SentimentAnalysisProject

Andica, Benedicto, Cautivar

2024-12-07

Reading the tweetsDF.csv file

```
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(stringr)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.5
## v lubridate 1.9.4     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
```

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

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

```
library(wordcloud)
```

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

```
library(RColorBrewer)
library(lubridate)
```

```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>", "", t
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, ]

head(chunk_data)
```

```
##   X    screenName
## 1 1      whourj31
## 2 2       nnainot
## 3 3   febry_sri_M
## 4 4 telehuntwatch
## 5 5    Typing0824
## 6 6   niccijsmith
##
## 1         a soldier angry at the support fund consolation money for the bereaved family of the itaew
## 2                                                                nah this itaewon tragedy really has m
## 3                                                                                                    
## 4   translation seoul residents lay flowers at a makeshift memorial near the site of the crush in ita
## 5 the itaewon stampede incident really caught me off guard. makes me notice how important it is to kn
## 6      what to do about my child what to do about my child park ga-youngs mother, choi seon-mi, said
##            created
## 1 30/10/2022 23:59
## 2 30/10/2022 23:59
## 3 30/10/2022 23:59
## 4 30/10/2022 23:59
## 5 30/10/2022 23:59
```

```
## 6 30/10/2022 23:59
##                                                                                 statusSource
## 1                <a href="https://www.fs-poster.com/" rel="nofollow">FS_Poster_App</a>
## 2 <a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>
## 3 <a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>
## 4                           <a href="https://ruprop.live" rel="nofollow">telehunt</a>
## 5 <a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>
## 6   <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>
##   Created_At_Round tweetSource
## 1  31/10/2022 0:00      others
## 2  31/10/2022 0:00     android
## 3  31/10/2022 0:00     android
## 4  31/10/2022 0:00      others
## 5  31/10/2022 0:00     android
## 6  31/10/2022 0:00      iphone
```

```r
write.csv(chunk_data, "cleaned_tweets.csv", row.names = FALSE)

write.csv(tweetsDF, "processed_tweets.csv", row.names = FALSE)

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
```

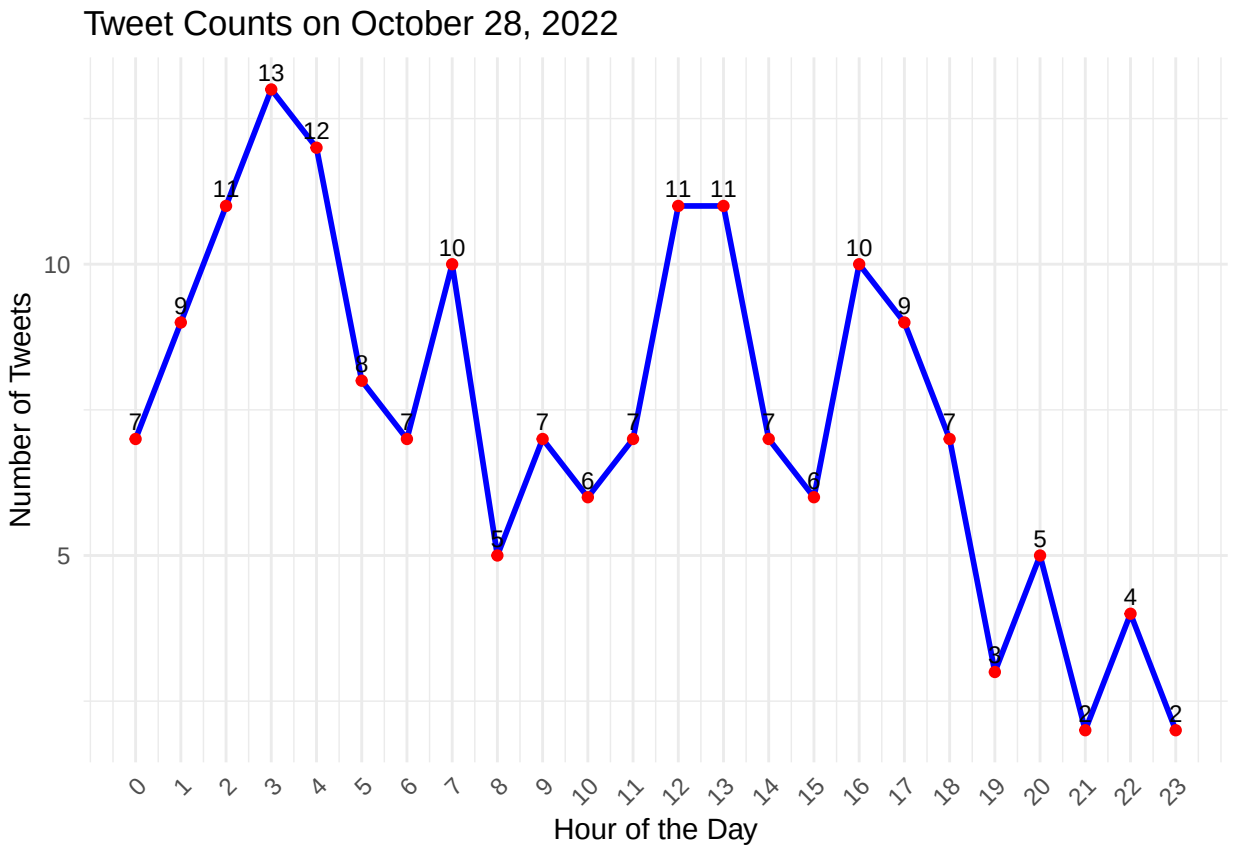
```
## Number of valid texts after stripping whitespace:  1000
```

