# Analyzing COVID19 Data

Saurabh Sant

12/07/2021

## Data cleaning and transforming

I will start by reading in the data from the four main CSV files. Get current data in the four files

```
library(stringr)
library(readr)
library(tidyverse)
```

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

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.7
## v tidyr   1.1.3     v forcats 0.5.1

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

```
library(lubridate)
```

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

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

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

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

After looking at global_cases and global_death, I would like to tidy those datasets and put each variable (date, cases, deaths) in their 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") %>%
  select(-c(Lat, Long))
global_cases
```

```
## # A tibble: 150,381 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
## # ... with 150,371 more rows
```

```
global_deaths <- global_deaths %>%
  pivot_longer(cols = -c('Province/State',
                         'Country/Region', Lat, Long),
               names_to = "date",
               values_to = "deaths") %>%
  select(-c(Lat, Long))
global_deaths
```

```
## # A tibble: 150,381 x 4
##    `Province/State` `Country/Region` date    deaths
##    <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
## # ... with 150,371 more rows
```

```
global <- global_cases %>%
  full_join(global_deaths) %>%
```

```
  rename(Country_Region = 'Country/Region',
         Province_State = 'Province/State') %>%
  mutate(date = mdy(date))
```

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

```
global
```

```
## # A tibble: 150,381 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 150,371 more rows
```

```
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:150381      Length:150381      Min.   :2020-01-22   Min.   :       0
##  Class :character   Class :character   1st Qu.:2020-06-04   1st Qu.:     103
##  Mode  :character   Mode  :character   Median :2020-10-17   Median :    1624
##                                        Mean   :2020-10-17   Mean   :  228116
##                                        3rd Qu.:2021-03-01   3rd Qu.:   36575
##                                        Max.   :2021-07-13   Max.   :33915385
##      deaths
##  Min.   :     0
##  1st Qu.:     1
##  Median :    25
##  Mean   :  5454
##  3rd Qu.:   623
##  Max.   :607784
```

Only use countries where cases are positive ($> 0$).

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

```
## # A tibble: 134,469 x 5
##    Province_State Country_Region date       cases deaths
##    <chr>          <chr>          <date>     <dbl>  <dbl>
##  1 <NA>           Afghanistan    2020-02-24     1      0
##  2 <NA>           Afghanistan    2020-02-25     1      0
##  3 <NA>           Afghanistan    2020-02-26     1      0
```

```
##  4 <NA>           Afghanistan    2020-02-27     1      0
##  5 <NA>           Afghanistan    2020-02-28     1      0
##  6 <NA>           Afghanistan    2020-02-29     1      0
##  7 <NA>           Afghanistan    2020-03-01     1      0
##  8 <NA>           Afghanistan    2020-03-02     1      0
##  9 <NA>           Afghanistan    2020-03-03     2      0
## 10 <NA>           Afghanistan    2020-03-04     4      0
## # ... with 134,459 more rows
```

Now, I will tidy and transform the COVID-19 data on cases and deaths in the US.

```
US_cases <- US_cases %>%
  pivot_longer(cols = -('UID':Combined_Key),
               names_to = "date",
               values_to = "cases")%>%
  select(Admin2:cases) %>%
  mutate(date = mdy(date)) %>%
  select(-c(Lat, Long_))
US_cases
```

```
## # A tibble: 1,801,338 x 6
##    Admin2  Province_State Country_Region Combined_Key         date       cases
##    <chr>   <chr>          <chr>          <chr>                <date>     <dbl>
##  1 Autauga Alabama        US             Autauga, Alabama, US 2020-01-22     0
##  2 Autauga Alabama        US             Autauga, Alabama, US 2020-01-23     0
##  3 Autauga Alabama        US             Autauga, Alabama, US 2020-01-24     0
##  4 Autauga Alabama        US             Autauga, Alabama, US 2020-01-25     0
##  5 Autauga Alabama        US             Autauga, Alabama, US 2020-01-26     0
##  6 Autauga Alabama        US             Autauga, Alabama, US 2020-01-27     0
##  7 Autauga Alabama        US             Autauga, Alabama, US 2020-01-28     0
##  8 Autauga Alabama        US             Autauga, Alabama, US 2020-01-29     0
##  9 Autauga Alabama        US             Autauga, Alabama, US 2020-01-30     0
## 10 Autauga Alabama        US             Autauga, Alabama, US 2020-01-31     0
## # ... with 1,801,328 more rows
```

```
US_death <- US_death %>%
  pivot_longer(cols = -('UID':Population),
               names_to = "date",
               values_to = "deaths")%>%
  select(Admin2:deaths) %>%
  mutate(date = mdy(date)) %>%
  select(-c(Lat, Long_))

US_death
```

```
## # A tibble: 1,801,338 x 7
##    Admin2  Province_State Country_Region Combined_Key      Population date
##    <chr>   <chr>          <chr>          <chr>                 <dbl> <date>
##  1 Autauga Alabama        US             Autauga, Alabama~      55869 2020-01-22
##  2 Autauga Alabama        US             Autauga, Alabama~      55869 2020-01-23
##  3 Autauga Alabama        US             Autauga, Alabama~      55869 2020-01-24
##  4 Autauga Alabama        US             Autauga, Alabama~      55869 2020-01-25
##  5 Autauga Alabama        US             Autauga, Alabama~      55869 2020-01-26
```

```
##  6 Autauga Alabama        US             Autauga, Alabama~      55869 2020-01-27
##  7 Autauga Alabama        US             Autauga, Alabama~      55869 2020-01-28
##  8 Autauga Alabama        US             Autauga, Alabama~      55869 2020-01-29
##  9 Autauga Alabama        US             Autauga, Alabama~      55869 2020-01-30
## 10 Autauga Alabama        US             Autauga, Alabama~      55869 2020-01-31
## # ... with 1,801,328 more rows, and 1 more variable: deaths <dbl>
```

Population and other variables are not in US_cases dataset however those variables are present in US_death dataset. So, Let's combine both us_cases and us_death tables to make a one dataset with all of the data.

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

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

```
by = c("Admin2", "Province_State", "Country_region", "Combined_Key", "date")

