COVID-19 Data Report

Robin Lutter

03-09-2021

Introduction

In the following I will analyze the COVID-19 Data. Here I listed the questions that I would like to answer and the information I would like to obtain from the analysis:

- Which states in the US had the highest number of cases and deaths so far?
- Take a closer look at the state with the highest numbers
  - Compare cases and deaths from 2020 with 2021
  - Consider the vaccination rate from 2021 and see how the numbers were affected
- In which way correlate the number of cases with the population of a state?

1. Include necessary libraries

```
library(tidyverse)
library(lubridate)
library(zoo)
library(gridExtra)
```

2. Import the datasets

The data for US cases and deaths is taken from the Johns Hopkins University on Github:

- https://github.com/CSSEGISandData/COVID-19/tree/master/csse_covid_19_data/csse_covid_19_time_series

The data for vaccinations in the US is taken from Kaggle:

- https://www.kaggle.com/paultimothymooney/usa-covid19-vaccinations

```
# prepare url's for cases and deaths data
base_url <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_c
file_names <- c("time_series_covid19_confirmed_US.csv",
                "time_series_covid19_deaths_US.csv")

urls <- str_c(base_url, file_names)

# read in the data sets
US_cases <- read_csv(urls[1])
US_deaths <- read_csv(urls[2])
```

```
# read in vaccination data
US_vaccinations <- read_csv("https://covid.ourworldindata.org/data/vaccinations/us_state_vaccinations.cs
```

3. Tidy the data

At this point we want to tidy the datasets which makes it easier to work with in the following steps.

```
# 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)
```

```
## # A tibble: 6 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
```

```
# 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)
```

```
## # A tibble: 6 x 7
##   Admin2 Province_State Country_Region Combined_Key Population date       deaths
##   <chr>  <chr>          <chr>          <chr>             <dbl> <date>      <dbl>
## 1 Autau~ Alabama        US             Autauga, Al~      55869 2020-01-22      0
## 2 Autau~ Alabama        US             Autauga, Al~      55869 2020-01-23      0
## 3 Autau~ Alabama        US             Autauga, Al~      55869 2020-01-24      0
## 4 Autau~ Alabama        US             Autauga, Al~      55869 2020-01-25      0
## 5 Autau~ Alabama        US             Autauga, Al~      55869 2020-01-26      0
## 6 Autau~ Alabama        US             Autauga, Al~      55869 2020-01-27      0
```

```
# tidy the vaccination data
US_vaccinations <- US_vaccinations |>
  rename(Province_State = location) |>
  select(date, Province_State, people_fully_vaccinated)

head(US_vaccinations)
```

```
## # A tibble: 6 x 3
##   date       Province_State people_fully_vaccinated
##   <date>     <chr>                            <dbl>
## 1 2021-01-12 Alabama                           7270
## 2 2021-01-13 Alabama                           9245
## 3 2021-01-14 Alabama                             NA
## 4 2021-01-15 Alabama                          13488
## 5 2021-01-16 Alabama                             NA
## 6 2021-01-17 Alabama                             NA
```

4. Process the data

Now we want combine the data of cases, deaths and vaccinations.

```
# join cases and death data
US_total <- US_cases |>
  full_join(US_deaths)

# combine county data
US_total_by_state <- US_total |>
  group_by(Province_State, Country_Region, date) |>
  summarize(cases = sum(cases), deaths = sum(deaths),
            Population = sum(Population)) |>
  select(Province_State, Country_Region, date, cases, deaths, Population) |>
  ungroup()

head(US_total_by_state)
```

```
## # A tibble: 6 x 6
##   Province_State Country_Region date       cases deaths Population
##   <chr>          <chr>          <date>     <dbl>  <dbl>      <dbl>
## 1 Alabama        US             2020-01-22     0      0    4903185
## 2 Alabama        US             2020-01-23     0      0    4903185
## 3 Alabama        US             2020-01-24     0      0    4903185
## 4 Alabama        US             2020-01-25     0      0    4903185
## 5 Alabama        US             2020-01-26     0      0    4903185
## 6 Alabama        US             2020-01-27     0      0    4903185
```

```
# combine cases and death data with vaccination data
US <- US_total_by_state |>
  merge(US_vaccinations, by=c("Province_State", "date"), all.x = TRUE) |>
  filter(!is.na(people_fully_vaccinated)) |>
  arrange(date)

# see that first documented vaccinations started 2021-01-12
head(US)
```

```
##   Province_State       date Country_Region   cases deaths Population
## 1        Alabama 2021-01-12            US  407848    5573    4903185
## 2         Alaska 2021-01-12            US   50544     225     740995
## 3 American Samoa 2021-01-12            US       0       0      55641
## 4        Arizona 2021-01-12            US  636100   10482    7278717
## 5       Arkansas 2021-01-12            US  259553    4121    3017804
## 6     California 2021-01-12            US 2832085   31177   39512223
##   people_fully_vaccinated
## 1                    7270
## 2                    5400
## 3                     260
## 4                    8343
## 5                       8
## 6                  100089
```

