Capstone

Sarah Ahn

November 12, 2018

```r
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
```

```r
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
```

```r
library(tidyr)
```

```
##
## Attaching package: 'tidyr'

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

```r
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
```

```r
library(SnowballC)
library(stringr)
```

```
##
## Attaching package: 'stringr'

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

```r
library(lubridate)
```

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

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

```r
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':
##
##     %>%
```

```
library(devtools)
library(pluralize)
library(NLP)

#Loading data
df <- read.csv("C:/Users/sarah ahn/Documents/Chang School/CAPSTONE/Text Analysis/Dataset/515k-hotel-revi
```

```
#limit dataset and remove irrelevant columnes
df[, c("Hotel_Address", "Additional_Number_of_Scoring", "Average_Score","Reviewer_Nationality", "days_si

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", "Tags")] <- lapply(df_reduced[ ,c("Negative_Review
df_reduced <- df_reduced %>%
  mutate(id = seq_along(Positive_Review)) %>%
  mutate(month = round_date(Review_Date, "month"))


#Date pre-processing

data("stop_words")
```

```r
my_stopwords <- data_frame(word = c("hotel","didn", "wasn", "bit", "2", "wih"))

pos_review <- df_reduced %>%
  filter(Positive_Review != "No Positive") %>%
  mutate(Positive_Review = gsub('comfy', 'comfortable', Positive_Review)) %>%
  unnest_tokens(word, Positive_Review) %>%
  anti_join(stop_words) %>%
  anti_join(my_stopwords) %>%
  count(word, sort=T) %>%
  ungroup()
```

```
## Joining, by = "word"
## Joining, by = "word"
```

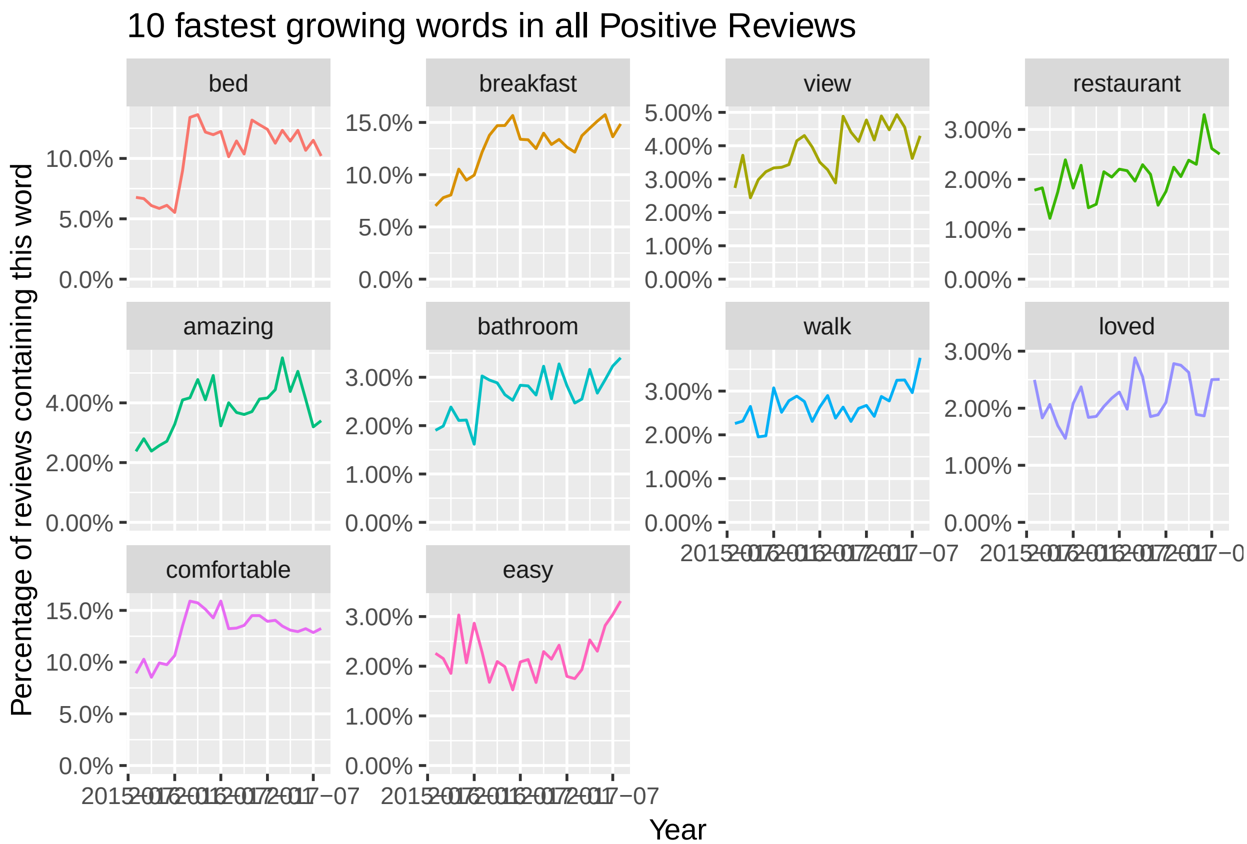
```r
neg_review <- df_reduced %>%
  filter(Negative_Review != "No Negative") %>%
  unnest_tokens(word, Negative_Review) %>%
  anti_join(my_stopwords) %>%
  anti_join(stop_words) %>%
  count(word, sort=T) %>%
  ungroup()
```

```
## Joining, by = "word"
## Joining, by = "word"
```

