Major Project

Sentiment Analysis of Twitter and Amazon Data using R Programming

By

RANDEEP SINGH

2K16/EMBA/526

Under Guidance of

DR. RAJAN YADAV

Professor, Delhi School of Management

Submitted in partial fulfillment of the requirement for award of

Master of Business Administration (EXECUTIVE)

Delhi School of Management
Delhi Technological University
Delhi-110042

DISCLAIMER

The views expressed in this project are personal and not of the organization and this project is done as a detailed study under the course from strategy perspective only.

DECLARATION

This is to certify that the project entitled '**Sentiment Analysis of Twitter and Amazon Data using R Programming**' has been successfully completed by **Randeep Singh** – **2K16/EMBA/526**. This is further certified that this project work is a record of bonafide work done by me.

Place: New Delhi	**Randeep Singh**
Date:	2K16/EMBA/526

Certificate

This is to certify that the project entitled ‘**Sentiment Analysis of Twitter and Amazon Data using R Programming**’ has been successfully completed by **Randeep Singh** – **2K16/EMBA/526**.

This is further certified that this project work is a record of bonafide work done by him under my guidance. The matter embodied in this report has not been submitted for award of any degree.

DR. RAJAN YADAV
Professor
Delhi School of Management
Delhi Technological University

Acknowledgement

I, Randeep Singh, wish to extend my gratitude to **Dr. Rajan Yadav**, Professor, Delhi School of Management (DSM), Delhi Technological University; for giving me all the guidance and valuable insights to take up this Semester Project.

I also take this opportunity to convey sincere thanks to all the faculty members for directing and advising during the course.

ABSTRACT

The advancement in the field of Internet, digital and social media has increased data sets to larger extent. **Sentiment Analysis** is the methodology of computationally identifying and categorizing opinions expressed in a piece of text, in order to determine whether the opinion writer's attitude towards a particular topic, product etc. is **positive**, **negative**, or **neutral**.

Real-time sentiment analysis is a challenging **machine learning task**, due to scarcity of labeled data and sudden changes in sentiment caused by real-world events that need to be instantly interpreted. In this project I propose solutions to save user time that they spend reading all the opinions on a particular topic or going through reviews about a product. And, help them make a better an instant informed decision. The attempt is to develop a system which could aggregate the reviews and opinions from various web sources using machine learning algorithm and provide an unbiased insight of consumer sentiments towards a product.

I have used **R programming techniques** to build a generalized solution for data extraction, data mining, data analysis and data visualization from web sources for **evaluating User Sentiments** towards a program or product.

The experiment has been conducted on below two areas:

- **Sentiments of people towards Make in India initiative.**
- **Customer Sentiments towards Baby products in Amazon.**

Data used in this study are tweets extracted for Make In India campaign from Twitter and online customer reviews for baby products collected from Amazon.com. The different **Data Analytics techniques** are used to extract or discover knowledge from the tweets and posted reviews so as to generate interesting insights of user's attitude towards an event or a particular product.

My research covers work ranging from real time stream data processing, real time content search, event detection, link mining, behavior mining, and sentiment analysis. I will also feature my ongoing system research that aims to support user-friendly real time online data analytics using publicly available content.

TABLE OF CONTENT

1

Introduction to Sentiment Analysis

Sentiment is a thought, an attitude, views, judgement or opinions prompted by feeling. **Sentiment analysis** studies user's sentiments towards certain objects. Internet Web is a resourceful and capable place with respect to sentiment information.

From a user's perspective, people are able to post their own content through various social media, such as forums, blogs, E-commerce sites or online social networking sites.

From a researcher's perspective, many social media sites release their application programming interfaces (APIs), prompting data collection and analysis by researchers and developers.

1.1 Sentiment Analysis: A Fascinating Problem

Sentiment Analysis is a process of research and evaluation that analyzes people's emotions, opinions, thoughts, viewpoints, sentiments and attitude towards objects or entities such as products, services, organizations, individuals, issues, events, topics, and their attributes.

There are also many different names with slight deviations in tasks,

- sentiment analysis
- opinion extraction
- sentiment mining
- opinion mining
- appraisal extraction
- subjectivity analysis
- emotion analysis
- affect analysis
- review mining

However, all of them fall under the umbrella of sentiment analysis or opinion mining.

Although natural language processing has been since years, little research has been done about people's viewpoints and sentiments. Post year 2000, the Opinion mining field has become a very active research area. There are several reasons for this.

- It has a wide arrange of applications in almost every domain. The industry adjoining sentiment analysis is flourishing due to the explosion of commercial applications. This provides a strong motivation for research.
- It offers many challenging research problems, which had never been studied before.
- For the first time in history, we have a huge volume of user opinionated data in the social media on the Web. A lot of research would not have been possible without this data. The rapid growth of sensitivity analysis coincides with that of the social media. In fact, it is now right at the core of the social media research.

"What other people think" has always been an important piece of information for most of us during the decision-making process. In the past before awareness of the Internet became widespread, we all used to ask our knowns (friends, relatives, colleagues) to recommend a car mechanic or to explain who and why they are planning to vote for in the public elections, consulted Consumer Reports to decide what product to buy or asked for reference letters regarding job applicants from colleagues. But the Internet and the Web have now (among other things) made it possible to find out about the experiences and opinions of those in the vast pool of users that are neither the personal acquaintances nor professional critics —that is, people we have never met or heard of. More and more people are making their views available to strangers via the Internet.

According to two surveys of more than 2000 American adults:

- 80% of Internet users have done online research on a product at least once;

- 20% do so on a typical day;

- among readers of online reviews of hotels, restaurants, and various services (e.g., travel or doctors), between 73% and 87% report that reviews had a vital influence on their purchase;

- consumers report that they are willing to pay from 20% to 99% more for a better 5-star-rated item than a 4-star-rated item.

- 32% have provided a rating on a service, product, brand or person via an online ratings medium, and 30% have posted an online comment or review regarding a brand, product or service.

I would like to point out that consumption and usage of products and services is not the only inspiration behind people’s looking for or expressing views or opinions online. A need for political and social information is another important factor.

The hunger and dependency of user upon online recommendations and advice that the data above discloses is one big reason behind the flood of interest in new systems that deal directly with opinions as a first-class object.

1.2 Early History

There has been an increasing interest in the field of Sentiment Analysis for quite a while now. The year 2001 onwards seems to mark the beginning of widespread awareness of the research problems and opportunities that sentiment analysis and opinion mining raise, and subsequently there have been literally hundreds of papers published on the subject.

The key factors behind this include:

- the rise of machine learning methods in natural language processing and information retrieval;

- the availability of datasets for machine learning algorithms to be trained on, due to the blossoming of the World Wide Web and, specifically, the development of review-aggregation web-sites; and, of course

- Realization of the attractive intellectual challenges and marketable and intelligence claims that the area deals with.

1.3 Sentiment Analysis Applications

With the speedy growth of social media that comprises of posting reviews, forum discussions, writing blogs/microblogs, tweets, comments, likes, ratings and postings in social network sites on the Web, users (individuals and organizations) have increasingly started using the content in these media for predictive analysis, forecasting and decision making. Today if consumer wants to buy a service or product, he or she is no longer limited to requesting friends, colleagues and family for views and ideas because there are many customer reviews, thoughts, opinions and discussions in public forums available on the Internet about the service or product.

However, searching, reviewing and monitoring opinion web pages on the Internet and cleansing the information contained in them remains a tough task because of the propagation of varied sites. Each site normally contains a enormous volume of judgement text that is not always easily decoded in long blogs posts and forum postings. The average reader will struggle identifying relevant websites and extracting, cleansing and summarizing the opinions in them. Automated opinion and sensitivity analysis systems are thus needed.

In recent years, we have witnessed that opinionated postings in social media have helped reshape businesses, and sway public sentiments and emotions, which have profoundly impacted on our social and political systems. Hence, it is critical to collect and study opinions data on the Web.

Due to these applications, industrial activities have flourished in recent years. Sensitivity analysis applications have spread to almost all possible area, from consumer products, services, politics, health and welfare, banks and financial services to cultural, social events and political events and elections. There have been at least 50-70 start-up companies in the USA alone. Many big corporations have also built their own in-house capabilities, e.g., Facebook, Amazon, Microsoft, Google, IBM, SAP and HP. These practical implementations and industrial interests have provided strong inspirations for research and study in sentiment analysis.

Sentiment Analysis can be done on all kind of texts, namely

- Newspaper texts
  - Financial News
  - Entertainment News
- Legal texts
- Novels
- E-mails
- SMS messages
- WhatsApp messages
- Customer Reviews
- Blog Posts
- Tweets
- Facebook Posts and so on.

Few of the **Applications of Sentiment Analysis** are

- Tracking sentiment towards politicians, movies, products.

- Improving Customer Relationship Models.

- Identifying what evokes strong emotions in people.

- Detecting happiness and well-being.

- Measuring the impact of activist movements through text generated in social media.

- Improving Automatic Dialogue systems.

- Improving Automatic Tutoring systems.

- Detecting how people use emotion-bearing-words and metaphors to persuade and coerce others.

1.4 Different Levels of Sentiment Analysis

Overall, Sentiment Analysis has been explored mainly at below levels:

1. **Document Level**:
The task at this level is to classify whether a whole text opinion document states a positive or negative sentiment. For e.g. if given a consumer product review, this process analyzes and determines whether the review comment expresses an overall opinion in that content about the particular product as neutral, positive or negative. This task is usually called as document level sentiment classification. This level of analysis assumes that each document expresses views on a single entity i.e. single product. Thus, it is not applicable to documents which assess or compare multiple objects and entities.

2. **Sentence Level**:
This task goes to the level of sentences and concludes if each of the sentence expressed a positive, negative or neutral opinion. Neutral typically means no opinion. This level of analysis is related to subjectivity classification closely which separates sentences (called objective sentences) that express factual information from sentences (called subjective sentences) that express subjective views and opinions. However, it should be noted that subjectivity is not equivalent to sentiment as many objective sentences can imply opinions, e.g., "*I bought the mobile yesterday and the screen has broken"*

3. **Entity and Aspect Level**:
The sentence level and document level analysis do not determine what exactly people liked and did not like. Aspect level performs in-depth analysis. Aspect level was previously called feature level i.e. feature-based opinion mining and summarization. Instead of looking at language constructs (documents, paragraphs, sentences, clauses or phrases), aspect level directly looks at the opinion itself. It is based on the idea that an opinion consists of a sentiment that is either positive or negative and a target. An opinion without identifying its target being is of partial or no use. We need to realize the importance of targets of opinions in order to understand the sentiment analysis problem better. For example, although the sentence "*although the manager is rude, I still love this hotel*" clearly has a positive tenor, we cannot say that this sentence is entirely positive. In fact, the sentence is positive about the hotel (emphasized), but negative about its manager (not emphasized). In many applications, opinion targets are described by objects or entities and/or their different aspects. Hence, the goal of this level of analysis is to discover sentiments on entities and/or their aspects.

1.5 Business Benefits of Sentiment Analysis

1) Better Customer Service

Sentiment Analysis gives valuable understandings about your present and future customers’ buying preferences, brand associations, opinions, interests, viewpoints, likes and dislikes in products/services/brands/events and much more. This vital information lets organizations to significantly improve their service and engagement activities by building on the positive viewpoints and formulating methods to combat negative sentiments.

2) Improve Brand

The organizations can calculate the perceptions about their brand, products and services, marketing activities, CSR initiatives, online information etc. in a quantitative manner. This information can be used for devising better and effective branding and marketing strategies and thus in improvement of brand status.

3) Be Competitive

Organizations can investigate sentiments about their competitors as well. This helps in benchmarking the performance against that of competitors. Using the sentiment analysis, the trends can be predicted and the specific strategies can be created to control these trends.

4) Gain Business Intelligence

Sentiment Analysis allows organizations by providing wide, intuitive information about their target user’s sentiments. These sentiments are like a gold mine of entry level entrepreneurs to explore new opportunities. Hence, Sentiment Analysis provides insightful business intelligence using which impactful decisions can be taken to monitor and prosper the business.

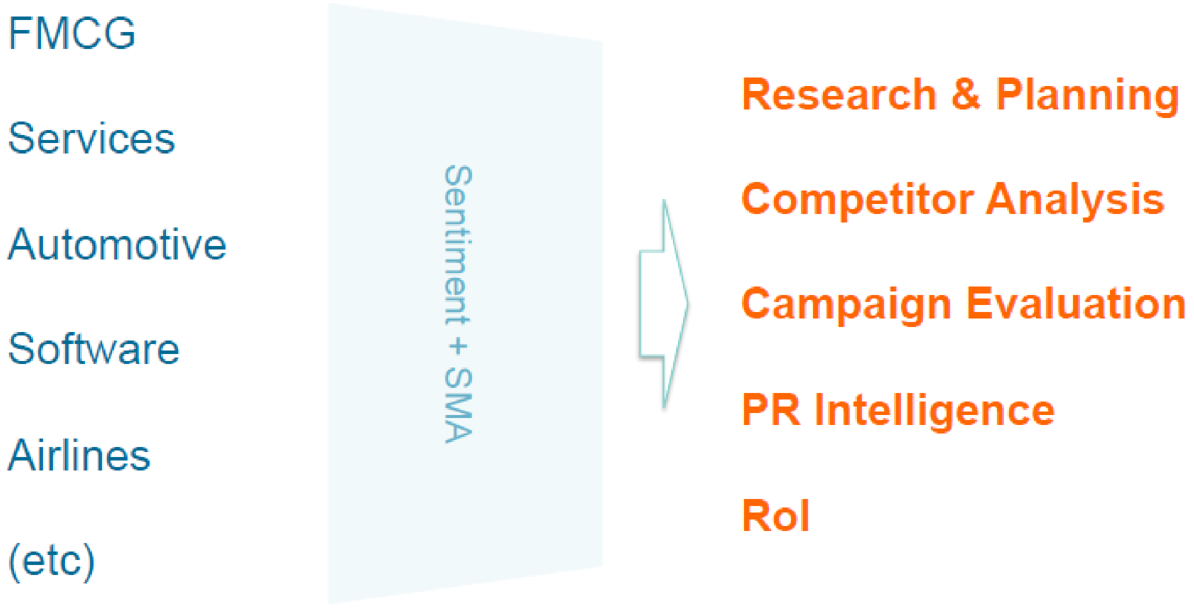
1.6 Use Cases of Sentiment Analysis

1. **Real-time Political Analysis**
   - **Data-driven media and journalism.**
   - **PR management for political figures and parties**

2. **Financial Analysis**

3. **Radicalization Detection**

4. **Marketing and Advertising**



2

Introduction to R Language

2.1 What is R?

R provides an open source environment for **statistical computing and graphics**. It is the product used actively by statisticians for a robust, programmable, portable, and open computing environment. It can provide solutions to the most complex and sophisticated problems,

- R has an effective **data handling and storage** facility.
- It contains a suite of **operators for calculations** of different matrices.
- It provides a vast, integrated collection of **different tools for data analysis**.
- It includes **graphical and visualization features** for data analysis and display on screen or on hardcopy.
- It is an easy and effective **coding language** which includes loops, arrays, conditionals, functions and input and output system.
- R provides a broad variety of **statistical analytical tools** for linear and nonlinear modelling, time-series analysis, classification, statistical tests etc)

The statisticians have studied and implemented many **specific statistical procedures** for a diverse variety of applications of different domains and developed **packages**, that are also freely available and integrate with R.

2.2 Advantages of R Language

- It is **completely free** and will always be so.
- It is **freely-available over the internet**.
- It **runs on many operating systems**:
- It is simple, **easy to learn, read and write.**
- It allows **statistical data analysis and visualization.**
- It can work on **objects of unlimited size and complexity** with a consistent, logical expression language.
- R has **lot of packages** and multiple ways of doing same things.

- It is supported by comprehensive **technical documentation** and user-contributed tutorials
- It is **fully programmable**, with its own sophisticated computer language. Monotonous procedures can easily be automated by user **scripts**. It is easy to write your own **functions**, and not too difficult to write whole **packages** if you invent some new analysis;
- All **source code** is published, so you can see the exact algorithms being used.
- It can **exchange data in MS-Excel**, text, fixed and delineated formats (e.g. CSV), so that existing datasets are easily imported, and results computed in R are easily exported.

2.3 R Console GUI

The default interface for both Windows and Mac OS/X is a simple GUI. It is referred to as “R console” because they provide a user friendly interface to the R command line, a script editor, a graphics output, and an online help. It does not consist of any menus dedicated for data manipulation or statistical analysis.

2.4 R Studio

RStudio is an excellent cross-platform IDE, integrated development environment of R. This environment includes the command line interface, a code editor, output graphs, history, help, workspace contents, and package manager all in one attractive interface.

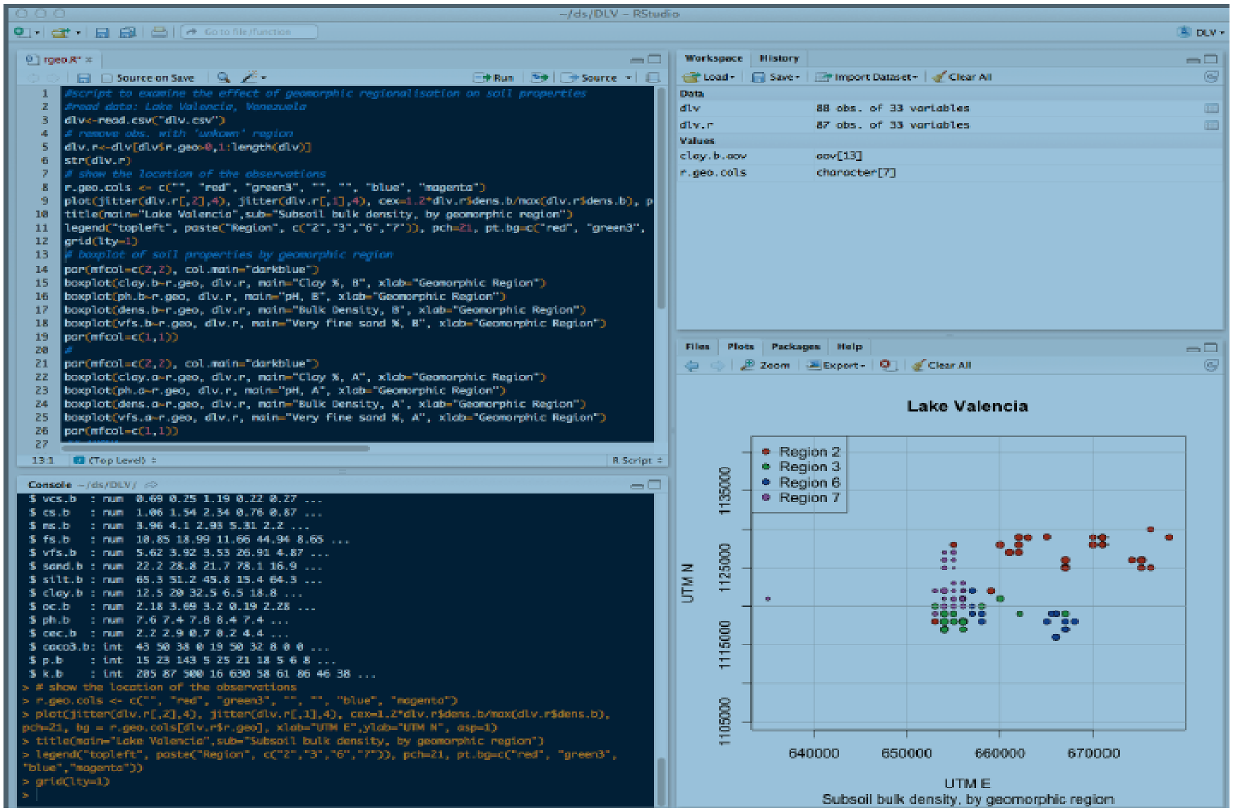
The typical use is:

1. Opening a script or starting a new script.
2. Alter the working directory to save script’s location.
3. Write R coding in the script.
4. Pass lines of code from the script to the command line and evaluate the output.
5. Examine any plots, graphs, wordclouds and save for later use
6. It involves editor for syntax highlighting, a console and debugging feature..

RStudio is an open supply software system although business versions are also provided with some improved features and it supports desktop computers which operate on windows, mac and Linux as well as on browser connected to RStudio.

Two versions available are:

a. **Rstudio desktop:** Software runs in the same way as desktop application.
b. **Rstudio server:** In this Rstudio is used to access web browser.



```r
#script to examine the effect of geomorphic regionalisation on soil properties
#read data: Lake Valencia, Venezuela
dlv<-read.csv("dlv.csv")
# remove obs. with 'unkown' region
dlv.r<-dlv[dlv$r.geo>0,1:length(dlv)]
str(dlv.r)
# show the location of the observations
r.geo.cols <- c("", "red", "green3", "", "", "blue", "magenta")
plot(jitter(dlv.r[,2],4), jitter(dlv.r[,1],4), cex=1.2*dlv.r$dens.b/max(dlv.r$dens.b), p
title(main="Lake Valencia",sub="Subsoil bulk density, by geomorphic region")
legend("topleft", paste("Region", c("2","3","6","7")), pch=21, pt.bg=c("red", "green3",
grid(lty=1)
# boxplot of soil properties by geomorphic region
par(mfcol=c(2,2), col.main="darkblue")
boxplot(clay.b~r.geo, dlv.r, main="Clay %, B", xlab="Geomorphic Region")
boxplot(ph.b~r.geo, dlv.r, main="pH, B", xlab="Geomorphic Region")
boxplot(dens.b~r.geo, dlv.r, main="Bulk Density, B", xlab="Geomorphic Region")
boxplot(vfs.b~r.geo, dlv.r, main="Very fine sand %, B", xlab="Geomorphic Region")
par(mfcol=c(1,1))
#
par(mfcol=c(2,2), col.main="darkblue")
boxplot(clay.a~r.geo, dlv.r, main="Clay %, A", xlab="Geomorphic Region")
boxplot(ph.a~r.geo, dlv.r, main="pH, A", xlab="Geomorphic Region")
boxplot(dens.a~r.geo, dlv.r, main="Bulk Density, A", xlab="Geomorphic Region")
boxplot(vfs.a~r.geo, dlv.r, main="Very fine sand %, A", xlab="Geomorphic Region")
par(mfcol=c(1,1))
```

```
 $ vcs.b  : num  0.69 0.25 1.19 0.22 0.27 ...
 $ cs.b   : num  1.06 1.54 2.34 0.76 0.87 ...
 $ ms.b   : num  3.96 4.1 2.93 5.31 2.2 ...
 $ fs.b   : num  10.85 18.99 11.66 44.94 8.65 ...
 $ vfs.b  : num  5.62 3.92 3.53 26.91 4.87 ...
 $ sand.b : num  22.2 28.8 21.7 78.1 16.9 ...
 $ silt.b : num  65.3 51.2 45.8 15.4 64.3 ...
 $ clay.b : num  12.5 20 32.5 6.5 18.8 ...
 $ oc.b   : num  2.18 3.69 3.2 0.19 2.28 ...
 $ ph.b   : num  7.6 7.4 7.8 8.4 7.4 ...
 $ cec.b  : num  2.2 2.9 0.7 0.2 4.4 ...
 $ caco3.b: int  43 50 38 0 19 50 32 8 0 0 ...
 $ p.b    : int  15 23 143 5 25 21 18 5 6 8 ...
 $ k.b    : int  285 87 500 16 630 58 61 86 46 38 ...
> # show the location of the observations
> r.geo.cols <- c("", "red", "green3", "", "", "blue", "magenta")
> plot(jitter(dlv.r[,2],4), jitter(dlv.r[,1],4), cex=1.2*dlv.r$dens.b/max(dlv.r$dens.b),
pch=21, bg = r.geo.cols[dlv.r$r.geo], xlab="UTM E",ylab="UTM N", asp=1)
> title(main="Lake Valencia",sub="Subsoil bulk density, by geomorphic region")
> legend("topleft", paste("Region", c("2","3","6","7")), pch=21, pt.bg=c("red", "green3",
"blue","magenta"))
> grid(lty=1)
> 
```

```
Data
dlv           88 obs. of 33 variables
dlv.r         87 obs. of 33 variables
Values
clay.b.aov    aov[13]
r.geo.cols    character[7]
```

The proposed work was carried out using RStudio Desktop. Features utilized were:

1. IDE was created specifically for R language.
    - Syntax is highlighted, completion of code and the smart indention.
    - From the source editor R program can be executed directly.
    - Rapidly switch to function definitions

2. Workflow is taken together.
    - Integrated R support and documentation.
    - Using projects multiple working directories can be easily managed.
    - Data viewer and workplace browser

3. Influential authoring and fixing.
    - Quickly detect and fix errors.
    - Tools Extensive package development.
    - Authoring with Sweave and R Markdown

3

Literature Review

Big data encompasses social networking web sites including **Twitter, Facebook and LinkedIn**. All these data sources are having many Applications in the real world. It consists of both structured and unstructured data of text, pictures and videos from which mining of knowledge regarding the latest workings of governments can be understood. It has been initially characterized by three V’s but now through five V’s. Recent days this is one of the most upcoming ones in the headlines; it is also fast-becoming a genuine force in originating planned insight and business intelligence which leads to strategic decisions from social media. Two additional dimensions of big data are variability and complexity i.e. varying data loads and to extract meaningful value from big data, we need ideal processing power and analytics capabilities.

Microblogging websites such as **Twitter** have grown to become a great source of various kinds of information. The typical reason is the nature of these microblogs on which users post real time tweets (messages) based on their viewpoints on different topics, discuss current affairs, complaints, and express positive, neutral or negative sentiment for goods they choose for daily usage.

Also in Today's era the **e-commerce** is developing rapidly these years, buying products on-line has become more and more fashionable owing to its variety of options, low cost value (high discounts) and quick supply systems, so abundant folks intend to do online shopping.. Users post review comments, ratings, opinions about product and also review about retailers from **Amazon**.

As the end users of these microblogging, ecommerce and social networks platforms grows rapidly, the data from these sources of information can be used in data mining and sentiment analysis tasks. For example, a manufacturing company may be interested in the following questions:

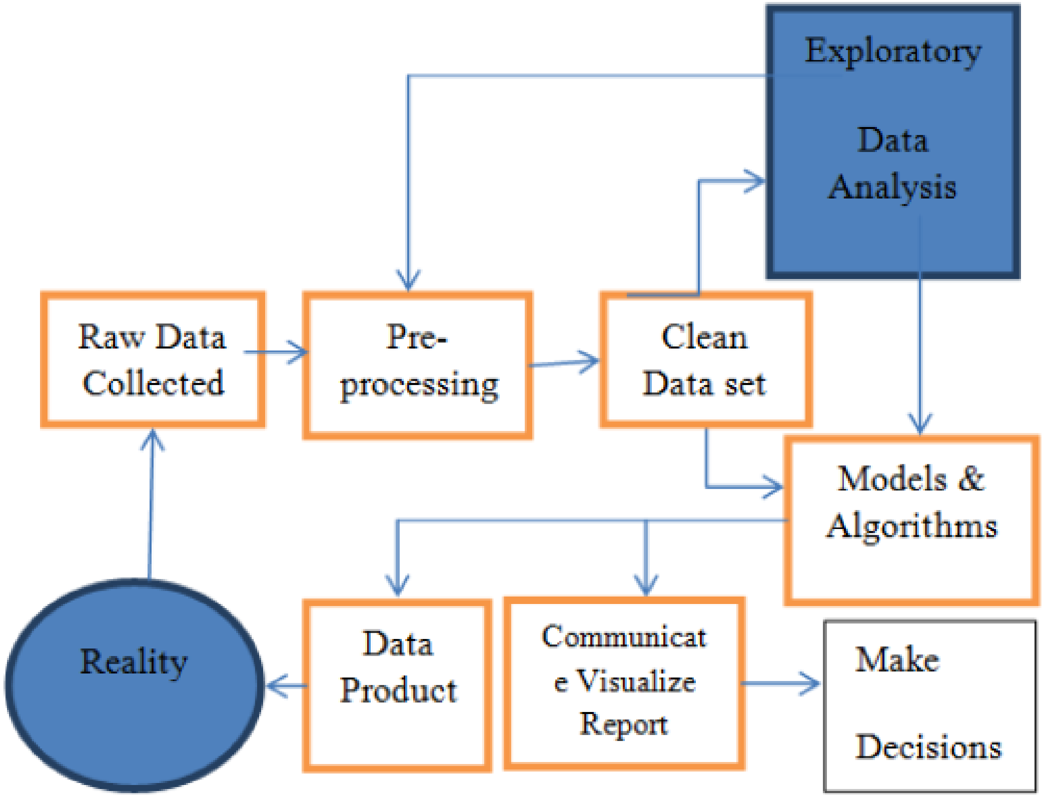
- What do people think about our product (service, product, company etc.)?
- How positive or negative are people about our product?
- What would people prefer our product to be like?

The Political parties and social organizations may be interested to know about people support, opinions and debates. All this information can be obtained from social networks, as their users post everyday what they like/dislike, and their opinions on many aspects of their life.

3.1 Data Analytics

Data analysis is the process of applying **organized and systematic statistical techniques to describe, recap, check and condense data**. It is a multistep process that involves collecting, cleaning, organizing and analyzing.

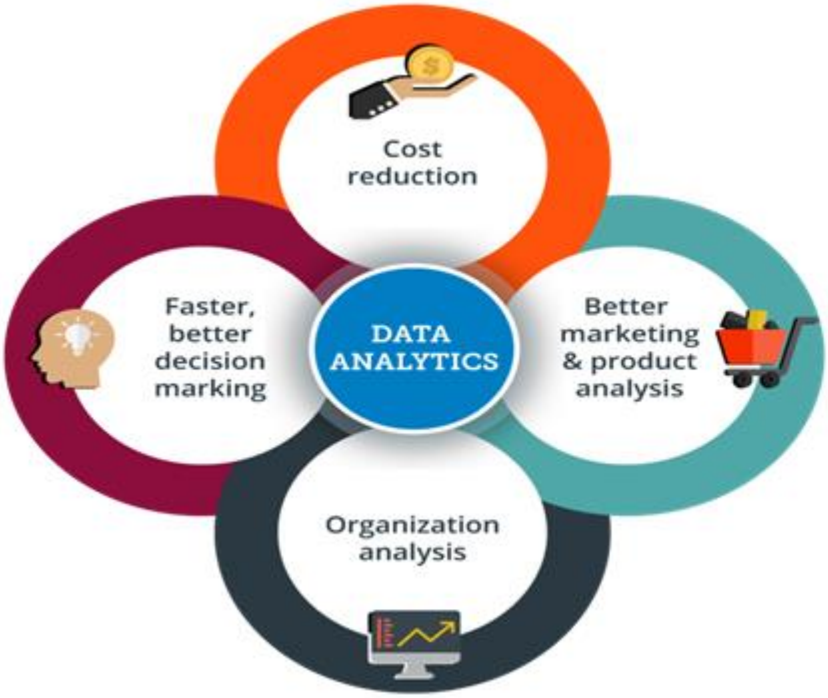
This extracted data is useful in suggesting modifications and supports decision-making. Data analysis has multiple aspects, approaches and uses. It has diverse techniques which can be applied in different businesses, science and social science domains. The process of data analysis is to obtain the raw data and convert it into information critical for decision making by users.



