FINAL PROJECT

Oanh

2023-08-22

Covid 19 Data Report

Introduction:

To begin, we need to install these necessary packages:(tidyverse),(lubridate), (ggplot2),(dplyr),(knitr)

```
library(tidyverse)
library(lubridate)
library(ggplot2)
library(knitr)
library(dplyr)
```

Read the data from the link.

```
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_
urls<-str_c(url_in, file_names)
global_cases<-read_csv(urls[1])
```

```
## Rows: 289 Columns: 1147
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1145): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

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

```
## Rows: 289 Columns: 1147
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1145): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

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

```
## Rows: 3342 Columns: 1154
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1148): UID, code3, FIPS, Lat, Long_, 1/22/20, 1/23/20, 1/24/20, 1/25/20...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

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

```
## Rows: 3342 Columns: 1155
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1149): UID, code3, FIPS, Lat, Long_, Population, 1/22/20, 1/23/20, 1/24...
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
head(global_cases)
```

```
## # A tibble: 6 x 1,147
##   ‘Province/State‘ ‘Country/Region‘   Lat  Long ‘1/22/20‘ ‘1/23/20‘ ‘1/24/20‘
##   <chr>            <chr>            <dbl> <dbl>     <dbl>     <dbl>     <dbl>
## 1 <NA>             Afghanistan       33.9 67.7          0         0         0
## 2 <NA>             Albania           41.2 20.2          0         0         0
## 3 <NA>             Algeria           28.0  1.66         0         0         0
## 4 <NA>             Andorra           42.5  1.52         0         0         0
## 5 <NA>             Angola           -11.2 17.9          0         0         0
## 6 <NA>             Antarctica       -71.9 23.3          0         0         0
## # i 1,140 more variables: ‘1/25/20‘ <dbl>, ‘1/26/20‘ <dbl>, ‘1/27/20‘ <dbl>,
## #   ‘1/28/20‘ <dbl>, ‘1/29/20‘ <dbl>, ‘1/30/20‘ <dbl>, ‘1/31/20‘ <dbl>,
## #   ‘2/1/20‘ <dbl>, ‘2/2/20‘ <dbl>, ‘2/3/20‘ <dbl>, ‘2/4/20‘ <dbl>,
## #   ‘2/5/20‘ <dbl>, ‘2/6/20‘ <dbl>, ‘2/7/20‘ <dbl>, ‘2/8/20‘ <dbl>,
## #   ‘2/9/20‘ <dbl>, ‘2/10/20‘ <dbl>, ‘2/11/20‘ <dbl>, ‘2/12/20‘ <dbl>,
## #   ‘2/13/20‘ <dbl>, ‘2/14/20‘ <dbl>, ‘2/15/20‘ <dbl>, ‘2/16/20‘ <dbl>,
## #   ‘2/17/20‘ <dbl>, ‘2/18/20‘ <dbl>, ‘2/19/20‘ <dbl>, ‘2/20/20‘ <dbl>, ...
```

Data Preparation and Cleaning

After looking at global_cases and global_deaths, I would like to tidy those datasets and put each variable (date, cases, deaths) in their 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),name
head(global_cases,10)
```

```
## # A tibble: 10 x 4
##    `Province/State` `Country/Region` date    cases
##    <chr>            <chr>            <chr>   <dbl>
##  1 <NA>             Afghanistan      1/22/20     0
##  2 <NA>             Afghanistan      1/23/20     0
##  3 <NA>             Afghanistan      1/24/20     0
##  4 <NA>             Afghanistan      1/25/20     0
##  5 <NA>             Afghanistan      1/26/20     0
##  6 <NA>             Afghanistan      1/27/20     0
##  7 <NA>             Afghanistan      1/28/20     0
##  8 <NA>             Afghanistan      1/29/20     0
##  9 <NA>             Afghanistan      1/30/20     0
## 10 <NA>             Afghanistan      1/31/20     0
```

```
global_deaths<-global_deaths %>% pivot_longer(cols = -c('Province/State','Country/Region',Lat, Long),nam
```

Combine cases in to deaths per date into one variable we will call global and rename our country region to get rid of slash mark and the same with province sate.

```
global<-global_cases %>% full_join(global_deaths) %>% rename(Country_Region='Country/Region',Province_St
```

```
## Joining with `by = join_by(`Province/State`, `Country/Region`, date)`
```

```
summary(global)
```

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

Filter out and keep only where the cases are positive.

