SentimentAnalysisProject

Andica, Benedicto, Cautivar

2024-12-07

Reading the tweetsDF.csv file

```r
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
```

```r
library(ggplot2)
library(stringr)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.5
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
library(syuzhet)
library(tm)
```

```
## Loading required package: NLP
##
## Attaching package: 'NLP'
##
## The following object is masked from 'package:ggplot2':
##
##     annotate
```

```r
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```r
library(RColorBrewer)
```

```r
tweetsDF <- read.csv("tweetsDF.csv")
```

Cleaning Text

```r
tweetsDF$text <- iconv(tweetsDF$text, from = "UTF-8", to = "ASCII//TRANSLIT", sub = "")

keywords <- "\\b(blackpink|yg|bornpink|lisa|jennie|rose|jisoo)\\b|:\\(\\(|&amp;|!|:\\(|&lt;/3|:|&lt;|/|

tweetsDF$text <- tolower(tweetsDF$text)
tweetsDF$text <- gsub("https\\S+", "", tweetsDF$text)
tweetsDF$text <- gsub("#", "", gsub("\n", " ", tweetsDF$text))
tweetsDF$text <- gsub("([@?]\\S+)", "", tweetsDF$text)
tweetsDF$text <- gsub("\\?", "", tweetsDF$text)
tweetsDF$text <- gsub("\\b\\d{2}\\.\\d{2}\\.\\d{4}\\b", "", tweetsDF$text)
tweetsDF$text <- gsub(keywords, "", tweetsDF$text, ignore.case = TRUE)
tweetsDF$text <- gsub("<a href=httptwitter.comdownloadandroid rel=nofollow>twitter for android<a>", "",
tweetsDF$text <- gsub("<a href= rel=nofollow>twitter web app<a>", "", tweetsDF$text)
tweetsDF$text <- gsub("<a href=httptwitter.comdownloadiphone rel=nofollow>twitter for iphone<a>", "", tw
tweetsDF$text <- gsub("<a href=([^>]*?) rel=nofollow>([^<]*?)<a>", "", tweetsDF$text)
tweetsDF$text <- gsub("30102022", "", tweetsDF$text)
tweetsDF$text <- gsub("\\s+", " ", tweetsDF$text)

create_chunks <- function(df, start_row, end_row) {
  return(df[start_row:end_row, ])
}

start_row <- 1
end_row <- 1000
chunk_data <- tweetsDF[start_row:end_row, ]

valid_texts <- chunk_data$text[chunk_data$text != ""]
cat("Number of valid texts before preprocessing: ", length(valid_texts), "\n")
```

```
## Number of valid texts before preprocessing:  1000
```

```r
if (length(valid_texts) > 0) {

  corpus <- Corpus(VectorSource(valid_texts))

  corpus <- tm_map(corpus, content_transformer(tolower))
  cat("Number of valid texts after converting to lowercase: ", length(corpus), "\n")

  corpus <- tm_map(corpus, removePunctuation)
  cat("Number of valid texts after removing punctuation: ", length(corpus), "\n")

  corpus <- tm_map(corpus, removeNumbers)
  cat("Number of valid texts after removing numbers: ", length(corpus), "\n")

  corpus <- tm_map(corpus, removeWords, stopwords("en"))
  cat("Number of valid texts after removing stopwords: ", length(corpus), "\n")

  corpus <- tm_map(corpus, stripWhitespace)
  cat("Number of valid texts after stripping whitespace: ", length(corpus), "\n")
```