Initially raw data is collected from a variety of sources through interviews, downloads from online sources, reading documents, sensors, cameras, videos, satellites, recording devices, etc. On the available data,

Data preprocessing is to be performed so as to organize the available data into a structured format. After preprocessing the data into the required format, data cleansing must be done so as to further prepare it for data analysis. After data cleaning there are several methods and algorithms available which can be applied on to the data. R programming packages can be used for data analysis and visualization

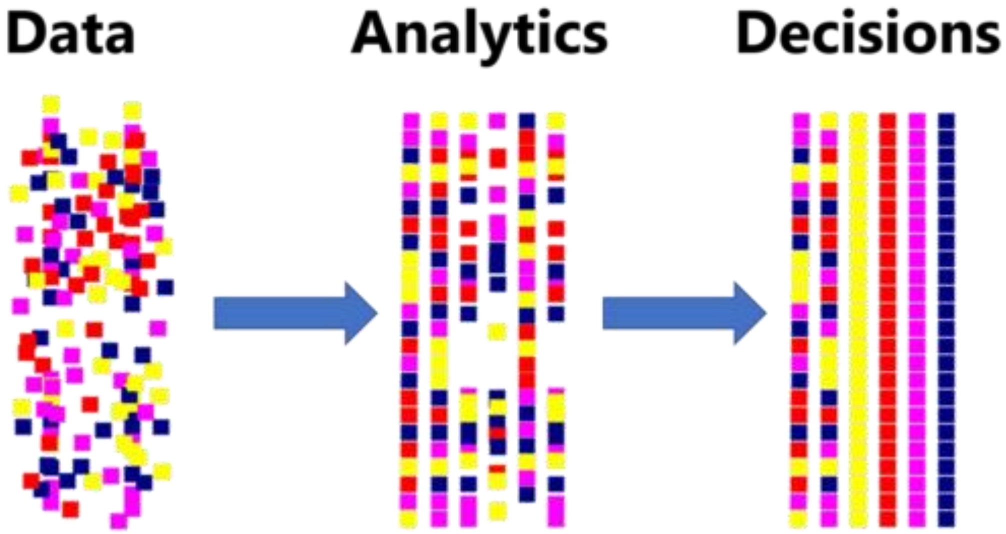
Need of Data Analytics


Cost reduction
Faster, better decision marking
DATA ANALYTICS
Better marketing & product analysis
Organization analysis

Data mining is like applying techniques to mold data to suit our requirement. Data mining is needed because different sources like social media, transactions, public data, enterprises data etc. generates data of increasing volume, and it is important to handle and analyze such a big data. It won't be wrong to say that social media is something we live by. In the 21st century social media has been the game changer, be it advertising, politics or globalization, it has been estimated that data is increasing faster than before and by the year 2020; about 1.7 megabytes of additional data will be generated each instant for each person on the earth. It is clear from the fact that the **number of internet users is now grown from millions to billions.**

Business Analytics:

The a process of examining large sets of data and achieving hidden patterns, correlations and other insights for achieving business goals



Processing large sets of data, either **descriptive analytics** to understand the inter relations or predictive analytics to discover new patterns of the current trends in the market. These data patterns are converted into actionable knowledge that can be used for decision making.

Big data analytics can support companies to better understand the knowledge and information contained within the data and will also help identify the key data that is most vital to the business and decisions making – both present and future.

In order to review and analyze such a vast volume of data, big data analytics is performed using dedicated software tools for predictive analytics, data mining and data optimization. Big Data tools like Clojure, Scala, Python, Hadoop and Java for NLP and Text Mining and R, MAT can be used data analytics.

3.2 About Twitter

Twitter is an online social media network site for news and other social networking service. The users post and interacts in twitter with messages known as "tweets". Tweets were originally restricted to 140 characters, but in 2017 this limit was doubled for all languages except Japanese, Korean, and Chinese. Only registered users can post tweets. Those who are not registered on Twitter can only read the tweets. Users access Twitter through its mobile app, website interface and also through Short Message Service (SMS).

Due to the nature of this microblogging service i.e. quick and short messages, people use acronyms, make spelling mistakes, use emoticons and other characters that express special meanings. Following is a brief terminology associated with tweets.

Emoticons: These are facial expressions that use punctuation and letters; they are used to express the user's emotions and mood concisely.

Target: A username is how you're identified on Twitter, and is always preceded immediately by the "@" symbol. Referring to other users in this manner automatically sends an alert to them.

Hashtags: any word or phrase immediately preceded by the # symbol. Users usually use hashtags to mark topics. This is primarily done to increase the visibility of their tweets.

Retweet (RT): A Tweet that is forwarded to the followers is known as a Retweet. It is mostly used to pass along news update or other discoveries on Twitter. Retweets always retain original attribution.

Modified Tweet (MT): Similar to Retweet, an abbreviation for "Modified Tweet." Placed before the Retweeted text when users manually retweet a message with modifications.

Twitter being a very prominent source of data generation leads the way. Twitter data in particular is very interesting because these tweets have a very important feature not provided by other networks. They happen at the "speed of thought" and the recent data is available to users to download and analyze in near real time. These tweets can be mined and analyzed and the results are of immense importance to various organizations. They can predict the outcome of elections, they can tell us the sentiments of people towards a product or service, they are highly valuable to business who wants to drive customers and extract profits.

3.3 About Amazon.com

Amazon.com is an ecommerce and cloud computing company. It is the largest Online retailer in the world as measured by revenue, and second largest after Alibaba Group basis on total sales. The amazon.com commenced as an online book store which later stretched to sell video/MP3 downloads and streaming, software, games, apparels, electronics, furniture, cosmetics, medicines, food, toys, jewelry etc.

Amazon.com has grown rapidly and been a microcosm for user-supplied reviews. Amazon.com has opened view access to its reviews to all the consumers, and allowed the consumers to post their review for any one of the millions of products on their website.

With this increase in anonymous human-generated content, there have been increasing trend and efforts to understand the information in the meaningful and accurate context. Hence, methods are being developed to determine the intent of the author. Understanding what online users think of its content can help a company market its product as well as mange its online reputation.

Amazon is one of leading e-commerce companies which possess and analyze these customers’ data to advance their service and revenue.

4

Research Objectives & Methodology

4.1 Motivation

This project research was motivated by my wish to explore the **sentiment analysis field** of machine learning since it allows approaching NLP (natural language processing) which is a very interesting and latest topic at present. Following my previous experience of learning R language and applying it in marketing analytics field, I applied the same technology with tweets and Amazon reviews to figure out which tweet or review is positive, negative or neutral.

4.2 Scope of the Research

The study aims to understand the effectiveness and **usage of R programming in developing Sentiment Analysis applications** for analyzing customer reviews, opinions and ratings about a particular product, service, brand, government or private organizations etc. and calculate the sentiments of people in order to evaluate whether the overall opinions are positive, negative or neutral towards that particular object.

For the scope of this project, the **user reviews and comments are retrieved from Twitter and Amazon** to calculate the sentiments. However, this application can be extended for usage across any other social media, ecommerce or microblogging sites to extract the opinions.

4.3 Research Problem

This study focuses on following problems –

1. Consumers want to take an informed decision of buying a certain product or service.

2. The Customer Reviews on the Internet has risen exponentially over the last decade and is an important resource for buying products or attending events. The user would like to see what others are saying about them.

3. The Companies want to make decisions about improvements to their products or services by analyzing customer satisfaction based on their feedback and opinions towards their products or services

4. The Political parties are interested to know if people support their programs or not.

5. The prediction analysis of events like elections using customers' positive or negative opinions on Social Media platforms towards a particular party.

6. The marketers and purchase departments need to review users' positive or negative opinions on its campaigning, advertising or product launching programs.

7. Understanding what online users think of its content can help a company market its product as well as mange its online reputation.

8. Twitter is one of the most useful interfaces to track sensitiveness, opinions, attitudes, emotions and feelings on the web for sentiment analysis, especially for tacking products, services, organization, brands or even people. The need is to develop a robust application to utilize twitter as a resource to measure customer sentiments.

4.4 Research Objectives

The Research objectives are as follows:

1. To demonstrate how **social media data can be a rich source of information** which, when harnessed by the marketers, can give the organisations an upper edge over its competitors.

2. To review the **effectiveness of R programming** for data extraction, data mining and analysis of social media data to evaluate customers sentiments towards an object (event, program, product, service, brand, organization etc.)

3. To **convert descriptive customer reviews into numeric sentiment scores** so that we can view and compare the customers' attitudes directly.

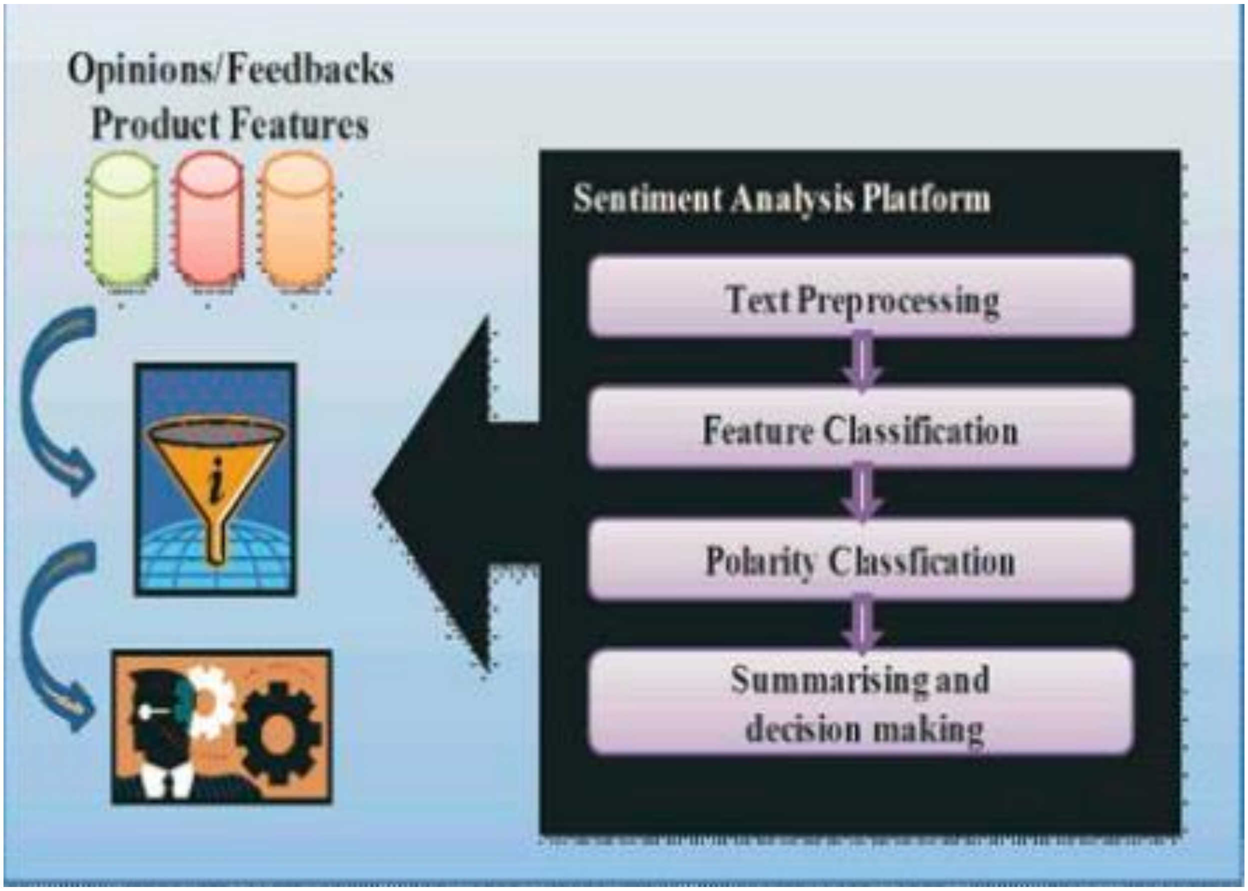
4. To use both statistical and linguistic approach to **categorize sentiments as positive and negative.**

5. To understand the approach of **creating a general sentiment analyzer solution** that allows users to intervene with the system. Hence, the system does not only analyze opinions at a specific type of product.

4.5 Research Methodology

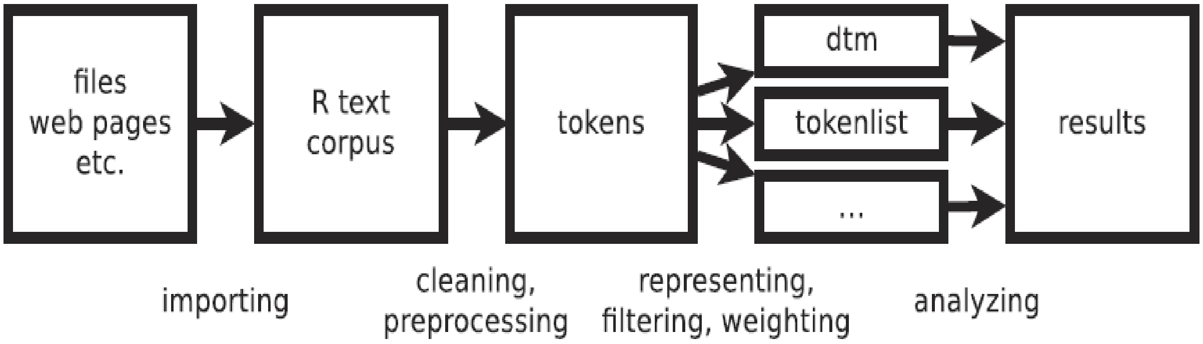
The **Lexicon based approach using R Programming** is used for this research.

Lexicon based approaches are quite popular in the sentiment analysis domain. These approaches involve tokenization of a particular quantity of text into unigrams, which are then assigned a polarity score. The aggregated sum of these scores is used to quantify the total sentiments around that particular text. It is usually categorized as positive, negative or neutral based on the calculated score.



The different phases involved in this approach are:

1. **Data Collection**
2. **Data Preparation**
3. **Data Loading**
4. **Data Analysis and Development of Statistics Dashboard**
5. **Documentation**



4.5.1 Data Collection

For analyzing of data or to perform text mining we have to first conduct the construction for a suitable dataset. The datasets can be retrieved using the existing API of the portal or by building a custom API.

Twitter API

It is an interface between the twitter application created by the user and the RStudio. The integration is performed by using R packages and function used for extracting tweets from twitter and loading them into RStudio.

The Search API allows developers to look up tweets containing a specific word or a phrase. The Search API of Twitter has a constraint that it could produce only 1500 tweets at one time. The **“twitteR” package** in R is used to extract the tweets from Twitter.

Amazon Scripts (Perl/Python)

Even though **Amazon** does not have Twitter-like API available to download reviews with, it has links attached to every review on all posted products, so one can technically navigate the site via product IDs.

There is couple of **Perl scripts** created by Andrea Esuli available on the web to obtain the reviews for the listed products. The primary script downloads the entire HTML page for the chosen product and the following script searches the file for information about the review, such as the product-id, product rating, review date and time, and review text. It can be integrated with RStudio by creating custom packages and function in R which is used for loading reviews from dataset file into RStudio.

The first step of any R text analytics project is to retrieve and upload the text in R console or R studio. The textual data can be stored in a wide variety of file formats such as CSV and TXT, but advanced packages are required for processing specific formatted text files such as JSON, HTML, and XML, and also for reading complex formats such as Word, Excel and PDF. The challenge is to work with these different packages and their different interfaces when different file formats are used together in one project.

A better and easy solution for this problem is using the **readtext package** that wraps various packages together to offer just one function for importing different types of data in an identical format.

4.5.2 Data Preparation

The complete texts must be tokenized into smaller text features, such as words or combination of words for performing text analysis methods. Also, the computational performance and correctness of multiple text analysis techniques can be improved by applying normalizing features, by eliminating "**stopwords**" i.e discarding the words that are designated as of no interest, and which are therefore discarded prior to analysis. Taken together, these preparatory steps are commonly referred to as "**preprocessing**".

Here we first discuss several of the most common preprocessing techniques, and show how to perform each technique.

1. Transformation/Cleansing

Data Transformation / Cleansing are an important component of the data mining process. It involves identification, removal of faults, discrepancies and inconsistency to enhance the quality of the dataset before initiating the analysis process. The tweets and comments are cleaned from irrelevant data to improve their quality. The certain elements that do not provide any information have to be removed before processing. The examples of such elements in tweets are as follows:

a) **Links:** People generally have a tendency to attach documents (images, blogs, videos, web direction etc.) along with their tweets. These links or URLs have to be eliminated since they are of no use to our analysis.

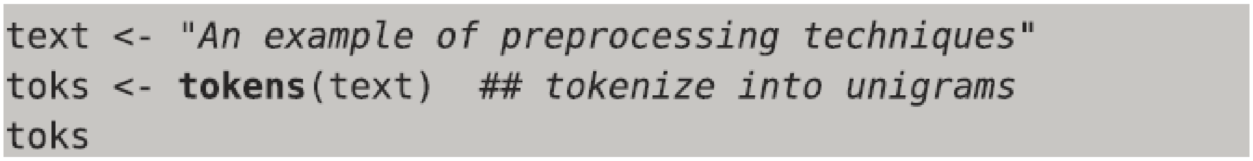
b) **Mentions:** Mentions are used in Twitter to reply, acknowledge or start a conversation. Mentions are always written using **"@"** sign followed by the username. These mentions do not contain any relevant information thus, are removed.

c) **Punctuations and other miscellaneous data:** Punctuations marks like quotes (""), commas (,) and semicolons (:) do not have any significant role in the analysis and hence, are removed from all the tweets present in the dataset.

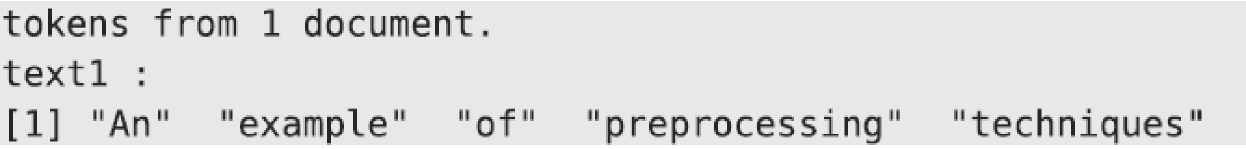
2. Tokenization

Tokenization is the process of splitting a text into tokens. This is crucial for computational text analysis, because full texts are too specific to perform any meaningful computations with. Mostly tokens are words, as these are the most common semantically significant components of texts.

For many languages, splitting texts by words can mostly be done with string processing at low level due to clear indicators of word boundaries, such as white spaces, colons, dots and commas. However, a good tokenizer must also be able to handle certain exceptions, such as the period in the title “Er”, which can be confused for a sentence border. Moreover, tokenization is more complex for languages where words are not clearly parted by white spaces, such as Chinese and Japanese. To deal with these cases, some tokenizers include dictionaries of patterns for splitting texts. The string package in R is frequently used for sentence/word disambiguation, for which it controls dictionaries. The package called **tokenizers** is used for this purpose.



```
text <- "An example of preprocessing techniques"
toks <- tokens(text)  ## tokenize into unigrams
toks
```

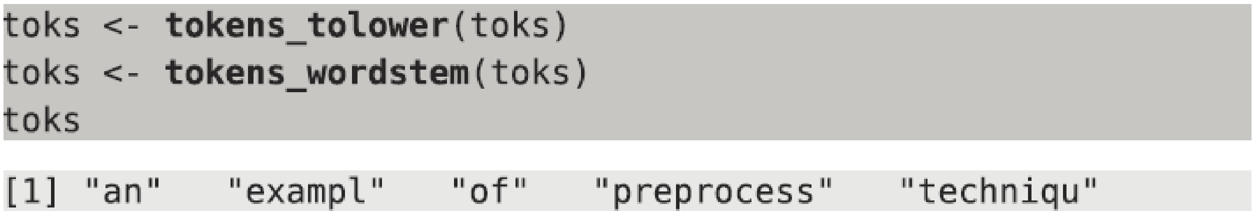


```
tokens from 1 document.
text1 :
[1] "An"  "example"  "of"  "preprocessing"  "techniques"
```

3. Normalization: Lowercasing and stemming

This process mostly refers to the transformation of words into a more constant form. This can be important if for a specific analysis a computer has to recognize when two words have approximately the same meaning, even if they are written slightly differently. Another benefit is that it reduces the size of the vocabulary. A simple but important normalization technique is to convert all text lower case. If we do not perform this transformation, then a computer will not recognize that two words are alike if one of them was capitalized because it occurred at the start of a sentence.

In R, the package is used in many text analysis packages to implement stemming. Lowercase conversion and character stemming or tokens can be performed with the *_tolower / *_wordstem functions, such as char_tolower to convert char objects to lower case, or tokens_wordstem to stem tokens.

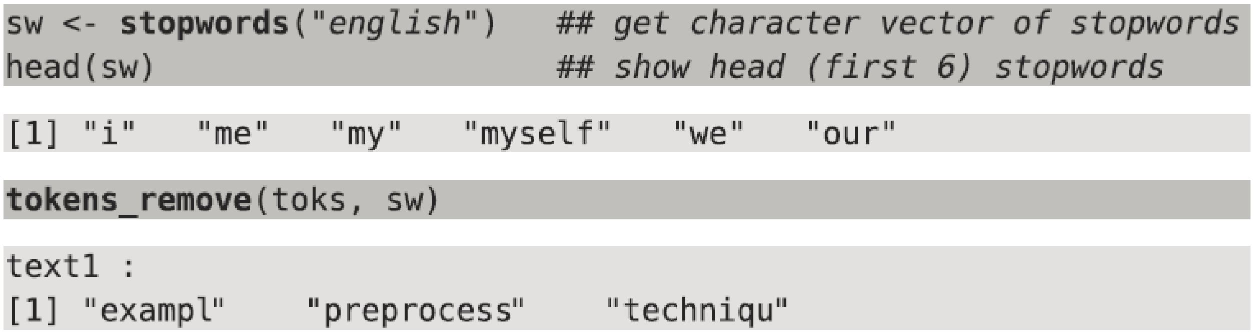


```
toks <- tokens_tolower(toks)
toks <- tokens_wordstem(toks)
toks
```

```
[1] "an"   "exampl"   "of"   "preprocess"   "techniqu"
```

4. Removing Stopwords

Common words such as “an”,”the”,”a” in the English language are seldom informative about the content of a text. These words needs to be filtered out for the benefit of reducing the data size, reducing computational load, and also improving accuracy. To remove these words in advance, they are matched to predefined lists of “stopwords” and removed. Several R packages of text analysis provide lists of stopword for multiple languages manually filter out stopwords.



```
sw <- stopwords("english")   ## get character vector of stopwords
head(sw)                     ## show head (first 6) stopwords
```

```
[1] "i"   "me"   "my"   "myself"   "we"   "our"
```

```
tokens_remove(toks, sw)
```

```
text1 :
[1] "exampl"    "preprocess"    "techniqu"
```

4.5.3 Data Loading

The resultant .csv file is then loaded into R-Studio which is further used for the analysis.

4.5.4 Data Analysis and Statistics Dashboard

There are different types of statistics that can be used to define, discover and analyze a text body. An example of a popular technique is to sort the information value of words inside a corpus and then visualize the most informative words as a word cloud to get a quick indication of what a text corpus is about. Text statistics such as average word and sentence length, word and syllable counts are also commonly used as an operationalization of concepts such as readability.

1 Creating "Bag of Words"

The simplest and most widely used lexicon based approach is the baseline approach (also called "**Bag of Words Approach**"). In this method, there are two dictionaries – that of the positively tagged words and negatively tagged words. After tokenization, each individual word of the tweet is searched within those dictionaries, and depending upon the location of the word, it is assigned a polarity score.

a) **Scoring**: If the individual token is found in the positive words dictionary, it is assigned a +1 polarity score value, if present in the negative words dictionary, a score of -1 and lastly, if not present in any of them, a score of 0 is assigned.

b) **Aggregation**: The total sum of the scores of each word present in the text is calculated and the on the basis of the final polarity value, the tweet can be categorized as positive, neutral or negative.

Consider a tweet from our dataset:

"Excellent things can be accomplished with ease when you get the best team in the world".

Following the technique explained above, each of the following words**-** "**excellent**", "**accomplished**", "**ease**" and "**best**" are given a sentiment score of +1 since they are present in the positive words dictionary. On aggregation, the total polarity score of +4 is obtained, indicating that the sentiment behind the tweet is positive.

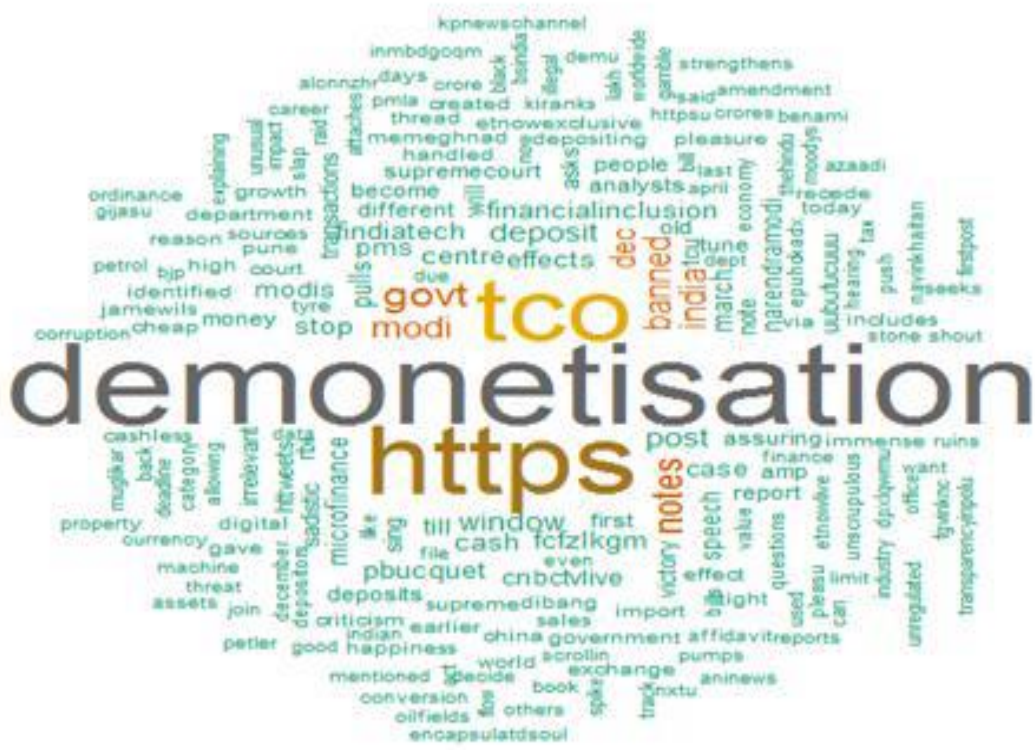
2 Creating “Word Cloud”

A wordcloud is a visual representation of frequently occured words in a collection of text files. The altitude and length of each word in this picture gives an indication of frequency of word occurrence in the entire text.

The context in which the word was used and the frequency of the occurrence can be analyzed which will help to analyze the exact sentiment of the people.

Example:

Word Cloud on tweets related on Demonetisation:



3 Creating Histogram/GGPlot/ Pie-Chart for the Sentiments

Histogram and GGPlot will help to make the comparative study of the sentiments on day to day basis. Pie-Chart provides us the exact percentage of people with positive or negative views.

Examples of Graphical representation in form of Histogram and Pie-Charts are shown below:

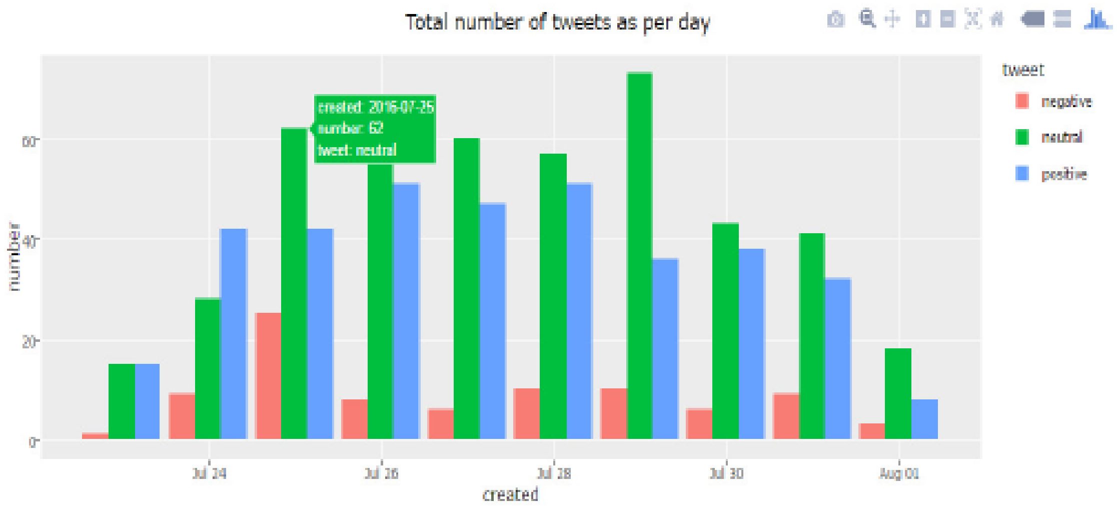
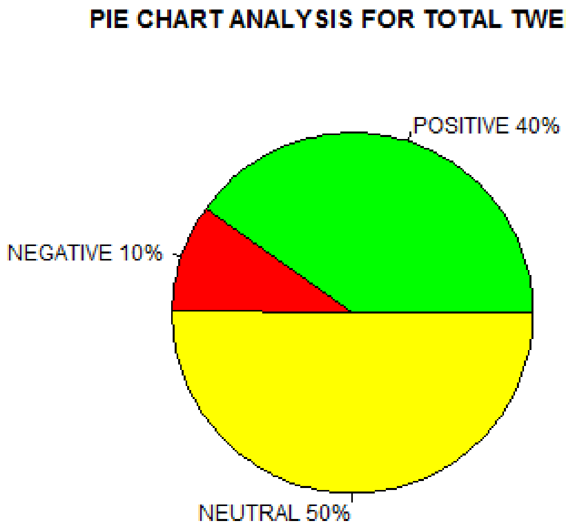


Fig. The histogram above showcases the distribution of tweets across Positive, Negative and Neural polarity per day.



