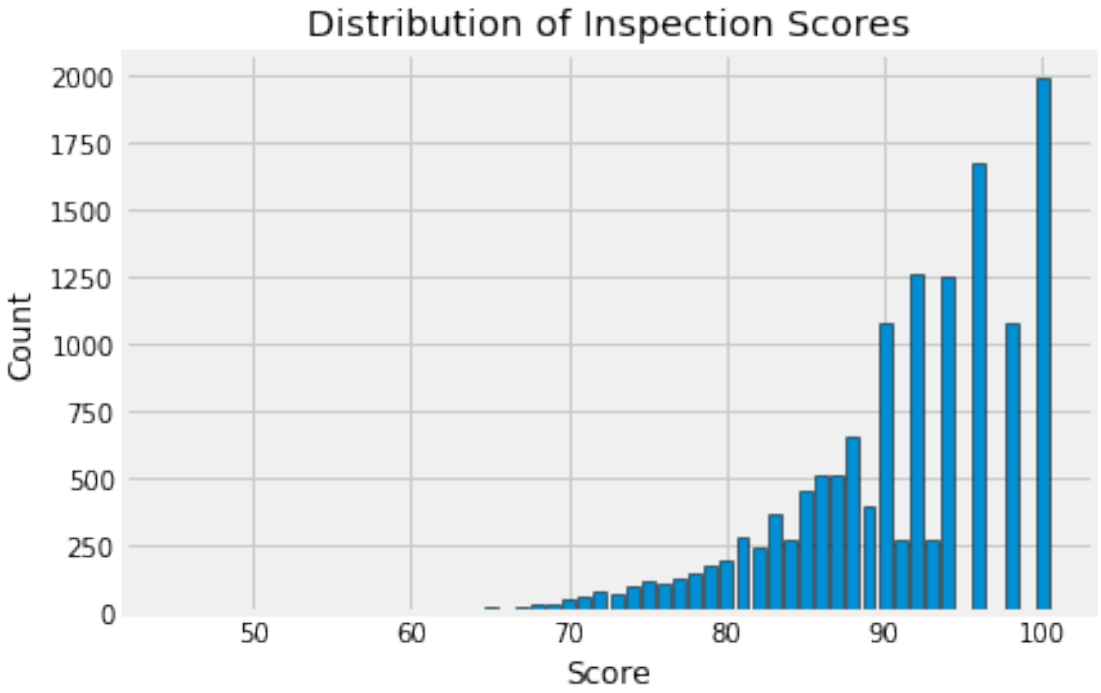
Notebook

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.



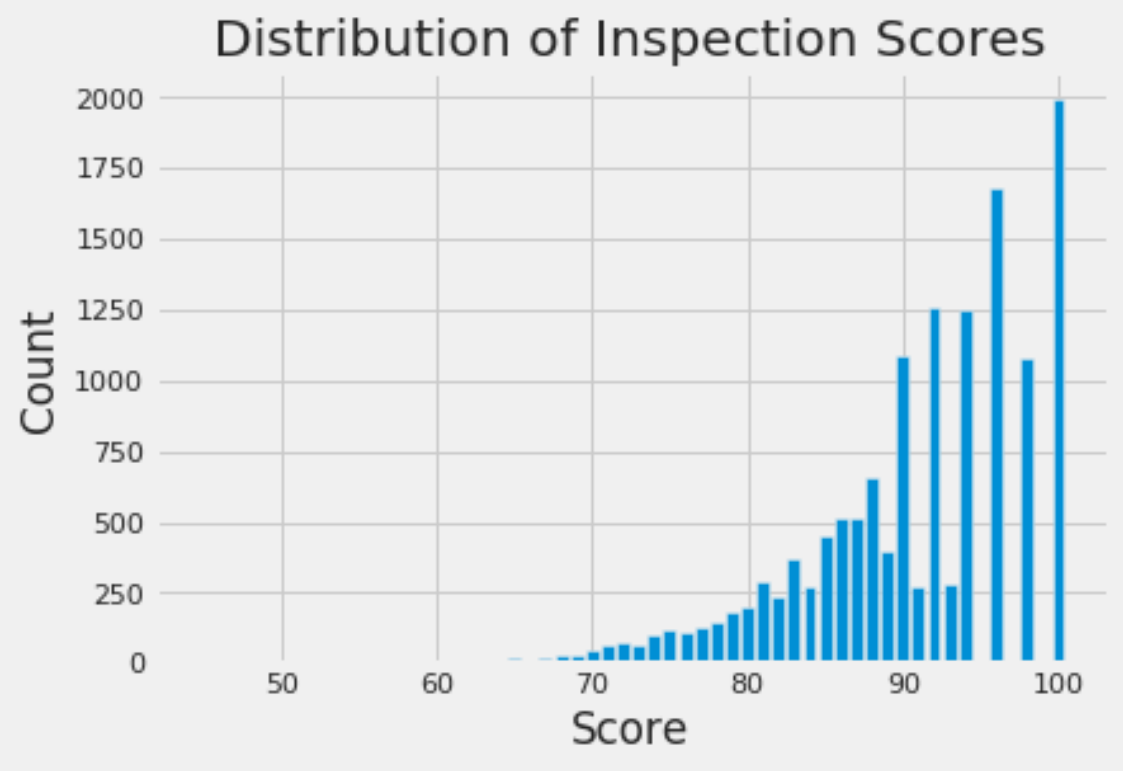
You might find this matplotlib.pyplot tutorial useful. Key syntax that you'll need:

```
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 [105]: plt.xlabel('Score')
          plt.ylabel('Count')
          plt.title('Distribution of Inspection Scores')
          ins_pos_score = ins.where(ins['score'] > 0)
          xrange = np.arange(0, 100, 1)
          ins = ins_pos_score.dropna()
          counts = ins.groupby('score').count()
          heights = counts['iid'].values
          plt.bar(counts.index, heights)
```

