Alec Arroyo Final Project

```
#Alec Arroyo
#Final Project
library("sqldf")
```

```
## Loading required package: gsubfn
```

```
## Loading required package: proto
```

```
## Loading required package: RSQLite
```

```
library("ggplot2")
library("openintro")
```

```
## Loading required package: airports
```

```
## Loading required package: cherryblossom
```

```
## Loading required package: usdata
```

```
library("tidyverse")
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v tibble  3.1.2     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
## v purrr   0.3.4
```

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

```
library("kernlab")
```

```
##
## Attaching package: 'kernlab'
```

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

```
## The following object is masked from 'package:ggplot2':
##
##     alpha
```

```
library("randomForest")
```

```
## randomForest 4.6-14
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

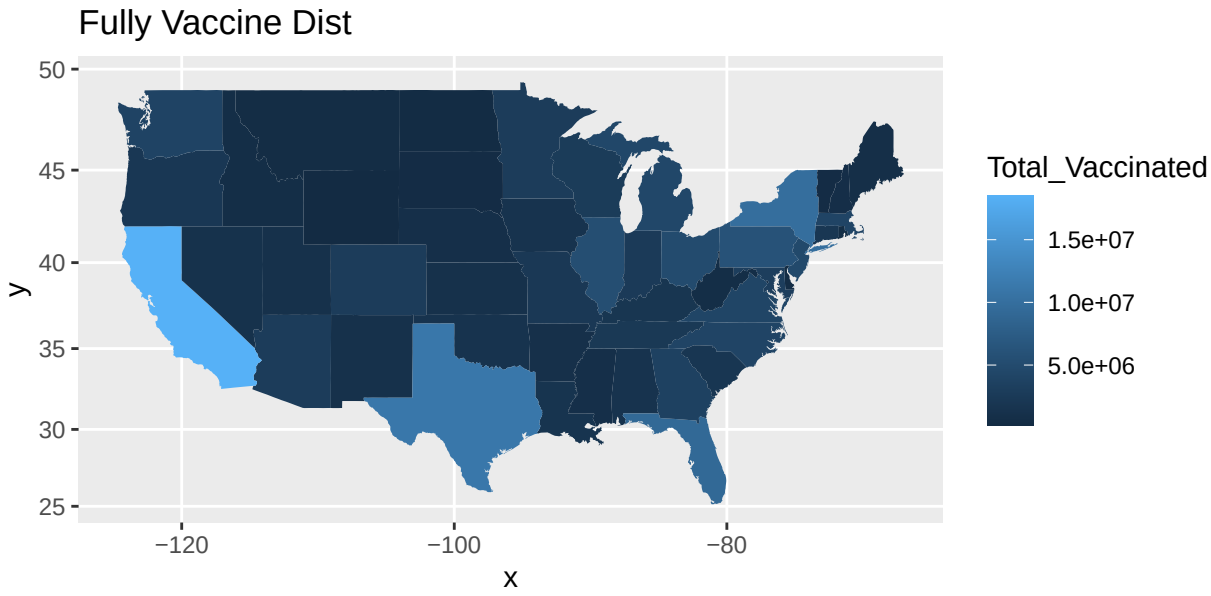
```
library("readxl")


url <- "https://data.cdc.gov/api/views/unsk-b7fc/rows.csv?accessType=DOWNLOAD"
vaccinedataset <- read.csv(url)
#seriescomplete_yes means people total vacced

#Convert Date field to type Date
vaccinedataset$Date <- as.Date(vaccinedataset$Date, "%m/%d/%Y")
#Grab most recent date and store into new dataframe
VaccineCurrentDate <- vaccinedataset[vaccinedataset$Date==max(vaccinedataset$Date),]

#Reorder by location
VaccineCurrentDate <- VaccineCurrentDate[order(VaccineCurrentDate$Location)]

#Get state names based on state abbreviations
stnames <- abbr2state(VaccineCurrentDate$Location)

#Add in stnames field to dataframe
VaccineCurrentDate <- data.frame(VaccineCurrentDate, stnames)

#Want to look for NA states
sqldf('select Location, stnames from VaccineCurrentDate')
```

