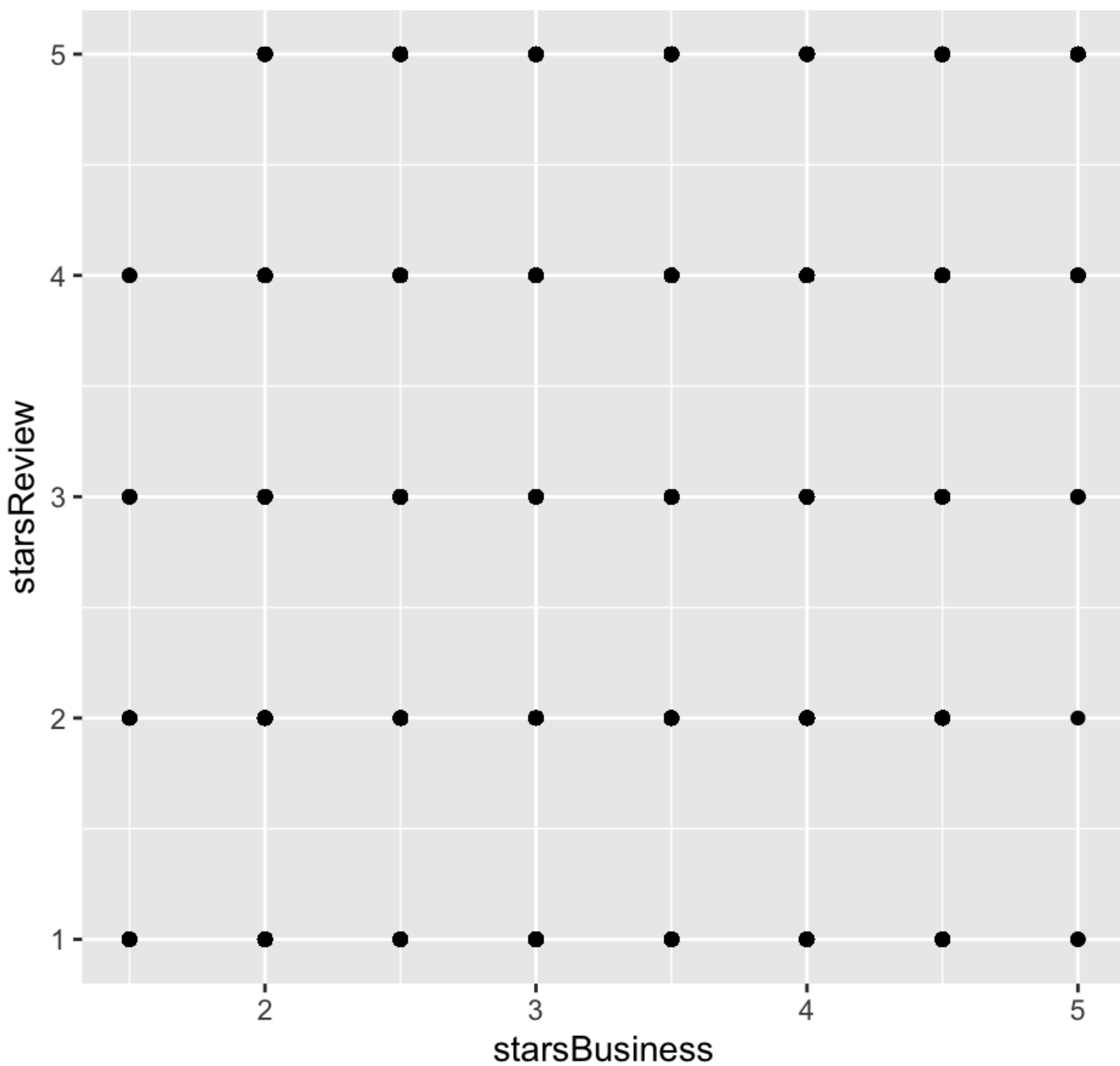
Raymond Sutanto, Cheng Lin Tsai, Yi Jen Chen
November 20, 2022
IDS 572 Assignment 3
Prof. Siddhartha Bhattacharyya

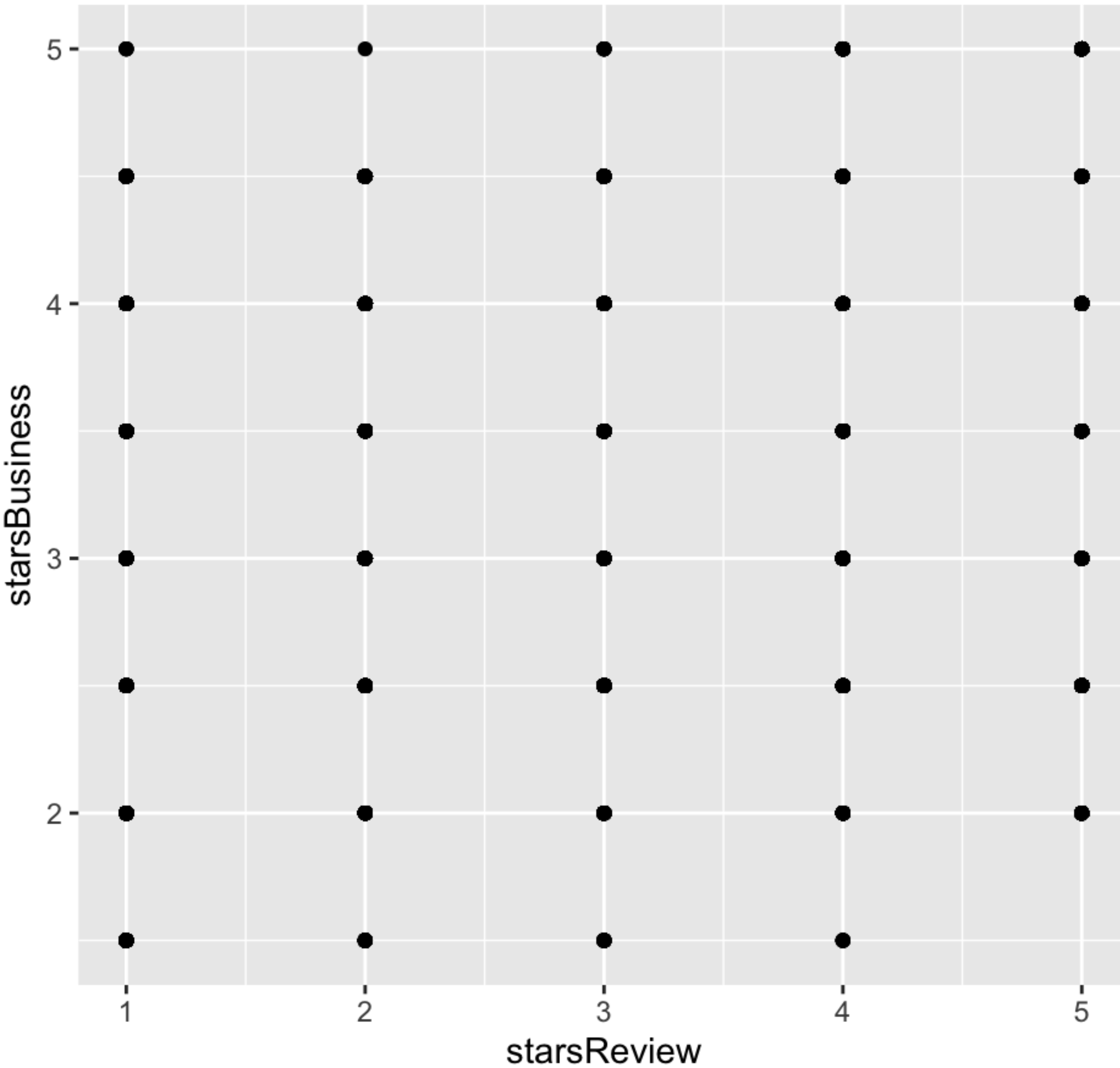
1. Explore the data.
(a) How does star ratings for reviews relate to the star-rating given in the dataset for businesses (attribute 'businessStars')? Can one be calculated from the other?

