

# SentimentAnalysis(Lumauag, Animas, Sanceda)

Matt Andrei Lumauag

2024-12-14

```
library(dplyr)
```

```
##  
## Attaching package: 'dplyr'  
  
## The following objects are masked from 'package:stats':  
##  
##   filter, lag  
  
## The following objects are masked from 'package:base':  
##  
##   intersect, setdiff, setequal, union
```

```
library(ggplot2)  
library(lubridate)
```

```
##  
## Attaching package: 'lubridate'  
  
## The following objects are masked from 'package:base':  
##  
##   date, intersect, setdiff, union
```

```
library(stringr)
```

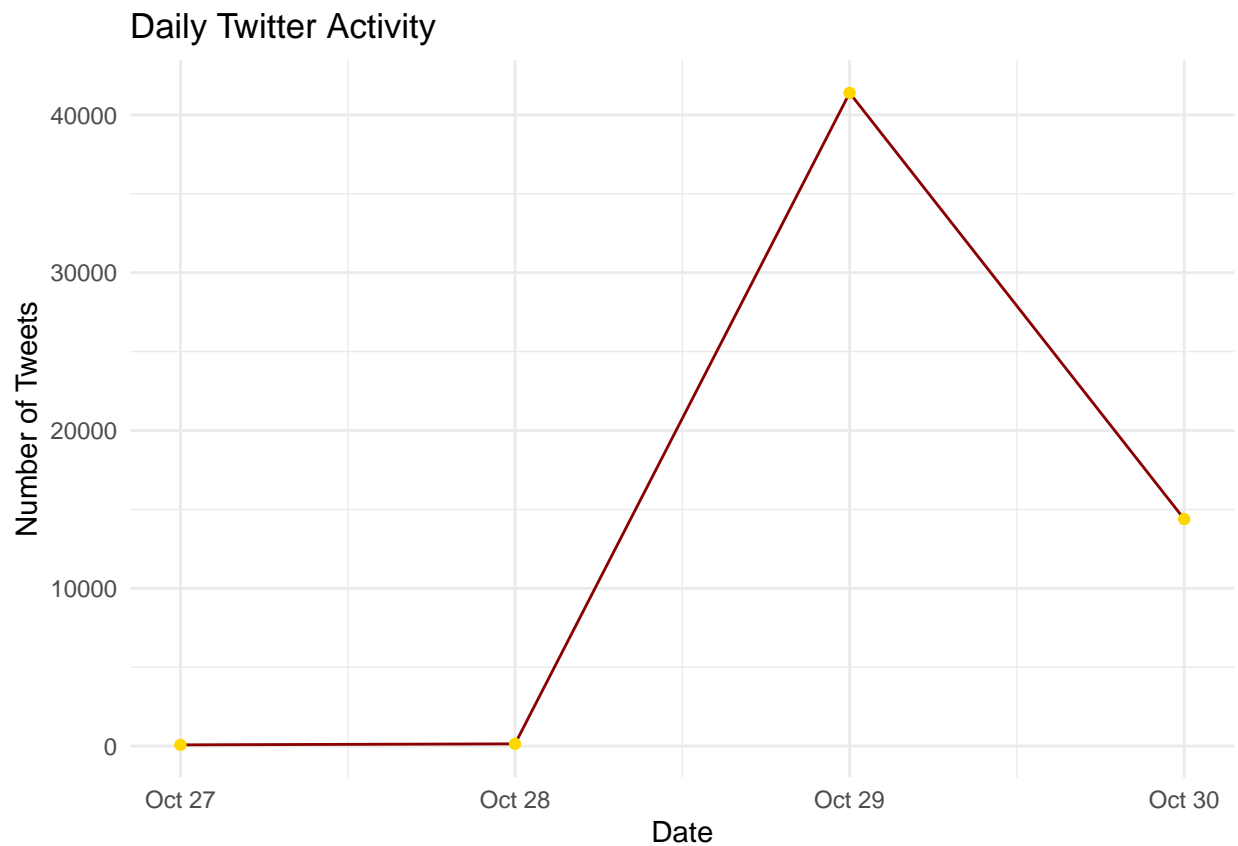
```
# Load dataset  
data_tweets <- read.csv("tweetsDF.csv")  
  
# Data cleaning and preparation  
cleaned_tweets <- data_tweets %>%  
  select(-c(statusSource, Created_At_Round)) %>%  
  mutate(timestamp = as.POSIXct(created, format = "%Y-%m-%d %H:%M:%S"),  
         tweet_date = as.Date(timestamp),  
         tweet_hour = hour(timestamp),  
         day_of_week = weekdays(timestamp)) %>%  
  distinct(text, .keep_all = TRUE)  
  
# Analyze daily tweet volume  
daily_tweets <- cleaned_tweets %>%
```

```

group_by(tweet_date) %>%
  summarise(total_tweets = n())

# Plot daily tweet trends
ggplot(daily_tweets, aes(x = tweet_date, y = total_tweets)) +
  geom_line(color = "darkred") +
  geom_point(color = "gold") +
  theme_minimal() +
  labs(title = "Daily Twitter Activity",
       x = "Date",
       y = "Number of Tweets")

