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: 330,327 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 330,317 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: 330,327 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 330,317 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: 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                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
```

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

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

```
## # A tibble: 306,827 x 5
##    Province_State Country_Region date       cases deaths
##    <chr>          <chr>          <date>     <dbl>  <dbl>
##  1 <NA>           Afghanistan    2020-02-24     5      0
##  2 <NA>           Afghanistan    2020-02-25     5      0
##  3 <NA>           Afghanistan    2020-02-26     5      0
```

```
##  4 <NA>           Afghanistan    2020-02-27     5      0
##  5 <NA>           Afghanistan    2020-02-28     5      0
##  6 <NA>           Afghanistan    2020-02-29     5      0
##  7 <NA>           Afghanistan    2020-03-01     5      0
##  8 <NA>           Afghanistan    2020-03-02     5      0
##  9 <NA>           Afghanistan    2020-03-03     5      0
## 10 <NA>           Afghanistan    2020-03-04     5      0
## # ... with 306,817 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: 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, 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 3,819,896 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: 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, 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 3,819,896 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: 3,819,906 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 3,819,896 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_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()
## )
```

```
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:3819906     Length:3819906     Length:3819906     Length:3819906
##  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.   :  -3073   Min.   :       0   Min.   :  -82.0
##  1st Qu.:2020-11-02   1st Qu.:    330   1st Qu.:    9917   1st Qu.:    4.0
##  Median :2021-08-15   Median :   2272   Median :   24892   Median :   37.0
##  Mean   :2021-08-15   Mean   :  14088   Mean   :   99604   Mean   :  186.9
##  3rd Qu.:2022-05-28   3rd Qu.:   8159   3rd Qu.:   64979   3rd Qu.:  122.0
##  Max.   :2023-03-09   Max.   :3710586   Max.   :10039107   Max.   :35545.0
```

```
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          Population        Combined_Key
```

```
##  Min.   :      0   Min.   :6.700e+01   Length:306827
##  1st Qu.:      7   1st Qu.:7.866e+05   Class :character
##  Median :    214   Median :6.948e+06   Mode  :character
##  Mean   :  14405   Mean   :2.890e+07
##  3rd Qu.:   3665   3rd Qu.:2.914e+07
##  Max.   :1123836   Max.   :1.380e+09
##                    NA's   :6729
```

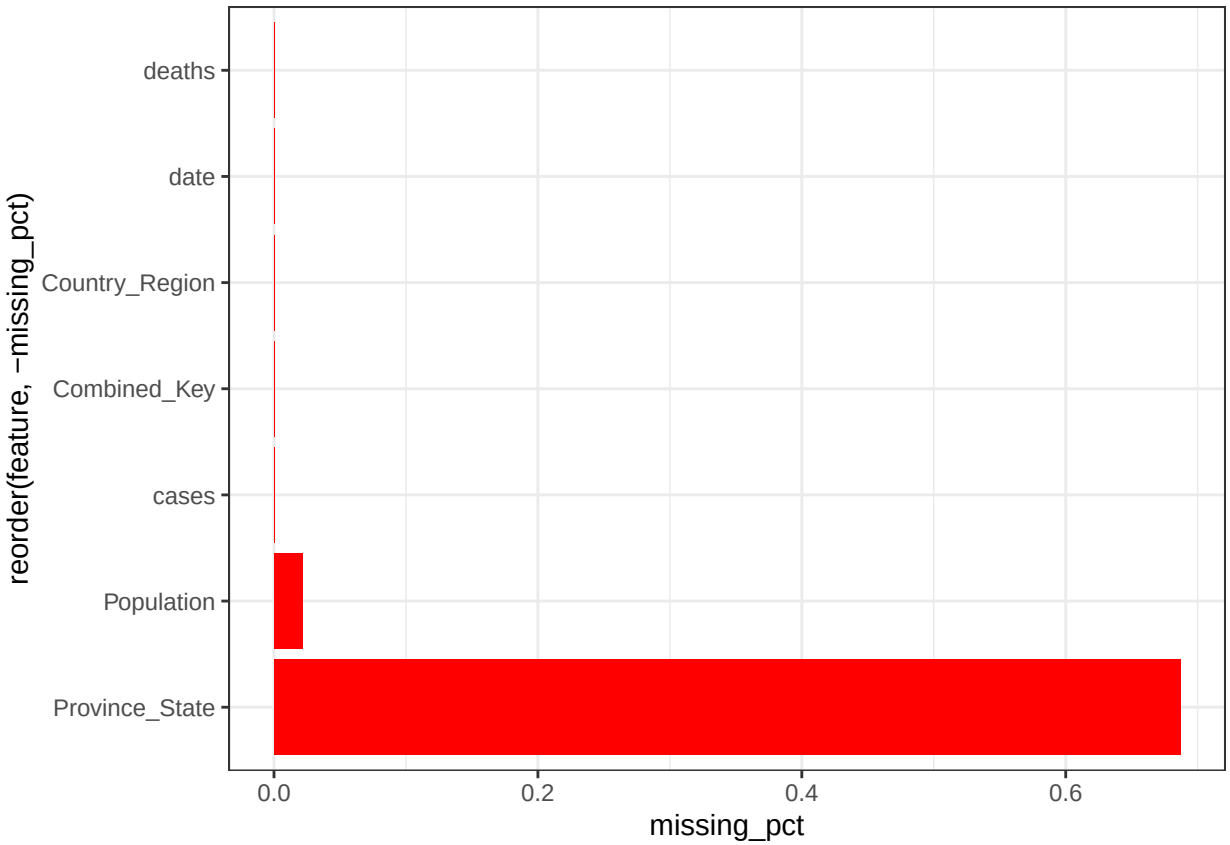
How many missing values are there for each variable? From the graph, we can see that there is a lot of missing data for Province/State as most countries do not have provinces.

