

# Text Mining on Twitter Data Using Visualizing Technique R

Aalia Bano Qureshi, Dr. Tripti Arjariya, Dr. Mohit Gangwar

Department of Computer Science & Engineering, Bhabha Engineering Research Institute, Bhopal, India

Prof. & Head, Department of Computer Science & Engineering, Bhabha Engineering Research Institute, Bhopal, India

Principal, Department of Computer Science & Engineering, Bhabha Engineering Research Institute, Bhopal, India

**ABSTRACT:** 'BIG DATA' is getting a more importance in different industries over the last year or two, on the basis of large amount of data is generated by different industries every day. The term Big Data is applied to large data sets so the traditional databases was unable to store that large data sets like a traditional way and unable to process such large datasets. It has huge potential to transform business and power in several ways. Here the challenge is not only storing the data, but also accessing and analyzing the required data in specified amount of time. One of the most powerful aspects of using R is that you can download free packages for so many tools and types of analysis. Text analysis is still somewhat in its infancy, but is very promising. It is estimated that as much as 80% of the world's data is unstructured, while most types of analysis only work with structured data. In this paper, we will explore the potential of R packages to analyze unstructured text and pre-processed the text and after that we can visualize the result.

**KEYWORDS:** Big data, R, Unstructure data, text mining, visualization

## I. INTRODUCTION

Over past ten years, industries and organizations don't need to store and perform much operations and analytics on data of the customers. But around from 2005, the need to transform everything into data is much entertained to satisfy the requirements of the people. So Big data came into picture in the real time business analysis of processing data. From 20th century onwards this WWW has completely changed the way of expressing their views. Present situation is completely they are expressing their thoughts through online blogs, discussion forms and also some online applications like Facebook, Twitter, etc. If we take Twitter as our example nearly 1TB of text data is generating within a week in the form of tweets. So, by this it is understand clearly how this Internet is changing the way of living and style of people. Among these tweets can be categorized by the hash value tags for which they are commenting and posting their tweets. So, now many companies and also the survey companies are using this for doing some analytics such that they can predict the success rate of their product or also they can show the different view from the data that they have collected for analysis. But, to calculate their views is very difficult in a normal way by taking these heavy data that are going to generate day by day.

Text mining has become a preferred approach to analyzing and understanding massive datasets not amenable to ancient qualitative analysis techniques. These tools are applied to a variety of knowledge issues, like understanding themes in social media or facilitating data retrieval in unstructured knowledge. Text mining will be a extremely useful gizmo within the beginnings of analysis exploration, permitting the matter knowledge to counsel themes and ideas to the scientist throughout analysis. this could give a helpful place to begin for framing any analysis queries and analysis approaches, notably if hypotheses and queries don't seem to be illustrious before (as is typical with an inductive analysis approach). what is more, these tools can even assist in cleansing and structuring text-based knowledge for future analysis in mental image or different graphical tools. And, additionally to the tangible analysis edges, text mining will be a fun and fruitful method of discovery!. Text Mining, is one in every of the foremost frequent however

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijirset.com](http://www.ijirset.com)

Vol. 6, Issue 3, March 2017

difficult exercise two-faced by beginners in knowledge science / analytics specialists. the most important challenge is one has to totally assess the underlying patterns in text, that too manually. For example: it's pretty common to delete numbers from the text before we tend to do any quite text mining. however what if we would like to extract one thing like "24/7". Hence, the text cleansing exercise is extremely personalised as per the target of the exercise and therefore the sort of text patterns.

R is each a language and surroundings orienting towards applied math computing and graphics creation (R Core Team, 2016). R is created on the market below the antelope General Public License; as a results of sturdy community involvement, there are various extensions, referred to as packages, developed over time, likewise as sturdy documentation. thanks to this extensibility and flexibility, R has remained systematically common for knowledge and text mining applications across several domains, and includes powerful text mining tools.

## II. LITERATURE REVIEW

Anjali Ganesh Jivani [22] mentioned that the aim of stemming is to scale back completely different grammatical forms or word kinds of a word like its noun, adjective, verb, adverb etc. The goal of stemming is to scale back inflectional types and generally derivationally connected kinds of a word to a standard base form. This paper discusses completely different ways of stemming and their comparisons in terms of usage, blessings yet as limitations. the fundamental distinction between stemming and lemmatization is additionally mentioned.

