Yelp

Analysis of the Impact of a Super-User

Group 1
Shreyas Gowda
Samuel Hung
Shreya Mary Kuruvilla
Rahul Zende

Table of Contents

Introduction

Overview of Yelp

This analysis is based on a Yelp dataset that was retrieved from Kaggle. Yelp is commonly known as a local-search service that helps users find the best local businesses, such as restaurants, dentists, barbers, and more. Businesses can use the website to post photos and information about the business. On the other hand, users can use the application to find these businesses and leave photos and reviews. Yelp has exploded in popularity since its inception in 2004, having a reported average of 69 million unique monthly visitors who use the “mobile version” of the website (Yelp, 2019). Additionally, Yelp’s primary means of generating revenue is through selling ad space to local businesses (Yelp, 2019).

Scope of Study

Dataset

The dataset provided business, review, and user data for 11 major metropolitan areas. Included in the main dataset were over 5.2 million user reviews and 174,000 businesses (Kaggle, 2018). Furthermore, this dataset includes review data spanning from 2010 up to the end of 2018. This data was separated into several different CSV files: business.csv, business_attributes.csv, business_hours.csv, review.csv, tip.csv, and the user.csv. The business check-in data was initially available in a json file format, which was transformed into a data frame using the *jsonlite* library.

The business.csv file provided a business’ information such as the business’ name, address, total rating, and review count. The unique identifier used for this file was a business ID.

The business_attributes.csv file provided various attributes of each business such as amenities provided and the ambience of the business.

The business_hours.csv file provided the working hours of each business.

The checkin.csv file provided specific check-in data for a business. This was detailed to the date and time that a user ‘checked in’ to a business. However, this file did not provide the actual user that checked in to a business. Furthermore, check ins to a business is reported by a user, and is not data that is collected and reported by the business itself.

The review.csv file provided each individual review that a user had left for a business. Included in this file was the user’s ID, the business’ ID, the stars of the review, the date of the review, and the actual text left by the user. There was a unique review ID used for each individual review.

The tips.csv file provided tips that users left for individual businesses. Tips could include what the business was known for and what other information a visitor may find useful for a business.

The user.csv file provided detailed information for each user. This included the overall number of reviews the user had left for businesses, the average stars the user had left in reviews, and the years that the user had achieved the ‘elite’ status. The unique identifier used was a user ID.

Focusing the analysis

Such a large dataset required significant computing power for analysis in R and Python. To better focus on data analysis, this research focused on the city of Las Vegas, as this city provided the greatest number of businesses and user reviews. Furthermore, the analysis focused primarily on restaurants as restaurants were the most prominent business type within the Las Vegas metropolitan area.

Problem Statement

Restaurants may want to consider how to improve the popularity of their own business and use Yelp’s platform to do so. Previous research had found that a one-star rating increase could lead to a 5-9% increase in restaurant revenue (Luca, 2016). Similarly, our research set out to determine if there are any relationships between **super-user** reviews and the popularity of a restaurant. Super-users are defined as users who have left over **300 reviews** at restaurants. This is a mere 0.02% of users in the dataset. While the available datasets didn’t provide any information on restaurant performance, our team measured popularity by the number of reviews left at the restaurant.

Final problem statement: Does a super user review for a restaurant in Las Vegas lead to more reviews on average at that restaurant?

Methods

Preparation of Data

The seven CSVs (5.14 GB in size) obtained from Kaggle were explored for commonalities - business_id and user_id was flagged as the two identifiers for rows in each dataset. To simplify further analysis, related datasets were merged and subsetted (for Las Vegas’ restaurants) together using either of these identifiers into four CSVs (), each containing the common identifier - business_id. This greatly simplified further data exploration.

Subset (progressing downwards)	Number of rows (reduced as we subset)
Yelp dataset	174567 listings
City of Las Vegas	26775 listings
Restaurants in Las Vegas	5902 listings

Exploratory Data Analysis

To decide upon a metric to qualify a certain Yelp user/reviewer as a ‘super-user’, the dataset was aggregated for calculating the count of restaurants reviewed, grouping by the user_id.

```
aggregate(business_id ~ user_id, data = reviews.users, FUN = NROW) # calculate the number of reviews per user/reviewer
```

When put into buckets, it was noticed that the following number of users had reviewed the said count of restaurants -

Minimum restaurants reviewed	No. of users satisfying the criteria
Count > 100	257
Count > 150	115
Count > 200	58
Count > 250	34
Count > 300	***15***

Thus, the last bucket was chosen to ensure we analyze a set of users who have a sufficiently distinguished profile on Yelp platform, in terms of number of restaurants they have visited and reviewed.

