Heike Deutelmoser
07.05.2017

Assessment-Web Scraping Project

Contents

Introduction 1
Scraping data from Twitter 2
Preparation 2
Scraping process 2
Cleaning the retrieved data 2
Analysis of the data 2
Dictionary 2
Count of word frequence 2
Word replacement 2
Document matrix 3
Word frequency 3
Wordcloud plot 5
Word associations 5
Topic plot 6
Dendrograms 7
Sentiment Analysis 8
Limitations of the Analysis 8

Introduction

In this project we use R to retrieve information from twitter.

Our goal is to retrieve information about swing dance and especially about the swing dance lindy hop in Toronto.

To represent and visualize the data retrieved we will use several different plots such as the sentiment plot and the wordcloud.

Scraping data from Twitter

Preparation

To prepare use to webscrape data from twitter, we define the api key, the api secret, the access token and the access token secret corresponding to our twitter account.

Furthermore, we install the packages we want to use and load their libraries.

Scraping process

Now we can define the search string containing the words for which we want to search in the twitter application. We also define the number of tweets we want to retrieve.

With the function searchTwitter we can finally download the tweets we wanted.

Cleaning the retrieved data

To clean the obtained strings in the variable tweets, we first decide that we want to focus on the twitter user torontolindyhop and filter the first 200 tweets of this user by applying the function userTimeline.

Then, we convert the filtered tweets to a dataframe by applying the function twListToDF. Now, we can look at distinct entries and distinct characteristics of the strings.

Following the cleaning process we apply the function iconv to all rows of the text in the dataframe.

Using the package “tm” (text mining) we can go on with cleaning the data by converting the text to lower case letters, removing URLs, anything other than English letters, stopword, extra whitespaces.

Analysis of the data

Dictionary

First, we want to create a dictionary with the words contained in the text. (results see R code)

Count of word frequency

Second, we want to count the word frequency of the words outdoor and Saturday. We define the function wordFreq which counts the repeats of the word in the text. And obtain that outdoor is used 3 times and Saturday appears 26 times in the text.

Word replacement

We define the function replaceWord and replace the word outdoor with open air and the word Saturday with weekend.

Document matrix

With the function TermDocumentMatrix we see the following results about the text:

```
terms: 572, documents: 199

Non-/sparse entries: 2473/111355

Sparsity           : 98%

Maximal term length: 15
```

And can see in which documents from 21 to 30 the word Saturday is used:

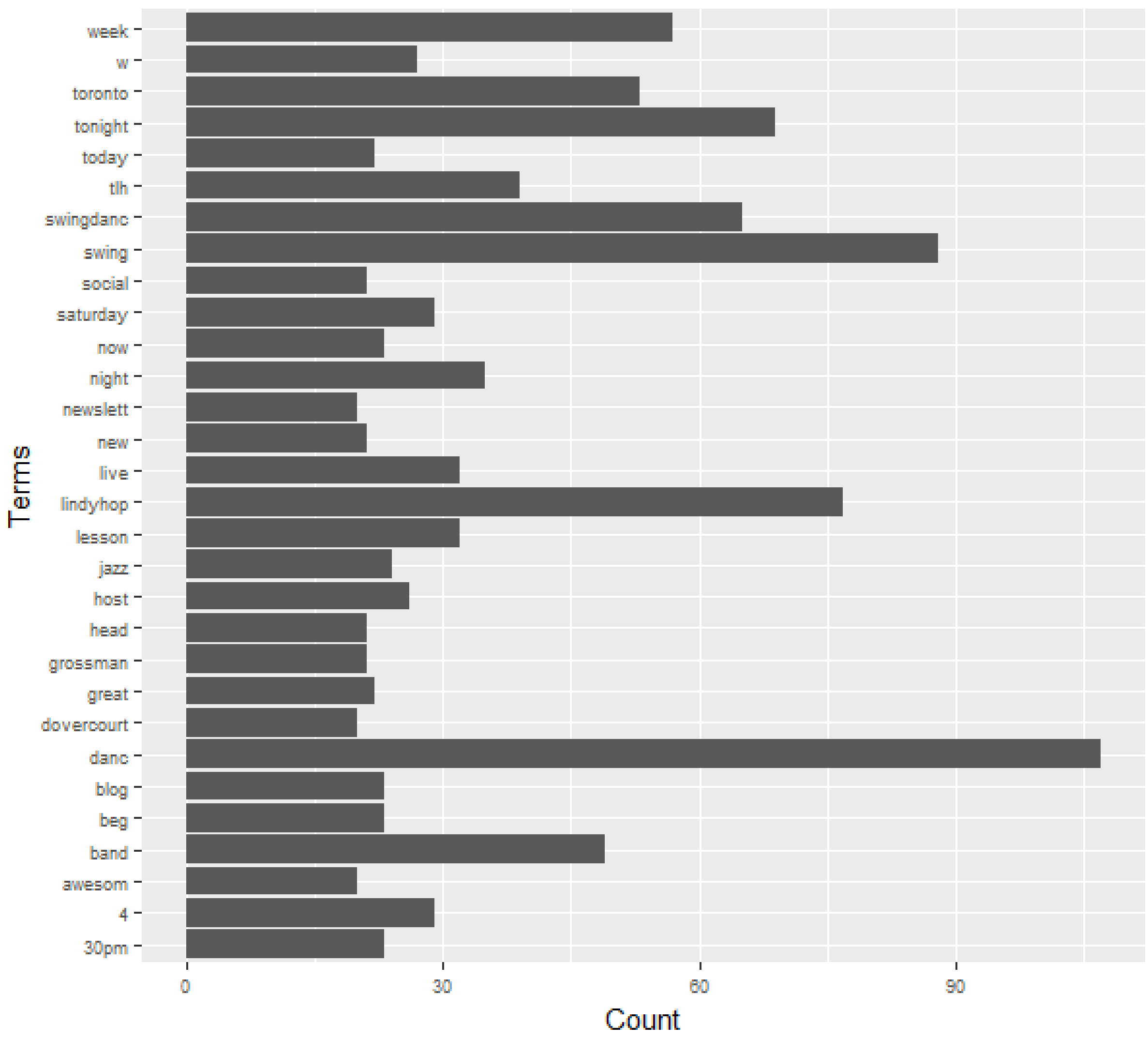
```
Docs

Terms      21 22 23 24 25 26 27 28 29 30

saturday  1  0  0  0  0  0  1  0  0  0
```

