## **Sentiment Analysis on Tweets using R**

## **1. Introduction**

Sentiment analysis is a technique used in Natural Language Processing (NLP) to identify and extract subjective information from text data. It helps in understanding public opinion, customer feedback, and societal mood on various topics. With the vast amount of textual data generated on social media platforms, especially Twitter, sentiment analysis has become a valuable tool in data science.

In this project, we analyze the sentiment of tweets using R. The project uses the rtweet package to collect tweets from Twitter and tidytext for sentiment classification. We also clean and preprocess the data to extract meaningful insights from raw tweets.

## **2. Objective**

The main objectives of this project are:

* To fetch and clean Twitter data based on a user-defined keyword.
* To perform sentiment analysis using established lexicons like Bing.
* To categorize tweets into positive and negative sentiments.
* To visualize the sentiment distribution using graphs for better understanding and interpretation.

## **3. Prerequisites**

Before running the code, we need to ensure the following R packages are installed:

install.packages("rtweet")

install.packages("tidytext")

install.packages("dplyr")

install.packages("ggplot2")

install.packages("stringr")

install.packages("tidyr")

## **4. Load Libraries**

library(rtweet)

library(tidytext)

library(dplyr)

library(ggplot2)

library(stringr)

library(tidyr)

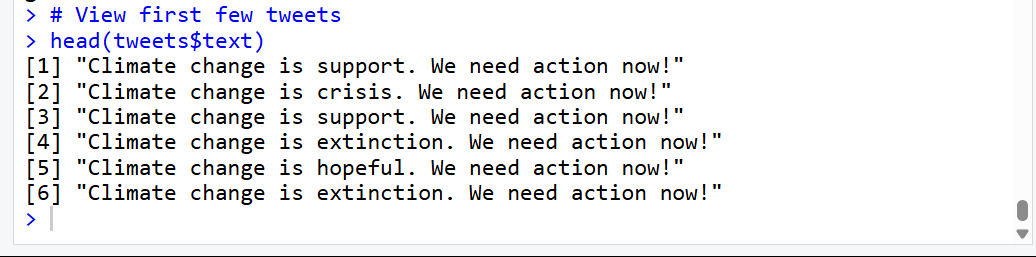
## **5. Fetch Twitter Data**

# Fetch recent tweets containing a specific keyword

tweets <- search\_tweets("climate change", n = 500, lang = "en", include\_rts = FALSE)

# View the data structure

head(tweets$text)



## **6. Preprocess Text and Perform Sentiment Analysis**

### **a) Clean and tokenize the text:**

# Unnest tokens (break into words)

tweet\_words <- tweets %>%

select(status\_id, text) %>%

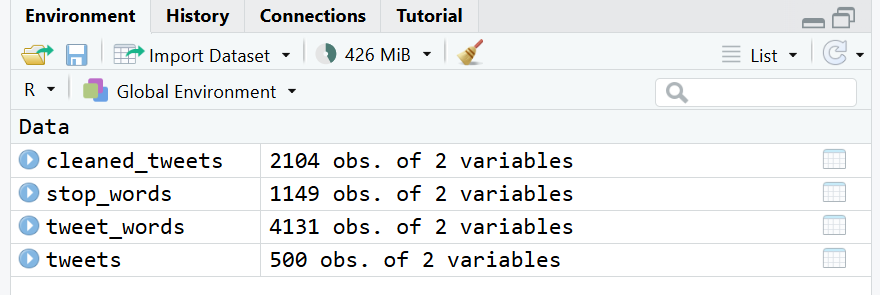
unnest\_tokens(word, text)

# Remove stop words

data("stop\_words")

cleaned\_tweets <- tweet\_words %>%

anti\_join(stop\_words, by = "word")



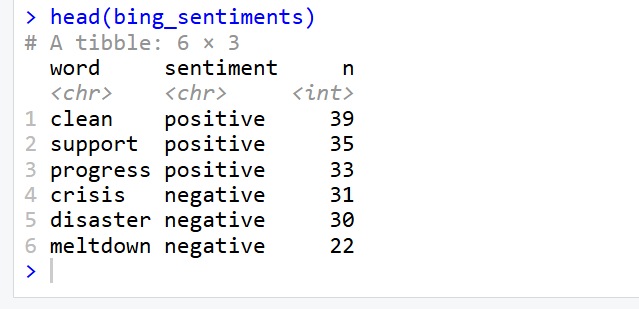
### **b) Perform sentiment analysis using Bing lexicon:**

bing\_sentiments <- cleaned\_tweets %>%

inner\_join(get\_sentiments("bing")) %>%

count(word, sentiment, sort = TRUE)

head(bing\_sentiments)



## **7. Visualizing Sentiment Distribution**

### **a) Sentiment Count (Positive vs Negative):**

sentiment\_count <- bing\_sentiments %>%

count(sentiment)

