

COVID-19 Vaccine Data Analysis

KCS

2/7/2022

COVID-19 Vaccine Data Analysis

In this report, we will be conduct an analysis of the publicly available COVID-19 vaccination data.

The primary goal of this analysis is to gain deeper insight into the relationship between vaccine administration and the number of new deaths from COVID-19. By analyzing trends over time, we aim to better understand the potential impact of vaccination efforts on reducing mortality rates during the pandemic.

First, please make sure to install and library() the packages below. These are essential to being able to reproduce this analysis in rmarkdown.

```
library(prettydoc)
library(tidyverse)
```

```
## — Attaching core tidyverse packages ————— tidyverse
## ✓ dplyr      1.1.3      ✓ readr      2.1.4
## ✓ forcats    1.0.0      ✓ stringr    1.5.0
## ✓ ggplot2    3.4.3      ✓ tibble     3.2.1
## ✓ lubridate  1.9.2      ✓ tidyr      1.3.0
## ✓ purrr      1.0.2
## — Conflicts ————— tidyverse
## ✗ dplyr::filter() masks stats::filter()
## ✗ dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) -
```

```
library(lubridate)
library(dplyr)
```

First we will perform exploratory analysis on the data and then do some statistical modelling for vaccine and new death cases

Locate, copy the file names and read the urls from the COVID public data from github

```
url_in<-"https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/csse_covid_19_time_series/time_series_covid19_confirmed_global.csv"
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 datasets
global_cases<-read_csv(urls[1])
```

```
## Rows: 289 Columns: 1147
## — Column specification —————
## Delimiter: ","
## chr      (2): Province/State, Country/Region
## dbl (1145): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20
##
## i Use `spec()` to retrieve the full column specification for this data
## i Specify the column types or set `show_col_types = FALSE` to silence this message
```

```
global_deaths<-read_csv(urls[2])
```

```
## Rows: 289 Columns: 1147
## — Column specification —————
## Delimiter: ","
## chr      (2): Province/State, Country/Region
## dbl (1145): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20
##
```

```
## i Use `spec()`` to retrieve the full column specification for the data
## i Specify the column types or set `show_col_types = FALSE` to suppress
```

```
US_cases<-read_csv(urls[3])
```

```
## Rows: 3342 Columns: 1154
## — Column specification —————
## Delimiter: ","
## chr (6): iso2, iso3, Admin2, Province_State, Country_Region,
## dbl (1148): UID, code3, FIPS, Lat, Long_, 1/22/20, 1/23/20, 1/24/20,
##
## i Use `spec()`` to retrieve the full column specification for the data
## i Specify the column types or set `show_col_types = FALSE` to suppress
```

```
US_deaths<-read_csv(urls[4])
```

```
## Rows: 3342 Columns: 1155
## — Column specification —————
## Delimiter: ","
## chr (6): iso2, iso3, Admin2, Province_State, Country_Region,
## dbl (1149): UID, code3, FIPS, Lat, Long_, Population, 1/22/20,
##
## i Use `spec()`` to retrieve the full column specification for the data
## i Specify the column types or set `show_col_types = FALSE` to suppress
```

Tidying and data wrangling

To keep things simple, we will focus on US cases only for this analysis. Join cases and deaths data together and get US data by state and US totals

```
##read datasets
#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_))
##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_))

##fulljoin cases and deaths
US<-US_cases %>%
  full_join(US_deaths)

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()

US_totals<- US_by_state %>%
  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()

```

Exploratory Analysis and Visualization

```

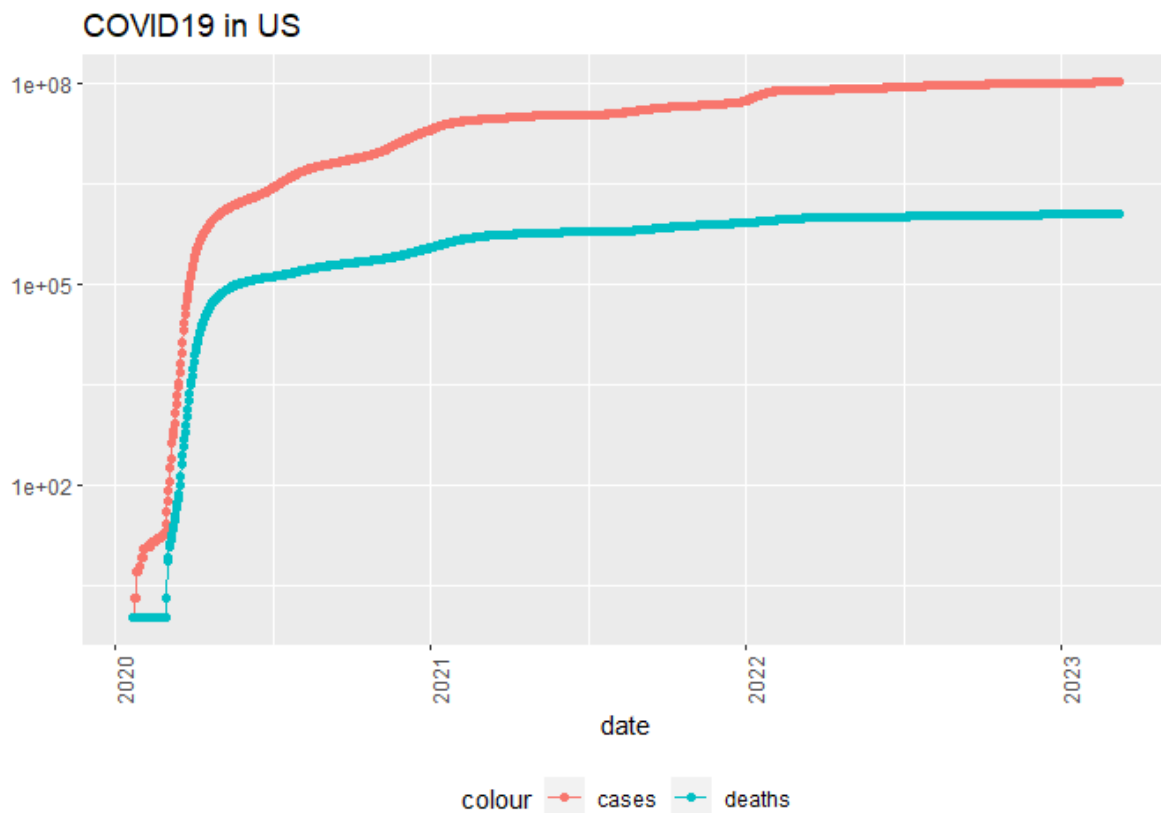
##Let's visualize our data and analyze it
US_totals %>%

```

```

filter(cases>0) %>%
ggplot(aes(x=date, y=cases))+
geom_line(aes(color="cases"))+
geom_point(aes(color="cases"))+
geom_line(aes(y=deaths, color="deaths"))+
geom_point(aes(y=deaths, color="deaths"))+
scale_y_log10()+
theme(legend.position="bottom",
      axis.text.x=element_text(angle = 90))+
labs(title = "COVID19 in US",y=NULL)

