

COVID-19 Vaccine Data Analysis

KCS

2/7/2022

COVID-19 Vaccine Data Analysis

In this report, we will be reproducing a part of COVID-19 data analysis done in Week #3 of the “Data Science as a field course” as well as do our own analysis with the public COVID-19 vaccination data.

The primary objective is to see what the relationship is with number of vaccines given and new deaths due to COVID.

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 packages ----- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5      v purrr   0.3.4
## v tibble  3.1.6      v dplyr   1.0.7
## v tidyr   1.1.4      v stringr 1.4.0
## v readr   2.1.1      v forcats 0.5.1
```

```
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()     masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##   date, intersect, setdiff, union
```

```
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 gggithub

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

```
## Rows: 282 Columns: 755
```

```
## -- Column specification -----
## Delimiter: ","
## chr   (2): Province/State, Country/Region
## dbl (753): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20, ...
```

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

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

```
## Rows: 282 Columns: 755
```

```
## -- Column specification -----
## Delimiter: ","
## chr   (2): Province/State, Country/Region
## dbl (753): Lat, Long, 1/22/20, 1/23/20, 1/24/20, 1/25/20, 1/26/20, 1/27/20, ...
```

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

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

```
## Rows: 3342 Columns: 762
```

```
## -- Column specification -----  
## Delimiter: ","  
## chr (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key  
## dbl (756): UID, code3, FIPS, Lat, Long_, 1/22/20, 1/23/20, 1/24/20, 1/25/20,...
```

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

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

```
## Rows: 3342 Columns: 763
```

```
## -- Column specification -----  
## Delimiter: ","  
## chr (6): iso2, iso3, Admin2, Province_State, Country_Region, Combined_Key  
## dbl (757): UID, code3, FIPS, Lat, Long_, Population, 1/22/20, 1/23/20, 1/24/...
```

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

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()

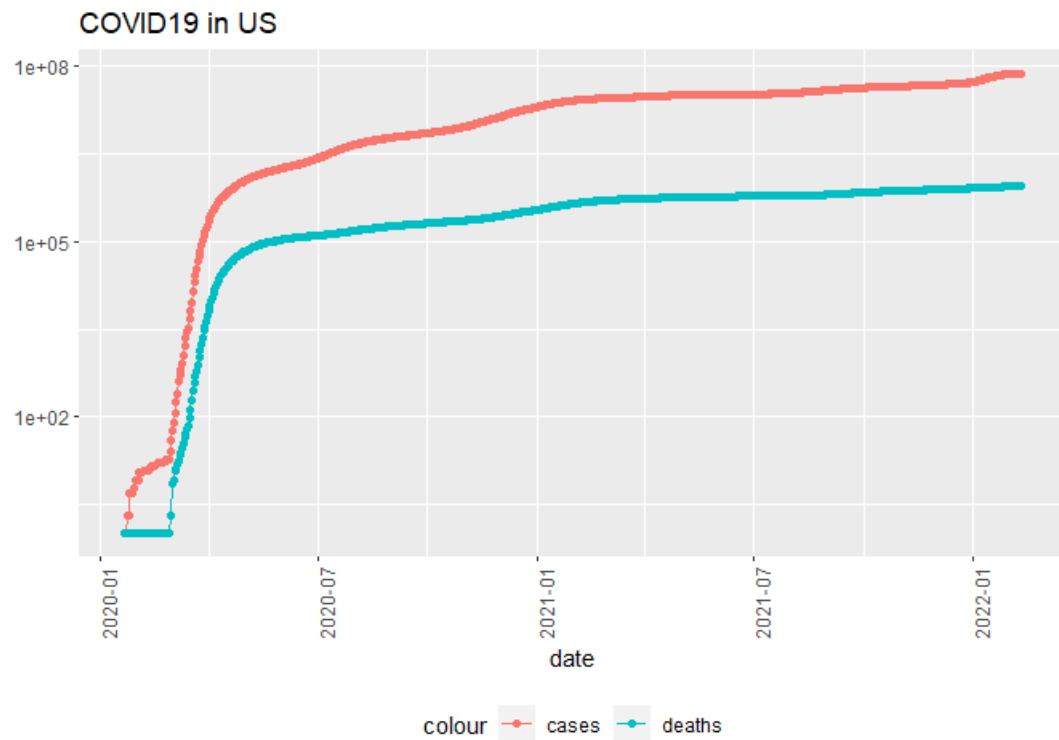
```

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 cases and deaths

