Johns Hopkins Covid-19 Data Analysis

Shyam Menon

06/04/2021

The data was obtained from COVID-19 Data Repository by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University.

Covid-19 changed the world as we know it. Within on year all systems around the world were affected and changed. There are many lessons to learn from this pandemic and the data is our way of figuring out what, where, and how everything happened. With this data we want to see the trends of disease throughout the United States. Which areas were affected more than others? Were there big differences or changes over time? Etc.

Dataset Maniuplation

Importing the Data: CCSE Covid-19 Time Series Data

The data is from Johns Hopkins Center for Systems Science and Engineering. The data sets that are used are the confirmed global cases, deaths for global cases, confirmed US cases, and deaths for US cases.

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

Summary of Field Descriptions

- Province_State - The name of the State within the USA.
- Country_Region - The name of the Country (US).
- Last_Update - The most recent date the file was pushed.
- Lat - Latitude.
- Long_ - Longitude.
- Confirmed - Aggregated case count for the state.
- Deaths - Aggregated death toll for the state.
- Recovered - Aggregated Recovered case count for the state.

- Active - Aggregated confirmed cases that have not been resolved (Active cases = total * cases - total recovered - total deaths).
- FIPS - Federal Information Processing Standards code that uniquely identifies counties within the USA.
- Incident_Rate - cases per 100,000 persons.
- Total_Test_Results - Total number of people who have been tested.
- People_Hospitalized - Total number of people hospitalized. (Nullified on Aug 31, see Issue #3083)
- Case_Fatality_Ratio - Number recorded deaths * 100/ Number confirmed cases.
- UID - Unique Identifier for each row entry.
- ISO3 - Officialy assigned country code identifiers.
- Testing_Rate - Total test results per 100,000 persons. The "total test results" are equal to "Total test results (Positive + Negative)" from COVID Tracking Project.
- Hospitalization_Rate - US Hospitalization Rate (%): = Total number hospitalized / Number cases. The "Total number hospitalized" is the "Hospitalized – Cumulative" count from COVID Tracking Project. The "hospitalization rate" and "Total number hospitalized" is only presented for those states which provide cumulative hospital data.

Cleaning and Transforming the Covid-19 Data

Getting rid of unneeded columns such as Lat/Long. Then changing the date from string to a date format. After cleaning the individual data sets, they are aggregated into a complete set for both global and US covid-19 data.

```
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))

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)

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

uid_loopup_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/U

uid <- read_csv(uid_loopup_url) %>%
    select(-c(Lat, Long_, Combined_Key, code3, iso2, iso3, Admin2))

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 Analysis

Summarizing the Data

Here we summarize the data for the US at both the state levels and the overall USA nationwide level.

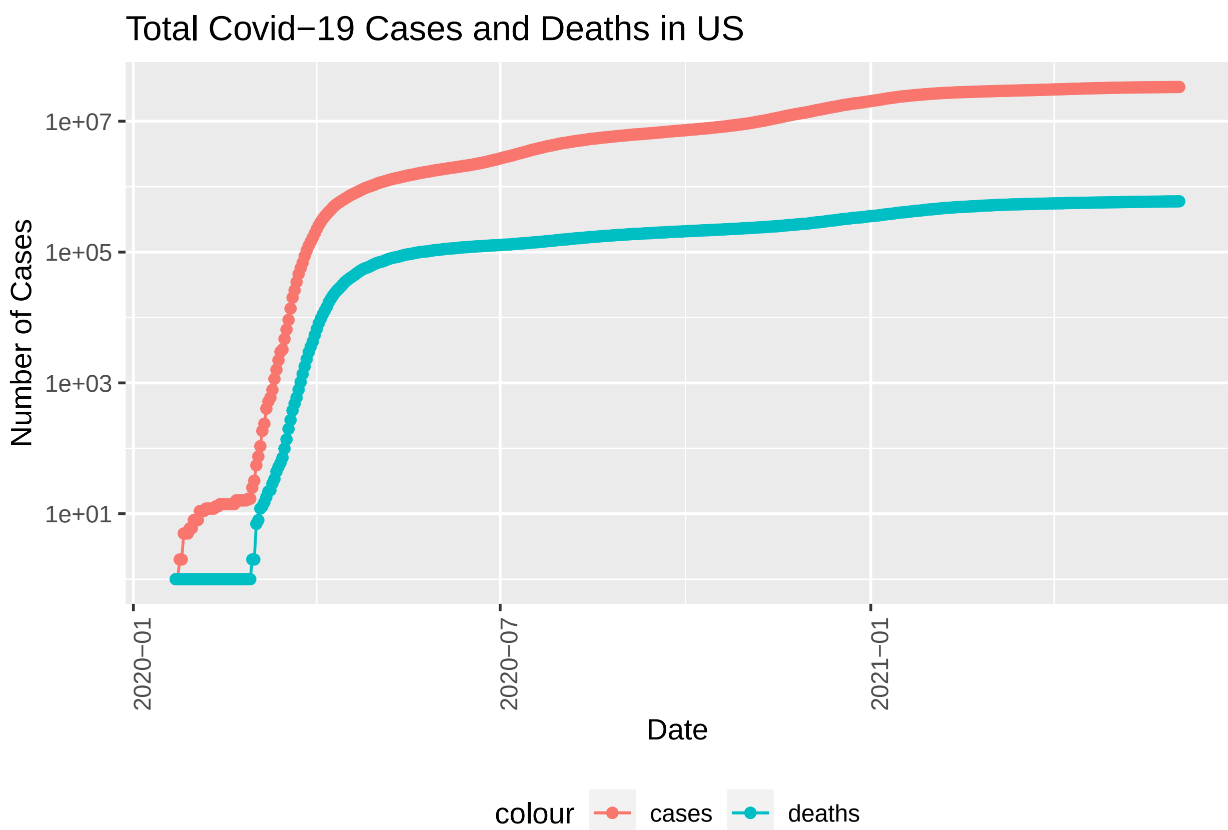
```
US_by_state <- US %>%
  group_by(Province_State, Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths),
            Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths *1000000 / Population) %>%
  select(Province_State, Country_Region, date,
         cases,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


