SENTIMENT ANALYSIS ON WHATSAPP CHATS

Insight to Emotional tones and Emojis

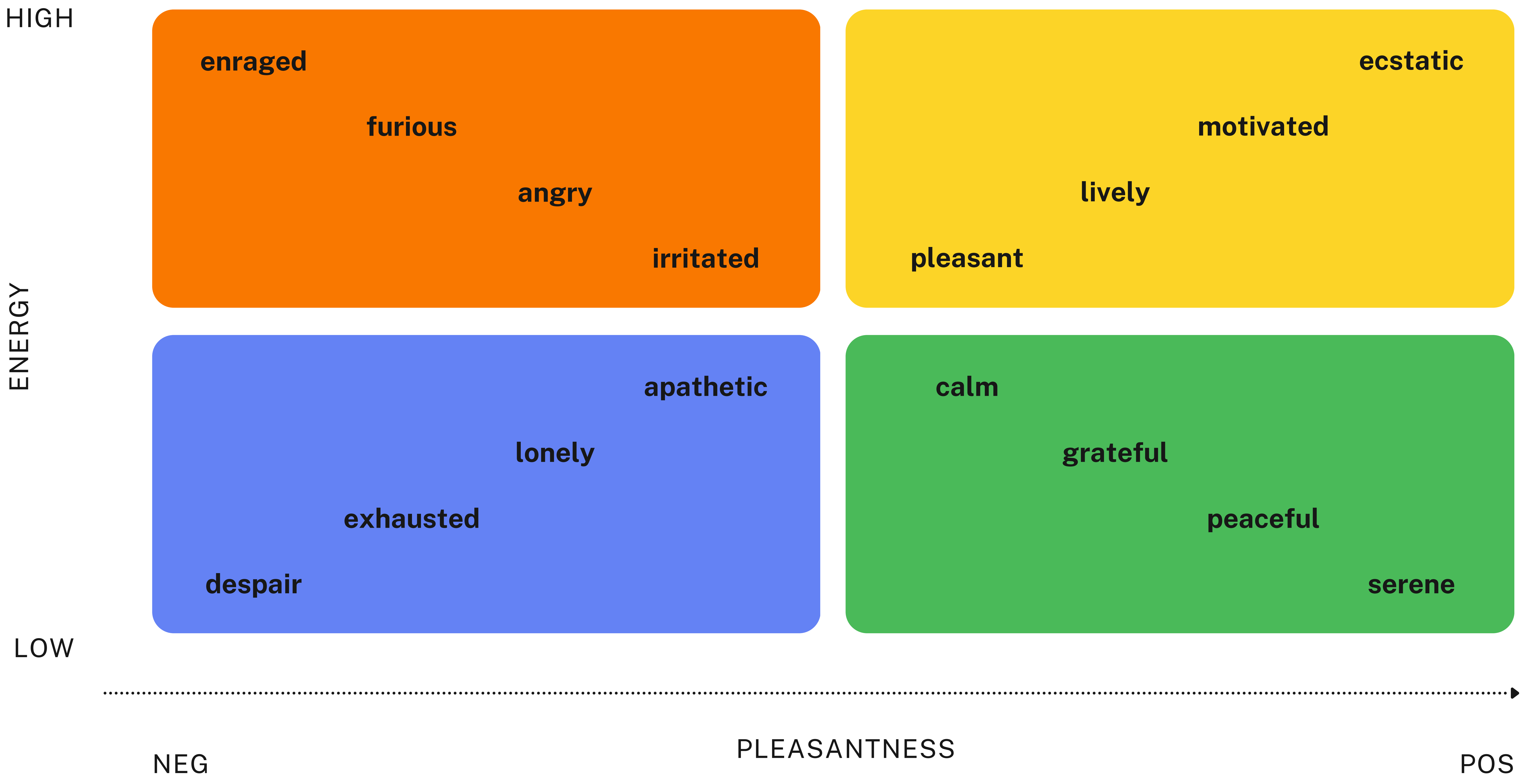
PROBLEM STATEMENT

WhatsApp-Analyzer is a statistical analysis tool for WhatsApp chats. Working on the chat files that can be exported from WhatsApp it generates various plots showing, for example, which other participant a user responds to the most.

We propose to employ dataset manipulation techniques to have a better understanding of WhatsApp chat present in our phones.

OBJECTIVE

This project entitled, "Sentiment Analysis on Whatsapp Chats" using R Tool aims to analyse the WhatsApp group chat conversations of people and measure the sentiments of each WhatsApp chat on their topic of interest. The ultimate aim is to build a sentiment analysis model and identify the words whether they are positive, negative, and also the magnitude of it.


HIGH
ENERGY
LOW
enraged
furious
angry
irritated
ecstatic
motivated
lively
pleasant
apathetic
lonely
exhausted
despair
calm
grateful
peaceful
serene
NEG
PLEASANTNESS
POS

IDENTIFYING WHERE THE EMOTIONS LIE IN THE SENTIMENT ANALYSIS CAN HELP YOU FIND WAYS TO ADDRESS THEM.

DATASET

The data set used for this project willl be directly exported from the selected whatsapp chat. This data is obtained as a .txt file which can be further converted into a .csv file and be imported by R studio.

ALGORITHMS TO BE USED

Sentiment analysis algorithms

Sentiment analysis can be used to quickly analyse the text of research papers, news articles, social media posts like tweets and more. **Social Sentiment Analysis is an algorithm tuned to analyse the sentiment of social media content.**

It can be called **Opinion mining** in other words.

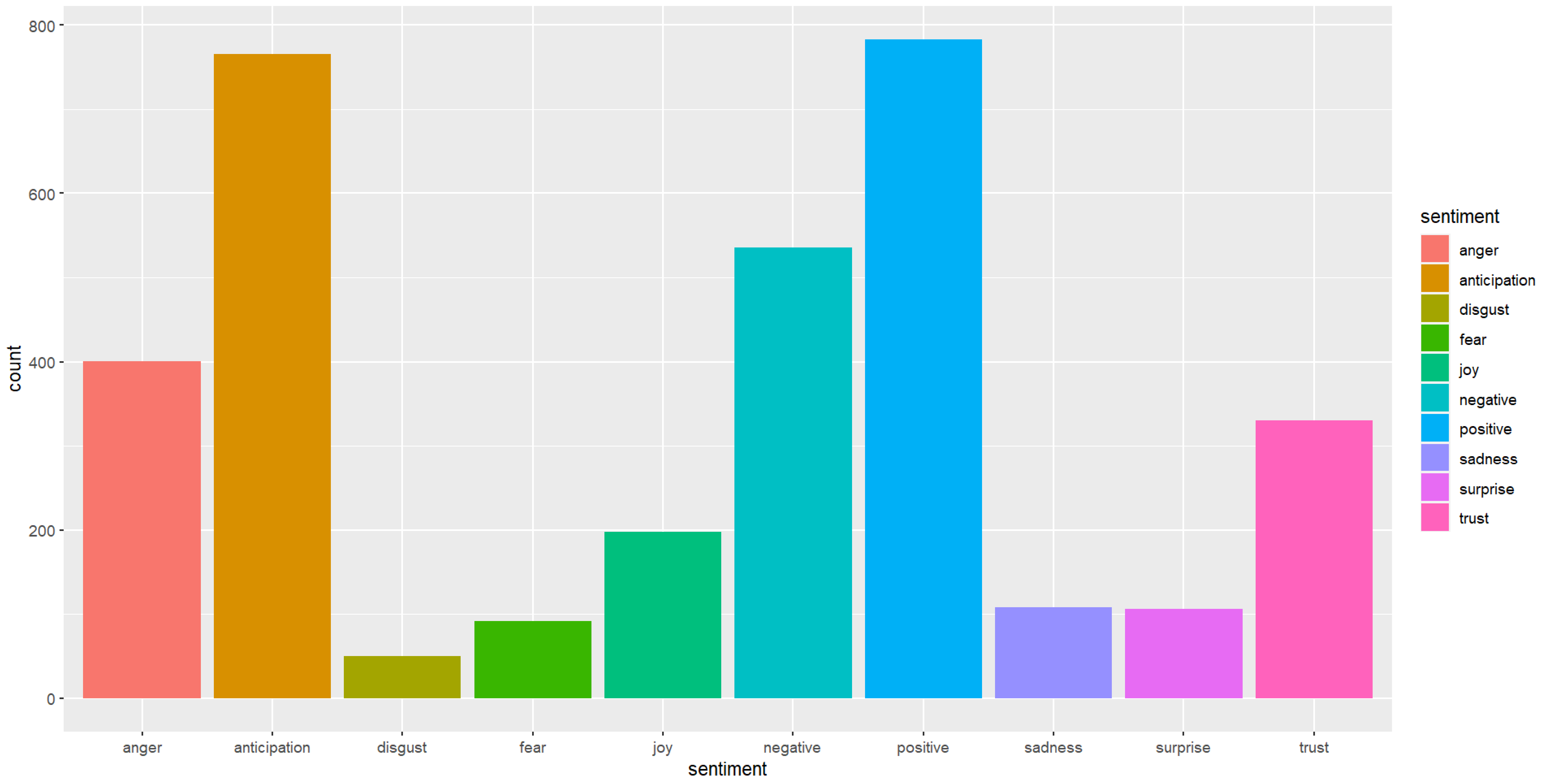
So by tidy text, we use **lexical analysis** to get the interpretations of the emotions.

