COVID data analysis

Akhil Thakur

9/21/2021

COVID Analysis

First we will load the required libraries for our analysis.

```
library(tidyverse)
library(lubridate)
```

Now we load our data directly from the John Hopkins University's github repository. For that we collect all the URL's of the CSV's.

```
url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covi
file_names <- c("confirmed_global.csv",
                    "deaths_global.csv",
                    "confirmed_US.csv",
                    "deaths_US.csv")
urls <- str_c(url_in,file_names)
```

We now read the CSV's corresponding to the global cases, global deaths, US cases and US deaths because of COVID. We store them in the appropriate variables.

```
global_cases <- read_csv(urls[1], show_col_types = FALSE)
global_deaths <- read_csv(urls[2], show_col_types = FALSE)
US_cases <- read_csv(urls[3], show_col_types = FALSE)
US_deaths <- read_csv(urls[4], show_col_types = FALSE)
```

Lets print the data to see the data's description.

```
global_cases
```

```
## # A tibble: 279 x 613
##    `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>              Antigua and Bar~  17.1 -61.8          0         0         0
##  7 <NA>              Argentina        -38.4 -63.6          0         0         0
```

```
##  8 <NA>              Armenia           40.1  45.0          0         0         0
##  9 Australian Capit~ Australia        -35.5 149.           0         0         0
## 10 New South Wales   Australia        -33.9 151.           0         0         0
## # ... with 269 more rows, and 606 more variables: 1/25/20 <dbl>, 1/26/20 <dbl>,
## #   1/27/20 <dbl>, 1/28/20 <dbl>, 1/29/20 <dbl>, 1/30/20 <dbl>, 1/31/20 <dbl>,
## #   2/1/20 <dbl>, 2/2/20 <dbl>, 2/3/20 <dbl>, 2/4/20 <dbl>, 2/5/20 <dbl>,
## #   2/6/20 <dbl>, 2/7/20 <dbl>, 2/8/20 <dbl>, 2/9/20 <dbl>, 2/10/20 <dbl>,
## #   2/11/20 <dbl>, 2/12/20 <dbl>, 2/13/20 <dbl>, 2/14/20 <dbl>, 2/15/20 <dbl>,
## #   2/16/20 <dbl>, 2/17/20 <dbl>, 2/18/20 <dbl>, 2/19/20 <dbl>, 2/20/20 <dbl>,
## #   2/21/20 <dbl>, 2/22/20 <dbl>, 2/23/20 <dbl>, 2/24/20 <dbl>, ...
```

```
global_deaths
```

```
## # A tibble: 279 x 613
##    `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>              Antigua and Bar~  17.1 -61.8          0         0         0
##  7 <NA>              Argentina        -38.4 -63.6          0         0         0
##  8 <NA>              Armenia           40.1  45.0          0         0         0
##  9 Australian Capit~ Australia        -35.5 149.           0         0         0
## 10 New South Wales   Australia        -33.9 151.           0         0         0
## # ... with 269 more rows, and 606 more variables: 1/25/20 <dbl>, 1/26/20 <dbl>,
## #   1/27/20 <dbl>, 1/28/20 <dbl>, 1/29/20 <dbl>, 1/30/20 <dbl>, 1/31/20 <dbl>,
## #   2/1/20 <dbl>, 2/2/20 <dbl>, 2/3/20 <dbl>, 2/4/20 <dbl>, 2/5/20 <dbl>,
## #   2/6/20 <dbl>, 2/7/20 <dbl>, 2/8/20 <dbl>, 2/9/20 <dbl>, 2/10/20 <dbl>,
## #   2/11/20 <dbl>, 2/12/20 <dbl>, 2/13/20 <dbl>, 2/14/20 <dbl>, 2/15/20 <dbl>,
## #   2/16/20 <dbl>, 2/17/20 <dbl>, 2/18/20 <dbl>, 2/19/20 <dbl>, 2/20/20 <dbl>,
## #   2/21/20 <dbl>, 2/22/20 <dbl>, 2/23/20 <dbl>, 2/24/20 <dbl>, ...
```

```
US_cases
```

```
## # A tibble: 3,342 x 620
##         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
## # ... with 3,332 more rows, and 611 more variables: Long_ <dbl>,
## #   Combined_Key <chr>, 1/22/20 <dbl>, 1/23/20 <dbl>, 1/24/20 <dbl>,
## #   1/25/20 <dbl>, 1/26/20 <dbl>, 1/27/20 <dbl>, 1/28/20 <dbl>, 1/29/20 <dbl>,
## #   1/30/20 <dbl>, 1/31/20 <dbl>, 2/1/20 <dbl>, 2/2/20 <dbl>, 2/3/20 <dbl>,
```

```
## #   2/4/20 <dbl>, 2/5/20 <dbl>, 2/6/20 <dbl>, 2/7/20 <dbl>, 2/8/20 <dbl>,
## #   2/9/20 <dbl>, 2/10/20 <dbl>, 2/11/20 <dbl>, 2/12/20 <dbl>, 2/13/20 <dbl>,
## #   2/14/20 <dbl>, 2/15/20 <dbl>, 2/16/20 <dbl>, 2/17/20 <dbl>, ...
```

```
US_deaths
```

```
## # A tibble: 3,342 x 621
##         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
## # ... with 3,332 more rows, and 612 more variables: Long_ <dbl>,
## #   Combined_Key <chr>, Population <dbl>, 1/22/20 <dbl>, 1/23/20 <dbl>,
## #   1/24/20 <dbl>, 1/25/20 <dbl>, 1/26/20 <dbl>, 1/27/20 <dbl>, 1/28/20 <dbl>,
## #   1/29/20 <dbl>, 1/30/20 <dbl>, 1/31/20 <dbl>, 2/1/20 <dbl>, 2/2/20 <dbl>,
## #   2/3/20 <dbl>, 2/4/20 <dbl>, 2/5/20 <dbl>, 2/6/20 <dbl>, 2/7/20 <dbl>,
## #   2/8/20 <dbl>, 2/9/20 <dbl>, 2/10/20 <dbl>, 2/11/20 <dbl>, 2/12/20 <dbl>,
## #   2/13/20 <dbl>, 2/14/20 <dbl>, 2/15/20 <dbl>, 2/16/20 <dbl>, ...
```