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

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

Check the maximum is a valid maximum or if it were a typo.

```
global %>% filter(cases >100000000)
```

```
## # A tibble: 80 x 5
##    Province_State Country_Region date           cases  deaths
##    <chr>          <chr>          <date>         <dbl>   <dbl>
##  1 <NA>           US             2022-12-20 100050937 1088341
##  2 <NA>           US             2022-12-21 100233060 1089383
##  3 <NA>           US             2022-12-22 100329204 1089979
##  4 <NA>           US             2022-12-23 100368433 1090186
##  5 <NA>           US             2022-12-24 100374955 1090208
##  6 <NA>           US             2022-12-25 100378169 1090223
##  7 <NA>           US             2022-12-26 100390601 1090252
##  8 <NA>           US             2022-12-27 100501536 1090608
##  9 <NA>           US             2022-12-28 100614880 1091598
## 10 <NA>           US             2022-12-29 100718983 1092522
## # i 70 more rows
```

We do the same with US_cases and US_deaths and combine cases in to deaths per date into one variable we will call US.

```
US_cases<-US_cases %>% pivot_longer(cols=-(UID:Combined_Key),names_to = "date", values_to = "cases") %>%
US_deaths<-US_deaths %>% pivot_longer(cols=-(UID:Population),names_to = "date", values_to = "deaths") %
```

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

```
## Joining with ‘by = join_by(Admin2, Province_State, Country_Region,
## Combined_Key, date)‘
```

```
global<- global %>% unite("Combined_Key", c(Province_State, Country_Region), sep=", ",na.rm=TRUE,remove=
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
## # i 306,817 more rows
```

Add population into global.

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

```
## Rows: 4321 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (7): iso2, iso3, FIPS, Admin2, Province_State, Country_Region, Combined_Key
## dbl (5): UID, code3, Lat, Long_, Population
##
## i Use ‘spec()‘ to retrieve the full column specification for this data.
## i Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
```

```
global<-global %>% left_join(uid, by=c("Province_State", "Country_Region")) %>% select(-c(UID,FIPS)) %>
global
```

```
## # A tibble: 306,827 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
## # i 306,817 more rows
```

Visualize US by sate

```
US_by_state<- US %>% group_by(Province_State, Country_Region, date) %>% summarise(cases=sum(cases), dea
```

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

```
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
## # i 66,284 more rows
## # i 1 more variable: Population <dbl>
```

```
US_totals <-US_by_state %>% group_by(Country_Region, date) %>% summarise(cases=sum(cases), deaths=sum(de
```