```
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: 214,113 x 6
##    Country_Region date       cases deaths deaths_per_mil 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
```

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: 66,294 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 66,284 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: 1,143 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
```

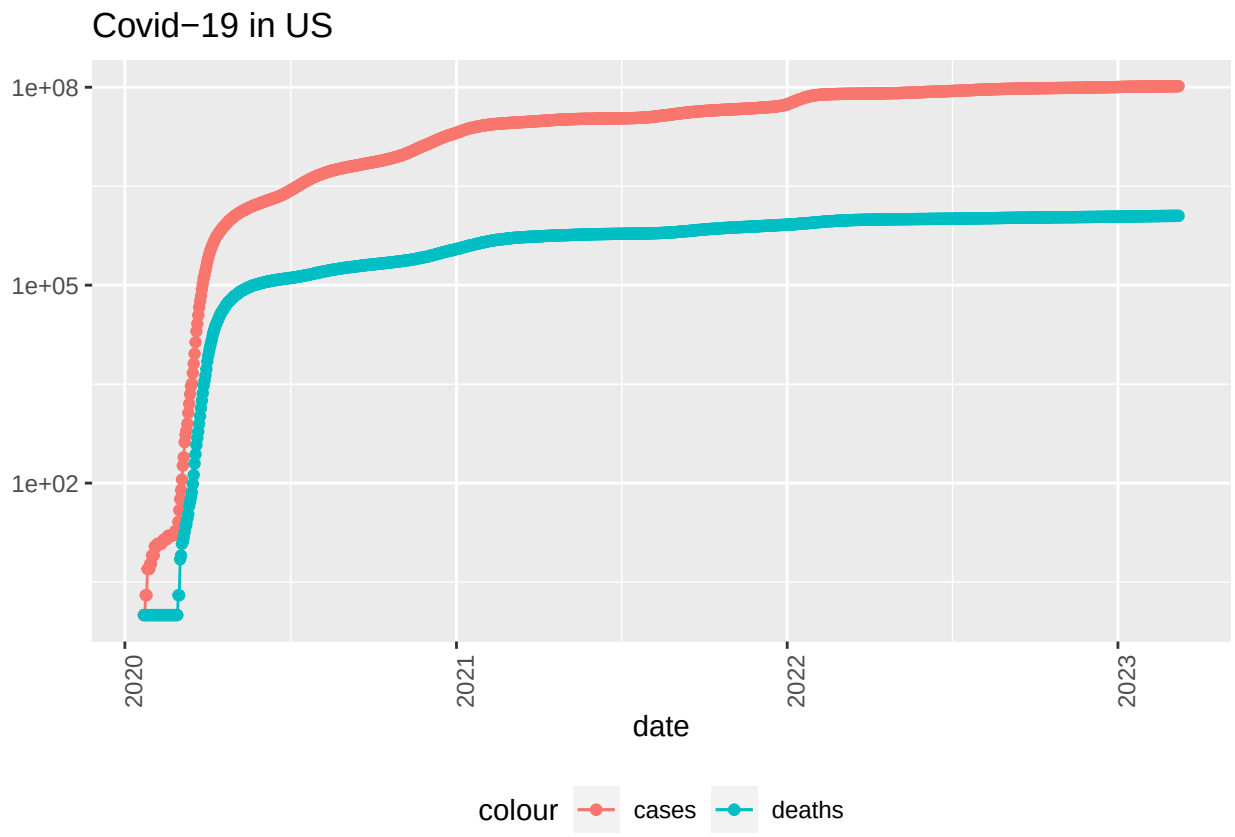
```
##  7 US              2020-01-28     5      1        0.00300  332875137
##  8 US              2020-01-29     6      1        0.00300  332875137
##  9 US              2020-01-30     6      1        0.00300  332875137
## 10 US              2020-01-31     8      1        0.00300  332875137
## # ... with 1,133 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             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
```

