Covid Data Analysis

Karan Juneja

19/09/2021

We need to analyse the covid data collected by john's hopkins university, and produce some insights on the data.

Loading Files

Source is the https://github.com/CSSEGISandData/COVID-19, Which is COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University. License : Creative Commons Attribution 4.0 International (CC BY 4.0) by the Johns Hopkins University.

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

Reading files

Reading the files using **read_csv**, since files are comma separated.

```
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)
```

Preprocessing

Here we preprocess the data, clean it up converting dates to date formats using lubridate. We also convert the data to a format that we can use to analyze.

```
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")
```

Summary of the data

```
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:169632      Length:169632      Min.   :2020-01-22   Min.   :       0
##  Class :character   Class :character   1st Qu.:2020-06-21   1st Qu.:     146
##  Mode  :character   Mode  :character   Median :2020-11-20   Median :    2307
##                                        Mean   :2020-11-20   Mean   :  287228
##                                        3rd Qu.:2021-04-21   3rd Qu.:   52096
##                                        Max.   :2021-09-20   Max.   :42289819
##      deaths
##  Min.   :     0
##  1st Qu.:     1
##  Median :    35
##  Mean   :  6621
##  3rd Qu.:   846
##  Max.   :676076
```

Removing all the rows on where there was no covid case.

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

Creating a key to join population data of a country with global dataframe.

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

Loading the population data from the github repository.

```
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.
```

Joining both the data frames using left join on "Province_State" and "Country_Region".

```
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: 153,618 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 153,608 more rows
```

Preprocessing United States Data

Here we preprocess the data, clean it up converting dates to date formats using lubridate. We also convert the data to a format that we can use to analyze US data, Also Combining the Deaths data with Cases data and creating a new dataframe called "US".

```
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")
```

```
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 `
```

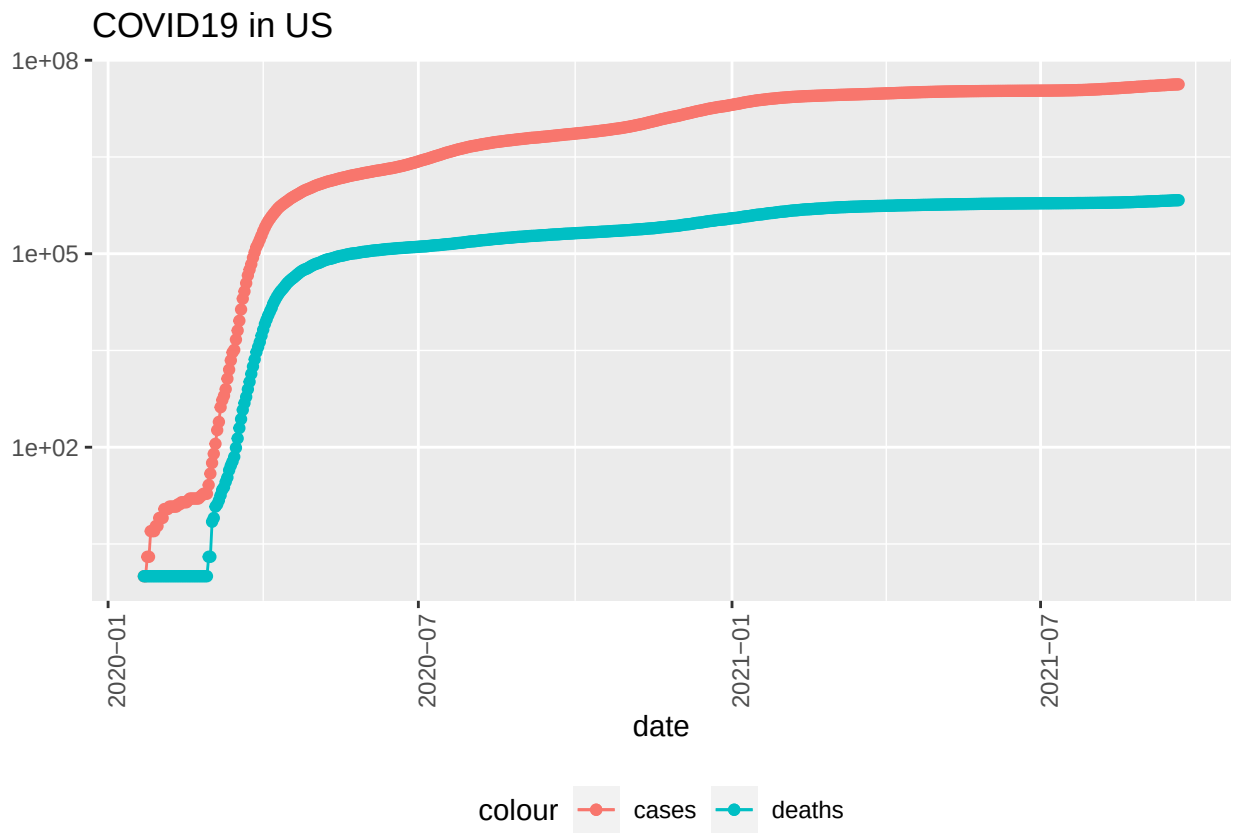
```
US_totals <- US_by_state %>%
  group_by(Country_Region, date) %>%
   summarize(cases = sum(cases), deaths = sum(deaths),
             Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths *1000000 / Population) %>%
  select(Country_Region, date,
         cases, deaths, deaths_per_mill,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

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

PLOTS

The plot below shows deaths and cases in united states from patient zero to today.