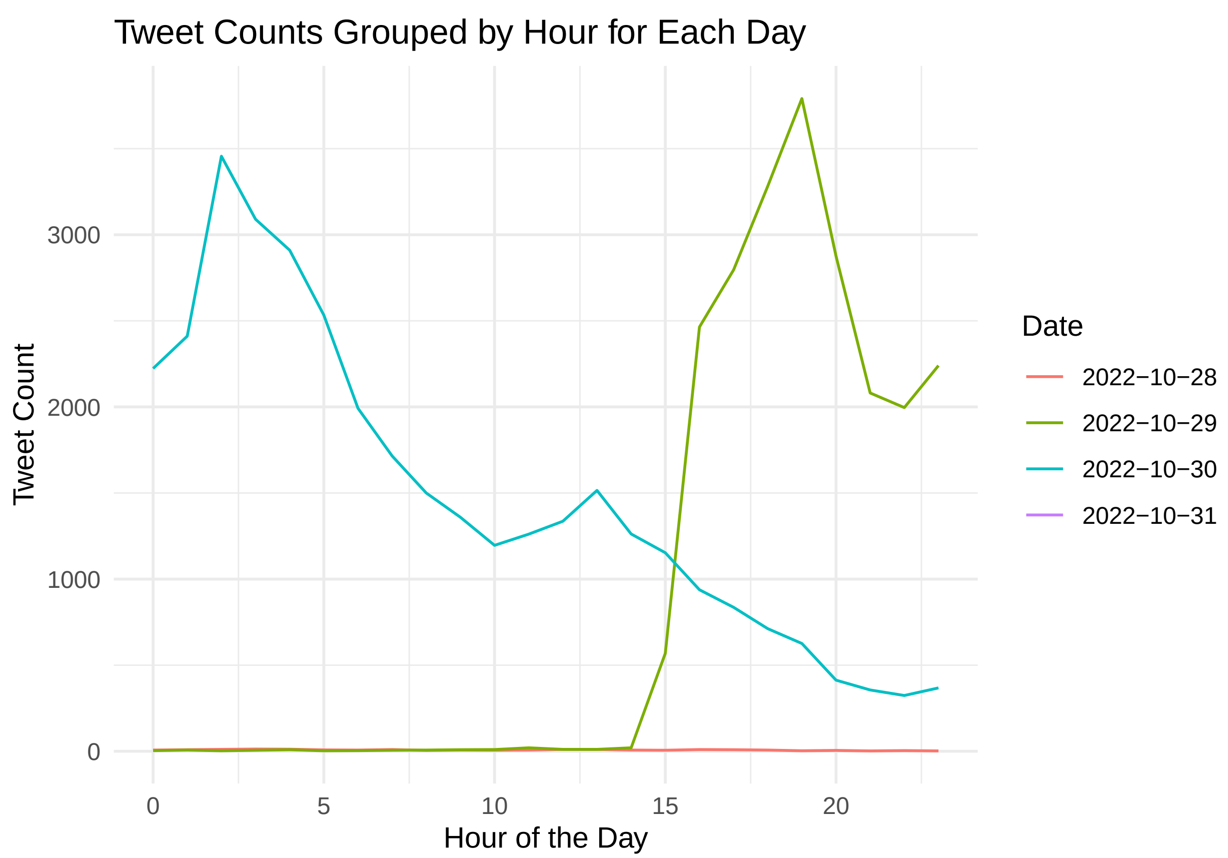
```
  if (length(corpus) > 0) {
    wordcloud(corpus,
              max.words = 100,
              random.order = FALSE,
              colors = brewer.pal(8, "Dark2"),
              scale = c(3, 0.5))
  } else {
    cat("No valid text left to create a word cloud.\n")
  }
} else {
  cat("No valid texts available to create a word cloud.\n")
}
```

```
## Warning in tm_map.SimpleCorpus(corpus, content_transformer(tolower)):
## transformation drops documents

## Number of valid texts after converting to lowercase:  1000

## Warning in tm_map.SimpleCorpus(corpus, removePunctuation): transformation drops
## documents

## Number of valid texts after removing punctuation:  1000

## Warning in tm_map.SimpleCorpus(corpus, removeNumbers): transformation drops
## documents

## Number of valid texts after removing numbers:  1000

## Warning in tm_map.SimpleCorpus(corpus, removeWords, stopwords("en")):
## transformation drops documents

## Number of valid texts after removing stopwords:  1000

## Warning in tm_map.SimpleCorpus(corpus, stripWhitespace): transformation drops
## documents

## Number of valid texts after stripping whitespace:  1000
```

Cleaning Dates

```
tweetsDF$Created_At_Round <- as.POSIXct(tweetsDF$Created_At_Round, format = "%d/%m/%Y %H:%M", tz = "UTC

tweetsDF$date <- as.Date(tweetsDF$Created_At_Round)
```

```r
tweetsDF$hour <- format(tweetsDF$Created_At_Round, "%H")

groupedData <- tweetsDF %>%
  group_by(date, hour) %>%
  summarise(count = n(), .groups = "drop")

groupedData$hour <- as.numeric(groupedData$hour)

ggplot(groupedData, aes(x = hour, y = count, color = as.factor(date), group = date)) +
  geom_line() +
  labs(
    title = "Tweet Counts Grouped by Hour for Each Day",
    x = "Hour of the Day",
    y = "Tweet Count",
    color = "Date"
  ) +
  theme_minimal() +
  theme(legend.position = "right")
```

