Covid_analysis

Siri Devarapalli

Import data

We are considering https://github.com/nytimes/covid-19-data and [Johns Hopkins University]https://github.com/CSSEGISandData/COVID-19 github sites.

```
url_in <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_cov
file_names <- c("_confirmed_global.csv","_deaths_global.csv","_confirmed_US.csv","_deaths_US.csv")
urls <- str_c(url_in,file_names)
```

```
global_cases <- read_csv(urls[1], show_col_types = FALSE)
global_deaths <- read_csv(urls[2], show_col_types = FALSE)
US_cases <- read_csv(urls[3], show_col_types = FALSE)
US_deaths <- read_csv(urls[4], show_col_types = FALSE)
```

Let's clean up the data to remove unnecessary columns and rename some columns for our convienience.

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

```
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:169353      Length:169353      Min.   :2020-01-22   Min.   :       0
##  Class :character   Class :character   1st Qu.:2020-06-21   1st Qu.:     146
##  Mode  :character   Mode  :character   Median :2020-11-20   Median :    2297
##                                        Mean   :2020-11-20   Mean   :  286349
##                                        3rd Qu.:2021-04-21   3rd Qu.:   51809
##                                        Max.   :2021-09-19   Max.   :42087432
```

```
##      deaths
##  Min.   :     0
##  1st Qu.:     1
##  Median :    35
##  Mean   :  6604
##  3rd Qu.:   841
##  Max.   :673763
```

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

Let's clean US_cases and US_deaths also :

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

US_deaths <- US_deaths %>%
  pivot_longer(cols = -(UID: Population), names_to = "date", values_to = "deaths") %>%
  dplyr::select(Admin2:deaths) %>%
  mutate(date = mdy(date)) %>% dplyr::select(-c(Lat,Long_))
# US contains case data and death data.
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)
```

We do not have population data, having population data will give insight.

```
uid <- read.csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/UI
uid <- uid %>% dplyr::select(-c(Lat,Long_,Combined_Key, code3,iso2,iso3,Admin2))
```

Merge the population data into global data

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

```
## # A tibble: 153,341 x 7
##    Province_State Country_Region date       cases deaths Population Combined_Key
##    <chr>          <chr>          <date>     <dbl>  <dbl>      <int> <chr>
##  1 <NA>           Afghanistan    2020-02-24     5      0         NA Afghanistan
##  2 <NA>           Afghanistan    2020-02-25     5      0         NA Afghanistan
```

```
##  3 <NA>           Afghanistan    2020-02-26     5      0         NA Afghanistan
##  4 <NA>           Afghanistan    2020-02-27     5      0         NA Afghanistan
##  5 <NA>           Afghanistan    2020-02-28     5      0         NA Afghanistan
##  6 <NA>           Afghanistan    2020-02-29     5      0         NA Afghanistan
##  7 <NA>           Afghanistan    2020-03-01     5      0         NA Afghanistan
##  8 <NA>           Afghanistan    2020-03-02     5      0         NA Afghanistan
##  9 <NA>           Afghanistan    2020-03-03     5      0         NA Afghanistan
## 10 <NA>           Afghanistan    2020-03-04     5      0         NA Afghanistan
## # ... with 153,331 more rows
```

Visualisation

To find out the total number cases per million in each state:

```
US_by_state <- US %>% group_by (Province_State, Country_Region, date) %>%
  summarise(cases = sum(cases), deaths = sum(deaths), Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths * 1000000 / Population) %>%
  dplyr::select(Province_State, Country_Region, date, cases, deaths, deaths_per_mill, Population) %>% u
```

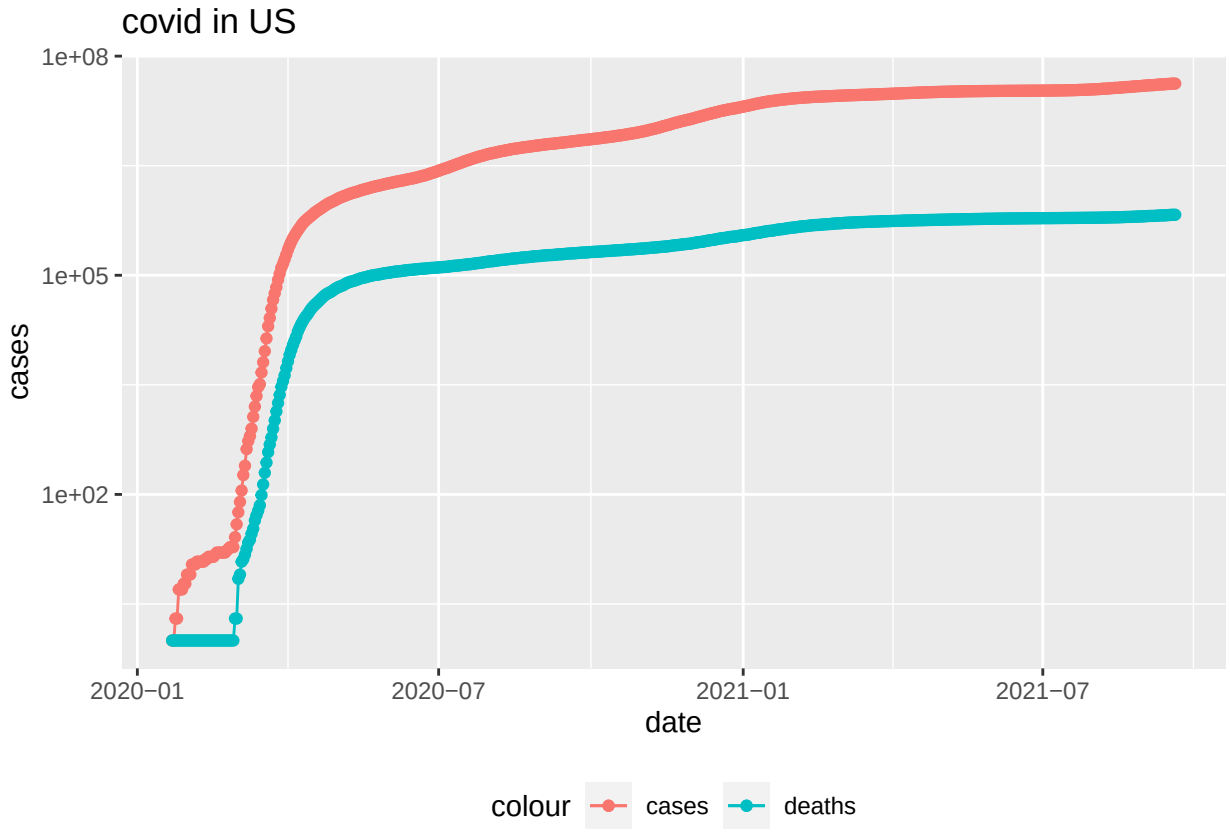
```
## `summarise()` has grouped output by 'Province_State', 'Country_Region'. You can override using the `
```

Similarly, let us find out deaths per million in each state.

