Johns Hopkins Covid-19 Data Analysis for DTSA-5301

Daniel South

2025-03-04

Johns Hopkins Covid-19 Data Analysis

COVID-19 (SARS-CoV-2) caused millions of deaths as it spread around the globe. Johns Hopkins tracked confirmed cases and deaths in the United States and worldwide in an effort to better understand how it was being transmitted and where it would be likely to cause issues in the near future.

This report reviews time series tracking data from Johns Hopkins and plots the number of cases over time in various regions.

Part 1: Import, Clean Up, and Prepare Data Sets

Import Data Files

```
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)

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

global_cases <- read_csv(urls[1])
```

```
## 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.
```

```
global_deaths <- read_csv(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.
```

```
US_cases <- read_csv(urls[3])
```

```
## 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.
```

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

```
## 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.
```

Pivot, Drop, and Rename Columns

```
tidy_gbl_cases = global_cases %>%
  pivot_longer(cols = -c("Province/State", "Country/Region", "Lat", "Long"),
               names_to="the_date",
               values_to="num_cases") %>%
  select(-c("Lat", "Long"))

tidy_gbl_cases = tidy_gbl_cases %>%
  rename('Country_Region' = 'Country/Region', 'Province_State' = 'Province/State') %>%
  mutate(the_date = mdy(the_date))



tidy_gbl_deaths = global_deaths %>%
  pivot_longer(cols = -c("Province/State", "Country/Region", "Lat", "Long"),
               names_to="the_date",
               values_to="num_deaths") %>%
  select(-c("Lat", "Long"))

tidy_gbl_deaths = tidy_gbl_deaths %>%
  rename('Country_Region' = 'Country/Region', 'Province_State' = 'Province/State') %>%
  mutate(the_date = mdy(the_date))


tidy_us_cases = US_cases %>%
  pivot_longer(cols = -c("UID", "iso2", "iso3", "code3", "FIPS", "Admin2", "Combined_Key", "Province_Sta
               names_to="the_date",
               values_to="num_cases") %>%
  select(-c("Lat", "Long_", "iso2", "iso3", "code3", "FIPS"))

tidy_us_cases = tidy_us_cases %>%
  rename('County' = 'Admin2') %>%
  mutate(the_date = mdy(the_date))


tidy_us_deaths = US_deaths %>%
  pivot_longer(cols = -c("UID", "FIPS", "Combined_Key", "Admin2",  "iso2", "iso3", "code3",  "Province_S
               names_to="the_date",
               values_to="num_deaths") %>%
  select(-c("Lat", "Long_", "iso2", "iso3", "code3", "FIPS"))

tidy_us_deaths = tidy_us_deaths %>%
  rename('County' = 'Admin2') %>%
  mutate(the_date = mdy(the_date))
```

Joins and Transformations

```
global_data = tidy_gbl_cases %>%
  full_join(tidy_gbl_deaths)
```

```
## Joining with `by = join_by(Province_State, Country_Region, the_date)`
```

```
us_data = tidy_us_cases %>%
  full_join(tidy_us_deaths)
```

```
## Joining with `by = join_by(UID, County, Province_State, Country_Region,
## Combined_Key, the_date)`
```

```
global_data = global_data %>% filter(num_cases > 0)

us_data = us_data %>% filter(num_cases > 0)
```

```
# Add Combined_Key and Population to the Global Data Set

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

uid = read_csv(uid_lookup_url) %>%
  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.
```

```
global_data2 = global_data %>%
  left_join(uid, by = c("Province_State", "Country_Region")) %>%
  select(-c(UID, FIPS)) %>%
  select(Province_State, Country_Region, the_date, num_cases, num_deaths, Population, Combined_Key)
```

Summaries of Prepared Data Sets

```
summary(us_data)
```

```
##       UID              County          Province_State     Country_Region
##  Min.   :      16   Length:3474292     Length:3474292     Length:3474292
##  1st Qu.:84018061   Class :character   Class :character   Class :character
##  Median :84029113   Mode  :character   Mode  :character   Mode  :character
##  Mean   :83434765
##  3rd Qu.:84045075
##  Max.   :84099999
##  Combined_Key          the_date            num_cases         Population
##  Length:3474292     Min.   :2020-01-22   Min.   :      1   Min.   :       0
##  Class :character   1st Qu.:2020-12-27   1st Qu.:    687   1st Qu.:   10953
##  Mode  :character   Median :2021-09-20   Median :   2849   Median :   26248
##                     Mean   :2021-09-19   Mean   :  15489   Mean   :  104502
##                     3rd Qu.:2022-06-15   3rd Qu.:   9345   3rd Qu.:   68098
##                     Max.   :2023-03-09   Max.   :3710586   Max.   :10039107
##    num_deaths
##  Min.   :    0.0
##  1st Qu.:   10.0
##  Median :   47.0
##  Mean   :  205.1
##  3rd Qu.:  137.0
##  Max.   :35545.0
```

```
summary(global_data2)
```

```
##  Province_State     Country_Region        the_date            num_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
##
##    num_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
```

