Are Chicago-area Chinese restaurants more or less likely to have Wi-Fi than Chicago-area Indian restaurants?

By Olubukola Ibrahim,
ISU

April 20, 2015

1 Introduction

There are many kinds of businesses in Chicago. Restaurants are lucrative businesses here in the United States; providing food and social point for customers. Like any business in the US, restaurants aim to satisfy its customers. Some restaurants continue to create and improve on existing techniques used to attract more customers. In Chicago, there are different kinds of restaurants. Most customers, who visit restaurants for a meal may require that a restaurant have Wi-Fi services. As a result, many restaurants have implemented Wi-Fi services as part of its business technqiues to satisfy more of its customers. This report focuses on two kinds of restaurants in Chicago- India, and Chinese restaurants. We will be using the Yelp API data source to collect data for each of these category of restaurants. This report will serve as a tutorial to help students learn how to perform all procedures for our OSEMN project- Obtain the data, Scrub the data, Explore the data, More exploring of the data using str, class, and summary command, and graphNing of table results.

2 Data: Obtain Data

Below are the the steps required to obtain the data from the Yelp data source.

2.1 First

The first step is to get your authentication keys which will be used to making the request on the Yelp data source.

```
consumerKey = "2aifVe3MSenp4H8JOZm08g"
consumerSecret = "jeS6rbhofi_AHlHGJx5JGV55OVA"
token = "ajihPQoK3DfL8c10RwW96NIC8ehEkvq4"
token_secret = "oX0XzaLvSKlLv-TBwspc6CpQnZE"
myapp = oauth_app("YELP", key=consumerKey, secret=consumerSecret)
sig=sign_oauth1.0(myapp, token=token,token_secret=token_secret)
```

2.2 Second

The second step is to use the paste0 command to request 20 for Indian Chicago-area restaurants

```
limit <- 20
URL <- paste0("http://api.yelp.com/v2/search/?limit=",limit,
              "&location=Chicago&term=indian%restaurant")
```

2.3 Third

The third step will make the request to the Yelp data source using the URL

```
responseList=GET(URL, sig)
responseContent = content(responseList)
```

2.4 Fourth

The fourth step will convert the data that was retrieved from the Yelp data source to a JSON object using the toJSON command and will also convert the data to a list object which will be used in creating a dataframe object called df.

```
jsonResponseContent <- toJSON(responseContent)
newResponseList=fromJSON(jsonResponseContent)
newDataframeObj <- data.frame(newResponseList)
```

2.5 Fifth

The fifth step will consist of column selections from the df dataframe and each one of them will be converted to a JSON object.

```
bRate <- newDataframeObj[,"businesses.rating"]
bName <- newDataframeObj[,"businesses.name"]
bReview <- newDataframeObj[,"businesses.review_count"]
bLocation <- newDataframeObj[,"businesses.location"]
  bCity <- bLocation[,"city"]
  bState <- bLocation[,"state_code"]
  bCountry <- bLocation[,"country_code"]
bIsClosed <- newDataframeObj[,"businesses.is_closed"]
bIsClaimed <- newDataframeObj[,"businesses.is_claimed"]

library(jsonlite)
bRatejson <- toJSON(bRate, pretty=TRUE)
bNamejson <- toJSON(bName, pretty=TRUE)
bReviewjson <- toJSON(bReview, pretty=TRUE)
bCityjson <- toJSON(bCity, pretty=TRUE)
bStatejson <- toJSON(bState, pretty=TRUE)
bCountryjson <- toJSON(bCountry, pretty=TRUE)
bIsClosedjson <- toJSON(bIsClosed, pretty=TRUE)
bIsClaimedjson <- toJSON(bIsClaimed, pretty=TRUE)
```

2.6 Sixth

The sixth step will convert each one of our JSON object to a list object which will be used to create a new data frame variable called newCSV.data. A csv file called Indian.csv will be used to store our new data frame variable-newCSV.data in a CSV format.

```
Ratings.mat <- fromJSON(bRatejson)
Name.mat <- fromJSON(bNamejson)
Reviews.mat <- fromJSON(bReviewjson)
City.mat <- fromJSON(bCityjson)
State.mat <- fromJSON(bStatejson)
Country.mat <- fromJSON(bCountryjson)
isClosed.mat <- fromJSON(bIsClosedjson)
```

```
isClaimed.mat <- fromJSON(bIsClaimedjson)

csv.file <- data.frame(Name.mat, Reviews.mat,
                       Ratings.mat, City.mat,
                       State.mat, Country.mat,
                       isClosed.mat, isClaimed.mat)
write.csv(csv.file, file="IndianData.csv")
```

3 Data: Scrub Data

The steps required for scrubbing the data are as follows.

