Yelp Case Study

By: Steven Wang

Data Cleaning

In [1]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
# print all the outputs in a cell
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"

import warnings
warnings.filterwarnings("ignore")

pd.set_option("display.max_rows", None)
```

In [3]:
```
df = pd.read_csv('yelp_data.csv',encoding='latin-1')
```

In [4]:
```
df.head()
```

Out[4]:

	Business - Monday Close	Business - Saturday Close	Business - Friday Open	Business - Saturday Open	Business - Monday Open	Business - Sunday Close	Business - Sunday Open	Business - Wednesday Close	Business - Thursday Close	Business - Wednesday Open	...	Review - Stars	Business - State	User - Id	Use Vote Coc
0	21:00	21:30	11:00	11:00	11:00	21:00	11:00	21:00	21:00	11:00	...	4	AZ	gKhJyiCKkG6dHPNhr3dhDQ	
1	21:00	22:30	11:00	11:00	11:00	21:30	11:00	21:00	21:00	11:00	...	3	AZ	kJc9YBRwmmZ_PG0uLHuEPQ	10
2	21:00	22:00	11:00	11:00	11:00	NaN	NaN	22:00	22:00	11:00	...	5	AZ	K4FAia2ly5MVnmBLfS-mCg	8
3	22:00	23:00	11:00	11:00	11:00	22:00	16:30	22:00	22:00	11:00	...	5	AZ	6VZNGc2h2Bn-uyuEXgOt5g	11
4	21:00	22:00	11:00	16:00	16:30	20:00	11:00	14:30	14:30	11:00	...	4	AZ	K4FAia2ly5MVnmBLfS-mCg	8

5 rows × 90 columns

In [5]:
```
#Drop Business hours & other columns as has many NA values
df.drop(['Business - Monday Close', 'Business - Saturday Close',
       'Business - Friday Open', 'Business - Saturday Open',
       'Business - Monday Open', 'Business - Sunday Close',
       'Business - Sunday Open', 'Business - Wednesday Close',
       'Business - Thursday Close', 'Business - Wednesday Open',
       'Business - Thursday Open', 'Business - Tuesday Close',
       'Business - Tuesday Open', 'Business - Restaurant?', 'Business - Friday Close', 'Number of Records',
       'Business - Accepts Credit Cards', 'Business - Accepts Insurance',
       'Business - Ages Allowed'], axis=1, inplace=True)
```

In [6]:
```
df.isna().sum()
```

Out[6]:
```
User - Years Elite                215807
Business - Alcohol                  5129
Business - Attire                   1742
Business - BYOB/Corkage           216241
Business - BYOB                   237958
Business - By Appointment Only    284729
Business - Caters                  27283
Business - Coat Check             237290
Business - Corkage                248491
Business - Delivery                 2717
Business - Dietary Restrictions   279361
Business - Dogs Allowed           230558
Business - Drive-Thru             260373
Business - Good For Dancing       237935
Business - Good For Groups          1726
Business - Good For Kids          203178
Business - Good for Kids            1675
Business - Happy Hour             235319
Business - Has TV                   3853
```

