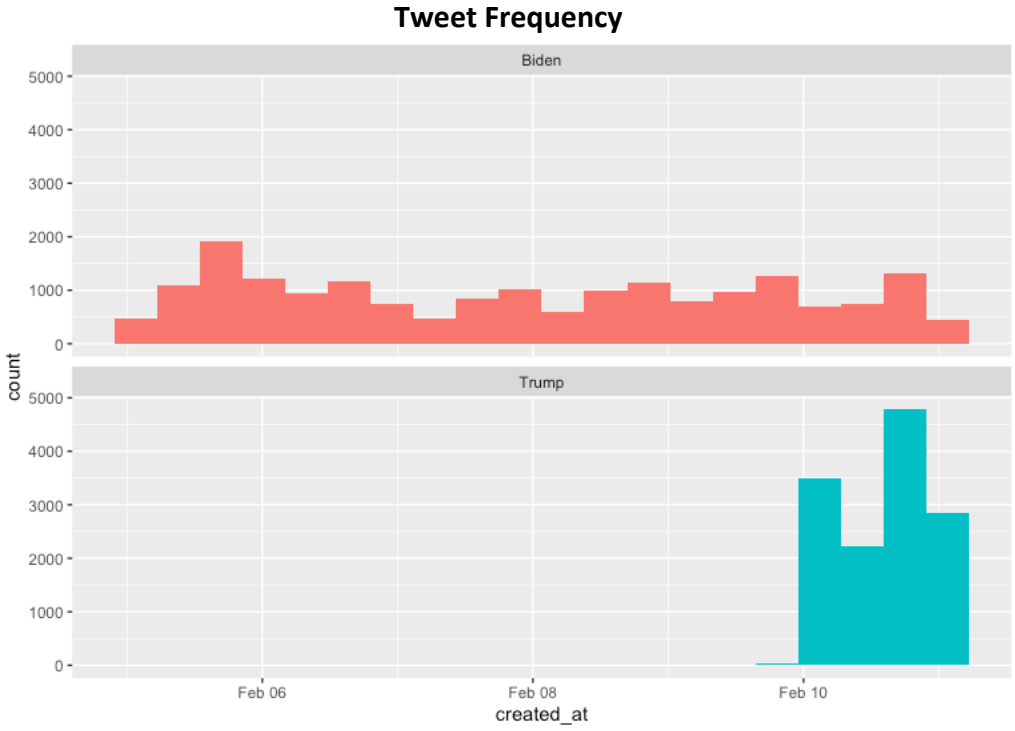
#Trump vs #Biden

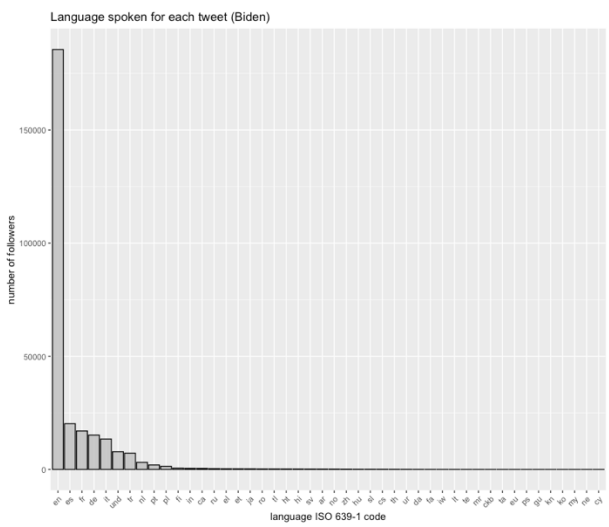
Twitter was Donald Trump's favorite medium of communication during his electoral campaign and presidency. Mr. Trump attracted many new users to the platform which Twitter happily welcomed. However, when Mr. Trump started using the platform to relay inaccurate, misleading or even false information in order to benefit his campaign, many pushed for limiting what Trump is allowed to share. For most of Mr. Trump's presidency, Twitter decided not to intervene in order to not control free speech. However, the situation was getting out of hand and they started adding informational messages under the tweets when the information shared could have been misleading or false. Recently, Mr. Trump tried to discredit Mr. Biden's electoral win by saying that the 2020 election was fraudulent. Mr. Trump pushed his most loyal supporters to storm the capitol to try to potentially destroy the votes of the electoral college. After this grave incident, Twitter took the decision to ban Mr. Trump from their platform.

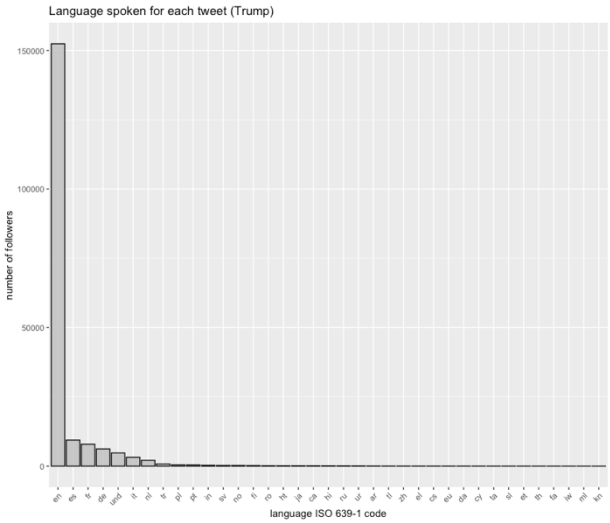
In this analysis, my goal is to better understand the political discourse on Twitter in a post-Trump area. Many of Mr. Trump's supporter announced that they would migrate to social platform that don't limit any speech. However, Mr. Trump is once more the center of attention due to his imminent impeachment trial.

I gather the 15,000 most recent tweets from the past 2 weeks that include the hashtag #trump and similarly for tweets with the hashtag #biden.



Here we notice that the daily posting volume with the #Trump hashtag is far greater. It takes less than 2 days to produce 15,000 tweets with that hashtag compared to over 3 days for the #biden hashtag.





Looking at the language spoken by each user that posted with these hashtags we noticed that the #Biden hashtag had a larger diversity in term of languages.

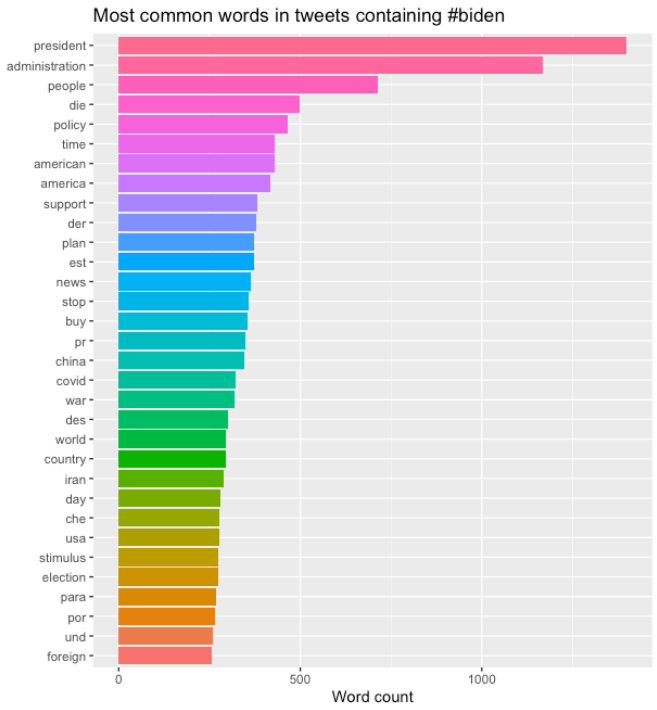
Biden:

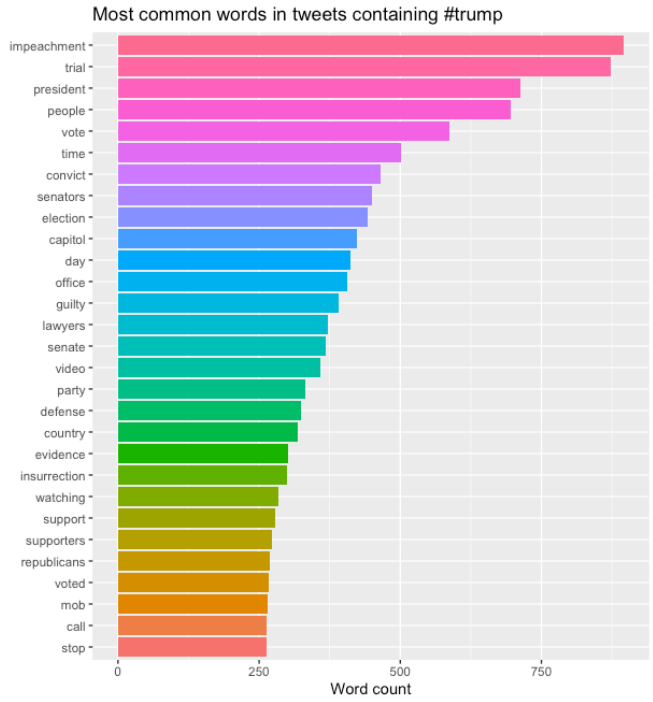
1	Brooklyn, NY	128
2	Manhattan, NY	103
3	Stratford, London	89
4	Ohio, USA	79
5	Rome, Lazio	78
6	Fayetteville, NC	57
7	Fingal, Ireland	52
8	New York, USA	52
9	Spain	50
10	Kensington, London	45

Trump:

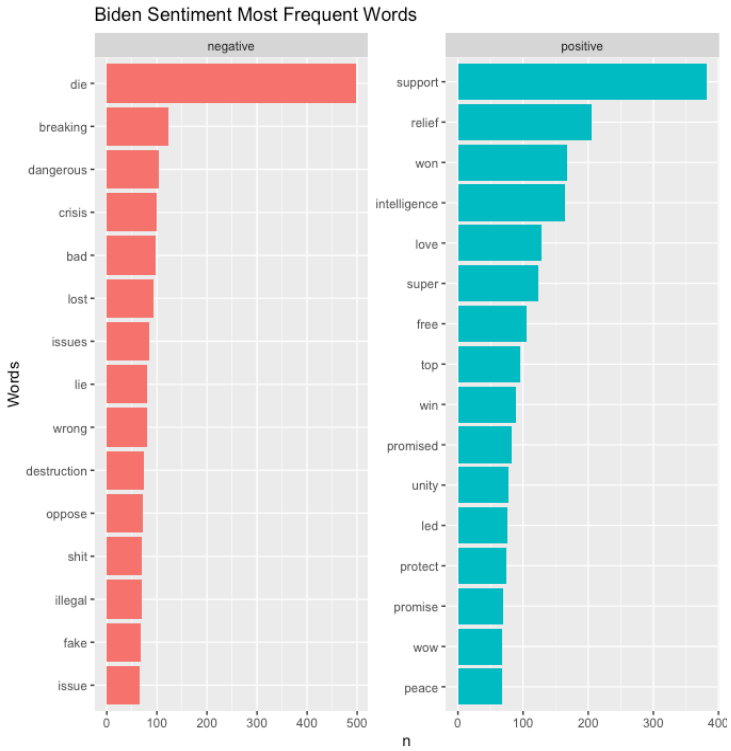
1	Kentucky, USA	113
2	Manhattan, NY	108
3	Washington, DC	97
4	Brooklyn, NY	70
5	Queens, NY	58
6	Carver, MN	54
7	Five Points, OH	49
8	Palm Springs, CA	49
9	Great Neck Gardens, NY	48
10	Rome, Lazio	48

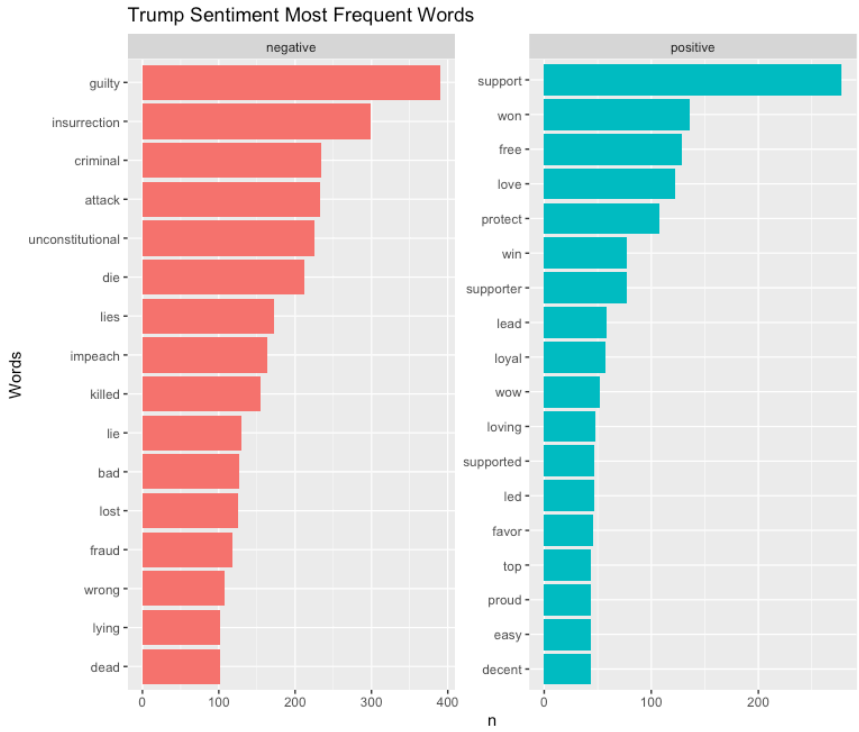
Additionally, looking at the locations from where each tweet originated, we can see that it mostly comes from the US. The trump hashtag mostly originated in the US, while the Biden one had greater diversity of location. The Trump hashtag was posted from more republican cities than the Biden hashtag.



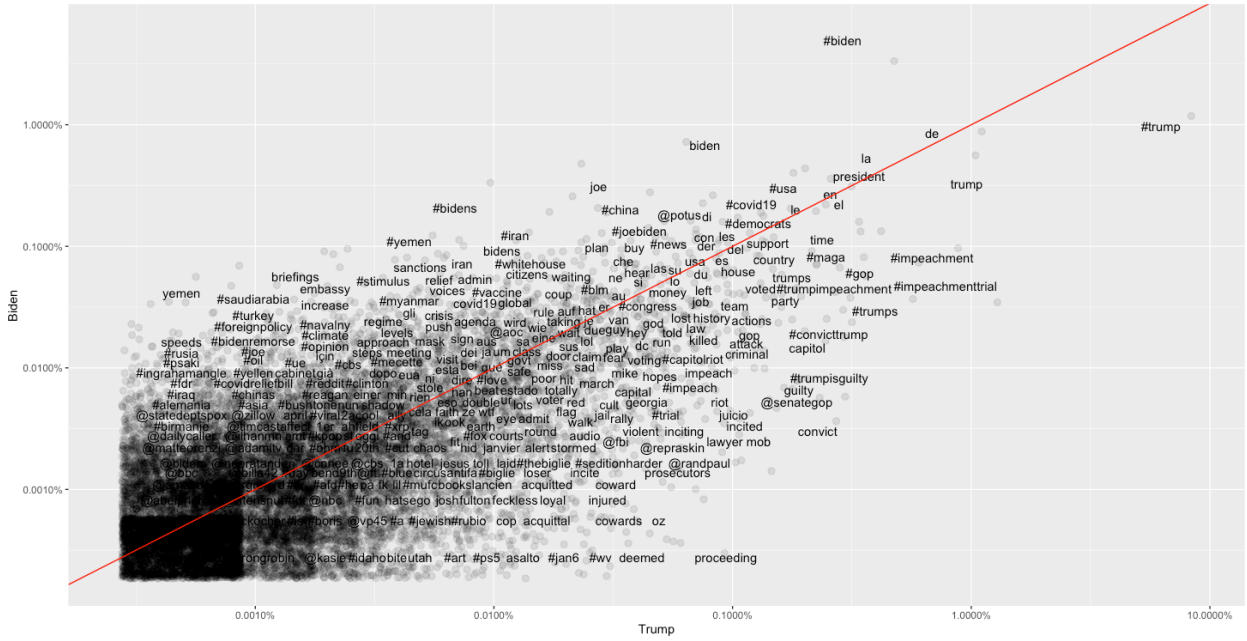


After removing stop words, non-English characters and URLs, I compiled the words that appeared the most often in tweets with each hashtag. We can see that the majority of tweets about Trump where regarding his upcoming impeachment trial. Most of the tweets about Biden are related to his administration and presidency. To further my analysis, I compiled the most common positive and negative words in tweets for each hashtag.

