COVID-19 Data

2022-10-10

Project Plan

I am going to use this data to compare the effect of the vaccine on COVID-19 rates and COVID-19 Deaths.

##Import and Clean I store the data files on my HDD to speed up load times. That is the wild If scripts. After that, I make all NA's 0 and remove unnessicary rows and columns, and combine all state data into 1 row.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
#This is to speed up download times if you have to rerun the entire process.

if (!file.exists("time_series_covid19_confirmed_global.csv")){
  Confirmed <- read.csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19
  write.csv(Confirmed, "time_series_covid19_confirmed_global.csv")
} else {
  Confirmed <- read.csv("time_series_covid19_confirmed_global.csv")
  Confirmed <- Confirmed[-c(1)]
}

if (!file.exists("time_series_covid19_deaths_global.csv")){
  Deaths <- read.csv("https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_da
   write.csv(Deaths, "time_series_covid19_deaths_global.csv")
} else {
  Deaths <- read.csv("time_series_covid19_deaths_global.csv")
  Deaths <- Deaths[-c(1)]
}

if (!file.exists("time_series_covid19_vaccine_doses_admin_US.csv")){
  Vaccine <- read.csv("https://raw.githubusercontent.com/govex/COVID-19/master/data_tables/vaccine_data
  Vaccine  <- Vaccine  %>% replace(is.na(.), 0)
   write.csv(Vaccine, "time_series_covid19_vaccine_doses_admin_US.csv")
} else {
  Vaccine <- read.csv("time_series_covid19_vaccine_doses_admin_US.csv")
  Vaccine <- Vaccine[-c(1)]
```

```
}

#Clean Data by remvoing unnessicary rows and coulums
Confirmed <- Confirmed %>% select(-c("UID", "iso2", "iso3", "code3", "FIPS", "Admin2", "Country_Region"
Deaths <- Deaths %>% select(-c("UID", "code3", "FIPS", "Lat", "Long_"))
Vaccine <- Vaccine %>% select(-c("UID","iso2", "iso3", "FIPS", "Admin2", "Lat", "Long_", "Combined_Key"

Confirmed <- Confirmed %>% group_by(Province_State) %>% summarize_if(is.numeric, sum)
Deaths <- Deaths %>% group_by(Province_State) %>% summarize_if(is.numeric, sum)
Confirmed <- Confirmed[-c(10,14),]
Vaccine <- Vaccine[-c(57:61),]
```

##Prep data for merge I will now pivot longer all the data and make each date it's own column. Then I will merge the data by date and by State into 1 giant data set.

```
library(lubridate)
```

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

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

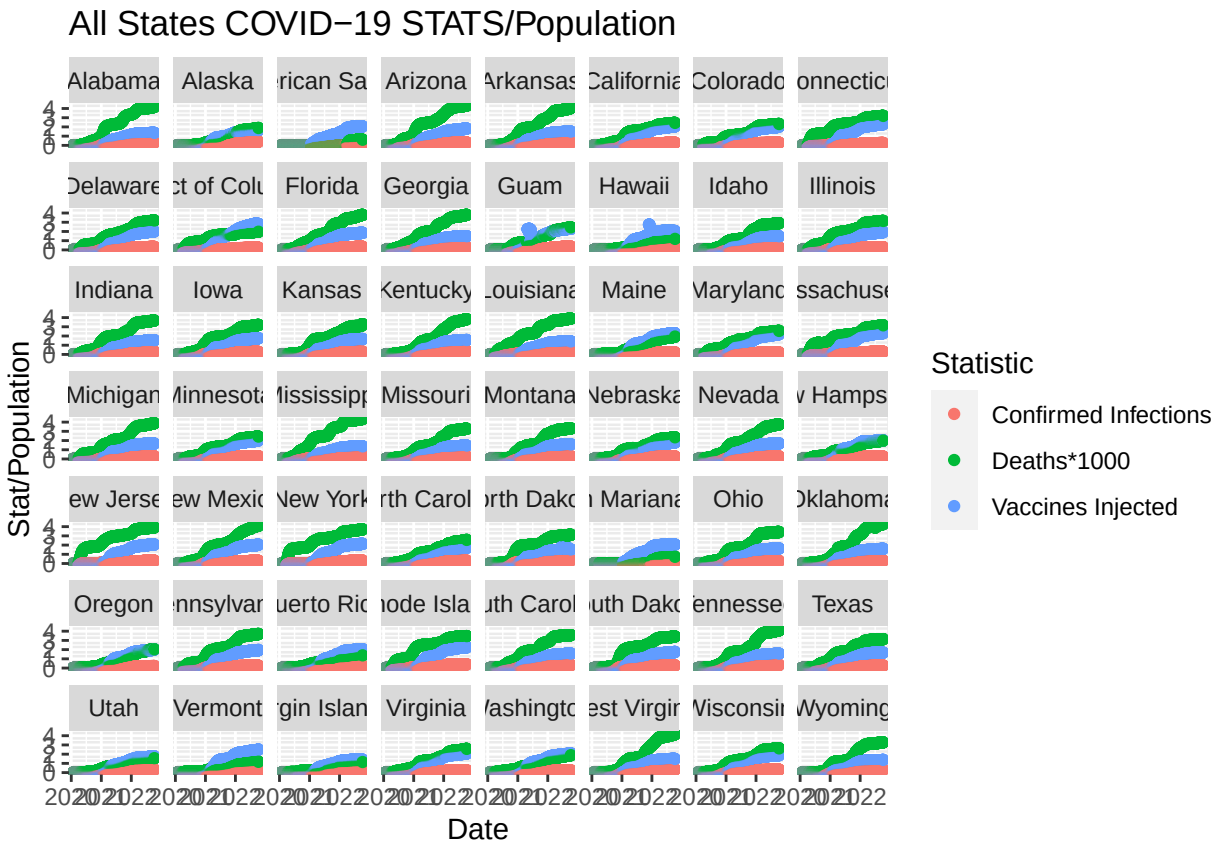
```
Confirmed <- pivot_longer(Confirmed, cols = -c("Province_State"), names_to = "Date", values_to = 'Confi
Confirmed$Date <- mdy(Confirmed$Date)

Deaths <- pivot_longer(Deaths, cols = -c("Province_State", "Population"), names_to = "Date", values_to =

Deaths$Date <- mdy(Deaths$Date)

Vaccine <- pivot_longer(Vaccine, cols = -c("Province_State"), names_to = "Date", values_to = 'Vaccinate
Vaccine$Date <- ymd(Vaccine$Date)