```
Business - Noise Level               5387
Business - Open 24 Hours           266638
Business - Order at Counter        262028
Business - Outdoor Seating           2337
Business - Parking                   3724
Business - Payment Types           283150
Business - Price Range               1310
Business - Smoking                 237193
Business - Take-out                  1865
Business - Takes Reservations        2155
Business - Waiter Service            3989
Business - Wheelchair Accessible    52463
Business - Wi-Fi                    21566
User - Average Stars                    0
Business - Id                           0
Business - Categories                   0
Business - City                         0
User - Compliments Cool            156912
User - Compliments Cute            228773
User - Compliments Funny           187831
User - Compliments Hot             178814
User - Compliments List            249352
User - Compliments More            203876
User - Compliments Note            147938
User - Compliments Photos          222710
User - Compliments Plain           149760
User - Compliments Profile         242737
User - Compliments Writer          175179
Review - Date                           0
User - Fans                             0
Business - Address                      0
City and State                          0
Latitude                                0
Longitude                               0
User - Name                             1
Business - Name                         0
Business - Neighborhoods                0
Business - Open?                        0
User - Review Count                     0
Business - Review Count                 0
Review - Id                             0
Business - Stars                        0
Review - Stars                          0
Business - State                        0
User - Id                               0
User - Votes Cool                       0
Review - Votes Cool                     0
User - Votes Funny                      0
Review - Votes Funny                    0
User - Votes Useful                     0
Review - Votes Useful                   0
User - Yelping Since                    0
dtype: int64
```

In [7]:
```
#Drop Columns with a lot of NA Values
df.drop(['User - Years Elite','Business - BYOB/Corkage', 'Business - BYOB',
       'Business - By Appointment Only','Business - Caters','Business - Coat Check', 'Business - Corkage', 'Business - Dietary Restriction
       'Business - Dogs Allowed','Business - Drive-Thru', 'Business - Good For Dancing','Business - Good For Kids',
       'Business - Happy Hour', 'Business - Open 24 Hours', 'Business - Order at Counter',
       'Business - Payment Types','Business - Smoking','Business - Wheelchair Accessible', 'Business - Wi-Fi',
       'User - Compliments Cool', 'User - Compliments Cute','User - Compliments Funny',
       'User - Compliments Hot', 'User - Compliments List', 'User - Compliments More',
       'User - Compliments Note', 'User - Compliments Photos', 'User - Compliments Plain',
       'User - Compliments Profile','User - Compliments Writer'],
        axis=1, inplace=True)
```

In [8]:
```
#Fill rest NA values No Response
df.fillna("No Response",inplace=True)
df.isna().sum().sum()
```

Out[8]:
```
0
```

In [9]:
```
df.duplicated(subset=None, keep='first').any()
```

Out[9]:
```
False
```

In [10]:
```
len(df)
```

Out[10]:
```
285276
```

In [11]:
```
#Unable to sort restaurant type due to large number of categories
df['Business - Categories'].nunique()
#df.groupby('Business - Categories')['Review - Id'].count().sample(50)
```

Out[11]:
```
1030
```

In [12]:
```
#Summary of the Most reviewed Restaurants
df.groupby('Business - Name')['Business - Name'].count().nlargest(15)
```

Out[12]:
```
Business - Name
Pita Jungle                  2114
Oregano's Pizza Bistro       1696
Cornish Pasty Company        1404
Pizzeria Bianco              1372
Lo-Lo's Chicken & Waffles    1187
Four Peaks Brewing Co        1040
Chipotle Mexican Grill       1034
FEZ                          1020
True Food Kitchen             977
Cibo                          964
Chino Bandido                 908
Postino Arcadia               862
In-N-Out Burger               846
Chompie's Deli                843
America's Taco Shop           819
Name: Business - Name, dtype: int64
```

Finding 1

Observe general information on year of review, average review score, restaurant score, user score, etc.

Then, find if there is any correlation between any of these variables

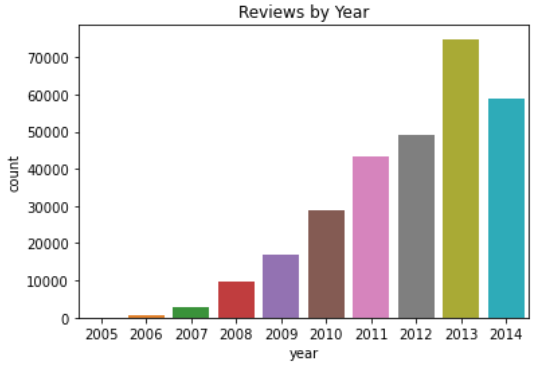
In [13]:
```
import seaborn as sns
import pandas as pd
import numpy as np
%pylab inline
```

