COVID19_steps.Rmd

T. S

2022-06-18

Import Data

```
##Get current Data in the four files
#they all begin the same way
###call the tidyverse library
library("tidyverse")
```

```
## Warning: package 'tidyverse' was built under R version 3.6.3
```

```
## -- Attaching packages --------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.1     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## Warning: package 'tibble' was built under R version 3.6.3
```

```
## Warning: package 'tidyr' was built under R version 3.6.3
```

```
## Warning: package 'readr' was built under R version 3.6.3
```

```
## Warning: package 'purrr' was built under R version 3.6.3
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
## Warning: package 'forcats' was built under R version 3.6.3
```

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

```
url_in<-"https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid
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)
```

Let's read in the data and see what we have.

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

##Tidy Data

I'm going to get rid of the Lat and Long because I don't need that for the analysis I'm going to be planning. And also, I want the country/region and province/state to be a little more r friendly, I also want to make this what I call tidy, which means what I really would like is to have each date on a separate row. Because what I'm looking at in this case, is the total cases. But it's the cases per each date. So I'm going to fix all of that.

I'm going to take what I had, its global cases and I'm going to pivot longer, which means I'm going to make each one into a row. Everything except the province/state, the country/region, the Lat/Long. The names was the column headings are now going to be date, and the values will go to cases. And then I will select everything except the Lat and Long, and I will see then what my result is. So now let's look at what global cases looks like.

```
#ctlr+alt+I cmd to insert chunk
global_cases<-global_cases %>%
  pivot_longer(cols = -c('Province/State','Country/Region', Lat, Long),
               names_to="date",
               values_to="cases")%>%
  select(-c(Lat,Long))
```

We are going to tidy others Global_death, US_cases and US_deaths

Pivot date and delete Lat, Long

```
#ctlr+alt+I cmd to insert chunk
global_deaths<-global_deaths %>%
  pivot_longer(cols = -c('Province/State','Country/Region', Lat, Long),
               names_to="date",
               values_to="deaths")%>%
  select(-c(Lat,Long))
```

We will combine the cases and deaths per date into one variable we will call global. So what we will do is we will join the cases with the deaths and then we will rename our country region, just to get rid of this slash mark and the same with province state. We'll also notice that our date was not a date objects we will make it a date object.

Combine Cases and deaths to global and change format of date

```
library("lubridate")
```

```
## Warning: package 'lubridate' was built under R version 3.6.3
```

```
##
## Attaching package: 'lubridate'
```

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

```
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")
```

```
Joining_by =c("Province/State", "Country/Region", "date")
```

```
global
```

```
## # A tibble: 330,327 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 330,317 more rows
```

```
summary(global)
```

```
##  Province_State     Country_Region          date
##  Length:330327      Length:330327      Min.   :2020-01-22
##  Class :character   Class :character   1st Qu.:2020-11-02
##  Mode  :character   Mode  :character   Median :2021-08-15
##                                        Mean   :2021-08-15
##                                        3rd Qu.:2022-05-28
##                                        Max.   :2023-03-09
##      cases                deaths
##  Min.   :        0   Min.   :      0
##  1st Qu.:      680   1st Qu.:      3
##  Median :    14429   Median :    150
##  Mean   :   959384   Mean   :  13380
##  3rd Qu.:   228517   3rd Qu.:   3032
##  Max.   :103802702   Max.   :1123836
```

I think I have a lot of case of rows that have no cases at all. So I think I would like to get rid of those. So I think what I will do is I will filter out and keep only where the cases are positive

Delete row with 0 case

```
global<-global %>% filter(cases>=1)
summary(global)
```

```
##  Province_State     Country_Region          date
##  Length:306827      Length:306827      Min.   :2020-01-22
##  Class :character   Class :character   1st Qu.:2020-12-12
##  Mode  :character   Mode  :character   Median :2021-09-16
##                                        Mean   :2021-09-11
##                                        3rd Qu.:2022-06-15
##                                        Max.   :2023-03-09
##      cases               deaths
##  Min.   :        1   Min.   :      0
##  1st Qu.:     1316   1st Qu.:      7
##  Median :    20365   Median :    214
##  Mean   :  1032863   Mean   :  14405
##  3rd Qu.:   271281   3rd Qu.:   3665
##  Max.   :103802702   Max.   :1123836
```

Now we need to verified that the maximum cases that we have in the summary is correct

```
global1<-global %>% filter(cases>103000000)
global1
```

```
## # A tibble: 23 x 5
##    Province_State Country_Region date           cases  deaths
##    <chr>          <chr>          <date>         <dbl>   <dbl>
##  1 <NA>           US             2023-02-15 103023231 1115741
##  2 <NA>           US             2023-02-16 103083910 1116851
##  3 <NA>           US             2023-02-17 103131898 1117572
##  4 <NA>           US             2023-02-18 103134605 1117589
##  5 <NA>           US             2023-02-19 103136077 1117590
##  6 <NA>           US             2023-02-20 103138119 1117663
##  7 <NA>           US             2023-02-21 103198669 1118025
##  8 <NA>           US             2023-02-22 103308832 1118886
##  9 <NA>           US             2023-02-23 103365511 1119521
## 10 <NA>           US             2023-02-24 103378408 1119573
## # ... with 13 more rows
```

