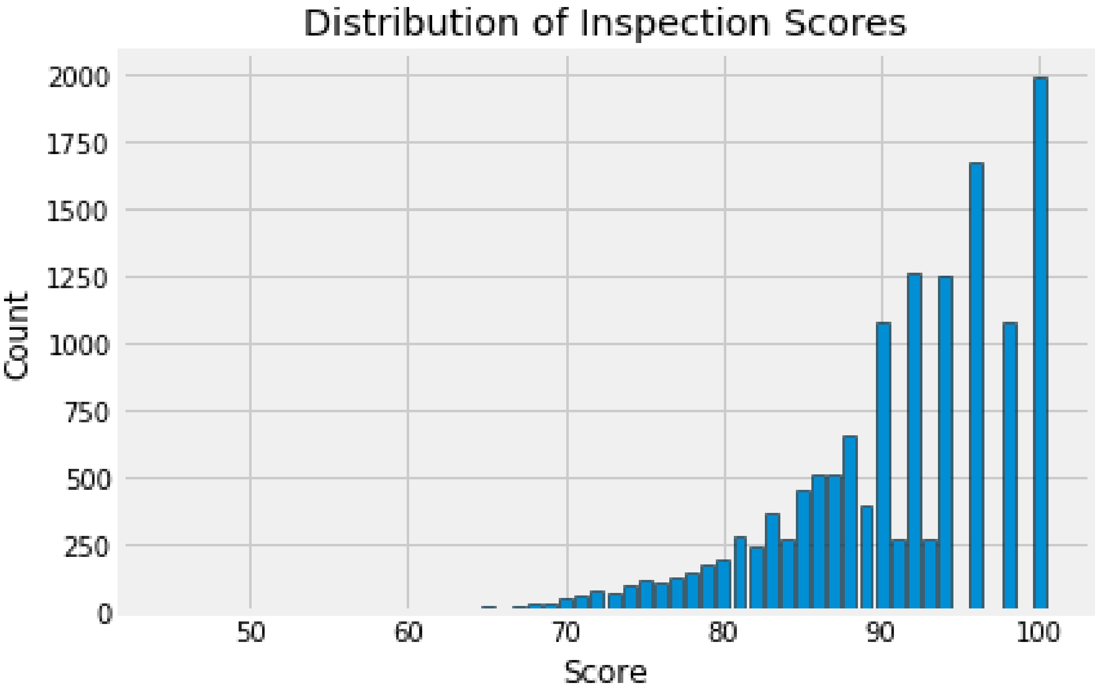
Notebook

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.

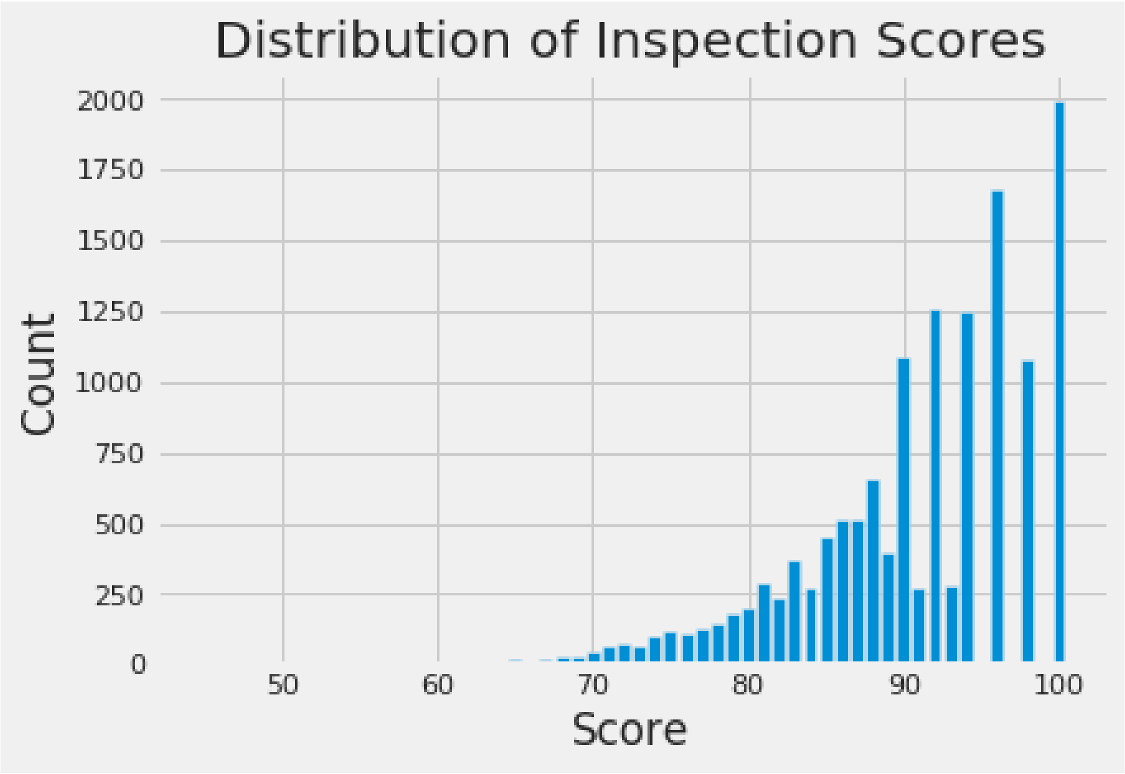


You might find this matplotlib.pyplot tutorial useful. Key syntax that you'll need:

```
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]: scores = ins[ins['score'] != -1]
        scores = scores['score'].value_counts().sort_index()
        plt.bar(scores.index, scores.values)
        plt.xlabel('Score')
        plt.ylabel('Count')
        plt.title('Distribution of Inspection Scores');
```


