Performing Sentiment Analysis on Climate Change-related Twitter Tweets

Author: Muralikrishnan Rajendran

Introduction

In this section, we will conduct sentiment analysis on Twitter data focusing on tweets related to climate change. By analyzing the sentiment expressed in these tweets, we aim to gain insights into public attitudes, opinions, and emotions regarding climate change. This analysis will provide valuable information on the overall sentiment and perception surrounding this important global issue.

This code file has the following dependencies -

- 1) R version: 4.1.2 (2021-11-01) and above
- 2) Installing the required libraries (as mentioned below)
- 3) Following Source files placed in the working directory of R environment. The file is publicly available in Kaggle:
  - **twitter_sentiment_data.csv**: **Kaggle**

Step 1: Pre-Steps

Install the following required libraries (if not installed already in your local R env):

```
# install.packages('ggplot2') install.packages('dplyr')
# install.packages('lubridate') install.packages('reshape2')
# install.packages('tidyverse') install.packages('tidytext')
# install.packages('sentimentr') install.packages('stringr')
# install.packages('wordcloud') install.packages('textplot')
# install.packages('gridExtra') install.packages('patchwork')
# install.packages('tm') install.packages('stringr')

# Load the required libraries
suppressWarnings(suppressMessages(library(ggplot2)))
suppressWarnings(suppressMessages(library(dplyr)))  # for %>%
suppressWarnings(suppressMessages(library(lubridate)))
suppressWarnings(suppressMessages(library(reshape2)))
suppressWarnings(suppressMessages(library(tidyverse)))
suppressWarnings(suppressMessages(library(tidytext)))
suppressWarnings(suppressMessages(library(sentimentr)))
suppressWarnings(suppressMessages(library(stringr)))
suppressWarnings(suppressMessages(library(wordcloud)))
suppressWarnings(suppressMessages(library(textplot)))
suppressWarnings(suppressMessages(library(gridExtra)))
```

```r
suppressWarnings(suppressMessages(library(patchwork)))
suppressWarnings(suppressMessages(library(tm)))
suppressWarnings(suppressMessages(library(stringr)))
```

Step 2: Loading the Twitter sentiment dataset, Data cleansing & EDA

Loading the Twitter sentiment dataset

The provided data source is a Twitter dataset related to climate change sentiment. It is stored in a CSV file named "twitter_sentiment_data.csv." The dataset is available on Kaggle. The dataset contains tweets related to climate change, and each tweet is labeled with a sentiment score. The sentiment score represents the attitude or belief expressed in the tweet regarding man-made climate change. The possible sentiment labels are as follows:

- sentiment legend
- 2(News): the tweet links to factual news about climate change
- 1(Pro): the tweet supports the belief of man-made climate change
- 0(Neutral: the tweet neither supports nor refutes the belief of man-made climate change
- -1(Anti): the tweet does not believe in man-made climate change

Researchers or data analysts can use this dataset to perform various sentiment analysis and natural language processing tasks related to climate change sentiment on Twitter. They can investigate the prevalence of different sentiments, analyze trending topics, or study public opinions regarding climate change based on the content of the tweets in the dataset.

```r
twitter_data <- read.csv("twitter_sentiment_data.csv")

# Summary Statistics
summary(twitter_data)
```

```
##    sentiment         message             tweetid
##  Min.   :-1.0000   Length:43943       Min.   :5.926e+17
##  1st Qu.: 0.0000   Class :character   1st Qu.:7.970e+17
##  Median : 1.0000   Mode  :character   Median :8.402e+17
##  Mean   : 0.8539                      Mean   :8.368e+17
##  3rd Qu.: 1.0000                      3rd Qu.:9.020e+17
##  Max.   : 2.0000                      Max.   :9.667e+17
```

```r
# row count
nrow(twitter_data)
```

```
## [1] 43943
```

Data Cleansing and Feature Engineering

Note that - Tweets extracted from social media have unicode characters in them, which might affect our NLP processing (Sentiment analysis), hence removing unicode values twitter_data$messages is required, as shown below -

```r
# Define a function to remove Unicode characters
remove_unicode <- function(text) {
    gsub("[^ -~]", "", text)
}

# Apply the remove_unicode function to the 'message' column
twitter_data$message <- sapply(twitter_data$message, remove_unicode)

# Displaying the unicod
head(twitter_data, 10)
```

```
##    sentiment
## 1         -1
## 2          1
## 3          1
## 4          1
## 5          2
## 6          0
## 7          2
## 8          2
## 9          0
## 10         1
##
## 1    @tiniebeany climate change is an interesting hustle as it was global warming but the planet stop
## 2  RT @NatGeoChannel: Watch #BeforeTheFlood right here, as @LeoDiCaprio travels the world to tackle
## 3                         Fabulous! Leonardo #DiCaprio's film on #climate change is brilliant!!! Do wa
## 4     RT @Mick_Fanning: Just watched this amazing documentary by leonardodicaprio on climate change.
## 5         RT @cnalive: Pranita Biswasi, a Lutheran from Odisha, gives testimony on effects of climate
## 6                                                                    Unamshow awache kujinga na
## 7         RT @cnalive: Pranita Biswasi, a Lutheran from Odisha, gives testimony on effects of climate
## 8                              RT @CCIRiviera: Presidential Candidate #DonaldTrump is dangerous on cli
## 9   RT @AmericanIndian8: Leonardo DiCaprio's climate change documentary is free for a week https://t
## 10     #BeforeTheFlood Watch #BeforeTheFlood right here, as @LeoDiCaprio travels the world to tackle
##          tweetid
## 1  7.929274e+17
## 2  7.931242e+17
## 3  7.931244e+17
## 4  7.931246e+17
## 5  7.931252e+17
## 6  7.931254e+17
## 7  7.931254e+17
## 8  7.931266e+17
## 9  7.931271e+17
## 10 7.931273e+17
```

```r
# datatype of columns
str(twitter_data)
```

```
## 'data.frame':    43943 obs. of  3 variables:
##  $ sentiment: int  -1 1 1 1 2 0 2 2 0 1 ...
##  $ message  : chr  "@tiniebeany climate change is an interesting hustle as it was global warming but
##  $ tweetid  : num  7.93e+17 7.93e+17 7.93e+17 7.93e+17 7.93e+17 ...
```

```r
# Checking for null values in columns
which(is.na(twitter_data$sentiment))
```

```
## integer(0)
```

```r
which(is.na(twitter_data$message))
```

```
## integer(0)
```

```r
which(is.na(twitter_data$tweetid))
```

```
## integer(0)
```

```r
# Check for null values in each column
null_columns <- colSums(is.na(twitter_data))

# Display columns with null values
cols_with_null <- names(null_columns[null_columns > 0])
cols_with_null
```

```
## character(0)
```

As evident from the above output, there are no columns with null values in the twitter_data data frame.

Bar-plot on twitter_data$sentiment

To understand the data distribution of sentiment values, bar plots are created as shown below -

```r
# Order the levels of the sentiment column
twitter_data$sentiment <- factor(twitter_data$sentiment, levels = c(2, 1, 0, -1))

sum(twitter_data$sentiment == 2)
```

```
## [1] 9276
```

```r
sum(twitter_data$sentiment == 1)
```

```
## [1] 22962
```

