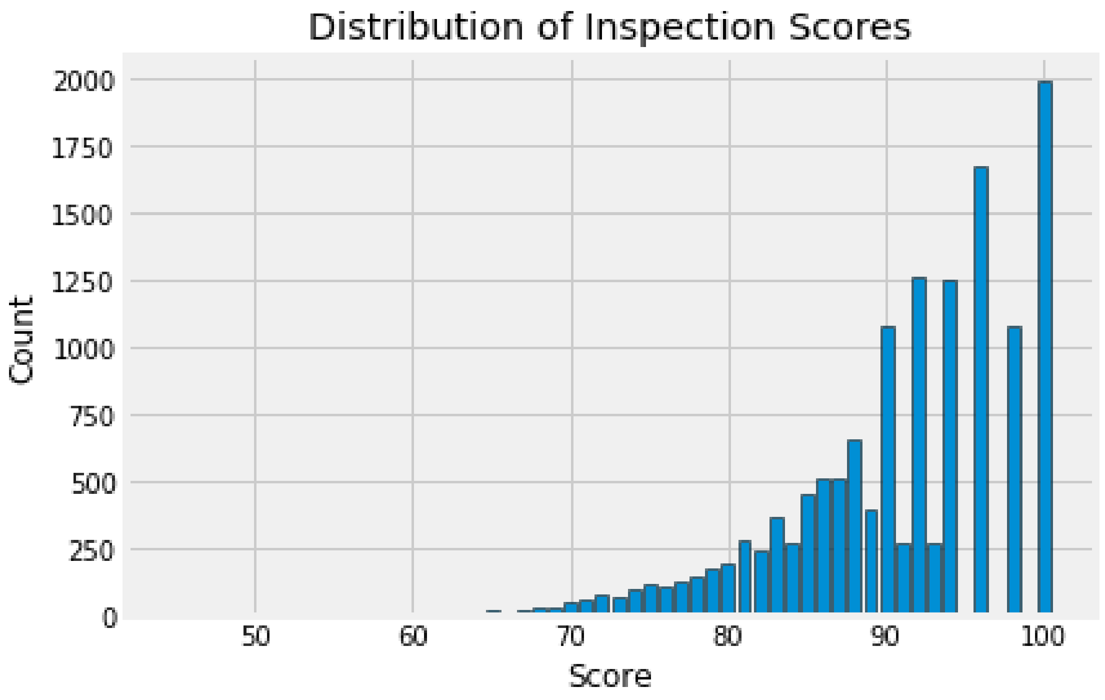
Notebook

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.

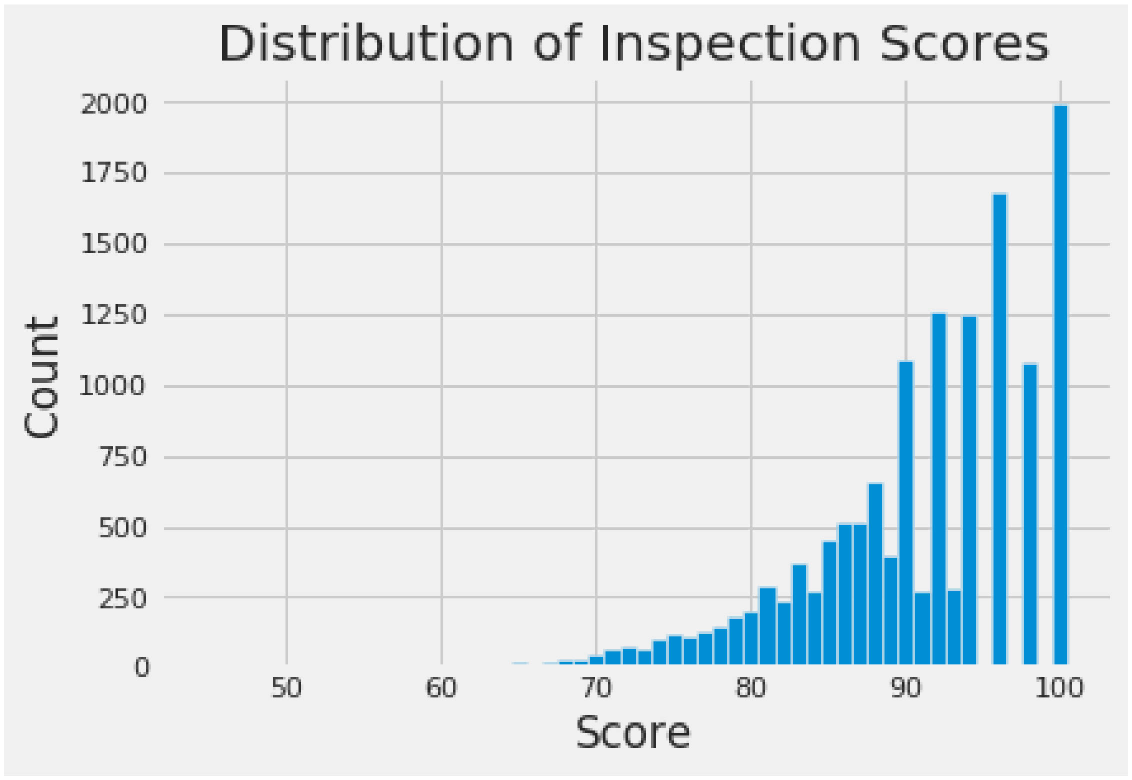


You might find this matplotlib.pyplot tutorial useful. Key syntax that you'll need:

```
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]: ins = ins[ins['score'] > 0]
        y_values = ins['score'].value_counts()
        x_values = y_values.keys()
        plt.bar(x_values, y_values, 1)

        plt.xlabel('Score')
        plt.ylabel('Count')
        plt.title('Distribution of Inspection Scores')
        plt.show()
```


