COVID-19 Data Analysis

1. Importing Data

Obtain the COVID-19 Data from Johns Hopkins on GitHub:

```
library(tidyverse)
library(lubridate)
url_in =
  "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_ti

file_names = c("time_series_covid19_confirmed_US.csv",
               "time_series_covid19_confirmed_global.csv",
               "time_series_covid19_deaths_US.csv",
               "time_series_covid19_deaths_global.csv",
               "time_series_covid19_recovered_global.csv")

urls = str_c(url_in, file_names)
```

Read using '<read_csv()>'...

```
global_cases = read_csv(urls[2])
global_deaths = read_csv(urls[4])
US_cases = read_csv(urls[1])
US_deaths = read_csv(urls[3])
```

2. Tidy Data

Time to tidy up the data and make it more "R" friendly. We will put the following variables in their own column: (date, cases, and deaths). Additionally, I will get rid of Lat and Long since I don't plan on doing any analysis with them. I will also rename the region and state so they are more tidy.

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

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

```
## # A tibble: 260,490 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 260,480 more rows
```

As we can see, the date column has also be changed to a date object and has been given its own row for the combined global data.

Printing summary of data...

```
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:260490      Length:260490      Min.   :2020-01-22   Min.   :       0
##  Class :character   Class :character   1st Qu.:2020-09-06   1st Qu.:     362
##  Mode  :character   Mode  :character   Median :2021-04-22   Median :    8240
##                                        Mean   :2021-04-22   Mean   :  657516
##                                        3rd Qu.:2021-12-07   3rd Qu.:  145498
##                                        Max.   :2022-07-23   Max.   :90390185
##      deaths
##  Min.   :      0
##  1st Qu.:      2
##  Median :    100
##  Mean   :  11060
##  3rd Qu.:   2191
##  Max.   :1026937
```

Looking at this summary shows me there is likely a number of rows without cases and consequently also no deaths. We can filter out these rows since there is unlikely any useful data that can provide us.

```
global = global %>%
  filter(cases>0)

summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:240791      Length:240791      Min.   :2020-01-22   Min.   :       1
##  Class :character   Class :character   1st Qu.:2020-10-12   1st Qu.:     836
##  Mode  :character   Mode  :character   Median :2021-05-20   Median :   12385
##                                        Mean   :2021-05-17   Mean   :  711307
##                                        3rd Qu.:2021-12-22   3rd Qu.:  180526
##                                        Max.   :2022-07-23   Max.   :90390185
##      deaths
```

```
##  Min.   :      0
##  1st Qu.:      6
##  Median :    147
##  Mean   :  11964
##  3rd Qu.:   2726
##  Max.   :1026937
```

Let's make sure that maximum number of cases is accurate by filtering out cases >90,000,000

```
global %>%
  filter(cases>90000000)
```

```
## # A tibble: 4 x 5
##   Province_State Country_Region date          cases  deaths
##   <chr>          <chr>          <date>        <dbl>   <dbl>
## 1 <NA>           US             2022-07-20 90046261 1025763
## 2 <NA>           US             2022-07-21 90200438 1026294
## 3 <NA>           US             2022-07-22 90367064 1026883
## 4 <NA>           US             2022-07-23 90390185 1026937
```

This initial check shows the data for July 2022 which coincides with when I am pulling this data.

Now I will repeat this tidying and transforming of the COVID19 US cases dataset.

```
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:Combined_Key),
                        names_to = "date",
                        values_to = "deaths") %>%
  select(Admin2:deaths) %>%
  mutate(date = mdy(date)) %>%
  select(-c(Lat,Long_))

US = US_cases %>%
  full_join(US_deaths) %>%
  filter(cases>0)
```