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)
```

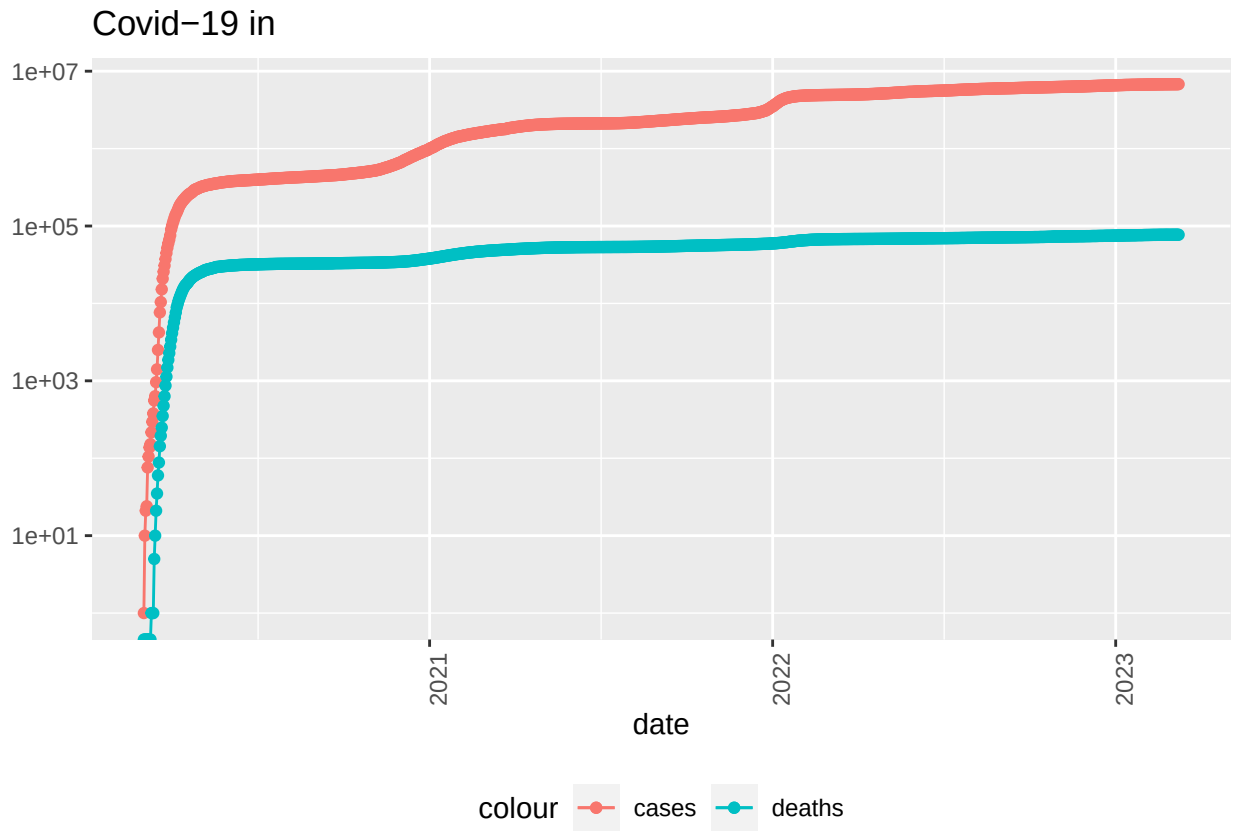


Looking at the same graph for New York State.

