You may have heard that two researchers at Clemson University [analyzed almost 3 millions tweets from the Internet Research Agency (IRA)](https://bigthink.com/stephen-johnson/analysis-of-3-million-russian-troll-tweets-reveals-5-distinct-types-of-accounts) – a “Russian troll factory”. In partnership with [FiveThirtyEight](https://fivethirtyeight.com/features/why-were-sharing-3-million-russian-troll-tweets/), they made all of their data available on [GitHub](https://github.com/fivethirtyeight/russian-troll-tweets/). So of course, I had to read the files into R, which I was able to do with this code:

files <- c("IRAhandle\_tweets\_1.csv",  
 "IRAhandle\_tweets\_2.csv",  
 "IRAhandle\_tweets\_3.csv",  
 "IRAhandle\_tweets\_4.csv",  
 "IRAhandle\_tweets\_5.csv",  
 "IRAhandle\_tweets\_6.csv",  
 "IRAhandle\_tweets\_7.csv",  
 "IRAhandle\_tweets\_8.csv",  
 "IRAhandle\_tweets\_9.csv")  
my\_files <- paste0("~/Downloads/russian-troll-tweets-master/",files)  
  
each\_file <- function(file) {  
 tweet <- read\_csv(file) }  
  
library(tidyverse)

tweet\_data <- NULL  
for (file in my\_files) {  
 temp <- each\_file(file)  
 temp$id <- sub(".csv", "", file)  
 tweet\_data <- rbind(tweet\_data, temp)  
}

Note that this is a large file, with 2,973,371 observations of 16 variables. Let’s do some cleaning of this dataset first. The researchers, Darren Linvill and Patrick Warren, identified 5 majors types of trolls:

* Right Troll: These Trump-supporting trolls voiced right-leaning, populist messages, but “rarely broadcast traditionally important Republican themes, such as taxes, abortion, and regulation, but often sent divisive messages about mainstream and moderate Republicans…They routinely denigrated the Democratic Party, e.g. @LeroyLovesUSA, January 20, 2017, “#ThanksObama We’re FINALLY evicting Obama. Now Donald Trump will bring back jobs for the lazy ass Obamacare recipients,” the authors wrote.
* Left Troll: These trolls mainly supported Bernie Sanders, derided mainstream Democrats, and focused heavily on racial identity, in addition to sexual and religious identity. The tweets were “clearly trying to divide the Democratic Party and lower voter turnout,” the authors told FiveThirtyEight.
* News Feed: A bit more mysterious, news feed trolls mostly posed as local news aggregators who linked to legitimate news sources. Some, however, “tweeted about global issues, often with a pro-Russia perspective.”
* Hashtag Gamer: Gamer trolls used hashtag games—a popular call/response form of tweeting—to drum up interaction from other users. Some tweets were benign, but many “were overtly political, e.g. @LoraGreeen, July 11, 2015, “#WasteAMillionIn3Words Donate to #Hillary.”
* Fearmonger: These trolls, who were least prevalent in the dataset, spread completely fake news stories, for instance “that salmonella-contaminated turkeys were produced by Koch Foods, a U.S. poultry producer, near the 2015 Thanksgiving holiday.”

But a quick table of the results of the variable, account\_category, shows 8 in the dataset.

table(tweet\_data$account\_category)

##   
## Commercial Fearmonger HashtagGamer LeftTroll NewsFeed   
## 122582 11140 241827 427811 599294   
## NonEnglish RightTroll Unknown   
## 837725 719087 13905

The additional three are Commercial, Non-English, and Unknown. At the very least, we should drop the Non-English tweets, since those use Russian characters and any analysis I do will assume data are in English. I’m also going to keep only a few key variables. Then I’m going to clean up this dataset to remove links, because I don’t need those for my analysis – I certainly wouldn’t want to follow them to their destination. If I want to free up some memory, I can then remove the large dataset.

reduced <- tweet\_data %>%  
 select(author,content,publish\_date,account\_category) %>%  
 filter(account\_category != "NonEnglish")  
  
library(qdapRegex)

##   
## Attaching package: 'qdapRegex'

reduced$content <- rm\_url(reduced$content)  
  
rm(tweet\_data)

Now we have a dataset of 2,135,646 observations of 4 variables. I’m planning on doing some analysis on my own of this dataset – and will of course share what I find – but for now, I thought I’d repeat a technique I’ve covered on this blog and demonstrate a new one.

