**Introduction to Sentiment Analysis:**

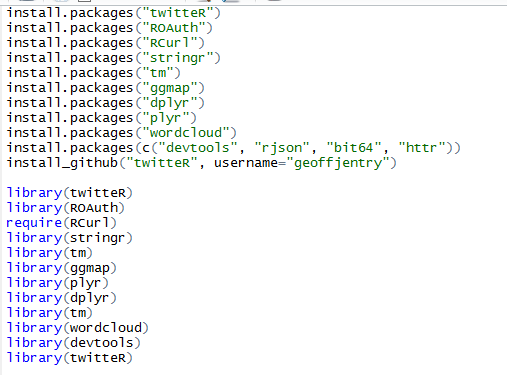
* Sentiment analysis (also known as opinion mining) refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials. Sentiment analysis is widely applied to reviews and social media for a variety of applications, ranging from marketing to customer service.
* Existing approaches to sentiment analysis can be grouped into three main categories: knowledge-based techniques, statistical methods, and hybrid approaches.

**Sentiment Analysis on PokemonGo using R**

Recently, the pokemonGo has become very popular among young people. Some of the many uses of social media analytics is sentiment analysis where we evaluate whether posts on a specific issue are positive or negative.

**Here are the steps to perform Sentiment Analysis using R:**

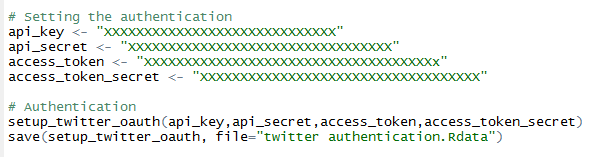
1. Install and Load the required packages



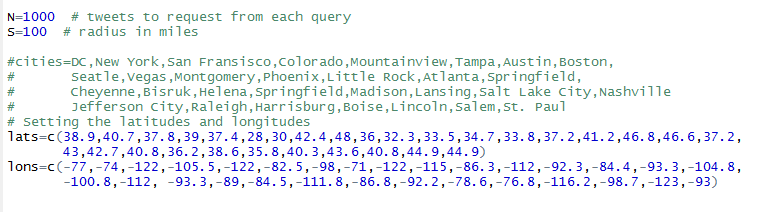
1. Create an account in Twitter and create an app. Get the api key, api secret, access token and access token secret from the app created in twitter.

**Refer:** [**http://iag.me/socialmedia/how-to-create-a-twitter-app-in-8-easy-steps/**](http://iag.me/socialmedia/how-to-create-a-twitter-app-in-8-easy-steps/)

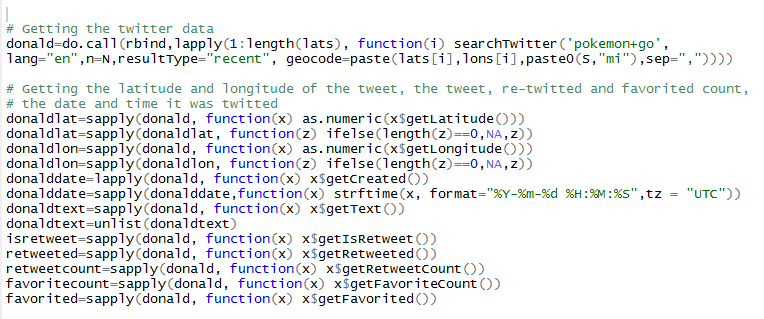
1. Set up the twitter authentication



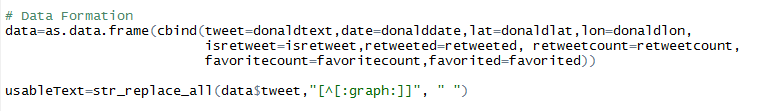
1. Set the number of tweets and the latitudes and longitudes of the cities from which the tweets are fetched



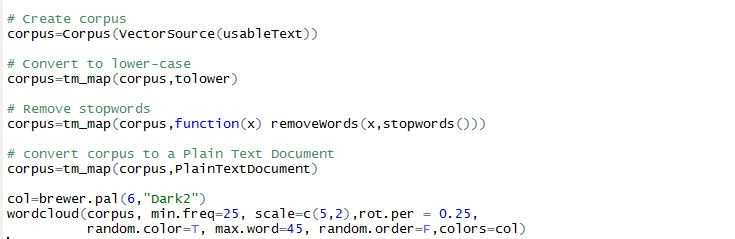
1. Get the tweets text and associated information such as retweeted count, favorite count, latitude and longitude etc.