```

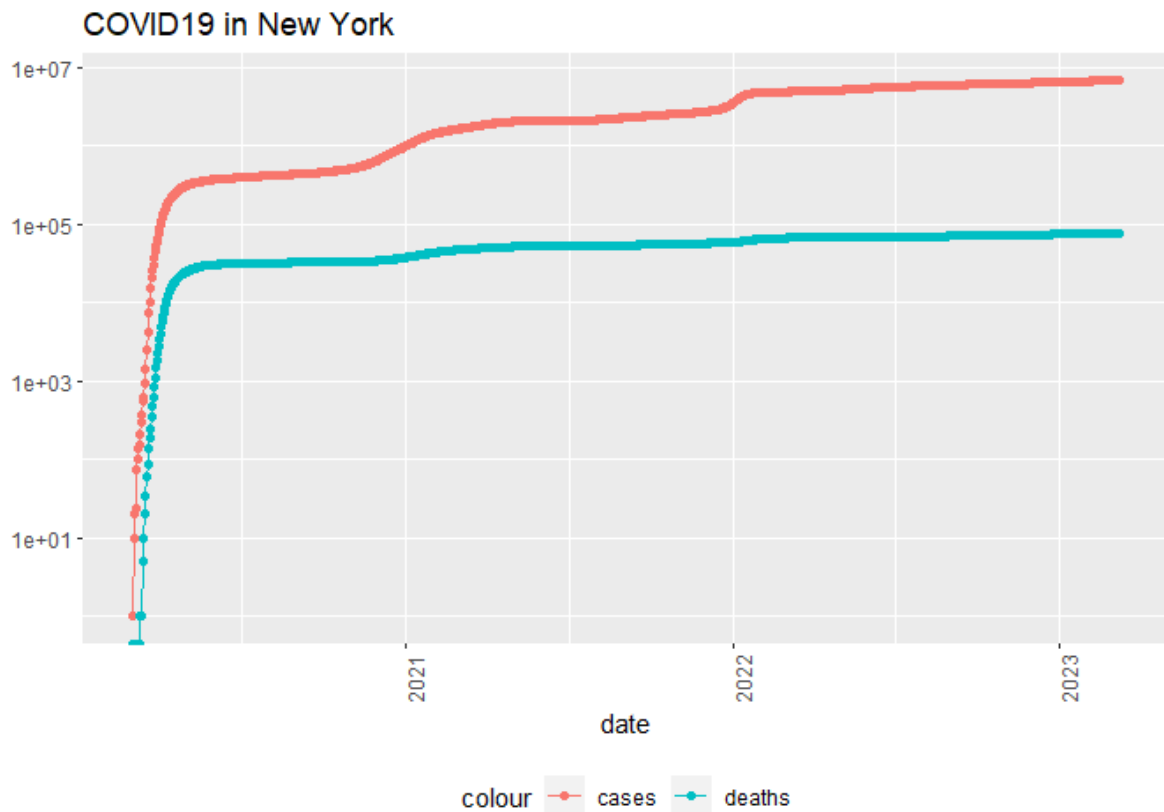


```

##Let's filter by the state of New York and visualize cas
state<- "New York"
US_by_state %>%
  filter(Province_State== state) %>%
  filter(cases>0) %>%
  ggplot(aes(x=date, y=cases))+
  geom_line(aes(color="cases"))+
  geom_point(aes(color="cases"))+
  geom_line(aes(y=deaths, color="deaths"))+
  geom_point(aes(y=deaths, color="deaths"))+

```

```
scale_y_log10()+
  theme(legend.position="bottom",
        axis.text.x=element_text(angle = 90))+
  labs(title = str_c("COVID19 in ", state),y=NULL)
```



```
##check what is the max death
max(US_totals$deaths)
```

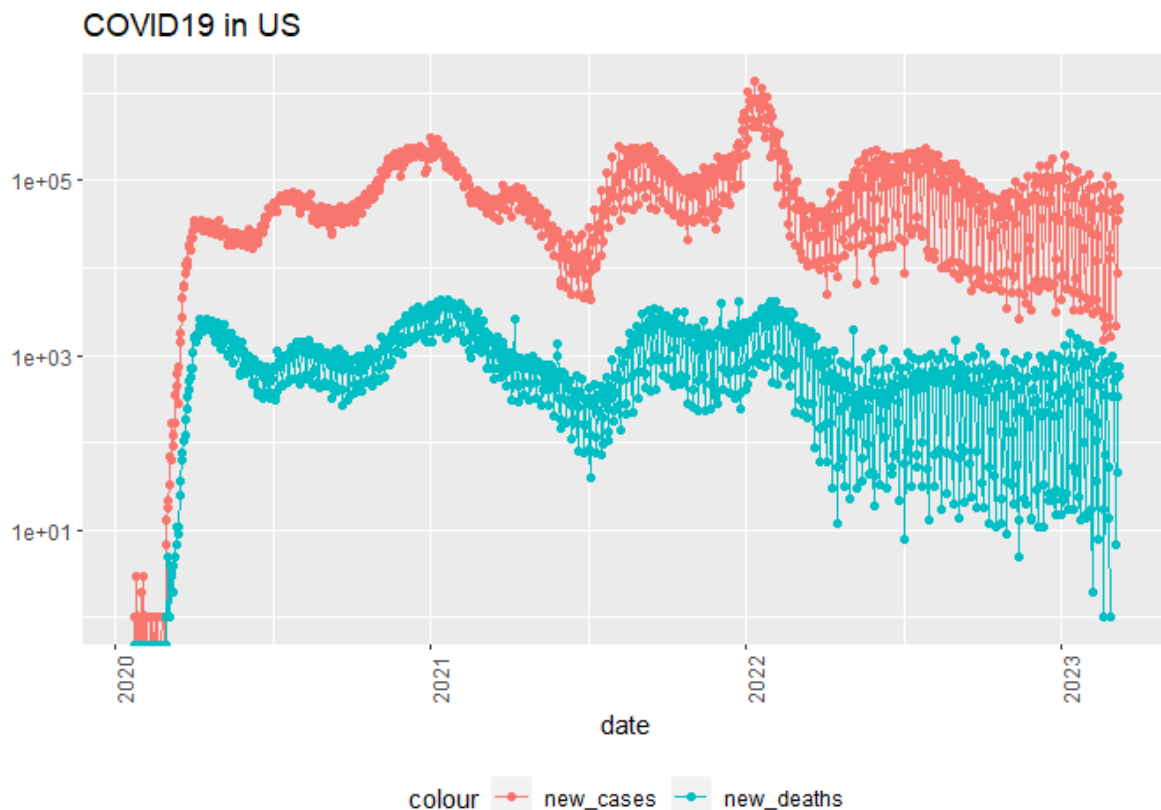
```
## [1] 1123836
```

```
##let's add in new_cases and new deaths as additional var
##This will be important when looking into vaccination dat
```

