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 [97]: display(bus.head(), ins.head(), ins2vio.head(), vio.head())
```

```
   business id column                          name                address  \
0               1000        HEUNG YUEN RESTAURANT            3279 22nd St
1             100010        ILLY CAFFE SF_PIER 39        PIER 39  K-106-B
2             100017  AMICI'S EAST COAST PIZZERIA             475 06th St
3             100026               LOCAL CATERING        1566 CARROLL AVE
4             100030            OUI OUI! MACARON  2200 JERROLD AVE STE C

           city state postal_code     latitude    longitude  phone_number
0  San Francisco    CA       94110    37.755282  -122.420493         -9999
1  San Francisco    CA       94133 -9999.000000 -9999.000000   14154827284
2  San Francisco    CA       94103 -9999.000000 -9999.000000   14155279839
3  San Francisco    CA       94124 -9999.000000 -9999.000000   14155860315
4  San Francisco    CA       94124 -9999.000000 -9999.000000   14159702675
```

```
              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
```

```
             iid     vid
0  97975_20190725  103124
1  85986_20161011  103114
2  95754_20190327  103124
3  77005_20170429  103120
4   4794_20181030  103138
```

```
                                      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
```

We can see in the business dataset that we have a phone number with the corresponding value of -9999. As phone numbers are not released in this format, we can determine that -9999 was using in place of Null or NaN values, which can cause issues in visualizing and interpreting the data.

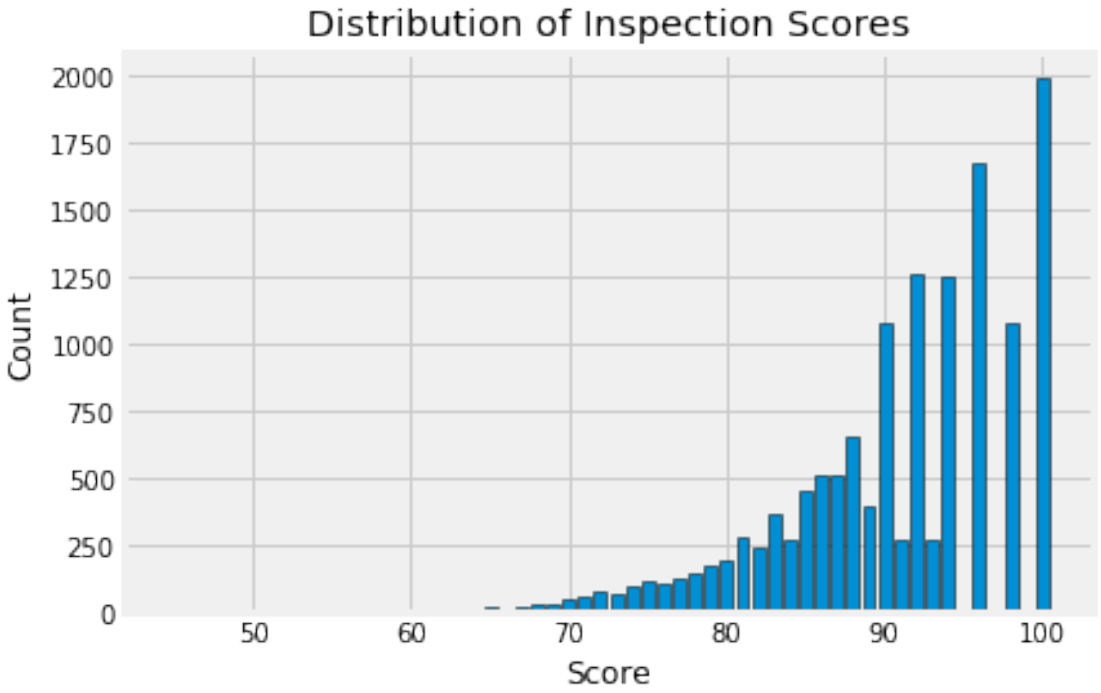
In the cell below, write the name of the restaurant with the lowest inspection scores ever. You can also head to yelp.com and look up the reviews page for this restaurant. Feel free to add anything interesting you want to share.

Lollipot

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.



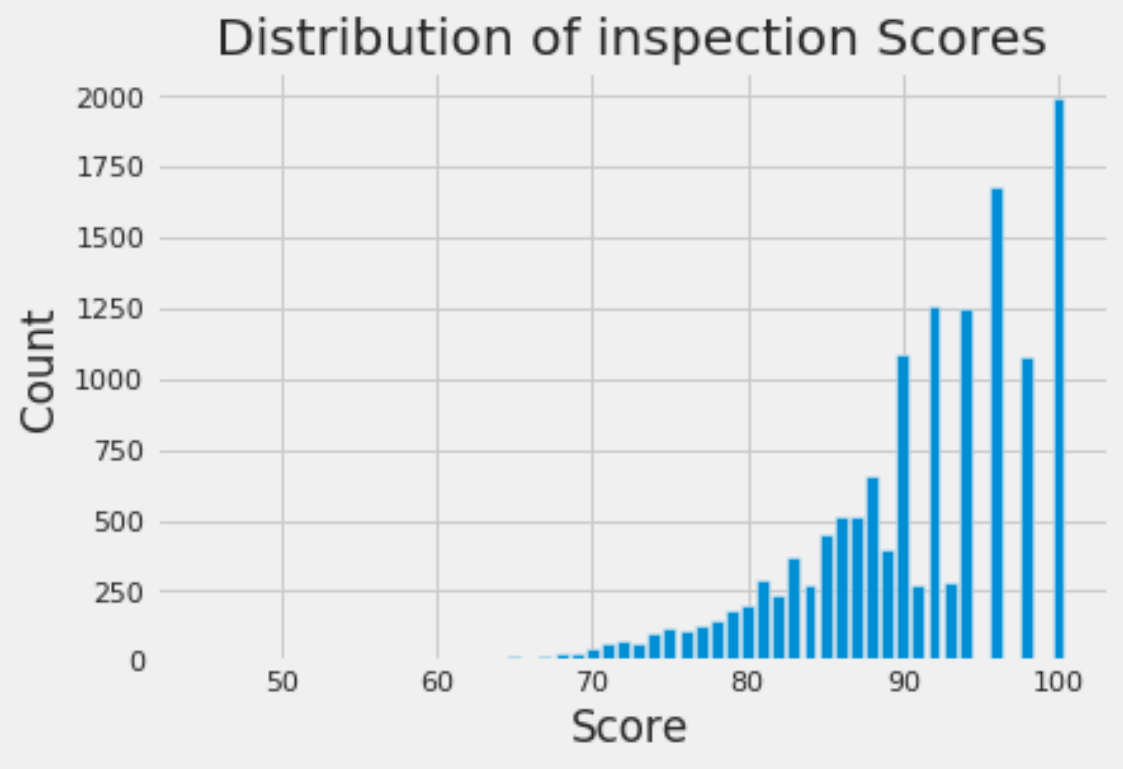
You might find this matplotlib.pyplot tutorial useful. Key syntax that you'll need:

```
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 [215]:
```
plt.bar(ins['score'].value_counts().keys(), ins['score'].value_counts())
plt.xlabel("Score")
plt.ylabel("Count")
plt.title("Distribution of inspection Scores")
```

