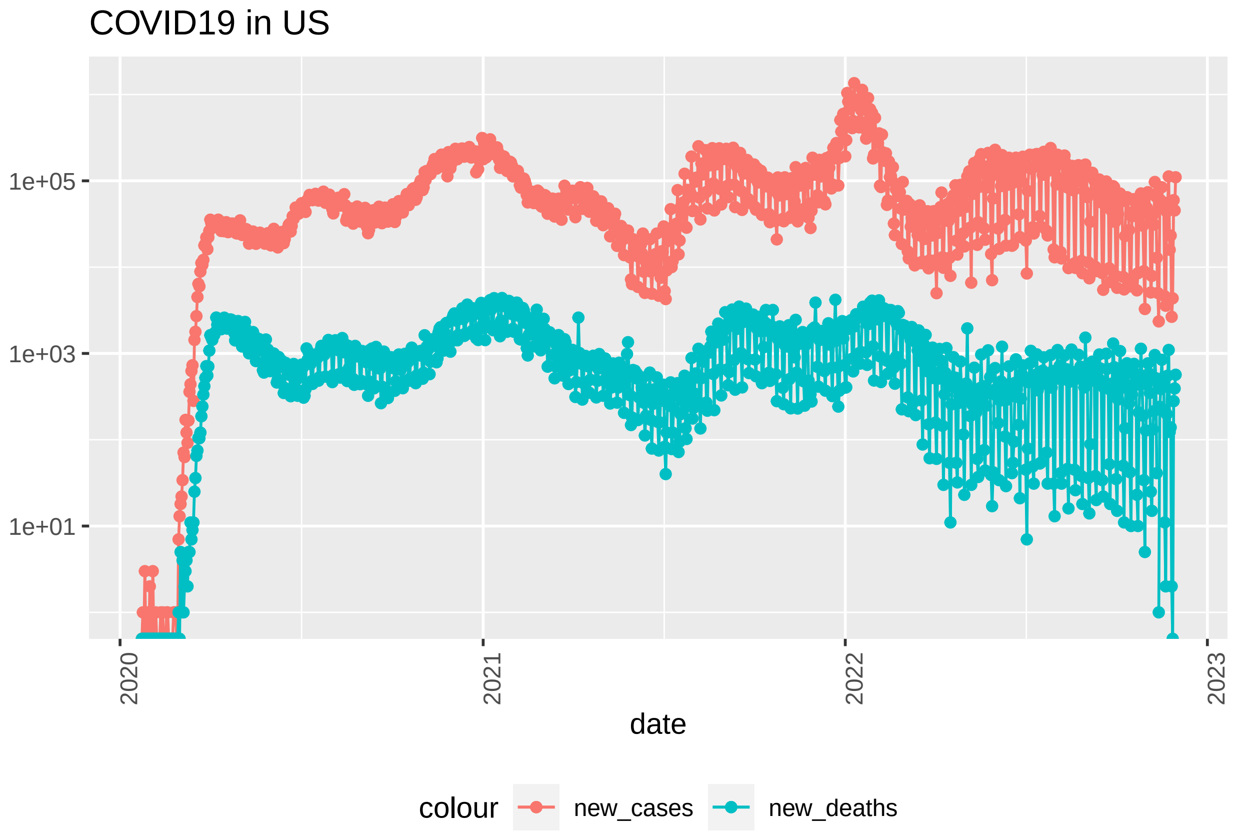
```
## # A tibble: 6 x 8
##   Country_Region date          cases  deaths deaths_pe~1 Popul~2 new_c~3 new_d~4
##   <chr>          <date>        <dbl>   <dbl>       <dbl>   <dbl>   <dbl>   <dbl>
## 1 US             2022-11-25 98566003 1079202       3242.  3.33e8   23226     139
## 2 US             2022-11-26 98568660 1079204       3242.  3.33e8    2657       2
## 3 US             2022-11-27 98573015 1079204       3242.  3.33e8    4355       0
## 4 US             2022-11-28 98632732 1079484       3243.  3.33e8   59717     280
## 5 US             2022-11-29 98678154 1079877       3244.  3.33e8   45422     393
## 6 US             2022-11-30 98788140 1080444       3246.  3.33e8  109986     567
## # ... with abbreviated variable names 1: deaths_per_mil, 2: Population,
## #   3: new_cases, 4: new_deaths
```

```
tail(US_totals %>% select(new_cases,new_deaths,everything()))
```

```
## # A tibble: 6 x 8
##   new_cases new_deaths Country_Region date          cases deaths death~1 Popul~2
##       <dbl>      <dbl> <chr>          <date>        <dbl>  <dbl>   <dbl>   <dbl>
## 1     23226        139 US             2022-11-25 98566003 1.08e6   3242.  3.33e8
## 2      2657          2 US             2022-11-26 98568660 1.08e6   3242.  3.33e8
## 3      4355          0 US             2022-11-27 98573015 1.08e6   3242.  3.33e8
## 4     59717        280 US             2022-11-28 98632732 1.08e6   3243.  3.33e8
## 5     45422        393 US             2022-11-29 98678154 1.08e6   3244.  3.33e8
## 6    109986        567 US             2022-11-30 98788140 1.08e6   3246.  3.33e8
## # ... with abbreviated variable names 1: deaths_per_mil, 2: Population
```

```
#Graph US_total with new_cases and new deaths
US_totals %>%
  ggplot(aes(x = date, y = new_cases)) +
  geom_line(aes(color = "new_cases"))+
  geom_point(aes(color = "new_cases")) +
  geom_line(aes(y = new_deaths, color = "new_deaths")) +
  geom_point(aes(y = new_deaths, color = "new_deaths")) +
  scale_y_log10()+
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 in US", y=NULL)
```

