Final Project Draft - STA 199

due Thursday, Oct 29 at 11:59p

Ten Out of Ten: Arushi Bhatia, Luke Vermeer, Kevin Wang, Lauren May

```r
r = getOption("repos")
r["CRAN"] = "http://cran.us.r-project.org"
options(repos = r)

install.packages("viridis")
```

```
##
## The downloaded binary packages are in
##  /var/folders/vv/dyymk5b54fx_05m5pb2sy5jh0000gn/T//Rtmp7jRYpV/downloaded_packages
```

```r
install.packages("stopwords")
```

```
##
## The downloaded binary packages are in
##  /var/folders/vv/dyymk5b54fx_05m5pb2sy5jh0000gn/T//Rtmp7jRYpV/downloaded_packages
```

```r
install.packages("tidyverse")
```

```
##
## The downloaded binary packages are in
##  /var/folders/vv/dyymk5b54fx_05m5pb2sy5jh0000gn/T//Rtmp7jRYpV/downloaded_packages
```

```r
install.packages("tidytext")
```

```
##
## The downloaded binary packages are in
##  /var/folders/vv/dyymk5b54fx_05m5pb2sy5jh0000gn/T//Rtmp7jRYpV/downloaded_packages
```

```r
install.packages("wordcloud")
```

```
##
## The downloaded binary packages are in
##  /var/folders/vv/dyymk5b54fx_05m5pb2sy5jh0000gn/T//Rtmp7jRYpV/downloaded_packages
```

```r
install.packages("sentimentr")
```

```
##
## The downloaded binary packages are in
##  /var/folders/vv/dyymk5b54fx_05m5pb2sy5jh0000gn/T//Rtmp7jRYpV/downloaded_packages
```

```r
library(viridis)
```

```
## Loading required package: viridisLite
```

```r
library(stopwords)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

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

```r
library(tidytext)
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```r
library(sentimentr)
```

Introduction

Relative to other forms of media, social media plays a much greater role in how people consume news in today's technology-driven society. A study conducted in 2016 by Pew Research points out how 62% of people get news on social media (Gottfried & Shearer, 2016). As a result, social media plays an integral role in politics, as the presence of a specific subset of information on a user's feed can influence the way they categorize and see the world around them. With more and more individuals on social media such as Twitter, the role that this platform can play is significantly greater than before.

Social media, and Twitter in particular, have become an increasingly large part of the political landscape in the wake of Donald Trump's 2016 election. Trump has been active on Twitter prior to and during his presidency and uses the platform as a tool to communicate with his constituency in real time, posting updates about policy, campaigning, and his feelings on everything from members of Congress to celebrities. He is one of the first politicians to use social media this frequently and has personally referred to his use of Twitter as "modern day presidential" (Trump, 2017).

Donald Trump's Twitter also has "unpresidented" reach among the American public, boasting a follower base of over 87 million. This makes him the second most-followed political personality and sixth most-followed overall account on Twitter (Wikipedia, 2020). On top of this, Trump's Twitter also receives significant attention in the media. Over 850,000 news articles have referenced his Twitter use since 2016 and 31% of his Tweets since then have received individual media coverage (Real Clear Politics, 2019).

Because Trump uses Twitter to convey his political agendas in short blurbs, analyzing his Tweets can give a unique insight into the way that he thinks. The research question we will be exploring is how President Trump's sentiment in his Tweets and its reception by his large Twitter following (popularity) varies across people, organizations, and policies. Our hypothesis regarding our research question is that Trump's tweets have negative sentiments towards opposing groups and policies and positive sentiments towards groups and policies that align with his beliefs. In addition, we believe the tweets with negative sentiments will be slightly more popular in terms of favorites and retweets.

Data description

The dataset was extracted from a website - TrumpTwitterArchive.com. The original curator of the data created their own Twitter scraper in order to obtain the data. They utilized Python, Selenium (which is a software suite that allows the automation of tests utilizing web browsers), and Tweepy (a Python library for accessing the Twitter API). Since Twitter makes it challenging to scrape all of a user's Tweets in one go, the way to get around this is to individually search for a specific day and extract all the Tweets from that user on that specific day. To do this manually would take ages, but the scraper that the curator built allows for automated accessing for any desired day and also a range of days. The scraper then obtains the Tweet ID, which contains all of the metadata of the Tweet, and then uses the metadata to obtain all the other information about the Tweet (such as the text, timestamp, number of favorites, etc.). This other information is then compiled into a dataset, which is made available to the public. This dataset is updated every minute, which also means that deleted Tweets would most likely also appear in this dataset.

This data set entitled trumpTweets includes 53,697 observations. Each individual observation is one of President Donald Trump's tweets. The original dataset contains 7 variables: source, text, created_at, retweet_count, favorite_count, is_retweeted, id_str. The descriptions of each of the original variables is given below.

- source: Original source where tweet was posted
- text: text of the tweet
- created_at: Date and time the tweet was posted/created, provides context
- retweet_count: number of retweets
- favorite_count: number of favorites
- is_retweeted: whether or not the tweet was originally posted on a different account and Trump retweeted
- id_str: The scrape.py script collects tweet ids. If you know a tweet's id number, you can get all the information available about that tweet using Tweepy
- text, timestamp, number of retweets / replies / favorites, geolocation, etc.

Glimpse of data

```
trumpTweets <- read_csv("data/trumpTweetCSV.csv")
glimpse(trumpTweets)
```

```
## Rows: 53,697
## Columns: 7
## $ source         <chr> "Twitter for iPhone", "Twitter for iPhone", "Twitter...
## $ text           <chr> "https://t.co/g5lGbRG4aE", "Will be interviewed by @...
## $ created_at     <chr> "10/8/20 12:08", "10/8/20 11:47", "10/8/20 2:57", "1...
## $ retweet_count  <dbl> 10524, 7434, 71908, 22541, 26360, 28868, 12852, 2362...
## $ favorite_count <dbl> 41050, 38386, 470060, 98083, 89625, 76854, 44386, 10...
## $ is_retweet     <lgl> FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FALSE, FAL...
## $ id_str         <dbl> 1.31418e+18, 1.31417e+18, 1.31404e+18, 1.31404e+18, ...
```

```
trumpTweets <- trumpTweets %>%
   mutate(obama = case_when((grepl("Obama", text , ignore.case = TRUE) |
                              grepl("Barack", text , ignore.case = TRUE)) &
                            !(grepl("Michelle", text , ignore.case = TRUE))~ 1,
                            TRUE ~0))

trumpTweets <- trumpTweets %>%
   mutate(biden = case_when((grepl("Biden", text , ignore.case = TRUE) |
                              grepl("Joe", text , ignore.case = TRUE)) ~ 1,
                            TRUE ~0))

trumpTweets <- trumpTweets %>%
   mutate(pelosi = case_when((grepl("Pelosi", text , ignore.case = TRUE) |
                              grepl("Nancy", text , ignore.case = TRUE)) &
                            ! grepl("@NancyMace", text, ignore.case = TRUE)~ 1,
                            TRUE ~0))


trumpTweets <- trumpTweets %>%
   mutate(kamala = case_when((grepl("Kamala", text , ignore.case = TRUE))~ 1,
                            TRUE ~0))
```

