CST4070 NLP Challenge

Code ▾

Introduction

This report presents an analysis of Airbnb reviews, focusing on comparing word usage between the earliest and most recent years in the dataset. The primary objective is to identify trends in how guests describe their experiences over time.

Data Loading

Hide

```
airbnb_data <- read.csv('reviews.csv')


sum(is.na(airbnb_data$id))
```

```
[1] 0
```

Hide

```
airbnb_data
```

listing_id	id	date	reviewer_id	reviewer_name	▶
<dbl>	<dbl>	<chr>	<int>	<chr>	
13913	8.077000e+04	2010-08-18	177109	Michael	
13913	3.675680e+05	2011-07-11	19835707	Mathias	
13913	5.295790e+05	2011-09-13	1110304	Kristin	
13913	5.954810e+05	2011-10-03	1216358	Camilla	
13913	6.129470e+05	2011-10-09	490840	Jorik	
13913	4.847959e+06	2013-05-28	6405442	Vera	
13913	8.142329e+06	2013-10-17	9195551	Honi	
13913	1.187659e+07	2014-04-17	5194009	Alessandro	
13913	4.666957e+07	2015-09-12	42970248	Oleh	
13913	6.455903e+07	2016-03-05	45337884	Mo	

1-10 of 1,887,519 rows | 1-5 of 6 columns Previous **1** 2 3 4 5 6 ... 100 Next

Hide

```
NA
NA
NA
NA
```

Handling Missing Values and Data Cleaning

- Checking for missing values in the dataset.
- Dropping rows with missing `comments`.
- Displaying the data structure after cleaning.

Hide

```
# Seeing missing values
colSums(is.na(airbnb_data))
```

```
   listing_id            id          date   reviewer_id 
            0             0             0             0 
reviewer_name      comments 
            0            39 
```

Hide

```
# drop rows with missing values
reviews_data <- airbnb_data |>
   drop_na(comments)

colSums(is.na(reviews_data))
```

```
   listing_id            id          date   reviewer_id 
            0             0             0             0 
reviewer_name      comments 
            0             0 
```

Hide

```
str(reviews_data)
```

```
'data.frame':   1887480 obs. of  6 variables:
 $ listing_id   : num  13913 13913 13913 13913 13913 ...
 $ id           : num  80770 367568 529579 595481 612947 ...
 $ date         : chr  "2010-08-18" "2011-07-11" "2011-09-13" "2011-10-03" ...
 $ reviewer_id  : int  177109 19835707 1110304 1216358 490840 6405442 9195551 5194009 4297024
8 45337884 ...
 $ reviewer_name: chr  "Michael" "Mathias" "Kristin" "Camilla" ...
 $ comments     : chr  "My girlfriend and I hadn't known Alina before we took the leap of fai
th to rent her flat. Alina just couldn't b"| __truncated__ "Alina was a really good host. The 
flat is clean and tidy - and really close to Finsbury Park station which is q"| __truncated__
"Alina is an amazing host. She made me feel right at home. It was more like hanging out with
a friend than a com"| __truncated__ "Alina's place is so nice, the room is big and clean, and
the bed is huge. Alina is a great host, and she made s"| __truncated__ ...
```

Checking for Duplicates

- Count of the number of duplicate entries in the dataset.

Hide

```
# count of duplicated data
sum(duplicated(reviews_data))
```

```
[1] 0
```

Hide

```
names(reviews_data)
```

```
[1] "listing_id"    "id"            "date"          "reviewer_id"  
[5] "reviewer_name" "comments"     
```

Hide

```
dim(reviews_data)
```

```
[1] 1887480       6
```

Hide

```
head(reviews_data)
```

	listing_id	id	date	reviewer_id	reviewer_name	▶
	<dbl>	<dbl>	<chr>	<int>	<chr>	
1	13913	80770	2010-08-18	177109	Michael	
2	13913	367568	2011-07-11	19835707	Mathias	
3	13913	529579	2011-09-13	1110304	Kristin	
4	13913	595481	2011-10-03	1216358	Camilla	
5	13913	612947	2011-10-09	490840	Jorik	
6	13913	4847959	2013-05-28	6405442	Vera	

6 rows | 1-6 of 6 columns

Hide

```
NA
NA
```

Date Conversion and Year Extraction

- Converting the date column to date format .
- sorting unique years to identify the earliest and latest years.