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

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

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

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

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

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

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

## Warning: Removed 3 rows containing missing values (‘geom_point()‘).
```



```
# Find top-ten state smallest deaths in thousand
US_state_totals <- US_by_state %>%
  group_by(Province_State) %>%
  summarize(deaths = max(deaths), cases = max(cases),
            population = max(Population),
            cases_per_thou = 1000 * cases/ population,
            deaths_per_thou = 1000 * deaths / population) %>%
  filter(cases > 0, population > 0)
US_state_totals %>%
  slice_min(deaths_per_thou, n= 10)%>%
  select(deaths_per_thou, cases_per_thou, everything())
```

```
## # A tibble: 10 x 6
##    deaths_per_thou cases_per_thou Province_State           deaths  cases popul~1
##              <dbl>          <dbl> <chr>                     <dbl>  <dbl>   <dbl>
##  1           0.611           149. American Samoa               34 8.26e3   55641
##  2           0.744           240. Northern Mariana Islands     41 1.32e4   55144
##  3           1.17            219. Virgin Islands              125 2.35e4  107268
##  4           1.23           259. Hawaii                     1737 3.67e5 1415872
##  5           1.23           235. Vermont                     770 1.47e5  623989
##  6           1.43           269. Puerto Rico                5367 1.01e6 3754939
##  7           1.59           330. Utah                       5110 1.06e6 3205958
##  8           1.94           244. Washington                14748 1.86e6 7614893
##  9           1.94           406. Alaska                     1436 3.01e5  740995
## 10           1.99           243. District of Columbia       1403 1.71e5  705749
## # ... with abbreviated variable name 1: population
```

```
# Find top-ten largest deaths in thousand
US_state_totals %>%
  slice_max(deaths_per_thou, n=10) %>%
  select(deaths_per_thou, cases_per_thou, everything())
```

```
## # A tibble: 10 x 6
##    deaths_per_thou cases_per_thou Province_State deaths   cases population
##              <dbl>          <dbl> <chr>           <dbl>   <dbl>      <dbl>
##  1            4.38           316. Mississippi     13036  940023    2976149
##  2            4.36           321. Arizona         31751 2337547    7278717
##  3            4.36           308. Oklahoma        17254 1220720    3956971
##  4            4.25           343. West Virginia    7611  614646    1792147
##  5            4.21           316. Alabama         20652 1549285    4903185
##  6            4.16           321. Arkansas        12564  968871    3017804
##  7            4.15           308. New Mexico       8702  646566    2096829
##  8            4.14           350. Tennessee       28305 2389250    6829174
##  9            4.01           294. Michigan        40085 2938443    9986857
## 10            3.95           321. New Jersey      35129 2848609    8882190
```

#Step 4: Modelling

```
#Linear Regression Model
mod <- lm(deaths_per_thou ~ cases_per_thou, data = US_state_totals)
summary(mod)
```

