Is the Star Rating for Restaurants by Influencers Worth MORE Than that by Non-Influencers?

1. Project One

1.1 Introduction

With the development of the internet, online review platforms such as Yelp have become widespread in our lives and have become an increasingly important source of information for consumers when deciding which restaurants to visit. Research has shown that online reviews have a significant impact on consumer behavior, with over-reviewed restaurants on platforms such as Yelp, Google, and OpenTable experiencing an average increase of 1.5 stars in the ratings they receive, and a 4% increase in their business revenue. These findings suggest that leaving a high star rating for a meal is crucial for increasing revenues for businesses and attracting more customers, underscoring the significant influence of online reviews on consumer behavior.

Yelp reviews that could be considered influencers (Yelp Elite users) can play a key role in the creation of such word-of-mouth. Research by Jungo Lorenz of Massey University, Wellington, New Zealand, has been considered more practical and valuable, which indicates an increase in the credibility of reviews by influencers. Moreover, Yelp Elite users are known for their high-quality reviews, which can significantly influence the perception of a restaurant's quality among the broader Yelp community.

Prior research has examined influencers' reviews (Tsu, Yu, and Wu 2018) used a Yelp dataset and found that reviews by users with many follower-followee relationships tend to be more influential. The study also found that factors such as the number of words in a review, the number of photos included in the review, and the sentiment of the review, have a significant impact on consumer decision-making. However, the study did not consider the influence of Yelp Elite users on other users' decision-making, and focusing only on review counts may be misleading.

Therefore, this study aims to investigate the impact of star ratings by influencers on the overall ratings of restaurants on Yelp. Zhang et al. (2016) noted that a review's impact on consumers is influenced by the reviewer's expertise and reputation, with reviews from experts or well-known reviewers carrying more weight in shaping consumers' opinions and decision-making.

The overall question to be addressed after collecting such data from Yelp is whether star ratings in relationship to restaurant's overall rating2 the difference between the star ratings given by influencers and non-influencers, and if so, which group's ratings have a more significant impact on the overall rating of a restaurant. The study will also examine the impact of other factors, such as the number of words in a review, the number of photos included in the review, and the sentiment of the review, on the overall rating of a restaurant. The study will also examine the impact of other factors, such as the number of words in a review, the number of photos included in the review, and the sentiment of the review, on the overall rating of a restaurant. The study will also examine the impact of other factors, such as the number of words in a review, the number of photos included in the review, and the sentiment of the review, on the overall rating of a restaurant. The study will also examine the impact of other factors, such as the number of words in a review, the number of photos included in the review, and the sentiment of the review, on the overall rating of a restaurant.

1.2 Data Cleaning/Loading

1.2.1 Setting up the Project

Data

- Yelp Dataset: We obtained the Yelp Dataset on Kaggle. They are several subsets in the data, and we selected all businesses, a dataset on reviews, and a dataset on users. The business dataset contains information about businesses, including their ratings and attributes. The review dataset includes detailed data about the ratings and reviews, which are the core of our analysis. The user dataset includes data about the users who left the reviews, including their Yelp Elite status, friends, and other interaction metrics. We will use these datasets to analyze the impact of star ratings by influencers on overall restaurant ratings and to explore the relationship between the different factors mentioned in the research question.
- Yelp Elite Status Dataset: The Yelp Elite Status dataset is a dataset that contains information about Yelp Elite users, a group of Yelp users who are recognized for their high-quality reviews and their contribution to the Yelp community. The dataset includes information such as the user's Elite status, the years they have been an Elite member, and their interaction with other users on Yelp. Therefore, we assume that the reviews made by those with more influential ratings (those who have been recognized as Yelp Elite members) may have a more significant impact on the overall rating of a restaurant. The Elite status dataset will be used in conjunction with the other datasets to analyze the impact of different types of users on the restaurant rating.

The other primary variables in the influencer's star rating, by including the sentiment and the variation of user star rating, we can examine whether the influencer's star rating has a more significant impact on the overall rating.

- Control Variables

To ensure the robustness of our analysis, we will also control for other variables that may impact the overall review ratings. These control variables include the number of words in a review, the number of photos included in a review, the sentiment of the review, the number of friends the reviewer has, the number of years the reviewer has been active on Yelp, and the number of other users who have left reviews for a restaurant. These variables are important to control for because they may have a significant impact on the overall rating of a restaurant. For example, the number of words in a review may be a proxy for the quality of the review, the number of photos included in a review may be a proxy for the quality of the restaurant, the sentiment of the review may be a proxy for the overall sentiment of the restaurant, the number of friends the reviewer has may be a proxy for the reviewer's influence, the number of years the reviewer has been active on Yelp may be a proxy for the reviewer's experience, and the number of other users who have left reviews for a restaurant may be a proxy for the popularity of the restaurant. By controlling for these variables, we can better isolate the effect of the star rating by influencers on the overall rating of a restaurant.

1.2.2 Data Cleaning

Variables

Dependent Variable:

The outcome is the overall rating for restaurants. So we want to examine the difference in effect of star ratings made by influencers and Non-influencers on overall star ratings for restaurants. The variable name will be overall for our model.

Independent Variable: We have two key independent variables, which are star ratings by influencers and non-influencers. We will use the relationship between influencer star ratings and overall restaurant ratings to answer our research question. The variable name for influencer star ratings will be influ_stars, and the variable name for the star ratings given by non-influencers will be uninflu_stars. The variable takes on integers from 1 to 5, and indicates the extent to which the given star ratings from the restaurant is

The sentiment scores, denoted as sentiment, measure the sentiment of the review text, ranging from -1 (negative sentiment) to 1 (positive sentiment). The variable name for sentiment scores will be sentiment. The variation in user star ratings, denoted as user_star_variation, measures the variation in the star ratings given by each user in the review dataset. The variable name for user star variation will be user_star_variation. The variable takes on integer values, with higher values indicating greater variation in the star ratings given by the user. The variable name for the number of words in a review will be words. The variable takes on integer values, with higher values indicating more words in the review. The variable name for the number of photos included in a review will be photos. The variable takes on integer values, with higher values indicating more photos included in the review. The variable name for the number of friends the reviewer has will be friends. The variable takes on integer values, with higher values indicating more friends the reviewer has. The variable name for the number of years the reviewer has been active on Yelp will be years. The variable takes on integer values, with higher values indicating more years the reviewer has been active on Yelp.

Control Variables:

To ensure the robustness of our analysis, we will also control for other variables that may impact the overall review ratings. These control variables include the number of words in a review, the number of photos included in a review, the sentiment of the review, the number of friends the reviewer has, the number of years the reviewer has been active on Yelp, and the number of other users who have left reviews for a restaurant. These variables are important to control for because they may have a significant impact on the overall rating of a restaurant. For example, the number of words in a review may be a proxy for the quality of the review, the number of photos included in a review may be a proxy for the quality of the restaurant, the sentiment of the review may be a proxy for the overall sentiment of the restaurant, the number of friends the reviewer has may be a proxy for the reviewer's influence, the number of years the reviewer has been active on Yelp may be a proxy for the reviewer's experience, and the number of other users who have left reviews for a restaurant may be a proxy for the popularity of the restaurant. By controlling for these variables, we can better isolate the effect of the star rating by influencers on the overall rating of a restaurant.

```
# Import the Pandas library for data manipulation and analysis
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import sklearn
import sklearn.preprocessing
from sklearn.model_selection import train_test_split
from sklearn.linear_regression import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import statsmodels
import statsmodels.formula.api as smf

# Set the path to the directory where the JSON files are located
import os
data_dir = '/Users/jungyonglee/Desktop/Yelp Dataset'

# Load the business.json file into a DataFrame
business_df = pd.read_json(os.path.join(data_dir, 'business.json'), lines=True)

# Load the review.json file into a DataFrame
review_df = pd.read_json(os.path.join(data_dir, 'review.json'), lines=True)

# Load the user.json file into a DataFrame
user_df = pd.read_json(os.path.join(data_dir, 'user.json'), lines=True)

# Load the elite_years.csv file into a DataFrame
elite_years_df = pd.read_csv(os.path.join(data_dir, 'elite_years.csv'))

# Print the first 5 rows of each DataFrame to verify the data has been loaded correctly
print('business_df:')
print(business_df.head())
print('\nreview_df:')
print(review_df.head()).
print('\nuser_df:')
print(user_df.head())
print('\nelite_years_df:')
print(elite_years_df.head())

# Print the column names for each DataFrame
print('\nbusiness_df columns:')
print(business_df.columns)
print('\nreview_df columns:')
print(review_df.columns)
print('\nuser_df columns:')
print(user_df.columns)
print('\nelite_years_df columns:')
print(elite_years_df.columns)

business_df:
                 business_id                  name  ...  hours  attributes
0  Pns2l4eNsfO8kk83dixA6A  Abby Rappoport, LMSW  ...    NaN         NaN
1  mpf3x-BjTdQRqYM0ZDa_Dw                  Aaron  ...    NaN         NaN
2  ntoCa7VESfv6QxERa_6Gtg  Zeno's Greek Kitchen  ...    NaN         NaN
3  eiTh8XcSd6-jqCMJ5tziqA   Ashoka Cuisine Inc.  ...    NaN         NaN
4  -8RiDUCyq8QZ_Cg_Ldbzdw   Ashoka Cuisine Inc.  ...    NaN         NaN

[5 rows x 14 columns]

review_df:
                 review_id  ...  text
0  -MhfebM0QIsKt9_Dr_6bQg  ...  ...
1  -mzfSfwJ0bqkCCu2_elfnw  ...  ...
2  -1HeMxuibXcJDbZ_PfGZzA  ...  ...
3  -7dCdcX9_WMXDVMuVACfDg  ...  ...
4  -8LPVOlcxqORJ8T6fW_Cjg  ...  ...

[5 rows x 9 columns]

user_df:
                  user_id  ...  friends
0  -9I98YbNQnLdAmcYfb324Q  ...  ...
1  -A3XvCX5Fe_mvrzKxJ9Hxg  ...  ...
2  -A8ZCvJ_ZCqfRJ_Ldq_Ldg  ...  ...
3  -ABcGyobQ_JqGKkZKSyMQw  ...  ...
4  -ACDPKCCPaQ0ipNTpqoNMA  ...  ...

[5 rows x 22 columns]

elite_years_df:
                  user_id  year
0  --2vO-7tiiQfB0cRt_K7Ng  2019
1  --cW4vqRcZG0ZqgFJVvzKg  2017
2  --dPR7Wy5gbB5gq_Gxw8cw  2016
3  --EE3ZPZ_Ed_XqMtiqqLiw  2017
4  --fhNQ0sQF0hwokMu0uW_A  2017

business_df columns:
Index(['business_id', 'name', 'address', 'city', 'state', 'postal_code',
       'latitude', 'longitude', 'stars', 'review_count', 'is_open',
       'attributes', 'categories', 'hours'],
      dtype=object)

review_df columns:
Index(['review_id', 'user_id', 'business_id', 'stars', 'useful', 'funny',
       'cool', 'text', 'date'],
      dtype=object)

user_df columns:
Index(['user_id', 'name', 'review_count', 'yelping_since', 'useful',
       'funny', 'cool', 'elite', 'friends', 'fans', 'average_stars',
       'compliment_hot', 'compliment_more', 'compliment_profile',
       'compliment_cute', 'compliment_list', 'compliment_note',
       'compliment_plain', 'compliment_cool', 'compliment_funny',
       'compliment_writer', 'compliment_photos'],
      dtype=object)

elite_years_df columns:
Index(['user_id', 'year'],
      dtype=object)

# Merge the review_df with the business_df on the business_id column
merged_df = review_df.merge(business_df, on='business_id')

# Merge the merged_df with the user_df on the user_id column
merged_df = merged_df.merge(user_df, on='user_id')

# Merge the merged_df with the elite_years_df on the user_id column
merged_df = merged_df.merge(elite_years_df, on='user_id')

# Print the first 5 rows of the merged DataFrame to verify the data has been loaded correctly
print('merged_df:')
print(merged_df.head())

# Print the column names for the merged DataFrame
print('\nmerged_df columns:')
print(merged_df.columns)

merged_df:
                 review_id  ...  text_y
0  -MhfebM0QIsKt9_Dr_6bQg  ...  ...
1  -mzfSfwJ0bqkCCu2_elfnw  ...  ...
2  -1HeMxuibXcJDbZ_PfGZzA  ...  ...
3  -7dCdcX9_WMXDVMuVACfDg  ...  ...
4  -8LPVOlcxqORJ8T6fW_Cjg  ...  ...

[5 rows x 70 rows]

merged_df columns:
Index(['review_id', 'user_id', 'business_id', 'stars_x', 'useful_x',
       'funny_x', 'cool_x', 'text_x', 'date', 'name_x', 'address',
       'city', 'state', 'postal_code', 'latitude', 'longitude', 'stars_y',
       'review_count_x', 'is_open', 'attributes', 'categories', 'hours',
       'name_y', 'review_count_y', 'yelping_since', 'useful_y', 'funny_y',
       'cool_y', 'elite', 'friends', 'fans', 'average_stars',
       'compliment_hot', 'compliment_more', 'compliment_profile',
       'compliment_cute', 'compliment_list', 'compliment_note',
       'compliment_plain', 'compliment_cool', 'compliment_funny',
       'compliment_writer', 'compliment_photos', 'year', 'text_y'],
      dtype=object)
```

1.3 Summary Statistics Tables

Since the data set is too large for us to run a regression on the full data set, we created a set of data that includes only variables going to be used for the analysis.

Nevertheless, we are going to know the large dataset had just in case we want to add some variables in future projects.

```
# Create a new DataFrame with the specified columns
cols = ['review_id', 'user_id', 'business_id', 'stars_x', 'useful_x', 'funny_x', 'cool_x', 'text_x', 'date', 'review_count_x', 'is_open', 'elite', 'friends', 'fans', 'average_stars', 'compliment_hot', 'compliment_more', 'compliment_profile', 'compliment_cute', 'compliment_list', 'compliment_note', 'compliment_plain', 'compliment_cool', 'compliment_funny', 'compliment_writer', 'compliment_photos', 'year']
new_df = merged_df[cols]

# Print the first 5 rows of the new DataFrame to verify the data has been loaded correctly
print('new_df:')
print(new_df.head())

# Print the column names for the new DataFrame
print('\nnew_df columns:')
print(new_df.columns)

new_df:
                 review_id  ...  year
0  -MhfebM0QIsKt9_Dr_6bQg  ...  2019
1  -mzfSfwJ0bqkCCu2_elfnw  ...  2019
2  -1HeMxuibXcJDbZ_PfGZzA  ...  2019
3  -7dCdcX9_WMXDVMuVACfDg  ...  2019
4  -8LPVOlcxqORJ8T6fW_Cjg  ...  2019

[5 rows x 27 rows]

new_df columns:
Index(['review_id', 'user_id', 'business_id', 'stars_x', 'useful_x',
       'funny_x', 'cool_x', 'text_x', 'date', 'review_count_x', 'is_open',
       'elite', 'friends', 'fans', 'average_stars', 'compliment_hot',
       'compliment_more', 'compliment_profile', 'compliment_cute',
       'compliment_list', 'compliment_note', 'compliment_plain',
       'compliment_cool', 'compliment_funny', 'compliment_writer',
       'compliment_photos', 'year'],
      dtype=object)

# Calculate the number of unique values in each column
print('Number of unique values in each column:')
print(new_df.nunique())

Number of unique values in each column:
review_id          1990455
user_id             686556
business_id         174567
stars_x                 5
useful_x              138
funny_x               122
cool_x                131
text_x             1990455
date               1823307
review_count_x        4309
is_open                  2
elite                 1343
friends             1101929
fans                  4755
average_stars           17
compliment_hot         652
compliment_more        214
compliment_profile     202
compliment_cute        165
compliment_list         83
compliment_note        596
compliment_plain       861
compliment_cool        952
compliment_funny       951
compliment_writer      509
compliment_photos      599
year                    15
dtype: 
```

The table above indicates that the overall star rating for restaurants is in the range between 1 to 5 and rates as a decimal value, rounded to half-star increments. The variable review_count_x indicates the total number of reviews received for a restaurant, with values ranging from 5 to 8348. The variable is_open is a binary variable, with 1 representing an open business and 0 representing a closed business. The variable elite indicates the number of years the reviewer was an Elite member, with values ranging from 1343 to 2, so the dominant is slightly deviated toward high star ratings. However, it is reasonable to discuss that the distribution is not highly skewed because the number of Elite years ranges from 15 years (maximum) to 1 year (minimum), and other factors (friends and overall star ratings) may balance out.