We will be using **tidy text package** and **ggplots** for **visual plots in R.**

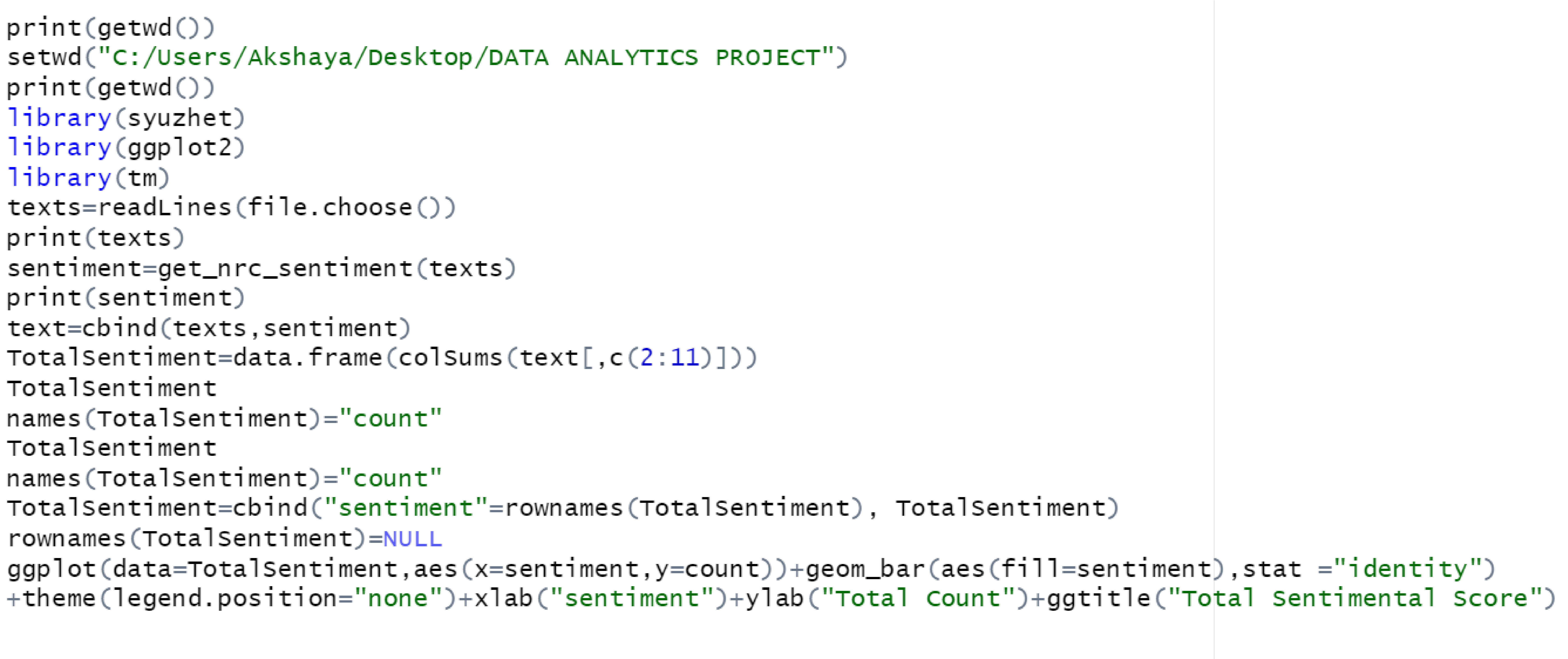
Python for the word cloud

IMPLEMENTATION

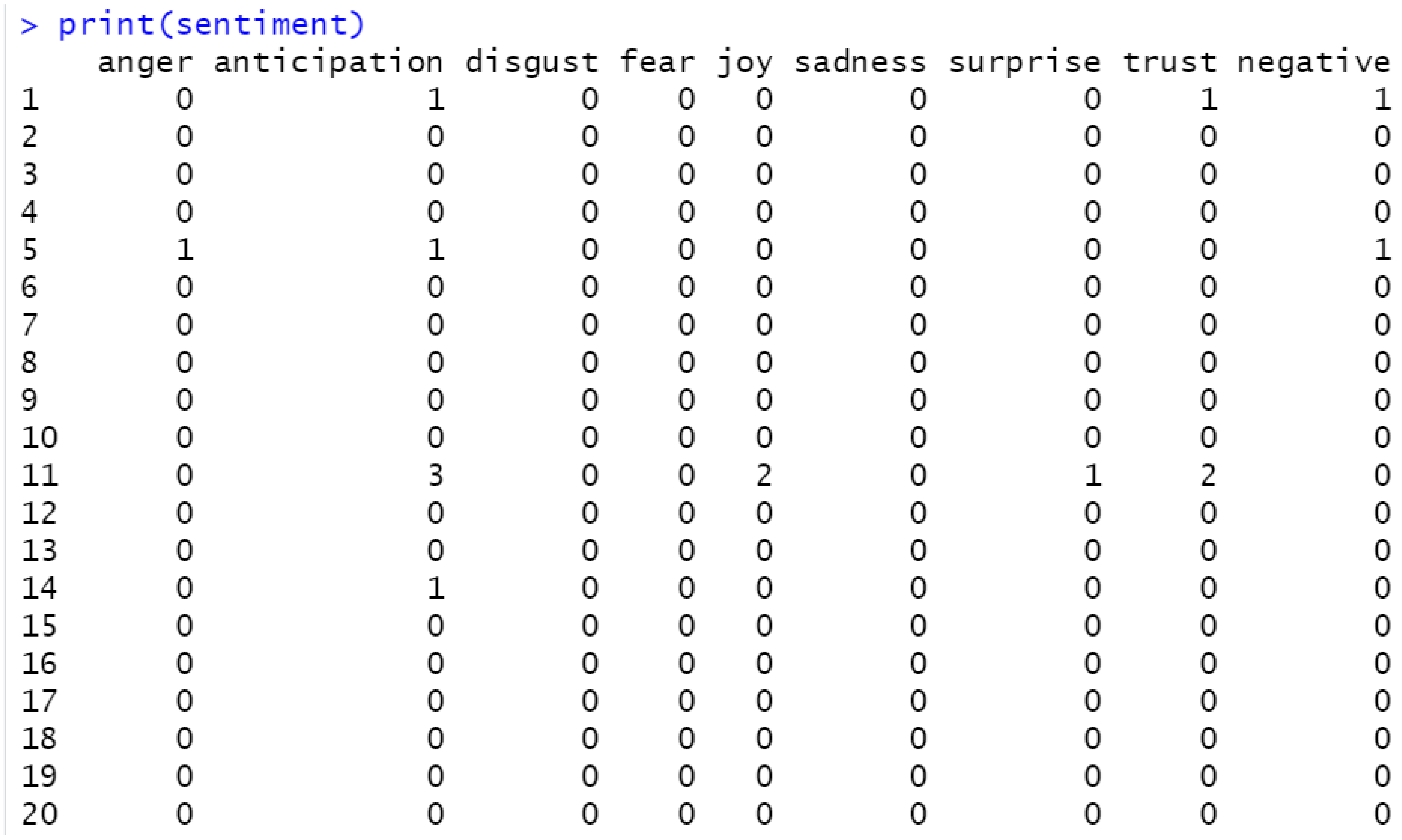
CATEGORISING EMOTIONS : PLOT



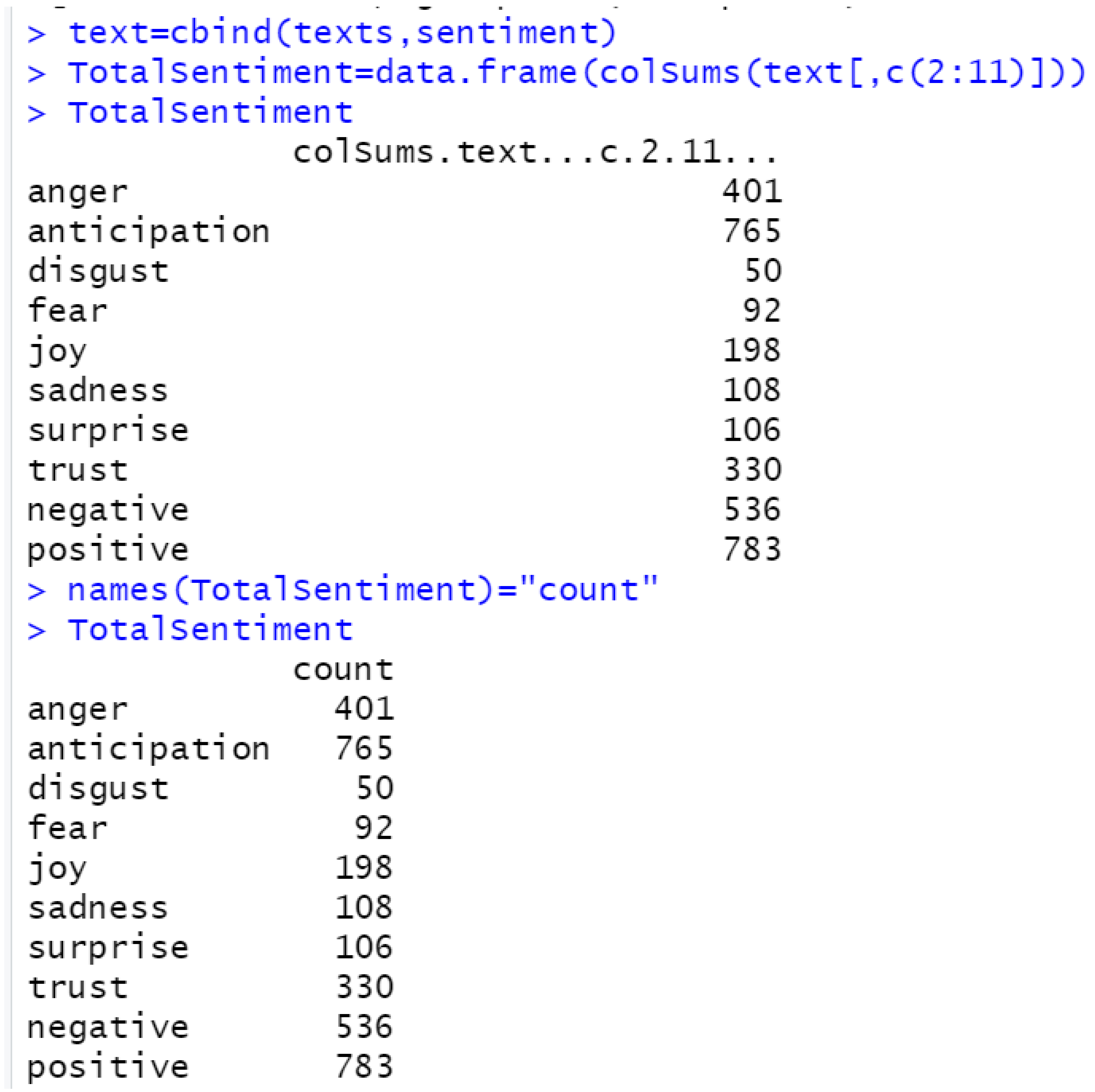
CODE :



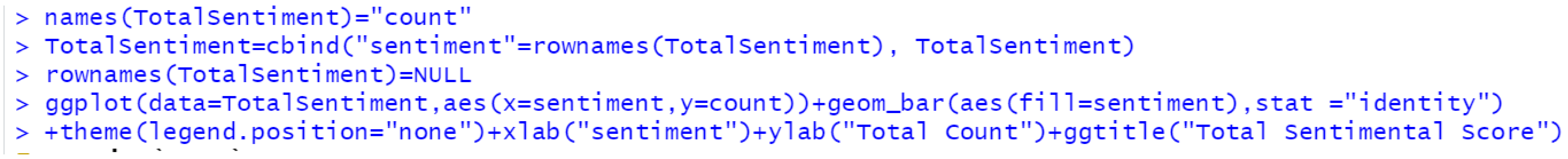
```
print(getwd())
setwd("C:/Users/Akshaya/Desktop/DATA ANALYTICS PROJECT")
print(getwd())
library(syuzhet)
library(ggplot2)
library(tm)
texts=readLines(file.choose())
print(texts)
sentiment=get_nrc_sentiment(texts)
print(sentiment)
text=cbind(texts,sentiment)
TotalSentiment=data.frame(colSums(text[,c(2:11)]))
TotalSentiment
names(TotalSentiment)="count"
TotalSentiment
names(TotalSentiment)="count"
TotalSentiment=cbind("sentiment"=rownames(TotalSentiment), TotalSentiment)
rownames(TotalSentiment)=NULL
ggplot(data=TotalSentiment,aes(x=sentiment,y=count))+geom_bar(aes(fill=sentiment),stat ="identity")
+theme(legend.position="none")+xlab("sentiment")+ylab("Total Count")+ggtitle("Total Sentimental Score")
```



```
> print(sentiment)
   anger anticipation disgust fear joy sadness surprise trust negative
1      0            1       0    0   0       0        0     1        1
2      0            0       0    0   0       0        0     0        0
3      0            0       0    0   0       0        0     0        0
4      0            0       0    0   0       0        0     0        0
5      1            1       0    0   0       0        0     0        1
6      0            0       0    0   0       0        0     0        0
7      0            0       0    0   0       0        0     0        0
8      0            0       0    0   0       0        0     0        0
9      0            0       0    0   0       0        0     0        0
10     0            0       0    0   0       0        0     0        0
