Twitter Data Mining

Tianwen Huan

12/10/2016

This project first used 'filterStream' code to track data including 'Trump' located in United States from Twitter. The original dataset include 28147 observations. Based on this dataset, I first did some simple data clean to make it suitable for later analysis. Based on this original data, I divided the further analysis into five parts:

1. Map

2. Word Cloud

3. Top Retweeted Analysis

4. Sentiment Analysis

5. Time Series

All the dataset and R document I created for this project have been uploaded to my github account. The materials can be found from the link below:

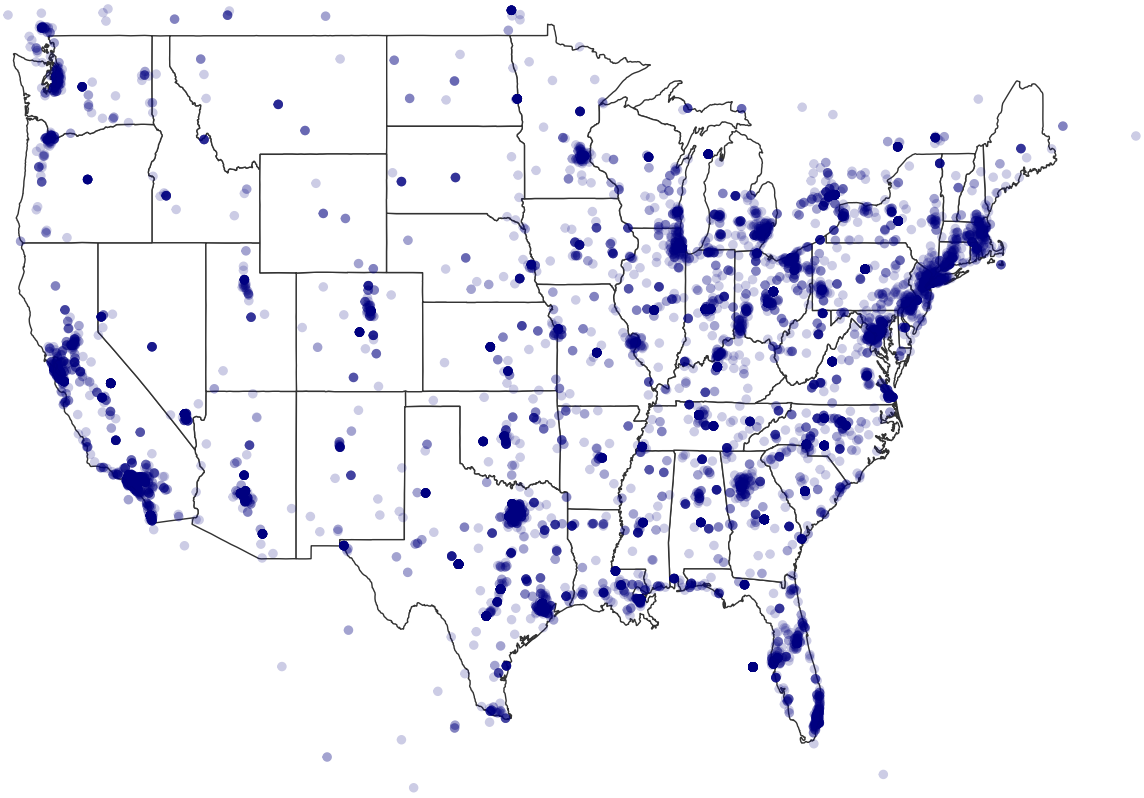
https://github.com/hhtiffany/MA615-Final-Project

1. Map

This map plotted all the available 'place_lat' and 'place_lon' points created from the original dataset. From the map below we can tell that people near the seaside and east pay more attention on Trump in Twitter than people live in the inland.

```
##
## Attaching package: 'maps'
```

