Text Analysis of Hotel Reviews from Booking.com

Elisa Cangialosi

April 29, 2019

In this report I perform text analysis of the publicly available review data posted on Booking for the major turistic italian cities: Milan, Rome, Florence Venice and Verona.

Below are some of the key findings.

The dataset comprises 12 columns, which are the following:

```r
colnames(en_reviewResponse)
```

```
##  [1] "hotelid"          "reviewer_name"    "reviewer_country"
##  [4] "review_title"     "review_date"      "stay_date"       
##  [7] "review_score"     "fullText"         "response"        
## [10] "helpfulvote"      "city"             "reviewID"
```

There are 87633 english reviews in the dataset.

Update stay date to a date format

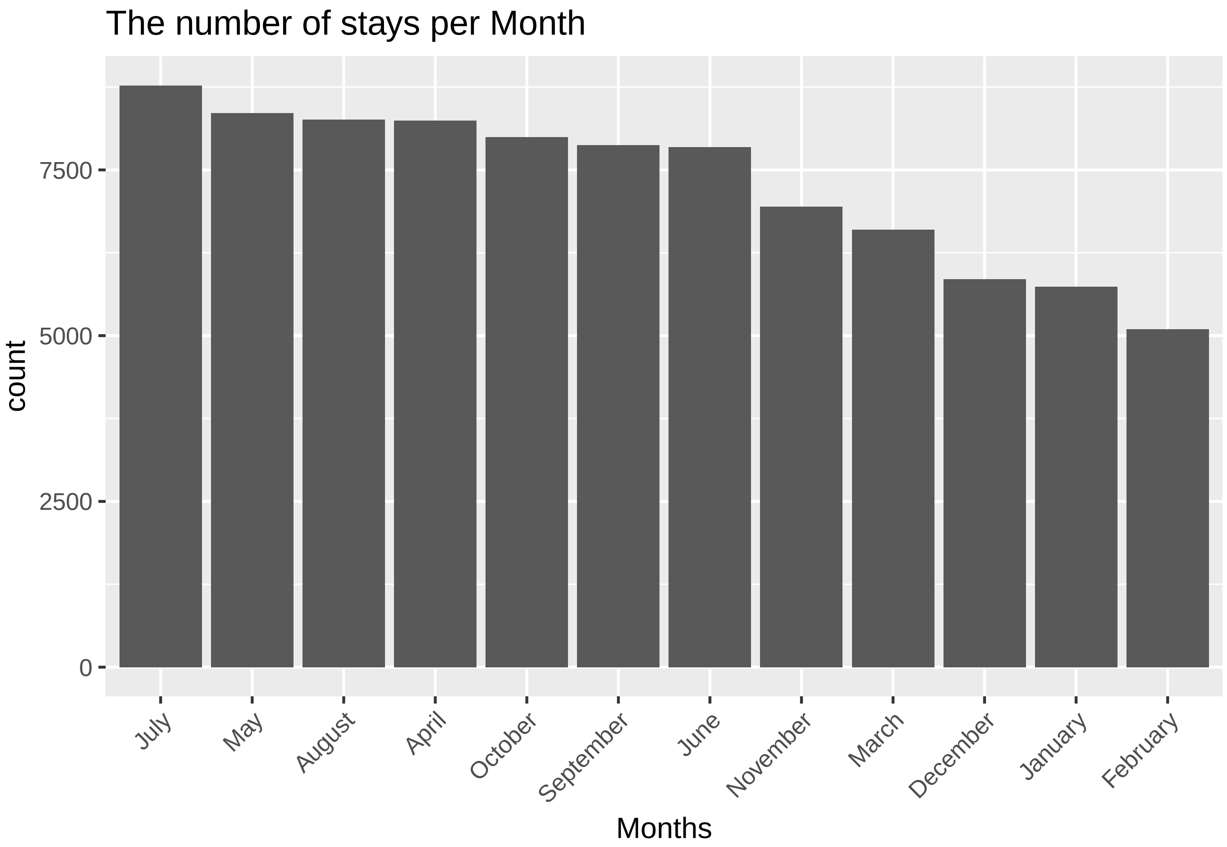
```r
toremoveStay = c("Stayed in ")
en_reviewResponse$stay_date <- gsub(paste0(toremoveStay,collapse = "|"),"", en_reviewResponse$stay_date
en_reviewResponse$stay_date <- as.Date(paste('01', en_reviewResponse$stay_date), format='%d %b %Y')
```

Update review date to a date format

```r
toremove = c("Reviewed: ")

en_reviewResponse$review_date <- gsub(paste0(toremove,collapse = "|"),"", en_reviewResponse$review_date
en_reviewResponse$review_date <- AsDate(en_reviewResponse$review_date)
```