US
```

```
## # A tibble: 1,801,338 x 8
##    Admin2 Province_State Country_Region Combined_Key date       cases Population
##    <chr>  <chr>          <chr>          <chr>        <date>     <dbl>      <dbl>
##  1 Autau~ Alabama        US             Autauga, Al~ 2020-01-22     0      55869
##  2 Autau~ Alabama        US             Autauga, Al~ 2020-01-23     0      55869
##  3 Autau~ Alabama        US             Autauga, Al~ 2020-01-24     0      55869
##  4 Autau~ Alabama        US             Autauga, Al~ 2020-01-25     0      55869
##  5 Autau~ Alabama        US             Autauga, Al~ 2020-01-26     0      55869
##  6 Autau~ Alabama        US             Autauga, Al~ 2020-01-27     0      55869
##  7 Autau~ Alabama        US             Autauga, Al~ 2020-01-28     0      55869
##  8 Autau~ Alabama        US             Autauga, Al~ 2020-01-29     0      55869
##  9 Autau~ Alabama        US             Autauga, Al~ 2020-01-30     0      55869
## 10 Autau~ Alabama        US             Autauga, Al~ 2020-01-31     0      55869
## # ... with 1,801,328 more rows, and 1 more variable: deaths <dbl>
```

Now, I will do the same for the global data so we can compare the data across countries as well. So, now I need to add population for each country and I find that same Johns Hopkins github has a CSV.

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

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_character(),
##   Admin2 = col_character(),
##   Province_State = col_character(),
##   Country_Region = col_character(),
##   Lat = col_double(),
##   Long_ = col_double(),
##   Combined_Key = col_character(),
##   Population = col_double()
## )
```

```
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)
```

## Data visualization

Summary of the data we have so far.

```
summary(US)
```

```
##     Admin2          Province_State     Country_Region     Combined_Key
##  Length:1801338     Length:1801338     Length:1801338     Length:1801338
##  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.   :      0   Min.   :       0   Min.   :    0.00
##  1st Qu.:2020-06-04   1st Qu.:     18   1st Qu.:    9917   1st Qu.:    0.00
##  Median :2020-10-17   Median :    437   Median :   24892   Median :    7.00
##  Mean   :2020-10-17   Mean   :   4196   Mean   :   99604   Mean   :   83.65
##  3rd Qu.:2021-03-01   3rd Qu.:   2167   3rd Qu.:   64979   3rd Qu.:   42.00
##  Max.   :2021-07-13   Max.   :1259992   Max.   :10039107   Max.   :24559.00
```

```
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:134469      Length:134469      Min.   :2020-01-22   Min.   :       1
##  Class :character   Class :character   1st Qu.:2020-07-05   1st Qu.:     288
##  Mode  :character   Mode  :character   Median :2020-11-09   Median :    3089
##                                        Mean   :2020-11-07   Mean   :  255110
##                                        3rd Qu.:2021-03-13   3rd Qu.:   53832
##                                        Max.   :2021-07-13   Max.   :33915385
##
##      deaths         Population        Combined_Key
```

```
##  Min.   :     0   Min.   :8.090e+02   Length:134469
##  1st Qu.:     3   1st Qu.:9.775e+05   Class :character
##  Median :    52   Median :7.497e+06   Mode  :character
##  Mean   :  6100   Mean   :3.006e+07
##  3rd Qu.:   880   3rd Qu.:3.102e+07
##  Max.   :607784   Max.   :1.380e+09
##                   NA's   :1778
```

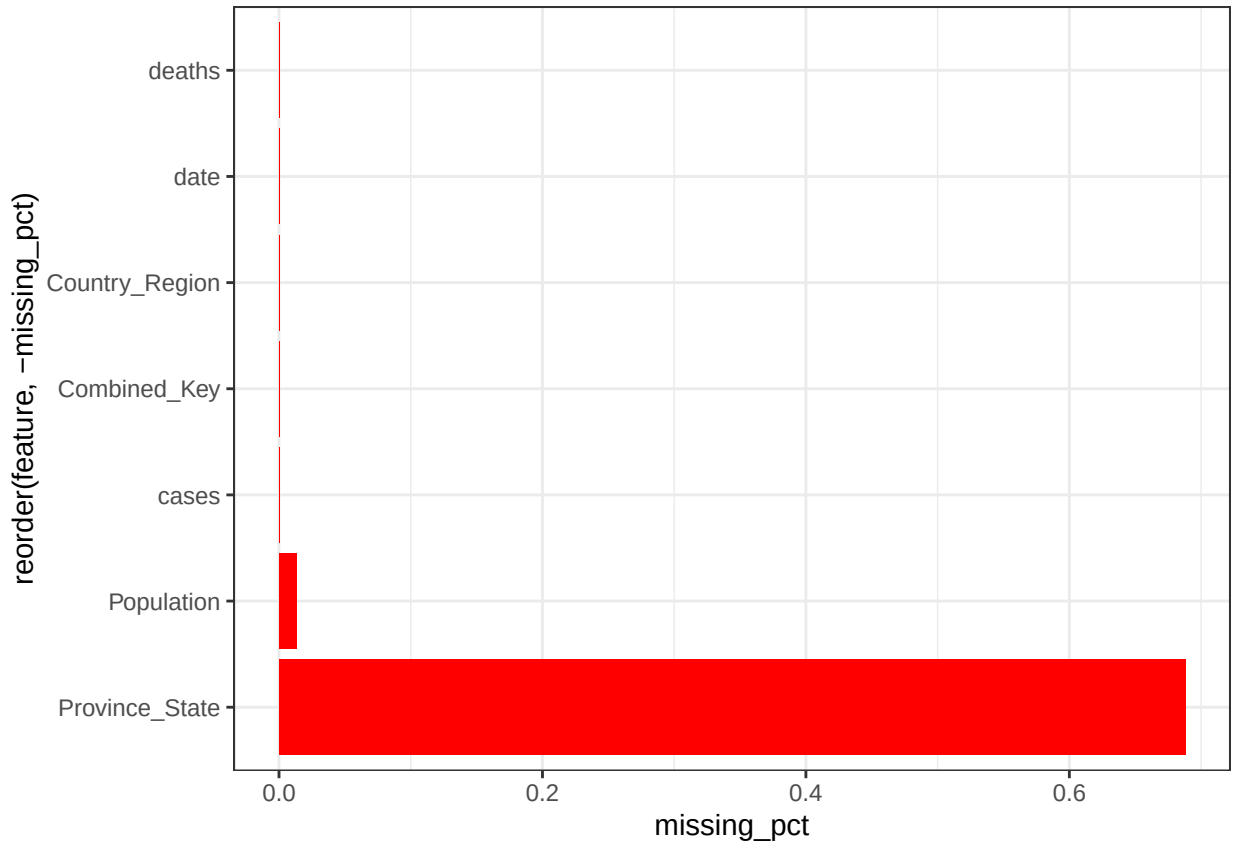
How many missing values are there for each variable? From the graph, we can see that there is a lot of missing data for Province/State as most countries do not have provinces.

```
missing_values <- global %>% summarize_each(funs(sum(is.na(.))/n()))
```

```
## Warning: `summarise_each_()` was deprecated in dplyr 0.7.0.
## Please use `across()` instead.
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
```

```
missing_values <- gather(missing_values, key="feature", value="missing_pct")
missing_values %>%
  ggplot(aes(x=reorder(feature,-missing_pct),y=missing_pct)) +
  geom_bar(stat="identity",fill="red")+
  coord_flip()+theme_bw()
```




```
by_countries <- global %>%
  group_by(Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths), Population = sum(Population)) %>%
  mutate(deaths_per_mil = deaths*1000000/Population)%>%
  select(Country_Region, date, cases, deaths, deaths_per_mil, Population)%>%
  ungroup()
```