```
US_totals <- US_by_state %>% group_by (Country_Region, date) %>%
  summarise(cases = sum(cases), deaths = sum(deaths), Population = sum(Population)) %>%
  mutate(deaths_per_mill = deaths * 1000000 / Population) %>%
  dplyr::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.
```

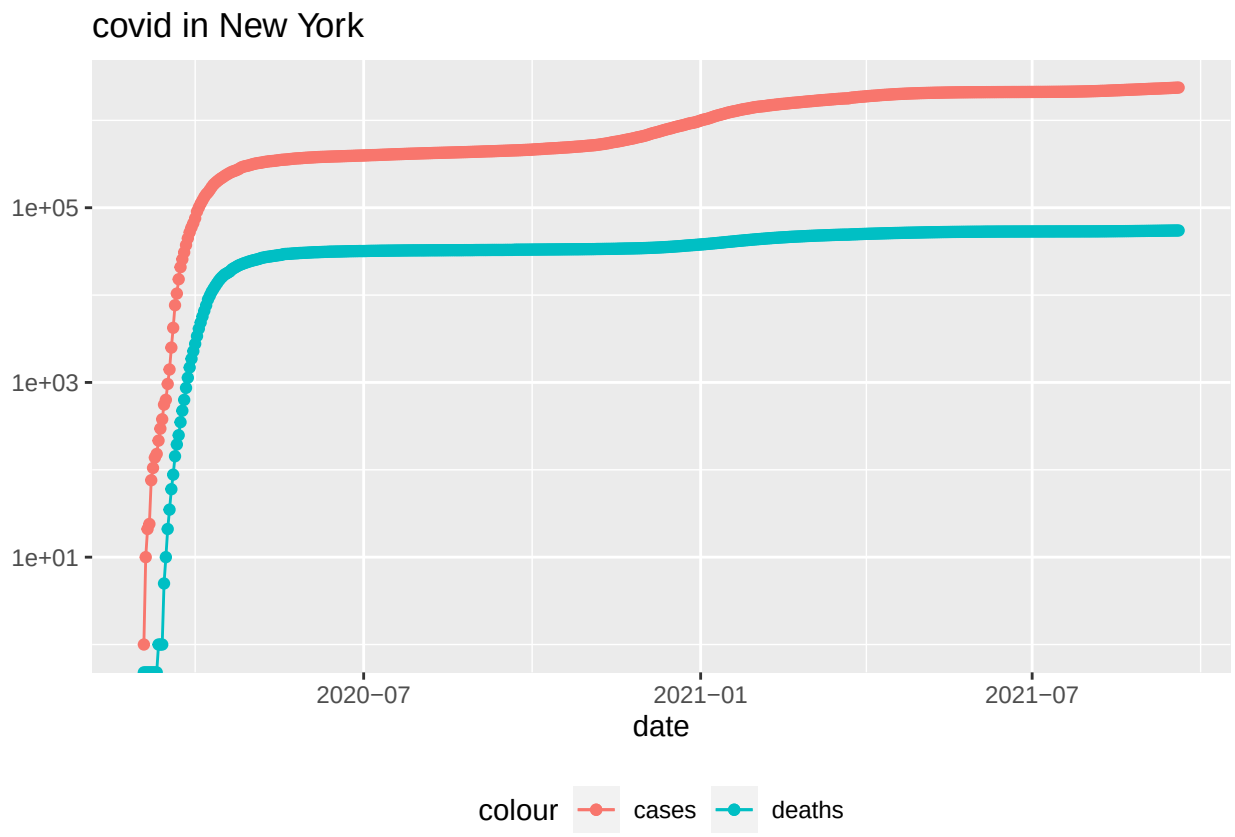
```
US_totals %>% filter(cases>0) %>% ggplot(aes(x = date, y = cases))+ geom_line(aes(color = "cases"))+
  geom_point(aes(color ="cases")) + geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) + scale_y_log10() + theme(legend.position ="bottom") +
  labs(title = "covid in US")
```



We can say that the cases in US are not increasing drastically.