Instead of having different datasets for each of those, lets combine the cases and deaths with their own variable names in a separate tibble.

```
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")
```

```
global
```

```
## # A tibble: 169,911 x 5
##    Province_State Country_Region date       cases deaths
##    <chr>          <chr>          <date>     <dbl>  <dbl>
##  1 <NA>           Afghanistan    2020-01-22     0      0
##  2 <NA>           Afghanistan    2020-01-23     0      0
##  3 <NA>           Afghanistan    2020-01-24     0      0
##  4 <NA>           Afghanistan    2020-01-25     0      0
##  5 <NA>           Afghanistan    2020-01-26     0      0
##  6 <NA>           Afghanistan    2020-01-27     0      0
##  7 <NA>           Afghanistan    2020-01-28     0      0
##  8 <NA>           Afghanistan    2020-01-29     0      0
##  9 <NA>           Afghanistan    2020-01-30     0      0
## 10 <NA>           Afghanistan    2020-01-31     0      0
## # ... with 169,901 more rows
```

```
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:169911      Length:169911      Min.   :2020-01-22   Min.   :       0
##  Class :character   Class :character   1st Qu.:2020-06-22   1st Qu.:     146
##  Mode  :character   Mode  :character   Median :2020-11-21   Median :    2318
##                                        Mean   :2020-11-21   Mean   :  288108
##                                        3rd Qu.:2021-04-22   3rd Qu.:   52404
##                                        Max.   :2021-09-21   Max.   :42410607
##      deaths
##  Min.   :     0.0
##  1st Qu.:     1.0
##  Median :    35.0
##  Mean   :  6637.6
##  3rd Qu.:   851.5
##  Max.   :678407.0
```

Lets remove the rows which have zero cases, since they can be outliers / some error in reporting which is not relevant for our current analysis.

```
global <- global %>% filter(cases > 0)
```

Similar to global lets create a tibble for US cases and deaths

```
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_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 <- US_cases %>%
  full_join(US_deaths)
```

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

To facilitate our analysis between countries we join the province and country as a Combined_Key.

```
global <- global %>%
  unite("Combined_Key",
                c(Province_State, Country_Region),
                          sep = ", ",
                na.rm = TRUE,
                remove = FALSE)
```

To have good context over the cases lets gather the population data.

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

```
## Rows: 4196 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.
```

We now dump un-necessary columns not required for our analysis.

```
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: 153,895 x 7
##    Province_State Country_Region date       cases deaths Population Combined_Key
##    <chr>          <chr>          <date>     <dbl>  <dbl>      <dbl> <chr>
##  1 <NA>           Afghanistan    2020-02-24     5      0   38928341 Afghanistan
```

```
##  2 <NA>           Afghanistan    2020-02-25     5      0   38928341 Afghanistan
##  3 <NA>           Afghanistan    2020-02-26     5      0   38928341 Afghanistan
##  4 <NA>           Afghanistan    2020-02-27     5      0   38928341 Afghanistan
##  5 <NA>           Afghanistan    2020-02-28     5      0   38928341 Afghanistan
##  6 <NA>           Afghanistan    2020-02-29     5      0   38928341 Afghanistan
##  7 <NA>           Afghanistan    2020-03-01     5      0   38928341 Afghanistan
##  8 <NA>           Afghanistan    2020-03-02     5      0   38928341 Afghanistan
##  9 <NA>           Afghanistan    2020-03-03     5      0   38928341 Afghanistan
## 10 <NA>           Afghanistan    2020-03-04     5      0   38928341 Afghanistan
## # ... with 153,885 more rows
```

Visualize the data

Lets summarize the cases by Province_State for each day in the US and also add a deaths_per_mill in a new tibble "US_by_state".

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

```
## `summarise()` has grouped output by 'Province_State', 'Country_Region'. You can override using the `
```

```
US_by_state
```

```
## # A tibble: 35,322 x 7
##    Province_State Country_Region date       cases deaths deaths_per_mill
##    <chr>          <chr>          <date>     <dbl>  <dbl>           <dbl>
##  1 Alabama        US             2020-01-22     0      0               0
##  2 Alabama        US             2020-01-23     0      0               0
##  3 Alabama        US             2020-01-24     0      0               0
##  4 Alabama        US             2020-01-25     0      0               0
##  5 Alabama        US             2020-01-26     0      0               0
##  6 Alabama        US             2020-01-27     0      0               0
##  7 Alabama        US             2020-01-28     0      0               0
##  8 Alabama        US             2020-01-29     0      0               0
##  9 Alabama        US             2020-01-30     0      0               0
## 10 Alabama        US             2020-01-31     0      0               0
## # ... with 35,312 more rows, and 1 more variable: Population <dbl>
```

Now lets do the above analysis but for the entire US for each day.