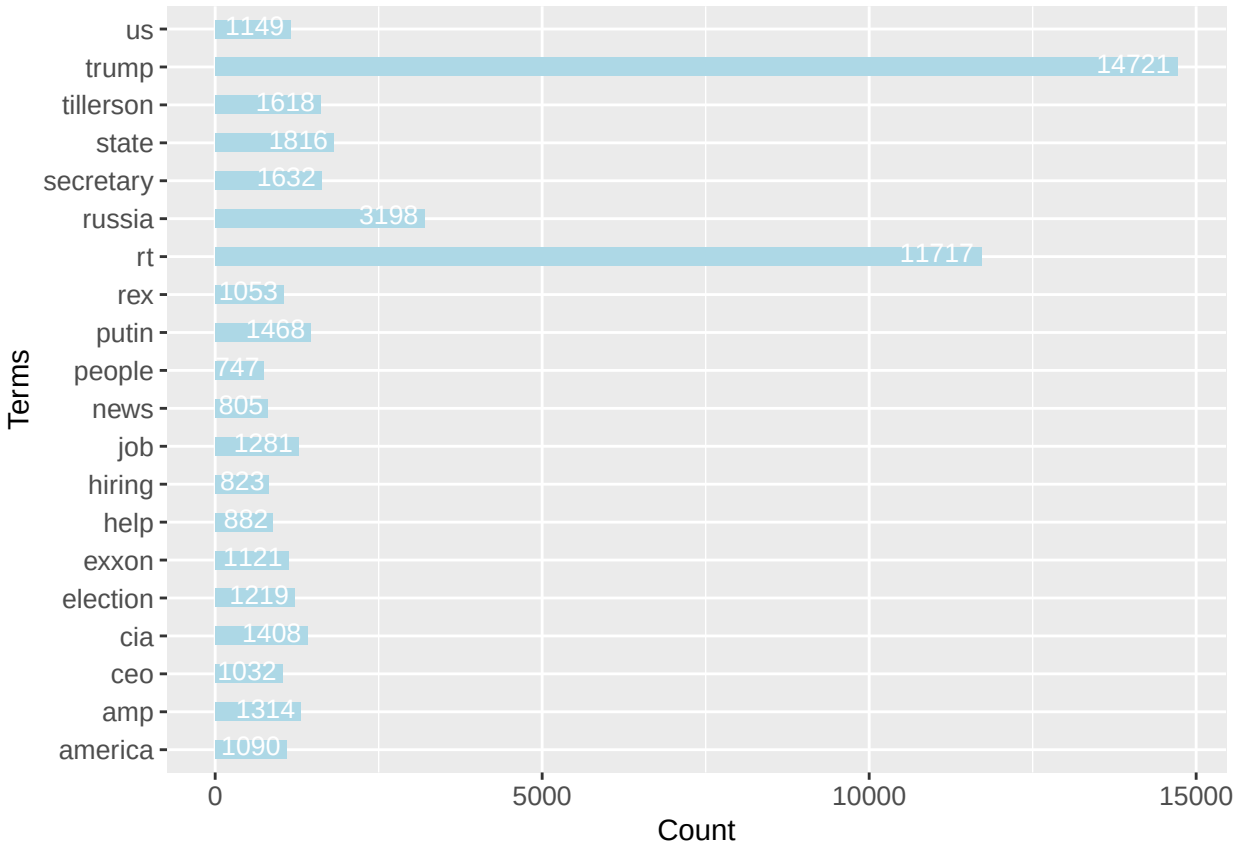
```
## The following object is masked from 'package:plyr':
##
##     ozone
```