```
state <-"New York"
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") +
  labs(title = str_c("covid in ", state), y = NULL)
```

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

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



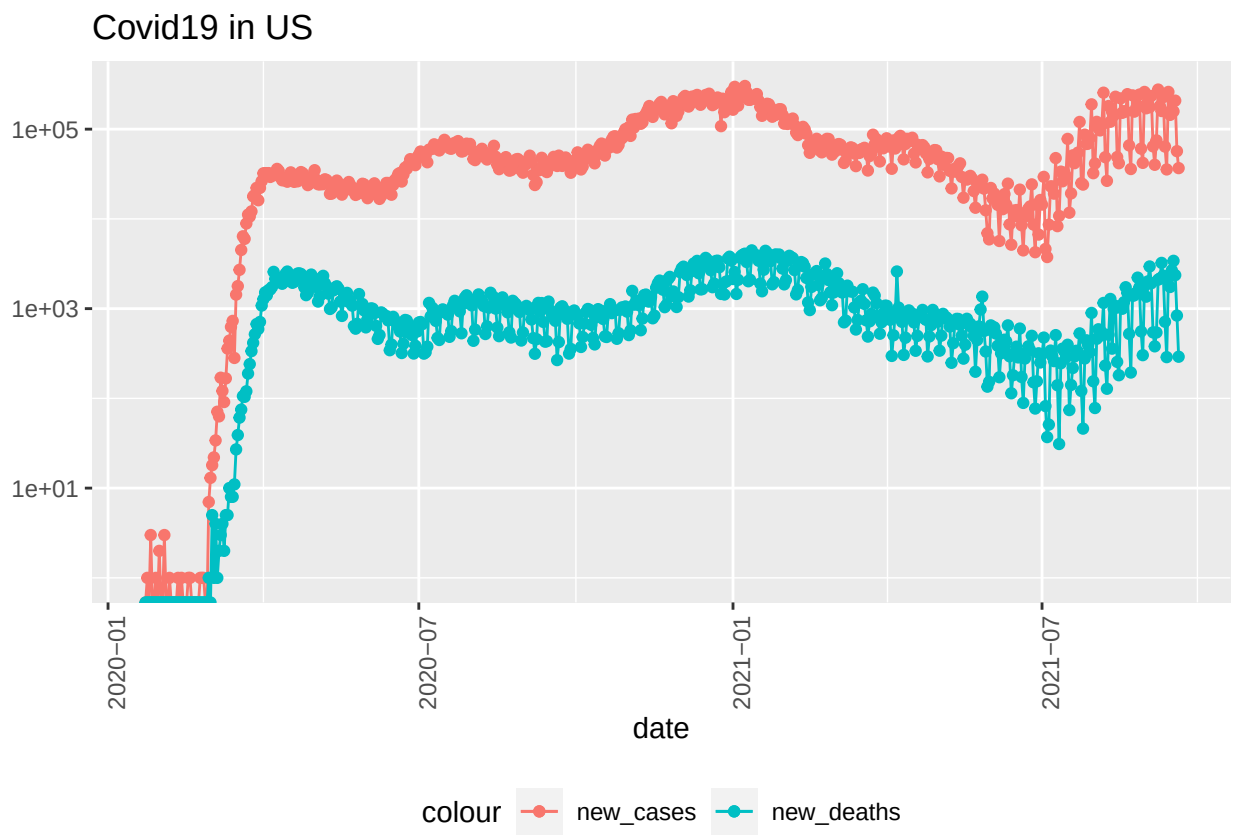
Analyze the data

```
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 = "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: Transformation introduced infinite values in continuous y-axis

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

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

## 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 1 rows containing missing values (geom_point).
```

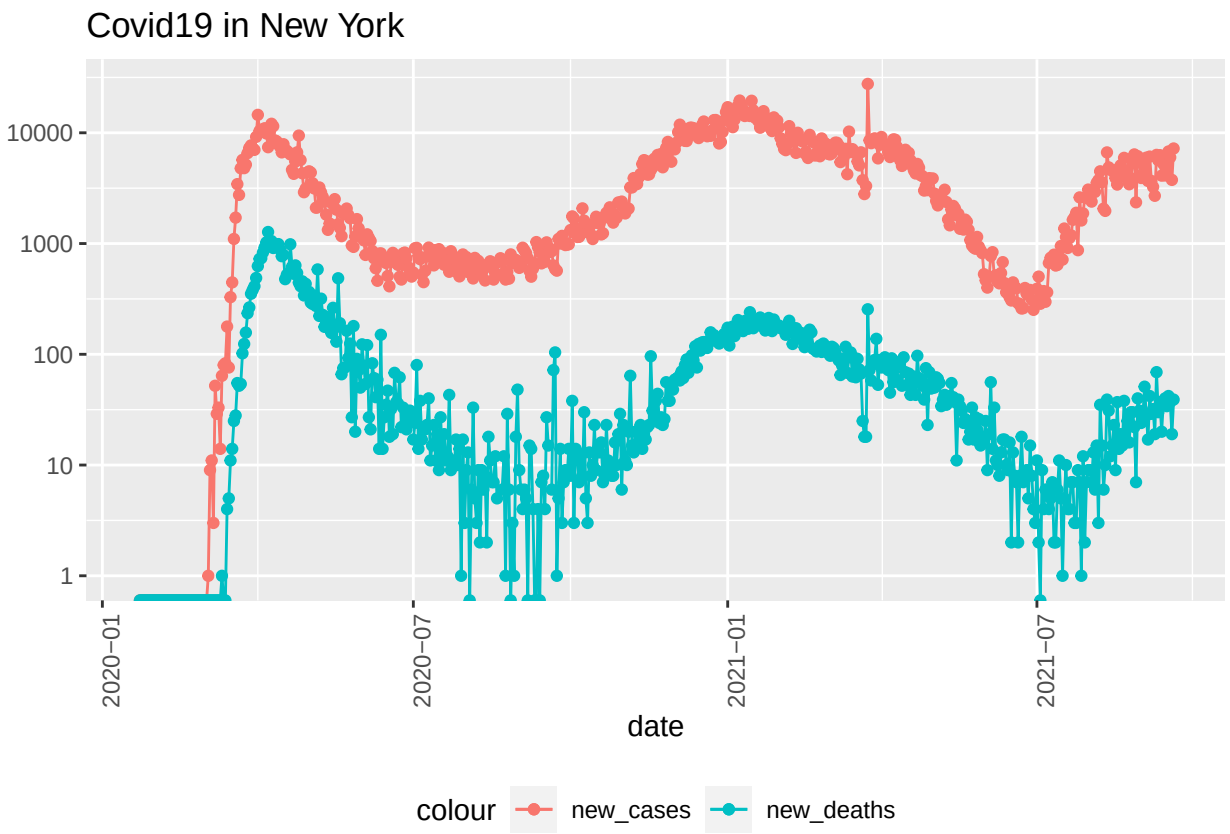


Visualise new_cases and new_deaths in New York state.