```
Populating the interactive namespace from numpy and matplotlib
```

In [14]:
```
df['year'] = pd.to_datetime(df['Review - Date']).dt.year
```

In [15]:
```
sns.countplot(x='year', data=df).set(title='Reviews by Year')
```

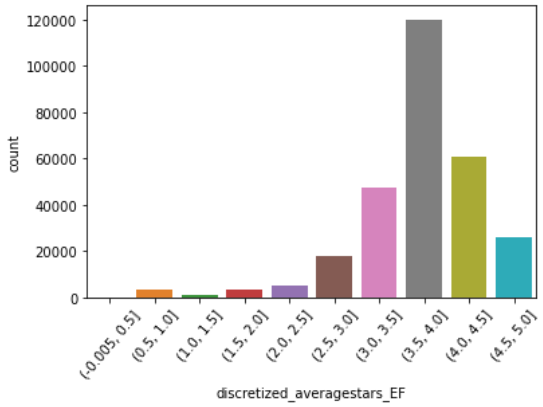
Out[15]:
```
[Text(0.5, 1.0, 'Reviews by Year')]
```



In [16]:
```
df['discretized_averagestars_EF']=pd.cut(df['User - Average Stars'], 10)
```

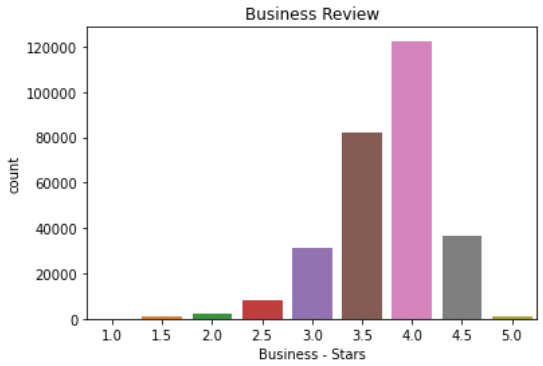
Average Reviews given by Reviewer (Range)

In [17]:
```
sns.countplot(x='discretized_averagestars_EF', data=df)\
        .tick_params(axis='x', rotation=50)
```



In [18]:
```
sns.countplot(x='Business - Stars', data=df).set(title='Business Review')
```

Out[18]:
```
[Text(0.5, 1.0, 'Business Review')]
```



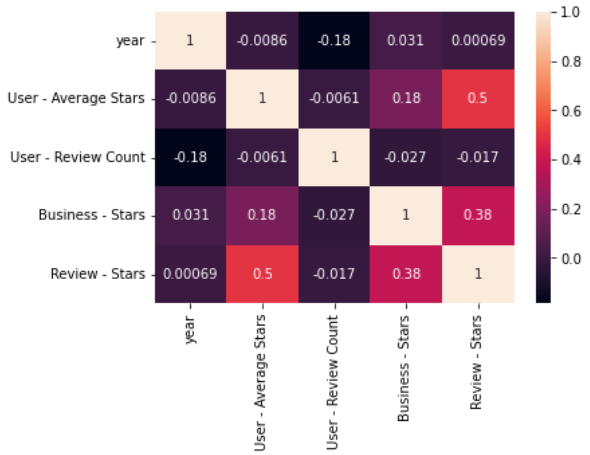
In [19]:
```
sns.countplot(x='Review - Stars', data=df).set(title='Review by Yelper')
```

Out[19]:
```
[Text(0.5, 1.0, 'Review by Yelper')]
```

In [20]:
```
df_corr = pd.DataFrame(df,columns=['year','User - Average Stars','User - Review Count',\
                                   'Business - Stars','Review - Stars','Business - City'])

corrMatrix = df_corr.corr()
sns.heatmap(corrMatrix, annot=True)
plt.show()
```

Out[20]:
```
<AxesSubplot:>
```



From the figure above, we can see that there is some correlation between User average score and the Review he/she gives. Also there is some between a Business overall rating and the Review Score it is given.

It is important to note due to the large volume & variety of data, it is difficult to perform correlation with more variables.

Finding 2

Observe based on the number of reviews by rating to see if/and how many people found the reviews useful

In [21]:
```
df_usefulreview = df.groupby('Review - Stars').agg({'Review - Stars':'count',\
            'Review - Votes Useful': 'sum'})\
            .rename(columns={'Review - Stars':'Number of Reviews',\
                    'Review - Votes Useful':'Number Review - Votes Useful'})
df_usefulreview.reset_index()
```

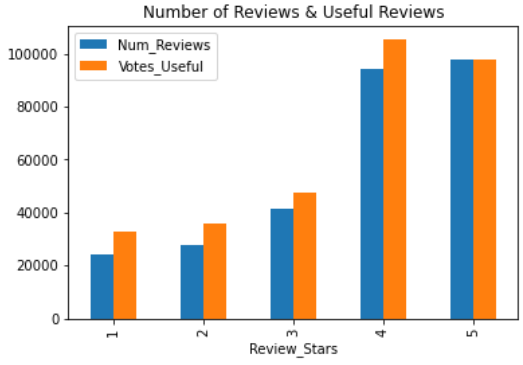
Out[21]:

	Review - Stars	Number of Reviews	Number Review - Votes Useful
0	1	24156	32961
1	2	27497	36020
2	3	41629	47302
3	4	94135	105247
4	5	97859	97732

In [22]:
```
df_usefulreview_graph = pd.DataFrame([['1', 24156, 32961], ['2', 27497, 36020],
                  ['3', 41629, 47302], ['4', 94135, 105247],
                  ['5', 97859, 97732]],
                 columns=['Review_Stars', 'Num_Reviews', 'Votes_Useful'])