```
##    Location              stnames
## 1        IA                 Iowa
## 2        MS          Mississippi
## 3        SC       South Carolina
## 4        DC District of Columbia
## 5        CA           California
## 6        TN            Tennessee
## 7        AZ              Arizona
## 8        UT                 Utah
## 9        FM                 <NA>
## 10       AK               Alaska
## 11      IH2                 <NA>
## 12       OK             Oklahoma
## 13      DD2                 <NA>
## 14       SD         South Dakota
## 15       VI                 <NA>
## 16       MN            Minnesota
## 17       CT          Connecticut
## 18       NC       North Carolina
## 19       LA            Louisiana
## 20       ME                Maine
## 21       PA         Pennsylvania
## 22       NE             Nebraska
```

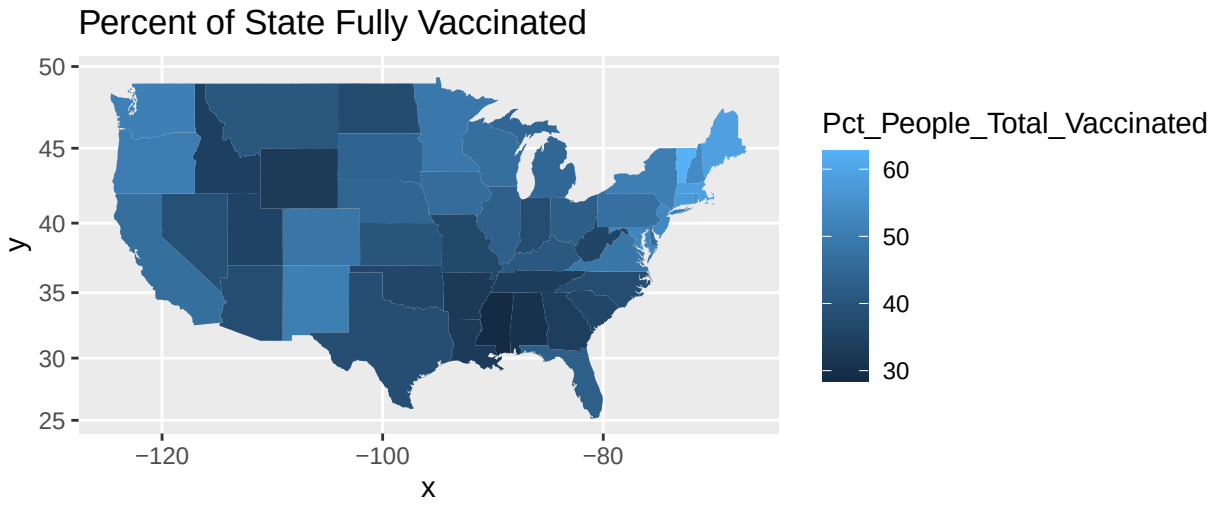
```
## 23       WY              Wyoming
## 24       FL              Florida
## 25       AR             Arkansas
## 26       ID                Idaho
## 27       RI         Rhode Island
## 28       DE             Delaware
## 29       MA        Massachusetts
## 30       KY              Kentucky
## 31      LTC                 <NA>
## 32       MI             Michigan
## 33       HI               Hawaii
## 34       NY             New York
## 35       WA           Washington
## 36       GU                 <NA>
## 37       KS               Kansas
## 38       MH                 <NA>
## 39       MO             Missouri
## 40       VT              Vermont
## 41       OH                 Ohio
## 42       CO             Colorado
## 43       RP                 <NA>
## 44       GA              Georgia
## 45       MP                 <NA>
## 46      BP2                 <NA>
## 47       ND         North Dakota
## 48       AL              Alabama
## 49       OR               Oregon
## 50       IN              Indiana
## 51       TX                Texas
## 52       VA             Virginia
## 53       IL             Illinois
## 54      VA2                 <NA>
## 55       WV        West Virginia
## 56       US                 <NA>
## 57       AS                 <NA>
## 58       MT              Montana
## 59       NH        New Hampshire
## 60       MD             Maryland
## 61       NM           New Mexico
## 62       NV               Nevada
## 63       PR                 <NA>
## 64       NJ           New Jersey
## 65       WI            Wisconsin
```

```
#Query to get rid of NAs for dataframe
VaccineCurrentDate <- sqldf('select * from VaccineCurrentDate where stnames != "NA"')

#Change name of columns
names(VaccineCurrentDate)[names(VaccineCurrentDate)=="Series_Complete_Yes"] <- "Total_Vaccinated"
names(VaccineCurrentDate)[names(VaccineCurrentDate)=="Series_Complete_Pop_Pct"] <- "Pct_People_Total_Va

#Make state names lowercase
VaccineCurrentDate$stnames <- tolower(VaccineCurrentDate$stnames)
```

