Capstone

Sarah Ahn

November 12, 2018

```
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(qdap)
```

```
## Loading required package: qdapDictionaries
```

```
## Loading required package: qdapRegex
```

```
##
## Attaching package: 'qdapRegex'
```

```
## The following object is masked from 'package:ggplot2':
##
##     %+%
```

```
## The following object is masked from 'package:dplyr':
##
##     explain
```

```
## Loading required package: qdapTools
```

```
##
## Attaching package: 'qdapTools'
```

```
## The following object is masked from 'package:dplyr':
##
##     id
```

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

```
##
## Attaching package: 'qdap'
```

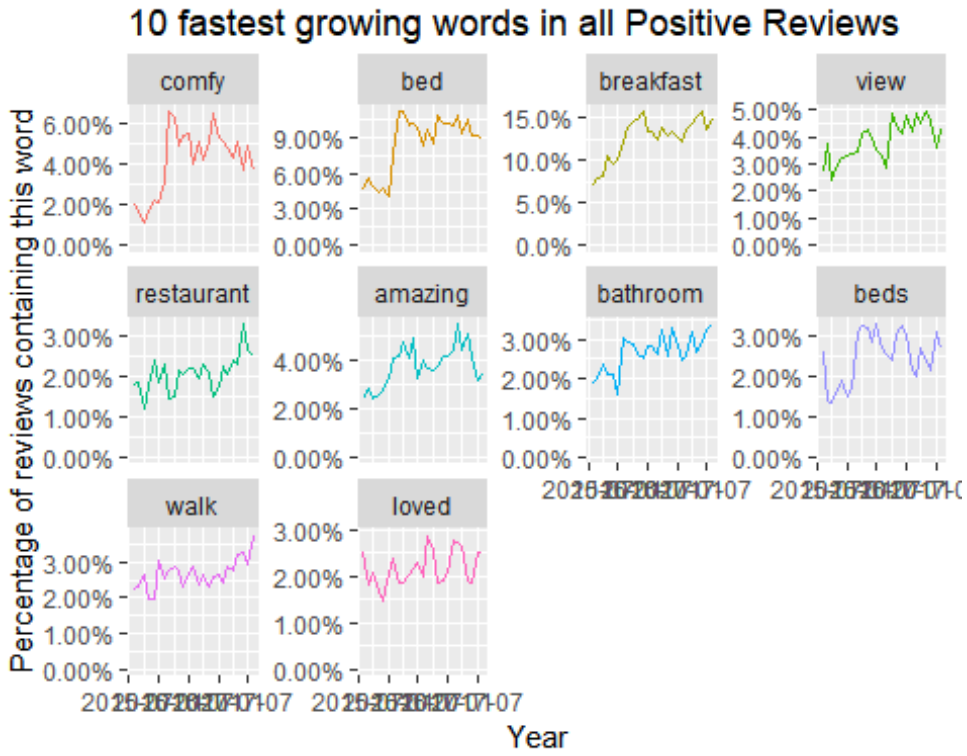
```
## The following object is masked from 'package:dplyr':
##
##     %>%

## The following object is masked from 'package:base':
##
##     Filter

library(tidyr)

##
## Attaching package: 'tidyr'

## The following object is masked from 'package:qdap':
##
##     %>%

library(tidytext)
library(tm)

## Loading required package: NLP

##
## Attaching package: 'NLP'

## The following object is masked from 'package:qdap':
##
##     ngrams

## The following object is masked from 'package:ggplot2':
##
##     annotate

##
## Attaching package: 'tm'

## The following objects are masked from 'package:qdap':
##
##     as.DocumentTermMatrix, as.TermDocumentMatrix

library(SnowballC)
library(stringr)

##
## Attaching package: 'stringr'

## The following object is masked from 'package:qdap':
##
##     %>%

library(wordcloud)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following object is masked from 'package:base':
##
##     date

library(broom)
library(scales)
library(LSAfun)

## Loading required package: lsa

## Loading required package: rgl

##
## Attaching package: 'rgl'

## The following object is masked from 'package:qdap':
##
##     %>%

library(lsa)
library(topicmodels)
library(sentimentr)
library(purrr)

##
## Attaching package: 'purrr'

## The following object is masked from 'package:LSAfun':
##
##     compose

## The following object is masked from 'package:scales':
##
##     discard

## The following object is masked from 'package:qdap':
##
##     %>%

#Loading data
df <- read.csv("C:/Users/sarah ahn/Documents/Chang School/CAPSTONE/Text
Analysis/Dataset/515k-hotel-reviews-data-in-europe/Hotel_Reviews.csv")

#limit dataset and remove irrelevant columnes
df[, c("Hotel_Address", "Additional_Number_of_Scoring",
"Average_Score","Reviewer_Nationality", "days_since_review", "lat", "lng")]
<- NULL

hotel_name <- df %>%
  group_by(Hotel_Name) %>%
```

```
  summarise(count=n())
hotel_name <- data.frame(hotel_name)
hotel_name <- hotel_name[order(hotel_name$count, decreasing=T), ]
hotel_name <- hotel_name[1:20,]

df_reduced <- df %>%
  filter(Hotel_Name %in% hotel_name$Hotel_Name)

sum(is.na(df_reduced))

## [1] 0

#Date cleaning

df_reduced$Review_Date <- as.Date(df_reduced$Review_Date, "%m/%d/%Y")
df_reduced[ ,c("Negative_Review", "Positive_Review")] <- lapply(df_reduced[
,c("Negative_Review", "Positive_Review")], as.character)
df_reduced$Tags <- as.character(df_reduced$Tags)
df_reduced <- df_reduced %>%
  mutate(id = seq_along(Positive_Review)) %>%
  mutate(month = round_date(Review_Date, "month"))


#Date pre-processing

data("stop_words")

pos_review <- df_reduced %>%
  unnest_tokens(word, Positive_Review) %>%
  anti_join(stop_words) %>%
  count(word, sort=T) %>%
  ungroup()

## Joining, by = "word"

neg_stop <- paste(c("didn", "wasn", "bit"), collapse = '|')

neg_review <- df_reduced %>%
  unnest_tokens(word, Negative_Review) %>%
  filter(!str_detect(word, neg_stop)) %>%
  anti_join(stop_words) %>%
  count(word, sort=T) %>%
  ungroup()

## Joining, by = "word"

#Bi-grams
pos_bigrams_filtered <- df_reduced %>%
  unnest_tokens(bigram, Positive_Review, token = "ngrams", n = 2) %>%
  separate(bigram, c("word1", "word2"), sep = " ") %>%
```

