Covid EDA

2022-05-30

nrow(vaccine_data) #1.45 mil nrow(uscovidweekly) #6014 #uscovidweekly - #take out delimiter (,) and get just county, #append "county", #then join by county name with vaccine_data

#then start graphing stuff together, perhaps the "new cases per 100k in uscovidweekly" and a measure of %vaccinated for an age group from vaccine_data.

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:lubridate':
##
##      hour, isoweek, mday, minute, month, quarter, second, wday, week,
##      yday, year
```

```
## The following objects are masked from 'package:dplyr':
##
##      between, first, last
```

```
## The following object is masked from 'package:purrr':
##
##      transpose
```

```
library(stringr)
library(dplyr)

stripuscovidweekly<-as.data.table(uscovidweekly)
#strip delimiter and put into new columns
stripuscovidweekly[, c("Recip_County", "State", "Country") := tstrsplit(key, ", ", fixed=TRUE)]

stripuscovidweekly$Recip_County <- paste0(stripuscovidweekly$Recip_County, " County")

#rename 'week' as 'Date' to match vaccine data
stripuscovidweekly<-rename(stripuscovidweekly,Date = week)

#Now we have "___ County" which will allow us to join with the vaccine data


#Filter out non-NY vaccine data:
nyvaccine_data<-vaccine_data%>%
  filter(Recip_State=="NY")
```

```r
#turn "Date" column to date format in vaccine data
nyvaccine_data$Date <- mdy(nyvaccine_data$Date)

monnyvaccine<-nyvaccine_data %>%
  mutate(weekday = wday(Date))%>%
  filter(weekday==2)

##turn "Date" column to date format in covid data
stripuscovidweekly$Date<- ymd(stripuscovidweekly$Date)


joined<-monnyvaccine %>%
  inner_join(stripuscovidweekly ,by =c("Recip_County","Date"))


testthis<-monnyvaccine%>%
  select(Date, Recip_County, Booster_Doses)
#write.csv(joined,"/Users/grantnielson/Documents/Research\\joineddata.csv", row.names = FALSE)
#^^saved it in the wrong spot but I moved it.

getwd()
```

```
## [1] "/Users/grantnielson/Documents/Research"
```

```r
dose1vsnewcases<-joined%>%
  select(Administered_Dose1_Pop_Pct,new_cases_per_100k)

#check if summed or not
difference<-joined%>%
  select(Administered_Dose1_Recip,Administered_Dose1_Pop_Pct,new_cases_per_100k,Date,Recip_County)%>%
  arrange(Recip_County,Date)

#difference %>%
  #lag(Administered_Dose1_Recip)%>%
  #mutate(new_vaccinations=(diff(difference$Administered_Dose1_Recip)))

#lag(difference$Administered_Dose1_Recip)


library(data.table)
DT = data.table(difference)




new<-DT[ , list(new_cases_per_100k,Administered_Dose1_Pop_Pct,Administered_Dose1_Recip ,Date,Recip_Coun
  mutate(new_vax_this_week=lag(new_vax))
```

```
## Warning in as.data.table.list(jval, .named = NULL): Item 6 has 3099 rows but
## longest item has 3100; recycled with remainder.
```

```
newjoined<-subset(new, select = -c(new_vax)) #this the one




updatedjoin<-joined %>%
  inner_join(newjoined, by =c("Recip_County","Date"))



#save the new
#write.csv(updatedjoin,"/Users/grantnielson/Documents/Research\\updatedjoineddata.csv", row.names = FAL
```
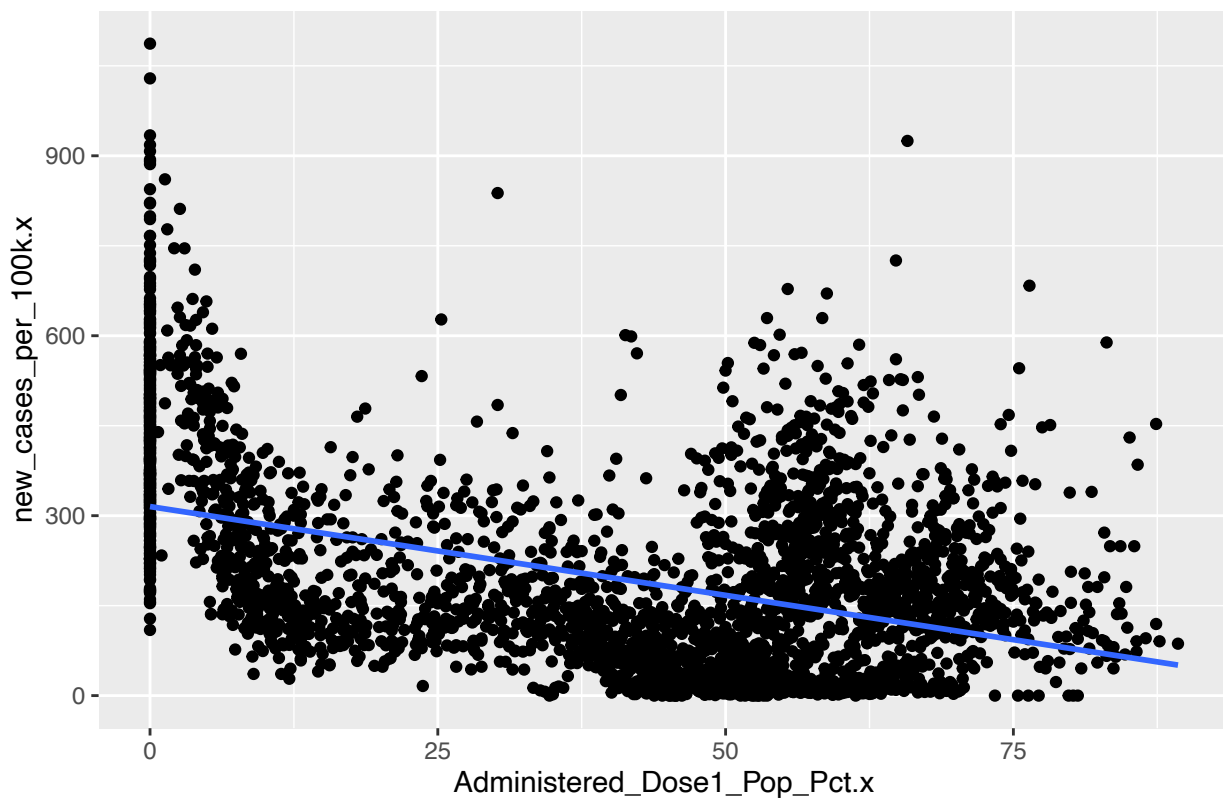
```
ggplot(data=updatedjoin, mapping=aes(x=Administered_Dose1_Pop_Pct.x,y=new_cases_per_100k.x)) + geom_poin
```

