Covid-19 Data Report

Deniz D.

2024-03-27

Introduction

In this report we aim to analyze the Covid-19 Data Set provided by Johns Hopkins University. This is part of the final project of the Data Science as a Field course.

We will load global and US data sets and also include the vaccination information. We will use the global set as the training data for our model and use the US data set as the testing data. Basically we will look at the relationship between number of deaths, cases and vaccination.

Data Source

More information about JHU Covid-19 data us available at https://coronavirus.jhu.edu/about/how-to-use-our-data

Both US and global covid-19 case and death information files can be found under: https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/

The file names are:

- time_series_covid19_confirmed_global.csv
- time_series_covid19_deaths_global.csv
- time_series_covid19_confirmed_US.csv
- time_series_covid19_deaths_US.csv

Population data can be fount at: https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/UID_ISO_FIPS_LookUp_Table.csv

Vaccination data for US is available at: https://raw.githubusercontent.com/govex/COVID-19/master/data_tables/vaccine_data/us_data/time_series/time_series_covid19_vaccine_us.csv

Global vaccination data can be found at: https://raw.githubusercontent.com/govex/COVID-19/master/data_tables/vaccine_data/global_data/time_series_covid19_vaccine_global.csv

Loading the Data

First, we load the necessary libraries.

```
library(tidyverse)
library(lubridate)
library(ggplot2)
library(dplyr)
library(caret)
```

Here, we specify the url and file names for the global and US data sets.

```
# common path to all files
url_in <-
      "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19

file_names <-
  c("time_series_covid19_confirmed_global.csv",
  "time_series_covid19_deaths_global.csv",
  "time_series_covid19_confirmed_US.csv",
  "time_series_covid19_deaths_US.csv")

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

Read in the data sets

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

Tidying and Transforming the Global Data Set

We will use the global data set as our training set.

First we pivot the global cases data set longer and remove the Lat, Long variables that we are not going to use.

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

We will do the same thing with the global deaths data.

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

Now we will join the global cases and deaths data sets. We convert the date to date type and we also filter out 0 cases.

```
global <- global_cases %>%
  full_join(global_deaths) %>%
  rename(Country_Region = 'Country/Region',
         Province_State = 'Province/State') %>%
  mutate(date = mdy(date)) %>%
      filter(cases > 0)
```

Let's check the global data set after our clean up.

```
summary(global)
```

```
##  Province_State     Country_Region          date                cases
##  Length:306827      Length:306827      Min.   :2020-01-22   Min.   :        1
```

```
##  Class :character   Class :character   1st Qu.:2020-12-12   1st Qu.:     1316
##  Mode  :character   Mode  :character   Median :2021-09-16   Median :    20365
##                                        Mean   :2021-09-11   Mean   :  1032863
##                                        3rd Qu.:2022-06-15   3rd Qu.:   271281
##                                        Max.   :2023-03-09   Max.   :103802702
##      deaths
##  Min.   :      0
##  1st Qu.:      7
##  Median :    214
##  Mean   :  14405
##  3rd Qu.:   3665
##  Max.   :1123836
```

Adding Population

Since there is no population information in the global data set we read in the population information.

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

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

We join the population data with the global data set.

```
global$Province_State[is.na(global$Province_State)] <- ""
global <- global %>%
      left_join(uid, by = c("Province_State", "Country_Region")) %>%
      select(-c(UID, FIPS)) %>%
      select(Province_State, Country_Region, date,
             cases, deaths, Population)
```

We sum the cases, deaths per country, per date and introduce the deaths_per_mill variable.

```
global_country <- global %>%
      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()
```

Adding Vaccination Information

From here onwards our analysis starts to differ from the lectures of Data Science as a Field Course.

Since vaccination information will be part of our model, we read in the vaccination data set.

```
vaccine_url <- "https://raw.githubusercontent.com/govex/COVID-19/master/data_tables/vaccine_data/global_
vaccine <- read.csv(vaccine_url, header=TRUE) %>%
      mutate(date = ymd(Date)) %>%
      select(-c(Date, Province_State, UID))
```

We join the vaccination data with the global country data set.

```
global_country <-  global_country %>%
      left_join(vaccine, by = c("Country_Region", "date"))
global_country$Doses_admin[is.na(global_country$Doses_admin)] = 0
```

We create new variables for new cases, new deaths and new doses administered by substracting the lags. There are some invalid entries in the data set, basically although cases are added up as time goes by, there is rarely a smaller number reported the next date. This will be handled by assigning 0 instead of a negative value for new cases etc. Also for the very first value for each country the lag would be NA, in that case again lag is not substracted.