```
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"))+
```

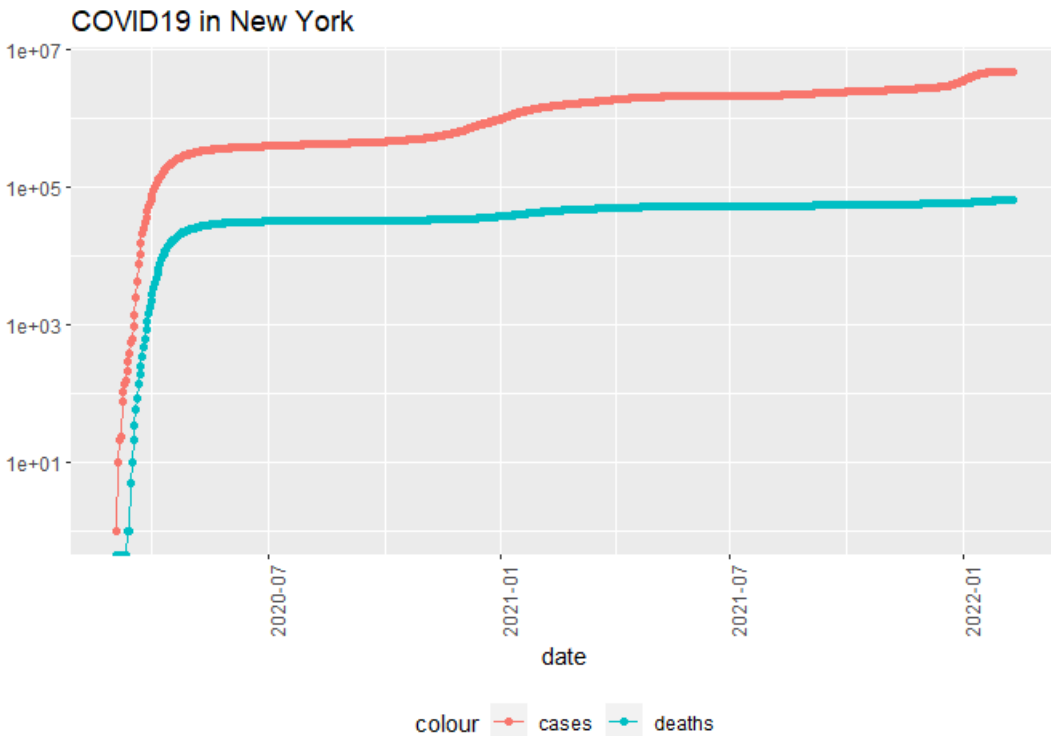
```
  geom_point(aes(y=deaths, color="deaths"))+
```

```
  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] 915434
```

```
##let's add in new_cases and new deaths as additional variables.
##This will be important when looking into vaccination date after
```

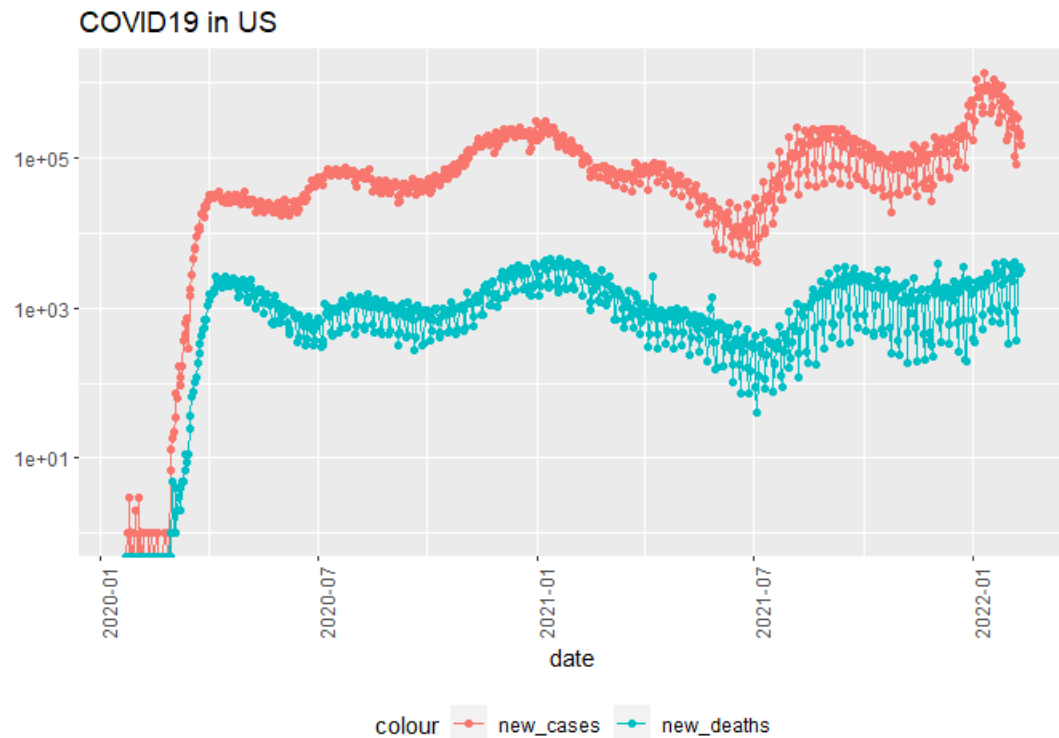
```
US_by_state<-US_by_state %>%
  mutate(new_cases= cases-lag(cases),
         new_deaths=deaths-lag(deaths))
US_totals<-US_totals %>%
  mutate(new_cases= cases-lag(cases),
         new_deaths=deaths-lag(deaths))
```