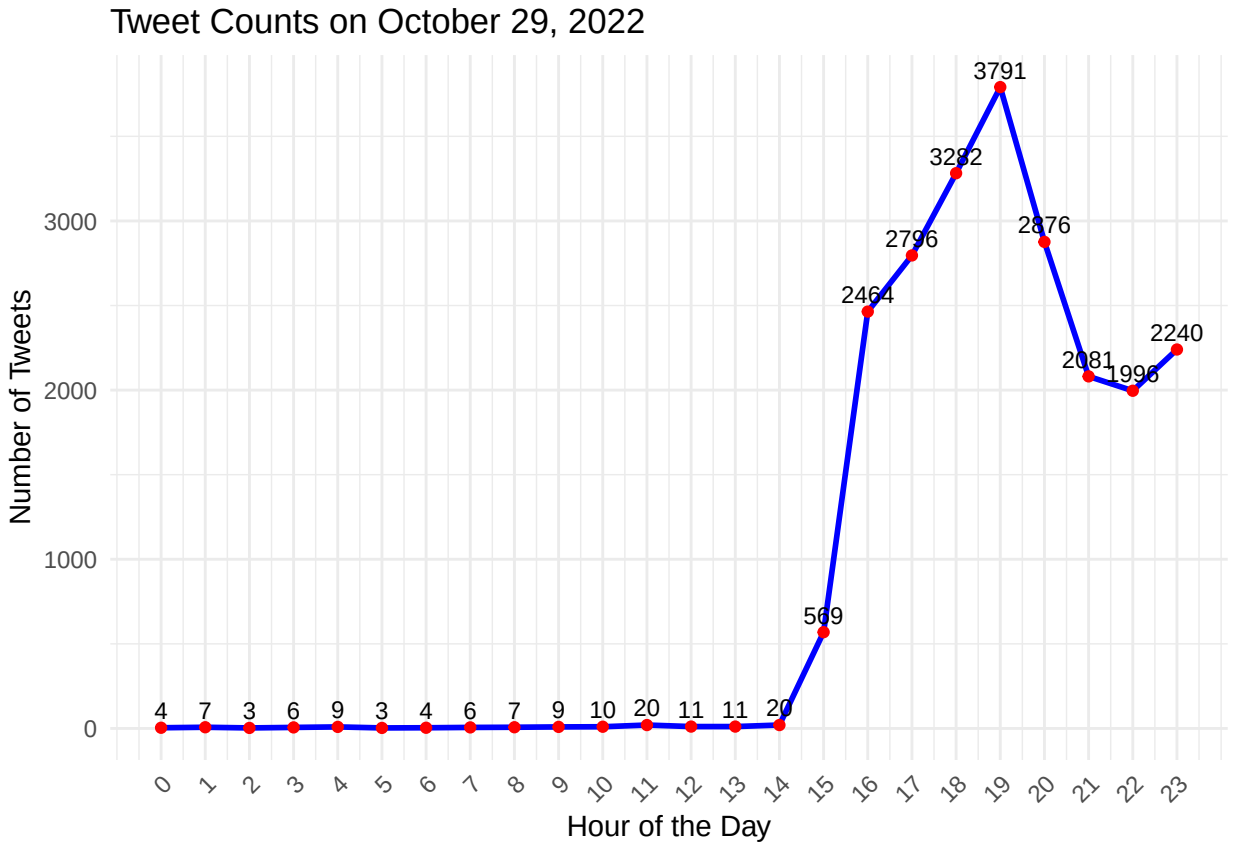
#NOTE: for the Wordcloud, it can change by changing also the range you want from the code above.

