Covid-19 Assignment

Cameron S.

2025-07-28

Libraries

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.2     v tibble    3.3.0
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.1.0
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(lubridate)
```

For the COVID-19 data, I am interested in better understanding this data. I would like to know more about the relationships between the cases and deaths between states in the US and if there is a way to predict the cases/deaths of a state based on it's population. Can we manipulate and clean this data and then analyze it to better understand how COVID affected the United States and the world? That is the key question for this analysis and assignment.

Lecture Code

COVID-19 Data

Here is the covid-19 data. The data includes information about global cases, deaths, and countries. The next data includes information about the US, states, cases, and deaths. The last dataset includes information about populations for global countries.

```
##################
# Global data load
global_confirm <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_
```

```
## 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_death <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19
```

```
## 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 data load
us_confirm <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_
```

```
## 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_death <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_da
```

```
## 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.
```

```
############################
# Global population data load
global_pop <- read_csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_
```

```
## 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.
```

For the class videos, we used a github repository to load in the data for the cases and deaths for global and US data. We also grabbed some population data in order to further our analysis.

Data Cleaning and Manipulation

```
# Pivot table: shift each row to be a date for the given country/region with a new "cases" column for t

# Global confirmed pivot
global_confirm_pivot <- global_confirm %>% pivot_longer(cols = -c('Province/State', 'Country/Region', '
                                                        names_to='date',
                                                        values_to='cases') %>%
                                       select(-c(Lat,Long))

# Global death pivot
global_death_pivot <- global_death %>% pivot_longer(cols = -c('Province/State', 'Country/Region', 'Lat'
                                                        names_to='date',
                                                        values_to='deaths') %>%
                                       select(-c(Lat,Long))

# US Confirmed pivot
us_confirm_pivot <- us_confirm %>% pivot_longer(cols = -c(UID:Combined_Key),
                                                names_to='date',
                                                values_to='cases') %>%
                                   select(Admin2:cases) %>%
                                   mutate(date = mdy(date)) %>%
                                   select(-c(Lat, Long_))

# US Deaths pivot
us_death_pivot <- us_death %>% pivot_longer(cols = -c(UID:Population),
                                                names_to='date',
                                                values_to='deaths') %>%
                                   select(Admin2:deaths) %>%
                                   mutate(date = mdy(date)) %>%
                                   select(-c(Lat, Long_))




# Create new "Global" variable to combine the global confirmed cases with global deaths
global <- global_confirm_pivot %>% full_join(global_death_pivot) %>%
                               rename(Country_Region = 'Country/Region',
                                      Province_State = 'Province/State') %>%
                               mutate(date = mdy(date))
```

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

```
# Create new column for Global data combining the province/state value with the country/region value
global <- global %>% unite("Combined_Key", c(Province_State, Country_Region), sep = ", ", na.rm = TRUE,

# Combine Global COVID data with the Global population data
global <- global %>% left_join(global_pop, by = c("Province_State", "Country_Region")) %>%
                     select(-c(UID, FIPS)) %>%
                     select(Province_State, Country_Region, date, cases, deaths, Population, Combined_K

# Filter out rows with 0 cases
global <- global %>% filter(cases > 0)




# Create new "US" variable to combine the US confirmed cases with the US deaths
US <- us_confirm_pivot %>% full_join(us_death_pivot)
```

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

```
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 `.groups` argument.
```

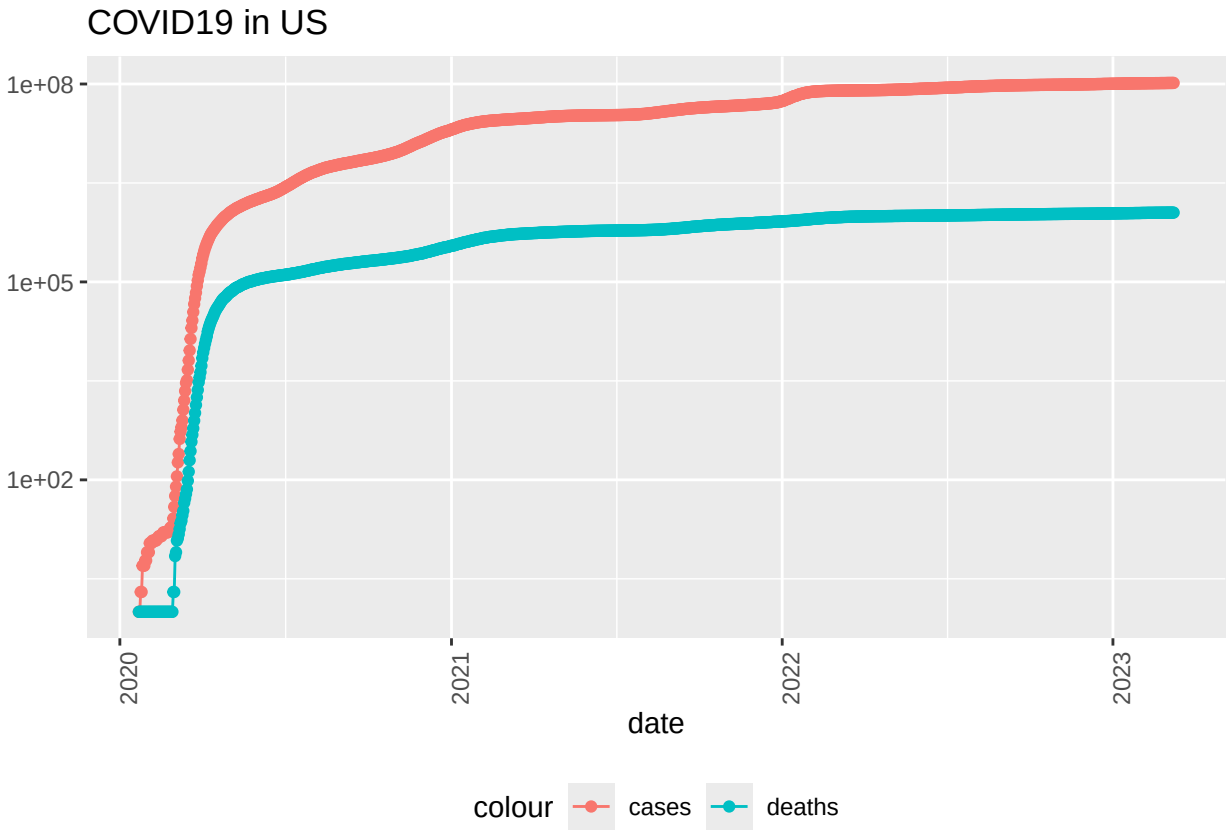
```
US_totals <- US_by_state %>%
             group_by(Country_Region, date) %>%
             summarize(cases = sum(cases), deaths = sum(deaths),
                       Population = sum(Population)) %>%
             mutate(deaths_per_mill = deaths*1000000 / Population) %>%
             select(Country_Region,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

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

The data cleaning we performed in class was to remove variables/columns that we didn't need, pivot and manipulate the US and global data to read and use the data easier, and create new variables and statistics. We also added in populations to analyze population related to cases and deaths per thousand and million citizens. We also filtered the data for non zero case values because we cannot have a negative number of cases.