Part 2 - Data Visualization

Summarize US Data

```
# Summarize US Data

US_by_state = us_data %>%
  group_by(Province_State, Country_Region, County, the_date) %>%
  summarize(num_cases = sum(num_cases), num_deaths = sum(num_deaths), Population = max(Population)) %>%
  mutate(deaths_per_mill = num_deaths * 1000000/Population) %>%
  select(Province_State, Country_Region, the_date, num_cases, num_deaths, deaths_per_mill, Population, C
  ungroup()
```

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

```
US_totals = US_by_state %>%
  group_by(Country_Region, the_date) %>%
  summarize(num_cases = sum(num_cases), num_deaths = sum(num_deaths), Population = sum(Population)) %>%
  mutate(deaths_per_mill = num_deaths * 1000000/Population) %>%
  select(Country_Region, the_date, num_cases, num_deaths, deaths_per_mill, Population) %>%
  ungroup()
```

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

Derive Columns for New Cases and Deaths

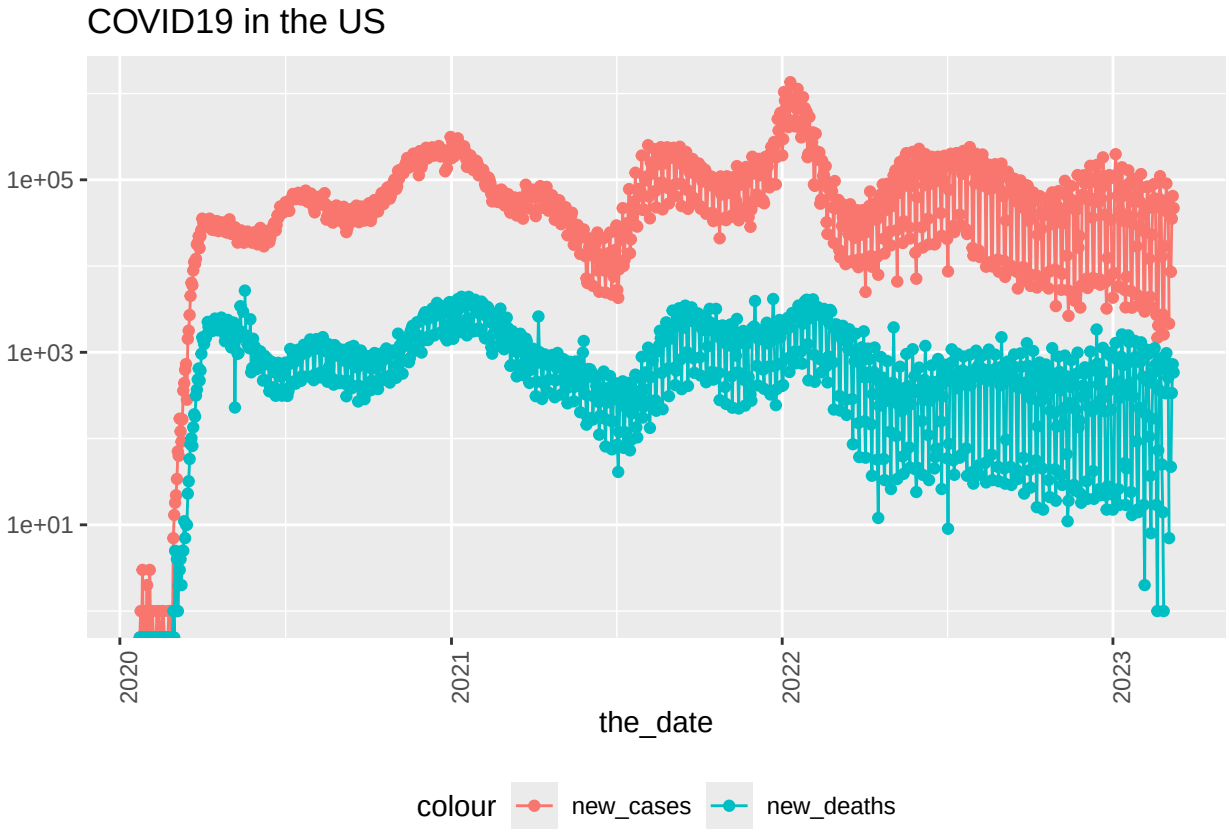
```
US_by_state = US_by_state %>%
  mutate(new_cases = num_cases - lag(num_cases),
         new_deaths = num_deaths - lag(num_deaths))

US_totals = US_totals %>%
  mutate(new_cases = num_cases - lag(num_cases),
         new_deaths = num_deaths - lag(num_deaths))

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

