Covid_19_Analysis

Swapnil_Sethi

9/21/2021

Note: I am using tidyverse, and lubridate libraries for my analysis. If you have not installed these libraries earlier, then install them and then run this report.

INDEX

libraries
SessionInfo
Question of Interest Data Collection
Read Covid Data
Global Covid Data Transformation
Read and Add Global Population Data
US Covid Data Transformation
add US states area data *(New)*
Visualize Global Data *(New, Contains 2 visuals)*
Visualize US Data
Outliers in pop_density data *(new, determine outliers in population density)*
Analyze the data
US States Covid analysis with population density *(New, contains 1 visual with liner regression model)*
Model the data [Biases]
Conclusion

libraries

```
library(tidyverse)
```

```
FALSE -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
FALSE v ggplot2 3.3.5     v purrr   0.3.4
FALSE v tibble  3.1.3     v dplyr   1.0.7
FALSE v tidyr   1.1.3     v stringr 1.4.0
FALSE v readr   2.0.1     v forcats 0.5.1
```

```
FALSE -- Conflicts ------------------------------------------ tidyverse_conflicts() --
FALSE x dplyr::filter() masks stats::filter()
FALSE x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
FALSE
FALSE Attaching package: 'lubridate'
```

```
FALSE The following objects are masked from 'package:base':
FALSE
FALSE     date, intersect, setdiff, union
```

SessionInfo

See this session info for more insights on the packages I am using. If you are not able to knit the report then you might consider updating your packages to the below versions.

```
sessionInfo()
```

```
## R version 4.1.1 (2021-08-10)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Big Sur 11.5.1
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1-arm64/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.7.10 forcats_0.5.1    stringr_1.4.0    dplyr_1.0.7
##  [5] purrr_0.3.4      readr_2.0.1      tidyr_1.1.3      tibble_3.1.3
##  [9] ggplot2_3.3.5    tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.1  xfun_0.25         haven_2.4.3       colorspace_2.0-2
##  [5] vctrs_0.3.8       generics_0.1.0    htmltools_0.5.1.1 yaml_2.2.1
##  [9] utf8_1.2.2        rlang_0.4.11      pillar_1.6.2      glue_1.4.2
## [13] withr_2.4.2       DBI_1.1.1         dbplyr_2.1.1      modelr_0.1.8
## [17] readxl_1.3.1      lifecycle_1.0.0   munsell_0.5.0     gtable_0.3.0
## [21] cellranger_1.1.0  rvest_1.0.1       evaluate_0.14     knitr_1.33
## [25] tzdb_0.1.2        fansi_0.5.0       broom_0.7.9       Rcpp_1.0.7
## [29] scales_1.1.1      backports_1.2.1   jsonlite_1.7.2    fs_1.5.0
## [33] hms_1.1.0         digest_0.6.27     stringi_1.7.3     grid_4.1.1
## [37] cli_3.0.1         tools_4.1.1       magrittr_2.0.1    crayon_1.4.1
## [41] pkgconfig_2.0.3   ellipsis_0.3.2    xml2_1.3.2        reprex_2.0.1
## [45] assertthat_0.2.1  rmarkdown_2.10    httr_1.4.2        rstudioapi_0.13
## [49] R6_2.5.1          compiler_4.1.1
```

Question of Interest

Covid-19 has affected people's lives in many ways all over the world. Today, through this analysis we will try to understand how Covid-19 has spread over time in different countries and will analyze the spread of Covid-19 in the United Kingdom. We will also go in depth to understand how it's spread in the different US States.

Data Collection

As you know, to do any analysis first we need to gather data. John Hopkins University has collected Covid-19 data from all over the world and published it in the GitHub repository for public use. We will use the same data for our analysis.

Let's connect to GitHub repository

```
## Get current Data from the four files.
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)
```

Read Covid data

Read the data and let's took quick look at it.

```
global_cases <- read_csv(urls[1], show_col_types = FALSE) #read Global cases data
global_deaths <- read_csv(urls[2], show_col_types = FALSE) #read Global deaths data
US_cases <- read_csv(urls[3], show_col_types = FALSE) #read US cases data
US_deaths <- read_csv(urls[4], show_col_types = FALSE) #read US deaths data
```

Global Covid Data Transformation

After looking at global_cases and global_deaths, I would like to tidy those datasets and put each variable (date, cases, deaths) in its own column. Also, I don't need Lat and Long for the analysis I am planning, so I will get rid of those and rename Region and State to be more R friendly.

```
global_cases <- global_cases %>%
    pivot_longer(cols = -c(`Province/State`,
                           `Country/Region`, Lat, Long),
                 names_to = "date",
                values_to = "cases") %>%      #pivot date and cases columns
  select(-c(Lat,Long))                    #remove lat and long columns

global_deaths <- global_deaths %>%
    pivot_longer(cols = -c(`Province/State`,
                           `Country/Region`, Lat, Long),
                 names_to = "date",
                values_to = "deaths") %>%      #pivot date and death columns
  select(-c(Lat, Long))                #remove lat and long

global <- global_cases %>%
  full_join(global_deaths) %>%  #combine both global cases and global deaths in a single dataframe glob
  rename(Country_Region = `Country/Region`,
         Province_State = `Province/State`) %>%    #rename columns
  mutate(date = mdy(date))    #change datatype of date column to date.
```

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

Now, let's take look at a summary of the data to see if there are problems

```
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
```

Everything looks good, except rows having min cases = 0

I don't need rows with cases = 0 for my analysis, so I will get rid of rows with no cases

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

Read and Add Global Population Data

We notice that we don't have population data for the world data. If we plan to do a comparative analysis between countries, we will want to add the population data to our global dataset.

Let's add population data and a variable called Combined_Key that combines the Province_State with the Country_Region

```
global <- global %>%
  unite("Combined_Key",
                c(Province_State, Country_Region),
                                sep = ", ",
                na.rm = TRUE,
                remove = FALSE)  #create a key column keeping original columns as it is.
```

First read 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)) #remove unnecessary columns
```

Add this population data to the global dataset.