```
## `geom_smooth()` using formula 'y ~ x'
```

Vaccines vs Cases


```
#5/30/22----------------------------

#y axis is Administered_Dose1_Pop_Pct.x
ggplot(data=updatedjoin, mapping=aes(x=Administered_Dose1_Pop_Pct.x,y=new_cases_per_100k.x)) + geom_poin
```
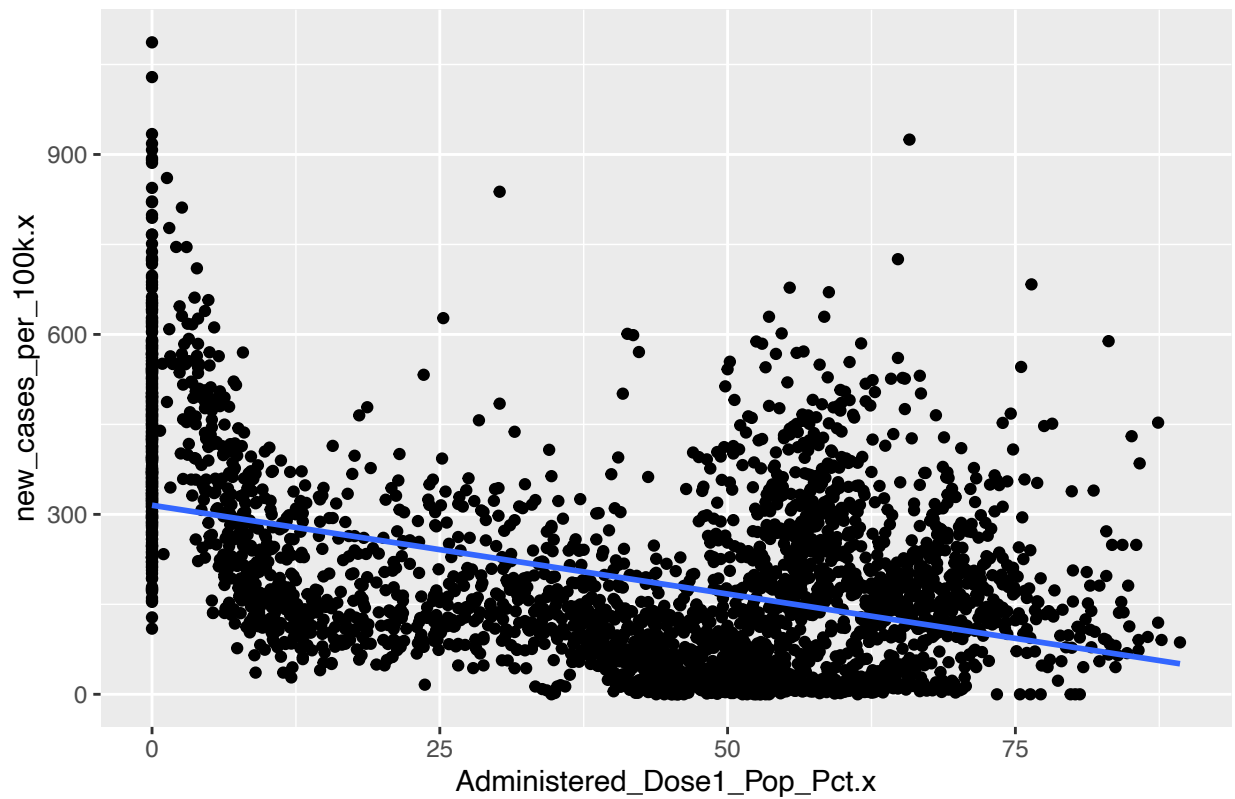
```
## `geom_smooth()` using formula 'y ~ x'
```
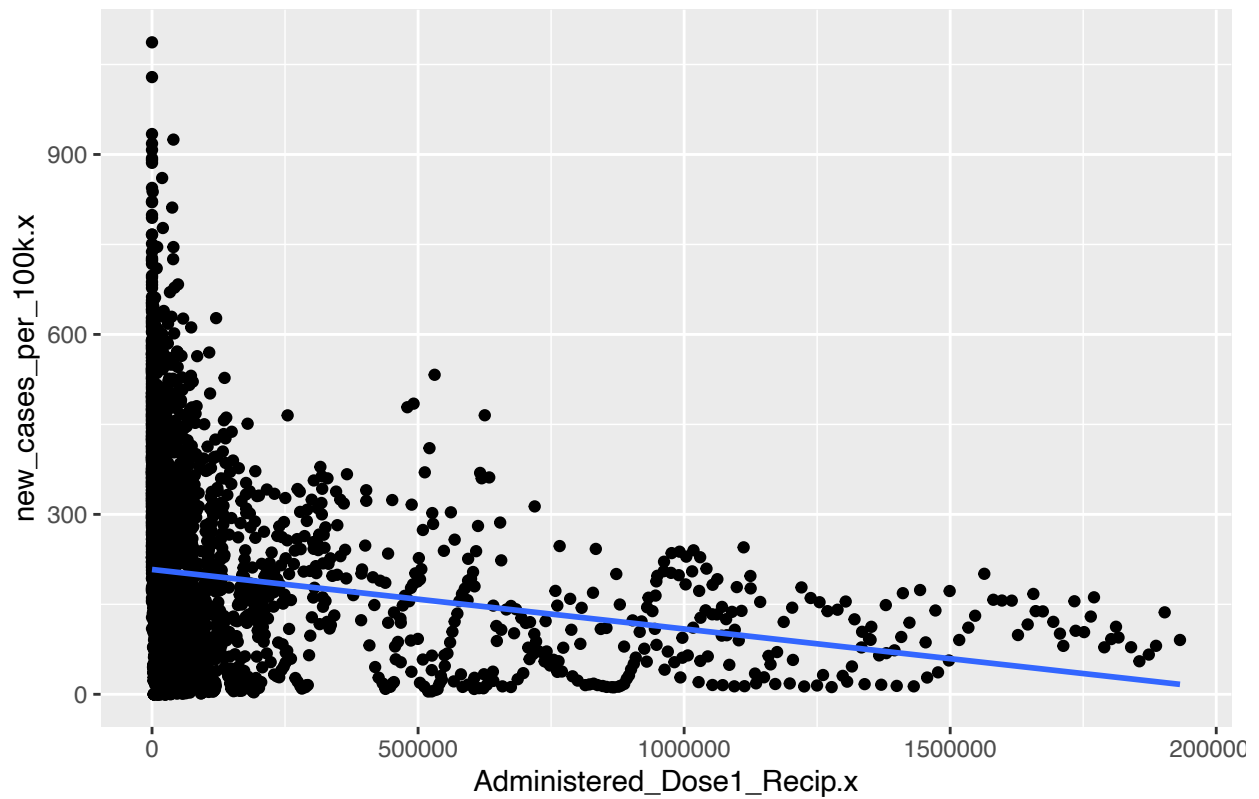
% of county vaxed vs new cases per 100k


```
#updatedjoin
ggplot(data=updatedjoin, mapping=aes(x=Administered_Dose1_Recip.x,y=new_cases_per_100k.x)) + geom_point