I've noticed that the US data set has a "population" variable while the global data set does not. I will transform the data some more to get them to look more similar to each other for ease of comparison later on.

```
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))

global = global %>%
  left_join(uid, by= c("Province_State", "Country_Region")) %>%
  select(-c(UID,FIPS)) %>%
  select(Province_State, Country_Region, date, cases, deaths,Combined_Key)

global
```

```
## # A tibble: 240,791 x 6
##    Province_State Country_Region date       cases deaths Combined_Key
##    <chr>          <chr>          <date>     <dbl>  <dbl> <chr>
##  1 <NA>           Afghanistan    2020-02-24     5      0 Afghanistan
##  2 <NA>           Afghanistan    2020-02-25     5      0 Afghanistan
##  3 <NA>           Afghanistan    2020-02-26     5      0 Afghanistan
##  4 <NA>           Afghanistan    2020-02-27     5      0 Afghanistan
##  5 <NA>           Afghanistan    2020-02-28     5      0 Afghanistan
##  6 <NA>           Afghanistan    2020-02-29     5      0 Afghanistan
##  7 <NA>           Afghanistan    2020-03-01     5      0 Afghanistan
##  8 <NA>           Afghanistan    2020-03-02     5      0 Afghanistan
##  9 <NA>           Afghanistan    2020-03-03     5      0 Afghanistan
## 10 <NA>           Afghanistan    2020-03-04     5      0 Afghanistan
## # ... with 240,781 more rows
```

3. Visualize Data

Now that we have gotten the two data sets to look similar to each other we can move ahead with visualizing the given data.

I will first group together the US data by state and calculate the death rate in that state per day.

```
US_by_state = US %>%
  group_by(Province_State, Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths)) %>%
  mutate(death_rate = deaths/cases) %>%
  select(Province_State, Country_Region, date, cases, deaths, death_rate) %>%
  ungroup()

US_by_state
```

```
## # A tibble: 49,934 x 6
##    Province_State Country_Region date       cases deaths death_rate
##    <chr>          <chr>          <date>     <dbl>  <dbl>      <dbl>
##  1 Alabama        US             2020-03-11     3      0          0
##  2 Alabama        US             2020-03-12     4      0          0
##  3 Alabama        US             2020-03-13     8      0          0
##  4 Alabama        US             2020-03-14    15      0          0
##  5 Alabama        US             2020-03-15    28      0          0
```

```
##  6 Alabama        US             2020-03-16    36      0          0
##  7 Alabama        US             2020-03-17    51      0          0
##  8 Alabama        US             2020-03-18    61      0          0
##  9 Alabama        US             2020-03-19    88      0          0
## 10 Alabama        US             2020-03-20   115      0          0
## # ... with 49,924 more rows
```