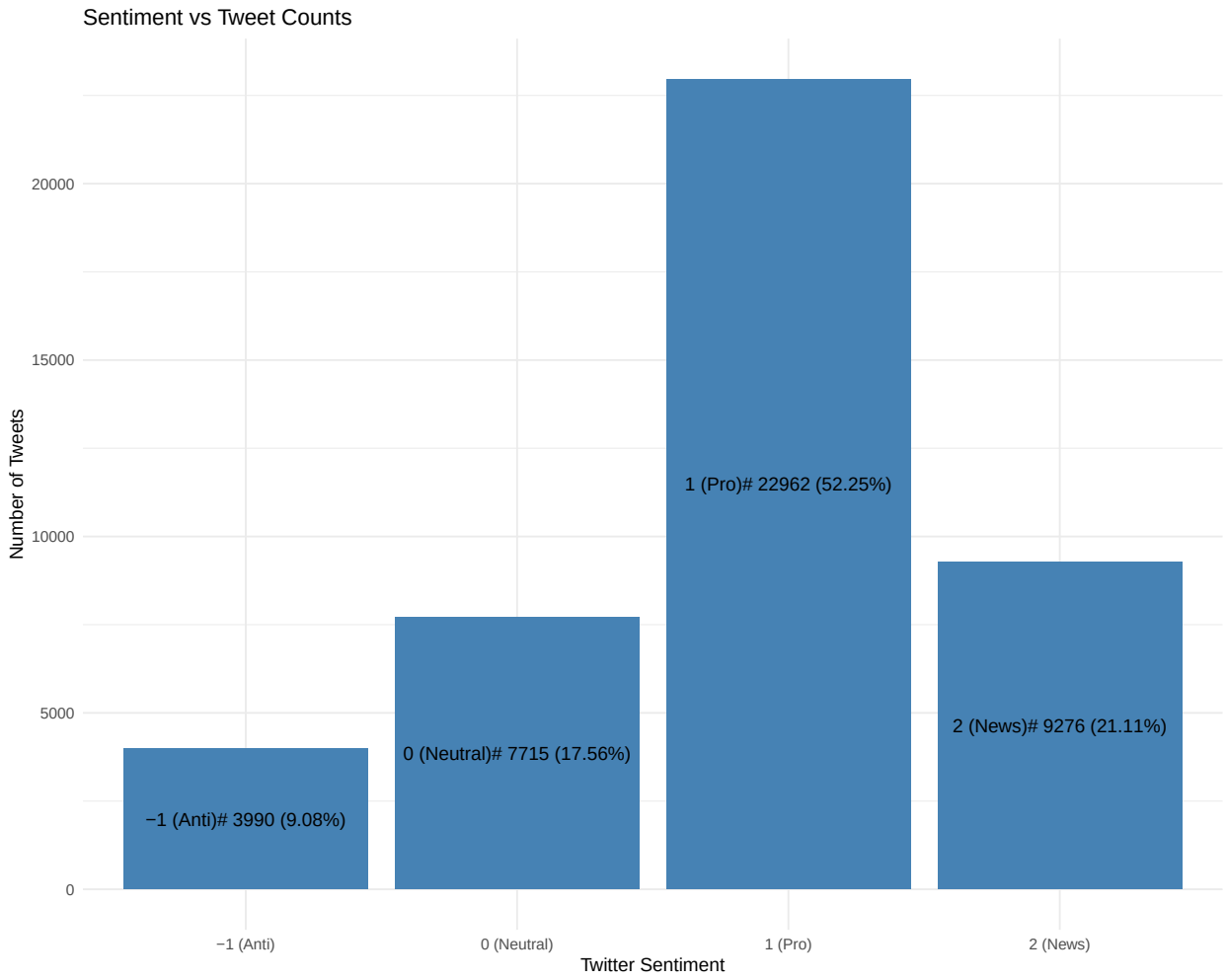
```r
sum(twitter_data$sentiment == 0)
```

```
## [1] 7715
```

```r
sum(twitter_data$sentiment == -1)
```

```
## [1] 3990
```

```r
# Calculate the counts of each sentiment level
sentiment_counts <- c(sum(twitter_data$sentiment == 2), sum(twitter_data$sentiment ==
    1), sum(twitter_data$sentiment == 0), sum(twitter_data$sentiment == -1))

# Create a data frame with sentiment labels and counts
sentiment_data <- data.frame(sentiment = c("Factual News", "Pro", "Neutral", "Anti"),
    count = sentiment_counts)

# Calculate the percentage of total tweets for each sentiment level
sentiment_data <- data.frame(sentiment = c("2 (News)", "1 (Pro)", "0 (Neutral)",
    "-1 (Anti)"), count = sentiment_counts, percentage =
    ↪  round((sentiment_counts/sum(sentiment_counts)) *
    100, 2))

# Plot the sentiment vs tweet counts with percentage labels
ggplot(sentiment_data, aes(x = sentiment, y = count)) + geom_bar(stat = "identity",
    fill = "steelblue") + geom_text(aes(label = paste0(sentiment, "# ", count, " (",
    percentage, "%)")), position = position_stack(vjust = 0.5), color = "black",
    size = 4) + labs(title = "Sentiment vs Tweet Counts", x = "Twitter Sentiment",
    y = "Number of Tweets") + theme_minimal()
```



- **Inference:**

From the bar chart, we could see the following distribution of sentiment levels among the tweets about climate change:

- 2 (News): The tweets that link to factual news about climate change. Total: **9,276 tweets (21.11%)**.
- 1 (Pro): The tweets that support the belief of man-made climate change. Total: **22,962 tweets (52.25%)**.
- 0 (Neutral): The tweets that neither support nor refute the belief of man-made climate change. Total: **7,715 tweets (17.56%)**.
- -1 (Anti): The tweets that do not believe in man-made climate change. Total: **3,990 tweets (9.08%)**.

Step 3: Statistical Analysis: Top N occurrences of tweet words (overall & per sentiment group)

- **Why determine the Top N occurrences of tweet words?**

Determining the top N occurrences of tweet words provides researchers with a valuable tool to summarize, analyze, and understand the content and trends within a large collection of tweets.

Some common use cases include:

- 1) Data summarization: By identifying the most frequent words or terms in a collection of tweets, researchers can effectively summarize the content and themes of the tweets. This helps in gaining insights into the popular topics, trends, or discussions happening on the platform.
- 2) Identifying key terms: Analyzing the top N occurrences allows researchers to identify the key terms or hashtags that are being widely used and discussed in the Twitter community. This information can be used to understand the important concepts and topics related to the research area.
- 3) Comparative analysis: By comparing the top N occurrences across different datasets or time periods, researchers can observe changes in the popularity or prevalence of certain words or topics. This comparative analysis can provide insights into evolving trends, shifts in public opinion, or emerging discussions in the Twitter sphere.

To avoid the error: **"Error: vector memory exhausted (limit reached?)"** while running **"word_counts <- rowSums(as.matrix(tdm))"**, we would need to subset the data as shown below -

```
nrow(twitter_data)
```

```
## [1] 43943
```

```
length(twitter_data$message[1:100])
```

```
## [1] 100
```

Create a vector of stop words, these stop words are determined after executing trial runs of the below -

```r
# Add more stop words as needed
filler_words <- c("the", "and", "is", "it", "in", "of", "for", "about", "that", "are",
    "you", "your", "how", "not", "have", "this", "doesn't", "will", "with", "who",
    "&amp;", "change.", "change,", "she's", "going", "from", "but", "because", "since",
    "there", "here", "to", "what", "where", "why", "or", "can", "a", "our", "more",
    "has", "via", "just", "all", "well", "its")
```

Subset I: 1:20000

Remove stop words from the text data:

```r
corpus <- Corpus(VectorSource(tolower(twitter_data$message[1:20000])))
```

In natural language processing, a corpus refers to a collection of text documents used for analysis. In the above code, the corpus is created using the Corpus function from the tm package. The Corpus function takes a source argument, which in this case is VectorSource(tolower(twitter_data$message[1:20000])). This source argument specifies the text data to be included in the corpus, which is the lowercased text from the first 20,000 messages in the twitter_data dataset.

```r
suppressWarnings(suppressMessages(corpus <- tm_map(corpus, removeWords, filler_words)))
```

By using tm_map from the tm package, the removeWords transformation is applied to each document in the corpus object, effectively removing the specified filler words from the text. The result is a modified corpus object with the filler words removed from each document.

```r
# Create a term-document matrix
tdm <- TermDocumentMatrix(corpus)
word_counts <- rowSums(as.matrix(tdm))  # will run for quite some time (~5-10 mins)
↪  depending on local machine config

# Sort the word counts in descending order
sorted_counts <- sort(word_counts, decreasing = TRUE)

# Get the top N occurrences
N <- 10
top_N <- head(sorted_counts, N)

# Print the top N occurrences
filtered_top_N_subset1 <- top_N[!(names(top_N) %in% filler_words)]
print(filtered_top_N_subset1)
```

```
## climate  global warming   trump believe  change  people
##   15545    4783    3405    2324    1653    1216     744
```

Subset II: 20000:43943

Repeating the above steps for subset - 20000:43943, we have -

```r
# Remove stop words from the text data
corpus <- Corpus(VectorSource(tolower(twitter_data$message[20000:43943])))
suppressWarnings(suppressMessages(corpus <- tm_map(corpus, removeWords, filler_words)))
```

```r
# Create a term-document matrix
tdm <- TermDocumentMatrix(corpus)
word_counts <- rowSums(as.matrix(tdm))  # will run for quite some time (~5-10 mins)
↪  depending on local machine config
```

```r
# Sort the word counts in descending order
sorted_counts <- sort(word_counts, decreasing = TRUE)

# Get the top N occurrences
N <- 10
top_N <- head(sorted_counts, N)

# Print the top N occurrences
filtered_top_N_subset2 <- top_N[!(names(top_N) %in% filler_words)]
print(filtered_top_N_subset2)
```

```
## climate  global warming  change   trump     new  people
##   17913    5751    3863    1417    1193     820     714
```

