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.

From initially glancing at the data, it appears that there are several issues:

1. The `bus["business id column"]` Series has entries of inconsistent length. The `bus["name"]` Series does not appear to be canonicalised to a form that avoids punctuation and capitalisation, so we may have duplicate entries where the same business exists in both capitalised and not-capitalised forms.
2. The `vio` DataFrame contains negative phone numbers (which may represent missing data) in addition to outrageously out-of-bounds GPS coordinates, which also may represent missing data.

```
In [17]: for frame in all_frames[0:3]:
             print(frame.head(10), "\n")
```

```
   business id column                          name  \
0                1000        HEUNG YUEN RESTAURANT
1              100010        ILLY CAFFE SF_PIER 39
2              100017  AMICI'S EAST COAST PIZZERIA
3              100026               LOCAL CATERING
4              100030             OUI OUI! MACARON
5              100036              Hula Truck (#2)
6              100039     GENKI CREPES & MINI MART
7              100041               UNCLE LEE CAFE
8              100055                Twirl and Dip
9              100058                  SF PITA HUB

                       address           city state postal_code  \
0                  3279 22nd St  San Francisco    CA       94110
1                PIER 39  K-106-B  San Francisco    CA       94133
2                     475 06th St  San Francisco    CA       94103
3                1566 CARROLL AVE  San Francisco    CA       94124
4         2200 JERROLD AVE STE C  San Francisco    CA       94124
5                   2 Marina Blvd  San Francisco    CA       94123
6                 330 CLEMENT ST  San Francisco    CA       94118
7                 3608 BALBOA ST  San Francisco    CA       94121
8  335 Martin Luther King Jr. Dr  San Francisco    CA       94118
9                   475 06TH ST  San Francisco    CA       94103

      latitude    longitude  phone_number
0    37.755282  -122.420493         -9999
1 -9999.000000 -9999.000000   14154827284
2 -9999.000000 -9999.000000   14155279839
3 -9999.000000 -9999.000000   14155860315
4 -9999.000000 -9999.000000   14159702675
5 -9999.000000 -9999.000000         -9999
6 -9999.000000 -9999.000000   14155376414
7 -9999.000000 -9999.000000         -9999
8 -9999.000000 -9999.000000   14155300260
9 -9999.000000 -9999.000000   14155642006
```