Out[215]:
```
Text(0.5, 1.0, 'Distribution of inspection Scores')
```

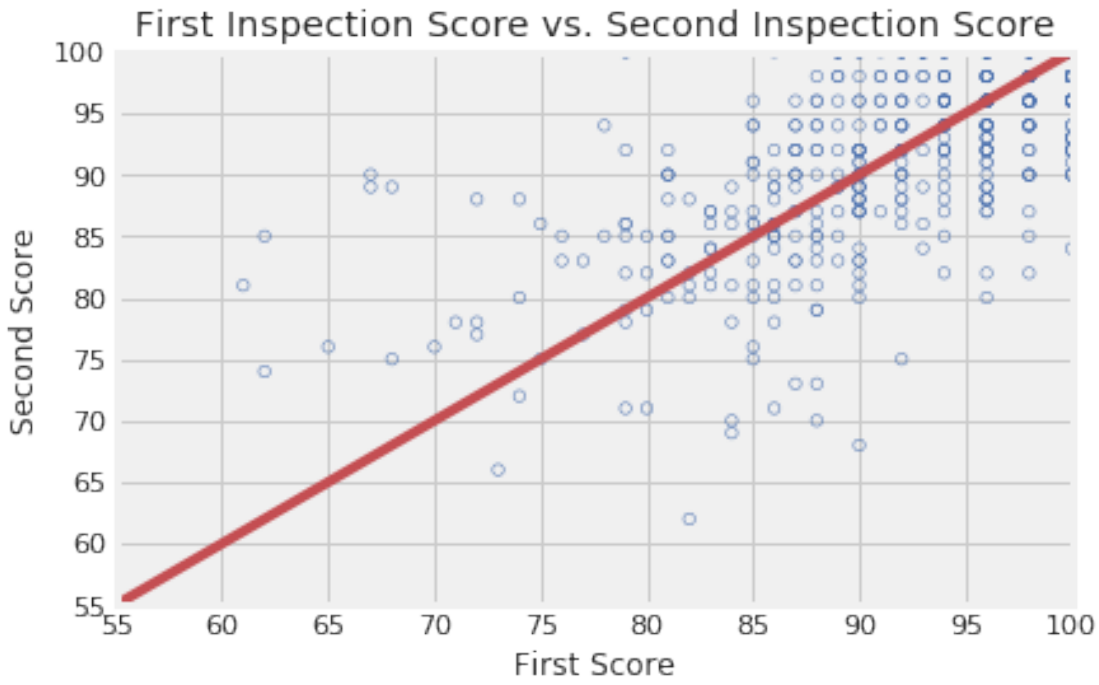


0.1.1 Question 6b

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 bar plot has a heavy skew to the left with a mode of 100 and a lot of gaps in between 95 to a 100. This can be due to the fact that many give out a 100 instead of a 99, thus due to the scoring process we have affinities for some values and not others.Another thing to notice is how the count levels off as the scores get lower. This could be due to lower scored restaurants going out of business.

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.