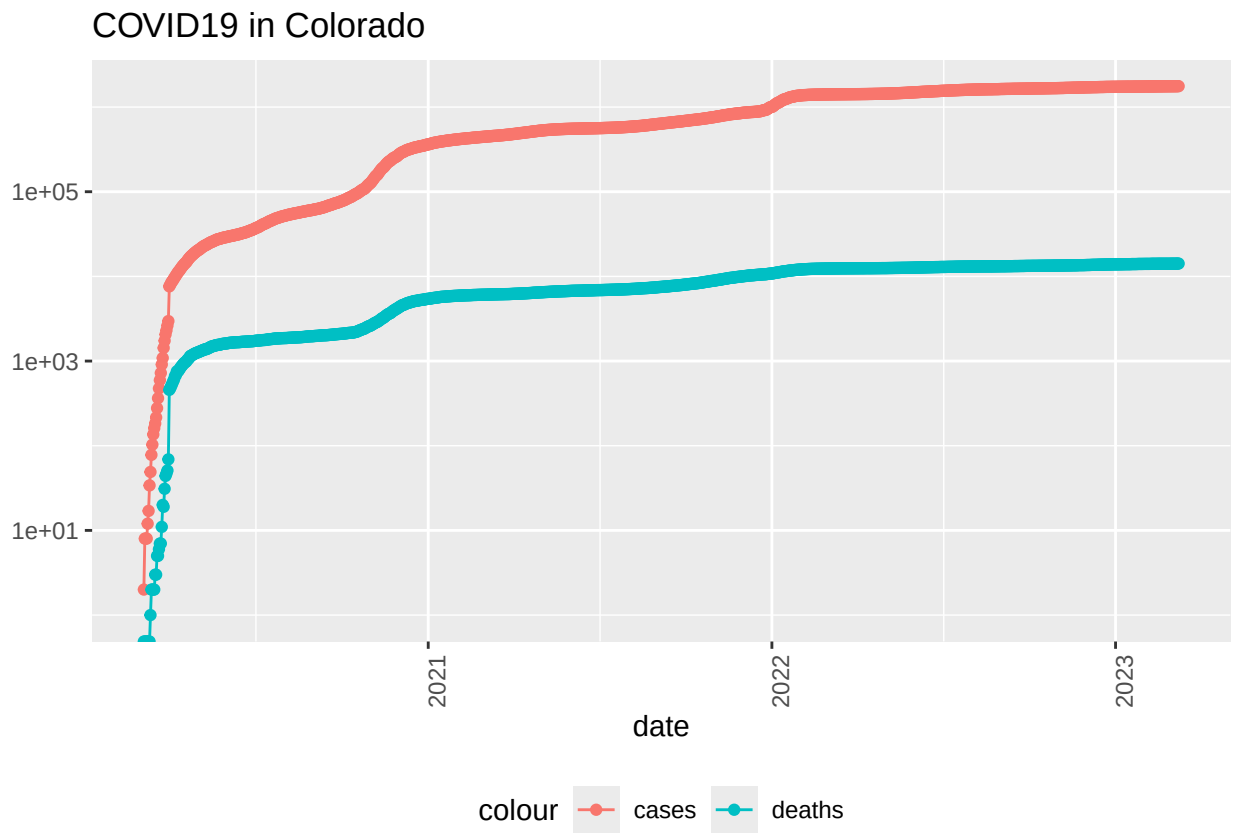
Data Visualizations

```
# US cases and deaths plot
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)
```




```
# Colorado cases and deaths
state <- "Colorado"
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 in scale_y_log10(): log-10 transformation introduced infinite values.
## log-10 transformation introduced infinite values.
```



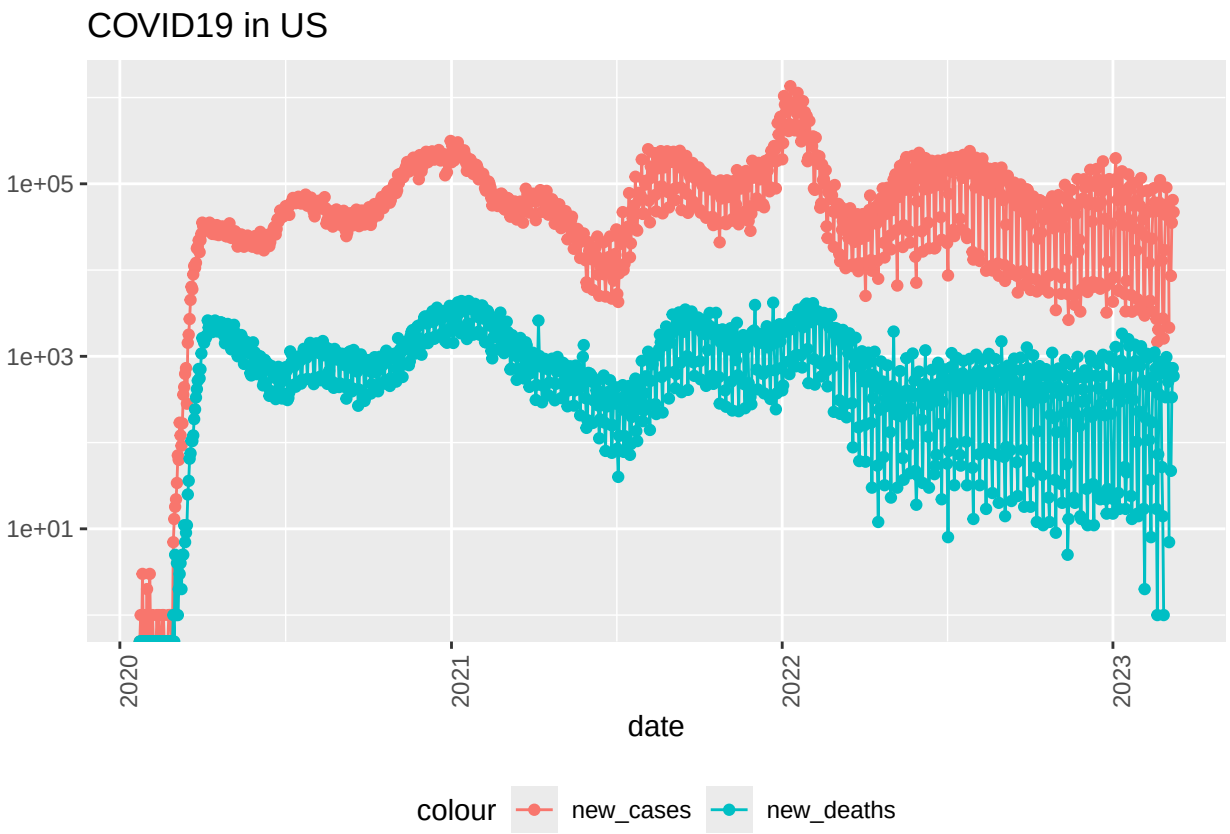
The visualizations we created for class were for showing the cases and deaths over time for the US and Colorado. I chose Colorado, and the pattern for both the US and Colorado were the same. There was a sharp increase early in 2020 and the steady drop off. Both visualizations showed spikes in cases/deaths in late 2020 and late 2021. This could be because of the winter season and people generally getting sick more in that season.

