COVID 19 Data Analysis

Vidit Vivek Sharma

21/09/2021

COVID 19 Analysis

Let us start with the analysis that we did in the class with Jane Wall.

We use this module to look at the steps in a data analysis in a reproducible manner using COVID-19 data. First we find some data sources.

We look at the [nytimes]https://github.com/nytimes/covid-19-data and [Johns Hopkins University]https://github.com/CSSEGISandData/COVID-19 github sites. After reviewing a little, we find that JHU gives more detail on their sources and data.

Step 1 - Identify and import the data

We start by reading in the data from the four main csv files.

```
## Get current Data in the four files
# they all begin the same way
url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covi
file_names <- c("confirmed_global.csv",
                    "deaths_global.csv",
                    "confirmed_US.csv",
                    "deaths_US.csv")
urls <- str_c(url_in,file_names)
```

Then we read in the data and see what we are working with.

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

Step 2 - Tidy up the data

After looking at global_cases and global_deaths, we would like to tidy those datasets and put each variable (date, cases, deaths) in its own column. Also, we don't need Lat and Long for the analysis we are planning, so we 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_deaths <- global_deaths %>%
    pivot_longer(cols = -c(`Province/State`,
                           `Country/Region`, Lat, Long),
                 names_to = "date",
                values_to = "deaths") %>%
  select(-c(Lat, Long))


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")
```

Lets look at the summary of the table we created.

```
# look at a summary of the data to see if there are problems
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:169353      Length:169353      Min.   :2020-01-22   Min.   :       0
##  Class :character   Class :character   1st Qu.:2020-06-21   1st Qu.:     146
##  Mode  :character   Mode  :character   Median :2020-11-20   Median :    2297
##                                        Mean   :2020-11-20   Mean   :  286349
##                                        3rd Qu.:2021-04-21   3rd Qu.:   51809
##                                        Max.   :2021-09-19   Max.   :42088171
##      deaths
##  Min.   :     0
##  1st Qu.:     1
##  Median :    35
##  Mean   :  6604
##  3rd Qu.:   841
##  Max.   :673774
```

Removing cases where the cases are equal to zero.

```
# get rid of rows with no cases
global <- global %>% filter(cases > 0)
```

```
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_deaths <- US_deaths %>%
    pivot_longer(cols = -(UID:Population),
                 names_to = "date",
                 values_to ="deaths") %>%
    select(Admin2:deaths) %>%
    mutate(date = mdy(date)) %>%
 select(-c(Lat, Long_))
US <- US_cases %>%
  full_join(US_deaths)
```

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

We notice that we don't have population data for the world data. If we plan to do comparative analysis So we add population data and a variable called Combined_Key that combines the Province_State with the Country_Region

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

Let's add in population data to the global dataset.

```
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))
```

```
## Rows: 4196 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)) %>%
  select(Province_State, Country_Region, date,
         cases, deaths, Population,
         Combined_Key)
global
```

```
## # A tibble: 153,341 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
## # ... with 153,331 more rows
```

Step 3 - Visualize the data

Let's focus our analysis on the US data for now.

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

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

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

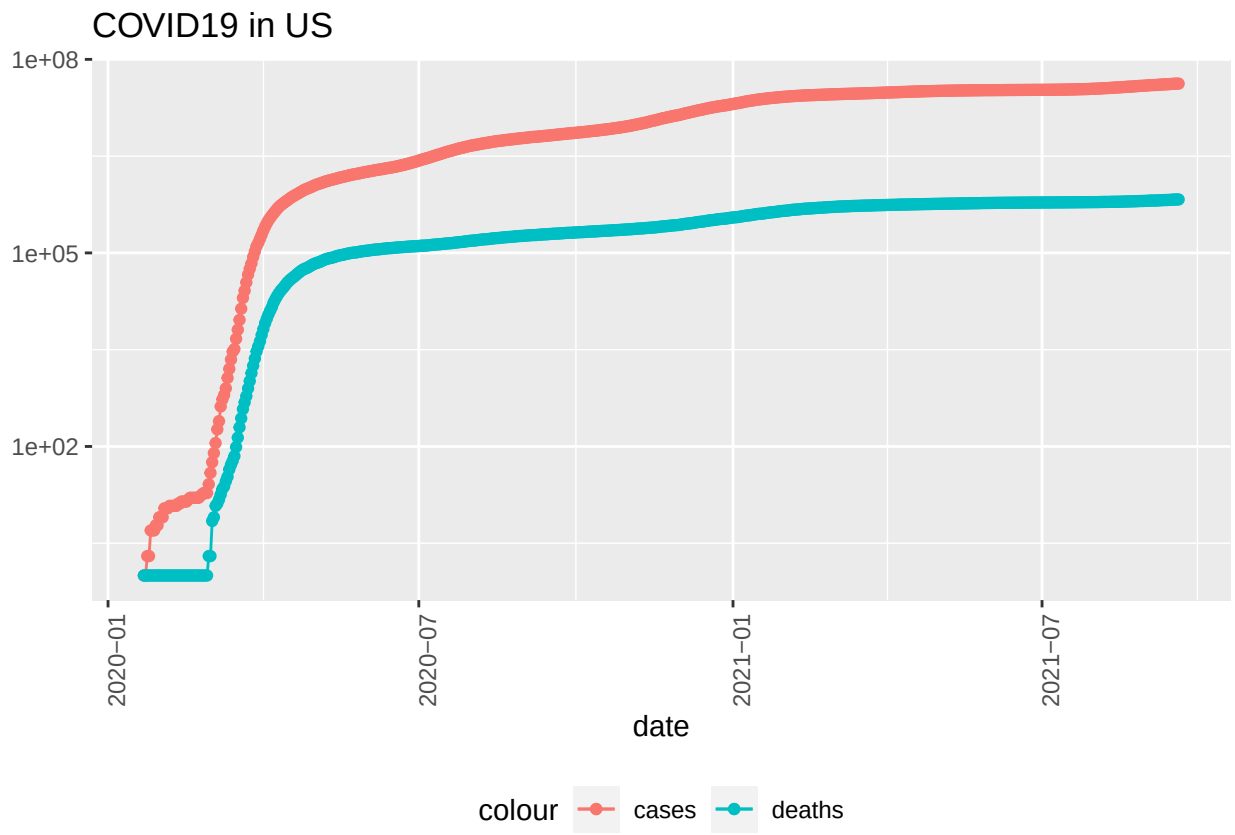
```
US_totals <- US_by_state %>%
  group_by(Country_Region, date) %>%
   summarize(cases = sum(cases), deaths = sum(deaths),
             Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths *1000000 / Population) %>%
  select(Country_Region, date,
         cases, deaths, deaths_per_mill,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