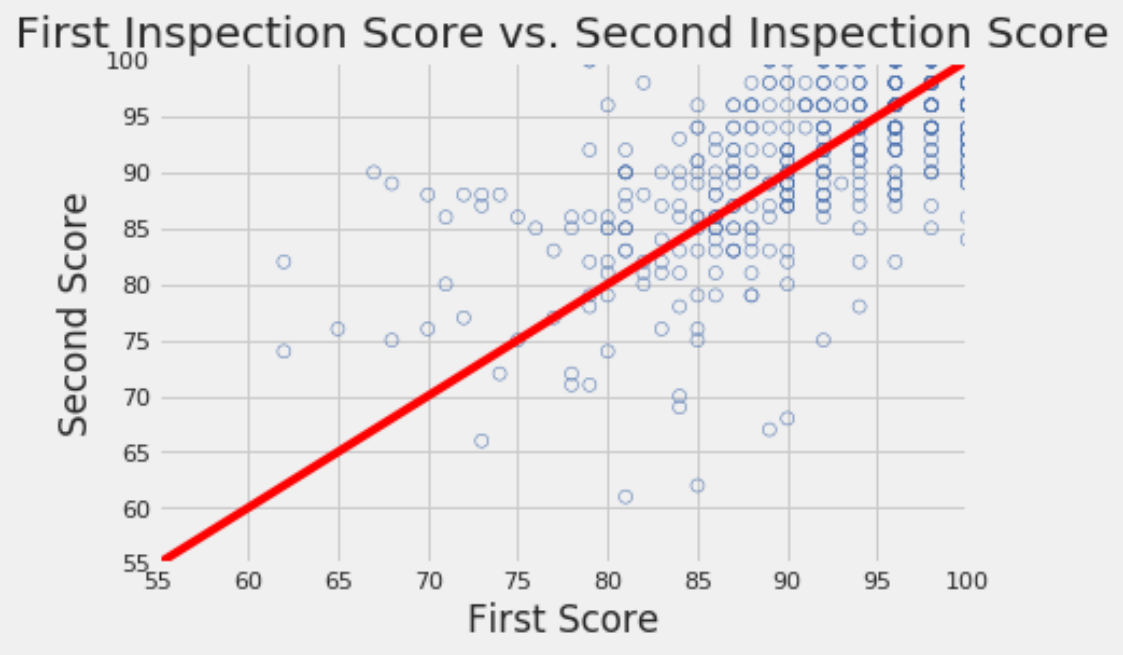
```
In [363]: a = []
          b = []

          for val in scores_pairs_by_business["score_pair"]:
              a.append(val[0])
              b.append(val[1])


          plt.scatter(a,b , facecolors='none', edgecolors = 'b')
          plt.plot([55,100], [55,100], color = 'red')
          plt.xlabel('First Score')
```

```
plt.ylabel('Second Score')
plt.title('First Inspection Score vs. Second Inspection Score')
plt.axis(xmin = 55 , xmax = 100, ymin = 55, ymax = 100)
```

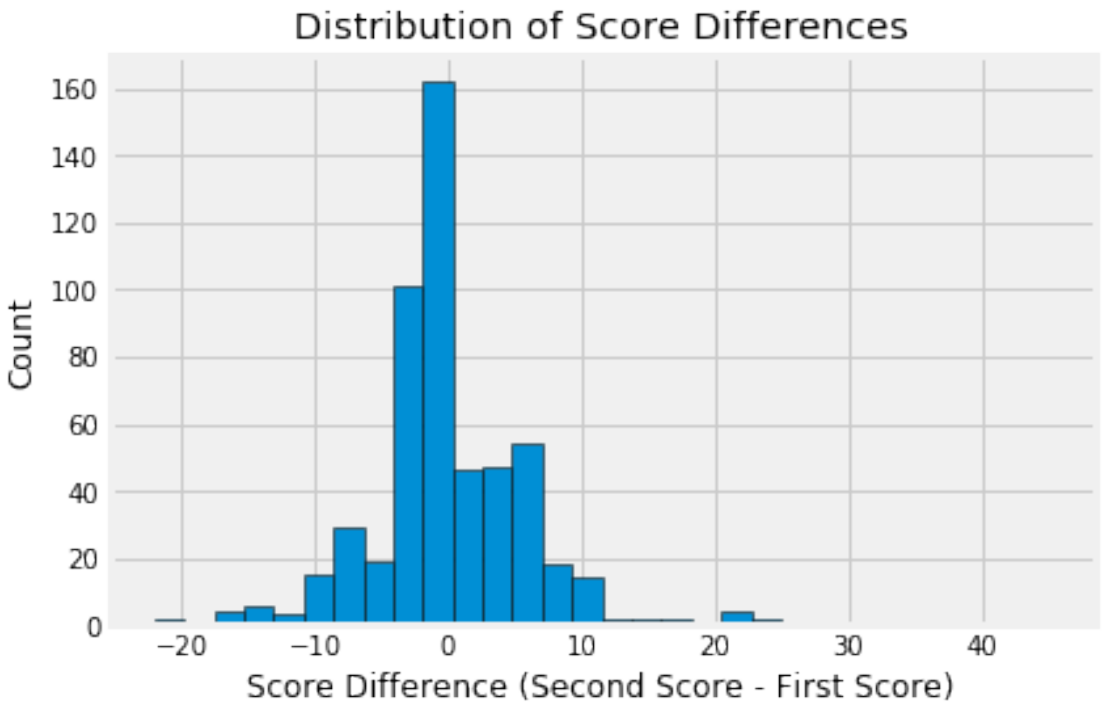
Out[363]: (55, 100, 55, 100)



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.

