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 the bus file, some resturants are missing or incomplete data such as business id, latitude, longitude, and phone number. In the ins file, we are missing the scores of certain resturants. For example, some of the business id in the corresponding bus file do not match the business id in the ins file and it makes it difficult to identify certain data across the files.

In [166]:
```
bus.head()
ins2vio.head()
ins.head()
vio.head()
```

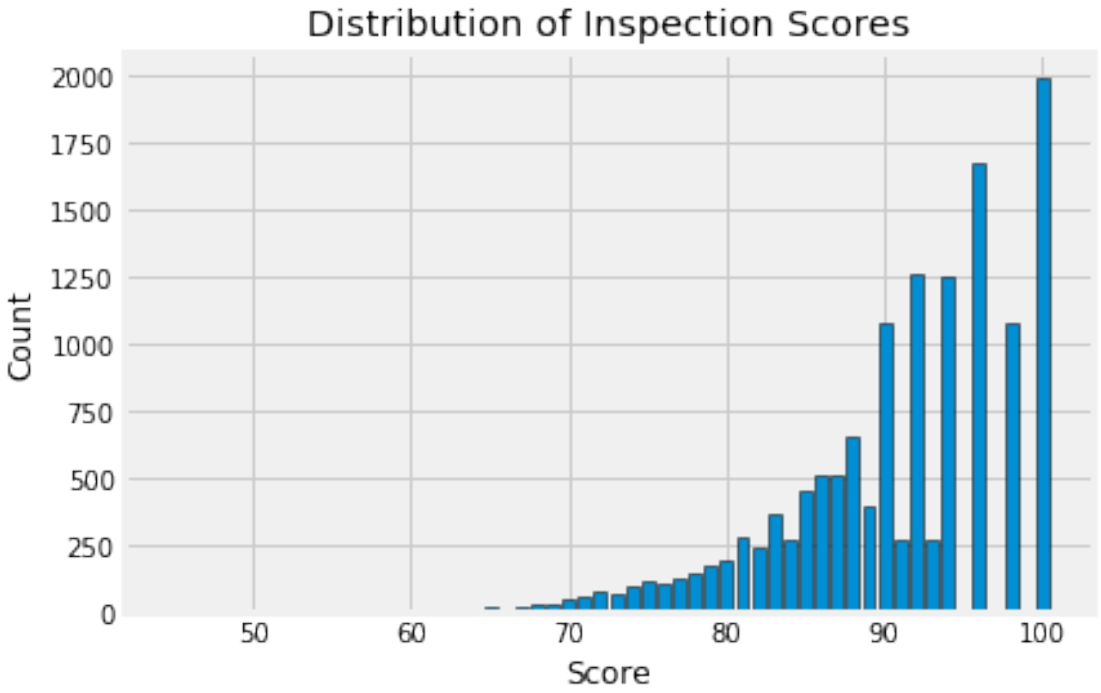
Out[166]:

	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

0.1 Question 6a

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.



