Airbnb Postings Analysis Report

Executive Summary:

A company based in America called Airbedand-Breakast.com, also called Airbnb, operates through online platforms primarily for rental purposes. This report highlights the detailed analysis of text content in the posting description to draw hidden business insights. Using a comprehensive text mining approach, the following analysis based on numerical and textual data implemented four key frameworks: word frequency analysis, sentiment analysis using the AFINN lexicon, n-gram analysis focusing on bigrams, and TF-IDF analysis. The findings uncover the most predominant terms in the listed descriptions and overall sentiment trends, the key phrase patterns, and unique words in individual postings. This analysis and their insights would provide actionable information for enhancing the listing quality and aligning with the marketing strategies to enhance customer experience and competitive positioning.

The analysis of the posting descriptions began by examining the most used language by the host through a word frequency bar chart. The visualization (appendix, picture 1) displays the top 20 frequent words after cleaning the text and removing the stop words, uncovering the main set of terms that reoccurred throughout the listings. These terms are dominant, appearing to capture themes such as location, amenities, and service quality. This highlights the listing templates and could help us tailor marketing emails or texts based on the customer value the most.

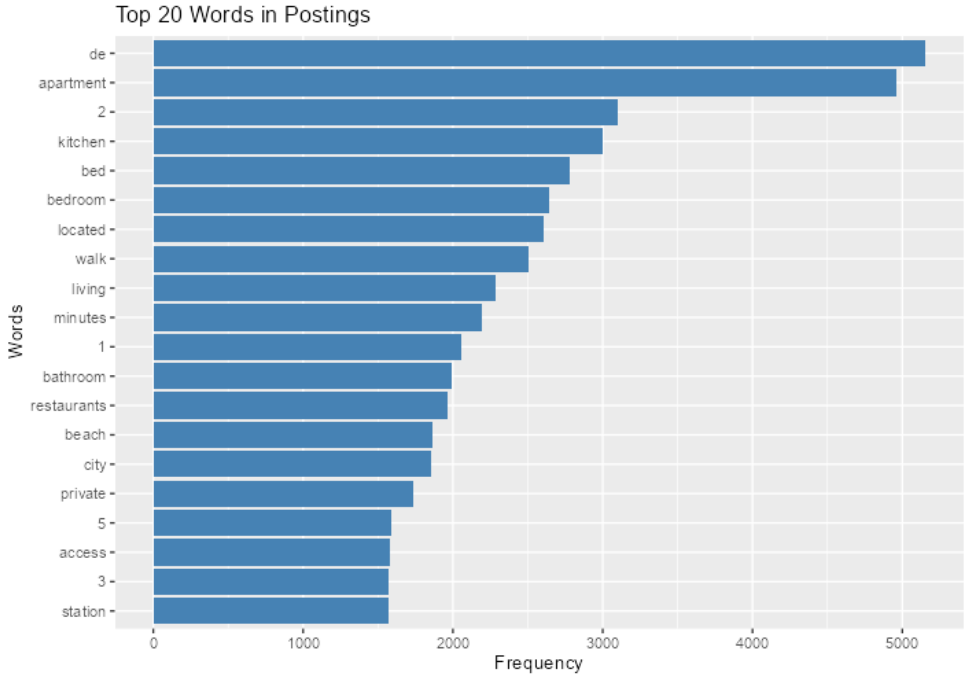
For further analysis, understanding the tone of the overall descriptions is crucial to learning about the trends of the language used towards a positive and negative or a neutral tone. To understand that a sentiment analysis has been conducted using the AFINN sentiment lexicon.

Which resulted in a deeper understanding of the distribution of segments across the postings, offering a visual representation of the tone (appendix, picture 2). Strikingly, the distribution of these scores helps us understand the indication of guest perceptions and expectations, indicating that the way listings are described has a direct impact on customer engagement and satisfaction.