US_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()
```

Visualizing the Data - United States Nationwide Totals

With this visualization we can see the total number of US covid-19 cases and deaths over time. As you can see from the graph, there is rapid spread of cases and deaths across the US.

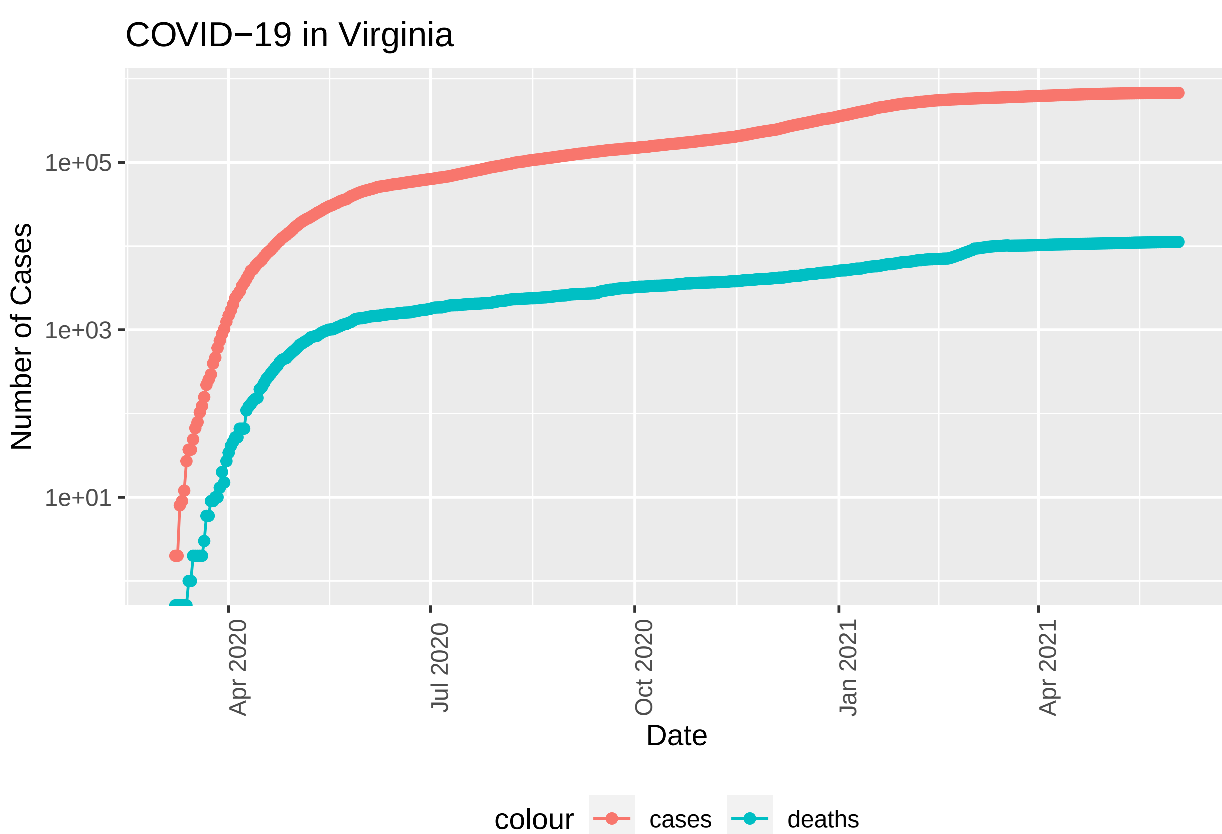
```
US_totals %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = cases)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "Total Covid-19 Cases and Deaths in US", y = "Number of Cases",
       x = "Date")
```



Visualizing the Data - Statewide Totals (Virginia)

With this visualization we can see the total number of statewide covid-19 cases and deaths over time. As you can see from the graph, there is rapid spread of cases and deaths across the state.