11     0            3       0    0   2       0        1     2        0
12     0            0       0    0   0       0        0     0        0
13     0            0       0    0   0       0        0     0        0
14     0            1       0    0   0       0        0     0        0
15     0            0       0    0   0       0        0     0        0
16     0            0       0    0   0       0        0     0        0
17     0            0       0    0   0       0        0     0        0
18     0            0       0    0   0       0        0     0        0
19     0            0       0    0   0       0        0     0        0
20     0            0       0    0   0       0        0     0        0
```



```
> text=cbind(texts,sentiment)
> TotalSentiment=data.frame(colSums(text[,c(2:11)]))
> TotalSentiment
             colSums.text...c.2.11...
anger                             401
anticipation                      765
disgust                            50
fear                               92
joy                               198
sadness                           108
surprise                          106
trust                             330
negative                          536
positive                          783
> names(TotalSentiment)="count"
> TotalSentiment
             count
anger          401
anticipation   765
disgust         50
fear            92
joy            198
sadness        108
surprise       106
trust          330
negative       536
positive       783
```



```
> names(TotalSentiment)="count"
> TotalSentiment=cbind("sentiment"=rownames(TotalSentiment), TotalSentiment)
> rownames(TotalSentiment)=NULL
> ggplot(data=TotalSentiment,aes(x=sentiment,y=count))+geom_bar(aes(fill=sentiment),stat ="identity")
> +theme(legend.position="none")+xlab("sentiment")+ylab("Total Count")+ggtitle("Total Sentimental Score")
```