```
##tail(US_totals %>% select(new_cases, new_deaths, everything()))
```

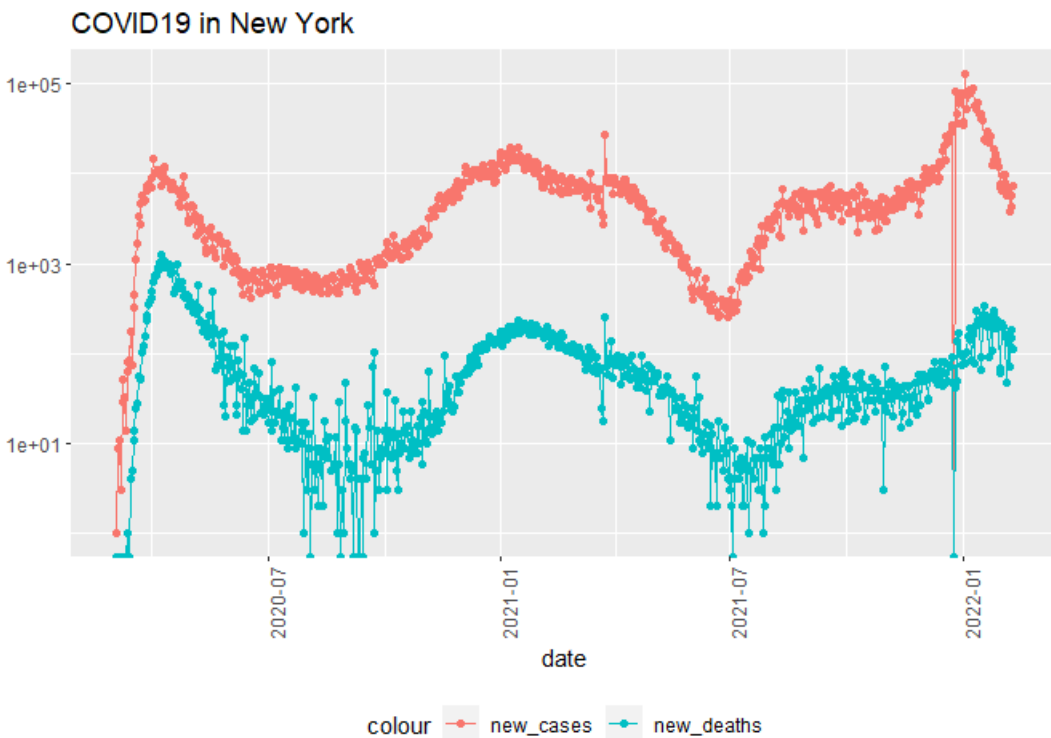
```
##Now graph new cases and new deaths
```

```
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)
```

Code for checking lowest and highest death. Now 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_totals)
summary(mod)
```

```
##
## Call:
## lm(formula = deaths_per_thou ~ cases_per_thou, data = US_state_totals)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.97780 -0.51920  0.08216  0.54464  0.97446
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.144021   0.381313  -0.378   0.707
## cases_per_thou  0.011667   0.001635   7.135 2.49e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6623 on 54 degrees of freedom
## Multiple R-squared:  0.4852, Adjusted R-squared:  0.4757
## F-statistic: 50.91 on 1 and 54 DF,  p-value: 2.489e-09
```

Plot the predicted deaths against actual death

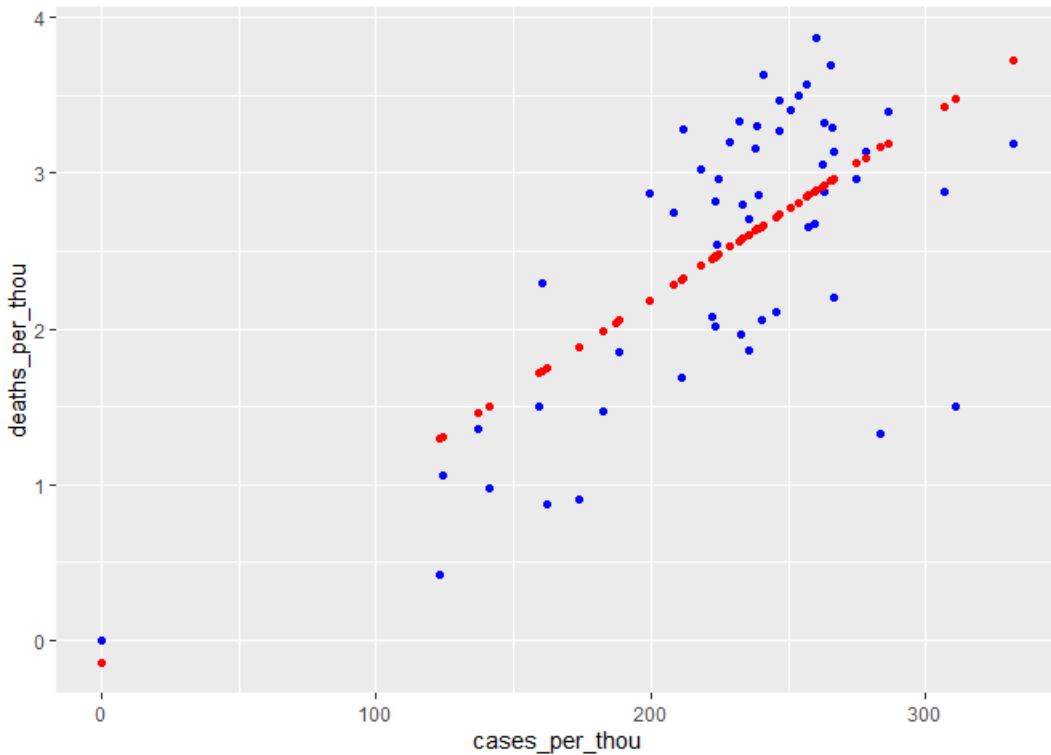
```
##check info
##US_state_totals %>% slice_min(cases_per_thou)
##US_state_totals %>% slice_max(cases_per_thou)

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