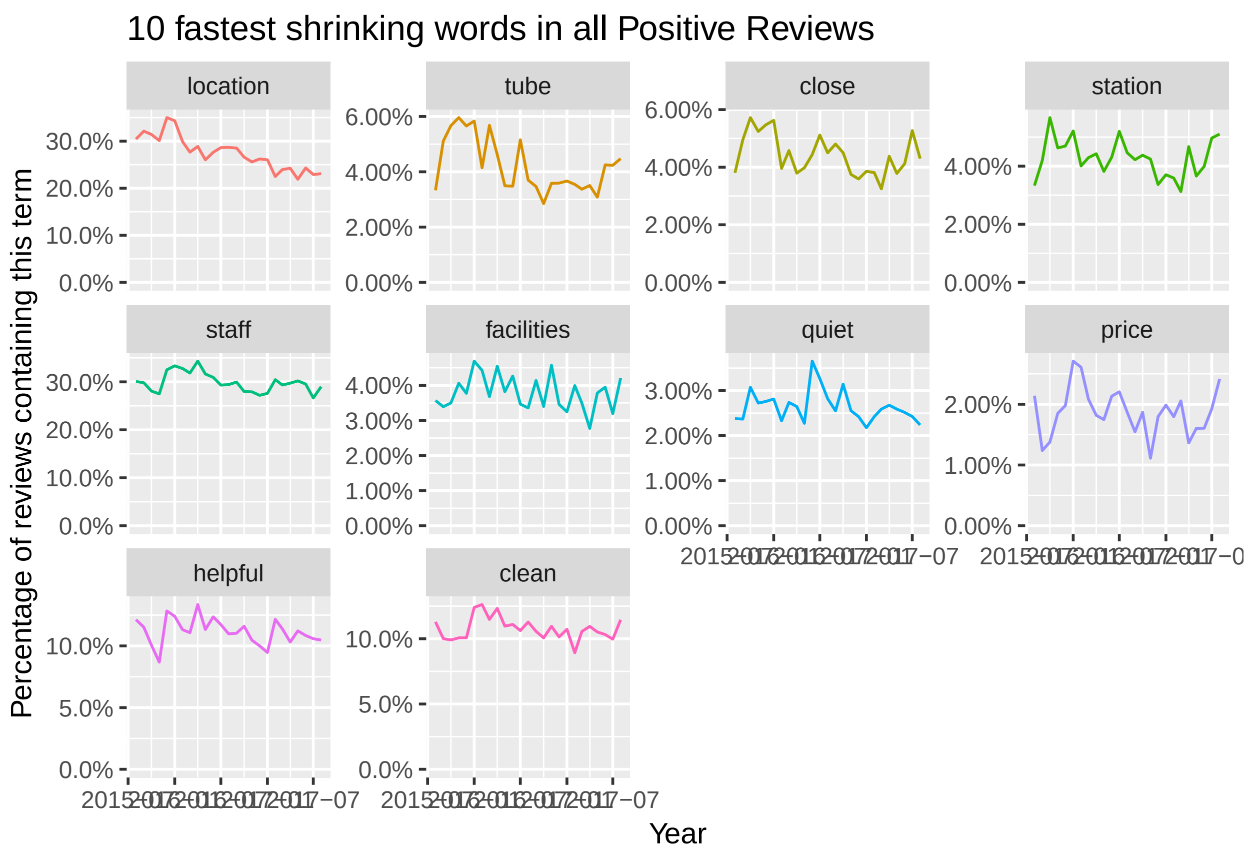
```r
#Bi-grams

pos_bigrams_filtered <- df_reduced %>%
  filter(Positive_Review != "No Positive") %>%
  mutate(Positive_Review = gsub('beds', 'bed', Positive_Review)) %>%
  unnest_tokens(bigram, Positive_Review, token = "ngrams", n = 2) %>%
  separate(bigram, c("word1", "word2"), sep = " ") %>%
  filter(!word1 %in% stop_words$word) %>%
  filter(!word2 %in% stop_words$word) %>%
  mutate(word1 = str_replace_all(word1, 'comfy', 'comfortable')) %>%
  mutate(word2 = str_replace_all(word2, 'comfy', 'comfortable')) %>%
  drop_na() %>%
  count(word1, word2, sort = TRUE)

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

df_reduced$Negative_Review <- gsub("\\<air con\\>", "air conditioning", df_reduced$Negative_Review)
df_reduced$Negative_Review <- gsub("(wi).(\\w+)", "\\1\\2", df_reduced$Negative_Review, ignore.case = T

stop_words5 <- data.frame(word = c("20", "minutes", "4", "star"))
stop_words5$word <- as.character(stop_words5$word)

neg_bigrams_filtered <- df_reduced %>%
  filter(Negative_Review != "No Negative") %>%
  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) %>%
```

```
  filter(!word1 %in% stop_words5$word) %>%
  filter(!word2 %in% stop_words5$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 %>%
  filter(Positive_Review != "No Positive") %>%
  mutate(Positive_Review = str_replace_all(Positive_Review, 'comfy', 'comfortable')) %>%
  mutate(Positive_Review = gsub('beds', 'bed', Positive_Review)) %>%
  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")
```



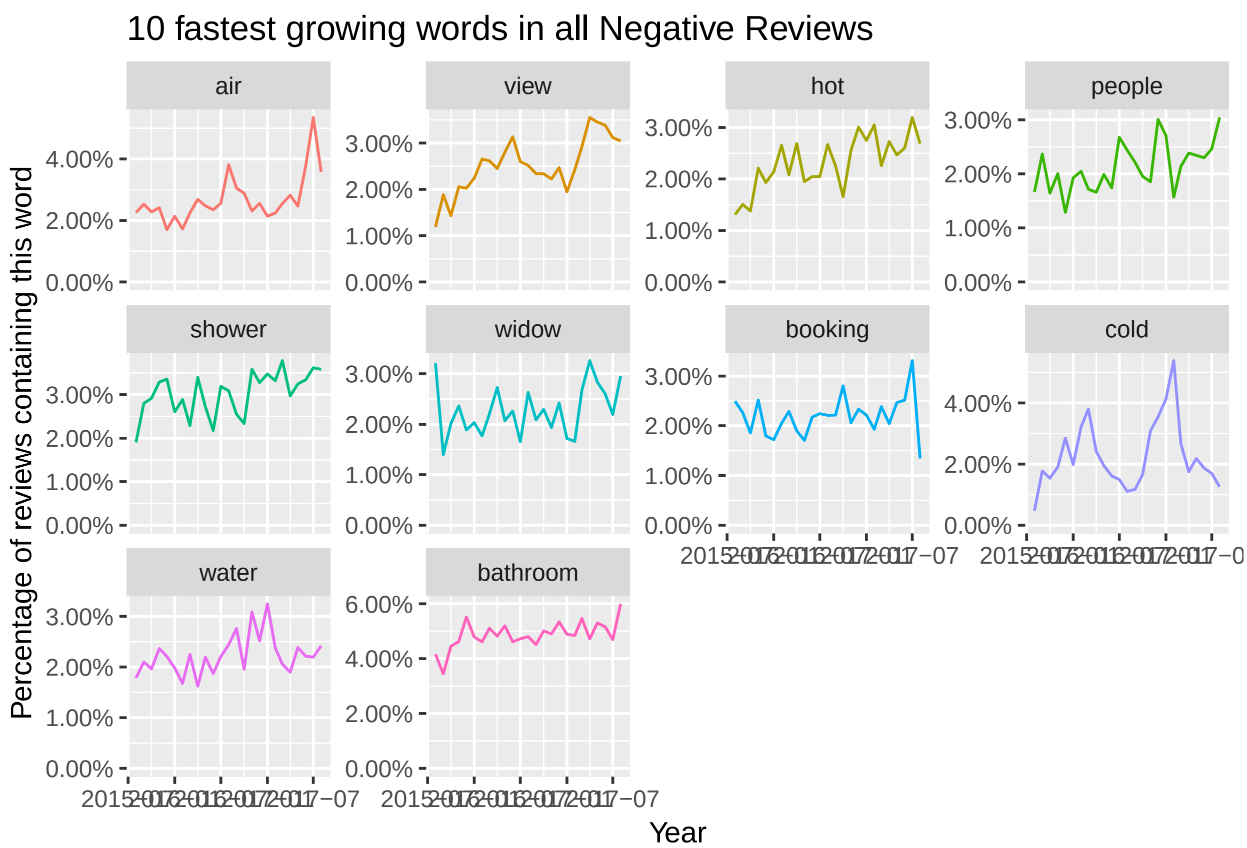

```
#Trends in Negative Reviews

neg_review_month <- df_reduced %>%
  distinct(Negative_Review, .keep_all = TRUE) %>%
  filter(Negative_Review != "No Negative") %>%
  unnest_tokens(word, Negative_Review, drop = FALSE) %>%
  distinct(id, word, .keep_all = TRUE) %>%
  anti_join(my_stopwords) %>%
  anti_join(stop_words, by = "word") %>%
  group_by(word) %>%
  mutate(word_total = n()) %>%
  ungroup()
```

```
## Joining, by = "word"
```