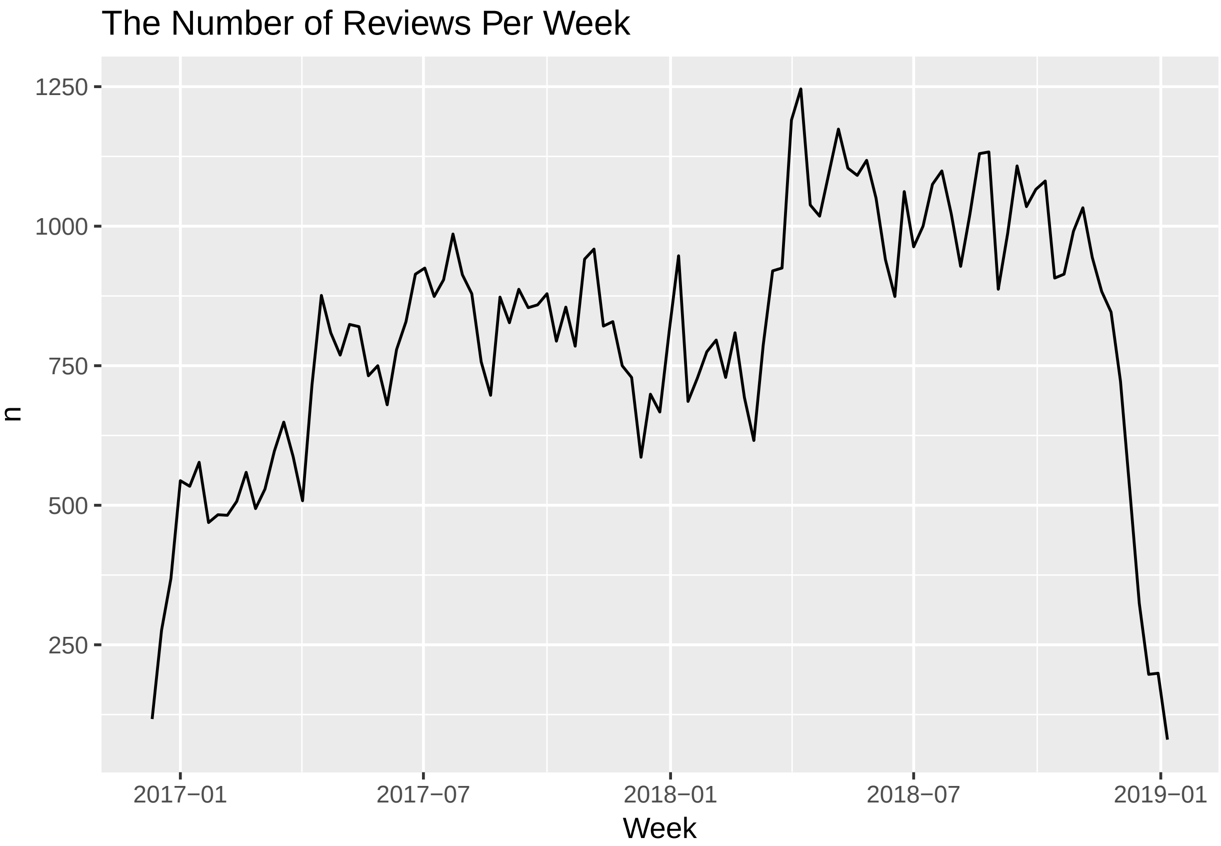
Let’s investigate which are the busiest months of the year for hotels.

```r
en_reviewResponse$monthStay <- format(en_reviewResponse$stay_date,'%B')

en_reviewResponse[!is.na(en_reviewResponse$monthStay),] %>%
  ggplot(aes(x = forcats::fct_infreq(monthStay))) +
  geom_bar() +
  scale_x_discrete(name = 'Months') +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  ggtitle('The number of stays per Month')
```



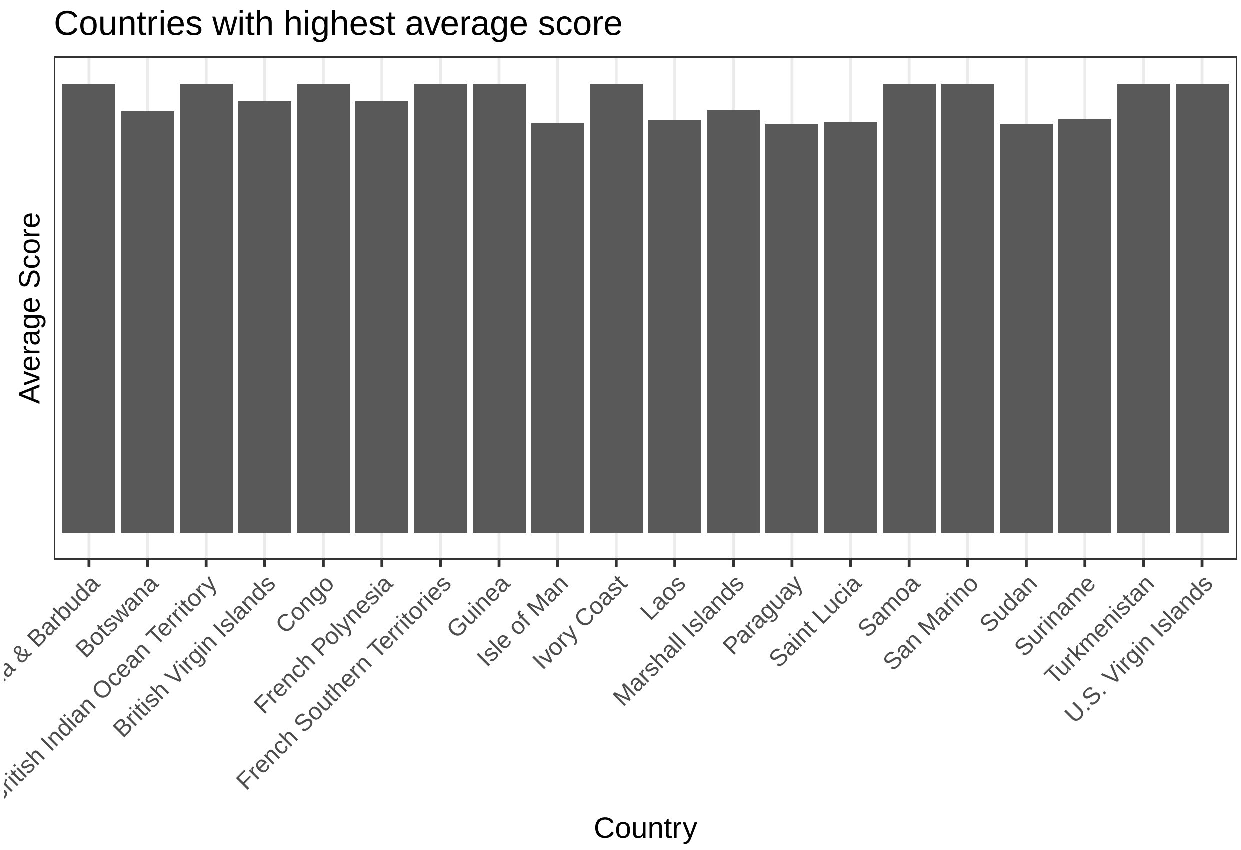
The highest number of weekly reviews was received within the half of 2018. The hotels received almost 1250 reviews in that week.

```
en_reviewResponse %>%
  count(Week = round_date(review_date, "week")) %>%
  ggplot(aes(Week, n)) +
  geom_line() +
  ggtitle('The Number of Reviews Per Week')
```