3.1 First

The first thing will need is our data. The read.csv command along with the file name for our CSV file is used to retrieve our data from where we had stored it on our computer.

```
indian.obj <- read.csv("Data/Indian.csv")
```

3.2 Second

In the second step that follows, we basically, access the columns in the CSV file and re-format their data types using both the as.character and as.numeric command in R.

```
indian.obj$Name <- as.character(indian.obj$Name)
indian.obj$City <- as.character(indian.obj$City)
indian.obj$State <- as.character(indian.obj$State)
indian.obj$Country <- as.character(indian.obj$Country)
indian.obj$isClosed <- as.character(indian.obj$isClosed)
indian.obj$isClaimed <- as.character(indian.obj$isClaimed)
indian.obj$Ratings <- as.numeric(indian.obj$Ratings)
indian.obj$Reviews <- as.numeric(indian.obj$Reviews)
```

3.3 Third

In this step we will use the grep command to select only rows that we require from our data file that was read into R. In the steps below we are selecting only rows the their city as Chicago, which is opened and currently owned. Also the subset command was used in the last line to select rows where its column name, Reviews, was between the range of 99 and 900.

```
indianCity <- indian.obj[grep("Chicago", indian.obj$City),]
indianIsClosed <- indianCity[grep("FALSE", indianCity$isClosed),]
indianIsClaimed <- indianIsClosed[grep("TRUE", indianIsClosed$isClaimed),]
subIndian <- subset(indianIsClaimed, Reviews >= 100 & Reviews <= 899)
```

3.4 Fourth

The result of our last line in the previous step, produces a dataframe object consisting of all columns. In this step, we use that dataframe to select on two columns, Name and Ratings.

```
selectIndian <- data.frame(subIndian$Name, subIndian$Ratings)
```

3.5 Step Six

In the last step of our scrubbing procedure, the rename command in the plyr R package was used for renaming the selectIndian dataframe column headers.

```
library(plyr)
newIndian <- rename(selectIndian,
                    replace=c("subIndian.Name"="Name",
                              "subIndian.Ratings"="Ratings"))
```

4 Data: Explore

In this section we shall execute three commands to explore and learn more about the data. The class command shows the data type of objects. The str command shows the number of observation in each object. The summary command shows the data type, observation and statistical values for each object created. Below are the use of each of these commands.

```
class(newIndian$Name)
"character"
class(newIndian$Ratings)
"numeric"
class(newIndian)
"data.frame"
```

```
str(newIndian$Name)
 chr [1:14] "Naansense" "India House Restaurant" "Rangoli" "Tandoor Char House" ...
str(newIndian$Ratings)
 num [1:14] 4 4 4 4 3.5 3.5 4 3.5 3.5 3.5 ...
str(newIndian)
'data.frame':  14 obs. of  2 variables:
 $ Name   : chr  "Naansense" "India House Restaurant" "Rangoli" "Tandoor Char House"
 $ Ratings: num  4 4 4 4 3.5 3.5 4 3.5 3.5 3.5 ...
```

```
> summary(newIndian$Name)
   Length     Class      Mode
       14 character character
> summary(newIndian$Ratings)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    3.5     3.5     3.5     3.8     4.0     4.5
> summary(newIndian)
     Name              Ratings
 Length:14          Min.   :3.5
 Class :character   1st Qu.:3.5
 Mode  :character   Median :3.5
                    Mean   :3.8
                    3rd Qu.:4.0
                    Max.   :4.5
```

There are 14 observations and 2 variables from the data object newIndian. The minimum and maximum allowed values in the Ratings column of the newIndian object is 3.5 and 4.5. The Name column of the newIndian is a character type, and the Ratings column of the newIndian object is a numeric type.