```
## # A tibble: 56 x 7
##   Province_State deaths  cases population cases_per_thou deaths_per_thou  pred
##   <chr>          <dbl>  <dbl>    <dbl>          <dbl>          <dbl> <dbl>
## 1 Alabama      17502 1.26e6  4903185         256.           3.57  2.85
## 2 Alaska       1114 2.30e5   740995         311.           1.50  3.48
## 3 American Samoa    0 1.8 e1    55641          0.324           0   -0.140
## 4 Arizona      26856 1.93e6  7278717         265.           3.69  2.95
## 5 Arkansas      9937 8.02e5  3017804         266.           3.29  2.96
## 6 California    82122 8.78e6  39512223         222.           2.08  2.45
## 7 Colorado     11583 1.29e6  5758736         223.           2.01  2.46
## 8 Connecticut   10219 7.12e5  3565287         200.           2.87  2.19
## 9 Delaware      2601 2.53e5   973764         259.           2.67  2.88
## 10 District of C~ 1305 1.33e5   705749         188.           1.85  2.05
## # ... with 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/covid-19-data/master/public/d
```

```
## Rows: 25636 Columns: 16
```

```
## -- Column specification -----
## Delimiter: ","
## chr   (1): location
## dbl  (14): total_vaccinations, total_distributed, people_vaccinated, people...
## date  (1): date
```

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

```
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.
## 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 = TRUE))
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was 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))

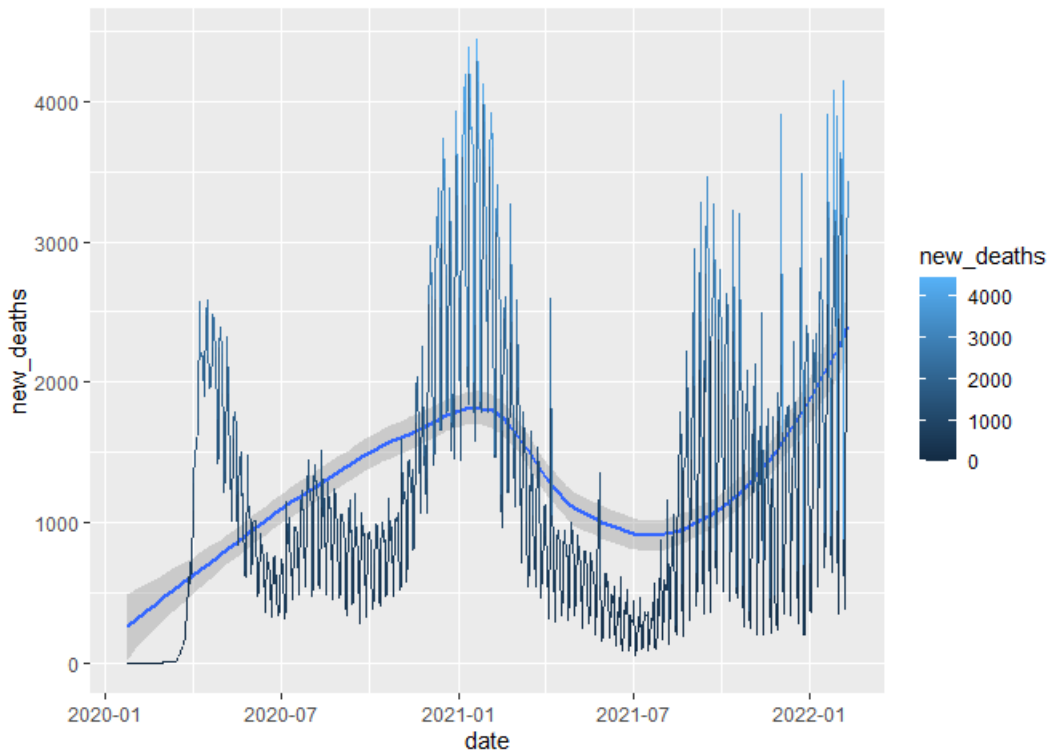
daily_vaccp<-ggplot(data=us_dat_with_vacc, aes(x=date))+
  geom_smooth(aes(y=daily_vaccinations, color= daily_vaccinations))+
  geom_line(aes(y=daily_vaccinations, color= daily_vaccinations))

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))