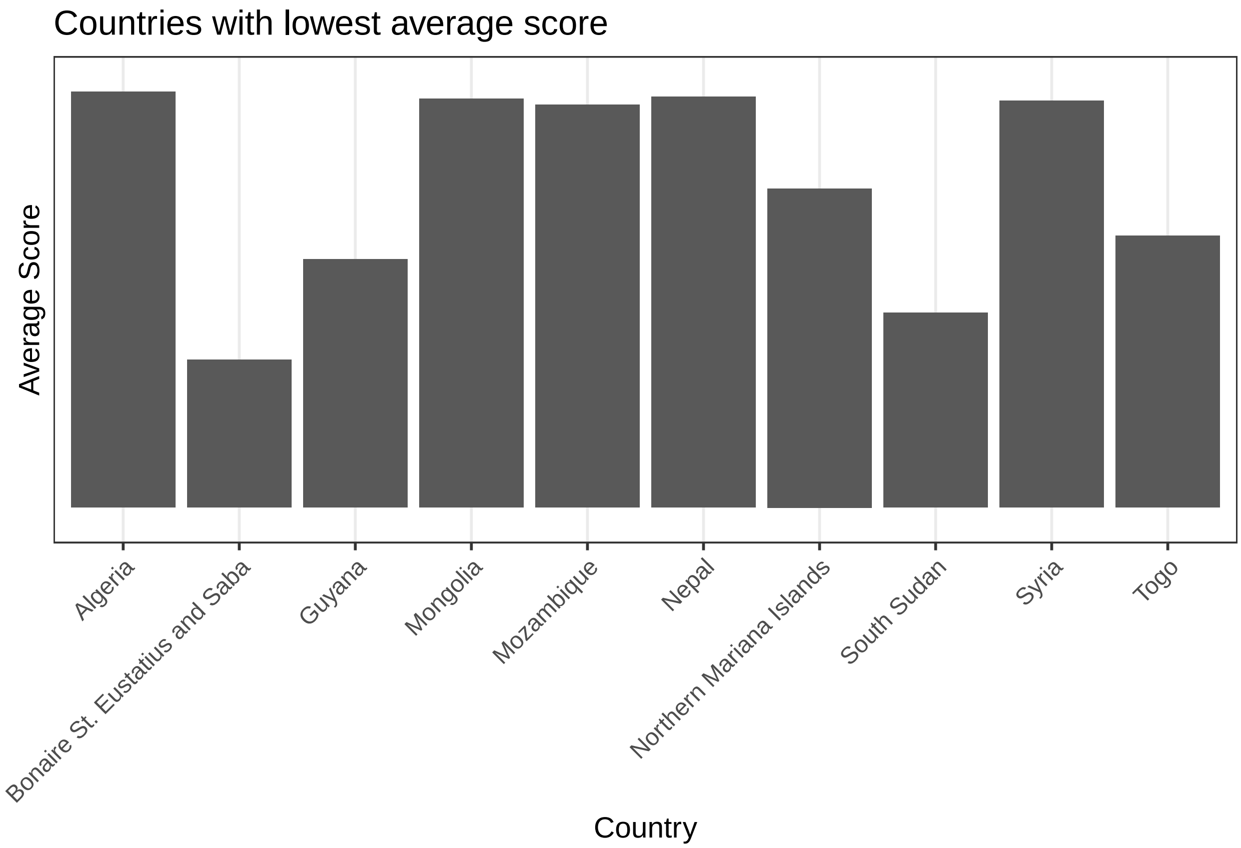
Countries with highest average score

```
avgscore_nation <- sqldf('SELECT reviewer_country, avg(review_score) as avg_score from en_reviewRespons

ggplot(avgscore_nation[1:20,],aes(x=reviewer_country, y=avg_score)) +
  geom_bar(stat = 'identity')+theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_x_discrete(name = 'Country') +
  scale_y_discrete(name = 'Average Score') +
  ggtitle('Countries with highest average score')
```