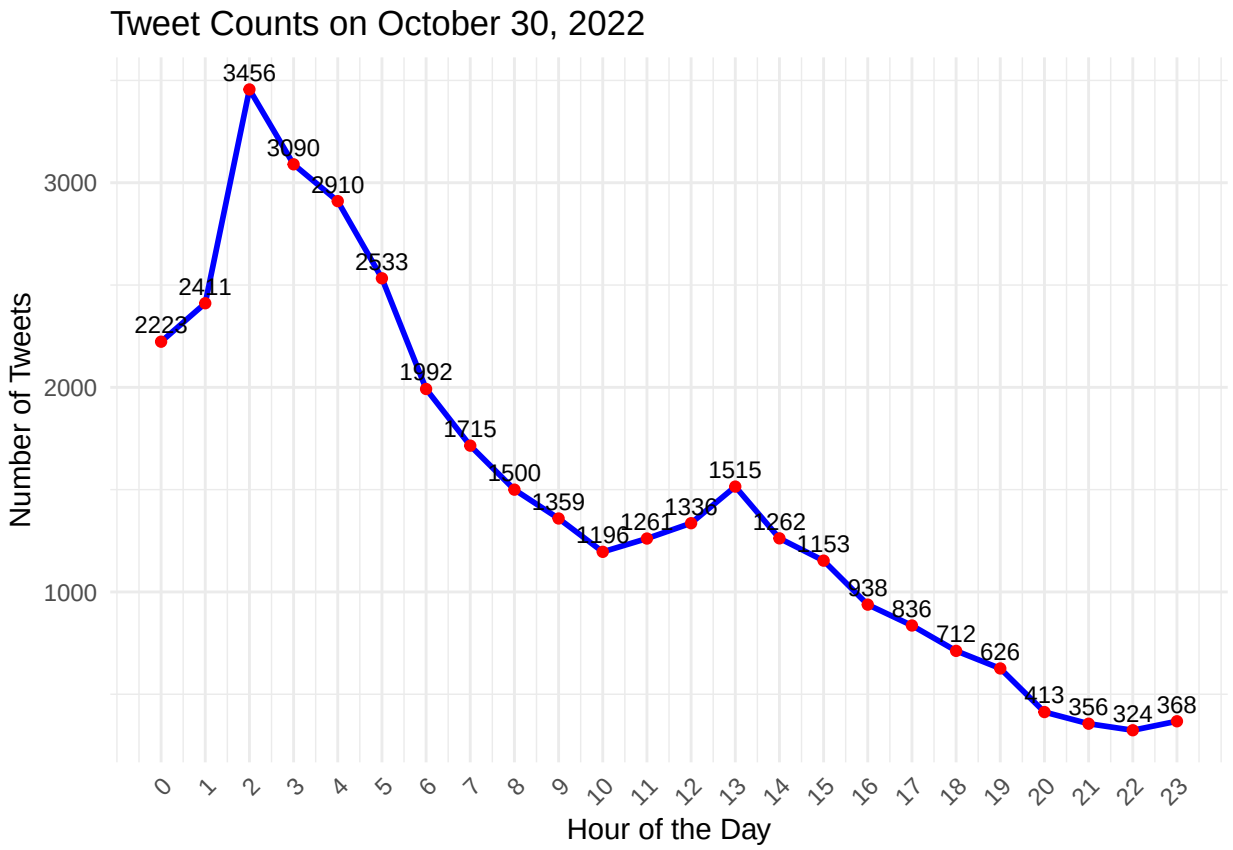
This word cloud reflects discussions about the Itaewon tragedy, with larger words like "Itaewon," "tragedy," and "happened" highlighting its focus. Terms such as "people," "victims," "Halloween," and "crowd" provide context, while words like "condolences," "pray," and "mourning" express grief and sympathy. It relates to the October 29, 2022, Itaewon crowd crush in Seoul, where overcrowding during Halloween festivities led to over 150 deaths and widespread mourning.