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

Lets plot the data for US as a whole.

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

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

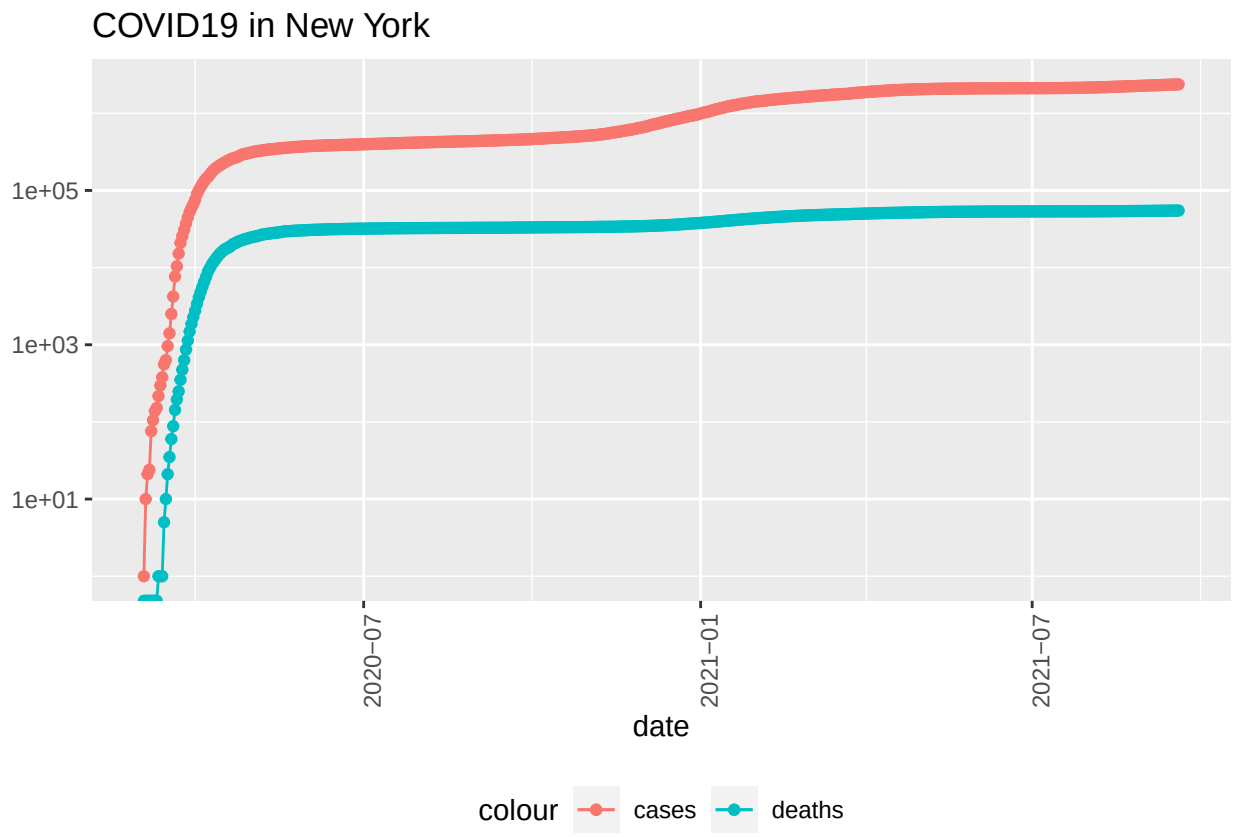


Lets plot the data for New York state.

```
state <- "New York"
US_by_state %>%
  filter(Province_State == state) %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = cases)) +
    geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) +