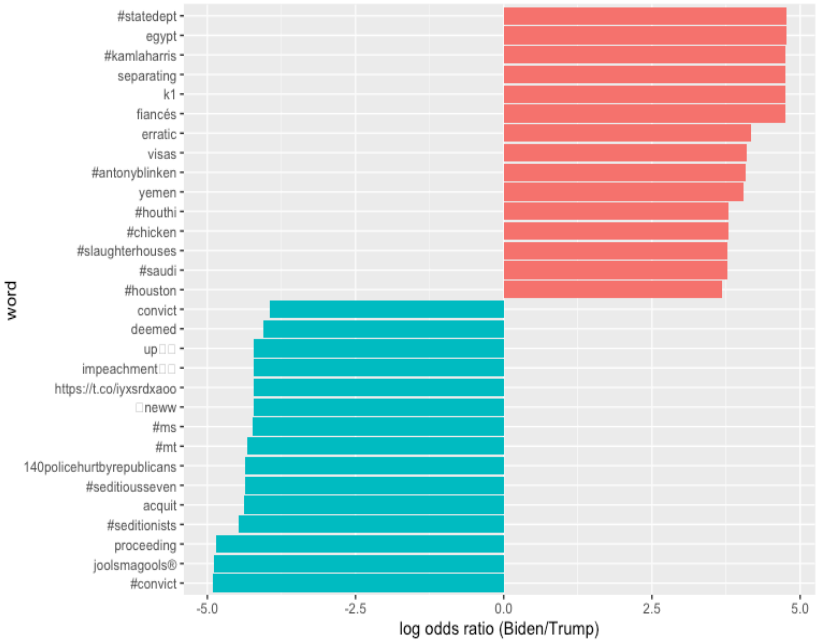




As we can see in the bar plots, there is a higher frequency of negative words in the Trump tweets. Mostly these words focus on his future trials. There are fewer negative tweets with the Biden hashtag, however the people tweeting negatively about Biden seem to have very strong sentiments since the highest occurring word is “kill”.

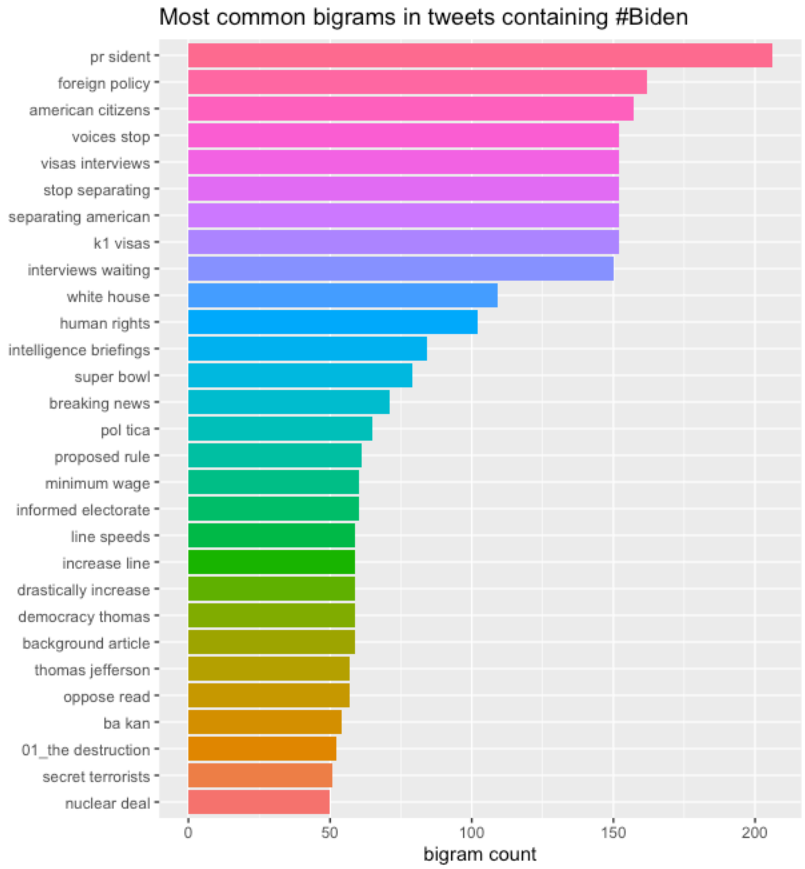


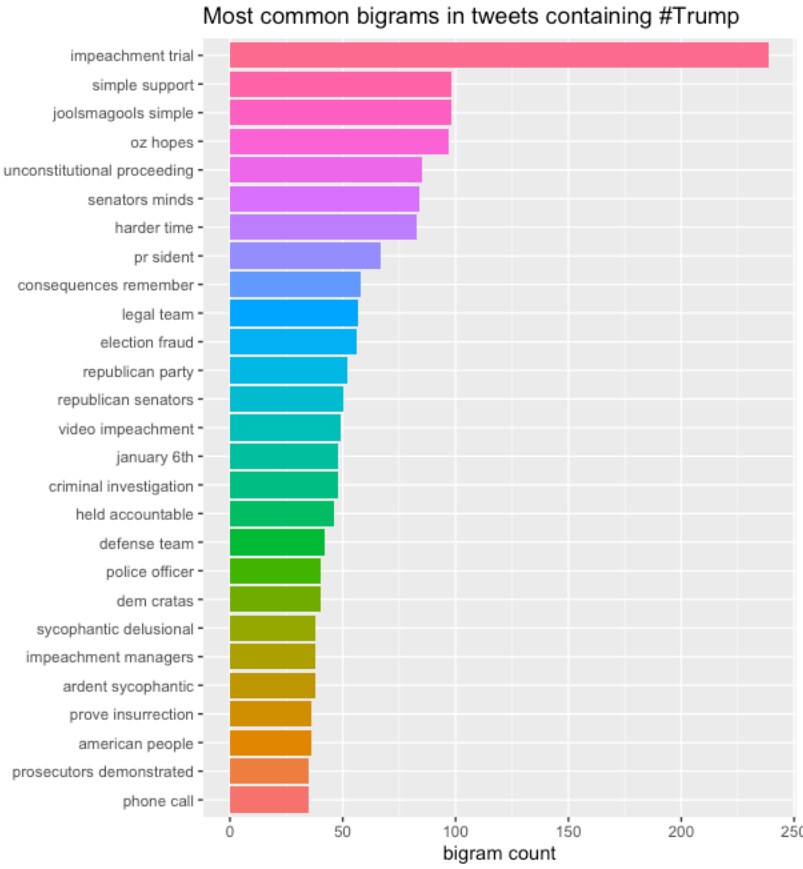
Instead of looking at the most often used word in general, I decided to look at the most often used word that only appear in either hashtags.
We can see that the words such as: impeachment, capitol, convict, mob, maga, violent, inciting, and other words in under the line are strongly associated with Trump. While the words potus, news, china, Iran, whitehouse, vaccine, and the other words above the line are strongly related to Biden. It seems like most tweets about Biden are talking about the current news and his actions compared to Trump tweets seem to be about the capitol riots and the upcoming impeachment trial.



To go more into details, I used the TF_IDF method to highlights the strongest words unique to each hashtag.
Convict is the most frequent word appearing only in the tweets with the #Trump.
I’m surprise not to see more words associated with Trump’s most loyal followers such as: recount, fraud, MAGA, America, First, 2nd amendment, Gun, etc.

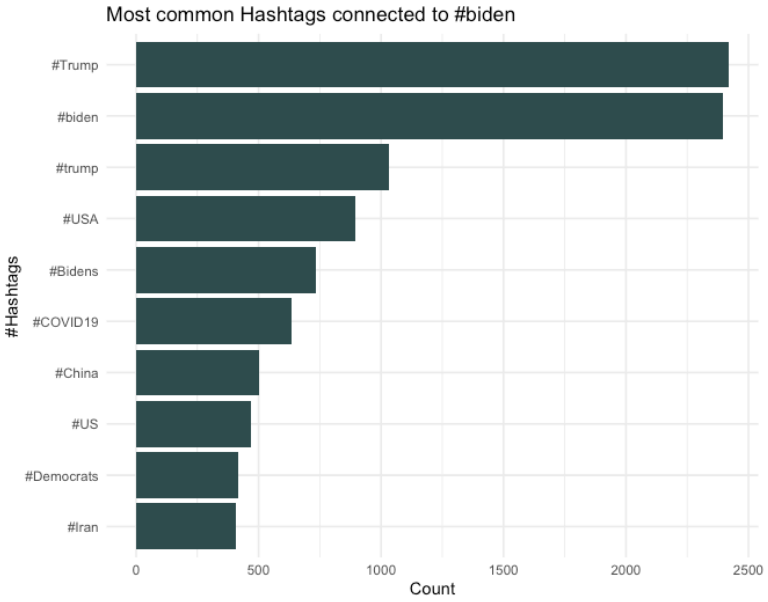
The one hashtag from that list that I was expecting is the #seditionistseven which are the 17 the attorney generals that sided with Trump in an effort to over-turn the election.

