Sentiment-Analysis(Corvera, Paclibar, Sabarillo)"

Rotciv Corvera, Jhon Albert Paclibar, Kirk Axl Dend Sabarillo

2024-12-10

```
#installing packages
install.packages("lubridate")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
library(lubridate)
```

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

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

```
install.packages("ggplot2")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
library(ggplot2)
install.packages("tidyverse")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
install.packages("tidytext")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
## (as 'lib' is unspecified)
```

```
library(tidytext)
library(tidyverse)
```

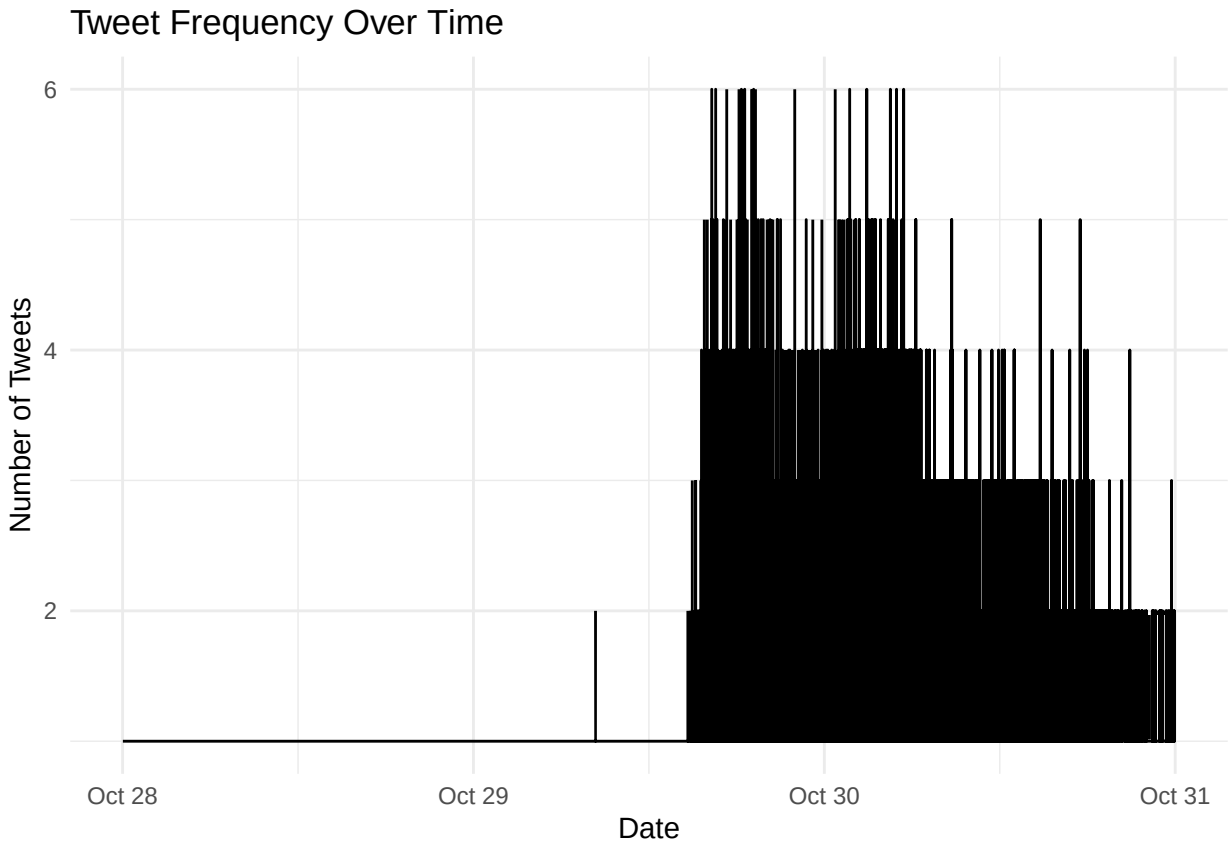
```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr   1.1.4     v stringr 1.5.1
## v forcats 1.0.0     v tibble  3.2.1
## v purrr   1.0.2     v tidyr   1.3.1
## v readr   2.1.5
```

```
## -- 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
```

```
# Read the CSV file
tweet_data <- read.csv("tweetsDF.csv", stringsAsFactors = FALSE)
```

```r
# Convert the created column to a datetime format
tweet_data$created <- ymd_hms(tweet_data$created)
```

```r
# Create a time series plot
ggplot(tweet_data, aes(x = created)) +
  geom_line(stat = "count") +
  labs(x = "Date", y = "Number of Tweets", title = "Tweet Frequency Over Time") +
  theme_minimal()
```




```r
#By analyzing the time series data, we can identify the specific
#date when user activity on the platform peaked,
#revealing when people were most active.
#Here, it shows that most people tweet during October 30.
```