```
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)
```

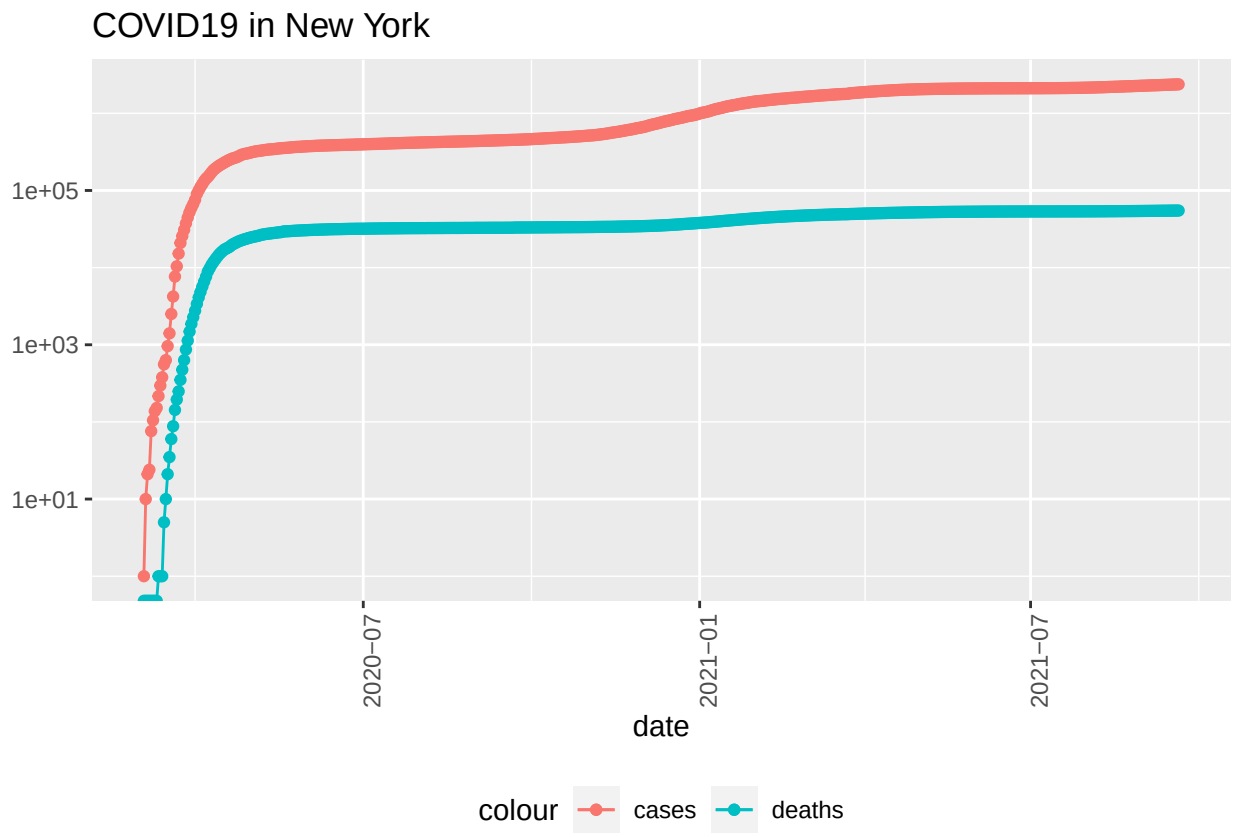


The plot below shows deaths and cases in New york from patient zero to today.