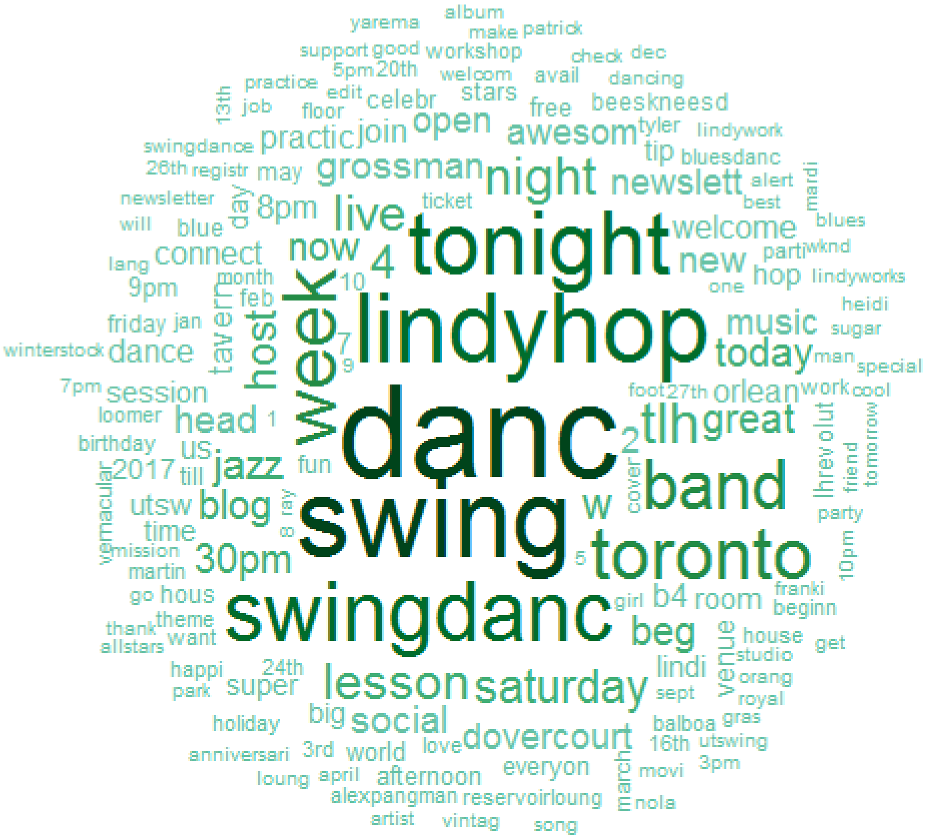
Word frequency

First we search for the most frequent words with the function findFreqTerms and save them with their frequency in a dataframe. Plotted we can visualize the most frequent words with their frequency:


Terms
week
w
toronto
tonight
today
tlh
swingdanc
swing
social
saturday
now
night
newslett
new
live
lindyhop
lesson
jazz
host
head
grossman
great
dovercourt
danc
blog
beg
band
awesom
4
30pm
0
30
60
90
Count

Wordcloud plot

Now, we want to use the function wordcloud and obtain the following plot:



Word associations

Finally we can search for word associations with the function findAssocs.

For the word tonight we obtain the following:

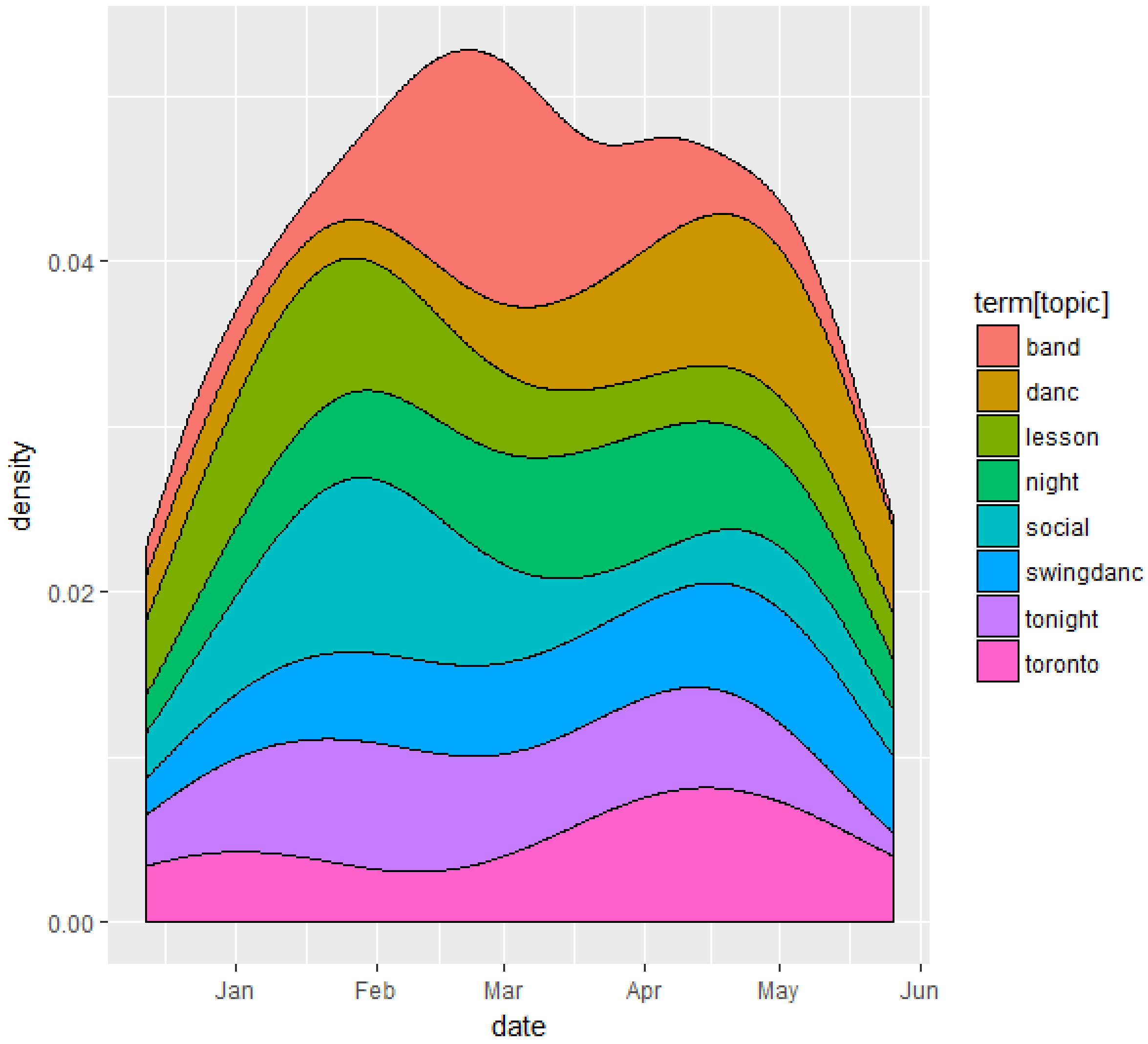
```
$tonight
lesson         beg          host          b4          9pm         social          7     dovercourt
0.42          0.39          0.34          0.34          0.31          0.29          0.29          0.28
night       welcome    alexpangman reservoirloung        miss        semest        saturday          utsw
0.28          0.25          0.24          0.24          0.24          0.24          0.23          0.23
w          hous          tyler          free
0.23          0.22          0.22          0.21
```

