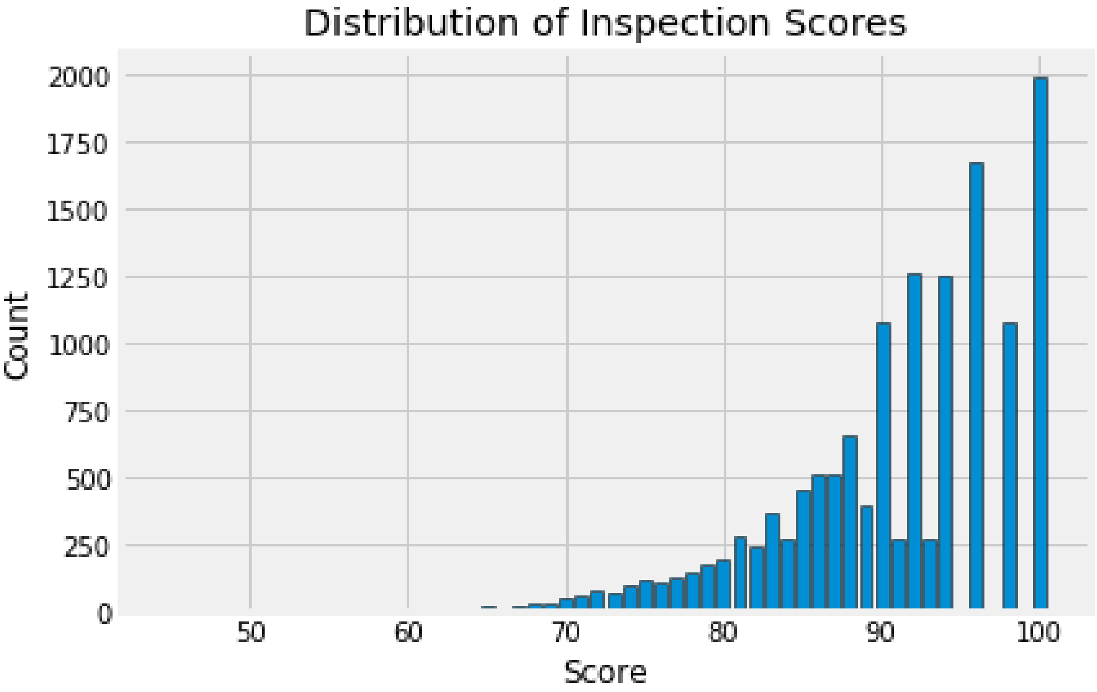
Notebook

February 19, 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.

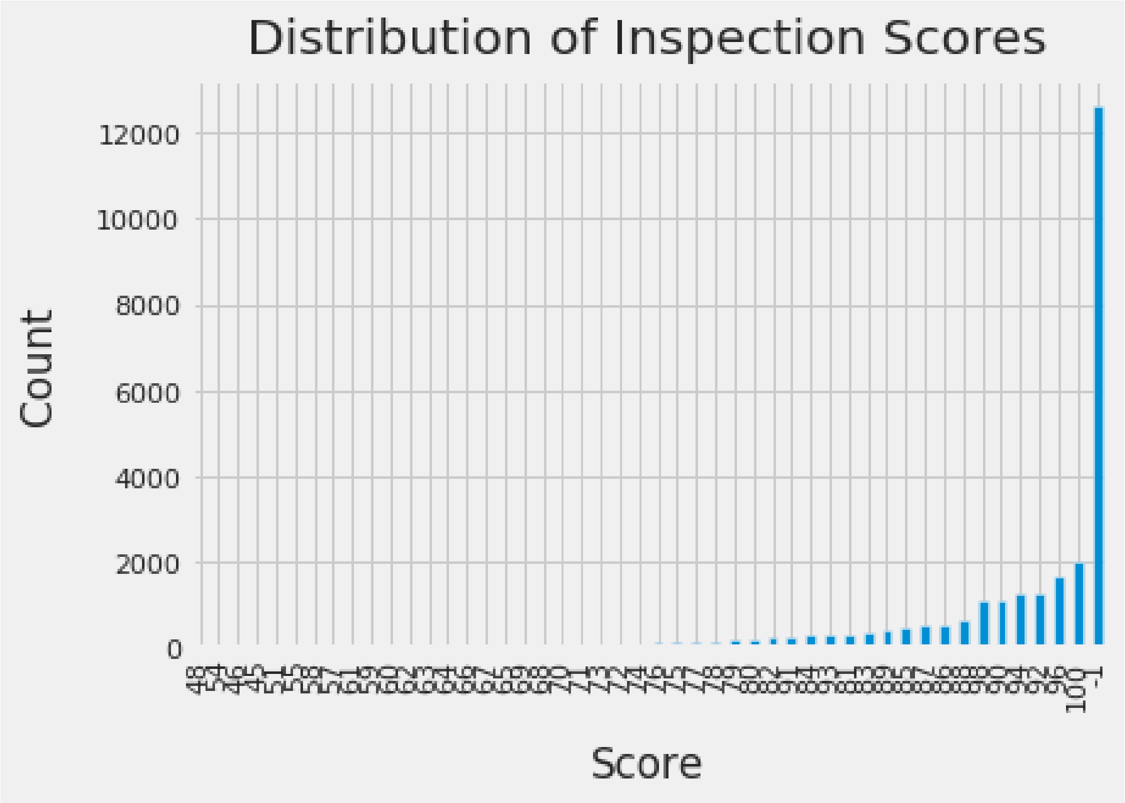


You might find this matplotlib.pyplot tutorial useful. Key syntax that you'll need:

```
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 [92]: ins_score = ins['score']
         ins_score.value_counts().sort_values().plot(kind='bar', rot = 90)
         plt.xlabel("Score", labelpad=14)
         plt.ylabel("Count", labelpad=14)
         plt.title("Distribution of Inspection Scores", y=1.02);
```


Distribution of Inspection Scores
Count
12000
10000
8000
6000
4000
2000
0
48
100
-1
Score

