Collecting, Manipulating and Blending Data from Twitter

Cameron Tomko

Problem 1: Sampling Twitter Data with Streaming API about a certain topic

- Select a topic that you are interested in, for example, "#WPI" or "#DataScience"
- Use Twitter Streaming API to sample a collection of tweets about this topic in real time. (It would be recommended that the number of tweets should be larger than 50, but smaller than 500.
- Store the tweets you downloaded into a local file (csv file)

```
# Load the tweets from file
tweetsDF = read.csv("C:\\Users\\camer\\OneDrive\\Documents\\WPI School\\2022-2023\\Spring 23\\DS 501\\H
tweets = data.frame(tweetsDF)
```

```
library(twitteR)
library(stringr)
setup_twitter_oauth(consumerKey, consumerSecret, accessToken, accessTokenSecret)
tweets = searchTwitter('#rstats', n=50)
tweetsDF = twListToDF(tweets)
```

- The topic of interest: < R >
- The total number of tweets collected: < 500 >

The topic of data that I decided to collected was on R, which is the current program language that is being used in the class. I have used R Studio in previous classes and the programming language is very relevant in my field of study, Data Science. The twitter data set has 500 rows of data across 17 columns. As I would not think R Studio would be an trending topic on Twitter, I was pleasantly surprised to see that R Studio had a large Twitter presence. To analyze the data set, I created new data frames, new fields of the original data frame, and utilized different packages to clean the data and count the frequencies of words, hashtags, and mentions.

Problem 2: Analyzing Tweets and Tweet Entities with Frequency Analysis

1. Word Count:

- Use the tweets you collected in Problem 1, and compute the frequencies of the words being used in these tweets.

In this section, I analyze the the most frequent words within the data set. To start, I needed to clean the data to ensure only words were available to be counted in the frequencies. To do this, I did preprocessed the data. This meant removing any punctuation, links to articles, pictures, or videos, hashtags, mentions, extra white spaces, and numbers. To visualize the frequencies of the words, I stored the words and their frequencies in a data frame and called the data frame to see the most popular terms. To calculate the frequencies, I used

document term frequency to create a data frame, which made a one row matrix that had all of the available words as columns. From there, counts for each word were done and this created the data frame of words and corresponding frequencies. This was the complete list of all terms and their corresponding frequencies. From here, I ordered the data based on the frequency of the words. The data set has 1,039 unique terms after cleaning. The range of word frequencies is from 132 to 1. I only displayed 75 terms because I felt like showing every term was excessive and unnecessary for the assignment. It also took me over the 10 page limit.

```
library(tm)
```

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

```
#cleaning the tweets
tweets_text <- tweets$text
tweets_text <- tolower(tweets_text)
tweets_text <- gsub("rt","", tweets_text)
tweets_text <- gsub("@\\w+","", tweets_text)
tweets_text <- gsub("https?://.+","", tweets_text)
tweets_text <- gsub("\\d+\\w*\\d*","", tweets_text)
tweets_text <- gsub("#\\w+","", tweets_text)
tweets_text <- gsub("[^\x01-\x7F]","", tweets_text)
tweets_text <- gsub("[[:punct:]]","", tweets_text)

# creating a new column in the tweets data frame for the cleaned data
tweets["cleaned_text"] <- tweets_text
# transforming the data frame to a corpus
tweets_corpus <- Corpus(VectorSource(tweets$cleaned_text))
# use Document Term Matrix to find the count for each word
tweets_dtm <- DocumentTermMatrix(tweets_corpus)

# creating a data frame to store the word frequencies
# which is sorted from highest to lowest
word_freq_tweets <- data.frame(sort(colSums(as.matrix(tweets_dtm)), decreasing=TRUE))
colnames(word_freq_tweets) <- c('Top Frequent Words')
head(word_freq_tweets,75)
```

```
##             Top Frequent Words
## the                        132
## with                        89
## cheat                       60
## sheet                       60
## and                         58
## useful                      57
## for                         55
## basic                       52
## expressions                 52
## ian                         52
## kopacka                     52
## regular                     52
## using                       50
## multiple                    46
## data                        44
## sociais                     44
```

```
## how                43
## one                43
## join               42
## mtodos             42
## dataframes         40
## dplyr              39
## tip                39
## are                31
## new                30
## package            29
## learning           27
## all                26
## you                22
## workshop           22
## about              22
## this               21
## believe            21
## can                21
## imputations        21
## amp                21
## packages           18
## from               17
## para               17
## into               16
## materials          15
## cran               15
## get                14
## wrote              14
## has                14
## that               13
## here               13
## post               13
## ggplot             13
## intro              13
## deep               12
## version            12
## blog               12
## cincia             12
## poltica            12
## dos                12
## tutorial           11
## make               11
## more               11
## metodologia        11
## cantinho           11
## couldnt            10
## yesterday          10
## time               10
## santos             10
## thiago             10
## online             10
## other              10
## fun                10
## history             9
```

```
## work                        9
## este                        9
## via                         9
## https                       9
## analysis                    9
```