## `geom_smooth()` using formula 'y ~ x'
```
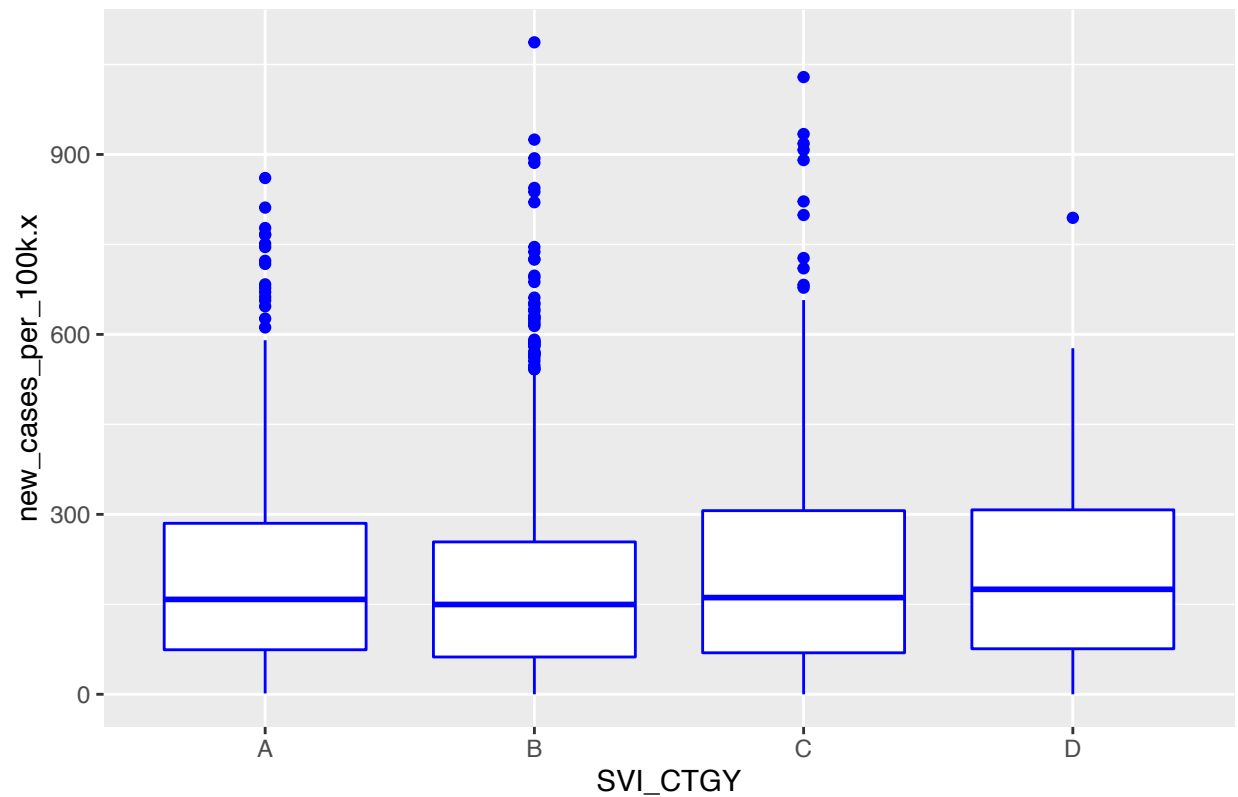
administered dose1 recip vs new cases per 100k


```
ggplot(data=updatedjoin, mapping=aes(x=SVI_CTGY,y=new_cases_per_100k.x)) + geom_boxplot(color="blue")+g
```

```
## 'geom_smooth()' using formula 'y ~ x'
```
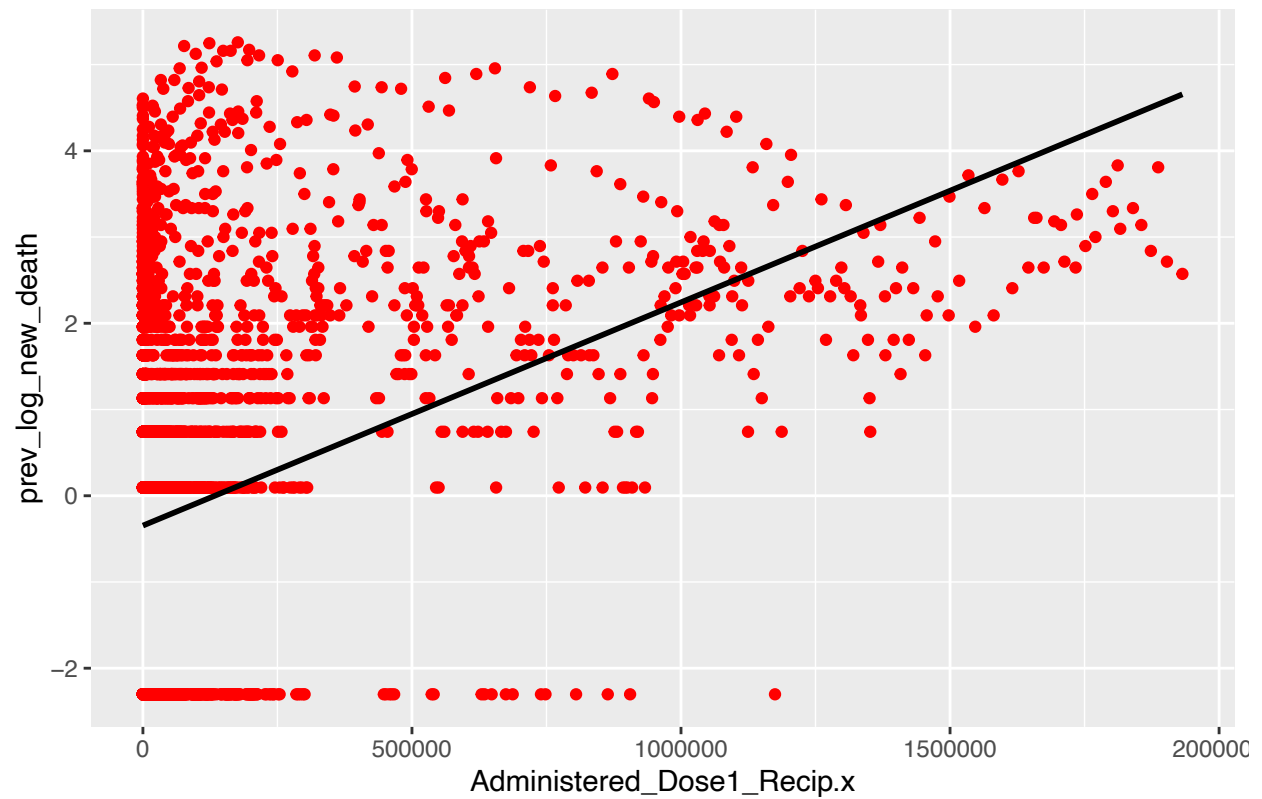
Social Vulnerability Index Category vs new cases per 100k