```
global <- global %>%
  left_join(uid, by = c("Province_State", "Country_Region")) %>% #add population data in global datafra
  select(-c(UID, FIPS)) %>%
  select(Province_State, Country_Region, date,
         cases, deaths, Population,
         Combined_Key)
```

US Covid Data Transformation

Now, let's look at US data

```
US_cases <- US_cases %>%
    pivot_longer(cols = -(UID:Combined_Key), #pivot data
                 names_to = "date",
                 values_to = "cases") %>%
    select(Admin2:cases) %>%
    mutate(date = mdy(date)) %>% #change datatype of date column
  select(-c(Lat, Long_))   #remove unnecessary columns

US_deaths <- US_deaths %>%
```

```
    pivot_longer(cols = -(UID:Population), #pivot data
                 names_to = "date",
                 values_to ="deaths") %>%
    select(Admin2:deaths) %>%
    mutate(date = mdy(date)) %>%  #change datatype of date column
 select(-c(Lat, Long_))  #remove unnecessary columns


US <- US_cases %>%
  full_join(US_deaths) #combine US cases and deaths data
```

Everything looks good, except rows having min cases = 0.

I don't need rows with cases = 0 for my analysis, so I will get rid of rows with no cases.

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

add US states area data

For our analysis, we will need US States area data and we don't have this data. Let's read this data and add it to the US dataframe.

First, read the area data and then combines it with the US data on the Province_State

```
area_lookup_url <- ("https://raw.githubusercontent.com/jakevdp/data-USstates/master/state-areas.csv")
area <- read_csv(area_lookup_url,show_col_types = FALSE) %>% mutate(Province_State = state) %>% select(


US <- US %>%
  left_join(area, by = c("Province_State"))

US <- US %>% mutate(pop_density = Population / `area (sq. mi)`)
```

Summarize US data

```
summary(US)
```

```
##     Admin2          Province_State     Country_Region     Combined_Key
##  Length:1740629     Length:1740629     Length:1740629     Length:1740629
##  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.:2020-08-15   1st Qu.:    171   1st Qu.:   11336   1st Qu.:    2.0
##  Median :2020-12-27   Median :    975   Median :   26734   Median :   18.0
##  Mean   :2020-12-26   Mean   :   5860   Mean   :  106757   Mean   :  111.7
##  3rd Qu.:2021-05-10   3rd Qu.:   3414   3rd Qu.:   69922   3rd Qu.:   64.0
##  Max.   :2021-09-21   Max.   :1446348   Max.   :10039107   Max.   :25870.0
##
##  area (sq. mi)     pop_density
##  Min.   :    68   Min.   :    0.000
##  1st Qu.: 44828   1st Qu.:    0.157
```

```
##  Median : 59441   Median :    0.451
##  Mean   : 82996   Mean   :    6.412
##  3rd Qu.: 83574   3rd Qu.:    1.378
##  Max.   :656425   Max.   :10378.662
##  NA's   :2760     NA's   :2760
```

Visualize Global Data

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

Let's look at the total number of cases over time and the total deaths over time for world as a whole and for a given country

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

```
## Adding missing grouping variables: `Province_State`
```

```
summary(global_by_country)
```

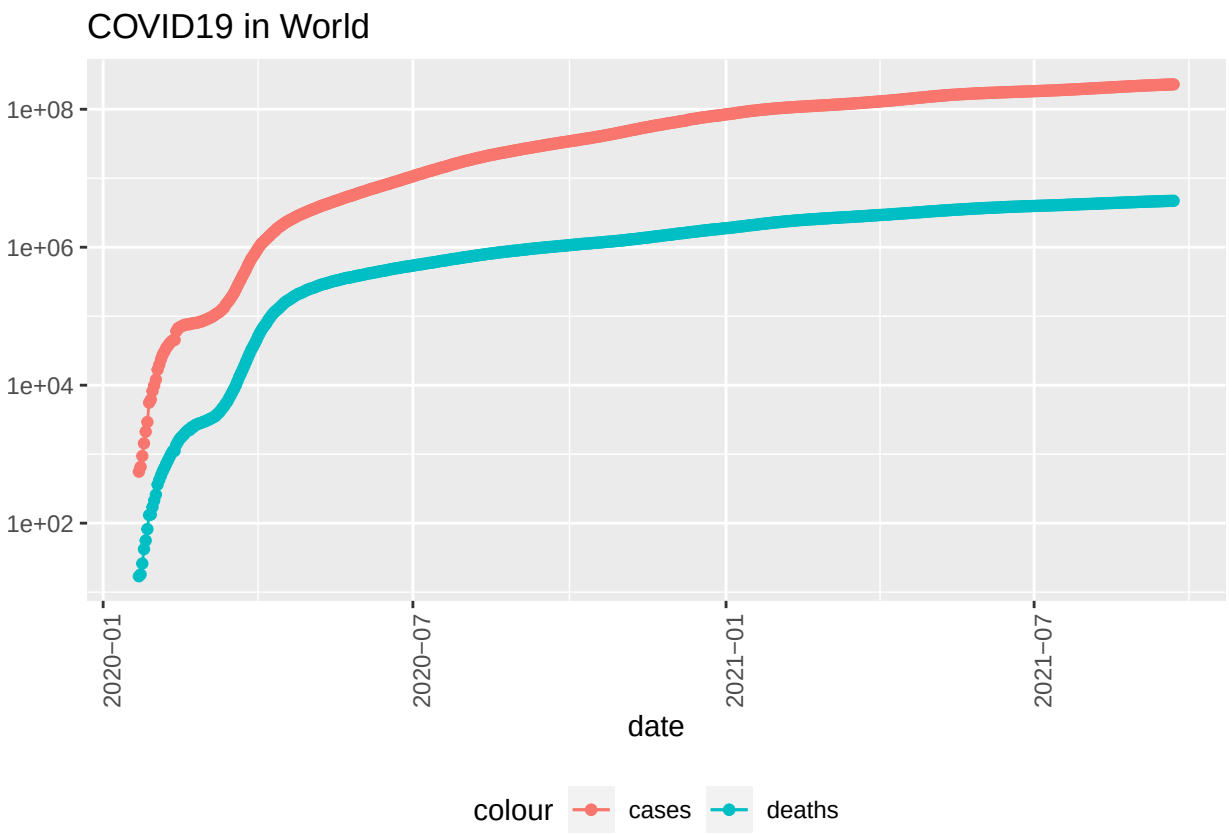
```
##  Province_State     Country_Region          date                cases
##  Length:153895      Length:153895      Min.   :2020-01-22   Min.   :       1
##  Class :character   Class :character   1st Qu.:2020-07-24   1st Qu.:     361
##  Mode  :character   Mode  :character   Median :2020-12-15   Median :    4224
##                                        Mean   :2020-12-12   Mean   :  318091
##                                        3rd Qu.:2021-05-05   3rd Qu.:   69708
##                                        Max.   :2021-09-21   Max.   :42410607
##
##      deaths       deaths_per_mill    cases_per_mill       Population
##  Min.   :     0   Min.   :   0.000   Min.   :     0.0   Min.   :8.090e+02
##  1st Qu.:     3   1st Qu.:   0.298   1st Qu.:   116.7   1st Qu.:9.775e+05
##  Median :    64   Median :  23.529   Median :  1490.3   Median :7.497e+06
##  Mean   :  7328   Mean   : 257.583   Mean   : 14682.5   Mean   :2.984e+07
##  3rd Qu.:  1232   3rd Qu.: 226.320   3rd Qu.: 14826.9   3rd Qu.:3.102e+07
##  Max.   :678407   Max.   :6037.454   Max.   :214327.8   Max.   :1.380e+09
##                   NA's   :2129       NA's   :2129       NA's   :2129
```

Let's create a new dataframe with aggregated data at the date level ie. e remove the country level granularity