```



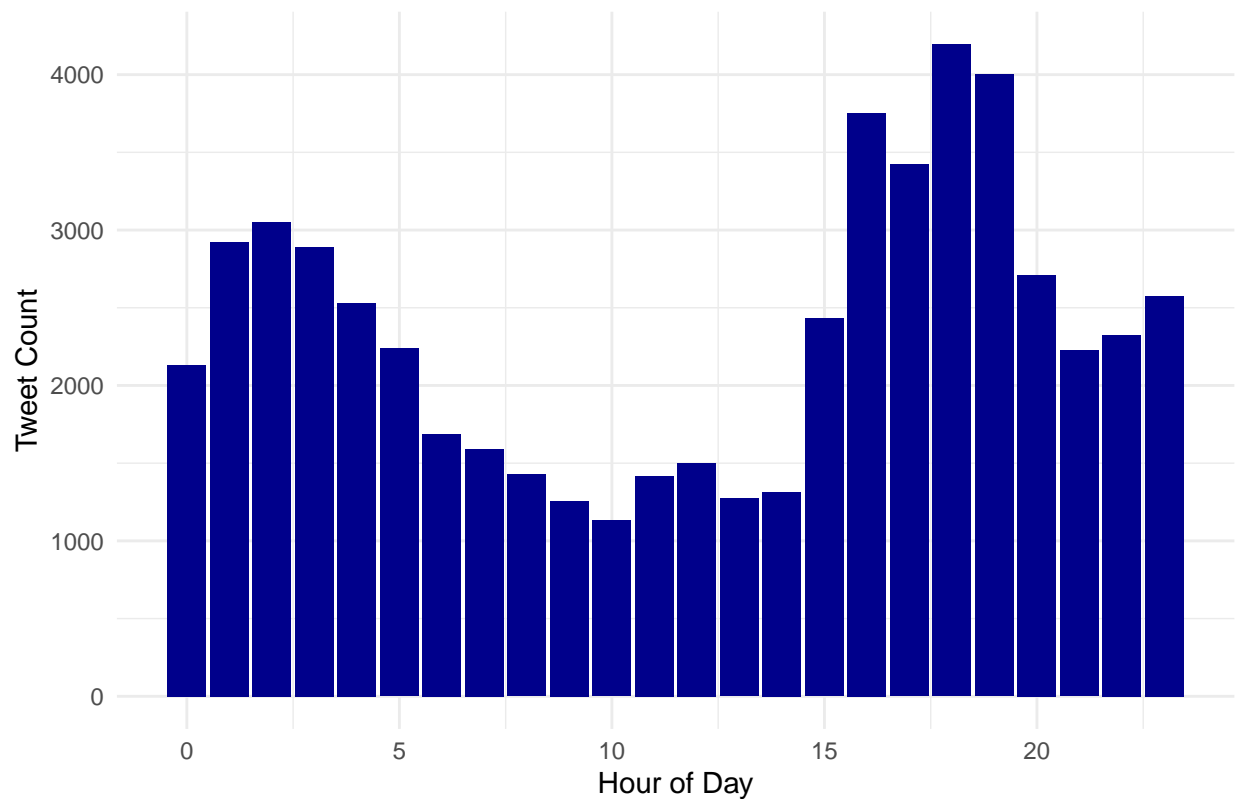
```

# Analyze hourly tweet patterns
hourly_tweets <- cleaned_tweets %>%
  group_by(tweet_hour) %>%
  summarise(total_tweets = n())

# Plot hourly tweet distribution
ggplot(hourly_tweets, aes(x = tweet_hour, y = total_tweets)) +
