COVID_Analysis

2024-10-24

Introduction

In this analysis, I used Johns Hopkins github for analyzing covid 19 cases from 2020 to 2023. The goal is to determine deaths per million compared to cases, whats states were the safest vs which had the most deaths and if there is a linear regression between population size and deaths.

```
url <- "https://github.com/CSSEGISandData/COVID-19/raw/master/csse_covid_19_data/csse_covid_19_time_seri

files <- 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")

total_urls <- str_c(url,files)
total_urls
```

```
## [1] "https://github.com/CSSEGISandData/COVID-19/raw/master/csse_covid_19_data/csse_covid_19_time_seri
## [2] "https://github.com/CSSEGISandData/COVID-19/raw/master/csse_covid_19_data/csse_covid_19_time_seri
## [3] "https://github.com/CSSEGISandData/COVID-19/raw/master/csse_covid_19_data/csse_covid_19_time_seri
## [4] "https://github.com/CSSEGISandData/COVID-19/raw/master/csse_covid_19_data/csse_covid_19_time_seri
## [5] "https://github.com/CSSEGISandData/COVID-19/raw/master/csse_covid_19_data/csse_covid_19_time_seri
```

```
confirmed_us <- read_csv(total_urls[1])
```

```
## Rows: 3342 Columns: 1154
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1148): UID, code3, FIPS, Lat, Long_, 1/22/20, 1/23/20, 1/24/20, 1/25/20...
##
## 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.
```

```
confirmed_global <- read_csv(total_urls[2])
```

```
## Rows: 289 Columns: 1147
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1145): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## 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.
```

```
deaths_us <- read_csv(total_urls[3])
```

```
## Rows: 3342 Columns: 1155
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1149): UID, code3, FIPS, Lat, Long_, Population, 1/22/20, 1/23/20, 1/24...
##
## 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.
```

```
deaths_global <- read_csv(total_urls[4])
```

```
## Rows: 289 Columns: 1147
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1145): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## 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.
```

```
recovered_global <- read_csv(total_urls[5])
```

```
## Rows: 274 Columns: 1147
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1145): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20,...
##
## 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.
```

Preparing the data

Cleaning global cases and deaths

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

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

```
  select(-c(Lat,Long))

deaths_global
```

```
## # 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
## # i 330,317 more rows
```

```
totals_global <- confirmed_global %>%
  full_join(deaths_global) %>%
  rename(Country_Region = 'Country/Region',
         Province_State = 'Province/State') %>%
  mutate(date = mdy(date))
```

```
## Joining with `by = join_by(`Province/State`, `Country/Region`, date)`
```

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

US deaths and cases totals

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

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

US_Totals <- confirmed_us %>%
  full_join(deaths_us)
```

```
## Joining with `by = join_by(Admin2, Province_State, Country_Region,
## Combined_Key, date)`
```

```
US_Totals
```

```
## # 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
## # i 3,819,896 more rows
## # i 1 more variable: deaths <dbl>
```

Global totals

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

uid_lookup <- "https://github.com/CSSEGISandData/COVID-19/raw/master/csse_covid_19_data/UID_ISO_FIPS_Lo
uid_df <- read_csv(uid_lookup) %>%
  select(-c(Lat, Long_, Combined_Key, code3, iso2, iso3, Admin2))
```

```
## Rows: 4321 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.
```

```
totals_global <- totals_global %>%
  left_join(uid_df, by = c("Province_State", "Country_Region")) %>%
  select(-c(UID, FIPS)) %>%
  select(Province_State, Country_Region, date, cases, deaths, Population, Combined_Key)

totals_global
```

```
## # A tibble: 306,827 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
## # i 306,817 more rows
```

US and by state totals.

```
us_state <- US_Totals %>%
  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 `.groups` argument.
```

```
us_state
```

```
## # A tibble: 66,294 x 7
##    Province_State Country_Region date       cases deaths deaths_per_mill
##    <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
## # i 66,284 more rows
## # i 1 more variable: Population <dbl>
```

```
us <- us_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()
```