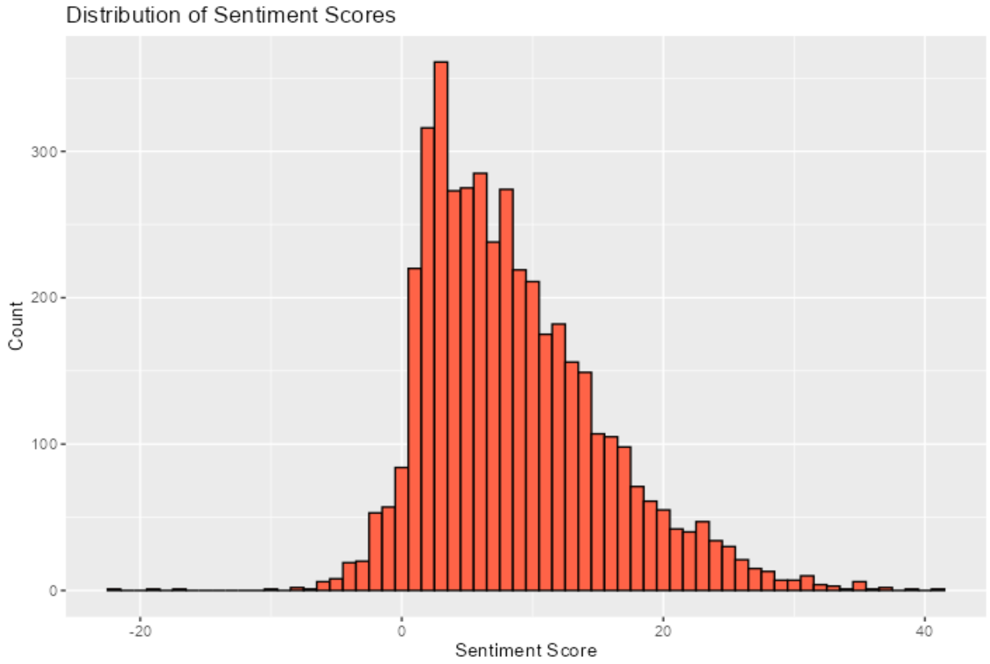
Building on this, the bigrams analysis revealed that combinations of words frequently appear together. As the bar chart (appendix picture 3) shows the top 15 bigrams, uncovering phrases like “spacious room” and “central location” could be the key feature of the listings. These terms are not only emphasized in their descriptions but also serve as potential indicators of the amenities, and it is a value proposition to attract guests. The TF-IDF framework highlighted the words that are distinctive to postings. As the bar chart (appendix, Picture 4) shows, the top 20 words with the highest TF-IDF scores identify terms that are extremely relevant in differentiating individual listings but are not as common in the entire dataset. These distinct terms can be used for market segmentation, improve the differentiation of the listings, enhance the search algorithm, thus providing a competitive advantage. An interactive dashboard was developed to enable dynamic exploration of these insights using an R shiny application. They consist of four individual tabs-each dedicated to the key analysis for the Airbnb management to seamlessly navigate through the frequency, sentiment analysis, bi-grams and TF-IDF visualizations. This tool helps with exploration in real-time and allows us to identify the trends and distinctive features through data-driven driven strategies. Overall, the integration of the text mining framework has highlighted the varied view of the Airbnb posting. The analysis reveals the dominant language patterns, distinctive tone in the sentiment, and unique phrases that differentiate from the individual listings. These insights are addressed to improve Airbnb optimization in the listing descriptions and enhance the marketing strategy and boost the customer engagement and positioning.

Appendix:

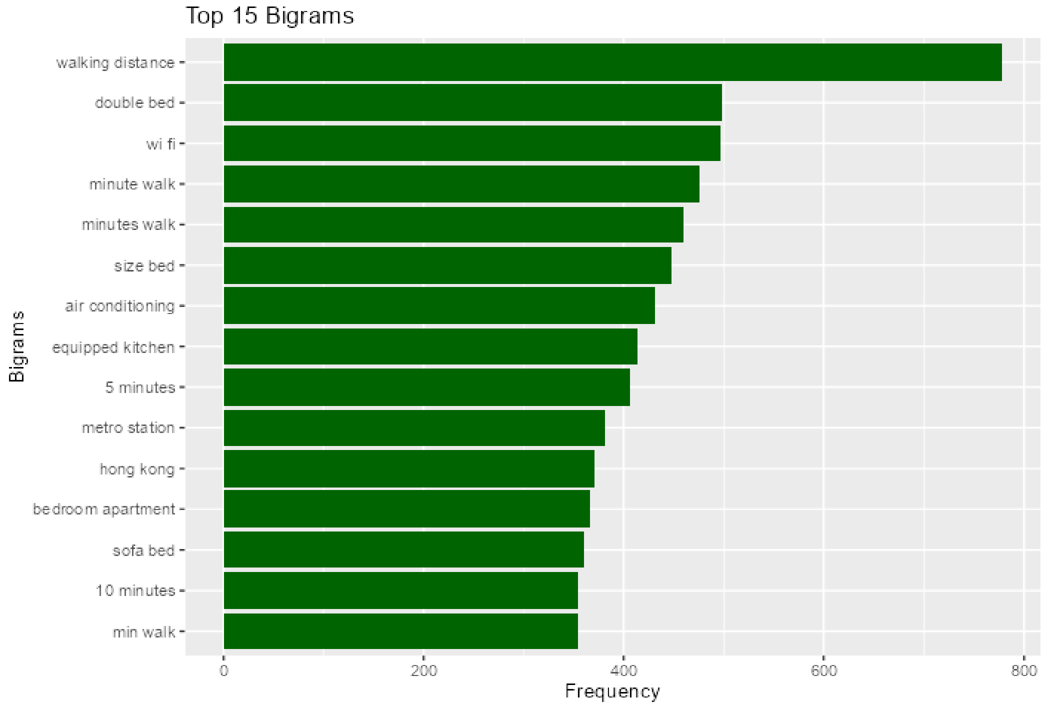
Picture 1: Frequency Analysis



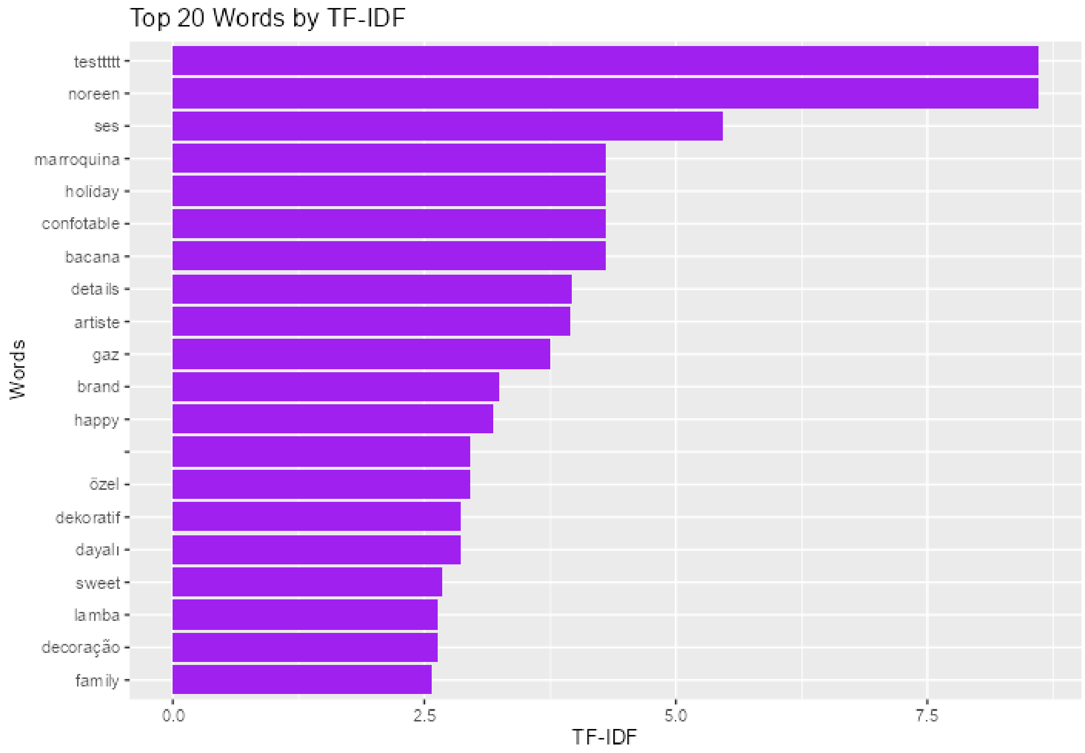
Picture 2: Sentiment Analysis



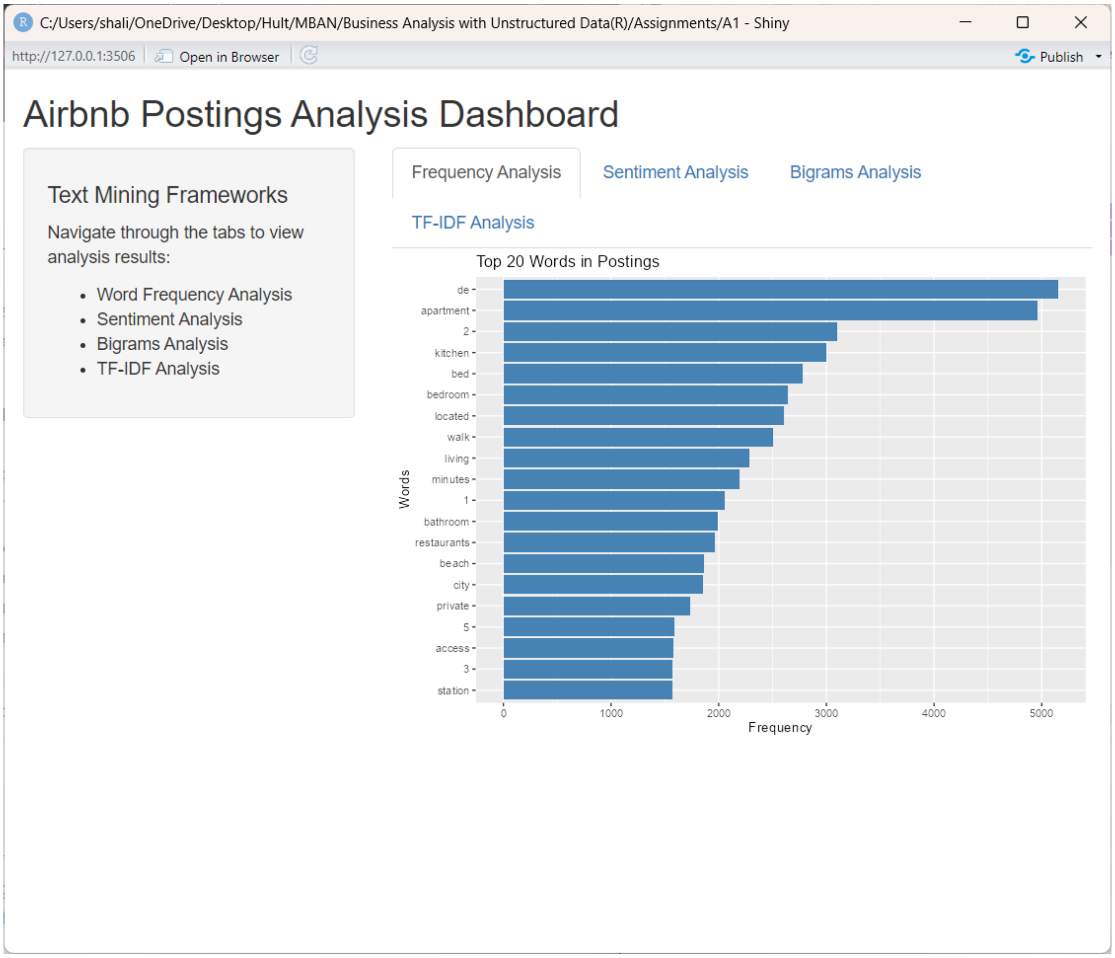
Picture 3: Bigram Analysis



Picture 4: TF-IDF Analysis



Interactive Dashboard using R shiny:



C:/Users/shali/OneDrive/Desktop/Hult/MBAN/Business Analysis with Unstructured Data(R)/Assignments/A1 - Shiny