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

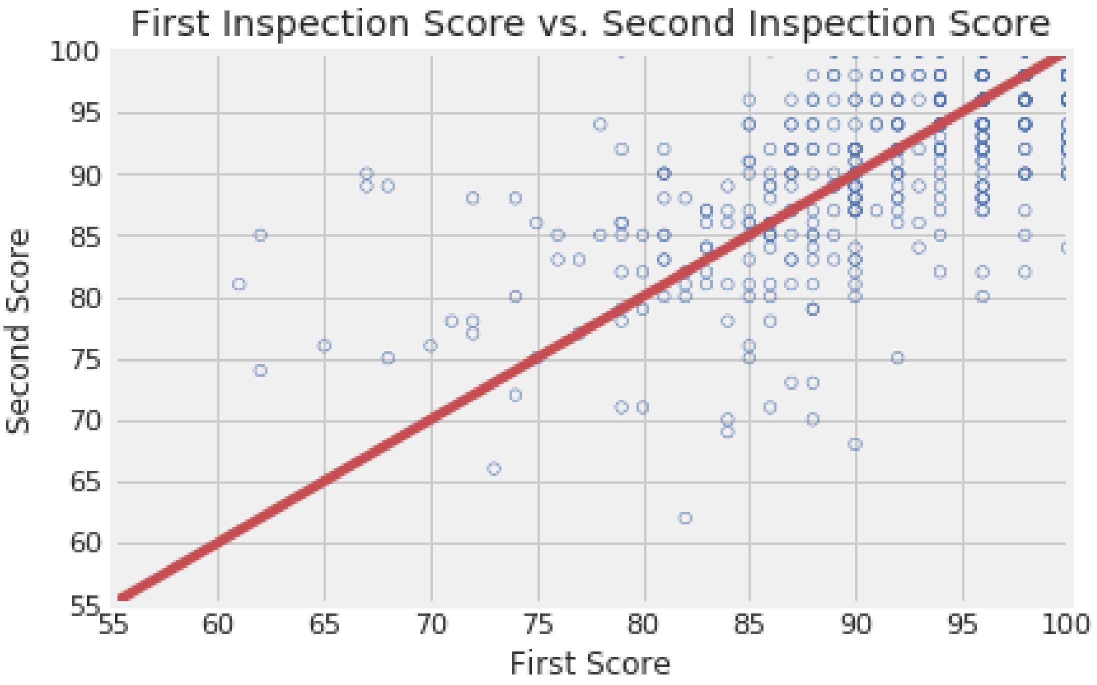
Describe the qualities of the distribution of the inspections scores based on your bar plot. Consider the mode(s), symmetry, tails, gaps, and anomalous values. Are there any unusual features of this distribution? What do your observations imply about the scores?

The qualities of the distribution of the inspections scores are as follows: 1. The main mode of the distribution is 100. 2. This is NOT a symmetric distribution. 3. There is a high frequency of gaps between the range 90 and 100. 4. There is a left tail

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.

The restaurant with the lowest inspection scores ever is ”Lollipot” recieving an inspection score of 45.

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.

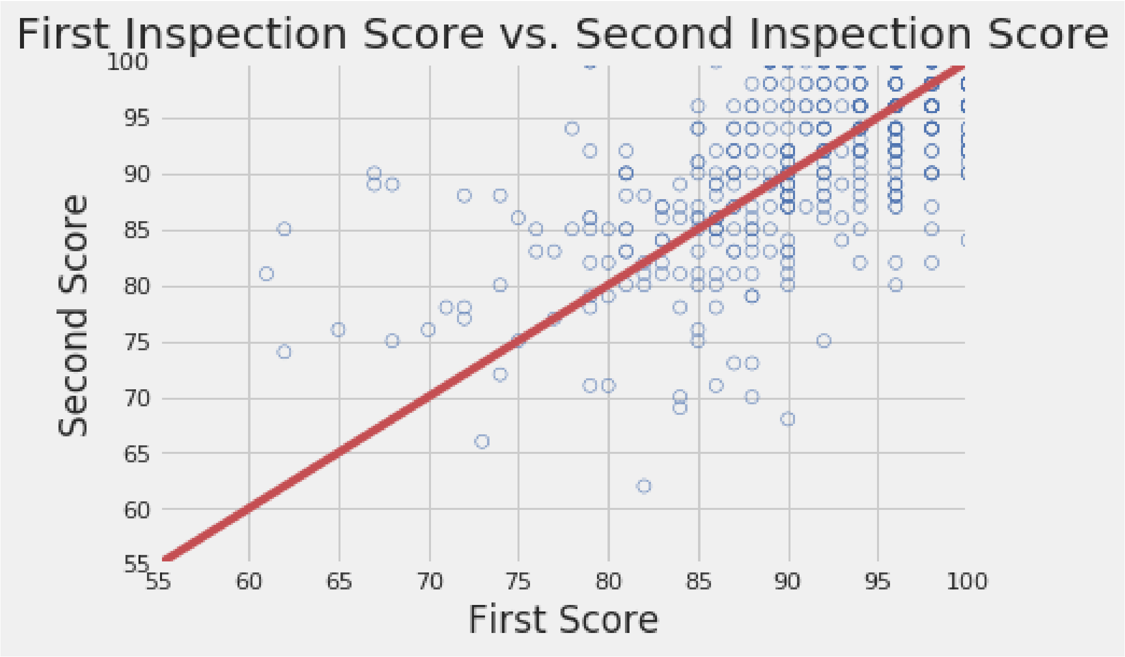
`plt.plot` for the reference line.

`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.

