Extensive sentiment and text analysis

Project Proposal: "How can identifying hate speech in tweets help businesses better manage their brand reputation online?"

Importance: By identifying and mitigating hate speech, Twitter can create a safer and more positive user environment, thereby protecting its brand reputation and user engagement.

Part1

Steps and Analysis:

Basic Cleaning

- The code loads various libraries needed for text processing and visualization.
- A **clean_tweet** function is defined to preprocess the tweets by removing user mentions, URLs, hashtags, special characters, and extra whitespaces. It also converts text to lowercase, expands contractions, corrects common slang, and handles emojis.

Removing Stop Words

- The tweets are tokenized into words, and stopwords (common words like "the", "and", etc.) are removed.
- The cleaned tweets are reconstructed by concatenating the remaining words.
- The cleaned tweets without stopwords are merged back into the original dataframe.
- The length of each cleaned tweet is calculated, and non-finite values are handled.

Library: syuzhet package

Granularity: Word-level sentiment analysis

Sentiment Detection: Primarily focuses on determining the overall sentiment of text (positive, negative, or neutral)

Output: Provides a numeric sentiment score for the entire text

Syuzhet sentiment analysis

- A function classify_sentiment is defined to classify each tweet as positive, negative, or neutral based on its sentiment score using the syuzhet package.
- The function is applied to each cleaned tweet, and the predicted sentiment is added to the dataframe.
- The syuzhet method uses the NRC Emotion Lexicon to calculate sentiment scores.

- It assigns a numeric sentiment score to each tweet based on the presence of positive and negative words.
- The sentiment score is then used to classify the tweet as positive, negative, or neutral.

Word Clouds

- Tweets are filtered into positive, negative, and neutral based on their predicted sentiment.
- Term-document matrices are created for each sentiment category.
- Word frequencies are calculated and sorted.
- Word clouds are generated for each sentiment category to visualize the most frequent words.