Combined <- right_join(Deaths, Confirmed, Vaccine, by = c("Province_State","Date"))
Combined <- left_join(Combined, Vaccine, by = c("Province_State","Date"))
Combined  <- Combined  %>% replace(is.na(.), 0)
Combined$Vaccinated <- Combined$Vaccinated/Combined$Population
Combined$"Deaths*1000" <- Combined$Deaths/Combined$Population*1000
Combined$Confirmed <- Combined$Confirmed/Combined$Population
Combined$Deaths <- NULL
```

##Graph The Data

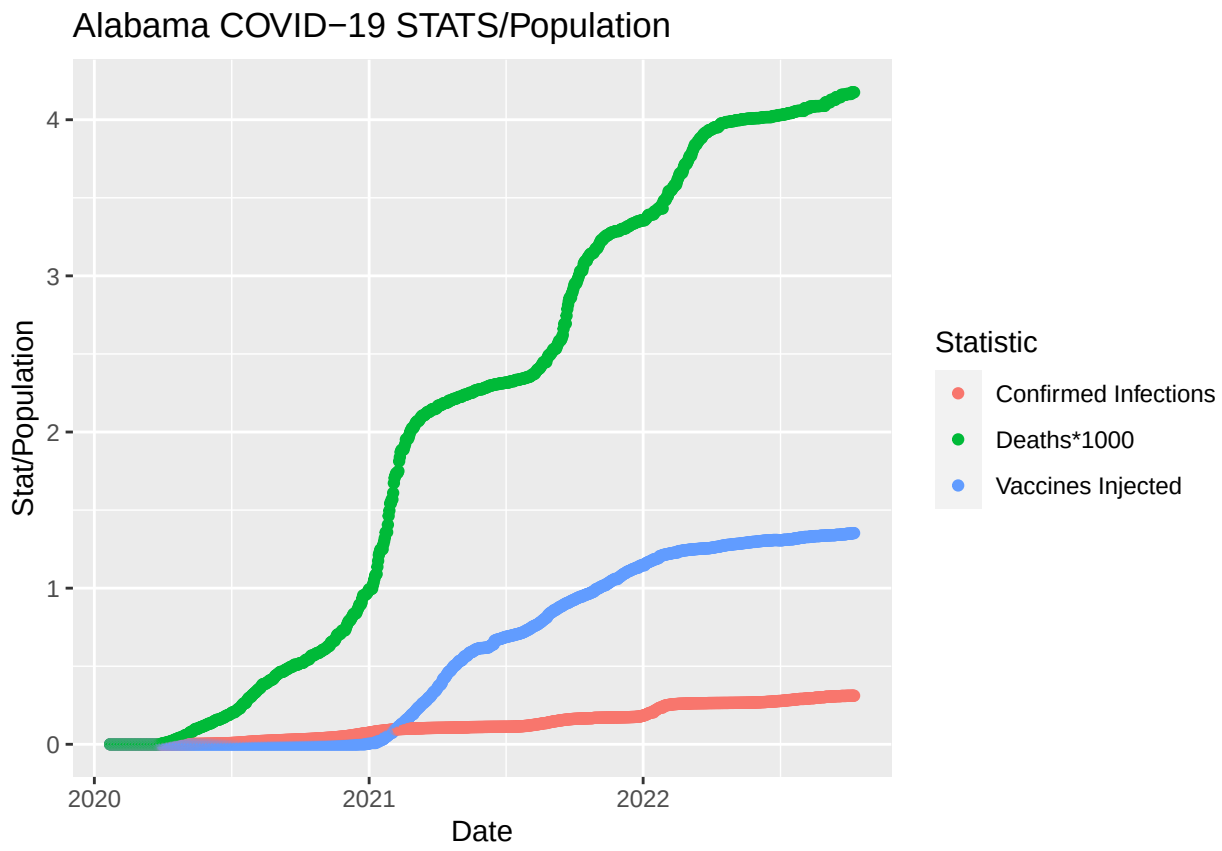
I will start with graphing the whole US Data with all 3 Columns. Please note deaths are multiplied by 1000 to be visable.

```
Graphable <- Combined
tmp <- colnames(Graphable)

tmp <- c("Province_State", "Population", "Date", "Confirmed Infections", "Vaccines Injected", "Deaths*1
colnames(Graphable) <- tmp
```

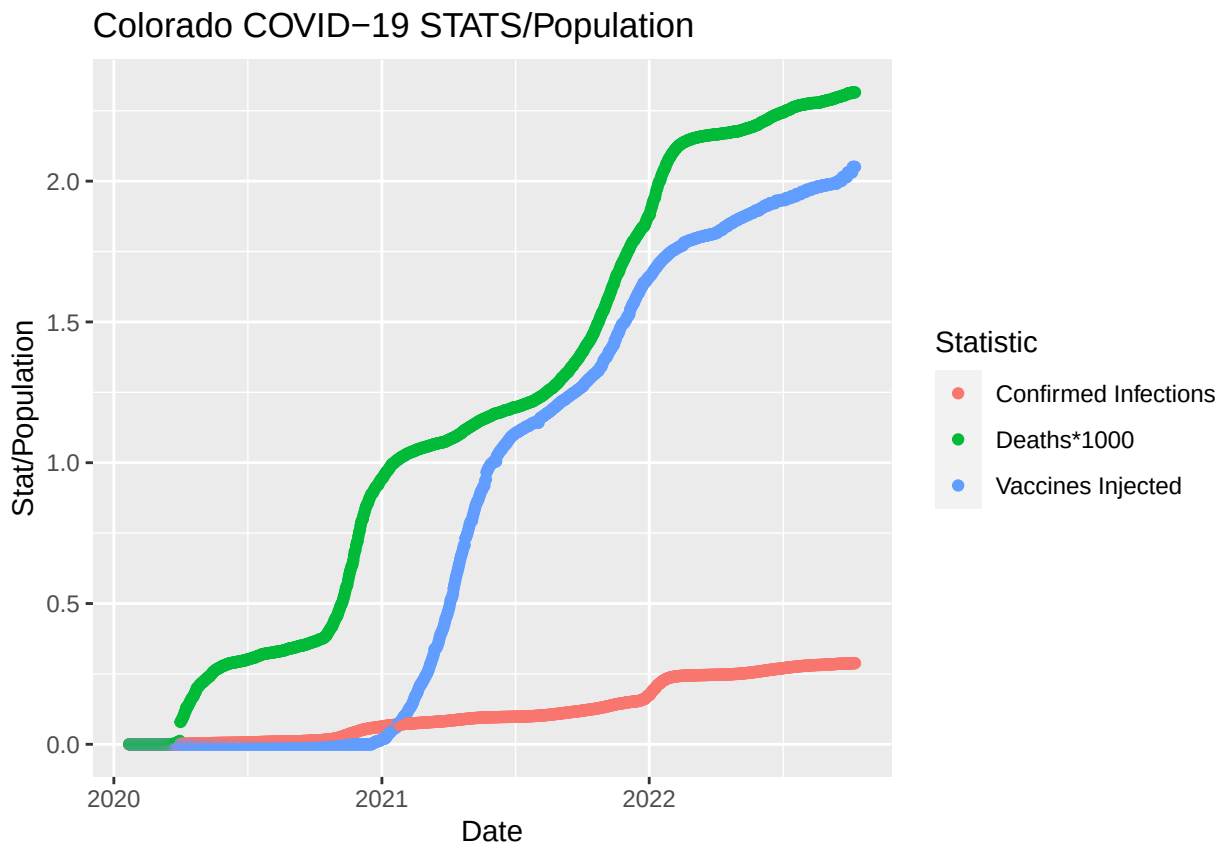
```
Graphable <- pivot_longer(Graphable, cols = -c(Province_State, Population, Date), names_to = "Type", val

Alabama <- Graphable %>% filter(Province_State == "Alabama")
Colorado <- Graphable %>% filter(Province_State == "Colorado")
Florida <- Graphable %>% filter(Province_State == "Florida")
California <- Graphable %>% filter(Province_State == "California")

color_key <- c("Confirmed" = "red", "deaths" = "green", "Vaccinated"= "blue")

