Covid 19 Data

T. Pacheco

2022-08-15

R Markdown

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --v ggplot2 3.3.6     v
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

## Warning: package 'tidyr' was built under R version 3.6.2

## Warning: package 'readr' was built under R version 3.6.2

## Warning: package 'purrr' was built under R version 3.6.2

## Warning: package 'forcats' was built under R version 3.6.2

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covi

file_names <- c("time_series_covid19_confirmed_US.csv", "time_series_covid19_confirmed_global.csv", "ti

urls <- str_c(url_in, file_names)
```

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

```
## Rows: 3342 Columns: 1020-- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1014): 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.
```

```
global_cases <- read_csv(urls[2])
```

```
## Rows: 289 Columns: 1013-- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1011): 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_deaths <- read_csv(urls[3])
```

```
## Rows: 3342 Columns: 1021-- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key
## dbl (1015): 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_deaths <- read_csv(urls[4])
```

```
## Rows: 289 Columns: 1013-- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): Province/State, Country/Region
## dbl (1011): 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.
```

Take the data sets for Global Cases and Global Deaths and make them usable sets with common columns. Then create a data set named Global that combines the useful data from both sets.

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 3.6.2
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

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

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

global <- global_cases %>%
  full_join(global_deaths) %>%
  rename(Country_Region = 'Country/Region', Province_State = 'Province/State') %>%
  mutate(date = mdy(date))
```

```
## Joining, by = c("Province/State", "Country/Region", "date")
```

Like global cases, the US data sets are made to be usable and then combined into the US data set.

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

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

us <- us_cases %>%
  full_join(us_deaths)
```

```
## Joining, by = c("Admin2", "Province_State", "Country_Region", "Combined_Key",
## "date")
```

```
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))
```

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

The graphs below show the rate of change in cases in both the United States and California, notice how the change in deaths is in line with cases.

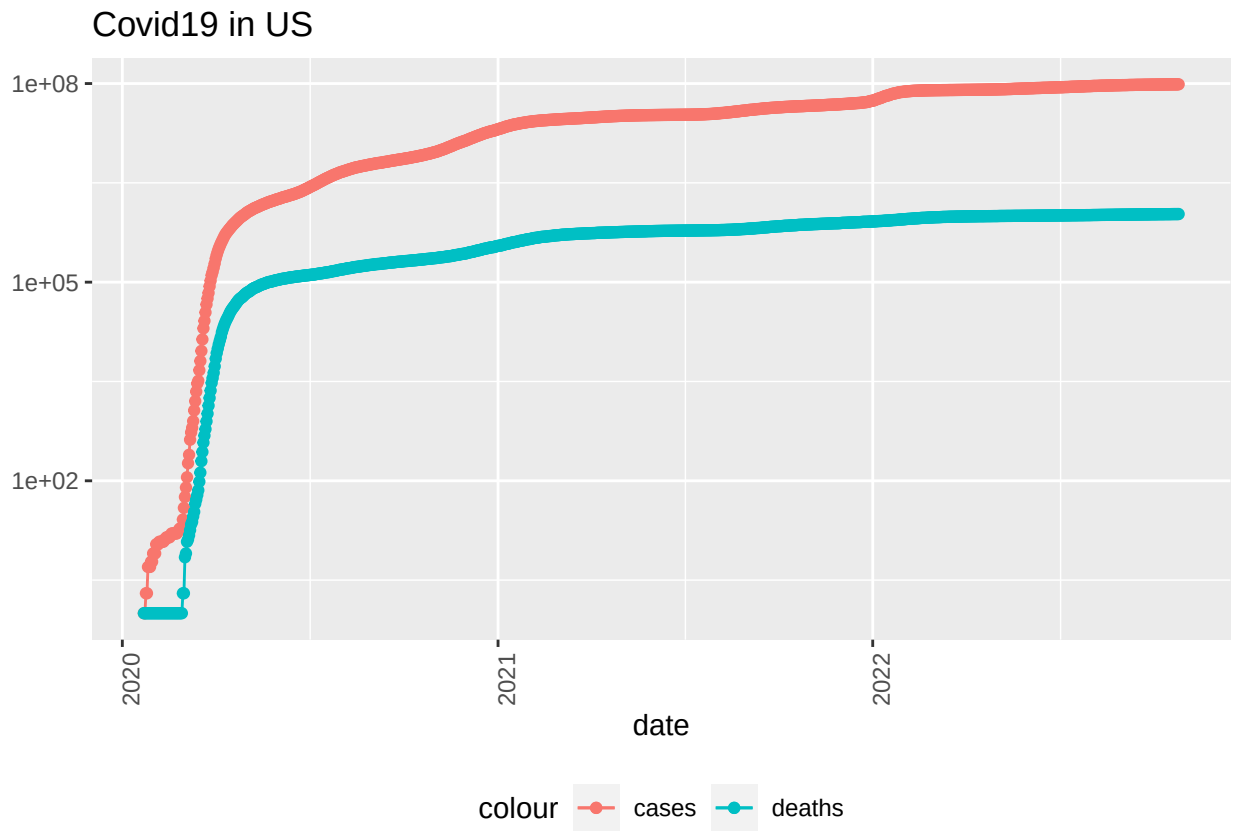
```
us_by_state <- us %>%
  group_by(Province_State, Country_Region, date) %>%
  summarize(cases = sum(cases), deaths = sum(deaths), Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths *1000000 / Population) %>%
  select(Province_State, Country_Region, date, cases, deaths, deaths_per_mill,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