Data Analysis

```
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))

# US cases and deaths plot
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)
```

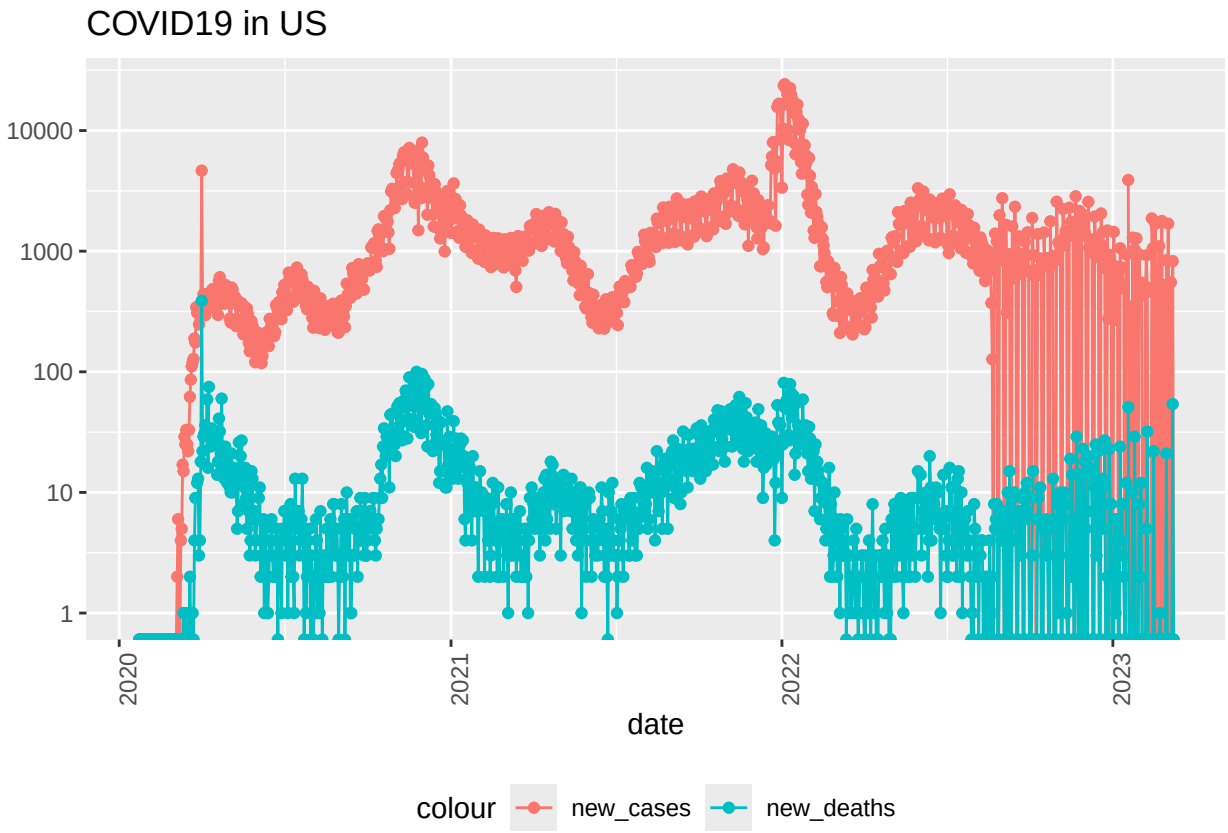
```
## Warning in transformation$transform(x): NaNs produced
```

```
## 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()`).
```



```
#
state <- "Colorado"
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 = "COVID19 in 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 5 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
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 / p
                   filter(cases > 0, population > 0)

# 10 smallest death count states
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 American Samoa            34 8.32e3      55641           150.           0.611
##  2 Northern Mariana Isl~     41 1.37e4      55144           248.           0.744
##  3 Virgin Islands           130 2.48e4     107268           231.           1.21
##  4 Hawaii                  1841 3.81e5    1415872           269.           1.30
##  5 Vermont                  929 1.53e5     623989           245.           1.49
##  6 Puerto Rico             5823 1.10e6    3754939           293.           1.55
##  7 Utah                    5298 1.09e6    3205958           340.           1.65
##  8 Alaska                  1486 3.08e5     740995           415.           2.01
##  9 District of Columbia    1432 1.78e5     705749           252.           2.03
## 10 Washington             15683 1.93e6    7614893           253.           2.06
```

```
# 10 largest death count states
US_state_totals %>% slice_max(deaths_per_thou, n = 10)
```