- **The stars ratings for reviews do not relate from the star-rating given in the dataset for businesses. We plot the starReviews data to determine starBusiness, but we don't see the relationship between the two, and vice versa. Therefore, we can't calculate one from the other.**

> ggplot(resReviewsData, aes(x= starsBusiness, y=starsReview)) +geom_point()



> ggplot(resReviewsData, aes(x= starsReview, y=starsBusiness)) +geom_point()

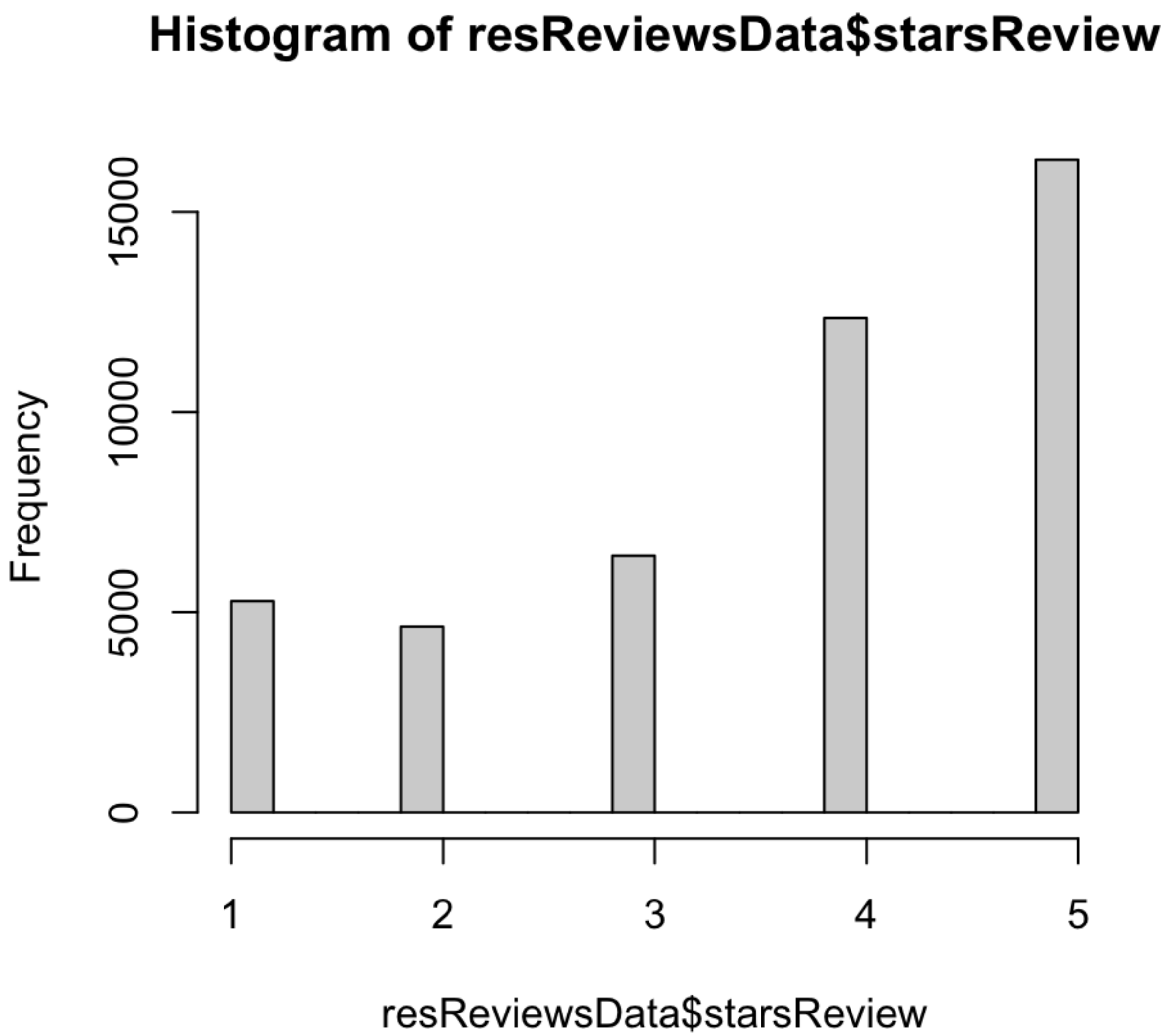


(b) Here, we will focus on star ratings for reviews. How are star ratings distributed? How will you use the star ratings to obtain a label indicating 'positive' or 'negative' – explain using the data, summaries, graphs, etc.?

- **We observed an unequal distribution from the five different numbers of star ratings. Based on the below histogram, the highest number of reviews are for 5 stars rating (with 16301 reviews) and the lowest number of reviews are for 2 stars rating (with 4649 reviews).**
- **For our observation, we classified 1 & 2 stars ratings as bad, and 4 & 5 stars ratings as positive. With this, if a user picks a positive rating, there is a possibility for the user to return to the restaurant. In addition, new users are more inclined to**

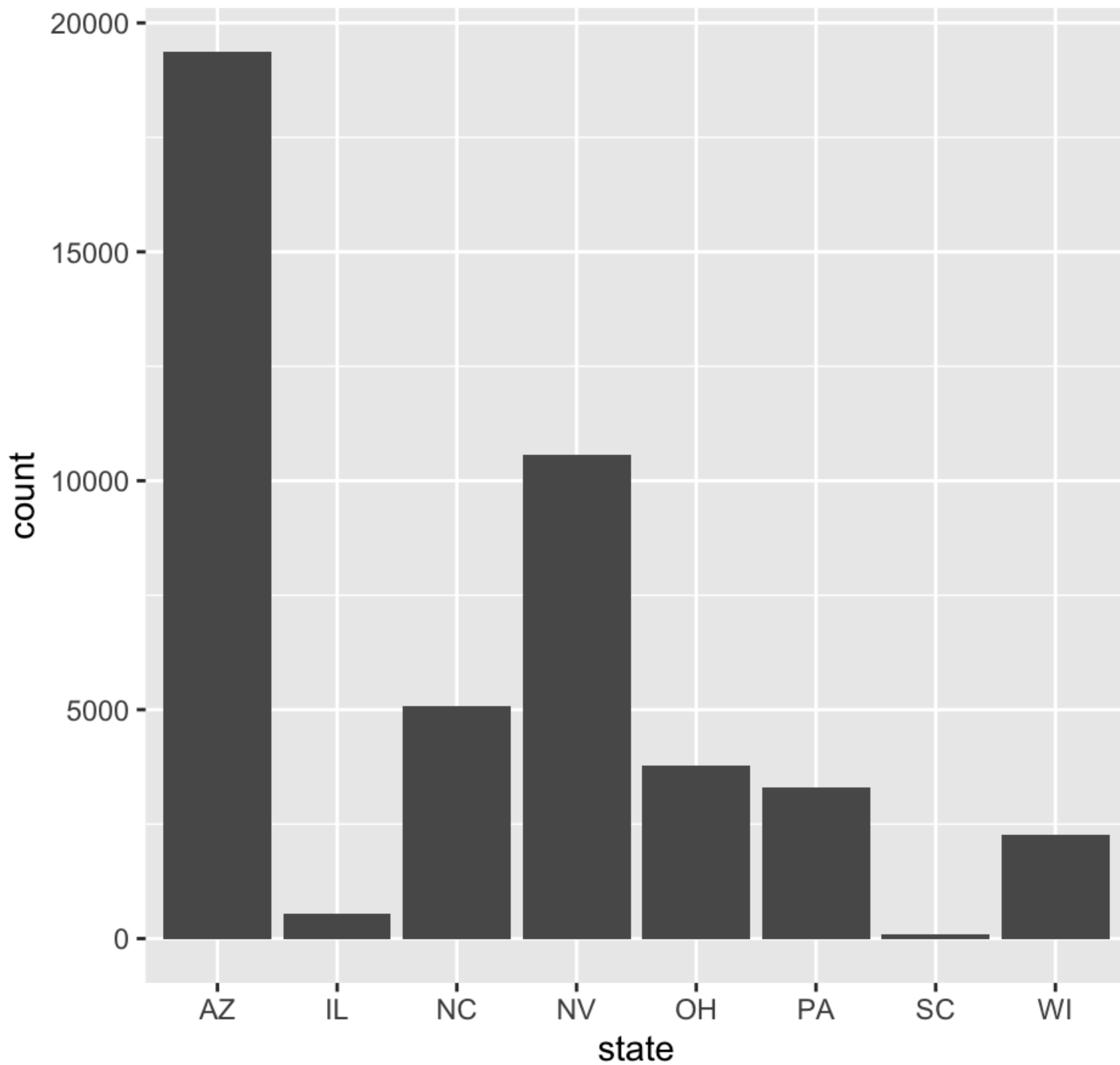
go to restaurants with higher ratings, which result in higher positive reviews, as we observed in the histogram.

- Furthermore, we will later disregard the three stars when creating the models since we assumed this to be a "neutral" evaluation with no positive/negative presence.



- We also distributed the number of reviews by state. To do this, we summed the total number of reviews for each state. The distribution of reviews by state is displayed in the below bar chart.
- According to the bar chart, the highest number of reviews are from Arizona (with 19367 reviews). Nevada is the state that has the second highest number of reviews (with 10563 reviews).

- **One possible explanation for this may be due to the fact that the majority of the respondents come from these two states and most residents of these states like to eat out and also like the restaurants they've reviewed.**



2. What are some words in the restaurant reviews indicative of positive and negative sentiment – identify at least 20 in each category. One approach for this is to determine the average star rating for a word based on star ratings of documents or reviews where the word occurs. Do these 'positive' and 'negative' words make sense in the context of user reviews for restaurants being considered? (For this, since we'd like to get a general sense of positive/negative terms, you may like to consider a pruned set of terms -- say, those which occur in a certain minimum and maximum number of documents).

