SIPS-twitter

Dylan Wiwad

2018-06-26

Scraping tweets and doing basic frequencies

I'm uploading this markdown doc, which will basically just be a bit more heavily annotated code, so people who are interested can see how simple it is to get data from Twitter!

Obligatory set up chunk, just to get a bunch of packages.

```
knitr::opts_chunk$set(echo = TRUE)
# Packages
library(formatR)
library(ggplot2)
library(stringr) # Cleaning the tweet text
library(twitteR) # Scraping twitter
library(ROAuth) # Need it to get authentication with twitter
library(plyr)
```

```
##
## Attaching package: 'plyr'
```

```
## The following object is masked from 'package:twitteR':
##
##     id
```

```
library(httr)
library(tm)
```

```
## Loading required package: NLP
```

```
##
## Attaching package: 'NLP'
```

```
## The following object is masked from 'package:httr':
##
##     content
```

```
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```
# defining a function to append to a list
lappend <- function(lst, obj) {
  lst[[length(lst)+1]] <- obj
  return(lst)
}
```

The first thing I'm going to do is grab all of the tweets with the SIPS2018 hashtag, using the searchTwitter function from the TwitteR package. In order to do this I had to setup an account with Twitter to get an API key. You can do this on apps.twitter.com. Then I entered the keys up above, where you would enter your own to use the script.

I have hidden the code that I used to set up my api key, because I can't share it as it is linked to my twitter account. However, here is the chunk I used:

api_key <- “”
api_secret <- “”
access_token <- “”
access_token_secret <- “”
setup_twitter_oauth(api_key, api_secret, access_token, access_token_secret)

If you want to use this code, or work with twitter, you need to make an account as described above, insert your keys in the quotation marks above and run the setup_twitter_oauth command.

So with that in mind, let's call twitter and get the tweets.

```
# Get all the tweets
tweets <- searchTwitter("#SIPS2018", n = 3000)
# Turn the tweets into a DF
df <- twListToDF(tweets)
# Remove retweets
og_tweets <- df[ which(df$isRetweet=='FALSE'),]

# Print out the first bit of the twitter data so you can see what it looks like
head(df)
```

```
##
## 1    RT @JessieSunPsych: .@katiecorker asks how we can overcome the tendency to classify and divide a
## 2    RT @cathleenogrady: Was chatting to @lakens about capricious science news leading to public mist
## 3   RT @dstephenlindsay: Pleased to announce new articulation of criteria for acceptance at Psycholog
## 4    RT @paulrconnor: I've been to a bunch of psychology conferences, but I've never been so impresse
## 5 RT @cruwelli: Hey #SIPS2018 &amp;amp; especially those with FOMO: if you want to read through some pape
## 6   RT @dstephenlindsay: Pleased to announce new articulation of criteria for acceptance at Psycholog
##   favorited favoriteCount replyToSN             created truncated
## 1     FALSE             0      <NA> 2018-06-26 21:50:42     FALSE
## 2     FALSE             0      <NA> 2018-06-26 21:48:47     FALSE
## 3     FALSE             0      <NA> 2018-06-26 21:48:26     FALSE
## 4     FALSE             0      <NA> 2018-06-26 21:47:48     FALSE
## 5     FALSE             0      <NA> 2018-06-26 21:47:19     FALSE
## 6     FALSE             0      <NA> 2018-06-26 21:47:12     FALSE
##   replyToSID                  id replyToUID
## 1       <NA> 1011728486040592391       <NA>
## 2       <NA> 1011728001610993670       <NA>
## 3       <NA> 1011727916152049664       <NA>
## 4       <NA> 1011727756911173632       <NA>
## 5       <NA> 1011727633544044544       <NA>
## 6       <NA> 1011727603034619904       <NA>
##                                                                    statusSource
## 1   <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>
## 2 <a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>
## 3 <a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>
## 4   <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>
## 5   <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>
## 6 <a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>
##        screenName retweetCount isRetweet retweeted longitude latitude
## 1      jaricheson            3      TRUE     FALSE        NA       NA
## 2 SanabriaLucenaD            9      TRUE     FALSE        NA       NA
## 3 SanabriaLucenaD           24      TRUE     FALSE        NA       NA
## 4    juliafstrand            1      TRUE     FALSE        NA       NA
## 5   lorraine_hope           21      TRUE     FALSE        NA       NA
## 6   EJWagenmakers           24      TRUE     FALSE        NA       NA
```

