Appendix

2020-05-15

Example of downloading tweets using rtweet

```r
rm(list=ls())
library(lubridate, quietly = T, warn.conflicts = F)  # function setdiff() masked
library(magrittr, quietly = T)

## rtweet's function search_fullarchive originally capped
## queries of tweet-downloads at 100 tweets. Since the
## premium Twitter API allows for up to 500 tweets,
## a user-created fix was used for downloading all tweets

library(devtools, quietly = T)

# remotes::install_github("kevintaylor/rtweet")

library(rtweet, quietly = T)

## Custom-made functions for file routines

## The following function was used to send queries
## to the Twitter API

# env_name <- "NGSSproject"

tweetdownload <- function(from, to){
  search_fullarchive(q="#NGSSchat", n=500, fromDate = from,
                     toDate = to, env_name = env_name)
}

## Download queries are done through timeframes of the format
## yyyymmddhhmm. The following function converts UCT timestamps
## to such timeframes which was later used to download gaps
## in between the storify data set.

UCT2frame <- function(datetime){
  frame <- strftime(datetime,"%Y-%m-%d %H:%M:%S", tz="UCT")
  frame <- substr(frame, 1, nchar(frame) - 3)
  frame <- gsub("-", "", frame)
  frame <- gsub(":", "", frame)
  frame <- gsub(" ", "", frame)
  return(frame)
}

## Through the following function, query results were saved
## in a .rda file format with a specific filname that comprised
## of the chronologically first and last tweets of the query.

save_rda_dl <- function(tweets_dl, stampstring=""){
  first <- tweets_dl$created_at %>% min() %>% UCT2frame()
  last <- tweets_dl$created_at %>% max() %>% UCT2frame()

  file_name <- paste(stampstring, "DATA_", first, "_TO_", last, ".rda", sep="")

  save(tweets_dl, file = file_name)
}

## Example usage

from <- as.POSIXct(1234, origin = "2020-01-01", tz = "UTC")
to <- as.POSIXct(9876, origin = "2020-01-01", tz = "UTC")

from  # Format in data
```

```
## [1] "2020-01-01 00:20:34 UTC"
```

```r
from <- UCT2frame(from)
to <- UCT2frame(to)

from  # query format
```

```
## [1] "202001010020"
```

```r
# tweets <- tweetdownload(from, to)

# save_rda_dl(tweets)
```

Calculating gaps in between Storify #NGSSchat database

```r
## Read storify data set

storifydata <- readRDS("storify_data.rds")

## Range of tweets in Storify data set timewise

min(storifydata$created_at)
```

```
## [1] "2012-05-11 00:01:09 UTC"
```

```r
max(storifydata$created_at)
```

```
## [1] "2017-11-17 04:08:07 UTC"
```

```r
## Order tweets chronologically

storifydata <- storifydata[order(storifydata$created_at),]

## Only include tweets posted between 2014 and 2016

start <- min(grep("2014", storifydata$created_at))
end <- max(grep("2016", storifydata$created_at))
```

```r
storifydata <- storifydata[start:end,]

## Calculate timegaps in between tweets

reference <- data.frame(
  t0 <- storifydata$created_at[1:nrow(storifydata)-1],
  t1 <- storifydata$created_at[2:nrow(storifydata)]
)
colnames(reference) <- c("t0", "t1")

## Adding the two time gaps from the beginning of 2014 to
## the first tweets in Storify data set of 2014 as well as
## from last tweet in Storify data set of 2016 to the end
## of 2016

start2014 <- as.POSIXct(strptime("2014-01-01 00:00:01",
                                 format="%Y-%m-%d %H:%M:%S"), tz="UTC")