```
global_country <- global_country %>%
      group_by(Country_Region) %>%
      mutate(new_cases = case_when(cases < lag(cases)  ~ 0,
                                   is.na(lag(cases)) ~ cases,
                                   cases >= lag(cases) ~ cases - lag(cases)),
             new_deaths = case_when(deaths < lag(deaths) ~ 0,
                                    is.na(lag(deaths)) ~ deaths,
                                    deaths >= lag(deaths) ~ deaths - lag(deaths)),
             new_doses_admin = case_when(Doses_admin < lag(Doses_admin) ~ 0,
                                         is.na(lag(Doses_admin)) ~ Doses_admin,
                                         Doses_admin >= lag(Doses_admin) ~
                                               Doses_admin - lag(Doses_admin))) %>%
      ungroup()
```

Let's check our global data set by country, global_country:

```
summary(global_country)
```

```
##  Country_Region          date                cases               deaths
##  Length:214113      Min.   :2020-01-22   Min.   :        1   Min.   :      0
##  Class :character   1st Qu.:2020-12-15   1st Qu.:     7504   1st Qu.:     98
##  Mode  :character   Median :2021-09-18   Median :    71705   Median :   1061
##                     Mean   :2021-09-13   Mean   :  1480108   Mean   :  20642
##                     3rd Qu.:2022-06-16   3rd Qu.:   579110   3rd Qu.:   8357
##                     Max.   :2023-03-09   Max.   :103802702   Max.   :1123836
##
##  deaths_per_mill     Population         Doses_admin
##  Min.   :   0.00   Min.   :8.090e+02   Min.   :0.000e+00
##  1st Qu.:  20.75   1st Qu.:2.083e+06   1st Qu.:0.000e+00
##  Median : 183.99   Median :9.006e+06   Median :4.887e+05
##  Mean   : 713.88   Mean   :3.413e+07   Mean   :3.048e+07
##  3rd Qu.:1059.93   3rd Qu.:2.914e+07   3rd Qu.:8.266e+06
##  Max.   :6658.38   Max.   :1.418e+09   Max.   :3.491e+09
##  NA's   :5861      NA's   :5861
##  People_at_least_one_dose   new_cases         new_deaths
##  Min.   :0.000e+00        Min.   :      0   Min.   :    0.00
##  1st Qu.:3.300e+05        1st Qu.:      0   1st Qu.:    0.00
##  Median :2.064e+06        Median :     38   Median :    0.00
##  Mean   :2.123e+07        Mean   :   3164   Mean   :   32.18
##  3rd Qu.:8.355e+06        3rd Qu.:    660   3rd Qu.:    7.00
##  Max.   :1.310e+09        Max.   :1354505   Max.   :59961.00
##  NA's   :73919
##  new_doses_admin
##  Min.   :        0
##  1st Qu.:        0
##  Median :        0
##  Mean   :    63434
##  3rd Qu.:      113
##  Max.   :225063079
```

##

Tidying and Transforming the US Data Set

We will use the US data set as our testing data.

We start by pivoting US_cases data set longer and removing variables that we are not going to use. We transform the date to date type.

```
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_))
```

We will do the same for the US_deaths dat set,

```
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_))
```

Now we will join the US_cases and US_deaths sets.

```
US <- US_cases %>%
  full_join(US_deaths)
```

Let's check US data set after our clean up.

```
summary(US)
```

```
##     Admin2          Province_State     Country_Region     Combined_Key
##  Length:3819906     Length:3819906     Length:3819906     Length:3819906
##  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.:2020-11-02   1st Qu.:    330   1st Qu.:    9917   1st Qu.:    4.0
##  Median :2021-08-15   Median :   2272   Median :   24892   Median :   37.0
##  Mean   :2021-08-15   Mean   :  14088   Mean   :   99604   Mean   :  186.9
##  3rd Qu.:2022-05-28   3rd Qu.:   8159   3rd Qu.:   64979   3rd Qu.:  122.0
##  Max.   :2023-03-09   Max.   :3710586   Max.   :10039107   Max.   :35545.0
```

We will group the US data set by states and add sum of cases and deaths and also new cases and new deaths. Just as in the global case, invalid entries and first cases are handled by assigning 0.

