# rtweet Exploration

Matthew Hendrickson

4/30/2019

## rtweet

rtweet (https://rtweet.info/ (https://rtweet.info/)) is a R package that makes interacting with Twitter easy. There are a few key interactions that you can perform with rtweet (per the rtweet site (https://rtweet.info/))

1. Search Tweets
2. Stream Tweets
3. Get Friends
4. Get Timelines
5. Get Favorites
6. Search Users
7. Get Trends
8. Post Actions

This document explores Tweets from specific users and compares activity across those users.

## Setup

Run this if you haven't installed the required packages.

```
# Only needed if you haven't arealdy installed the packages
#install.packages(c("rtweet", "devtools", "tidyverse", "gridExtra", "lubridate", "kableExtra"))
# Needed to load the packages
library("rtweet")
library("devtools")
library("tidyverse")
library("gridExtra")
library("lubridate")
library("kableExtra")
library("scales")
```

## Connecting to Data

You'll need to set up and authorize the Twitter API. This is explained here (https://rtweet.info/articles/auth.html).

Once you've completed this task, connect to the Twitter API. The Twitter API limits the number of search results to 18.000 every 15 minutes. Keep this in mind if you're pulling a large amount of data. The rtweet package can help manage this with `retryonratelimit = TRUE` . This command will help manage the search results in accordance with the Twitter API.

Note that I've used `rstudioapi::askForSecret` . This prompt the user to manually type or copy their credentials into R Studio.

```
appname <- rstudioapi::askForSecret("Twitter App Name") # name of twitter app
key     <- rstudioapi::askForSecret("API Key") # api key
secret  <- rstudioapi::askForSecret("API Secret") # api secret
token   <- create_token(app = appname,
                        consumer_key = key,
                        consumer_secret = secret)
```

## Getting Specific Users

Let’s pull tweet data for a few users to create a comparison. Let’s begin with the R for Data Science community’s Tweets Slack community sign up (bit.ly/R4DSslack), based on this text (https://r4ds.had.co.nz/). This community is great for anyone wanting to learn more R. has a great learning community, let’s begin with the R4DS handle’s Tweets.

Let’s see how many records we need to pull to get the R4DScommunity timeline to ensure we don’t approach the 18,000 results limit. First, we’ll pull the most recent record and get the value in the `statuses_count` field. The number of statuses is the number of Tweets the account has posted.

```
# See how many times R4DScommunity has Tweeted
r4ds_count <- get_timeline(c("R4DScommunity"), n = 1) # Get most recent record
r4ds_count$statuses_count # Get number of Tweets (statuses) = 2843 as of 2019/05/02
```

```
## [1] 2844
```

```
# rm(r4ds_count) # Removes r4ds_count dataframe
```

We should have no issue with the 18,000 result limit. I’ve included `retryonratelimit = TRUE` here to illustrate how that looks, though it is unnecessary. Let’s now pull all Tweets for the R4DScommunity handle. Adjust the `n = 5000` if necessary depending on the results of the previous call.

```
# Get R4DScommunity's Tweets
r4ds <- get_timeline(c("R4DScommunity"), n = 5000, retryonratelimit = TRUE)
# Adjust datestamp to EST from UTC
r4ds$created_at_est <- r4ds$created_at - 18000 # 5 hours in seconds
# Check adjustment
head(select(r4ds, created_at, created_at_est))
```

```
## # A tibble: 6 x 2
##   created_at          created_at_est
##   <dttm>              <dttm>
## 1 2019-05-03 03:33:51 2019-05-02 22:33:51
## 2 2019-05-02 20:27:52 2019-05-02 15:27:52
## 3 2019-05-02 20:27:41 2019-05-02 15:27:41
## 4 2019-05-02 20:27:34 2019-05-02 15:27:34
## 5 2019-05-02 14:21:41 2019-05-02 09:21:41
## 6 2019-05-02 14:21:34 2019-05-02 09:21:34
```

```
# Save results for future use to avoid having to call Twitter API
saveRDS(r4ds, "r4ds.rds")
```

That’s it! This gets us the R for Data Science Tweet timeline. Keep in mind the Twitter API 18,000 search results per 15 minutes if you adjust the number of records in the code above (or below).

Now, let’s get another users’ data. Feel free to swap out another handle (or even yours)!

Let’s check to make sure we won’t approach the 18,000 result limit. If we do, simply include `retryonratelimit = TRUE` to have the call pause and restart every 15 minutes.

```
mh_count <- get_timeline(c("mjhendrickson"), n = 1) # Get most recent record
mh_count$statuses_count # Get number of Tweets (statuses) = 1425 as of 2019/05/02
```

```
## [1] 1425
```

```
# rm(mh_count) # Removes mh_count dataframe
```

