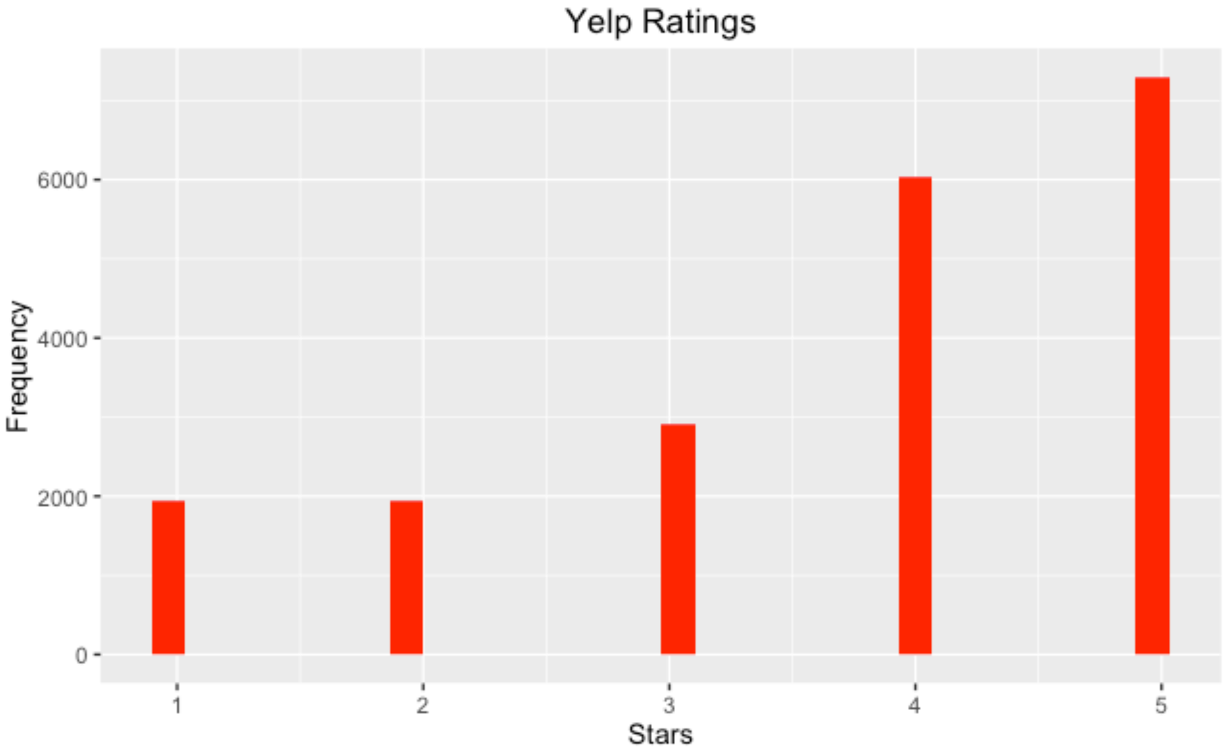
Yelp Data Analyis.
Justin Gage

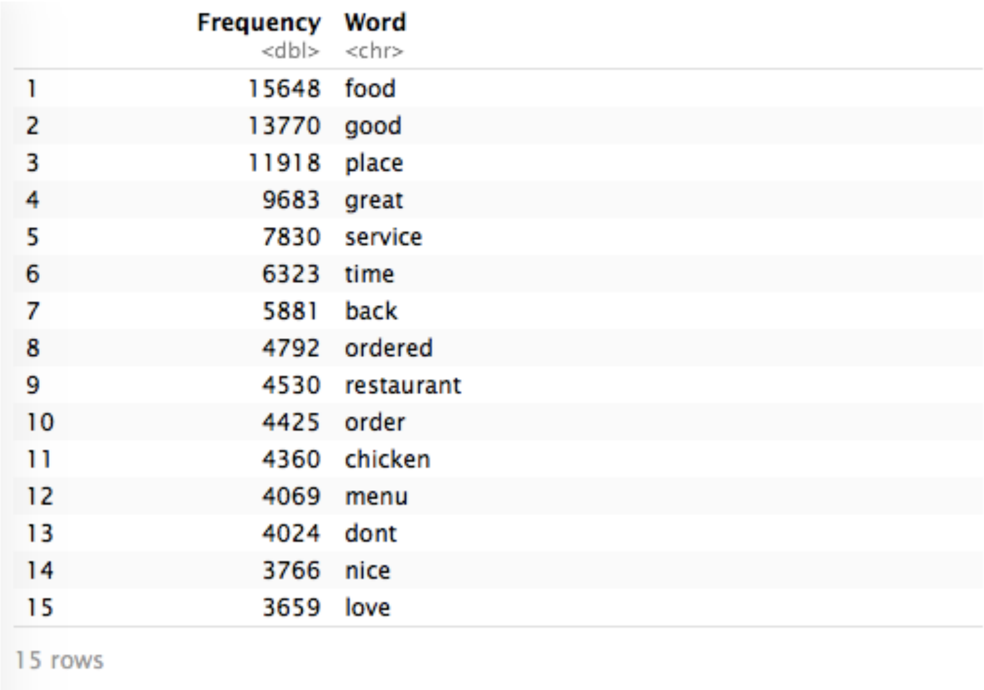
Here's some analysis of the distribution of Yelp ratings for restaurants in our data set. First, a histogram of rating stars.



