Sentiments-Analysis

2024-12-18

```
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
```

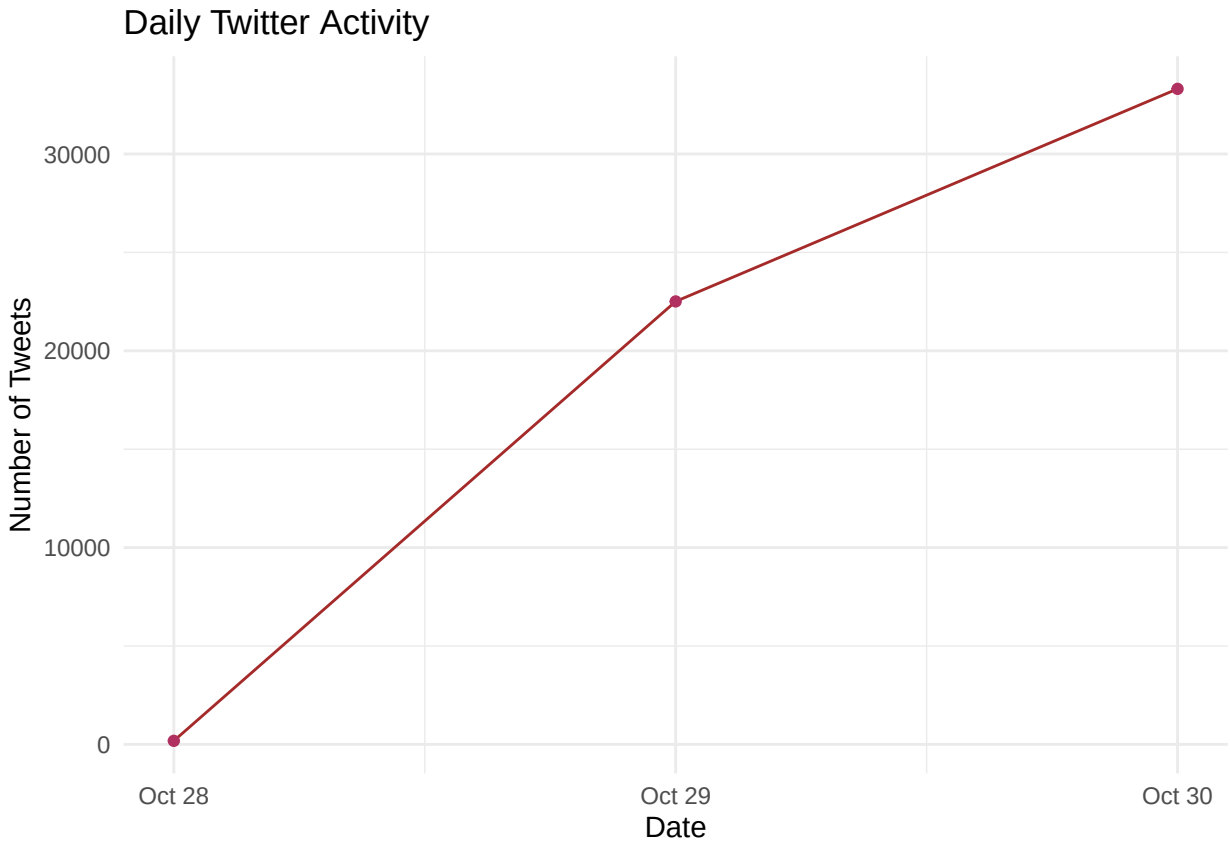
```
