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 = 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//RtmpUaioQ1/downloaded_packages
```

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

```
##
## The downloaded binary packages are in
##  /var/folders/vv/dyymk5b54fx_05m5pb2sy5jh0000gn/T//RtmpUaioQ1/downloaded_packages
```

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

```
##
## The downloaded binary packages are in
##  /var/folders/vv/dyymk5b54fx_05m5pb2sy5jh0000gn/T//RtmpUaioQ1/downloaded_packages
```

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

```
##
## The downloaded binary packages are in
##  /var/folders/vv/dyymk5b54fx_05m5pb2sy5jh0000gn/T//RtmpUaioQ1/downloaded_packages
```

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

```
##
## The downloaded binary packages are in
##  /var/folders/vv/dyymk5b54fx_05m5pb2sy5jh0000gn/T//RtmpUaioQ1/downloaded_packages
```

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

```
##
## The downloaded binary packages are in
##  /var/folders/vv/dyymk5b54fx_05m5pb2sy5jh0000gn/T//RtmpUaioQ1/downloaded_packages
```

```
install.packages("lubridate")
```

```
##
##   There is a binary version available but the source version is later:
##           binary  source needs_compilation
## lubridate  1.7.9 1.7.9.2              TRUE
```

```
## installing the source package 'lubridate'
```

```
library(viridis)
```

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

```
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()
```

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

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

```
library(sentimentr)
library(lubridate)
```

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

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

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.

Methodology

Our first step in beginning to analyze the relationships between subject, sentiment, and reception of President Donald Trump's tweets was to manipulate the data set, creating some new variables that we could use for analysis. We started by creating a set of identifier variables for different people that his tweets might be about. These variables were created using a mutate command to set them as either 1 or 0, 1 if the person was mentioned in a tweet, and 0 if they weren't. We created these variables for a total of 11 relevant political figures: Barack Obama, Joe Biden, Hillary Clinton, Alexandria Ocasio-Cortez, Nancy Pelosi, Kamala Harris, Mike Pence, Mitch McConnell, Amy Coney Barrett, and Nikki Haley. For each of the person identifier variables, we searched for people's full names, their Twitter handles, and commonly-used nicknames to determine whether or not a particular tweet was about them. We also repeated the same process to create topical identifier variables for a range of subjects that are prevalent in the nation's political discourse. The topics were as follows: COVID-19, climate change, abortion, the Black Lives Matter movement, guns, news, immigration, Russia, and the United States. For each topic we searched for the name of the thing outright (such as "BLM" or "climate change"), as well as words and phrases that are commonly used in conjunction with these topics. For example, tweets that mentioned ICE or the term "border wall" were categorized under immigration, and tweets about CNN, FOX News and "media" all went under the "news" topic. Next, using the pivot command in r, we created a variable for "person," a categorical variable that told us which of the identifier variables for people were equal to "1" for each tweet. The output was a variable that had a value of either the name of one of the 11 people we searched for or "other." From the new "person" variable, we created two new variables called party and gender. The party variable separated the 11 people we were looking at into either democrats or republicans and assigned tweets about them to the appropriate party

category. We did the same for the gender of each of the people giving tweets about these figures a value of either "male" or "female." Next, we used the package sentimentr to identify the sentiment of each of Trump's tweets. The package examines each tweet, identifying positive and negative words based on a lexicon made by the package's creator Tyler Rinker. It then aggregates a sentiment score for the tweet, assigning a value of -1 to words it deems negative, 1 for positive words and 0 for neutral words. This results in an overall sentiment score. For each tweet, we created a new variable called ave_sentiment that contained the raw, numeric sentiment score of the text. We then used the mutate command to create a new variable called posNeg that grouped sentiment scores into three categories: positive, neutral, or negative. Finally, we used the separate command in r to break up the variable "created_at" into two separate variables for date and time. After this, each observation had a date variable in the mm/dd/yy format and a time variable in the 24-hour time format. After these manipulations, the data set contained the following variables:

- source: the version of Twitter used to share the tweet. For example "Twitter for iPhone."
- text: the full text of the tweet.
- created_at: gives the date (in mm/dd/yy format) and time (in 24-hour format) at which the tweet was published.
- retweet_count: number of retweets that the tweet received.
- favorite_count: number of favorites that the tweet received.
- is_retweet: assigns a value of either true or false for whether the tweet is originally written by Trump or is a retweet of someone else.
- id_str: The scrape.py script collects tweet ids. If you know a tweet's id you can get all the information available about that tweet using Tweepy - text, timestamp, number of retweets / replies / favorites, geolocation, etc.
- obama: 1 or 0 for whether the tweet talks about Barack Obama.
- biden: 1 or 0 for whether the tweet talks about Joe Biden.
- pelosi: 1 or 0 for whether the tweet talks about Nancy Pelosi.
- kamala: 1 or 0 for whether the tweet talks about Kamala Harris.
- hillary: 1 or 0 for whether the tweet talks about Hillary Clinton. -aoc: 1 or 0 for whether the tweet talks about Alexandria Ocasio-Cortez.
- pence: 1 or 0 for whether the tweet talks about Mike Pence.
- mcconnell: 1 or 0 for whether the tweet talks about Mitch McConnell.
- fauci: 1 or 0 for whether the tweet talks about Dr. Anthony Fauci.
- amy: 1 or 0 for whether the tweet talks about Amy Coney Barrett.
- nikki: 1 or 0 for whether the tweet talks about Nikki Haley.
- covid: 1 or 0 for whether the tweet talks about COVID-19.
- climateChange: 1 or 0 for whether the tweet talks about climate change.
- abortion: 1 or 0 for whether the tweet talks about abortion.
- blm: 1 or 0 for whether the tweet talks about the Black Lives Matter movement.
- guns: 1 or 0 for whether the tweet talks about guns.
- news: 1 or 0 for whether the tweet talks about news media.
- usa: 1 or 0 for whether the tweet talks about the United States.
- russia: 1 or 0 for whether the tweet talks about Russia.
- immigration: 1 or 0 for whether the tweet talks about immigration.
- person: categorical variable for the name of the person that the tweet is about (from among the 11 looked at in this analysis).
- party: political party of the person who is talked about in the tweet (either democrat or republican).
- gender: gender of the person who is talked about in the tweet (either male or female).
- ave_sentiment: numeric sentiment score of the tweet.
- posNeg: categorical sentiment of the tweet (either positive, neutral or negative).
- ate: date that the tweet was posted (mm/dd/yy).
- time: time that the tweet was posted (24-hour format).

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))


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))
```

```
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 %>%
   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)|
                            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)
```

```
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.
```

```
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.

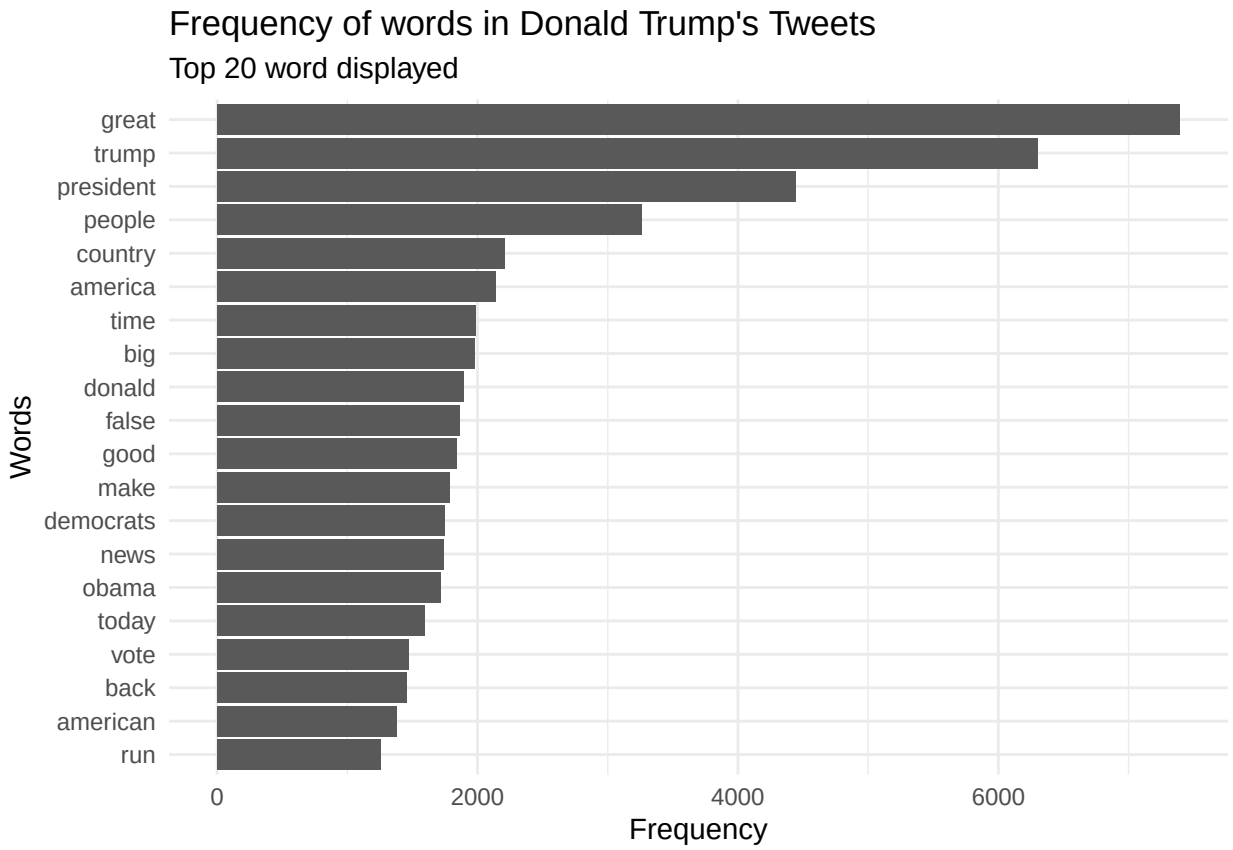


Figure 2: Frequency of Donald Trump's top 20 words that are used in his Tweets. Just like in Figure 1, some of the words that were not part of the actual text of the Tweet or were artifacts of Tweets (such as URLs, account names, and years) were removed to clean the data.

