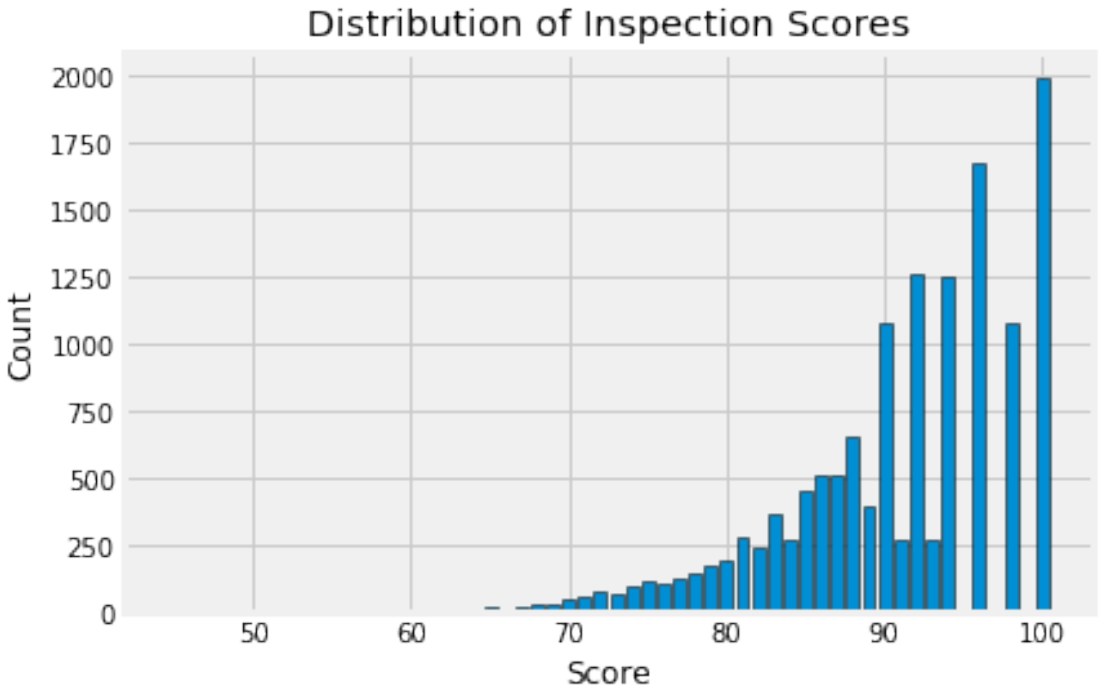
Notebook

February 24, 2020

0.0.1 Question 1a

Let's look at the distribution of inspection scores. As we saw before when we called head on this data frame, inspection scores appear to be integer values. The discreteness of this variable means that we can use a barplot to visualize the distribution of the inspection score. Make a bar plot of the counts of the number of inspections receiving each score.

It should look like the image below. It does not need to look exactly the same (e.g., no grid), but make sure that all labels and axes are correct.

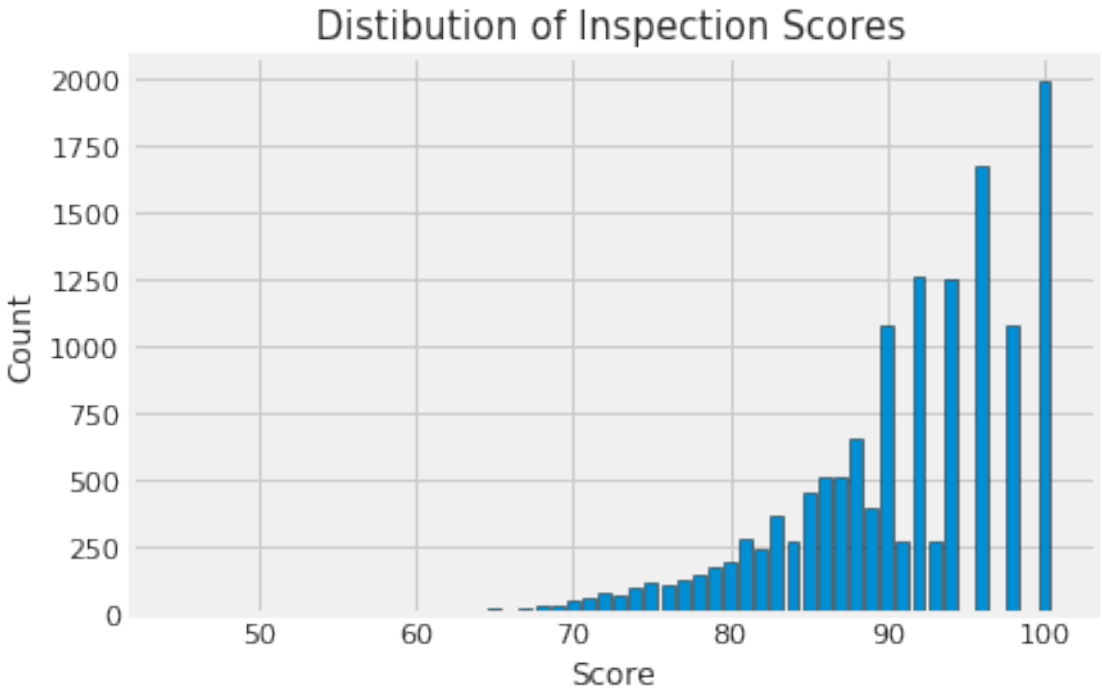


You might find this matplotlib.pyplot tutorial useful. Key syntax that you'll need:

```
plt.bar
plt.xlabel
plt.ylabel
plt.title
```

Note: If you want to use another plotting library for your plots (e.g. plotly, sns) you are welcome to use that library instead so long as it works on DataHub. If you use seaborn sns.countplot(), you may need to manually set what to display on xticks.

```
In [4]: %matplotlib inline
        ins_q1a = ins[~(ins['score'] == -1)]
        ins_q1a = ins_q1a[["iid","score"]].groupby('score').count()
        plt.bar(ins_q1a.index, ins_q1a["iid"], edgecolor='k')
        plt.xlabel("Score", fontsize=12)
        plt.ylabel("Count", fontsize=12)
        plt.title("Distibution of Inspection Scores", fontsize=14.5);
```


Distibution of Inspection Scores
Count
2000
1750
1500
1250
1000
750
500
250
0
50
60
70
80
90
100
Score

0.0.2 Question 1b

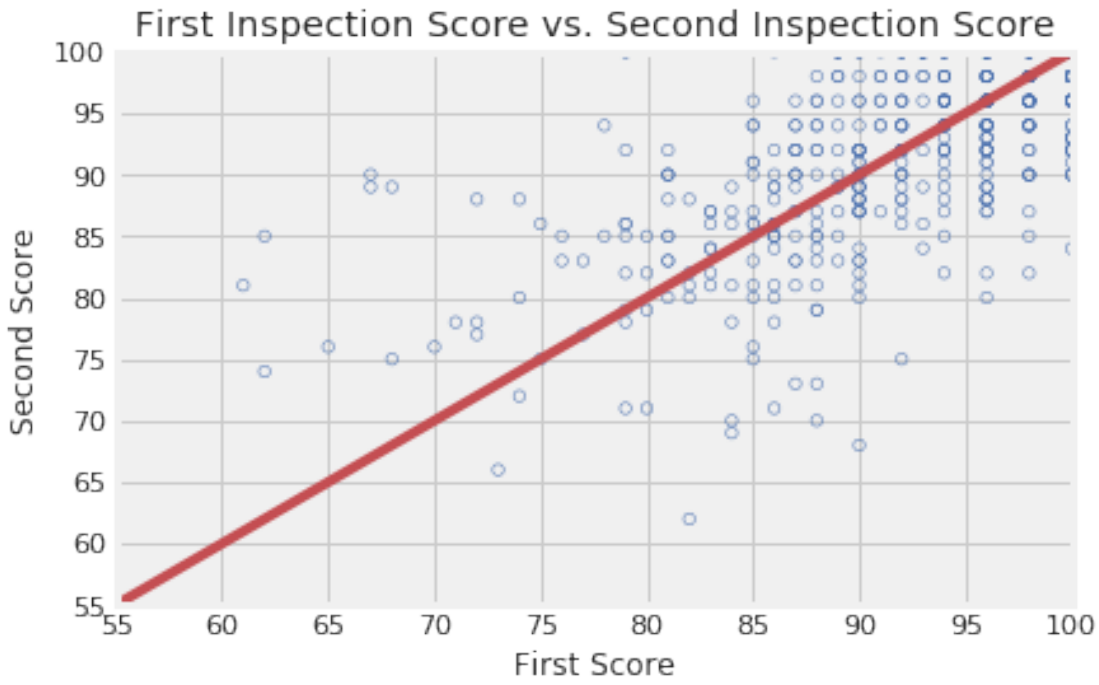
Describe the qualities of the distribution of the inspections scores based on your bar plot. Consider the mode(s), symmetry, tails, gaps, and anomalous values. Are there any unusual features of this distribution? What do your observations imply about the scores?