Countries with lowest average score

```r
ggplot(tail(avgscore_nation, 10), aes(x=reviewer_country, y=avg_score)) +
  geom_bar(stat = 'identity')+theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  scale_x_discrete(name = 'Country') +
  scale_y_discrete(name = 'Average Score') +
  ggtitle('Countries with lowest average score')
```

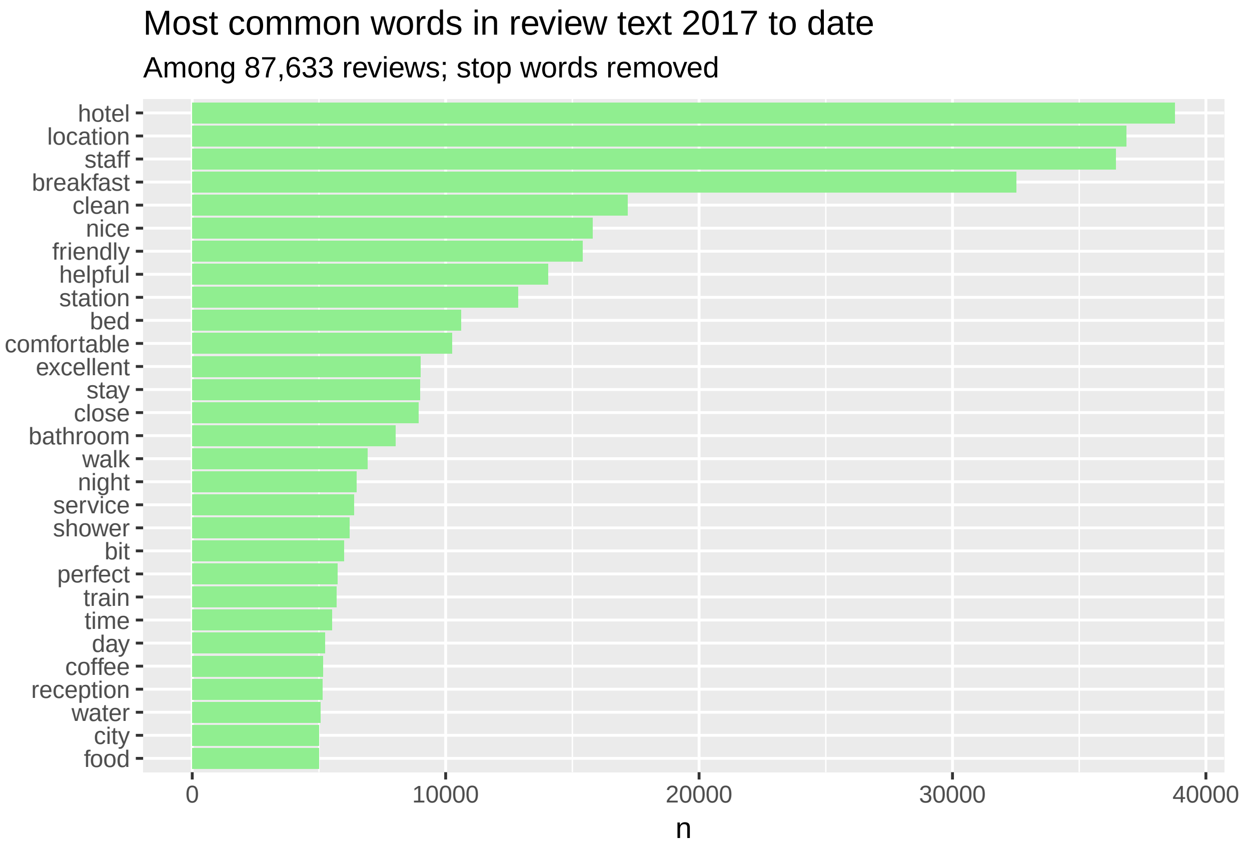


Italian Average Score Nation

```r
it_avgScore <- avgscore_nation %>% filter(reviewer_country == 'Italy')
```

What are the most commonly occuring words in English reviews?

