Are mask sentiments of tweets related to vaccination rates of states in the US?

Henok Adbaru

12/15/2021

Introduction

The Covid-19 pandemic is a global pandemic that has affected the livelihood of most people in the United States. Mask mandates were implemented early on to help combat the spread of the virus. However, the response and attitude to mask use in the US varies greatly from person to person. Some states also choose to exercise less intensive mask guidelines than others, creating a very disjointed response to the pandemic. With the Covid-19 vaccines being available and accessible for over half a year at the time of this project, we also notice the vaccination rates varying greatly from state to state in the US. This project aims to examine if there is a connection with the public sentiment of mask use and the vaccination rates of US states.

The analysis will be done using two approaches. One analysis will be using the raw Vader scores for each tweet. A second analysis will be using a weighted Vader score based on the number of favorites each Tweet receives. The rational for the second approach is to try to take passive Twitter users into account in the analysis, as a Twitter user would likely favorite a tweet that matches their own sentient regarding mask use. The vaccination rates by states are acquired from CDC's public use files.

All relevant files can be found in my GitHub directory here:
https://github.com/henoka94/SURV-727-Term-Paper

Note: this analysis only looks at vaccination rates and sentiment analysis of mainland USA. Hawaii, Alaska, and other island territories are excluded.

First, relevant R libraries and data files are read in.

```
#Read in necessary R libraries and assign Google key used in geocoding

library(rtweet)
library(readr)
library(SentimentAnalysis)
library(usmap)
library(maps)
library(rgdal)
library(ggmap)
library(tidyverse)
library(DBI)
```

```r
library(bigrquery)
library(dbplyr)
library(qdap)
library(stringr)
library(quanteda)
library(vader)
library(GGally)
library(sf)
library(spData)
library(tidygeocoder)
library(stringr)
library(sf)
library(revgeo)
library(ggsn)
library(scales)

#The following 3 lines of codes will need to be changed to match the download
location of the input files when code is ran.
load("C:\\Users\\Henny\\Documents\\mask_tweets.RData")
load("C:\\Users\\Henny\\Documents\\tweets_state.RData")
us_vax <- read.csv("C:\\Users\\Henny\\Documents\\HW\\SURV 727 -
Data\\covid19_vaccinations_in_the_united_states.csv")
```

Section 1: Processing Tweets and assigning Vader scores

We start by searching Twitter for relevant tweets pertaining to mask use. We filter on tweets containing "mask", "masks", "#mask", or "#mask".

```r
# search tweets
mask_tweets <- search_tweets ("mask OR masks OR #mask OR #mask", n = 1000000,
retryonratelimit = TRUE)
save(mask_tweets,file="mask_tweets.Rdata")
```

One of the downsides to big data analysis is that it's not feasible to check every single match on whether the Tweet captured in the search truly pertains to Covid related mask use. However, we do get a significant number of hits (n = 1,011,186)

Next, we need to drop any Tweets that appear outside of the US:

```r
mask_tweets_coord <-lat_lng(mask_tweets)

tweets <- mask_tweets_coord %>%
  drop_na(lat) %>%
  drop_na(lng) %>%
  filter(lat > 24.39631 & lat < 49.38436 & lng > -124.84897 & lng < -
66.88544)
```

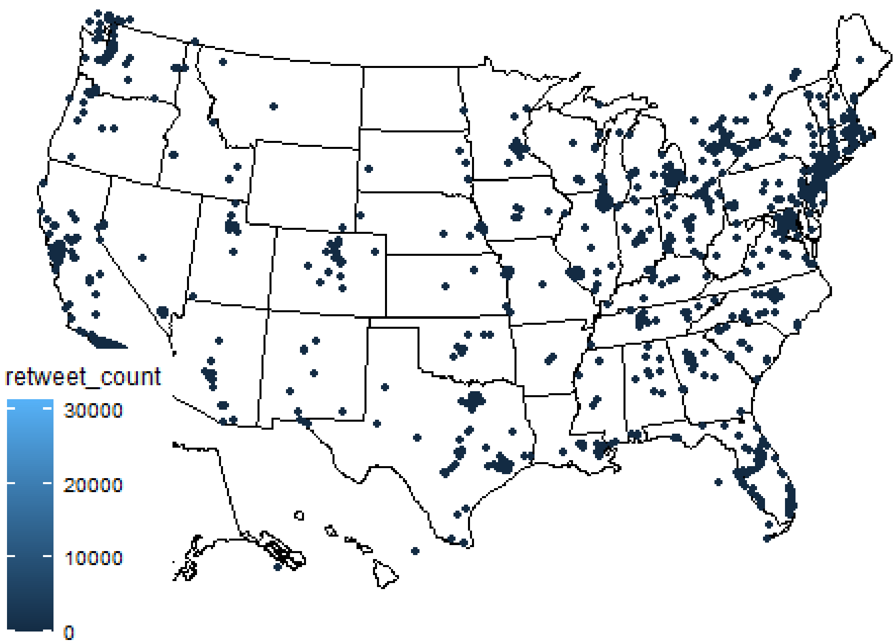
Unfortunately the vast majority of the tweets are either outside mainland USA, or simply does not include geocoding in their metadata. The resulting number of usable tweets for analysis are now n = 2,896.

The Tweets will now be plotted on a US map, to see the geographic distribution of Tweets collected. Some additional processing steps will be made to remove Tweets that are found otuside of mainland USA.

```
#map <- get_stamenmap(bbox = c(-124.8489,24.39631,-66.88544,49.38436 ), zoom
= 10, maptype = "toner-hybrid")
map <- plot_usmap("states")
map
```

```
coord <- tweets %>%
  select(lng,lat)

coord$ID <- coord$lat*coord$lng
tweets_transformed <- usmap_transform(coord)

## Warning in showSRID(uprojargs, format = "PROJ", multiline = "NO",
prefer_proj =
## prefer_proj): Discarded datum unknown in Proj4 definition

coord_join <- tweets_transformed %>%
  select(lng.1,lat.1,ID)

coord <- coord %>%
```

```
  left_join(coord_join, by = c("ID" = "ID"))


tweets <- tweets %>%
  add_column(lng1 = coord$lng.1) %>%
  add_column(lat1 = coord$lat.1)

tweets <-rowid_to_column(tweets)

map + geom_point(data = tweets, aes(x = lng1, y = lat1, color =
retweet_count), size = 1 )
```




```
#NE outliers
remove1 <- tweets %>%
  filter(lat > 41.7 & lng > -82.09 & lng < -79.29)

remove2 <- tweets %>%
  filter(lat > 43.4 & lng > -82.09 & lng < -75.4)

remove3 <- tweets %>%
  filter(lat > 45 & lng > -78 & lng < -72.928)

#NW outliers
remove4 <- tweets %>%
  filter(lat > 48.49 & lng > -124.7 & lng < -123.2)
```

