Sentiment Analysis Project

Cahutay, Camarista, Josue

2024-12-10

- BSIT-2A

```
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)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.2

## Warning: package 'tidyr' was built under R version 4.4.2

## Warning: package 'readr' was built under R version 4.4.2

## Warning: package 'lubridate' was built under R version 4.4.2

## -- 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
```

```r
library(syuzhet)
```

```
## Warning: package 'syuzhet' was built under R version 4.4.2
```

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

full_df <- tweetsDF
```

Codechunks for modifying/cleaning the dataset

```r
# Codes for cleaning the dataset
clean <- full_df %>%
  mutate(text = str_to_lower(text),
         text = str_remove_all(text, "http\\S+"),
         text = str_remove_all(text, "@\\w+"),
         text = str_remove_all(text, "#\\w+"),
         text = str_remove_all(text, "\\d+"),
         text = str_remove_all(text, "[^\\w\\s]"),
         text = str_squish(text),
         sentiment = get_sentiment(text, method = "bing"))

# Modified the dataset, added a date and hour column converted from the "created" column of tweetsDF
clean <- clean %>%
  mutate(date = ymd_hms(created)) %>%
  mutate(hour = hour(date))
```

END OF CLEANING

Preparing data for Trend Analysis

- For the trend analysis, we focused on the frequency of tweets posted per hour regarding the Itaewon incident to observe how tweet activity changed over time. By analyzing tweet counts at different hours, we aimed to identify when significant updates or news related to the incident triggered spikes in user engagement. This data helps us understand the timing of public reactions and responses, providing insights into how information spread and when people were most active in discussing the event.

```r
# Summarize tweet frequency by hour and group them by days
hourly_summary <- clean %>%
  group_by(day = as.Date(date), hour) %>%
  summarise(tweet_count = n(), .groups = "drop")

# Split data into separate subsets for plotting by day
day1_tweets <- hourly_summary %>% filter(day == unique(day)[1])
day2_tweets <- hourly_summary %>% filter(day == unique(day)[2])
day3_tweets <- hourly_summary %>% filter(day == unique(day)[3])
```

— Codes for plotting trend analysis —

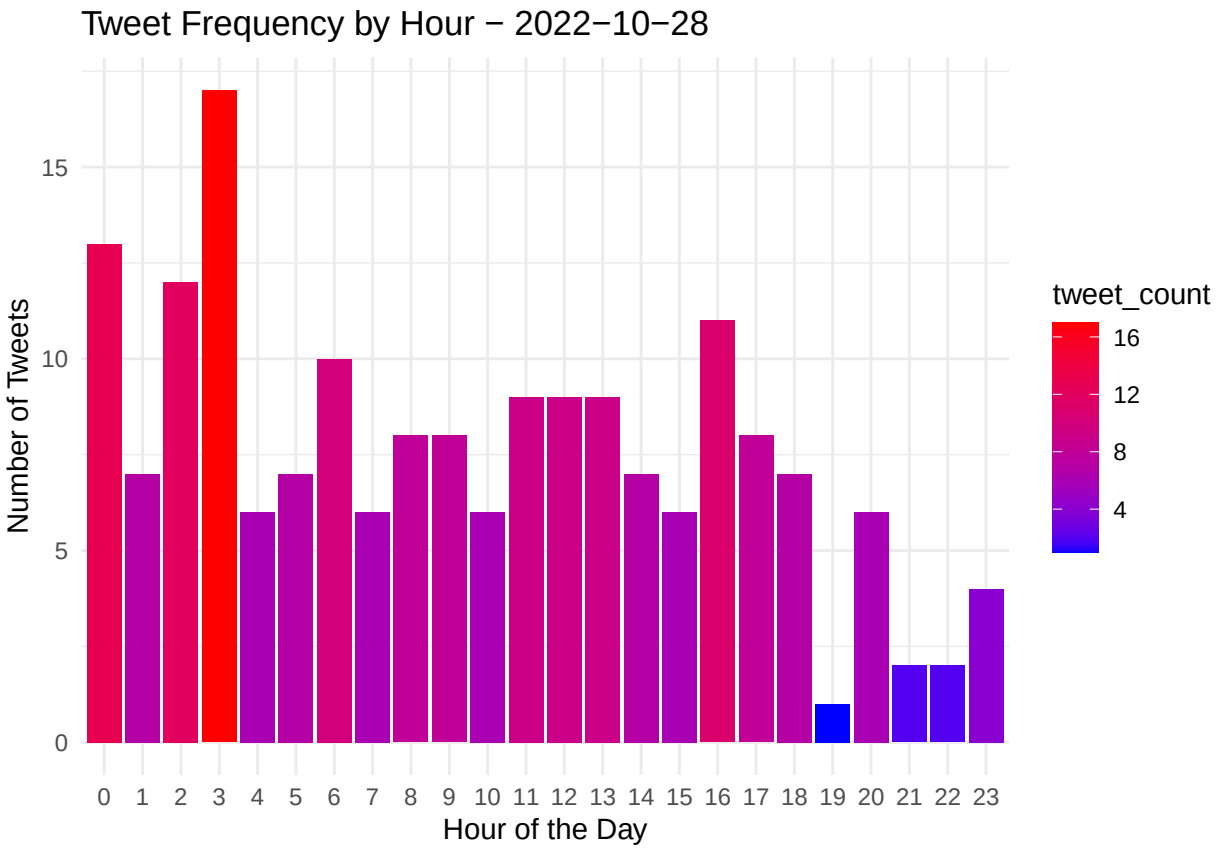
```r
# Graph for the first day
day1_graph <- ggplot(day1_tweets, aes(x = factor(hour), y = tweet_count, fill = tweet_count)) +
  geom_bar(stat = "identity") +
  labs(title = paste("Tweet Frequency by Hour -", unique(day1_tweets$day)),
       x = "Hour of the Day",
       y = "Number of Tweets") +
  scale_fill_gradient(low = "blue", high = "red") +
  scale_x_discrete(breaks = as.character(0:23)) +
  theme_minimal()

