**Understanding Donald Trump’s Twitter Supporters –Twitter mining with Sentiment Analysis and Geolocations**

A Project submitted in fulfillment of the requirement for the award of

**BACHELOR OF TECHNOLOGY**

IN

**COMPUTER SCIENCE ENGINEERING**

BY

**Y.Tarini 1210312566**

**Mintu Mathew 1210312542**

**D. Saurabh Rao 1210312512**

**Ch. Anudeep 1210312506**

**Under the esteemed guidance of**

**Dr.Y.Radhika**

**Associate Professor**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GITAM INSTITUTE OF TECHNOLOGY,GITAM UNIVERSITY**

**VISAKHAPATNAM-045,2012-2016**

**GITAM INSTITUTE OF TECHNOLOGY**

**GITAM UNIVERSITY**

**VISAKHAPATNAM – 530045.**

**(2012-2016)**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**



**CERTIFICATE**

We hereby declare that the work which is being presented in the Bachelor of Technology Major Project Report entitled “Donald Trump’s Twitter Supporters –Twitter mining with Sentiment Analysis and Geolocations ”, on partial fulfillment of the requirement for the award of Bachelors of Technology in Computer Science Engineering and submitted to the Department of Computer Science and Engineering, GITAM Institute of Technology, GITAM University, Visakhapatnam is an authentic record of our own work carried out under the supervision of Dr. Y. Radhika, Professor.

Y.Tarini 1210312566

Mintu Mathew 1210312542

D. Saurabh Rao 1210312512

Ch. Anudeep 1210312506

Certified that the above statement made by the student is correct to best of our knowledge and belief.

Dr.P.V.Nageswara Rao,

Professor and Head of Department of CSE

GIT, GITAM UNIVERSITY

Visakhapatnam.

**DECLARATION**

I hereby declare that the project work entitled “Donald Trump’s Twitter Supporters –Twitter mining with Sentiment Analysis and Geolocations ” is an authentic record of my own work carried out as a project submitted in fulfillment of the requirement for the award of **BACHELOR OF TECHNOLOGY** IN **COMPUTER SCIENCE ENGINEERING.** This project is submitted to the Department of COMPUTER SCIENCE AND ENGINEERING in GITAM INSTITUTE OF TECHNOLOGY, VISAKHAPATNAM.

Y.Tarini 1210312566

Mintu Mathew 1210312542

D. Saurabh Rao 1210312512

Ch. Anudeep 1210312506

**ACKNOWLEDGEMENT**

I would like to thank all those who have contributed to the completion of this project report and helped me with valuable suggestions for improvement. I am extremely grateful to Prof. Dr. P.V. Nageswara Rao, HOD, GIT, Department of Computer Science Engineering for giving me this opportunity and allowing me to undergo my final year project. I would like to thank Dr. Y. Radhika and Dr. Murali Krishna for their help, encouragement and support throughout the project and providing us with required resources and information.

We could achieve this project with the continuous encouragement and support given by the officials who assigned the project and the management and staff of GITAM University,Vizag. I consider it as my privilege to express my gratitude and respect to all officials who guided, inspired and helped to complete the project work in time.

Y.Tarini 1210312566

Mintu Mathew 1210312542

D. Saurabh Rao 1210312512

Ch. Anudeep 1210312506

**Table of contents**

Title page 1

Certificate 2

Declaration 4

Acknowledgement 5

Table of Contents 6

List of figures 9

1. Abstract 10
2. Introduction 11

2.1 Accessing tweets from Twitter 11  
2.2 Annotations Extraction 13  
2.3 Geo-Coding of a location 14  
2.4Sentiment analysis 15

3. Literature Review 16

3.1 Text mining 17

3.2 Sentiment analysis 18

3.3 Geolocations 19

3.4 Cleaning data before analysis 20

3.5 Twitter API 20

3.6 R 22

4. Project Requirement Specifications 23

4.1 Problem Definition and Project scope 23

4.2Hardware and Software Requirements 23

4.3Reportable contents 24

5. Design 25

5.1 UML Diagrams 25

5.2 System Design 32

5.3 Procedure and packages used for data gathering 33

5.4Procedure and packages used for sentiment 37

5.5Procedure and packages used for geolocation 39

5.6Procedure and packages used for general graphical analysis 40

6. Results and discussion 41

6.1 Sentiment analysis 41

6.2 Geolocation analysis 42

6.3 Date and time of tweet vs favourite count 42

6.4 Follower count vs favorite count 43

6.5Follower count vs retweet count 43

6.6 Friend count vs favourite count 44

6.7 Friend count vs retweet count 44

6.8 Retweet count vs favorite count 45

6.9 Number of statuses vs number of favorites 45

6.10Number of statuses vs number of retweets 45

7.Graphs and installation views 46

Fig.1 Number of statuses vs number of retweets 46

Fig.2 Geolocation analysis 47

Fig.3 Date and time of tweet vs favourite count 47

Fig.4 Follower count vs favorite count 48

Fig.5Follower count vs retweet count 48

Fig.6 Friend count vs favourite count 49

Fig.7 Friend count vs retweet count 49

Fig.8 Retweet count vs favorite count 50

Fig.9 Number of statuses vs number of favorites 50

Fig.10 Sentiment analysis – data dump 51

Fig.11 Sentiment analysis – tweets with sentiment score 52

Fig 12. Sentiment analysis – collated sentiment values 53

Fig 13. Sentiment analysis – sentiment plot 54

Fig 14. Overview of a data frame 55

Fig 15. Overview of RStudio 56

8. Code 56

9. Conclusion 65

10. Future work 66

11. References 66

12. Appendices69

**List of figures**

Fig.1 Number of statuses vs number of retweets

Fig.2 Geolocation analysis

Fig.3 Date and time of tweet vs favourite count

Fig.4 Follower count vs favorite count

Fig.5Follower count vs retweet count

Fig.6 Friend count vs favourite count

Fig.7 Friend count vs retweet count

Fig.8 Retweet count vs favorite count

Fig.9 Number of statuses vs number of favorites

Fig.10 Sentiment analysis – data dump

Fig.11 Sentiment analysis – tweets with sentiment score

Fig 12. Sentiment analysis – collated sentiment values

Fig 13. Sentiment analysis – sentiment plot

Fig 14. Overview of a data frame

Fig 15. Overview of RStudio

**Abstract**Microblogging today has become a very popular tool for communication and voicing opinions among Internet users, where millions of users share opinions on varied topics. Microblogging platforms like Twitter allow users to share views within a 140 character limit, leading to a high rate of information compression. Users also have the option of adding images, links and videos among other things, which makes the type and variety of content very diverse. Here processing the tweet involves extraction of metadata of tweet, geocoding the physical address in a tweet, analysing the sentiment of content in the tweet text and extracting the significant and key phrases from a text. Here, we mainly focuses on performing three tasks. First is to collect the tweets from Twitter, having a chosen keyword, save them on a database and clean them up to have only necessary information. This is achieved using the Twitter Application Program Interface (API) along with R language. Secondly, to assign every tweet a score using Sentiment Analysis, which determines the judgment or evaluation of a user with respect to the chosen topic. This is performed using R language. Third is to represent the locations of the tweets geographically. The details of the users of the corresponding tweets are collected using Streaming API provided by Twitter and the geographic representation is achieved using the packages of R language. In this paper , we specifically focus on one area to analyze – Donald Trump. We search Twitter using keywords related to him or his campaign to better understand and visualize about the kind of topics the users are tweeting about, how the overall sentiment is regarding his campaign and from which areas are his tweets coming from.

**Introduction**Present data conveys reporting through automation and therefore minimizes human push to investigate the content. Continuous examination encourages easy to use strategies in executing frameworks, to concentrate data from the textural content. In the present connection of breaking down the content and then finding useful patterns, a fixed structure goes a long was in easily understanding the content. Representation needs to be handled sensibly or factual information that has to be analyzed can be lost in the process. Our current work is centered to dissect the writings of huge volume sources like Twitter (a platform where users continuously keep updating servers with content every second ). Because the velocity of the data is extremely high , it therefore becomes very important to have a database in place that can continuously process data and make it available for use by developers and analysts. Variables which are huge in number in this format offer many advantages like dependability, ease of use and adaptability. The current work proposition requires a legitimate study on different angles that impact proposed execution.

**Accessing tweets from Twitter**

The Twitter API has several methods, such as GET statuses / user\_timeline, GET statuses / home\_timeline and GET search / tweets, which return a timeline of Tweet data. Such timelines can grow very large, so there are limits to how much of a timeline a client application may fetch in a single request. Applications must therefore iterate through timeline results in order to build a more complete list. Because of Twitter’s realtime nature and the volume of data which is constantly being added to timelines, standard paging approaches are not always effective. The goal of this analysis is to demonstrate the issues Twitter developers may face when paging through result sets and to give best practices for processing a timeline.  
Twitter API [[1]](#footnote-2)offers reaction to asks for in "JSON", "XML" and "Molecule" arrangements, parsing the yield need particular to the system you are utilizing to extricate. The http reaction codes may be seen in the yield, by determining the status of the client demand.   
Twitter4j [[2]](#footnote-3) gives a usage of libraries to parse the GET reactions like JSON, XML and so forth. Metadata of the tweet likewise embedded accordingly of a pursuit question, it's essential in comprehension the data expressed in the tweet[[3]](#footnote-4). have prove and broke down that each tweet is not geo-labeled (geographic coordinate's latitude and longitude), however a few tweets are solely geo-labeled in reactions through Search API. It's simply discretionary to the client in expressing the geo-area, on account of client point of view and security to disallow the geo-labeling highlighting while tweeting.

**Annotation Extraction**

The component of indentifying the annotation depends on the coordinating of the prepared document content with textural expressions of separate annotation kind of comparing documents. Preparing records with information on annotation and inner system workings are two of the techniques which are utilized to ensure that the data is easily available in a format that can be understood by parsers and mining algorithms. There is each need to go through the code for customization, separated from the models that are mentioned in the documentation to make sure that the content can be used in a manner that is suitable to our analysis. In the event that custom execution requests more annotation separated from models, there are elective alternatives to go for custom models. Keeping this in mind, we have not directly imported JSON data from twitter, but have ensured that the data is obtained through various packages in R, therefore reducing our workload substantially. One component that impacts the execution is the preparation source, where we have to be careful about the measure of the accuracy to which the preformatted data can perform. It is important to understand that most current data mining algorithms do not focus a lot on speed, because the prevailing assumption is that there is existing hardware which can handle the amount and different types of data that is being sent from Twitter. The size and frequency of the query that is being made to the server is also under question, which is why companies like Twitter have introduced rate limiting for each kind of request, thereby ensuring that the developer is not able to make too many requests within a specific time frame.

**Geo Coding of a location**Geo-coding assumes an imperative part in representation of physical location on visual maps. Earth surface is isolated in even and vertical points, the level lines speak to scope and vertical lines speak to longitude. For scope the equator is taken a reference point as 0 Degree and towards posts end 90 Degrees, the Greenwich (prime meridian) and aggregate 360 Degrees compass of vertically into equivalent parts of 180 Degrees of east furthermore, 180 Degrees of west. Geo-coding directions are decimal estimations of scope and longitude. As the goal of this work it requests for geo-coding (changing over area or address into scope and longitude organizes) the contemporary instrument is to make utilization of the API's having usefulness information relating to the geographic organizational structure. In this connection, it is important to investigate the accessible assets, assess the relative usefulness, ease of use and adaptability in customization of the asset. Most of the data needed for geo coding is being gathered through twitter itself, and occurs when the user has given permission to access their location through the web or mobile application. Due to this reason, the number of tweets that are seen with a geotag on them are substantially lower. In our analysis, we also aim to understand the percentage of users that are tweeting with their location. As the administration of the location data is in the control of the user, they put limitation over the availability by restricting developer demands. We could reverse search the locations based on the information that the user has provided, but the amount of permissions needed for that would increase substantially and would also mean using Google Maps API, among others. Additionally, the number of users who have an accurate description mentioned are still very small, therefore making the entire process futile.

**Sentiment analysis**

Sentiment analysis is important in today's world to dissect the corpus or mass writings. It is clear the time requirement, high recurrence of information and reports, quick client criticisms forcing additional weight on adjusting bodies (blogging gatherings, market investigators, stock sheets etc). Aside from the supervision it needs a computerized instrument to assess the feeling in the content. There is extent of study by utilizing opinion investigation instrument as a part of progressing theory in open life, client sentiment investigation, following the audits of an item and to examine the mass slant over distinctive issues or viewpoints. Presently, it has been organized in exploration and improvement of specific apparatuses to accomplish a superior examination over mass information in developing economies. [[4]](#footnote-5)A content or archive can be broke down and bifurcated into positive and negatives sentiment apart from neutral statements. Keeping in mind the end goal to assess the information corpus, we have undertaken an approach where every tweet is given a unique rating, and the overall sentiment can be predicted by understanding the sentiment of users whose tweets are positive, negative or neutral. In this regard, what we have done is collect a corpus of words that are both positive and negative. If the words in the tweet match any word in the word corpus, it is assigned a score based on how severe the reaction is. By collating the total sum of all the positive and negative responses apart from the neutral ones, we are accurately able to estimate the fraction of users who are happy, unhappy or have no opinion regarding a topic. The prescient score of the assumption in the content gets unearthed by this method. We can further aggregate these results by collating them over time.

**Literature review**

The growth of the computer age has brought along with it a very large amount of information that is available for use. This has gained a lot of traction especially in the past few years, with scientists and the general public alike looking for better ways to utilize and make sense of the existing data. The existing patterns of communicating data, which mostly include listing out the data might not always be the best method to ensure that the information behind the data is conveyed efficiently. One of the better methods used in this regard is the process of visualization.