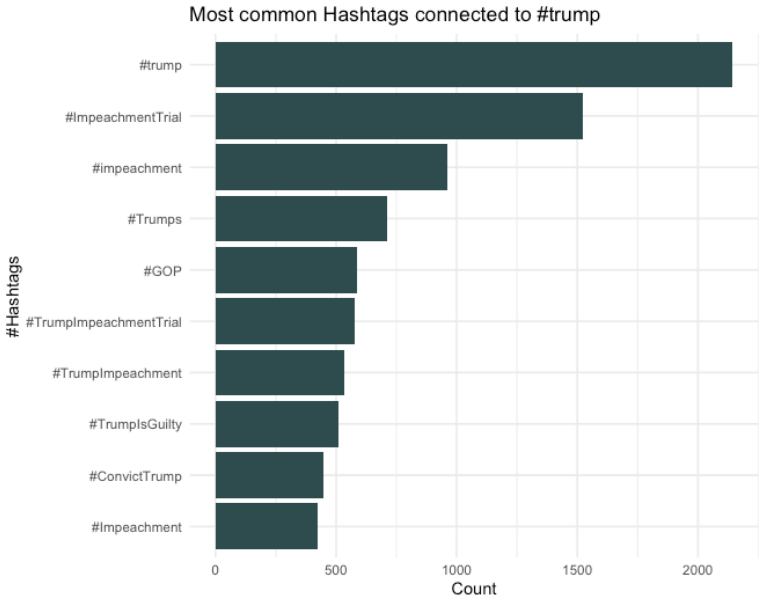




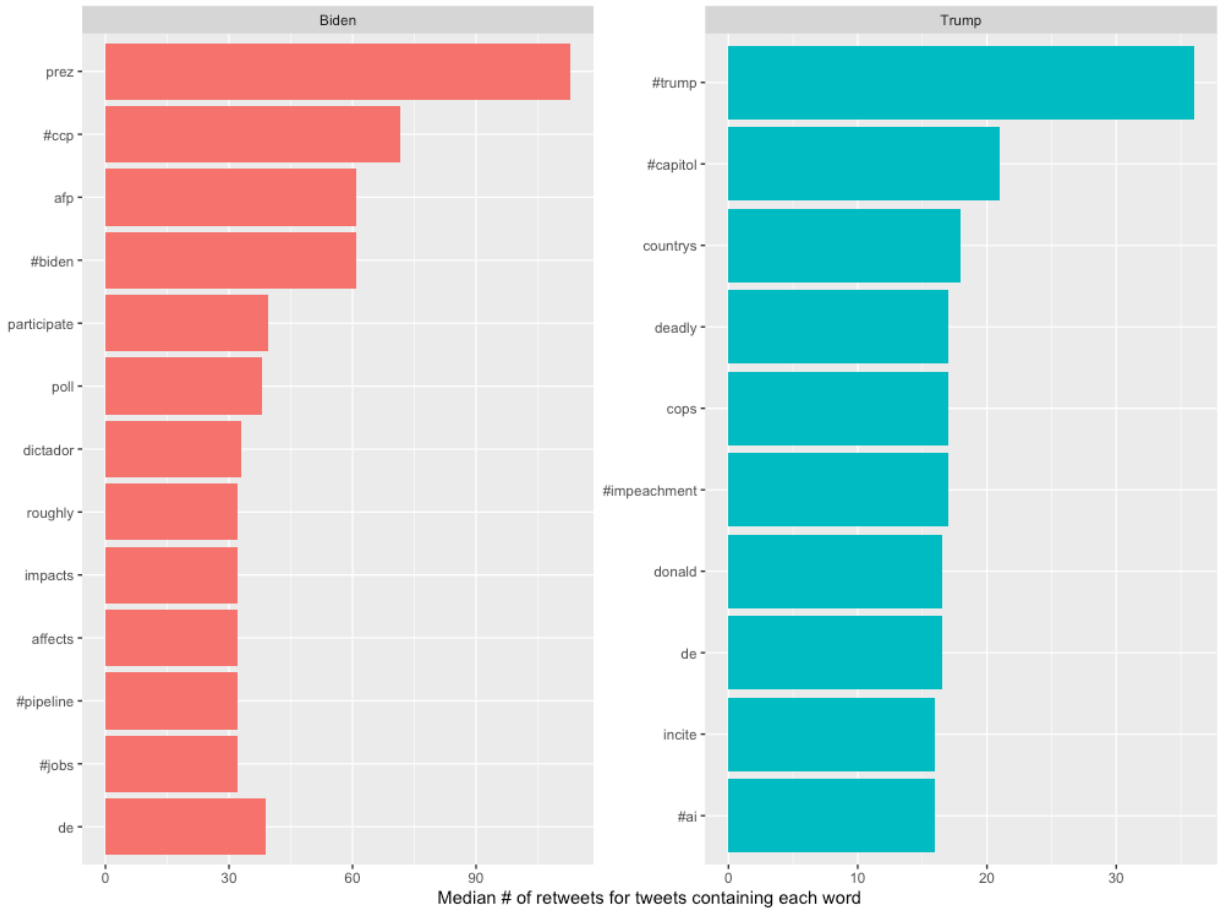
After highlighting the most comment bigrams for each hashtag, we can see that the main focus of the tweets with the #Trump are about the impeachment trial by far. We do observe some bigrams in favor of Trump such as unconstitutional proceeding and election fraud. The tweets with the #Biden are focused on Biden's new policies such foreign policy, the stop of separating immigrants and their children, k1 visas, and other breaking news.

To verify this sentiment, I analyze the hashtags the most associated with each of our main hashtags.





These graphs confirm our previous sentiment that the tweets with the hashtag Biden are mostly about the president ‘s new policies compared to the tweets with the trump hashtag that focus on the impeachment trial.



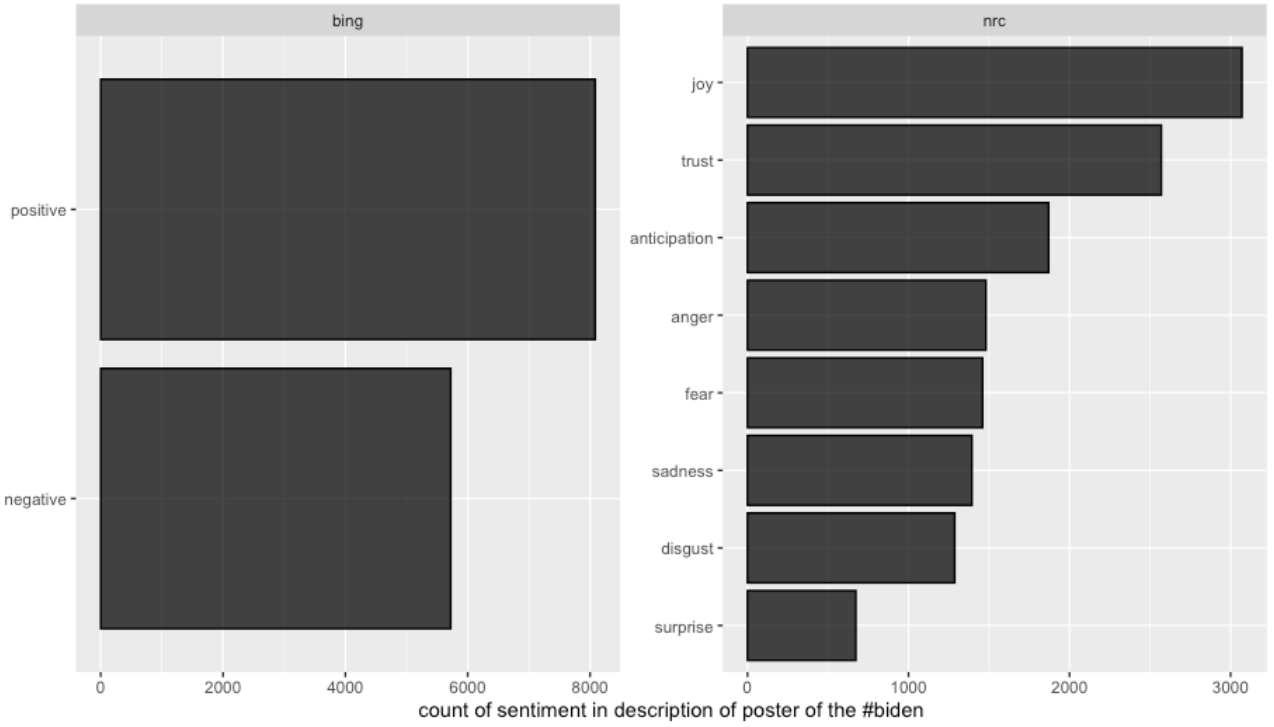
The tweet with the trump hashtags seems to be generating less attention (retweets) with only one word associated with a median of over 30 retweets. Comparatively, over 15 words are generating over 30 median retweets when associated with the Biden hashtag.