fig, ax = plt.subplots()
df_usefulreview_graph.plot.bar(x='Review_Stars', ax=ax).set(title='Number of Reviews & Useful Reviews')
```

Out[22]:
```
[Text(0.5, 1.0, 'Number of Reviews & Useful Reviews')]
```



We can perform similar analysis with other criteria (such as if other users found reviews funny/cool/sad, etc.). Usefulness is likely the most focused criteria.

Finding 3

Plot where the restaurants are located. See if there are any hotspot restaurants for this dataset.

In [23]:
```
# import libraries
import geopandas as gpd
from shapely.geometry import Point, Polygon
import matplotlib.pyplot as plt
```

In [24]:
```
#Download a SHP file to map Arizona. Note: Was difficult for me to find a public one with labels including cities and boundries
street_map = gpd.read_file('azcounties.shp')
```

In [25]:
```
crs = {'init': 'epsg:4326'}
# x & y coordinates into single feature
geometry = [Point(xy) for xy in zip(df['Longitude'], df['Latitude'])]
# create Geopandas dataframe
geo_df = gpd.GeoDataFrame(df,
 crs = crs,
 geometry = geometry)
```

In [26]:
```
fig, ax = plt.subplots(figsize=(15,15))
# add .shp mapfile to axes
street_map.plot(ax=ax, alpha=0.4,color='grey')

# add geodataframe to axes
# make datapoints transparent using alpha
# assign size of points using markersize
geo_df.plot(column='Review - Stars',ax=ax,alpha=0.5, legend=True,markersize=10, cmap='rainbow')


plt.title('Review Restaurants Arizona', fontsize=15,fontweight='bold')