Now I will look at US totals for cases and deaths on a given day

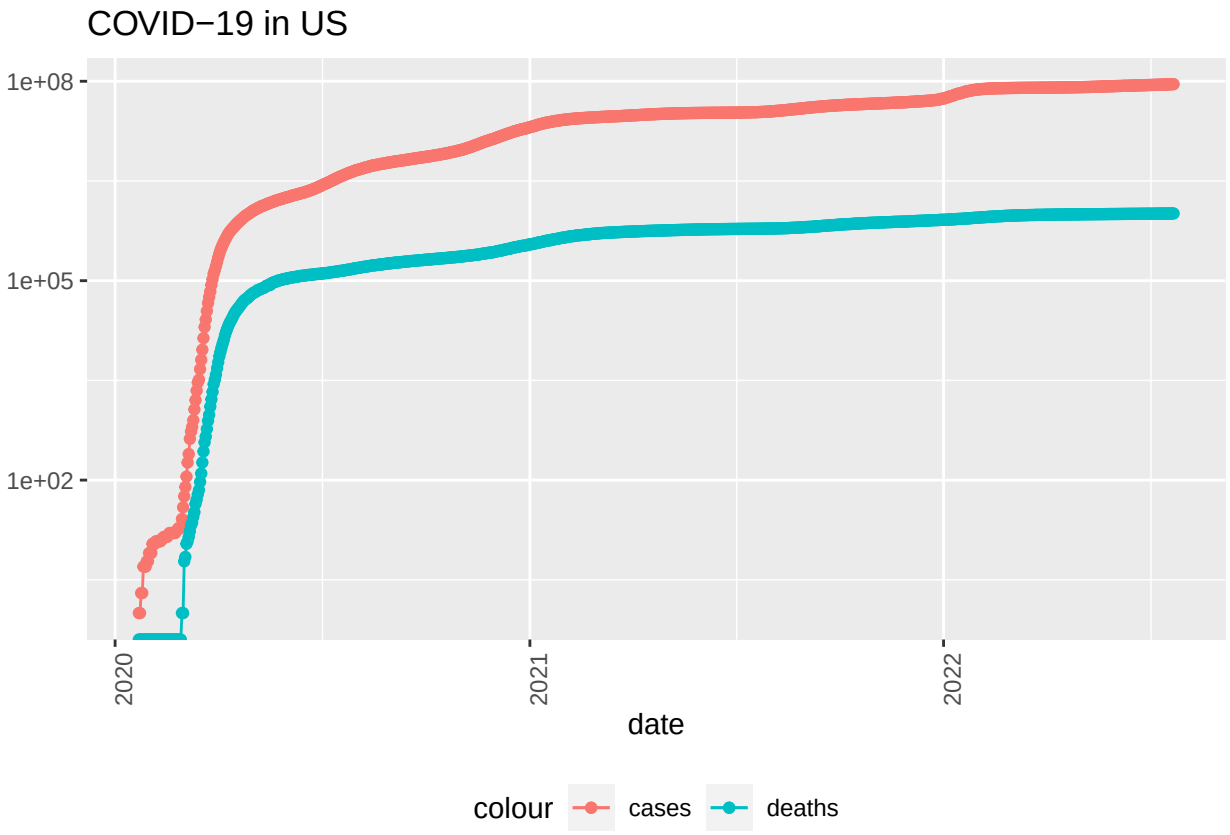
```
US_totals = US_by_state %>%
  group_by(Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths)) %>%
  mutate(death_rate = deaths/cases) %>%
  select(Country_Region, date, cases, deaths, death_rate) %>%
  ungroup()

US_totals
```

```
## # A tibble: 914 x 5
##    Country_Region date       cases deaths death_rate
##    <chr>          <date>     <dbl>  <dbl>      <dbl>
##  1 US             2020-01-22     1      0          0
##  2 US             2020-01-23     1      0          0
##  3 US             2020-01-24     2      0          0
##  4 US             2020-01-25     2      0          0
##  5 US             2020-01-26     5      0          0
##  6 US             2020-01-27     5      0          0
##  7 US             2020-01-28     5      0          0
##  8 US             2020-01-29     6      0          0
##  9 US             2020-01-30     6      0          0
## 10 US             2020-01-31     8      0          0
## # ... with 904 more rows
```

Time to visualize the total US cases vs deaths...

