

Twitter Mining in R

Installing packages

```
install.packages("twitteR")
```

```
install.packages("RCurl")
```

```
require("twitteR")
```

```
require("RCurl")
```

Getting twitter keys

create a twitter account and go to the url apps.twitter.com and collect four keys from there

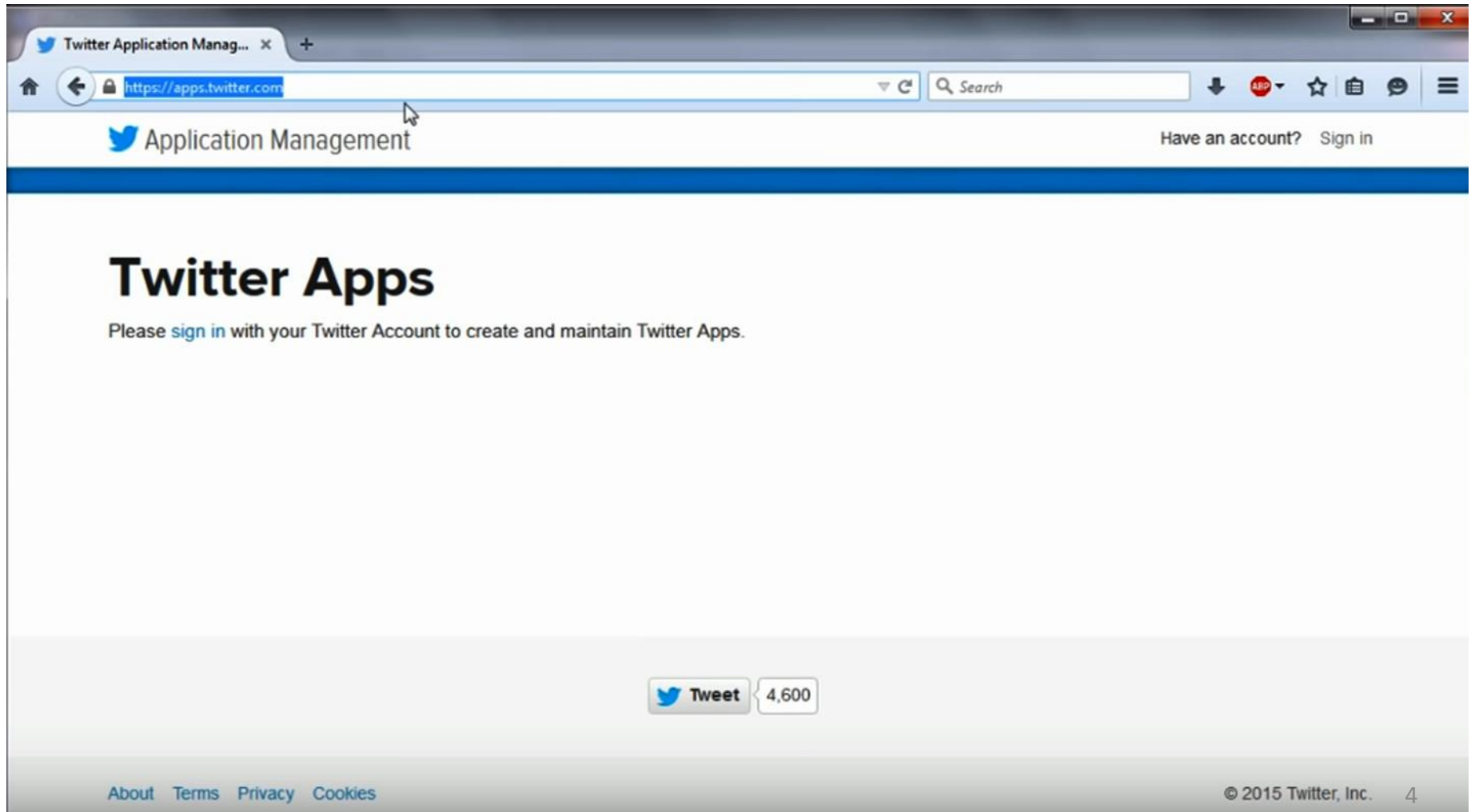
consumer_key

consumer_Secret

access_token

access_secret

Creating twitter App



Creating twitter App

https://apps.twitter.com/app/new

Application Management

Create an application

Application Details

Name *

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

Creating twitter App

R_Twitter_Mining_data

Test OAuth

DetailsSettingsKeys and Access TokensPermissions

Application Settings

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

Consumer Key (API Key)