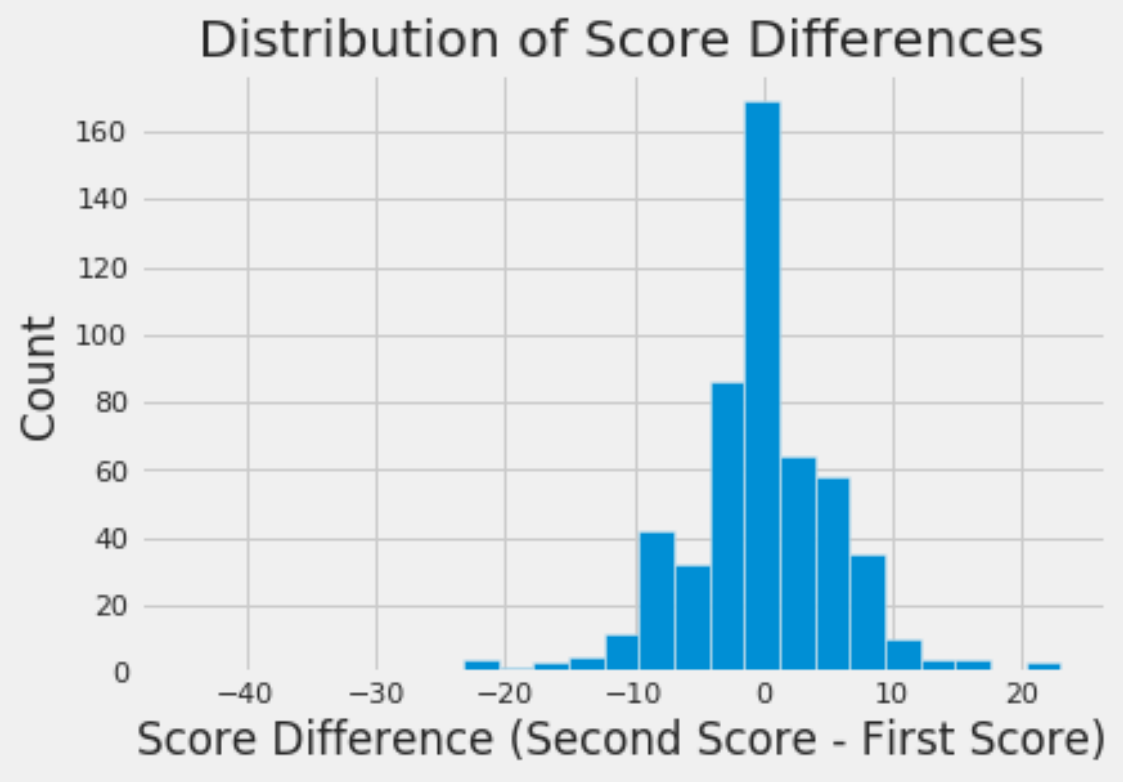
Hint: Use `plt.hist()` Try changing the number of bins when you call `plt.hist()`.

```
In [366]: c= []
          for i in range(len(a)):
              c.append(a[i] - b[i])

          plt.hist(c, bins = 25)
```

```
plt.xlabel('Score Difference (Second Score - First Score)')
plt.ylabel('Count')
plt.title('Distribution of Score Differences')
```

Out[366]: Text(0.5, 1.0, 'Distribution of Score Differences')



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

Hint: What does the slope represent?

The slope represents a restaurant that did not change its score from the first check to the second check. If values lie above the line then the restaurant improved while if the value fell below the line it had gotten worse. The scatter plot has values seemingly even across the line, seeming that there were an equal amount of restaurants that improved and gotten worse.