5
Applications developed on Sentiment Analysis

As part of this project research, I have created below two applications to demonstrate the effectiveness of R programming in analyzing customer sentiments:

1. **Sentiment Analysis of text extracted from tweets made on “MAKE IN INDIA” campaign, a prestigious initiative by the Government of India.**

2. **Sentiment Analysis of Customers towards Baby products using the reviews comments posted on Amazon ecommerce portal.**

These applications are built for the purpose of analyzing data, although RStudio performs well with the statistical problems and proceeds with a user friendly interface.

These systems could be applied to compute the sentiment in any field where data analytics plays important role. The system will use R language to clean, extract, analyze and show desired results.

5.1 Make in India Campaign

The Make in India was an initiative launched by Prime Minister Mr. Narendra Modi in September 2014 in order to encourage the manufacturing sector in India and to transform India into a global design and manufacturing hub.

The hashtag **#MakeinIndia** has been trending over the past 2 years. The government, organisation and people have been tweeting on various announcement and opinions related to the #MakeInIndia campaign.

This initiative is to make India a Global Manufacturing Hub giving a call to business leaders, potential partners and investors throughout the world. It should be shown as a credible initiative to its upcoming depositors. There is visible momentum, energy and optimism. So to observe the moment of Make-In-India, I have targeted the information from social web sites like Twitter to do sentiment analysis to see that among the world population, how much positive and negative opinion is there on Make-In-India taking a sample of 200 current tweets about Make-In-India.

To know more about **Make in India** campaign, visit: http://www.makeinindia.com/home

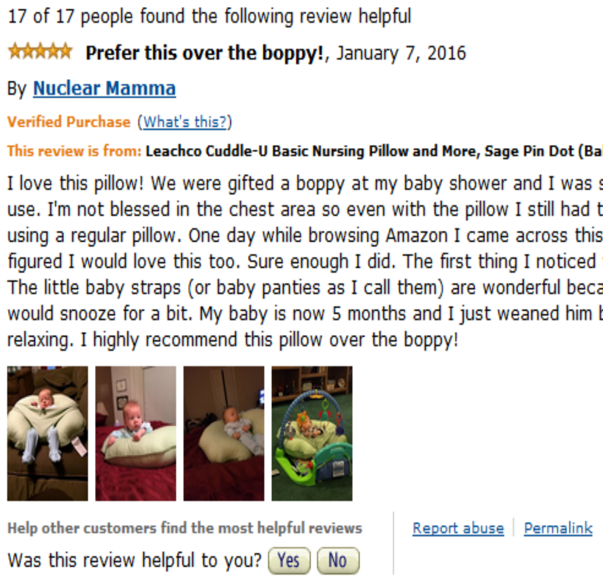
5.2 Amazon Customer Reviews of Baby Products

Amazon.com has grown tremendously and has been a microcosm for customer provided reviews. Amazon.com has opened view access to its reviews to all the consumers, and allowed the consumers to post their review for any one of the millions of products on their website.

With this increase in anonymous user originated data, manual and automated efforts must be put to convert that data into meaningful and accurate information and also develop methods to determine the real intent of the customer adding the review comments.

I have analyzed the sentiment analysis for **Baby Products** in Amazon. These products are critical for Parents and kids and there are around millions of review comments posted on different categories of baby products in Amazon.

Below is an example of format of a customer review in Amazon.



17 of 17 people found the following review helpful

★★★★★ **Prefer this over the boppy!**, January 7, 2016

By **Nuclear Mamma**

Verified Purchase (What's this?)

This review is from: **Leachco Cuddle-U Basic Nursing Pillow and More, Sage Pin Dot (Baby Product)**

I love this pillow! We were gifted a boppy at my baby shower and I was super excited to try it out. Once baby came I found it very awkward to use. I'm not blessed in the chest area so even with the pillow I still had to prop my baby up. Eventually I stopped using the boppy and started using a regular pillow. One day while browsing Amazon I came across this and decided to try it. I have leachco' pregnancy pillow and love it so I figured I would love this too. Sure enough I did. The first thing I noticed was its about twice as thick as my boppy original which is great for me! The little baby straps (or baby panties as I call them) are wonderful because after the baby would get boob drunk I would strap him in and he would snooze for a bit. My baby is now 5 months and I just weaned him but we are still using the pillow. It's great for tummy time and just plain ol' relaxing. I highly recommend this pillow over the boppy!

Help other customers find the most helpful reviews

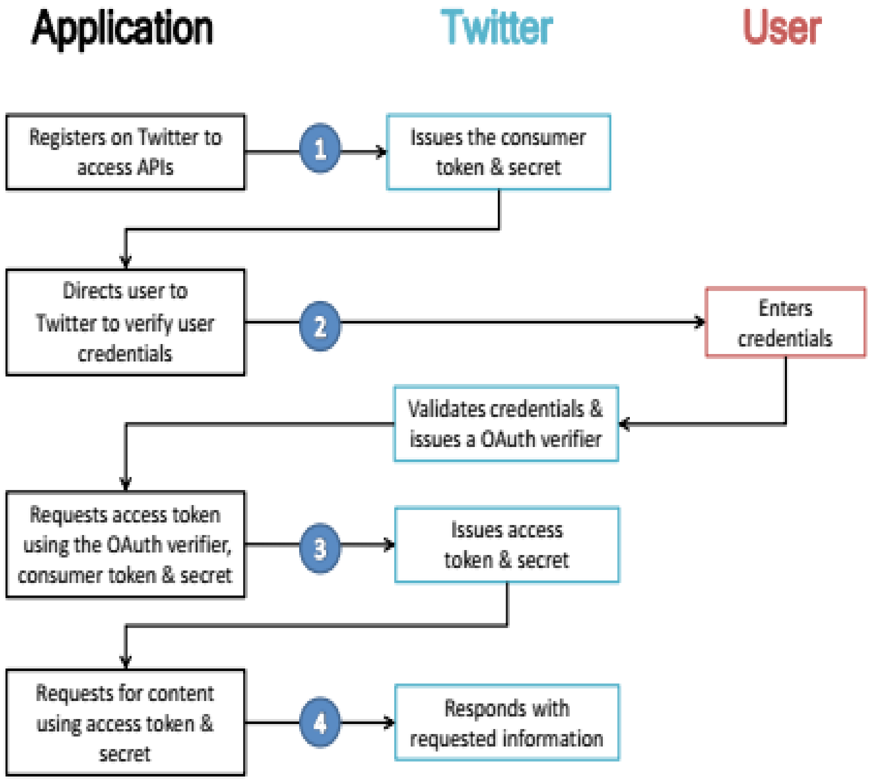
Was this review helpful to you?

6
Implementation

6.1 Twitter Analysis of Make in India

To observe the moment of Make-In-India, I have extracted the information in the form of tweets from Twitter to perform sentiment analysis to see that among the entire population, how much positive, neutral or negative opinion is there on Make-In-India taking a sample of current tweets about Make-In-India.

6.1.1 Twitter Architecture


Application
Twitter
User
Registers on Twitter to access APIs
1
Issues the consumer token & secret
Directs user to Twitter to verify user credentials
2
Enters credentials
Validates credentials & issues a OAuth verifier
Requests access token using the OAuth verifier, consumer token & secret
3
Issues access token & secret
Requests for content using access token & secret
4
Responds with requested information

Twitter Environment

Twitter API Authentication process is carried out using OAuth package of R. The above figure displays the steps involved in usage of OAuth to Access Twitter API. A Twitter application needs to be created to run Twitter API.

- Consumers need to register with Twitter. It continues in providing a key and secret key to consumer which can be used in the application to be authenticated.

- These keys are to be used to create a Twitter link through which Authentication process gets initiated. Verification of users identity is done by Twitter and issues PIN 3 called as verifier. The user needs to offer this pin to the application.

- Next application process uses this PIN to request for an Access Token and Access Secret, exclusive to the user from Twitter API.

- Token and secret key information are cached for further use. It can be accomplished through GetUserAccessKeySecret.

R-Studio

R-Studio is the environment developed for statistical analysis and a Graphical view of the large data sets. R-Studio is rich in packages, nearly 8000 packages are available. I have used R-Studio Interface in the implementation.

R-Packages

R-packages are a collection of R functions which is a compiled code on sample data. These functions are stored under the name of R-Library in its environment. During installation period, by default R installs a set of packages. Remaining packages need to be installed and loaded separately as and when they are required by the specific application.

The below given packages are used in the implementation:

- **twitteR:** The Twitter web API that provides an interface.

- **ROAuth**: It provides an interface to the OAuth 1.0 specification. This package allows users to connect the server of their choice and authenticate via OAuth package.

- **plyr**: It is a group of tools that resolves a common group of problems. We need to break a big problem down into number of manageable pieces, operate on them and then put all of them back together.

- **stringr:** It is a set of packages that make string functions more easier, simpler and reliable for usage.

- **ggplot2:** It is an implementation of the grammar of graphics in R. We can build up a plot step by step from numerous data sources.

- **RColorBrewer**: It provides palettes for coloring nice plots/charts shaded according to a variable.

- **tm**: it is a framework for text mining applications inside R.

- **wordcloud***:* It provides feature to create pretty viewing wordclouds while performing Text Mining.

- **sentiment***:* Used for sentiment analysis that includes classification of positivity, neutral and negativity classification.

6.1.2 Methodology

Following steps needs to be followed for performing twitter analysis.

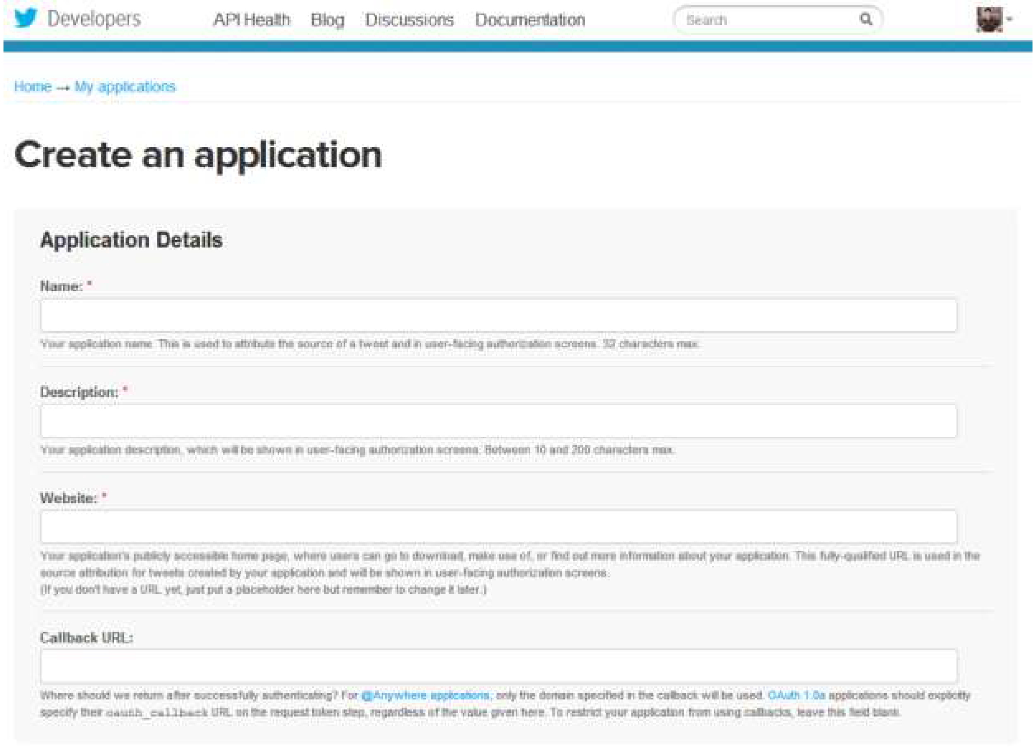
1 **Create a Twitter Application.**
2 **Execute Twitter API code through R-Studio.**
3 **Collect Twitter data archives.**
4 **Standardize the data.**
5 **Classify the Data with R Tool commands.**
6 **Run R commands for processing the tweets.**
7 **Establish R Plotter, Histogram and Wordcloud to view results.**
8 **Interpretation**

6.1.2.1 Create a Twitter Application

The prerequisite to execute Twitter Analysis is to create a twitter application. This application will permit you to perform analysis by linking your R console to the twitter using the Twitter API. The steps for creating your twitter applications are:

By using this link *https://dev.twitter.com* and login by using your twitter account.

Then go to **My Applications → Create a new application**.



Create an application

Application Details

Name: *

Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description: *

Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website: *

Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL:

Where should we return after successfully authenticating? For @Anywhere applications, only the domain specified in the callback will be used. OAuth 1.0a applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

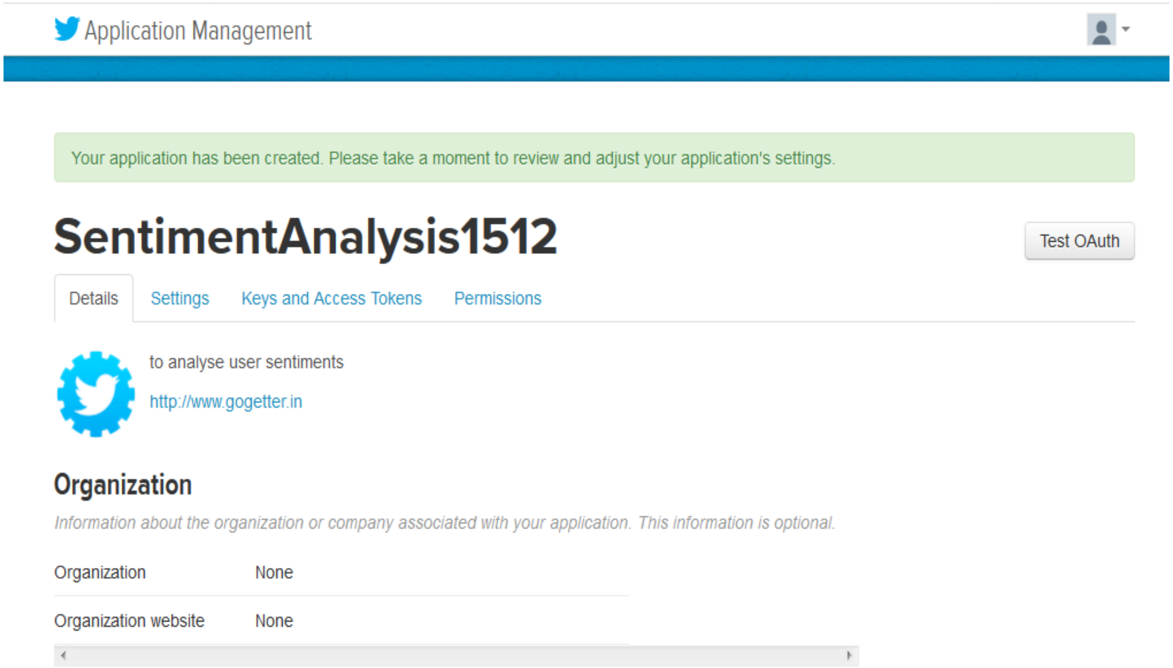
Steps to be followed:

1. First give your application a name, description about your application in limited words not more than i.e. (10).
2. Name your website's URL or your blog address (in case you don't have any website).
3. No need to fill the Callback URL leave blank for now.
4. Complete other regulations and create your twitter application.

Once, all the steps are done, the created application will show as below. Please note the **Consumer key** and **Consumer Secret number** as they will be used in RStudio later.

Consumer Key Example: "**Rnpxxxxxxxxxxxxxxxxxwl7s1oP**"

Consumer Secret Number Example: **"Sghxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx0ZEE"**



Your application has been created. Please take a moment to review and adjust your application's settings.

SentimentAnalysis1512

to analyse user sentiments

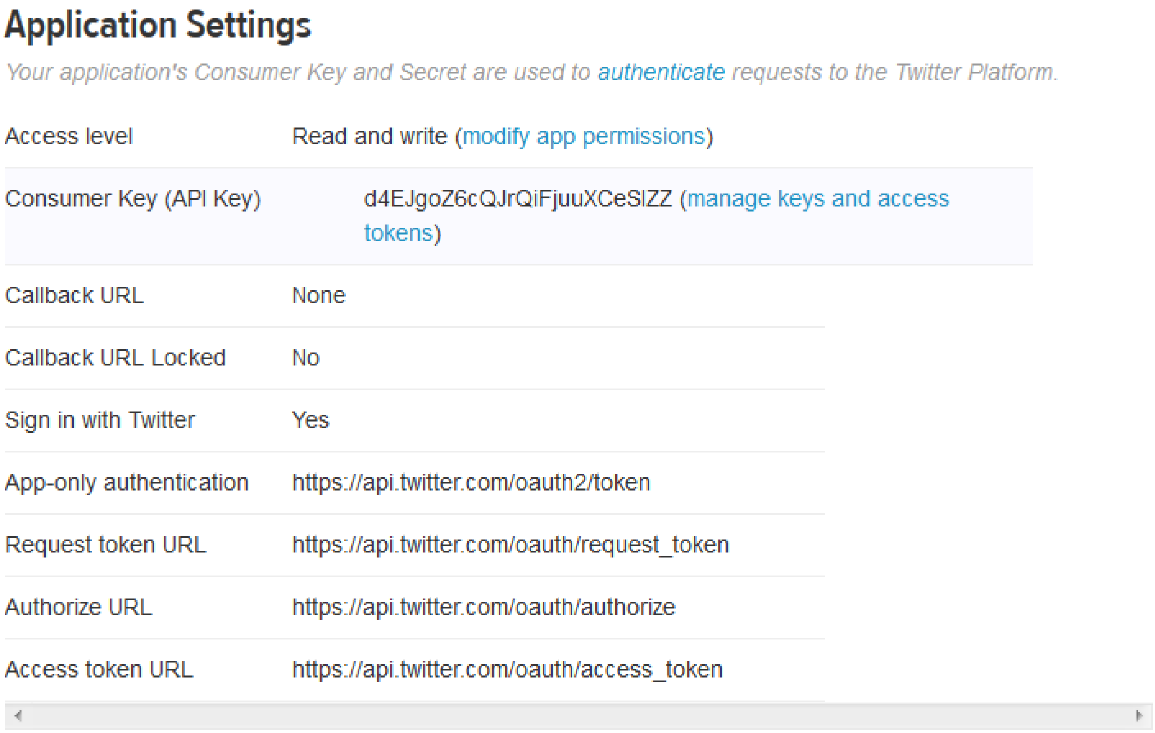
http://www.gogetter.in

Organization

Information about the organization or company associated with your application. This information is optional.

Organization	None
Organization website	None

Now open the app and notice your app will have following token that are to be used later step.



Application Settings

Your application's Consumer Key and Secret are used to authenticate requests to the Twitter Platform.

Access level	Read and write (modify app permissions)
Consumer Key (API Key)	d4EJgoZ6cQJrQiFjuuXCeSlZZ (manage keys and access tokens)
Callback URL	None
Callback URL Locked	No
Sign in with Twitter	Yes
App-only authentication	https://api.twitter.com/oauth2/token
Request token URL	https://api.twitter.com/oauth/request_token
Authorize URL	https://api.twitter.com/oauth/authorize
Access token URL	https://api.twitter.com/oauth/access_token

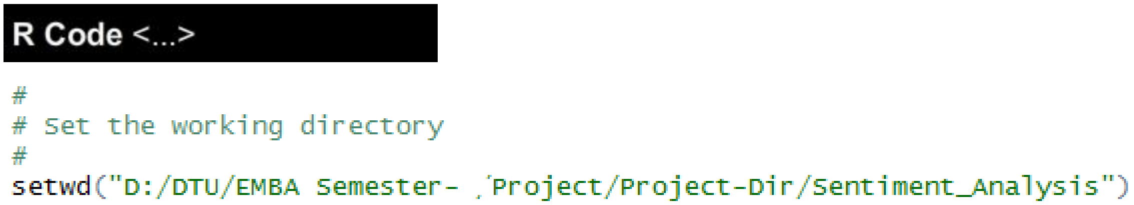
This step is completed. Now further processes will be performed in RStudio.

6.1.2.2 Execute Twitter API code through R-Studio

Twitter Search Application Program Interface code has to be executed from R Studio. To have the interface to Twitter tweets, a connection has to be established to twitter web site. Then we need to search for our tweets and save them to CSV file.

As part of Twitter API process there are many R packages that have to be installed first through install command of R. and imported to R through library command.

Step 1: Set up the working directory

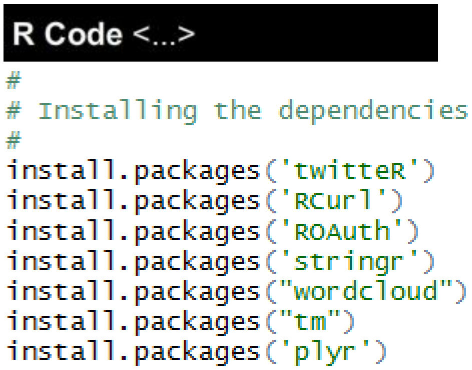


R Code <...>

```
#
# Set the working directory
#
setwd("D:/DTU/EMBA Semester- ,'Project/Project-Dir/Sentiment_Analysis")
```

This will set up the project's folder in your computer as the working directory. This will help RStudio to get and set the dependencies without explicitly specifying the folder's path all the time.

Step 2: Installing the packages

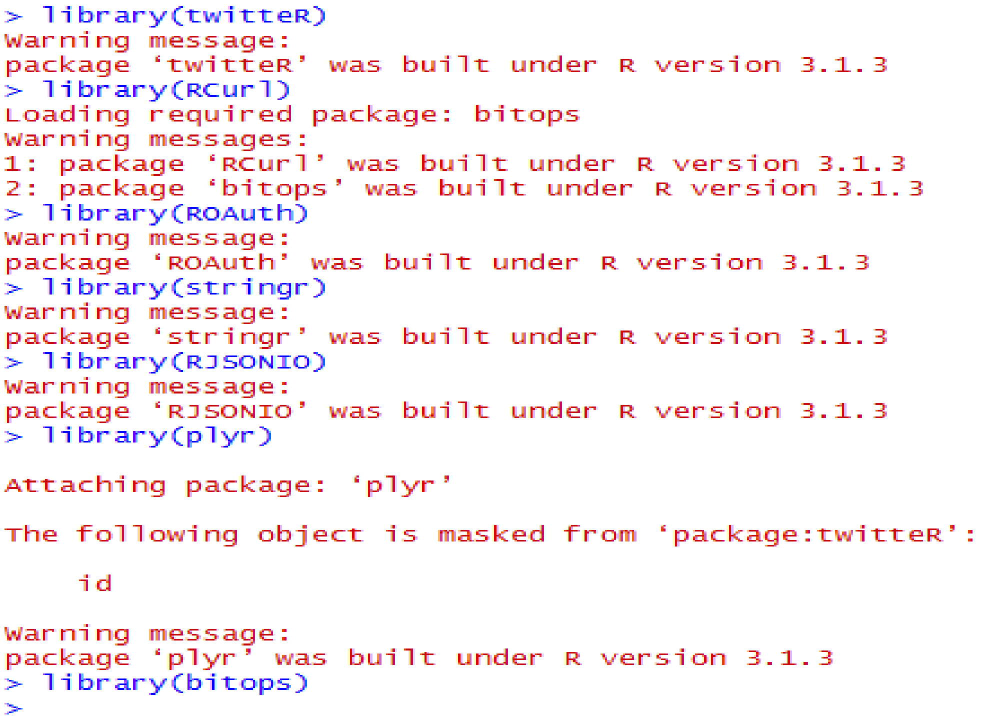


R Code <...>

```
#
# Installing the dependencies
#
install.packages('twitteR')
install.packages('RCurl')
install.packages('ROAuth')
install.packages('stringr')
install.packages("wordcloud")
install.packages("tm")
install.packages('plyr')
```

After executing the script section, console will look like this shown below:

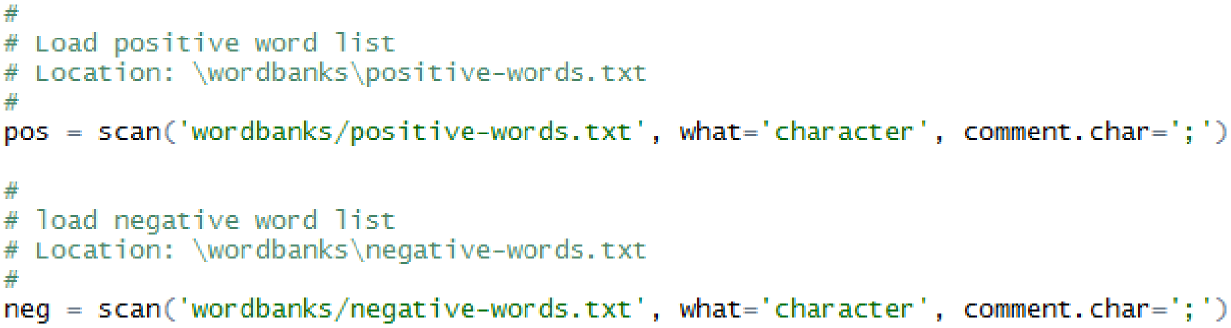
OUTPUT:



```
> library(twitteR)
Warning message:
package ‘twitteR’ was built under R version 3.1.3
> library(RCurl)
Loading required package: bitops
Warning messages:
1: package ‘RCurl’ was built under R version 3.1.3
2: package ‘bitops’ was built under R version 3.1.3
> library(ROAuth)
Warning message:
package ‘ROAuth’ was built under R version 3.1.3
> library(stringr)
Warning message:
package ‘stringr’ was built under R version 3.1.3
> library(RJSONIO)
Warning message:
package ‘RJSONIO’ was built under R version 3.1.3
> library(plyr)

Attaching package: ‘plyr’

The following object is masked from ‘package:twitteR’:

    id

Warning message:
package ‘plyr’ was built under R version 3.1.3
> library(bitops)
>
```

Step 3: Loading the positive and the negative word list

R Code <...>

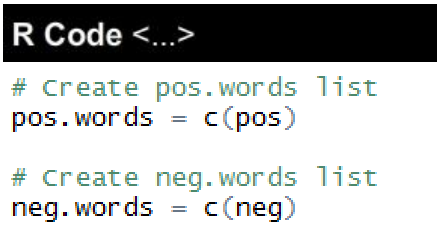


```
#
# Load positive word list
# Location: \wordbanks\positive-words.txt
#
pos = scan('wordbanks/positive-words.txt', what='character', comment.char=';')

#
# load negative word list
# Location: \wordbanks\negative-words.txt
#
neg = scan('wordbanks/negative-words.txt', what='character', comment.char=';')
```

We are using scan () method to read the .txt files.

We are reading the positive-words.txt from the specified location and similarly reading the negative-words.txt.

Step 4: Creating word list

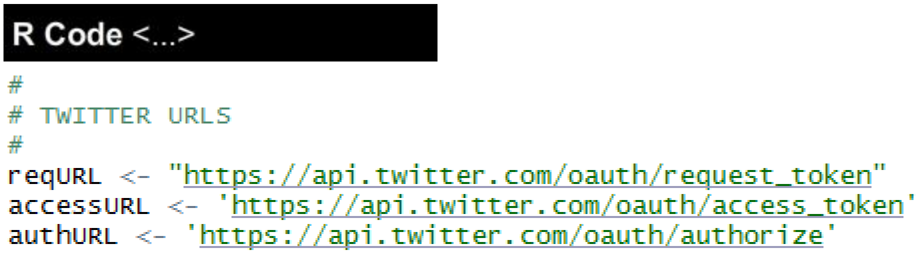


R Code <...>

```
# Create pos.words list
pos.words = c(pos)

# Create neg.words list
neg.words = c(neg)
```

In this step we are creating the word list using c () method.

Step 5: Setting up the Twitter URLs

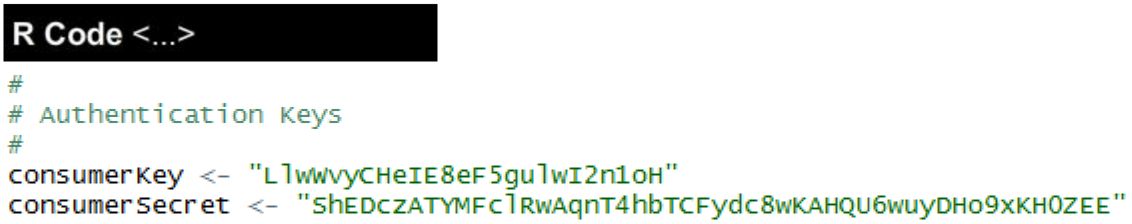


R Code <...>

```
#
# TWITTER URLS
#
reqURL <- "https://api.twitter.com/oauth/request_token"
accessURL <- 'https://api.twitter.com/oauth/access_token'
authURL <- 'https://api.twitter.com/oauth/authorize'
```

In this step we are setting the required URLs to hit while doing authentication.