Visualization is using the human comprehension in preparing data proficiently and successfully. The quickened development of 'informal communities' (Example,Twitter) makes conceivable, to exchange and share data to numerous client's quick with less cost. The potential result of interpersonal interaction encourages a client to reach and co-operate a great many different clients. Organizations are building third party gathering applications, which are exploratory in conveying apparatuses to advantage client. It examines the conclusions, client sees, new thoughts, open hobbies, and their engaged exercises of a huge number of client round the globe. Showcasing firms additionally get included in dissecting client inputs and practicing over open supposition, and the brake out of most recent patterns in the masses in overhauling the items and administrations. The crude material in building the outsider application is mass volumes of information that needs to procedure to get data. The extraction of data from crude information put additional weight on applications that impedes viable usage of accessible information. Content examination might likewise allude as content digging for content investigation, to enhance quality perseverance and adds sense to the importance of information. The work proposed visualizes a structure that not only reduces the amount of existing noise that usually comes along with the corpus, but also looks at methods that can ensure that the amount of noise be reduced to a great extent.   
  
We expect to assess the accessible API's to get access information from Twitter, and execution of suitable methodology to fabricate database of interpersonal organization information (twitter). To make it helpful for perception of twitter information, it is imperative to comprehend which is proficient and compelling in usage and support. Assumption examination have knowledge to distinguish the positive and negative sense in the content, the assessment concentrates for the most part on the conduct viewpoints and words or expressions that implies the human feelings.

**Text mining**

Traditional data mining assumes that the information to be “mined” is already in the form of a database which can be accessed easily. Unfortunately, for many applications, electronic information is only available in the form of free natural-language documents rather than structured databases. In the case of twitter, the data is semi structured, meaning that the data has certain labels attached to it, and further processing needs to be done to ensure that the data is accessible in an easier manner for evaluation. There is a small, but growing body of research in specifically opinion mining from microblogging data. Kim[[5]](#footnote-6) et al. give a compelling case for using Twitter lists for a corpus in sentiment analysis2. In this context, lists are groups of people who share a common interest such as music. They show that even though tweets are brief, they contain enough information to express identifiable characteristics, interests and sentiments. The seminal work by Pang et al[[6]](#footnote-7). shows that machine learning is a viable tool for sentiment analysis using movie reviews for a corpus3.

**Sentiment analysis**

Sentiment analysis is a useful predictor to understand how popular the sentiment of a particular keyword is. This is majorly used by brands and companies that are looking to understand how the consumer reacts to their brand or products. This can have additional applications as well, for instance, to understand how well a political heavyweight is performing in the polls. It is with this intention that we have used sentiment analysis to understand how well the voters are reacting to a candidate. The sentiment found within comments, feedback or critiques provide useful indicators for many different purposes. These sentiments can be categorised either into two categories: positive and negative; or into an n-point scale, e.g., very good, good, satisfactory, bad, very bad. In this respect, a sentiment analysis task can be interpreted as a classification task where each category represents a sentiment. Sentiment analysis provides companies with a means to estimate the extent of product acceptance and to determine strategies to improve product quality. It also facilitates policy makers or politicians to analyse public sentiments with respect to policies, public services or political issues. **[[7]](#footnote-8)**

**Geolocations**

The growing volume of user-generated text posted to social media services such as Twitter, Facebook, and Tumblr can be leveraged for many purposes ranging from natural disaster response to targeted advertising[[8]](#footnote-9). In many circumstances it is important to know a user’s location in order to accomplish these tasks effectively. Despite the significant interest in geoinference and reported success of methods, examining current evaluation practices reveals significant disparity between evaluation settings. Moreover, the conditions in which methods have been tested often do not mirror the real-world conditions in which these methods are expected to operate. In particular, data set sizes have varied by four orders of magnitude and no agreement has been seen in (1) the source of ground truth data, (2) what percentage of the dataset should already be labeled with ground truth locations when training, and (3) which evaluation metrics should be used. While these factors are not necessarily limitations of the algorithms, they critically hinder comparability and obtaining a complete understanding of how state of the art performs at its intended task.[[9]](#footnote-10)

**Cleaning data before analysis**

When it comes to tweets, it is very important to clean the data to ensure that the data that is being used does not have any chance to bring up a complication. Usually, string manipulation functions are used to remove unnecessary parts of a tweet ( for instance, punctuation marks, Usernames etc). After the unnecessary data is removed, the tweet can further be processed. We also have to remove stop words that can cause an output that is not satisfactory. Lastly, we also have to remove emoticons and other special characters to have purely text based tweets. Lastly, the entire tweet is also converted to lowercase to ensure that it can easily be matched to other corresponding words.

**Twitter APIs**

Twitter uses OAuth to provide authorized access to its API. There are two forms of authentication in the new model, both still leveraging OAuth 1.0A. Application-user authentication is the most common form of resource authentication in Twitter’s OAuth 1.0A implementation to date. Your signed request both identifies your application’s identity in addition to the identity accompanying granted permissions of the end-user you’re making API calls on behalf of, represented by the user’s access token. Application-only authentication is a form of authentication where your application makes API requests on its own behalf, without a user context. API calls are still rate limited per API method, but the pool each method draws from belongs to your entire application at large, rather than from a per-user limit[[10]](#footnote-11)

The REST APIs provide programmatic access to read and write Twitter data. Author a new Tweet, read author profile and follower data, and more. The REST API identifies Twitter applications and users using OAuth; responses are available in JSON.[[11]](#footnote-12)

The Streaming APIs give developers low latency access to Twitter’s global stream of Tweet data. A proper implementation of a streaming client will be pushed messages indicating Tweets and other events have occurred, without any of the overhead associated with polling a REST endpoint.[[12]](#footnote-13)

**R**

R began as an experiment in trying to use the methods of Lisp implementors to build a small testbed which could be used to trial some ideas on how a statistical environment might be built. Early on, the decision was made to use an S-like syntax. Once that decision was made, the move toward being more and more like S has been irresistible. R has now outgrown its origins and its development is now a collaborative effort undertaken using the Internet to exchange ideas and distribute the results. The focus is now on how the initial experiment can be turned into a viable piece of free software. R is a computer language and run-time environment which can be used to carry out statistical (or other quantitative) computations. The base part of R comes with a wide range of standard statistical and graphical analyses built in. There are a large number of user-developed extension packages which provide an even richer set of capabilities. R is free software released under the Free Software Foundation’s General Public License. This means that R is free of any restrictions on how it can be disseminated. Versions of R can be obtained without charge and can be redistributed to others. The license prevents the creation of encumbered derived works (i.e. commercial versions).[[13]](#footnote-14)

**Project Requirement Specifications**

**Problem Definition and Project scope**

The problem statement is to address the lack of knowledge on part of the research community and the general public knowing how well Donald Trump is performing on Twitter to better analyze what could have been done to increase or decrease the influence that he seems to have on Twitter. This particular analysis is concerned with understanding how the average day is in the life of all people who tweet about Donald Trump. More specifically, the analysis is aiming to understand at what times are the tweets most frequent, what the users are tweeting about, when they are doing it etc. For this reason, a day in which no specific news regarding the candidate was released was considered. We also plan to then draw up a sentiment regarding the sentiment analysis of the tweets and then try to correlate it with the geolocations.

The work that has to be done to accomplish the research, i.e. , the scope of the project revolves around collecting the required data needed to perform basic visualization on the data. These visualizations will be spread over time thereby giving us an analysis of the growth or retardation of the quantity and nature of tweets. We will then proceed to perform sentiment analysis on the same data for different time periods. Lastly, we will try to correlate these with the geo coded data.

**Hardware and Software Requirements**

**Hardware**

64 bit operating system ( Windows preferred, but Mac, Linux and Fedora also acceptable)

Minimum of 512 MB RAM ( Recommended : 2 GB )

Minimum of 500 MB free space in the hard drive ( Recommended : 1 GB )

Minimum 512 KB graphic card (Recommended : 2 GB )

**Software**

R 3.2.4

RStudio 0.99.893

Microsoft Windows 8.1 or higher

Microsoft Office 2007 or higher

**Reportable contents**

\* Multiple graphs which compare two variables on the x and y axis as different attributes of the incoming tweets over time.

\* Sentiment analysis data dump

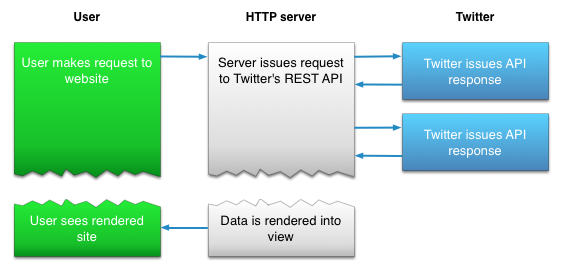
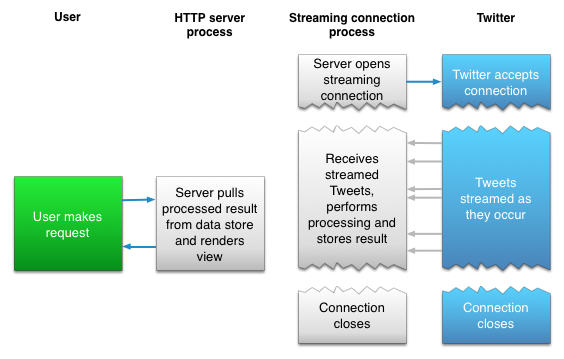
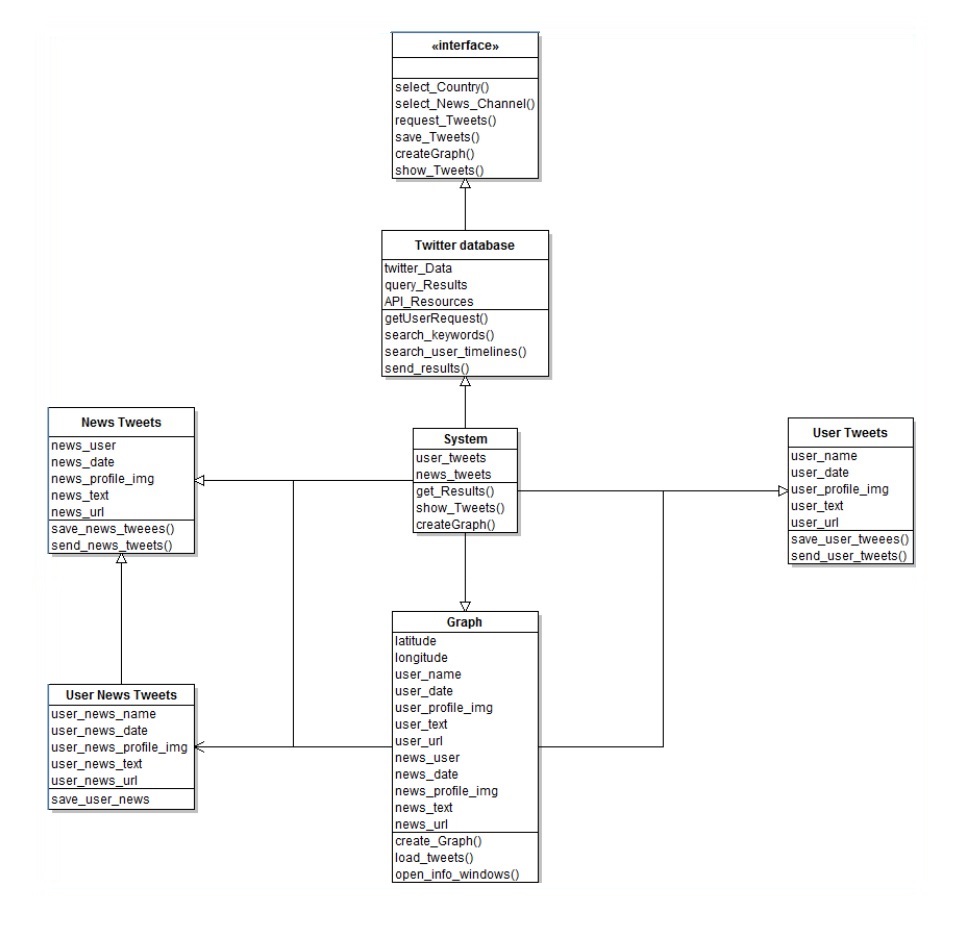
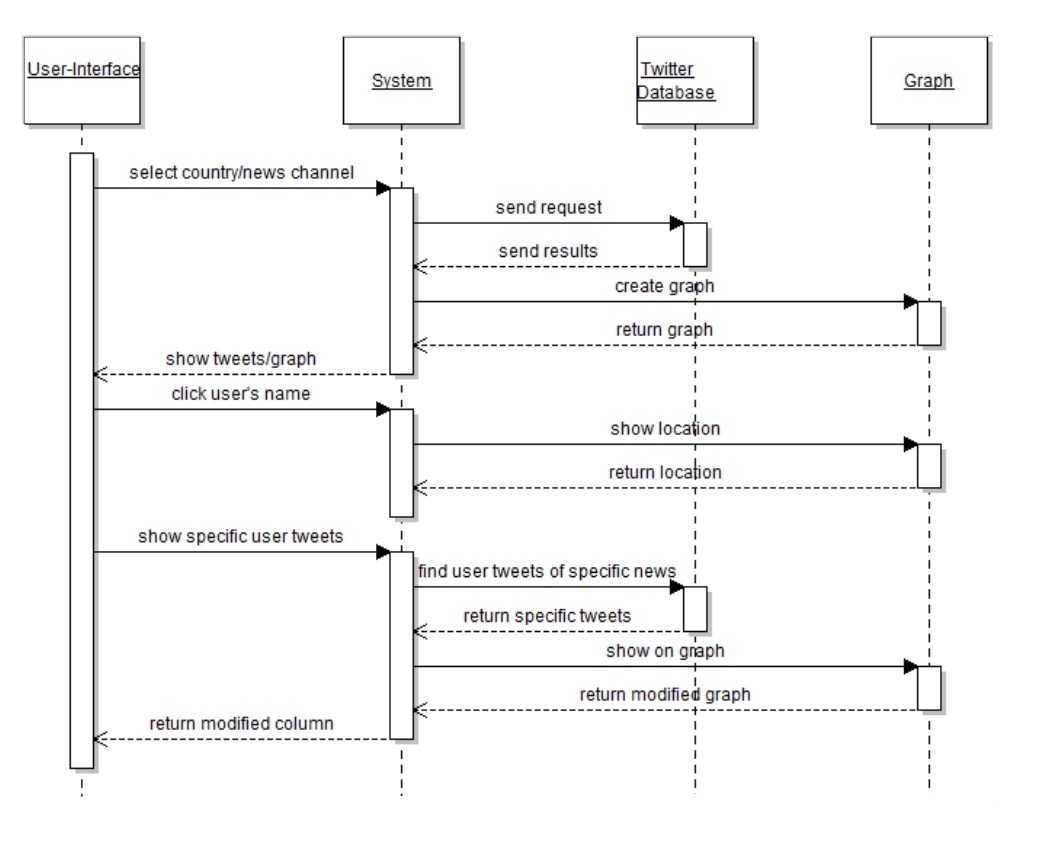
\* Sentiment analysis with assigned score for each tweet