```
## # A tibble: 6 x 8
##   new_cases new_deaths Country_Region the_date   num_cases num_deaths
##       <dbl>      <dbl> <chr>          <date>         <dbl>      <dbl>
## 1      2147          7 US             2023-03-04 103650837    1121067
## 2     -3862        -38 US             2023-03-05 103646975    1121029
## 3      8564         47 US             2023-03-06 103655539    1121076
## 4     35371        337 US             2023-03-07 103690910    1121413
## 5     64861        727 US             2023-03-08 103755771    1122140
## 6     46931        584 US             2023-03-09 103802702    1122724
## # i 2 more variables: deaths_per_mill <dbl>, Population <dbl>
```

Visualize US Data

```
US_totals %>%
  ggplot(aes(x = the_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 the US", y = NULL)
```

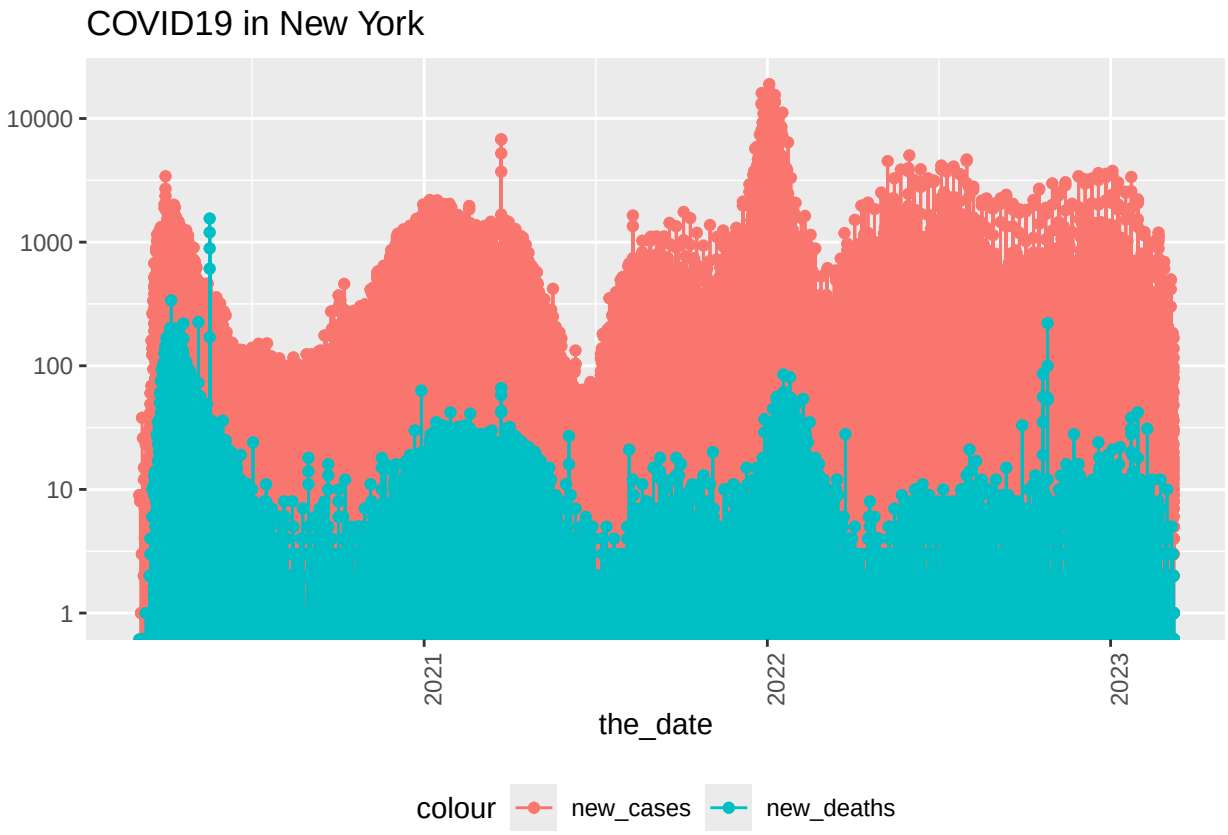
```
## Warning in transformation$transform(x): NaNs produced

