

1)ABSTRACT:

The abstract aims to highlight the process of conducting sentiment analysis and creating a word cloud using R with Twitter data. Sentiment analysis is a powerful technique that involves determining the emotional tone of a text, which can provide valuable insights into public opinion and trends. In this study, we demonstrate how to extract tweets using the Twitter API and preprocess the text data by removing special characters, URLs, and irrelevant information. We then employ a sentiment analysis library to assign sentiment scores (positive, negative, or neutral) to each tweet based on the contained language.

Furthermore, we delve into the creation of a word cloud, a visual representation of frequently occurring words in the dataset. This visualization aids in identifying prominent themes and popular terms within the collected tweets. By utilizing the 'tm' and 'wordcloud' packages in R, we showcase the step-by-step process of tokenizing, cleaning, and transforming the text into a word cloud. The size and color of words in the cloud are determined by their frequency of occurrence, allowing us to quickly grasp the most prevalent sentiments and topics present in the Twitter data.

Overall, this abstract encapsulates a comprehensive guide to performing sentiment analysis and generating a word cloud using R with Twitter data. The presented methodology equips researchers, analysts, and data enthusiasts with valuable tools to gain meaningful insights from social media content and explore the underlying sentiments and themes expressed by Twitter users.

2)PROBLEM STATEMENT:

In today's digital age, social media platforms have evolved into dynamic spaces where individuals express opinions, emotions, and perspectives on a wide range of topics. Extracting meaningful insights from this vast pool of user-generated content poses a significant challenge. One of the prominent issues is the need to decipher the sentiments embedded within these textual data, as understanding public opinion is crucial for decision-making processes in various fields including marketing, politics, and social research.

Furthermore, the sheer volume of data generated on platforms like Twitter demands efficient techniques to distill relevant information. Special characters, URLs, and extraneous content often hinder the accuracy of sentiment analysis, necessitating robust preprocessing methods to enhance the quality of results. Moreover, while sentiment analysis provides a structured understanding of emotional tones, it might not capture the nuanced thematic aspects present in the data.

Another pertinent challenge lies in visually representing the salient themes and frequently occurring terms within the Twitter dataset. Traditional methods of data presentation can be cumbersome and fail to capture the essence of the underlying sentiments and trends. Word clouds offer a visually intuitive approach, but their creation involves intricate processes of data transformation, tokenization, and selection of meaningful terms.

Addressing these challenges, this study endeavors to develop a comprehensive approach using the R programming language. By seamlessly integrating sentiment analysis and word cloud generation from Twitter data, this research seeks to provide a practical solution for extracting valuable insights and shedding light on the complex interplay of sentiments and themes inherent in user-generated content.

3)INTRODUCTION:

Sentiment analysis has emerged as a powerful tool for deciphering public opinion and gauging sentiment towards specific topics or events. Twitter, with its vast repository of public opinion data in the form of tweets, stands as a rich source for sentiment analysis. R, programming language designed for data analysis and visualization, proves to be an ideal choice for this task. Leveraging R's capabilities, you can effectively gather Twitter data, meticulously clean and prepare it for analysis, perform sentiment analysis, and generate insightful visualizations to a deeper understanding of public sentiment. This process involves collecting tweets related to the topic or event of interest using the rtweet package, cleaning and pre-processing the data to remove noise and inconsistencies, applying the sentiment package to classify tweet as positive, negative, or neutral, and finally, utilizing the ggplot2 package to create compelling visualizations that effectively convey the distribution of sentiment across the collected tweets. you can transform Twitter's vast collection of opinions into actionable insights. This enables you to prevalent sentiment trends, understand the underlying reasons behind those trends, and uncover hidden patterns that would otherwise remain obscured. Whether you are tracking brand perception, gauging public opinion on political issues, or analyzing consumer sentiment towards a product launch, sentiment analysis with R empowers you to extract valuable insights from Twitter's vast social media landscape.

4)METHODOLOGY:

The provided code follows a systematic approach to analyzing sentiment from Twitter data. It involves data collection, text preprocessing, term document matrix creation, visualization, and sentiment analysis. The visualizations provide insights into the frequency of words used in tweets about Apple and the overall sentiment distribution. The sentiment analysis results provide a quantitative measure of the positive and negative sentiment expressed in tweets.

Lexicon-Based Approach

The provided code snippet demonstrates the application of a lexicon-based sentiment analysis approach, specifically using the NRC Lexicon, to analyze Apple-related tweets. It involves data preparation, text cleaning, TDM creation, visualization, and sentiment analysis to provide insights into the overall sentiment of the tweets.

STEPS:

Data Collection and Preprocessing

1. Import Data: The `read.csv()` function is used to import the Twitter data into a data frame named `apple`.
2. Text Conversion: The `iconv()` function is used to convert the text in the text column of the `apple` data frame from the default encoding to UTF-8 encoding.

3. **Corpus Creation:** The `Corpus()` function from the `tm` package is used to create a corpus from the text data.
4. **Text Cleaning:** A series of `tm_map()` functions are used to clean the text data by converting it to lowercase, removing punctuation, removing numbers, removing stop words, removing URLs, removing specific words, replacing specific words, and stripping whitespace.
5. **Term Document Matrix:** The `TermDocumentMatrix()` function is used to create a term document matrix (TDM) from the cleaned text data. The TDM represents the frequency of each word in each document (tweet).
6. **TDM Conversion:** The `as.matrix()` function is used to convert the TDM into a matrix for further analysis.

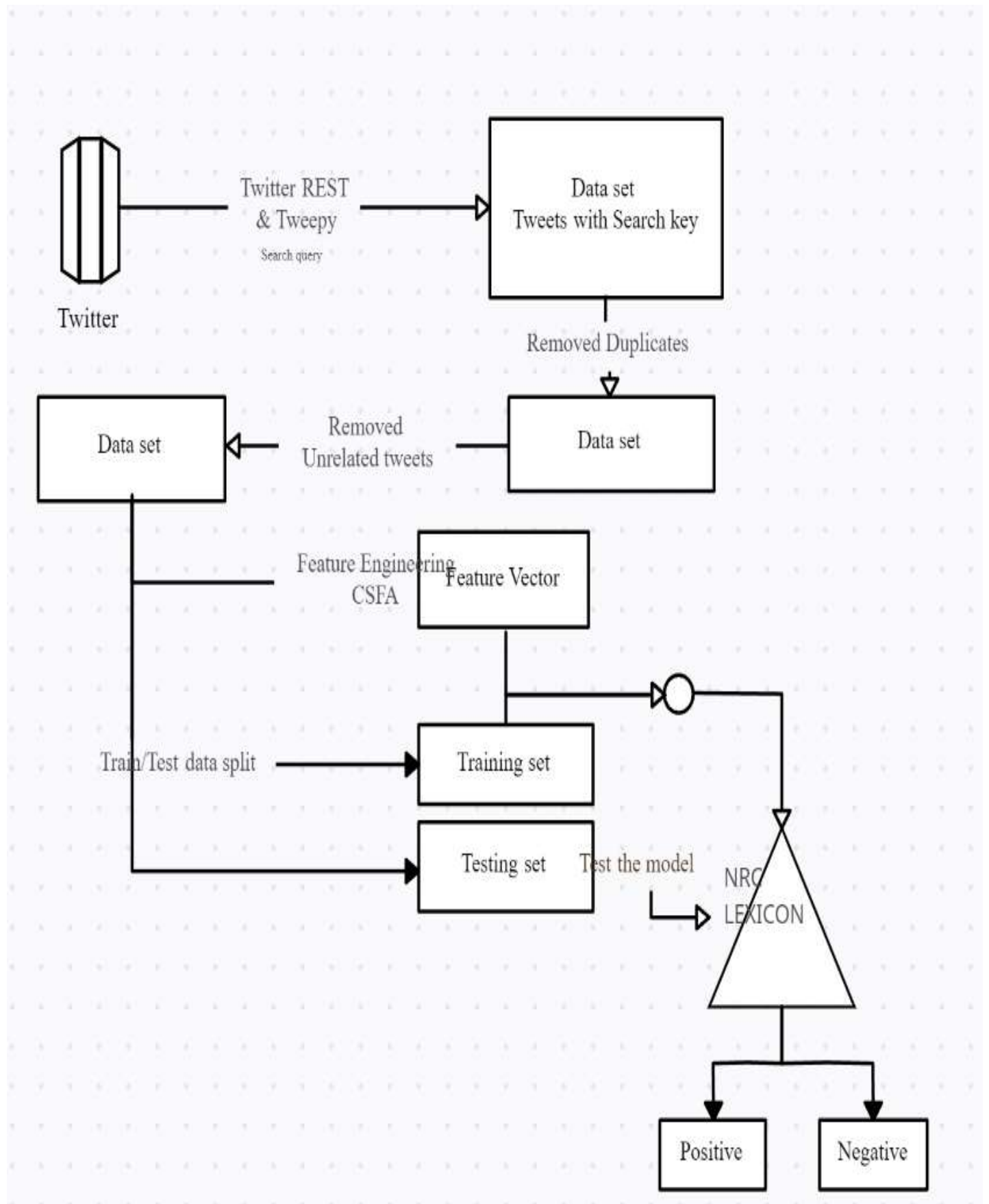
Visualization

1. **Bar Plot:** The `barplot()` function is used to create a bar plot of the frequency of the top 25 words in the TDM.
2. **Word Cloud:** The `wordcloud()` function is used to create a word cloud of the top 150 most frequent words in the TDM.
3. **Word Cloud 2:** The `wordcloud2()` function is used to create a more customizable word cloud with various options for shape, color, and rotation.