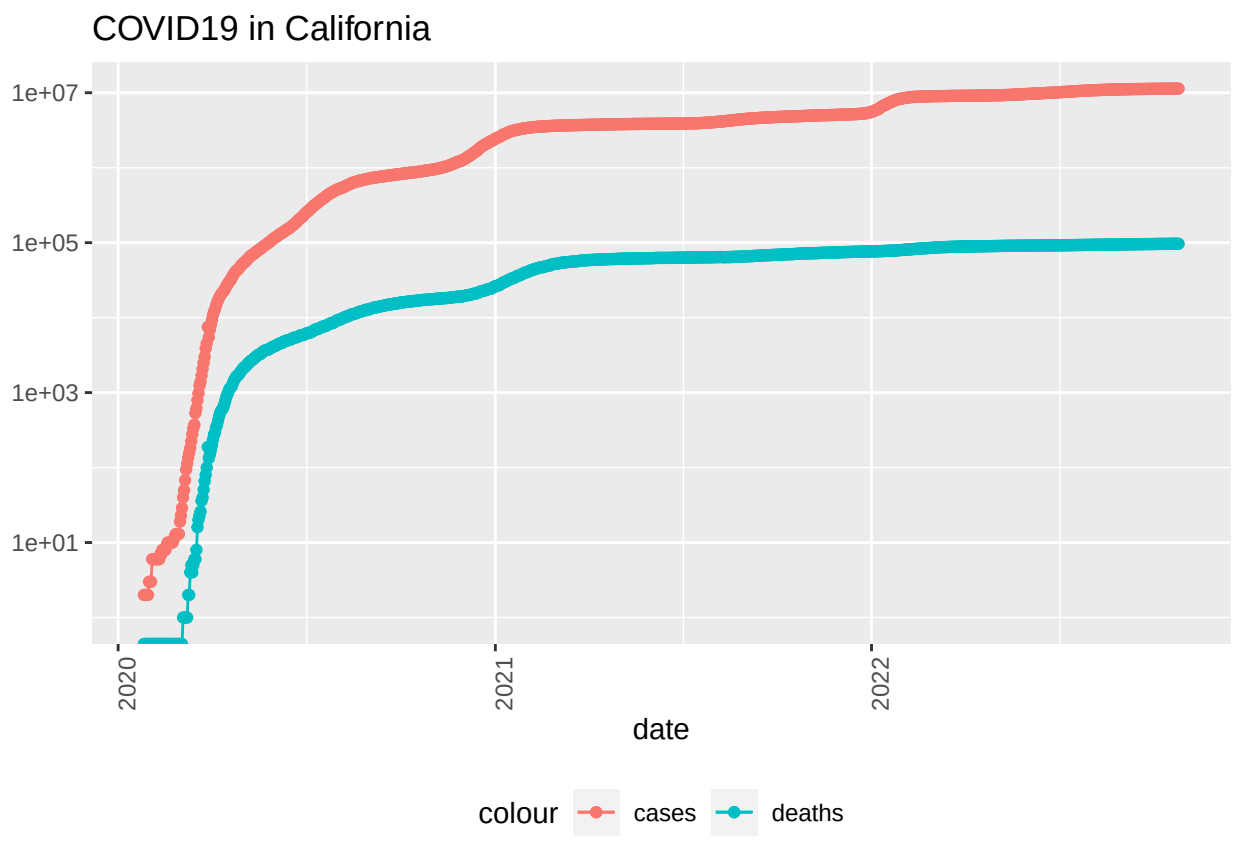
```
us_totals %>%
  filter(cases > 0) %>%
  ggplot(aes(x = date, y = cases)) + geom_line(aes(color = "cases")) + geom_point(aes(color = "cases"))
```




```
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"))
```

```
## Warning: Transformation introduced infinite values in continuous y-axis
```