```
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") +
   theme(axis.text.x = element_text(angle = 45, hjust = 1))
```

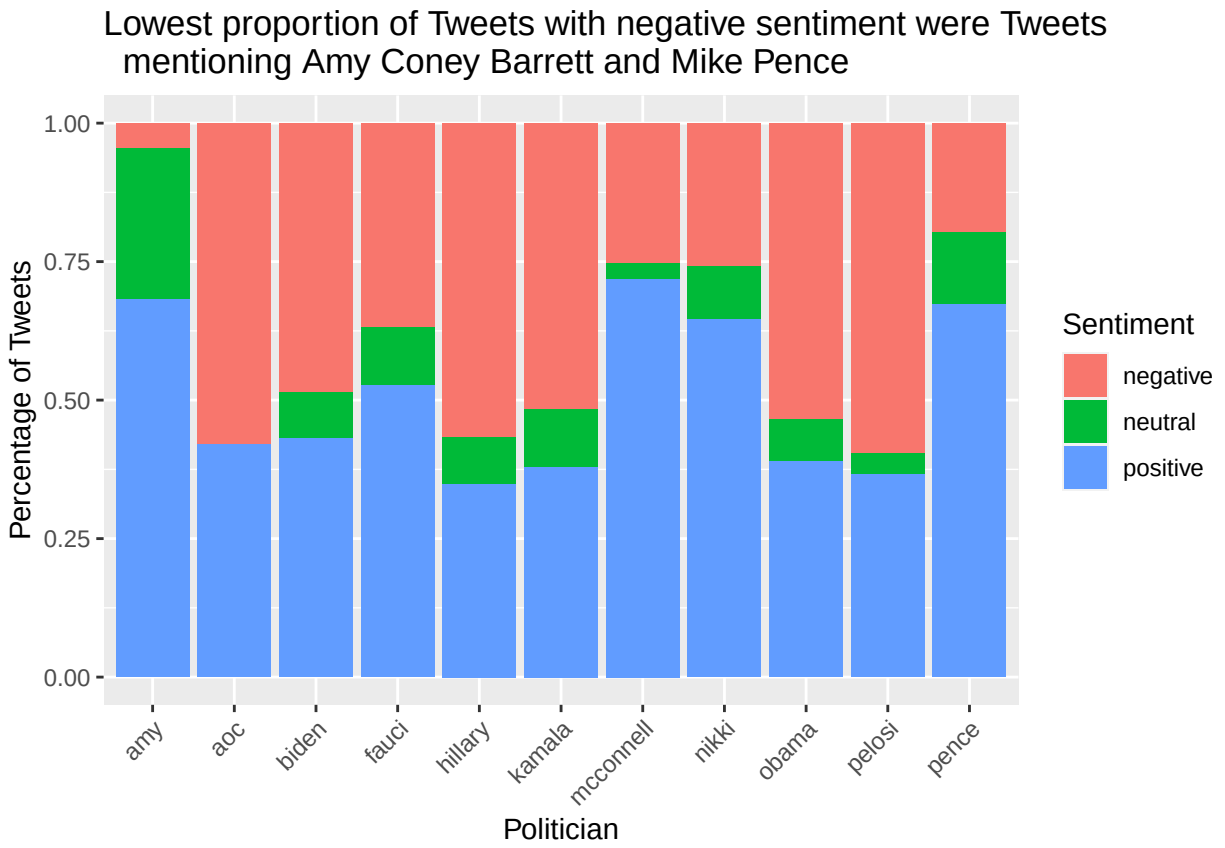


Figure 3: Illustrates the proportion of Tweets with average negative (<0), neutral (=0), and positive (>0) sentiment that mention a specific politician.

```
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"))
```

```
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") +
   geom_hline(yintercept=0, linetype="dashed", color = "red")
```

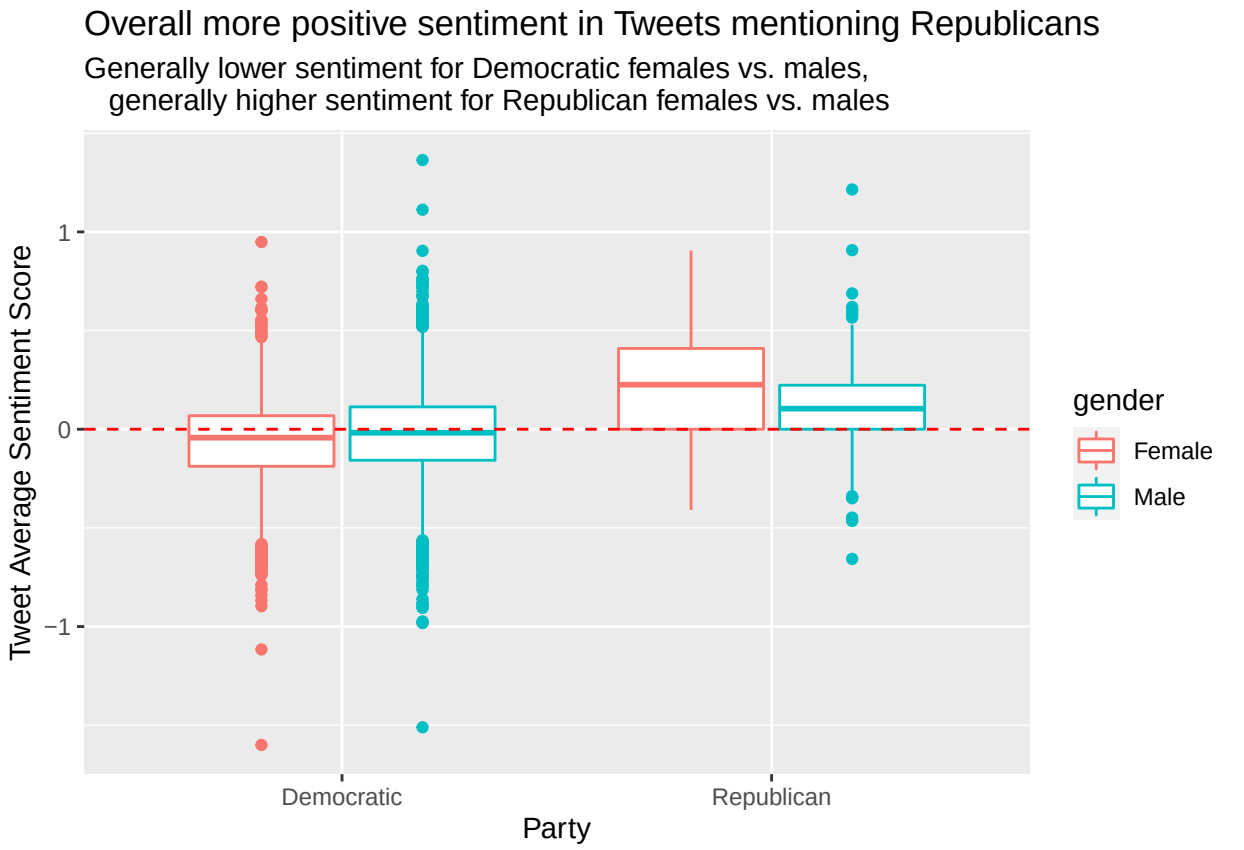


Figure 4: Boxplot showing the average sentiment scores for Tweets, divided by political party of the person mentioned in the Tweet, and split within party to show any differences in how he talks about politicians of different genders within either the Democratic or Republican party.