```
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
```



if we see the y axis in the graph above the values peak at 1e+05 which is a huge number and doesn't really tell us much, therefore using the lag() function we will create new columns **new_cases** and **new_deaths** i.e the number of cases per day and the number of deaths per day.

```
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))
```

Now creating a new graph with the columns **new_cases** and **new_deaths**, we can observe the trends in more depth per day.

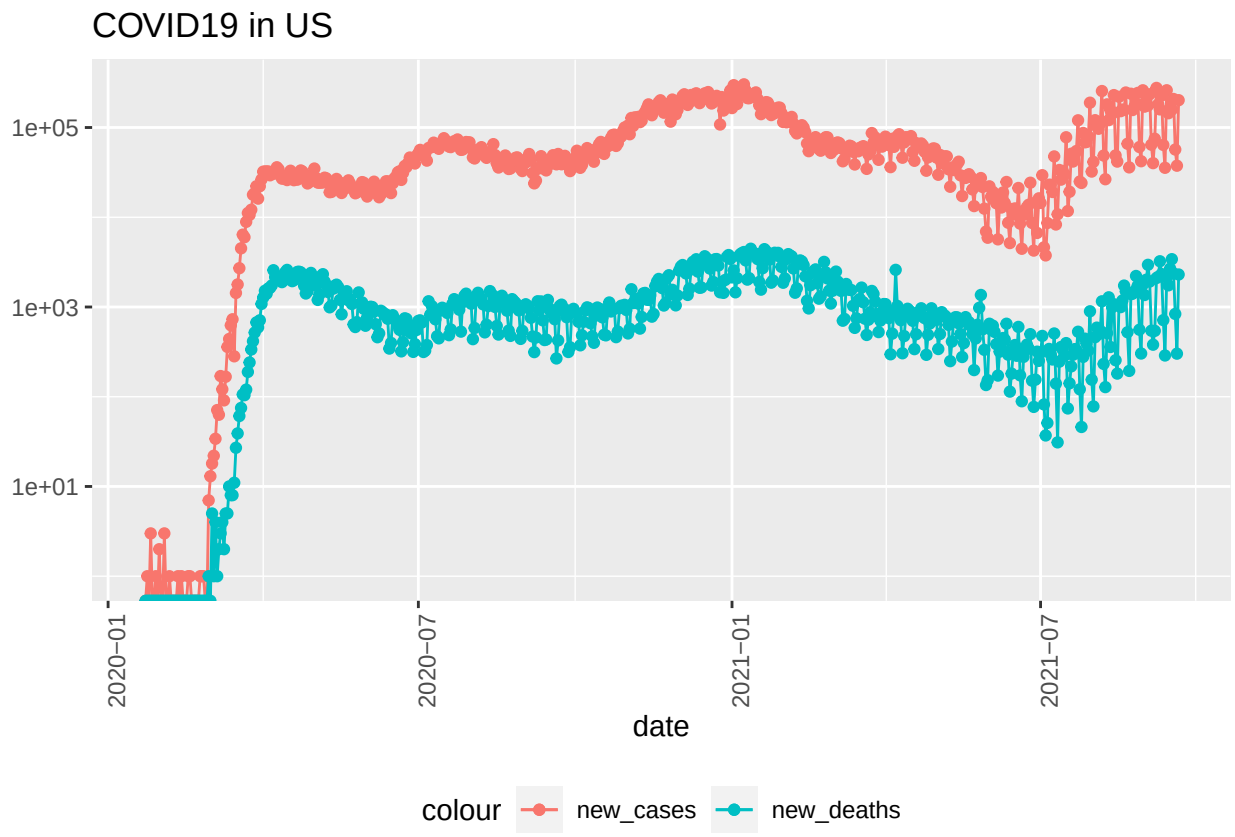
```
US_totals %>%
  ggplot(aes(x = date, y = new_cases)) +
    geom_line(aes(color = "new_cases")) +
  geom_point(aes(color = "new_cases")) +
