Final Project

Group 4

2023-12-22

```
library(rvest)
library(httr)
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(polite)
```

```
url <- "https://www.airlinequality.com/airline-reviews/emirates/page/1/?sortby=post_date%3ADesc&pagesiz
session <- bow(url, user_agent = "Educational")
session
```

```
## <polite session> https://www.airlinequality.com/airline-reviews/emirates/page/1/?sortby=post_date%3A
##     User-agent: Educational
##     robots.txt: 2 rules are defined for 1 bots
##    Crawl delay: 5 sec
##   The path is scrapable for this user-agent
```

```
webpage <- read_html(url)

ReviewTextP1 <- webpage %>%
  html_nodes("div.text_content") %>%
  html_text()
```

```
url <- "https://www.airlinequality.com/airline-reviews/emirates/page/2/?sortby=post_date%3ADesc&pagesiz
session <- bow(url, user_agent = "Educational")
session
```

```
## <polite session> https://www.airlinequality.com/airline-reviews/emirates/page/2/?sortby=post_date%3A
##     User-agent: Educational
##     robots.txt: 2 rules are defined for 1 bots
##    Crawl delay: 5 sec
##   The path is scrapable for this user-agent
```

```
webpage <- read_html(url)

ReviewTextP2 <- webpage %>%
```

```
  html_nodes("div.text_content") %>%
  html_text()
```

```
url <- "https://www.airlinequality.com/airline-reviews/emirates/page/3/?sortby=post_date%3ADesc&pagesiz
session <- bow(url, user_agent = "Educational")
session
```

```
## <polite session> https://www.airlinequality.com/airline-reviews/emirates/page/3/?sortby=post_date%3A
##     User-agent: Educational
##     robots.txt: 2 rules are defined for 1 bots
##    Crawl delay: 5 sec
##   The path is scrapable for this user-agent
```

```
webpage <- read_html(url)

ReviewTextP3<- webpage %>%
  html_nodes("div.text_content") %>%
  html_text()
```

```
R1 <- as.data.frame(ReviewTextP1)
R2 <- as.data.frame(ReviewTextP2)
R3 <- as.data.frame(ReviewTextP3)

names(R1) <- names(R2) <- names(R3) <- "Emirates Costumer Reviews"

Reviews_300 <- rbind(R1, R2, R3)
```

```
library(tidytext)
library(tidyr)
library(dplyr)
library(ggplot2)
library(wordcloud)
```

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

```
# Load the reviews into a tidy format
tidy_reviews <- Reviews_300 %>%
  mutate(review_id = row_number()) %>%
  unnest_tokens(word, "Emirates Costumer Reviews") %>%
  anti_join(stop_words)  # Remove common stop words
```