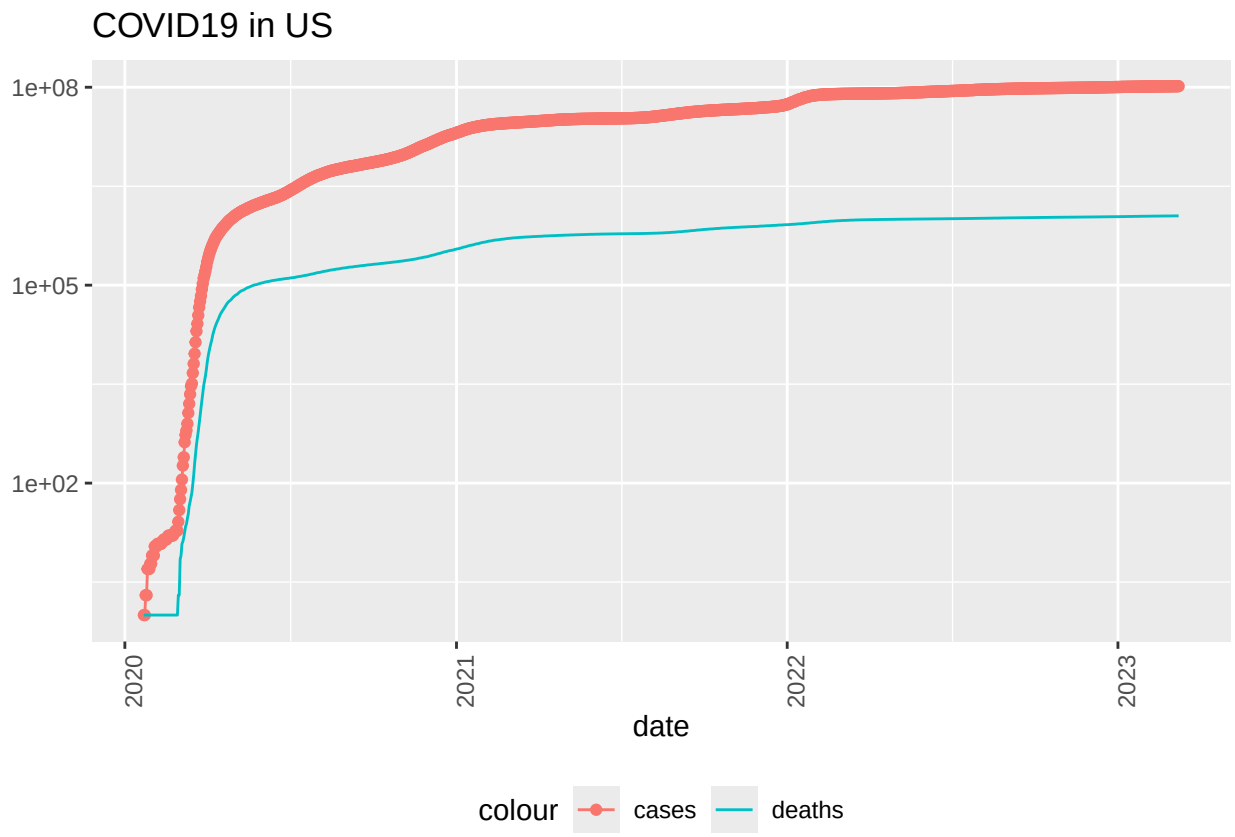
```
## `summarise()` has grouped output by 'Country_Region'. You can override using
## the `.groups` argument.
```

```
tail(us)
```

```
## # A tibble: 6 x 6
##   Country_Region date           cases  deaths deaths_per_mill 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
```