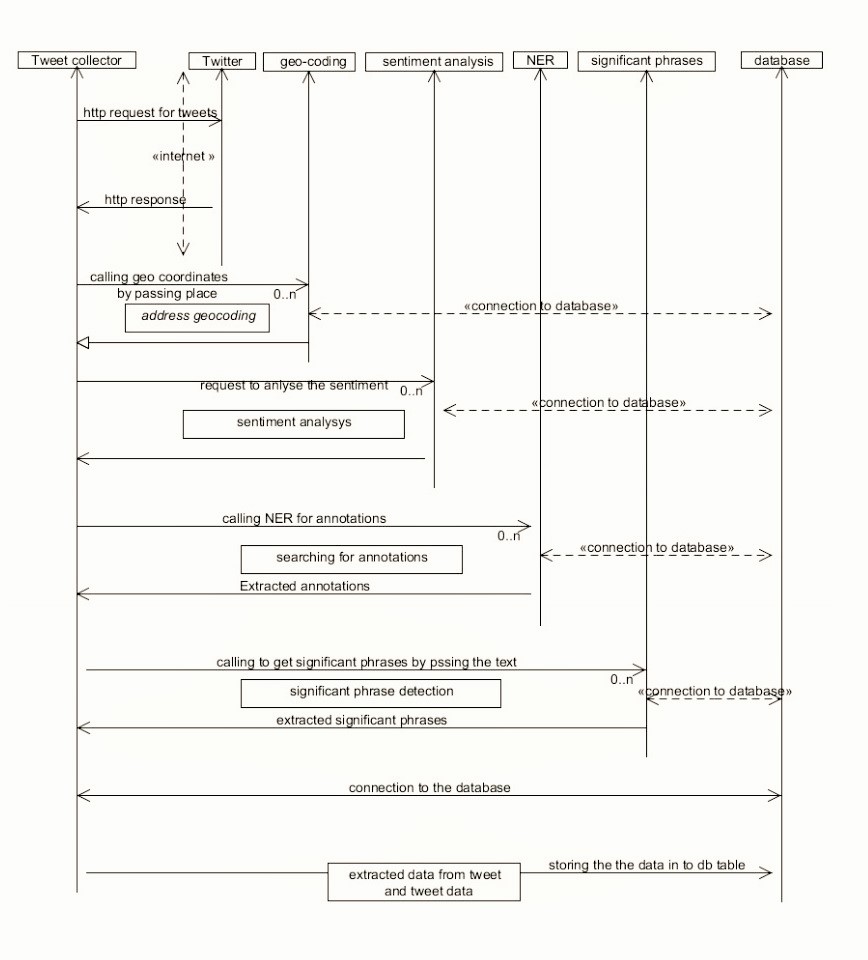
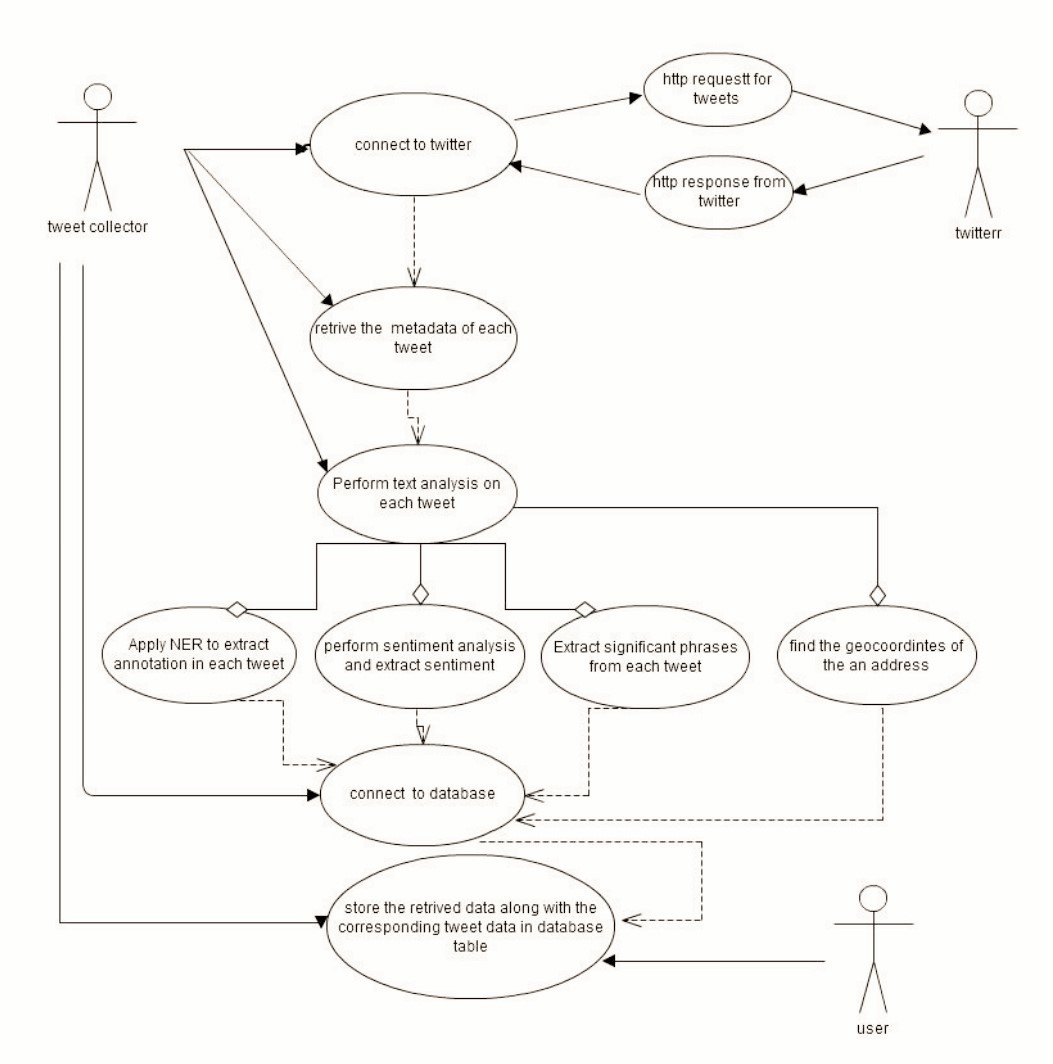
\* Sentiment analysis overall positive , negative and neutral tweets.

\* Sentiment analysis graphical representation of change in sentiment over time.

\* Geolocation based graphs over time.

**Design**

**UML Diagrams****Getting tweets from twitter procedure** **Procedure for maintaining a persistent HTTP connection** **Geocode class diagram** **Geocode sequence diagram** **Geocode Use case diagram**

Sentiment   
  
Senwkjfb ****  
Geo ewkjfb **Sentiment Analysis class diagram   
  
Sentiment Analysis sequence diagram   
  
Sentiment Analysis Use Case diagram**

**System model**

Twitter REST API

Twitter Streaming API

Connection type - HTTPS

Application code

Input text

Data Pre-processing

Input parameters for topical filtering using Python

Saving to database

Filtering unnecessary variables

Database with only relevant data

Data Mining

Database n

Database 3

Database 2

Database 1

….

Multiple copies of same data for different purposes

Location analysis

Sub issue classification using a Bayesian learning system

Sentiment Analysis(Opinion Mining)

Influencers

Data Post-Processing

Analysis and visualization overtime

Conclusion, Future scope

A Analysis and visualization nalysis and visualization overtime

Analysis and visualization overtime

Ana Analysis and visualization overtime

lysiefgf3gf3kjgn3gkj3bgkj3n3kjn3s a Analysis and visualization nd visualization overtime

**Procedure and packages used for gathering data**

* The consumer key , consumer secret key , access key and access secret key are initialized.
* OAuth environment is set up with above mentioned variables.
* Cacert.pem file is downloaded for enabling RCurl.
* Handshake is initiated with credentials
* .RData file is saved and loaded for specific instance.
* Tweets are saved in JSON format from Twitter.
* Tweets are parsed into a data frame.
* Since the tweets are ordered by date , it is easy to perform different kinds of analysis on them.

**twitteR**It is an R based Twitter Client that provides an interface to the Twitter web API.The current version used is 1.1.9 [[14]](#footnote-15).

**RCurl**

A wrapper for 'libcurl' <http://curl.haxx.se/libcurl/> Provides functions to allow one to compose general HTTP requests and provides convenient functions to fetch URIs, get & post forms, etc. and process the results returned by the Web server. This provides a great deal of control over the HTTP/FTP/... connection and the form of the request while providing a higher-level interface than is available just using R socket connections. Additionally, the underlying implementation is robust and extensive, supporting FTP/FTPS/TFTP (uploads and downloads), SSL/HTTPS, telnet, dict, ldap, and also supports cookies, redirects, authentication, etc.[[15]](#footnote-16)

**base64enc**

This package provides tools for handling base64 encoding. It is more flexible than the orphaned base64 package.[[16]](#footnote-17)

**devtools**

This is a collection of package development tools.[[17]](#footnote-18)

**tm**

It is a framework for text mining applications within R.[[18]](#footnote-19)

**streamR**

This package provides a series of functions that allow R users to access Twitter's filter, sample, and user streams, and to parse the output into data frames.[[19]](#footnote-20)

**RJSONIO**

This is a package that allows conversion to and from data in Javascript object notation (JSON) format. This allows R objects to be inserted into Javascript/ECMAScript/ActionScript code and allows R programmers to read and convert JSON content to R objects. This is an alternative to rjson package. Originally, that was too slow for converting large R objects to JSON and was not extensible. rjson's performance is now similar to this package, and perhaps slightly faster in some cases. This package uses methods and is readily extensible by defining methods for different classes, vectorized operations, and C code and callbacks to R functions for deserializing JSON objects to R. The two packages intentionally share the same basic interface. This package (RJSONIO) has many additional options to allow customizing the generation and processing of JSON content. This package uses libjson rather than implementing yet another JSON parser. The aim is to support other general projects by building on their work, providing feedback and benefit from their ongoing development[[20]](#footnote-21).

**stringr**

A consistent, simple and easy to use set of wrappers around the fantastic 'stringi' package. All function and argument names (and positions) are consistent, all functions deal with "NA"'s and zero length vectors in the same way, and the output from one function is easy to feed into the input of another.[[21]](#footnote-22)

**ROAuth**

Provides an interface to the OAuth 1.0 specification allowing users to authenticate via OAuth to the server of their choice[[22]](#footnote-23).

**Procedure and packages used for sentiment analysis**

* Cacert.pem file is downloaded to enable RCurl
* Access URL, Request URL and Auth URL are declared.
* Consumer key and Consumer Secret are declared.
* Credentials are saved and handshake is initiated.
* After handshake is complete, the .Rdata environment is loaded and setup.
* Function for searching tweets with the required keyword is started. The data dump is put into an excel file at this point.
* The score evaluation function is used at this point.
* In the score evaluation function, punctuation marks are removed after converting to UTF-8 format, sentences are split into words and the total words are made into a list for each tweet.
* Each word in a tweet is compared against the database of positive and negative words, and the score is computed.
* By this point , we have an excel file of each tweet and the score associated with it.
* We then generate another excel file which calculates the total number of positive, negative and neutral tweets for a day.
* We then plot these points in a graphical format and display the result in a .jpeg image.