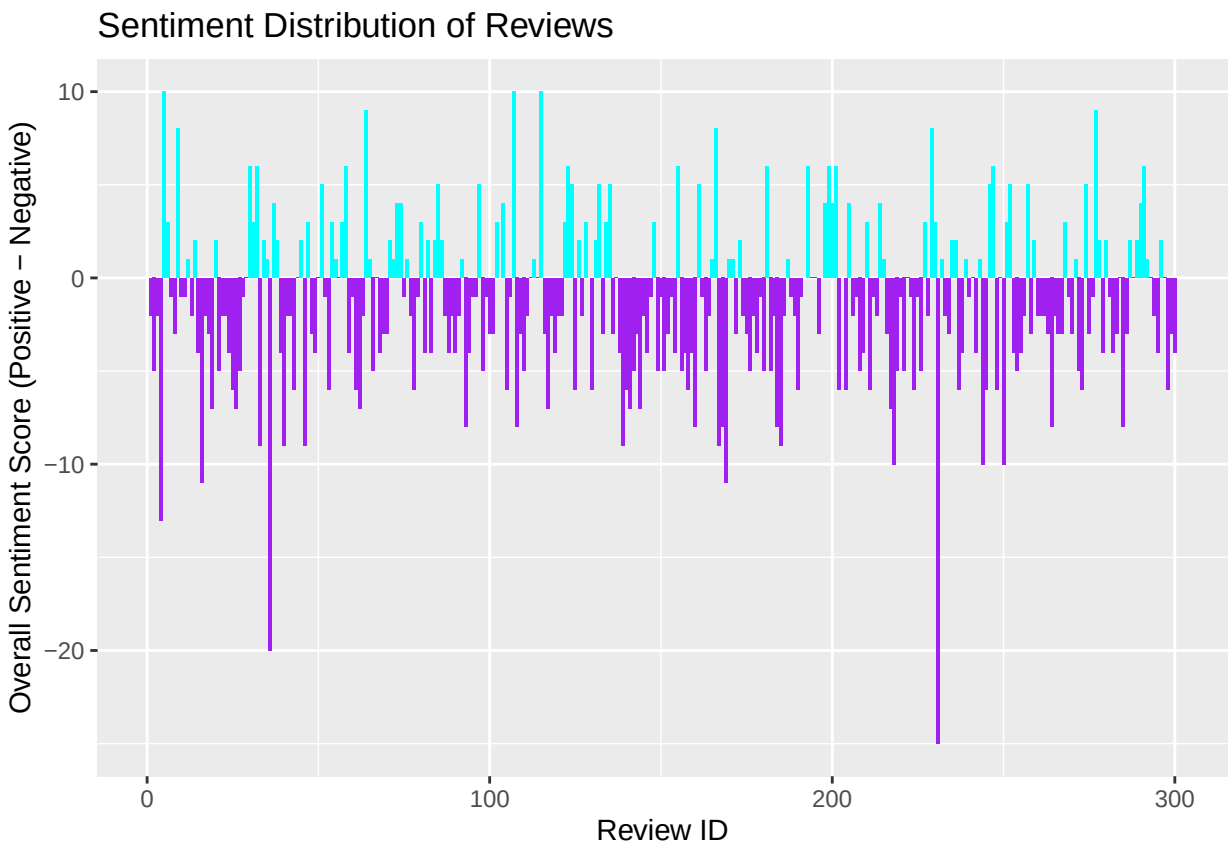
```
## Joining with `by = join_by(word)`
```

```
# Perform sentiment analysis
sentiments <- get_sentiments("bing")
sentiment_scores <- tidy_reviews %>%
  inner_join(sentiments, by = "word") %>%
  count(review_id, sentiment) %>%
  spread(sentiment, n, fill = 0) %>%
  mutate(overall_sentiment = positive - negative)

# Create a bar plot to visualize sentiment distribution
ggplot(sentiment_scores, aes(x = review_id, y = overall_sentiment, fill = factor(overall_sentiment > 0)
  geom_col() +
  scale_fill_manual(values = c("purple", "cyan"), guide = FALSE) +
  labs(title = "Sentiment Distribution of Reviews",
       x = "Review ID",
```

```
       y = "Overall Sentiment Score (Positive - Negative)")
```

```
## Warning: The `guide` argument in `scale_*()` cannot be `FALSE`. This was deprecated in
## ggplot2 3.3.4.
## i Please use "none" instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
# Create word clouds for positive and negative words
positive_words <- tidy_reviews %>%
  inner_join(sentiments, by = "word") %>%
  filter(sentiment == "positive") %>%
  count(word, sort = TRUE)

negative_words <- tidy_reviews %>%
  inner_join(sentiments, by = "word") %>%
  filter(sentiment == "negative") %>%
  count(word, sort = TRUE)
# Convert word frequency data to a named vector for wordcloud
positive_word_freq <- with(positive_words, setNames(n, word))
negative_word_freq <- with(negative_words, setNames(n, word))

# Create word clouds
wordcloud(words = names(positive_word_freq), freq = positive_word_freq, scale = c(3, 0.5), max.words = 5
```


```
wordcloud(words = names(negative_word_freq), freq = negative_word_freq, scale = c(3, 0.5), max.words = 5
```