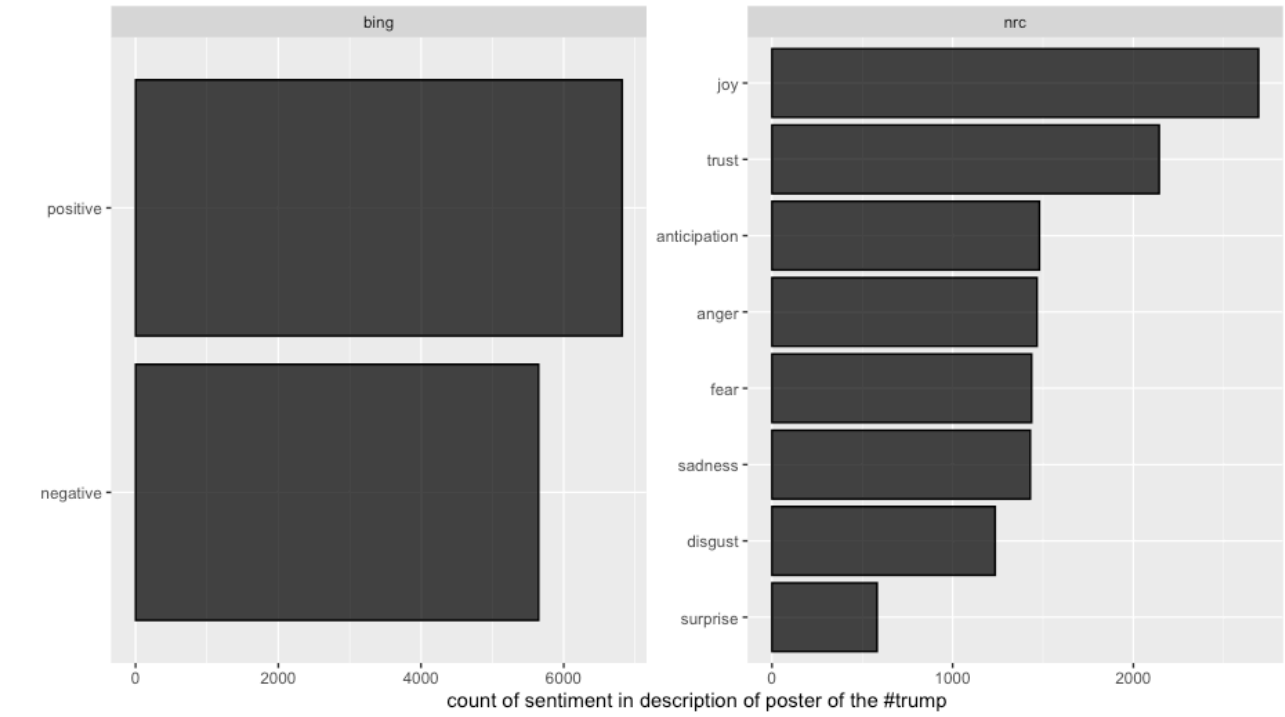
Lastly, I analyze the sentiment in each user's Twitter description (their bio) when they posted tweets with either hashtags.

<u>Twitter Description of user that posted a tweet with the Biden hashtag:</u>

<u>Twitter Description of user that that posted a tweet with the #Trump:</u>

Here we can observe a word cloud of the positive and negative words in user's bio. To better understand the sentiment in the user's bio, I used the Bing and nrc lexicons.





Comparatively, these graphs look somewhat similar. However, the count of positive decriptions for the user with the #Biden hashtag is much higher than the trump one. Similarly in the nrc,

the people that posted a Biden tweet where conveying more Joy, Trust, and anticipation than the people that tweeted with the #Trump hashtag.

Conclusion:
Overall, it seems that the people that tweeted with the #Trump hashtag in the past week had a more negative sentiment in their tweets and bio compared to the people that tweeted with the #Biden hashtag.

I was surprised to see very little of typical lexicon from the Trump supporters associated with the tweets including the #Trump hashtags. This is likely due to three reason:

- People are less vocals than during the reelection campaigns
- Trump supporters have moved on from Trump's loss and are less likely to tweet about Trump
- The most loyal Trump supporters have probably moved to other platform (such as Parler) that doesn't restrict their speech, since Mr. Trump's bans from Twitter.
-

It seemed like a lot of the content of the tweets with the #Trump hashtag was regarding his imminent trial which explains the higher volume of tweets especially in the last two days.

Twitter's decision of banning Trump from their platform and strengthening their policy on misinformation seem to have effectively limited misleading information regarding the election and other related issues. Very few of the tweets with the trump hashtag were supporting Trump's previous misinformation campaign. I recommend other social media platform to follow Twitter's new policy. Additionally, social media need to educate their user about the risks of fake news and how to safely share accurate information.

Social media platforms have for a long time refused to engage in the moderation of political speech. However, as we've seen with the capitol riots it's a double-edged sword.

To create a better online environment for all, it's important for social media to notify users when a tweet is potentially misleading. The system of providing information and facts under tweets with potentially incorrect information is a great way to limit the spread of "fake news".

Fact checkers are playing an increasingly important role while we're entering the age of deep-fakes and strong political division.

Real time data analysis is the key to monitor the truthfulness of readily available information and social media platforms need to heavily invest in protection their users from misleading information and create a system to limit the circulation of purposely fake information for personal/political gain.

Appendix

ABC news. (n.d.). Retrieved February 11, 2021, from https://abcnews.go.com/Politics/2020-Electoral-Interactive-Map

Conger, K., & Isaac, M. (2021, January 16). Inside twitter's decision to cut off trump. Retrieved February 11, 2021, from https://www.nytimes.com/2021/01/16/technology/twitter-donald-trump-jack-dorsey.html

Elsinghorst, S. (n.d.). Characterizing twitter followers with tidytext. Retrieved February 11, 2021, from https://shiring.github.io/text_analysis/2017/06/28/twitter_post

Haberman, M. (2021, January 10). Stripped of Twitter, TRUMP faces a new challenge: How to command attention. Retrieved February 11, 2021, from https://www.nytimes.com/2021/01/10/us/politics/trump-twitter.html?searchResultPosition=2

Kris Shaffer. (2019, February 14). Mining Twitter data with R, TidyText, and tags. Retrieved February 11, 2021, from https://www.pushpullfork.com/mining-twitter-data-tidy-text-tags/

Lab, T. (2020, January 08). Exploring tweets in r. Retrieved February 11, 2021, from https://medium.com/@traffordDataLab/exploring-tweets-in-r-54f6011a193d

Mining Emma Watson Twitter data with R. (2018, February 02). Retrieved February 11, 2021, from https://www.promptcloud.com/blog/data-mining-analytics-emma-watson-tweets-with-r/

Quealy, K. (2021, January 19). The complete list of Trump's Twitter Insults (2015-2021). Retrieved February 11, 2021, from https://www.nytimes.com/interactive/2021/01/19/upshot/trump-complete-insult-list.html?searchResultPosition=4

Robinson, J. (n.d.). Text mining with r: A tidy approach. Retrieved February 11, 2021, from https://www.tidytextmining.com/twitter.html