Cleaning Dates

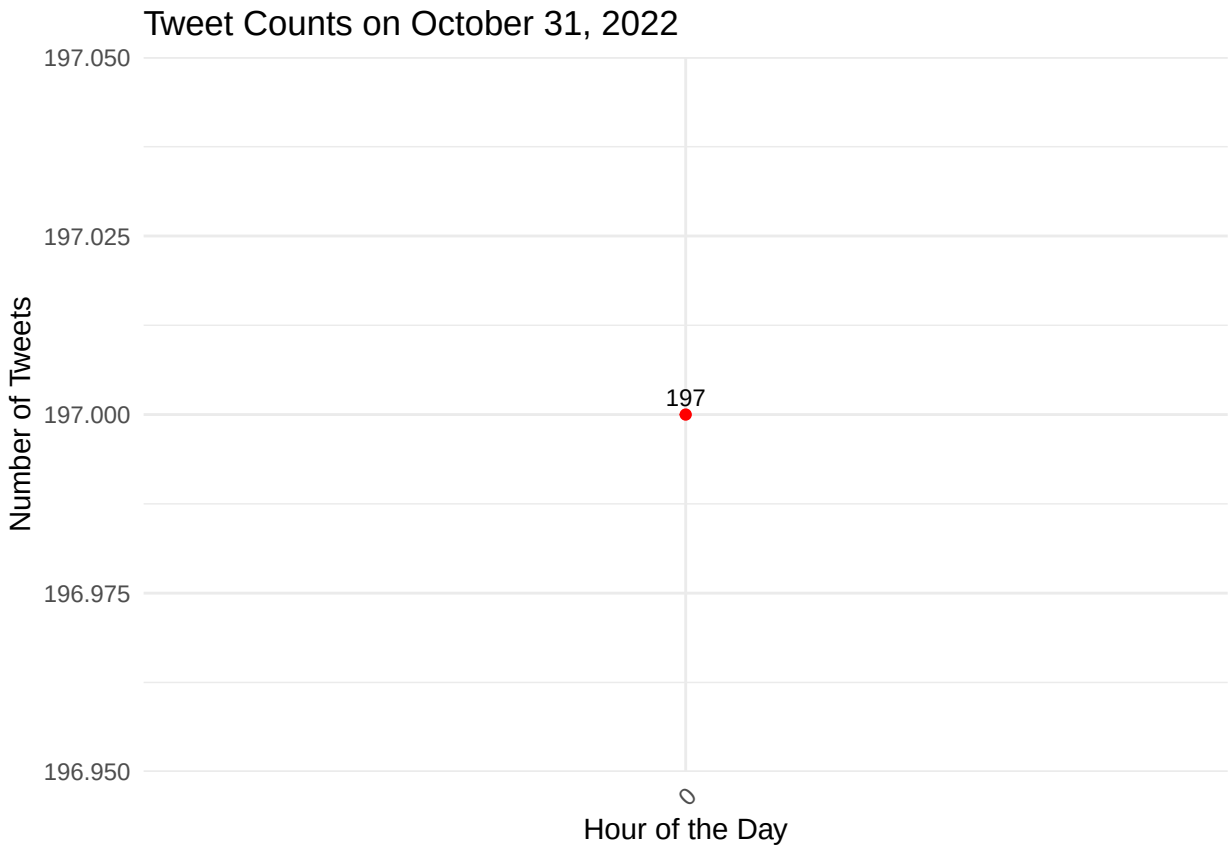
```
tweetsDF$Created_At_Round <- as.POSIXct(tweetsDF$Created_At_Round, format = "%d/%m/%Y %H:%M", tz = "UTC

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

hourly_tweets <- tweetsDF %>%
  group_by(date, hour) %>%
  summarise(tweet_count = n(), .groups = "drop") %>%
  mutate(hour = as.numeric(hour))

plots <- lapply(unique(hourly_tweets$date), function(current_date) {
  # Filter data for the current date
  date_data <- hourly_tweets %>%
    filter(date == current_date)

  ggplot(date_data, aes(x = hour, y = tweet_count)) +
    geom_line(color = "blue", linewidth = 1) +
    geom_point(color = "red") +
    geom_text(aes(label = tweet_count), vjust = -0.5, color = "black", size = 3) +
    scale_x_continuous(breaks = 0:23) +
    labs(
      title = paste("Tweet Counts on", format(current_date, "%B %d, %Y")),
      x = "Hour of the Day",
      y = "Number of Tweets"
    ) +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1))
})

for(i in seq_along(plots)) {
  print(plots[[i]])
}
```