Distribution 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

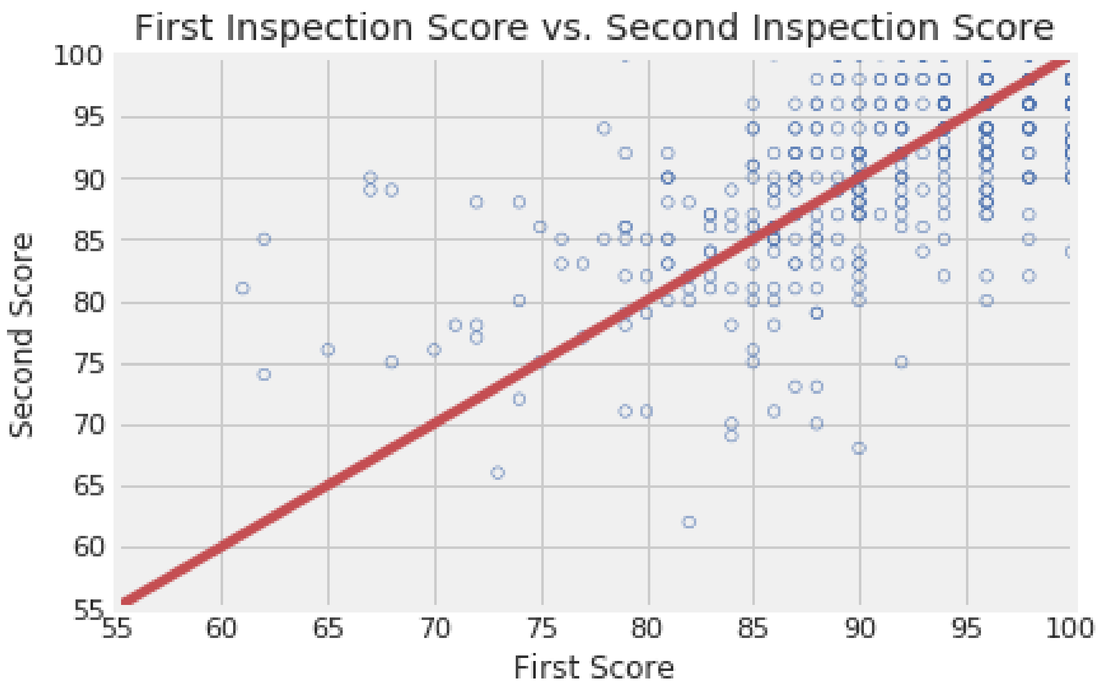
Describe the qualities of the distribution of the inspections scores based on your bar plot. Consider the mode(s), symmetry, tails, gaps, and anomalous values. Are there any unusual features of this distribution? What do your observations imply about the scores?

There are many gaps from scores 90 - 100, which tells us that there are certain scores that were never given, despite the decently large sample size. The left tail values trail off starting at around 75, so most scores are 75 and up. An unusual feature of this distribution is that the most concentrated amount of scores given are above 90, which shows that most scores given to restaurants are 90 and up. The most unusual feature of the distribution is that the score of 100 was given the most, implying that a perfect score is very common to achieve.

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

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.

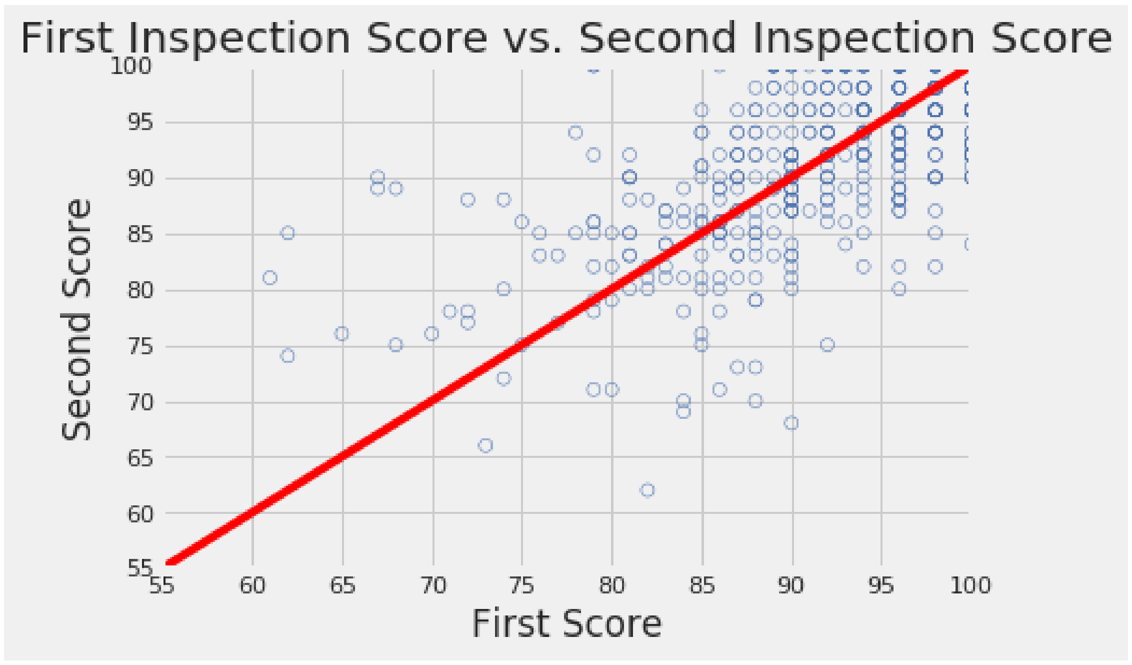
`plt.plot` for the reference line.