- Display a table of the top 30 words (ONLY) with their counts

After calculating the frequencies for all of the words, I cut down the 1,032 word down to the top 30 words based on frequency. This was done by calling the previously sorted dataframe and using the head() function to take the first 30 rows of the dataframe. When looking at the frequencies of word, it was interesting to see smaller words with less meaning like 'the', 'with', 'and', and 'for' at the top of the list. Ignoring these terms, there is insight that can be gained as many people are searching for cheat sheets for R as 'cheat' appear 63 times and 'cheat' appeared 60 times; it does seem like these words often occur in the same tweet. Two words that confused me at first glance were "ian" and "kopacka" as I had never heard of these terms in R, nonetheless any other programming languages After further research, I found that these two terms were referencing the name of a person, Ian Kopacka, who has helped create cheat sheets for R Studio. I found this very fascinating that his name occurred 52 times in the span of 500 tweets.

```r
# displaying the top 30 words in the data frame table
colnames(word_freq_tweets) <- c('Top 30 Frequent Words')
head(word_freq_tweets, 30)
```

```
##             Top 30 Frequent Words
## the                           132
## with                           89
## cheat                          60
## sheet                          60
## and                            58
## useful                         57
## for                            55
## basic                          52
## expressions                    52
## ian                            52
## kopacka                        52
## regular                        52
## using                          50
## multiple                       46
## data                           44
## sociais                        44
## how                            43
## one                            43
## join                           42
## mtodos                         42
## dataframes                     40
## dplyr                          39
## tip                            39
## are                            31
## new                            30
## package                        29
## learning                       27
## all                            26
## you                            22
## workshop                       22
```

2. Find the most popular tweets in your collection of tweets

- Please display a table of the top 10 tweets (ONLY) that are the most popular among your collection, i.e., the tweets with the largest number of retweet counts.

The next section I looked at was the most popular tweets. The paper defined the most popular tweets as ones with the largest number of retweet counts. To find the top 10 most popular, I sorted the retweetCount column in the dataset. After I returned a final dataframe, I saw that 2 tweets appeared multiple times in the top 10 with the same retweetCount. To bypass this, I decided to only look at unique tweets so that the top 10 would not return any duplicate tweets. The top 10 retweeted tweets are shown below. I would also like to note that the highest number of retweetCount was 811 and the next highest was only 237 retweets, which is a large difference and emphasizes the importance of that tweet within the R Studio community. There were also a large number of tweets that had 112 retweets. It was interesting to look at as some were duplicates, but others were different. I guess this was just a coincidence.

```
# sort the tweets data frame by the retweet count
retweet_sorted <- tweets[order(tweets$retweetCount, decreasing = TRUE),]
# return top 10 unique tweets based on retweet count
top10_retweet <- data.frame(head(unique(retweet_sorted[,c("text")]), 10))
# renaming the column of top 10 tweets
colnames(top10_retweet) <- c('Top 10 Retweeted Tweets')
top10_retweet
```

```
##
## 1   RT @ClausWilke: Over the years, movies have converged to a length of ~100 min.  4 lines of code
## 2   RT @danielphadley: Add logos and gifs to plots in #rstats : https://t.co/i40eL4EbP3, or, Vincent
## 3                        RT @Rbloggers: ggplot2 - Easy way to mix multiple graphs on the same page h
## 4   RT @dataandme:  useful cheat sheet: "Basic Regular Expressions in R" by Ian Kopacka https://t.co/
## 5  RT @tjpalanca: "The point being that media isn't biased in that your timeline is." #rstats #datab
## 6  RT @Rbloggers: Machine Learning Explained: supervised learning, unsupervised learning, and reinfo
## 7  RT @rOpenSci: [blog] Announcing the rOpenSci Fellowships Program https://t.co/4lgCMUR0yQ Applicat
## 8  RT @dsquintana: New post: An #Rstats script to calculate statistical power for a random-effects m
## 9  RT @R_Programming: New Grand Test added to 'Learn R By Intensive Practice' video course #rstats h
## 10                RT @R_Programming: R Tip: How to join multiple dataframes in one go using dplyr. #
```

3. Find the most popular Tweet Entities in your collection of tweets

Please display a table of the top 10 hashtags (ONLY), top 10 user mentions (ONLY) that are the most popular in your collection of tweets.