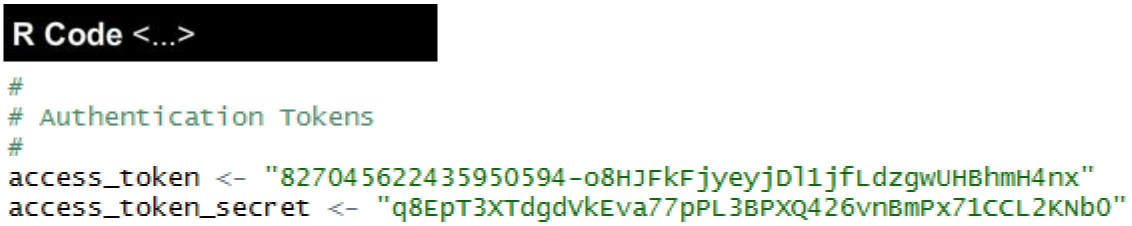
Step 6: Setting up the Twitter Authentication Keys



R Code <...>

```
#
# Authentication Keys
#
consumerKey <- "LlwWvyCHeIE8eF5gulwI2n1oH"
consumerSecret <- "ShEDczATYMFclRwAqnT4hbTCFydc8wKAHQU6wuyDHo9xKHOZEE"
```

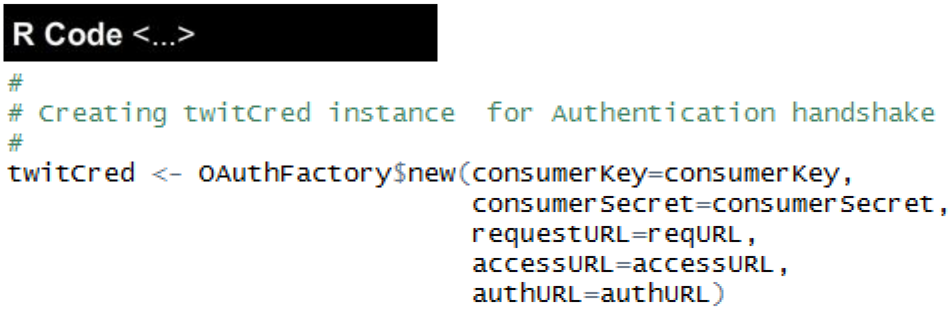
Step 7: Setting up the Twitter Authentication Tokens



R Code <...>

```
#
# Authentication Tokens
#
access_token <- "827045622435950594-o8HJFkFjyeyjDl1jfLdzgwUHBhmH4nx"
access_token_secret <- "q8EpT3XTdgdVkEva77pPL3BPXQ426vnBmPx71CCL2KNbO"
```

Step 8: Creating twitCred instance for Authentication handshake

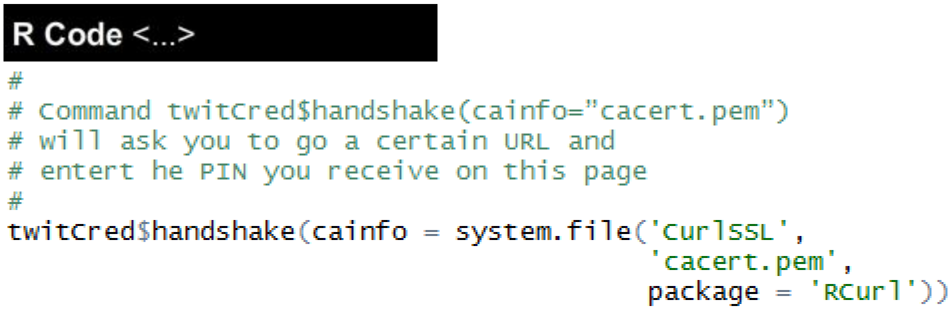


R Code <...>

```
#
# Creating twitCred instance  for Authentication handshake
#
twitCred <- OAuthFactory$new(consumerKey=consumerKey,
                             consumerSecret=consumerSecret,
                             requestURL=reqURL,
                             accessURL=accessURL,
                             authURL=authURL)
```

We are using OAuthFactory module to create a new instance named twitCred containing all the required information for authentication.

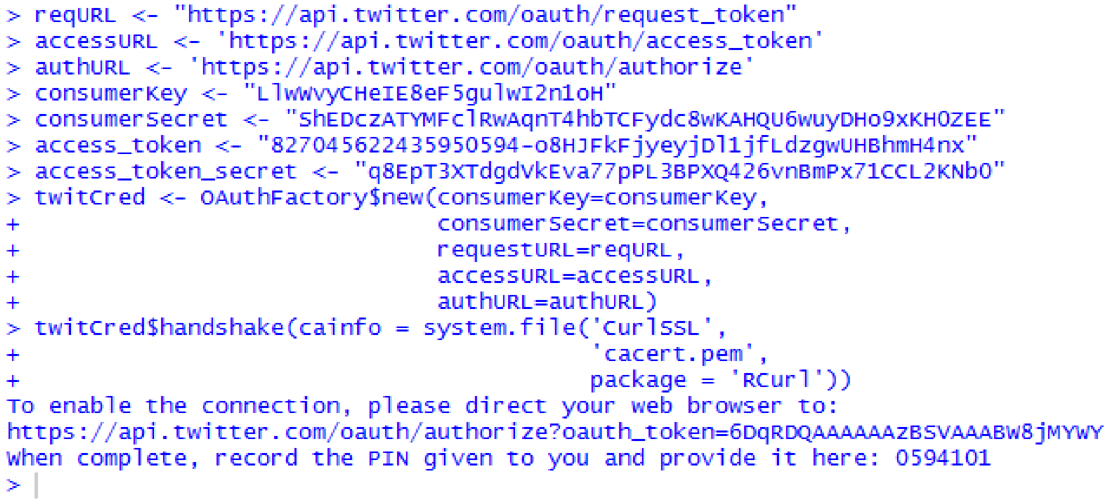
Step 9: Twitter Authentication handshaking



R Code <...>

```
#
# Command twitCred$handshake(cainfo="cacert.pem")
# will ask you to go a certain URL and
# entert he PIN you receive on this page
#
twitCred$handshake(cainfo = system.file('CurlSSL',
                                        'cacert.pem',
                                        package = 'RCurl'))
```

OUTPUT:

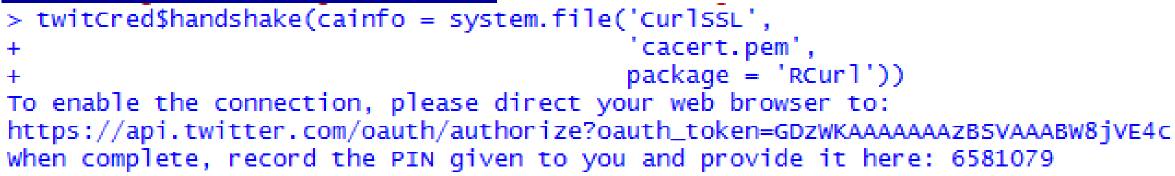


```
> reqURL <- "https://api.twitter.com/oauth/request_token"
> accessURL <- 'https://api.twitter.com/oauth/access_token'
> authURL <- 'https://api.twitter.com/oauth/authorize'
> consumerKey <- "LlwWvyCHeIE8eF5gulwI2n1oH"
> consumerSecret <- "ShEDczATYMFclRwAqnT4hbTCFydc8wKAHQU6wuyDHo9xKHOZEE"
> access_token <- "827045622435950594-o8HJFkFjyeyjDl1jfLdzgwUHBhmH4nx"
> access_token_secret <- "q8EpT3XTdgdVkEva77pPL3BPXQ426vnBmPx71CCL2KNb0"
> twitCred <- OAuthFactory$new(consumerKey=consumerKey,
+                              consumerSecret=consumerSecret,
+                              requestURL=reqURL,
+                              accessURL=accessURL,
+                              authURL=authURL)
> twitCred$handshake(cainfo = system.file('CurlSSL',
+                                         'cacert.pem',
+                                         package = 'RCurl'))
To enable the connection, please direct your web browser to:
https://api.twitter.com/oauth/authorize?oauth_token=6DqRDQAAAAAAzBSVAAABW8jMYWY
When complete, record the PIN given to you and provide it here: 0594101
> 
```

Twitter Authentication Pages:
https://api.twitter.com/oauth/authorize?oauth_token=GDzWKAAAAAAAzBSVAAABW8jVE4c

We need to copy the number appearing in the authentication page and enter this number in RStudio for completing the authentication process.

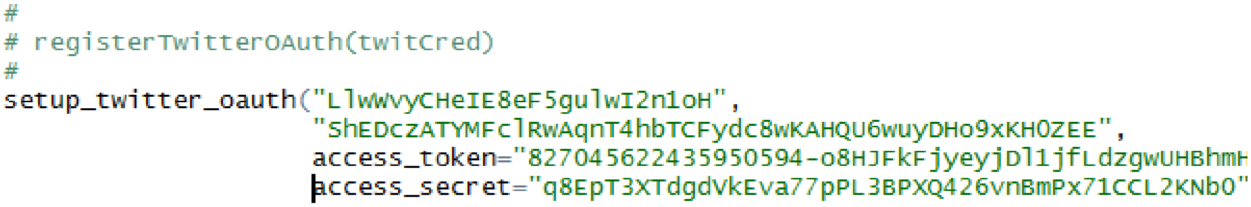
OUTPUT:



```
> twitCred$handshake(cainfo = system.file('CurlSSL',
+                                         'cacert.pem',
+                                         package = 'RCurl'))
To enable the connection, please direct your web browser to:
https://api.twitter.com/oauth/authorize?oauth_token=GDzWKAAAAAAAzBSVAAABW8jVE4c
When complete, record the PIN given to you and provide it here: 6581079
```

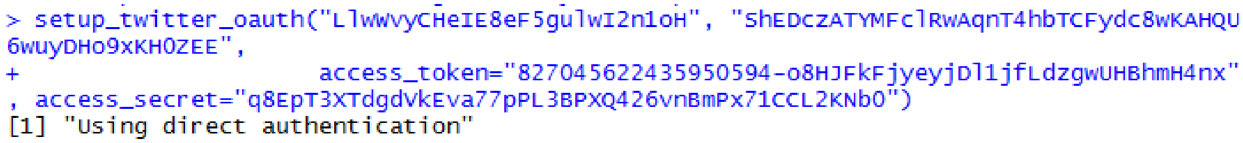
Step 10: Register Twitter Oath for communication

R Code <...>



```
#
# registerTwitterOAuth(twitCred)
#
setup_twitter_oauth("LlwWvyCHeIE8eF5gulwI2n1oH",
                    "ShEDczATYMFclRwAqnT4hbTCFydc8wKAHQU6wuyDHo9xKH0ZEE",
                    access_token="827045622435950594-o8HJFkFjyeyjDl1jfLdzgwUHBhmH
                    access_secret="q8EpT3XTdgdVkEva77pPL3BPXQ426vnBmPx71CCL2KNb0"
```

OUTPUT:



```
> setup_twitter_oauth("LlwWvyCHeIE8eF5gulwI2n1oH", "ShEDczATYMFclRwAqnT4hbTCFydc8wKAHQU
6wuyDHo9xKH0ZEE",
+                    access_token="827045622435950594-o8HJFkFjyeyjDl1jfLdzgwUHBhmH4nx"
, access_secret="q8EpT3XTdgdVkEva77pPL3BPXQ426vnBmPx71CCL2KNb0")
[1] "Using direct authentication"
```

6.1.2.3 Collect Twitter Data Archives

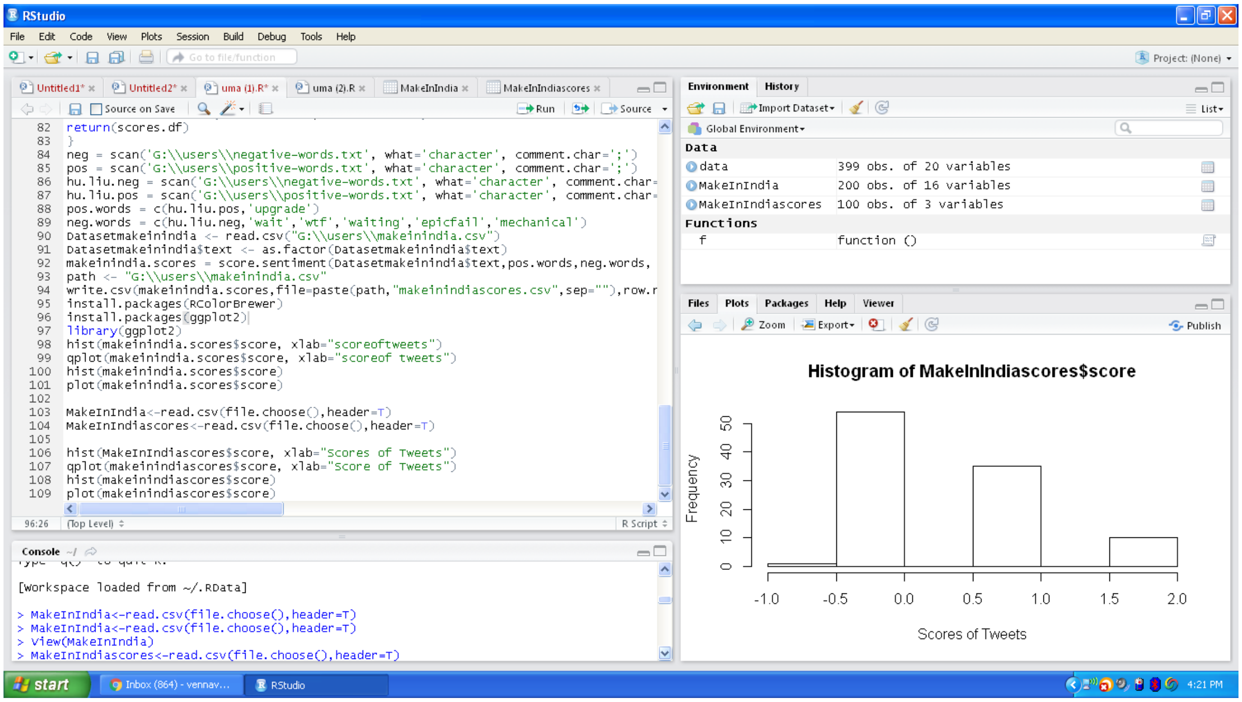
In this module, we retrieve the tweets associated to any keyword of the area. The Search Twitter function is used to extend the final phase of downloading tweets from the timeline. Now this list of tweets is converted into data frame (.df). The .df data frame is converted into .csv format file.

R Code <...>

```
#

makeinindia.list <- searchTwitter("#makeinindia", n=200,lang="en")
```

The above command returns the tweets from source for the last one week data about the product i.e Make-In- India (#makeinindia) where language is English and expecting 200 tweets in the command.

Execution of code in R-Studio

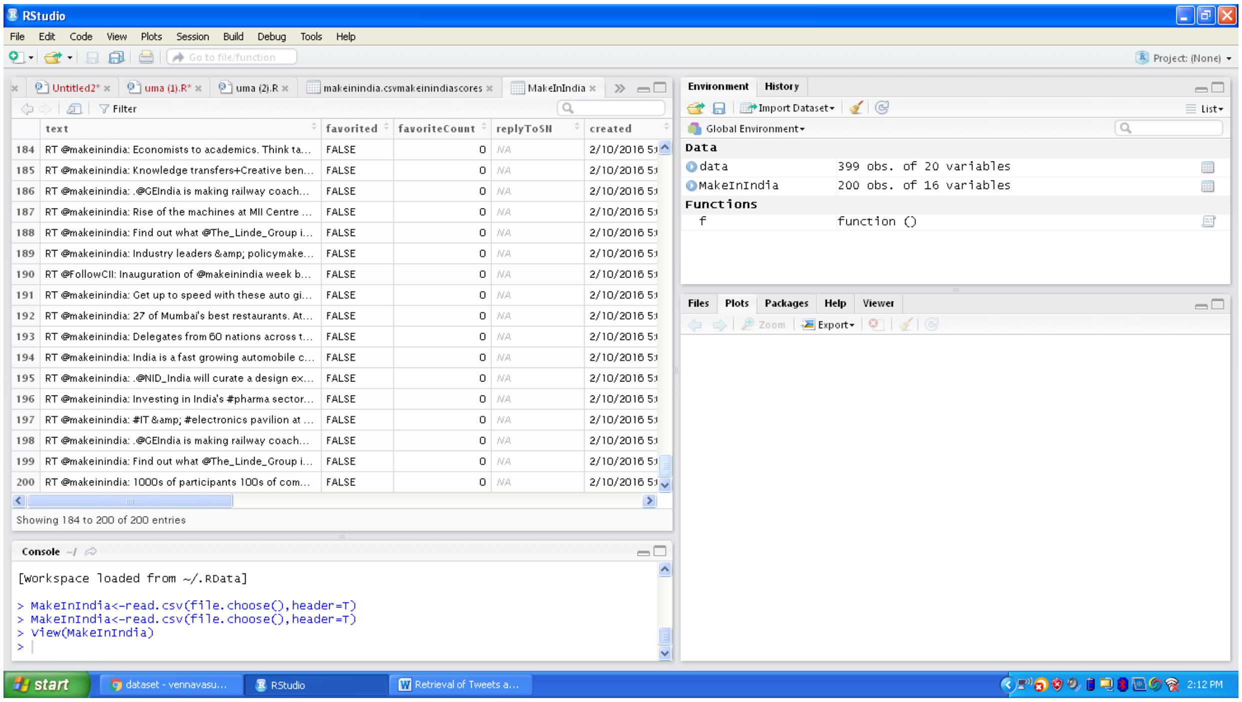


```
return(scores.df)
}
neg = scan('G:\\users\\negative-words.txt', what='character', comment.char=';')
pos = scan('G:\\users\\positive-words.txt', what='character', comment.char=';')
hu.liu.neg = scan('G:\\users\\negative-words.txt', what='character', comment.char=
hu.liu.pos = scan('G:\\users\\positive-words.txt', what='character', comment.char=
pos.words = c(hu.liu.pos,'upgrade')
neg.words = c(hu.liu.neg,'wait','wtf','waiting','epicfail','mechanical')
Datasetmakeinindia <- read.csv("G:\\users\\makeinindia.csv")
Datasetmakeinindia$text <- as.factor(Datasetmakeinindia$text)
makeinindia.scores = score.sentiment(Datasetmakeinindia$text,pos.words,neg.words,
path <- "G:\\users\\makeinindia.csv"
write.csv(makeinindia.scores,file=paste(path,"makeinindiascores.csv",sep=""),row.r
install.packages(RColorBrewer)
install.packages(ggplot2)
library(ggplot2)
hist(makeinindia.scores$score, xlab="scoreoftweets")
qplot(makeinindia.scores$score, xlab="scoreof tweets")
hist(makeinindia.scores$score)
plot(makeinindia.scores$score)

MakeInIndia<-read.csv(file.choose(),header=T)
MakeInIndiascores<-read.csv(file.choose(),header=T)

hist(MakeInIndiascores$score, xlab="Scores of Tweets")
qplot(makeinindiascores$score, xlab="Score of Tweets")
hist(makeinindiascores$score)
plot(makeinindiascores$score)
```

```
[Workspace loaded from ~/.RData]

> MakeInIndia<-read.csv(file.choose(),header=T)
> MakeInIndia<-read.csv(file.choose(),header=T)
> View(MakeInIndia)
> MakeInIndiascores<-read.csv(file.choose(),header=T)
```

```
Data
data                399 obs. of 20 variables
MakeInIndia         200 obs. of 16 variables
MakeInIndiascores   100 obs. of 3 variables
Functions
f                   function ()
```

Imported Make-In-India data set is displayed in R-studio



	text	favorited	favoriteCount	replyToSN	created
184	RT @makeinindia: Economists to academics. Think ta...	FALSE	0	NA	2/10/2016 5:
185	RT @makeinindia: Knowledge transfers+Creative ben...	FALSE	0	NA	2/10/2016 5:
186	RT @makeinindia: .@GEIndia is making railway coach...	FALSE	0	NA	2/10/2016 5:
187	RT @makeinindia: Rise of the machines at MII Centre ...	FALSE	0	NA	2/10/2016 5:
188	RT @makeinindia: Find out what @The_Linde_Group i...	FALSE	0	NA	2/10/2016 5:
189	RT @makeinindia: Industry leaders &amp; policymake...	FALSE	0	NA	2/10/2016 5:
190	RT @FollowCII: Inauguration of @makeinindia week b...	FALSE	0	NA	2/10/2016 5:
191	RT @makeinindia: Get up to speed with these auto gi...	FALSE	0	NA	2/10/2016 5:
192	RT @makeinindia: 27 of Mumbai's best restaurants. At...	FALSE	0	NA	2/10/2016 5:
193	RT @makeinindia: Delegates from 60 nations across t...	FALSE	0	NA	2/10/2016 5:
194	RT @makeinindia: India is a fast growing automobile c...	FALSE	0	NA	2/10/2016 5:
195	RT @makeinindia: .@NID_India will curate a design ex...	FALSE	0	NA	2/10/2016 5:
196	RT @makeinindia: Investing in India's #pharma sector...	FALSE	0	NA	2/10/2016 5:
197	RT @makeinindia: #IT &amp; #electronics pavilion at ...	FALSE	0	NA	2/10/2016 5:
198	RT @makeinindia: .@GEIndia is making railway coach...	FALSE	0	NA	2/10/2016 5:
199	RT @makeinindia: Find out what @The_Linde_Group i...	FALSE	0	NA	2/10/2016 5:
200	RT @makeinindia: 1000s of participants 100s of com...	FALSE	0	NA	2/10/2016 5:

Showing 184 to 200 of 200 entries

```
[Workspace loaded from ~/.RData]

> MakeInIndia<-read.csv(file.choose(),header=T)
> MakeInIndia<-read.csv(file.choose(),header=T)
> View(MakeInIndia)
>
```

```
Data
data          399 obs. of 20 variables
MakeInIndia   200 obs. of 16 variables
Functions
f             function ()
```

6.1.2.4 Standardize the Data

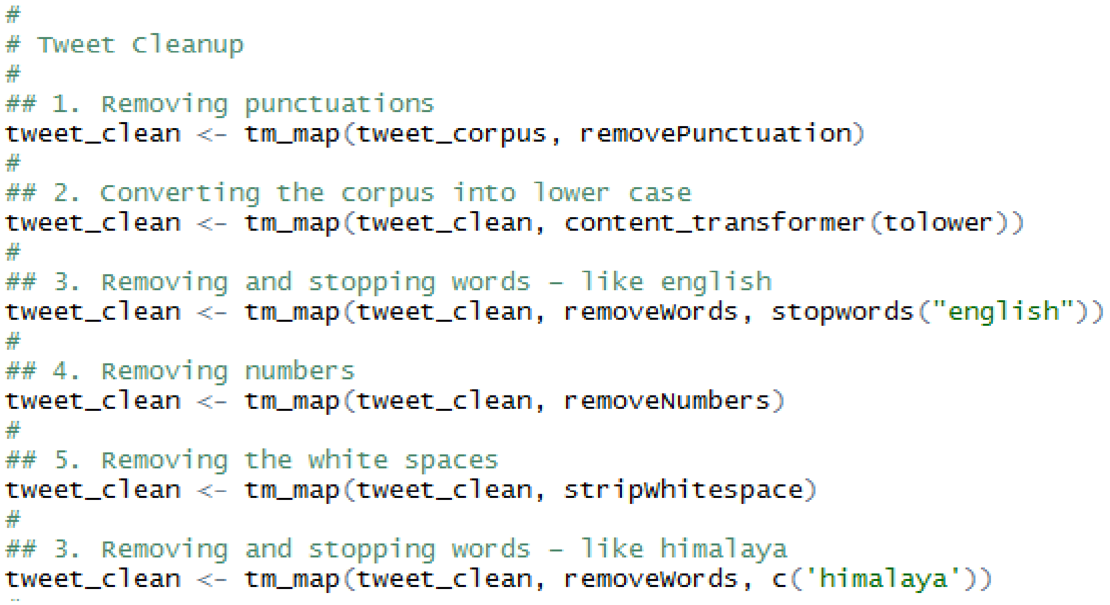
Once we have the tweets we just need to apply some functions to convert these tweets into some useful information. This process is called as standardizing the data. Removal of extra symbols which doesn't give any meaning to the tweets reduces the burden for classification.

Before performing data pre-processing we extract data from the twitter. To clean up the sentences with R we use gsub() function. Easiest way to clean the columns of our data frame with regular expression And last we get pre-processed data.

The cleaning of tweets requires the following steps:

- Remove html links from the tweets
- Remove retweet entities
- Remove all hashtags
- Remove all @people
- Remove all punctuation
- Remove all numbers
- Remove all unnecessary white spaces
- Convert all text into lowercase and
- Remove duplicates

R Code <...>



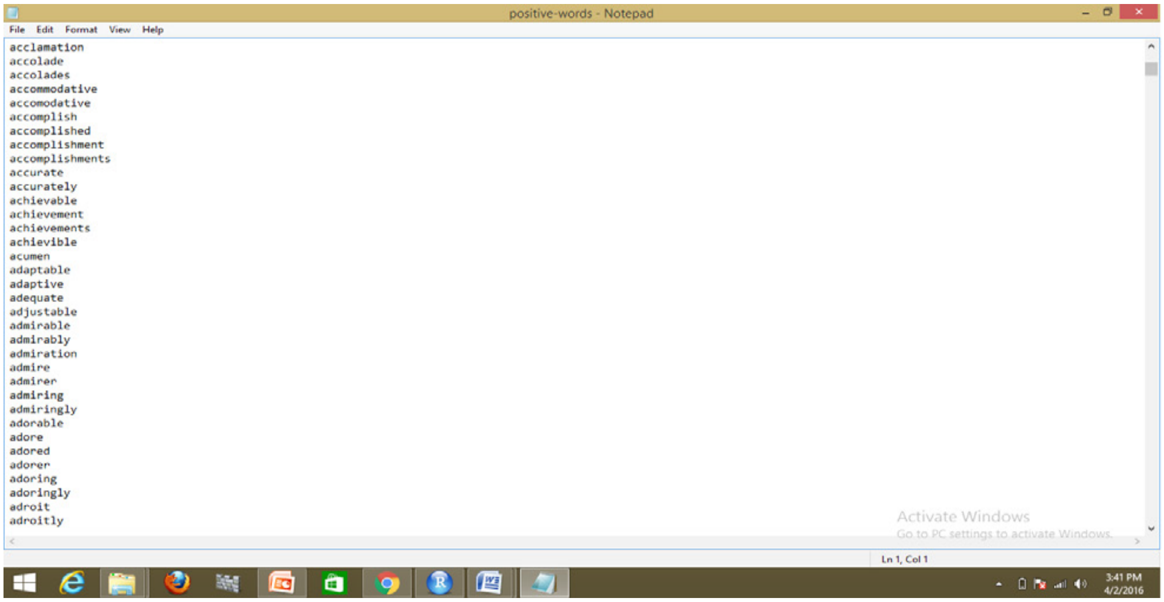
```
#
# Tweet Cleanup
#
## 1. Removing punctuations
tweet_clean <- tm_map(tweet_corpus, removePunctuation)
#
## 2. Converting the corpus into lower case
tweet_clean <- tm_map(tweet_clean, content_transformer(tolower))
#
## 3. Removing and stopping words - like english
tweet_clean <- tm_map(tweet_clean, removeWords, stopwords("english"))
#
## 4. Removing numbers
tweet_clean <- tm_map(tweet_clean, removeNumbers)
#
## 5. Removing the white spaces
tweet_clean <- tm_map(tweet_clean, stripWhitespace)
#
## 3. Removing and stopping words - like himalaya
tweet_clean <- tm_map(tweet_clean, removeWords, c('himalaya'))
```

6.1.2.5 Classify the Data

The process of sentiment analysis is to calculate the synchronization of the words of the tweets with respect to Positive word list6 and negative word list. For this negative word list and positive word list to be downloaded and need to be saved to working directory.

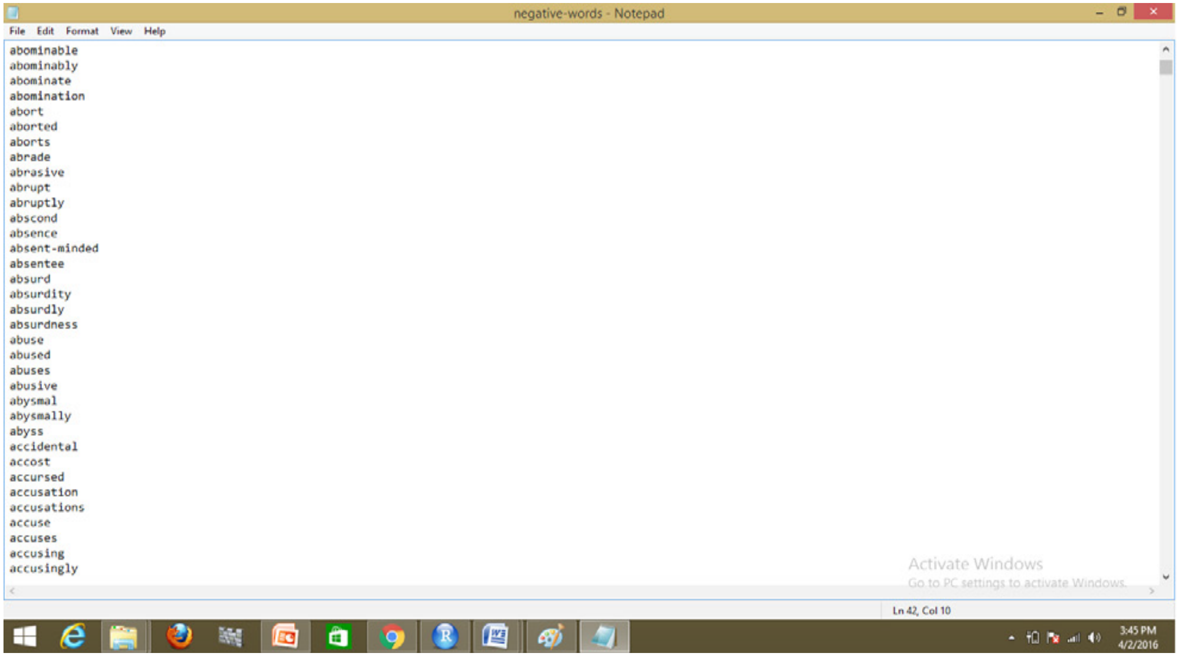
Sentiment analysis requires two additional packages PlyR and StringR to manipulate strings.

File of Positive words:



```
acclamation
accolade
accolades
accommodative
accomodative
accomplish
accomplished
accomplishment
accomplishments
accurate
accurately
achievable
achievement
achievements
achievible
acumen
adaptable
adaptive
adequate
adjustable
admirable
admirably
admiration
admire
admirer
admiring
admiringly
adorable
adore
adored
adorer
adoring
adoringly
adroit
adroitly
```

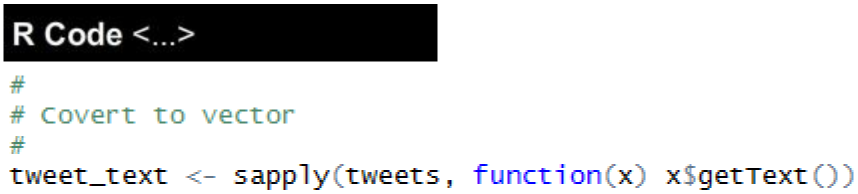
File of Negative words:



```
abominable
abominably
abominate
abomination
abort
aborted
aborts
abrade
abrasive
abrupt
abruptly
abscond
absence
absent-minded
absentee
absurd
absurdity
absurdly
absurdness
abuse
abused
abuses
abusive
abysmal
abysmally
abyss
accidental
accost
accursed
accusation
accusations
accuse
accuses
accusing
accusingly
```

6.1.2.6 Run R commands for processing the tweets

Step 1: Convert the received tweets into vectors



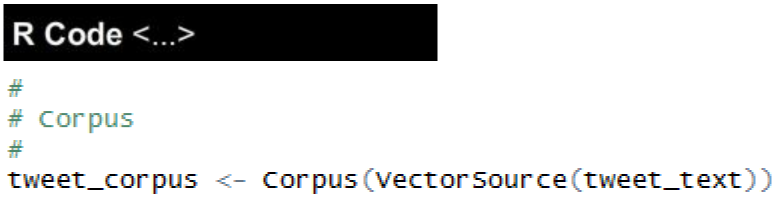
R Code <...>

```
#
# Covert to vector
#
tweet_text <- sapply(tweets, function(x) x$getText())
```

A **vector** is a sequence of data elements of the same basic type. Members in a vector are officially called components. Nevertheless, we will just call them members in this site. Here is a vector containing three numeric values 2, 3 and 5. > c (2, 3, 5)

sapply () is used for traversing over the data in a vector and calling specified function for each item.