positive tweets

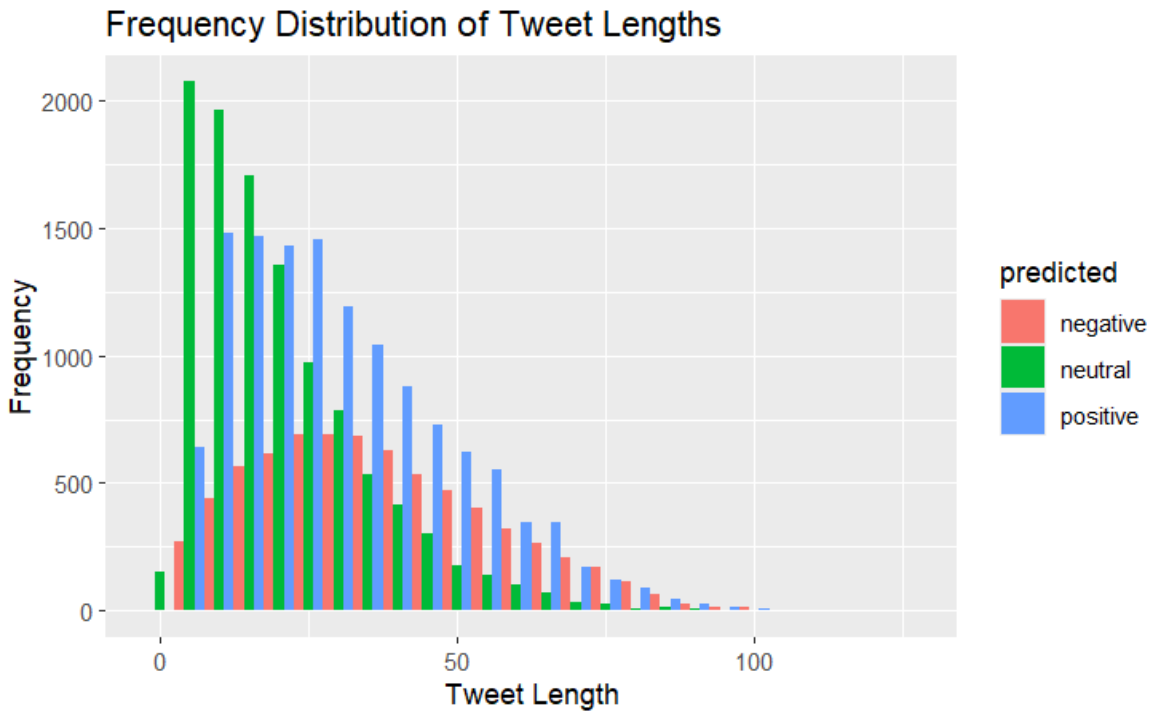
negative tweets

neutral tweets

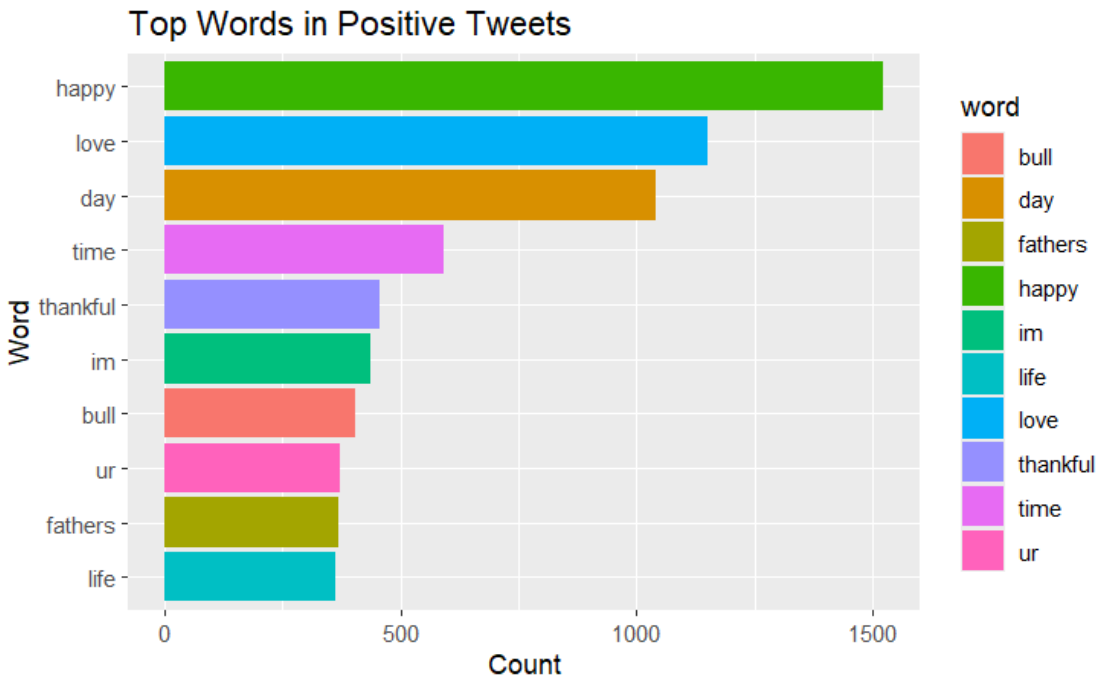
Frequency Distribution of Tweet Lengths

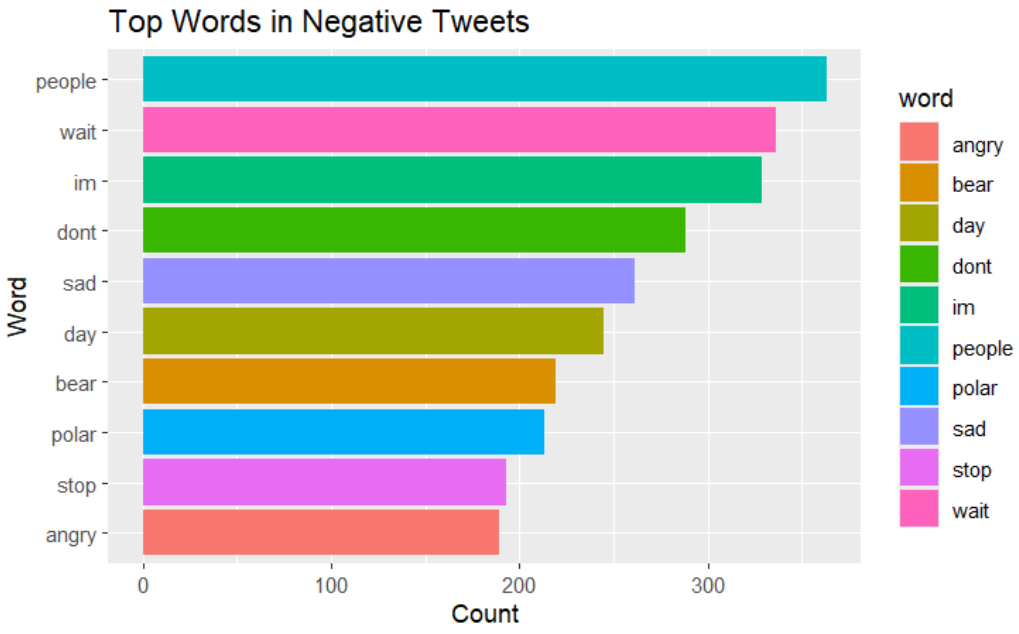
A histogram is created to visualize the distribution of tweet lengths, colored by the predicted sentiment.

