# post02: Scraping Social Media Data

*Karl Walter*

## Introduction

In Stat 133 we have reviewed and explored the power of web scraping. Packages such as XML, xml2, and rvest – all of which were used in lab12 – have helped to simplify and expedite the process of web scraping.

But, what are the other libraries and techniques available to the R community to aid in web scraping? If you want to get the most recent trends from social media sites (Twitter, Instagram, Facebook) what is the best way to go about that?

There is a package 'SocialMediaLab' which allows the user to explore trends through Facebook, Twitter, Instagram, and YouTube data.

In this blog post, we will explore how to set up and use the SocialMedia Lab package. More specifically, we will be exploring its power through looking at trends within Tweets that contain the keyword @nbcsnl

## Setting Up

Before jumping into the syntax and examples of SocialMediaLab, we need to go through how to get everyting set up.

### (1) Install the Package

Use the command **'install.packages('SocialMediaLab')'** at the command line.

```
library(SocialMediaLab)
```

```
## Warning: package 'SocialMediaLab' was built under R version 3.4.2
```

### (2) Set Up API Login

In order to access the data from the social media websites you want to explore you need to be tied to a particular account. In order to set up this connection, follow the directions presented in the following article.

That article includes the following steps: (1) Create an app at Twitter

2. On your Twitter app profile, under "My Applications" click "Create new Application." Set the callback URL as http://127.0.0.1:1410. For website URL, you may simply put the homepage of your github repo.
3. Now you are ready for authentication. Note: These values are unique to MY authentication and will not work for you. You will have to click a button under the "Keys and Access Tokens" tab which will generate your access token and access token secret.

```
api_key <- "sbi0yibdsDLKgvO17Ci8wwmzB"
api_secret <- "NMZ5ygI2jiL0K7XSIghLAfNzSzkmUm4onGTlgEBQzKO8qqVn1y"
access_token <- "339293748-5vXetSfDjEJ8glCVOfRjjQAOPdKVUGQxmThVW0gr"
access_token_secret <- "hfkW5MPIHbjIWioFHErGP431V3QlvtYF22WtbFPjw5wBU"

AuthenticateWithTwitterAPI(api_key, api_secret, access_token,
             access_token_secret)
```

```
## [1] "Using direct authentication"
```

```
## NULL
```

Now run a simple check to ensure that your authentication has worked. Here we will collect 5 Tweets, and show the text of the first.

```
st <- "@nbcsnl"
nt <- 5
lan <- "en"
dat <- CollectDataTwitter(searchTerm = st, numTweets = nt, language = lan)
```

```
## Now retrieving data based on search term: @nbcsnl
## Done
## Cleaning and sorting the data...
## Done
```

```
head(dat$text, 1)
```

```
## [1] "RT @Trollin_Trump: Apparently @realDonaldTrump hates the fact a black woman played him on @nbcsnl. So plea
se do your civic duty and RT theÃ¢â<U+0082>¬Â¦"
```

## The Power of Social Media Lab: Exploring via Twitter

### Get Set Up

With SocialMedia Lab you can look at the text, authors, times, geolocations (when available) and other data of the tweets you collect. Let's run through a few examples. For each example we will use the following data set.

```
# Collect 1500 tweets that mention the Saturday Night Live handle (@nbcsnl).
st <- "@nbcsnl"
nt <- 1500
lan <- "en"
dat <- CollectDataTwitter(searchTerm = st, numTweets = nt, language = lan)
```

```
## Now retrieving data based on search term: @nbcsnl
## Done
## Cleaning and sorting the data...
## Done
```

```
# Here are the following columns within the collected data
print("Twitter Data Columns")
```

```
## [1] "Twitter Data Columns"
```

```
print(colnames(dat))
```

```
##  [1] "text"             "favorited"       "favoriteCount"
##  [4] "replyToSN"        "created_at"      "truncated"
##  [7] "replyToSID"       "id"              "replyToUID"
## [10] "statusSource"     "screen_name"     "retweetCount"
## [13] "isRetweet"        "retweeted"       "longitude"
## [16] "latitude"         "from_user"       "reply_to"
## [19] "users_mentioned" "retweet_from"    "hashtags_used"
```

Now let's look at what each of these columns gives us.