```
US_by_state <- US %>%
      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()

US_by_state <- US_by_state %>%
      mutate(new_cases = case_when(cases < lag(cases)  ~ 0,
                                   is.na(lag(cases)) ~ cases,
                                   cases >= lag(cases) ~ cases - lag(cases)),
             new_deaths = case_when(deaths < lag(deaths) ~ 0,
                                    is.na(lag(deaths)) ~ deaths,
                                    deaths >= lag(deaths) ~ deaths - lag(deaths)))
```

Adding Vaccination Information

Here we get the vaccination information for US

```
vac_url <- "https://raw.githubusercontent.com/govex/COVID-19/master/data_tables/vaccine_data/us_data/ti
vaccine_US <- read.csv(vac_url, header=TRUE) %>%
      mutate(date = ymd(Date)) %>%
      select(-c(Date, UID, Country_Region))
```

We join the US vaccination data with the US_by_state data.

```
US_by_state <-  US_by_state %>%
      left_join(vaccine_US, by = c("Province_State", "date"))
```

We add the variable new_doses_admin for new doses administered each day.

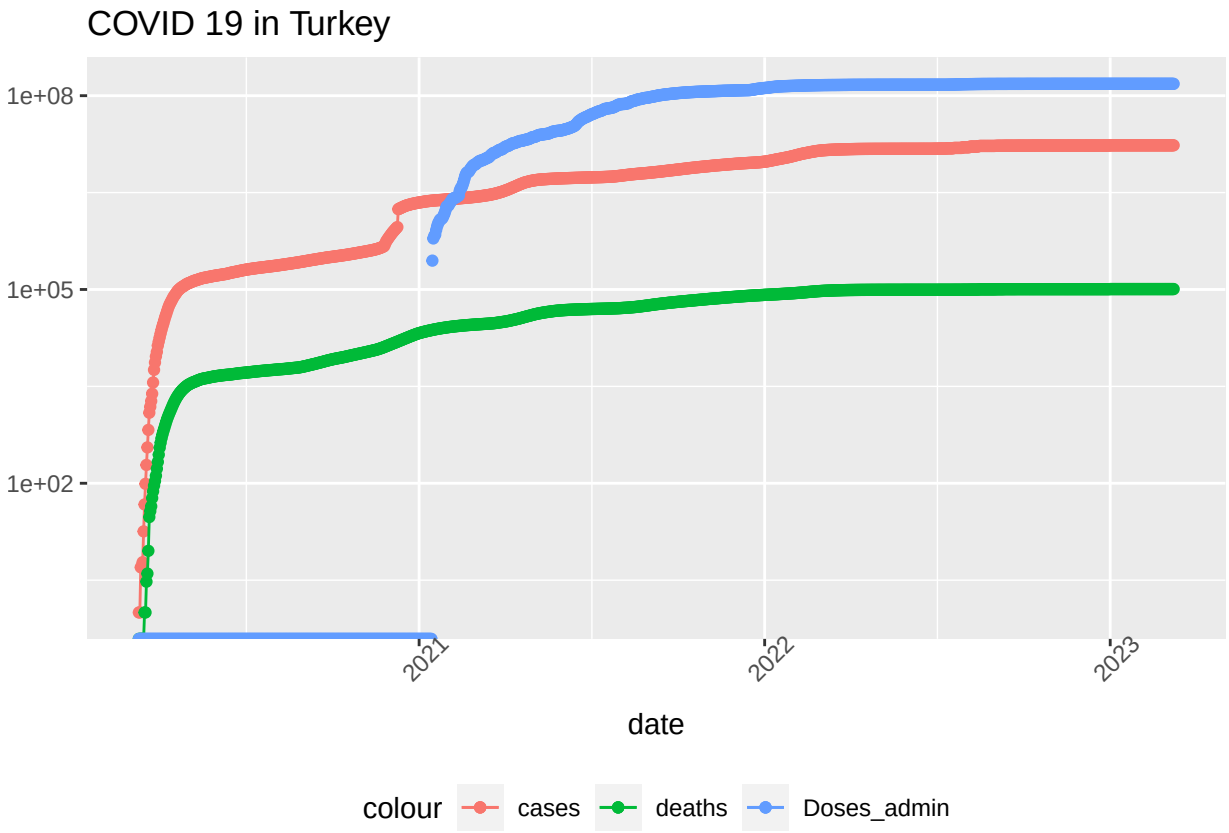
```
US_by_state$Doses_admin[is.na(US_by_state$Doses_admin)] = 0