1. Form the data and removing the graphical characters from the data created



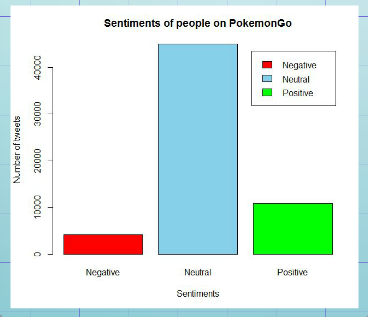
1. Creating the corpus and the word cloud





1. Assigning few positive words, negative words and checking if the tweet contains these words and plotting the bar chart accordingly.





**From this plot, we could see that Sentiments of people on PokemonGo is Neutral.**

**Sentiment Analysis on Dallas Shootout using R and Shiny**

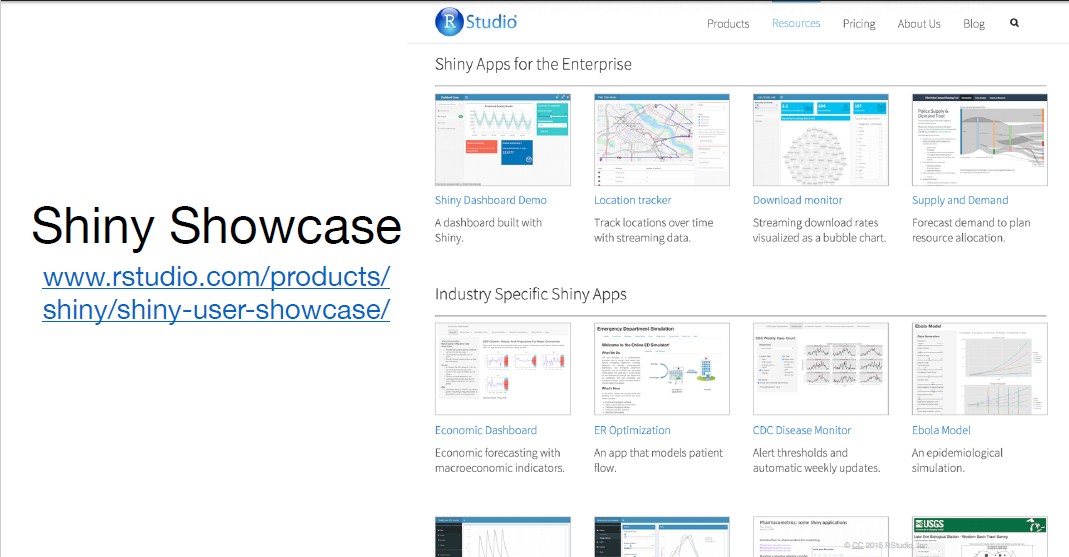
**Purpose of Shiny server:**

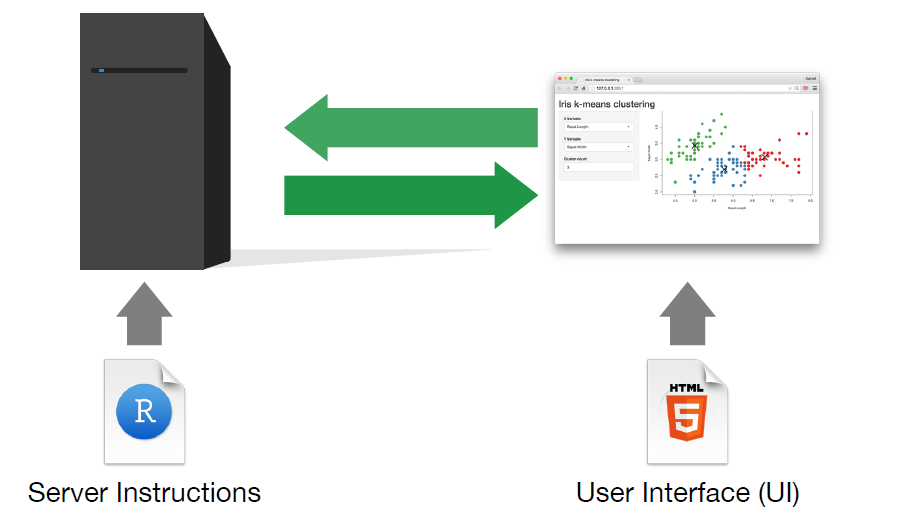
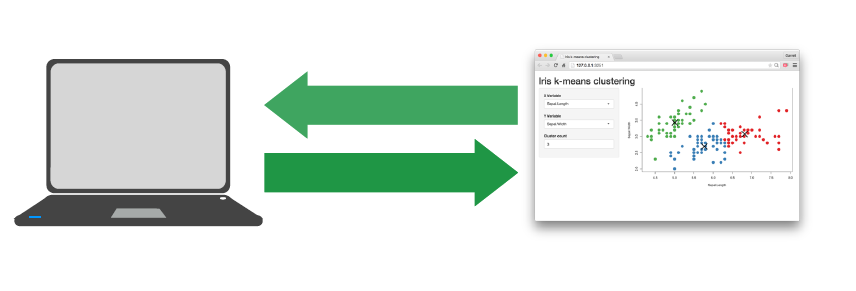