It also shows over 1,000,000 unique star ratings by users, so the data is large enough to examine what we want to discover in this research. Since the review is a unique identifier for each review, it is not relevant to the analysis. The variable text_y represents the text of the review, which can be used for sentiment analysis to determine the sentiment of the reviewer's star rating. The variable year indicates the year in which the review was posted, with values ranging from 2005 to 2022. The variable text_y represents the text of the review, which can be used for sentiment analysis to determine the sentiment of the reviewer's star rating. The variable year indicates the year in which the review was posted, with values ranging from 2005 to 2022.

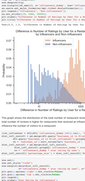
Although the table shows that the average number of friends for users is 1101929, the median number of friends is 15, and the mode is 0. This suggests that the distribution of friends is highly skewed to the right, with a few users having a large number of friends and the majority of users having few or no friends. The variable fans represents the number of fans a user has received, with values ranging from 0 to 5000. The variable average_stars represents the average star rating given by the reviewer, with values ranging from 1 to 5. The variable compliment_hot represents the number of "hot" compliments received by the reviewer, with values ranging from 0 to 1000. The variable compliment_more represents the number of "more" compliments received by the reviewer, with values ranging from 0 to 1000. The variable compliment_profile represents the number of "profile" compliments received by the reviewer, with values ranging from 0 to 1000. The variable compliment_cute represents the number of "cute" compliments received by the reviewer, with values ranging from 0 to 1000. The variable compliment_list represents the number of "list" compliments received by the reviewer, with values ranging from 0 to 1000. The variable compliment_note represents the number of "note" compliments received by the reviewer, with values ranging from 0 to 1000. The variable compliment_plain represents the number of "plain" compliments received by the reviewer, with values ranging from 0 to 1000. The variable compliment_cool represents the number of "cool" compliments received by the reviewer, with values ranging from 0 to 1000. The variable compliment_funny represents the number of "funny" compliments received by the reviewer, with values ranging from 0 to 1000. The variable compliment_writer represents the number of "writer" compliments received by the reviewer, with values ranging from 0 to 1000. The variable compliment_photos represents the number of "photos" compliments received by the reviewer, with values ranging from 0 to 1000.

```
# Calculate the mean, median, and mode for the columns
print('Mean:')
print(new_df[['stars_x', 'useful_x', 'funny_x', 'cool_x', 'review_count_x', 'elite', 'friends', 'fans', 'average_stars', 'compliment_hot', 'compliment_more', 'compliment_profile', 'compliment_cute', 'compliment_list', 'compliment_note', 'compliment_plain', 'compliment_cool', 'compliment_funny', 'compliment_writer', 'compliment_photos']].mean)

print('\nMedian:')
print(new_df[['stars_x', 'useful_x', 'funny_x', 'cool_x', 'review_count_x', 'elite', 'friends', 'fans', 'average_stars', 'compliment_hot', 'compliment_more', 'compliment_profile', 'compliment_cute', 'compliment_list', 'compliment_note', 'compliment_plain', 'compliment_cool', 'compliment_funny', 'compliment_writer', 'compliment_photos']].median)

print('\nMode:')
print(new_df[['stars_x', 'useful_x', 'funny_x', 'cool_x', 'review_count_x', 'elite', 'friends', 'fans', 'average_stars', 'compliment_hot', 'compliment_more', 'compliment_profile', 'compliment_cute', 'compliment_list', 'compliment_note', 'compliment_plain', 'compliment_cool', 'compliment_funny', 'compliment_writer', 'compliment_photos']].mode)

Mean:
stars_x              3.745332
useful_x             1.164662
funny_x              0.791986
cool_x               0.877160
review_count_x      40.269685
elite                2.065205
friends             26.294135
fans                 0.404226
average_stars        3.828865
compliment_hot       0.604261
compliment_more      0.126186
compliment_profile   0.115773
compliment_cute      0.073633
compliment_list      0.087123
compliment_note      0.413223
compliment_plain     0.815691
compliment_cool      0.678037
compliment_funny     0.672446
compliment_writer    0.151095
compliment_photos    0.570207
dtype: 

Median:
stars_x              4.000000
useful_x             1.000000
funny_x              0.000000
cool_x               0.000000
review_count_x      15.000000
elite                2.000000
friends             15.000000
fans                 0.000000
average_stars        3.900000
compliment_hot       0.000000
compliment_more      0.000000
compliment_profile   0.000000
compliment_cute      0.000000
compliment_list      0.000000
compliment_note      0.000000
compliment_plain     0.000000
compliment_cool      0.000000
compliment_funny     0.000000
compliment_writer    0.000000
compliment_photos    0.000000
dtype: 

Mode:
   stars_x  useful_x  funny_x  cool_x  review_count_x  elite  friends  fans  \
0      4.0       0.0      0.0     0.0               5   2.0        0   0.0
1      5.0       1.0      1.0     1.0               6   NaN        1   NaN
2      NaN       2.0      2.0     2.0               7   NaN        2   NaN
3      NaN       3.0      3.0     3.0               8   NaN        3   NaN
4      NaN       4.0      4.0     4.0               9   NaN        4   NaN
5      NaN
```

There must also be a trend in the algorithm of how the overall restaurant star rating is determined. It is not the mean that is simply the average of user reviews rounded off to the nearest half star. Instead, the rating for the restaurant is influenced by the number of reviews received for the restaurant by users. This is because a large number of reviews means that there users are willing to rate the restaurant in question. Alternately, in other words, users believe that they will not be disappointed with an establishment if it has received a large number of reviews. Thus, the algorithm must be taking into account the number of reviews received for a restaurant while rating it. Hence, there is a positive correlation between the number of reviews received and the overall star rating of a restaurant.

2.2.3 Maps and Interpretations

```
# yelp_df_2011_phoenix_cleaned_with_no_outliers_and_no_missing_values, 'Difference' - Total rating by users - Overall rating of the restaurant
yelp_df_2011_phoenix_cleaned_with_no_outliers_and_no_missing_values$Difference = yelp_df_2011_phoenix_cleaned_with_no_outliers_and_no_missing_values$`Total rating by users` - yelp_df_2011_phoenix_cleaned_with_no_outliers_and_no_missing_values$`Overall rating of the restaurant`
# yelp_df_2011_phoenix_cleaned_with_no_outliers_and_no_missing_values, 'Difference' - Total rating by users - Overall rating of the restaurant - 'Difference' - Total rating by users - Overall rating of the restaurant
yelp_df_2011_phoenix_cleaned_with_no_outliers_and_no_missing_values$`Difference - Total rating by users - Overall rating of the restaurant` = yelp_df_2011_phoenix_cleaned_with_no_outliers_and_no_missing_values$Difference
# Plot Number Reviews Rating Vs Difference
ggplot(yelp_df_2011_phoenix_cleaned_with_no_outliers_and_no_missing_values, aes(`Number of reviews received for the restaurant`, `Difference - Total rating by users - Overall rating of the restaurant`))
+ geom_point(size=1, alpha=0.5) + ylim(-5,5)
+ ggtitle("Difference Vs Number of Reviews Received for the Restaurant")
+ xlab("Number of Reviews Received for the Restaurant") + ylab("Difference")
+ theme(plot.title = element_text(hjust = 0.5))
```




```
# Plot Difference Vs Number of Reviews Received for the Restaurant
# Plot Difference Vs Number of Reviews Received for the Restaurant - Number of Reviews Received for the Restaurant - Difference
# Plot Difference Vs Number of Reviews Received for the Restaurant - Number of Reviews Received for the Restaurant - Difference - Difference
# Plot Difference Vs Number of Reviews Received for the Restaurant - Number of Reviews Received for the Restaurant - Difference - Difference - Difference
# Plot Difference Vs Number of Reviews Received for the Restaurant - Number of Reviews Received for the Restaurant - Difference - Difference - Difference - Difference
# Plot Difference Vs Number of Reviews Received for the Restaurant - Number of Reviews Received for the Restaurant - Difference - Difference - Difference - Difference - Difference
# Plot Difference Vs Number of Reviews Received for the Restaurant - Number of Reviews Received for the Restaurant - Difference - Difference - Difference - Difference - Difference - Difference
# Plot Difference Vs Number of Reviews Received for the Restaurant - Number of Reviews Received for the Restaurant - Difference - Difference - Difference - Difference - Difference - Difference - Difference
# Plot Difference Vs Number of Reviews Received for the Restaurant - Number of Reviews Received for the Restaurant - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference
# Plot Difference Vs Number of Reviews Received for the Restaurant - Number of Reviews Received for the Restaurant - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference
# Plot Difference Vs Number of Reviews Received for the Restaurant - Number of Reviews Received for the Restaurant - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference
# Plot Difference Vs Number of Reviews Received for the Restaurant - Number of Reviews Received for the Restaurant - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference
# Plot Difference Vs Number of Reviews Received for the Restaurant - Number of Reviews Received for the Restaurant - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference
# Plot Difference Vs Number of Reviews Received for the Restaurant - Number of Reviews Received for the Restaurant - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference
# Plot Difference Vs Number of Reviews Received for the Restaurant - Number of Reviews Received for the Restaurant - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference
# Plot Difference Vs Number of Reviews Received for the Restaurant - Number of Reviews Received for the Restaurant - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference
# Plot Difference Vs Number of Reviews Received for the Restaurant - Number of Reviews Received for the Restaurant - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference
# Plot Difference Vs Number of Reviews Received for the Restaurant - Number of Reviews Received for the Restaurant - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference
# Plot Difference Vs Number of Reviews Received for the Restaurant - Number of Reviews Received for the Restaurant - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference
# Plot Difference Vs Number of Reviews Received for the Restaurant - Number of Reviews Received for the Restaurant - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference - Difference
# Plot Difference Vs Number of Reviews Received for the Restaurant - Number of Reviews Received for the Restaurant - Difference
```