Topic plot

First we divide the words in different topis by the LDA function of the package topicmodels and obtain the first seven terms of every topic with the function terms:

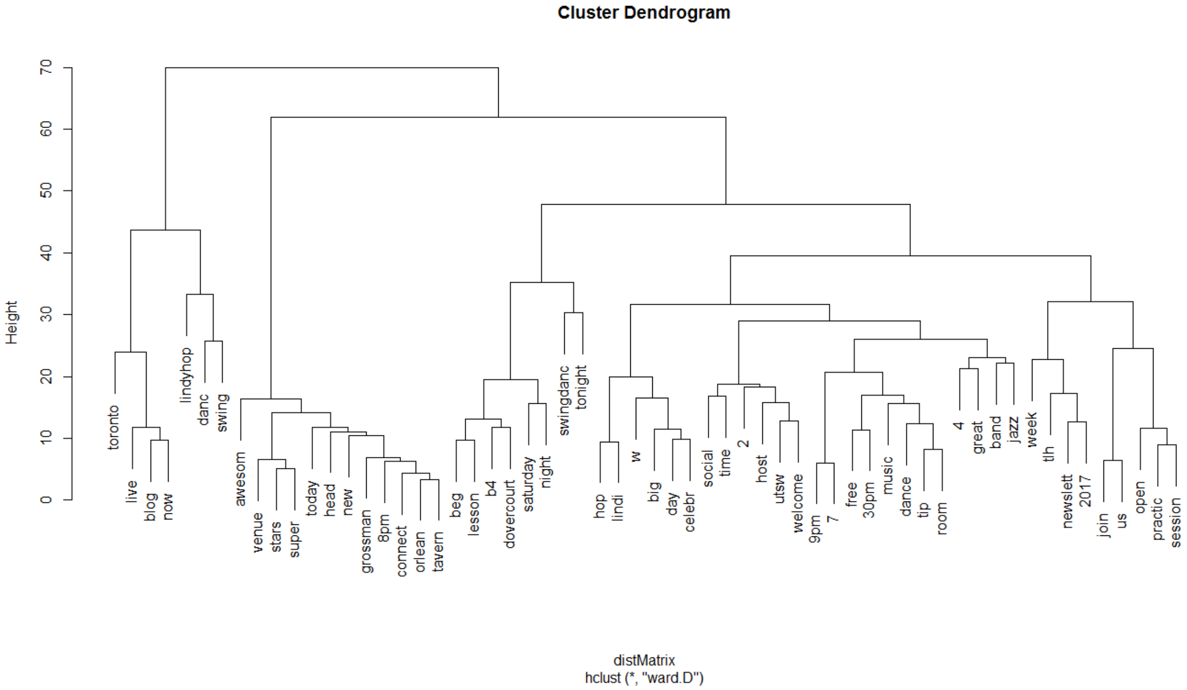
```
Topic 1    Topic 2     Topic 3 Topic 4     Topic 5     Topic 6     Topic 7    Topic 8
[1,] "tonight"   "lesson"     "danc"  "swing"     "swing"     "swing"     "danc"      "lindyhop"
[2,] "swingdanc" "tonight"     "band"  "lindyhop"  "danc"      "danc"       "week"      "band"
[3,] "danc"       "toronto"     "week"  "week"       "lindyhop"  "lindyhop"  "toronto"  "swingdanc"
[4,] "social"     "b4"          "live"  "swingdanc" "swingdanc" "tlh"        "lindyhop" "today"
[5,] "toronto"    "swingdanc"  "tlh"    "toronto"    "tavern"     "tonight"    "live"      "head"
[6,] "band"       "w"           "w"      "danc"       "30pm"       "swingdanc" "newslett" "music"
[7,] "night"      "dovercourt" "jazz"   "blog"       "connect"    "night"      "social"    "awesom"
```