```
global_totals <- global_by_country %>%
  group_by( date) %>%
   summarize(cases = sum(cases), deaths = sum(deaths),
             Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths *1000000 / Population) %>% #calculate deaths per thousand
  mutate(cases_per_mill = cases *1000000 / Population)%>%  #calculate cases per thousand
  select( date,
         cases, deaths, deaths_per_mill, cases_per_mill) %>%  #select required columns
  ungroup()
```

Analyze global cases and deaths over time

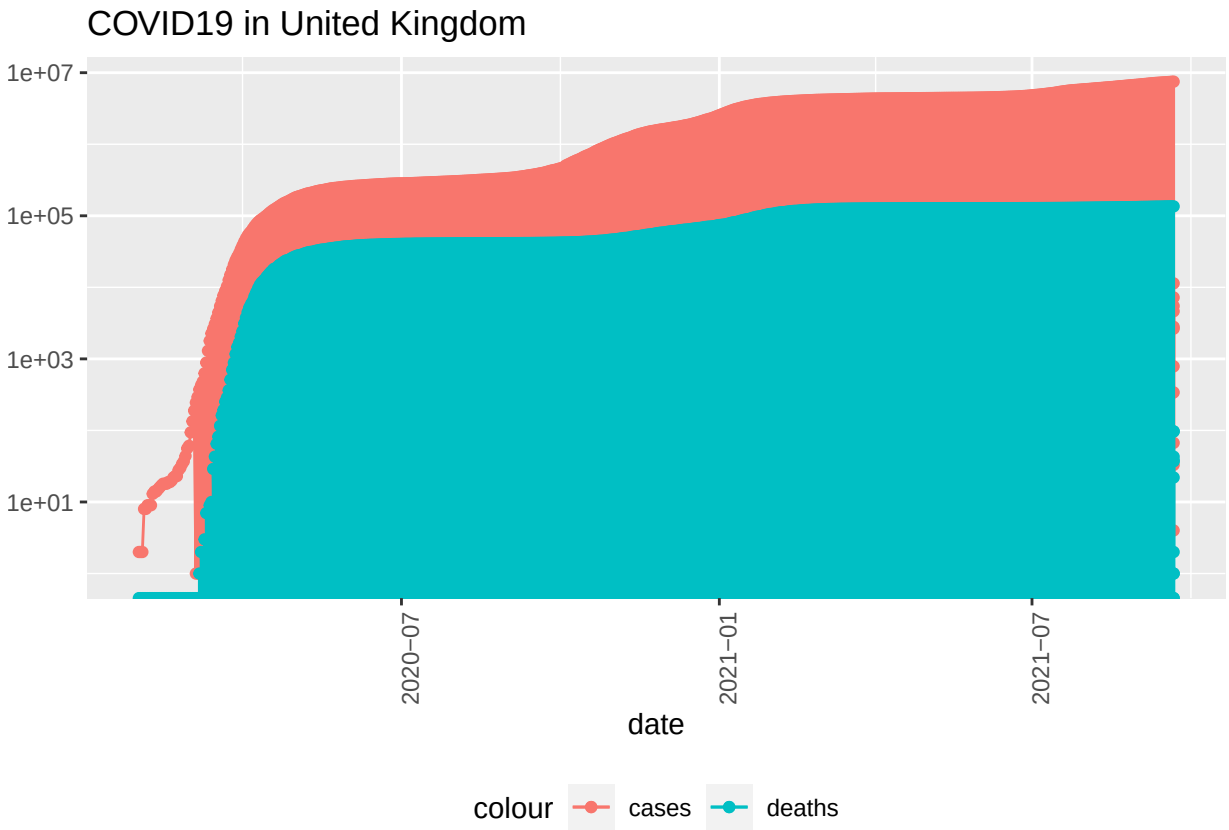
```
global_totals %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = cases)) +
    geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) +
    scale_y_log10() +
    theme(legend.position="bottom",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



##United Kingdom total cases and total deaths over time

```
country <- "United Kingdom"
global_by_country %>%
  filter(Country_Region == country) %>%
  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 ", country), y= NULL)
```

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

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



Visualize US Data

let's summarize US data first and look for anomalies

```
summary(US)
```

```
##     Admin2          Province_State     Country_Region     Combined_Key
##  Length:1740629     Length:1740629     Length:1740629     Length:1740629
##  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.:2020-08-15   1st Qu.:    171   1st Qu.:   11336   1st Qu.:    2.0
##  Median :2020-12-27   Median :    975   Median :   26734   Median :   18.0
##  Mean   :2020-12-26   Mean   :   5860   Mean   :  106757   Mean   :  111.7
##  3rd Qu.:2021-05-10   3rd Qu.:   3414   3rd Qu.:   69922   3rd Qu.:   64.0
##  Max.   :2021-09-21   Max.   :1446348   Max.   :10039107   Max.   :25870.0
##
##  area (sq. mi)     pop_density
```

```
##  Min.   :    68   Min.   :    0.000
##  1st Qu.: 44828   1st Qu.:    0.157
##  Median : 59441   Median :    0.451
##  Mean   : 82996   Mean   :    6.412
##  3rd Qu.: 83574   3rd Qu.:    1.378
##  Max.   :656425   Max.   :10378.662
##  NA's   :2760     NA's   :2760
```

Outliers in pop_density data

As you can see here, for population density mean is far greater than 3rd quartile, How is this possible? let's figure it out.

Let's take a look at top 10 state with higher population density.

```
US %>% group_by(Province_State) %>% summarize(pop_density = mean(pop_density)) %>%
  slice_max(pop_density, n = 10) %>% select(Province_State, Province_State, pop_density)
```

```
## # A tibble: 10 x 2
##    Province_State       pop_density
##    <chr>                      <dbl>
##  1 District of Columbia     10379.
##  2 Delaware                   127.
##  3 Rhode Island               116.
##  4 Connecticut                 71.9
##  5 Massachusetts               47.1
##  6 New Jersey                  46.6
##  7 Hawaii                      25.6
##  8 Maryland                    20.5
##  9 Puerto Rico                 13.4
## 10 New Hampshire               13.4
```

see "District of Columbia" has population density 10378.66176, which is too higher than rest of the state, hence it will create a bias in our analysis.. let's get rid of it..

```
US <- US %>% filter(Province_State!= "District of Columbia")

summary(US)
```

```
##     Admin2          Province_State     Country_Region     Combined_Key
##  Length:1740074     Length:1740074     Length:1740074     Length:1740074
##  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.:2020-08-15   1st Qu.:    171   1st Qu.:   11336   1st Qu.:    2.0
##  Median :2020-12-27   Median :    974   Median :   26729   Median :   18.0
##  Mean   :2020-12-26   Mean   :   5853   Mean   :  106566   Mean   :  111.5
##  3rd Qu.:2021-05-10   3rd Qu.:   3410   3rd Qu.:   69872   3rd Qu.:   64.0
##  Max.   :2021-09-21   Max.   :1446348   Max.   :10039107   Max.   :25870.0
##
##  area (sq. mi)     pop_density
##  Min.   :  1545   Min.   :  0.0000
```

```
##  1st Qu.: 44828   1st Qu.:  0.1571
##  Median : 59441   Median :  0.4506
##  Mean   : 83022   Mean   :  3.0981
##  3rd Qu.: 83574   3rd Qu.:  1.3707
##  Max.   :656425   Max.   :413.5476
##  NA's   :2760     NA's   :2760
```

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

Let's look at the total number of cases over time and the total deaths over time for the US as a whole and for a given state.