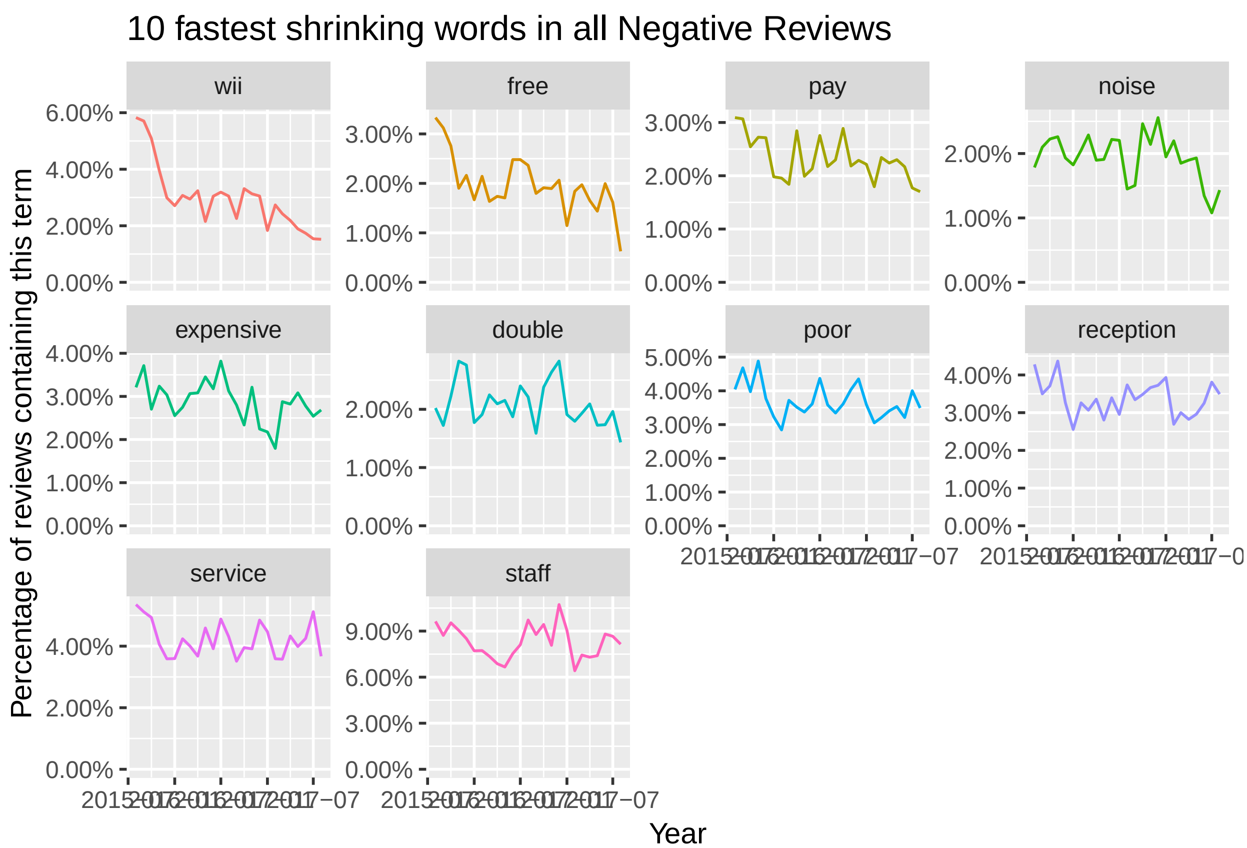
```
neg_per_month <- df_reduced %>%
  group_by(month) %>%
  summarize(month_total = n())

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))

stop_words6 <- data.frame(word = c("day", "time", "couldn", "paid"))
stop_words6$word <- as.character(stop_words6$word)

neg_slopes %>%
  anti_join(stop_words6) %>%
  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")
```

```
## Joining, by = "word"
```




```
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")
```




```r
#Leisure Reviews
df_leisure <- df_reduced[with(df_reduced, str_detect(Tags, 'Leisure')),]
df_leisure <- df_leisure[ , c("Positive_Review","Negative_Review", "month", "Reviewer_Score")]
leisure <- unite(df_leisure, "Reviews", Positive_Review, Negative_Review, sep = " ")

stop_words2 <- c("No Positive", "No Negative")
leisure$Reviews <- removeWords(leisure$Reviews, stop_words2)

#Business Reviews
df_business <- df_reduced[with(df_reduced, str_detect(Tags, 'Business')),]
df_business <- df_business[ , c("Positive_Review","Negative_Review", "month", "Reviewer_Score")]
business <- unite(df_business, "Reviews", Positive_Review, Negative_Review, sep = " ")
business$Reviews <- removeWords(business$Reviews, stop_words2)

travellers <- bind_rows(leisure %>%
                          mutate(type = "Leisure"),
                        business %>%
                          mutate(type = "Business"))
```

```r
#Travellers word frequency

stop_words4 <- data.frame(word = c("o2", "sse"))
stop_words4$word <- as.character(stop_words4$word)

travellers_freq <- travellers %>%
  mutate(Reviews = singularize(Reviews)) %>%
```

```r
  unnest_tokens(word, Reviews) %>%
  anti_join(stop_words) %>%
  anti_join(stop_words4) %>%
  group_by(type) %>%
  count(word, sort=T) %>%
  left_join(travellers %>%
              group_by(type) %>%
              summarise(total = n())) %>%
  mutate(freq = n/total)
```