ggplot(data = Graphable, mapping = aes(x = Date, y = Vals, color = Type)) + geom_point() + labs(x = "Da
```



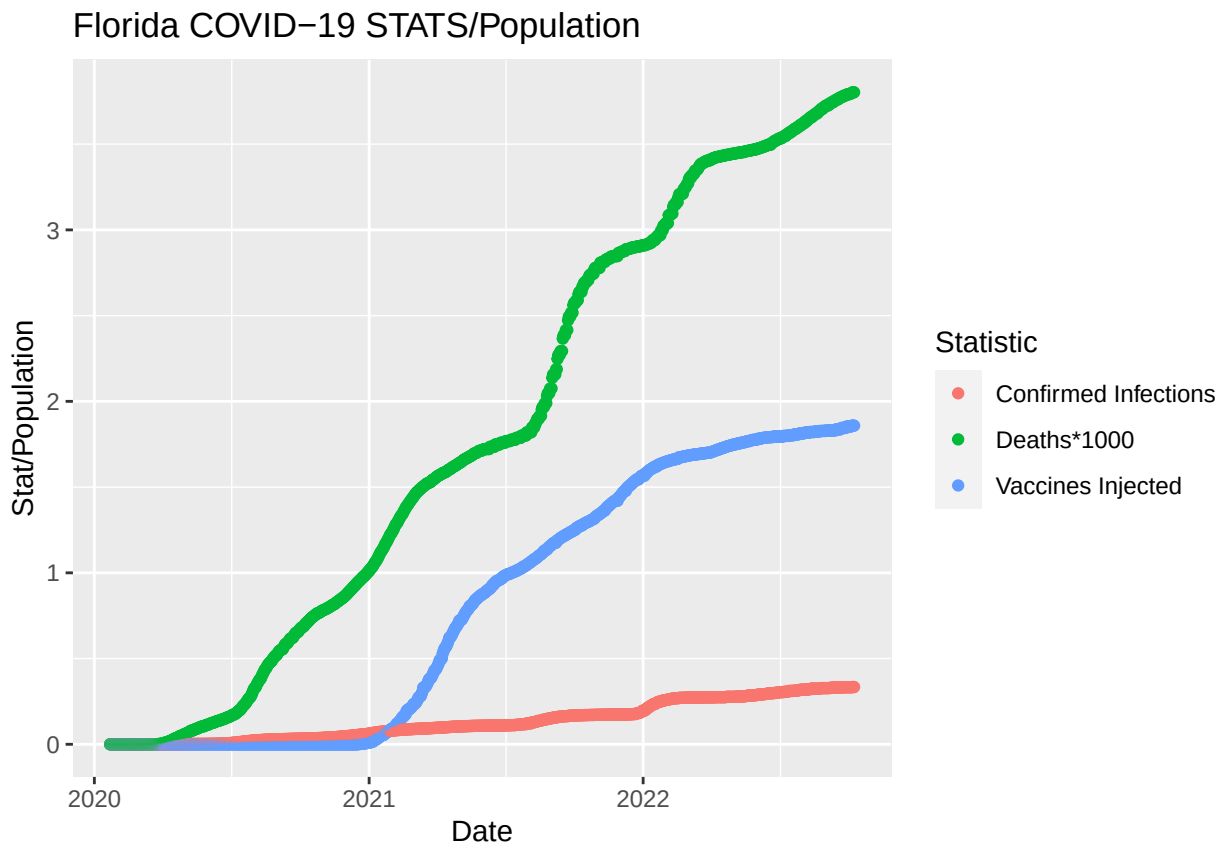

```
ggplot(data = Alabama, mapping = aes(x = Date, y = Vals, color = Type)) + geom_point() + labs(x = "Date
```



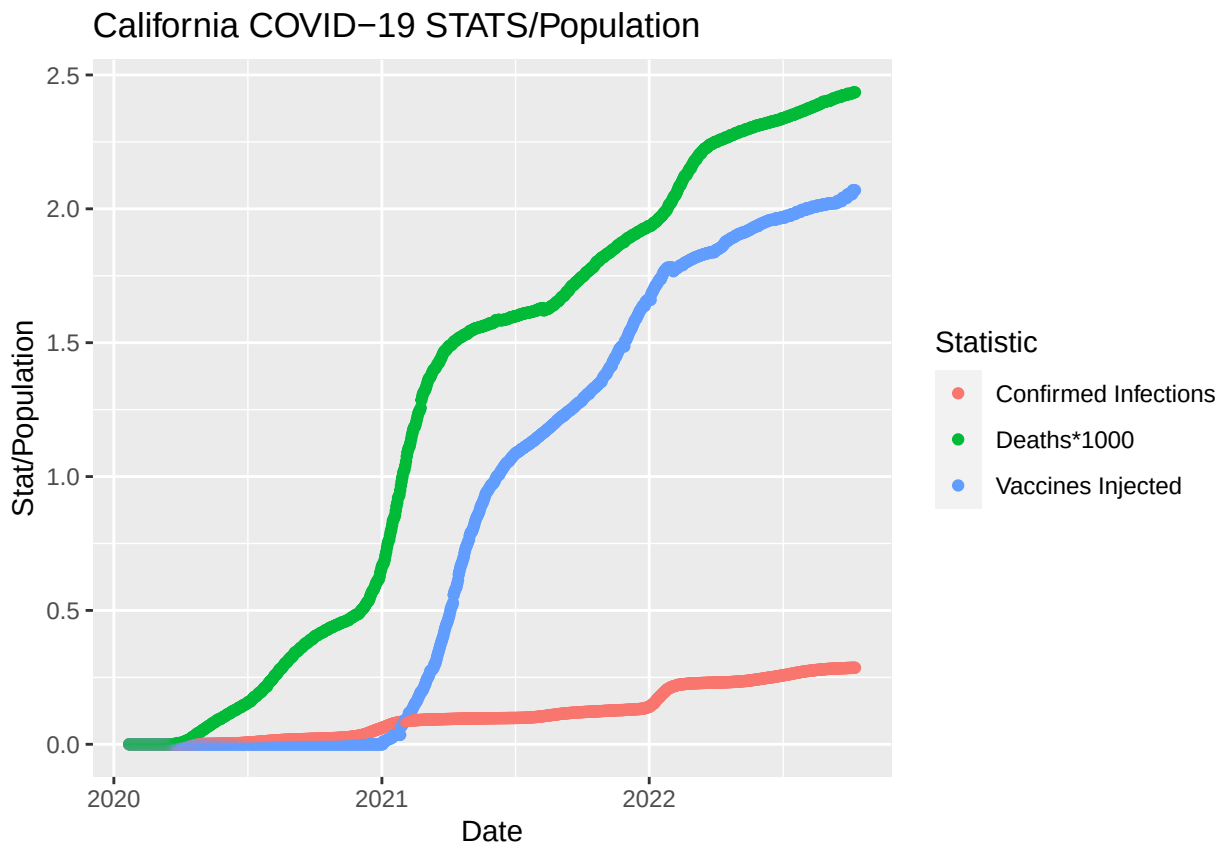

```
ggplot(data = Colorado, mapping = aes(x = Date, y = Vals, color = Type)) + geom_point()+ labs(x = "Date
```




```
ggplot(data = Florida, mapping = aes(x = Date, y = Vals, color = Type)) + geom_point()+ labs(x = "Date"
```