  geom_line(aes(y = new_deaths, color = "new_deaths")) +
  geom_point(aes(y = new_deaths, color = "new_deaths")) +
    scale_y_log10() +
    theme(legend.position="bottom",
          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).
```

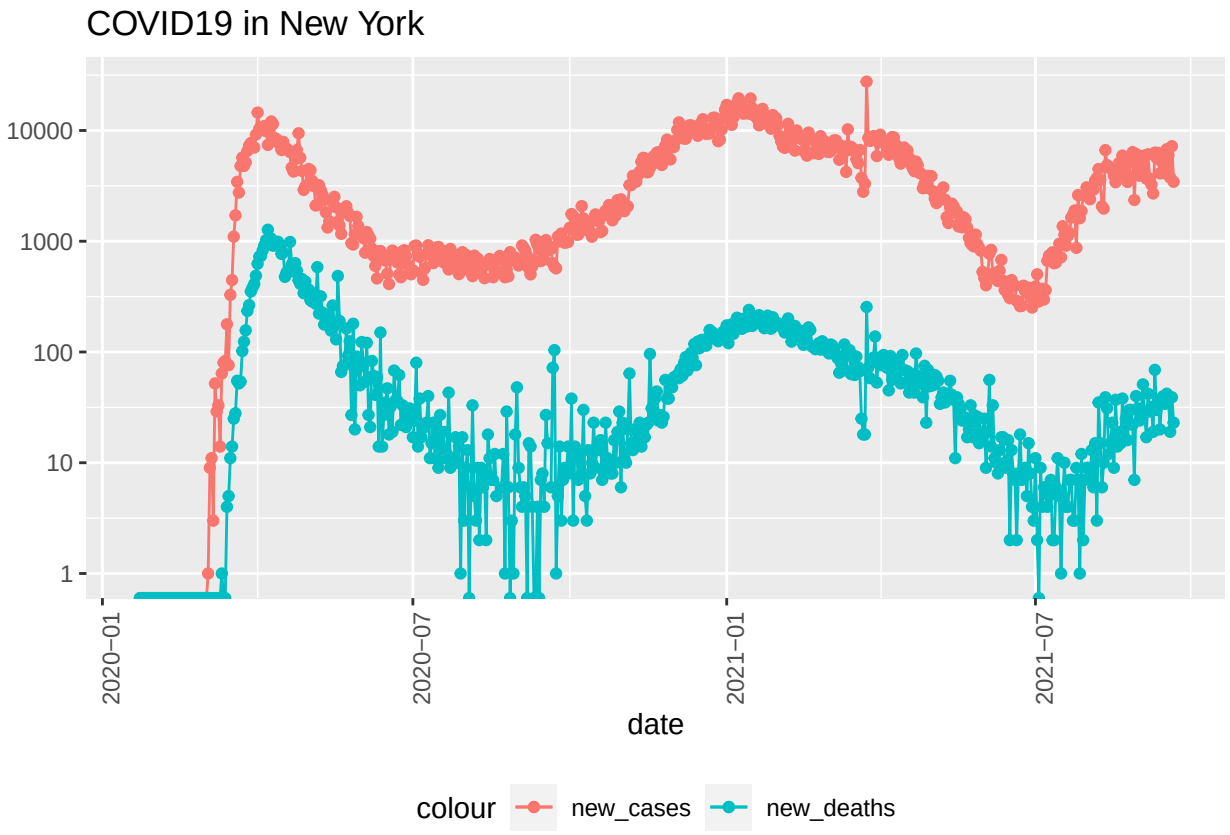


Same graph as above but only for 1 state i.e new york, but still it doesn't show the full story.

```
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).
```



Below we are finding out the best and worst states in terms of deaths and cases per 1000. We use the **slice_min** to find the top 10 best states and **slice_max** to find the top 10 worst states in terms of deaths and cases per 1000 people.

