COVID19 Data from John Hopkins

Antony Sequeira

2/9/2022

Following script installs the required libraries in Mac OSX

This section can be copied to a file or input into R console.
You could also download it from my repo at https://github.com/asequeir-edu-2022/dtsa5301final

```
#!/usr/bin/env Rscript
r = getOption("repos")
r["CRAN"] = "http://cran.us.r-project.org"
options(repos = r)

print("Installing R libraries")
install.packages("chron")
install.packages("tidyverse")
install.packages("tinytex")

tinytex::install_tinytex()
```

Overview

For fulfillment of DTSA-5301 finals *COVID19 dataset from the Johns Hopkins github site* part of the assignment.

TODO explain data source

TODO explain data

TODO clean up data

TODO create various ARDs

TODO create plots (2)

TODO model (1)

TODO bias

Data source

The main data source is the github repository at https://github.com/CSSEGISandData/COVID-19

We will use the data from the folder at
https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series

Two of the time series tables are for the US confirmed cases and deaths, reported at the county level. They are `time_series_covid19_confirmed_US.csv` and `time_series_covid19_deaths_US.csv` respectively.

We will focus on analyzing the data for United States only.

These two files will provide our data for the analysis.

```
covid19_confirmed_url = "https://github.com/CSSEGISandData/COVID-19/raw/master/csse_covid_19_data/csse_
covid19_deaths_url = "https://github.com/CSSEGISandData/COVID-19/raw/master/csse_covid_19_data/csse_cov

covid19_confirmed_raw <- read_csv(covid19_confirmed_url)
covid19_deaths_raw <- read_csv(covid19_deaths_url)
```

Clean up data

We clean up the data mainly through the following three operations:
- variables to factor for appropriate columns
- date types from strings
- remove unneeded columns
- pivot

Pivot long

```
covid19_confirmed <- covid19_confirmed_raw %>%
  pivot_longer(cols = -c("UID", "iso2", "iso3", "code3", "FIPS", "Admin2",
                         "Province_State","Country_Region", "Lat", "Long_", "Combined_Key"),
                       names_to = "date",
                       values_to = "cases") %>%
  select(-c(Lat, Long_)) ## unwanted columns?

covid19_deaths <- covid19_deaths_raw %>%
  pivot_longer(cols = -c("UID", "iso2", "iso3", "code3", "FIPS", "Admin2",
                         "Province_State","Country_Region", "Lat", "Long_", "Combined_Key", "Population
                       names_to = "date",
                       values_to = "deaths") %>%
  select(-c(Lat, Long_)) ## unwanted columns?
```

Join cases and deaths data and change column types (factor, date)

```
covid19 <- covid19_confirmed %>%
  full_join(covid19_deaths) %>%
  mutate(iso3 = factor(iso3)) %>%
  mutate(Admin2 = factor(Admin2)) %>%
  mutate(Province_State = factor(Province_State)) %>%
  mutate(Country_Region = factor(Country_Region)) %>%
  mutate(date = mdy(date)) %>%
  # mutate(new_cases = cases - lag(cases)) %>%
  # mutate(new_deaths = deaths - lag(deaths)) %>%
  select (-c(UID, iso2, code3, FIPS))
```

```
## Joining, by = c("UID", "iso2", "iso3", "code3", "FIPS", "Admin2", "Province_State", "Country_Region"
```

```
# covid19 <- covid19 %>% filter(cases == 0)
```

Missing data columns and plans to handle them

There is missing data in the following columns:

```
names(which(colSums(is.na(covid19)) > 0))
```

```
## [1] "Admin2"
```

```
# filter(covid19, is.na(Admin2))
```

I do not need cruise ships data for this report.

```
covid19 <- covid19 %>%
  filter(iso3 == "USA") %>%  # filter non states for simplicity
  filter(!is.na(Admin2))     # ignore cruise ships
```