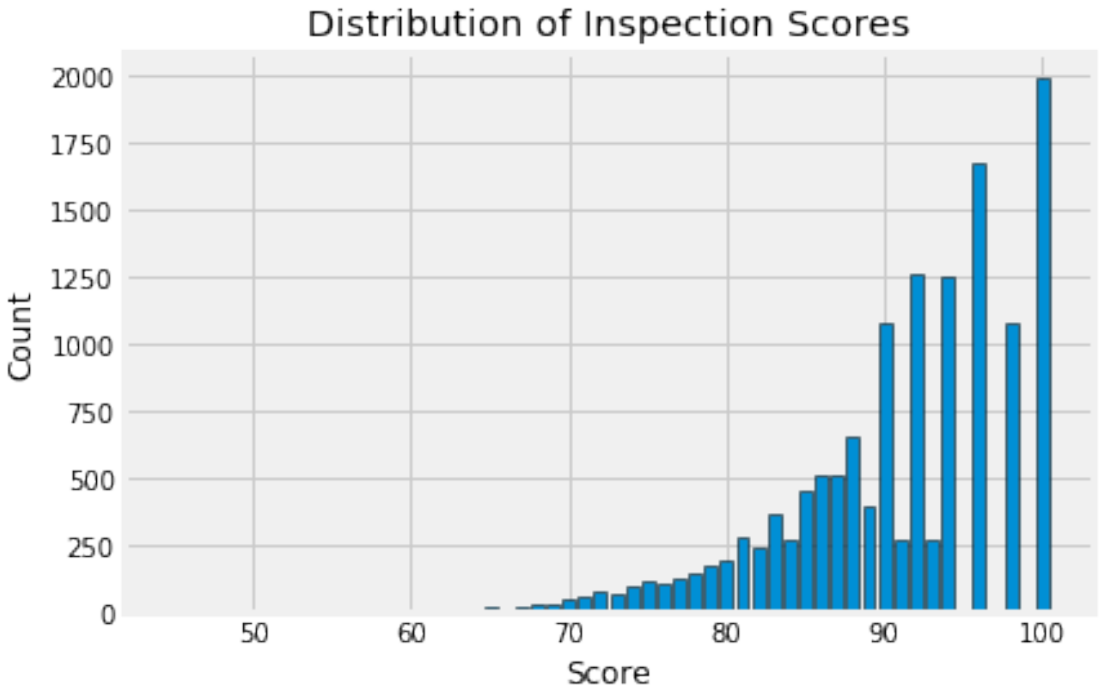
	iid	date	score	type
0	100010_20190329	03/29/2019 12:00:00 AM	-1	New Construction
1	100010_20190403	04/03/2019 12:00:00 AM	100	Routine - Unscheduled
2	100017_20190417	04/17/2019 12:00:00 AM	-1	New Ownership
3	100017_20190816	08/16/2019 12:00:00 AM	91	Routine - Unscheduled
4	100017_20190826	08/26/2019 12:00:00 AM	-1	Reinspection/Followup
5	100017_20190912	09/12/2019 12:00:00 AM	-1	Reinspection/Followup
6	100026_20190418	04/18/2019 12:00:00 AM	-1	New Ownership
7	100030_20190612	06/12/2019 12:00:00 AM	-1	New Ownership
8	100030_20190826	08/26/2019 12:00:00 AM	-1	New Ownership
9	100036_20190325	03/25/2019 12:00:00 AM	-1	Structural Inspection

	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
5	Food safety certificate or food handler card n…	Low Risk	103157
6	Foods not protected from contamination	Moderate Risk	103133
7	High risk food holding temperature	High Risk	103103
8	High risk vermin infestation	High Risk	103114
9	Improper cooking time or temperatures	High Risk	103106

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.