```
dems <- peopleData %>%
  filter(party=="Democratic")

repubs <- peopleData %>%
  filter(party=="Republican")

t.test(dems$ave_sentiment,
       repubs$ave_sentiment,
       mu = 0,
       var.equal = FALSE,
       alternative = "less",
       conf.level = 0.95)
```

```
##
```

```
##  Welch Two Sample t-test
##
## data:  dems$ave_sentiment and repubs$ave_sentiment
## t = -14.678, df = 469.92, p-value < 2.2e-16
## alternative hypothesis: true difference in means is less than 0
## 95 percent confidence interval:
##        -Inf -0.1414063
## sample estimates:
##   mean of x   mean of y
## -0.03922378  0.12006845
```

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

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

```
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 %>%
   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)
```

```
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") +
   geom_hline(yintercept=0, linetype="dashed", color = "red")
```

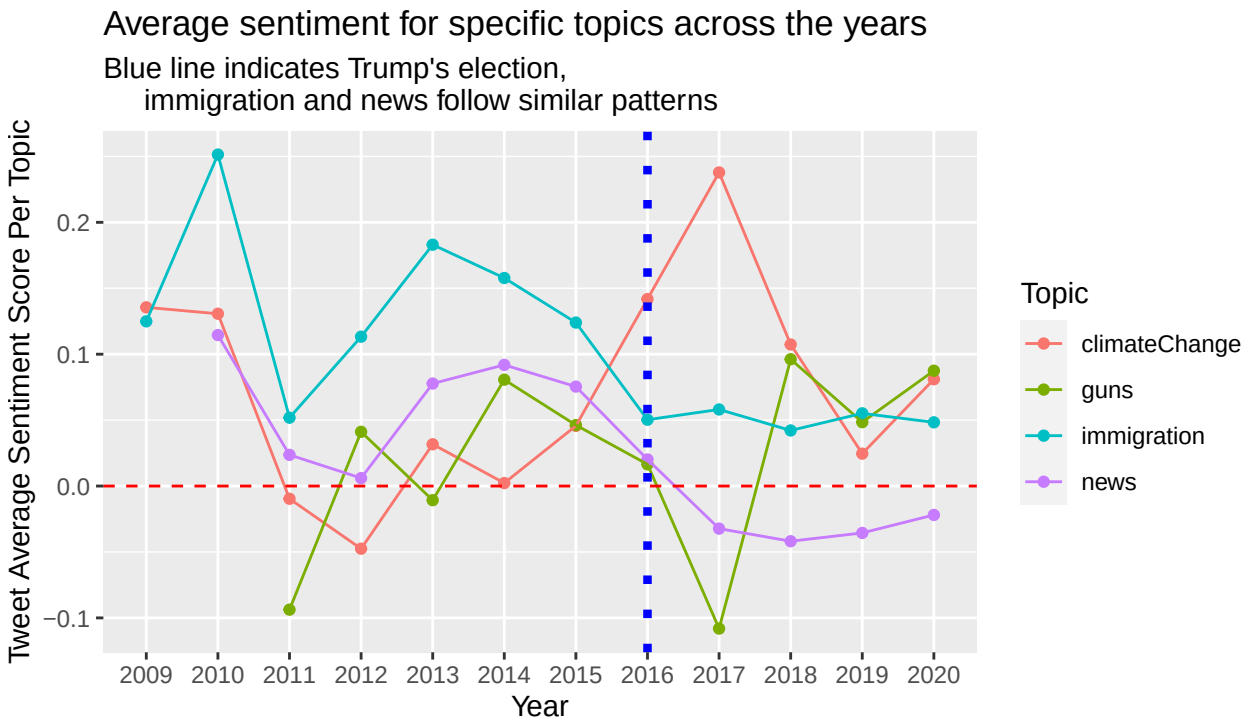


Figure 5: Shows change of average sentiment of Tweets referring to a specific topic across the years. Only showing a few topics, we selected the ones that have been Tweeted about for a long time. The vertical blue line indicates when Trump first became elected.

```
yearVar <- tweetsWithSentimentDateTimeFormat  %>%
   mutate(year = year(date))

#CLIMATE CHANGE
pre2016CC <- yearVar %>%
  filter(year==2016 & topic=="climateChange")

post2016CC <- yearVar %>%
  filter(year>=2016 & topic=="climateChange")

t.test(pre2016CC$ave_sentiment,
       post2016CC$ave_sentiment,
       mu = 0,
       var.equal = FALSE,
       alternative = "two.sided",
       conf.level = 0.95)
```

```
##
##  Welch Two Sample t-test
##
## data:  pre2016CC$ave_sentiment and post2016CC$ave_sentiment
## t = 0.72595, df = 26.822, p-value = 0.4742
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.06757291  0.14153277
## sample estimates:
## mean of x mean of y
## 0.1417982 0.1048183
```

```r
#GUNS
pre2016guns <- yearVar %>%
  filter(year==2016 & topic=="guns")

post2016guns <- yearVar %>%
  filter(year>=2016 & topic=="guns")

t.test(pre2016guns$ave_sentiment,
       post2016guns$ave_sentiment,
       mu = 0,
       var.equal = FALSE,
       alternative = "two.sided",
       conf.level = 0.95)
```

```
##
##  Welch Two Sample t-test
##
## data:  pre2016guns$ave_sentiment and post2016guns$ave_sentiment
## t = -1.157, df = 27.246, p-value = 0.2573
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1423921  0.0396828
## sample estimates:
##  mean of x  mean of y
## 0.01647441 0.06782909
```

```r
#IMMIGRATION
pre2016imm <- yearVar %>%
  filter(year==2016 & topic=="immigration")

post2016imm <- yearVar %>%
  filter(year>=2016 & topic=="immigration")

t.test(pre2016imm$ave_sentiment,
       post2016imm$ave_sentiment,
       mu = 0,
       var.equal = FALSE,
       alternative = "two.sided",
       conf.level = 0.95)
```

```
##
##  Welch Two Sample t-test
##
## data:  pre2016imm$ave_sentiment and post2016imm$ave_sentiment
## t = 0.021915, df = 297.88, p-value = 0.9825
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.03226623  0.03299297
## sample estimates:
##  mean of x  mean of y
## 0.05030422 0.04994085
```

```r
#NEWS
pre2016News <- yearVar %>%
  filter(year==2016 & topic=="news")
```

```r
post2016News <- yearVar %>%
  filter(year>=2016 & topic=="news")

t.test(pre2016News$ave_sentiment,
       post2016News$ave_sentiment,
       mu = 0,
       var.equal = FALSE,
       alternative = "two.sided",
       conf.level = 0.95)
```

```
##
##  Welch Two Sample t-test
##
## data:  pre2016News$ave_sentiment and post2016News$ave_sentiment
## t = 3.3319, df = 559.89, p-value = 0.0009195
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  0.01832116 0.07094526
## sample estimates:
##   mean of x   mean of y
##  0.02015388 -0.02447932
```

```r
polarization_tweets <- tweetsWithSentiment %>%
   mutate(polarization = abs(ave_sentiment)) %>%
   mutate(retweet_size = case_when(retweet_count <= 20000 ~ 1,
      retweet_count > 20000 & retweet_count <= 40000 ~ 2,
      retweet_count > 40000 & retweet_count <= 60000 ~ 3,
      retweet_count > 60000 & retweet_count <= 80000 ~ 4,
      retweet_count > 80000 & retweet_count <= 100000 ~ 5,
      retweet_count > 100000 ~ 6))

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

```
##
##  Pearson's Chi-squared test
##
## data:  table(polarization_tweets$posNeg, polarization_tweets$retweet_size)
## X-squared = 663.34, df = 10, p-value < 2.2e-16
```

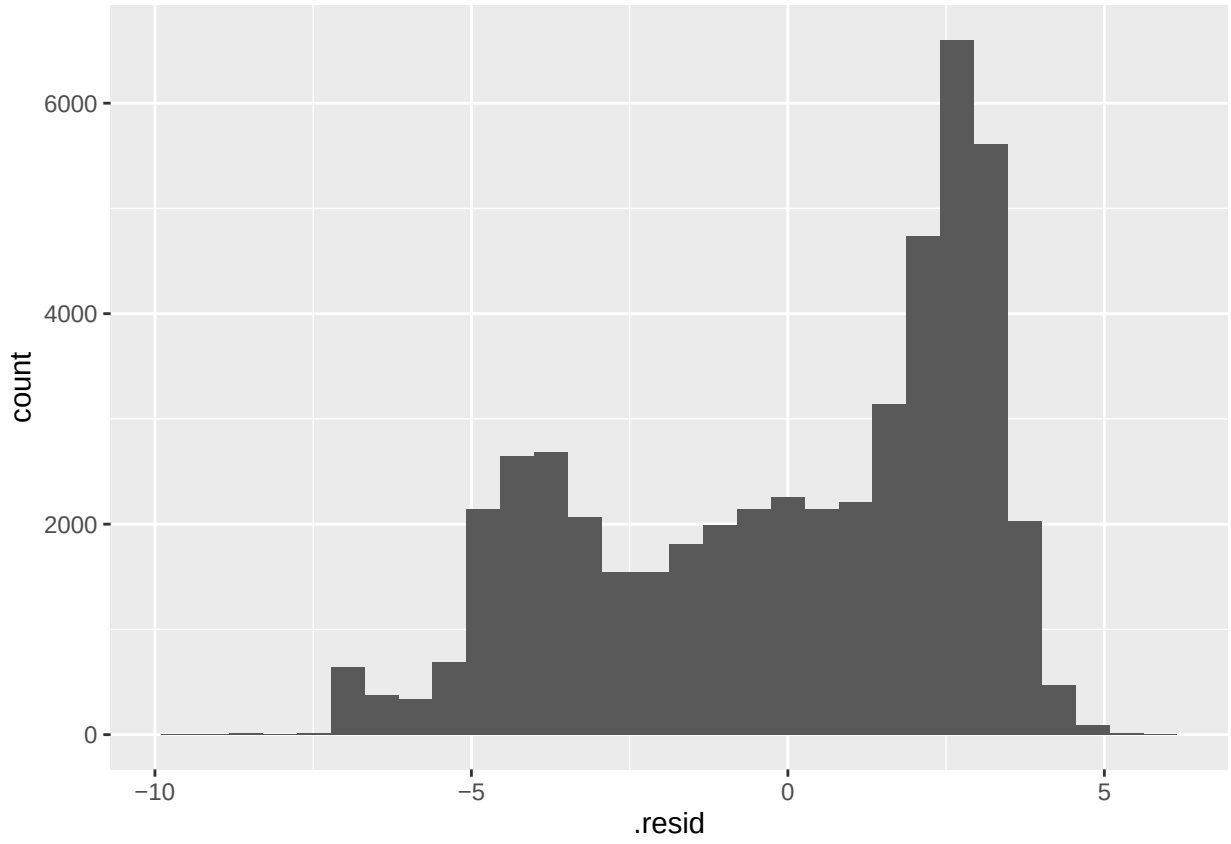
```r
tweetsWithSentimentWithoutLink <- tweetsWithSentimentWithoutLink%>%
   filter(retweet_count!=0 & !is.na(retweet_count)) %>%
   mutate(log_retweet_count = log(retweet_count))

person_retweets <- lm(log_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)    6.86     0.0141    487.   0.
##  2 obama         -0.599    0.0557    -10.8  6.16e- 27
```

