PDS Final Project

Damann, Plutzer, Caragine, Lee

4/30/2020

```
## -- Attaching packages --------------------------------------------------------------------- tidyverse 1
## v ggplot2 3.2.1     v purrr   0.3.3
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   1.0.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
## -- Conflicts ------------------------------------------------------------------------ tidyverse_conflic
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
##
## Attaching package: 'data.table'
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
## The following object is masked from 'package:purrr':
##
##     transpose
## Loading required package: NLP
##
## Attaching package: 'NLP'
## The following object is masked from 'package:ggplot2':
##
##     annotate
## stm v1.3.5 successfully loaded. See ?stm for help.
##  Papers, resources, and other materials at structuraltopicmodel.com
## Package version: 2.0.1
## Parallel computing: 2 of 4 threads used.
## See https://quanteda.io for tutorials and examples.
##
## Attaching package: 'quanteda'
## The following objects are masked from 'package:tm':
##
##     as.DocumentTermMatrix, stopwords
## The following objects are masked from 'package:NLP':
##
##     meta, meta<-
## The following object is masked from 'package:utils':
##
##     View
```

#BOT PROJECT

Introduction

While there has been growing concern over the prevalence of bots in social media and how they impact the national discourse on politics and various issues, there has not been much research regarding what sort of information the bots are disseminating and how their message has changed over time. In this project, we wanted to look into this issue by examining a data set of Twitter bot activity during the 2016 election.

Data

In October 2018, Twitter released data on the 3,613 accounts associated with the Russian Internet Research Agency. This company is a known "troll farm" that facilitates the generation of fake accounts and news aimed at influencing public opinion. Our analysis explores the content of the nearly 3 million tweets in this corpus, seeking to understand what types of information the bots introduce.

In an initial exploration of the data, we find that a substantial amount of the tweets are not original. Thirty-six percent of all English tweets from Russian bots are retweeted from other accounts.

```
sum(iraTweets$is_retweet == TRUE)/length(iraTweets$is_retweet)
```

```
## [1] 0.3612952
```

We noticed that many tweets dealt with topics of race and racially-charged issues. Below, we find what percentage of all tweets in the data set mention race.

```
sum(str_detect(str_to_lower(iraTweets$tweet_text),pattern = 'black lives matter|blm|#takeaknee|black|#bl
```

```
## [1] 0.04329268
```

While this is not a high percentage of tweets, Donald Trump and Hillary Clinton are only mentioned 7.6% of tweets.

```
sum(str_detect(str_to_lower(iraTweets$tweet_text),pattern = 'trump|donald|hillary|clinton'))/length(iraT
```

```
## [1] 0.07620894
```