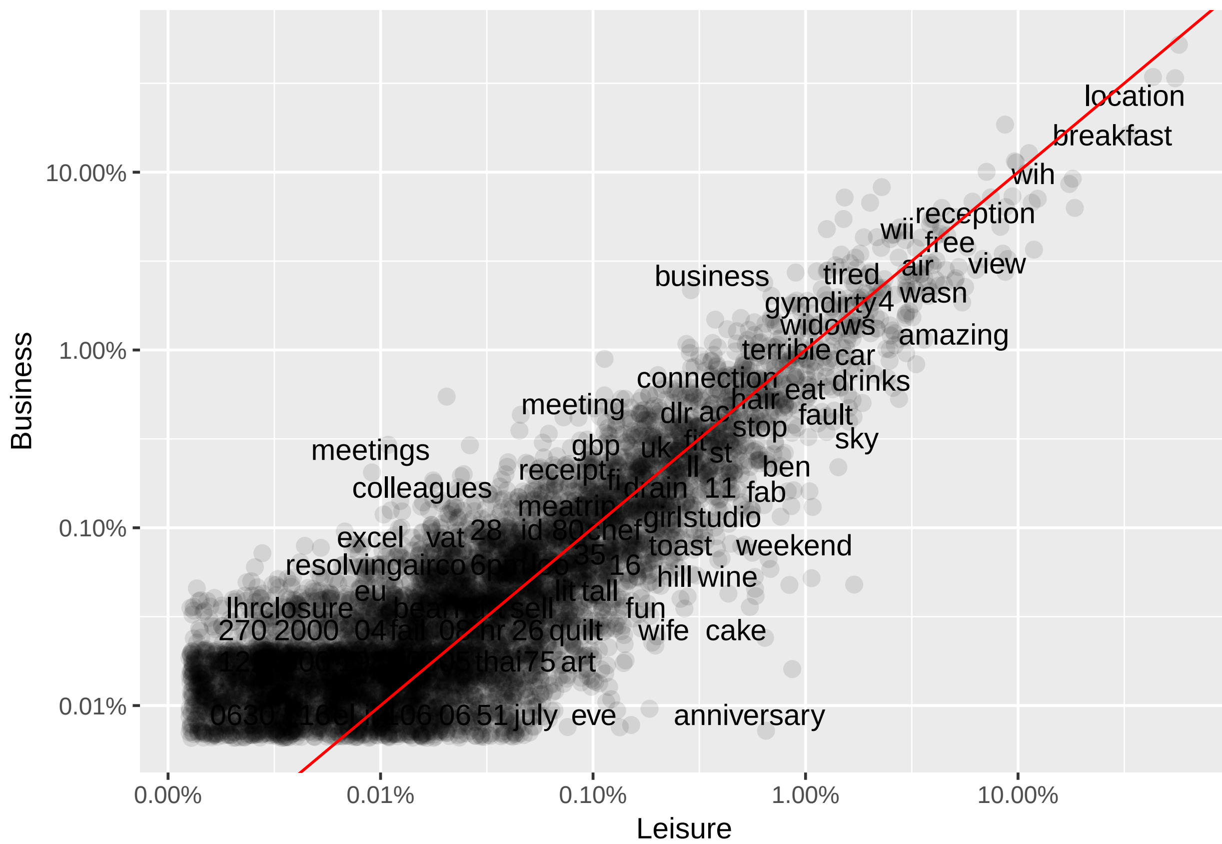
```
## Joining, by = "word"
## Joining, by = "word"
```

```
## Joining, by = "type"
```

```r
travellers_freq <- travellers_freq %>%
  select(type, word, freq) %>%
  spread(type, freq) %>%
  arrange(Leisure, Business)

ggplot(travellers_freq, aes(Leisure, Business)) +
  geom_jitter(alpha = 0.1, size = 2.5, width = 0.25, height = 0.25) +
  geom_text(aes(label = word), check_overlap = TRUE, vjust = 1.5) +
  scale_x_log10(labels = percent_format()) +
  scale_y_log10(labels = percent_format()) +
  geom_abline(color = "red")
```

```
## Warning: Removed 14997 rows containing missing values (geom_point).
```

```
## Warning: Removed 14997 rows containing missing values (geom_text).
```




```
#travellers word usage



travellers_usage <- travellers %>%
  unnest_tokens(word, Reviews) %>%
  anti_join(stop_words) %>%
  anti_join(stop_words4) %>%
  mutate(word = singularize(word)) %>%
  count(word, type) %>%
  group_by(word) %>%
  filter(sum(n) >= 10) %>%
  ungroup() %>%
  spread(type, n, fill = 0) %>%
  mutate_if(is.numeric, funs((. + 1) / (sum(.) + 1))) %>%
  mutate(logratio = log(Business / Leisure)) %>%
  arrange(desc(logratio))
```

```
## Joining, by = "word"
```

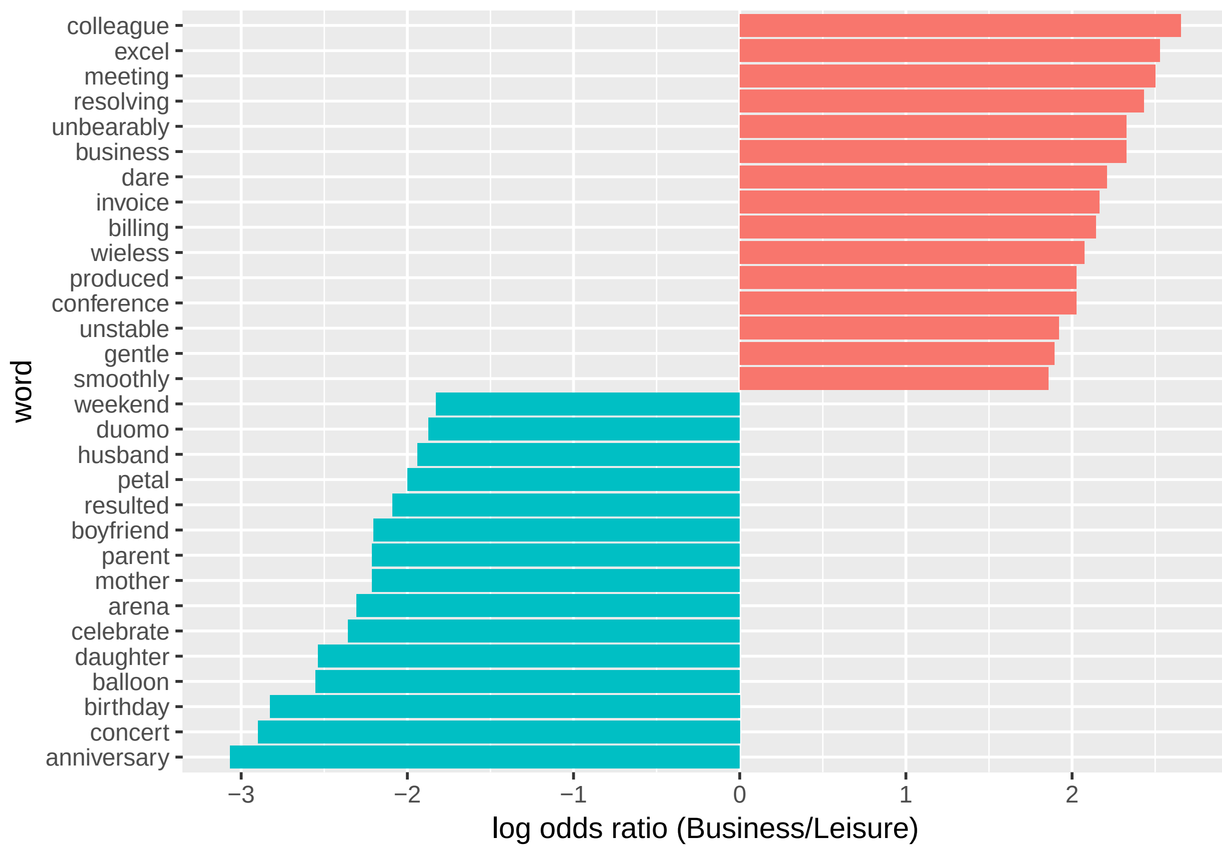
```
## Joining, by = "word"
```

```
#equally likely?
travellers_usage %>%
  arrange(abs(logratio))
```

```
## # A tibble: 3,985 x 4
##    word       Business    Leisure    logratio
```