  geom_bar(stat = "identity", fill = "darkblue") +
  theme_minimal() +
  labs(title = "Hourly Distribution of Tweets",
       x = "Hour of Day",
       y = "Tweet Count")

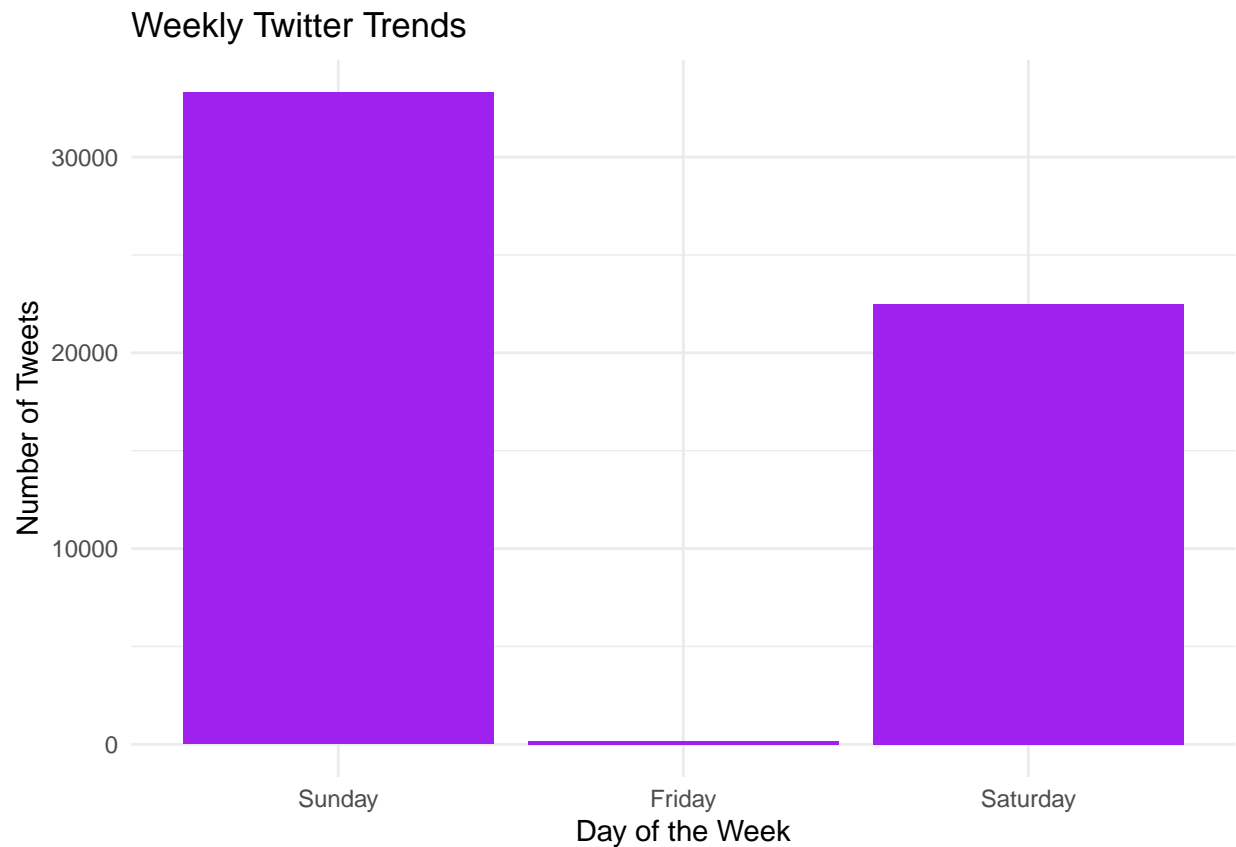
```