Sentiment Analysis

1. **Sentiment Scores:** The `get_nrc_sentiment()` function from the `syuzhet` package is used to obtain sentiment scores for each tweet using the NRC lexicon.
2. **Sentiment Distribution:** A bar plot is created to visualize the distribution of sentiment scores across all tweets.
3. **Positive and Negative Sentiment:** A separate bar plot is created to visualize the distribution of positive and negative sentiment scores across all tweets.

5)PROPOSED SYSTEM:



6)LITERATURE SURVEY:

TITLE	AUTHOR	YEAR PUBLISHED	DESCRIPTION	PROPOSED METHODS
A Survey on Sentiment Analysis and Visualization	Ameni Boumaiza	NOVEMBER 2015	this paper likely serves as a comprehensive review and analysis of sentiment analysis methodologies, challenges, applications, and visualization techniques available up to its publication date, aiming to provide insights into the state of the art in sentiment analysis research and its practical implications.	Lexicon-Based , Machine Learning Models, Sentiment Analysis Deep Learning Architectures Visualization Techniques
Sentiment Analysis and Visualization of Social Media Data	Amir Salarpour, Mohammad Hossein Bamneshin, Dimitris Proios	November 2014	Methods for collecting and preprocessing data from Twitter, facebook, etc. interpreting informal language, emojis, contextual nuances in social media data Guide for researchers/practitioners in understanding sentiment analysis in social media analytics	Data Collection, Data Preprocessing, Sentiment Analysis Techniques , Aspect-Based Sentiment Analysis , Visualization Tools and Techniques
Sentiment Analysis, Visualization and Classification of Summarized News Articles: A Novel Approach	Siddhaling Urologin	August ,2018	It outlines methods to processing and summarizing news content Integration of sentiment analysis tools and visual representations for emotional trend depiction Machine learning models for categorizing news articles by sentiment and content Unified methodology for sentiment-driven analysis of summarized news using advanced classification and visualization strategies	Text Summarization Techniques, TF-IDF, TextRank , Sentiment Analysis Methods , LSTM or Transformer models , Classification Algorithms , Random Forests, Gradient Boosting ,
Sentiment Analysis, Tweet Analysis and Visualization on Big Data Using Apache Spark and Hadoop	Sujala D Shetty	2021	This paper explores Sentiment and tweet analysis on big data Leveraging Apache Spark and Hadoop Processing vast volumes of tweets Applying sentiment analysis techniques Visualizing results using distributed computing frameworks Utilization of Spark and Hadoop for sentiment analysis and tweet sentiment visualization in big data analytics	Data Collection and Preprocessing , tokenization, and handling text data characteristics , Tweet Analysis and Feature Engineering , Scalability and Performance Optimization , Spark's RDDs .

Visualization of Real Time Big Data Through Sentiment Analysis	Vijaya Lakshmi Saragadam, Dr. S. Parasuraman Vijaya Lakshmi Saragadam, Dr. S. Parasuraman	03-10-2022	It focuses on processing vast streams of data in real-time, applying sentiment analysis algorithms to extract emotions or opinions, and visualizing these insights instantly. The study aims to demonstrate the live representation of sentiments through graphical or interactive visualizations, enabling swift and actionable insights from continuously arriving big data sources.	Real-time Data Processing , Sentiment Analysis Algorithms , Visualization Tools and Libraries , Streaming Analytics Platforms , Data Pipeline Architecture , parallelization
Data filtering and visualization for sentiment analysis of ecommerce website	V.Keerthana, P.Prasannakumar, Dr.Ebenezer Abishek.B, Mr.C. Arul Stephen, Dr.A. Vijayalakshmi	2021	This paper focuses on Filtering and visualizing data from an e-commerce website Robust filtering techniques for extracting textual data (reviews, comments) Utilizing visualization tools (charts, sentiment heatmaps) Displaying sentiment trends derived from filtered data Insights into customer opinions and emotions regarding products/services	Natural Language Processing (NLP) Libraries , Sentiment Analysis APIs or Libraries, Visualization Tools and Libraries , Web Development Frameworks .
A Review on Sentiment Analysis and Visualization of Customer Reviews	Apurva V. Gundla, Manisha S. Otari	September 2015	It covers Methodologies Sentiment extraction from customer feedback Focus on natural language processing and machine learning techniques Importance of visual representations like sentiment timelines, word clouds Presentation of sentiment trends from diverse customer reviews Evaluating effectiveness of sentiment analysis methods and visualization tools Interpreting customer sentiments across different industries/domains .	Aspect-Based Sentiment Analysis , Domain-Specific Analysis, Evaluation and Comparative Analysis , Data Collection and Preprocessing
Sentiment Analysis on Twitter Data using R for Understanding Perspectives on E-Healthcare	Sonia Saini, Vinod Kumar Shukla, Ritu Punhani, Ruchika Bathla.	April-2019	The paper explores sentiment analysis on Twitter data, focusing on sentiments and attitudes toward E-Healthcare. It emphasizes the role of social networking sites in generating user opinions and uses R programming for sentiment classification into eight distinct categories and positive/negative sentiments.	Future Research Directions, Conclusion, Analysis and Visualization, Sentiment Analysis.

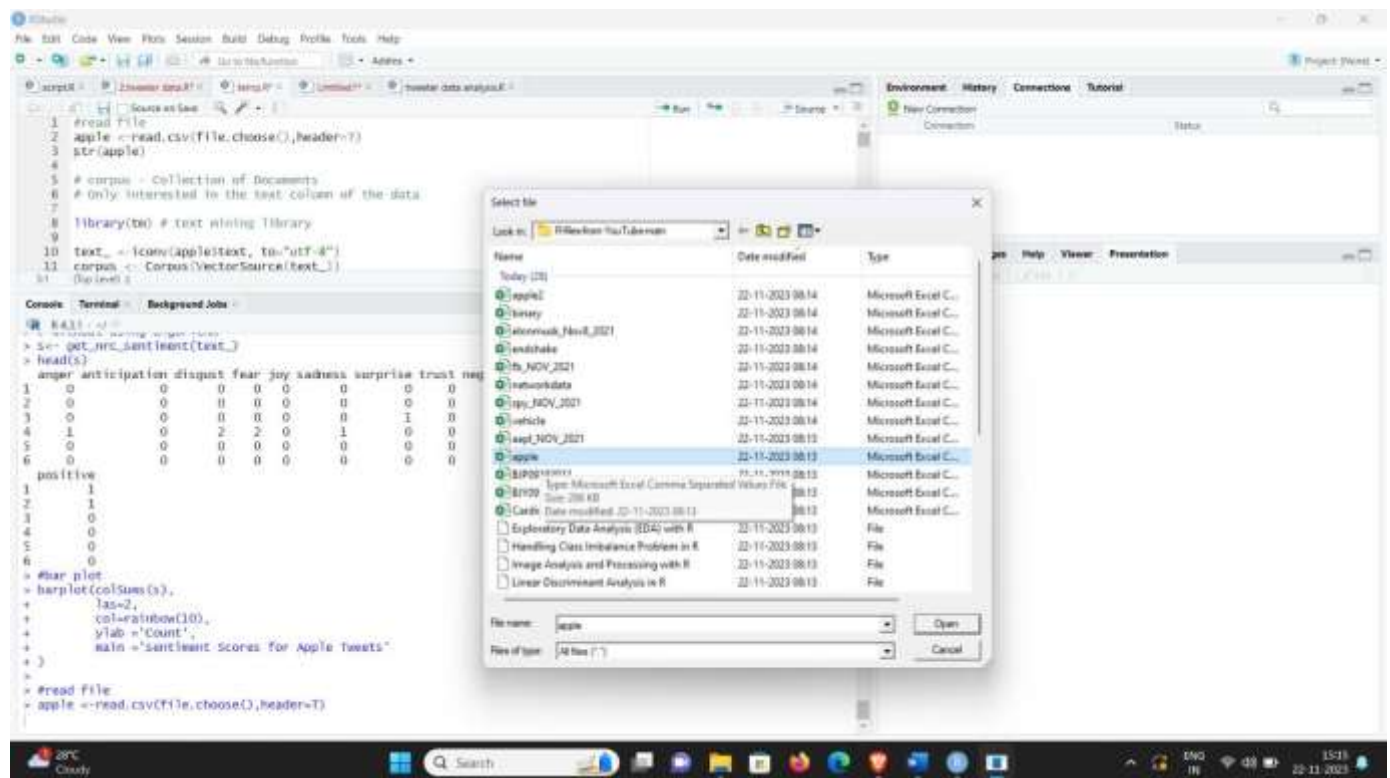
Sentiment Analysis Framework for Twitter Data using R: A Lexicon-Based Approach	Paramita Ray, Amlan Chakrabarti.	February-2017	It Covers Sentiment analysis in social media and electronic platforms. Significance for organizations and consumers in decision-making R software used for Twitter sentiment analysis via Twitter API Methodology includes data collection, pre-processing, lexicon-based sentiment analysis Dictionary-based sentiment analysis and large-scale sentiment estimation algorithm Handling acronym replacement and emoticon detection.	Lexicon-Based Sentiment Analysis, Acronym Replacement, Emoticon Detection, Document and Aspect-Level Analysis
Sentiment Analysis on COVID-19 Twitter Data Streams Using Deep Belief Neural Networks	Jatla Srikanth, Avula Damodaram, Yuvaraja Teekaraman, Ramya Kuppusamy, Amruth Ramesh Thelkar.	06 May 2022	The paper discusses social media, particularly Twitter, as a platform for expressing opinions during events like COVID-19. insights for public health responses and misinformation challenges Methodology: sentiment analysis for gauging public sentiment on social distancing using Twitter data. tokenization, filtering, stemming, N-gram models Deep Belief Neural Network (DBN) with pseudo labelling for tweet. DBN with bigram N-gram models outperforms other models (90.3% accuracy) Potential of analysis to guide location-specific pandemic responses for decision-makers and medical professionals.	Deep Belief Neural Network (DBN), tokenization, filtering, stemming, and N-gram model creation for feature extraction.
Sentiment Knowledge Discovery in Twitter Live Streaming Data using R Language	Gayathri, K. Poonilavu, Navithra, Rajeshwari.	03-october-2022	Twitter's importance for real-time information and opinions Microblogging platforms as valuable for sentiment analysis Sentiment analysis within Natural Language Processing (NLP).positive, negative, neutral, or negation Automation of text classification via algorithms Sentiment polarity categorized at sentence and review levels for Twitter data Understanding sentiment analysis across diverse mediums, including images R language's flexibility and customization for analysis.	Sentiment Polarity, Feature Extraction, Model Enhancement, Scope Expansion.