Now let check what is in US_cases and do the same Tidy process and we did with global_cases I have these weird codes, UID, iso2, iso3, code3, FIPS, Admin2, province_state, country region, Lat, Long. So there's some things in there that I don't need. So what I'm going to start with is, I know I want to pivot all these data. And I think I'm going to keep Admin2 through the number of cases once I get that done ## Tidy US_Cases Let check which variables we got in US_Cases

```
US_cases
```

```
## # A tibble: 3,342 x 1,154
##         UID iso2  iso3  code3  FIPS Admin2   Province_State Country_Region
##       <dbl> <chr> <chr> <dbl> <dbl> <chr>    <chr>          <chr>
```

```
##  1 84001001 US    USA     840  1001 Autauga  Alabama        US
##  2 84001003 US    USA     840  1003 Baldwin  Alabama        US
##  3 84001005 US    USA     840  1005 Barbour  Alabama        US
##  4 84001007 US    USA     840  1007 Bibb     Alabama        US
##  5 84001009 US    USA     840  1009 Blount   Alabama        US
##  6 84001011 US    USA     840  1011 Bullock  Alabama        US
##  7 84001013 US    USA     840  1013 Butler   Alabama        US
##  8 84001015 US    USA     840  1015 Calhoun  Alabama        US
##  9 84001017 US    USA     840  1017 Chambers Alabama        US
## 10 84001019 US    USA     840  1019 Cherokee Alabama        US
## # ... with 3,332 more rows, and 1,146 more variables: Lat <dbl>,
## #   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>, ...
```

```
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: 3,819,906 x 6
##    Admin2  Province_State Country_Region Combined_Key      date       cases
##    <chr>   <chr>          <chr>          <chr>             <date>     <dbl>
##  1 Autauga Alabama        US             Autauga, Alabama~ 2020-01-22     0
##  2 Autauga Alabama        US             Autauga, Alabama~ 2020-01-23     0
##  3 Autauga Alabama        US             Autauga, Alabama~ 2020-01-24     0
##  4 Autauga Alabama        US             Autauga, Alabama~ 2020-01-25     0
##  5 Autauga Alabama        US             Autauga, Alabama~ 2020-01-26     0
##  6 Autauga Alabama        US             Autauga, Alabama~ 2020-01-27     0
##  7 Autauga Alabama        US             Autauga, Alabama~ 2020-01-28     0
##  8 Autauga Alabama        US             Autauga, Alabama~ 2020-01-29     0
##  9 Autauga Alabama        US             Autauga, Alabama~ 2020-01-30     0
## 10 Autauga Alabama        US             Autauga, Alabama~ 2020-01-31     0
## # ... with 3,819,896 more rows
```

Combined_key puts together the county and the state and then I have the date and the number of cases.

```
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: 3,819,906 x 7
##    Admin2  Province_State Country_Region Combined_Key Population date
##    <chr>   <chr>          <chr>          <chr>              <dbl> <date>
##  1 Autauga Alabama        US             Autauga, Al~       55869 2020-01-22
##  2 Autauga Alabama        US             Autauga, Al~       55869 2020-01-23
##  3 Autauga Alabama        US             Autauga, Al~       55869 2020-01-24
##  4 Autauga Alabama        US             Autauga, Al~       55869 2020-01-25
##  5 Autauga Alabama        US             Autauga, Al~       55869 2020-01-26
##  6 Autauga Alabama        US             Autauga, Al~       55869 2020-01-27
##  7 Autauga Alabama        US             Autauga, Al~       55869 2020-01-28
##  8 Autauga Alabama        US             Autauga, Al~       55869 2020-01-29
##  9 Autauga Alabama        US             Autauga, Al~       55869 2020-01-30
## 10 Autauga Alabama        US             Autauga, Al~       55869 2020-01-31
## # ... with 3,819,896 more rows, and 1 more variable: deaths <dbl>
```

I'm going to join it by province_state and country_region. And then I'm going to select everything except for those columns and then I will select these columns from it. And when I do that and look at global, now I see that for each country I have added the population to that data set.

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

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

```
Joining_by =c("Admin2", "Province/State", "Country/Region", "Combined_Key", "date")
US
```

```
## # A tibble: 3,819,906 x 8
##    Admin2  Province_State Country_Region Combined_Key      date       cases
##    <chr>   <chr>          <chr>          <chr>             <date>     <dbl>
##  1 Autauga Alabama        US             Autauga, Alabama~ 2020-01-22     0
##  2 Autauga Alabama        US             Autauga, Alabama~ 2020-01-23     0
##  3 Autauga Alabama        US             Autauga, Alabama~ 2020-01-24     0
##  4 Autauga Alabama        US             Autauga, Alabama~ 2020-01-25     0
##  5 Autauga Alabama        US             Autauga, Alabama~ 2020-01-26     0
##  6 Autauga Alabama        US             Autauga, Alabama~ 2020-01-27     0
##  7 Autauga Alabama        US             Autauga, Alabama~ 2020-01-28     0
##  8 Autauga Alabama        US             Autauga, Alabama~ 2020-01-29     0
##  9 Autauga Alabama        US             Autauga, Alabama~ 2020-01-30     0
## 10 Autauga Alabama        US             Autauga, Alabama~ 2020-01-31     0
## # ... with 3,819,896 more rows, and 2 more variables: Population <dbl>,
## #   deaths <dbl>
```

Now we notice we don't have population data for the world data. And if we're going to do comparative analysis between the countries, we will want to add the population data to our global data set.