Cases compared to deaths in US over time

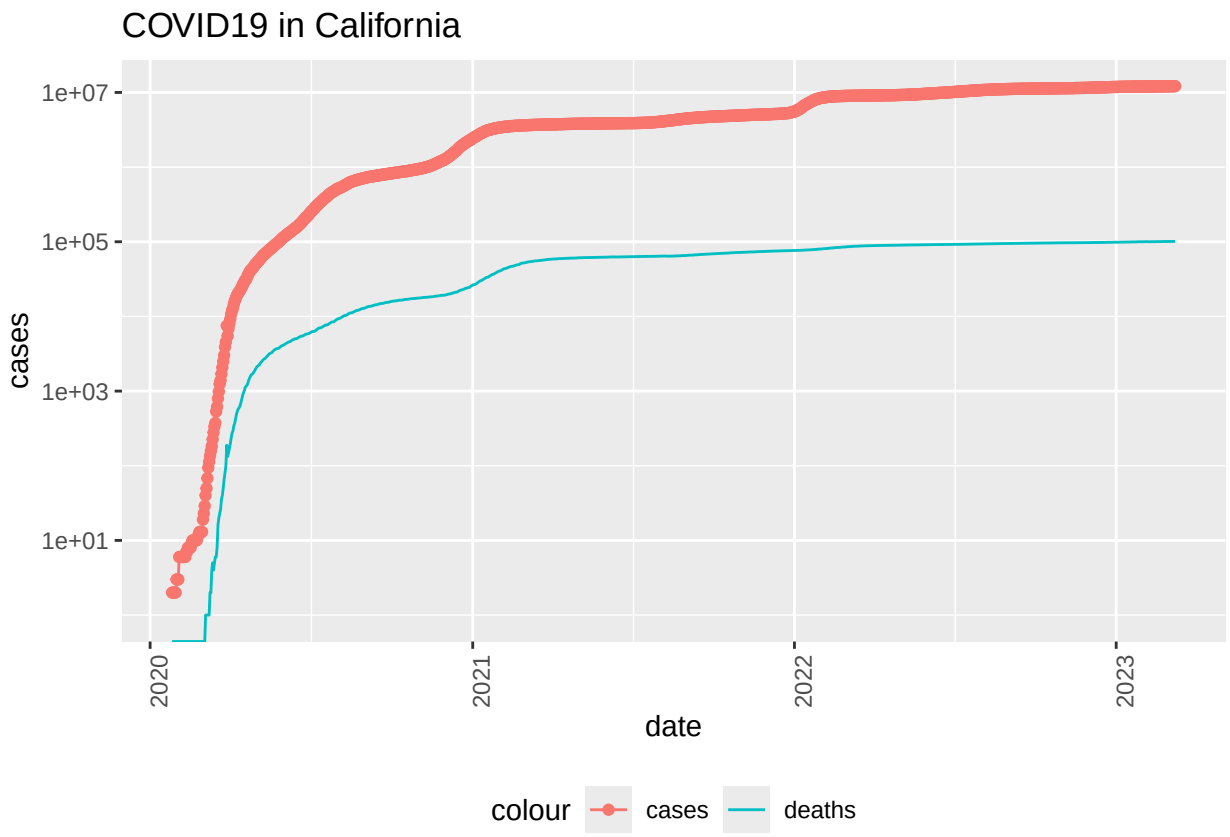
```
us_vis <- us %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = cases)) +
  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +
  geom_line(aes(y = deaths, color = "deaths")) +
  scale_y_log10() +
  theme(legend.position = "bottom",
        axis.text.x = element_text(angle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

us_vis
```



Cases compared to deaths in California over time.

```
state <- "California"
  us_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")) +
    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 scale_y_log10(): log-10 transformation introduced infinite values.
```



The plot below shows two trends: one for daily new cases and another for daily new deaths over time.