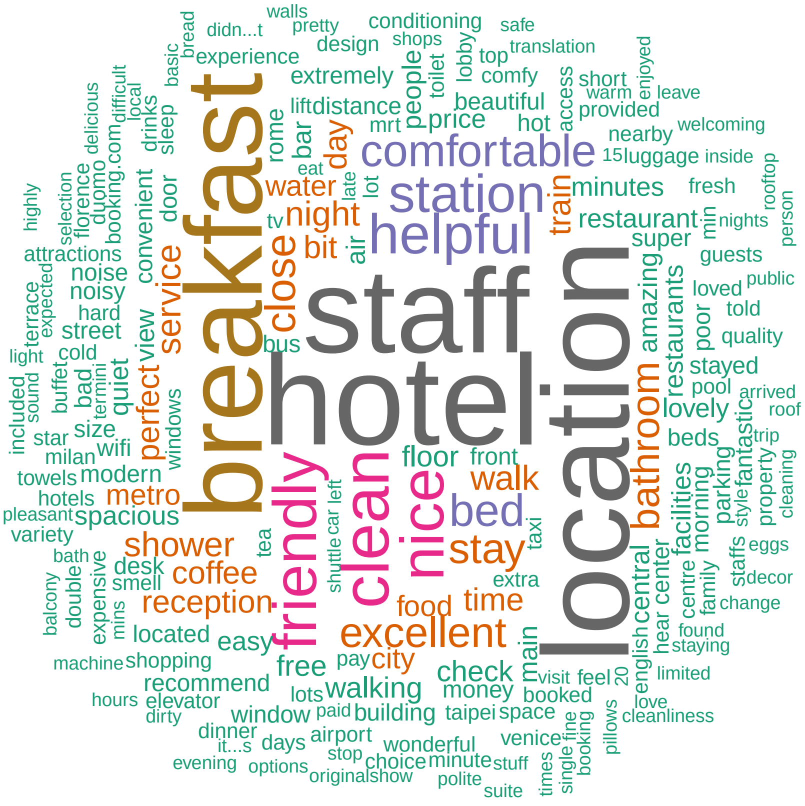
```r
review_subject <- en_reviewResponse %>%
  unnest_tokens(word, fullText, token = "ngrams", n = 1) %>%
  anti_join(stop_words)

my_stopwords <- data_frame(word = c(as.character(1:10)))
review_subject <- review_subject %>%
  anti_join(my_stopwords)

review_subject %>%
  count(word, sort = TRUE) %>%
  filter(n > 5000) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
  geom_col(fill = "lightgreen") +
  xlab(NULL) +
  coord_flip() +
  labs(title = "Most common words in review text 2017 to date",
       subtitle = "Among 87,633 reviews; stop words removed")
```



The importance of words can be illustrated in a wordcloud.

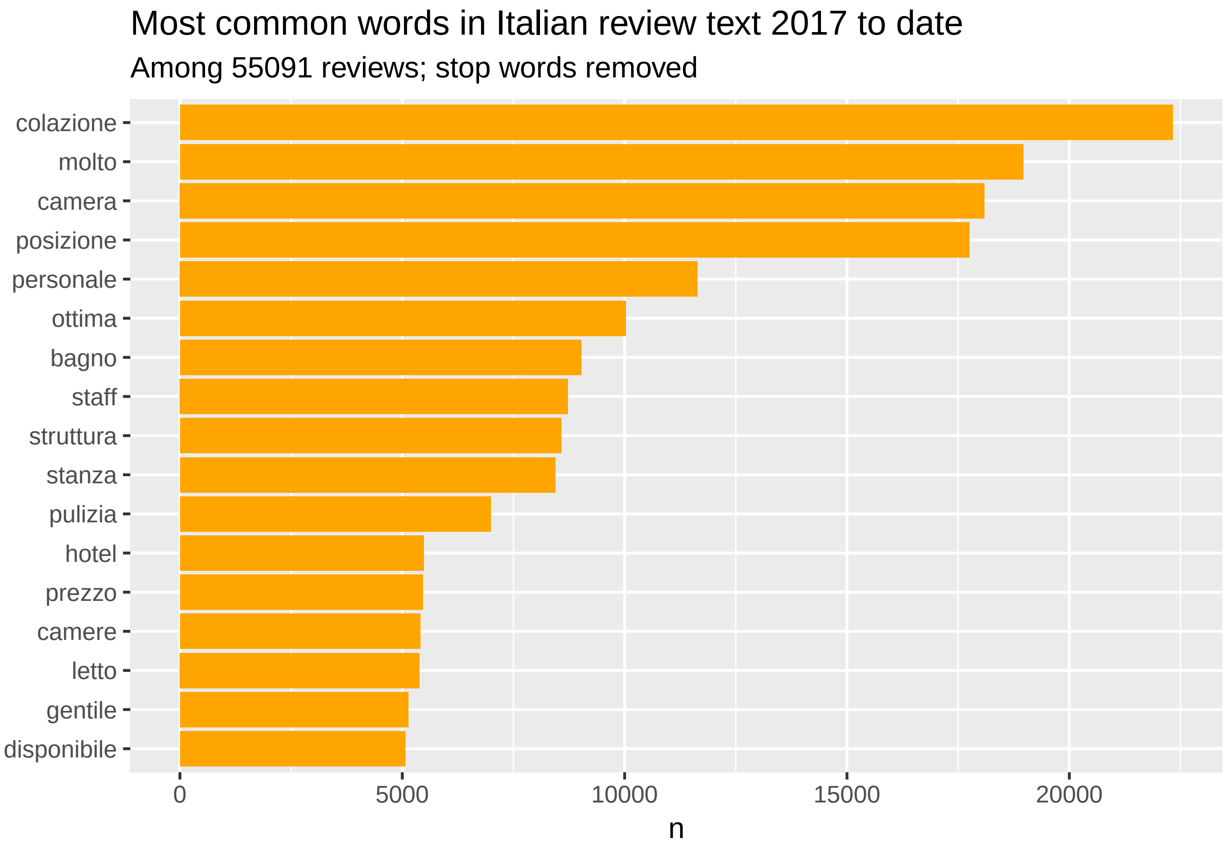
The word cloud clearly shows that "hotel", "location", "breakfast" and "staff" are the four most important words in Booking reviews in italian tourist cities.

```
freqWords <- review_subject %>% count(word, sort = TRUE)
set.seed(1234)
wordcloud(words = freqWords$word, freq = freqWords$n, min.freq = 1000,
          max.words=400, random.order=FALSE, rot.per=0.35,
          colors=brewer.pal(8, "Dark2"))
```