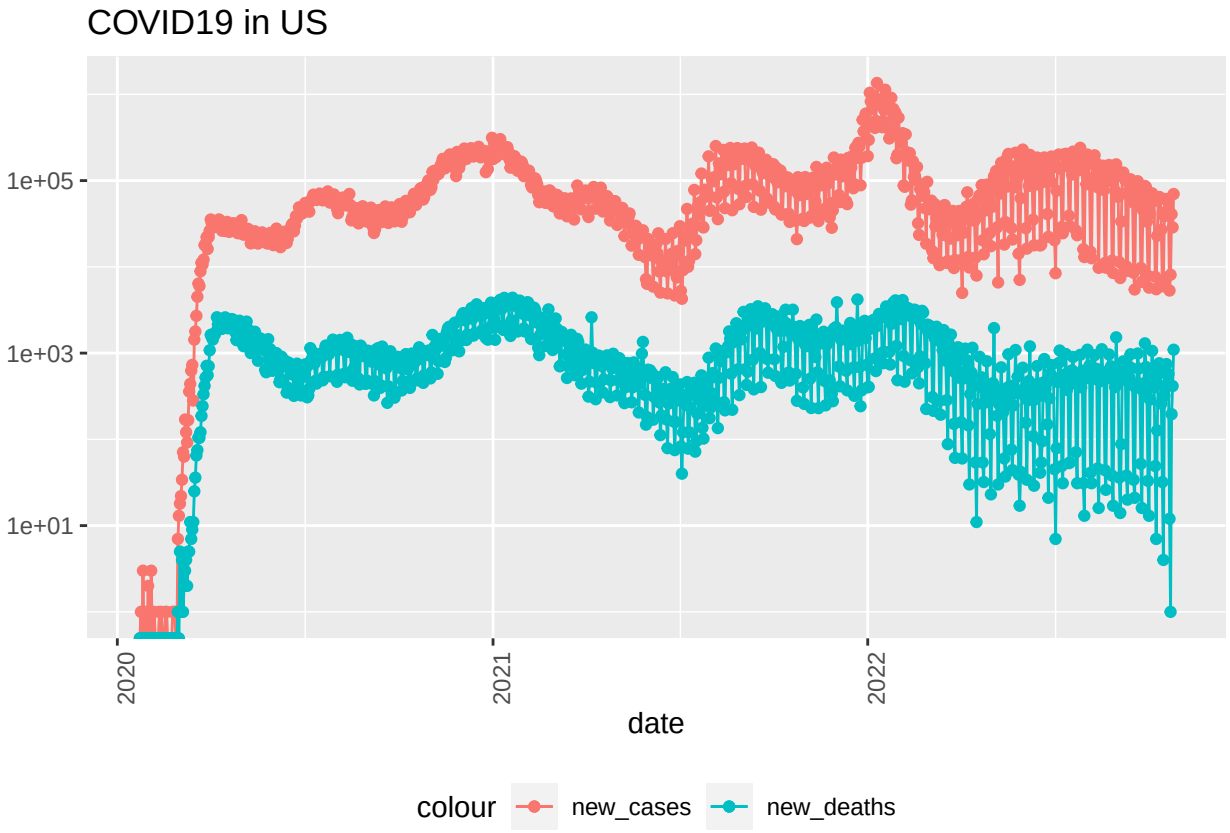
```
## Warning: Transformation introduced infinite values in continuous y-axis
```




```
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_totals %>%
  ggplot(aes(x = date, y = new_cases)) + geom_line(aes(color = "new_cases")) + geom_point(aes(color = "
```

```
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis
```