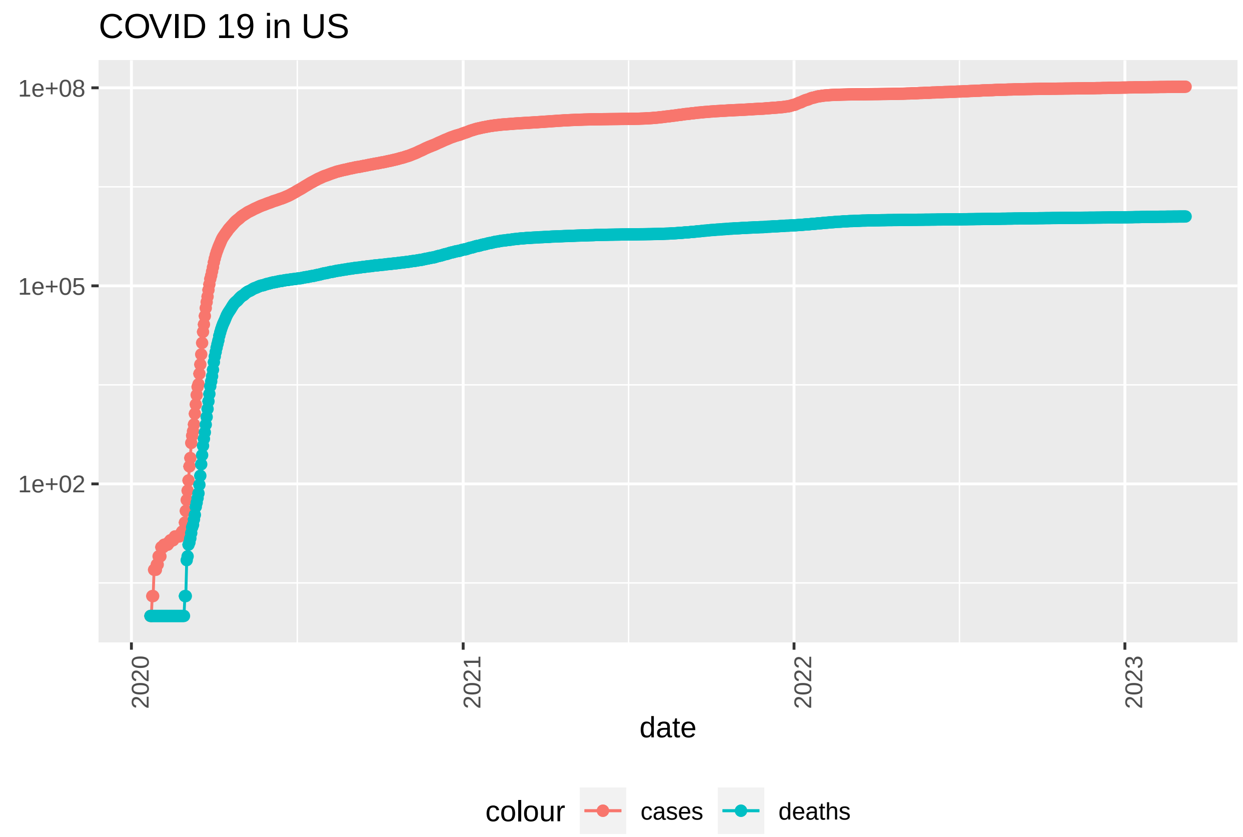
```
## ‘summarise()‘ has grouped output by ’Country_Region’. You can override using
## the ‘.groups‘ argument.
```

```
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
## # i 1,133 more rows
```

Make plot

```
US_totals %>% filter(cases>0) %>% ggplot(aes(x=date, y = cases)) +geom_line(aes(color="cases"))+geom_poi
```