Merging the two subsets

```r
filtered_top_N_subset1 <- data.frame(filtered_top_N_subset1)  # Convert to data frame

pivoted_data1 <- filtered_top_N_subset1 %>%
    rownames_to_column(var = "word") %>%
    pivot_longer(-word, names_to = "variable", values_to = "count")

print(pivoted_data1)
```

```
## # A tibble: 7 x 3
##   word    variable               count
##   <chr>   <chr>                  <dbl>
## 1 climate filtered_top_N_subset1 15545
## 2 global  filtered_top_N_subset1  4783
## 3 warming filtered_top_N_subset1  3405
## 4 trump   filtered_top_N_subset1  2324
## 5 believe filtered_top_N_subset1  1653
## 6 change  filtered_top_N_subset1  1216
## 7 people  filtered_top_N_subset1   744
```

```r
filtered_top_N_subset2 <- data.frame(filtered_top_N_subset2)  # Convert to data frame

pivoted_data2 <- filtered_top_N_subset2 %>%
    rownames_to_column(var = "word") %>%
    pivot_longer(-word, names_to = "variable", values_to = "count")

print(pivoted_data2)
```

```
## # A tibble: 7 x 3
```

```
##   word    variable               count
##   <chr>   <chr>                  <dbl>
## 1 climate filtered_top_N_subset2 17913
## 2 global  filtered_top_N_subset2  5751
## 3 warming filtered_top_N_subset2  3863
## 4 change  filtered_top_N_subset2  1417
## 5 trump   filtered_top_N_subset2  1193
## 6 new     filtered_top_N_subset2   820
## 7 people  filtered_top_N_subset2   714
```

```
summary(pivoted_data1)
```

```
##      word             variable            count
##  Length:7           Length:7           Min.   :  744
##  Class :character   Class :character   1st Qu.: 1434
##  Mode  :character   Mode  :character   Median : 2324
##                                        Mean   : 4239
##                                        3rd Qu.: 4094
##                                        Max.   :15545
```

```
pivoted_data1 <- pivoted_data1 %>%
    select(-variable)
pivoted_data2 <- pivoted_data2 %>%
    select(-variable)
```

```
# Merge the datasets based on the 'word' column
merged_dataset <- merge(pivoted_data1, pivoted_data2, by = colnames(pivoted_data1),
    all = TRUE)

# Print the merged dataset
print(merged_dataset)
```

```
##       word count
## 1  believe  1653
## 2   change  1216
## 3   change  1417
## 4  climate 15545
## 5  climate 17913
## 6   global  4783
## 7   global  5751
## 8      new   820
## 9   people   714
## 10  people   744
## 11   trump  1193
## 12   trump  2324
## 13 warming  3405
## 14 warming  3863
```

```
## Aggregating by word, in order to remove duplicate rows
aggregated_data <- aggregate(count ~ word, merged_dataset, sum)

print(aggregated_data)
```

```
##      word count
## 1 believe  1653
## 2  change  2633
## 3 climate 33458
## 4  global 10534
## 5     new   820
## 6  people  1458
## 7   trump  3517
## 8 warming  7268
```

To Plot as a bar chart:

Calculate the percentage of total tweets for each word:

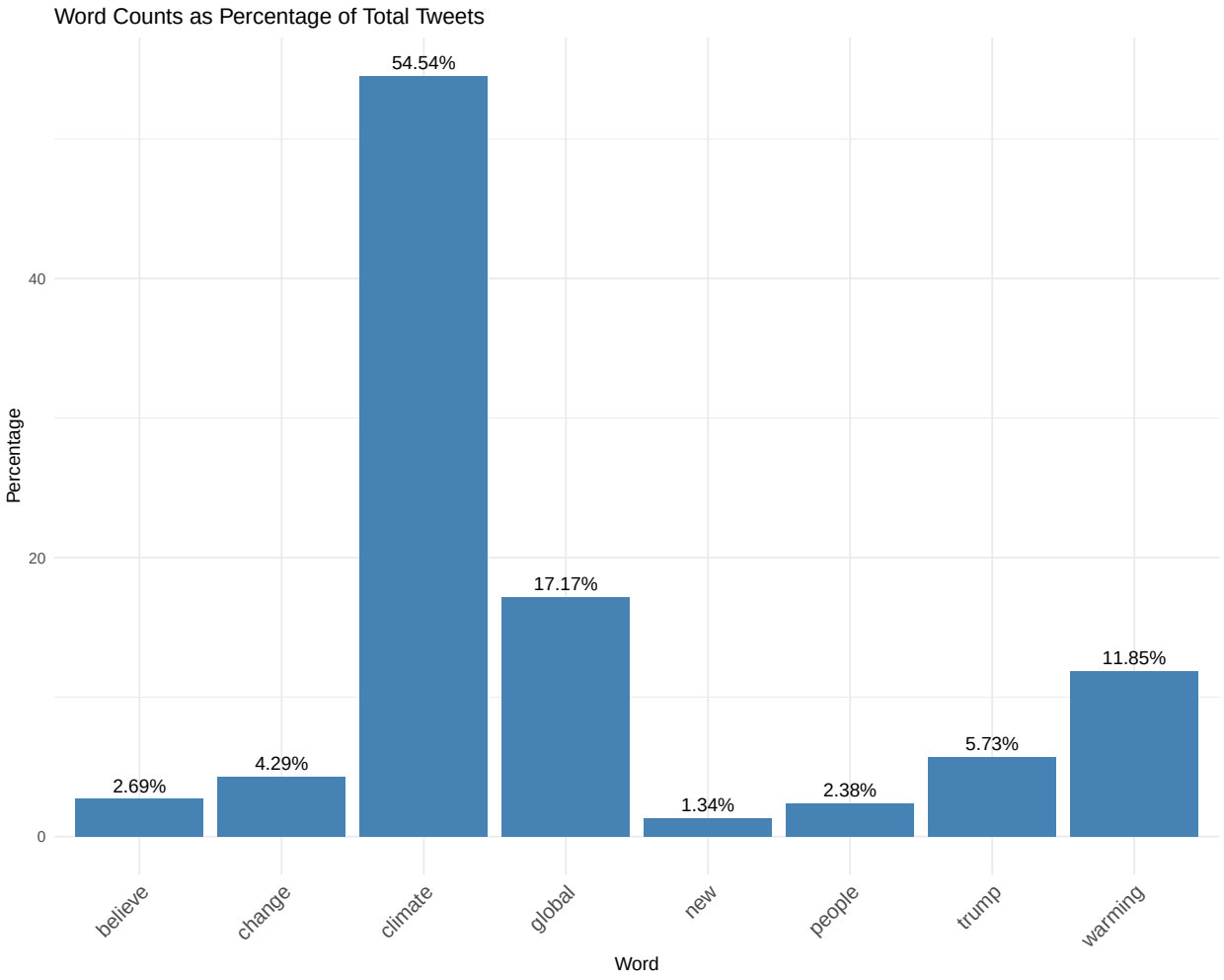
```
aggregated_data$percentage <- round(aggregated_data$count/sum(aggregated_data$count) *
    100, 2)
```

Sort the data by percentage in descending order:

```
aggregated_data <- aggregated_data[order(aggregated_data$percentage, decreasing = TRUE),
    ]
```

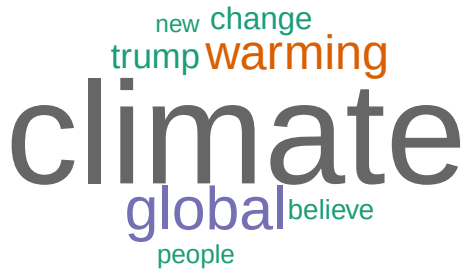
Create the bar plot:

```
plot1 <- ggplot(aggregated_data, aes(x = word, y = percentage)) + geom_bar(stat =
↪  "identity",
    fill = "steelblue") + geom_text(aes(label = paste0(percentage, "%")), vjust = -0.5,
    size = 4) + labs(title = "Word Counts as Percentage of Total Tweets", x = "Word",
    y = "Percentage") + theme_minimal() + theme(axis.text.x = element_text(angle = 45,
    hjust = 1, size = 12))

plot1
```



Generate word cloud

```
wordcloud(aggregated_data$word, aggregated_data$count, scale = c(7, 1), random.order =
↪  FALSE,
    colors = brewer.pal(8, "Dark2"))  ## from wordcloud package
```



- **Inference:**
  - From the previous outputs and plots, the following statistical inferences can be made:
    - The word **"believe"** appears **1,653** times, which accounts for **2.69%** of the total tweets.
    - The word **"change"** appears **2,633** times, which accounts for **4.29%** of the total tweets.
    - The word **"climate"** appears **33,458** times, which accounts for **54.54%** of the total tweets.
    - The word **"global"** appears **10,534** times, which accounts for **17.17%** of the total tweets.
    - The word **"new"** appears **820** times, which accounts for **1.34%** of the total tweets.
    - The word **"people"** appears **1,458** times, which accounts for **2.38%** of the total tweets.
    - The word **"trump"** appears **3,517** times, which accounts for **5.73%** of the total tweets.
    - The word **"warming"** appears **7,268** times, which accounts for **11.85%** of the total tweets.
  - Further insights from the results are as follows:
    - The word **"climate"** has the highest count, appearing in **33,458** tweets, which indicates that climate-related discussions are a significant topic of conversation in the analyzed tweets.
    - The words **"global"** and **"warming"** also have relatively high counts, with **10,534** and **7,268** occurrences, respectively. This suggests that discussions around global warming and its impact on the environment are prevalent.
    - The word **"trump"** appears **3,517** times, indicating that there is a notable mention of former President Donald Trump in the context of climate change. This might suggest the impact of Trump's policies or statements on climate-related matters.
    - The word **"change"** has a substantial count of **2,633**, reflecting the emphasis on the concept of change in relation to climate issues.