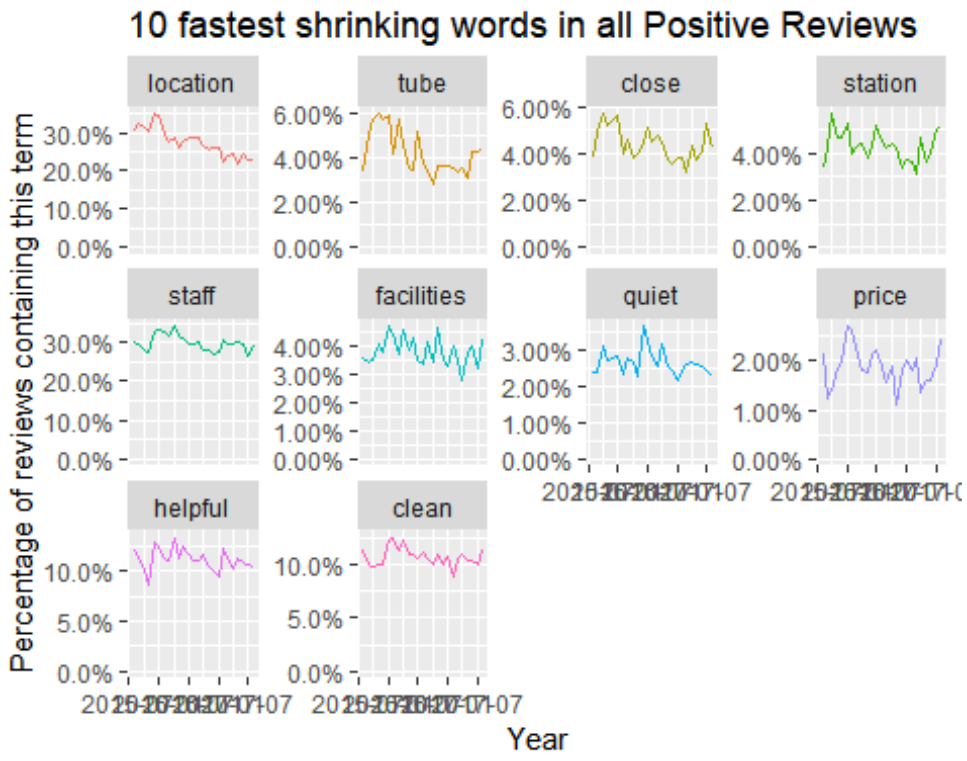
```r
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word) %>%
  drop_na() %>%
  count(word1, word2, sort = TRUE)

pos_bigrams <- pos_bigrams_filtered %>%
  unite(bigram, word1, word2, sep=" ")

df_reduced$Negative_Review <- gsub('wi|fi|wifi', 'wifi',
df_reduced$Negative_Review)
df_reduced$Negative_Review <- gsub('con|conditioning', 'conditioning',
df_reduced$Negative_Review)

neg_bigrams_filtered <- df_reduced %>%
  unnest_tokens(bigram, Negative_Review, token = "ngrams", n = 2) %>%
  separate(bigram, c("word1", "word2"), sep = " ") %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word) %>%
  drop_na() %>%
  count(word1, word2, sort = TRUE)

neg_bigrams <- neg_bigrams_filtered %>%
  unite(bigram, word1, word2, sep=" ")

#Trends in positive reviews
pos_review_month <- df_reduced %>%
  distinct(Positive_Review, .keep_all = TRUE) %>%
  unnest_tokens(word, Positive_Review, drop = FALSE) %>%
  distinct(id, word, .keep_all = TRUE) %>%
  anti_join(stop_words, by = "word") %>%
  group_by(word) %>%
  mutate(word_total = n()) %>%
  ungroup()

pos_per_month <- df_reduced %>%
  group_by(month) %>%
  summarize(month_total = n())

pos_month_count <- pos_review_month %>%
  filter(word_total >= 1000) %>%
  count(word, month) %>%
  complete(word, month, fill = list(n = 0)) %>%
  inner_join(pos_per_month, by = "month") %>%
  mutate(percent = n / month_total) %>%
  mutate(year = year(month) + yday(month) / 365)

pos_mod <- ~ glm(cbind(n, month_total - n) ~ year, ., family = "binomial")

pos_slopes <- pos_month_count %>%
```

```
  nest(-word) %>%
  mutate(model = map(data, pos_mod)) %>%
  unnest(purrr::map(model, tidy)) %>%
  filter(term == "year") %>%
  arrange(desc(estimate))

pos_slopes %>%
  head(10) %>%
  inner_join(pos_month_count, by = "word") %>%
  mutate(word = reorder(word, -estimate)) %>%
  ggplot(aes(month, n / month_total, color = word)) +
  geom_line(show.legend = FALSE) +
  scale_y_continuous(labels = scales::percent_format()) +
  facet_wrap(~ word, scales = "free_y") +
  expand_limits(y = 0) +
  labs(x = "Year",
       y = "Percentage of reviews containing this word",
       title = "10 fastest growing words in all Positive Reviews")
```



