# Problem 5:

## Main Task: Create UI to access statistical results

Data: For doing statistical summarization on the R-Shiny, I have chosen the Election topic which is trending with around 5000 tweets per second. The data crawling is done using **SteamR** Twitter streaming API package for R.

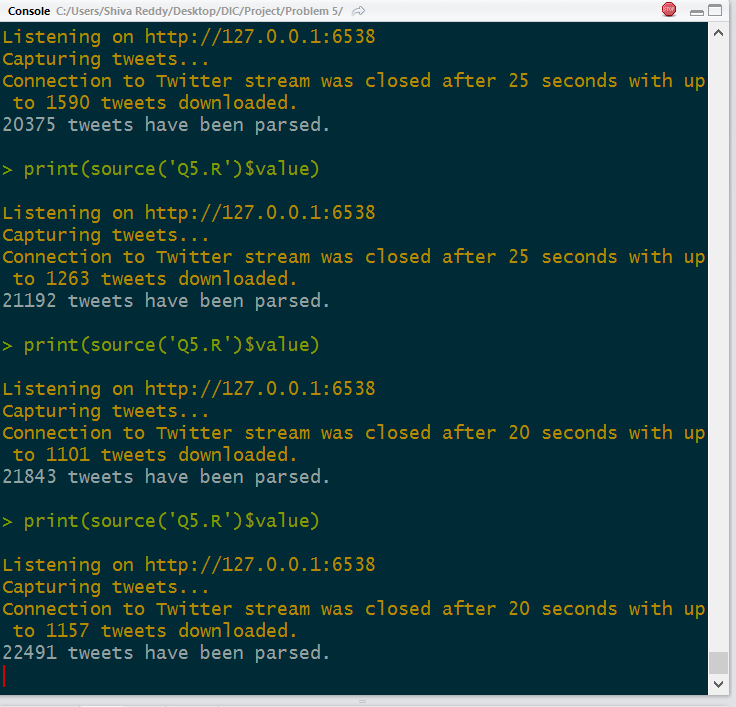
After loading the OAuth, we can search for tweets real-time. The input User-Interface in the R Shiny is as below:



The input slider is used to determine the time to collect the tweets in the Streaming API. We then process the tweets based on how many tweets are relevant and then make a count of those tweets.

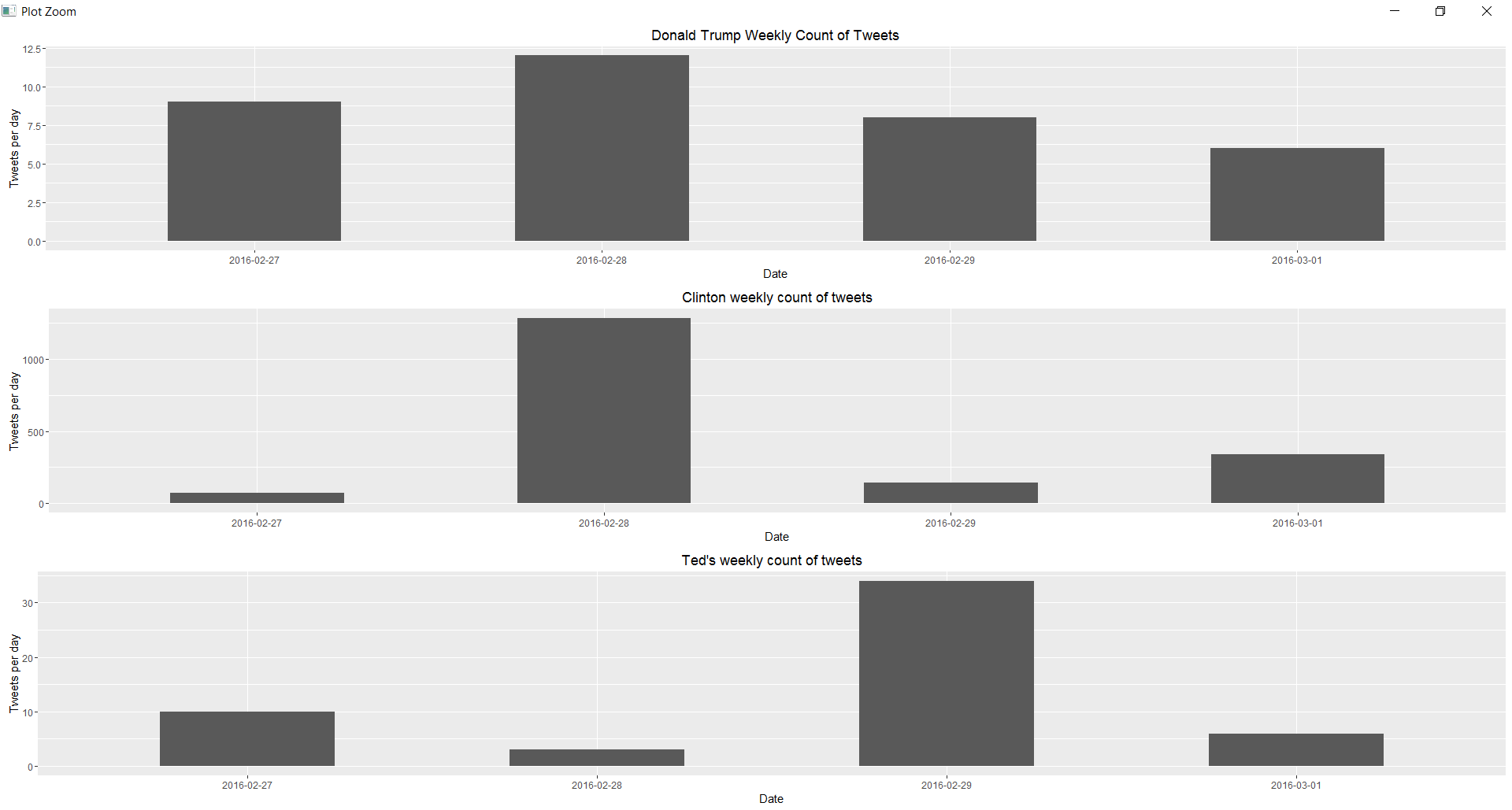
We can use this data not only on Election data but on any trending topic which can be classified into categories for sentiment analysis.