```
ggplot(data = California, mapping = aes(x = Date, y = Vals, color = Type)) + geom_point() + labs(x = "Da
```



##Modeling

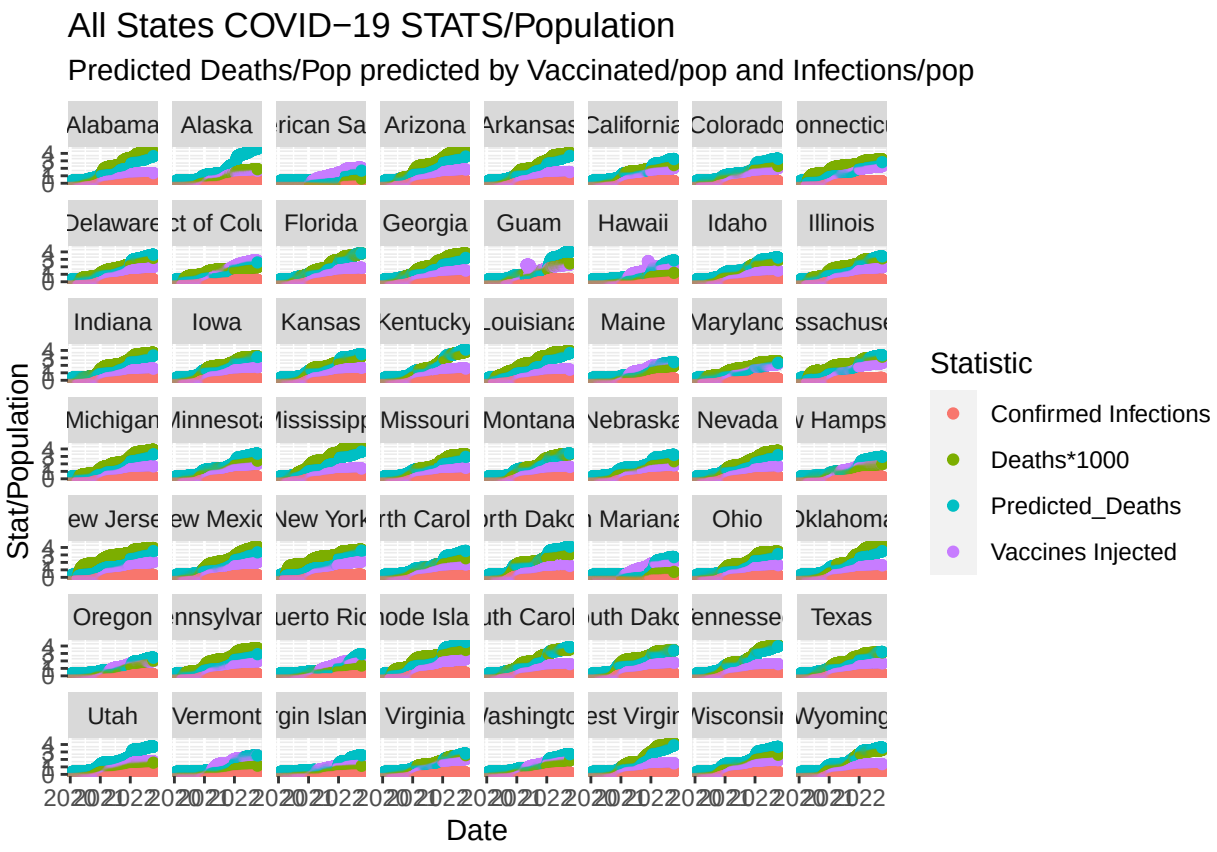
So from this data it's obvious that vaccines had an impact on deaths. I think this is a linear relationship and can be modeled. I'm going to try and use LM to find the relationship between Cases and Vaccination and Deaths and Vaccination.

```
tmp <- colnames(Combined)
tmp[6] <- "Deaths"
colnames(Combined) <- tmp
model1 <- lm(formula = Deaths ~ Confirmed + Vaccinated, data = Combined)
model2 <- lm(formula = Confirmed ~ Vaccinated, data = Combined)
```

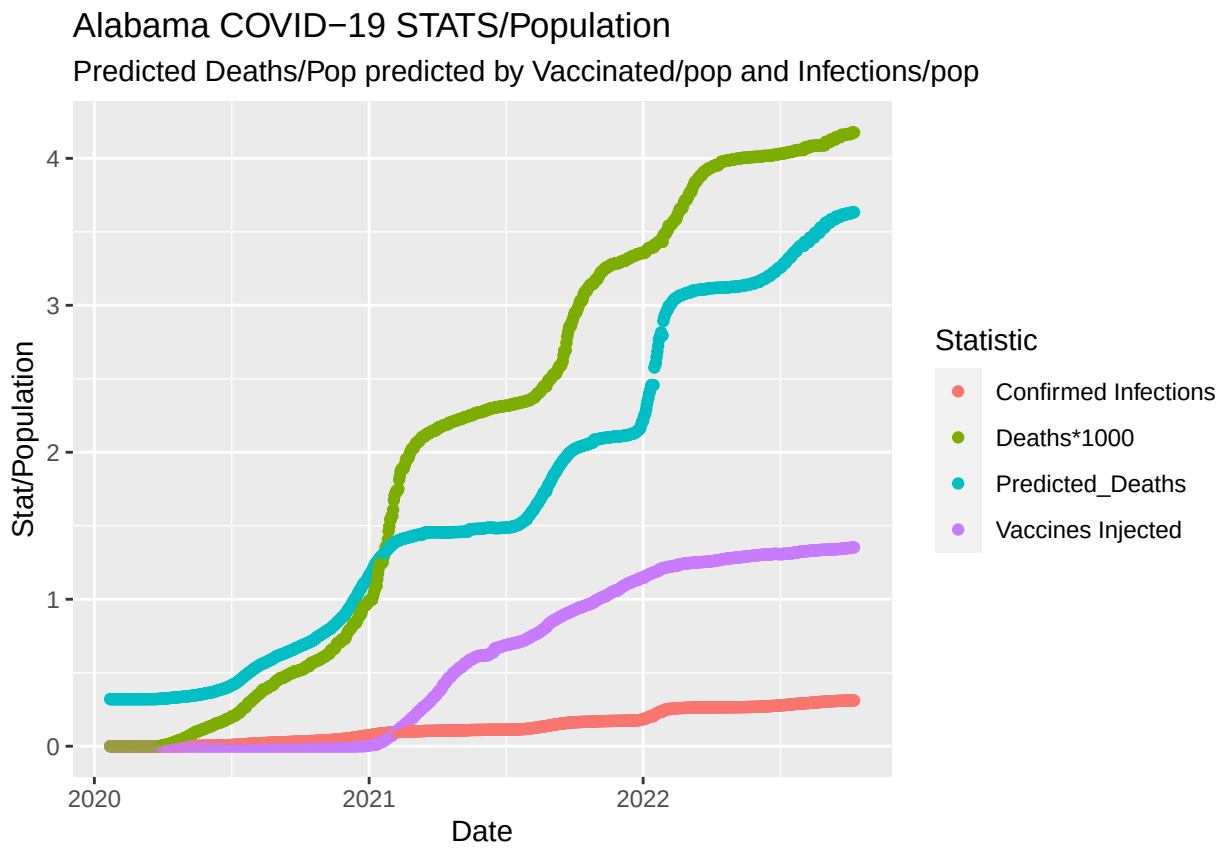
##Plotting the model I'll start with plotting the first model which is Deaths predicted by confirmed and vaccinated

```
Graphable <- Combined
Graphable$Predicted_Deaths <- predict(model1, Graphable)
tmp <- colnames(Graphable)