reference <- rbind(reference[1,], reference)

reference[1,1] <- start2014
reference[1,2] <- reference[2,1]

end2016 <- as.POSIXct(strptime("2016-12-31 23:59:59",
                               format="%Y-%m-%d %H:%M:%S"), tz="UTC")

reference <- rbind(reference, reference[nrow(reference),])

reference[nrow(reference),2] <- end2016
reference[nrow(reference),1] <- reference[nrow(reference)-1,2]

## Create time differences in between tweets

reference$delta <- seconds_to_period(reference$t1 - reference$t0)

## Order pairs of tweets by biggest time gaps

reference <- reference[order(reference$delta, decreasing = T),]

## Select the 48 biggest time gaps (effectively choosing all
## time gaps larger than 6 hours)

reference <- reference[1:48,]

## Order these 48 time gaps chronologically for better overview

reference <- reference[order(reference$t0),]

## Read in downloads of queries addressing these time gaps
## Note: Also ordered chronologically

fn <- list(
"DATA_201401011738_TO_201404032154.rda",
"DATA_201404040214_TO_201404180058.rda",
"DATA_201404180247_TO_201405020054.rda",
"DATA_201405020209_TO_201405160017.rda",
"DATA_201405160207_TO_201405291751.rda",
"DATA_201405300204_TO_201406130059.rda",
"DATA_201406130213_TO_201406270054.rda",
"DATA_201406270211_TO_201407101021.rda",
"DATA_201407101310_TO_201407110048.rda",
"DATA_201407110202_TO_201408010054.rda",
"DATA_201408010243_TO_201408150059.rda",
"DATA_201408150201_TO_201408280927.rda",
"DATA_201408280947_TO_201408290020.rda",
"DATA_201408290207_TO_201409120057.rda",
"DATA_201410030134_TO_201410170056.rda",
"DATA_201410170223_TO_201410292040.rda",
"DATA_201410292101_TO_201410310059.rda",
"DATA_201411070232_TO_201411141125.rda",
"DATA_201411141412_TO_201411210159.rda",
"DATA_201411210314_TO_201412050159.rda",
"DATA_201412190303_TO_201501090159.rda",
"DATA_201501230255_TO_201502060159.rda",
"DATA_201503052243_TO_201503130039.rda",
"DATA_201503131023_TO_201503200059.rda",
"DATA_201503200223_TO_201504030059.rda",
"DATA_201505270116_TO_201506050057.rda",
"DATA_201506050240_TO_201506190059.rda",
"DATA_201506190201_TO_201507030056.rda",
"DATA_201507030211_TO_201507170058.rda",
"DATA_201507170213_TO_201507310058.rda",
"DATA_201507310230_TO_201508070057.rda",
"DATA_201508070201_TO_201508210004.rda",
"DATA_201508211409_TO_201509040100.rda",
"DATA_201509040200_TO_201509180059.rda",
"DATA_201509180200_TO_201510020056.rda",
"DATA_201510020225_TO_201510160100.rda",
"DATA_201510160942_TO_201511060154.rda",
"DATA_201511061825_TO_201511200159.rda",
"DATA_201511200307_TO_201512040200.rda",
"DATA_201512040303_TO_201512180200.rda",
"DATA_201512180306_TO_201601080159.rda",
"DATA_201601080311_TO_201601220159.rda",
"DATA_201601220356_TO_201602050201.rda",
"DATA_201602051650_TO_201602190200.rda",
"DATA_201602190314_TO_201603040159.rda",
"DATA_201604251954_TO_201605060103.rda",
"DATA_201605060227_TO_201605061332.rda",
"DATA_201605061353_TO_201612312335.rda"
)

## While combining queries to new data frame, count
## number of new tweets per query in time gaps

all <- data.frame()
```

```r
len <- c()
for (name in fn){
  load(name)
  current <- tweets_dl
  len <- c(len, nrow(current))
  current <- current[order(current$created_at),]
  all <- rbind(all, current)
}

## Assign number of new tweets to time gaps

reference$n_new <- len