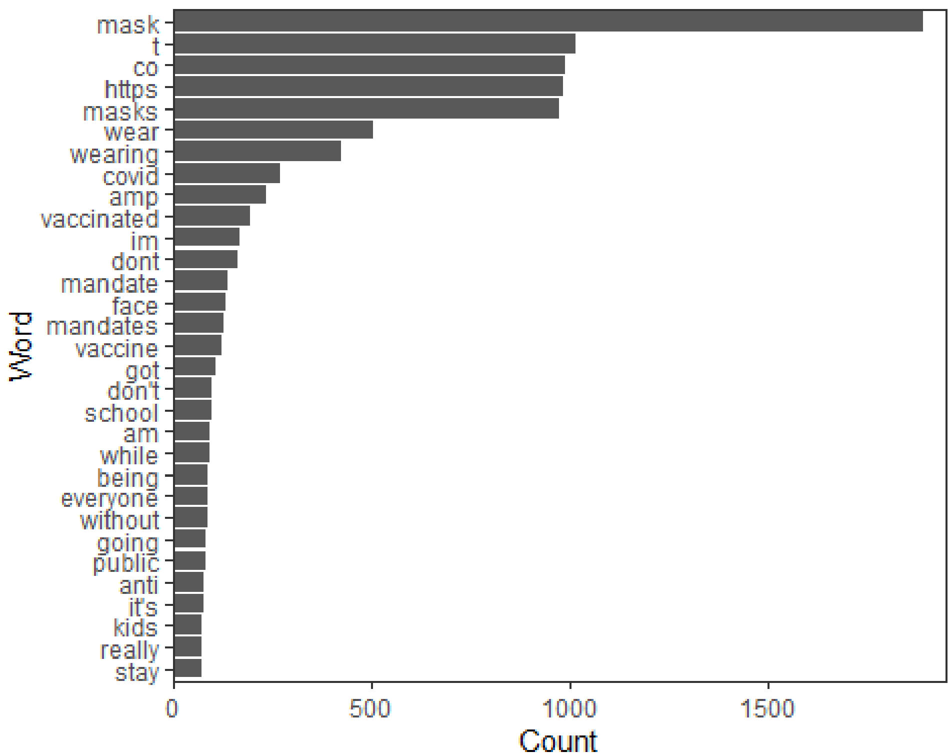
```
remove5 <- tweets %>%
  filter(lat > 48.279 & lng > -123.88 & lng < -123.2)

remove6 <- tweets %>%
  filter(lat > 49)

#South outliers
remove7 <- tweets %>%
  filter(lat < 25.3)

remove8 <- tweets %>%
  filter(lat < 26 & lng > -101 & lng < -99)

remove9 <- tweets %>%
  filter(lat < 29 & lng > -88 & lng < -82.8)

remove <-
rbind(remove1,remove2,remove3,remove4,remove5,remove6,remove7,remove8,remove9
)

map + geom_point(data = remove9, aes(x = lng1, y = lat1, color =
retweet_count), size = 1 )
```


```
#Make a "not in" function to remove the
'%!in%' <- function(x,y)!('%in%'(x,y))
```

```
tweets_usa <- tweets %>%
 filter(rowid %!in% remove$rowid)

map + geom_point(data = tweets_usa, aes(x = lng1, y = lat1, color =
retweet_count), size = 1 )
```

After the last step of Tweet processing, the final number of Tweets available for analysis is n = 2,641. Next, we'll do a plot of most common words found in the collected tweets (not including common stop words).

```
frequent_terms <- freq_terms(tweets_usa["text"], 30)
bagtweets <- tweets_usa$text %>% iconv("latin1", "ASCII", sub="") %>%
scrubber() %sw% qdapDictionaries::Top200Words
frequent_terms <- freq_terms(bagtweets, 30)
plot(frequent_terms)
```



Then, we move onto calculate the sentiment scores. We will use Vader scores for sentiment analysis for Twitter.