Vishal Gupta et.al [23] has analysed the stemmer's performance and effectiveness in applications like lexicon varies across languages. A typical easy stemmer rule involves removing suffixes employing a list of frequent suffixes, whereas a additional complicated one would use morphological information to derive a stem from the words. The paper provides an in depth define of common stemming techniques and existing stemmers for Indian languages.

K.K. Agbele [24] mentioned the technique for developing pervasive computing applications that square measure versatile and pliable for users. During this context, however, info retrieval (IR) is usually outlined in terms of location and delivery of documents to a user to satisfy their info want. In most cases, morphological variants of words have similar linguistics interpretations and may be thought of as equivalent for the aim of IR applications. The rule Context-Aware Stemming (CAS) is planned, that may be a changed version of the extensively used Porter's stemmer. Considering solely generated purposeful stemming words because the stemmer output, the results show that the changed rule considerably reduces the error rate of Porter's rule from seventy six.7% to 6.7% while not compromising the effectuality of Porter's rule.

Hassan Saif [25] has investigated whether or not removing stop words helps or hampers the effectiveness of Twitter sentiment classification ways. For this investigation he has applied, six {different|totally completely different|completely different} stop word identification ways to Twitter knowledge from six different datasets and observe however removing stop words affects 2 well-known supervised sentiment classification ways. The result shows that victimisation pre-compiled lists of stop words negatively impact the performance of Twitter sentiment classification approaches. On the opposite hand, the dynamic generation of stop word lists, by removing those rare terms showing one time within the corpus seems to be the best technique for maintaining a high classification performance whereas reducing the information scantness and considerably shrinking the feature area.

## III RELATED WORK

There are several text classification and clustering methods. Most text categorization techniques reduce this large number of features by eliminating stop words, or stemming. This is effective to a certain extent but the remaining number of features is still huge. It is important to use feature selection methods to handle the high dimensionality of data for effective text categorization. Feature selection in text classification focuses on identifying relevant information without affecting the accuracy of the classifier. There are many feature selection methods. Mainly they are classified as filter and wrapper feature selection methods. One of the feature selection methods can be choose to identify relevant

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijirset.com](http://www.ijirset.com)

Vol. 6, Issue 3, March 2017

information from documents based on their content. Selection of feature selection is an important and challenging step in text mining.

## IV. PROBLEM DEFINITION

Text mining can help an organization derive potentially valuable business insights from text-based content such as word documents, email and postings on social media streams like Facebook, Twitter and LinkedIn. Data mining or Text mining plays a important role in decision making because through these mining techniques we can analyse the data and on the basis of result we can take a decision. Now a days social media sites like twitter are widely used to share user options on various topics, twitter gives a platform to user to share their views and thoughts on various field like political, industrial, education and there is a petabytes of data generated by twitter in a day.

So the mining techniques are used to analysis the social twitter data thorough we get large amount of datasets to analysis.so the analysis of twitter data provides a better way for making decision.

## V. PROPOSED WORK

Text mining of Twitter data with R packages *twitteR*, *tm* and *ggplot2*

### 1. *twitteR*

Package *twitteR* provides access to Twitter information, 1st we have a tendency to square measure making a twitter streaming API known as twitter app by that we are able to get out access token keys . through this package we have a tendency to square measure retrieving information known as tweets from the twitter server and supply storage.

### 2. *tm*

*tm* package provides functions for text mining, the most structure for managing documents in *Tm* may be a questionable Corpus, representing a group of text documents. A corpus is associate abstract conception, and there will exist many implementations in parallel. The default implementation is that the questionable VCorpus.

### 3. *ggplot2*

*ggplot2* package may be a plotting system for R, supported the synchronic linguistics of graphics, that tries to require the nice elements of base and lattice graphics and none of the unhealthy elements. It takes care of the many of the fiddly details that build plotting a trouble (like drawing legends) similarly as providing a strong model of graphics that produces it simple to provide advanced multi-layered graphics.

## VI. PROPOSED METHODOLOGY

*Our Steps or Algorithm Steps will follow:*

Step 1: first we create a twitter API for retrieving text from twitter for analysis.

Step 2: after retrieving we transformed the text, tweets are first converted to a data frame and then to a corpus. After that, the corpus needs a couple of transformations, including changing letters to lower case, removing punctuations/numbers and removing stop words.