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

```
summary(US_by_state)
```

```
##  Province_State     Country_Region          date                cases
##  Length:31686       Length:31686       Min.   :2020-01-22   Min.   :      1
##  Class :character   Class :character   1st Qu.:2020-07-24   1st Qu.:  12363
##  Mode  :character   Mode  :character   Median :2020-12-13   Median : 108148
##                                        Mean   :2020-12-12   Mean   : 321410
##                                        3rd Qu.:2021-05-03   3rd Qu.: 407246
##                                        Max.   :2021-09-21   Max.   :4651497
##
##      deaths      deaths_per_mill  cases_per_mill       Population
##  Min.   :    0   Min.   :   0.0   Min.   :    0.15   Min.   :       0
##  1st Qu.:  267   1st Qu.: 186.7   1st Qu.: 8430.41   1st Qu.: 1344212
##  Median : 2030   Median : 699.1   Median :40751.22   Median : 3754939
##  Mean   : 6121   Mean   :   Inf   Mean   :     Inf   Mean   : 5852174
##  3rd Qu.: 7325   3rd Qu.:1618.8   3rd Qu.:99895.98   3rd Qu.: 6863772
##  Max.   :68087   Max.   :   Inf   Max.   :     Inf   Max.   :39512223
##                  NA's   :576
##   pop_density
##  Min.   :   0.00
##  1st Qu.:  42.86
##  Median :  93.53
##  Mean   : 186.07
##  3rd Qu.: 199.57
##  Max.   :1068.26
##  NA's   :2760
```

```
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()
```

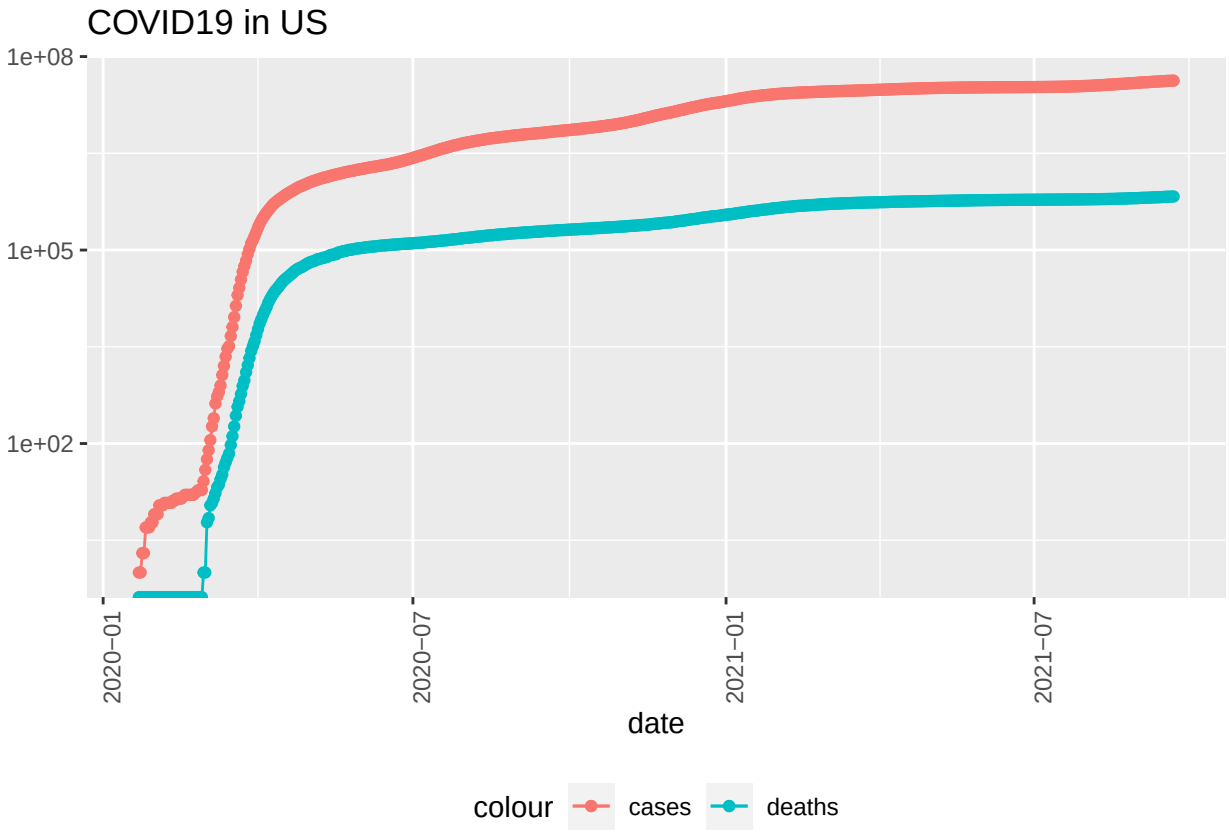
We want to visualize total cases and deaths in US, for that we will create new dataframe with aggregate data at date level i.e. we shall get rid of the state granularity of data.