Section 2: Calculate Vader scores and create the weighted Vader scheme

```
data('DictionaryGI')
DictionaryGI$positive[1:100]
```

```
##   [1] "abide"            "ability"          "able"             "abound"
##   [5] "absolve"          "absorbent"        "absorption"       "abundance"
##   [9] "abundant"         "accede"           "accentuate"       "accept"
##  [13] "acceptable"       "acceptance"       "accessible"       "accession"
##  [17] "acclaim"          "acclamation"      "accolade"         "accommodate"
##  [21] "accommodation"    "accompaniment"    "accomplish"
"accomplishment"
##  [25] "accord"           "accordance"       "accountable"      "accrue"
##  [29] "accuracy"         "accurate"         "accurateness"     "achieve"
##  [33] "achievement"      "acknowledgement"  "acquaint"         "acquaintance"
##  [37] "acquit"           "acquittal"        "actual"           "actuality"
##  [41] "adamant"          "adaptability"     "adaptable"        "adaptation"
##  [45] "adaptive"         "adept"            "adeptness"        "adequate"
##  [49] "adherence"        "adherent"         "adhesion"         "adhesive"
##  [53] "adjunct"          "adjust"           "adjustable"       "adjustment"
##  [57] "admirable"        "admiration"       "admire"           "admirer"
##  [61] "admit"            "admittance"       "adorable"         "adore"
```

```
##  [65] "adorn"            "adornment"        "adroit"           "adroitly"
##  [69] "adulation"        "adult"            "advance"          "advancement"
##  [73] "advantage"        "advantageous"     "advent"
"adventuresome"
##  [77] "adventurous"      "advisable"        "advocacy"         "affability"
##  [81] "affable"          "affection"        "affectionate"     "affiliate"
##  [85] "affiliation"      "affinity"         "affirm"           "affirmation"
##  [89] "affirmative"      "affix"            "affluence"        "affluent"
##  [93] "afloat"           "aggregate"        "aggregation"      "agile"
##  [97] "agility"          "agreeable"        "agreement"        "aid"
```

```
DictionaryGI$negative[1:100]
```

```
##   [1] "abandon"         "abandonment"     "abate"           "abdicate"
##   [5] "abhor"           "abject"          "abnormal"        "abolish"
##   [9] "abominable"      "abrasive"        "abrupt"          "abscond"
##  [13] "absence"         "absent"          "absent-minded"   "absentee"
##  [17] "absurd"          "absurdity"       "abuse"           "abyss"
##  [21] "accident"        "accost"          "accursed"        "accusation"
##  [25] "accuse"          "ache"            "acrimonious"     "acrimony"
##  [29] "addict"          "addiction"       "admonish"        "admonition"
##  [33] "adulterate"      "adulteration"    "adultery"        "adversary"
##  [37] "adverse"         "adversity"       "affectation"     "afflict"
##  [41] "affliction"      "afraid"          "against"         "aggravate"
##  [45] "aggravation"     "aggression"      "aggressive"      "aggressiveness"
##  [49] "aggressor"       "aggrieve"        "aghast"          "agitate"
##  [53] "agitation"       "agitator"        "agonize"         "agony"
##  [57] "ail"             "ailment"         "aimless"         "alarm"
##  [61] "alarming"        "alas"            "alibi"           "alien"
##  [65] "alienate"        "alienation"      "allegation"      "allege"
##  [69] "aloof"           "altercation"     "ambiguity"       "ambiguous"
##  [73] "ambivalent"      "ambush"          "amiss"           "amputate"
##  [77] "anarchist"       "anarchy"         "anger"           "angry"
##  [81] "anguish"         "animosity"       "annihilate"      "annihilation"
##  [85] "annoy"           "annoyance"       "anomalous"       "anomaly"
##  [89] "antagonism"      "antagonist"      "antagonistic"    "antagonize"
##  [93] "anti-social"     "antipathy"       "antiquated"      "antitrust"
##  [97] "anxiety"         "anxious"         "anxiousness"     "apathetic"
```

```
data_dictionary_LSD2015$negative[1:50]
```

```
##  [1] "a lie"             "abandon*"          "abas*"             "abattoir*"
##  [5] "abdicat*"          "aberra*"           "abhor*"            "abject*"
##  [9] "abnormal*"         "abolish*"          "abominab*"         "abominat*"
## [13] "abrasiv*"          "absent*"           "abstrus*"          "absurd*"
## [17] "abus*"             "accident*"         "accost*"           "accursed*"
## [21] "accusation*"       "accuse*"           "accusing*"         "acerbi*"
## [25] "ache*"             "aching*"           "achy"              "acomia*"
## [29] "acrimon*"          "adactylism*"       "addict*"           "admonish*"
## [33] "admonition*"       "adulterat*"        "adultery*"         "advers*"
## [37] "affectation*"      "affected*"         "affected manner*"  "afflict*"
```

```
## [41] "affright*"        "affront*"         "afraid*"          "against"
## [45] "ageism"           "ageist"           "aggravat*"        "aggress*"
## [49] "aggressiv*"       "aggriev*"
```

```
data_dictionary_LSD2015$positive[1:50]
```

```
##  [1] "ability*"     "abound*"      "absolv*"      "absorbent*"
"absorption*"
##  [6] "abundanc*"    "abundant*"    "acced*"       "accentuat*"   "accept*"
## [11] "accessib*"    "acclaim*"     "acclamation*" "accolad*"
"accommodat*"
## [16] "accomplish*"  "accord"       "accordan*"    "accorded*"    "accords"
## [21] "accountab*"   "accru*"       "accuracy*"    "accurat*"
"accustom*"
## [26] "ace"          "aced"         "aces"         "achiev*"
"acquaintanc*"
## [31] "acquiesc*"    "active*"      "actual"       "actuali*"
"adaptab*"
## [36] "adaptive"     "adept*"       "adequat*"     "adhere*"
"admirab*"
## [41] "admiration*"  "admire*"      "admiring*"    "admit*"       "adopt*"
## [46] "adorable"     "adorably"     "adoration"    "adore*"       "adoring"
```

```
data_dictionary_LSD2015$neg_positive[1:50]
```

```
##  [1] "best not"         "better not"       "no damag*"        "no no"
##  [5] "not ability*"     "not able"         "not abound*"      "not
absolv*"
##  [9] "not absorbent*"   "not absorption*"  "not abundanc*"    "not
abundant*"
## [13] "not acced*"       "not accentuat*"   "not accept*"      "not
accessib*"
## [17] "not acclaim*"     "not acclamation*" "not accolad*"     "not
accommodat*"
## [21] "not accomplish*"  "not accord"       "not accordan*"    "not
accorded*"
## [25] "not accords"      "not accountab*"   "not accru*"       "not
accuracy*"
## [29] "not accurat*"     "not accustom*"    "not ace"          "not aced"
## [33] "not aces"         "not achiev*"      "not acquaintanc*" "not
acquiesc*"
## [37] "not active*"      "not actual"       "not actuali*"     "not
adaptab*"
## [41] "not adaptive"     "not adept*"       "not adequat*"     "not
adhere*"
## [45] "not admirab*"     "not admiration*"  "not admire*"      "not
admiring*"
## [49] "not admit*"       "not adopt*"
```

```
data_dictionary_LSD2015$neg_negative[1:50]
```

```
##  [1] "not a lie"              "not abandon*"           "not abas*"
##  [4] "not abattoir*"          "not abdicat*"           "not aberra*"
##  [7] "not abhor*"             "not abject*"            "not abnormal*"
## [10] "not abolish*"           "not abominab*"          "not abominat*"
## [13] "not abrasiv*"           "not absent*"            "not abstrus*"
## [16] "not absurd*"            "not abus*"              "not accident*"
## [19] "not accost*"            "not accursed*"          "not accusation*"
## [22] "not accuse*"            "not accusing*"          "not acerbi*"
## [25] "not ache*"              "not aching*"            "not achy"
## [28] "not acomia*"            "not acrimon*"           "not adactylism*"
## [31] "not addict*"            "not admonish*"          "not admonition*"
## [34] "not adulterat*"         "not adultery*"          "not advers*"
## [37] "not affectation*"       "not affected*"          "not affected manner*"
## [40] "not afflict*"           "not affright*"          "not affront*"
## [43] "not afraid*"            "not against"            "not ageism"
## [46] "not ageist"             "not aggravat*"          "not aggress*"
## [49] "not aggressiv*"         "not aggriev*"
```

```
sentiment <- analyzeSentiment(iconv(as.character(tweets_usa$text), to='UTF-
8'))
tokenized <- tokens_lookup(tokens(tweets_usa$text),
dictionary=data_dictionary_LSD2015, exclusive=FALSE)
sentiment$LCpos <- sapply(tokenized, function(x) sum(x=='POSITIVE') -
sum(x=='NEG_POSITIVE') + sum(x=='NEG_NEGATIVE'))
sentiment$LCneg <- sapply(tokenized, function(x) sum(x=='NEGATIVE') -
sum(x=='NEG_NEGATIVE') + sum(x=='NEG_POSITIVE'))
sentiment$LC <- (sentiment$LCpos-sentiment$LCneg)/sentiment$WordCount

vader_scores <- vader_df(tweets_usa$text)
sentiment$Vader <- vader_scores$compound



summary(sentiment$Vader)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -0.9720 -0.2960  0.0000  0.0598  0.4590  0.9850
```

```
summary(sentiment$SentimentGI)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -0.600000 -0.080000  0.000000  0.007079  0.100000  0.600000
```

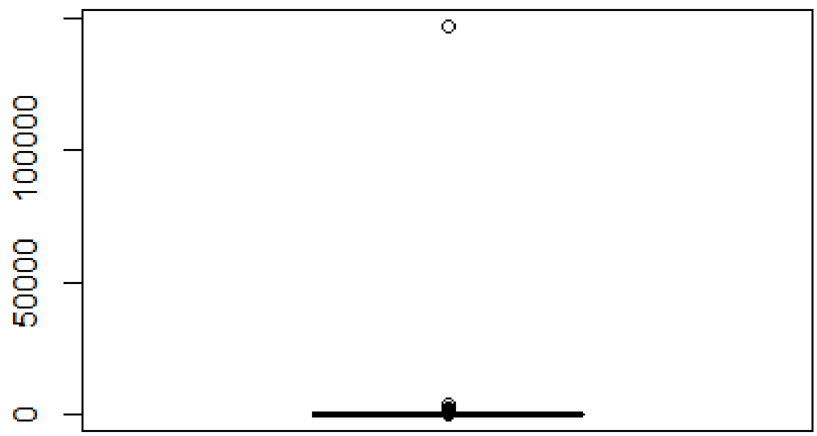
```
tweets_usa$Vader <- sentiment$Vader
```

Next we take a look at the distributions of the number of favorites each tweets receive.