2. Word Cloud

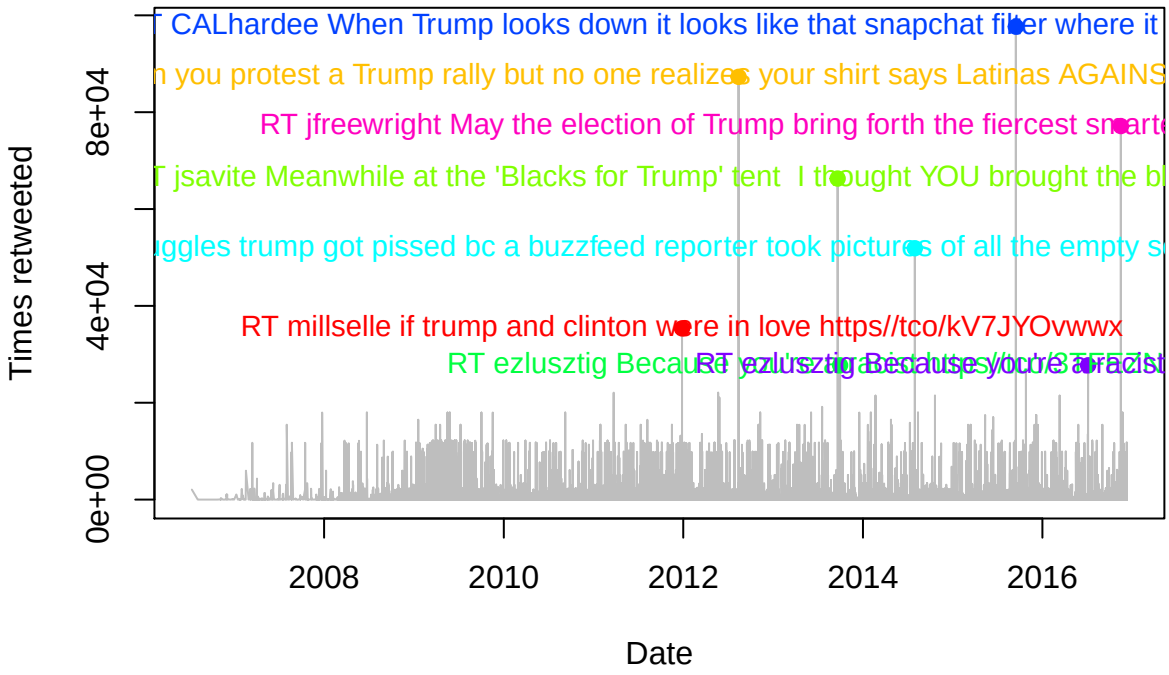
From the picture below we can find that people really interested in 'Job', 'Russia' and 'Sate Security' when they talk about Trump on Twitter.

The barplot below has shown the top 20 popular words people used when they mentioned Trump on Twitter, which has revealed the area people most considerded about, such as 'Job', 'Russia', 'Sate Security' and so on. Just the same as the Word Cloud has shown.