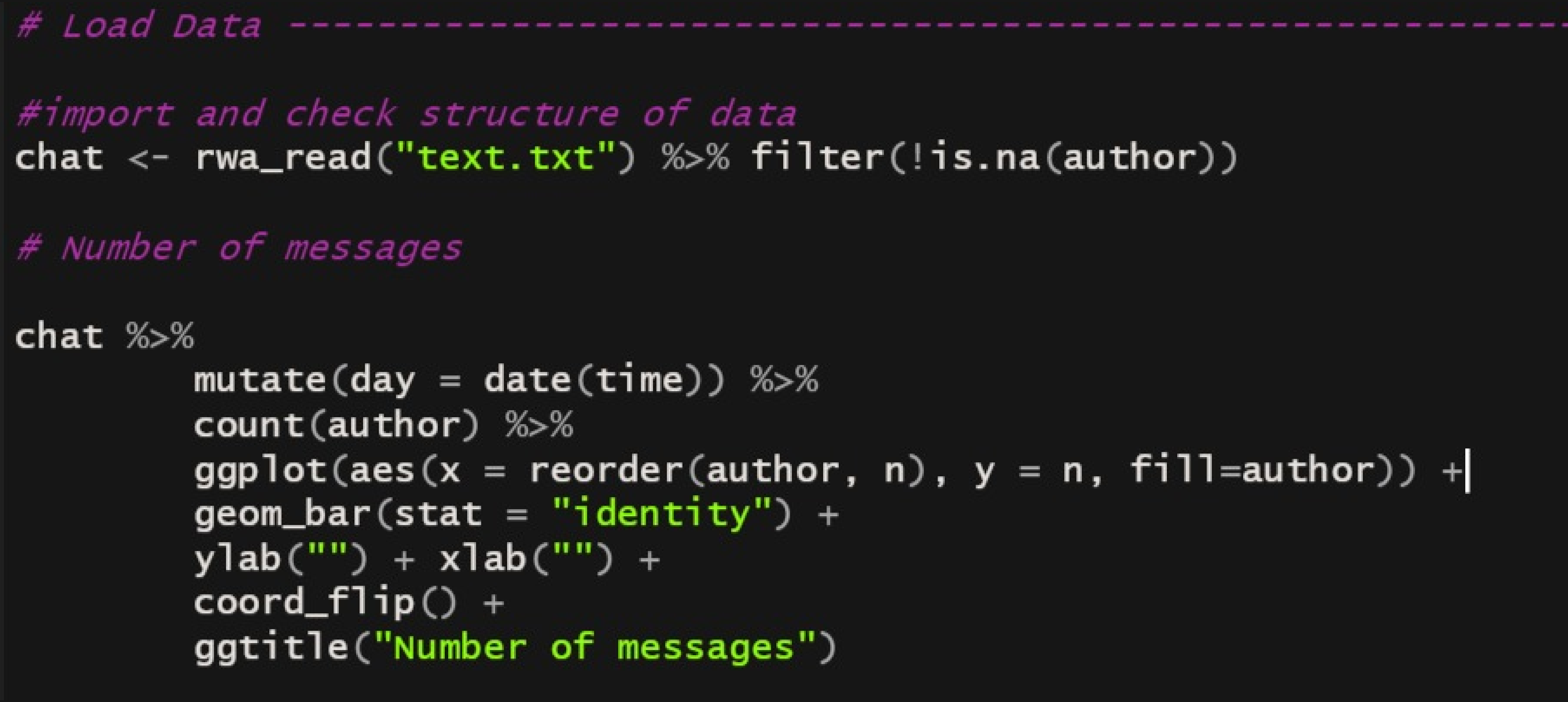
- getwd() – Get the current working directory.
- setwd('Path/To/Some/Directory') – Set current working directory
- syuzhet: Extracts Sentiment - Derived Plot Arcs from Text

Extracts sentiment and sentiment-derived plot arcs from text using a variety of sentiment dictionaries conveniently packaged for consumption by R users.

- tm: It provides a set of predefined sources, e.g., DirSource, VectorSource, or DataframeSource, which handle a directory, a vector interpreting each component as document, or data frame like structures (like CSV files), respectively.
- ggplot2: The package in R Programming Language also termed as Grammar of Graphics is a free, open-source, and easy-to-use visualization package widely used in R. It is the most powerful visualization package
- The get_nrc_sentiment function returns a data frame in which each row represents a sentence from the original file. The columns include one for each emotion type was well as the positive or negative sentiment valence.

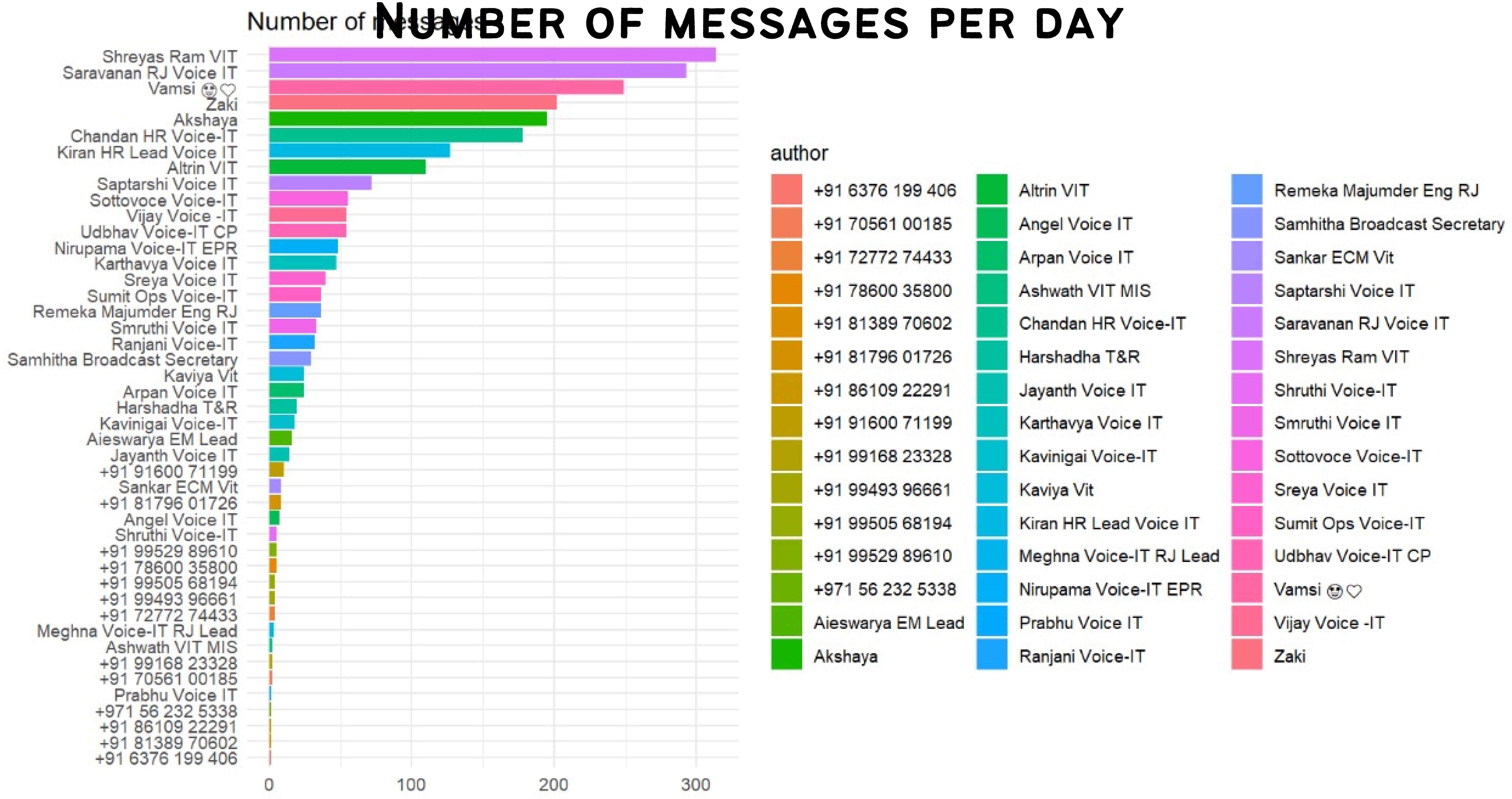
- mutate(): adds new variables and preserves existing ones; transmute() adds new variables and drops existing ones. New variables overwrite existing variables of the same name. Variables can be removed by setting their value to NULL
- count(): lets you quickly count the unique values of one or more variables: df %>% count(a, b) is roughly equivalent to df %>% group_by(a, b) %>% summarise(n = n()).
- unnest_tokens():Split a column into tokens, flattening the table into one-token-per-row. This function supports non-standard evaluation through the tidyeval framework.
- top_n() has been superseded in favour of slice_min()/slice_max(). While it will not be deprecated in the near future, retirement means that we will only perform critical bug fixes, so we recommend moving to the newer alternatives.
- The filter() function is used to produce a subset of the data frame, retaining all rows that satisfy the specified conditions. The filter() method in R programming language can be applied to both grouped and ungrouped data.

