COVID_finals

DS Student

5/7/2022

Finals assignment

The instructions for the assignment are to *"Import, tidy and analyze the COVID19 data set from the Johns Hopkins github site. This is the same data set I used in class. Feel free to repeat and reuse what I did if you want to. Be sure your project is reproducible and contains some visualization and analysis that is unique to your project. You may use the data to do any analysis that is of interest to you. You should include at least two visualizations and one model. Be sure to identify any bias possible in the data and in your analysis."*

I followed the step by step process delineated in the lecture videos for week 3 to import, wrangle, analyse and visualize the data. I wrote the code in an R Markdown documents and made sure to document each step clearly so that the analysis will be reproducible. I also included the model mentioned in the lecture. Intermingled with this process I included additional analyses, visualizations, and models.

Import and read data sets

```
# import and read files

url_in <- "https://github.com/CSSEGISandData/COVID-19/raw/master/csse_covid_19_data/csse_covid_19_time_s

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

urls <- str_c(url_in, file_names)

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

Explore and tidy data sets for the US

Explore `us_cases`

```
head(us_cases, 10)
```

```
## # A tibble: 10 x 854
##          UID iso2  iso3  code3  FIPS Admin2   Province_State Country_Region   Lat
##        <dbl> <chr> <chr> <dbl> <dbl> <chr>    <chr>          <chr>          <dbl>
##  1 84001001 US    USA     840  1001 Autauga  Alabama        US              32.5
##  2 84001003 US    USA     840  1003 Baldwin  Alabama        US              30.7
##  3 84001005 US    USA     840  1005 Barbour  Alabama        US              31.9
##  4 84001007 US    USA     840  1007 Bibb     Alabama        US              33.0
##  5 84001009 US    USA     840  1009 Blount   Alabama        US              34.0
##  6 84001011 US    USA     840  1011 Bullock  Alabama        US              32.1
##  7 84001013 US    USA     840  1013 Butler   Alabama        US              31.8
##  8 84001015 US    USA     840  1015 Calhoun  Alabama        US              33.8
##  9 84001017 US    USA     840  1017 Chambers Alabama        US              32.9
## 10 84001019 US    USA     840  1019 Cherokee Alabama        US              34.2
## # ... with 845 more variables: Long_ <dbl>, Combined_Key <chr>, 1/22/20 <dbl>,
## #   1/23/20 <dbl>, 1/24/20 <dbl>, 1/25/20 <dbl>, 1/26/20 <dbl>, 1/27/20 <dbl>,
## #   1/28/20 <dbl>, 1/29/20 <dbl>, 1/30/20 <dbl>, 1/31/20 <dbl>, 2/1/20 <dbl>,
## #   2/2/20 <dbl>, 2/3/20 <dbl>, 2/4/20 <dbl>, 2/5/20 <dbl>, 2/6/20 <dbl>,
## #   2/7/20 <dbl>, 2/8/20 <dbl>, 2/9/20 <dbl>, 2/10/20 <dbl>, 2/11/20 <dbl>,
## #   2/12/20 <dbl>, 2/13/20 <dbl>, 2/14/20 <dbl>, 2/15/20 <dbl>, 2/16/20 <dbl>,
## #   2/17/20 <dbl>, 2/18/20 <dbl>, 2/19/20 <dbl>, 2/20/20 <dbl>, ...
```

Tidy `us_cases`

```
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_))


head(us_cases, 10)
```

```
## # A tibble: 10 x 6
##    Admin2  Province_State Country_Region Combined_Key          date       cases
##    <chr>   <chr>          <chr>          <chr>                 <date>     <dbl>
##  1 Autauga Alabama        US             Autauga, Alabama, US 2020-01-22     0
##  2 Autauga Alabama        US             Autauga, Alabama, US 2020-01-23     0
##  3 Autauga Alabama        US             Autauga, Alabama, US 2020-01-24     0
##  4 Autauga Alabama        US             Autauga, Alabama, US 2020-01-25     0
##  5 Autauga Alabama        US             Autauga, Alabama, US 2020-01-26     0
##  6 Autauga Alabama        US             Autauga, Alabama, US 2020-01-27     0
##  7 Autauga Alabama        US             Autauga, Alabama, US 2020-01-28     0
##  8 Autauga Alabama        US             Autauga, Alabama, US 2020-01-29     0
##  9 Autauga Alabama        US             Autauga, Alabama, US 2020-01-30     0
## 10 Autauga Alabama        US             Autauga, Alabama, US 2020-01-31     0
```

Explore `us_deaths`

```
head(us_deaths)
```

```
## # A tibble: 6 x 855
##        UID iso2  iso3  code3  FIPS Admin2  Province_State Country_Region   Lat
##      <dbl> <chr> <chr> <dbl> <dbl> <chr>   <chr>          <chr>          <dbl>
## 1 84001001 US    USA     840  1001 Autauga Alabama        US              32.5
## 2 84001003 US    USA     840  1003 Baldwin Alabama        US              30.7
## 3 84001005 US    USA     840  1005 Barbour Alabama        US              31.9
## 4 84001007 US    USA     840  1007 Bibb    Alabama        US              33.0
## 5 84001009 US    USA     840  1009 Blount  Alabama        US              34.0
## 6 84001011 US    USA     840  1011 Bullock Alabama        US              32.1
## # ... with 846 more variables: Long_ <dbl>, Combined_Key <chr>,
## #   Population <dbl>, 1/22/20 <dbl>, 1/23/20 <dbl>, 1/24/20 <dbl>,
## #   1/25/20 <dbl>, 1/26/20 <dbl>, 1/27/20 <dbl>, 1/28/20 <dbl>, 1/29/20 <dbl>,
## #   1/30/20 <dbl>, 1/31/20 <dbl>, 2/1/20 <dbl>, 2/2/20 <dbl>, 2/3/20 <dbl>,
## #   2/4/20 <dbl>, 2/5/20 <dbl>, 2/6/20 <dbl>, 2/7/20 <dbl>, 2/8/20 <dbl>,
## #   2/9/20 <dbl>, 2/10/20 <dbl>, 2/11/20 <dbl>, 2/12/20 <dbl>, 2/13/20 <dbl>,
## #   2/14/20 <dbl>, 2/15/20 <dbl>, 2/16/20 <dbl>, 2/17/20 <dbl>, ...
```

Tidy us_deaths

```
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_))

head(us_deaths, 10)
```

```
## # A tibble: 10 x 7
##    Admin2  Province_State Country_Region Combined_Key      Population date
##    <chr>   <chr>          <chr>          <chr>                  <dbl> <date>
##  1 Autauga Alabama        US             Autauga, Alabama~      55869 2020-01-22
##  2 Autauga Alabama        US             Autauga, Alabama~      55869 2020-01-23
##  3 Autauga Alabama        US             Autauga, Alabama~      55869 2020-01-24
##  4 Autauga Alabama        US             Autauga, Alabama~      55869 2020-01-25
##  5 Autauga Alabama        US             Autauga, Alabama~      55869 2020-01-26
##  6 Autauga Alabama        US             Autauga, Alabama~      55869 2020-01-27
##  7 Autauga Alabama        US             Autauga, Alabama~      55869 2020-01-28
##  8 Autauga Alabama        US             Autauga, Alabama~      55869 2020-01-29
##  9 Autauga Alabama        US             Autauga, Alabama~      55869 2020-01-30
## 10 Autauga Alabama        US             Autauga, Alabama~      55869 2020-01-31
## # ... with 1 more variable: deaths <dbl>
```

Join the US data sets

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

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

```
head(us, 10)
```

```
## # A tibble: 10 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
## # ... with 1 more variable: deaths <dbl>
```

Evaluate the `summary` for potential problems.

```
summary(us)
```

```
##     Admin2          Province_State     Country_Region     Combined_Key
##  Length:2817306     Length:2817306     Length:2817306     Length:2817306
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##       date                cases           Population           deaths
##  Min.   :2020-01-22   Min.   :  -3073   Min.   :       0   Min.   :  -82.0
##  1st Qu.:2020-08-19   1st Qu.:    118   1st Qu.:    9917   1st Qu.:    1.0
##  Median :2021-03-18   Median :   1295   Median :   24892   Median :   22.0
##  Mean   :2021-03-18   Mean   :   8931   Mean   :   99604   Mean   :  140.2
##  3rd Qu.:2021-10-15   3rd Qu.:   4980   3rd Qu.:   64979   3rd Qu.:   85.0
##  Max.   :2022-05-13   Max.   :2907721   Max.   :10039107   Max.   :32022.0
```

If minimum number of `cases` or `deaths` is a negative number, `filter` rows where either the `cases` or `deaths` variable is entered as less than 0.

```
#filter data entries where the number of cases or deaths is a negative number
us <- us %>%
  filter(cases > -1) %>%
  filter(deaths > -1)


summary(us)
```

```
##     Admin2          Province_State     Country_Region     Combined_Key
##  Length:2817304     Length:2817304     Length:2817304     Length:2817304
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##       date                cases           Population           deaths
##  Min.   :2020-01-22   Min.   :      0   Min.   :       0   Min.   :    0.0
##  1st Qu.:2020-08-19   1st Qu.:    118   1st Qu.:    9917   1st Qu.:    1.0
##  Median :2021-03-18   Median :   1295   Median :   24909   Median :   22.0
##  Mean   :2021-03-17   Mean   :   8931   Mean   :   99604   Mean   :  140.2
##  3rd Qu.:2021-10-15   3rd Qu.:   4980   3rd Qu.:   64979   3rd Qu.:   85.0
##  Max.   :2022-05-13   Max.   :2907721   Max.   :10039107   Max.   :32022.0
```

Explore and tidy the global data sets

Explore `global_cases`

```
head(global_cases)
```

```
## # A tibble: 6 x 847
##   `Province/State` `Country/Region`   Lat  Long `1/22/20` `1/23/20` `1/24/20`
##   <chr>            <chr>            <dbl> <dbl>     <dbl>     <dbl>     <dbl>
## 1 <NA>             Afghanistan       33.9 67.7          0         0         0
## 2 <NA>             Albania           41.2 20.2          0         0         0
## 3 <NA>             Algeria           28.0  1.66         0         0         0
## 4 <NA>             Andorra           42.5  1.52         0         0         0
## 5 <NA>             Angola           -11.2 17.9          0         0         0
## 6 <NA>             Antarctica       -71.9 23.3          0         0         0
## # ... with 840 more variables: 1/25/20 <dbl>, 1/26/20 <dbl>, 1/27/20 <dbl>,
## #   1/28/20 <dbl>, 1/29/20 <dbl>, 1/30/20 <dbl>, 1/31/20 <dbl>, 2/1/20 <dbl>,
## #   2/2/20 <dbl>, 2/3/20 <dbl>, 2/4/20 <dbl>, 2/5/20 <dbl>, 2/6/20 <dbl>,
## #   2/7/20 <dbl>, 2/8/20 <dbl>, 2/9/20 <dbl>, 2/10/20 <dbl>, 2/11/20 <dbl>,
## #   2/12/20 <dbl>, 2/13/20 <dbl>, 2/14/20 <dbl>, 2/15/20 <dbl>, 2/16/20 <dbl>,
## #   2/17/20 <dbl>, 2/18/20 <dbl>, 2/19/20 <dbl>, 2/20/20 <dbl>, 2/21/20 <dbl>,
## #   2/22/20 <dbl>, 2/23/20 <dbl>, 2/24/20 <dbl>, 2/25/20 <dbl>, ...
```

Tidy `global_cases`

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

head(global_cases)
```

```
## # A tibble: 6 x 4
##   `Province/State` `Country/Region` date    cases
##   <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
```

Explore `global_deaths`

```
head(global_deaths)
```

```
## # A tibble: 6 x 847
##   ‘Province/State‘ ‘Country/Region‘   Lat  Long ‘1/22/20‘ ‘1/23/20‘ ‘1/24/20‘
##   <chr>            <chr>            <dbl> <dbl>     <dbl>     <dbl>     <dbl>
## 1 <NA>             Afghanistan       33.9 67.7          0         0         0
## 2 <NA>             Albania           41.2 20.2          0         0         0
## 3 <NA>             Algeria           28.0  1.66         0         0         0
## 4 <NA>             Andorra           42.5  1.52         0         0         0
## 5 <NA>             Angola           -11.2 17.9          0         0         0
## 6 <NA>             Antarctica       -71.9 23.3          0         0         0
## # ... with 840 more variables: 1/25/20 <dbl>, 1/26/20 <dbl>, 1/27/20 <dbl>,
## #   1/28/20 <dbl>, 1/29/20 <dbl>, 1/30/20 <dbl>, 1/31/20 <dbl>, 2/1/20 <dbl>,
## #   2/2/20 <dbl>, 2/3/20 <dbl>, 2/4/20 <dbl>, 2/5/20 <dbl>, 2/6/20 <dbl>,
## #   2/7/20 <dbl>, 2/8/20 <dbl>, 2/9/20 <dbl>, 2/10/20 <dbl>, 2/11/20 <dbl>,
## #   2/12/20 <dbl>, 2/13/20 <dbl>, 2/14/20 <dbl>, 2/15/20 <dbl>, 2/16/20 <dbl>,
## #   2/17/20 <dbl>, 2/18/20 <dbl>, 2/19/20 <dbl>, 2/20/20 <dbl>, 2/21/20 <dbl>,
## #   2/22/20 <dbl>, 2/23/20 <dbl>, 2/24/20 <dbl>, 2/25/20 <dbl>, ...
```

Tidy `global_deaths`

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

head(global_cases, 10)
```

```
## # A tibble: 10 x 4
##    ‘Province/State‘ ‘Country/Region‘ date    cases
##    <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
```

Join the global data sets

`full_join` the two global data sets and `rename`two of the columns. `mutate` date to a date object.

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

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

```
head(global, 10)
```

```
## # A tibble: 10 x 5
##    Province_State Country_Region date       cases deaths
##    <chr>          <chr>          <date>     <dbl>  <dbl>
##  1 <NA>           Afghanistan    2020-01-22     0      0
##  2 <NA>           Afghanistan    2020-01-23     0      0
##  3 <NA>           Afghanistan    2020-01-24     0      0
##  4 <NA>           Afghanistan    2020-01-25     0      0
##  5 <NA>           Afghanistan    2020-01-26     0      0
##  6 <NA>           Afghanistan    2020-01-27     0      0
##  7 <NA>           Afghanistan    2020-01-28     0      0
##  8 <NA>           Afghanistan    2020-01-29     0      0
##  9 <NA>           Afghanistan    2020-01-30     0      0
## 10 <NA>           Afghanistan    2020-01-31     0      0
```

Evaluate the `summary` for potential problems.