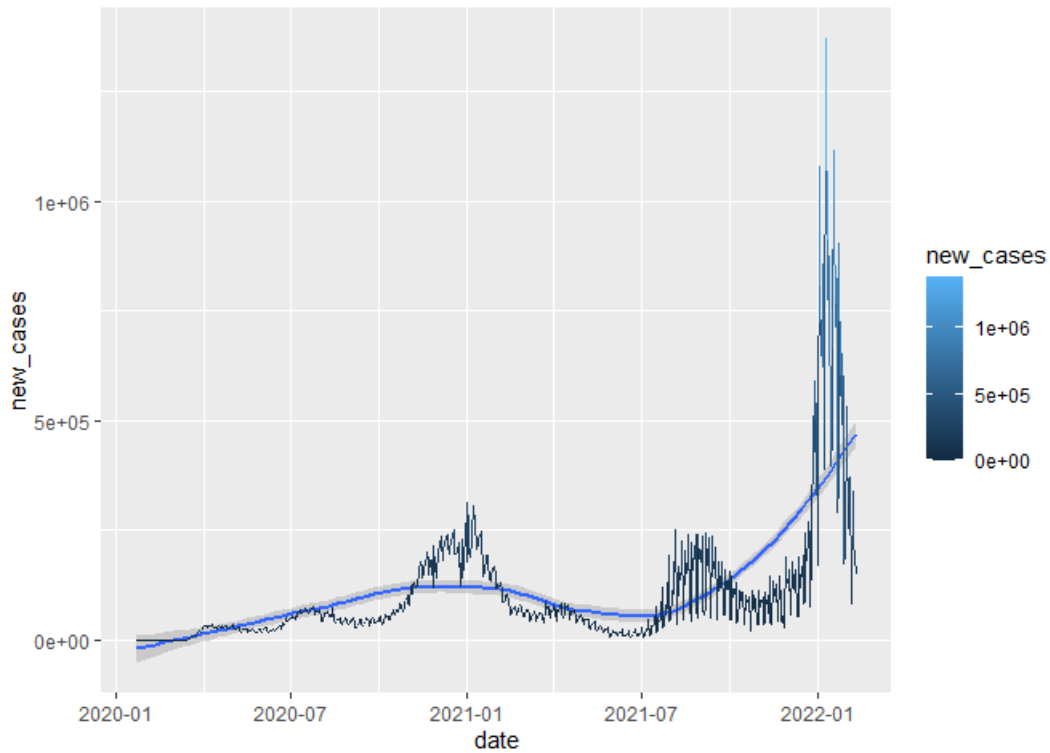
newdeathp;newcasep;daily_vaccp;total_vaccp

```

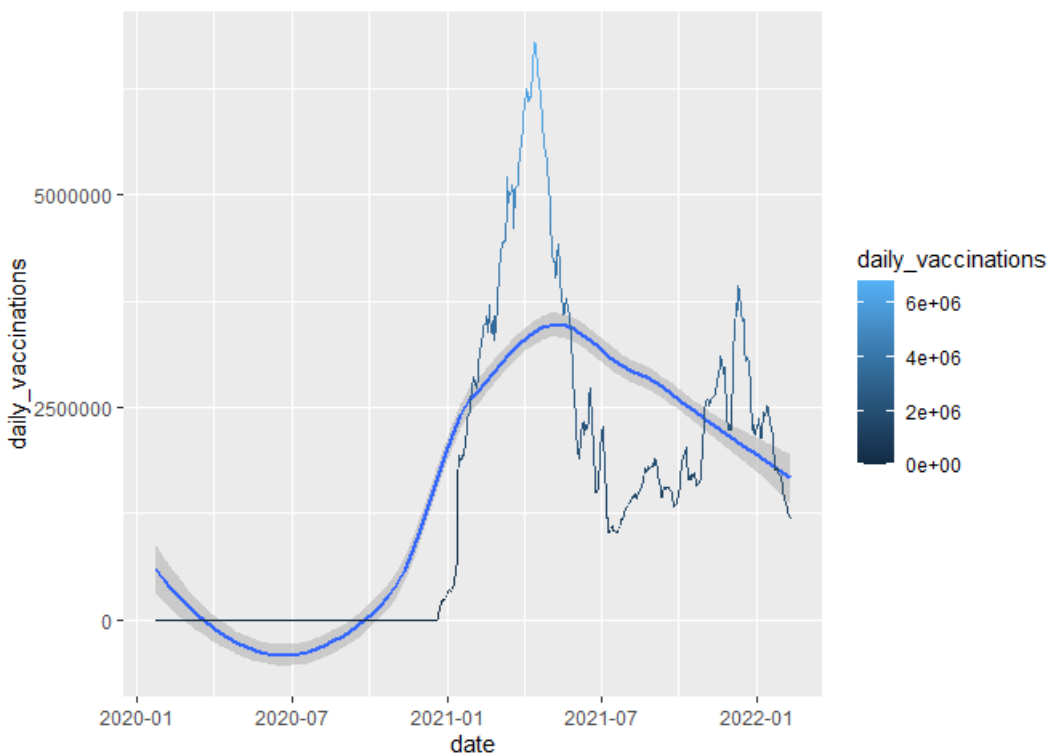
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



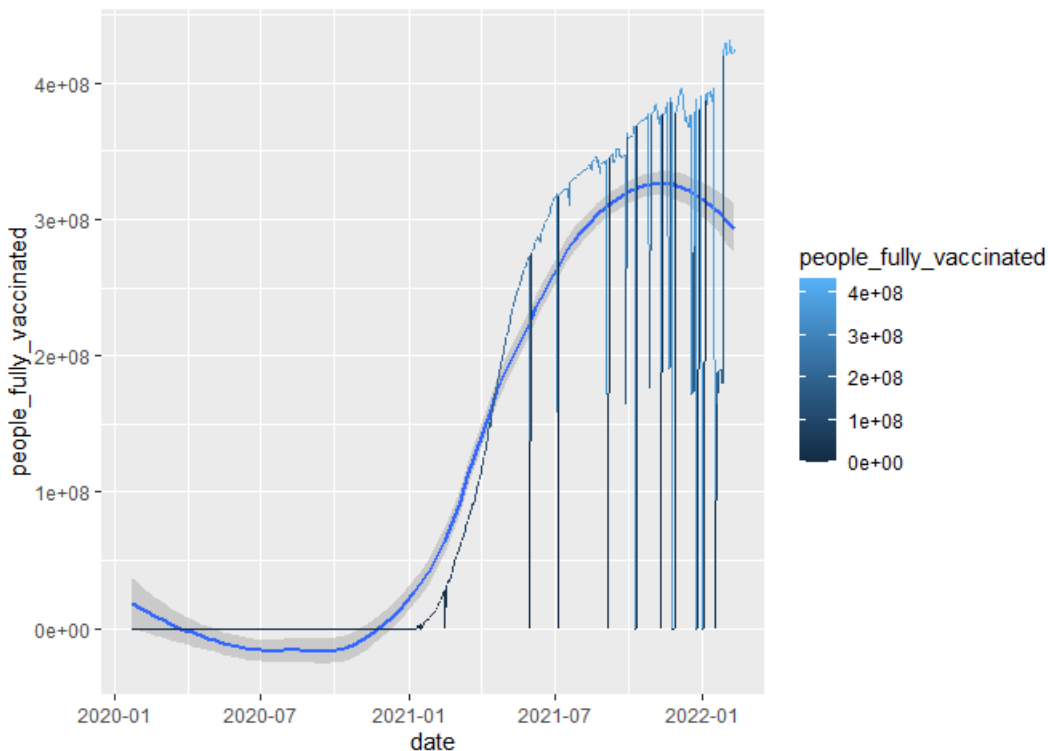
```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