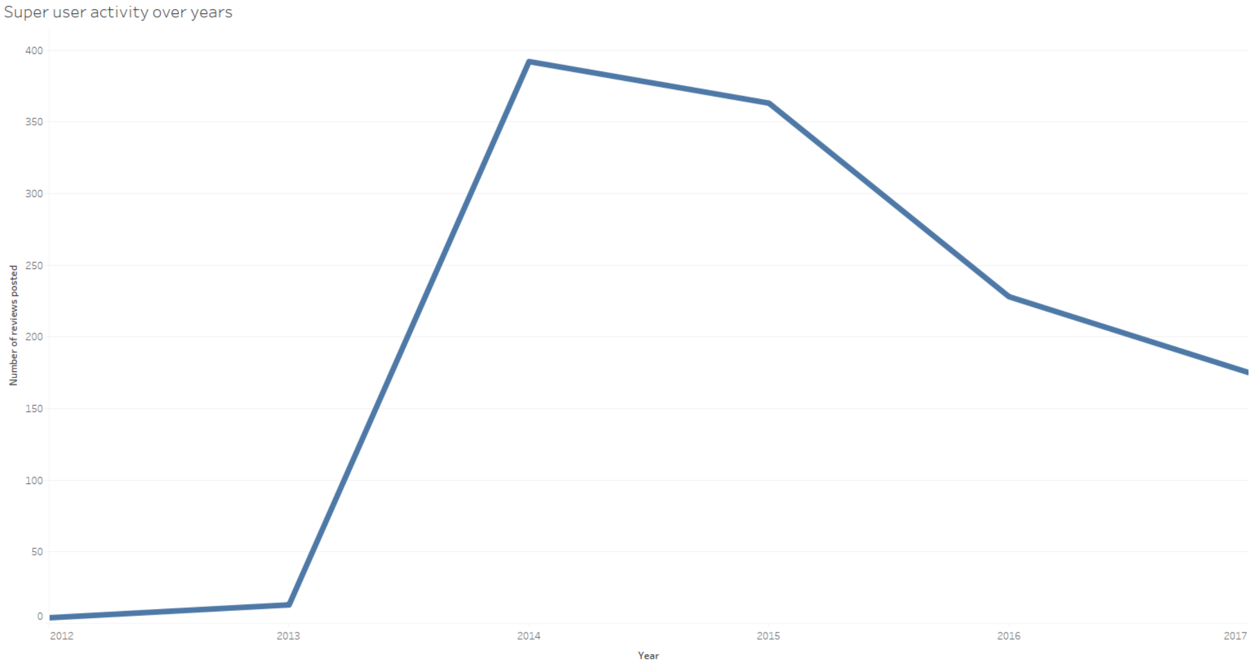
Next came the issue of picking a single user amongst this group of 15 to conduct an impact analysis. For this step, we decided to find out how many restaurants each of these 15 individuals actually reviewed from 2012 to 2017.

User_id	Number of restaurants reviewed
y3FcL4bLy0eLlkb0SDPnBQ	302
48vRThjhuhiSQINQ2KV8Sw	304
L8P5OWO1Jh4B2HLa1Fnbng	316
JaqcCU3nxReTW2cBLHounA	330