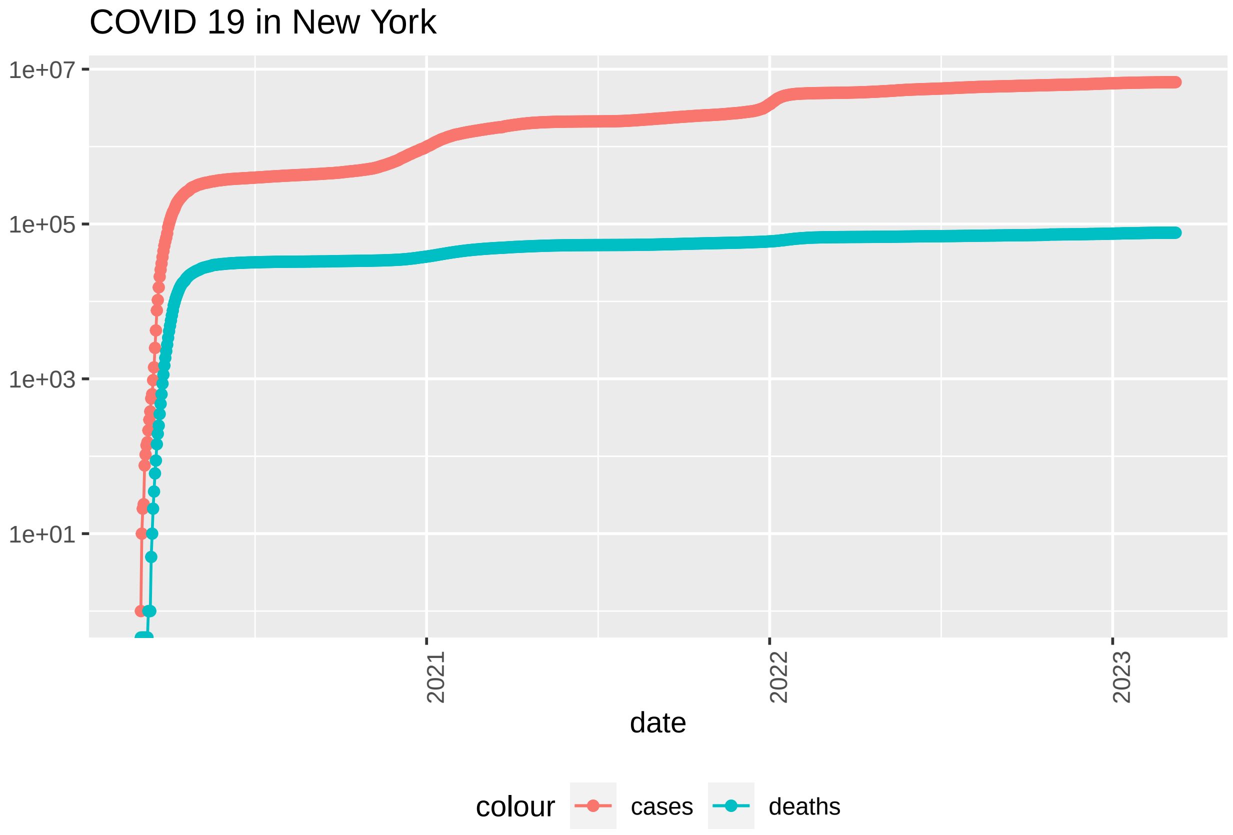


I will do the same plot for New York State

```
state<-"New York"
```

```
US_by_state %>% filter(Province_State==state) %>% filter(cases>0) %>% ggplot(aes(x=date, y = cases)) +g
```