library(ggplot2)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

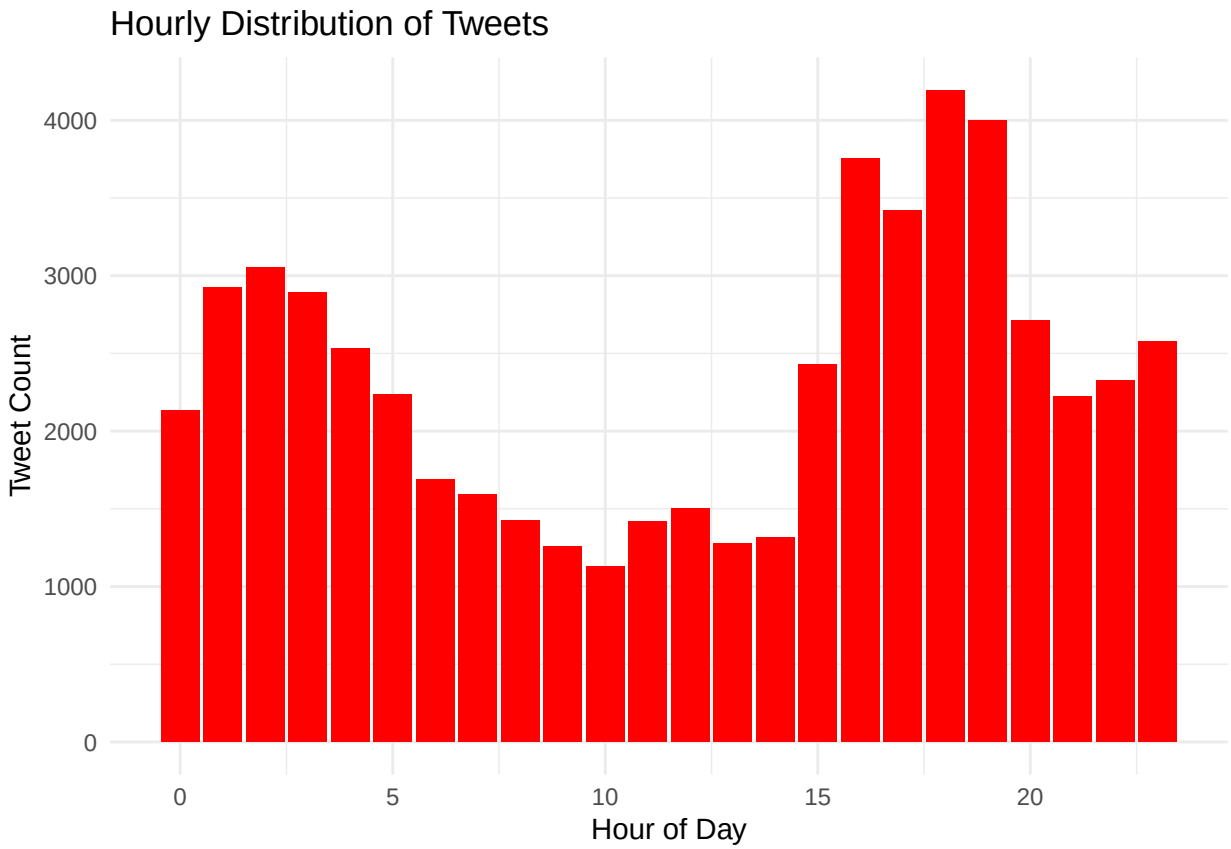
```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(stringr)