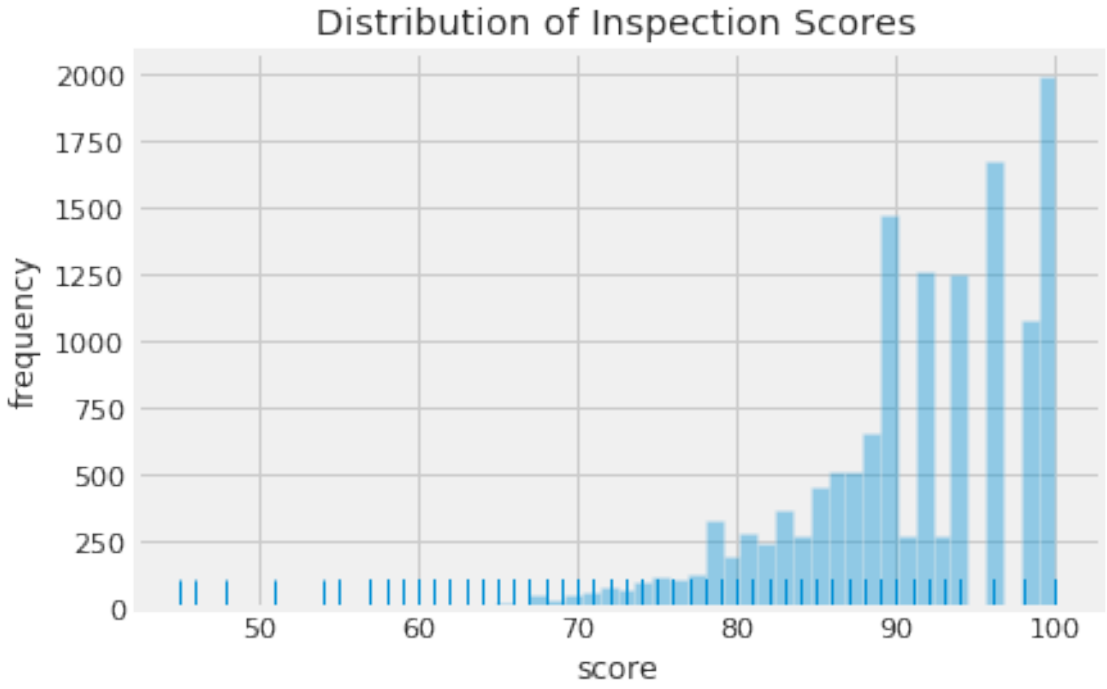


You might find this matplotlib.pyplot tutorial useful. Key syntax that you'll need:

```
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 [75]: (sns.distplot(ins[ins["score"] > 0]["score"],
                       norm_hist=False, kde=False, rug=True, label="hi")
              .set(xlabel="score",
                   ylabel="frequency",
                   title="Distribution of Inspection Scores"));
```



0.1.1 Question 6b

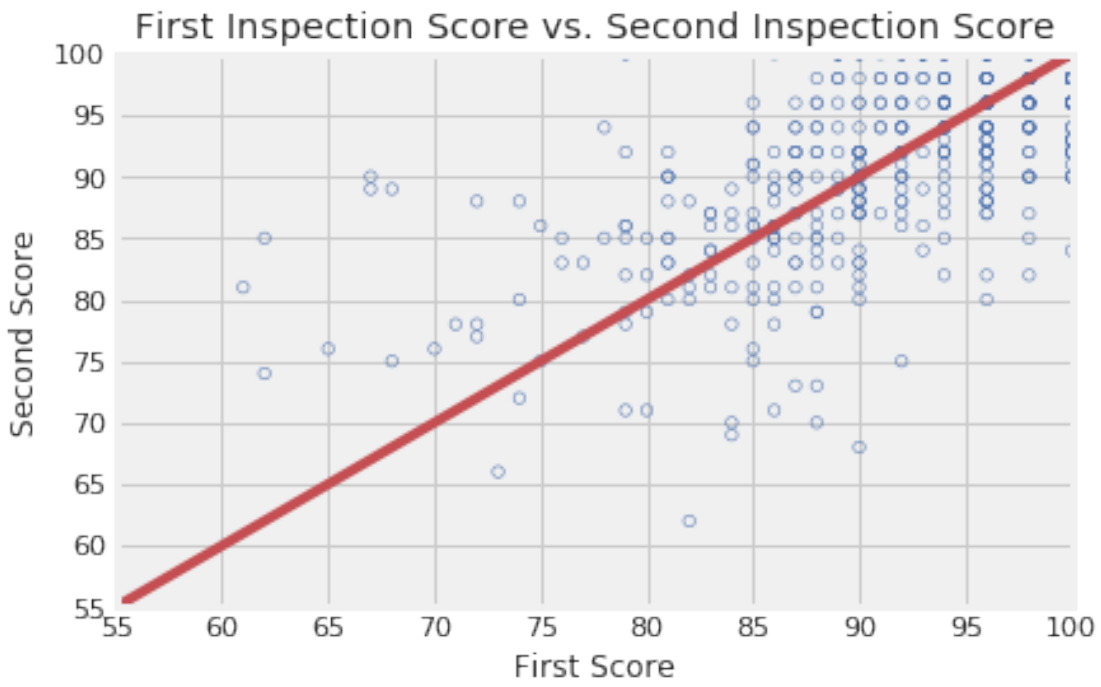
Describe the qualities of the distribution of the inspections scores based on your bar plot. Consider the mode(s), symmetry, tails, gaps, and anomalous values. Are there any unusual features of this distribution? What do your observations imply about the scores?

It appears that the distribution of scores has a modes at (roughly) 90, 96, and 100. It seems to be skewed heavily left, and increases steadily and gaplessly before about 90, after which there are quite severe gaps in every other "bucket."

