Yelp Challenge Project Report

Tingting Zhang, Yi Pan
University of Washington

1. Introduction

In this project, we run some interesting queries on the Yelp dataset. Yelp dataset (https://www.yelp.com/dataset_challenge/) includes 11,537 businesses, 8,282 check-in sets, 43,873 users, and 229,907 reviews in json format from the greater Phoenix, AZ metropolitan area. Based on such a rich dataset and inner relationship between different types of data, we expect to figure out some interesting facts and further analyze and visualize some user and business behaviors through integrating different query results.

2. Preliminaries

The project is divided into two steps, storing json data in a relational database system Postgres and running simple queries to find some useful facts about the businesses, users, and reviews, and further analyze the relationship between a business's success and its geographic location, the weekly and daily cycle of a business check-in information, the behaviors of nice and mean users, and what makes a review useful through data visualization and statistical analysis.

2.1. Database Schema Design

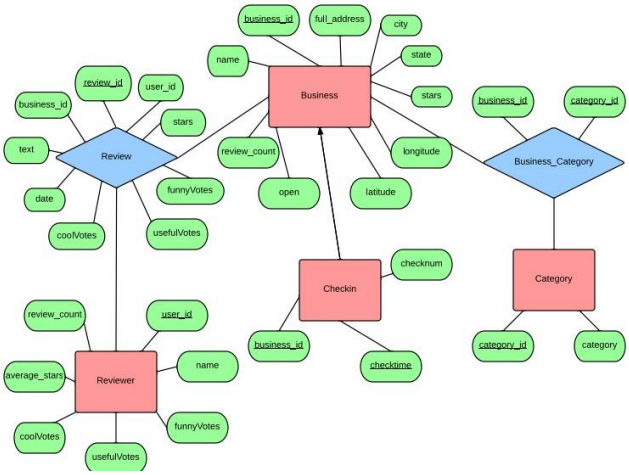

Figure 1 ER Diagram for yelp dataset

We designed a relational database schema (Figure1), created a database in Postgres, parsed the json data using a json-simple package, and populated the json data into the database. In the database schema, we made some changes to the original json data structure to reduce duplicates and make data more accessible for query and analysis. In the json format, each business has a category attribute, which stores a list of categories the business belongs to. Each category is repeated many times in different business records. Given such a many-to-many relationship between Business and Category, we created three tables, Business, Business_Category, and Category, to reduce duplicate data and make data more consistent. Apart from these three tables, we created a Reviewer table to store the user information, Review table to store the review information, and Checkin table to store check-in information for all the businesses. We also flatted some data to make it more accessible for query and analysis. For example, both reviewer data and review data has a votes attribute, which stores three kinds of votes in the form of {"votes": {"funny": int, "useful": int, "cool": int}. We flatted the votes into three attributes, funnyVotes, usefulVotes, and coolVotes, so that the votes for each type can be easily fetched. What's more, the original check-in info is stored in the form of {"checkin_info": {"11-3 (checkin_starttime, which means Tuesday 11:00am)": int (checkin count), …}}. This nested structure is confusing, therefore, we decided to revert it into a more accessible and flatter way with schema <business_id, checktime, checknum>. We converted the checkin_starttime from hour-day (i-j) to hours (24*j+i) and created two attributes, checktime and checknum to store the check-in start-time and check-in numbers for each business.

2.2. Import Data into Postgres

We implemented the data population with postgres database and populated the data with a JDBC connection. When reading in the json data object, it is important to be careful about the data type: it could be a nested type like JSONObject which is a hash table or basic data type String, Double, Long, Boolean. The strategy to populate data into schema could be called "insert and update". The program firstly inserts each entry with its primary key, which is usually a business_id, user_id or etc. Then it updates the tuple with other attributes such as "stars" and "votes" based on the primary key (set the where condition). One tricky part is how to build the Category table from the business data object. For each category of a business, the program searches in the current category table to find whether it exists. If not, generate a new category id and insert it into category and business_category table. If it does exist, get the category id and insert the corresponding tuple into the business_category table.

As for runtime, it takes approximately 30 minutes to populate every data into the database.