```
state <- "Virginia"
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("COVID-19 in ", state), y = "Number of Cases",
       x = "Date")
```



Further Analysis - Gathering new cases

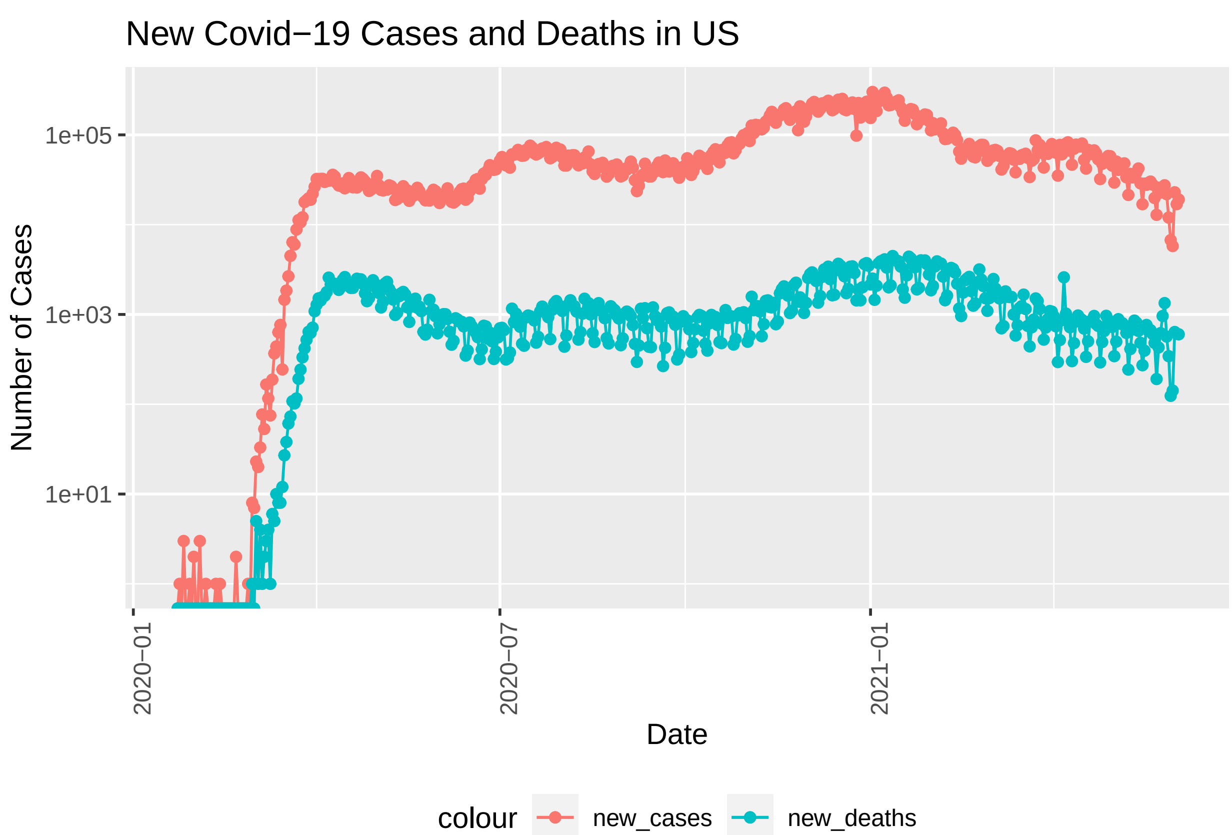
Another question of interest is trying to find the number of new cases that are occurring. Here we are gathering the number of new cases for both state and US.