Code for number of messages per day:

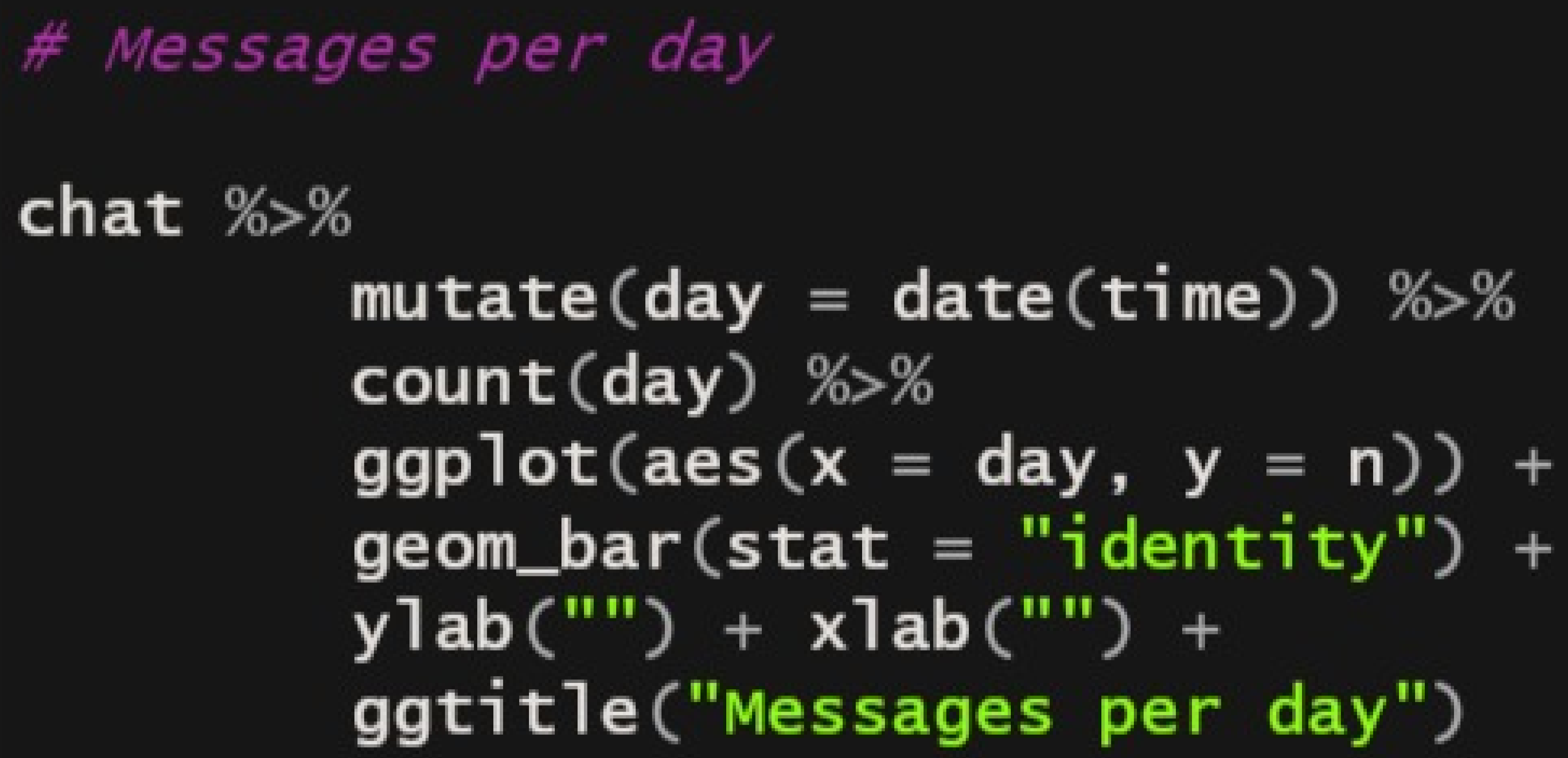


```r
# Load Data ----------------------------------------------------------

#import and check structure of data
chat <- rwa_read("text.txt") %>% filter(!is.na(author))

# Number of messages

chat %>%
        mutate(day = date(time)) %>%
        count(author) %>%
        ggplot(aes(x = reorder(author, n), y = n, fill=author)) +
        geom_bar(stat = "identity") +
        ylab("") + xlab("") +
        coord_flip() +
        ggtitle("Number of messages")
```