## Overview of new downloaded tweets in time gaps

reference
```

```
##                         t0                  t1           delta n_new
## 1      2014-01-01 00:00:01 2014-04-03 23:38:12 92d 23H 38M 11S   282
## 473    2014-04-04 02:14:45 2014-04-18 01:00:22 13d 22H 45M 37S   202
## 814    2014-04-18 02:47:59 2014-05-02 01:01:46 13d 22H 13M 47S   164
## 927    2014-05-02 02:09:30 2014-05-16 00:22:21 13d 22H 12M 51S   109
## 1137   2014-05-16 02:07:46 2014-05-29 18:13:38  13d 16H 5M 52S   108
## 1362   2014-05-30 02:04:04 2014-06-13 01:00:21 13d 22H 56M 17S   157
## 1786   2014-06-13 02:13:29 2014-06-27 00:57:46 13d 22H 44M 17S   288
## 2130   2014-06-27 02:11:25 2014-07-10 13:10:43 13d 10H 59M 18S   162
## 2131   2014-07-10 13:10:43 2014-07-11 00:52:35     11H 41M 52S    26
## 2333   2014-07-11 02:02:49 2014-08-01 00:59:44 20d 22H 56M 55S   224
## 2660   2014-08-01 02:43:59 2014-08-15 01:00:22 13d 22H 16M 23S   260
## 3086   2014-08-15 02:01:32 2014-08-28 09:47:14  13d 7H 45M 42S   194
## 3087   2014-08-28 09:47:14 2014-08-29 00:23:06     14H 35M 52S    39
## 3342   2014-08-29 02:07:48 2014-09-12 00:58:19 13d 22H 50M 31S   236
## 3642   2014-09-12 02:10:30 2014-10-17 01:00:04 34d 22H 49M 34S   500
## 3996   2014-10-17 02:23:30 2014-10-29 21:01:29 12d 18H 37M 59S   184
## 3997   2014-10-29 21:01:29 2014-10-31 01:00:23   1d 3H 58M 54S    76
## 4328   2014-10-31 02:17:00 2014-11-14 14:12:38 14d 11H 55M 38S   500
## 4329   2014-11-14 14:12:38 2014-11-21 02:00:17  6d 11H 47M 39S   141
## 4666   2014-11-21 03:14:24 2014-12-05 02:01:43 13d 22H 47M 19S   222
## 4946   2014-12-05 03:02:49 2015-01-09 02:00:29 34d 22H 57M 40S   500
## 5131   2015-01-09 03:11:47 2015-02-06 02:00:34 27d 22H 48M 47S   500
## 5519   2015-02-06 02:57:39 2015-03-13 00:44:26 34d 21H 46M 47S   500
## 6265   2015-03-13 10:23:32 2015-03-20 01:00:09  6d 14H 36M 37S   757
## 7157   2015-03-20 02:23:47 2015-04-03 01:00:45 13d 22H 36M 58S   587
## 7692   2015-04-03 02:02:48 2015-06-05 01:00:05 62d 22H 57M 17S   500
## 8284   2015-06-05 02:40:44 2015-06-19 01:00:22 13d 22H 19M 38S   532
## 8652   2015-06-19 02:01:28 2015-07-03 00:57:27 13d 22H 55M 59S   500
## 9097   2015-07-03 02:11:10 2015-07-17 01:00:43 13d 22H 49M 33S   301
## 9441   2015-07-17 02:13:01 2015-07-31 01:00:55 13d 22H 47M 54S   354
## 10116  2015-07-31 02:30:55 2015-08-07 00:59:10  6d 22H 28M 15S   214
## 10388  2015-08-07 02:01:10 2015-08-21 00:22:36 13d 22H 21M 26S   513
## 10881  2015-08-21 14:09:56 2015-09-04 01:01:09 13d 10H 51M 13S   367
## 11299  2015-09-04 02:00:03 2015-09-18 01:00:40  13d 23H 0M 37S   422
## 11734  2015-09-18 02:00:53 2015-10-02 01:01:27  13d 23H 0M 34S   475
## 12057  2015-10-02 02:05:25 2015-10-16 01:01:51 13d 22H 56M 26S   500
## 12538  2015-10-16 02:18:45 2015-11-06 02:01:57 20d 23H 43M 12S   500
## 13009  2015-11-06 03:05:03 2015-11-20 02:00:52 13d 22H 55M 49S   500
## 13463  2015-11-20 03:07:44 2015-12-04 02:01:01 13d 22H 53M 17S   430
## 13959  2015-12-04 03:03:11 2015-12-18 02:01:12  13d 22H 58M 1S   442
## 14305  2015-12-18 03:06:25 2016-01-08 02:00:23 20d 22H 53M 58S   373
## 14727  2016-01-08 03:02:07 2016-01-22 02:00:42 13d 22H 58M 35S   500
## 15215  2016-01-22 03:56:12 2016-02-05 02:02:11  13d 22H 5M 59S   507
## 15932  2016-02-05 16:50:37 2016-02-19 02:01:52  13d 9H 11M 15S   569
## 16485  2016-02-19 03:14:59 2016-03-04 02:00:08  13d 22H 45M 9S   481
## 16946  2016-03-04 03:00:05 2016-05-06 01:04:15  62d 22H 4M 10S   493
## 17224  2016-05-06 02:27:31 2016-05-06 13:35:40       11H 8M 9S    17
## 172261 2016-05-06 13:53:14 2016-12-31 23:59:59 239d 10H 6M 45S 16689
```