Machine Learning Model for Arabic Sentiment Analysis of COVID-19 Conspiracy Theories on Twitter	Abdullah Al-Hashedi, Belal Al-Fuhaidi, Abdulqader M. Mohsen, Yousef Ali, Hasan Ali Gamal Al-Kaf, Wedad Al-Sorori, Naseebah Maqtary.	13 Jan 2022	This paper discuss the lack of Arabic sentiment analysis on COVID-19 conspiracy theories in social media Importance of sentiment analysis during the pandemic for understanding public opinions machine learning model for Arabic sentiment analysis on Twitter Word2Vec for word embedding using pretrained CBOW models. Naïve Bayes classifier; exploring single-based and ensemble-based classifiers with/without SMOTE.	Word Embedding, Classifier Evaluation, SMOTE Implementation.
---	---	-------------	---	--

7)IMPLEMENTATION:

1) DATA COLLECTION:

READ FILE :



2) DATA CLEANING: CLEAN TEXT:

```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source | Source on Save | Search
1 #read file
2 apple <- read.csv(file.choose(), header=T)
3 str(apple)
4
5 # corpus - Collection of Documents
6 # only interested in the text column of the data
7
8 library(tm) # text mining library
9
10 text_ <- iconv(apple$text, to="utf-8")
11 corpus <- Corpus(VectorSource(text_))
12 # (Tip: View)
```

Console | Terminal | Background Jobs

```
R 4.3.1
data.frame: 1 x 10
$ text
[1] "RT @option_snipper: $AAPL beat on both eps and revenues. SEES 4Q REV. $498-$528, EST. $49.1B https://t.co/hfHxgJ0IOB" "RT @option_snipper: $AAPL beat on both eps and revenues. SEES 4Q REV. $498-$528, EST. $49.1B https://t.co/hfHxgJ0IOB" "Let's see this break all timers. $AAPL 156.89" "RT @sylviaCap: Things might get ugly for $aapl with the iPhone delay. With $aapl down that means almost all of t"
$ favoriteCount
[1] 0 0 0 0 0 0 0 0 0 0
$ replyToSN
[1] NA NA NA NA NA NA NA NA NA NA
$ created
[1] "2017-08-01 20:31:56" "2017-08-01 20:31:55" "2017-08-01 20:31:55" "2017-08-01 20:31:55"
$ truncated
[1] FALSE FALSE FALSE FALSE FALSE FALSE
$ replyToID
[1] NA NA NA NA NA NA NA NA NA NA
$ id
[1] 8.02e+17 8.92e+17 8.92e+17 8.92e+17 8.92e+17
$ replyToID
[1] NA NA NA NA NA NA NA NA NA NA
$ statusSource
[1] "<a href='\"http://twitter.com/download/iphone/\"' rel='\"nofollow/\">Twitter for iPhone</a>" "<a href='\"http://twitter.com/download/iphone/\"' rel='\"nofollow/\">Twitter for iPhone</a>" "<a href='\"http://stocktwits.com/\"' rel='\"nofollow/\">Stocktwits web</a>" "<a href='\"http://twitter.com/download/android/\"' rel='\"nofollow/\">Twitter for Android</a>"
$ screenName
[1] "KnowledgeMC" "Miguelina" "beckyrlu" "MarvTheBoxer"
$ retweetCount
[1] 3 3 0 85 0 10 30 9 10 1
$ isRetweet
[1] TRUE TRUE FALSE TRUE FALSE TRUE
$ retweeted
[1] FALSE FALSE FALSE FALSE FALSE
$ longitude
[1] NA NA NA NA NA NA NA NA NA NA
$ latitude
[1] NA NA NA NA NA NA NA NA NA NA
```

> #build corpus
> library(tm)
Loading required package: RLP
> corpus <- iconv(apple\$text, to="utf-8")
> corpus <- Corpus(VectorSource(corpus))

```
RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Source | Source on Save | Search
1 #read file
2 apple <- read.csv(file.choose(), header=T)
3 str(apple)
4
5 # corpus - Collection of Documents
6 # only interested in the text column of the data
7
8 library(tm) # text mining library
9
10 text_ <- iconv(apple$text, to="utf-8")
11 corpus <- Corpus(VectorSource(text_))
12 # (Tip: View)
```

Console | Terminal | Background Jobs

```
R 4.3.1
> #build corpus
> library(tm)
Loading required package: RLP
> corpus <- iconv(apple$text, to="utf-8")
> corpus <- Corpus(VectorSource(corpus))
> inspect(corpus[1:5])
<<SimpleCorpus>>
Metadata: corpus specific: 1, document level (indexed): 0
Content: documents: 5

[1] RT @option_snipper: $AAPL beat on both eps and revenues. SEES 4Q REV. $498-$528, EST. $49.1B https://t.co/hfHxgJ0IOB
[2] RT @option_snipper: $AAPL beat on both eps and revenues. SEES 4Q REV. $498-$528, EST. $49.1B https://t.co/hfHxgJ0IOB
[3] Let's see this break all timers. $AAPL 156.89
[4] RT @sylviaCap: Things might get ugly for $aapl with the iPhone delay. With $aapl down that means almost all of the
FANG stocks were down pos.
[5] $AAPL - wow! This was supposed to be a throw-away quarter and AAPL beats by over 500 million in revenue! Trillion
on dollar company by 2018!
> #clean text
> corpus <- tm_map(corpus, tolower)
Warning message:
In tm_map(SimpleCorpus(corpus, tolower)) : transformation drops documents
> inspect(corpus[1:5])
<<SimpleCorpus>>
Metadata: corpus specific: 1, document level (indexed): 0
Content: documents: 5
```


RStudio

```

1 #read file
2 apple <- read.csv(fFile, choose(), header=T)
3 str(apple)
4
5 # corpus - Collection of Documents
6 # only interested in the text column of the data
7
8 library(tm) # text mining library
9
10 text_ <- iconv(apple$text, to="utf-8")
11 corpus <- Corpus(VectorSource(text_))
12 (Top Level) :