Tweet Counts on October 28, 2022
Number of Tweets
Hour of the Day
10
5
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
7 9 11 13 12 8 7 10 5 7 6 7 11 11 7 6 10 9 7 3 5 2 4 2


Tweet Counts on October 29, 2022
Number of Tweets
Hour of the Day
3000
2000
1000
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
4 7 3 6 9 3 4 6 7 9 10 20 11 11 20 569 2464 2796 3282 3791 2876 2081 1996 2240




```
## `geom_line()`: Each group consists of only one observation.
## i Do you need to adjust the group aesthetic?
```




```
summary_per_date <- tweetsDF %>%
  group_by(date) %>%
  summarise(
    total_tweets = n(),
    unique_hours = n_distinct(format(Created_At_Round, "%H"))
  )
print(summary_per_date)
```

```
## # A tibble: 4 x 3
##   date       total_tweets unique_hours
##   <date>            <int>        <int>
## 1 2022-10-28          179           24
## 2 2022-10-29        22225           24
## 3 2022-10-30        35485           24
## 4 2022-10-31          197            1
```

1. Graph for October 28, 2022:

#This graph displays a more detailed distribution of tweets across 24 hours. Tweet activity fluctuates throughout the day, with peaks at hours like 2 AM, 3 AM, and 10 PM, where the count reaches 13 tweets, 12 tweets, and 11 tweets respectively. There are noticeable valleys during other hours, such as 6 AM and 9 PM. #Insight: The variability suggests that tweet activity on this day may align with specific events or user engagement patterns, such as late-night or evening activity. Peaks likely coincide with events or discussions of interest at those times.

2. Graph for October 29, 2022:

#This graph illustrates a dramatic spike in tweet counts during the late afternoon (4 PM to 7 PM), where the count surges to a maximum of 3,791 tweets at 7 PM. The activity is minimal for the rest of the day, with a slight recovery around 10 PM (2,240 tweets). #Insight: The sharp rise and high volume of tweets suggest a major event or trending topic occurred in the late afternoon and early evening. This pattern contrasts with the otherwise low activity during the rest of the day.

3. Graph for October 30, 2022:

This graph shows tweet counts across the hours of October 30, 2022. The x-axis represents the hours of the day (0 to 23), while the y-axis shows the number of tweets.