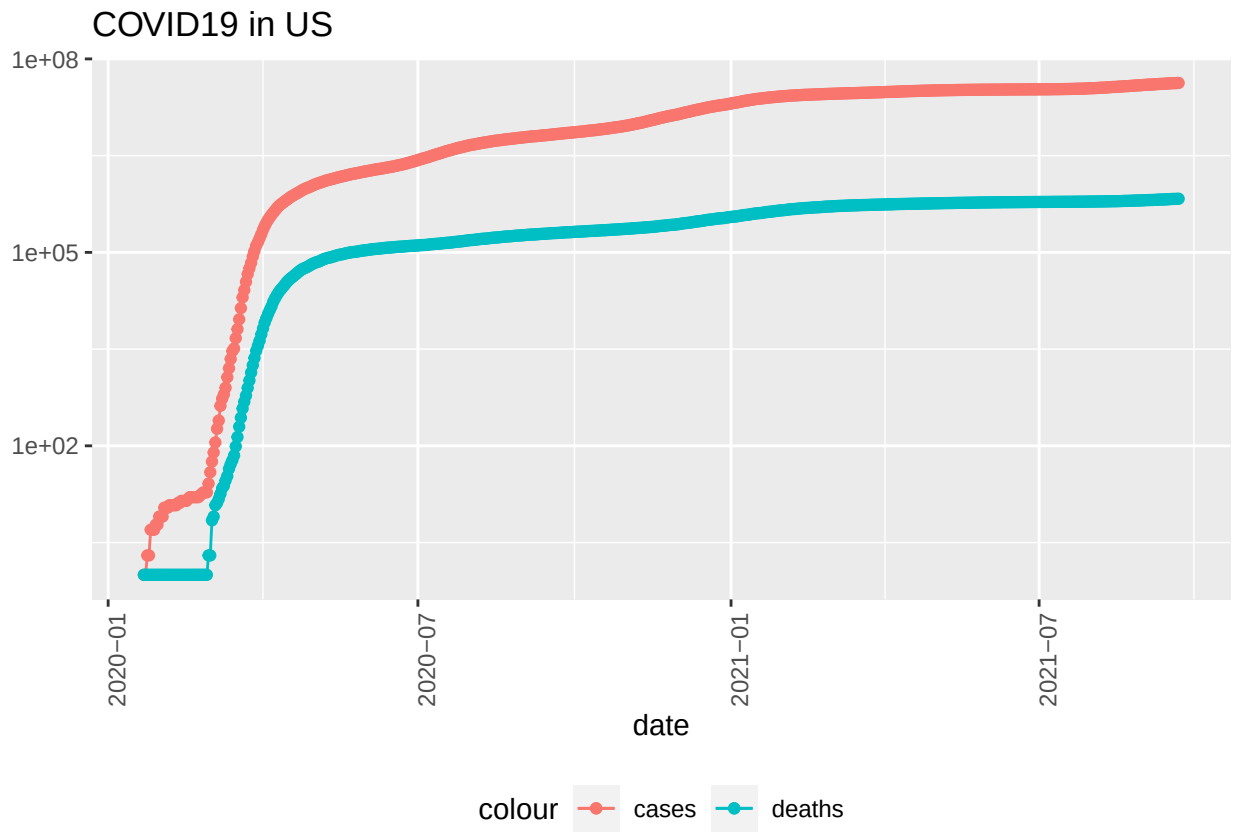
```
US_totals <- US_by_state %>%
  group_by(Country_Region, date) %>%
   summarize(cases = sum(cases), deaths = sum(deaths),
             Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths *1000000 / Population) %>%
```

```
  select(Country_Region, date,
         cases, deaths, deaths_per_mill, Population) %>%
  ungroup()
```

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

Lets Plot the COVID cases and Deaths in the US.

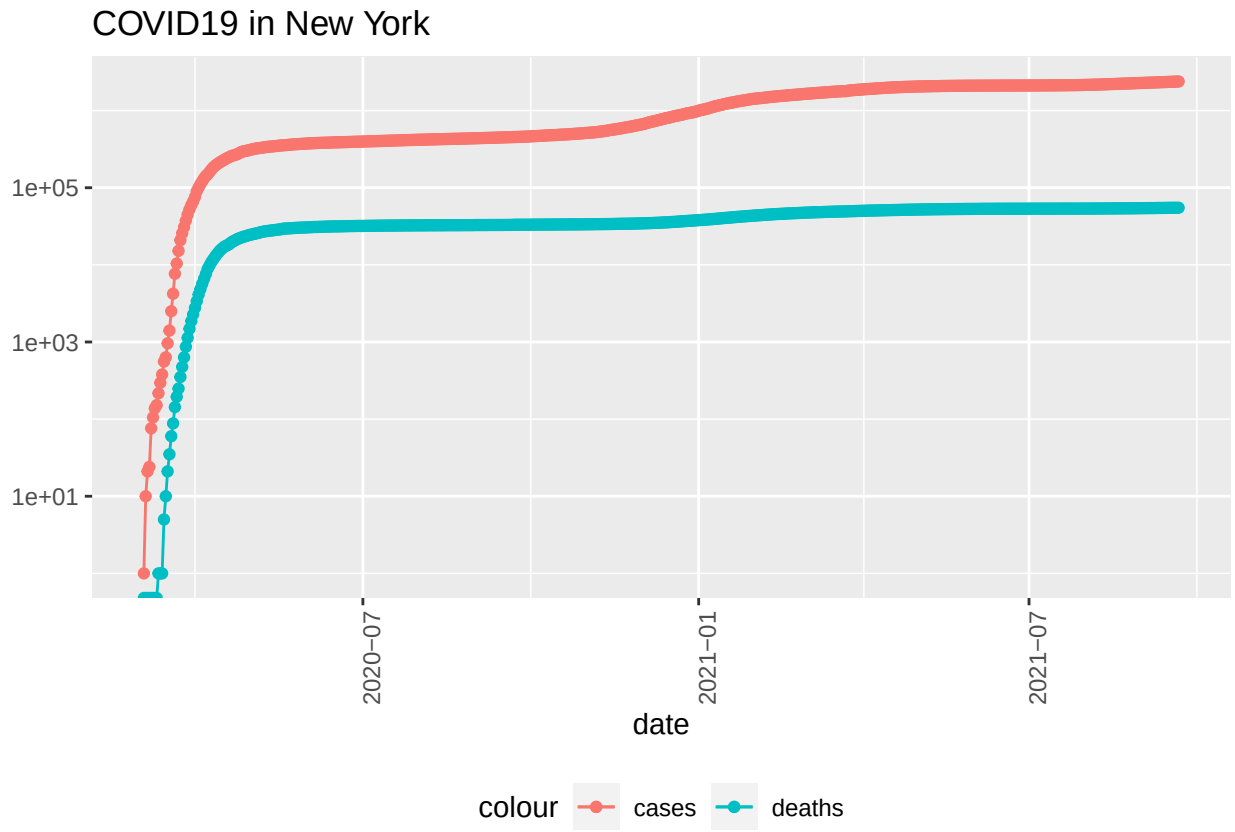
```
US_totals %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = cases)) +
    geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) +
    scale_y_log10() +
    theme(legend.position="bottom",
          axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 in US", y= NULL)
```