What are the most commonly occuring words in Italian reviews?

```
it_review_subject <- it_reviewResponse %>%
  unnest_tokens(word, fullText, token = "ngrams", n = 1)

it_review_subject <- it_review_subject %>%
  filter(!word %in% stopwords("italian")) %>%
  anti_join(my_stopwords)

it_review_subject %>%
  count(word, sort = TRUE) %>%
  filter(n > 5000) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
  geom_col(fill = "orange") +
  xlab(NULL) +
  coord_flip() +
  labs(title = "Most common words in Italian review text 2017 to date",
       subtitle = "Among 55091 reviews; stop words removed")
```

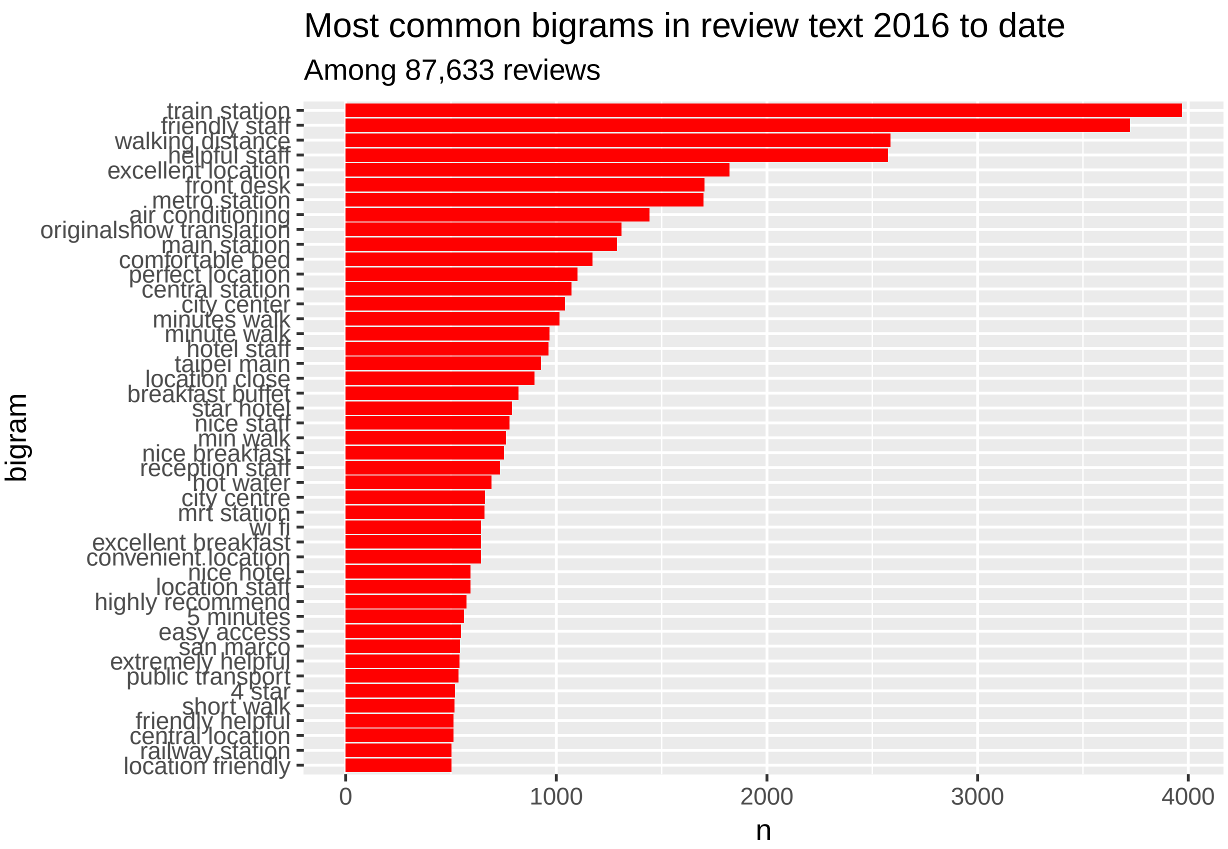


We often want to understand the relationship between words in a review. What sequences of words are common across review text? Given a sequence of words, what word is most likely to follow? What words have the strongest relationship with each other?