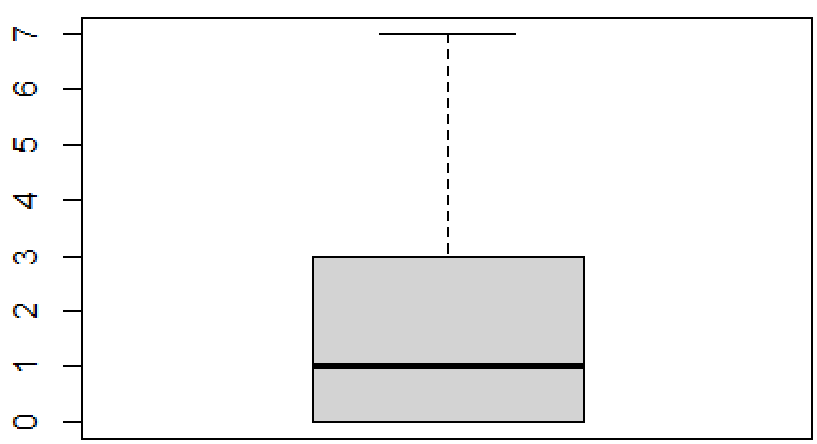
```
summary(tweets_usa$favorite_count)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
##      0.00      0.00      1.00     65.23      3.00 147083.00
```

```
boxplot(tweets_usa$favorite_count)
```




```r
boxplot(tweets_usa$favorite_count, outline = FALSE)
```



The vast majority of tweets have between 0 to 7 favorites each. Any tweets with greater than 7 favorites are considered outliers within the total distributions of tweets. For the favorites grouping, a 7 size category will be made where the score will be multiplied by the corresponding favorites size category each tweet recieve. 0 will be the first size group, as about a quarter of the tweets collected recieved no retweets. Tweets with 1-7 retweets will be given a size category 2. The remaining categories are as follows:

Size 1: 0 Favorites

Size 2: 1-7 Favorites

Size 3: 8-50 Favorites

Size 4: 51-100 Favorites

Size 5: 101-500 Favorites

Size 6: 501-1,000 Favorites

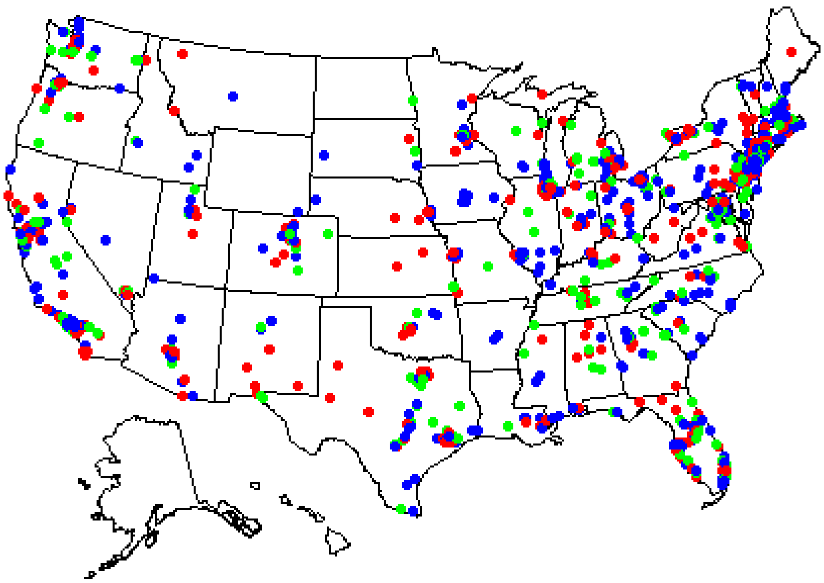
Size 7: 1001+ Favorites

```
tweets_usa$favorite_size <- ifelse(tweets_usa$favorite_count == 0,1,
       ifelse(tweets_usa$favorite_count < 7,2,
              ifelse(tweets_usa$favorite_count < 50,3,
                     ifelse(tweets_usa$favorite_count < 100,4,
                            ifelse(tweets_usa$favorite_count < 500,5,
                                   ifelse(tweets_usa$favorite_count <
1000,6,7
       ))))))

table(tweets_usa$favorite_size)

##
##    1    2    3    4    5    6    7
## 1160 1135  282   24   33    2    5
```

Next we plot the Vader scores on a US map, one for each analysis method.

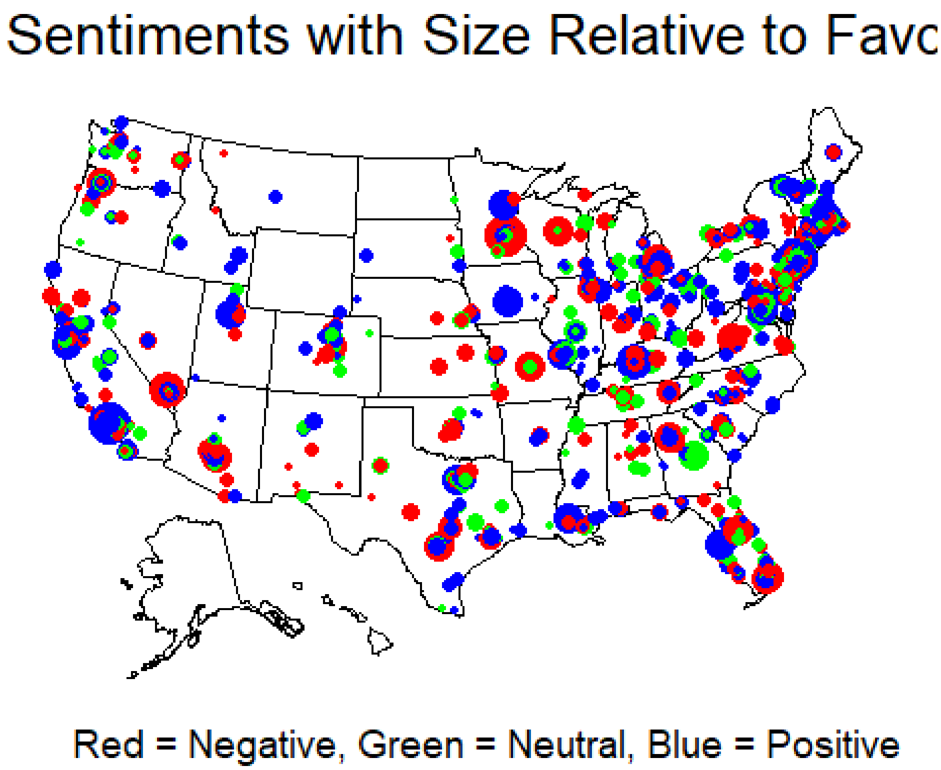
```
#equal size
map + geom_point(data = tweets_usa, aes(x = lng1, y = lat1,
                                        color = ifelse(Vader < 0,'red',
ifelse(Vader > 0, 'blue','green' )))) +
      scale_color_identity() +
  labs(title="Sentiments with Equal Size", caption = "Red = Negative, Green =
Neutral, Blue = Positive") +
  theme(text = element_text(size = 17.5))
```