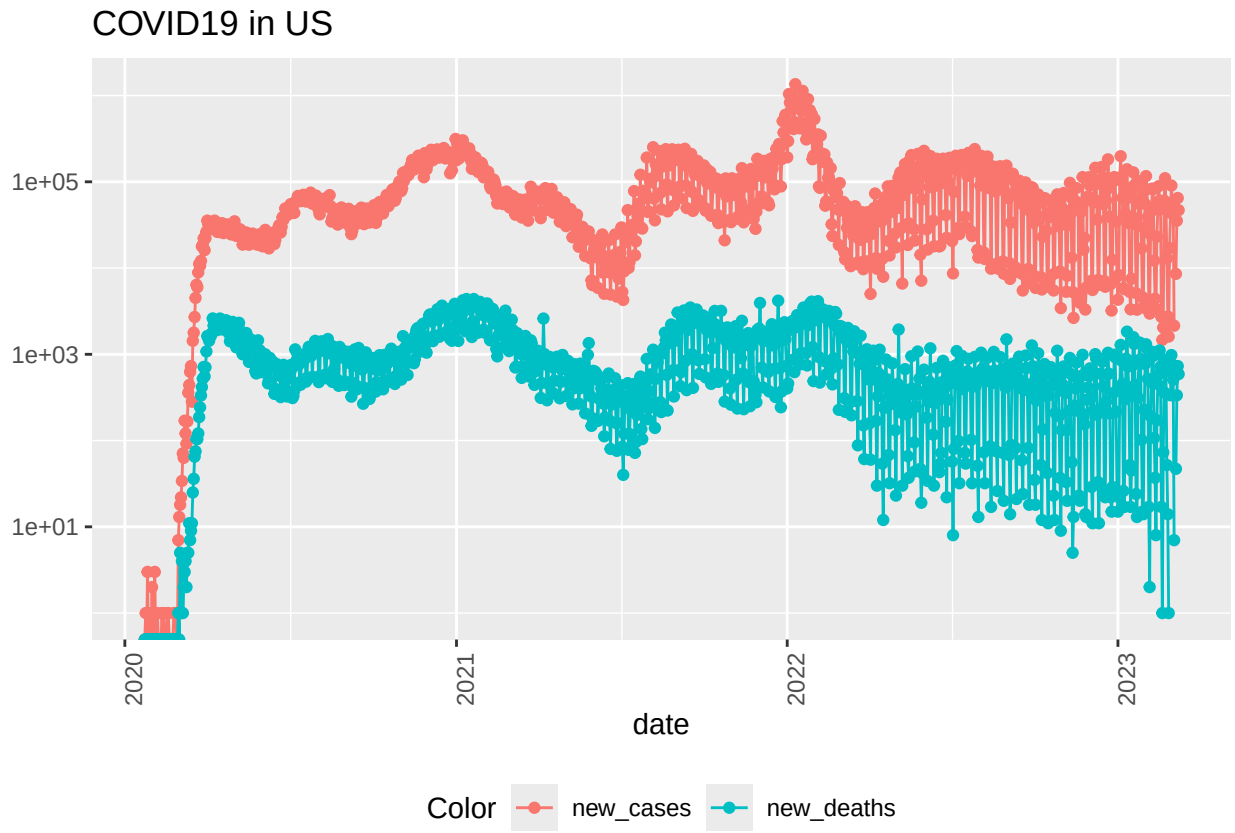
```
us_state <- us_state %>%
  mutate(new_cases = cases - lag(cases),
         new_deaths = deaths - lag(deaths))
us <- us %>%
  mutate(new_cases = cases - lag(cases),
         new_deaths = deaths - lag(deaths))

tail(us %>% select(new_cases, new_deaths, everything()))
```

```
## # A tibble: 6 x 8
##   new_cases new_deaths Country_Region date          cases deaths deaths_per_mill
##       <dbl>      <dbl> <chr>          <date>        <dbl>  <dbl>           <dbl>
## 1      2147          7 US             2023-03-04   1.04e8 1.12e6           3371.
## 2     -3862        -38 US             2023-03-05   1.04e8 1.12e6           3371.
## 3      8564         47 US             2023-03-06   1.04e8 1.12e6           3371.
## 4     35371        335 US             2023-03-07   1.04e8 1.12e6           3372.
## 5     64861        730 US             2023-03-08   1.04e8 1.12e6           3374.
## 6     46931        590 US             2023-03-09   1.04e8 1.12e6           3376.
## # i 1 more variable: Population <dbl>
```

```
us_new_vis <- us %>%
  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, color = "Color")

us_new_vis
```