```
## 
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou, data = US_state_totals)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -2.3209 -0.6082  0.1276  0.6679  1.1986 
## 
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)    
## (Intercept)    -0.323647   0.717002  -0.451    0.654    
## cases_per_thou  0.011298   0.002402   4.705 1.81e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.85 on 54 degrees of freedom
## Multiple R-squared:  0.2907, Adjusted R-squared:  0.2776 
## F-statistic: 22.13 on 1 and 54 DF,  p-value: 1.807e-05
```

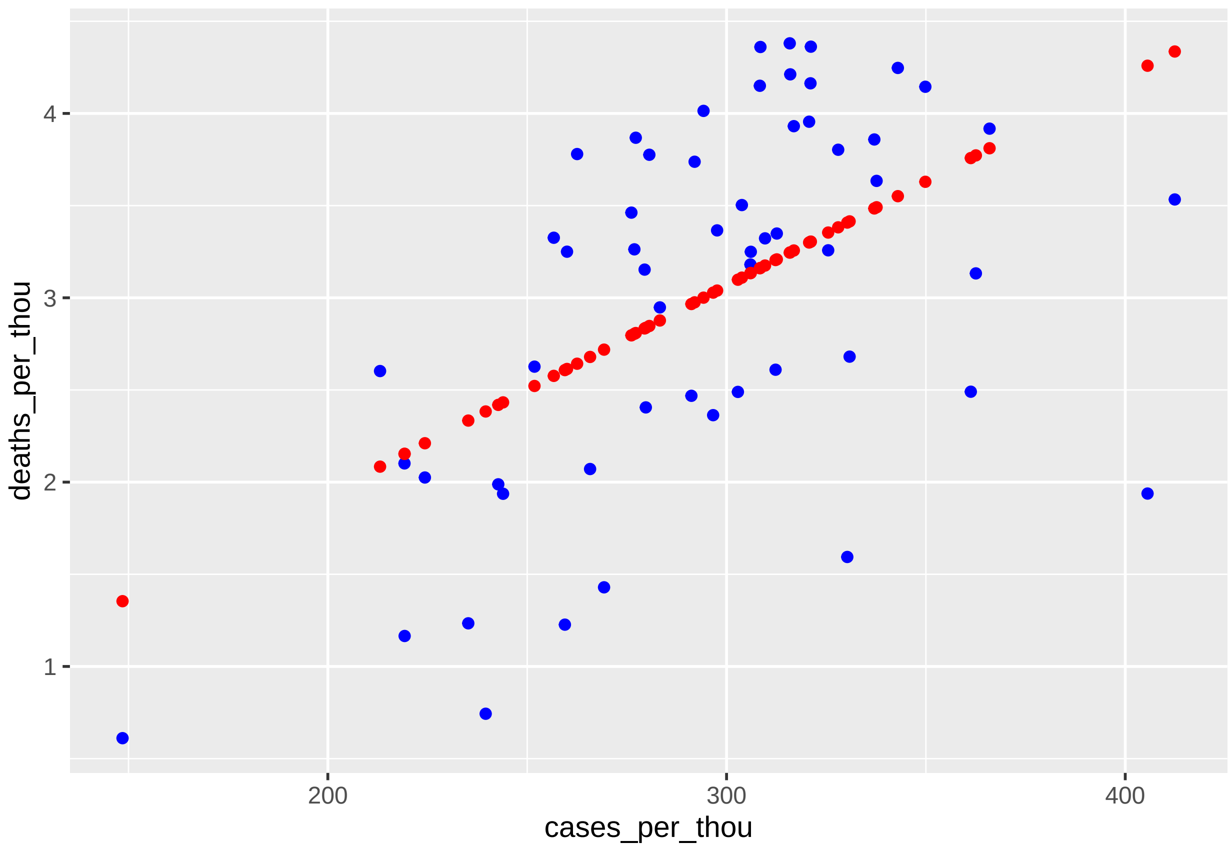
```
x_grid <- seq(1,151)
new_df <- tibble(cases_per_thou = x_grid)
US_state_totals %>% mutate(pred = predict(mod))
```