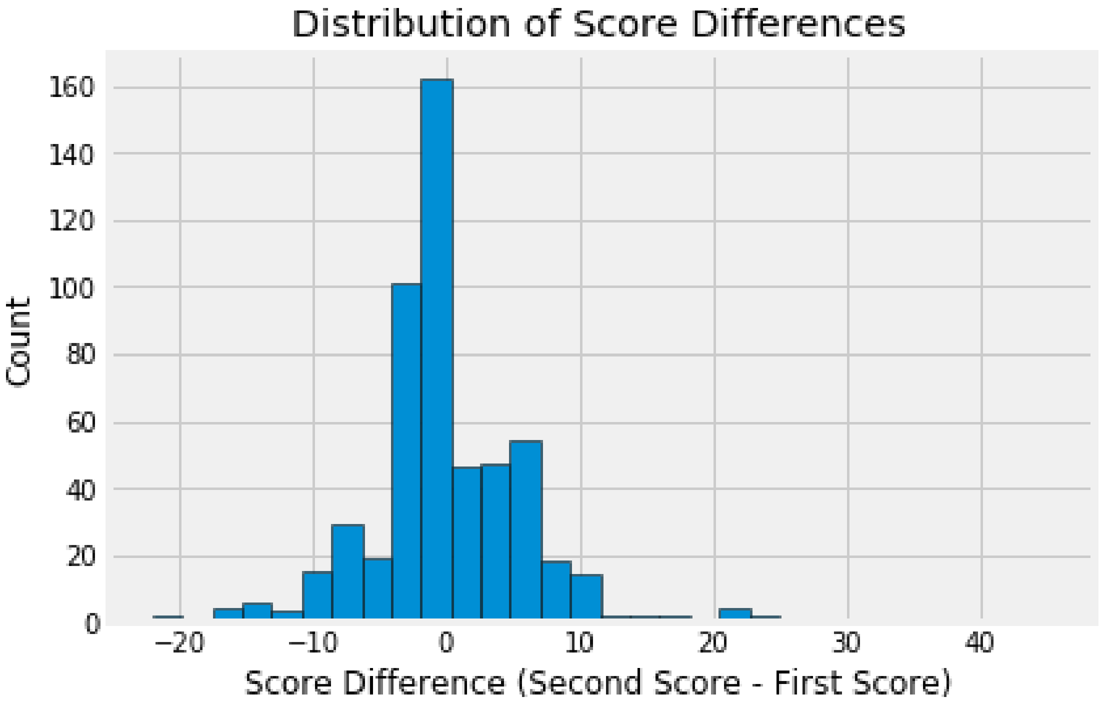
```
In [15]: first_score = list(zip(*scores_pairs_by_business['score_pair'].tolist()))[0]
         second_score = list(zip(*scores_pairs_by_business['score_pair'].tolist()))[1]
         plt.scatter(first_score, second_score, facecolors='none', edgecolors='b')
         plt.plot(np.arange(55, 101), np.arange(55, 101), color = '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
100
95
90
85
80
75
70
65
60
55
First Score
55
60
65
70
75
80
85
90
95
100

0.0.3 Question 2d

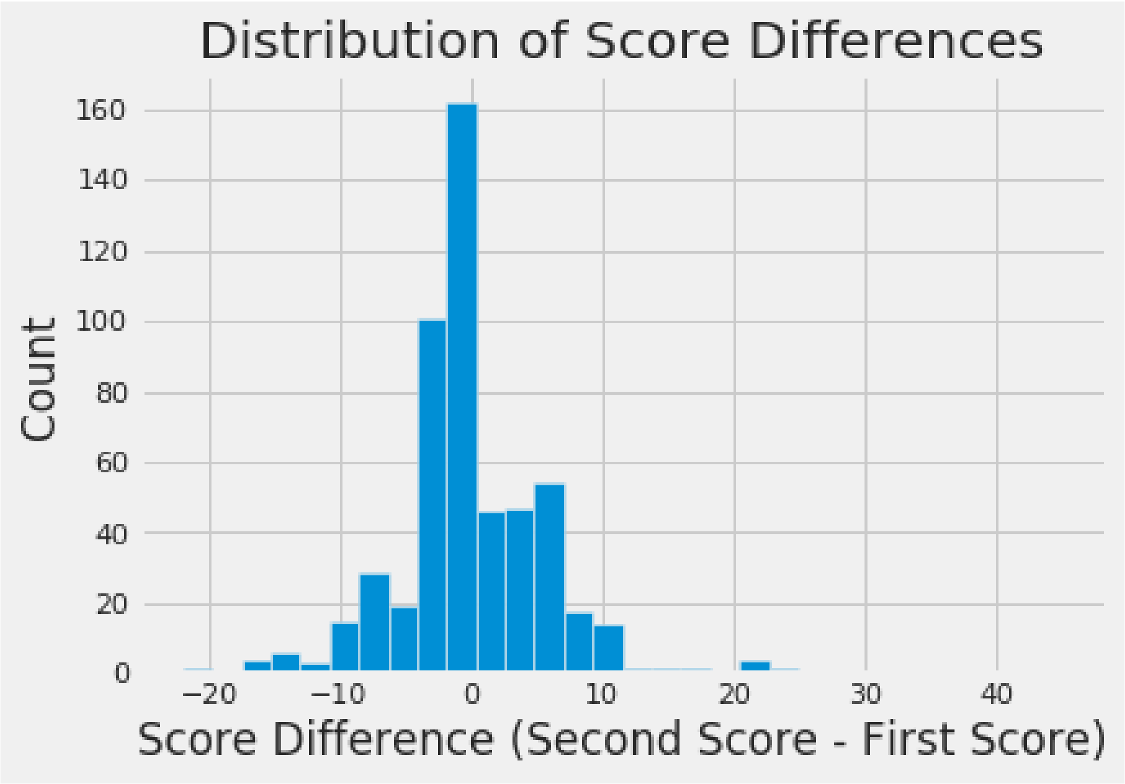
Another way to compare the scores from the two inspections is to examine the difference in scores. Subtract the first score from the second in `scores_pairs_by_business`. Make a histogram of these differences in the scores. We might expect these differences to be positive, indicating an improvement from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


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 restaurants' scores tend to improve from the first to the second inspection, I would expect to see those restaurants represented by a dot to have a higher y-value than x-value. From the scatter plot I observed that restaurants that improved from their first to their second inspection were above the line of regression (fixed at slope 1). I observed that restaurants whom did not have a change in score from their first to their second inspection would be on the line of regression and restaurants whom did not improve from their first to their second inspection are represented by dots that are located below the regression line.