- **We summarize the average star ratings for each word, then look for the words' occurrences across different ratings, to determine whether certain terms are a sign of a positive or bad review.**
- **Words like restaurant, food, time, and service that are often used across all star ratings were deleted because they do not necessarily indicate a positive or negative assessment**
- **The majority of the terms in both positive and negative sentiments make sense, yet some neutral words are classified as positive after looking at the top words linked to a good/negative assessment.**
- **We further reduce the set by eliminating words that appear more than 10,000 times to address this. This cutoff is established without removing terms that are obviously positive or negative by counting the number of times certain words appear in the top 20 positive list. In this case, the threshold is 10,000.**

Top 10 words in the review prior to first adjustment

word <chr>	n <int>
the	262187
and	173415
a	128521
i	127100
to	108132
was	95499
of	73822
it	68169
is	62864
for	58251

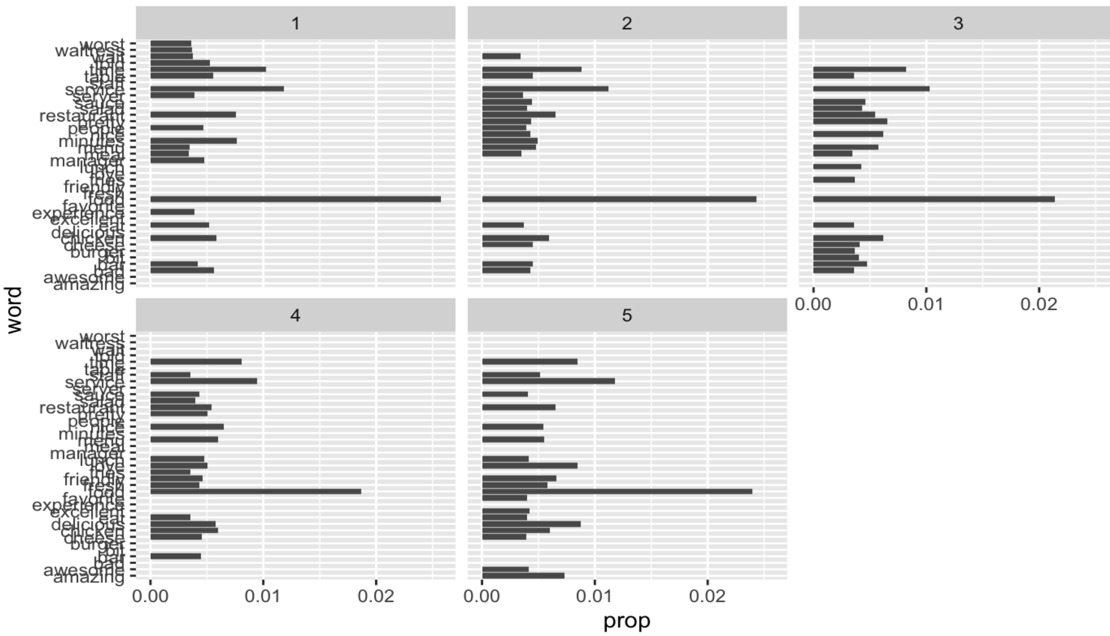
Top 10 words in the review after the first adjustment (remove stopwords)

starsReview <dbl>	word <chr>	n <int>
5	food	11773
4	food	8836
5	service	5796
3	food	5776
1	food	5266
2	food	4909
4	service	4468
5	delicious	4278
5	love	4150
5	time	4143

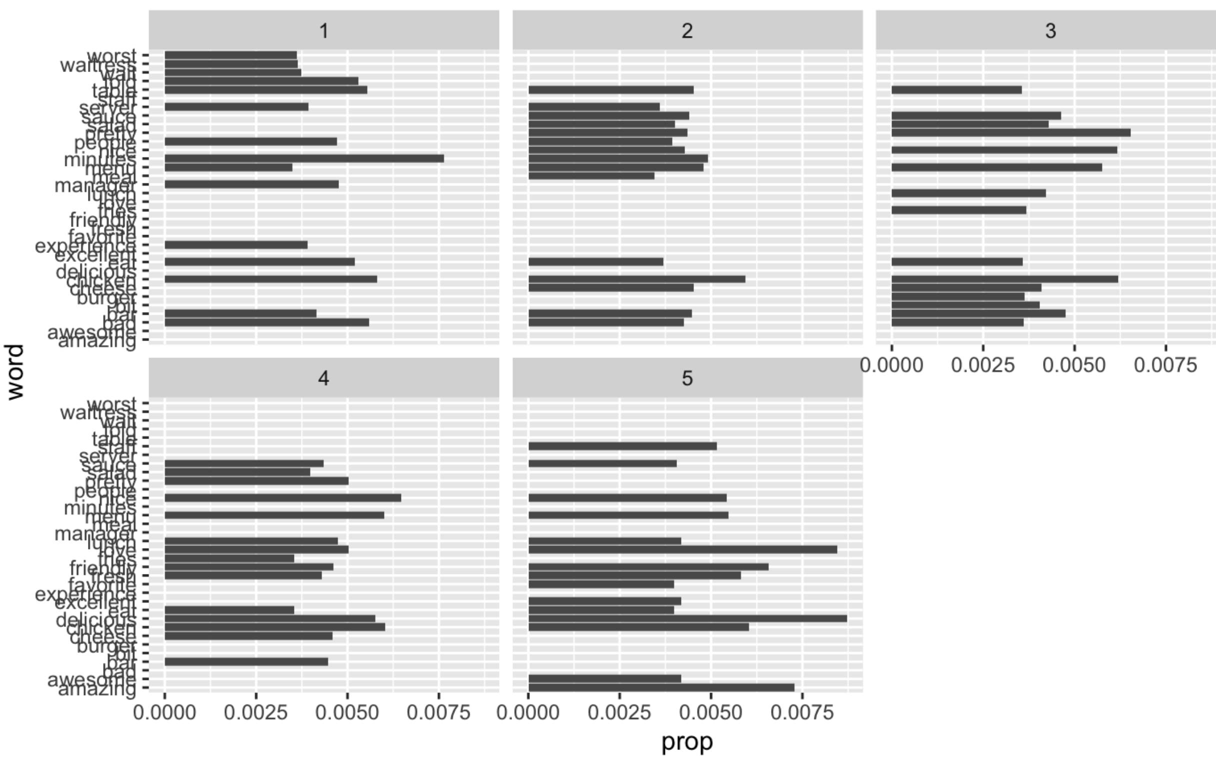
Example: check the proportion of 'love' among reviews with 1,2,3,4, and 5 stars

starsReview <dbl>	word <chr>	n <int>	prop <dbl>
5	love	4150	0.008457823
4	love	2373	0.005021319
3	love	833	0.003088777
2	love	438	0.002170100
1	love	312	0.001525725

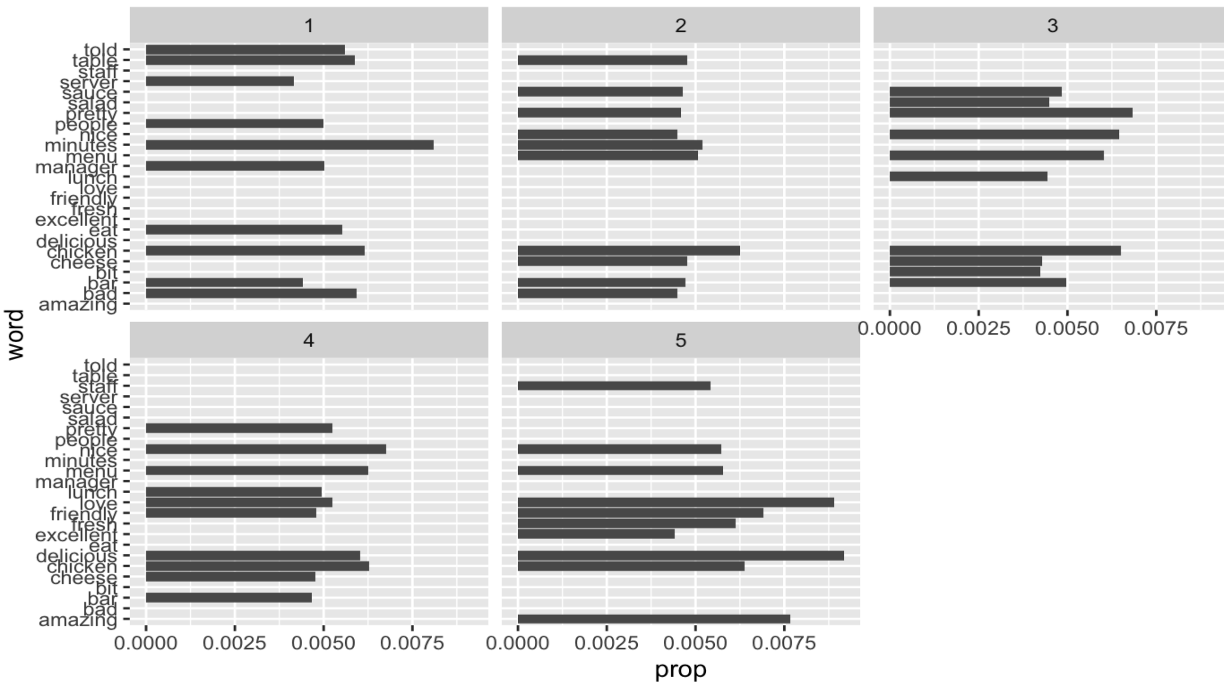
The proportion of different words by star ratings



The proportion of different words by star ratings after removing words like ‘food’, ‘time’, ‘restaurant’, and ‘service’ that occur in all ratings



The proportion of different words by star ratings after pruning a set of words that occur more than 10,000 times in all reviews



Top 20 words that are related with the highest rating

word	totWS
<chr>	<dbl>
amazing	0.05242405
bar	0.06239026
cheese	0.06217452
chicken	0.09060236
delicious	0.07651684
eat	0.05737317
food	0.33340521
fresh	0.05957076
friendly	0.06470365
love	0.07750665

word <chr>	totWS <dbl>
lunch	0.06034794
menu	0.08175634
nice	0.08242848
pretty	0.06106264
restaurant	0.09092206
salad	0.05800279
sauce	0.06361492
service	0.16209004
staff	0.05840553
time	0.12687322