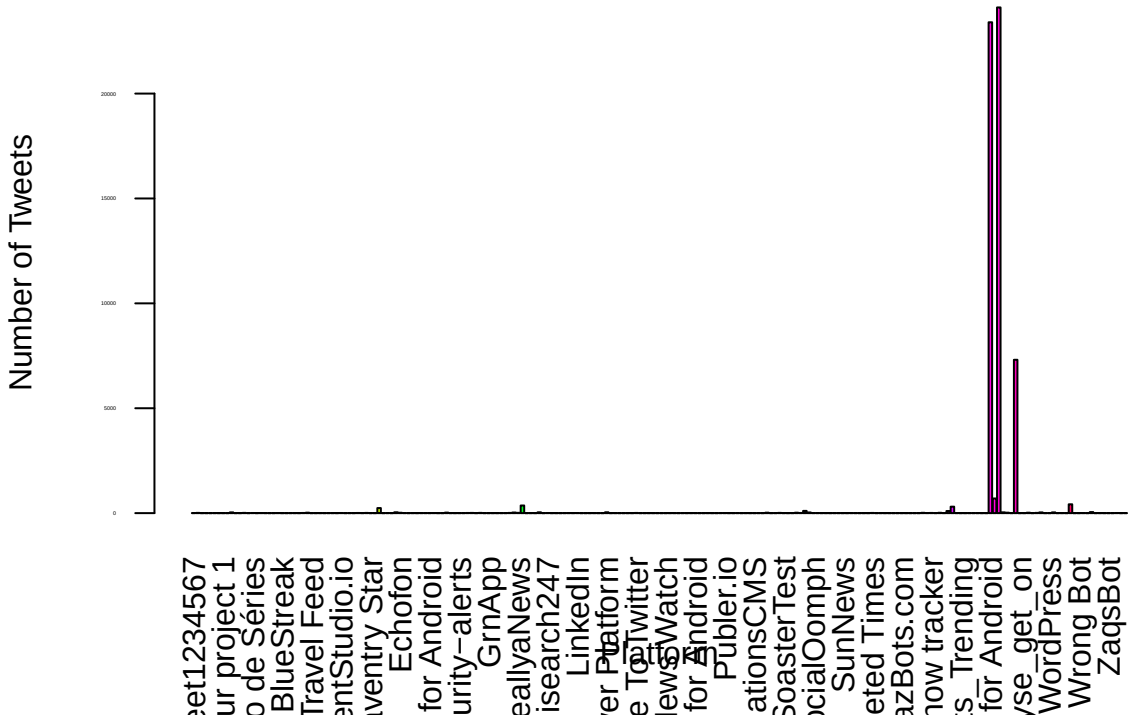
#The data reveals a peak in activity at 2 AM, with 3,456 tweets, followed by a sharp decline by 4 AM. From there, tweet counts gradually decrease until reaching a low of 324 tweets at 10 PM. There is some fluctuation, with a slight increase in the late morning and early afternoon, peaking again at 1 PM with 1,515 tweets, before continuing the downward trend toward the evening. #An insight from this graph is that tweet activity is highest during early morning hours, possibly indicating late-night discussions or time zone effects. The decrease in activity throughout the day suggests that users tweet less in the evening. This pattern could be useful for scheduling content to maximize engagement.

4.Graph for October 31, 2022:

#This graph shows only one data point, indicating a total of 197 tweets recorded at a specific hour. The flat, singular point suggests a lack of detailed hourly data or minimal tweet activity spread across the day. #Insight: The activity on this day is sparse or represents incomplete data, which makes it hard to draw meaningful conclusions.