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



## Analyzing about no new cases First trasform our data again by adding new_cases and new_deaths variables

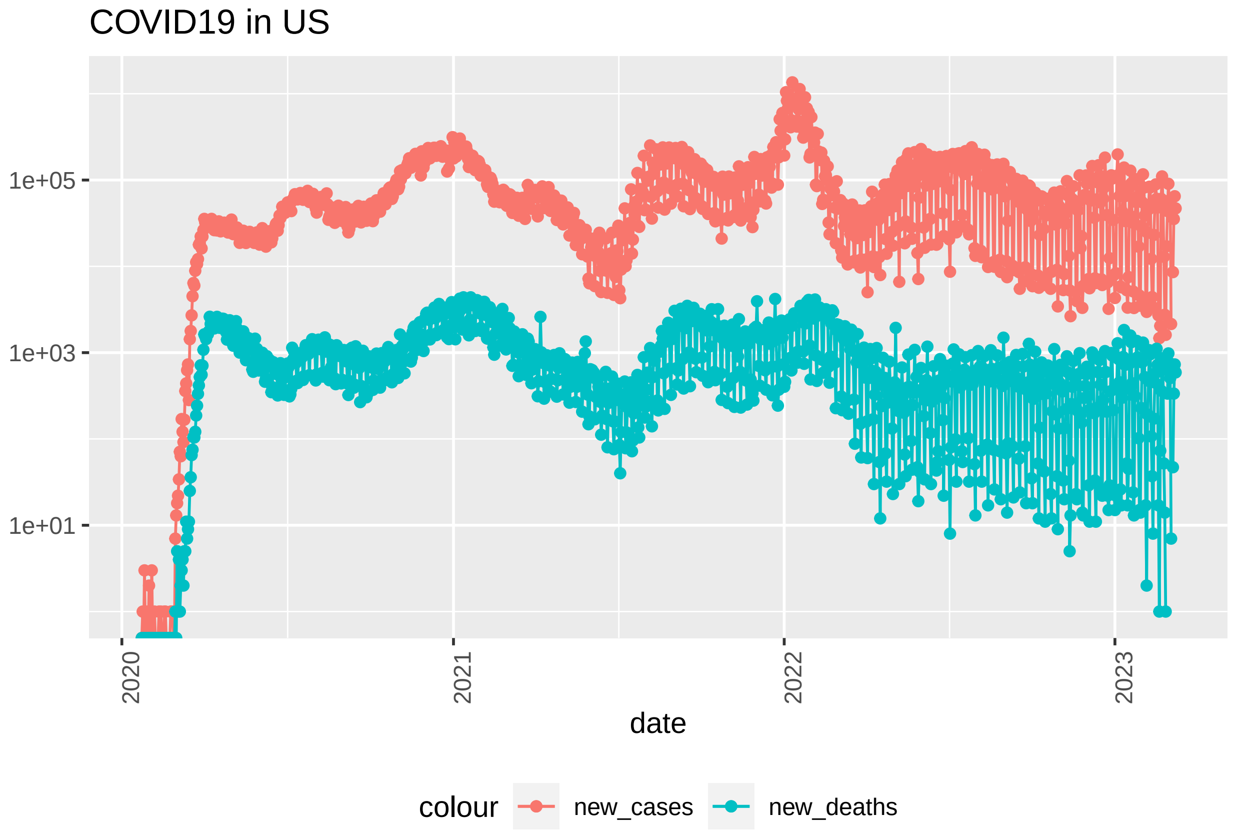
```
US_by_state<-US_by_state %>% mutate(new_cases=cases-lag(cases), new_deaths=deaths-lag(deaths))
US_totals<-US_totals%>% mutate(new_cases=cases-lag(cases), new_deaths=deaths-lag(deaths))
tail(US_totals,10)
```

```
## # A tibble: 10 x 8
##    Country_Region date         cases deaths deaths_per_mill Population new_cases
##    <chr>          <date>       <dbl>  <dbl>           <dbl>      <dbl>     <dbl>
##  1 US             2023-02-28 1.03e8 1.12e6           3364.  332875137     43628
##  2 US             2023-03-01 1.04e8 1.12e6           3367.  332875137     90417
##  3 US             2023-03-02 1.04e8 1.12e6           3370.  332875137     55885
##  4 US             2023-03-03 1.04e8 1.12e6           3371.  332875137     58933
##  5 US             2023-03-04 1.04e8 1.12e6           3371.  332875137      2147
##  6 US             2023-03-05 1.04e8 1.12e6           3371.  332875137     -3862
##  7 US             2023-03-06 1.04e8 1.12e6           3371.  332875137      8564
##  8 US             2023-03-07 1.04e8 1.12e6           3372.  332875137     35371
##  9 US             2023-03-08 1.04e8 1.12e6           3374.  332875137     64861
## 10 US             2023-03-09 1.04e8 1.12e6           3376.  332875137     46931
## # i 1 more variable: new_deaths <dbl>
```

Make plot

```
US_totals %>% ggplot(aes(x=date, y = new_cases)) +geom_line(aes(color="new_cases"))+geom_point(aes(colo
```