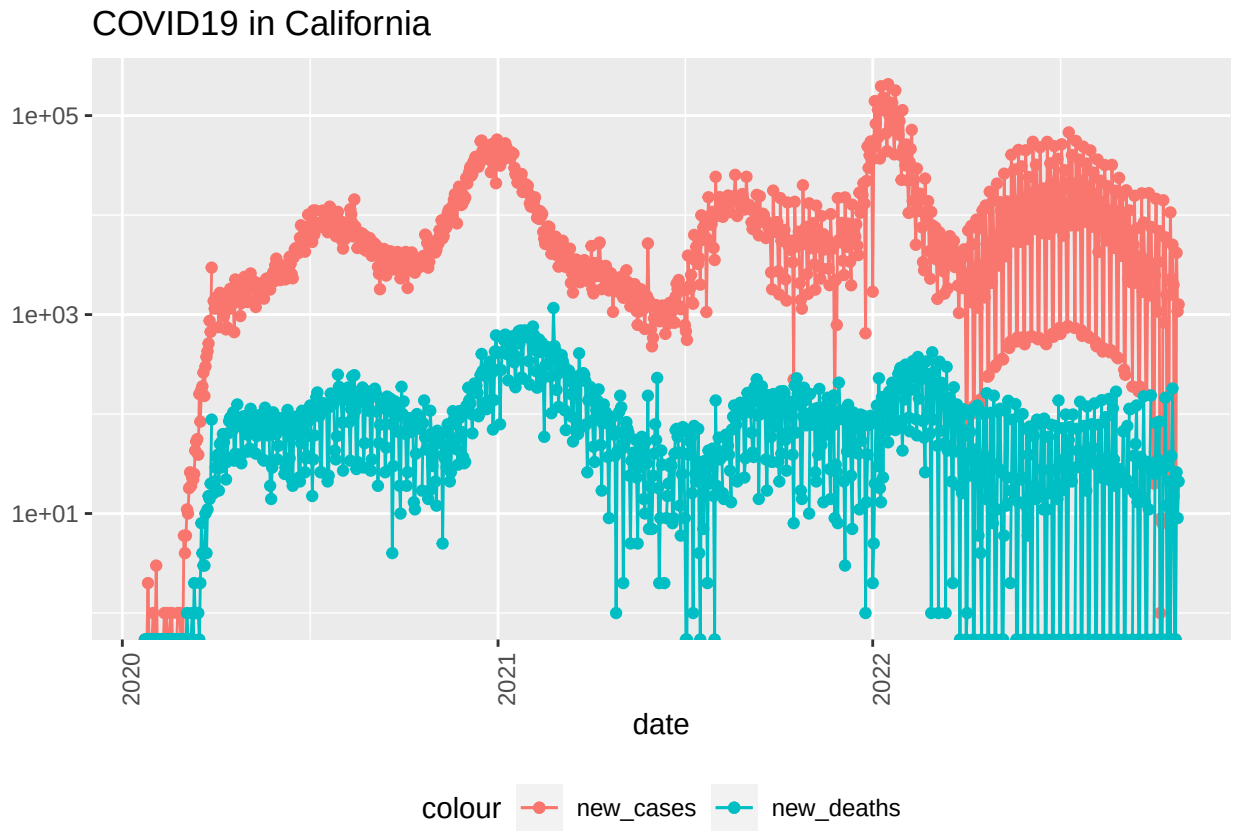
```
## Warning: Removed 1 row(s) containing missing values (geom_path).

## Warning: Removed 1 rows containing missing values (geom_point).

## Warning: Removed 1 row(s) containing missing values (geom_path).

## Warning: Removed 2 rows containing missing values (geom_point).
```




```
state <- "California"
us_by_state %>%
  filter(Province_State == state) %>%
  ggplot(aes(x = date, y = new_cases)) + geom_line(aes(color = "new_cases")) + geom_point(aes(color = "
```