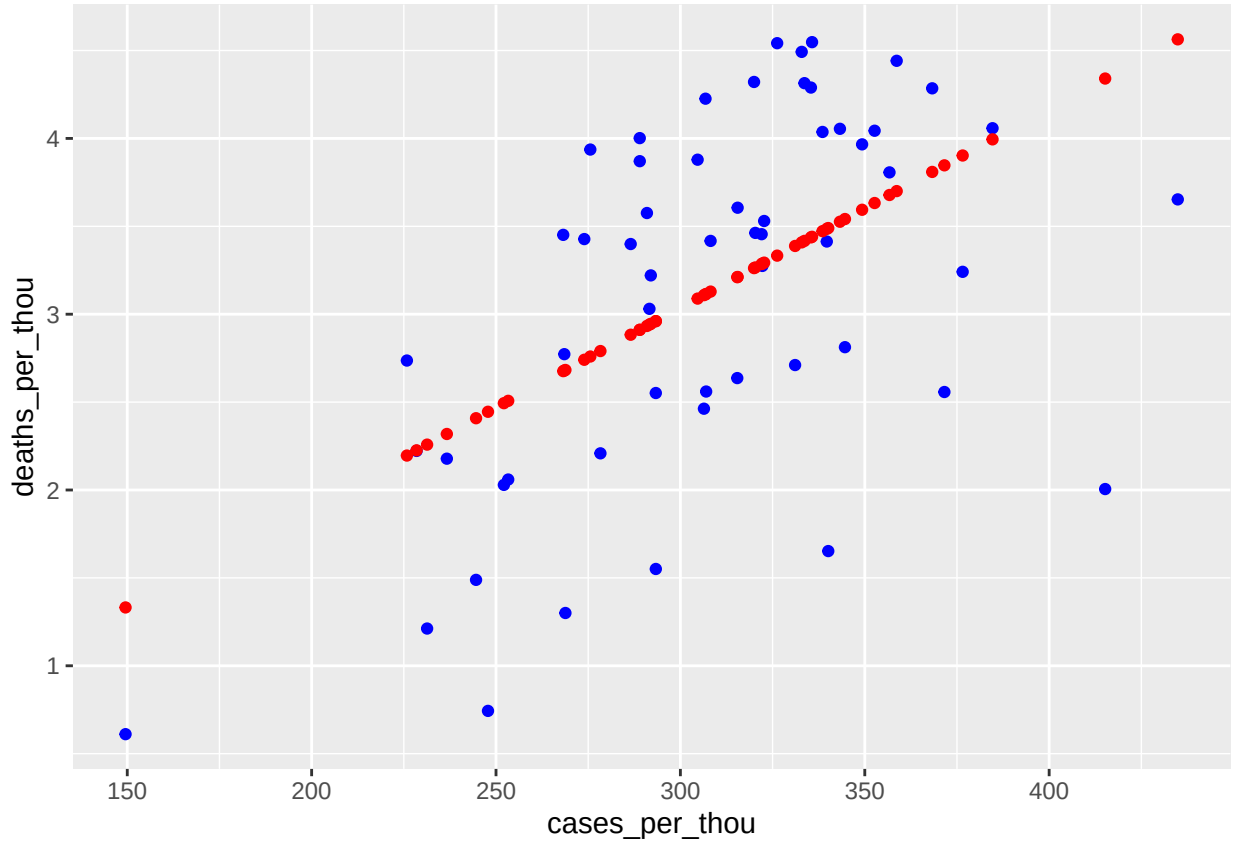
```
## # A tibble: 10 x 6
##    Province_State deaths   cases population cases_per_thou deaths_per_thou
##    <chr>           <dbl>   <dbl>      <dbl>          <dbl>           <dbl>
##  1 Arizona         33102 2443514    7278717           336.            4.55
##  2 Oklahoma        17972 1290929    3956971           326.            4.54
##  3 Mississippi     13370  990756    2976149           333.            4.49
##  4 West Virginia    7960  642760    1792147           359.            4.44
##  5 New Mexico       9061  670929    2096829           320.            4.32
##  6 Arkansas        13020 1006883    3017804           334.            4.31
##  7 Alabama         21032 1644533    4903185           335.            4.29
##  8 Tennessee       29263 2515130    6829174           368.            4.28
##  9 Michigan        42205 3064125    9986857           307.            4.23
## 10 Kentucky        18130 1718471    4467673           385.            4.06
```

For the data analysis done in class, we focused on visualizing the US data. For this, we looked into the total cases and deaths per state as well as the cases and deaths per thousand. The data showed a pretty consistent pattern of low population matched with low cases/deaths and vice verse for high population.

The lag told us that cases eventually subsided over time and that lead to a decrease in deaths as well. This was something we could not see with the totals. I also visualized the lag for Colorado as that is where I am from, and a similar pattern arose, but it seemed that cases dropped off more than average for the US.

Data Model

```
# Deaths per thousand as represented by cases per thousand
mod <- lm(deaths_per_thou ~ cases_per_thou, data = US_state_totals)

# Predict on US state totals
US_tot_w_pred <- US_state_totals %>% mutate(pred = predict(mod))

# Plot predicted values compared to actual values
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")
```



This is the final model from the class assignment. For this model, we attempted to predict deaths per thousand in the population as a function of cases per thousand. The model shows a general upward trend of the number of deaths with the an increase in the number of cases per state. This intuitively makes sense as when there are more cases, there should be more deaths. There are some interesting points though that show a low number of deaths with a large number of cases. This could be due to certain states taking quartantine and other social restrictions more seriously during lockdown. Another possible reason could be that the states have better healthcare than other states and are able to provide better services to the positive cases and prevent more deaths. More analysis of the specific states could be done in order to prevent personal bias of states with significantly higher deaths to cases ratio.

Personal Analysis

Population Over Time

Above has been the lecture code, but I want to look more at population and it's potential relationship with the number of cases and deaths.