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

```
by_countries
```

```
## # A tibble: 94,104 x 6
##    Country_Region date       cases deaths deaths_per_mil Population
##    <chr>          <date>     <dbl>  <dbl>          <dbl>      <dbl>
##  1 Afghanistan    2020-02-24     1      0              0   38928341
##  2 Afghanistan    2020-02-25     1      0              0   38928341
##  3 Afghanistan    2020-02-26     1      0              0   38928341
##  4 Afghanistan    2020-02-27     1      0              0   38928341
##  5 Afghanistan    2020-02-28     1      0              0   38928341
##  6 Afghanistan    2020-02-29     1      0              0   38928341
##  7 Afghanistan    2020-03-01     1      0              0   38928341
##  8 Afghanistan    2020-03-02     1      0              0   38928341
##  9 Afghanistan    2020-03-03     2      0              0   38928341
## 10 Afghanistan    2020-03-04     4      0              0   38928341
## # ... with 94,094 more rows
```

Using the US data set, I will group by state and by region. Then, I will summarize by summing the cases and deaths by states since each state had multiple counties.

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

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

```
US_by_state
```

```
## # A tibble: 31,262 x 7
##    Province_State Country_Region date       cases deaths deaths_per_mil
##    <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 31,252 more rows, and 1 more variable: Population <dbl>
```

Now lets group the US_by_state dataset by country region

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

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

```
US_totals
```

```
## # A tibble: 539 x 6
##    Country_Region date       cases deaths deaths_per_mil 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
```

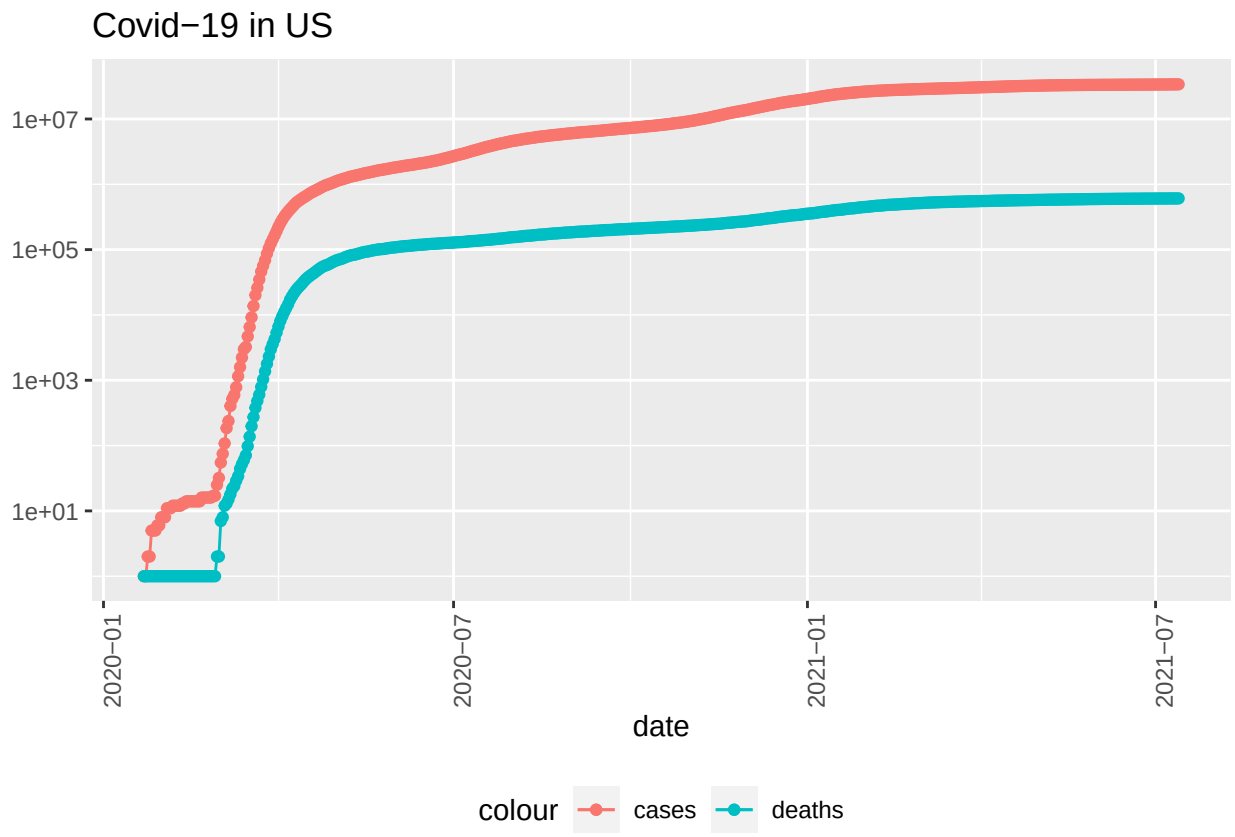
```
##  7 US             2020-01-28     5      1        0.00300  332875137
##  8 US             2020-01-29     6      1        0.00300  332875137
##  9 US             2020-01-30     6      1        0.00300  332875137
## 10 US             2020-01-31     8      1        0.00300  332875137
## # ... with 529 more rows
```

```
tail(US_totals)
```

```
## # A tibble: 6 x 6
##   Country_Region date          cases deaths deaths_per_mil Population
##   <chr>          <date>        <dbl>  <dbl>          <dbl>      <dbl>
## 1 US             2021-07-08 33790505 606489          1822.  332875137
## 2 US             2021-07-09 33838746 606993          1823.  332875137
## 3 US             2021-07-10 33847784 607132          1824.  332875137
## 4 US             2021-07-11 33853948 607156          1824.  332875137
## 5 US             2021-07-12 33888961 607399          1825.  332875137
## 6 US             2021-07-13 33915385 607784          1826.  332875137
```

