615 final

He Guo

11/30/2019

Review Analysis

This study use review file in Yelp dataset. There is a stars variable record the stars of each business. This study wants to mark the stars of thoes business as High, Mid, and Low. When stars greater to 4, rate is High. When stars is greater and equals to 3, and lower and equals to 4, rate is Mid. When stars is lower than 3, rate is Low.

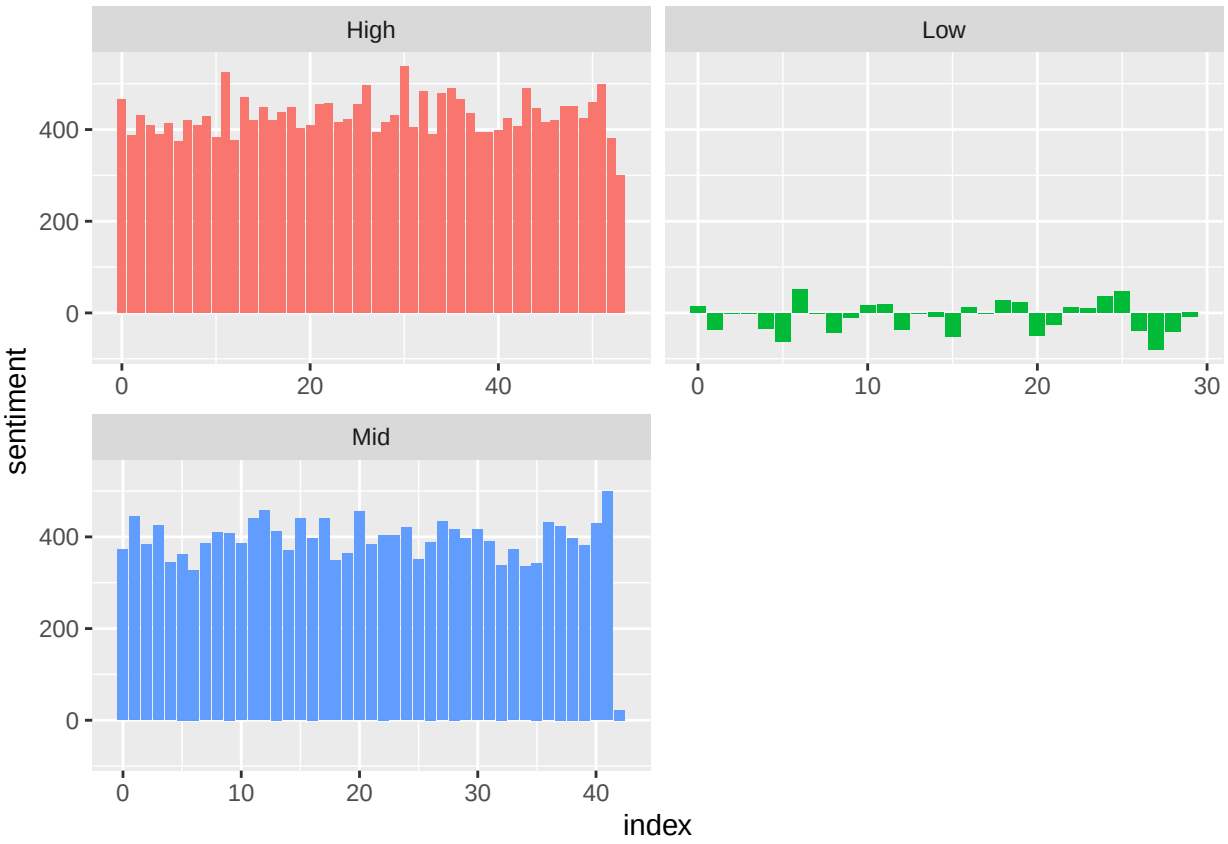
The study apply text analysis by using text variable and rate variable. Text is review from users of each business. At first, the study separate the text variable in to words. In the other words, the study splits sentences into words, and group by the Rate variable. Since this study wants to analysis review between different Rate.

Sentiment

The current study want to analysis the positive/negative sentiment about the review from userd. The the study use lexicon from tidytext package, and we use bing lexicon categorizes words to apply this analysis. The bing lexicon categorizes words has positive and negative categories.

The study want to estimates how sentiment changes within different Rate categories. After that the study find the sentiment score for each word usinf Bing lexicon and inner_join() methods. Because the review section is too large, it is not good for doing text analysis. Then, the study decide using 80% lines to apply this analysis.