Again, we will have no issue with the rate limit. Let's pull all Tweets from the handle. Adjust the `n = 5000` if necessary depending on the results of the previous call.

```
# Get mjhendrickson's Tweets
mh <- get_timeline(c("mjhendrickson"), n = 5000, retryonratelimit = TRUE)
# Adjust datestamp to EST from UTC
mh$created_at_est <- mh$created_at - 18000 # 5 hours in seconds
# Check adjustment
head(select(mh, created_at, created_at_est))
```

```
## # A tibble: 6 x 2
##   created_at          created_at_est
##   <dttm>              <dttm>
## 1 2019-05-02 21:01:26 2019-05-02 16:01:26
## 2 2019-05-02 20:11:01 2019-05-02 15:11:01
## 3 2019-05-02 16:29:14 2019-05-02 11:29:14
## 4 2019-05-02 16:23:52 2019-05-02 11:23:52
## 5 2019-05-02 03:42:28 2019-05-01 22:42:28
## 6 2019-05-02 03:08:04 2019-05-01 22:08:04
```

```
# Save results for future use to avoid having to call Twitter API
saveRDS(mh, "mh.rds")
```

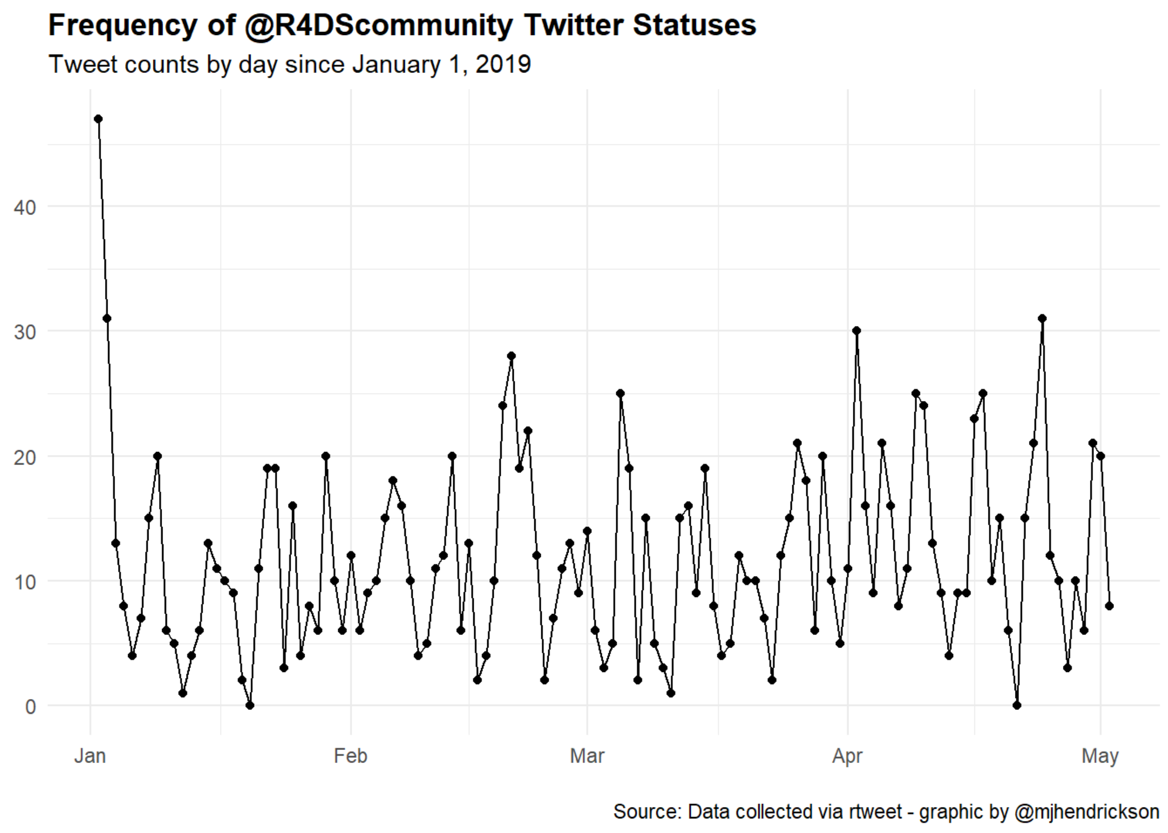
# Create Comparison Plots

Now that we've pulled together the most recent 9,000 tweets for the two accounts above, let's examine the frequency of Tweets for both.

## R For Data Science's Tweets

I'm assigning the plot to the object `r4ds_plot`, then plotting that object. I'm doing this so I can assemble multiple plots together at the end.

```
r4ds_plot <-
r4ds %>%
  filter(created_at_est >= "2019-01-01") %>% # R4DS handle started 2018-04-14
ts_plot("days", trim = 1L) +
  geom_point() +
  theme_minimal() +
  theme(
    legend.title = element_blank(),
    legend.position = "bottom",
    plot.title = element_text(face = "bold")) +
  labs(
    x = NULL,
    y = NULL,
    title = "Frequency of @R4DScommunity Twitter Statuses",
    subtitle = "Tweet counts by day since January 1, 2019",
    caption = "\nSource: Data collected via rtweet - graphic by @mjhendrickson"
  )
r4ds_plot
```