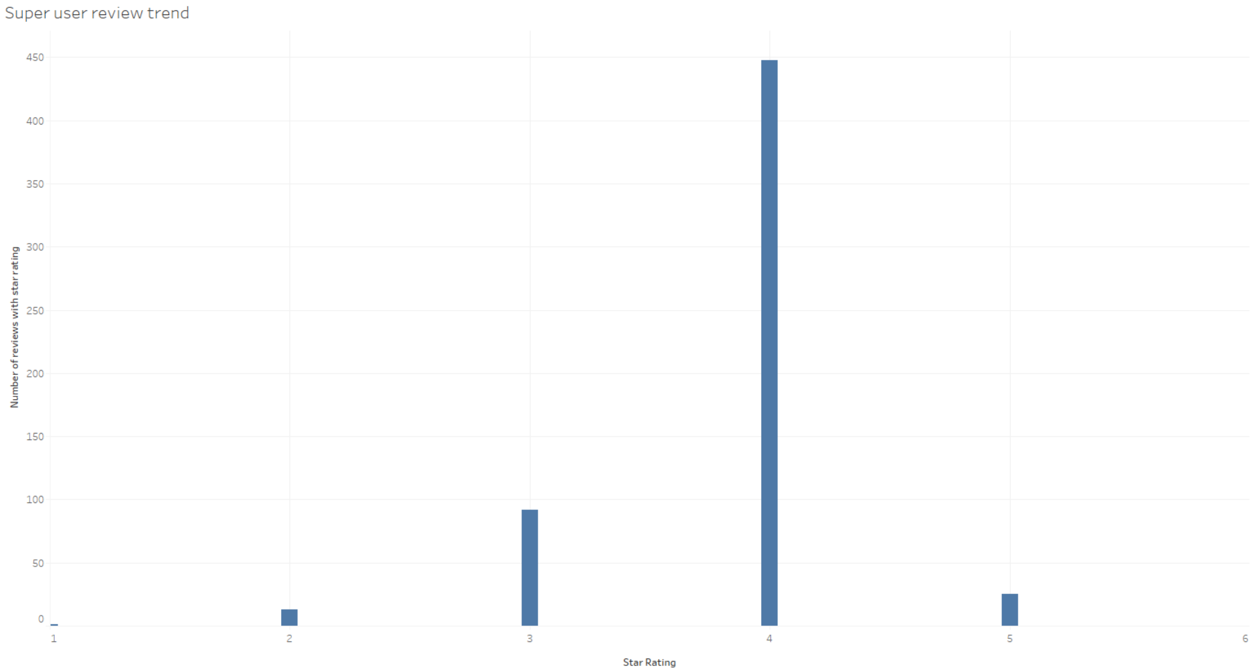
tH0uKD-vNwMoEc3Xk3Cbdg	401
qewG3X2O4X6JKskxyyqFwQ	403
3nDUQBjKyVor5wV0reJChg	410
N3oNEwh0qgPqPP3Em6wJXw	416
8DEyKVyplnOcSKx39vatbg	422
U4INQZOPSUaj8hMjLlZ3KA	444
n86B7IkbU20AkxlFX_5aew	486
C2C0GPKvzWWnP57Os9eQ0w	521
UYcmGbelzRa0Q6JqzLoguw	618
PKEzKWv_FktMm2mGPjwd0Q	781
bLbSNkLggFnqwNNzzq-Ijw (*Stefany*)	***1175***

As seen here, the top scorer is user Stefany - who has reviewed 1175 restaurants, which is more than the next highest score by a large margin. It made sense to pick her for our analysis, since we would get more data to conduct further analysis upon (more data = better analysis).

The following visualizations demonstrate that Stefany has been active since 2012 (gaining in activity steeply in 2013) through 2017. Also, she has been a fair reviewer, rating most restaurants with 4 stars (most ratings) and otherwise.



The trend of count of User Id for Date Year. The data is filtered on User Id, which keeps bLbSNkLggFnqwNNzzq-ljw.



The plot of count of Stars for Stars. The data is filtered on User Id, which keeps bLbSNkLggFnqwNNzzq-Ijw. The view is filtered on Stars, which keeps non-Null values only.

Moving on, we subset the data - choosing only the restaurants Stefany visited and partitioning them into 2 buckets - the reviews for the 1175 restaurants before and after she visited them. These two buckets were aggregated based on the date column - in order to find out the number of reviews posted per day. The calculation involved computing the total number of reviews posted before and after Stefany visited a particular restaurant (N), and also computing the time elapsed for two-time frames - between 2012 and the day Stefany visited the restaurant (days elapsed), and between the day she visited the restaurant till 2017 (days elapsed). Dividing N by the number of days elapsed is an approximation of the reviews posted per day for a specific restaurant, within each of those time frames. This data was further used to perform the hypothesis testing.

In a similar manner as above, the textual data of reviews posted by multiple users (before and after Stefany visited it) was partitioned to perform a sentiment analysis of the review content - in order to assess any apparent differences.

Hypothesis Testing

Hypothesis testing is the use of statistics to determine the probability that a given hypothesis is true. The usual process of hypothesis testing consists of four steps.

1. Formulate the null hypothesis H_0 (commonly, that the observations are the result of pure chance) and the alternative hypothesis H_A (commonly, that the observations show a real effect combined with a component of chance variation).