- **Mode:** Seems like an exp distribution, or part of a normal distribution or so
- **Symmetry:** No
- **Tails:** Nothing unusual
- **Gaps:** Score 95, 97, 99 are missing
- **Anomalous values:** Numbers of score 82, 84, 89, 91, 93, 98 are respectively lower than nearby scores (for 98 it's 96 and 100)
- **Imply:** Numbers of scores are not continuous and smooth, implying the chances that some of the number may be key point of qualitively difference of grading, leading to their nearby scores' missing or significant reduction in number.

Use the cell above to identify the restaurant with the lowest inspection scores ever. Be sure to include the name of the restaurant as part of your answer in the cell below. You can also head to yelp.com and look up the reviews page for this restaurant. Feel free to add anything interesting you want to share.

Lollipot This placed is reported closed, and the owner seems to still own money to one of its supplier. See the latest one star review at https://www.yelp.com/biz/lollipot-san-francisco?hrid=ref5f4BCaz7s8_r0K63W8Q&utm_campaign=www_review_share_popup&utm_medium=copy_link&utm_

Now, create your scatter plot in the cell below. It does not need to look exactly the same (e.g., no grid) as the sample below, but make sure that all labels, axes and data itself are correct.



Key pieces of syntax you'll need:

`plt.scatter` plots a set of points. Use `facecolors='none'` and `edgecolors=b` to make circle markers with blue borders.

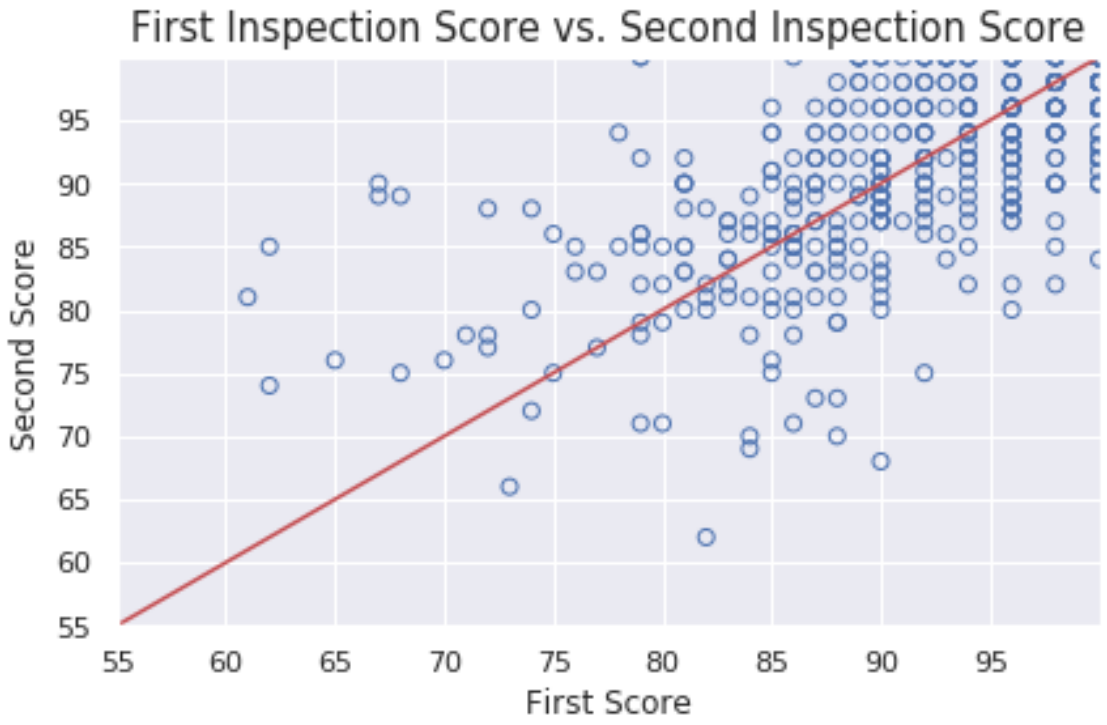
`plt.plot` for the reference line.

`plt.xlabel`, `plt.ylabel`, `plt.axis`, and `plt.title`.

Note: If you want to use another plotting library for your plots (e.g. `plotly`, `sns`) you are welcome to use that library instead so long as it works on DataHub.

Hint: You may find it convenient to use the `zip()` function to unzip scores in the list.

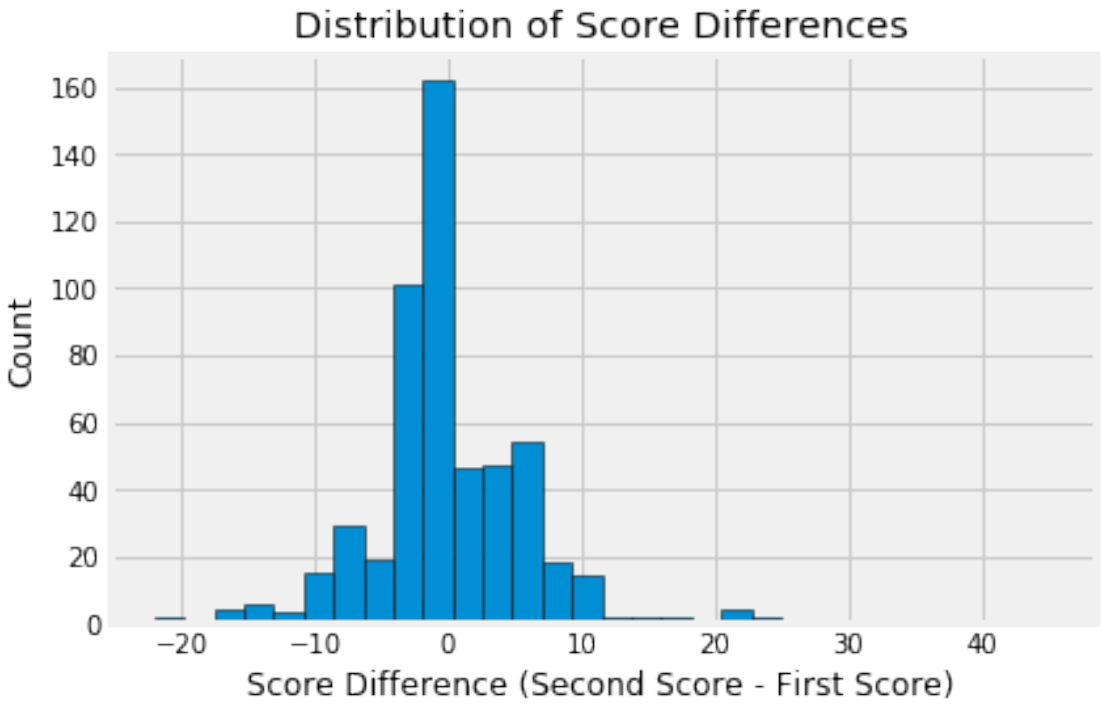
```
In [29]: # first_score, second_score created as implied by hint
         first_score = list(i[0] for i in scores_pairs_by_business.reset_index()["score_pair"])
         second_score = list(i[1] for i in scores_pairs_by_business.reset_index()["score_pair"])
         plt.scatter(first_score, \
                     second_score, \
                     facecolors='none', \
                     edgecolors='b')
         x = np.linspace(55, 100, 2)
         plt.plot(x,x,'r')
         plt.xlabel("First Score", fontsize=12)
         plt.ylabel("Second Score", fontsize=12)
         plt.axis([55, 100, 55, 100])
         plt.yticks(ticks=np.arange(55, 100, 5))
         plt.xticks(ticks=np.arange(55, 100, 5))
         plt.title("First Inspection Score vs. Second Inspection Score", fontsize=14.5);
```