Sentiments with Equal Size

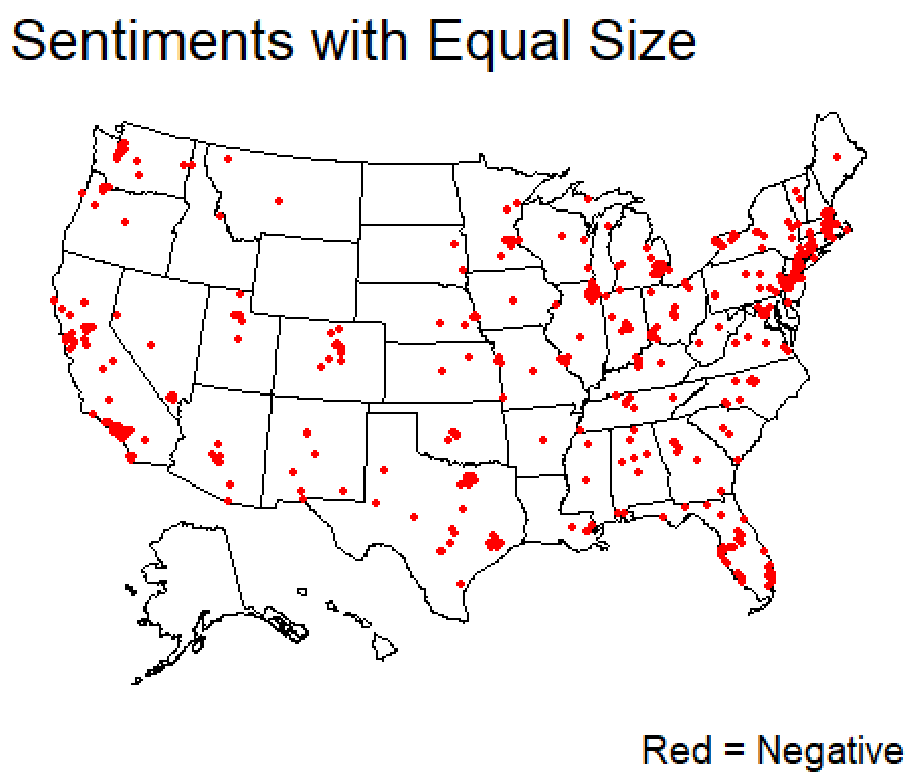


Red = Negative, Green = Neutral, Blue = Positive

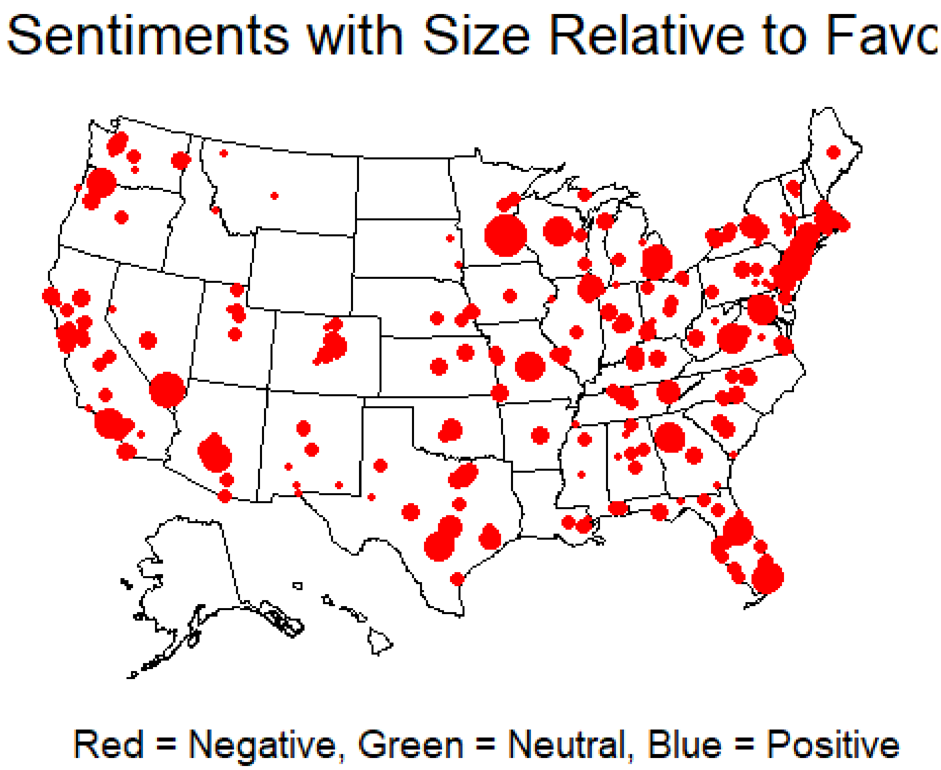
```r
#size = fav_size
map + geom_point(data = tweets_usa, aes(x = lng1, y = lat1,
                 color = ifelse(Vader < 0,'red', ifelse(Vader > 0,
'blue','green' ))),
                 size = tweets_usa$favorite_size) + scale_color_identity() +
  labs(title="Sentiments with Size Relative to Favorites", caption = "Red =
Negative, Green = Neutral, Blue = Positive") +
  theme(text = element_text(size = 17.5))
```



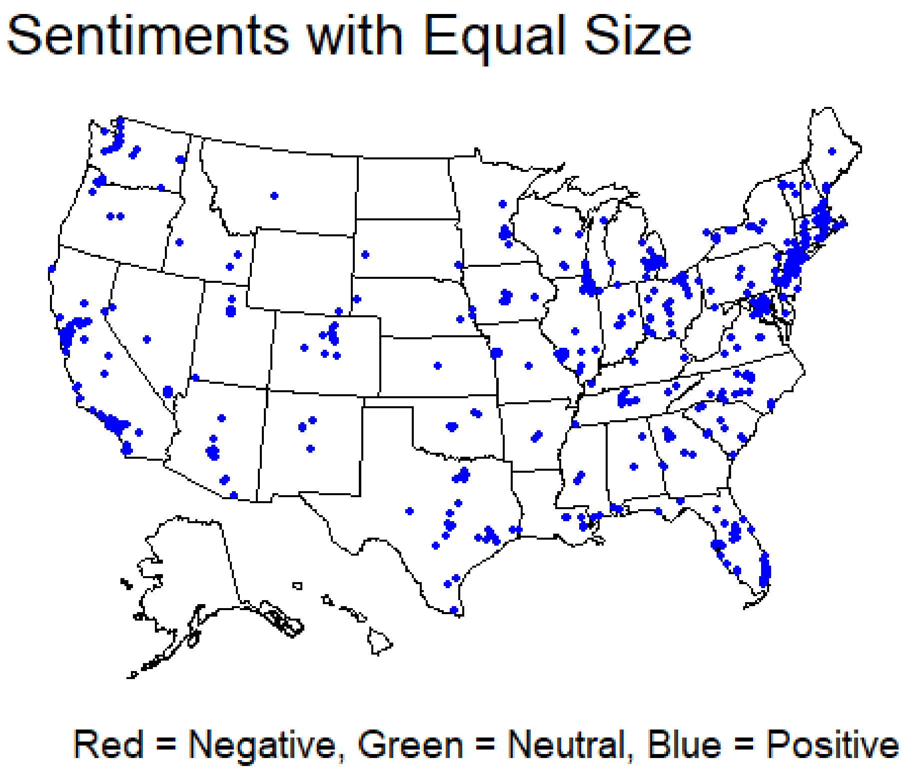

```
#tweets with negative vader scores
neg_tweets <-tweets_usa %>%
  filter(Vader < 0)

map + geom_point(data = neg_tweets, aes(x = lng1, y = lat1, color =
ifelse(Vader < 0,'red', ifelse(Vader > 0, 'blue','green' ))), size = 1 ) +
  scale_color_identity()+
  labs(title="Sentiments with Equal Size", caption = "Red = Negative") +
  theme(text = element_text(size = 17.5))
```