Frequency of @R4DScommunity Twitter Statuses

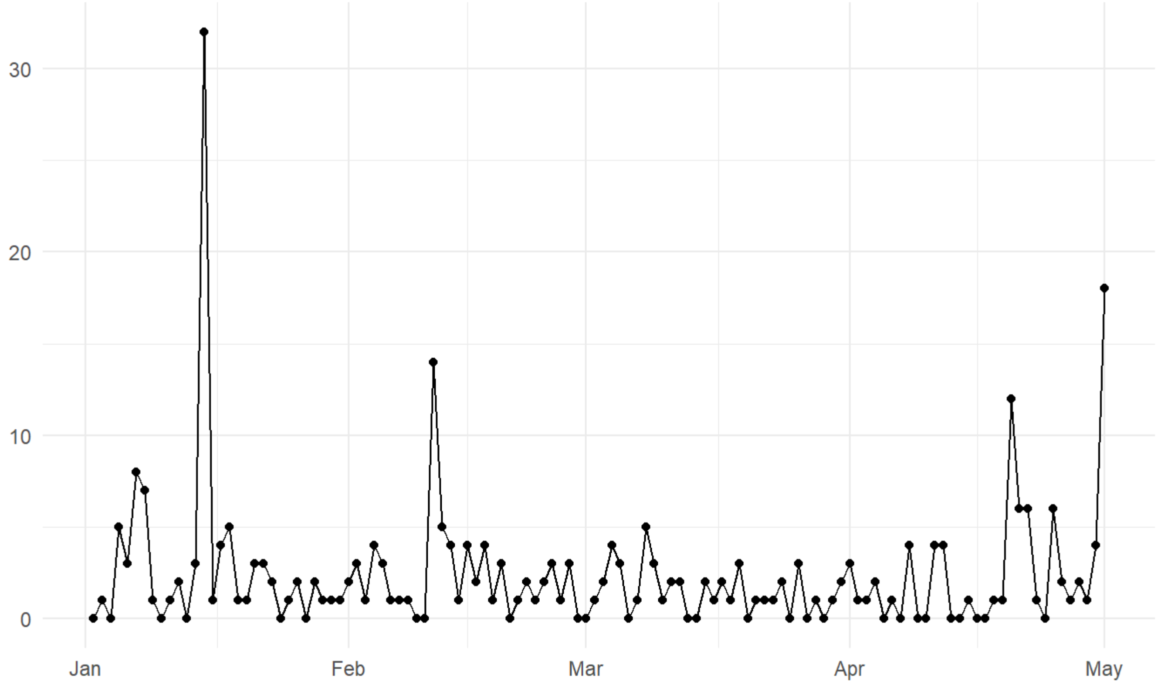
Tweet counts by day since January 1, 2019

Source: Data collected via rtweet - graphic by @mjhendrickson

## My Tweets

Now lets create the plot for the second handle.

```
mh_plot <-
mh %>%
  filter(created_at_est >= "2019-01-01") %>%
ts_plot("days", trim = 1L) +
  geom_point() +
  theme_minimal() +
  theme(
    legend.title = element_blank(),
    legend.position = "bottom",
    plot.title = element_text(face = "bold")) +
  labs(
    x = NULL,
    y = NULL,
    title = "Frequency of @mjhendrickson Twitter Statuses",
    subtitle = "Tweet counts by day since January 1, 2019",
    caption = "\nSource: Data collected via rtweet - graphic by @mjhendrickson"
  )
mh_plot
```



**Frequency of @mjhendrickson Twitter Statuses**

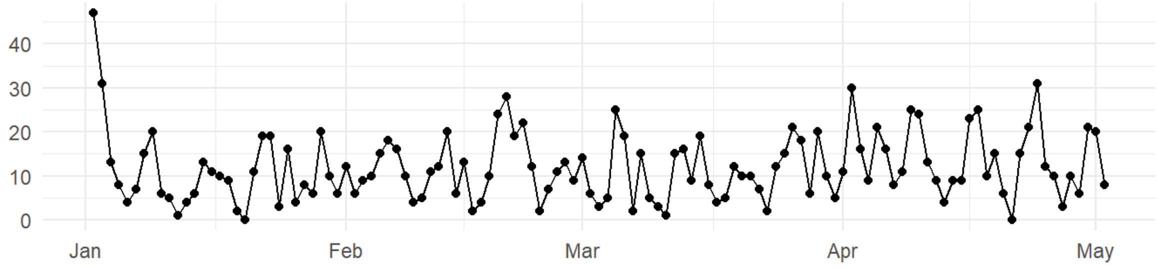
Tweet counts by day since January 1, 2019

Source: Data collected via rtweet - graphic by @mjhendrickson

## Let’s put it all together

Here, we blend the two plots together with `gridExtra`. Since the plots were filtered to the same time-frames, we can easily stack the plots to compare Tweet activity across the two handles.