First Inspection Score vs. Second Inspection Score
Second Score
First Score
55
60
65
70
75
80
85
90
95

0.0.3 Question 2d

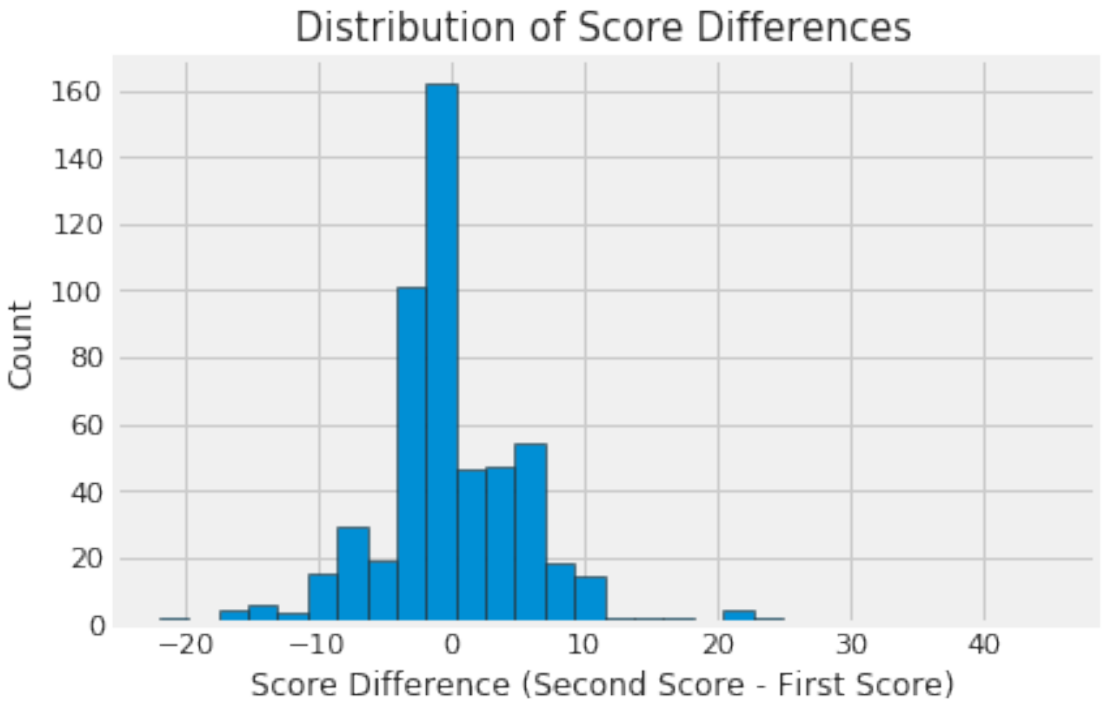
Another way to compare the scores from the two inspections is to examine the difference in scores. Subtract the first score from the second in `scores_pairs_by_business`. Make a histogram of these differences in the scores. We might expect these differences to be positive, indicating an improvement from the first to the second inspection.

The histogram should look like this:



Hint: Use `second_score` and `first_score` created in the scatter plot code above.
Hint: Convert the scores into numpy arrays to make them easier to deal with.
Hint: Use `plt.hist()` Try changing the number of bins when you call `plt.hist()`.

```
In [17]: first_score = np.array(first_score)
         second_score = np.array(second_score)
         x = second_score-first_score
         plt.hist(x, bins=30, edgecolor='k')
         plt.xlabel("Score Difference (Second Score - First Score)", fontsize=12)
         plt.ylabel("Count", fontsize=12)
         plt.title("Distribution of Score Differences", fontsize=14.5);
```


Distribution of Score Differences
Count
160
140
120
100
80
60
40
20
0
−20
−10
0
10
20
30
40
Score Difference (Second Score - First Score)

0.0.4 Question 2e

If restaurants' scores tend to improve from the first to the second inspection, what do you expect to see in the scatter plot that you made in question 2c? What do you oberve from the plot? Are your observations consistent with your expectations?

Hint: What does the slope represent?

What do you expect to see in the scatter plot that you made in question 2c?

Dots tend to be above the reference line.

What do you oberve from the plot?

Dots seem to be equally distributed near the line.

Are your observations consistent with your expectations?

No.

0.0.5 Question 2f

If a restaurant's score improves from the first to the second inspection, how would this be reflected in the histogram of the difference in the scores that you made in question 2d? What do you oberve from the plot? Are your observations consistent with your expectations? Explain your observations in the language of Statistics: for instance, the center, the spread, the deviation etc.

How would this be reflected in the histogram of the difference in the scores that you made in question 2d?

Bars tend to be normally distributed in positive x.

What do you oberve from the plot?

Bars seem to be normally distributed equally abound x = 0.

Are your observations consistent with your expectations?

No all.

Explain your observations in the language of Statistics: for instance, the center, the spread, the deviation etc.

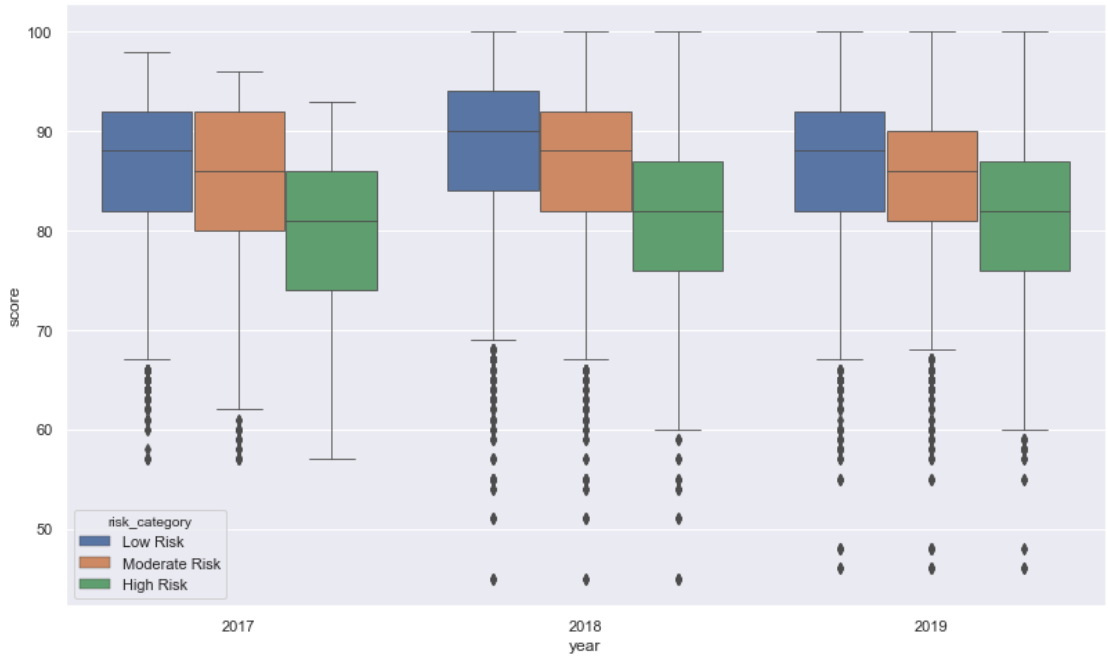
For instance, the center is expected to be positive, the spread is expected to be normally distributed around the center.