Out[105]: <BarContainer object of 47 artists>



0.0.2 Question 1b

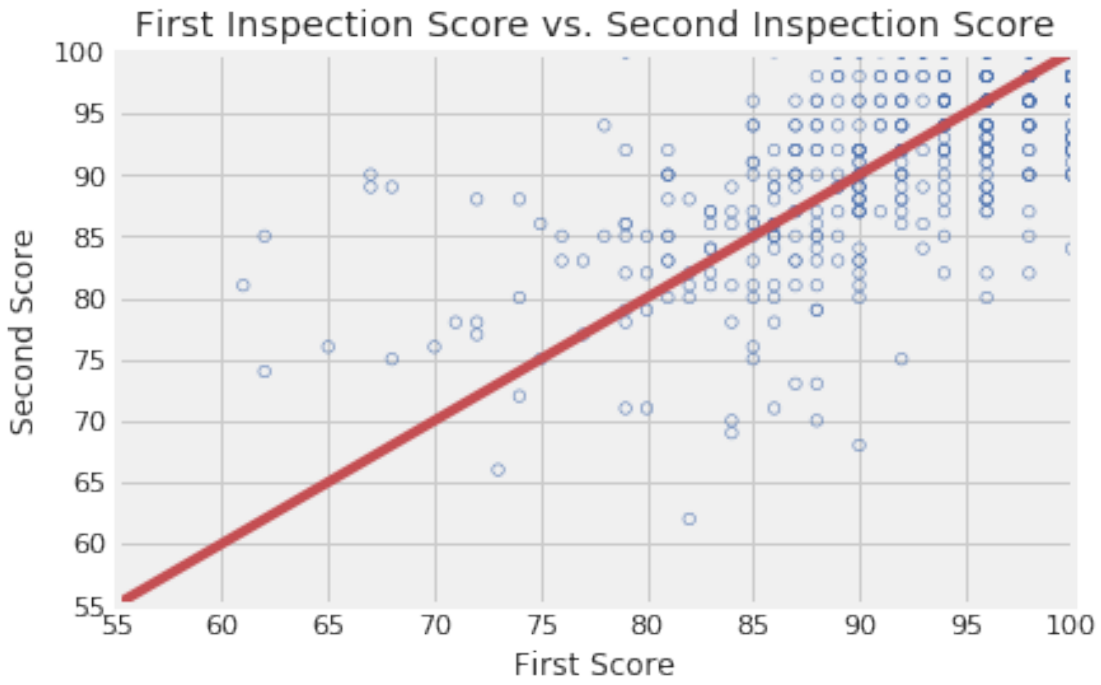
Describe the qualities of the distribution of the inspections scores based on your bar plot. Consider the mode(s), symmetry, tails, gaps, and anomalous values. Are there any unusual features of this distribution? What do your observations imply about the scores?

First, the mode is 100. It's not symmetric. The tails are on the left (around 65). There are gaps, but they are most obvious from 90-100. No outliers. Very few (below 500 counts) scored 91 or 93. But much more (1300 counts) scored 91 or 94. Large proportion of restaurants are scored really high (above 90).

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.

Amici's East Coast Pizzeria has the lowest inspection scores. I found this review particularly amusinf on yelp. "Insane prices for tiny portions of average food. Whatever they can get away with charging they will. The only thing East Coast about this place is the big middle finger to their customers."

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.