US_by_state <- US_by_state %>%
      mutate(new_doses_admin = case_when(Doses_admin < lag(Doses_admin) ~ 0,
                                         is.na(lag(Doses_admin)) ~ Doses_admin,
                                         Doses_admin >= lag(Doses_admin) ~
                                               Doses_admin - lag(Doses_admin))) %>%
      filter(cases > 0)
```

Exploring the Data

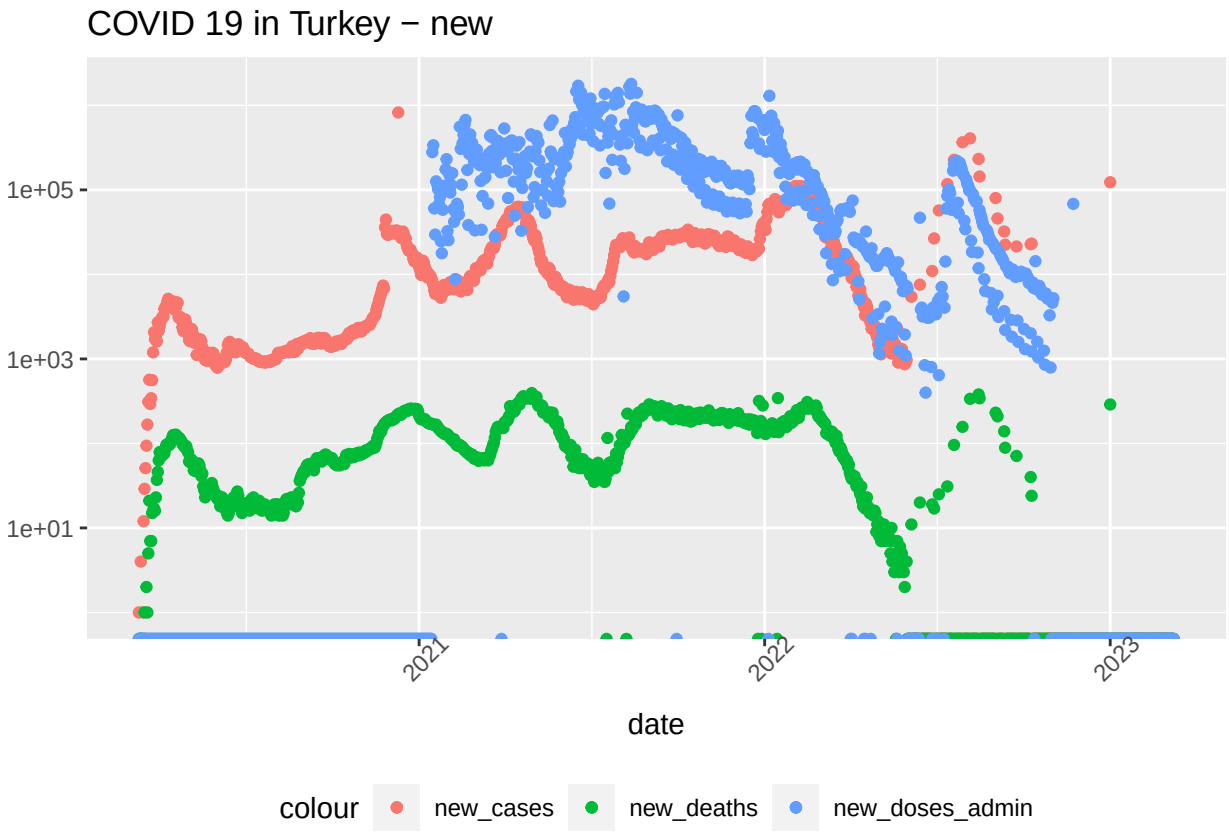
We take a look at our data by selecting a country (Turkey) and plotting cases, deaths and vaccine doses administered per date.