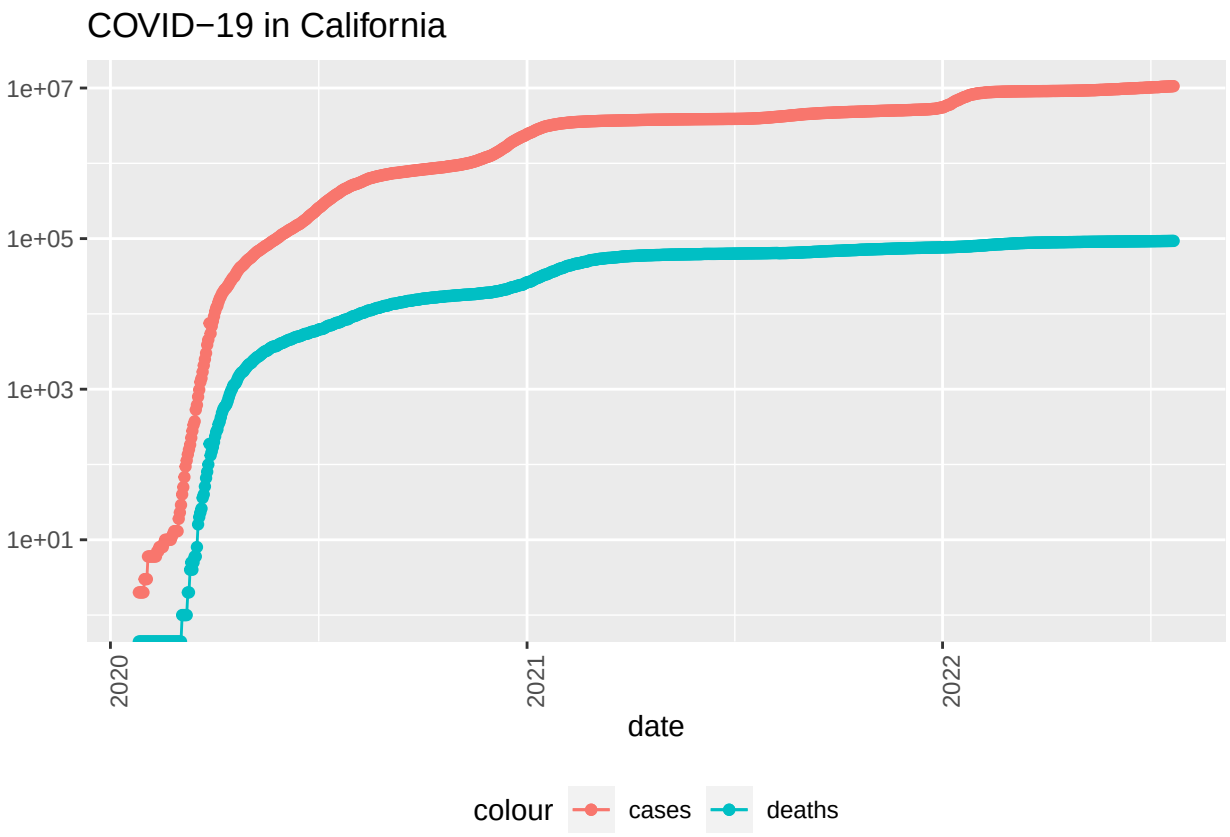
```
US_totals %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = cases)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases"))+
  geom_line(aes(y = deaths, color = "deaths"))+
  geom_point(aes(y = deaths, color = "deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID-19 in US", y = NULL)
```



We can see the deaths curve is reasonably shifted down but otherwise follows the same pattern as the cases.

Lets visualize the total cases vs. deaths in the state of California...

```
state = "California"
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 = NULL)
```



We can see the shape of California's data is very close to the shape of the US totals data. This is likely due to the high population of California relative to the US. We can also see the pattern of the cases vs deaths is consistent with the entire US, both seemingly suggesting the deaths and cases have leveled off.

Let's look at what date had the maximum deaths in California and all of the US.

```
max(US_by_state$date)
```

```
## [1] "2022-07-23"
```

```
max(US_by_state$deaths)
```

```
## [1] 93209
```

```
max(US_totals$date)
```

```
## [1] "2022-07-23"
```

```
max(US_totals$deaths)
```

```
## [1] 1026274
```