```
us <- map_data("state")

newmap <- ggplot(VaccineCurrentDate, aes(map_id=stnames), inherit.aes = FALSE)
newmap <- newmap + geom_map(map=us, aes(fill=Total_Vaccinated))
newmap <- newmap + expand_limits(x=us$long, y=us$lat)
newmap <- newmap + coord_map() + ggtitle("Fully Vaccine Dist")
newmap
```




```
#Brighter the color of state, the more a state is vaccinated
newmappercent <- ggplot(VaccineCurrentDate, aes(map_id=stnames), inherit.aes = FALSE)
newmappercent <- newmappercent + geom_map(map=us, aes(fill=Pct_People_Total_Vaccinated))
newmappercent <- newmappercent + expand_limits(x=us$long, y=us$lat)
newmappercent <- newmappercent + coord_map() + ggtitle("Percent of State Fully Vaccinated")
newmappercent
```

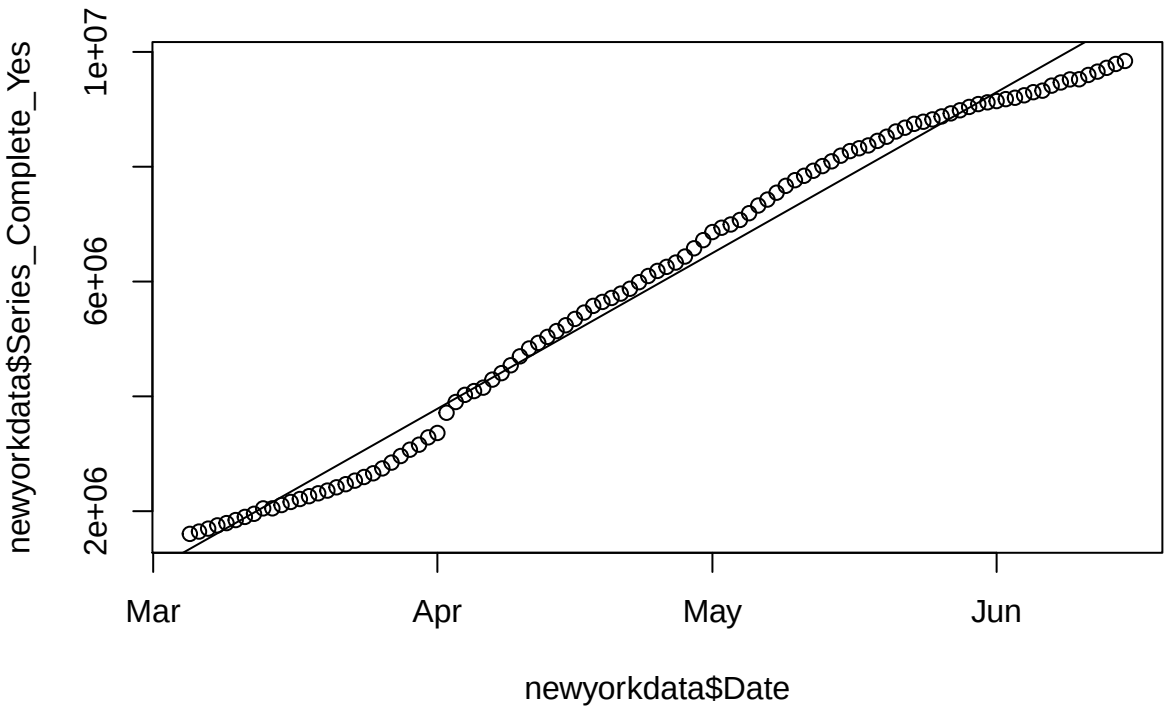



```
################RUNNING LINEAR MODELS
########Checking out new york data and make prediction of how many total vaccines in 2022

newyorkdata <- sqldf('select * from vaccinedataset where location="NY"')
```

```r
newyorkdata <- newyorkdata[newyorkdata$Date >= "2021-03-05",]
#Plot graph
plot(newyorkdata$Date, newyorkdata$Series_Complete_Yes)
lmyork <- lm(formula=Series_Complete_Yes~Date, newyorkdata)
#Summary says we're 91% accurate
summary(lmyork)
```

```
##
## Call:
## lm(formula = Series_Complete_Yes ~ Date, data = newyorkdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -721876 -287693   87836  258694  455960
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.688e+09  2.054e+07  -82.22   <2e-16 ***
## Date         9.040e+04  1.096e+03   82.51   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 330600 on 101 degrees of freedom
## Multiple R-squared:  0.9854, Adjusted R-squared:  0.9852
## F-statistic:  6807 on 1 and 101 DF,  p-value: < 2.2e-16
```