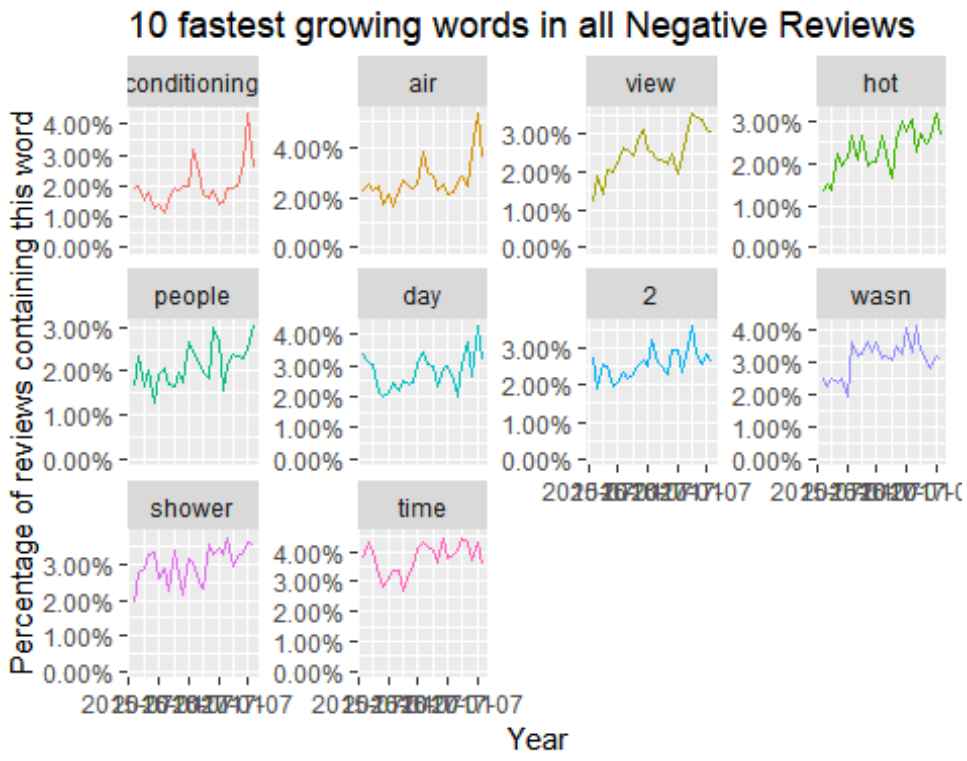

```
pos_slopes %>%
  tail(10) %>%
  inner_join(pos_month_count, by = "word") %>%
  mutate(word = reorder(word, estimate)) %>%
  ggplot(aes(month, n / month_total, color = word)) +
  geom_line(show.legend = FALSE) +
  scale_y_continuous(labels = scales::percent_format()) +
  facet_wrap(~ word, scales = "free_y") +
```

```
  expand_limits(y = 0) +
  labs(x = "Year",
       y = "Percentage of reviews containing this term",
       title = "10 fastest shrinking words in all Positive Reviews")
```



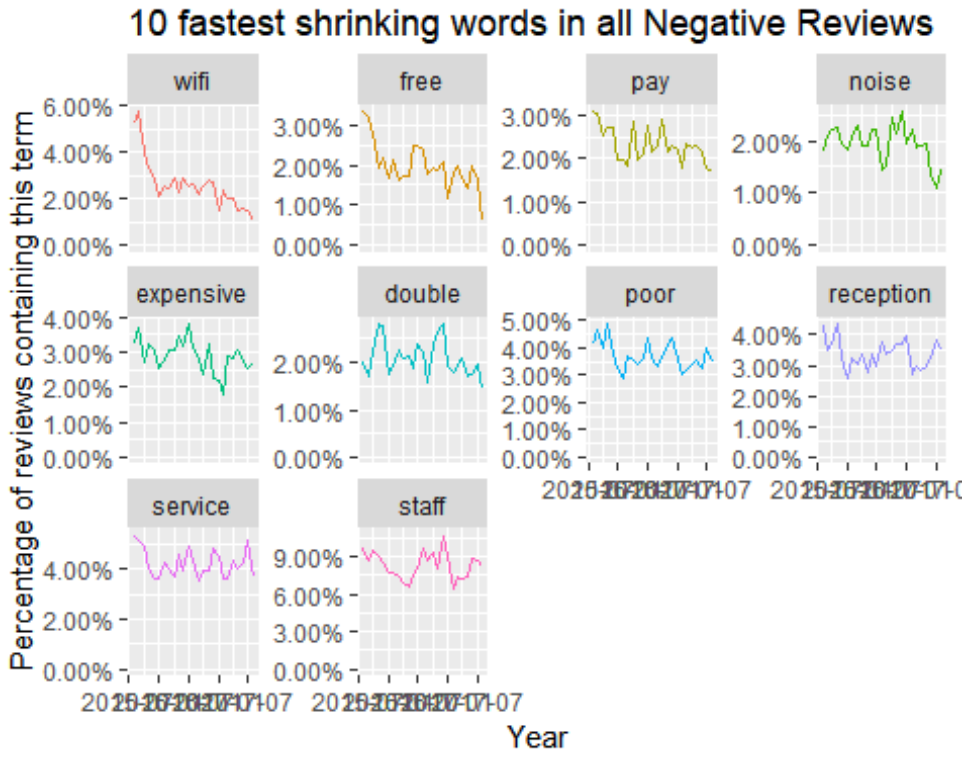

```
#Trends in Negative Reviews
neg_review_month <- df_reduced %>%
  distinct(Negative_Review, .keep_all = TRUE) %>%
  unnest_tokens(word, Negative_Review, drop = FALSE) %>%
  distinct(id, word, .keep_all = TRUE) %>%
  anti_join(stop_words, by = "word") %>%
  group_by(word) %>%
  mutate(word_total = n()) %>%
  ungroup()

neg_per_month <- df_reduced %>%
  group_by(month) %>%
  summarize(month_total = n())

neg_month_count <- neg_review_month %>%
  filter(word_total >= 1000) %>%
  count(word, month) %>%
  complete(word, month, fill = list(n = 0)) %>%
  inner_join(pos_per_month, by = "month") %>%
  mutate(percent = n / month_total) %>%
  mutate(year = year(month) + yday(month) / 365)
```

```r
neg_mod <- ~ glm(cbind(n, month_total - n) ~ year, ., family = "binomial")

neg_slopes <- neg_month_count %>%
  nest(-word) %>%
  mutate(model = purrr::map(data, pos_mod)) %>%
  unnest(purrr::map(model, tidy)) %>%
  filter(term == "year") %>%
  arrange(desc(estimate))

neg_slopes %>%
  head(10) %>%
  inner_join(neg_month_count, by = "word") %>%
  mutate(word = reorder(word, -estimate)) %>%
  ggplot(aes(month, n / month_total, color = word)) +
  geom_line(show.legend = FALSE) +
  scale_y_continuous(labels = scales::percent_format()) +
  facet_wrap(~ word, scales = "free_y") +
  expand_limits(y = 0) +
  labs(x = "Year",
       y = "Percentage of reviews containing this word",
       title = "10 fastest growing words in all Negative Reviews")
```