0.0.2 Question 1b

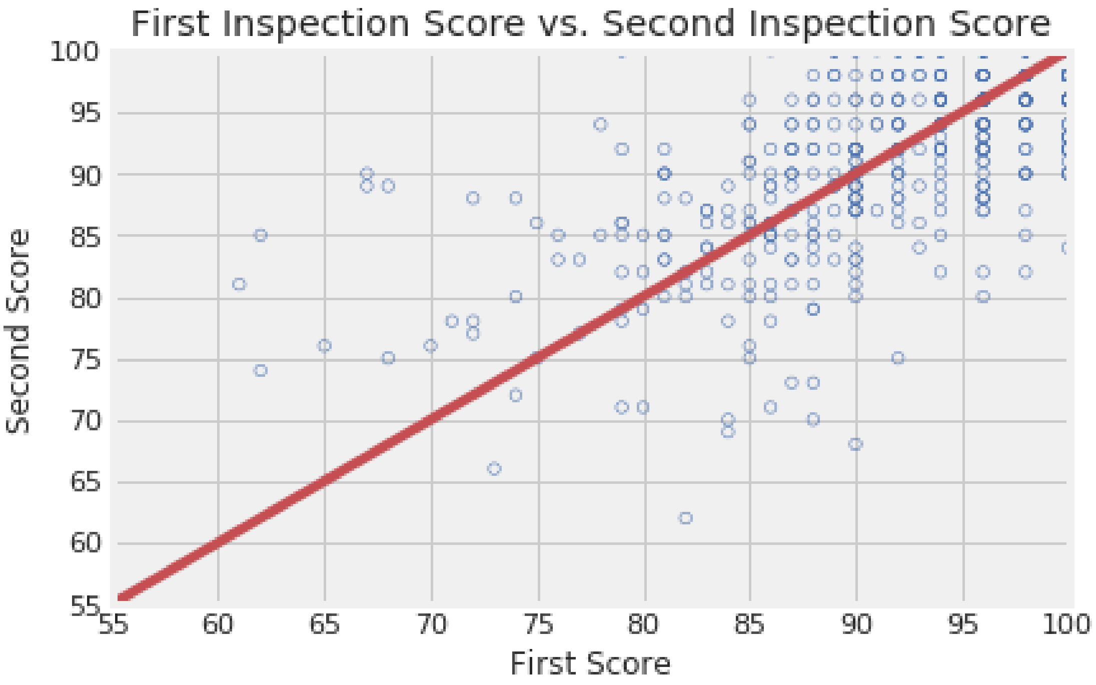
Describe the qualities of the distribution of the inspections scores based on your bar plot. Consider the mode(s), symmetry, tails, gaps, and anomalous values. Are there any unusual features of this distribution? What do your observations imply about the scores?

The scores seem to be skewed towards the higher inspection scores. The mode is 100 which means that most businesses have full inspection scores and there is a left tail which means that not alot of businesses have extremely low scores. This could mean that the businesses that have low inspection scores had to close down.

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.

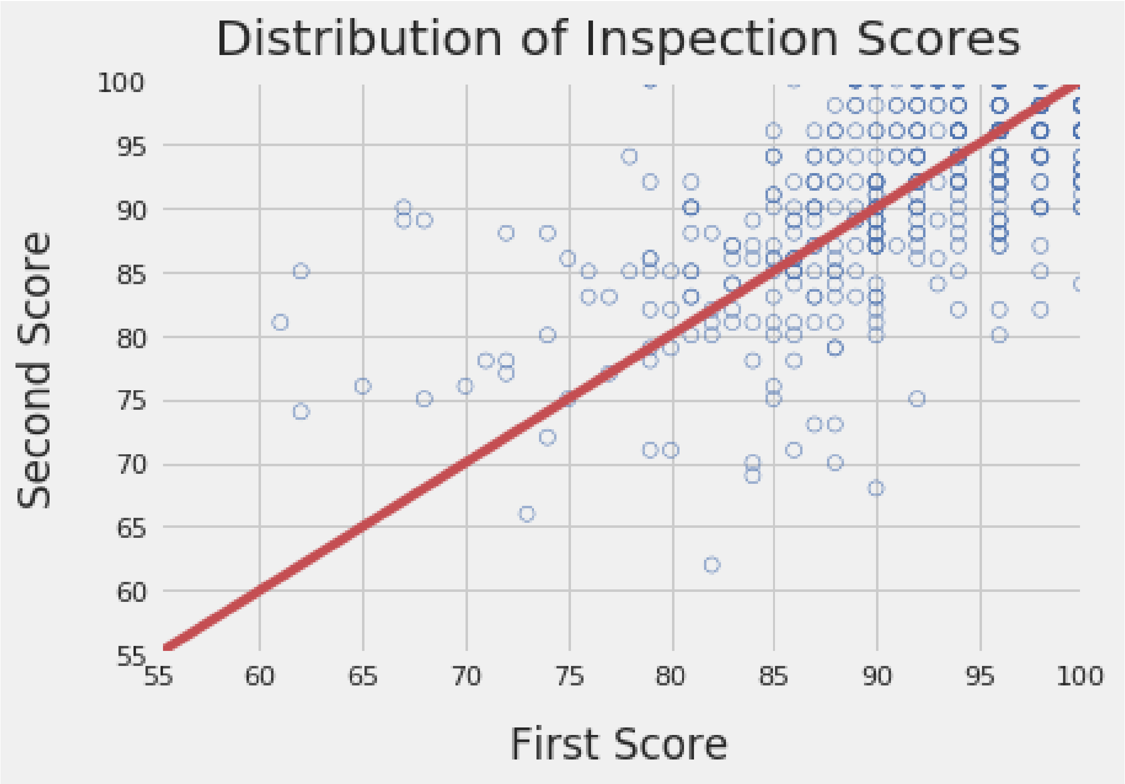
`plt.xlabel`, `plt.ylabel`, `plt.axis`, and `plt.title`.

