Data 100/200 Homework 3 Written

Jackie Hu

TOTAL POINTS

7.75 / 14

QUESTION 1

1 Question 1a 1 / 1

✓ + 1 pts **Correct **

Describes a potential problem with the data.

+ 0 pts **Incorrect/Blank**

QUESTION 2

2 Question 5a 0.75 / 1

+ 1 pts Correct

✓ + 0.5 pts **Correct barplot but missing axis labels, titles, etc.**

✓ + 0.5 pts **Correct axis labels & title, with an attempt at the correct plot**

+ 0 pts Blank/Incorrect

- 0.25 **Point adjustment**

X-axis ticks aren't legible - try `plt.xticks` in the future

QUESTION 3

3 Question 5b 1 / 2

+ 2 pts Fully correct

✓ + 1 pts **Describes distribution to be unimodal at 100, left-skewed, and an unusual feature, such as bumpiness with more even numbers**

+ 1 pts Describe what observation implies about the score, such as the violations being even

+ 0 pts Incorrect

QUESTION 4

4 Question 6b 1 / 2

+ 2 pts Fully correct

+ 1 pts Correct plot but missing labels or title

✓ + 1 pts **Correct axis and labels with incorrect plot**

+ 0 pts Incorrect/Blank

QUESTION 5

5 Question 6c 1 / 1

✓ + 1 pts **Correct**

We expect the points to fall above the line of slope 1, but don't see this in our scatter plot. The second inspection is often worse than the first.

+ 0 pts **Incorrect/Blank/Not declaring preference**

QUESTION 6

6 Question 6d 1 / 3

+ 3 pts Correct axes, hue, data, and legend

+ 2 pts Minor error such as no legend, wrong order of risk categories, or includes 2016 (for example)

✓ + 1 pts **Major errors such as not grouped by hue or incorrect data or axes (for example)**

+ 0 pts Blank or completely incorrect

QUESTION 7

7 Question 7 2 / 4

+ 4 pts Dataframe shows combination of Pandas operations performed on it, text description provides reasonable interpretation of result.

OR

Visualization is well designed and reflects the conclusion made about the data.

+ 1 pts Properly cleaned data and encoding OR variety of Pandas functions used.

✓ + 1 pts **Reasonable or Insightful metric and some visualization or some dataframe**

+ 1 pts Properly Labeled with title, legend OR Clear Table (not too many extraneous columns) and clear column names

✓ + 1 pts **Clear Conclusion**

+ 0 pts Incorrect/empty

Use the `head` command on your three files again. This time, describe at least one potential problem with the data you see. Consider issues with missing values and bad data.

```
In [15]: bus.head()
```

Out[15]:

	business id column	name	address	city	state	postal_code	latitude	longitude	phone_number
0	1000	HEUNG YUEN RESTAURANT	3279 22nd St	San Francisco	CA	94110	37.755282	-122.420493	-9999
1	100010	ILLY CAFFE SF_PIER 39	PIER 39 K-106-B	San Francisco	CA	94133	-9999.000000	-9999.000000	14154827284
2	100017	AMICI'S EAST COAST PIZZERIA	475 06th St	San Francisco	CA	94103	-9999.000000	-9999.000000	14155279839
3	100026	LOCAL CATERING	1566 CARROLL AVE	San Francisco	CA	94124	-9999.000000	-9999.000000	14155860315
4	100030	OUI OUI! MACARON	2200 JERROLD AVE STE C	San Francisco	CA	94124	-9999.000000	-9999.000000	14159702675

```
In [16]: ins.head()
```

Out[16]:

	iid	date	score	type
0	100010_20190329	03/29/2019 12:00:00 AM	-1	New Construction
1	100010_20190403	04/03/2019 12:00:00 AM	100	Routine - Unscheduled
2	100017_20190417	04/17/2019 12:00:00 AM	-1	New Ownership
3	100017_20190816	08/16/2019 12:00:00 AM	91	Routine - Unscheduled
4	100017_20190826	08/26/2019 12:00:00 AM	-1	Reinspection/Followup

```
In [17]: vio.head()
```