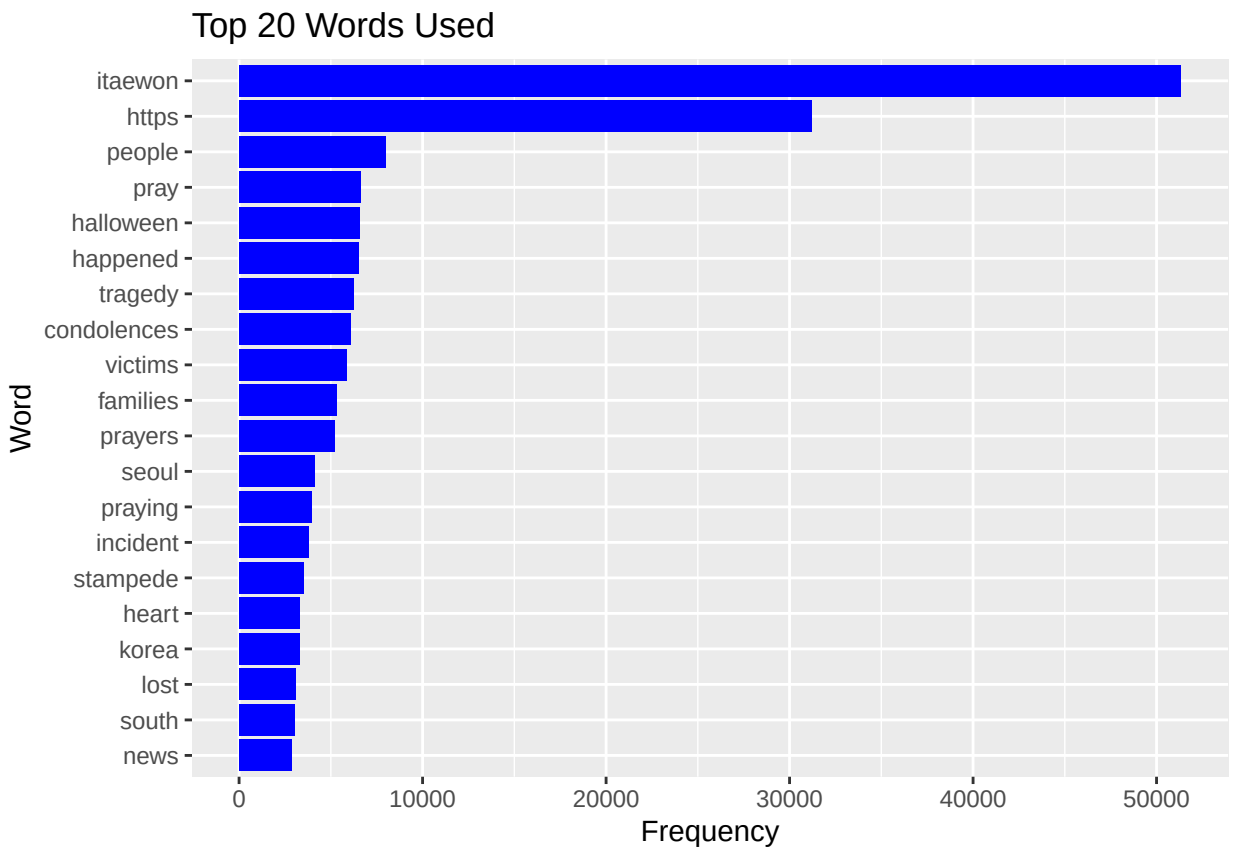
```r
# Create a tokenized data frame
tokens <- tweet_data %>%
  unnest_tokens(word, text)
```

```r
# Remove stop words
tokens <- tokens %>%
  anti_join(stop_words)
```

```
## Joining with `by = join_by(word)`
```

```r
# Count the frequency of each word
word_counts <- tokens %>%
  count(word, sort = TRUE)
```

```r
# Visualize the top 20 words
word_counts %>%
  head(20) %>%
  ggplot(aes(x = reorder(word, n), y = n)) +
  geom_col(fill = "blue") +
  labs(x = "Word", y = "Frequency", title = "Top 20 Words Used") +
  coord_flip()
```



