1. Connect your instance of R to Twitter using the instructions you were given and ensuring that you can debug your installation.

Srinivasa Goteti

[Test OAuth](https://dev.twitter.com/apps/13612804/oauth)

Primary tabs

* [Details](https://apps.twitter.com/app/13612804)
* [Settings](https://apps.twitter.com/app/13612804/settings)
* [Keys and Access Tokens(active tab)](https://apps.twitter.com/app/13612804/keys)
* [Permissions](https://apps.twitter.com/app/13612804/permissions)

Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key) gfEeOcZBTc7opkA9kfOKFNqMj

Consumer Secret (API Secret) wk0HX9bmZrSL8IYLG9fKVOJbqQ7TsT4zrdA0GK5mjEPo9HGGqM

Access Level Read, write, and direct messages ([modify app permissions](https://apps.twitter.com/app/13612804/permissions))

Owner gotetisriharsha

Owner ID 140462594

Top of Form

Application Actions

[Regenerate Consumer Key and Secret](https://apps.twitter.com/app/13612804/recreate_keys)[Change App Permissions](https://apps.twitter.com/app/13612804/permissions)

Bottom of Form

Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

Access Token 140462594-T8cwsEWSrO3Xq7pxTwrZCli5Rxe0ieeBYxl6LaOE

Access Token Secret lrNhamRnMPP9yZ9GKq8ayfdIsSe6uPz1WR1RVvKrDxQqN

Access Level Read, write, and direct messages

Owner gotetisriharsha

Owner ID 140462594

1. Use the card with the Twitter search terms you were given in class.  It contains:
   * the name of a famous person
   * a random search concept (you may turn this into a hashtag for greater efficiency)

R Code:   
  
install.packages("twitteR")

install.packages("wordcloud")

install.packages("tm")

install.packages("qdap")

library(twitteR)

library(wordcloud)

library(tm)

library(qdap)

## Potentially necessary file for Windows

#download.file(url="http://curl.haxx.se/ca/cacert.pem", destfile="cacert.pem")

consumer\_key <- 'gfEeOcZBTc7opkA9kfOKFNqMj'

consumer\_secret <- 'wk0HX9bmZrSL8IYLG9fKVOJbqQ7TsT4zrdA0GK5mjEPo9HGGqM'

access\_token <- '140462594-T8cwsEWSrO3Xq7pxTwrZCli5Rxe0ieeBYxl6LaOE'

access\_secret <- 'lrNhamRnMPP9yZ9GKq8ayfdIsSe6uPz1WR1RVvKrDxQqN'

setup\_twitter\_oauth(consumer\_key, consumer\_secret, access\_token, access\_secret)

## Necessary file for Windows

#download.file(url="http://curl.haxx.se/ca/cacert.pem", destfile="cacert.pem")

################## Text Retrieval and Wordcloud ###############################

#Code (with some changes) from http://davetang.org/muse/2013/04/06/using-the-r\_twitter-package/

#download.file(url="http://davetang.org/muse/2013/04/06/using-the-r\_twitter-package/", destfile = "cacert.pem")

###############################################################################

library(twitteR)

## Looks like the cainfo is no longer needed

DonaldTrump <- searchTwitter("#DonaldTrump", n=500)

mrm <- searchTwitter("#roommate", n=500)

## should get 1500

length(DonaldTrump)

length(mrm)  
  
  
Output:

|  |
| --- |
| > consumer\_key <- 'gfEeOcZBTc7opkA9kfOKFNqMj'  > consumer\_secret <- 'wk0HX9bmZrSL8IYLG9fKVOJbqQ7TsT4zrdA0GK5mjEPo9HGGqM'  > access\_token <- '140462594-T8cwsEWSrO3Xq7pxTwrZCli5Rxe0ieeBYxl6LaOE'  > access\_secret <- 'lrNhamRnMPP9yZ9GKq8ayfdIsSe6uPz1WR1RVvKrDxQqN'  > setup\_twitter\_oauth(consumer\_key, consumer\_secret, access\_token, access\_secret)  [1] "Using direct authentication"  > library(twitteR)  > DonaldTrump <- searchTwitter("#DonaldTrump", n=500)  > mrm <- searchTwitter("#roommate", n=500)  > length(DonaldTrump)  [1] 500  > length(mrm)  [1] 500 |
|  |
| |  | | --- | | > | |

1. Search Twitter for the profile of the famous person.  In their last 1,000 tweets, which hashtags did they use most frequently?  Display a histogram.