After json data is populated into the database, basic statistics on data are collected. The database includes 11,537 businesses, 508 categories, 30891 business_category sets, 43,873 users (reviewers), 229,907 reviews, and 262,764 check-in sets. The relational data is consistent with json data, except that the relational database separates category from business, and flats check-in information into checktime and checknum for each check-in information pair in the original json data.

3. Data Analysis

After the dataset is populated into the relational database, we integrate some simple and advanced query results as well as apply statistical methods to analyze how much of a business's success is related to location, when is a business most busy weekly and daily, whether "mean" and "nice" users behave in the same way, and what makes a review useful.

3.1. Descriptive analysis

3.1.1 How much is a business's success is really location, location, location?

We use business review counts, check-in counts, and stars, as three proxies for a business's success, geographic information (latitude and longitude) as its location, and want to see how these two variables correlate with each other. Given the importance of a business's location to its success will depend on the business category, we choose three relative abundant business categories, restaurants, shopping malls, and nightlife, to explore whether there exist some patterns for each category. Specifically, we plot the 50 most successful businesses (blue dots) and 50 least successful businesses (red dots) in the same graph and find the relationship between a business's success and its location differs from category to category.

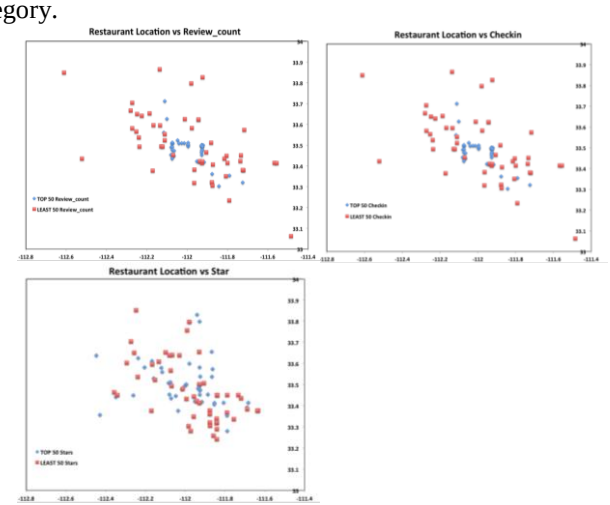

Figure 2 Scatter plot of locations of restaurants with top 50 and least 50 review counts/checkin counts/ stars

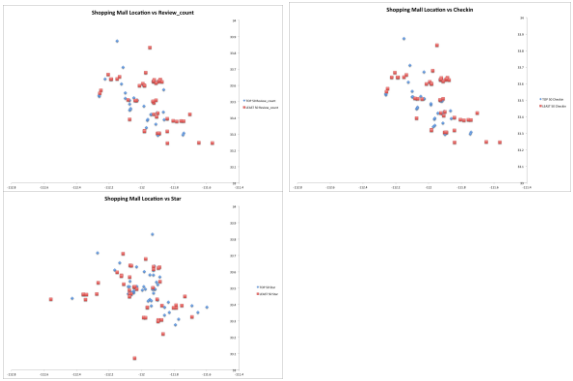

Figure 3 Scatter plot of locations of shopping malls with top 50 and least 50 review counts/checkin counts/ stars

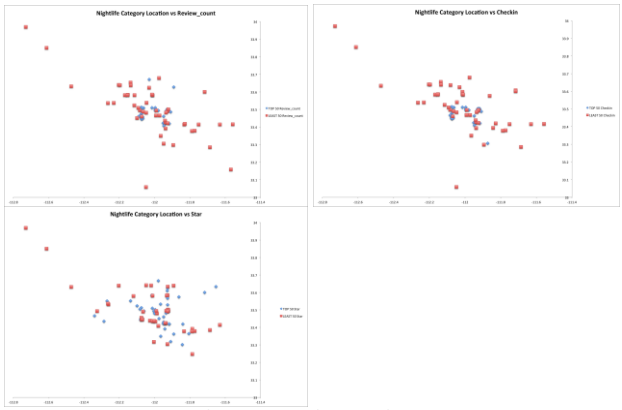

Figure 4 Scatter plot of locations of nightlife with top 50 and least 50 review counts/checkin counts/ stars

For restaurant category (Figure 2), the businesses with the most review counts and check-in counts are geographically clustered in a central region, while the other businesses seem to scatter randomly. However, this trend disappears if we use business stars as proxy for success. The central cluster area might have convenient transportation and entertainment places around, which attract many customers to eat at this area, such as downtown.