```

Environment History Connections Tutorial

File Plots Packages Help Viewer Presentation

Console Terminal Background Jobs

```

R 4.3.1
on dollar company by 2018!
> #clean text
> corpus<-tm_map(corpus, tolower)
Warning message:
In tm_map.SimpleCorpus(corpus, tolower) : transformation drops documents
> inspect(corpus [1:5])
<<SimpleCorpus>>
Metadata: corpus specific: 1, document level (indexed): 0
Content: documents: 5

[1] rt @option_snippet: Saapl beat on both eps and revenues. sees 4q rev. $49b-$52b, est. $49.1b https://t.co/hfhxqj0iob
[2] rt @option_snippet: Saapl beat on both eps and revenues. sees 4q rev. $49b-$52b, est. $49.1b https://t.co/hfhxqj0iob
[3] let's see this break all timers. Saapl 156.89
[4] rt @sylvacap: things might get ugly for Saapl with the iphone delay. with Saapl down that means almost all of the fang stocks were down pos.
[5] Saapl - wow! this was supposed to be a throw-away quarter and aapl beats by over 500 million in revenue! trillion on dollar company by 2018!
> corpus<-tm_map(corpus, removePunctuation)
Warning message:
In tm_map.SimpleCorpus(corpus, removePunctuation) : transformation drops documents
> inspect(corpus[1:5])
<<SimpleCorpus>>
Metadata: corpus specific: 1, document level (indexed): 0
Content: documents: 1

```

20°C Cloudy

Search

22-11-2023

RStudio

```

1 #read file
2 apple <- read.csv(fFile, choose(), header=T)
3 str(apple)
4
5 # corpus - Collection of Documents
6 # only interested in the text column of the data
7
8 library(tm) # text mining library
9
10 text_ <- iconv(apple$text, to="utf-8")
11 corpus <- Corpus(VectorSource(text_))
12 (Top Level) :

```

Environment History Connections Tutorial

File Plots Packages Help Viewer Presentation

Console Terminal Background Jobs

```

R 4.3.1
he fang stocks were down pos.
[5] Saapl - wow! this was supposed to be a throw-away quarter and aapl beats by over 500 million in revenue! trillion on dollar company by 2018!
> corpus<-tm_map(corpus, removePunctuation)
Warning message:
In tm_map.SimpleCorpus(corpus, removePunctuation) : transformation drops documents
> inspect(corpus[1:5])
<<SimpleCorpus>>
Metadata: corpus specific: 1, document level (indexed): 0
Content: documents: 1

[1] rt @option_snippet: aapl beat on both eps and revenues sees 4q rev 49b52b est 491b httpstcofhxqj0iob
> corpus<- tm_map(corpus, removeNumbers)
Warning message:
In tm_map.SimpleCorpus(corpus, removeNumbers) : transformation drops documents
> inspect(corpus[1:5])
<<SimpleCorpus>>
Metadata: corpus specific: 1, document level (indexed): 0
Content: documents: 5

[1] rt @option_snippet: aapl beat on both eps and revenues sees q rev 1b est b httpstcofhxqj0iob
[2] rt @option_snippet: aapl beat on both eps and revenues sees q rev 1b est b httpstcofhxqj0iob
[3] lets see this break all timers aapl
[4] rt sylvacap things might get ugly for aapl with the iphone delay with aapl down that means almost all of the fang stocks were down pos.
[5] aapl - wow this was supposed to be a throwaway quarter and aapl beats by over . million in revenue trillion dolla

```

20°C Cloudy

Search

22-11-2023

Script

```

1 #read file
2 apple <- read.csv(file.choose(), header=T)
3 str(apple)
4
5 # corpus - Collection of Documents
6 # only interested in the text column of the data
7
8 library(tm) # text mining library
9
10 text_ <- iconv(apple$text, to="utf-8")
11 corpus <- Corpus(VectorSource(text_))
12 # (Tip Level)

```

Console

```

R 4.3.1 > corpus <- tm_map(corpus, removeNumbers)
Warning message:
In tm_map.SimpleCorpus(corpus, removeNumbers) :
  transformation drops documents
> inspect(corpus[1:5])
<<SimpleCorpus>>
Metadata: corpus specific: 1, document level (indexed): 0
Content: documents: 5

[1] rt optionsniper aapl beat on both eps and revenues sees q rev bb est b httpstcofhxqjlob
[2] rt optionsniper aapl beat on both eps and revenues sees q rev bb est b httpstcofhxqjlob
[3] lets see this break all timers aapl
[4] rt sylvacap things might get ugly fur aapl with the iphone delay with aapl down that means almost all of the fa
ng stocks were down pos.
[5] aapl! wow this was supposed to be a throwaway quarter and aapl beats by over . million in revenue trillion dolla
r company by
> cleanset <- tm_map(corpus, removeWords, stopwords("english"))
Warning message:
In tm_map.SimpleCorpus(corpus, removeWords, stopwords("english")) :
  transformation drops documents
> inspect(cleanset[1:5])
<<SimpleCorpus>>
Metadata: corpus specific: 1, document level (indexed): 0
Content: documents: 1

[1] rt optionsniper aapl beat eps revenues sees q rev bb est b httpstcofhxqjlob
> removeURL <- tm_map(cleanset, content_transformer(removeURL))
Error: object 'removeURL' not found

```

Script

```

1 #read file
2 apple <- read.csv(file.choose(), header=T)
3 str(apple)
4
5 # corpus - Collection of Documents
6 # only interested in the text column of the data
7
8 library(tm) # text mining library
9
10 text_ <- iconv(apple$text, to="utf-8")
11 corpus <- Corpus(VectorSource(text_))
12 # (Tip Level)

```

Console

```

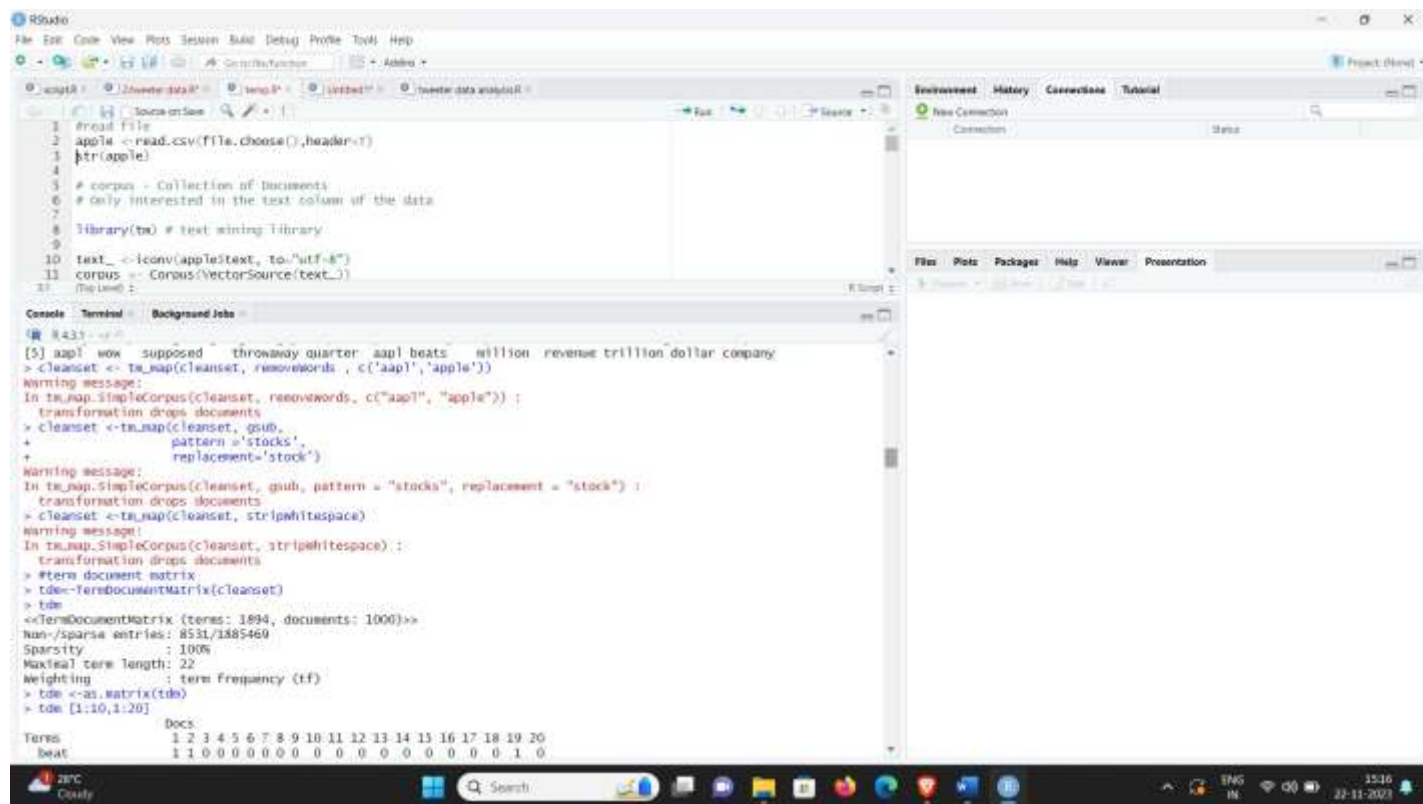
R 4.3.1 > inspect(cleanset[1:5])
<<SimpleCorpus>>
Metadata: corpus specific: 1, document level (indexed): 0
Content: documents: 1

[1] rt optionsniper aapl beat eps revenues sees q rev bb est b httpstcofhxqjlob
> removeURL <- tm_map(cleanset, content_transformer(removeURL))
Error: object 'removeURL' not found
> inspect(cleanset[1:5])
<<SimpleCorpus>>
Metadata: corpus specific: 1, document level (indexed): 0
Content: documents: 1

[1] rt optionsniper aapl beat eps revenues sees q rev bb est b httpstcofhxqjlob
[2] rt optionsniper aapl beat eps revenues sees q rev bb est b httpstcofhxqjlob
[3] lets see break timers aapl
[4] rt sylvacap things might get ugly aapl iphone delay aapl means almost fang stocks pos.
[5] aapl! wow supposed throwaway quarter aapl beats million revenue trillion dollar company
> cleanset <- tm_map(cleanset, removeWords, c("aapl", "apple"))
Warning message:
In tm_map.SimpleCorpus(cleanset, removeWords, c("aapl", "apple")) :
  transformation drops documents
> cleanset <- tm_map(cleanset, gsub,
+ pattern = "stocks",
+ replacement = "stock")
Warning message:
In tm_map.SimpleCorpus(cleanset, gsub, pattern = "stocks", replacement = "stock") :
  transformation drops documents