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 worst restaurant is Lollipot. According to Yelp, it was closed in 2018 for consistently violating cleanliness standards for food contact surfaces and utensils. Additionally, it also stored food at unsafe temperatures.

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:

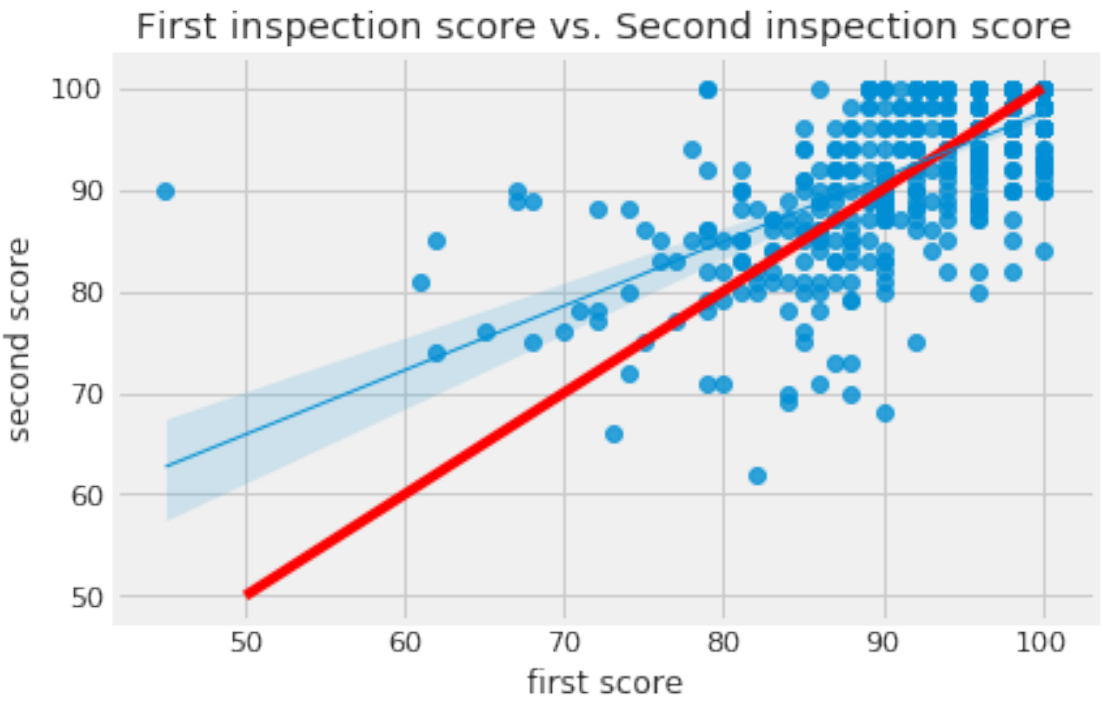
`plt.scatter` plots a set of points. Use `facecolors='none'` and `edgecolors=b` to make circle markers with blue borders.

`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.