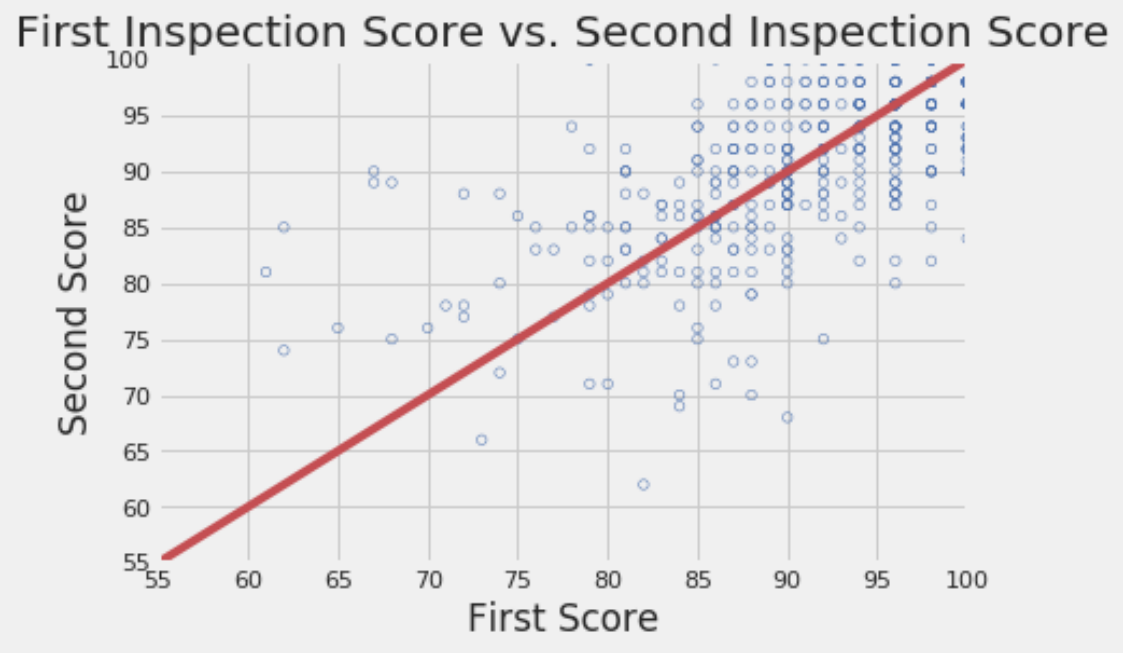
`plt.plot` for the reference line.

`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.

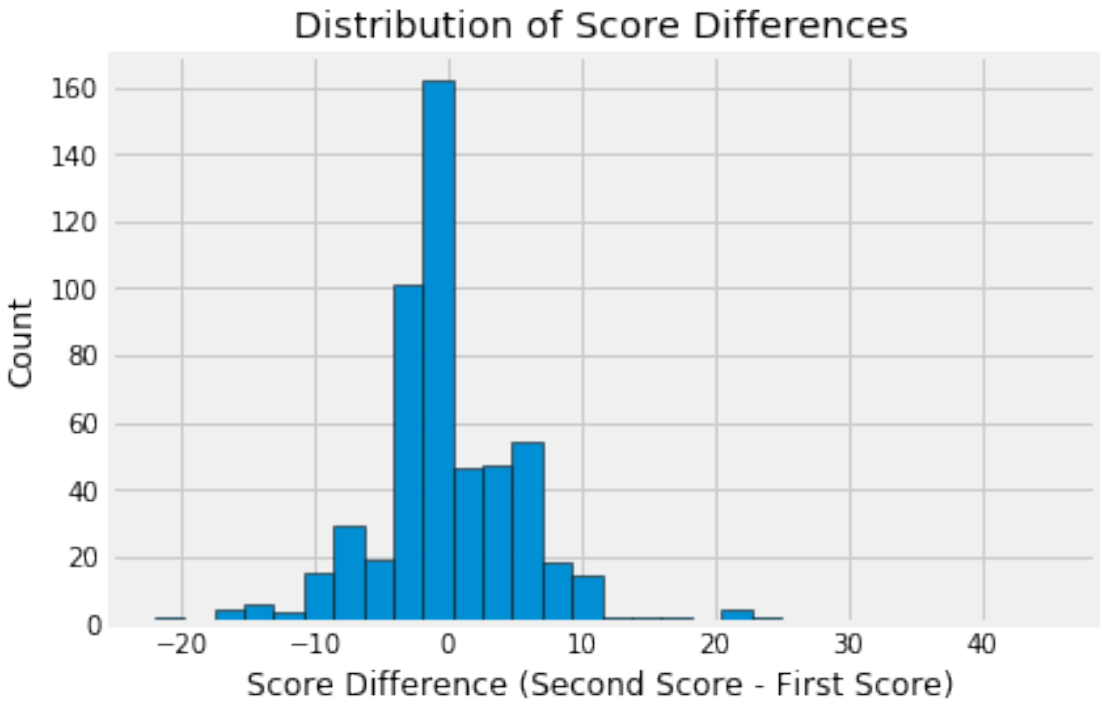
```
In [116]: first_score, second_score = zip(*scores_pairs_by_business['score_pair'])
          plt.scatter(first_score, second_score, s=20, facecolors='none', edgecolors='b')
          plt.plot([55, 100], [55, 100], 'r-')
          plt.xlabel('First Score')
          plt.ylabel('Second Score')
          plt.axis([55, 100, 55, 100])
          plt.title("First Inspection Score vs. Second Inspection Score");
```