```
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)
```

```
## # 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    263      55144           4.77          0.0363
##  2 Vermont                     298  31764     623989          50.9           0.478
##  3 Hawaii                      714  75911    1415872          53.6           0.504
##  4 Virgin Islands               67   6489     107268          60.5           0.625
##  5 Alaska                      474 102471     740995         138.            0.640
##  6 Maine                       984  83910    1344212          62.4           0.732
##  7 Puerto Rico                3076 179225    3754939          47.7           0.819
##  8 Oregon                     3594 313161    4217737          74.2           0.852
##  9 Utah                       2804 494378    3205958         154.            0.875
## 10 Washington                 7271 628488    7614893          82.5           0.955
```

```
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      9270  476100    2976149           160.            3.11
##  2 New Jersey      27202 1134851    8882190           128.            3.06
##  3 Louisiana       13473  728831    4648794           157.            2.90
##  4 New York        54927 2377102   19453561           122.            2.82
##  5 Alabama         13210  772311    4903185           158.            2.69
##  6 Arizona         19513 1068823    7278717           147.            2.68
##  7 Massachusetts   18455  795543    6892503           115.            2.68
##  8 Rhode Island     2815  169350    1059361           160.            2.66
##  9 Arkansas         7482  485452    3017804           161.            2.48
## 10 Florida         51884 3528698   21477737           164.            2.42
```

So my interest in the data is how do we find when the covid cases in a country have peaked and also has the deaths been peaked?

From the total cases and death plots above we really can't see the actual trends such as the trends which might tell us when the cases are plateauing, details about the what wave are we in?? and which part of the wave are we in etc. So we can use **Moving averages** for these analysis.

Below are the plots made using 7 day **simple moving average** and 30 day *SMA* using the tidyquant package.

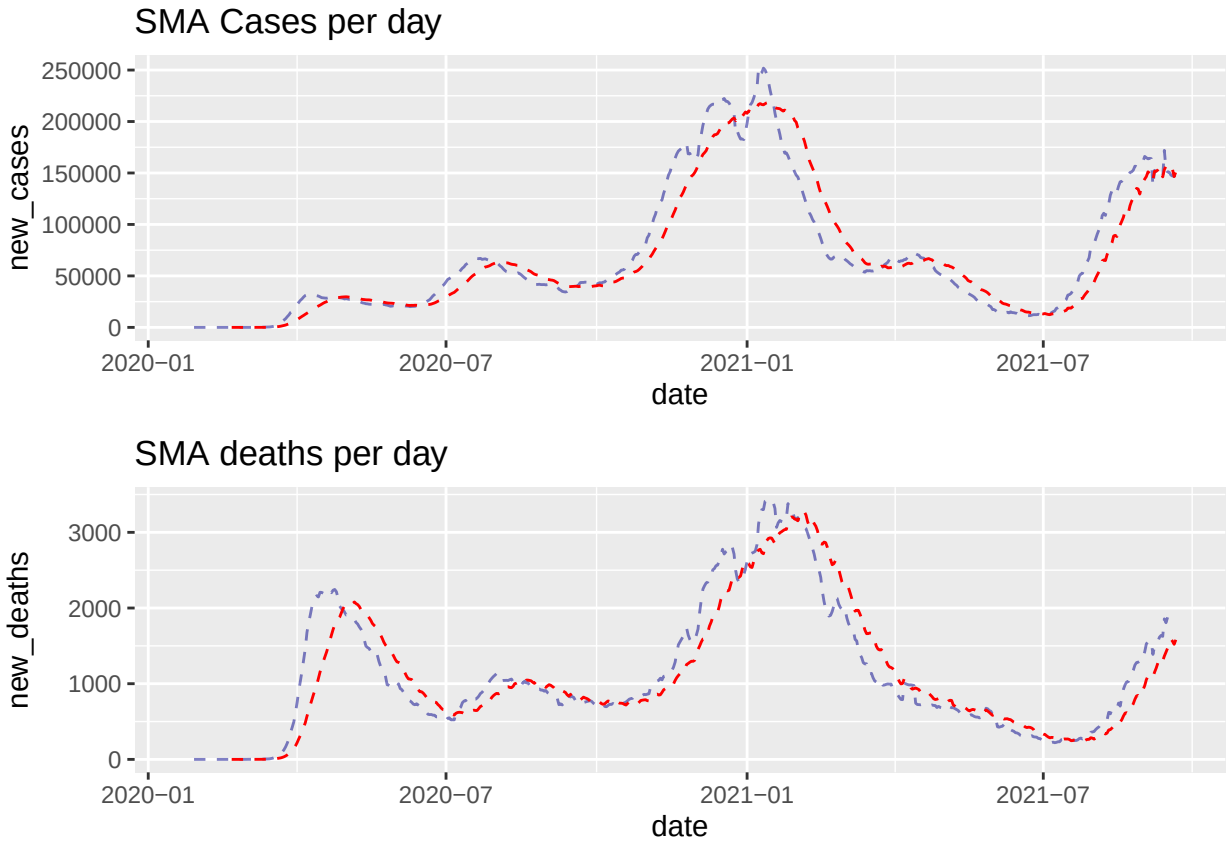
We can see from the graph that in the 2nd wave as soon as the 30 day sma plateaued after that the cases started falling. We can see the same trend happening in the third wave that's ongoing right now, but i think we can confidently deduce that the cases are plateauing and we might see a huge drop in cases soon.

Let's look at the deaths, so in the first wave deaths peaked at 2000 deaths a day, 2nd wave they peaked at 3000 deaths a day, but right now in the ongoing 3rd wave the deaths still haven't plateaued and that's a scary concern. But since the cases have started to plateau and also vaccine rollouts, the deaths might not reach the 2nd wave peak of 3000 deaths a day.

