CSE 587 – Data Intensive Computing

Problem 5 – Stream Processing

Author: Bhavika Jain
bhavikap@buffalo.edu

Objective: To process and analyze streaming data and create a UI dashboard in RShiny showing live statistical summary of data.

Approach:

As Presidential Election (2016) in United States is amongst the most trending topic on Twitter, I proceeded with doing live streaming of tweets on 'Elections' in USA. Using RShiny, I created a dashboard wherein, I am showing a summary of number of tweets posted on Twitter for each of the Republican and Democratic presidential candidates: Donald Trump, Hillary Clinton, Ted Cruz, Bernie Sanders, Ben Carson and John Kasich on a daily basis. Based on the stream of tweets processed live, I am also showing a summary of total number of tweets posted on Twitter for Republican and Democratic Party on a daily basis.

For the weekly summary of trends on election, I collected tweets for a week except today's date and displaying it on the dashboard alongside the daily summary. The weekly statistics shows the number of tweets posted for each party and presidential candidate in the last 6 days.

StreamR package in Twitter:

In order to do live streaming of tweets from Twitter, I used a streamR API package in R to get live tweets on 'Election'. The filterStream function in streamR opens a connection to Twitter's Streaming API and returns public statuses that match one or more filter predicates. By processing the tweets returned, I am doing a summary of total number of tweets posted for each of the presidential candidates and parties by a particular time on a daily basis.

RShiny:

First Step, I installed the '**shiny'** package in RStudio to start using RShiny. In order to create a RShiny app, I created two files, **'server.R'** and **'ui.R'**. In the ui.R file, I created the layout of my UI dashboard that will be displayed on the web portal. The server.R file contains the logic of processing live tweets and calculating the number of tweets posted for each party and presidential candidates daily. Further, in order to build a reactive output, I used an **'observer'** inside shinyServer. The streaming and processing of tweets happens inside the observer. Using a time interval of 30 seconds, I am re-executing the functionality inside observer of collecting and processing tweets and refreshing the UI dashboard with most recent summary stats on Election. The time interval of 30 seconds is created using **'invalidateLater'**.

For creating the UI, I used the **'shiny dashboard'**, which provides a way of dividing the screen into multiple panels for simple display of results.

WordCloud:

On the dashboard I am also displaying a word cloud for each of the presidential candidates. For this, I installed the package '**WordCloud'** in RStudio.

In the word cloud, I am showing all the words which have minimum frequency of 50 in the tweets processed for each presidential candidate.

Author: Bhavika Jain
bhavikap@buffalo.edu

The WordCloud is also refreshed as live tweets are processed.

Statistical Analysis:

Based on the stream of tweets received, I am parsing each tweets text using **'grepl'** to find if it contains the name of any our presidential candidate. If it contains the name of a particular candidate than that candidate's count of tweets is increased to 1. For example, if the tweet text contains the word 'Trump', then the count of tweets posted for Trump on twitter is increased to 1 (this count is added to prev count we got before 30 seconds). Similar summary of counts is obtained for other candidates.

For showing statistical summary of counts for party, I am adding the counts of tweets of each of the republican candidate to get the total count for Republican Party. For Democratic Party I am adding the counts of each of the democratic presidential candidate to get a total count of tweets.