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

# Data cleaning and preparation
processed_tweets <- 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_summary <- processed_tweets %>%
  group_by(tweet_date) %>%
  summarise(total_tweets = n())

# Plot daily tweet trends
ggplot(daily_summary, aes(x = tweet_date, y = total_tweets)) +
  geom_line(color = "brown") +
  geom_point(color = "maroon") +
  theme_minimal() +
  labs(title = "Daily Twitter Activity",
```

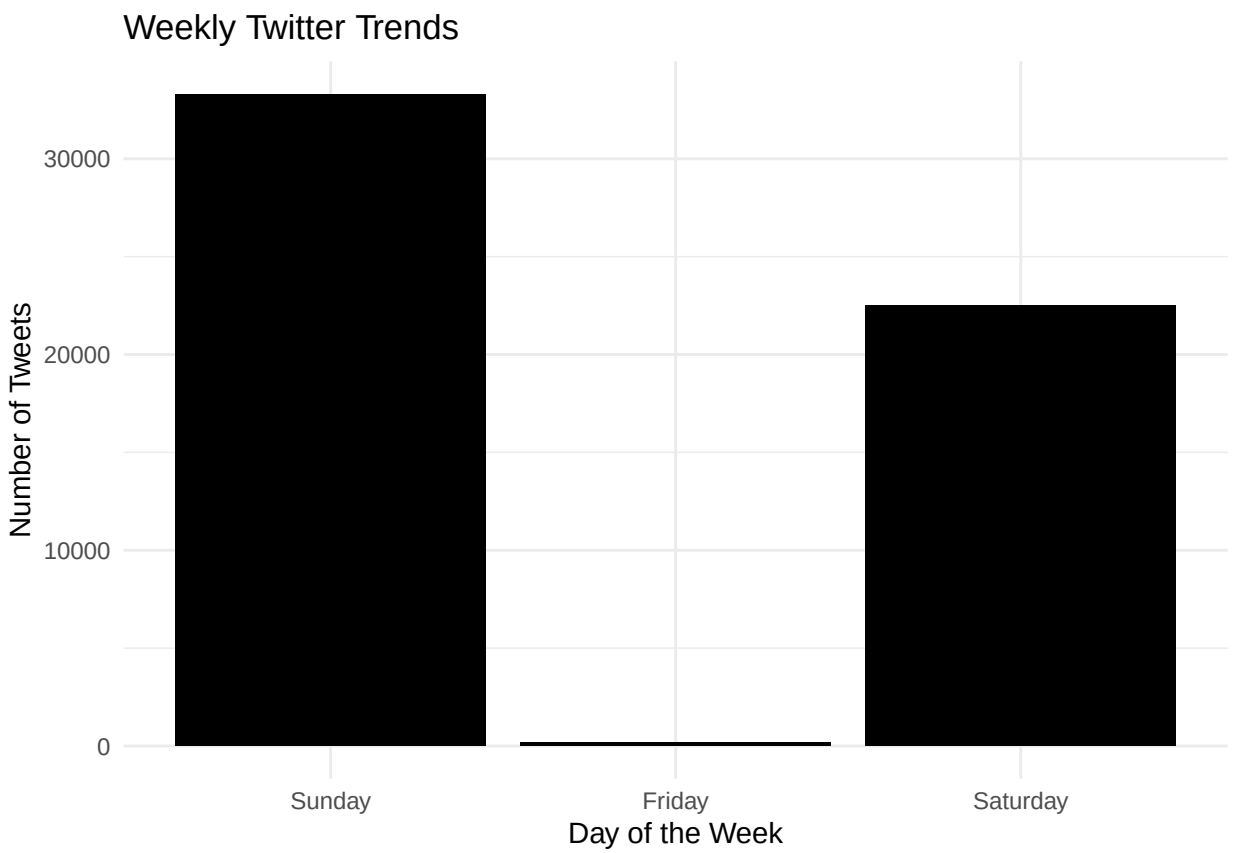
```
       x = "Date",
       y = "Number of Tweets")
```



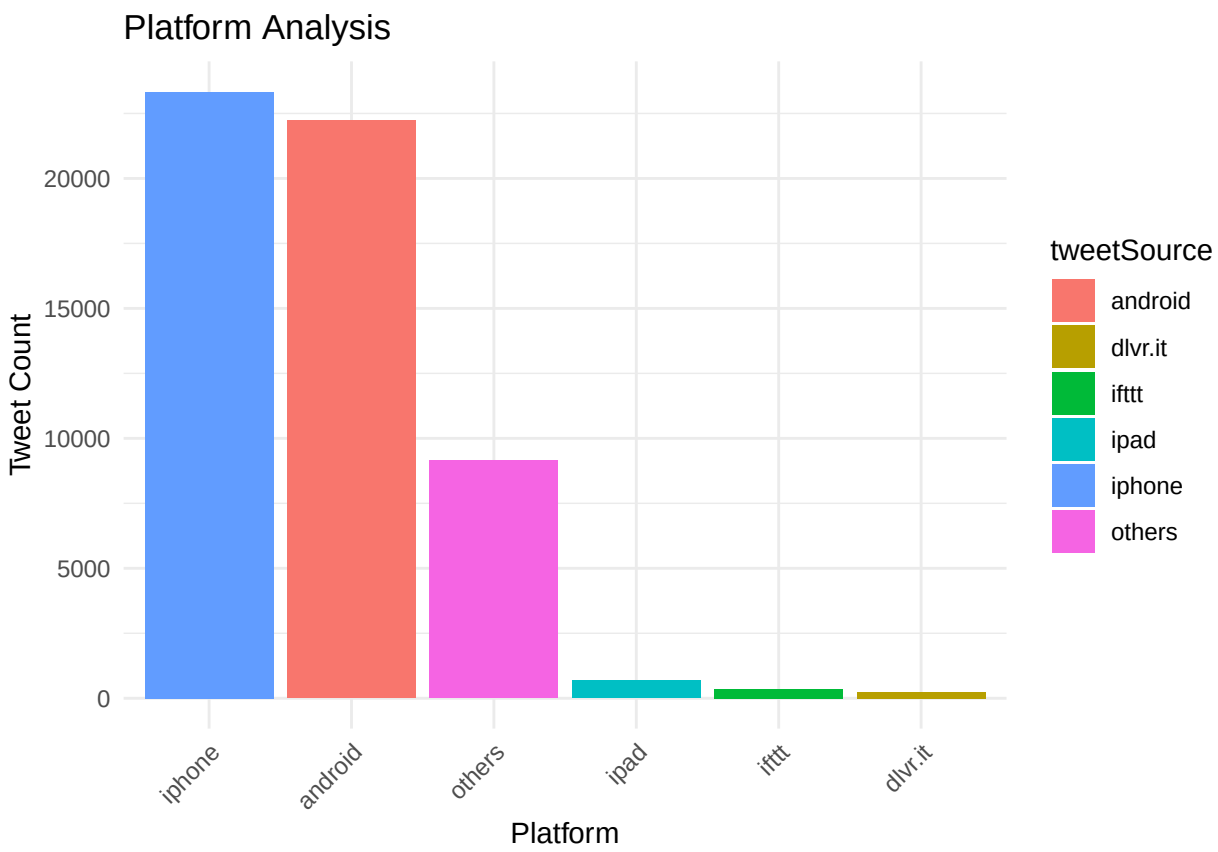