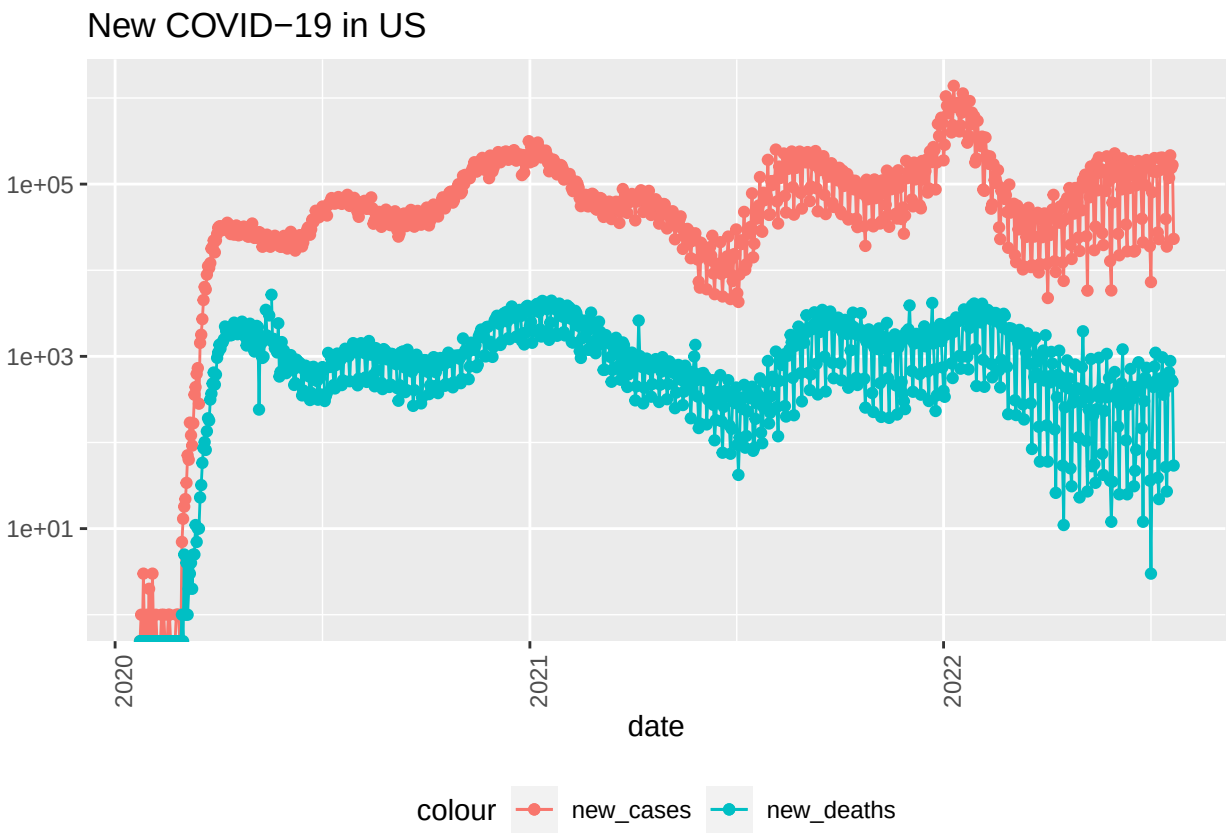
Running this analysis that the max deaths to date is today, suggesting the deaths and cases are still increasing. This begs questioning of whether or not the cases have truly leveled off.

We will add new columns to the existing data sets so that we can see the new cases and new deaths everyday.

```
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))
```

Now that we have added these new columns, let's visualize the data once more.

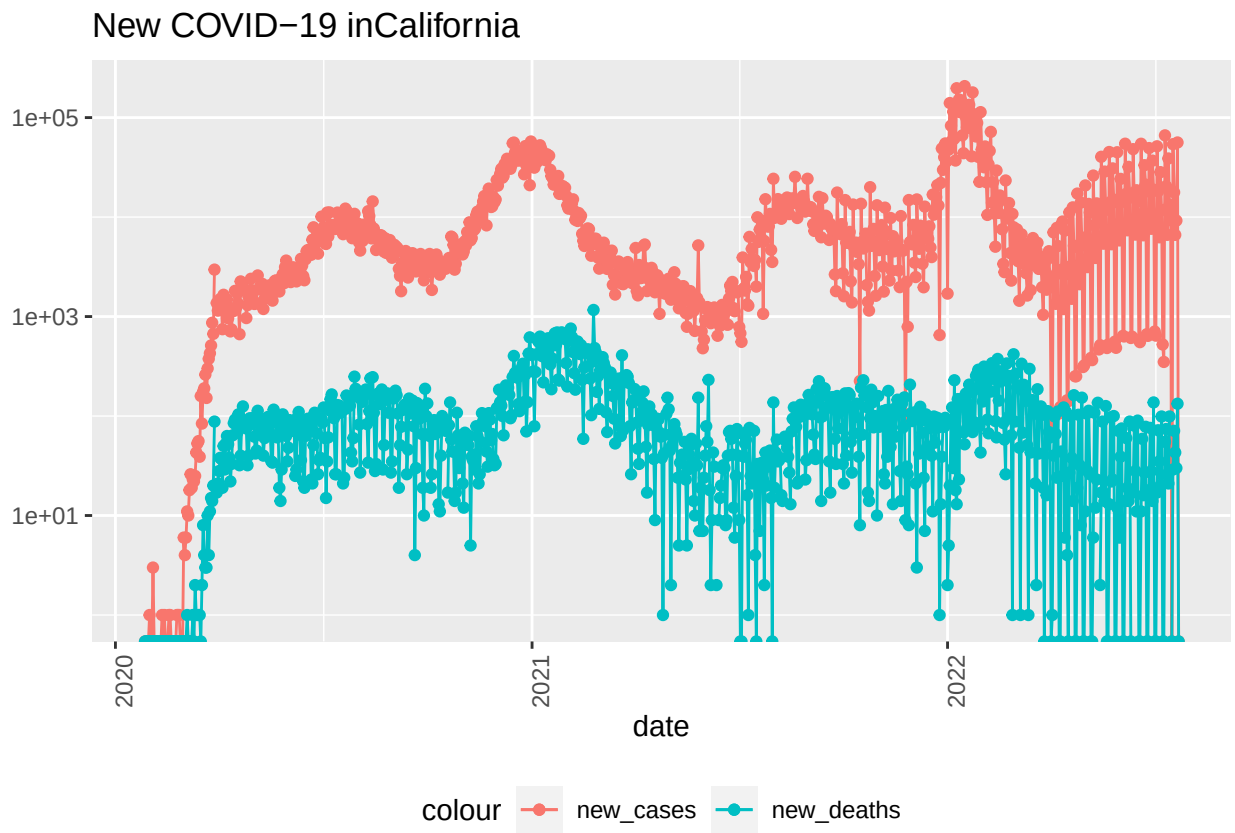
```
US_totals %>%
  ggplot(aes(x = date, y = new_cases)) +
  geom_line(aes(color = "new_cases"))+
  geom_point(aes(color= "new_cases")) +
  geom_line(aes(y = new_deaths, color = "new_deaths")) +
  geom_point(aes(y = new_deaths, color = "new_deaths"))+
  scale_y_log10()+
  theme(legend.position = "bottom",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



Here we can see more variation in the graph more indicative of rising and falling counts of cases and deaths per day.

Let's take a look at how California is doing. . .