Out[17]:

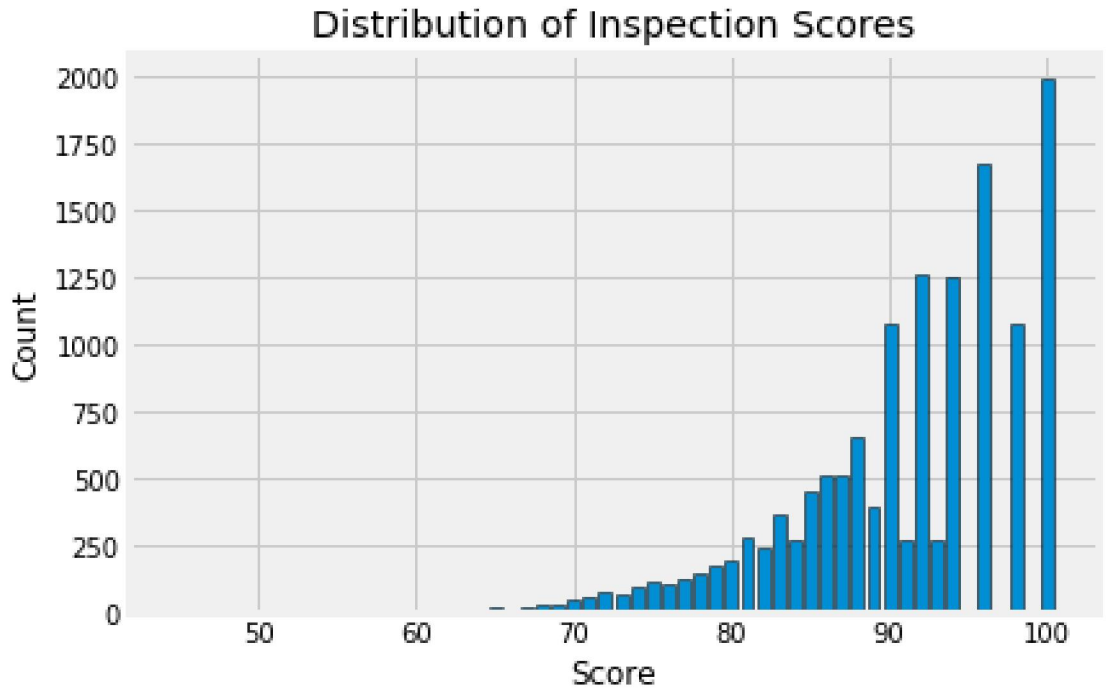
	description	risk_category	vid
0	Consumer advisory not provided for raw or unde…	Moderate Risk	103128
1	Contaminated or adulterated food	High Risk	103108
2	Discharge from employee nose mouth or eye	Moderate Risk	103117
3	Employee eating or smoking	Moderate Risk	103118
4	Food in poor condition	Moderate Risk	103123

- For each table the business id is inconsistent in format, assuming we will need this identification to join data together.
- bus and ins containing bad data like negative numbers and bad formatting.
- ins table's iid seems to containing 2 different identification.

0.1 Question 5a

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 bar plot 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:

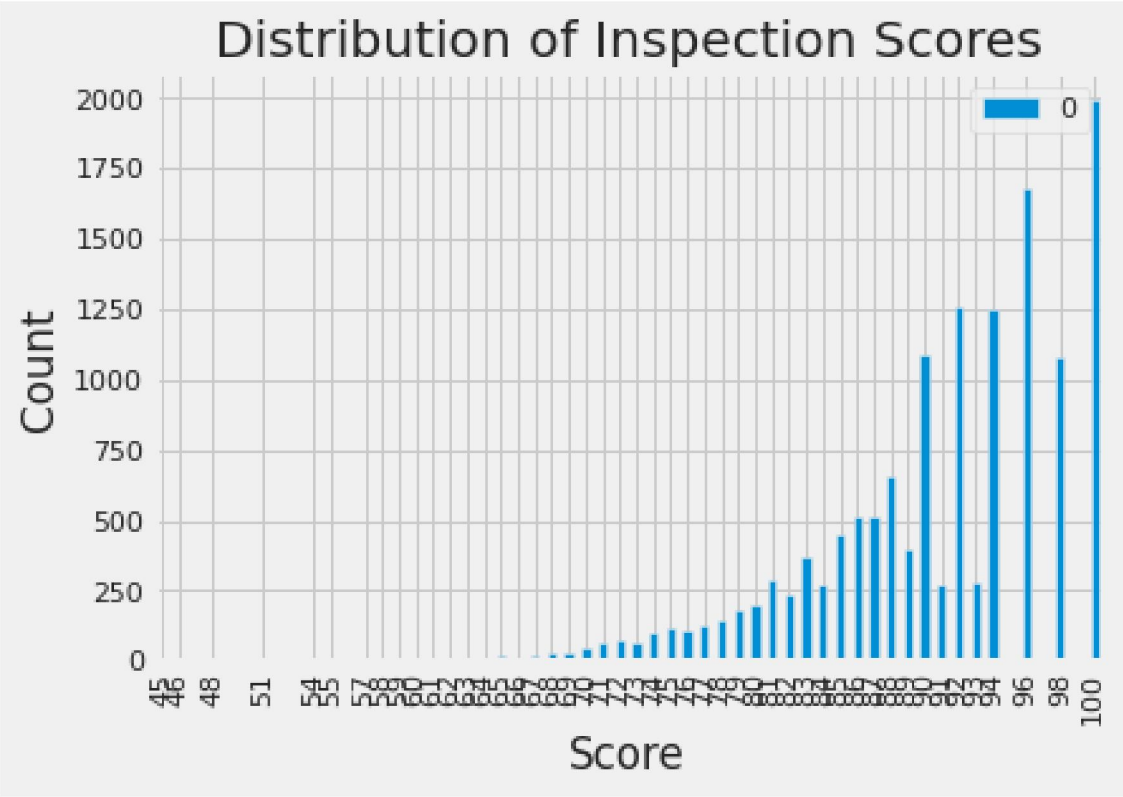
```
plt.bar
plt.xlabel
plt.ylabel
plt.title
```

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 [124]:
```
ins_grouped = ins.groupby(ins.score).size()
ins_grouped_df = ins_grouped.to_frame()
```

In [125]:
```
ax = ins_grouped_df.plot.bar()
plt.xlabel('Score')
plt.ylabel('Count')
plt.title('Distribution of Inspection Scores')
```