```
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:239412      Length:239412      Min.   :2020-01-22   Min.   :       0
##  Class :character   Class :character   1st Qu.:2020-08-19   1st Qu.:     284
##  Mode  :character   Mode  :character   Median :2021-03-18   Median :    6340
##                                        Mean   :2021-03-18   Mean   :  554666
##                                        3rd Qu.:2021-10-15   3rd Qu.:  121880
##                                        Max.   :2022-05-13   Max.   :82421624
##      deaths
##  Min.   :     0
##  1st Qu.:     2
##  Median :    82
##  Mean   : 10151
##  3rd Qu.:  1900
##  Max.   :999518
```

The minimum number of cases is zero. `filter` out rows with no cases.

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

summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:220713      Length:220713      Min.   :2020-01-22   Min.   :       1
##  Class :character   Class :character   1st Qu.:2020-09-24   1st Qu.:     693
##  Mode  :character   Mode  :character   Median :2021-04-14   Median :   10127
##                                        Mean   :2021-04-10   Mean   :  601658
##                                        3rd Qu.:2021-10-29   3rd Qu.:  149729
##                                        Max.   :2022-05-13   Max.   :82421624
##      deaths
##  Min.   :     0
##  1st Qu.:     5
##  Median :   125
##  Mean   : 11011
##  3rd Qu.:  2399
##  Max.   :999518
```

Now the minimum number of cases in any row is one.

Check to ensure the Max in each of the `cases` and `deaths` columns is not a typographical error.

```
global %>% filter(cases > 80000000)
```

```
## # A tibble: 47 x 5
##    Province_State Country_Region date          cases deaths
##    <chr>          <chr>          <date>        <dbl>  <dbl>
##  1 <NA>           US             2022-03-28 80001286 978260
##  2 <NA>           US             2022-03-29 80025464 979243
##  3 <NA>           US             2022-03-30 80064646 980411
##  4 <NA>           US             2022-03-31 80110284 980927
##  5 <NA>           US             2022-04-01 80142499 981756
##  6 <NA>           US             2022-04-02 80154308 981912
##  7 <NA>           US             2022-04-03 80159063 981970
##  8 <NA>           US             2022-04-04 80187971 982557
##  9 <NA>           US             2022-04-05 80213572 983070
## 10 <NA>           US             2022-04-06 80254519 984195
## # ... with 37 more rows
```

Check the `deaths` column.

```
global %>% filter(deaths > 990000)
```

```
## # A tibble: 24 x 5
##    Province_State Country_Region date          cases deaths
##    <chr>          <chr>          <date>        <dbl>  <dbl>
##  1 <NA>           US             2022-04-20 80804471 990330
##  2 <NA>           US             2022-04-21 80864015 990713
##  3 <NA>           US             2022-04-22 80954071 991269
##  4 <NA>           US             2022-04-23 80973785 991320
##  5 <NA>           US             2022-04-24 80987251 991349
##  6 <NA>           US             2022-04-25 81057431 991634
##  7 <NA>           US             2022-04-26 81102716 991958
##  8 <NA>           US             2022-04-27 81192697 992759
##  9 <NA>           US             2022-04-28 81267145 993126
## 10 <NA>           US             2022-04-29 81328234 993553
## # ... with 14 more rows
```

Add population data to global data set and a Combined_Key variable

`Combined_Key` variable combines `Province_State` and `Country_Region` into one variable.

```
global <- global %>%
  unite("Combined_Key",
        c(Province_State, Country_Region),
        # combines the column names with a space
        sep = ", ",
        na.rm = TRUE,
        remove = FALSE)
```

Add `Population` from a csv file at the same Johns Hopkins website

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

Read the csv file in and look at the column names

```
uid <- read_csv(uid_lookup_url, show_col_types = FALSE)

colnames(uid)
```

```
##  [1] "UID"            "iso2"           "iso3"           "code3"
##  [5] "FIPS"           "Admin2"         "Province_State" "Country_Region"
##  [9] "Lat"            "Long_"          "Combined_Key"   "Population"
```

`select` desired columns

```
uid <- uid %>%
  select(-c(Lat, Long_, Combined_Key, code3, iso2, iso3, Admin2))

head(uid, 10)
```

```
## # A tibble: 10 x 5
##      UID FIPS  Province_State Country_Region      Population
##    <dbl> <chr> <chr>          <chr>                    <dbl>
##  1     4 <NA>  <NA>           Afghanistan           38928341
##  2     8 <NA>  <NA>           Albania                2877800
##  3    10 <NA>  <NA>           Antarctica                  NA
##  4    12 <NA>  <NA>           Algeria               43851043
##  5    20 <NA>  <NA>           Andorra                  77265
##  6    24 <NA>  <NA>           Angola                32866268
##  7    28 <NA>  <NA>           Antigua and Barbuda      97928
##  8    32 <NA>  <NA>           Argentina             45195777
##  9    51 <NA>  <NA>           Armenia                2963234
## 10    40 <NA>  <NA>           Austria                9006400
```

`left_join` the global population data set to the global COVID data set. Reorder the new data set columns (use `select`).

```
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)

head(global, 10)
```

```
## # A tibble: 10 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
```

Although the Province_State column may be empty for many countries, it is reported for some. We can explore this by printing out the `unique` `Province_State` values in the `global` data set.

```
head(unique(global$Province_State), 15)
```

```
##  [1] NA                             "Australian Capital Territory"
##  [3] "New South Wales"              "Northern Territory"
##  [5] "Queensland"                   "South Australia"
##  [7] "Tasmania"                     "Victoria"
##  [9] "Western Australia"            "Alberta"
## [11] "British Columbia"             "Diamond Princess"
## [13] "Grand Princess"               "Manitoba"
## [15] "New Brunswick"
```

Analyse the global data

To get the number of daily cases and deaths by country we need to `group_by` `Province_State`, `Country_Region`, and `date` and then `sum` the `cases`, `deaths`, and `Population` for each country.`Calculate death rate as deaths per million and add as a column (using`mutate`).`select`the column names to include.`ungroup‘ the data set.

```
global_summary <- global %>%
  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(Country_Region, date, cases, deaths, deaths_per_mill, Population)  %>%
  ungroup() %>%
  select(-Province_State)
```

```
## ‘summarise()‘ has grouped output by ’Province_State’, ’Country_Region’. You can override using the ‘
## Adding missing grouping variables: ‘Province_State‘
```

```
tail(global_summary, 10)
```

```
## # A tibble: 10 x 6
##    Country_Region date        cases deaths deaths_per_mill Population
##    <chr>          <date>      <dbl>  <dbl>           <dbl>      <dbl>
##  1 Zimbabwe       2022-05-04 248050   5471            368.   14862927
##  2 Zimbabwe       2022-05-05 248050   5471            368.   14862927
##  3 Zimbabwe       2022-05-06 248214   5473            368.   14862927
##  4 Zimbabwe       2022-05-07 248214   5473            368.   14862927
##  5 Zimbabwe       2022-05-08 248352   5476            368.   14862927
##  6 Zimbabwe       2022-05-09 248536   5479            369.   14862927
##  7 Zimbabwe       2022-05-10 248642   5481            369.   14862927
##  8 Zimbabwe       2022-05-11 248642   5481            369.   14862927
##  9 Zimbabwe       2022-05-12 248943   5481            369.   14862927
## 10 Zimbabwe       2022-05-13 249131   5482            369.   14862927
```

Look at the data for France

```r
France <- global_summary %>%
  filter(cases > 0 & Country_Region == 'France')

tail(France, 10)
```

```
## # A tibble: 10 x 6
##    Country_Region date          cases deaths deaths_per_mill Population
##    <chr>          <date>        <dbl>  <dbl>           <dbl>      <dbl>
##  1 France         2022-05-04 28026502 143176           2194.   65249843
##  2 France         2022-05-05 28070727 143298           2196.   65249843
##  3 France         2022-05-06 28108929 143408           2198.   65249843
##  4 France         2022-05-07 28146887 143470           2199.   65249843
##  5 France         2022-05-08 28176211 143524           2200.   65249843
##  6 France         2022-05-09 28181004 143655           2202.   65249843
##  7 France         2022-05-10 28229588 143772           2203.   65249843
##  8 France         2022-05-11 28269887 143868           2205.   65249843
##  9 France         2022-05-12 28305934 143952           2206.   65249843
## 10 France         2022-05-13 28334484 144048           2208.   65249843
```

Analyze the US data

Analyze data for a state - group by state Look at the column names for the US data set again

```r
colnames(us)
```

```
## [1] "Admin2"         "Province_State" "Country_Region" "Combined_Key"
## [5] "date"           "cases"          "Population"     "deaths"
```

`filter` the data for WA state and look at the data set

```r
us_WA <- us %>%
  filter(Province_State == 'Washington')

head(us_WA, 10)
```

```
## # A tibble: 10 x 8
##    Admin2 Province_State Country_Region Combined_Key date       cases Population
##    <chr>  <chr>          <chr>          <chr>        <date>     <dbl>      <dbl>
##  1 Adams  Washington     US             Adams, Wash~ 2020-01-22     0      19983
##  2 Adams  Washington     US             Adams, Wash~ 2020-01-23     0      19983
##  3 Adams  Washington     US             Adams, Wash~ 2020-01-24     0      19983
##  4 Adams  Washington     US             Adams, Wash~ 2020-01-25     0      19983
##  5 Adams  Washington     US             Adams, Wash~ 2020-01-26     0      19983
##  6 Adams  Washington     US             Adams, Wash~ 2020-01-27     0      19983
##  7 Adams  Washington     US             Adams, Wash~ 2020-01-28     0      19983
##  8 Adams  Washington     US             Adams, Wash~ 2020-01-29     0      19983
##  9 Adams  Washington     US             Adams, Wash~ 2020-01-30     0      19983
## 10 Adams  Washington     US             Adams, Wash~ 2020-01-31     0      19983
## # ... with 1 more variable: deaths <dbl>
```

Each state contains multiple counties. To get the number of daily cases and deaths by state we need to `group_by` `Province_State`, `Country_Region`, and `date` and then sum the `cases`, `deaths`, and `Population` for the counties comprising each state.`Calculate death rate as deaths per million and add as a column (using`mutate`).select`the column names to include.`ungroup‘ the data set.

```
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.
```

```
tail(us_by_state, 10)
```

```
## # A tibble: 10 x 7
##    Province_State Country_Region date        cases deaths deaths_per_mill
##    <chr>          <chr>          <date>      <dbl>  <dbl>           <dbl>
##  1 Wyoming        US             2022-05-04 156745   1814           3134.
##  2 Wyoming        US             2022-05-05 156745   1814           3134.
##  3 Wyoming        US             2022-05-06 156745   1814           3134.
##  4 Wyoming        US             2022-05-07 156745   1814           3134.
##  5 Wyoming        US             2022-05-08 156745   1814           3134.
##  6 Wyoming        US             2022-05-09 156745   1814           3134.
##  7 Wyoming        US             2022-05-10 157031   1817           3139.
##  8 Wyoming        US             2022-05-11 157031   1817           3139.
##  9 Wyoming        US             2022-05-12 157031   1817           3139.
## 10 Wyoming        US             2022-05-13 157031   1817           3139.
## # ... with 1 more variable: Population <dbl>
```

Look at the data for WA state in the last 10 days.

```
us_WA <- us_by_state %>%
  filter(Province_State == 'Washington')

tail(us_WA, 10)
```

```
## # A tibble: 10 x 7
##    Province_State Country_Region date         cases deaths deaths_per_mill
##    <chr>          <chr>          <date>       <dbl>  <dbl>           <dbl>
##  1 Washington     US             2022-05-04 1508065  12716           1670.
##  2 Washington     US             2022-05-05 1508065  12716           1670.
##  3 Washington     US             2022-05-06 1508065  12716           1670.
##  4 Washington     US             2022-05-07 1508065  12716           1670.
##  5 Washington     US             2022-05-08 1508065  12716           1670.
##  6 Washington     US             2022-05-09 1519327  12742           1673.
##  7 Washington     US             2022-05-10 1519327  12742           1673.
##  8 Washington     US             2022-05-11 1524078  12770           1677.
##  9 Washington     US             2022-05-12 1524078  12770           1677.
## 10 Washington     US             2022-05-13 1530430  12791           1680.
## # ... with 1 more variable: Population <dbl>
```

Analyze data for the US - group all states together Analyze the daily `cases` and `deaths` in the entirely of the US. Look at the data for the US in the last 10 days.

```
us_total <- us %>%
  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.
```

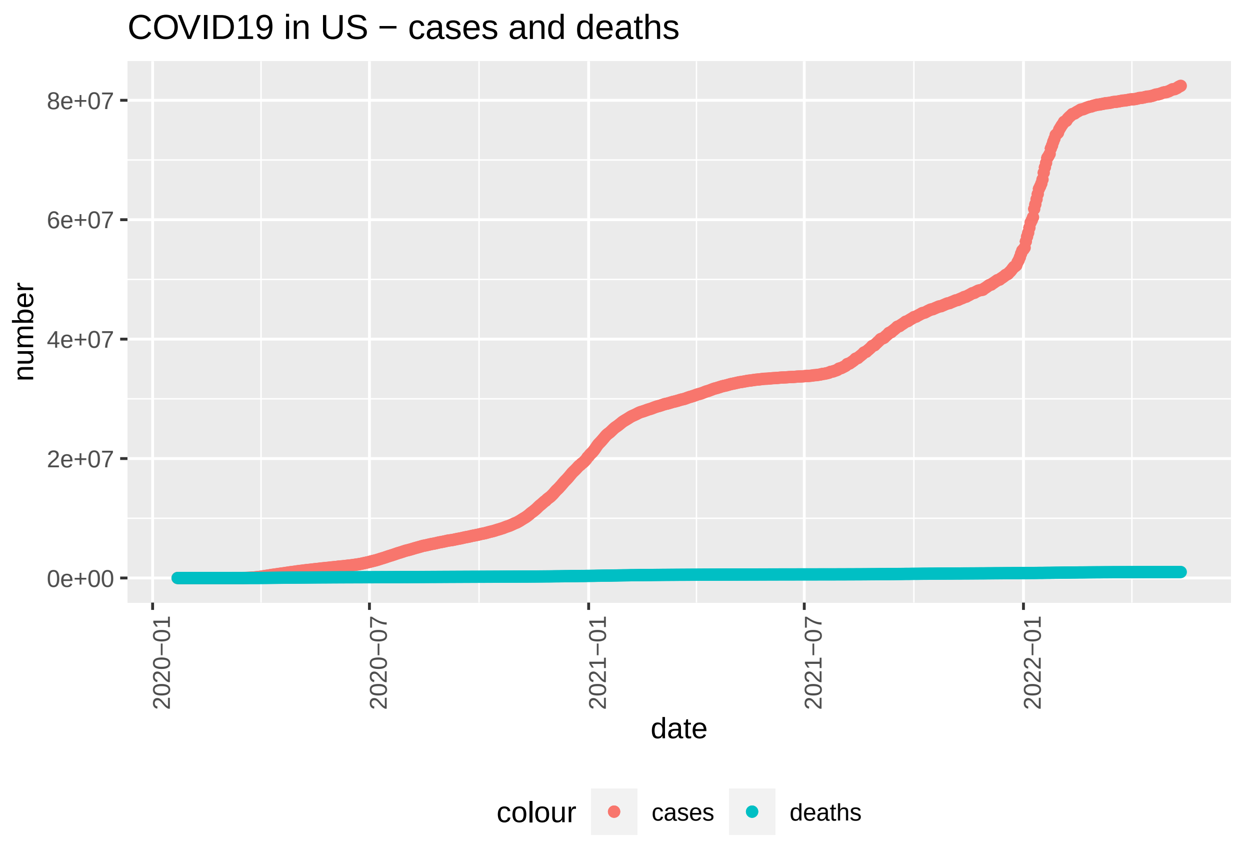
```
tail(us_total, 10)
```

```
## # A tibble: 10 x 6
##    Country_Region date          cases deaths deaths_per_mill Population
##    <chr>          <date>        <dbl>  <dbl>           <dbl>      <dbl>
##  1 US             2022-05-04 81620724 996656           2994.  332875137
##  2 US             2022-05-05 81710925 997046           2995.  332875137
##  3 US             2022-05-06 81834681 997400           2996.  332875137
##  4 US             2022-05-07 81858498 997503           2997.  332875137
##  5 US             2022-05-08 81863479 997526           2997.  332875137
##  6 US             2022-05-09 81973661 997740           2997.  332875137
##  7 US             2022-05-10 82059839 998048           2998.  332875137
##  8 US             2022-05-11 82223174 998997           3001.  332875137
##  9 US             2022-05-12 82325687 999125           3002.  332875137
## 10 US             2022-05-13 82421624 999518           3003.  332875137
```