```
state <- "New York"
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("Covid19 in ",state), y= NULL)
```

```
## 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 1 rows containing missing values (geom_point).

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

## Warning: Removed 6 rows containing missing values (geom_point).
```



What are worst states and the best states ?

```
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 / population) %>%
  filter(cases > 0, population > 0)
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 Northern Mariana Islands      2    263      55144           4.77          0.0363
##  2 Vermont                     298  31634     623989          50.7           0.478
##  3 Hawaii                      714  75480    1415872          53.3           0.504
##  4 Virgin Islands               67   6458     107268          60.2           0.625
##  5 Alaska                      469 100360     740995         135.            0.633
##  6 Maine                       984  83910    1344212          62.4           0.732
##  7 Puerto Rico                3074 179144    3754939          47.7           0.819
##  8 Oregon                     3569 309841    4217737          73.5           0.846
##  9 Utah                       2787 490985    3205958         153.            0.869
## 10 Washington                 7201 620752    7614893          81.5           0.946
```

```
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 Mississippi      9214  473413    2976149           159.            3.10
##  2 New Jersey      27190 1133228    8882190           128.            3.06
##  3 Louisiana       13418  725637    4648794           156.            2.89
##  4 New York        54904 2373659   19453561           122.            2.82
##  5 Alabama         13210  770391    4903185           157.            2.69
##  6 Arizona         19513 1066803    7278717           147.            2.68
##  7 Massachusetts   18445  790953    6892503           115.            2.68
##  8 Rhode Island     2812  168449    1059361           159.            2.65
##  9 Arkansas         7434  484317    3017804           160.            2.46
## 10 Florida         51240 3528698   21477737           164.            2.39
```

Model the data

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

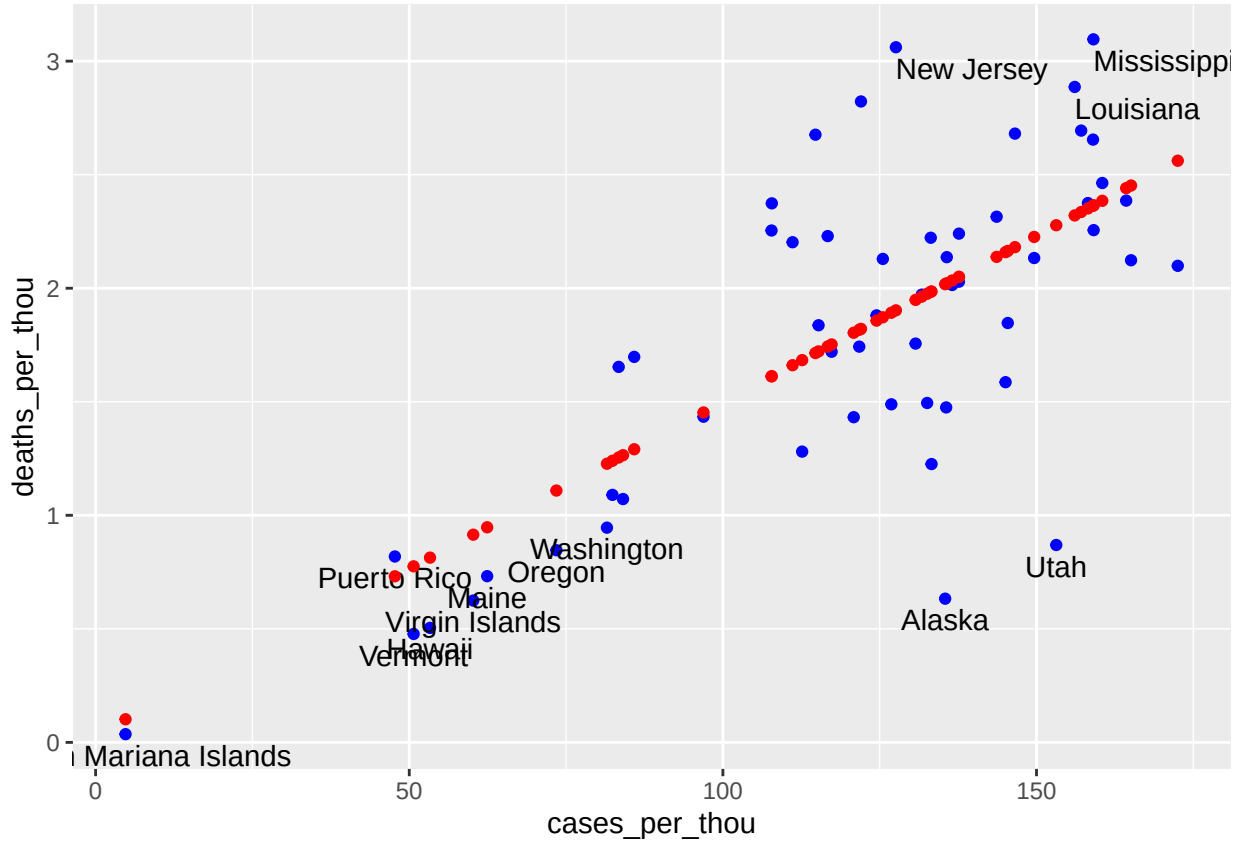
```
##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou, data = US_state_totals)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.40805 -0.29388 -0.02109  0.27406  1.15861
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.031981   0.241179   0.133    0.895
## cases_per_thou 0.014662   0.001924   7.619 4.56e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5044 on 53 degrees of freedom
```