Step 3: In many cases, words need to be stemmed to retrieve their radicals. For instance, "example" and "examples" are both stemmed to "example". However, after that, one may want to complete the stems to their original forms, so that the words would look "normal".

## International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

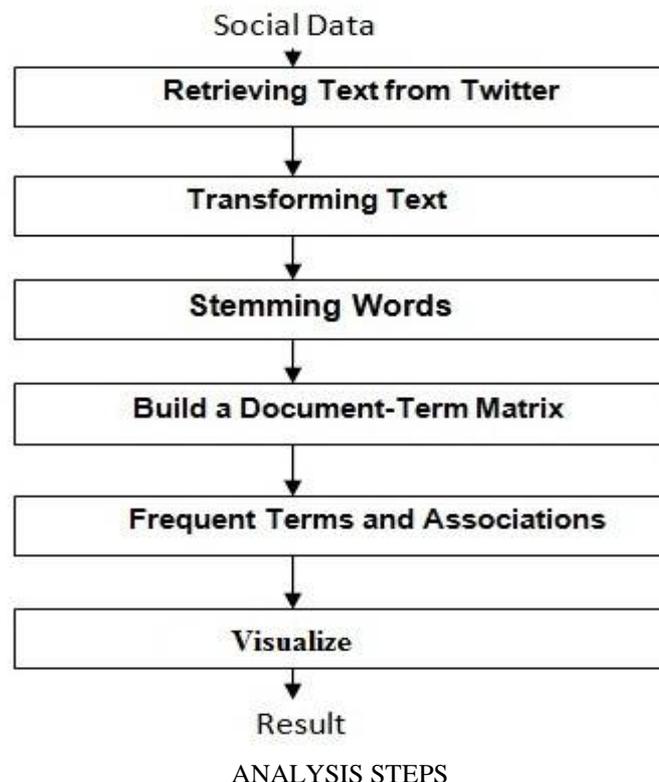
Website: [www.ijirset.com](http://www.ijirset.com)

Vol. 6, Issue 3, March 2017

Step 4: after transforming and stemming process is done then we build a document term matrix. Based on the matrix, many data mining tasks can be done, for example, clustering, classification and association analysis.

Step 5: with the help of matrix we can identify the frequent words and their association between words.

Step 6: After building a document-term matrix, we can show the importance of words with a ggplot2 package.



### VII. EXPERIMENTAL & RESULT ANALYSIS

All the experiments were performed using an i5-2410M CPU @ 2.30 GHz processor and 4 GB of RAM running Windows. After that we can install R base core on windows and Rstudio and then we are fetching real tweets and after that performing text mining on that collected tweets. So, to achieve this we are going to follow the following methods:

- Collecting Tweets.
- Text mining on Tweets.
- Analyze the result.

Collecting Tweets:-

For collecting tweets we need a R package called twitterR and OAuth package to authenticate user consumer and token keys.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijirset.com](http://www.ijirset.com)

Vol. 6, Issue 3, March 2017

```
library(twitter)
library(ROAuth)

consumer_key<- 'xxxxxxxxxxxxxxxxxxxx'
consumer_secret<- 'xxxxxxxxxxxxxxxxxxxx'
access_token<- 'xxxxxxxxxxxxxxxxxxxx'
access_secret<- 'xxxxxxxxxxxxxxxxxxxx'

setup_twitter_oauth(consumer_key, consumer_secret, access_token,access_secret)

tweets <- userTimeline("RTextMining", n = 3200)
```

For consumer key and access tokens we need to create a twitter app through which we can generate our twitter keys and put into twitter\_oauth and then we are storing to collect tweets on RTextMining with framesize of 3200 and stored into tweets variable.

### Text Mining on Tweets:-

For performing text mining on tweets we need a package called tm package which internally consist natural language processing NLP package. Figure 1 shows the loading and performing operation through tm package.