# Graph for the second day
day2_graph <- ggplot(day2_tweets, aes(x = factor(hour), y = tweet_count, fill = tweet_count)) +
  geom_bar(stat = "identity") +
  labs(title = paste("Tweet Frequency by Hour -", unique(day2_tweets$day)),
       x = "Hour of the Day",
       y = "Number of Tweets") +
  scale_fill_gradient(low = "blue", high = "red") +
  scale_x_discrete(breaks = as.character(0:23)) +
  theme_minimal()

# Graph for the third day
day3_graph <- ggplot(day3_tweets, aes(x = factor(hour), y = tweet_count, fill = tweet_count)) +
  geom_bar(stat = "identity") +
  labs(title = paste("Tweet Frequency by Hour -", unique(day3_tweets$day)),
       x = "Hour of the Day",
       y = "Number of Tweets") +
  scale_fill_gradient(low = "blue", high = "red") +
  scale_x_discrete(breaks = as.character(0:23)) +
  theme_minimal()
```

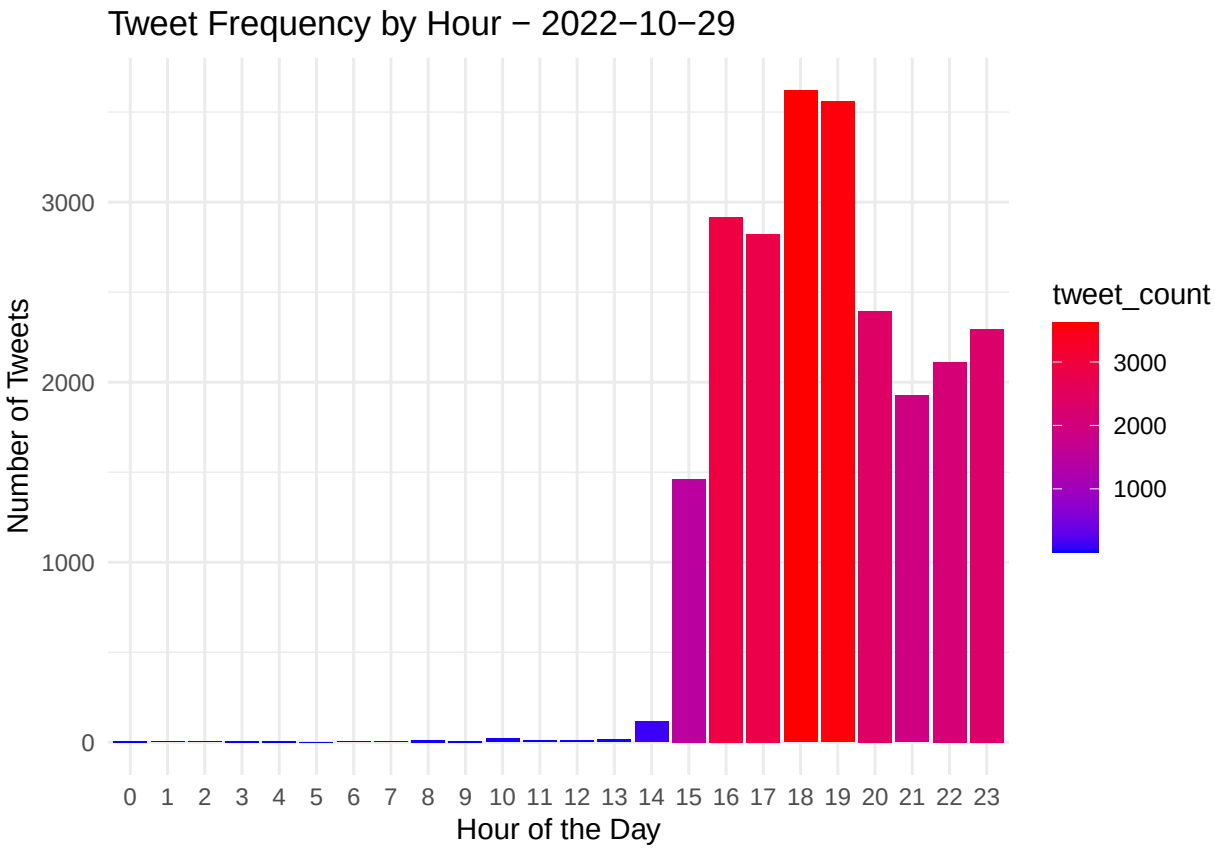
— Below are visualization and explanations of the graphs for Trend Analysis —

```r
# 2022-10-28
day1_graph
```



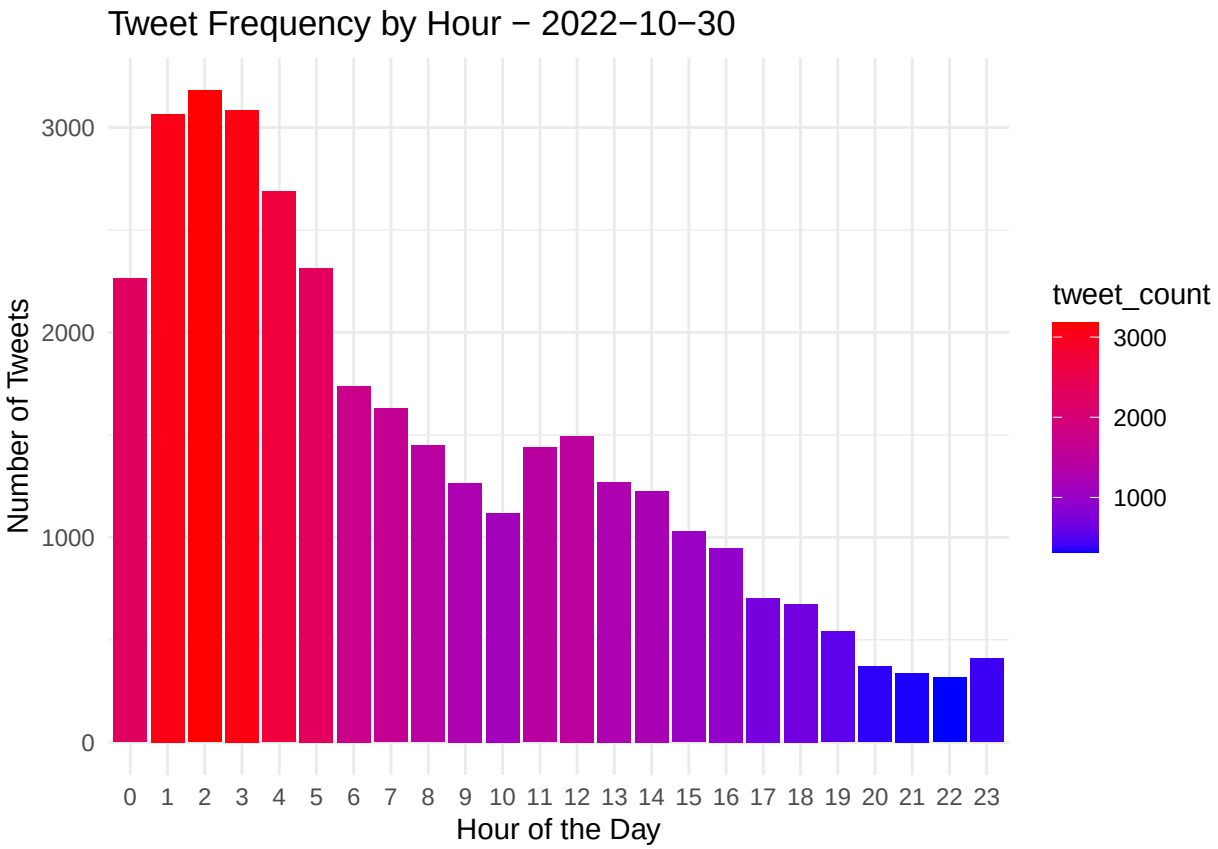
- The first graph shows very low tweet activity throughout the day, with no significant peaks or surges. This pattern indicates that the day was relatively uneventful in terms of the incident in focus. The absence of notable tweet volume suggests that discussions were likely unrelated to the tragedy, as it had not yet occurred or gained any attention.