R Code:   
  
mrm2 = userTimeline("realDonaldTrump",n=1000)

tw=twListToDF(mrm2)

vec=tw$text

extract.hashes = function(vec){

## hash.pattern = "#[[:alpha:]]+"

hash.pattern = "#[[:alnum:]]+"

have.hash = grep(x = vec, pattern = hash.pattern)

hash.matches = gregexpr(pattern = hash.pattern,

text = vec[have.hash])

extracted.hash = regmatches(x = vec[have.hash], m = hash.matches)

df = data.frame(table(tolower(unlist(extracted.hash))))

colnames(df) = c("tag","freq")

df = df[order(df$freq,decreasing = TRUE),]

return(df)

}

dat = head(extract.hashes(vec),50)

dat2 = transform(dat,tag = reorder(tag,freq))

library(ggplot2)

p = ggplot(dat2, aes(x = tag, y = freq)) + geom\_bar(stat = "identity", fill = "blue")

p + coord\_flip() + labs(title = "Hashtag frequencies in the tweets of the Obama team (@BarackObama)")  
  
  
Output:   
  
  
> mrm2 = userTimeline("realDonaldTrump",n=1000)

> tw=twListToDF(mrm2)

> vec=tw$text

> extract.hashes = function(vec){

+

+ ## hash.pattern = "#[[:alpha:]]+"

+ hash.pattern = "#[[:alnum:]]+"

+ have.hash = grep(x = vec, pattern = hash.pattern)

+

+ hash.matches = gregexpr(pattern = hash.pattern,

+ text = vec[have.hash])

+ extracted.hash = regmatches(x = vec[have.hash], m = hash.matches)

+

+ df = data.frame(table(tolower(unlist(extracted.hash))))

+ colnames(df) = c("tag","freq")

+ df = df[order(df$freq,decreasing = TRUE),]

+ return(df)

+ }

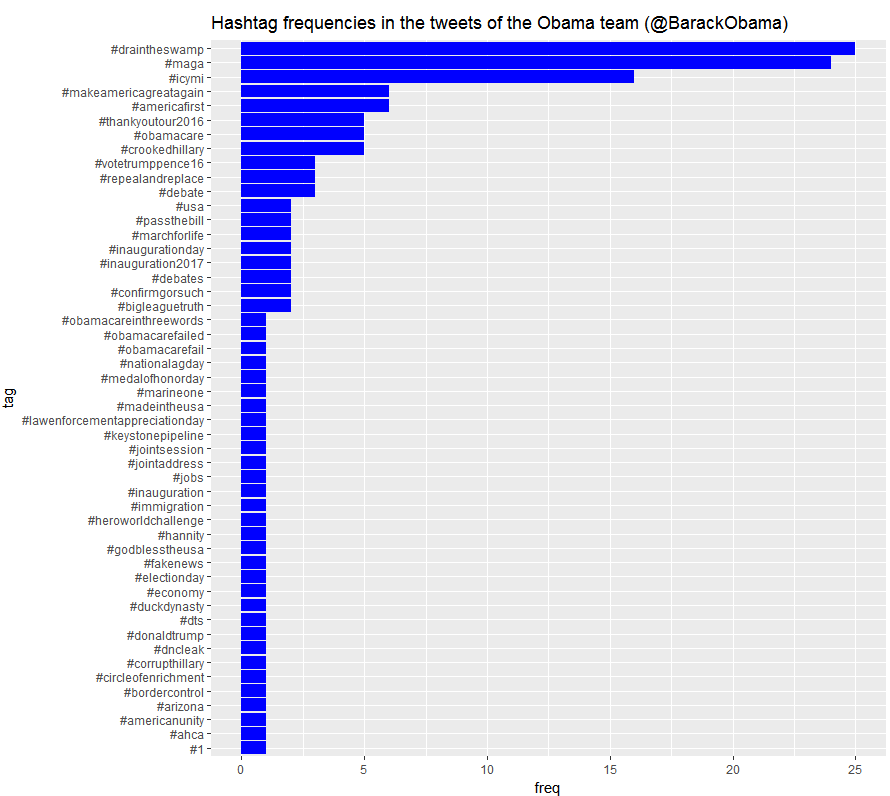
> dat = head(extract.hashes(vec),50)