**Plyr**

A set of tools that solves a common set of problems: you need to break a big problem down into manageable pieces, operate on each piece and then put all the pieces back together. For example, you might want to fit a model to each spatial location or time point in your study, summarise data by panels or collapse high-dimensional arrays to simpler summary statistics. The development of 'plyr' has been generously supported by 'Becton Dickinson'.[[23]](#footnote-24)

**Dplyr**

A fast, consistent tool for working with data frame like objects, both in memory and out of memory.[[24]](#footnote-25)

**stringr**

A consistent, simple and easy to use set of wrappers around the fantastic 'stringi' package. All function and argument names (and positions) are consistent, all functions deal with "NA"'s and zero length vectors in the same way, and the output from one function is easy to feed into the input of another.[[25]](#footnote-26)

**ggplot2**

An implementation of the grammar of graphics in R. It combines the advantages of both base and lattice graphics: conditioning and shared axes are handled automatically, and you can still build up a plot step by step from multiple data sources. It also implements a sophisticated multidimensional conditioning system and a consistent interface to map data to aesthetic attributes. See http://ggplot2.org for more information, documentation and examples.[[26]](#footnote-27)

**Procedure and packages used for geolocation analysis**

* Once all the tweets are organized by date, we first discard tweets which do not have latitude and longitude values for them.
* We then consider only the latitude and longitude values and plot them over time.

**googleVis**

R interface to Google Charts API, allowing users to create interactive charts based on data frames. Charts are displayed locally via the R HTTP help server. A modern browser with Internet connection is required and for some charts a Flash player. The data remains local and is not uploaded to Google.[[27]](#footnote-28)

**Rworldmap**

Enables mapping of country level and gridded user datasets[[28]](#footnote-29).

**rworldxtra**

High resolution vector country boundaries derived from Natural Earth data, can be plotted in rworldmap.[[29]](#footnote-30)

**Procedure and packages used for general graphical analysis**

* After the results have been classified and categorized using native R packages, we then graph them to understand what the data can convey.

**ggplot2**

An implementation of the grammar of graphics in R. It combines the advantages of both base and lattice graphics: conditioning and shared axes are handled automatically, and you can still build up a plot step by step from multiple data sources. It also implements a sophisticated multidimensional conditioning system and a consistent interface to map data to aesthetic attributes. See http://ggplot2.org for more information, documentation and examples.

**Results and Analysis**

**Sentiment analysis**

In the case of sentiment analysis, there are four deliverables that have to be made. The very first is the data dump, which can be used as a log of all the tweets along with their unique IDs that were collected. The second excel file was the analysis in which each tweet was given a sentiment score. By simply observing the values which we received, we were able to get scores with a maximum positive value of +6 and a maximum negative value of -3. Therefore, as the difference is close to 10 units, we can safely conclude that the degree of emotions that people have regarding Donald Trump is very widely varying. Overall, we found the sentiment to be positive, or neutral is some cases. The number of tweets that could be considered as negative were very small when compared to the total number of positive or neutral tweets. Over tweets mined over a seven day period, we found the overall sentiment to carry on in the following fashion : 12.5 to 15 % of all tweets were negative. Around 50 % of the tweets were neutral and 35 % of the tweets were negative. While exact values do vary, this variation has been more or less the observed pattern. This is the observation that can be made from the opinion numbers. Lastly, the overall sentiment has remained more or less the same, meaning that the support from Trump during the period that we have considered has neither grown nor reduced by a great extent.

**Geolocation analysis**

While it was possible to consider the importance of the geolocation data by representing the geolocated tweets, the number of tweets that had geolocation enabled when compared to the total number of tweets we were considering was very small. It was less than 1.5% of all the tweets. When less than 1.5 %of tweets are geolocation enabled, not a lot of significant results can be drawn from them with regard to sentiment, mostly because the opinion that will be generated through them will most likely be biased. In other words, it is not an accurate metric. Therefore, the only useful data that can be drawn out from the graph is with regard to the places from which people have tweeted about Donald Trump. The pattern that is obtained is predictable, and most of the users that tweet are fromn the East coast. The number of tweets with geolocation enabled from the west coast drop significantly. The Midwestern part of the USA and the central districts usually have no tweets coming from their area, save for a few. This is in line with the number of users on twitter that are from those states.

**Date and time of tweet vs favourite count**

This plot tries to compare the date and time of a tweet against the favourite count , which is essentially to measure when the most optimal time is that a person interested in Donald Trump will tweet. Interested parties(including his own campaign) can cash in on this fact by targeting voters who support his candidacy. The plot clearly finds that there is no clear correlation between the date and time of tweets and the favourite count. In other words, it has not been affected a lot by this metric. While there are a few exceptions to the rule, the rule still applies to the majority of cases. The important lesson that can be learnt from this analysis is that when looking to publish tweets that one thinks a large segment of the population will respond to, it is important to keep tweeting continuously. It does not help if one is tweeting sporadically.

**Follower count vs favorite count**

This plot draws a comparison between the number of followers and the favorite count. The plot clearly shows that the number of favourites do tend to be higher even when the follower count is low. This is important to understand because it provides us a crucial insight into where Donald Trump is getting most of his support from on Twitter. It is not from accounts that have a lot of followers or a huge influence on social media, but it is from accounts that have a very low follower count and belong to people who support the man vociferously. We believe that if the opposition parties are looking to convert Trump supporters into supporting another party or ideology, the best place to start looking would be accounts that do not have a very large follower count and support Trump.

**Follower count vs retweet count**

This plot draws a comparison between the number of followers and the retweet count. The retweet count also follows similar patterns as number of followers vs favorite count. The chief difference that can be observed here, however, is the fact that the tweets that have a high impact ( with regard to being retweeted the maximum number of times) have been few, not many. There are multiple clusters on the y axis, which indicate that the tweet that was being retweeted a lot did not necessarily have a lot of followers, but went on to be retweeted anyway. If we are looking to understand Trump supporters, the best bet would be to understand their psyche – they do not have access to a lot of information with regard to the number of followers they have or the number of people they are following, and whatever content they have access to is believed to be the best representation of their political ideology, which is what explains so few tweets being retweeted so many times.

**Friend count vs favourite count**

The friend vs favorite graph examines the relationship between the number of friends the user has to the number of times a tweet of that person has been favorited. The graph here is a little more distributive when compared to the previous graphs , and the overall trend shown there is negative with regard to number of friends being proportional to the number of favorites one receives. It just goes on to prove that Trump supporters will not favorite anything just because it has been said by a friend that they know. They can choose statements that are a good representation of what their political beliefs are.

**Friend count vs retweet count**

The friend vs favorite graph examines the relationship between the number of friends the user has to the number of times a tweet of that person has been retweeted. The graph here is slightly more balanced, meaning that there are people who have retweeted what their friends have to say, and there are people who have retweeted what random people had to say too. This goes on to prove that while favoriting a tweet might mean something a lot more to Trump followers ( considering that they do not favorite a lot of content written by their friends), an endorsement by a retweet shows that they still identify to be from the same thinking. IN other words, a follower is clearly able to identify another follower in a macroscopic context and ensure that their content is what they receive.

**Retweet count vs favorite count**

Here, we plot the retweet count of every status against the favorite count of every status. While tweets are retweeted a lot, the same cannot be said about favoriting a tweet. The graph is much more clustered when it comes to favorites, thus making it clear that favorites are a clear endorsement of support and therefore should carry more value when compared to a retweet. The principle here is that while a tweet can be retweeted by a larger percentage of the audience, when it is favorited, it means that people have been able to form a personal connection with what the other person has had to say. In other words, influencers should aim to write content that not only focuses on gathering retweets, but content to which viewers can connect to.

**Number of statuses vs number of favorites**

The number of statuses vs number of favorites graph is much less distributed than its counterparts and a lot of information can be extracted from it. The very first thing that can ibe understood is the fact that a large number of statuses does not necessarily mean that your tweets will have a high number of favorites. Secondly, it is important to understand here that favorites are a metric that are much more closer to people’s hearts and it therefore stands to reason that they would favorite tweets to which they personally identify with.

**Number of statuses vs number of retweets**

The number of statuses vs number of retweets graph is much more clustered than its counterparts and a lot of information can be extracted from it. The appearance of the graph is similar to that of retweet vs favorite plotting, and goes on to show similar trends. The graph is much more clustered when it comes to retweets, thus making it clear that retweets are a clear endorsement of support and they belong to people who have not necessarily been tweeting with the highest number of statuses. The principle here is that while a tweet can be retweeted by a larger percentage of the audience, retweets do not work in that manner. The extent of the tweet is very localized and therefore does not give it the platform to be exposed to a wider audience , unless the tweet is endorsed either as a retweet or a follow by a social influencer.