```
# 2022-10-29
day2_graph
```



- The second graph indicates a significant rise in tweet activity starting around 3 PM, with a pronounced peak around 6 PM. This surge corresponds to the unfolding of the tragedy, as the incident likely occurred or was reported widely during this period. The increase in tweets shows how social media became a key platform for spreading information, expressing reactions, and coordinating responses as the tragedy unfolded.

```
# 2022-10-30
day3_graph
```



- The third graph displays a high volume of tweets during the early morning hours, peaking between midnight and 3 AM. Afterward, there is a steady decline in tweet activity throughout the rest of the day. The elevated tweet volume during the early hours likely reflects the aftermath of the Itaewon tragedy, as people continued sharing updates, reactions, and discussions in real time. The gradual decline suggests that as the day progressed, the urgency to tweet about the incident decreased, possibly due to the dissemination of official information or a shift in public focus.

END OF TREND ANALYSIS

Preparing data for Sentiment Analysis

- For the sentiment analysis, we focused on categorizing tweets into positive, neutral, and negative sentiments to observe emotional patterns throughout the day. This approach helps us understand how sentiments shift over time, providing valuable insights for responding to audience reactions.

```
# Categorize sentiment into Positive, Neutral, and Negative by hour per day
sentiment_summary <- clean %>%
  mutate(sentiment_category = ifelse(sentiment > 0, "Positive",
                                     ifelse(sentiment < 0, "Negative", "Neutral"))) %>%
  group_by(day = as.Date(date), hour, sentiment_category) %>%
  summarise(count = n(), .groups = "drop")