`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.

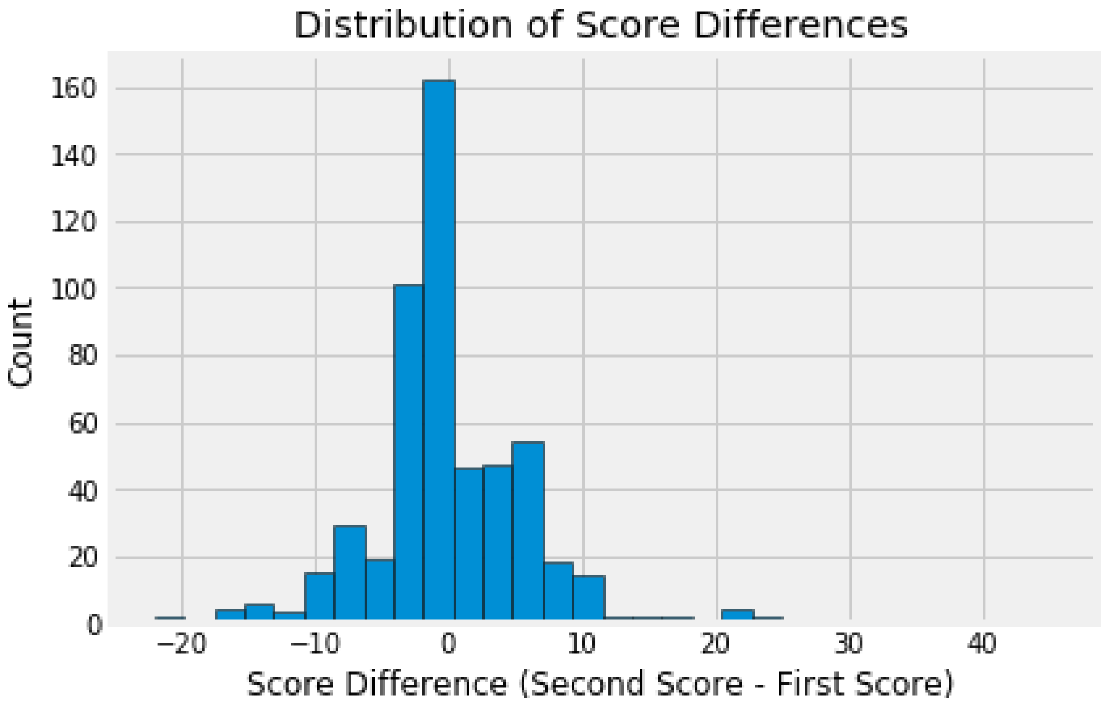
```
In [15]: unzip = list(zip(scores_pairs_by_business['score_pair']))
         first_level = [i[0] for i in unzip]
         x_values = [i[0] for i in first_level]
         y_values = [i[1] for i in first_level]

         plt.scatter(x_values, y_values, facecolors = 'none', edgecolors = 'b')
         plt.xlabel('First Score')
         plt.ylabel('Second Score')
         plt.title('First Inspection Score vs. Second Inspection Score')
         plt.axis((55, 100, 55, 100))
         plt.plot(np.arange(55, 101), np.arange(55, 101), color = 'red')
         plt.show()
```