First Inspection Score vs. Second Inspection Score
Second Score
First Score
100
95
90
85
80
75
70
65
60
55
55
60
65
70
75
80
85
90
95
100

0.0.3 Question 2d

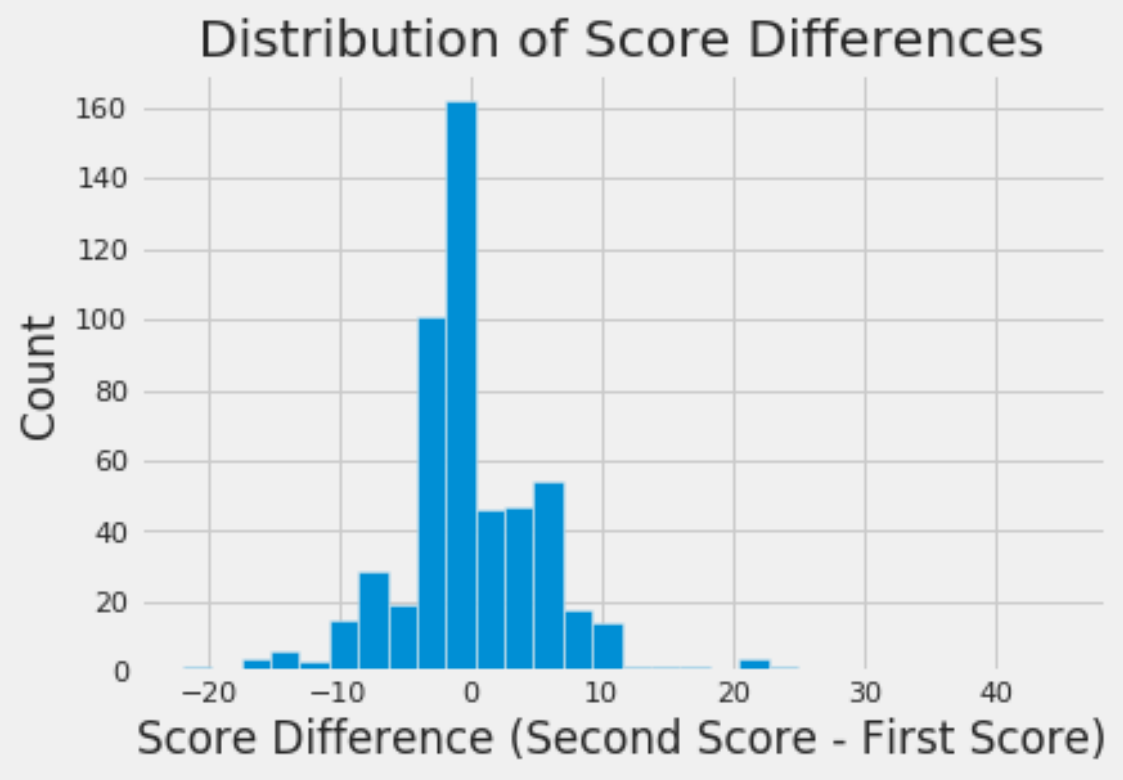
Another way to compare the scores from the two inspections is to examine the difference in scores. Subtract the first score from the second in `scores_pairs_by_business`. Make a histogram of these differences in the scores. We might expect these differences to be positive, indicating an improvement from 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 [119]: diffs = np.array(second_score) - np.array(first_score)
          plt.hist(diffs, bins=30)
          plt.title("Distribution of Score Differences")
          plt.xlabel("Score Difference (Second Score - First Score)")
          plt.ylabel("Count");
```


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?

If the restaurants tend to improve from the first to the second inspection, we would expect to see the points in the scatter plot fall above the line of slope 1. We would also expect to see the histogram of the difference in scores to be shifted toward positive values. Interestingly, we don't see this. The second inspection often is worse than first.