```r
nrow(storifydata)  # Sum of tweets in storify data
```

```
## [1] 17226
```

```r
nrow(all)  # Sum of downloaded tweets in time gaps
```

```
## [1] 33097
```

Analysis of tweets downloaded between gaps in Storify #NGSSchat database

```r
## Created data frames of statistics for favourites
## and retweets in storify data and gap data

favourites <- data.frame(

M = c(mean(storifydata$favorite_count), mean(all$favorite_count)),
Med = c(median(storifydata$favorite_count), median(all$favorite_count)),
SD = c(sd(storifydata$favorite_count), sd(all$favorite_count)),
Min = c(min(storifydata$favorite_count), min(all$favorite_count)),
Max = c(max(storifydata$favorite_count), max(all$favorite_count))

)

favourites$M <- round(favourites$M, 2)
favourites$SD <- round(favourites$SD, 2)

rownames(favourites) <- c("Storifydata", "Gapdata")

retweets <- data.frame(
M = c(mean(storifydata$retweet_count), mean(all$retweet_count)),
Med = c(median(storifydata$retweet_count), median(all$retweet_count)),
SD = c(sd(storifydata$retweet_count), sd(all$retweet_count)),
Min = c(min(storifydata$retweet_count), min(all$retweet_count)),
Max = c(max(storifydata$retweet_count), max(all$retweet_count))
)

retweets$M <- round(retweets$M, 2)
retweets$SD <- round(retweets$SD, 2)

rownames(retweets) <- c("Storifydata", "Gapdata")

favourites
```

```
##                M Med   SD Min Max
## Storifydata 1.20   1 1.75   0  38
## Gapdata     0.93   0 2.66   0 248
```

```
retweets
```

```
##                M Med   SD Min Max
## Storifydata 0.35   0 1.04   0  53
## Gapdata     0.43   0 1.93   0 198
```

Selecting tweets within rush hours to re-download and compare

```
set.seed(123)

## Select days with more than 300 tweets in storify data set

## Extract days from timestamps in storify data set

time <- storifydata$created_at

days <- format(time, format='%Y-%m-%d')

## Aggregate number of tweets over days

freq <- table(days)

## Exclusion of days with less than 300 tweets

freq <- freq[300 <= freq]

## For each year, sample three days by subsetting
## all days by year and sampling three days in each subset

freq <- data.frame(freq)

t2014 <- freq[grep("2014", freq$days),]
t2014 <- t2014[sample(nrow(t2014), 3), ]

t2015 <- freq[grep("2015", freq$days),]
t2015 <- t2015[sample(nrow(t2015), 3), ]

t2016 <- freq[grep("2016", freq$days),]
t2016 <- t2016[sample(nrow(t2016), 3), ]

result <- rbind(t2014, t2015, t2016)
colnames(result) <- c("Date", "Number of tweets")

result  # Days that were redownloaded
```

```
##          Date Number of tweets
## 3  2014-06-13              424
## 10 2014-11-21              337
## 2  2014-04-18              341
## 28 2015-12-18              346
## 21 2015-09-04              418
## 15 2015-06-05              592
## 32 2016-02-19              553
## 30 2016-01-22              488
## 33 2016-03-04              461
```

Comparing re-downloaded tweets with the Storify #NGSSchat database

```
rm(list=ls())

## Timeframes and number of tweets in original data

days <- read.table(header=T, text="
day ntweets
2014-06-13  424
2014-11-21  337