    - The words **"believe"**, **"new"**, and **"people"** have relatively lower counts compared to other terms, indicating that they are less frequently mentioned in the analyzed tweets.
  - These insights provide a glimpse into the key topics and themes discussed in relation to climate change on Twitter, highlighting the focus on climate, global warming, change, and the involvement of notable figures like Trump.

Top N words vs Sentiments

- **What is the underlying objective of classifying the top N words according to their sentiments?**
  - The purpose of categorizing the top N words based on their sentiments is to gain insights into the prevailing attitudes and opinions expressed in the text data.
  - By categorizing the words into different sentiment categories, we can understand the distribution and frequency of positive, negative, or neutral sentiments.
  - This categorization allows us to analyze sentiment patterns, identify trends, and extract meaningful information about people's opinions, perceptions, and emotions related to the topic of interest.
  - Such analysis can be valuable for various applications, including sentiment analysis, opinion mining, market research, and understanding public sentiment.

To categorize the top N words based on their sentiments, we will be looping through sentiments (2,1,0,-1), as shown below -

```
# Initialize an empty dataframe
result_df_2 <- data.frame(word = character(), sentiment = numeric(), sum = numeric(),
    stringsAsFactors = FALSE)

# Filter the data based on the current sentiment
filtered_data <- subset(twitter_data, sentiment == 2)

# Loop through each word in aggregated_data$word
for (word in aggregated_data$word) {
    # Count occurrences of the current word
    count_word <- str_count(filtered_data$message, paste0("\\b", word, "\\b"))

    # Sum the occurrences
    sum_word <- sum(count_word)

    # Add the word, sentiment, count, and sum to the result dataframe
    result_df_2 <- rbind(result_df_2, data.frame(word = word, sentiment = 2, sum =
↪   sum_word))
}


# Print the result dataframe
print(result_df_2)
```

```
##      word sentiment  sum
## 1 climate         2 7792
## 2  global         2 1284
## 3 warming         2 1162
```

```
## 4   trump         2    5
## 5  change         2 7756
## 6 believe         2   67
## 7  people         2   81
## 8     new         2  281
```

```r
# Initialize an empty dataframe
result_df_1 <- data.frame(word = character(), sentiment = numeric(), sum = numeric(),
    stringsAsFactors = FALSE)

# Filter the data based on the current sentiment
filtered_data <- subset(twitter_data, sentiment == 1)

# Loop through each word in aggregated_data$word
for (word in aggregated_data$word) {
    # Count occurrences of the current word
    count_word <- str_count(filtered_data$message, paste0("\\b", word, "\\b"))

    # Sum the occurrences
    sum_word <- sum(count_word)

    # Add the word, sentiment, count, and sum to the result dataframe
    result_df_1 <- rbind(result_df_1, data.frame(word = word, sentiment = 1, sum =
↪   sum_word))
}


# Print the result dataframe
print(result_df_1)
```

```
##      word sentiment   sum
## 1 climate         1 18767
## 2  global         1  4011
## 3 warming         1  3778
## 4   trump         1   151
## 5  change         1 18712
## 6 believe         1  1916
## 7  people         1  1043
## 8     new         1   515
```

```r
# Initialize an empty dataframe
result_df_0 <- data.frame(word = character(), sentiment = numeric(), sum = numeric(),
    stringsAsFactors = FALSE)

# Filter the data based on the current sentiment
filtered_data <- subset(twitter_data, sentiment == 0)

# Loop through each word in aggregated_data$word
for (word in aggregated_data$word) {
    # Count occurrences of the current word
    count_word <- str_count(filtered_data$message, paste0("\\b", word, "\\b"))

    # Sum the occurrences
```

```
    sum_word <- sum(count_word)

    # Add the word, sentiment, count, and sum to the result dataframe
    result_df_0 <- rbind(result_df_0, data.frame(word = word, sentiment = 0, sum =
↪   sum_word))
}


# Print the result dataframe
print(result_df_0)
```

```
##      word sentiment  sum
## 1 climate         0 4255
## 2  global         0 3032
## 3 warming         0 3004
## 4   trump         0   36
## 5  change         0 4277
## 6 believe         0  226
## 7  people         0  171
## 8     new         0  100
```

```
# Initialize an empty dataframe
result_df_minus1 <- data.frame(word = character(), sentiment = numeric(), sum =
↪   numeric(),
    stringsAsFactors = FALSE)

# Filter the data based on the current sentiment
filtered_data <- subset(twitter_data, sentiment == -1)

# Loop through each word in aggregated_data$word
for (word in aggregated_data$word) {
    # Count occurrences of the current word
    count_word <- str_count(filtered_data$message, paste0("\\b", word, "\\b"))

    # Sum the occurrences
    sum_word <- sum(count_word)

    # Add the word, sentiment, count, and sum to the result dataframe
    result_df_minus1 <- rbind(result_df_minus1, data.frame(word = word, sentiment = -1,
        sum = sum_word))
}


# Print the result dataframe
print(result_df_minus1)
```

```
##      word sentiment  sum
## 1 climate        -1 2121
## 2  global        -1 1733
## 3 warming        -1 1729
## 4   trump        -1    6
## 5  change        -1 2083
```

```
## 6 believe        -1  136
## 7  people        -1  151
## 8     new        -1   30
```

```
result_df <- dplyr::bind_rows(result_df_2, result_df_1, result_df_0, result_df_minus1)

print(result_df)
```

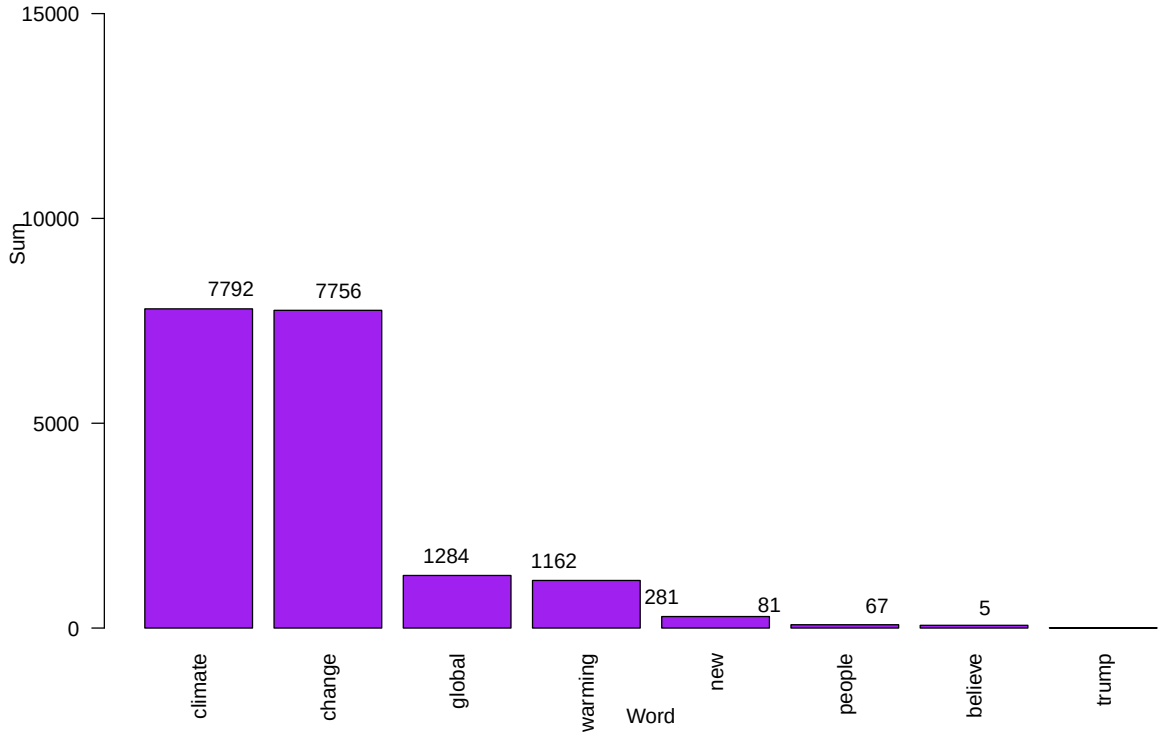
```
##       word sentiment   sum
## 1  climate         2  7792
## 2   global         2  1284
## 3  warming         2  1162
## 4    trump         2     5
## 5   change         2  7756
## 6  believe         2    67
## 7   people         2    81
## 8      new         2   281
## 9  climate         1 18767
## 10  global         1  4011
## 11 warming         1  3778
## 12   trump         1   151
## 13  change         1 18712
## 14 believe         1  1916
## 15  people         1  1043
## 16     new         1   515
## 17 climate         0  4255
## 18  global         0  3032
## 19 warming         0  3004
## 20   trump         0    36
## 21  change         0  4277
## 22 believe         0   226
## 23  people         0   171
## 24     new         0   100
## 25 climate        -1  2121
## 26  global        -1  1733
## 27 warming        -1  1729
## 28   trump        -1     6
## 29  change        -1  2083
## 30 believe        -1   136
## 31  people        -1   151
## 32     new        -1    30
```

Visualizing the above merged data, we have:

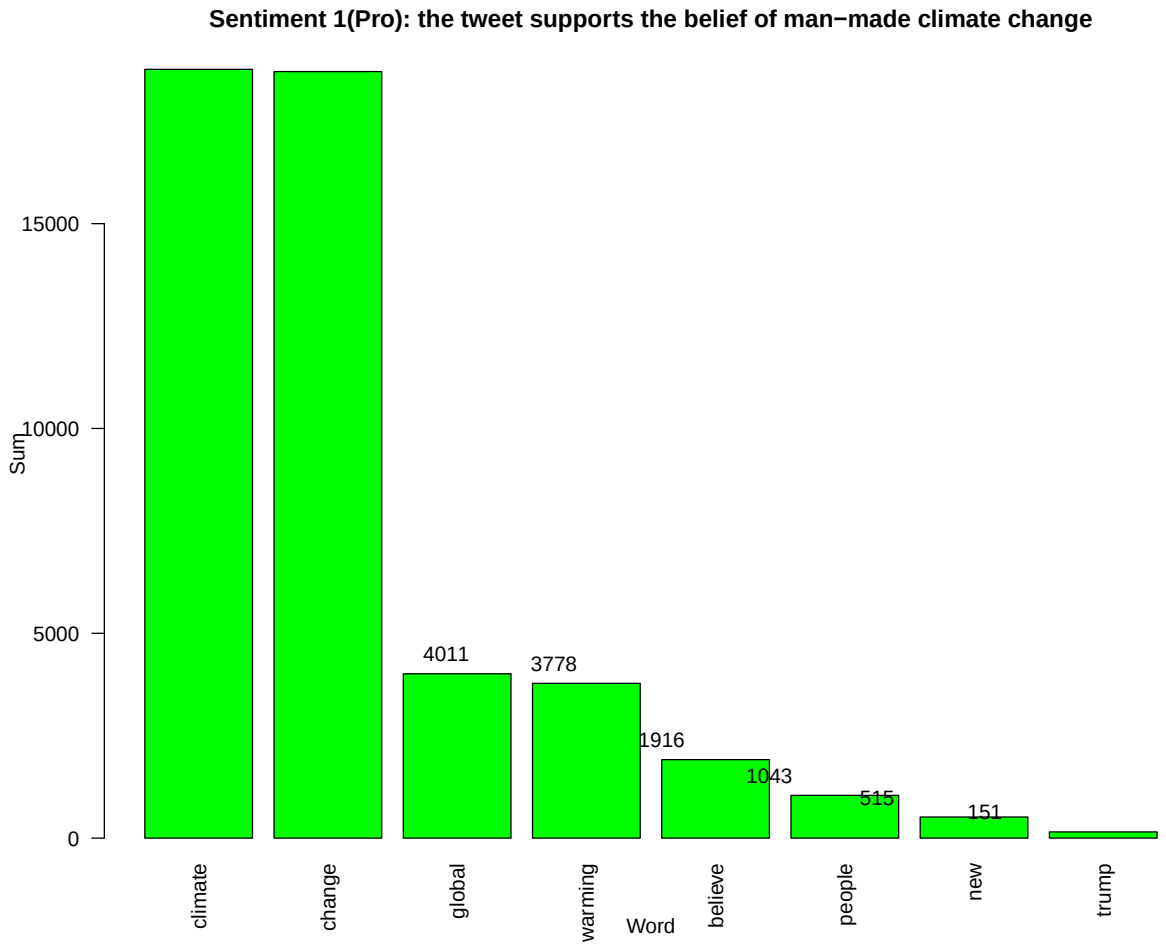
```
# Sort the data by sum in descending order
result_df_sorted <- result_df[order(result_df$sum, decreasing = TRUE), ]

# Plot the first axis
barplot(result_df_sorted$sum[result_df_sorted$sentiment == 2], names.arg =
↪  result_df_sorted$word[result_df_sorted$sentiment ==
    2], main = "Sentiment 2 (News): the tweet links to factual news about climate
    ↪  change",
    xlab = "Word", ylab = "Sum", col = "purple", ylim = c(0, max(result_df_sorted$sum)),
```

```
    las = 2)
text(x = 1:length(result_df_sorted$word[result_df_sorted$sentiment == 2]), y =
↪   result_df_sorted$sum[result_df_sorted$sentiment ==
    2], labels = result_df_sorted$sum[result_df_sorted$sentiment == 2], pos = 3)
```

Sentiment 2 (News): the tweet links to factual news about climate change

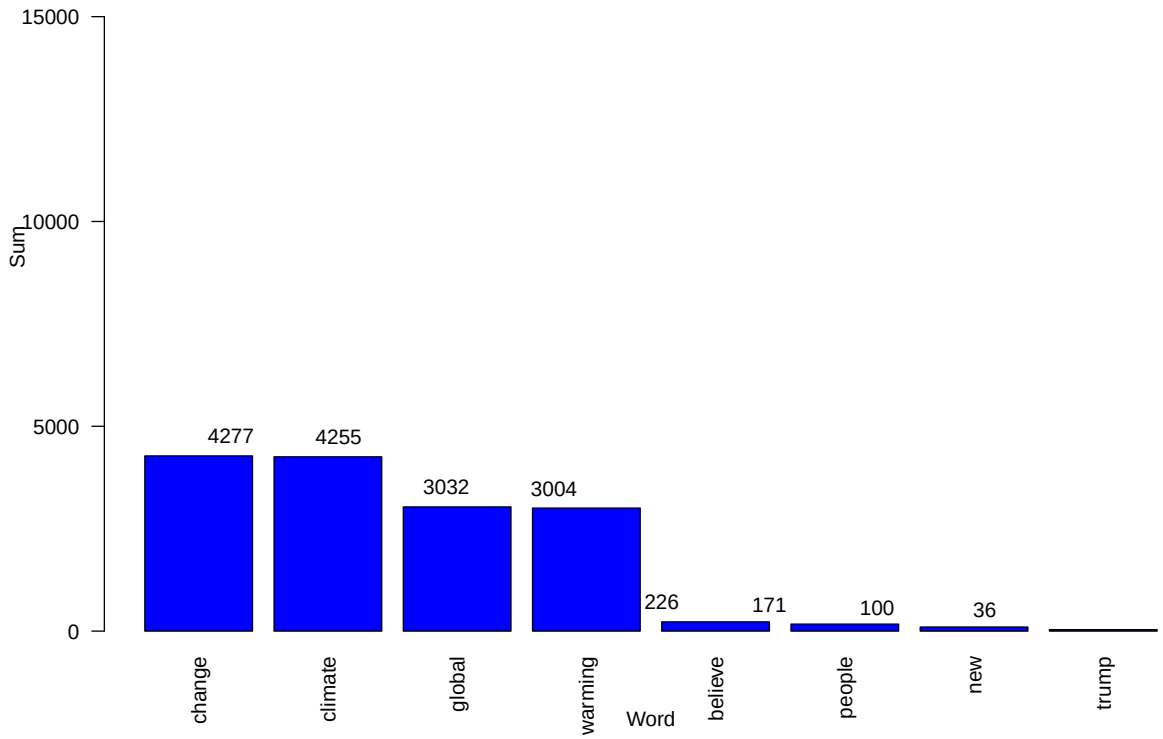



```
# Plot the second axis
barplot(result_df_sorted$sum[result_df_sorted$sentiment == 1], names.arg =
↪   result_df_sorted$word[result_df_sorted$sentiment ==
    1], main = "Sentiment 1(Pro): the tweet supports the belief of man-made climate
    ↪  change",
    xlab = "Word", ylab = "Sum", col = "green", ylim = c(0, max(result_df_sorted$sum)),
    las = 2)
text(x = 1:length(result_df_sorted$word[result_df_sorted$sentiment == 1]), y =
↪   result_df_sorted$sum[result_df_sorted$sentiment ==
    1], labels = result_df_sorted$sum[result_df_sorted$sentiment == 1], pos = 3)
```