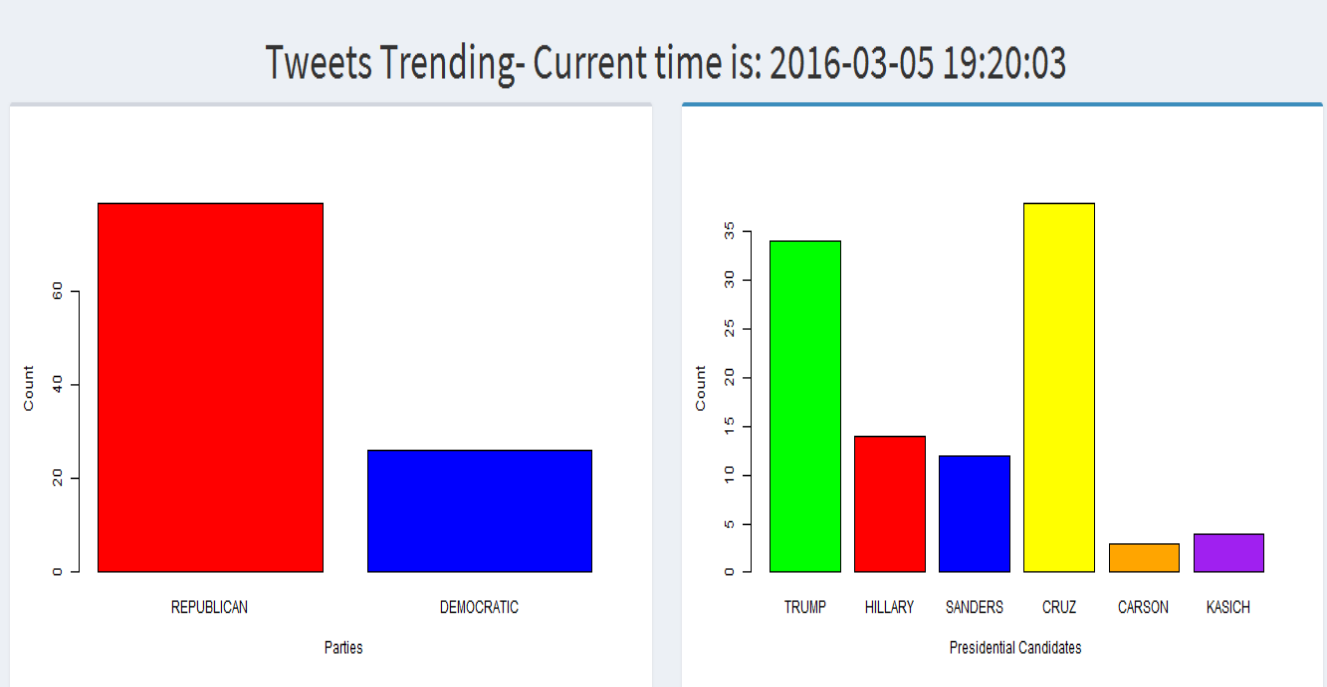
Below plots shows how daily and weekly statistical summary of Election tweets are displayed on the Shiny Dashboard:

The UI dashboard contains two sections :

- One panel shows the summary of total number of tweets posted by 'current time' for each of the parties. This indicates which party is currently more trending on twitter.
- Another panel shows the summary of total number of tweets posted for each presidential candidate by 'current time'.
  From this we can infer which party and candidate is currently more trending on twitter.