2. Identify a test statistic that can be used to assess the truth of the null hypothesis.

3. Compute the P-value, which is the probability that a test statistic at least as significant as the one observed would be obtained assuming that the null hypothesis were true. The smaller the P-value, the stronger the evidence against the null hypothesis.
4. Compare the p-value to an acceptable significance value α (sometimes called an alpha value). If p<= α, that the observed effect is statistically significant, the null hypothesis is ruled out, and the alternative hypothesis is valid.

Hypothesis test

We will conduct a hypothesis test to determine if there is an increase in the average number of reviews after a super user has reviewed for that restaurant. We set up a hypothesis test as follows:
H_0: There is no difference in the average number of reviews after a super user has reviewed a restaurant $\mu_0= \mu_A$
H_A: The average number of reviews increases after a super user has reviewed a restaurant $\mu_0< \mu_A$

Summary of Data

We are trying to estimate the difference in average number of reviews at a restaurant after a super user has reviewed. We are looking at the average number of reviews before and after a super user has visited and reviewed the restaurant. Hence these two observations are paired.
Two sets of observations are paired if each observation in one set has a special correspondence with exactly one observation in the other data set.

Hence, we use a Paired Samples T-Test.

The Paired Samples T-test is used to compare the means between two related groups of samples.
Conditions for Paired Samples T-Test:

1. Paired Observations: The data has been collected from measuring the average number of reviews before and after a super user has reviewed for the same restaurant.
2. The sample size is large. We have 1139 restaurants in our dataset. The T distribution will approximate to a normal model.
3. Paired sample differences should follow a normal model. Since we have a large sample, we can relax this condition.

Paired T-Test to determine P-value:

Here we are doing a one-sided hypothesis T-Test at a 95% confidence interval if the average number of reviews.after is greater than average number of reviews.before.

```
t.test(test$reviews.after,test$reviews.before, alternative = "greater", mu = 0, paired = TRUE, var.equal = FALSE,conf.level = 0.95)
```

Result and Analysis:

We get a P-value of 1. There is no evidence to show that there has been an increase in average number of reviews after a super user has reviewed at a restaurant.
P-value of 1 means that there is no increase in the average number of reviews for this sample after a super user has reviewed for that restaurant. If we take a random sample from this dataset and compute the difference in means of average number of reviews before and after a super user has reviewed, we will always get a negative value.
Hence, we reject the null hypothesis.

Further Analysis and Findings

We decided to look at the other side of the test to see if the average number of reviews has reduced after a super user has reviewed a restaurant.

Paired T-Test to determine P-value:

Here we are doing a one-sided hypothesis T-Test at a 95% confidence interval if the average number of reviews.after is lesser than average number of reviews.before.
t.test(test$reviews.after,test$reviews.before, alternative = "lesser", mu = 0, paired = TRUE, var.equal = FALSE,conf.level = 0.95)

Result and Analysis:

We get a test statistic of -6.0572 and the corresponding P-value is very less (9.4e-10). This provides evidence that the average number of reviews is reducing after a super user has reviewed.
In this case, we have overwhelming evidence to reject the null hypothesis.

Statistics

Point Estimate of the differences: -0.09412849
Standard Deviation: 0.524462
Standard Error: 0.015540
95% confidence interval: [-0.1245, -0.0636]
We are 95% confident that the true population difference (for the city of Las Vegas) in the average number of reviews before and after a super user has reviewed for a particular restaurant is between -0.1245 and -0.0636.

Text Mining Analysis

In our study, the objective of text mining analysis was to obtain a general sense of whether the quality of reviews changed after a super user check-in. We achieved this using the **bag-of-words model**. This is a simple, yet powerful approach of creating a vocabulary from a collection of words and mapping them to their frequencies of appearance. In our case, the collection of words is derived from user reviews before and after super user check-ins. The detailed steps of text mining analysis are mentioned below:

1. Creation of a Corpus

   Corpus is a collection of documents, and in this case, documents are user reviews. Two corpuses are to be created, from reviews before and after super user check-ins. This is achieved using the *tm* package in R. The original number of reviews are almost 100,000 which becomes difficult for processing. Since we only aim to obtain a general idea of review trends, a random sample of 10,000 reviews were subsetted using *quanteda* package.