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

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis

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

```
## Warning: Transformation introduced infinite values in continuous y-axis

## Warning in self$trans$transform(x): NaNs produced

## Warning: Transformation introduced infinite values in continuous y-axis

## Warning: Removed 1 row(s) containing missing values (geom_path).

## Warning: Removed 3 rows containing missing values (geom_point).

## Warning: Removed 1 row(s) containing missing values (geom_path).

## Warning: Removed 13 rows containing missing values (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 = 10
  filter(cases > 0, population > 0)


us_state_totals %>%
  slice_min(deaths_per_thou, n = 10) %>%
  select(deaths_per_thou, cases_per_thou, everything())
```

```
## # A tibble: 10 x 6
##    deaths_per_thou cases_per_thou Province_State           deaths  cases popul~1
##              <dbl>          <dbl> <chr>                     <dbl>  <dbl>   <dbl>
##  1           0.611           148. American Samoa               34 8.25e3   55641
##  2           0.725           239. Northern Mariana Islands     40 1.32e4   55144
##  3           1.16            217. Virgin Islands              124 2.33e4  107268
##  4           1.19            232. Vermont                     740 1.45e5  623989
##  5           1.20            256. Hawaii                     1704 3.62e5 1415872
##  6           1.40            260. Puerto Rico                5243 9.77e5 3754939
##  7           1.57            325. Utah                       5047 1.04e6 3205958
##  8           1.91            405. Alaska                     1413 3.00e5  740995
##  9           1.91            241. Washington                14550 1.84e6 7614893
## 10           1.96            221. Maine                      2641 2.97e5 1344212
## # ... with abbreviated variable name 1: population
```

```
us_state_totals %>%
  slice_max(deaths_per_thou, n = 10) %>%
  select(deaths_per_thou, cases_per_thou, everything())
```

```
## # A tibble: 10 x 6
##    deaths_per_thou cases_per_thou Province_State deaths   cases population
##              <dbl>          <dbl> <chr>           <dbl>   <dbl>      <dbl>
##  1            4.36           314. Mississippi     12968  933065    2976149
##  2            4.33           314. Arizona         31548 2287886    7278717
##  3            4.31           305. Oklahoma        17048 1205519    3956971
##  4            4.19           312. Alabama         20533 1531305    4903185
##  5            4.19           339. West Virginia    7502  606794    1792147
##  6            4.13           318. Arkansas        12452  958675    3017804
##  7            4.12           299. New Mexico       8631  626168    2096829
##  8            4.11           345. Tennessee       28074 2357243    6829174
##  9            3.93           289. Michigan        39250 2886176    9986857
## 10            3.93           314. New Jersey      34877 2787227    8882190
```

Based on the most recent data, the linear model does not fit well with the deaths per cases see throughout the United States. Some States have jumped in death rate while others have dropped off.

```
mod <- lm(deaths_per_thou ~ cases_per_thou, data = us_state_totals)

us_state_totals %>%
  slice_min(cases_per_thou)
```

```
## # A tibble: 1 x 6
##   Province_State deaths cases population cases_per_thou deaths_per_thou
##   <chr>           <dbl> <dbl>      <dbl>          <dbl>           <dbl>
## 1 American Samoa     34  8253      55641           148.           0.611
```