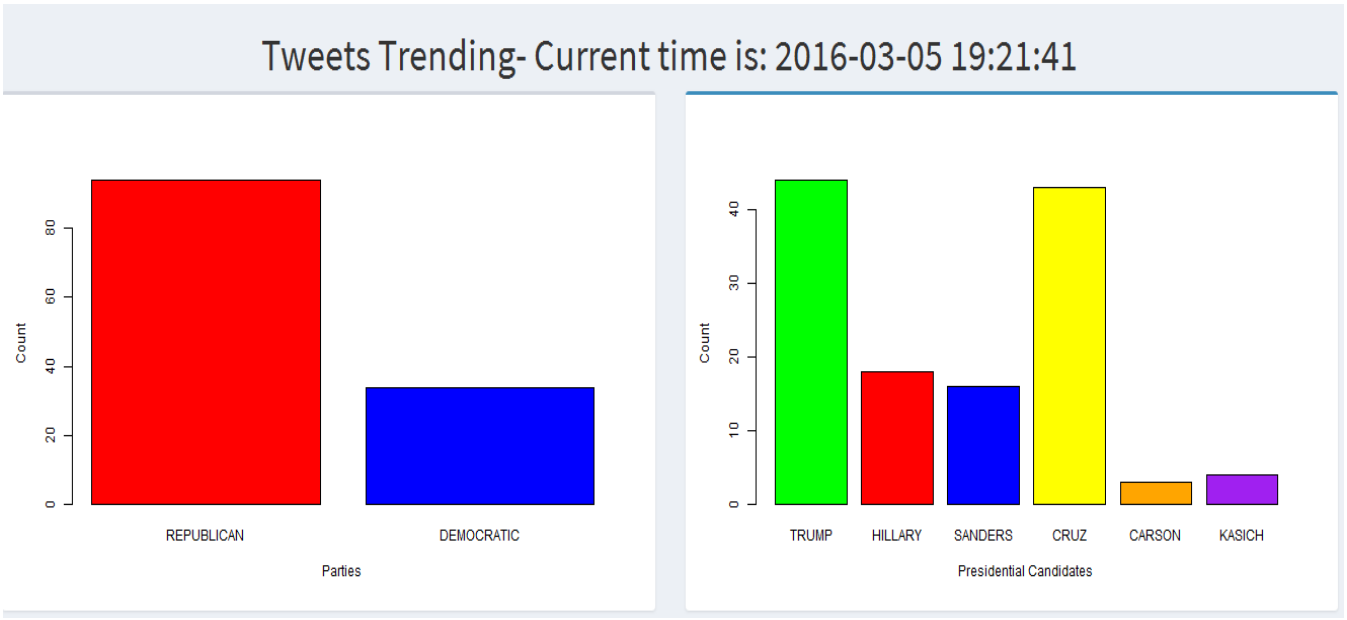
P.S : 'Current Time' means the system time which is also displayed on the dashboard as shown below.

Author: Bhavika Jain
bhavikap@buffalo.edu



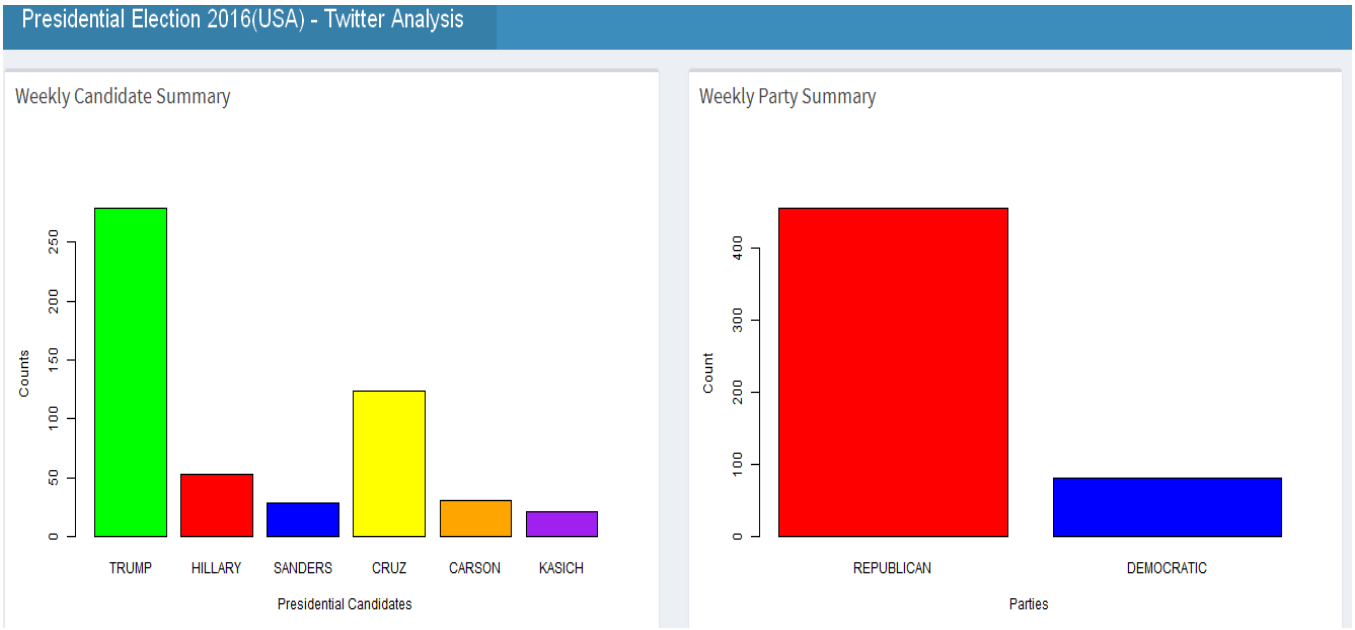
As seen in the next plot below, the count of tweets got updated on the dashboard as tweets are processed live. For example, the number of tweets posted for Donald Trump increased to more than 40 in next 30 seconds.