```r
trumpTweets <- trumpTweets %>%
   mutate(hillary = case_when((grepl("Hillary", text , ignore.case = TRUE) |
                           grepl("Clinton", text , ignore.case = TRUE)) &
                        !(grepl("Bill Clinton", text , ignore.case = TRUE))~ 1,
                          TRUE ~0))

trumpTweets <- trumpTweets %>%
   mutate(aoc = case_when((grepl("AOC", text , ignore.case = TRUE) |
                           grepl("Ocasio", text , ignore.case = TRUE) |
                          grepl("Cortez", text , ignore.case = TRUE)|
                           grepl("Alexandria", text , ignore.case = TRUE))~ 1,
                          TRUE ~0))

trumpTweets <- trumpTweets %>%
   mutate(pence = case_when((grepl("Pence", text , ignore.case = TRUE)) ~ 1,
                          TRUE ~0))

trumpTweets <- trumpTweets %>%
   mutate(nikki = case_when((grepl("Nikki Haley", text , ignore.case = TRUE)|
                           grepl("Nikki", text , ignore.case = TRUE)) ~ 1,
                          TRUE ~0))


trumpTweets <- trumpTweets %>%
  mutate(mcconnell = case_when(((grepl("McConnell", text , ignore.case = TRUE) |
                           grepl("Mitch", text , ignore.case = TRUE) |
                          grepl("@Team_Mitch", text , ignore.case = TRUE)) &
                        ! grepl("@mitchellvii", text , ignore.case = TRUE))~ 1,
                          TRUE ~0))

trumpTweets <- trumpTweets %>%
   mutate(fauci = case_when((grepl("Fauci", text , ignore.case = TRUE)) ~ 1,
                          TRUE ~0))

trumpTweets <- trumpTweets %>%
   mutate(amy = case_when((grepl("Coney Barrett", text , ignore.case = TRUE) |
                           grepl("ACB", text , ignore.case = TRUE)|
                           grepl("Judge Amy", text , ignore.case = TRUE)) ~ 1,
                          TRUE ~0))
```

```r
trumpTweets <- trumpTweets %>%
   mutate(covid = case_when((grepl("COVID", text , ignore.case = TRUE) |
                           grepl("Corona", text , ignore.case = TRUE) |
                    grepl("Chinese virus", text , ignore.case = TRUE) |
                      grepl("China virus", text , ignore.case = TRUE) |
                      grepl("Chinese plague", text, ignore.case = TRUE) |
                      grepl("Wuhan virus", text , ignore.case = TRUE) |
                      grepl("Kung flu", text , ignore.case = TRUE)|
                      grepl("cdc", text , ignore.case = TRUE)|
                      grepl("fauci", text , ignore.case = TRUE)) ~ 1,
                          TRUE ~0))

trumpTweets <- trumpTweets %>%
```

```r
  mutate(climateChange = case_when((grepl("Climate change", text ,
                                          ignore.case = TRUE) |
                          grepl("Global warming", text , ignore.case = TRUE) |
                      grepl("Environment", text , ignore.case = TRUE) |
                grepl("Paris Accord", text , ignore.case = TRUE) |
                 grepl("Paris Climate Agreement", text, ignore.case = TRUE) |
                      grepl("Pollute", text , ignore.case = TRUE) |
                      grepl("Pollution", text , ignore.case = TRUE)|
                      grepl("Greta Thunberg", text, ignore.case = TRUE)|
                      grepl("Greta", text, ignore.case = TRUE)) ~ 1,
                            TRUE ~0))

trumpTweets <- trumpTweets %>%
  mutate(abortion = case_when((grepl("Abortion", text , ignore.case = TRUE) |
                        grepl("Planned parenthood", text , ignore.case = TRUE) |
                      grepl("Abort", text , ignore.case = TRUE) |
                        grepl("pro-life", text , ignore.case = TRUE) |
                        grepl("pro-choice", text , ignore.case = TRUE)) ~ 1,
                            TRUE ~0))


trumpTweets <- trumpTweets %>%
  mutate(blm = case_when((grepl("BLM", text , ignore.case = TRUE) |
                       grepl("Black lives matter", text , ignore.case = TRUE) |
                      grepl("loot", text , ignore.case = TRUE) |
                        grepl("riot", text , ignore.case = TRUE)) ~ 1,
                            TRUE ~0))

trumpTweets <- trumpTweets %>%
  mutate(guns = case_when((grepl("NRA", text , ignore.case = TRUE) |
                          grepl("Second amendment", text , ignore.case = TRUE) |
                      grepl("gun", text , ignore.case = TRUE) |
                        grepl("bear arms", text , ignore.case = TRUE) |
                        grepl("rifle", text , ignore.case = TRUE) ) ~ 1,
                            TRUE ~0))

trumpTweets <- trumpTweets %>%
  mutate(news = case_when((grepl("news", text , ignore.case = TRUE) |
                      grepl("CNN", text , ignore.case = TRUE) |
                        grepl("Fox", text , ignore.case = TRUE) |
                        grepl("media", text , ignore.case = TRUE)) ~ 1,
                            TRUE ~0))

trumpTweets <- trumpTweets %>%
  mutate(usa = case_when((grepl("usa", text , ignore.case = TRUE) |
                              grepl("america", text , ignore.case = TRUE) |
                      grepl("united states", text , ignore.case = TRUE) |
                        grepl("our country", text , ignore.case = TRUE)) ~ 1,
                            TRUE ~0))

trumpTweets <- trumpTweets %>%
  mutate(russia = case_when((grepl("russia", text , ignore.case = TRUE) |
                              grepl("putin", text , ignore.case = TRUE)|
```

```r
                            grepl("communism", text , ignore.case = TRUE)) ~ 1,
                            TRUE ~0))

trumpTweets <- trumpTweets %>%
   mutate(immigration = case_when((grepl("immigrants", text ,
                                      ignore.case = TRUE) |
                              grepl("immigration", text , ignore.case = TRUE) |
                      grepl("foreigner", text , ignore.case = TRUE) |
                        grepl("open border", text , ignore.case = TRUE) |
                        grepl("the wall", text , ignore.case = TRUE) |
                        grepl("border wall", text , ignore.case = TRUE)|
                        grepl("Mexicans", text , ignore.case = TRUE)|
                        grepl("refugees", text , ignore.case = TRUE)|
                        grepl("ice", text , ignore.case = TRUE)|
                        grepl("border patrol", text , ignore.case = TRUE)|
                        grepl("cages", text , ignore.case = TRUE)) ~ 1,
                            TRUE ~0))


trumpTweets$element_id <- 1:nrow(trumpTweets)
```

```r
trumpTweetsSentiment <- sentiment_by(trumpTweets$text)
```

```
## Warning: Each time `sentiment_by` is run it has to do sentence boundary disambiguation when a
## raw `character` vector is passed to `text.var`. This may be costly of time and
## memory.  It is highly recommended that the user first runs the raw `character`
## vector through the `get_sentences` function.
```

```r
tweetsWithSentiment <- left_join(trumpTweets, trumpTweetsSentiment,
                                 by = "element_id")

tweetsWithSentiment <- tweetsWithSentiment %>%
   mutate(posNeg = case_when(ave_sentiment>0 ~ "positive",
                             ave_sentiment<0 ~ "negative",
                             ave_sentiment ==0 ~ "neutral"))

tweetsWithSentimentWithoutLink <- tweetsWithSentiment %>%
  filter(!grepl("^https*", text))
```

```
## Warning in strwidth(words[i], cex = size[i], ...): conversion failure on 'don't'
## in 'mbcsToSbcs': dot substituted for <e2>

## Warning in strwidth(words[i], cex = size[i], ...): conversion failure on 'don't'
## in 'mbcsToSbcs': dot substituted for <80>