    scale_y_log10() +
    theme(legend.position="bottom",
          axis.text.x = element_text(angle = 90)) +
  labs(title = str_c("COVID19 in ", state), y= NULL)
```

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

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



Step 4 - Analyse the data

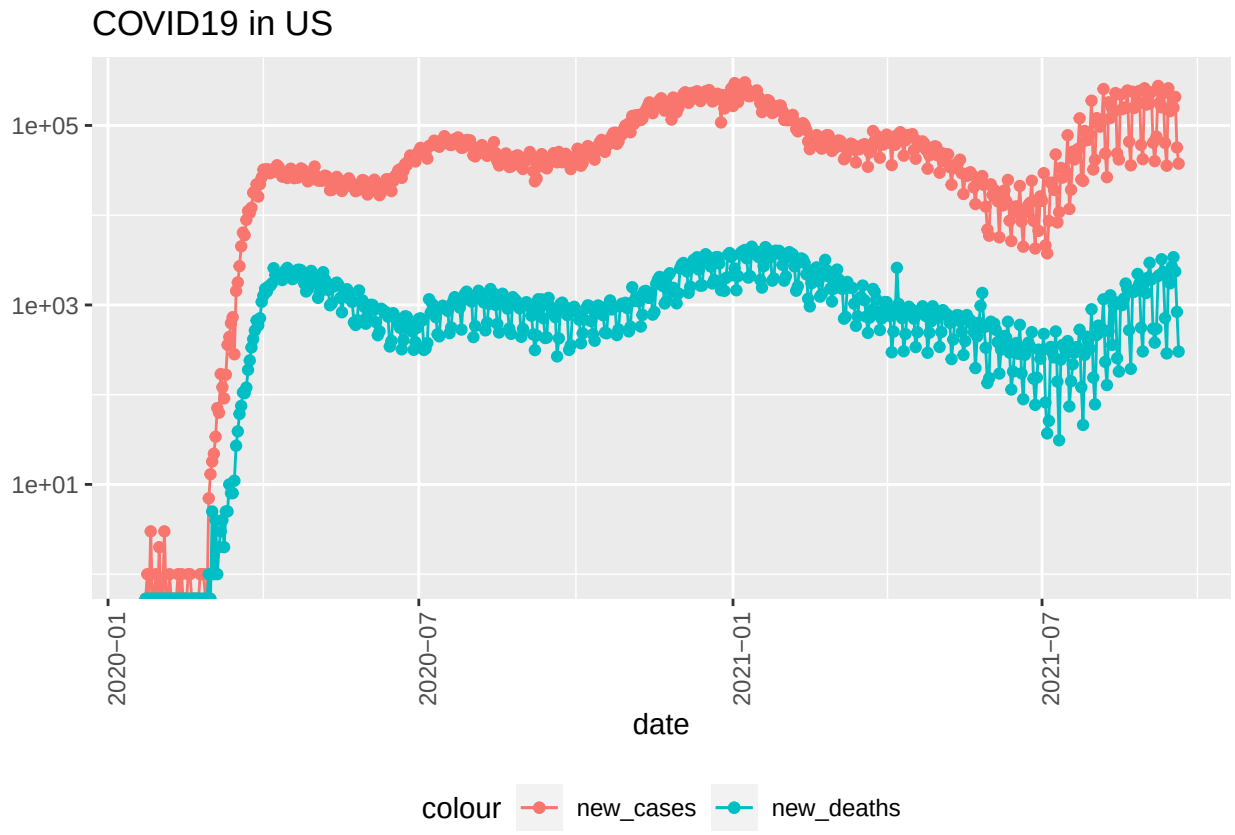
So our graph looks like COVID has leveled off.Lets look at the number of new cases and deaths per day.