Visualize the US data

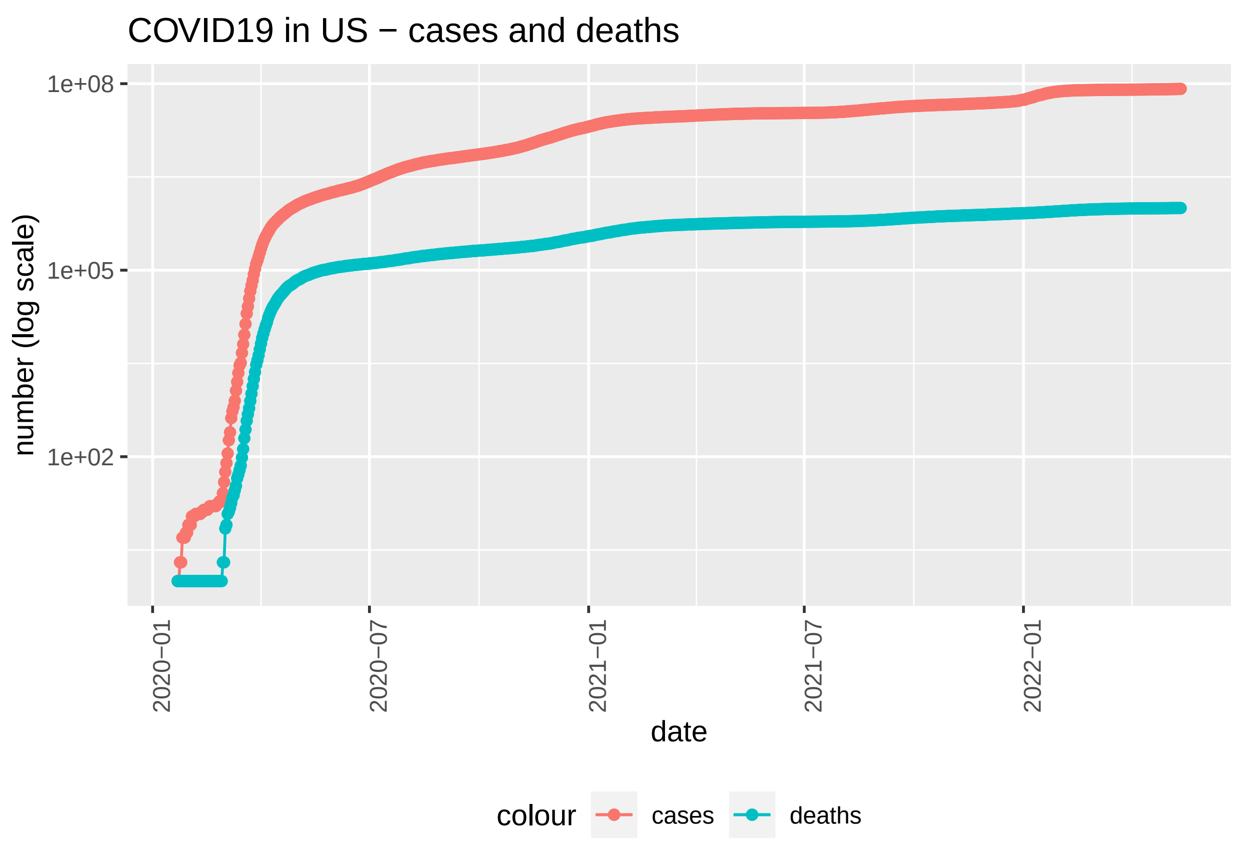
`filter` the us_total to only include dates with cases. Plot the `cases` as `geom_point`. Plot the `deaths` as `geom_point`.

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



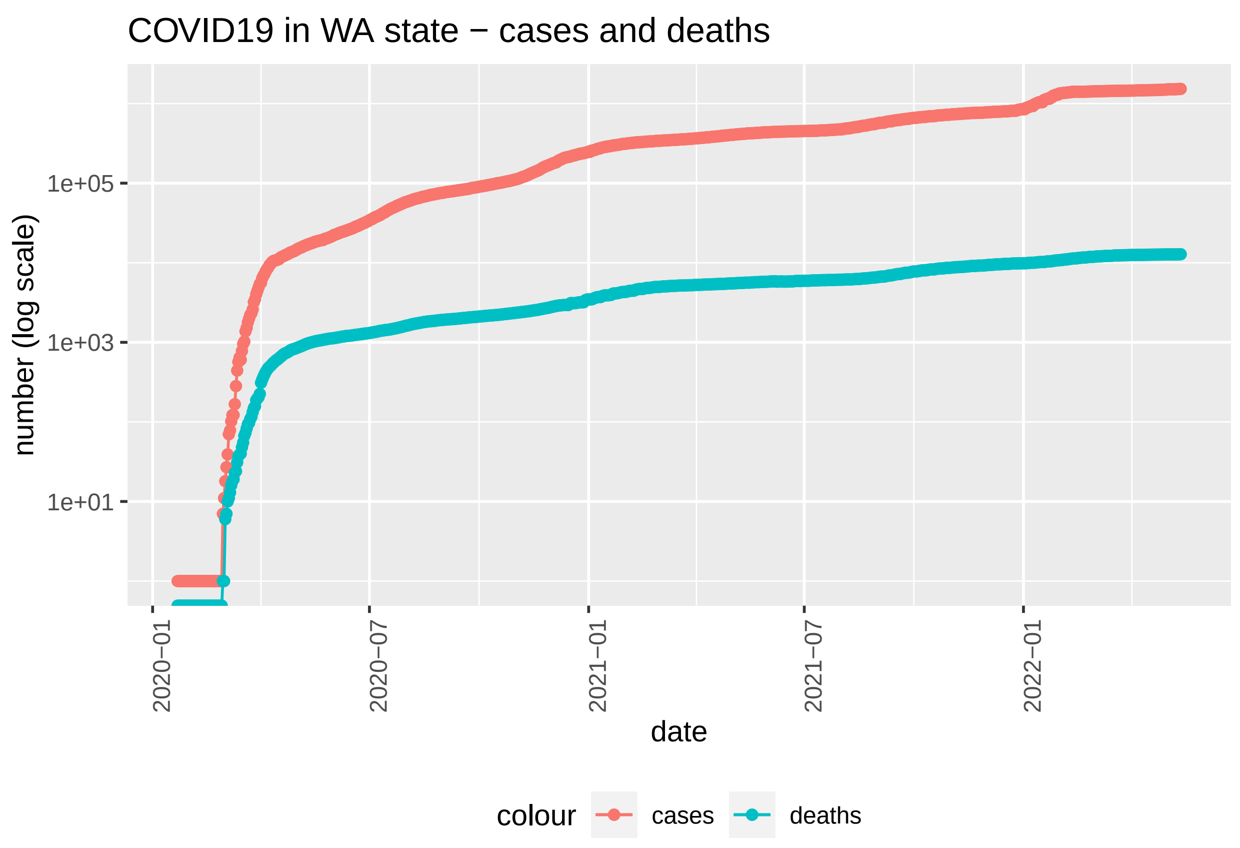
Optimize the plot by: - Plotting the `cases` and `deaths` as `geom_line` as well. - Changing the y-axis to a logarithmic scale (`scale_y_log`).

```
us_total %>%
  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 - cases and deaths", y = "number (log scale)")
```



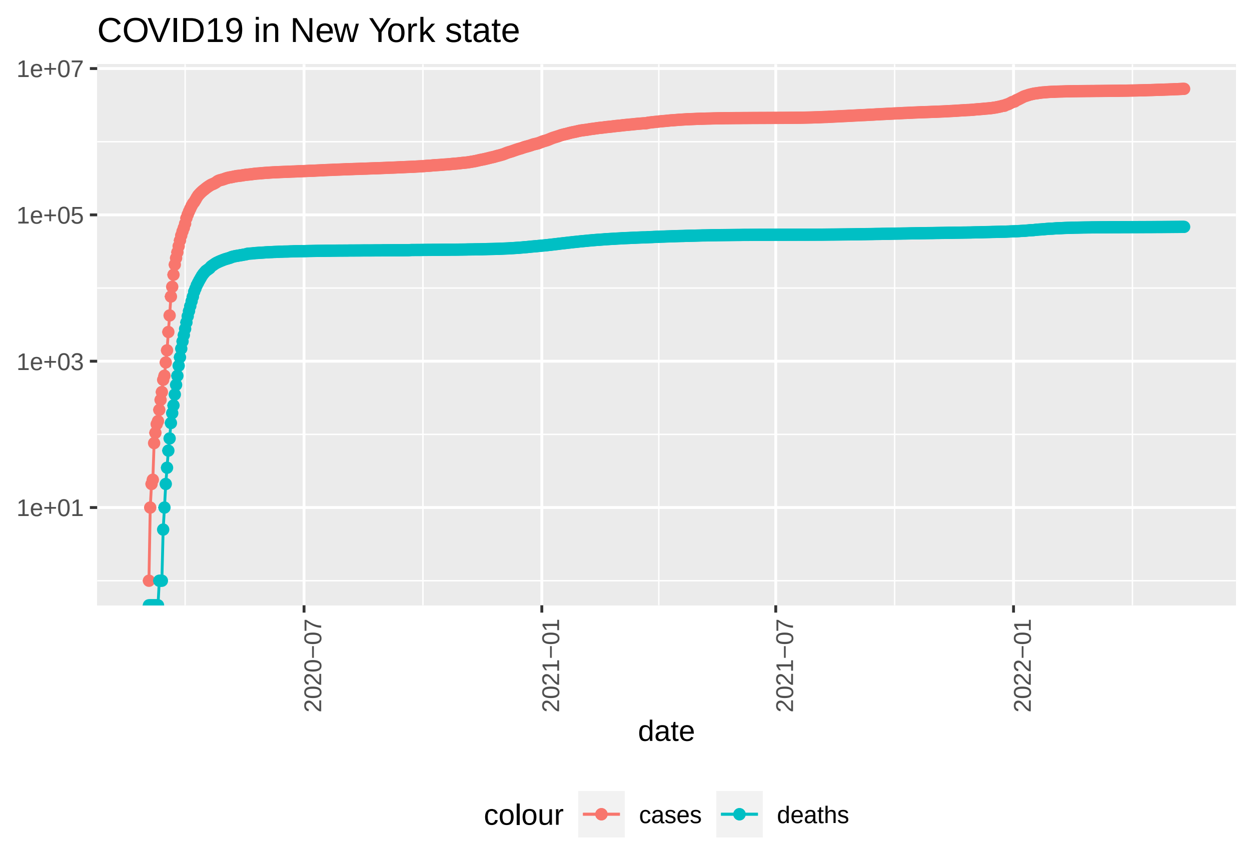
Visualize the data for *WA state*.

```
us_by_state %>%
  filter(cases > 0 & Province_State == 'Washington') %>%
  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 WA state - cases and deaths", y = "number (log scale)")
```



Visualize the data for *New York state*.

```
us_by_state %>%
  filter(cases > 0 & Province_State == 'New York') %>%
  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 New York state", y = NULL)
```



The visualizations suggest that the number of cases deaths have leveled off. Is this true?

Analyzing data

Explore the us_total data set: - current date - max total deaths - max deaths per million - max popultion

```
max(us_total$date)
```

```
## [1] "2022-05-13"
```

```
max(us_total$deaths)
```

```
## [1] 999518
```

```
max(us_total$deaths_per_mill)
```

```
## [1] 3002.681
```

```
max(us_total$Population)
```

```
## [1] 332875137
```

Explore the us_by_state data set for the same data: - current date - max total deaths - max deaths per million - max population

```r
max(us_by_state$date)
```

```
## [1] "2022-05-13"
```

```r
max(us_by_state$deaths)
```

```
## [1] 90782
```

```r
max(us_by_state$deaths_per_mill)
```

```
## [1] NaN
```

```r
max(us_by_state$Population)
```

```
## [1] 39512223
```

Transform data - add new variables

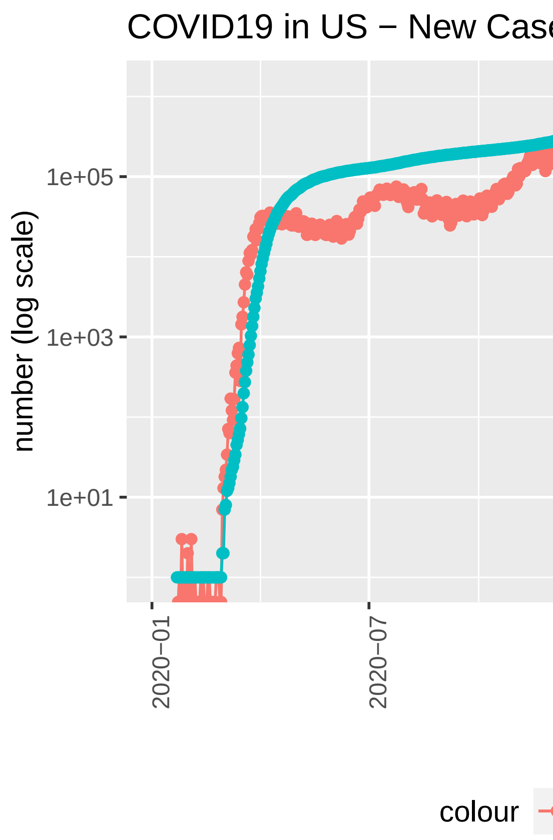
Add new variables: - `us_by_states` add: `new_cases` and `new_deaths` - `us_total` add: `new_cases` and `new_deaths`

```r
us_by_state <- us_by_state %>%
  mutate(new_cases = cases - lag(cases),
         new_deaths = deaths - lag(deaths))

us_total <- us_total %>%
  mutate(new_cases = cases - lag(cases),
         new_deaths = deaths - lag(deaths))

tail(us_total %>%
       select(new_cases, new_deaths, everything()), 10)
```

```
## # A tibble: 10 x 8
##    new_cases new_deaths Country_Region date          cases deaths deaths_per_mill
##        <dbl>      <dbl> <chr>          <date>        <dbl>  <dbl>           <dbl>
##  1    114107       1959 US             2022-05-04  8.16e7 996656           2994.
##  2     90201        390 US             2022-05-05  8.17e7 997046           2995.
##  3    123756        354 US             2022-05-06  8.18e7 997400           2996.
##  4     23817        103 US             2022-05-07  8.19e7 997503           2997.
##  5      4981         23 US             2022-05-08  8.19e7 997526           2997.
##  6    110182        214 US             2022-05-09  8.20e7 997740           2997.
##  7     86178        308 US             2022-05-10  8.21e7 998048           2998.
##  8    163335        949 US             2022-05-11  8.22e7 998997           3001.
##  9    102513        128 US             2022-05-12  8.23e7 999125           3002.
## 10     95937        393 US             2022-05-13  8.24e7 999518           3003.
## # ... with 1 more variable: Population <dbl>
```