2. Pre-processing of text

   This is the most tedious but important step in the process. The purpose of the data cleaning process is multi-fold. Firstly, words which have no potential of providing insights are to be removed. This involves stop words such as 'the', 'and' etc., numbers and punctuations. Secondly, words which have the same meaning or context have to be treated equally. 'Stemming' is performed to reduce words to their root form. E.g., 'great', 'greatest', 'greater' are all reduced to the word 'great' because they all indicate the same emotion. The pre-processing of data is achieved using the *tm* and *SnowballC* packages. Below are the steps for data cleaning of user reviews:

   - Converted to lowercase
   - Removed numbers, punctuations
   - Removed stopwords
   - Stripped whitespaces
   - Stemming of words

The cleaned data is now ready for the next part of the analysis.

3. Word frequency mapping and visualization

   A 'Term Document Matrix' is created by mapping each word to the number of appearances in user reviews. We are able to run this step without any errors because of the reduced size of reviews to 10,000 words in the first step.

   A word-cloud visualization is created using *wordcloud* and *RColorBrewer* packages.

Results

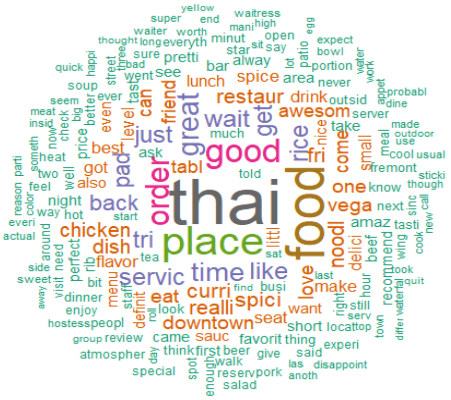
Hypothesis Testing

From our hypothesis test, we can conclude that there is no increase in the average number of reviews after a super user has reviewed that restaurant.
Alternatively, we have seen that there has been a decrease in the average number of reviews after a super user has reviewed that restaurant.

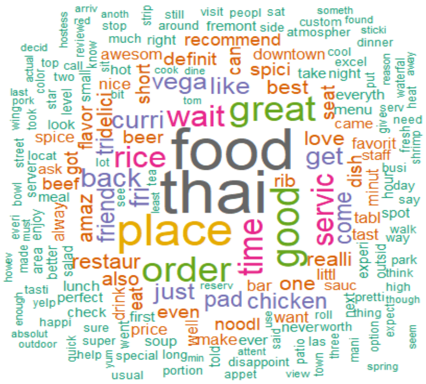
Hence, there is a correlation between a super user and the average number of reviews. Since this is an observational study, we cannot establish a causal link. We can establish a causal link by conducting an experiment.

Word Cloud Visualization

As described in the ‘Methodologies’ section, the bag-of-words model was used to perform qualitative analysis of user reviews. The results of this analysis were depicted using Word Cloud visualizations. Below are the results:



Before super-user review



After super-user review

Insights from the word cloud visualizations:

- There appears to be no visible difference between the general sentiment of user reviews before and after a super user check-in.
- Thai food is a popular choice of cuisine in Las Vegas with words such as ‘thai’ and ‘phad’ and possibly, ‘noodle’.
- Different characteristics of a restaurant that affect user reviews include ‘service’, ‘waiting time’ and ‘flavour’ and ‘taste’ of dishes.
- Words such as ‘place’ and ‘time’ offer little insight but appear quite often.
- Certain words such as ‘good’ can also have a negative connotation if combined with a word such as ‘not’, and therefore be misleading.
- The sentiment of the user from a word such as ‘spicy’ depends purely on the context, for e.g., if the user enjoys spicy food.

Overall, the text mining analysis does not reveal a distinction between quality of reviews before and after super-user check-in. However, the word clouds do let us obtain a few insights that could possibly be valuable.

Discussion

Limitations

There were several limitations to this study, primarily various potential confounding factors that have led to a lack of improvement of reviews for restaurants. For example, it may be that restaurant quality may have decreased over time, resulting in less visitors and fewer reviews. Additionally, super-users who leave low ratings at a restaurant may have experienced an off-day for that restaurant. The head chef may have been out that day or service may have been poorer. Additionally, our analysis failed to account for the day that a review was left at a restaurant. Restaurants tend to receive more visitors on the weekends. If a super-user left a review on the weekend, it wouldn't be too surprising to find that there were fewer reviews left or fewer check-ins during the following days.

Furthermore, there were limitations with the text-mining analysis. There were few words that offered overall insight into the performance of a restaurant. Some reviews and words could be seen as evoking a positive sentiment, but these words could also indicate a negative sentiment depending on the context of the review. For example, 'good' is seen as positive and 'not good' is negative. In the process of frequency mapping of words, the context is often lost. For example, the emotion behind the word 'spicy' depends on whether the user enjoys spicy food or not.