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

```
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)
```

Now,let's visualize US total cases and deaths over time

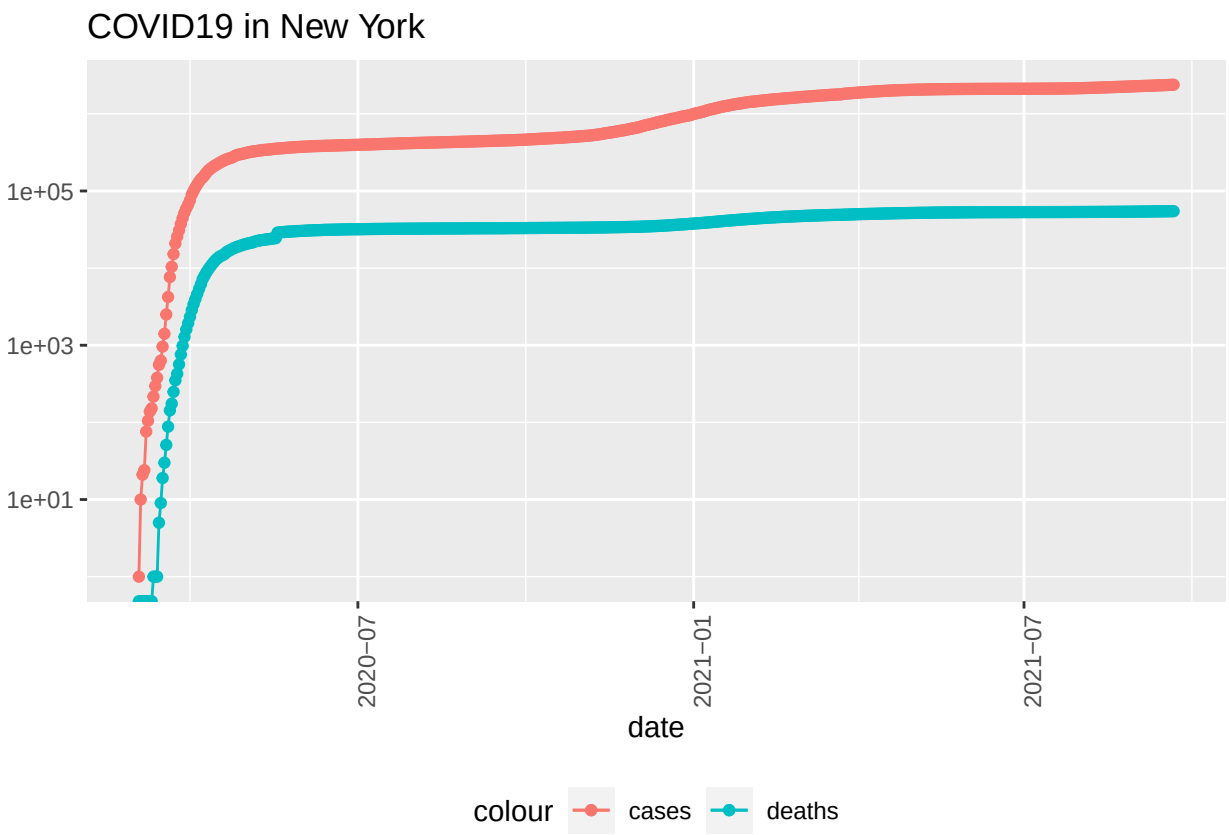
```
## Warning: Transformation introduced infinite values in continuous y-axis

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



```
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)
```

New York state total cases and total deaths over time

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

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



Analyze the data

Total deaths in the US as of 2021-09-08 is 6.52657^{5}.

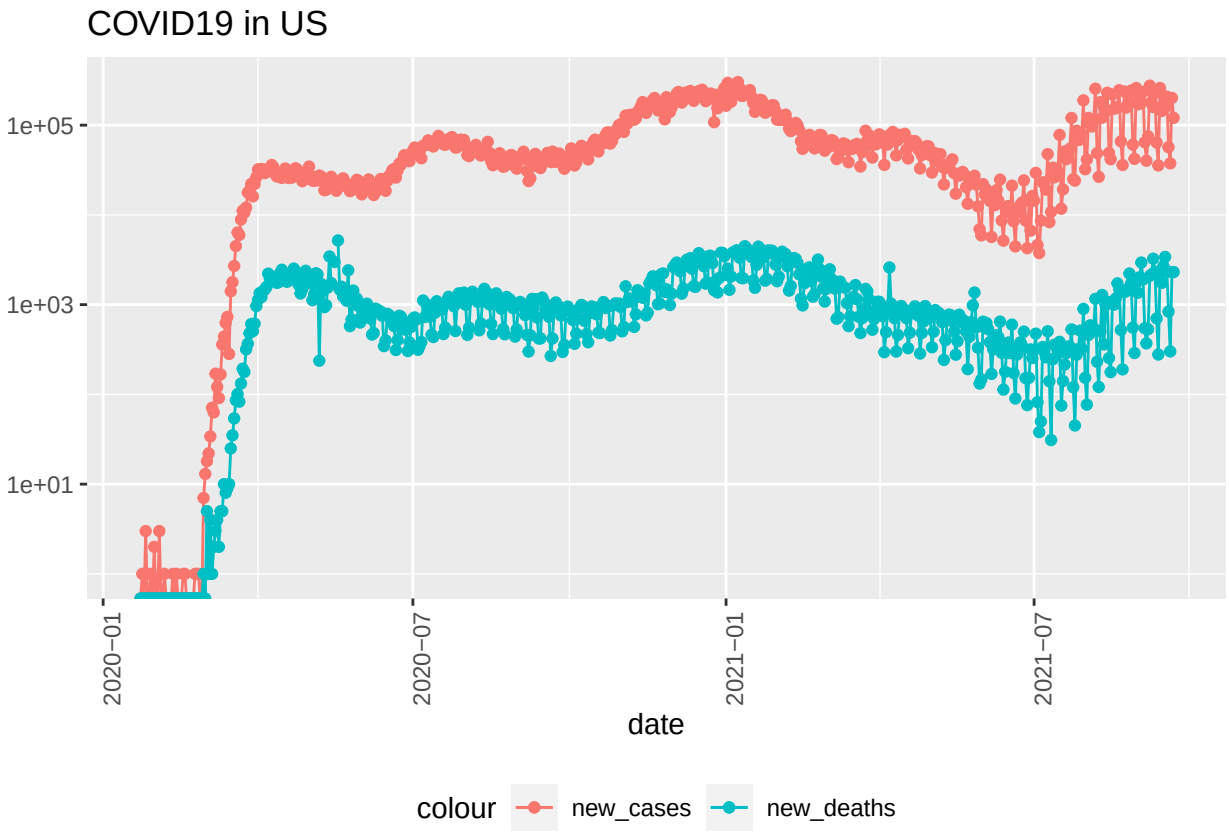
So our graph looks like COVID has leveled off. Is that true? Look at the number of new cases and deaths per day.