Visualize transformed data

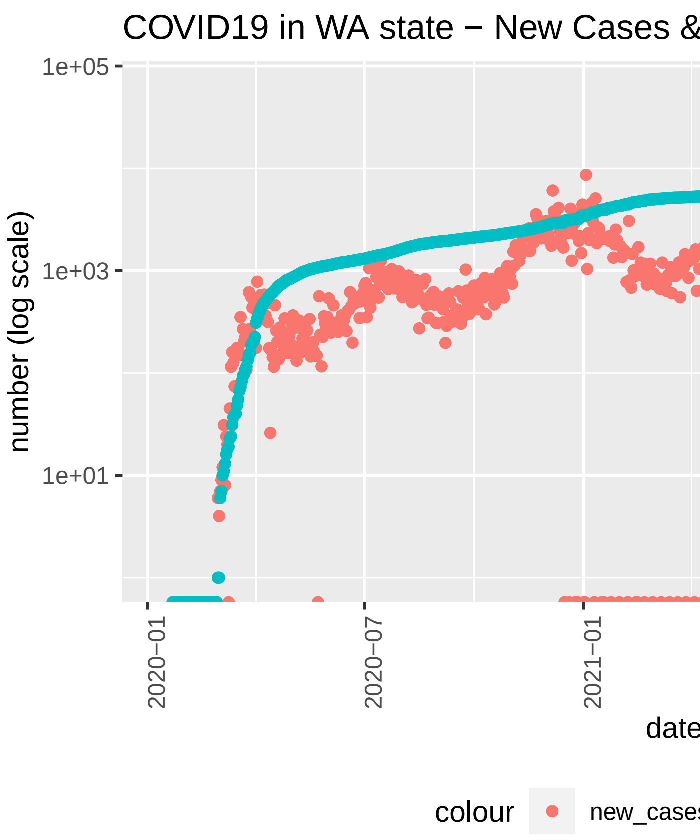
```
us_total %>%
  ggplot(aes(x = date, y = new_cases)) +
  geom_line(aes(color = "new_cases")) +
  geom_point(aes(color = "new_cases")) +
  geom_line(aes(y = deaths, color = "new_deaths")) +
  geom_point(aes(y = deaths, color = "new_deaths")) +
  scale_y_log10() +
  theme(legend.position="bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 in US - New Cases & New Deaths", y = "number (log scale)")
```



Visualize the new_cases and new_deaths in the us_total data set

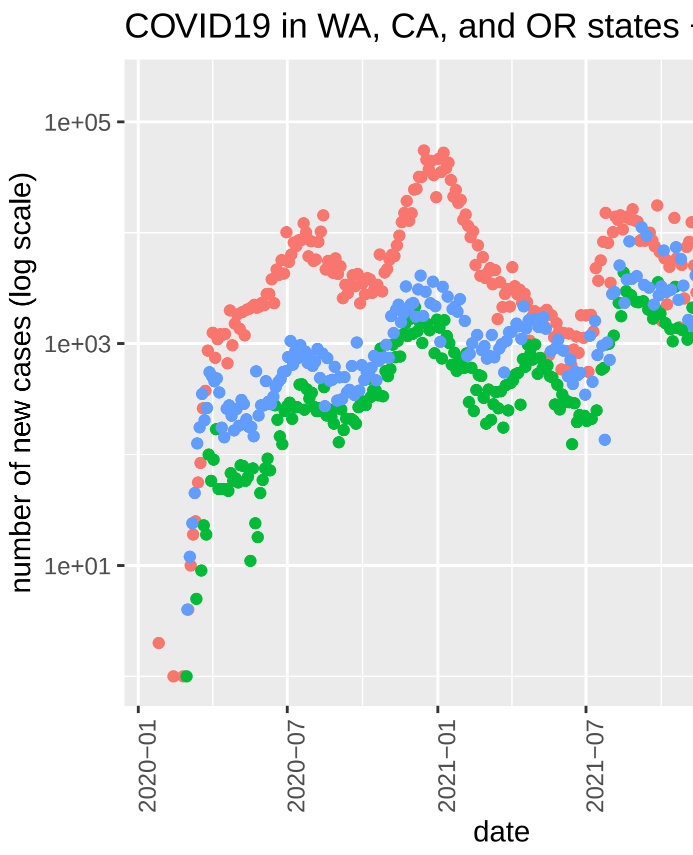
Observation: The plot demonstrates that the daily have flattened and there are fluctuations in the number of daily `new_cases`. Last peak of daily `new_cases` - the largest peak so far - was in January 2022.

```
us_by_state %>%
  filter(cases > 0 & Province_State == 'Washington') %>%
  ggplot(aes(x = date, y = new_cases)) +
  #geom_line(aes(color = "new_cases")) +
  geom_point(aes(color = "new_cases")) +
  #geom_line(aes(y = deaths, color = "new_deaths")) +
  geom_point(aes(y = deaths, color = "new_deaths")) +
  scale_y_log10() +
  theme(legend.position="bottom",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 in WA state - New Cases & New Deaths", y = "number (log scale)")
```



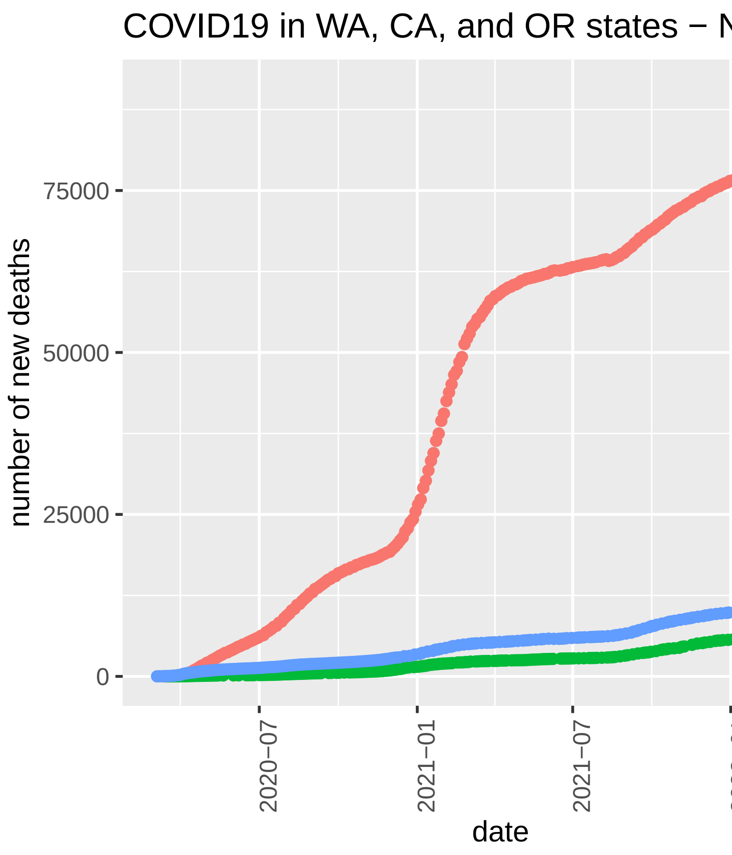
Visualize the new_cases and new_deaths in WA state

```
us_by_state %>%
  filter(new_cases > 0 & Province_State == c("Washington", "California", "Oregon") ) %>%
  ggplot(aes(x = date, y = new_cases, group = Province_State, color = factor(Province_State))) +
  geom_point() +
  scale_y_log10() +
  theme(legend.position="right",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 in WA, CA, and OR states - New Cases", y = "number of new cases (log scale)")
```



Visualize the trend of new cases in three western states

```
# number of new deaths is numeric (i.e., not a log scale)
us_by_state %>%
  filter(new_deaths > 0 & Province_State == c("Washington", "California", "Oregon") ) %>%
  ggplot(aes(x = date, y = deaths, group = Province_State, color = factor(Province_State))) +
  geom_point() +
  #scale_y_log10() +
  theme(legend.position="right",
        axis.text.x = element_text(angle = 90)) +
  labs(title = "COVID19 in WA, CA, and OR states - New Deaths", y = "number of new deaths")
```



Visualize the trend of deaths in three western states

Analyse data: What are the best and worst states?

Transform data to a new table which summarizes the `deaths`, `cases`, `cases_per_thou`, and `deaths_per_thou` in each state. `arrange` the rows in ascending number of `deaths` in each state.

```
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) %>%
  arrange(deaths)

us_state_totals
```

```
## # A tibble: 56 x 6
##    Province_State        deaths  cases population cases_per_thou deaths_per_thou
##    <chr>                  <dbl>  <dbl>      <dbl>          <dbl>           <dbl>
##  1 American Samoa            30   5999      55641           108.           0.539
##  2 Northern Mariana Isl~     34  11305      55144           205.           0.617
##  3 Virgin Islands           113  18037     107268           168.           1.05
##  4 Guam                     361  48496     164229           295.           2.20
##  5 Vermont                  648 128471     623989           206.           1.04
##  6 Alaska                  1268 256089     740995           346.           1.71
##  7 District of Columbia    1340 144675     705749           205.            1.90
##  8 Hawaii                  1434 258422    1415872           183.            1.01
##  9 Wyoming                 1817 157031     578759           271.            3.14
## 10 North Dakota            2272 242462     762062           318.            2.98
## # ... with 46 more rows
```

The 10 states with the **lowest death rates per 1000 population**.

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

```
## # A tibble: 10 x 6
##    Province_State        deaths_per_thou cases_per_thou deaths   cases population
##    <chr>                           <dbl>          <dbl>  <dbl>   <dbl>      <dbl>
##  1 American Samoa                  0.539           108.     30 6.00e3      55641
##  2 Northern Mariana Isl~           0.617           205.     34 1.13e4      55144
##  3 Hawaii                          1.01            183.   1434 2.58e5    1415872
##  4 Vermont                         1.04            206.    648 1.28e5     623989
##  5 Virgin Islands                  1.05            168.    113 1.80e4     107268
##  6 Puerto Rico                     1.13            158.   4250 5.94e5    3754939
##  7 Utah                            1.49            293.   4761 9.39e5    3205958
##  8 Washington                      1.68            201.  12791 1.53e6    7614893
##  9 Alaska                          1.71            346.   1268 2.56e5     740995
## 10 Maine                           1.74            189.   2335 2.54e5    1344212
```

The 10 states with the **highest death rates per 1000 population**.

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

```
## # A tibble: 10 x 6
##    Province_State deaths_per_thou cases_per_thou deaths   cases population
##    <chr>                    <dbl>          <dbl>  <dbl>   <dbl>      <dbl>
##  1 Mississippi               4.19           269.  12457  801527    2976149
##  2 Arizona                   4.15           279.  30230 2030925    7278717
##  3 Oklahoma                  4.04           264.  15996 1044179    3956971
##  4 Alabama                   4.00           266.  19628 1304710    4903185
##  5 West Virginia             3.85           282.   6893  505528    1792147
##  6 Tennessee                 3.85           298.  26265 2036315    6829174
##  7 Arkansas                  3.78           278.  11415  838251    3017804
##  8 New Jersey                3.78           260.  33537 2313062    8882190
##  9 Louisiana                 3.72           254.  17295 1178806    4648794
## 10 New Mexico                3.63           251.   7607  526137    2096829
```

The 10 states with the **highest case rates per 1000 population**.

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

```
## # A tibble: 10 x 6
##    Province_State cases_per_thou deaths_per_thou deaths   cases population
##    <chr>                   <dbl>           <dbl>  <dbl>   <dbl>      <dbl>
##  1 Rhode Island             360.            3.35   3552  381271    1059361
##  2 Alaska                   346.            1.71   1268  256089     740995
##  3 North Dakota             318.            2.98   2272  242462     762062
##  4 Kentucky                 299.            3.54  15797 1336858    4467673
##  5 Tennessee                298.            3.85  26265 2036315    6829174
##  6 Guam                     295.            2.20    361   48496     164229
##  7 Utah                     293.            1.49   4761  938864    3205958
##  8 South Carolina           288.            3.47  17869 1481975    5148714
##  9 West Virginia            282.            3.85   6893  505528    1792147
## 10 Wisconsin                282.            2.49  14502 1639365    5822434
```

```
us_state_totals
```

```
## # A tibble: 56 x 6
##    Province_State        deaths  cases population cases_per_thou deaths_per_thou
##    <chr>                  <dbl>  <dbl>      <dbl>          <dbl>           <dbl>
##  1 American Samoa            30   5999      55641           108.           0.539
##  2 Northern Mariana Isl~     34  11305      55144           205.           0.617
##  3 Virgin Islands           113  18037     107268           168.           1.05
##  4 Guam                     361  48496     164229           295.           2.20
##  5 Vermont                  648 128471     623989           206.           1.04
##  6 Alaska                  1268 256089     740995           346.           1.71
##  7 District of Columbia    1340 144675     705749           205.           1.90
##  8 Hawaii                  1434 258422    1415872           183.           1.01
##  9 Wyoming                 1817 157031     578759           271.           3.14
## 10 North Dakota            2272 242462     762062           318.           2.98
## # ... with 46 more rows
```

My Additional Analyses and Visualizations In these analyses I tried to filter and limit the number of states to approximately 10 so that the visualizations are not too crowded. I looked at the states with the highest death rates per 1000 and highest case rates per thousand.