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 [104]: plt.scatter(list(zip(*list(scores_pairs_by_business['score_pair'])))[0],list(zip(*list(scores_
          plt.plot(np.arange(55,106,5), np.arange(55, 106, 5), '-r', label ='y=x')
          plt.xlabel("First Score", labelpad=14)
          plt.ylabel("Second Score", labelpad=14)
          plt.title("Distribution of Inspection Scores", y=1.02);
          plt.axis([55, 100, 55, 100])
```

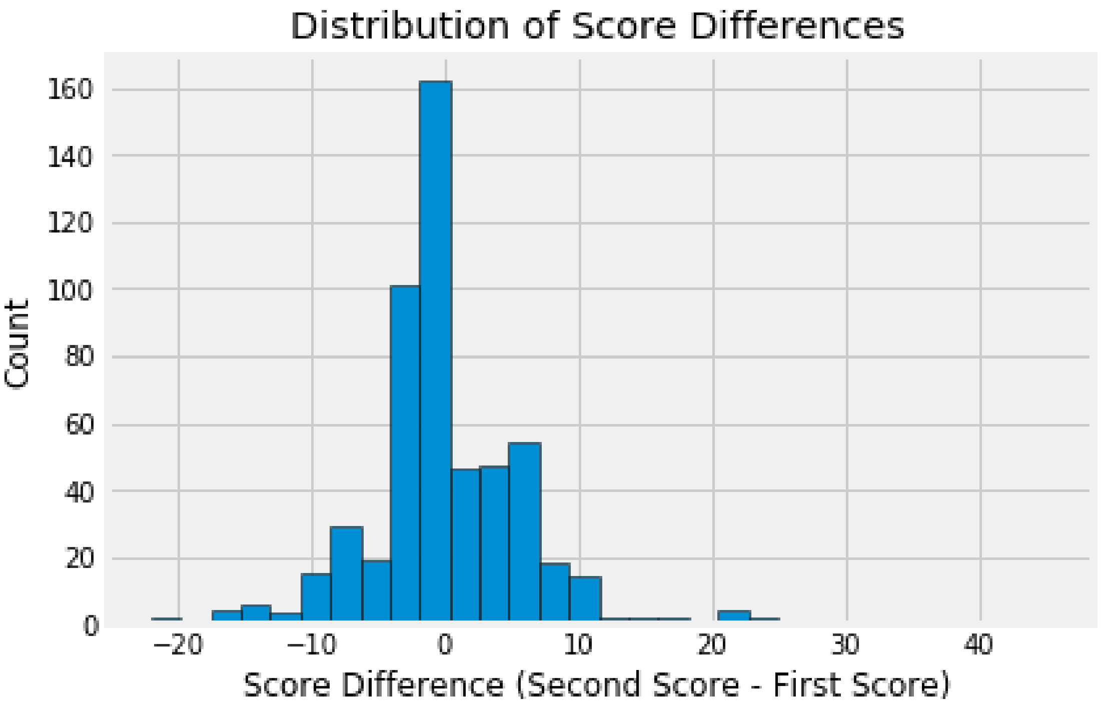
```
Out[104]: [55, 100, 55, 100]
```


Distribution of Inspection Scores
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

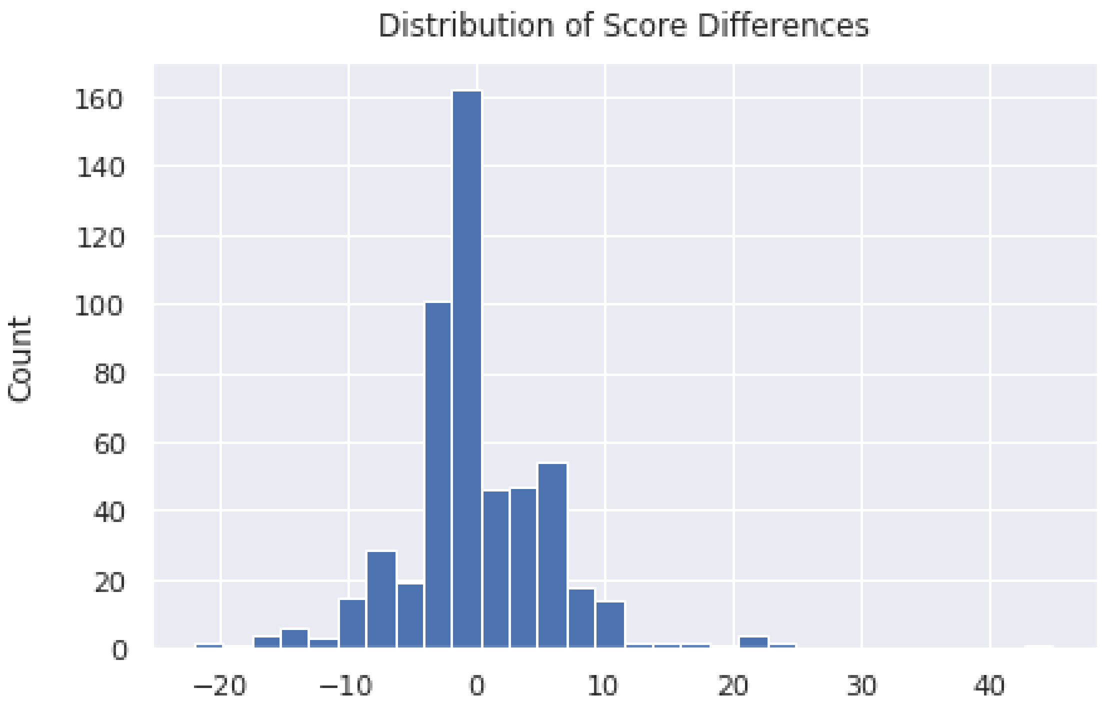
Another way to compare the scores from the two inspections is to examine the difference in scores. Subtract the first score from the second in `scores_pairs_by_business`. Make a histogram of these differences in the scores. We might expect these differences to be positive, indicating an improvement from 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 [188]: plt.hist(np.array(list(zip(*list(scores_pairs_by_business['score_pair']))))[1]-np.array(list(
          plt.ylabel("Count", labelpad=14)
          plt.title("Distribution of Score Differences", y=1.02);
```


Distribution of Score Differences
Count
0
20
40
60
80
100
120
140
160
−20
−10
0
10
20
30
40

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' scores tend to improve from the first to the second inspection, we expect to see in the scatter plot that there would be a strong positive correlation. We observe from the plot that there is a positive correlation and the observations are consistent with my expectations. We would expect that the positive correlation line would be even more positive which would mean that scores have a higher second score than the first score.