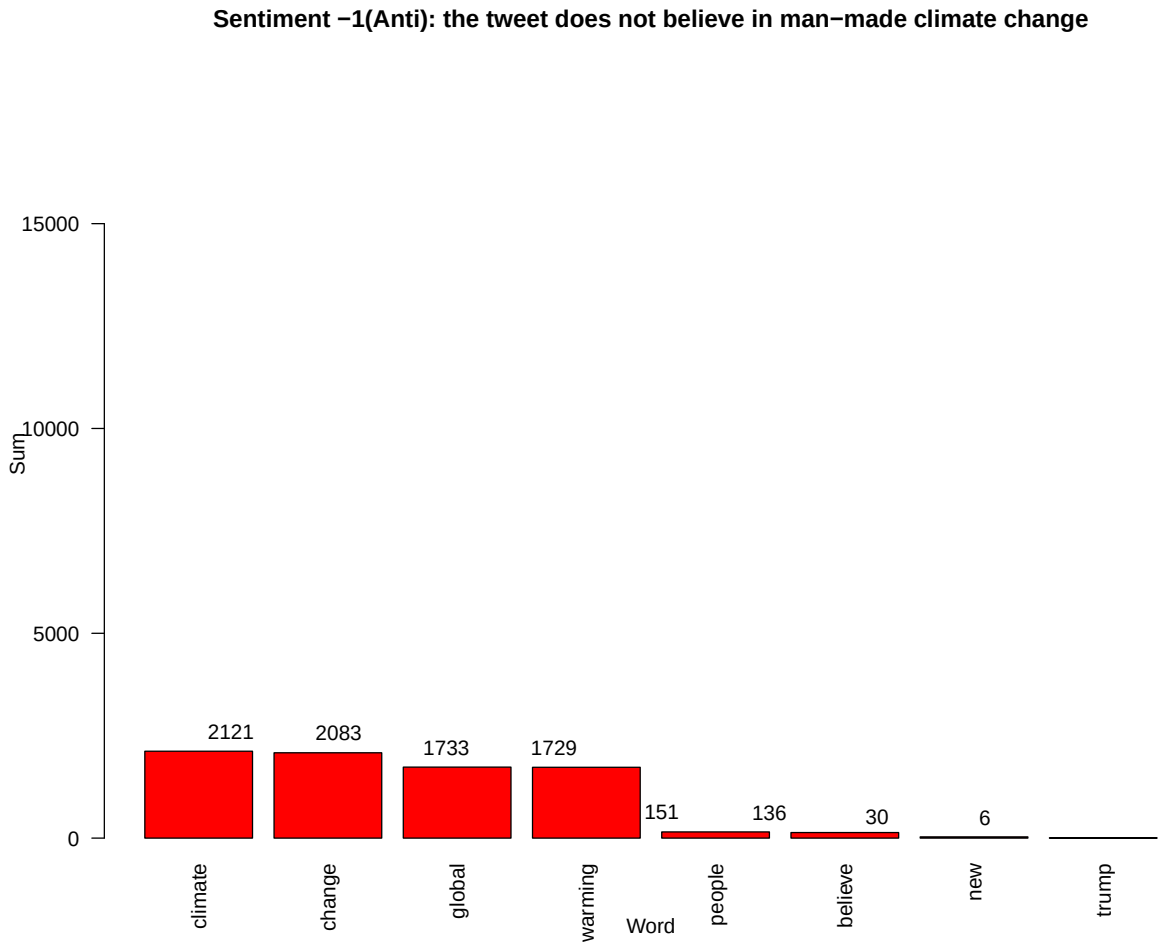



```
# Plot the third axis
barplot(result_df_sorted$sum[result_df_sorted$sentiment == 0], names.arg =
↪  result_df_sorted$word[result_df_sorted$sentiment ==
   0], main = "Sentiment 0(Neutral): the tweet neither supports nor refutes the belief
   ↪  of man-made climate change",
   xlab = "Word", ylab = "Sum", col = "blue", ylim = c(0, max(result_df_sorted$sum)),
   las = 2)
text(x = 1:length(result_df_sorted$word[result_df_sorted$sentiment == 0]), y =
↪  result_df_sorted$sum[result_df_sorted$sentiment ==
   0], labels = result_df_sorted$sum[result_df_sorted$sentiment == 0], pos = 3)
```

Sentiment 0(Neutral): the tweet neither supports nor refutes the belief of man−made climate change




```
# Plot the fourth axis
barplot(result_df_sorted$sum[result_df_sorted$sentiment == -1], names.arg =
↪  result_df_sorted$word[result_df_sorted$sentiment ==
   -1], main = "Sentiment -1(Anti): the tweet does not believe in man-made climate
   ↪  change",
   xlab = "Word", ylab = "Sum", col = "red", ylim = c(0, max(result_df_sorted$sum)),
   las = 2)
text(x = 1:length(result_df_sorted$word[result_df_sorted$sentiment == -1]), y =
↪  result_df_sorted$sum[result_df_sorted$sentiment ==
   -1], labels = result_df_sorted$sum[result_df_sorted$sentiment == -1], pos = 3)
```



- **Inference:**

The top 3 words from each sentiment category:

- 1) News Sentiment: [2 - News: the tweet links to factual news about climate change]
  - "climate" - **7,792** tweets
  - "change" - **7,756** tweets
  - "global" - **1,284** tweets
  - The presence of words like "climate," "change," and "global" indicates that news tweets about climate change often focus on these topics. The relatively high count of tweets containing these words suggests a significant amount of factual news content related to climate change.
- 2) Positive Sentiment: [1 - Pro: the tweet supports the belief of man-made climate change]
  - "climate" - **18,767** tweets
  - "change" - **18,712** tweets
  - "global" - **4,011** tweets
  - The repeated occurrence of words like "climate," "change," and "global" in positive sentiment tweets signifies strong support for the belief in man-made climate change. The higher tweet counts for these words indicate a positive attitude towards addressing climate change and promoting awareness.

- 3) Neutral Sentiment: [0 - Neutral: the tweet neither supports nor refutes the belief of man-made climate change]
  - "change" - **4,277** tweets
  - "climate" - **4,255** tweets
  - "global" - **3,032** tweets
  - The occurrence of words like "climate," "change," and "global" in neutral sentiment tweets suggests that these topics are commonly discussed without taking a specific stance. The similar tweet counts for these words indicate a balanced representation of neutral opinions on climate change.
- 4) Negative Sentiment: [-1 - Anti: the tweet does not believe in man-made climate change]
  - "climate" - **2,121** tweets
  - "change" - **2,083** tweets
  - "global" - **1,733** tweets
  - The presence of words like "climate," "change," and "global" in negative sentiment tweets suggests skepticism or disbelief in man-made climate change. The lower tweet counts for these words indicate a relatively smaller proportion of tweets expressing negative sentiments towards climate change.

Step 4: Sentiment Analysis: How well does the sentiment scores reflect user sentiment?

- **What is the purpose of Sentiment analysis?**
  Sentiment analysis is performed to understand and quantify the sentiment or emotional tone expressed in a piece of text, such as tweets, reviews, or comments. It involves analyzing the text to determine whether it expresses a positive, negative, or neutral sentiment. The goal is to extract insights about people's opinions, attitudes, or emotions towards a particular topic or entity.
- **What are Sentiment scores?**
  Sentiment scores, also known as polarity scores, are numerical values assigned to each piece of text to represent the sentiment expressed. These scores indicate the degree of positivity or negativity in the text. Typically, sentiment scores range from -1 to 1, where -1 represents strong negative sentiment, 1 represents strong positive sentiment, and 0 represents neutral sentiment.

For example, let's consider a tweet: "I absolutely loved the new movie! The acting was brilliant, and the storyline kept me engaged throughout." In this case, a sentiment analysis algorithm would assign a positive sentiment score close to 1, indicating that the tweet expresses a highly positive sentiment. The algorithm would analyze the words "loved," "brilliant," and "engaged" to determine the positive sentiment.