0.1.4 Question 7f

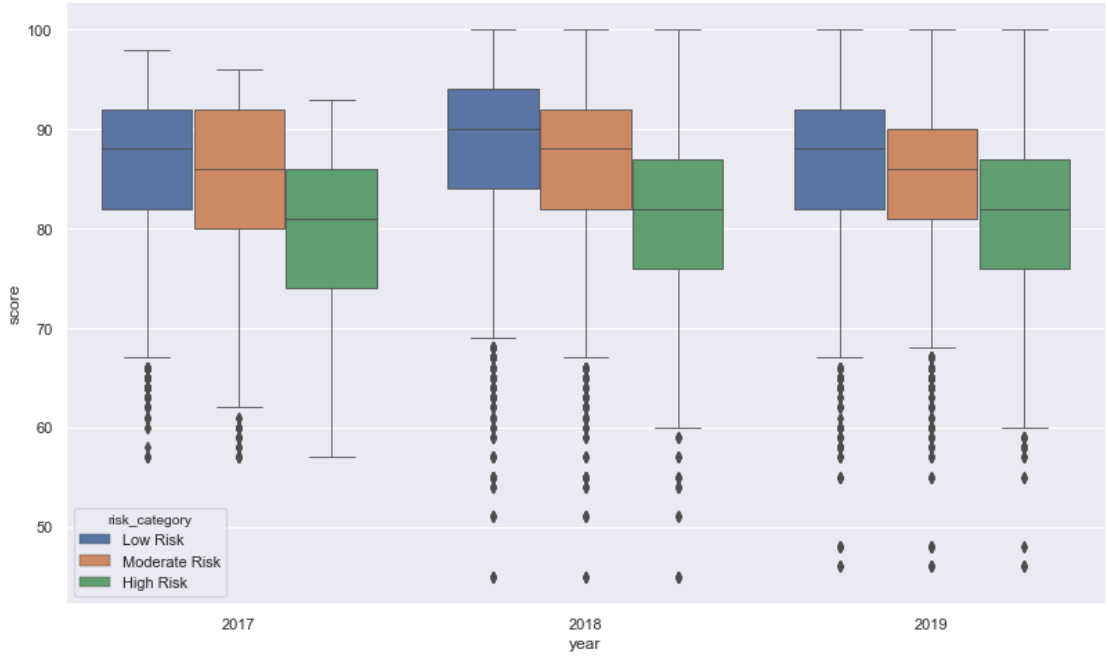
If a restaurant's score improves from the first to the second inspection, how would this be reflected in the histogram of the difference in the scores that you made in question 7d? What do you oberve from the plot? Are your observations consistent with your expectations? Explain your observations in the language of Statistics: for instance, the center, the spread, the deviation etc.

A distribution with a mean above 0 would mean that most restaurants had improved while a mean below 0 would mean most restaurants had gotten worse. A mean around 0 would mean that there was almost an even split. The plot seems to have an almost even split as the mean/center of the distribution is around 0. There is almost no skew as the graph has the same positive and negative deviation.

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!