## Warning in strwidth(words[i], cex = size[i], ...): conversion failure on 'don't'
## in 'mbcsToSbcs': dot substituted for <99>

## Warning in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt =
## rotWord * : conversion failure on 'don't' in 'mbcsToSbcs': dot substituted for
## <e2>

## Warning in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt =
## rotWord * : conversion failure on 'don't' in 'mbcsToSbcs': dot substituted for
## <80>

## Warning in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt =
```

```
## rotWord * : conversion failure on 'don’t' in 'mbcsToSbcs': dot substituted for
## <99>

## Warning in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt =
## rotWord * : font metrics unknown for Unicode character U+2019

## Warning in strwidth(words[i], cex = size[i], ...): conversion failure on 'it’s'
## in 'mbcsToSbcs': dot substituted for <e2>

## Warning in strwidth(words[i], cex = size[i], ...): conversion failure on 'it’s'
## in 'mbcsToSbcs': dot substituted for <80>

## Warning in strwidth(words[i], cex = size[i], ...): conversion failure on 'it’s'
## in 'mbcsToSbcs': dot substituted for <99>

## Warning in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt =
## rotWord * : conversion failure on 'it’s' in 'mbcsToSbcs': dot substituted for
## <e2>

## Warning in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt =
## rotWord * : conversion failure on 'it’s' in 'mbcsToSbcs': dot substituted for
## <80>

## Warning in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt =
## rotWord * : conversion failure on 'it’s' in 'mbcsToSbcs': dot substituted for
## <99>

## Warning in text.default(x1, y1, words[i], cex = size[i], offset = 0, srt =
## rotWord * : font metrics unknown for Unicode character U+2019
```

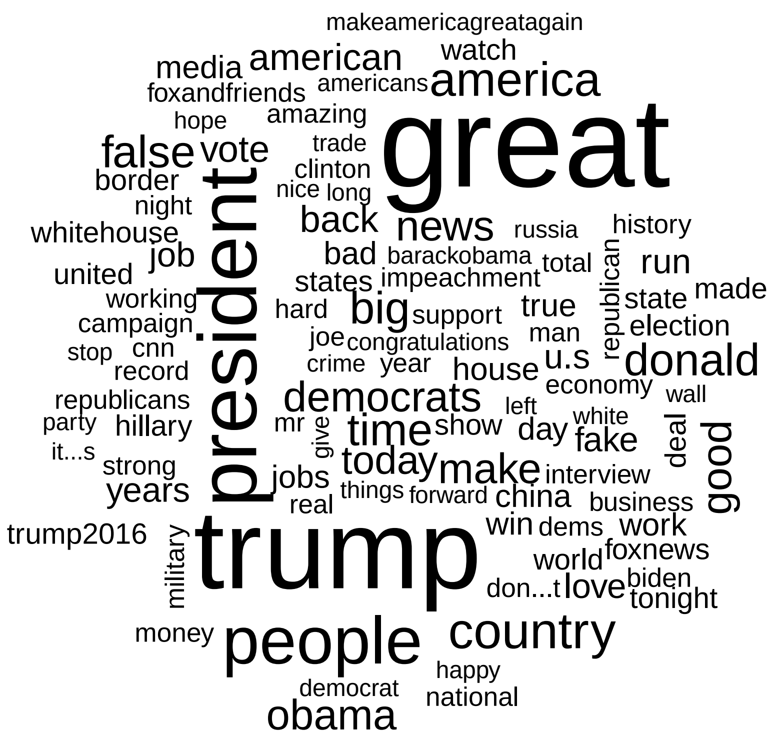


Figure 1: Word cloud of the top 100 words Donald Trump uses in his Tweets. Some of the words that were not part of the actual text of the Tweet or were artifacts of Tweets (such as URLs, account names, and years) were removed to clean the data.

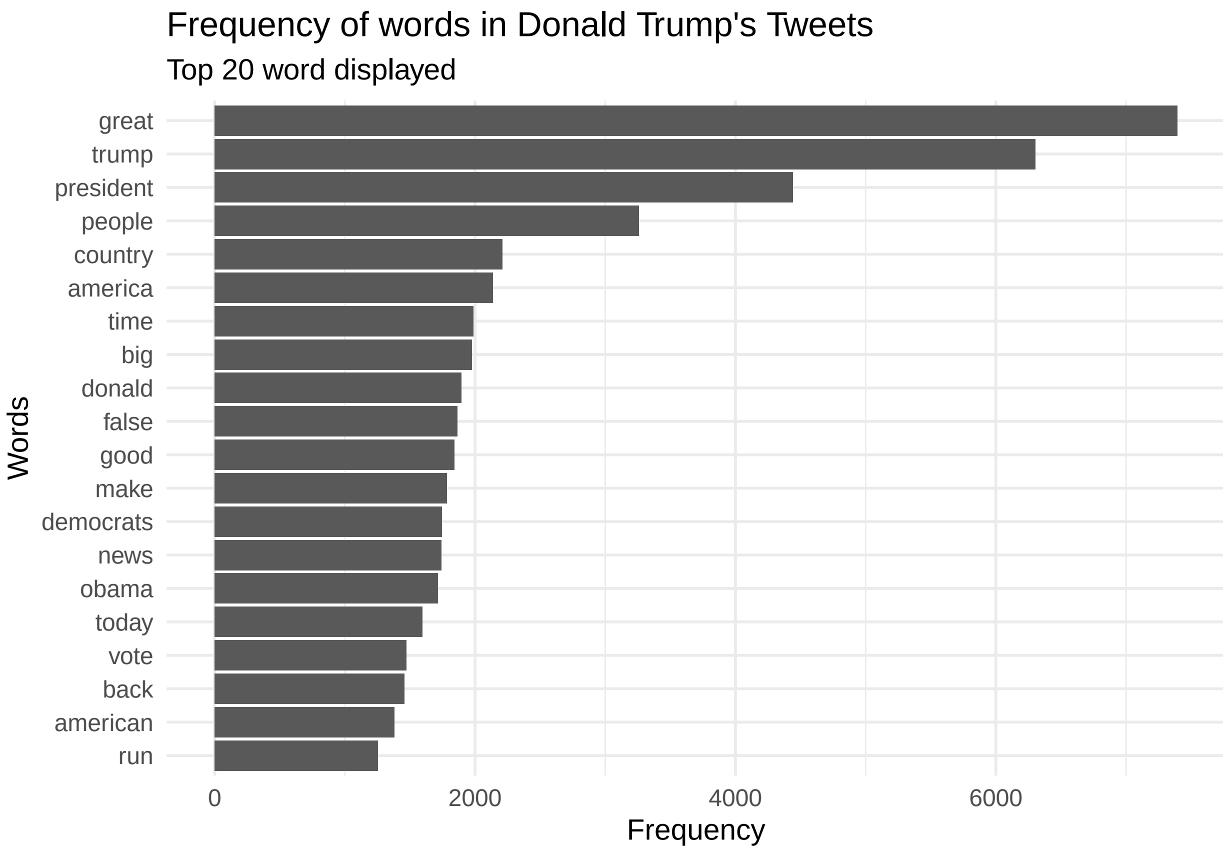


Figure 2: Frequency of Donald Trump's top 20 words that are used in his Tweets. Just like in Figure 1, some of the words that were not part of the actual text of the Tweet or were artifacts of Tweets (such as URLs, account names, and years) were removed to clean the data.