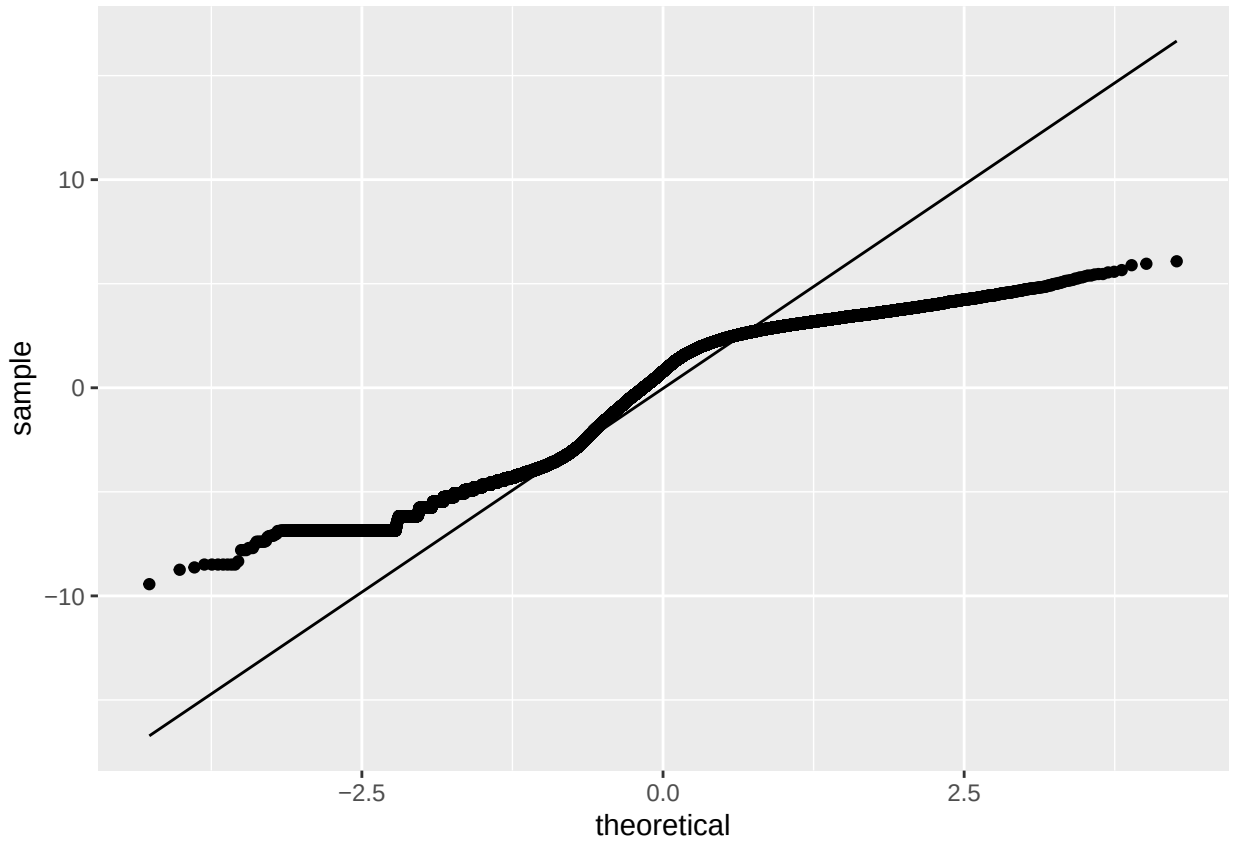
```
##  3 biden          1.63     0.0875     18.7  2.21e- 77
##  4 pelosi         2.58     0.131      19.6  2.93e- 85
##  5 kamala         1.77     0.550       3.22 1.27e-  3
##  6 hillary        1.94     0.0829     23.4  2.86e-120
##  7 aoc            1.89     0.532       3.56 3.70e-  4
##  8 pence          1.77     0.180       9.86 6.45e- 23
##  9 nikki         -0.734    0.569      -1.29 1.97e-  1
## 10 mcconnell      0.827    0.299       2.77 5.68e-  3
## 11 amy            1.76     0.646       2.72 6.50e-  3
## 12 fauci          2.81     0.679       4.14 3.45e-  5
```

```
aug <- augment(person_retweets)

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

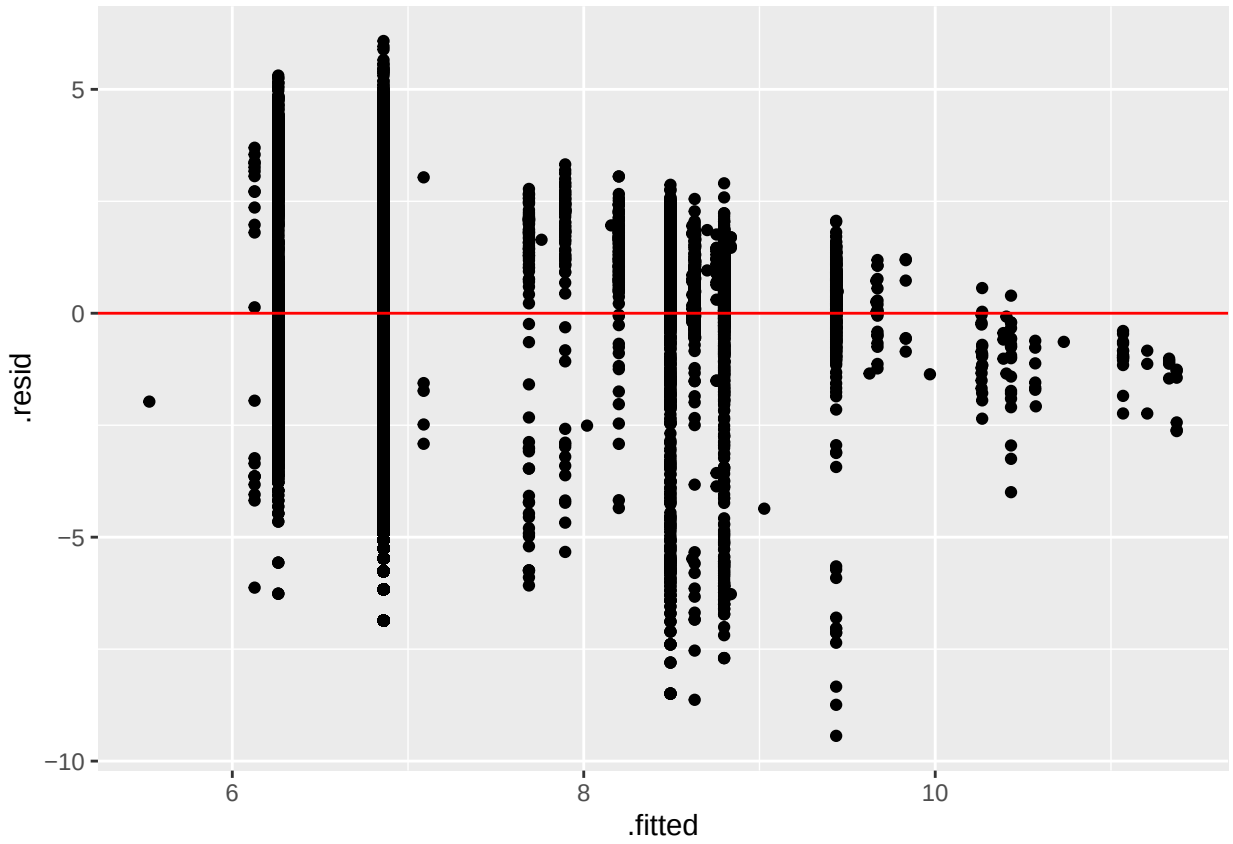
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
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))
```