library(tidytext)  
  
tweetwords <- reduced %>%  
 unnest\_tokens(word, content) %>%  
 anti\_join(stop\_words)

## Joining, by = "word"

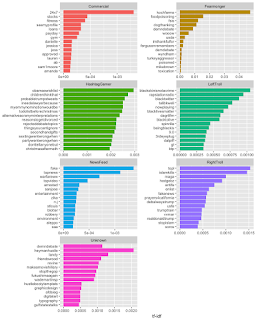
wordcounts <- tweetwords %>%  
 count(account\_category, word, sort = TRUE) %>%  
 ungroup()  
  
head(wordcounts)

## # A tibble: 6 x 3  
## account\_category word n  
##   
## 1 NewsFeed news 124586  
## 2 RightTroll trump 95794  
## 3 RightTroll rt 86970  
## 4 NewsFeed sports 47793  
## 5 Commercial workout 42395  
## 6 NewsFeed politics 38204

First, I’ll conduct a TF-IDF analysis of the dataset. This code is a repeat from a [previous post](http://www.deeplytrivial.com/2018/07/statistics-sunday-more-text-analysis.html).

tweet\_tfidf <- wordcounts %>%  
 bind\_tf\_idf(word, account\_category, n) %>%  
 arrange(desc(tf\_idf))  
  
tweet\_tfidf %>%  
 mutate(word = factor(word, levels = rev(unique(word)))) %>%  
 group\_by(account\_category) %>%  
 top\_n(15) %>%  
 ungroup() %>%  
 ggplot(aes(word, tf\_idf, fill = account\_category)) +  
 geom\_col(show.legend = FALSE) +  
 labs(x = NULL, y = "tf-idf") +  
 facet\_wrap(~account\_category, ncol = 2, scales = "free") +  
 coord\_flip()

## Selecting by tf\_idf

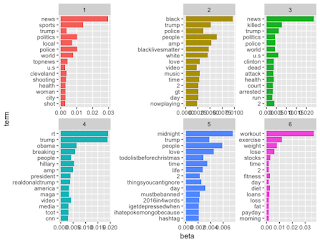
[](https://i2.wp.com/1.bp.blogspot.com/-SbdpFBt5MZk/W3Bd-Wvi3-I/AAAAAAAANaY/PBSzebwUl-cKMjQDOqHAKcovfHRpvdNlwCLcBGAs/s1600/tfidf_tweets1.png?ssl=1)

But another method of examining terms and topics in a set of documents is [Latent Dirichlet Allocation (LDA)](http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf), which can be conducted using the R package, topicmodels. The only issue is that LDA requires a [document term matrix](https://en.wikipedia.org/wiki/Document-term_matrix). But we can easily convert our wordcounts dataset into a DTM with the cast\_dtm function from tidytext. Then we run our LDA with topicmodels. Note that LDA is a random technique, so we set a random number seed, and we specify how many topics we want the LDA to extract (k). Since there are 6 account types (plus 1 unknown), I’m going to try having it extract 6 topics. We can see how well they line up with the account types.

tweets\_dtm <- wordcounts %>%  
 cast\_dtm(account\_category, word, n)  
  
library(topicmodels)  
tweets\_lda <- LDA(tweets\_dtm, k = 6, control = list(seed = 42))  
tweet\_topics <- tidy(tweets\_lda, matrix = "beta")

Now we can pull out the top terms from this analysis, and plot them to see how they lined up.

top\_terms <- tweet\_topics %>%  
 group\_by(topic) %>%  
 top\_n(15, beta) %>%  
 ungroup() %>%  
 arrange(topic, -beta)  
  
top\_terms %>%  
 mutate(term = reorder(term, beta)) %>%  
 ggplot(aes(term, beta, fill = factor(topic))) +  
 geom\_col(show.legend = FALSE) +  
 facet\_wrap(~topic, scales = "free") +  
 coord\_flip()

[](https://i0.wp.com/4.bp.blogspot.com/-JAkFUPH8zic/W3BeKIg5ubI/AAAAAAAANac/ztZ8NY_jneUuH7zhoiCu49FdFqrKbZbLgCLcBGAs/s1600/LDA_tweets.png?ssl=1)