```
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 in", state), y = NULL)
```



After looking at one state's data, it seemed logical to look at other state's data. Namely which state is the worst off and which is the best off? There are different ways to go about approaching this..

But first lets transform the data again. . . and look for the 10 states with the lowest death rate.

```
US_state_totals = US_by_state %>%
  group_by(Province_State) %>%
  summarize(deaths = max(deaths), cases = max(cases)) %>%
  mutate(death_rate =deaths/cases ) %>%
  filter(cases >0)


US_state_totals %>%
  slice_min(death_rate, n = 10) %>%
  select(Province_State, death_rate, everything())
```

```
## # A tibble: 10 x 4
##    Province_State           death_rate deaths   cases
##    <chr>                         <dbl>  <dbl>   <dbl>
##  1 Diamond Princess            0           0      49
##  2 Northern Mariana Islands    0.00282    35   12398
##  3 American Samoa              0.00442    33    7471
##  4 Alaska                      0.00466  1309  281203
##  5 Hawaii                      0.00481  1548  321869
##  6 Utah                        0.00486  4884 1004426
##  7 Vermont                     0.00502   689  137264
##  8 Virgin Islands              0.00556   120   21569
##  9 Puerto Rico                 0.00566  4723  833809
## 10 Guam                        0.00696   377   54163
```

And now the highest death rate...

```
US_state_totals %>%
  slice_max(death_rate, n = 10)%>%
  select(Province_State, death_rate, everything())
```