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



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



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

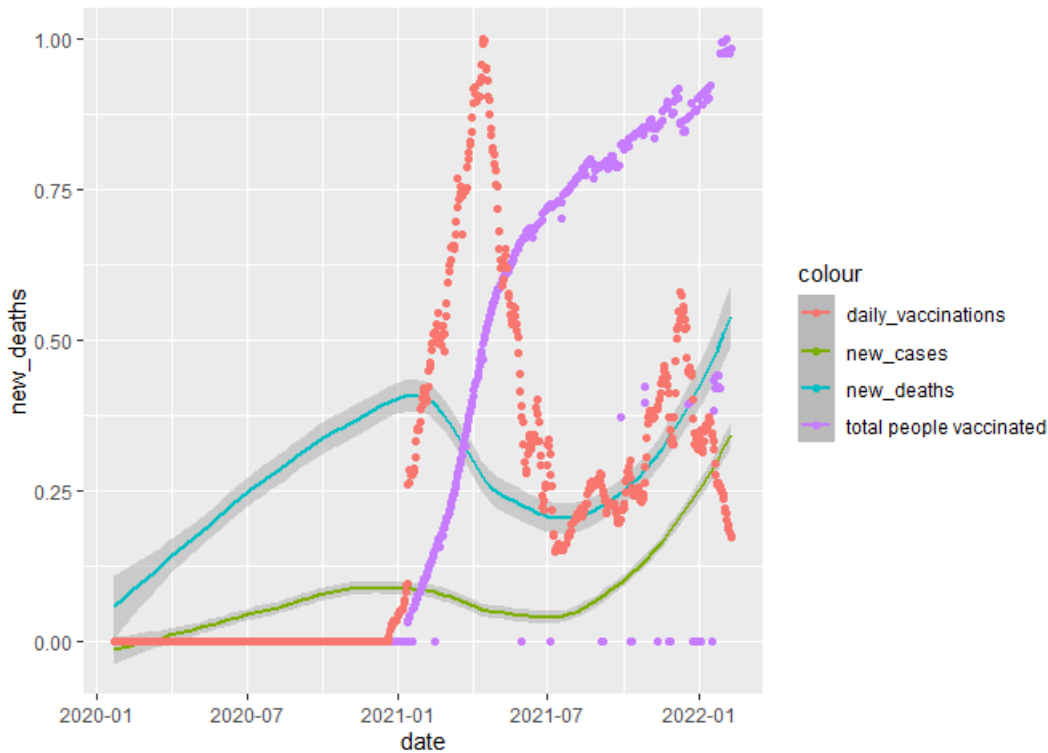
Create function Normalization function

```
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 dataset.

The spike in new_deaths may be due to lifting mask restrictions for many states and people deciding to get together 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 especially Omicron

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

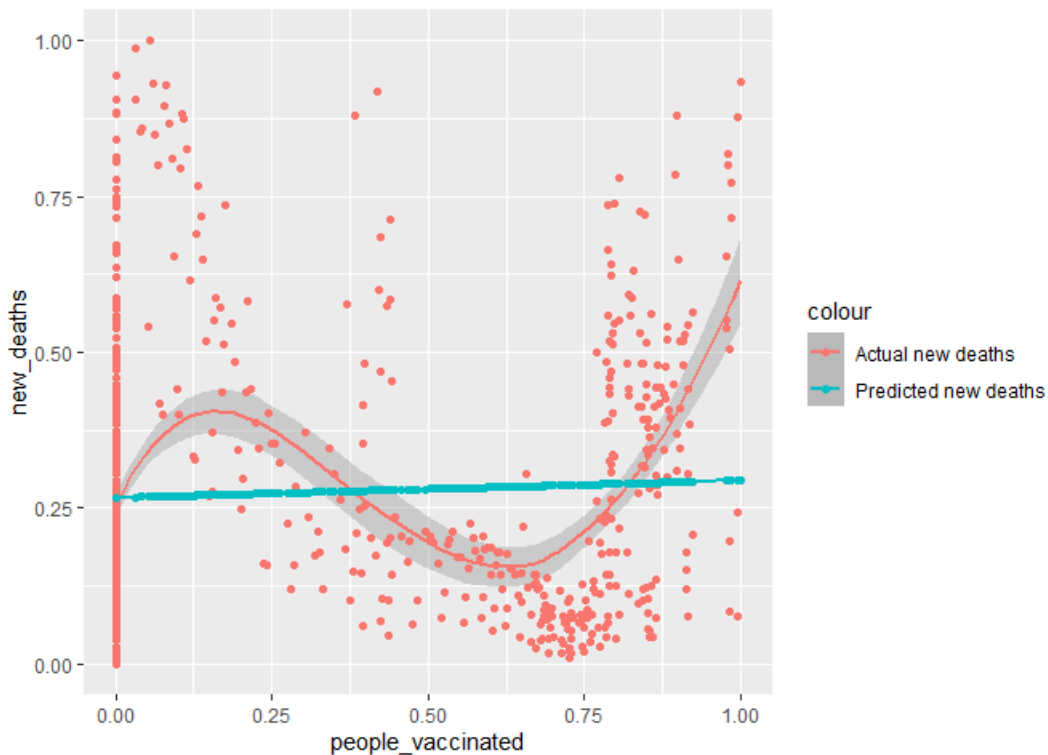
```
##
## Call:
## lm(formula = new_deaths ~ people_vaccinated, data = normdat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.27712 -0.17294 -0.05246  0.12726  0.73223
##
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.26621    0.01074   24.79  <2e-16 ***