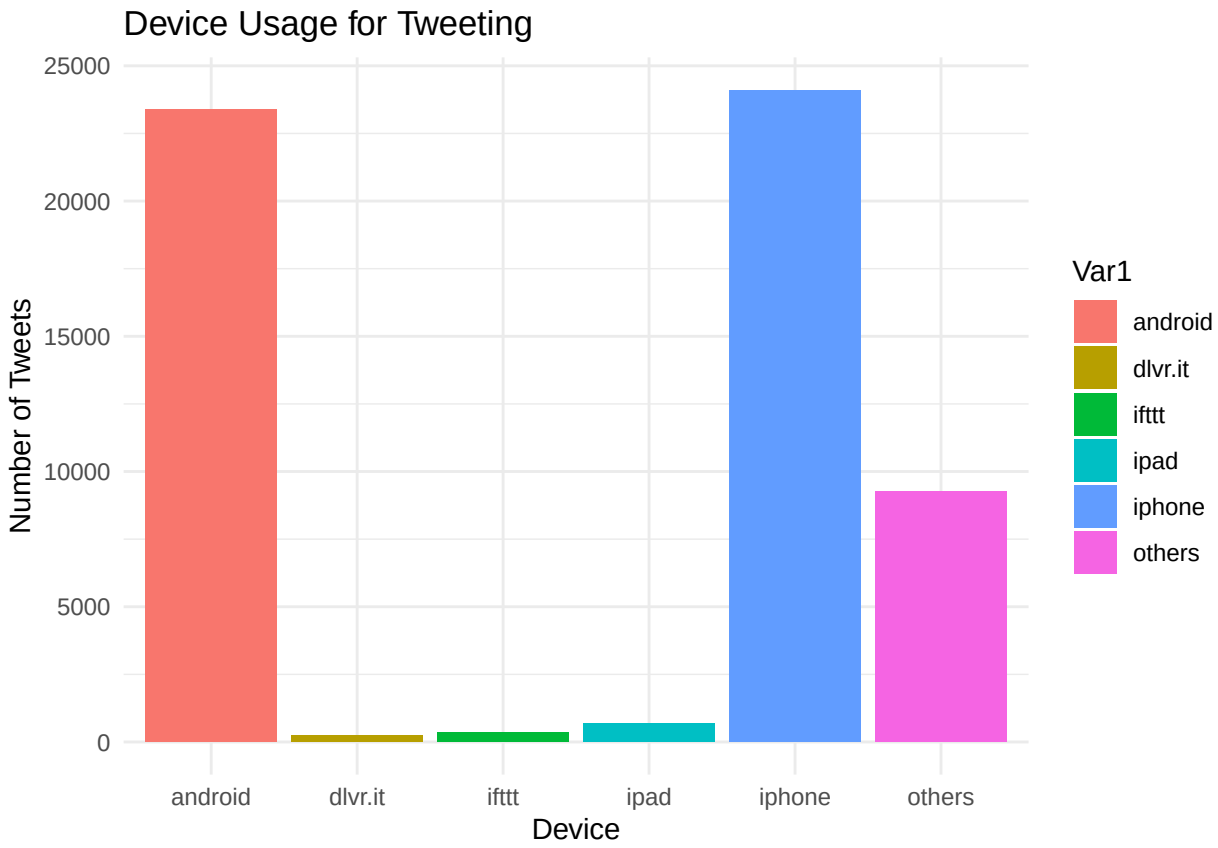

```r
#Using the bar chart, we can pinpoint the most frequently
#used word in the tweets during that period.
#This allows us to infer and identify the recent event that captured people's attention.

#Here, the data reveals that the most tweeted word was
#"Itaewon," a district located in Seoul, South Korea.
```

```r
# Count the frequency of each device
device_counts <- table(tweet_data$tweetSource)
```

```r
# Convert the table to a data frame for ggplot2
device_df <- as.data.frame(device_counts)
```

```r
# Create a bar plot using ggplot2
ggplot(device_df, aes(x = Var1, y = Freq, fill = Var1)) +
  geom_bar(stat = "identity") +
  labs(x = "Device", y = "Number of Tweets", title = "Device Usage for Tweeting") +
  theme_minimal()
```




```
#The graph illustrates the devices most commonly used for posting tweets.
#The results indicate that the majority of users tweeted from iPhones,
#with Android devices coming in second, showing only a small gap between them.
```