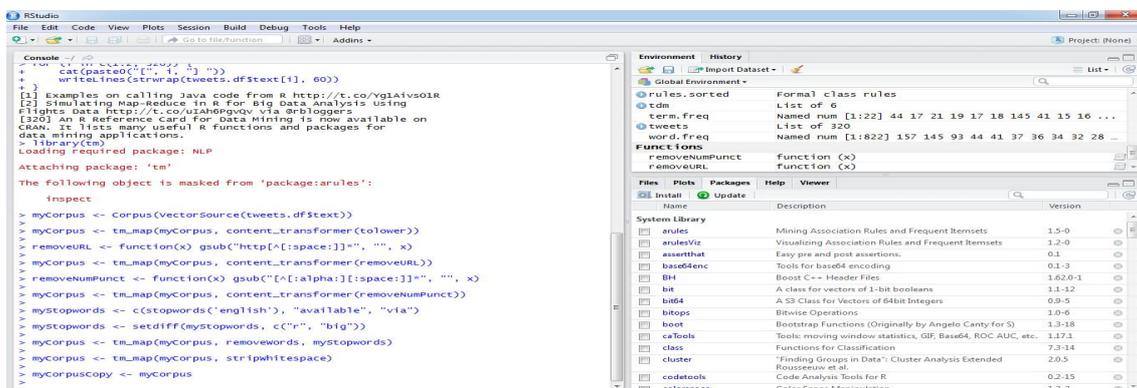


Figure 1: Performing text pre-processing using tm package

We are using corpus for pre-processing twitter text and tm package consist many function through which we can remove numbers, punctual, and special characters coming from text. And first we are converting all the text into lower case to remove noise and we can also eliminate url's coming from text.

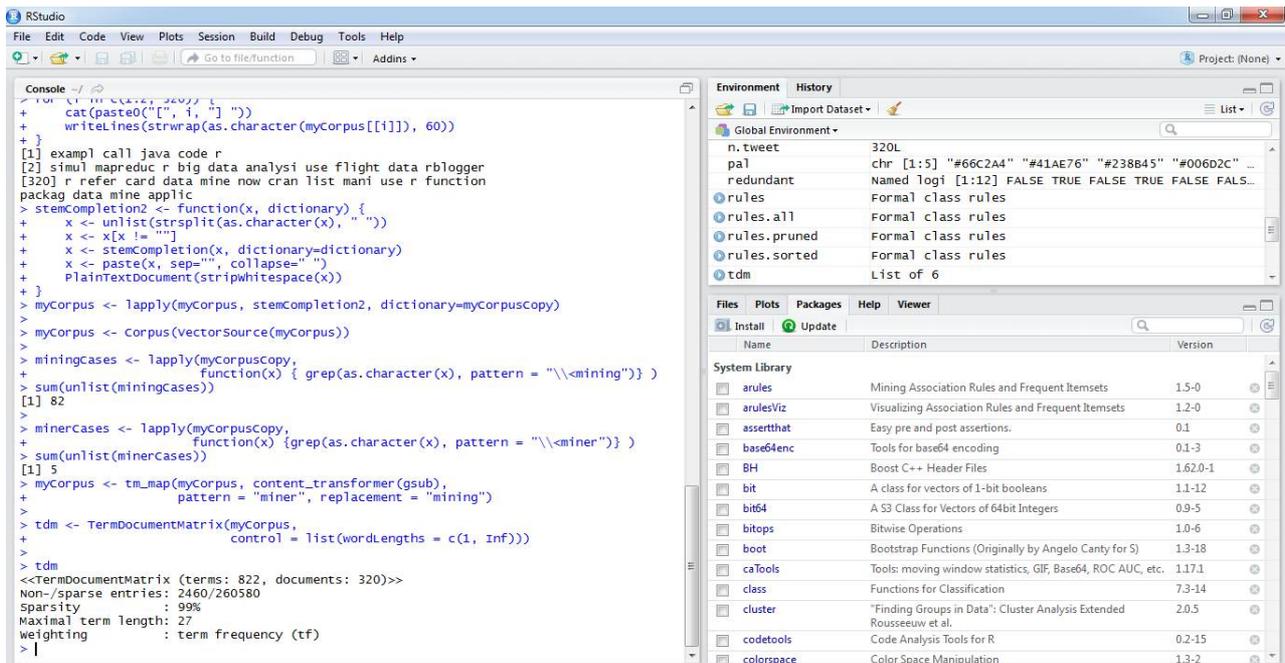
After pre-processing using tm package we can perform stemming on pre-processed data for this we can use Snowball C package. Figure 2 describe the stemming is perform on pre-processed text. For this we can collect same meaning types word separately such as example, example, examples comes into same category.