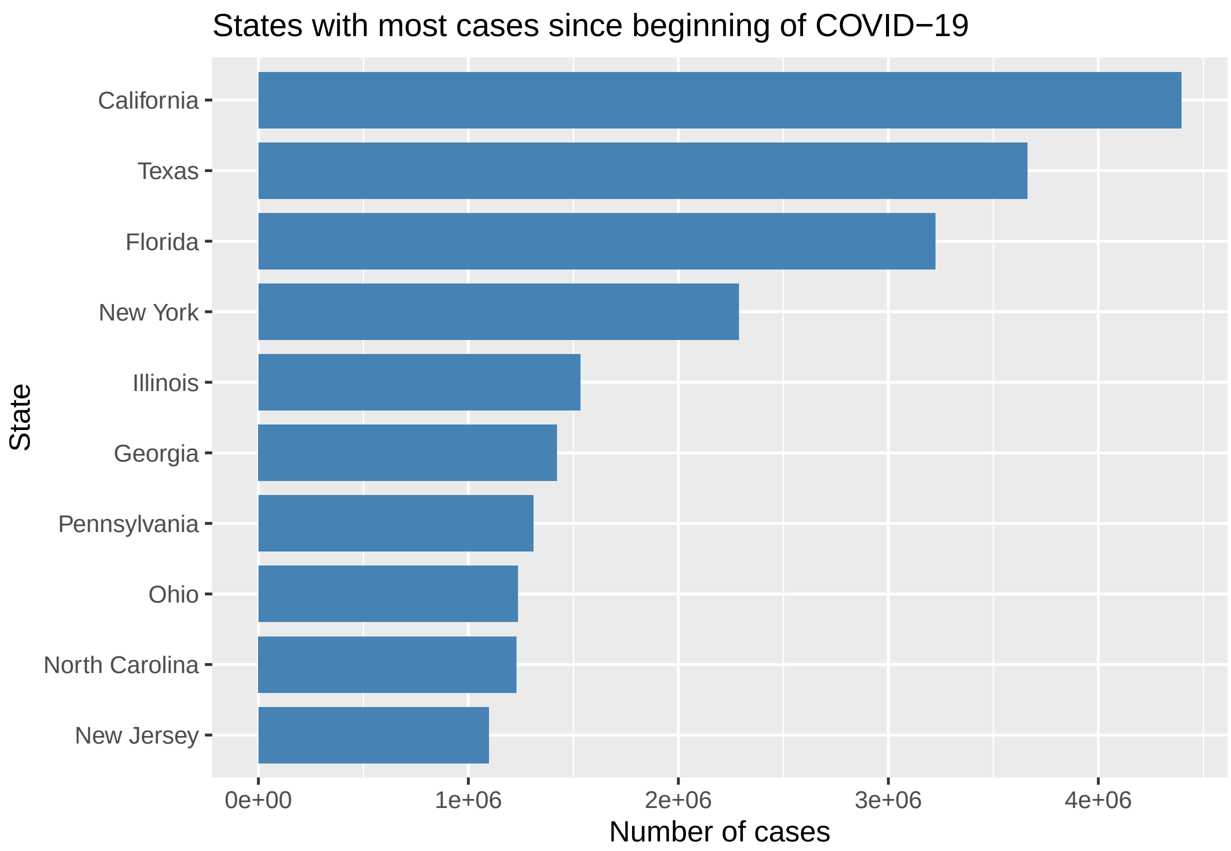
The data shows that first documented vaccination happened in January 2021. This information will help us later on to choose an appropriate time interval for comparing data from 2020 with data from 2021.
But first lets analyze which states had the highest numbers of cases and deaths since beginning of the pandemic.

5. Analyze the data of the states

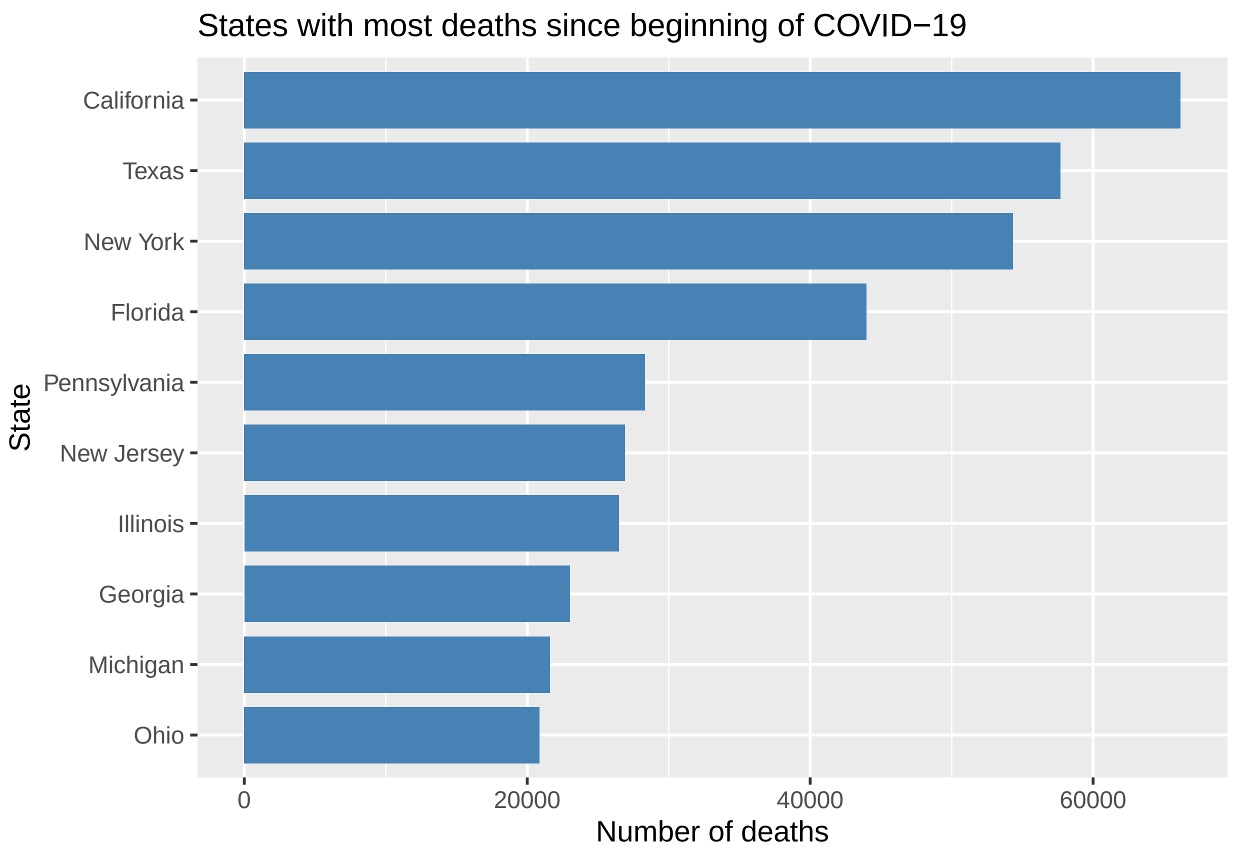
```
# get the cases and deaths for each state
# by using the last() method we take the latest value for cases and deaths
# which is already the accumulated value
US_total_by_state_acc <- US_total_by_state |>
  group_by(Province_State) |>
  summarize(cases = last(cases), deaths = last(deaths),
            population = max(Population)) |>
  filter(cases > 0, population > 0) |>
  ungroup()

head(US_total_by_state_acc)
```

```
## # A tibble: 6 x 4
##   Province_State   cases deaths population
##   <chr>            <dbl>  <dbl>      <dbl>
## 1 Alabama         709732  12330    4903185
## 2 Alaska           90060    448     740995
## 3 Arizona        1020133  18879    7278717
## 4 Arkansas        458234   7003    3017804
## 5 California     4394711  66182   39512223
## 6 Colorado        620268   7170    5758736
```

```
# plot states with most cases overall
US_total_by_state_acc |>
  slice_max(cases, n=10) |>
  ggplot(aes(x = cases, y = reorder(Province_State, cases))) +
  geom_bar(stat = "identity", fill = "steelblue", width = 0.8) +
  labs(title = "States with most cases since beginning of COVID-19",
       x = "Number of cases", y = "State") +
  theme(plot.title = element_text(size="12"))
```



```
# plot states with most deaths overall
US_total_by_state_acc |>
  slice_max(deaths, n=10) |>
  ggplot(aes(x = deaths, y = reorder(Province_State, deaths))) +
  geom_bar(stat = "identity", fill = "steelblue", width = 0.8) +
  labs(title = "States with most deaths since beginning of COVID-19",
       x = "Number of deaths", y = "State") +
  theme(plot.title = element_text(size="12"))
```