Impact of Analysis

Our findings did not result in any statistically significant relationship between super-user reviews and restaurant reviews. There may be other metrics that can help determine the performance of restaurants. It would be interesting to research food trends that crop up over time and whether or not that plays a role in how well a restaurant performs. For example, the text analysis found that 'Thai' was one of the most frequently used words in reviews. Is Thai food a more popular food category compared to others? Is this popularity due to food trends where certain foods are only popular for a certain amount of time?

Furthermore, restaurants could conduct a more detailed text analysis to gather valuable insights related to the features of the restaurant that causes a user to provide high ratings. For example, restaurants could see if involve fewer waiting times, higher availability of parking, the quality of service, the taste of foods, and more in a review could improve review count and future check-ins at the restaurant.

Restaurants could also conduct a more detailed text mining analysis. They could conduct a sentiment analysis by mapping terms with a 'sentiment' value. For example, a 0 could be given to words with a neutral sentiment. This would help to correlate reviews with ratings and would help to look for a more quantitative analysis. The text mining analysis could also be validated by calculating an accuracy percentage as the percentage of reviews with positive sentiment that had ratings greater than 3. The context of the word can be retained by using more advanced text mining packages such as *quanteda*. Restaurants could examine whether a review with a certain sentiment score was related to future reviews having a negative or positive sentiment. We recommend restaurants to consider conducting these additional analyses to research ways to improve business performance.

References

1. Barr, Christopher, et al. *OpenIntro Statistics*. OpenIntro, Inc., 2015.
2. Hung, Samuel. “IMT 573_Yelp_Final.” *Kaggle*, 2019, www.kaggle.com/shung93/imt-573-yelp-final?scriptVersionId=11645124.*
3. Luca, Michael. “Reviews, Reputation, and Revenue: The Case of Yelp.com.” *Reviews, Reputation, and Revenue: The Case of Yelp.com - Working Paper - Harvard Business School*, 1 Sept. 2011, www.hbs.edu/faculty/Pages/item.aspx?num=41233.
4. Yelp, Inc. “Yelp Dataset.” *Kaggle*, 5 Feb. 2019, www.kaggle.com/yelp-dataset/yelp-dataset.