```

TERM DOCUMENT METRIX:



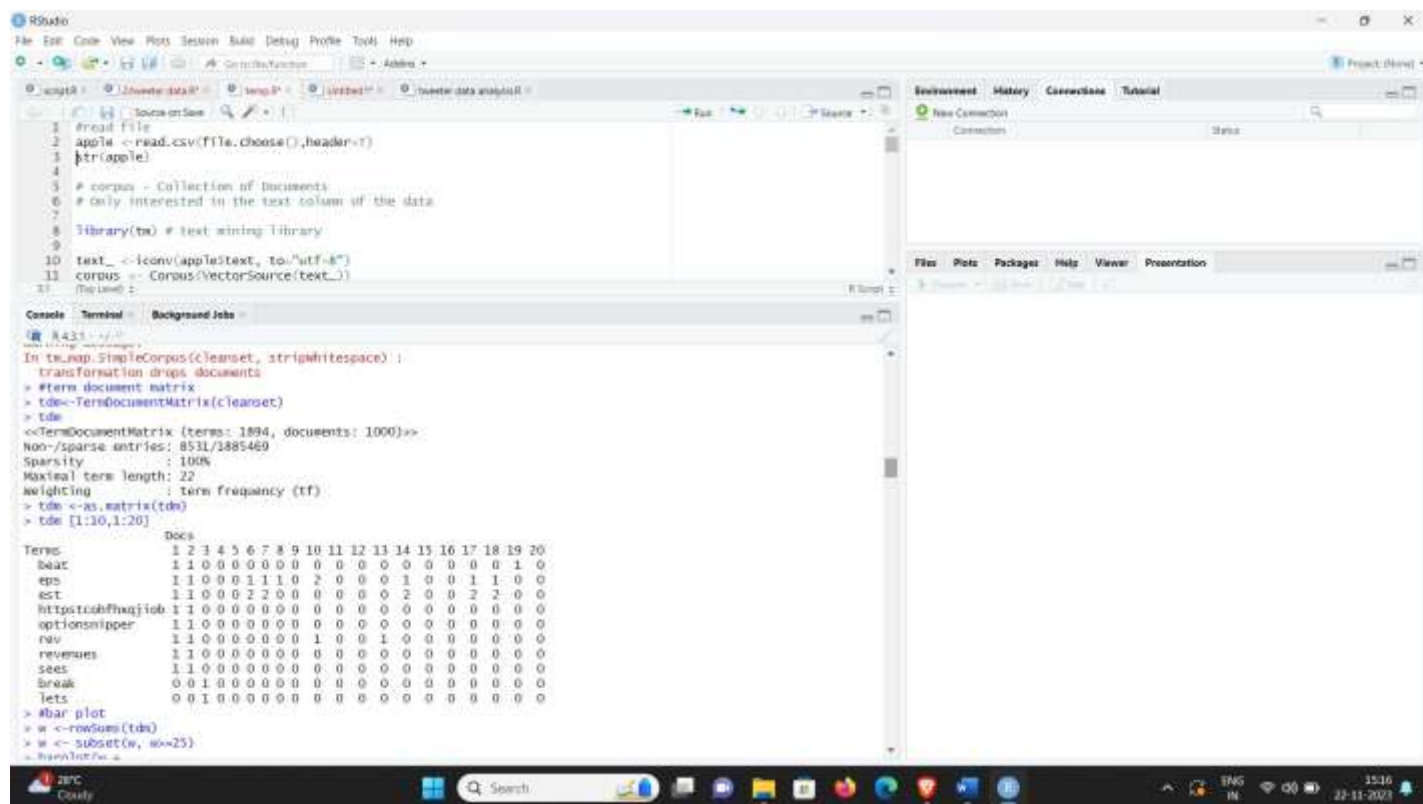
The screenshot shows an RStudio session where a CSV file is read and processed into a Term Document Matrix (TDM). The console output shows the following steps:

```
# Read file
apple <- read.csv(file.choose(), header=T)
# corpus - Collection of Documents
# Only interested in the text column of the data
library(tm) # text mining library
text_ <- iconv(apple$text, to="utf-8")
corpus <- Corpus(VectorSource(text_))

# Cleanse
cleanset <- tm_map(corpus, removeWords, c("apple", "apple"))
Warning message:
In tm_map.SimpleCorpus(cleanset, removeWords, c("apple", "apple")) :
  transformation drops documents
cleanset <- tm_map(cleanset, gsub,
+ pattern = 'stocks',
+ replacement = 'stock')
Warning message:
In tm_map.SimpleCorpus(cleanset, gsub, pattern = "stocks", replacement = "stock") :
  transformation drops documents
cleanset <- tm_map(cleanset, stripwhitespace)
Warning message:
In tm_map.SimpleCorpus(cleanset, stripwhitespace) :
  transformation drops documents
# term document matrix
tdm <- TermDocumentMatrix(cleanset)
> tdm
<-TermDocumentMatrix (terms: 1894, documents: 1000)>
Non-/sparse entries: 851/1885469
Sparsity: 100%
Maximal term length: 22
Weighting: term frequency (tf)
> tdm <- as.matrix(tdm)
> tdm[1:10, 1:20]
```

Terms	Docs	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
beat		1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

TERM MATRIX PLOT:



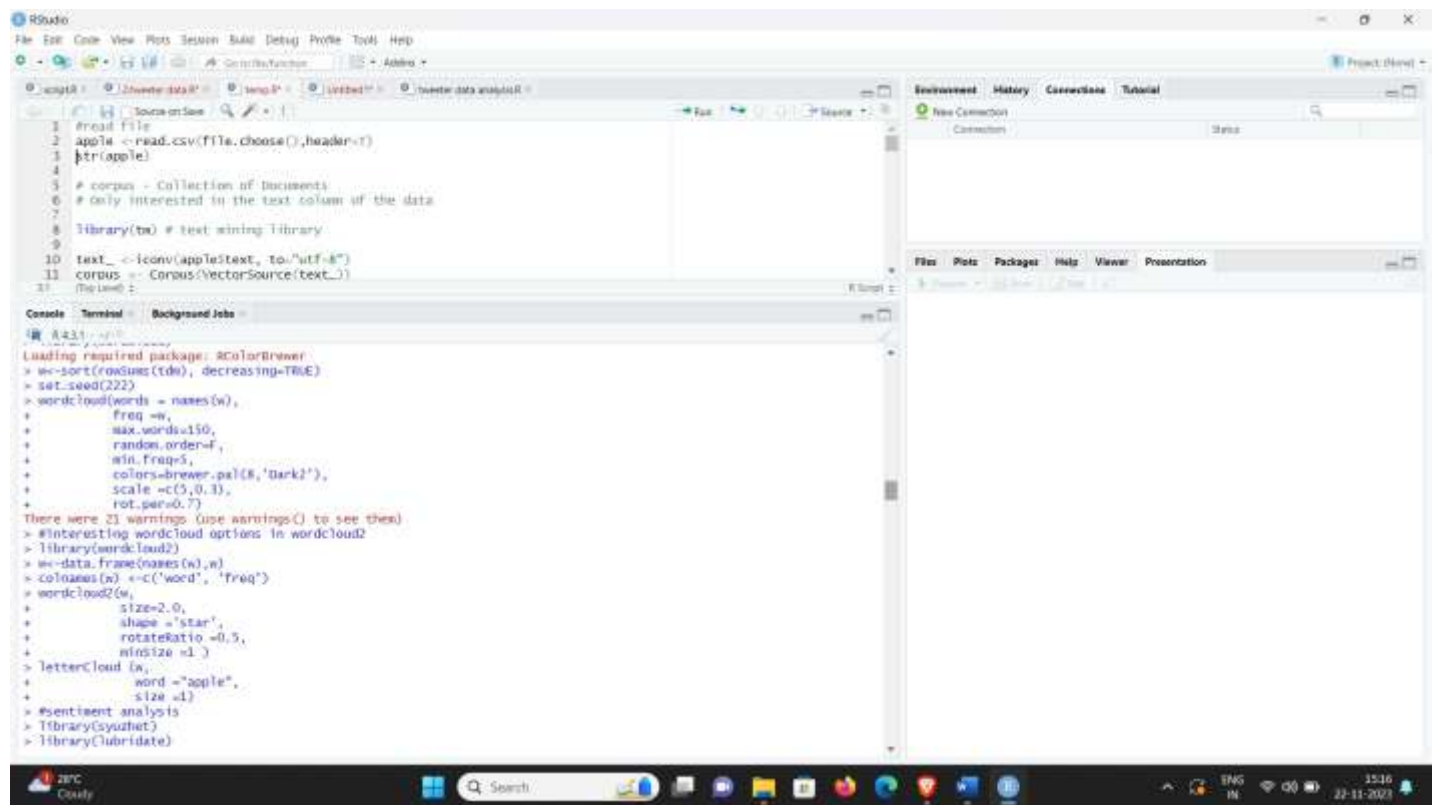
The screenshot shows an RStudio session where a Term Document Matrix (TDM) is created and then plotted. The console output shows the following steps:

```
# Cleanse
cleanset <- tm_map(corpus, stripwhitespace)
Warning message:
In tm_map.SimpleCorpus(cleanset, stripwhitespace) :
  transformation drops documents
# term document matrix
tdm <- TermDocumentMatrix(cleanset)
> tdm
<-TermDocumentMatrix (terms: 1894, documents: 1000)>
Non-/sparse entries: 851/1885469
Sparsity: 100%
Maximal term length: 22
Weighting: term frequency (tf)
> tdm <- as.matrix(tdm)
> tdm[1:10, 1:20]
```

Terms	Docs	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
beat		1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
eps		1	1	0	0	0	1	1	1	0	2	0	0	0	1	0	0	1	1	0	0
est		1	1	0	0	0	2	2	0	0	0	0	0	0	2	0	0	2	2	0	0
https://ohfhaqjib		1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
optionsnipper		1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
raw		1	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
revenues		1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
sees		1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
break		0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
lets		0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

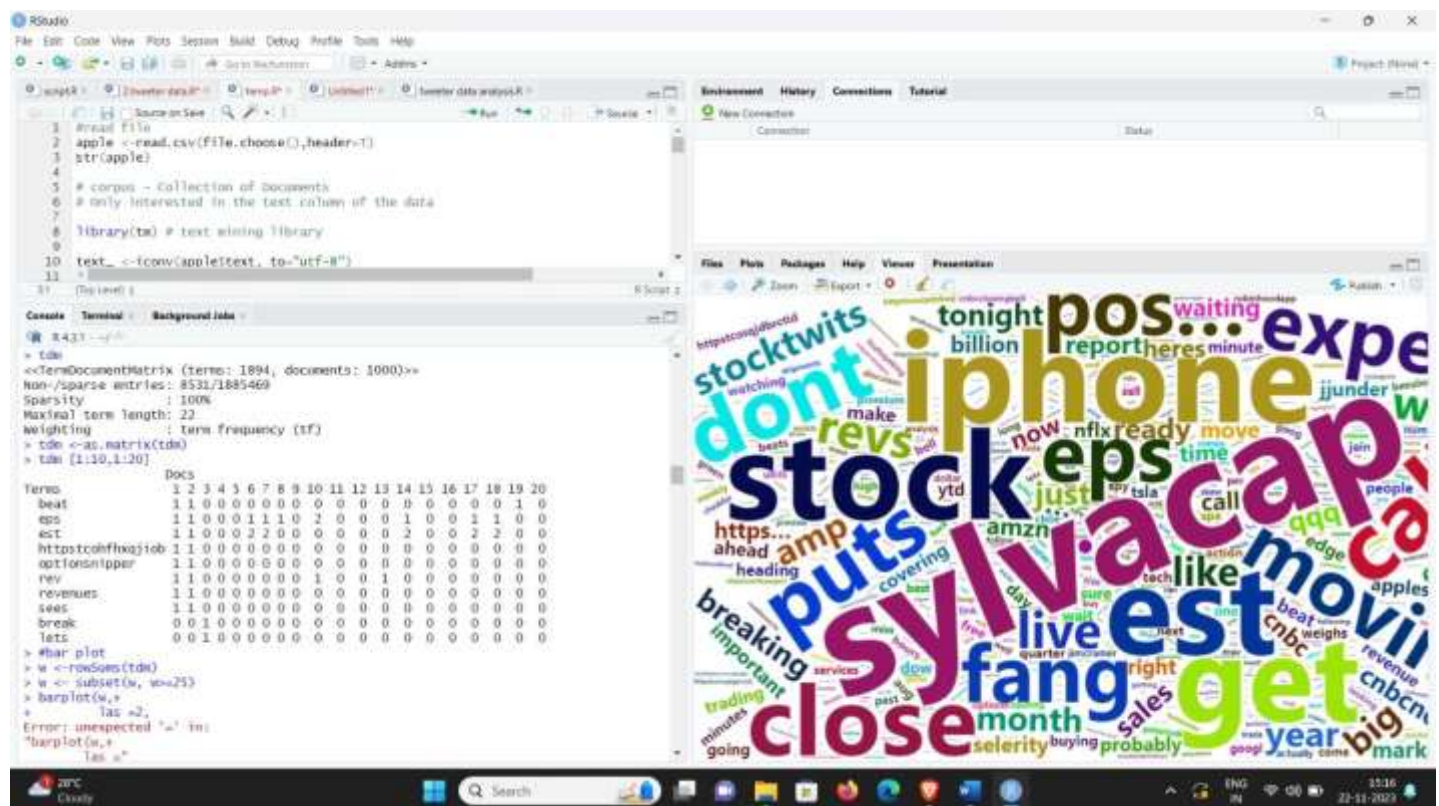
After creating the TDM, the user runs the following commands to create a bar plot:

```
> #bar plot
> w <- rowSums(tdm)
> w <- subset(w, w>25)
> barplot(w)
```