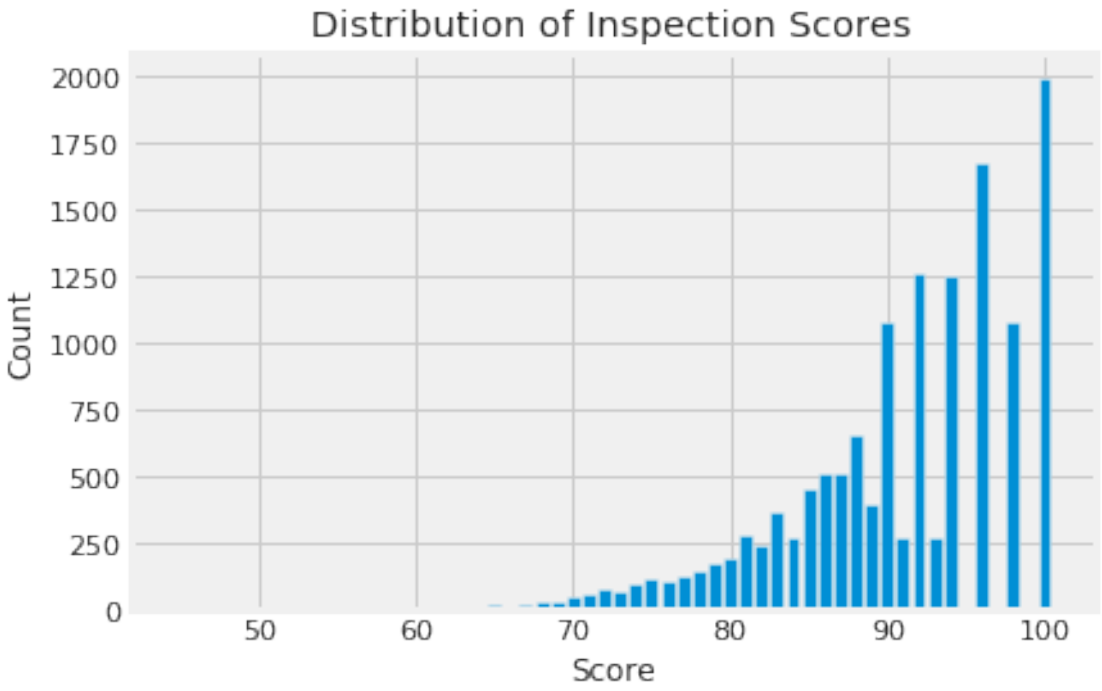
You might find this matplotlib.pyplot tutorial useful. Key syntax that you'll need:

```
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 [74]:
```
temp = ins[ins['score'] >= 0]['score'].value_counts()
plt.title('Distribution of Inspection Scores')
plt.xlabel('Score')
plt.ylabel('Count')
plt.bar(temp.keys(), temp)
```

Out[74]: <BarContainer object of 47 artists>



0.1.1 Question 6b

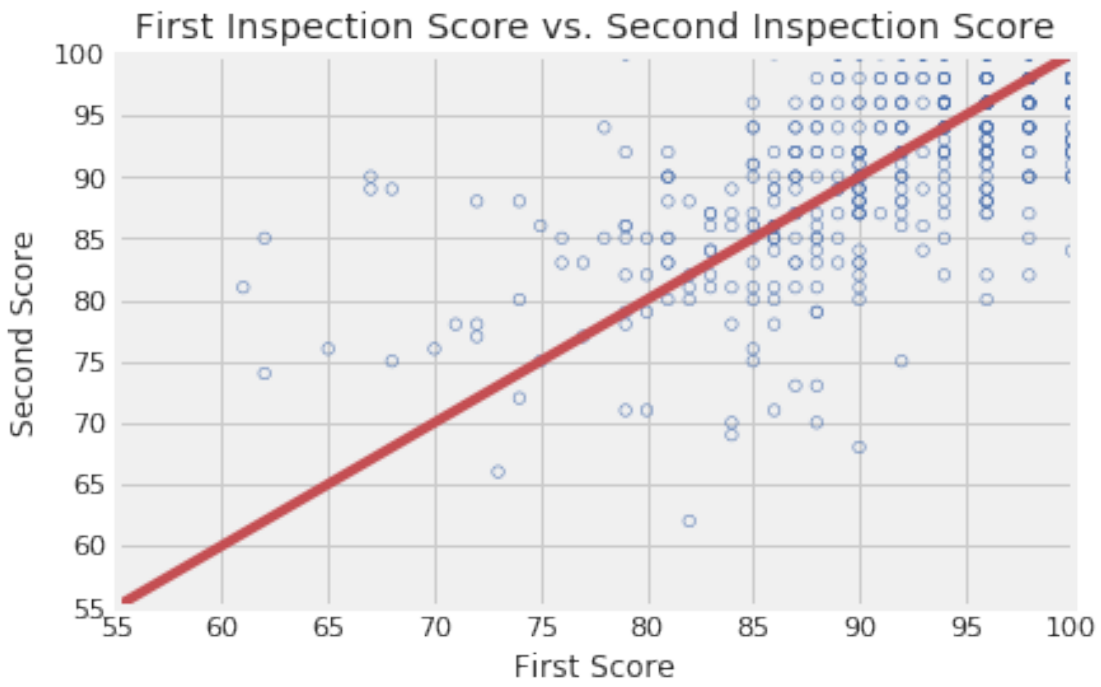
Describe the qualities of the distribution of the inspections scores based on your bar plot. Consider the mode(s), symmetry, tails, gaps, and anomalous values. Are there any unusual features of this distribution? What do your observations imply about the scores?

The mode of is around 100 meaning that businesses are doing pretty well in terms of inspections. The majority of businesses had inspection scores of above 90. But there is a left tail that there exists several businesses that are dpong poorly. So the average scores for San Francisco restaurants would be loere as a result.

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.

Type your answer here, replacing this text.

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.

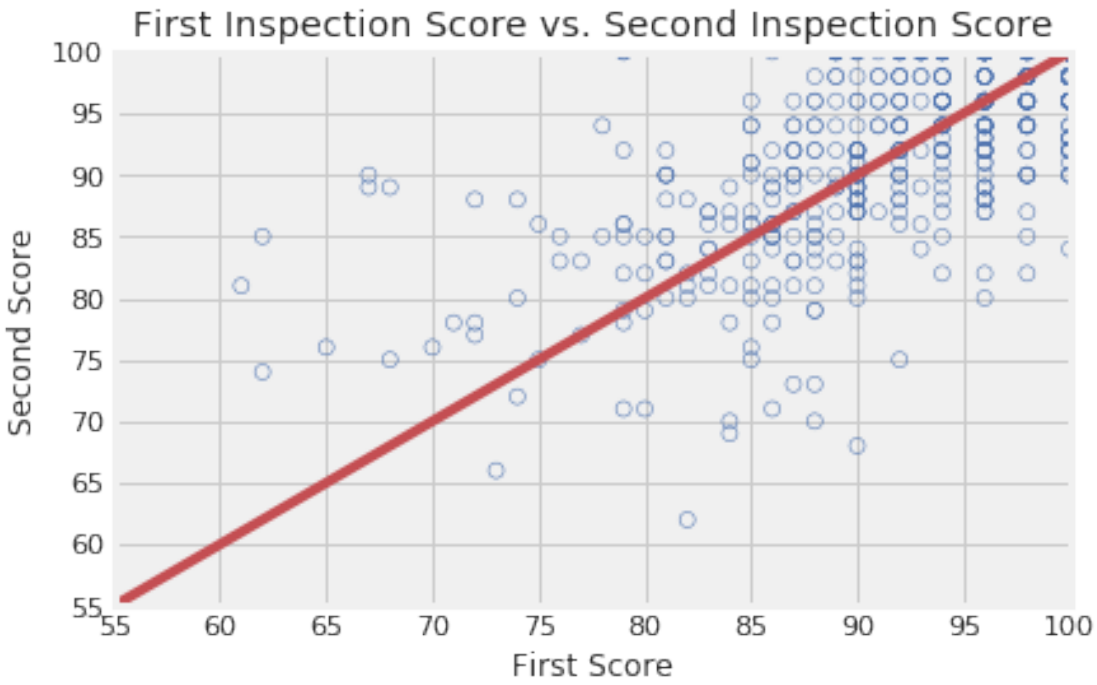
`plt.plot` for the reference line.

`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 [241]: plt.xlabel("First Score")
          plt.ylabel("Second Score")
          plt.title("First Inspection Score vs. Second Inspection Score")
          plt.axis([55, 100, 55, 100])
          plt.scatter(scores_pairs_by_business['score_pair'].str[0], scores_pairs_by_business['score_pa
          plt.plot(np.arange(55,101), np.arange(55,101), color='r')
```