```r
#Draw line of best fit
abline(lmyork)
```




```r
#Predict how many people will be totally vaccinated by 2022 based on linear model
predict(lmyork, data.frame(Date=as.Date("2021-06-09")))
```

```
##        1
## 10021634
```

```
#testing age as a factor in total vaccinations
lmyorkold <- lm(formula=Series_Complete_Yes~Series_Complete_65Plus+Series_Complete_18Plus+Series_Comple
summary(lmyorkold)
```

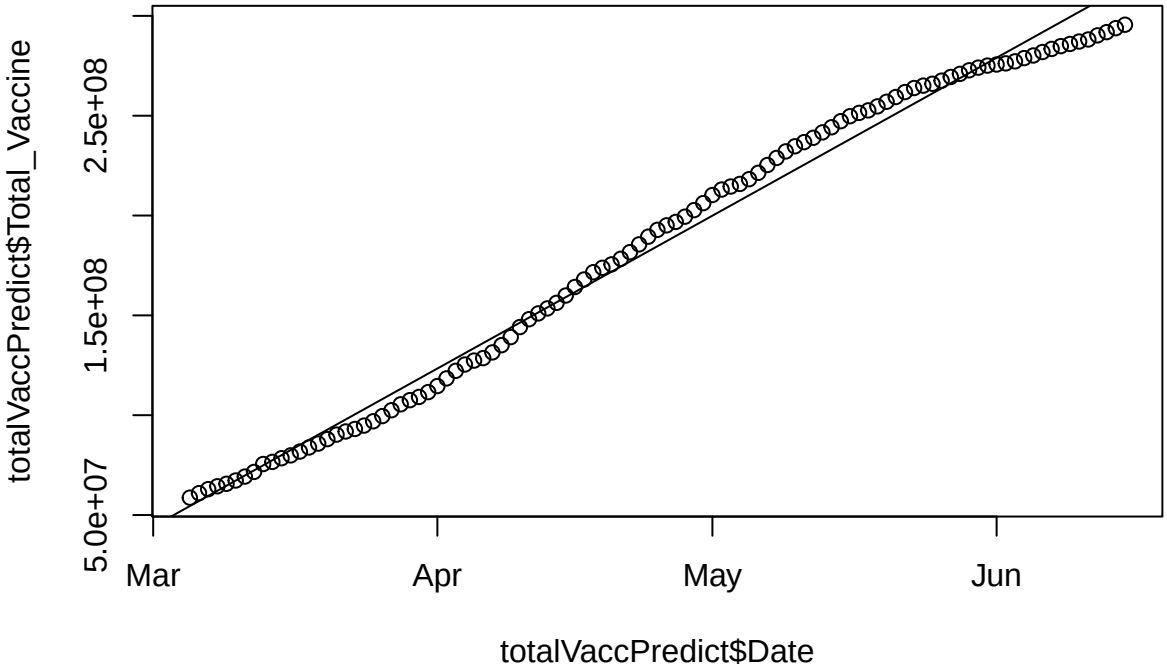
```
##
## Call:
## lm(formula = Series_Complete_Yes ~ Series_Complete_65Plus + Series_Complete_18Plus +
##     Series_Complete_12Plus, data = newyorkdata)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -41083 -15128  -5449   1116 191901
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)              1.901e+04  2.065e+04   0.920  0.35956
## Series_Complete_65Plus -8.733e-02  3.874e-02  -2.254  0.02639 *
## Series_Complete_18Plus  1.032e+00  1.064e-02  97.036  < 2e-16 ***
## Series_Complete_12Plus  6.798e-03  2.136e-03   3.183  0.00195 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 37830 on 99 degrees of freedom
## Multiple R-squared:  0.9998, Adjusted R-squared:  0.9998
## F-statistic: 1.759e+05 on 3 and 99 DF,  p-value: < 2.2e-16
```

```
#Do same thing for all of US vaccination totals##########

totalVaccPredict <- sqldf('select distinct SUM(Series_Complete_Yes) as Total_Vaccine, Date from vaccine
totalVaccPredict <- totalVaccPredict[!totalVaccPredict$Total_Vaccine == 0,]

plot(totalVaccPredict$Date, totalVaccPredict$Total_Vaccine)
lmUS <- lm(formula=Total_Vaccine~Date, totalVaccPredict)
#Summary says we're 99% accurate
summary(lmUS)
```