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijirset.com](http://www.ijirset.com)

Vol. 6, Issue 3, March 2017



```

RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function Addins
Project: (None)

Console
> for (i in 1:length(words)) {
+   cat(paste0("[", i, "] "))
+   writeLines(strwrap(as.character(myCorpus[[i]]), 60))
+ }
[1] exampl call java code r
[2] simul mapreduc r big data analysi use flight data rblogger
[320] r refer card data mine now cran list mani use r function
packag data mine applic
> stemCompletion2 <- function(x, dictionary) {
+   x <- unlist(strsplit(as.character(x), " "))
+   x <- x[x != ""]
+   x <- stemCompletion(x, dictionary=dictionary)
+   x <- paste(x, sep=" ", collapse=" ")
+   PlainTextDocument(stripwhitespace(x))
+ }
> myCorpus <- lapply(myCorpus, stemCompletion2, dictionary=myCorpuscopy)
> myCorpus <- Corpus(VectorSource(myCorpus))
> miningCases <- lapply(myCorpuscopy,
+   function(x) { grep(as.character(x), pattern = "\\<mining\\>") })
> sum(unlist(miningCases))
[1] 82
> minerCases <- lapply(myCorpuscopy,
+   function(x) { grep(as.character(x), pattern = "\\<miner\\>") })
> sum(unlist(minerCases))
[1] 5
> myCorpus <- tm_map(myCorpus, content_transformer(gsub),
+   pattern = "miner", replacement = "mining")
> tdm <- TermDocumentMatrix(myCorpus,
+   control = list(wordLengths = c(1, Inf)))
> tdm
<<TermDocumentMatrix (terms: 822, documents: 320)>>
Non-/sparse entries: 2460/260580
Sparsity : 99%
Maximal term length: 27
weighting : term frequency (tf)
> |

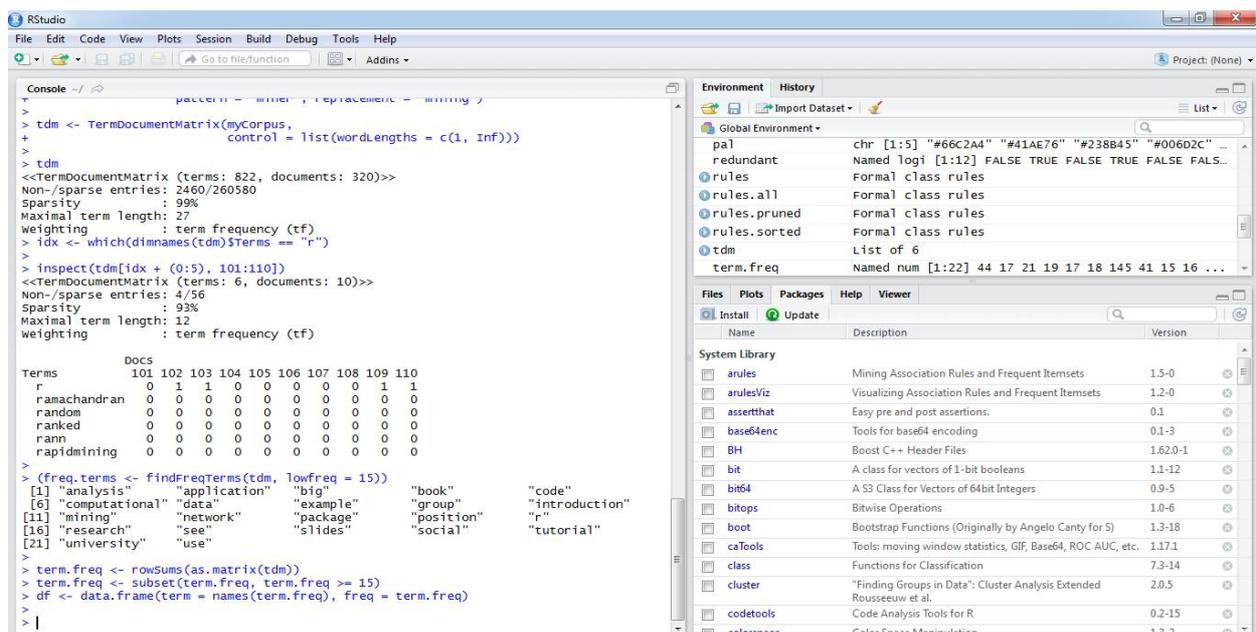
Environment History
Global Environment
n.tweet 320L
pal chr [1:5] "#66C2A4" "#41AE76" "#238B45" "#006D2C" ...
redundant Named logi [1:12] FALSE TRUE FALSE TRUE FALSE FALS...
rules Formal class rules
rules.all Formal class rules
rules.pruned Formal class rules
rules.sorted Formal class rules
tdm List of 6

Files Plots Packages Help Viewer
Install Update
Name Description Version
System Library
arules Mining Association Rules and Frequent Itemsets 1.5-0
arulesViz Visualizing Association Rules and Frequent Itemsets 1.2-0
assertthat Easy pre and post assertions. 0.1
base64enc Tools for base64 encoding 0.1-3
BH Boost C++ Header Files 1.62.0-1
bit A class for vectors of 1-bit booleans 1.1-12
bit64 A S3 Class for Vectors of 64bit Integers 0.9-5
bitops Bitwise Operations 1.0-6
boot Bootstrap Functions (Originally by Angelo Canty for S) 1.3-18
caTools Tools: moving window statistics, GIF, Base64, ROC AUC, etc. 1.17.1
class Functions for Classification 7.3-14
cluster "Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al. 2.0.5
codetools Code Analysis Tools for R 0.2-15
colorspace Color Space Manipulation 1.3-2