Lets visualize the cases and deaths in the US and see how they have been trending over time.

```
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 = "Covid-19 in US", y = NULL)
```

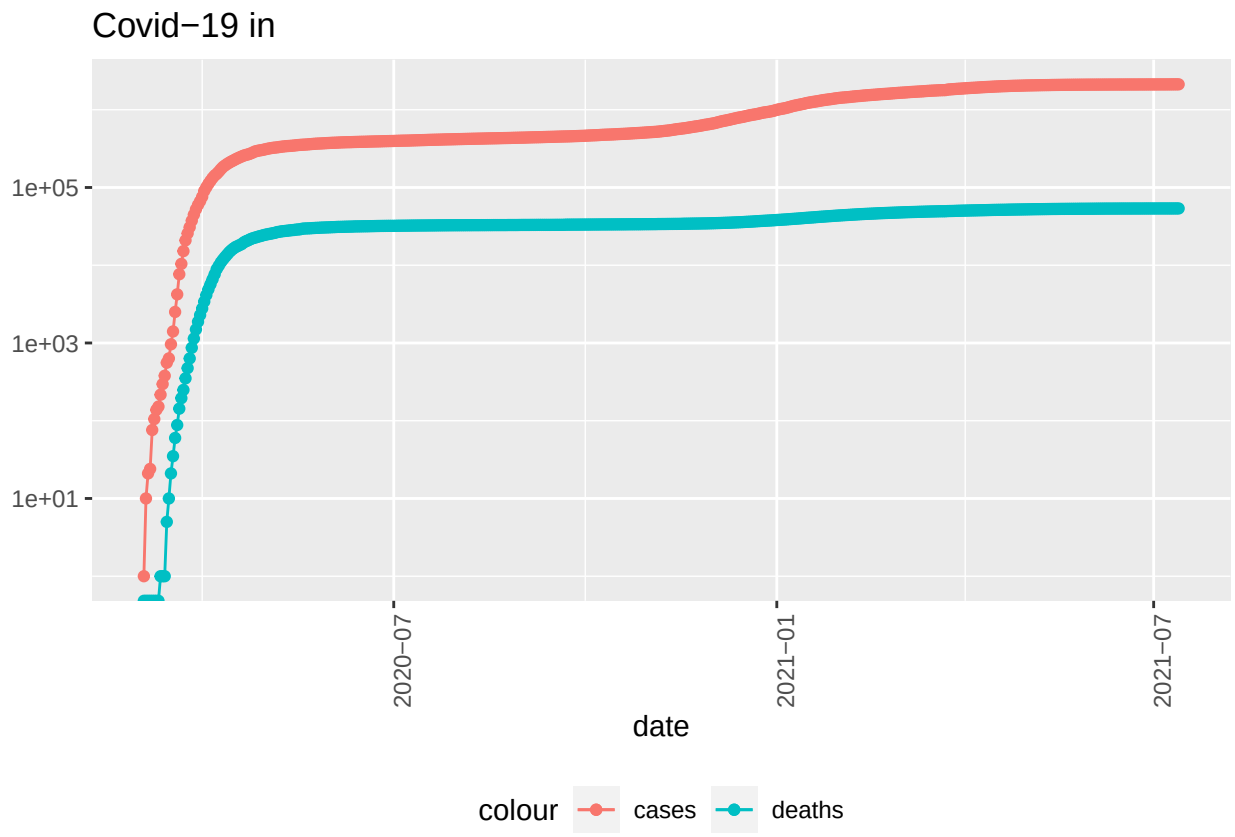



Looking at the same graph for New York State.

```
US_by_state %>%
  filter(Province_State == "New York") %>%
  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 = "Covid-19 in ", y = NULL)
```

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

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




Based on our graphs, it appears that the COVID cases have leveled off which raises some questions. Is the number of new cases flat? So, we will further transform and analyze the data to test our hypothesis.

## Data Analysis

```
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))
head(US_totals)
```

```
## # A tibble: 6 x 8
##   Country_Region date       cases deaths deaths_per_mil Population new_cases
##   <chr>          <date>     <dbl>  <dbl>          <dbl>      <dbl>     <dbl>
## 1 US             2020-01-22     1      1        0.00300  332875137        NA
## 2 US             2020-01-23     1      1        0.00300  332875137         0
## 3 US             2020-01-24     2      1        0.00300  332875137         1
## 4 US             2020-01-25     2      1        0.00300  332875137         0
## 5 US             2020-01-26     5      1        0.00300  332875137         3
## 6 US             2020-01-27     5      1        0.00300  332875137         0
## # ... with 1 more variable: new_deaths <dbl>
```

```
tail(US_totals)
```

```
## # A tibble: 6 x 8
##   Country_Region date          cases deaths deaths_per_mil Population new_cases
##   <chr>          <date>        <dbl>  <dbl>          <dbl>      <dbl>     <dbl>
## 1 US             2021-07-08 33790505 606489          1822.  332875137     20061
## 2 US             2021-07-09 33838746 606993          1823.  332875137     48241
## 3 US             2021-07-10 33847784 607132          1824.  332875137      9038
## 4 US             2021-07-11 33853948 607156          1824.  332875137      6164
## 5 US             2021-07-12 33888961 607399          1825.  332875137     35013
## 6 US             2021-07-13 33915385 607784          1826.  332875137     26424
## # ... with 1 more variable: new_deaths <dbl>
```

Now, we will graph with the new variables (new_cases, new_deaths) to see the change in cases and deaths over each day.