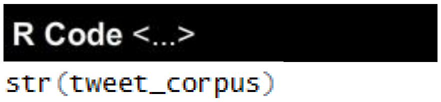
Step: 2: Convert the vector to corpus



R Code <...>

```
#
# Corpus
#
tweet_corpus <- Corpus(VectorSource(tweet_text))
```

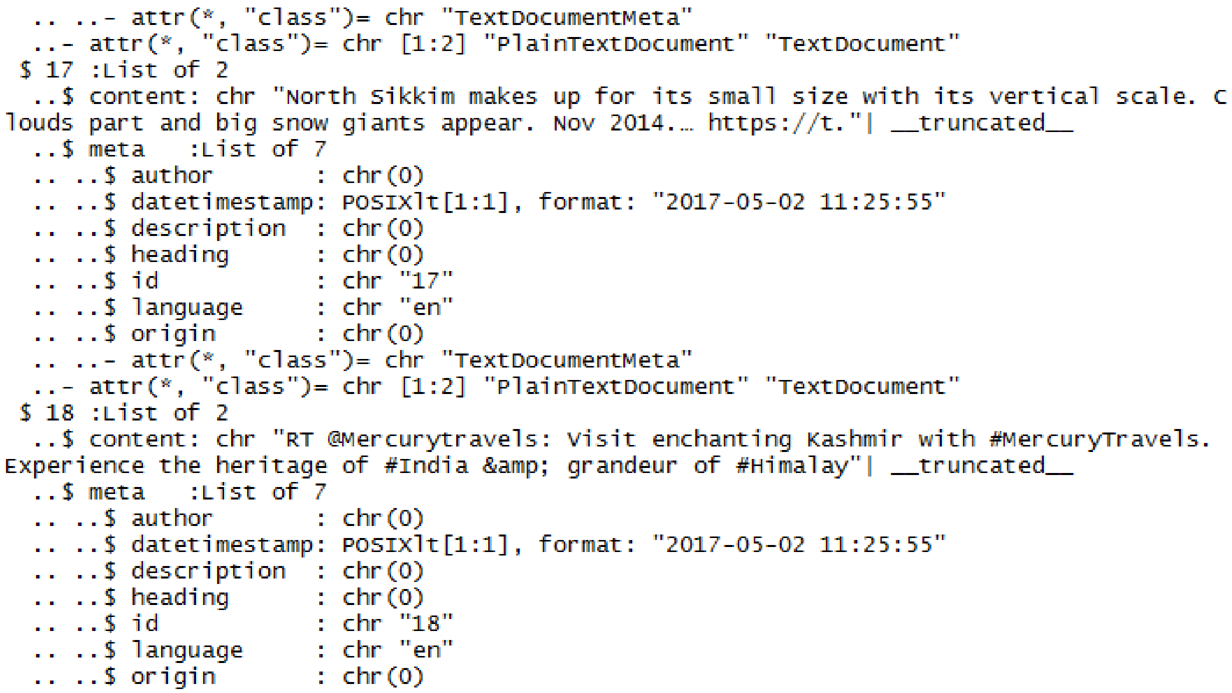
For checking the structure of the tweet_corpus, use:



R Code <...>

```
str(tweet_corpus)
```

OUTPUT:

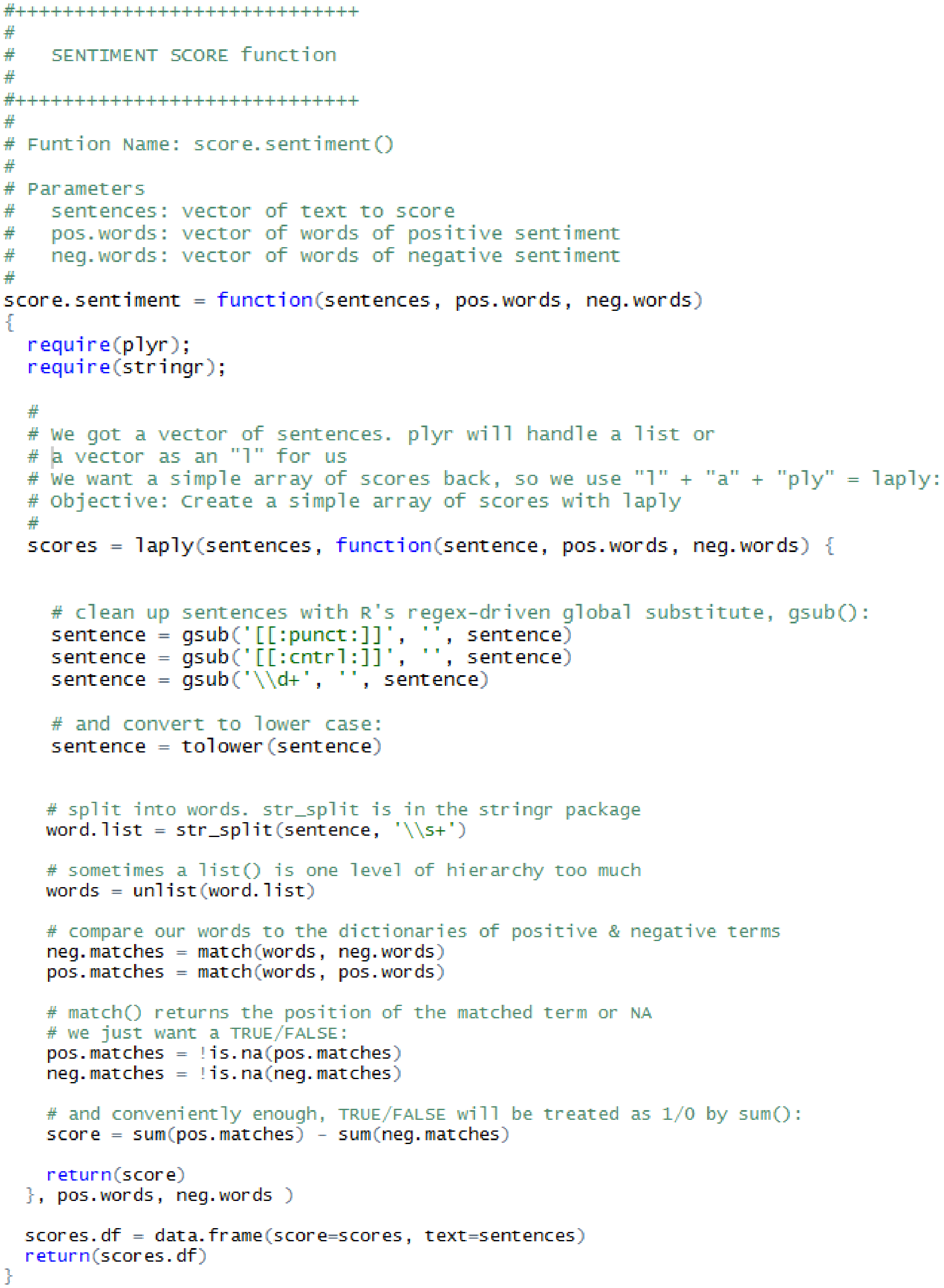


```
  .. ..- attr(*, "class")= chr "TextDocumentMeta"
  ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
 $ 17 :List of 2
  ..$ content: chr "North Sikkim makes up for its small size with its vertical scale. C
louds part and big snow giants appear. Nov 2014.… https://t."| __truncated__
  ..$ meta   :List of 7
  .. ..$ author       : chr(0)
  .. ..$ datetimestamp: POSIXlt[1:1], format: "2017-05-02 11:25:55"
  .. ..$ description  : chr(0)
  .. ..$ heading      : chr(0)
  .. ..$ id           : chr "17"
  .. ..$ language     : chr "en"
  .. ..$ origin       : chr(0)
  .. ..- attr(*, "class")= chr "TextDocumentMeta"
  ..- attr(*, "class")= chr [1:2] "PlainTextDocument" "TextDocument"
 $ 18 :List of 2
  ..$ content: chr "RT @Mercurytravels: Visit enchanting Kashmir with #MercuryTravels.
Experience the heritage of #India &amp; grandeur of #Himalay"| __truncated__
  ..$ meta   :List of 7
  .. ..$ author       : chr(0)
  .. ..$ datetimestamp: POSIXlt[1:1], format: "2017-05-02 11:25:55"
  .. ..$ description  : chr(0)
  .. ..$ heading      : chr(0)
  .. ..$ id           : chr "18"
  .. ..$ language     : chr "en"
  .. ..$ origin       : chr(0)
```

The main structure for managing documents in **tm** (text mining package) is a so-called **Corpus**, representing a collection of text documents.

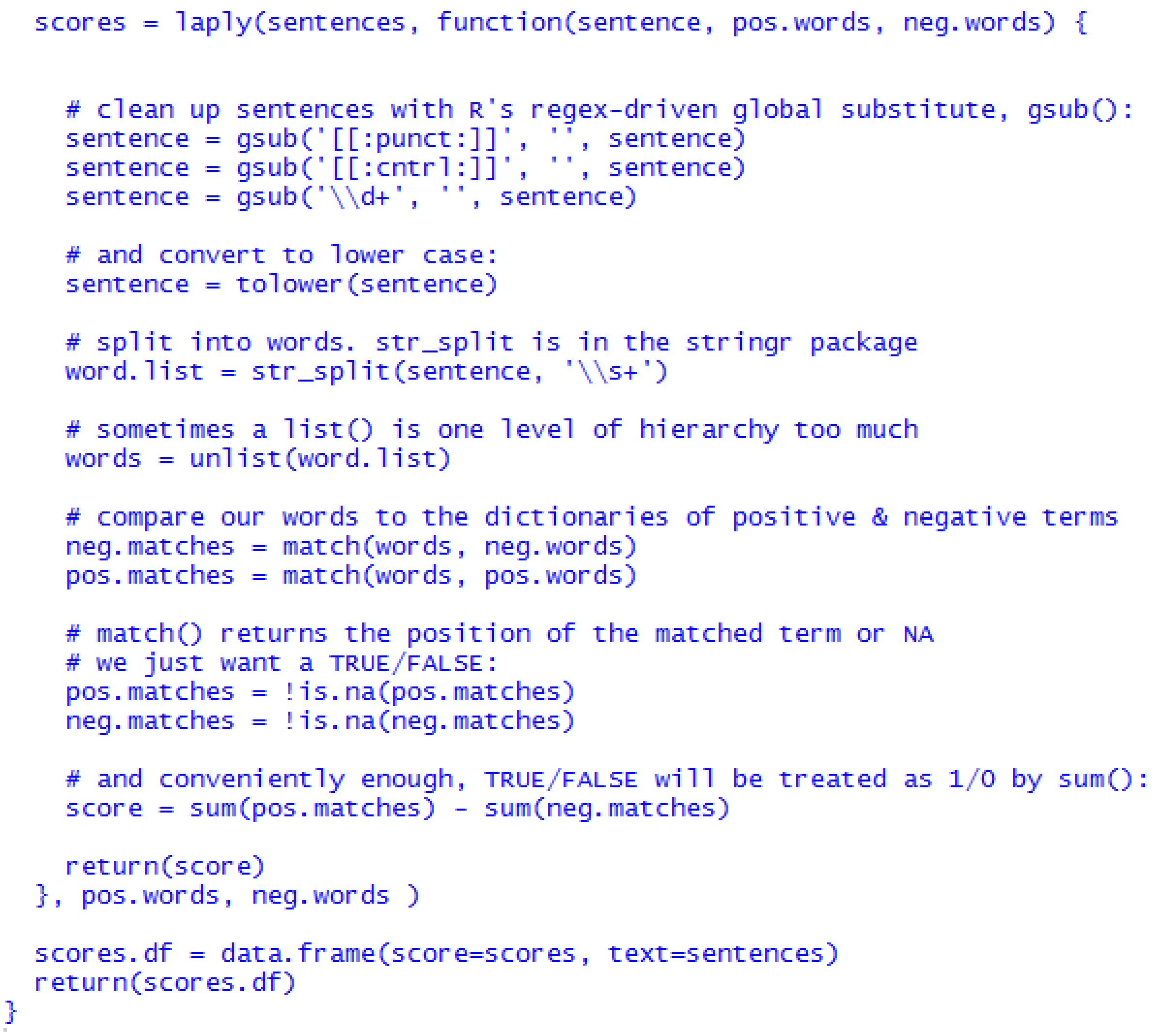
Step 3: Sentiment Score Logic

R Code <...>



```
#++++++++++++++++++++++++++++++
#
#   SENTIMENT SCORE function
#
#++++++++++++++++++++++++++++++
#
# Funtion Name: score.sentiment()
#
# Parameters
#   sentences: vector of text to score
#   pos.words: vector of words of positive sentiment
#   neg.words: vector of words of negative sentiment
#
score.sentiment = function(sentences, pos.words, neg.words)
{
  require(plyr);
  require(stringr);

  #
  # We got a vector of sentences. plyr will handle a list or
  # a vector as an "l" for us
  # We want a simple array of scores back, so we use "l" + "a" + "ply" = laply:
  # Objective: Create a simple array of scores with laply
  #
  scores = laply(sentences, function(sentence, pos.words, neg.words) {


    # clean up sentences with R's regex-driven global substitute, gsub():
    sentence = gsub('[[:punct:]]', '', sentence)
    sentence = gsub('[[:cntrl:]]', '', sentence)
    sentence = gsub('\\d+', '', sentence)

    # and convert to lower case:
    sentence = tolower(sentence)


    # split into words. str_split is in the stringr package
    word.list = str_split(sentence, '\\s+')

    # sometimes a list() is one level of hierarchy too much
    words = unlist(word.list)

    # compare our words to the dictionaries of positive & negative terms
    neg.matches = match(words, neg.words)
    pos.matches = match(words, pos.words)

    # match() returns the position of the matched term or NA
    # we just want a TRUE/FALSE:
    pos.matches = !is.na(pos.matches)
    neg.matches = !is.na(neg.matches)

    # and conveniently enough, TRUE/FALSE will be treated as 1/0 by sum():
    score = sum(pos.matches) - sum(neg.matches)

    return(score)
  }, pos.words, neg.words )

  scores.df = data.frame(score=scores, text=sentences)
  return(scores.df)
}
```

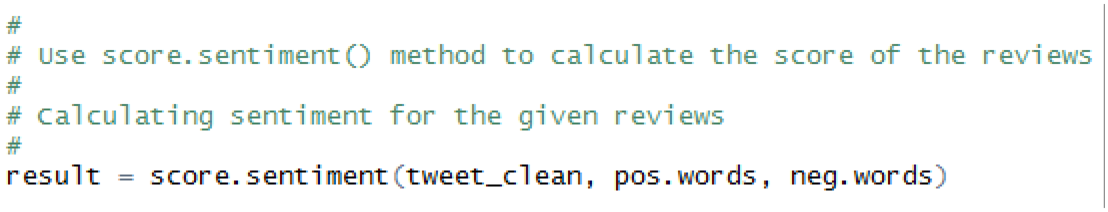
OUTPUT:



```
  scores = laply(sentences, function(sentence, pos.words, neg.words) {


    # clean up sentences with R's regex-driven global substitute, gsub():
    sentence = gsub('[[:punct:]]', '', sentence)
    sentence = gsub('[[:cntrl:]]', '', sentence)
    sentence = gsub('\\d+', '', sentence)

    # and convert to lower case:
    sentence = tolower(sentence)

    # split into words. str_split is in the stringr package
    word.list = str_split(sentence, '\\s+')

    # sometimes a list() is one level of hierarchy too much
    words = unlist(word.list)

    # compare our words to the dictionaries of positive & negative terms
    neg.matches = match(words, neg.words)
    pos.matches = match(words, pos.words)

    # match() returns the position of the matched term or NA
    # we just want a TRUE/FALSE:
    pos.matches = !is.na(pos.matches)
    neg.matches = !is.na(neg.matches)

    # and conveniently enough, TRUE/FALSE will be treated as 1/0 by sum():
    score = sum(pos.matches) - sum(neg.matches)

    return(score)
  }, pos.words, neg.words )

  scores.df = data.frame(score=scores, text=sentences)
  return(scores.df)
}
```

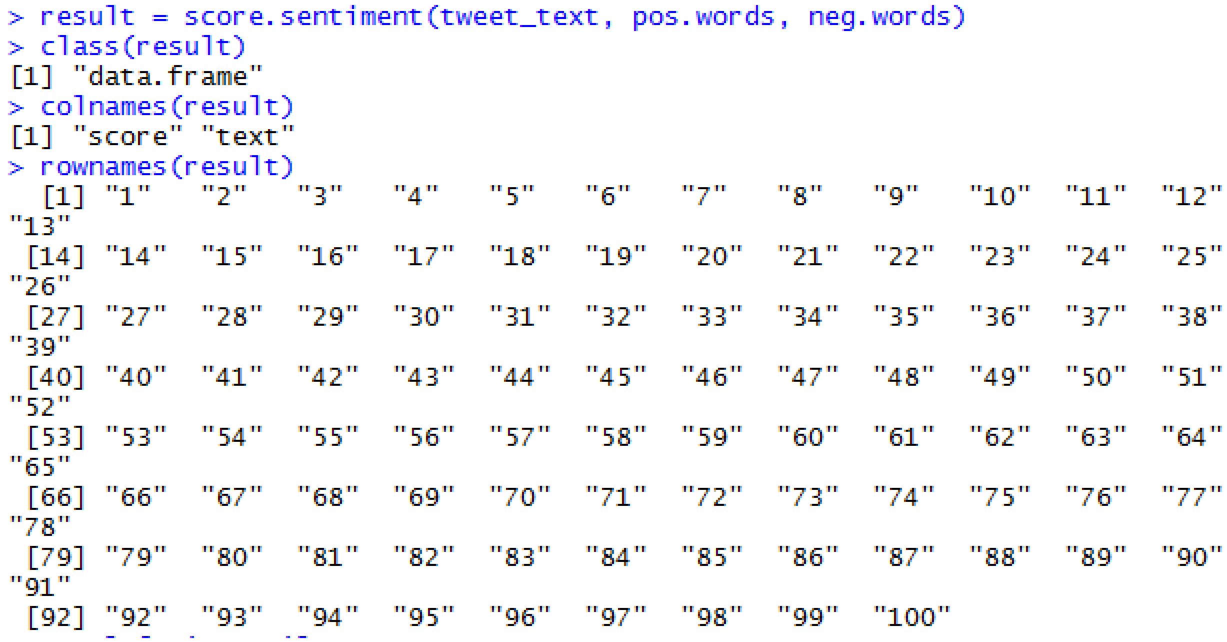
Step 4: Sentiment Score Calculation

R Code <...>



```
#
# Use score.sentiment() method to calculate the score of the reviews
#
# Calculating sentiment for the given reviews
#
result = score.sentiment(tweet_clean, pos.words, neg.words)
```

OUTPUT:



```
> result = score.sentiment(tweet_text, pos.words, neg.words)
> class(result)
[1] "data.frame"
> colnames(result)
[1] "score" "text"
> rownames(result)
  [1] "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9"   "10"  "11"  "12"
"13"
 [14] "14"  "15"  "16"  "17"  "18"  "19"  "20"  "21"  "22"  "23"  "24"  "25"
"26"
 [27] "27"  "28"  "29"  "30"  "31"  "32"  "33"  "34"  "35"  "36"  "37"  "38"
"39"
 [40] "40"  "41"  "42"  "43"  "44"  "45"  "46"  "47"  "48"  "49"  "50"  "51"
"52"
 [53] "53"  "54"  "55"  "56"  "57"  "58"  "59"  "60"  "61"  "62"  "63"  "64"
"65"
 [66] "66"  "67"  "68"  "69"  "70"  "71"  "72"  "73"  "74"  "75"  "76"  "77"
"78"
 [79] "79"  "80"  "81"  "82"  "83"  "84"  "85"  "86"  "87"  "88"  "89"  "90"
"91"
 [92] "92"  "93"  "94"  "95"  "96"  "97"  "98"  "99"  "100"
```

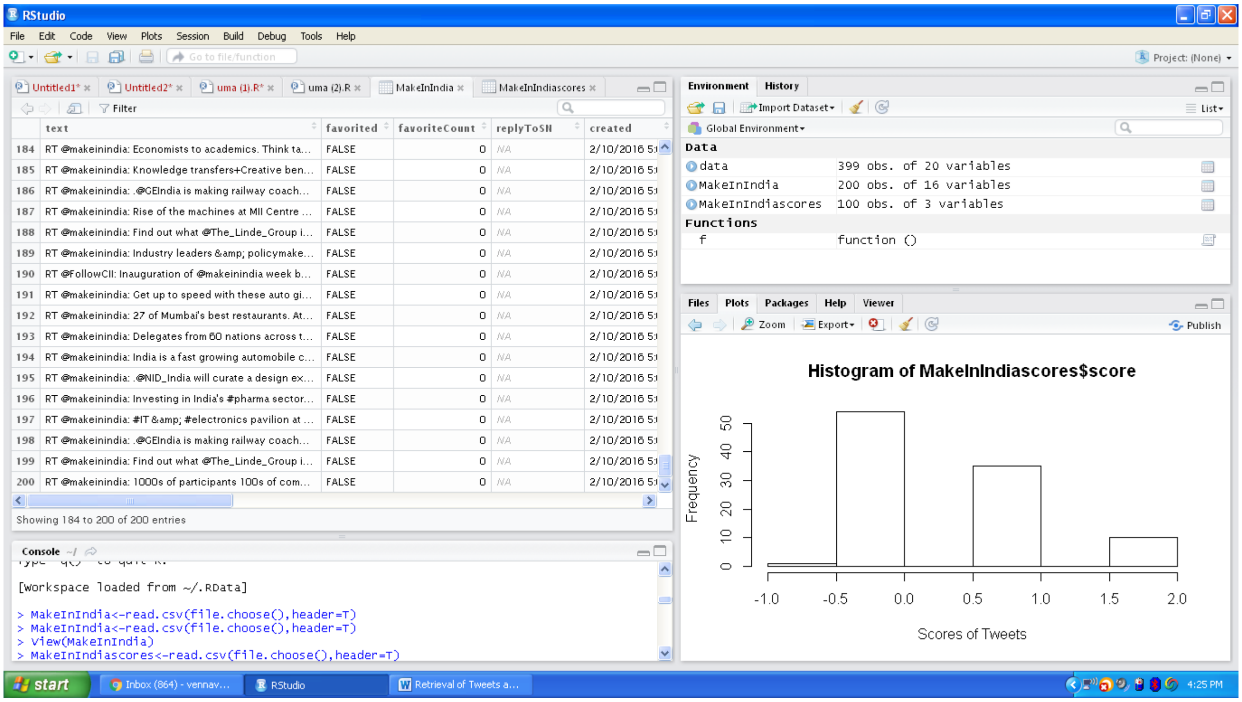
6.1.2.7 Establish R Plotter, Histogram and Wordcloud

The process of sentiment analysis is to calculate the synchronization of the words of the tweets with respect.

Make-In-India dataset status in terms of public opinions is visualized here. It is displayed using visual histograms and other related plots to visualize the sentiments of the user. It is achieved using **hist function**. I have used a package **RColorBrewer** to add the colors.

All the tweets are considered on to X-axis and the corresponding scores are on the Y-axis. The bar chart tells us that either score that is highlighted to give a decision on the Make-In-India schemes into people’s opinions.

Make in India Sentiment Histogram



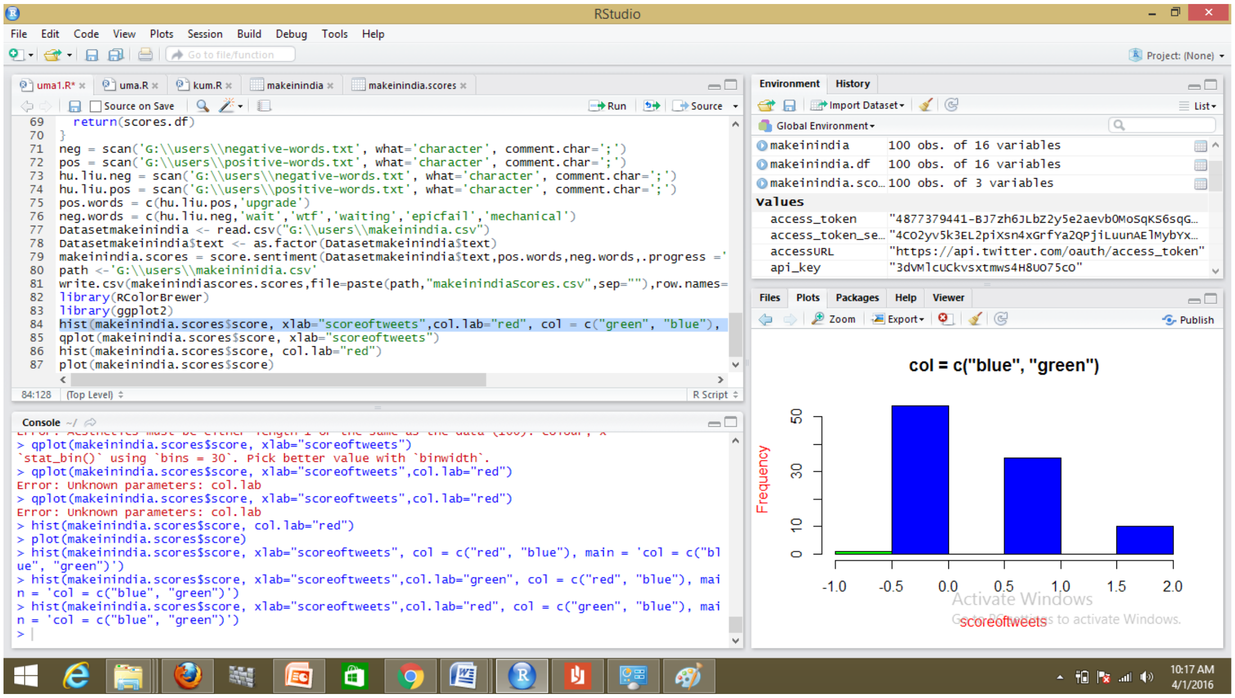
	text	favorited	favoriteCount	replyToSN	created
184	RT @makeinindia: Economists to academics. Think ta...	FALSE	0	NA	2/10/2016 5:
185	RT @makeinindia: Knowledge transfers+Creative ben...	FALSE	0	NA	2/10/2016 5:
186	RT @makeinindia: .@GEIndia is making railway coach...	FALSE	0	NA	2/10/2016 5:
187	RT @makeinindia: Rise of the machines at MII Centre ...	FALSE	0	NA	2/10/2016 5:
188	RT @makeinindia: Find out what @The_Linde_Group i...	FALSE	0	NA	2/10/2016 5:
189	RT @makeinindia: Industry leaders &amp; policymake...	FALSE	0	NA	2/10/2016 5:
190	RT @FollowCII: Inauguration of @makeinindia week b...	FALSE	0	NA	2/10/2016 5:
191	RT @makeinindia: Get up to speed with these auto gi...	FALSE	0	NA	2/10/2016 5:
192	RT @makeinindia: 27 of Mumbai's best restaurants. At...	FALSE	0	NA	2/10/2016 5:
193	RT @makeinindia: Delegates from 60 nations across t...	FALSE	0	NA	2/10/2016 5:
194	RT @makeinindia: India is a fast growing automobile c...	FALSE	0	NA	2/10/2016 5:
195	RT @makeinindia: .@NID_India will curate a design ex...	FALSE	0	NA	2/10/2016 5:
196	RT @makeinindia: Investing in India's #pharma sector...	FALSE	0	NA	2/10/2016 5:
197	RT @makeinindia: #IT &amp; #electronics pavilion at ...	FALSE	0	NA	2/10/2016 5:
198	RT @makeinindia: .@GEIndia is making railway coach...	FALSE	0	NA	2/10/2016 5:
199	RT @makeinindia: Find out what @The_Linde_Group i...	FALSE	0	NA	2/10/2016 5:
200	RT @makeinindia: 1000s of participants 100s of com...	FALSE	0	NA	2/10/2016 5:

Showing 184 to 200 of 200 entries

```
Data
data                399 obs. of 20 variables
MakeInIndia         200 obs. of 16 variables
MakeInIndiascores   100 obs. of 3 variables
Functions
  f                 function ()
```

```
[Workspace loaded from ~/.RData]

> MakeInIndia<-read.csv(file.choose(),header=T)
> MakeInIndia<-read.csv(file.choose(),header=T)
> View(MakeInIndia)
> MakeInIndiascores<-read.csv(file.choose(),header=T)
```

The histogram displays the frequency of tweets with respect to scores allocated to each tweet. The x-axis shows the score of each tweet as a negative, neutral and positive integer. A positive score shows positive or good sentiments related to that particular tweet whereas a negative score represents negative or bad sentiments related to the same. A score of zero or close to zero specifies a neutral sentiment. The more positive the score, better are the sentiments of the person tweeting and vice-versa.