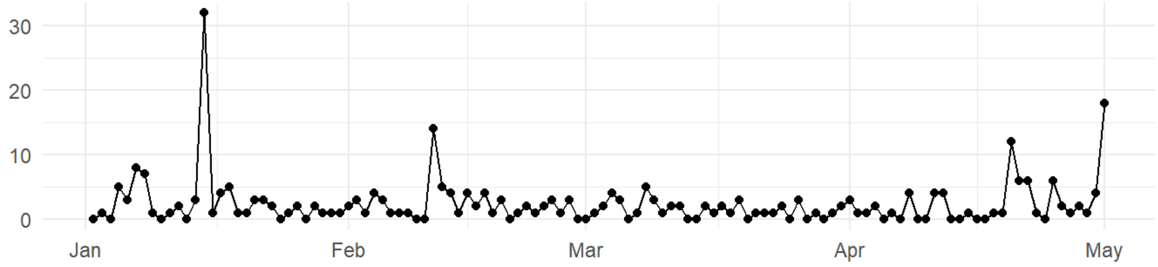
```
grid.arrange(r4ds_plot, mh_plot)
```



**Frequency of @R4DScommunity Twitter Statuses**

Tweet counts by day since January 1, 2019

Source: Data collected via rtweet - graphic by @mjhendrickson



**Frequency of @mjhendrickson Twitter Statuses**

Tweet counts by day since January 1, 2019

Source: Data collected via rtweet - graphic by @mjhendrickson

Clearly, the R for Data Science handle is much more active than mine!

But it looks like something happened mid-January! Let’s find out what I was up to.

# What Happened Mid-January?

Let’s start off by getting the Tweet counts by day for January.

```
mh %>%
  filter(created_at_est >= "2019-01-10" &
         created_at_est <= "2019-01-20") %>%
  group_by(floor_date(created_at_est, unit = "day")) %>% # uses lubridate, rounds to day
  summarize(n = n())
```

```
## # A tibble: 8 x 2
##   `floor_date(created_at_est, unit = "day")`     n
##   <dttm>                                     <int>
## 1 2019-01-10 00:00:00                            1
## 2 2019-01-12 00:00:00                            2
## 3 2019-01-14 00:00:00                           31
## 4 2019-01-15 00:00:00                            4
## 5 2019-01-16 00:00:00                            2
## 6 2019-01-17 00:00:00                            3
## 7 2019-01-18 00:00:00                            5
## 8 2019-01-19 00:00:00                            1
```

We can see that January 14th had 31 Tweets! But what were they?

```
# Search the created_at_est field for records starting with "2019-01-14"
# to get contents of all tweets fro 2019-01-14
mh_jan_14 <- filter(mh, grepl("2019-01-14", created_at_est, fixed = TRUE))
# kable used here to display emoji, links, etc.
kable(select(mh_jan_14, created_at_est, text))
```

| created_at_est | text |
|---|---|
| 2019-01-14 22:28:40 | @StatGarrett @WeAreRLadies Thanks for the reminder of the WIP status. I hope you’re enjoying rstudio::conf. I wasn’t able to make it, but the tweets have been great so far. I’ll definitely be on the lookout for videos later. |
| 2019-01-14 22:15:58 | @WeAreRLadies More than happy to spread great resources! Thank you for being the curator this week! |
| 2019-01-14 22:03:39 | @WeAreRLadies Happy to share quality content. Credit to @sharon000 for the great work! |
| 2019-01-14 21:42:53 | @sharon000 Thank you for the great content! |
| 2019-01-14 21:38:52 | With #rstudioconf about to take over our timelines, I made an easy way to stay on top of everything #rstudioconf on Twitter! Part #FOMO reducer, part tweet explorer, part Twitter leader board. Hope you like it 😁<br>https://t.co/f23r4Qk4ac (https://t.co/f23r4Qk4ac) |
| 2019-01-14 21:36:03 | I went on a bit of a tagging spree and now you can see all my linguistics/NLP @kaggle Kernels in one place: https://t.co/m8W4Yxvd94 (https://t.co/m8W4Yxvd94)<br>(If you haven’t worked with a Kernel before, it’s a free in-browser hosted data science coding environment that includes GPU access. 👀💻💸) |
| 2019-01-14 21:07:13 | Anyone interested in #rstats, either just getting started or with a lot of experience, should check out this thread (and the recent @WeAreRLadies tweets) for numerous great resources. https://t.co/oGXqt5DrvQ (https://t.co/oGXqt5DrvQ) |
| 2019-01-14 21:06:09 | @WeAreRLadies The “Do More with R” @infoworld page is useful if you like short videos. Also from @sharon000 https://t.co/dD2dXb0MH8 (https://t.co/dD2dXb0MH8) |
| 2019-01-14 21:04:43 | @WeAreRLadies .@sharon000 has a good page for “How Do I?”, which helps when you need to know how to do something in R. https://t.co/LtsbDyTFp9 (https://t.co/LtsbDyTFp9) |