Summary of the cleaned up data

```
summary(covid19)
```

```
##   iso3                Admin2          Province_State    Country_Region
##  ASM:      0   Unassigned:  38352   Texas    : 192512   US:2448512
##  GUM:      0   Washington:  23312   Georgia  : 121072
##  MNP:      0   Jefferson :  19552   Virginia : 101520
##  PRI:      0   Franklin  :  18800   Kentucky :  91744
##  USA:2448512   Jackson   :  18048   Missouri :  88736
##  VIR:      0   Lincoln   :  18048   Kansas   :  80464
##                (Other)   :2312400   (Other)  :1772464
##  Combined_Key            date                cases            Population
##  Length:2448512     Min.   :2020-01-22   Min.   :      0   Min.   :       0
##  Class :character   1st Qu.:2020-07-27   1st Qu.:     78   1st Qu.:    9738
##  Mode  :character   Median :2021-01-31   Median :   1044   Median :   24528
##                     Mean   :2021-01-31   Mean   :   7265   Mean   :  100964
##                     3rd Qu.:2021-08-07   3rd Qu.:   4108   3rd Qu.:   65422
##                     Max.   :2022-02-11   Max.   :2752398   Max.   :10039107
##
##      deaths
##  Min.   :    0.0
##  1st Qu.:    1.0
##  Median :   19.0
##  Mean   :  125.2
##  3rd Qu.:   74.0
##  Max.   :29764.0
##
```

filter data

```
covid19 <- covid19 %>%
  filter(cases > 0)
```

Data issues

There are deaths with population zero. This is because the deaths are recorded with unassigned county (Alaska).

Visualization and Analysis

Generate necessary analysis ready data.

Generate per state aggregated data.

```
covid19_by_state <- covid19 %>%
  group_by(Province_State, Country_Region, date) %>%
  summarise(cases = sum(cases), deaths = sum(deaths), Population = sum(Population)) %>%
  mutate(deaths_per_million = deaths * 1000000 / Population) %>%
```

```
  select(Province_State, Country_Region, date,
         cases, deaths, deaths_per_million, Population) %>%
  ungroup()
```

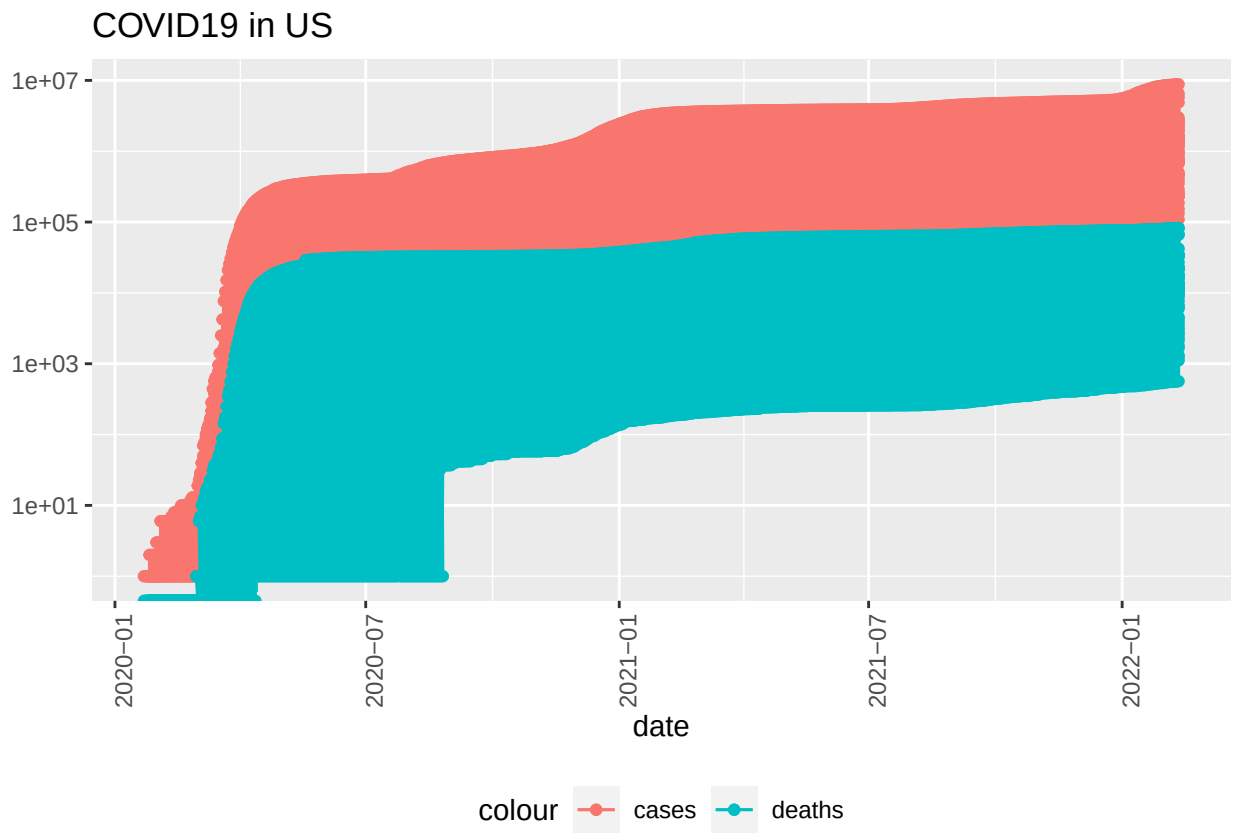
```
## `summarise()` has grouped output by 'Province_State', 'Country_Region'. You can override using the `
```