As we can see, there is and increasing trend of cases and deaths in US throughout the pandemic. Lets see how the New York is affected by the COVID19 Pandemic. Lets plot the cases and deaths in New York because of COVID19.

```
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("COVID19 in ", state), y= NULL)
```

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

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



Analyze the data

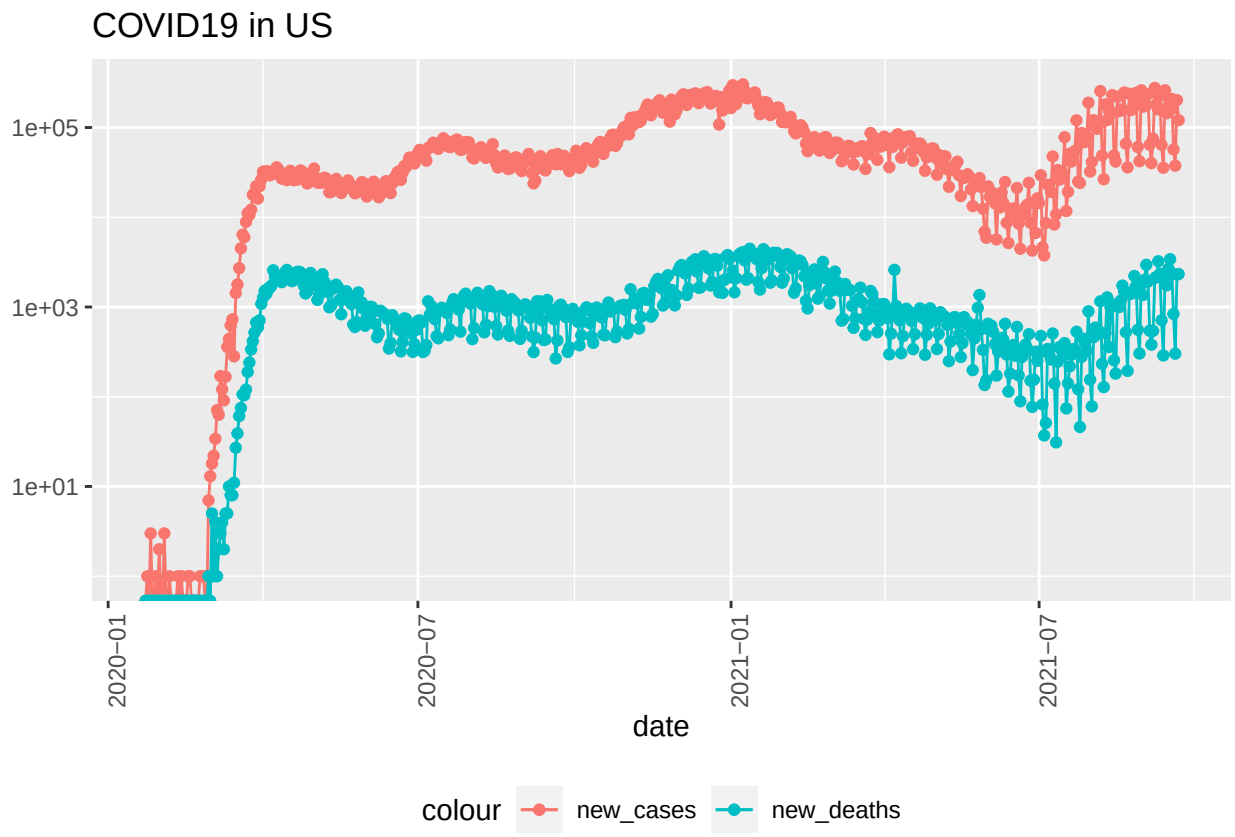
To get deeper understanding of the data, lets calculate and all the new_cases and new_deaths to the US_by_state and US_totals tibble.

```
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))
```