0.0.5 Question 2f

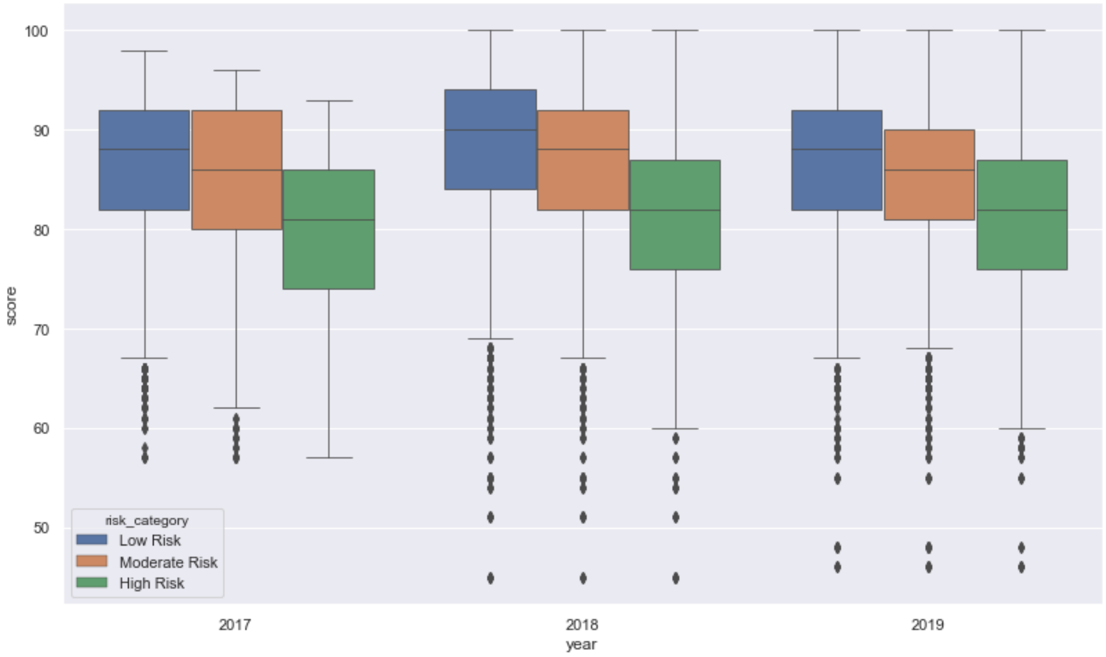
If a restaurant's score improves from the first to the second inspection, how would this be reflected in the histogram of the difference in the scores that you made in question 2d? What do you oberve from the plot? Are your observations consistent with your expectations? Explain your observations in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

If a restaurant's score improves from the first to the second inspection, this would be reflected on the histogram as the bars that represent intervals of positive integers (which denotes the restaurants that have improved from their first to their second inspection). I observed from the histogram that there was a large count of restuarants who had the same score for both the first and the second inspection or who had a reduction of one point on their second inspection. From the histogram we see that the mode is 160 (which are restuarants who had no change in inspection score from their first to second or had a reduction of a point in their inspection score from their first to second). The histogram is skewed to the left and thus not symmetric. The histogram has some outliers visible for restuarants that improved tremendously in their scores. As well the histogram does not look normalish rather has a long left tail.

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:



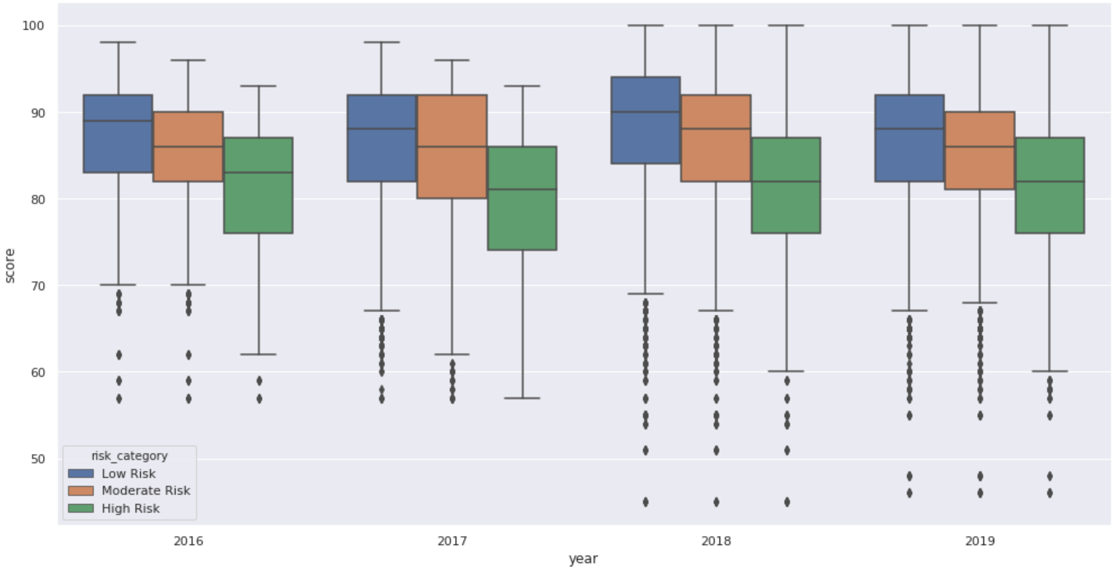
Hint: Use `sns.boxplot()`. Try taking a look at the first several parameters.

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