So let's add a population data and a variable called Combined_key, that combines these two things the province_state in the country_region together, so that I'll have a similar sort of data set. So first, let me combine.

I'm going to do a Combined_keys, I'm going to use unite, which will combine together province_state, country_region. It will combine it with a comma and a space and put it in Combined_key in the global data set.

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

So now if I look at my global data set, it should have the same variables other than population.

```
global
```

```
## # A tibble: 306,827 x 6
##    Combined_Key Province_State Country_Region date       cases deaths
##    <chr>        <chr>          <chr>          <date>     <dbl>  <dbl>
##  1 Afghanistan  <NA>           Afghanistan    2020-02-24     5      0
##  2 Afghanistan  <NA>           Afghanistan    2020-02-25     5      0
##  3 Afghanistan  <NA>           Afghanistan    2020-02-26     5      0
##  4 Afghanistan  <NA>           Afghanistan    2020-02-27     5      0
##  5 Afghanistan  <NA>           Afghanistan    2020-02-28     5      0
##  6 Afghanistan  <NA>           Afghanistan    2020-02-29     5      0
##  7 Afghanistan  <NA>           Afghanistan    2020-03-01     5      0
##  8 Afghanistan  <NA>           Afghanistan    2020-03-02     5      0
##  9 Afghanistan  <NA>           Afghanistan    2020-03-03     5      0
## 10 Afghanistan  <NA>           Afghanistan    2020-03-04     5      0
## # ... with 306,817 more rows
```

So now I need to add population and so I find that same Johns Hopkins website has a CSV. I'm going to put that in a url.

Get global population data from the links below

```
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))
```

```
##
## -- Column specification --------------------------------------------------
## cols(
##   UID = col_double(),
##   iso2 = col_character(),
##   iso3 = col_character(),
##   code3 = col_double(),
##   FIPS = col_double(),
##   Admin2 = col_character(),
##   Province_State = col_character(),
##   Country_Region = col_character(),
##   Lat = col_double(),
##   Long_ = col_double(),
```

```
##   Combined_Key = col_character(),
##   Population = col_double()
## )
```

Now we need to join uid with the global dataset by province_state and Country_Region so we can have global with population

```
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: 306,827 x 7
##    Province_State Country_Region date       cases deaths Population
##    <chr>          <chr>          <date>     <dbl>  <dbl>      <dbl>
##  1 <NA>           Afghanistan    2020-02-24     5      0   38928341
##  2 <NA>           Afghanistan    2020-02-25     5      0   38928341
##  3 <NA>           Afghanistan    2020-02-26     5      0   38928341
##  4 <NA>           Afghanistan    2020-02-27     5      0   38928341
##  5 <NA>           Afghanistan    2020-02-28     5      0   38928341
##  6 <NA>           Afghanistan    2020-02-29     5      0   38928341
##  7 <NA>           Afghanistan    2020-03-01     5      0   38928341
##  8 <NA>           Afghanistan    2020-03-02     5      0   38928341
##  9 <NA>           Afghanistan    2020-03-03     5      0   38928341
## 10 <NA>           Afghanistan    2020-03-04     5      0   38928341
## # ... with 306,817 more rows, and 1 more variable: Combined_Key <chr>
```

Visualizing, Analysing, and Modeling Data

```
options(dplyr.summarise.inform = FALSE)

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()

US_by_state
```

```
## # A tibble: 66,294 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 66,284 more rows, and 1 more variable: Population <dbl>
```

```
US_totals <- US %>%
  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()

US_totals
```