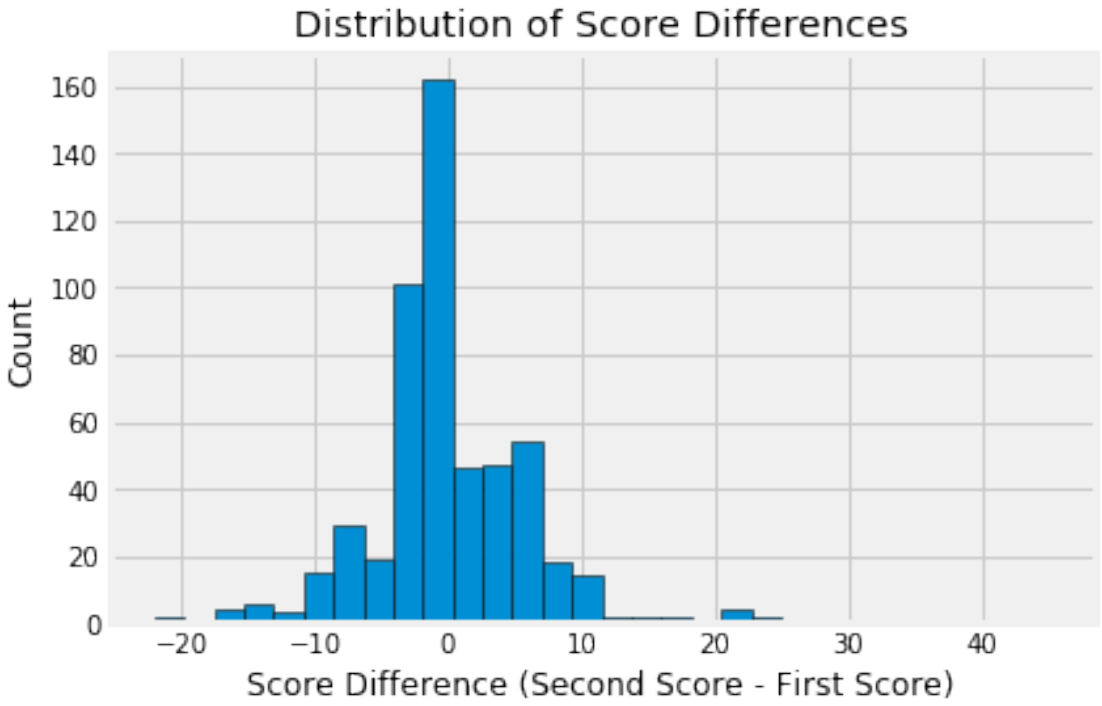
```
Out[241]: [<matplotlib.lines.Line2D at 0x7fade4ae0290>]
```