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

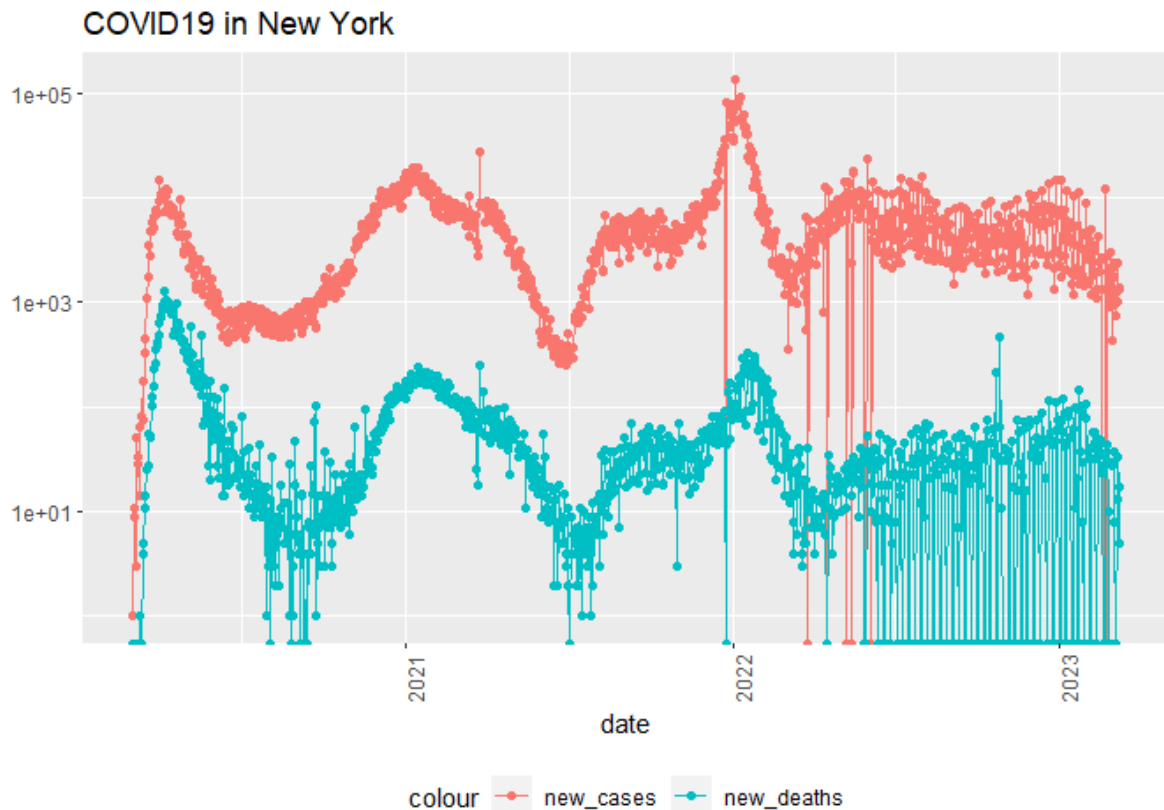
```
##tail(US_totals %>% select(new_cases, new_deaths, everyt

##Now graph new cases and new deaths
US_totals %>%
  ggplot(aes(x=date, y=new_cases))+
  geom_line(aes(color="new_cases"))+
  geom_point(aes(color="new_cases"))+
  geom_line(aes(y=new_deaths, color="new_deaths"))+
  geom_point(aes(y=new_deaths, color="new_deaths"))+
  scale_y_log10()+
  theme(legend.position="bottom",
        axis.text.x=element_text(angle = 90))+
  labs(title = "COVID19 in US",y=NULL)
```



```
##Let's single out NEw York and see how that Looks
US_by_state %>%
  filter(Province_State== state) %>%
  filter(cases>0) %>%
  ggplot(aes(x=date, y=new_cases))+
  geom_line(aes(color="new_cases"))+
```

```
geom_point(aes(color="new_cases"))+
geom_line(aes(y=new_deaths, color="new_deaths"))+
geom_point(aes(y=new_deaths, color="new_deaths"))+
scale_y_log10()+
theme(legend.position="bottom",
      axis.text.x=element_text(angle = 90))+
labs(title = str_c("COVID19 in ", state),y=NULL)
```



##which are the worst and bast states?

```
US_state_totals<- US_by_state %>%
  group_by(Province_State) %>%
  summarize(deaths=max(deaths), cases=max(cases),
            population=max(Population),
            cases_per_thou=1000*cases/population,
            deaths_per_thou=1000*deaths/population) %>%
  filter(cases>0,population>0)
```

Below is the code for checking lowest and highest death. Now we need to run the code for this analysis


```

##states with lowest deaths
US_state_totals%>%
  slice_min(deaths_per_thou,n=10) %>%
  select(deaths_per_thou, cases_per_thou, everything())

##stats with highest death
US_state_totals%>%
  slice_max(deaths_per_thou,n=10) %>%
  select(deaths_per_thou, cases_per_thou, everything())

```

Modelling and Statistical Analysis

We will use a linear model to predict deaths from cases

```

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

```

```

##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou, data = US_state_
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.3352 -0.5978  0.1491  0.6535  1.2086
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.36167    0.72480  -0.499    0.62
## cases_per_thou  0.01133    0.00232   4.881 9.76e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8615 on 54 degrees of freedom
## Multiple R-squared:  0.3061, Adjusted R-squared:  0.2933
## F-statistic: 23.82 on 1 and 54 DF,  p-value: 9.763e-06

```

Plot the predicted deaths against actual death

```

##check info
##US_state_totals %>% slice_min(cases_per_thou)
##US_state_totals %>% slice_max(cases_per_thou)

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

```

```

## # A tibble: 56 × 7
##   Province_State deaths  cases population cases_per_thou deat
##   <chr>          <dbl>  <dbl>         <dbl>         <dbl>
## 1 Alabama        21032 1.64e6    4903185         335.
## 2 Alaska         1486 3.08e5     740995         415.
## 3 American Samoa    34 8.32e3     55641         150.
## 4 Arizona        33102 2.44e6    7278717         336.
## 5 Arkansas        13020 1.01e6    3017804         334.
## 6 California     101159 1.21e7   39512223         307.
## 7 Colorado        14181 1.76e6    5758736         306.
## 8 Connecticut     12220 9.77e5    3565287         274.
## 9 Delaware        3324 3.31e5     973764         340.
## 10 District of Co.. 1432 1.78e5     705749         252.
## # i 46 more rows

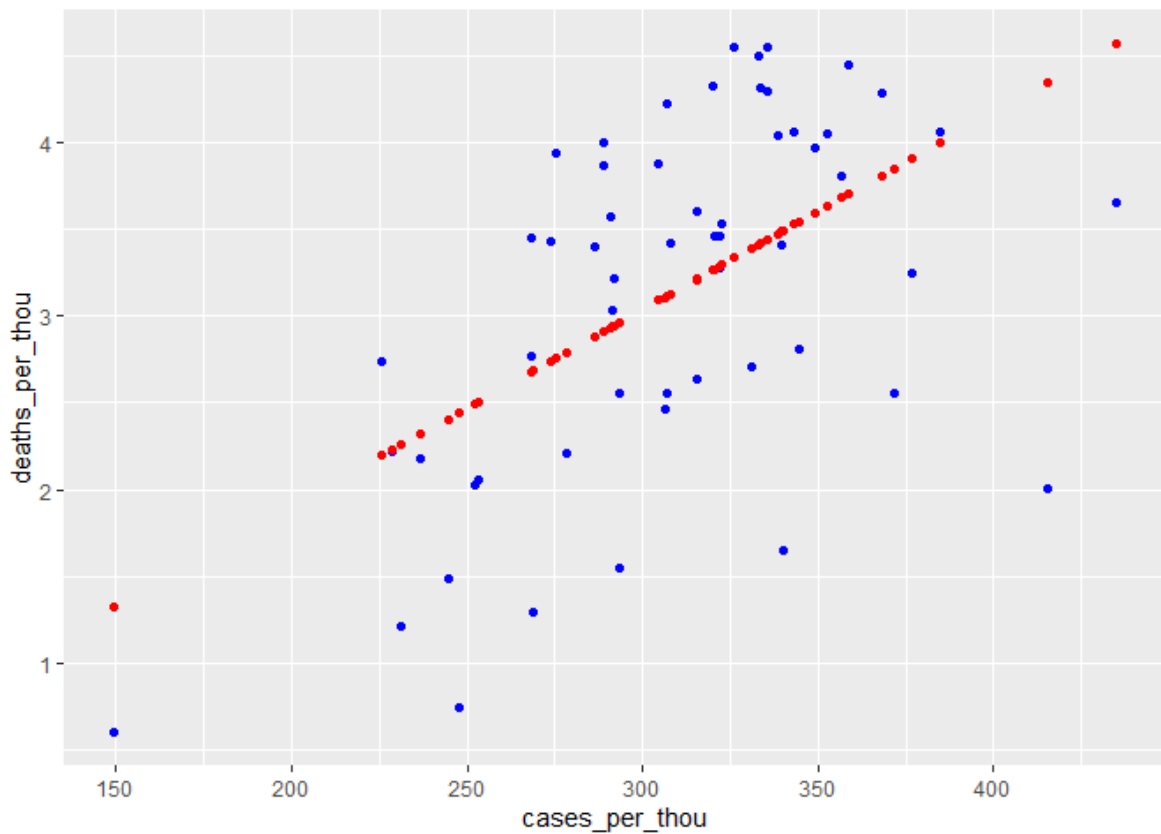
```

```

US_tot_w_pred<-US_state_totals%>% mutate(pred=predict(mod))

##finally plot cases per thousand vs deaths per thousand
US_tot_w_pred %>% ggplot()+
  geom_point(aes(x=cases_per_thou,
                 y=deaths_per_thou),
             color="blue")+
  geom_point(aes(x=cases_per_thou,
                 y=pred),
             color="red")