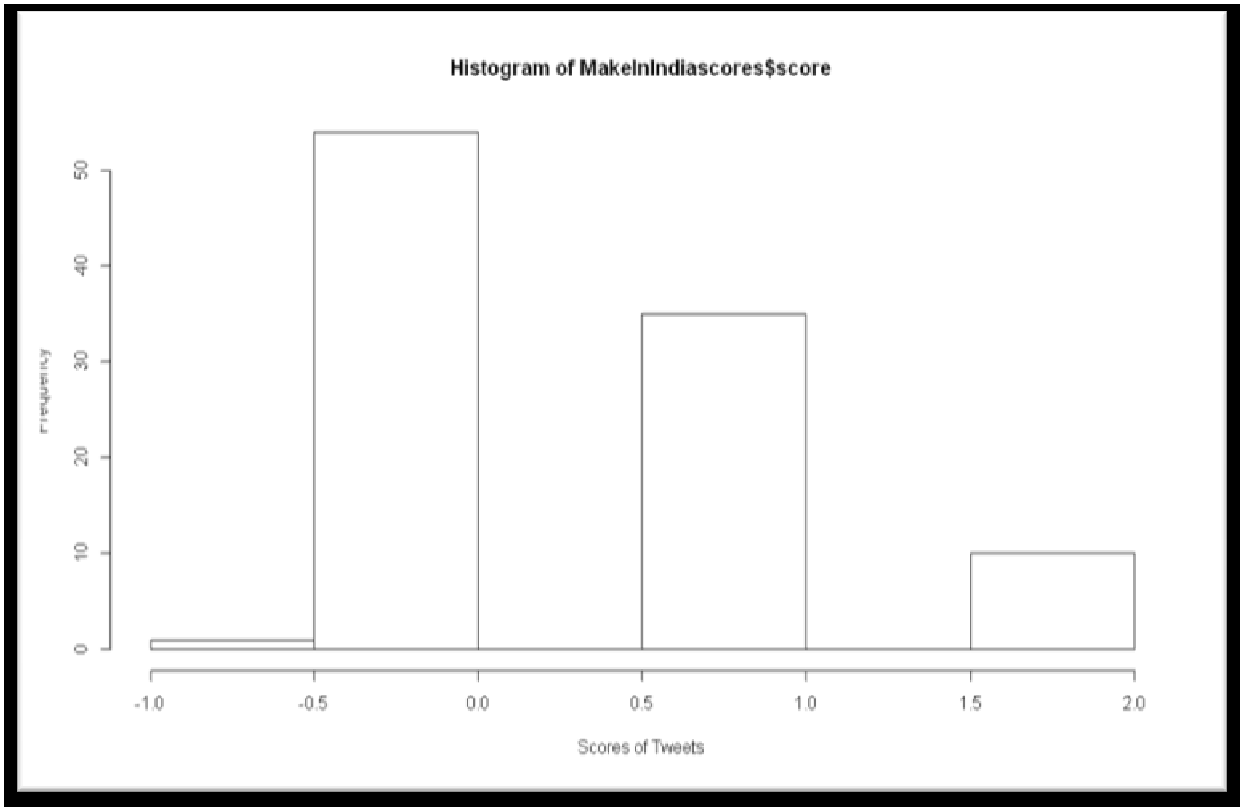
```
  return(scores.df)
}
neg = scan('G:\\users\\negative-words.txt', what='character', comment.char=';')
pos = scan('G:\\users\\positive-words.txt', what='character', comment.char=';')
hu.liu.neg = scan('G:\\users\\negative-words.txt', what='character', comment.char=';')
hu.liu.pos = scan('G:\\users\\positive-words.txt', what='character', comment.char=';')
pos.words = c(hu.liu.pos,'upgrade')
neg.words = c(hu.liu.neg,'wait','wtf','waiting','epicfail','mechanical')
Datasetmakeinindia <- read.csv("G:\\users\\makeinindia.csv")
Datasetmakeinindia$text <- as.factor(Datasetmakeinindia$text)
makeinindia.scores = score.sentiment(Datasetmakeinindia$text,pos.words,neg.words,.progress ='
path <-'G:\\users\\makeininidia.csv'
write.csv(makeinindiascores.scores,file=paste(path,"makeindiaScores.csv",sep=""),row.names=
library(RColorBrewer)
library(ggplot2)
hist(makeinindia.scores$score, xlab="scoreoftweets",col.lab="red", col = c("green", "blue"),
qplot(makeinindia.scores$score, xlab="scoreoftweets")
hist(makeinindia.scores$score, col.lab="red")
plot(makeinindia.scores$score)
```

84:128 (Top Level) R Script

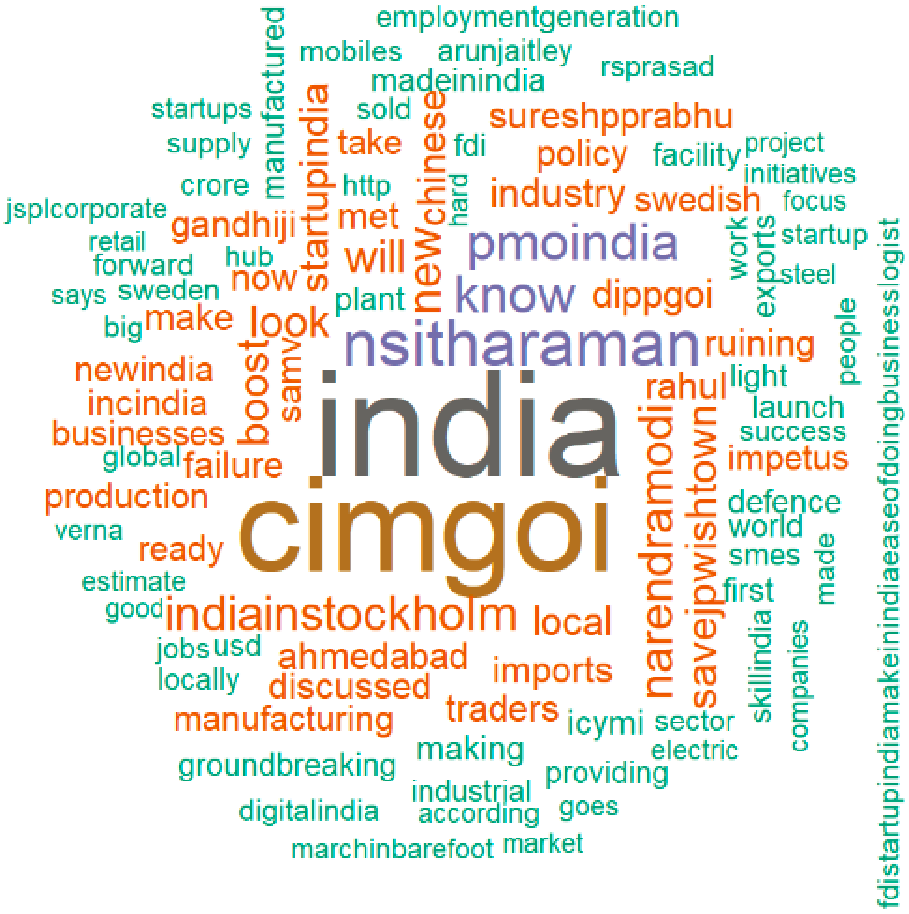
Console ~/

```
> qplot(makeinindia.scores$score, xlab="scoreoftweets")
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
> qplot(makeinindia.scores$score, xlab="scoreoftweets",col.lab="red")
Error: Unknown parameters: col.lab
> qplot(makeinindia.scores$score, xlab="scoreoftweets",col.lab="red")
Error: Unknown parameters: col.lab
> hist(makeinindia.scores$score, col.lab="red")
> plot(makeinindia.scores$score)
> hist(makeinindia.scores$score, xlab="scoreoftweets", col = c("red", "blue"), main = 'col = c("bl
ue", "green")')
> hist(makeinindia.scores$score, xlab="scoreoftweets",col.lab="green", col = c("red", "blue"), mai
n = 'col = c("blue", "green")')
> hist(makeinindia.scores$score, xlab="scoreoftweets",col.lab="red", col = c("green", "blue"), mai
n = 'col = c("blue", "green")')
> |
```

```
makeinindia       100 obs. of 16 variables
makeinindia.df    100 obs. of 16 variables
makeinindia.sco…  100 obs. of 3 variables
Values
  access_token       "4877379441-BJ7zh6JLbZ2y5e2aevbOMoSqKS6sqG…
  access_token_se…   "4CO2yv5k3EL2piXsn4xGrfYa2QPjiLuunAElMybYx…
  accessURL          "https://api.twitter.com/oauth/access_token"
  api_key            "3dVMlcUCkvsxtmws4H8UO75cO"
```



Make in India - Word Cloud



After the necessary cleaning, a word cloud was made to understand the most frequently used terms. The following observations are made:

The most common word that was used in the tweet was ‘India’, followed by ‘cimgoi’ - the official handle of the Commerce Minister (@CimGOI). The Commerce Minister, Nirmala Sitharaman was tagged more often in the tweets more frequent than the Prime Minister Narendra Modi & the PMO Office. The new Commerce Minister Suresh Prabhu is also gaining popularity in the tweets associated with Make in India.

Some words gain popularity based on an event or a trend and occur in the tweet over a period of time. The word usage depends upon the popularity, significant and the reach of the hashtag. E.g. Twitter users tagged #SaveJPWishtown hashtag with the Make in India campaign in order to gain attention from the government. Many homebuyers wanted the government to have a resolution to the issue of not getting their flats delivered as promised.

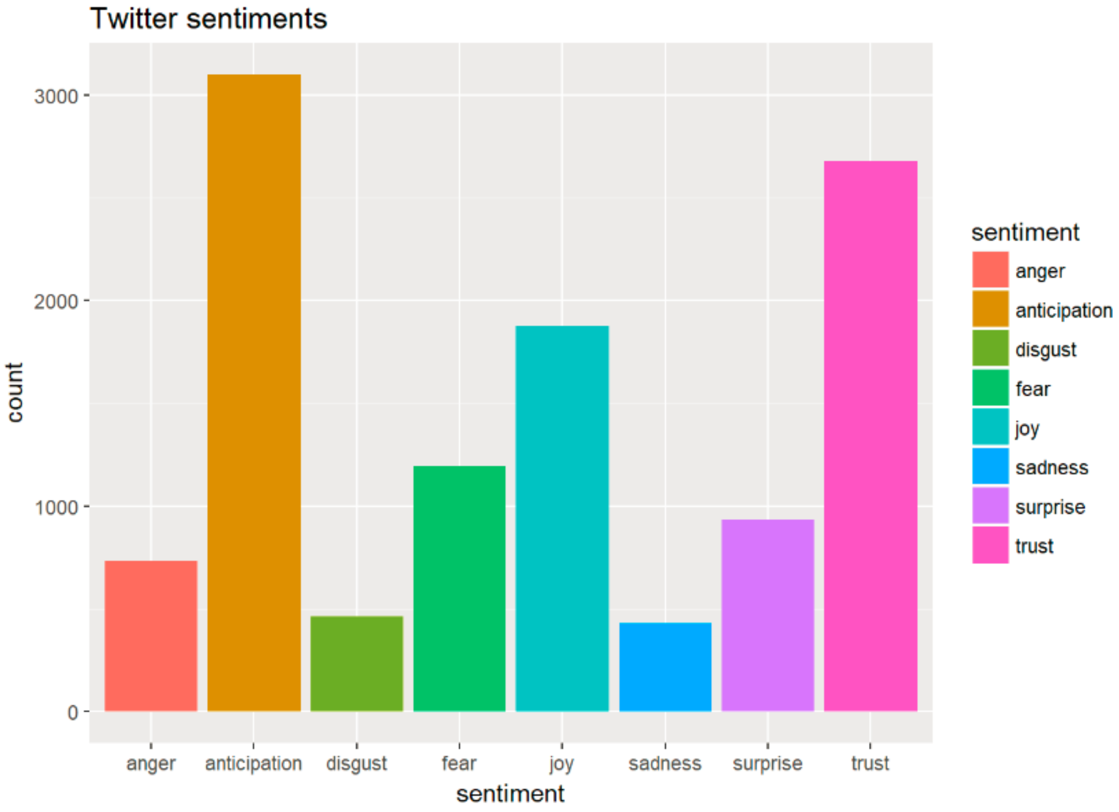
Another popular word that was used is #IndiainStockholm. The aim of this campaign by MakeinIndia was to attract and create awareness about Swedish investments in India. Most tweets during the period were focused towards Swedish investments prior to the two-day workshop that will be held in Stockholm, Sweden in October.

Sentiment Analysis:

A sentiment analysis is performed to determine whether tweets were positive, negative or neutral. It's also known as text or opinion mining deriving the views, emotions or attitude of the user about Make in India Campaign.

```
##
## negative  neutral positive
##      199    10057     4967
```

Emotions:



We can observe the anticipation and trust level score the highest. People believe that the Make in India campaign will help transform the nation. They believe in the government to deliver the promises laid out by them. There is also a sense of joy and surprise when a new announcement is made.

6.1.2.8 Interpretation

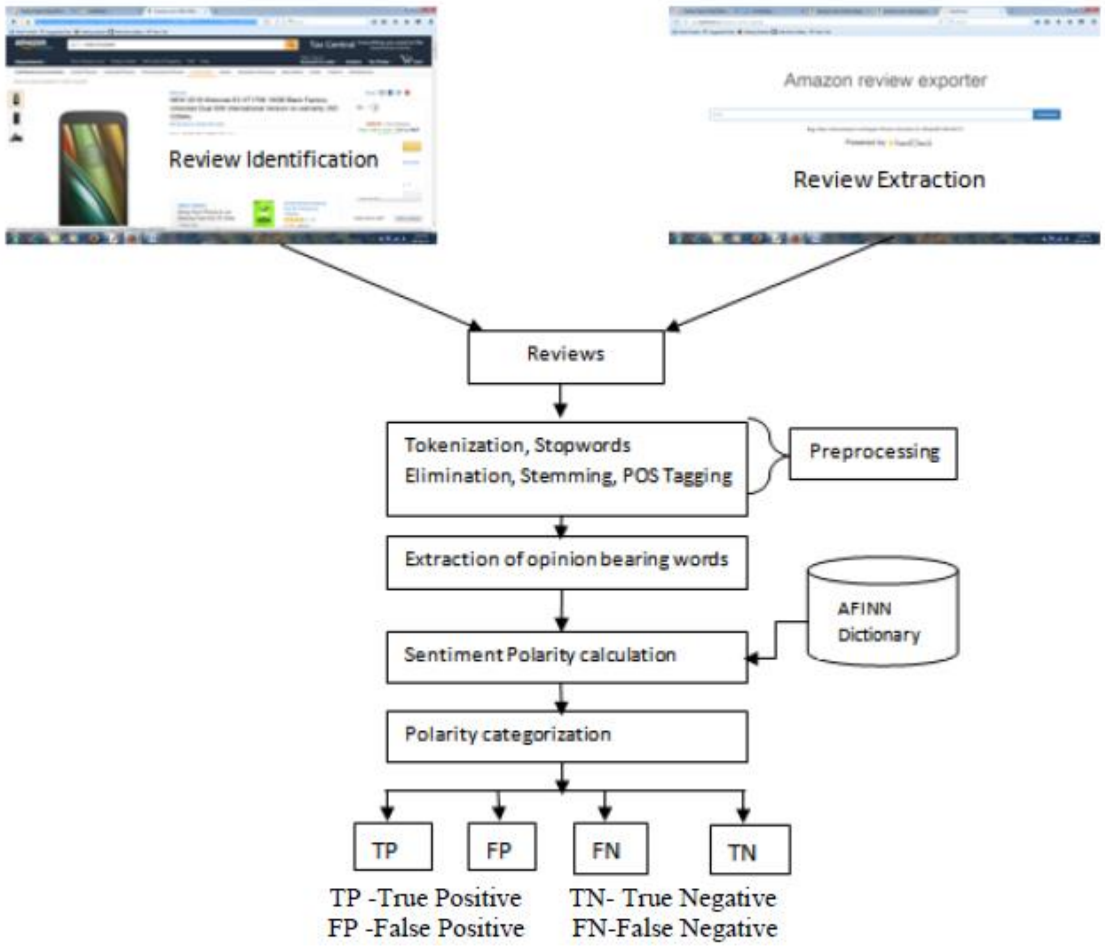
Make in India campaign overall has a good brand perception. Most of the feedback that is associated with #MakeinIndia is neutral or positive. However, the reach of the campaign is yet to reach the non-followers of the campaign. Currently, most promoters are somehow related with the government or BJP. To be effective, the government should try to get corporates also promote the #MakeinIndia hashtag on their pages. There is a huge level of trust associated with the government. People anticipated that the initiative will help transform the nation.

6.2 Customer Reviews Analysis of Amazon Baby Products

Amazon is also one of leading e-commerce companies which possess and analyze these customers' data to advance their service and revenue. The data set we used in our project is Amazon product data.

We have analyzed the sentiment analysis for **Baby Products in Amazon**. Since Amazon has huge resources of customer reviews with their rates. Therefore, my major area of interest in this research is to convert descriptive data in the form of customer reviews into quantified format (numeric sentiment scores) so that we can view the customers' attitudes directly.

6.2.1 Architecture



Tools and Packages used in R:

In this application I have used following **packages**:

- **ROAuth**: It provides an interface to the OAuth 1.0 specification. This package allows users to connect the server of their choice and authenticate via OAuth package.

- **plyr**: It is a group of tools that resolves a common group of problems. We need to break a big problem down into number of manageable pieces, operate on them and then put all of them back together.

- **stringr:** It is a set of packages that make string functions more easier, simpler and reliable for usage.

- **ggplot2:** It is an implementation of the grammar of graphics in R. We can build up a plot step by step from numerous data sources.

- **RColorBrewer**: It provides palettes for coloring nice plots/charts shaded according to a variable.

- **tm**: it is a framework for text mining applications inside R.

- **wordcloud***:* This package supports in creating pretty viewing word clouds in Text Mining.

- **sentiment***:* Used for sentiment analysis that includes Bayesian classifiers for positivity and negativity classification.

- **tm**: it is a framework for text mining applications inside R.

- **wordcloud***:* It supports in creating pretty viewing word clouds in Text Mining.

- **sentiment***:* It is a R package with tools for sentiment analysis including Bayesian classifiers for positivity/negativity and emotion classification.

6.2.2 Methodology

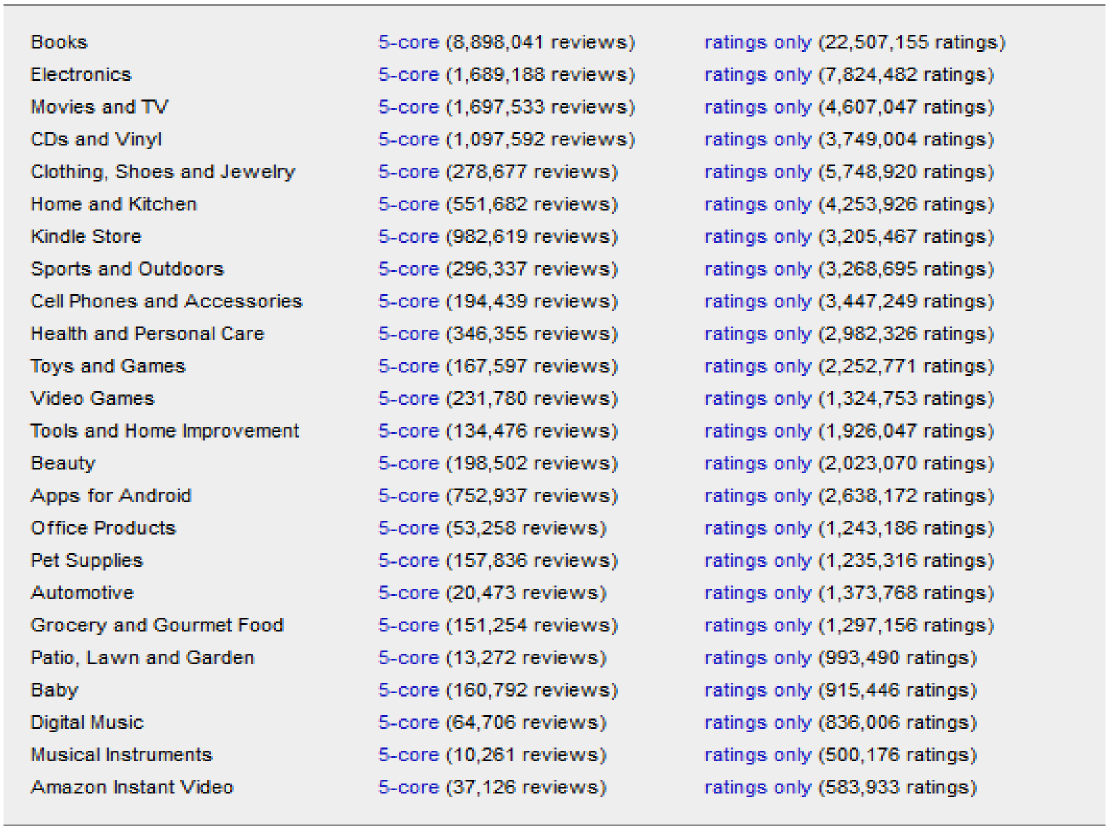
Following steps needs to be followed for performing sentiment analysis of Amazon reviews.

1 Collect Customer Reviews data
2 Execute Installation code in RStudio
3 Standardize the data
4 Classify the Data with R Tool commands
5 Run R commands for processing the Data file
6 Establish R Plotter, Histogram and Wordcloud to view results
7 Interpretation

6.2.2.1 Collect Customer Reviews Data

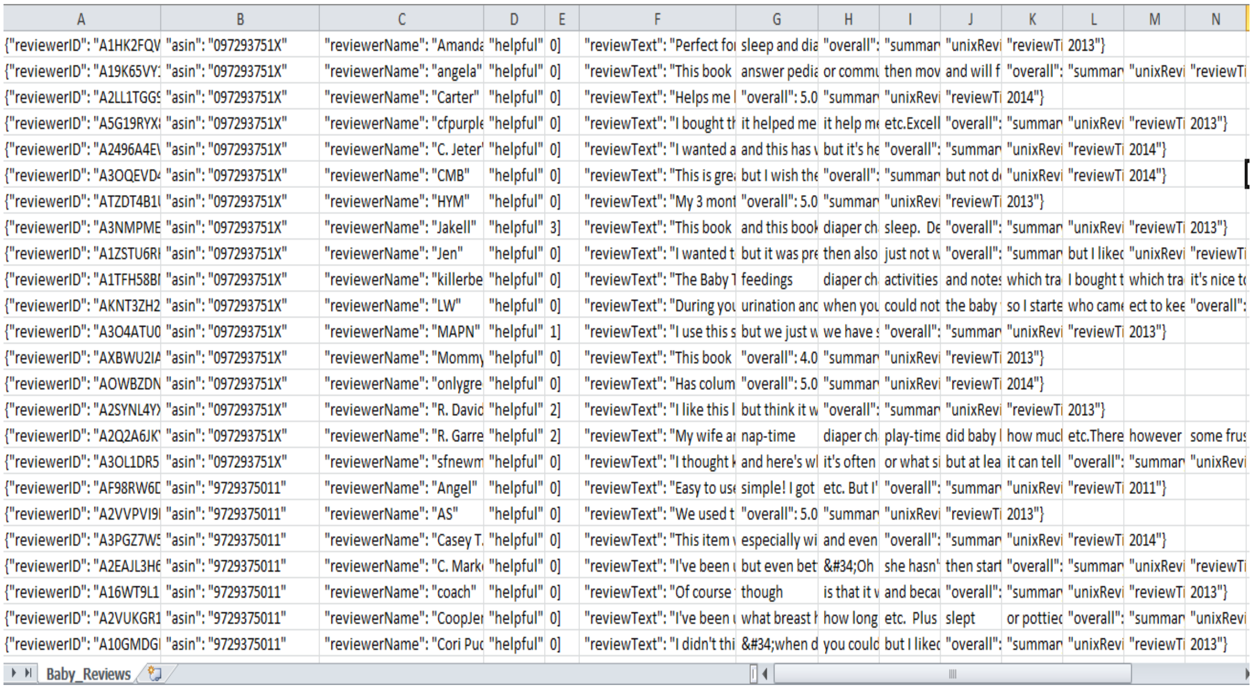
Data used is a set of product reviews collected from amazon.com on baby products.

The dataset was downloaded from http://jmcauley.ucsd.edu/data/amazon/ and the text file downloaded was having more than 1.5 million reviews for different product category.



Books	5-core (8,898,041 reviews)	ratings only (22,507,155 ratings)
Electronics	5-core (1,689,188 reviews)	ratings only (7,824,482 ratings)
Movies and TV	5-core (1,697,533 reviews)	ratings only (4,607,047 ratings)
CDs and Vinyl	5-core (1,097,592 reviews)	ratings only (3,749,004 ratings)
Clothing, Shoes and Jewelry	5-core (278,677 reviews)	ratings only (5,748,920 ratings)
Home and Kitchen	5-core (551,682 reviews)	ratings only (4,253,926 ratings)
Kindle Store	5-core (982,619 reviews)	ratings only (3,205,467 ratings)
Sports and Outdoors	5-core (296,337 reviews)	ratings only (3,268,695 ratings)
Cell Phones and Accessories	5-core (194,439 reviews)	ratings only (3,447,249 ratings)
Health and Personal Care	5-core (346,355 reviews)	ratings only (2,982,326 ratings)
Toys and Games	5-core (167,597 reviews)	ratings only (2,252,771 ratings)
Video Games	5-core (231,780 reviews)	ratings only (1,324,753 ratings)
Tools and Home Improvement	5-core (134,476 reviews)	ratings only (1,926,047 ratings)
Beauty	5-core (198,502 reviews)	ratings only (2,023,070 ratings)
Apps for Android	5-core (752,937 reviews)	ratings only (2,638,172 ratings)
Office Products	5-core (53,258 reviews)	ratings only (1,243,186 ratings)
Pet Supplies	5-core (157,836 reviews)	ratings only (1,235,316 ratings)
Automotive	5-core (20,473 reviews)	ratings only (1,373,768 ratings)
Grocery and Gourmet Food	5-core (151,254 reviews)	ratings only (1,297,156 ratings)
Patio, Lawn and Garden	5-core (13,272 reviews)	ratings only (993,490 ratings)
Baby	5-core (160,792 reviews)	ratings only (915,446 ratings)
Digital Music	5-core (64,706 reviews)	ratings only (836,006 ratings)
Musical Instruments	5-core (10,261 reviews)	ratings only (500,176 ratings)
Amazon Instant Video	5-core (37,126 reviews)	ratings only (583,933 ratings)

The dataset was downloaded in the csv format as below.



A	B	C	D	E	F	G	H	I	J	K	L	M	N
{"reviewerID": "A1HK2FQV	"asin": "097293751X"	"reviewerName": "Amanda	"helpful"	0]	"reviewText": "Perfect fo	sleep and dia	"overall":	"summar	"unixRevi	"reviewTi	2013"}		
{"reviewerID": "A19K65VY	"asin": "097293751X"	"reviewerName": "angela"	"helpful"	0]	"reviewText": "This book	answer pedia	or commu	then mov	and will f	"overall":	"summar	"unixRevi	"reviewTi
{"reviewerID": "A2LL1TGG	"asin": "097293751X"	"reviewerName": "Carter"	"helpful"	0]	"reviewText": "Helps me	"overall": 5.0	"summar	"unixRevi	"reviewTi	2014"}			
{"reviewerID": "A5G19RYX	"asin": "097293751X"	"reviewerName": "cfpurple	"helpful"	0]	"reviewText": "I bought th	it helped me	it help me	etc.Excell	"overall":	"summar	"unixRevi	"reviewTi	2013"}
{"reviewerID": "A2496A4E	"asin": "097293751X"	"reviewerName": "C. Jeter	"helpful"	0]	"reviewText": "I wanted a	and this has	but it's he	"overall":	"summar	"unixRevi	"reviewTi	2014"}	
{"reviewerID": "A3OQEVD	"asin": "097293751X"	"reviewerName": "CMB"	"helpful"	0]	"reviewText": "This is gre	but I wish the	"overall":	"summar	but not d	"unixRevi	"reviewTi	2014"}	
{"reviewerID": "ATZDT4B1	"asin": "097293751X"	"reviewerName": "HYM"	"helpful"	0]	"reviewText": "My 3 mont	"overall": 5.0	"summar	"unixRevi	"reviewTi	2013"}			
{"reviewerID": "A3NMPME	"asin": "097293751X"	"reviewerName": "Jakell"	"helpful"	3]	"reviewText": "This book	and this book	diaper ch	sleep. De	"overall":	"summar	"unixRevi	"reviewTi	2013"}
{"reviewerID": "A1ZSTU6R	"asin": "097293751X"	"reviewerName": "Jen"	"helpful"	0]	"reviewText": "I wanted t	but it was pre	then also	just not w	"overall":	"summar	but I liked	"unixRevi	"reviewTi
{"reviewerID": "A1TFH58B	"asin": "097293751X"	"reviewerName": "killerbe	"helpful"	0]	"reviewText": "The Baby T	feedings	diaper ch	activities	and notes	which tra	I bought t	which tra	it's nice to
{"reviewerID": "AKNT3ZH2	"asin": "097293751X"	"reviewerName": "LW"	"helpful"	0]	"reviewText": "During you	urination and	when you	could not	the baby	so I starte	who came	ect to kee	"overall":
{"reviewerID": "A3O4ATU0	"asin": "097293751X"	"reviewerName": "MAPN"	"helpful"	1]	"reviewText": "I use this s	but we just w	we have s	"overall":	"summar	"unixRevi	"reviewTi	2013"}	
{"reviewerID": "AXBWU2IA	"asin": "097293751X"	"reviewerName": "Mommy	"helpful"	0]	"reviewText": "This book	"overall": 4.0	"summar	"unixRevi	"reviewTi	2013"}			
{"reviewerID": "AOWBZDN	"asin": "097293751X"	"reviewerName": "onlygre	"helpful"	0]	"reviewText": "Has colum	"overall": 5.0	"summar	"unixRevi	"reviewTi	2014"}			
{"reviewerID": "A2SYNL4Y	"asin": "097293751X"	"reviewerName": "R. David	"helpful"	2]	"reviewText": "I like this I	but think it w	"overall":	"summar	"unixRevi	"reviewTi	2013"}		
{"reviewerID": "A2Q2A6JK	"asin": "097293751X"	"reviewerName": "R. Garre	"helpful"	2]	"reviewText": "My wife ar	nap-time	diaper ch	play-time	did baby l	how much	etc.There	however	some frus
{"reviewerID": "A3OL1DR5	"asin": "097293751X"	"reviewerName": "sfnewm	"helpful"	0]	"reviewText": "I thought k	and here's wh	it's often	or what si	but at lea	it can tell	"overall":	"summar	"unixRevi
{"reviewerID": "AF98RW6D	"asin": "9729375011"	"reviewerName": "Angel"	"helpful"	0]	"reviewText": "Easy to use	simple! I got	etc. But I'	"overall":	"summar	"unixRevi	"reviewTi	2011"}	
{"reviewerID": "A2VVPVI9	"asin": "9729375011"	"reviewerName": "AS"	"helpful"	0]	"reviewText": "We used t	"overall": 5.0	"summar	"unixRevi	"reviewTi	2013"}			
{"reviewerID": "A3PGZ7W5	"asin": "9729375011"	"reviewerName": "Casey T.	"helpful"	0]	"reviewText": "This item	especially wi	and even	"overall":	"summar	"unixRevi	"reviewTi	2014"}	
{"reviewerID": "A2EAJL3H6	"asin": "9729375011"	"reviewerName": "C. Mark	"helpful"	0]	"reviewText": "I've been	but even bet	&#34;Oh	she hasn'	then start	"overall":	"summar	"unixRevi	"reviewTi
{"reviewerID": "A16WT9L1	"asin": "9729375011"	"reviewerName": "coach"	"helpful"	0]	"reviewText": "Of course	though	is that it	and becau	"overall":	"summar	"unixRevi	"reviewTi	2013"}
{"reviewerID": "A2VUKGR1	"asin": "9729375011"	"reviewerName": "CoopJen	"helpful"	0]	"reviewText": "I've been	what breast	how long	etc. Plus	slept	or pottied	"overall":	"summar	"unixRevi
{"reviewerID": "A10GMDG	"asin": "9729375011"	"reviewerName": "Cori Puc	"helpful"	0]	"reviewText": "I didn't thi	&#34;when d	you could	but I liked	"overall":	"summar	"unixRevi	"reviewTi	2013"}

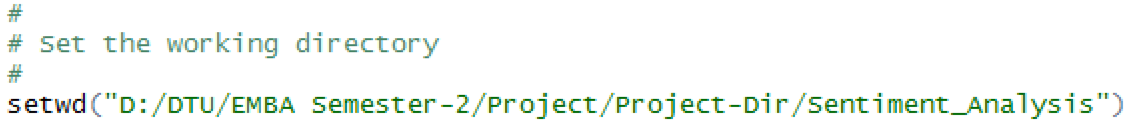
Review structure

- **reviewerID** :- ID of the reviewer, e.g. A1HK2FQW6KXQB2
- **asin** :- ID of the product, e.g. 097293751X
- **reviewerName** :- name of the reviewer
- **helpful** :- helpfulness rating of the review, e.g. 0,2,3
- **reviewText** :- text of the review
- **overall :-** rating of the product
- **summary -** summary of the review
- **unixReviewTime** – unix time of the review
- **reviewTime** – date and time of the review

6.2.2.2 Execute Installation code in R-Studio

Step 1: Set up the working directory

R Code <...>

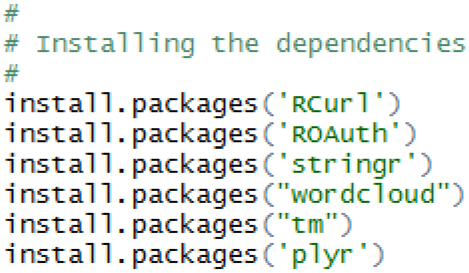


```
#
# Set the working directory
#
setwd("D:/DTU/EMBA Semester-2/Project/Project-Dir/Sentiment_Analysis")
```

For setting up the project's folder in the computer as the working directory. This will help RStudio to get and set the dependencies without explicitly specifying the folder's path all the time.