```
## # A tibble: 10 x 4
##    Province_State death_rate deaths   cases
##    <chr>               <dbl>  <dbl>   <dbl>
##  1 Grand Princess     0.0291      3     103
##  2 Pennsylvania       0.0151  46047 3058316
##  3 Oklahoma           0.0146  16216 1108553
##  4 Mississippi        0.0146  12603  866040
##  5 Georgia            0.0143  38933 2721391
##  6 Nevada             0.0142  11189  789674
##  7 Alabama            0.0141  19872 1407699
##  8 Arizona            0.0141  30698 2179180
##  9 Michigan           0.0141  37291 2652659
## 10 New Mexico         0.0140   8191  584404
```

Looking at this data can tell us a lot, but looking into more detail we can see the numbers for less populated states could have a skewed rate. This could indicate there is a better way to represent this data more uniformly.

3. Model Data

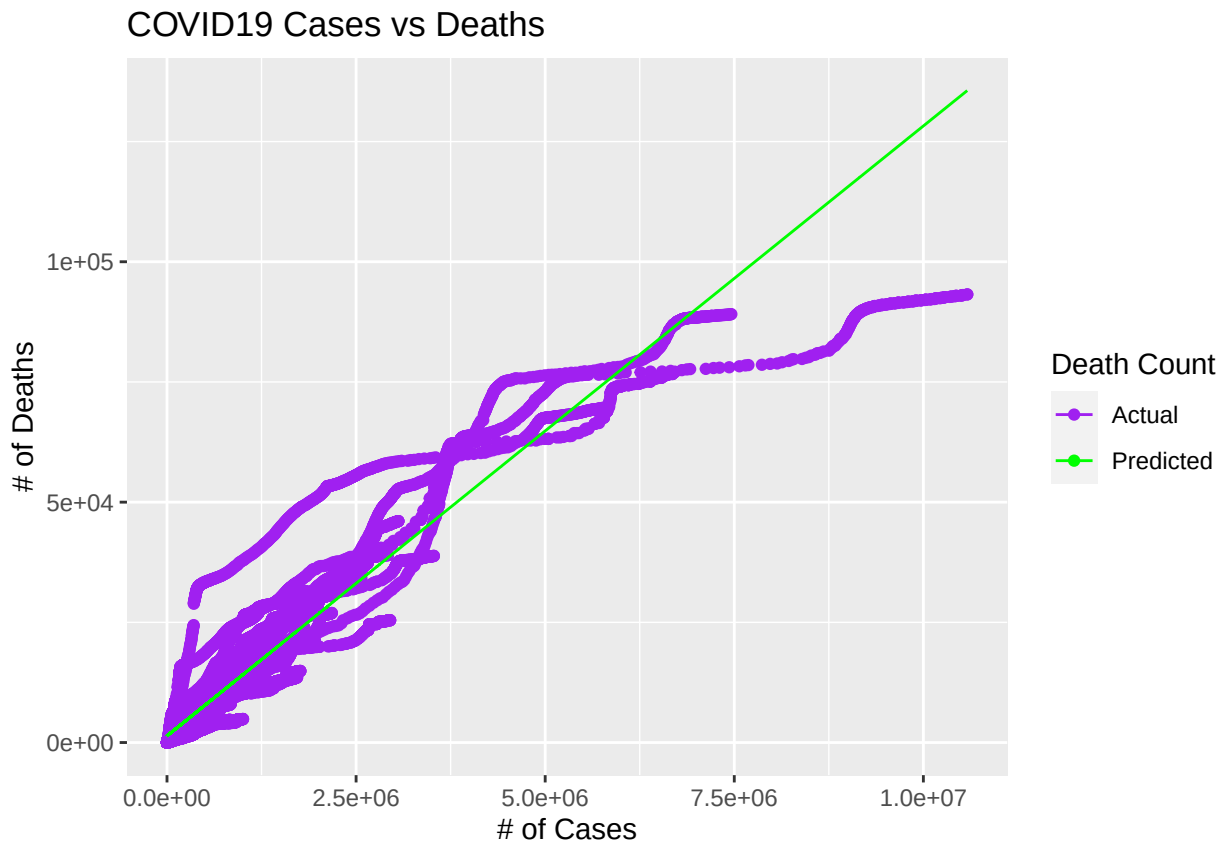
Let's apply a model our working data set. I chose to model deaths as a linear function of cases..