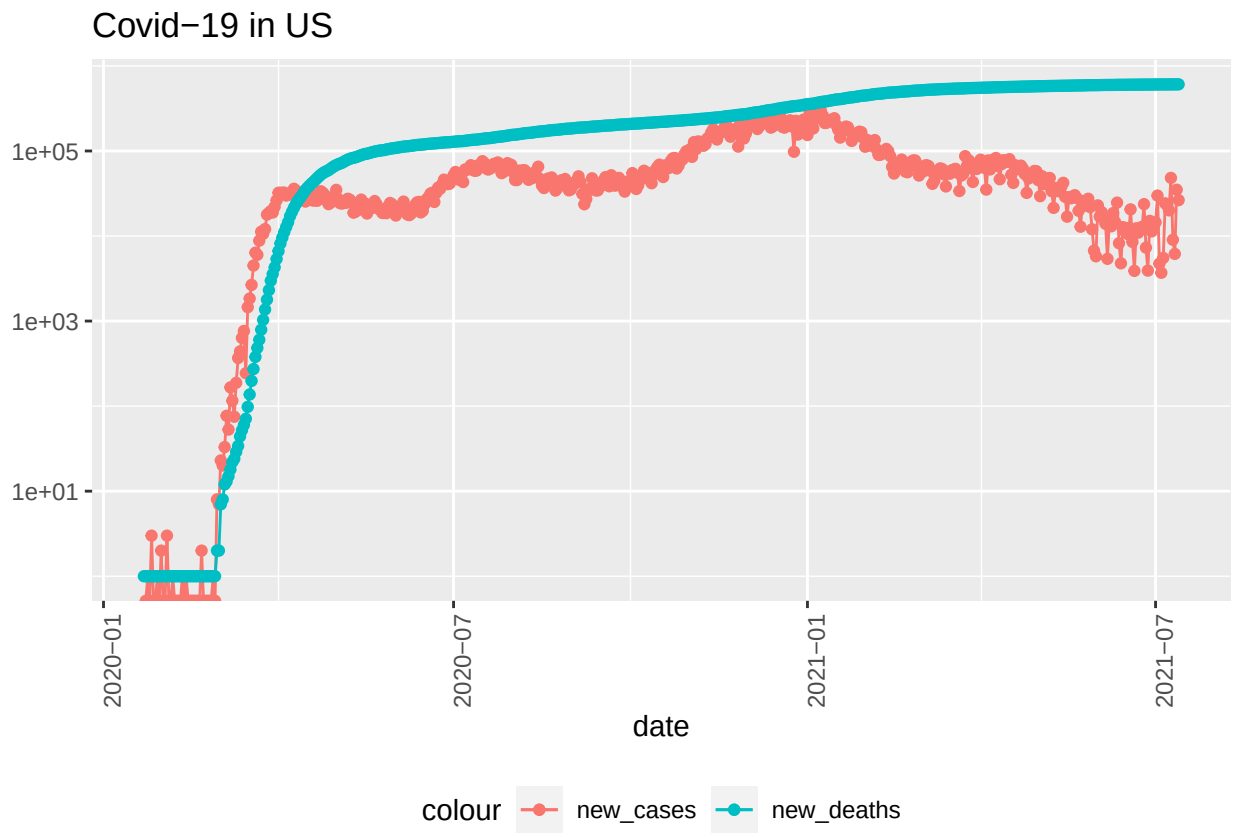
```
US_totals %>%
  ggplot(aes(x = date, y = new_cases)) +
  geom_line(aes(color = "new_cases")) +
  geom_point(aes(color = "new_cases")) +
  geom_line(aes(y = deaths, color = "new_deaths")) +
  geom_point(aes(y = deaths, color = "new_deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom", axis.text.x = element_text(angle = 90)) +
  labs(title = "Covid-19 in US", y = NULL)
```