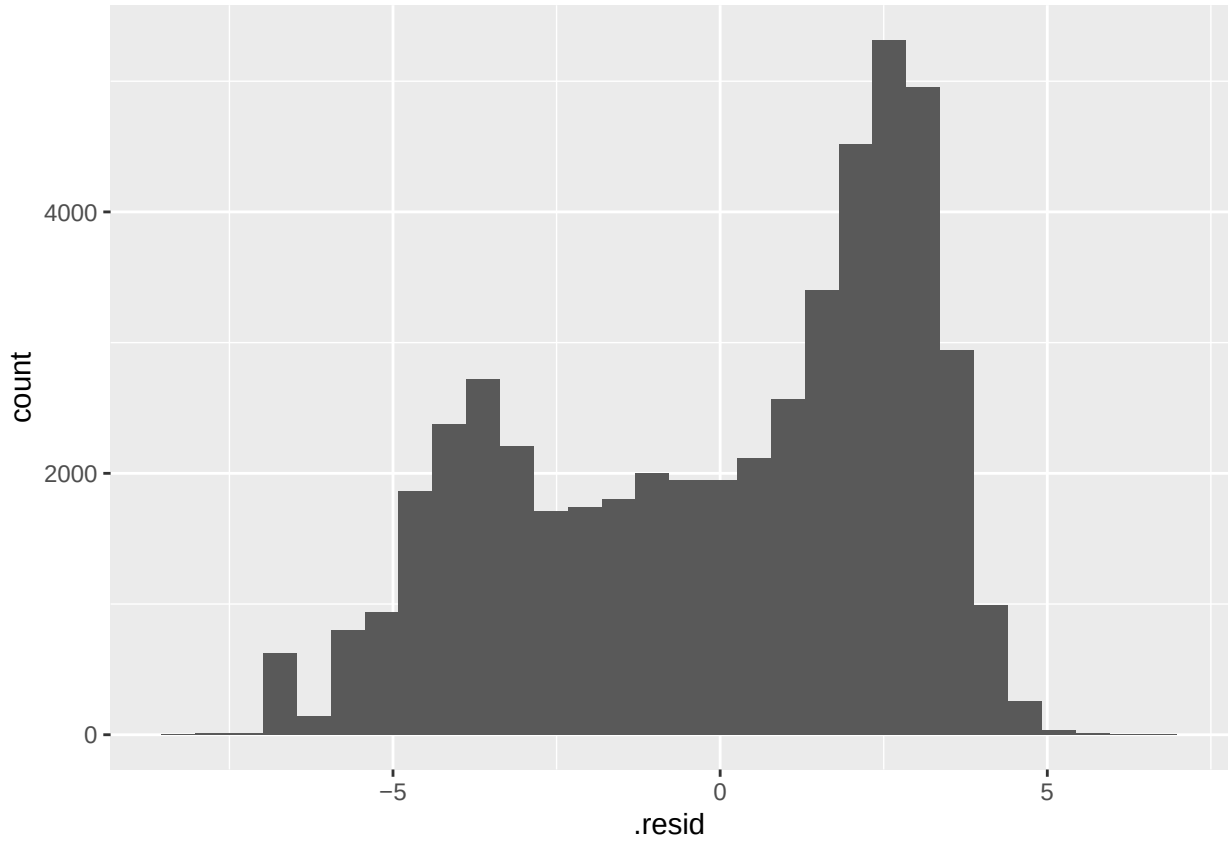
```
##   correlation
## 1   0.5192801
```

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

tidy(topic_retweets)
```

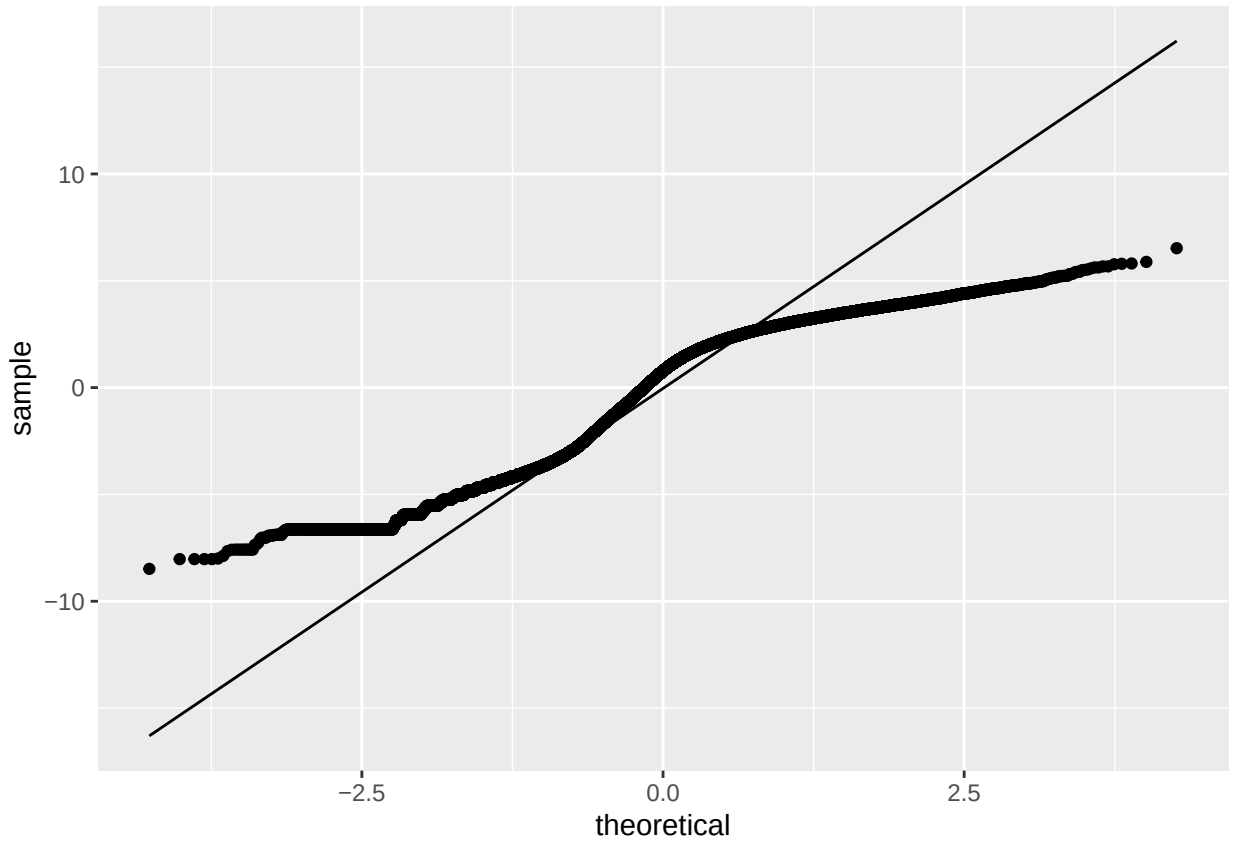
```
## # A tibble: 10 x 5
##    term          estimate std.error statistic   p.value
##    <chr>            <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)      6.63     0.0159    418.   0.
##  2 covid            2.15     0.123      17.5  2.83e- 68
##  3 climateChange   -1.60     0.149     -10.8  5.80e- 27
##  4 abortion         2.31     0.542       4.27 2.00e-  5
##  5 blm              1.02     0.154       6.61 3.76e- 11
##  6 guns             1.08     0.137       7.88 3.32e- 15
##  7 news             0.950    0.0418     22.7  1.06e-113
##  8 usa              1.39     0.0375     37.2  2.51e-298
##  9 russia           2.06     0.104      19.7  2.25e- 86
## 10 immigration     -0.419    0.0430     -9.76 1.79e- 22
```

```
aug <- augment(topic_retweets)