2014-04-18  341
2015-12-18  346
2015-09-04  418
2015-06-05  592
2016-02-19  553
2016-01-22  488
2016-03-04  461"
)

## Read in storify data

storifydata <- readRDS("storify_data.rds")

## Order tweets chronologically

storifydata <- storifydata[order(storifydata$created_at),]

## Select days of re-downloaded days from storify data set

old <- data.frame()

for (day in days$day){
  index <- grep(day, as.character(storifydata$created_at))
  old <- rbind(old, storifydata[index,])
}

## Read in new data and order each data set chronologically

fn <- list(
  "DATA_201406130033_TO_201406132339.rda",
  "DATA_201411210000_TO_201411212212.rda",
  "DATA_201404180000_TO_201404182146.rda",
  "DATA_201512180030_TO_201512182355.rda",
  "DATA_201509040005_TO_201509042315.rda",
  "DATA_201506050002_TO_201506052015.rda",
  "DATA_201602190002_TO_201602192307.rda",
  "DATA_201601220009_TO_201601222320.rda",
  "DATA_201603040026_TO_201603042206.rda"
)

new <- data.frame()
for (name in fn){
  load(name)
  current <- tweets_dl
  current <- current[order(current$created_at),]
  new <- rbind(new, current)
}

new_n <- c()

## Compare number of tweets between both data sets for each day

for (day in days$day){
  new_n <- c(new_n, grep(day, as.character(new$created_at)) %>% length())
}

days$n_newtweets <- new_n
colnames(days) <- c("Day", "Storify", "Re_DL")
days$Estimate <- (as.numeric(days[,2]) / as.numeric(days[,3])) * 100

days
```

```
##          Day Storify Re_DL Estimate
## 1 2014-06-13     424   635 66.77165
## 2 2014-11-21     337   662 50.90634
## 3 2014-04-18     341   596 57.21477
## 4 2015-12-18     346   574 60.27875
## 5 2015-09-04     418   639 65.41471
## 6 2015-06-05     592   852 69.48357
## 7 2016-02-19     553   861 64.22764
## 8 2016-01-22     488   751 64.98003
## 9 2016-03-04     461   693 66.52237
```

```
sum(days$Storify)  # Sum of tweets in storify data during selected days
```

```
## [1] 3960
```

```
sum(days$Re_DL)  # Sum of tweets in re-downloaded data during selected days
```

```
## [1] 6263
```

```
sum(days$Storify) * 100 / sum(days$Re_DL)  # % estimate
```

```
## [1] 63.22848
```

Have tweets been deleted since the download of the storify data set?

```
## With the function base::setdiff() the following sets are selected
## through looking at the tweet IDs of the given tweets

## 1) Tweets that are in old but not in new (deleted tweets) and
## 2) Tweets that are in new but not in old (tweets missing in Storify #NGSSchat database)
```

```r
## Old and not in new (deleted tweets)

deleted <- base::setdiff(old$status_id, new$status_id)
length(deleted)
```

```
## [1] 43
```

```r
(length(deleted) / nrow(old)) * 100  # %
```

```
## [1] 1.085859
```

```r
## Corrected estimate of completeness of storify data set assuming no tweets
## would have been deleted

sum(days$Storify) * 100 / (sum(days$Re_DL) + length(deleted))  # % corrected estimate
```

```
## [1] 62.79734
```

```r
## This corrected estimate can also be reformulated by counting
## the number of tweets that are in the new, but not in the storify data set

missing <- base::setdiff(new$status_id, old$status_id)
length(missing)
```

```
## [1] 2346
```

```r
(nrow(old) / (nrow(old) + length(missing))) * 100
```

```
## [1] 62.79734
```

Which kind of data within rush hours misses in the Storify #NGSSchat database?