0.0.5 Question 2f

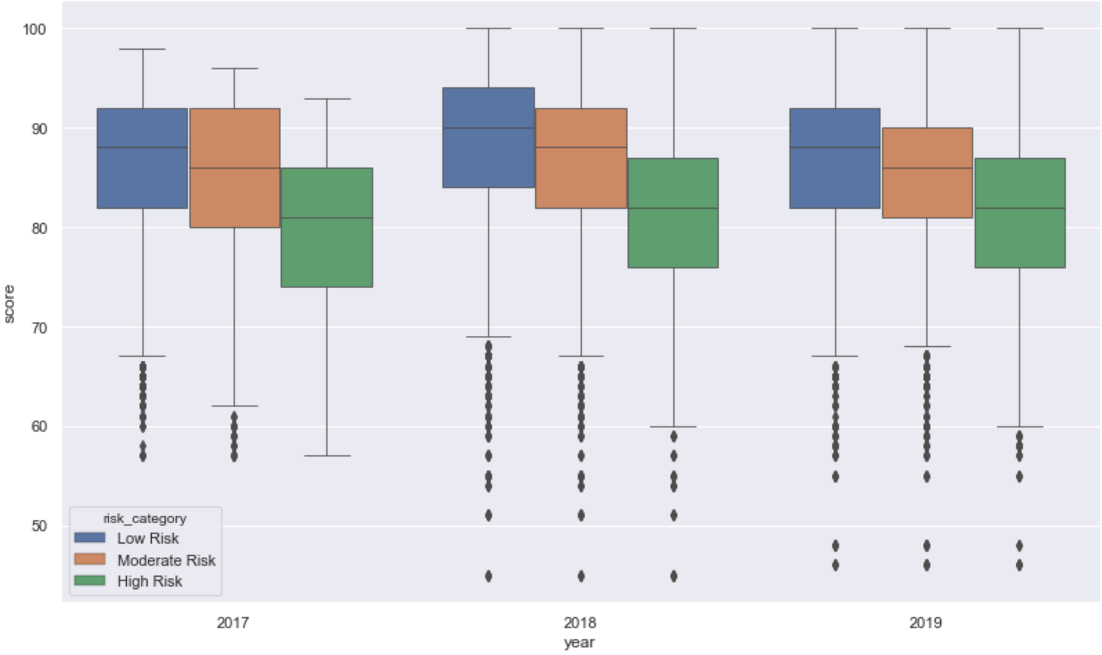
If a restaurant's score improves from the first to the second inspection, how would this be reflected in the histogram of the difference in the scores that you made in question 2d? What do you oberve from the plot? Are your observations consistent with your expectations? Explain your observations in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

If the restaurant's scores improve from the first to the second inspection, there would be a greater distribution of positive values as that would mean that the difference from the first score improved and increased to the higher score. We observe from the plot that there are many values in the bin immediately left of 0. This means that there were actually many values in which the difference between the first score and the second score is negative. The observations are not consistent with my expectations because I expected more scores to improve and have a spread in the positive bins. The center seems to be a little less than 0 and the deviations are near 0 with no major outliers.