**Kaggle environment was used for some analysis due to limited computer resources.*

Appendix

1. Code for subsetting and EDA

```
setwd("Z:/yelp-version 6")
business <- read.csv("yelp_business.csv") # 174567 businesses in all
LV.businesses <- subset(business, city == "Las Vegas") # 26775 businesses
LV.restaurants <- subset(LV.businesses, grepl("*Restaurant*", LV.businesses$categories)) # 5902 restaurants
write.csv(LV.businesses, file = "LV_businesses.csv", row.names = FALSE)
write.csv(LV.restaurants, file = "LV_restaurants.csv", row.names = FALSE)

# subset the business_attribs_hrs dataset for LV restaurants
LV.business.attribs.hrs <- read.csv("LV_business_attribs_hrs.csv")
LV.restaurant.attribs.hrs <- merge(LV.restaurants, LV.business.attribs.hrs, by = "business_id")
write.csv(LV.restaurant.attribs.hrs, file = "LV_restaurant_attribs_hrs.csv", row.names = FALSE)

# subset the business_checkins dataset for LV restaurants
LV.business.checkins <- read.csv("LV_business_checkins.csv")
LV.restaurant.checkins <- merge(LV.restaurants, LV.business.checkins, by = "business_id")
write.csv(LV.restaurant.checkins, file = "LV_restaurant_checkins.csv", row.names = FALSE)

# subset the business_tips dataset for LV restaurants
LV.business.tips <- read.csv("LV_business_tips.csv")
LV.restaurant.tips <- merge(LV.restaurants, LV.business.tips, by = "business_id")
write.csv(LV.restaurant.tips, file = "LV_restaurant_tips.csv", row.names = FALSE)
```

```
# subset the business_reviews_users dataset for LV restaurants
LV.business.reviews.users <- read.csv("LV_business_reviews_users.csv")
LV.restaurants.reviews.users <- merge(LV.restaurants, LV.business.reviews.users, by =
"business_id")
write.csv(LV.restaurants.reviews.users, file = "LV_restaurants_reviews_users.csv",
row.names = FALSE)

# load all data for doing analysis
attribs.hrs <- read.csv("LV_restaurant_attribs_hrs.csv")
checkins <- read.csv("LV_restaurant_checkins.csv")
tips <- read.csv("LV_restaurant_tips.csv")
restaurants <- read.csv("LV_restaurants.csv")
reviews.users <- read.csv("LV_restaurants_reviews_users.csv")

# decide on a criteria to identify super-users in Las Vegas
colnames(reviews.users)
NROW(unique(reviews.users$user_id)) # 337874 unique users/reviewers
no.of.reviews.per.users <- aggregate(business_id ~ user_id, data = reviews.users, FUN =
NROW) # calculate the number of reviews per user/reviewer
NROW(subset(no.of.reviews.per.users, business_id > 100)) # 257 users
NROW(subset(no.of.reviews.per.users, business_id > 150)) # 115 users
NROW(subset(no.of.reviews.per.users, business_id > 200)) # 58 users
NROW(subset(no.of.reviews.per.users, business_id > 250)) # 34 users
NROW(subset(no.of.reviews.per.users, business_id > 300)) # 15 users --> seems like a
decent choice to go forward with

# we find out the super-users based on identified criteria
library(dplyr)
subsetted.restaurant.reviews.and.dates <- select(reviews.users, business_id, user_id, date) #
all restaurants here, with business & user & date
subsetted.restaurant.reviews.and.dates$date <-
(as.character(subsetted.restaurant.reviews.and.dates$date) %>% as.Date)
## temp1 <- aggregate(business_id ~ user_id, data = temp, FUN = NROW)  --- this is
rendered useless now, we have an extract in the csv
list.of.super.users <- subset(temp1, business_id > 300)[1]
number.of.reviews.posted <- subset(temp1, business_id > 300)[2]
LV.super.users <- cbind(list.of.super.users, number.of.reviews.posted)
LV.super.users <- LV.super.users %>% rename(num_of_reviews = business_id)
write.csv(LV.super.users, file = "LV_super_users_vs_number_of_reviews.csv", row.names =
FALSE)

# moving on, we subset for the restaurants that our super-users have reviewed
super.users.and.reviewed.restaurants <- inner_join(temp, list.of.super.users, by = "user_id")
write.csv(super.users.and.reviewed.restaurants, file =
"LV_super_users_and_restaurants_reviewed.csv", row.names = FALSE)
temp3 <- select(super.users.and.reviewed.restaurants, business_id, user_id)
temp4 <- aggregate(user_id ~ business_id, data = temp3, FUN = NROW)
temp4[order(temp4$user_id, decreasing = TRUE),][1,1]
subset(super.users.and.reviewed.restaurants, business_id == "qqs7LP4TXAoOrSlaKRfz3A")
```

```
# getting the dates this restaurant was reviewed by super-users

# over here, we find out some info about reviews posted for the restaurant mentioned above
tmp1 <- subset(subsetted.restaurant.reviews.and.dates, business_id ==
"qqs7LP4TXAoOrSlaKRfz3A") # 1093 reviews in all
summary(tmp1$date)
# for this restaurant --
  # min date is 2010-04-08
  # mean is     2014-07-22
  # median is   2014-07-07
  # max is      2017-12-07

# now we try to find the impact created by our super-user (bLbSNkLggFnqwNNzzq-Ijw)
# this was when 3nDUQBjKyVor5wV0reJChg reviewed : 2014-07-08
NROW(subset(subsetted.restaurant.reviews.and.dates, business_id ==