Plot

```
covid19_by_state %>%
  ggplot(aes(x = date, y = cases)) +

  geom_line(aes(color = "cases")) +
  geom_point(aes(color = "cases")) +

  geom_line(aes(y = deaths, color = "deaths")) +
  geom_point(aes(y = deaths, color = "deaths")) +

  scale_y_log10() +

  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +

  labs(title = "COVID19 in US", y = NULL)
```

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

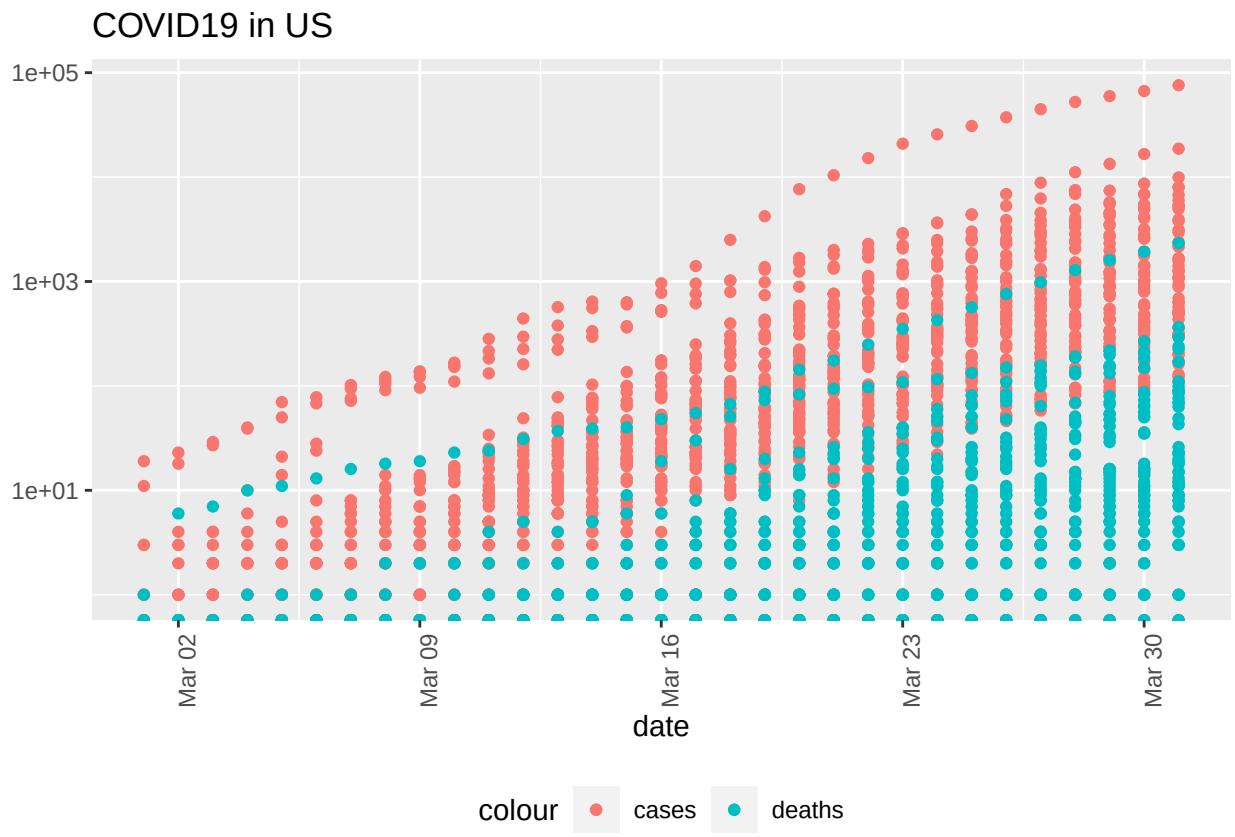
```
## Warning: Transformation introduced infinite values in continuous y-axis
```



Plot for just one month

```
covid19_by_state %>%
  filter (year(date) == 2020) %>%
  filter (month(date) == 3) %>%
  # filter (month(date) == 4 | month(date) == 5) %>%
  # filter (day(date) < 25) %>%
  ggplot(aes(x = date, y = cases)) +

  geom_point(aes(color = "cases")) +

  geom_point(aes(y = deaths, color = "deaths")) +

  scale_y_log10() +

  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +

  labs(title = "COVID19 in US", y = NULL)
```

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



Summarize by monthly numbers

```
covid19_by_state_month <- covid19_by_state %>%
  group_by(Province_State,  month = lubridate::floor_date(date, "month")) %>%
  summarise(cases = min(cases), # use value from the 1st of the month
            deaths = min(deaths),
            Population=max(Population),
            cases_per_million = cases * 1000000 / Population,
            deaths_per_million = deaths * 1000000 / Population
            ) %>%  # occurrences by month

#  mutate(cases_per_million = cases * 1000000 / Population) %>%
#  mutate(deaths_per_million = deaths * 1000000 / Population)  %>%

  ungroup()
```