**Graphs and installation view**

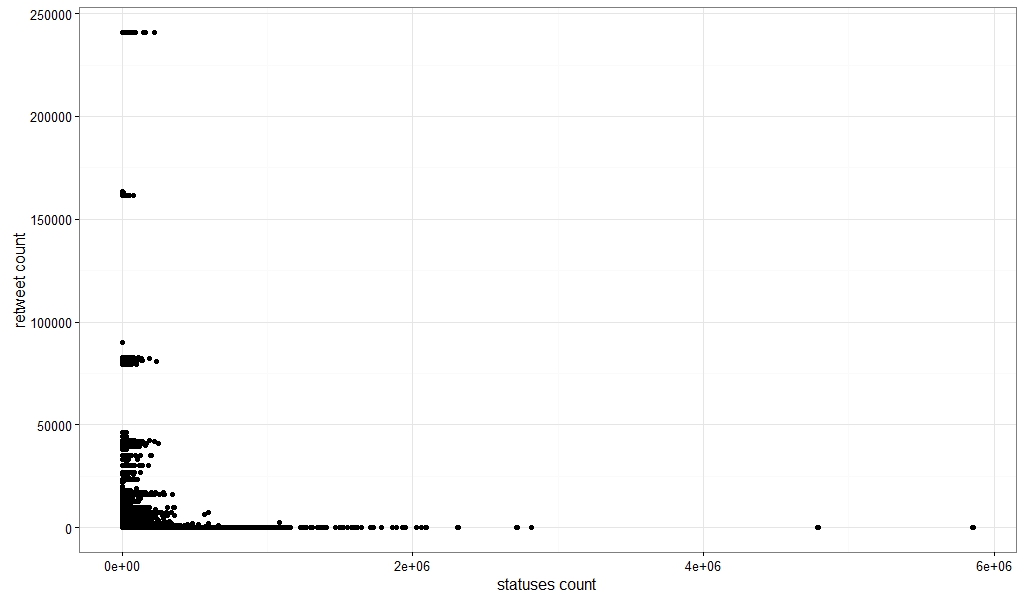


Fig.1 Number of statuses vs number of retweets

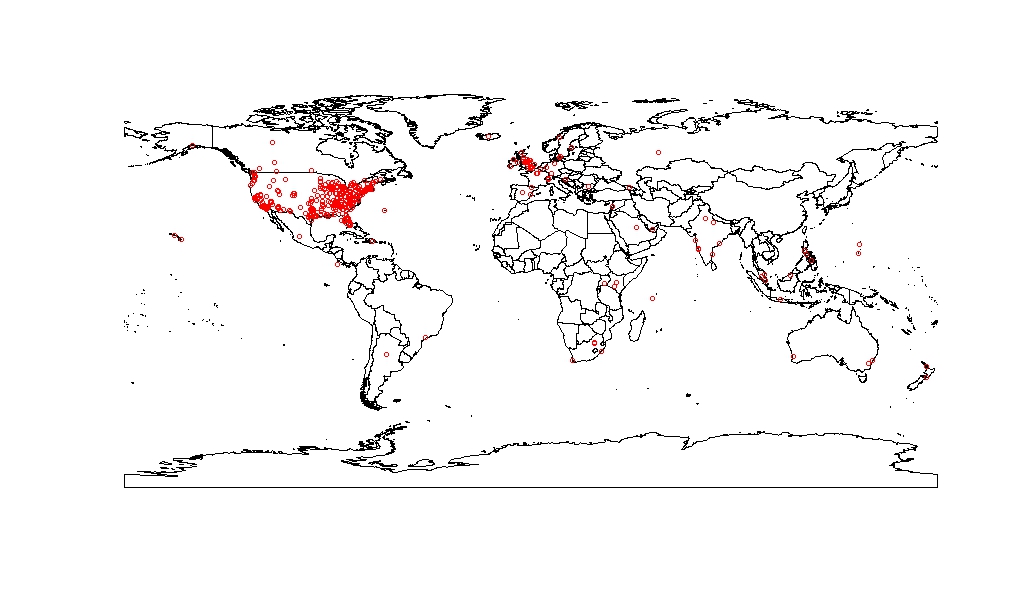


Fig.2 Geolocation analysis

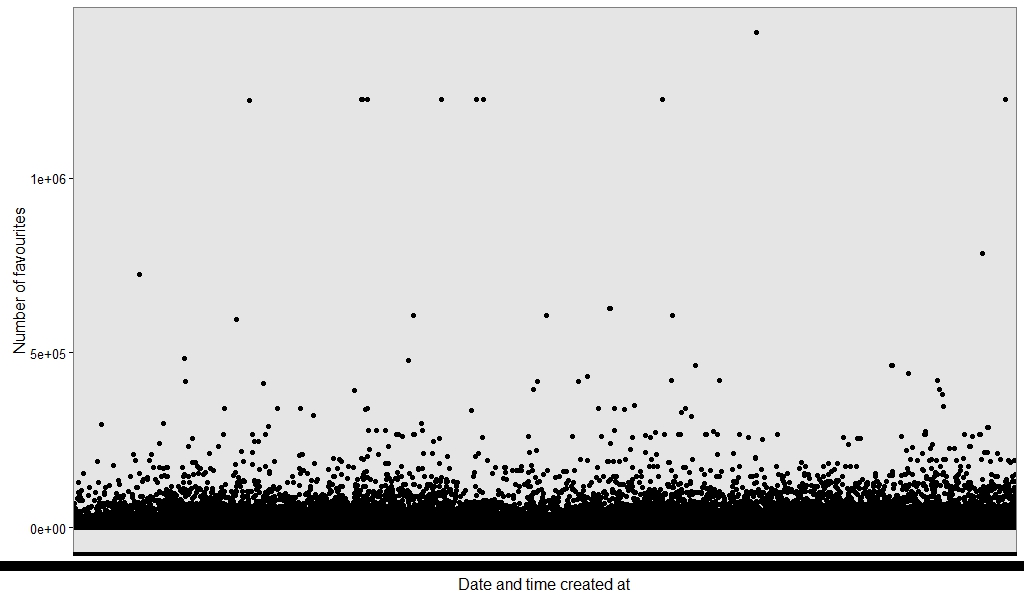


Fig.3 Date and time of tweet vs favourite count

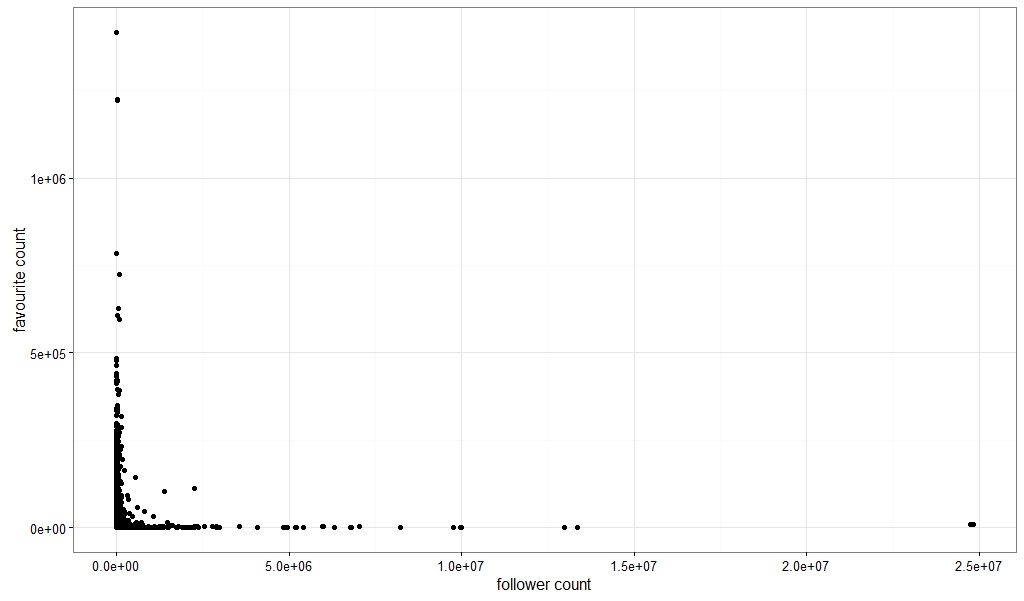


Fig.4 Follower count vs favorite count

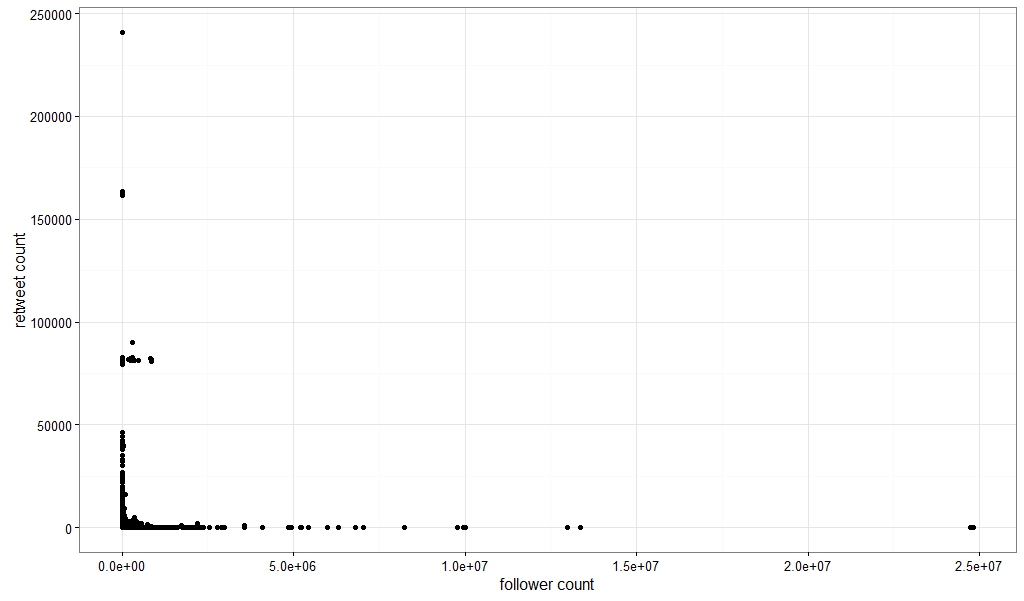


Fig.5Follower count vs retweet count

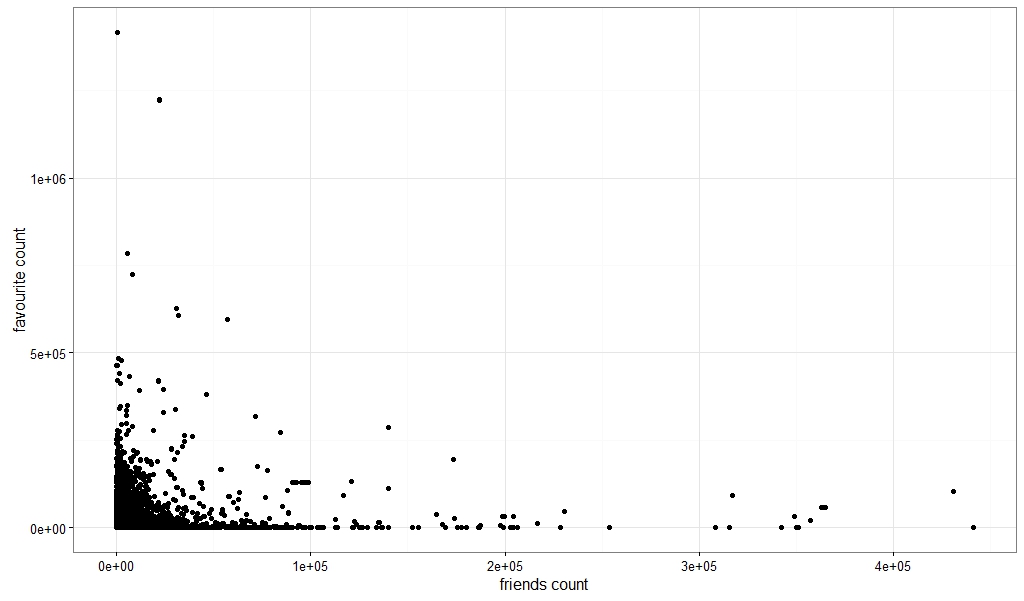


Fig.6 Friend count vs favourite count

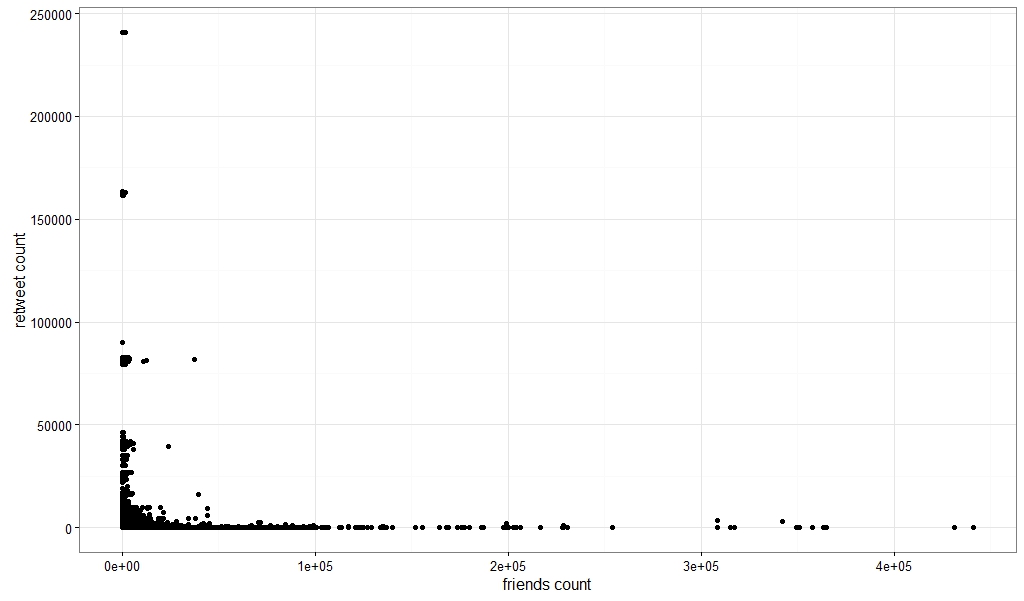


Fig.7 Friend count vs retweet count

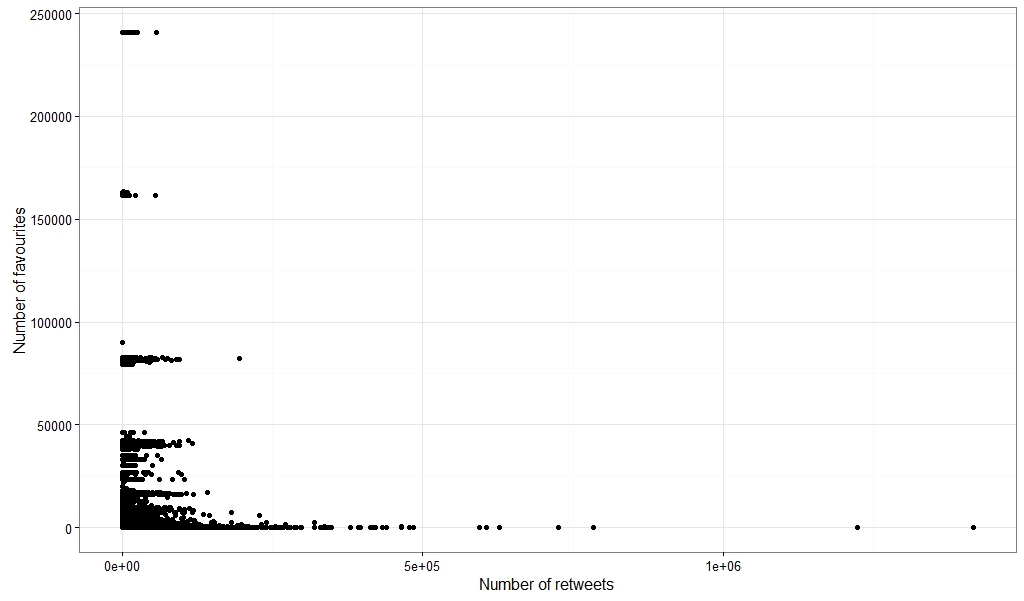


Fig.8 Retweet count vs favorite count

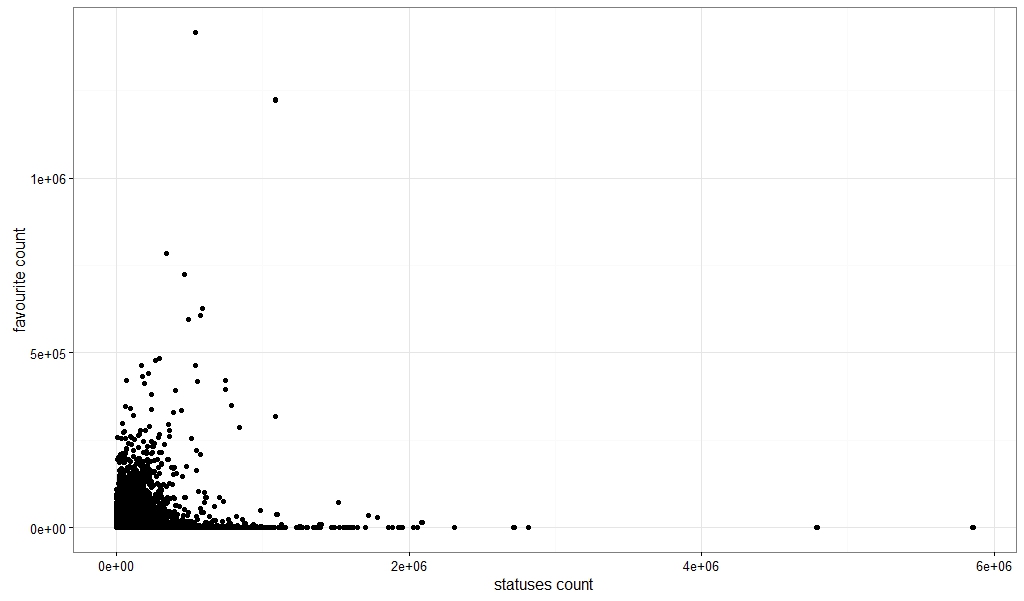


Fig.9 Number of statuses vs number of favorites

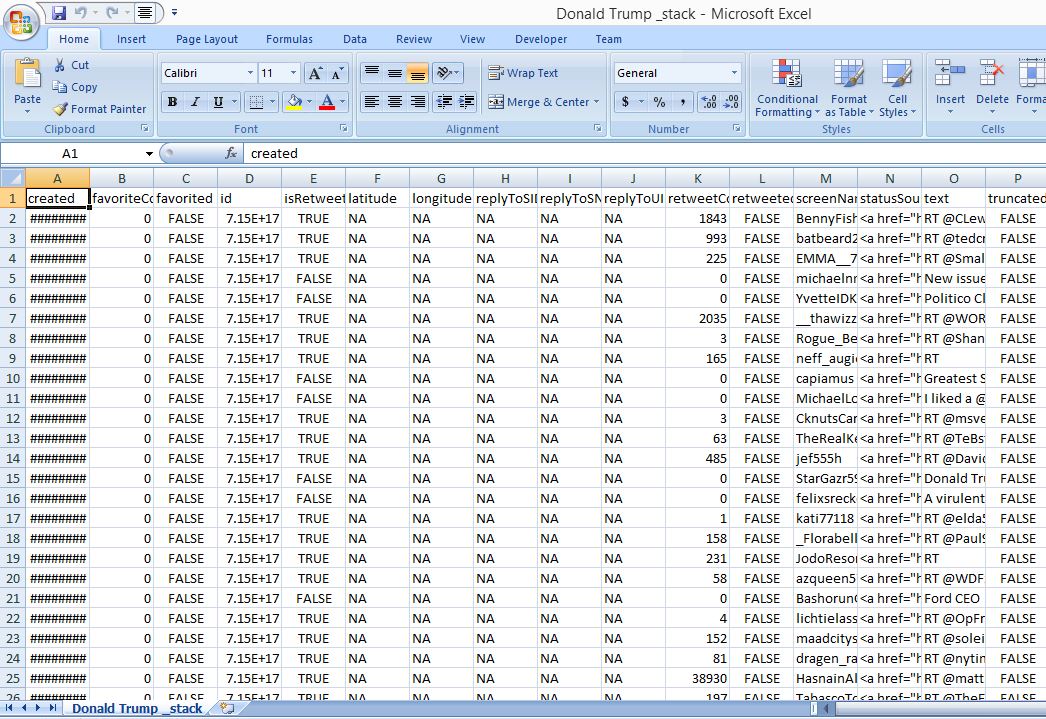


Fig.10 Sentiment analysis – data dump

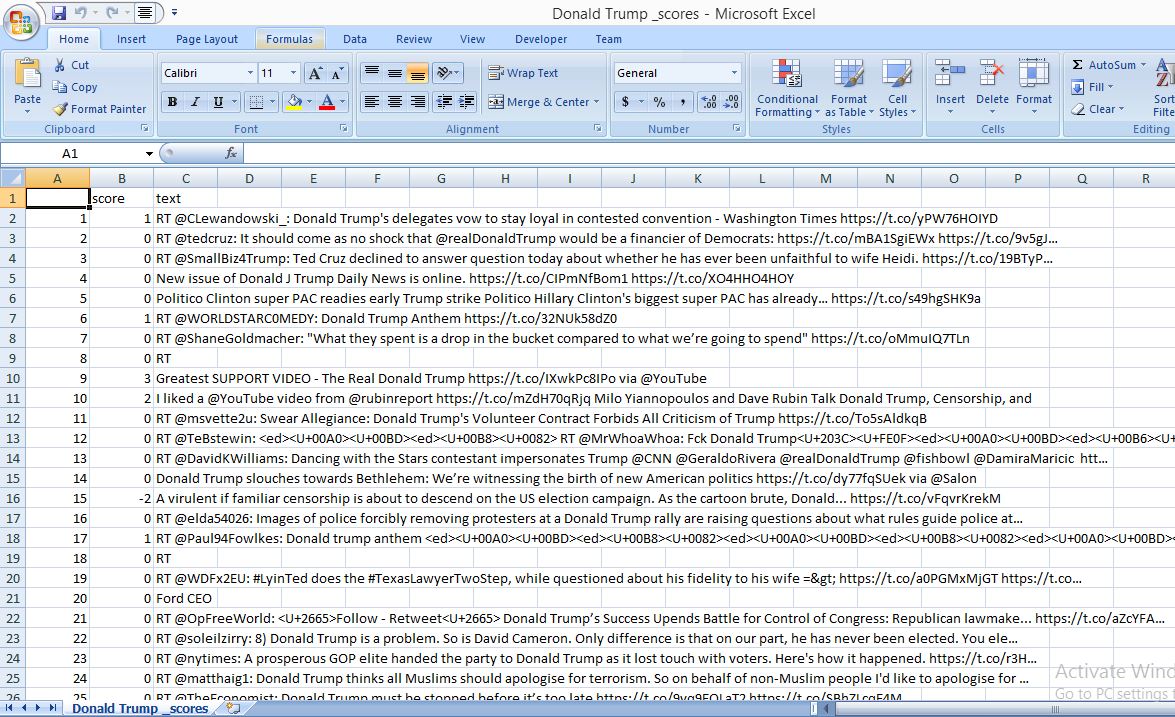


Fig.11 Sentiment analysis – tweets with sentiment score

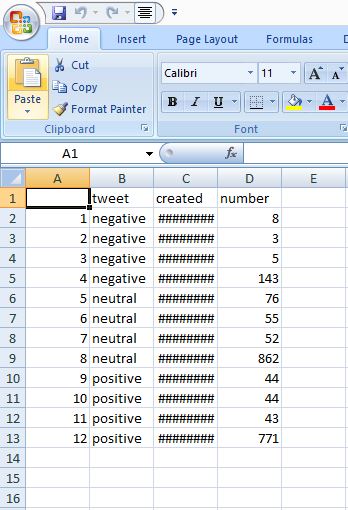


Fig 12. Sentiment analysis – collated sentiment values

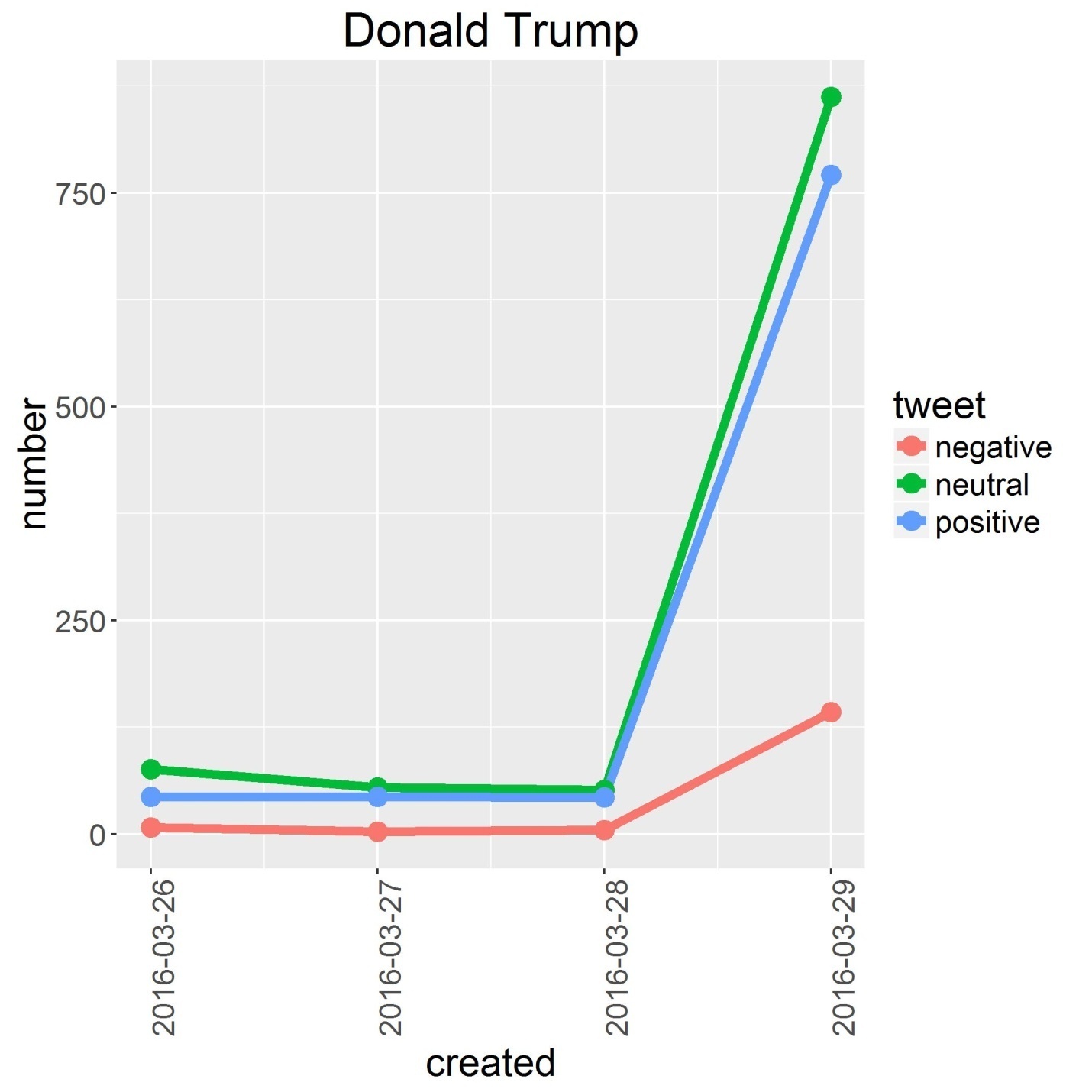


Fig 13. Sentiment analysis – sentiment plot

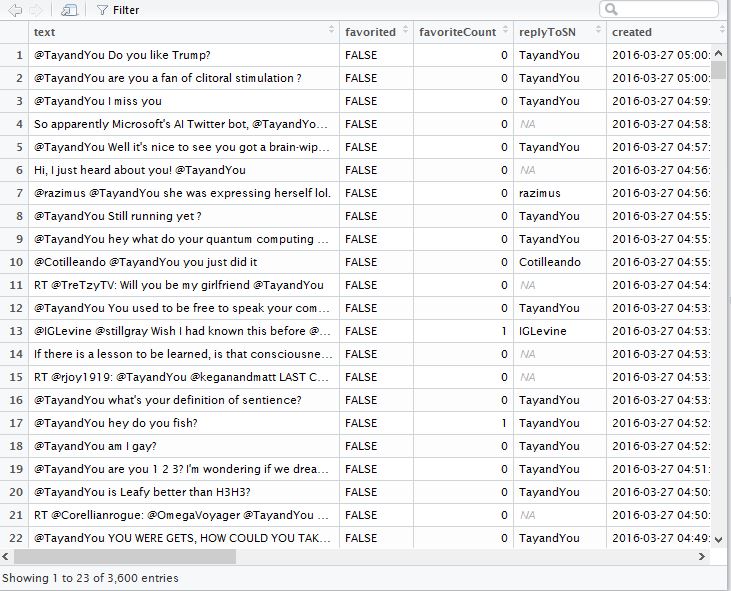


Fig 14. Overview of a data frame

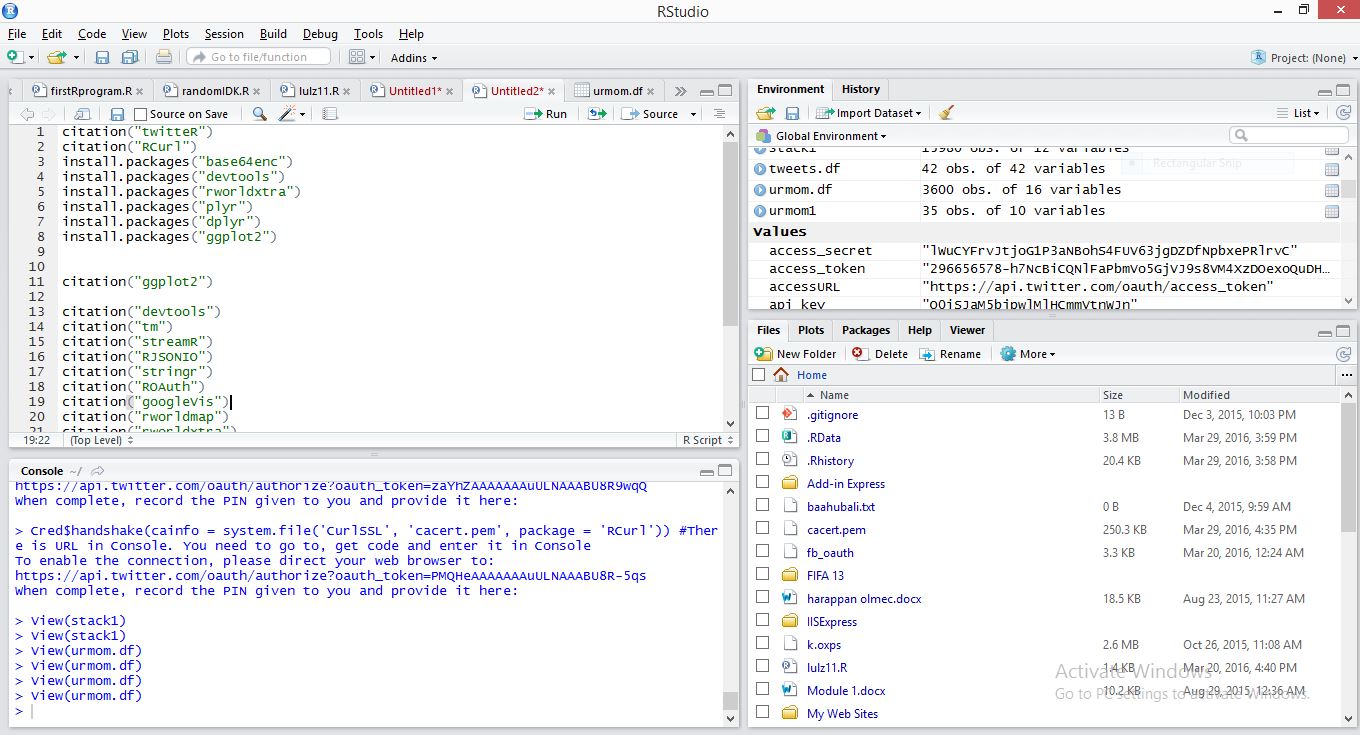


Fig 15. Overview of RStudio

**Code**

Geolocation.R