While the observations shows that the center is around 0, the spread is expected to be normally distributed around the center.

0.0.6 Question 2g

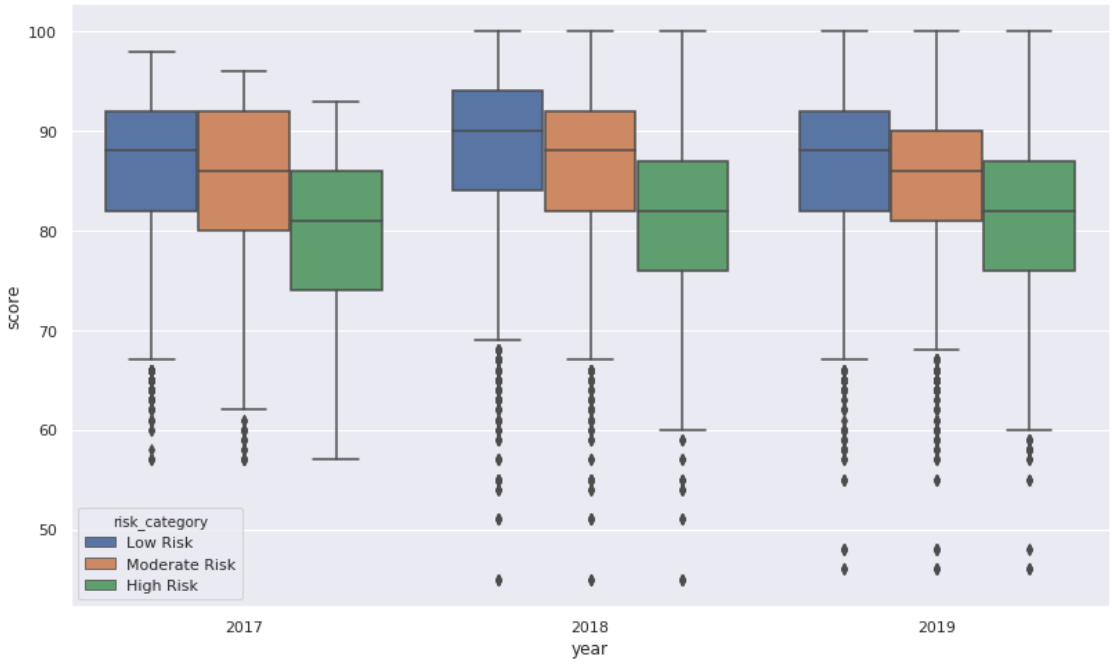
To wrap up our analysis of the restaurant ratings over time, one final metric we will be looking at is the distribution of restaurant scores over time. Create a side-by-side boxplot that shows the distribution of these scores for each different risk category from 2017 to 2019. Use a figure size of at least 12 by 8.

The boxplot should look similar to the sample below:



Hint: Use `sns.boxplot()`. Try taking a look at the first several parameters.
Hint: Use `plt.figure()` to adjust the figure size of your plot.

```
In [18]: ins_named_q2g = ins_named.merge(ins2vio, left_on = 'iid', right_on = 'iid')
         ins_named_q2g = ins_named_q2g.merge(vio, left_on = 'vid', right_on = 'vid')
         ins_named_q2g = ins_named_q2g[(ins_named_q2g["score"] >= 0) & (ins_named_q2g["year"] >= 2017)]

         # Do not modify this line
         sns.set()
         plt.figure(figsize=(12,8))
         ax = sns.boxplot( \
             data = ins_named_q2g, \
             x = "year", \
             y = "score", \
             hue = "risk_category", \
             hue_order = ["Low Risk", "Moderate Risk", "High Risk"]
         )
         ax.set_ylabel("score")
         ax.set_xlabel("year")
         ax.set_title("");
```

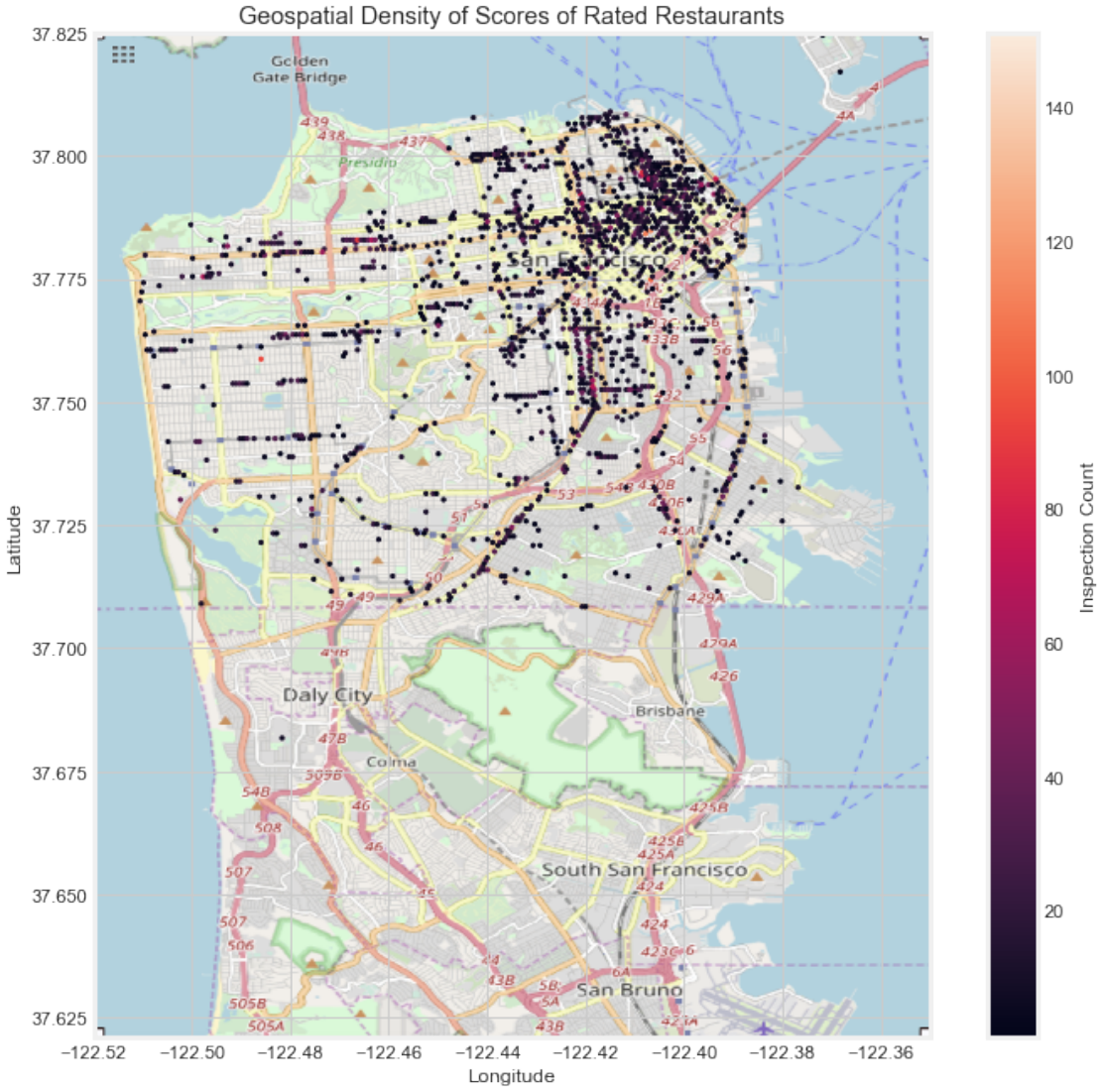

100
90
80
70
60
50
score
risk_category
Low Risk
Moderate Risk
High Risk
2017
2018
2019
year

0.0.7 Question 3b

Now that we have our DataFrame ready, we can start creating our geospatial hexbin plot.