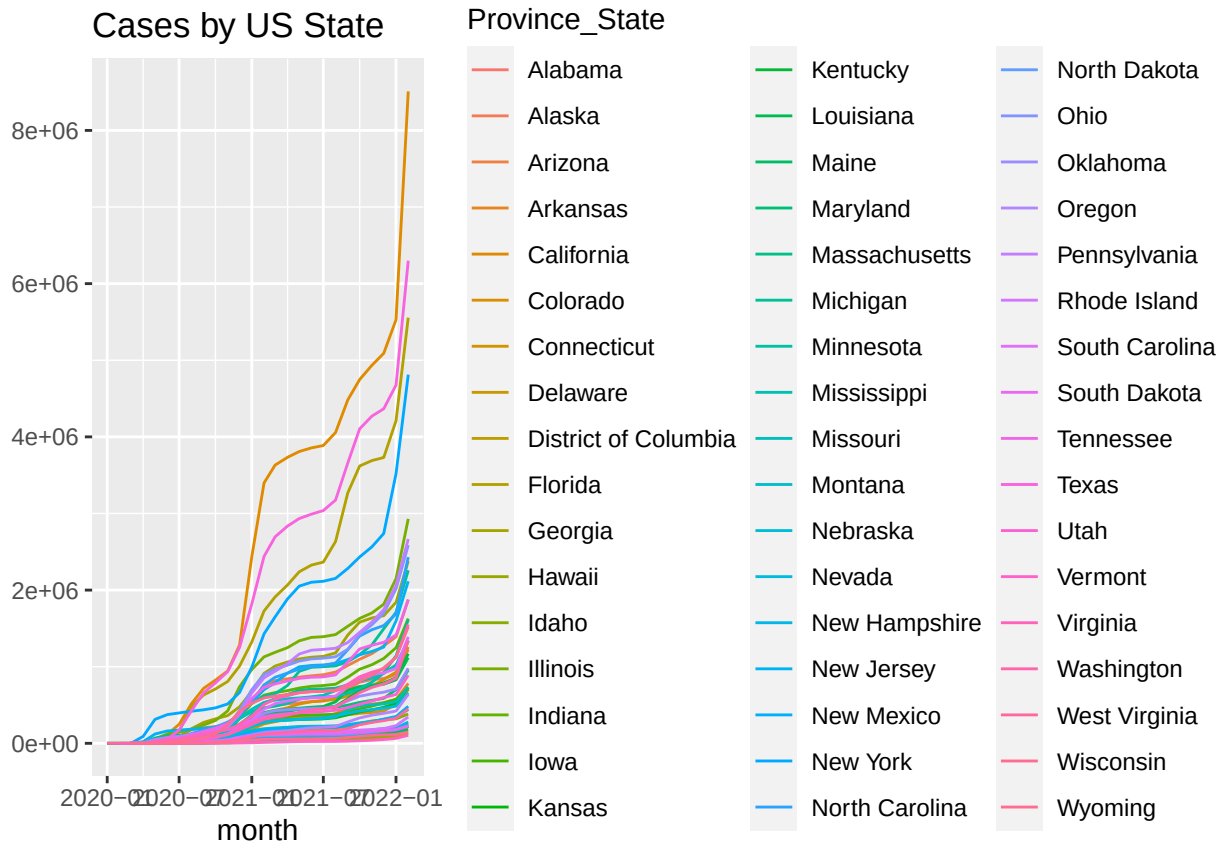
```
## `summarise()` has grouped output by 'Province_State'. You can override using the `.groups` argument.
```

```
# summary(covid19_by_state_month)
```