```

Figure 2: Performing stemming on preprocessed text

After stemming process we can develop a term-document matrix and find the frequency of each term using association properties. Figure 3 describe the operation on tdm (term document matrix) and finding most frequent keywords.



```

RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function Addins
Project: (None)

Console
> pattern = "miner", replacement = "mining")
> tdm <- TermDocumentMatrix(myCorpus,
+   control = list(wordLengths = c(1, Inf)))
> tdm
<<TermDocumentMatrix (terms: 822, documents: 320)>>
Non-/sparse entries: 2460/260580
Sparsity : 99%
Maximal term length: 27
weighting : term frequency (tf)
> idx <- which(dimnames(tdm)$Terms == "r")
> inspect(tdm[idx + (0:5), 101:110])
<<TermDocumentMatrix (terms: 6, documents: 10)>>
Non-/sparse entries: 4/56
Sparsity : 93%
Maximal term length: 12
weighting : term frequency (tf)
>
Terms Docs
101 102 103 104 105 106 107 108 109 110
r 0 1 1 0 0 0 0 0 1 1
ramachandran 0 0 0 0 0 0 0 0 0 0
random 0 0 0 0 0 0 0 0 0 0
ranked 0 0 0 0 0 0 0 0 0 0
rann 0 0 0 0 0 0 0 0 0 0
rapidmining 0 0 0 0 0 0 0 0 0 0
> (freq.terms <- findFreqTerms(tdm, lowfreq = 15))
[1] "analysis" "application" "big" "book" "code"
[6] "computational" "data" "example" "group" "introduction"
[11] "mining" "network" "package" "position" "r"
[16] "research" "see" "slides" "social" "tutorial"
[21] "university" "use"
> term.freq <- rowSums(as.matrix(tdm))
> term.freq <- subset(term.freq, term.freq >= 15)
> df <- data.frame(term = names(term.freq), freq = term.freq)
> |

Environment History
Global Environment
pal chr [1:5] "#66C2A4" "#41AE76" "#238B45" "#006D2C" ...
redundant Named logi [1:12] FALSE TRUE FALSE TRUE FALSE FALS...
rules Formal class rules
rules.all Formal class rules
rules.pruned Formal class rules
rules.sorted Formal class rules
tdm List of 6
term.freq Named num [1:22] 44 17 21 19 17 18 145 41 15 16 ...

Files Plots Packages Help Viewer
Install Update
Name Description Version
System Library
arules Mining Association Rules and Frequent Itemsets 1.5-0
arulesViz Visualizing Association Rules and Frequent Itemsets 1.2-0
assertthat Easy pre and post assertions. 0.1
base64enc Tools for base64 encoding 0.1-3
BH Boost C++ Header Files 1.62.0-1
bit A class for vectors of 1-bit booleans 1.1-12
bit64 A S3 Class for Vectors of 64bit Integers 0.9-5
bitops Bitwise Operations 1.0-6
boot Bootstrap Functions (Originally by Angelo Canty for S) 1.3-18
caTools Tools: moving window statistics, GIF, Base64, ROC AUC, etc. 1.17.1
class Functions for Classification 7.3-14
cluster "Finding Groups in Data": Cluster Analysis Extended Rousseeuw et al. 2.0.5
codetools Code Analysis Tools for R 0.2-15
colorspace Color Space Manipulation 1.3-2