```
mod = lm(deaths ~ cases, data = US_by_state)
summary(mod)
```

```
##
## Call:
## lm(formula = deaths ~ cases, data = US_by_state)
##
```

```
## Residuals:
##    Min     1Q Median     3Q    Max
## -42395  -1431  -1138    610  25837
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.389e+03  2.384e+01   58.26   <2e-16 ***
## cases       1.269e-02  1.879e-05  675.17   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4632 on 49932 degrees of freedom
## Multiple R-squared:  0.9013, Adjusted R-squared:  0.9013
## F-statistic: 4.559e+05 on 1 and 49932 DF,  p-value: < 2.2e-16
```

```
US_tot_w_pred = US_by_state %>%
  mutate(pred = predict(mod))
US_tot_w_pred %>% ggplot() +
  geom_point(aes(x = cases, y = deaths, color = "Actual")) +
  geom_line(aes(x = cases, y = pred, color = "Predicted"))+
  scale_color_manual(name = "Death Count", values = c("Actual" = "purple", "Predicted" = "green"))+
  xlab("# of Cases")+
  ylab("# of Deaths")+
  ggtitle("COVID19 Cases vs Deaths")
```



Here we see that for a period of time (at the lower end of case numbers i.e. earlier on in the pandemic) the linear model predicts the death counts quite well. However, later on we see more of a plateau shape in

the actual cases vs death data points, indicating although cases were increasing the death rates decreased. This can possibly be explained by the implementation of government enforced lockdowns and with the roll out of vaccinations nation wide. With this observed shape, it may be more appropriate to assume that an exponential decay or logarithmic model would be more appropriate.

4. Conclusions

Some conclusions we can draw from this data set include the following:

- Over the course of a given year there has been some leveling off across the US of deaths as a result of COVID-19.
- The case (and death) rate follow a pattern of peaking and declining at similar times during the year.
- We can also see the lower populated states generally have a lower death rate.
- As time has passed during this pandemic, the cases vs deaths relationship has started to deviate from a linear relationship and resemble more of an exponential decay relationship.

Possible sources of bias that could have been influenced in this analysis include my experience in this pandemic could have made me more interested in aspects of this data that are more relevant to me personally. This could be looking into my home state or states I want to visit. In relation to global data I could be biased towards different countries I've visited or attribute high death rates in third world countries to proverty without any data to support that conclusion. Knowing these possible sources of bias, I tried to include looking at different places I don't have any personal ties to to give a better overall picture of the data set.

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-apple-darwin20.4.0 (64-bit)
## Running under: macOS Monterey 12.3
##
## Matrix products: default
## BLAS:   /usr/local/Cellar/openblas/0.3.17/lib/libopenblasp-r0.3.17.dylib
## LAPACK: /usr/local/Cellar/r/4.1.1/lib/R/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.8.0 forcats_0.5.1   stringr_1.4.0   dplyr_1.0.7
##  [5] purrr_0.3.4     readr_2.1.0     tidyr_1.1.4     tibble_3.1.6
##  [9] ggplot2_3.3.5   tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.7       assertthat_0.2.1 digest_0.6.28    utf8_1.2.2
##  [5] R6_2.5.1         cellranger_1.1.0 backports_1.4.0  reprex_2.0.1
##  [9] evaluate_0.14    httr_1.4.2       highr_0.9        pillar_1.6.4
## [13] rlang_0.4.12     curl_4.3.2       readxl_1.3.1     rstudioapi_0.13
## [17] rmarkdown_2.11   labeling_0.4.2   bit_4.0.4        munsell_0.5.0
## [21] broom_0.7.10     compiler_4.1.1   modelr_0.1.8     xfun_0.28
## [25] pkgconfig_2.0.3  htmltools_0.5.2  tidyselect_1.1.1 fansi_0.5.0
## [29] crayon_1.4.2     tzdb_0.2.0       dbplyr_2.1.1     withr_2.4.2
```

```
## [33] grid_4.1.1       jsonlite_1.7.2   gtable_0.3.0     lifecycle_1.0.1
## [37] DBI_1.1.1        magrittr_2.0.1   scales_1.1.1     cli_3.1.0
## [41] stringi_1.7.5    vroom_1.5.6      farver_2.1.0     fs_1.5.0
## [45] xml2_1.3.2       ellipsis_0.3.2   generics_0.1.1   vctrs_0.3.8
## [49] tools_4.1.1      bit64_4.0.5      glue_1.5.0       hms_1.1.1
## [53] parallel_4.1.1   fastmap_1.1.0    yaml_2.2.1       colorspace_2.0-2
## [57] rvest_1.0.2      knitr_1.36       haven_2.4.3
```