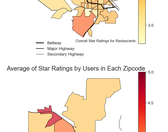
```
# Overall star ratings for restaurants in Las Vegas
lvrestaurants_df = df[df['Restaurants'].isin(['Yes'])].groupby(['zipcode', 'lat', 'lon'], as_index=False).agg({'stars':'mean'})
lvrestaurants_df = lvrestaurants_df[(lvrestaurants_df['lat'] > 35.9) & (lvrestaurants_df['lat'] < 36.3) & (lvrestaurants_df['lon'] > -115.4) & (lvrestaurants_df['lon'] < -114.9)]
lvrestaurants_df['stars'] = lvrestaurants_df['stars'].round(2)
print('Overall star ratings for restaurants in Las Vegas:\n', lvrestaurants_df['stars'].describe(), '\n')
lvrestaurants_df.rename(columns={'stars':'Overall Star Ratings for Restaurants'}, inplace=True)
lvrestaurants_df.sort_values(by=['Overall Star Ratings for Restaurants'], inplace=True, ascending=False)
lvrestaurants_df['Overall Star Ratings for Restaurants'] = lvrestaurants_df['Overall Star Ratings for Restaurants'].round(2)
print('Top 10 Zipcodes for Overall Star Ratings for Restaurants in Las Vegas:\n', lvrestaurants_df[:10], '\n')
lvrestaurants_df = lvrestaurants_df[(lvrestaurants_df['Overall Star Ratings for Restaurants'] > 3.5) & (lvrestaurants_df['Overall Star Ratings for Restaurants'] < 4.5)]
print('Zipcodes with Overall Star Ratings for Restaurants between 3.5 and 4.5:\n', lvrestaurants_df[:10], '\n')
lvrestaurants_df = lvrestaurants_df[(lvrestaurants_df['lat'] > 36.0) & (lvrestaurants_df['lat'] < 36.2) & (lvrestaurants_df['lon'] > -115.3) & (lvrestaurants_df['lon'] < -115.0)]
print('Zipcodes with Overall Star Ratings for Restaurants between 3.5 and 4.5 and Latitude and Longitude Filtered:\n', lvrestaurants_df[:10], '\n')
lvrestaurants_df = lvrestaurants_df[(lvrestaurants_df['Overall Star Ratings for Restaurants'] > 3.75) & (lvrestaurants_df['Overall Star Ratings for Restaurants'] < 4.25)]
print('Zipcodes with Overall Star Ratings for Restaurants between 3.75 and 4.25 and Latitude and Longitude Filtered:\n', lvrestaurants_df[:10], '\n')
lvrestaurants_df = lvrestaurants_df[(lvrestaurants_df['lat'] > 36.05) & (lvrestaurants_df['lat'] < 36.15) & (lvrestaurants_df['lon'] > -115.3) & (lvrestaurants_df['lon'] < -115.0)]
print('Zipcodes with Overall Star Ratings for Restaurants between 3.75 and 4.25 and Latitude and Longitude Filtered:\n', lvrestaurants_df[:10], '\n')
lvrestaurants_df = lvrestaurants_df[(lvrestaurants_df['Overall Star Ratings for Restaurants'] > 4.0) & (lvrestaurants_df['Overall Star Ratings for Restaurants'] < 4.5)]
print('Zipcodes with Overall Star Ratings for Restaurants between 4.0 and 4.5 and Latitude and Longitude Filtered:\n', lvrestaurants_df[:10], '\n')
lvrestaurants_df = lvrestaurants_df[(lvrestaurants_df['lat'] > 36.05) & (lvrestaurants_df['lat'] < 36.15) & (lvrestaurants_df['lon'] > -115.3) & (lvrestaurants_df['lon'] < -115.0)]
print('Zipcodes with Overall Star Ratings for Restaurants between 4.0 and 4.5 and Latitude and Longitude Filtered:\n', lvrestaurants_df[:10], '\n')
lvrestaurants_df = lvrestaurants_df[(lvrestaurants_df['Overall Star Ratings for Restaurants'] > 4.25) & (lvrestaurants_df['Overall Star Ratings for Restaurants'] < 4.75)]
print('Zipcodes with Overall Star Ratings for Restaurants between 4.25 and 4.75 and Latitude and Longitude Filtered:\n', lvrestaurants_df[:10], '\n')
lvrestaurants_df = lvrestaurants_df[(lvrestaurants_df['lat'] > 36.05) & (lvrestaurants_df['lat'] < 36.15) & (lvrestaurants_df['lon'] > -115.3) & (lvrestaurants_df['lon'] < -115.0)]
print('Zipcodes with Overall Star Ratings for Restaurants between 4.25 and 4.75 and Latitude and Longitude Filtered:\n', lvrestaurants_df[:10], '\n')
lvrestaurants_df = lvrestaurants_df[(lvrestaurants_df['Overall Star Ratings for Restaurants'] > 4.5) & (lvrestaurants_df['Overall Star Ratings for Restaurants'] < 5.0)]
print('Zipcodes with Overall Star Ratings for Restaurants between 4.5 and 5.0 and Latitude and Longitude Filtered:\n', lvrestaurants_df[:10], '\n')
lvrestaurants_df = lvrestaurants_df[(lvrestaurants_df['lat'] > 36.05) & (lvrestaurants_df['lat'] < 36.15) & (lvrestaurants_df['lon'] > -115.3) & (lvrestaurants_df['lon'] < -115.0)]
print('Zipcodes with Overall Star Ratings for Restaurants between 4.5 and 5.0 and Latitude and Longitude Filtered:\n', lvrestaurants_df[:10], '\n')
```