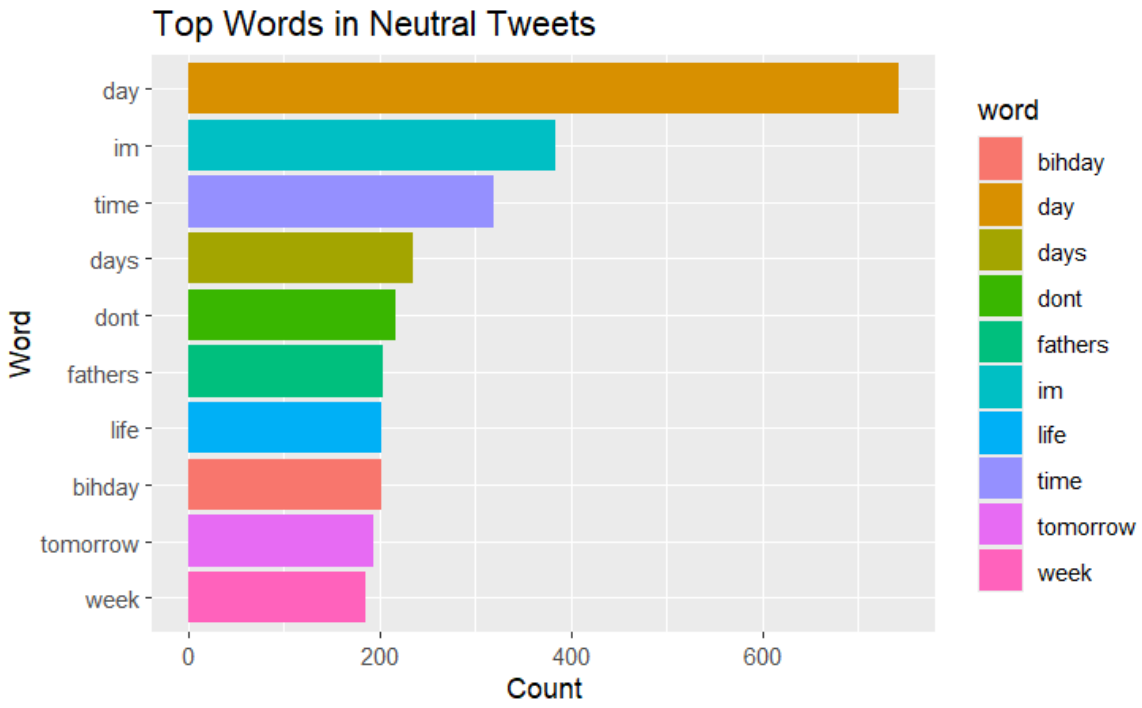
This plot provides us the length of each type of tweet classified by sentiment. This gives us an insight about how each tweet can be possibly classified into just based on length.



most common words in positive, negative, and neutral tweets are counted and visualized using bar plots.







Part 2

- This method uses custom word associations derived from your text data.
- Words are manually categorized into positive, negative, and neutral based on their associations.
- The sentiment of a tweet is determined by counting the occurrences of these associated words.
- The tweet is classified as positive, negative, or neutral based on the highest count of associated words.