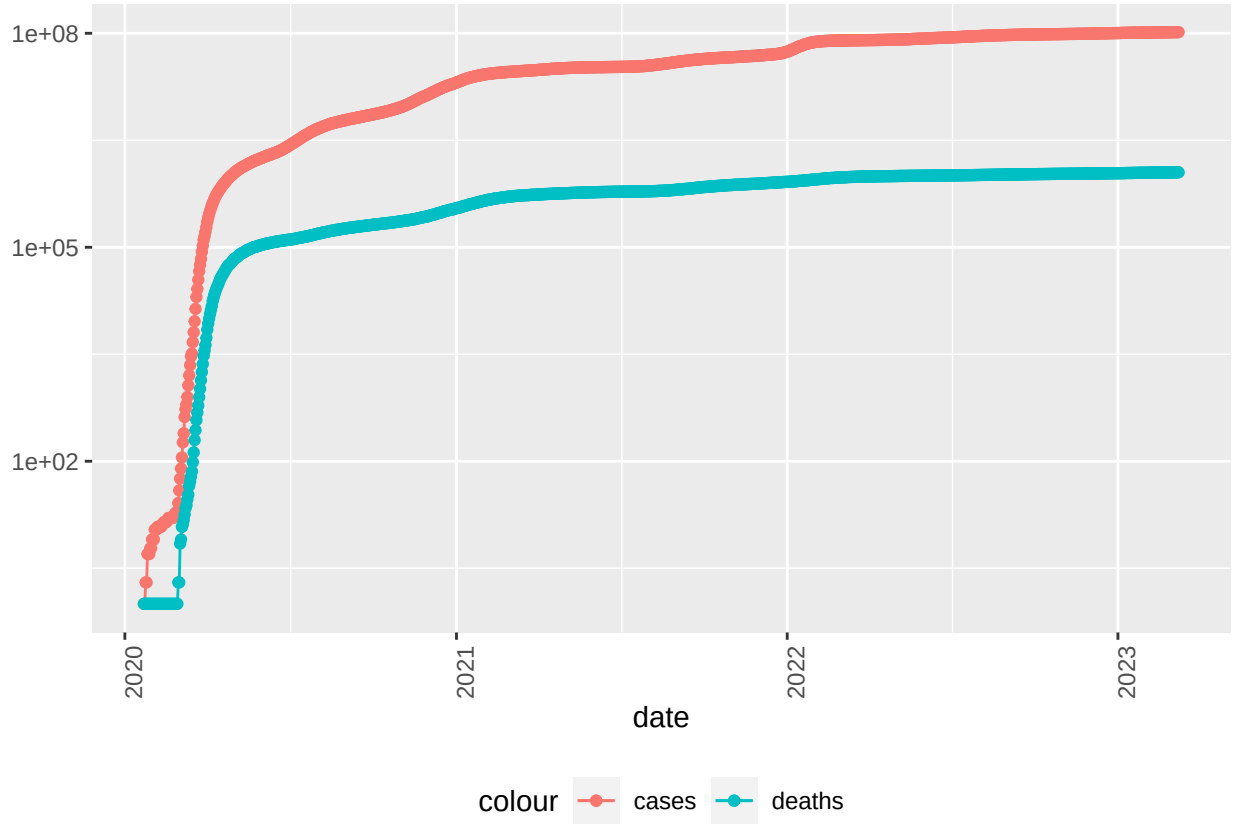
```
## # A tibble: 1,143 x 6
##    Country_Region date       cases deaths deaths_per_mill Population
##    <chr>          <date>     <dbl>  <dbl>           <dbl>      <dbl>
##  1 US             2020-01-22     1      1         0.00300  332875137
##  2 US             2020-01-23     1      1         0.00300  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 1,133 more rows
```

```
tail(US_totals)
```

```
## # A tibble: 6 x 6
##   Country_Region date           cases  deaths deaths_per_mill Population
##   <chr>          <date>         <dbl>   <dbl>           <dbl>      <dbl>
## 1 US             2023-03-04 103650837 1122172           3371.  332875137
## 2 US             2023-03-05 103646975 1122134           3371.  332875137
## 3 US             2023-03-06 103655539 1122181           3371.  332875137
## 4 US             2023-03-07 103690910 1122516           3372.  332875137
## 5 US             2023-03-08 103755771 1123246           3374.  332875137
## 6 US             2023-03-09 103802702 1123836           3376.  332875137
```

Let Visualze US_total

```
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(tittle = "COVID19 in US", y= NULL)
```



New york case

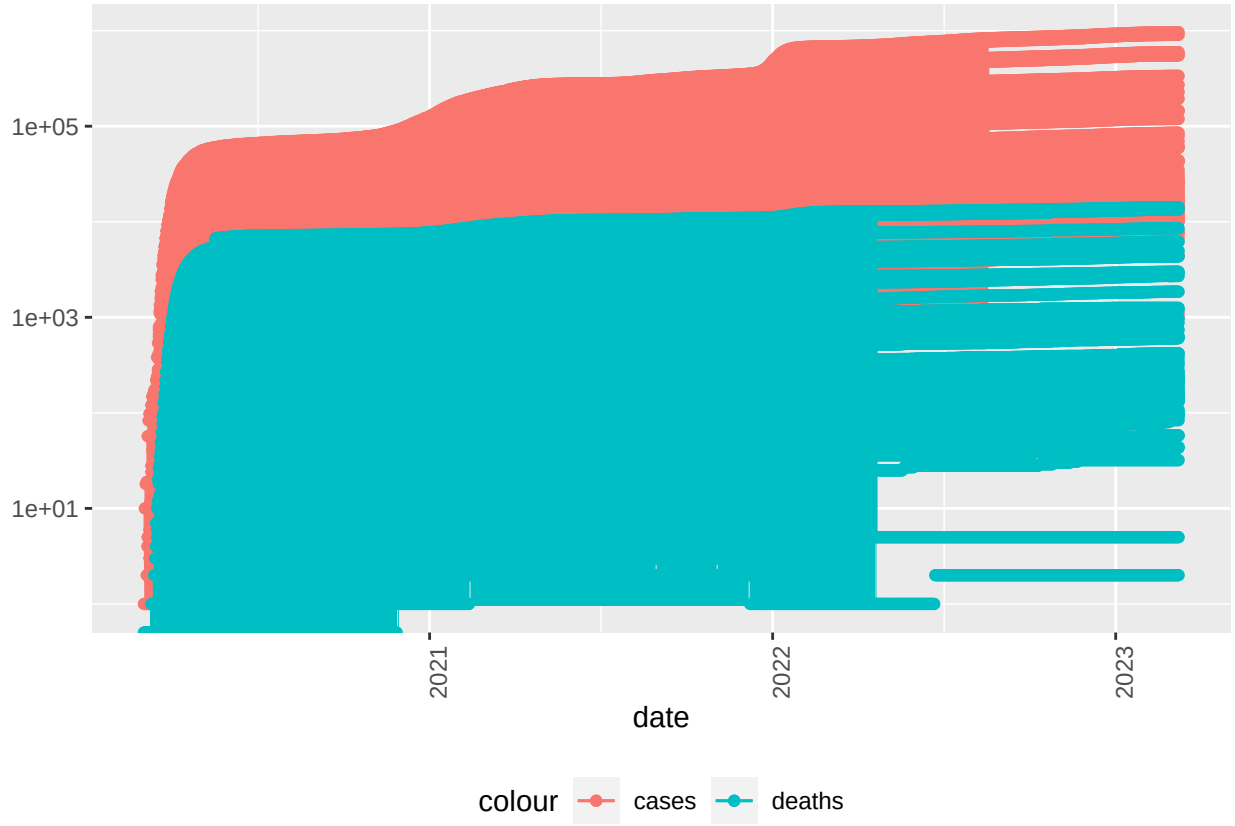
```
state <- "New York"
US %>%

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

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

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



##Let check the maximum date and death of our data