As we can see, California had both the most cases and deaths since beginning of COVID 19. Therefore we will take a closer look at this specific state.

6. Prepare the data for California

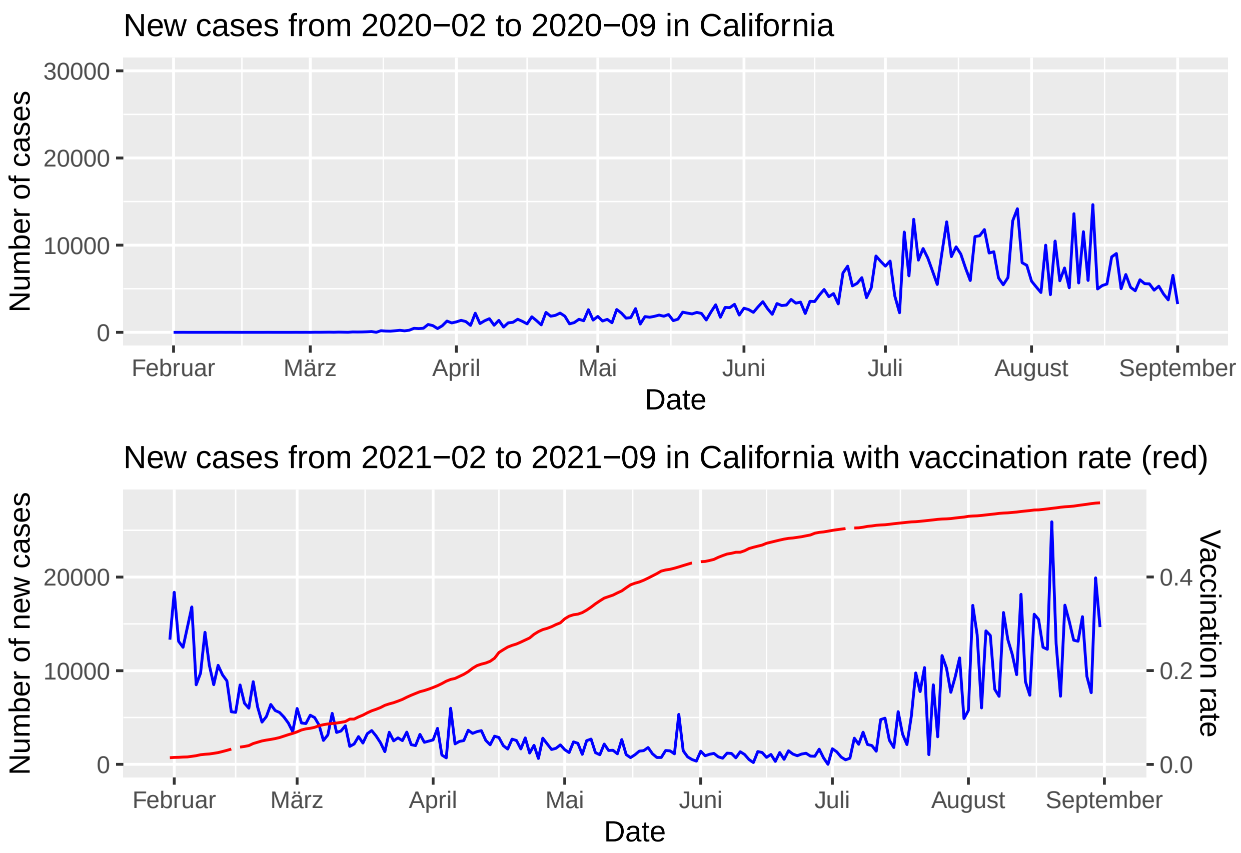
```
# prepare california data
US_California <- US_total_by_state |>
  merge(US_vaccinations, by=c("Province_State", "date"), all.x = TRUE) |>
  filter(Province_State == "California") |>
  mutate(new_cases = cases - lag(cases),
         new_deaths = deaths - lag(deaths),
         vaccination_ratio = people_fully_vaccinated / Population,
         death_ratio = deaths / cases) |>
  mutate(new_cases = ifelse(new_cases < 0, 0, new_cases)) |>
  mutate(new_deaths = ifelse(new_deaths < 0, 0, new_deaths)) |>
  select(-c(Country_Region, Province_State))

head(US_California)
```

```
##         date cases deaths Population people_fully_vaccinated new_cases
## 1 2020-01-22     0      0   39512223                      NA        NA
## 2 2020-01-23     0      0   39512223                      NA         0
## 3 2020-01-24     0      0   39512223                      NA         0
## 4 2020-01-25     0      0   39512223                      NA         0
## 5 2020-01-26     2      0   39512223                      NA         2
## 6 2020-01-27     2      0   39512223                      NA         0
##   new_deaths vaccination_ratio death_ratio
## 1         NA                NA         NaN
## 2          0                NA         NaN
## 3          0                NA         NaN
## 4          0                NA         NaN
## 5          0                NA           0
## 6          0                NA           0
```