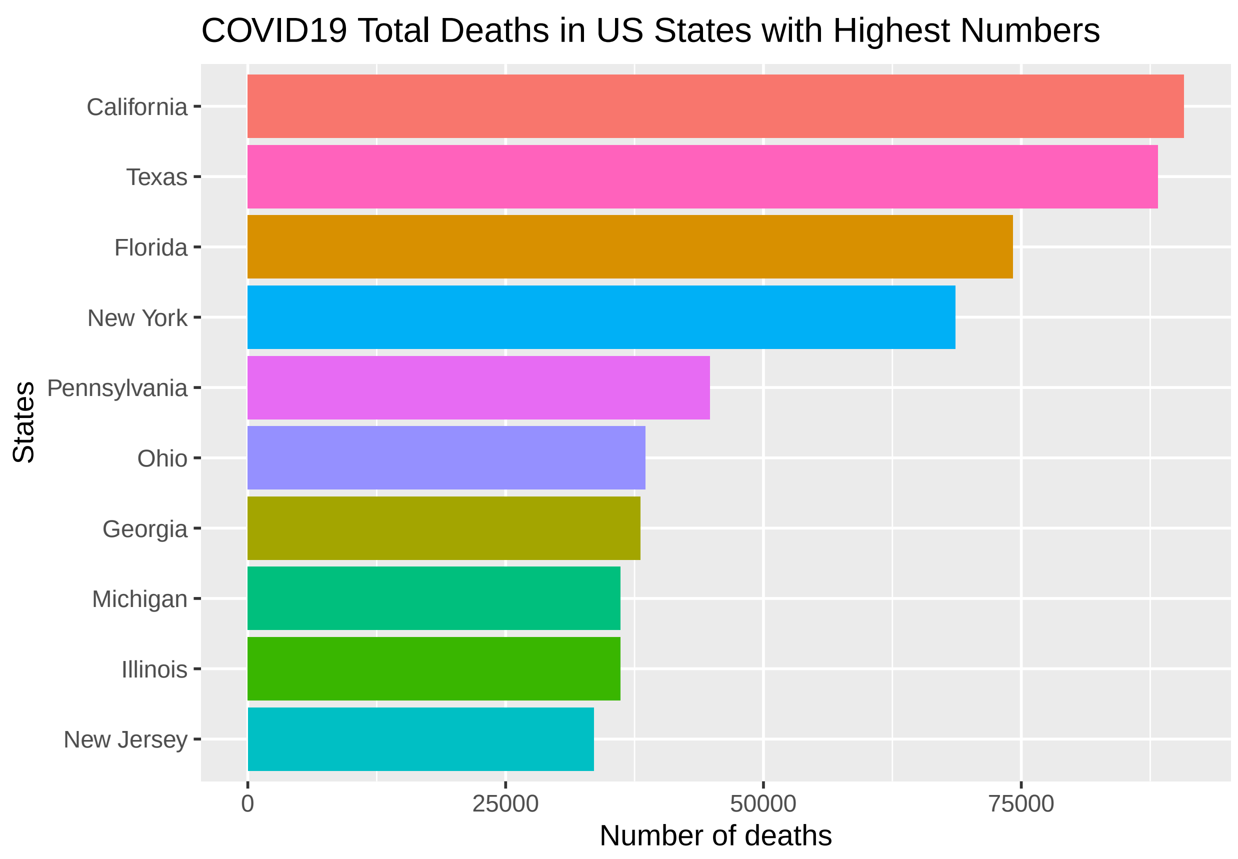
Visualize the **total deaths per state** for those states that have a total death of *greater than 33000*. First, `filter` the data and analyze it in a table; `arrange` in ascending order of number of deaths.

```
deaths_states_1 <- us_state_totals %>%
  filter(deaths > 33000) %>%
  arrange(deaths) %>%
  select(Province_State, deaths)

deaths_states_1
```

```
## # A tibble: 10 x 2
##    Province_State deaths
##    <chr>           <dbl>
##  1 New Jersey      33537
##  2 Illinois        36138
##  3 Michigan        36140
##  4 Georgia         38086
##  5 Ohio            38550
##  6 Pennsylvania    44814
##  7 New York        68603
##  8 Florida         74178
##  9 Texas           88240
## 10 California      90782
```

Visualize the above table as a bar graph (with geom_col()). Use `fct_reorder()` from forcats package to reorder the States in descending order of number of total **deaths**.

```
deaths_states_1 %>%
  ggplot(aes(x = fct_reorder(Province_State, deaths), y = deaths, group = Province_State, fill = Provinc
  geom_col(show.legend = FALSE) +
  coord_flip() +
  labs(x = "States", y = "Number of deaths", title = "COVID19 Total Deaths in US States with Highest Nu
```

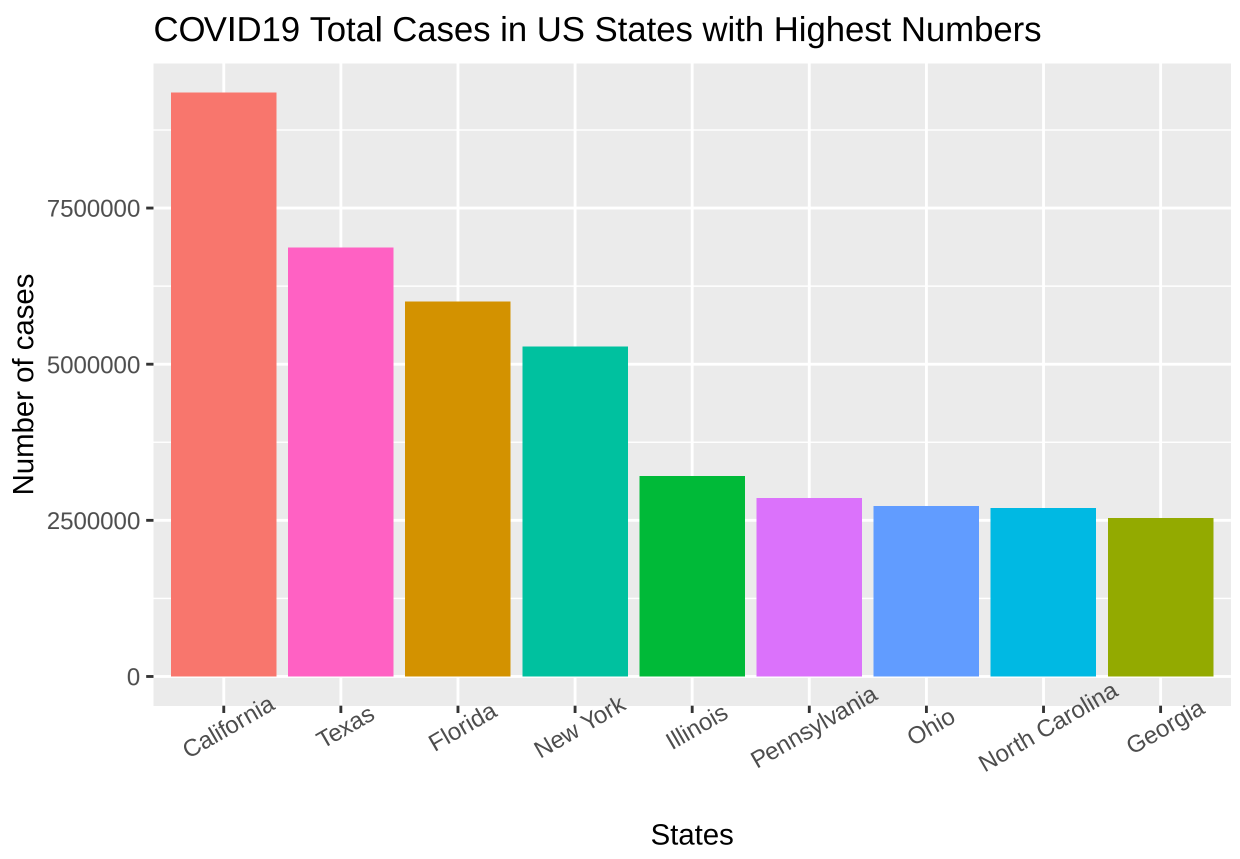


```
deaths_states_1
```

```
## # A tibble: 10 x 2
##    Province_State deaths
##    <chr>           <dbl>
##  1 New Jersey      33537
##  2 Illinois        36138
##  3 Michigan        36140
##  4 Georgia         38086
##  5 Ohio            38550
##  6 Pennsylvania    44814
##  7 New York        68603
##  8 Florida         74178
##  9 Texas           88240
## 10 California      90782
```

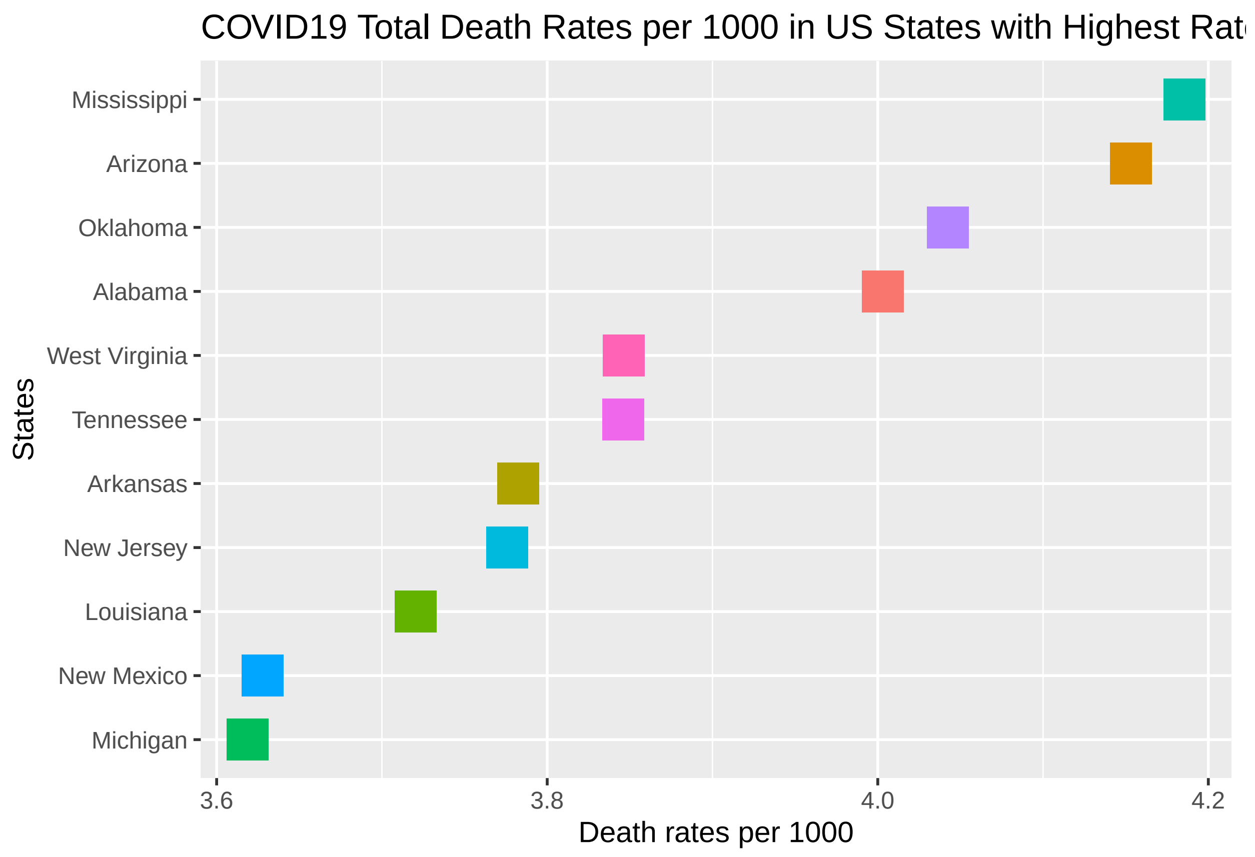
Reverse ordering , i.e., ascending, is achieved with `fct_rev()`.

```
cases_states <- us_state_totals %>%
  filter(cases > 2500000) %>%
  ggplot(aes(x = fct_rev(fct_reorder(Province_State, cases)), y = cases, group = Province_State, fill =
  geom_col(show.legend = FALSE) +
  theme(legend.position="bottom",
        axis.text.x = element_text(angle = 30)) +
  labs(x = "States", y = "Number of cases", title = "COVID19 Total Cases in US States with Highest Numbe


cases_states
```



Visualize the states with the **highest death rates per 1000**.

```
# need to improve the labeling
deaths_states_2 <- us_state_totals %>%
  filter(deaths_per_thou > 3.6) %>%
  ggplot(aes( x = fct_reorder(Province_State, deaths_per_thou), y =deaths_per_thou, group = Province_Sta
  geom_point(show.legend = FALSE, size = 7, shape = 15) +
  coord_flip() +
  theme(legend.position="bottom",
        axis.text.x = element_text(angle = 0)) +
  labs(title = "COVID19 Total Death Rates per 1000 in US States with Highest Rates", x = "States", y =

deaths_states_2
```



Question: What is the trend of new_deaths in the states which have had the highest death rates in the last 2 years?

At what time points during the pandemic have they had the highest rate of `new_deaths`?

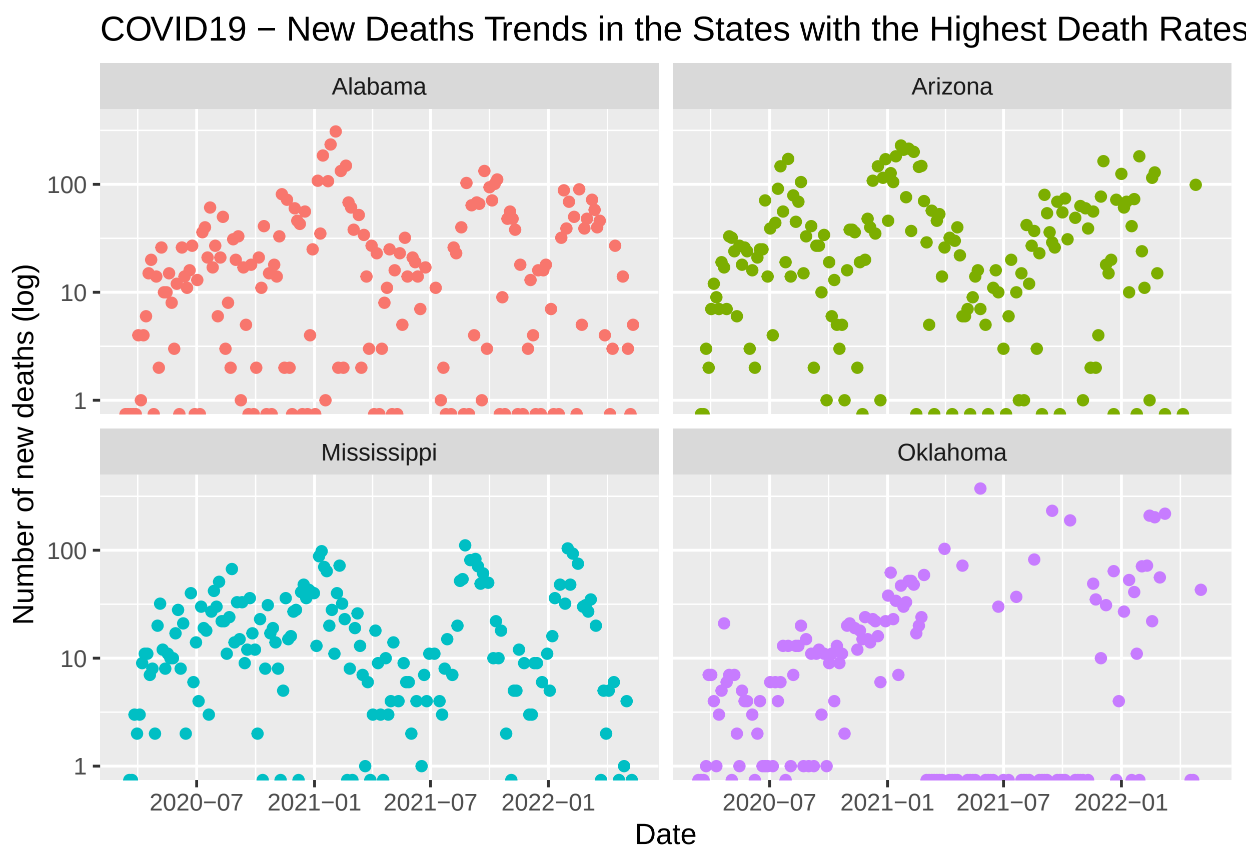
Has the number of `new_deaths` fluctuated with adopted preventative measures and policies and vaccinations in these states? To answer this question we need more data regarding the type and timeline of the policies, vaccination administration as well as the introduction of new COVID strains in the specific locale.

```
# filtered the deaths_per_thou to greater than 3.9, so that I would have less than 5 states to visualiz

states_highest_death_rate <- us_state_totals %>%
  filter(deaths_per_thou > 3.9) %>%
  select(Province_State)

# use deframe() to change the tibble to a vector
states_highest = deframe(states_highest_death_rate)
states_highest
```

```
## [1] "Mississippi" "Oklahoma"    "Alabama"     "Arizona"
```