Hourly Distribution of Tweets



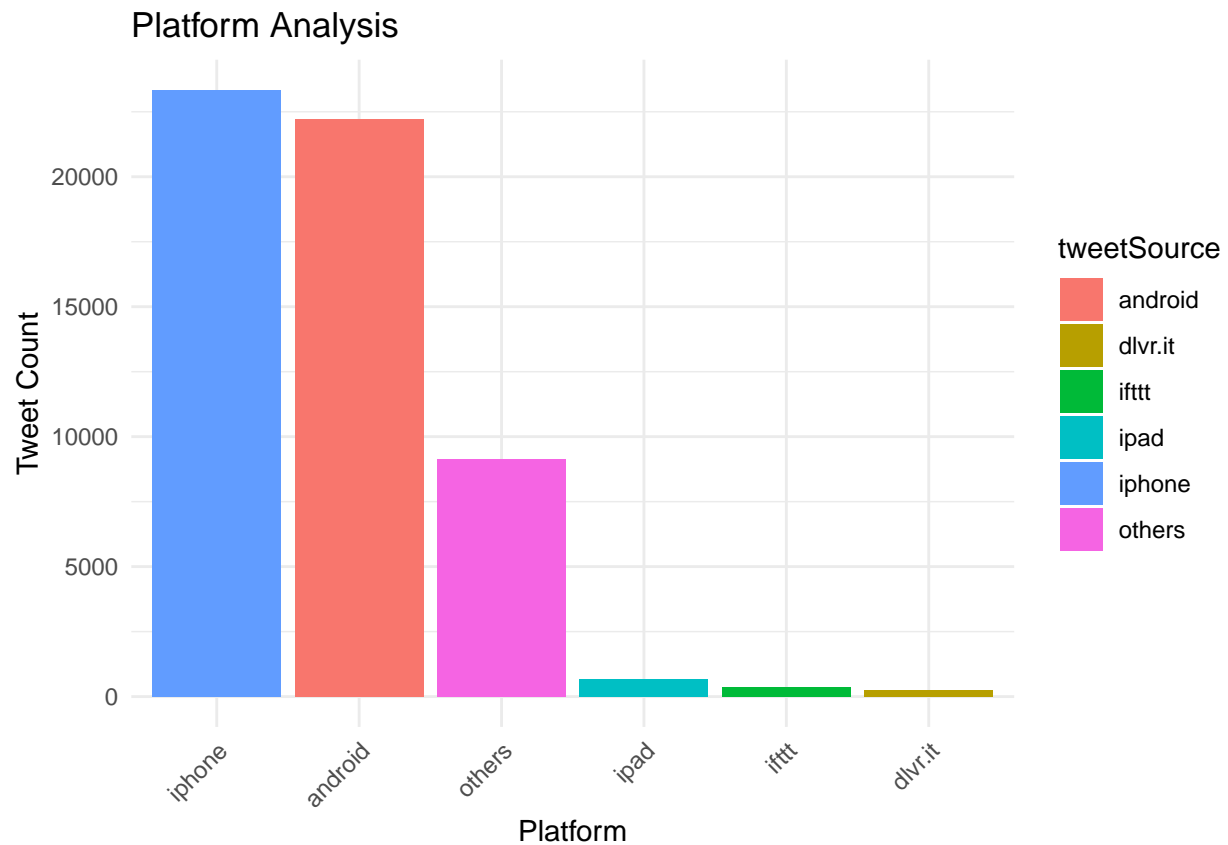
```
# Analyze weekly tweet activity
weekly_tweets <- cleaned_tweets %>%
  group_by(day_of_week) %>%
  summarise(total_tweets = n()) %>%
  mutate(day_of_week = factor(day_of_week,
                              levels = c("Sunday", "Monday", "Tuesday", "Wednesday",
                                           "Thursday", "Friday", "Saturday")))

# Plot weekly tweet trends
ggplot(weekly_tweets, aes(x = day_of_week, y = total_tweets)) +
  geom_bar(stat = "identity", fill = "purple") +
  theme_minimal() +
  labs(title = "Weekly Twitter Trends",
       x = "Day of the Week",
       y = "Number of Tweets")
```



```
# Analyze usage by platform
platform_analysis <- cleaned_tweets %>%
  group_by(tweetSource) %>%
  summarise(total_usage = n())

# Plot platform usage
ggplot(platform_analysis, aes(x = reorder(tweetSource, -total_usage), y = total_usage, fill = tweetSource)) +
  geom_bar(stat = "identity") +
  theme_minimal() +
  labs(title = "Platform Analysis",
       x = "Platform",
       y = "Tweet Count") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



#### daily\_tweets