For shopping mall category (Figure 3), the most successful and least successful businesses seem to be randomly distributed in the region. Considering that shopping malls usually have chain stores, some of them will be located in central areas while others are located in distant areas. The specific location of one store might doesn't make a big difference for its success.

For nightlife category (Figure 4), the nightlife entertainment with the most review counts and check-in counts are even more geographically clustered in the central region than restaurants, while the other nightlife entertainment seem to scatter randomly. The most successful nightlife entertainment diffuses from the central area if the business star is used to approximate business success.

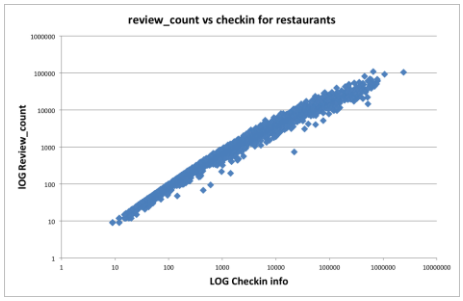

Figure 5 Log-log linear relationship between review counts and check-in counts

Through data visualization and analysis, we don't find a close connection between a business's success and its location. To make it realistic, the relationship between a business's success and its location depends on business category. For both restaurants and nightlife category, it seems the most successful businesses tend to geographically cluster in the central area while the others scatter randomly. However, there

are no such clustering phenomena for shopping mall category. The three proxies for business success, business review counts, checkin counts and stars, will show different patterns. Particularly, as shown in Figure 2-4, review counts and checkin counts represent very similar businesses. After we plot the review counts and checkin counts of all businesses in Figure 5, we find these two indices of business success are log-log linearly correlated.

3.1.2 When is a business most busy weekly and daily?

Restaurant category is chosen to explore a business's weekly cycle and daily cycle, because restaurants have the most records in the yelp business table. Common sense tells us that restaurants tend to be busiest during the weekend night. In order to test the above idea, we select all the restaurant check-in information, including check-in time and check-in counts for each restaurant. In order to get the check-in counts for each day and for each hour, query 1 and 2 are used distinctively (Figure 6).

```
create table restaurants_checkin as
select b.name, ch.checktime as checkin_time, ch.checknum as checkin_count
from Category c, Business_Category bc, Business b, Checkin ch
where c.category_id = bc.category_id and bc.business_id = b.business_id
      and b.business_id = ch.business_id
      and c.category = 'Restaurants'
order by checkin_time, checkin_count;

--Query 1
select s, sum(r.checkin_count) as checkin_count
from restaurants_checkin as r, generate_series(24,168,24) as s
where r.checkin_time < s and r.checkin_time >= s - 24
group by s
order by s;

--Query 2
select concat(s, ':00 - ', s + 1, ':00') as Checkin_time,
       sum(r.checkin_count) as Checkin_count
from restaurants_checkin as r, generate_series(0,23,1) as s
where r.checkin_time % 24 = s
group by s
order by s;
```

Figure 6 Queries to get weekly and daily check-in counts for restaurants

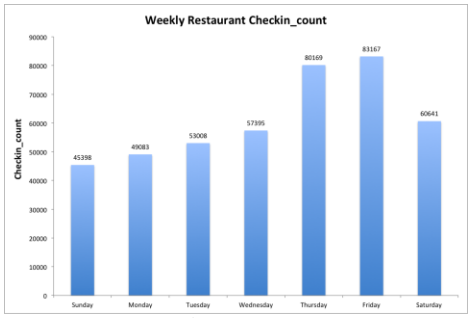

Figure 7 Histogram of restaurants weekly check-in counts

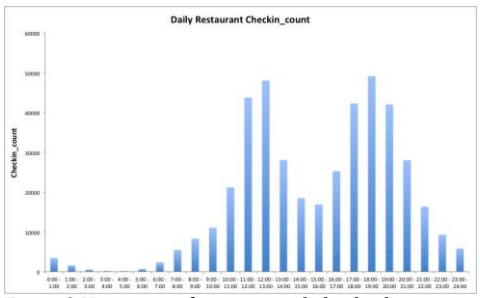

Figure 8 Histogram of restaurants daily check-in counts

We can easily find that restaurants are busiest on Friday and Thursday (Figure 7), during 11:00-13:00 and 17:00-20:00 (Figure 8). Friday and Thursday, rather than Saturday and Sunday, are the busiest days for restaurants, which is out of our expectation. Both lunch hour and dinner hour are the busiest time for restaurants, which amends our presumption that restaurants tend to have more customers at night.