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

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

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

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

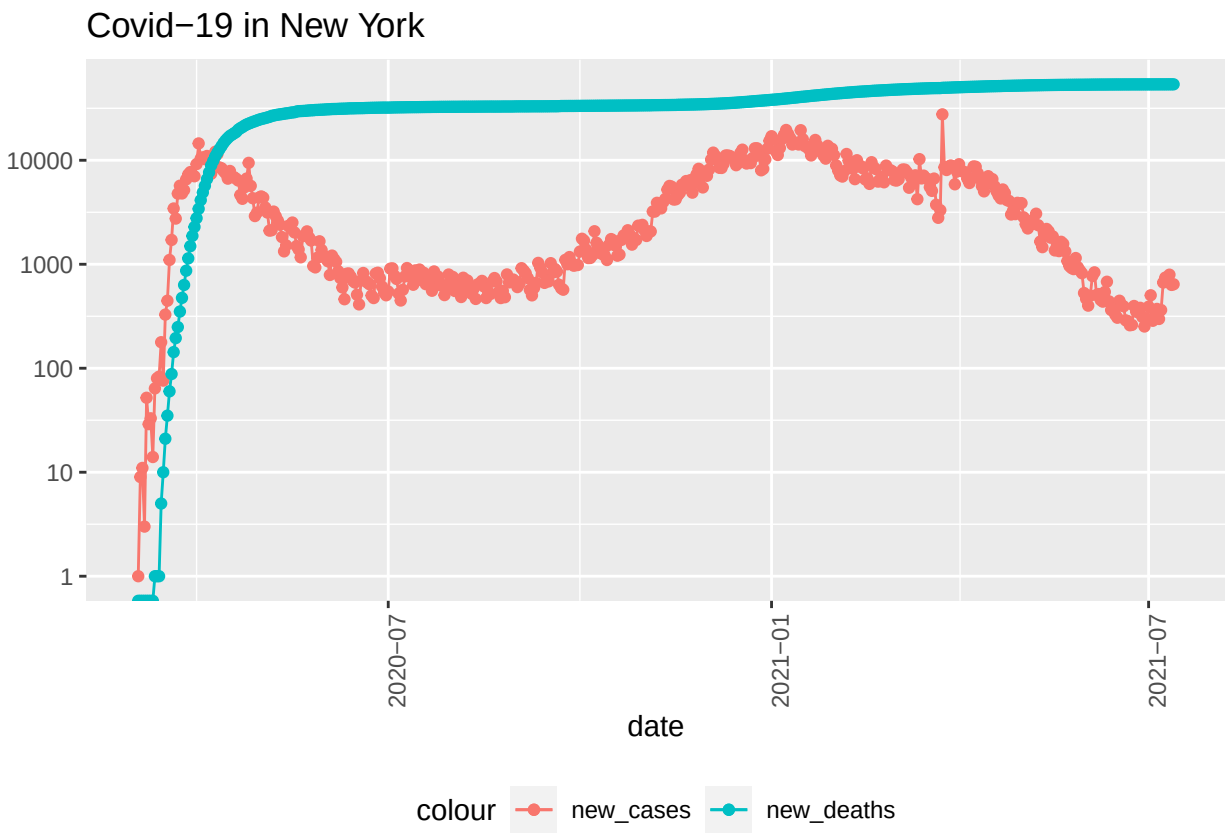



Analyzing the changes in COVID-19 cases and deaths in New York. After the transormation, we are able to see the fluctuations in COVID-19 cases over time.

```
state <- "New York"
US_by_state %>%
  filter(Province_State == "New York") %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = new_cases)) +
  geom_line(aes(color = "new_cases")) +
  geom_point(aes(color = "new_cases")) +
  geom_line(aes(y = deaths, color = "new_deaths")) +
  geom_point(aes(y = deaths, color = "new_deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom", axis.text.x = element_text(angle = 90)) +
  labs(title = str_c("Covid-19 in ", state), y = NULL)
```

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

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




What are the best and the worst states? best and worst countries? To measure this we will look at cases and deaths per 1,000 people.

```
country_totals <- global %>%
  group_by(Country_Region) %>%
  summarise(death = max(deaths), cases = max(cases),
            deaths_per_thou = 1000*deaths/Population,
          cases_per_thou = 1000*cases/Population) %>%
  filter(cases > 0)
```

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

```
country_totals
```

```
## # A tibble: 134,469 x 5
## # Groups:   Country_Region [194]
##    Country_Region death  cases deaths_per_thou cases_per_thou
##    <chr>          <dbl>  <dbl>           <dbl>          <dbl>
##  1 Afghanistan     5791 134653               0           3.46
##  2 Afghanistan     5791 134653               0           3.46
##  3 Afghanistan     5791 134653               0           3.46
##  4 Afghanistan     5791 134653               0           3.46
##  5 Afghanistan     5791 134653               0           3.46
##  6 Afghanistan     5791 134653               0           3.46
##  7 Afghanistan     5791 134653               0           3.46
```

```
##  8 Afghanistan     5791 134653              0          3.46
##  9 Afghanistan     5791 134653              0          3.46
## 10 Afghanistan     5791 134653              0          3.46
## # ... with 134,459 more rows
```

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

```
## # A tibble: 55 x 6
##    Province_State       deaths   cases population cases_per_thou deaths_per_thou
##    <chr>                 <dbl>   <dbl>      <dbl>          <dbl>           <dbl>
##  1 Alabama               11402  555215    4903185          113.            2.33
##  2 Alaska                  381   71905     740995           97.0           0.514
##  3 Arizona               18055  901906    7278717          124.            2.48
##  4 Arkansas               5970  358949    3017804          119.            1.98
##  5 California            63984 3845180   39512223           97.3           1.62
##  6 Colorado               6861  563642    5758736           97.9           1.19
##  7 Connecticut            8279  350245    3565287           98.2           2.32
##  8 Delaware               1695  110112     973764          113.            1.74
##  9 District of Columbia   1144   49536     705749           70.2           1.62
## 10 Florida               38157 2404895   21477737          112.            1.78
## # ... with 45 more rows
```

Top 10 best states in terms of lowest cases and deaths related to COVID-19.

```
US_state_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 Province_State        deaths  cases population
##              <dbl>          <dbl> <chr>                 <dbl>  <dbl>      <dbl>
##  1          0.0363           3.32 Northern Mariana Isl~     2    183      55144
##  2          0.298            37.7 Virgin Islands           32   4043     107268
##  3          0.368            27.3 Hawaii                  521  38605    1415872
##  4          0.413            39.3 Vermont                 258  24497     623989
##  5          0.514            97.0 Alaska                  381  71905     740995
##  6          0.641            51.5 Maine                   862  69285    1344212
##  7          0.664            50.0 Oregon                 2800 211065    4217737
##  8          0.680            37.5 Puerto Rico            2555 140974    3754939
##  9          0.749           131.  Utah                  2402 420685    3205958
## 10          0.791            60.1 Washington             6022 457814    7614893
```