Method

We use the `stm` library, employing a topic model to examine what content Twitter bots distributed during and following the 2016 U.S. presidential election. We split the data into groups of tweets being published before and after several events of interest: the first presidential debate (September 26), Clinton's emails and Trump's Access Hollywood tape are leaked (October 7), the election (November 8) and the Trump's inauguration (January 20). Because these tweets contain limited text, constrained to a maximum of 140 characters, we concatenate the text of all tweets made by an individual bot account as a single document. Next, we processed the data to remove hashtags, mentions, English stop words, links and emojis. We assemble the documents into a document frequency matrix with the `quanteda` library. This object stores individuals documents as rows and frequency of terms as columns.

```
data_list = list(group1,group2,group3,group4,group5,group6,group7) # Assembling a list to make iteratio

master_dfm = dfm(" ",groups = "empty") # Intialize a dfm with an empty document
docscount = 0 # To keep track of the total number of documents in the dfm
for (i in 1:length(data_list)) { #This outer loop iterates over the time group divisions
  group_df = data_list[[i]]
  unique_handles = unique(group_df$userid) #Find the unique userid's for this time group
  start = proc.time() #Timing each group for the function. This will print out the time it takes to do
  for (j in 1:length(unique_handles)) { #Iterate over each of the unique userid's
    df = group_df[group_df$userid == unique_handles[j]] #Subset the group dataframe down to just the cu
    ### Initial text cleanup
    doc_string = unlist(str_split(tolower(df$tweet_text),pattern = " ")) #Create a vector of text for t
```

```r
    ######## PLAY AROUND WITH FILTERING AND TEXT CLEANING BETWEEN HERE AND THE NEXT BREAK ########

    doc_string = gsub("#\\w+ *", "", doc_string) #Remove hashtags
    doc_string = gsub("@\\w+ *", "", doc_string) #Remove user mentions
    doc_string = gsub("t.co\\w+ *", "", doc_string) #Remove twitter condensed links
    doc_string = gsub("tco\\w+ *", "", doc_string) #Remove twitter condensed links
    doc_string = gsub("http\\w+ *", "", doc_string) #Remove links
    doc_string = gsub("[^\x01-\x7F]", "", doc_string) #Remove emojis and non-ascii characters
    doc_string = gsub('[[:digit:]]+', '', doc_string) #Removing numbers... not sure if it's a good idea
    doc_string = removePunctuation(doc_string)
    doc_string = gsub("tco\\w+ *", "", doc_string) #Remove twitter condensed links
    doc_string = doc_string[!(doc_string %in% stopwords("en"))] #Removing common english stopwords
    doc_string = doc_string[doc_string != ""] #This is necessary because "" elements cause some problem
    doc_string = doc_string[doc_string != "rt"] #Getting rid of these because I think they are unlikely
    doc_string = doc_string[doc_string != "O"]
    doc_string = doc_string[doc_string != "("] #These are apparently widely used bit I don't know why
    doc_string = doc_string[doc_string != ")"]
    doc_string = gsub("amp", "", doc_string) #remove all the amps

    ########### END OF TEXT CLEANING AND FILTERING ##############################

    ## Creating frequency matrix element for this document
    doc_corpus = corpus(paste(doc_string,collapse = " "))
    doc_name = paste("g",i,"-",unique_handles[j],sep = "") #Making a name for each doc
    doc_dfm = dfm(doc_corpus,groups = doc_name)

    #Attaching this document to the master document-frequency matrix
    master_dfm = rbind(master_dfm,doc_dfm)
  }
  docscount = docscount + length(unique_handles)
  print("Finished group")
  print(proc.time()-start) #Prints out the time elapsed for the text processing for this group
}
```

```
## [1] "Finished group"
##    user  system elapsed
##  98.167  22.439 137.623
## [1] "Finished group"
##    user  system elapsed
## 108.766  30.930 156.529
## [1] "Finished group"
##    user  system elapsed
## 140.310  25.439 128.748
## [1] "Finished group"
##    user  system elapsed
## 123.779  12.050  88.821
## [1] "Finished group"
##    user  system elapsed
## 137.377  20.503 110.598
## [1] "Finished group"
##    user  system elapsed
## 123.651  13.064  86.952
## [1] "Finished group"
```

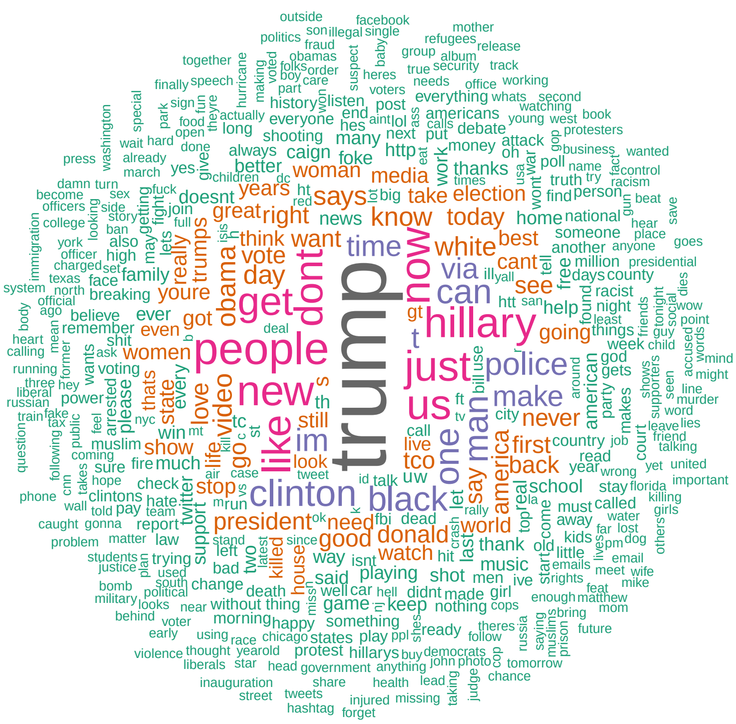
```
##    user  system elapsed
## 117.678  14.758  93.507
```