```
## # A tibble: 56 x 7
##    Province_State       deaths    cases population cases_per_thou deaths~1  pred
##    <chr>                 <dbl>    <dbl>      <dbl>          <dbl>    <dbl> <dbl>
##  1 Alabama               20652  1549285    4903185           316.    4.21   3.25
##  2 Alaska                 1436   300544     740995           406.    1.94   4.26
##  3 American Samoa           34     8263      55641           149.    0.611  1.35
##  4 Arizona               31751  2337547    7278717           321.    4.36   3.30
##  5 Arkansas              12564   968871    3017804           321.    4.16   3.30
##  6 California            97529 11505424   39512223           291.    2.47   2.97
##  7 Colorado              13609  1708264    5758736           297.    2.36   3.03
##  8 Connecticut           11587   926947    3565287           260.    3.25   2.61
##  9 Delaware               3172   316956     973764           325.    3.26   3.35
## 10 District of Columbia   1403   171317     705749           243.    1.99   2.42
## # ... with 46 more rows, and abbreviated variable name 1: deaths_per_thou
```

```
US_tot_w_pred <- US_state_totals %>% mutate(pred = predict(mod))
US_tot_w_pred
```

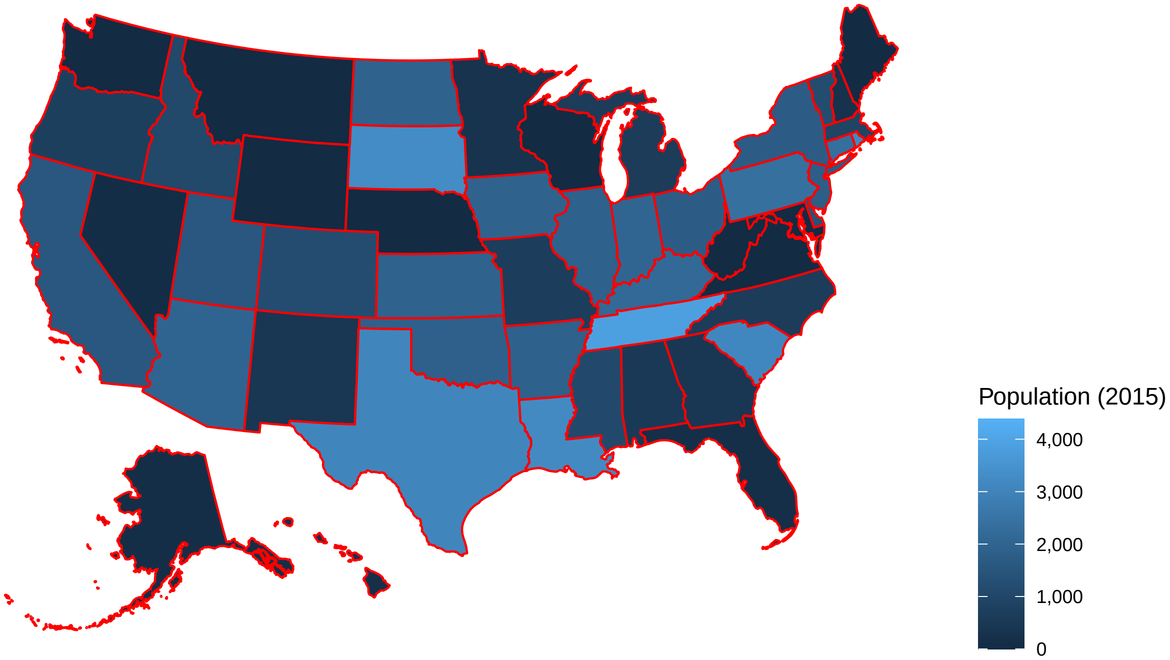
```
## # A tibble: 56 x 7
##    Province_State       deaths    cases population cases_per_thou deaths~1  pred
##    <chr>                 <dbl>    <dbl>      <dbl>          <dbl>    <dbl> <dbl>
##  1 Alabama               20652  1549285    4903185           316.    4.21   3.25
##  2 Alaska                 1436   300544     740995           406.    1.94   4.26
##  3 American Samoa           34     8263      55641           149.    0.611  1.35
##  4 Arizona               31751  2337547    7278717           321.    4.36   3.30
##  5 Arkansas              12564   968871    3017804           321.    4.16   3.30
##  6 California            97529 11505424   39512223           291.    2.47   2.97
##  7 Colorado              13609  1708264    5758736           297.    2.36   3.03
##  8 Connecticut           11587   926947    3565287           260.    3.25   2.61
##  9 Delaware               3172   316956     973764           325.    3.26   3.35
## 10 District of Columbia   1403   171317     705749           243.    1.99   2.42
## # ... with 46 more rows, and abbreviated variable name 1: deaths_per_thou
```