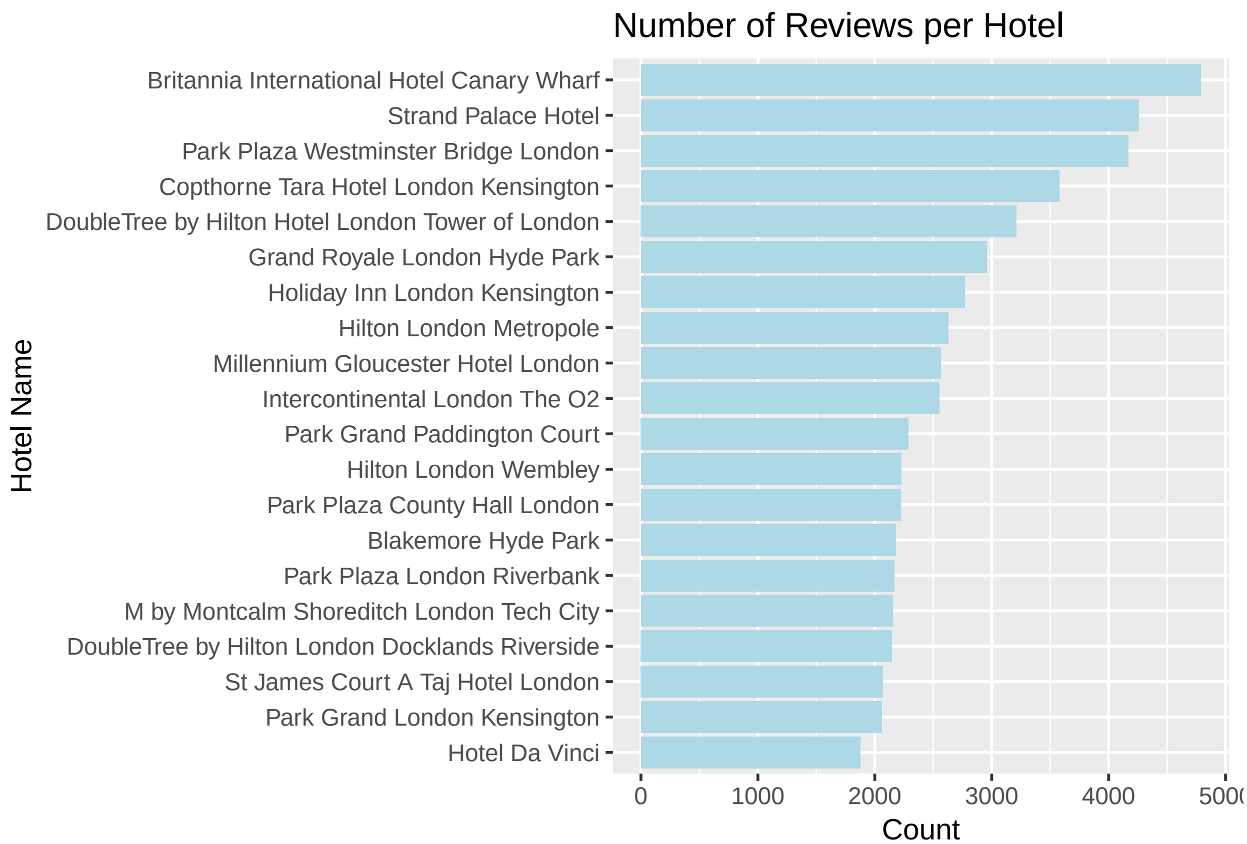
```
##    <chr>         <dbl>     <dbl>      <dbl>
##  1 missed   0.000155  0.000155   0.0000785
##  2 bath     0.00165   0.00165   -0.000371
##  3 staying  0.00110   0.00110   -0.000885
##  4 mattress 0.000667  0.000666   0.00119
##  5 advise   0.0000966 0.0000968 -0.00212
##  6 local    0.000386  0.000387  -0.00212
##  7 hotel    0.0283    0.0284    -0.00339
##  8 slept    0.000377  0.000375   0.00375
##  9 advice   0.0000870 0.0000866  0.00375
## 10 beaten   0.0000290 0.0000289  0.00375
## # ... with 3,975 more rows
```

```r
#less or more likely?
travellers_usage %>%
  group_by(logratio < 0) %>%
  top_n(15, abs(logratio)) %>%
  ungroup() %>%
  mutate(word = reorder(word, logratio)) %>%
  ggplot(aes(word, logratio, fill = logratio < 0)) +
  geom_col(show.legend = FALSE) +
  coord_flip() +
  ylab("log odds ratio (Business/Leisure)") +
  scale_fill_discrete(name = "", labels = c("Business", "Leisure"))
```