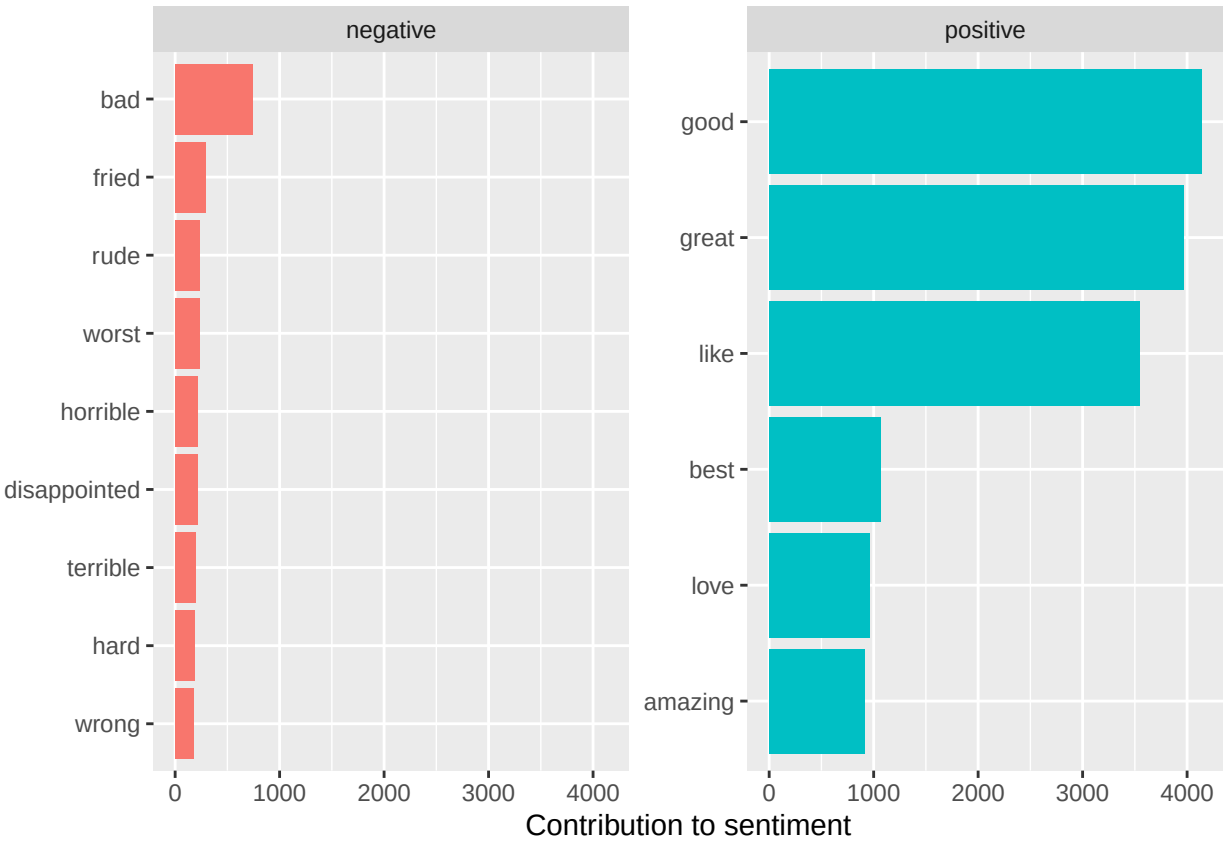
According to the following plot, users post positive review to business in High and mid Rate. Users post a higher positive sentiment score to business in High Rate. There are some negative score to business in low Rate. It means that people will post neigative review to business in low Rate.



Then the study wants to know what is the most common positive score and neigative score for the review from users. The current study uses count() with arguments of both word and sentiment, we find out how much each word contributed to positive and negative sentiment.

The following plot shows that the words contirbutes the most to negative is bad. The words contirbutes the most to positive is good.

```
## Selecting by n
```



Words Clouds

The following plot is words clouds. It shows the most frequency words are food, service.

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

```
## Warning in wordcloud(word, n, max.words = 100): service could not be fit on
## page. It will not be plotted.
```

The study use group_by and join to get the most frequence words in Reviews text from users in Yelp.