Plot the `new_deaths` in these states.

```
us_by_state %>%
  filter(new_cases > 0 & Province_State == states_highest) %>%
  ggplot(aes(x = date, y = new_deaths, group = Province_State, color = factor(Province_State))) +
  geom_point(show.legend = FALSE) +
  scale_y_log10() +
  # used facet_wrap to graph the data for each state individually
  facet_wrap(~ Province_State, ncol = 2) +
  labs(title = "COVID19 - New Deaths Trends in the States with the Highest Death Rates", y = "Number of
```



Question: What is the trend of new_deaths in the states which have had the lowest death rates in the last 2 years?

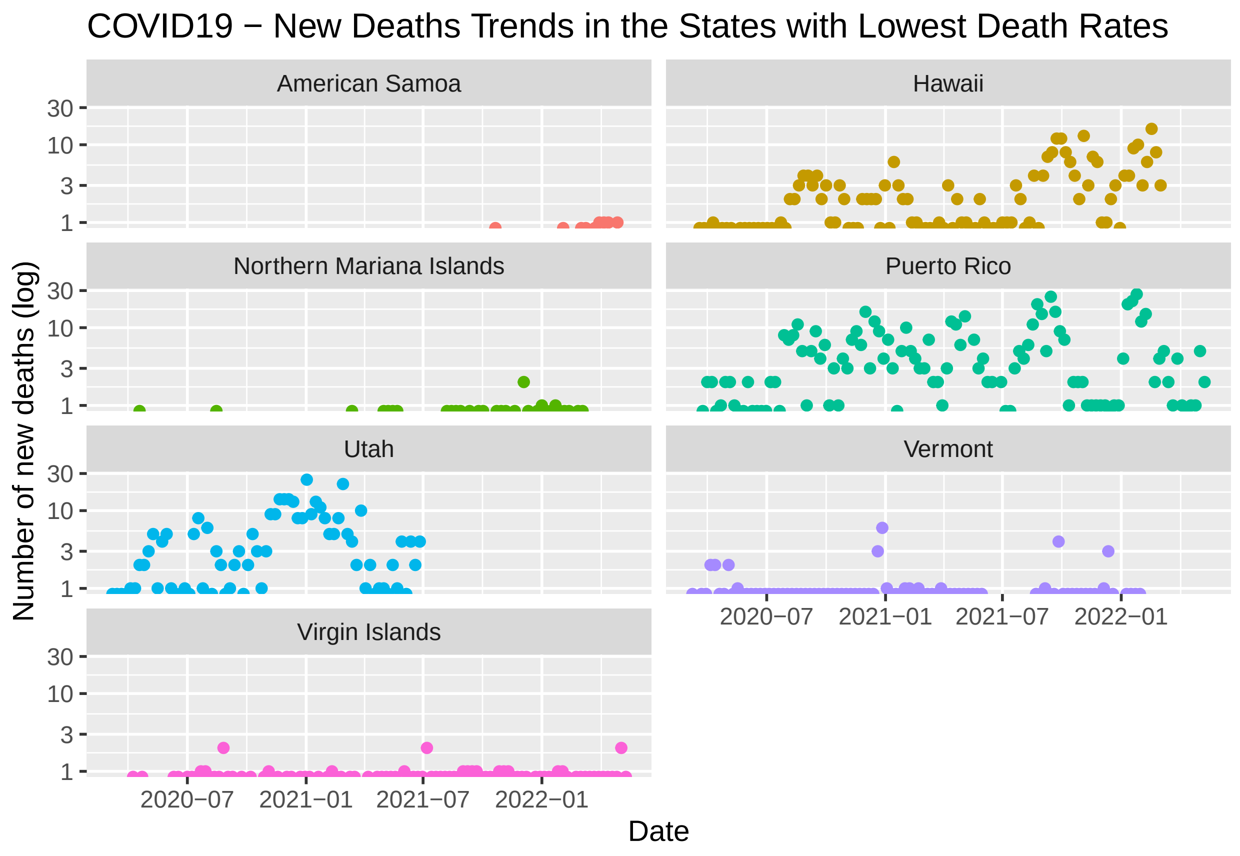
How do these compare to the states which have had the highest death rates?

```
# filtered the deaths_per_thou to greater than 3.9, so that I would have less than 5 states to visualiz

states_lowest_death_rate <- us_state_totals %>%
  filter(deaths_per_thou < 1.5) %>%
  select(Province_State)

# use deframe() to change the tibble to a vector
states_lowest = deframe(states_lowest_death_rate)
states_lowest
```

```
## [1] "American Samoa"           "Northern Mariana Islands"
## [3] "Virgin Islands"           "Vermont"
## [5] "Hawaii"                   "Puerto Rico"
## [7] "Utah"
```

Plot the `new_deaths` in these states.

```
us_by_state %>%
  filter(new_cases > 0 & Province_State == states_lowest) %>%
  ggplot(aes(x = date, y = new_deaths, group = Province_State, color = factor(Province_State))) +
  geom_point(show.legend = FALSE) +
  scale_y_log10() +
  facet_wrap(~ Province_State, ncol = 2) +
  labs(title = "COVID19 - New Deaths Trends in the States with Lowest Death Rates", y = "Number of new 
```



Observation: Except for the state with very low numbers of new deaths, in both the states with the highest and lowest new deaths we can discern a cyclical pattern of high and lows that may correspond to the emergence of the COVID19 virus strains or specific national events (such as travel and gatherings during the Christmas and New Year holidays).

Modeling the Data - part 1

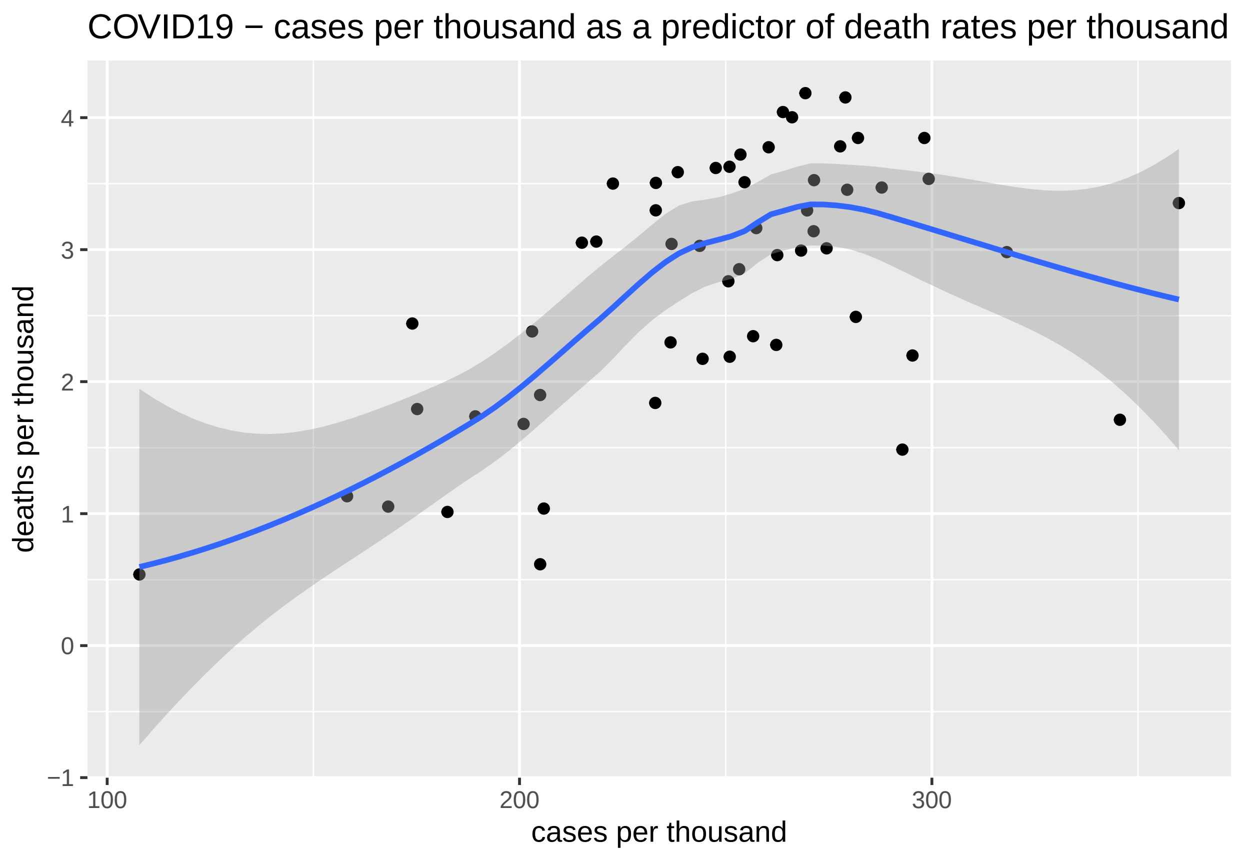
As enumerated in the week 3 lecture on 'Modeling Data' some variables that can be added and considered for to the model include population density, climate of the area, political affiliation, extent of the lock down, etc.

Model 1: `cases_per_thou` as a predictor for `deaths_per_thou` (a linear model)

Plot the two variables as a scatter plot to see their relationship.

```
us_state_totals %>% ggplot(aes(cases_per_thou,deaths_per_thou)) +
  geom_point() +
  geom_smooth() +
  labs(title = "COVID19 - cases per thousand as a predictor of death rates per thousand", x = "cases pe
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



The linear model …

```
mod_1 <- lm(deaths_per_thou ~ cases_per_thou, data = us_state_totals)

summary(mod_1)
```

```
##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou, data = us_state_totals)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2440 -0.5617  0.1102  0.6547  1.1506
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.213801   0.594120  -0.360     0.72
## cases_per_thou  0.012063   0.002365   5.102 4.47e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7968 on 54 degrees of freedom
## Multiple R-squared:  0.3252, Adjusted R-squared:  0.3127
## F-statistic: 26.03 on 1 and 54 DF,  p-value: 4.469e-06
```

We can calculate`slice_min` and `slice_max` of the variable `cases_per_thou` to determine the range of values in `cases_per_thou`.

```
min_cases <- us_state_totals %>%
  slice_min(cases_per_thou) %>%
  select(cases_per_thou)

min_cases = as.integer(min_cases)
min_cases
```

```
## [1] 107
```

```
max_cases <- us_state_totals %>%
  slice_max(cases_per_thou) %>%
  select(cases_per_thou)

max_cases = as.integer(max_cases)
max_cases
```

```
## [1] 359
```

The `cases_per_thou` variable therefore ranges from 107 to 359.

`mutate` a new variable for the predicted death rate per thousand (`pred`) and `arrange` the tibble' in ascending order for the value of `pred`.

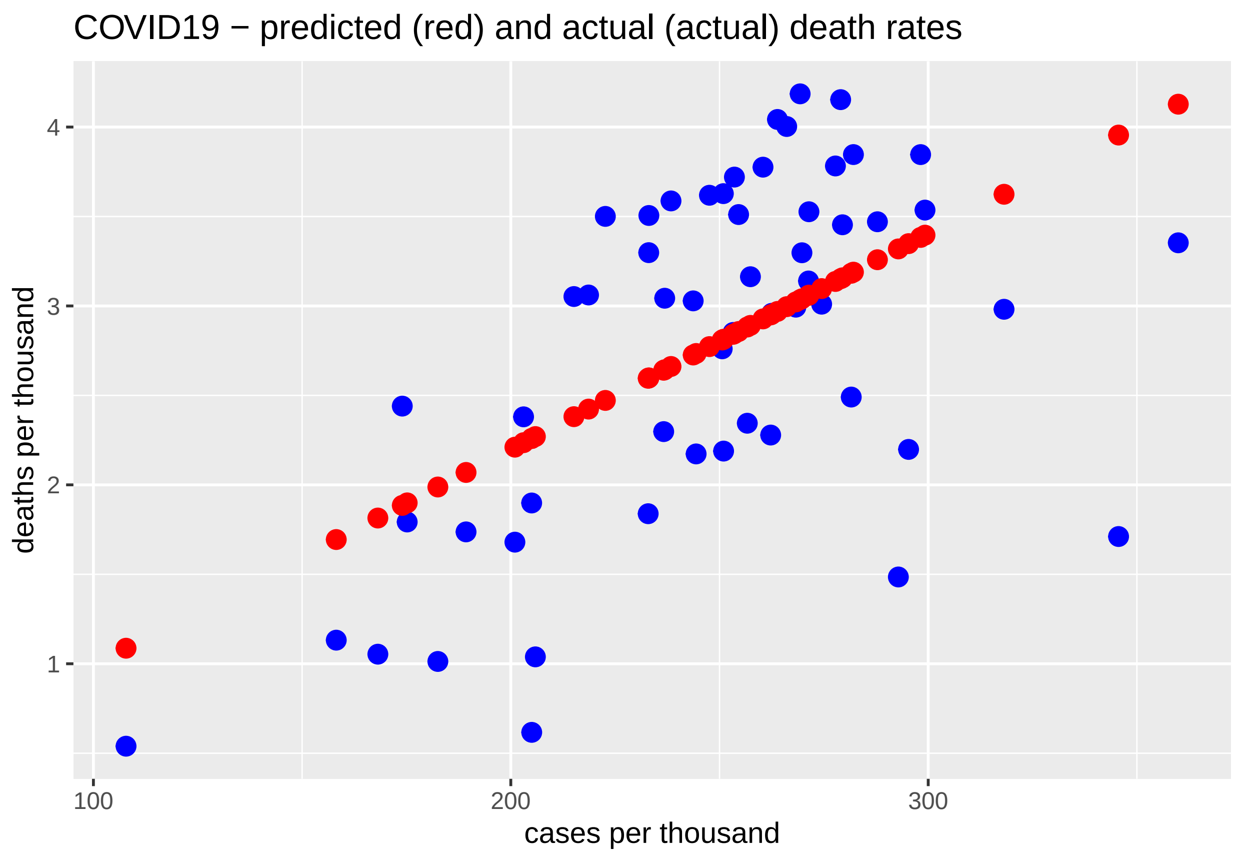
Look at the 10 highest predicted death rates using `tail`.