```



Vaccination Data Analysis

We will use similar techniques to read the vaccinations public data and merge it with the US_by_state data.

```
datvac<-read_csv("https://raw.githubusercontent.com/owid/covid19/master/covid19/covid19.csv")
```

```
## Rows: 54628 Columns: 16
## — Column specification —————
## Delimiter: ","
## chr   (1): location
## dbl  (14): total_vaccinations, total_distributed, people_vaccin
## date  (1): date
##
## i Use `spec()` to retrieve the full column specification for the data
## i Specify the column types or set `show_col_types = FALSE` to silence
```

```
View(datvac)
```

```
datvac$Province_State<-datvac$location

USvacc<-merge(US_by_state, datvac,
              by=c("Province_State", "date"),
              all = TRUE)
```

Missing values will be prominent because vaccinations weren't available until beginning of 2021. To address this, we will fill in na values with 0 to indicate no vaccinations

```
USvacc[is.na(USvacc)] <- 0
```

Convert variable to appropriate data type. Let's calculate some variables to visualize and then summarize all of the data by country (US).

```
USvacc2<-USvacc %>% mutate_if(is.character, as.factor)

us_dat_with_vacc<-USvacc2 %>%
  group_by(date) %>%
  summarise_if(is.numeric, funs(sum(., na.rm=TRUE)))
```

```
## Warning: `funs()` was deprecated in dplyr 0.8.0.
## i Please use a list of either functions or lambdas:
##
## # Simple named list: list(mean = mean, median = median)
##
## # Auto named with `tibble::lst()`: tibble::lst(mean, median)
##
## # Using lambdas list(~ mean(., trim = .2), ~ median(., na.rm =
## Call `lifecycle::last_lifecycle_warnings()` to see where this w
## generated.
```

```
us_dat_with_vacc<- us_dat_with_vacc %>%
  mutate(cases_per_thou=1000*cases/Population,
         deaths_per_thou=1000*deaths/Population,
         vacc_per_thou=1000*people_vaccinated/Population,
         new_cases_pt=1000*new_cases/Population,
```

```

    new_deaths_pt=1000*new_deaths/Population,
    vacc_per_thou=1000*people_vaccinated/Population
  ) %>%
  filter(new_cases>-1,new_deaths>-1,
         Population>0)

```

Let's visualize, new deaths, new cases, daily vaccinations and total vaccinations to see what the trends are.

```

newdeathp<-ggplot(data=us_dat_with_vacc, aes(x=date))+
  geom_smooth(aes(y=new_deaths, color= new_deaths))+
  geom_line(aes(y=new_deaths, color= new_deaths))

newcasep<-ggplot(data=us_dat_with_vacc, aes(x=date))+
  geom_smooth(aes(y=new_cases, color= new_cases))+
  geom_line(aes(y=new_cases, color= new_cases))

daily_vaccp<-ggplot(data=us_dat_with_vacc, aes(x=date))+
  geom_smooth(aes(y=daily_vaccinations, color= daily_vaccinations))+
  geom_line(aes(y=daily_vaccinations, color= daily_vaccinations))

total_vaccp<-ggplot(data=us_dat_with_vacc, aes(x=date))+
  geom_smooth(aes(y=people_fully_vaccinated, color= people_fully_vaccinated))+
  geom_line(aes(y=people_fully_vaccinated, color= people_fully_vaccinated))

newdeathp;newcasep;daily_vaccp;total_vaccp

```

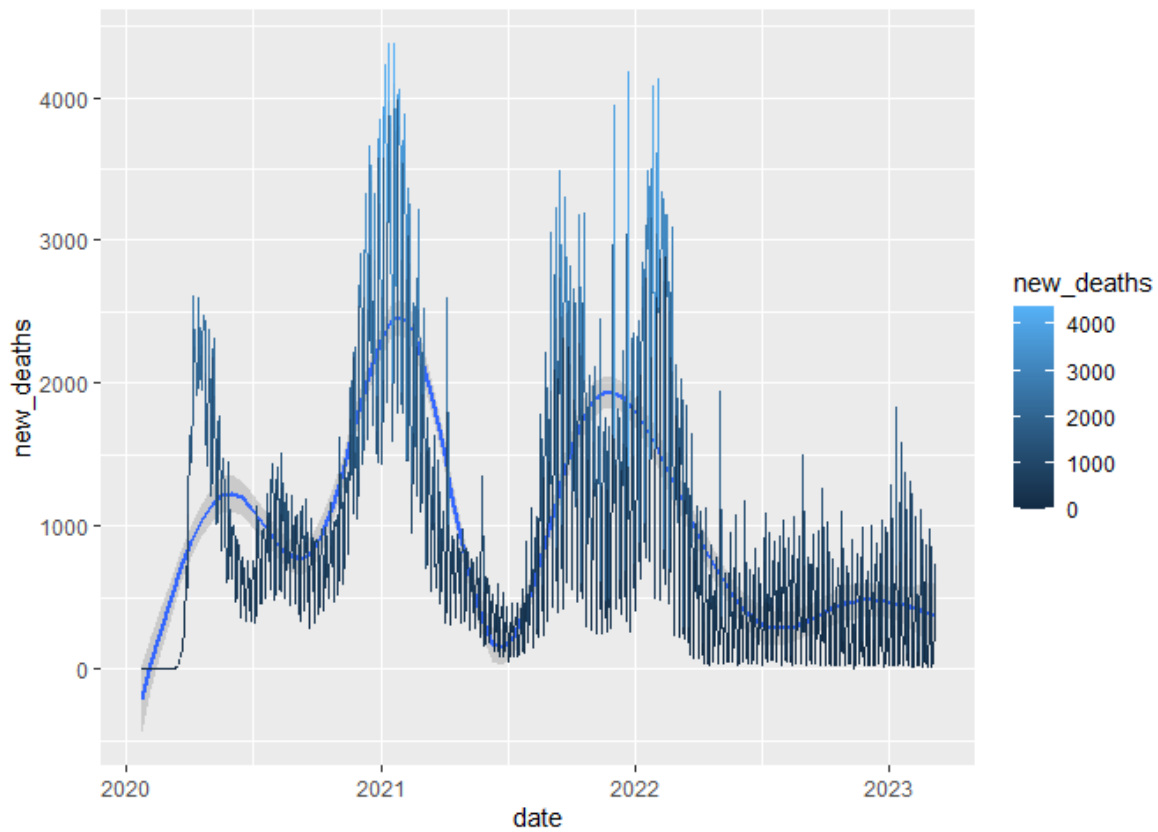
```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs

```

```

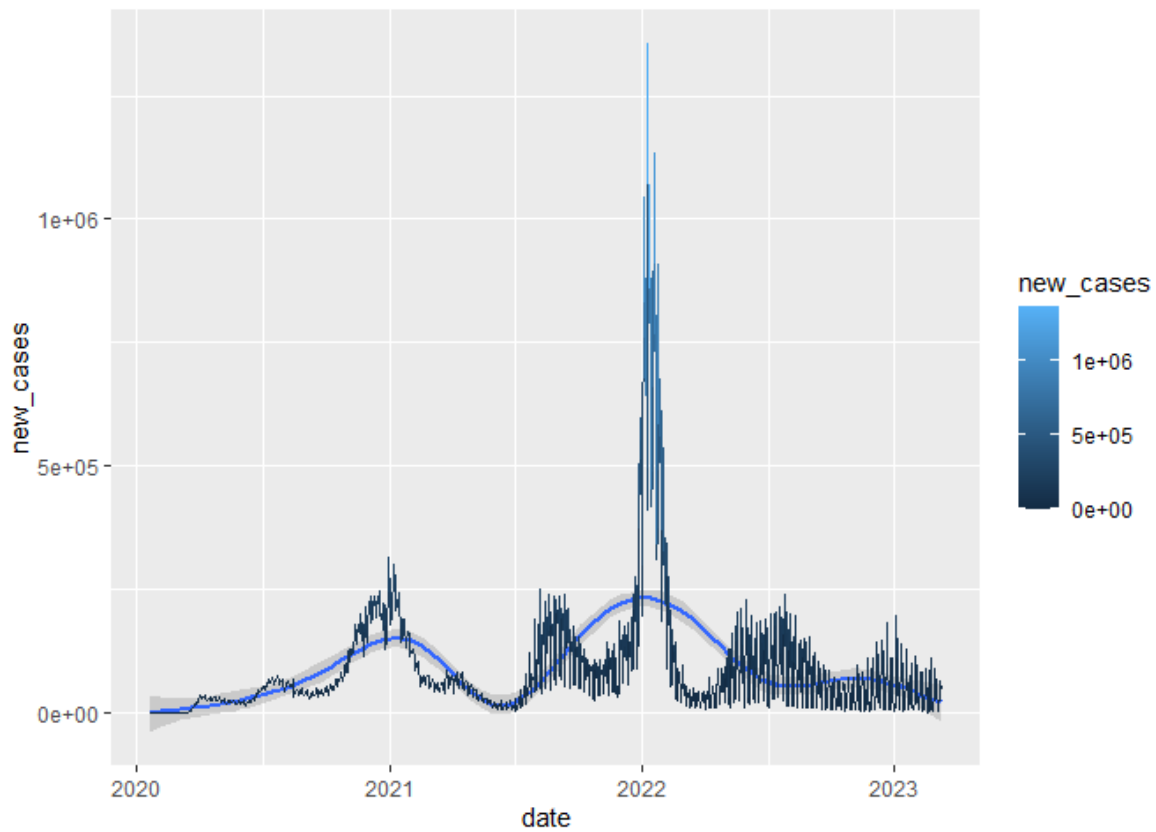
## Warning: The following aesthetics were dropped during statistical
## i This can happen when ggplot fails to infer the correct groups
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a
##   variable into a factor?

```



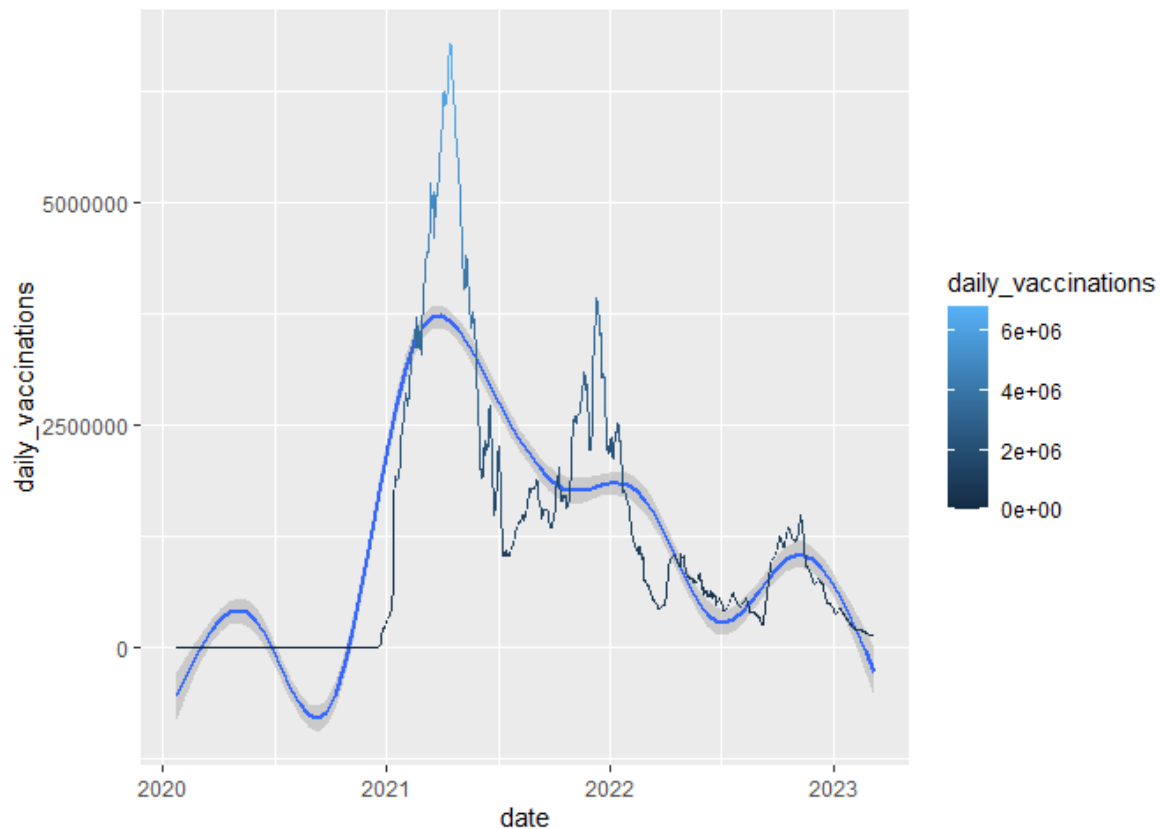
```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs
```

```
## Warning: The following aesthetics were dropped during statistic
## i This can happen when ggplot fails to infer the correct group
## the data.
## i Did you forget to specify a `group` aesthetic or to convert
## variable into a factor?
```