# set latitiude and longitude boundaries for map display
plt.xlim(-108.9,-115)
plt.ylim(31,38)

plt.show()
```

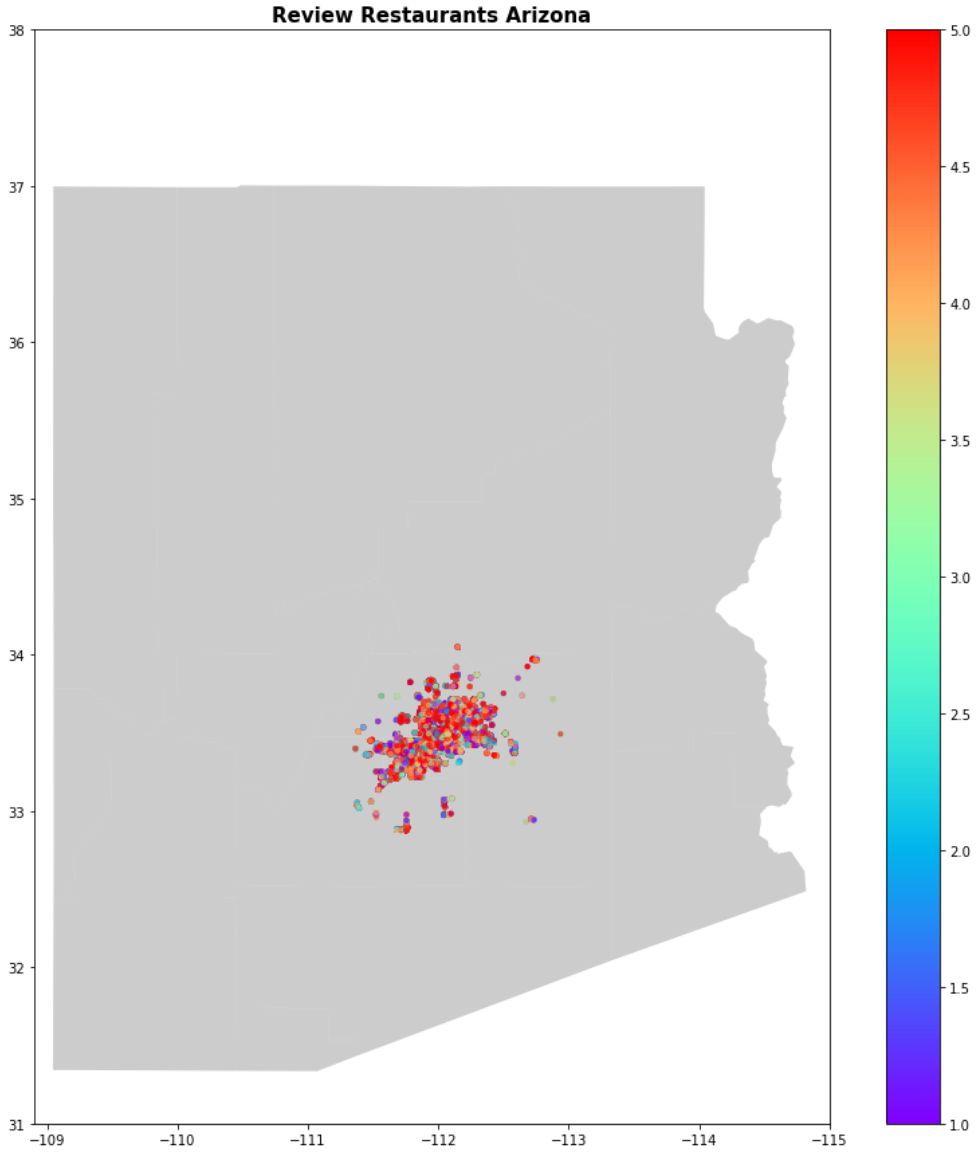
Out[26]:
```
<AxesSubplot:>
```

Out[26]:
```
<AxesSubplot:>
```

Out[26]:
```
Text(0.5, 1.0, 'Review Restaurants Arizona')
```

Out[26]:
```
(-108.9, -115.0)
```

Out[26]:
```
(31.0, 38.0)
```



Zoomed in GeoPlot

In [27]:

```
fig, ax = plt.subplots(figsize=(15,15))

street_map.plot(ax=ax, alpha=0.4,color='grey')

geo_df.plot(column='Review - Stars',ax=ax,alpha=0.5, legend=True,markersize=10,cmap='rainbow')

plt.title('Review Restaurants Arizona (Zoomed)', fontsize=15,fontweight='bold')

plt.xlim(-111.25,-113)
plt.ylim(32.75,34.15)