---

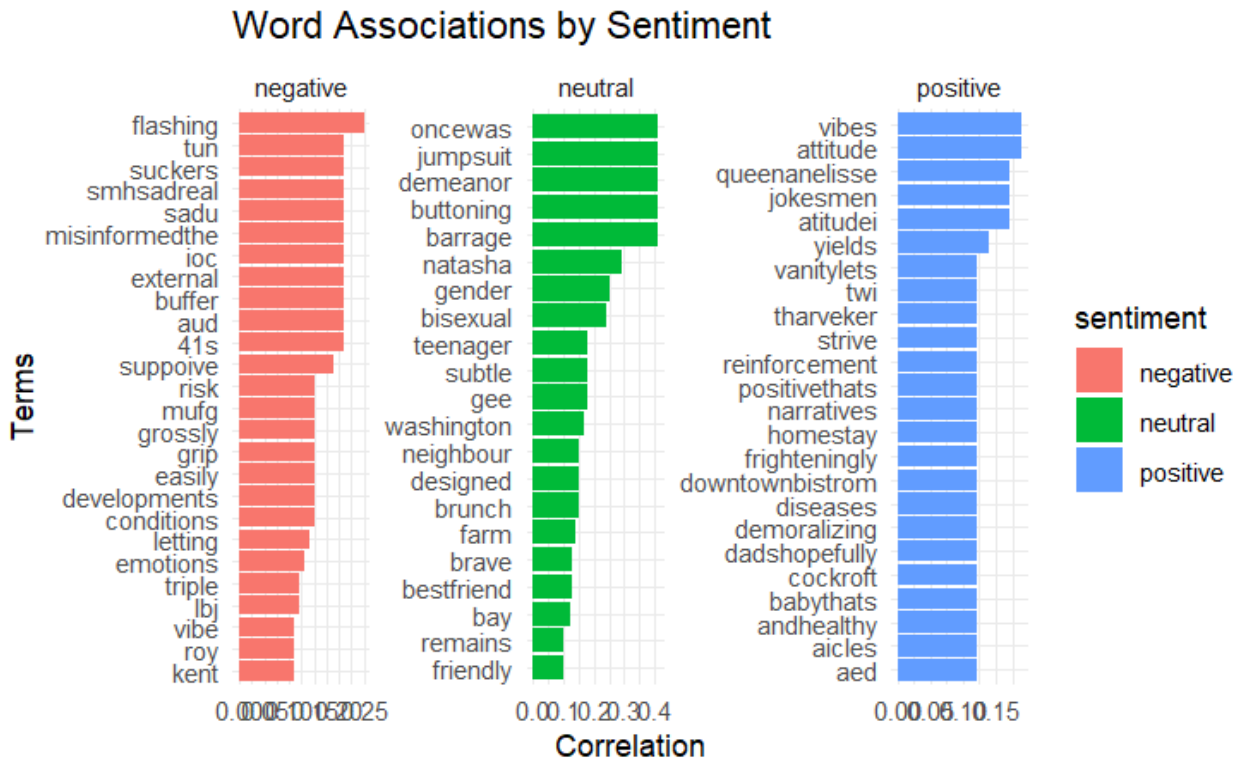
diff between syuzhet and word association sentiment

syuzhet Method: Useful for a general sentiment analysis where a predefined lexicon is sufficient. It provides a quick and straightforward way to classify sentiment and can be useful for initial analyses or when working with standard text data.

Custom Classification: Useful for more specific and nuanced sentiment analysis tailored to your dataset. It allows you to incorporate domain-specific knowledge and capture sentiment more accurately in specialized contexts.

---

visualize word associations by sentiment



intuition

- Once we have our lists of sentiment-associated words, we can analyze any new piece of text. The intuition is to look for the presence of these words in the text and count how many positive, negative, and neutral words there are.
- By comparing these counts, we can determine the overall sentiment of the text. If there are more positive words, the text is classified as positive. If there are more negative words, it's classified as negative. If neither positive nor negative words dominate, the text is classified as neutral.

Comparison with syuzhet and word association

- **Method 1 syuzhet:**
- Predicted Sentiment: "positive"

- Both methods would classify the tweet as "positive," but the second method also provides the exact score, offering more detail about the sentiment's intensity.
- **Method 2 word association:**
- For a tweet with a numeric sentiment score of 0.8:
- Sentiment Score: 0.8
- Predicted Sentiment: "positive"

Calculate Correlations:

The find Assocs function calculates the correlation between the specified terms and all other terms in the DTM.