```
#prevlognewdeath
ggplot(data=updatedjoin, mapping=aes(x=Administered_Dose1_Recip.x,y=prev_log_new_death)) + geom_point(co
```

```
## 'geom_smooth()' using formula 'y ~ x'
```
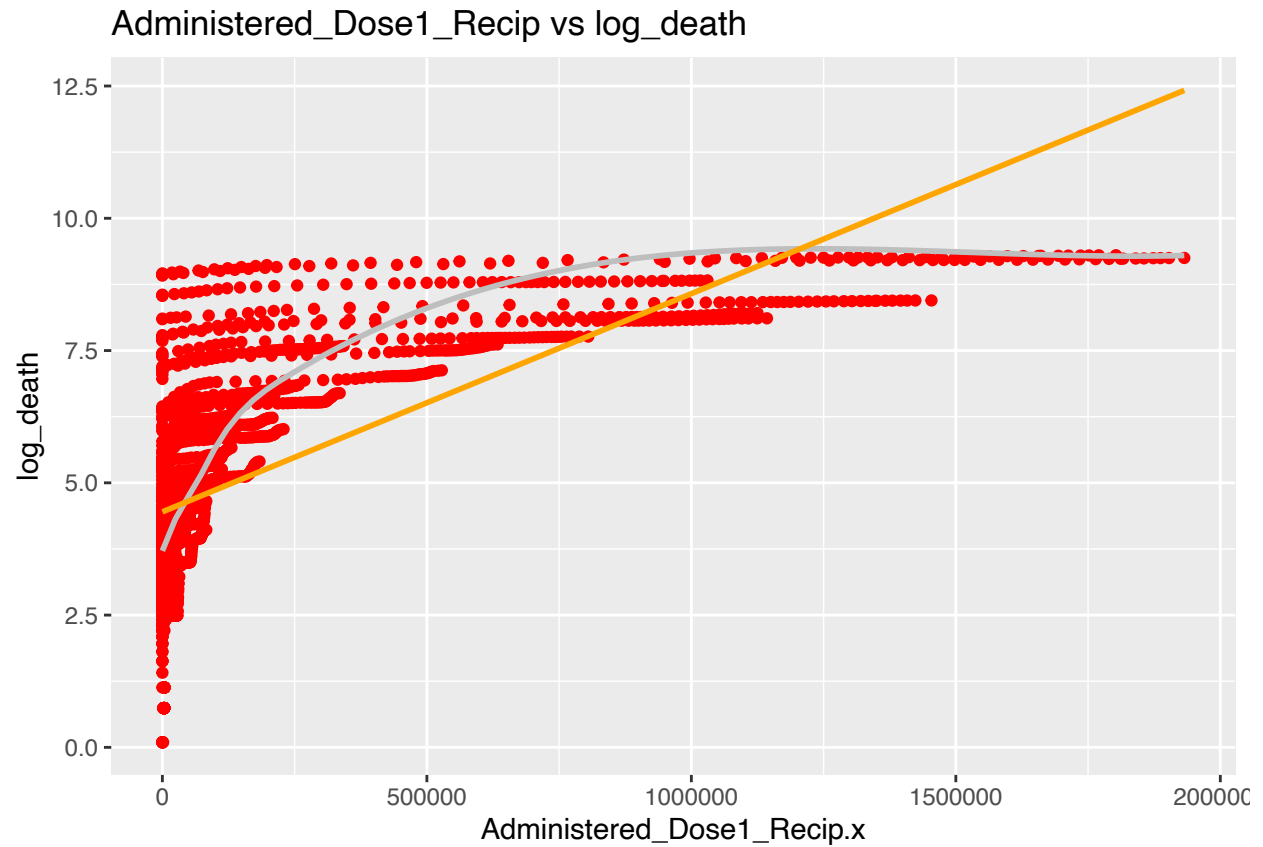
Administered_Dose1_Recip vs prev_log_new_death