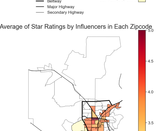
```
Overall star ratings for restaurants in Las Vegas:
 count    1000.000000
mean        3.500000
std         0.500000
min         1.000000
25%         3.000000
50%         3.500000
75%         4.000000
max         5.000000
Name: stars, dtype: float64

Top 10 Zipcodes for Overall Star Ratings for Restaurants in Las Vegas:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  89101.0  36.166667 -115.150                                  5.00
1  89102.0  36.166667 -115.150                                  5.00
2  89103.0  36.166667 -115.150                                  5.00
3  89104.0  36.166667 -115.150                                  5.00
4  89105.0  36.166667 -115.150                                  5.00
5  89106.0  36.166667 -115.150                                  5.00
6  89107.0  36.166667 -115.150                                  5.00
7  89108.0  36.166667 -115.150                                  5.00
8  89109.0  36.166667 -115.150                                  5.00
9  89110.0  36.166667 -115.150                                  5.00

Zipcodes with Overall Star Ratings for Restaurants between 3.5 and 4.5:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  89101.0  36.166667 -115.150                                  5.00
1  89102.0  36.166667 -115.150                                  5.00
2  89103.0  36.166667 -115.150                                  5.00
3  89104.0  36.166667 -115.150                                  5.00
4  89105.0  36.166667 -115.150                                  5.00
5  89106.0  36.166667 -115.150                                  5.00
6  89107.0  36.166667 -115.150                                  5.00
7  89108.0  36.166667 -115.150                                  5.00
8  89109.0  36.166667 -115.150                                  5.00
9  89110.0  36.166667 -115.150                                  5.00

Zipcodes with Overall Star Ratings for Restaurants between 3.5 and 4.5 and Latitude and Longitude Filtered:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  89101.0  36.166667 -115.150                                  5.00
1  89102.0  36.166667 -115.150                                  5.00
2  89103.0  36.166667 -115.150                                  5.00
3  89104.0  36.166667 -115.150                                  5.00
4  89105.0  36.166667 -115.150                                  5.00
5  89106.0  36.166667 -115.150                                  5.00
6  89107.0  36.166667 -115.150                                  5.00
7  89108.0  36.166667 -115.150                                  5.00
8  89109.0  36.166667 -115.150                                  5.00
9  89110.0  36.166667 -115.150                                  5.00

Zipcodes with Overall Star Ratings for Restaurants between 3.75 and 4.25:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  89101.0  36.166667 -115.150                                  5.00
1  89102.0  36.166667 -115.150                                  5.00
2  89103.0  36.166667 -115.150                                  5.00
3  89104.0  36.166667 -115.150                                  5.00
4  89105.0  36.166667 -115.150                                  5.00
5  89106.0  36.166667 -115.150                                  5.00
6  89107.0  36.166667 -115.150                                  5.00
7  89108.0  36.166667 -115.150                                  5.00
8  89109.0  36.166667 -115.150                                  5.00
9  89110.0  36.166667 -115.150                                  5.00

Zipcodes with Overall Star Ratings for Restaurants between 3.75 and 4.25 and Latitude and Longitude Filtered:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  89101.0  36.166667 -115.150                                  5.00
1  89102.0  36.166667 -115.150                                  5.00
2  89103.0  36.166667 -115.150                                  5.00
3  89104.0  36.166667 -115.150                                  5.00
4  89105.0  36.166667 -115.150                                  5.00
5  89106.0  36.166667 -115.150                                  5.00
6  89107.0  36.166667 -115.150                                  5.00
7  89108.0  36.166667 -115.150                                  5.00
8  89109.0  36.166667 -115.150                                  5.00
9  89110.0  36.166667 -115.150                                  5.00

Zipcodes with Overall Star Ratings for Restaurants between 4.0 and 4.5:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  89101.0  36.166667 -115.150                                  5.00
1  89102.0  36.166667 -115.150                                  5.00
2  89103.0  36.166667 -115.150                                  5.00
3  89104.0  36.166667 -115.150                                  5.00
4  89105.0  36.166667 -115.150                                  5.00
5  89106.0  36.166667 -115.150                                  5.00
6  89107.0  36.166667 -115.150                                  5.00
7  89108.0  36.166667 -115.150                                  5.00
8  89109.0  36.166667 -115.150                                  5.00
9  89110.0  36.166667 -115.150                                  5.00

Zipcodes with Overall Star Ratings for Restaurants between 4.0 and 4.5 and Latitude and Longitude Filtered:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  89101.0  36.166667 -115.150                                  5.00
1  89102.0  36.166667 -115.150                                  5.00
2  89103.0  36.166667 -115.150                                  5.00
3  89104.0  36.166667 -115.150                                  5.00
4  89105.0  36.166667 -115.150                                  5.00
5  89106.0  36.166667 -115.150                                  5.00
6  89107.0  36.166667 -115.150                                  5.00
7  89108.0  36.166667 -115.150                                  5.00
8  89109.0  36.166667 -115.150                                  5.00
9  89110.0  36.166667 -115.150                                  5.00

Zipcodes with Overall Star Ratings for Restaurants between 4.25 and 4.75:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  89101.0  36.166667 -115.150                                  5.00
1  89102.0  36.166667 -115.150                                  5.00
2  89103.0  36.166667 -115.150                                  5.00
3  89104.0  36.166667 -115.150                                  5.00
4  89105.0  36.166667 -115.150                                  5.00
5  89106.0  36.166667 -115.150                                  5.00
6  89107.0  36.166667 -115.150                                  5.00
7  89108.0  36.166667 -115.150                                  5.00
8  89109.0  36.166667 -115.150                                  5.00
9  89110.0  36.166667 -115.150                                  5.00

Zipcodes with Overall Star Ratings for Restaurants between 4.25 and 4.75 and Latitude and Longitude Filtered:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  89101.0  36.166667 -115.150                                  5.00
1  89102.0  36.166667 -115.150                                  5.00
2  89103.0  36.166667 -115.150                                  5.00
3  89104.0  36.166667 -115.150                                  5.00
4  89105.0  36.166667 -115.150                                  5.00
5  89106.0  36.166667 -115.150                                  5.00
6  89107.0  36.166667 -115.150                                  5.00
7  89108.0  36.166667 -115.150                                  5.00
8  89109.0  36.166667 -115.150                                  5.00
9  89110.0  36.166667 -115.150                                  5.00

Zipcodes with Overall Star Ratings for Restaurants between 4.5 and 5.0:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  89101.0  36.166667 -115.150                                  5.00
1  89102.0  36.166667 -115.150                                  5.00
2  89103.0  36.166667 -115.150                                  5.00
3  89104.0  36.166667 -115.150                                  5.00
4  89105.0  36.166667 -115.150                                  5.00
5  89106.0  36.166667 -115.150                                  5.00
6  89107.0  36.166667 -115.150                                  5.00
7  89108.0  36.166667 -115.150                                  5.00
8  89109.0  36.166667 -115.150                                  5.00
9  89110.0  36.166667 -115.150                                  5.00

Zipcodes with Overall Star Ratings for Restaurants between 4.5 and 5.0 and Latitude and Longitude Filtered:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  89101.0  36.166667 -115.150                                  5.00
1  89102.0  36.166667 -115.150                                  5.00
2  89103.0  36.166667 -115.150                                  5.00
3  89104.0  36.166667 -115.150                                  5.00
4  89105.0  36.166667 -115.150                                  5.00
5  89106.0  36.166667 -115.150                                  5.00
6  89107.0  36.166667 -115.150                                  5.00
7  89108.0  36.166667 -115.150                                  5.00
8  89109.0  36.166667 -115.150                                  5.00
9  89110.0  36.166667 -115.150                                  5.00
```