Top 20 words that are connected with the lowest rating

word <chr>	totWS <dbl>
appalling	6.871970e-05
bait	6.106831e-05
boyardee	7.019231e-05
canceled	6.106831e-05
choked	6.768451e-05
choking	6.874726e-05
disgrace	6.925841e-05
drawer	6.740350e-05
filth	5.881056e-05
gina	6.849721e-05

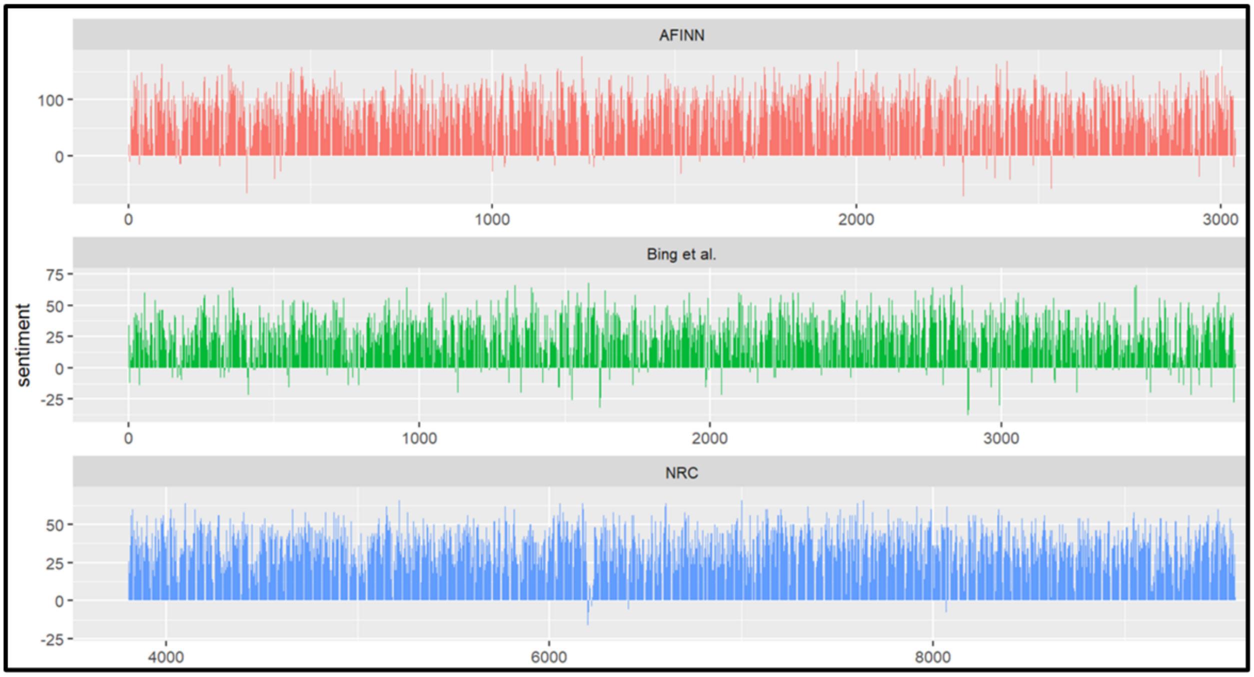
word <chr>	totWS <dbl>
helper	6.994225e-05
inexcusable	6.504447e-05
inspector	5.777538e-05
roaches	6.638823e-05
scolding	6.608730e-05
spills	6.492326e-05
stored	6.395840e-05
swallowed	6.758323e-05
toilets	6.781336e-05
violent	7.024319e-05

3. We will consider three dictionaries, available through the tidytext package – (i) the extended sentiment lexicon developed by Prof Bing Liu, (ii) the NRC dictionary of terms denoting different sentiments, and (iii) the AFINN dictionary which includes words commonly used in user-generated content in the web. The first specifies lists of positive and negative words, the second provides lists of words denoting different sentiment (for eg., positive, negative, joy, fear, anticipation, …), while the third gives a list of words with each word being associated with a positivity score from -5 to +5.

(a) How many matching terms (i.e. terms in your data which match the dictionary terms) are there for each of the dictionaries?

- **There are 6786 terms in the bing dictionary, 13872 terms in the nrc dictionary, and 2477 terms in the AFINN dictionary.**
- **Across the three dictionaries, we found 2154 matching terms.**

(b) What is the overlap in matching terms between the different dictionaries? Based on this, do you think any of the three dictionaries will be better at picking up sentiment information from you text of reviews? (c) Consider the positive and negative terms you determined in Q 2 above; which of these terms match with terms in each of the three dictionaries?

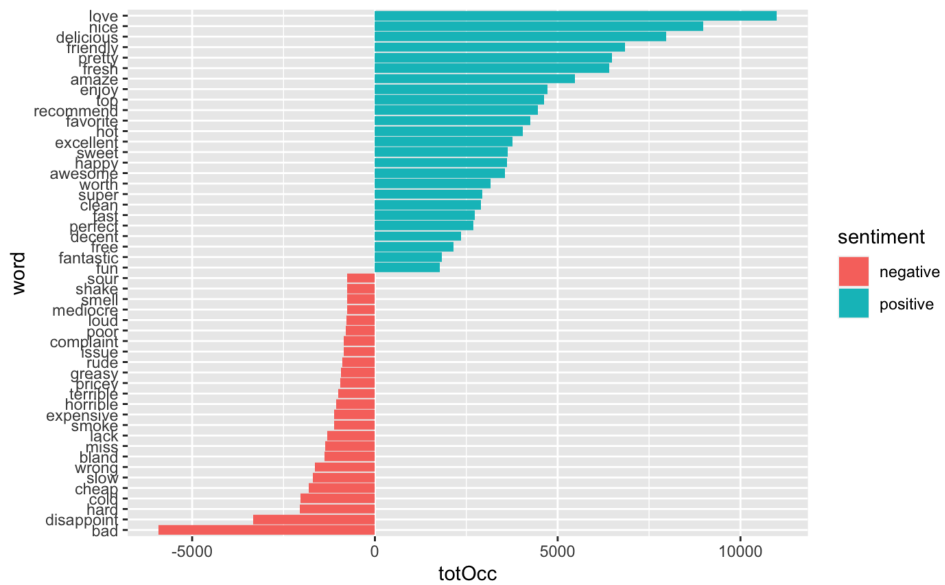


- **To answer this question we visualize the sentiments provided by each of the dictionary, and based on that we will decide if one dictionary is better than the rest.**
- **The three different lexicons for calculating sentiment give results that are different in an absolute sense but have similar relative trajectories throughout.**
- **The AFINN lexicon gives the largest absolute values, with high positive values.**
- **The lexicon from Bing has lower absolute value comparing the three Sentiment Dictionaries and seems to label larger blocks of contiguous positive or negative text.**
- **The NRC results are shifted higher relative to the other two, labeling the text more positively**
- **Analyzing the above data, we find that Bing is better at picking up sentiment information from our text of reviews. It is better able to pick both positive and negative sentiments.**

4. Consider a basic approach (not developing a predictive model like a decision tree, random forests etc.) to use the dictionary based on positive and negative terms to predict sentiment (positive or negative based on star rating) of a review. One approach for this is: based on each dictionary, obtain an aggregated positiveScore and a negativeScore for each review; for the AFINN dictionary, an aggregate positivity score can be obtained for each review.

(a) Describe how you obtain the aggregated scores, and predictions based on these scores
(b) What is the performance of this approach (for each dictionary). Does any dictionary perform better?

- **A sentiment analysis is an extension of text mining that analyzes the opinions and subjectivity of content. It can reveal both customer attitudes and their pervasive likes and dislikes when applied to restaurant reviews. For sentiment analysis, we used a lexicon-based approach**.
- **The graph below shows the top & lowest occurrences of 25 words using Bing Dictionary**



- **In order to determine whether words have a favorable or negative attitude, we used the Bing lexicon. After that, we summed up the sentiment words for each review. A sentiment score was then constructed based on the percentage of positive and negative words. Results of the analysis summarized versus star ratings are shown in the table below:**

starsReview <dbl>	avgPos <dbl>	avgNeg <dbl>	avgSentiSc <dbl>
1	0.3013553	0.6986447	−0.3972893
2	0.4403782	0.5596218	−0.1192436
3	0.6044573	0.3955427	0.2089145
4	0.7424808	0.2575192	0.4849617
5	0.8242529	0.1757471	0.6485058

- **Based on aggregated reviews, we can further classify reviews (predict) as high or low.**
- **The confusion matrix of using BING Dictionary is as follows and its accuracy is 80.95%.**

```
          Reference
Prediction    -1     1
        -1  8108  5615
        1   1548 22325
                                         
               Accuracy : 0.8095          
                 95% CI : (0.8055, 0.8134)
    No Information Rate : 0.7432          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.5614          
                                          
 Mcnemar's Test P-Value : < 2.2e-16       
                                          
            Sensitivity : 0.8397          
            Specificity : 0.7990          
         Pos Pred Value : 0.5908          
         Neg Pred Value : 0.9352          
             Prevalence : 0.2568          
         Detection Rate : 0.2157          
   Detection Prevalence : 0.3650          
      Balanced Accuracy : 0.8194          
                                          
       'Positive' Class : -1              
```