Hide

```
library(lubridate)

#converting the date column to date format from char
reviews_data$date <- ymd(reviews_data$date)
#reviews_data$id <- as.integer(reviews_data$id)
class(reviews_data$date)
```

```
[1] "Date"
```

Hide

```
unique_years <- reviews_data |>
  mutate(year = format(date, "%Y")) |>
  filter(!is.na(as.numeric(year))) |>  # Remove non-numeric year values
  pull(year) |>
  unique() |>
  sort()

print(unique_years)
```

```
 [1] "2009" "2010" "2011" "2012" "2013" "2014" "2015" "2016" "2017"
[10] "2018" "2019" "2020" "2021" "2022" "2023" "2024"
```

Hide

```
head(reviews_data)
```

	listing_id <dbl>	id <dbl>	date <date>	reviewer_id <int>	reviewer_name <chr>	▶
1	13913	80770	2010-08-18	177109	Michael	
2	13913	367568	2011-07-11	19835707	Mathias	
3	13913	529579	2011-09-13	1110304	Kristin	
4	13913	595481	2011-10-03	1216358	Camilla	
5	13913	612947	2011-10-09	490840	Jorik	
6	13913	4847959	2013-05-28	6405442	Vera	

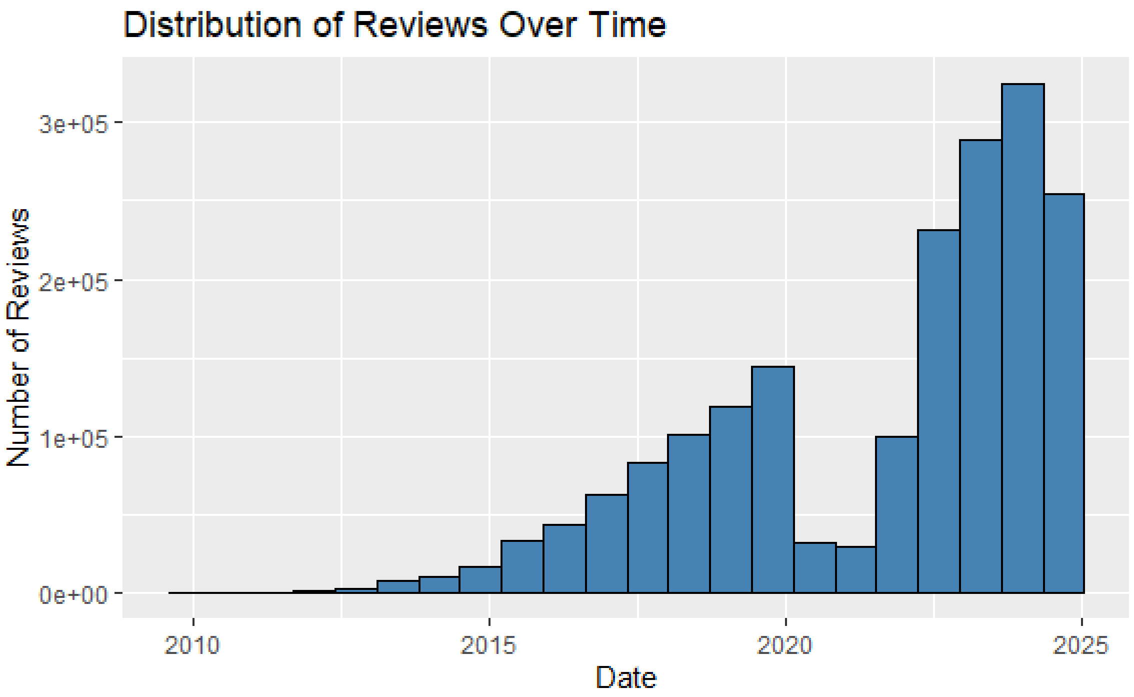
6 rows | 1-6 of 6 columns

Hide

```
NA
```

Hide

```
# histogram to understand the distribution of the reviews
library(ggplot2)

ggplot(reviews_data, aes(x = date)) +
  geom_histogram(bins = 22, fill = "steelblue", color = "black") +
  labs(title = "Distribution of Reviews Over Time",
       x = "Date",
       y = "Number of Reviews")
```



Distribution of reveiws overtime overveiw

The histogram shows that the number of reviews in the earliest years is lower compared to recent years. This trend could be attributed to various factors, including the growth of Airbnb listings over time, increased user adoption, and changes in review culture, all contributing to a rise in review numbers.

Splitting the Data into Early and Recent Groups

- Splitting the dataset into two groups based on the first and last available years (2009 and 2024).