The sentiment scores are created using the sentiment function from the sentimentr package, as shown below -

```
# Create a function to calculate sentiment scores
calculate_sentiment_score <- function(text) {
    sentiment <- sentiment(text, polarity_dt = lexicon::hash_sentiment_jockers_rinker)
    mean_sentiment <- mean(sentiment$sentiment)
    return(mean_sentiment)
}

# Apply the sentiment score calculation to the 'message' column in the data
# frame
twitter_data$sentiment_score <- sapply(twitter_data$message, calculate_sentiment_score)
↪  # will run for quite some time (~15-25 mins) depending on local machine config
```

```r
# View the updated data frame
head(twitter_data, 10)
```

```
##    sentiment
## 1         -1
## 2          1
## 3          1
## 4          1
## 5          2
## 6          0
## 7          2
## 8          2
## 9          0
## 10         1
##
## 1    @tiniebeany climate change is an interesting hustle as it was global warming but the planet sto
## 2  RT @NatGeoChannel: Watch #BeforeTheFlood right here, as @LeoDiCaprio travels the world to tackle
## 3                        Fabulous! Leonardo #DiCaprio's film on #climate change is brilliant!!! Do w
## 4     RT @Mick_Fanning: Just watched this amazing documentary by leonardodicaprio on climate change.
## 5         RT @cnalive: Pranita Biswasi, a Lutheran from Odisha, gives testimony on effects of climat
## 6                                                                      Unamshow awache kujinga na
## 7         RT @cnalive: Pranita Biswasi, a Lutheran from Odisha, gives testimony on effects of climat
## 8                             RT @CCIRiviera: Presidential Candidate #DonaldTrump is dangerous on cl
## 9   RT @AmericanIndian8: Leonardo DiCaprio's climate change documentary is free for a week https://t
## 10     #BeforeTheFlood Watch #BeforeTheFlood right here, as @LeoDiCaprio travels the world to tackle
##          tweetid sentiment_score
## 1  7.929274e+17      0.05221201
## 2  7.931242e+17      0.12001984
## 3  7.931244e+17      0.21338835
## 4  7.931246e+17      0.06933752
## 5  7.931252e+17     -0.08728716
## 6  7.931254e+17      0.11094004
## 7  7.931254e+17     -0.08728716
## 8  7.931266e+17      0.09036961
## 9  7.931271e+17      0.10910895
## 10 7.931273e+17      0.07349684
```

```r
nrow(twitter_data)
```

```
## [1] 43943
```

```r
head(twitter_data[order(twitter_data$sentiment_score, decreasing = TRUE), ], 10)
```

```
##       sentiment
## 36511         1
## 34875         0
## 3666          1
## 12516         0
## 13938         1
## 16819         1
## 13606         1
```

```
## 39867         1
## 24734         1
## 5021          1
##
## 36511        RT @BirdsGetStarted: #DemsDo*Care about equality*Care about voting rights*Care about cli
## 34875  RT @mannyrull: does anyone think global warming is a good thing I love Lady Gaga I think she's
## 3666               @realDonaldTrump one thing I would like to ask is please be mindful of our ecosyst
## 12516 @PatriciaRobson9 @cenkuygur Their messages of being pro climate change, pro gay rights, pro imm
## 13938                                                                                  RT @Darthcoal
## 16819      RT @MikeCarlton01: Most Australians:Are happy with 18cAre happy with SSMBelieve in climate
## 13606                                                    I know global warming is bad but I'm not ha
## 39867     more hopeful, less realistic #scibucketlist:develop adaptive solutions to protect midwest c
## 24734   RT @heyyPJ: We care more about emails and obstruction of justice instead of creating proper l
## 5021      RT @martasubira: Catalonia is happy to share goals &amp; commitment on climate change with
##             tweetid sentiment_score
## 36511 9.628423e+17       1.2969194
## 34875 9.566349e+17       1.1472794
## 3666  7.969437e+17       1.1468293
## 12516 8.268629e+17       1.1180340
## 13938 8.376745e+17       1.0333333
## 16819 8.441691e+17       1.0062306
## 13606 8.348585e+17       0.9911825
## 39867 6.750407e+17       0.9701425
## 24734 8.902357e+17       0.9601587
## 5021  7.981221e+17       0.9503289
```

```
head(twitter_data[order(twitter_data$sentiment_score, decreasing = FALSE), ], 10)
```

```
##       sentiment
## 28268         0
## 3257          1
## 43193         1
## 23572         1
## 8407          1
## 16719         1
## 32974        -1
## 12730         0
## 34717         0
## 12401        -1
##
## 28268      vatican enforced climate change=mad mafia pope wants more anti-abortion/anti-contraceptio
## 3257                                        dumbass racist, sexist, misogynist, lying and unqualifie
## 43193       RT @davidsirota: Lots of partisan outrage about media coverage of candidates  much less c
## 23572                            1.5 billion for climate change denying creationist anti-abortion, a
## 8407                                          RT @MeganNeuringer: if u think pizzagate is real but
## 16719 @realDonaldTrump fascist, misogynist holocaust &amp; climate change denier liar #notmypresident
## 32974        RT @TheMarkPantano: Leftists suffering depression and anxiety from worrying about climat
## 12730                RT @BAKANEK1: we dont get a lot of new tatsunari pictures but when we do angels
## 34717                                                 People continually bash global warming, but i
## 12401      RT @bengoldacre: Green politicians' weird anti science thing really is tiresome and unhelp
##             tweetid sentiment_score
## 28268 9.249504e+17       -1.826484
## 3257  7.966831e+17       -1.500000
```

```
## 43193 7.767915e+17       -1.472971
## 23572 8.793804e+17       -1.368454
## 8407  8.064137e+17       -1.324876
## 16719 8.439576e+17       -1.187715
## 32974 9.534711e+17       -1.187500
## 12730 8.285547e+17       -1.173714
## 34717 9.562470e+17       -1.167203
## 12401 8.260153e+17       -1.106854
```

- **Inference:**
  - The output displays the top 10 rows of the twitter_data dataframe, sorted by the sentiment_score column in both ascending and descending order. Each row contains the sentiment, message, tweetid, and sentiment_score values.
  - These results provide insights into the sentiment of the tweets in the dataset, with higher sentiment scores indicating more positive sentiment and lower scores indicating more negative sentiment. It allows for further analysis and understanding of the sentiment distribution in the dataset.

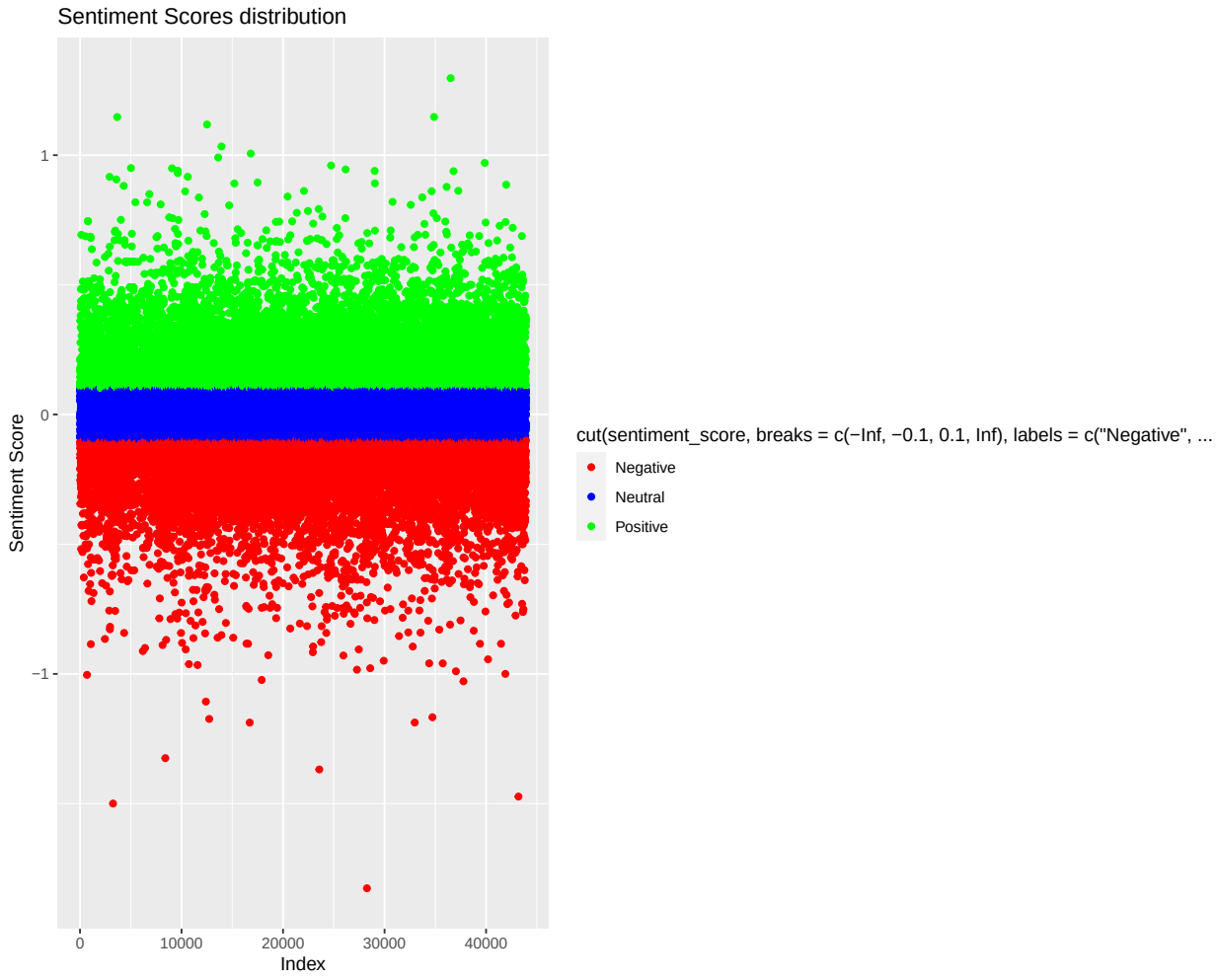
Analyzing the relationship between the sentiment_score and message columns in the twitter_data dataframe reveals additional insights:

- Tweets with Positive Sentiment scores: In general, Tweets with higher sentiment scores often contain positive language, expressing support, enthusiasm, or agreement with regards to climate change. These tweets may contain words and phrases indicating optimism, proactive actions, and belief in the urgency of addressing climate change. However, it's important to note that there are tweet messages with a positive sentiment score that express disbelief or skepticism towards the notion of climate change. This can be attributed to the nature of sentiment analysis using natural language processing (NLP), where the sentiment score is determined based on the overall sentiment conveyed by the words and phrases used in the message. In such cases, although the sentiment score may indicate positivity, the actual sentiment expressed in the message may differ, highlighting the complexities and nuances of sentiment analysis in capturing the true sentiment behind a text.
- Tweets with Negative Sentiment scores: Tweets with lower sentiment scores tend to have negative language, reflecting skepticism, denial, or criticism of climate change. These tweets may contain words and phrases indicating disbelief, questioning the validity of climate change, or expressing negative emotions towards related topics. Also, as mentioned before, there are tweet messages with a negative sentiment score that may convey support or agreement with the concept of climate change. This is because sentiment analysis using natural language processing (NLP) assigns sentiment scores based on the language and context used in the text. In some cases, negative sentiment scores may be given to tweets that discuss the negative impacts or consequences of climate change, even though the overall sentiment expressed in the message aligns with the belief in climate change. This highlights the challenges of sentiment analysis and the importance of considering the context and nuances of the text when interpreting sentiment scores.

