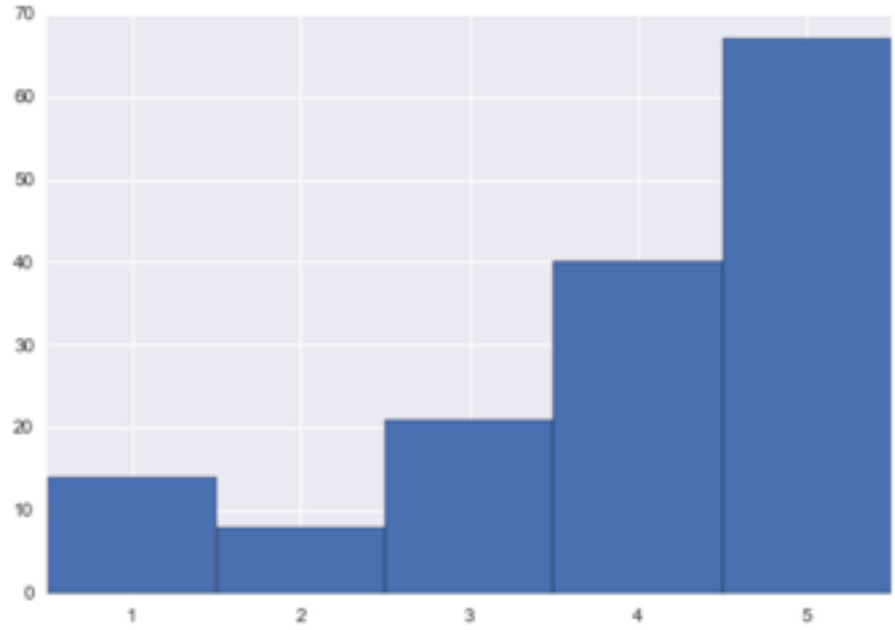
Navraj Narula
CS591 K1
Homework 2

Analysis and Visualizations based on Code Output

Problem # 1 - Sentiment Analysis

```
In [6]: draw_viz(info)
```



After parsing the json file containing Yelp reviews as well as reading in postive and negative words, I compared the text of Yelp reviews against positive and negative words. This simple algorithm was used to calculate the score of each text block:

positive word = +1
negative word = -1
neutral word = +0 (or do nothing)

These scores were saved in a tuple along with the the star rating given by the user. The scores-stars tuple look something like:

```
In [7]: runfile('/Users/navienarula/Desktop/hw2datasci/hw.py',
wdir='/Users/navienarula/Desktop/hw2datasci')
[(4, 0.7984031936127745), (5, 0.5988023952095809), (5,
0.8739076154806492), (5, 0.7364544976328248), (3, 0.9980039920159681),
(1, 0.8455034588777863), (4, 0.6659267480577137), (5,
0.9950248756218907), (5, 0.9991673605328892), (3, 0.5828476269775188),
(1, 0.5226084980686208), (1, 0.5291005291005291), (4,
0.7984031936127745), (4, 0.5380476556495004), (5, 0.6357856494096277),
(2, 0.9975062344139651), (4, 0.6644518272425249), (1,
0.49751243781094534), (1, 0.5449591280653951), (5, 0.5787950539331755),
(5, 0.49751243781094534), (5, 0.6466784244562022), (4,
0.9980039920159681), (5, 0.9983361064891847), (5, 0.8559201141226819),
(5, 0.7992007992007992), (4, 0.5706134094151213), (5,
0.8122461730709154), (4, 0.9987515605493134), (5, 0.5860048259220958)]
```

Although discrepancy in terms of the words people used and the stars rating they gave exist, it is more often the case that their words “match” their scores. A higher score indicates a higher positivity rate, and vice versa in terms of a negative score. It is interesting, though, that a score of “1” contains more positive words than a score of “2.”

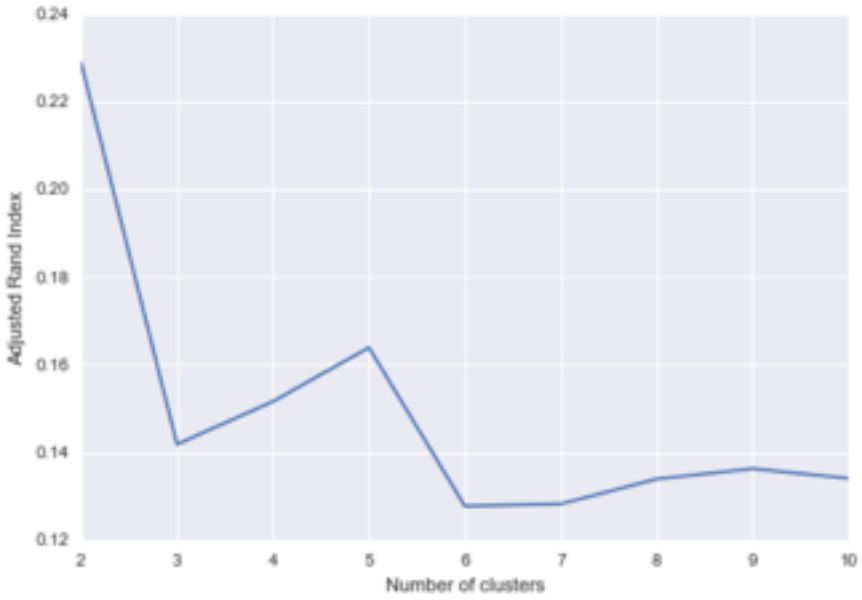
Problem # 2 - Clusters

After limiting the data to only Las Vegas food places and only taking note of categories, attributes, latitude, and longitude, I decided to visualize the data using k-means ++ clustering. In order to do so, I first needed to determine the right k value to use.

I did this by referring **sc_evaluate_clusters,** a function found in the lecture notes.

```
restaurants = get_info(yelp)

vec = DictVectorizer()

R = vec.fit_transform(restaurants).toarray()

features = vec.get_feature_names()

sc_evaluate_clusters(R,10)
```

After passing in max_cluster of 10, I retrieved this graph:



Since the high peak is at 5, I will use a value of 5 as my k.

```
R = vec.fit_transform(restaurants).toarray()

features = vec.get_feature_names()

k = 5

k_means(features, R, k)
```

This is the visualization that resulted from running my k_means function.

Here are the specific clusters representing the restaurants, by color:

Top terms per cluster:

Cluster 0: (BLUE)
Good for Kids

Attire=casual

Alcohol=none

Take-out

Accepts Credit Cards

lunch

Caters

Good For Groups

Delivery

Fast Food

Cluster 1: (GREEN)
Sushi Bars

Japanese

Alcohol=beer_and_wine

Takes Reservations

Ramen

Waiter Service

dinner

Korean

Price Range

vegetarian

Cluster 2: (RED)
Nightlife

Bars

Happy Hour

Alcohol=full_bar

Smoking=yes

Price Range

Smoking=no

valet

Attire=dressy

Sports Bars

Cluster 3: (CYAN)
Convenience Stores

Gas & Service Stations

Automotive

Delis

Food

Alcohol=full_bar

longitude

Wi-Fi=no

Good For Groups

Attire=casual

Cluster 4: (MAGENTA)
Hotels

Hotels & Travel

Casinos

Event Planning & Services

Attire=formal

Shaved Ice

Arts & Entertainment

Food Stands

longitude

Food Trucks

I notice that each outlier has the potential for being an outlier and that most groups are clustered towards the middle latitude and longitude, which is downtown — this lies correctly with the assumption that most restaurants in Las Vegas would be at that location.