```
In [17]: # Do not modify this line
         sns.set()

         descrip = vio.merge(ins2vio, how = 'inner')
         year_score = descrip.merge(ins, how = 'inner')
         year_des_score = year_score[['risk_category', 'year', 'score']]
         filter_year_des_score = year_des_score[year_des_score['score'] != -1]
         plt.figure(figsize = (15.0, 8.0))
         sns.boxplot(data=filter_year_des_score, x = 'year', y='score', hue = 'risk_category',
                     hue_order = ['Low Risk', 'Moderate Risk', 'High Risk'])

Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x7fdfc1df3ef0>
```

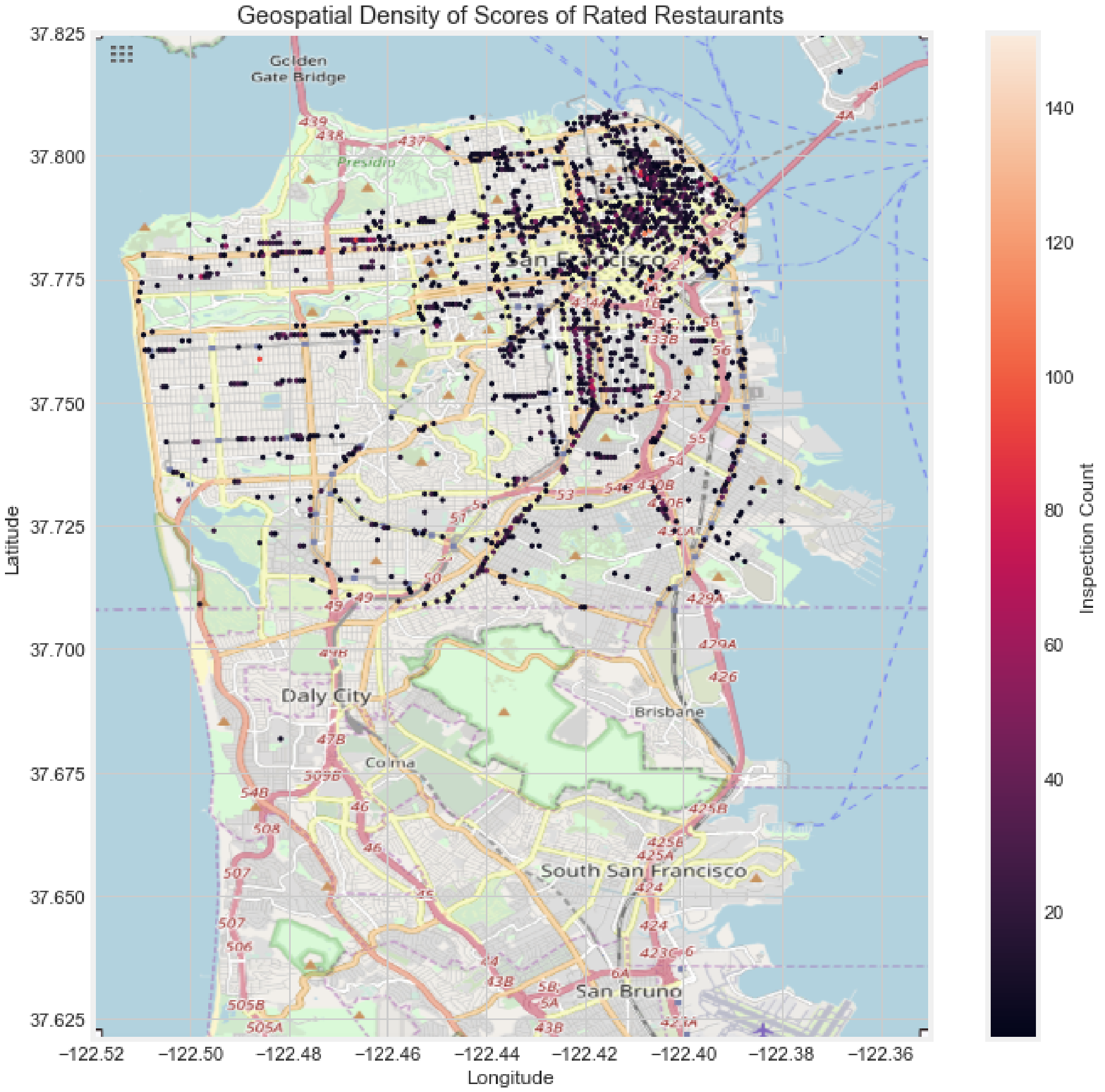

100
90
80
70
60
50
score
risk_category
Low Risk
Moderate Risk
High Risk
2016
2017
2018
2019
year

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.

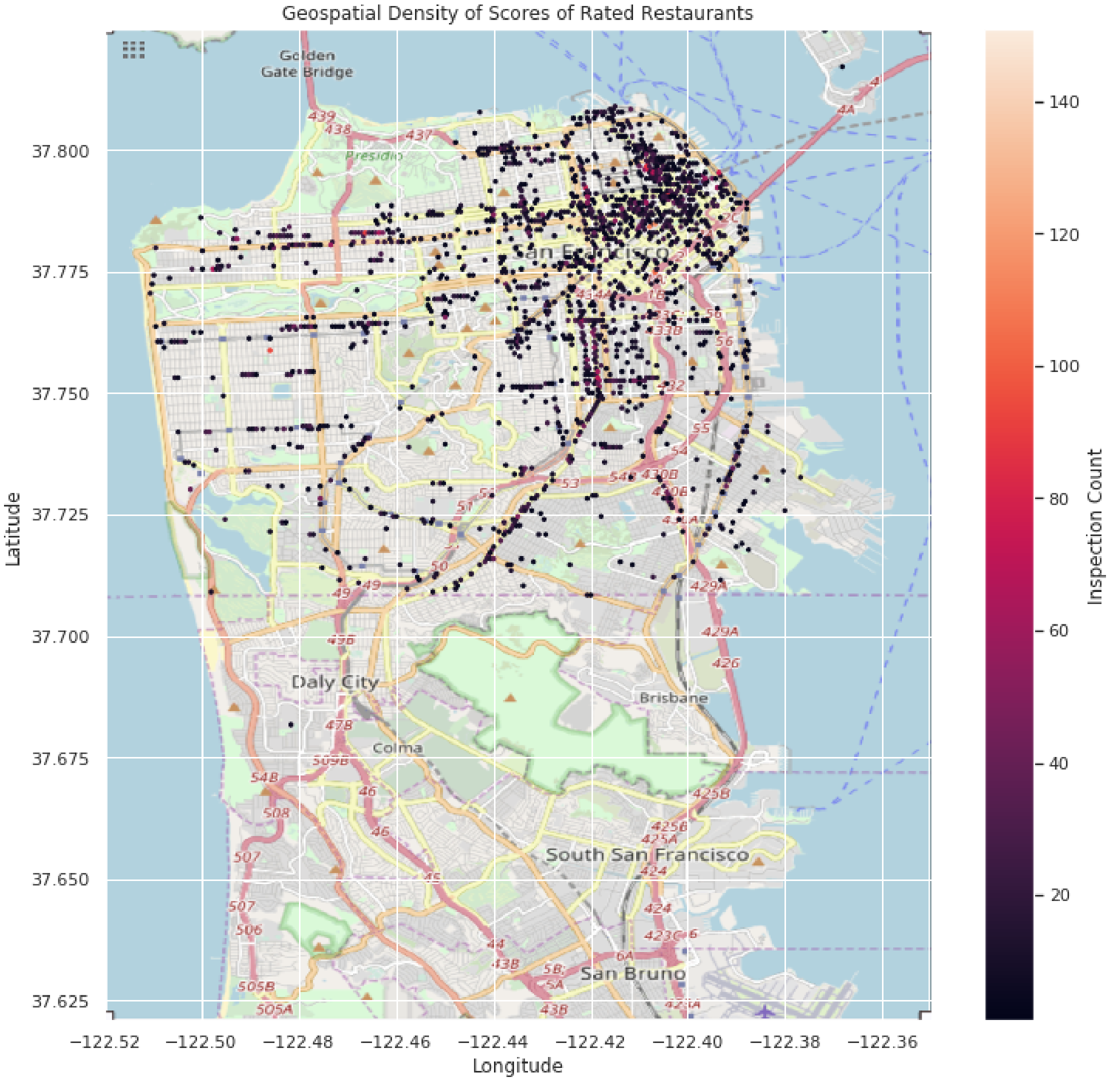
Hint: Try using a `gridsize` of 200 when creating your hexbin plot; it makes the plot cleaner.