## people_vaccinated 0.02774    0.02255    1.23   0.219
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2239 on 748 degrees of freedom
## Multiple R-squared:  0.002019,    Adjusted R-squared:  0.0006851
## F-statistic: 1.513 on 1 and 748 DF,  p-value: 0.219
```

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

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

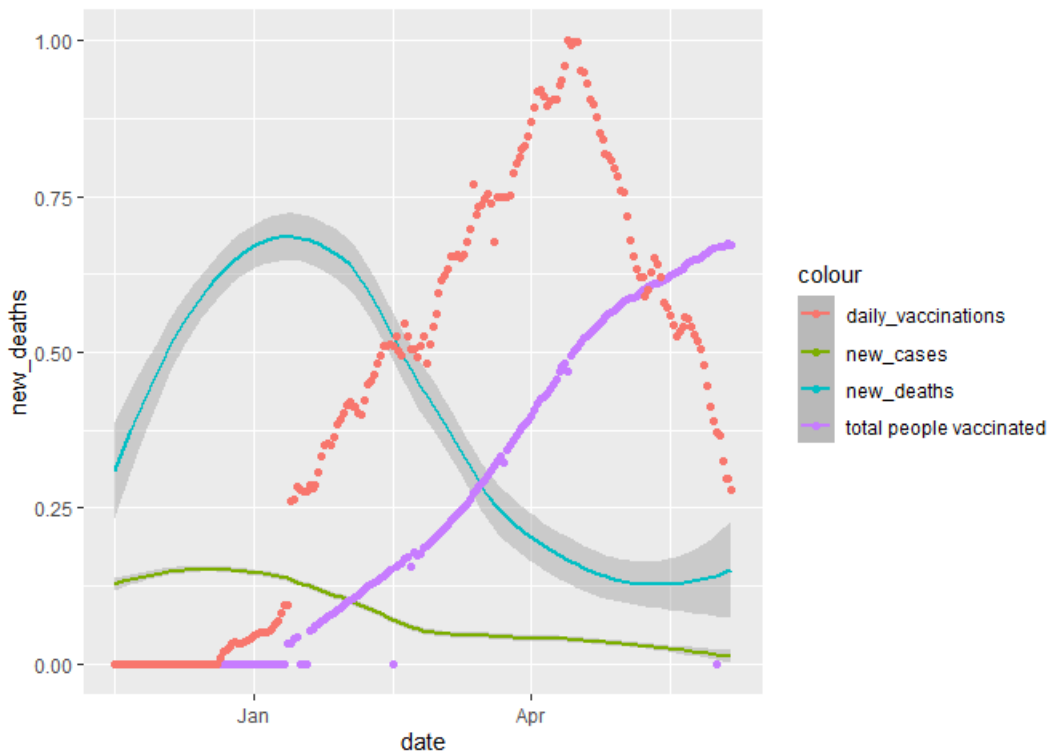
Now let's see what happens if we focus on the part of the data where vaccinations were starting to be given.

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's 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)
```