```
# Visualize it
US_tot_w_pred %>% ggplot() +
  geom_point(aes(x = cases_per_thou, y= deaths_per_thou), color = "blue") +
  geom_point(aes(x = cases_per_thou, y = pred), color = "red")
```

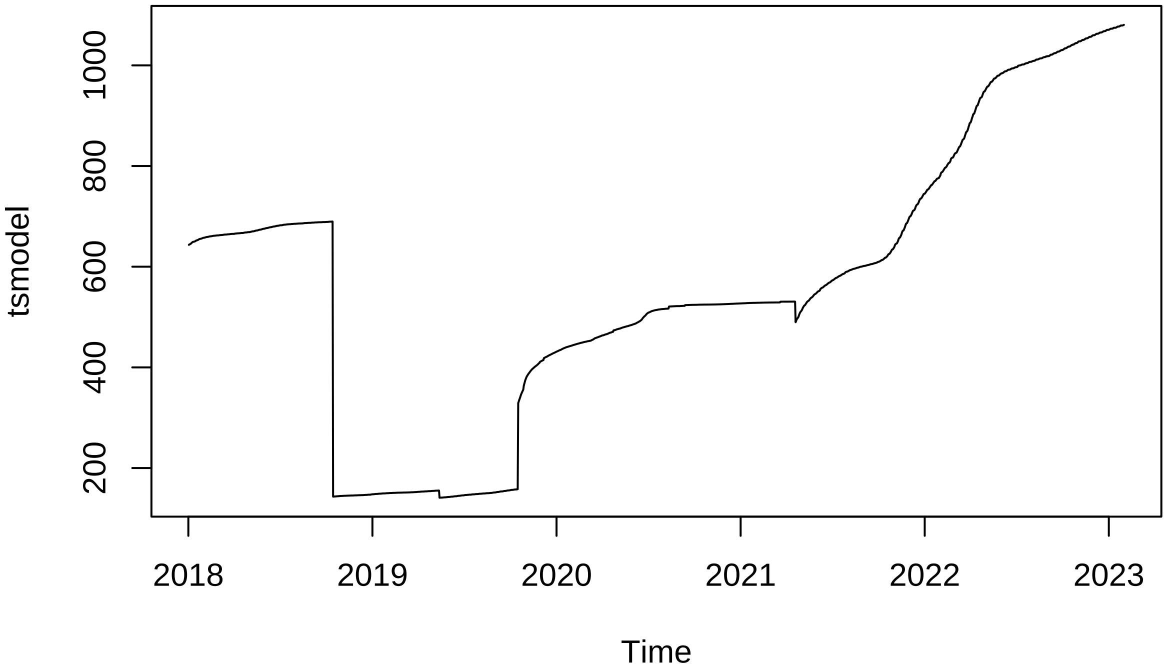


#Step 5: My two unique visualizations and one model

```
#Produce map to present  number of case and deaths.
mapdata <- map_data("world")
view(mapdata)
mapdata <- left_join(mapdata, global, by=c('region'='Country_Region'))
mapcases <- ggplot(mapdata, aes(x = long, y= lat, group=group)) +
  geom_polygon(aes(fill = cases), color = 'black')
mapdeaths <- ggplot(mapdata, aes(x = long, y= lat, group=group)) +
  geom_polygon(aes(fill = deaths), color = 'black')

#Produce map to present deaths_per_mil in US.
colnames(US_by_state)[1] <- "state"
plot_usmap(data = US_by_state, values = "deaths_per_mil", color = "red") +
  scale_fill_continuous(name = "Population (2015)", label = scales::comma) +
  theme(legend.position = "right")
```



```
#Develop the model
global_group <- global %>%
  group_by(date) %>%
  mutate(deaths_per_thou = deaths / 1000) %>%
  select(date,deaths_per_thou)
#develop simple time-series model
tsmodel <- ts(global_group$deaths_per_thou,start = c(2018,2,8),frequency = 365)
#Plot the result
plot(tsmodel)
```


tsmodel
200
400
600
800
1000
2018
2019
2020
2021
2022
2023
Time