I will first look at just one state and the population change over time. I will also refer back the the US_totals dataframe the compare the state to the whole of the US.

```
# State population dataframe. I chose Colorado because that is where I am from.
state_pop <- US_by_state %>% select(Province_State, date, cases, Population, deaths) %>% filter(Provinc

# Summary stats of populations
summary(state_pop$Population)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 5758736 5758736 5758736 5758736 5758736 5758736
```

```
summary(US_totals$Population)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 332875137 332875137 332875137 332875137 332875137 332875137
```

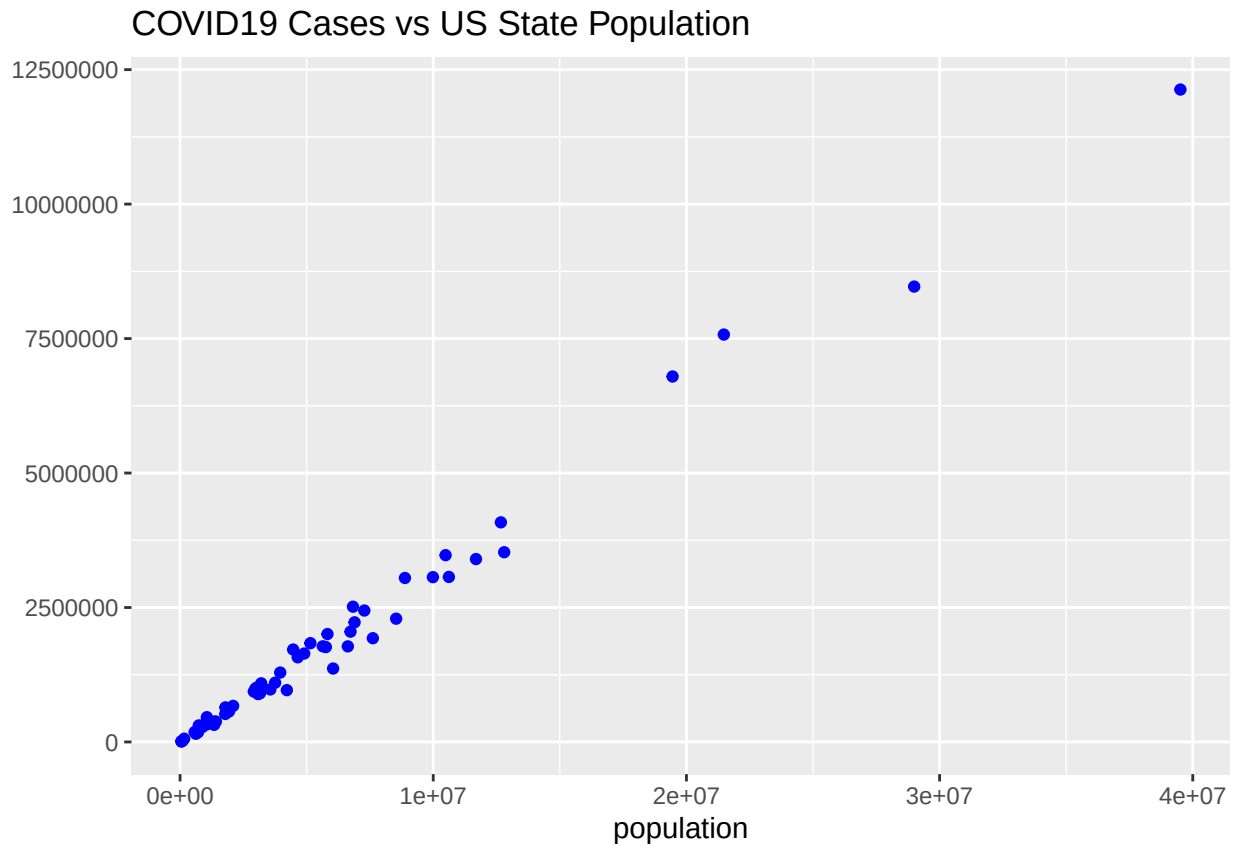
Although I want to continue this analysis, the data appears to not be supported for this. The population data for Colorado and the whole of the US does not change over the course of the 3-4 years of the data. For this reason, I will need to shift my analysis.

Instead, I will look at US_state_totals dataframe in order to find some relationship between the states population and the deaths and cases.

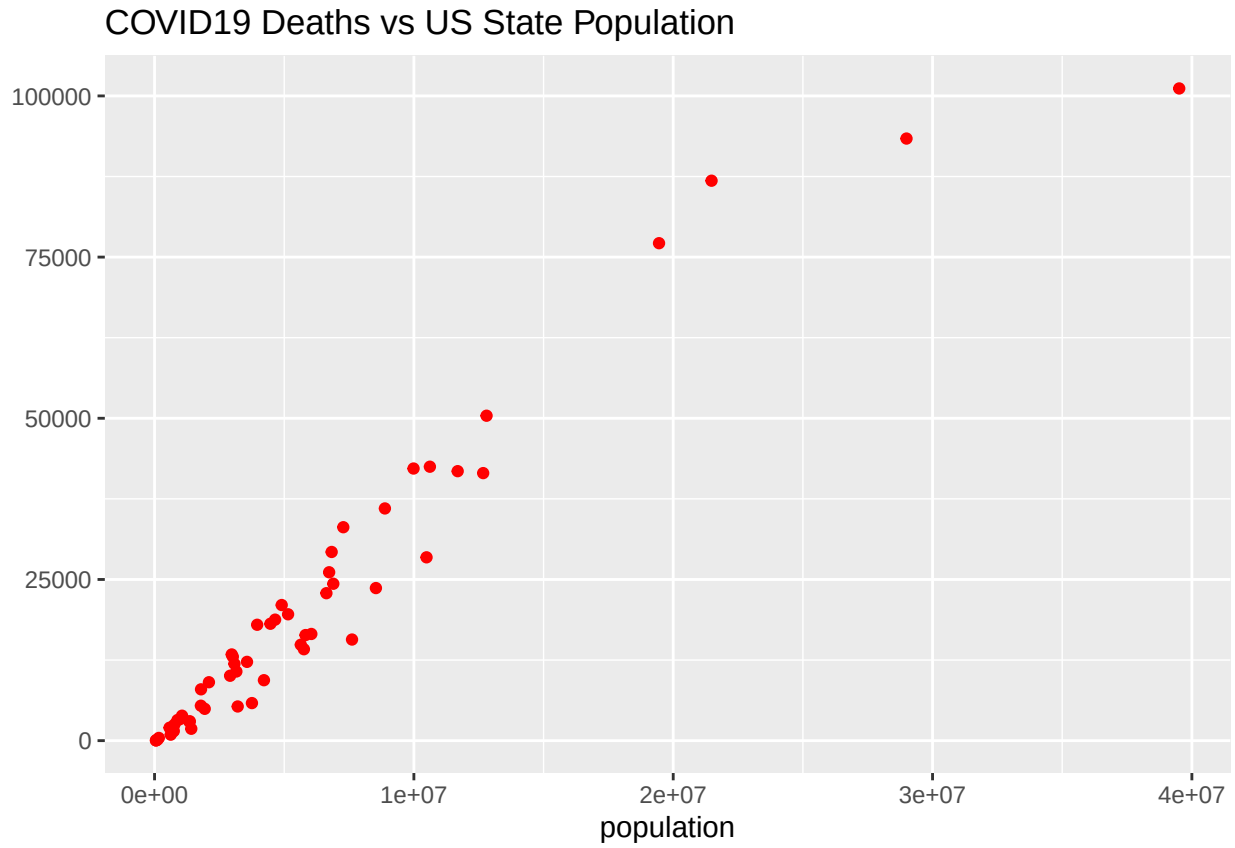
Cases and Deaths relationship with Population