To begin exploring the data, we use the `topfeatures()` command to view the 50 most frequently used words in our DFM. Unsurprisingly, there are several mentions of Donald Trump and Hillary Clinton.

```
##     trump    people       new        us      just   hillary       now
##     19863      9540      9268      9039      8946      8939      8651
##      dont       get      like   clinton     black       one       man
##      8512      8372      8083      7238      6973      6812      6709
##       can        im    police         t       via      time      make
##      6702      6352      6246      5308      5217      4998      4973
##      know       day     obama     white      says     video      want
##      4795      4791      4665      4523      4341      4326      4145
##        go president   america       tco      good       say     right
##      4119      4048      3928      3865      3861      3797      3758
##      vote    donald      love     first     never      back       see
##      3739      3720      3666      3628      3624      3570      3531
##     today     think      need      cant     watch      take     going
##      3423      3404      3367      3291      3277      3241      3214
##     media
##      3195
```

A more aesthetic way to depict the most frequently used words is through a word plot.

```r
textplot_wordcloud(master_dfm, min_count = 25, random_order = FALSE,
                   rotation = .25,
                   color = RColorBrewer::brewer.pal(8, "Dark2"))
```

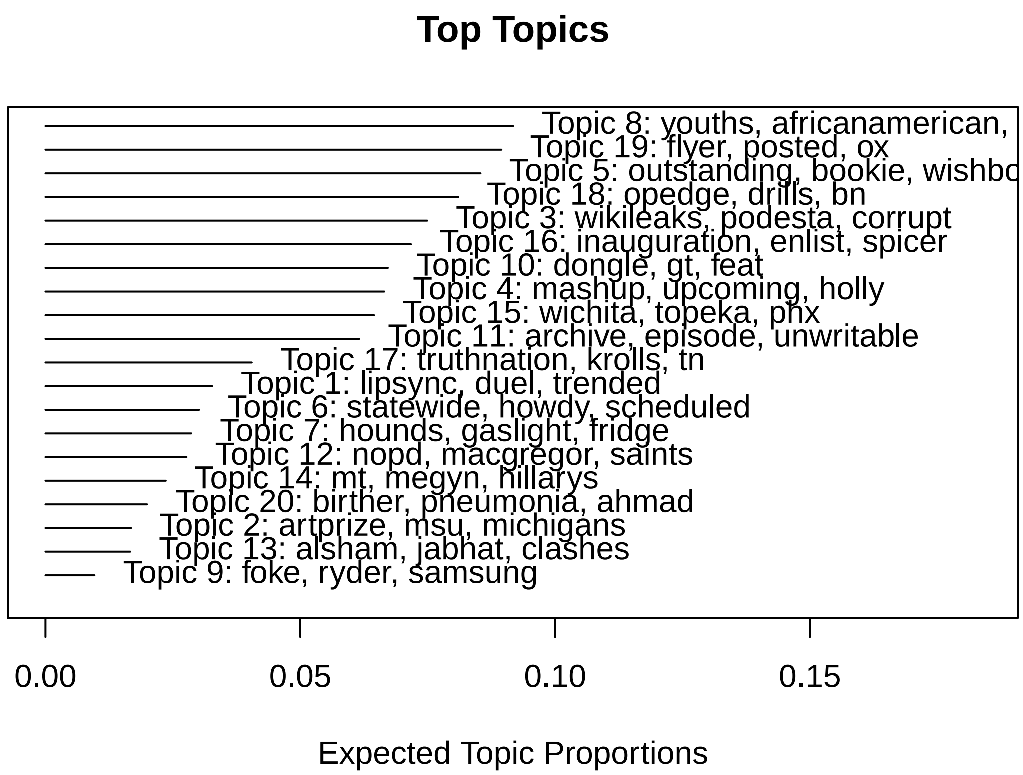


After processing the data, we train the topic model to return 20 topics with the following code:

```r
stm = stm(master_dfm,K = 20, verbose = F, init.type = "Spectral")
```