Next is to just pull out the text variable - all I'm concerned with right now is counting words. So grabbing that column and cleaning it a bit. First I split each tweet on spaces so every word becomes it's own element, then append every word of every tweet to a new list called "together." I also print out the first few rows of the "text" column, so you can see what I'm working with.

```
content <- og_tweets$text
head(content)
```

```
## [1] "Almost too tired from a long, productive conference to do more social things except to thank eve
## [2] "The list goes on and I'm tired. So check out the progress and promise here https://t.co/4hIXTdEr
## [3] "And plans are in the works to create a rubric for evaluating and incentivizing teaching reproduc
## [4] "If anyone has seen @dstephenlindsay please tell him his car is here at the Holiday Inn. #SIPS201
## [5] "So sad to be leaving #SIPS2018 but had such a wonderful time! Already looking forward to next ye
## [6] "I've been to a bunch of psychology conferences, but I've never been so impressed by so many peop
```

```
# split the tweets
tweets_split <- strsplit(content, " ")
together <- c() # This makes our empty list we are going to append each word to

# Append every word to our list using the lappend function from above
for (row in tweets_split){
  for (word in row){
    together <- lappend(together, word)
  }
}
```

Now, this is what we end up with in our "together" list:

```
head(together, n=25)
```

```
##  [1] "Almost"                "too"
##  [3] "tired"                 "from"
##  [5] "a"                     "long,"
##  [7] "productive"            "conference"
##  [9] "to"                    "do"
## [11] "more"                  "social"
## [13] "things"                "except"
## [15] "to"                    "thank"
## [17] "everyone"              "at"
## [19] "#SIPS2018..."            "https://t.co/DQOnXPlmFP"
## [21] "The"                   "list"
## [23] "goes"                  "on"
## [25] "and"
```

Next is a bit more cleaning, but this time on just the individual words. I'm gonna remove emojis, make every word lower case, remove stopwords, remove punctuation (including hashtags), and then make it a dataframe.

```
# Deletes all non-alpha-numeric characters
together <- iconv(together, 'ASCII', 'UTF-8', sub='')
# Makes everything lower case
together <- tolower(together)
# Brings in our dictionary of stopwords from the TM package
stopwords <- stopwords('en')
# Removes any element from "stopwords" from our list
together <- removeWords(together, stopwords)
# Removes punctuation
together <- removePunctuation(together)
```

```
# Turns out newly cleaned list into a single column DF
together <- as.data.frame(together)
```

Visualizing the tweets

Most frequent words

Now that we have a nice clean data set, I'll pull out the info that we want and visualize it! In the first line, I take our new dataframe of the words and then count each individual word.