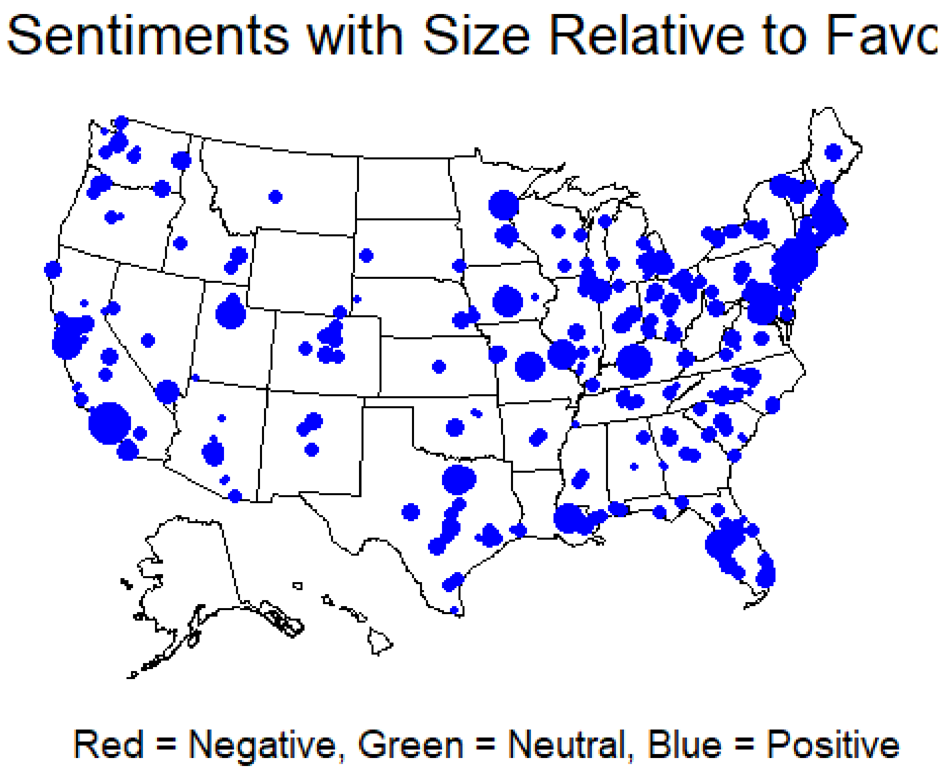

```
map + geom_point(data = neg_tweets, aes(x = lng1, y = lat1,
                                        color = ifelse(Vader < 0,'red',
ifelse(Vader > 0, 'blue','green' ))),
                 size = neg_tweets$favorite_size) + scale_color_identity() +
  labs(title="Sentiments with Size Relative to Favorites", caption = "Red =
Negative, Green = Neutral, Blue = Positive") +
  theme(text = element_text(size = 17.5))
```




```
#tweets with positive vader scores
pos_tweets <-tweets_usa %>%
  filter(Vader > 0)

map + geom_point(data = pos_tweets, aes(x = lng1, y = lat1, color =
ifelse(Vader < 0,'red', ifelse(Vader > 0, 'blue','green' ))), size = 1 ) +
  scale_color_identity()+
  labs(title="Sentiments with Equal Size", caption = "Red = Negative, Green =
Neutral, Blue = Positive") +
  theme(text = element_text(size = 17.5))
```




```
map + geom_point(data = pos_tweets, aes(x = lng1, y = lat1,
                                        color = ifelse(Vader < 0,'red',
ifelse(Vader > 0, 'blue','green' ))),
                 size = pos_tweets$favorite_size) + scale_color_identity() +
  labs(title="Sentiments with Size Relative to Favorites", caption = "Red =
Negative, Green = Neutral, Blue = Positive") +
  theme(text = element_text(size = 17.5))
```



For the most part, the tweets are coming from more populated states in the country. Wyoming and North Dakota have no tweets in them at all and therefore will be excluded from the analysis. Unfortunately, a good amount of states have only a few tweets.

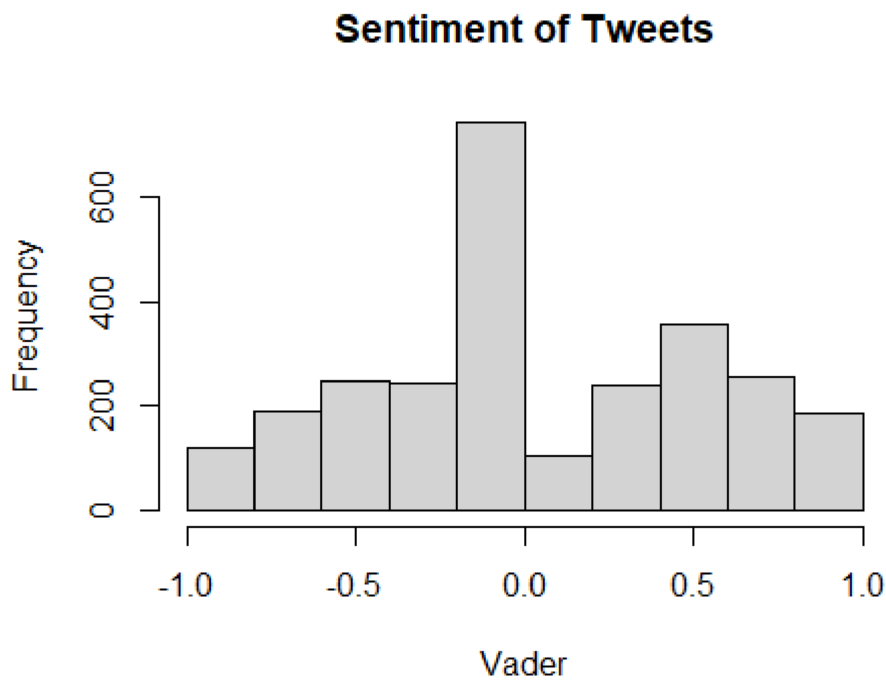
Next, the tweets will be assigned a state value by reverse geocoding. This uses Google's API.

```
head(us_vax)

location <- revgeo(latitude = tweets_usa$lat, longitude = tweets_usa$lng,
                   provider = "google", API =
"AIzaSyD934u_6vqJy5W8JcWJJ8tx0VPpPkcICB0", output = "frame")
table(location$state)
tweets_usa$state <- location$state

state_vax <- us_vax %>%
  select(state = State.Territory.Federal.Entity, vax_rate =
Percent.of.Total.Pop.Fully.Vaccinated.by.State.of.Residence)

#New York needs to be renamed to match the indicator used in CDC's name for
New York in their vaccination file.
state_vax$state[state_vax$state=="New York State"] <- "New York"

unique(tweets_state$state)
state_vax$state
```