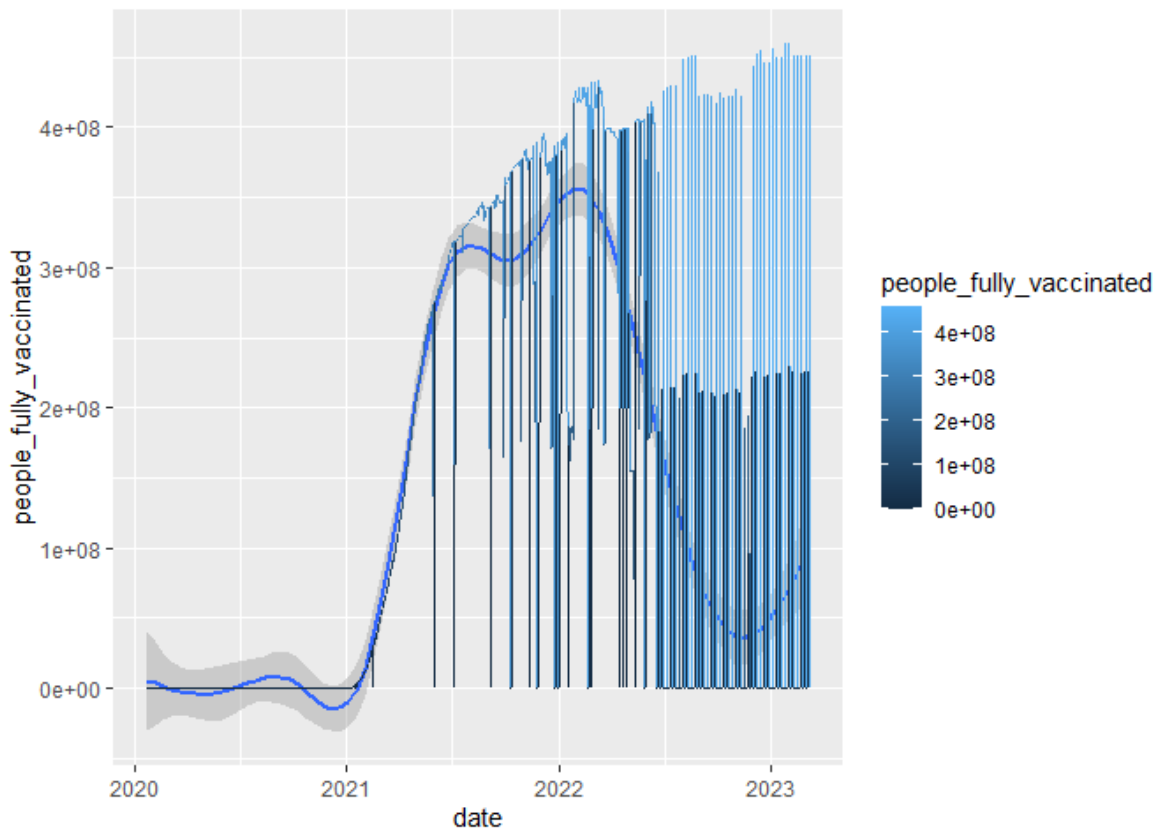
```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs
```

```
## Warning: The following aesthetics were dropped during statistical
## i This can happen when ggplot fails to infer the correct group
## the data.
## i Did you forget to specify a `group` aesthetic or to convert a
## variable into a factor?
```



```
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs
```

```
## Warning: The following aesthetics were dropped during statistical
## i This can happen when ggplot fails to infer the correct group:
##   the data.
## i Did you forget to specify a `group` aesthetic or to convert a
##   variable into a factor?
```

Looks like there might be some kind of relationship between vaccinations and new deaths

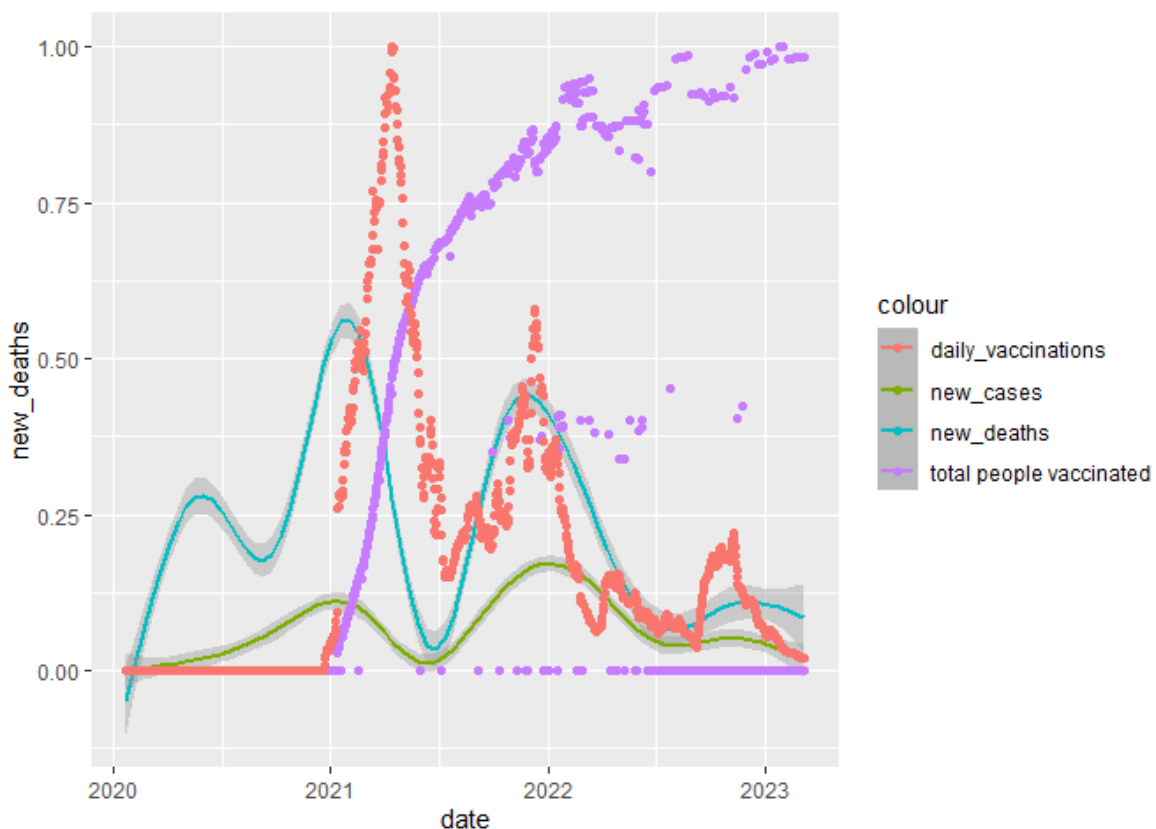
Now let's normalize the data so that the variables are in a 0-1 scale so we can see all data together in a graph

Create Normalization function:

```
norm_function<-function(dataa)
{
  normalize<-function(x){
    return((x-min(x))/(max(x)-min(x)))
  }
  datimpute<-select_if(dataa, is.numeric)
  norm<-as.data.frame(lapply(datimpute,normalize))
  norm2<-select_if(dataa,negate(is.numeric))
  all_dat<-cbind(norm2,norm)
  return(all_dat)
}

normdat<-norm_function(us_dat_with_vacc)
```

```
##visualize data
normdat%>% ggplot()+
  geom_smooth(aes(x=date,
                  y=new_deaths,
                  color="new_deaths"))+
  geom_smooth(aes(x=date,
                  y=new_cases,
                  color="new_cases"))+
  geom_point(aes(x=date,
                 y=people_vaccinated,
                 color="total people vaccinated"))+
  geom_point(aes(x=date,
                 y=daily_vaccinations,
                 color="daily_vaccinations"))
```



Looking at the overall summary of the US data, the low model fit suggests there may be other factors not included in the model that are more predictive of new deaths. Perhaps masks mandates, number of people working at home, and amount of social distancing could all have a stronger relationship with the data and predicting deaths. Those data

are not in the data set.