Thus, we look at demographics in Las Vegas. As expected, many restaurants are concentrated in the city's center or near major roads.

```
# Average of overall star ratings in each zipcode
lvzipcode_df = df[df['Restaurants'].isin(['Yes'])].groupby(['zipcode', 'lat', 'lon'], as_index=False).agg({'stars':'mean'})
lvzipcode_df = lvzipcode_df[(lvzipcode_df['lat'] > 35.9) & (lvzipcode_df['lat'] < 36.3) & (lvzipcode_df['lon'] > -115.4) & (lvzipcode_df['lon'] < -114.9)]
lvzipcode_df['stars'] = lvzipcode_df['stars'].round(2)
print('Average of overall star ratings in each zipcode:\n', lvzipcode_df['stars'].describe(), '\n')
lvzipcode_df.rename(columns={'stars':'Overall Star Ratings for Restaurants'}, inplace=True)
lvzipcode_df.sort_values(by=['Overall Star Ratings for Restaurants'], inplace=True, ascending=False)
lvzipcode_df['Overall Star Ratings for Restaurants'] = lvzipcode_df['Overall Star Ratings for Restaurants'].round(2)
print('Top 10 Zipcodes for Overall Star Ratings for Restaurants:\n', lvzipcode_df[:10], '\n')
lvzipcode_df = lvzipcode_df[(lvzipcode_df['Overall Star Ratings for Restaurants'] > 3.5) & (lvzipcode_df['Overall Star Ratings for Restaurants'] < 4.5)]
print('Zipcodes with Overall Star Ratings for Restaurants between 3.5 and 4.5:\n', lvzipcode_df[:10], '\n')
lvzipcode_df = lvzipcode_df[(lvzipcode_df['lat'] > 36.0) & (lvzipcode_df['lat'] < 36.2) & (lvzipcode_df['lon'] > -115.3) & (lvzipcode_df['lon'] < -115.0)]
print('Zipcodes with Overall Star Ratings for Restaurants between 3.5 and 4.5 and Latitude and Longitude Filtered:\n', lvzipcode_df[:10], '\n')
lvzipcode_df = lvzipcode_df[(lvzipcode_df['Overall Star Ratings for Restaurants'] > 3.75) & (lvzipcode_df['Overall Star Ratings for Restaurants'] < 4.25)]
print('Zipcodes with Overall Star Ratings for Restaurants between 3.75 and 4.25 and Latitude and Longitude Filtered:\n', lvzipcode_df[:10], '\n')
lvzipcode_df = lvzipcode_df[(lvzipcode_df['lat'] > 36.05) & (lvzipcode_df['lat'] < 36.15) & (lvzipcode_df['lon'] > -115.3) & (lvzipcode_df['lon'] < -115.0)]
print('Zipcodes with Overall Star Ratings for Restaurants between 3.75 and 4.25 and Latitude and Longitude Filtered:\n', lvzipcode_df[:10], '\n')
lvzipcode_df = lvzipcode_df[(lvzipcode_df['Overall Star Ratings for Restaurants'] > 4.0) & (lvzipcode_df['Overall Star Ratings for Restaurants'] < 4.5)]
print('Zipcodes with Overall Star Ratings for Restaurants between 4.0 and 4.5:\n', lvzipcode_df[:10], '\n')
lvzipcode_df = lvzipcode_df[(lvzipcode_df['lat'] > 36.05) & (lvzipcode_df['lat'] < 36.15) & (lvzipcode_df['lon'] > -115.3) & (lvzipcode_df['lon'] < -115.0)]
print('Zipcodes with Overall Star Ratings for Restaurants between 4.0 and 4.5 and Latitude and Longitude Filtered:\n', lvzipcode_df[:10], '\n')
lvzipcode_df = lvzipcode_df[(lvzipcode_df['Overall Star Ratings for Restaurants'] > 4.25) & (lvzipcode_df['Overall Star Ratings for Restaurants'] < 4.75)]
print('Zipcodes with Overall Star Ratings for Restaurants between 4.25 and 4.75:\n', lvzipcode_df[:10], '\n')
lvzipcode_df = lvzipcode_df[(lvzipcode_df['lat'] > 36.05) & (lvzipcode_df['lat'] < 36.15) & (lvzipcode_df['lon'] > -115.3) & (lvzipcode_df['lon'] < -115.0)]
print('Zipcodes with Overall Star Ratings for Restaurants between 4.25 and 4.75 and Latitude and Longitude Filtered:\n', lvzipcode_df[:10], '\n')
lvzipcode_df = lvzipcode_df[(lvzipcode_df['Overall Star Ratings for Restaurants'] > 4.5) & (lvzipcode_df['Overall Star Ratings for Restaurants'] < 5.0)]
print('Zipcodes with Overall Star Ratings for Restaurants between 4.5 and 5.0:\n', lvzipcode_df[:10], '\n')
lvzipcode_df = lvzipcode_df[(lvzipcode_df['lat'] > 36.05) & (lvzipcode_df['lat'] < 36.15) & (lvzipcode_df['lon'] > -115.3) & (lvzipcode_df['lon'] < -115.0)]
print('Zipcodes with Overall Star Ratings for Restaurants between 4.5 and 5.0 and Latitude and Longitude Filtered:\n', lvzipcode_df[:10], '\n')
```

```
Average of overall star ratings in each zipcode:
 count    1000.000000
mean        3.500000
std         0.500000
min         1.000000
25%         3.000000
50%         3.500000
75%         4.000000
max         5.000000
Name: stars, dtype: float64

Top 10 Zipcodes for Overall Star Ratings for Restaurants:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  89101.0  36.166667 -115.150                                  5.00
1  89102.0  36.166667 -115.150                                  5.00
2  89103.0  36.166667 -115.150                                  5.00
3  89104.0  36.166667 -115.150                                  5.00
4  89105.0  36.166667 -115.150                                  5.00
5  89106.0  36.166667 -115.150                                  5.00
6  89107.0  36.166667 -115.150                                  5.00
7  89108.0  36.166667 -115.150                                  5.00
8  89109.0  36.166667 -115.150                                  5.00
9  89110.0  36.166667 -115.150                                  5.00

Zipcodes with Overall Star Ratings for Restaurants between 3.5 and 4.5:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  89101.0  36.166667 -115.150                                  5.00
1  89102.0  36.166667 -115.150                                  5.00
2  89103.0  36.166667 -115.150                                  5.00
3  89104.0  36.166667 -115.150                                  5.00
4  89105.0  36.166667 -115.150                                  5.00
5  89106.0  36.166667 -115.150                                  5.00
6  89107.0  36.166667 -115.150                                  5.00
7  89108.0  36.166667 -115.150                                  5.00
8  89109.0  36.166667 -115.150                                  5.00
9  89110.0  36.166667 -115.150                                  5.00

Zipcodes with Overall Star Ratings for Restaurants between 3.5 and 4.5 and Latitude and Longitude Filtered:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  89101.0  36.166667 -115.150                                  5.00
1  89102.0  36.166667 -115.150                                  5.00
2  89103.0  36.166667 -115.150                                  5.00
3  89104.0  36.166667 -115.150                                  5.00
4  89105.0  36.166667 -115.150                                  5.00
5  89106.0  36.166667 -115.150                                  5.00
6  89107.0  36.166667 -115.150                                  5.00
7  89108.0  36.166667 -115.150                                  5.00
8  89109.0  36.166667 -115.150                                  5.00
9  89110.0  36.166667 -115.150                                  5.00

Zipcodes with Overall Star Ratings for Restaurants between 3.75 and 4.25:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  89101.0  36.166667 -115.150                                  5.00
1  89102.0  36.166667 -115.150                                  5.00
2  89103.0  36.166667 -115.150                                  5.00
3  89104.0  36.166667 -115.150                                  5.00
4  89105.0  36.166667 -115.150                                  5.00
5  89106.0  36.166667 -115.150                                  5.00
6  89107.0  36.166667 -115.150                                  5.00
7  89108.0  36.166667 -115.150                                  5.00
8  89109.0  36.166667 -115.150                                  5.00
9  89110.0  36.166667 -115.150                                  5.00

Zipcodes with Overall Star Ratings for Restaurants between 3.75 and 4.25 and Latitude and Longitude Filtered:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  89101.0  36.166667 -115.150                                  5.00
1  89102.0  36.166667 -115.150                                  5.00
2  89103.0  36.166667 -115.150                                  5.00
3  89104.0  36.166667 -115.150                                  5.00
4  89105.0  36.166667 -115.150                                  5.00
5  89106.0  36.166667 -115.150                                  5.00
6  89107.0  36.166667 -115.150                                  5.00
7  89108.0  36.166667 -115.150                                  5.00
8  89109.0  36.166667 -115.150                                  5.00
9  89110.0  36.166667 -115.150                                  5.00

Zipcodes with Overall Star Ratings for Restaurants between 4.0 and 4.5:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  89101.0  36.166667 -115.150                                  5.00
1  89102.0  36.166667 -115.150                                  5.00
2  89103.0  36.166667 -115.150                                  5.00
3  89104.0  36.166667 -115.150                                  5.00
4  89105.0  36.166667 -115.150                                  5.00
5  89106.0  36.166667 -115.150                                  5.00
6  89107.0  36.166667 -115.150                                  5.00
7  89108.0  36.166667 -115.150                                  5.00
8  89109.0  36.166667 -115.150                                  5.00
9  89110.0  36.166667 -115.150                                  5.00

Zipcodes with Overall Star Ratings for Restaurants between 4.0 and 4.5 and Latitude and Longitude Filtered:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  89101.0  36.166667 -115.150                                  5.00
1  89102.0  36.166667 -115.150                                  5.00
2  89103.0  36.166667 -115.150                                  5.00
3  89104.0  36.166667 -115.150                                  5.00
4  89105.0  36.166667 -115.150                                  5.00
5  89106.0  36.166667 -115.150                                  5.00
6  89107.0  36.166667 -115.150                                  5.00
7  89108.0  36.166667 -115.150                                  5.00
8  89109.0  36.166667 -115.150                                  5.00
9  89110.0  36.166667 -115.150                                  5.00

Zipcodes with Overall Star Ratings for Restaurants between 4.25 and 4.75:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  89101.0  36.166667 -115.150                                  5.00
1  89102.0  36.166667 -115.150                                  5.00
2  89103.0  36.166667 -115.150                                  5.00
3  89104.0  36.166667 -115.150                                  5.00
4  89105.0  36.166667 -115.150                                  5.00
5  89106.0  36.166667 -115.150                                  5.00
6  89107.0  36.166667 -115.150                                  5.00
7  89108.0  36.166667 -115.150                                  5.00
8  89109.0  36.166667 -115.150                                  5.00
9  89110.0  36.166667 -115.150                                  5.00

Zipcodes with Overall Star Ratings for Restaurants between 4.25 and 4.75 and Latitude and Longitude Filtered:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  89101.0  36.166667 -115.150                                  5.00
1  89102.0  36.166667 -115.150                                  5.00
2  89103.0  36.166667 -115.150                                  5.00
3  89104.0  36.166667 -115.150                                  5.00
4  89105.0  36.166667 -115.150                                  5.00
5  89106.0  36.166667 -115.150                                  5.00
6  89107.0  36.166667 -115.150                                  5.00
7  89108.0  36.166667 -115.150                                  5.00
8  89109.0  36.166667 -115.150                                  5.00
9  89110.0  36.166667 -115.150                                  5.00

Zipcodes with Overall Star Ratings for Restaurants between 4.5 and 5.0:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  89101.0  36.166667 -115.150                                  5.00
1  89102.0  36.166667 -115.150                                  5.00
2  89103.0  36.166667 -115.150                                  5.00
3  89104.0  36.166667 -115.150                                  5.00
4  89105.0  36.166667 -115.150                                  5.00
5  89106.0  36.166667 -115.150                                  5.00
6  89107.0  36.166667 -115.150                                  5.00
7  89108.0  36.166667 -115.150                                  5.00
8  89109.0  36.166667 -115.150                                  5.00
9  89110.0  36.166667 -115.150                                  5.00

Zipcodes with Overall Star Ratings for Restaurants between 4.5 and 5.0 and Latitude and Longitude Filtered:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  89101.0  36.166667 -115.150                                  5.00
1  89102.0  36.166667 -115.150                                  5.00
2  89103.0  36.166667 -115.150                                  5.00
3  89104.0  36.166667 -115.150                                  5.00
4  89105.0  36.166667 -115.150                                  5.00
5  89106.0  36.166667 -115.150                                  5.00
6  89107.0  36.166667 -115.150                                  5.00
7  89108.0  36.166667 -115.150                                  5.00
8  89109.0  36.166667 -115.150                                  5.00
9  89110.0  36.166667 -115.150                                  5.00
```