Out[125]:
```
Text(0.5, 1.0, 'Distribution of Inspection Scores')
```

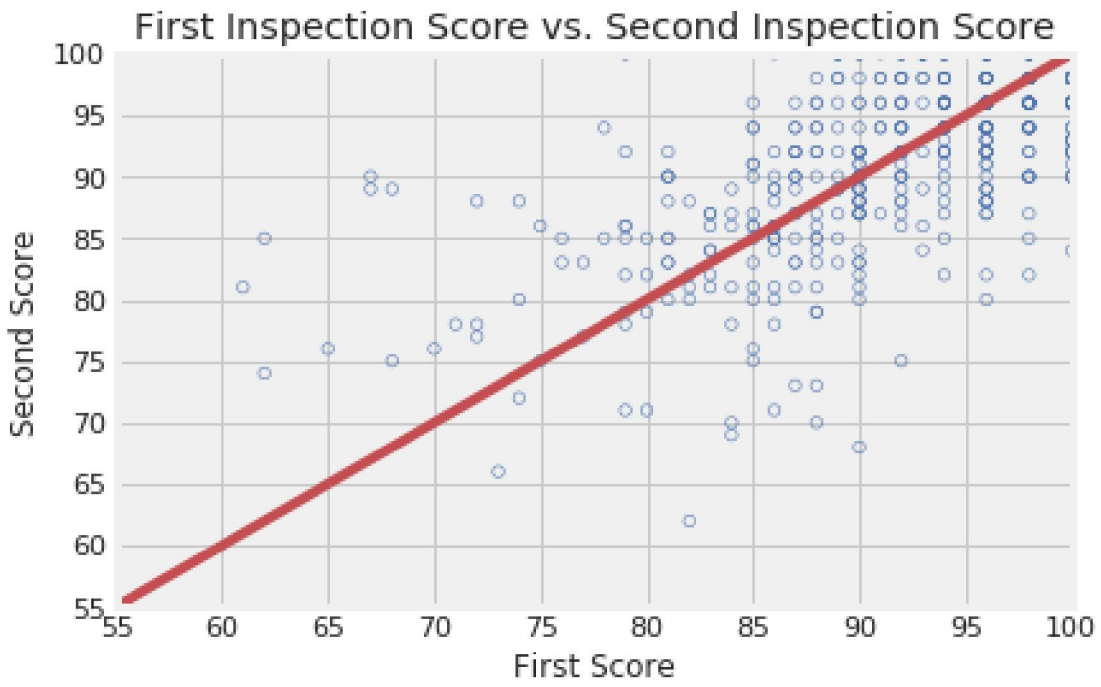


0.1.1 Question 5b

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 pretty heavily skewed towards 100. There seems to be unevenly distribution around 90, and the score is not distributed evenly in each tiers, for example, the score 90 and 92 has lots of counts, but 91 has a considerably less count.

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`.

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

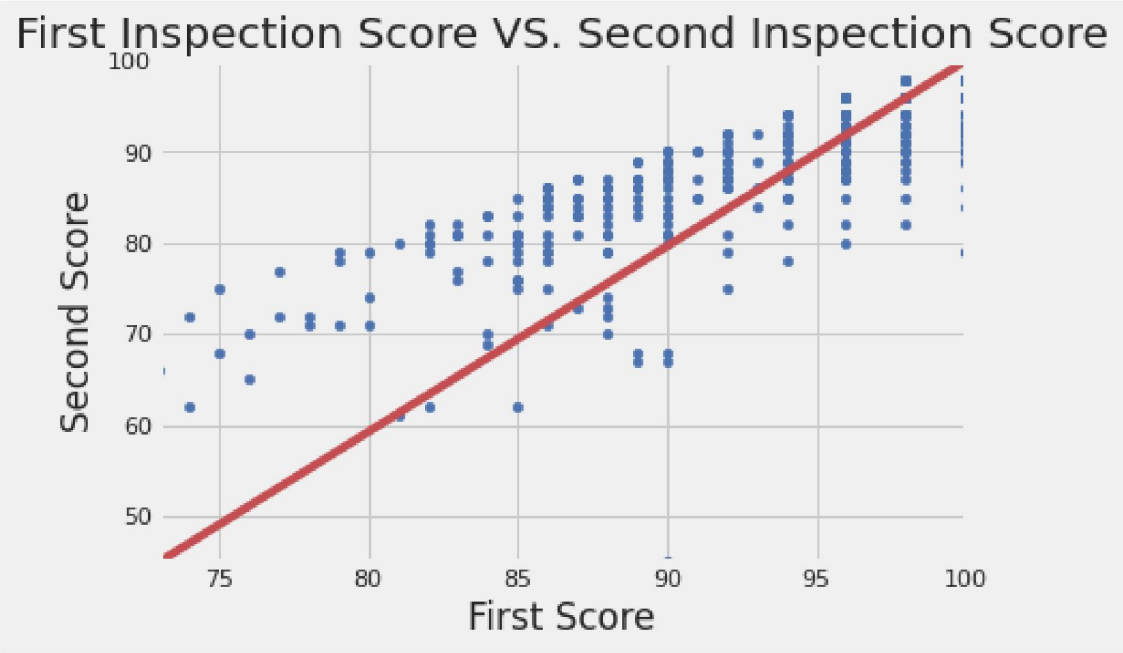
```
In [149]: ax1 = pair.plot.scatter(x='score_x_x',
                                  y='score_x_y',
                                  facecolors='none',
                                  edgecolors='b')

          minx = min(pair['score_x_x'])
          miny = min(pair['score_x_y'])
          maxx = max(pair['score_x_x'])
          maxy = max(pair['score_x_y'])
```

```
plt.xlabel('First Score')
plt.ylabel('Second Score')
plt.title('First Inspection Score VS. Second Inspection Score')
plt.xlim([minx, maxx])
plt.ylim([miny, maxy])
plt.plot((minx,maxx), (miny, maxy), 'r')
```

```
*c* argument looks like a single numeric RGB or RGBA sequence, which should be avoided as value-mapping
```