| created_at_est | text |
|---|---|
| 2019-01-14 21:03:13 | @WeAreRLadies Topic specific help - I found this helpful when looking for how to do X in R. via @UC_Rstats https://t.co/LnBaYVV5Oa (https://t.co/LnBaYVV5Oa) |
| 2019-01-14 21:01:49 | @WeAreRLadies Efficient R Programming by Colin Gillespie @jumping_uk and @robinlovelace https://t.co/gGE5Qa7QDO (https://t.co/gGE5Qa7QDO) |
| 2019-01-14 20:59:41 | @WeAreRLadies Exploratory Data Analysis by @rdpeng https://t.co/SWn2xyHW3T (https://t.co/SWn2xyHW3T) |
| 2019-01-14 20:58:58 | @WeAreRLadies The Tidyverse Cookbook by @StatGarrett. https://t.co/9zooapHn9S (https://t.co/9zooapHn9S) |
| 2019-01-14 20:58:05 | @WeAreRLadies From the https://t.co/kP2M4ic6En (https://t.co/kP2M4ic6En), An Introduction to R. https://t.co/oQoMT3toP6 (https://t.co/oQoMT3toP6) |
| 2019-01-14 20:56:23 | @WeAreRLadies Also Tidy Evaluation by @hadleywickham and @_lionelhenry (at least I hope that’s the right Lionel Henry). https://t.co/zHM9wTQzYb (https://t.co/zHM9wTQzYb) |
| 2019-01-14 20:54:08 | @WeAreRLadies Thanks for sharing these wonderful resources. I have a bunch I’d like to add. Cran has a few good resources. They may look bland, but there’s a lot of great info. https://t.co/7hH3Lsq07X (https://t.co/7hH3Lsq07X) https://t.co/RDkcbJMYv6 (https://t.co/RDkcbJMYv6) |
| 2019-01-14 20:43:10 | 12. “Happy Git and GitHub for the useR” by @JennyBryan and @jimhester_ guides us how to use the benefits of #git and #github for your #rstats code!<br>https://t.co/02IysNdwCj (https://t.co/02IysNdwCj) https://t.co/qQPWkHlnBj (https://t.co/qQPWkHlnBj) |
| 2019-01-14 20:42:58 | 11. “Fundamentals of Data Visualization” by @ClausWilke. The book guides it’s readers on how to get the right #dataviz as accurately and with as much impact as possible! Our beloved #rstats is the language used. https://t.co/o1fRlKQ5N7 (https://t.co/o1fRlKQ5N7) https://t.co/Rem506feLE (https://t.co/Rem506feLE) |
| 2019-01-14 20:33:22 | 10. “Hands-On Programming with R” by the also incredibly productive @StatGarrett! A project oriented resource to learn #RStats and #datascience.<br>https://t.co/PQz5lJMqUC (https://t.co/PQz5lJMqUC) https://t.co/TISge5bChw (https://t.co/TISge5bChw) |
| 2019-01-14 20:29:01 | 9. “Learning Statistics with R” by @djnavarro. It contains all of the glorious material for her introductory statistics class using #rstats! https://t.co/eYHdFt7ql5 (https://t.co/eYHdFt7ql5) https://t.co/MTuLZMXkxr (https://t.co/MTuLZMXkxr) |
| 2019-01-14 20:28:57 | 8. “R Markdown: The Definitive Guide” by @xieyihui @fly_upside_down &amp; @StatGarrett . A great technical reference guide to.. you guessed it.. RMarkdown in #rstats!<br>https://t.co/lUHNXQfCMe (https://t.co/lUHNXQfCMe) https://t.co/RlJlBH09NY (https://t.co/RlJlBH09NY) |
| 2019-01-14 20:28:48 | 7. “Data Visualization. A practical introduction” by @kjhealy. Using #rstats examples, the book teaches us the foundations of #dataviz! https://t.co/gXLIrz8mHq (https://t.co/gXLIrz8mHq) https://t.co/GOc7eeyZWu (https://t.co/GOc7eeyZWu) |
| 2019-01-14 20:28:34 | 6. Stat 545 by @STAT545. UBC Statistics course in data wrangling, exploration, and analysis with R, taught by @JennyBryan! https://t.co/HoCwNSlI3y (https://t.co/HoCwNSlI3y) https://t.co/3ut9vrT9Rq (https://t.co/3ut9vrT9Rq) |
| 2019-01-14 20:28:30 | 5. “Text Mining with R. A Tidy Approach” by @juliasilge and @drob. What a gem! This book demystifies the scary world that is analyzing text. Highly recommended! #rstats https://t.co/HH7z27B7ga (https://t.co/HH7z27B7ga) https://t.co/wn0syVooEx (https://t.co/wn0syVooEx) |
| 2019-01-14 20:28:26 | 4. More presents from @hadleywickham with the book “R Packages”. A great resource for turning your precious #rstats code into reusable packages to share with others! https://t.co/XcQ0pHrZMW (https://t.co/XcQ0pHrZMW) https://t.co/aMltGvsc6m (https://t.co/aMltGvsc6m) |
| 2019-01-14 20:28:20 | 3. “Advanced R” by @hadleywickham who seems to have discovered a new time dimension to work in! A great resource to further your #rstats skills or transition from other languages.<br>https://t.co/6kvrngbmy9 (https://t.co/6kvrngbmy9) https://t.co/e2bR1iO4F9 (https://t.co/e2bR1iO4F9) |
| 2019-01-14 20:28:14 | 2. “An Introduction to Statistical and Data Sciences via R” by @old_man_chester and @rudeboybert. An incredibly beginner friendly introduction to both #datascience and #statistics concepts as well as #RStats. Very nicely done! https://t.co/7kzfunXe0v (https://t.co/7kzfunXe0v) |