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 [182]: ins_merge= pd.merge(ins, ins2vio, how = 'inner', on = 'iid')
          ins_merge1 = pd.merge(ins_merge, vio, how = 'inner', on = 'vid')
          ins_merge1 = ins_merge1[ins_merge1["year"] != 2016]
          ins_merge1
```

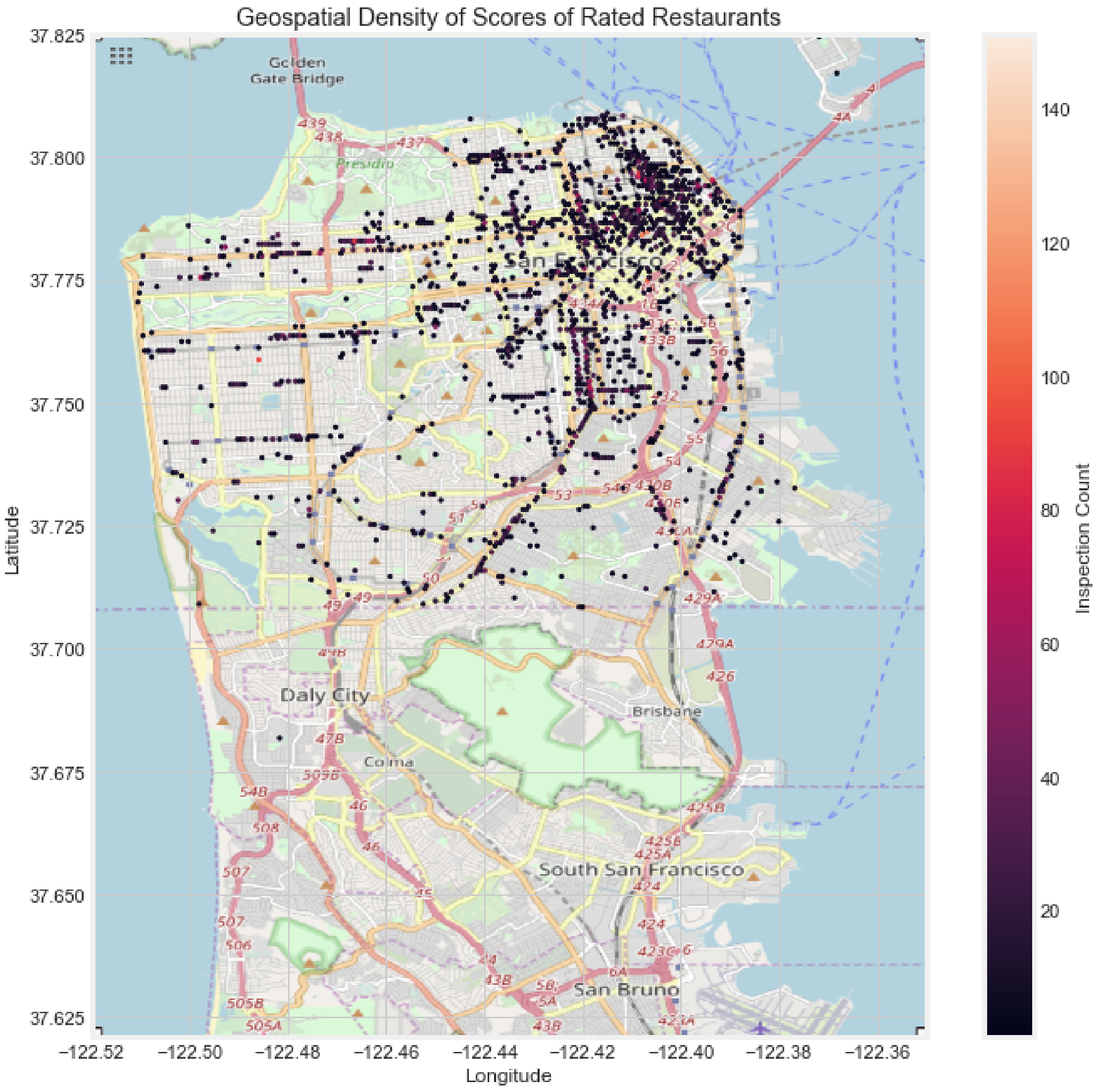
```
Out[182]:                     iid                     date  score                  type  \
          0      100017_20190816  08/16/2019 12:00:00 AM     91  Routine - Unscheduled
          1      100041_20190520  05/20/2019 12:00:00 AM     83  Routine - Unscheduled
          … Omitting 33 lines …
          40209  No restroom facility within 200 feet of mobile…     High Risk

          [37455 rows x 11 columns]
```

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.

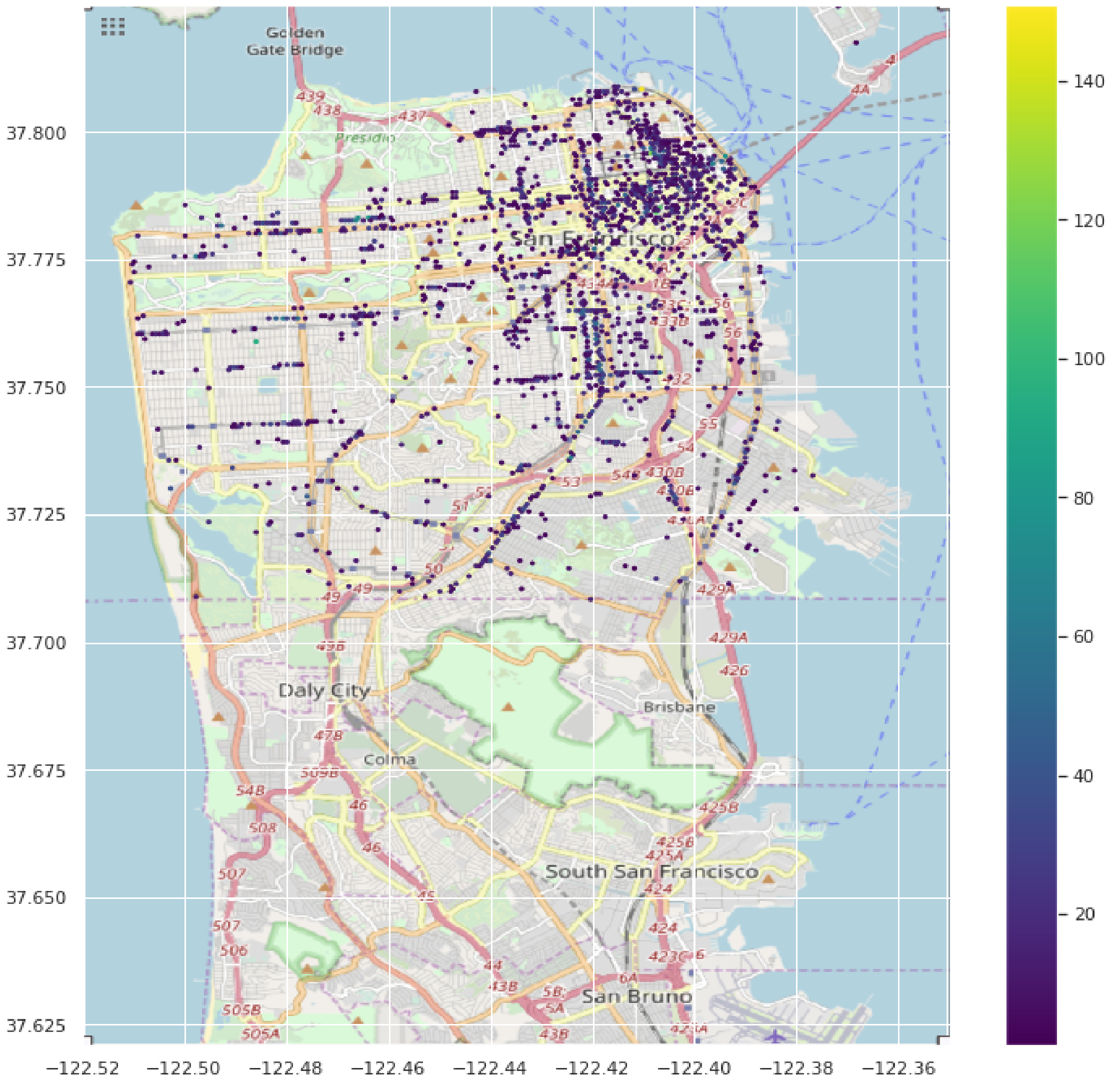
Hint: For the 2 functions we mentioned above, try looking at the parameter `reduce_C_function`, which determines the aggregate function for the hexbin plot.

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 [192]: # 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,
                    cmap="viridis")
plt.colorbar()

# 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
```