tmp <- c("Province_State", "Population", "Date", "Confirmed Infections", "Vaccines Injected", "Deaths*1
colnames(Graphable) <- tmp
Graphable <- pivot_longer(Graphable, cols = -c(Province_State, Population, Date), names_to = "Type", va
Alabama <- Graphable %>% filter(Province_State == "Alabama")
Colorado <- Graphable %>% filter(Province_State == "Colorado")
Florida <- Graphable %>% filter(Province_State == "Florida")
California <- Graphable %>% filter(Province_State == "California")
```

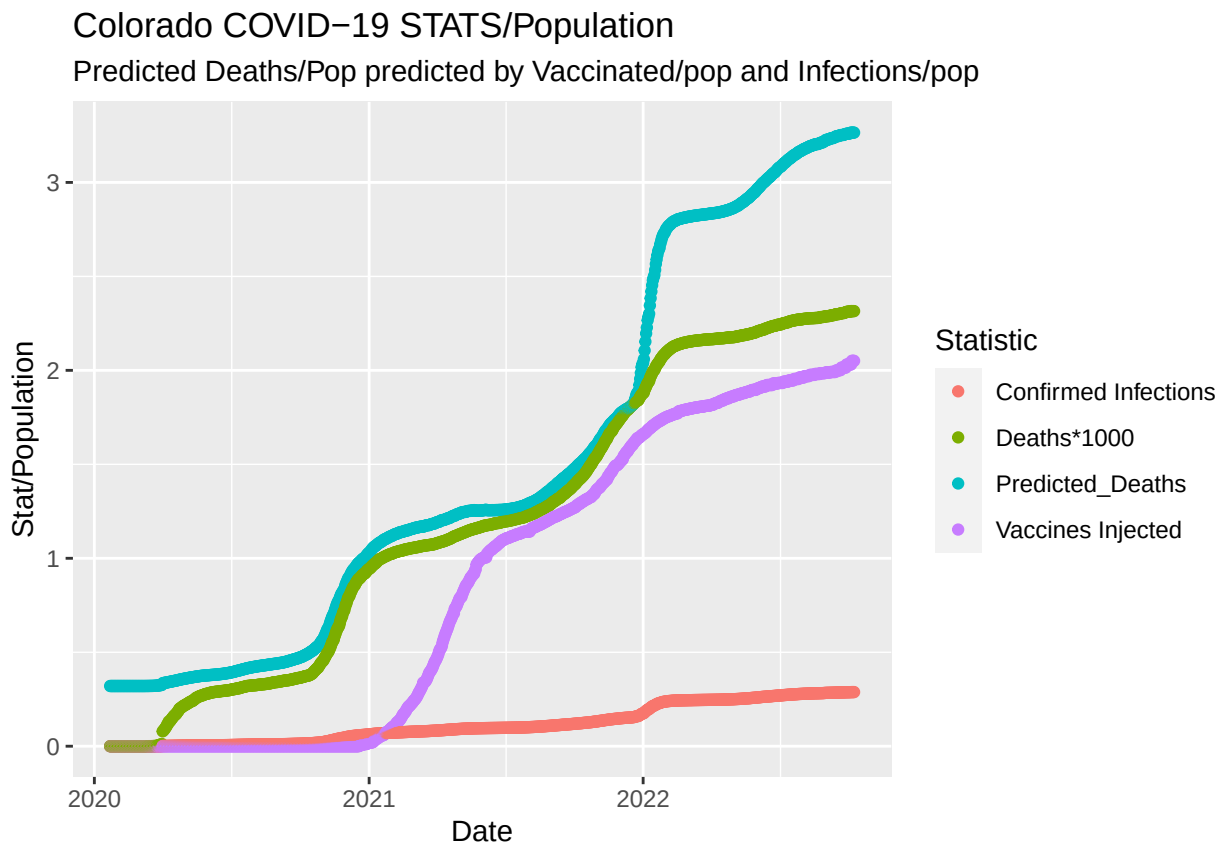
```
ggplot(data = Graphable, mapping = aes(x = Date, y = Vals, color = Type)) + geom_point() + labs(x = "Da
```



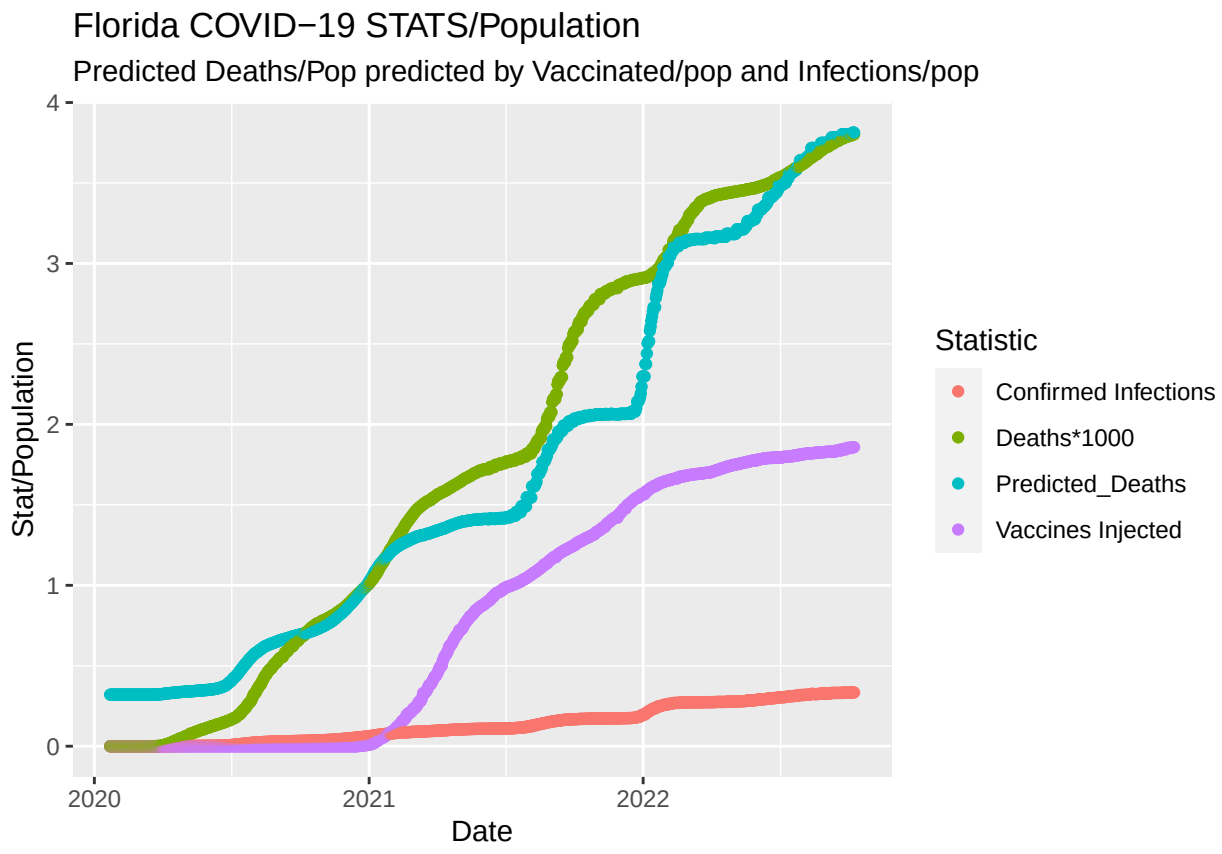