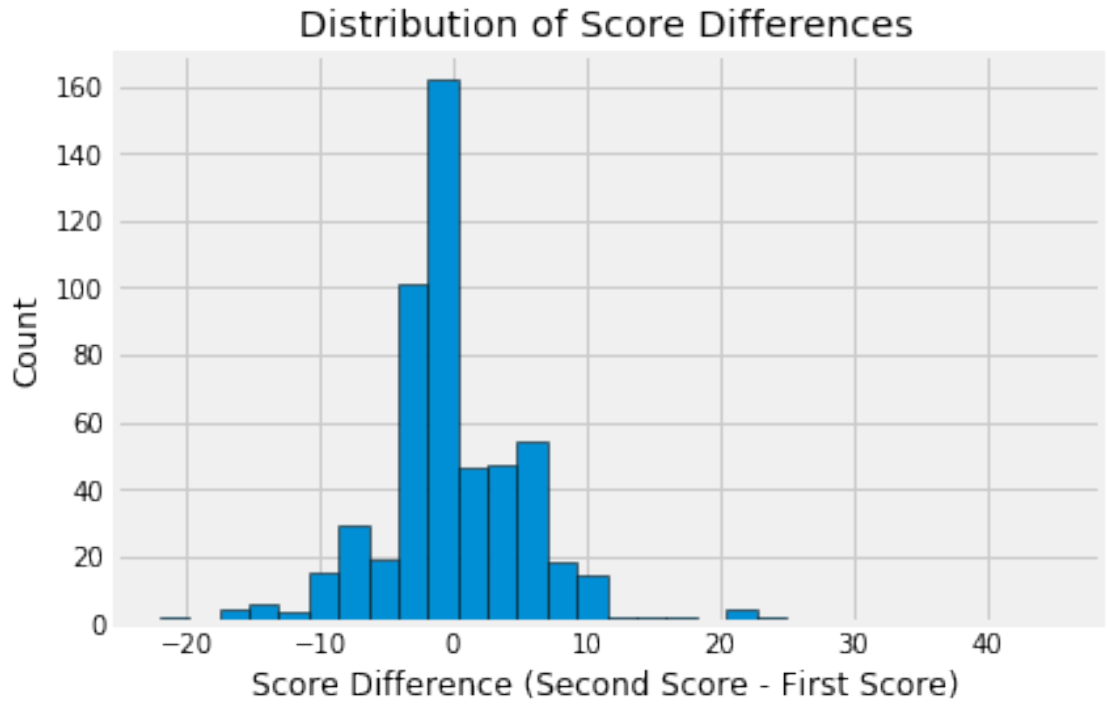
```
In [86]: scores_unzipped = list(zip(*scores_pairs_by_business["score_pair"].tolist()))
         axs = sns.lineplot(x=[50, 100], y=[50, 100], color="red")
         (sns.regplot(x=scores_unzipped[0], y=scores_unzipped[1], ax=axs, line_kws={"linewidth":1})
             .set(xlabel="first score", ylabel="second score",
                  title="First inspection score vs. Second inspection score"));
```


First inspection score vs. Second inspection score
second score
first score
100
90
80
70
60
50
50
60
70
80
90
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:

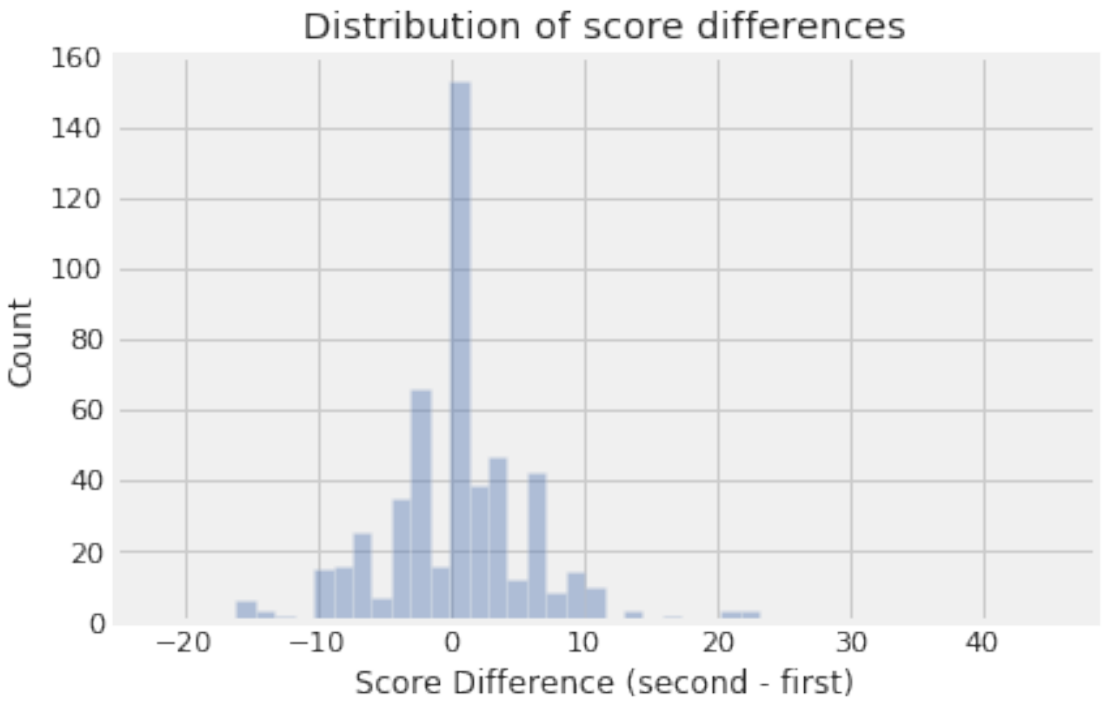


Hint: Use `second_score` and `first_score` created in the scatter plot code above.

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 [87]: scores_diff = np.array(scores_unzipped[1]) - np.array(scores_unzipped[0])
         (sns.distplot(scores_diff, norm_hist=False, kde=False, color="b")
             .set(xlabel="Score Difference (second - first)",
                  ylabel="Count",
                  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 - first)

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 scores tended to improve, we should see lower x values correlate with higher y values across the board, so we would have a shallower positive slope close to the top of the graph. When we compare the linear regression line (blue) to the reference 1-by-1 slope (red) we see that there is a slightly higher second score.

I would have expected a far higher increase than the data suggests, because (as a layman) it seems that for a restaurant to require multiple inspections per year, there would have to be some egregious issue that needed to be rectified. The data is thus not consistent with my expectation.

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 scores tended to improve, we would see a histogram with a majority positive values, ie. a mode higher than zero.

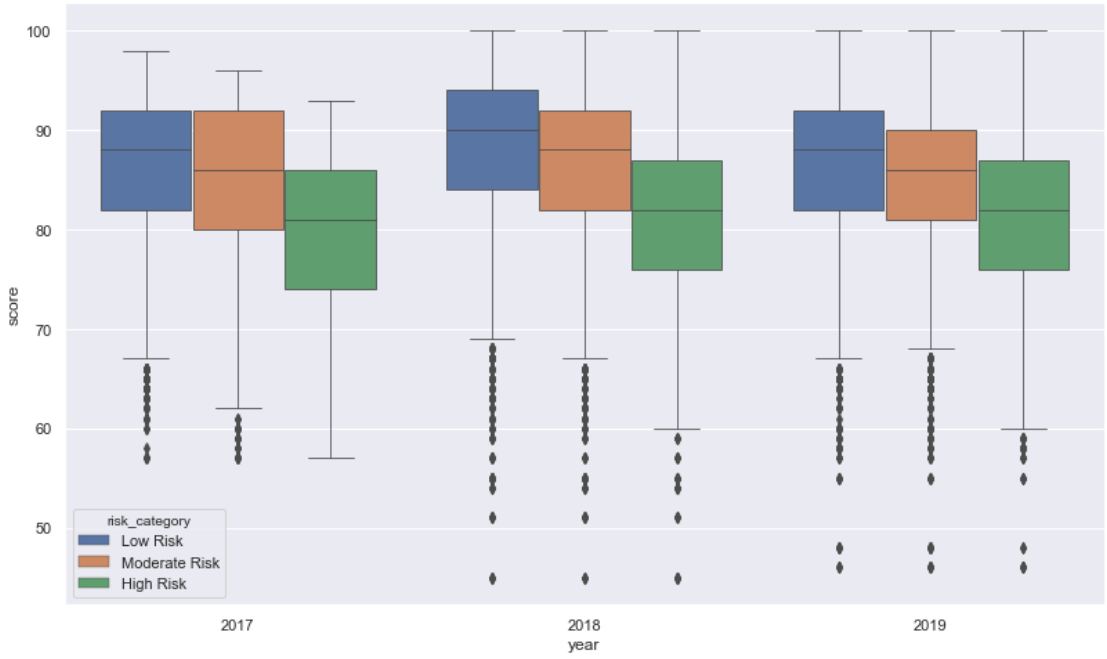
Looking at the scatter plot, there does not appear to be such a mode: the distribution is unimodal and centred at zero, with roughly symmetric spread to either side, albeit slightly more extreme outliers to the left. Thus if we take the outliers into account, we can explain why the regression in 7c points to a slightly higher second score.