```
# Analyze hourly tweet patterns
hourly_summary <- processed_tweets %>%
  group_by(tweet_hour) %>%
  summarise(total_tweets = n())

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




```r
# Analyze weekly tweet activity
weekly_summary <- processed_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_summary, aes(x = day_of_week, y = total_tweets)) +
  geom_bar(stat = "identity", fill = "black") +
  theme_minimal() +
  labs(title = "Weekly Twitter Trends",
       x = "Day of the Week",
       y = "Number of Tweets")
```




```r
# Analyze usage by platform
platform_summary <- processed_tweets %>%
  group_by(tweetSource) %>%
  summarise(total_usage = n())

# Plot platform usage
ggplot(platform_summary, 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))
```

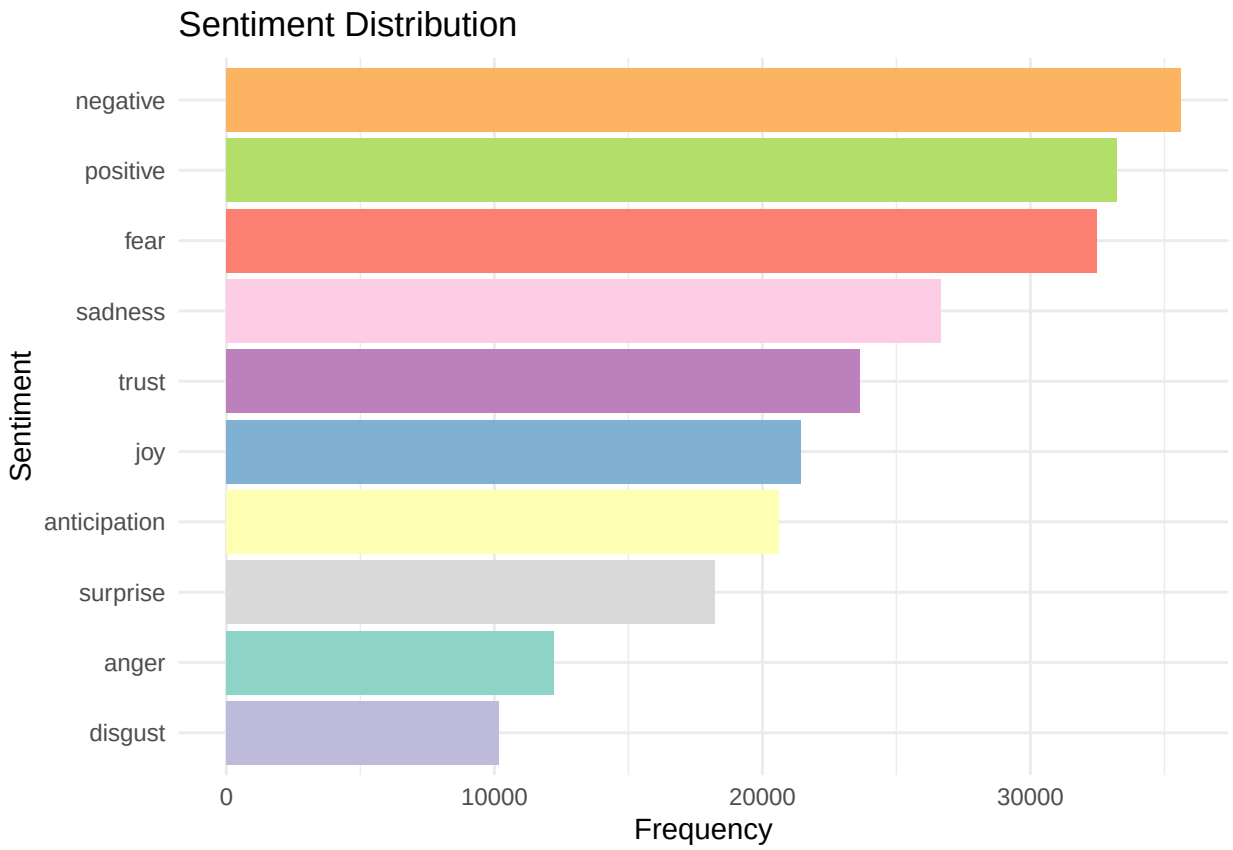



```
library(dplyr)
library(tidytext)
library(ggplot2)
library(textdata)

selected_tweet_id <- 1

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

unique_tweets <- tweet_data %>%
  select(text) %>%
  distinct(text, .keep_all = TRUE)


# Tokenize tweet text
tokenized_tweets <- unique_tweets %>%
  unnest_tokens(word, text)

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

# Perform sentiment analysis
nrc_sentiment_lexicon <- get_sentiments("nrc")
sentiment_distribution <- clean_tokens %>%
  inner_join(nrc_sentiment_lexicon, by = "word") %>%
```

```
  count(sentiment, sort = TRUE)
```

```
## Warning in inner_join(., nrc_sentiment_lexicon, by = "word"): Detected an unexpected many-to-many rel
## 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_distribution, 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 = "Set3")
```




```
sentiment_distribution
```

```
##       sentiment     n
## 1      negative 35597
## 2      positive 33241
## 3          fear 32481
## 4       sadness 26655
## 5         trust 23628
## 6           joy 21429
## 7  anticipation 20600
```

```
## 8      surprise 18234
## 9         anger 12221
## 10      disgust 10166
```

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

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

cleaned_tweet_data <- tweets_dataset %>%
  select(created, text) %>%
  distinct(text, .keep_all = TRUE) %>%
  filter(!is.na(text))