Rul, C. (2019, September 30). A guide to mining and analysing tweets with r. Retrieved February 11, 2021, from https://towardsdatascience.com/a-guide-to-mining-and-analysing-tweets-with-r-2f56818fdd16

Sanyal, P. (2020, December 10). Who are #SeditiousSeventeen? Twitter trend emerges as 17 Gop attorneys general try to overturn election results. Retrieved February 11, 2021, from https://meaww.com/seditious-seventeen-twitter-trend-17-attorneys-general-overturn-election-results-biden-won-donald-lost

Wanjiru, M. (2019, August 14). The power of social Media Analytics: Twitter text mining Using R. Retrieved February 11, 2021, from https://medium.com/@wanjirumaggie45/the-power-of-social-media-analytics-twitter-text-mining-using-r-1fceb26ac32b

Winter, E. (2018, December 04). Data mining Twitter using R: A guide for the very online. Retrieved February 11, 2021, from https://www.dataforprogress.org/blog/2018/11/28/data-mining-twitter-using-r-a-guide-for-the-very-online

R code

```
library(rtweet)
library(maps)
library(textdata)
library(dplyr)
library(tidyr)
library(tidytext)
library(tidyverse)
library(ggplot2)
library(stringr)



library(scales)

# Trump data
trump <- search_tweets("#trump", n =15000, include_rts = FALSE,
retryonratelimit=TRUE)

# Biden data
biden <-  search_tweets("#biden", n=15000, include_rts = FALSE,
retryonratelimit=TRUE)

# tokenize the data

# add new stop words to the list
custom_stop_words <- tribble(
  # Column names should match stop_words
  ~word, ~lexicon,
  # Add http, win, and t.co as custom stop words
  "de", "CUSTOM",
  "la", "CUSTOM",
  "el", "CUSTOM",
  "en", "CUSTOM",
  "le", "CUSTOM",
  "juicio", "CUSTOM",
```

```
  "castor", "CUSTOM",
  "es", "CUSTOM",
  "tico", "CUSTOM",
  "il", "CUSTOM",
  "del", "CUSTOM",
  "se", "CUSTOM",
  "les","CUSTOM",
  "di","CUSTOM",
  "con","CUSTOM",
  "los","CUSTOM",
  "al","CUSTOM",
  "joe","CUSTOM",
  "biden","CUSTOM",
  "trump", "CUSTOM",
  "donald", "CUSTOM"

)
# Bind the custom stop words to stop_words
stop_words2 <- stop_words %>%
  bind_rows(custom_stop_words)

reg_words <- "([^A-Za-z_\\d#@']|'(?![A-Za-z_\\d#@]))"
tidy_trump <- trump %>%
  filter(!str_detect(text, "^RT")) %>%
  mutate(text = str_replace_all(text, "https://t.co/[A-Za-z\\d]+|http://[A-
Za-z\\d]+|&amp;|&lt;|&gt;|RT|https", "")) %>%
  unnest_tokens(word, text, token = "regex", pattern = reg_words) %>%
  filter(!word %in% stop_words2$word,
         str_detect(word, "[a-z]"))

tidy_biden <- biden %>%
  filter(!str_detect(text, "^RT")) %>%
  mutate(text = str_replace_all(text, "https://t.co/[A-Za-z\\d]+|http://[A-
Za-z\\d]+|&amp;|&lt;|&gt;|RT|https", "")) %>%
  unnest_tokens(word, text, token = "regex", pattern = reg_words) %>%
  filter(!word %in% stop_words2$word,
         str_detect(word, "[a-z]"))

########################
## TRUMP analysis ######
########################

# most frequent words
tidy_trump %>%
  count(word, sort=TRUE) %>%
  filter(n>200) %>%
  mutate(word = reorder(word, n))
```

```
most_freq_trump <- tidy_trump %>%
  count(word, sort=TRUE) %>%
  filter(substr(word, 1, 1) != '#', # omit hashtags
         substr(word, 1, 1) != '@') %>% # omit Twitter handles
  mutate(word = reorder(word, n))

#visualize most common words
tidy_trump %>%
  count(word, sort=TRUE) %>%
  filter(substr(word, 1, 1) != '#', # omit hashtags
         substr(word, 1, 1) != '@', # omit Twitter handles
         n > 250) %>% # only most common words
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = word)) +
  geom_bar(stat = 'identity') +
  xlab(NULL) +
  ylab('Word count') +
  ggtitle(paste('Most common words in tweets containing #trump')) +
  theme(legend.position="none") +
  coord_flip()

# bigrams
tidy_bigrams_trump <- trump %>%
  filter(!str_detect(text, "^RT")) %>%
  mutate(text = str_replace_all(text, "https://t.co/[A-Za-z\\d]+|http://[A-
Za-z\\d]+|&amp;|&lt;|&gt;|RT|https", "")) %>%
  unnest_tokens(word, text, token = "regex", pattern = reg_words) %>%
  mutate(next_word = lead(word)) %>%
  filter(!word %in% stop_words2$word, # remove stop words
         !next_word %in% stop_words2$word, # remove stop words
         substr(word, 1, 1) != '@', # remove user handles to protect privacy
         substr(next_word, 1, 1) != '@', # remove user handles to protect
privacy
         substr(word, 1, 1) != '#', # remove hashtags
         substr(next_word, 1, 1) != '#',
         str_detect(word, "[a-z]"), # remove words containing ony numbers or
symbols
         str_detect(next_word, "[a-z]")) %>% # remove words containing ony
numbers or symbols
  filter(user_id == lead(user_id)) %>% # needed to ensure bigrams to cross
from one tweet into the next
  unite(bigram, word, next_word, sep = ' ') %>%
  select(bigram, created_at, user_id, quoted_followers_count,
quoted_friends_count, quoted_location)
```

```
tidy_bigrams_trump %>%
  count(bigram, sort=TRUE) %>%
  mutate(bigram = reorder(bigram, n))

#plot bigrams
tidy_bigrams_trump %>%
  count(bigram, sort=TRUE) %>%
  filter(n >= 35) %>%
  mutate(bigram = reorder(bigram, n)) %>%
  ggplot(aes(bigram, n, fill = bigram)) +
  geom_bar(stat = 'identity') +
  xlab(NULL) +
  ylab(paste('bigram count')) +
  ggtitle(paste('Most common bigrams in tweets containing #Trump')) +
  theme(legend.position="none") +
  coord_flip()

# Sentiment Analysis
trump_sentiment <-  most_freq_trump %>%
  inner_join(get_sentiments("bing")) %>%
  group_by(sentiment) %>%
  top_n(15, n) %>%
  ungroup() %>%
  mutate(word2 = fct_reorder(word, n))
ggplot(trump_sentiment, aes(x = word2, y = n, fill = sentiment)) +
geom_col(show.legend = FALSE) +
  facet_wrap(~ sentiment, scales = "free") +
  coord_flip() +
  labs(
    title = "Trump Sentiment Most Frequent Words",
    x = "Words" )



#####################
### Biden Analysis ##
#####################
# most frequent words
tidy_biden %>%
  count(word, sort=TRUE) %>%
  filter(n>200) %>%
  mutate(word = reorder(word, n))



 most_freq_biden <- tidy_biden %>%
  count(word, sort=TRUE) %>%
  filter(substr(word, 1, 1) != '#', # omit hashtags
```

```
         substr(word, 1, 1) != '@') %>% # omit Twitter handles
  mutate(word = reorder(word, n))

#visualize most common words
tidy_biden %>%
  count(word, sort=TRUE) %>%
  filter(substr(word, 1, 1) != '#', # omit hashtags
         substr(word, 1, 1) != '@', # omit Twitter handles
         n > 250) %>% # only most common words
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = word)) +
  geom_bar(stat = 'identity') +
  xlab(NULL) +
  ylab('Word count') +
  ggtitle(paste('Most common words in tweets containing #biden')) +
  theme(legend.position="none") +
  coord_flip()

# bigrams
tidy_bigrams_biden <- biden %>%
  filter(!str_detect(text, "^RT")) %>%
  mutate(text = str_replace_all(text, "https://t.co/[A-Za-z\\d]+|http://[A-
Za-z\\d]+|&amp;|&lt;|&gt;|RT|https", "")) %>%
  unnest_tokens(word, text, token = "regex", pattern = reg_words) %>%
  mutate(next_word = lead(word)) %>%
  filter(!word %in% stop_words2$word, # remove stop words
         !next_word %in% stop_words2$word, # remove stop words