```
peopleData <- tweetsWithSentimentWithoutLink %>%
  pivot_longer(obama:amy, names_to = "person", values_to = "existenceOfPerson")

peopleData <- peopleData %>% filter(existenceOfPerson==1)

ggplot(data=peopleData, aes(x=person, fill = factor(posNeg))) +
  geom_bar(position = "fill") +
  labs(title="Lowest proportion of Tweets with negative sentiment were Tweets
  mentioning Amy Coney Barrett and Mike Pence",
                x="Politician",y ="Percentage of Tweets", fill="Sentiment",
  tag="Figure 3") +
   theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

Figure 3

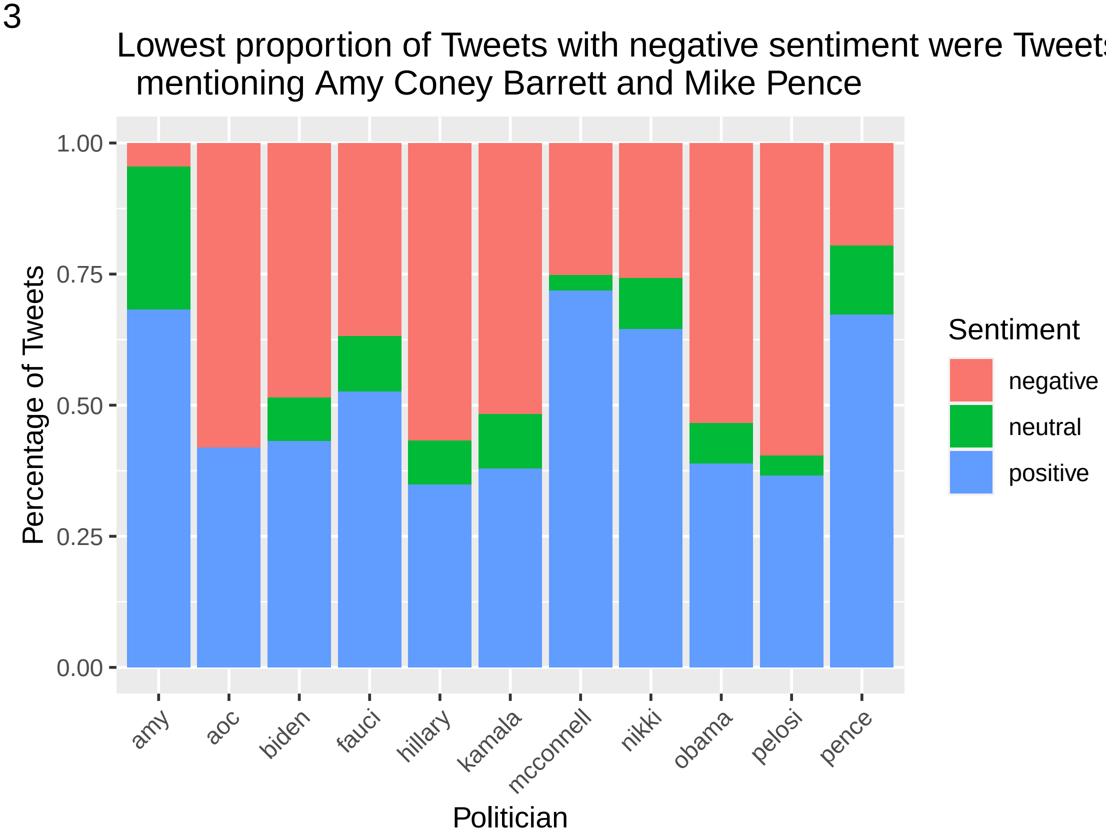


Figure 3: Illustrates the proportion of Tweets with average negative (<0), neutral (=0), and positive (>0) sentiment that mention a specific politician.

```
peopleData <- peopleData %>%
   mutate(party=case_when((person == "obama" |
               person == "biden"|
               person == "kamala" |
               person == "pelosi" |
               person == "aoc"|
                  person == "hillary") ~ "Democratic",
               (person == "pence" |
                   person == "amy" |
                   person == "niki" |
                  person == "mcconnell") ~ "Republican"))

peopleData <- peopleData %>%
   mutate(gender=case_when((person=="obama" |
              person=="biden"|
               person=="pence"|
               person=="fauci"|
               person=="mcconnell") ~ "Male",
               (person=="kamala" |
                   person=="pelosi" |
                   person=="aoc"|
                   person=="amy" |
                   person=="nikki"|
                  person=="hillary") ~ "Female"))
```

```r
ggplot(data = peopleData %>%
          filter(party=="Democratic" | party == "Republican"),
       mapping = aes(x = party, y = ave_sentiment,color=gender)) +
   geom_boxplot() +
labs(title = "Overall more positive sentiment in Tweets mentioning Republicans",
       subtitle="Generally lower sentiment for Democratic females vs. males,
   generally higher sentiment for Republican females vs. males",
        x = "Party", y = "Tweet Average Sentiment Score",
   tag="Figure 4") +
   geom_hline(yintercept=0, linetype="dashed", color = "red")
```