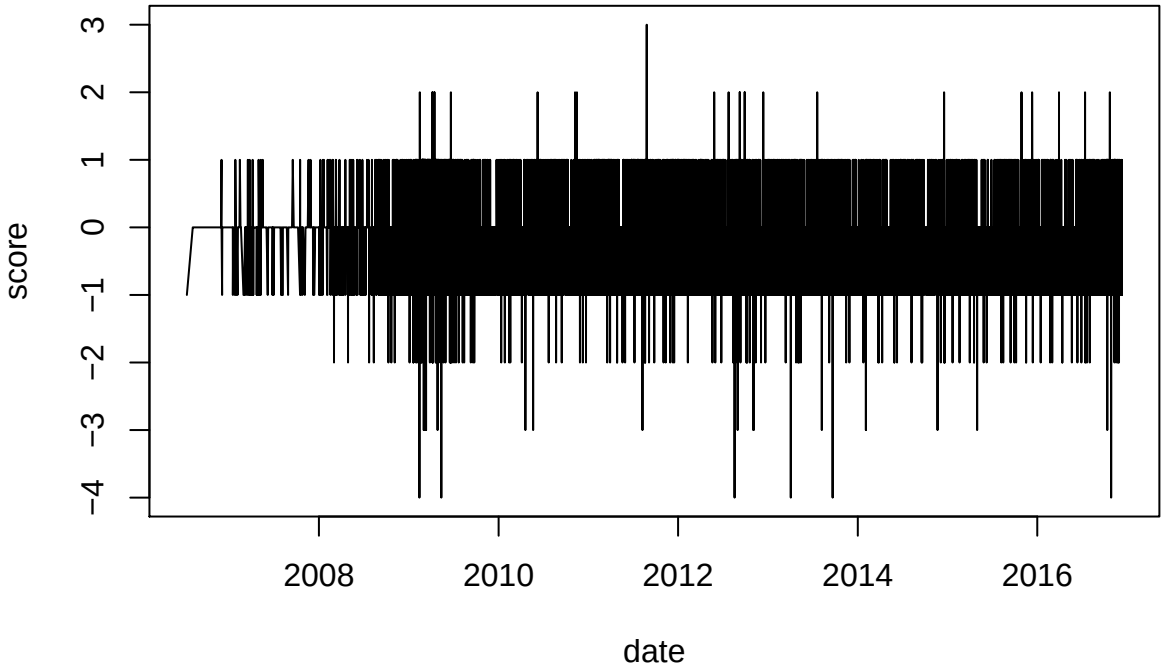
3. Top Retweeted Tweetsop

This plot shows the top retweeted twitters related to Trump in different year. The most retweeted twitters are created in recent years. This is normal because the time passing made old twitter hard to be found again by people.



4. Sentiment Analysis

I used the 'sentiment' function to get the polarity column. After that, I created a 'score' column based on the polarity column, let score equals to 1 if the polarity is positive, -1 if the polarity is negative, otherwise 0. The higher the score, the more positive. The lower the score, the more negative. Then I sumed daily score and created a time series dataset. The result showed a liitle bit more negative than positive.



5. Time Sries for 2016

This plot shows the daily tweets number for 2016.