Hide

```r
year_range <- reviews_data |>
  mutate(year = format(date, "%Y")) |>
  pull(year) |>
  unique() |>
  sort()
# Initialising the first and last available years
earliest_year <- min(year_range)
latest_year <- max(year_range)

# dataframes for early and recent reviews
early_reviews <- reviews_data |>
  mutate(year = as.numeric(format(date, "%Y"))) |>
  filter(year == earliest_year)

recent_reviews <- reviews_data |>
  mutate(year = as.numeric(format(date, "%Y"))) |>
  filter(year == latest_year)

nrow(early_reviews)
```

```
[1] 1
```

Hide

```r
nrow(recent_reviews)
```

```
[1] 399848
```

Hide

```r
head(early_reviews)
```

	listing_id <dbl>	id <dbl>	date <date>	reviewer_id <int>	reviewer_name <chr>
1	15400	21032	2009-12-21	53815	Hailey

1 row | 1-6 of 7 columns

Hide

```r
head(recent_reviews)
```

	listing_id <dbl>	id <dbl>	date <date>	reviewer_id <int>	reviewer_name <chr>
1	13913	1.148453e+18	2024-05-03	183479282	Gemma
2	13913	1.175245e+18	2024-06-09	35008871	Zehra
3	13913	1.197766e+18	2024-07-10	106138105	Isabell
4	83027	1.093380e+18	2024-02-17	358768588	Nidal

	listing_id <dbl>	id <dbl>	date <date>	reviewer_id <int>	reviewer_name <chr>	▸
5	15400	1.120257e+18	2024-03-25	100840491	Tim	
6	15400	1.144822e+18	2024-04-28	3321262	Beth Ann	

6 rows | 1-6 of 7 columns

Hide

```
NA
NA
NA
```

Text proccessing

- conversion of the comments column to lowercase
- removal of punctuation marks
- splitting the text into individual words (tokens)
- removing stop words that don’t mean anything to the analysis
- removing single characters and html syntax such as (br)

Hide

```
library(dplyr)
library(tidytext)
library(stopwords)
```

```
Warning: package ‘stopwords’ was built under R version 4.4.2
```

Hide

```
process_text <- function(temp_df) {
  temp_df <- temp_df |>
    mutate(comments_lower = tolower(comments)) |>
    mutate(comments_lower = gsub("[[:punct:]]", " ", comments_lower)) |>
     mutate(comments_lower = gsub("\\b\\w{1}\\b", "", comments_lower)) |>
    mutate(comments_lower = gsub("\\bbr\\b", "",
comments_lower)) |>
    unnest_tokens(word, comments_lower) |>
    anti_join(get_stopwords(language = "en") |>
                rename(word = word), by = "word")

  return(temp_df)
}

early_reviews <- process_text(early_reviews)
recent_reviews <- process_text(recent_reviews)

head(early_reviews)
```

	listing_id <dbl>	id <dbl>	date <date>	reviewer_id <int>	reviewer_name <chr>	▸
1	15400	21032	2009-12-21	53815	Hailey	
2	15400	21032	2009-12-21	53815	Hailey	
3	15400	21032	2009-12-21	53815	Hailey	
4	15400	21032	2009-12-21	53815	Hailey	
5	15400	21032	2009-12-21	53815	Hailey	
6	15400	21032	2009-12-21	53815	Hailey	

6 rows | 1-6 of 8 columns

Hide

```
head(recent_reviews)
```

	listing_id <dbl>	id <dbl>	date <date>	reviewer_id <int>	reviewer_name <chr>	▸
1	13913	1.148453e+18	2024-05-03	183479282	Gemma	
2	13913	1.148453e+18	2024-05-03	183479282	Gemma	
3	13913	1.148453e+18	2024-05-03	183479282	Gemma	
4	13913	1.148453e+18	2024-05-03	183479282	Gemma	
5	13913	1.148453e+18	2024-05-03	183479282	Gemma	
6	13913	1.148453e+18	2024-05-03	183479282	Gemma	

6 rows | 1-6 of 8 columns

Hide

```
NA
NA
```

Hide

```
names(early_reviews)
```

```
[1] "listing_id"    "id"            "date"          "reviewer_id"
[5] "reviewer_name" "comments"      "year"          "word"
```

Hide

```
names(recent_reviews)
```

```
[1] "listing_id"    "id"            "date"          "reviewer_id"
[5] "reviewer_name" "comments"      "year"          "word"
```

Hide

```
dim(early_reviews)
```

```
[1] 18  8
```

Hide

```
dim(recent_reviews)
```

```
[1] 10331343        8
```

lemmatisation

- applying lemmatisation to the word column in both the early and recent reviews datasets reducing words to their root form, helping to standardize variations of words and improve the accuracy of analysis.