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

Lets Visualize the number of new cases and deaths per day to see if that raises new questions.

```
US_totals %>%
  ggplot(aes(x = date, y = new_cases)) +
    geom_line(aes(color = "new_cases")) +
  geom_point(aes(color = "new_cases")) +
  geom_line(aes(y = new_deaths, color = "new_deaths")) +
  geom_point(aes(y = new_deaths, color = "new_deaths")) +
    scale_y_log10() +
    theme(legend.position="bottom",
          axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 in US", y= NULL)
```

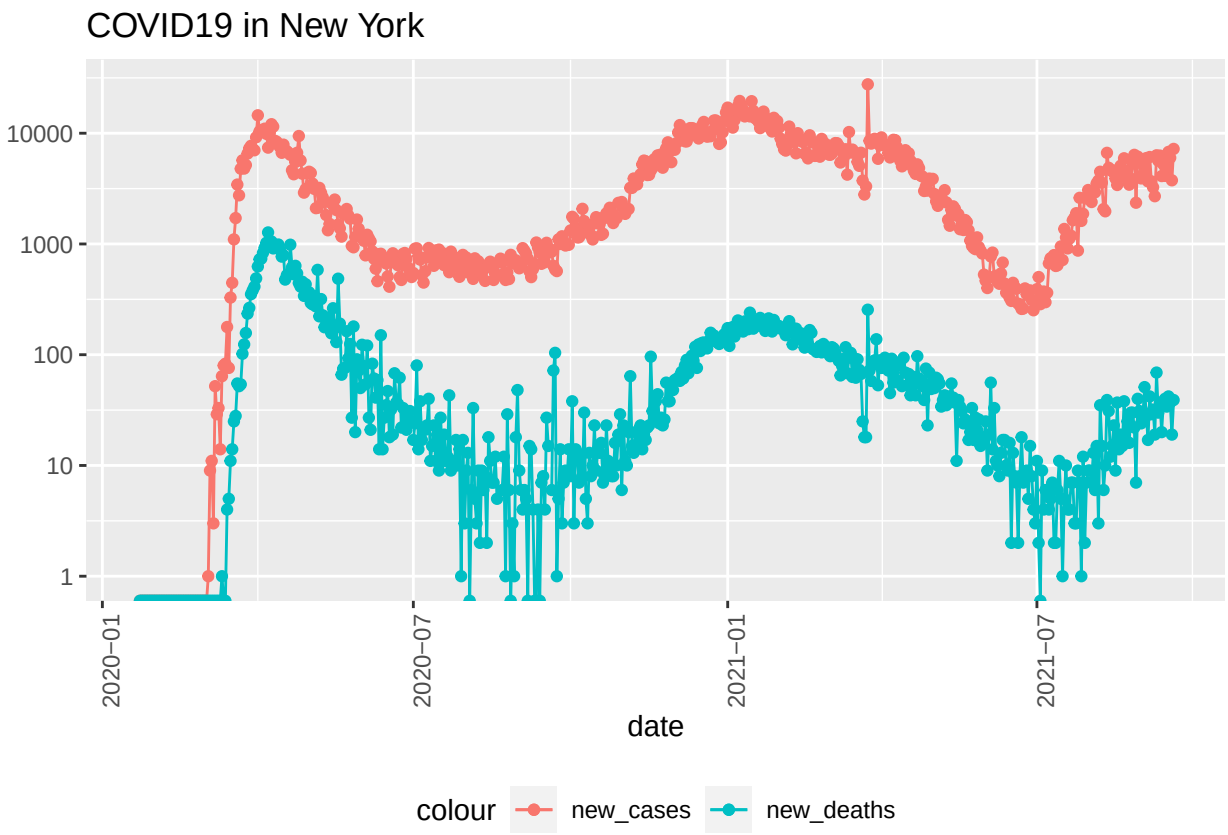
```
## Warning: Transformation introduced infinite values in continuous y-axis

## 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

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

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

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

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




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

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

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

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

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

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

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

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

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

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

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

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

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

## Warning: Removed 6 rows containing missing values (geom_point).
```



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

```
US_state_totals <- US_by_state %>%
  group_by(Province_State) %>%
  summarize(deaths = max(deaths), cases = max(cases),
            population = max(Population),
            cases_per_thou = 1000* cases / population,
            deaths_per_thou = 1000 * deaths / population) %>%
  filter(cases > 0, population > 0)
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 Northern Mariana Islands      2    263      55144           4.77          0.0363
##  2 Vermont                     298  31634     623989          50.7           0.478
##  3 Hawaii                      714  75480    1415872          53.3           0.504
##  4 Virgin Islands               67   6458     107268          60.2           0.625
##  5 Alaska                      469 100360     740995         135.            0.633
##  6 Maine                       984  83910    1344212          62.4           0.732
##  7 Puerto Rico                3074 179144    3754939          47.7           0.819
##  8 Oregon                     3569 309841    4217737          73.5           0.846
##  9 Utah                       2787 490985    3205958         153.            0.869
## 10 Washington                 7201 620752    7614893          81.5           0.946
```

```
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 Mississippi      9214  473413    2976149           159.            3.10
##  2 New Jersey      27190 1133228    8882190           128.            3.06
##  3 Louisiana       13418  725637    4648794           156.            2.89
##  4 New York        54904 2373659   19453561           122.            2.82
##  5 Alabama         13210  770391    4903185           157.            2.69
##  6 Arizona         19513 1066803    7278717           147.            2.68
##  7 Massachusetts   18445  790953    6892503           115.            2.68
##  8 Rhode Island     2812  168449    1059361           159.            2.65
##  9 Arkansas         7445  485056    3017804           161.            2.47
## 10 Florida         51240 3528698   21477737           164.            2.39
```

Step 5 - Model the data

We might need to introduce more variables here to build a model. Which do you want to consider? Population density, extent of lockdown, political affiliation, climate of the area? When you determine the factors you want to try, add that data to your dataset, and then visualize and model and see if your variable has a statistically significant effect.