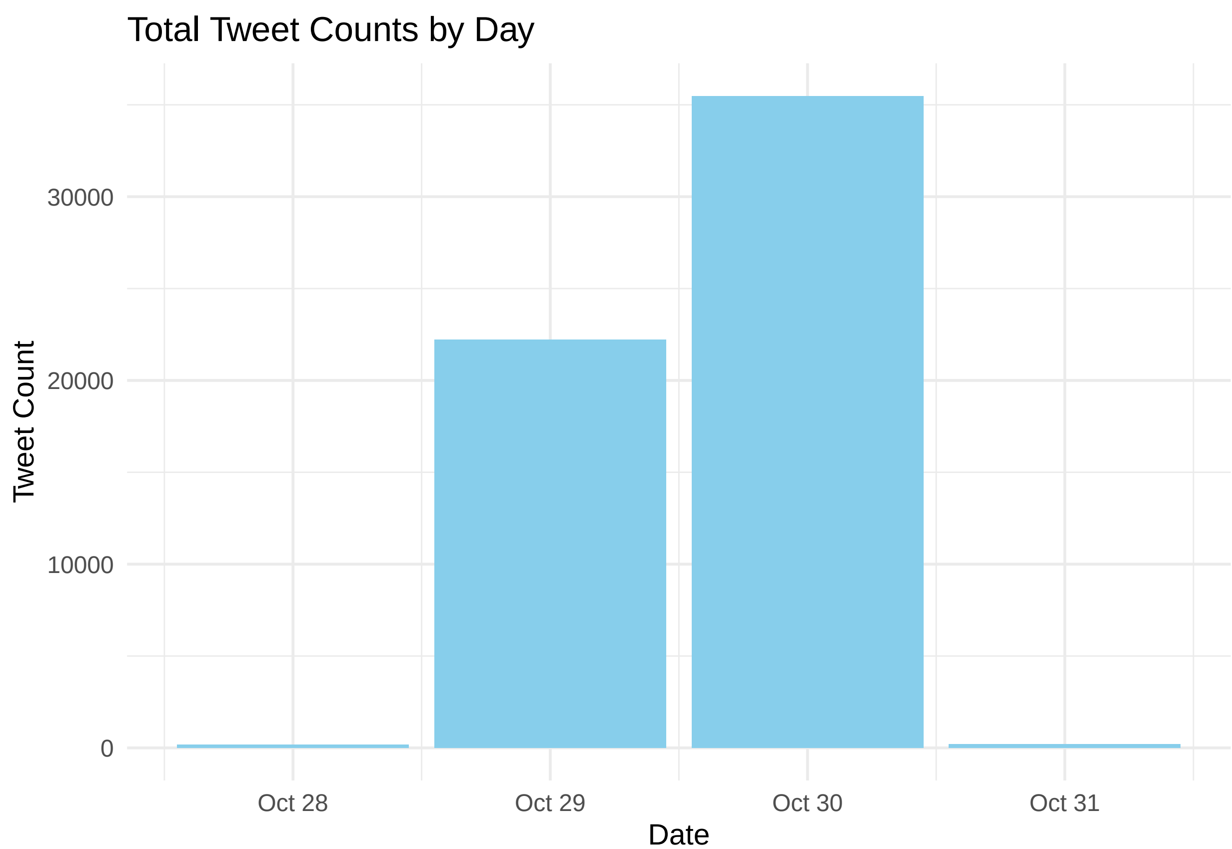



```r
dailyCounts <- tweetsDF %>%
  group_by(date) %>%
  summarise(total_tweets = n(), .groups = "drop")

print(dailyCounts)
```

```
## # A tibble: 4 x 2
##   date       total_tweets
```

```
##   <date>           <int>
## 1 2022-10-28         179
## 2 2022-10-29       22225
## 3 2022-10-30       35485
## 4 2022-10-31         197
```

```
ggplot(dailyCounts, aes(x = date, y = total_tweets)) +
  geom_bar(stat = "identity", fill = "skyblue") +
  labs(
    title = "Total Tweet Counts by Day",
    x = "Date",
    y = "Tweet Count"
  ) +
  theme_minimal()
```