Correlation measures how changes in the frequency of one term relate to changes in the frequency of another term. High correlation indicates that the terms frequently appear together in the same documents.

1. **Threshold for Correlation**:
    - The **corlimit** parameter sets a threshold for the minimum correlation value to consider. Only associations with a correlation value above this threshold are included in the results.
    - In this case, **corlimit = 0.1** means only associations with a correlation higher than 0.1 will be considered.

Intuition Behind Word Associations

- **Co-Occurrence**: Words that frequently appear together in the same documents are likely to be related in some meaningful way. For example, in tweets, words like "happy" and "joy" might co-occur frequently, indicating a strong association.
- **Contextual Understanding**: By analyzing which words are associated with terms like "positive", "negative", and "neutral", we can gain insights into the context in which these sentiments are expressed. For instance, words associated with "positive" might include "happy", "love", "great", etc., while words associated with "negative" might include "sad", "angry", "bad", etc.

Example

Suppose we have the following tweets:

1. "I am very happy with the service."
2. "The weather is bad and makes me sad."

3. "Love this beautiful day!"

After creating the DTM, we might get something like this:

	happy	sad	love	service	weather	bad	beautiful	day
Tweet 1	1	0	0	1	0	0	0	0
Tweet 2	0	1	0	0	1	1	0	0
Tweet 3	0	0	1	0	0	0	1	1

Now, let's find associations for the term "happy":

- The term "happy" has a correlation with terms that appear in the same documents as "happy".
- If "happy" and "service" frequently appear together in the same tweets, the correlation between "happy" and "service" will be high.

Output Interpretation

```r
print(associations)
```

The output of `print(associations)` might look like this:

```r
$positive
        vibes   attitude  happiness   great
        0.19     0.17     0.15        0.12

$negative
        sad     bad       terrible    angry
        0.21    0.20      0.18        0.15

$neutral
        day     weather   average     normal
        0.22    0.20      0.18        0.16
```

- **$positive**: Words like "vibes", "attitude", "happiness", and "great" are associated with "positive" with the given correlation values.
- **$negative**: Words like "sad", "bad", "terrible", and "angry" are associated with "negative".
- **$neutral**: Words like "day", "weather", "average", and "normal" are associated with "neutral".

Sentiment Classification Function

1. **Split Text into Words**: Each tweet is split into individual words.
2. **Count Sentiment Words**: The function counts how many of these words are in the predefined lists of positive, negative, and neutral words.
3. **Compare Scores**: The function compares the counts (scores) to determine which sentiment is most represented in the tweet.
4. **Classify Sentiment**: Based on the highest score, the tweet is classified as positive, negative, or neutral.
5. **Apply to Dataset**: This classification is applied to each tweet in the dataset, and the results are stored in a new column.

Part 3: sentimentr

1. **Calculating Sentiment Scores**:

---

sentiment_scores <- sentiment_by(train_data$cleaned_tweet_no_stop)

- The **sentiment_by** function is applied to the **cleaned_tweet_no_stop** column of **train_data**. This function calculates sentiment scores for each tweet.
- **Intuition**:
    - **sentiment_by** breaks down each tweet into its component sentences.
    - It then calculates a sentiment score for each sentence by analyzing the presence of positive and negative words.
    - The scores for each sentence are aggregated to provide an overall sentiment score for each tweet.
- The resulting **sentiment_scores** object contains sentiment scores for each tweet, including additional metadata.

2. **Extracting Average Sentiment Scores**:

---

train_data$sentimentr_score <- sentiment_scores$ave_sentiment

---

- The **ave_sentiment** column from the **sentiment_scores** object is extracted and added to the **train_data** dataframe as a new column **sentimentr_score**.
- **Intuition**:
    - The **ave_sentiment** column represents the average sentiment score for each tweet, calculated by averaging the sentiment scores of its component sentences.
    - A positive **sentimentr_score** indicates an overall positive sentiment, a negative score indicates an overall negative sentiment, and a score around zero indicates a neutral sentiment.

Detailed Breakdown

1. **Sentence-Level Analysis**:
    - The **sentimentr** package is designed to analyze sentiment at the sentence level, which allows for more nuanced sentiment analysis compared to word-level methods.
    - By breaking down the text into sentences, the package can capture the sentiment of complex texts where different sentences might express different sentiments.
2. **Sentiment Calculation**:
    - The sentiment score for each sentence is calculated based on the presence and intensity of positive and negative words.