```
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))
```

Visualizing the New Cases in the US

Here we can see the number of new covid-19 cases in the United States.

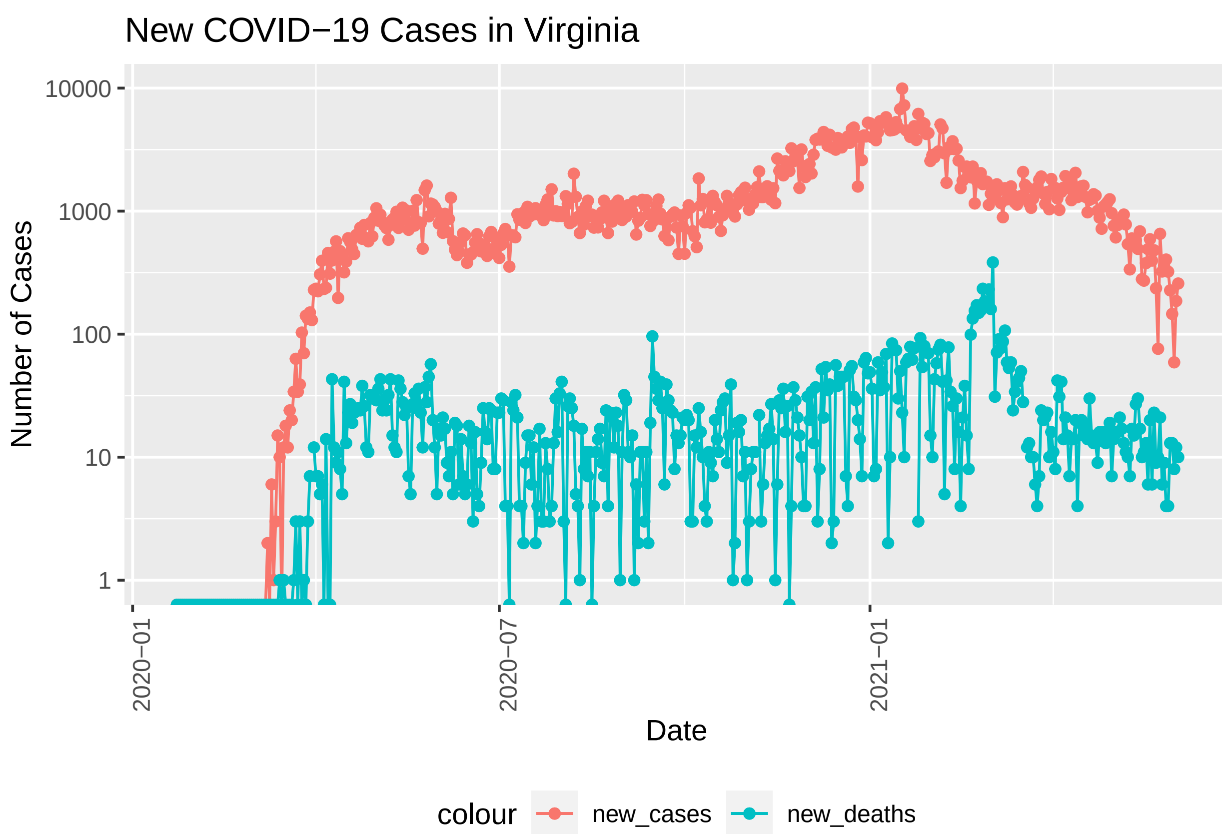
```
US_totals %>%
  ggplot(aes(x = date, y = new_cases)) +
  geom_line(aes(color = "new_cases")) +
  geom_point(aes(color = "new_cases")) +
  geom_line(aes(y = new_deaths, color = "new_deaths")) +
  geom_point(aes(y = new_deaths, color = "new_deaths")) +
  scale_y_log10() +
  theme(legend.position =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