         substr(word, 1, 1) != '@', # remove user handles to protect privacy
         substr(next_word, 1, 1) != '@', # remove user handles to protect
privacy
         substr(word, 1, 1) != '#', # remove hashtags
         substr(next_word, 1, 1) != '#',
         str_detect(word, "[a-z]"), # remove words containing ony numbers or
symbols
         str_detect(next_word, "[a-z]")) %>% # remove words containing ony
numbers or symbols
  filter(user_id == lead(user_id)) %>% # needed to ensure bigrams to cross
from one tweet into the next
  unite(bigram, word, next_word, sep = ' ') %>%
  select(bigram, created_at, user_id, quoted_followers_count,
quoted_friends_count, quoted_location)


tidy_bigrams_biden %>%
  count(bigram, sort=TRUE) %>%
  mutate(bigram = reorder(bigram, n))

#plot bigrams
```

```
tidy_bigrams_biden %>%
  count(bigram, sort=TRUE) %>%
  filter(n >= 50) %>%
  mutate(bigram = reorder(bigram, n)) %>%
  ggplot(aes(bigram, n, fill = bigram)) +
  geom_bar(stat = 'identity') +
  xlab(NULL) +
  ylab(paste('bigram count')) +
  ggtitle(paste('Most common bigrams in tweets containing #Biden')) +
  theme(legend.position="none") +
  coord_flip()



# Sentiment Analysis
biden_sentiment <-  most_freq_biden %>%
  inner_join(get_sentiments("bing")) %>%
  group_by(sentiment) %>%
  top_n(15, n) %>%
  ungroup() %>%
  mutate(word2 = fct_reorder(word, n))
ggplot(biden_sentiment, aes(x = word2, y = n, fill = sentiment)) +
geom_col(show.legend = FALSE) +
  facet_wrap(~ sentiment, scales = "free") +
  coord_flip() +
  labs(
    title = "Biden Sentiment Most Frequent Words",
    x = "Words" )

########################
## Joined analysis######
########################

# Frequency
tweets <- bind_rows(trump %>%
                      mutate(person = "Trump"),
                    biden %>%
                      mutate(person = "Biden"))


ggplot(tweets, aes(x = created_at, fill = person)) +
  geom_histogram(position = "identity", bins = 20, show.legend = FALSE) +
  facet_wrap(~person, ncol = 1)

library(tidytext)
library(stringr)

# Word Frequency
```

```
remove_reg <- "&amp;|&lt;|&gt;"
tidy_tweets <- tweets %>%
  filter(!str_detect(text, "^RT")) %>%
  mutate(text = str_remove_all(text, remove_reg)) %>%
  unnest_tokens(word, text, token = "tweets") %>%
  filter(!word %in% stop_words$word,
         !word %in% str_remove_all(stop_words$word, "'"),
         str_detect(word, "[a-z]"))

frequency <- tidy_tweets %>%
  group_by(person) %>%
  count(word, sort = TRUE) %>%
  left_join(tidy_tweets %>%
              group_by(person) %>%
              summarise(total = n())) %>%
  mutate(freq = n/total)

library(tidyr)

frequency <- frequency %>%
  select(person, word, freq) %>%
  spread(person, freq) %>%
  arrange(Trump, Biden)

library(scales)

ggplot(frequency, aes(Trump, Biden)) +
  geom_jitter(alpha = 0.1, size = 2.5, width = 0.25, height = 0.25) +
  geom_text(aes(label = word), check_overlap = TRUE, vjust = 1.5) +
  scale_x_log10(labels = percent_format()) +
  scale_y_log10(labels = percent_format()) +
  geom_abline(color = "red")

#Comparing word usage
word_ratios <- tidy_tweets %>%
  filter(!str_detect(word, "^@")) %>%
  count(word, person) %>%
  group_by(word) %>%
  filter(sum(n) >= 50) %>%
  ungroup() %>%
  spread(person, n, fill = 0) %>%
  mutate_if(is.numeric, list(~(. + 1) / (sum(.) + 1))) %>%
  mutate(logratio = log(Biden / Trump)) %>%
  arrange(desc(logratio))

word_ratios %>%
  arrange(abs(logratio))
word_ratios %>%
```

```
  group_by(logratio < 0) %>%
  top_n(15, abs(logratio)) %>%
  ungroup() %>%
  mutate(word = reorder(word, logratio)) %>%
  ggplot(aes(word, logratio, fill = logratio < 0)) +
  geom_col(show.legend = FALSE) +
  coord_flip() +
  ylab("log odds ratio (Biden/Trump)") +
  scale_fill_discrete(name = "", labels = c("Biden", "Trump"))


tidy_tweets %>%
  filter(!is.na(place_full_name)) %>%
  count(place_full_name, sort = TRUE) %>%
  top_n(5)


# Top Tweeting Location
tidy_trump %>%
  filter(!is.na(place_full_name)) %>%
  count(place_full_name, sort = TRUE) %>%
  top_n(10)

tidy_biden %>%
  filter(!is.na(place_full_name)) %>%
  count(place_full_name, sort = TRUE) %>%
  top_n(10)

# most retweeted tweet
trump %>%
  arrange(-retweet_count) %>%
  slice(1) %>%
  select(created_at, screen_name, text, retweet_count)

# top tweeters
trump %>%
  count(screen_name, sort = TRUE) %>%
  top_n(10) %>%
  mutate(screen_name = paste0("@", screen_name))

# top hashtags
trump_hash <- trump %>%
  unnest_tokens(hashtag, text, "tweets", to_lower = FALSE) %>%
  filter(str_detect(hashtag, "^#"),
         hashtag != "#Trump") %>%
  count(hashtag, sort = TRUE) %>%
  top_n(10)
```

```
  ggplot(trump_hash,aes(x=reorder(hashtag, +n), y=n))+
  geom_bar(stat="identity", fill="darkslategray")+
    theme_minimal() +
    xlab("#Hashtags") + ylab("Count") + coord_flip() +
    ggtitle("Most common Hashtags connected to #trump")



biden_hash <- biden %>%
  unnest_tokens(hashtag, text, "tweets", to_lower = FALSE) %>%
  filter(str_detect(hashtag, "^#"),
         hashtag != "#Biden") %>%
  count(hashtag, sort = TRUE) %>%
  top_n(10)



ggplot(biden_hash,aes(x=reorder(hashtag, +n), y=n))+
  geom_bar(stat="identity", fill="darkslategray")+
  theme_minimal() +
  xlab("#Hashtags") + ylab("Count") + coord_flip() +
  ggtitle("Most common Hashtags connected to #biden")

#######################################
# Basic Word cloud
library(wordcloud)
data("stop_words")

tweets_wc <- tweets %>%
  group_by(person) %>%
  ungroup() %>%
  unnest_tokens(word, text)%>%
  filter(person == "Trump") %>%
  anti_join(stop_words) %>%
  count(word, sort=T)

tweets_wc %>%
  with(wordcloud(word, n, max.words = 100))

library(reshape2)

tweets_wc %>%
  inner_join(get_sentiments('nrc')) %>%
  count(word, sentiment, sort=TRUE) %>%
  acast(word ~sentiment, value.var="n", fill=0) %>%
  comparison.cloud(xxxxxxx = c("grey10", "grey60"),
                   max.words=500, scale=c(1, 0.5), random.order = T)
#######################################
```

```
tidy_tweets2 <- tweets %>%
  filter(!str_detect(text, "^(RT|@)")) %>%
  mutate(text = str_remove_all(text, remove_reg)) %>%
  unnest_tokens(word, text, token = "tweets", strip_url = TRUE) %>%
  filter(!word %in% stop_words2$word,
         !word %in% str_remove_all(stop_words2$word, "'"))

totals <- tidy_tweets %>%
  group_by(person, retweet_count) %>%
  summarise(favs = first(retweet_count)) %>%
  group_by(person) %>%
  summarise(total_favs = sum(favs))

word_by_rts <- tidy_tweets %>%
  group_by(retweet_count, word, person) %>%
  summarise(rts = first(retweet_count)) %>%
  group_by(person, word) %>%
  summarise(retweet_count = median(rts), uses = n()) %>%
  left_join(totals) %>%
  filter(retweet_count != 0) %>%
  ungroup()

word_by_rts %>%
  filter(uses >= 5) %>%
  arrange(desc(retweet_count))

word_by_rts %>%
  filter(uses >= 5) %>%
  group_by(person) %>%
  top_n(10, retweet_count) %>%
  arrange(retweet_count) %>%
  ungroup() %>%
  mutate(word = factor(word, unique(word))) %>%
  ungroup() %>%