First things first, I will visualize the scatter plot of population on the x-axis and the number of cases on the y-axis. Then the same with deaths on the y-axis.

```
# Cases and pop
US_state_totals %>% ggplot() + geom_point(aes(x = population, y = cases), color="blue") + labs(title =
```



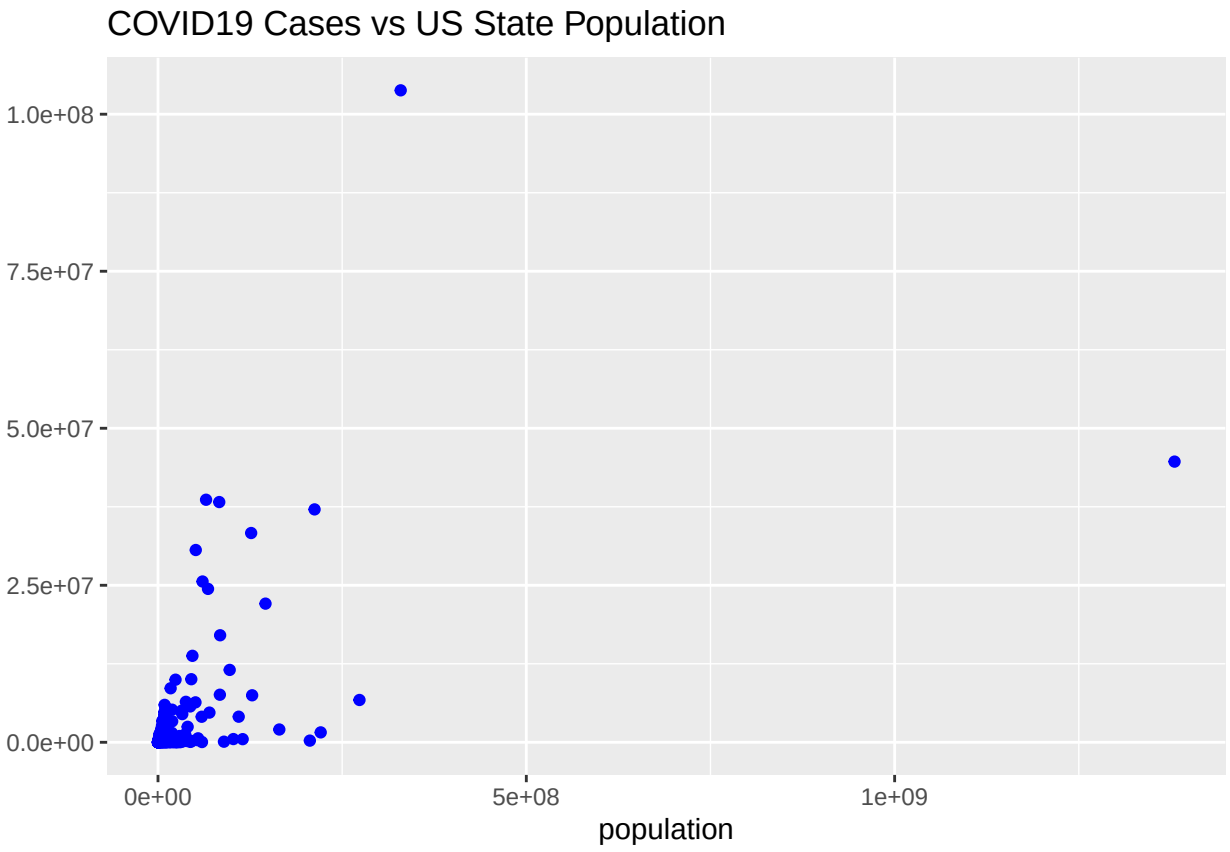
```
# Deaths and pop
US_state_totals %>% ggplot() + geom_point(aes(x = population, y = deaths), color="red") + labs(title =
```



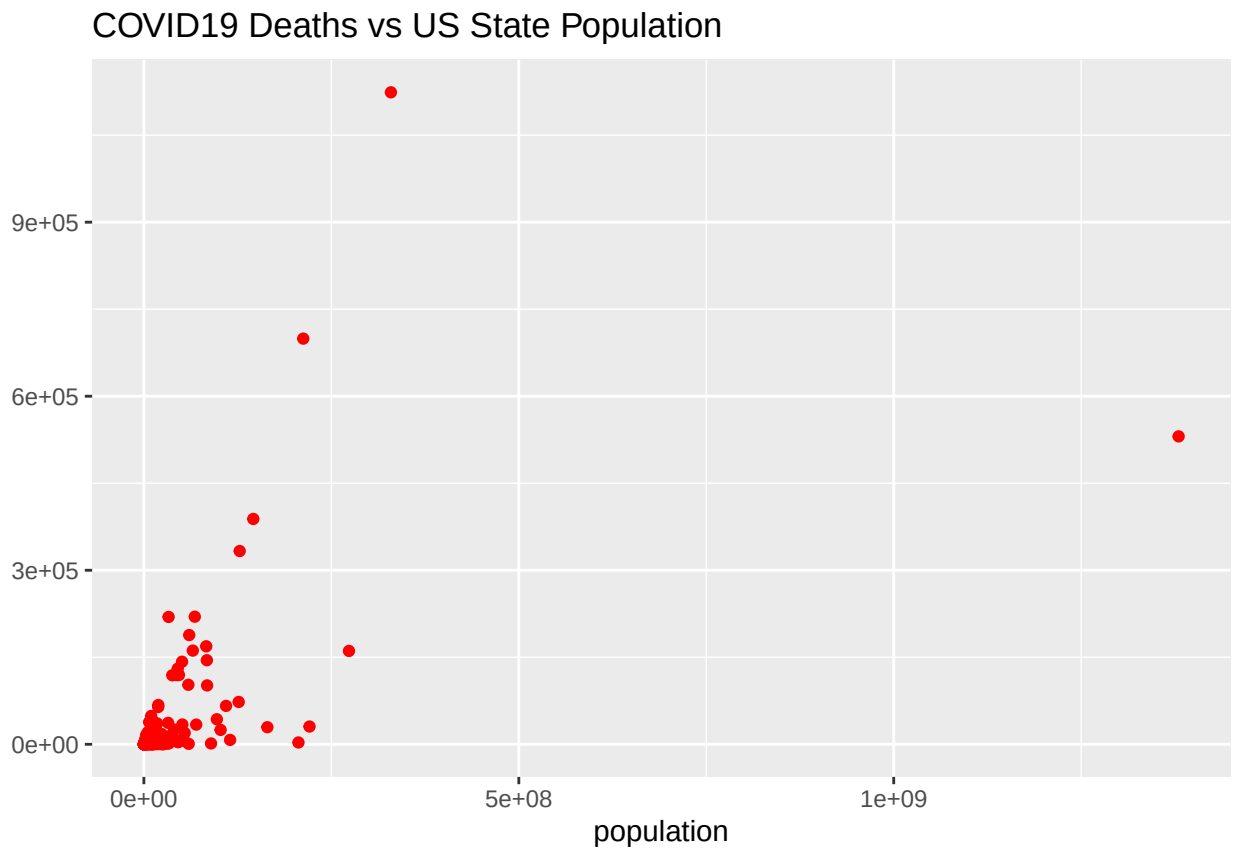
There appears to be a very strong positive linear relationship between the population and cases and population and deaths. Interestingly, the number of deaths appears to increase more with population than the number of cases.