The spike in new_deaths may be due to lifting mask restrictions in many states and people deciding to get together for social gatherings because they are tired of the pandemic. A major factor also is that new variants of COVID-19 were discovered in 2021 such as Delta and Omicron variants.

```
modvacc<-lm(new_deaths~people_vaccinated, data=normdat)
summary(modvacc)
```

```
##
## Call:
## lm(formula = new_deaths ~ people_vaccinated, data = normdat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.26339 -0.16023 -0.06297  0.08285  0.79413
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.203771   0.008048  25.319  <2e-16 ***
## people_vaccinated 0.071584   0.016716   4.282   2e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2103 on 1137 degrees of freedom
## Multiple R-squared:  0.01587,    Adjusted R-squared:  0.01501
## F-statistic: 18.34 on 1 and 1137 DF,  p-value: 2.005e-05
```

```
modvaccpred<- normdat%>% mutate(pred=predict(modvacc))

modvaccpred %>% ggplot()+
  geom_smooth(aes(x=people_vaccinated,
                  y=new_deaths,
                  color="Actual new deaths"))+
  geom_smooth(aes(x=people_vaccinated,
                  y=pred,
```

```

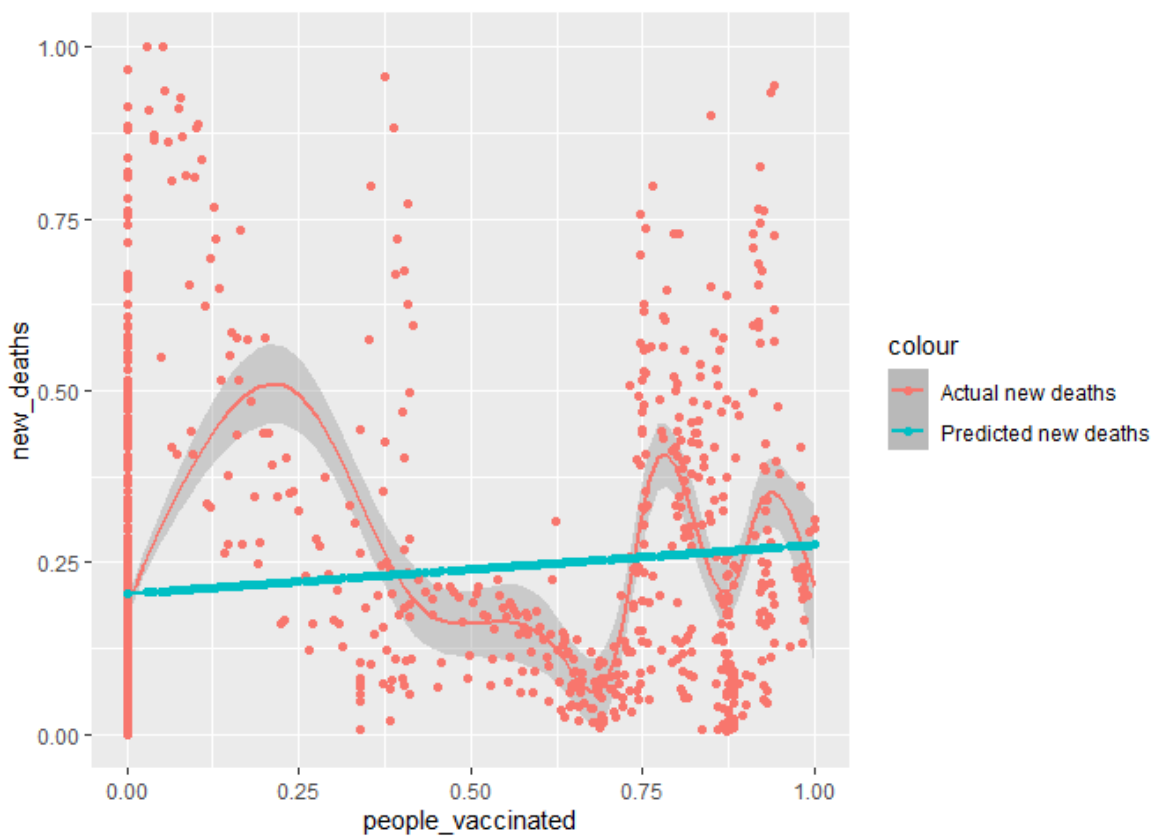
        color="Predicted new deaths"))+
geom_point(aes(x=people_vaccinated,
               y=new_deaths,
               color="Actual new deaths"))+
geom_point(aes(x=people_vaccinated,
               y=pred,
               color="Predicted new deaths"))

```

```

## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs
## `geom_smooth()` using method = 'gam' and formula = 'y ~ s(x, bs

```



We can see that deaths dropped around the time vaccinations with given.

Now let's see what happens if we focus on the part of the data where vaccinations were starting to be given.

Let's zoom in on the data and visualize the results.

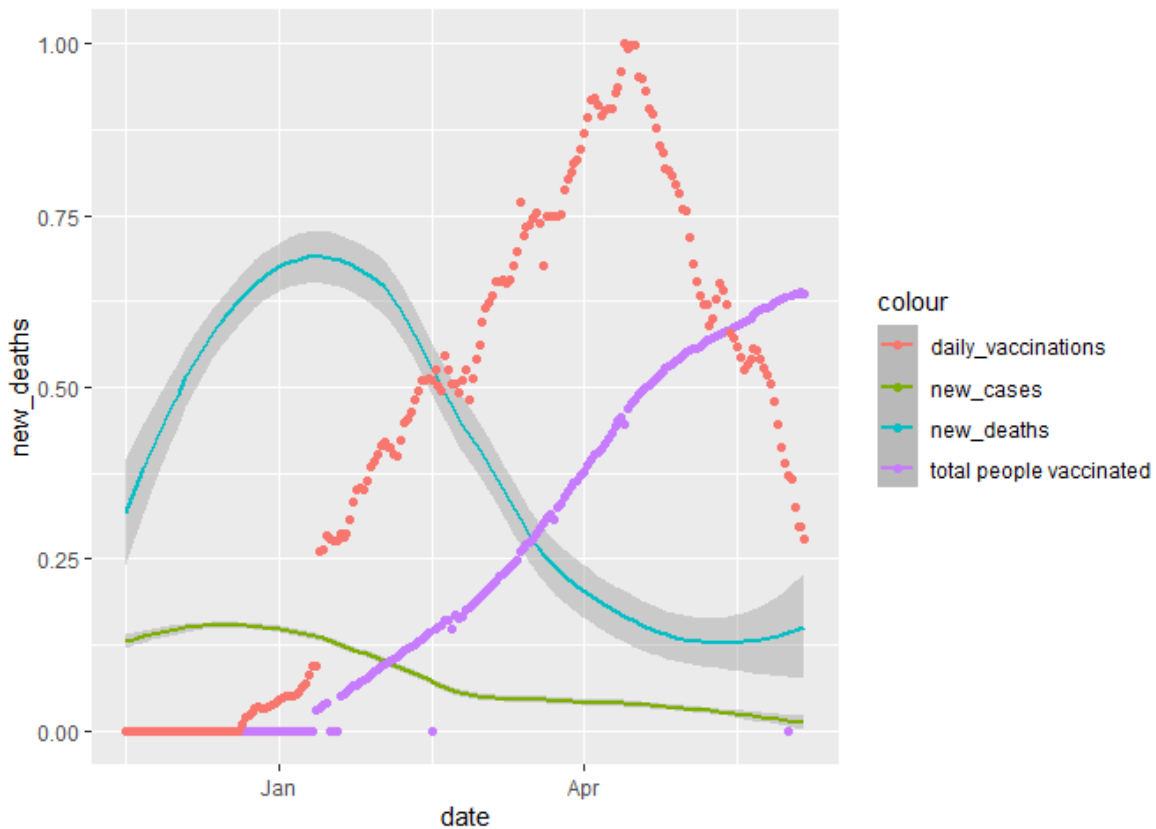
```

normdat2<- normdat[300:500,]

```

```
normdat2%>% ggplot()+
  geom_smooth(aes(x=date,
                  y=new_deaths,
                  color="new_deaths"))+
  geom_smooth(aes(x=date,
                  y=new_cases,
                  color="new_cases"))+
  geom_point(aes(x=date,
                 y=people_vaccinated,
                 color="total people vaccinated"))+
  geom_point(aes(x=date,
                 y=daily_vaccinations,
                 color="daily_vaccinations"))
```