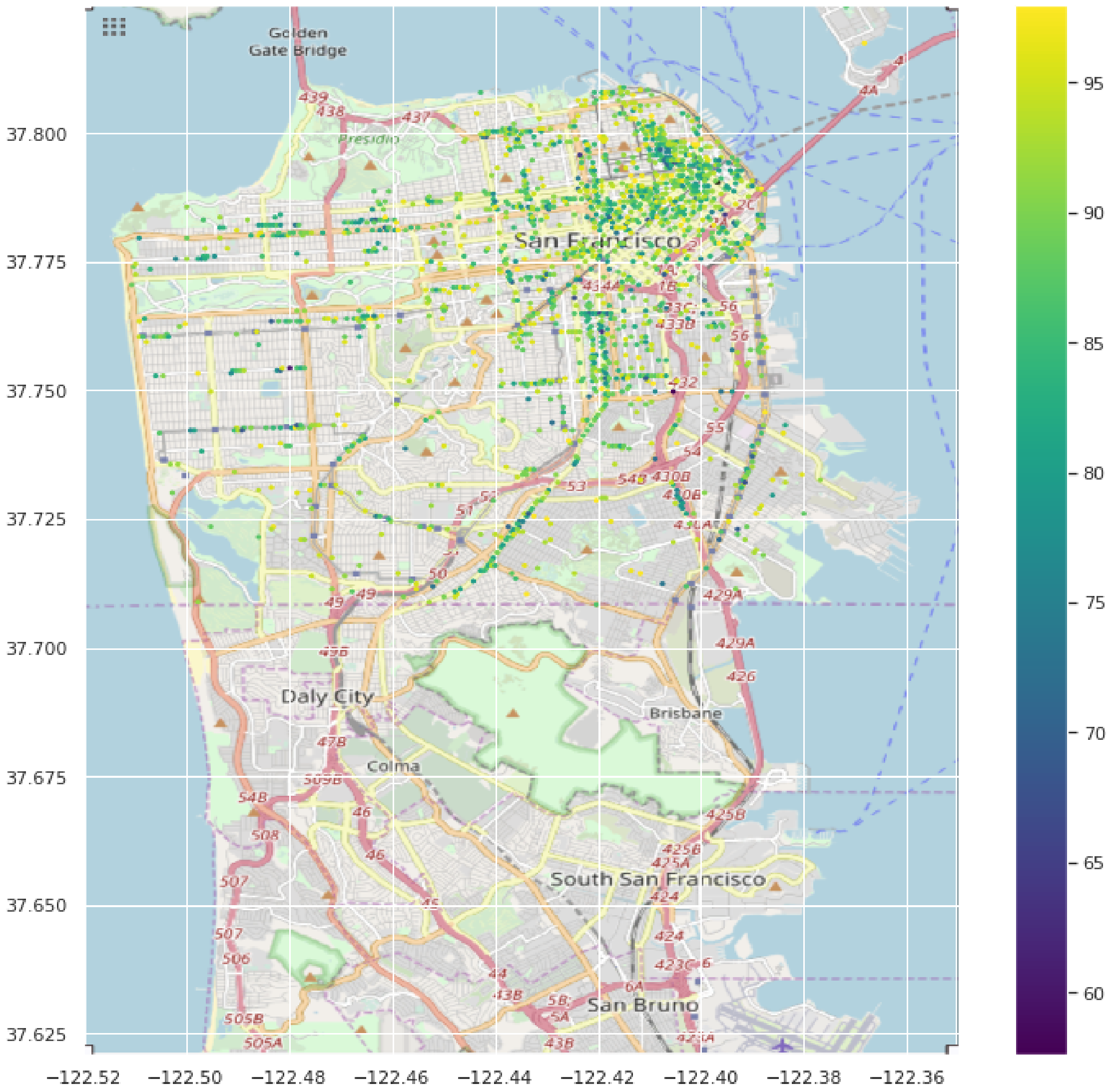

Golden Gate Bridge
Presidio
San Francisco
Daly City
Colma
Brisbane
South San Francisco
San Bruno
37.800
37.775
37.750
37.725
37.700
37.675
37.650
37.625
−122.52
−122.50
−122.48
−122.46
−122.44
−122.42
−122.40
−122.38
−122.36
140
120
100
80
60
40
20

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 [193]: # 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,
                             cmap="viridis")
          plt.colorbar()