I also want to take a look at the global cases.

```
# Get global deaths and cases per country
global_by_country <- global %>%
                     group_by(Country_Region) %>%
                     summarize(deaths = max(deaths), cases = max(cases), population = max(Population)) %
                     filter(cases > 0, population > 0)

# Cases and pop
global_by_country %>% ggplot() + geom_point(aes(x = population, y = cases), color="blue") + labs(title =
```




```
# Deaths and pop
global_by_country %>% ggplot() + geom_point(aes(x = population, y = deaths), color="red") + labs(title =
```

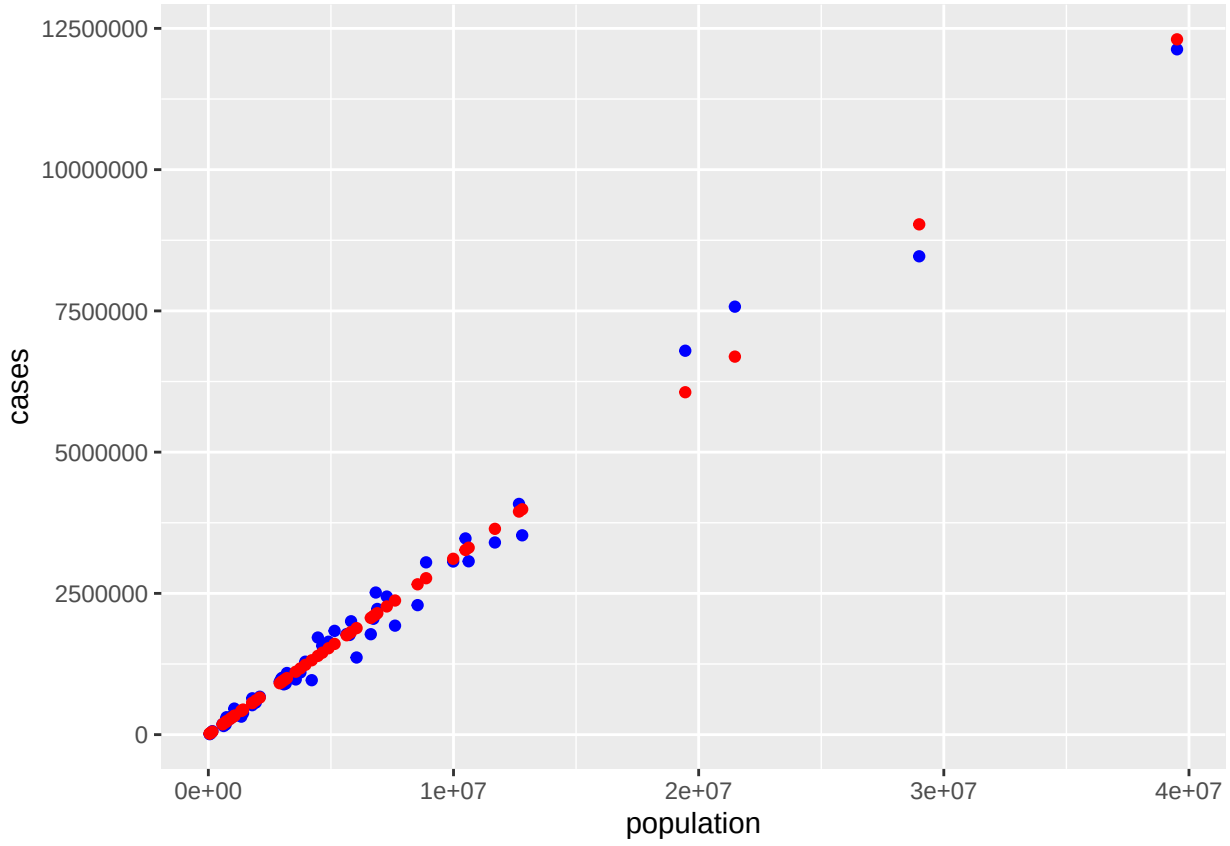


The global values appear to also be increasing in a positive linear fashion. The difference seems that there are some instances where both cases and deaths do not match the same linear increase that most other countries do. For this reason, I will stick to making a model for the US states and population.