Out[149]: [<matplotlib.lines.Line2D at 0x7fba87285ee0>]



0.1.2 Question 6c

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 7c? What do you oberve from the plot? Are your observations consistent with your expectations?

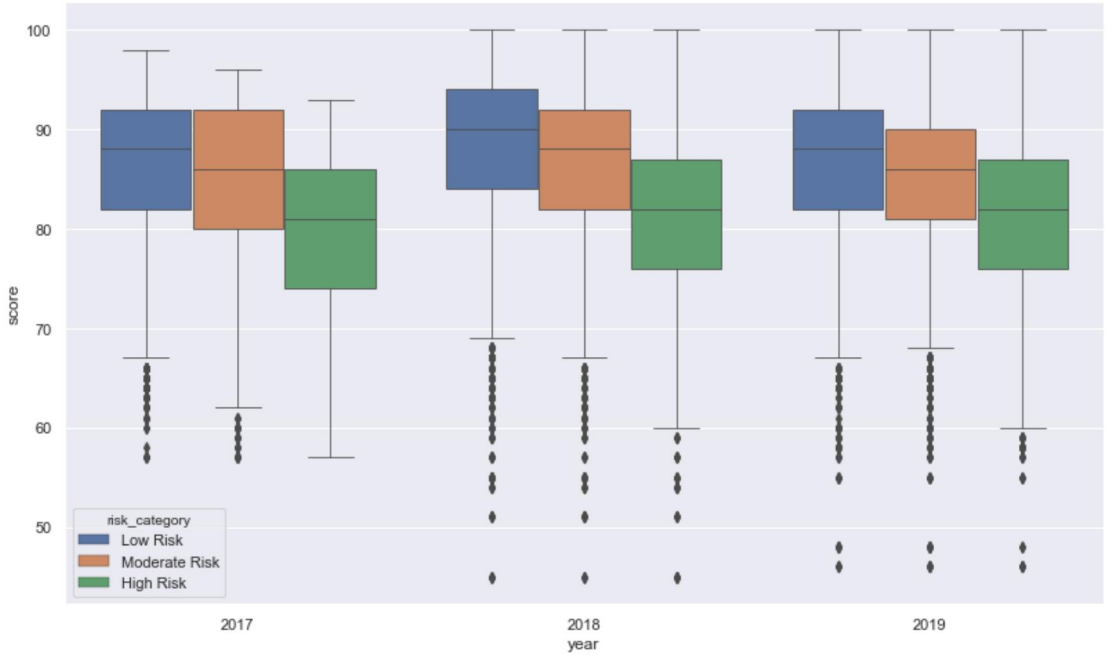
Hint: What does the slope represent?

- If the scores tend to impove than most the points should be distributed in the upper quartile, since the line marked the boundary of scores being the same, therefore if the second score is higher than the first one, the points should be locate ablove the line.
- From the graph, we can see that the distribution is almost equally distributed around the line, the business that has lower first score tends to have higher higher score the second time; the business that has higher score the first time tends to cluster around the line or slightly below the line, especially the ones' score close to 100.

0.1.3 Question 6d

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. Make sure the boxes are in the correct order!



Hint: Use `sns.boxplot()`. Try taking a look at the first several parameters. The documentation is linked here!