```
country <- "Turkey"
global_country %>%
      filter(Country_Region == country) %>%
      ggplot(aes(x = date, y = cases)) +
      geom_line(aes(y = cases, color = "cases")) +
      geom_point(aes(y = cases, color = "cases")) +
      geom_line(aes(y = deaths, color = "deaths")) +
      geom_point(aes(y=deaths, color="deaths")) +
      geom_point(aes(y= Doses_admin, color = "Doses_admin")) +
      scale_y_log10() +
      theme(legend.position = "bottom",
            axis.text.x = element_text(angle = 45)) +
      labs(title="COVID 19 in Turkey", y = NULL)
```



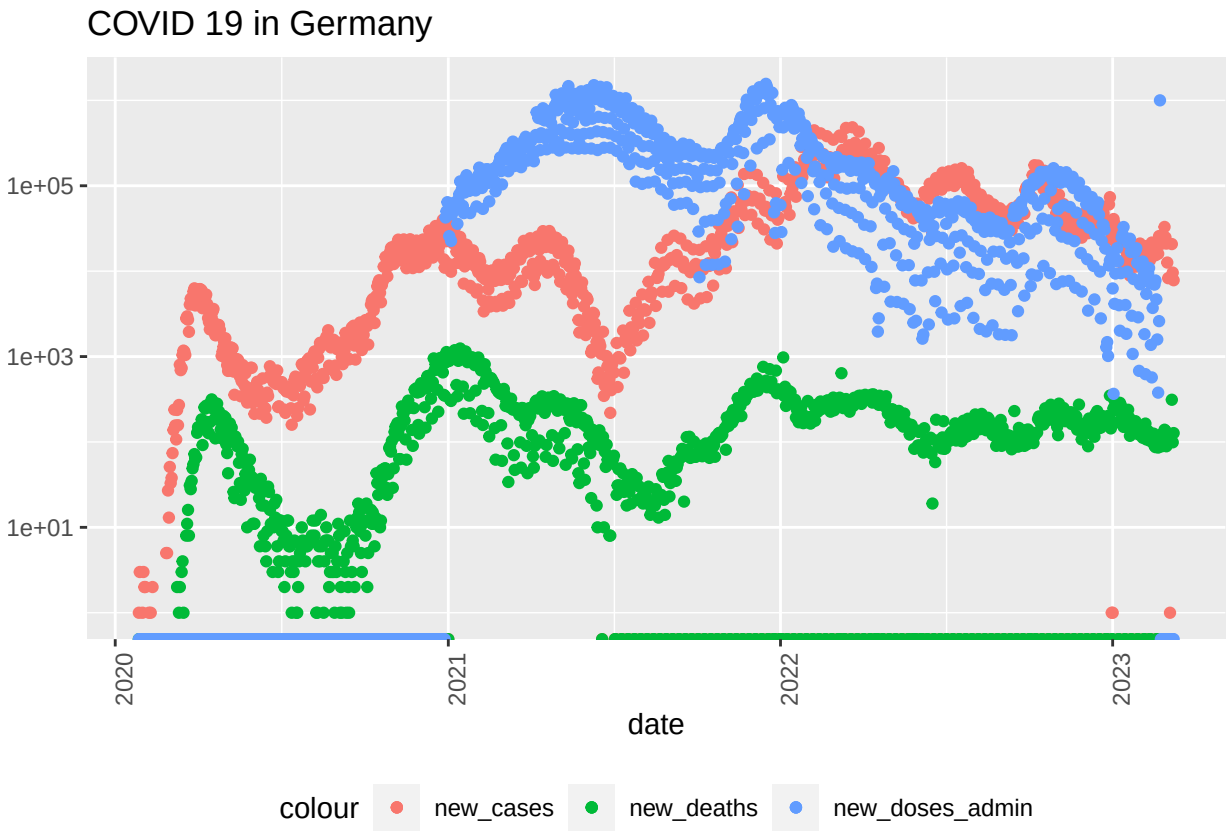
To get a more detailed picture, we will take a look at new cases, new deaths and new doses administered. We can see surprising parallels.

```
country <- "Turkey"
global_country %>%
      filter(Country_Region == country) %>%
      ggplot(aes(x = date, y = new_cases)) +
      geom_point(aes(color = "new_cases")) +
      geom_point(aes(y= new_deaths, color="new_deaths")) +
      geom_point(aes(y = new_doses_admin, color = "new_doses_admin")) +
      scale_y_log10() +
      theme(legend.position = "bottom",
            axis.text.x = element_text(angle = 45)) +
      labs(title="COVID 19 in Turkey - new", y = NULL)
```



Let's check for another country (Germany) if we see a similar picture. There seems a strong relationship between number of new deaths, new cases and new vaccinations.

```
country <- "Germany"
global_country %>%
      filter(Country_Region == country) %>%
      ggplot(aes(x = date, y = new_cases)) +
      geom_point(aes(color = "new_cases")) +
      geom_point(aes(y=new_deaths, color="new_deaths")) +
      geom_point(aes(y=new_doses_admin, color= "new_doses_admin")) +
      scale_y_log10() +
      theme(legend.position = "bottom",
            axis.text.x = element_text(angle = 90)) +
      labs(title="COVID 19 in Germany", y = NULL)
```



Modeling the Data

We will fit a generalized linear model. We will use the global by country data (global_country) as our training set. We will try to see the relationship between new deaths, new cases and new doses administered.