And some summary statistics:

- The average number of reviews per restaurant is 2.438968.
- The average user in the sample contributed 1.1614 reviews.
- Restaurants that have been marked as Good For Lunch received 7,724 reviews, while restaurants without the marking received 12,417.
- However, even with the higher volume, the average ratings are similar: 3.74 starts for restaurants that aren't Good For Lunch, and 3.72 for ones that are.

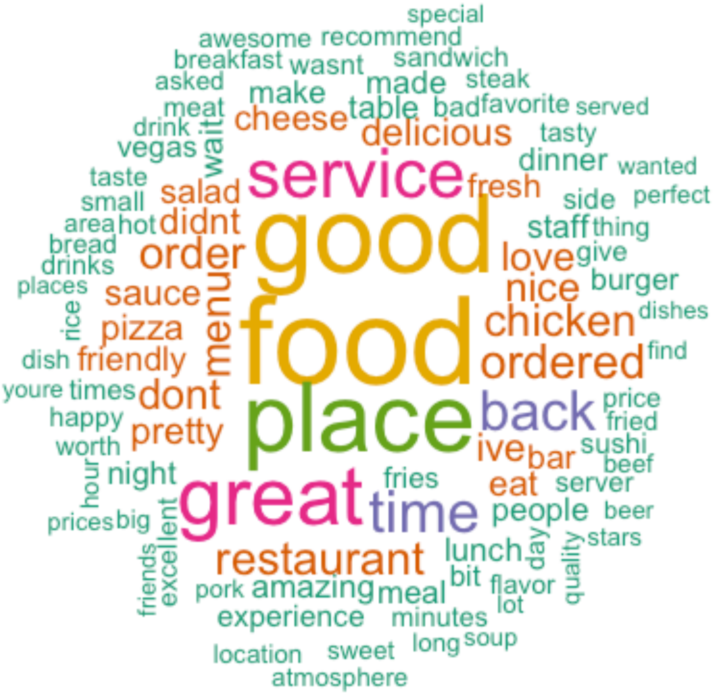
Here are the 15 most commonly occurring words among all reviews.



	Frequency	Word
	<dbl>	<chr>
1	15648	food
2	13770	good
3	11918	place
4	9683	great
5	7830	service
6	6323	time
7	5881	back
8	4792	ordered
9	4530	restaurant
10	4425	order
11	4360	chicken
12	4069	menu
13	4024	dont
14	3766	nice
15	3659	love

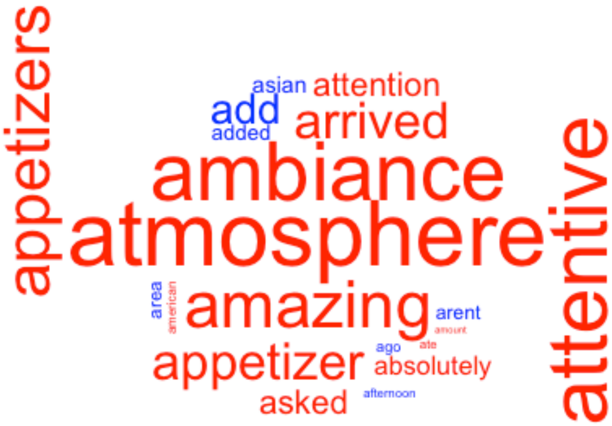
15 rows

And here’s a word cloud of the most commonly appearing words in reviews.



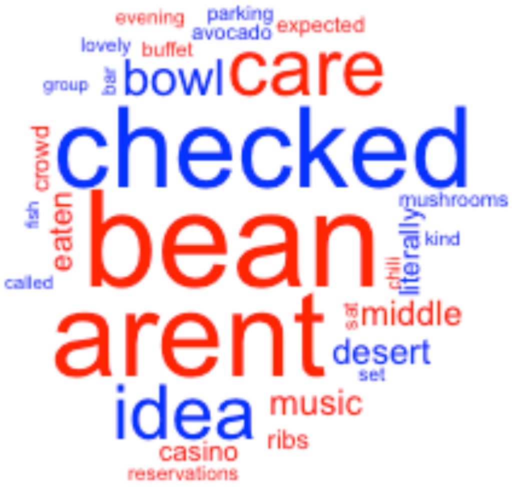
Question 3.

Based on the dimensions of the Document Term Matrix, there are 44,738 unique words in our reviews. Here’s a word cloud where size corresponds to strength of correlation with whether a restaurant is good for lunch or not. Red words have a negative correlation, and blue words have a positive one.



To predict whether or not a restaurant will be good for lunch, we can run a logistic regression against the Good For Lunch variable, where the predicting variables are all of the unique words in the Document Term Matrix.

Here’s a word cloud that contains the words with the top 15 positive and negative words (in blue and red, respectively).



To see how well our model performs, we can select a probability threshold – we'll use 0.9 to be safe. When comparing the predicted values to actual GoodForLunch values with this probability threshold, we get a mean of 0.99375 – pretty darn good!

When predicting our test data, the results aren't as promising – holding the threshold, the predicted values are only correct 52.5% of the time. As is often true in Machine Learning, our model performs very well on the data we fit it on, but isn't very strong on new test data. This could be due to overfitting.