Consumer Secret (API Secret)

Access LevelRead and write (modify app permissions)

Owner

Owner ID

Application Actions

Regenerate Consumer Key and SecretChange App Permissions

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

Access Token Secret

Setting up twitter keys

```
consumer_key    <-    ""
```

```
consumer_Secret <-    ""
```

```
access_token    <-    ""
```

```
access_secret   <-    ""
```

Authentication & Searching

```
setup_twitter_oauth(consumer_key,consumer_Secret,access_token,access_secret)
```

Now search for liverpool football club

```
LFC_tweets <- searchTwitter("LFC",  
n=10, lang = "en")
```


Showing tweets

LFC_tweets

str(LFC_tweets)

LFC_tweets[1:3]

Showing LFC tweets

A screenshot of an R Console window. The window has a blue title bar with the text 'R Console' and standard window control buttons (minimize, maximize, close) on the right. The main area is white and contains several lines of text representing tweet data. The text is organized into five groups, each starting with an index in square brackets (e.g., [[6]]). Each group contains a single line of text in a dark blue font, which appears to be a tweet. The tweets mention various users and topics related to Liverpool FC (LFC), including mentions of Steven Gerrard and a 'Death star' reference. At the bottom of the console, there is a red prompt character '>' followed by a vertical cursor bar. A horizontal scrollbar is visible at the very bottom of the window.

```
[[6]]  
[1] "AmirAshraf_LFC: RT @SheScreamSteph: Lmfao RT @MonicaLair$  
  
[[7]]  
[1] "Mild_lfc: RT @PFA: VIDEO: @LFC's Steven Gerrard, Kolo To$  
  
[[8]]  
[1] "Mild_lfc: RT @LFC: #LFC captain Steven Gerrard was prese$  
  
[[9]]  
[1] "riye_lfc: 'Death star' welcomes first patients: The UK's$  
  
[[10]]  
[1] "riye_lfc: Campaigns step up with 10 days to go: Scotland$  
  
> |
```

First three tweets from LFC

```
> LFC_tweets[1:3]
```

```
[[1]]
```

```
[1] "NizamuddinAwang: RT @MPBFirmino9: Few weeks back United and Spurs fans were laughing that they would derail #LFC's title challenge and save football by help..."
```

```
[[2]]
```

```
[1] "RhysAshcroft1: I blame @LFC for being so involved in my dreams last night that im now suuuper late for work"
```

```
[[3]]
```

```
[1] "junior4run: Liverpool reserve team have more spectators than Mancity vs Cardiff last night. #YNWA #lfc"
```

Text mining of tweets

Two packages **tm** and **wordcloud** are needed

```
install.packages("tm")
```

```
install.packages("wordcloud")
```

```
require(tm)
```

```
require(wordcloud)
```

Searching tweets

now searching tweets for earthquake

```
himearth <-  
searchTwitter('earthquake+himalaya', lang =  
en", n= 50 , resultType = "recent")
```

```
class(himearth)
```

Converting **list** into **chr**

himearth is a **list** we convert into a **character** vector

```
himearth_text <- sapply(himearth, function(x)  
  x$getText())
```

```
str(himearth_text)
```

```
> str(himearth_text)  
chr [1:500] "Nepal Mountain Villages 'Completely Washed Away' By Quake http://t.co/v\$
```

Making corpus

Corpus is the main structure of managing text

```
him_corpus<- Corpus(VectorSource(himearth_text))  
him_corpus
```

See into the corpus

```
inspect(him_corpus)  
inspect(him_corpus[1])
```

Removing punctuations

now we clean up the corpus by eliminating the unnecessary words like by, and, punctuations etc

removing punctuations

```
him_clean <-  
tm_map(him_corpus,removePunctuation)
```


Uppercase to lowercase

transforming all uppercase letters to lowercase

```
him_clean <-  
tm_map(him_clean,content_transformer(tolower))
```