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

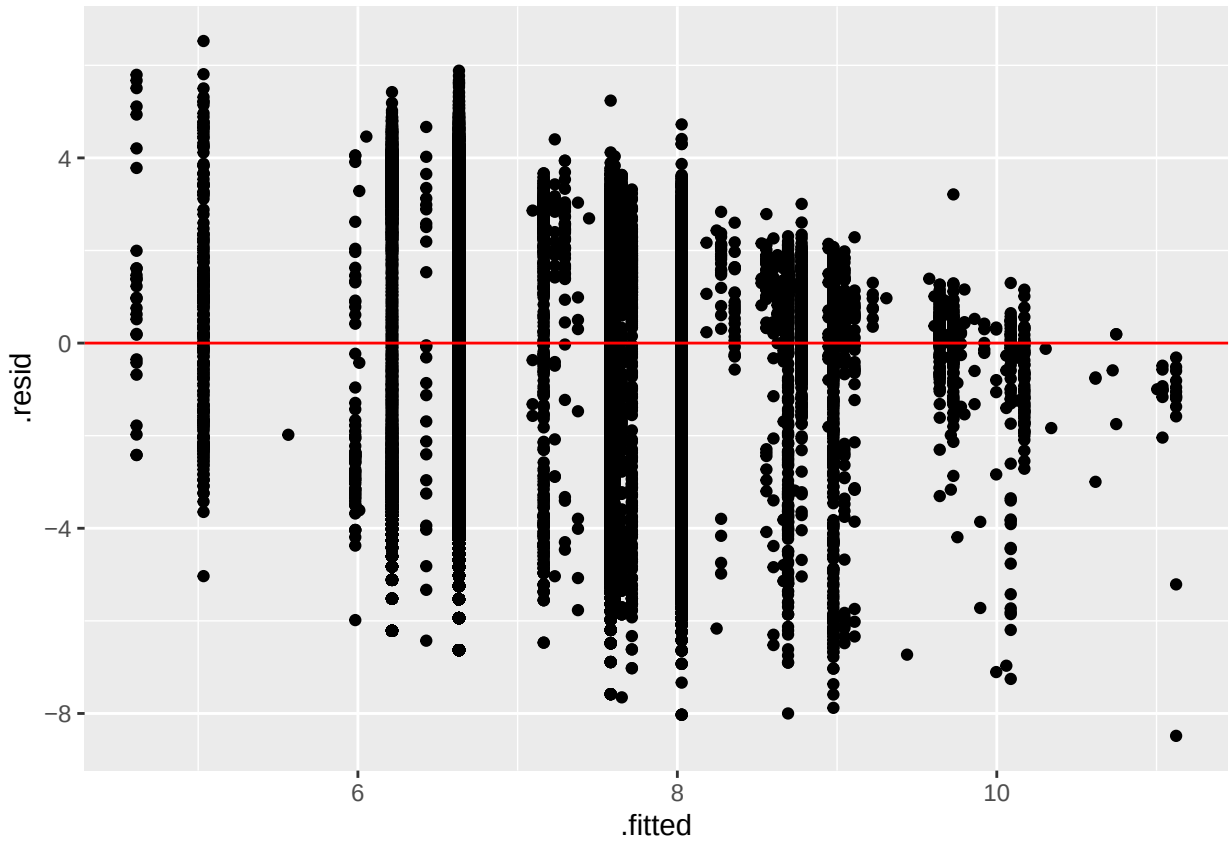
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
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))
```

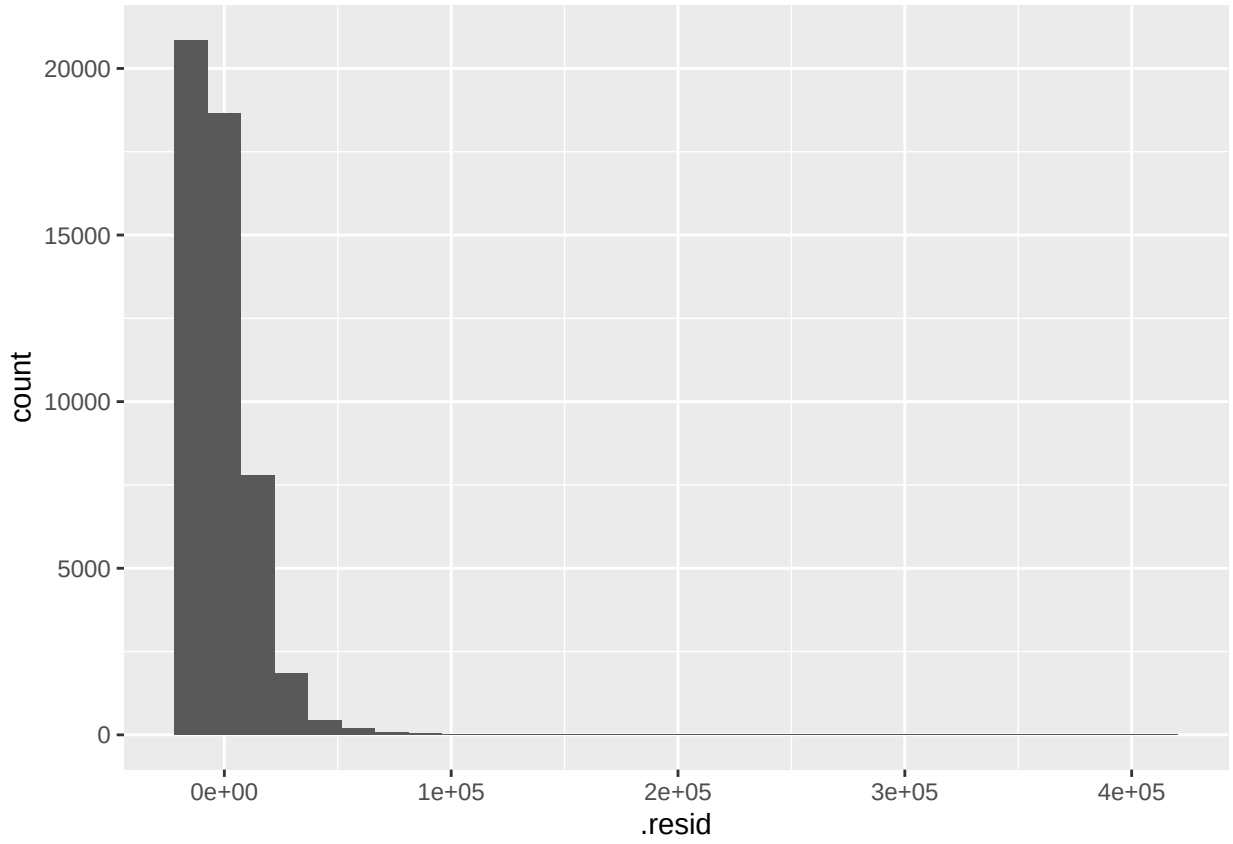
```
##   correlation
## 1   0.5192801
```

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

tidy(person_retweets)
```

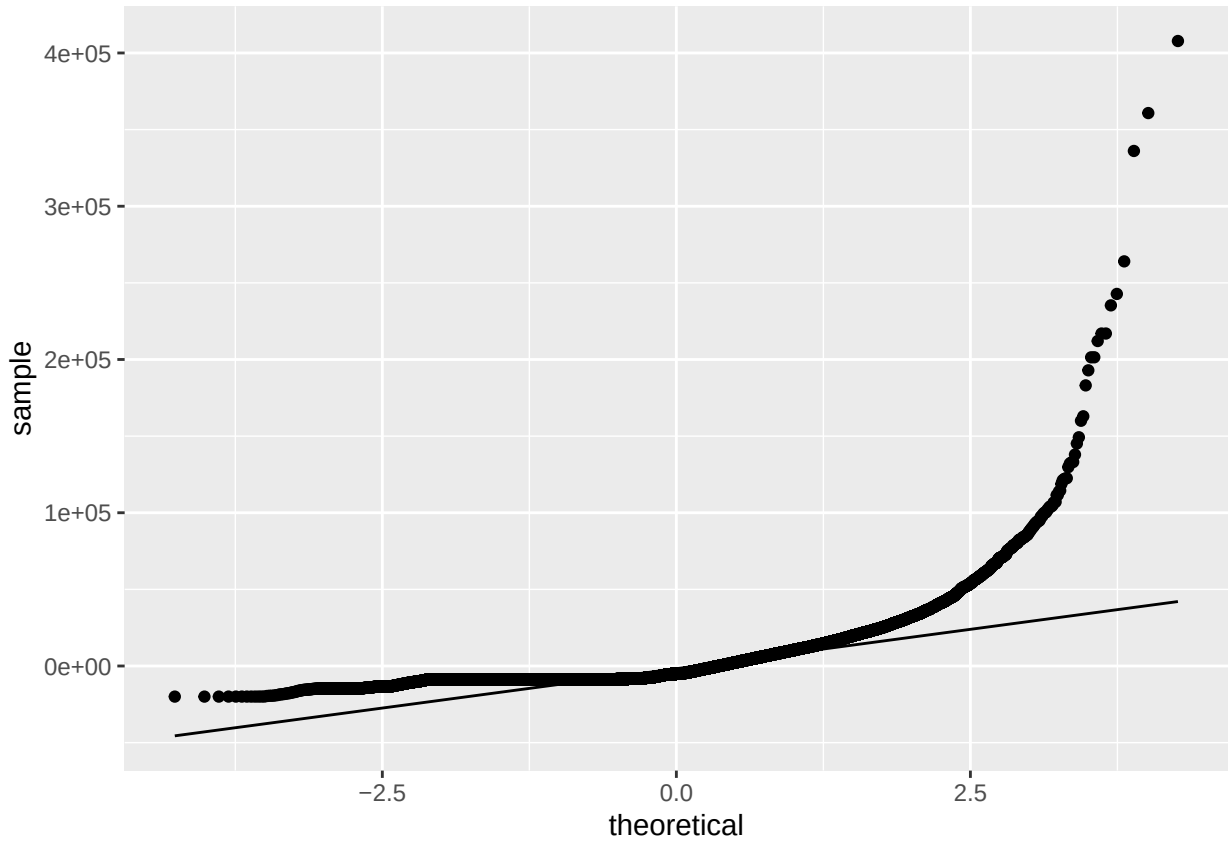
```
## # A tibble: 12 x 5
##    term        estimate std.error statistic  p.value
##    <chr>          <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)    8748.      62.0   141.    0.
##  2 obama         -3373.     245.    -13.8   5.28e-43
##  3 biden          5977.     385.     15.5   3.24e-54
##  4 pelosi        11215.     578.     19.4   1.37e-83
##  5 kamala         5030.    2421.      2.08  3.78e- 2
##  6 hillary        4840.     365.     13.3   3.80e-40
##  7 aoc            7608.    2341.      3.25  1.15e- 3
##  8 pence          1092.     790.      1.38  1.67e- 1
##  9 nikki         -2815.    2505.     -1.12  2.61e- 1
## 10 mcconnell      1612.    1316.      1.23  2.21e- 1
## 11 amy            2143.    2840.      0.755 4.51e- 1
## 12 fauci         11340.    2985.      3.80  1.46e- 4
```

```
aug <- augment(person_retweets)

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

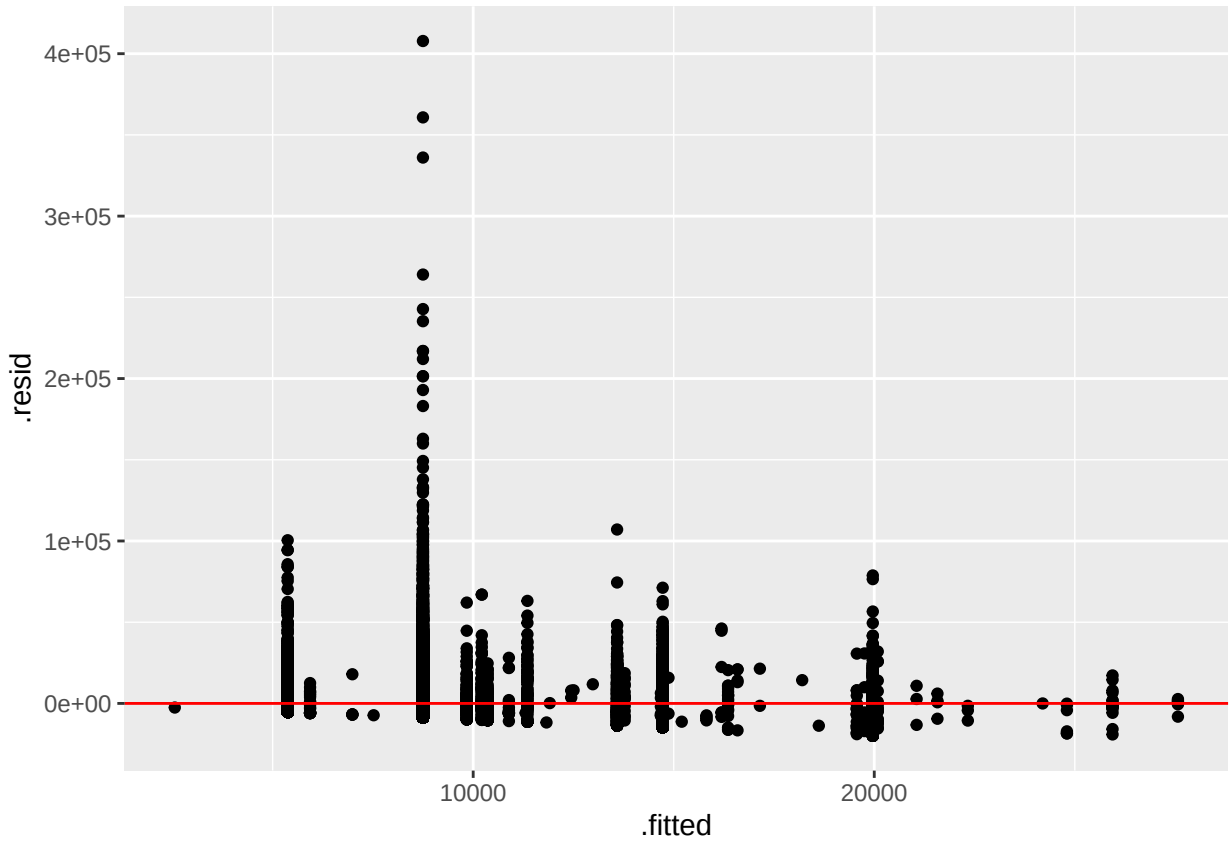
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
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))
```