- **The confusion matrix of using NRC Dictionary is as follows and its accuracy is 74.87%:**

```
          Reference
Prediction    -1     1
        -1  7264  7008
        1   2625 21439
                                         
               Accuracy : 0.7487          
                 95% CI : (0.7443, 0.7531)
    No Information Rate : 0.742           
    P-Value [Acc > NIR] : 0.001394        
                                          
                  Kappa : 0.4265          
                                          
 Mcnemar's Test P-Value : < 2.2e-16       
                                          
            Sensitivity : 0.7346          
            Specificity : 0.7536          
         Pos Pred Value : 0.5090          
         Neg Pred Value : 0.8909          
             Prevalence : 0.2580          
         Detection Rate : 0.1895          
   Detection Prevalence : 0.3723          
      Balanced Accuracy : 0.7441          
                                          
       'Positive' Class : -1              
```

- **The confusion matrix of using AFINN Dictionary is as follows and its accuracy is 83.71% :**

```
          Reference
Prediction    -1     1
        -1  5842  2415
        1   3575 24944

               Accuracy : 0.8371
                 95% CI : (0.8333, 0.8409)
    No Information Rate : 0.7439
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5545

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.6204
            Specificity : 0.9117
         Pos Pred Value : 0.7075
         Neg Pred Value : 0.8746
             Prevalence : 0.2561
         Detection Rate : 0.1589
   Detection Prevalence : 0.2245
      Balanced Accuracy : 0.7660

       'Positive' Class : -1
```

- **Predicted numbers are based on star ratings, while expected values are based on sentiment scores we generate. As we can see, AFINN dictionary has the highest accuracy, so we can conclude that it is better at predicting highs and lows.**

5. Develop models to predict review sentiment. For this, split the data randomly into training and test sets. To make run times manageable, you may take a smaller sample of reviews (minimum should be 10,000). You should consider models built using only the terms matching the sentiment dictionaries, as well as by using a broader list of terms (the idea here being, maybe words other than only the dictionary terms can be useful). You should develop at least three different types of models (Naïve Bayes, and at least two others of your choice ….Lasso logistic regression (why Lasso?), xgb, random forest (use ranger for faster run-times) – use the same three modeling techniques with each of the dictionaries, with the combination of dictionary terms, and with the broader set of terms.

(a)How do you evaluate performance? Which performance measures do you use, why?
(b)Which types of models does your team choose to develop, and why? Do you use term frequency, tfidf, or other measures, and why?

- **To build out Test, Train, and Validation sets, we randomly subsample 50% of the full dataset. In order to balance computation time with model quality, we require at least 10,000 data points when building models. We divide the subsample 70% for Training, 20% for Testing, and 10% for Validation. The datasets were divided as follows: Bing dataset, NRC dataset, AFINN dataset, and Broader Terms (no dictionary)**.

- **The Naive Bayes, SVM, and Random Forest algorithms were developed. These models, and RF classifiers in particular, are well suited for handling high dimensional noisy data in text categorization. Some of these improve accuracy by accounting for majority vote on predictions.**

- **Yes, TF,IDF, TF-IDF, have been used. How frequently a word appears in a document, or its term frequency (tf), is a sign of its value. However, there are other terms that are used frequently but may not be crucial; in English, these would probably be words like "the," "is," "of," and so on. Before studying the papers, we might add these terms to a list of stop words and strike them out, although it's likely that some of these words are more crucial in particular texts than others. A stop word list is not a very sophisticated approach to control term frequency for commonly used words.**

- **Inverse document frequency (idf), which lessens the weight of commonly used terms while raising the weight of less frequently used words in a collection of documents, is another approach. This can be used in conjunction with term frequency to compute a term's tf-idf, or term frequency adjusted for rarity of use (the two quantities multiplied together). The tf-idf statistic is used to assess a word's significance inside a group (or corpus) of documents, such as a single book within a group of books or a single website within a group of websites.**

- **We created a Document Term Matrix (DTM) using TF-IDF to give our text data a structured format for model building. In DTM, columns represent words and rows indicate reviews.**

(c) Develop models using only the sentiment dictionary terms – try the three different dictionaries; how do the dictionaries compare in terms of predictive performance? Then with a combination of the three dictionaries, ie. combine all dictionary terms. What is the size of the document-term matrix? Should you use stemming or lemmatization when using the dictionaries? Why?

- **These three lexicons are all built using unigrams, or single words. Many English words are included in these lexicons, and each word is given a score for its positive or negative mood as well as potential emotions like happiness, rage, melancholy, and so on. The nrc lexicon divides words into categories of positive and negative, anger, anticipation, disgust, fear, joy, grief, surprise, and trust using a binary system ("yes"/"no"). The Bing Lexicon divides words into positive and negative categories in a binary approach. The AFINN lexicon rates words from -5 to 5, with lower scores suggesting more negative emotion and higher scores indicating more positive emotion.**

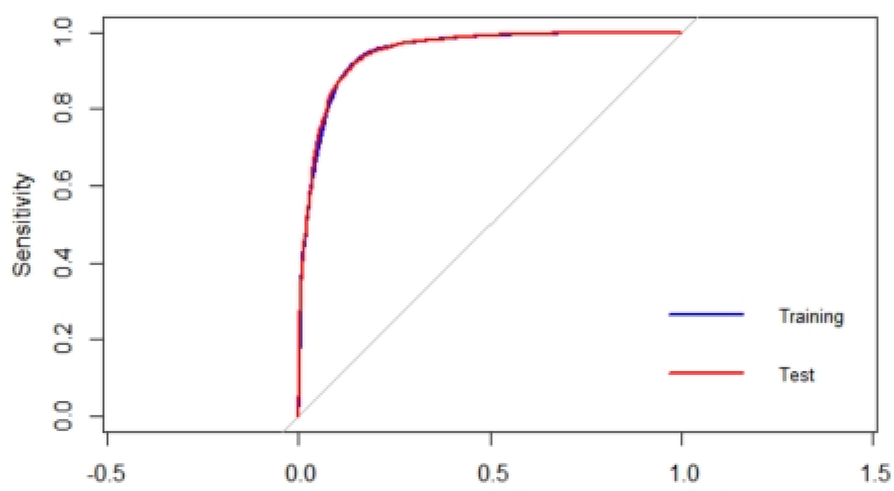
1. Bing Dictionary
1-1. It categorizes words into positive and negative category
1-2. Prior to developing models with Bing Dictionary, we have created the dataset that is to be used for developing the model.
1-3. In order to attach a sentiment to each term in the Yelp dataset that matches a word in the Bing Dictionary, we performed an inner join between the two datasets. Positive or negative sentiment is present here.

hiLo <dbl>	n <int>
-1	9656
1	27940

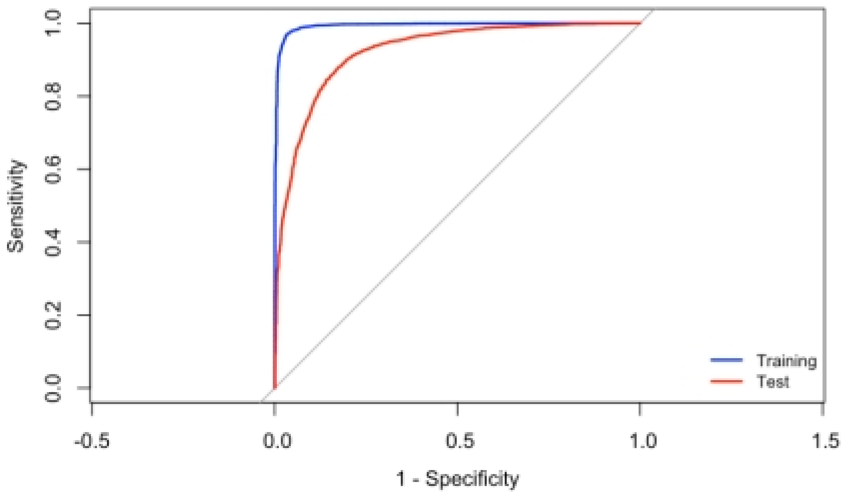
Random Forest Model :

- **We tested with various numbers of trees and mtry values for the Random Forest model. The many models we have developed and the results they produce are displayed in the table below. We used accuracy and the ROC (Receiver Operating Characteristic) Curve as the parameters for performance evaluation.**

Random Forest Model 1 ROC:



Random Forest Model 2 ROC:



Confusion Matrix for 2 Random Forest models on Training and Test Dataset

```
Number of independent variables:  1130
Mtry:                             33
Target node size:                 10
Variable importance mode:         permutation
Splitrule:                        gini
OOB prediction error (Brier s.):  0.08510545
      preds
actual FALSE TRUE
    -1  2283  197
    1     79 7441
      preds
actual FALSE TRUE
    -1  1652  792
    1    372 7184
Setting direction: controls < cases
Setting direction: controls < cases
[1] 0.6374783
      preds
actual FALSE TRUE
    -1  2402   78
    1    250 7270
      preds
actual FALSE TRUE
    -1  1929  515
    1    684 6872
```

2.NRC Dictionary

2-1. NRC Dictionary categorizes words into “yes”/”no" categories like negative,positive,disgust,anger,anticipation,surprise,joy,trust,fear and sadness.
2-2. Prior to developing models with Nrc Dictionary, we have created the dataset that is to be used for developing the model.