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

# Tokenize text
tokenized_words_data <- cleaned_tweet_data %>%
  unnest_tokens(word, text)

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

# NRC Sentiments
nrc_sentiment_lexicon <- get_sentiments("nrc")
word_sentiment_data <- filtered_tokenized_words %>%
  inner_join(nrc_sentiment_lexicon, by = "word") %>%
  count(created, sentiment, sort = TRUE)
```

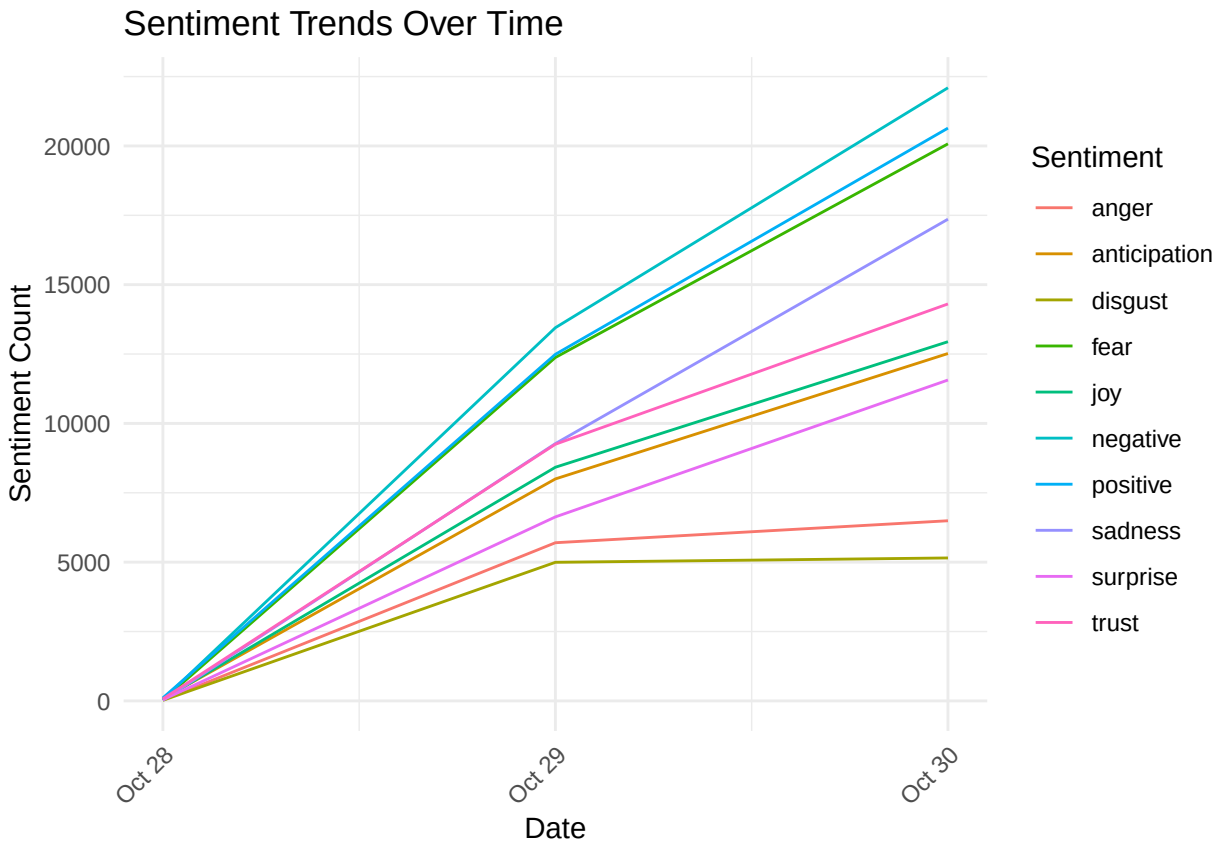
```
## Warning in inner_join(., nrc_sentiment_lexicon, by = "word"): Detected an unexpected many-to-many rel
## i Row 1 of `x` matches multiple rows in `y`.
## i Row 1995 of `y` matches multiple rows in `x`.
## i If a many-to-many relationship is expected, set `relationship =
##   "many-to-many"` to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