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

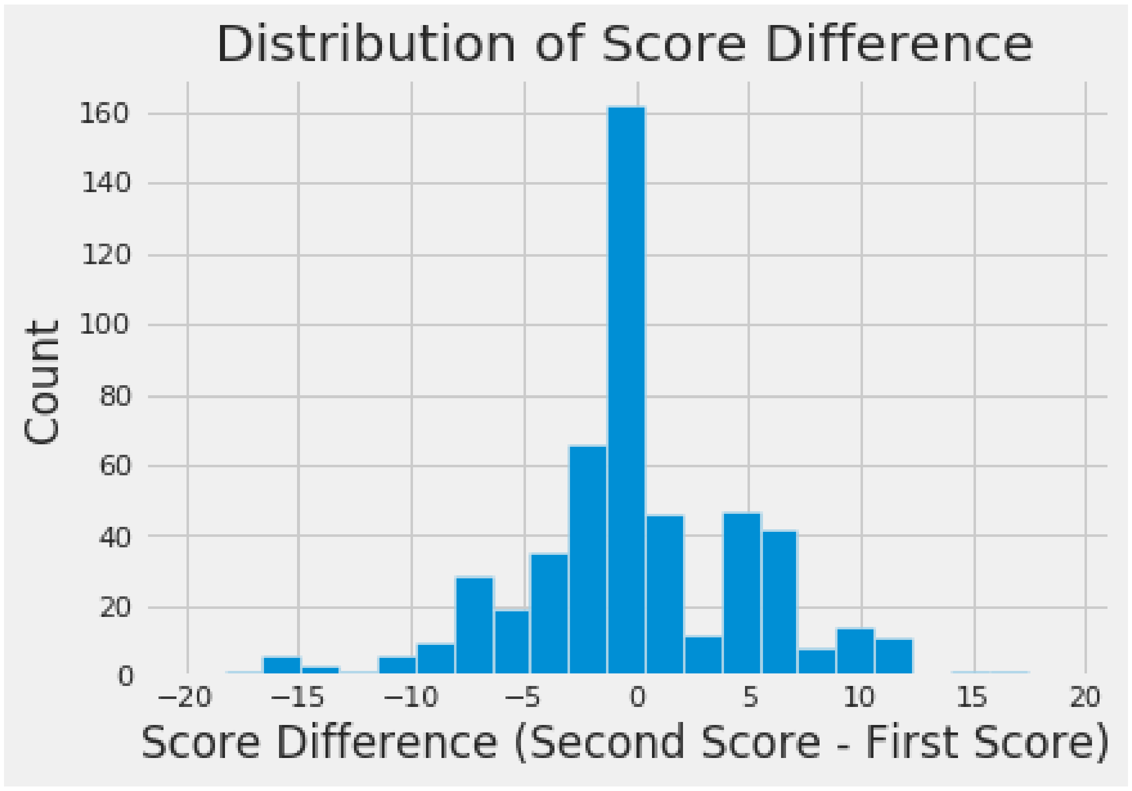
Another way to compare the scores from the two inspections is to examine the difference in scores. Subtract the first score from the second in `scores_pairs_by_business`. Make a histogram of these differences in the scores. We might expect these differences to be positive, indicating an improvement from 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 [16]: unzip = list(zip(scores_pairs_by_business['score_pair']))
         first_level = [i[0] for i in unzip]
         x_values_list = [i[0] for i in first_level]
         y_values_list = [i[1] for i in first_level]