```
cofit <- glm(new_deaths ~ new_cases + new_doses_admin, data = global_country, na.action = na.omit)

summary(cofit)
```

```
##
## Call:
## glm(formula = new_deaths ~ new_cases + new_doses_admin, data = global_country,
##     na.action = na.omit)
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.650e+01  4.213e-01   39.16   <2e-16 ***
## new_cases       4.680e-03  2.252e-05  207.76   <2e-16 ***
## new_doses_admin 1.377e-05  5.603e-07   24.57   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 36753.91)
##
##     Null deviance: 9529947232  on 214112  degrees of freedom
## Residual deviance: 7869380408  on 214110  degrees of freedom
## AIC: 2858387
##
```

```
## Number of Fisher Scoring iterations: 2
```

Testing the Model

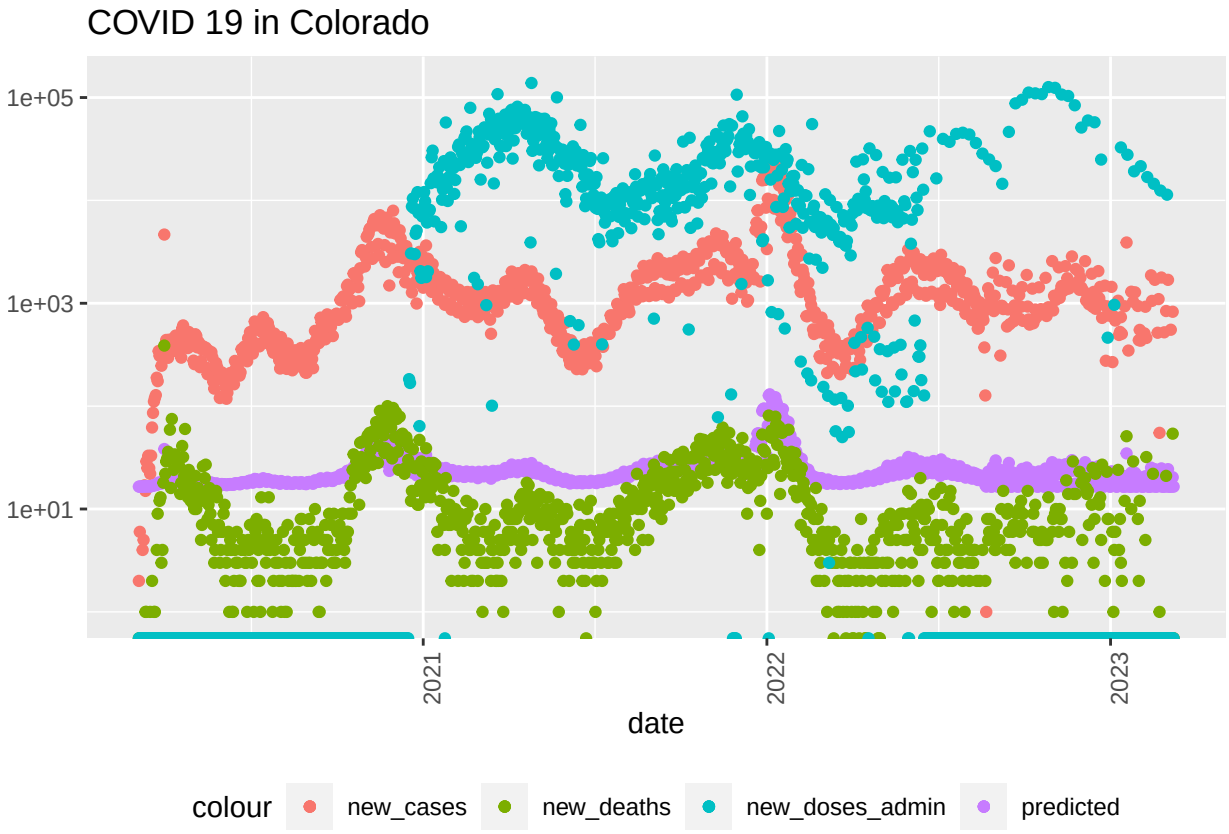
We will use the US_by_state as our testing data set.

Let's see how our model performs for Colorado. The predicted new deaths are in color purple.

