Sentiment Analysis Project

Noblezada, Camasa, Cabia

2024-12-13

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.1
## v purrr     1.0.2
## -- 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(tidytext)
library(dplyr)
library(stringr)
library(ggplot2)
library(sentimentr)
library(lubridate)
```

```
# Load the dataset
tweetsDF <- read_csv("/cloud/project/ProjectDS/tweetsDF.csv")
```

```
## New names:
## Rows: 58086 Columns: 7
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (4): screenName, text, statusSource, tweetSource dbl (1): ...1 dttm (2):
## created, Created_At_Round
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

Cleaning the tweet text

```
tweetsDF <- tweetsDF %>%
  mutate(
    text = text %>%
      iconv(from = "UTF-8", to = "ASCII//TRANSLIT", sub = "") %>% # Remove non-ASCII characters
      tolower() %>% # Convert to lowercase
      str_remove_all("https\\S+") %>% # Remove URLs
      str_remove_all("[#\\n]") %>% # Remove hashtags and newlines
      str_remove_all("[@?]\\S+") %>% # Remove mentions
      str_remove_all("\\?") %>% # Remove question marks
      str_remove_all("\\b\\d{2}\\.\\d{2}\\.\\d{4}\\b") %>% # Remove dates in dd.mm.yyyy format
      str_remove_all("<a href=httptwitter.comdownloadiphone rel=nofollow>twitter for iphone<a>") %>% # 
      str_remove_all("<a href=([^>]*?) rel=nofollow>([^<]*?)<a>") %>%
      str_remove_all("<a href=httptwitter.comdownloadandroid rel=nofollow>twitter for android<a>") %>%
      str_remove_all("<a href= rel=nofollow>twitter web app<a>") %>%
      str_remove_all("30102022") %>% # Remove specific date
      str_squish() # Remove extra whitespace
  )
tweetsDF <- tweetsDF %>%
  mutate(date = ymd_hms(created)) %>%
  mutate(hour = hour(date))
```

```
## Warning: There was 1 warning in `mutate()`.
## i In argument: `date = ymd_hms(created)`.
## Caused by warning:
## !  2 failed to parse.
```

```
print(tweetsDF)
```

```
## # A tibble: 58,086 x 9
##     ...1 screenName   text  created             statusSource Created_At_Round
##    <dbl> <chr>        <chr> <dttm>              <chr>        <dttm>
##  1     1 whourj31     a so~ 2022-10-30 23:59:43 "<a href=\"~ 2022-10-31 00:00:00
##  2     2 nnainot      nah ~ 2022-10-30 23:59:32 "<a href=\"~ 2022-10-31 00:00:00
##  3     3 febry_sri_M  pray~ 2022-10-30 23:59:31 "<a href=\"~ 2022-10-31 00:00:00
##  4     4 telehuntwat~ tran~ 2022-10-30 23:59:28 "<a href=\"~ 2022-10-31 00:00:00
##  5     5 Typing0824   the ~ 2022-10-30 23:59:20 "<a href=\"~ 2022-10-31 00:00:00
##  6     6 niccijsmith  what~ 2022-10-30 23:59:04 "<a href=\"~ 2022-10-31 00:00:00
##  7     7 502SPIDEY    can'~ 2022-10-30 23:58:56 "<a href=\"~ 2022-10-31 00:00:00
##  8     8 maeannesala~ pray~ 2022-10-30 23:58:45 "<a href=\"~ 2022-10-31 00:00:00
##  9     9 bigvirtue1   bigv~ 2022-10-30 23:58:37 "<a href=\"~ 2022-10-31 00:00:00
## 10    10 ashxxy       ther~ 2022-10-30 23:58:31 "<a href=\"~ 2022-10-31 00:00:00
## # i 58,076 more rows
## # i 3 more variables: tweetSource <chr>, date <dttm>, hour <int>
```