         x_values_array = np.array(x_values_list)
         y_values_array = np.array(y_values_list)
         difference = y_values_array - x_values_array
         plt.hist(difference, bins = np.arange(-20, 20, 1.7))
         plt.xlabel('Score Difference (Second Score - First Score)')
         plt.ylabel('Count')
         plt.title('Distribution of Score Difference')
         plt.show()
```



Distribution of Score Difference
Count
160
140
120
100
80
60
40
20
0
−20
−15
−10
−5
0
5
10
15
20
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?

I would expect to see more dots above the reference line than below, since the slope of the reference line plots scores that are equal in the first and second inspection. Looking at the scatter plot, my observations are that about an equal amount of restaurants did much better, and about an equal amount also did much worse when comparing the second and first scores. My observations are not consistent with my expectations since I expected much more plots above the reference line.

0.0.5 Question 2f

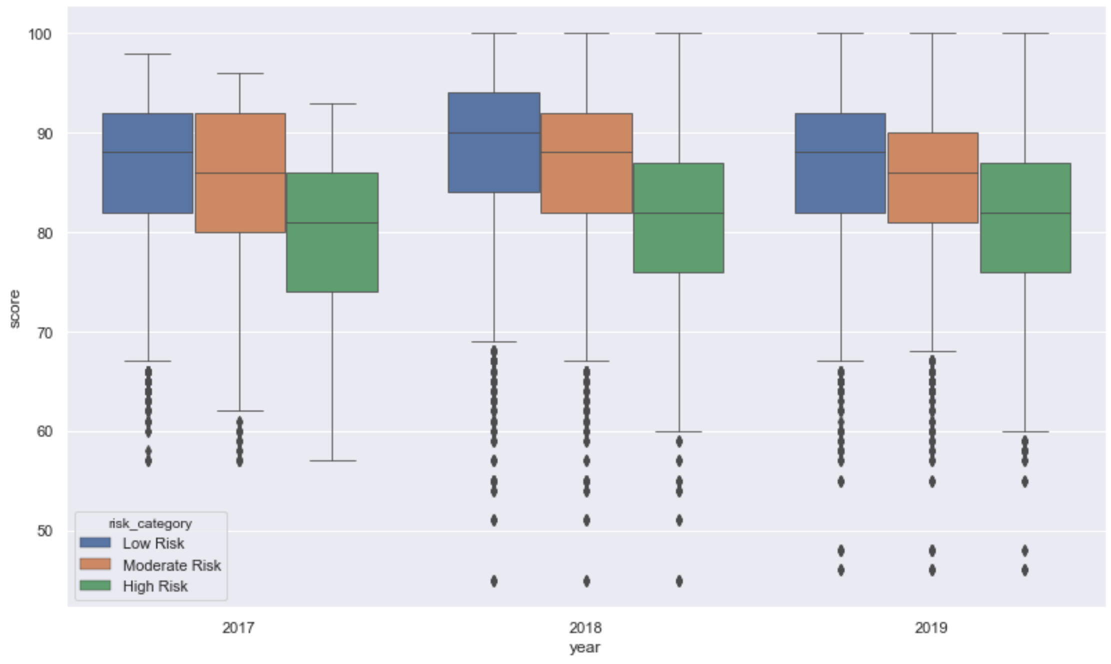
If a restaurant's score improves from the first to the second inspection, how would this be reflected in the histogram of the difference in the scores that you made in question 2d? What do you oberve from the plot? Are your observations consistent with your expectations? Explain your observations in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

An improvement from the first to the second inspection would be reflected as a distribution centered somewhere to the right of zero of the histogram, since positive x values means that the second score was higher than the first score. However, the histogram shows a distribution that is centered on 0, which means that most restaurants did not change much from their first scores. The spread of the distribution is more or less equal, both ending about 20 units away from the center. The deviation is about 4 units, which means that the distribution of scores is pretty tightly centered on zero. This is not consistent with my expectations, since an improvement of scores would mean that the deviation is more loose as scores are scattered in the positive range rather than being centered on zero.

0.0.6 Question 2g

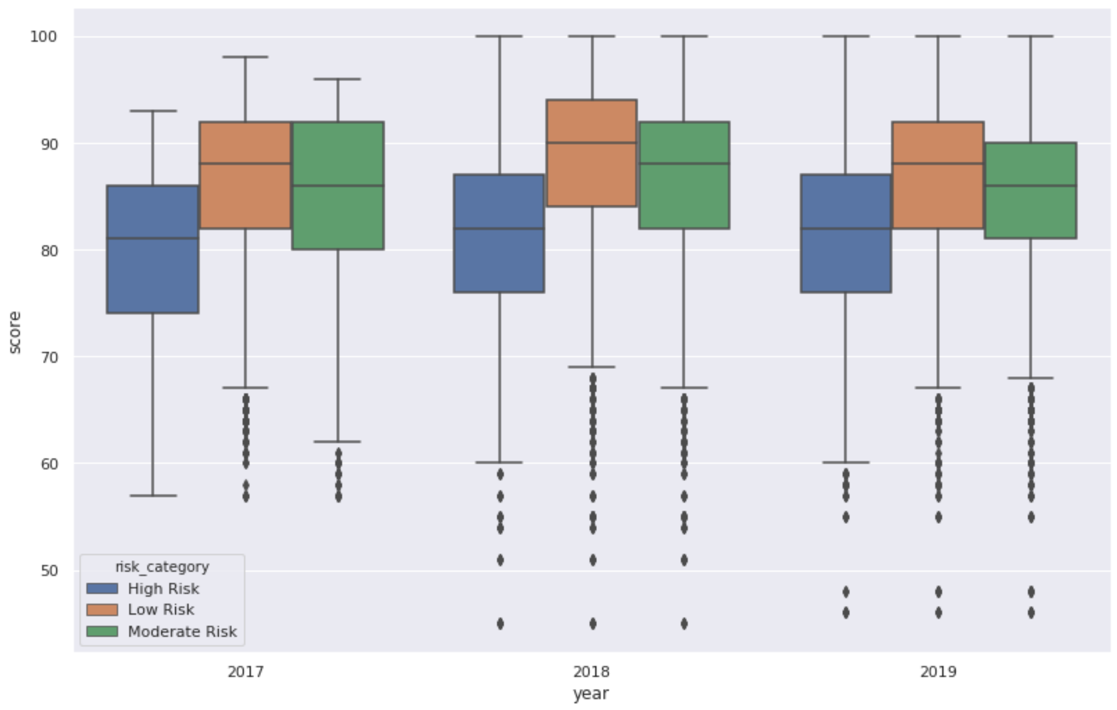
To wrap up our analysis of the restaurant ratings over time, one final metric we will be looking at is the distribution of restaurant scores over time. Create a side-by-side boxplot that shows the distribution of these scores for each different risk category from 2017 to 2019. Use a figure size of at 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 [17]: # Do not modify this line
         sns.set()

         first_merge = ins.merge(ins2vio)
         second_merge = first_merge.merge(vio)
         second_merge = second_merge[second_merge['year'] > 2016]

         plt.figure(figsize = (12, 8))
         sns.boxplot(x = 'year', y = 'score', hue = 'risk_category', data = second_merge)

Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x7fdb0dbc10f0>
```