TBD slice_min and slice_max

Compare a few states

```
covid19_by_state_month %>%
  group_by(month) %>%
  ggplot(aes(x=month, y=cases, group=Province_State, color=Province_State)) +
  geom_line() +
  labs(title = "Cases by US State", y = NULL)
```




```
print.data.frame(covid19_by_state_month %>%
                   filter(month == as.Date("2022-01-01")) %>%
                   select (-c(month, cases_per_million)) %>%
                   arrange(cases)
                 )
```

```
##          Province_State   cases deaths Population deaths_per_million
## 1               Vermont   64447    471     623989           754.8210
## 2  District of Columbia   94286   1211     705749          1715.9075
## 3               Wyoming  115638   1526     578759          2636.6761
## 4                Hawaii  115642   1094    1415786           772.7157
## 5                 Maine  146736   1531    1344212          1138.9572
## 6                Alaska  157169    978     728809          1341.9154
## 7          North Dakota  174626   2012     762062          2640.2051
## 8          South Dakota  179204   2486     884659          2810.1223
## 9              Delaware  183880   2286     973764          2347.5914
```

```
## 10              Montana  197724   2906    1068778          2718.9931
## 11        New Hampshire  198667   1961    1359711          1442.2182
## 12         Rhode Island  231096   3066    1059361          2894.1975
## 13                Idaho  319382   4162    1787065          2328.9584
## 14        West Virginia  328162   5336    1792147          2977.4343
## 15             Nebraska  338257   3338    1934408          1725.5925
## 16           New Mexico  350043   5855    2096829          2792.3116
## 17               Oregon  421263   5655    4217737          1340.7664
## 18               Nevada  484641   8419    3080156          2733.3031
## 19          Connecticut  510188   9160    3565287          2569.2181
## 20               Kansas  520388   6970    2913314          2392.4644
## 21          Mississippi  543737  10450    2976149          3511.2489
## 22             Arkansas  570641   9180    3017804          3041.9471
## 23                 Iowa  575501   7858    3155070          2490.5945
## 24                 Utah  636992   3787    2315956          1635.1779
## 25             Maryland  700553  11758    6045680          1944.8598
## 26             Oklahoma  708938  12419    3956971          3138.5118
## 27            Louisiana  828695  14986    4648794          3223.6318
## 28           Washington  849075   9853    7614893          1293.9118
## 29             Kentucky  856145  12118    4467673          2712.3740
## 30              Alabama  896614  16455    4903185          3355.9819
## 31             Colorado  929275  10271    5758736          1783.5511
## 32       South Carolina  975320  14636    5148714          2842.6516
## 33             Missouri 1013458  16227    6626371          2448.8517
## 34            Minnesota 1022212  10656    5639632          1889.4850
## 35             Virginia 1118518  15587    8535519          1826.1338
## 36            Wisconsin 1120663  11173    5822434          1918.9569
## 37        Massachusetts 1140614  20273    6863772          2953.6238
## 38              Indiana 1246854  18386    6732219          2731.0460
## 39              Arizona 1389708  24354    7278717          3345.9193
## 40            Tennessee 1412302  20842    6829174          3051.9064
## 41           New Jersey 1596644  29053    8882190          3270.9276
## 42       North Carolina 1686667  19426   10488084          1852.1972
## 43             Michigan 1710325  29020    9986857          2905.8191
## 44              Georgia 1839879  31443   10617423          2961.4531
## 45                 Ohio 2016095  31794   11689100          2719.9699
## 46         Pennsylvania 2059613  36714   12801989          2867.8356
## 47             Illinois 2149548  30254   12671821          2387.5022
## 48             New York 3517696  59209   19453561          3043.6073
## 49              Florida 4209927  62504   21477737          2910.1762
## 50                Texas 4675575  75744   28995881          2612.2331
## 51           California 5528658  76520   39512223          1936.6159
```

```
print.data.frame(covid19_by_state_month %>%
                   filter(month == as.Date("2022-01-01")) %>%
                   select (-c(month, cases_per_million)) %>%
                   arrange(deaths_per_million)
                 )
```