```
cases <- US_totals %>%
  ggplot(aes(x = date, y = new_cases)) +
```

```
   geom_ma(ma_fun = SMA, n = 7,alpha = 0.5) +
  geom_ma(ma_fun = SMA, n = 30, color = "red")+
  labs(title = "SMA Cases per day")
deaths <- US_totals %>%
  ggplot(aes(x = date, y = new_deaths)) +
  geom_ma(ma_fun = SMA, n = 7,alpha = 0.5) +
  geom_ma(ma_fun = SMA, n = 30, color = "red")+
  labs(title = "SMA deaths per day")

plot_grid(cases,deaths,ncol=1,align='v')
```



The state of alabama shows almost the same pattern as the whole of US but, we can see that the cases have started to drop and the 7 day sma kind of proves that as well as 30 day sma concludes the findings.

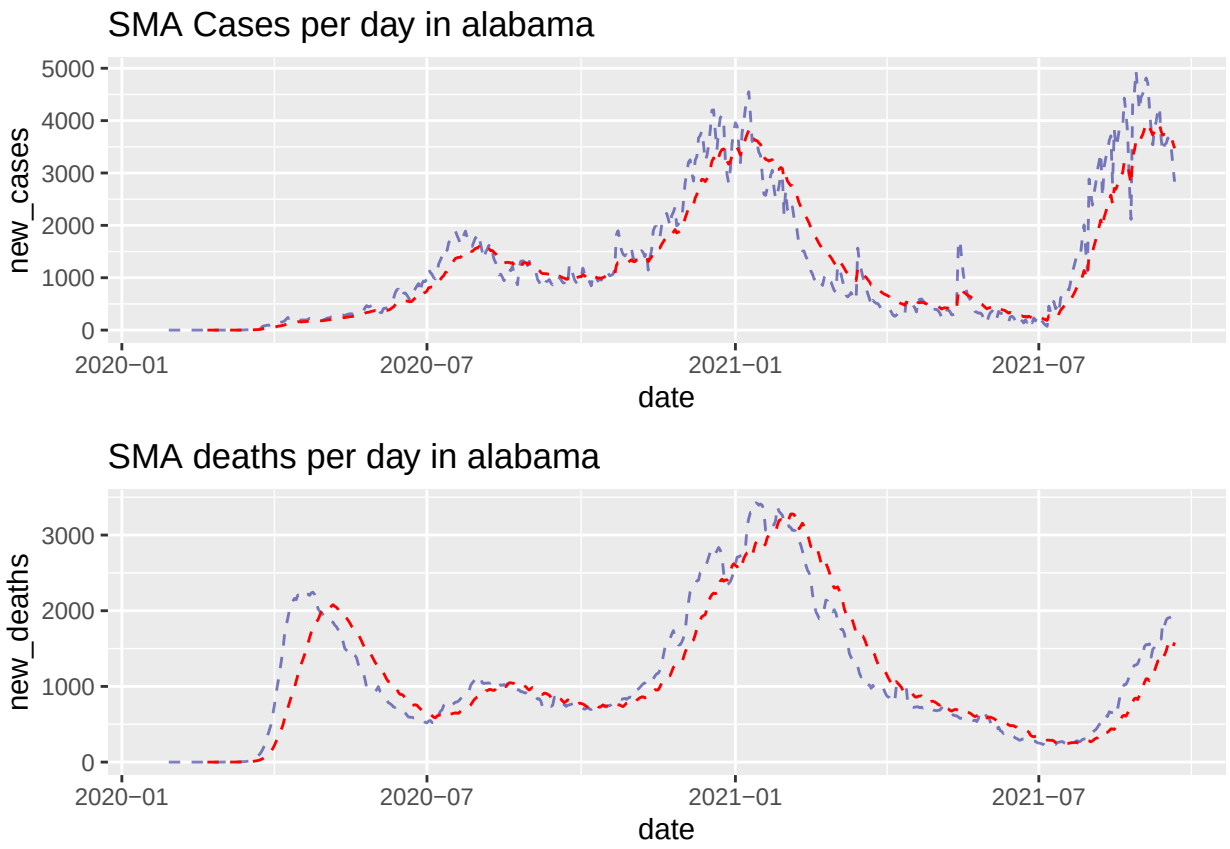
But the deaths still are rising in alabama too its a scary thing.