```
##
## Call:
## lm(formula = Total_Vaccine ~ Date, data = totalVaccPredict)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -19193407  -6590093    153923   7845612  11790803
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.764e+10  5.104e+08  -93.35   <2e-16 ***
## Date         2.552e+06  2.723e+04   93.71   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8217000 on 101 degrees of freedom
## Multiple R-squared:  0.9886, Adjusted R-squared:  0.9885
```

```
## F-statistic:  8782 on 1 and 101 DF,  p-value: < 2.2e-16
```

```
#Draw line of best fit
abline(lmUS)
```



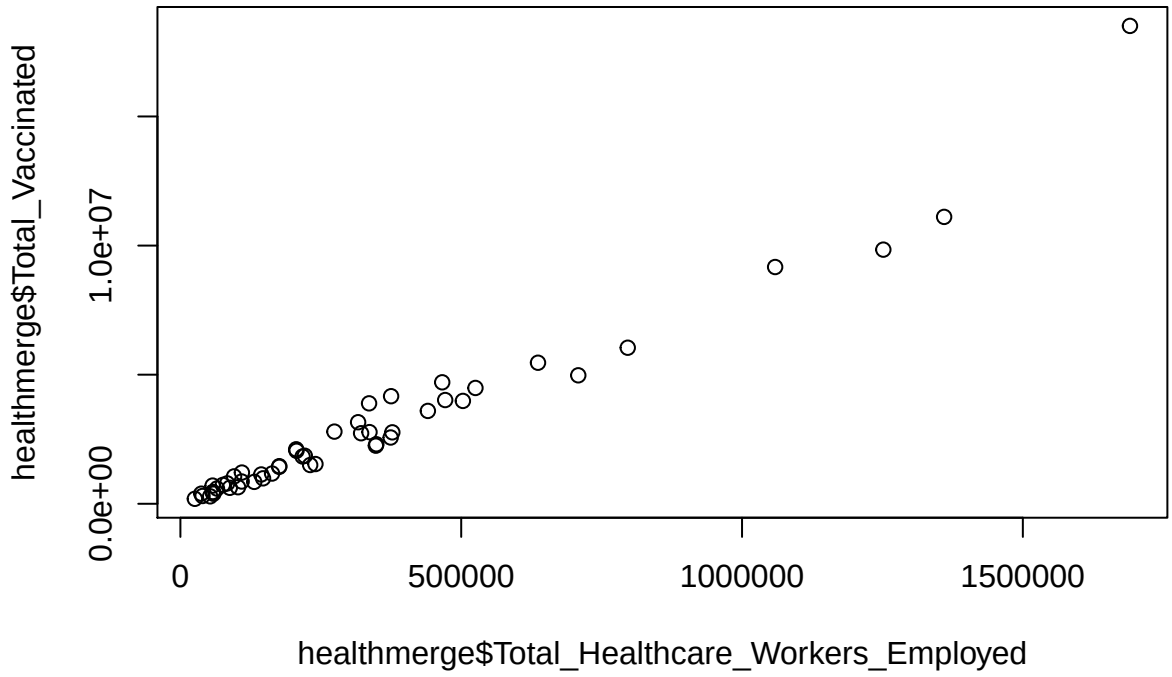

```
#Predict how many people will be totally vaccinated by 2022 based on linear model
predict(lmUS, data.frame(Date=as.Date("2021-07-9")))
```