```
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
plt.hexbin(x =rated_geo['longitude'], y = rated_geo['latitude'], C = rated_geo['score'],
           reduce_C_function = np.size, gridsize = 200)
plt.colorbar().set_label('Inspection Count')
plt.title('Geospatial Density of Scores of Rated Restaurants')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
# 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

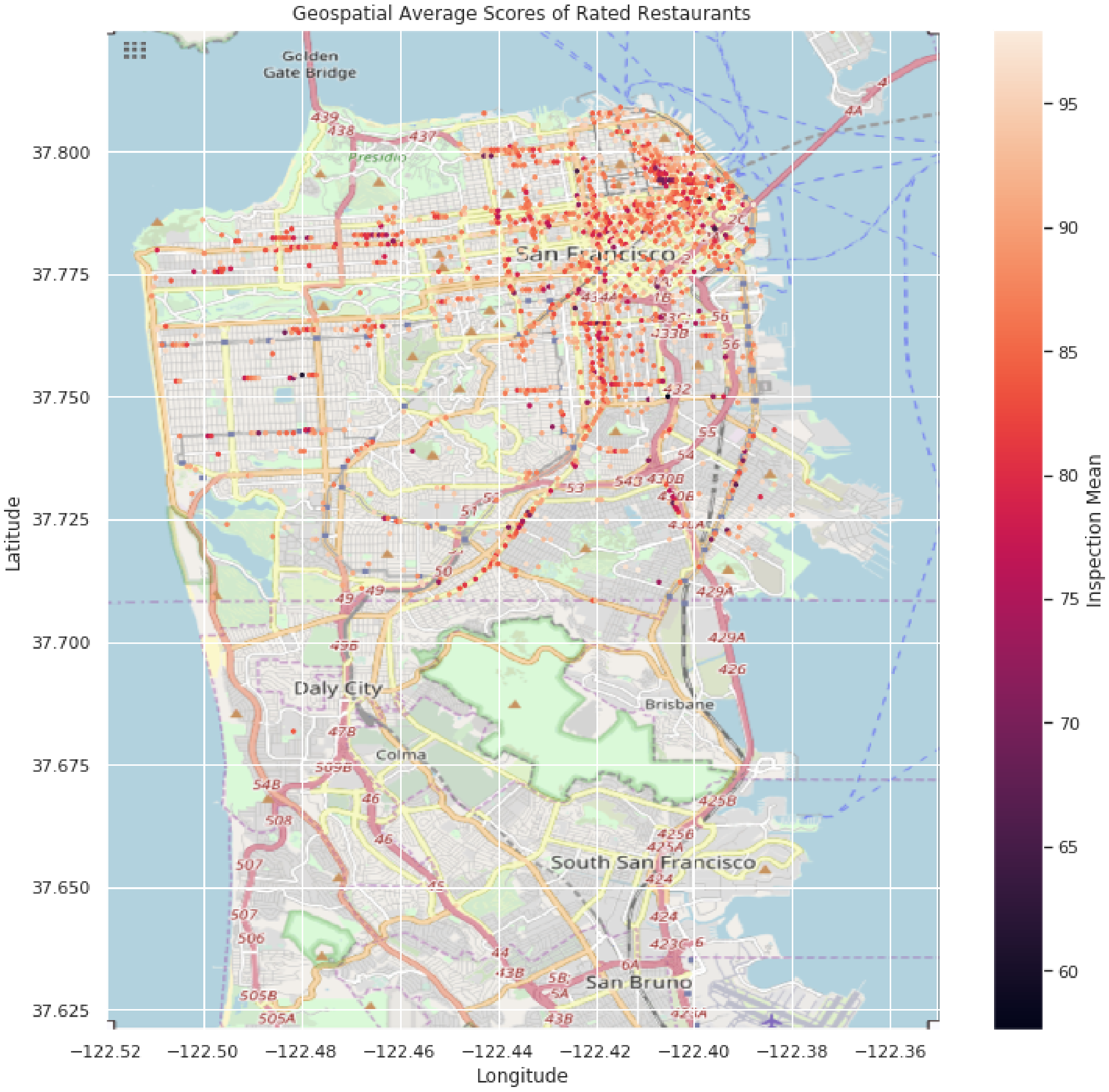
Now that we've created our geospatial hexbin plot for the density of inspection scores for restaurants in San Francisco, let's also create another hexbin plot that visualizes the **average inspection scores** for restaurants in San Francisco.

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
         plt.hexbin(x =rated_geo['longitude'], y = rated_geo['latitude'], C = rated_geo['score'],
                    reduce_C_function = np.mean, gridsize = 200)
         plt.colorbar().set_label('Inspection Mean')
         plt.title('Geospatial Average Scores of Rated Restaurants')
         plt.xlabel('Longitude')
         plt.ylabel('Latitude')


         # 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 Average Scores of Rated Restaurants
Latitude
37.800
37.775
37.750
37.725
37.700
37.675
37.650
37.625
Longitude
−122.52
−122.50
−122.48
−122.46
−122.44
−122.42
−122.40
−122.38
−122.36
Inspection Mean
95
90
85
80
75
70
65
60
Golden Gate Bridge
Presidio
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?

In the first hexbin plot (where we aggregated over the inspection count) I noticed that there are a small visible quantity of locations where the inspections tend to be more frequent; however, it is quite difficult to tell being that the large quantity of black dots scattered over the maps. This could lead to the possibility of those dots covering the other red/orange tone dots representing a higher inspection count. On the other hand, for the second hexbin plot (where we aggregate over the inspection mean) I noticed that there was a small portion of locations where the average inspection score tended to be low. Regarding the 2 hexbin plots above it seems that there is a connection among the frequency of inspections and the average inspection score. In fact it can be depicted by the 2 plots above that inspection scores are more frequent for the locations that have a low inspection score on average. This is understandable because if a restaurant has low inspection scores then they will have more frequent inspection visits in order to push the restaurant into a clearer and healthier state.

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)!