Using the `rated_geo` DataFrame from 3a, produce a geospatial hexbin plot that shows the inspection count for all restaurant locations in San Francisco.

Your plot should look similar to the one below:



Hint: Use `pd.DataFrame.plot.hexbin()` or `plt.hexbin()` to create the hexbin plot.

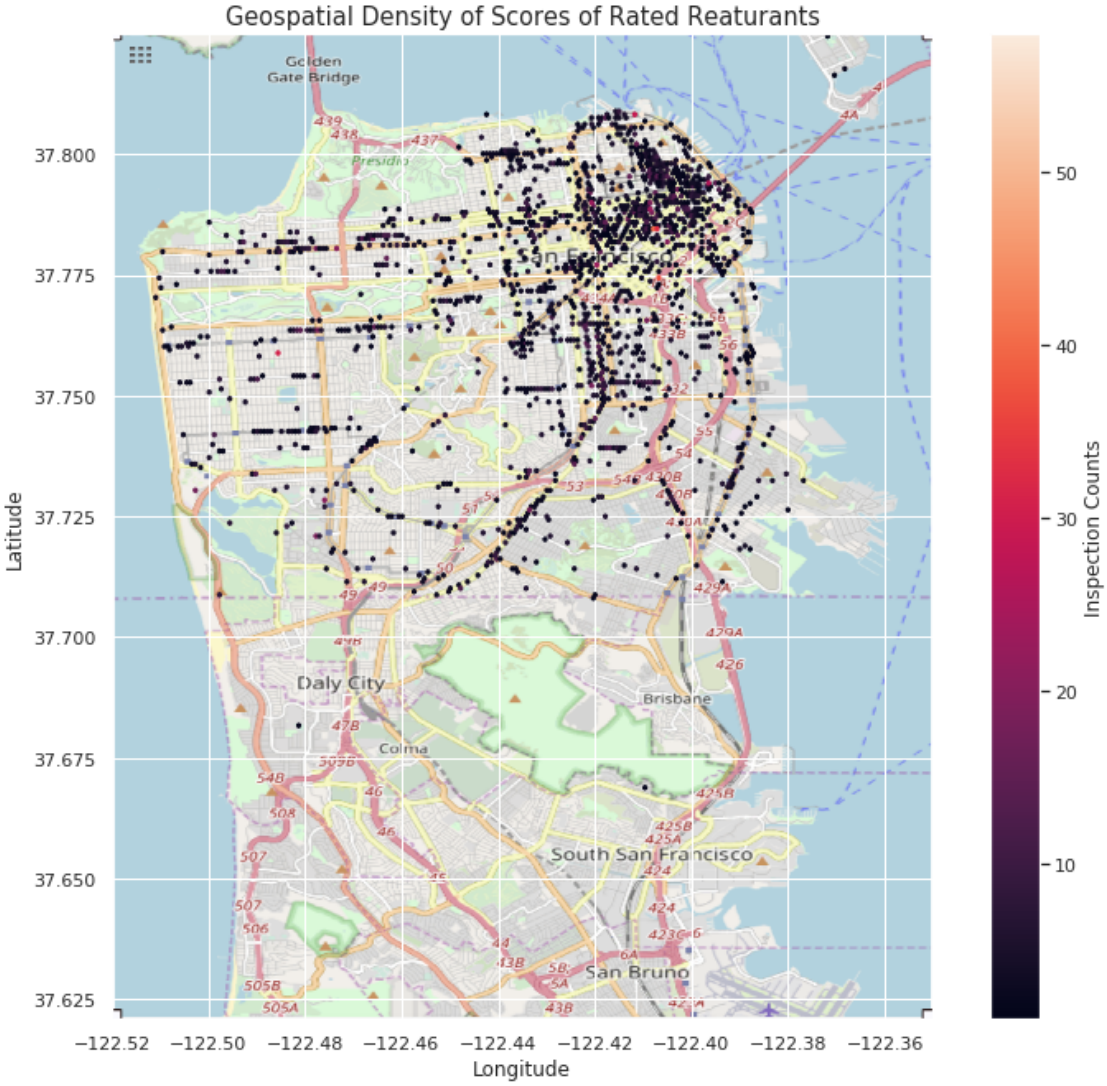
Hint: For the 2 functions we mentioned above, try looking at the parameter `reduce_C_function`, which determines the aggregate function for the hexbin plot.

Hint: Use `fig.colorbar()` to create the color bar to the right of the hexbin plot.

Hint: Try using a `gridsize` of 200 when creating your hexbin plot; it makes the plot cleaner.

```
In [21]: # DO NOT MODIFY THIS BLOCK
         min_lon = rated_geo['longitude'].min()
         max_lon = rated_geo['longitude'].max()
         min_lat = rated_geo['latitude'].min()
```

```
max_lat = rated_geo['latitude'].max()
max_score = rated_geo['score'].max()
min_score = rated_geo['score'].min()
bound = ((min_lon, max_lon, min_lat, max_lat))
min_lon, max_lon, min_lat, max_lat
map_bound = ((-122.5200, -122.3500, 37.6209, 37.8249))
# DO NOT MODIFY THIS BLOCK

# Read in the base map and setting up subplot
# DO NOT MODIFY THESE LINES
basemap = plt.imread('./data/sf.png')
fig, ax = plt.subplots(figsize = (11,11))
ax.set_xlim(map_bound[0],map_bound[1])
ax.set_ylim(map_bound[2],map_bound[3])
# DO NOT MODIFY THESE LINES


# Create the hexbin plot
hb = plt.hexbin(\
                rated_geo["longitude"], \
                rated_geo["latitude"], \
                C = rated_geo["score"], \
                gridsize = 200 , \
                reduce_C_function = len, \
                #cmap = "Wistia" \
               )
cbar = fig.colorbar(hb, ax=ax)
cbar.ax.set_ylabel('Inspection Counts')
plt.xlabel("Longitude", fontsize=12)
plt.ylabel("Latitude", fontsize=12)
plt.title("Geospatial Density of Scores of Rated Reaturants", fontsize=14.5)

# Setting aspect ratio and plotting the hexbins on top of the base map layer
# DO NOT MODIFY THIS LINE
ax.imshow(basemap, zorder=0, extent = map_bound, aspect= 'equal');
# DO NOT MODIFY THIS LINE
```