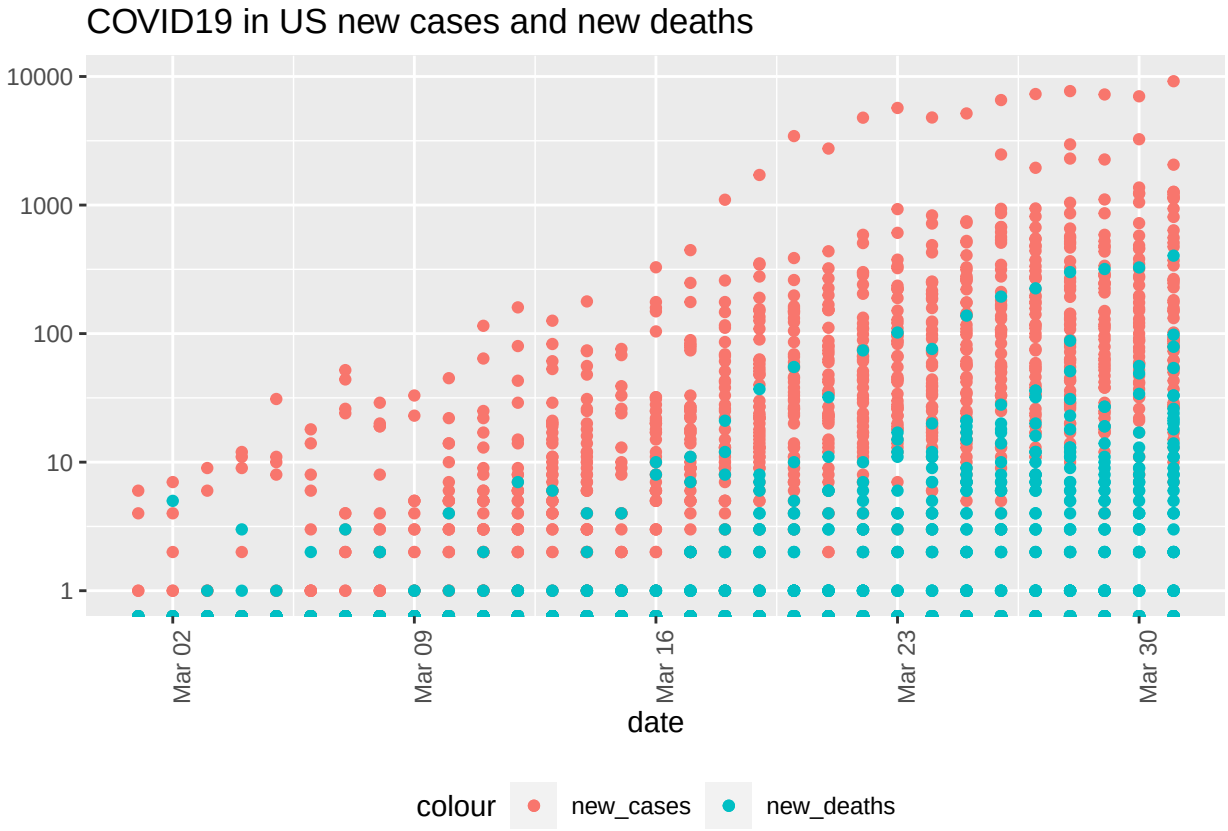
```
##          Province_State   cases deaths Population deaths_per_million
## 1               Vermont   64447    471     623989           754.8210
## 2                Hawaii  115642   1094    1415786           772.7157
## 3                 Maine  146736   1531    1344212          1138.9572
## 4            Washington  849075   9853    7614893          1293.9118
```

```
## 5                Oregon  421263   5655    4217737          1340.7664
## 6                Alaska  157169    978     728809          1341.9154
## 7         New Hampshire  198667   1961    1359711          1442.2182
## 8                  Utah  636992   3787    2315956          1635.1779
## 9  District of Columbia   94286   1211     705749          1715.9075
## 10             Nebraska  338257   3338    1934408          1725.5925
## 11             Colorado  929275  10271    5758736          1783.5511
## 12             Virginia 1118518  15587    8535519          1826.1338
## 13       North Carolina 1686667  19426   10488084          1852.1972
## 14            Minnesota 1022212  10656    5639632          1889.4850
## 15            Wisconsin 1120663  11173    5822434          1918.9569
## 16           California 5528658  76520   39512223          1936.6159
## 17             Maryland  700553  11758    6045680          1944.8598
## 18                Idaho  319382   4162    1787065          2328.9584
## 19             Delaware  183880   2286     973764          2347.5914
## 20             Illinois 2149548  30254   12671821          2387.5022
## 21               Kansas  520388   6970    2913314          2392.4644
## 22             Missouri 1013458  16227    6626371          2448.8517
## 23                 Iowa  575501   7858    3155070          2490.5945
## 24          Connecticut  510188   9160    3565287          2569.2181
## 25                Texas 4675575  75744   28995881          2612.2331
## 26              Wyoming  115638   1526     578759          2636.6761
## 27         North Dakota  174626   2012     762062          2640.2051
## 28             Kentucky  856145  12118    4467673          2712.3740
## 29              Montana  197724   2906    1068778          2718.9931
## 30                 Ohio 2016095  31794   11689100          2719.9699
## 31              Indiana 1246854  18386    6732219          2731.0460
## 32               Nevada  484641   8419    3080156          2733.3031
## 33           New Mexico  350043   5855    2096829          2792.3116
## 34         South Dakota  179204   2486     884659          2810.1223
## 35       South Carolina  975320  14636    5148714          2842.6516
## 36         Pennsylvania 2059613  36714   12801989          2867.8356
## 37         Rhode Island  231096   3066    1059361          2894.1975
## 38             Michigan 1710325  29020    9986857          2905.8191
## 39              Florida 4209927  62504   21477737          2910.1762