plt.show()
```

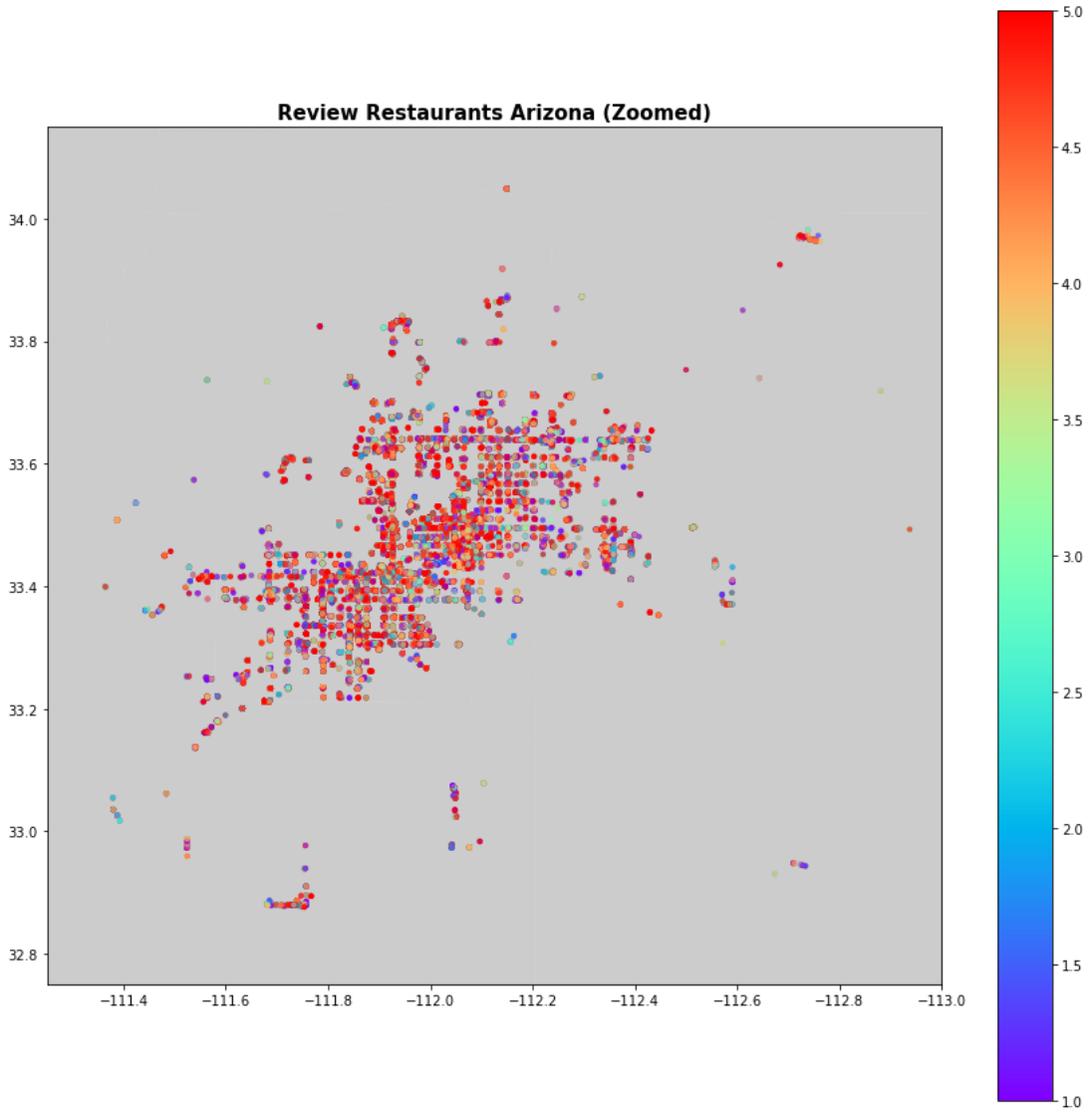
Out[27]: <AxesSubplot:>

Out[27]: <AxesSubplot:>

Out[27]: Text(0.5, 1.0, 'Review Restaurants Arizona (Zoomed)')

Out[27]: (-111.25, -113.0)

Out[27]: (32.75, 34.15)



Summary of number of reviews & score of restaurants by city

In [28]:
```
df.groupby('Business - City').agg({'Review - Id':'count',\
            'Review - Stars': 'mean'})\
            .rename(columns={'Review - Id':'Number of Reviews',\
                    'Review - Stars':'Avg Restaurant Scores'})\
            .sort_values(['Number of Reviews','Avg Restaurant Scores'], ascending=[False, False])[:20]
```

Out[28]:

Business - City	Number of Reviews	Avg Restaurant Scores
Phoenix	107856	3.806937
Scottsdale	57441	3.799795
Tempe	31296	3.692549
Chandler	20194	3.685104
Mesa	17785	3.738656
Gilbert	12592	3.775810
Glendale	11542	3.604401
Peoria	5345	3.554724
Surprise	3308	3.474002
Goodyear	2905	3.485370
Avondale	2238	3.720286
Cave Creek	2213	3.762765
Queen Creek	1864	3.657189
Paradise Valley	1054	3.833017

Business - City	Number of Reviews	Avg Restaurant Scores
Fountain Hills	874	3.782609
Casa Grande	736	3.599185
Litchfield Park	691	3.464544
Apache Junction	564	3.570922
Anthem	542	3.898524
Maricopa	504	3.309524

Based on the geoplot, we observe most plot points hover between scores of 4 - 5. There is a sizeable amount of scores that fall below 4 as it also brings the average down. The numerical summary is illustrated in the Summary above.