```
##         1
## 376003949
```

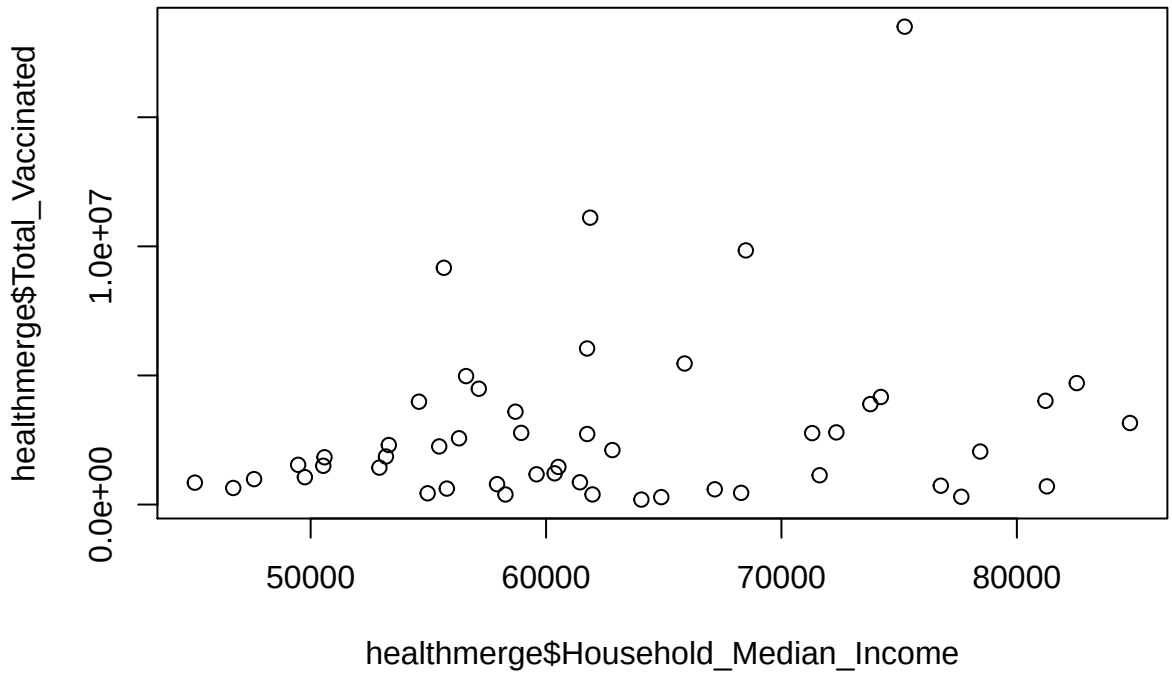
```
#########################USE THIS
url4 <- "/Users/alec_arroyo/Documents/Sryacuse Data Science Courses/Introduction to Data Science/raw_da
healthwork <- read.csv(url4)

healthwork$Location <- tolower(healthwork$Location)
healthwork <- healthwork[,c(-3)]
healthmerge <- merge(VaccineCurrentDate, healthwork, by.x="stnames",by.y="Location")


url5 <- "/Users/alec_arroyo/Downloads/csvData.csv"
medianmoney <- read.csv(url5)
medianmoney$State <- tolower(medianmoney$State)

healthmerge <- merge(healthmerge, medianmoney, by.x="stnames",by.y="State")


#url6 <- "blob:https://worldpopulationreview.com/dd2b812f-fb5a-42ee-82d8-b3627be298f1"
url6 <- "/Users/alec_arroyo/Downloads/dd2b812f-fb5a-42ee-82d8-b3627be298f1.csv"
crimepop <- read.csv(url6)
crimepop$State <- tolower(crimepop$State)

healthmerge <- merge(healthmerge, crimepop, by.x="stnames",by.y="State")
```

```
url7 <- "/Users/alec_arroyo/Documents/Sryacuse Data Science Courses/Introduction to Data Science/Flu Va
flu <- read_excel(url7)
flu$Location <- tolower(flu$Location)

healthmerge <- merge(healthmerge, flu, by.x="stnames",by.y="Location")