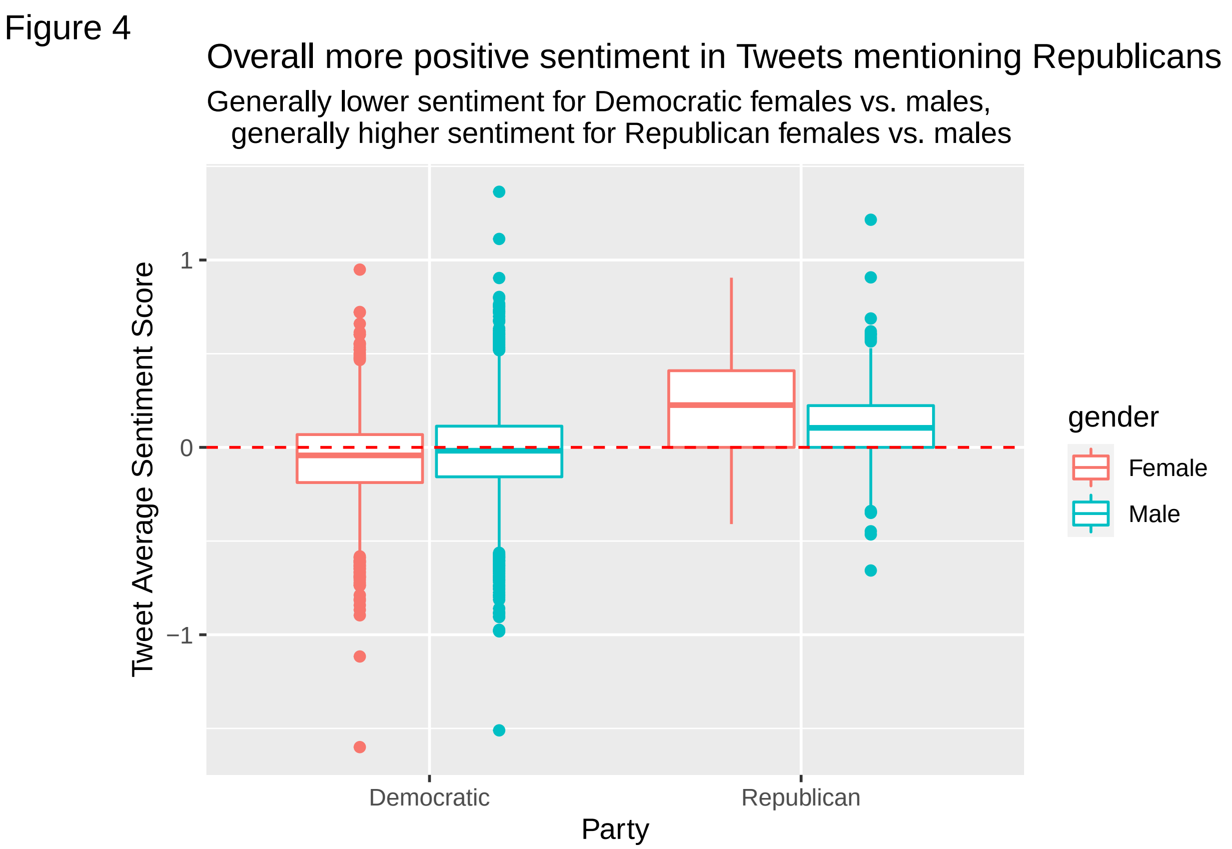


Figure 4: Boxplot showing the average sentiment scores for Tweets, divided by political party of the person mentioned in the Tweet, and split within party to show any differences in how he talks about politicians of different genders within either the Democratic or Republican party.

```r
tweetsWithSentimentDateTime <- tweetsWithSentimentWithoutLink %>%
 separate(created_at, c("date", "time"), sep = " ")
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 3 rows [52223,
## 52346, 52373].
```

```r
tweetsWithSentimentDateTimeFormat <- tweetsWithSentimentDateTime %>%
 mutate(date = as.Date(date, tryFormats = c("%m/%d/%y")))%>%
 pivot_longer(covid:immigration, names_to = "topic", values_to =
                "existenceOfTopic") %>%
 filter(existenceOfTopic == 1)

tweetsWithSentimentDateTimeFormatGroup <- tweetsWithSentimentDateTimeFormat %>%
```

```r
  mutate(month = format(date, "%m"), year = format(date, "%Y")) %>%
  group_by(topic, year) %>%
  summarize(avgSentimentForTopic = mean(ave_sentiment))
```

```
## `summarise()` regrouping output by 'topic' (override with `.groups` argument)
```

```r
temp <- tweetsWithSentimentDateTimeFormatGroup %>% filter(topic=="climateChange"
  | topic=="news" | topic == "guns" | topic == "immigration") %>%
  filter(!is.na(year))

ggplot(temp,
       aes(x= year, y = avgSentimentForTopic, color = topic,
           group=topic)) + theme(plot.margin = unit(c(1,0,1,0), "cm")) +
  geom_point() +
  geom_line() +
  geom_vline(xintercept = "2016", linetype="dotted",
             color = "blue", size=1.5) +
  labs(title = "Average sentiment for specific topics across the years",
       subtitle="Blue line indicates Trump's election,
   immigration and news follow similar patterns",
  x = "Year", y = "Tweet Average Sentiment Score Per Topic", color="Topic",
  tag="Figure 5") +
  geom_hline(yintercept=0, linetype="dashed", color = "red")
```

Figure 5

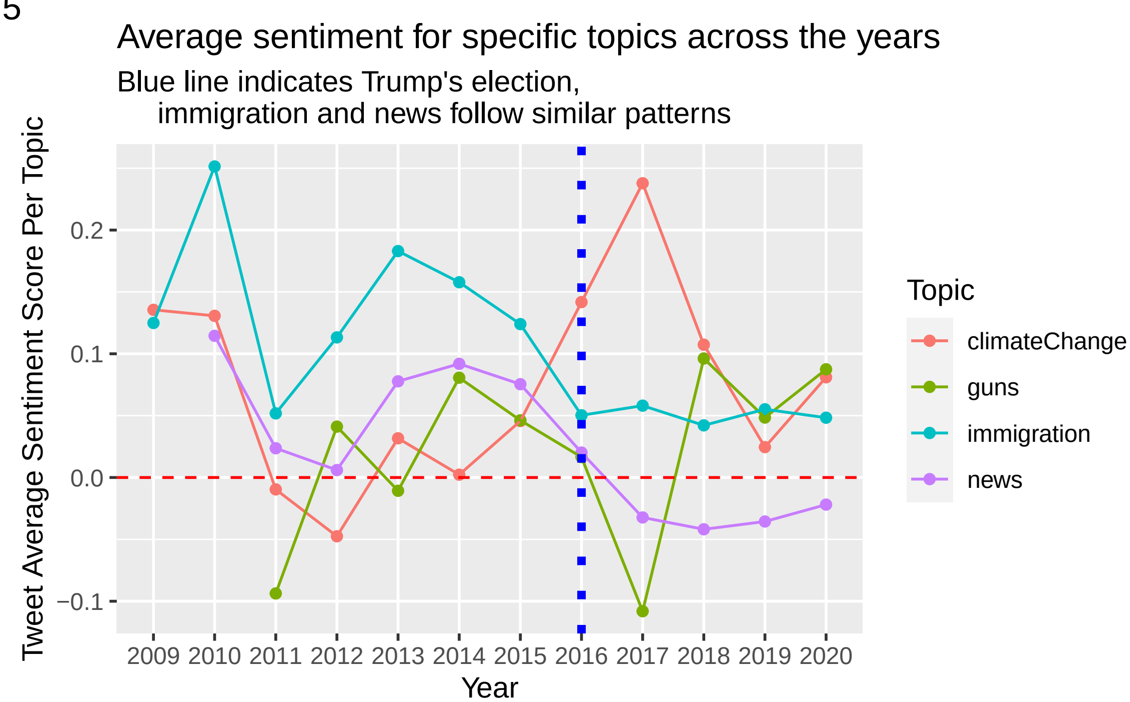


Figure 5: Shows change of average sentiment of Tweets referring to a specific topic across the years. Only showing a few topics, we selected the ones that have been Tweeted about for a long time. The vertical blue line indicates when Trump first became elected.

```r
polarization_tweets <- tweetsWithSentimentWithoutLink %>%
  mutate(polarization = abs(ave_sentiment))

chisq.test(table(polarization_tweets$posNeg,
```

```
polarization_tweets$retweet_count))
```