```
## # A tibble: 4 x 2
##   tweet_date total_tweets
##   <date>         <int>
## 1 2022-10-27         78
## 2 2022-10-28        143
## 3 2022-10-29       41392
## 4 2022-10-30       14387
```

#### hourly\_tweets

```
## # A tibble: 24 x 2
##   tweet_hour total_tweets
##   <int>         <int>
## 1      0         2131
## 2      1         2922
## 3      2         3050
## 4      3         2892
## 5      4         2529
## 6      5         2237
## 7      6         1688
## 8      7         1592
## 9      8         1428
## 10     9         1256
```

```
## # i 14 more rows
```

```
weekly_tweets
```

```
## # A tibble: 3 x 2
##   day_of_week total_tweets
##   <fct>         <int>
## 1 Friday           181
## 2 Saturday        22509
## 3 Sunday          33310
```

```
platform_analysis
```

```
## # A tibble: 6 x 2
##   tweetSource total_usage
##   <chr>         <int>
## 1 android        22227
## 2 dlvr.it         241
## 3 ifttt           364
## 4 ipad            685
## 5 iphone        23336
## 6 others          9147
```

```
library(dplyr)
library(tidytext)
```

```
## Warning: package 'tidytext' was built under R version 4.4.2
```

```
library(ggplot2)
library(tidytext)
library(textdata)
```

```
## Warning: package 'textdata' was built under R version 4.4.2
```

```
Selection <- 1

sentiment_data <- read.csv("tweetsDF.csv")

cleaned_sentiments <- sentiment_data %>%
  select(text) %>%
  distinct(text, .keep_all = TRUE)

# Tokenize tweet text
tokenized_data <- cleaned_sentiments %>%
  unnest_tokens(word, text)

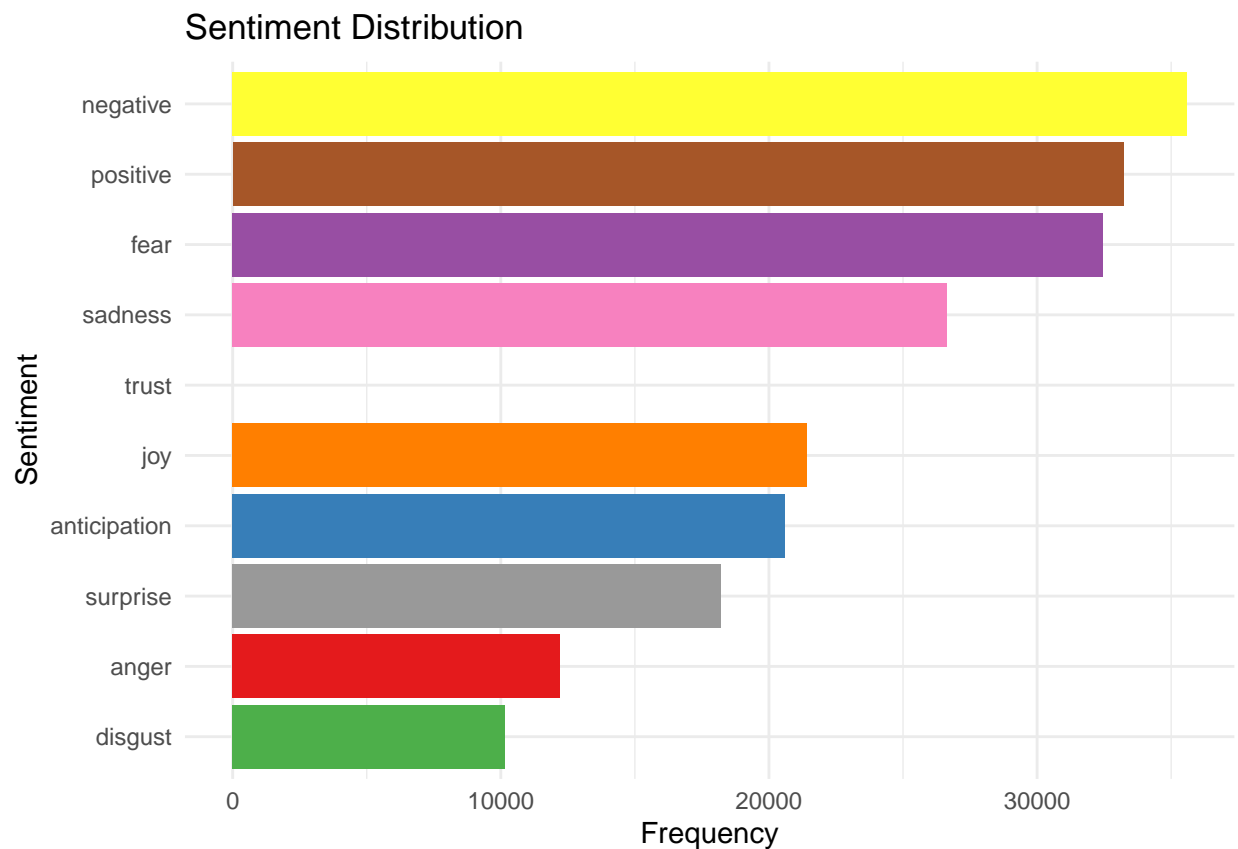
# Remove common stop words
data("stop_words")
filtered_tokens <- tokenized_data %>%
  anti_join(stop_words, by = "word")
```