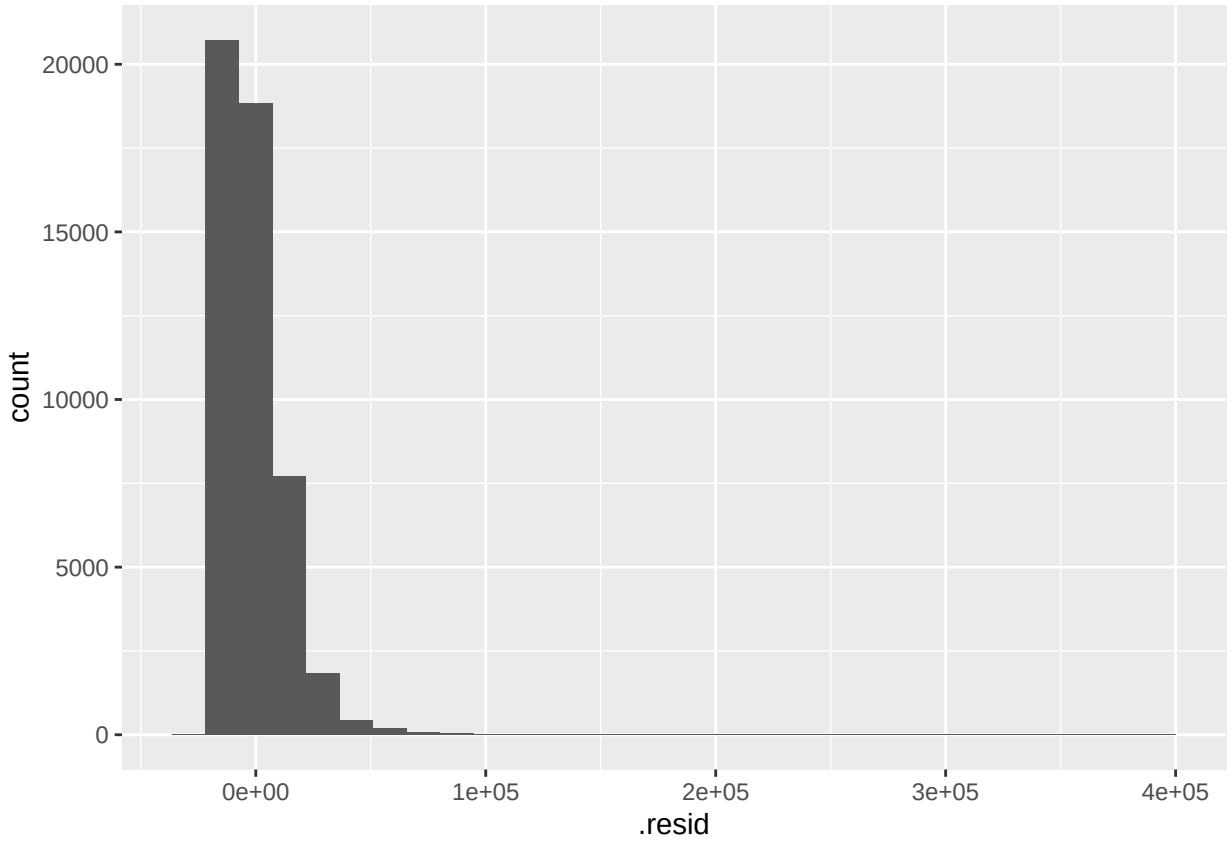
```
##   correlation
## 1   0.5192801
```

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

tidy(topic_retweets)
```

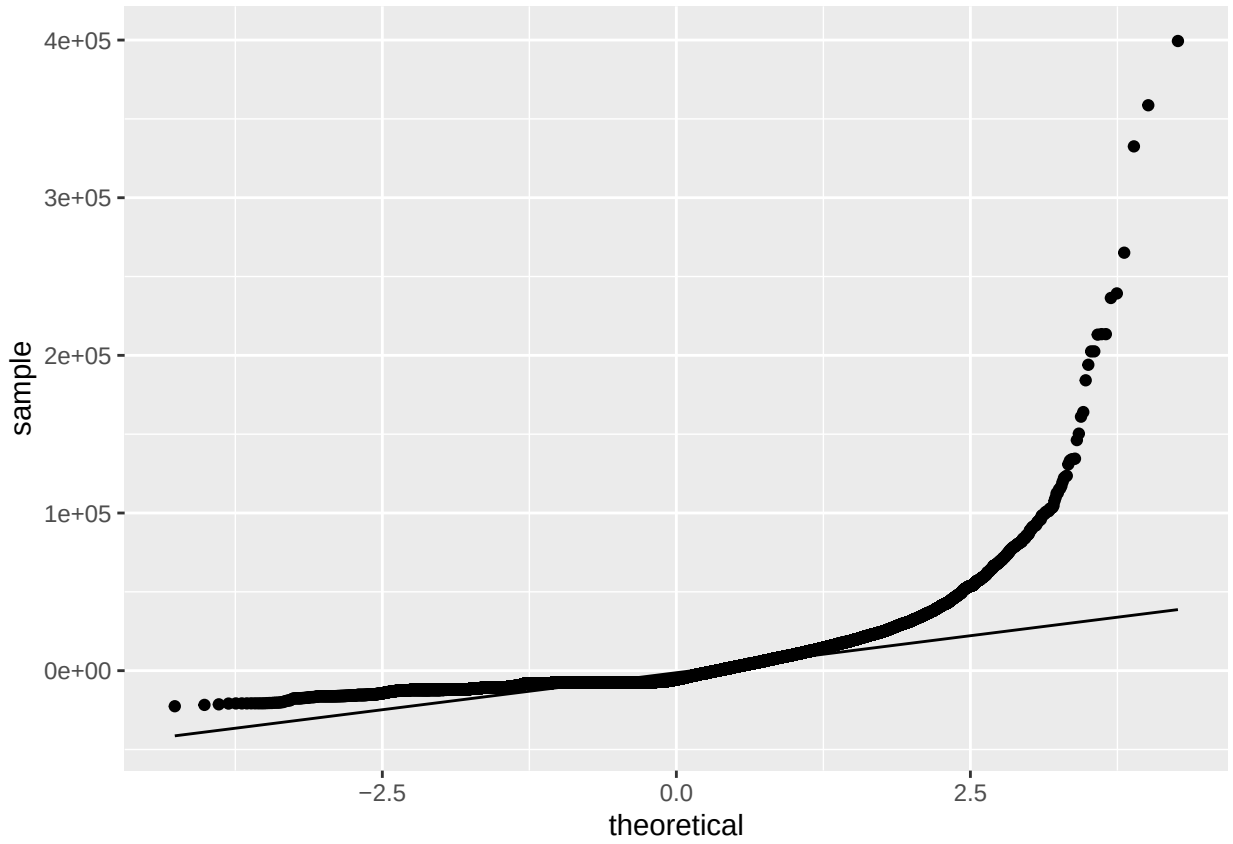
```
## # A tibble: 10 x 5
##    term          estimate std.error statistic   p.value
##    <chr>            <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)      7659.      70.2    109.   0.
##  2 covid            6278.     543.      11.6  7.58e- 31
##  3 climateChange   -5683.     658.      -8.63 6.14e- 18
##  4 abortion         9911.    2402.       4.13 3.70e-  5
##  5 blm              5351.     682.       7.84 4.53e- 15
##  6 guns             1793.     608.       2.95 3.21e-  3
##  7 news             3217.     185.      17.4  2.03e- 67
##  8 usa              4577.     166.      27.6  7.42e-166
##  9 russia           8555.     462.      18.5  3.30e- 76
## 10 immigration       595.     190.       3.13 1.78e-  3
```