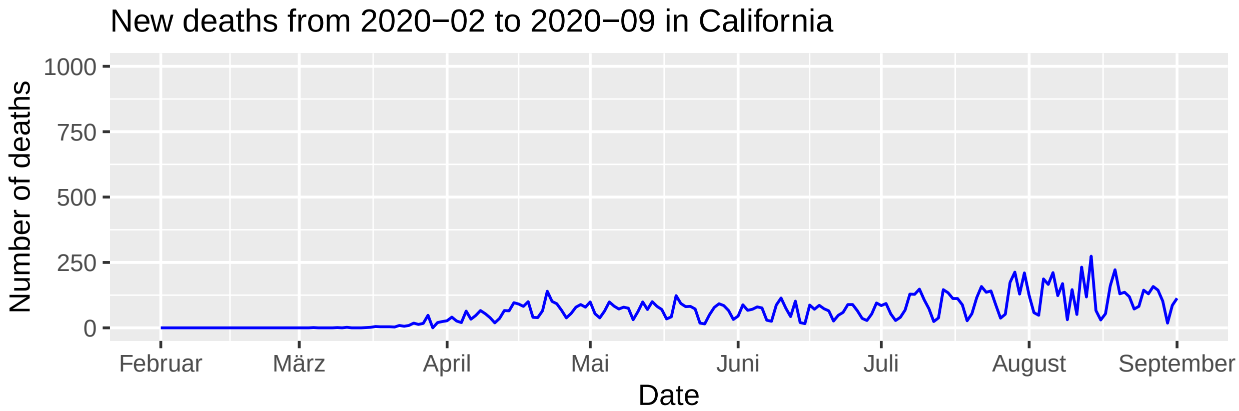
For the following observations I have choosen the time interval from February to September which I want to compare for the years 2020 and 2021. As we have seen, the vaccination data starts in January 2021. So we make sure we have a time interval in 2021 where vaccination data is fully available, that we can compare to an interval completely free of vaccination data in 2020.
Even though there is missing data in some of the columns, I won't be a problem for the further analysis. On reason is the choosing of the time intervals as explained above. The other reason is that I am only interested in qualitative analysis, where a few missing data point in between won't matter that much.

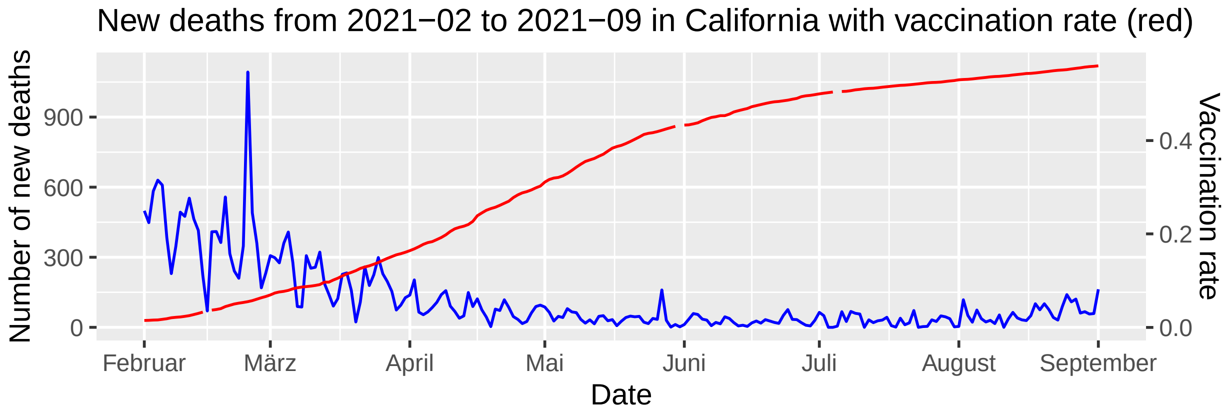
7. Show the data for California

```r
# compare cases of 2020 and 2021
plot1 <- US_California |>
  filter(date >= "2020-02-01" & date <= "2020-09-01") |>
  ggplot(aes(x = date)) +
  geom_line(aes(y = new_cases), color = "blue") +
  ylim(0, 30000) +
  scale_x_date(date_breaks = "1 month", date_labels = "%B") +
  labs(title = "New cases from 2020-02 to 2020-09 in California",
       x = "Date", y = "Number of cases") +
  theme(plot.title = element_text(size="12"))

plot2 <- US_California |>
  filter(date >= "2021-01-31" & date <= "2021-08-31") |>
  ggplot(aes(x = date)) +
  geom_line(aes(y = new_cases), color = "blue") +
  geom_line(aes(y = vaccination_ratio * 50000), color = "red") +
  scale_x_date(date_breaks = "1 month", date_labels = "%B") +
  scale_y_continuous(name = "Number of new cases",
                     sec.axis = sec_axis(~./50000, name = "Vaccination rate")) +
  labs(title = "New cases from 2021-02 to 2021-09 in California with vaccination rate (red)",
       x = "Date") +
  theme(plot.title = element_text(size="12"))

grid.arrange(plot1, plot2, ncol = 1, nrow = 2)
```



```
# compare deaths of 2020 and 2021
plot3 <- US_California |>
  filter(date >= "2020-02-01" & date <= "2020-09-01") |>
  ggplot(aes(x = date)) +
  geom_line(aes(y = new_deaths), color = "blue") +
  ylim(0, 1000) +
  scale_x_date(date_breaks = "1 month", date_labels = "%B") +
  labs(title = "New deaths from 2020-02 to 2020-09 in California",
       x = "Date", y = "Number of deaths") +
  theme(plot.title = element_text(size="12"))

plot4 <- US_California |>
  filter(date >= "2021-02-01" & date <= "2021-09-01") |>
  ggplot(aes(x = date)) +
  geom_line(aes(y = new_deaths), color = "blue") +
  geom_line(aes(y = vaccination_ratio * 2000), color = "red") +
  scale_x_date(date_breaks = "1 month", date_labels = "%B") +
  scale_y_continuous(name = "Number of new deaths",
                     sec.axis = sec_axis(~./2000, name = "Vaccination rate")) +
  labs(title = "New deaths from 2021-02 to 2021-09 in California with vaccination rate (red)",
       x = "Date") +
  theme(plot.title = element_text(size="12"))

grid.arrange(plot3, plot4, ncol = 1, nrow = 2)
```