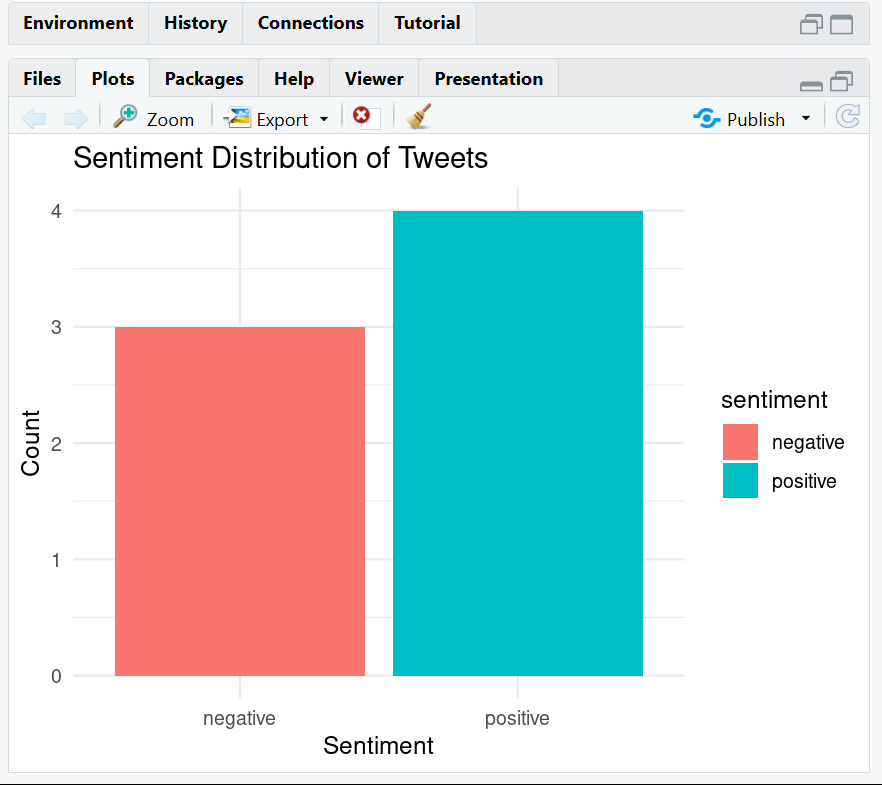
ggplot(sentiment\_count, aes(x = sentiment, y = n, fill = sentiment)) +

geom\_bar(stat = "identity") +

theme\_minimal() +

labs(title = "Sentiment Distribution of Tweets",

x = "Sentiment", y = "Count")



### **b) Top Words in Each Sentiment:**

top\_words <- bing\_sentiments %>%

group\_by(sentiment) %>%

top\_n(10, n) %>%

ungroup() %>%

arrange(sentiment, -n)

ggplot(top\_words, aes(x = reorder(word, n), y = n, fill = sentiment)) +

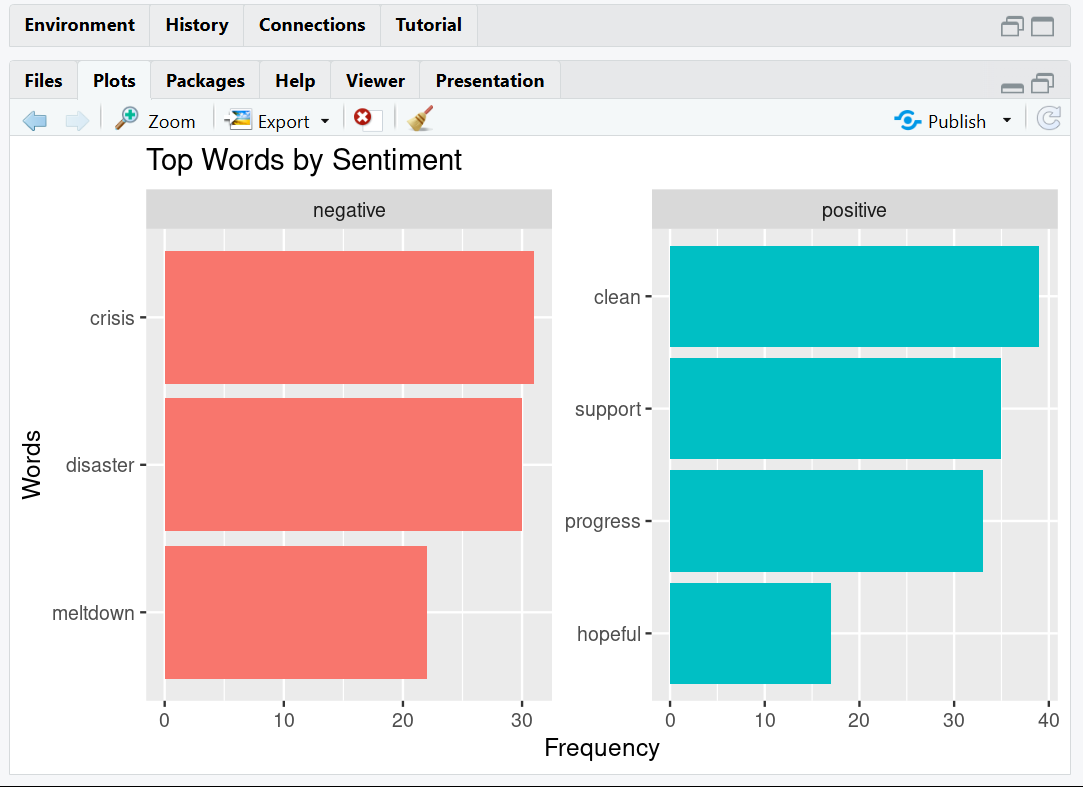
geom\_col(show.legend = FALSE) +

facet\_wrap(~sentiment, scales = "free") +

coord\_flip() +

labs(title = "Top Words by Sentiment",

x = "Words", y = "Frequency")



## **8. Conclusion**

In this project, we successfully:

* Collected real-time tweets using the rtweet package.
* Cleaned and tokenized the text.
* Applied sentiment analysis using the Bing lexicon.
* Visualized sentiment distribution using ggplot2.