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

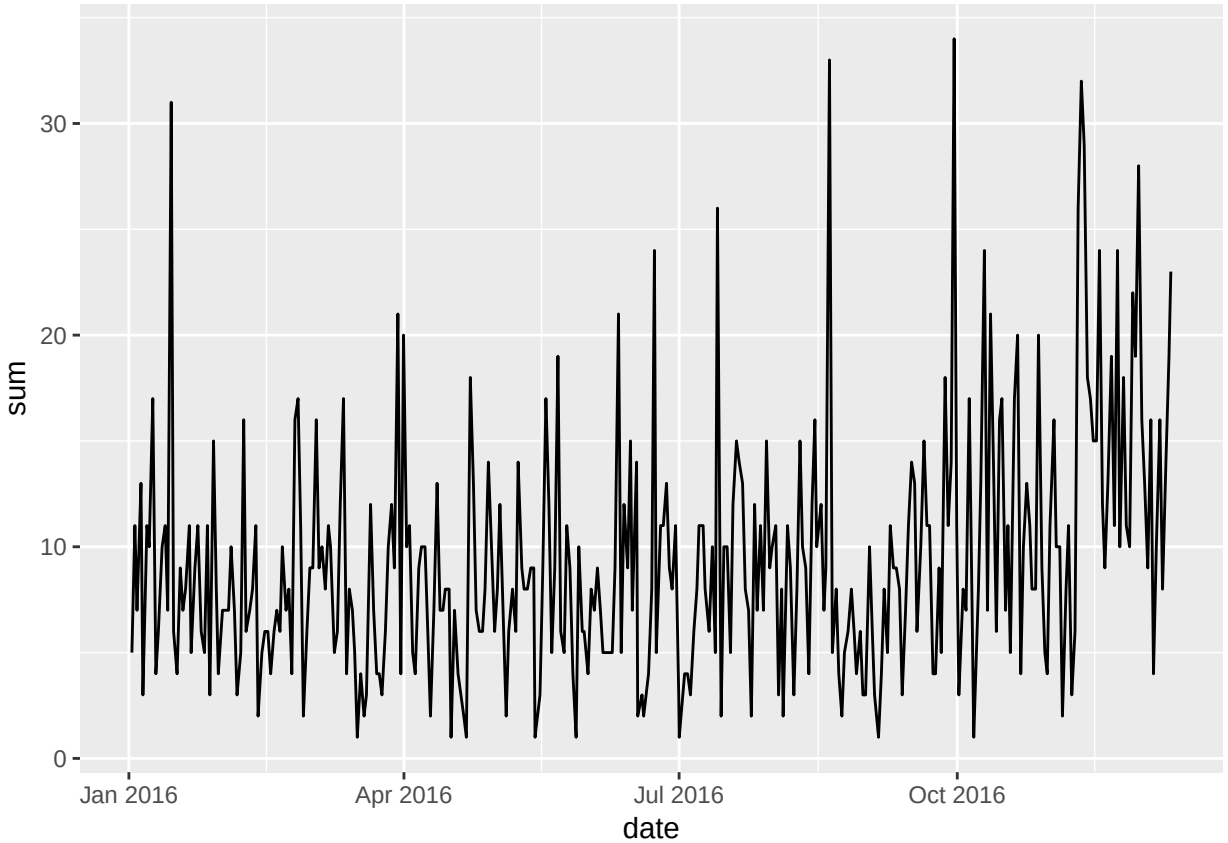
```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

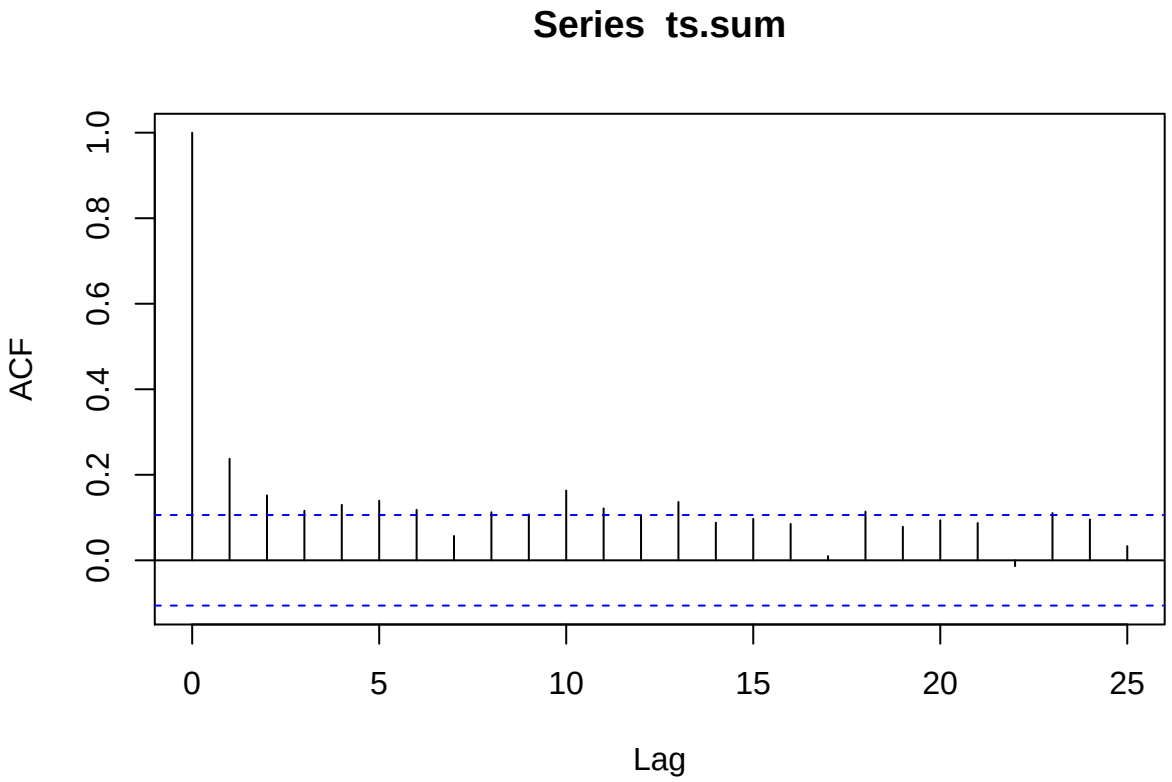
```
##
## Attaching package: 'xts'
```

```
## The following objects are masked from 'package:data.table':
##
##     first, last
```

```
## The following objects are masked from 'package:dplyr':
##
##     first, last
```



The ACF plot helps to check the autocorrelation. The result suggested that there is no significant correlation to be captured by a model. The autocorrelations are within the 5% significance bands.


Series ts.sum
ACF
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25
Lag