Cleaning the StatucSource Column

```
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 bar plot graph, 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.

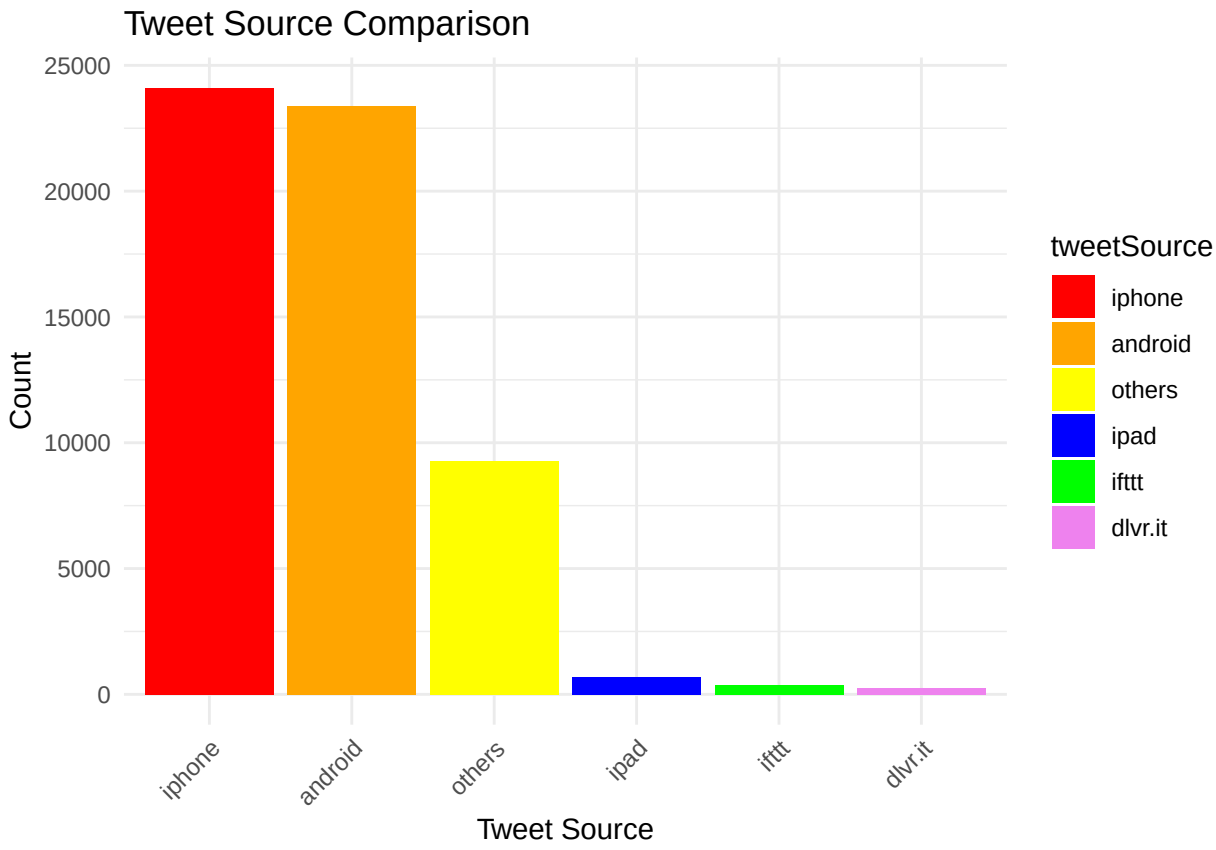
Chunk of Codes for Cleaning and Making an Graph about the TweetSource(Iphone, Android, others etc.)