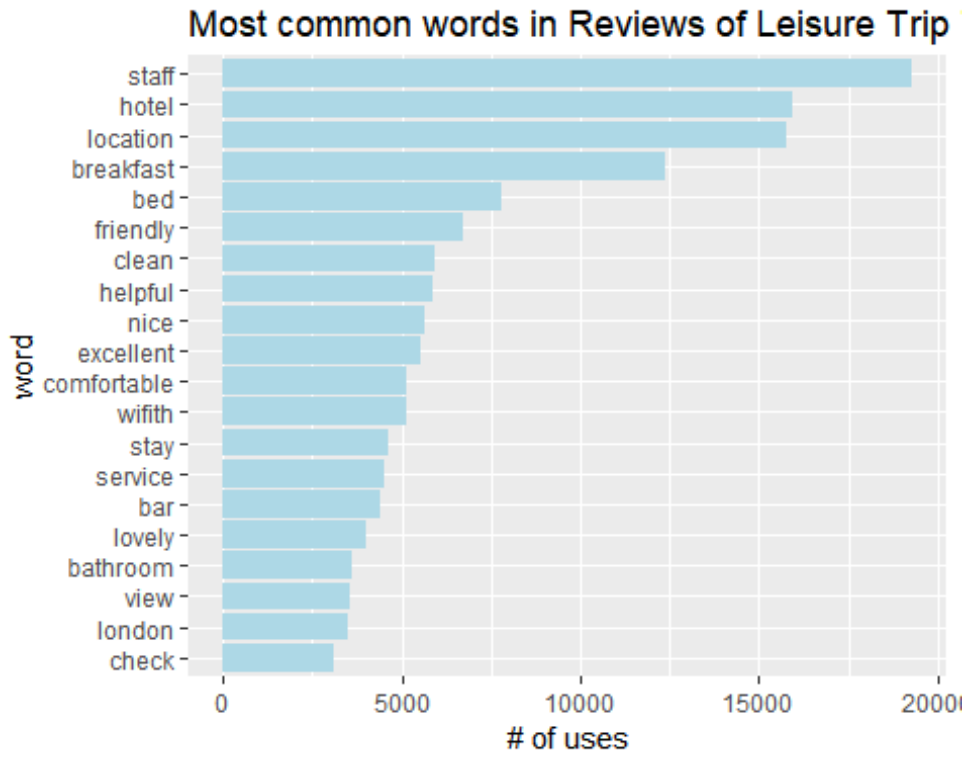

```r
neg_slopes %>%
  tail(10) %>%
  inner_join(neg_month_count, by = "word") %>%
```

```
  mutate(word = reorder(word, estimate)) %>%
  ggplot(aes(month, n / month_total, color = word)) +
  geom_line(show.legend = FALSE) +
  scale_y_continuous(labels = scales::percent_format()) +
  facet_wrap(~ word, scales = "free_y") +
  expand_limits(y = 0) +
  labs(x = "Year",
       y = "Percentage of reviews containing this term",
       title = "10 fastest shrinking words in all Negative Reviews")
```



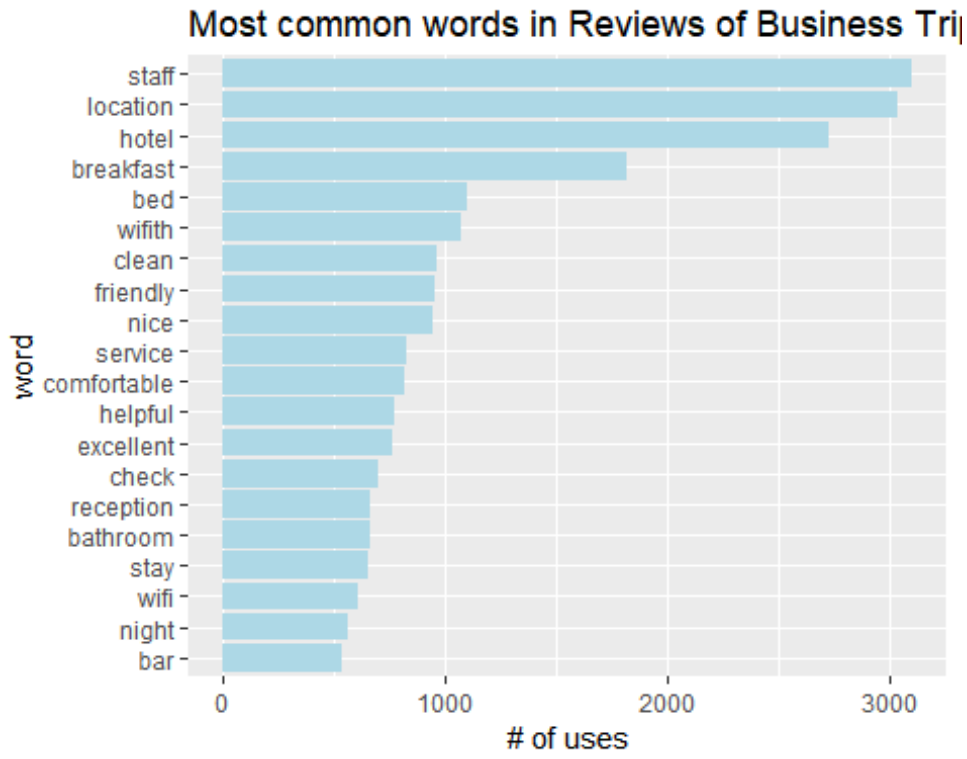

```
#Leisure Reviews
df_leisure <- df_reduced[with(df_reduced, str_detect(Tags, 'Leisure')),]


all_leisure <- data.frame(
  reviews = c(df_leisure$Positive_Review, df_leisure$Negative_Review),
  month = c(df_leisure$month, df_leisure$month)
)

all_leisure$reviews <- as.character(all_leisure$reviews)

all_leisure_review <- all_leisure %>%
  unnest_tokens(word, reviews) %>%
  filter(!str_detect(word, 'positive|negative|didn')) %>%
  anti_join(stop_words) %>%
```

```
  count(word, sort=T) %>%
  ungroup()

## Joining, by = "word"

all_leisure_review %>%
  head(20) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
  geom_col(fill = "lightblue") +
  scale_y_continuous() +
  coord_flip() +
  labs(title = "Most common words in Reviews of Leisure Trip Travellers",
       y = "# of uses")
```