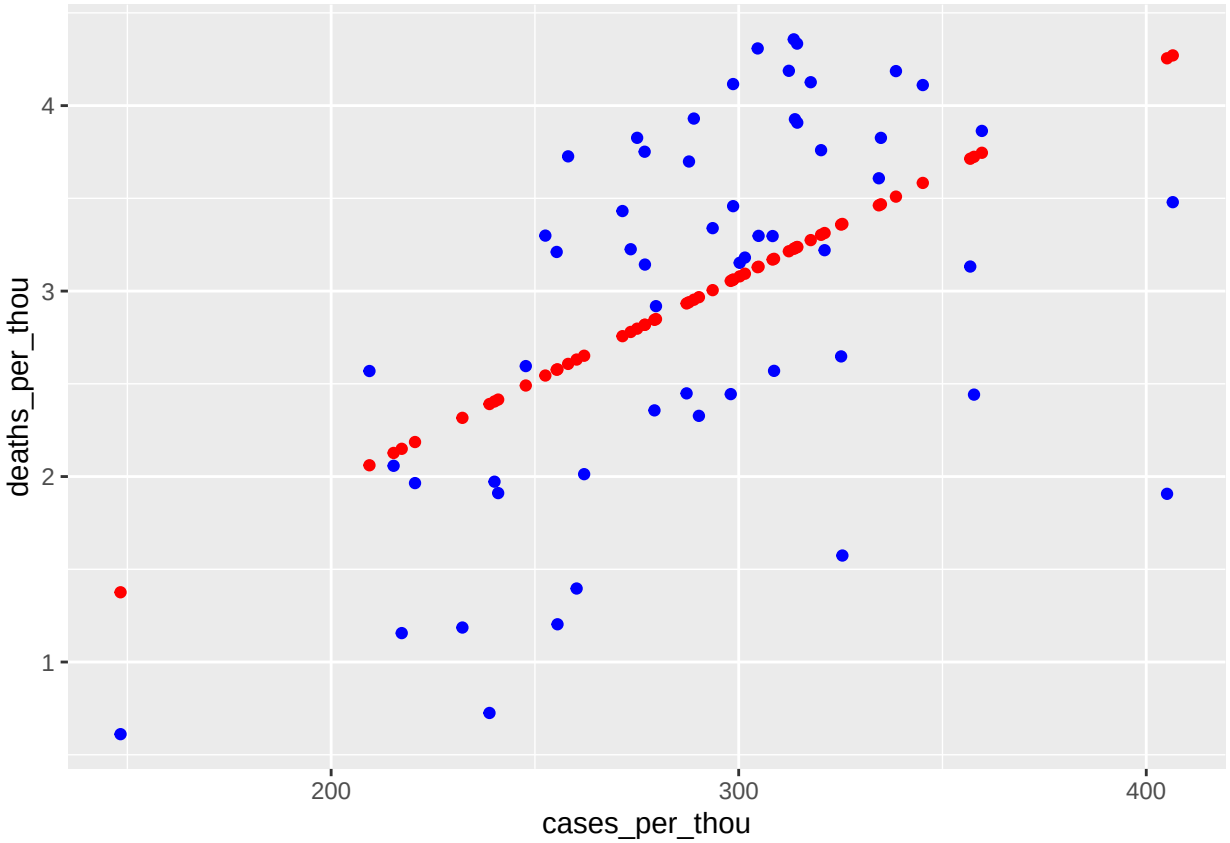
```
us_state_totals %>%
  slice_max(cases_per_thou)
```

```
## # A tibble: 1 x 6
##   Province_State deaths  cases population cases_per_thou deaths_per_thou
##   <chr>           <dbl>  <dbl>      <dbl>          <dbl>           <dbl>
## 1 Rhode Island     3686 430636    1059361           407.            3.48
```

```
x_grid <- seq(1, 151)
new_df <- tibble(cases_per_thou = x_grid)
us_state_totals %>%
  mutate(pred = predict(mod))
```

```
## # A tibble: 56 x 7
##    Province_State       deaths    cases population cases_per_thou deaths~1  pred
##    <chr>                 <dbl>    <dbl>      <dbl>          <dbl>    <dbl> <dbl>
##  1 Alabama               20533  1531305    4903185           312.    4.19   3.21
##  2 Alaska                 1413   300177     740995           405.    1.91   4.26
##  3 American Samoa           34     8253      55641           148.    0.611  1.38
##  4 Arizona               31548  2287886    7278717           314.    4.33   3.24
##  5 Arkansas              12452   958675    3017804           318.    4.13   3.27
##  6 California            96748 11348431   39512223           287.    2.45   2.93
##  7 Colorado              13402  1671397    5758736           290.    2.33   2.97
##  8 Connecticut           11448   910367    3565287           255.    3.21   2.58
##  9 Delaware               3136   312655     973764           321.    3.22   3.31
## 10 District of Columbia   1392   169436     705749           240.    1.97   2.40
## # ... with 46 more rows, and abbreviated variable name 1: deaths_per_thou
## # i Use `print(n = ...)` to see more rows
```

```
us_tot_w_pred <- us_state_totals %>%
  mutate(pred = predict(mod))

us_tot_w_pred %>%
  ggplot() + geom_point(aes(x = cases_per_thou, y = deaths_per_thou), color = "blue") + geom_point(aes(
```


deaths_per_thou
4
3
2
1
200
300
400
cases_per_thou