```
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))
```

```
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)
```

Visualize these to see if that raises new questions

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

## 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 3 rows containing missing values (geom_point).
```

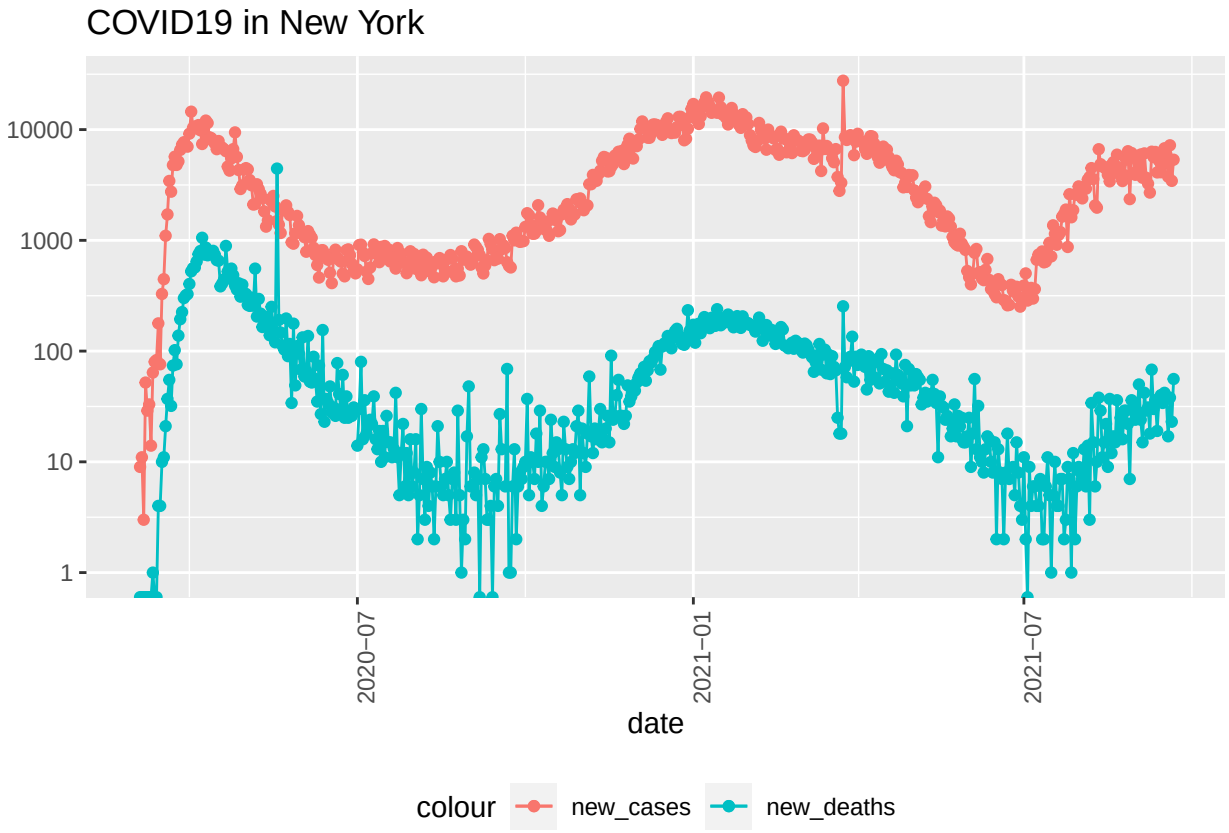


Plot a state

```
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 4 rows containing missing values (geom_point).
```



worst and best states? How to measure this? Perhaps look at case rates and death rates per 1000 people?

```
US_state_totals <- US_by_state %>%
  group_by(Province_State) %>%
  summarize(deaths = max(deaths), cases = max(cases),
            population = max(Population), pop_density =max(pop_density),
            cases_per_thou = 1000* cases / population,
            deaths_per_thou = 1000 * deaths / population) %>%
  filter(cases > 0, population > 0)
```

States with minimum death rates per thousand

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

```
## # A tibble: 10 x 7
##    Province_State           deaths  cases population pop_density cases_per_thou
##    <chr>                     <dbl>  <dbl>      <dbl>       <dbl>          <dbl>
##  1 Northern Mariana Islands      2    265      55144       NA              4.81
##  2 Vermont                     301  31911     623989       64.9           51.1
##  3 Hawaii                      714  76191    1415872      130.            53.8
##  4 Virgin Islands               68   6516     107268       NA             60.7
##  5 Alaska                      480 103327     728809        1.11         142.
##  6 Maine                      1002  84542    1344212       38.0           62.9
```

```
##  7 Puerto Rico                3092 179523    3754939     1068.            47.8
##  8 Oregon                     3624 314841    4217737       42.9           74.6
##  9 Washington                 7315 631023    7614893      107.            82.9
## 10 Utah                       2829 495704    2785478       32.8          178.
## # ... with 1 more variable: deaths_per_thou <dbl>
```

States with maximum death rates per thousand

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

```
## # A tibble: 10 x 7
##    Province_State deaths   cases population pop_density cases_per_thou
##    <chr>           <dbl>   <dbl>      <dbl>       <dbl>          <dbl>
##  1 Mississippi      9331  477769    2976149        61.4           161.
##  2 New Jersey      27240 1137016    8882190      1018.            128.
##  3 Louisiana       13558  730099    4648794        89.7           157.
##  4 New York        54695 2382450   19453561       357.            122.
##  5 Alabama         13460  775531    4903185        93.5           158.
##  6 Massachusetts   18480  796925    6863772       650.            116.
##  7 Arizona         19584 1070757    7278717        63.8           147.
##  8 Rhode Island     2816  169686    1059361       686.            160.
##  9 Arkansas         7499  486853    3017804        56.7           161.
## 10 Florida         51889 3528698   21477737       327.            164.
## # ... with 1 more variable: deaths_per_thou <dbl>
```

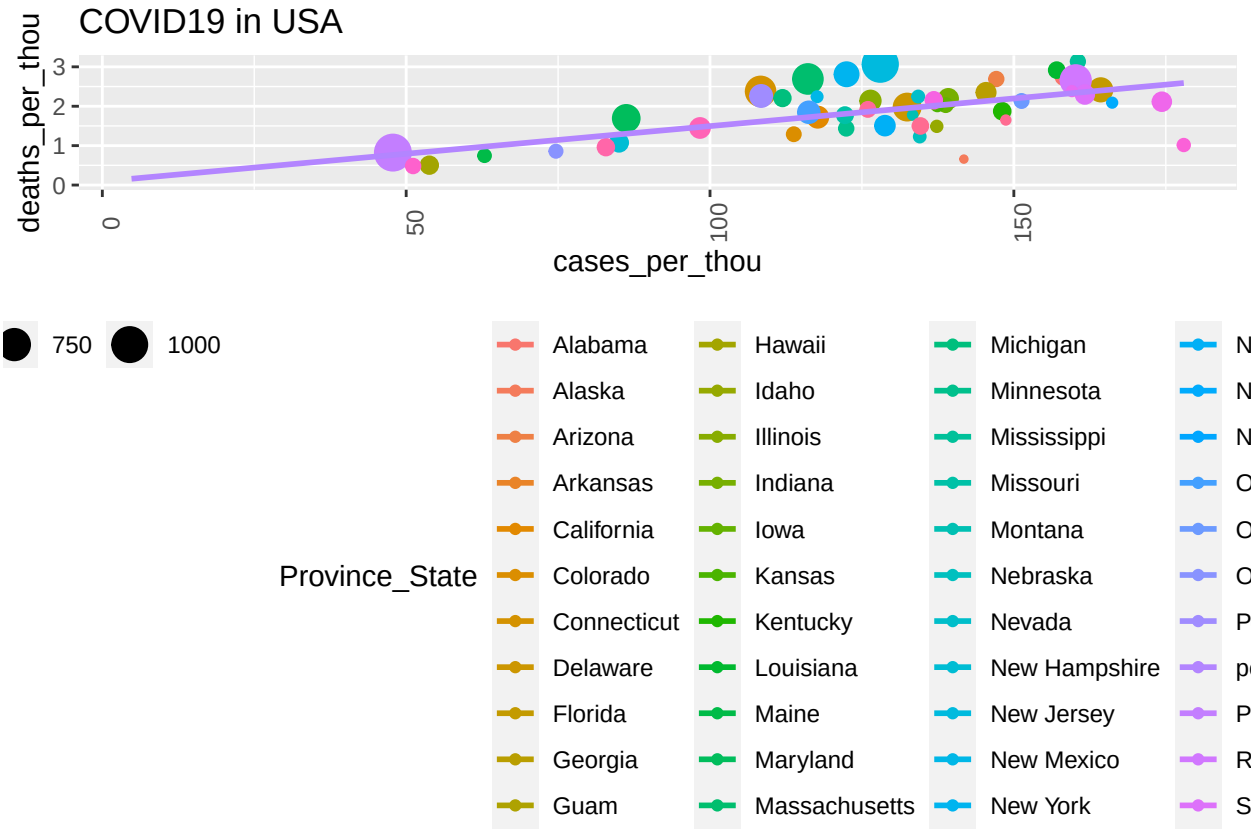
US States Covid analysis with population density