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

```
##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou, data = US_state_totals)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4081 -0.2939 -0.0211  0.2741  1.1586
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.031944   0.241155   0.132    0.895
## cases_per_thou 0.014662   0.001924   7.620 4.53e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5044 on 53 degrees of freedom
## Multiple R-squared:  0.5228, Adjusted R-squared:  0.5138
## F-statistic: 58.07 on 1 and 53 DF,  p-value: 4.534e-10
```

```
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 Islands      2   263      55144           4.77          0.0363
```

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

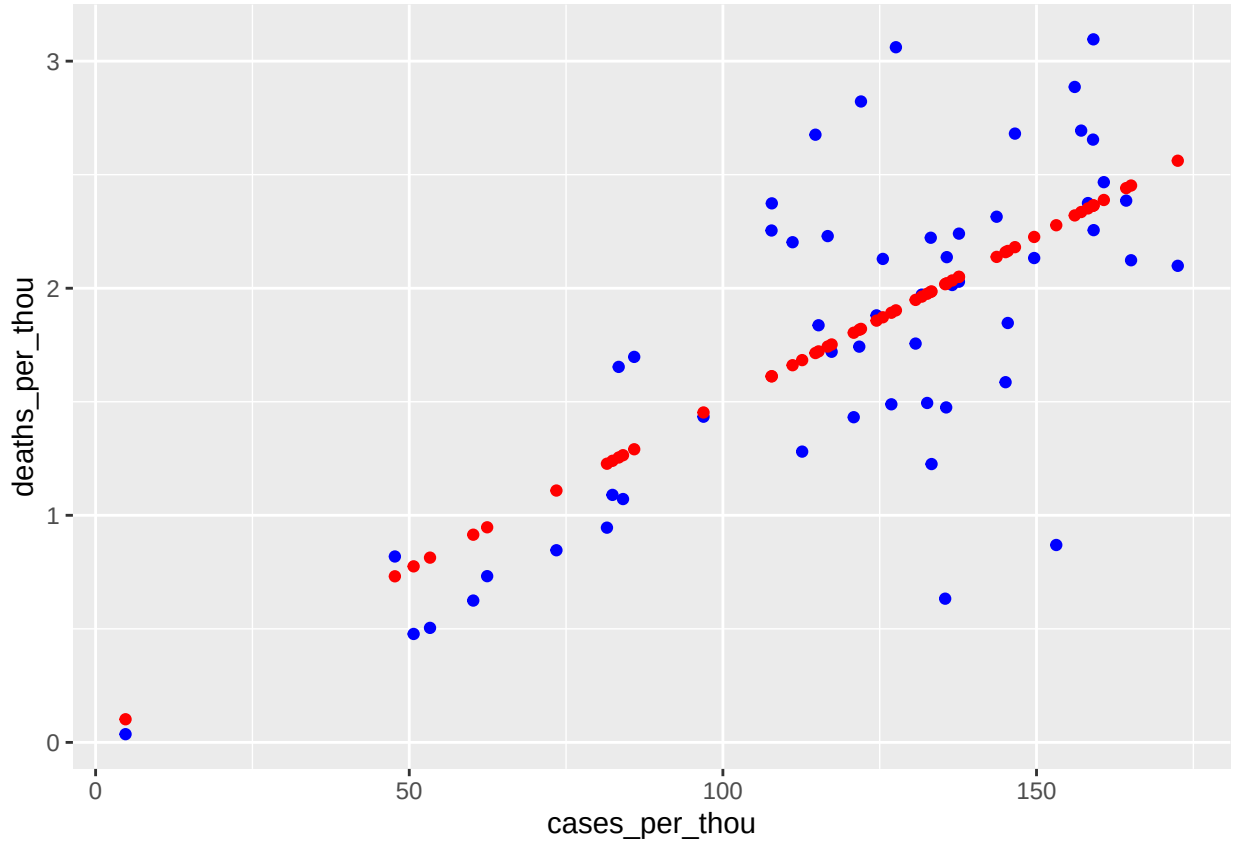
```
## # A tibble: 1 x 6
##   Province_State deaths   cases population cases_per_thou deaths_per_thou
##   <chr>           <dbl>   <dbl>      <dbl>          <dbl>           <dbl>
## 1 Tennessee       14332 1178168    6829174           173.            2.10
```

```
x_grid <- seq(1, 151)
new_df <- tibble(cases_per_thou = x_grid)
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          13210 7.70e5    4903185          157.            2.69   2.34
##  2 Alaska             469 1.00e5     740995          135.            0.633  2.02
##  3 Arizona          19513 1.07e6    7278717          147.            2.68   2.18
##  4 Arkansas          7445 4.85e5    3017804          161.            2.47   2.39
##  5 California       67966 4.64e6   39512223          117.            1.72   1.75
##  6 Colorado          7374 6.49e5    5758736          113.            1.28   1.68
##  7 Connecticut       8463 3.84e5    3565287          108.            2.37   1.61
##  8 Delaware          1920 1.28e5     973764          132.            1.97   1.96
##  9 District of Co~   1167 5.89e4     705749           83.4           1.65   1.25
## 10 Florida          51240 3.53e6   21477737          164.            2.39   2.44
## # ... with 45 more rows
```