3.1.3 Whether do the "nice" and "mean" users behave similarly?

First, we are curious about the overall generosity of users who write yelp reviews. Specifically, how many stars (average_stars) will the majority of users give to businesses they have experienced? We create a table called user_star_count to store the number of users for each average_stars, and allocate the continuous value average_stars into 5 discrete bins ([0-1], (1-2], (2-3], (3-4], (4-5]) by running query 3 (Figure 9).

```
create table user_star_count as
select average_stars, count(user_id) as user_count
from Reviewer
group by average_stars
order by average_stars desc;

--Query 3
select concat(s-1, '-', s) as avg_star, sum(u.user_count) as user_count
from user_star_count as u, generate_series(1,5,1) as s
where u.average_stars <= s and u.average_stars > s - 1
group by s
order by s;
```

Figure 9 Queries to get the number of users for each bin of user average stars

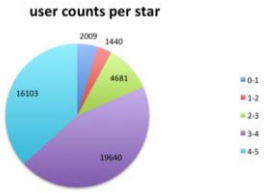

Figure 10 Pie chart of user counts for each user star bin

In Figure 10, we can see that the majority of users have average_stars of 3-5, which means that most yelp users are generous to write positive reviews for businesses, while less than a quarter of users tend to write negative reviews for businesses.

Before comparing the other behaviors of nice and mean users, we assign users with average_stars of 3-5 (excluding 3 and including 5) as nice users, and users with average_stars of 0-3 (including 0 and including 3) as mean users for the sake of simplicity. Particularly, we want to explore whether the nice and mean users will behave in the same way with regard to the number of reviews they write, the kind of businesses they review (popular or not), and the type of reviews they vote (funny, useful or cool).

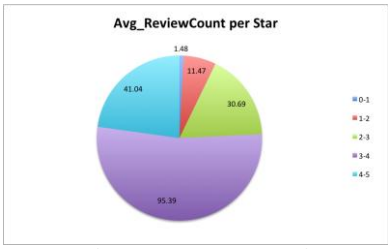

Figure 11 Pie chart of average review counts for each user star bin

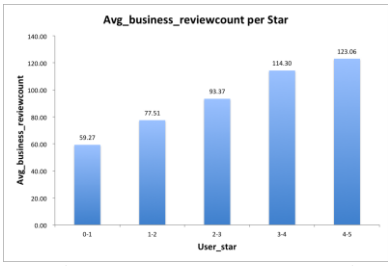

Figure 12 Histogram of average business review counts for each user star bin

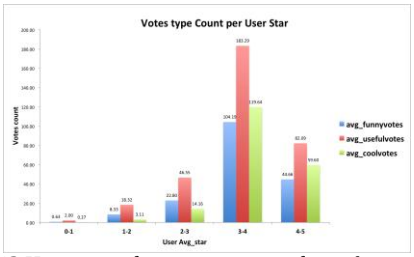

Figure 13 Histogram of votes type counts for each user star bin

In Figure 11, nice users tend to write more reviews than mean users. It seems nice users are also prolific users. In Figure 12, average business review counts are used to measure a business's popularity. Usually nice users write reviews for more popular businesses while mean users write reviews for less popular businesses. In Figure 13, generally users vote more useful reviews than funny and cool reviews, and there is no general difference between nice users and mean users in terms of the vote types.

3.2. Statistical analysis: What makes review useful

Out of interest in what makes review useful and given the possible impact variable varies, we decide to apply statistical regression modeling to figure out what factors have impact on the usefulness of reviews.