```
state <- "Colorado"
fstate <- US_by_state %>%
      filter(Province_State == state)

predicted = predict(cofit, fstate)

fstate %>%
      ggplot(aes(x = date, y = new_cases)) +
      geom_point(aes(color = "new_cases")) +
      geom_point(aes(y = predicted, color="predicted")) +
      geom_point(aes(y=new_deaths, color="new_deaths")) +
      geom_point(aes(y=new_doses_admin, color= "new_doses_admin")) +
      scale_y_log10() +
      theme(legend.position = "bottom",
            axis.text.x = element_text(angle = 90)) +
      labs(title="COVID 19 in Colorado", y = NULL)
```



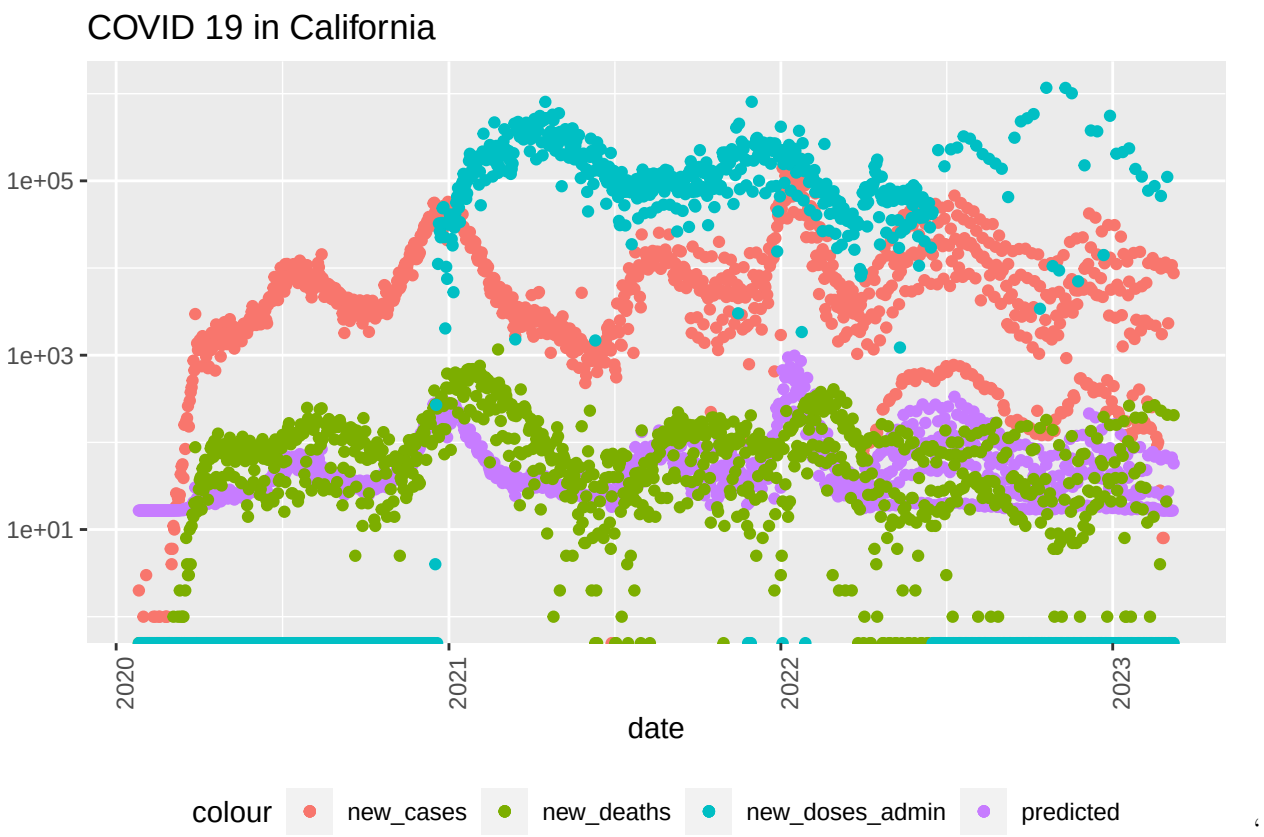
Next we will try our model on California data. Here our model performed better.