```
state <- "Alabama"
cases <- US_by_state %>%
  filter(Province_State == state) %>%
  ggplot(aes(x = date, y = new_cases)) +
  geom_ma(ma_fun = EMA, n = 7,alpha=0.5) +                    # Plot 30-day EMA
  geom_ma(ma_fun = EMA, n = 30, color = "red")+
  labs(title = "SMA Cases per day in alabama")

deaths <- US_totals %>%
  ggplot(aes(x = date, y = new_deaths)) +
  geom_ma(ma_fun = SMA, n = 7,alpha = 0.5) +
```

```
  geom_ma(ma_fun = SMA, n = 30, color = "red")+
  labs(title = "SMA deaths per day in alabama")

plot_grid(cases,deaths,ncol=1,align='v')
```



Model

Lets create a linear model with target as *death_per_thou* and predictor as *cases_per_thou*, what we mean by creating the model is that given the *cases_per_thou* can we predict the *death_per_thou*, using a linear regression model.

```
mod <- lm(deaths_per_thou ~ cases_per_thou, data = US_state_totals)
summary(mod)
```

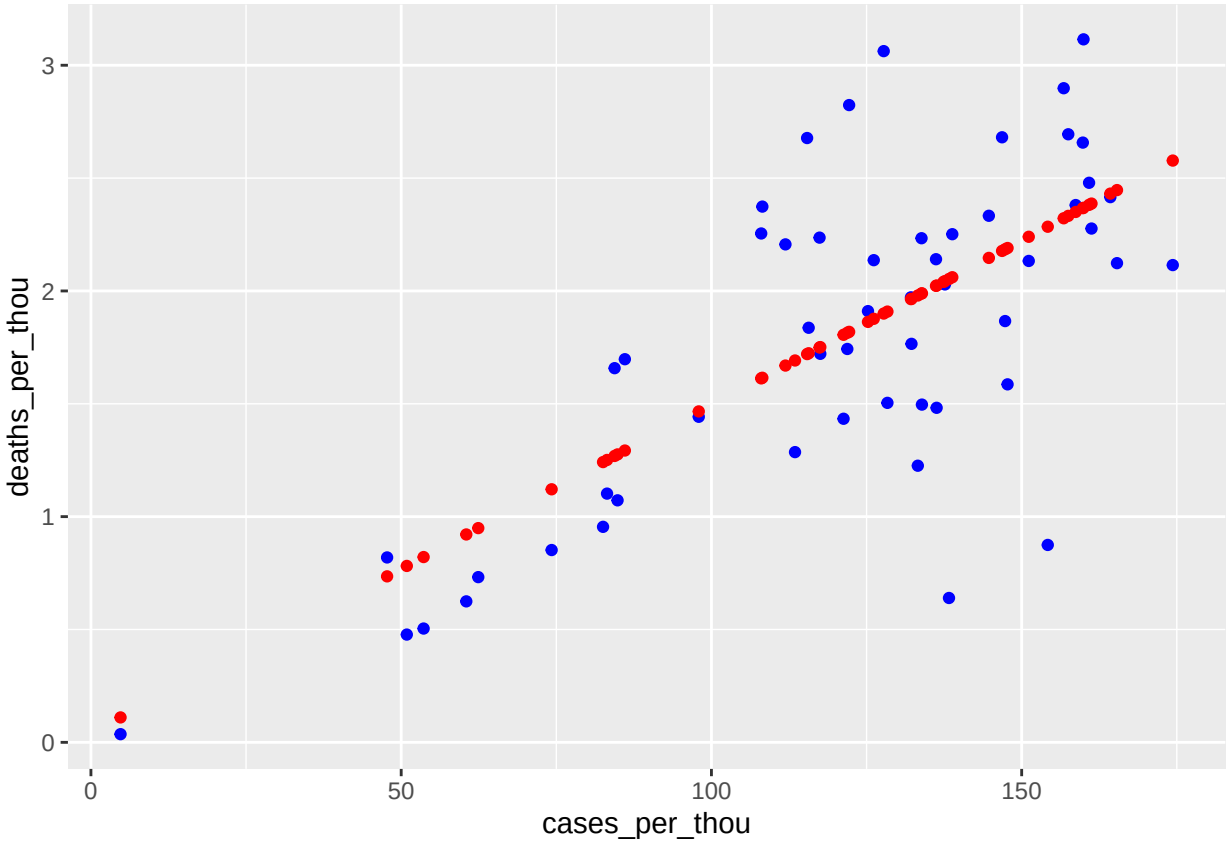
```
##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou, data = US_state_totals)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.41327 -0.30027 -0.01562  0.27513  1.16265
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.040990   0.242819   0.169    0.867
## cases_per_thou 0.014549   0.001925   7.557 5.74e-10 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5079 on 53 degrees of freedom
## Multiple R-squared:  0.5186, Adjusted R-squared:  0.5095
## F-statistic:  57.1 on 1 and 53 DF,  p-value: 5.741e-10
```

```
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          13210 7.72e5    4903185          158.            2.69   2.33