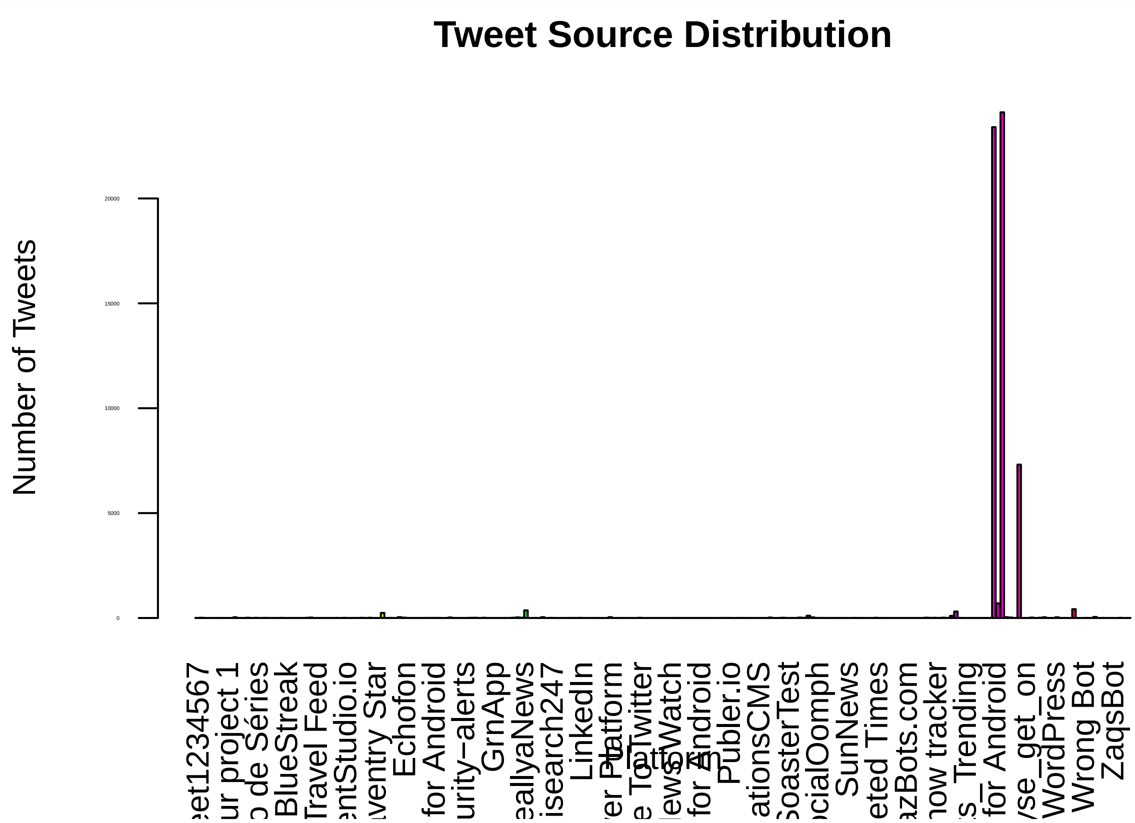

```
tweetsDF$date <- as.Date(tweetsDF$Created_At_Round)
tweetsDF$hour <- format(tweetsDF$Created_At_Round, "%H")

tweets_per_hour_per_date <- tweetsDF %>%
  group_by(date, hour) %>%
  summarise(tweet_count = n(), .groups = "drop")

tweets_per_hour_per_date$hour <- as.numeric(tweets_per_hour_per_date$hour)

print(tweets_per_hour_per_date)
```

```
## # A tibble: 73 x 3
##    date        hour tweet_count
```

```
##     <date>     <dbl>       <int>
##   1 2022-10-28     0           7
##   2 2022-10-28     1           9
##   3 2022-10-28     2          11
##   4 2022-10-28     3          13
##   5 2022-10-28     4          12
##   6 2022-10-28     5           8
##   7 2022-10-28     6           7
##   8 2022-10-28     7          10
##   9 2022-10-28     8           5
## 10 2022-10-28     9           7
## # i 63 more rows
```

Cleaning the StatucSource Column

```r
tweetsDF$statusSource_clean <- gsub("<.*?>", "", tweetsDF$statusSource)

statusCounts <- table(tweetsDF$statusSource_clean)

barplot(statusCounts,
        main = "Tweet Source Distribution",
        xlab = "Platform",
        ylab = "Number of Tweets",
        col = rainbow(length(statusCounts)),
        las = 2,
        cex.axis = 0.15)
```



#The first graph, a bar plot, illustrates the distribution of tweets across various source platforms. It reveals a highly skewed pattern, where a small number of dominant platforms, such as Twitter for iPhone and Twitter for Android, contribute the majority of tweets. Meanwhile, most other sources show minimal tweet counts. This emphasizes the significant role mainstream platforms play in driving Twitter activity, while less prominent sources have little impact on overall tweet volumes.