## 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 7 rows containing missing values or values outside the scale range
## (`geom_point()`).
```


COVID19 in the US
1e+05
1e+03
1e+01
2020
2021
2022
2023
the_date
colour
new_cases
new_deaths

Visualize NY State Data

```
state_name = "New York"

US_by_state %>%
  filter(Province_State == state_name) %>%
  filter(num_cases > 0) %>%
  ggplot(aes(x = the_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_name), y = NULL)
```

```
## 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 297 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 377 rows containing missing values or values outside the scale range
## (`geom_point()`).
```


COVID19 in New York
10000
1000
100
10
1
2021
2022
2023
the_date
colour
new_cases
new_deaths

Part 3 - Data Analysis

```
US_state_totals = US_by_state %>%
  group_by(Province_State) %>%
  summarize(num_deaths = max(num_deaths), num_cases = max(num_cases),
            population=max(Population),
            cases_per_thou = 1000 * num_cases / population,
            deaths_per_thou = 1000 * num_deaths / population
            ) %>%
  filter(num_cases > 0, population > 0)
```

US States with the Highest Death Rates

```
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 num_deaths num_cases population
##              <dbl>          <dbl> <chr>               <dbl>     <dbl>      <dbl>
##  1            14.7           257. Puerto Rico          5823    101521     395326
##  2             9.51          571. Florida             25840   1552197    2716940
##  3             5.55          376. New York            14219    963672    2559903
##  4             5.21          307. Michigan             9107    537017    1749343
##  5             4.95          333. West Virginia         881     59239     178124
##  6             4.26          436. Rhode Island         2724    278748     638931
##  7             4.20          341. Arizona             18846   1530296    4485414
##  8             4.11          396. South Carolina       2154    207151     523542
##  9             4.11          296. Nevada               9313    671243    2266715
## 10             4.05          350. New Jersey           3771    326525     932202
```

US States with the Lowest Death Rates

```
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 num_deaths num_cases population
##              <dbl>          <dbl> <chr>               <dbl>     <dbl>      <dbl>
##  1           0.611           150. American Samoa         34      8320      55641
##  2           0.744           248. Northern Mari~         41     13666      55144
##  3           1.21            231. Virgin Islands        130     24813     107268
##  4           1.39            271. Hawaii               1355    264197     974563
##  5           1.40            242. Vermont               230     39572     163774
##  6           1.50            228. Virginia             1720    261545    1147532
##  7           1.56            244. Washington           3512    549865    2252782
##  8           1.58            354. Utah                 1830    410508    1160437
##  9           1.69            225. Maine                 498     66462     295003
## 10           1.70            352. North Carolina       1892    391609    1111761
```

Part 4 - Data Modeling

If we fit the US state totals to a linear model, we can use visualization to determine whether the model can predict cases accurately.

Summary of the Linear Model applied to US State Totals

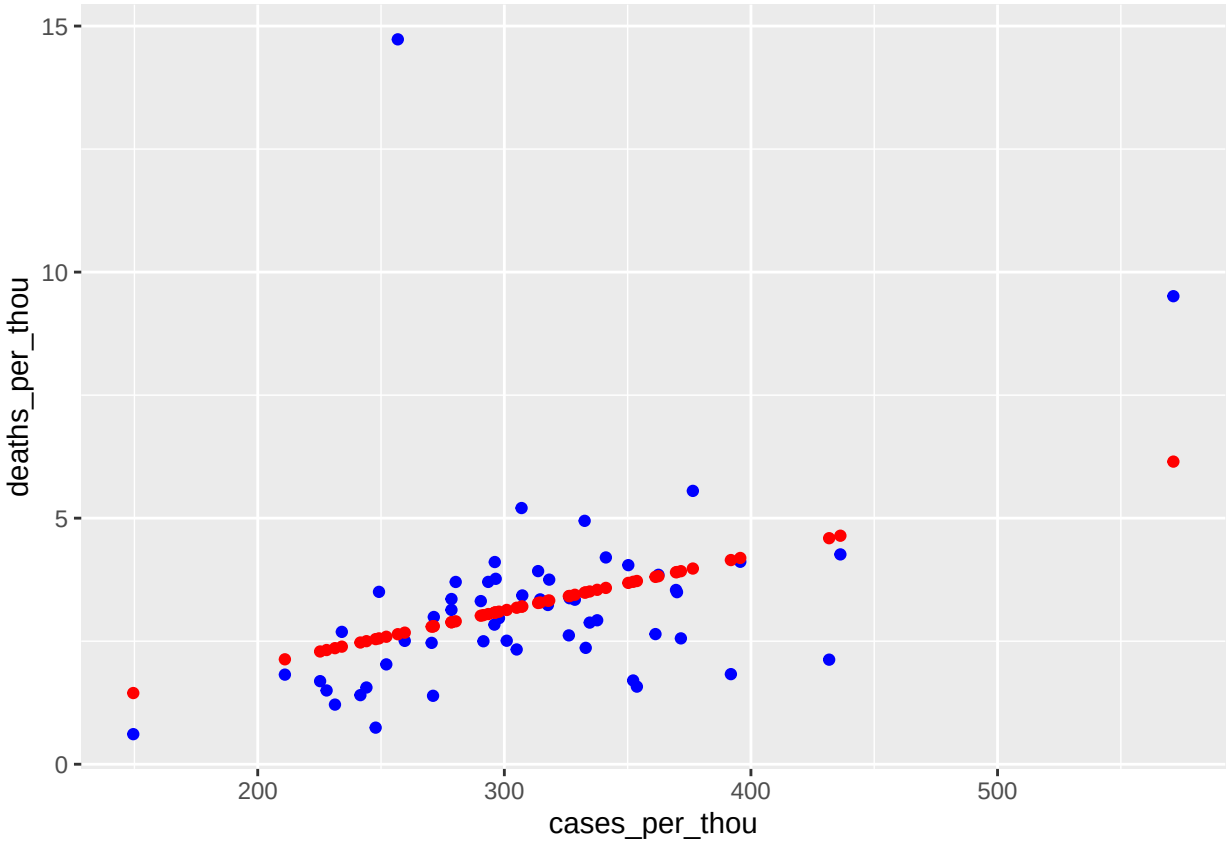
```
lin_mod = lm(deaths_per_thou ~ cases_per_thou, data = US_state_totals)
summary(lin_mod)
```