Number of messages per day

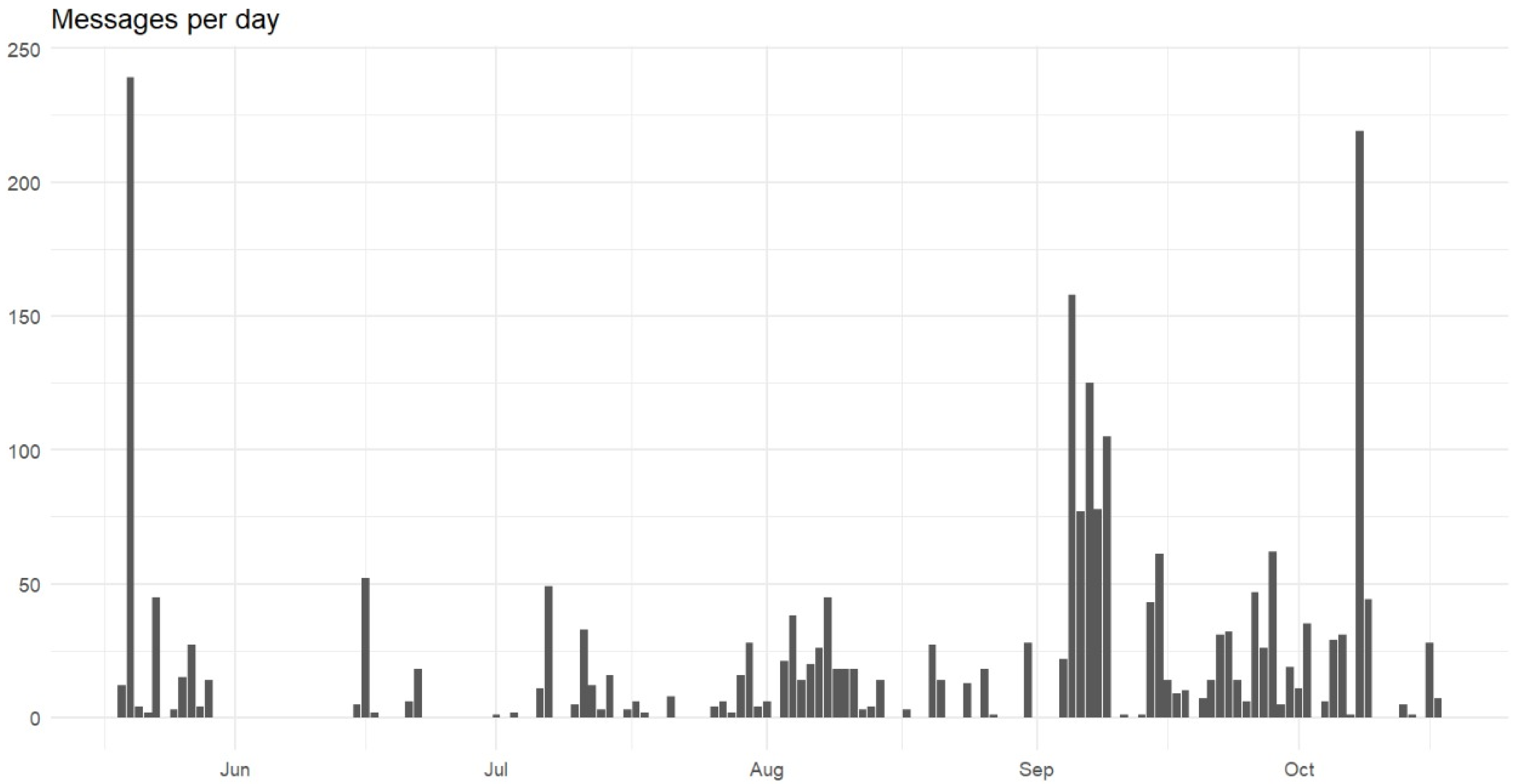


Code for messages per day:

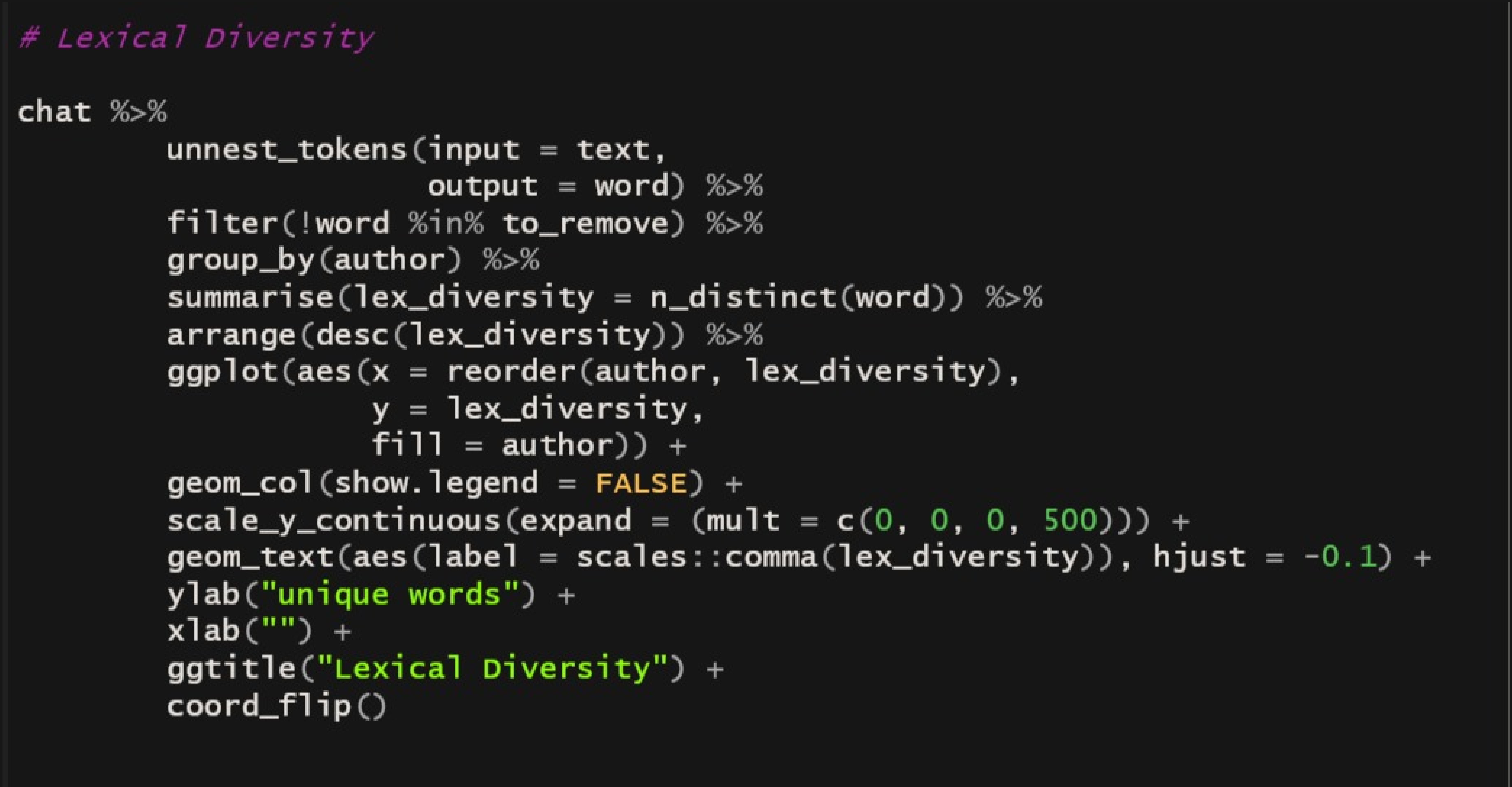


```
# Messages per day

chat %>%
        mutate(day = date(time)) %>%
        count(day) %>%
        ggplot(aes(x = day, y = n)) +
        geom_bar(stat = "identity") +
        ylab("") + xlab("") +
        ggtitle("Messages per day")
```