Shiny server lets you put shiny web applications and interactive documents online. Take your shiny apps and share them with your organization or the world.

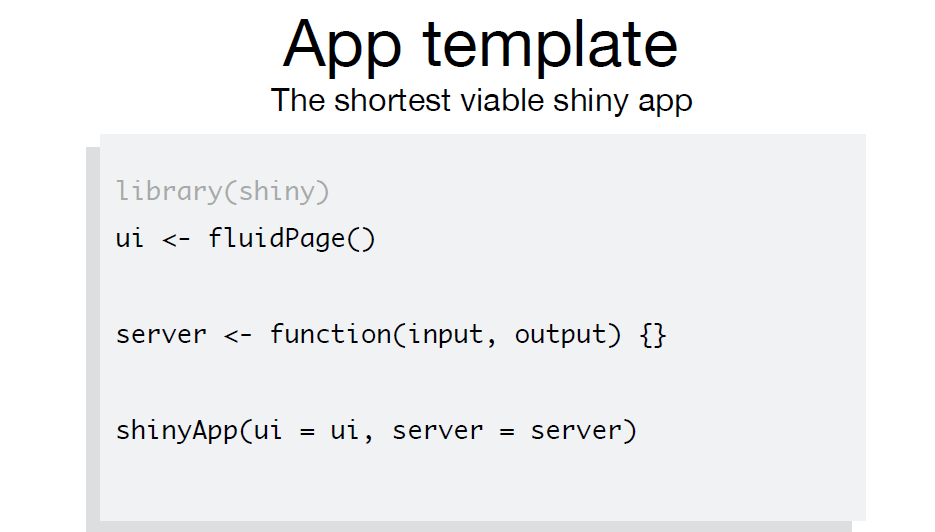
Shiny server lets you go beyond static charts, and lets you manipulate the data. Users can sort, filter, or change assumptions in real-time. Shiny server empowers your users to customize your analysis for their specific needs and extract more insight from the data.

Shiny Server Pro adds enterprise grade scaling, security, and admin features to the basic open source edition.