5 Results

5.1 Tables

In this section, you will learn how to plot tables for the two data sets-myIndian.New and myChinese.New using the *kable* function in the *knitr* package. Table 1 represents the customer ratings for the Chicago area Indian restaurants and Table 2 represents the customer ratings for the Chicago area Chinese restaurants. Table 3 is the table of focus for the OSEMN project. It represents the average customer ratings for both kinds of Chicago area restaurants. Table 3 will be used for plotting our graphs in the next section.

Table 1: Indian Restaurants Ratings

```
> knitr::kable(newIndian, digits = 2, caption = "Indian Restaurants")

|Name                                  | Ratings|
|:-------------------------------------|-------:|
|Naansense                             |     4.0|
|India House Restaurant                |     4.0|
|Rangoli                               |     4.0|
|Tandoor Char House                    |     4.0|
|Bombay Wraps                          |     3.5|
|Jaipur                                |     3.5|
|Cumin                                 |     4.0|
|Mughal India                          |     3.5|
|Gaylord Fine Indian Cuisine           |     3.5|
|The Indian Garden                     |     3.5|
|Chicago Curry House                   |     3.5|
|Nepal House - Indian and Nepali Cuisine |     3.5|
|Luzzat                                |     4.5|
|Ghareeb Nawaz Express                 |     3.5|
```

Table 2: Chinese Restaurants Ratings

```
> knitr::kable(newChinese, digits = 2, caption = "Chinese Restaurants")

|Name                                    | Ratings|
|:---------------------------------------|-------:|
|Lao Sze Chuan-Downtown                  |     3.5|
|Chi Cafe                                |     4.0|
|Shanghai Terrace at The Peninsula Chicago |     4.0|
|Chengdu Impression                      |     4.0|
```

```
|Yummy Yummy Asian Cuisine                 |     4.0|
|MingHin Cuisine                           |     4.0|
|Yum Cha Dim Sum Parlor                    |     3.5|
|Fat Rice                                  |     4.0|
|Lao Hunan                                 |     3.5|
|Go 4 Food                                 |     4.0|
|Wow Bao                                   |     3.5|
|Take Me Out                               |     4.0|
|MAK Restaurant                            |     4.0|
|Wow Bao                                   |     3.5|
```

Table 3: Average Restaurants Ratings

```
> knitr::kable(avg.df, digits = 2, caption = "Average Rating")

|Restaurant Type |Average          |
|:---------------|:----------------|
|Indian          |3.75             |
|Chinese         |3.82142857142857 |
```

5.2 Graphs

Figure 1, is a graphical bar graph representation of our Table 3 in the previous section. The graphs shows that between the two kinds of restaurants, only a little fraction of difference in their average customer ratings exist. This little difference is not significant enough to draw a conclusion that the either one of the restaurants is likely to have Wi-Fi over the other. In fact, the graphs shows that both of these Chicago area restaurants are likely to have Wi-Fi.

Figure 2, on the other hands is a graphical strip plot graph showing the average customer ratings of our two Chicago area restaurants. The information in this graph is the same as Figure 1 but it displays this information only differently as a strip plot. Again not so much of a difference between this two kinds of restaurants and therefore both restaurants are likely to have Wi-Fi services.

Figure 1: Average customer rating: Bar Graph

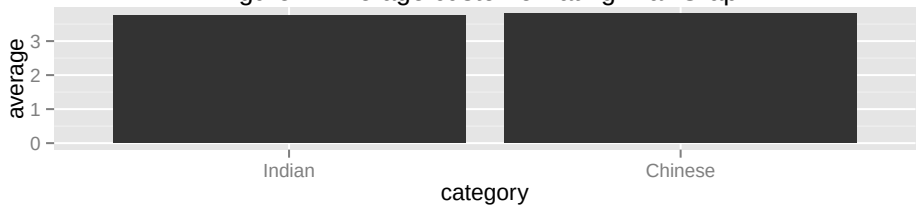

Table 2: Average customer rating: Strip Plots Graph