```
## Multiple R-squared:  0.5227, Adjusted R-squared:  0.5137
## F-statistic: 58.05 on 1 and 53 DF,  p-value: 4.556e-10
```

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

```
## # A tibble: 55 x 7
##    Province_State  deaths  cases population cases_per_thou deaths_per_thou  pred
##    <chr>            <dbl>  <dbl>      <dbl>          <dbl>           <dbl> <dbl>
##  1 Alabama          13210 7.70e5    4903185          157.            2.69   2.34
##  2 Alaska             469 1.00e5     740995          135.            0.633  2.02
##  3 Arizona          19513 1.07e6    7278717          147.            2.68   2.18
##  4 Arkansas          7434 4.84e5    3017804          160.            2.46   2.38
##  5 California       67966 4.64e6   39512223          117.            1.72   1.75
##  6 Colorado          7374 6.49e5    5758736          113.            1.28   1.68
##  7 Connecticut       8463 3.84e5    3565287          108.            2.37   1.61
##  8 Delaware          1920 1.28e5     973764          132.            1.97   1.96
##  9 District of Co~   1167 5.89e4     705749           83.4           1.65   1.25
## 10 Florida          51240 3.53e6   21477737          164.            2.39   2.44
## # ... with 45 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_text(aes(x =cases_per_thou,y = deaths_per_thou,
                label = ifelse(deaths_per_thou>quantile(deaths_per_thou, 0.95) ,
                               Province_State,'')),vjust = 1.5,hjust = 0) +
  geom_text(aes(x=cases_per_thou, y =deaths_per_thou,
                label = ifelse(deaths_per_thou<1,Province_State,'')),vjust = 1.5) +
geom_point(aes(x = cases_per_thou, y = pred), color = "red")
```



From the above scatter plot Utah and Alaska are performing better than new jersey, Mississippi and Louisiana.

To find out the deaths because of Covid in each state we can measure deaths/cases

```
US_state_totals <- US_state_totals %>% mutate(death_per_cases = deaths/cases)
glimpse(US_state_totals)
```

```
## Rows: 55
## Columns: 7
## $ Province_State  <chr> "Alabama", "Alaska", "Arizona", "Arkansas", "Californi~
## $ deaths          <dbl> 13210, 469, 19513, 7434, 67966, 7374, 8463, 1920, 1167~
## $ cases           <dbl> 770391, 100360, 1066803, 484317, 4636268, 648642, 3843~
## $ population      <dbl> 4903185, 740995, 7278717, 3017804, 39512223, 5758736, ~
## $ cases_per_thou  <dbl> 157.12052, 135.43951, 146.56470, 160.48657, 117.33756,~
## $ deaths_per_thou <dbl> 2.6941672, 0.6329327, 2.6808296, 2.4633807, 1.7201259,~
## $ death_per_cases <dbl> 0.017147137, 0.004673177, 0.018291100, 0.015349451, 0.~
```

Let us model the data again.

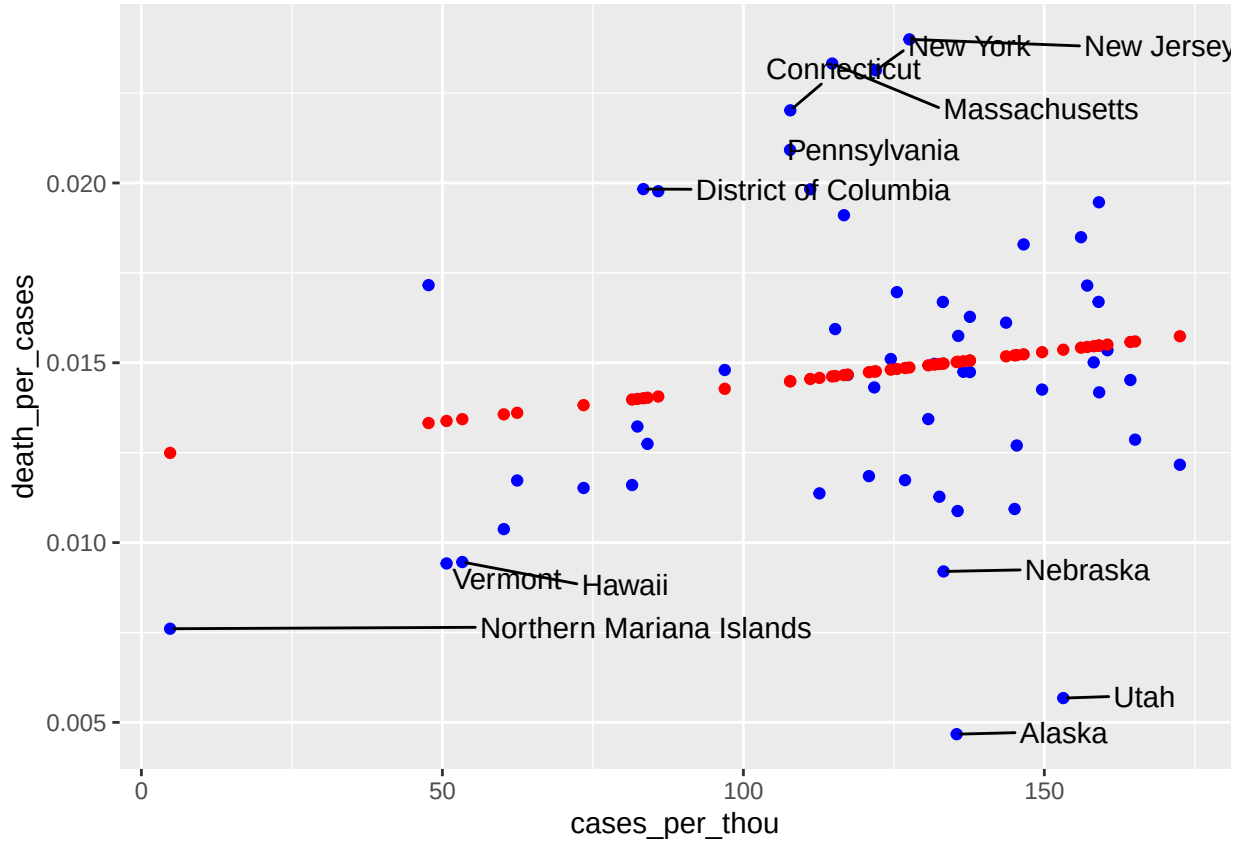
```
mod1 <- lm(death_per_cases ~ cases_per_thou, data = US_state_totals)
summary(mod1)
```