Top 10 worst States in terms of highest cases and deaths related to COVID-19

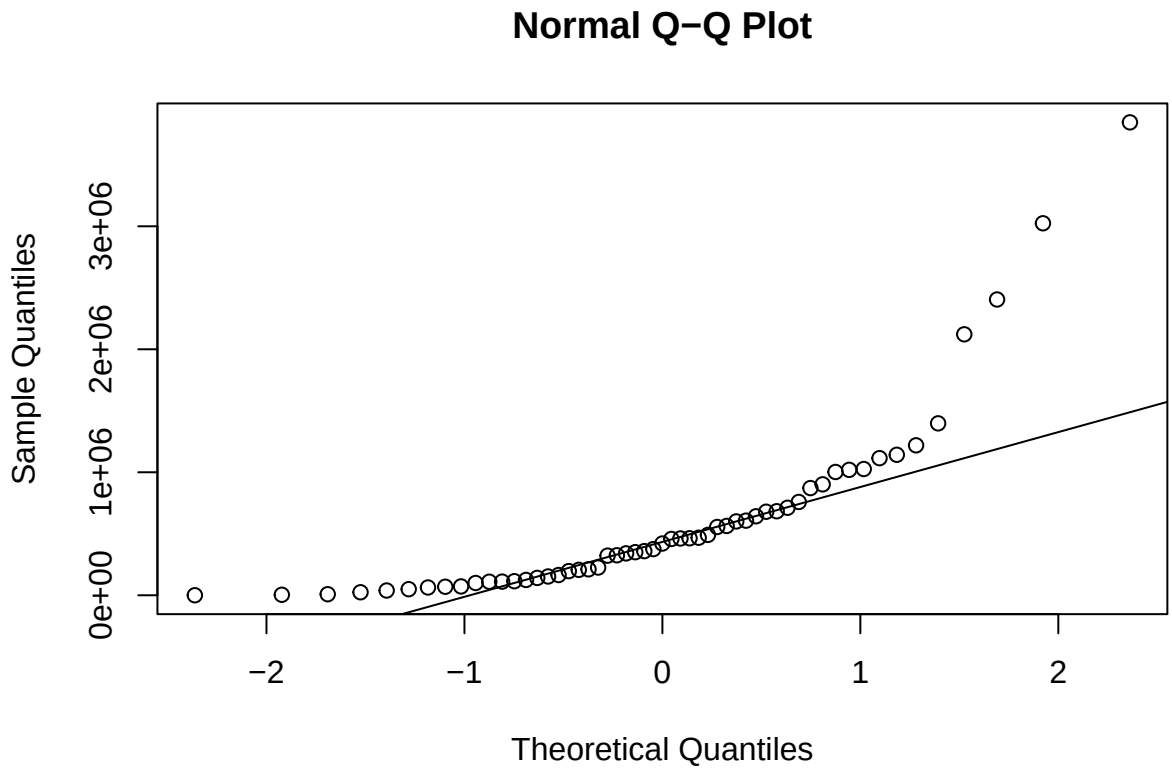
```
US_state_totals %>%
  slice_max(deaths_per_thou, n = 10) %>%
  select(deaths_per_thou, cases_per_thou, everything())
```

```
## # A tibble: 10 x 6
##    deaths_per_thou cases_per_thou Province_State deaths   cases population
##              <dbl>          <dbl> <chr>           <dbl>   <dbl>      <dbl>
##  1            2.99           116. New Jersey      26516 1026649    8882190
##  2            2.76           109. New York        53743 2121761   19453561
##  3            2.61           103. Massachusetts   18012  711446    6892503
##  4            2.58           144. Rhode Island     2731  152842    1059361
##  5            2.50           109. Mississippi      7451  325072    2976149
##  6            2.48           124. Arizona         18055  901906    7278717
##  7            2.33           113. Alabama         11402  555215    4903185
##  8            2.32           106. Louisiana       10798  490904    4648794
##  9            2.32            98.2 Connecticut     8279  350245    3565287
## 10            2.30           141. South Dakota     2039  124652     884659
```

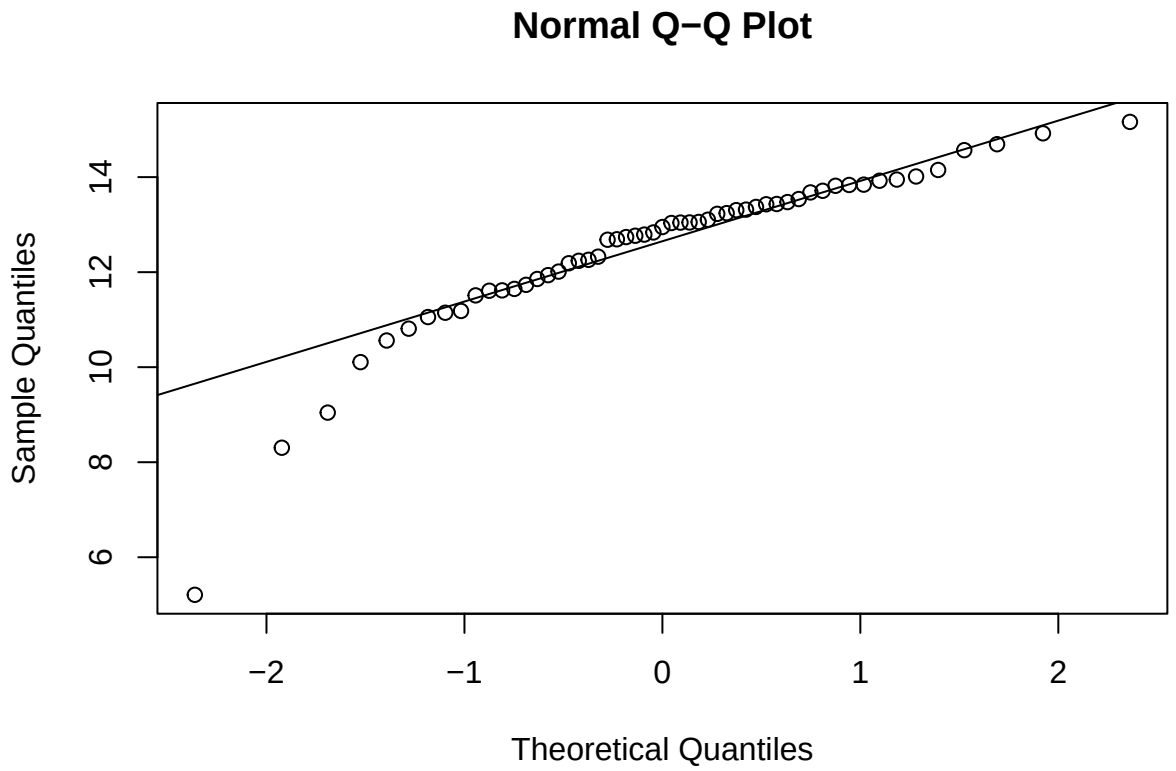
### Data Modelling

To determine if the cases variable is normal, we will visually verifying the Normal Q-Q plot to see if it follows the line. Having normalized data can help us accurately conduct various tests.

```
qqnorm(US_state_totals$cases)
qqline(US_state_totals$cases)
```



```
qqnorm(log(US_state_totals$cases))
qqline(log(US_state_totals$cases))
```