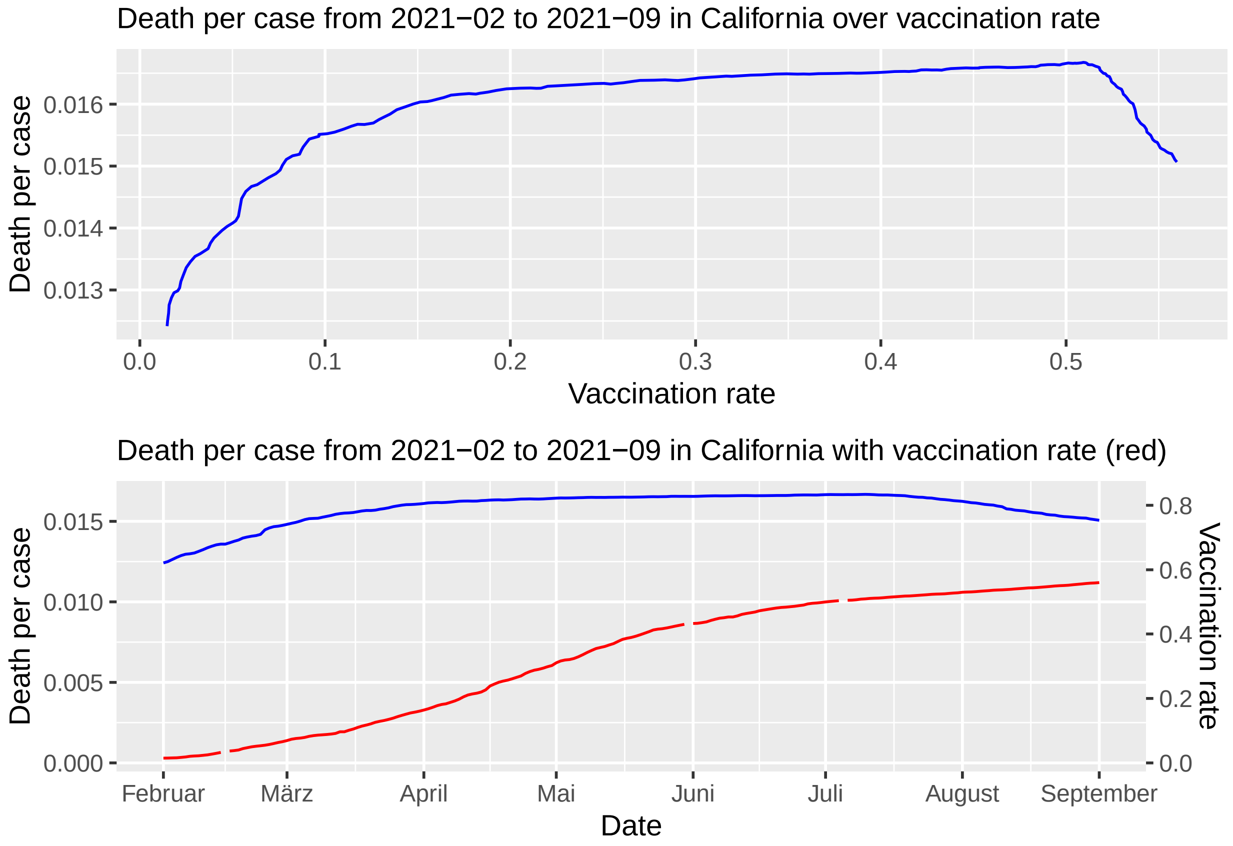




Looking at the cases, we see that in 2021 we have more new cases at the end of August / start of September than in 2020. Even though the vaccination rate is above 50% at this time. This can be an indicator that even fully vaccinated people can be infected with the virus, which is true as far as we know from current researches. Furthermore, a lot of circumstances changed during that 1 year period. For example new mutations of the virus exist now, that didn't exist in 2020. Those mutations can also be a reason why we have more new cases now, since they are probably more infectious.
Looking at the deaths, we see a kind of opposite behavior. Even though we have more cases, the number of deaths is about the same as last year, maybe even smaller. This shows, that the vaccination does not protect from getting infected though, but it is likely to prevent a severe course of disease in most cases.

Now I want to take a closer look at the relation between new cases and deaths.