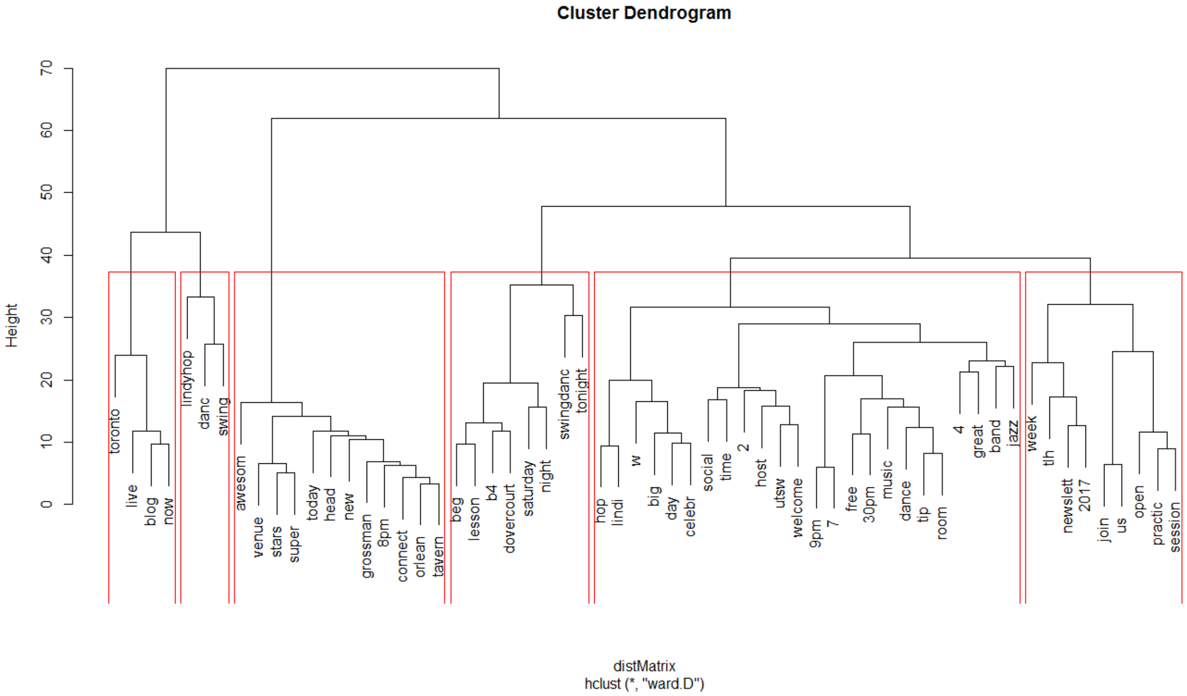
Plotting the topics with ggplot we obtain the following:



Dendrograms

By applying the function hclust and rect.hclust to the cluster terms we obtain the dendrograms:



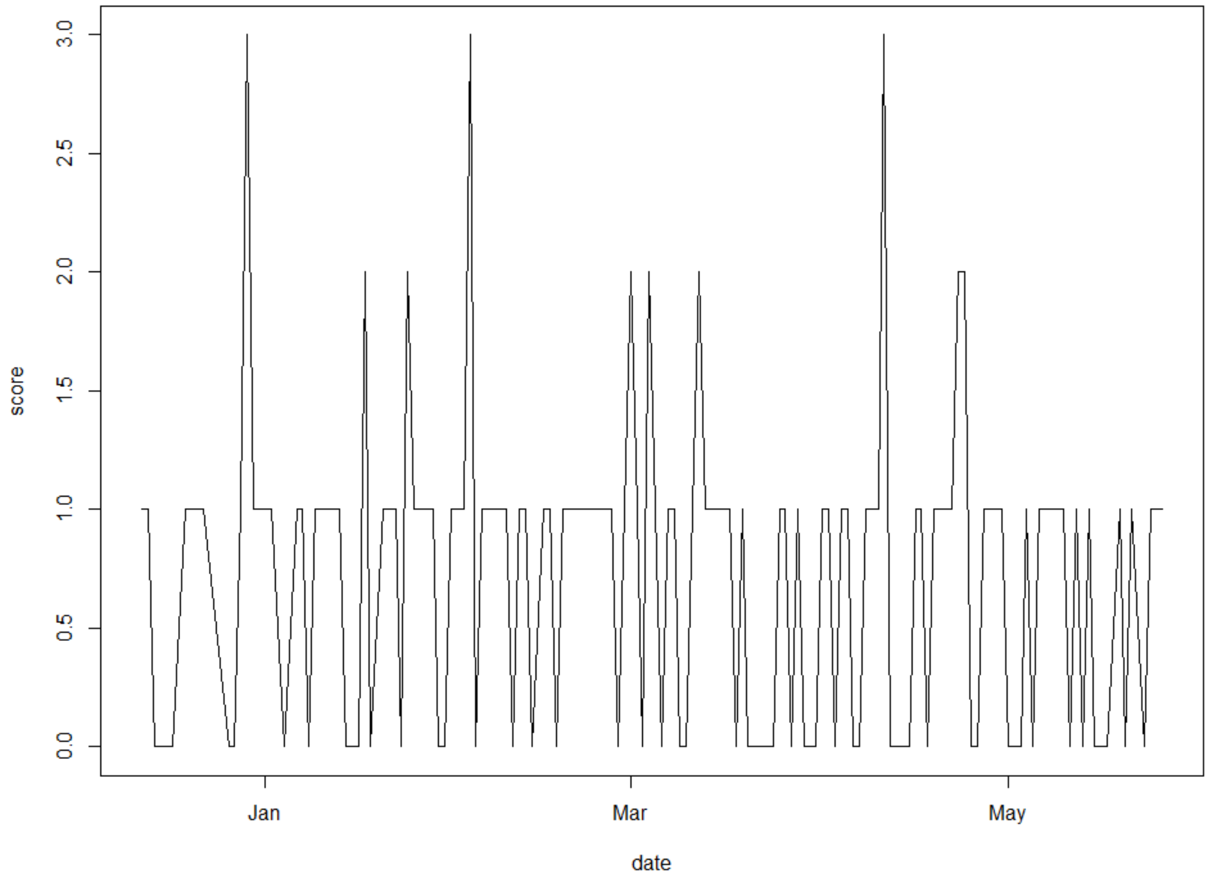


Sentiment Analysis

We obtain the sentiments with the function sentiment of the package sentiment:

```
neutral positive
89      110
```

For the plot we assign the neutral words the value 0 and the positive words the value 1:



Limitations of the Analysis

For further information of the topic lindy hop in Toronto we would need to retrieve data from more users than only one. In a real case I would have been searching for more users referring to lindy hop in Toronto and add them to the analysis or rather don't filter the data too much.