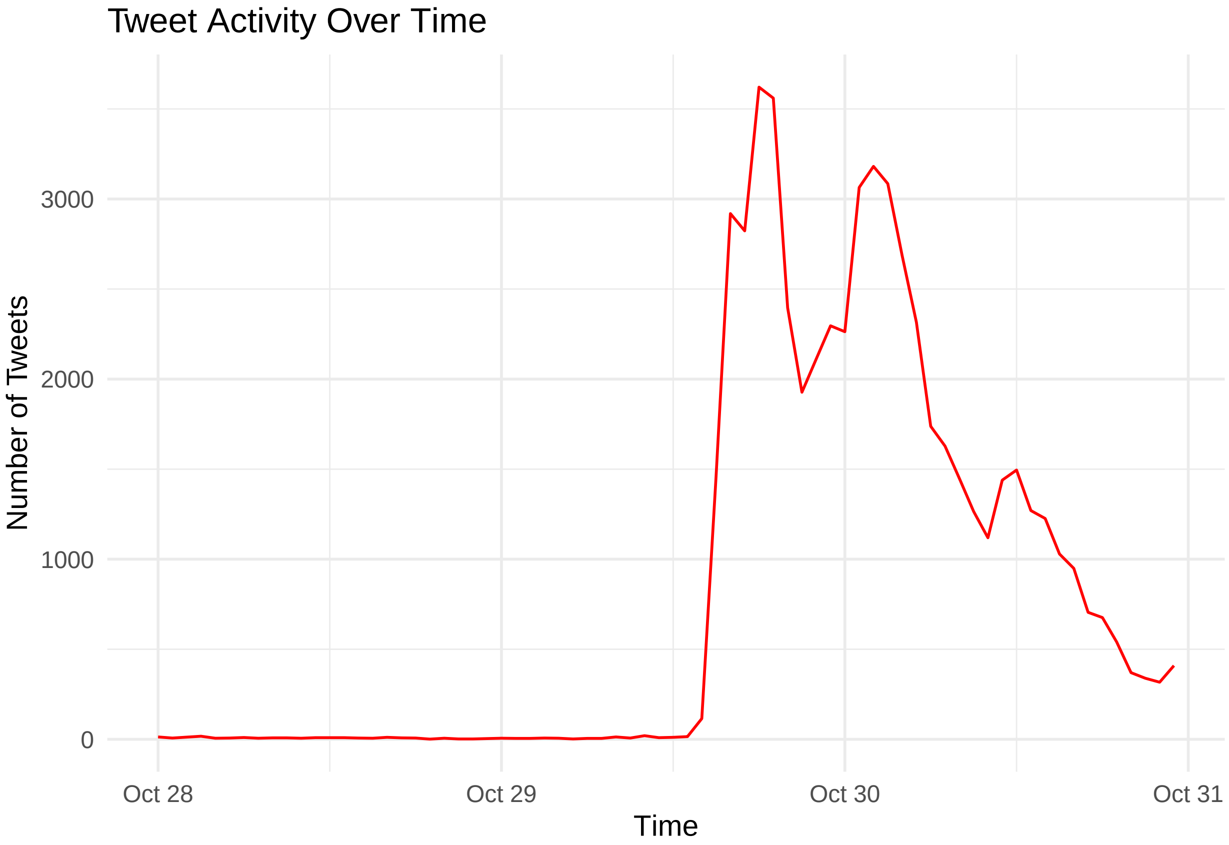
```
TweetsDF <- head(tweetsDF, 1000)

write.csv(TweetsDF, "CleanTweets.csv", row.names = FALSE)
```

```
colnames(tweetsDF)
```

```
## [1] "...1"             "screenName"       "text"             "created"
## [5] "statusSource"     "Created_At_Round" "tweetSource"      "date"
## [9] "hour"
```

Trend Analysis

```
# Load and preprocess the dataset
tweets_df <- read.csv("/cloud/project/ProjectDS/tweetsDF.csv")
tweets_df$created <- ymd_hms(tweets_df$created)

# Group tweets by hour and count
tweets_per_time <- tweets_df %>%
  mutate(hour = floor_date(created, "hour")) %>%
  count(hour)

# Plot the trend analysis
ggplot(tweets_per_time, aes(x = hour, y = n)) +
  geom_line(color = "red") +
  labs(
    title = "Tweet Activity Over Time",
    x = "Time",
    y = "Number of Tweets"
  ) +
  theme_minimal()
```



# Observations: # - Tweets were minimal before midnight on October 29. # - A significant spike occurred after 10:30 PM on October 29, coinciding with the Itaewon tragedy. # - Activity peaked in the early morning of October 30, declined through the morning, and rose again in the evening.