Lets predict deaths per thousand using cases per thousand and then add a new predict column to compare the predict and actual values. From the model summary, we can interpret that a 10% increase in population will result in roughly 12.3% increase in COVID-19 cases.

```
mod <- lm(log(cases) ~ log(population), data = US_state_totals)
summary(mod)
```

```
##
## Call:
## lm(formula = log(cases) ~ log(population), data = US_state_totals)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.34471 -0.19640  0.08656  0.23678  0.83355
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -5.86855    0.77323   -7.59 5.08e-10 ***
## log(population)  1.22945    0.05145   23.90  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.497 on 53 degrees of freedom
## Multiple R-squared:  0.9151, Adjusted R-squared:  0.9135
## F-statistic: 571.1 on 1 and 53 DF,  p-value: < 2.2e-16
```

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

```
## # A tibble: 1 x 6
##   Province_State          deaths cases population cases_per_thou deaths_per_thou
##   <chr>                    <dbl> <dbl>      <dbl>          <dbl>           <dbl>
## 1 Northern Mariana Islan~      2   183      55144           3.32          0.0363
```

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

```
## # A tibble: 1 x 6
##   Province_State deaths  cases population cases_per_thou deaths_per_thou
##   <chr>           <dbl>  <dbl>      <dbl>          <dbl>           <dbl>
## 1 North Dakota     1563 110899     762062           146.            2.05
```

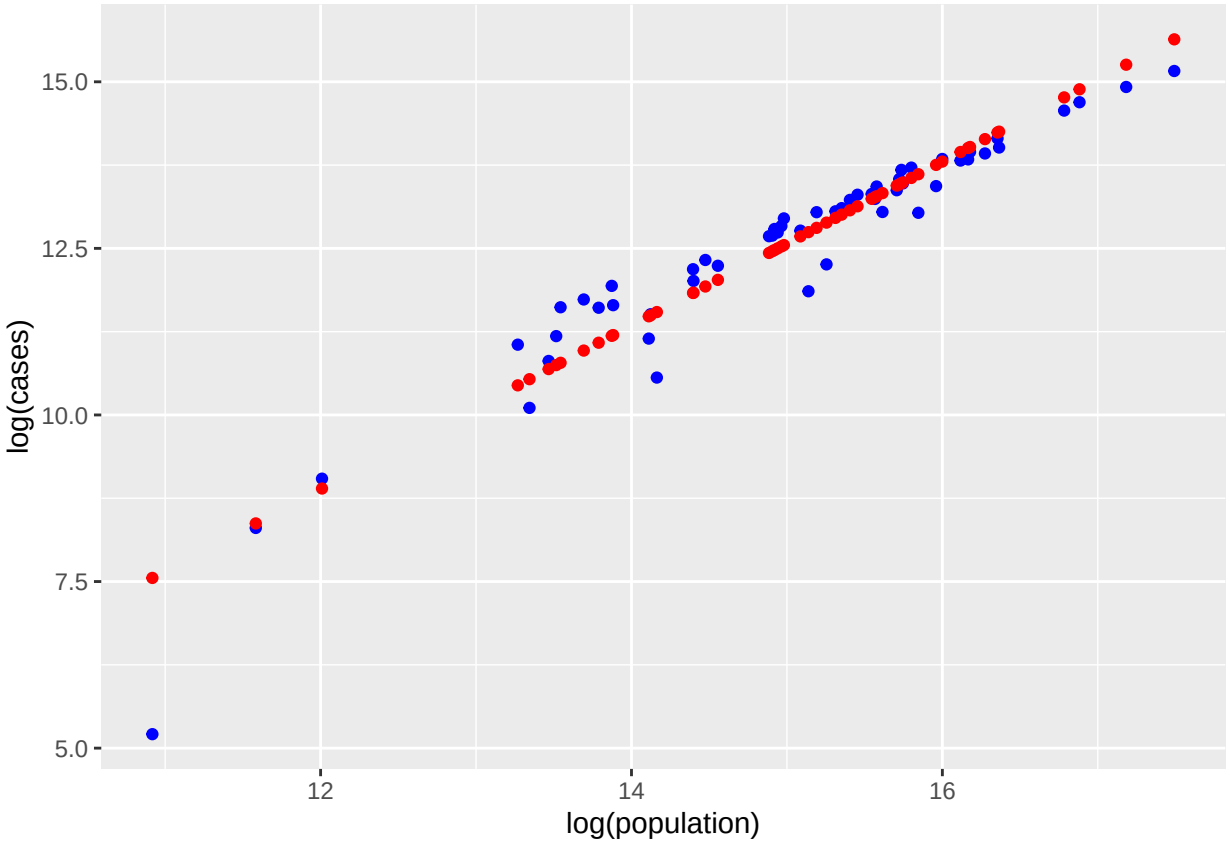
```
US_state_totals %>%
  mutate(pred = predict(mod))
```

```
## # A tibble: 55 x 7
##    Province_State  deaths  cases population cases_per_thou deaths_per_thou  pred
##    <chr>            <dbl>  <dbl>      <dbl>          <dbl>           <dbl> <dbl>
##  1 Alabama          11402 5.55e5    4903185          113.            2.33   13.1
##  2 Alaska             381 7.19e4     740995           97.0           0.514  10.7
##  3 Arizona          18055 9.02e5    7278717          124.            2.48   13.6
##  4 Arkansas          5970 3.59e5    3017804          119.            1.98   12.5
##  5 California       63984 3.85e6   39512223           97.3           1.62   15.6
##  6 Colorado          6861 5.64e5    5758736           97.9           1.19   13.3
##  7 Connecticut       8279 3.50e5    3565287           98.2           2.32   12.7
##  8 Delaware          1695 1.10e5     973764          113.            1.74   11.1
##  9 District of Co~   1144 4.95e4     705749           70.2           1.62   10.7
## 10 Florida          38157 2.40e6   21477737          112.            1.78   14.9
## # ... with 45 more rows
```

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

We will now visualize to compare predicted and actual values in order to see how our model is doing in predicting the deaths per thousand.

```
US_tot_w_pred %>% ggplot() +
  geom_point(aes(x=log(population), y = log(cases)), color = "blue") +
  geom_point(aes(x=log(population), y=pred), color = "red")
```


15.0
12.5
10.0
7.5
5.0
log(cases)
12
14
16
log(population)