```
ggplot(data = Alabama, mapping = aes(x = Date, y = Vals, color = Type)) + geom_point() + labs(x = "Date
```



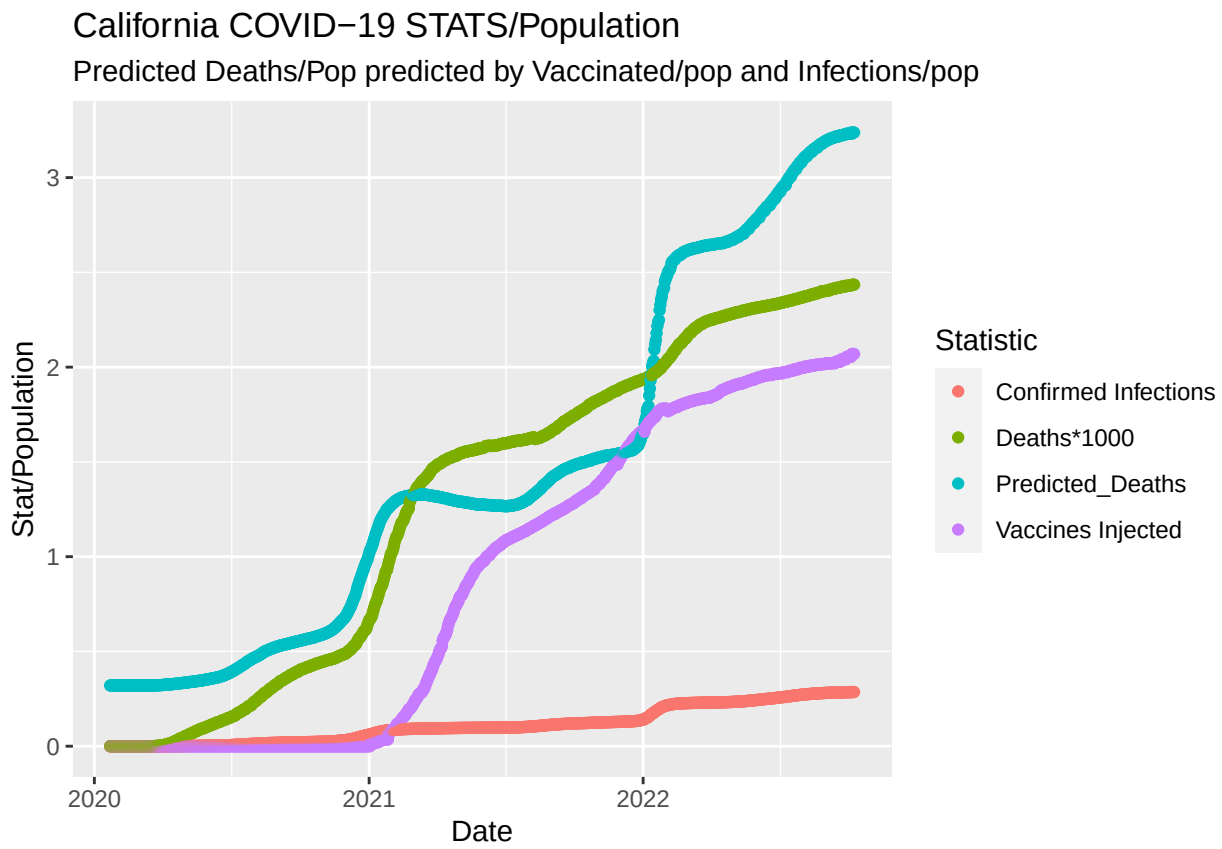

```
ggplot(data = Colorado, mapping = aes(x = Date, y = Vals, color = Type)) + geom_point()+ labs(x = "Date
```




```
ggplot(data = Florida, mapping = aes(x = Date, y = Vals, color = Type)) + geom_point()+ labs(x = "Date"
```



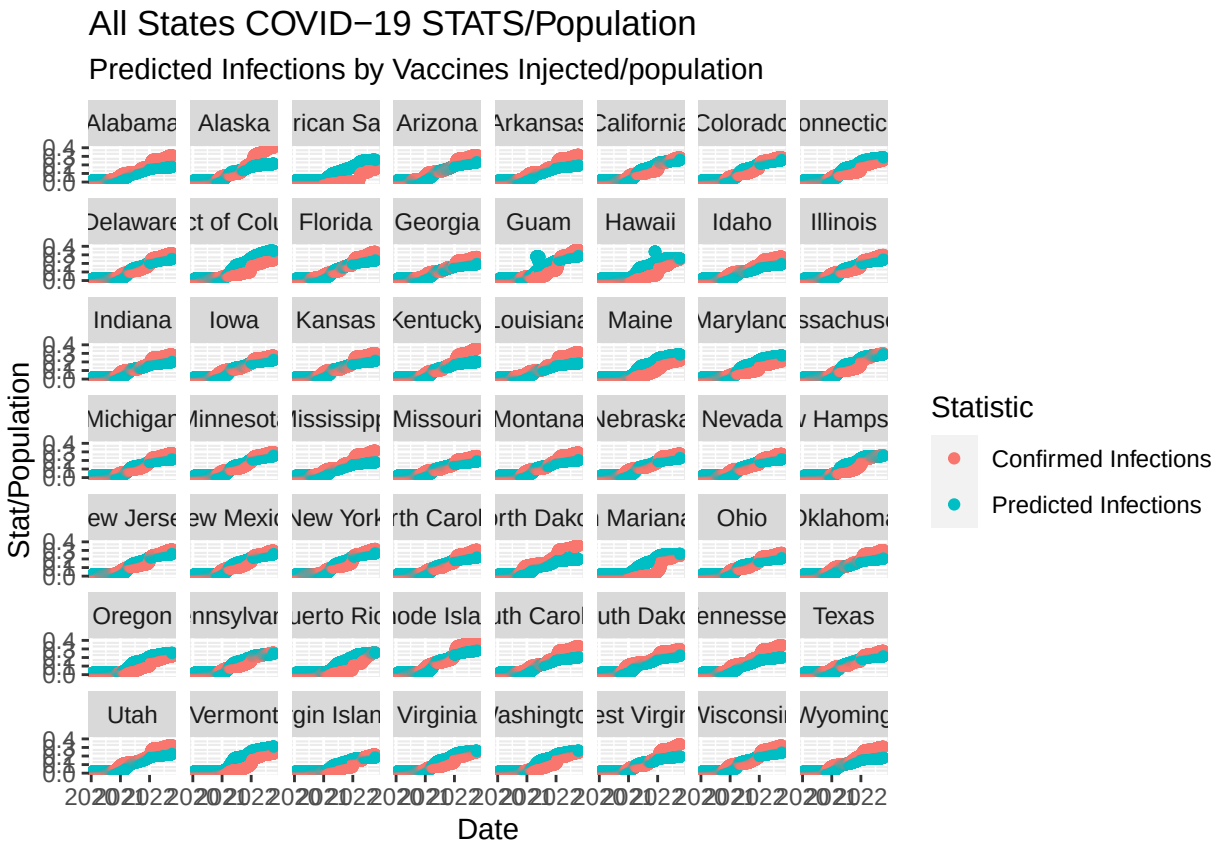