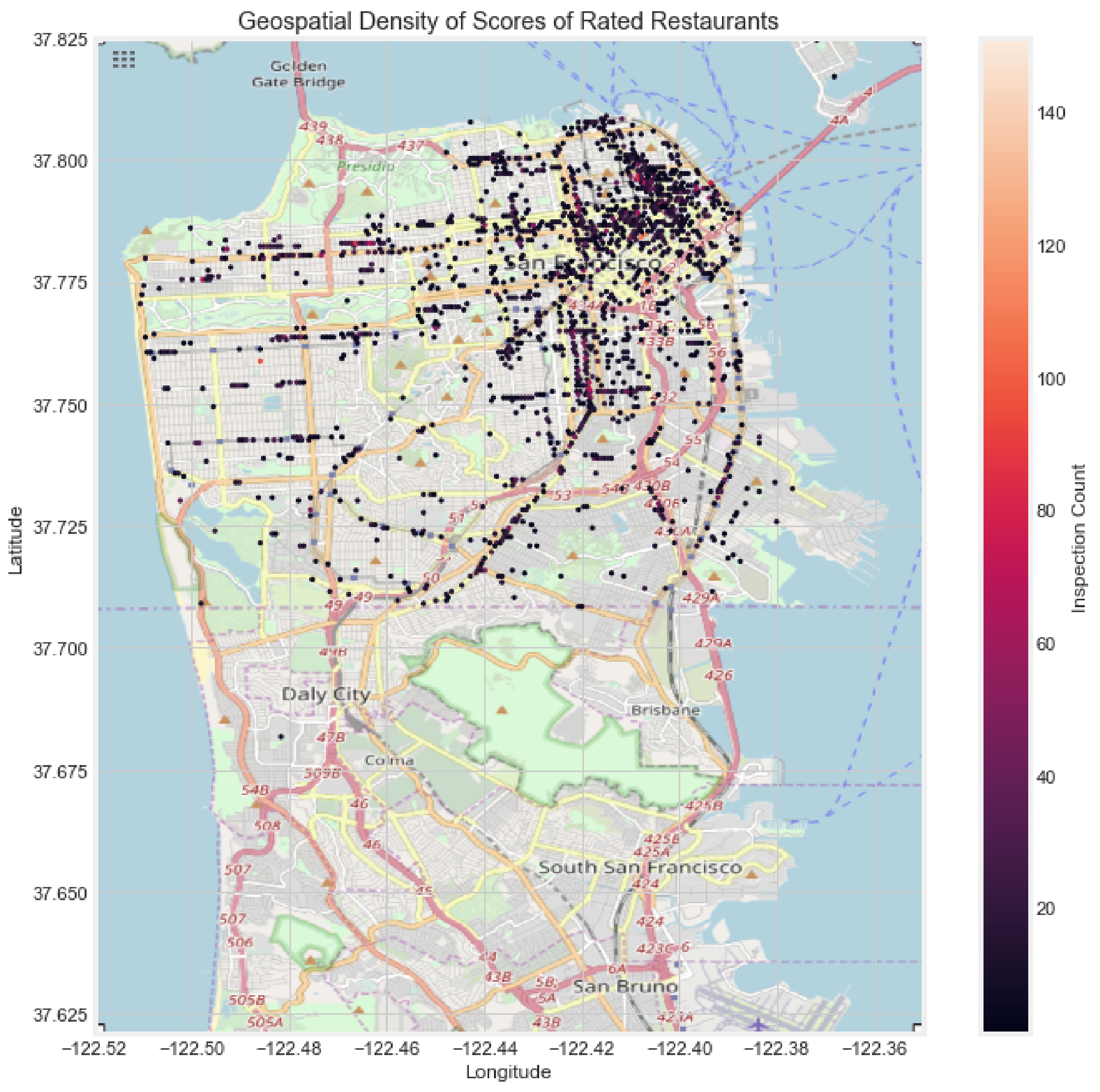
100
90
80
70
60
50
score
risk_category
High Risk
Low Risk
Moderate Risk
2017
2018
2019
year

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.

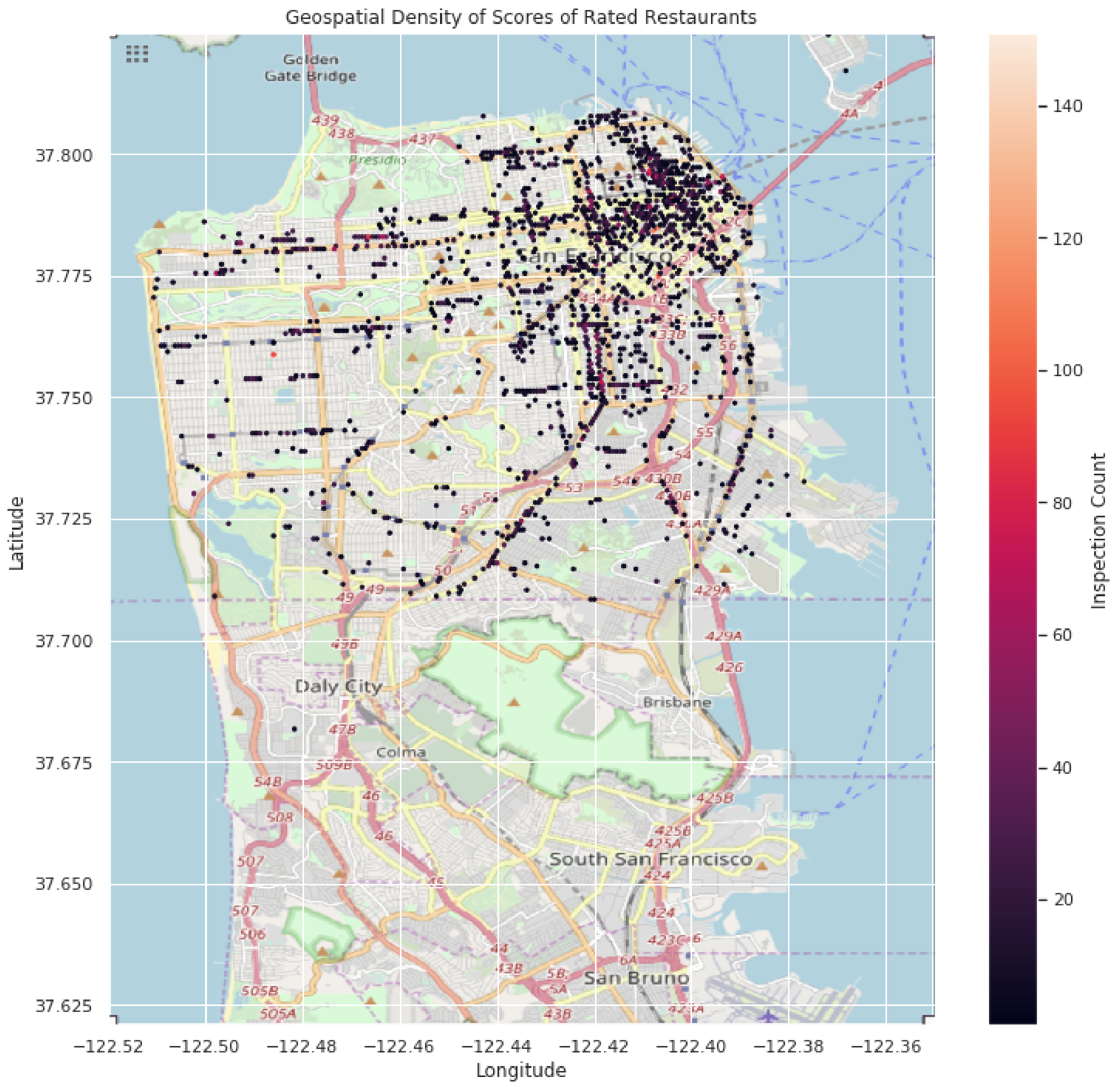
Hint: For the 2 functions we mentioned above, try looking at the parameter `reduce_C_function`, which determines the aggregate function for the hexbin plot.