Removing stopwords, digits, whitespaces

Removing **stopwords** like and is or etc

```
him_clean <-  
tm_map(him_clean,removeWords,stopwords("english"))
```

remove numbers and digits

```
him_clean <- tm_map(him_clean,removeNumbers)
```

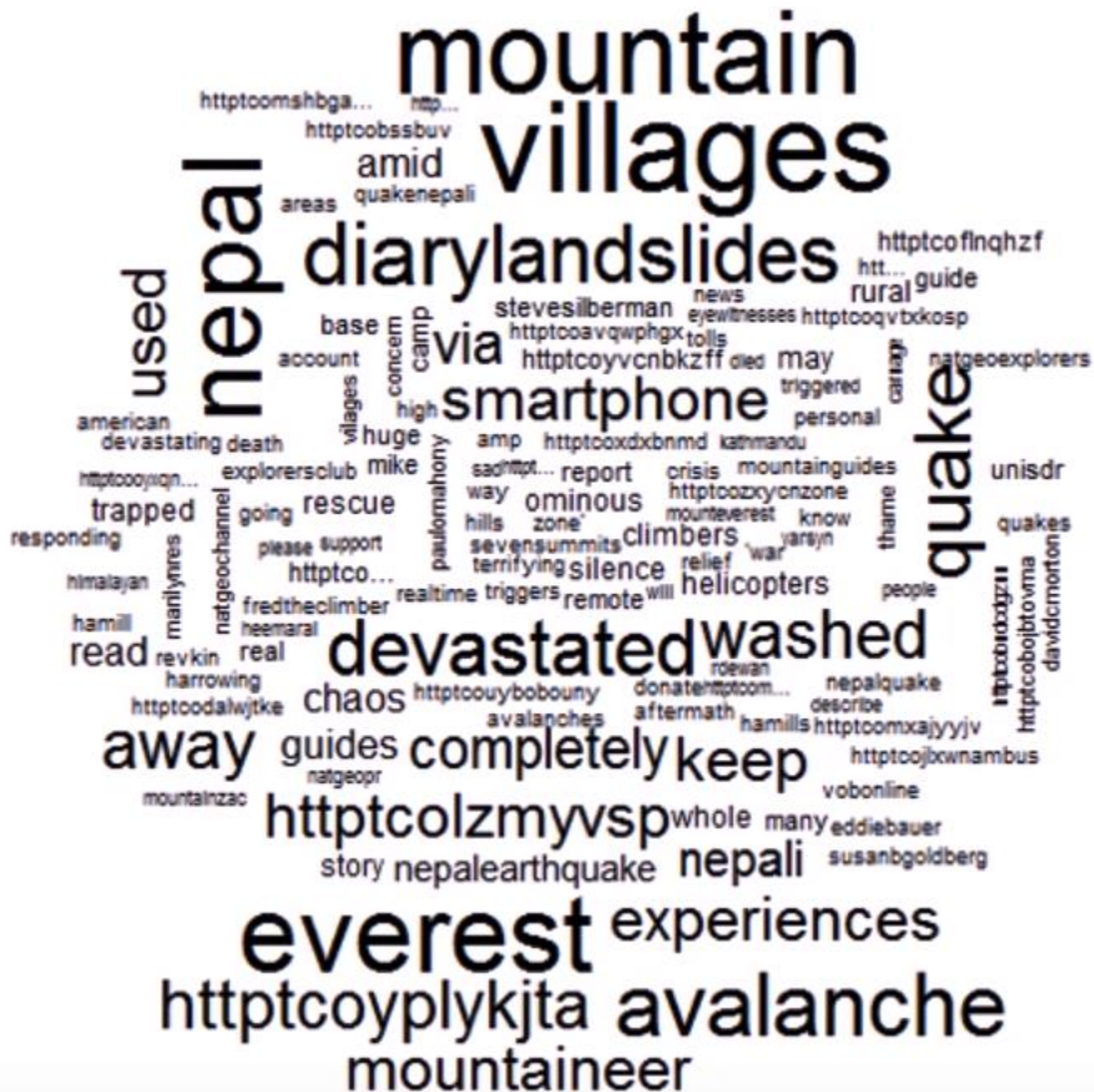
removing whitespace

```
him_clean <- tm_map(him_clean,stripWhitespace)
```

Removing any word

remove himalaya and earthquake b/c it is obvious

```
him_clean <- tm_map(him_clean,removeWords,  
c("himalaya","earthquake"))
```



Graph of words by wordcloud

```
wordcloud(him_clean)
```

```
wordcloud(him_clean,random.order = F)
```