```
state <- "California"
fstate <- US_by_state %>%
      filter(Province_State == state) %>%
```

```
      filter(cases > 0)

predicted = predict(cofit, fstate)

fstate %>%
      ggplot(aes(x = date, y = new_cases)) +
      geom_point(aes(color = "new_cases")) +
      geom_point(aes(y = predicted, color="predicted")) +
      geom_point(aes(y=new_deaths, color="new_deaths")) +
      geom_point(aes(y=new_doses_admin, color= "new_doses_admin")) +
      scale_y_log10() +
      theme(legend.position = "bottom",
            axis.text.x = element_text(angle = 90)) +
      labs(title="COVID 19 in California", y = NULL)
```



Now we will apply our model to the entire US_by_state data set. Our model performance metrics are not that impressive.

```
predict_US <- predict(cofit, US_by_state)
# model performance metrics
data.frame(R2 = R2(predict_US, US_by_state$new_deaths),
        RMSE = RMSE(predict_US, US_by_state$new_deaths),
        MAE = MAE(predict_US, US_by_state$new_deaths))
```

```
##          R2     RMSE      MAE
## 1 0.1717031 46.49944 21.90667
```

Conclusion

The data we focused on mostly, namely new cases, new deaths and new vaccine doses administered follow an interesting pattern and have many ups and downs. Maybe some of these can be explained by new variants of covid, as those spread people might have gone for vaccination.

While our model showed some initial hope and reasonably good match in some cases it fell short overall to provide overarching explanation of the patterns.

There are probably much better ways to model infectious diseases other then linear models. As future improvement suggestion different models can be explored.

Potential Sources of Bias

There is potential bias in the reporting phase of all data. Sometimes even political motivations cause states or countries under report cases or deaths. Also deaths can be under reported by not testing for covid-19.

I also have personal biases regarding covid-19. There is so much misinformation and conspiracy theories surrounding covid-19 that is a personal pet-peeve of mine. My personal biases probably affected even which variables I chose to work with.

Appendix

```
sessionInfo()
```

```
## R version 4.3.1 (2023-06-16)
## Platform: aarch64-apple-darwin20 (64-bit)
## Running under: macOS Sonoma 14.4
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.3-arm64/Resources/lib/libRlapack.dylib;  LAPACK v
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## time zone: Europe/Istanbul
## tzcode source: internal
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] caret_6.0-94    lattice_0.22-5  lubridate_1.9.2 forcats_1.0.0
##  [5] stringr_1.5.1   dplyr_1.1.4     purrr_1.0.2     readr_2.1.4
##  [9] tidyr_1.3.0     tibble_3.2.1    ggplot2_3.4.4   tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
##  [1] gtable_0.3.4         xfun_0.41            recipes_1.0.8
##  [4] tzdb_0.4.0           vctrs_0.6.5          tools_4.3.1
##  [7] generics_0.1.3       curl_5.2.0           stats4_4.3.1
## [10] parallel_4.3.1       fansi_1.0.6          highr_0.10
## [13] ModelMetrics_1.2.2.2 pkgconfig_2.0.3      Matrix_1.6-3
## [16] data.table_1.14.10   lifecycle_1.0.4      farver_2.1.1
## [19] compiler_4.3.1       munsell_0.5.0        codetools_0.2-19
## [22] htmltools_0.5.7      class_7.3-22         yaml_2.3.8
```

```
## [25] prodlim_2023.08.28   crayon_1.5.2          pillar_1.9.0
## [28] MASS_7.3-60          gower_1.0.1           iterators_1.0.14
## [31] rpart_4.1.19         foreach_1.5.2         parallelly_1.36.0
## [34] nlme_3.1-162         lava_1.7.3            tidyselect_1.2.0
## [37] digest_0.6.33        stringi_1.8.3         future_1.33.0
## [40] reshape2_1.4.4       listenv_0.9.0         splines_4.3.1
## [43] fastmap_1.1.1        grid_4.3.1            colorspace_2.1-0
## [46] cli_3.6.2            magrittr_2.0.3        survival_3.5-5
## [49] utf8_1.2.4           future.apply_1.11.0   withr_2.5.2
## [52] scales_1.3.0         bit64_4.0.5           timechange_0.2.0
## [55] rmarkdown_2.25       globals_0.16.2        bit_4.0.5
## [58] nnet_7.3-19          timeDate_4022.108     hms_1.1.3
## [61] evaluate_0.23        knitr_1.45            hardhat_1.3.0
## [64] rlang_1.1.2          Rcpp_1.0.11           glue_1.6.2
## [67] pROC_1.18.5          ipred_0.9-14          vroom_1.6.5
## [70] rstudioapi_0.15.0    R6_2.5.1              plyr_1.8.9
```