Hide

```
# Apply lemmatization to the 'word' column
early_reviews <- early_reviews |>
  mutate(word = lemmatize_words(word))

recent_reviews <- recent_reviews |>
  mutate(word = lemmatize_words(word))

head(early_reviews$word)
```

```
[1] "love"     "stay"     "phillipa" "place"    "chelsea" 
[6] "flat"
```

Hide

```
head(recent_reviews$word)
```

```
[1] "alina"    "really"   "lovely"   "host"     "friendly"
[6] "welcome"
```

Hide

```
# Count of word frequency for each group
early_word_freq <- early_reviews |>
  count(word, sort = TRUE)

recent_word_freq <- recent_reviews |>
  count(word, sort = TRUE)

# View top words
head(early_word_freq, 25)
```

	word	**n**
	<chr>	<int>
1	love	2
2	stay	2
3	chelsea	1
4	close	1
5	flat	1
6	fun	1
7	great	1
8	lot	1
9	lovely	1
10	metro	1

1-10 of 16 rows Previous **1** 2 Next

Hide

```
head(recent_word_freq,15)
```

	word	**n**
	<chr>	<int>
1	stay	206807
2	great	154258
3	place	139682
4	good	118906
5	location	104541
6	host	103797
7	clean	88872
8	london	85325
9	de	77760
10	recommend	65764

1-10 of 15 rows Previous **1** 2 Next

Hide

```
NA
```

Hide

```
# merging early and recent word counts
early_bow <- early_reviews |> count(word, sort = TRUE)
recent_bow <- recent_reviews |> count(word, sort = TRUE)
# Merge early and recent word counts
word_comparison <- full_join(early_bow, recent_bow, by = "word", suffix = c("_early", "_recen
t")) |>
  replace_na(list(n_early = 0, n_recent = 0)) |>
  mutate(diff = n_recent - n_early) |>
  arrange(desc(abs(diff)))
# View words with the biggest increase or decrease
head(word_comparison, 10)
```

	word <chr>	n_early <int>	n_recent <int>	diff <int>
1	stay	2	206807	206805
2	great	1	154258	154257
3	place	1	139682	139681
4	good	0	118906	118906
5	location	0	104541	104541
6	host	0	103797	103797
7	clean	0	88872	88872
8	london	0	85325	85325
9	de	0	77760	77760
10	recommend	0	65764	65764

1-10 of 10 rows

Hide

```
NA
```

Observations

- The most frequently used words in recent reviews relate to positive experiences such as stay, great and clean
- Earlier reviews had significantly fewer occurrences of these words, likely due to a smaller dataset size or different writing styles.
- The increase in mentions of “location” suggests that guests have placed more emphasis on geographical convenience over time.
- The presence of the word “London” among the most used words in recent reviews indicates a location specific pattern in the dataset.

Hide

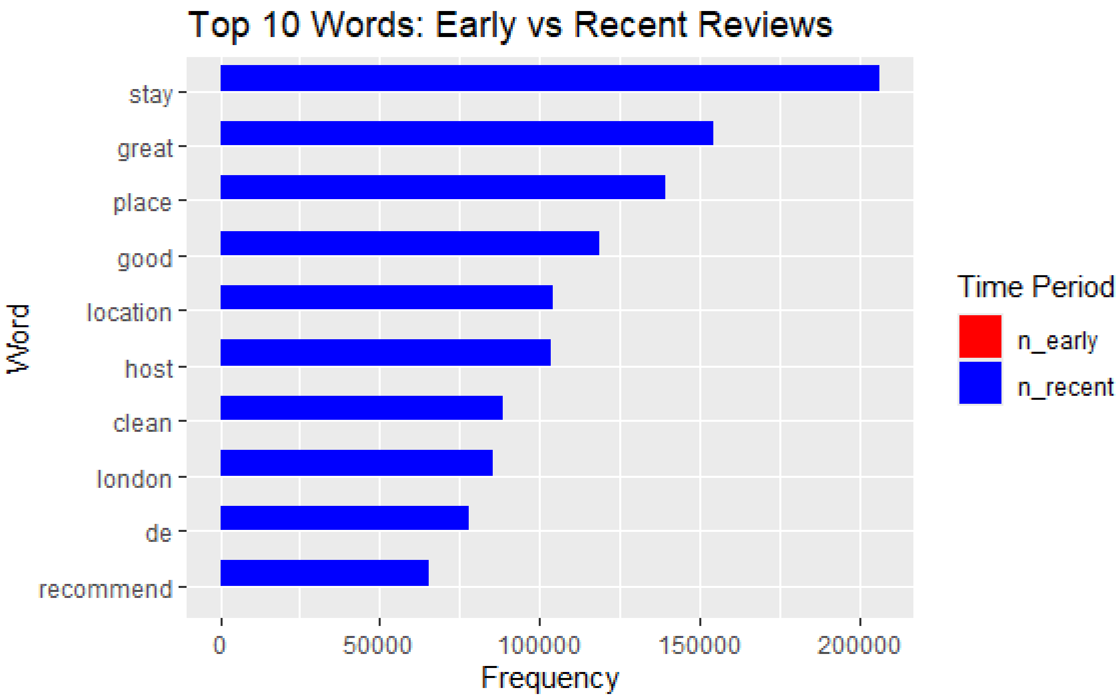
```
# Select top 10 words with the biggest absolute change
top_words <- word_comparison |>
  slice_max(order_by = abs(diff), n = 10) |>
  pivot_longer(cols = c(n_early, n_recent), names_to = "time_period", values_to = "count")


top_words
```