```
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             2023-03-04 103650837 1.12e6          3371.  332875137      2147
## 2 US             2023-03-05 103646975 1.12e6          3371.  332875137     -3862
## 3 US             2023-03-06 103655539 1.12e6          3371.  332875137      8564
## 4 US             2023-03-07 103690910 1.12e6          3372.  332875137     35371
## 5 US             2023-03-08 103755771 1.12e6          3374.  332875137     64861
## 6 US             2023-03-09 103802702 1.12e6          3376.  332875137     46931
## # ... 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 in self$trans$transform(x): NaNs produced
```

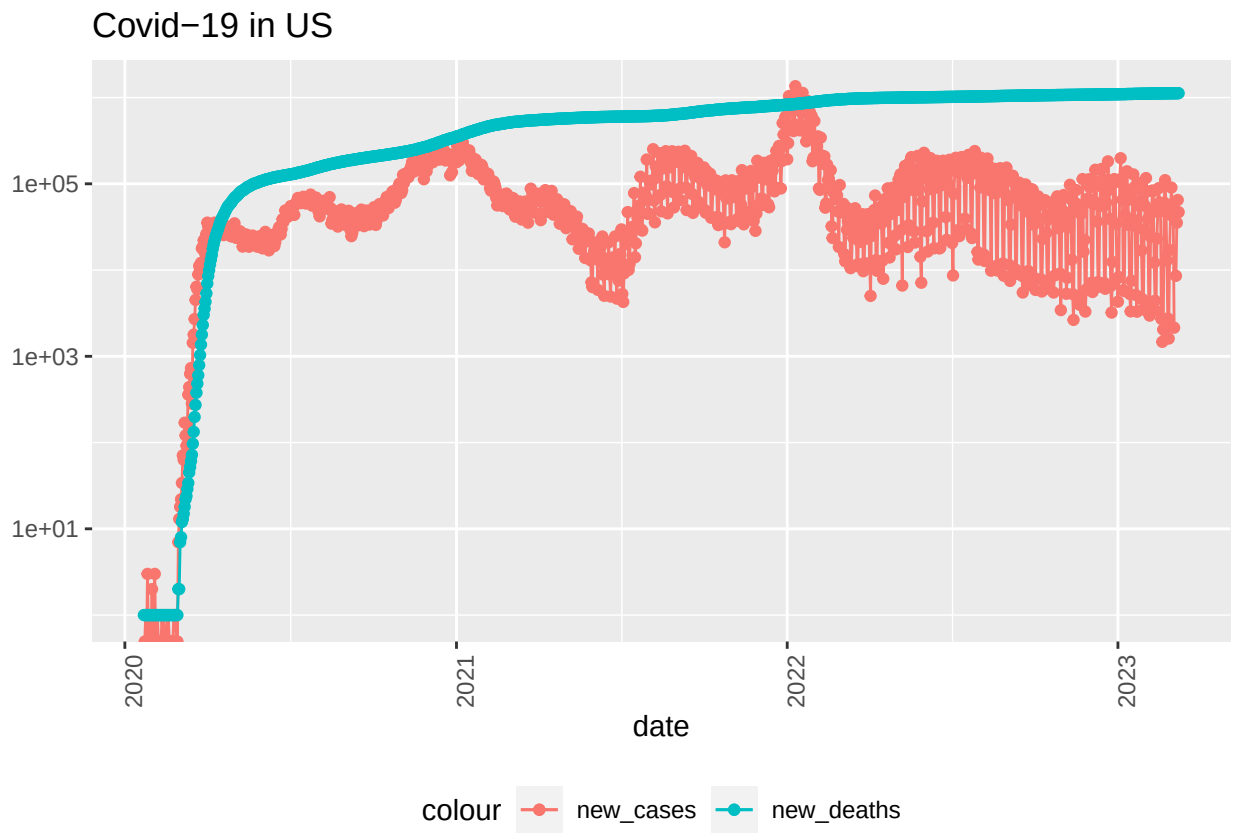
```
## Warning: Transformation introduced infinite values in continuous y-axis
```

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

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

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

```
## Warning: Removed 2 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.