##  2 Alaska             474 1.02e5     740995          138.            0.640  2.05
##  3 Arizona          19513 1.07e6    7278717          147.            2.68   2.18
##  4 Arkansas          7482 4.85e5    3017804          161.            2.48   2.38
##  5 California       68019 4.64e6   39512223          118.            1.72   1.75
##  6 Colorado          7405 6.53e5    5758736          113.            1.29   1.69
##  7 Connecticut       8463 3.86e5    3565287          108.            2.37   1.62
##  8 Delaware          1920 1.29e5     973764          132.            1.97   1.96
##  9 District of Co~   1170 5.96e4     705749           84.4           1.66   1.27
## 10 Florida          51884 3.53e6   21477737          164.            2.42   2.43
## # ... with 45 more rows
```

```
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")
```



Bias

- Dataset Bias
  - How the cases are counted?,if a person takes 3 tests and comes positive 3 times is that also counted as a single case or a multiple cases?
- Personal Bias
  - So my bias was that due to vaccinations,there would be less number of cases.But the cases in the third wave seem to be peaking at the same level as the 2nd wave when there were no vaccinations.

Conclusion

Covid is not gone, even with the rigorous vaccine rollouts we can see that the number of cases per day in the 3rd wave seem to have peaked out at the same level as the number of cases per day during the 2nd Wave.

But deaths haven't peaked out, yes at the same time in the 2nd wave at the same number of cases there were more deaths then now (due to vaccinations?).But the deaths Moving average graph hasn't peaked yet so time will tell how many lives will be lost due to this pandemic.But looking at the countries such as the UK the third wave even if it had as many cases per day as the 2nd wave, but it still had way fewer deaths then in the 2nd wave.So lets hope for the best and Mask Up.