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

## Warning in scale_y_log10(): log-10 transformation introduced infinite values.

## Warning in transformation$transform(x): NaNs produced

## Warning in scale_y_log10(): log-10 transformation introduced infinite values.

## Warning in transformation$transform(x): NaNs produced

## Warning in scale_y_log10(): log-10 transformation introduced infinite values.

## Warning in transformation$transform(x): NaNs produced

## Warning in scale_y_log10(): log-10 transformation introduced infinite values.

## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 2 rows containing missing values or values outside the scale range
## (`geom_point()`).

## Warning: Removed 1 row containing missing values or values outside the scale range
## (`geom_line()`).

## Warning: Removed 4 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
us_state_totals <- us_state %>%
  group_by(Province_State) %>%
  summarize(deaths = max(deaths), cases = max(cases),
            population = max(Population),
            cases_per_1k = (1000 * cases) / population,
            deaths_per_1k = (1000 * deaths) / population) %>%
  filter(cases > 0, population > 0)

top_states <- us_state_totals %>%
  slice_min(deaths_per_1k, n = 10) %>%
  select(deaths_per_1k, cases_per_1k, everything())

worst_states <- us_state_totals %>%
  slice_max(deaths_per_1k, n = 10) %>%
  select(deaths_per_1k, cases_per_1k, everything())

top_states
```

```
## # A tibble: 10 x 6
##    deaths_per_1k cases_per_1k Province_State           deaths   cases population
##            <dbl>        <dbl> <chr>                     <dbl>   <dbl>      <dbl>
##  1         0.611         150. American Samoa               34    8320      55641
##  2         0.744         248. Northern Mariana Islands     41   13666      55144
##  3         1.21          231. Virgin Islands              130   24813     107268
##  4         1.30          269. Hawaii                     1841  380608    1415872
```

```
##  5         1.49          245. Vermont                   929  152618     623989
##  6         1.55          293. Puerto Rico              5823 1101469    3754939
##  7         1.65          340. Utah                     5298 1090346    3205958
##  8         2.01          415. Alaska                   1486  307655     740995
##  9         2.03          252. District of Columbia     1432  177945     705749
## 10         2.06          253. Washington              15683 1928913    7614893
```

```
worst_states
```

```
## # A tibble: 10 x 6
##    deaths_per_1k cases_per_1k 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
```

Modeling the data