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 [20]: # 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
rated_geo = rated_geo[rated_geo['longitude'] != 0]
rated_geo = rated_geo[rated_geo['latitude'] != 0]
x = rated_geo['longitude']
y = rated_geo['latitude']
big_c = rated_geo['score']
hexbin = plt.hexbin(x, y, C = big_c, gridsize = 200, reduce_C_function = np.size)
colorbar= fig.colorbar(hexbin)

plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.title('Geospatial Density of Scores of Rated Restaurants')
colorbar.set_label('Inspection Count')


# 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
```



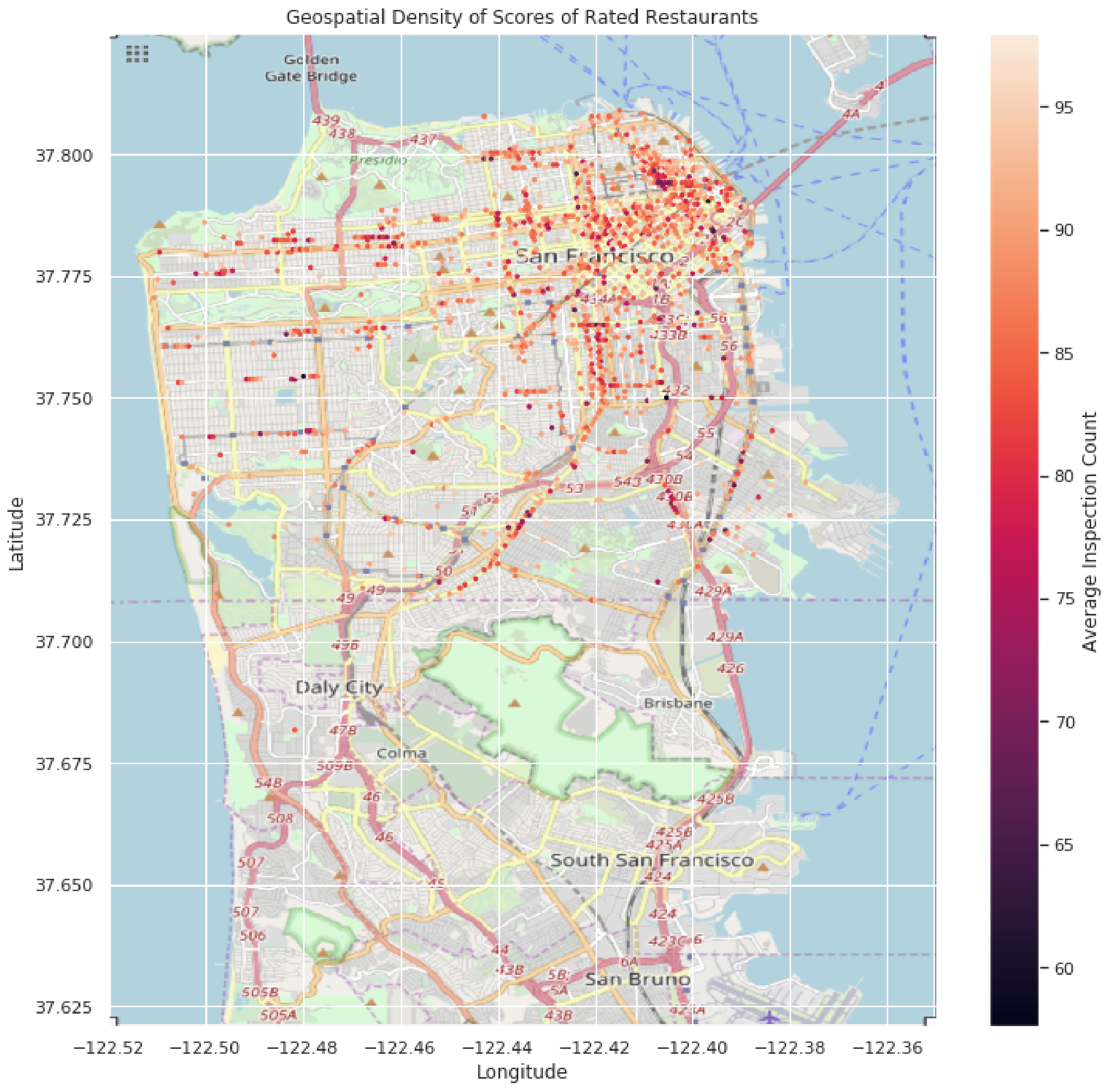
Geospatial Density of Scores of Rated Restaurants
Latitude
Longitude
Inspection Count
37.800
37.775
37.750
37.725
37.700
37.675
37.650
37.625
−122.52
−122.50
−122.48
−122.46
−122.44
−122.42
−122.40
−122.38
−122.36
140
120
100
80
60
40
20
Golden Gate Bridge
Presidio
San Francisco
Daly City
Colma
Brisbane
South San Francisco
San Bruno

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 [21]:
```
# 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
rated_geo = rated_geo[rated_geo['longitude'] != 0]
rated_geo = rated_geo[rated_geo['latitude'] != 0]
x = rated_geo['longitude']
y = rated_geo['latitude']
big_c = rated_geo['score']
hexbin = plt.hexbin(x, y, C = big_c, gridsize = 200, reduce_C_function = np.mean)
colorbar= fig.colorbar(hexbin)

plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.title('Geospatial Density of Scores of Rated Restaurants')
colorbar.set_label('Average Inspection Count')


# 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
Latitude
Longitude
Average Inspection Count
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
95
90
85
80
75
70
65
60
Golden Gate Bridge
San Francisco
Daly City
Colma
Brisbane
South San Francisco
San Bruno

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?

I noticed that tourist attractions, such as Chinatown and Union Square, tends to have more frequent inspections. It is also in these areas where the average inspection score tends to be lower than other locations. These locations also have more frequent inspections, which may lead to a lower average inspection score. These lower inspection scores may be due to the constant influx of customers which causes the restaurant staff to overlook or occasionally forget certain procedures that may have lead to lower inspection scores.

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:

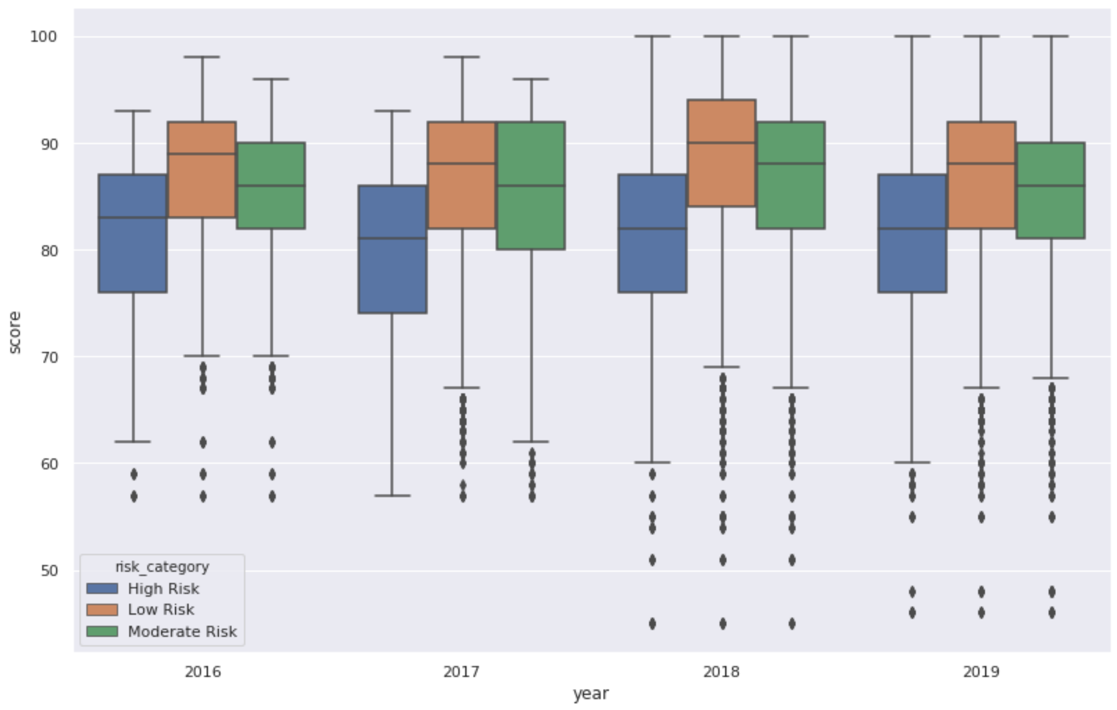
- **Great** (4-5 points): The chart is well designed, and the data computation is correct. The text written articulates a reasonable metric and correctly describes the relevant insight and answer to the question you are interested in.
- **Passing** (3-4 points): A chart is produced but with some flaws 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 [22]: # YOUR DATA PROCESSING AND PLOTTING HERE
         sns.set()
         first_merge = ins.merge(ins2vio)
         second_merge = first_merge.merge(vio)
         plt.figure(figsize = (12, 8))
         sns.boxplot(x = 'year', y = 'score', hue = 'risk_category', data = second_merge)

         # YOUR EXPLANATION HERE (in a comment)
         # This visualization is very interesting becauase it shows that from 2016 to 2019, the distrib
         # the high risk category did not change much despite the influx of new construction in 2017. H
         # low risk scores peaked before falling back down to similar score averages in 2016 and 2017, 
         # moderate category scores compared to the year before in 2016. Seeing the distribution throug
         # that the distribution of scores between the risk categories did not change much.

Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x7fdb0daa2828>
```


100
90
80
70
60
50
score
risk_category
High Risk
Low Risk
Moderate Risk
2016
2017
2018
2019
year