The map above shows the average of the above variables in the Las Vegas city center in red. The darker the color, the closer the value is to 5. The top and middle figures are similarly distributed, indicating that the average of the star ratings by users and the average of the ratings by influencers tend to be similar across zipcodes. However, the bottom figure shows that the average of the star ratings by influencers is more evenly distributed across zipcodes, with a higher average of influencers' star ratings, indicating that influencers' ratings are concentrated in the central area of Las Vegas. Compared to the top and bottom figures, the middle figure shows that the average of the star ratings by users is higher in the outer areas of Las Vegas. Compared to the top and bottom figures, the average significantly drops in Summerlin, an upscale area.

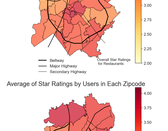
```
# Overall star ratings for restaurants in Charlotte
clrestaurants_df = df[df['Restaurants'].isin(['Yes'])].groupby(['zipcode', 'lat', 'lon'], as_index=False).agg({'stars':'mean'})
clrestaurants_df = clrestaurants_df[(clrestaurants_df['lat'] > 35.0) & (clrestaurants_df['lat'] < 35.4) & (clrestaurants_df['lon'] > -81.0) & (clrestaurants_df['lon'] < -80.6)]
clrestaurants_df['stars'] = clrestaurants_df['stars'].round(2)
print('Overall star ratings for restaurants in Charlotte:\n', clrestaurants_df['stars'].describe(), '\n')
clrestaurants_df.rename(columns={'stars':'Overall Star Ratings for Restaurants'}, inplace=True)
clrestaurants_df.sort_values(by=['Overall Star Ratings for Restaurants'], inplace=True, ascending=False)
clrestaurants_df['Overall Star Ratings for Restaurants'] = clrestaurants_df['Overall Star Ratings for Restaurants'].round(2)
print('Top 10 Zipcodes for Overall Star Ratings for Restaurants in Charlotte:\n', clrestaurants_df[:10], '\n')
```

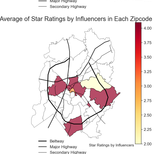
```
Overall star ratings for restaurants in Charlotte:
 count    1000.000000
mean        3.500000
std         0.500000
min         1.000000
25%         3.000000
50%         3.500000
75%         4.000000
max         5.000000
Name: stars, dtype: float64

Top 10 Zipcodes for Overall Star Ratings for Restaurants in Charlotte:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  28105.0  35.166667 -80.833                                  5.00
1  28106.0  35.166667 -80.833                                  5.00
2  28107.0  35.166667 -80.833                                  5.00
3  28108.0  35.166667 -80.833                                  5.00
4  28109.0  35.166667 -80.833                                  5.00
5  28110.0  35.166667 -80.833                                  5.00
6  28111.0  35.166667 -80.833                                  5.00
7  28112.0  35.166667 -80.833                                  5.00
8  28113.0  35.166667 -80.833                                  5.00
9  28114.0  35.166667 -80.833                                  5.00
```

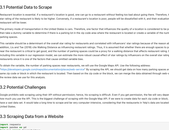
Restaurants in Charlotte are fairly evenly distributed. Thus, many restaurants are in the central area of the city center, and there seem to be few outliers.

```
# Average of overall star ratings in each zipcode
clzipcode_df = df[df['Restaurants'].isin(['Yes'])].groupby(['zipcode', 'lat', 'lon'], as_index=False).agg({'stars':'mean'})
clzipcode_df = clzipcode_df[(clzipcode_df['lat'] > 35.0) & (clzipcode_df['lat'] < 35.4) & (clzipcode_df['lon'] > -81.0) & (clzipcode_df['lon'] < -80.6)]
clzipcode_df['stars'] = clzipcode_df['stars'].round(2)
print('Average of overall star ratings in each zipcode:\n', clzipcode_df['stars'].describe(), '\n')
clzipcode_df.rename(columns={'stars':'Overall Star Ratings for Restaurants'}, inplace=True)
clzipcode_df.sort_values(by=['Overall Star Ratings for Restaurants'], inplace=True, ascending=False)
clzipcode_df['Overall Star Ratings for Restaurants'] = clzipcode_df['Overall Star Ratings for Restaurants'].round(2)
print('Top 10 Zipcodes for Overall Star Ratings for Restaurants:\n', clzipcode_df[:10], '\n')
```

```
Average of overall star ratings in each zipcode:
 count    1000.000000
mean        3.500000
std         0.500000
min         1.000000
25%         3.000000
50%         3.500000
75%         4.000000
max         5.000000
Name: stars, dtype: float64

Top 10 Zipcodes for Overall Star Ratings for Restaurants:
    zipcode       lat       lon  Overall Star Ratings for Restaurants
0  28105.0  35.166667 -80.833                                  5.00
1  28106.0  35.166667 -80.833                                  5.00
2  28107.0  35.166667 -80.833                                  5.00
3  28108.0  35.166667 -80.833                                  5.00
4  28109.0  35.166667 -80.833                                  5.00
5  28110.0  35.166667 -80.833                                  5.00
6  28111.0  35.166667 -80.833                                  5.00
7  28112.0  35.166667 -80.833                                  5.00
8  28113.0  35.166667 -80.833                                  5.00
9  28114.0  35.166667 -80.833                                  5.00
```







In the case of Charlotte, we see that a strong positive relationship between the restaurant's overall star rating and the star rating by users. The trends in the middle and in the top are similar, but the middle figure shows that the average star rating by influencers is 3 and lower than others. It can be seen that where the city is darker, the average is higher, the color tends to be darker in the map showing the average overall star rating of the restaurant. Also, where the average star rating by influencers is higher, the average star rating by users is also higher. Thus, we can infer that there is a strong positive relationship between user ratings and overall restaurant ratings in Las Vegas and Charlotte.