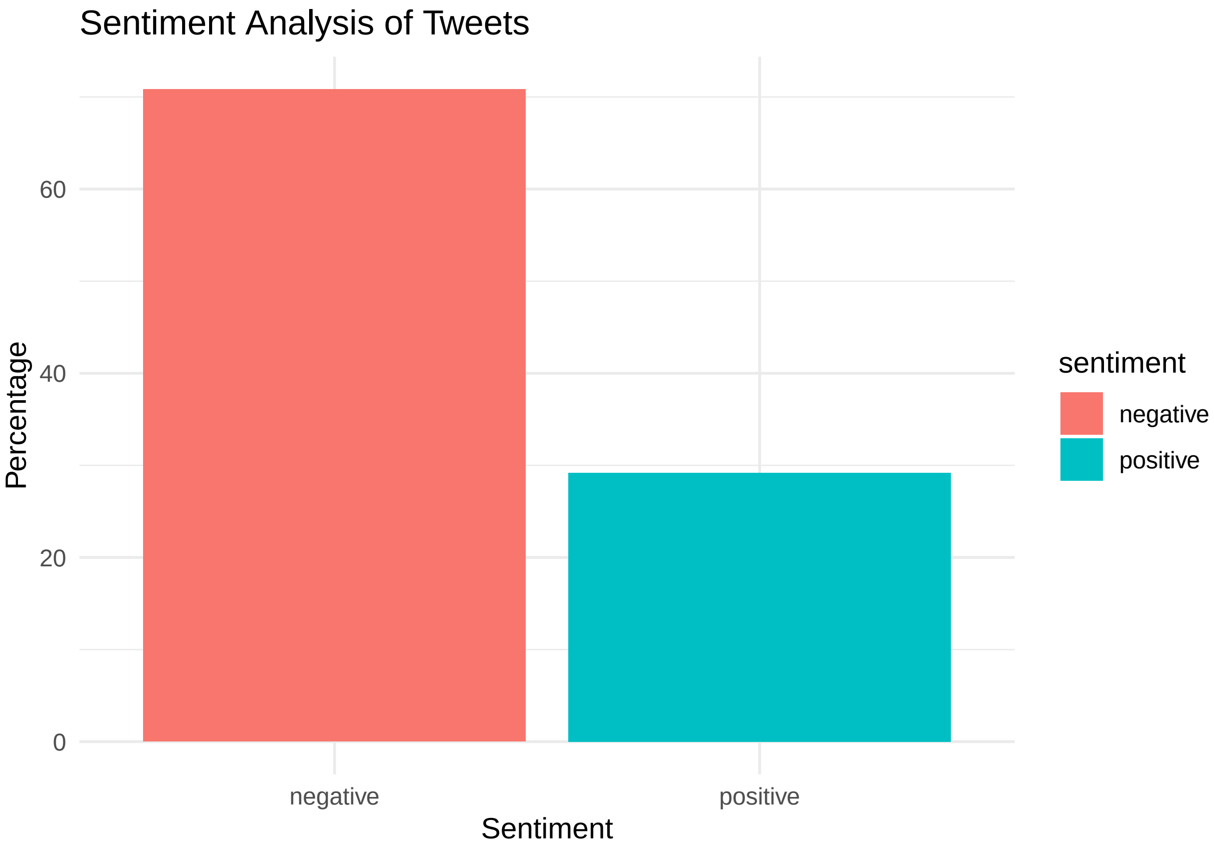
SENTIMENT ANALYSIS

```
# Perform sentiment analysis on tweets to examine emotional responses.

# Load sentiment lexicon and preprocess tweets
sentiments <- get_sentiments("bing")
tweet_tokens <- tweets_df %>%
  unnest_tokens(word, text) %>%
  inner_join(sentiments, by = "word")
```

```r
# Count sentiments and calculate percentages
sentiment_counts <- tweet_tokens %>%
  count(sentiment) %>%
  mutate(percentage = n / sum(n) * 100)

# Print sentiment counts
print(sentiment_counts)
```

```
##   sentiment     n percentage
## 1  negative 49285    70.8301
## 2  positive 20297    29.1699
```