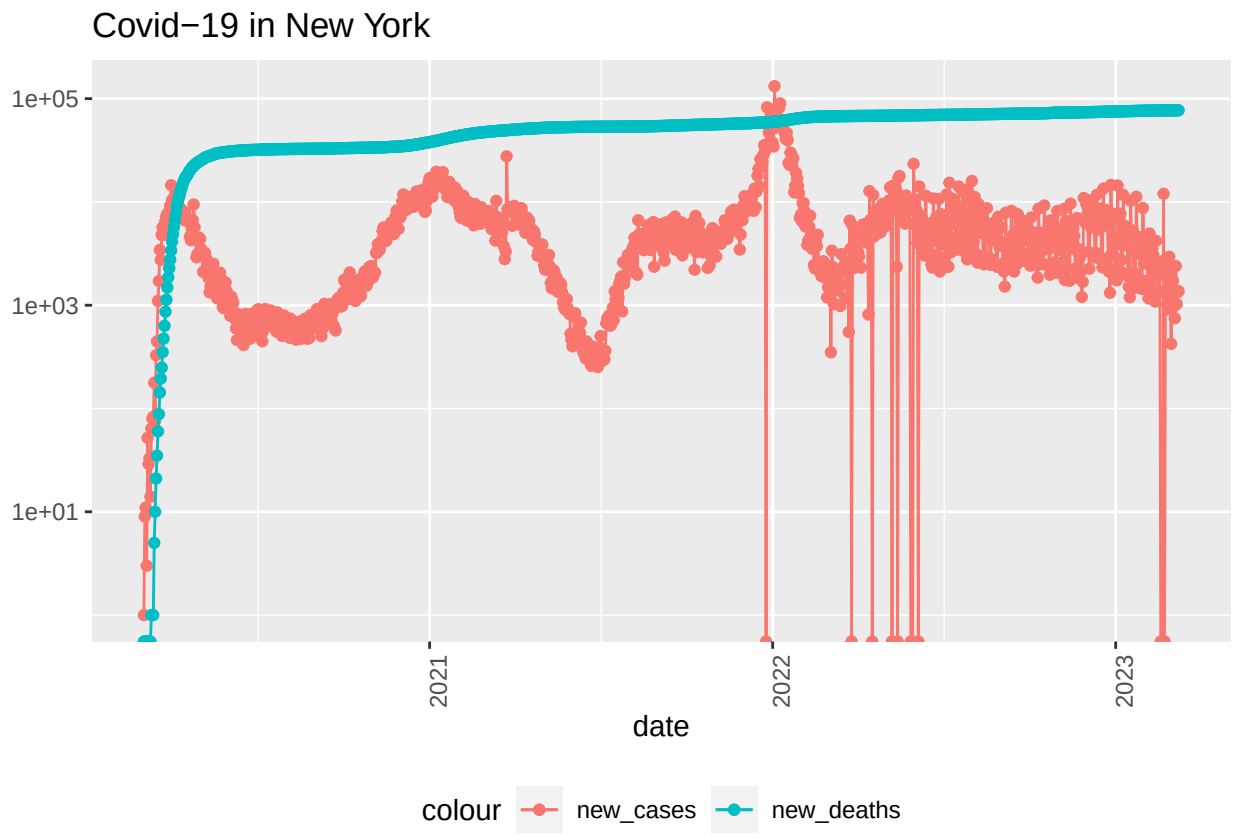
```
state <- "New York"
US_by_state %>%
  filter(Province_State == "New York") %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = new_cases)) +
  geom_line(aes(color = "new_cases")) +
  geom_point(aes(color = "new_cases")) +
  geom_line(aes(y = deaths, color = "new_deaths")) +
  geom_point(aes(y = 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

## 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: 306,827 x 5
## # Groups:   Country_Region [201]
##    Country_Region death  cases deaths_per_thou cases_per_thou
##    <chr>          <dbl>  <dbl>           <dbl>          <dbl>
##  1 Afghanistan     7896 209451               0           5.38
##  2 Afghanistan     7896 209451               0           5.38
##  3 Afghanistan     7896 209451               0           5.38
##  4 Afghanistan     7896 209451               0           5.38
##  5 Afghanistan     7896 209451               0           5.38
##  6 Afghanistan     7896 209451               0           5.38
##  7 Afghanistan     7896 209451               0           5.38
```

```
##  8 Afghanistan     7896 209451              0          5.38
##  9 Afghanistan     7896 209451              0          5.38
## 10 Afghanistan     7896 209451              0          5.38
## # ... with 306,817 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: 56 x 6
##    Province_State       deaths   cases population cases_per_thou deaths_per_thou
##    <chr>                 <dbl>   <dbl>      <dbl>          <dbl>           <dbl>
##  1 Alabama               21032  1.64e6    4903185           335.            4.29
##  2 Alaska                 1486  3.08e5     740995           415.            2.01
##  3 American Samoa           34  8.32e3      55641           150.            0.611
##  4 Arizona               33102  2.44e6    7278717           336.            4.55
##  5 Arkansas              13020  1.01e6    3017804           334.            4.31
##  6 California           101159  1.21e7   39512223           307.            2.56
##  7 Colorado              14181  1.76e6    5758736           306.            2.46
##  8 Connecticut           12220  9.77e5    3565287           274.            3.43
##  9 Delaware               3324  3.31e5     973764           340.            3.41
## 10 District of Columbia   1432  1.78e5     705749           252.            2.03
## # ... with 46 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           0.611           150. American Samoa           34 8.32e3      55641
##  2           0.744           248. Northern Mariana Isl~    41 1.37e4      55144
##  3           1.21            231. Virgin Islands          130 2.48e4     107268
##  4           1.30            269. Hawaii                 1841 3.81e5    1415872
##  5           1.49            245. Vermont                 929 1.53e5     623989
##  6           1.55            293. Puerto Rico            5823 1.10e6    3754939
##  7           1.65            340. Utah                   5298 1.09e6    3205958
##  8           2.01            415. Alaska                 1486 3.08e5     740995
##  9           2.03            252. District of Columbia   1432 1.78e5     705749
## 10           2.06            253. Washington            15683 1.93e6    7614893
```