```
ggplot(data=updatedjoin, mapping=aes(x=Administered_Dose1_Recip.x,y=log_death)) + geom_point(color="red
```
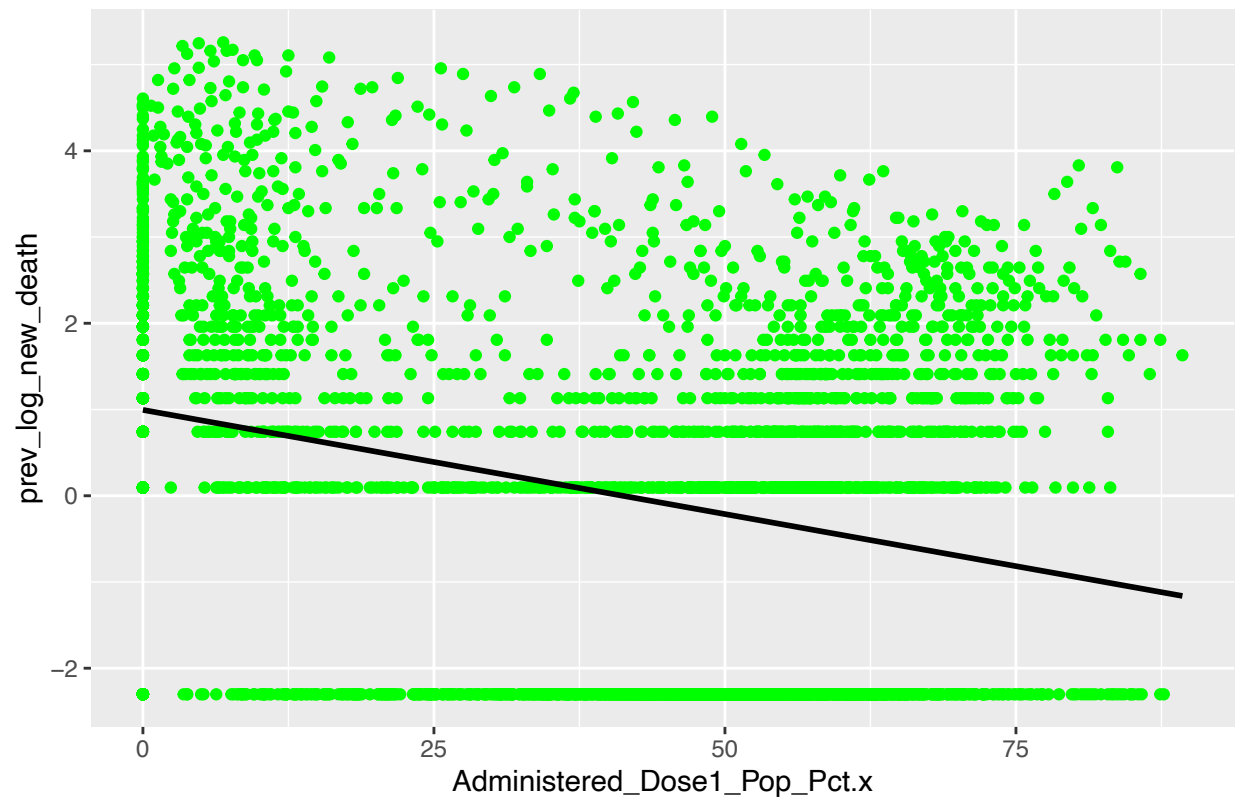
```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```