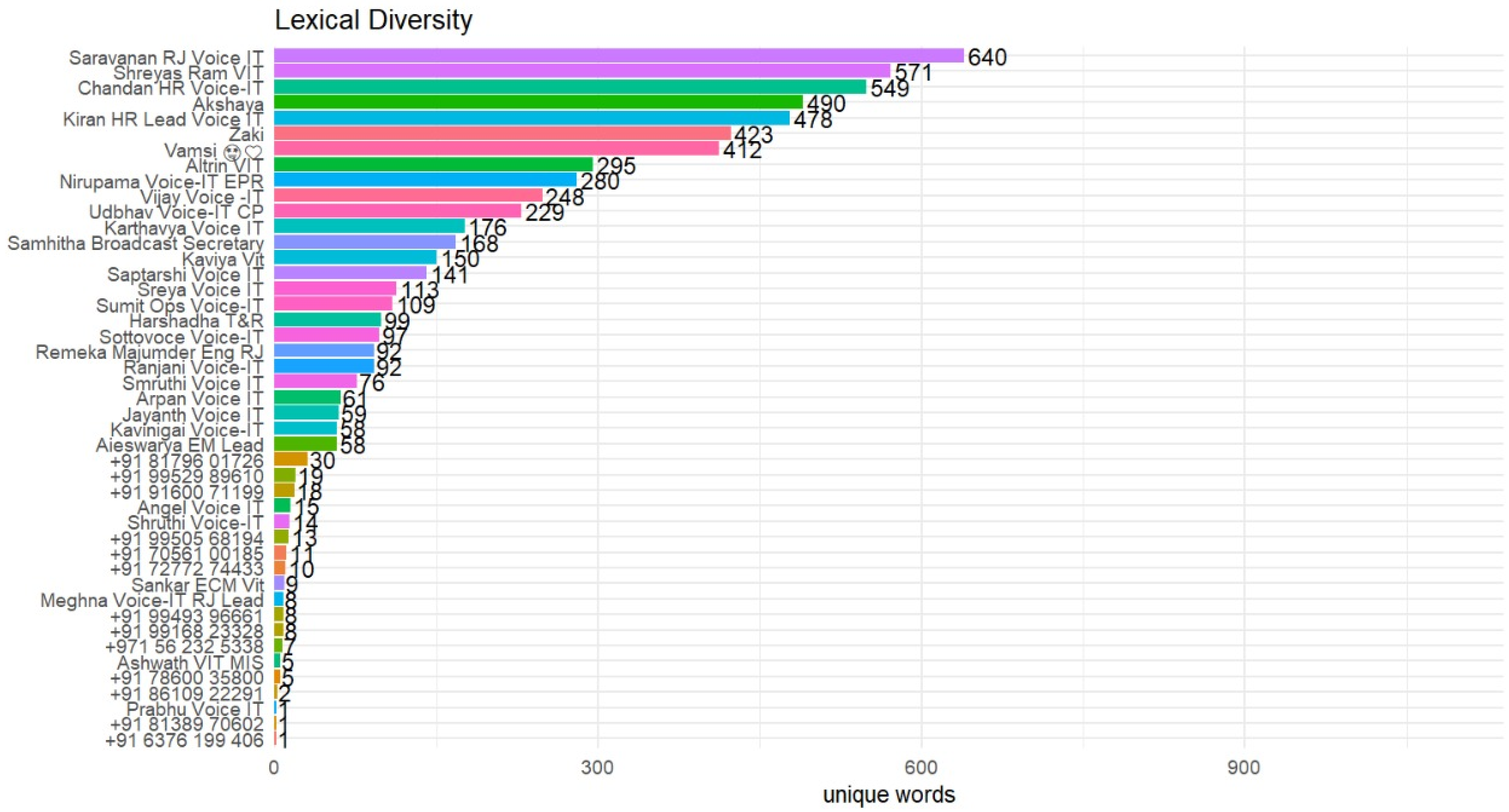
Messages per day



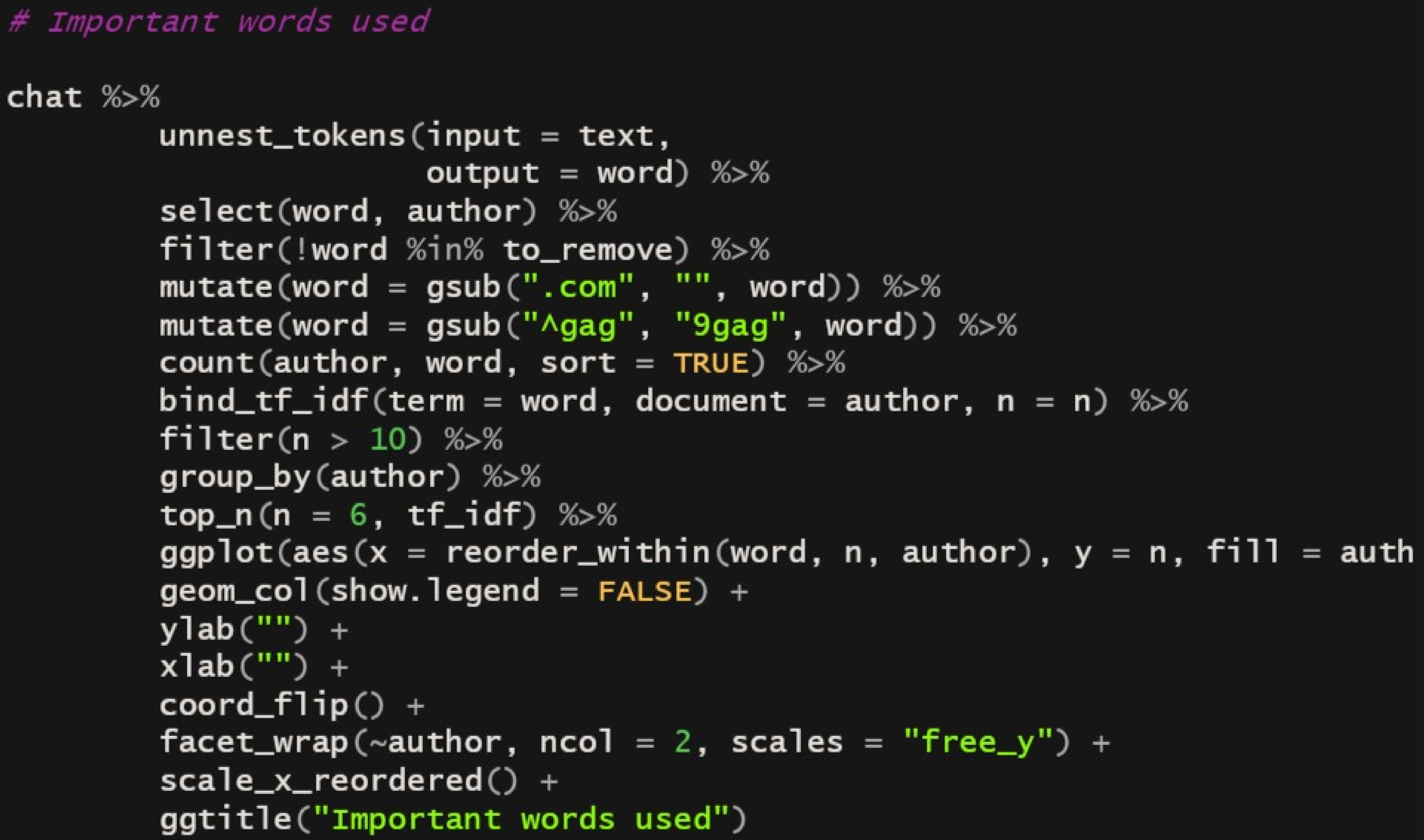
Code for lexical diversity



```r
# Lexical Diversity

chat %>%
        unnest_tokens(input = text,
                      output = word) %>%
        filter(!word %in% to_remove) %>%
        group_by(author) %>%
        summarise(lex_diversity = n_distinct(word)) %>%
        arrange(desc(lex_diversity)) %>%
        ggplot(aes(x = reorder(author, lex_diversity),
                   y = lex_diversity,
                   fill = author)) +
        geom_col(show.legend = FALSE) +
        scale_y_continuous(expand = (mult = c(0, 0, 0, 500))) +
        geom_text(aes(label = scales::comma(lex_diversity)), hjust = -0.1) +
        ylab("unique words") +
        xlab("") +
        ggtitle("Lexical Diversity") +
        coord_flip()
```