```
# Perform sentiment analysis
nrc_lexicon <- get_sentiments("nrc")
sentiment_counts <- filtered_tokens %>%
  inner_join(nrc_lexicon, by = "word") %>%
  count(sentiment, sort = TRUE)
```

```
## Warning in inner_join(., nrc_lexicon, by = "word"): Detected an unexpected many-to-many relationship
## i Row 1 of 'x' matches multiple rows in 'y'.
## i Row 1995 of 'y' matches multiple rows in 'x'.
## i If a many-to-many relationship is expected, set 'relationship =
##   "many-to-many"' to silence this warning.
```

```
# Plot sentiment distribution
ggplot(sentiment_counts, aes(x = reorder(sentiment, n), y = n, fill = sentiment)) +
  geom_bar(stat = "identity", show.legend = FALSE) +
  coord_flip() +
  theme_minimal() +
  labs(title = "Sentiment Distribution",
       x = "Sentiment",
       y = "Frequency") +
  scale_fill_brewer(palette = "Set1")
```

```
## Warning in RColorBrewer::brewer.pal(n, pal): n too large, allowed maximum for palette Set1 is 9
## Returning the palette you asked for with that many colors
```



```
sentiment_counts
```

```
##      sentiment      n
## 1    negative 35579
## 2    positive 33225
## 3      fear 32463
## 4    sadness 26640
## 5     trust 23613
## 6      joy 21414
## 7 anticipation 20586
## 8    surprise 18218
## 9      anger 12215
## 10   disgust 10162
```

```
library(dplyr)
library(tidytext)
library(ggplot2)
library(lubridate)

# Load the dataset
tweets_data <- read.csv("tweetsDF.csv")

# Data Cleaning
cleaned_tweets <- tweets_data %>%
  select(created, text) %>%
  distinct(text, .keep_all = TRUE) %>%
  filter(!is.na(text))