In this section, I looked at the top 10 hashtags and the top 10 mentions from the data set. For both sets of data, I extracted all of the contents that started with a '#' or '@' and added them to a new data frame by separating the hashtags and mentions as individual terms, meaning they did not have any association to the rows which they stemmed from. From this point, I was able to count the frequencies of the hashtags and mentions and do a cutoff for the top 10. For the hashtags, there were three different variations of #rstats, ranking at #1, #5, and #8, so I decided to remove the uppercase letters and convert all characters of the hashtags to lowercase. The placement of #rstats was still at #1, but there was a more even distribution of the dataset being shown in the top 10 as #rstats did not take up three places. For the top 10 mentions, I anticipated to see @IanKopacka on the list as the name had two places in the top 30 of the entire data set's word frequency, but this was not the case as he did not appear. The range of the top 10 of mentions did fall off quickly as the mentions ranged from 60 mentions to 8 mentions.

```r
#top 10 hashtags
library(stringr)

#extract all hashtags from the text data
tweets_hashtag <- str_extract_all(tweets$text, "#\\w+")
# disconnect hashtags from each other to make it easier to count
tweets_hashtag <- unlist(tweets_hashtag)
# turning the hashtags to lowercase
tweets_hashtag <- tolower(tweets_hashtag)
# turn data into a data frame
hashtag_count <- data.frame(table(tweets_hashtag))
# count number of hashtags and order by highest count
sorted_hashtag_count <- hashtag_count[order(hashtag_count$Freq, decreasing = TRUE),]
# take top 10 of hashtag frequencies of dataframe
top10_hashtags <- head(sorted_hashtag_count[, c("tweets_hashtag")],10)
# create new dataframe for top 10
top10_hashtags_df <- data.frame(top10_hashtags)
# display dataframe
top10_hashtags_df
```

```
##      top10_hashtags
## 1           #rstats
## 2      #datascience
## 3            #regex
## 4           #sods17
## 5  #machinelearning
## 6          #esa2017
## 7          #bigdata
## 8          #dataviz
## 9              #ai
## 10           #rlang
```

```r
#top 10 mentions
library(stringr)

#extract all mentions from the text data
tweets_mentions <- str_extract_all(tweets$text, "@\\w+")
# disconnect mentions from each other to make it easier to count
tweets_mentions <- unlist(tweets_mentions)
# turn data into a data frame
mention_count <- data.frame(table(tweets_mentions))
# count number of mentions and order by highest count
sorted_mention_count <- mention_count[order(mention_count$Freq, decreasing = TRUE),]
# take top 10 of mention frequencies of dataframe
top10_mentions <- head(sorted_mention_count[, c("tweets_mentions")],10)
# create new dataframe for top 10
top10_mentions_df <- data.frame(top10_mentions)
# display dataframe
top10_mentions_df
```

```
##     top10_mentions
## 1       @dataandme
## 2       @Rbloggers
```

```
## 3   @R_Programming
## 4         @ucfagls
## 5     @gp_pulipaka
## 6        @noamross
## 7        @naupakaz
## 8      @bhaskar_vk
## 9        @DataCamp
## 10 @ScientistTrump
```

Problem 3 (Optional): Explore the data

- Run some additional experiments with your data to gain familiarity with the twitter data and twitter API