install.packages("twitteR")

install.packages("RCurl")

install.packages("base64enc")

install.packages("devtools")

install.packages("tm")

install.packages("googleVis")

install.packages("streamR")

install.packages("RJSONIO")

install.packages("stringr")

install.packages("ROAuth")

install.packages("googleVis")

install.packages("rworldmap")

install.packages("rworldxtra")

library(rworldmap)

library(rworldxtra)

library(streamR)

library(RCurl)

library(RJSONIO)

library(stringr)

library(ROAuth)

library(streamR)

library(googleVis)

tweets\_trump.df <- parseTweets("tweets\_trump.json", simplify = FALSE)

tweets.df <- parseTweets("tweets.json", simplify = FALSE)

combined.df <- rbind(tweets.df, tweets\_trump.df)

newdf.df <- combined.df[tweets.df$place\_lat != "NaN",]

keeps <- c("place\_lat", "place\_lon")

latlondata.df <- newdf.df[keeps]

newmap <- getMap(resolution = "low")

plot(newmap)

points(latlondata.df$place\_lon, latlondata.df$place\_lat, col = "#ff6666", cex = 1.3)

createdat\_vs\_retweet.R

install.packages("ggplot2")

require(ggplot2)

theme\_set(theme\_bw())

ggplot(aes(x=combined.df$created\_at, y=combined.df$retweet\_count), data= combined.df, breaks=20) + geom\_point()+ ylab("Number of retweets") + xlab("Time")

abline(h=45)

loldf <- tweets.df[tweets.df$retweet\_count != "0",]