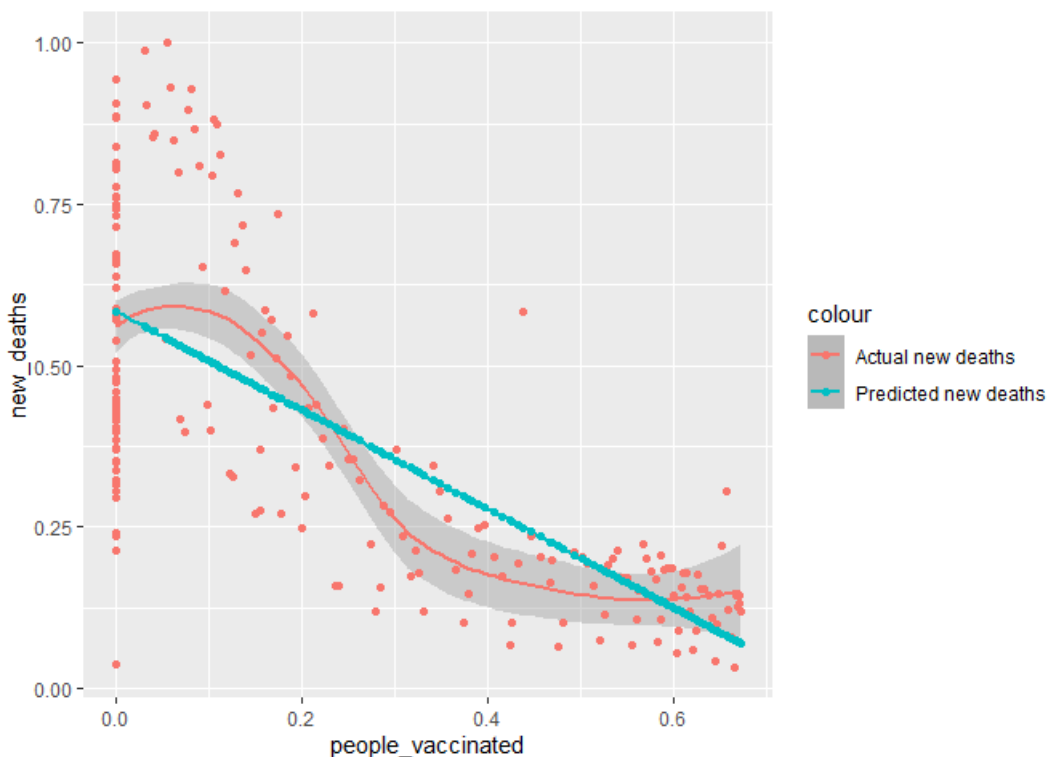
```
##
## Call:
## lm(formula = new_deaths ~ people_vaccinated, data = normdat2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.54813 -0.12460 -0.00310  0.07321  0.45828
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      0.58460     0.01828   31.97  <2e-16 ***
## people_vaccinated -0.76470     0.05202  -14.70  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1789 on 199 degrees of freedom
## Multiple R-squared:  0.5206, Adjusted R-squared:  0.5181
## F-statistic: 216.1 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 got 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, it is a possibility that you can die from it

In our analysis of the vaccination data we see there isn't much correlation overall with total vaccinated people and prediction of new deaths. However when we focus on only the important timepoints when vaccinations were administered, we 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 datasets. 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 4.1.0 (2021-05-18)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 19043)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United States.1252
## [2] LC_CTYPE=English_United States.1252
## [3] LC_MONETARY=English_United States.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United States.1252
##
## attached base packages:
## [1] stats      graphics  grDevices  utils      datasets  methods    base
##
## other attached packages:
## [1] lubridate_1.8.0 forcats_0.5.1  stringr_1.4.0  dplyr_1.0.7
## [5] purrr_0.3.4    readr_2.1.1    tidyr_1.1.4    tibble_3.1.6
## [9] ggplot2_3.3.5  tidyverse_1.3.1 prettydoc_0.4.1
##
## loaded via a namespace (and not attached):
## [1] Rcpp_1.0.7      lattice_0.20-44  assertthat_0.2.1 digest_0.6.29
## [5] utf8_1.2.2      R6_2.5.1         cellranger_1.1.0 backports_1.4.1
## [9] reprex_2.0.1    evaluate_0.14    highr_0.9       httr_1.4.2
## [13] pillar_1.6.5    rlang_0.4.12     curl_4.3.2      readxl_1.3.1
## [17] rstudioapi_0.13 jquerylib_0.1.4   Matrix_1.3-3    rmarkdown_2.11
## [21] splines_4.1.0   labeling_0.4.2   bit_4.0.4       munsell_0.5.0
## [25] broom_0.7.12    compiler_4.1.0   modelr_0.1.8    xfun_0.29
## [29] pkgconfig_2.0.3 mgcv_1.8-35      htmltools_0.5.2 tidyselect_1.1.1
## [33] fansi_1.0.2     crayon_1.4.2     tzdb_0.2.0      dbplyr_2.1.1
## [37] withr_2.4.3     grid_4.1.0       nlme_3.1-152    jsonlite_1.7.3
## [41] gtable_0.3.0    lifecycle_1.0.1  DBI_1.1.2       magrittr_2.0.1
## [45] scales_1.1.1    cli_3.1.1        stringi_1.7.6   vroom_1.5.7
## [49] farver_2.1.0    fs_1.5.2         xml2_1.3.3      bslib_0.3.1
```

```
## [53] ellipsis_0.3.2    generics_0.1.1    vctrs_0.3.8      tools_4.1.0
## [57] bit64_4.0.5       glue_1.6.0        hms_1.1.1        parallel_4.1.0
## [61] fastmap_1.1.0     yaml_2.2.2        colorspace_2.0-2 rvest_1.0.2
## [65] knitr_1.37        haven_2.4.3       sass_0.4.0
```