What are the most common bigrams in our reviews?

```r
review_bigrams <- en_reviewResponse %>%
  unnest_tokens(bigram, fullText, token = "ngrams", n = 2)

bigrams_separated <- review_bigrams %>%
  separate(bigram, c("word1", "word2"), sep = " ")
bigrams_filtered <- bigrams_separated %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word)

bigrams_united <- bigrams_filtered %>%
  unite(bigram, word1, word2, sep = " ")
bigrams_united %>%
  count(bigram, sort = TRUE)
```

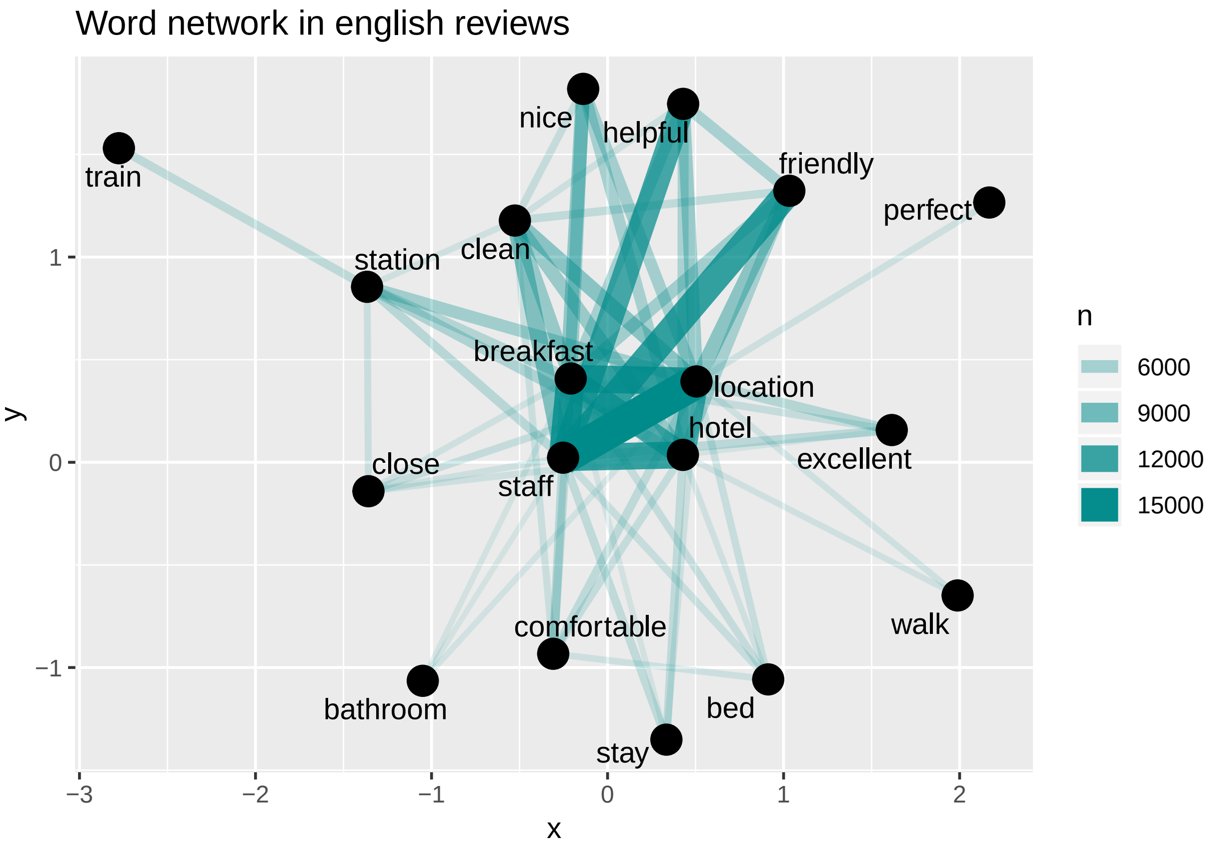
```r
bigrams_united %>%
  count(bigram, sort = TRUE) %>%
  filter(n > 500) %>%
  mutate(bigram = reorder(bigram, n)) %>%
  ggplot(aes(bigram, n)) +
  geom_col(fill = "red") +
  coord_flip() +
  labs(title = "Most common bigrams in review text 2016 to date",
       subtitle = "Among 87,633 reviews")
```



The above visualizes the common bigrams in English reviews, showing those that occurred at least 500 times and where neither word was a stop-word.

Visualize bigrams in word networks:

```
title_word_pairs <- review_subject %>%
  pairwise_count(word, reviewID, sort = TRUE, upper = FALSE)

set.seed(1234)
title_word_pairs %>%
  filter(n >= 3000) %>%
  graph_from_data_frame() %>%
  ggraph(layout = "kk") +
  geom_edge_link(aes(edge_alpha = n, edge_width = n), edge_colour = "cyan4") +
  geom_node_point(size = 5) +
  geom_node_text(aes(label = name), repel = TRUE,
                 point.padding = unit(0.2, "lines")) +
  ggtitle('Word network in english reviews')
```



The network graph shows strong connections between the top several words ("friendly", "staff", "excellent" and "location", "train" and "station").

Sentiment Analysis

One way to analyze the sentiment of a text is to consider the text as a combination of its individual words and the sentiment content of the whole text as the sum of the sentiment content of the individual words. Sentiment analysis can be done as an inner join. Three sentiment lexicons are available via the get_sentiments() function. Let's look at the words with a joy score from the NRC lexicon. What are the most common joy words?