```
ggplot(data = California, mapping = aes(x = Date, y = Vals, color = Type)) + geom_point() + labs(x = "Da
```



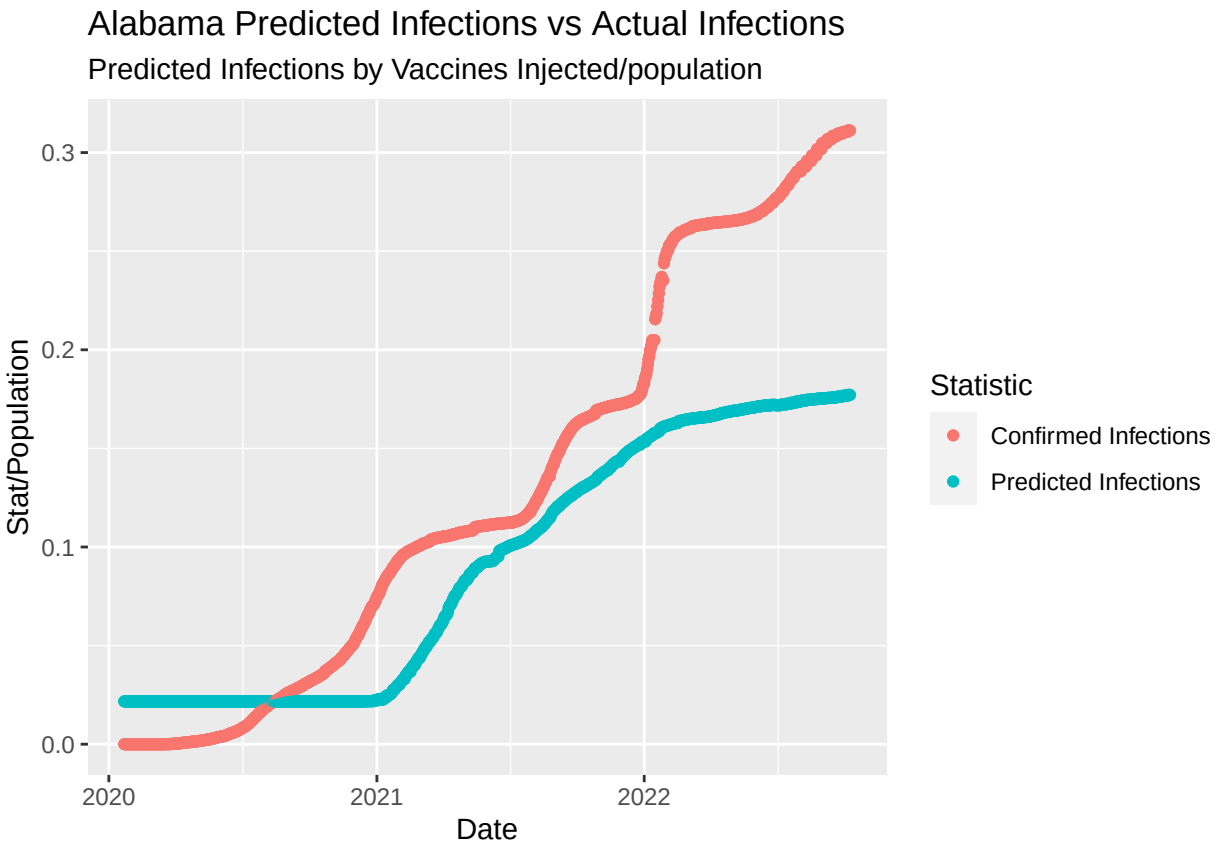
That is a pretty close match, at least visually.

And Finally, I'm going to look at the plot for Vaccinations vs new Infections

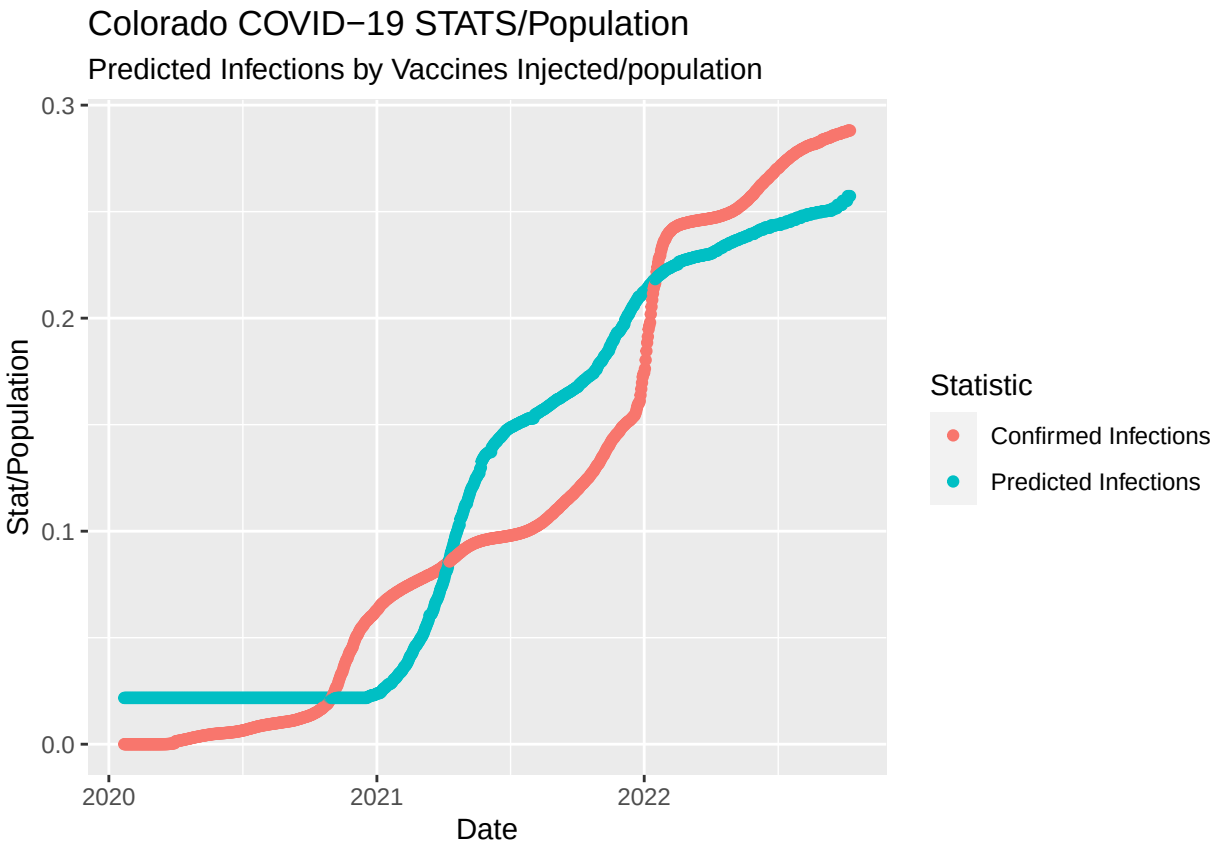
```
Graphable <- Combined
Graphable$Predicted_Infections <- predict(model2, Graphable)

tmp <- colnames(Graphable)
tmp <- c("Province_State", "Population", "Date", "Confirmed Infections", "Vaccines_Injected", "Deaths*1
colnames(Graphable) <- tmp

Graphable <- pivot_longer(Graphable, cols = -c(Province_State, Population, Date, "Deaths*1000", Vaccines

Alabama <- Graphable %>% filter(Province_State == "Alabama")
Colorado <- Graphable %>% filter(Province_State == "Colorado")
Florida <- Graphable %>% filter(Province_State == "Florida")
California <- Graphable %>% filter(Province_State == "California")

ggplot(data = Graphable, mapping = aes(x = Date, y = Vals, color = Type)) + geom_point() + labs(x = "Da
```



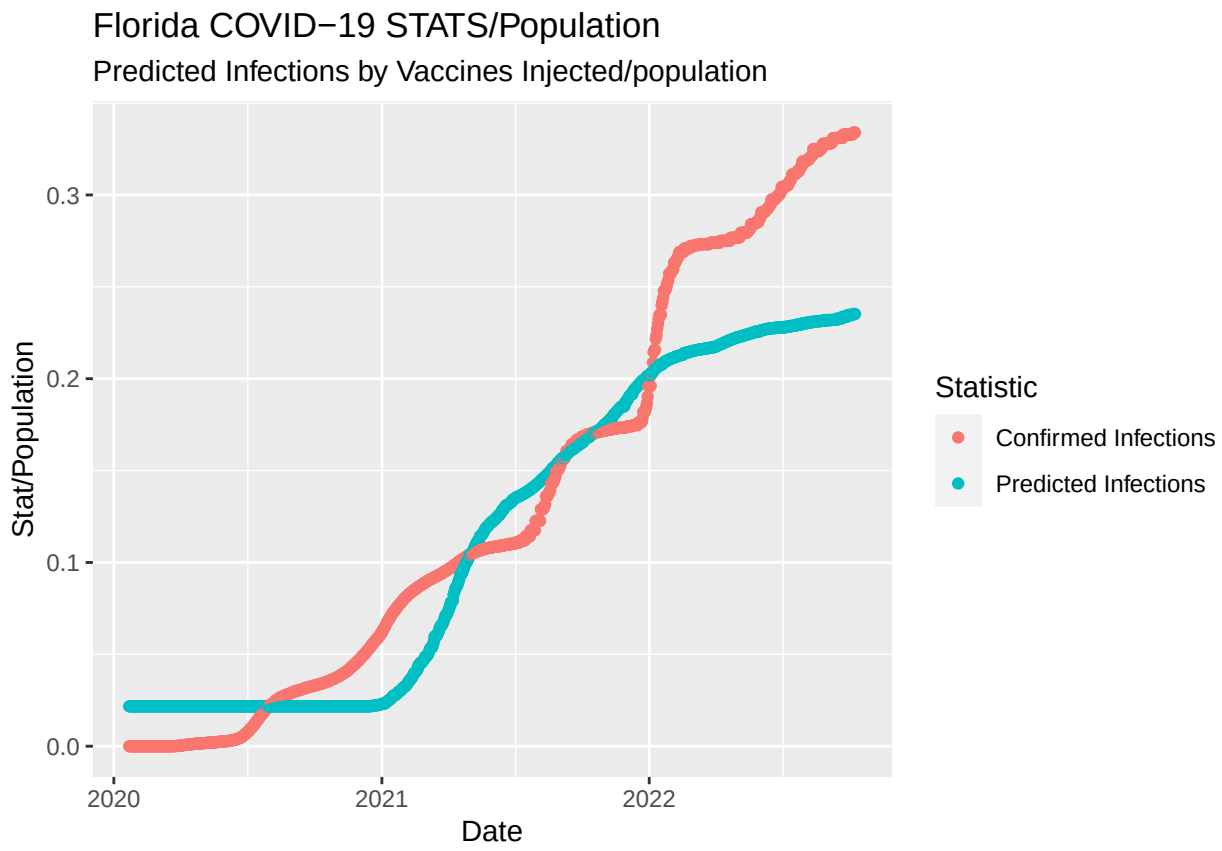