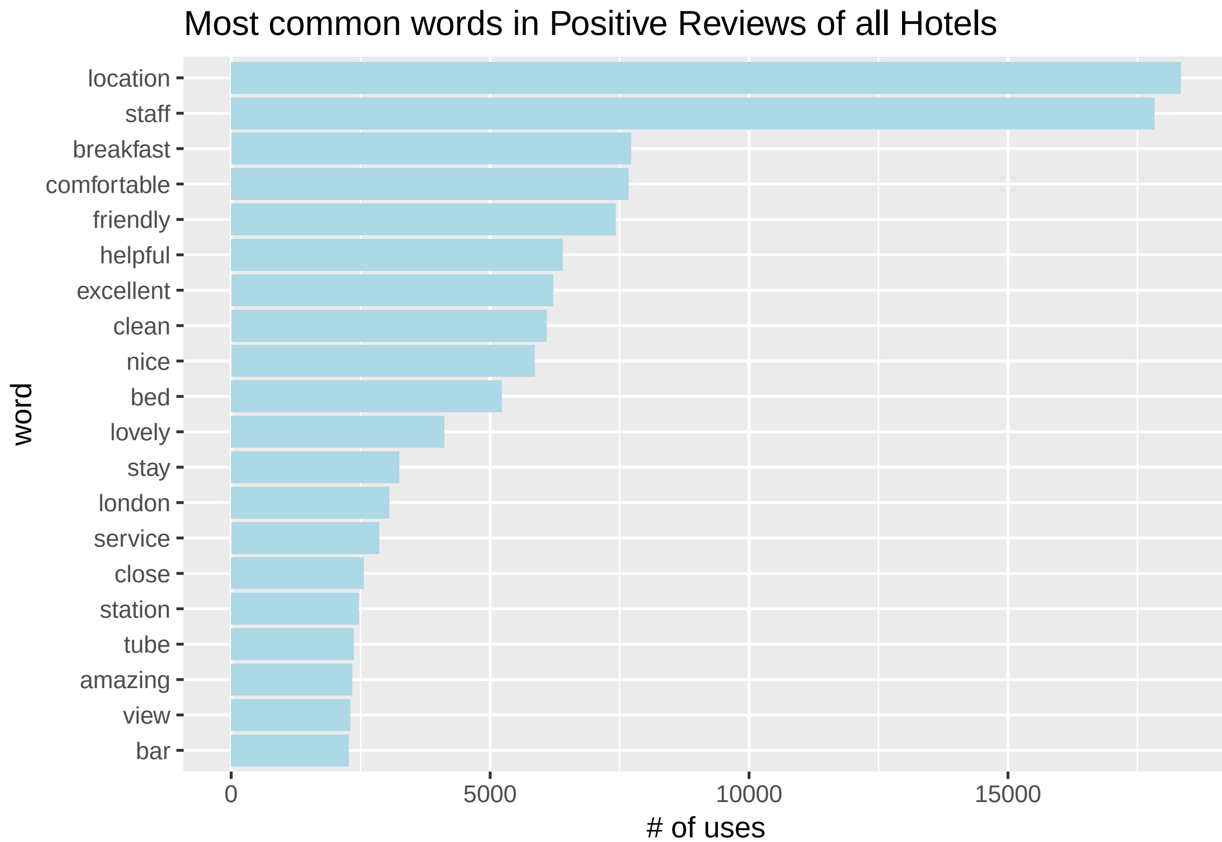



```r
#Data Visualization
```

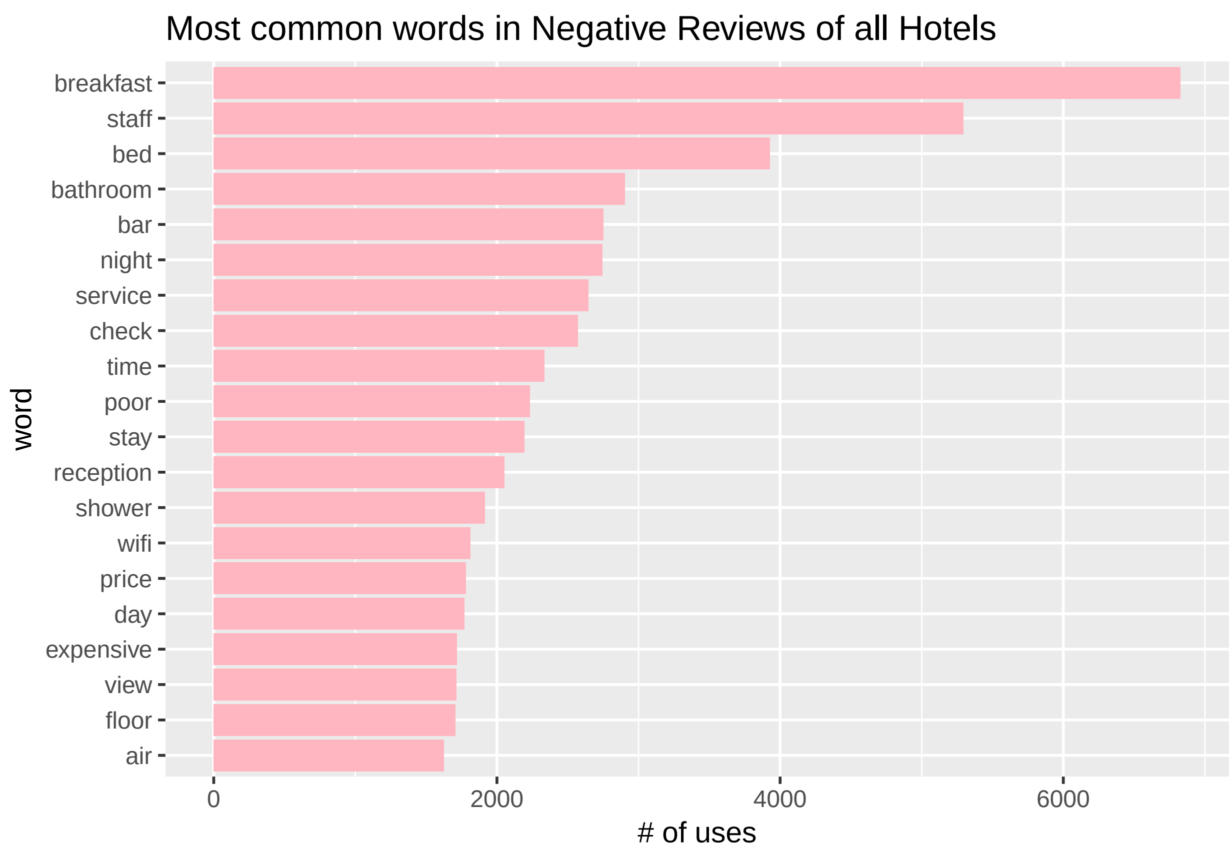
```r
hotel_name %>%
  mutate(Hotel_Name = reorder(Hotel_Name, count)) %>%
  ggplot(aes(Hotel_Name, count)) +
  geom_col(fill = "lightblue") +
  scale_y_continuous() +
  coord_flip() +
  labs(x = 'Hotel Name',title = "Number of Reviews per Hotel", y = 'Count')
```



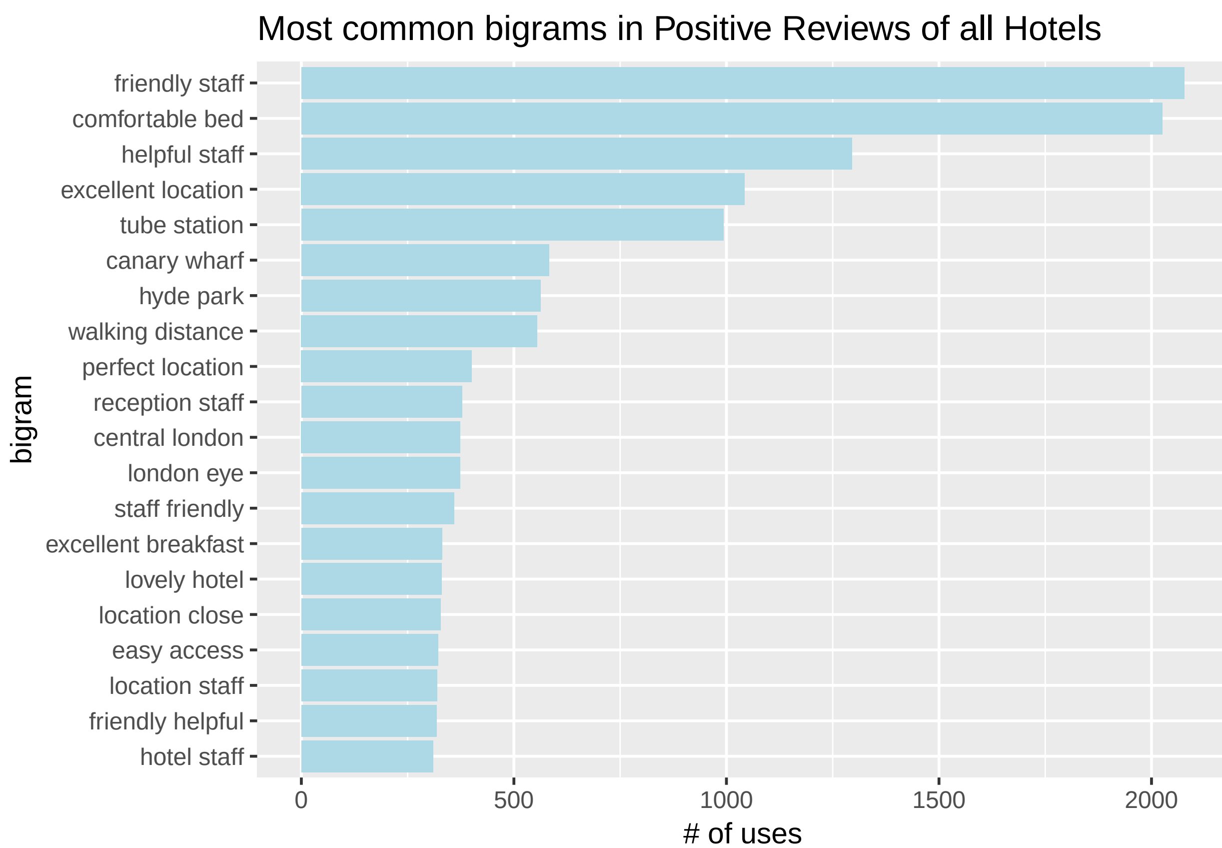

```r
#Word frequency of postive and negative reviews
pos_review %>%
  head(20) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
    geom_col(fill = "lightblue") +
    scale_y_continuous() +
    coord_flip() +
    labs(title = "Most common words in Positive Reviews of all Hotels",
 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



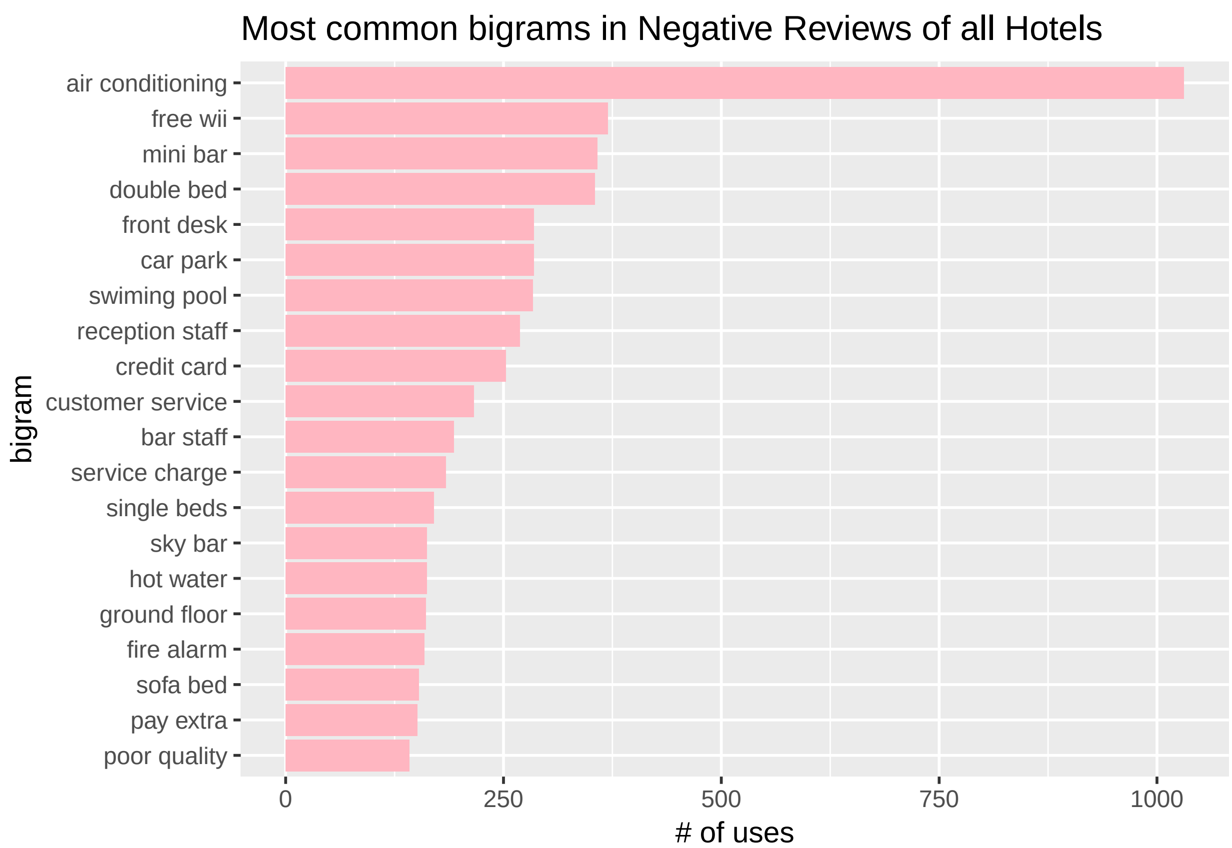