- These scores are influenced by context, negations, and amplifiers (e.g., "very", "not"), which modify the sentiment intensity.

3. **Aggregation**:
    - The scores of individual sentences within a tweet are aggregated to produce an average sentiment score for the entire tweet.
    - This aggregation provides a more balanced view of the sentiment, especially for longer texts with mixed sentiments.

Differences with syuzhet and word association sentiment

---

1. Sentiment Analysis Using syuzhet Method:

Approach: Uses the NRC Emotion Lexicon, a predefined lexicon to assign sentiment scores to words and calculate the overall sentiment of a text.

Granularity: Analyzes sentiment at the word level and aggregates the scores to produce a sentiment score for the entire tweet.

Classification: Sentiment scores are categorized as positive, negative, or neutral based on their numeric values.

2. Word Association-Based Sentiment Classification Method:

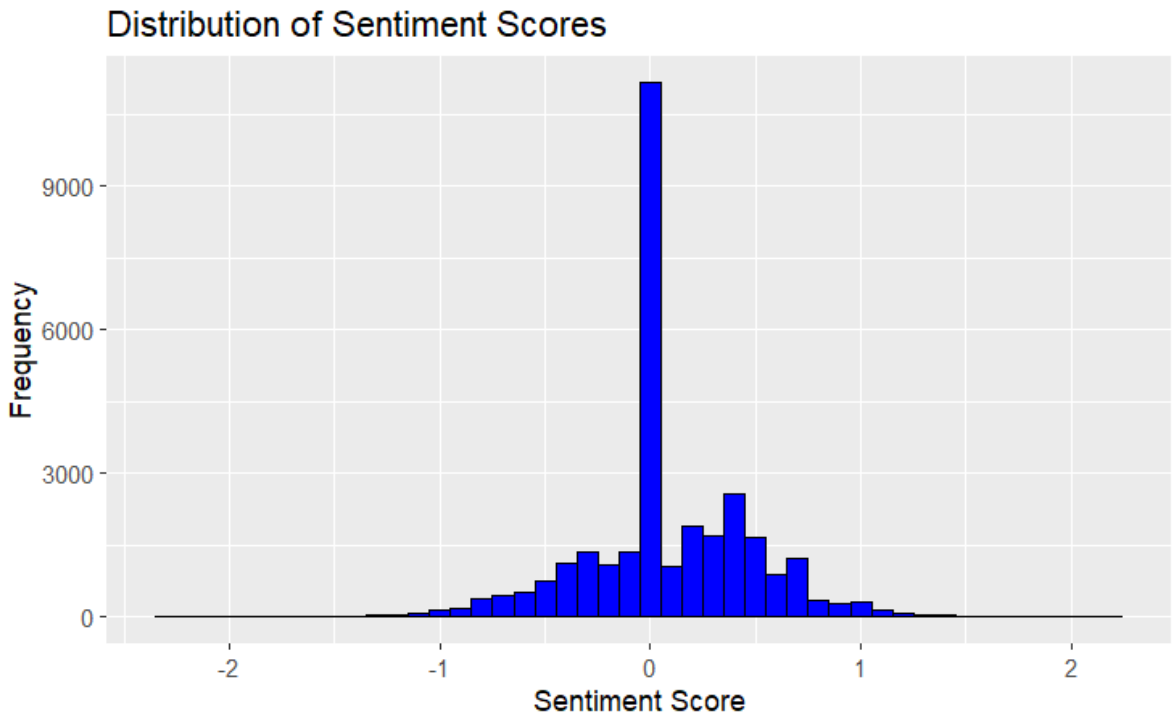
Approach: Uses custom word lists derived from the dataset to classify sentiments based on word associations.

Granularity: Counts the occurrences of positive, negative, and neutral words in each tweet to determine the overall sentiment.

Classification: Compares the counts of sentiment-associated words to classify the sentiment of each tweet.