```

Figure 3: Term-Document matrix & frequent keywords

# International Journal of Innovative Research in Science, Engineering and Technology

(An ISO 3297: 2007 Certified Organization)

Website: [www.ijirset.com](http://www.ijirset.com)

Vol. 6, Issue 3, March 2017

Analysis the result:-

After text mining using tm package we can analyse the text mining result using visualizing package called ggplot2 , In tm package we can find the frequent terms and using ggplot2 we can visualize these frequent terms based on their count value which indicates how many time the terms is coming from the text. Through visualization we can easily say that which terms is coming frequently or which term is least frequent. Figure 4 indicates the terms along with their count.

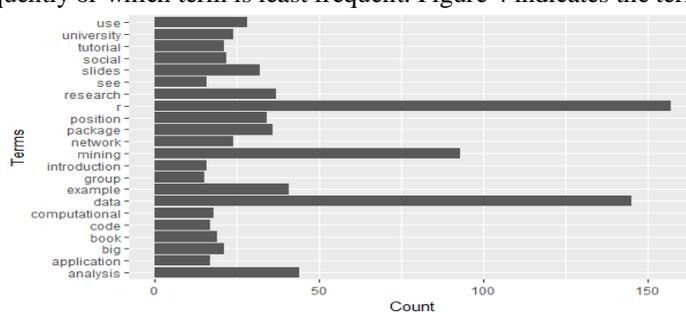


Figure 4: Frequency of terms in the text

## VIII. CONCLUSION

Twitter data is very useful in decision making because its provide a variety of opinions on various topics. so the texting mining will perform on twitter data and we are using a visualizing techniques called R which comes with a variety of packages. Through which we are uses twitterR package and ROAuth package we can authenticate and fetching text from the tweets and stored into tweets variable, After that we can perform text mining on the text or preprocessing a text with the help of tm package. With the help of these we can eliminate the redundancy and numbers, special characters, url's from the text and then we performing the stemming on text and create a term-document matrix to find frequent terms and their association. After that we can visualize the result and say that which keywords are most or least frequent.

## REFERENCES

- [01] Mehmet Ula\_ Çak\_r” Text Mining Analysis in Turkish Language Using BigData Tools” 2016 IEEE 40th Annual Computer Software and Applications.
- [02] Anjali Ganesh Jivani , A Comparative Study of Stemming Algorithms, International Journal of Computer, Technology and Application, Volume 2, ISSN:2229-6093.
- [03] Vishal Gupta, Gurpreet Singh Lehal, A Survey of Common Stemming Techniques and Existing Stemmers for Indian Languages, Journal of Emerging Techniloiigies in Web Intelligence, VOL. 5, NO. 2, MAY 2013.
- [04] Agbele, A.O. Adesina, N.A. Azeez, & A.P. Abidoye , Context-Aware Stemming Algorithm for Semantically Related Root Words, African Journal of Computing & ICT.
- [05] Hassan Saif, Miriam Fernandez, Yulan He, Harith Alani , On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter.
- [06] Vishal Gupta and Gurpreet S. Lehal, A Survey of Text Mining Techniques and Applications, JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE, VOL. 1, NO. 1, AUGUST 2009.
- [07] Porter M.F, Snowball: A language for stemming algorithms. 2001.
- [08] Mladenic Dunja, Automatic word lemmatization. Proceedings B of the 5th International Multi- Conference Information Society IS. 2002, 153-159.
- [09] Ning Zhong, Yuefeng Li, and Sheng-Tang Wu, “Effective Pattern Discovery for Text Mining”, *IEEE Transactions on Knowledge And Data Engineering*, Vol. 24, No.1, January 2012..
- [10] Yuefeng Li, Sheng-Tang Wu and Xiaohui Tao, “Effective Pattern Taxonomy Mining in Text Documents”, *ACM*, October 26–30, 2008.
- [11] Pattan Kalesha, M. Babu Rao and Ch. Kavitha, “Efficient Preprocessing and Patterns Identification Approach for Text Mining”, *International Journal of Computer Trends and Technology (IJCTT)*, Vol. 6, No. 2, December 2013.
- [12] Zakaria Elberichi, Abdelattif Rahmoun and Mohamed Bentaalah, “Using WordNet for Text Categorization”, *International Arab Journal of Information Technology*, Vol. 5, No. 1, January 2008.