          # 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
```


Golden Gate Bridge
Presidio
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
−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

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?

The districts in which the inspection tends to be more frequent is near the Financial District. The locations where the average inspection scores tends to be low is near Golden Gate park where there are less food places. In the Financial District, there are many food places because many people work in this concentrated area. Therefore, more food inspections take place there. The average inspection scores that tend to be low are away from the Financial District. This could mean that there are less inspections so the restaurants are not concerned about making their score higher. There is a connection of having more inspections and having higher average inspection scores as the Financial District area all has mainly high score dots.

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:

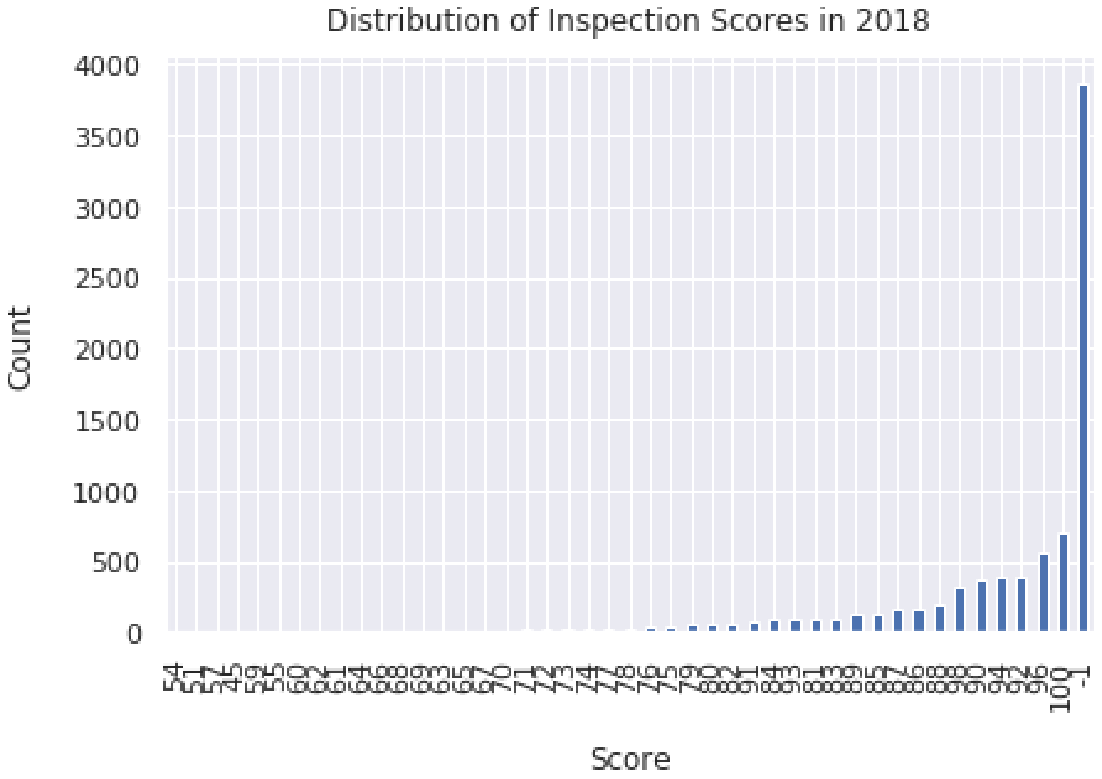
- **Great** (4-5 points): The chart is well designed, and the data computation is correct. The text written articulates a reasonable metric and correctly describes the relevant insight and answer to the question you are interested in.
- **Passing** (3-4 points): A chart is produced but with some flaws 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.

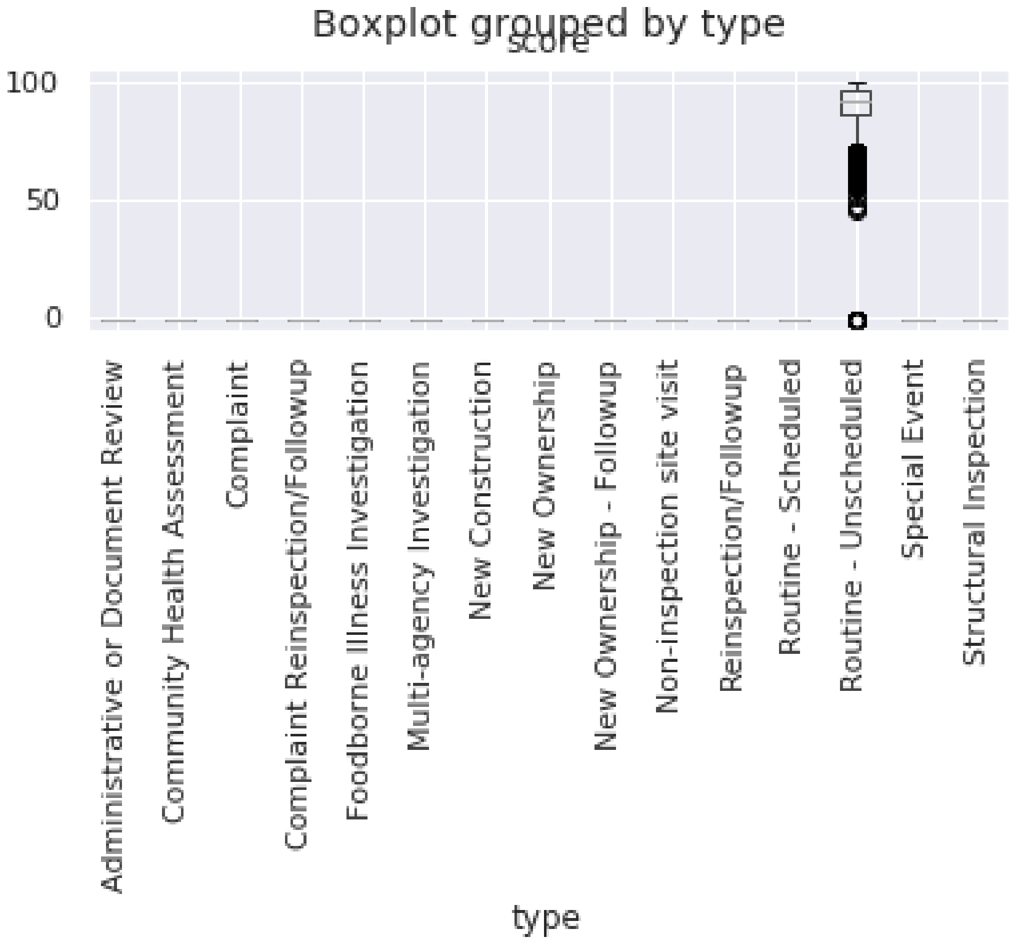
We will lean towards being generous with the grading. We might also either discuss in discussion or post on Piazza some examplar analysis you have done (with your permission)!

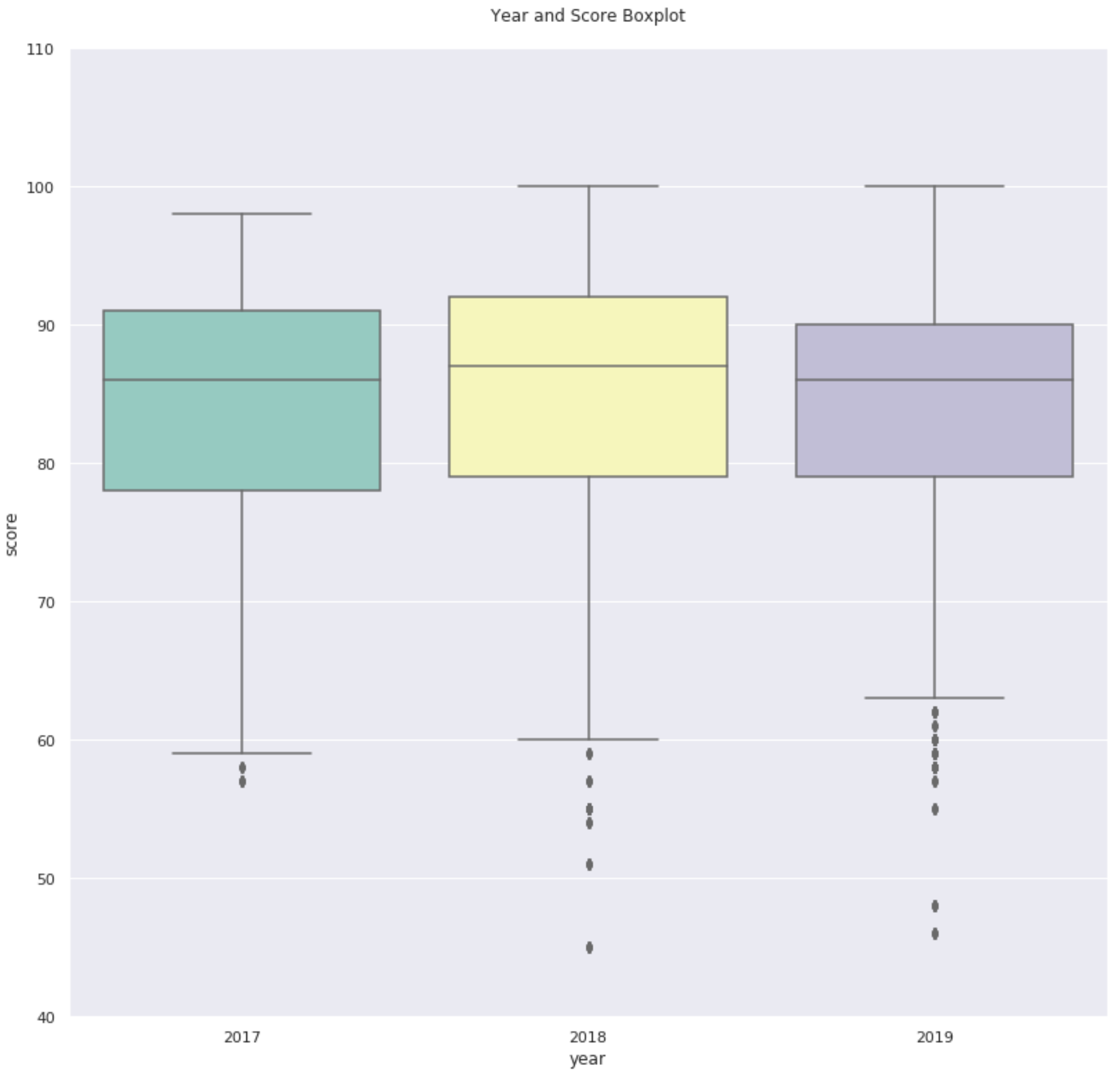
You should have the following in your answers: * a few visualizations; Please limit your visualizations to 5 plots. * a few sentences (not too long please!)

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 [195]: ins2018 = ins[ins['year'] == 2018]
          ins_score = ins2018['score']