```
us_state_w_pred <- us_state_totals %>%
  mutate(pred = predict(mod_1)) %>%
  arrange(pred) %>%
  select(Province_State, pred, deaths_per_thou, cases_per_thou, everything())

tail(us_state_w_pred, 10)
```

```
## # A tibble: 10 x 7
##    Province_State  pred deaths_per_thou cases_per_thou deaths   cases population
##    <chr>          <dbl>           <dbl>          <dbl>  <dbl>   <dbl>      <dbl>
##  1 Wisconsin       3.18            2.49           282.  14502 1639365    5822434
##  2 West Virginia   3.19            3.85           282.   6893  505528    1792147
##  3 South Carolina  3.26            3.47           288.  17869 1481975    5148714
##  4 Utah            3.32            1.49           293.   4761  938864    3205958
##  5 Guam            3.35            2.20           295.    361   48496     164229
##  6 Tennessee       3.38            3.85            298.  26265 2036315    6829174
##  7 Kentucky        3.40            3.54            299.  15797 1336858    4467673
##  8 North Dakota    3.62            2.98            318.   2272  242462     762062
##  9 Alaska          3.96            1.71            346.   1268  256089     740995
## 10 Rhode Island    4.13            3.35            360.   3552  381271    1059361
```

Plot the actual `deaths_per_thou` and the predicted death per thousand (`pred`) and compare the values.

```
us_state_w_pred  %>% ggplot +
  geom_point(aes(x = cases_per_thou, y = deaths_per_thou), color = "blue", size = 3) +
  geom_point(aes(x = cases_per_thou, y = pred), color = "red", size = 3) +
  labs(title = "COVID19 - predicted (red) and actual (actual) death rates", x = "cases per thousand", y
```



The predicted values which were shown in the week 3 lecture that was recorded approximately a year ago demonstrated that `mod_1` did a relatively good job of predicting the `death_per_thou` at the lower and higher ends of the `cases_per_thou` range.

However, this is not necessarily as true when we analyze the data for the last 2 years - probably secondary to a host of additional variables that have been introduced such as more widespread vaccinations and new COVID strains with different degrees of contagiousness and disease severity.

Modeling the Data - part 2

The second linear model will evaluate population density as an independent variable and cases per thousand as the dependent variable using the US data set.

The population densities for the year 2020 are from the **"List of states and territories of the United States by population density"**, link, on Wikepedia.

I could not find this info readily as a csv file and therefore, I chose to add the population density (per kilometer squared) to a vector, `pop_density`, and then add this vector as a column to the `us_states` data set.

```
# Population density of the 56 states (in alphabetical order)

pop_density <- c(38.3, 0.5, 251, 24.3, 22.3, 98.0, 21.5, 288, 196, 4361, 155, 71.9, 283, 87.5, 8.6, 89.


head(pop_density)
```

```
## [1]  38.3   0.5 251.0  24.3  22.3  98.0
```

Add this vector as a column to `us_state_totals`.

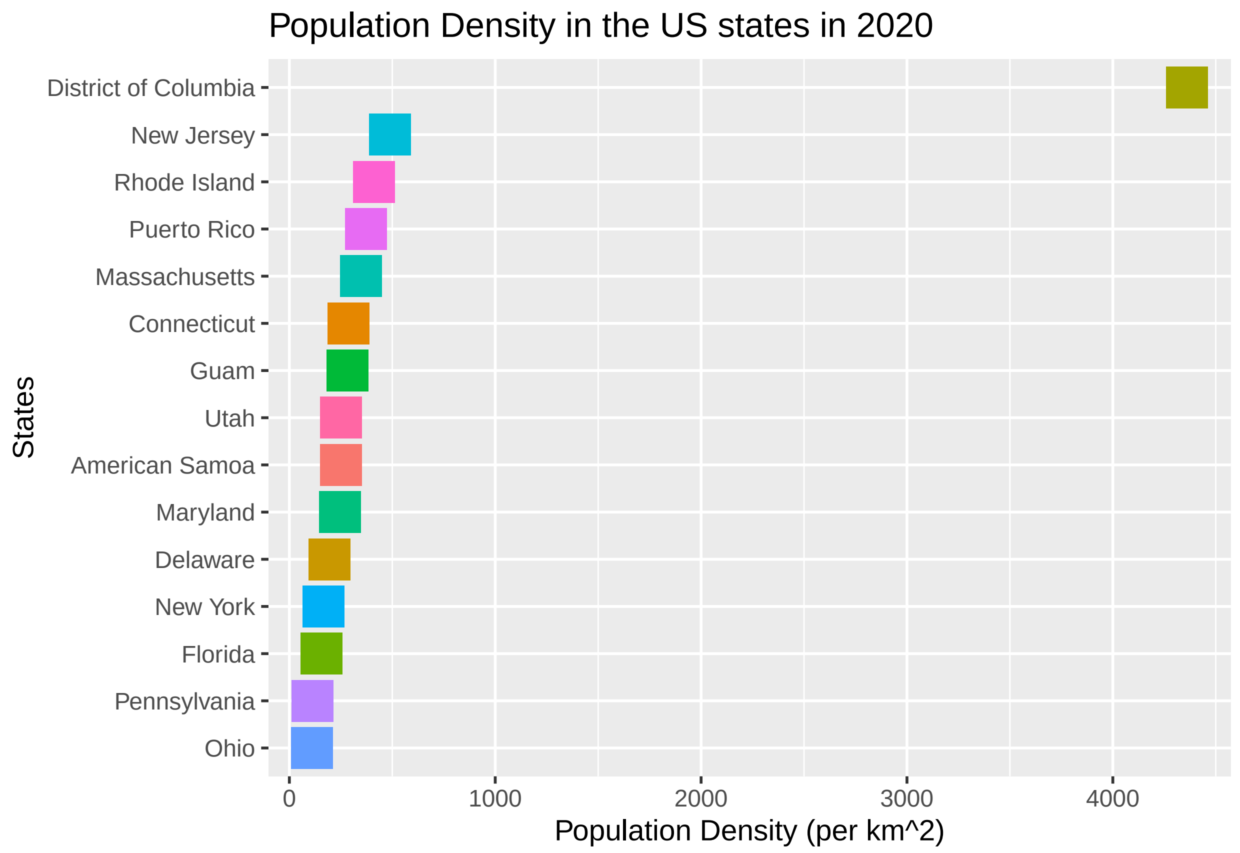
```
# arrange the tibble in alphaberical order (bases on Province_State) to match the order in the pop_dens
us_state_totals <- us_state_totals %>%
  arrange(Province_State)

# add the new column of pop_density to the tibble
us_state_totals$Pop_Density <- pop_density

# print out the new tibble and check that the correct values in the Pop_Density column have been associ
us_state_totals %>%
  select(Province_State, Pop_Density, everything())
```

```
## # A tibble: 56 x 7
##    Province_State       Pop_Density deaths   cases population cases_per_thou
##    <chr>                      <dbl>  <dbl>   <dbl>      <dbl>          <dbl>
##  1 Alabama                     38.3  19628 1304710    4903185           266.
##  2 Alaska                       0.5   1268  256089     740995           346.
##  3 American Samoa             251       30    5999      55641           108.
##  4 Arizona                     24.3  30230 2030925    7278717           279.
##  5 Arkansas                    22.3  11415  838251    3017804           278.
##  6 California                  98    90782 9349673   39512223           237.
##  7 Colorado                    21.5  12513 1407405    5758736           244.
##  8 Connecticut                288    10914  779460    3565287           219.
##  9 Delaware                   196     2931  267265     973764           274.
## 10 District of Columbia      4361     1340  144675     705749           205.
## # ... with 46 more rows, and 1 more variable: deaths_per_thou <dbl>
```

Observation: The District Of Columbia is somewhat of an outlier with a population density of 4361 per kilometers squared. New Jersey has the second highest population density.

Evaluate the range of values for population density in the 56.

```
us_state_totals %>%
  filter(Pop_Density > 100) %>%
  ggplot(aes(x = fct_reorder(Province_State, Pop_Density), Pop_Density,  group = Province_State, color =
  geom_point(show.legend = FALSE, size = 7, shape = 15) +
  coord_flip() +
  theme(legend.position="bottom",
        axis.text.x = element_text(angle = 0)) +
  labs(title = "Population Density in the US states in 2020", x = "States", y = "Population Density (pe
```



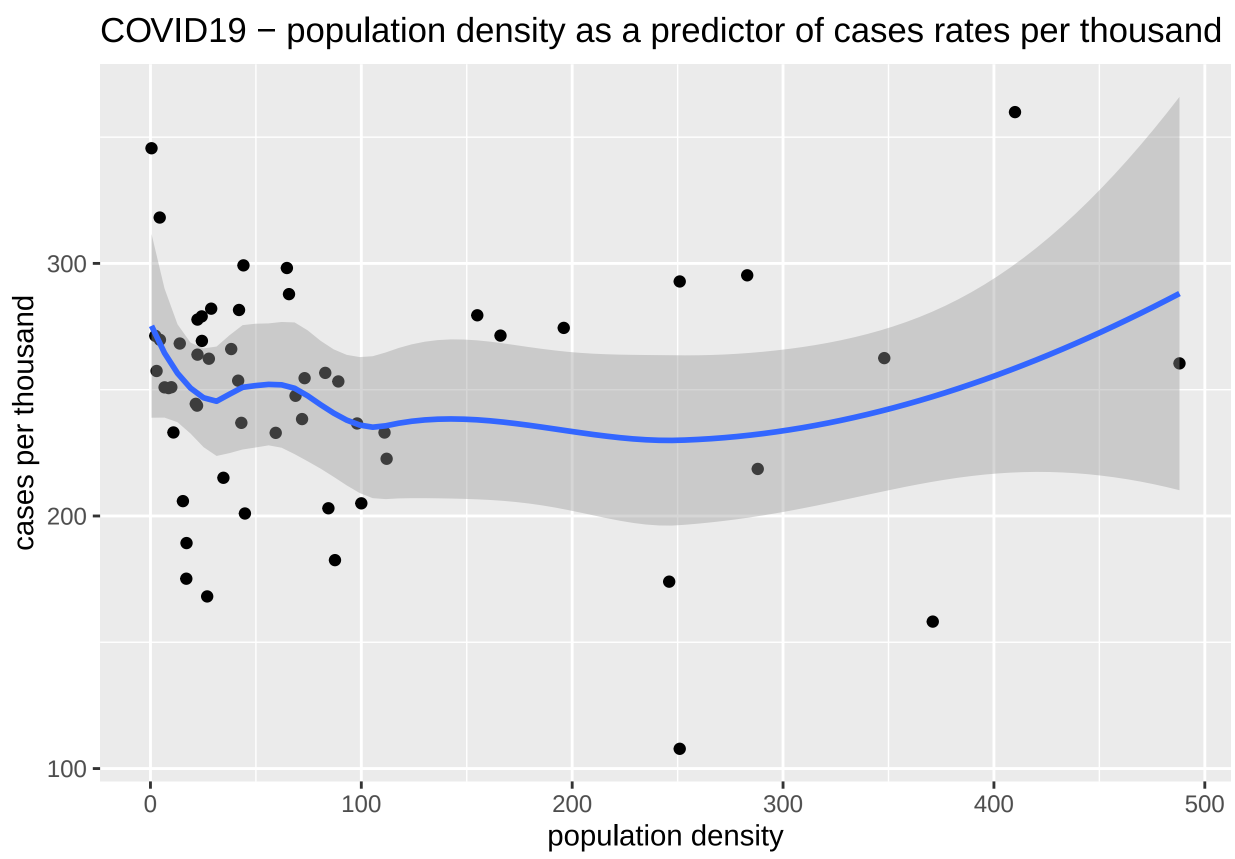
Observation: the population density for District of Columbia (DC) is an outlier.

Model 2`Pop_Density` as a predictor for cases_per_thou and`deaths_per_thou` (a linear model)

Plot the two variables as a scatter plot to see their relationship. As the population density for DC is an outlier, I will not include DC in the model.