```
max(US_totals$date)
```

```
## [1] "2023-03-09"
```

```
max(US_totals$deaths)
```

```
## [1] 1123836
```

Let now group our global data

```
# let look at Global by country, so we will group the number by 'Province_State' followed by 'Country_R
global_by_country <- global %>%
  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()
global_by_country
```

```
## # A tibble: 306,827 x 7
##    Province_State Country_Region date       cases deaths deaths_per_mill
##    <chr>          <chr>          <date>     <dbl>  <dbl>           <dbl>
##  1 Alberta        Canada         2020-03-06     1      0               0
##  2 Alberta        Canada         2020-03-07     2      0               0
##  3 Alberta        Canada         2020-03-08     4      0               0
##  4 Alberta        Canada         2020-03-09     7      0               0
##  5 Alberta        Canada         2020-03-10     7      0               0
##  6 Alberta        Canada         2020-03-11    19      0               0
##  7 Alberta        Canada         2020-03-12    19      0               0
##  8 Alberta        Canada         2020-03-13    29      0               0
##  9 Alberta        Canada         2020-03-14    29      0               0
## 10 Alberta        Canada         2020-03-15    39      0               0
## # ... with 306,817 more rows, and 1 more variable: Population <dbl>
```

```
# First, we will group the data by the country followed by date.
global_totals <- global_by_country%>%
  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()
global_totals
```

```
## # A tibble: 214,113 x 6
##    Country_Region date       cases deaths deaths_per_mill Population
##    <chr>          <date>     <dbl>  <dbl>           <dbl>      <dbl>
##  1 Afghanistan    2020-02-24     5      0               0   38928341
##  2 Afghanistan    2020-02-25     5      0               0   38928341
##  3 Afghanistan    2020-02-26     5      0               0   38928341
##  4 Afghanistan    2020-02-27     5      0               0   38928341
##  5 Afghanistan    2020-02-28     5      0               0   38928341
##  6 Afghanistan    2020-02-29     5      0               0   38928341
##  7 Afghanistan    2020-03-01     5      0               0   38928341
##  8 Afghanistan    2020-03-02     5      0               0   38928341
##  9 Afghanistan    2020-03-03     5      0               0   38928341
## 10 Afghanistan    2020-03-04     5      0               0   38928341
## # ... with 214,103 more rows
```

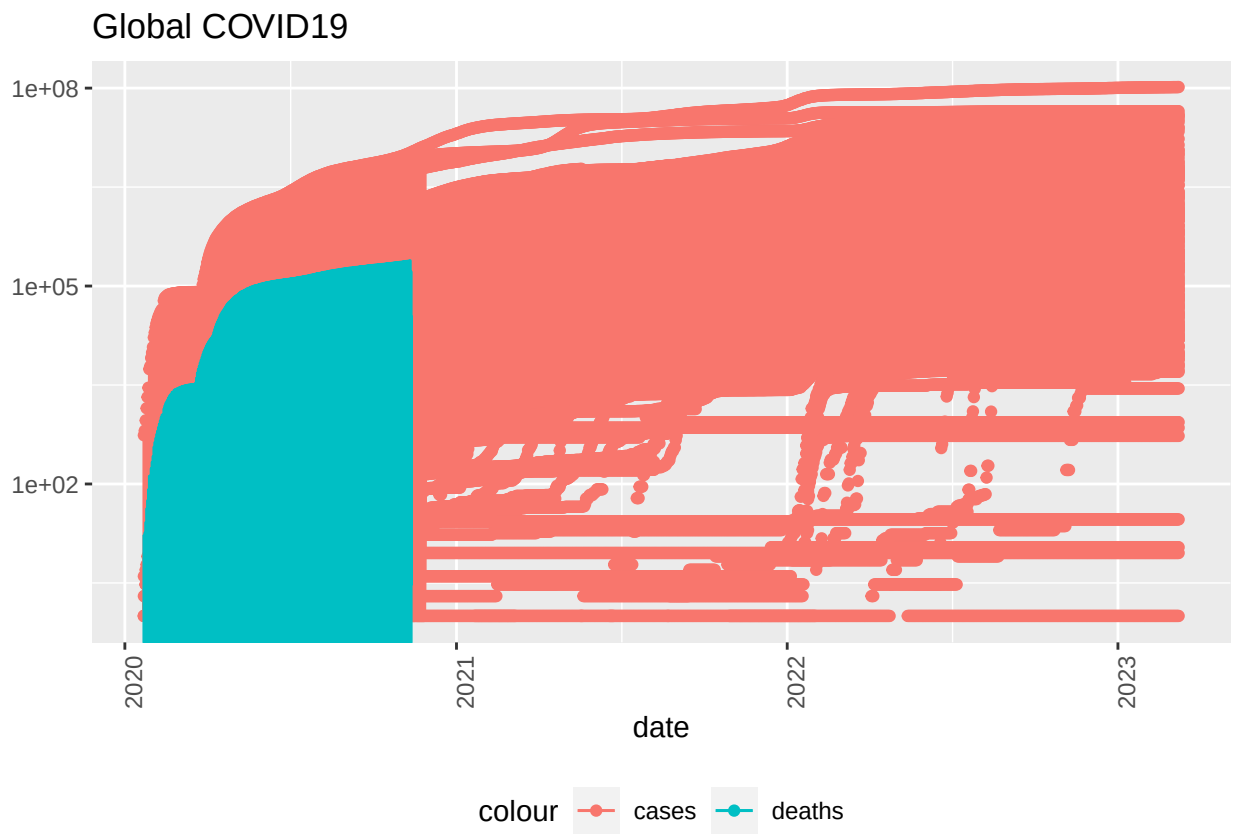
```
tail(global_totals)
```