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

```
In [150]: ins.head()
```

```
Out[150]:                iid                    date  score                    type  \
          1   100010_20190403  04/03/2019 12:00:00 AM    100  Routine - Unscheduled
          3   100017_20190816  08/16/2019 12:00:00 AM     91  Routine - Unscheduled
          15  100041_20190520  05/20/2019 12:00:00 AM     83  Routine - Unscheduled
          20  100055_20190425  04/25/2019 12:00:00 AM     98  Routine - Unscheduled
          21  100055_20190912  09/12/2019 12:00:00 AM     82  Routine - Unscheduled
```

```
       bid  timestamp  year  Missing_score
1   100010 2019-04-03  2019          False
3   100017 2019-08-16  2019          False
15  100041 2019-05-20  2019          False
20  100055 2019-04-25  2019          False
21  100055 2019-09-12  2019          False
```

In [151]:
```
vio.head()
```

Out[151]:

	description	risk_category	vid
0	Consumer advisory not provided for raw or unde…	Moderate Risk	103128
1	Contaminated or adulterated food	High Risk	103108
2	Discharge from employee nose mouth or eye	Moderate Risk	103117
3	Employee eating or smoking	Moderate Risk	103118
4	Food in poor condition	Moderate Risk	103123

In [152]:
```
plt_df = ins[['bid','year','score']]
plt_df = plt_df.merge(ins2vio_vio, how= 'left', on = 'bid').drop(['bid','iid', 'vid','descrip
plt_df = plt_df[(plt_df['year'] == 2017) | (plt_df['year'] == 2018) | (plt_df['year'] == 2019
plt_df
```

Out[152]:

	year	score	risk_category
0	2019	100	NaN
1	2019	91	High Risk
2	2019	91	Low Risk
3	2019	83	High Risk
4	2019	83	Low Risk
…	…	…	…
114516	2018	84	Moderate Risk
114517	2018	84	Moderate Risk
114518	2018	84	Low Risk
114519	2018	84	Low Risk
114520	2018	84	Low Risk

[105764 rows x 3 columns]

In [153]:
```
plt_df = plt_df.groupby(['year', 'risk_category'])['score'].apply(list).to_frame()
sc = plt_df.explode('score').reset_index()
sc
```

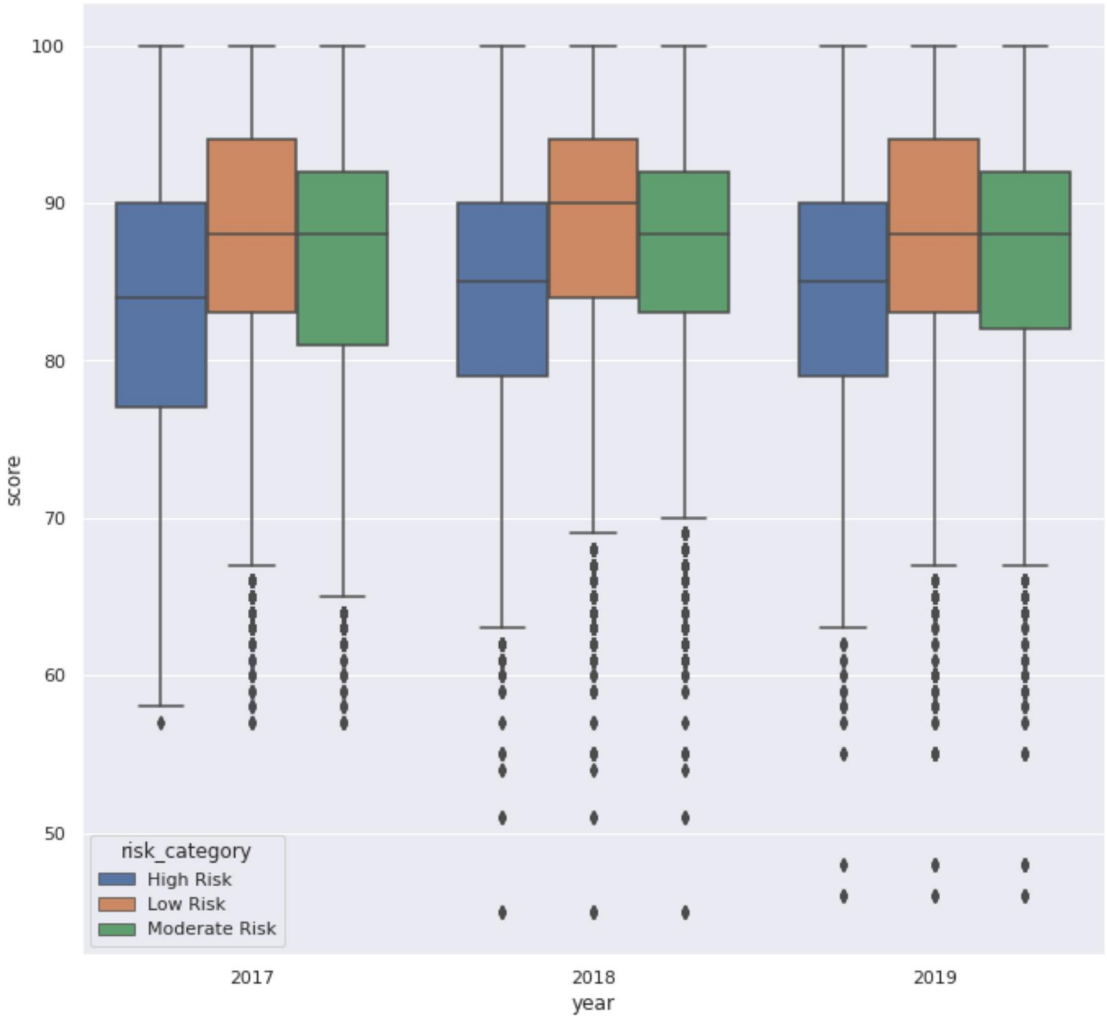
Out[153]:

	year	risk_category	score
0	2017	High Risk	74
1	2017	High Risk	74
2	2017	High Risk	74

```
3        2017       High Risk    74
4        2017       High Risk    74
…        …             …   …
105137   2019   Moderate Risk    80
105138   2019   Moderate Risk    80
105139   2019   Moderate Risk    80
105140   2019   Moderate Risk    80
105141   2019   Moderate Risk    80

[105142 rows x 3 columns]
```

In [154]:
```
# Do not modify this line
sns.set()
import seaborn as sns
plt.figure(figsize = (10,10))
ax = sns.boxplot(x='year', y="score", data=sc, hue="risk_category")
```



0.1.4 Grading

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

- **Great** (4 points):
  - For a dataframe, a combination of pandas operations (such as groupby, pivot, merge) is used to answer a relevant question about the data. The text description provides a reasonable interpretation of the result.
  - For a visualization, the chart is well designed and the data computation is correct. The conclusion based on the visualization articulates a reasonable metric and correctly describes the relevant insight and answer to the question you are interested in.
- **Passing** (1-3 points):
  - For a dataframe, computation is flawed or very simple. The conclusion doesn't fully address the question, but reasonable progress has been made toward answering it.
  - For a visualization, a chart is produced but with some flaws such as bad encoding. The conclusion based on the visualization is incomplete but makes some sense.
- **Unsatisfactory** (0 points):
  - For a dataframe, no computation is performed, or the conclusion does not match what is computed at all.
  - For a visualization, 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 exemplary analysis you have done (with your permission)!

You should have the following in your answers: * a question you want to explore about the data. * either of the following: * a few computed dataframes. * a few visualizations. * a few sentences summarizing what you found based on your analysis and how that answered your question (not too long please!)

Please limit the number of your computed dataframes and visualizations **you plan on showing** to no more than 5.

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 [160]:
```
# YOUR QUESTION HERE (in a comment)
# wahat are some of the zip codes that has most of the restaurant, therefore considered a goo
# YOUR DATA PROCESSING AND PLOTTING HERE
#bus.set_index('bid', inplace=True)
bus.head()
zipcounts = bus['postal_code'].value_counts()
zipcounts.name = 'counts'
```

```
plt.figure(figsize = (10,10))
zipcounts.plot.bar()

# YOUR SUMMARY AND CONCLUSION HERE (in a comment)
# The top 3 zip codes are 94103, 94110, 94102; from the map we can see those zip code are rea
```

Out[160]: <AxesSubplot:>