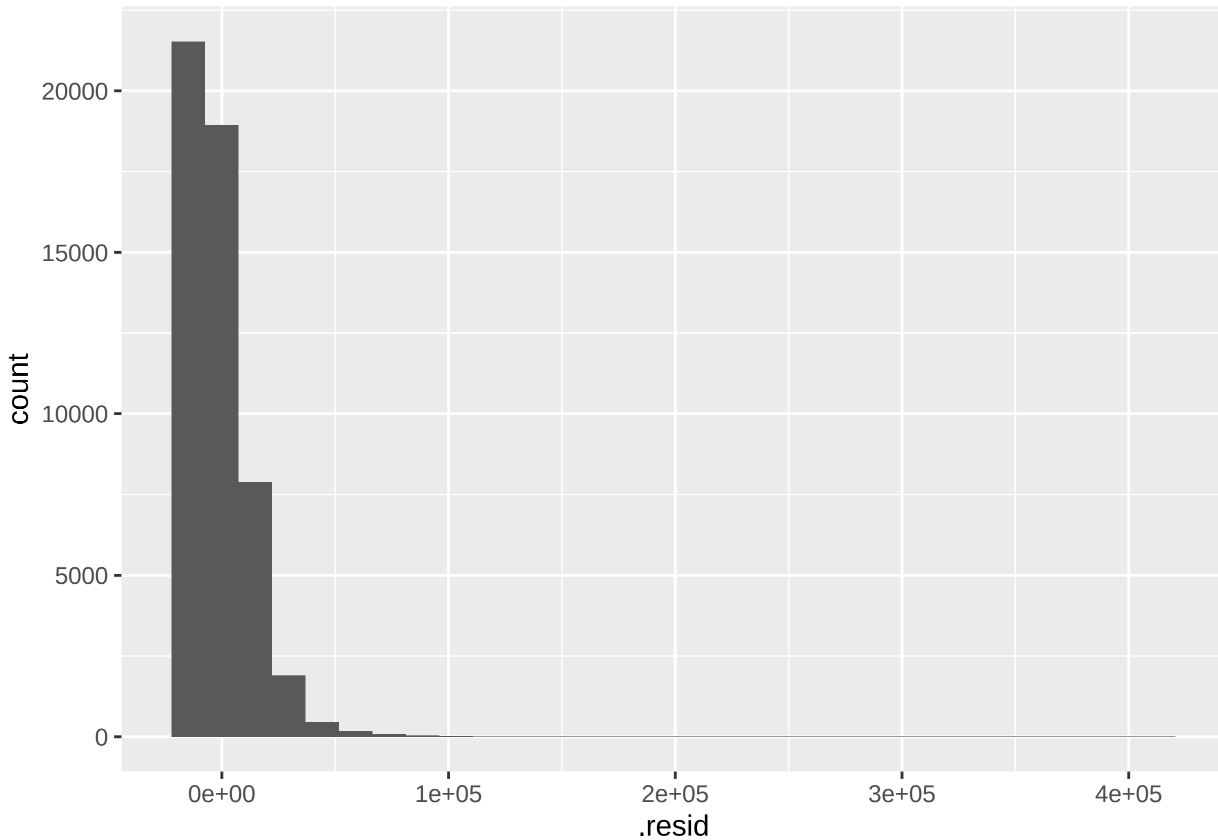
```
## Warning in chisq.test(table(polarization_tweets$posNeg,
## polarization_tweets$retweet_count)): Chi-squared approximation may be incorrect

##
##  Pearson's Chi-squared test
##
## data:  table(polarization_tweets$posNeg, polarization_tweets$retweet_count)
## X-squared = 41580, df = 39554, p-value = 6.949e-13
```

```
person_retweets <- lm(retweet_count ~ obama + biden + pelosi + kamala + hillary
                      + aoc + pence + nikki + mcconnell + amy + fauci,
                 data = tweetsWithSentimentWithoutLink)

tidy(person_retweets)
```

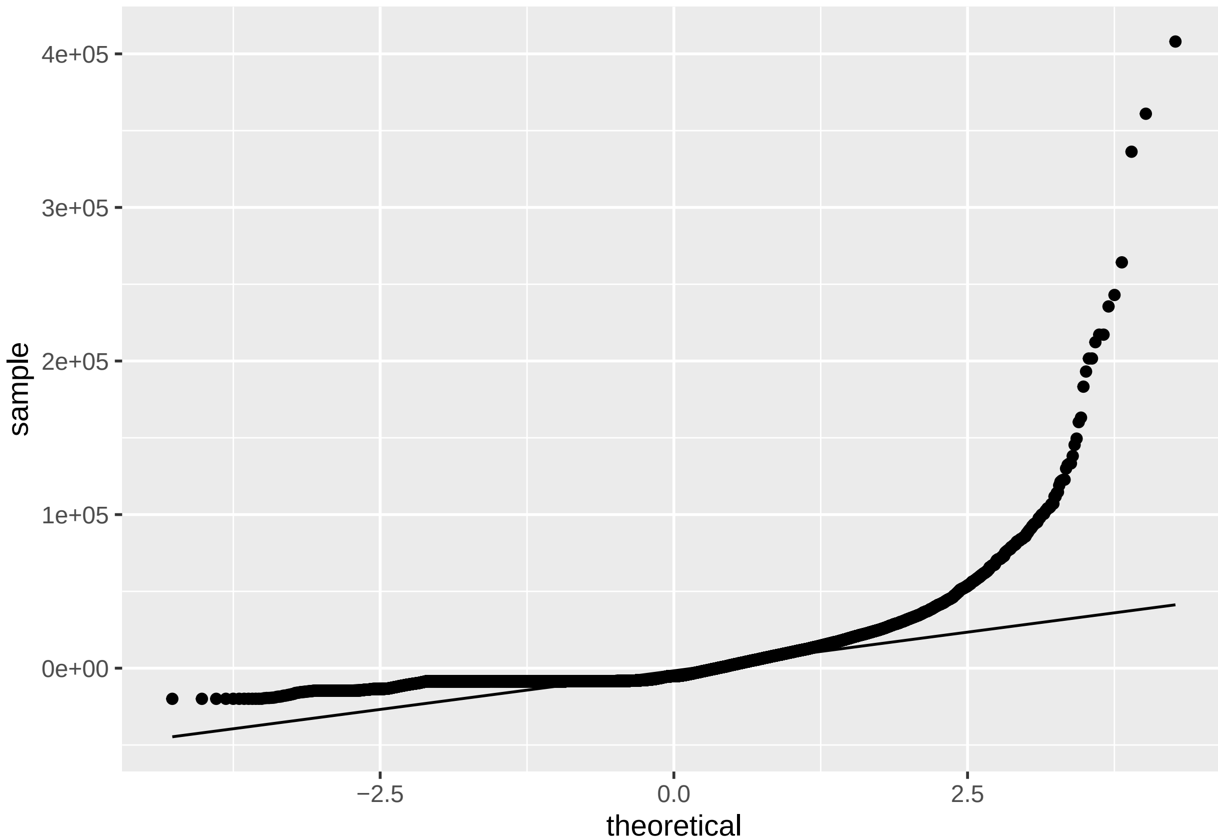
```
## # A tibble: 12 x 5
##    term        estimate std.error statistic  p.value
##    <chr>          <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)    8533.      60.9   140.    0.
##  2 obama         -3182.     243.    -13.1   4.62e-39
##  3 biden          6094.     382.     16.0   3.02e-57
##  4 pelosi        11419.     574.     19.9   1.21e-87
##  5 kamala         5182.    2406.      2.15  3.13e- 2
##  6 hillary        5023.     362.     13.9   1.16e-43
##  7 aoc            7753.    2326.      3.33  8.62e- 4
##  8 pence          1258.     783.      1.61  1.08e- 1
##  9 nikki         -2838.    2444.     -1.16  2.46e- 1
## 10 mcconnell      1595.    1295.      1.23  2.18e- 1
## 11 amy            1854.    2758.      0.672 5.01e- 1
## 12 fauci         11554.    2967.      3.89  9.86e- 5
```

```
aug <- augment(person_retweets)

ggplot(data=aug, aes(x=.resid)) + geom_histogram()
```