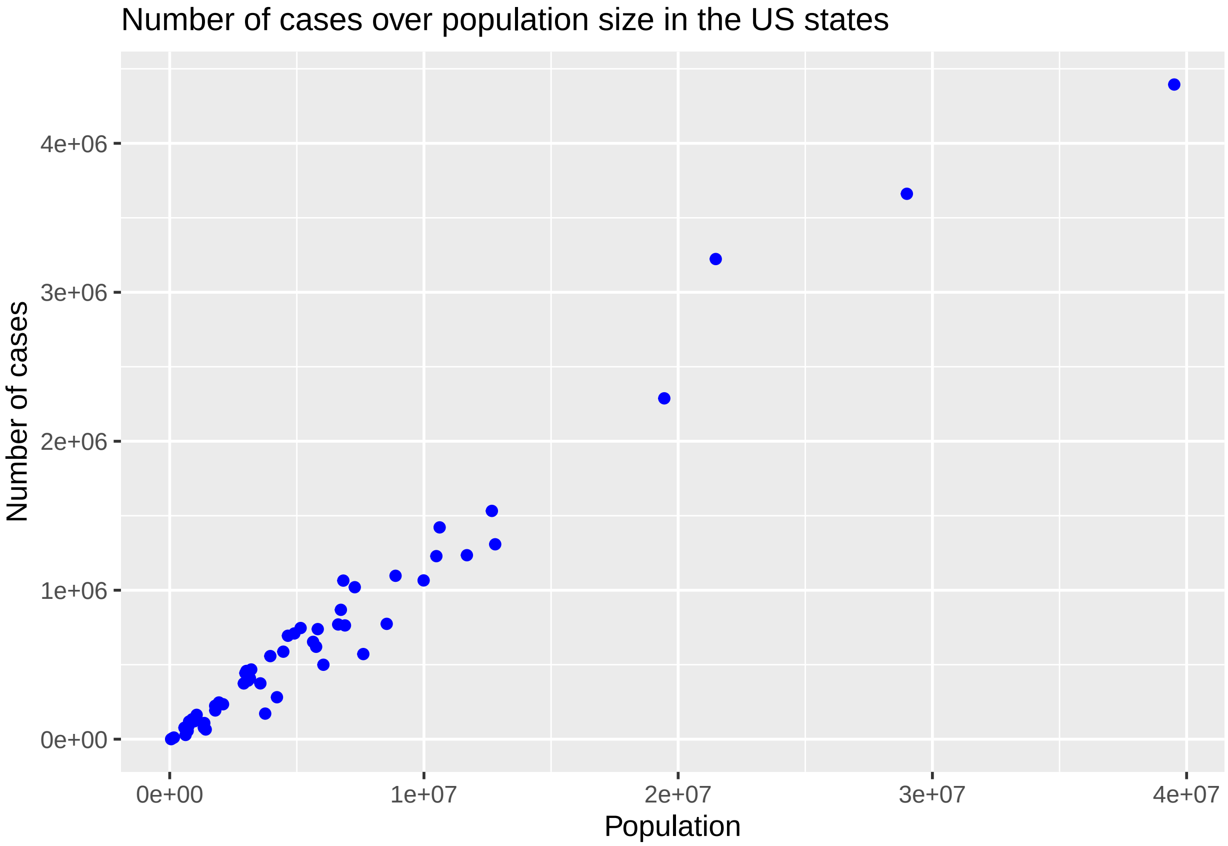
```
# analyze the death ratio with respect to the vaccination rate
plot5 <-  US_California |>
  filter(date >= "2021-02-01" & date <= "2021-09-01") |>
  ggplot(aes(x = vaccination_ratio)) +
  geom_line(aes(y = death_ratio), color = "blue") +
  scale_x_continuous(breaks = seq(0, 0.6, by = 0.1)) +
  labs(title = "Death per case from 2021-02 to 2021-09 in California over vaccination rate",
       x = "Vaccination rate", y = "Death per case") +
  theme(plot.title = element_text(size="11"))

# analyze the death and vaccination rate over time
plot6 <- US_California |>
  filter(date >= "2021-02-01" & date <= "2021-09-01") |>
  ggplot(aes(x = date)) +
  geom_line(aes(y = death_ratio), color = "blue") +
  geom_line(aes(y = vaccination_ratio / 50), color = "red") +
  scale_x_date(date_breaks = "1 month", date_labels = "%B") +
  scale_y_continuous(name = "Death per case",
                     sec.axis = sec_axis(~.*50, name = "Vaccination rate")) +
  labs(title = "Death per case from 2021-02 to 2021-09 in California with vaccination rate (red)",
       x = "Date", y = "Deaths per case") +
  theme(plot.title = element_text(size="11"))

grid.arrange(plot5, plot6, ncol = 1, nrow = 2)
```



In the first graph, we see that the number of deaths per case significantly declines as soon the vaccination rate reaches around 50%. This is also reflected in the bottom graph. The 50% vaccination rate was reached around July / August. At this time, the ratio of deaths per cases also shows a decline. Those insights can all be indicators that the vaccine helps to protect from severe courses of disease. But of course, to be sure, we would have to consider more features in this analysis.