```
us_state_totals_minusDC <- us_state_totals %>%
  filter(Province_State != "District of Columbia")

us_state_totals_minusDC %>%
  ggplot(aes(Pop_Density, cases_per_thou)) +
  geom_point() +
  geom_smooth() +
  labs(title = "COVID19 - population density as a predictor of cases rates per thousand", x = "populati
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



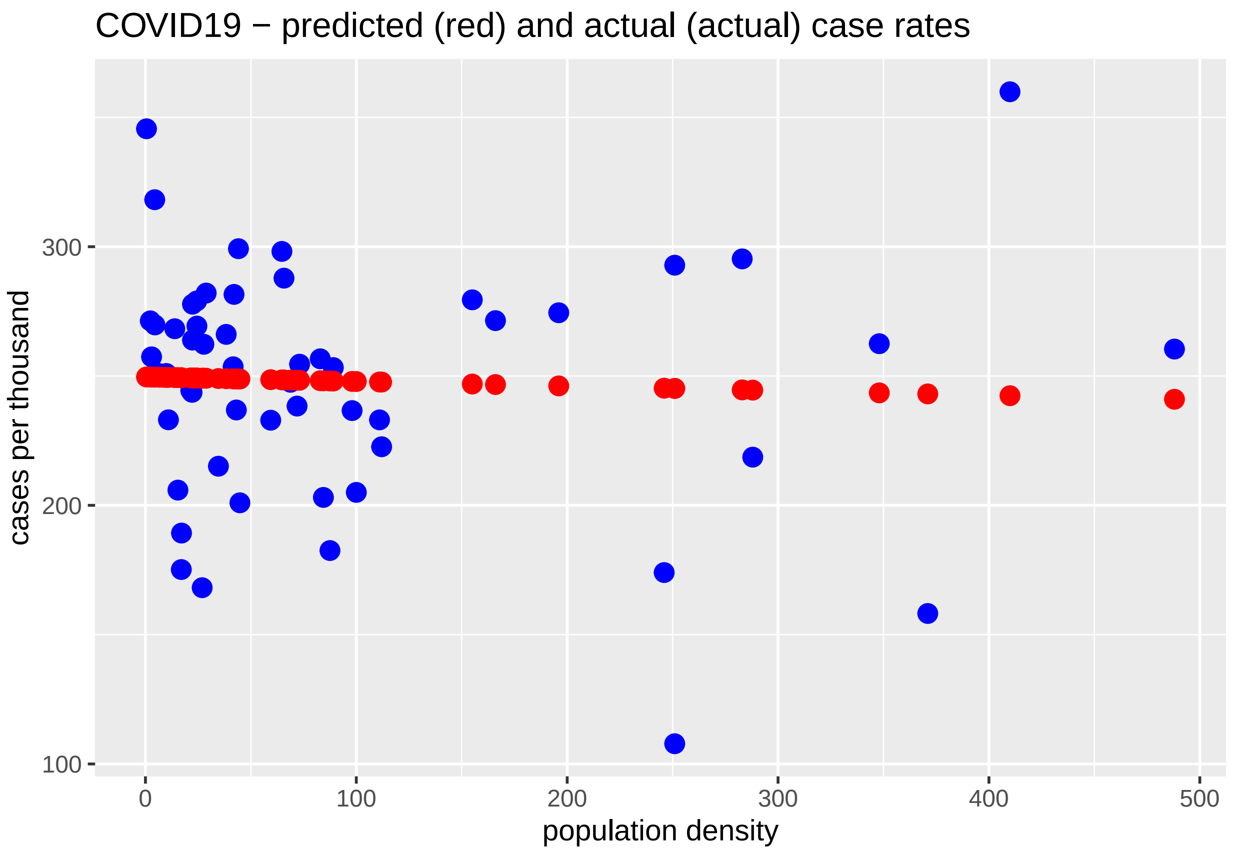
Linear model for the two variables.

```
mod_2 <- lm(cases_per_thou ~ Pop_Density, data = us_state_totals_minusDC)

summary(mod_2)
```

```
##
## Call:
## lm(formula = cases_per_thou ~ Pop_Density, data = us_state_totals_minusDC)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -137.383  -20.698    5.198   26.504  117.524
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 249.64726    8.04111  31.046   <2e-16 ***
## Pop_Density  -0.01772    0.05389  -0.329    0.744
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 45.87 on 53 degrees of freedom
## Multiple R-squared:  0.002036,   Adjusted R-squared:  -0.01679
## F-statistic: 0.1081 on 1 and 53 DF,  p-value: 0.7436
```

Plot the prediction for `cases_per_thou` based on `Pop_Density`.

```
us_state_w_pred_2 <- us_state_totals_minusDC  %>%
  mutate(cases_pred = predict(mod_2)) %>%
  ggplot() +
  geom_point(aes(x = Pop_Density, y = cases_per_thou), color = "blue", size = 3) +
  geom_point(aes(x = Pop_Density, y = cases_pred), color = "red", size = 3) +
  labs(title = "COVID19 - predicted (red) and actual (actual) case rates", x = "population density", y

us_state_w_pred_2
```



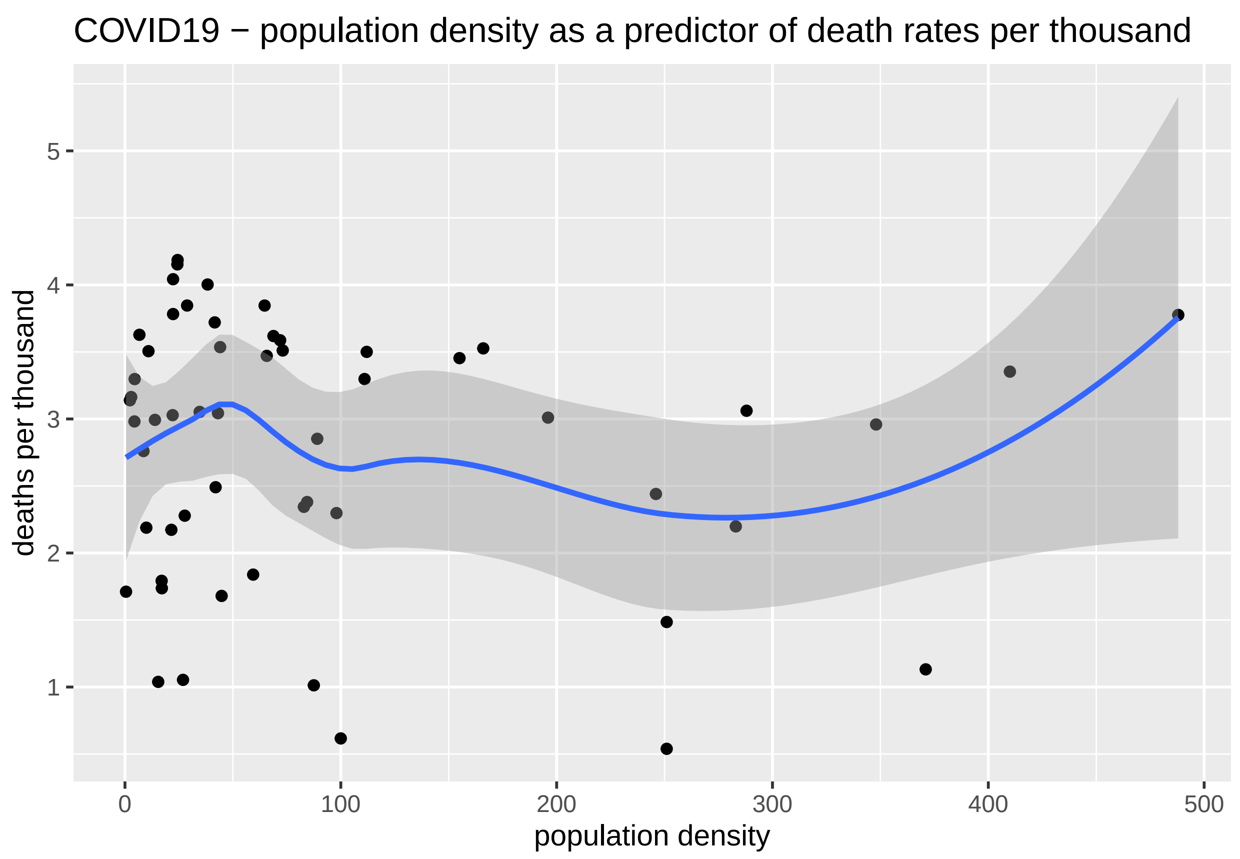
Observation: population density does not explain the variations in cases per thousand across different states. Note that this is overall a very big picture view of the problem as more optimally population density in different counties, cities or even neighborhoods could be used as a predictive variable (in conjunction with other predictive variables such as household income, . . .).

Model 3 `Pop_Density` as a predictor for `deaths_per_thou` (a linear model)

The same process can be used to evaluate the relationship between these two variables. I will replicate the steps in Model 2 as a chain of code chunks for this model.

```
us_state_totals_minusDC %>%
  ggplot(aes(Pop_Density, deaths_per_thou)) +
  geom_point() +
  geom_smooth() +
  labs(title = "COVID19 - population density as a predictor of death rates per thousand", x = "populati
```

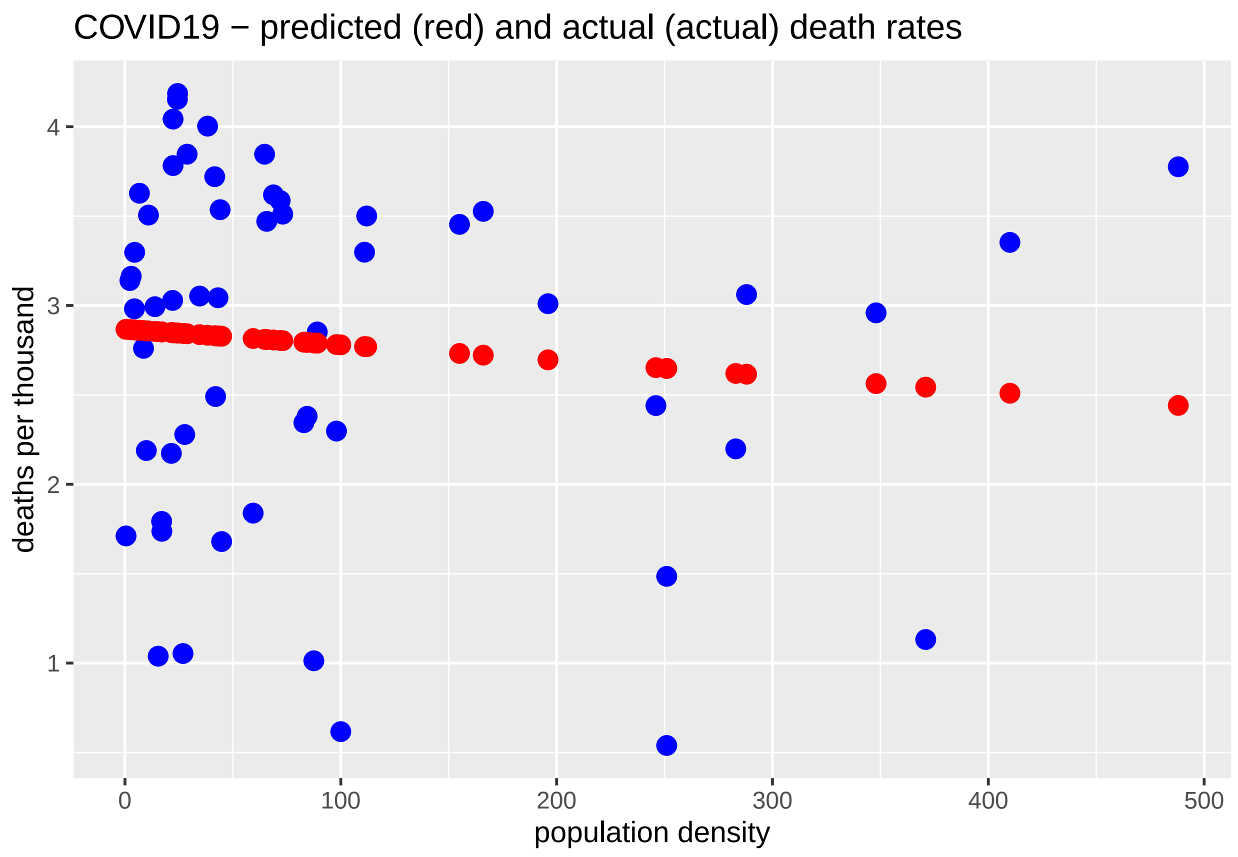
```
## ‘geom_smooth()‘ using method = ’loess’ and formula ’y ~ x’
```



```
mod_3 <- lm(deaths_per_thou ~ Pop_Density, data = us_state_totals_minusDC)

summary(mod_3)
```

```
##
## Call:
## lm(formula = deaths_per_thou ~ Pop_Density, data = us_state_totals_minusDC)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1632 -0.6171  0.2157  0.7489  1.3398
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.867110   0.169394  16.926   <2e-16 ***
## Pop_Density -0.000873   0.001135  -0.769    0.445
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9663 on 53 degrees of freedom
## Multiple R-squared:  0.01104,    Adjusted R-squared:  -0.007623
## F-statistic: 0.5915 on 1 and 53 DF,  p-value: 0.4453
```

```
us_state_w_pred_3 <- us_state_totals_minusDC  %>%
  mutate(deaths_pred = predict(mod_3)) %>%
  ggplot() +
  geom_point(aes(x = Pop_Density, y = deaths_per_thou), color = "blue", size = 3) +
  geom_point(aes(x = Pop_Density, y = deaths_pred), color = "red", size = 3) +
  labs(title = "COVID19 - predicted (red) and actual (actual) death rates", x = "population density", y

us_state_w_pred_3
```



Observation: Similar to Model 2, population density does not explain the variations in deaths per thousand across different states.

Sources of Bias

The sources of bias in the US and global data sets are due to how the data were collected and reported and a multitude of other factors. Different states and countries may have different criteria for counting an individual as having contracted COVID or attributing a death to COVID. The population density and lock down and masking policies effect the transmission of the virus in the community as well the cohorts who are more likely to get exposed. Another source of bias is the different strains of the COVID virus - these may all not have the same degree of contagiousness or cause the comparable disease severity. Access to the COVID vaccine and the timing of vaccination during the pandemic have varied both between and within countries - moreover, political affiliations have clouded individual's choices in receiving the vaccine.

The reporting bias could result in under reporting of both cases and deaths. These data also do not capture the age of those affected and those who succumbed to the disease - the susceptibility of different age groups to the evolving strains may also have changed over time.

Conclusion

For this assignment I replicated the code that used by Dr.Wall in the week three lectures and added my own analyses, visualizations and models. The assignment is somewhat lengthy as I have tried to meticulously document the exploratory and analytic steps. Moreover, I am new to R and I have learned a lot from the impressive online resources available for R programming and tidyverse which I tried to implement in the analyses and visualizations.

This assignment reinforced how complex data wrangling, analysis, visualization and modeling can be. I look forward to learning more on these subjects in future courses.

```
sessionInfo()
```

Session Info

```
## R version 4.1.2 (2021-11-01)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22000)
##
## 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] lubridate_1.8.0 forcats_0.5.1   stringr_1.4.0   dplyr_1.0.7
##  [5] purrr_0.3.4     readr_2.1.1     tidyr_1.1.4     tibble_3.1.6
##  [9] ggplot2_3.3.5   tidyverse_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_1.0.8       lattice_0.20-45  assertthat_0.2.1 digest_0.6.29
##  [5] utf8_1.2.2       R6_2.5.1         cellranger_1.1.0 backports_1.4.1
##  [9] reprex_2.0.1     evaluate_0.14    httr_1.4.2       highr_0.9
## [13] pillar_1.6.4     rlang_1.0.2      curl_4.3.2       readxl_1.3.1
## [17] rstudioapi_0.13  Matrix_1.3-4     rmarkdown_2.14   splines_4.1.2
## [21] labeling_0.4.2   bit_4.0.4        munsell_0.5.0    broom_0.7.11
## [25] compiler_4.1.2   modelr_0.1.8     xfun_0.30        pkgconfig_2.0.3
## [29] mgcv_1.8-38      htmltools_0.5.2  tidyselect_1.1.1 fansi_1.0.2
## [33] crayon_1.4.2     tzdb_0.2.0       dbplyr_2.1.1     withr_2.4.3
## [37] grid_4.1.2       nlme_3.1-157     jsonlite_1.7.3   gtable_0.3.0
## [41] lifecycle_1.0.1  DBI_1.1.2        magrittr_2.0.1   scales_1.2.0
## [45] cli_3.1.0        stringi_1.7.6    vroom_1.5.7      farver_2.1.0
## [49] fs_1.5.2         xml2_1.3.3       ellipsis_0.3.2   generics_0.1.1
## [53] vctrs_0.3.8      tools_4.1.2      bit64_4.0.5      glue_1.6.0
## [57] hms_1.1.1        parallel_4.1.2   fastmap_1.1.0    yaml_2.2.1
## [61] colorspace_2.0-2 rvest_1.0.2      knitr_1.37       haven_2.4.3
```