Title

Does Yelp Friendship Influence Reviews?
Patricia Tressel, November 18, 2015

Introduction

On social media platforms like Facebook, the friend relationship is at the core of the interaction between users, and clearly affects user actions on the site. But on Yelp, where the primary function is reviewing and finding businesses, does the friend relationship do anything at all?

Using the Yelp Challenge data, http://www.yelp.com/dataset_challenge, let's see if we can determine whether friendship affects the most important active user behavior – reviewing a business. We'll limit this to looking at the star rating, and ask:

Does a negative or positive review by a user affect later reviews by their friends, by an amount different from any change in non-friend reviews?

This is also an exercise in attempting to do analysis on relational data, using a relational database, so we'll conclude with some tips and tricks. The code is an unholy mixture of bash, Python, SQL, and R, and may be found at: https://github.com/ptressel/yelp_friend_analysis

Methods and Data

Overview and terminology

We'll compare the ratings of a business by a user's friends, before and after the user posts a review. Because a change in ratings might be due to an actual change in the business quality, we'll use the change in non-friend reviews before and after the same date as a baseline. That is, we'll compare the average rating of each business by friends of each user before and after that user's review, versus the change in non-friend ratings, and see if friend ratings shift in the direction of the user's rating more than do non-friend reviews.

But first – a little information about the friend relationship on Yelp.

- It is mutual – a friend request must be accepted to take effect.
- It is not the same as following (being a fan of) a user – that is unidirectional.
- Only 52.5% of Yelp users in the dataset even have friends.
- Previously, reviews by friends were shown prominently in the list of reviews of a business, but that was removed in a redesign of the Yelp website on or before 2014-02-11.

The Yelp Challenge data includes the following fields that are relevant:

- For each user, their list of friends.
- For each review, the author, business, rating, date.

These are refactored into a relational database, starting with:

- friend table, relating pairs of users
- review table, relating user and business, with rating, date

Additional tables and indices are used to support query construction and optimization, and hold intermediate and final results.

This terminology will be used to describe the question and results:

- *Target review* means the review of which the influence is being measured.
- *Target reviewer* means the author of the target review.
- *Friend* or *non-friend* means other users who review the same business, who are either friends or not of the target reviewer.
- *Before* and *after* refer to reviews of the same business that occur on dates <= or > the target review.

Description of the question

For a given target review, compute the mean rating of the same business by friends and non-friends of the target reviewer, for reviews before the target review (which would not have been influenced by it) and after (which might have been influenced by it).

For friends and non-friends separately, compute the *difference in the mean rating, before and after the target review*, taken in the *same direction as the difference between the mean rating before and the target review rating.*

The above query produces two values for each distinct target reviewer and business that had a complete set of the four friend / non-friend, before / after cases. Again, one value represents the possible influence of the review on friends, and the other on non-friends.

The null hypothesis is that there is no difference between the friend and non-friend values, indicating no influence of friendship on reviews.

Since we have oriented the difference in the direction of the target review, this will let us look beyond just whether there is a difference or not. We are computing a shift in ratings toward the target review. If friends of the target reviewer react more strongly than non-friends toward the target review, then we would expect a positive difference between the friend and non-friend values, but if they react less strongly than non-friends, or even react against the target review rating, we would expect a negative difference of differences. So to see the size of the effect, if any, we can compute the difference between the friend and non-friend differences, and take the mean.

Restrictions on included data

In order to have paired before and after results for friends and non-friends, we must have samples that fall into all four cases. Some restrictions are intended to cut down the size of the queries, or assure that most results will have at least a few reviews by other reviewers falling into all four cases:

- Only include businesses with more than a minimum number (50) of reviews.
- Only include target reviewers with more than a minimum number (50) of friends. The "friend" case is much more sparse than the non-friend case, so without this restriction, there would be a lot of wasted query time, as low-friend target reviewers would have empty sets for the two friend cases.
- To reduce dependence on samples that may be idiosyncratic, require at least a small minimum (3) of other reviews for all four cases.
- To simplfy the queries, although one user might review a business more than once, only the first review by a user is included as a target review.

The number of complete samples remaining was 41097. Restricting to only reviews after the site reorganization, the number of complete samples was 999.

There is no restriction on the other, non-target, reviewers. They are not required to have any minimum number of friends. Note that it is the influence on these other reviewers that is being measured, so we do not want to require that they have any friends at all.

Statistical test

An appropriate test is Wilcoxon's signed rank test: We have paired data, coming from non-normal distributions, and want to know if the distribution of the difference between the pairs is centered on zero. Recall the ratings are

discrete values in a short range, so their differenences are also discrete and bounded, so their distributions cannot be normal.

Results

Drumroll...

Wilcoxon's signed rank test for the entire date range rejects the null hypothesis with p-value 3.73×10^{-179}.

Well, that was unexpected.

The p-value is effectively zero. So there is apparently a statistically significant difference between the review ratings of friends versus non-friends, before and after a user's review.