> dat2 = transform(dat,tag = reorder(tag,freq))

> library(ggplot2)

> p = ggplot(dat2, aes(x = tag, y = freq)) + geom\_bar(stat = "identity", fill = "blue")

> p + coord\_flip() + labs(title = "Hashtag frequencies in the tweets of the Obama team (@BarackObama)")



1. Search Twitter for the hashtag you have built from the card you were given.  You should obtain at least 300 tweets.  Then build a wordcloud.

Include your R code and your output with your submission.

R Code:   
  
  
install.packages("twitteR")

install.packages("wordcloud")

install.packages("tm")

install.packages("qdap")

library(twitteR)

library(wordcloud)

library(tm)

library(qdap)

## Potentially necessary file for Windows

#download.file(url="http://curl.haxx.se/ca/cacert.pem", destfile="cacert.pem")

consumer\_key <- 'gfEeOcZBTc7opkA9kfOKFNqMj'

consumer\_secret <- 'wk0HX9bmZrSL8IYLG9fKVOJbqQ7TsT4zrdA0GK5mjEPo9HGGqM'

access\_token <- '140462594-T8cwsEWSrO3Xq7pxTwrZCli5Rxe0ieeBYxl6LaOE'

access\_secret <- 'lrNhamRnMPP9yZ9GKq8ayfdIsSe6uPz1WR1RVvKrDxQqN'

setup\_twitter\_oauth(consumer\_key, consumer\_secret, access\_token, access\_secret)

## Necessary file for Windows

#download.file(url="http://curl.haxx.se/ca/cacert.pem", destfile="cacert.pem")

################## Text Retrieval and Wordcloud ###############################

#Code (with some changes) from http://davetang.org/muse/2013/04/06/using-the-r\_twitter-package/

#download.file(url="http://davetang.org/muse/2013/04/06/using-the-r\_twitter-package/", destfile = "cacert.pem")

###############################################################################

library(twitteR)

## Looks like the cainfo is no longer needed

DonaldTrump <- searchTwitter("#DonaldTrump", n=500)

mrm <- searchTwitter("#roommate", n=500)

## should get 1500

length(DonaldTrump)

length(mrm)

## [1] 1500

str(DonaldTrump [1:1])

str(mrm [1:1])

## This is what the Twitter attributes look like. Awesome, huh?

## Save text

DonaldTrump\_text <- sapply(DonaldTrump, function(x) x$getText())

mrm\_text <- sapply(mrm, function(x) x$getText())

## Create corpus

mrm\_text\_corpus <- Corpus(VectorSource(mrm\_text))

DonaldTrump\_text\_corpus <- Corpus(VectorSource(DonaldTrump\_text))

## Clean up

## Transform special characters to readable latin1

mrm\_text\_corpus$text <- sapply(mrm\_text\_corpus$text,function(row) iconv(row, "latin1", "ASCII", sub=""))

DonaldTrump\_text\_corpus$text <- sapply(DonaldTrump\_text\_corpus$text,function(row) iconv(row, "latin1", "ASCII", sub=""))

## Make all lowercase

DonaldTrump\_text\_corpus <- tm\_map(DonaldTrump\_text\_corpus, content\_transformer(tolower))

mrm\_text\_corpus <- tm\_map(mrm\_text\_corpus, content\_transformer(tolower))

## Remove punctuation

DonaldTrump\_text\_corpus <- tm\_map(DonaldTrump\_text\_corpus, removePunctuation)

mrm\_text\_corpus <- tm\_map(mrm\_text\_corpus, removePunctuation)