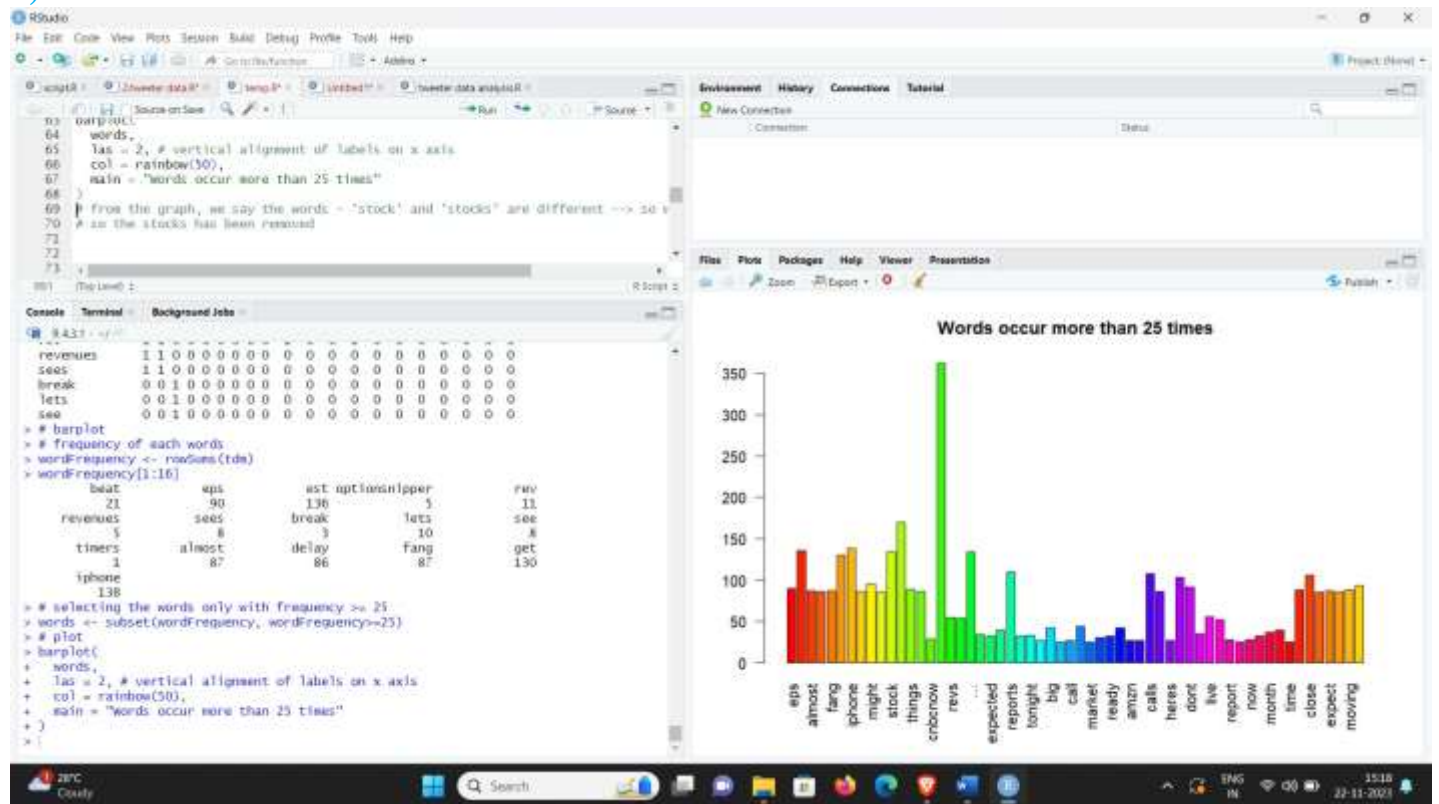
3) DATA PREPROCESSING:

WORD CLOUD:

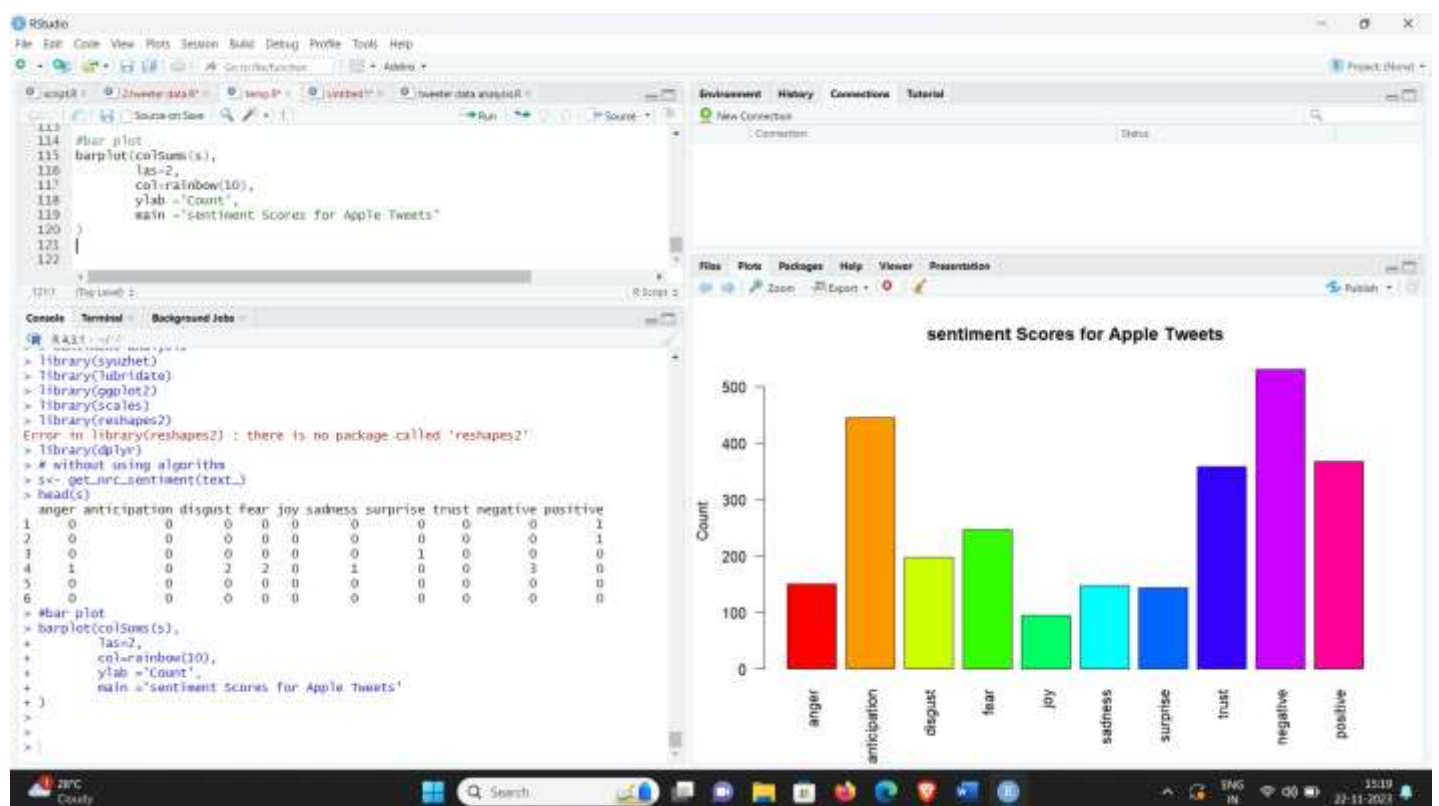


A word cloud centered around the word 'earnings'. Other prominent words include 'stock', 'phone', 'reports', 'might', 'fang', 'expect', 'ugly', 'calls', 'delay', 'anything', 'don't', 'sylvacap', 'puts', 'today', 'things', 'get close', 'pos.', 'means moving', 'investor', 'share', 'earnings', 'report', 'analyst', 'company', 'market', 'investor', 'share', 'earnings', 'report', 'analyst', 'company', 'market'.

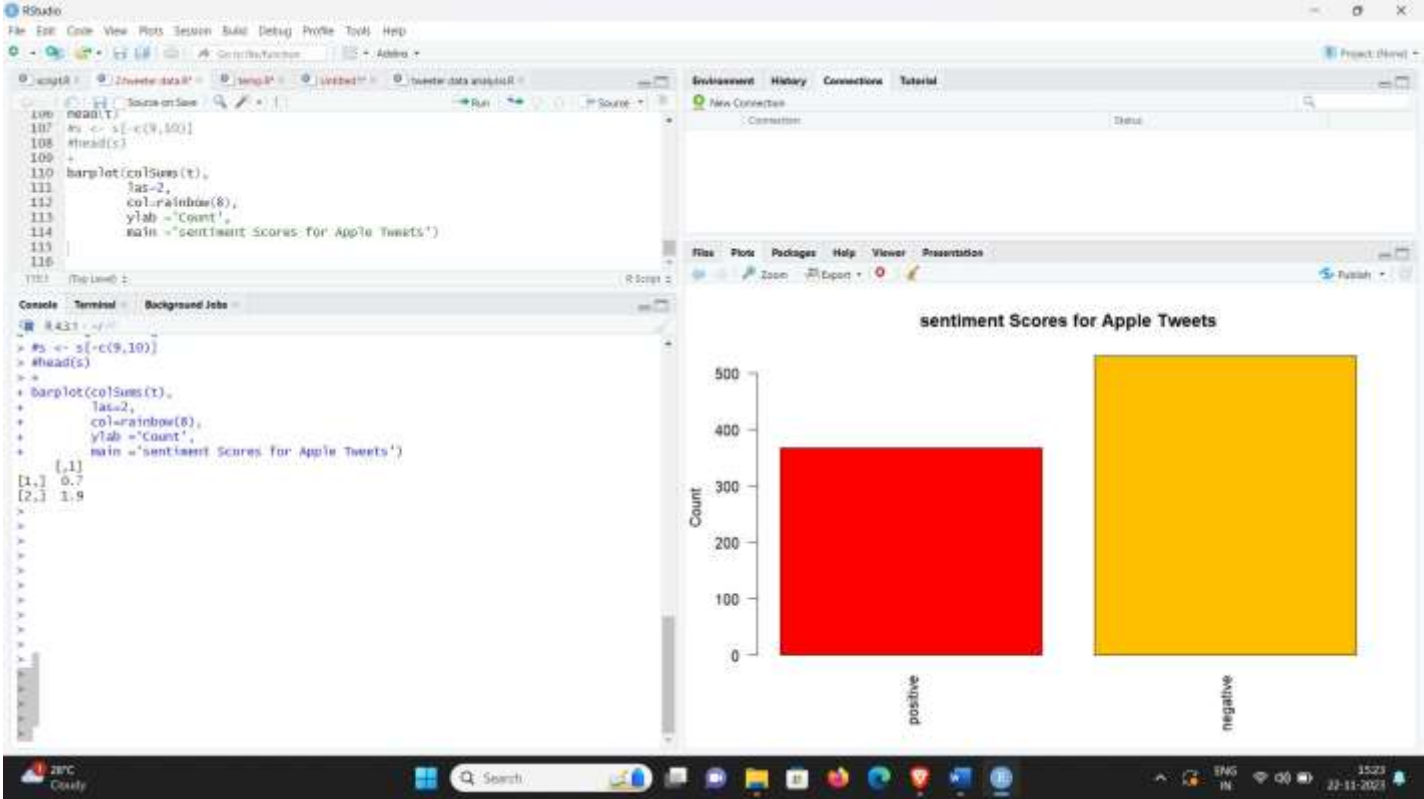
4) SENTIMENT ANALYSIS:



5)lexicon based algorithm - NRC lexicon



NRC LEXICON_sentiment



8.SOURCE CODE:

#read file

```
apple <-read.csv(file.choose(),header=T)
```

```
str(apple)
```

#build corpus

```
library(tm)
```

```
corpus <-iconv(apple$text,to="utf-8")
```

```
corpus <- Corpus(VectorSource(corpus))
```

```
inspect(corpus[1:5])
```

#clean text

```
corpus<-tm_map(corpus, tolower )
```

```
inspect(corpus [1:5])
```

```
corpus<-tm_map(corpus ,removePunctuation)
```

```
inspect(corpus[1:5])
```

```
corpus<- tm_map(corpus , removeNumbers)
```

```
inspect(corpus[1:5])
```

```
cleanset<- tm_map(corpus , removeWords, stopwords('english'))
```

```
inspect(cleanset[1:5])
```

```
removeURL <- tm_map(cleanset, content_transformer(removeURL))
```

```
inspect(cleanset[1:5])
```

```
cleanset <- tm_map(cleanset, removeWords , c('aapl','apple'))
```

```
cleanset <-tm_map(cleanset, gsub,
```

```
    pattern ='stocks',
```

```
    replacement='stock')
```

```
cleanset <-tm_map(cleanset, stripWhitespace)
```

#term document matrix

```
tdm<-TermDocumentMatrix(cleanset)
```

```
tdm
```

```
tdm <- as.matrix(tdm)
tdm [1:10,1:20]
```

#bar plot

```
w <- rowSums(tdm)
w <- subset(w, w >= 25)
barplot(w,
        las = 2,
        col = rainbow(50))
```

#word cloud

```
library(wordcloud)
w <- sort(rowSums(tdm), decreasing = TRUE)
set.seed(222)
wordcloud(words = names(w),
          freq = w,
          max.words = 150,
          random.order = F,
          min.freq = 5,
          colors = brewer.pal(8, 'Dark2'),
          scale = c(5, 0.3),
          rot.per = 0.7)
```

#interesting wordcloud options in wordcloud2

```
library(wordcloud2)
w <- data.frame(names(w), w)
colnames(w) <- c('word', 'freq')
wordcloud2(w,
          size = 0.5,
          shape = 'star',
          rotateRatio = 0.5,
          minSize = 1 )
letterCloud (w,
          word = "apple",
          size = 1)
```

#sentiment analysis

```
library(syuzhet)
library(lubridate)
library(ggplot2)
library(scales)
library(dplyr)
```

#read file

```
apple <- read.csv(file.choose(), header =T)  
tweets <-iconv(apple$text, to= 'utf-8')
```

#obtain sentiment scores

lexicon based algorithm - NRC lexicon

```
s<- get_nrc_sentiment(tweets)  
head(s)  
#bar plot  
barplot(colSums(s),  
        las=2,  
        col=rainbow(10),  
        ylab ='Count',  
        main ='sentiment Scores for Apple Tweets')
```

```
#tweets[4]
```

```
#get_nrc_sentiment('delay')
```

```
t <- data.frame(  
  positive = s$positive,  
  negative = s$negative  
)
```

```
head(t)
```

```
#s <- s[-c(9,10)]
```

```
#head(s)
```

```
barplot(colSums(t),  
        las=2,  
        col=rainbow(10),  
        ylab ='Count',  
        main ='sentiment Scores for Apple Tweets')
```

9.ADVANTEGES:

- 1. Open-source and cost-effective:** R is a free and open-source programming language, making it accessible to a wide range of users, including students, researchers, and businesses. This eliminates the need for expensive software licenses, making it a cost-effective solution for Twitter data analysis.
- 2. Versatility and flexibility:** R offers a vast collection of packages specifically designed for text mining and sentiment analysis, providing users with a variety of tools and techniques to analyze Twitter data. This flexibility allows users to customize their analysis to suit their specific needs and research questions.
- 3. Reproducibility and transparency:** R scripts are easily reproducible, allowing others to replicate the analysis and verify the results. This transparency promotes scientific rigor and collaboration.
- 4. Powerful data visualization:** R provides a rich set of data visualization tools that can be used to effectively communicate the results of sentiment analysis. These visualizations can help users identify patterns, trends, and insights from Twitter data.
- 5. Integration with other tools:** R can be easily integrated with other data analysis and visualization tools, allowing users to combine different techniques and create comprehensive analyses. This integration enhances the power and scope of Twitter data analysis.
- 6. Active community support:** R has a large and active community of users and developers who provide support, share knowledge, and contribute to the development of new packages. This community support ensures that users can find

solutions to their problems and stay up-to-date with the latest advancements in sentiment analysis.

10. CONCLUSION:

The task of big data analysis is not only important but also a necessity. In fact many organizations that have implemented Big Data are realizing significant competitive advantage compared to other organizations with no Big Data efforts. The project is intended to analyse the Twitter Big Data and come up with significant insights which cannot be determined otherwise. As twitter post are very important source of opinion on different issues and topics. It can give a keen insight about a topic and can be a good source of analysis. Analysis can help in decision making in various areas.

Sentiment analysis of Twitter data using R can be an effective approach to understand the emotional tone of tweets and gain insights from large volumes of data. By leveraging R's powerful libraries for data collection, data pre-processing, sentiment analysis, feature engineering, model training, model evaluation, model deployment, visualization, and interpretation, a comprehensive system can be developed for sentiment analysis of Twitter data.

11. REFERENCES:

- I. Boumaiza, A. (2015, November). A Survey on Sentiment Analysis and Visualization
- II. Salarpour, A., Bamneshin, M. H., & Proios, D. (2014, November). Sentiment Analysis and Visualization of Social Media Data.
- III. Urologin, S. (2018, August). Sentiment Analysis, Visualization and Classification of Summarized News Articles: A Novel Approach.
- IV. Shetty, S. D. (2021). Sentiment Analysis, Tweet Analysis and Visualization on Big Data Using Apache Spark and Hadoop.
- V. Saragadam, V. L., & Parasuraman, S. (2022, March 10). Visualization of Real Time Big Data Through Sentiment Analysis.
- VI. Keerthana, V., Prasannakumar, P., Ebenezer Abishek.B, Dr., Stephen, C. A., & Vijayalakshmi, Dr. A. (2021). Data filtering and visualization for sentiment analysis of e-commerce website.
- VII. Gundla, A. V., & Otari, M. S. (2015, September). A Review on Sentiment Analysis and Visualization of Customer Reviews.
- VIII. Saini, S., Shukla, V. K., Punhani, R., & Bathla, R. (April, 2019). Sentiment Analysis on Twitter Data using R for Understanding Perspectives on E-Healthcare.
- IX. Ray, P., & Chakrabarti, A. (February, 2017). Sentiment Analysis Framework for Twitter Data using R: A Lexicon-Based Approach.
- X. Srikanth, J., Damodaram, A., Teekaraman, Y., Kuppusamy, R., & Thelkar, A. R. (2022, May 6). Sentiment Analysis on COVID-19 Twitter Data Streams Using Deep Belief Neural Networks.

XI. Gayathri, Poonilavu, K., Navithra, & Rajeshwari. (2022, October 3).

Sentiment Knowledge Discovery in Twitter Live Streaming Data using R Language.

XII. Al-Hashedi, A., Al-Fuhaidi, B., Mohsen, A. M., Ali, Y., Al-Kaf, H. A. G., Al-Sorori, W., & Maqtary, N. (2022, January 13). Machine Learning Model for Arabic Sentiment Analysis of COVID-19 Conspiracy Theories on Twitter.