Dependence on prominence of friend reviews

Is this due to showing friend reviews at the top of the review list for a business? That is, does the effect vanish after the Yelp site reorganization, after which that feature is gone? There are not nearly as many samples for this case, so we don't expect as definitive a result... but is there any effect?

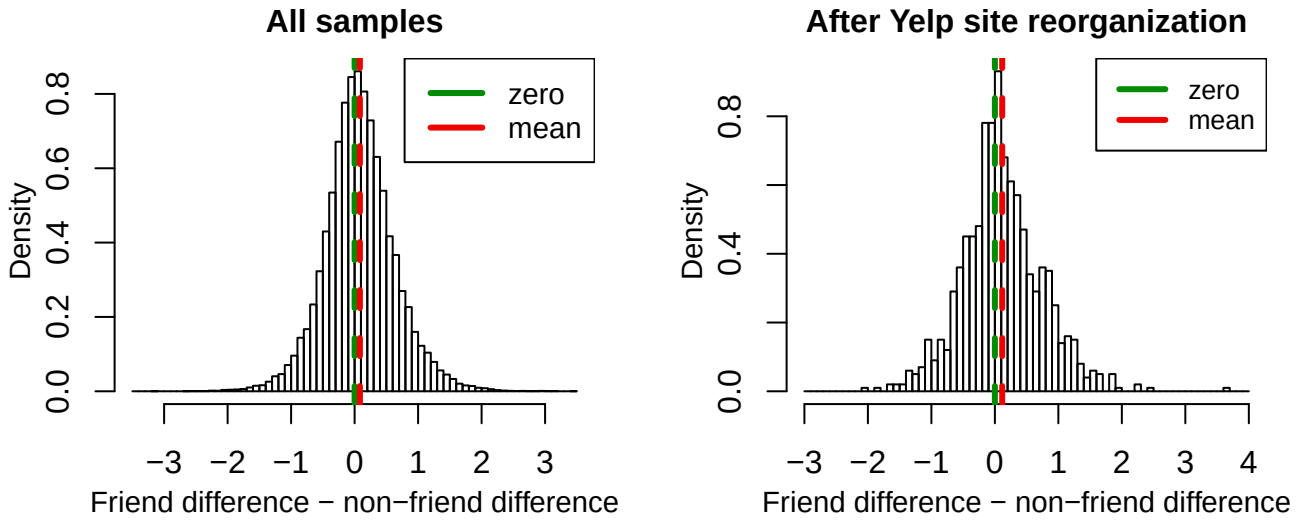
Wilcoxon's signed rank test still rejects the null, in this case with p-value 1.77×10^{-7}.

So the effect persists even without featured friend reviews.

Size of the effect

So, it's statistically significant... but is it meaningful? That is, is the effect so tiny that it's irrelevant? The mean difference between the friend and non-friend changes in ratings over all dates is 0.083. Restricted to reviews after the Yelp site reorganization, the mean is 0.115.

To visualize the size of the effect, we'll histogram the differences between the friend and non-friend before / after differences, and show where the mean difference of differences is, compared to zero.



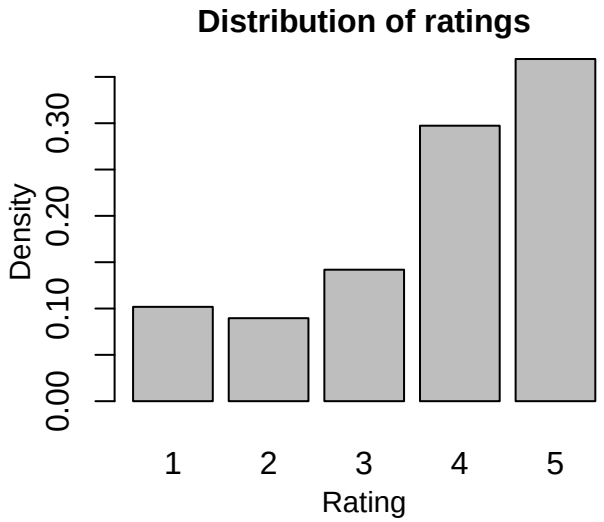
So both are about 0.1. That doesn't seem large, but what is an appropriate scale against which to evaluate the effect size? The range of a mean of ratings is 1-5, so the before / after difference has a range of -4 to 4 and the range of the difference between two before / after differences is -8 to 8, so has a span of 16. On that scale, looking at just the smaller effect for all dates, the mean is shifted from 0 by about 0.08 / 16 = 0.005 or 0.5%. But is that