# Filter each day's data for plotting graphs
```

```r
day1_sentiments <- sentiment_summary %>% filter(day == unique(day)[1])
day2_sentiments <- sentiment_summary %>% filter(day == unique(day)[2])
day3_sentiments <- sentiment_summary %>% filter(day == unique(day)[3])
```

— Codes for plotting sentiment analysis —

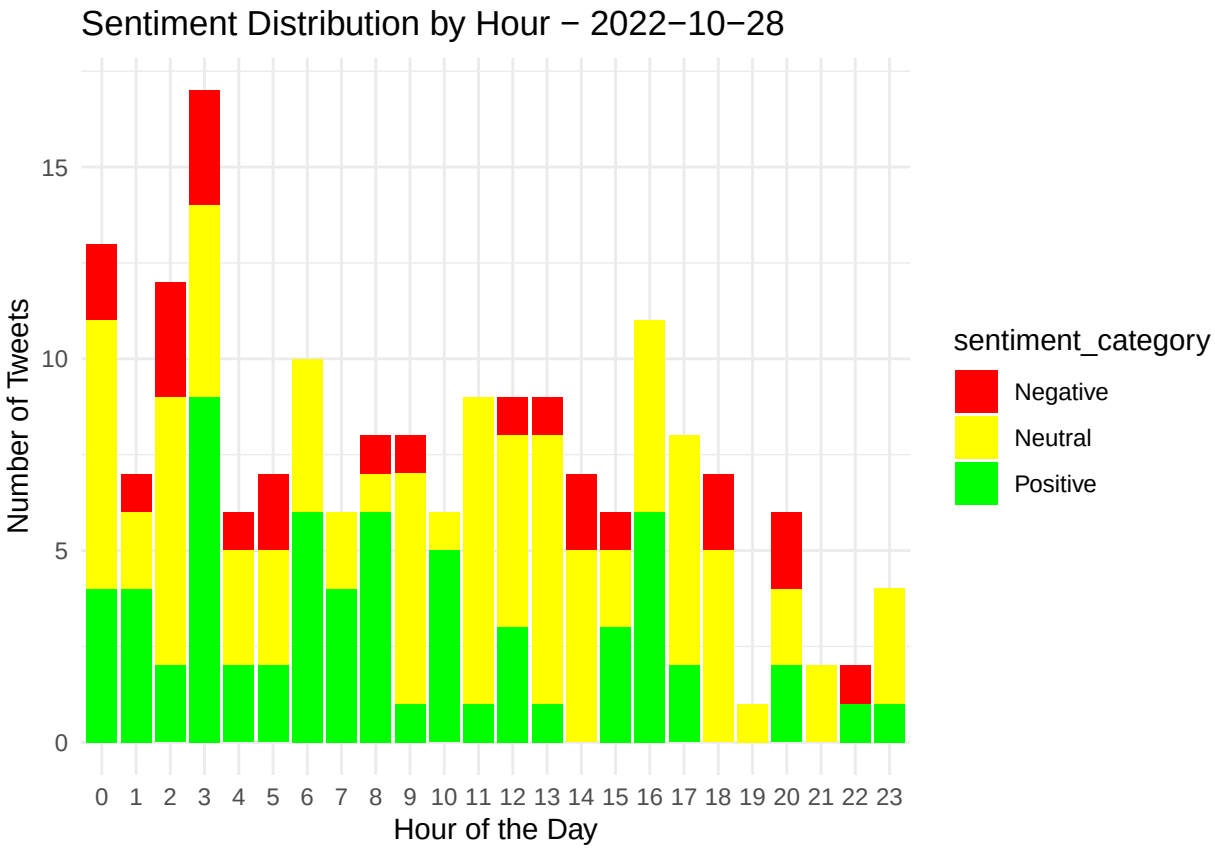
```r
# Plot sentiment trends for the first day
sentiment_day1_plot <- ggplot(day1_sentiments, aes(x = factor(hour), y = count, fill = sentiment_categor
  geom_bar(stat = "identity", position = "stack") +
  labs(title = paste("Sentiment Distribution by Hour -", unique(day1_sentiments$day)),
       x = "Hour of the Day",
       y = "Number of Tweets") +
  scale_fill_manual(values = c("red", "yellow", "green")) +
  scale_x_discrete(breaks = as.character(0:23)) +
  theme_minimal()

# Plot sentiment trends for the second day
sentiment_day2_plot <- ggplot(day2_sentiments, aes(x = factor(hour), y = count, fill = sentiment_categor
  geom_bar(stat = "identity", position = "stack") +
  labs(title = paste("Sentiment Distribution by Hour -", unique(day2_sentiments$day)),
       x = "Hour of the Day",
       y = "Number of Tweets") +
  scale_fill_manual(values = c("red", "yellow", "green")) +
  scale_x_discrete(breaks = as.character(0:23)) +
  theme_minimal()