Step 2: Installing the packages

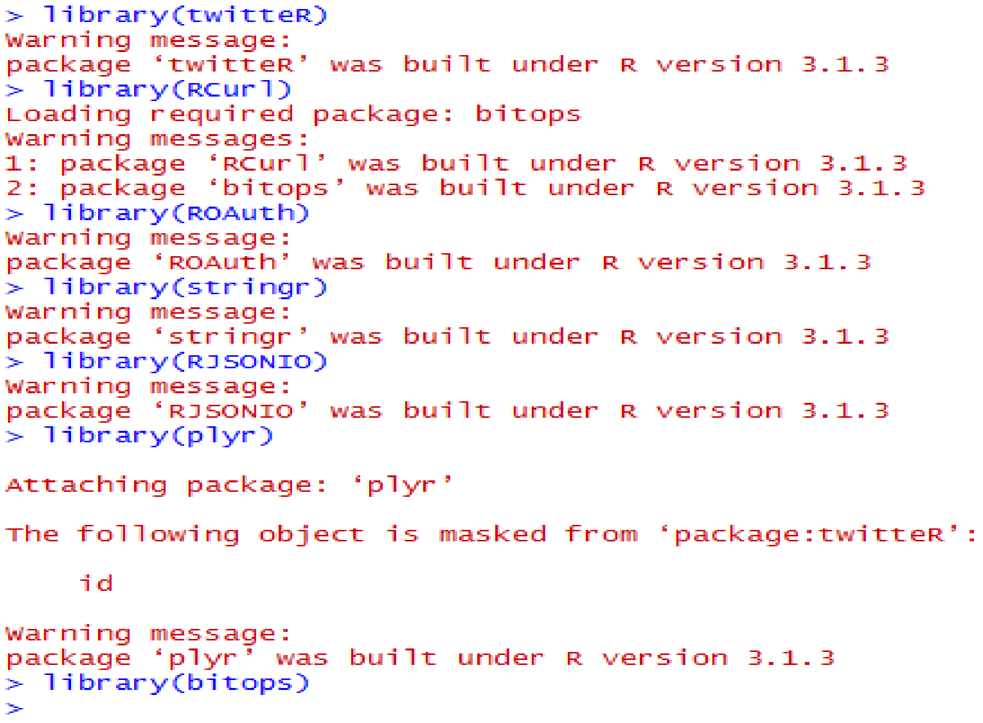
R Code <...>



```
#
# Installing the dependencies
#
install.packages('RCurl')
install.packages('ROAuth')
install.packages('stringr')
install.packages("wordcloud")
install.packages("tm")
install.packages('plyr')
```

After executing the script section, console will look like this shown below:

OUTPUT:



```
> library(twitteR)
Warning message:
package ‘twitteR’ was built under R version 3.1.3
> library(RCurl)
Loading required package: bitops
Warning messages:
1: package ‘RCurl’ was built under R version 3.1.3
2: package ‘bitops’ was built under R version 3.1.3
> library(ROAuth)
Warning message:
package ‘ROAuth’ was built under R version 3.1.3
> library(stringr)
Warning message:
package ‘stringr’ was built under R version 3.1.3
> library(RJSONIO)
Warning message:
package ‘RJSONIO’ was built under R version 3.1.3
> library(plyr)

Attaching package: ‘plyr’

The following object is masked from ‘package:twitteR’:

    id

Warning message:
package ‘plyr’ was built under R version 3.1.3
> library(bitops)
>
```

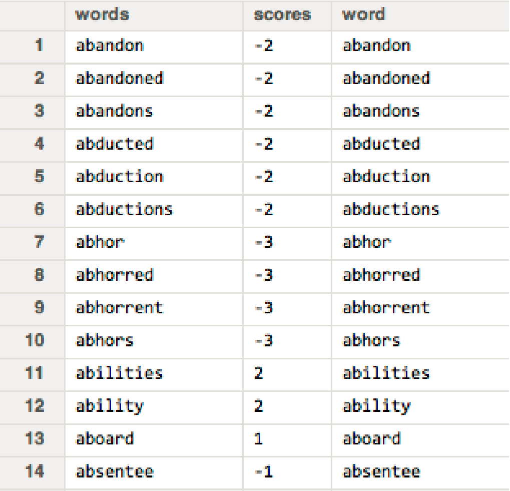
6.2.2.3 Standardize the data (Pre-processing)

As the dataset is from Amazon.com, the data is in the form of text. The text data is highly prone to inconsistencies. This step is very important as it extract out unwanted words from tweets. To make the data more relevant for analysis, text preprocessing is performed.

- **Tokenization**
- **Stopword Elimination**
- **Stemming**
- **Part-Of-Speech tagging**

6.2.2.4 Classify the Data

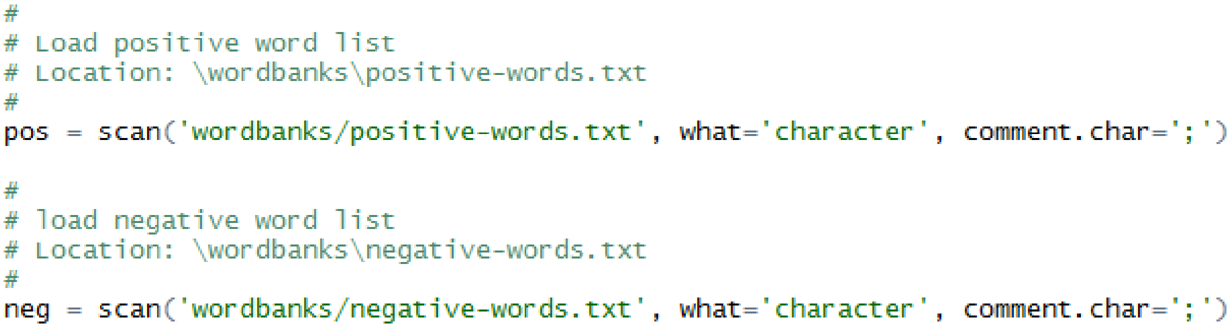
In order to analyze the sentiment in nature sentence, we need to build a sentiment word base, which contains a list of bad words and a list of good words. For this purpose the AFINN wordlist is used, which has 2477 words and phrases rated from -5, which present very negative viewpoint, to +5, which imply very positive attitudes. Due to the uncertainty and simplicity, the author of the word base did not consider any words as neutral because it is hard to define a word as neutral attitudes.



	words	scores	word
1	abandon	-2	abandon
2	abandoned	-2	abandoned
3	abandons	-2	abandons
4	abducted	-2	abducted
5	abduction	-2	abduction
6	abductions	-2	abductions
7	abhor	-3	abhor
8	abhorred	-3	abhorred
9	abhorrent	-3	abhorrent
10	abhors	-3	abhors
11	abilities	2	abilities
12	ability	2	ability
13	aboard	1	aboard
14	absentee	-1	absentee

Step 1: Loading the positive and the negative word list

R Code <...>

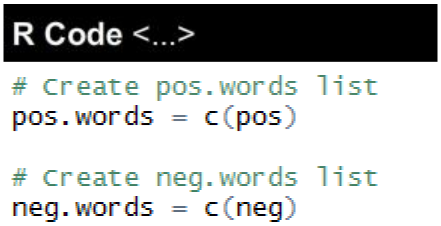


```
#
# Load positive word list
# Location: \wordbanks\positive-words.txt
#
pos = scan('wordbanks/positive-words.txt', what='character', comment.char=';')

#
# load negative word list
# Location: \wordbanks\negative-words.txt
#
neg = scan('wordbanks/negative-words.txt', what='character', comment.char=';')
```

We are using scan () method to read the .txt files.

We are reading the positive-words.txt from the specified location and similarly reading the negative-words.txt.

Step 2: Creating word list

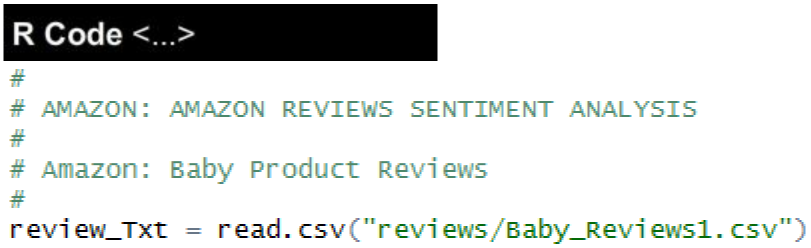


R Code <...>

```
# Create pos.words list
pos.words = c(pos)

# Create neg.words list
neg.words = c(neg)
```

In this step we are creating the word list using c () method.

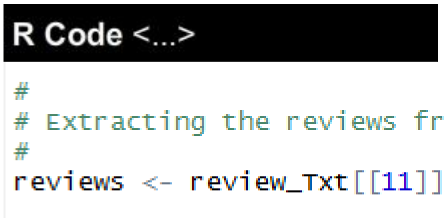
Step 3: Reading Amazon Reviews of Baby products csv file



R Code <...>

```
#
# AMAZON: AMAZON REVIEWS SENTIMENT ANALYSIS
#
# Amazon: Baby Product Reviews
#
review_Txt = read.csv("reviews/Baby_Reviews1.csv")
```

Step 4: Extracting Amazon Reviews from csv file



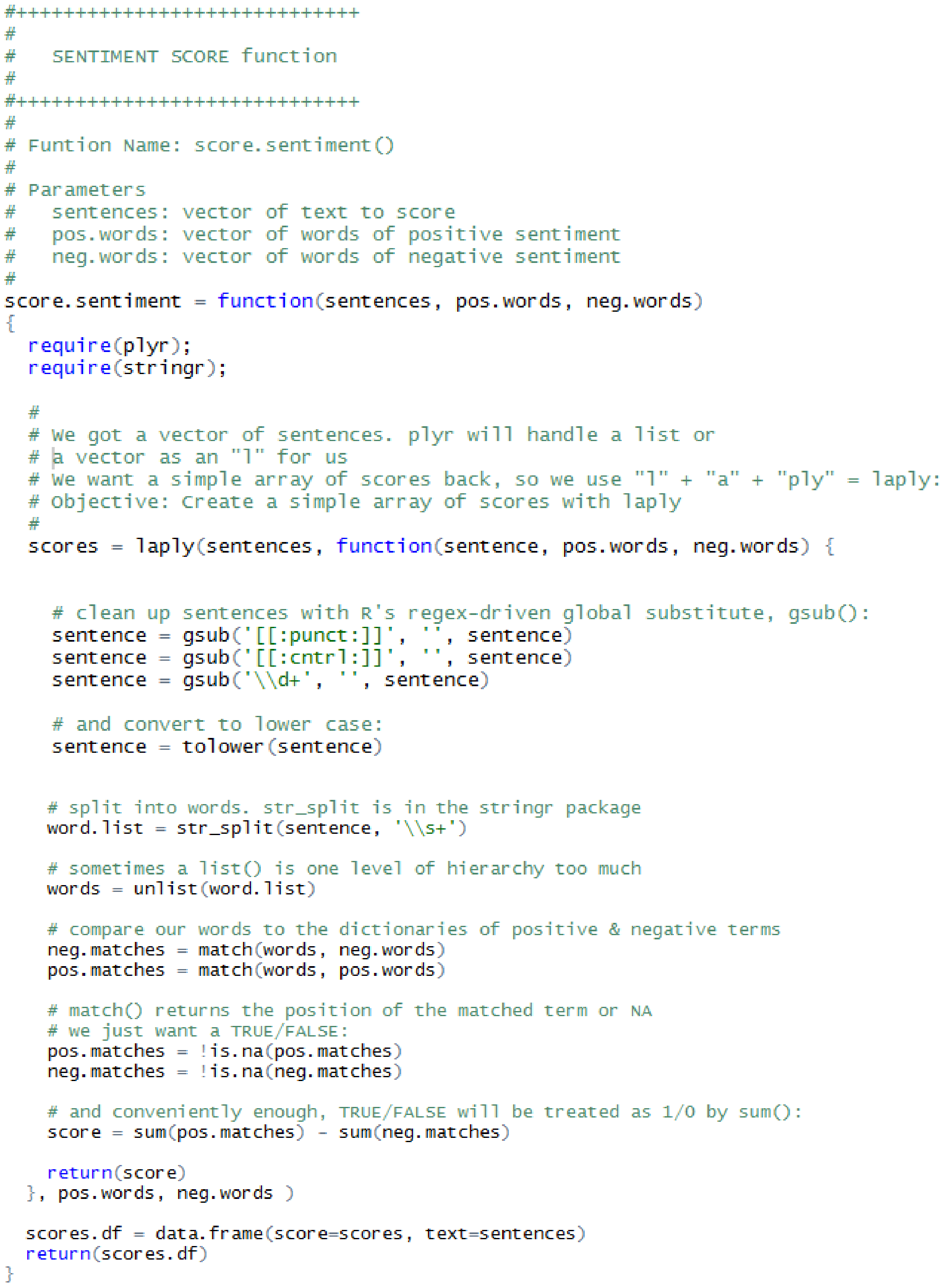
R Code <...>

```
#
# Extracting the reviews from the csv file
#
reviews <- review_Txt[[11]]
```

6.2.2.5 Run R commands for processing the Data file

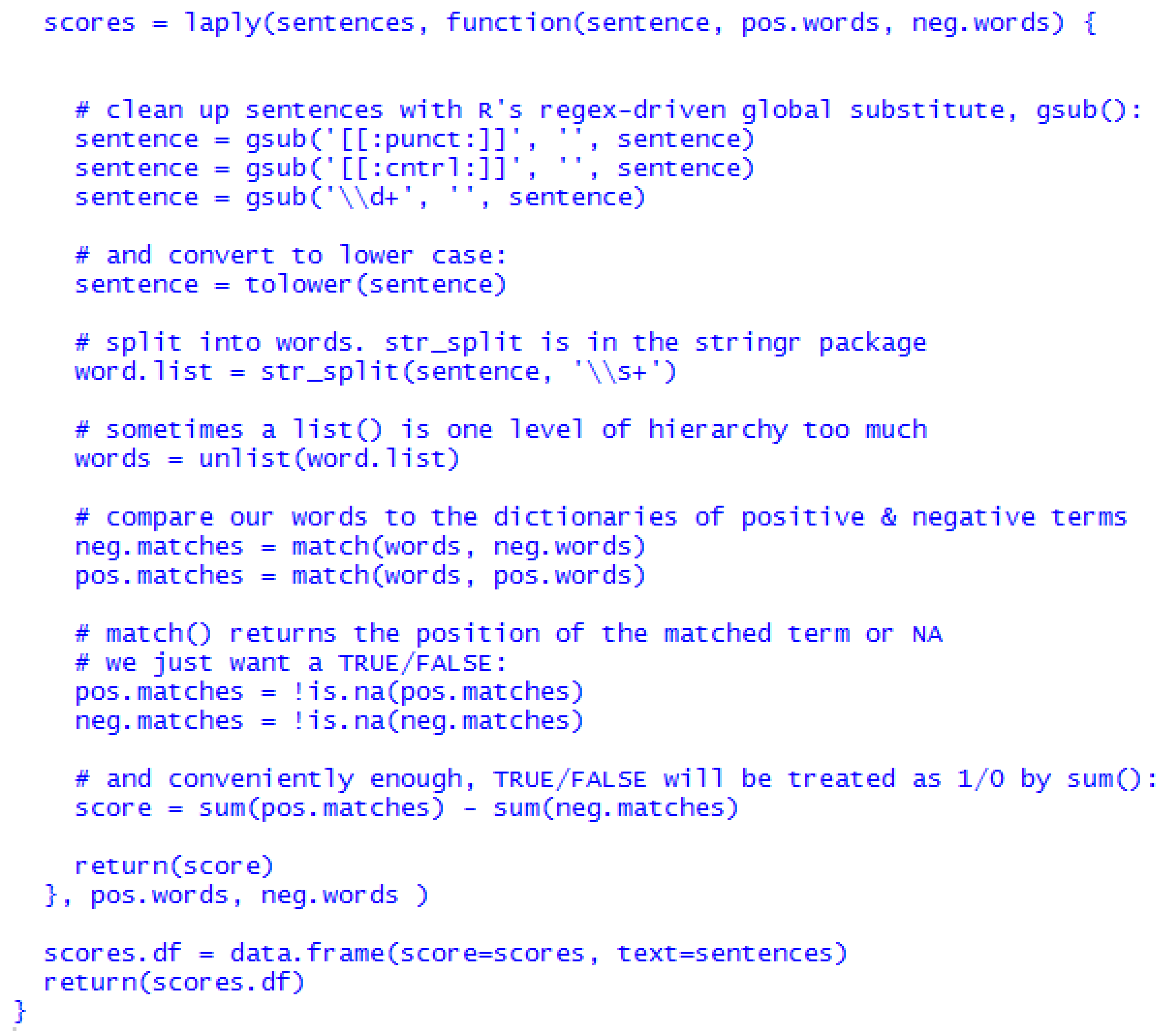
Step 1: Sentiment Score Logic

R Code <...>



```
#++++++++++++++++++++++++++++++
#
#   SENTIMENT SCORE function
#
#++++++++++++++++++++++++++++++
#
# Funtion Name: score.sentiment()
#
# Parameters
#   sentences: vector of text to score
#   pos.words: vector of words of positive sentiment
#   neg.words: vector of words of negative sentiment
#
score.sentiment = function(sentences, pos.words, neg.words)
{
  require(plyr);
  require(stringr);

  #
  # We got a vector of sentences. plyr will handle a list or
  # a vector as an "l" for us
  # We want a simple array of scores back, so we use "l" + "a" + "ply" = laply:
  # Objective: Create a simple array of scores with laply
  #
  scores = laply(sentences, function(sentence, pos.words, neg.words) {


    # clean up sentences with R's regex-driven global substitute, gsub():
    sentence = gsub('[[:punct:]]', '', sentence)
    sentence = gsub('[[:cntrl:]]', '', sentence)
    sentence = gsub('\\d+', '', sentence)

    # and convert to lower case:
    sentence = tolower(sentence)


    # split into words. str_split is in the stringr package
    word.list = str_split(sentence, '\\s+')

    # sometimes a list() is one level of hierarchy too much
    words = unlist(word.list)

    # compare our words to the dictionaries of positive & negative terms
    neg.matches = match(words, neg.words)
    pos.matches = match(words, pos.words)

    # match() returns the position of the matched term or NA
    # we just want a TRUE/FALSE:
    pos.matches = !is.na(pos.matches)
    neg.matches = !is.na(neg.matches)

    # and conveniently enough, TRUE/FALSE will be treated as 1/0 by sum():
    score = sum(pos.matches) - sum(neg.matches)

    return(score)
  }, pos.words, neg.words )

  scores.df = data.frame(score=scores, text=sentences)
  return(scores.df)
}
```

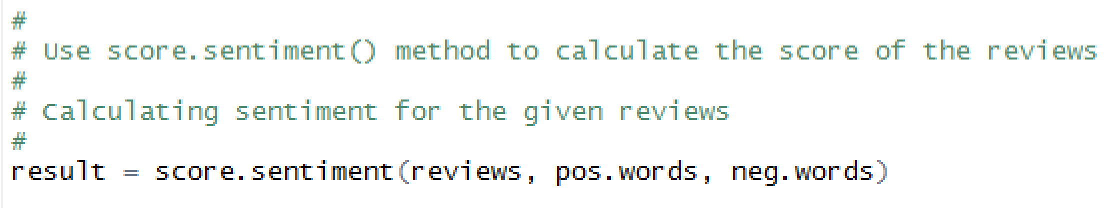
OUTPUT:



```
  scores = laply(sentences, function(sentence, pos.words, neg.words) {

    # clean up sentences with R's regex-driven global substitute, gsub():
    sentence = gsub('[[:punct:]]', '', sentence)
    sentence = gsub('[[:cntrl:]]', '', sentence)
    sentence = gsub('\\d+', '', sentence)

    # and convert to lower case:
    sentence = tolower(sentence)

    # split into words. str_split is in the stringr package
    word.list = str_split(sentence, '\\s+')

    # sometimes a list() is one level of hierarchy too much
    words = unlist(word.list)

    # compare our words to the dictionaries of positive & negative terms
    neg.matches = match(words, neg.words)
    pos.matches = match(words, pos.words)

    # match() returns the position of the matched term or NA
    # we just want a TRUE/FALSE:
    pos.matches = !is.na(pos.matches)
    neg.matches = !is.na(neg.matches)

    # and conveniently enough, TRUE/FALSE will be treated as 1/0 by sum():
    score = sum(pos.matches) - sum(neg.matches)

    return(score)
  }, pos.words, neg.words )

  scores.df = data.frame(score=scores, text=sentences)
  return(scores.df)
}
```

Step 2: Sentiment Score Calculation

R Code <...>



```
#
# Use score.sentiment() method to calculate the score of the reviews
#
# Calculating sentiment for the given reviews
#
result = score.sentiment(reviews, pos.words, neg.words)
```

OUTPUT:

```
> rownames(result)
  [1] "1"   "2"   "3"   "4"   "5"   "6"   "7"   "8"   "9"   "10"  "11"  "12"
 [13] "13"  "14"  "15"  "16"  "17"  "18"  "19"  "20"  "21"  "22"  "23"  "24"
 [25] "25"  "26"  "27"  "28"  "29"  "30"  "31"  "32"  "33"  "34"  "35"  "36"
 [37] "37"  "38"  "39"  "40"  "41"  "42"  "43"  "44"  "45"  "46"  "47"  "48"
 [49] "49"  "50"  "51"  "52"  "53"  "54"  "55"  "56"  "57"  "58"  "59"  "60"
 [61] "61"  "62"  "63"  "64"  "65"  "66"  "67"  "68"  "69"  "70"  "71"  "72"
 [73] "73"  "74"  "75"  "76"  "77"  "78"  "79"  "80"  "81"  "82"  "83"  "84"
 [85] "85"  "86"  "87"  "88"  "89"  "90"  "91"  "92"  "93"  "94"  "95"  "96"
 [97] "97"  "98"  "99"  "100" "101" "102" "103" "104" "105" "106" "107" "108"
[109] "109" "110" "111" "112" "113" "114" "115" "116" "117" "118" "119" "120"
[121] "121" "122" "123" "124" "125" "126" "127" "128" "129" "130" "131" "132"
[133] "133" "134" "135" "136" "137" "138" "139" "140" "141" "142" "143" "144"
[145] "145" "146" "147" "148" "149" "150" "151" "152" "153" "154" "155" "156"
[157] "157" "158" "159" "160" "161" "162" "163" "164" "165" "166" "167" "168"
[169] "169" "170" "171" "172" "173" "174" "175" "176" "177" "178" "179" "180"
[181] "181" "182" "183" "184" "185" "186" "187" "188" "189" "190" "191" "192"
[193] "193" "194" "195" "196" "197" "198" "199" "200" "201" "202" "203" "204"
[205] "205" "206" "207" "208" "209" "210" "211" "212" "213" "214" "215" "216"
[217] "217" "218" "219" "220" "221" "222" "223" "224" "225" "226" "227" "228"
[229] "229" "230" "231" "232" "233" "234" "235" "236" "237" "238" "239" "240"
[241] "241" "242" "243" "244" "245" "246" "247" "248" "249" "250" "251" "252"
[253] "253" "254" "255" "256" "257" "258" "259" "260" "261" "262" "263" "264"
[265] "265" "266" "267" "268" "269" "270" "271" "272" "273" "274" "275" "276"
[277] "277" "278" "279" "280" "281" "282" "283" "284" "285" "286" "287" "288"
[289] "289" "290" "291" "292" "293" "294" "295" "296" "297" "298" "299" "300"
[301] "301" "302" "303" "304" "305" "306" "307" "308" "309" "310" "311" "312"
[313] "313" "314" "315" "316" "317" "318" "319" "320" "321" "322" "323" "324"
[325] "325" "326" "327" "328" "329" "330" "331" "332" "333" "334" "335" "336"
[337] "337" "338" "339" "340" "341" "342" "343" "344" "345" "346" "347" "348"
[349] "349" "350" "351" "352" "353" "354" "355" "356" "357" "358" "359" "360"
[361] "361" "362" "363" "364" "365" "366" "367" "368" "369" "370" "371" "372"
[373] "373" "374" "375" "376" "377" "378" "379" "380" "381" "382" "383" "384"
[385] "385" "386" "387" "388" "389" "390" "391" "392" "393" "394" "395" "396"
[397] "397" "398" "399" "400" "401" "402" "403" "404" "405" "406" "407" "408"
```

```
[409] "409"  "410"  "411"  "412"  "413"  "414"  "415"  "416"  "417"  "418"  "419"  "420"
[421] "421"  "422"  "423"  "424"  "425"  "426"  "427"  "428"  "429"  "430"  "431"  "432"
[433] "433"  "434"  "435"  "436"  "437"  "438"  "439"  "440"  "441"  "442"  "443"  "444"
[445] "445"  "446"  "447"  "448"  "449"  "450"  "451"  "452"  "453"  "454"  "455"  "456"
[457] "457"  "458"  "459"  "460"  "461"  "462"  "463"  "464"  "465"  "466"  "467"  "468"
[469] "469"  "470"  "471"  "472"  "473"  "474"  "475"  "476"  "477"  "478"  "479"  "480"
[481] "481"  "482"  "483"  "484"  "485"  "486"  "487"  "488"  "489"  "490"  "491"  "492"
[493] "493"  "494"  "495"  "496"  "497"  "498"  "499"  "500"  "501"  "502"  "503"  "504"
[505] "505"  "506"  "507"  "508"  "509"  "510"  "511"  "512"  "513"  "514"  "515"  "516"
[517] "517"  "518"  "519"  "520"  "521"  "522"  "523"  "524"  "525"  "526"  "527"  "528"
[529] "529"  "530"  "531"  "532"  "533"  "534"  "535"  "536"  "537"  "538"  "539"  "540"
[541] "541"  "542"  "543"  "544"  "545"  "546"  "547"  "548"  "549"  "550"  "551"  "552"
[553] "553"  "554"  "555"  "556"  "557"  "558"  "559"  "560"  "561"  "562"  "563"  "564"
[565] "565"  "566"  "567"  "568"  "569"  "570"  "571"  "572"  "573"  "574"  "575"  "576"
[577] "577"  "578"  "579"  "580"  "581"  "582"  "583"  "584"  "585"  "586"  "587"  "588"
[589] "589"  "590"  "591"  "592"  "593"  "594"  "595"  "596"  "597"  "598"  "599"  "600"
[601] "601"  "602"  "603"  "604"  "605"  "606"  "607"  "608"  "609"  "610"  "611"  "612"
[613] "613"  "614"  "615"  "616"  "617"  "618"  "619"  "620"  "621"  "622"  "623"  "624"
[625] "625"  "626"  "627"  "628"  "629"  "630"  "631"  "632"  "633"  "634"  "635"  "636"
[637] "637"  "638"  "639"  "640"  "641"  "642"  "643"  "644"  "645"  "646"  "647"  "648"
[649] "649"  "650"  "651"  "652"  "653"  "654"  "655"  "656"  "657"  "658"  "659"  "660"
[661] "661"  "662"  "663"  "664"  "665"  "666"  "667"  "668"  "669"  "670"  "671"  "672"
[673] "673"  "674"  "675"  "676"  "677"  "678"  "679"  "680"  "681"  "682"  "683"  "684"
[685] "685"  "686"  "687"  "688"  "689"  "690"  "691"  "692"  "693"  "694"  "695"  "696"
[697] "697"  "698"  "699"  "700"  "701"  "702"  "703"  "704"  "705"  "706"  "707"  "708"
[709] "709"  "710"  "711"  "712"  "713"  "714"  "715"  "716"  "717"  "718"  "719"  "720"
[721] "721"  "722"  "723"  "724"  "725"  "726"  "727"  "728"  "729"  "730"  "731"  "732"
[733] "733"  "734"  "735"  "736"  "737"  "738"  "739"  "740"  "741"  "742"  "743"  "744"
[745] "745"  "746"  "747"  "748"  "749"  "750"  "751"  "752"  "753"  "754"  "755"  "756"
[757] "757"  "758"  "759"  "760"  "761"  "762"  "763"  "764"  "765"  "766"  "767"  "768"
[769] "769"  "770"  "771"  "772"  "773"  "774"  "775"  "776"  "777"  "778"  "779"  "780"
[781] "781"  "782"  "783"  "784"  "785"  "786"  "787"  "788"  "789"  "790"  "791"  "792"
[793] "793"  "794"  "795"  "796"  "797"  "798"  "799"  "800"  "801"  "802"  "803"  "804"
[805] "805"  "806"  "807"  "808"  "809"  "810"  "811"  "812"  "813"  "814"  "815"  "816"
[817] "817"  "818"  "819"  "820"  "821"  "822"  "823"  "824"  "825"  "826"  "827"  "828"
[829] "829"  "830"  "831"  "832"  "833"  "834"  "835"  "836"  "837"  "838"  "839"  "840"
[841] "841"  "842"  "843"  "844"  "845"  "846"  "847"  "848"  "849"  "850"  "851"  "852"
```

```
[853] "853"  "854"  "855"  "856"  "857"  "858"  "859"  "860"  "861"  "862"  "863"  "864"
[865] "865"  "866"  "867"  "868"  "869"  "870"  "871"  "872"  "873"  "874"  "875"  "876"
[877] "877"  "878"  "879"  "880"  "881"  "882"  "883"  "884"  "885"  "886"  "887"  "888"
[889] "889"  "890"  "891"  "892"  "893"  "894"  "895"  "896"  "897"  "898"  "899"  "900"
[901] "901"  "902"  "903"  "904"  "905"  "906"  "907"  "908"  "909"  "910"  "911"  "912"
[913] "913"  "914"  "915"  "916"  "917"  "918"  "919"  "920"  "921"  "922"  "923"  "924"
[925] "925"  "926"  "927"  "928"  "929"  "930"  "931"  "932"  "933"  "934"  "935"  "936"
[937] "937"  "938"  "939"  "940"  "941"  "942"  "943"  "944"  "945"  "946"  "947"  "948"
[949] "949"  "950"  "951"  "952"  "953"  "954"  "955"  "956"  "957"  "958"  "959"  "960"
[961] "961"  "962"  "963"  "964"  "965"  "966"  "967"  "968"  "969"  "970"  "971"  "972"
[973] "973"  "974"  "975"  "976"  "977"  "978"  "979"  "980"  "981"  "982"  "983"  "984"
[985] "985"  "986"  "987"  "988"  "989"  "990"  "991"  "992"  "993"  "994"  "995"  "996"
[997] "997"  "998"  "999"  "1000"
```