Administered_Dose1_Recip vs log_death


```
ggplot(data=updatedjoin, mapping=aes(x=Administered_Dose1_Pop_Pct.x,y=prev_log_new_death)) + geom_point
```

```
## 'geom_smooth()' using formula 'y ~ x'
```
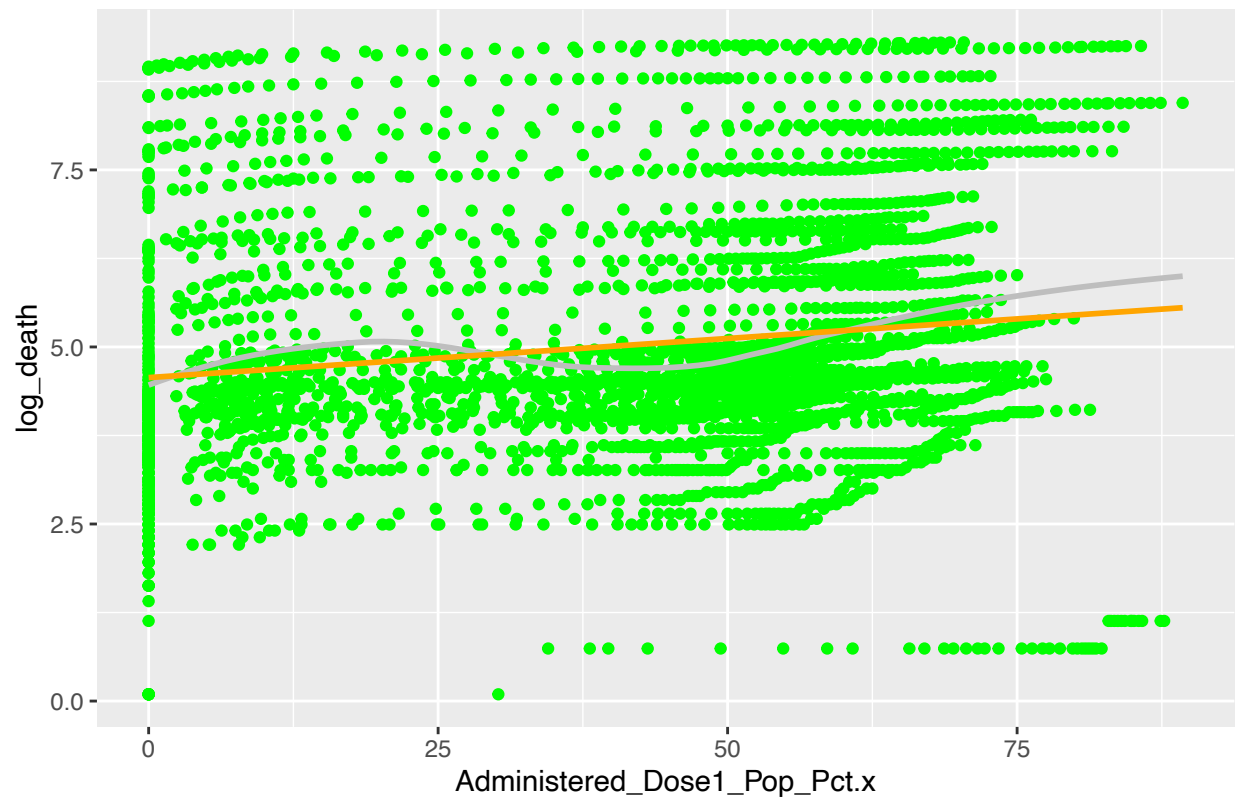
Administered_Dose1_Recip vs prev_log_new_death


```
ggplot(data=updatedjoin, mapping=aes(x=Administered_Dose1_Pop_Pct.x,y=log_death)) + geom_point(color="g
```

```
## `geom_smooth()` using formula 'y ~ x'
## `geom_smooth()` using formula 'y ~ x'
```