```
##
## Call:
## lm(formula = death_per_cases ~ cases_per_thou, data = US_state_totals)
```

```
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0103473 -0.0028109 -0.0003222  0.0019254  0.0091248
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    1.240e-02  2.008e-03   6.176 9.49e-08 ***
## cases_per_thou 1.934e-05  1.602e-05   1.207    0.233
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.004199 on 53 degrees of freedom
## Multiple R-squared:  0.02675,    Adjusted R-squared:  0.008388
## F-statistic: 1.457 on 1 and 53 DF,  p-value: 0.2328
```

Is there any other to find out the states that handled covid better than others ? Is there any way to find out the number of people who died because of Covid ? Let us find out how many people died out of the cases reported.

```
library(ggrepel)
US_tot_w_pred1 <- US_state_totals %>% mutate(pred = predict(mod1))
US_tot_w_pred1 %>% ggplot() +
geom_point(aes(x = cases_per_thou, y = death_per_cases), color = "blue") +
  geom_text_repel(aes(x =  cases_per_thou, y = death_per_cases,
                      label = ifelse(death_per_cases > quantile(death_per_cases, 0.9) ,
                                     Province_State,'')),vjust = 1.5,hjust = 0) +
  geom_text_repel(aes(x =  cases_per_thou, y = death_per_cases,
                      label = ifelse(death_per_cases < 0.01 ,Province_State,'')),vjust = 1.5)+
  geom_point(aes(x = cases_per_thou, y = pred), color = "red")
```



We see that Nebraska, Utah and Alaska are the best states whereas New jersey, New york and Massachusetts are the worst.

Let us explore the possibility that air quality effects covid, as Covid-19 targets the lungs.

America's Health Rankings provides an analysis of national health on a state-by-state basis by evaluating a historical and comprehensive set of health, environmental and socioeconomic data to determine national health benchmarks and state rankings. We only need the Ranking of each state for our analysis.

```
AQI <- read_csv("https://www.americashealthrankings.org/api/v1/downloads/210")
```

```
## Rows: 49660 Columns: 11

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (4): Report Type, Measure Name, State Name, Source
## dbl (7): Edition, Rank, Value, Score, Lower CI, Upper CI, Source Year

##
## 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.
```

```
AQI <- AQI %>% distinct(`State Name`,`Rank`) %>% select(c(`State Name`,`Rank`))
AQI <- na.omit(AQI)
AQI <- AQI[1:50,]
AQI <- rename(AQI,Province_State = `State Name`)
glimpse(AQI)
```

```
## Rows: 50
## Columns: 2
## $ Province_State <chr> "Alaska", "Alabama", "Arkansas", "Arizona", "California~
## $ Rank           <dbl> 21, 47, 43, 40, 22, 19, 5, 20, 46, 49, 4, 8, 39, 26, 33~
```