Compare Platforms over-time

```r
tweetsDF$Created_At_Round <- as.Date(tweetsDF$Created_At_Round)

platformTimeSeries <- table(tweetsDF$Created_At_Round, tweetsDF$statusSource_clean)

platformTimeSeriesDF <- as.data.frame(platformTimeSeries)

library(tidyr)
platformTimeSeriesReshaped <- platformTimeSeriesDF %>%
  pivot_wider(names_from = Var2, values_from = Freq, values_fill = list(Freq = 0))

platformTimeSeriesReshaped$Var1 <- as.Date(platformTimeSeriesReshaped$Var1)

all_dates <- seq(min(platformTimeSeriesReshaped$Var1), max(platformTimeSeriesReshaped$Var1), by = "day")
platformTimeSeriesReshaped <- merge(platformTimeSeriesReshaped, data.frame(Var1 = all_dates), by = "Var

library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```r
platformTimeSeriesLong <- melt(platformTimeSeriesReshaped, id.vars = "Var1", variable.name = "Platform"

library(ggplot2)
ggplot(platformTimeSeriesLong, aes(x = Var1, y = TweetCount, color = Platform)) +
  geom_line() +
  labs(x = "Date", y = "Number of Tweets", title = "Tweets by Platform Over Time") +
  theme_minimal() +
  theme(legend.title = element_blank())
```

#The second graph provides a detailed list of the various platforms and apps used as tweet sources. It highlights the extensive diversity of tools integrated with Twitter, including both popular and niche platforms. The presence of many low-contribution sources suggests that some are specialized tools or automated systems (bots) with limited activity. This diversity showcases Twitter's versatility in accommodating a wide range of users and applications, from casual users to businesses leveraging automated posting tools.