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

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

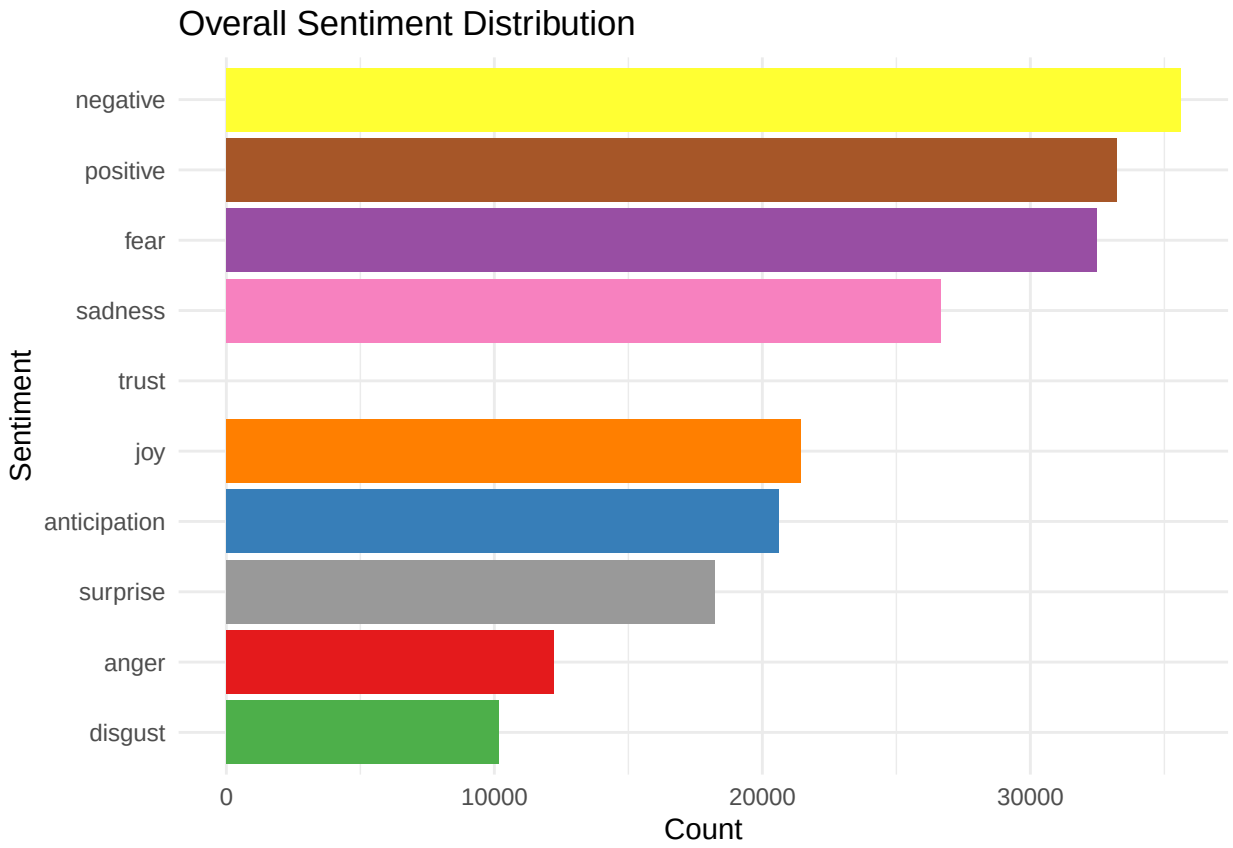
```r
# Plot Sentiment Trends
ggplot(sentiment_trend_data, aes(x = created, y = daily_sentiment_count, color = sentiment)) +
  geom_line() +
  theme_minimal() +
  labs(title = "Sentiment Trends Over Time",
       x = "Date",
```

```
       y = "Sentiment Count",
       color = "Sentiment") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```




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

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




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

negative_sentiment_count <- word_sentiment_data %>%
  filter(sentiment == "negative") %>%
  summarise(negative_tweet_count = sum(n))

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

```
## [1] "Number of Positive Tweets:  33241"
```

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

```
## [1] "Number of Negative Tweets:  35597"
```

```
# Check if negative sentiment outweighs positive sentiment
if (negative_sentiment_count$negative_tweet_count > positive_sentiment_count$positive_tweet_count) {
  action_message <- "Alert: Negative sentiment is prevalent. It is advisable to take immediate steps to
} else {
  action_message <- "Positive sentiment is leading. It may be beneficial to strengthen positive messagi
}
```