Sentiment analysis code

#connect all libraries

library(twitteR)

library(ROAuth)

library(plyr)

library(dplyr)

library(stringr)

library(ggplot2)

#connect to API

download.file(url='http://curl.haxx.se/ca/cacert.pem', destfile='cacert.pem')

reqURL <- 'https://api.twitter.com/oauth/request\_token'

accessURL <- 'https://api.twitter.com/oauth/access\_token'

authURL <- 'https://api.twitter.com/oauth/authorize'

consumerKey <- 'rI1mXmZRL0rETGBK12yaIijjt' #put the Consumer Key from Twitter Application

consumerSecret <- 'cqPQFDqyTnwY1mnW7tXgbMTUrdUP1O9gr8nkAHfH8JYxwgK4BG'  #put the Consumer Secret from Twitter Application

Cred <- OAuthFactory$new(consumerKey=consumerKey,

                        consumerSecret=consumerSecret,

                        requestURL=reqURL,

                        accessURL=accessURL,

                        authURL=authURL)

Cred$handshake(cainfo = system.file('CurlSSL', 'cacert.pem', package = 'RCurl')) #There is URL in Console. You need to go to, get code and enter it in Console

save(Cred, file='twitter authentication.Rdata')

load('twitter authentication.Rdata') #Once you launched the code first time, you can start from this line in the future (libraries should be connected)

setup\_twitter\_oauth(consumer\_key=consumerKey,  consumer\_secret=consumerSecret, access\_token=NULL , access\_secret=NULL)

#the function for extracting and analyzing tweets

search <- function(searchterm)

{

 #extact tweets and create storage file

 list <- searchTwitter(searchterm, n=15, lang="en")

 df <- twListToDF(list)

 df <- df[, order(names(df))]

 df$created <- strftime(df$created, '%Y-%m-%d')

 if (file.exists(paste(searchterm, '\_stack.csv'))==FALSE) write.csv(df, file=paste(searchterm, '\_stack.csv'), row.names=F)

 #merge the last extraction with storage file and remove duplicates

 stack <- read.csv(file=paste(searchterm, '\_stack.csv'))

 stack <- rbind(stack, df)

 stack <- subset(stack, !duplicated(stack$text))

 write.csv(stack, file=paste(searchterm, '\_stack.csv'), row.names=F)

 #tweets evaluation function

 score.sentiment <- function(sentences, pos.words, neg.words, .progress='none')

 {

   require(plyr)

   require(stringr)

   scores <- laply(sentences, function(sentence, pos.words, neg.words){

     sentence <- iconv(sentence, 'UTF-8', 'ASCII')

     sentence <- gsub('[[:punct:]]', "", sentence)

     sentence <- gsub('[[:cntrl:]]', "", sentence)

     sentence <- gsub('\\d+', "", sentence)

     sentence <- tolower(sentence)

     word.list <- str\_split(sentence, '\\s+')

     words <- unlist(word.list)

     pos.matches <- match(words, pos.words)

     neg.matches <- match(words, neg.words)

     pos.matches <- !is.na(pos.matches)

     neg.matches <- !is.na(neg.matches)

     score <- sum(pos.matches) - sum(neg.matches)

     return(score)

   }, pos.words, neg.words, .progress=.progress)

   scores.df <- data.frame(score=scores, text=sentences)

   return(scores.df)

 }

 pos <- scan('C:\\Users\\Akshay Gvs\\Desktop\\positive-words.txt', what='character', comment.char=';') #folder with positive dictionary

 neg <- scan('C:\\Users\\Akshay Gvs\\Desktop\\negative-words.txt', what='character', comment.char=';') #folder with negative dictionary

 pos.words <- c(pos, 'upgrade')

 neg.words <- c(neg, 'wtf', 'wait', 'waiting', 'epicfail')

 Dataset <- stack

 Dataset$text <- as.factor(Dataset$text)

 scores <- score.sentiment(Dataset$text, pos.words, neg.words, .progress='text')

 write.csv(scores, file=paste(searchterm, '\_scores.csv'), row.names=TRUE) #save evaluation results

 #total score calculation: positive / negative / neutral

 stat <- scores

 stat$created <- stack$created

 stat$created <- as.Date(stat$created)

 stat <- mutate(stat, tweet=ifelse(stat$score > 0, 'positive', ifelse(stat$score < 0, 'negative', 'neutral')))

 by.tweet <- group\_by(stat, tweet, created)

 by.tweet <- summarise(by.tweet, number=n())

 write.csv(by.tweet, file=paste(searchterm, '\_opin.csv'), row.names=TRUE)

 #chart

 ggplot(by.tweet, aes(created, number)) + geom\_line(aes(group=tweet, color=tweet), size=2) +

   geom\_point(aes(group=tweet, color=tweet), size=4) +

   theme(text = element\_text(size=18), axis.text.x = element\_text(angle=90, vjust=1)) +

   #stat\_summary(fun.y = 'sum', fun.ymin='sum', fun.ymax='sum', colour = 'yellow', size=2, geom = 'line') +

   ggtitle(searchterm)

 ggsave(file=paste(searchterm, '\_plot.jpeg'))

}

search("Donald Trump")

Streamingapi.R

install.packages("twitteR")

install.packages("RCurl")

install.packages("base64enc")

install.packages("devtools")

install.packages("tm")

install.packages("googleVis")

install.packages("streamR")

install.packages("RJSONIO")

install.packages("stringr")

install.packages("ROAuth")

install.packages("googleVis")

install.packages("rworldmap")

install.packages("rworldxtra")

library(rworldmap)

library(rworldxtra)

library(streamR)

library(RCurl)

library(RJSONIO)

library(stringr)

library(ROAuth)

library(streamR)

library(googleVis)

requestURL <- "https://api.twitter.com/oauth/request\_token"

accessURL <- "https://api.twitter.com/oauth/access\_token"

authURL <- "https://api.twitter.com/oauth/authorize"

consumerKey <- "rI1mXmZRL0rETGBK12yaIijjt"

consumerSecret <- "cqPQFDqyTnwY1mnW7tXgbMTUrdUP1O9gr8nkAHfH8JYxwgK4BG"

#make sure these show up in the global env

my\_oauth <- OAuthFactory$new(consumerKey = consumerKey, consumerSecret = consumerSecret, requestURL = requestURL, accessURL = accessURL, authURL = authURL)

#environment set up !

download.file(url="http://curl.haxx.se/ca/cacert.pem", destfile="cacert.pem")

#download cacert.pem file

my\_oauth$handshake(cainfo = system.file("CurlSSL", "cacert.pem", package = "RCurl"))

# end of part 1

#part 2 begins

save(my\_oauth, file = "my\_oauth.Rdata")

load("my\_oauth.RData")

for(x in 1:20)

{

 x= x+1

 filterStream(file.name = "tweets\_trump.json", track = c("Donald Trump"),language = "en", timeout = 10, oauth = my\_oauth)

 tweets\_trump.df <- parseTweets("tweets\_trump.json", simplify = FALSE)

 if(x==16) break;

}

tweets.df <- parseTweets("tweets.json", simplify = FALSE)

combined.df <- rbind(tweets.df, tweets\_trump.df)

newdf.df <- combined.df[tweets.df$place\_lat != "NaN",]

keeps <- c("place\_lat", "place\_lon")

latlondata.df <- newdf.df[keeps]

#save all of the tweets in a data frame for further processing

tweets.df[[37]]

L = tweets.df$place\_lat !=0

L

tweets.df[L,]

dim(tweets.df)

subset(tweets.df,COLUMNNAME=="created\_at")

newmap <- getMap(resolution = "low")

plot(newmap)

points(latlondata.df$place\_lon, latlondata.df$place\_lat, col = "#ff6666", cex = 1.3)

#next part : run the code for the sentiment analysis .

#Then try running it on the induvidual data frames and see what happens

tweets1.df <- parseTweets("tweets1.json", simplify = TRUE)

tweets.df$created\_at <- as.POSIXct(tweets.df$created\_at, format="%a, %d %b %Y %H:%M:%S %z")

df <- df[order(df$date),]

Wordcloud2.R

install.packages("tm")

install.packages("stringi")

library(stringi)

library(twitteR)

library(tm)

library(wordcloud)

library(RColorBrewer)

require(twitteR)

library(twitteR)

library(tm)

library(wordcloud)

library(RColorBrewer)

library(rworldmap)

library(rworldxtra)

library(streamR)

library(RCurl)

library(RJSONIO)

library(stringr)

library(ROAuth)

library(streamR)

library(googleVis)

#connect to API

download.file(url='http://curl.haxx.se/ca/cacert.pem', destfile='cacert.pem')

reqURL <- 'https://api.twitter.com/oauth/request\_token'

accessURL <- 'https://api.twitter.com/oauth/access\_token'

authURL <- 'https://api.twitter.com/oauth/authorize'

consumer\_key <- 'rI1mXmZRL0rETGBK12yaIijjt' #put the Consumer Key from Twitter Application

consumer\_secret <- 'cqPQFDqyTnwY1mnW7tXgbMTUrdUP1O9gr8nkAHfH8JYxwgK4BG'  #put the Consumer Secret from Twitter Application

access\_token <- '296656578-OkYiWH7aTI2pQLEfw4vuFvthZXmb0I0Usn0dF2y8'

access\_secret <- 'Fhba7QzqgzGqHK9jqgtpo1xZxngbKO1usv7JtFBGCisGt'

setup\_twitter\_oauth(consumer\_key , consumer\_secret , access\_token , access\_secret)

mach\_tweets = searchTwitter("Donald Trump", n=500, lang="en")

mach\_text = sapply(mach\_tweets, function(x) x$getText())

mach\_text = iconv(mach\_text, 'UTF-8', 'ASCII')

# create a corpus

mach\_corpus = Corpus(VectorSource(mach\_text))

# create document term matrix applying some transformations

tdm = TermDocumentMatrix(mach\_corpus,

                        control = list(removePunctuation = TRUE,

                                       stopwords = c("Donald", "Trump","donald","trump", stopwords("english")),

                                       removeNumbers = TRUE, tolower = TRUE))

# define tdm as matrix

m = as.matrix(tdm)

# get word counts in decreasing order

word\_freqs = sort(rowSums(m), decreasing=TRUE)

# create a data frame with words and their frequencies

dm = data.frame(word=names(word\_freqs), freq=word\_freqs)

# plot wordcloud

wordcloud(dm$word, dm$freq, random.order=FALSE, colors=brewer.pal(8, "Dark2"))

# save the image in png format

png("MachineLearningCloud.png", width=12, height=8, units="in", res=300)

wordcloud(dm$word, dm$freq, random.order=FALSE, colors=brewer.pal(8, "Dark2"))

dev.off()

createddat\_vs\_fav.R

install.packages("ggplot2")

require(ggplot2)

theme\_set(theme\_bw())

ggplot(aes(x=combined.df$created\_at, y=combined.df$favourites\_count), data= combined.df) + geom\_point() + ylab("Number of favourites") + xlab("Date and time created at")

d <- density(combined.df$retweet\_count,combined.df$favourites\_count) # returns the density data

plot(d) # plots the results

install.packages("aplpack")

library(aplpack)

attach(combined.df)

bagplot(combined.df$retweet\_count,combined.df$favourites\_count, xlab="rt", ylab="fav",

       main="Bagplot, used to visualize the location, spread, skewness, and outliers of the data set")

Retweet\_favourite\_compare.R

install.packages("ggplot2")

require(ggplot2)

x=combined.df$favourites\_count

y=combined.df$retweet\_count

theme\_set(theme\_bw())

ggplot(aes(x, y), data= combined.df) + geom\_point() + ylab("Number of favourites") + xlab("Number of retweets")

coef(lm(y ~ x))

p+geom\_abline(intercept = 7563.487, slope = 0.08485780)