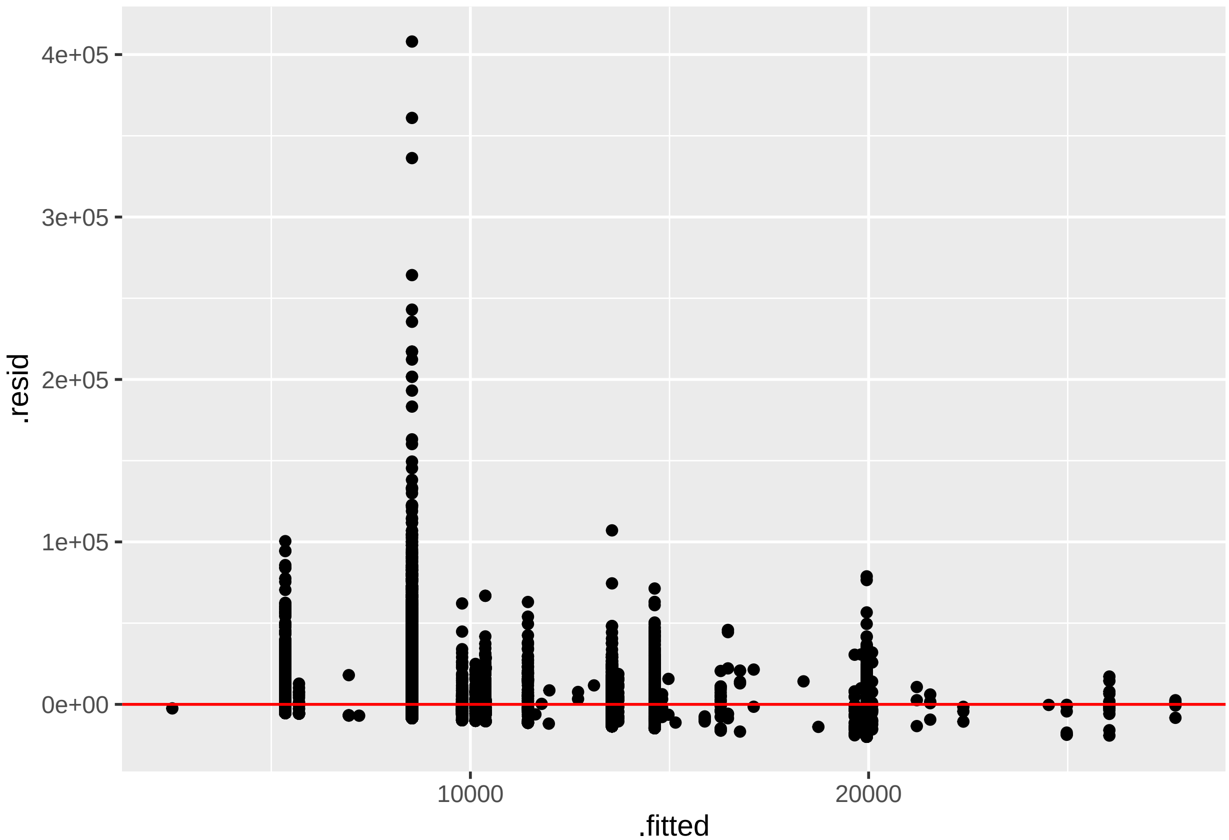
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```r
ggplot(aug, mapping = aes(sample = .resid)) +
  stat_qq() + stat_qq_line()
```




```
ggplot(data=aug, aes(x=.fitted, y=.resid)) + geom_point()+ geom_hline(yintercept=0, col="red")
```




```
#fix this, correlation between people?
trees %>% summarize(correlation = cor(Height, Girth))
```

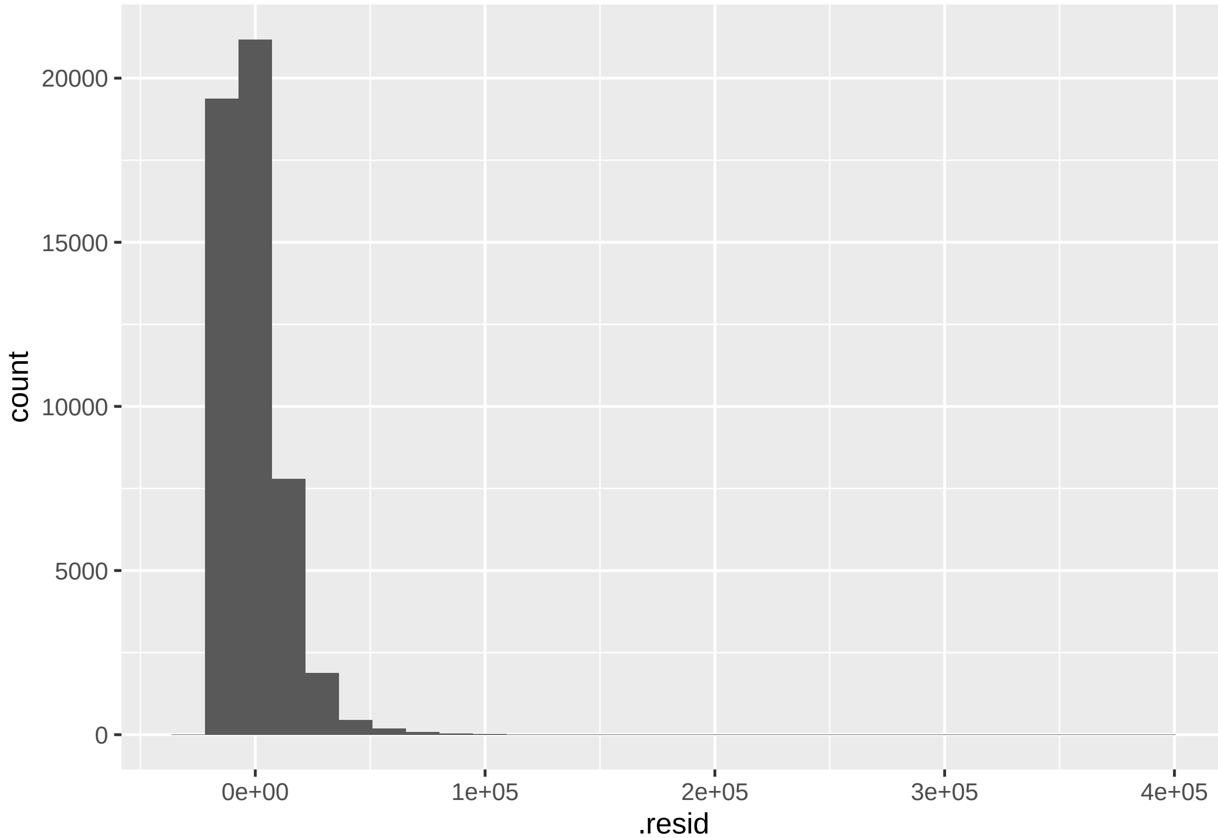
```
##   correlation
## 1   0.5192801
```

```
topic_retweets <- lm(retweet_count ~ covid + climateChange + abortion + blm +
                        guns + news + usa + russia + immigration,
                 data = tweetsWithSentimentWithoutLink)