```
US_tot_w_pred1<-  full_join(US_tot_w_pred1,AQI)
```

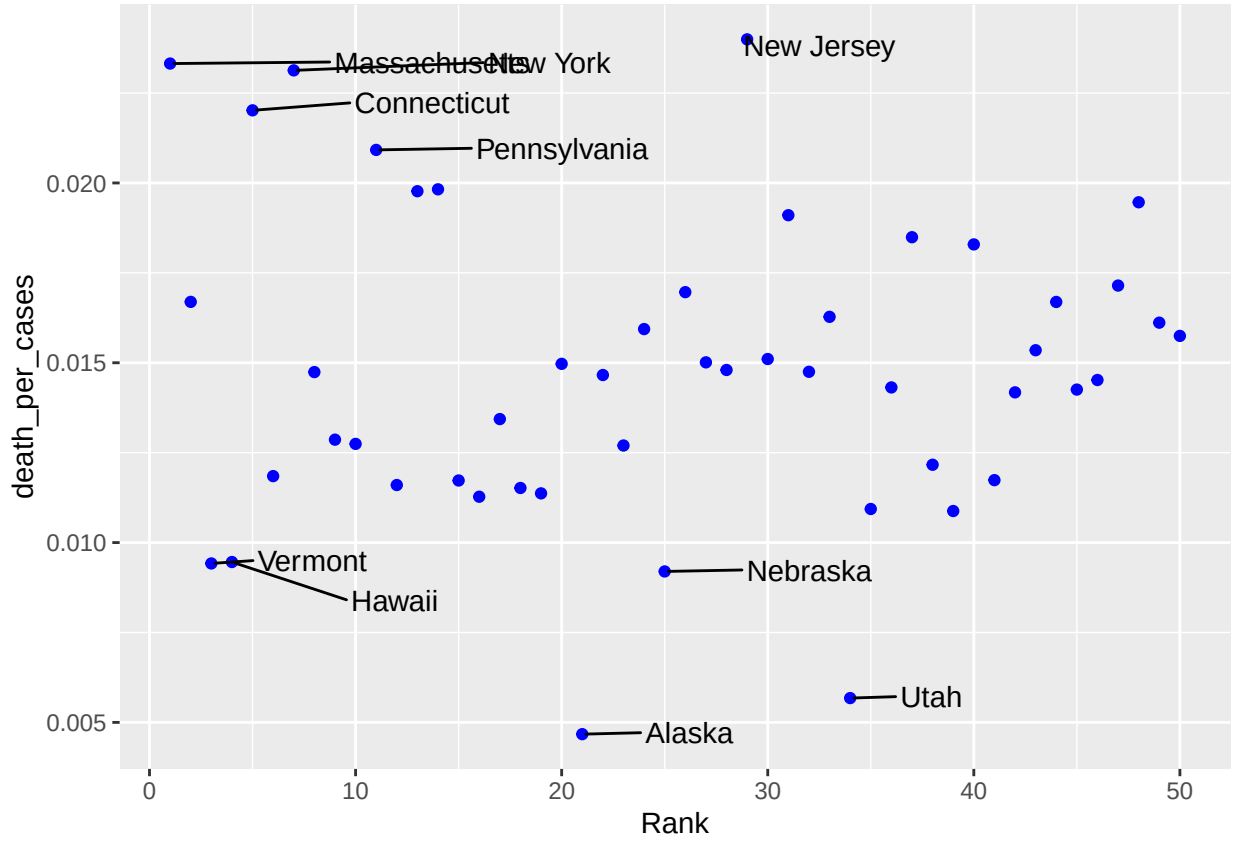
```
## Joining, by = "Province_State"
```

```
ggplot(data = US_tot_w_pred1) + geom_point(aes(x = Rank, y = death_per_cases), color = "blue") +
  geom_text_repel(aes(x =  Rank, y = death_per_cases,
                      label = ifelse(death_per_cases > quantile(death_per_cases, 0.9) ,
                                     Province_State,'')),vjust = 1.5,hjust = 0)+
  geom_text_repel(aes(x =  Rank, y = death_per_cases,
                      label = ifelse(death_per_cases < 0.01 ,Province_State,'')),vjust = 1.5)
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```

```
## Warning: Removed 5 rows containing missing values (geom_text_repel).
```

```
## Warning: Removed 5 rows containing missing values (geom_text_repel).
```