word <chr>	diff <int>	time_period <chr>	count <int>
stay	206805	n_early	2
stay	206805	n_recent	206807
great	154257	n_early	1
great	154257	n_recent	154258
place	139681	n_early	1
place	139681	n_recent	139682
good	118906	n_early	0
good	118906	n_recent	118906
location	104541	n_early	0
location	104541	n_recent	104541

1-10 of 20 rows Previous **1** 2 Next

Hide

```
#  bar chart
ggplot(top_words, aes(x = reorder(word, count), y = count, fill = time_period)) +
  geom_col(position = "dodge") +
  coord_flip() +
  labs(title = "Top 10 Words: Early vs Recent Reviews",
       x = "Word",
       y = "Frequency",
       fill = "Time Period") +
  scale_fill_manual(values = c("n_early" = "red", "n_recent" = "blue"))
```



Hide

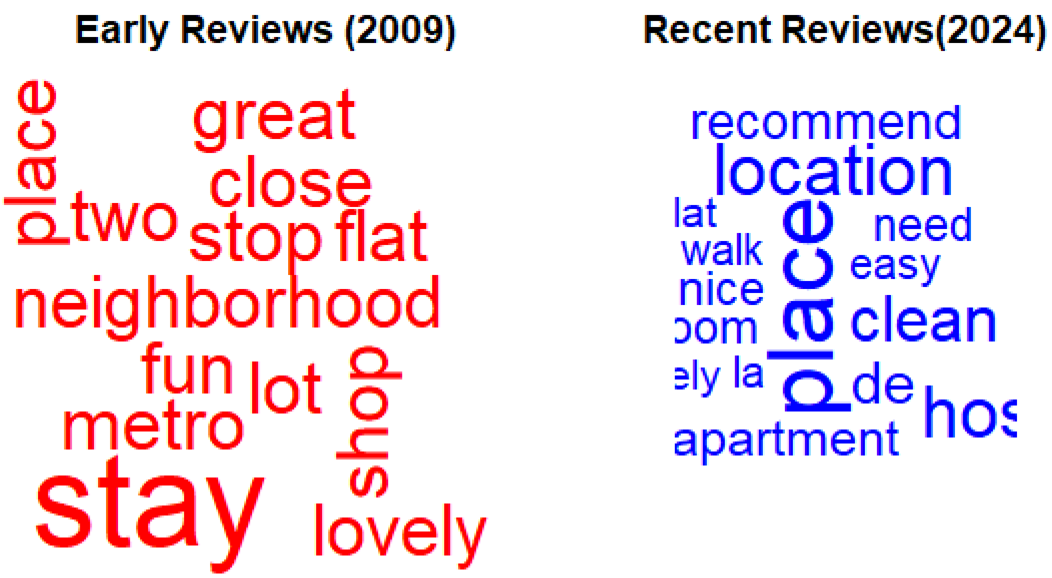
```
NA
NA
```

Hide

```


library(wordcloud)

# Setting up plotting area into two plots
par(mfrow = c(1, 2))

# Generating word clouds
wordcloud(words = early_bow$word, freq = early_bow$n, max.words = 20, colors = "red")
title("Early Reviews (2009)")
```

Hide

```


wordcloud(words = recent_bow$word, freq = recent_bow$n, max.words = 20, colors = "blue")
title("Recent Reviews(2024)")
```



Hide

```
NA
NA
```

Conclusion

- This analysis highlights how Airbnb guest's language has evolved over time.
- The increasing frequency of words like stay, great and clean suggests a growing trend in positive guest feedback.
- Reviews in 2024 are more frequent, reflecting Airbnb's popularity and changes in user behavior..