Lets see the trends of new_cases and new_deaths in the US.

```
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

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

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

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

## Warning: Removed 1 row(s) containing missing values (geom_path).

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

## Warning: Removed 1 row(s) containing missing values (geom_path).

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

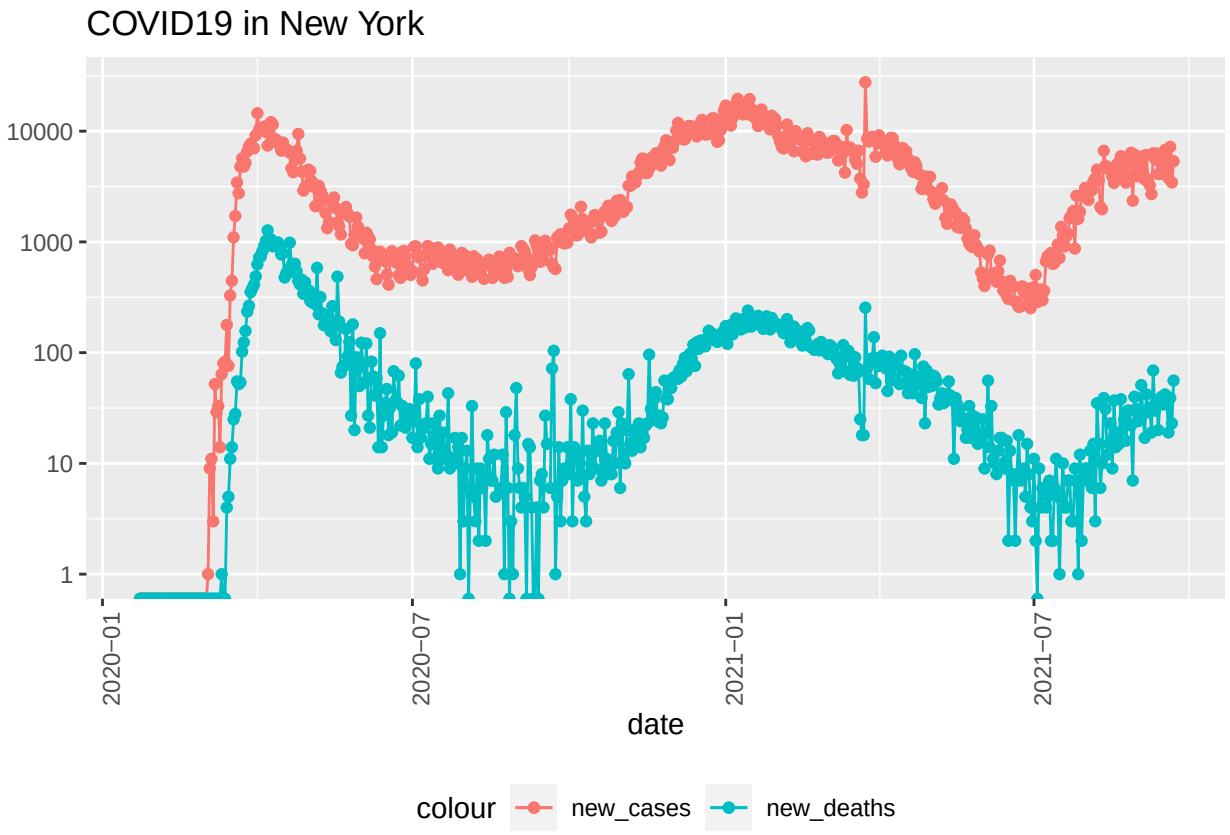


We can see that in some days, we have a spike in new_cases and new_deaths. Lets see the statistics of the state New York with respect to new_cases and new_deaths.

```
state <- "New York"
US_by_state %>%
  filter(Province_State == state) %>%
  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 = str_c("COVID19 in ", state), y= NULL)
```

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

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

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

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

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

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

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

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

## Warning: Removed 1 row(s) containing missing values (geom_path).

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

## Warning: Removed 1 row(s) containing missing values (geom_path).

## Warning: Removed 6 rows containing missing values (geom_point).
```



We can see that even in New York, there are some days with greater spikes in cases and deaths. Now, Lets check how the other states are affected. For that lets calculate the cases_per_thou and deaths_per_thou and add it to the tibble US_state_totals.

```
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)
```

Lets see the top 10 states which have the minimum deaths_per_thou.

```
US_state_totals %>%
  slice_min(deaths_per_thou, n = 10)
```

```
## # A tibble: 10 x 6
##    Province_State           deaths  cases population cases_per_thou deaths_per_thou
##    <chr>                     <dbl>  <dbl>      <dbl>          <dbl>           <dbl>
##  1 Northern Mariana Islands      2    265      55144           4.81          0.0363
##  2 Vermont                     301  31911     623989          51.1           0.482
##  3 Hawaii                      714  76191    1415872          53.8           0.504
##  4 Virgin Islands               68   6516     107268          60.7           0.634
##  5 Alaska                      480 103327     740995         139.            0.648
##  6 Maine                      1002  84542    1344212          62.9           0.745
##  7 Puerto Rico                3092 179523    3754939          47.8           0.823
##  8 Oregon                     3624 314841    4217737          74.6           0.859
##  9 Utah                       2829 495704    3205958         155.            0.882
## 10 Washington                 7315 631023    7614893          82.9           0.961
```

As we can see Northern Mariana Islands have the lowest deaths_per_thou. Lets see which are the top 10 states with respect to deaths per thousand people.

```
US_state_totals %>%
  slice_max(deaths_per_thou, n = 10)
```

```
## # A tibble: 10 x 6
##    Province_State deaths   cases population cases_per_thou deaths_per_thou
##    <chr>           <dbl>   <dbl>      <dbl>          <dbl>           <dbl>
##  1 Mississippi      9331  477769    2976149           161.            3.14
##  2 New Jersey      27240 1137016    8882190           128.            3.07
##  3 Louisiana       13558  730099    4648794           157.            2.92
##  4 New York        54983 2382450   19453561           122.            2.83
##  5 Alabama         13460  775531    4903185           158.            2.75
##  6 Arizona         19584 1070757    7278717           147.            2.69
##  7 Massachusetts   18480  796925    6892503           116.            2.68
##  8 Rhode Island     2816  169686    1059361           160.            2.66
##  9 Arkansas         7499  486853    3017804           161.            2.48
## 10 Florida         51889 3528698   21477737           164.            2.42
```

The state Mississippi has the highest deaths per thousand people followed closely by New Jersey.