3.2.1. Data Collection

For our analysis we used the data from table Reviewer, Review and Business. Based on the past researches, we suggest that factors include stars and review count of the business, review count, average star, useful votes of reviewer, as well as stars, date and useful votes of review could have an impact on the useful vote of review. We take useful vote number as an indicator of the usefulness of reviews.

As the data selected are all quantitative and there is no missing data. Given the large amount of data, we choose only the data from June 1st 2011 to June 1st 2012 and only pick 10 values on each day. To leave out the outliers, we only pick the reviews with votes less than 20.

Here is the query:

*With temp as (Select c.stars, c.review_count, a.review_count, a.average_stars, a.usefulVotes, b.stars, b.date, b.usefulVotes from Reviewer as a, Review as b,Business as c where a.user_id = b.user_id and c.business_id = b.business_id and b.date >= '2011-06-01' and date < '2012-06-01' and b.usefulVotes <= 20) select * from(select row_number() over(partition by temp.date order by temp.date) as r, temp.* from temp) x where x.r <= 10;*

3.2.2. Exploratory Analysis

By observing the plain data of their levels, characteristics and features, we found the data selected are all quantitative. Besides, we examined tables and plots of the data and found there is no missing data. Then we did some exploratory analysis. First we make the scatter plots by having useful votes on different explanatory variables and draw a histogram.

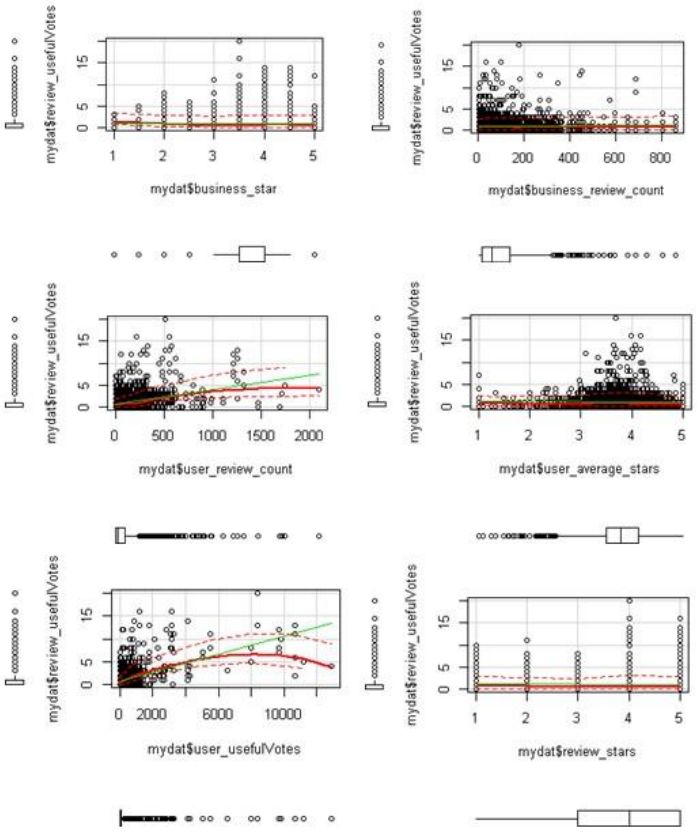

Figure 14. Scatter plot of usefulVotes on varies factor

From the above scatter plots(Figure 14) we can see a clear corresponding relationship between user_review_count and review_usefulVotes as well as that between user_usefulVotes and review_userfulVotes. Though there is a density at the left bottom part in every plot, the linearity looks good.

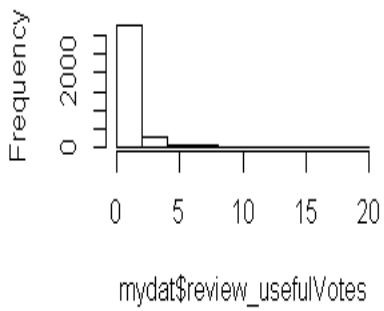

Figure 15. Histogram of usefulVotes on varies factor

From Figure 15 we cansee most of number of useful votes for the reviews fall into the 0-5 interval while there are little falling into other intervals.