```
model <- lm(deaths_per_1k ~ cases_per_1k, data = us_state_totals)
summary(model)
```

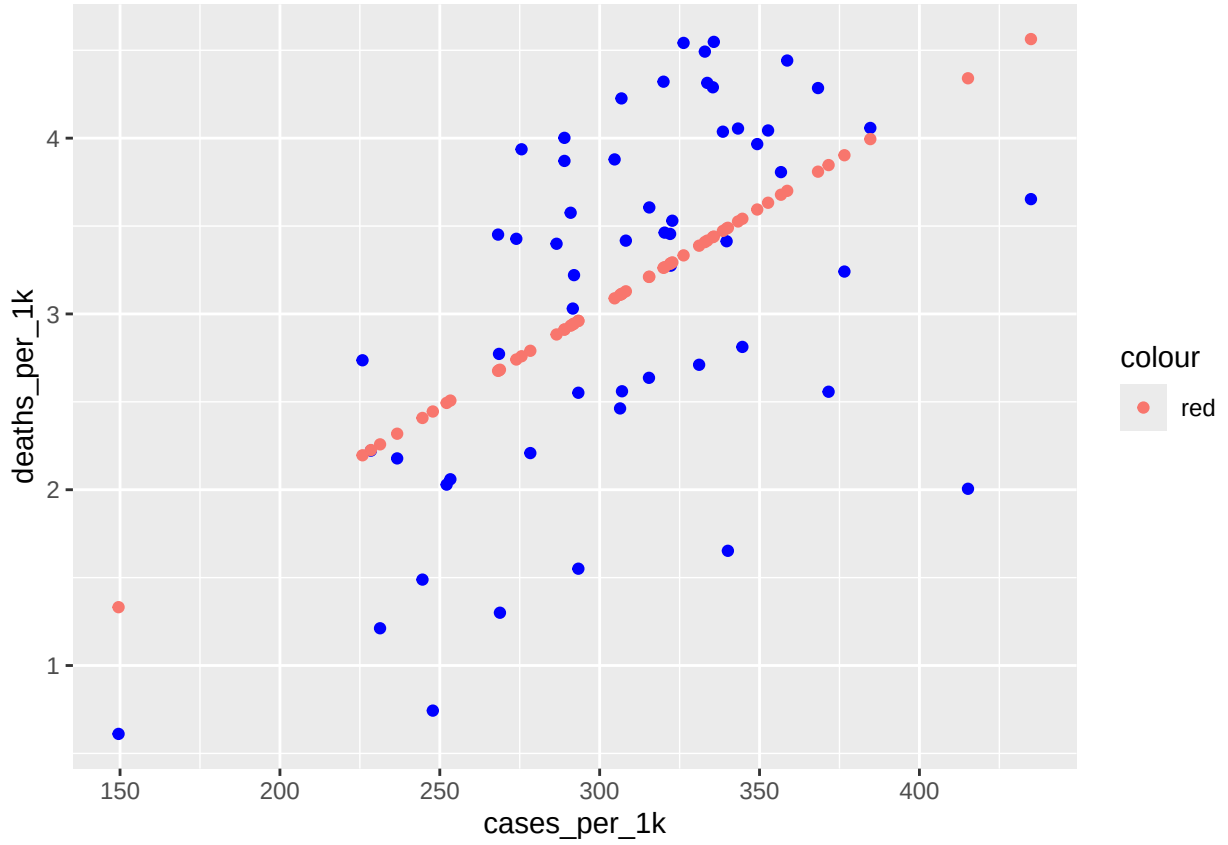
```
##
## Call:
## lm(formula = deaths_per_1k ~ cases_per_1k, data = us_state_totals)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3352 -0.5978  0.1491  0.6535  1.2086
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.36167    0.72480  -0.499     0.62
## cases_per_1k  0.01133    0.00232   4.881 9.76e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8615 on 54 degrees of freedom
## Multiple R-squared:  0.3061, Adjusted R-squared:  0.2933
## F-statistic: 23.82 on 1 and 54 DF,  p-value: 9.763e-06
```

```
model
```

```
##
## Call:
```

```
## lm(formula = deaths_per_1k ~ cases_per_1k, data = us_state_totals)
##
## Coefficients:
##  (Intercept)  cases_per_1k
##     -0.36167       0.01133
```

```
top_states <- us_state_totals %>%
  slice_min(deaths_per_1k)

worst_states <- us_state_totals %>%
  slice_max(deaths_per_1k)

us_w_predictions <- us_state_totals %>% mutate(prediction = predict(model))

model_vis <- us_w_predictions %>%
  ggplot() +
  geom_point(aes(x = cases_per_1k, y = deaths_per_1k), color = "blue") +
  geom_point(aes(x = cases_per_1k, y = prediction, color = "red"))

model_vis
```



“‘ ### Conclusion

Overall, the model shows that there is a statistically significant positive relationship between cases per 1000 and deaths per 1000, with an estimated increase in deaths per 1000 of 0.01133 for each additional case per 1000 people. However, the R-squared value suggests that there's still a large amount of variation in deaths per 1000 not explained by cases per 1000, adding more variables may help create a more accurate prediction.

Bias

A first possible source of bias is what third party variables could be throwing off the analysis. Are the cases vs deaths reported legitimately or could there be some error. Secondly, forgotten variables, which areas had more access to vaccines? what is the age difference in overall cases? and lastly, could the data be skewed due to data censoring? These are all possible limitations that could improve the validity when the proper solutions are applied.