Let us use the model to predict the State total cases and also plot the actual data for comparision.

```
US_tot_w_pred <- US_state_totals %>% mutate(pred = predict(mod))
US_tot_w_pred %>% ggplot() +
geom_point(aes(x = cases_per_thou, y = deaths_per_thou), color = "blue") +
geom_point(aes(x = cases_per_thou, y = pred), color = "red")
```



From the above graph we can see that the there is more concentration towards the prediction line as compared to the rest of the graph.

Step 6 - Additional Analysis (Beyond what is done in the class)

Ratio of death and cases can be used to comment on the medical infrastructure of a country or State? First we need to analyse the number of deaths and cases.

Let us take a look at the number of cases compared to the population and also to the number of death versus the number of cases of a country or a state. This could help us understand the extent of medical infrastructure in the country or state.

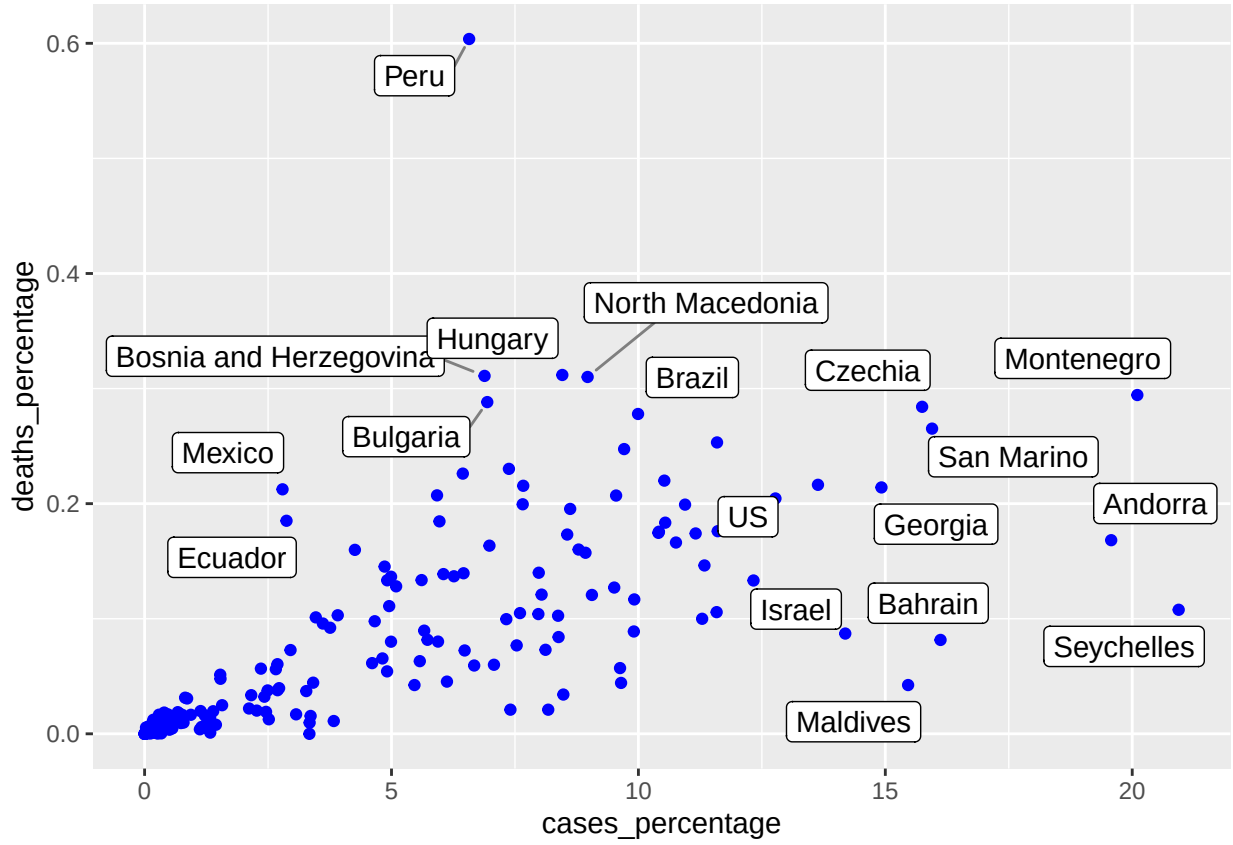
Lets first take a look at the global data for percentage of cases and the percentages of deaths compared to the population.

```
global_total <- global %>%
  group_by(Country_Region) %>%
  summarize(deaths = max(deaths), cases = max(cases),
            population = max(Population),
            cases_percentage = (cases / population)*100,
            deaths_percentage = (deaths / population)*100,
            death_cases_per =(deaths / cases)*100) %>%
  filter(cases > 0, population > 0)
```