names(healthmerge)[names(healthmerge)=="Total.Health.Care.Employment"] <- "Total_Healthcare_Workers_Empl
names(healthmerge)[names(healthmerge)=="HouseholdIncome"] <- "Household_Median_Income"
names(healthmerge)[names(healthmerge)=="homicideRate2017"] <- "Crime_Rate"
```

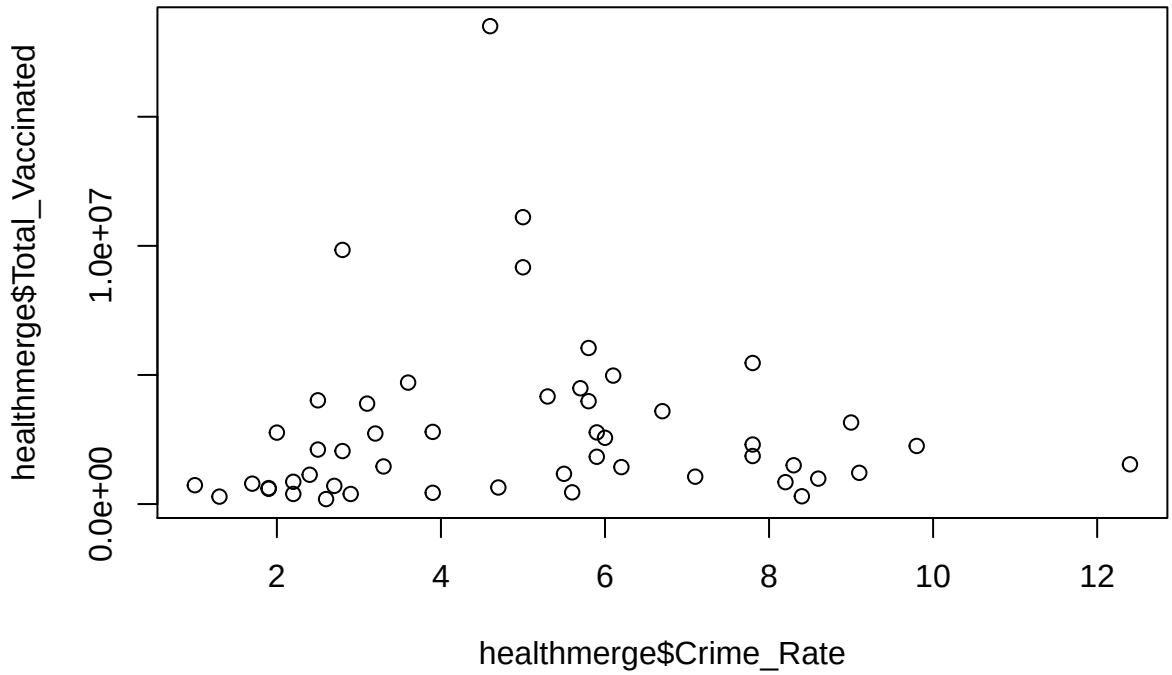
```
plot(healthmerge$Total_Healthcare_Workers_Employed, healthmerge$Total_Vaccinated)
```



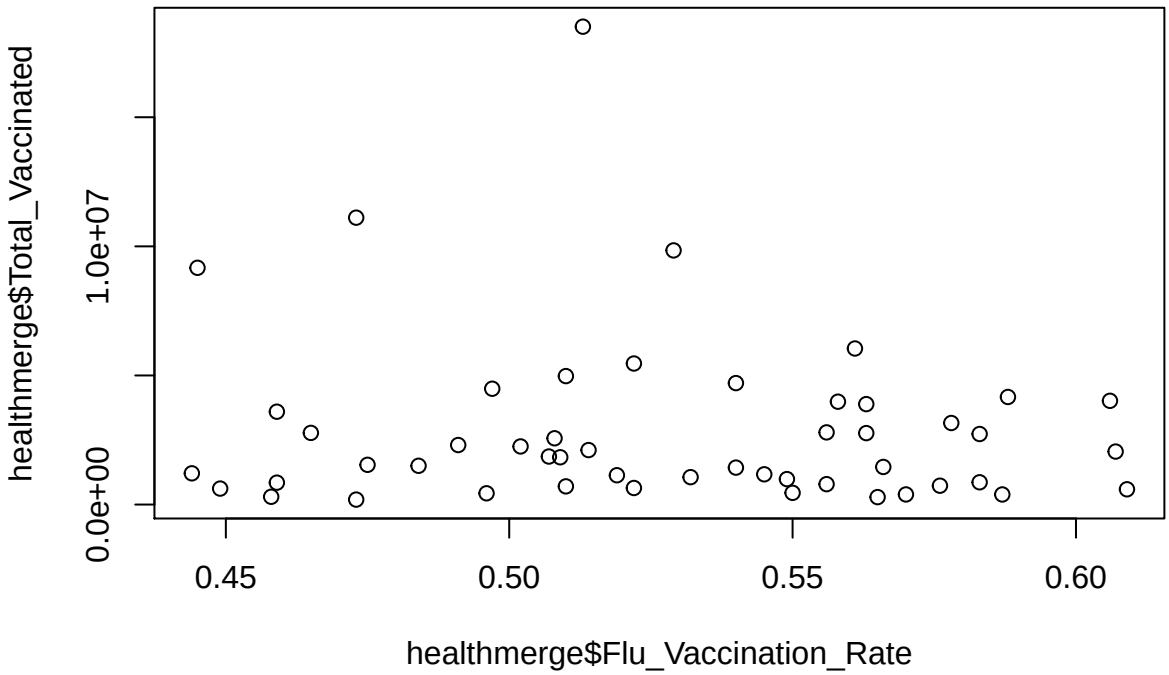

```
plot(healthmerge$Household_Median_Income, healthmerge$Total_Vaccinated)
```




```
plot(healthmerge$Crime_Rate, healthmerge$Total_Vaccinated)
```




```
plot(healthmerge$Flu_Vaccination_Rate, healthmerge$Total_Vaccinated)
```




```
lmUSDate <- lm(formula=Total_Vaccinated~Total_Healthcare_Workers_Employed+Household_Median_Income+Crime_
#Summary says we're 99% accurate
summary(lmUSDate)
```