Our topic model is a model for how these tweet documents are generated. In the model, terms in the tweets are drawn randomly from different topics. Different sets of tweets pulls from various topics more than others. –bulk this. Below, the figure depicts the 20 topics found by the model. The terms following each topic number are the most frequently used words exclusive to that topic. On their own, these terms are not particularly meaningful. We analyze the documents that came from each of these topics to get a better idea of what each topic is really about.

```
plot(stm, labeltype = c("frex"))
```



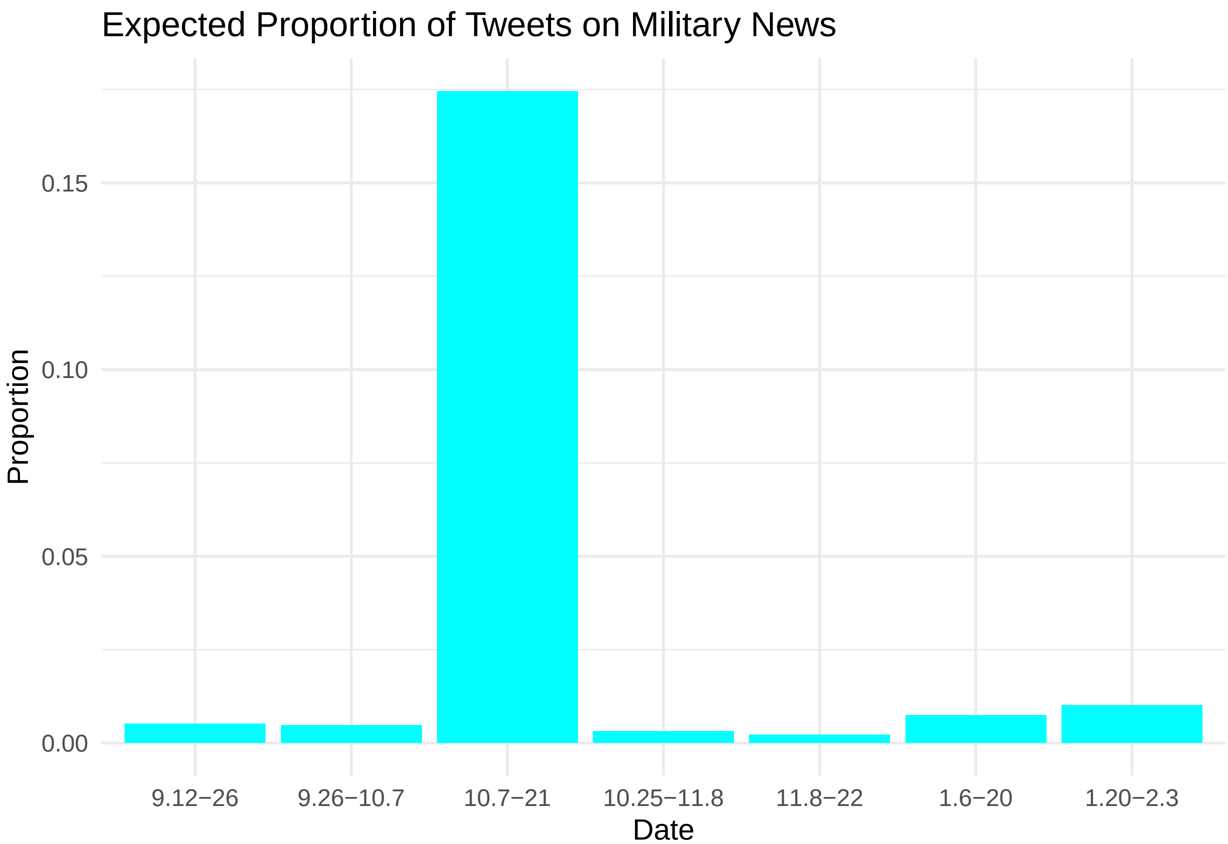
discuss plot; While the words displayed next to the topics give a hint at the content of the topic, we look more closely at the four topics we found most interesting.