```
aug <- augment(topic_retweets)

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

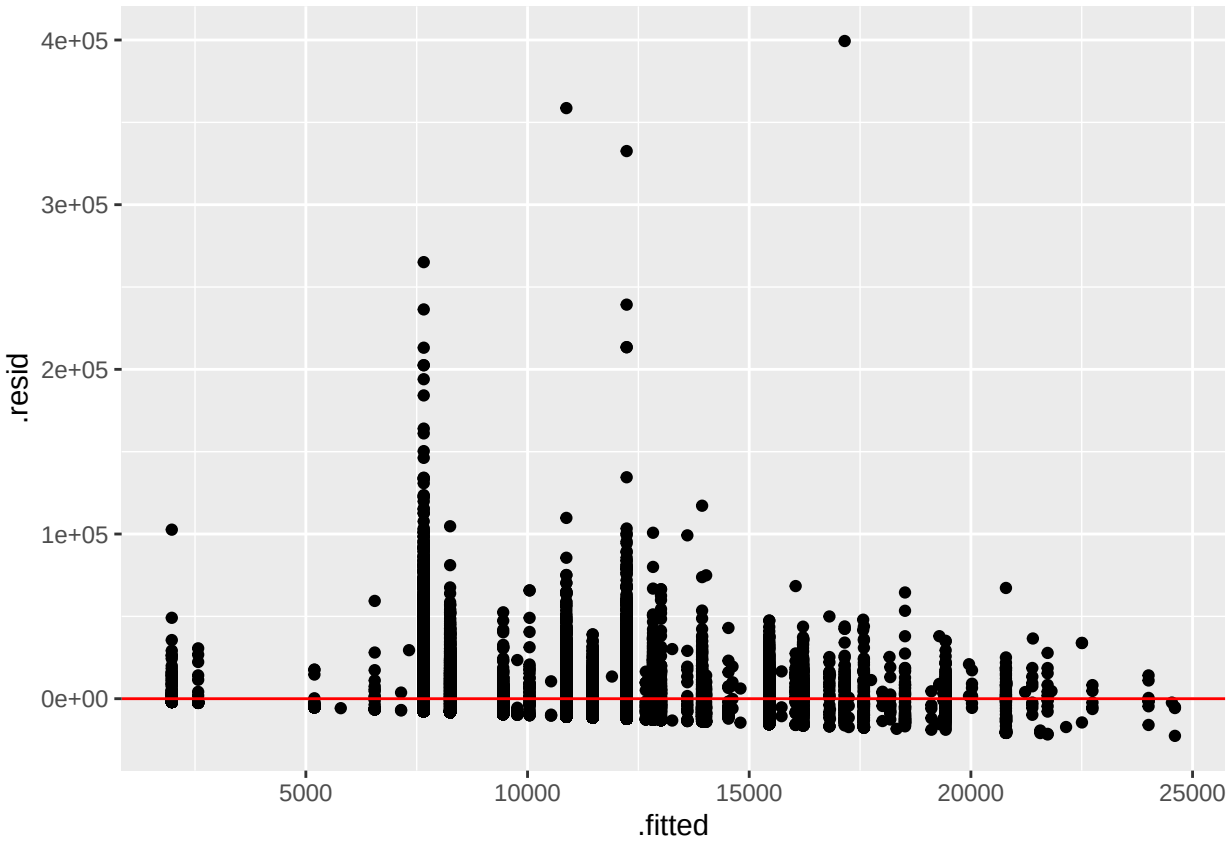
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
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
```

Results

(Figure 1) The purpose of this visualization is to analyze what words are most commonly found in Donald Trump's tweet. Using a word cloud, the most common words are displayed with the most frequent words being displayed larger. Given the results, we can see that some of the words he most commonly uses are Trump, president, people, country, america, and time. While these words don't necessarily indicate the full scope of what he discusses on his Twitter account, we are able to draw the conclusion that most of his tweets are in regard to his role as the President of the United States. We believed it would be interesting to see if a common theme would develop among his most common words, which is partially indicated in our results.

(Figure 3) Figure 3 shows the relative proportions of tweets with positive, neutral and negative overall sentiment that Donald Trump has shared about each of the people covered in our dataset. By creating a bar plot, we were able to actively compare the sentiments Trump tends to have towards specific people making it easier to visualize the results. Overall, it appears that Alexadria Ocasio-Cortez, Hillary Clinton, and Nancy Pelosi recieved the highest proportion of tweets with negative sentiment. Amy Coney Barrett had the lowest proportion of negative tweets, followed by Vice President Mike Pence. On the opposite side of the spectrum, Senate Majority Leader Mitch McConnell had the highest proportion of positive tweets while Amy Coney Barrett, Mike Pence, and Nikki Haley also received a significant majority of tweets with a positive sentiment. These results were particularly interesting because it shows that Trump tends to tweet about people of the same party with positive sentiment more frequently compared to the people of the democratic party.

(Chi Square Test) The variable, polarization, is defined as how intense the sentiment of Donald Trump's tweet is. This hypothesis test will analyze if the polarization of the tweet and the amount of retweets are

dependent or independent of each other. We chose to analyze this because we believed it to indicate the public's response to his tweets. We can draw the conclusion that the public reaction is dependent on how intense of a sentiment Trump expresses if we reject the null hypothesis.

We are performing a hypothesis test using a chi-square test. We are performing this hypothesis test at the 0.05 significance levels. Here are our null and alternative hypotheses:

H_0: The polarization of the tweet and the amount of retweets are independent; there is no association between the two variables.

H_1: The polarization of the tweet and the amount of retweets are NOT independent; there is an association between the two variables

Under the assumption that the null hypothesis is true, the chi-square test statistic follows a Chi-square distribution with degrees of freedom equal to 39554.

The value of our test statistic, chi-square, is 41580.

The p-value of our chi-square test was 6.949e-13, which is less than the significance level of 0.05. This means we reject the null hypothesis. We are concluding that there is sufficient evidence to suggest that the presence of throat pain and ASA classification are NOT independent or that there is sufficient evidence to suggest that there is an association between the two variables. More specifically, this hypothesis test shows that the amount of retweets are dependent on how polarizing President Trump's tweets are.

The purpose of these visualizations is to indicate the different sentiment he displays towards male versus female politicians. We decided to divide this into two visualizations by party affiliation because we believed the results could be skewed by differing political beliefs, and since we just wanted to learn if he had a different sentiment towards strictly males or females, we figured this was the best way to isolate the analysis of the gender variable. This visualization of the Democrats shows the average sentiment score among the Democratic politicians, showing specifically the difference in sentiment between male and female politicians using a box plot. This visualization shows that when Donald Trump discusses Democratic politicians, he has a similar sentiment towards male and female, which is fairly close to "neutral". We also noticed that for this visualization, the male Democrats had a larger range of sentiments than the female Democrats. In the visualization of male and female Republicans, the females had a slightly higher average sentiment compared to their male counterparts. This could indicate that he talks about males in his party in a less positive manner as compared to the females. We also recognized that, again, the male Republicans had a larger range of sentiments than the female Republicans, showing that his sentiments are more likely to fluctuate when discussing a male in his tweets.

(Topics over time) The purpose of this visualization is to see if Donald Trump's sentiments towards specific "hot button topics" change over time. We compiled all the tweets into a line plot with time as the x-axis and average sentiment as the y-axis. By taking the average sentiment of each topic by year, we were able to create this visualization and see how his sentiments in his tweets changed when discussing the topics. ### 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