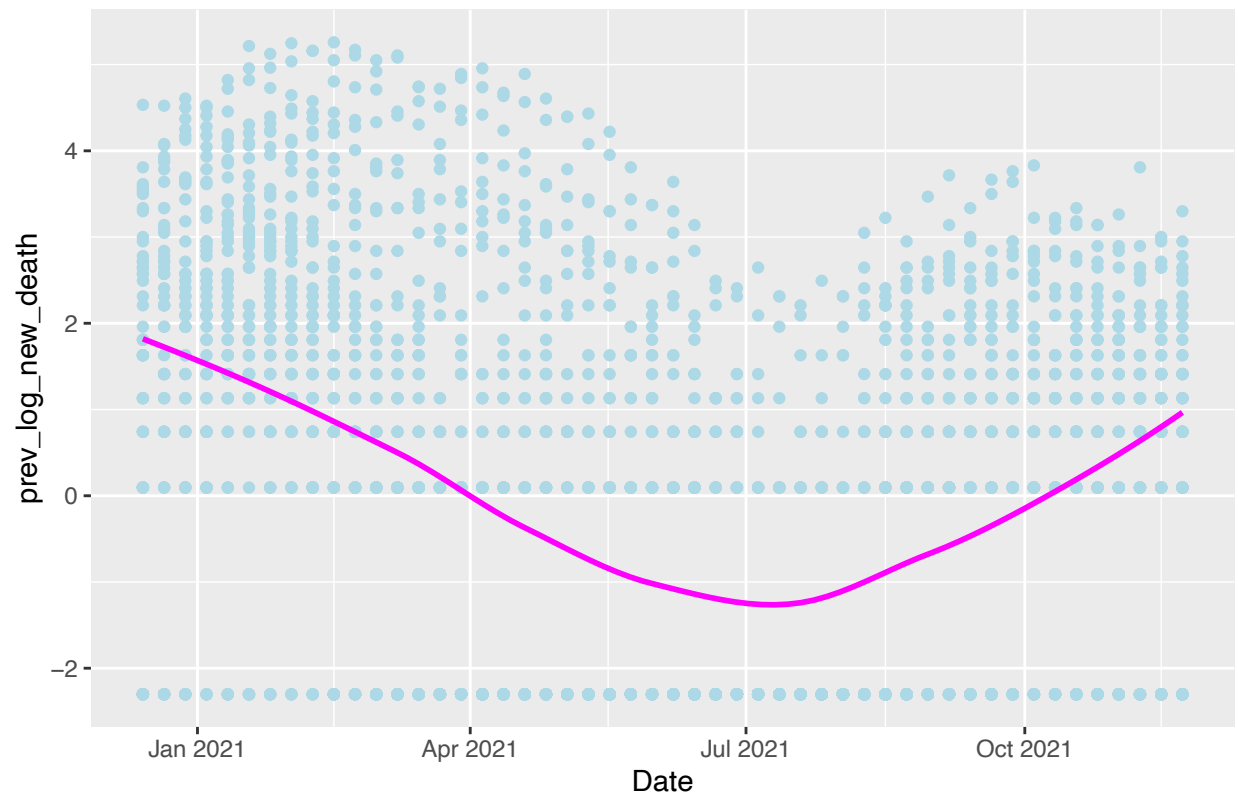
Administered_Dose1_Recip vs log_death


```r
#date
ggplot(data=updatedjoin, mapping=aes(x=Date,y=prev_log_new_death)) + geom_point(color="Lightblue")+geom_
```

```
## 'geom_smooth()' using formula 'y ~ x'
```
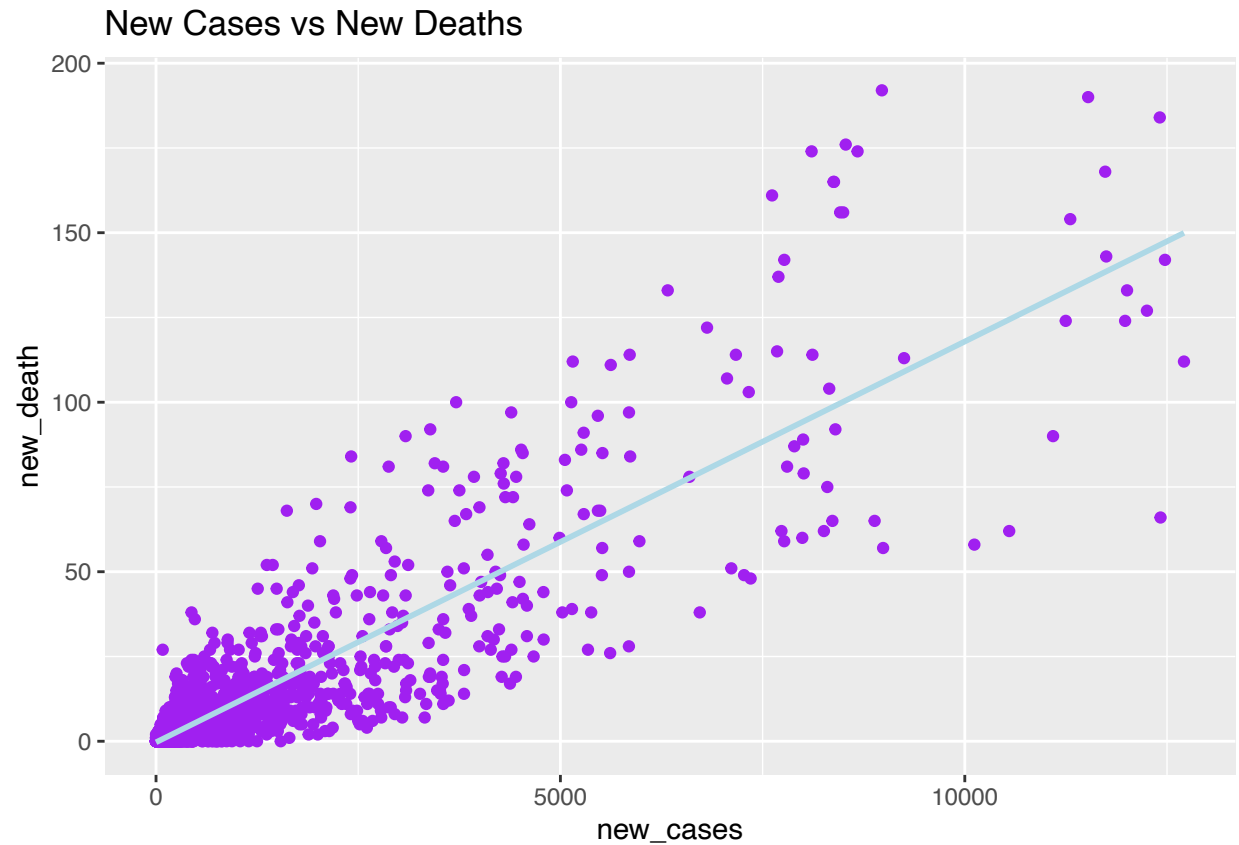
Date vs prev_log_new_death

```
#death vs cases
ggplot(data=updatedjoin, mapping=aes(x=new_cases,y=new_death)) + geom_point(color="purple")+geom_smooth
```

```
## 'geom_smooth()' using formula 'y ~ x'
```