0.0.5 Question 2f

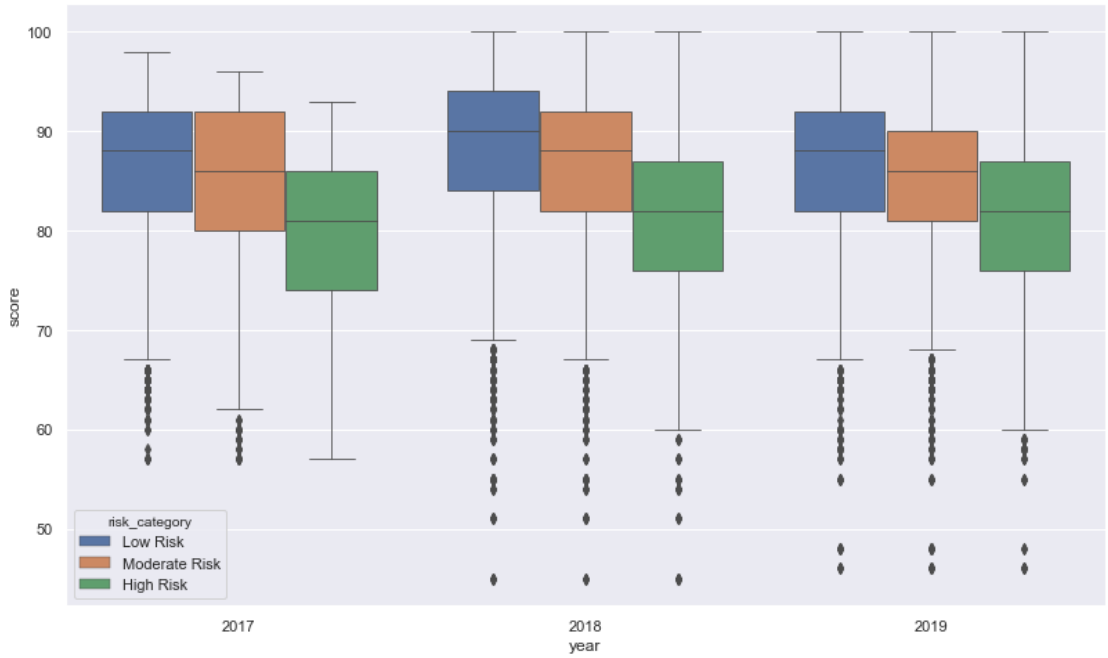
If a restaurant's score improves from the first to the second inspection, how would this be reflected in the histogram of the difference in the scores that you made in question 2d? What do you oberve from the plot? Are your observations consistent with your expectations? Explain your observations in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

The histogram of differences shows a unimodal distribution centered at 0, hinting that the average restaurant does not see a change in score between their first and second inspection. This distribution has long tails with some scores being as low as -20 and others as high as 20-30.

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:

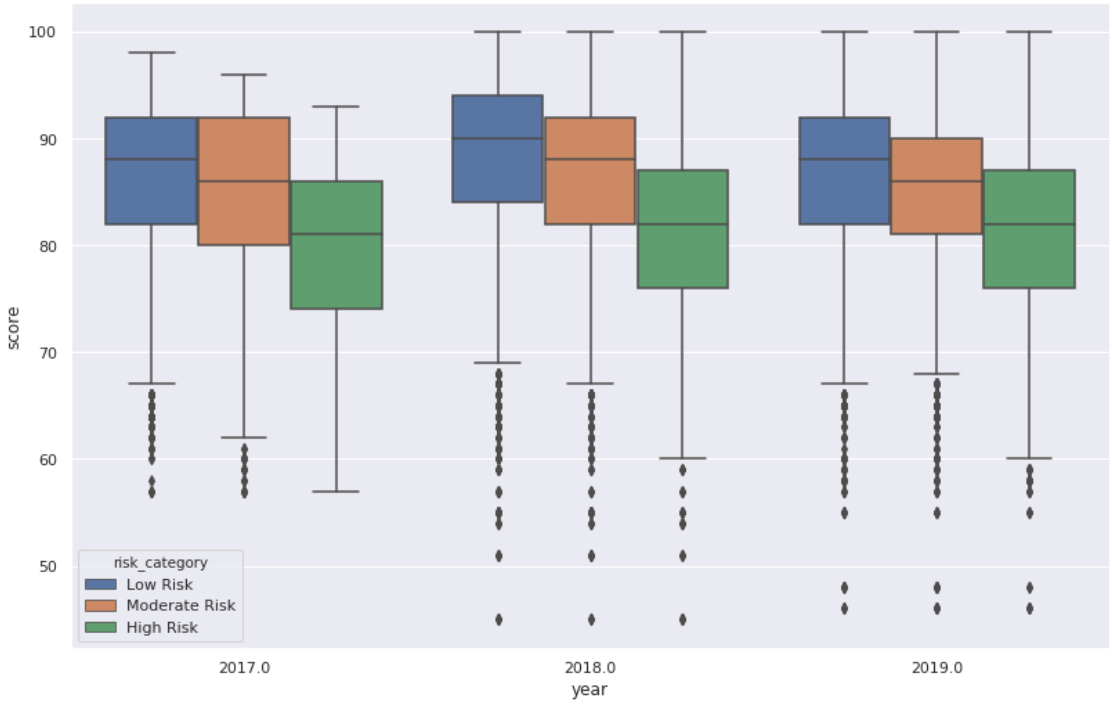


Hint: Use `sns.boxplot()`. Try taking a look at the first several parameters.

Hint: Use `plt.figure()` to adjust the figure size of your plot.