The first topic prominent in the corpus is military news. Several novel tweets from bot accounts discuss ongoing U.S. involvement in foreign conflicts. For example, many tweets discuss news on Syria during the time of the U.S. Raqqa campaign starting in October 2016.

`iraqi air force destroys #is convoy north of #ramadi, #iraq https://t.co/zoa1vfur8g`

`un-brokered cease-fire begins in yemen https://t.co/lmrcaehkjm #isis`

As seen in the figure below, the prevalence of this topic peaks near the time of the election.

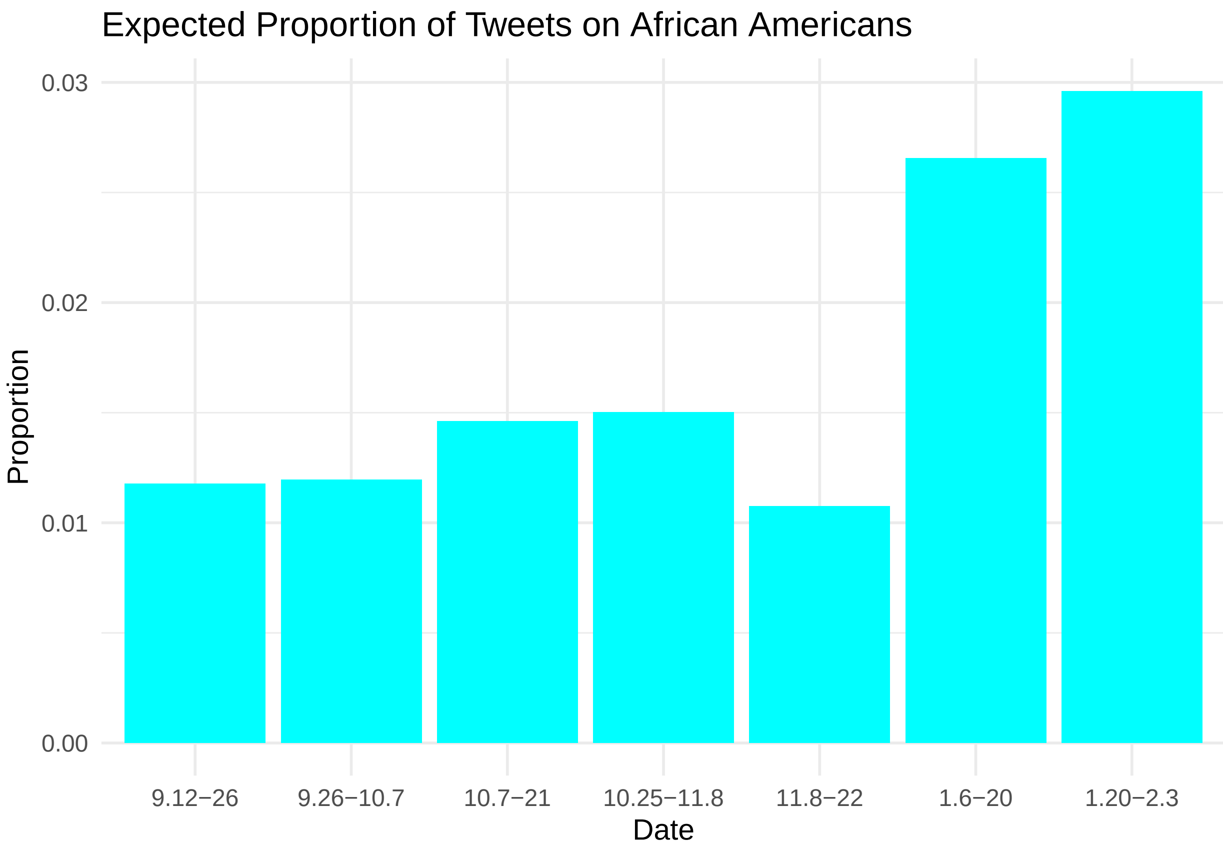


There is a general trend for bots to discuss racially-charged topics. One topic from the model focuses on the experiences, plights and successes of African Americans. For example,

`today we remind you of these black men & youths who were killed by police https://t.co/t6kwdtxogv"`