```
ggplot(data = Alabama, mapping = aes(x = Date, y = Vals, color = Type)) + geom_point() + labs(x = "Date
```



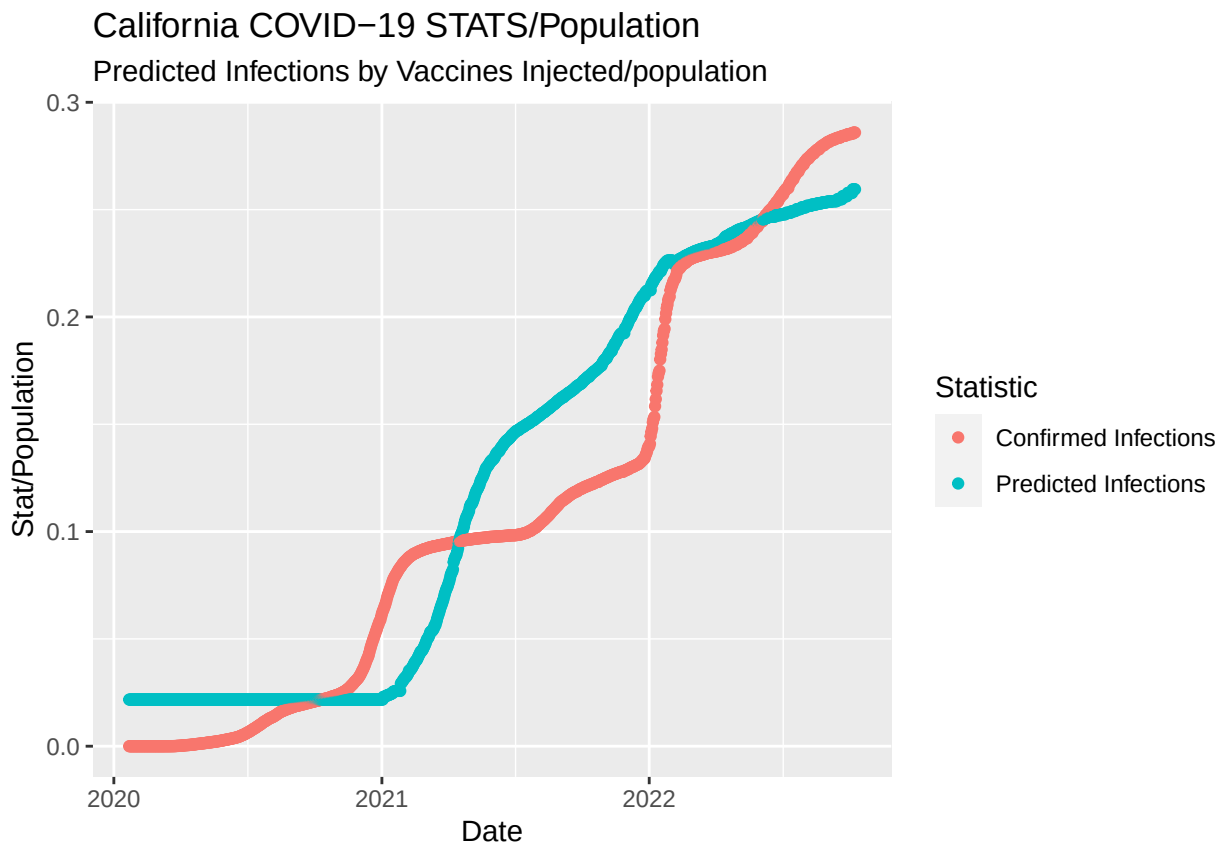

```
ggplot(data = Colorado, mapping = aes(x = Date, y = Vals, color = Type)) + geom_point()+ labs(x = "Date
```




```
ggplot(data = Florida, mapping = aes(x = Date, y = Vals, color = Type)) + geom_point()+ labs(x = "Date"
```




```
ggplot(data = California, mapping = aes(x = Date, y = Vals, color = Type)) + geom_point() + labs(x = "Da
```



Wow, Confirmed Infections is very close to predicted infections as predicted only by vaccinations.

##Biases The largest bias is selection bias. We know that the confirmed cases was way lower than the actual cases, at some times by an order of magnitude. We also know that governments manipulated the data and activly prevented tests from being given out. With that, the data is still conclusive on all points.

##Conclusions

Vaccines Work to reduce the infection rate and the death rate.

```
sessionInfo()
```

```
## R version 4.2.1 (2022-06-23 ucrt)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 22621)
##
## 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
##
## 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_1.0.9
##  [5] purrr_0.3.4     readr_2.1.2     tidyr_1.2.0     tibble_3.1.7
##  [9] ggplot2_3.3.6   tidyverse_1.3.2
##
## loaded via a namespace (and not attached):
##  [1] tidyselect_1.1.2    xfun_0.31           haven_2.5.0
##  [4] gargle_1.2.0        colorspace_2.0-3    vctrs_0.4.1
##  [7] generics_0.1.3      htmltools_0.5.3     yaml_2.3.5
## [10] utf8_1.2.2          rlang_1.0.4         pillar_1.8.0
## [13] glue_1.6.2          withr_2.5.0         DBI_1.1.3
## [16] dbplyr_2.2.1        modelr_0.1.8        readxl_1.4.0
## [19] lifecycle_1.0.1     munsell_0.5.0       gtable_0.3.0
## [22] cellranger_1.1.0    rvest_1.0.2         evaluate_0.15
## [25] labeling_0.4.2      knitr_1.39          tzdb_0.3.0
## [28] fastmap_1.1.0       fansi_1.0.3         highr_0.9
## [31] broom_1.0.0         scales_1.2.0        backports_1.4.1
## [34] googlesheets4_1.0.0 jsonlite_1.8.0      farver_2.1.1
## [37] fs_1.5.2            hms_1.1.1           digest_0.6.29
## [40] stringi_1.7.8       grid_4.2.1          cli_3.3.0
## [43] tools_4.2.1         magrittr_2.0.3      crayon_1.5.1
## [46] pkgconfig_2.0.3     ellipsis_0.3.2      xml2_1.3.3
## [49] reprex_2.0.1        googledrive_2.0.0   assertthat_0.2.1
## [52] rmarkdown_2.14      httr_1.4.3          rstudioapi_0.13
## [55] R6_2.5.1            compiler_4.2.1
```