```r
## Possibility: Data sets differ in inclusion of retweets

## Exclude manually copy-pasted retweets "RT @" and automatically
## created retweets (indicated by the variable is_retweet)

new2 <- new[new$is_retweet==F, ]
new2 <- new2[-grep("RT @", new2$text),]

old2 <- old[old$is_retweet==F, ]
old2 <- old2[-grep("RT @", old2$text),]

new_n2 <- c()
old_n2 <- c()

for (day in days$Day){
  new_n2 <- c(new_n2, grep(day, as.character(new2$created_at)) %>% length())
  old_n2 <- c(old_n2, grep(day, as.character(old2$created_at)) %>% length())
}

days$Storify_noRT <- old_n2
days$Re_DL_noRT <- new_n2
days$Estimate2 <- (as.numeric(days[,5]) / as.numeric(days[,6])) * 100
```

```
days
```

```
##          Day Storify Re_DL Estimate Storify_noRT Re_DL_noRT Estimate2
## 1 2014-06-13     424   635 66.77165          406        451  90.02217
## 2 2014-11-21     337   662 50.90634          321        484  66.32231
## 3 2014-04-18     341   596 57.21477          333        433  76.90531
## 4 2015-12-18     346   574 60.27875          346        480  72.08333
## 5 2015-09-04     418   639 65.41471          418        488  85.65574
## 6 2015-06-05     592   852 69.48357          592        640  92.50000
## 7 2016-02-19     553   861 64.22764          553        684  80.84795
## 8 2016-01-22     488   751 64.98003          488        609  80.13136
## 9 2016-03-04     461   693 66.52237          461        537  85.84730
```

```
sum(days$Storify_noRT)  # Sum of tweets in storify data during selected days without retweets
```

```
## [1] 3918
```

```
sum(days$Re_DL_noRT)  # Sum of tweets in re-downloaded data during selected days without retweets
```

```
## [1] 4806
```

```
sum(days$Storify_noRT) * 100 / sum(days$Re_DL_noRT)  # % estimate
```

```
## [1] 81.5231
```

```
## Corrected estimate assuming no tweets would have been deleted

missing2 <- base::setdiff(new2$status_id, old2$status_id)
(nrow(old2) / (nrow(old2) + length(missing2))) * 100  # % corrected estimate
```

```
## [1] 80.83351
```

```
## Conclusion: Storify data set is still not complete but seems
## more complete when excluding retweets (80.83%)
```

Is the storify data set complete during the most busy hours of #NGSSchat?

```
## Possibility: Most hours of the days that were looked at were not
## busy and excluded from storify data set (see download of time gaps)

## Only looking at tweets between 01:00 and 03:00 UTC (most activity)

## Subsetting data sets

old3 <- old[which(hour(old$created_at) %in% 1:2),]  # 01:00-02:59 UTC
new3 <- new[which(hour(new$created_at) %in% 1:2),]

old_n3 <- c()
new_n3 <- c()

for (day in days$Day){
  new_n3 <- c(new_n3, grep(day, as.character(new3$created_at)) %>% length())
  old_n3 <- c(old_n3, grep(day, as.character(old3$created_at)) %>% length())
}

## Estimate
```

```
sum(old_n3) * 100 / sum(new_n3)  # % estimate
```

```
## [1] 68.04383
```

```
## Corrected estimate assuming no tweets would have been deleted

missing3 <- base::setdiff(new3$status_id, old3$status_id)
(nrow(old3) / (nrow(old3) + length(missing3))) * 100  # % corrected estimate
```

```
## [1] 67.52228
```

```
## Conclusion: Dataset during rush hours and between 01:00 and 03:00 UTC more
## complete than whole days but still only 67.5% complete.

## Subsetting retweets only *and* tweets between 01:00 and 03:00 UTC

new4 <- new3[new3$is_retweet==F, ]
new4 <- new4[-grep("RT @", new4$text),]

old4 <- old3[old3$is_retweet==F, ]
old4 <- old4[-grep("RT @", old4$text),]

new_n4 <- c()
old_n4 <- c()

for (day in days$Day){
  new_n4 <- c(new_n4, grep(day, as.character(new4$created_at)) %>% length())
  old_n4 <- c(old_n4, grep(day, as.character(old4$created_at)) %>% length())
}

## Estimate

sum(old_n4) * 100 / sum(new_n4)  # % estimate
```

```
## [1] 84.75113
```

```
## Corrected estimate assuming no tweets would have been deleted

missing4 <- base::setdiff(new4$status_id, old4$status_id)
(nrow(old4) / (nrow(old4) + length(missing4))) * 100  # % corrected estimate
```

```
## [1] 83.9722
```

Summary of estimates of completeness of Storify data set within NGSS chat sessions

The storify data is incomplete within rush hours. The share to which the storify data set is complete increases slightly when only looking at data during the most busy hours and increases drastically when excluding retweets. This points to the fact that different exclusion criteria were applied. Futhermore, the data is incomplete to similar degrees between all re-downloaded days.