### DATA COLUMNS

**text:** text of the tweet [character]

**favorited:** if tweet was favorited [logical]

**favoriteCount:** number of times tweet was favorited [double]

**replyToSN:** If the tweet is a reply, this is the handle (w/o '@') of who the tweet is replying to. If it is not a reply, it is NA. [character]

**created_at:** time (UTC) at which tweet was tweeted [character]

**truncated:** if the tweet was truncated [logical]

**replyToSID:** If the tweet is a reply, this is the numeric id of the tweet it is replying to. If it is not a reply, it is NA.

**id:** the tweet's numeric id [character]

**replyToUID:** If the tweet is a reply, this is the numeric id of who the tweet is replying to. If it is not a reply, it is NA. [character]

**statusSource:** URL of the status [character]

**screen_name:** the author's twitter handle (w/o '@') [character]

**retweetCount:** number of retweets the tweet received [double]

**isRetweet:** if the tweet is a retweet [logical]

**retweeted:** if the tweet was retweeted [logical]

**longitude:** This is the longitude of the tweet's geolocation. It is NA if there is no geolocation [character]

**latitude:** This is the latitude of the tweet's geolocation. It is NA if there is no geolocation [character]

**from_user:** the author's twitter handle (w/o '@') – the same as 'screen_name'. [character]

**reply_to:** If the tweet is a reply, this is the handle (w/o '@') of who the tweet is replying to. If it is not a reply, it is NA – the same as 'replyToSN'. [character]

**users_mentioned:** users mentioned in the tweet [character]

**retweet_from:** who the user retweeted. NA if not a retweet [character]

**hashtags_used:** hashtags used in tweet [character]

## Examples

### Example 1: The most favorited tweet.

```
mostfaved <- dat[dat$favoriteCount == max(dat$favoriteCount),]
print(paste("Number of Favorites: ", mostfaved$favoriteCount))
```

```
## [1] "Number of Favorites:  83"
```

```
print(paste("Author of Tweet: " , mostfaved$screen_name))
```

```
## [1] "Author of Tweet:  playbill"
```

```
print(paste("Text of Tweet: ", mostfaved$text))
```

```
## [1] "Text of Tweet:  You have to watch this sneak peek of The Carol Burnett 50th Anniversary Special, with @Off
icialBPeters, @KChenowethÃ¢â<U+0082>¬Â¦ https://t.co/xsAPPf4sdK"
```

And, we can actually find this tweet (with a simple search). Here it is! Note: The most favorited tweet may change by the time you run this code.

### Example 2: The most retweeted tweet.

```
mostrt<- head(dat[dat$retweetCount == max(dat$retweetCount),], 1)
print(paste("Number of Retweets: ", mostrt$retweetCount))
```

```
## [1] "Number of Retweets:  106077"
```

```
print(paste("Author of Tweet: " , mostrt$screen_name))
```

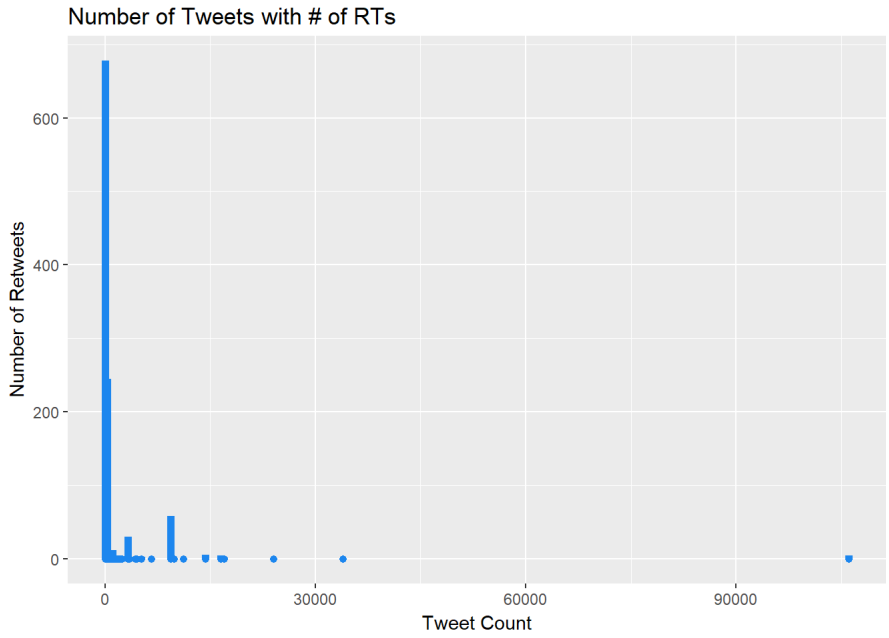
```
## [1] "Author of Tweet:  zaynsmol"
```

```
print(paste("Text of Tweet: ", mostrt$text))
```