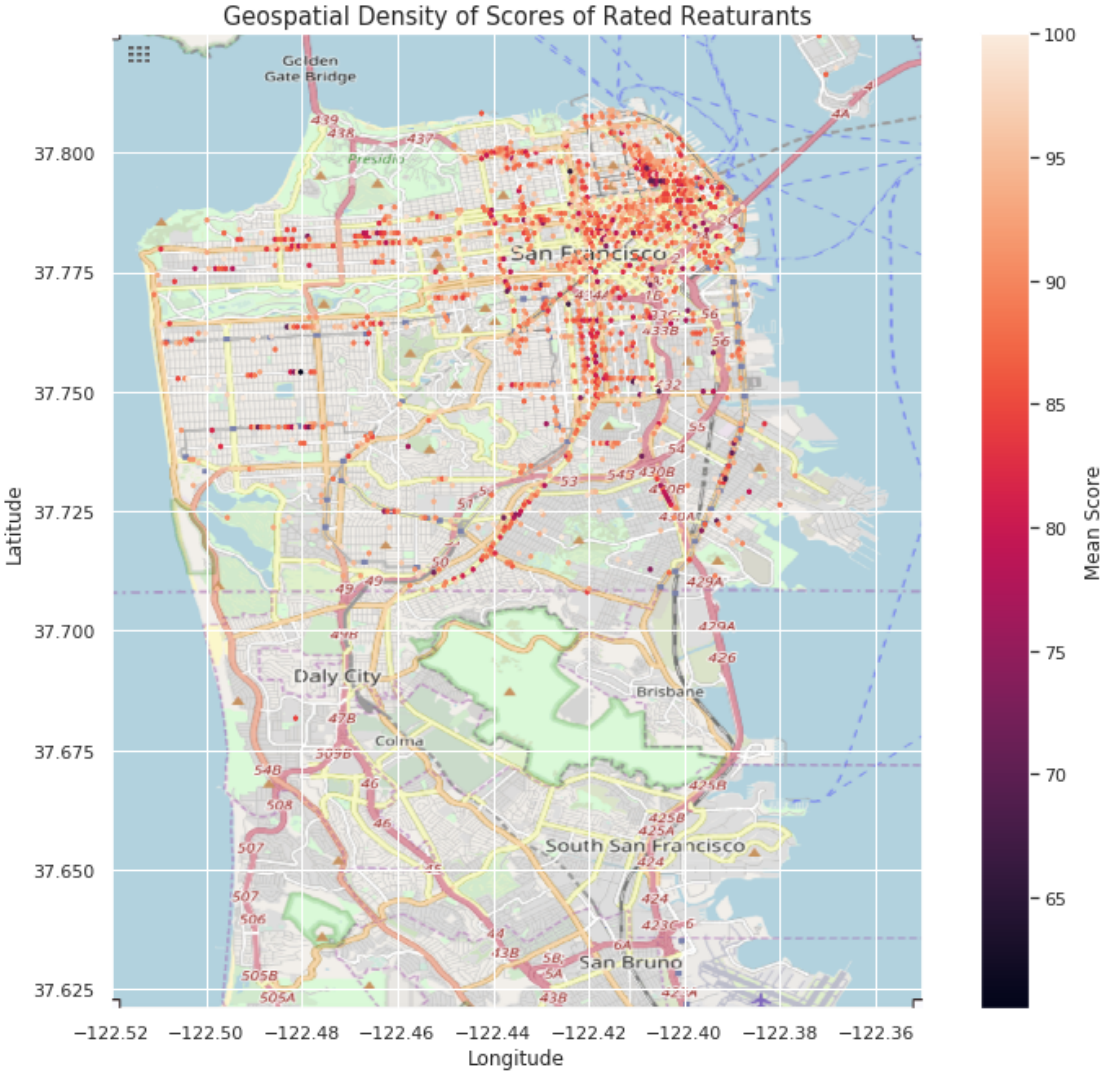

Geospatial Density of Scores of Rated Reaturants
Golden Gate Bridge
San Francisco
Daly City
Brisbane
Colma
South San Francisco
San Bruno
Latitude
37.800
37.775
37.750
37.725
37.700
37.675
37.650
37.625
−122.52
−122.50
−122.48
−122.46
−122.44
−122.42
−122.40
−122.38
−122.36
Longitude
Inspection Counts
50
40
30
20
10

0.0.8 Question 3c

Now that we've created our geospatial hexbin plot for the density of inspection scores for restaurants in San Francisco, let's also create another hexbin plot that visualizes the **average inspection scores** for restaurants in San Francisco.

Hint: If you set up everything correctly in 3b, you should only need to change 1 parameter here to produce the plot.

In [22]:
```
# Read in the base map and setting up subplot
# DO NOT MODIFY THESE LINES
basemap = plt.imread('./data/sf.png')
fig, ax = plt.subplots(figsize = (11,11))
ax.set_xlim(map_bound[0],map_bound[1])
ax.set_ylim(map_bound[2],map_bound[3])
# DO NOT MODIFY THESE LINES

# Create the hexbin plot
hb = plt.hexbin(\
                rated_geo["longitude"], \
                rated_geo["latitude"], \
                C = rated_geo["score"], \
                gridsize = 200 , \
                reduce_C_function = np.mean, \
                #cmap = "Wistia" \
               )
cbar = fig.colorbar(hb, ax=ax)
cbar.ax.set_ylabel('Mean Score')
plt.xlabel("Longitude", fontsize=12)
plt.ylabel("Latitude", fontsize=12)
plt.title("Geospatial Density of Scores of Rated Reaturants", fontsize=14.5)


# Setting aspect ratio and plotting the hexbins on top of the base map layer
# DO NOT MODIFY THIS LINE
ax.imshow(basemap, zorder=0, extent = map_bound, aspect= 'equal');
# DO NOT MODIFY THIS LINE
```


Geospatial Density of Scores of Rated Reaturants
Golden Gate Bridge
San Francisco
Daly City
Brisbane
Colma
South San Francisco
San Bruno
Presidio
37.800
37.775
37.750
37.725
37.700
37.675
37.650
37.625
Latitude
−122.52
−122.50
−122.48
−122.46
−122.44
−122.42
−122.40
−122.38
−122.36
Longitude
100
95
90
85
80
75
70
65
Mean Score

0.0.9 Question 3d

Given the 2 hexbin plots you have just created above, did you notice any connection between the first plot where we aggregate over the **inspection count** and the second plot where we aggregate over the **inspection mean**? In several sentences, comment your observations in the cell below.

Here're some of the questions that might be interesting to address in your response:

- Roughly speaking, did you notice any of the actual locations (districts/places of interest) where inspection tends to be more frequent? What about the locations where the average inspection score tends to be low?
- Is there any connection between the locations where there are more inspections and the locations where the average inspection score is low?
- What have might led to the connections that you've identified?

- **Roughly speaking, did you notice any of the actual locations (districts/places of interest) where inspection tends to be more frequent? What about the locations where the average inspection score tends to be low?**

Before answering the question, I would ike to point out my observation strategy as following. Places where dots is denser IS NOT considered inspected more frequently, as it only represents more resturants within specific area. Dots whose color in q3b is brighter IS considered inspected more frequently.

122.407W, 17.795N (West of Financial District) That whole area seems to get more frequent inspection.

Generally all parts of San Franciso other than Union Square tends to have low average inspection score, especially around Chinatown.

- **Is there any connection between the locations where there are more inspections and the locations where the average inspection score is low?**

In my opinion the connection would be trivial, as whole area west of Financial District tends to get more inspections, which includes places that average inspection score is tipically high (near Union Square) and low (near Chinatown)

- **What have might led to the connections that you've identified?**

Since I've identified none major connections as mentioned, I would guess that food inspection frequency is based more on other factors rather than merely past inspection score.

One possible facor is popularity, as popular area have more customers where food safty tends to be more important.

0.0.10 Grading

Since the assignment is more open ended, we will have a more relaxed rubric, classifying your answers into the following three categories:

- **Great** (4-5 points): The chart is well designed, and the data computation is correct. The text written articulates a reasonable metric and correctly describes the relevant insight and answer to the question you are interested in.
- **Passing** (3-4 points): A chart is produced but with some flaws such as bad encoding. The text written is incomplete but makes some sense.
- **Unsatisfactory** (<= 2 points): No chart is created, or a chart with completely wrong results.