```
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
```

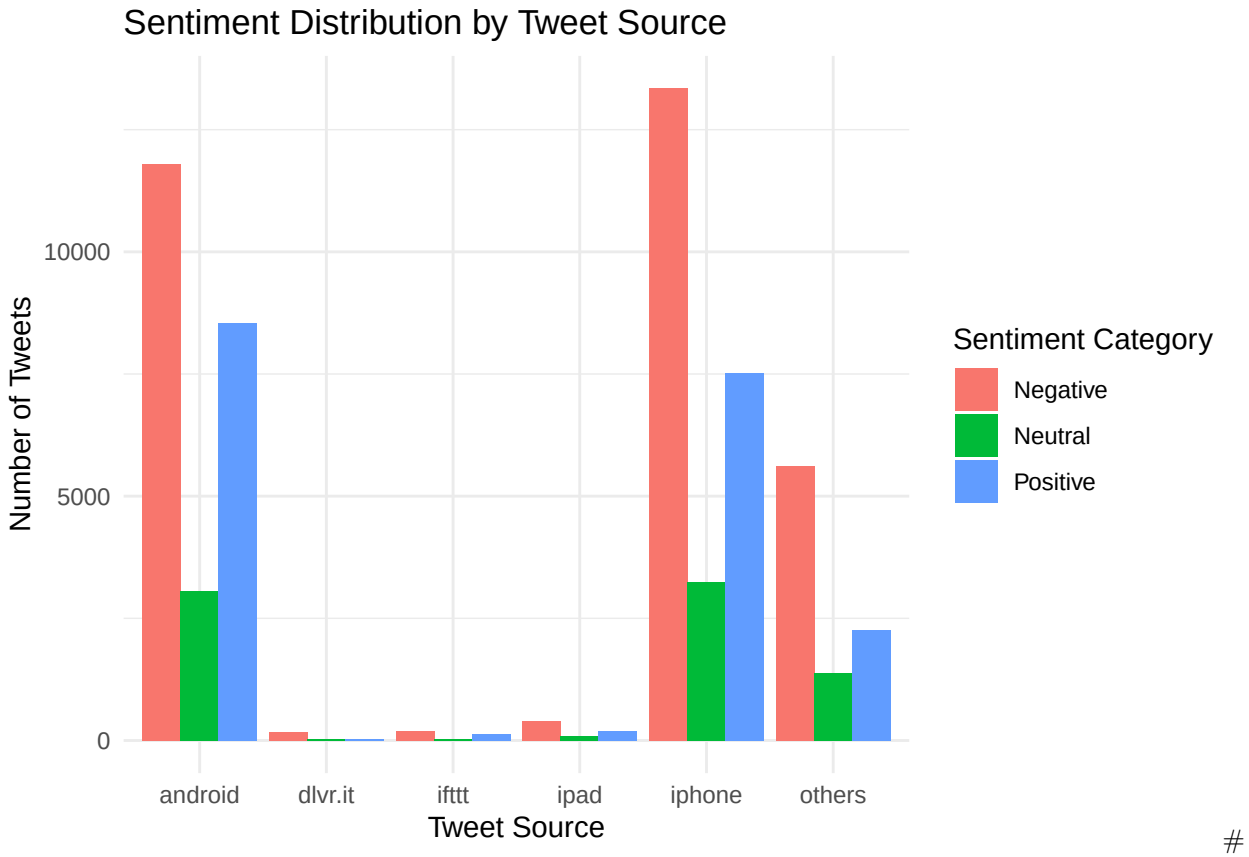


This bar chart compares the number of tweets across different sources. The x-axis represents the tweet sources (e.g., iPhone, Android, others), and the y-axis shows the count of tweets.

#The iPhone and Android are the most popular sources, with iPhone slightly leading in tweet count, followed closely by Android. The "others" category is the third-largest contributor but significantly lower than the first two. Other sources, such as iPad, IFTTT, and dlvr.it, have minimal tweet counts. #An insight from this chart is that most tweets are generated from mobile devices, particularly iPhones and Androids, highlighting their dominance as platforms for social media engagement. Other sources contribute relatively little to the overall tweet volume. This could reflect platform preferences among users or device availability.

```
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()
```



#

This bar chart shows the distribution of tweet sentiment (negative, neutral, and positive) based on the source of tweets. The x-axis represents the tweet sources (e.g., Android, iPhone, others), while the y-axis indicates the number of tweets. #From the chart, the majority of tweets are from Android and iPhone devices. For Android, there are more negative tweets compared to positive and neutral ones, with positive tweets being the second most frequent. Similarly, iPhone tweets have a high number of negative sentiments, followed by positive and neutral sentiments. Tweets from other sources (e.g., "others") follow a similar trend with fewer overall counts compared to Android and iPhone. #An insight here is that both Android and iPhone are the dominant platforms for tweeting, with negative sentiments being the most common category for both. This could reflect general user sentiment trends, but it might also indicate biases in how users from these platforms engage online.