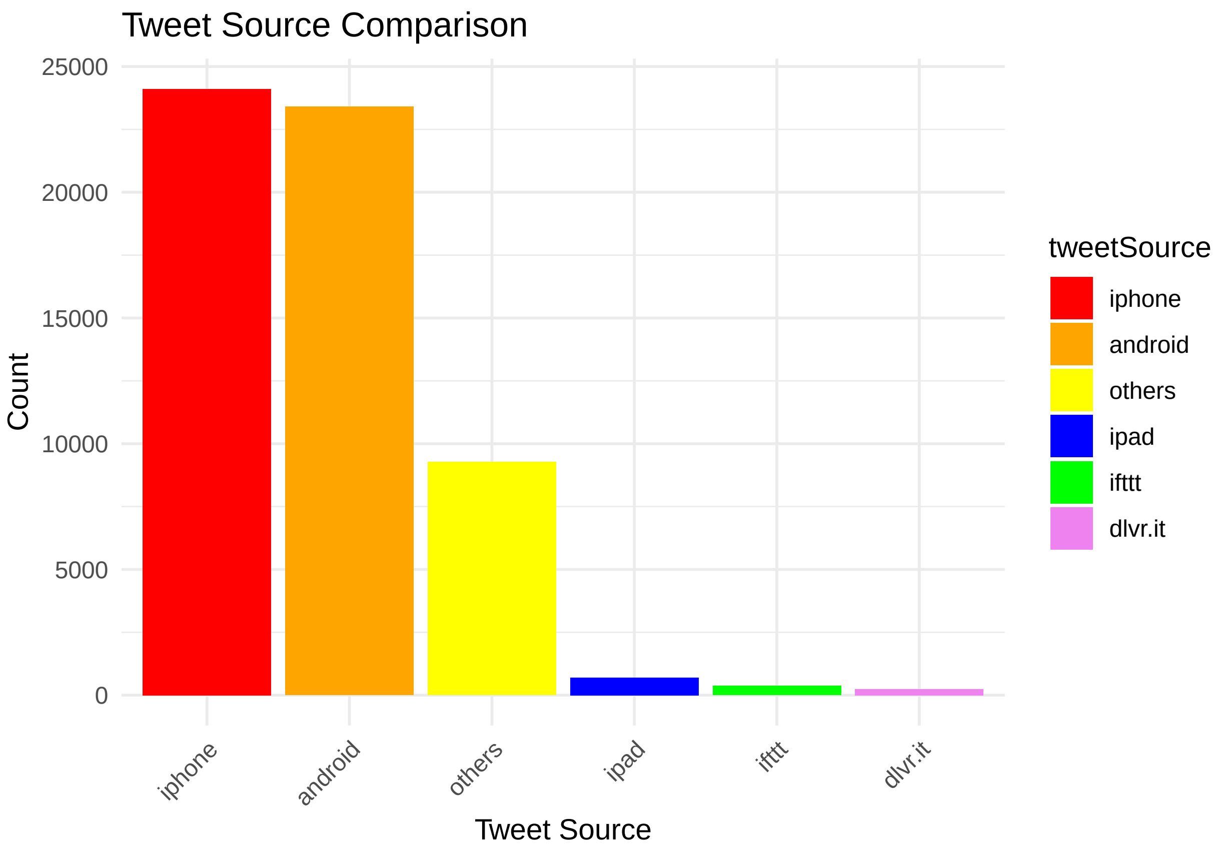
Chunk of Codes for Cleaning and Making an Graph about the TweetSource(Iphone, Android, others etc.)

```
library(ggplot2)
library(readr)
library(dplyr)

print(colnames(tweetsDF))
```

```
##  [1] "X"                    "screenName"          "text"
##  [4] "created"              "statusSource"        "Created_At_Round"
##  [7] "tweetSource"          "date"                "hour"
## [10] "statusSource_clean"
```

```
TweetSourceCounts <- tweetsDF %>%
  group_by(tweetSource) %>%
  summarize(Count = n()) %>%
  arrange(desc(Count))

TweetSourceCounts$tweetSource <- factor(TweetSourceCounts$tweetSource,
```

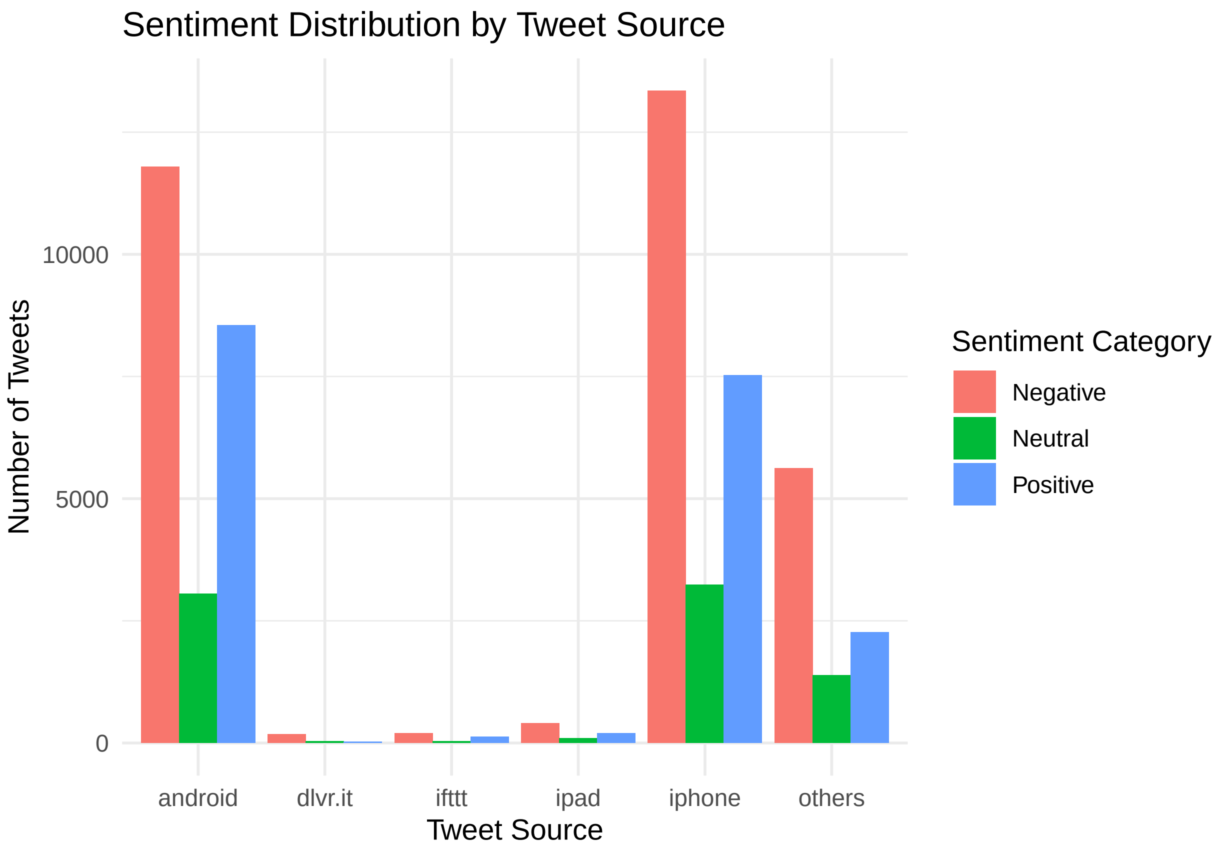
```
                                          levels = TweetSourceCounts$tweetSource)


ggplot(TweetSourceCounts, aes(x = reorder(tweetSource, -Count), y = Count, fill = tweetSource)) +
  geom_bar(stat = "identity") +
  labs(title = "Tweet Source Comparison",
       x = "Tweet Source",
       y = "Count") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_fill_manual(values = c("iphone" = "red", "android" = "orange", "others" = "yellow", "ipad" = "bl
```