# Convert dates
cleaned_tweets$created <- as.Date(cleaned_tweets$created)

# Tokenize text
tokenized_words <- cleaned_tweets %>%
  unnest_tokens(word, text)

# Remove stop words
data("stop_words")
tokenized_words <- tokenized_words %>%
  anti_join(stop_words, by = "word")

# NRC Sentiments
nrc_sentiments <- get_sentiments("nrc")
word_sentiment <- tokenized_words %>%
  inner_join(nrc_sentiments, by = "word") %>%
  count(created, sentiment, sort = TRUE)
```

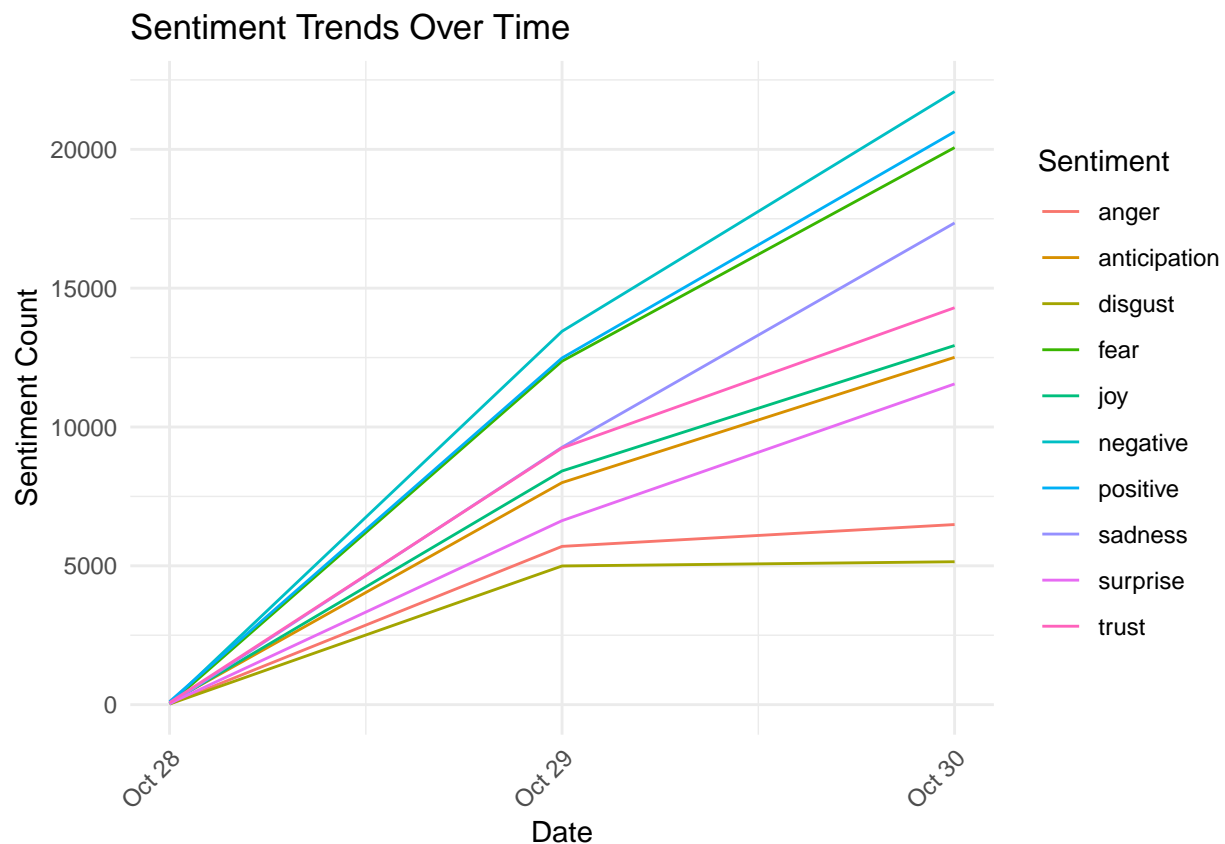
```
## Warning in inner_join(., nrc_sentiments, by = "word"): Detected an unexpected many-to-many relationship
## i Row 1 of 'x' matches multiple rows in 'y'.
## i Row 1995 of 'y' matches multiple rows in 'x'.
## i If a many-to-many relationship is expected, set 'relationship =
##   "many-to-many"' to silence this warning.
```