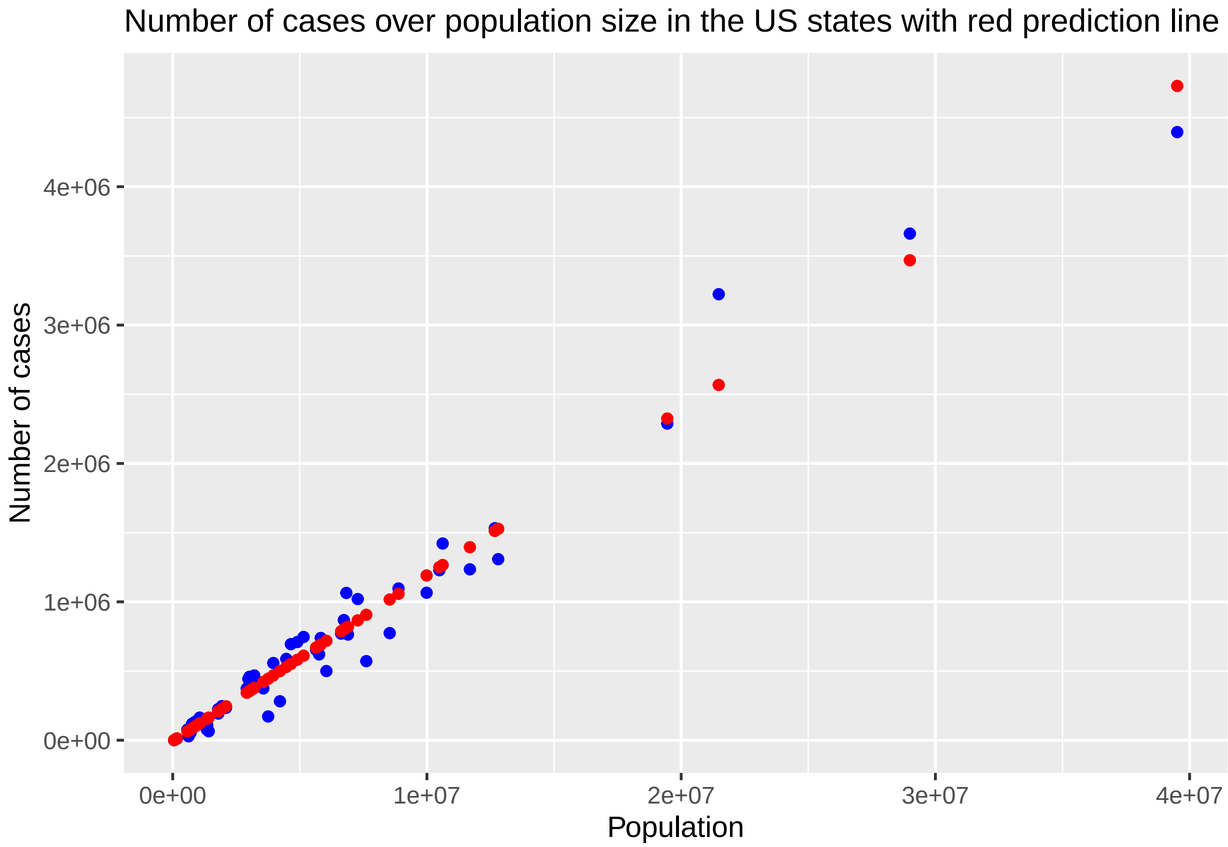
8. Analyze the relation between cases and population

```
# plot the cases of the population size
US_total_by_state_acc |>
  ggplot(aes(x = population)) +
  geom_point(aes(y = cases), color = "blue") +
  labs(title = "Number of cases over population size in the US states",
       x = "Population", y = "Number of cases") +
  theme(plot.title = element_text(size="12"))
```



The plots shows that the states with higher population are likely to have higher number of cases. Lets create a model that visualizes this assumption.

```
US_model <- lm(cases ~ population, data = US_total_by_state_acc)

# add a prediction column
US_total_by_state_acc <- US_total_by_state_acc |>
  mutate(cases_prediction = predict(US_model))

# visualize the prediction of murders
US_total_by_state_acc |> ggplot() +
  geom_point(aes(x = population, y = cases), color = "blue") +
  geom_point(aes(x = population, y = cases_prediction), color = "red") +
  labs(title = "Number of cases over population size in the US states with red prediction line",
       x = "Population", y = "Number of cases") +
  theme(plot.title = element_text(size="12"))
```



As supposed before we have quite linear relation between the number of cases and the total population of a state.

9. Conclusion and talking about bias

I hope this report gave you a solid overview over the COVID 19 Data. Of course not all aspects of the datasets were covered in this report. For example the global data was not covered at all in this analysis. Also we could have used more data from the vaccinations, like the type of the vaccine. This information surely would allow a deeper understanding of some results.

As with all data analysis, there is always some sort of bias. On the one hand, the data itself can be biased in such a way, that some records are not covered in the data set at all, due to certain (unknown) circumstances. On the other hand, the one who analysis the data can be biased. One example could be a bias in the presentation of the results, which can lead to a kind of controlled focus of the reader of the report. For example, if I would have made some of the plots visually much more noticeable than others, I can make the reader focus on facts that I want the reader to focus on. The avoid such bias, I tried to make all plots as equally appealing as possible.
Another source of bias can be the choice of the model. Since I already assumed a linearly increasing number of cases over the population according to my plot, my first choice for a model was also a linear one. At this point, one should dive in deeper and try out other types of models to see, if the initial assumption was appropriate. Also we could try to analyze the reasons for the existing variances between the data points and the model.