## Remove stopwords

DonaldTrump\_text\_corpus <- tm\_map(DonaldTrump\_text\_corpus, function(x)removeWords(x,stopwords()))

mrm\_text\_corpus <- tm\_map(mrm\_text\_corpus, function(x)removeWords(x,stopwords()))

## Tadah! Build the wordcloud and ignore the errors!

wordcloud(DonaldTrump\_text\_corpus)

wordcloud(mrm\_text\_corpus)  
  
  
  
Output:

|  |
| --- |
| > str(DonaldTrump [1:1])  List of 1  $ :Reference class 'status' [package "twitteR"] with 17 fields  ..$ text : chr "RT @Your\_Name\_Here\_: #DonaldTrump is STILL buying ROBOTS to fill his #Twitter Page Lost #Followers Get REAL fans at https://t.c"| \_\_truncated\_\_  ..$ favorited : logi FALSE  ..$ favoriteCount: num 0  ..$ replyToSN : chr(0)  ..$ created : POSIXct[1:1], format: "2017-04-01 02:54:35"  ..$ truncated : logi FALSE  ..$ replyToSID : chr(0)  ..$ id : chr "848005643403313152"  ..$ replyToUID : chr(0)  ..$ statusSource : chr "<a href=\"http://twitter.com/download/android\" rel=\"nofollow\">Twitter for Android</a>"  ..$ screenName : chr "ToonCheek\_Back"  ..$ retweetCount : num 308  ..$ isRetweet : logi TRUE  ..$ retweeted : logi FALSE  ..$ longitude : chr(0)  ..$ latitude : chr(0)  ..$ urls :'data.frame': 1 obs. of 5 variables:  .. ..$ url : chr "https://t.co/FVWtEt2vpL"  .. ..$ expanded\_url: chr "http://mohrpublicitycart.com/shop/10000-twitter-followers-added-to-your-twitter-account-fast/"  .. ..$ display\_url : chr "mohrpublicitycart.com/shop/10000-twi…""| \_\_truncated\_\_  .. ..$ start\_index : num 116  .. ..$ stop\_index : num 139  ..and 53 methods, of which 39 are possibly relevant:  .. getCreated, getFavoriteCount, getFavorited, getId, getIsRetweet, getLatitude, getLongitude, getReplyToSID, getReplyToSN, getReplyToUID,  .. getRetweetCount, getRetweeted, getRetweeters, getRetweets, getScreenName, getStatusSource, getText, getTruncated, getUrls, initialize,  .. setCreated, setFavoriteCount, setFavorited, setId, setIsRetweet, setLatitude, setLongitude, setReplyToSID, setReplyToSN, setReplyToUID,  .. setRetweetCount, setRetweeted, setScreenName, setStatusSource, setText, setTruncated, setUrls, toDataFrame, toDataFrame#twitterObj  > str(mrm [1:1])  List of 1  $ :Reference class 'status' [package "twitteR"] with 17 fields  ..$ text : chr "RT @LDW\_sg: Do you like #leedongwook 's unique cooking skills?\xed<U+00A0><U+00BD>\xed<U+00B8>\u0082Watch him battle with chick"| \_\_truncated\_\_  ..$ favorited : logi FALSE  ..$ favoriteCount: num 0  ..$ replyToSN : chr(0)  ..$ created : POSIXct[1:1], format: "2017-04-01 01:26:03"  ..$ truncated : logi FALSE  ..$ replyToSID : chr(0)  ..$ id : chr "847983365680578561"  ..$ replyToUID : chr(0)  ..$ statusSource : chr "<a href=\"http://twitter.com/download/android\" rel=\"nofollow\">Twitter for Android</a>"  ..$ screenName : chr "marsmizal"  ..$ retweetCount : num 30  ..$ isRetweet : logi TRUE  ..$ retweeted : logi FALSE  ..$ longitude : chr(0)  ..$ latitude : chr(0)  ..$ urls :'data.frame': 0 obs. of 4 variables:  .. ..$ url : chr(0)  .. ..$ expanded\_url: chr(0)  .. ..$ dispaly\_url : chr(0)  .. ..