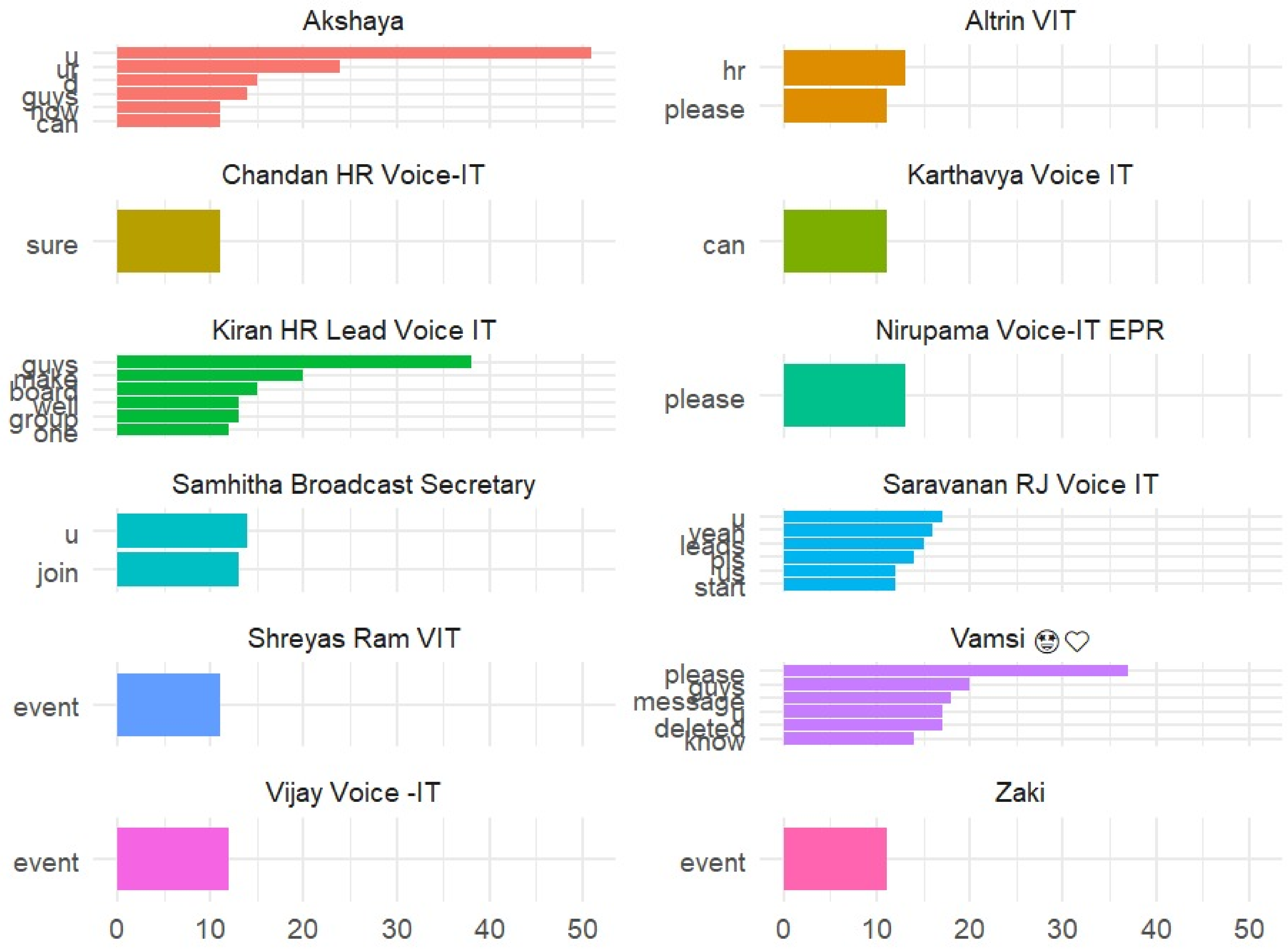

Lexical Diversity
Saravanan RJ Voice IT
Shreyas Ram VIT
Chandan HR Voice-IT
Akshaya
Kiran HR Lead Voice IT
Zaki
Vamsi
Altrin VIT
Nirupama Voice-IT EPR
Vijay Voice -IT
Udbhav Voice-IT CP
Karthavya Voice IT
Samhitha Broadcast Secretary
Kaviya Vit
Saptarshi Voice IT
Sreya Voice IT
Sumit Ops Voice-IT
Harshadha T&R
Sottovoce Voice-IT
Remeka Majumder Eng RJ
Ranjani Voice-IT
Smruthi Voice IT
Arpan Voice IT
Jayanth Voice IT
Kavinigai Voice-IT
Aieswarya EM Lead
+91 81796 01726
+91 99529 89610
+91 91600 71199
Angel Voice IT
Shruthi Voice-IT
+91 99505 68194
+91 70561 00185
+91 72772 74433
Sankar ECM Vit
Meghna Voice-IT RJ Lead
+91 99493 96661
+91 99168 23328
+971 56 232 5338
Ashwath VIT MIS
+91 78600 35800
+91 86109 22291
Prabhu Voice IT
+91 81389 70602
+91 6376 199 406
640
571
549
490
478
423
412
295
280
248
229
176
168
150
141
113
109
99
97
92
92
76
61
59
58
58
30
19
18
15
14
13
11
10
9
8
8
8
7
5
5
2
1
1
1
0
300
600
900
unique words

CODE FOR IMPORTANT WORDS :



```
# Important words used

chat %>%
        unnest_tokens(input = text,
                      output = word) %>%
        select(word, author) %>%
        filter(!word %in% to_remove) %>%
        mutate(word = gsub(".com", "", word)) %>%
        mutate(word = gsub("^gag", "9gag", word)) %>%
        count(author, word, sort = TRUE) %>%
        bind_tf_idf(term = word, document = author, n = n) %>%
        filter(n > 10) %>%
        group_by(author) %>%
        top_n(n = 6, tf_idf) %>%
        ggplot(aes(x = reorder_within(word, n, author), y = n, fill = auth
        geom_col(show.legend = FALSE) +
        ylab("") +
        xlab("") +
        coord_flip() +
        facet_wrap(~author, ncol = 2, scales = "free_y") +
        scale_x_reordered() +
        ggtitle("Important words used")
```

Important words used



List of packages and keywords used

- rwhatsapp - rwhatsapp is a small yet robust package that provides some infrastructure to work with WhatsApp text data in R.
- ggplot2 - ggplot2 package in R Programming Language also termed as Grammar of Graphics is a free, open-source, and easy-to-use visualization package widely used in R.
- lubridate - Lubridate is an R package that makes it easier to work with dates and times.
- ggimage - Supports image files and graphic objects to be visualized in 'ggplot2' graphic system.
- tidytext - tidytext is an R package that applies the principles of the tidyverse to analyzing text.
- stopwords - stopwords is an R package that provides easy access to stopwords in more than 50 languages in the Stopwords ISO library.

- tidymodels - The tidymodels framework is a collection of R packages for modeling and machine learning using tidyverse principles.
- tidyverse - The tidyverse is an opinionated collection of R packages designed for data science.

FINDINGS

This project involves the way how a set of text messages can determine the context and emotion of the discussion.
We use the tidy text package for lexical analysis, Also we would try to integrate R with Python to create a word cloud to represent the commonly used words in the chat log.

FINDINGS

We analysed the emotions such as anger,disgust, positivity, surprise etc in a bar plot.
Then saw how many messages is sent by each member in the group.
Also we found how many messages are sent the group .
We analysed the lexical diveristy of the data.
We visualised bar plots to show the important words used by members.

CONCLUSION

In conclusion, when we see the brighter side of the project, we can help people develop their character by helping them suggest proper vocabulary to guide them on the right path to reach a sustainable position with the other end person in terms of relations, business agreements and many more.
The interesting part about analysing the emotions using emojis, lexical data and other parts of a chat gives us more insight into the context of the conversation. This project can be beneficial in many areas like cybercrime to detect the crimes happening through WhatsApp chats blackmailing, analysing the fraud detection of fake forward news in groups etc. The applications of this concept has no limit.

REFERENCES

- Kaur, R. (2019). Insight to Emotional tones in WhatsApp Through Sentiment Analysis. IJRAR.
- Winarko, E., &Cherid, A. (2017, November). Recognizing the sarcastic statement on WhatsApp Group with Indonesian language text. In 2017 International Conference on Broadband Communication, Wireless Sensors and Powering (BCWSP) (pp. 1-6).IEEE.

- Waterloo, S. F., Baumgartner, S. E., Peter, J., & Valkenburg, P. M. (2018). Norms of online expressions of emotion: Comparing Facebook, Twitter, Instagram, and WhatsApp. new media & society, 20(5), 1813-1831.
- Preotiuc-Pietro, D., et al.: Trendminer: an architecture for real-time analysis of social media text. In: Proceedings of the Workshop on Real-Time Analysis and Mining of Social Streams (2012)
- Medhat, W., Hassan, A., &Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. Ain Shams engineering journal, 5(4), 1093-1113.
- Vinodhini, G., & Chandrasekaran, R. M. (2012). Sentiment analysis and opinion mining: a survey. International Journal, 2(6), 282-292.

Thank You