```
## All estimates

## All data

days[1:4]
```

```
##          Day Storify Re_DL Estimate
## 1 2014-06-13     424   635 66.77165
## 2 2014-11-21     337   662 50.90634
## 3 2014-04-18     341   596 57.21477
## 4 2015-12-18     346   574 60.27875
## 5 2015-09-04     418   639 65.41471
## 6 2015-06-05     592   852 69.48357
## 7 2016-02-19     553   861 64.22764
## 8 2016-01-22     488   751 64.98003
## 9 2016-03-04     461   693 66.52237
```

```
sum(days[2])
```

```
## [1] 3960
```

```
sum(days[3])
```

```
## [1] 6263
```

```
(sum(days[2]) / sum(days[3])) * 100
```

```
## [1] 63.22848
```

```
(nrow(old) / (nrow(old) + length(missing))) * 100  # % assuming no tweets have been deleted
```

```
## [1] 62.79734
```

```
## Without retweets

days[c(1, 5:7)]
```

```
##          Day Storify_noRT Re_DL_noRT Estimate2
## 1 2014-06-13          406        451  90.02217
## 2 2014-11-21          321        484  66.32231
## 3 2014-04-18          333        433  76.90531
## 4 2015-12-18          346        480  72.08333
## 5 2015-09-04          418        488  85.65574
## 6 2015-06-05          592        640  92.50000
## 7 2016-02-19          553        684  80.84795
## 8 2016-01-22          488        609  80.13136
## 9 2016-03-04          461        537  85.84730
```

```
sum(days[5])
```

```
## [1] 3918
```

```
sum(days[6])
```

```
## [1] 4806
```

```
(sum(days[5]) / sum(days[6])) * 100
```

```
## [1] 81.5231
```

```
(nrow(old2) / (nrow(old2) + length(missing2))) * 100  # % assuming no tweets have been deleted
```

```
## [1] 80.83351
```

```
## Without most busy hours

days[2] <- old_n3
```

```
days[3] <- new_n3
days[4] <- days[2] / days[3]
days[1:4]
```

```
##          Day Storify Re_DL  Estimate
## 1 2014-06-13     424   594 0.7138047
## 2 2014-11-21     303   587 0.5161840
## 3 2014-04-18     341   552 0.6177536
## 4 2015-12-18     316   497 0.6358149
## 5 2015-09-04     418   603 0.6932007
## 6 2015-06-05     592   790 0.7493671
## 7 2016-02-19     491   734 0.6689373
## 8 2016-01-22     444   622 0.7138264
## 9 2016-03-04     459   588 0.7806122
```

```
sum(days[2])
```

```
## [1] 3788
```

```
sum(days[3])
```

```
## [1] 5567
```

```
(sum(days[2]) / sum(days[3])) * 100
```

```
## [1] 68.04383
```

```
(nrow(old3) / (nrow(old3) + length(missing3))) * 100  # % assuming no tweets have been deleted
```

```
## [1] 67.52228
```

```
## Without retweets nor most busy hours

days[2] <- old_n4
days[3] <- new_n4
days[4] <- days[2] / days[3]
days[1:4]
```

```
##          Day Storify Re_DL  Estimate
## 1 2014-06-13     406   441 0.9206349
## 2 2014-11-21     287   440 0.6522727
## 3 2014-04-18     333   412 0.8082524
## 4 2015-12-18     316   432 0.7314815
## 5 2015-09-04     418   471 0.8874735
## 6 2015-06-05     592   619 0.9563813
## 7 2016-02-19     491   608 0.8075658
## 8 2016-01-22     444   532 0.8345865
## 9 2016-03-04     459   465 0.9870968
```

```
sum(days[2])
```

```
## [1] 3746
```

```
sum(days[3])
```

```
## [1] 4420
```

```
(sum(days[2]) / sum(days[3])) * 100
```

```
## [1] 84.75113
```

```
(nrow(old4) / (nrow(old4) + length(missing4))) * 100  # % assuming no tweets have been deleted
```

```
## [1] 83.9722
```

Considering different possible criteria of exclusion by Storify administrators

```
## Further analyzing tweets, that could possibly have been sorted out by
## certain criteria by Storify administrators

## Sorted out is defined as existing in re-downloads but not in Storify data set
## Not sorted out is defined as existing in Storify data set

sorted_out <- new[!new$status_id %in% old$status_id,]
not_sorted_out <- old

## Possibility: Retweets have been sorted out predominantly by being a retweet

## Get retweets by the variable is_retweet as well as pattern search in text
## (manual retweeting with RT @)

c(which(sorted_out$is_retweet==T), grep("RT @", sorted_out$text)) %>%
  unique() %>%
  length() * 100 /
  nrow(sorted_out)
```

```
## [1] 60.40068
```

```
c(which(not_sorted_out$is_retweet==T), grep("RT @", not_sorted_out$text)) %>%
  unique() %>%
  length() * 100 /
  nrow(not_sorted_out)
```

```
## [1] 1.060606
```