Visualizing the New Cases in Virginia

Here we can see the number of new covid-19 cases in the state of Virginia.

```
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("New COVID-19 Cases in ", state), y = "Number of Cases",
       x = "Date")
```



Further Analysis - Calculating deaths and cases per thousand

Another question of interest is trying to find the number of deaths and cases that are occurring per thousand and see which states had the highest and lowest of the range.

```
US_state_totals <- US_by_state %>%
  group_by(Province_State) %>%
  summarize(deaths = max(deaths), cases = max(cases),
```

```r
            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) %>%
  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          0.261           32.7  Virgin Islands            28   3512     107268
##  3          0.354           25.7  Hawaii                   501  36402    1415872
##  4          0.409           38.8  Vermont                  255  24240     623989
##  5          0.498           95.0  Alaska                   369  70408     740995
##  6          0.623           50.6  Maine                    837  67986    1344212
##  7          0.636           48.0  Oregon                  2683 202247    4217737
##  8          0.670           37.0  Puerto Rico             2515 138873    3754939
##  9          0.720          127.   Utah                    2308 406825    3205958
## 10          0.764           57.6  Washington              5821 438544    7614893
```

```r
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            2.96          115.  New Jersey      26253 1017044    8882190
##  2            2.74          108.  New York        53357 2103768   19453561
##  3            2.60          103.  Massachusetts   17893  707523    6892503
##  4            2.56          143.  Rhode Island     2715  151936    1059361
##  5            2.46          107.  Mississippi      7324  318048    2976149
##  6            2.43          121.  Arizona         17653  882691    7278717
##  7            2.31           97.5 Connecticut      8247  347748    3565287
##  8            2.28          140.  South Dakota     2020  124242     884659
##  9            2.28          111.  Alabama         11188  545028    4903185
## 10            2.28          102.  Louisiana       10605  472617    4648794
```

Modeling the data

Linear model, deaths per thousand as a function of cases per thousand. The model shows that it does a good job at predicting at the lower end, while at the higher ends there might be other factors that also come into play.