The below screen shot shows a sample of tweets returned for just over a span of 20 seconds for the hashtag “#Election2016”



As we can see, that is a huge number of responses for us to analyze, we can improvise this application to run the streaming without time out and update the stats without any delay.

Weekly Data Analysis:

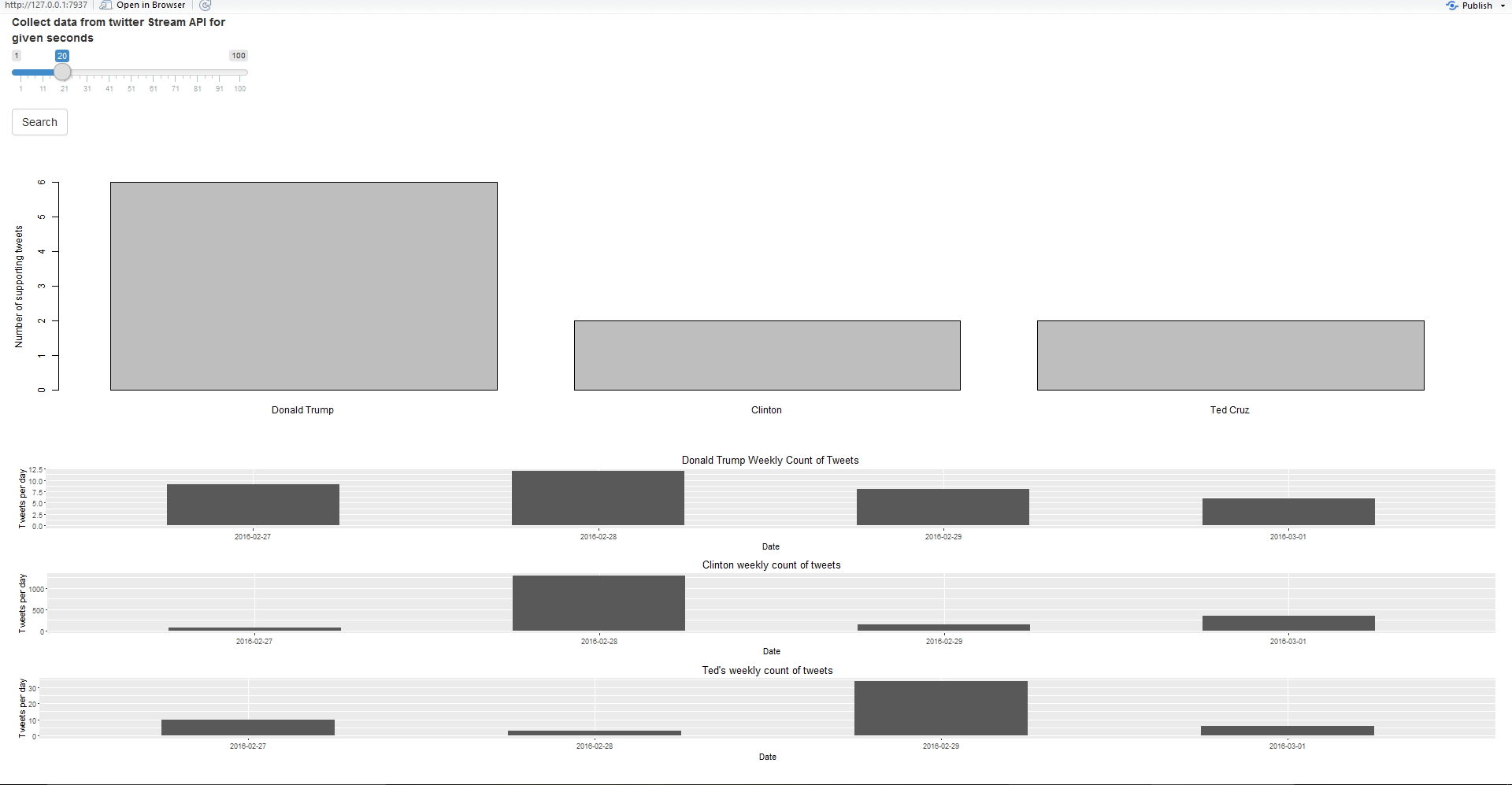
Below the Streaming analysis we can see the weekly tweets count for Elections summarized.

Conclusion:

This is a simple approach to do statistics on Streaming data and existing data accumulated earlier.

We can compare the current trend of events to past data and the results will be interesting.

Given time we can apply advanced text mining concepts to find out the patterns in various tweets that we collect.   
  
Complete view:



References:

My submission with for this project includes references from below links:

1. http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html

2. http://stackoverflow.com/questions/11370323/learning-to-understand-plyr-ddply

3. http://ww2.coastal.edu/kingw/statistics/R-tutorials/dataframes.html

4. R-Cookbook

5. http://stackoverflow.com/questions/19655431/reading-multiple-csv-files-in-r