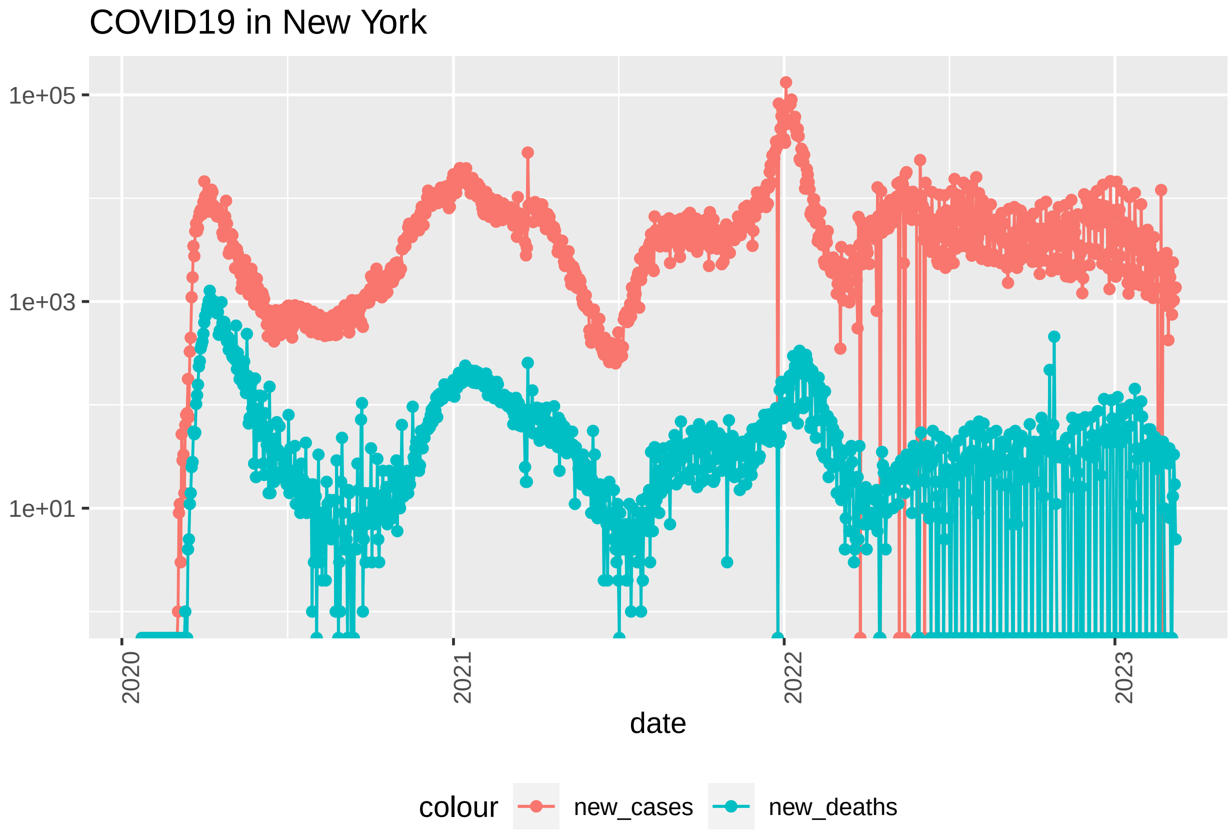
```
## Warning in self$trans$transform(x): NaNs produced

## 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 containing missing values (`geom_line()`).

## Warning: Removed 2 rows containing missing values (`geom_point()`).

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

## Warning: Removed 4 rows containing missing values (`geom_point()`).
```



I will do the same plot for New York State

```
state<-"New York"
US_by_state %>%filter(Province_State==state)%>% ggplot(aes(x=date, y = new_cases)) +geom_line(aes(color=
```

```
## 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 containing missing values (‘geom_line()‘).

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

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

## Warning: Removed 9 rows containing missing values (‘geom_point()‘).
```



## Analyzing the worst and the best state

```
US_state_totals<-US_by_state %>% group_by(Province_State) %>% summarise(deaths=max(deaths), cases=max(ca
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 American Samoa            34 8.32e3      55641           150.           0.611
##  2 Northern Mariana Isl~     41 1.37e4      55144           248.           0.744
##  3 Virgin Islands           130 2.48e4     107268           231.           1.21
##  4 Hawaii                  1841 3.81e5    1415872           269.           1.30
##  5 Vermont                  929 1.53e5     623989           245.           1.49
##  6 Puerto Rico             5823 1.10e6    3754939           293.           1.55
##  7 Utah                    5298 1.09e6    3205958           340.           1.65
##  8 Alaska                  1486 3.08e5     740995           415.           2.01
##  9 District of Columbia    1432 1.78e5     705749           252.           2.03
## 10 Washington             15683 1.93e6    7614893           253.           2.06
```

The best state is American Samoa

```
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 Arizona         33102 2443514    7278717           336.            4.55
##  2 Oklahoma        17972 1290929    3956971           326.            4.54
##  3 Mississippi     13370  990756    2976149           333.            4.49
##  4 West Virginia    7960  642760    1792147           359.            4.44
##  5 New Mexico       9061  670929    2096829           320.            4.32
##  6 Arkansas        13020 1006883    3017804           334.            4.31
##  7 Alabama         21032 1644533    4903185           335.            4.29
##  8 Tennessee       29263 2515130    6829174           368.            4.28
##  9 Michigan        42205 3064125    9986857           307.            4.23
## 10 Kentucky        18130 1718471    4467673           385.            4.06
```