| created_at_est | text |
|---|---|
| 2019-01-14 20:28:06 | Since I’m a complete pack rat, I’ll now share with you a bunch of great, free online books to help learn #rstats.<br>• The almighty “R 4 Data Science” by @hadleywickham and @StatGarrett. Teaches us how to structure, transform, visualize and model #data!<br>https://t.co/QxlDus8tiy (https://t.co/QxlDus8tiy) https://t.co/z2EuIefUP6 (https://t.co/z2EuIefUP6) |
| 2019-01-14 13:35:14 | If you can’t attend #rstudioconf in person this year, check out some of last year’s talks shared by @dataandme !! https://t.co/nmrX1mVd67 (https://t.co/nmrX1mVd67) |
| 2019-01-14 09:27:13 | Happy 2019! We are pumped to kick off the year at #rstudioconf with Austin local, @LittleMissData! #RLadies https://t.co/0UyGJeOGwj (https://t.co/0UyGJeOGwj) |
| 2019-01-14 09:26:13 | The @R4DScommunity welcomes you to week 3 of #TidyTuesday! We’re exploring data about space launches!!<br>📂 https://t.co/sElb4fcv3u (https://t.co/sElb4fcv3u) 📰 https://t.co/86Xl3egAyv (https://t.co/86Xl3egAyv)<br>#r4ds #tidyverse #rstats #dataviz https://t.co/cDOvAAHXeS (https://t.co/cDOvAAHXeS) |

On January 14th, WeAreRLadies (https://twitter.com/WeAreRLadies) had a thread about sharing resources. I guess I went a little crazy.

## What Else Can We Get?

Reviewing a few records also lets us explore the data we get back from Twitter

```
colnames(mh)
```

```
##  [1] "user_id"                 "status_id"
##  [3] "created_at"              "screen_name"
##  [5] "text"                    "source"
##  [7] "display_text_width"      "reply_to_status_id"
##  [9] "reply_to_user_id"        "reply_to_screen_name"
## [11] "is_quote"                "is_retweet"
## [13] "favorite_count"          "retweet_count"
## [15] "hashtags"                "symbols"
## [17] "urls_url"                "urls_t.co"
## [19] "urls_expanded_url"       "media_url"
## [21] "media_t.co"              "media_expanded_url"
## [23] "media_type"              "ext_media_url"
## [25] "ext_media_t.co"          "ext_media_expanded_url"
## [27] "ext_media_type"          "mentions_user_id"
## [29] "mentions_screen_name"    "lang"
## [31] "quoted_status_id"        "quoted_text"
## [33] "quoted_created_at"       "quoted_source"
## [35] "quoted_favorite_count"   "quoted_retweet_count"
## [37] "quoted_user_id"          "quoted_screen_name"
## [39] "quoted_name"             "quoted_followers_count"
## [41] "quoted_friends_count"    "quoted_statuses_count"
## [43] "quoted_location"         "quoted_description"
## [45] "quoted_verified"         "retweet_status_id"
## [47] "retweet_text"            "retweet_created_at"
## [49] "retweet_source"          "retweet_favorite_count"
## [51] "retweet_retweet_count"   "retweet_user_id"
## [53] "retweet_screen_name"     "retweet_name"
## [55] "retweet_followers_count" "retweet_friends_count"
## [57] "retweet_statuses_count"  "retweet_location"
## [59] "retweet_description"     "retweet_verified"
## [61] "place_url"               "place_name"
## [63] "place_full_name"         "place_type"
## [65] "country"                 "country_code"
## [67] "geo_coords"              "coords_coords"
## [69] "bbox_coords"             "status_url"
## [71] "name"                    "location"
## [73] "description"             "url"
## [75] "protected"               "followers_count"
## [77] "friends_count"           "listed_count"
## [79] "statuses_count"          "favourites_count"
## [81] "account_created_at"      "verified"
## [83] "profile_url"             "profile_expanded_url"
## [85] "account_lang"            "profile_banner_url"
## [87] "profile_background_url"  "profile_image_url"
## [89] "created_at_est"
```

For a full list of what's included in a data pull like this, go to the Tweet Data Dictionary here (https://developer.twitter.com/en/docs/tweets/data-dictionary/overview/tweet-object.html). But of interest to me are:

- `created_at` = UTC time of Tweet
- `text` = content of the Tweet
- `source` = utility used to post the Tweet
- `reply_to_screen_name` = username of original Tweet's author if the Tweet was a reply
- `is_retweet` = TRUE/FALSE if this was a re-tweet
- `favorite_count` = number of times the Tweet was favorited
- `retweet_count` = number of times the Tweet was re-tweeted
- `hashtags` = string of all hashtags used
- `urls_url` = string of all urls
- `mentions_screen_name` = string of screen_names mentioned
- A whole host of fields on re-tweets, such as the user, their follower count, friend count, reach, location (if enabled), text

- Geolocation fields, such as place name, coordinates, location
- User info, including `followers_count` , `friends_count` , `favorites_count`

## Let’s do a quick analysis of a few interesting fields

### Tweet Sources

What’s my mix of sources? Do I Tweet from my phone? What about desktop?