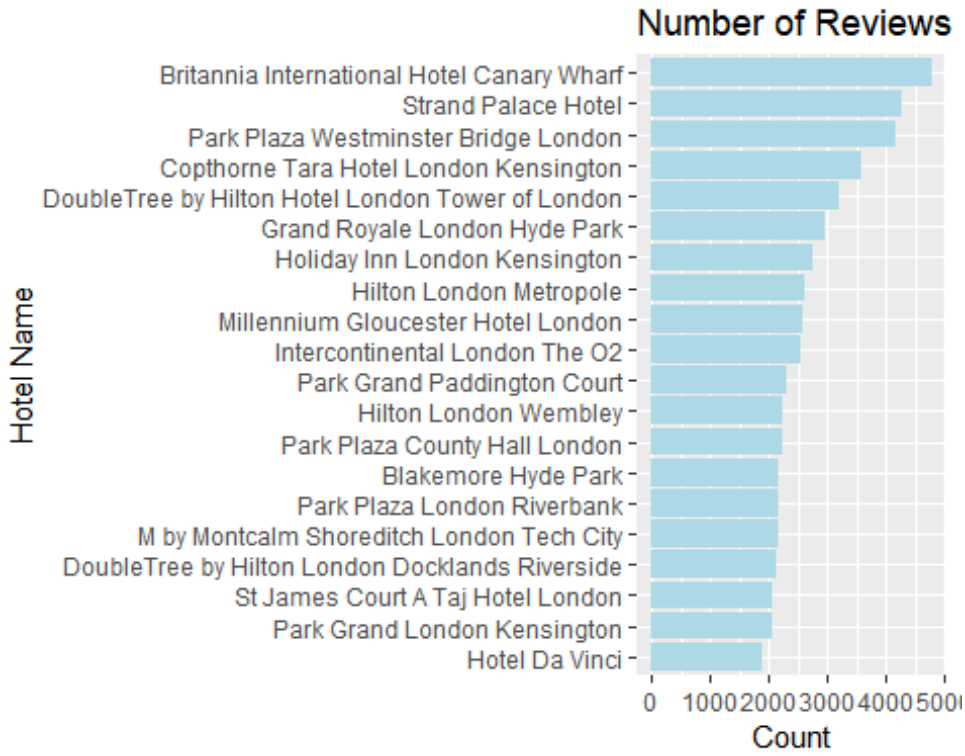

```
#Business Reviews
df_business <- df_reduced[with(df_reduced, str_detect(Tags, 'Business')),]

all_business <- data.frame(
  reviews = c(df_business$Positive_Review, df_business$Negative_Review),
  month = c(df_business$month, df_business$month)
)

all_business$reviews <- as.character(all_business$reviews)

all_business_review <- all_business %>%
  unnest_tokens(word, reviews) %>%
```

```
  filter(!str_detect(word, 'positive|negative|didn|london')) %>%
  anti_join(stop_words) %>%
  count(word, sort=T) %>%
  ungroup()

## Joining, by = "word"

all_business_review %>%
  head(20) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
  geom_col(fill = "lightblue") +
  scale_y_continuous() +
  coord_flip() +
  labs(title = "Most common words in Reviews of Business Trip Travellers",
       y = "# of uses")
```

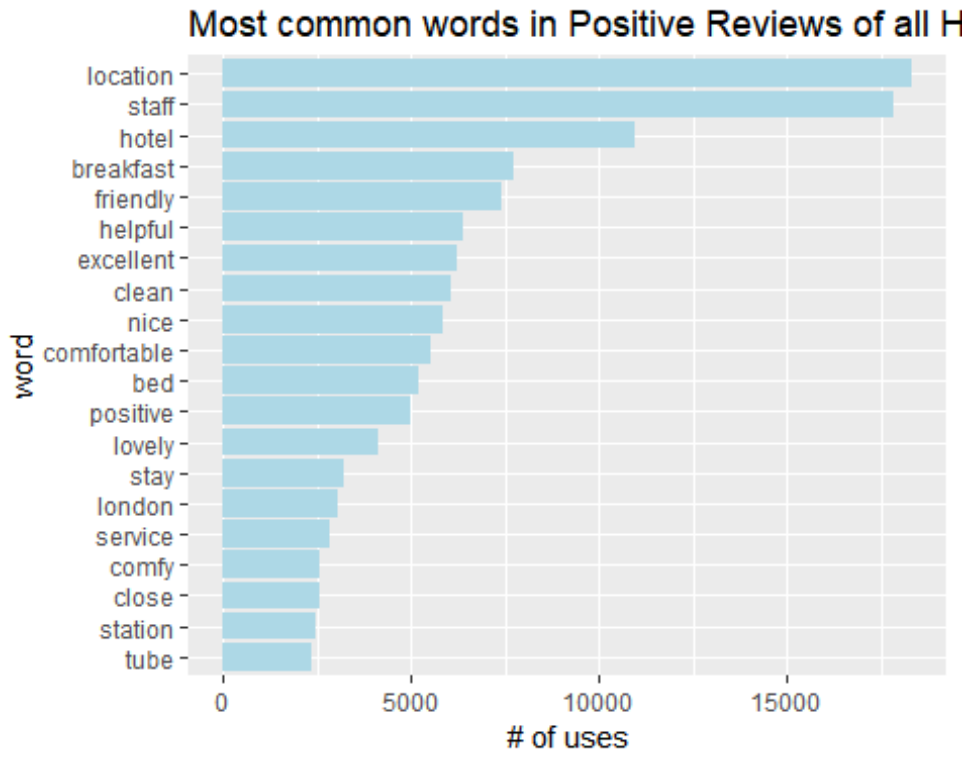



```
#Data Visualization
min(df_reduced$Review_Date)

## [1] "2015-08-04"

max(df_reduced$Review_Date)

## [1] "2017-08-03"

hotel_name %>%
  mutate(Hotel_Name = reorder(Hotel_Name, count)) %>%
```