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

0.1.2 Question 7d

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.

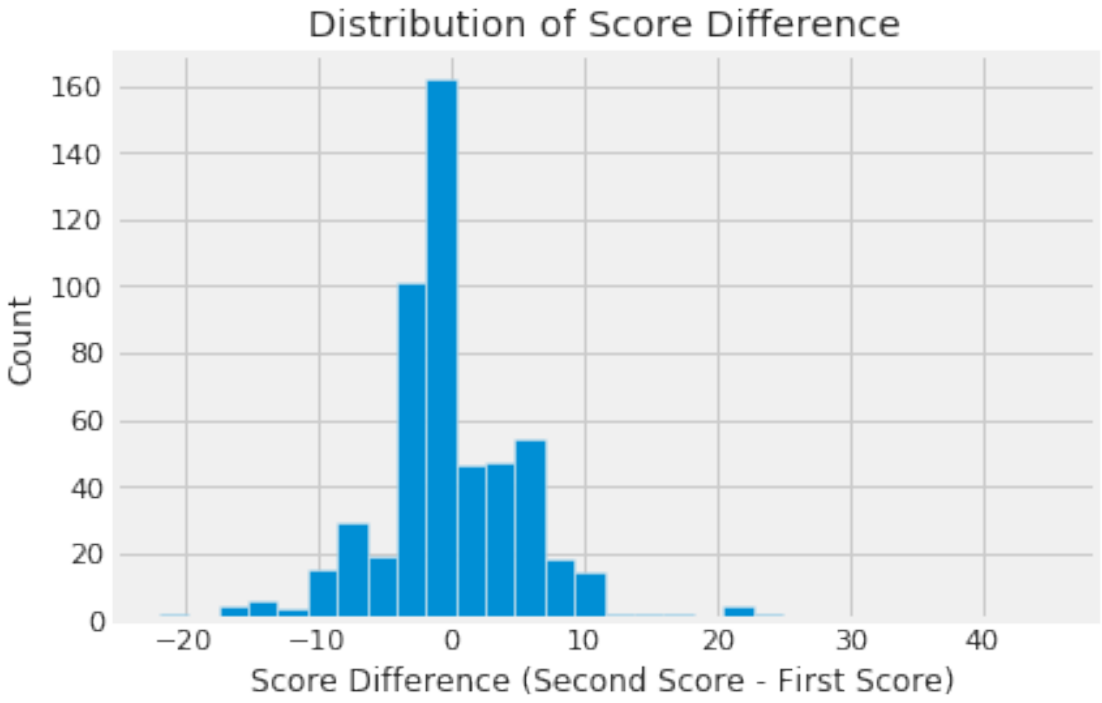
Hint: Use `plt.hist()` Try changing the number of bins when you call `plt.hist()`.

```
In [262]: plt.xlabel("Score Difference (Second Score - First Score)")
          plt.ylabel("Count")
          plt.title("Distribution of Score Difference")
```

```
first_score = scores_pairs_by_business['score_pair'].str[0]
second_score = scores_pairs_by_business['score_pair'].str[1]
score_diff = second_score - first_score

plt.hist(score_diff, bins=30)
```