$ indices : num(0)  ..and 53 methods, of which 39 are possibly relevant:  .. getCreated, getFavoriteCount, getFavorited, getId, getIsRetweet, getLatitude, getLongitude, getReplyToSID, getReplyToSN, getReplyToUID,  .. getRetweetCount, getRetweeted, getRetweeters, getRetweets, getScreenName, getStatusSource, getText, getTruncated, getUrls, initialize,  .. setCreated, setFavoriteCount, setFavorited, setId, setIsRetweet, setLatitude, setLongitude, setReplyToSID, setReplyToSN, setReplyToUID,  .. setRetweetCount, setRetweeted, setScreenName, setStatusSource, setText, setTruncated, setUrls, toDataFrame, toDataFrame#twitterObj  > DonaldTrump\_text <- sapply(DonaldTrump, function(x) x$getText())  > mrm\_text <- sapply(mrm, function(x) x$getText())  > mrm\_text\_corpus <- Corpus(VectorSource(mrm\_text))  > DonaldTrump\_text\_corpus <- Corpus(VectorSource(DonaldTrump\_text))  > mrm\_text\_corpus$text <- sapply(mrm\_text\_corpus$text,function(row) iconv(row, "latin1", "ASCII", sub=""))  > DonaldTrump\_text\_corpus$text <- sapply(DonaldTrump\_text\_corpus$text,function(row) iconv(row, "latin1", "ASCII", sub=""))  > DonaldTrump\_text\_corpus <- tm\_map(DonaldTrump\_text\_corpus, content\_transformer(tolower))  Error in FUN(content(x), ...) : invalid input 'A "witch hunt" might be  what your opponent dealt with.  í ¼í¾¶One of these things is...í ¼í¾¶  #haikuaday  #russiagate  #DonaldTrump' in 'utf8towcs'  > mrm\_text\_corpus <- tm\_map(mrm\_text\_corpus, content\_transformer(tolower))  Error in FUN(content(x), ...) :  invalid input 'RT @LDW\_sg: Do you like #leedongwook 's unique cooking skills?í ½í¸‚Watch him battle with chickens in #Roommate &amp; #Goblin @ViuSG  Meet #LDWinSG Aâ€¦' in 'utf8towcs'  > DonaldTrump\_text\_corpus <- tm\_map(DonaldTrump\_text\_corpus, removePunctuation)  > mrm\_text\_corpus <- tm\_map(mrm\_text\_corpus, removePunctuation)  > DonaldTrump\_text\_corpus <- tm\_map(DonaldTrump\_text\_corpus, function(x)removeWords(x,stopwords()))  > mrm\_text\_corpus <- tm\_map(mrm\_text\_corpus, function(x)removeWords(x,stopwords()))  > wordcloud(DonaldTrump\_text\_corpus)  Warning messages:  1: In wordcloud(DonaldTrump\_text\_corpus) :  httpstcofvwtet2vplâ€ could not be fit on page. It will not be plotted.  2: In wordcloud(DonaldTrump\_text\_corpus) :  donaldtrump could not be fit on page. It will not be plotted.  > wordcloud(mrm\_text\_corpus)  Warning messages:  1: In wordcloud(mrm\_text\_corpus) :  homemadeporn could not be fit on page. It will not be plotted.  2: In wordcloud(mrm\_text\_corpus) :  stellar could not be fit on page. It will not be plotted.  3: In wordcloud(mrm\_text\_corpus) :  blowjob could not be fit on page. It will not be plotted.  4: In wordcloud(mrm\_text\_corpus) :  space could not be fit on page. It will not be plotted.  5: In wordcloud(mrm\_text\_corpus) :  shipleyebilirim could not be fit on page. It will not be plotted.  6: In wordcloud(mrm\_text\_corpus) :  looking could not be fit on page. It will not be plotted.  7: In wordcloud(mrm\_text\_corpus) :  httpstcowpi2tus9yc could not be fit on page. It will not be plotted.  8: In wordcloud(mrm\_text\_corpus) :  sharing could not be fit on page. It will not be plotted.  9: In wordcloud(mrm\_text\_corpus) :  disgruntled could not be fit on page. It will not be plotted. |
|  |
| |  | | --- | | > | |

## 