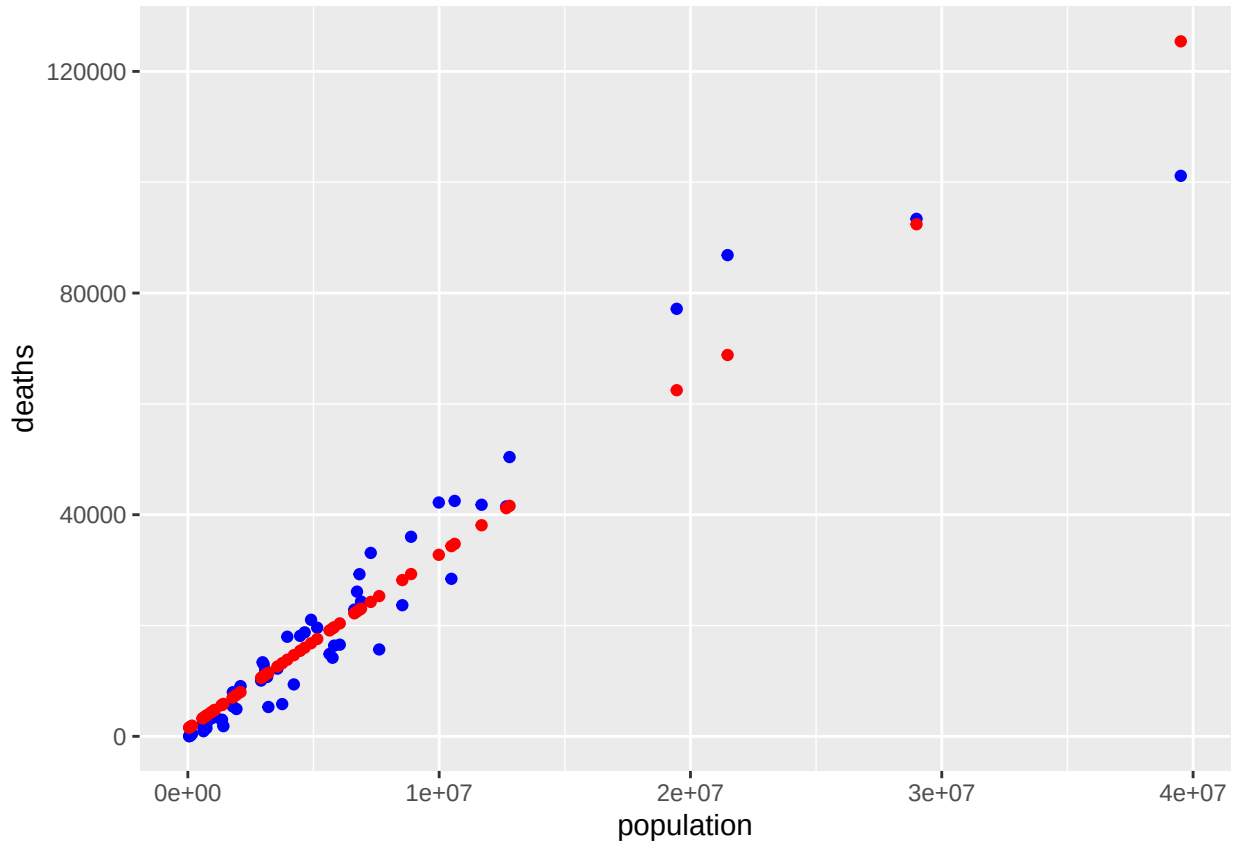
US cases and deaths model

Next, I will create a linear regression model to predict each of the variables (cases and deaths) as a function of population.

```
#############
# Cases Model

# Cases per state as a function of state population
cases_model <- lm(cases ~ population, data = US_state_totals)

# Predict on US state totals
cases_pred <- US_state_totals %>% mutate(pred = predict(cases_model))

# Plot predicted values compared to actual values
cases_pred %>% ggplot() + geom_point(aes(x = population, y = cases), color="blue") +
                  geom_point(aes(x = population, y = pred), color = "red")
```




```
##############
# Deaths Model

# Deaths per state as a function of state population
deaths_model <- lm(deaths ~ population, data = US_state_totals)

# Predict on US state totals
deaths_pred <- US_state_totals %>% mutate(pred = predict(deaths_model))

# Plot predicted values compared to actual values
deaths_pred %>% ggplot() + geom_point(aes(x = population, y = deaths), color="blue") +
                  geom_point(aes(x = population, y = pred), color = "red")
```



Bias and Conclusions

For the Covid 19 dataset, I looked at the cases and deaths of the data and was able to investigate and analyse the relationships between them. I also performed analysis the population over the time and how it might relate to the cases and deaths. However, the outcome was that the dataset for the population did not provide changes over time and therefore was not able to be used. Instead, I analysed the relationship between cases and population and deaths and population. In this way, I was able to find out that the states with larger population had larger case and death numbers which makes sense. However, the increase in deaths with population was much steeper/larger than the increase in cases with population. This could mean many things, but one of those could be that due to a larger population and larger cases, the chances of more deaths is much greater. For a personal bias, I wanted to analyse specific states and their relationship to cases/deaths. I opted to not do this as my personal bias against would potentially affect this. Instead, I decided to analyse all states together and not differentiate between any one state. This was how I tried to mitigate my personal bias. Another point of personal bias is that I am from the United States, so analyzing the US could skew my analysis of the global population. What I could do is also analyse all country totals at the end of the data set and create a similar population to case and population to deaths model without looking at any of the countries. This would also remove personal bias against other countries.

Appendix

```
sessionInfo()
```

```
## R version 4.5.1 (2025-06-13 ucrt)
```

```
## Platform: x86_64-w64-mingw32/x64
## Running under: Windows 11 x64 (build 26100)
##
## Matrix products: default
##   LAPACK version 3.12.1
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## time zone: America/Denver
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] lubridate_1.9.4 forcats_1.0.0   stringr_1.5.1   dplyr_1.1.4
##  [5] purrr_1.1.0     readr_2.1.5     tidyr_1.3.1     tibble_3.3.0
##  [9] ggplot2_3.5.2   tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] bit_4.6.0          gtable_0.3.6       crayon_1.5.3       compiler_4.5.1
##  [5] tidyselect_1.2.1   parallel_4.5.1     scales_1.4.0       yaml_2.3.10
##  [9] fastmap_1.2.0      R6_2.6.1           labeling_0.4.3     generics_0.1.4
## [13] curl_6.4.0         knitr_1.50         pillar_1.11.0      RColorBrewer_1.1-3
## [17] tzdb_0.5.0         rlang_1.1.6        utf8_1.2.6         stringi_1.8.7
## [21] xfun_0.52          bit64_4.6.0-1      timechange_0.3.0   cli_3.6.5
## [25] withr_3.0.2        magrittr_2.0.3     digest_0.6.37      grid_4.5.1
## [29] vroom_1.6.5        rstudioapi_0.17.1  hms_1.1.3          lifecycle_1.0.4
## [33] vctrs_0.6.5        evaluate_1.0.4     glue_1.8.0         farver_2.1.2
## [37] rmarkdown_2.29     tools_4.5.1        pkgconfig_2.0.3    htmltools_0.5.8.1
```