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

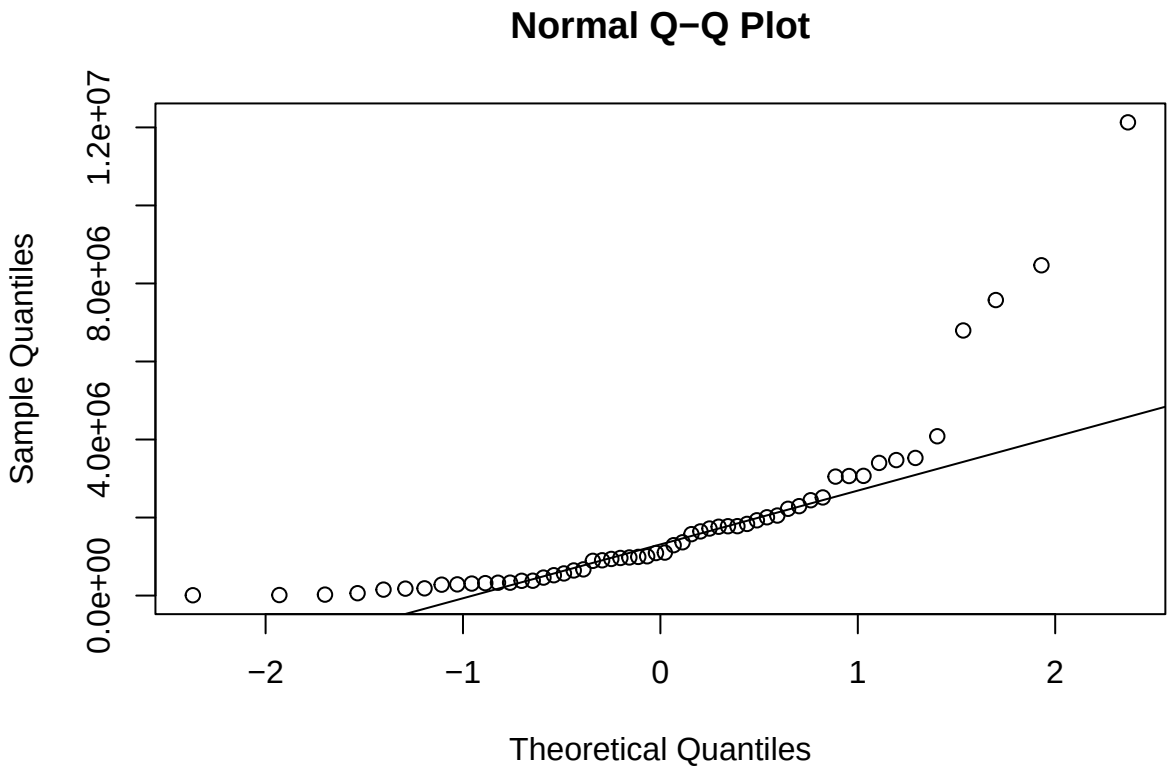
```
US_state_totals %>%
  slice_max(deaths_per_thou, n = 10) %>%
  select(deaths_per_thou, cases_per_thou, everything())
```

```
## # A tibble: 10 x 6
##    deaths_per_thou cases_per_thou Province_State deaths   cases population
##              <dbl>          <dbl> <chr>           <dbl>   <dbl>      <dbl>
##  1            4.55           336. Arizona         33102 2443514    7278717
##  2            4.54           326. Oklahoma        17972 1290929    3956971
##  3            4.49           333. Mississippi     13370  990756    2976149
##  4            4.44           359. West Virginia    7960  642760    1792147
##  5            4.32           320. New Mexico       9061  670929    2096829
##  6            4.31           334. Arkansas        13020 1006883    3017804
##  7            4.29           335. Alabama         21032 1644533    4903185
##  8            4.28           368. Tennessee       29263 2515130    6829174
##  9            4.23           307. Michigan        42205 3064125    9986857
## 10            4.06           385. Kentucky        18130 1718471    4467673
```