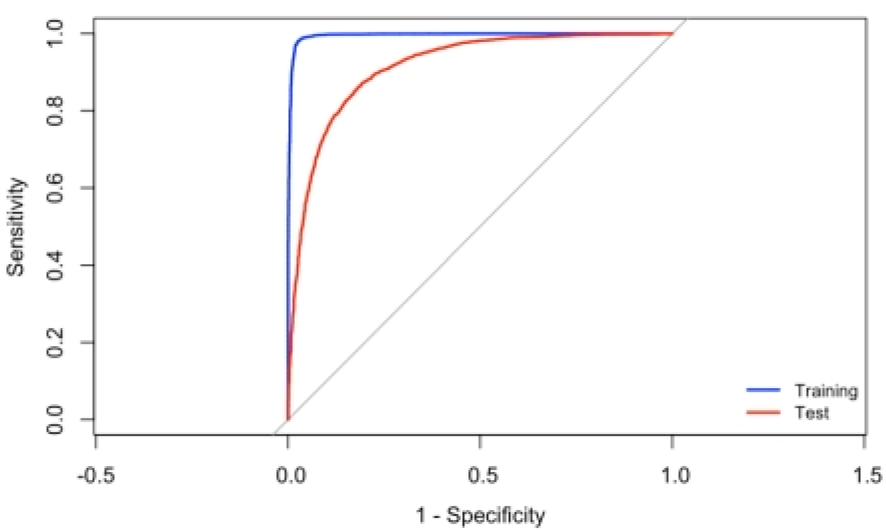
2-3. In order to attach a sentiment to each term in the Yelp dataset that matches a word in the Nrc Dictionary, we performed an inner join between the two datasets. Here, the emotions include disgust, anger, disgust, surprise, delight, joy, trust, fear, and sadness.
2-4. The inner join output consisted of words where each word was linked to several sentiments in Nrc Dictionary.

hiLo <dbl>	n <int>
-1	9889
1	28447

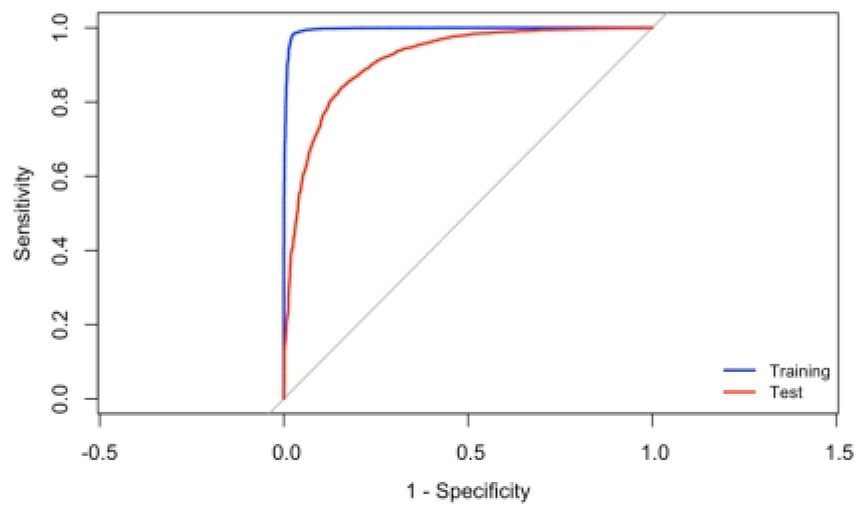
Random Forest Model :

- **We tested with various values of the mtry value and the number of trees for the random forest model. The various models we have developed are displayed in the table below along with the results they provide. Accuracy and the ROC (Receiver Operating Characteristic) Curve were taken into consideration when evaluating performance.**

Random Forest Model 1 ROC:



Random Forest Model 2 ROC:



Confusion Matrix

```
Number of independent variables:  1558
Mtry:                             39
Target node size:                 10
Variable importance mode:         permutation
Splitrule:                        gini
OOB prediction error (Brier s.):  0.09609612
      preds
actual FALSE TRUE
    -1  2343  187
    1     36 7434
      preds
actual FALSE TRUE
    -1  1501  958
    1    309 7232
Setting direction: controls < cases
Setting direction: controls < cases
[1] 0.6215601
      preds
actual FALSE TRUE
    -1  2465   65
    1    126 7344
      preds
actual FALSE TRUE
    -1  1848  611
    1    704 6837
```

3. AFINN Dictionary

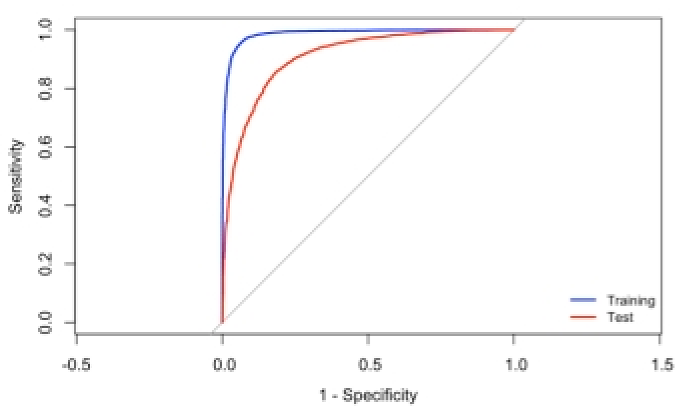
3-1. AFINN Dictionary assigns the words score ranging from -5 to 5 where positive scores indicate positive sentiment whereas negative score indicates negative sentiment.
3-2. Prior to developing models with AFINN Dictionary, we have created the dataset that is to be used for developing the model.
3-3. In order to attach a sentiment to each term in the Yelp dataset that matches an entry in the AFINN Dictionary, we performed an inner join between those two datasets. Here, the feeling is between -5 and 5.

hiLo <dbl>	n <int>
-1	9417
1	27359

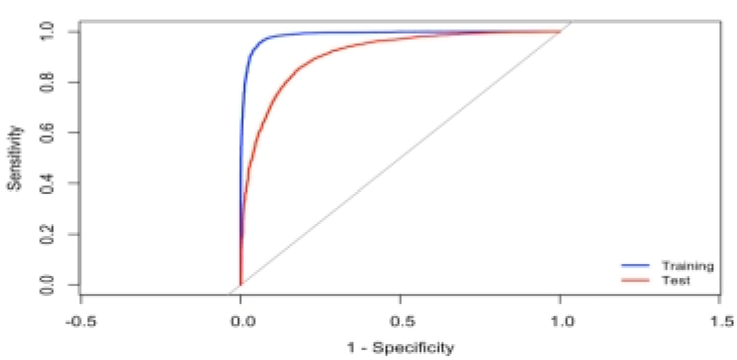
Random Forest Model :

- **We tested with various numbers of trees and mtry values for the Random Forest model. The many models we have developed and the results they produce are displayed in the table below. We used accuracy and the ROC (Receiver Operating Characteristic) Curve as the parameters for performance evaluation.**

ROC Curve for Model 1-



ROC Curve for Model 2 -



Confusion Matrix

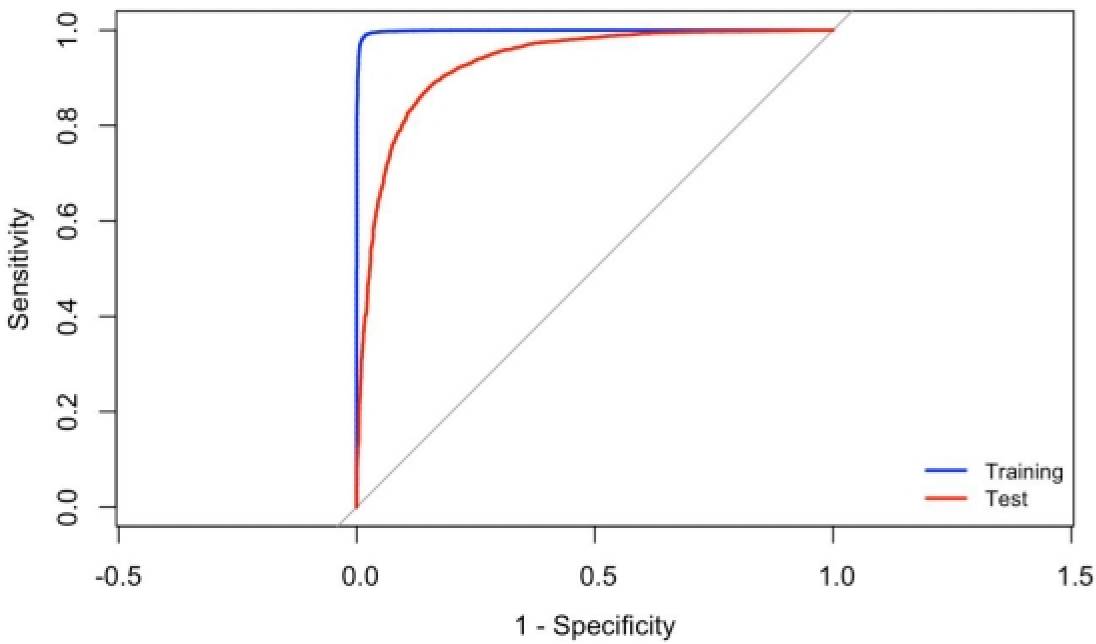
```
      1    343 7174
        preds
actual FALSE TRUE
    -1  1895  583
     1   819 6703
Computing permutation importance.. Progress: 53%. Estimated remaining time: 28 seconds.
        preds
actual FALSE TRUE
    -1  2163  320
     1   113 7404
        preds
actual FALSE TRUE
    -1  1617  861
     1   437 7085
Setting direction: controls < cases
Setting direction: controls < cases
[1] 0.6501571
        preds
actual FALSE TRUE
    -1  2330  153
     1   339 7178
        preds
actual FALSE TRUE
    -1  1908  570
     1   819 6703
```

4. Combining All three Dictionaries (bing,Nrc and AFINN)
4-1. We have produced the dataset that will be utilized to build the models before using a merged dictionary.
4-2. We merged the Matched words of all three dictionaries into a single combined dictionary matched dataset.
4-3. Each word in the single combined dictionary matched dataset could have multiple sentiments drawn from all the dictionaries.