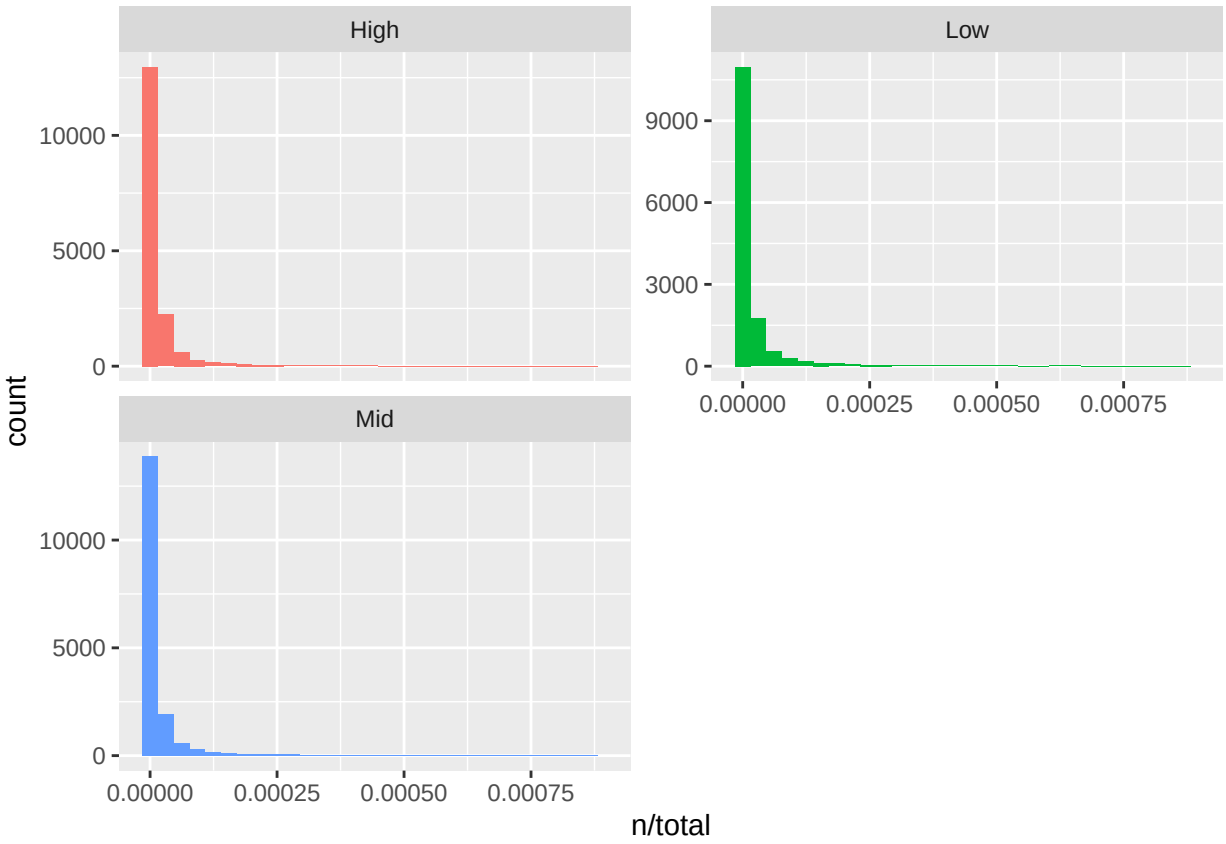
```
## Joining, by = "review.Rate"
```

n is is the number of times that word is used in Reviews text and total is the total words in Reviews text. In the followling figure, it shows that the distribution of n/total for each Rate category.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