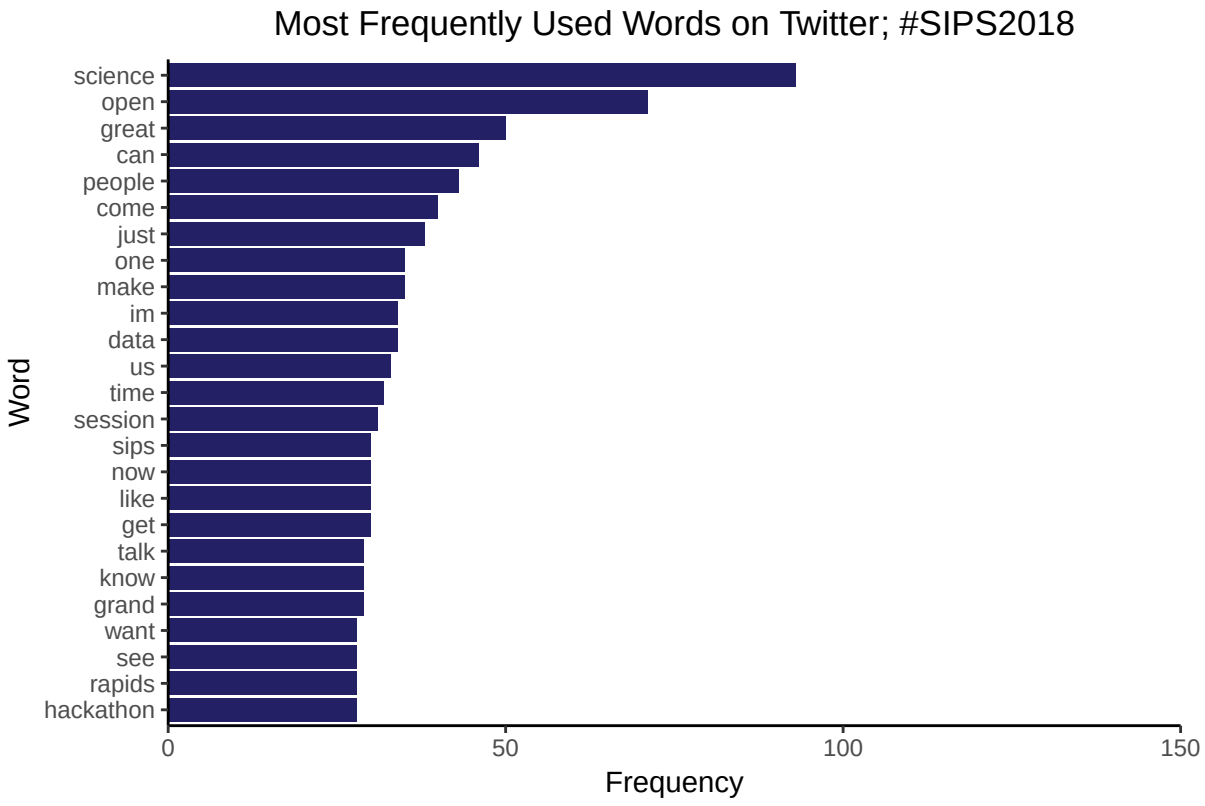
The next line orders the new dataframe in reverse, so the most frequently used words are right at the top. The next line, I delete the top two because the first is just nothing, and the second is #SIPS2018 - which is obviously present in every single tweet. So it's not all that informative.

Then, I just trim the dataset so it's only the top 25 tweets.

Then, I just make a plot using ggplot!

```
counts <- count(together$together)
counts <- counts[order(-counts$freq), ]
counts <- counts[-c(1, 2), ]
counts <- head(counts, n = 25)

sips_words <- ggplot(counts, aes(x = reorder(x, freq), y = freq)) +
    geom_bar(stat = "identity", fill = "#232066") + coord_flip() +
    labs(x = "Word", y = "Frequency") + ggtitle("Most Frequently Used Words on Twitter; #SIPS2018") +
    theme_bw() + theme(panel.border = element_blank(), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"),
    plot.title = element_text(hjust = 0.5), plot.margin = unit(c(0.5,
        0.5, 0.5, 0.5), "cm")) + scale_y_continuous(expand = c(0,
    0), limits = c(0, 150))
sips_words
```