3.2.3. Regression modeling:

Explore the possible models with regsubsets function and by comparing the bic, cp and R square values, we choose regression model:

review_usefulVotes~

user_review_count+user_usefulVotes+review_stars.

Then we run a ANOVA analysis based on the regression model.

```
Call:
lm(formula = review_usefulVotes ~ user_review_count + user_usefulVotes +
    review_stars, data = mydat2)

Residuals:
    Min      1Q  Median      3Q     Max
-9.5476 -0.7927 -0.6805  0.2816 14.0428

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        1.097e+00  7.210e-02  15.218  < 2e-16 ***
user_review_count -7.894e-04  2.091e-04  -3.776 0.000162 ***
user_usefulVotes   1.105e-03  4.357e-05  25.358  < 2e-16 ***
review_stars      -7.623e-02  1.773e-02  -4.300 1.76e-05 ***
---
Signif. codes:  0 ?***?0.001 ?**?0.01 ?*?0.05 ?.?0.1 ? ?1

Residual standard error: 1.359 on 3656 degrees of freedom
Multiple R-squared: 0.2584, Adjusted R-squared: 0.2578
F-statistic: 424.6 on 3 and 3656 DF,  p-value: < 2.2e-16
```

Figure 16 ANOVA analysis

Based on the statistic result above (Figure 16), we can see the three factors: count of users' reviews, count of users' useful votes and review stars are having a significant impact on the usefulness of the review. The three factors all have p-value less than 0.0001, which are far less than the p-level value 0.05 and indicate a great statistical significance. The interesting fact is the relationship between usefulness of reviews with users review count and review stars are negative. It seems less productive reviewers will receive more useful votes on their reviews and the negative reviews are winning more votes, which may because that negative reviews tell true and useful stories. There is no doubt that the useful votes of users and useful votes of the reviews are positively correlated. It is straightforward that the users, who used to wrote many useful reviews and receive more votes, tend to produce more useful reviews.

3.2.4. Model validation

Then we checked the regression model with normality with QQ plot and check non-constant variance with plot of residuals over fitted values.

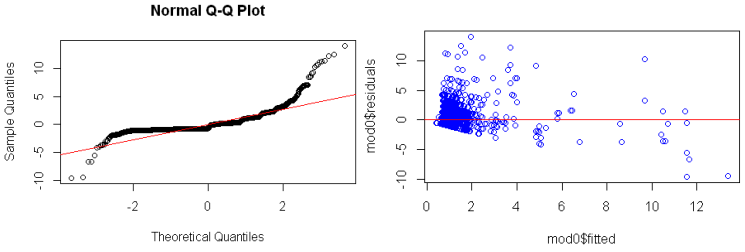

Figure 17 Normality check and variance check

From Figure 17 we can see the normality is good despite the skew on the right upper side. The residual plot shows a dense area at the very first and the points are nicely located around the red line, which indicates the predict value is close to the actual value.

To check the nonlinearity, we use added variable plot and component pluts residual plot.

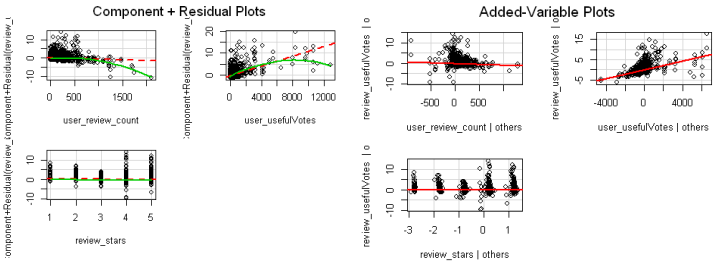

Figure 18 non-linearity check plots

From Figure 18 above, we can see there is no clear awry shape, that is to say, the non-linearity is good in the model.

3.2.5. Time series feature

We are thinking whether the useful votes may have a pattern on the time line. Then we produce a time series plot with data from June 2011 to June 2012.