**Understanding Shiny App:**



Every Shiny app is maintained by a computer running R



**Steps for running the Demo:**

* **Step1:** Add elements to your app as arguments to fluidPage()
* **Step2:** Create an input with the input function
* **Step3:** Build your application with output functions
* **Step4:** Output() adds a space in the ui for an R object. You must build the object in the server function
* **Step5:** Run the application after installing the following packages

1. library(tm)
2. library(shinyapp)
3. library(wordcloud)
4. library(twitteR)
5. library(methods)
6. library(ROAuth)
7. require(RCurl)
8. library(stringr)
9. library(ggmap)
10. library(plyr)
11. library(dplyr)
12. library(RColorBrewer)
13. library(rsconnect)
14. library(devtools)Key, consumerSecret, accessToken, accessTokenSecret)

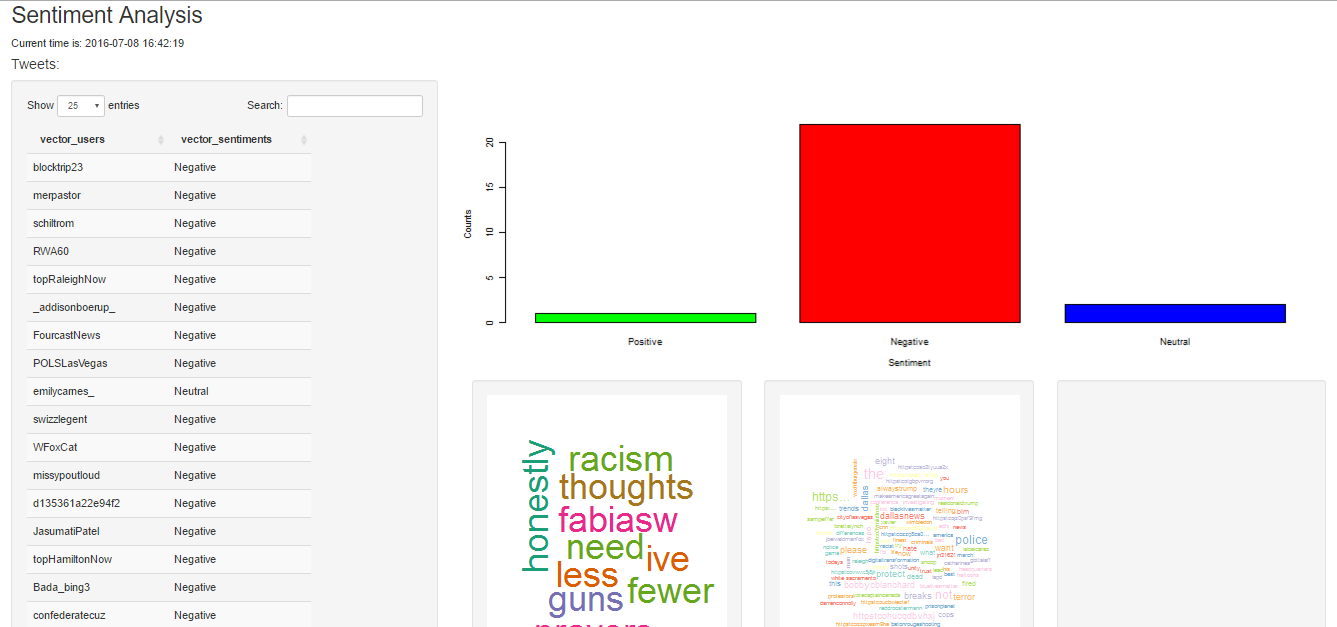
* **Step6:** Publish your application to the shiny server and access it with the URL below

**raghuram02.shinyapps.io/Team5\_TwitterSentimentAnalysis/**

**Code wise Workflow of the application:**

1. User inputs topics + number of tweets to be displayed
2. Authenticate Twitter Api with credentials (consumer key, consumer secret, access token, access token secret)
3. Fetch tweets using topic and number of tweets
4. Read data from sample lexical words(positive, negative, neutral)
5. Find positive and negative matches
6. Render histogram plots for positive neutral and negative tweets
7. Create corpus, plot word cloud for frequently occurring words

**Sample Output:**

****

**Challenges Faced**

* Since large number of tweets are fetched from the server, the R studio was unable to with stand the load and the session will timeout at times.
* Although, our shiny dashboard was successfully published in the shiny server, the dashboard didnt appear in the browser as the server ran out of memory. We can extend the memory size only if we have a premium account.