```
## Conclusion: 60.4% of tweets that were sorted out were retweets. Only
## 1.1% of tweets that were not sorted out were retweets. Still, there were
## possibly more criteria of exclusion.

## Possibility: Tweets that were not posted between 1 and 3 UTC (most
## activity) were sorted out

which(!hour(sorted_out$created_at) %in% 1:2) %>% length() * 100 / nrow(sorted_out)
```

```
## [1] 22.33589
```

```
which(!hour(not_sorted_out$created_at) %in% 1:2) %>% length() * 100 / nrow(not_sorted_out)
```

```
## [1] 4.343434
```

```
## Conclusion: 22.3% of tweets that were sorted out were not posted between.
## 1 and 3 UTC. Only 4.3% of tweets that were not sorted out were not posted between
## 1 and 3 UTC. Still, there were possibly more criteria of exclusion.

## Excluding all retweets and posts outside of 1-3 UTC from sorted out tweets
## for further analysis
```

```r
rt_index <- c(which(sorted_out$is_retweet==T), grep("RT @", sorted_out$text)) %>% unique()
time_index <- which(!hour(sorted_out$created_at) %in% 1:2)

sorted_out2 <- sorted_out[-unique(c(rt_index, time_index)),]

(1 - nrow(sorted_out2)  / nrow(sorted_out)) * 100
```

```
## [1] 69.52259
```

```r
## Retweets and timeframe 1-3 UTC explains 69.5% of all sorted out tweets

## Possibility: Short tweets with no rich content have been sorted out

not_sorted_out$text %>% na.omit() %>% nchar() %>% mean()
```

```
## [1] 98.09318
```

```r
not_sorted_out$text %>% na.omit() %>% nchar() %>% sd()
```

```
## [1] 33.54303
```

```r
not_sorted_out$text %>% na.omit() %>% nchar() %>% summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   71.00  103.00   98.09  129.00  151.00
```

```r
sorted_out2$text %>% na.omit() %>% nchar() %>% mean()
```

```
## [1] 91.57343
```

```r
sorted_out2$text %>% na.omit() %>% nchar() %>% sd()
```

```
## [1] 35.83203
```

```r
sorted_out2$text %>% na.omit() %>% nchar() %>% summary()
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   23.00   59.00   94.00   91.57  126.00  151.00
```

```r
## Conclusion: Length of tweets does not seem to have been
## an exclusion criterion for Storify administrators

## Possibility: Certain users were systematically excluded

users_in_sorted_out2 <- sorted_out2$user_id %>% unique()
users_in_not_sorted_out <- not_sorted_out$user_id %>% unique()

## User that were only in the sorted out tweets

possibly_excluded_users <- base::setdiff(users_in_sorted_out2, users_in_not_sorted_out)

## Get amount of tweets posted by these users in sorted out tweets

amount <- c()

for (i in 1:length(possibly_excluded_users)){
  amount <- c(amount, length(grep(possibly_excluded_users[i], sorted_out2$user_id)))
}
```

```
data.frame(possibly_excluded_users, amount)
```

```
##    possibly_excluded_users amount
## 1               2740332152      1
## 2               2618716136      1
## 3                485723891      1
## 4                242535287      1
## 5                735163872      1
## 6                 18603531      1
## 7                437791317      2
## 8               1388220798      4
## 9                341617430      1
## 10              2988710739      3
## 11              1671862620      1
## 12              3074511560      1
## 13                 7032662      1
## 14              2760083384      1
## 15              1305350636      2
## 16              1093562454      1
## 17               235769765      1
```

```
sum(amount) * 100 / nrow(sorted_out2)
```

```
## [1] 3.356643
```

```
## Excluded users can only account for 3.4% of all excluded tweets
## (without retweets and tweets outside of 1-3 UTC).
## No user with more than 4 tweets has been excluded in this subset.
## It is not likely that certain users have been systematically excluded by
## Storify administrators

## End of algorithmic reconstrucion of exclusion criteria for tweets
```