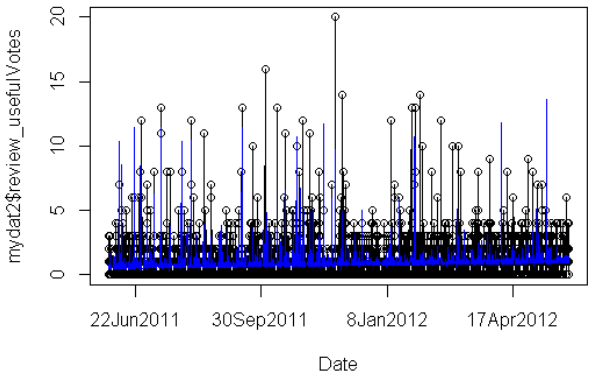

Figure 19. Time series plot of useful reviews

From the Figure 19 above we can see there is no clear pattern on time as most reviews fall into the 0-5 vote count interval and have no clear difference on time. As to the reviews with votes more than 5, we can see there are two low points around August 2011 and December 2012. It could be a result of that people are so busy taking vocation or shopping that they do not have time and energy to write useful votes, or it just happens by chance.

3.3. Suggestions on how to design database schema serving data analysis

It is very useful to predict what analysis we are going to make when designing the database schema. The data, which is aggregated or in a hierarchy organization, say, category list and check-in data, could be flatted to be put into the database. Then the original data can be recovered with the help of aggregate queries. One idea is to break down data objects into data records without harming the integrity while bringing convenience to the future data analysis. Rationalize the data model into a relational database is good, but don't overdone it, leading to complex queries.

Besides, special attention needs to be taken during the type conversion between the json data type and postgres data type. The precision of the numbers should be kept in the database while the space size should not be too expensive as an expense. Database performance should also be pre-evaluated when transforming the data into basic number, fixed character, variable character and date or complex data type like array, json object.

4. Conclusion and Future Work

In our project, we find some interesting facts about users, business and review through integrating some query results and doing statistical analysis.

For business, the relationship between a business's success and its location depends on business category. In our example, both the most successful restaurants and nightlife businesses are well clustered in the central area, while the others are randomly distributed in the region. However, this pattern disappears if business star is chosen to measure business success. The pattern is not consistent across different categories. For example, we didn't find such pattern between business location and success for shopping malls.

For business and check-in data, we only chose restaurant data for analysis. It seems that most customers check in for restaurants at noon and dinner every Thursday and Friday.

For users, the majority of them are generous to give high review stars. After comparing the behaviors of nice users (who give review stars above 3) and mean users (who give review stars below and include 3), we find nice users tend to write more reviews for more popular businesses than men users.

Based on the statistical modeling and analysis, we find the review count and useful votes of user who write the review, as well as the review stars have an impact on the useful votes of the review. Among them user review count and review stars have negative relationship with the useful votes of review, the user useful votes have positive negative relationship with the useful votes of review.

We also have suggestions on how to design database schema: 1. Predict what data analysis methods and procedures will be used when doing the database schema design to make it best suit for the future needs; 2. Keep integrity of the original data when loading them into the database, especially when doing type conversion between different data types.

The Yelp dataset used in this project is only limited to the review data at Phoenix and only limited properties of Business, Review, User and Check-in. We are expecting to have more data to validate and improve our data model. More information on the users' recent activities, geographical information, and the interaction between the users on the reviews could be retrieved and be exploited to improve the data analysis. Besides, we only did the research on how the quantitative properties of the data affect the reviews' usefulness. Actually the semantic meaning of reviews contributes the most to the usefulness of the reviews. A better model includes proper semantic analysis and machine learning could be applied to improve the accuracy of predicting results.

5. Reference:

1. JSON Functions and Operators http://www.postgresql.org/docs/devel/static/functions-json.html
2. Momjian, Bruce. PostgreSQL: introduction and concepts. Vol. 192. Addison-Wesley, 2001.
3. Ramakrishnan, Raghu, and Johannes Gehrke. Database management systems. Osborne/McGraw-Hill, 2000.
4. Json usage. https://code.google.com/p/json-simple/downloads/list.
5. Cardinal, Rudolf N., and Michael RF Aitken. ANOVA for the behavioural sciences researcher. Lawrence Erlbaum Associates Publishers, 2006.
6. Kutner, Michael H., Chris Nachtsheim, and John Neter. "Applied linear regression models." (2004).
7. Kester, Walter Allan, ed. Data conversion handbook. Newnes, 2005.