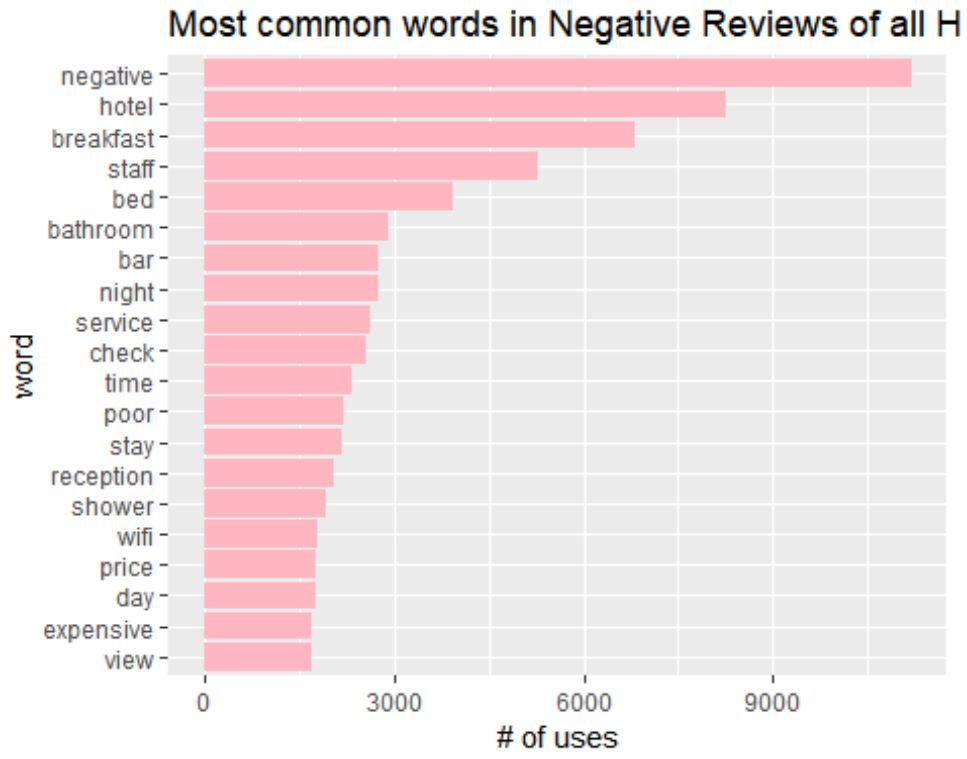
```
  ggplot(aes(Hotel_Name, count)) +
  geom_col(fill = "lightblue") +
  scale_y_continuous() +
  coord_flip() +
  labs(x = 'Hotel Name',title = "Number of Reviews per Hotel", y = 'Count')
```



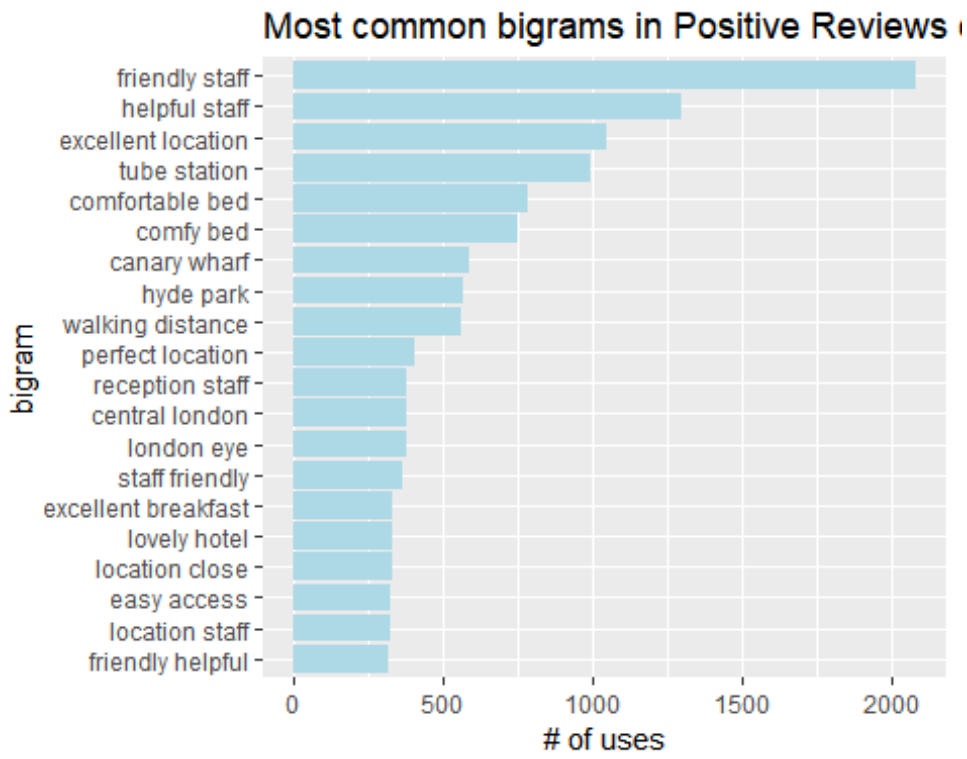

```
#Word frequency of postive and negative reviews
pos_review %>%
  head(20) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
    geom_col(fill = "lightblue") +
    scale_y_continuous() +
    coord_flip() +
    labs(title = "Most common words in Positive Reviews of all Hotels",
         y = "# of uses")
```