Calculated Sentiment Score

View 1:

```
> result$score
  [1]  2  2  3  0  1  3  1  1  2  0  0  2  1  1  2  1  1  3  1  1  1  2  0  8  1  2  1  1 -2  2
 [31]  5  1 -1 -2  1  0 -1  1  0  1  1  2  1  3  1  1  2  1  0 -1  1  3  4  4  1  1  0  0  5  2
 [61]  0  4  2  1  1  1  0  3  1  4  1  0  0 -1 -1 -2  1  0  4  1 -1  1  2  1  0  1  1  3  1  0
 [91]  2 -5 -2  1  0  4  0  0 -2  1  2  1  0  1 -4  2  2 -1  1  1 -2  2  1  2  2 -1  1  1  0  1
[121]  1  2 -1  0  3  1  1 -1  0 -1 -1  3  5  2  0 -1  0  1  4  3  4  0  0  2  0 -2  4  2  1  0
[151]  0  0  0  0  1  0 -1 -8  0  0 -1  1  2  3 -3 -1  1  0  0  0  2  0 -3  1  1  0  2  1 -2 -1
[181]  0  1  1  1  0  1 -7  1  1 -3  2  4  0 -4  3  1  1 -1  2  0  2  1  2  1  1  1  3  2  1  3
[211] -2  3  1 -2  1  1 -2  3  0 -1  0  0 -4  0 -1  2 -8  0  2  1  0  0 -2  1 -1 -4  1  0  0  1
[241]  0  2  0  2  1  3  0  1  2  0 -2  3  2  0 -1  1  1  1 -3 -3  1  1  1  1  3  2  0 -3  2  1
[271] -2  2  0  0  1  1  5  2  2  2  2  0  2  2 -1  2  3  1  1  1  0 -2 -3  2  1  2  1 -2  1  0
[301] -1 -1  1  2  0 -1  0  0 -1 -2 -3 -2  3  3  2  5 -2 -1  0  1 -2  3 -1  1  1  1 -2  3  0  0
[331]  0  4  1  0  2  0  1 -1  3  0  1  2  2  0  2  1  4  0  3  0 -1  0  0  2  3  2  2  0  0  0
[361]  1  0  2  0  1 -1  3  4  2  1 -2  3  0  2  0  0  1  0  2  0  2  0  0 -1 -1  4  0  3  3  3
[391]  0  1 -1 -1  0  1  0  2  3  1  0  0  1  0  1  2  0  2 -1  1  0  2 -5 -3 -1 -2  0  0  1 -1
[421] -4  0  1  1 -2 -1  0 -4 -1  1  0 -1  0  4  0  0  0  3  1  0  0 -2 -1 -3 -1  0  2  3  2  0
[451]  0  0 -2  2  0  0  0 -1  4  1  0  1  1  3  2  0  2  0 -1 -2  2  0 -1  0  0  3  0  2  1  1
[481]  2  0  1  2  1  0  0  2  4  1  0  1  2 -1  0 -2  0  0  2  1  0  0  1  0 -1  1  2  0  0  1
[511]  1  0  0 -1  0  3  0  1 -1 -2  0  1  1  0  0  2  0  0  5  1  1 -2  1  1  1  1  1  0  6 -1
[541]  1  0 -1  2  0  1  1 -1  1  0 -1  1  0  0  0  0 -1  0 -1  0 -2  2  0  0  2 -1  0  1 -3  2
[571]  0  0  1  0  0  0 -3  5  1  0  0 -2  1 -2 -3  0  2 -2 -3 -1  0 -2 -2  2  0  0 -1  1 -1 -1
[601]  0 -1  0 -2  3  1 -1 -1  1  0  3  1  1 -3  2  0  2 -2 -1 -2  1  1  0 -3  1 -1  0  0  0  6
[631]  1  1  5  0 -3  0  0  2  0  0  6  0  3 -1  2  2  2  1  3  1  4  2  3  1 -1  0  1 -1  2  1
[661]  1  6 -2  1  0 -1  0  1  1 -1  0  2 -1  0  2  2  1  2  1  1  0  0  1  0  0  4  3  1  3  1
[691]  1  3  0  1  2  1 -1  2  6  2 -1  1  1 -2  0  2  1  3  3  0  2 -1  1  2 -1  1  0  3  2  3
[721]  1  1  0  0  0  5 -2  3  0  1  1  2  4  1  0  3  1 -2 -6  1  3  1  1  0 -1  3  4 -3  0  0
[751]  1  0  3  0  0  1  8  2  2  1  2  3  0  4  3  2  4  0  4  1  3  5  2  0  1  2  2  1  0  0
[781]  7  4  1  3  0  1  0  3  1  0  2  0  0  0  4  1  1  2  1  1  0  3  0  0  1  0  6 -1 -2  0
[811]  4  0 -1 -1  0  1  0  1  0  0  4  5 -2 -4  4  0 -1  0  1  0 -1  1  0  0  1 10  0  3  3  1
[841]  0  0  1  1  2 -1  0  2  0  1  1  0  1  1  4  2 -1  0  0  4  0  2  0  1  0  0  0  3  1  0
[871]  0  2  0 -1  1  0  3  1  1  2 -1  0  0  0  0  0  2  1  0  1  0  0 -2  0  0  4  3  1  3  0
```

```
[901]  2  0  1  1  1  2  0  0  0  4 -2  0  0  0  1  9  2  2  4  1  1  2  1  1  3  1  3  2  4  4
[931]  1  1  2 -1  6  1  1  2  0  5  5  1  0  2  1  3  5  1  0  4  0  2  0  1  6 -1 -1  1  2  6
[961]  2  5  8  3  0  4  2  2  1  2  0  4  2  0  2  7  2  2  0  2  1  1  0  1  2 -2  0 -1  0  1
[991]  2 -1  0  1  0  0  1  2 -2
```

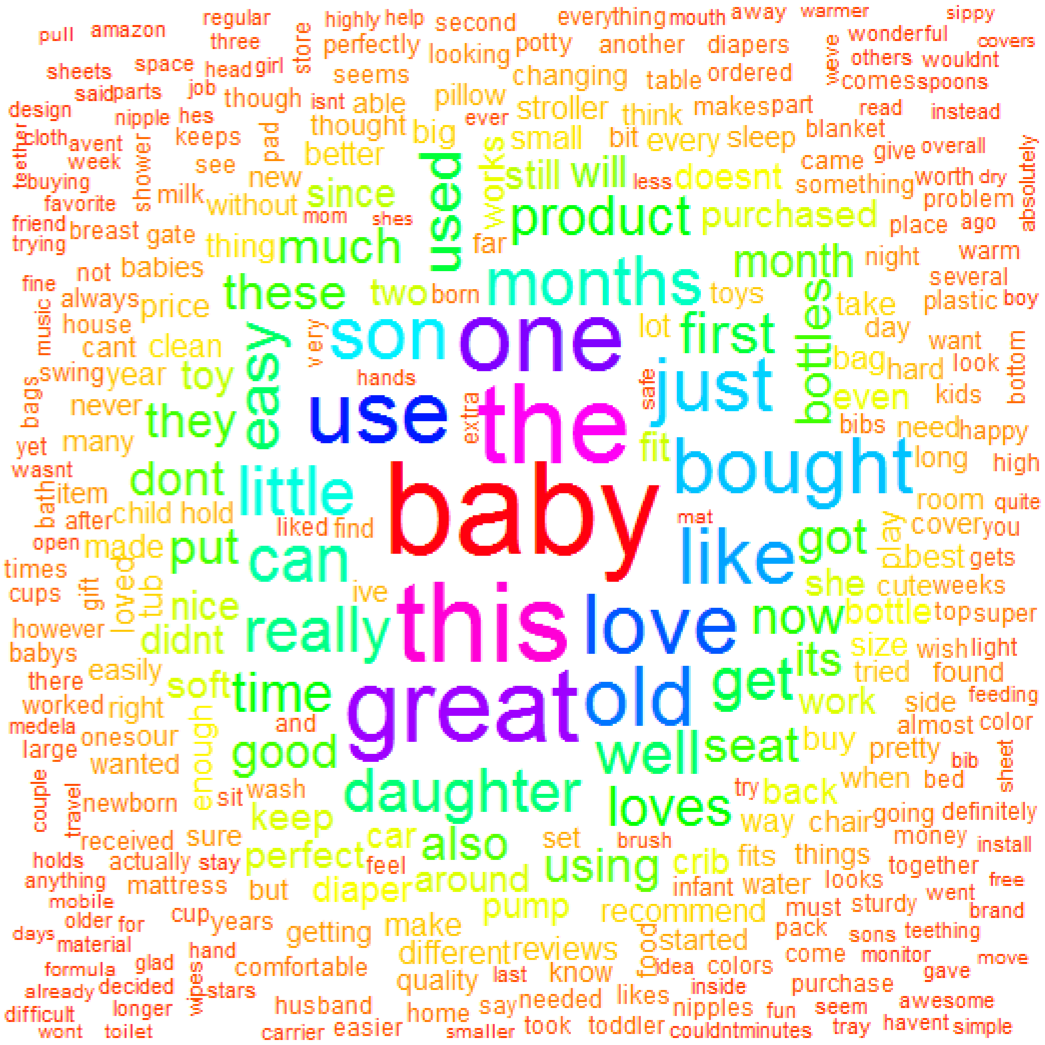
View 2:

```
> result[,1]
  [1]  2  2  3  0  1  3  1  1  2  0  0  2  1  1  2  1  1  3  1  1  1  2  0  8  1  2  1  1 -2  2
 [31]  5  1 -1 -2  1  0 -1  1  0  1  1  2  1  3  1  1  2  1  0 -1  1  3  4  4  1  1  0  0  5  2
 [61]  0  4  2  1  1  1  0  3  1  4  1  0  0 -1 -1 -2  1  0  4  1 -1  1  2  1  0  1  1  3  1  0
 [91]  2 -5 -2  1  0  4  0  0 -2  1  2  1  0  1 -4  2  2 -1  1  1 -2  2  1  2  2 -1  1  1  0  1
[121]  1  2 -1  0  3  1  1 -1  0 -1 -1  3  5  2  0 -1  0  1  4  3  4  0  0  2  0 -2  4  2  1  0
[151]  0  0  0  0  1  0 -1 -8  0  0 -1  1  2  3 -3 -1  1  0  0  0  2  0 -3  1  1  0  2  1 -2 -1
[181]  0  1  1  1  0  1 -7  1  1 -3  2  4  0 -4  3  1  1 -1  2  0  2  1  2  1  1  1  3  2  1  3
[211] -2  3  1 -2  1  1 -2  3  0 -1  0  0 -4  0 -1  2 -8  0  2  1  0  0 -2  1 -1 -4  1  0  0  1
[241]  0  2  0  2  1  3  0  1  2  0 -2  3  2  0 -1  1  1  1 -3 -3  1  1  1  1  3  2  0 -3  2  1
[271] -2  2  0  0  1  1  5  2  2  2  2  0  2  2 -1  2  3  1  1  1  0 -2 -3  2  1  2  1 -2  1  0
[301] -1 -1  1  2  0 -1  0  0 -1 -2 -3 -2  3  3  2  5 -2 -1  0  1 -2  3 -1  1  1  1 -2  3  0  0
[331]  0  4  1  0  2  0  1 -1  3  0  1  2  2  0  2  1  4  0  3  0 -1  0  0  2  3  2  2  0  0  0
[361]  1  0  2  0  1 -1  3  4  2  1 -2  3  0  2  0  0  1  0  2  0  2  0  0 -1 -1  4  0  3  3  3
[391]  0  1 -1 -1  0  1  0  2  3  1  0  0  1  0  1  2  0  2 -1  1  0  2 -5 -3 -1 -2  0  0  1 -1
[421] -4  0  1  1 -2 -1  0 -4 -1  1  0 -1  0  4  0  0  0  3  1  0  0 -2 -1 -3 -1  0  2  3  2  0
[451]  0  0 -2  2  0  0  0 -1  4  1  0  1  1  3  2  0  2  0 -1 -2  2  0 -1  0  0  3  0  2  1  1
[481]  2  0  1  2  1  0  0  2  4  1  0  1  2 -1  0 -2  0  0  2  1  0  0  1  0 -1  1  2  0  0  1
[511]  1  0  0 -1  0  3  0  1 -1 -2  0  1  1  0  0  2  0  0  5  1  1 -2  1  1  1  1  1  0  6 -1
[541]  1  0 -1  2  0  1  1 -1  1  0 -1  1  0  0  0  0 -1  0 -1  0 -2  2  0  0  2 -1  0  1 -3  2
[571]  0  0  1  0  0  0 -3  5  1  0  0 -2  1 -2 -3  0  2 -2 -3 -1  0 -2 -2  2  0  0 -1  1 -1 -1
[601]  0 -1  0 -2  3  1 -1 -1  1  0  3  1  1 -3  2  0  2 -2 -1 -2  1  1  0 -3  1 -1  0  0  0  6
[631]  1  1  5  0 -3  0  0  2  0  0  6  0  3 -1  2  2  2  1  3  1  4  2  3  1 -1  0  1 -1  2  1
[661]  1  6 -2  1  0 -1  0  1  1 -1  0  2 -1  0  2  2  1  2  1  1  0  0  1  0  0  4  3  1  3  1
[691]  1  3  0  1  2  1 -1  2  6  2 -1  1  1 -2  0  2  1  3  3  0  2 -1  1  2 -1  1  0  3  2  3
[721]  1  1  0  0  0  5 -2  3  0  1  1  2  4  1  0  3  1 -2 -6  1  3  1  1  0 -1  3  4 -3  0  0
[751]  1  0  3  0  0  1  8  2  2  1  2  3  0  4  3  2  4  0  4  1  3  5  2  0  1  2  2  1  0  0
[781]  7  4  1  3  0  1  0  3  1  0  2  0  0  0  4  1  1  2  1  1  0  3  0  0  1  0  6 -1 -2  0
[811]  4  0 -1 -1  0  1  0  1  0  0  4  5 -2 -4  4  0 -1  0  1  0 -1  1  0  0  1 10  0  3  3  1
[841]  0  0  1  1  2 -1  0  2  0  1  1  0  1  1  4  2 -1  0  0  4  0  2  0  1  0  0  0  3  1  0
[871]  0  2  0 -1  1  0  3  1  1  2 -1  0  0  0  0  0  2  1  0  1  0  0 -2  0  0  4  3  1  3  0
[901]  2  0  1  1  1  2  0  0  0  4 -2  0  0  0  1  9  2  2  4  1  1  2  1  1  3  1  3  2  4  4
[931]  1  1  2 -1  6  1  1  2  0  5  5  1  0  2  1  3  5  1  0  4  0  2  0  1  6 -1 -1  1  2  6
[961]  2  5  8  3  0  4  2  2  1  2  0  4  2  0  2  7  2  2  0  2  1  1  0  1  2 -2  0 -1  0  1
[991]  2 -1  0  1  0  0  1  2 -2  1
```

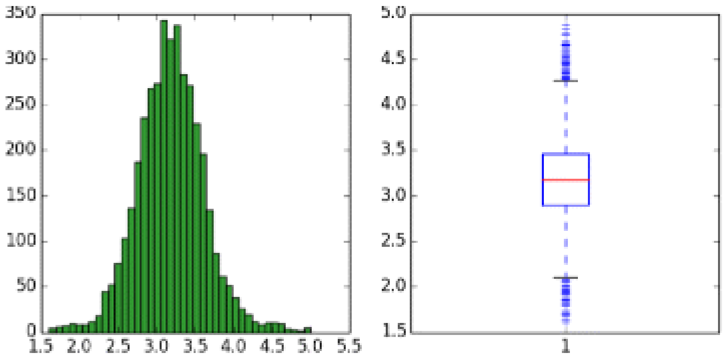
6.2.2.6 Establish R Plotter, Histogram and Wordcloud

The process of sentiment analysis is to calculate the synchronization of the words of the tweets with respect

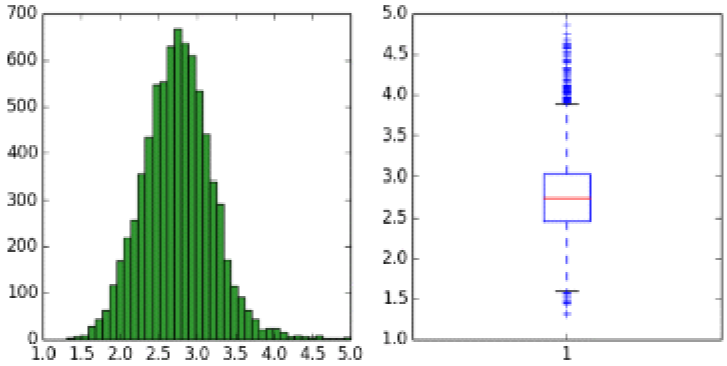
<u>Word Cloud for Amazon Baby product Reviews</u>



Sentiment score information for word tokens Positive word tokens

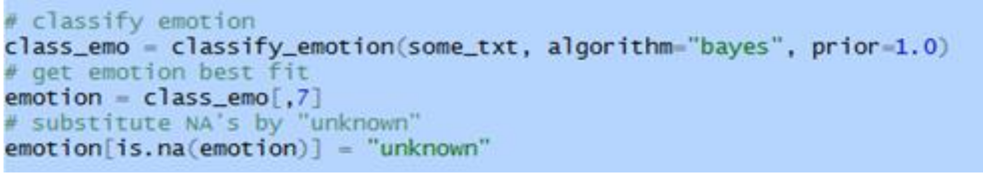


Sentiment score information for word tokens Negative word tokens



Calculating Emotions

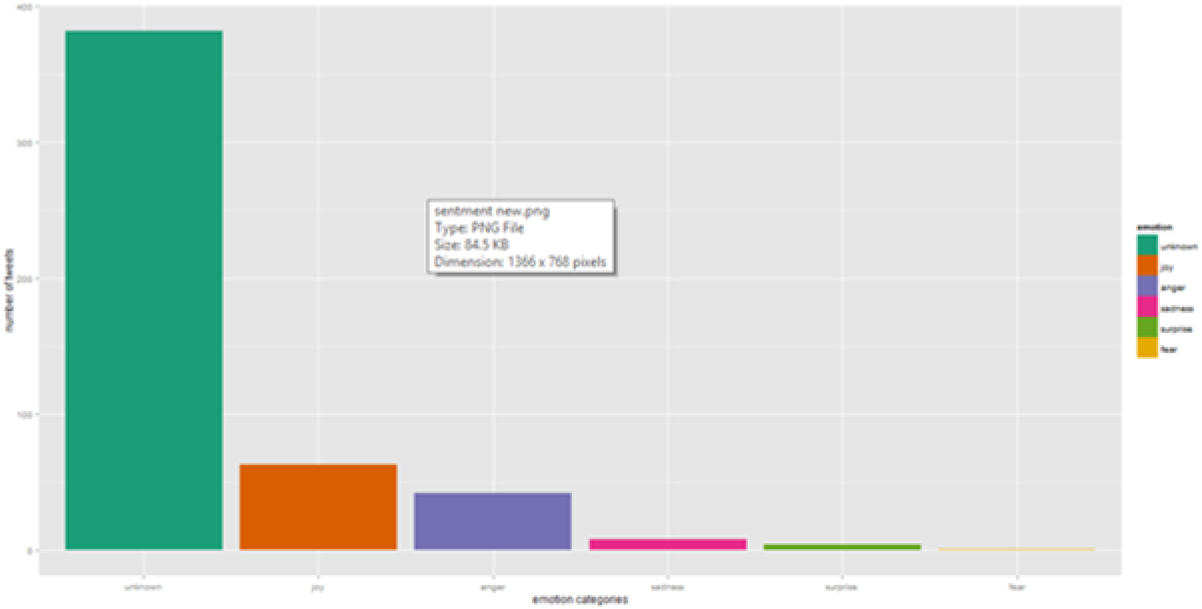
R Code <...>



```
# classify emotion
class_emo = classify_emotion(some_txt, algorithm="bayes", prior=1.0)
# get emotion best fit
emotion = class_emo[,7]
# substitute NA's by "unknown"
emotion[is.na(emotion)] = "unknown"
```

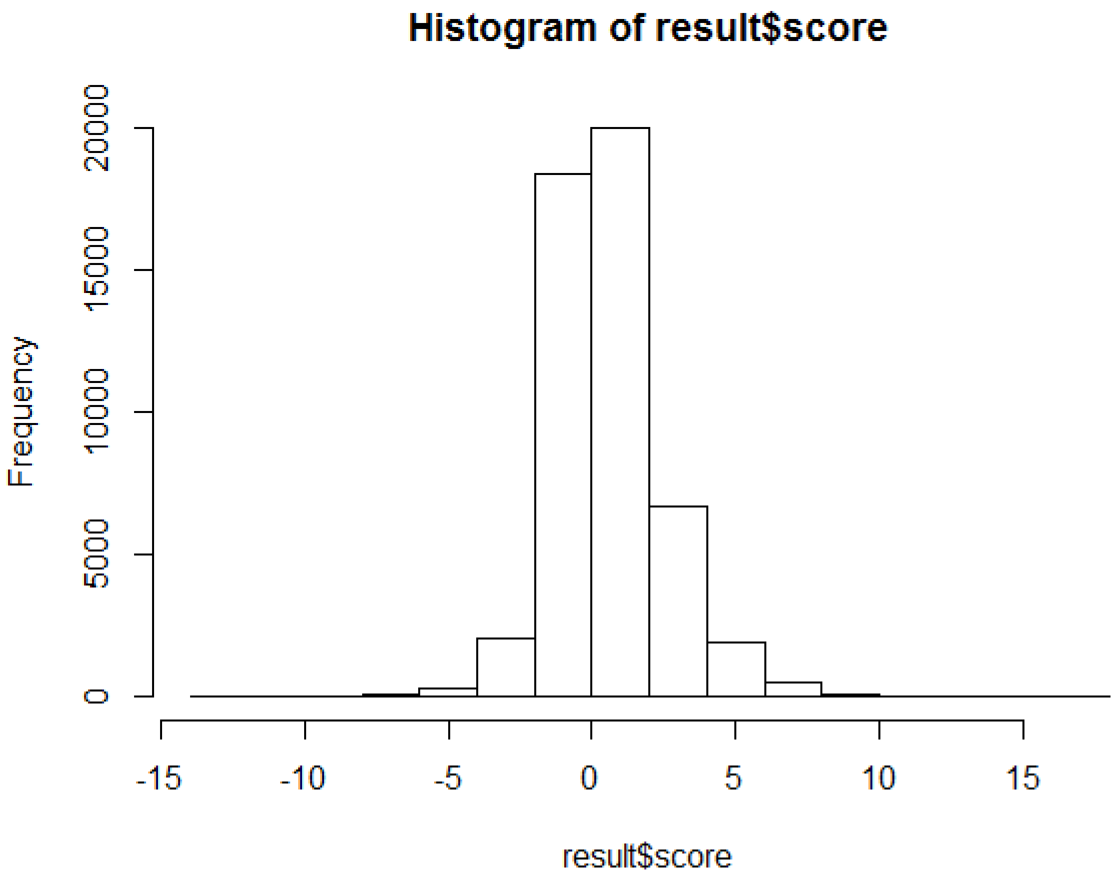
OUTPUT:

Emotions:



sentiment new.png
Type: PNG File
Size: 84.5 KB
Dimension: 1366 x 768 pixels

Histogram for the Sentiment Result



6.2.2.7 Interpretation

The Baby Products sold over Amazon has overall average brand perception. Most of the feedback that is associated with Baby Products is neutral.

7
Conclusion

With the rapidly expanding social networks, it is challenging to analyze its large data using existing data mining tools. We have shown that our Architecture to access **Twitter, Amazon and R-Studio Environment** analyzes large data for decision making.

We have shown through our experiments to do Sentiment Analysis on retrieved "**Make-In-India**" **data from Twitter** that the numbers of people have given positive, neutral and negative opinions on the scheme "MII". And mostly the opinion is neutral or positive towards this program.

We have also shown through our experiments to do Sentiment Analysis on retrieved **Online Customer Reviews on Baby Products data from Amazon** that the numbers of people have given positive, neutral and negative reviews for the baby products available to buy on Amazon. Most of the customer's opinion is neutral towards all the categories of baby products. This can also be further extended to segregate the sentiments based on different sellers and categories of a product.

These experiments can also be extended to extract information from other social web media or blogging portals like **Facebook, LinkedIn, TripAdvisor, Zomato, Glassdoor** and many more.

This study can be concluded by stating that the **social media data can be a rich source of information** which, when harnessed by the marketers, can give the organisations an upper edge over its competitors. Analysis of this huge chunk of unstructured data can lead to actionable items and help marketers to better understanding the customer's behaviour, perceptions and feelings. Aso, it can help organisations in improving their products and services by listening to their customers in real time. Organisations should build suitable team and should be willing to invest in social media mining to reap the benefits of this data.

With the experiments, it is also advisable to conclude **R Statistical Tool is sufficiently used for the analysis of large sets of data**. This can be further extended to use PYTHON for more analysis of big data.

.

8

REFERENCES

1. Liu, Bing. Sentiment Analysis and Opinion Mining, Synthesis Lectures on Human Language Technologies 5.1 (2012): 1-167.
2. Joachims, Thorsten Text categorization with support vector machines: Learning with many relevant features.,Springer Berlin Heidelberg, 1998.
3. Abdul-Mageed, Muhammad, Mona T. Diab, and Mohammed Korayem. Subjectivity and sentiment analysis of modern standard Arabic. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers. 2011.
4. Akkaya, Cem, Janyce Wiebe, and Rada Mihalcea. Subjectivity word sense disambiguation. in Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP-2009). 2009.
5. Alm, Ebba Cecilia Ovesdotter. Affect in text and speech, 2008: ProQuest.
6. Andreevskaia, Alina and Sabine Bergler. Mining WordNet for fuzzy sentiment: Sentiment tag extraction from WordNet glosses. in Proceedings of Conference of the European Chapter of the Association for Computational Linguistics (EACL-06). 2006.
7. Andreevskaia, Alina and Sabine Bergler. When specialists and generalists work together: Overcoming domain dependence in sentiment tagging. In Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL-2008). 2008.
8. Andrzejewski, David and Xiaojin Zhu. Latent Dirichlet Allocation with topic-in-set knowledge. in Proceedings of NAACL HLT. 2009.
9. Andrzejewski, David, Xiaojin Zhu, and Mark Craven. Incorporating domain knowledge into topic modeling via Dirichlet forest priors. In Proceedings of ICML. 2009.
10. Archak, Nikolay, Anindya Ghose, and Panagiotis G. Ipeirotis. Show me the money!: deriving the pricing power of product features by mining consumer reviews. in Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD-2007). 2007.
11. Asher, Nicholas, Farah Benamara, and Yvette Yannick Mathieu. Distilling opinion in discourse: A preliminary study. in Proceedings of the International Conference on Computational Linguistics (COLING-2008): Companion volume: Posters and Demonstrations. 2008.
12. Asur, Sitaram and Bernardo A. Huberman. Predicting the future with social media. Arxiv preprint arXiv:1003.5699, 2010.
13. Aue, Anthony and Michael Gamon. Customizing sentiment classifiers to new domains: a case study. in Proceedings of Recent Advances in Natural Language Processing (RANLP-2005). 2005.

14. Banea, Carmen, Rada Mihalcea, and Janyce Wiebe. Multilingual subjectivity: are more languages better? in Proceedings of the International Conference on Computational Linguistics (COLING-2010). 2010.
15. Banea, Carmen, Rada Mihalcea, Janyce Wiebe, and Samer Hassan. Multilingual subjectivity analysis using machine translation. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2008). 2008.
16. Bar-Haim, Roy, Elad Dinur, Ronen Feldman, Moshe Fresko, and Guy Goldstein. Identifying and Following Expert Investors in Stock Microblogs. in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP-2011). 2011.
17. Sentiment analysis. (2016). Wikipedia. Retrieved 6 May 2016, from https://en.wikipedia.org/wiki/Sentiment_analysis
18. Ana M. Diaz Martin, Hector David et al (2014). Using Twitter to engage with customers: a data mining approach. Retrieved from: www.emeraldinsight.com/1066-2243.htm.
19. Miljana Mitic, Fotis Misopoulos, Alexandros Kapoulas et al (2014). Uncovering customer service experiences with Twitter: the case of airline industry. Retrieved from: www.emeraldinsight.com/0025-1747.htm
20. SocialBakers.com. (2016). Social Media Marketing, Statistics and Monitoring Tools. Retrieved on 30th April 2016, from http://www.socialbakers.com/.
21. Ernst Young (2015), Social Media Marketing India Trends Study 2nd Edition. Retrieved on 30th April 2016, from http://www.ey.com/Publication/vwLUAssets/EY-social-media-marketing-india-trends-study-2014/$FILE/EY-social-media-marketing-india-trends-study-2014.pdf
22. Tillkeyling.com (2016), A workaround for Twitter's Search-API limitations: Using the Twitter Websearch and Facepager. Retrieved on 30th Jan 2016, from: http://tillkeyling.com/a-workaround-for-twitters-search-api-limitations-using-the-twitter-websearch-and-facepager.html
23. Statista.com(2016). The statistics Portal | Statistics and facts on internet usage in India. Retrieved on 30th Jan 2016, from: http://www.statista.com/topics/2157/internet-usage-in-india/
24. What is opinion mining (sentiment mining)? - Definition from WhatIs.com. (2016). SearchBusinessAnalytics(2016). Retrieved 6 May 2016, from http://searchbusinessanalytics.techtarget.com/definition/opinion-mining-sentiment-mining