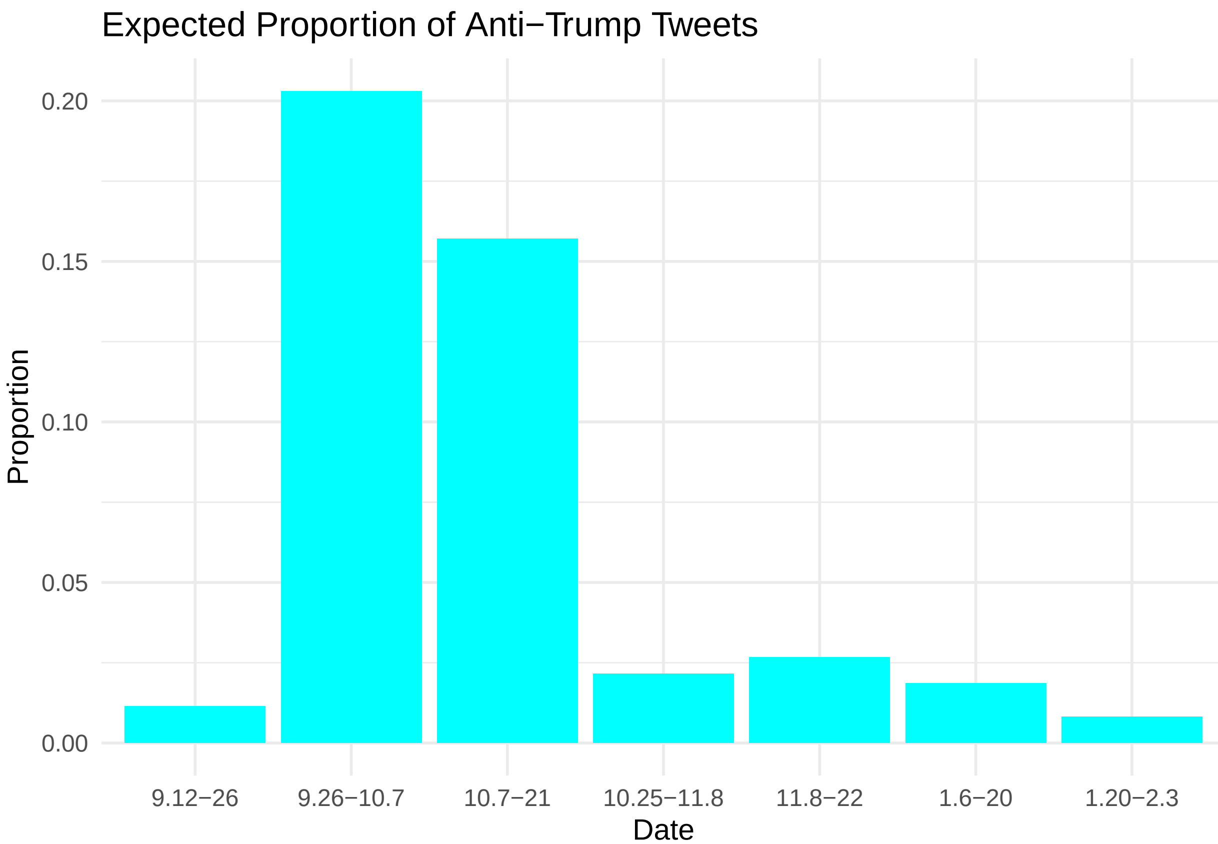
`"white guy taken home to his parents. black kid killed. white jury acquits. welcome to amerikkka. nothi`

In the figure below, it is clear that more tweets discuss these issues after the election. It is possible that these tweets are a reaction to the election of Donald Trump. We cannot specify the cause of the increased frequency of this topic in the period from October 7 to 21.