---

sentiment score distribution:

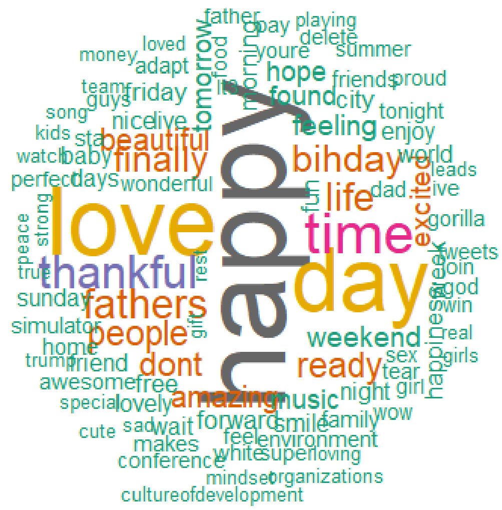


Insights from the Histogram

1. **Dominance of Neutral Sentiment**:
    - The peak at zero suggests that a large portion of the tweets are neutral, with neither strong positive nor strong negative sentiment.
2. **Balanced Sentiment Distribution**:
    - The symmetrical shape indicates that there are roughly equal numbers of positive and negative tweets, reflecting a balanced sentiment distribution in the dataset.
3. **Sentiment Extremes**:
    - There are fewer tweets with extreme sentiment scores (very high positive or very low negative), indicating that most tweets are moderate in their sentiment.

```
negative  neutral positive
    7576    10628    12476
```

- **Positive**: If the sentiment score (**sentimentr_score**) is greater than 0, the tweet is classified as "positive".
- **Negative**: If the sentiment score is less than 0, the tweet is classified as "negative".
- **Neutral**: If the sentiment score is exactly 0, the tweet is classified as "neutral".


happy
love
day
time
thankful
fathers
bihday
life
finally
beautiful
people
dont
ready
weekend
amazing
music
feeling
hope
found
tomorrow
morning
excited
happiness
week
father
pay
playing
delete
youre
summer
friends
proud
city
tonight
enjoy
world
leads
live
gorilla
tweets
join
god
win
real
girls
sex
tear
girl
wow
night
family
smile
forward
feel
environment
white
super
loving
mindset
organizations
cultureofdevelopment
conference
makes
wait
sad
cute
lovely
free
special
awesome
friend
trump
home
simulator
sunday
true
peace
strong
perfect
watch
kids
song
guys
team
money
loved
adapt
friday
nice
baby
days
wonderful
food
fun
dad
rest
gift


bull
people
wait
sad
dont
day
bear
dominate
polar
climb
angry
time
stop
bad
racist
life
miss
hate
black
orlando
cold
trump
fuck
racing
lost
youre
news
pay
home
night
live
leave
wrong
hard
week
women
racism
gun
happy
dead
didnt
love
friends
girls
play
watching
strong

---

difference between syuzhet and get_nrc_sentiment

1. syuzhet Method

***Approach**:*

***Library**: **syuzhet** package*

***Granularity**: Word-level sentiment analysis*

***Sentiment Detection**: Primarily focuses on determining the overall sentiment of text (positive, negative, or neutral)*

***Output**: Provides a numeric sentiment score for the entire text*

2. get_nrc_sentiment Method

***Approach**:*

***Library**: **syuzhet** package using the NRC Emotion Lexicon*

***Granularity**: Word-level emotion analysis*

***Sentiment Detection**: Detects a wide range of emotions (anger, anticipation, disgust, fear, joy, sadness, surprise, trust) along with overall positive and negative sentiments*