Session Info

```
## - Session info ---------------------------------------------------------------
##  setting  value
##  version  R version 4.1.1 (2021-08-10)
##  os       Windows 10 x64
##  system   x86_64, mingw32
##  ui       RTerm
##  language (EN)
##  collate  German_Germany.1252
##  ctype    German_Germany.1252
##  tz       Europe/Berlin
##  date     2021-09-03
##
## - Packages -------------------------------------------------------------------
##  package     * version date       lib source
##  assertthat    0.2.1   2019-03-21 [1] CRAN (R 4.1.0)
##  backports     1.2.1   2020-12-09 [1] CRAN (R 4.1.0)
##  bit           4.0.4   2020-08-04 [1] CRAN (R 4.1.0)
##  bit64         4.0.5   2020-08-30 [1] CRAN (R 4.1.0)
##  broom         0.7.9   2021-07-27 [1] CRAN (R 4.1.0)
##  cachem        1.0.5   2021-05-15 [1] CRAN (R 4.1.0)
##  callr         3.7.0   2021-04-20 [1] CRAN (R 4.1.0)
##  cellranger    1.1.0   2016-07-27 [1] CRAN (R 4.1.0)
##  cli           3.0.1   2021-07-17 [1] CRAN (R 4.1.0)
##  colorspace    2.0-2   2021-06-24 [1] CRAN (R 4.1.0)
##  crayon        1.4.1   2021-02-08 [1] CRAN (R 4.1.0)
##  curl          4.3.2   2021-06-23 [1] CRAN (R 4.1.0)
##  DBI           1.1.1   2021-01-15 [1] CRAN (R 4.1.0)
##  dbplyr        2.1.1   2021-04-06 [1] CRAN (R 4.1.0)
##  desc          1.3.0   2021-03-05 [1] CRAN (R 4.1.0)
##  devtools      2.4.2   2021-06-07 [1] CRAN (R 4.1.0)
##  digest        0.6.27  2020-10-24 [1] CRAN (R 4.1.0)
##  dplyr       * 1.0.7   2021-06-18 [1] CRAN (R 4.1.0)
##  ellipsis      0.3.2   2021-04-29 [1] CRAN (R 4.1.0)
##  evaluate      0.14    2019-05-28 [1] CRAN (R 4.1.0)
##  fansi         0.5.0   2021-05-25 [1] CRAN (R 4.1.0)
##  farver        2.1.0   2021-02-28 [1] CRAN (R 4.1.0)
##  fastmap       1.1.0   2021-01-25 [1] CRAN (R 4.1.0)
##  forcats     * 0.5.1   2021-01-27 [1] CRAN (R 4.1.0)
##  fs            1.5.0   2020-07-31 [1] CRAN (R 4.1.0)
##  generics      0.1.0   2020-10-31 [1] CRAN (R 4.1.0)
##  ggplot2     * 3.3.5   2021-06-25 [1] CRAN (R 4.1.0)
##  glue          1.4.2   2020-08-27 [1] CRAN (R 4.1.0)
##  gridExtra   * 2.3     2017-09-09 [1] CRAN (R 4.1.1)
##  gtable        0.3.0   2019-03-25 [1] CRAN (R 4.1.0)
##  haven         2.4.3   2021-08-04 [1] CRAN (R 4.1.0)
##  highr         0.9     2021-04-16 [1] CRAN (R 4.1.0)
##  hms           1.1.0   2021-05-17 [1] CRAN (R 4.1.0)
##  htmltools     0.5.1.1 2021-01-22 [1] CRAN (R 4.1.0)
##  httr          1.4.2   2020-07-20 [1] CRAN (R 4.1.0)
##  jsonlite      1.7.2   2020-12-09 [1] CRAN (R 4.1.0)
##  knitr         1.33    2021-04-24 [1] CRAN (R 4.1.0)
##  labeling      0.4.2   2020-10-20 [1] CRAN (R 4.1.0)
##  lattice       0.20-44 2021-05-02 [2] CRAN (R 4.1.1)
##  lifecycle     1.0.0   2021-02-15 [1] CRAN (R 4.1.0)
##  lubridate   * 1.7.10  2021-02-26 [1] CRAN (R 4.1.0)
##  magrittr      2.0.1   2020-11-17 [1] CRAN (R 4.1.0)
##  memoise       2.0.0   2021-01-26 [1] CRAN (R 4.1.0)
##  modelr        0.1.8   2020-05-19 [1] CRAN (R 4.1.0)
##  munsell       0.5.0   2018-06-12 [1] CRAN (R 4.1.0)
##  pillar        1.6.2   2021-07-29 [1] CRAN (R 4.1.0)
##  pkgbuild      1.2.0   2020-12-15 [1] CRAN (R 4.1.0)
##  pkgconfig     2.0.3   2019-09-22 [1] CRAN (R 4.1.0)
##  pkgload       1.2.1   2021-04-06 [1] CRAN (R 4.1.0)
##  prettyunits   1.1.1   2020-01-24 [1] CRAN (R 4.1.0)
##  processx      3.5.2   2021-04-30 [1] CRAN (R 4.1.0)
##  ps            1.6.0   2021-02-28 [1] CRAN (R 4.1.0)
##  purrr       * 0.3.4   2020-04-17 [1] CRAN (R 4.1.0)
##  R6            2.5.1   2021-08-19 [1] CRAN (R 4.1.1)
##  Rcpp          1.0.7   2021-07-07 [1] CRAN (R 4.1.0)
##  readr       * 2.0.1   2021-08-10 [1] CRAN (R 4.1.1)
##  readxl        1.3.1   2019-03-13 [1] CRAN (R 4.1.0)
##  remotes       2.4.0   2021-06-02 [1] CRAN (R 4.1.0)
##  reprex        2.0.1   2021-08-05 [1] CRAN (R 4.1.0)
##  rlang         0.4.11  2021-04-30 [1] CRAN (R 4.1.0)
##  rmarkdown     2.10    2021-08-06 [1] CRAN (R 4.1.1)
##  rprojroot     2.0.2   2020-11-15 [1] CRAN (R 4.1.0)
##  rstudioapi    0.13    2020-11-12 [1] CRAN (R 4.1.0)
##  rvest         1.0.1   2021-07-26 [1] CRAN (R 4.1.0)
##  scales        1.1.1   2020-05-11 [1] CRAN (R 4.1.0)
##  sessioninfo   1.1.1   2018-11-05 [1] CRAN (R 4.1.0)
##  stringi       1.7.3   2021-07-16 [1] CRAN (R 4.1.0)
##  stringr     * 1.4.0   2019-02-10 [1] CRAN (R 4.1.0)
##  testthat      3.0.4   2021-07-01 [1] CRAN (R 4.1.0)
##  tibble      * 3.1.3   2021-07-23 [1] CRAN (R 4.1.0)
##  tidyr       * 1.1.3   2021-03-03 [1] CRAN (R 4.1.0)
##  tidyselect    1.1.1   2021-04-30 [1] CRAN (R 4.1.0)
##  tidyverse   * 1.3.1   2021-04-15 [1] CRAN (R 4.1.0)
##  tzdb          0.1.2   2021-07-20 [1] CRAN (R 4.1.0)
##  usethis       2.0.1   2021-02-10 [1] CRAN (R 4.1.0)
##  utf8          1.2.2   2021-07-24 [1] CRAN (R 4.1.0)
##  vctrs         0.3.8   2021-04-29 [1] CRAN (R 4.1.0)
##  vroom         1.5.4   2021-08-05 [1] CRAN (R 4.1.0)
##  withr         2.4.2   2021-04-18 [1] CRAN (R 4.1.0)
##  xfun          0.25    2021-08-06 [1] CRAN (R 4.1.0)
##  xml2          1.3.2   2020-04-23 [1] CRAN (R 4.1.0)
##  yaml          2.2.1   2020-02-01 [1] CRAN (R 4.1.0)
##  zoo         * 1.8-9   2021-03-09 [1] CRAN (R 4.1.1)
##
## [1] C:/Users/Robin/Documents/R/win-library/4.1
## [2] C:/Program Files/R/R-4.1.1/library
```