Model the data

Lets build a linear model to understand the relation between deaths_per_thou and cases_per_thou. We can use this model to further predict the variables.

```
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
## -1.42361 -0.29946 -0.02553  0.27086  1.16161
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.044428   0.243933   0.182    0.856
## cases_per_thou 0.014536   0.001927   7.543 6.04e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5097 on 53 degrees of freedom
## Multiple R-squared:  0.5177, Adjusted R-squared:  0.5086
## F-statistic:  56.9 on 1 and 53 DF,  p-value: 6.036e-10
```

Lets see which state has the lowest cases per thousand people.

```
US_state_totals %>% slice_min(cases_per_thou)
```

```
## # A tibble: 1 x 6
##   Province_State           deaths cases population cases_per_thou deaths_per_thou
##   <chr>                     <dbl> <dbl>      <dbl>          <dbl>           <dbl>
## 1 Northern Mariana Islands      2   265      55144           4.81          0.0363
```

Same as the deaths per thousand, Northern Mariana Islands have the lowest death per thousand people. Lets calculate which state has the highest cases per thousand people.

```
US_state_totals %>% slice_max(cases_per_thou)
```

```
## # A tibble: 1 x 6
##   Province_State deaths   cases population cases_per_thou deaths_per_thou
##   <chr>           <dbl>   <dbl>      <dbl>          <dbl>           <dbl>
## 1 Tennessee       14441 1190689    6829174           174.            2.11
```

Unlike the above case, where the state with the lowest deaths per thousand and minimum cases per thousand are same, the state Tennessee has the highest cases per thousand people.

Now using our model, lets try to predict the deaths per thousand add it to our tibble.

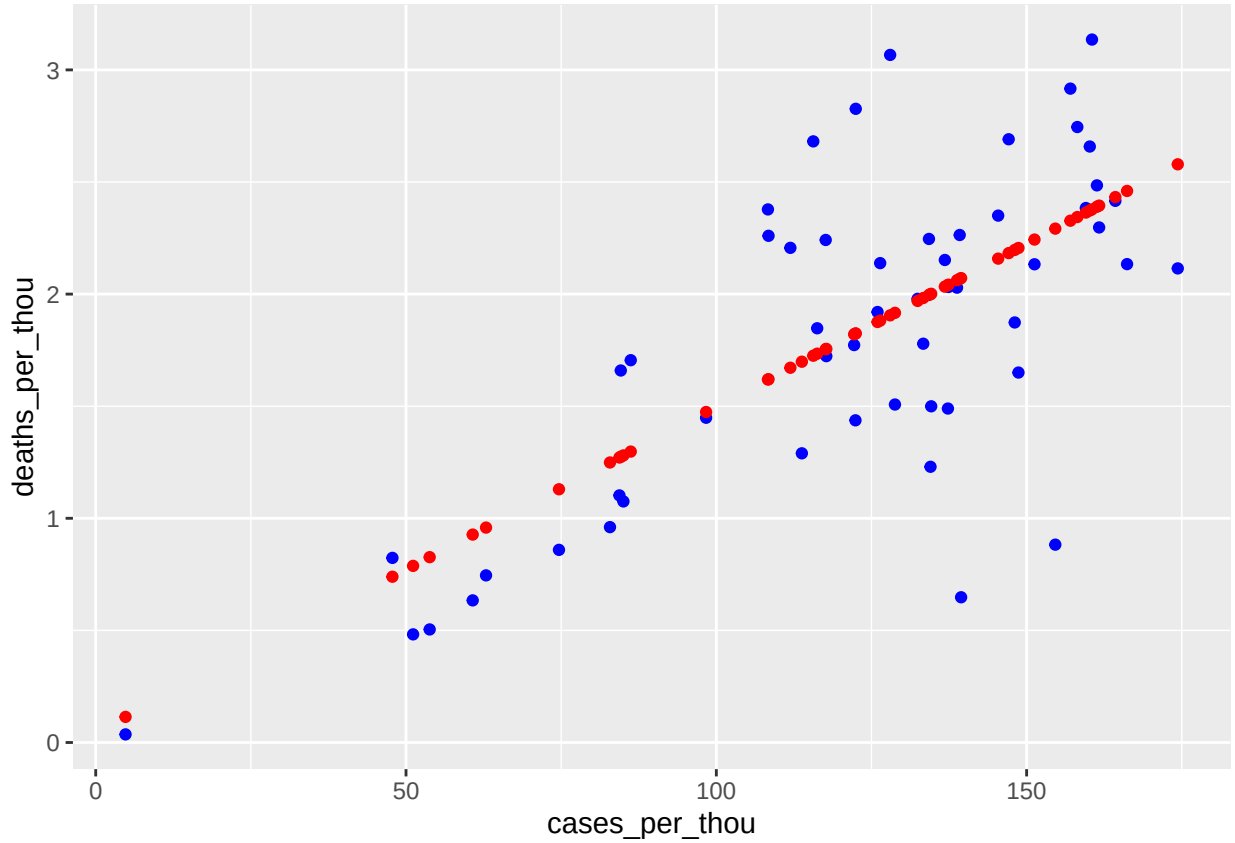
```
x_grid <- seq(1, 151)
new_df <- tibble(cases_per_thou = x_grid)
US_state_totals %>% mutate(pred = predict(mod))
```

```
## # A tibble: 55 x 7
##    Province_State  deaths  cases population cases_per_thou deaths_per_thou  pred
##    <chr>            <dbl>  <dbl>      <dbl>          <dbl>           <dbl> <dbl>
##  1 Alabama          13460 7.76e5    4903185          158.            2.75   2.34
##  2 Alaska             480 1.03e5     740995          139.            0.648  2.07
##  3 Arizona          19584 1.07e6    7278717          147.            2.69   2.18
##  4 Arkansas          7499 4.87e5    3017804          161.            2.48   2.39
```

```
##  5 California       68087 4.65e6   39512223          118.            1.72   1.76
##  6 Colorado          7428 6.55e5    5758736          114.            1.29   1.70
##  7 Connecticut       8477 3.86e5    3565287          108.            2.38   1.62
##  8 Delaware          1927 1.29e5     973764          132.            1.98   1.97
##  9 District of Co~   1171 5.97e4     705749           84.6           1.66   1.27
## 10 Florida          51889 3.53e6   21477737          164.            2.42   2.43
## # ... with 45 more rows
```

Lets plot the predicted value with the actual deaths per thousand people.