```
## # A tibble: 6 x 6
##   Country_Region date        cases deaths deaths_per_mill Population
##   <chr>          <date>      <dbl>  <dbl>           <dbl>      <dbl>
## 1 Zimbabwe       2023-03-04 264127   5668            381.   14862927
```

```
## 2 Zimbabwe       2023-03-05 264127   5668            381.   14862927
## 3 Zimbabwe       2023-03-06 264127   5668            381.   14862927
## 4 Zimbabwe       2023-03-07 264127   5668            381.   14862927
## 5 Zimbabwe       2023-03-08 264276   5671            382.   14862927
## 6 Zimbabwe       2023-03-09 264276   5671            382.   14862927
```

```
# To visualize the global_totals and plot the graph between 'cases' and 'death'
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")) +
  scale_y_log10() +
  theme(legend.position="bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Global COVID19", y = NULL)
```

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

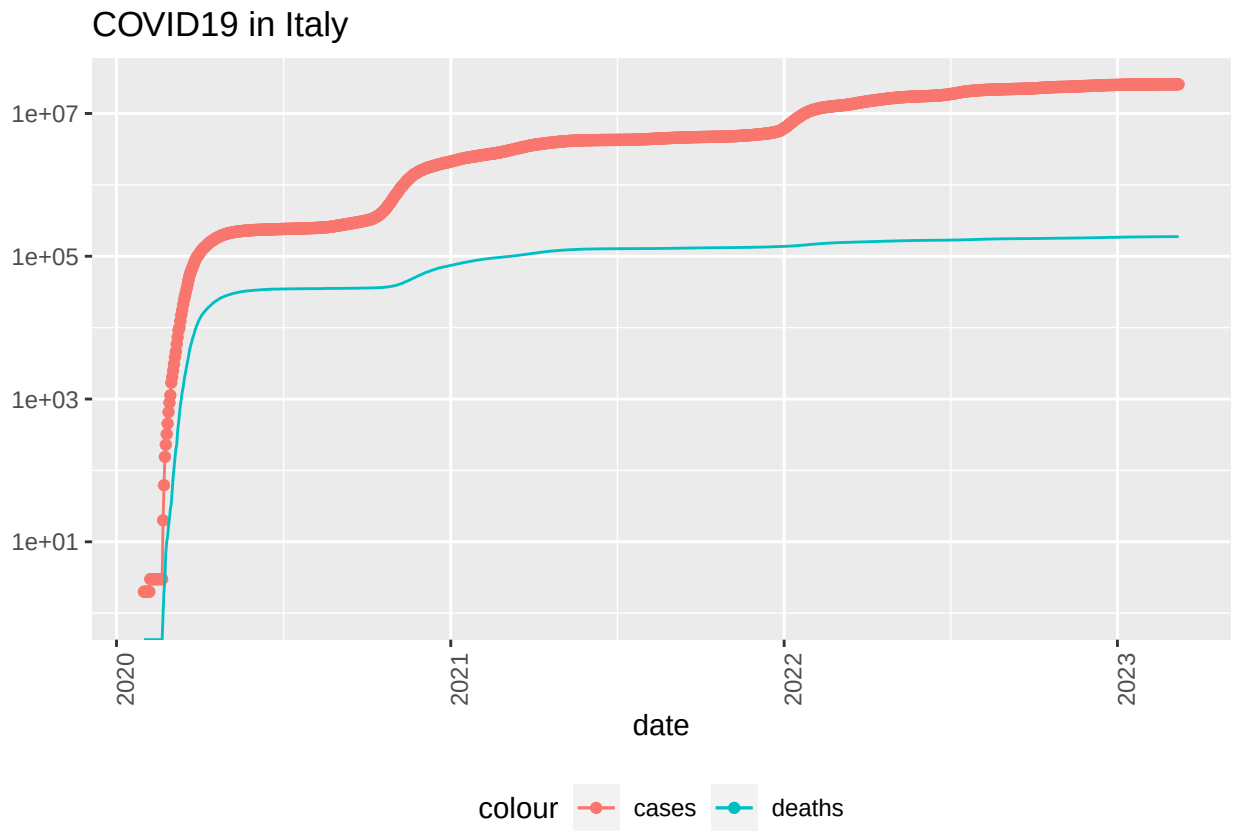


We can plot the graph for some country such as Italy and Canada