# Plot sentiment trends for the third day
sentiment_day3_plot <- ggplot(day3_sentiments, aes(x = factor(hour), y = count, fill = sentiment_categor
  geom_bar(stat = "identity", position = "stack") +
  labs(title = paste("Sentiment Distribution by Hour -", unique(day3_sentiments$day)),
       x = "Hour of the Day",
       y = "Number of Tweets") +
  scale_fill_manual(values = c("red", "yellow", "green")) +
  scale_x_discrete(breaks = as.character(0:23)) +
  theme_minimal()
```

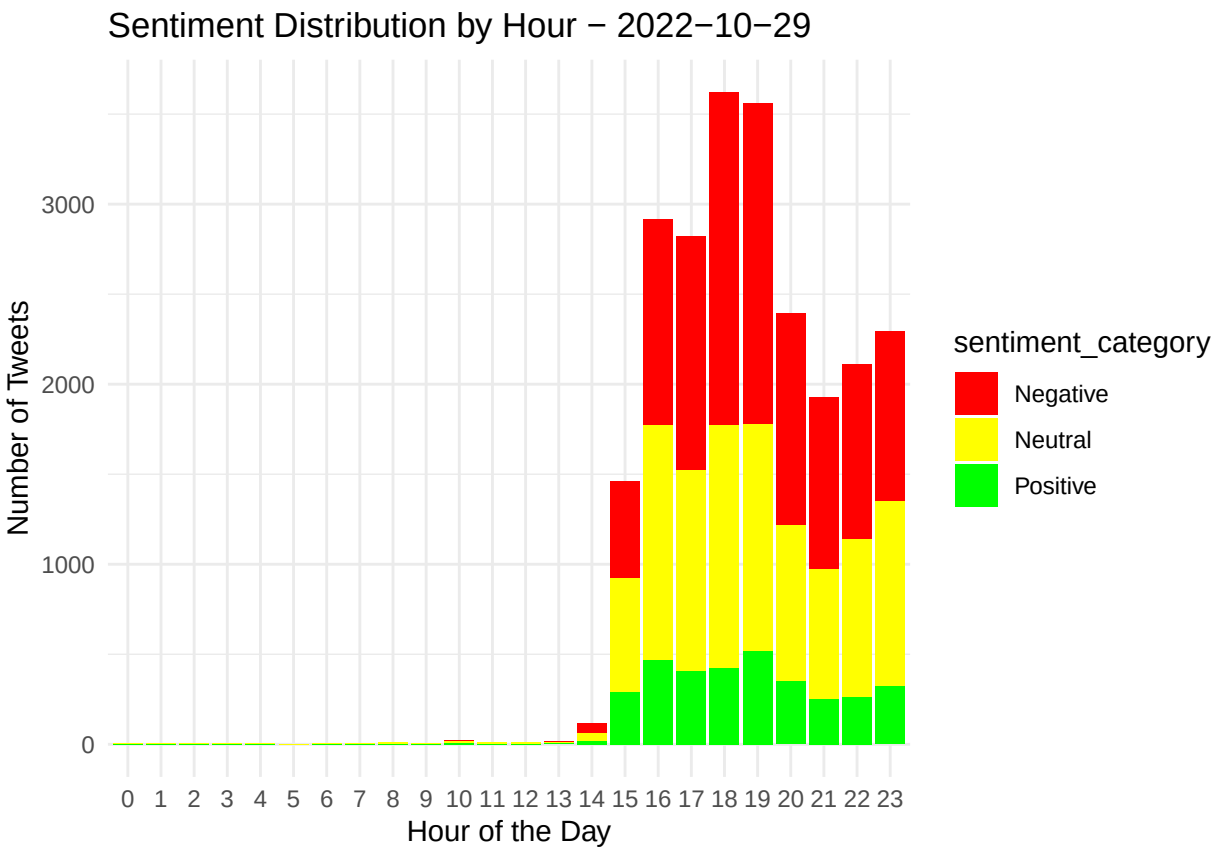
—Visualization and explanation of graphs for Sentiment Analysis