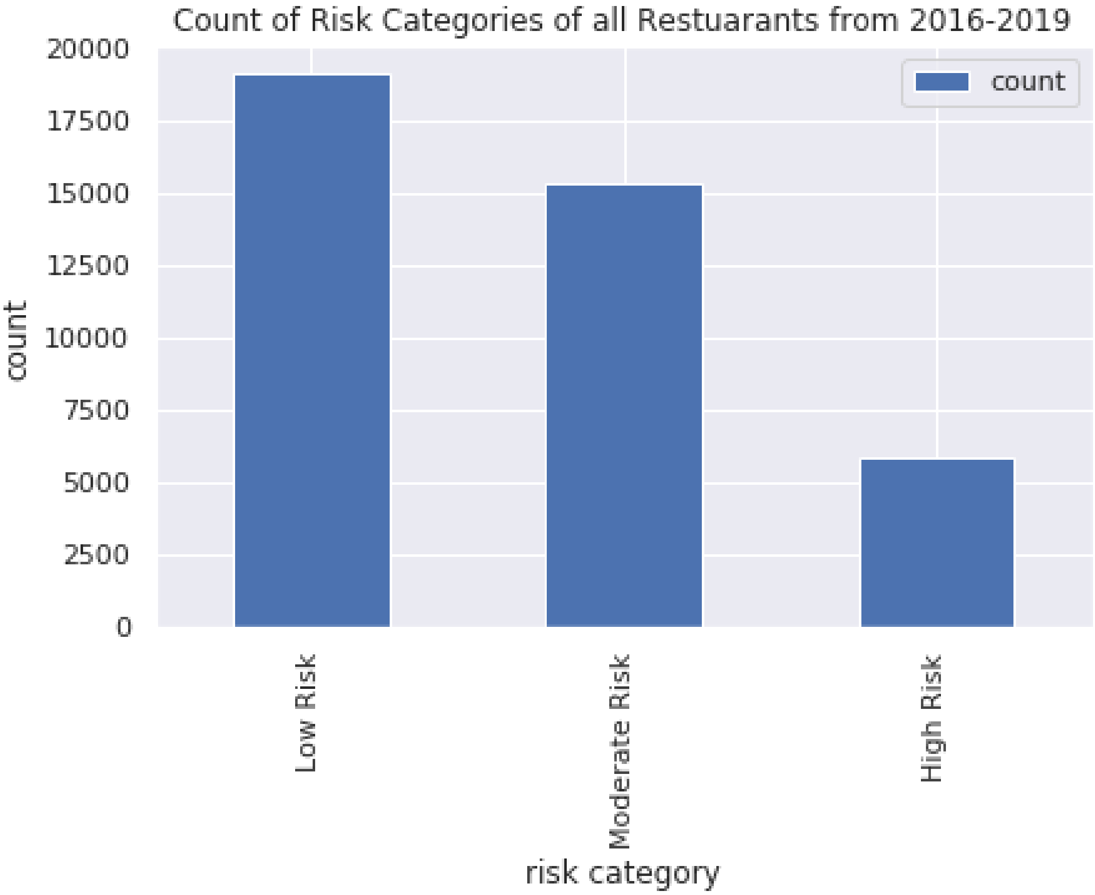
You should have the following in your answers: * a few visualizations; Please limit your visualizations to 5 plots. * a few sentences (not too long please!)

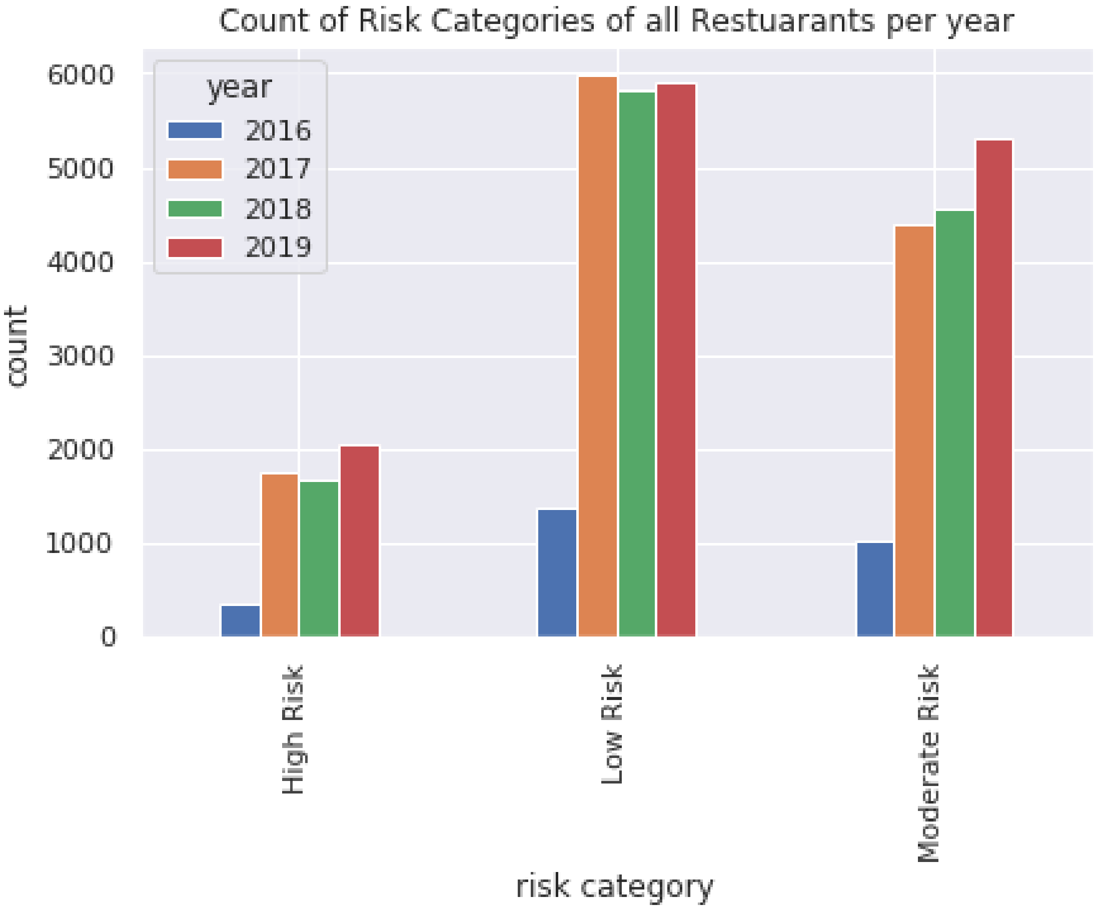
Please note that you will only receive support in OH and Piazza for Matplotlib and seaborn questions. However, you may use some other Python libraries to help you create you visualizations. If you do so, make sure it is compatible with the PDF export (e.g., Plotly does not create PDFs properly, which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