I’m not going to lie, I’m busy and want a consistent feed to Twitter. So how much do I Tweet through Buffer?

Let’s find out.

```
mh %>%
  group_by(source) %>%
  summarize(n = n())
```

```
## # A tibble: 7 x 2
##   source                   n
##   <chr>                <int>
## 1 Buffer                 267
## 2 Twitter for Android    523
## 3 Twitter for iPad         1
## 4 Twitter for iPhone      26
## 5 Twitter for Websites     1
## 6 Twitter Web App        112
## 7 Twitter Web Client     494
```

Interesting! I clearly prefer Tweeting from my phone closely followed by the web client.

I Tweeted just once from my old iPad. That was when I mostly consumed Tweets to stay on top of news during my doctoral program.

I had to search the Twitter developer pages (https://developer.twitter.com/en/docs/twitter-for-websites/overview.html) to determine the source of `Twitter for Websites` . This will show if you click the Tweet button on a website. Not sure what spurred me to Tweet from any specific website.

```
mh %>%
  filter(str_detect(source, "Twitter for Websites")) %>%
  select(source, created_at_est, text, urls_expanded_url, mentions_screen_name)
```

```
## # A tibble: 1 x 5
##   source   created_at_est      text       urls_expanded_u~ mentions_screen~
##   <chr>    <dttm>              <chr>      <list>           <list>
## 1 Twitter~ 2013-09-12 21:23:33 88% of st~ <chr [1]>        <chr [1]>
```

A Revolution Analytics blog post (https://blog.revolutionanalytics.com/2013/09/statistician-survey-results.html) from September 2013 inspired me to Tweet directly from the page.

### Preferred Hashtags

Now that I know my preferred method of Tweeting, what do I Tweet about? One way to check is to look at hashtag usage.

### How do Others Interact with my Tweets?

### How long are my Tweets?

```
mh %>%
  #filter(created_at_est >= "2019-01-10") # add filter for pre 280 character limit
  group_by(display_text_width) %>%
  #tally(sort = TTRUE, wt = NULL)
  summarize(n())
```

```
## # A tibble: 237 x 2
##    display_text_width `n()`
##                 <dbl> <int>
##  1                  0     1
##  2                  2     1
##  3                  7     1
##  4                 11     1
##  5                 13     1
##  6                 15     1
##  7                 16     1
##  8                 17     1
##  9                 19     1
## 10                 22     2
## # ... with 227 more rows
```

I liked to push the limits of my Tweets. 140 used to be the character limit. Let’s check again to see if that holds *after* the character limit increase (2017-11-07).

```
mh %>%
  filter(created_at_est >= "2017-11-07") %>%
  group_by(display_text_width) %>%
  tally(sort = TRUE, wt = NULL)
```

```
## # A tibble: 232 x 2
##    display_text_width     n
##                 <dbl> <int>
##  1                140   229
##  2                139    38
##  3                144    17
##  4                 85     8
##  5                116     7
##  6                121     7
##  7                 86     6
##  8                101     6
##  9                112     6
## 10                124     6
## # ... with 222 more rows
```

## Are my Tweets original, or re-Tweets??

```
mh %>%
  group_by(is_retweet) %>%
  tally(sort = TRUE, wt = NULL)
```

```
## # A tibble: 2 x 2
##   is_retweet     n
##   <lgl>      <int>
## 1 FALSE        968
## 2 TRUE         456
```

Whew! I’m about a 2:1 Tweet:Re-Tweet ratio. Does this vary by source?

```
mh %>%
  group_by(source, is_retweet) %>%
  tally(sort = FALSE, wt = NULL)
```

```
## # A tibble: 12 x 3
## # Groups:   source [7]
##    source              is_retweet     n
##    <chr>               <lgl>      <int>
##  1 Buffer              FALSE        266
##  2 Buffer              TRUE           1
##  3 Twitter for Android FALSE        238
##  4 Twitter for Android TRUE         285
##  5 Twitter for iPad    TRUE           1
##  6 Twitter for iPhone  FALSE          7
##  7 Twitter for iPhone  TRUE          19
##  8 Twitter for Websites FALSE         1
##  9 Twitter Web App     FALSE         66
## 10 Twitter Web App     TRUE          46
## 11 Twitter Web Client  FALSE        390
## 12 Twitter Web Client  TRUE         104
```

I don’t re-Tweet through Buffer. Of Tweets from my phone, the majority are re-Tweets. About 3/4 of Tweets from the browser are original.

### How engaging are my Tweets? Let’s start with some simple descriptive stats.

```
mh %>%
  filter(is_retweet == "FALSE") %>% # Keeps only original Tweets
  summarize(
    min  =    min(favorite_count),
    med  = median(favorite_count),
    mean =   mean(favorite_count),
    max  =    max(favorite_count),
    iqr  =    IQR(favorite_count),
    n    =      n()
  )
```

```
## # A tibble: 1 x 6
##     min   med  mean   max   iqr     n
##   <dbl> <dbl> <dbl> <dbl> <dbl> <int>
## 1     0     1  2.90   165     3   968
```

Not unexpected. The median is 0, mean is ~3. I had one good tweet at 165!