http://127.0.0.1:3506 | Open in Browser | Publish

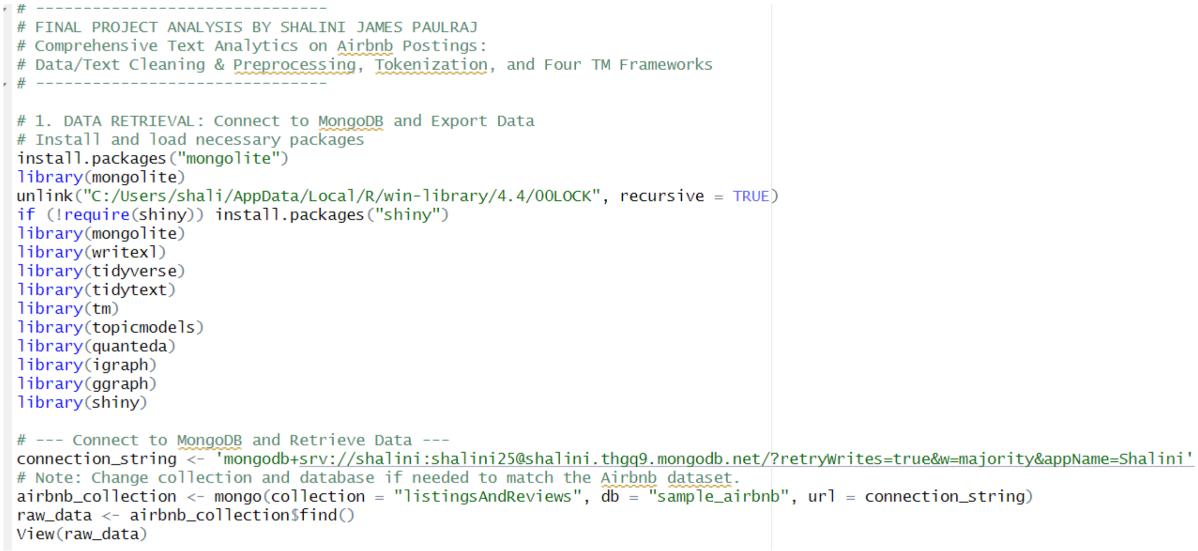
Airbnb Postings Analysis Dashboard

Text Mining Frameworks

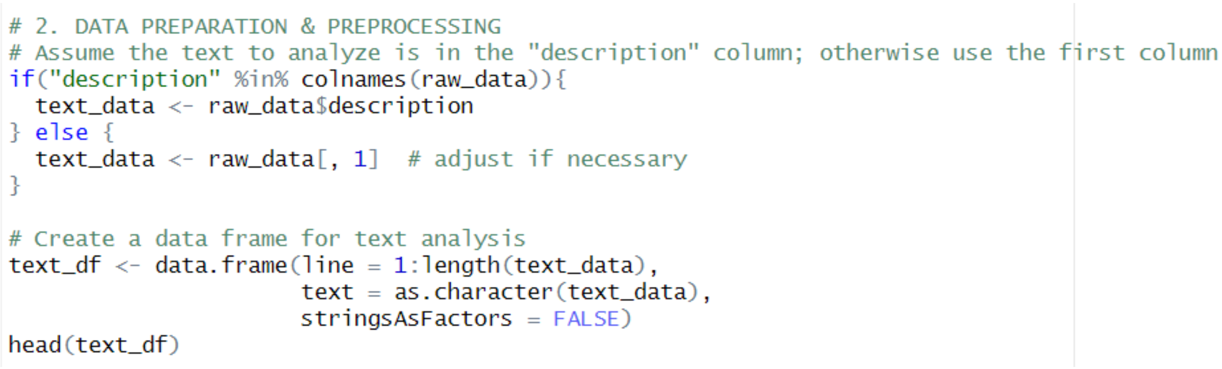
Navigate through the tabs to view analysis results:

- Word Frequency Analysis
- Sentiment Analysis
- Bigrams Analysis
- TF-IDF Analysis

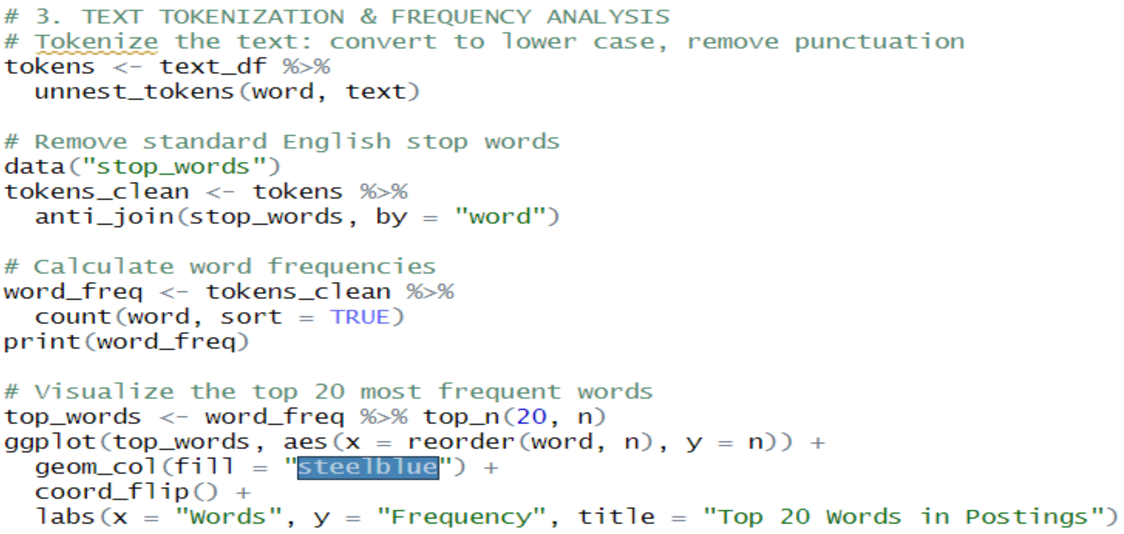
Screenshot the R code:



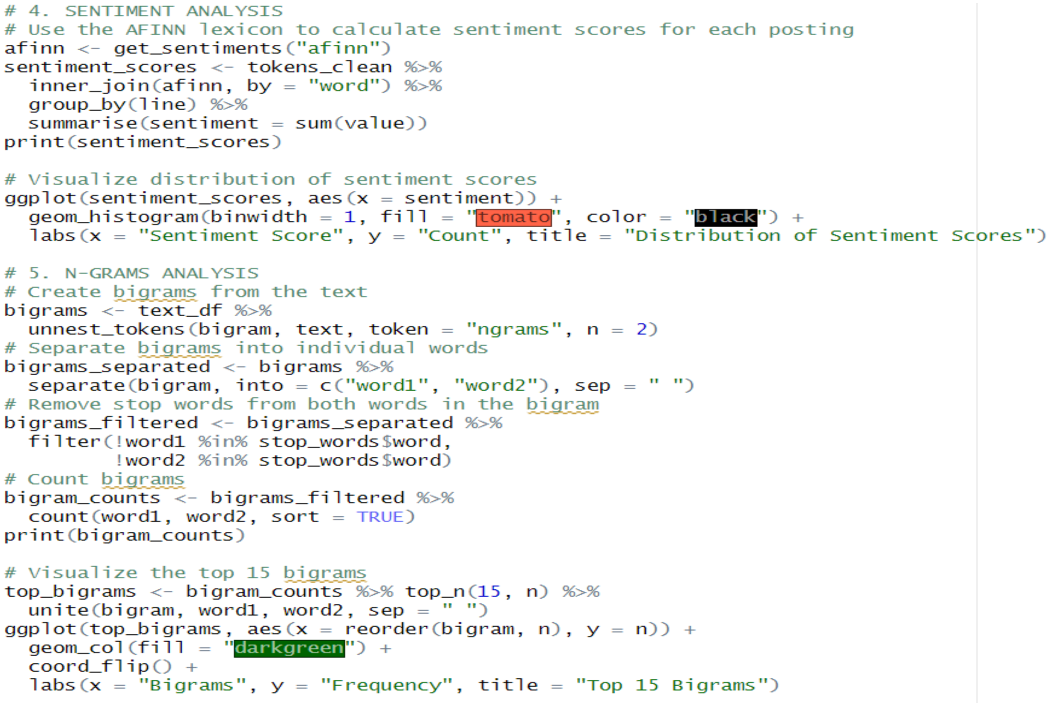
```
# ------------------------------
# FINAL PROJECT ANALYSIS BY SHALINI JAMES PAULRAJ
# Comprehensive Text Analytics on Airbnb Postings:
# Data/Text Cleaning & Preprocessing, Tokenization, and Four TM Frameworks
# ------------------------------

# 1. DATA RETRIEVAL: Connect to MongoDB and Export Data
# Install and load necessary packages
install.packages("mongolite")
library(mongolite)
unlink("C:/Users/shali/AppData/Local/R/win-library/4.4/00LOCK", recursive = TRUE)
if (!require(shiny)) install.packages("shiny")
library(mongolite)
library(writexl)
library(tidyverse)
library(tidytext)
library(tm)
library(topicmodels)
library(quanteda)
library(igraph)
library(ggraph)
library(shiny)

# --- Connect to MongoDB and Retrieve Data ---
connection_string <- 'mongodb+srv://shalini:shalini25@shalini.thgq9.mongodb.net/?retryWrites=true&w=majority&appName=Shalini'
# Note: Change collection and database if needed to match the Airbnb dataset.
airbnb_collection <- mongo(collection = "listingsAndReviews", db = "sample_airbnb", url = connection_string)
raw_data <- airbnb_collection$find()
View(raw_data)
```



```
# 2. DATA PREPARATION & PREPROCESSING
# Assume the text to analyze is in the "description" column; otherwise use the first column
if("description" %in% colnames(raw_data)){
  text_data <- raw_data$description
} else {
  text_data <- raw_data[, 1]  # adjust if necessary
}

# Create a data frame for text analysis
text_df <- data.frame(line = 1:length(text_data),
                      text = as.character(text_data),
                      stringsAsFactors = FALSE)
head(text_df)
```



```
# 3. TEXT TOKENIZATION & FREQUENCY ANALYSIS
# Tokenize the text: convert to lower case, remove punctuation
tokens <- text_df %>%
  unnest_tokens(word, text)

# Remove standard English stop words
data("stop_words")
tokens_clean <- tokens %>%
  anti_join(stop_words, by = "word")

# Calculate word frequencies
word_freq <- tokens_clean %>%
  count(word, sort = TRUE)
print(word_freq)

# Visualize the top 20 most frequent words
top_words <- word_freq %>% top_n(20, n)
ggplot(top_words, aes(x = reorder(word, n), y = n)) +
  geom_col(fill = "steelblue") +
  coord_flip() +
  labs(x = "Words", y = "Frequency", title = "Top 20 Words in Postings")
```