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



We can see that our linear model fits snugly into the data and shows an upward trend.

Conclusion

We can clearly observe that there is an increasing trends and signs of new waves of COVID from the data. We have only considered few variables like cases, deaths, cases per thousand , deaths per thousand. There might be various other variables like the availability of medical facilities, density of population, vaccination status, etc which might strongly correlate with the new spikes in the data.

For the specific case of Olympics, Lets see if it had any effect on the COVID cases in Japan.

Trend of COVID cases in Japan during the Olympics

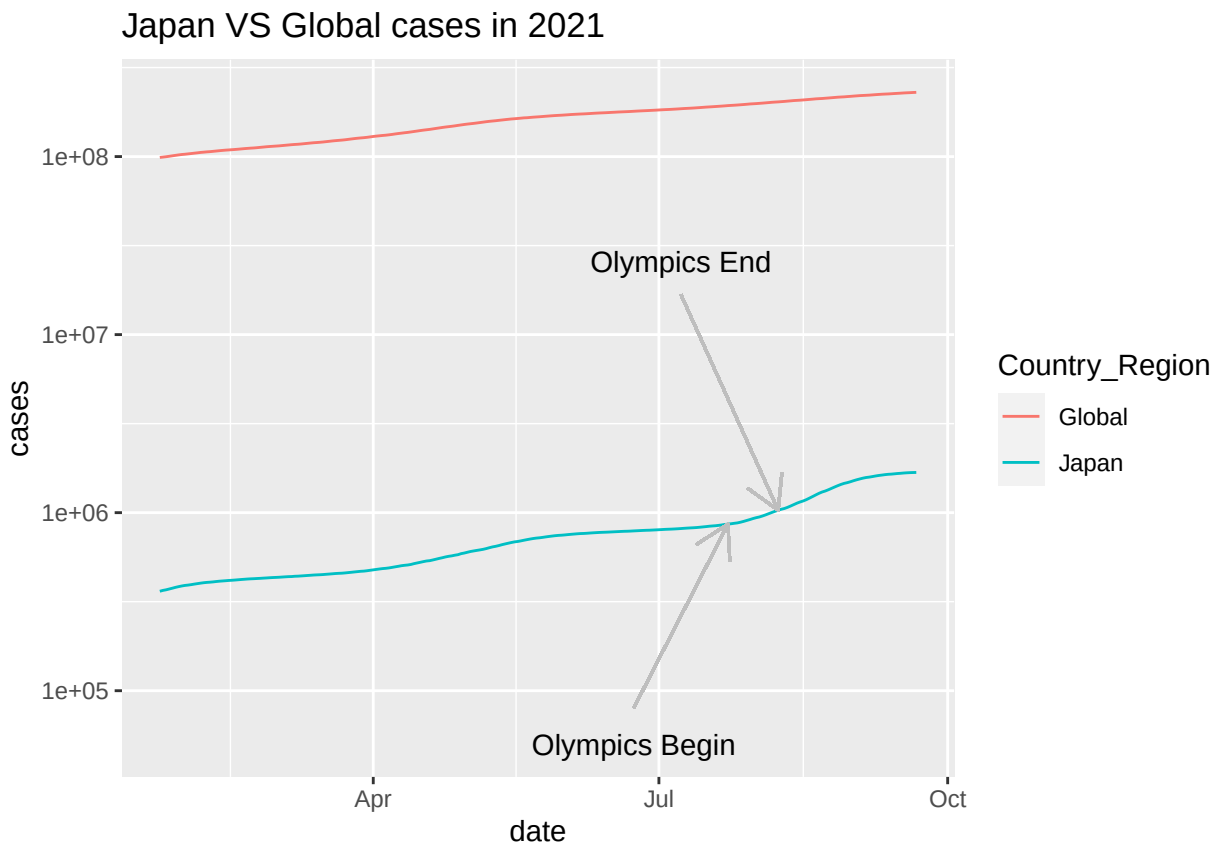
```
JPN_cases <-
  global %>%
  filter(Country_Region == "Japan") %>%
  group_by(date) %>%
  summarize(cases = sum(cases)) %>%
  mutate(Country_Region="Japan") %>%
  ungroup()

GBL_cases <-
  global %>%
  group_by(date) %>%
  summarize(cases = sum(cases)) %>%
  mutate(Country_Region="Global") %>%
  ungroup()
```

We now filter to include only the cases in 2021. Since Olympics are conducted during July-Aug 2021, we can see if there is any inflexion in the curve during this time period.

```
JPN_GBL_cases <- JPN_cases %>%
  full_join(GBL_cases) %>%
  mutate(new_cases = cases - lag(cases))
```