The graph titled “Tweet Source Comparison” illustrates the distribution of tweets based on their source. The majority of tweets are posted using iPhone and Android, which dominate the chart with the highest counts. These two sources significantly outperform others, highlighting their widespread use among Twitter users. The third most common source is categorized as Others, though its count is notably lower compared to iPhone and Android. Meanwhile, platforms like iPad, ifttt, and dlvr.it contribute only a small fraction of tweets, indicating limited usage.

This distribution suggests that mobile devices, particularly iPhones and Android smartphones, are the primary tools for engaging on Twitter. The “Others” category likely represents a mix of niche or less common platforms. Automated tools like ifttt and dlvr.it are used sparingly, possibly for specific purposes such as scheduled or automated posts. Businesses and marketers looking to target Twitter users should prioritize strategies that cater to mobile users, particularly those on iPhone and Android devices, given their overwhelming share. Further analysis of the “Others” category might reveal additional insights about underutilized platforms or unique user behaviors.

```r
tweetsDF$sentiment <- get_sentiment(tweetsDF$text, method = "syuzhet")

tweetsDF <- tweetsDF %>%
  mutate(sentiment_category = case_when(
    sentiment > 0 ~ "Positive",
    sentiment == 0 ~ "Neutral",
    sentiment < 0 ~ "Negative"
  ))
sentiment_by_source <- tweetsDF %>%
  group_by(tweetSource, sentiment_category) %>%
  summarize(count = n(), .groups = 'drop')

ggplot(sentiment_by_source, aes(x = tweetSource, y = count, fill = sentiment_category)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(
    title = "Sentiment Distribution by Tweet Source",
    x = "Tweet Source",
    y = "Number of Tweets",
    fill = "Sentiment Category"
  ) +
  theme_minimal()
```



#

The grouped bar chart provides valuable insights into the sentiment distribution of tweets across different sources, such as iPhone, Android, and other platforms. By examining the chart, we can observe which sentiment is most dominant for each tweet source. For example, tweets from platforms like "iPhone" may have a higher proportion of Positive sentiment, while sources like "Android" could show a mix of Positive, Neutral, and Negative sentiments. This variation suggests that sentiment trends differ across platforms. Additionally, the chart reveals the volume of tweets per source, highlighting how certain platforms, like iPhone, generate more tweets compared to others like "dlvr.it" or "IFTTT". This discrepancy may indicate the popularity of certain sources. From a strategic standpoint, understanding these sentiment trends can be crucial for businesses or analysts, as positive sentiments could indicate more favorable user engagement, while negative sentiments from specific sources may signal user dissatisfaction or areas for improvement. Overall, the graph helps in recognizing patterns that can inform marketing strategies, customer engagement, and platform-related decision-making.