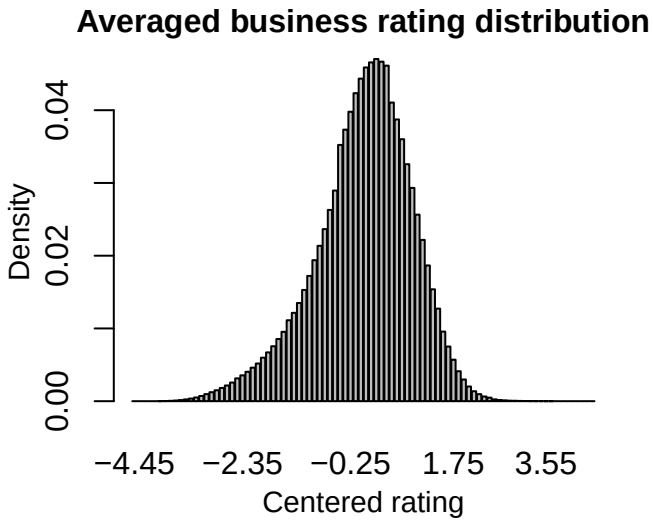
fair? There aren't even any samples out at the ends of that range – the actual range is about 7. Even then, the tails are very low probability, as we can see in the plot.

Another way we can see that the scale shouldn't be based on the full 1-5 range of ratings is that it is not based on the actual range *for each business*. If we histogram the ratings irrespective of business, then yes, we see a they are spread over all rating values with no central peak, and with some preference for higher ratings.

But that does not account for the differences in quality of businesses. That may cause such a spread merely because each different business merits a different rating, not because users are fickle in their reviews. If, instead, we compute the distribution of ratings *for each business separately*, then we can adjust for different business qualities by *aligning the means of those distributions* before averaging them.

Here are the raw distribution of ratings, and the averaged per-business distributions, where the mean of each is shifted to zero before averaging. The standard deviation of the latter is 0.94.





The traditional scale for a distance from the mean of a distribution would be the standard deviation deviation of the same distribution. Here, that is 0.552, and with that as the scale, *the relative effect size is* 0.15 *or* 15%*!*

Discussion

We have a very highly statistically significant effect, that may or may not be of a meaningful size, depending on what scale we use to measure it against. Using the traditional scale, the effect size appears at least somewhat meaningful. Given this, would it be worthwhile for Yelp to promote friendship? or drop it?

- It's good if it increases engagement. Review influence may indicate it does.
- Perhaps not good if it skews ratings.

Are there hints as to what might lead to such an effect? Standard caution applies: we can't distinguish causation from correlation. Here are some possibilities, that might be avenues for further investigation:

- Before the site reorganization, users saw reviews by friends featured prominently when they viewed a business page. After the change, users may still see new reviews by friends in their news feed, or in a friend's profile.
- Users may send friend requests to users whose reviews they like.
- People who choose to be friends may have similar tastes.
- Yelp friends may be friends "in real life".
- There are in-person Yelp events. Users who meet there may send friend requests to each other.

Advice for using a relational database and SQL

This project appeared infeasible at first – an initial attempt at a query to compute some preliminary results had to be killed after running for 24 hours. Now, the entire process takes less than half a day on my not-high-end laptop. Here are recommendations based on that experience. Some of these are specific to MySQL.

The main bottleneck will be disk I/O:

- Do not include any unnecessary information in tables. Pull out only the columns needed for a query into a new table, if needed.
- Replace long identifiers by sequence number keys.
- Reduce all fields to the minimum width.
- Do preliminary analysis to identify data that can be excluded, e.g. because it provides little information. Filter these from queries or remove them from tables.

The database system's query optimizer may not produce a good execution plan:

- Look at the query plan with EXPLAIN.
- Make sure that everything that needs an index has one.
- Examine the query for subtle inefficiencies.
- Control the plan explicitly with hints and subqueries.
- Control the plan by breaking up the query into separate steps or by making temporary tables.
- Use a different database system that has a better query planner.

General advice:

- Break up large queries into batches – batched queries may run significantly faster in total than the query run as a whole. For instance, split keys into ranges. The field used for the range must be indexed. If the table has multiple fields in its primary key, add a sequence number field with an index to split into batches.
- Use inequalities or BETWEEN to specify batch ranges, not LIMIT, which is slow.
- A hash table index is not useful for range or inequality queries – make sure the database system is using a B-tree or other ordered index.
- Some computations are more efficiently done in a procedural language.
- If you need hierarchical or recursive queries, switch to a database system that supports recursive common table expressions. That does not include MySQL...

Tips for muliple languages in knitr:

- To pass information between code chunks, write it out to disk or database.
- For disk files, use a commonly supported format such as CSV or JSON.
- Don't rely on knitr caching in R if work is done in another language between two R chunks.
- Have a look at the knitiron tool for using IPython and matplotlib with knitr.

And, a trick: In a procedural language that has iteration, it is easy to perform a task using multiple different sets of parameters. To simulate this in SQL:

- Construct a table to hold the parameter sets, with a column for each parameter, and a row for each set of parameter values needed. Include this table in the FROM clause, and use its columns in constraints in the WHERE clause.
- If you need all combinations of some parameter values, rather than just specific sets of values, make one table for each parameter, with a single column, and include all values for that parameter as rows. Include all of these tables in FROM and use their columns in WHERE. An example is included in the code for this project.