```
# Sentiment Trends
sentiment_trends <- word_sentiment %>%
  group_by(created, sentiment) %>%
  summarise(daily_sentiment_count = sum(n)) %>%
  ungroup()
```

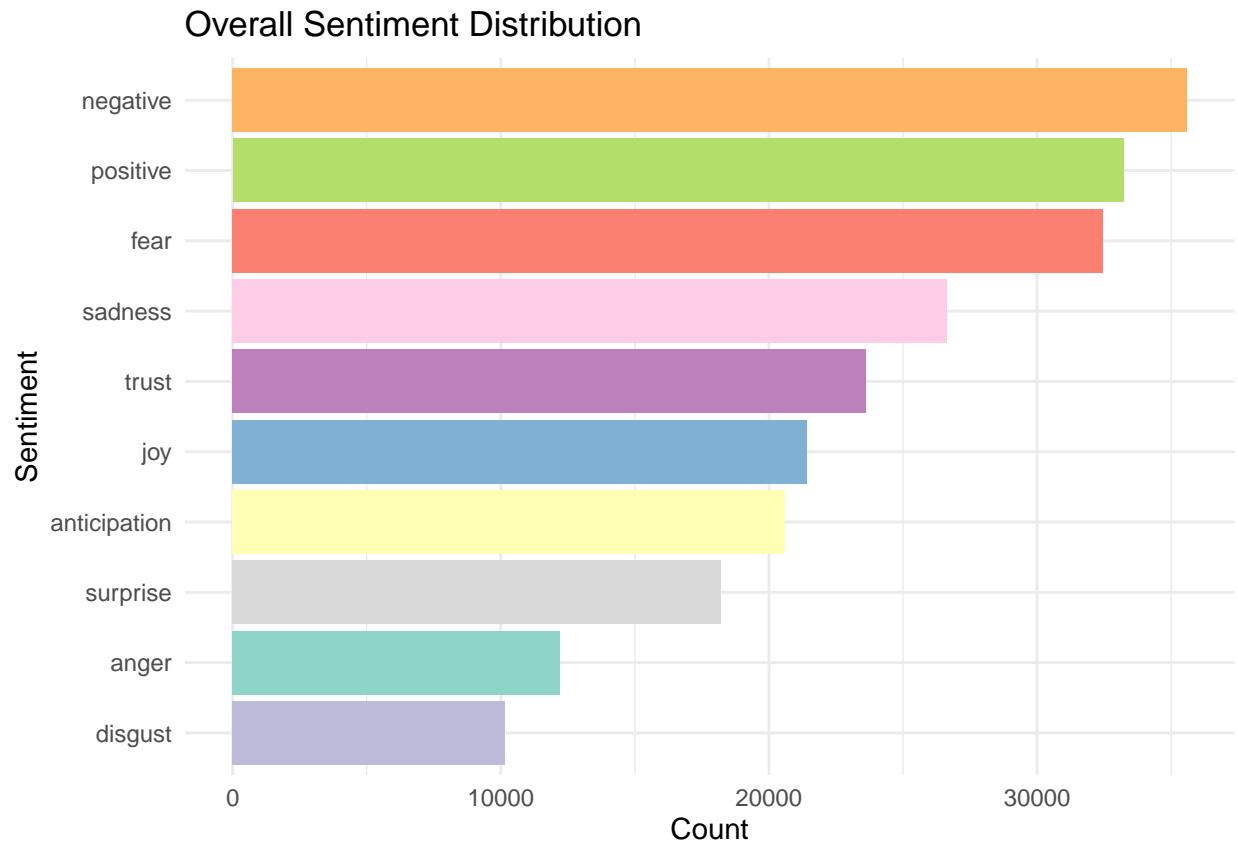
## 'summarise()' has grouped output by 'created'. You can override using the  
## '.groups' argument.

```
# Plot Sentiment Trends
ggplot(sentiment_trends, aes(x = created, y = daily_sentiment_count, color = sentiment)) +
  geom_line() +
  theme_minimal() +
  labs(title = "Sentiment Trends Over Time",
       x = "Date",
       y = "Sentiment Count",
       color = "Sentiment") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



```
# Sentiment Distribution
sentiment_distribution <- word_sentiment %>%
  group_by(sentiment) %>%
  summarise(sentiment_count = sum(n)) %>%
  ungroup()
```

```
# Plot Sentiment Distribution
ggplot(sentiment_distribution, aes(x = reorder(sentiment, sentiment_count), y = sentiment_count, fill =
  geom_bar(stat = "identity", show.legend = FALSE) +
  coord_flip() +
  theme_minimal() +
  labs(title = "Overall Sentiment Distribution",
    x = "Sentiment",
    y = "Count") +
  scale_fill_brewer(palette = "Set3")
```



```
# Positive and Negative Tweets
positive_tweets_count <- word_sentiment %>%
  filter(sentiment == "positive") %>%
  summarise(positive_tweet_count = sum(n))

negative_tweets_count <- word_sentiment %>%
  filter(sentiment == "negative") %>%
  summarise(negative_tweet_count = sum(n))

print(paste("Number of Positive Tweets: ", positive_tweets_count$positive_tweet_count))
```

```
## [1] "Number of Positive Tweets: 33225"
```

```
print(paste("Number of Negative Tweets: ", negative_tweets_count$negative_tweet_count))
```

```
## [1] "Number of Negative Tweets: 35579"
```

```
if (negative_tweets_count$negative_tweet_count > positive_tweets_count$positive_tweet_count) {  
  message <- "Warning: The number of negative sentiments is high. Immediate action is recommended to address this issue."  
} else {  
  message <- "Positive feedback is dominant. Consider amplifying positive campaigns to maintain momentum."  
}
```