Again, I would have expected a far higher increase than the data suggests, because (as a layman) it seems that for a restaurant to require multiple inspections per year, there would have to be some egregious issue that needed to be rectified. The data is thus not consistent with my expectation.

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.

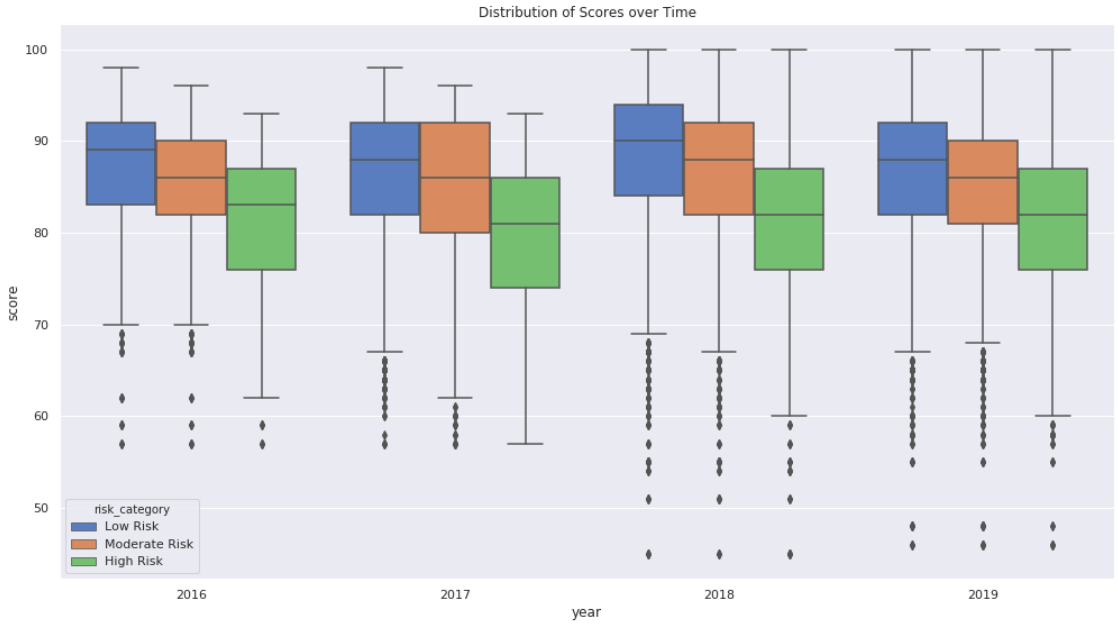
```
In [88]: # Do not modify this line
         sns.set()

         f, axes = plt.subplots(2, 1, figsize=(15, 20))
         ordering = ["Low Risk", "Moderate Risk", "High Risk"]
         yrs_df = (vio.merge(ins2vio, on="vid")
                   .merge(ins[ins["score"] > 0][["iid", "score", "year"]], on="iid")
                   [["year", "risk_category", "score"]])

         ax = (sns.boxplot(x="year", y="score", hue="risk_category",
```

```
                    data=yrs_df, hue_order=ordering, palette="muted", ax=axes[0])
      .set(title="Distribution of Scores over Time"))

ax = (sns.violinplot(x="year", y="score", hue="risk_category",
                     data=yrs_df, hue_order=ordering, palette="muted", ax=axes[1])
      .set(title="Distribution of Scores over Time (with KDE)"))
```