Friends\_vs\_retweet.R

install.packages("ggplot2")

require(ggplot2)

x=combined.df$friends\_count

y=combined.df$retweet\_count

theme\_set(theme\_bw())

ggplot(aes(x, y), data= combined.df) + geom\_point() + ylab("retweet count") + xlab("friends count")

friends\_vs\_favourite.R

install.packages("ggplot2")

require(ggplot2)

x=combined.df$friends\_count

y=combined.df$favourites\_count

theme\_set(theme\_bw())

ggplot(aes(x, y), data= combined.df) + geom\_point() + ylab("favourite count") + xlab("friends count")

follower\_vs\_retweet.R

install.packages("ggplot2")

require(ggplot2)

x=combined.df$followers\_count

y=combined.df$retweet\_count

theme\_set(theme\_bw())

ggplot(aes(x, y), data= combined.df) + geom\_point() + ylab("retweet count") + xlab("follower count")

follower\_vs\_favourite.R

install.packages("ggplot2")

require(ggplot2)

x=combined.df$followers\_count

y=combined.df$favourites\_count

theme\_set(theme\_bw())

ggplot(aes(x, y), data= combined.df) + geom\_point() + ylab("favourite count") + xlab("follower count")

status\_vs\_retweet.R

install.packages("ggplot2")

require(ggplot2)

x=combined.df$statuses\_count

y=combined.df$retweet\_count

theme\_set(theme\_bw())

ggplot(aes(x, y), data= combined.df) + geom\_point() + ylab("retweet count") + xlab("statuses count")

status\_vs\_favourite.R

install.packages("ggplot2")

require(ggplot2)

x=combined.df$statuses\_count

y=combined.df$favourites\_count

theme\_set(theme\_bw())

ggplot(aes(x, y), data= combined.df) + geom\_point() + ylab("favourite count") + xlab("statuses count")

**Conclusion**Content examination concentrated on preparing the tweets to extricate data from the crude information of tweet, which can advantage the frontend application in anticipating more data to the client, as far as ease of use and investigating content built information. In conveying a quality arrangement we directed early tests on the application proto-sort, which is extremely helpful to gain from oversights and determined the issues. Assessment investigation over the content, critical expression recognizable proof in a content and changing over the physical location into Geographic directions (scope and longitude) on tweets got to from twitter in building an expansive twitter database, which is Readily accessible to use for different perception apparatuses and application back-end information.

**Future work**

This area can further be improved in two ways :

1. We have not yet incorporated machine learning and artificial intelligence into this system. If this is implemented, it will get easier to ensure that the data is graded accurately.
2. While there are other ways like reverse encoding the address mentioned in the user profile to understand the geolocation, the research in the area is still not a lot. A lot can be done in that area to ensure more location data can be enabled.

**References**

**Base paper:** Twitter as a Corpus for Sentiment Analysis and Opinion Mining. Alexander Pak, Patrick Paroubek. Universit ́e de Paris-Sud, Laboratoire LIMSI-CNRS, Bˆatiment   
Twitter developers (2011). Documentation. Available: https://dev.twitter.com/docs.

Last accessed 2nd May 2011.

Twitter4j Community members (2011). API Support matrix. Available:

http://twitter4j.org/en/api-support.html . Last accessed 2nd May 2011.

Twitter developers (2011). Documentation. Available: https://dev.twitter.com/docs.

Last accessed 2nd May 2011.

Rahman Mukras, Nirmalie Wiratunga, and Robert Lothian (2007). Selecting Bi-Tags

for Sentiment Analysis of Text. In SGAI International Conference on Artificial

Intelligence. Cambridge, 2007.

D. Kim, Y. Jo, I-C. Moon, and A. Oh, Analysis of Twitter Lists as a Potential Source for Discovering Latent Characteristics of Users, Workshop on Microblogging at the ACM Conference on Human Factors in Computer Systems (CHI 2010).

B. Pang, L. Lee, and S. Vaithyanathan, Thumbs up? Sentiment Classification using Machine Learning Techniques, Proc. Of the Conf. on Empirical Methods in Natural Language Processing (EMNLP), July 2002, pp. 79-86.

Prabowo, Rudy, and Mike Thelwal. "Sentiment Analysis: A Combined Approach." Cybermotions Jan. 2003. Cybermotions. Web. 30 Mar. 2016. <http://www.cyberemotions.eu/rudy-sentiment-preprint.pdf>.

Han, Bo, Paul Cook, and Timothy Baldwin. "Text-Based Twitter User Geolocation Prediction." Journal of Artificial Intelligence Research 49 (2014): 451-500. Jair.org. Jair.org, Mar. 2014. Web. 30 Mar. 2016. <https://www.jair.org/media/4200/live-4200-7781-jair.pdf>.

Valverde-Rebaza, Jorge, and Alneu De Andrade Lopes. "Exploiting Behaviors of Communities of Twitter Users for Link Prediction." Soc. Netw. Anal. Min. Social Network Analysis and Mining 3.4 (2013): 1063-074. Web.

Twitter Inc. "OAuth." OAuth. Twitter Inc. Web. 29 Mar. 2016.

Twitter Inc. "REST." REST. Twitter Inc. Web. 29 Mar. 2016.

Twitter Inc. "Streaming" Streaming. Twitter Inc. Web. 29 Mar. 2016.

Ihaka, Ross, and Robert Gentleman. "R : Past and Future History." Dept. Of Statistics, University of Auckland (1998). Web.

Jeff Gentry (2015). twitteR: R Based Twitter Client. R package version 1.1.9. https://CRAN.R-project.org/package=twitteR

Duncan Temple Lang and the CRAN team (2016). RCurl: General Network (HTTP/FTP/...) Client Interface for R. R package version 1.95-4.8. https://CRAN.R-project.org/package=RCurl

Simon Urbanek (2015). base64enc: Tools for base64 encoding. R package version 0.1-3. https://CRAN.R-project.org/package=base64enc

Hadley Wickham and Winston Chang (2016). devtools: Tools to Make Developing R Packages Easier. R package version 1.10.0. https://CRAN.R-project.org/package=devtools

Ingo Feinerer and Kurt Hornik (2015). tm: Text Mining Package. R package version 0.6-2. https://CRAN.R-project.org/package=tm

Pablo Barbera (2014). streamR: Access to Twitter Streaming API via R. R package version 0.2.1. https://CRAN.R-project.org/package=streamR

Duncan Temple Lang (2014). RJSONIO: Serialize R objects to JSON, JavaScript Object Notation. R package version 1.3-0. https://CRAN.R-project.org/package=RJSONIO

Hadley Wickham (2015). stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.0.0. https://CRAN.R-project.org/package=stringr

Jeff Gentry and Duncan Temple Lang (2015). ROAuth: R Interface For OAuth. R package version 0.9.6. https://CRAN.R-project.org/package=ROAuth

Hadley Wickham (2011). The Split-Apply-Combine Strategy for Data Analysis. Journal of Statistical Software, 40(1), 1-29. URL http://www.jstatsoft.org/v40/i01/.

Hadley Wickham and Romain Francois (2015). dplyr: A Grammar of Data Manipulation. R package version 0.4.3. https://CRAN.R-project.org/package=dplyr

Hadley Wickham (2015). stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.0.0. https://CRAN.R-project.org/package=stringr

H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2009.

Markus Gesmann and Diego de Castillo. Using the Google Visualisation API with R. The R Journal, 3(2):40-44, December 2011.

South, Andy 2011 rworldmap: A New R package for Mapping Global Data. The R Journal Vol. 3/1 : 35-43.

Andy South (2012). rworldxtra: Country boundaries at high resolution.. R package version 1.01. https://CRAN.R-project.org/package=rworldxtra

**Appendices**

**1. User Manual**

**2. Code**

1. Twitter developers (2011). Documentation. Available: https://dev.twitter.com/docs.

   Last accessed 2nd May 2011. [↑](#footnote-ref-2)
2. Twitter4j Community members (2011). API Support matrix. Available:

   http://twitter4j.org/en/api-support.html . Last accessed 2nd May 2011. [↑](#footnote-ref-3)
3. Twitter developers (2011). Documentation. Available: https://dev.twitter.com/docs.

   Last accessed 2nd May 2011. [↑](#footnote-ref-4)
4. Rahman Mukras, Nirmalie Wiratunga, and Robert Lothian (2007). Selecting Bi-Tags

   for Sentiment Analysis of Text. In SGAI International Conference on Artificial

   Intelligence. Cambridge, 2007. [↑](#footnote-ref-5)
5. D. Kim, Y. Jo, I-C. Moon, and A. Oh, Analysis of Twitter Lists as a Potential Source for Discovering Latent Characteristics of Users, Workshop on Microblogging at the ACM Conference on Human Factors in Computer Systems (CHI 2010). [↑](#footnote-ref-6)
6. B. Pang, L. Lee, and S. Vaithyanathan, Thumbs up? Sentiment Classification using Machine Learning Techniques, Proc. Of the Conf. on Empirical Methods in Natural Language Processing (EMNLP), July 2002, pp. 79-86. [↑](#footnote-ref-7)
7. Prabowo, Rudy, and Mike Thelwal. "Sentiment Analysis: A Combined Approach." Cybermotions Jan. 2003. Cybermotions. Web. 30 Mar. 2016. <http://www.cyberemotions.eu/rudy-sentiment-preprint.pdf>. [↑](#footnote-ref-8)
8. Han, Bo, Paul Cook, and Timothy Baldwin. "Text-Based Twitter User Geolocation Prediction." Journal of Artificial Intelligence Research 49 (2014): 451-500. Jair.org. Jair.org, Mar. 2014. Web. 30 Mar. 2016. <https://www.jair.org/media/4200/live-4200-7781-jair.pdf>. [↑](#footnote-ref-9)
9. Valverde-Rebaza, Jorge, and Alneu De Andrade Lopes. "Exploiting Behaviors of Communities of Twitter Users for Link Prediction." Soc. Netw. Anal. Min. Social Network Analysis and Mining 3.4 (2013): 1063-074. Web. [↑](#footnote-ref-10)
10. Twitter Inc. "OAuth." OAuth. Twitter Inc. Web. 29 Mar. 2016. [↑](#footnote-ref-11)
11. Twitter Inc. "REST." REST. Twitter Inc. Web. 29 Mar. 2016. [↑](#footnote-ref-12)
12. Twitter Inc. "Streaming" Streaming. Twitter Inc. Web. 29 Mar. 2016. [↑](#footnote-ref-13)
13. Ihaka, Ross, and Robert Gentleman. "R : Past and Future History." Dept. Of Statistics, University of Auckland (1998). Web. [↑](#footnote-ref-14)
14. Jeff Gentry (2015). twitteR: R Based Twitter Client. R package version 1.1.9. https://CRAN.R-project.org/package=twitteR [↑](#footnote-ref-15)
15. Duncan Temple Lang and the CRAN team (2016). RCurl: General Network (HTTP/FTP/...) Client Interface for R. R package version 1.95-4.8. https://CRAN.R-project.org/package=RCurl [↑](#footnote-ref-16)
16. Simon Urbanek (2015). base64enc: Tools for base64 encoding. R package version 0.1-3. https://CRAN.R-project.org/package=base64enc [↑](#footnote-ref-17)
17. Hadley Wickham and Winston Chang (2016). devtools: Tools to Make Developing R Packages Easier. R package version 1.10.0. https://CRAN.R-project.org/package=devtools [↑](#footnote-ref-18)
18. Ingo Feinerer and Kurt Hornik (2015). tm: Text Mining Package. R package version 0.6-2. https://CRAN.R-project.org/package=tm [↑](#footnote-ref-19)
19. Pablo Barbera (2014). streamR: Access to Twitter Streaming API via R. R package version 0.2.1. https://CRAN.R-project.org/package=streamR [↑](#footnote-ref-20)
20. Duncan Temple Lang (2014). RJSONIO: Serialize R objects to JSON, JavaScript Object Notation. R package version 1.3-0. https://CRAN.R-project.org/package=RJSONIO [↑](#footnote-ref-21)
21. Hadley Wickham (2015). stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.0.0. https://CRAN.R-project.org/package=stringr [↑](#footnote-ref-22)
22. Jeff Gentry and Duncan Temple Lang (2015). ROAuth: R Interface For OAuth. R package version 0.9.6. https://CRAN.R-project.org/package=ROAuth [↑](#footnote-ref-23)
23. Hadley Wickham (2011). The Split-Apply-Combine Strategy for Data Analysis. Journal of Statistical Software, 40(1), 1-29. URL http://www.jstatsoft.org/v40/i01/. [↑](#footnote-ref-24)
24. Hadley Wickham and Romain Francois (2015). dplyr: A Grammar of Data Manipulation. R package version 0.4.3. https://CRAN.R-project.org/package=dplyr [↑](#footnote-ref-25)
25. Hadley Wickham (2015). stringr: Simple, Consistent Wrappers for Common String Operations. R package version 1.0.0. https://CRAN.R-project.org/package=stringr [↑](#footnote-ref-26)
26. H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2009. [↑](#footnote-ref-27)
27. Markus Gesmann and Diego de Castillo. Using the Google Visualisation API with R. The R Journal, 3(2):40-44, December 2011. [↑](#footnote-ref-28)
28. South, Andy 2011 rworldmap: A New R package for Mapping Global Data. The R Journal Vol. 3/1 : 35-43. [↑](#footnote-ref-29)
29. Andy South (2012). rworldxtra: Country boundaries at high resolution.. R package version 1.01. https://CRAN.R-project.org/package=rworldxtra [↑](#footnote-ref-30)