```
In [120]: # Do not modify this line
          sns.set()
          first = pd.merge(vio[['risk_category', 'vid']], ins2vio, how='left')
          second = pd.merge(first, ins[['iid', 'year', 'score']], how='left')
          merge = second[second['year'] >= 2017]
          plt.figure(figsize=(12, 8))
          sns.boxplot(merge['year'], merge['score'], hue=merge['risk_category'],
                      hue_order=['Low Risk', 'Moderate Risk', 'High Risk'])
```

```
Out[120]: <matplotlib.axes._subplots.AxesSubplot at 0x7fa8ae29e898>
```

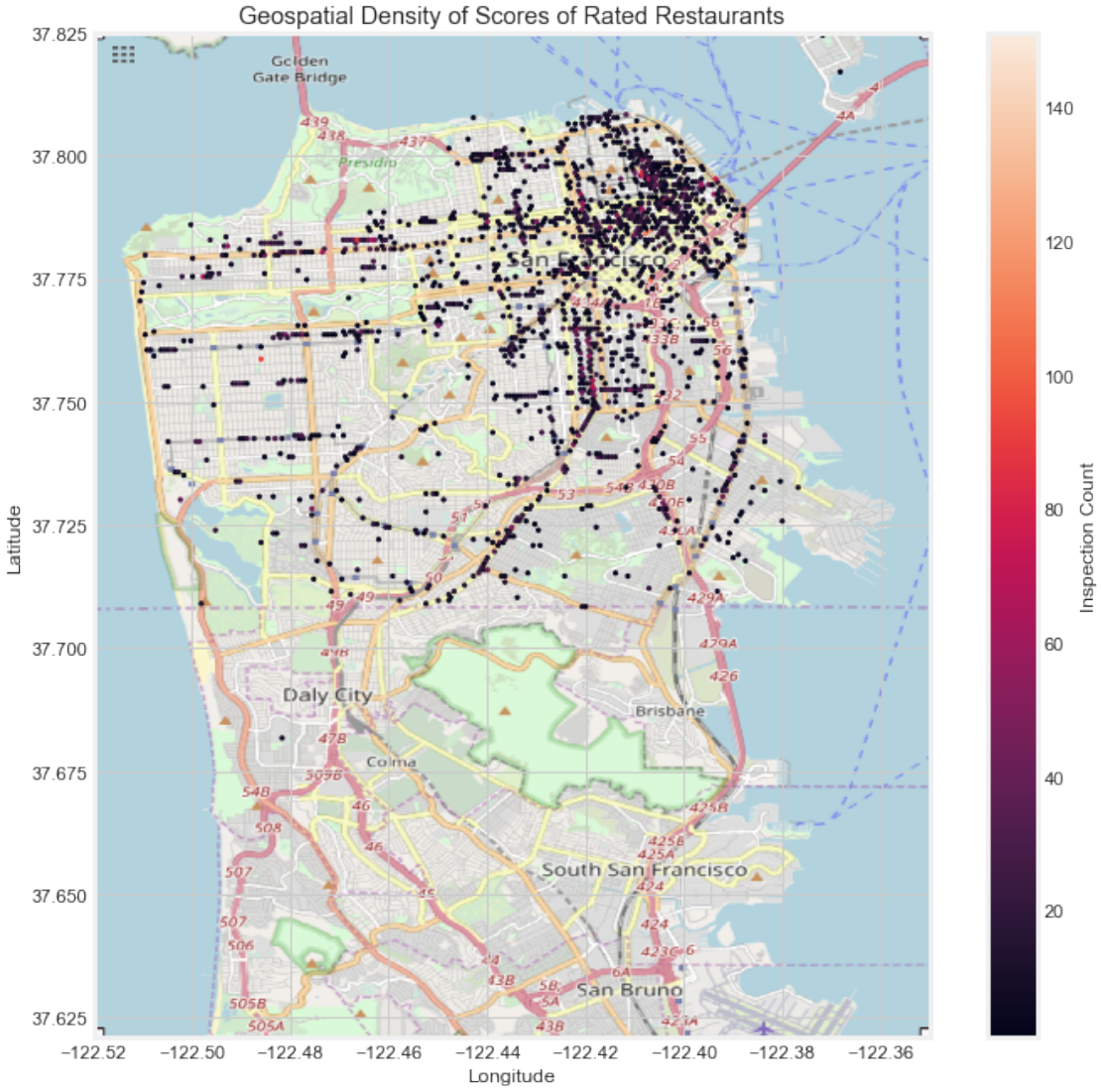

100
90
80
70
60
50
score
risk_category
Low Risk
Moderate Risk
High Risk
2017.0
2018.0
2019.0
year

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

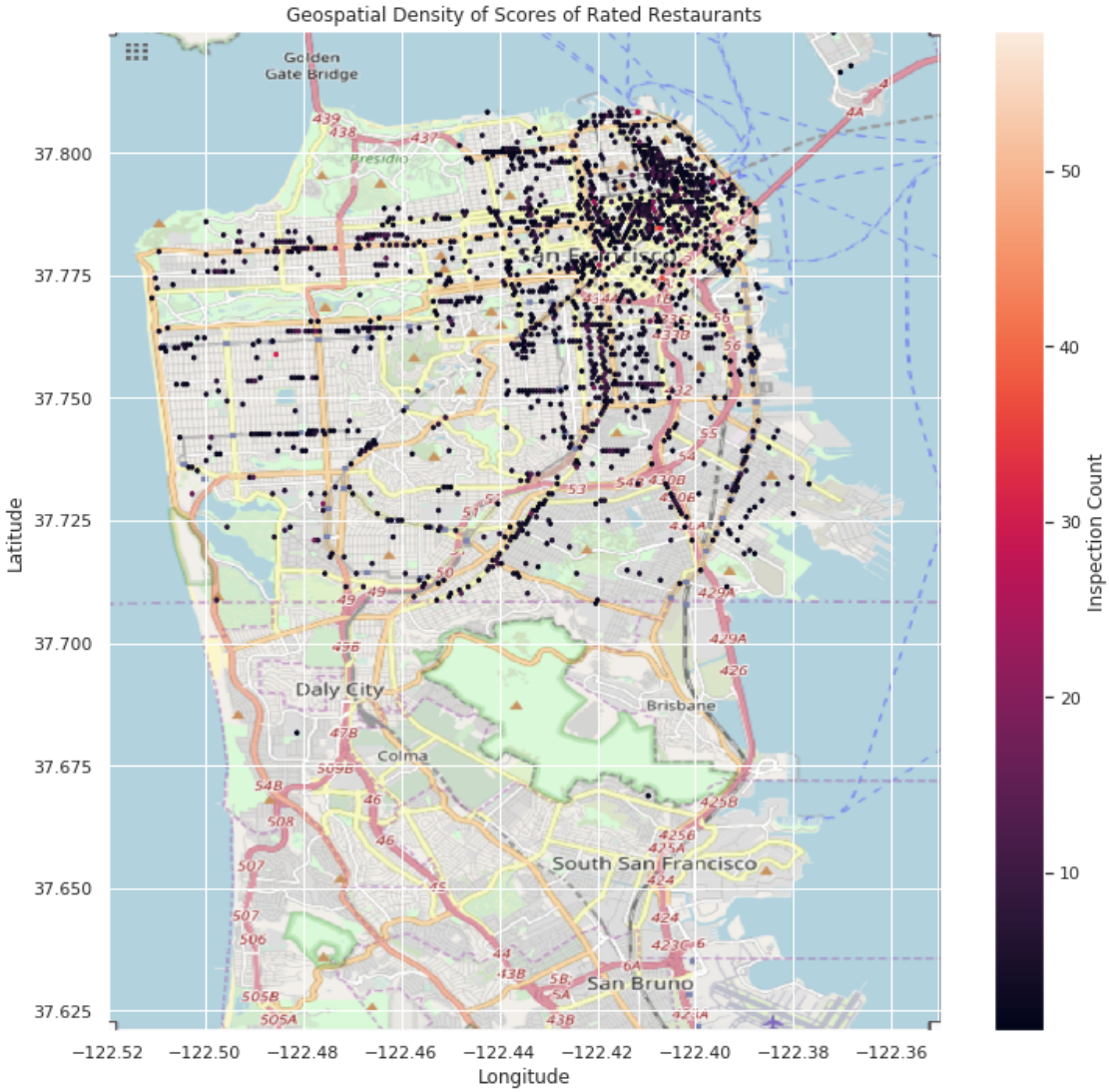
```
max_lat = rated_geo['latitude'].max()
max_score = rated_geo['score'].max()
min_score = rated_geo['score'].min()
bound = ((min_lon, max_lon, min_lat, max_lat))
min_lon, max_lon, min_lat, max_lat
map_bound = ((-122.5200, -122.3500, 37.6209, 37.8249))
# DO NOT MODIFY THIS BLOCK