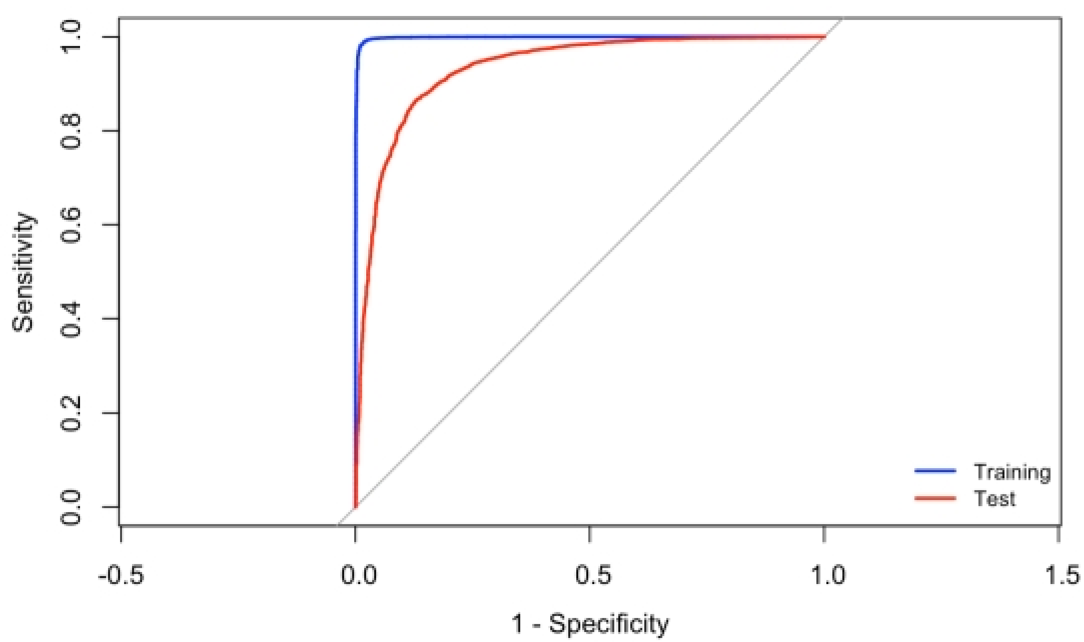
Random Forest Model :

- **We tested with various numbers of trees and mtry values for the Random Forest model. The many models we have developed and the results they produce are displayed in the table below. We used accuracy and the ROC (Receiver Operating Characteristic) Curve as the parameters for performance evaluation.**

ROC Curve for Model 1 -



ROC Curve for Model 2 –



Confusion Matrix –

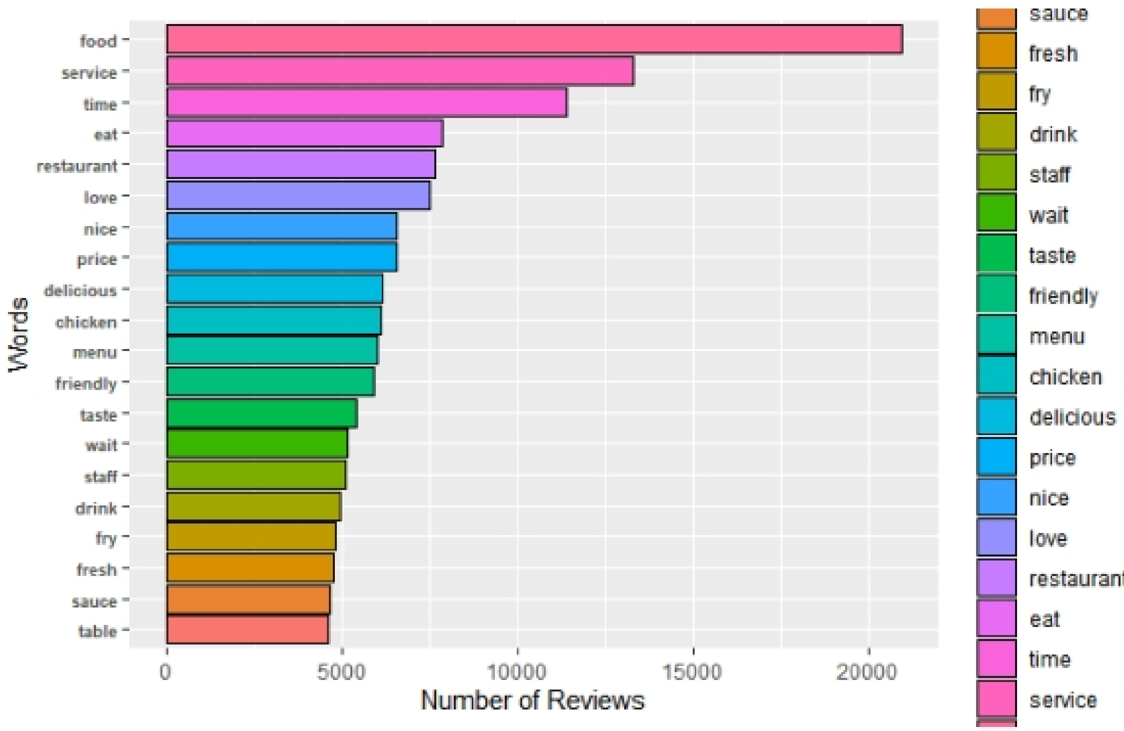
```
Number of independent variables:  2104
Mtry:                             45
Target node size:                 10
Variable importance mode:         permutation
Splitrule:                        gini
OOB prediction error (Brier s.):  0.0883768
      preds
actual FALSE TRUE
    -1  2349  141
    1     19 7491
      preds
actual FALSE TRUE
    -1  1550  902
    1    240 7308
Setting direction: controls < cases
Setting direction: controls < cases
[1] 0.6077816
      preds
actual FALSE TRUE
    -1  2443   47
    1     73 7437
      preds
actual FALSE TRUE
    -1  1869  583
    1    490 7058
```

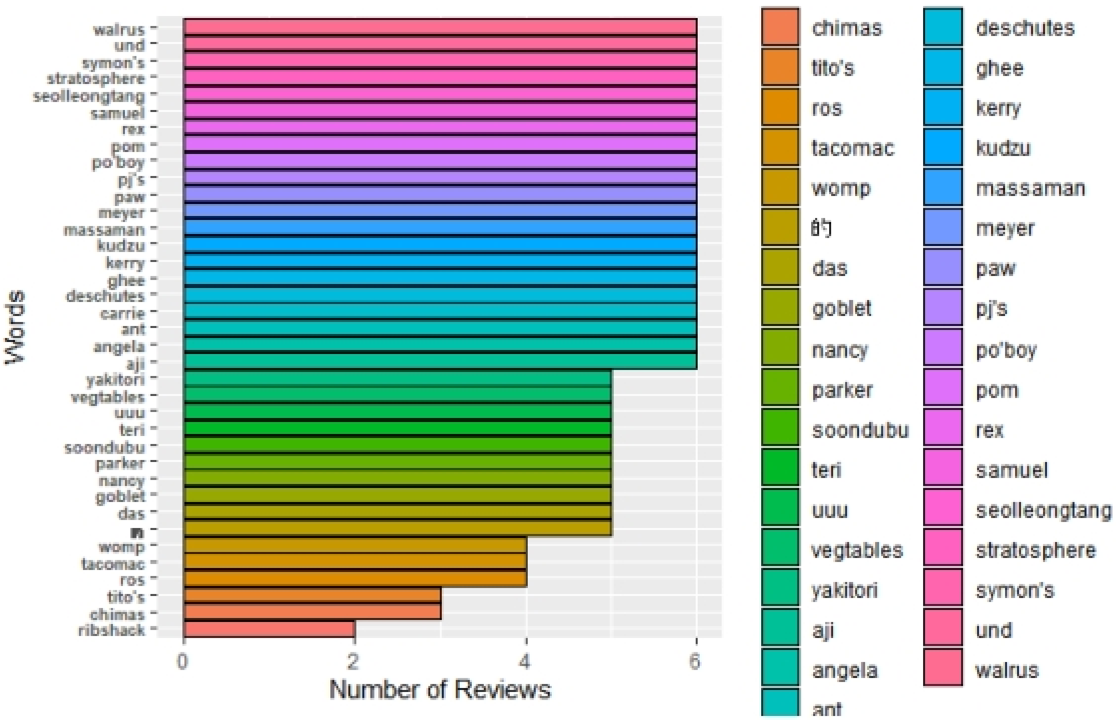
(d) Develop models using a broader list of terms (i.e. not restricted to the dictionary terms only) – how do you obtain these terms? Will you use stemming or lemmatization here, and why?