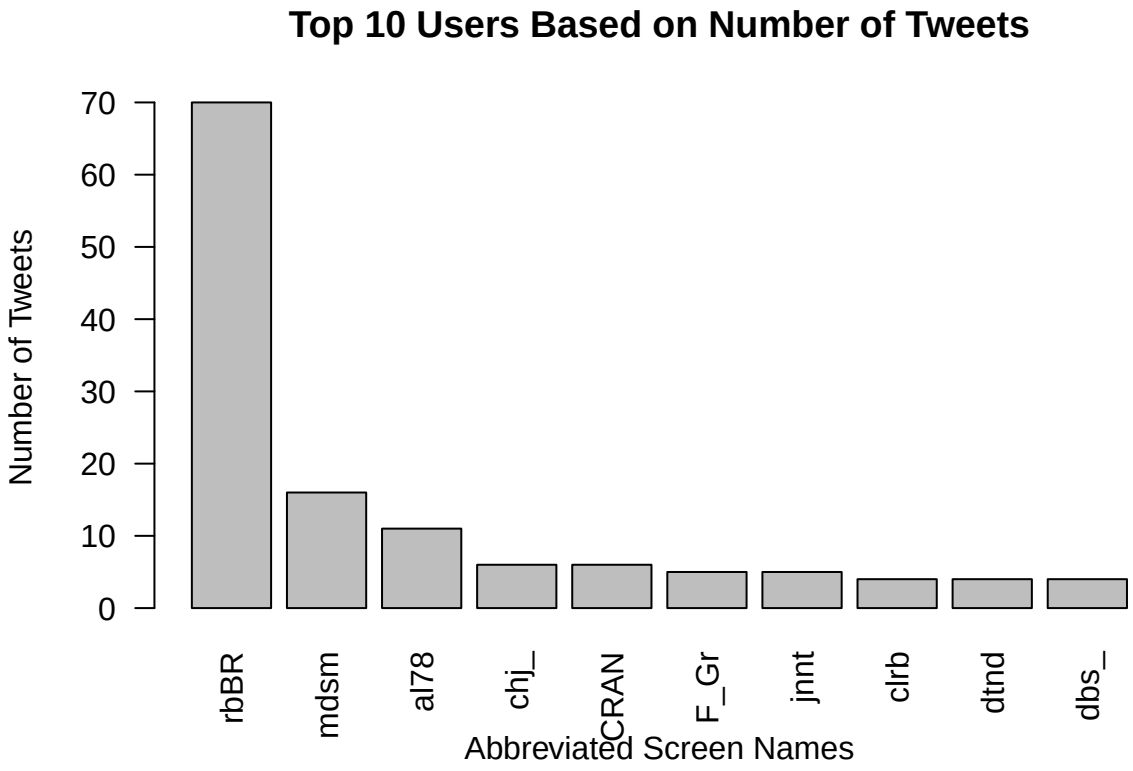
The last topic I looked at was to see the frequency that users have tweeted during the span of the data collection. The first tweet collected was at 2017-08-05 23:48:24 and the last collected was at 2017-08-06 20:25:32; this spans just a couple hours short of a full day. To count the frequency that a user tweets, I used the screen name of a user to show which user is tweeting. From that, I was able to sort the top 10 users and plot them through a bar plot. The user screen names have been abbreviated to allow them to fit as x axis labels. I expected to see that there would be no outliers to the dataset as the span is only roughly a day. To my surprise, there was one user, @rbloggersBR, that tweeted 70 times during that span. Furthermore, after looking at the user's tweets, I noticed that all of the tweets were in Spanish, which was an interesting takeaway and explained why I did not recognize a few of the top 30 most frequent terms.

```
# top users based on the number of tweets which is displayed in a bar chart

# getting the screen names
screen_name <- tweets$screenName
# creating a dataframe for the screen names and counts
screen_name_df <- data.frame(table(screen_name))
# count screen name and order by highest count
screen_name_count <- screen_name_df[order(screen_name_df$Freq, decreasing = TRUE),]
screen_name_top_10 <- head(screen_name_count,10)
screen_name_top_10
```

```
##         screen_name Freq
## 238     rbloggersBR   70
## 196        mdsumner   16
## 6      alevergara78   11
## 52           chj_vc    6
## 59  CRANberriesFeed    6
## 97         F_Gergis    5
## 160      jonintweet    5
## 57      clairebotai    4
## 64        dataandme    4
## 70   debashis_dutta    4
```

```
#create the bar chart
barplot(height = screen_name_top_10$Freq,
        main = "Top 10 Users Based on Number of Tweets",ylab = 'Number of Tweets', xlab = 'Abbreviated S
        names.arg = abbreviate(screen_name_top_10$screen_name),
        las = 2)
```


Top 10 Users Based on Number of Tweets
Number of Tweets
0
10
20
30
40
50
60
70
rbBR
mdsm
al78
chj_
CRAN
F_Gr
jnnt
clrb
dtnd
dbs_
Abbreviated Screen Names