## 40        Massachusetts 1140614  20273    6863772          2953.6238
## 41              Georgia 1839879  31443   10617423          2961.4531
## 42        West Virginia  328162   5336    1792147          2977.4343
## 43             Arkansas  570641   9180    3017804          3041.9471
## 44             New York 3517696  59209   19453561          3043.6073
## 45            Tennessee 1412302  20842    6829174          3051.9064
## 46             Oklahoma  708938  12419    3956971          3138.5118
## 47            Louisiana  828695  14986    4648794          3223.6318
## 48           New Jersey 1596644  29053    8882190          3270.9276
## 49              Arizona 1389708  24354    7278717          3345.9193
## 50              Alabama  896614  16455    4903185          3355.9819
## 51          Mississippi  543737  10450    2976149          3511.2489
```

```
covid19_by_state <- covid19_by_state %>%
  mutate(new_cases = cases - lag(cases),
         new_deaths = deaths - lag(deaths)
         )
```

```r
covid19_by_state %>%
  filter (year(date) == 2020) %>%
  filter (month(date) == 3) %>%
  # filter (month(date) == 4 | month(date) == 5) %>%
  # filter (day(date) < 25) %>%
  ggplot(aes(x = date, y = new_cases)) +

  geom_point(aes(color = "new_cases")) +

  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 new cases and new deaths", 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: Removed 55 rows containing missing values (geom_point).

## Warning: Removed 52 rows containing missing values (geom_point).
```




```
# try population to cases model
#model_data <- covid19_by_state_month %>%
#  filter(month == as.Date("2022-01-01"))
#summary(model_data)
california_data = covid19_by_state_month %>%
  filter(cases > 0) %>%
  filter(Province_State == "California")
model <- lm(deaths ~ cases, data = california_data)
summary(model)
```

```
##
## Call:
## lm(formula = deaths ~ cases, data = california_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -30560.0  -4159.6    202.2   5769.1  10130.0
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.580e+03  2.496e+03   1.835   0.0789 .
## cases       1.247e-02  7.280e-04  17.135 5.79e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8422 on 24 degrees of freedom
```

```
## Multiple R-squared:  0.9244, Adjusted R-squared:  0.9213
## F-statistic: 293.6 on 1 and 24 DF,  p-value: 5.792e-15
```

```
#x_grid <- seq(80000,28000)
#new_df <- tibble(cases_per_million = x_grid)
#model_data_pred <- model_data %>% mutate(pred = predict(model))
california_data_pred  <- california_data %>%
  mutate(pred = predict(model))
```

```
california_data_pred %>%
  ggplot(aes(x = cases, y = deaths)) +

  geom_point(aes(color = "deaths")) +

  geom_point(aes(y = pred, color = "pred")) +

  # scale_y_log10() +

  theme(legend.position = "bottom",
        axis.text.x = element_text(angle = 90)) +

  labs(title = "COVID19 in US model fit", y = NULL)
```