tidy(topic_retweets)
```

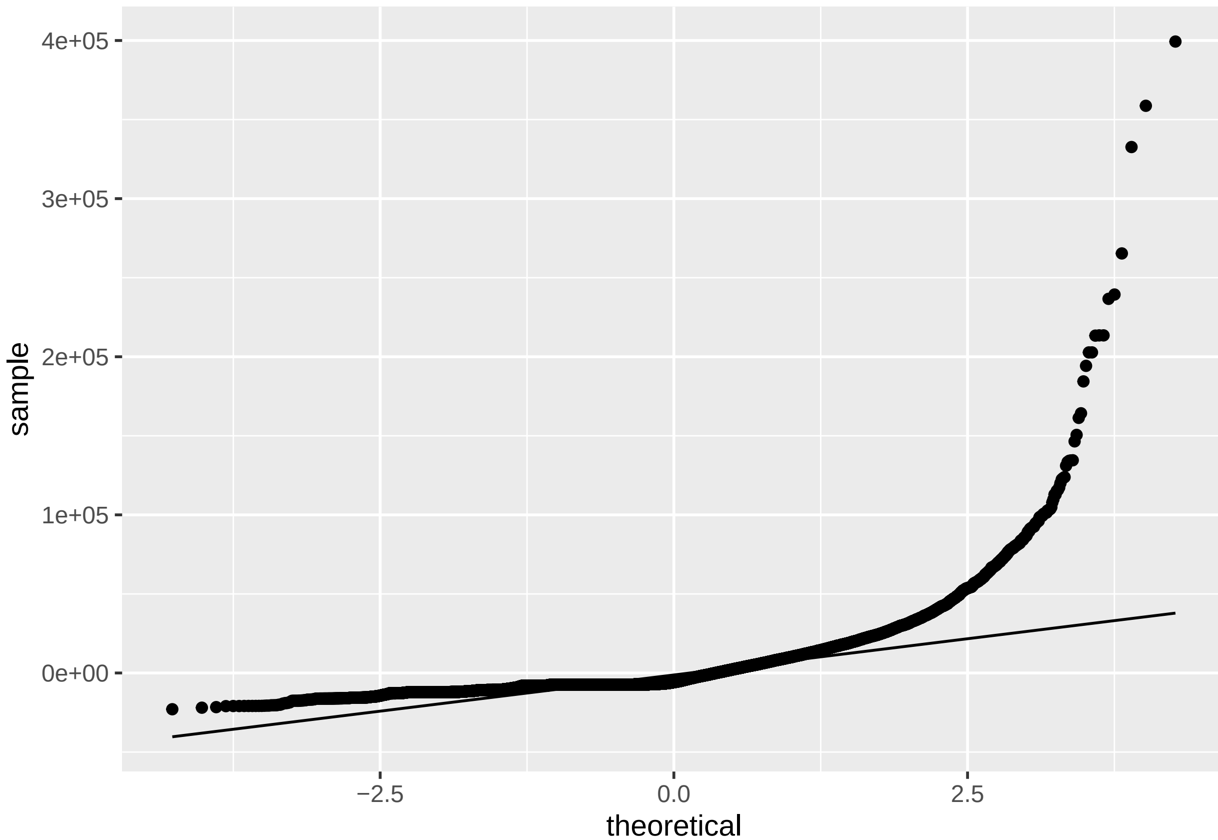
```
## # A tibble: 10 x 5
##    term          estimate std.error statistic   p.value
##    <chr>            <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)      7431.      68.7    108.   0.
##  2 covid            6394.     538.      11.9  1.79e- 32
##  3 climateChange   -5536.     650.      -8.51 1.72e- 17
##  4 abortion        10043.    2385.       4.21 2.55e-  5
##  5 blm              5425.     676.       8.03 1.00e- 15
##  6 guns             1916.     602.       3.18 1.46e-  3
##  7 news             3380.     183.      18.4  1.15e- 75
##  8 usa              4751.     164.      28.9  5.56e-182
##  9 russia           8728.     459.      19.0  2.25e- 80
## 10 immigration       668.     188.       3.56 3.68e-  4
```

```
aug <- augment(topic_retweets)

ggplot(data=aug, aes(x=.resid)) + geom_histogram()
```

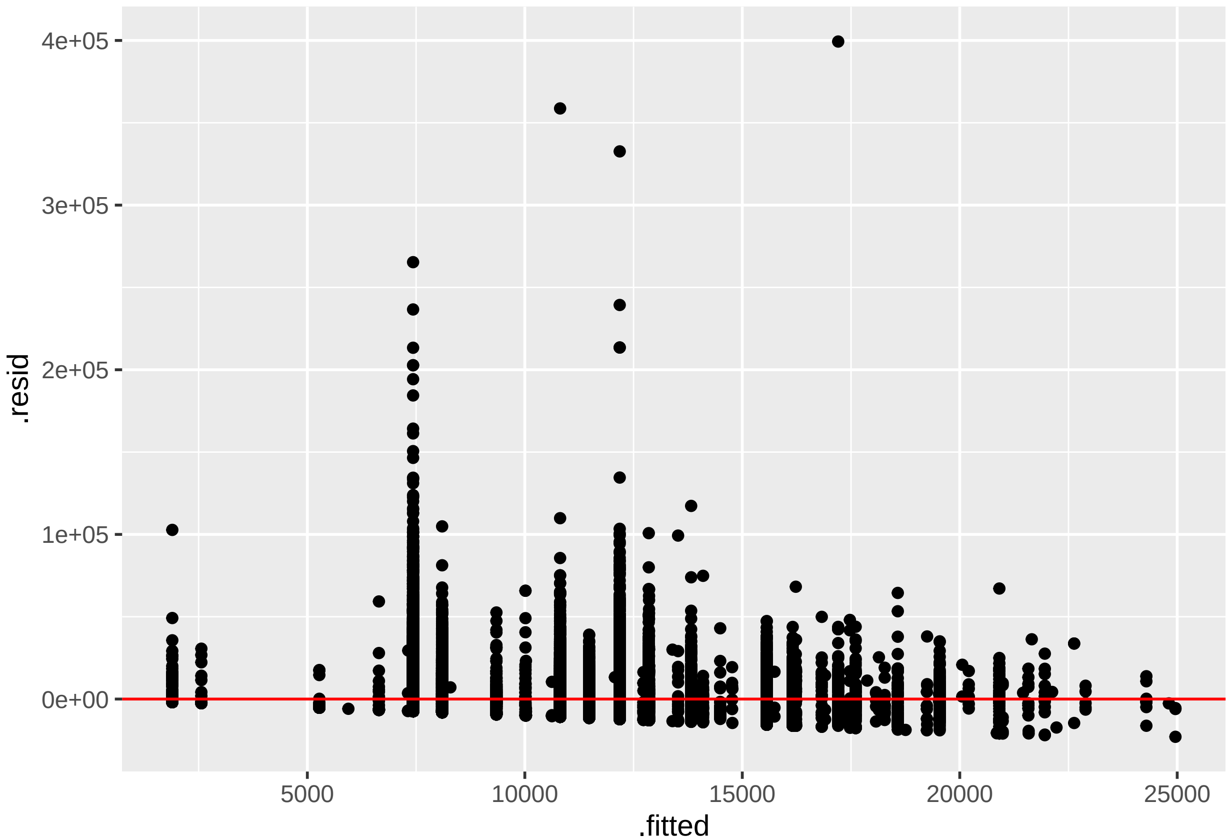
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```r
ggplot(aug, mapping = aes(sample = .resid)) +
  stat_qq() + stat_qq_line()
```




```
ggplot(data=aug, aes(x=.fitted, y=.resid)) + geom_point()+ geom_hline(yintercept=0, col="red")
```




```
#fix this, correlation between topics?
trees %>% summarize(correlation = cor(Height, Girth))
```

```
##   correlation
## 1   0.5192801
```

Sources

http://www.trumptwitterarchive.com/about

https://www.journalism.org/2016/05/26/news-use-across-social-media-platforms- 2016/

https://www.realclearpolitics.com/articles/2019/09/11/numbers_show_how_trumps_tweets_drive_the_news_cycle_141217.html

https://en.wikipedia.org/wiki/List_of_most-followed_Twitter_accounts

https://twitter.com/realDonaldTrump?ref_src=twsrc%5Egoogle%7Ctwcamp%5Eserp%7 Ctwgr%5Eauthor