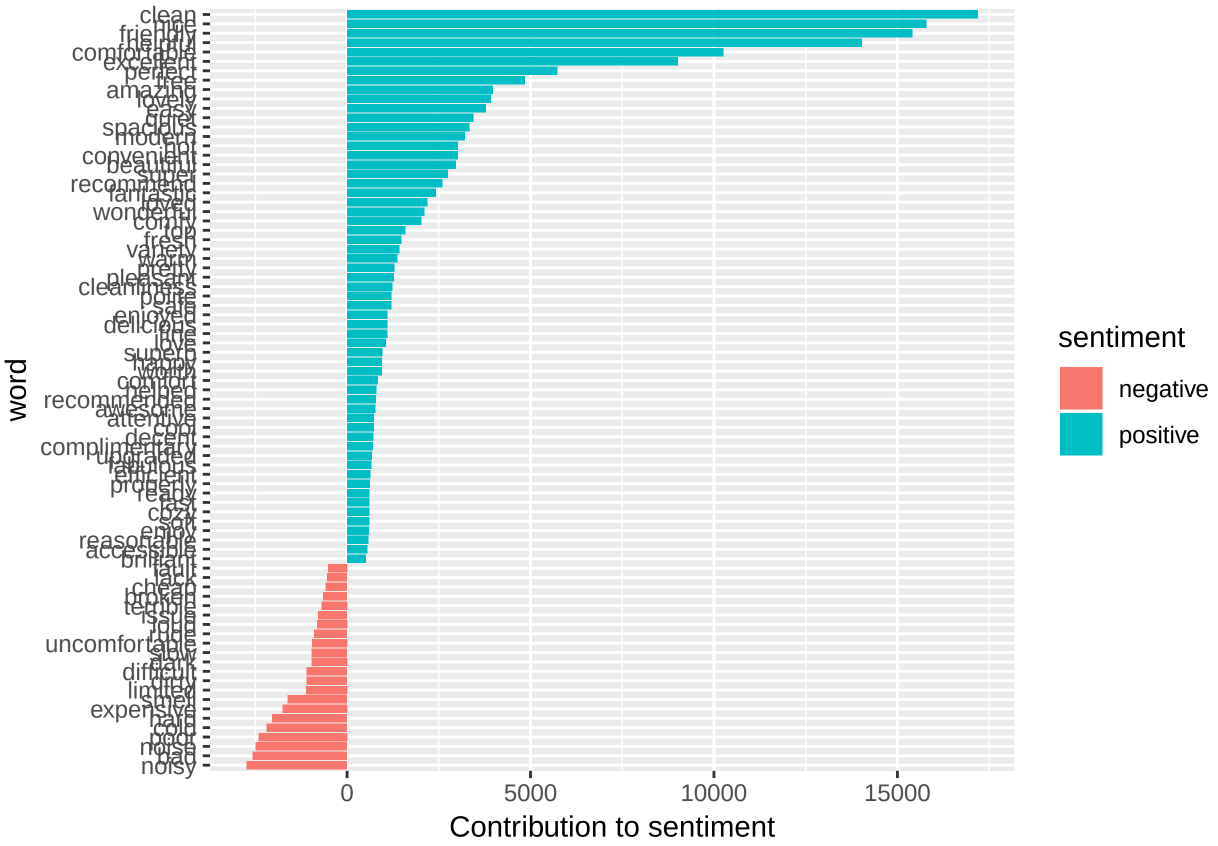
```
nrcjoy <- get_sentiments("nrc") %>%
  filter(sentiment == "joy")

review_subject %>%
  semi_join(nrcjoy) %>%
  count(word, sort = TRUE)
```

```
## # A tibble: 436 x 2
##    word          n
##    <chr>     <int>
##  1 clean     17190
##  2 friendly  15401
##  3 helpful   14035
##  4 excellent  9010
##  5 perfect    5730
##  6 food        5004
##  7 lovely      3925
##  8 money       2968
##  9 beautiful   2960
## 10 wonderful   2107
## # ... with 426 more rows
```

```
bing <- get_sentiments("bing")

bing_words <- review_subject %>%
  inner_join(bing) %>%
  count(word, sentiment, sort = TRUE)
```

Analyze word counts that contribute to each sentiment.

```
bing_words %>%
  filter(n > 500) %>%
  mutate(n = ifelse(sentiment == "negative", -n, n)) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n, fill = sentiment)) +
  geom_col() +
  coord_flip() +
  labs(y = "Contribution to sentiment")
```


clean
nice
friendly
helpful
comfortable
excellent
perfect
free
amazing
lovely
easy
quiet
spacious
modern
hot
convenient
beautiful
super
recommend
fantastic
loved
wonderful
comfy
top
fresh
variety
warm
pretty
pleasant
cleanliness
polite
safe
enjoyed
delicious
fine
love
superb
happy
worth
comfort
helped
recommended
awesome
attentive
cool
decent
complimentary
upgraded
fabulous
efficient
properly
ready
fast
cozy
soft
enjoy
reasonable
accessible
brilliant
fault
lack
cheap
broken
terrible
issue
loud
rude
uncomfortable
slow
dark
difficult
dirty
limited
smell
expensive
hard
cold
poor
noise
bad
noisy
word
Contribution to sentiment
sentiment
negative
positive