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 [89]:
```
from scipy.stats import linregress

def re_index(df):
    df["acc"] = np.arange(len(df))
    return df

def get_r(df):
    _, _, r, _, _ = linregress(x=df["timestamp"].astype(np.int64),
                               y=df["acc"].astype(np.int64))
    return r

desc_time = (vio
             .merge(ins2vio, on="vid")
             .merge(ins, on="iid")[["description", "timestamp"]]
             .sort_values(by="timestamp", ascending=True))
desc_time["acc"] = 0
desc_time = desc_time.groupby("description").apply(re_index)
r_coefs = desc_time.groupby("description").apply(get_r)
r_coefs.sort_values()

# I tried to check if there was a linear correlation between
# the timestamp and the accumulated number of violations,
# grouped by the violation description. Using
# scipy.stats.linregress, I calculated the correlation
# coefficient for each group, casting the timestamp to its
# Unix epoch representation. From the results, which are
# remarkably close to +1.0, we can see that there is a
# fairly strong positive linear correlation between the
```

```
        # timestamp and the accumulated number of violations for
        # each group. Then it follows that each violation is given
        # out at approximately consistent rates. There are
        # exceptions for the violations which focus more mobile food
        # facilities eg. food trucks, which might be because
        # inspections thereof occur on a less fixed interval.
```

```
/srv/conda/envs/data100/lib/python3.7/site-packages/scipy/stats/_stats_mstats_common.py:130: RuntimeWarn
  slope = r_num / ssxm
/srv/conda/envs/data100/lib/python3.7/site-packages/scipy/stats/_stats_mstats_common.py:140: RuntimeWarn
  t = r * np.sqrt(df / ((1.0 - r + TINY)*(1.0 + r + TINY)))
/srv/conda/envs/data100/lib/python3.7/site-packages/scipy/stats/_stats_mstats_common.py:142: RuntimeWarn
  sterrest = np.sqrt((1 - r**2) * ssym / ssxm / df)
```

Out[89]:
```
description
No restroom facility within 200 feet of mobile food facility                0.000000
Mobile food facility with unapproved operating conditions                   0.883533
Mobile food facility stored in unapproved location                          0.889342
Mobile food facility not operating with an approved commissary              0.902196
Noncompliance with Gulf Coast oyster regulation                             0.912143
                                                                                …
Improper food storage                                                       0.997622
Inadequate food safety knowledge or lack of certified food safety manager   0.997635
Unclean hands or improper use of gloves                                     0.997755
Unclean or degraded floors walls or ceilings                                0.998566
Inadequate ventilation or lighting                                          0.998707
Length: 65, dtype: float64
```

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 [90]: zip_scores_df = (vio.merge(ins2vio, on="vid")
                         .merge(ins[ins["score"] > 0]
                                [["iid", "score", "timestamp", "bid"]], on="iid")
                         .merge(bus[["bid", "postal5"]], on="bid")
                         .sort_values(by="timestamp", ascending=True)
                          [["score", "postal5"]]
                         .groupby("postal5")
                         .agg(lambda s: s.mean())
                         .reset_index())

         f, axes = plt.subplots(1, 2, figsize=(15,7))
         (sns.barplot(x="postal5", y="score", data=zip_scores_df, ax=axes[0])
             .set(xlabel="ZIP code", title="Mean score for each ZIP code"))
         (sns.violinplot(y=zip_scores_df["score"], ax=axes[1])
             .set(title="Distribution of mean scores", ylim=(0, 100)))

         for ax in f.axes:
             matplotlib.pyplot.sca(ax)
             plt.xticks(rotation=60)

         # I tried to look for some correlation between the ZIP code
         # of the restaurants being investigated, and the subsequent
         # scores that they received. I didn't find anything super
         # interesting, since the graph does appear to show that the
```

```
# distribution of means is tightly distributed around the
# mid-80s in a unimodal distribution, skewed right (towards
# higher scores) slightly. Ideally, we would test this
# hypothesis against the distribution of all known health
# scores, but since that dataset is not available, we can't
# go any further.
```