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



There is a much higher adjusted r-squared this time indicating a better fit of the data during the time in which vaccinations were given.

```
modvacc2<-lm(new_deaths~people_vaccinated, data=normdat2)
summary(modvacc2)
```

```
##
## Call:
## lm(formula = new_deaths ~ people_vaccinated, data = normdat2)
##
## Residuals:
```

	Min	1Q	Median	3Q	Max
	-0.54772	-0.12355	-0.00162	0.07371	0.45378

```
##
## Coefficients:
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	0.58932	0.01824	32.31	<2e-16 ***
people_vaccinated	-0.81513	0.05474	-14.89	<2e-16 ***

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1785 on 199 degrees of freedom
## Multiple R-squared:  0.527, Adjusted R-squared:  0.5246
## F-statistic: 221.7 on 1 and 199 DF, p-value: < 2.2e-16
```

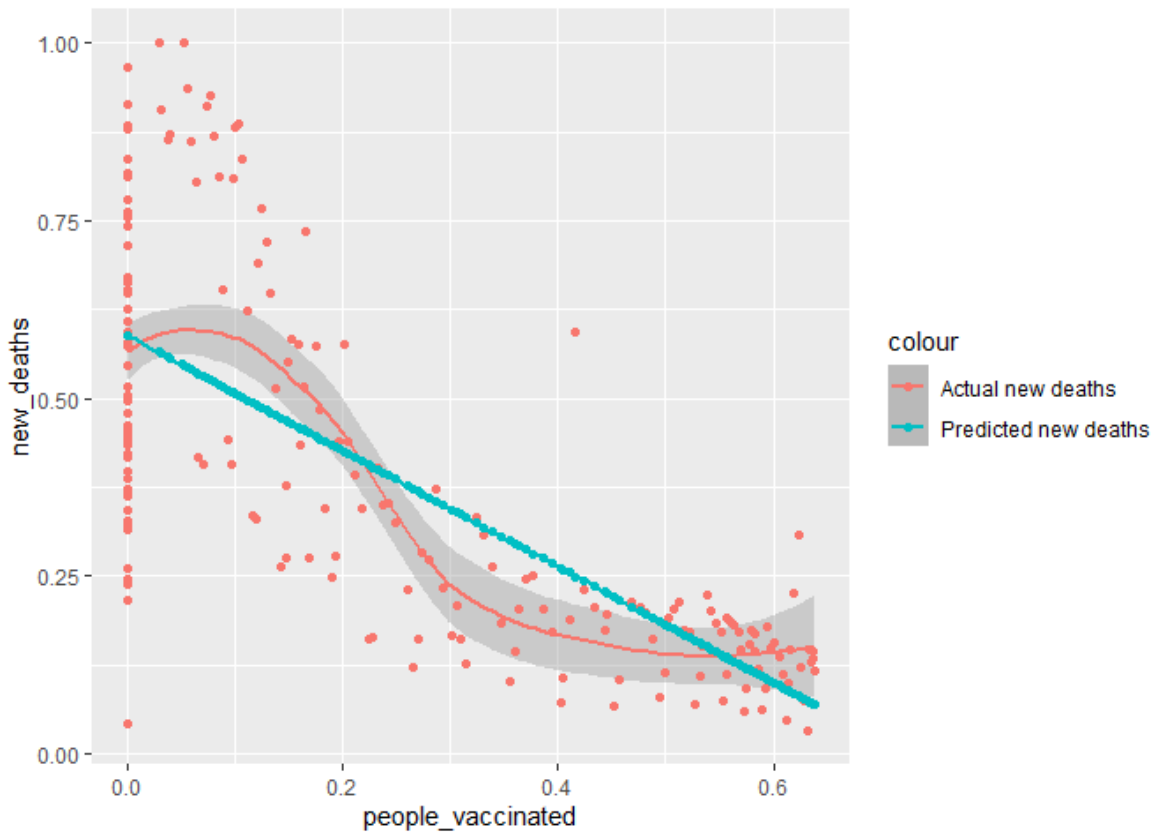
```
modvaccpred2<- normdat2%>% mutate(pred=predict(modvacc2))
```

We can see that new deaths dropped as more and more people received vaccinations.

```
modvaccpred2 %>% ggplot()+
  geom_smooth(aes(x=people_vaccinated,
                  y=new_deaths,
                  color="Actual new deaths"))+
  geom_smooth(aes(x=people_vaccinated,
                  y=pred,
                  color="Predicted new deaths"))+
  geom_point(aes(x=people_vaccinated,
                  y=new_deaths,
                  color="Actual new deaths"))+
  geom_point(aes(x=people_vaccinated,
```

```
y=pred,  
color="Predicted new deaths"))
```

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



Conclusions and Bias Statements

In our analysis, we see that cases is a good predictor of deaths. This makes sense because if one gets COVID, there a high possibility (at least in compared to other human viruses like H1N1) that you can die from it.

In our analysis of the vaccination data we see there isn't much of correlation overall with total vaccinated people and prediction of new deaths. However when we focus on only the important time points when vaccinations were administered, we can see decreasing new deaths, indicating that vaccines were effective at preventing new deaths from COVID.

Possible biases include:

1. Recording and reporting of cases may differ for each state. We can see some number go lower then back to higher values as evidence for this.
2. Lag in new cases, new deaths due to vaccine.
3. Other factors were not included in the analysis because they were not in the data sets. These include, but are not limited to: mask mandate lengths and easing of restrictions, stay at home orders and social distance. These may also be difficult to quantify.

```
sessionInfo()
```

```
## R version 4.3.1 (2023-06-16 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 11 x64 (build 22631)
##
## Matrix products: default
##
##
## locale:
## [1] LC_COLLATE=English_United States.utf8
## [2] LC_CTYPE=English_United States.utf8
## [3] LC_MONETARY=English_United States.utf8
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.utf8
##
## time zone: America/Los_Angeles
## tzcode source: internal
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods
##
## other attached packages:
## [1] lubridate_1.9.2 forcats_1.0.0  stringr_1.5.0  dplyr_1.1.
## [5] purrr_1.0.2    readr_2.1.4    tidyr_1.3.0    tibble_3.2
## [9] ggplot2_3.4.3  tidyverse_2.0.0 prettydoc_0.4.1
```



```
##
## loaded via a namespace (and not attached):
## [1] sass_0.4.7          utf8_1.2.3          generics_0.1.3      latt
## [5] stringi_1.7.12      hms_1.1.3           digest_0.6.33       magr
## [9] evaluate_0.22       grid_4.3.1          timechange_0.2.0    fast
## [13] Matrix_1.6-1        jsonlite_1.8.7      mgcv_1.8-42         fans
## [17] scales_1.2.1        jquerylib_0.1.4     cli_3.6.1           rlan
## [21] crayon_1.5.2        splines_4.3.1       bit64_4.0.5         muns
## [25] withr_2.5.0         cachem_1.0.8        yaml_2.3.7          tool
## [29] parallel_4.3.1      tzdb_0.4.0          colorspace_2.1-0    curl
## [33] vctrs_0.6.3         R6_2.5.1            lifecycle_1.0.3     bit_
## [37] vroom_1.6.3         pkgconfig_2.0.3     pillar_1.9.0        bsli
## [41] gtable_0.3.4        glue_1.6.2          highr_0.10          xfun
## [45] tidyselect_1.2.0    rstudioapi_0.15.0   knitr_1.43          farv
## [49] nlme_3.1-162        htmltools_0.5.6     labeling_0.4.3      rmar
## [53] compiler_4.3.1
```