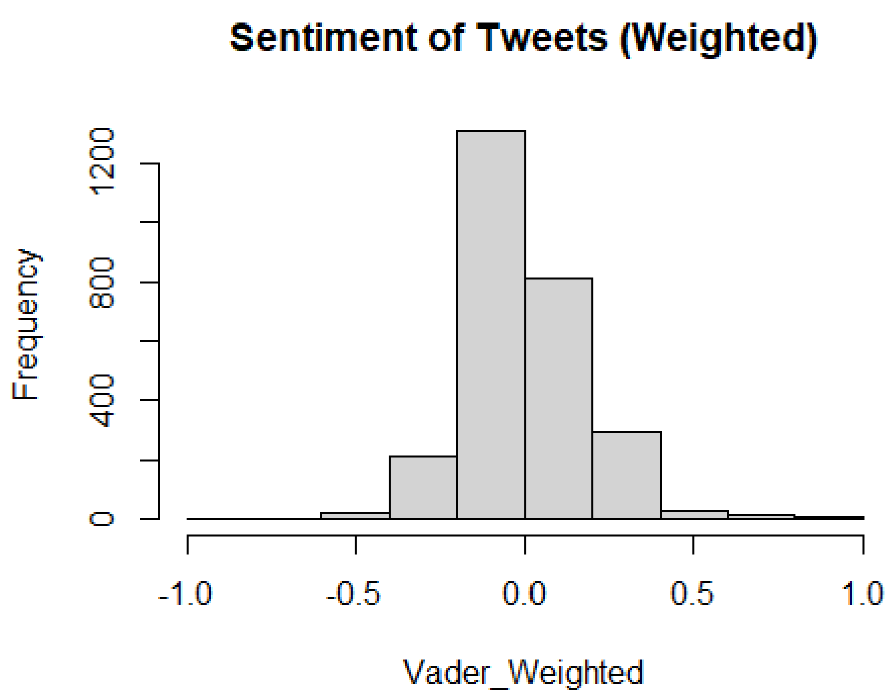
```
tweets_state <- tweets_usa %>%
  left_join(state_vax, by = c("state" = "state"))
```

One last set of plots to produce before we move on to the state level analysis is looking at the frequency of Vader scores, from each analysis method. We will transform the Vader scores of the weighted score such that the range of the Vader scores are from -1 to 1, to match the range of the unweighted scores. This is done by dividing the total Vader scores by the absolute maximum value of weighted Vader scores (5.6910).

```
tweets_state$vax_rate <- as.numeric(tweets_state$vax_rate)

table(tweets_state$vax_rate)

##
## 45.5   46 46.5 47.1 49.1 49.2 49.4 49.8 50.9 51.5 51.8   52 52.2 52.7 53.6
54.7
##    8    1   23   14    3   23   66   58   38   63   24   20   10   22   80
65
## 54.9 55.2 55.9 57.2 57.9   59 60.1 61.9 62.2 63.3 63.9 64.1 64.6 65.1 65.2
65.5
##   74  340   21   16   15   77   30  162  135   54  431   24   44    9   34
92
## 65.7 68.4 68.5 69.1 71.9 72.8 72.9 73.5 73.8
##   67   41   69  289   92   27    8    8   12

tweets_state$Vader_weighted <- tweets_state$Vader*tweets_state$favorite_size
tweets_state$Vader_weighted_trans <-
(tweets_state$Vader*tweets_state$favorite_size)/5.6910

summary(tweets_state$Vader)

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.97200 -0.29600  0.00000  0.05736  0.45900  0.98500

summary(tweets_state$Vader_weighted_trans)

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.93112 -0.07275  0.00000  0.01799  0.11386  1.00000

hist(tweets_state$Vader, main='Sentiment of Tweets', xlab='Vader')
```




```
hist(tweets_state$Vader_weighted_trans, main='Sentiment of Tweets
(Weighted)', xlab='Vader_Weighted')
```



Next we'll group Tweets by state and and see if the two Vader scores are correlated with the vaccination rates by states.

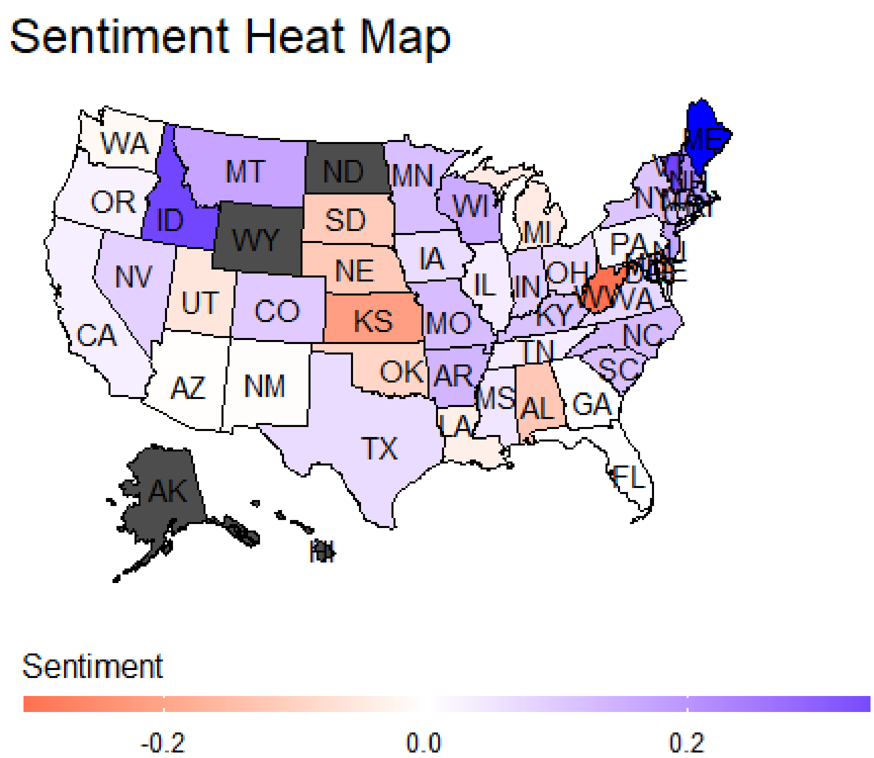
```
tweets_grouped <- tweets_state %>%
  select(state, Vader,Vader_weighted,Vader_weighted_trans,
vax_rate,favorite_size) %>%
  group_by(state) %>%
  summarize(
    n = n(),
    Vader_mean = mean(Vader),
    Vader_median = median(Vader),
    Vader_mean_weighted = mean(Vader_weighted),
    Vader_median_weigthed = median(Vader_weighted),
    Vader_mean_weighted_trans =  mean(Vader_weighted_trans),
    Vader_median_weighted_trans = median(Vader_weighted_trans),
    vax_rate = mean(vax_rate)
  ) %>%
  filter(
    n > 1
  )


cor(x =tweets_grouped$Vader_mean, y = tweets_grouped$vax_rate,
use="pairwise.complete.obs")

## [1] 0.3295965

cor(x =tweets_grouped$Vader_mean_weighted_trans, y = tweets_grouped$vax_rate,
use="pairwise.complete.obs")

## [1] 0.3338989

median(tweets_grouped$vax_rate, na.rm = TRUE)

## [1] 57.2
```

With both version of Vader scores yielding a correlation of 0.33, there is a slight positive correlation between the mean Vader score and the vaccination rates by state.