"qqs7LP4TXAoOrSlaKRfz3A" & date > "2014-07-08")) # 543
NROW(subset(subsetted.restaurant.reviews.and.dates, business_id ==
"qqs7LP4TXAoOrSlaKRfz3A" & date < "2014-07-08")) # 549



# now we create a loop to find out the impact of a specific super-user on our set of restaurants
sample.of.restaurants <- unique(super.users.and.reviewed.restaurants[1])
#set.seed(10)
#sample.of.restaurants <-
super.users.and.reviewed.restaurants[sample(nrow(super.users.and.reviewed.restaurants),
50), ][1] # randomly select 50 restaurants
super.user <- "bLbSNkLggFnqwNNzzq-Ijw" # choose the super-user who has posted most
reviews, for better hit-rate
unique(select(subset(reviews.users, user_id == "bLbSNkLggFnqwNNzzq-Ijw"), name.y)) #
Stefany

collected.stats <- data.frame(business_id = character(), reviews.before = numeric(),
reviews.after = numeric())
collected.stats$business_id <- as.character(collected.stats$business_id)

for (i in 1:nrow(sample.of.restaurants)) {
  restaurant.business.id <- as.character(sample.of.restaurants[i,])
  tmp2 <- subset(subsetted.restaurant.reviews.and.dates, business_id ==
restaurant.business.id)
  date.super.user.visited <- as.integer(subset(tmp2, user_id == "bLbSNkLggFnqwNNzzq-
Ijw")[3])
  if ((is.na(date.super.user.visited))) {}
  else {
    collected.stats[i,1] <- as.character(restaurant.business.id)
    temmp1 <- subset(subsetted.restaurant.reviews.and.dates, date < date.super.user.visited &
business_id == restaurant.business.id)
    number.of.reviews.posted <- nrow(temmp1);
    if (number.of.reviews.posted != 0) {days.elapsed <- as.numeric((aggregate(date ~
business_id, data = temmp1, FUN = max)[2] - aggregate(date ~ business_id, data = temmp1,
FUN = min)[2])[1])}
    collected.stats[i,2] <- number.of.reviews.posted/days.elapsed
    # collected.stats[i,2] <- nrow(subset(subsetted.restaurant.reviews.and.dates, date <
date.super.user.visited & business_id == restaurant.business.id))
    temmp2 <- subset(subsetted.restaurant.reviews.and.dates, date > date.super.user.visited &
business_id == restaurant.business.id)
    number.of.reviews.posted <- nrow(temmp2);
    if (number.of.reviews.posted != 0) {days.elapsed <- as.numeric((aggregate(date ~
business_id, data = temmp2, FUN = max)[2] - aggregate(date ~ business_id, data = temmp2,
FUN = min)[2])[1])}
    collected.stats[i,3] <- number.of.reviews.posted/days.elapsed
    # collected.stats[i,3] <- nrow(subset(subsetted.restaurant.reviews.and.dates, date >
date.super.user.visited & business_id == restaurant.business.id))
  }
}

# nrow(subset(subsetted.restaurant.reviews.and.dates, date > "2014-09-22" & business_id ==
"Wyjk6RBeOPQr7td5Tqwksw")) # user_id == "bLbSNkLggFnqwNNzzq-Ijw" & business_id ==
"Tv19MQrLgdsvSG0myMYZBw"
# nrow(subset(subsetted.restaurant.reviews.and.dates, date < "2014-09-22" & business_id ==
"Wyjk6RBeOPQr7td5Tqwksw")) # user_id == "bLbSNkLggFnqwNNzzq-Ijw" & business_id ==
"Tv19MQrLgdsvSG0myMYZBw"

# for hypothesis testing and t-test
collected.stats.non.na <- subset(collected.stats, business_id != "NA")
write.csv(collected.stats.non.na, file = "stats_2.csv", row.names = FALSE)

# for word cloud
selected.columns.for.word.cloud <- select(reviews.users, business_id, user_id, date, text)
temp.merged <- merge(selected.columns.for.word.cloud, collected.stats.non.na,
by="business_id")
merged.and.selected <- select(temp.merged, business_id, user_id, date, text)
merged.and.selected$date <- (as.character(merged.and.selected$date) %>% as.Date)

reviews.posted.before <- data.frame(text = character())
reviews.posted.before$text <- as.character(reviews.posted.before$text)

reviews.posted.after <- data.frame(text = character())
reviews.posted.after$text <- as.character(reviews.posted.after$text)

for (i in 1:nrow(merged.and.selected)) {
  res.bus.id <- as.character(merged.and.selected[i,1])
  tmpx <- subset(merged.and.selected, business_id == res.bus.id)
  date.sup.usr.vistd <- subset(tmpx, user_id == "bLbSNkLggFnqwNNzzq-Ijw")[3]
  tempxx1 <- subset(tmpx, date < date.sup.usr.vistd & business_id == res.bus.id)
  for (j in 1:nrow(tempxx1)) {
  reviews.posted.before <- add_row(reviews.posted.before, text = tempxx1[j,4])
  }
  tempxx2 <- subset(tmpx, date > date.sup.usr.vistd & business_id == res.bus.id)
  for (k in 1:nrow(tempxx2)) {
```

```
  reviews.posted.after <- add_row(reviews.posted.after, text = tempxx2[k,4])
  }
}

write.csv(reviews.posted.before, file = "before_wc.csv", row.names = FALSE)
write.csv(reviews.posted.after, file = "after_wc.csv", row.names = FALSE)
```