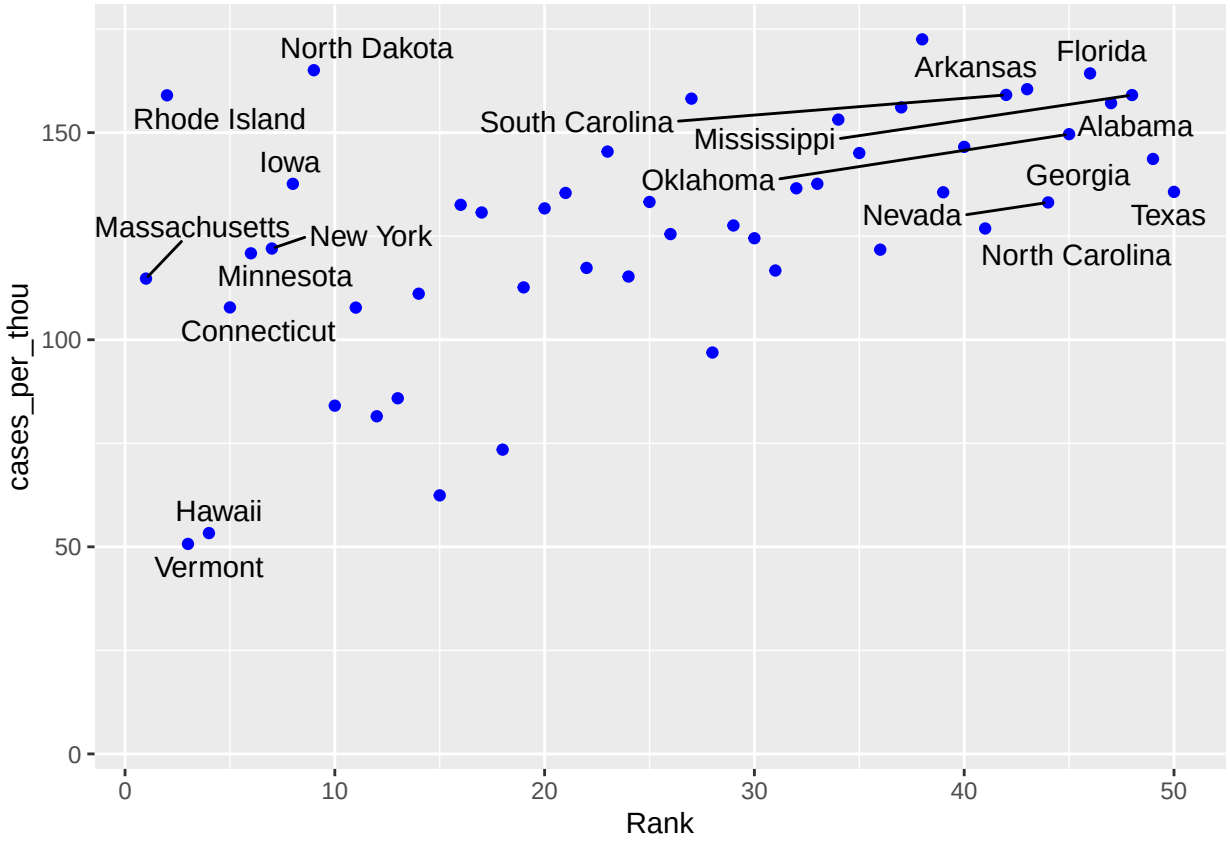



```
ggplot(data = US_tot_w_pred1) + geom_point(aes(x = Rank, y = cases_per_thou), color = "blue") +
  geom_text_repel(aes(x =  Rank, y = cases_per_thou,label = ifelse(Rank > 40 ,Province_State,'')),
                  vjust = 0.5,hjust = 0) +
  geom_text_repel(aes(x =  Rank, y = cases_per_thou,
                      label = ifelse(Rank <10 ,Province_State,'')),vjust = 0.5,hjust = 0)
```

```
## Warning: Removed 5 rows containing missing values (geom_point).
```

```
## Warning: Removed 5 rows containing missing values (geom_text_repel).
```

```
## Warning: Removed 5 rows containing missing values (geom_text_repel).
```



From the above graph it is clear that Air quality of vermont and Hawaii is good but air quality does not have anything to do in terms of deaths.

Model the data

```
mod2 <- lm(death_per_cases ~ Rank, data = US_tot_w_pred1)
summary(mod2)
```

```
##
## Call:
## lm(formula = death_per_cases ~ Rank, data = US_tot_w_pred1)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -0.0101837 -0.0030663 -0.0001165  0.0020843  0.0091747
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.496e-02  1.212e-03  12.337   <2e-16 ***
## Rank        -4.764e-06  4.138e-05  -0.115    0.909
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.004222 on 48 degrees of freedom
```

```
##   (5 observations deleted due to missingness)
## Multiple R-squared:  0.0002761,  Adjusted R-squared:  -0.02055
## F-statistic: 0.01326 on 1 and 48 DF,  p-value: 0.9088
```

Conclusions

- Nebraska, Utah and Alaska have handled the pandemic in a better way.
- New jersey, New york and Massachusetts are the worst states to handle the pandemic.
- Air quality of Vermont and Hawaii is the best.
- Air quality doesn't seem to have anything to do with covid deaths. This might be becuase air quality does not impact a person's health when that person is in quarantine.
- All the worst states are in the North east part of the country where air traffic is significantly higher than states located in the center of country.

Biases

1. I considered all the deaths reported during covid were because of covid-19 alone.
2. I did not consider pre existing conditions.
3. I did not consider the lifestyle of residents in each state.

```
sessionInfo()
```

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19042)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] ggrepel_0.9.1    lubridate_1.7.10 forcats_0.5.1    stringr_1.4.0
##  [5] dplyr_1.0.7      purrr_0.3.4      readr_2.0.1      tidyr_1.1.3
##  [9] tibble_3.1.3     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_0.6.27     utf8_1.2.2
##  [5] R6_2.5.1          cellranger_1.1.0  backports_1.2.1   reprex_2.0.1
##  [9] evaluate_0.14     httr_1.4.2        highr_0.9         pillar_1.6.2
## [13] rlang_0.4.11      curl_4.3.2        readxl_1.3.1      rstudioapi_0.13
## [17] rmarkdown_2.10    labeling_0.4.2    bit_4.0.4         munsell_0.5.0
## [21] broom_0.7.9       compiler_4.1.1    modelr_0.1.8      xfun_0.25
## [25] pkgconfig_2.0.3   htmltools_0.5.1.1 tidyselect_1.1.1  fansi_0.5.0
## [29] crayon_1.4.1      tzdb_0.1.2        dbplyr_2.1.1      withr_2.4.2
```

```
## [33] grid_4.1.1        jsonlite_1.7.2    gtable_0.3.0      lifecycle_1.0.0
## [37] DBI_1.1.1         magrittr_2.0.1    scales_1.1.1      cli_3.0.1
## [41] stringi_1.7.3     vroom_1.5.4       farver_2.1.0      fs_1.5.0
## [45] xml2_1.3.2        ellipsis_0.3.2    generics_0.1.0    vctrs_0.3.8
## [49] tools_4.1.1       bit64_4.0.5       glue_1.4.2        hms_1.1.0
## [53] parallel_4.1.1    yaml_2.2.1        colorspace_2.0-2  rvest_1.0.1
## [57] knitr_1.33        haven_2.4.3
```