Out[262]:
```
(array([  2.,   1.,   4.,   6.,   3.,  15.,  29.,  19., 101., 162.,  46.,
         47.,  54.,  18.,  14.,   2.,   2.,   2.,   1.,   4.,   2.,   0.,
          0.,   0.,   0.,   0.,   0.,   0.,   0.,   1.]),
 array([-22.        , -19.76666667, -17.53333333, -15.3       ,
        -13.06666667, -10.83333333,  -8.6       ,  -6.36666667,
         -4.13333333,  -1.9       ,   0.33333333,   2.56666667,
          4.8       ,   7.03333333,   9.26666667,  11.5       ,
         13.73333333,  15.96666667,  18.2       ,  20.43333333,
         22.66666667,  24.9       ,  27.13333333,  29.36666667,
         31.6       ,  33.83333333,  36.06666667,  38.3       ,
         40.53333333,  42.76666667,  45.        ]),
 <a list of 30 Patch objects>)
```



0.1.3 Question 7e

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 resturants' scores tend to improve, I would expect to see more plots above the red line because the y axis represents the second score, while the x axis represents the first score. While I observe that many hover near the red line, there appears to be more plots near the top right, especially right below the red line. Since the scatter plot below the line appears to be more dense, this does not meet up to the expecctation.

0.1.4 Question 7f

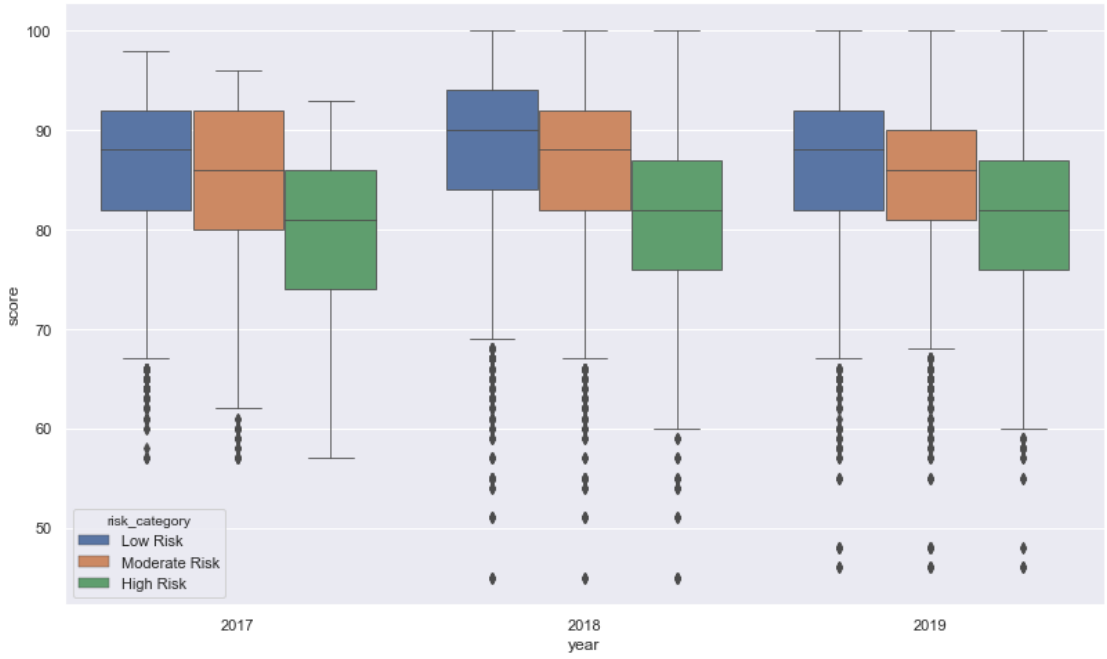
If a restaurant's score improves from the first to the second inspection, how would this be reflected in the histogram of the difference in the scores that you made in question 8d? What do you oberve from the plot? Are your observations consistent with your expectations? Explain your observations in the language of Statistics: for instance, the center, the spread, the deviation etc.