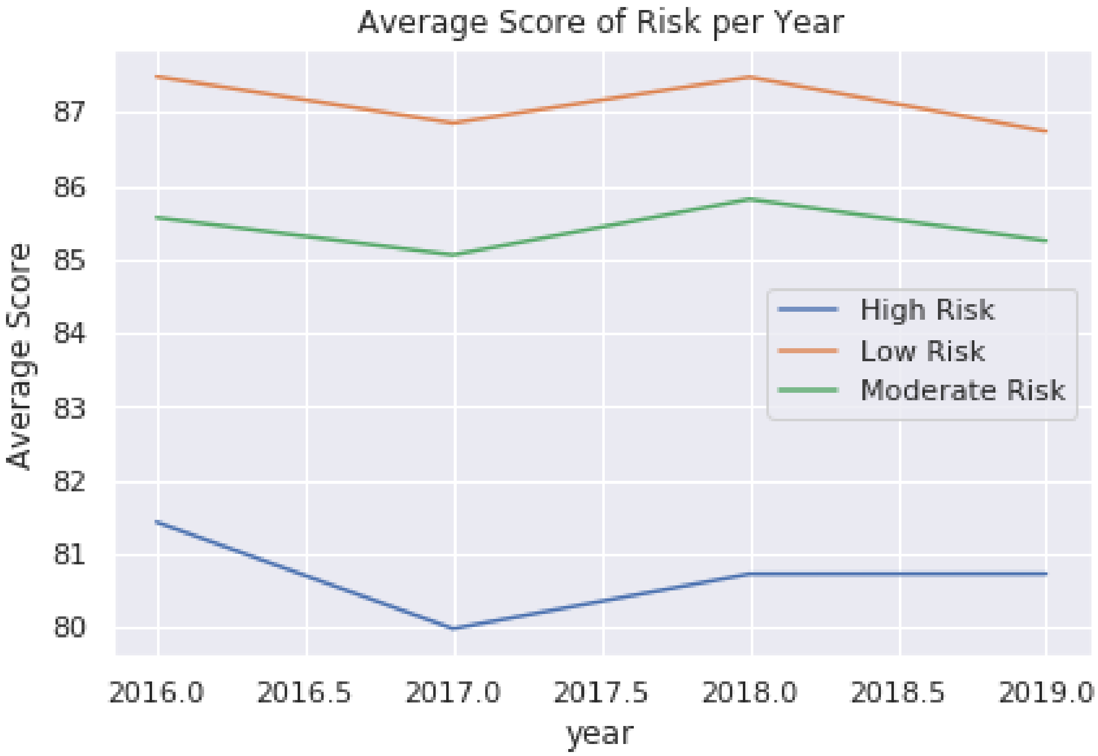
```
In [22]: # YOUR DATA PROCESSING AND PLOTTING HERE
         ins_vio_vid = ins.merge(ins2vio, how = 'inner')
         ins_vio = ins_vio_vid.merge(vio, how = 'inner')
         ins_vio_risk = ins_vio[['risk_category']]
         risk_count = ins_vio_risk['risk_category'].value_counts().to_frame().reset_index().rename(colu
                                                                              {'index': 'risk category', 'risk_ca
         risk_count.plot.bar(x = 'risk category', y = 'count')
         plt.ylabel('count')
         plt.title('Count of Risk Categories of all Restuarants from 2016-2019')
         # YOUR EXPLANATION HERE (in a comment)
         #This is a visualization of the number of restuarants categorized in a certain risk category
         #through the years 2016-2019. As a result we can depict that throughout 2016-2019 more 'Low Ri
         #labels were given than 'Moderate Risk' and 'High Risk'. In addition more 'Moderate Risk' labe
         #'High Risk'. Thus, overall there were a smaller quantity of 'High Risk' labels given.

         # YOUR DATA PROCESSING AND PLOTTING HERE
         ins_vio_risk = ins_vio[['risk_category', 'year', 'bid']].rename(columns = {'risk_category' : '
         risk_yr_count = ins_vio_risk.pivot_table(index = 'risk category', columns = 'year', aggfunc =
         risk_yr_count.plot.bar()
         plt.ylabel('count')
         plt.title('Count of Risk Categories of all Restuarants per year')
         # YOUR EXPLANATION HERE (in a comment)
         #This is a visualization of the number of restuarants categorized in a certain risk category p
         #This depicts the different counts (of restuarants) per risk category per year. We can see tha
         #as the years progessed that more labels of 'risk category' for restaurants were given which i
         #since as years pass more restuarants are constructed/open. However, there is two interesting
         #First being that in 2017 there was more 'High Risk' labels given to restaurants than in 2018.
         #there was more 'Moderate Risk' labels given to restaurants than in 2018 and 2019.
         # YOUR DATA PROCESSING AND PLOTTING HERE
         risk_yr_score = ins_vio[['risk_category', 'year', 'score']]
         filter_score = risk_yr_score[risk_yr_score['score'] != -1].groupby(by = ['risk_category',
                                                                                  'year']).mean().reset
         pv_risk_score_yr = filter_score.pivot_table(index = 'year', columns = 'risk_category')
         fig = pv_risk_score_yr.plot.line()
```

```
fig.legend(['High Risk', 'Low Risk', 'Moderate Risk'])
fig.set_ylabel('Average Score');
fig.set_title('Average Score of Risk per Year');
# YOUR EXPLANATION HERE (in a comment)
#This visualization depicts the trend of the average score of a particular risk per year. We c
#interesting observation that in 2016 the average score for all 'Risk Categories' was the comb
#years 2016-2019.
```




Count of Risk Categories of all Restuarants per year
year
2016
2017
2018
2019
count
0
1000
2000
3000
4000
5000
6000
High Risk
Low Risk
Moderate Risk
risk category


Average Score of Risk per Year
Average Score
87
86
85
84
83
82
81
80
2016.0
2016.5
2017.0
2017.5
2018.0
2018.5
2019.0
year
High Risk
Low Risk
Moderate Risk