```
country <- "Italy"
global_by_country %>%
  filter(Country_Region == country) %>%
  filter(cases > 0) %>%
```

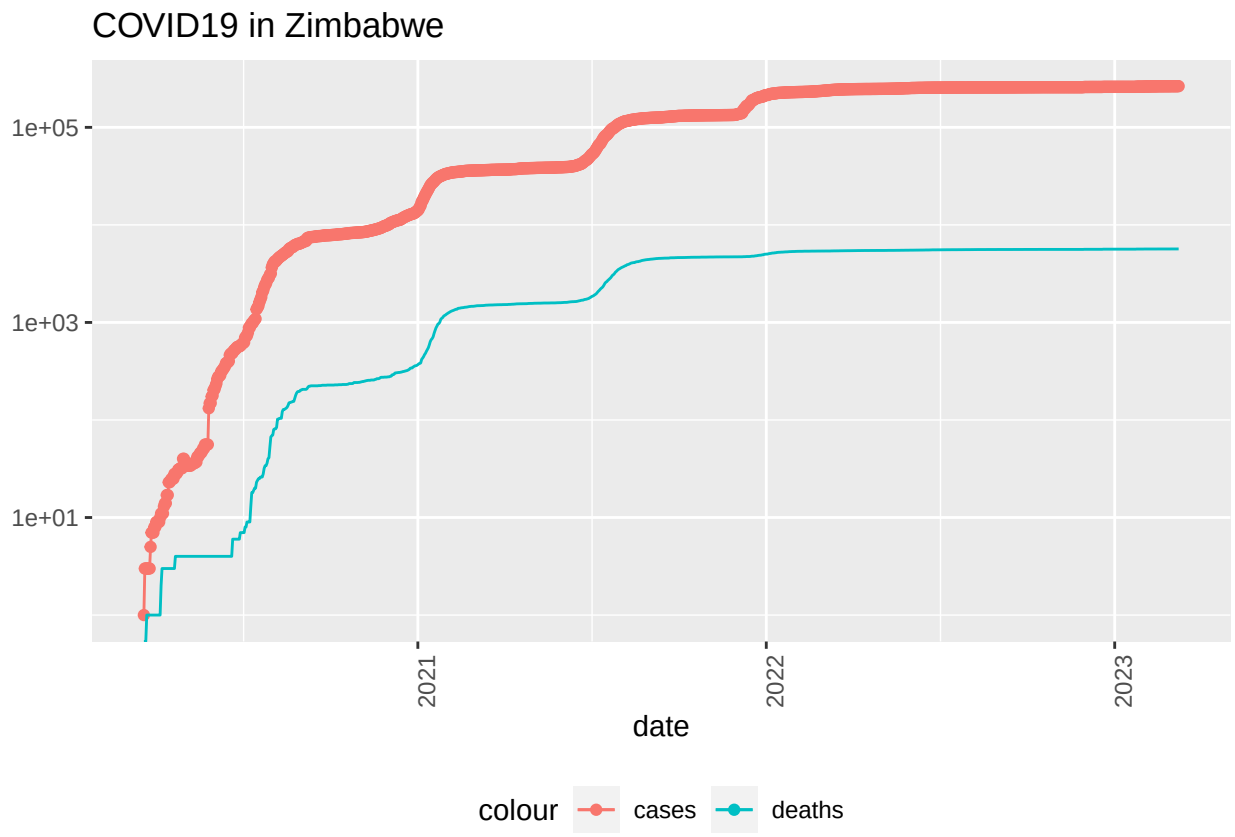
```
  ggplot(aes(x = date, y = cases)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  scale_y_log10() +
  theme(legend.position="bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 in Italy",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

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




```
country <- "Zimbabwe"
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")) +
  scale_y_log10() +
  theme(legend.position="bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 in Zimbabwe", y = NULL)
```