If the resturants' scores improve from the first to second inspection, there should be positive score differences. Therefore, you can expect to see more on the right-side of the histogram (left skewed). However, when you look at the the histogram, you see a large amount of score-differences landing right below zero and making the histogram slightly right skewed. This doesn't meet the expectation of having a graph that has more positive score differences and left skewed graph.

0.1.5 Question 7g

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!

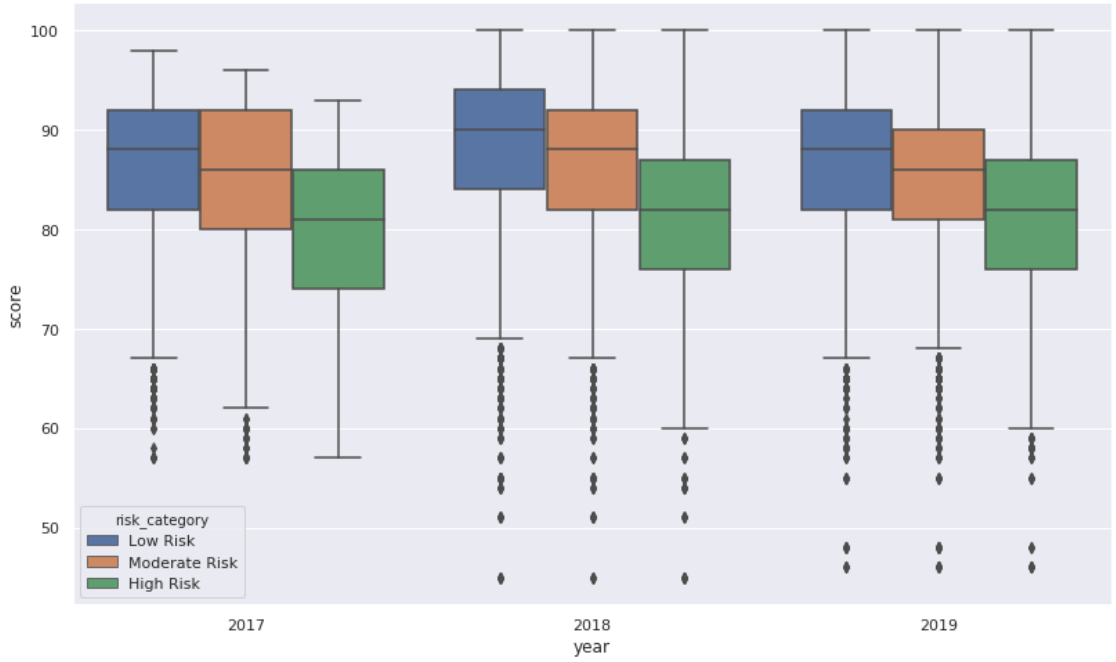
Hint: Use `plt.figure()` to adjust the figure size of your plot.

```
In [288]: # Do not modify this line
          sns.set()

          table1 = ins2vio.merge(vio, on='vid')
          table2 = table1.merge(ins_named, on = 'iid')
          table2 = table2[(table2['year'] != 2016) & (table2['Missing Score'] == False)]
          plt.figure(figsize=(12,8))


          sns.boxplot(x = 'year', y = 'score', hue = 'risk_category', data = table2 , hue_order = ['Low
```

Out[288]: <matplotlib.axes._subplots.AxesSubplot at 0x7fadcba19a10>



1 8: Open Ended Question

1.1 Question 8a

1.1.1 Compute Something Interesting

Play with the data and try to compute something interesting about the data. Please try to use at least one of groupby, pivot, or merge (or all of the above).

Please show your work in the cell below and describe in words what you found in the same cell. This question will be graded leniently but good solutions may be used to create future homework problems.

Please have both your code and your explanation in the same one cell below. Any work in any other cell will not be graded.