Lets plot the cases_percentage and deaths_percentage for the global data.

```
global_total %>%
  ggplot(aes(x = cases_percentage, y = deaths_percentage )) +
  geom_point(color="blue")+ geom_label_repel(aes(label = Country_Region),
                                             box.padding=0.35,
                                             point.padding = 0.5,
                                             segment.color = 'grey50')
```

```
## Warning: ggrepel: 172 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



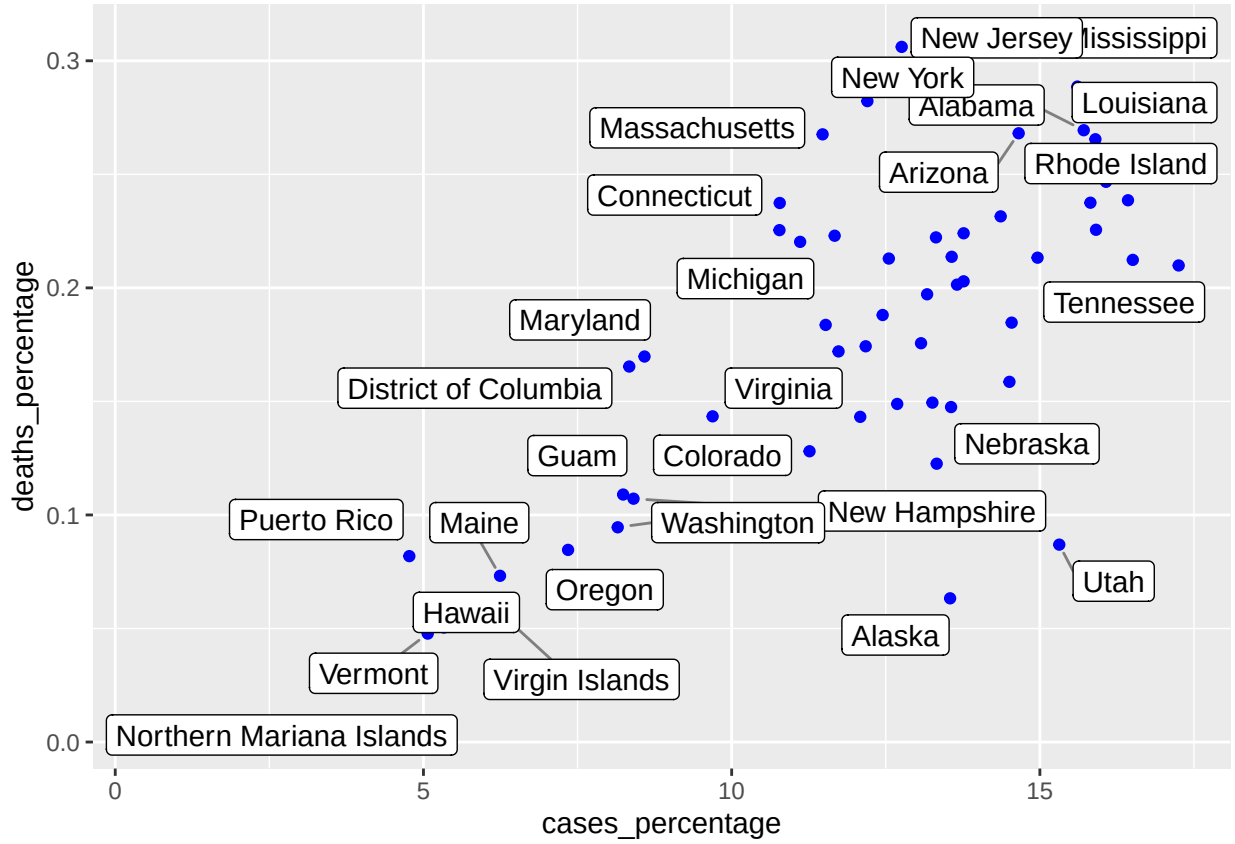
Lets do the same thing for US state data.