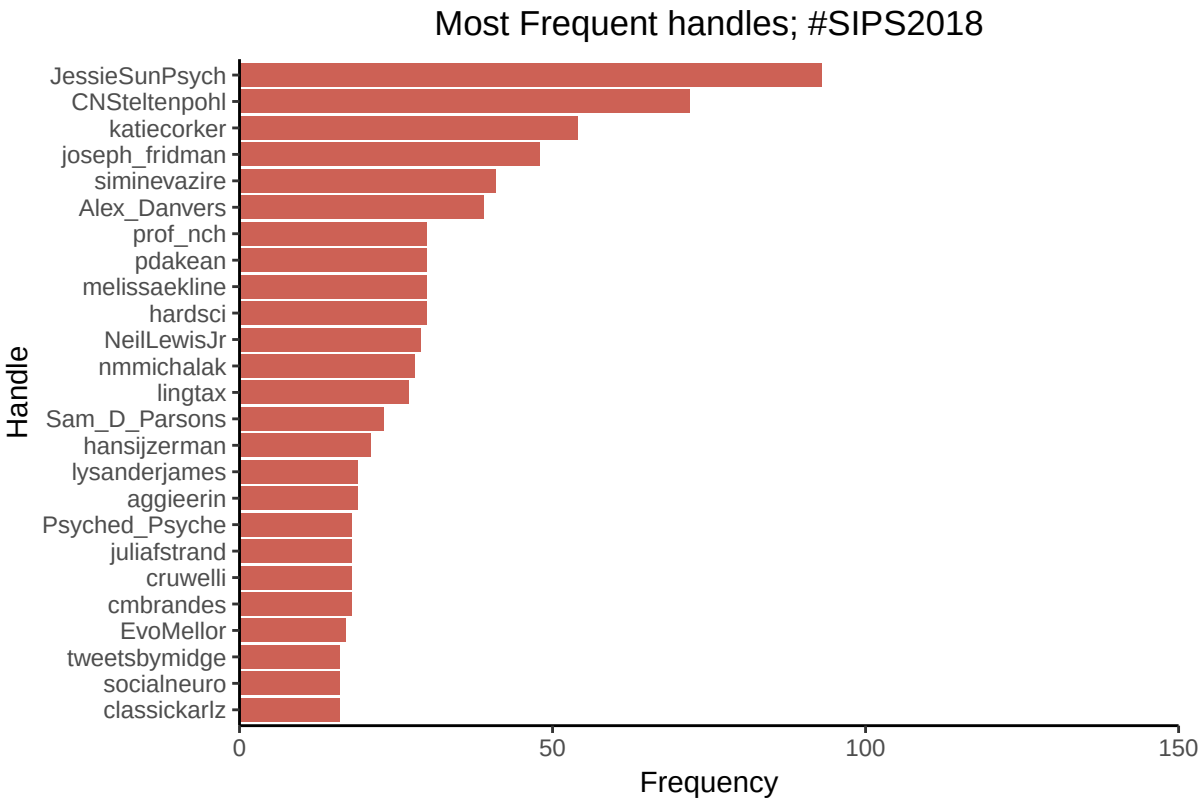


Most frequent tweeters

In the next block I do all the same as above, but just with twitter handles!

```
name_counts <- count(df$screenName)
name_counts <- name_counts[order(-name_counts$freq), ]
name_counts <- head(name_counts, n = 25)

sips_names <- ggplot(name_counts, aes(x = reorder(x, freq), y = freq)) +
    geom_bar(stat = "identity", fill = "#CD6155") + coord_flip() +
    labs(x = "Handle", y = "Frequency") + ggtitle("Most Frequent handles; #SIPS2018") +
    theme_bw() + theme(panel.border = element_blank(), panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"),
    plot.title = element_text(hjust = 0.5), plot.margin = unit(c(0.5,
        0.5, 0.5, 0.5), "cm")) + scale_y_continuous(expand = c(0,
    0), limits = c(0, 150))
sips_names
```



The most 'impactful' tweets

Looking at the most liked tweets, it does not make a ton of sense to visualize, as the content of the tweet is important.

So, in this small little block I just make a new little dataset ordered by most to least number of likes and then print the top 5!

```
like_counts <- og_tweets[order(-og_tweets$favoriteCount),]
head(like_counts$text, n=5)
```

```
## [1] "Everyone makes mistakes during data analysis. Literally everyone. The question is not what erro
## [2] "If you care about research integrity, open science, and reproducibility, then follow #SIPS2018 
## [3] "Sorry to miss #SIPS2018 this time, but the other life priority won this time. Get some good wor
## [4] "On one hand, wishing I could be at #SIPS2018. On the other hand, I'm excited to announce that I
## [5] "Nine years ago @jinxgoh was a sophomore in my Intro Stats class at @BardCollege. Now the studen
```

And that's the basics of twitter scraping, with the SIPS2018 hashtag!