In [490]:
```
enter =['Special Event', 'Complaint Reinspection/Followup', 'New Construction',
        'Structural Inspection', 'Routine - Scheduled', 'Non-inspection site visit',
        'New Ownership - Followup', 'Foodborne Illness Investigation', 'New Ownership',
        'Reinspection/Followup', 'Complaint','Routine - Unscheduled']
ref = pd.DataFrame(enter, risk['Total Type'])
#First merging tables to get the risks and inspection types together in a single table
vio_merged = vio.merge(ins2vio, how='left', on = 'vid')
type_vio = vio_merged.merge(ins, how='left', on = 'iid')
risk = type_vio.pivot_table(index = 'type', columns = 'risk_category', aggfunc = 'size')

risk = risk.fillna(0)
risk['Total Type'] = risk.sum(axis=1)
risk = risk.sort_values(by='Total Type')



risk
ref



#Created a dataframe based on the level of risk of each type category.
#Made a quantative chart of how many type of risk are in each level and create a new column t
#see how many are done total in each category. Did this by merging the ins2vio and vio on to
#vio and then merged that with the ins. And then created a pivot table with the type as the r
#used each risk category as the columns for pivot table
```

Out[490]:

	0
Total	Type
7.0	Special Event
12.0	Complaint Reinspection/Followup
12.0	New Construction
25.0	Structural Inspection
47.0	Routine - Scheduled
62.0	Non-inspection site visit
66.0	New Ownership - Followup
122.0	Foodborne Illness Investigation
480.0	New Ownership
554.0	Reinspection/Followup
1304.0	Complaint
37519.0	Routine - Unscheduled

1.1.2 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 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** (1-3 points): A chart is produced but with some flaws such as bad encoding. The text written is incomplete but makes some sense.
- **Unsatisfactory** (0 points): No chart is created, or a chart with completely wrong results.

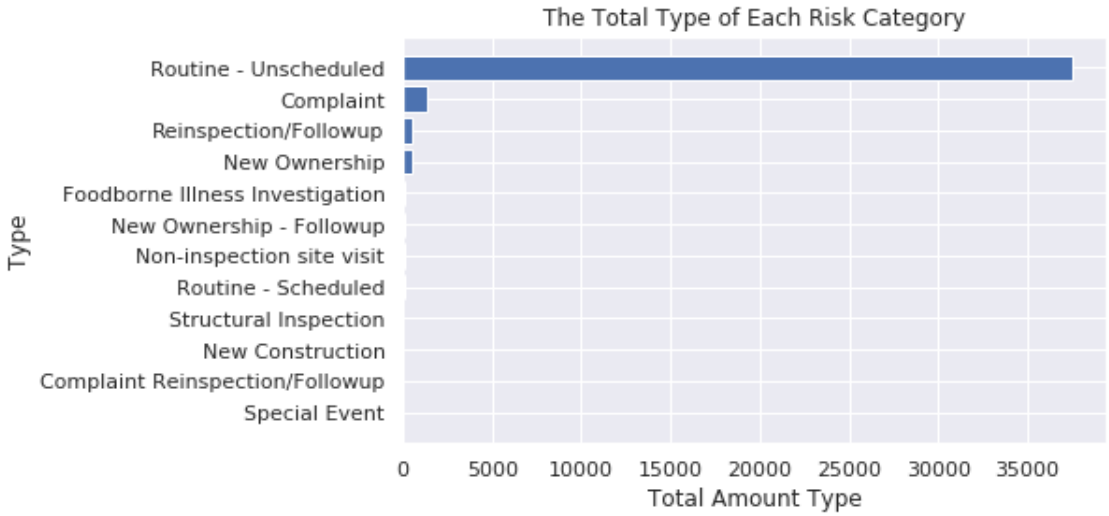
We will lean towards being generous with the grading. We might also either discuss in discussion or post on Piazza some examplar analysis you have done (with your permission)!

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 [492]:
```
# YOUR DATA PROCESSING AND PLOTTING HERE
plt.title('The Total Type of Each Risk Category')
plt.xlabel('Total Amount Type')
plt.ylabel('Type')
temp = type_vio['type'].value_counts().keys().copy()
plt.barh(enter, risk['Total Type'])
# Created a bar graph to show each type and the the category of the risk level that each type
```

Out[492]:
```
<BarContainer object of 12 artists>
```


The Total Type of Each Risk Category
Routine - Unscheduled
Complaint
Reinspection/Followup
New Ownership
Foodborne Illness Investigation
New Ownership - Followup
Non-inspection site visit
Routine - Scheduled
Structural Inspection
New Construction
Complaint Reinspection/Followup
Special Event
Type
0
5000
10000
15000
20000
25000
30000
35000
Total Amount Type