          ins_score.value_counts().sort_values().plot(kind='bar', rot = 90)
          plt.xlabel("Score", labelpad=14)
          plt.ylabel("Count", labelpad=14)
          plt.title("Distribution of Inspection Scores in 2018", y=1.02);


          boxplot = ins.boxplot(column=['score'], by='type',
                                layout=(2, 1), rot = 90)



          plt.figure(figsize= (12, 12))
          ax = sns.boxplot(x="year", y="score",
                           data=ins_merge1, palette="Set3")
          plt.ylim(40, 110)
          plt.title("Year and Score Boxplot", y=1.02);

          #In the first plot, I showed the distribution of inspection scores in 2018. We see that overa
          #is similar to the distribution of all the inspection scores so 2018 does not seem to differ
          #years.

          #The second boxplot grouped by type shows that most of the inspections were routine-unschedul
          #are usually unscheduled inspections so as to check whether the restaurant is complying with
          #standards without having time to prepare beforehand.

          #The third boxplot shows the scores per year and this can be useful to see that the average s
          #overall, the scores are on average, around 86, but there are inspection scores that range fr
```


Distribution of Inspection Scores in 2018
Count
4000
3500
3000
2500
2000
1500
1000
500
0
54
51
57
45
59
55
60
62
61
64
66
68
69
63
65
67
70
71
72
73
74
77
78
76
75
79
80
82
91
84
93
81
83
89
85
87
86
88
98
90
94
92
96
100
-1
Score


Boxplot grouped by type
score
100
50
0
Administrative or Document Review
Community Health Assessment
Complaint
Complaint Reinspection/Followup
Foodborne Illness Investigation
Multi-agency Investigation
New Construction
New Ownership
New Ownership - Followup
Non-inspection site visit
Reinspection/Followup
Routine - Scheduled
Routine - Unscheduled
Special Event
Structural Inspection
type


Year and Score Boxplot
110
100
90
80
70
60
50
40
score
2017
2018
2019
year