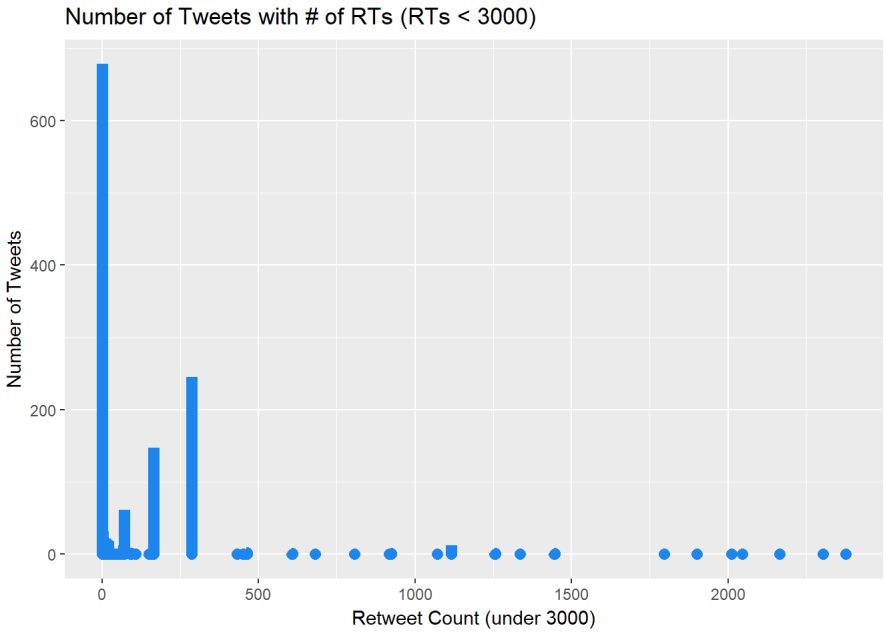
```
## [1] "Text of Tweet:  RT @Harry_Styles: Always lovely being at @nbcsnl \nShould be a funny night. Hope you enjoy
.x"
```

### Example 3: Breaking down Number of Retweets

```
# Barplot of the breakdown of the number of retweets
ggplot(data = dat, aes(retweetCount)) +
  geom_bar(color = 'dodgerblue2', size = 2) +
  ggtitle("Number of Tweets with # of RTs") +
  labs(x = "Tweet Count", y = "Number of Retweets")
```




```
# Zoom in on plot (where retweetCount < 3000)
datnew <- dat[dat$retweetCount < 3000,]
ggplot(data = datnew, aes(retweetCount)) +
  geom_bar(color = 'dodgerblue2', size = 3) +
  ggtitle("Number of Tweets with # of RTs (RTs < 3000)") +
  labs(x = "Retweet Count (under 3000)", y = "Number of Tweets")
```




## Example 4: Find the location of a tweet!

Not many tweets are associated with a location, but there are a few. Let's try to see if we can pinpoint the location of a tweet.

```
latitude <- dat[! is.na(dat$latitude),]['latitude'][[1]]
longitude <- dat[! is.na(dat$latitude),]['longitude'][[1]]
print.noquote("COORDINATES ")
```

```
## [1] COORDINATES
```

```
print.noquote(paste("Longitude: ", longitude))
```