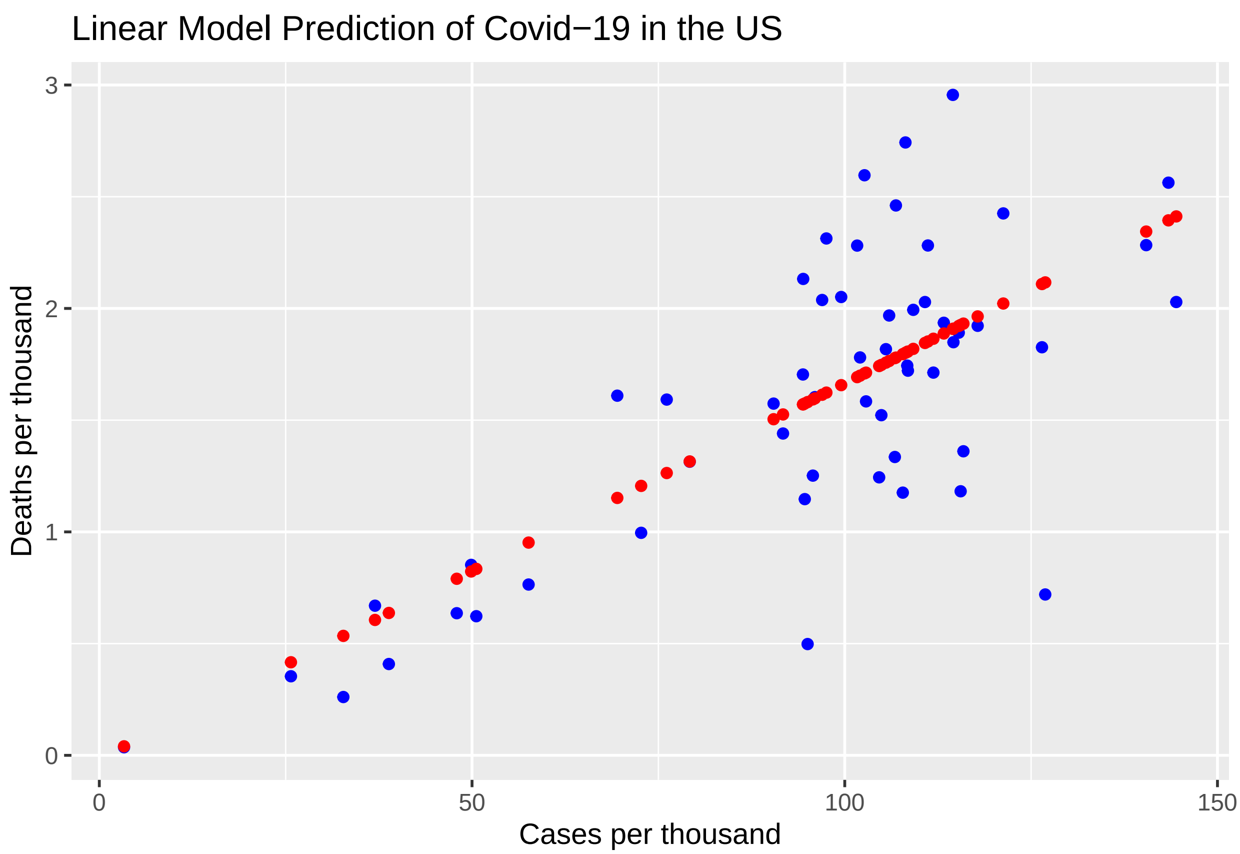
```r
mod <- lm(deaths_per_thou ~ cases_per_thou, data = US_state_totals)
summary(mod)
```

```
##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou, data = US_state_totals)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.39665 -0.21830 -0.03013  0.19337  1.04734
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.015367   0.209201  -0.073    0.942
## cases_per_thou  0.016800   0.002106   7.979 1.21e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4618 on 53 degrees of freedom
## Multiple R-squared:  0.5457, Adjusted R-squared:  0.5372
## F-statistic: 63.67 on 1 and 53 DF,  p-value: 1.207e-10
```

```r
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          11188 5.45e5    4903185          111.            2.28   1.85
##  2 Alaska             369 7.04e4     740995           95.0           0.498  1.58
##  3 Arizona          17653 8.83e5    7278717          121.            2.43   2.02
##  4 Arkansas          5842 3.42e5    3017804          113.            1.94   1.89
##  5 California       63345 3.79e6   39512223           96.0           1.60   1.60
##  6 Colorado          6603 5.45e5    5758736           94.6           1.15   1.57
##  7 Connecticut       8247 3.48e5    3565287           97.5           2.31   1.62
##  8 Delaware          1668 1.09e5     973764          112.            1.71   1.86
##  9 District of Co~   1136 4.90e4     705749           69.5           1.61   1.15
## 10 Florida          36973 2.33e6   21477737          108.            1.72   1.81
## # ... with 45 more rows
```

```r
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") +
  labs(title = str_c("Linear Model Prediction of Covid-19 in the US"),
       y = "Deaths per thousand", x = "Cases per thousand")
```



The model is shown here by the red data points, and the covid data set is represented in blue.

Conclusion

The Johns Hopkins Covid-19 data set was very interesting to analyze. The distribution of covid throughout the United States vary greatly from state to state. Nonetheless, the data shows the spread of covid-19 has been rapid from the onset of the very first cases to present day time. This is critical to understand in order to prepare ourselves for future diseases and/or other public health crises.The data collected provides valuable insights that everyone should learn from.

While the data collected was vast and plentiful, there is still many variables that come into play. The model that was assessed predicts a linear fashion of distribution, which was seen for the lower stages of cases; however, there was an increase in variation as the cases per thousand increased. This indicates that there are multiple variables that affect the death rates as the cases increases over time.

It should be mentioned that bias could also play a major role in data collection and reporting. In terms of data collection, people may be reluctant to report their covid symptoms or there may be some patients that had covid but was asymptomatic. Therefore, the data collected must always be assessed for bias. Another form of bias is selection bias when analyzing the data. All groups must be considered and randomized testing should be implemented in order to reduce bias. Also, one must keep in mind that there are can be racial and ethnic bias in reporting and collecting of data as well.

Covid-19 changed the world as we know it. Within one year all systems around the world were affected and changed. There are many lessons to learn from this pandemic and the data is our way of figuring out what, where, and how everything happened. There are many variables to consider and one should always have a cyclical process of data science analysis. It is a never ending process of learning.