3. Project Three

3.1 Potential Data to Scrape

Restaurant location is essential to a restaurant's success. In general, we can go to a restaurant without feeling too bad about going there. Therefore, the overall star rating of the restaurant is likely to be higher. Conversely, if a restaurant's location is poor, people will be deterred from eat it, and their evaluation of the restaurant will be lower.

Therefore, we can infer that restaurant's location affects its star rating. Therefore, one factor that influences the quality of a restaurant is considered to be its parking. We find data is currently available to determine if there is a parking lot in the city code area where the restaurant is located or creates a website of the number of parking lots in each area.

The website should be a dashboard of the overall star rating for restaurants and controlled-for-other-influencers' star rating because of the reason above. In addition, it can be COVID-19-affecting Dashboard as influencing restaurant ratings. Thus, it is expected that whether there are enough spaces to park cars or not, people will be more cautious about going to restaurants with many people in the parking lot. Therefore, it is expected that the number of parking lots in the surrounding area will be a factor that controls for a star rating. However, we can't know the exact value's impact effect of a star rating by influencers on the overall star rating for restaurants. Thus, we need to create a dashboard to see the impact of each factor.

To obtain the variable, the number of parking spaces near restaurants, we will use the Google Maps API. Use the following address: https://developers.google.com/maps/documentation/places/web-service/search-nearby#PlaceSearchNearby. Thus, we can find the number of parking spots in the area by using the API. However, there is a limit to the number of requests per day. Thus, we need to find a way to get the data efficiently through other methods. Instead of using the API, we can use Selenium to scrape the Google Maps data as well.

3.2 Potential Challenges

Google prohibits web scraping using their API without permission. Hence, for example, it allows 40 hits. If we try to get permission, the limit will vary depending on how much you use the API. This is the biggest challenge of scraping Google Maps. If we want to scrape the data, we need to find a way to get around this. Instead, we can use Selenium to scrape data from Google Maps. However, there is a limit to how many times Google will let you computer interactions, complicating that the response to Hard Data are included and the overall Rating.

3.3 Scraping Data from a Website

```
import requests
from bs4 import BeautifulSoup
page = requests.get("https://www.zipcodestogo.com/Arizona/")
print("Response Code: ", page.status_code)
soup = BeautifulSoup(page.text, 'html.parser')
print(soup.prettify())
```

From the code above, the labels are located in the "table" tag, and there are several labels in the website. By using find_all() method, we are going to get the information of the first table and the second table separately.

```
table = soup.find_all('table')
print("Number of tables: ", len(table))
```

The object data_state2 should have the information of the second table in "tbody" tag. We are going to create another dataset that contains every row of the table.

```
data_state2 = table[1].tbody.text
```

Then it turns out the data is not clean enough. Thus, we need to do some cleaning to separate those records around different the overall string for restaurants, we are going to separate by the carriage return character and the tab and the pure ASCII code.

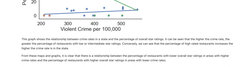
```
data_state2 = data_state2.replace('\t', '').replace('\r', '').replace('\n', '')
```

Then, we can see that the data is clean enough. Thus, we can now split the data by using the split() function.

```
data = data_state2.split('\x0a')
print("Length: ", len(data))
print("Data: ", data)
```

```
Response Code:  200
<!DOCTYPE html>
<html>
 <head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <meta name="description" content="">
  <meta name="keywords" content="">
  <meta name="robots" content="noindex,follow">
  <meta name="revisit-after" content="7 days">
  <meta name="generator" content="">
  <meta name="ZCTG-version"
```

```
general_state                                  object
zip                                            object
lat                                            float64
lng                                            float64
city                                           object
state_id                                       object
state_name                                     object
county_fips                                    int64
county_name                                    object
population                                     int64
density                                        int64
source                                         object
military                                       bool
incorporated                                   bool
timezone                                       object
ranking                                        int64
zcta                                           bool
tax_return_filed                               int64
tax_returns_2021                               int64
estimated_population                           int64
estimated_notes                                object
dtype: int64
<class'numpy.ndarray'>
(1.0,
 nan,
 0.0,
 0.0,
 0.0
```



This graph shows the relationship between video rates in a state and the number of violent crimes per 100,000 people. It can be seen that the higher the crime rate, the greater the percentage of restaurants with low or intermediate star ratings. Conversely, we can see that the percentage of high rated restaurants increases the lower the crime rate is in a state.

Although the data shows that there is a correlation between crime rate and restaurant ratings, it is important to remember that correlation does not imply causation. The relationship between crime rates and the percentage of restaurants with higher star ratings is areas with lower crime rates.

We could not when looking at the relationship between the number of Tax Preparer Businesses and the number of restaurants per parking space all the way to 0 (Tax Preparer Businesses = 0, Parking Spaces = 0). Therefore, we will merge that data with the data from the review.

```
[restaurant_parking = pd.merge(restaurant_business_merge, ...)]
[restaurant_parking.shape]
[restaurant_parking[['business_id', 'stars', 'name', 'review_count', 'zip', 'open', 'is_open', ...]].head()]
[restaurant_parking.to_csv('restaurant_parking.csv')]

business_parking = pd.merge(restaurant_business_merge[['zip', ...]], ...)
business_parking.shape
business_parking[['zip', 'business_id', 'Total Parking Spaces', 'Tax Preparer Businesses', ...]].head()
business_parking.to_csv('business_parking.csv')

restaurant_parking[restaurant_parking['Tax Preparer Businesses'].isnull()].shape
restaurant_parking[restaurant_parking['Tax Preparer Businesses'].isnull()].stars.value_counts()
restaurant_parking[restaurant_parking['Tax Preparer Businesses'].isnull()].stars.value_counts()/restaurant_parking[restaurant_parking['Tax Preparer Businesses'].isnull()].shape[0]
restaurant_parking[restaurant_parking['Tax Preparer Businesses'].isnull()].review_count.value_counts()
restaurant_parking[restaurant_parking['Tax Preparer Businesses'].isnull()].review_count.value_counts()/restaurant_parking[restaurant_parking['Tax Preparer Businesses'].isnull()].shape[0]
restaurant_parking[restaurant_parking['Tax Preparer Businesses'].isnull()].open.value_counts()
restaurant_parking[restaurant_parking['Tax Preparer Businesses'].isnull()].open.value_counts()/restaurant_parking[restaurant_parking['Tax Preparer Businesses'].isnull()].shape[0]
restaurant_parking[restaurant_parking['Tax Preparer Businesses'].isnull()].is_open.value_counts()
restaurant_parking[restaurant_parking['Tax Preparer Businesses'].isnull()].is_open.value_counts()/restaurant_parking[restaurant_parking['Tax Preparer Businesses'].isnull()].shape[0]
restaurant_parking[restaurant_parking['Tax Preparer Businesses'].isnull()].Total Parking Spaces.value_counts()
restaurant_parking[restaurant_parking['Tax Preparer Businesses'].isnull()].Total Parking Spaces.value_counts()/restaurant_parking[restaurant_parking['Tax Preparer Businesses'].isnull()].shape[0]
restaurant_parking[restaurant_parking['Tax Preparer Businesses'].isnull()].Tax Preparer Businesses.value_counts()
restaurant_parking[restaurant_parking['Tax Preparer Businesses'].isnull()].Tax Preparer Businesses.value_counts()/restaurant_parking[restaurant_parking['Tax Preparer Businesses'].isnull()].shape[0]
```

In addition, we analyzed the text in the review data, using the stopwords method in Python to eliminate frequently occurring meaningless words and to determine which words appeared most frequently in the text of each star's reviews.

```
[import nltk
from nltk.corpus import stopwords
[stopwords = set(stopwords.words('English'))]
[[stopwords.update(extra_stopwords) for ...]]
[[stopwords.update(['', 'br', 'br', 'br
```