```r
neg_review %>%
  head(20) %>%
  mutate(word = reorder(word, n)) %>%
  ggplot(aes(word, n)) +
    geom_col(fill = "lightpink") +
    scale_y_continuous() +
    coord_flip() +
    labs(title = "Most common words in Negative Reviews of all Hotels",
         y = "# of uses")
```




```
#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")
```




```r
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")
```

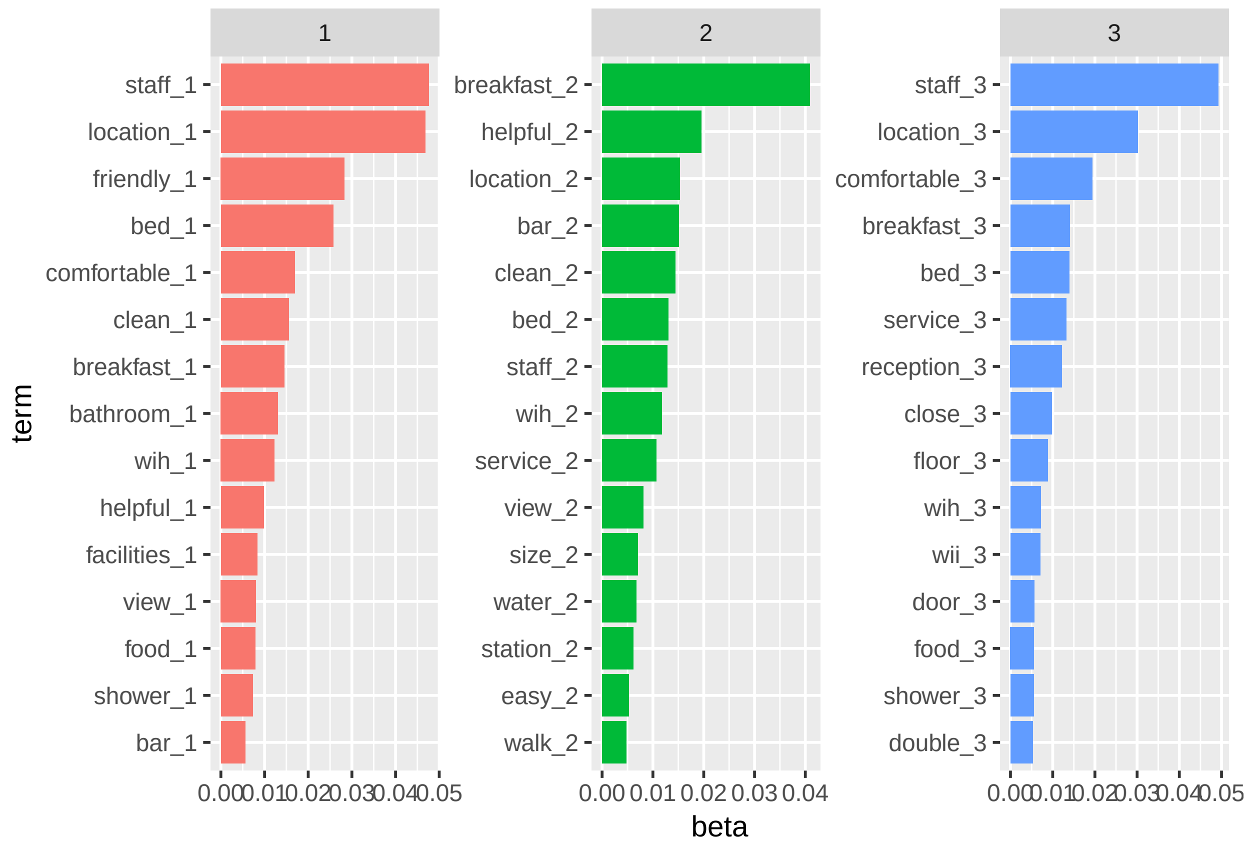



```r
#Topic Modelling
stop_words3 <- data_frame(word = c("london", "hotel", "bit", "didn", "wasn", "told", "positive", "negati
                                   "air", "stayed", "amazing", "perfect", "not", "3", "night", "day", "l
                                   ,"fantastic", "nice", "excellent", "loved", "check", "minutes", "free
                                   "booked", "price", "money", "paid", "4", "expensive"))

travellers_topic <- travellers %>%
  mutate(id = seq_along(Reviews)) %>%
  mutate(Reviews = gsub('comfy', 'comfortable', Reviews)) %>%
  mutate(Reviews = gsub('beds', 'bed', Reviews))


trav_dtm <- travellers_topic %>%
    unnest_tokens(word, Reviews) %>%
    anti_join(stop_words3) %>%
    anti_join(stop_words) %>%
    count(id, word, sort=TRUE) %>%
    ungroup() %>%
    cast_dtm(id, word, n)
```