```r
# 2022-10-28
sentiment_day1_plot
```



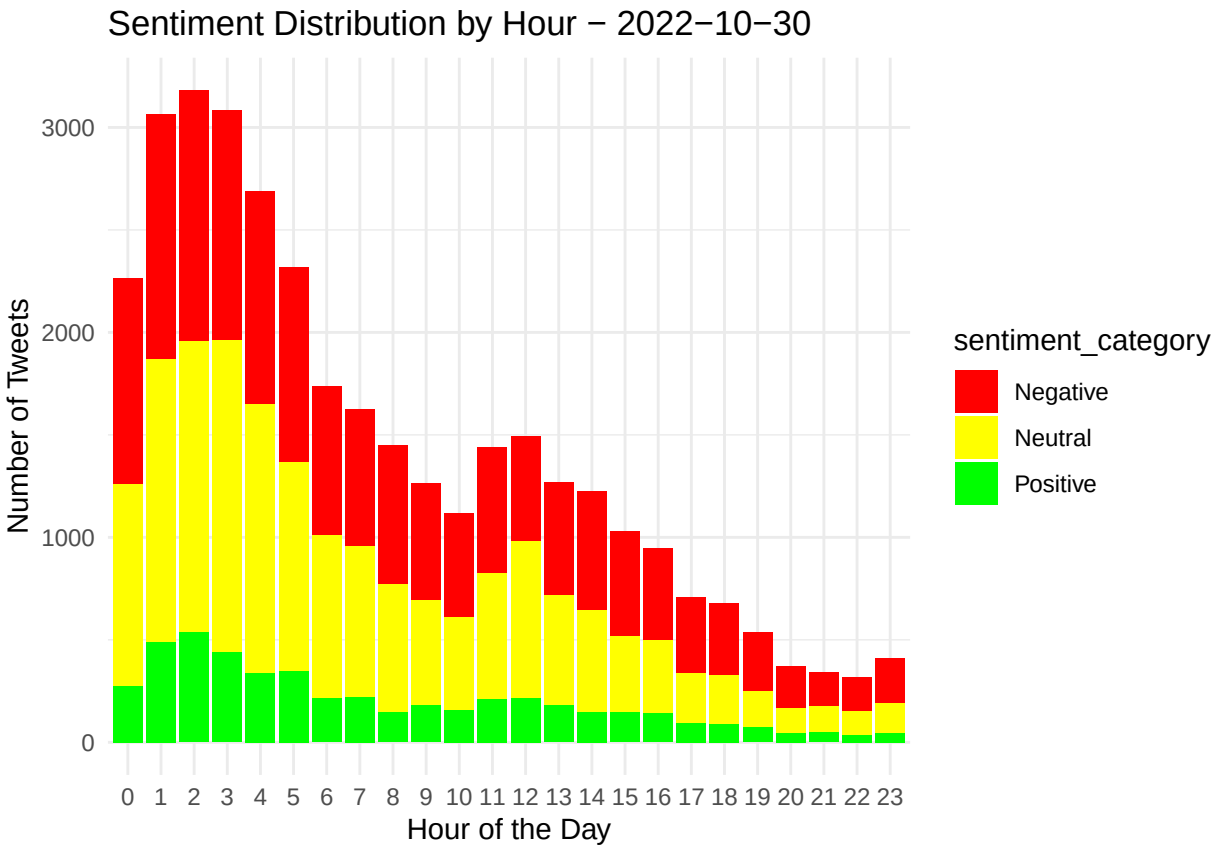
- The first graph shows a relatively low volume of tweets, with no significant peaks compared to the other days. The sentiment appears balanced, with positive, neutral, and negative tweets distributed throughout the day. This indicates that the event had not yet occurred or gained significant attention. The tweets likely reflect routine discussions or unrelated topics.

```
# 2022-10-29
sentiment_day2_plot
```



- The second graph indicates the absence of tweeter users during a 15-hour time frame, likely due to the insuing tragedy with a sharp increase in tweets starting from the afternoon (around 3 PM), with a peak between 6 PM and 7 PM. This trend likely corresponds to the timeline of the tragedy and its initial reports. The predominance of negative tweets aligns with the immediate aftermath, as information about the disaster began circulating widely. The spike in activity suggests widespread attention and concern as the event unfolded.

```
# 2022-10-30
sentiment_day3_plot
```



- The third graph shows a large volume of tweets throughout the day, peaking during the early morning hours (midnight to 3 AM) and gradually decreasing as the day progresses. Negative tweets dominate during the early hours, likely reflecting real-time reactions to the Itaewon tragedy, as people expressed shock, grief, or frustration. The steady decline in tweets as the day progresses suggests that the immediate emotional response subsided, giving way to more measured discussions or updates.

END OF SENTIMENT ANALYSIS