Let's see how the Deaths and cases are correlated in different US states as per population density

```
state <- "District of Columbia"
(US_state_totals %>%
  filter(Province_State != state) %>%
  ggplot( aes(x = cases_per_thou , y = deaths_per_thou)) +
  geom_point(aes(size = pop_density ,  color = Province_State)) +
    geom_smooth(aes(color ="pop_density"),method="lm",se = FALSE)+
    theme(legend.position="bottom",
          axis.text.x = element_text(angle = 90)) +
  labs(title = str_c("COVID19 in USA")))
```

```
## `geom_smooth()` using formula 'y ~ x'
```

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



from the graph we can state that, states having higher population density has higher death rate than the state with lower population density.

But, wait a sec, Have you observed Puerto Rico, it is one of the highest populated state, but have very low cases and death rate than other state.

let's analyze Pureto Rico state to see what's going on there.

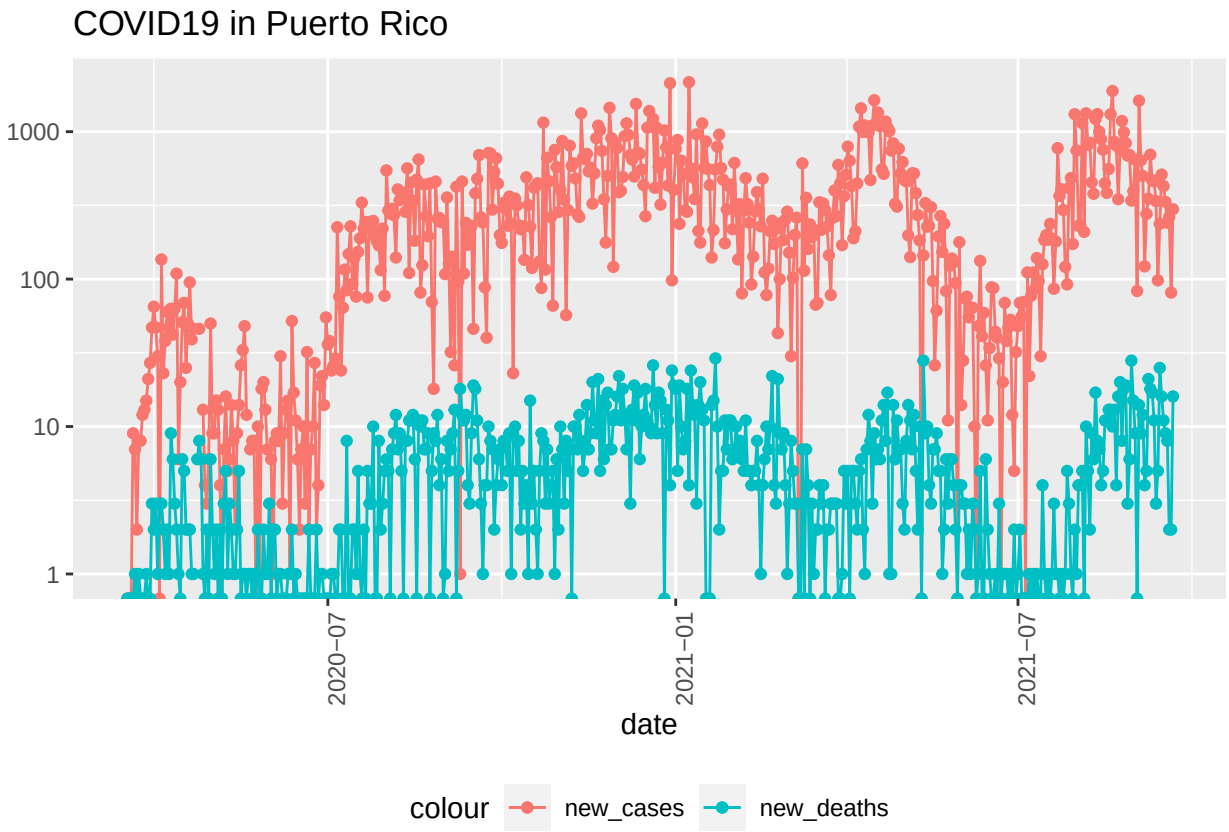
```
US_state_totals %>% filter(Province_State == "Puerto Rico")
```

```
## # A tibble: 1 x 7
##   Province_State deaths  cases population pop_density cases_per_thou
##   <chr>           <dbl>  <dbl>      <dbl>       <dbl>          <dbl>
## 1 Puerto Rico      3092 179523    3754939       1068.           47.8
## # ... with 1 more variable: deaths_per_thou <dbl>
```

Everything looks okay on aggregate level. let's analyze it in more detail at daily level

```
state <- "Puerto Rico"
(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 6 rows containing missing values (geom_point).
## Warning: Removed 1 row(s) containing missing values (geom_path).
## Warning: Removed 2 rows containing missing values (geom_point).
```



Everything looks good in analysis we have done. Could be some other external factors affected the numbers, or Puerto Rico did something different than other states to control spread of Covid-19. But currently, we do not have more attribute to analyze these external factors.