```
## Joining, by = "word"
## Joining, by = "word"
```

```r
trav_lda <- LDA(trav_dtm, k = 3, control = set.seed(1234))
trav_topics <- tidytext::tidy(trav_lda, matrix = "beta")

trav_top <- trav_topics %>%
```

```r
  group_by(topic) %>%
  top_n(15, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

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




```r
trav_doc <- tidytext::tidy(trav_lda, matrix = "gamma")
trav_doc2 <- spread(trav_doc, topic, gamma)
trav_doc2 <- data.frame(trav_doc2)
trav_doc2$document <- as.numeric(trav_doc2$document)
trav_doc2 <- trav_doc2[order(trav_doc2$document), ]
names(trav_doc2) <- c("id", "Topic_1_hotel_rooms", "Topic_2_staff", "Topic_3_location")

max_topic <- cbind(trav_doc2, Topic = names(trav_doc2[2:4])[apply(trav_doc2[2:4],1,which.max)]  )
```

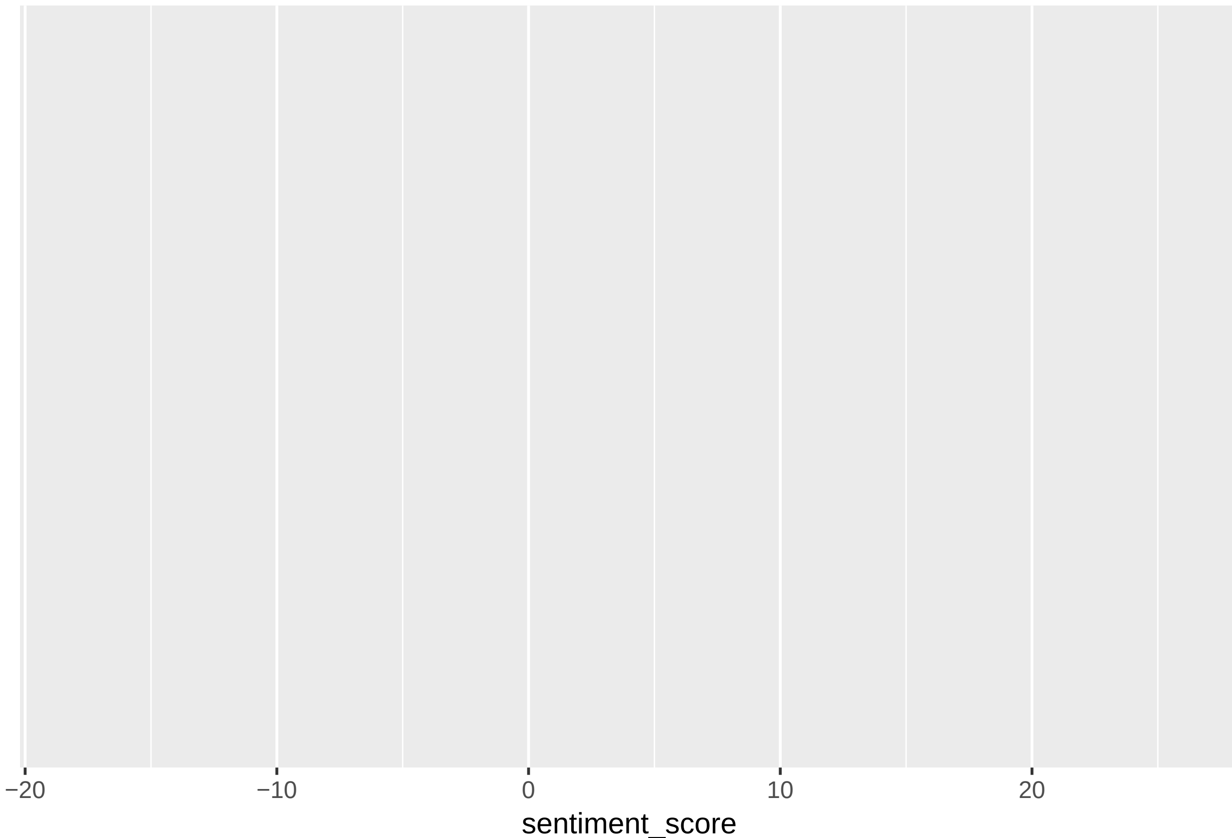
```r
head(trav_doc2)
```

```
##       id Topic_1_hotel_rooms Topic_2_staff Topic_3_location
## 1      1           0.3256901     0.3529234        0.3213865
## 10912  2           0.3238252     0.3347758        0.3413990
## 21888  3           0.3409699     0.3475427        0.3114874
## 32777  4           0.3412398     0.3265211        0.3322391
## 43675  5           0.3337379     0.3385706        0.3276914
## 47822  6           0.3400372     0.3216614        0.3383014
```

```r
#Sentiment Analysis
trav_sentiment <- travellers_topic
trav_sentiment$Reviews <- removeWords(trav_sentiment$Reviews, stop_words2)

trav_sentiment <- trav_sentiment %>%
  unnest_tokens(word, Reviews) %>%
  anti_join(stop_words) %>%
  inner_join(get_sentiments("bing")) %>%
  group_by(id, sentiment) %>%
  summarise(count = n()) %>%
  spread(sentiment, count, fill=0) %>%
  mutate(sentiment_score = positive - negative) %>%
  mutate(review_sentiment = ifelse(sentiment_score == 0, "Neutral", ifelse(sentiment_score > 0, "Positi
```

```
## Joining, by = "word"
## Joining, by = "word"
```

```r
trav_sentiment %>%
  ggplot(aes(sentiment_score, ))
```




```r
#Building dataframe for analysis

df_analysis <- travellers_topic %>%
  full_join(max_topic)
```

```
## Joining, by = "id"
```

```r
df_analysis <- df_analysis %>%
  full_join(trav_sentiment)
```

```
## Joining, by = "id"
```

```r
df_analysis <- df_analysis[, c("type", "id", "Topic", "review_sentiment", "Reviewer_Score")]


df_analysis [ , c("type", "review_sentiment")] <- lapply(df_analysis[, c("type", "review_sentiment")], a
df_analysis$id <- NULL
```

```r
#Reression Analysis
model1 <- lm(data = df_analysis, Reviewer_Score ~ .,)
summary(model1)
```

```
##
## Call:
## lm(formula = Reviewer_Score ~ ., data = df_analysis)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -6.4185 -0.9066  0.2815  1.0815  3.8771
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)               6.36088    0.02456  258.98   <2e-16 ***
## typeLeisure               0.51190    0.01986   25.78   <2e-16 ***
## TopicTopic_2_staff       -0.23796    0.01727  -13.78   <2e-16 ***
## TopicTopic_3_location    -0.20354    0.01758  -11.58   <2e-16 ***
## review_sentimentNeutral   0.96767    0.02702   35.81   <2e-16 ***
## review_sentimentPositive  2.04572    0.01812  112.88   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.522 on 45243 degrees of freedom
##   (7898 observations deleted due to missingness)
## Multiple R-squared:  0.2592, Adjusted R-squared:  0.2591
## F-statistic:  3166 on 5 and 45243 DF,  p-value: < 2.2e-16
```