  ggplot(aes(word, retweet_count, fill = person)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ person, scales = "free", ncol = 2) +
  coord_flip() +
  labs(x = NULL,
       y = "Median # of retweets for tweets containing each word")


# language spoken
tidy_trump %>%
  count(lang) %>%
  droplevels() %>%
  ggplot(aes(x = reorder(lang, desc(n)), y = n)) +
  geom_bar(stat = "identity", color = 'black', fill = 'grey', alpha = 0.8) +
```

```
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)) +
  labs(x = "language ISO 639-1 code",
       y = "number of followers")+
  ggtitle("Language spoken for each tweet (Trump)")


library(tidytext)
library(SnowballC)


# analysis of people's description
tidy_descr_t <- trump %>%
  unnest_tokens(word, description) %>%
  mutate(word_stem = wordStem(word)) %>%
  anti_join(stop_words2, by = "word") %>%
  filter(!grepl("\\.|http", word))


tidy_descr_t %>%
  count(word_stem, sort = TRUE) %>%
  filter(n > 275) %>%
  ggplot(aes(x = reorder(word_stem, n), y = n)) +
  geom_col(color = "grey3", fill = "grey1", alpha = 0.8) +
  coord_flip() +
  labs(x = "",
       y = "count of word stem in all followers' descriptions")

#################3
tidy_descr_ngrams <- trump %>%
  unnest_tokens(bigram, description, token = "ngrams", n = 2) %>%
  filter(!grepl("\\.|http", bigram)) %>%
  separate(bigram, c("word1", "word2"), sep = " ") %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)



bigram_counts <- tidy_descr_ngrams %>%
  count(word1, word2, sort = TRUE)

bigram_counts %>%
  filter(n > 125) %>%
  ggplot(aes(x = reorder(word1, -n), y = reorder(word2, -n), fill = n)) +
  geom_tile(alpha = 0.8, color = "white") +
  scale_fill_gradientn(colours = c('black', 'white')) +
  coord_flip() +
  theme(legend.position = "right") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)) +
```

```
  labs(x = "first word in pair",
       y = "second word in pair")


library(igraph)
library(ggraph)

bigram_graph <- bigram_counts %>%
  filter(n > 5) %>%
  graph_from_data_frame()

set.seed(1)

a <- grid::arrow(type = "closed", length = unit(.15, "inches"))

ggraph(bigram_graph, layout = "fr") +
  geom_edge_link(aes(edge_alpha = n), show.legend = FALSE,
                 arrow = a, end_cap = circle(.07, 'inches')) +
  geom_node_point(color =  'black', size = 1, alpha = 0.8) +
  geom_node_text(aes(label = name), vjust = 1, hjust = 0.5) +
  theme_void()


bigrams_separated <- trump %>%
  unnest_tokens(bigram, description, token = "ngrams", n = 2) %>%
  filter(!grepl("\\.|http", bigram)) %>%
  separate(bigram, c("word1", "word2"), sep = " ") %>%
  filter(word1 == "not" | word1 == "no") %>%
  filter(!word2 %in% stop_words$word)

tidy_descr_sentiment <- tidy_descr_t %>%
  left_join(select(bigrams_separated, word1, word2), by = c("word" =
"word2")) %>%
  inner_join(get_sentiments("nrc"), by = "word") %>%
  inner_join(get_sentiments("bing"), by = "word") %>%
  rename(nrc = sentiment.x, bing = sentiment.y) %>%
  mutate(nrc = ifelse(!is.na(word1), NA, nrc),
         bing = ifelse(!is.na(word1) & bing == "positive", "negative",
                       ifelse(!is.na(word1) & bing == "negative", "positive",
bing)))

tidy_descr_sentiment %>%
  filter(nrc != "positive") %>%
  filter(nrc != "negative") %>%
  gather(x, y, nrc, bing) %>%
  count(x, y, sort = TRUE) %>%
  filter(n > 10) %>%
  ggplot(aes(x = reorder(y, n), y = n)) +
```

```
  facet_wrap(~ x, scales = "free") +
  geom_col(color = 'black', fill = 'grey3', alpha = 0.8) +
  coord_flip() +
  labs(x = "",
       y = "count of sentiment in description of poster of the #trump")


#########
# Sentiment in #biden description
# analysis of people's description
tidy_descr_b <- biden %>%
  unnest_tokens(word, description) %>%
  mutate(word_stem = wordStem(word)) %>%
  anti_join(stop_words2, by = "word") %>%
  filter(!grepl("\\.|http", word))


tidy_descr_b %>%
  count(word_stem, sort = TRUE) %>%
  filter(n > 275) %>%
  ggplot(aes(x = reorder(word_stem, n), y = n)) +
  geom_col(color = "grey3", fill = "grey1", alpha = 0.8) +
  coord_flip() +
  labs(x = "",
       y = "count of word stem in all followers' descriptions")

#################
tidy_descr_ngrams_b <- biden %>%
  unnest_tokens(bigram, description, token = "ngrams", n = 2) %>%
  filter(!grepl("\\.|http", bigram)) %>%
  separate(bigram, c("word1", "word2"), sep = " ") %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)



bigram_counts_b <- tidy_descr_ngrams_b %>%
  count(word1, word2, sort = TRUE)

bigram_counts_b %>%
  filter(n > 125) %>%
  ggplot(aes(x = reorder(word1, -n), y = reorder(word2, -n), fill = n)) +
  geom_tile(alpha = 0.8, color = "white") +
  scale_fill_gradientn(colours = c('black', 'white')) +
  coord_flip() +
  theme(legend.position = "right") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, hjust = 1)) +
  labs(x = "first word in pair",
```

```
       y = "second word in pair")


library(igraph)
library(ggraph)

bigram_graph_b <- bigram_counts_b %>%
  filter(n > 5) %>%
  graph_from_data_frame()

set.seed(1)

a <- grid::arrow(type = "closed", length = unit(.15, "inches"))

ggraph(bigram_graph_b, layout = "fr") +
  geom_edge_link(aes(edge_alpha = n), show.legend = FALSE,
                 arrow = a, end_cap = circle(.07, 'inches')) +
  geom_node_point(color =  'black', size = 1, alpha = 0.8) +
  geom_node_text(aes(label = name), vjust = 1, hjust = 0.5) +
  theme_void()


bigrams_separated_b <- biden %>%
  unnest_tokens(bigram, description, token = "ngrams", n = 2) %>%
  filter(!grepl("\\.|http", bigram)) %>%
  separate(bigram, c("word1", "word2"), sep = " ") %>%
  filter(word1 == "not" | word1 == "no") %>%
  filter(!word2 %in% stop_words$word)

tidy_descr_sentiment_b <- tidy_descr_t %>%
  left_join(select(bigrams_separated_b, word1, word2), by = c("word" =
"word2")) %>%
  inner_join(get_sentiments("nrc"), by = "word") %>%
  inner_join(get_sentiments("bing"), by = "word") %>%
  rename(nrc = sentiment.x, bing = sentiment.y) %>%
  mutate(nrc = ifelse(!is.na(word1), NA, nrc),
         bing = ifelse(!is.na(word1) & bing == "positive", "negative",
                       ifelse(!is.na(word1) & bing == "negative", "positive",
bing)))

tidy_descr_sentiment_b %>%
  filter(nrc != "positive") %>%
  filter(nrc != "negative") %>%
  gather(x, y, nrc, bing) %>%
  count(x, y, sort = TRUE) %>%
  filter(n > 10) %>%
  ggplot(aes(x = reorder(y, n), y = n)) +
  facet_wrap(~ x, scales = "free") +
```

```
  geom_col(color = 'black', fill = 'grey3', alpha = 0.8) +
  coord_flip() +
  labs(x = "",
       y = "count of sentiment in description of poster of the #biden")

library(reshape2)
tidy_descr_sentiment_b %>%
  count(word, bing, sort = TRUE) %>%
  acast(word ~ bing, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("red", "green"),
                   max.words = 100)

library(ggplot2)
library(paletteer)

tidy_descr_sentiment %>%
  count(word, bing, sort = TRUE) %>%
  acast(word ~ bing, value.var = "n", fill = 0) %>%
  comparison.cloud(colors = c("red", "green"),
                   max.words = 100)
```