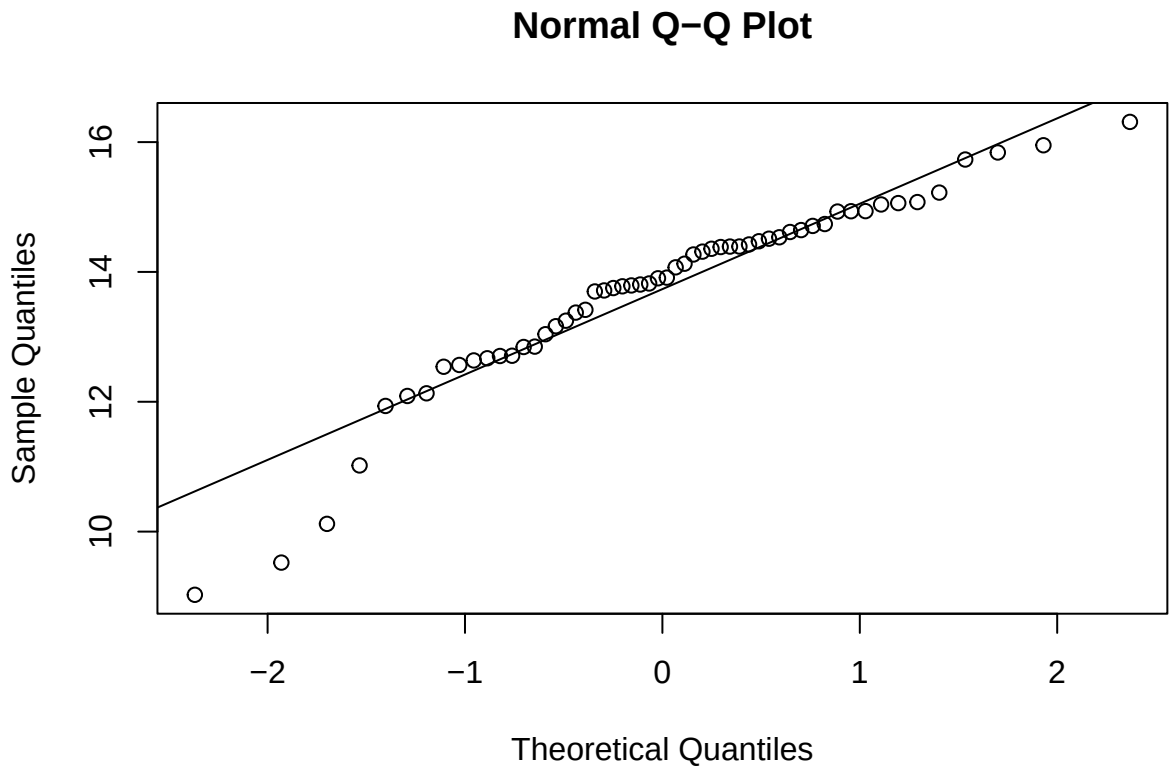
Data Modelling

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
## -0.56300 -0.09642  0.00871  0.09057  0.39584
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     -1.74051    0.24184  -7.197 1.97e-09 ***
## log(population)  1.03691    0.01616  64.169  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.169 on 54 degrees of freedom
## Multiple R-squared:  0.9871, Adjusted R-squared:  0.9868
## F-statistic:  4118 on 1 and 54 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 American Samoa     34  8320      55641           150.           0.611
```