```
# 4. SENTIMENT ANALYSIS
# Use the AFINN lexicon to calculate sentiment scores for each posting
afinn <- get_sentiments("afinn")
sentiment_scores <- tokens_clean %>%
  inner_join(afinn, by = "word") %>%
  group_by(line) %>%
  summarise(sentiment = sum(value))
print(sentiment_scores)

# Visualize distribution of sentiment scores
ggplot(sentiment_scores, aes(x = sentiment)) +
  geom_histogram(binwidth = 1, fill = "tomato", color = "black") +
  labs(x = "Sentiment Score", y = "Count", title = "Distribution of Sentiment Scores")

# 5. N-GRAMS ANALYSIS
# Create bigrams from the text
bigrams <- text_df %>%
  unnest_tokens(bigram, text, token = "ngrams", n = 2)
# Separate bigrams into individual words
bigrams_separated <- bigrams %>%
  separate(bigram, into = c("word1", "word2"), sep = " ")
# Remove stop words from both words in the bigram
bigrams_filtered <- bigrams_separated %>%
  filter(!word1 %in% stop_words$word,
         !word2 %in% stop_words$word)
# Count bigrams
bigram_counts <- bigrams_filtered %>%
  count(word1, word2, sort = TRUE)
print(bigram_counts)

# Visualize the top 15 bigrams
top_bigrams <- bigram_counts %>% top_n(15, n) %>%
  unite(bigram, word1, word2, sep = " ")
ggplot(top_bigrams, aes(x = reorder(bigram, n), y = n)) +
  geom_col(fill = "darkgreen") +
  coord_flip() +
  labs(x = "Bigrams", y = "Frequency", title = "Top 15 Bigrams")
```

```
# 6. TF-IDF ANALYSIS
# Calculate TF-IDF for words grouped by document (each posting is a document)
tfidf <- tokens_clean %>%
  count(line, word, sort = TRUE) %>%
  bind_tf_idf(word, line, n) %>%
  arrange(desc(tf_idf))
print(tfidf)

# Visualize top words by TF-IDF
top_tfidf <- tfidf %>% group_by(word) %>% summarise(tf_idf = max(tf_idf)) %>% top_n(20, tf_idf)
ggplot(top_tfidf, aes(x = reorder(word, tf_idf), y = tf_idf)) +
  geom_col(fill = "purple") +
  coord_flip() +
  labs(x = "Words", y = "TF-IDF", title = "Top 20 Words by TF-IDF")

# Business Insight:
# - Frequency analysis reveals the most common words in the posting descriptions.
# - Sentiment analysis (using the AFINN lexicon) shows overall sentiment trends per posting.
# - N-grams analysis identifies common word pairs (bigrams) that may capture key phrases.
# - TF-IDF analysis highlights unique words that differentiate individual postings.
```