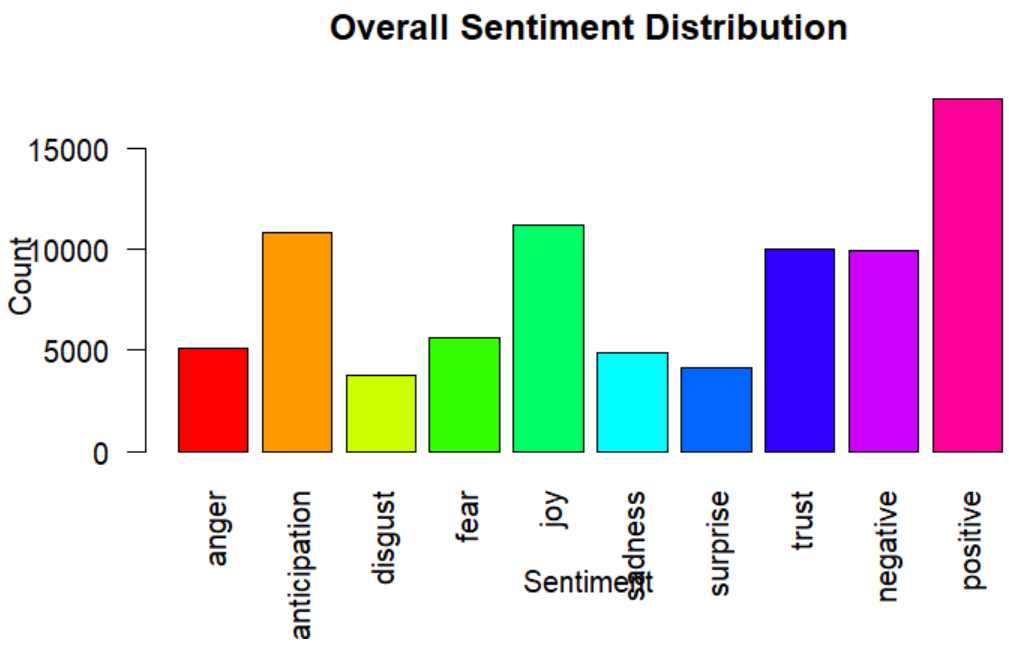
***Output**: Provides counts for each detected emotion and sentiment for the entire text*

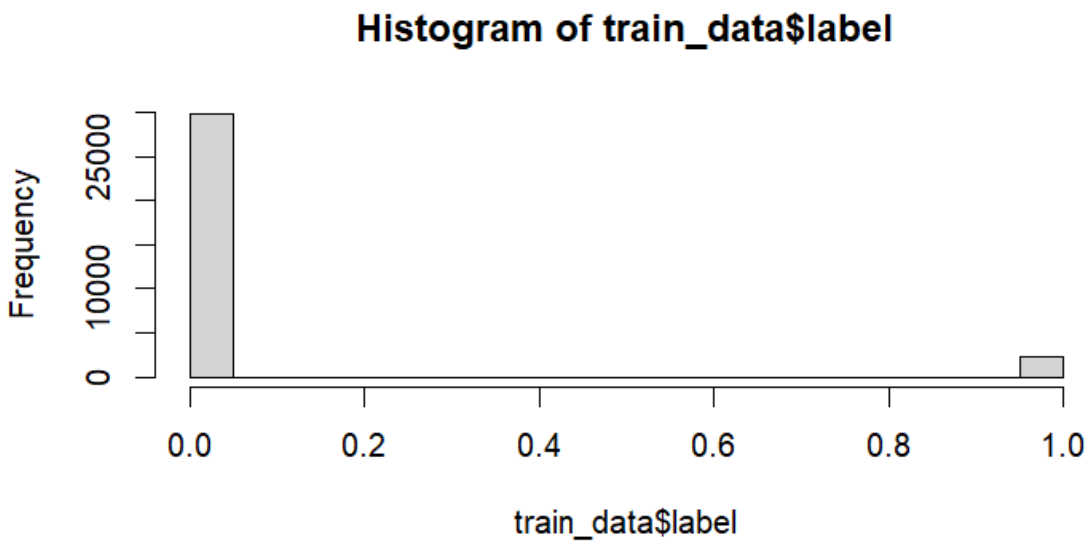
While both methods serve the purpose of sentiment analysis, they cater to different needs. The `syuzhet` method is more straightforward and focuses on overall sentiment, making it suitable for applications where a general sentiment overview is needed. On the other hand, the `get_nrc_sentiment` method provides a detailed emotional breakdown, making it suitable for applications that require an in-depth understanding of various emotions present in the text.

Emotion Analysis

get_nrc_sentiment Function: This function calculates the presence of specific emotions and sentiments in each tweet using the NRC Emotion Lexicon. The emotions include anger, anticipation, disgust, fear, joy, sadness, surprise, and trust, along with overall positive and negative sentiments.

nrc_sentiment Dataframe: The function returns a dataframe where each row corresponds to a tweet, and each column represents the count of words associated with a particular emotion or sentiment.





Robust analysis to identify sentiments of tweets.
Valuable insights for twitter business to handle ethical implications of their platform

Awareness for Investors