```
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 Rhode Island     3870 460697    1059361           435.            3.65
```

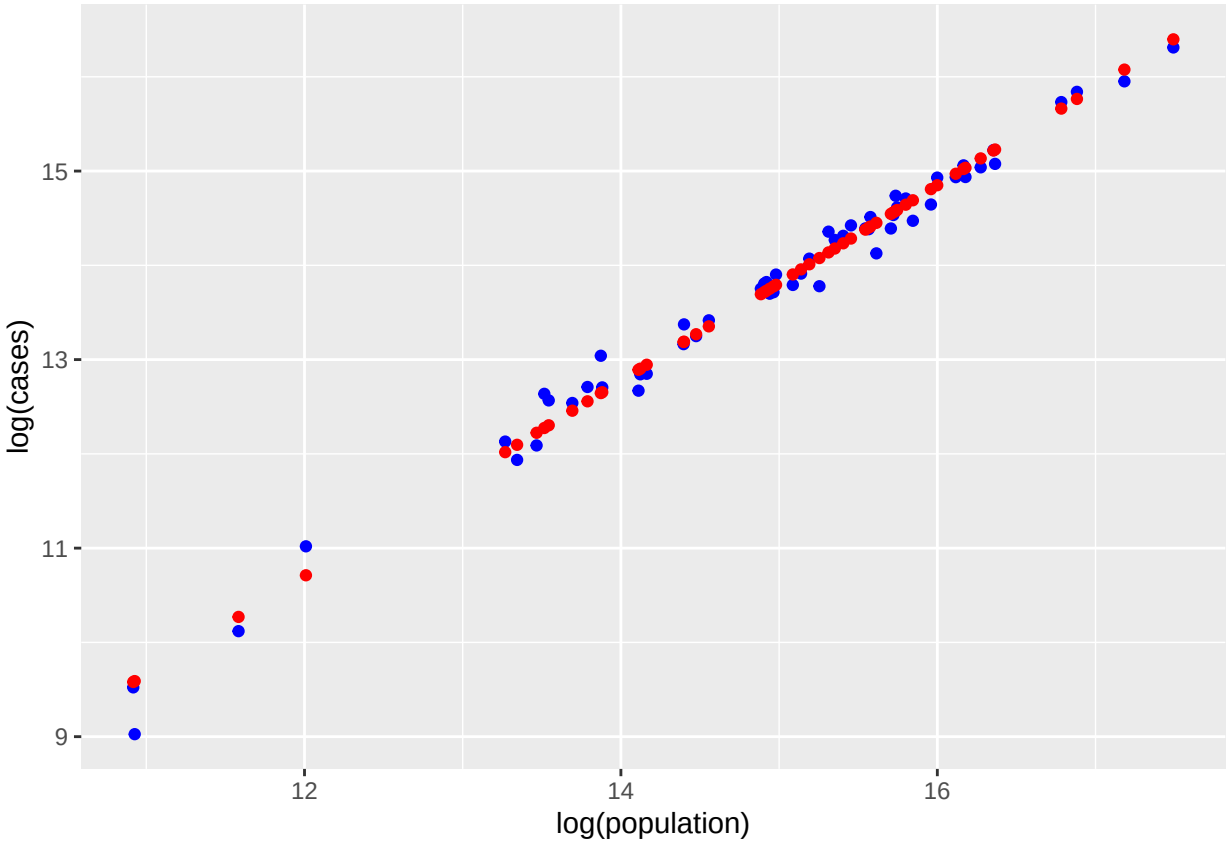
```
US_state_totals %>%
  mutate(pred = predict(mod))
```

```
## # A tibble: 56 x 7
##    Province_State  deaths  cases population cases_per_thou deaths_per_thou  pred
##    <chr>            <dbl>  <dbl>      <dbl>          <dbl>           <dbl> <dbl>
##  1 Alabama          21032 1.64e6    4903185           335.            4.29  14.2
##  2 Alaska            1486 3.08e5     740995           415.            2.01  12.3
##  3 American Samoa      34 8.32e3      55641           150.           0.611  9.59
##  4 Arizona          33102 2.44e6    7278717           336.            4.55  14.6
##  5 Arkansas         13020 1.01e6    3017804           334.            4.31  13.7
##  6 California      101159 1.21e7   39512223           307.            2.56  16.4
##  7 Colorado         14181 1.76e6    5758736           306.            2.46  14.4
##  8 Connecticut      12220 9.77e5    3565287           274.            3.43  13.9
##  9 Delaware          3324 3.31e5     973764           340.            3.41  12.6
## 10 District of Co~   1432 1.78e5     705749           252.            2.03  12.2
## # ... with 46 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
13
11
9
log(cases)
12
14
16
log(population)