Author: Bhavika Jain
bhavikap@buffalo.edu



Weekly Summary:

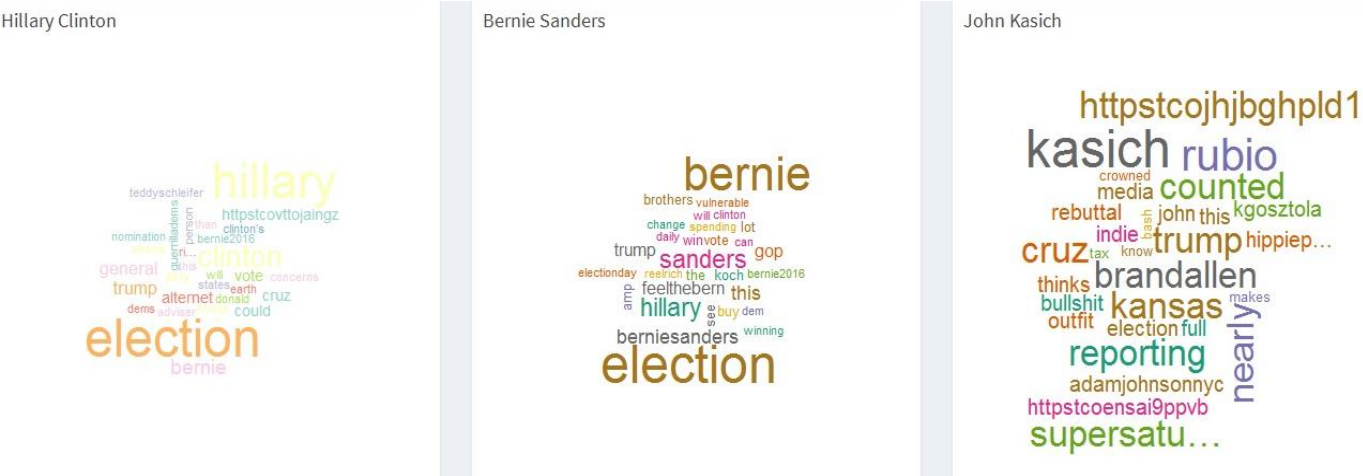
Below plot shows the weekly summary of tweets displayed on the dashboard.



Author: Bhavika Jain
bhavikap@buffalo.edu

WordCloud:

Below figures shows the display of word Cloud on dashboard for each candidate.



Author: Bhavika Jain
bhavikap@buffalo.edu

Conclusion:

Twitter and RShiny helps us in building applications that can process and analyze streaming data, the results of which can be further used for drawing inferences in various domains for example, in this project we did it for elections.

In this project, based on the daily results we can infer which party and presidential candidate is currently more trending on Twitter.

Key Learnings:

By working on this project, I gained in knowledge in below mentioned areas:

- Building a RShiny App.
- Collecting and processing live twitter tweets.
- Statistical Analysis of Tweets.
- Refreshing UI dashboard using RShiny reactive display.
- WordCloud.

P.S: I have attached the json tweets I collected for a week in the project's zip folder. Please set the directory in the script where this file will be stored before running the application, as the script needs to read this json file for showing weekly summary.

Author: Bhavika Jain
bhavikap@buffalo.edu