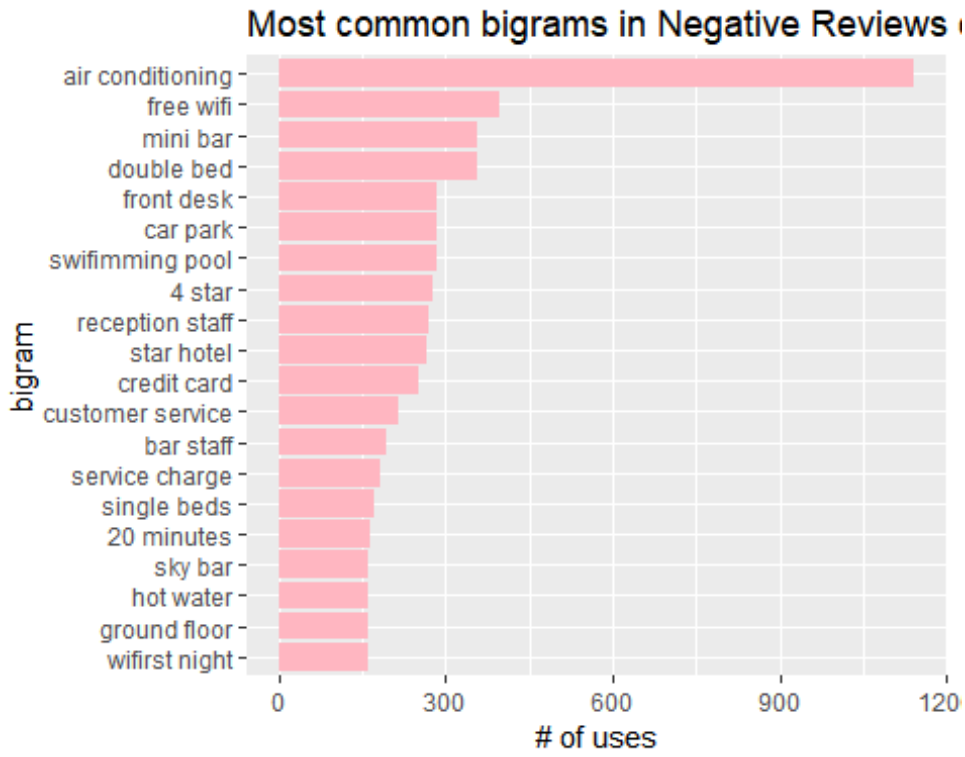

```
neg_review %>%
  head(20) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
    geom_col(fill = "lightpink") +
    scale_y_continuous() +
    coord_flip() +
    labs(title = "Most common words in Negative Reviews of all Hotels",
         y = "# of uses")
```




```r
#bigram visualization
pos_bigrams %>%
  head(20) %>%
  mutate(bigram = reorder(bigram, n)) %>%
  ggplot(aes(bigram, n)) +
  geom_col(fill = "lightblue") +
  scale_y_continuous() +
  coord_flip() +
  labs(title = "Most common bigrams in Positive Reviews of all Hotels",
       y = "# of uses")
```

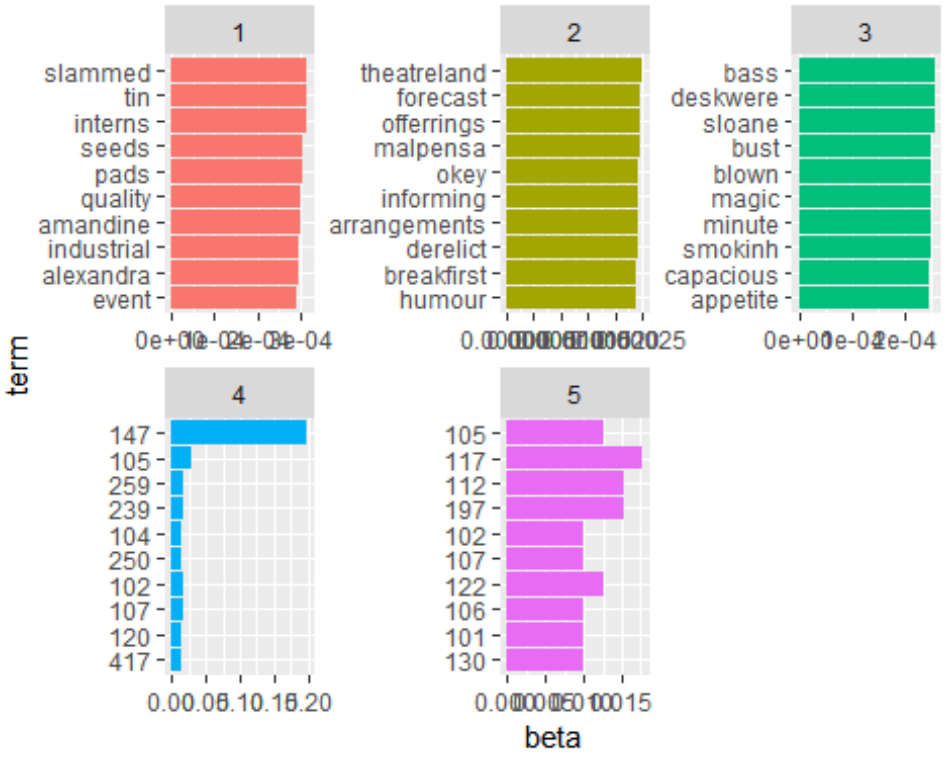



```
neg_bigrams %>%
  head(20) %>%
  mutate(bigram = reorder(bigram, n)) %>%
  ggplot(aes(bigram, n)) +
  geom_col(fill = "lightpink") +
  scale_y_continuous() +
  coord_flip() +
  labs(title = "Most common bigrams in Negative Reviews of all Hotels",
       y = "# of uses")
```