New Cases vs New Deaths


```
#which( colnames(updatedjoin)=="Date" )

#include in glm (from updatedjoin3)
#Date
#Recip_County
#Administered_Dose1_Recip.x
#Administered_Dose1_Pop_Pct.x
#Series_Complete_Yes
#Series_Complete_Pop_Pct
#SVI_CTGY #maybe
#Metro_status
#death
#new_cases_per_100k.x
#new_vax_this_week

#updatedjoin4<-updatedjoin3 %>% select(Date,
#Recip_County,
#Administered_Dose1_Recip.x,
#Administered_Dose1_Pop_Pct.x,
#Series_Complete_Yes,
#Series_Complete_Pop_Pct,
#SVI_CTGY,
#Metro_status,
#death,
#new_cases_per_100k.x,
#new_vax_this_week,prev_log_new_death)
```

```
#linear model:
#which( colnames(updatedjoin)=="FIPS" )

#updatedjoin1<-updatedjoin %>% relocate(75, .after = last_col())

#updatedjoin2<-read.csv("updatedjoineddata.csv", header=T,stringsAsFactors = TRUE)
#updatedjoin3<-updatedjoin2[ , colSums(is.na(updatedjoin2)) < nrow(updatedjoin2)]



#covid.lm1 <- bestglm(updatedjoin4, IC="BIC", method="exhaustive", TopModels=)
#covid.lm<-covid.lm$BestModel
#vif(covid.lm)
```