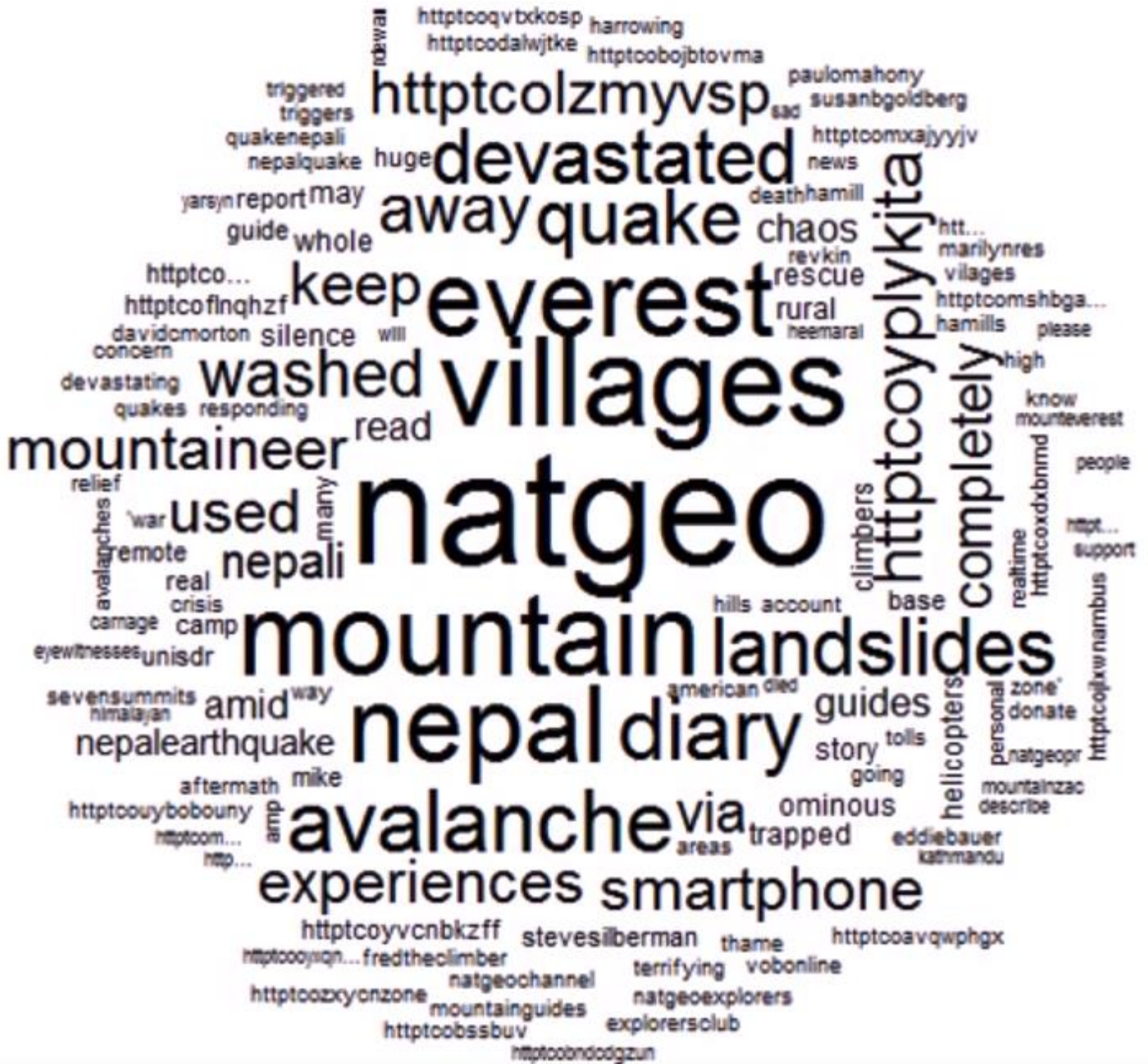
defining scale to enlarge the words min and max

```
wordcloud(him_clean,random.order = F, scale =  
c(3,.5))
```

Cleaning commands

```
> him_clean <- tm_map(him_corpus, removePunctuation)
> him_clean <- tm_map(him_clean, content_transformer(tolower))
> him_clean <- tm_map(him_clean, removeWords, stopwords("english"))
> him_clean <- tm_map(him_clean, removeNumbers)
> him_clean <- tm_map(him_clean, stripWhitespace)
> him_clean <- tm_map(him_clean, removeWords, c("himalaya", "earthquake"))
```


Bigger
words
are
arrang
ed



Rainbow colors

```
wordcloud(him_clean,random.order = F, scale =  
c(6,.5))
```

```
wordcloud(him_clean,random.order = F, scale =  
c(5,2.5))
```

color rainbow effect

```
wordcloud(him_clean,random.order = F, colors =  
rainbow(50))
```


Rainbow colors

defining max number of words

```
wordcloud(him_clean, random.order = F,  
max.words=40, colors = rainbow(50))
```

Rainbow effect