The worst state is Arizona

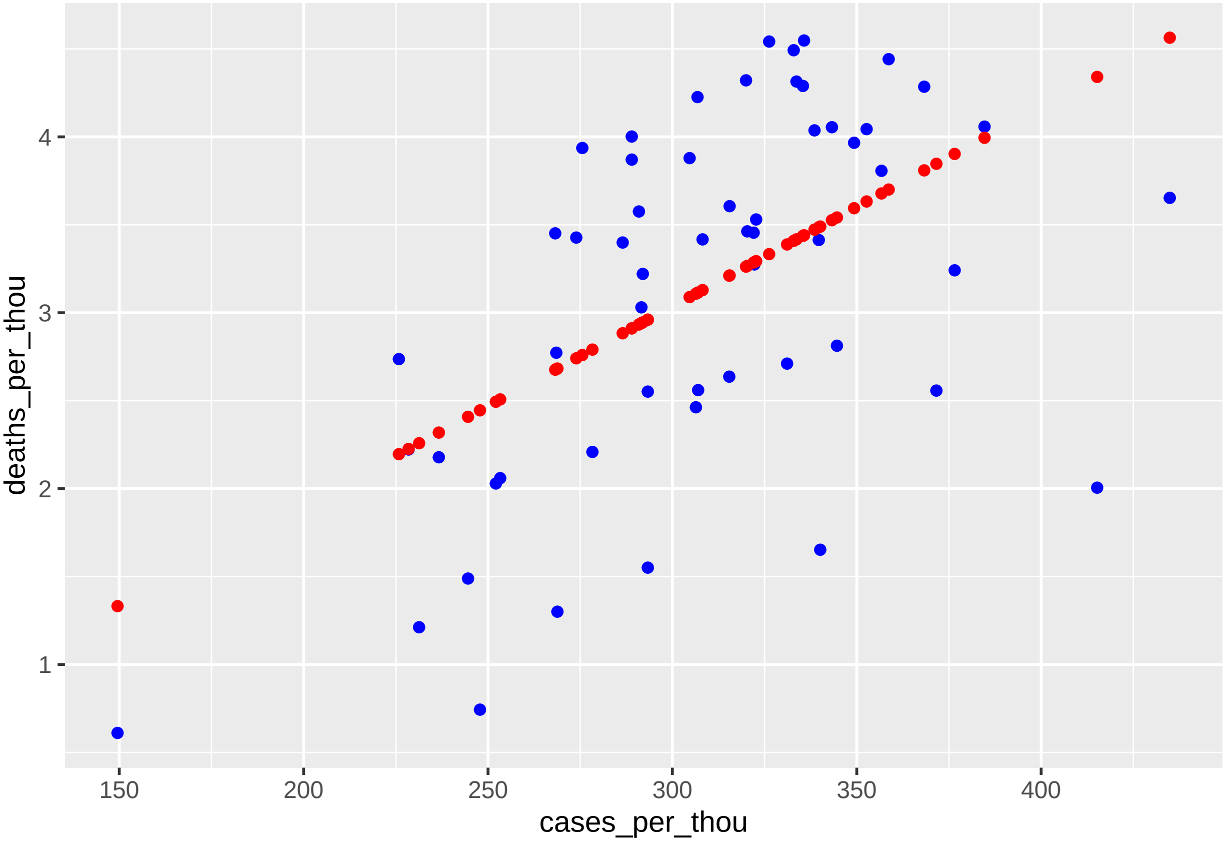
##Modeling

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

```
##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou, data = US_state_totals)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3352 -0.5978  0.1491  0.6535  1.2086
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.36167    0.72480  -0.499     0.62
## cases_per_thou  0.01133    0.00232   4.881 9.76e-06 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 0.8615 on 54 degrees of freedom
## Multiple R-squared:  0.3061, Adjusted R-squared:  0.2933
## F-statistic: 23.82 on 1 and 54 DF,  p-value: 9.763e-06
```

Make plot

```
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_poi
```



Based on the linear regression results, we can conclude that there is a positive relationship between the number of cases per thousand and the number of deaths per thousand. In other words, as the number of cases per thousand increases, the number of deaths per thousand increases also . The R-squared value of 0.2933 indicates that the model explains approximately 29% of the variability in the number of deaths per thoudand. However, it’s important to note that correlation does not imply causation, and there may be other factors that contribute to the number deaths beyond just number of cases .