```
##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou, data = US_state_totals)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4679 -0.8250 -0.2047  0.3763 12.0871
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.221073   1.278002  -0.173  0.86331
## cases_per_thou  0.011151   0.004029   2.768  0.00772 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.972 on 54 degrees of freedom
## Multiple R-squared:  0.1242, Adjusted R-squared:  0.108
## F-statistic:  7.66 on 1 and 54 DF,  p-value: 0.007717
```

```
predict(lin_mod)
```

```
##        1        2        3        4        5        6        7        8
## 3.820980 4.592879 1.446300 3.583249 3.419253 3.900394 3.415979 2.904284
##        9       10       11       12       13       14       15       16
## 3.510175 2.590439 6.149392 2.673883 3.922515 2.801817 3.492234 3.100046
##       17       18       19       20       21       22       23       24
## 2.885106 3.134976 3.079279 3.904822 3.442955 2.291111 2.389319 2.805353
##       25       26       27       28       29       30       31       32
## 3.202010 3.179978 3.051288 3.085208 3.327052 3.286224 3.081004 3.029646
##       33       34       35       36       37       38       39       40
## 3.684734 3.017752 3.976614 3.706693 4.148625 2.542351 2.884959 3.205381
##       41       42       43       44       45       46       47       48
## 2.131923 2.557423 2.642478 4.643695 4.190971 3.543998 3.276849 2.795381
##       49       50       51       52       53       54       55       56
## 3.723540 2.473236 2.358296 2.320403 2.500634 3.487353 3.807557 3.321170
```

```
state_predicitons = US_state_totals %>% mutate(pred = predict(lin_mod))

state_predicitons %>% 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")
```


15
10
5
0
deaths_per_thou
200
300
400
500
cases_per_thou

Part 5 - Conclusions

Potential Biases

Efforts to mitigate the spread of Covid-19 became controversial. Businesses closures led to financial hardships. Supply shortages led to difficult conditions in hospitals. Lockdowns led to feelings of isolation. Infection rates rose and fell in unpredictable patterns. People have strong opinions to this day regarding public health measures that were deployed during the pandemic.

It's conceivable that cases were under-reported in some countries and states. Politicians in some areas encouraged suppression of infection counts to encourage a rapid end to mitigation efforts.

Predictions

The linear model would not have served as an effective predictor of Covid-19 cases. Additional data points would be needed in order to create a useful model.