Scatter plot for showing the sentiment scores distribution

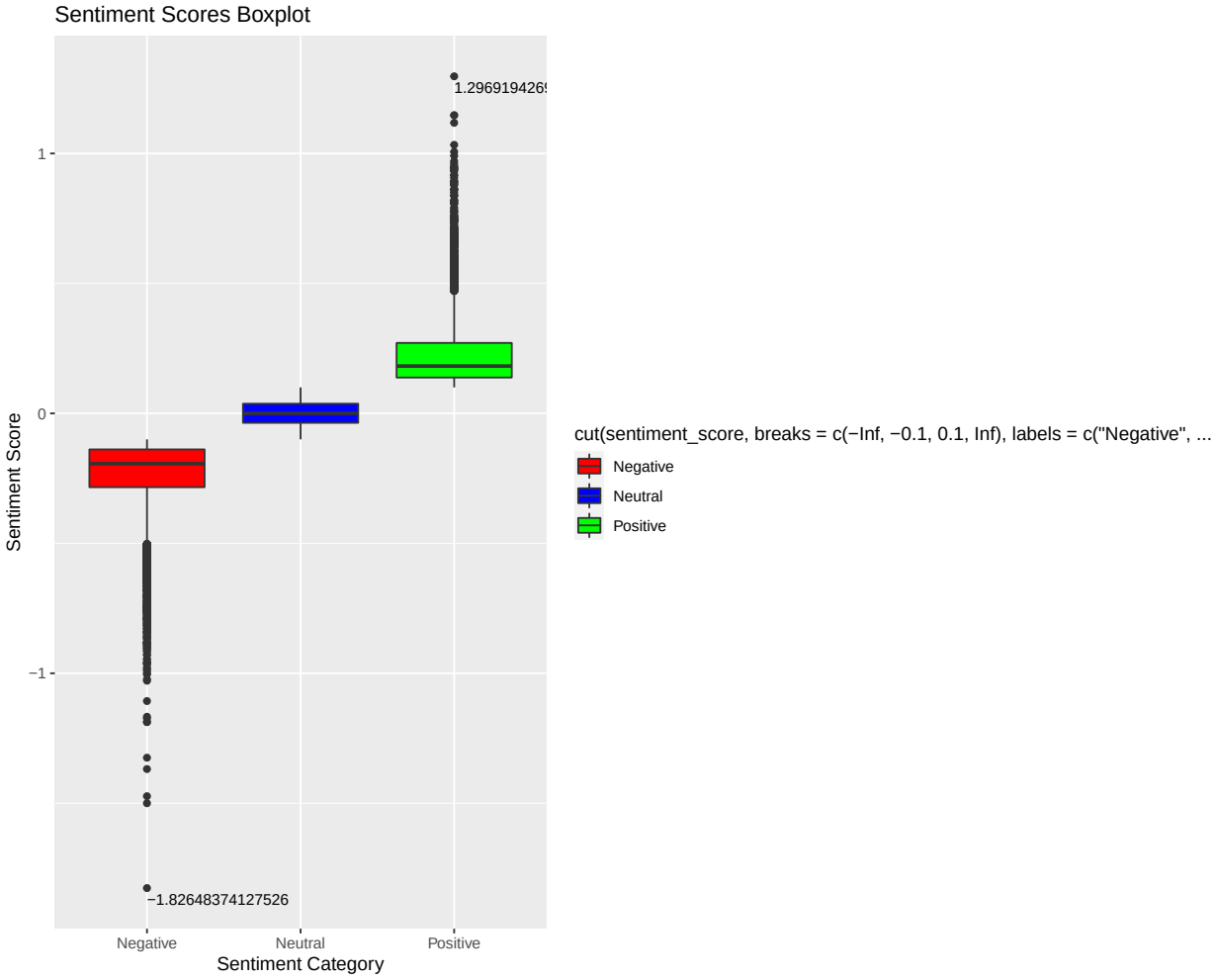
Plotting the scatterplots for the above created sentiment scores to understand the distribution of scores, we have -

```
ggplot(twitter_data, aes(x = 1:length(sentiment_score), y = sentiment_score, color =
↪  cut(sentiment_score,
    breaks = c(-Inf, -0.1, 0.1, Inf), labels = c("Negative", "Neutral", "Positive")))) +
    geom_point() + labs(title = "Sentiment Scores distribution", x = "Index", y =
    ↪  "Sentiment Score") +
    scale_color_manual(values = c("red", "blue", "green"))
```



Creating Box plot for outlier detection -

```
ggplot(twitter_data, aes(x = cut(sentiment_score, breaks = c(-Inf, -0.1, 0.1, Inf),
    labels = c("Negative", "Neutral", "Positive")), y = sentiment_score, fill =
    ↪  cut(sentiment_score,
    breaks = c(-Inf, -0.1, 0.1, Inf), labels = c("Negative", "Neutral", "Positive")))) +
    geom_boxplot() + geom_text(data = filter(twitter_data, sentiment_score ==
    ↪  max(sentiment_score)),
    aes(label = sentiment_score), hjust = 0, vjust = 1.5, color = "black", size = 3) +
    geom_text(data = filter(twitter_data, sentiment_score == min(sentiment_score)),
        aes(label = sentiment_score), hjust = 0, vjust = 1.5, color = "black", size = 3)
        ↪  +
    labs(title = "Sentiment Scores Boxplot", x = "Sentiment Category", y = "Sentiment
    ↪  Score") +
    scale_fill_manual(values = c("red", "blue", "green"))
```



- **Inferences:**
  - The scatter plot helped to visualize the distribution of sentiment scores in the Twitter data. The x-axis represents the index of the tweets, and the y-axis represents the sentiment scores.
  - The sentiment scores are color-coded into three categories: "Negative," "Neutral," and "Positive." The plot shows the distribution of sentiment scores across the tweets.
  - The box plot is used for outlier detection in the sentiment scores. The x-axis represents the sentiment categories ("Negative," "Neutral," and "Positive"), and the y-axis represents the sentiment scores.
  - The fill color of the boxes corresponds to the sentiment categories. The plot also includes text labels indicating the maximum sentiment score for both positive and negative outliers.
  - The analysis from the scatter plot indicates that the negative sentiment scores and positive sentiment scores are evenly distributed across the tweets. This means that there is a relatively equal volume of tweets with negative sentiment and tweets with positive sentiment. The distribution suggests that there is a balance between expressions of negative and positive sentiments regarding the topic being analyzed (e.g., climate change).
  - The analysis from the boxplot reveals the presence of outliers in both negative and positive sentiment scores. The maximum positive outlier is at 1.29, while the maximum negative outlier is at -1.83.
  - These outliers indicate extreme sentiment scores that deviate significantly from the majority of the sentiment scores in their respective categories.

Step 5: Conclusion

- **Statistical Analysis: Top N occurrences of tweet words (overall & per sentiment group)**
  - The analysis of Twitter tweets on climate change revealed several key insights. From the word frequency analysis, it was found that "climate" is the most commonly mentioned word, appearing in 33,458 tweets, followed by "global" with 10,534 occurrences. This indicates that climate-related discussions are highly prevalent on Twitter.
  - The word "trump" was mentioned in 3,517 tweets, suggesting his impact on climate-related matters. The word "change" had 2,633 occurrences, emphasizing the focus on the concept of change in the context of climate issues.
  - Further exploration based on sentiment revealed the top 3 words associated with each sentiment category. For news sentiment, "climate," "change," and "global" were frequently used, indicating a focus on factual news about climate change.
  - Positive sentiment tweets had repeated occurrences of "climate," "change," and "global," showing strong support for man-made climate change belief. Neutral sentiment tweets mentioned "climate," "change," and "global" without taking a specific stance, reflecting balanced representation. Negative sentiment tweets also featured "climate," "change," and "global," indicating skepticism or disbelief in man-made climate change.
  - Overall, this analysis provides valuable insights into the key topics and themes of climate change discussions on Twitter, highlighting the prevalence of climate-related content and the varying sentiments expressed by users.

- **Sentiment Analysis: How well does the sentiment scores reflect user sentiment?**
  - The sentiment scores provides insightful observations on sentiment analysis i.e., the general mood of the tweet messages. The twitter_data dataframe's top 10 rows are sorted by sentiment_score, revealing tweets with higher sentiment scores expressing positive sentiments, while lower scores convey negative sentiments.
  - However, it's important to note that some positive sentiment tweets may express disbelief or skepticism towards climate change, and vice versa for negative sentiment tweets. This highlights the complexities of sentiment analysis using NLP.
  - The scatter plot illustrated the distribution of sentiment scores, showing an even volume of negative and positive sentiment tweets. The box plot identified outliers with extreme sentiment scores i.e., maximum positive outlier is at 1.29, while the maximum negative outlier is at -1.83, indicating significant deviations from the majority.
  - Overall, the analysis provided valuable insights into sentiment distribution and the challenges of sentiment analysis in capturing the true sentiment behind text data.