# 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
x = plt.hexbin(rated_geo['longitude'], rated_geo['latitude'], rated_geo['score'], reduce_C_fu
cb = fig.colorbar(x)
cb.set_label('Inspection Count')
plt.ylabel('Latitude')
plt.xlabel('Longitude')
plt.title('Geospatial Density of Scores of Rated Restaurants')



# 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 Restaurants
Golden Gate Bridge
San Francisco
Daly City
Colma
Brisbane
South San Francisco
San Bruno
Latitude
Longitude
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
Inspection Count
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.

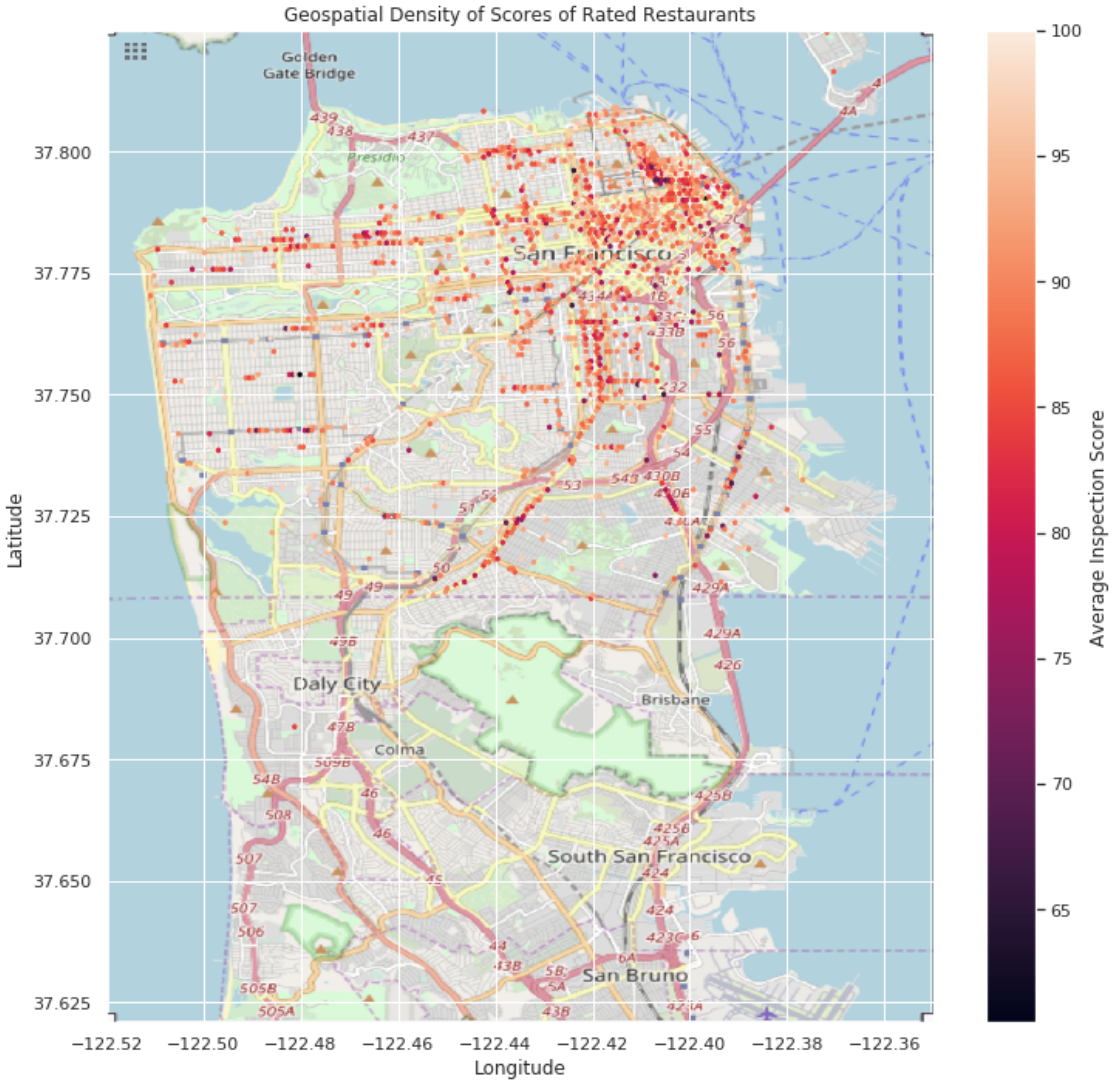
Hint: If you set up everything correctly in 3b, you should only need to change 1 parameter here to produce the plot.

In [100]:
```
# 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
x = plt.hexbin(rated_geo['longitude'], rated_geo['latitude'], rated_geo['score'], reduce_C_fu
cb = fig.colorbar(x)
cb.set_label('Average Inspection Score')
plt.ylabel('Latitude')
plt.xlabel('Longitude')
plt.title('Geospatial Density of Scores of Rated Restaurants')



# 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 Restaurants
Golden Gate Bridge
San Francisco
Daly City
Brisbane
Colma
South San Francisco
San Bruno
Latitude
Longitude
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
Average Inspection Score
100
95
90
85
80
75
70
65

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?

In the area of San Francisco, restaurants are very densely packed together, in this area, we see a few restaurants which are generally lower inspections scores than other areas as there are darker reds and purples. In this area, inspections tend to be the most frequent because of the density of restaurant buildings. The connection seems to be that this area is a much more densely packed part of the city, so it's likely that real estate is much more expensive and it's harder for a business to run operationally without too many health code violations. This would lead to the connection we've identified as restaurants that have a lower inspection score are probably more likely to receive a return inspection to ensure that everything has been brought up to code (or not).

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 [ ]:
```
# YOUR DATA PROCESSING AND PLOTTING HERE

        # YOUR EXPLANATION HERE (in a comment)
```