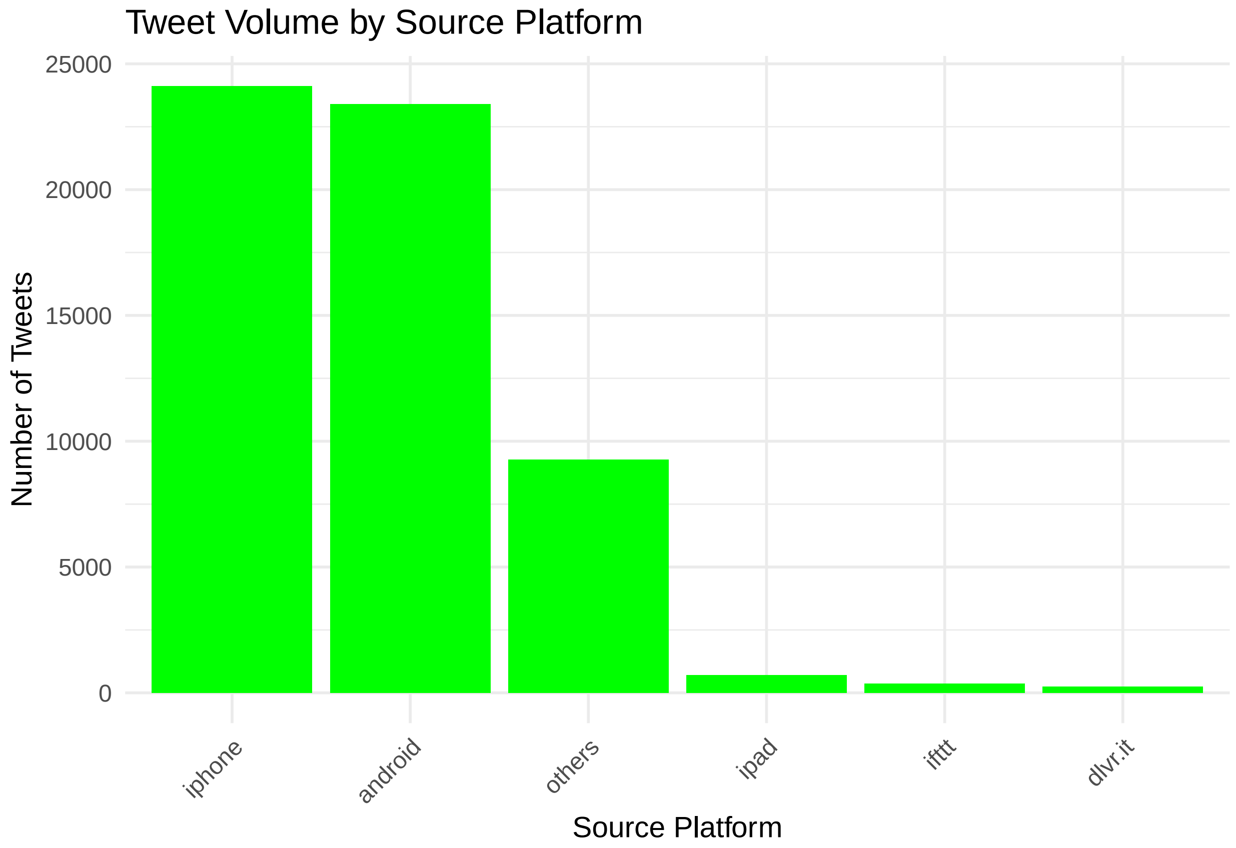
```r
# Plot sentiment analysis
ggplot(sentiment_counts, aes(x = sentiment, y = percentage, fill = sentiment)) +
  geom_bar(stat = "identity") +
  labs(
    title = "Sentiment Analysis of Tweets",
    x = "Sentiment",
    y = "Percentage"
  ) +
  theme_minimal()
```



#

Observations: # - The plot shows the proportion of positive and negative sentiments in the tweets. # - Helps to understand the overall emotional tone of the conversations.

```r
# Analyze the dataset to determine the distribution of tweets across different source platforms.
# Group tweets by source and calculate the count for each source
tweet_source_analysis <- tweets_df %>%
  group_by(tweetSource) %>%
  summarise(Number_of_Tweets = n()) %>%
  arrange(desc(Number_of_Tweets))

# Plot the number of tweets by source
ggplot(tweet_source_analysis, aes(x = reorder(tweetSource, -Number_of_Tweets), y = Number_of_Tweets)) +
  geom_bar(stat = "identity", fill = "green") +
  labs(
    title = "Tweet Volume by Source Platform",
    x = "Source Platform",
    y = "Number of Tweets"
  ) +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



# Observations: # - The bar graph illustrates the distribution of tweets by source platform. # - The x-axis displays various platforms (e.g., iPhone, Android, others), while the y-axis represents the total tweet count from each source. # - iPhone and Android dominate as the most frequently used platforms for engaging with the content.