Assignment 1: Streaming Twitter

In this assignment you will create a twitter app, use it to stream tweets, parse the tweets and store the result. Note: Hyperlinks are highlighted using bold font.

1 Create Project Folder

Open R Studio and create a project folder (File/New Project...) in a suitable place on your harddrive.

2 Acquiring OAuth Credentials

OAuth provides our R software with access to our twitter app and through that to the twitter stream.

2.1 Email Account

Open up a new Email account (e.g., with **Gmail**) and store login and pass in a text file (e.g., using WordPad or TextEdit). Store text file in project folder.

2.2 Twitter Account

Open up a new **twitter account** using your newly created Email account and store login and pass in the text with the mail account credentials. Use whatever name you prefer. Verify your account using your phone.

2.3 Twitter App

Create a **twitter app**. Come up with an app name and description and use http://www.dirkwulff.org in the website field. Next go to *Keys and Access* and copy the *Consumer Key* and *Consumer Secret* into your text file.

3 Streaming Twitter

3.1 First Script

In RStudio open a new R script (File/New File/R Script) and save it in your project folder.

3.2 Install `ROAuth` and `streamR`

Install and load packages `ROAuth` and `streamR` using `install.packages()` and `library()`.

3.3 Setup OAuth

Setup OAuth by passing on the consumer key and secret, as well as the following URLs to `OAuthFactory$new()` and assigning it to `my_oauth` (Note: Accessing a function (or method) as an element of another object is unusual in R but very common in other, more object-oriented languages such as Python.):

- https://api.twitter.com/oauth/request_token
- http://api.twitter.com/oauth/access_token
- http://api.twitter.com/oauth/authorize

Then execute `my_oauth$handshake(cainfo = system.file("CurlSSL", "cacert.pem", package = "RCurl"))` and follow the instructions in the console.

Next save the my_oauth object in the project folder for future purposes using `saveRDS(my_oauth,'mypath/myfilename.RDS')`. When in a new session reload the object using `my_oauth = readRDS('mypath/myfilename.RDS')` rather than conducting a new handshake.

3.4 Stream Twitter

Use `filterStream()` to stream tweets (see `?filterStream`). Store tweets in new object `my_stream` (required `file.name = ""`). Choose a search term of your liking and pass it to the function using the `track` argument. Also make sure to pass on `my_oauth` and set `timeout` to a reasonable duration, e.g., 60(s).

Make sure that you have collected at least a few tweets using `length(my_stream)`

More info on streaming parameters **here**.

4 Processing Tweets

4.1 Install `jsonlite`

Install and load `jsonlite`. You know how.

4.2 Parse JSON

Create an empty list names `parsed_stream`. Iterate over the tweets. At every iteration pass on the individual tweet to `fromJSON()`, extract the elements 'created_at', 'text', 'source', 'lang', 'user$screen_name', 'user$location', 'user$description', 'user$followers_count', 'user$friends_count', 'user$statuses_count', and store a vector of the elements in `parsed_stream`. Note that not every tweet contains all elements.

More info on the content of a tweet **here** and **here**.

4.3 Process Data

Create a `data.frame` named `data_stream` that contains the contents of `parsed_stream`. Elements should occupy the columns and all missing elements shoud be replaced by `NA` (see `?NA`). Requires a loop and if-statements. When ready save `data_stream` in project folder using `saveRDS()` (or `write.csv()`).

End