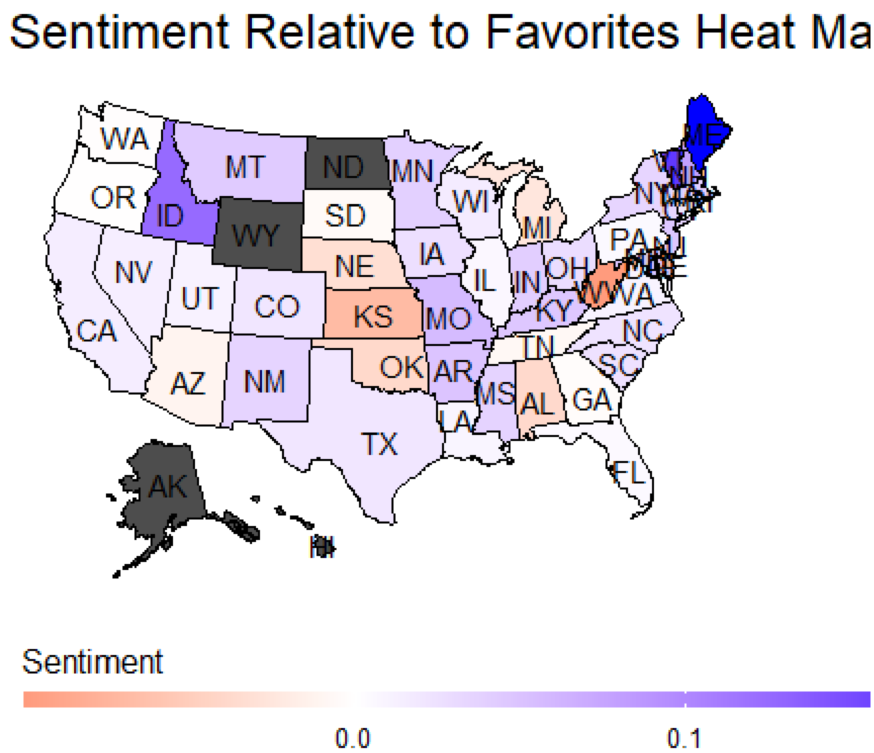
Next we'll create heat maps of the Vader scores and vaccination rates to visualize the overlaps by state. The median vaccination rate will be used as the middle point for the heat map for the vaccination rate heat map.

```
#Unweighted Scores
usmap::plot_usmap(data = tweets_grouped, values = "Vader_mean", labels = T) +
  labs(title = "Sentiment Heat Map", fill = 'Sentiment') +
  #scale_fill_gradientn(colours=heat.colors(10),na.value="grey90",
  scale_fill_gradient2(low = "red",mid = "white", high = "blue",
                          na.value="grey30",
                       guide = guide_colourbar(barwidth = 25, barheight =
0.4,
```

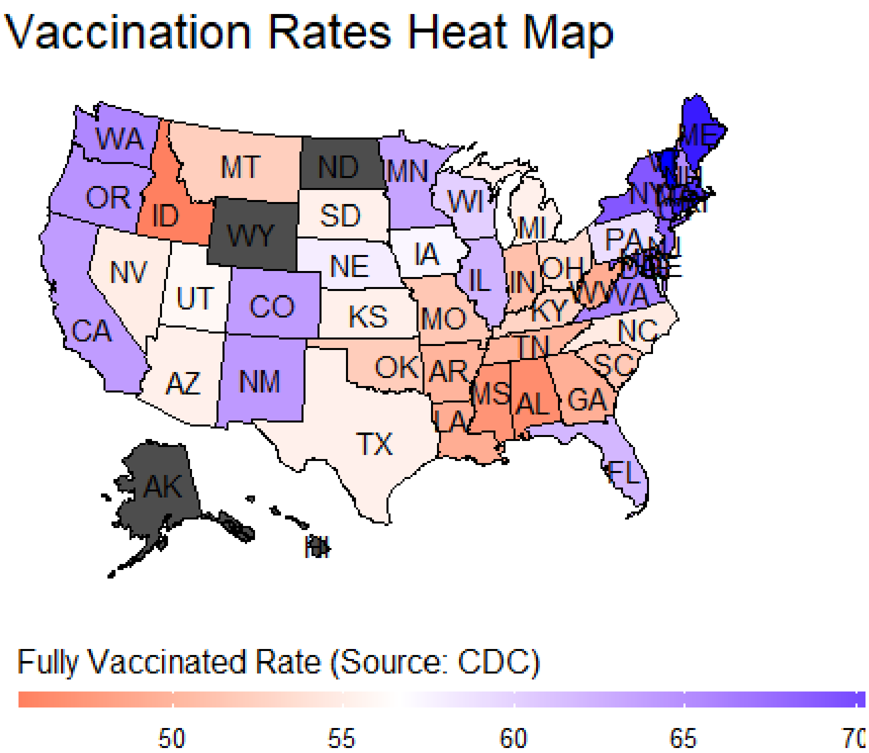
```
                                                      #put legend title on top of
legend
                                                      title.position = "top")) +
  # put legend at the bottom, adjust legend title and text font sizes
  theme(title = element_text(size=15),
        legend.position = "bottom",
        legend.title=element_text(size=12),
        legend.text=element_text(size=10))
```




```
#Weighted Scores
usmap::plot_usmap(data = tweets_grouped, values =
"Vader_mean_weighted_trans", labels = T) +
  labs(title = "Sentiment Relative to Favorites Heat Map", fill =
'Sentiment') +
  #scale_fill_gradientn(colours=heat.colors(10),na.value="grey90",
  scale_fill_gradient2(low = "red",mid = "white", high = "blue",
                       midpoint = 0, na.value="grey30",
                       guide = guide_colourbar(barwidth = 25, barheight =
0.4,
                                                      #put legend title on top of
legend
                                                      title.position = "top")) +
  # put legend at the bottom, adjust legend title and text font sizes
  theme(title = element_text(size=15),
        legend.position = "bottom",
        legend.title=element_text(size=12),
        legend.text=element_text(size=10))
```




```
#Vax rates
usmap::plot_usmap(data = tweets_grouped, values = "vax_rate", labels = T) +
  labs(title = "Vaccination Rates Heat Map", fill = 'Fully Vaccinated Rate
(Source: CDC)') +
  #scale_fill_gradientn(colours=heat.colors(10),na.value="grey90",
  scale_fill_gradient2(low = "red",mid = "white", high = "blue",
                       midpoint = 56.55, na.value="grey30",
                       guide = guide_colourbar(barwidth = 25, barheight =
0.4,
                                               #put legend title on top of
legend
                                               title.position = "top")) +
  # put legend at the bottom, adjust legend title and text font sizes
  theme(title = element_text(size=15),
        legend.position = "bottom",
        legend.title=element_text(size=12),
        legend.text=element_text(size=10))
```



The heat maps look similar between the Vader score maps and the vaccination rate map. The states that seem to differ in color are mostly from states that have really small number of tweets used for analysis. However, it can't be determined for sure if the small sample size is the main reason for disparing heat maps in these states.

Conclusion

Even though there's a slight positive correlation between mean Vader scores in a state relative to its vaccination rate, we can't conclude for sure whether there's a relationship between the two. Several limitations were presented in this research, including small n (number of tweets) in several states, and that the weighting scheme for the Vader scores could be adjusted. A downside to the weighting approach is the geographic location of Twitter users that favorites another tweet is not available. Collecting Tweets over an extended period of time would've helped with addressing the small n and should be considered if this study were to be replicated using the free rtweet package.