- **Prepare the dataset needed for the model before creating models using a larger list of terms. The pre-processing processes for the dataset are listed**

- **Dataset Creation**

- **We first checked the number of the reviews in which each word occurred.**
- **Then, we identified the words that appeared in the top 20 reviews as well as the last 20 reviews, and we generated graphs for each of these words. The words are arranged in descending order of the number of reviews in the graph below.**


food
service
time
eat
restaurant
love
nice
price
delicious
chicken
menu
friendly
taste
wait
staff
drink
fry
fresh
sauce
table
Words
0
5000
10000
15000
20000
Number of Reviews
sauce
fresh
fry
drink
staff
wait
taste
friendly
menu
chicken
delicious
price
nice
love
restaurant
eat
time
service



- **D-1. We then removed the words which occur in greater than 90% of the reviews and less than 30% of the reviews**
- **D-2.The terms from the Yelp dataset that match the reduced words were then obtained by performing an inner join on the reduced words and the Yelp data set. The information from the Yelp dataset will be in the matched data, which will consist of the reduced terms.**
- **D-3.The Document Term Matrix, which indicates the frequency of terms for each word that exists in the review, was then calculated for the matching dataset. The reduced dataset's words were shown as columns in the Document Term Matrix together with two new columns (Reviews Id and Star Rating), and the rows represented the values of the tfidf column.**
- **D-4.The next step is to filter out Star rating 3 as it is neutral rating and we wish to go ahead with positive Star rating(4,5) and Negative Star rating(1,2).**
- **D-5. We then replaced the NULL values in Document Term Matrix with 0's.**

- **Here, stemming is not being employed. Lemmatization was only employed because it preserves the word's meaning when done correctly. However, stemming affects the meaning of the term by removing the prefix and suffix.**

(e) Compare performance of the models. How does performance here relate to that from Question 4 above. Explain your findings (and is this what you expected).

- **Performance of the Models :**

- **Random Forest Model :**

- **We tested with various numbers of trees and mtry values for the Random Forest model. The many models we have developed and the results they produce are displayed in the table below.**

```
 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.2732
            Specificity : 0.8806
         Pos Pred Value : 0.9160
         Neg Pred Value : 0.2027
             Prevalence : 0.8266
         Detection Rate : 0.2258
   Detection Prevalence : 0.2465
      Balanced Accuracy : 0.5769

       'Positive' Class : -1

Confusion Matrix and Statistics

      preds
actual   -1    1
    -1 2258  207
    1  6008 1527

               Accuracy : 0.3785
                 95% CI : (0.369, 0.3881)
    No Information Rate : 0.8266
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0662

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.2732
            Specificity : 0.8806
         Pos Pred Value : 0.9160
         Neg Pred Value : 0.2027
             Prevalence : 0.8266
         Detection Rate : 0.2258
   Detection Prevalence : 0.2465
      Balanced Accuracy : 0.5769

       'Positive' Class : -1

   user  system elapsed
 80.196   3.858  86.934
      predicted
actual   -1    1
    -1 2465    0
    1     0 7535
      predicted
actual   -1    1
    -1  653 1812
    1    54 7481
```

- **We computed the accuracies with and without Laplace Smoothing using Naive Bayes Models for both the test set and the training dataset. The precision in this situation was really poor (less than 40 percent). The fact that all of the features in naïve bayes are taken to be independent of one another is the most plausible reason for this. In the case of text mining, this does not take place. Since they can handle non-linearities in the data, SVMs that use the radial basis function kernel are more likely to perform better in terms of performance.**

- **Naive Bayes performs best when the attributes are unrelated to one another, which is uncommon in real life. However, it still functions effectively even when the characteristics are not independent. When the classes (binary and many) are clearly distinct from one another, naive Bayes performs quite well. Additionally, after comparing all the models for the three dictionaries, the combined dictionary, and a larger list of terms, we find that the SVM with BING Dictionary has the greatest test accuracy of all the models (89%). Our best model is thus that one.**

- **SVM has the highest test accuracy, so we've chosen it as our best model even though Random forest and SVM both have test accuracy in the range of 84–88%.**

6. Consider some of the attributes for restaurants – this is specified as a list of values for various attributes in the ‘attributes’ column. Extract different attributes (see note below).

(a) Consider a few interesting attributes and summarize how many restaurants there are by values of these attributes; examine if star ratings vary by these attributes.
(b) For one of your models (choose your ‘best’ model from above), does prediction accuracy vary by certain restaurant attributes? You do not need to look into all attributes; choose a few which you think may be interesting, and examine these.

Note: for question 6, you will consider the values in the ‘attribute’ column. This has values of multiple attributes, separated by a ‘|’. Further, some of the values, like Ambience, carry a list of True/False values (like, for example, Ambience: {'romantic': False, 'intimate': False, 'classy': False, 'hipster': False, …}. Care must be taken to extract values for different attributes. You can consider developing a separate dataframe with review_id, attribute, and then process this further to extract values for the different attributes.

- **After building models for Random forest, Naive Bayes, and SVM using all three dictionaries, we concluded that SVM is the best model.**

- **Because Naive Bayes performs badly on both Training and Test data, we decide against using it. Less than 80% of words in the combined dictionary and all three dictionaries are accurate.**

- **Because Random Forest's accuracy on the training dataset is 97 percent, it overfits for all combinations of the number of trees and mtry. For this reason, we decided against using it. The following qualities were looked at:**

	review_id	attName	attValue
1	-K5z7DzXHJgEC1tsTLfFeA	Alcohol	full_bar
2	-K5z7DzXHJgEC1tsTLfFeA	Ambience	{'romantic': False, 'intimate': False, 'classy': False, 'hipster': Fa...
3	-K5z7DzXHJgEC1tsTLfFeA	BusinessAcceptsCreditCards	True
4	-K5z7DzXHJgEC1tsTLfFeA	BusinessParking	{'garage': True, 'street': False, 'validated': False, 'lot': False, 'v...
5	-K5z7DzXHJgEC1tsTLfFeA	GoodForKids	False
6	-K5z7DzXHJgEC1tsTLfFeA	GoodForMeal	{'dessert': False, 'latenight': False, 'lunch': False, 'dinner': True...
7	-K5z7DzXHJgEC1tsTLfFeA	HasTV	True
8	-K5z7DzXHJgEC1tsTLfFeA	NoiseLevel	average
9	-K5z7DzXHJgEC1tsTLfFeA	OutdoorSeating	True
10	-K5z7DzXHJgEC1tsTLfFeA	RestaurantsAttire	casual
11	-K5z7DzXHJgEC1tsTLfFeA	RestaurantsDelivery	False
12	-K5z7DzXHJgEC1tsTLfFeA	RestaurantsGoodForGroups	True
13	-K5z7DzXHJgEC1tsTLfFeA	RestaurantsPriceRange2	3
14	-K5z7DzXHJgEC1tsTLfFeA	RestaurantsReservations	True
15	-K5z7DzXHJgEC1tsTLfFeA	RestaurantsTableService	True
16	-K5z7DzXHJgEC1tsTLfFeA	RestaurantsTakeOut	False
17	-K5z7DzXHJgEC1tsTLfFeA	WheelchairAccessible	True
18	-K5z7DzXHJgEC1tsTLfFeA	WiFi	paid
19	2tjghSImOPf4A9L4zhByRQ	Alcohol	full_bar
20	2tjghSImOPf4A9L4zhByRQ	Ambience	{'romantic': False, 'intimate': False, 'classy': False, 'hipster': Fa...
21	2tjghSImOPf4A9L4zhByRQ	BusinessAcceptsCreditCards	True
22	2tjghSImOPf4A9L4zhByRQ	BusinessParking	{'garage': True, 'street': False, 'validated': False, 'lot': False, 'v...
23	2tjghSImOPf4A9L4zhByRQ	GoodForKids	False
24	2tjghSImOPf4A9L4zhByRQ	GoodForMeal	{'dessert': False, 'latenight': False, 'lunch': False, 'dinner': True...
25	2tjghSImOPf4A9L4zhByRQ	HasTV	True
26	2tjghSImOPf4A9L4zhByRQ	NoiseLevel	average
27	2tjghSImOPf4A9L4zhByRQ	OutdoorSeating	True
28	2tjghSImOPf4A9L4zhByRQ	RestaurantsAttire	casual
29	2tjghSImOPf4A9L4zhByRQ	RestaurantsDelivery	False
30	2tjghSImOPf4A9L4zhByRQ	RestaurantsGoodForGroups	True
31	2tjghSImOPf4A9L4zhByRQ	RestaurantsPriceRange2	3
32	2tjghSImOPf4A9L4zhByRQ	RestaurantsReservations	True
33	2tjghSImOPf4A9L4zhByRQ	RestaurantsTableService	True
34	2tjghSImOPf4A9L4zhByRQ	RestaurantsTakeOut	False
35	2tjghSImOPf4A9L4zhByRQ	WheelchairAccessible	True

	amb	n
1	'upscale'	135
2	character(0)	4139
3	'trendy'	1504
4	'casual'	29018
5	{'romantic'	659
6	'divey'	1073
7	c(" 'hipster'", " 'casual'")	158
8	c(" 'trendy'", " 'casual'")	127
9	'classy'	802
10	c(" 'classy'", " 'trendy'", " 'casual'")	40
11	'intimate'	235
12	c(" 'classy'", " 'upscale'")	516
13	c(" 'intimate'", " 'classy'")	81
14	c(" 'divey'", " 'casual'")	147
15	c(" {'romantic'", " 'intimate'", " 'casual'")	45
16	'touristy'	466
17	c(" 'upscale'", " 'casual'")	38
18	'hipster'	432
19	c(" 'hipster'", " 'touristy'", " 'trendy'", " 'casual'")	145
20	c(" 'touristy'", " 'casual'")	112
21	c(" {'romantic'", " 'classy'", " 'trendy'", " 'upscale'")	35
22	c(" {'romantic'", " 'intimate'", " 'classy'", " 'upscale'")	44
23	c(" 'classy'", " 'trendy'")	136