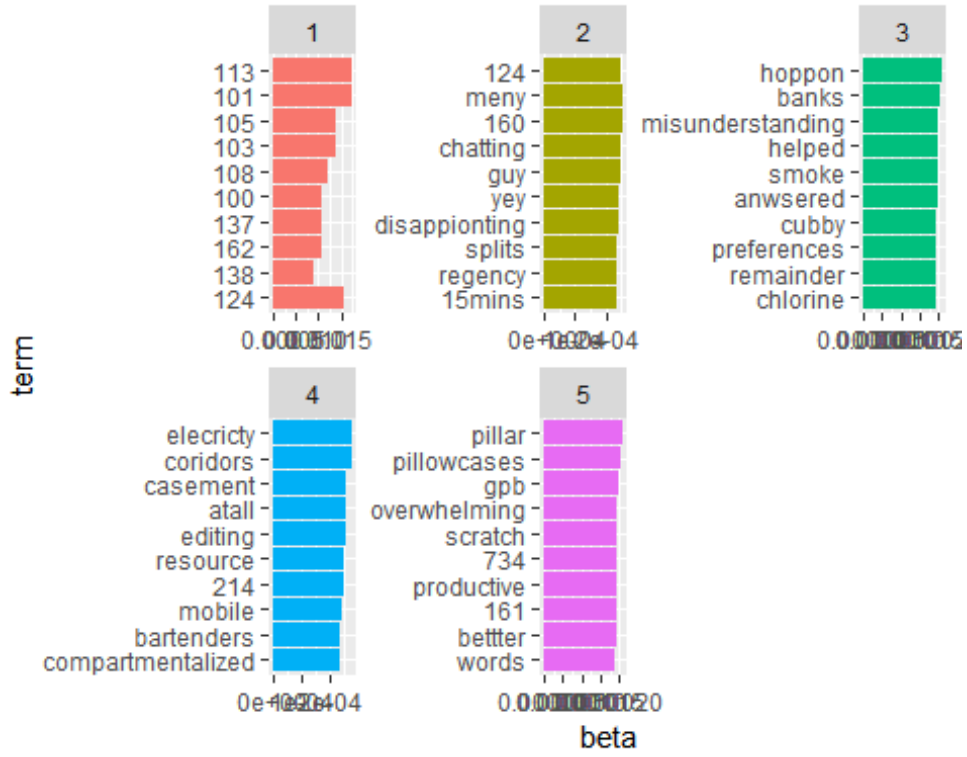

```
#Topic Modelling - used LDA

leisure_review_pos <- df_leisure %>%
  unnest_tokens(word, Positive_Review) %>%
  anti_join(stop_words) %>%
  count(word, sort=T) %>%
  ungroup()

## Joining, by = "word"

leisure_review_neg <- df_leisure %>%
  unnest_tokens(word, Negative_Review) %>%
  anti_join(stop_words) %>%
  count(word, sort=T) %>%
  ungroup()

## Joining, by = "word"

leisure_p_corp <- VCorpus(VectorSource(leisure_review_pos))
leisure_p_dtm <- DocumentTermMatrix(leisure_p_corp)
leisure_p_m <- as.matrix(leisure_p_dtm)

leisure_p_lda <- LDA(leisure_p_m, k=5, control=list(seed=1234))
leisure_p_tidy <- tidy(leisure_p_lda)

leisure_p_terms <- leisure_p_tidy %>%
  group_by(topic) %>%
```

```
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

leisure_p_terms %>%
  mutate(term = reorder(term, beta)) %>%
  group_by(topic, term) %>%
  arrange(desc(beta)) %>%
  ungroup() %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()
```




```
leisure_n_corp <- VCorpus(VectorSource(leisure_review_neg))

leisure_n_dtm <- DocumentTermMatrix(leisure_n_corp)
leisure_n_m <- as.matrix(leisure_n_dtm)
leisure_n_lda <- LDA(leisure_n_m, k=5, control=list(seed=1234))
leisure_n_tidy <- tidy(leisure_n_lda)

leisure_n_terms <- leisure_n_tidy %>%
  group_by(topic) %>%
  top_n(10, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)
```

```
leisure_n_terms %>%
  mutate(term = reorder(term, beta)) %>%
  group_by(topic, term) %>%
  arrange(desc(beta)) %>%
  ungroup() %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()
```




```
#why numbers? - trying to figure out

#Building dataframe for analysis

df_leisure2 <- df_leisure
df_leisure2 <- data.frame(
  ratings = c(df_leisure2$Reviewer_Score),
  reviews = paste(df_leisure$Positive_Review, " ",
df_leisure$Negative_Review),
  traveller_group = rep('leisure',nrow(df_leisure2))
)

df_leisure2$reviews <- as.character(df_leisure2$reviews)
df_leisure2$traveller_group <- as.character(df_leisure2$traveller_group)

df_business2 <- df_business
```

```r
df_business2 <- data.frame(
  ratings = c(df_business$Reviewer_Score),
  reviews = paste(df_business$Positive_Review, " ",
df_business$Negative_Review),
  traveller_group = rep('business', nrow(df_business2))
)
df_business2$reviews <- as.character(df_business2$reviews)
df_business2$traveller_group <- as.character(df_business2$traveller_group)

df_analysis <- data.frame(
  rating = c(df_leisure2$ratings, df_business2$ratings),
  reviews = c(df_leisure2$reviews, df_business2$reviews),
  traveller_group = c(df_leisure2$traveller_group,
df_business2$traveller_group)
)

df_analysis$reviews <- as.character(df_analysis$reviews)

#Sentiment Scoring
df_analysis <- df_analysis %>%
  mutate(id = seq_along(reviews)) %>%
  mutate(sent_scores = sentiment(get_sentences(reviews))$sentiment)

#Reression Analysis
model1 <- lm(data = df_analysis, rating ~ sent_scores + traveller_group)
summary(model1)

##
## Call:
## lm(formula = rating ~ sent_scores + traveller_group, data = df_analysis)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.1324 -0.9283  0.2127  1.1564  4.4352
##
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)
## (Intercept)              7.14945    0.01713  417.27   <2e-16 ***
## sent_scores              1.75004    0.01509  116.00   <2e-16 ***
## traveller_groupleisure  0.51536    0.01844   27.95   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.553 on 53144 degrees of freedom
## Multiple R-squared:  0.2209, Adjusted R-squared:  0.2209
## F-statistic:  7534 on 2 and 53144 DF,  p-value: < 2.2e-16
```