```
US_total <- US_by_state %>%
  group_by(Province_State) %>%
  summarize(deaths = max(deaths), cases = max(cases),
            population = max(Population),
            cases_percentage = (cases / population)*100,
            deaths_percentage = (deaths / population)*100,
            death_cases_per =(deaths / cases)*100) %>%
  filter(cases > 0, population > 0)
```

```
US_total %>%
  ggplot(aes(x = cases_percentage, y = deaths_percentage )) +
  geom_point(color="blue")+ geom_label_repel(aes(label = Province_State),
```

```
                                        box.padding=0.35,
                                        point.padding = 0.5,
                                        segment.color = 'grey50')
```

```
## Warning: ggrepel: 27 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```



The above graphs show us the relation between the cases_percentage and deaths_percentage.

Now let us take look at the death to cases percentage with countries that have population greater than 1 million to find out the top 10 worst countries.

```
global_total %>%
  filter(population>1000000)%>%
  select(Country_Region,deaths,cases,population,death_cases_per)%>%
  slice_max(death_cases_per, n = 10)
```

```
## # A tibble: 10 x 5
##    Country_Region deaths   cases population death_cases_per
##    <chr>           <dbl>   <dbl>      <dbl>           <dbl>
##  1 Yemen            1643    8667   29825968           19.0
##  2 Peru           199066 2167008   32971846            9.19
##  3 Mexico         271503 3569677  127792286            7.61
##  4 Sudan            2878   37995   43849269            7.57
##  5 Syria            2127   30709   17500657            6.93
```

```
##  6 Ecuador         32661  507003   17643060            6.44
##  7 Egypt           16970  296929  102334403            5.72
##  8 Somalia          1063   19004   15893219            5.59
##  9 Taiwan*           840   16141   23816775            5.20
## 10 Liberia           283    5904    5057677            4.79
```

Now let us find the top 10 worst states in US. Here worst state refer to the states where the percentage of death to cases is maximum. Here we also take the states with the populations greater than 10000 to exclude the islands and other smaller states with very small population.

```
US_total %>%
  filter(population>10000)%>%
  select(Province_State,deaths,cases,population,death_cases_per)%>%
  slice_max(death_cases_per, n = 10)
```

```
## # A tibble: 10 x 5
##    Province_State       deaths   cases population death_cases_per
##    <chr>                 <dbl>   <dbl>      <dbl>           <dbl>
##  1 New Jersey            27190 1133228    8882190            2.40
##  2 Massachusetts         18445  790953    6892503            2.33
##  3 New York              54904 2373659   19453561            2.31
##  4 Connecticut            8463  384342    3565287            2.20
##  5 Pennsylvania          28858 1379478   12801989            2.09
##  6 District of Columbia   1167   58851     705749            1.98
##  7 Michigan              21997 1109643    9986857            1.98
##  8 Maryland              10263  519097    6045680            1.98
##  9 Mississippi            9214  473413    2976149            1.95
## 10 New Mexico             4675  244720    2096829            1.91
```

Here we can build a model on the bases of death to cases percentage with the population of the US state

```
mod1 <- lm(death_cases_per~population, data = US_total)
summary(mod1)
```

```
##
## Call:
## lm(formula = death_cases_per ~ population, data = US_total)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.92101 -0.29200 -0.03219  0.26812  0.88171
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.377e+00  7.226e-02  19.050   <2e-16 ***
## population  1.588e-08  7.702e-09   2.062   0.0441 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4096 on 53 degrees of freedom
## Multiple R-squared:  0.07429,    Adjusted R-squared:  0.05682
## F-statistic: 4.253 on 1 and 53 DF,  p-value: 0.04409
```

Conclusion

From the various graphs we can see that Covid cases and deaths follow a similar trends. although the scale of the graph is logarithmic, we can see that the pecks are the times when a specific wave is going on and the number of cases are reduction because of the people getting vaccinated.

ALso according to the above analysis, we can say that the states with a higher death to cases percentage have a lower medical infrastructure compared to the ones that have a lesser. As the states with lower death to cases percentage, here the people got better medical care and their better treatment led to a speedy recovery. Thus the states with higher death to cases percentage, couldn't provide better medical care and that might be one of the reasons for a higher death count.

Possible sources of bias

Here we need to take a look at the possible sources of bias as there can be reports with data that already have a bias and also those whole introduce a bias while in the report.

Here one of the possible sources of the bias is the data source , as we do not know that how the data is being calculated had how it is being reported. Another possible source of bias can be the medical data reported, we have come across various cases where the cases are under reported or the cause of death has been stated as something else although Covid with the reason the patient went into that state. Another possible source of bias is the percentage of population that was too scared to get them self tested for covid 19 when their first symptoms stated to emerge. Another possible source of bias can be me, as I have developed the report based on my understanding thus a bias in my understand can get translated into the report.