We will lean towards being generous with the grading. We might also either discuss in discussion or post on Piazza some examplar analysis you have done (with your permission)!

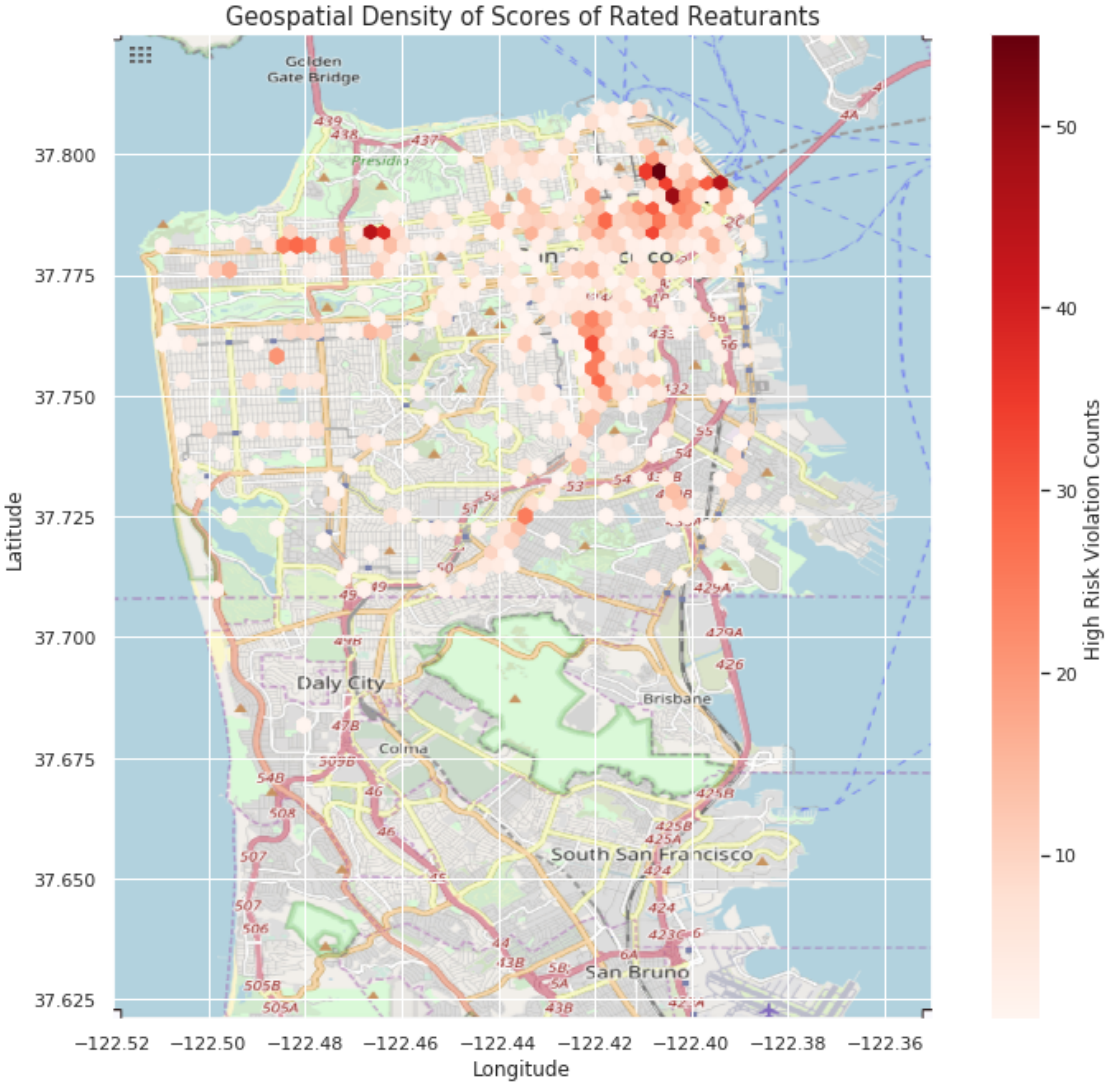
You should have the following in your answers: * a few visualizations; Please limit your visualizations to 5 plots. * a few sentences (not too long please!)

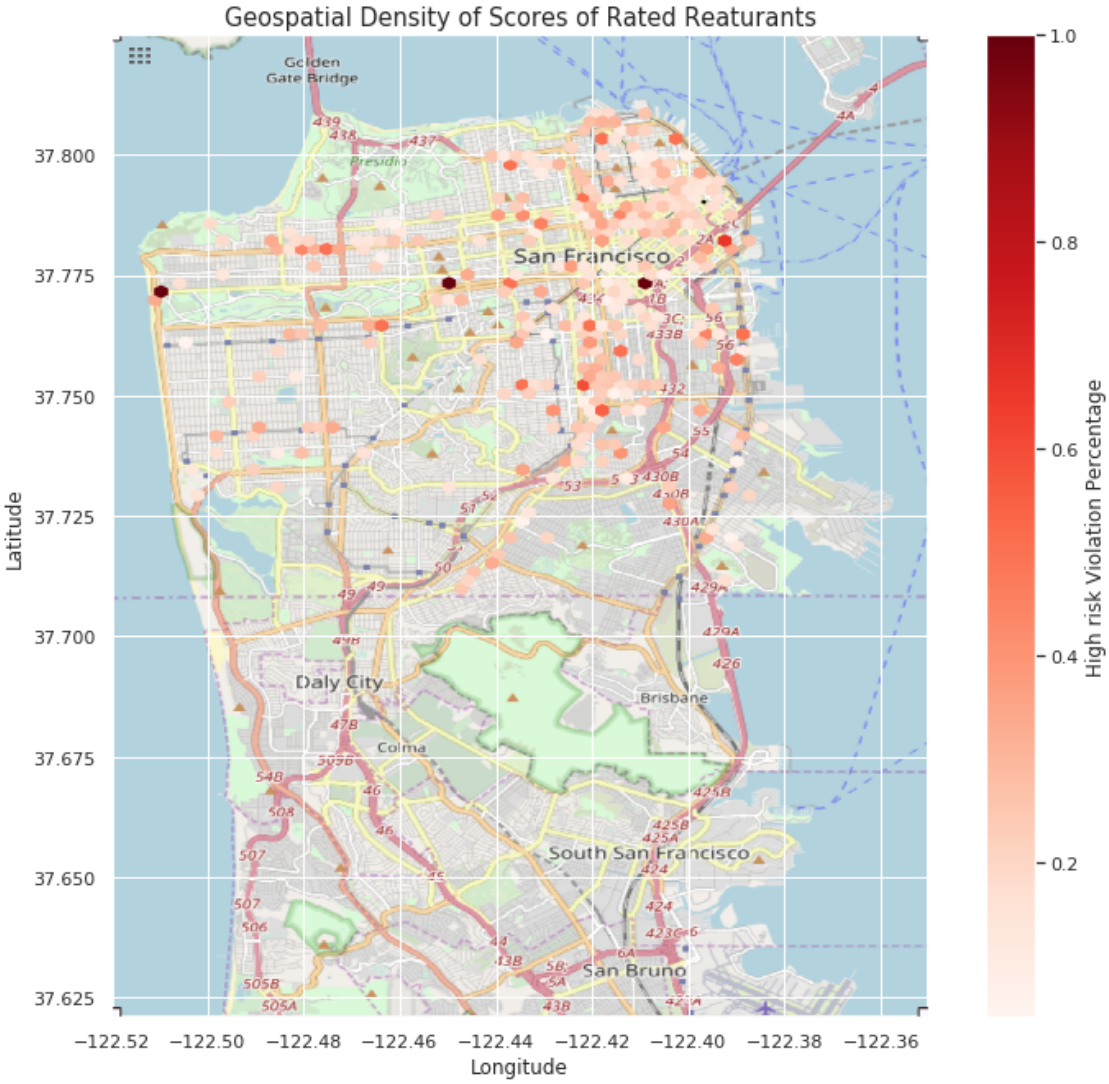
Please note that you will only receive support in OH and Piazza for Matplotlib and seaborn questions. However, you may use some other Python libraries to help you create you visualizations. If you do so, make sure it is compatible with the PDF export (e.g., Plotly does not create PDFs properly, which we need for Gradescope).