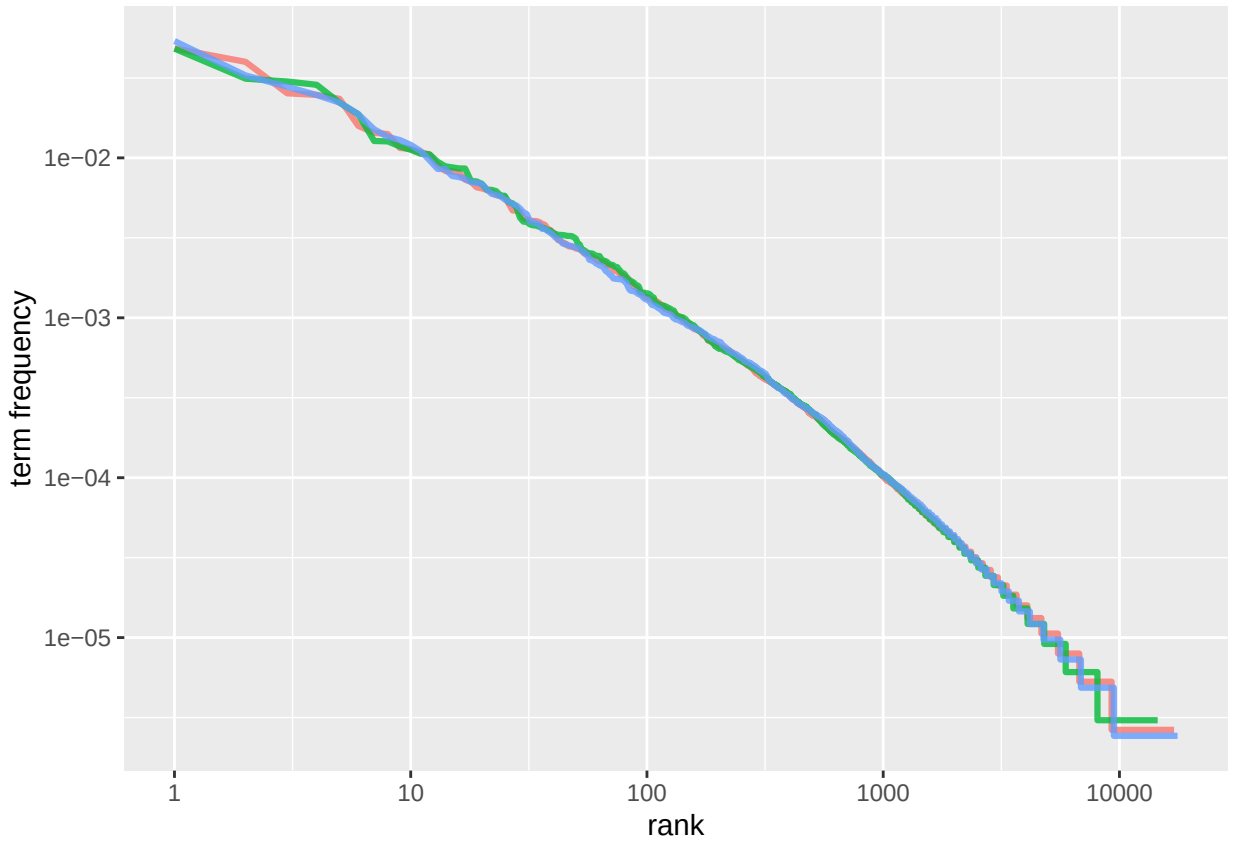
```
## Warning: Removed 454 rows containing non-finite values (stat_bin).
```

```
## Warning: Removed 3 rows containing missing values (geom_bar).
```