```
##
## Call:
## lm(formula = Total_Vaccinated ~ Total_Healthcare_Workers_Employed +
##     Household_Median_Income + Crime_Rate + Flu_Vaccination_Rate,
##     data = healthmerge)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1481110  -187990    -5099   253661  3046604
##
## Coefficients:
##                                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)                      -1.664e+06  1.374e+06  -1.211   0.2321
## Total_Healthcare_Workers_Employed  9.035e+00  2.717e-01  33.256   <2e-16 ***
## Household_Median_Income            2.674e+01  1.061e+01   2.522   0.0153 *
## Crime_Rate                        -8.241e+03  4.171e+04  -0.198   0.8443
## Flu_Vaccination_Rate              -2.439e+05  2.436e+06  -0.100   0.9207
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 652000 on 45 degrees of freedom
## Multiple R-squared:  0.9647, Adjusted R-squared:  0.9616
## F-statistic: 307.9 on 4 and 45 DF,  p-value: < 2.2e-16
```

```
#Create field for category whether total vacc is low or high (43.034% higher or lower derived from mean
healthmerge$goodbad <- ifelse(healthmerge$Pct_People_Total_Vaccinated>43.034 ,"High", "Low")
healthmerge$goodbad <- as.character(healthmerge$goodbad)
healthmerge$goodbad <- as.factor(healthmerge$goodbad)
```

```
#################################################################TESTING

#Dataset with varaibles we want to test with
randomVariables <- data.frame(healthmerge$Total_Vaccinated, healthmerge$Pct_People_Total_Vaccinated, hea

#Rename columns
names(randomVariables)[names(randomVariables)=="healthmerge.Total_Vaccinated"] <- "Total_Vaccinated"
names(randomVariables)[names(randomVariables)=="healthmerge.Pct_People_Total_Vaccinated"] <- "Pct_People
names(randomVariables)[names(randomVariables)=="healthmerge.Total_Healthcare_Workers_Employed"] <- "Tota
names(randomVariables)[names(randomVariables)=="healthmerge.Household_Median_Income"] <- "Household_Medi
names(randomVariables)[names(randomVariables)=="healthmerge.Crime_Rate"] <- "Crime_Rate"
names(randomVariables)[names(randomVariables)=="healthmerge.Flu_Vaccination_Rate"] <- "Flu_Vaccination_R
names(randomVariables)[names(randomVariables)=="healthmerge.Series_Complete_12Plus"] <- "Series_Complete
names(randomVariables)[names(randomVariables)=="healthmerge.Series_Complete_18Plus"] <- "Series_Complete
names(randomVariables)[names(randomVariables)=="healthmerge.Series_Complete_65Plus"] <- "Series_Complete
names(randomVariables)[names(randomVariables)=="healthmerge.goodbad"] <- "HighLowRating"

#RandomForest Algorithm
randrftotvacc <- randomForest(x=randomVariables[,-10], y=randomVariables[,10])
randrftotvacc
```

```
##
## Call:
##  randomForest(x = randomVariables[, -10], y = randomVariables[,      10])
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 2%
## Confusion matrix:
##      High Low class.error
## High   24   1        0.04
## Low     0  25        0.00
```

```
importance(randrftotvacc)
```

```
##                                  MeanDecreaseGini
## Total_Vaccinated                        0.7433569
## Pct_People_Total_Vaccinated            13.0013608
## Total_Healthcare_Workers_Employed       0.4996011
## Household_Median_Income                 3.0656581
## Crime_Rate                              1.8428302
## Flu_Vaccination_Rate                    3.6233579
## Series_Complete_12Plus                  0.6064925
## Series_Complete_18Plus                  0.6197739
## Series_Complete_65Plus                  0.5081287
```

```
#testing
prediction <- predict(randrftotvacc, randomVariables[1,-10])
prediction
```

```
##   1
## Low
## Levels: High Low
```

```r
#Proof
randomVariables[1,]
```

```
##   Total_Vaccinated Pct_People_Total_Vaccinated
## 1          1501171                        30.6
##   Total_Healthcare_Workers_Employed Household_Median_Income Crime_Rate
## 1                            231070                   50536        8.3
##   Flu_Vaccination_Rate Series_Complete_12Plus Series_Complete_18Plus
## 1                0.484                1501142                1486422
##   Series_Complete_65Plus HighLowRating
## 1                 565427           Low
```