In [23]:
```
# YOUR DATA PROCESSING AND PLOTTING HERE
pd.options.mode.chained_assignment = None  # default='warn'
########################################
# High Risk Violation Counts Map
########################################
ins_q4 = ins.merge(bus, left_on = 'bid', right_on = 'bid')
ins_q4 = ins_q4.merge(ins2vio, left_on = 'iid', right_on = 'iid')
ins_q4 = ins_q4.merge(vio, left_on = 'vid', right_on = 'vid')

highrisk_geo = ins_q4[['latitude', 'longitude', 'score', 'risk_category']]
highrisk_geo = highrisk_geo[ \
    (highrisk_geo["score"] >= 0) & \
    (highrisk_geo["longitude"] > -9999) & \
    (highrisk_geo["longitude"] < 0) & \
    (highrisk_geo["latitude"] > -9999) & \
    (highrisk_geo["risk_category"] == "High Risk") \
]
highrisk_geo

# DO NOT MODIFY THIS BLOCK
min_lon = rated_geo['longitude'].min()
max_lon = rated_geo['longitude'].max()
min_lat = rated_geo['latitude'].min()
max_lat = rated_geo['latitude'].max()
max_score = rated_geo['score'].max()
min_score = rated_geo['score'].min()
bound = ((min_lon, max_lon, min_lat, max_lat))
min_lon, max_lon, min_lat, max_lat
map_bound = ((-122.5200, -122.3500, 37.6209, 37.8249))
# DO NOT MODIFY THIS BLOCK

# Read in the base map and setting up subplot
# DO NOT MODIFY THESE LINES
basemap = plt.imread('./data/sf.png')
```

```
fig, ax = plt.subplots(figsize = (11,11))
ax.set_xlim(map_bound[0],map_bound[1])
ax.set_ylim(map_bound[2],map_bound[3])
# DO NOT MODIFY THESE LINES

# Create the hexbin plot
hb = plt.hexbin( \
    highrisk_geo["longitude"], \
    highrisk_geo["latitude"], \
    C = highrisk_geo["score"], \
    gridsize = 50 , \
    reduce_C_function = len, \
    cmap = "Reds" \
)
cbar = fig.colorbar(hb, ax=ax)
cbar.ax.set_ylabel('High Risk Violation Counts')
plt.xlabel("Longitude", fontsize=12)
plt.ylabel("Latitude", fontsize=12)
plt.title("Geospatial Density of Scores of Rated Reaturants", fontsize=14.5)

# Setting aspect ratio and plotting the hexbins on top of the base map layer
# DO NOT MODIFY THIS LINE
ax.imshow(basemap, zorder=0, extent = map_bound, aspect= 'equal');
# DO NOT MODIFY THIS LINE

########################################
# High Risk Violation Percentage Map
########################################
ins_q4 = ins.merge(bus, left_on = 'bid', right_on = 'bid')
ins_q4 = ins_q4.merge(ins2vio, left_on = 'iid', right_on = 'iid')
ins_q4 = ins_q4.merge(vio, left_on = 'vid', right_on = 'vid')

risk_geo = ins_q4[['bid', 'latitude', 'longitude', 'score', 'risk_category']]

risk_geo = risk_geo[ \
    (risk_geo["score"] >= 0) & \
    (risk_geo["longitude"] > -9999) & \
    (risk_geo["longitude"] < 0) & \
    (risk_geo["latitude"] > -9999) \
]

risk_geo_count = risk_geo    .groupby("bid") \
                             .agg({'risk_category': len}) \
                             .rename(columns={"risk_category":"risk_count"})
highrisk_geo_count = risk_geo[(risk_geo["risk_category"] == "High Risk")]
highrisk_geo_count = highrisk_geo_count .groupby("bid") \
                                        .agg({'risk_category': len}) \
                                        .rename(columns={"risk_category":"risk_count"})
highrisk_geo_percent = risk_geo[['latitude', 'longitude']]
highrisk_geo_percent["high_risk_percent"] =  highrisk_geo_count["risk_count"]\
                                            .div(risk_geo_count["risk_count"])
highrisk_geo_percent = highrisk_geo_percent[highrisk_geo_percent["high_risk_percent"] >= 0]
highrisk_geo_percent
```

```
# DO NOT MODIFY THIS BLOCK
min_lon = rated_geo['longitude'].min()
max_lon = rated_geo['longitude'].max()
min_lat = rated_geo['latitude'].min()
max_lat = rated_geo['latitude'].max()
max_score = rated_geo['score'].max()
min_score = rated_geo['score'].min()
bound = ((min_lon, max_lon, min_lat, max_lat))
min_lon, max_lon, min_lat, max_lat
map_bound = ((-122.5200, -122.3500, 37.6209, 37.8249))
# DO NOT MODIFY THIS BLOCK

# Read in the base map and setting up subplot
# DO NOT MODIFY THESE LINES
basemap = plt.imread('./data/sf.png')
fig, ax = plt.subplots(figsize = (11,11))
ax.set_xlim(map_bound[0],map_bound[1])
ax.set_ylim(map_bound[2],map_bound[3])
# DO NOT MODIFY THESE LINES

# Create the hexbin plot
hb = plt.hexbin( \
    highrisk_geo_percent["longitude"], \
    highrisk_geo_percent["latitude"], \
    C = highrisk_geo_percent["high_risk_percent"], \
    gridsize = 50 , \
    reduce_C_function = np.mean, \
    cmap = "Reds" \
)
cbar = fig.colorbar(hb, ax=ax)
cbar.ax.set_ylabel('High risk Violation Percentage')
plt.xlabel("Longitude", fontsize=12)
plt.ylabel("Latitude", fontsize=12)
plt.title("Geospatial Density of Scores of Rated Reaturants", fontsize=14.5)

# Setting aspect ratio and plotting the hexbins on top of the base map layer
# DO NOT MODIFY THIS LINE
ax.imshow(basemap, zorder=0, extent = map_bound, aspect= 'equal');
# DO NOT MODIFY THIS LINE



# YOUR EXPLANATION HERE (in a comment)
# High Risk Violation Counts Map and High Risk Violation Percentage Map are drawn to show
# that despite some areas have more high risk inspection counts, they are not so risky as
# high risk event percent (of all inspections) are generally evenly distributed around the
# whole city.
```


Geospatial Density of Scores of Rated Reaturants
Golden Gate Bridge
Daly City
Brisbane
Colma
South San Francisco
San Bruno
37.800
37.775
37.750
37.725
37.700
37.675
37.650
37.625
Latitude
−122.52
−122.50
−122.48
−122.46
−122.44
−122.42
−122.40
−122.38
−122.36
Longitude
50
40
30
20
10
High Risk Violation Counts


Geospatial Density of Scores of Rated Reaturants
Golden Gate Bridge
San Francisco
Daly City
Colma
Brisbane
South San Francisco
San Bruno
37.800
37.775
37.750
37.725
37.700
37.675
37.650
37.625
Latitude
−122.52
−122.50
−122.48
−122.46
−122.44
−122.42
−122.40
−122.38
−122.36
Longitude
1.0
0.8
0.6
0.4
0.2
High risk Violation Percentage