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




```
# Now we want to analyze more to see where the min and max deaths occur
global_country_totals <- global_by_country %>%
  group_by(Country_Region) %>%
  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)

# Use the slice_min to show just the smallest for n = 10
global_country_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 Country_Region deaths cases population
##              <dbl>          <dbl> <chr>           <dbl> <dbl>      <dbl>
##  1        0            35.8       Holy See            0    29        809
##  2        0           240.        Tuvalu              0  2828      11792
##  3        0.000233      0.0000388 Korea, North        6     1   25778815
##  4        0.00320       4.51      Burundi            38 53631   11890781
##  5        0.0118        0.467     Chad              194  7679   16425859
##  6        0.0123        1.64      South Sudan       138 18368   11193729
##  7        0.0130        0.393     Niger             315  9508   24206636
##  8        0.0131        1.86      Tajikistan        125 17786    9537642
##  9        0.0134        2.31      Benin             163 27999   12123198
```

```
## 10        0.0142        0.718     Tanzania          846 42906   59734213
```

```
# Use the slice_max to show just the largest for n = 10
global_country_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 Country_Region   deaths  cases population
##              <dbl>          <dbl> <chr>             <dbl>  <dbl>      <dbl>
##  1            6.66           136. Peru             219539 4.49e6   32971846
##  2            5.50           187. Bulgaria          38228 1.30e6    6948445
##  3            5.05           227. Hungary           48762 2.20e6    9660350
##  4            4.96           122. Bosnia and Herz~  16280 4.02e5    3280815
##  5            4.64           166. North Macedonia    9662 3.47e5    2083380
##  6            4.47           460. Montenegro         2808 2.89e5     628062
##  7            4.38           309. Croatia           17987 1.27e6    4105268
##  8            4.25           458. Georgia           16971 1.83e6    3989175
##  9            3.97           431. Czechia           42491 4.62e6   10708982
## 10            3.87           491. Slovakia          21035 2.67e6    5434712
```

```
summary(global_by_country)
```

```
##  Province_State     Country_Region          date
##  Length:306827      Length:306827      Min.   :2020-01-22
##  Class :character   Class :character   1st Qu.:2020-12-12
##  Mode  :character   Mode  :character   Median :2021-09-16
##                                        Mean   :2021-09-11
##                                        3rd Qu.:2022-06-15
##                                        Max.   :2023-03-09
##
##      cases              deaths        deaths_per_mill
##  Min.   :        1   Min.   :      0   Min.   :   0.000
##  1st Qu.:     1316   1st Qu.:      7   1st Qu.:   4.393
##  Median :    20365   Median :    214   Median : 113.822
##  Mean   :  1032863   Mean   :  14405   Mean   : 595.662
##  3rd Qu.:   271281   3rd Qu.:   3665   3rd Qu.: 822.404
##  Max.   :103802702   Max.   :1123836   Max.   :6658.378
##                                        NA's   :6729
##    Population
##  Min.   :6.700e+01
##  1st Qu.:7.866e+05
##  Median :6.948e+06
##  Mean   :2.890e+07
##  3rd Qu.:2.914e+07
##  Max.   :1.380e+09
##  NA's   :6729
```

```
# Population has 4897 rows of NA. We will just remove it since it is only 2% of the data.
na = c(which(is.na(global_by_country$Population)))
global_by_country_no_na = global_by_country[-na,]
```

```r
# Linear Model
lm_case_population = lm(deaths ~ cases + Population, global_by_country_no_na)
summary(lm_case_population)
```

```
##
## Call:
## lm(formula = deaths ~ cases + Population, data = global_by_country_no_na)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -308907   -2860   -1274   -1017  356989
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.017e+03  5.793e+01   17.56   <2e-16 ***
## cases       1.105e-02  1.221e-05  904.94   <2e-16 ***
## Population  7.048e-05  6.644e-07  106.09   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30250 on 300095 degrees of freedom
## Multiple R-squared:  0.7954, Adjusted R-squared:  0.7954
## F-statistic: 5.834e+05 on 2 and 300095 DF,  p-value: < 2.2e-16
```

```r
lm_case = lm(deaths ~ cases, global_by_country_no_na)
summary(lm_case)
```

```
##
## Call:
## lm(formula = deaths ~ cases, data = global_by_country_no_na)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -324993   -2480   -2423   -2010  357250
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.423e+03  5.744e+01   42.19   <2e-16 ***
## cases       1.165e-02  1.104e-05 1055.39   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 30810 on 300096 degrees of freedom
## Multiple R-squared:  0.7878, Adjusted R-squared:  0.7878
## F-statistic: 1.114e+06 on 1 and 300096 DF,  p-value: < 2.2e-16
```

To conclude

1. We were able to visualize covid 19 cases and deaths for US then we also visialize the covid 19 cases and deadths worlwide

Bias Source

Bias can occured if the data collection was not and if people who died from other sickness was recorded as covid 19 death. Another source was bias is that developping country did not have covid test kit in the early stage of covid 19 that leaad to less record of covid cases. Even when the covid test kit was available on the developping country they was not as much as covid kit as in Country like united state so the number of person tested covid positives was lower than in non developping country. The last bias is that as many people in developping country did not under go covid 19 test while sick there death was not recorded as covid 19 death since most developping country does not have all the ressources required to conduct autposy.