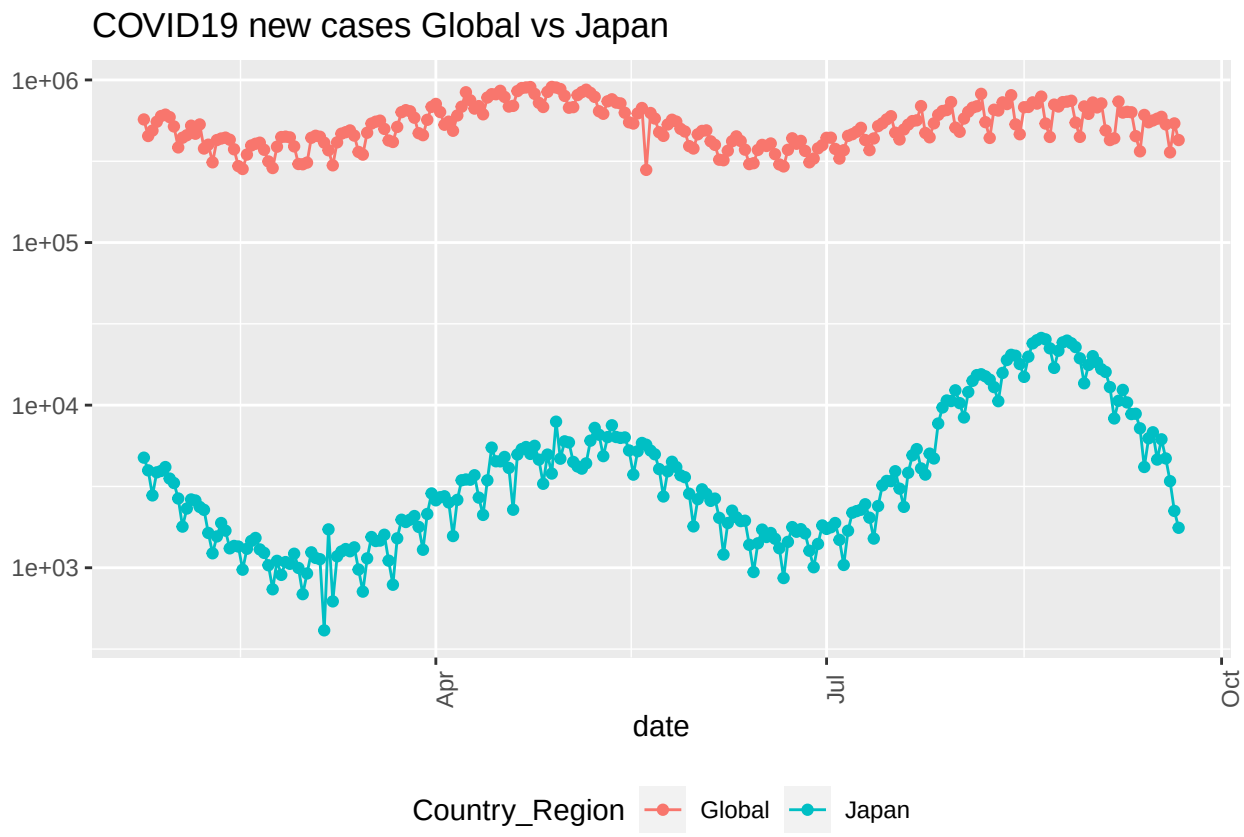
```
## Joining, by = c("date", "cases", "Country_Region")
```

```
JPN_GBL_cases_Olympics <- JPN_GBL_cases %>%
  filter(date>=as.Date("2021-01-23") & date<=as.Date("2021-12-08"))


ggplot(JPN_GBL_cases_Olympics, aes(date, cases, colour=Country_Region)) + geom_line() +
  annotate("text", x=as.Date("2021-06-23"), y=50000, label= "Olympics Begin") +
  annotate("text", x =as.Date("2021-07-08"), y=25803280, label = "Olympics End") +
  geom_segment(aes(x = as.Date("2021-06-23"), xend=as.Date("2021-07-23"),
                   y = 80000, yend=862585), colour="Grey",
               arrow = arrow(length = unit(0.5, "cm")), show.legend = FALSE) +
  geom_segment(aes(x = as.Date("2021-07-08"), xend=as.Date("2021-08-08"),
                   y = 16803280, yend=1032080), colour="Grey",
               arrow = arrow(length = unit(0.5, "cm")), show.legend = FALSE) +
    scale_y_log10() +  labs(title = "Japan VS Global cases in 2021")
```



At first look, we can see that during the Olympics there is a change in the shape of the curve of cases in Japan. Let's do further analysis on the new cases.

```
ggplot(JPN_GBL_cases_Olympics, aes(x = date, y = new_cases, colour=Country_Region)) +
    geom_line() +
  geom_point() +
    scale_y_log10() +
    theme(legend.position="bottom",
          axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 new cases Global vs Japan", y= NULL)
```



We can see that the new cases peaked during the month September. We can infer that the rise in new cases is because of the event in august i.e Olympics. After the Olympics concluded, there is a steady drop in new cases, which supports our point.

Lets Calculate the mean and standard Deviation of new cases before and after the Olympics.

```
JPN_GBL_cases_bfr_Olym <- JPN_GBL_cases %>% filter(date>=as.Date("2021-01-23")
                                                   & date<=as.Date("2021-06-23"))
mean_bfr <- mean(JPN_GBL_cases_bfr_Olym$new_cases)
sd_bfr <- sd(JPN_GBL_cases_bfr_Olym$new_cases)

mean_o <- mean(JPN_GBL_cases_Olympics$new_cases)
sd_o <- sd(JPN_GBL_cases_Olympics$new_cases)

data <- data.frame(Time = c("Before Olympics", "After Olympics"),
                   mean_new_cases = c(mean_bfr, mean_o), sd_new_cases = c(sd_bfr, sd_o))
as_tibble(data)
```

```
## # A tibble: 2 x 3
##   Time            mean_new_cases sd_new_cases
##   <chr>                    <dbl>        <dbl>
## 1 Before Olympics        268986.      292628.
## 2 After Olympics         273803.      290477.
```

Clearly, there is an increase in the mean of new cases because of Olympics and the standard deviation is little less than what it was before.