Zipf's Law

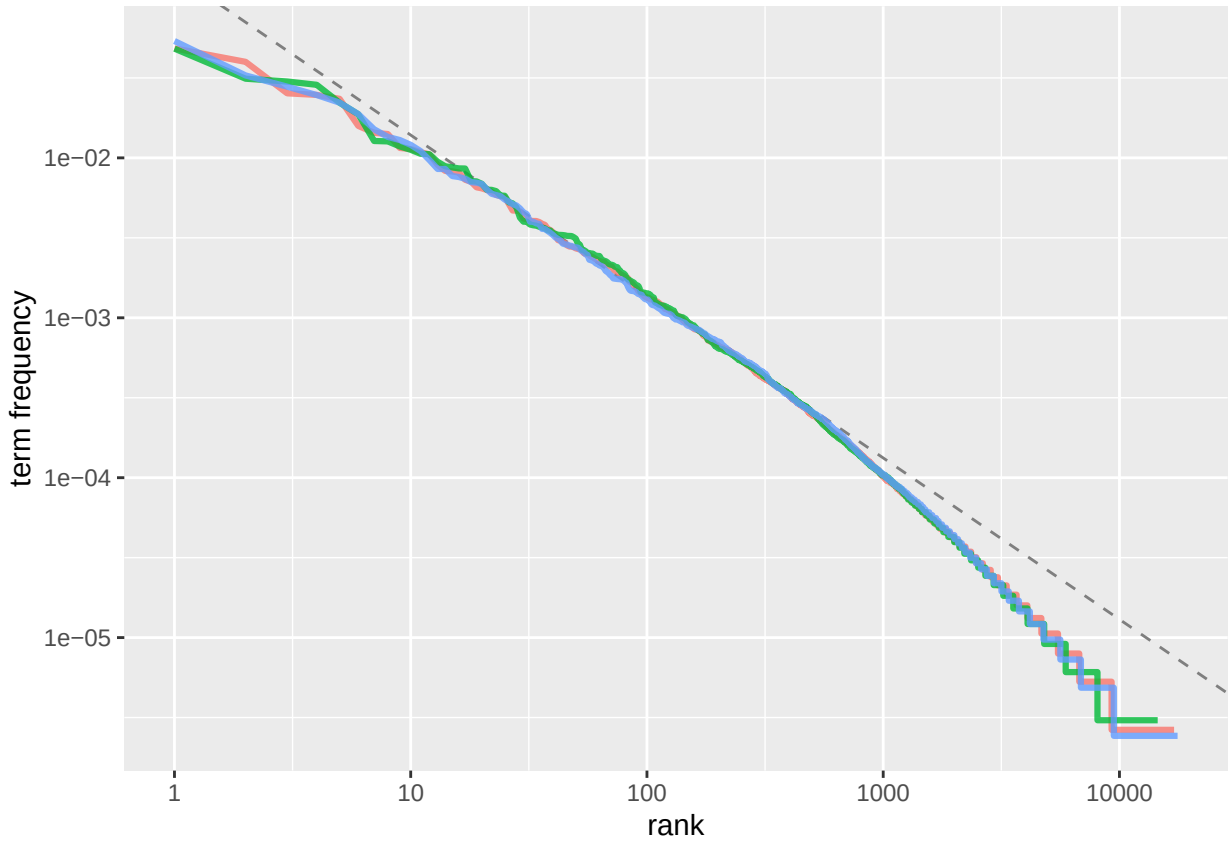
The study use Zipf's Law for Reviews text. The rank variable gives the rank of each word in the frequency table, and the table is order by n. Zipf's Law is visualized by the plot of freq_by_rank. Rank is on x-axis and term frequency is on y-axis. The plot is on log scales.



According to the plot, we can see that three categories are simlilar to each other, and the relationship between rank and frequency have a negative slop. The study fit a linear regression with $log(termfrequency)$ and $log(rank)$.

```
##
## Call:
## lm(formula = log10(`term frequency`) ~ log10(rank), data = rank_subset)
##
## Coefficients:
## (Intercept)  log10(rank)
##     -0.8401      -1.0138
```

After get the linear regression, we add the fitted line into ggplot.



Consecutive Words

Next, the stuy wants to analysis the most common phrase for three Rate category. The study use unnest_tokens() to estimate the pairs of two consecutive words. The study separate the phrase in word1, and word2. After delete the words does not have meaning, the study use unite() to recombine word1 and word2.

The study compute the tf_idf for bigrams across Rate category. tf_idf return how important a phrase in review text of users in Yelp. The following plot visualized the tf_idf within each Rate category.

```
## Warning in kableExtra::kable_styling(., bootstrap_options = c("striped", :
## Please specify format in kable. kableExtra can customize either HTML or
## LaTeX outputs. See https://haozhu233.github.io/kableExtra/ for details.
```

review.Rate	bigram	n	tf	idf	tf_idf
Low	horrible service	23	0.0007358	1.098612	0.0008084
Low	horrible customer	18	0.0005759	1.098612	0.0006327
High	amazing job	25	0.0005485	1.098612	0.0006025
Low	slow slow	15	0.0004799	1.098612	0.0005272
Low	terrible customer	15	0.0004799	1.098612	0.0005272
Low	worst customer	15	0.0004799	1.098612	0.0005272

```
## Selecting by tf_idf
```


High
amazing job
yum yum
favorite restaurant
extremely knowledgeable
perfectly cooked
wow wow
service love
nail art
smog check
panna cotta
clean environment
Low
horrible service
horrible customer
worst customer
terrible customer
slow slow
0 stars
horrible experience
negative stars
poor customer
water pump
low quality
bill luke
Mid
pleasantly surprised
beef hash
pour le
chain restaurant
free times
3.5 5
3.5 stars
zucchini fries
hand pulled
bbq ribs
tf_idf