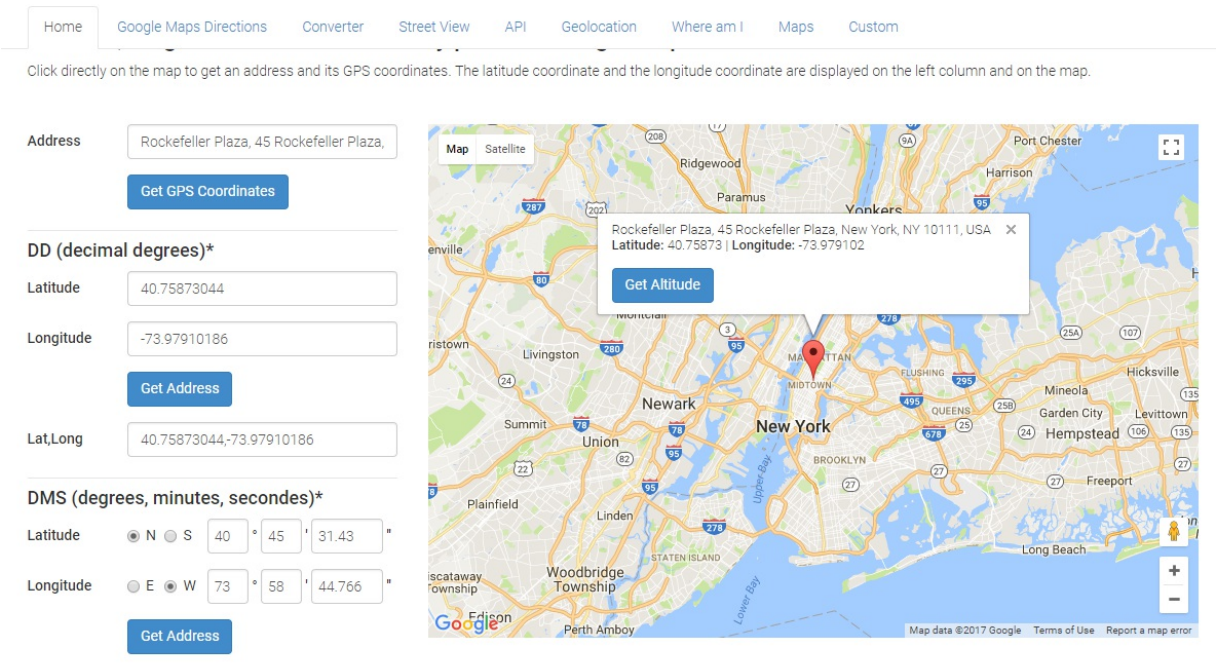
```
## [1] Longitude:  -73.97910186
```

```
print.noquote(paste("Latitude: ", latitude))
```

```
## [1] Latitude:  40.75873044
```

Now, let's look this myster location up using the following online mapper.

Here is a screenshot of the website:




Rockefeller Plaza Found with Lat and Lon of Tweet

CHECK IT OUT!! The one tweet that had assigned latitude and longitude coordinates was sent from Rockefeller Plaza, which is where Saturday Night Live (recall that our keyword was @nbcsnl) is shot!

# Creating a general "description" function

What were the other characteristics of this tweet we were able to track the location of? It seems that it has been pretty common to want to know the author, number of retweets, and text of a tweet. So let's create a single function that we can call in order to instantly print out some quick stats about a tweet.

```
description <- function(df_row) {
  print(paste("Author of Tweet: " , df_row$screen_name))
  print(paste("Number of Retweets: ", df_row$retweetCount))
  print(paste("Text of Tweet: ", df_row$text))
}
```

Now let's call the function we created to learn more about the mystery tweet located at Rockefeller Plaza.

```
description(dat[! is.na(dat$latitude),])
```

```
## [1] "Author of Tweet:  marywilesmakeup"
## [1] "Number of Retweets:  0"
## [1] "Text of Tweet:  So looking forward to watching #saoirseronan host @nbcsnl tonight hair been a great week g
ettingÃ¢â<U+0082>¬Â¦ https://t.co/MTp5aniEtb"
```

After a quick google search we find that this is the tweet of '@marywilesmakeup', who after a little bit of digging (this part is less stats related, but randomly super interesting) is actually a MAKEUP ARTIST FOR SNL.

Here is an Instagram post of her getting this week's (12/2) host, Saoirse Ronan, ready for the show.

# Wrapping Up

In this blog we…

- Introduced a new, powerful, web scraping tool (SocialMediaLab)
- Went through the basic functions it provides
- Explored its functionality via a few examples
    1. Finding the most favorited tweet
    2. Finding the most retweeted tweet
    3. Looking at the Retweet statistics
    4. Found the location of a tweet
- Developed a 'description' function which prints basic info about a selected row
- Found that our Example #4 tweet was actually a makeup artist at SNL!

**To the person grading this, have a great rest of Dead Week, and good luck on Finals!**