Model the data

We might need to introduce more variables here to build a model. Which do you want to consider? Population density, extent of lock down, political affiliation, climate of the area? When you determine the factors you

want to try, add that data to your dataset, and then visualize and model and see if your variable has a statistically significant effect.

Let's regress the deeaths per thousand on cases per thousand

```
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
## -1.57387 -0.30726 -0.01599  0.27150  1.17823
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.092272   0.251774   0.366    0.715
## cases_per_thou 0.014032   0.001969   7.126 3.12e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5237 on 52 degrees of freedom
## Multiple R-squared:  0.494,  Adjusted R-squared:  0.4843
## F-statistic: 50.77 on 1 and 52 DF,  p-value: 3.121e-09
```

look at the state with minimum cases per thousand

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

```
## # A tibble: 1 x 7
##   Province_State           deaths cases population pop_density cases_per_thou
##   <chr>                     <dbl> <dbl>      <dbl>       <dbl>          <dbl>
## 1 Northern Mariana Islands      2   265      55144          NA           4.81
## # ... with 1 more variable: deaths_per_thou <dbl>
```

look at the state with maximum cases per thousand

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

```
## # A tibble: 1 x 7
##   Province_State deaths  cases population pop_density cases_per_thou
##   <chr>           <dbl>  <dbl>      <dbl>       <dbl>          <dbl>
## 1 Utah             2829 495704    2785478        32.8           178.
## # ... with 1 more variable: deaths_per_thou <dbl>
```

let's try to predict number of deaths wrt cases

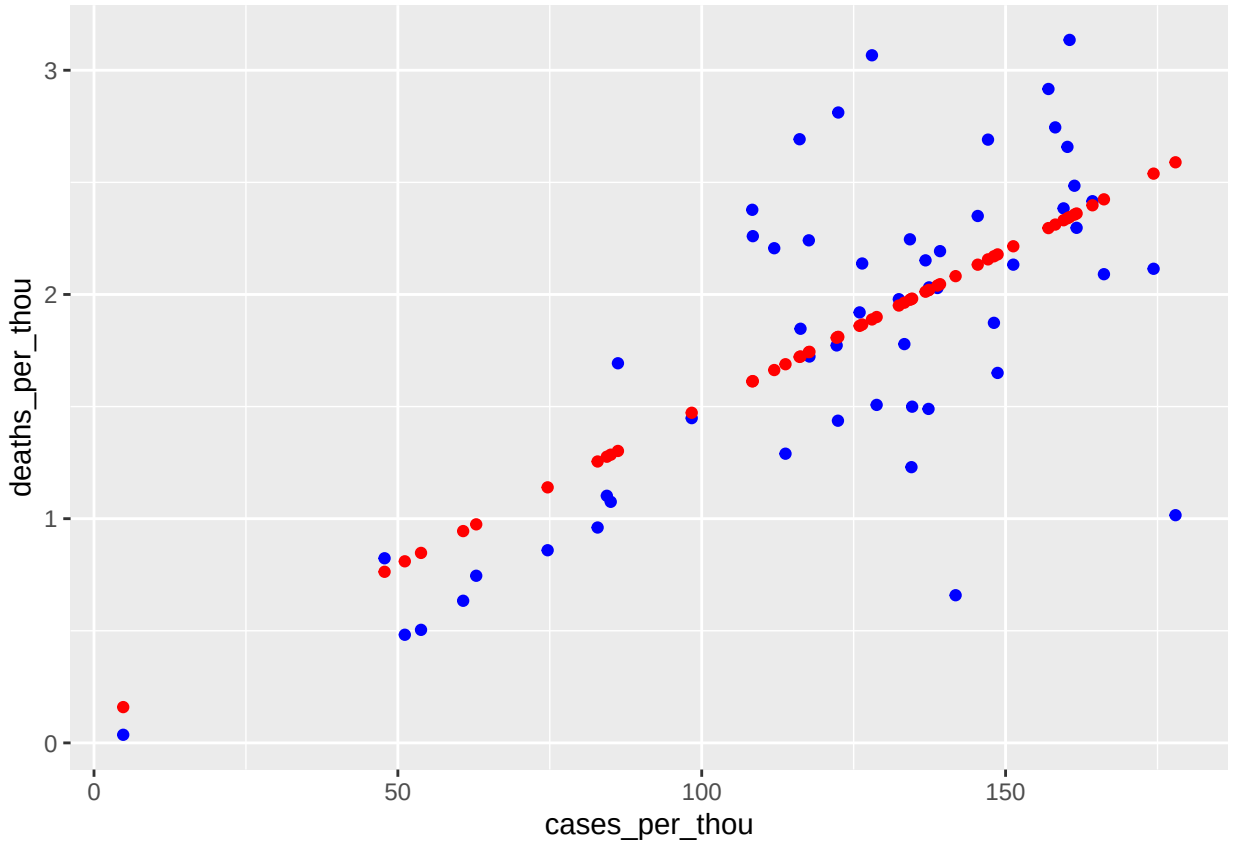
```
x_grid <- seq(1, 151)
new_df <- tibble(cases_per_thou = x_grid)
US_state_totals %>% mutate(pred = predict(mod))
```

```
## # A tibble: 54 x 8
##    Province_State deaths   cases population pop_density cases_per_thou
##    <chr>           <dbl>   <dbl>      <dbl>       <dbl>          <dbl>
##  1 Alabama         13460  775531    4903185       93.5            158.
##  2 Alaska            480  103327     728809        1.11           142.
##  3 Arizona         19584 1070757    7278717       63.8            147.
```

```
##  4 Arkansas          7499  486853    3017804       56.7           161.
##  5 California       68087 4651497   39512223      241.            118.
##  6 Colorado          7428  655244    5758736       55.3           114.
##  7 Connecticut       8477  386182    3565287      643.            108.
##  8 Delaware          1927  128964     973764      498.            132.
##  9 Florida          51889 3528698   21477737      327.            164.
## 10 Georgia          24951 1543960   10617423      179.            145.
## # ... with 44 more rows, and 2 more variables: deaths_per_thou <dbl>,
## #   pred <dbl>
```

let's visualize it

```
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")
```



Biases

Possible Sources of biases

1. Data Collection Bias In n different ways data collection bias can occur. Some of them I am listing here: 1. people with covid not getting tested 2. possible multiple test for a person 3. Nursing house deaths not counted 4. How to count a death as a covid 5. different data from different places 6. False positive and false negative results

2. Algorithm Selection Bias We are linearly regressing Covid deaths with the Covid cases, due to time limit. But linear regression algorithm is not best in our case.

3. Result interpretation bias Some doctors may analyze Pneumonia as a Covid or vice versa This is one example of result interpretation bias.

Conclusion

After analyzing covid data, we can conclude that there is a positive correlation between the number of cases and the number of deaths. Also, we can say that there is a positive correlation between covid cases, deaths, and population density. As population density increases covid cases and deaths also rise.