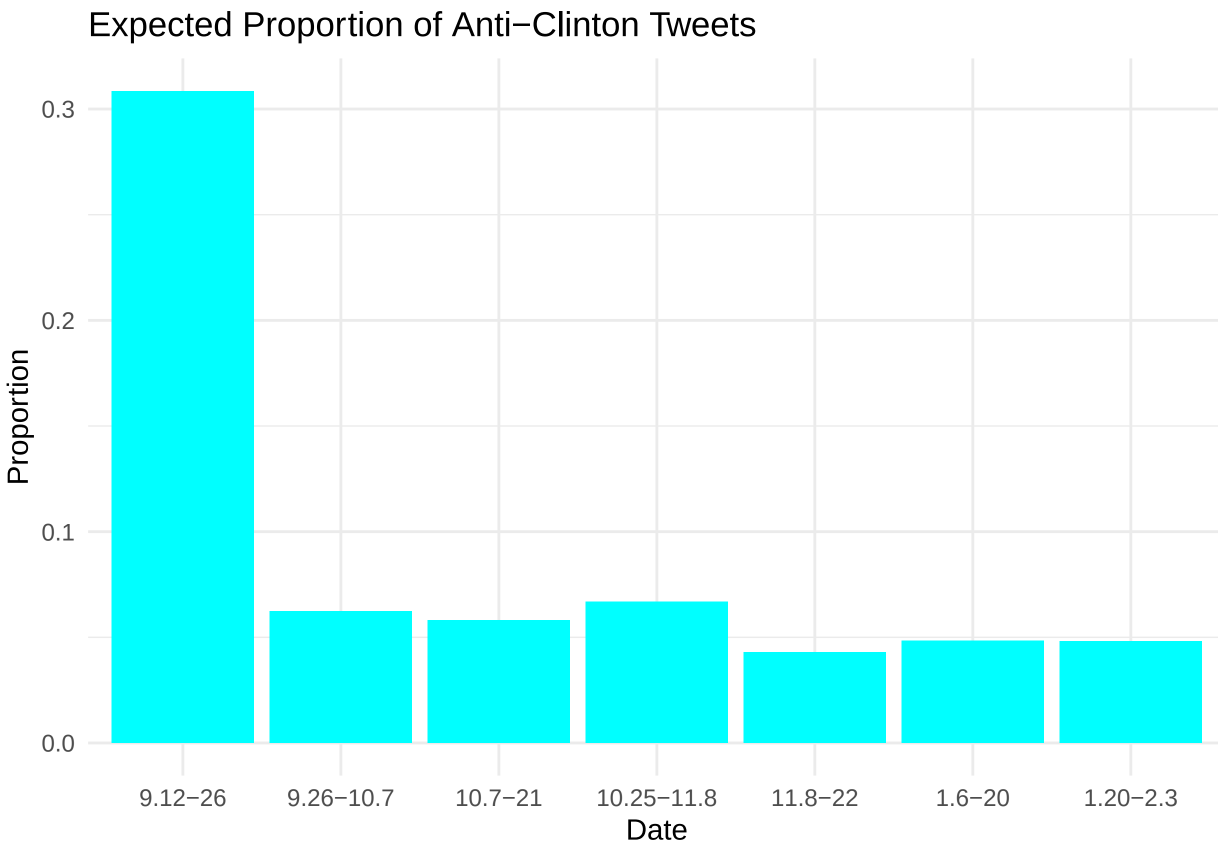


Many of the tweets in this corpus are starkly partisan. In the figure below, it is clear that anti-Trump tweets are common starting the two weeks prior to the election and after the election. It is surprising, perhaps, that there is a dip in anti-Trump tweets following October 7. While Clinton's e-mails were leaked during this time period, it is also the same time period that Trump's Access Hollywood tape leaked.

`if no one stops him and Putin from taking power, we'll all be dead by spring`



However, there is a large celebration of Trump's victory from Twitter bots immediately before the inauguration. These tweets focus on typical ultra-conservative topics from the election. The plot below shows the dramatic increase in the celebratory tweets around the time of the inauguration.



##Conclusion

While the findings from this project certainly expand upon the existing knowledge on bots and their misinformation campaigns, there are still many questions that have yet to be answered. For example, while our project figured out what sort of topics bots are most likely to tweet and how their engagement in these topics changed over the course of the 2016 presidential election, further research could be conducted on the efficacy of such misinformation campaigns. Perhaps future research could measure and analyze the success of bot tweets in spreading misinformation by looking into the number of retweets or interactions that non-bot Twitter users have with bot tweets. By doing research on this subject, we could better understand whether bots are successful in spreading misinformation, and whether their success varies across different topics or different time periods.