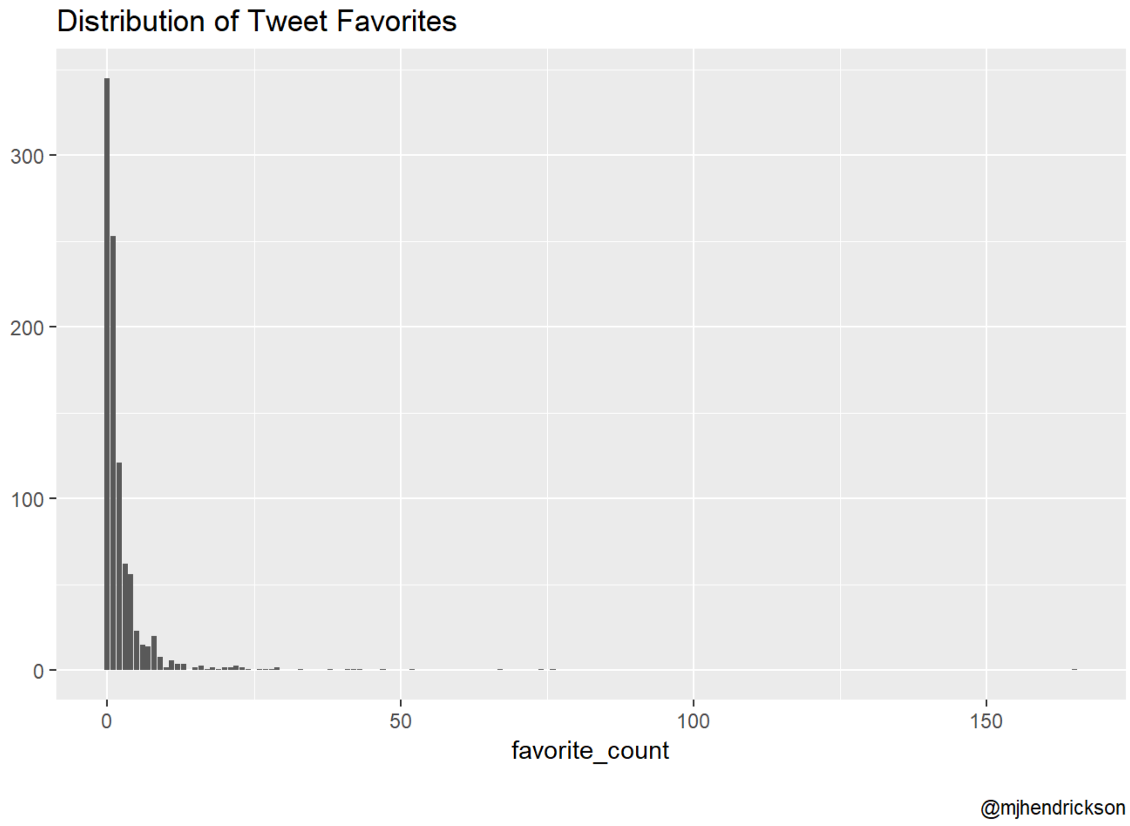
Let’s look at the distribution.

```
mh %>%
  filter(is_retweet == "FALSE") %>% # Keeps only original Tweets
  group_by(favorite_count) %>%
  summarize(n = n()) %>%
  arrange(desc(favorite_count))
```

```
## # A tibble: 39 x 2
##    favorite_count     n
##             <int> <int>
##  1            165     1
##  2             76     1
##  3             74     1
##  4             67     1
##  5             52     1
##  6             47     1
##  7             43     1
##  8             42     1
##  9             41     1
## 10             38     1
## # ... with 29 more rows
```

It is clear that very few of my Tweets recieved a large number of likes. If that isn't clear enough, let's visualize the distribution.

```
mh %>%
  filter(is_retweet == "FALSE") %>% # Keeps only original Tweets
ggplot() +
  geom_bar(mapping = aes(x = favorite_count)) +
  #scale_x_continuous(name = " ", breaks = c(5, 165)) + # Enable if filtered
  scale_y_continuous(name = " ", labels = comma) +
  labs(title = "Distribution of Tweet Favorites",
       caption = "\n @mjhendrickson")
```



```
mh %>%
  filter(is_retweet == "FALSE") %>% # Keeps only original Tweets
  summarize(
    min  =    min(retweet_count),
    med  = median(retweet_count),
    mean =   mean(retweet_count),
    max  =    max(retweet_count),
    iqr  =    IQR(retweet_count),
    n    =      n()
  )
```

```
## # A tibble: 1 x 6
##     min   med  mean   max   iqr     n
##   <dbl> <dbl> <dbl> <dbl> <dbl> <int>
## 1     0     0 0.785    47     1   968
```

As expected, the numbers are even lower than `favorite_counts` . Though, I did have one with 47 re-Tweets! Let’s see if it was the same Tweet.

Now let’s look at a combined engagement metric of favorites and re-Tweets.