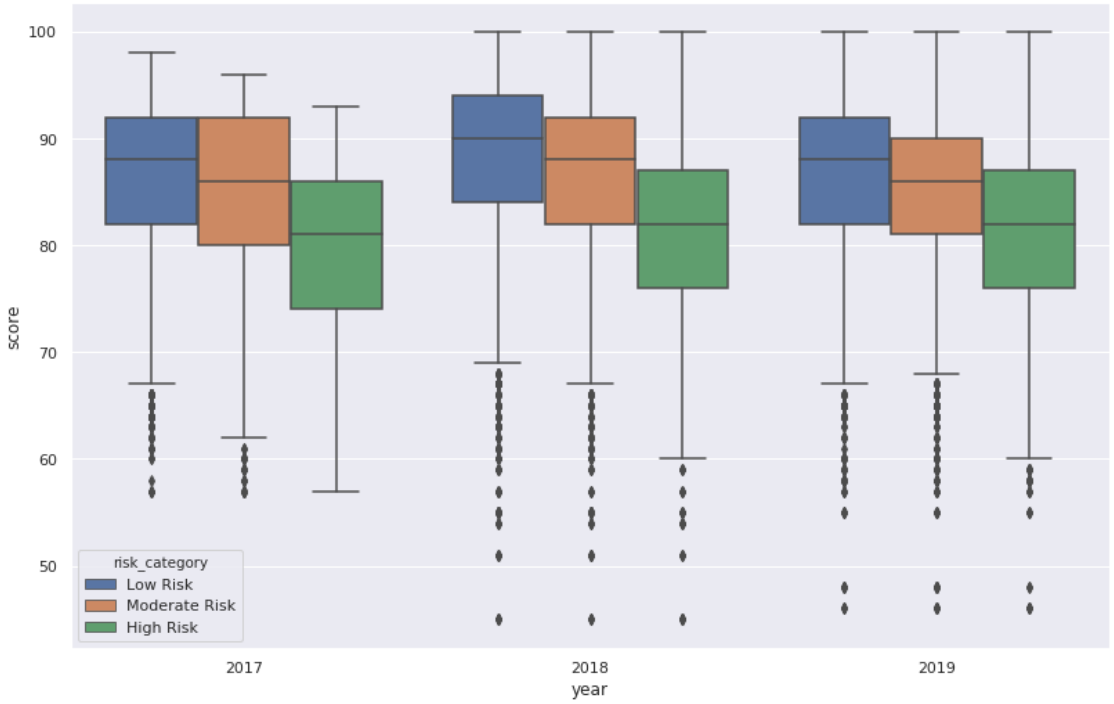


Hint: Use `sns.boxplot()`. Try taking a look at the first several parameters. The documentation is linked here!

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

```
In [391]: # Do not modify this line
          sns.set()
          new = pd.merge(ins[ins["year"] != 2016], ins2vio, on = "iid", how = "left")
          new= pd.merge(new, vio, on = "vid", how = "left")[["score", "year", "risk_category"]]
          plt.figure(figsize=(12,8))
          sns.boxplot(x = "year", y = "score", hue = "risk_category", data = new, hue_order = ['Low Risk
```

```
Out[391]: <matplotlib.axes._subplots.AxesSubplot at 0x7fe537b64250>
```


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

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.

1.1.2 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): Uses a combination of pandas operations (such as groupby, pivot, merge) to answer a relevant question about the data. The text description provides a reasonable interpretation of the result.
- **Passing** (1-3 points): Computation is flawed or very simple. The text description is incomplete but makes some sense.
- **Unsatisfactory** (0 points): No computation is performed, or a computation with completely wrong results.

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 [426]: seasons = pd.DataFrame({"month": [12, 1, 2, 3,4,5,6,7,8,9,10,11],
                                  "Seasons": ["Winter","Winter","Winter","Spring","Spring","Spring","Su

          ins["month"] = ins["timestamp"].dt.month
          ins_by_season = pd.merge(ins, seasons, how="left", on = "month")

          ins_by_season.groupby("Seasons").agg(np.mean)["score"]
          #See the distribution based on seasons to answer questions such as when are we more likely to
          #Can potentially con
```

```
<ipython-input-426-407077ab859c>:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.h
  ins["month"] = ins["timestamp"].dt.month
```

```
Out[426]: Seasons
          Fall      91.012286
          Spring    91.071283
          Summer    89.734577
          Winter    90.672256
          Name: score, dtype: float64
```

1.1.3 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): 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 [ ]:
```
# YOUR DATA PROCESSING AND PLOTTING HERE

count_season = ins_by_season['Seasons'].value_counts()
count_season

plt.bar(ins['score'].value_counts().keys(), ins['score'].value_counts())
plt.xlabel("Score")
plt.ylabel("Count")
plt.title("Distribution of inspection Scores")


# YOUR EXPLANATION HERE (in a comment)
```