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 [17]: display(bus.head())
         display(ins.head())
         display(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
```

```
                                      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 the bus data, there is a phone number of -9999. This is clearly nad data, as that is not a valid phone number. This puts into question the validity of the rest of the data in that row as well.

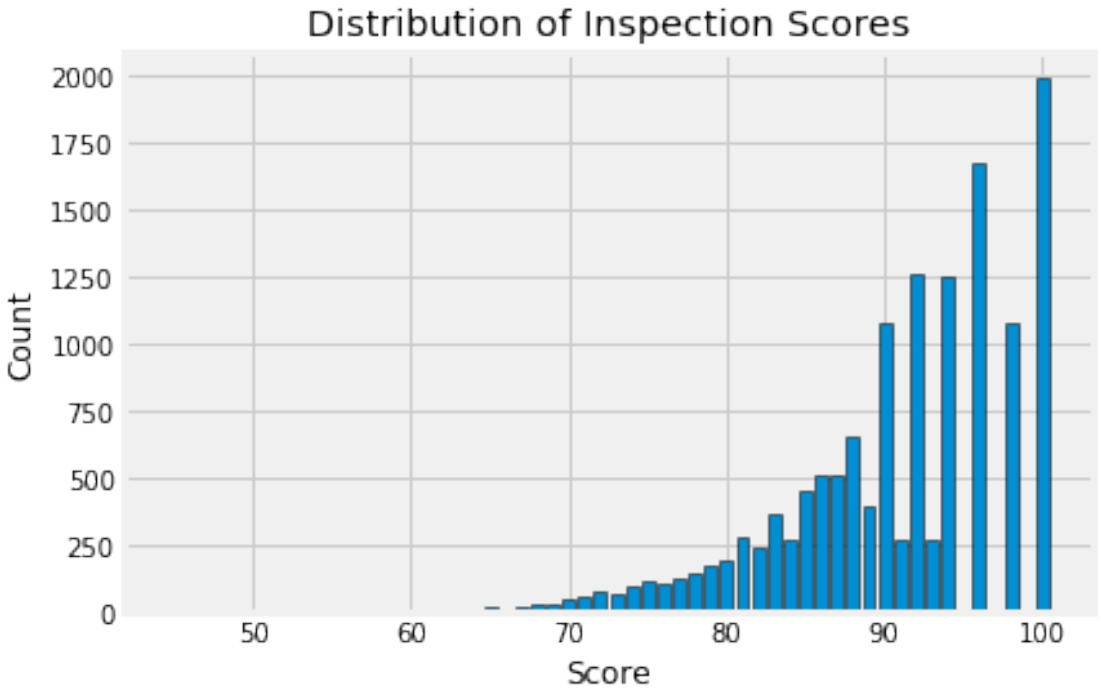
In the cell below, write the name of the restaurant with the lowest inspection scores ever. You can also head to yelp.com and look up the reviews page for this restaurant. Feel free to add anything interesting you want to share.

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:

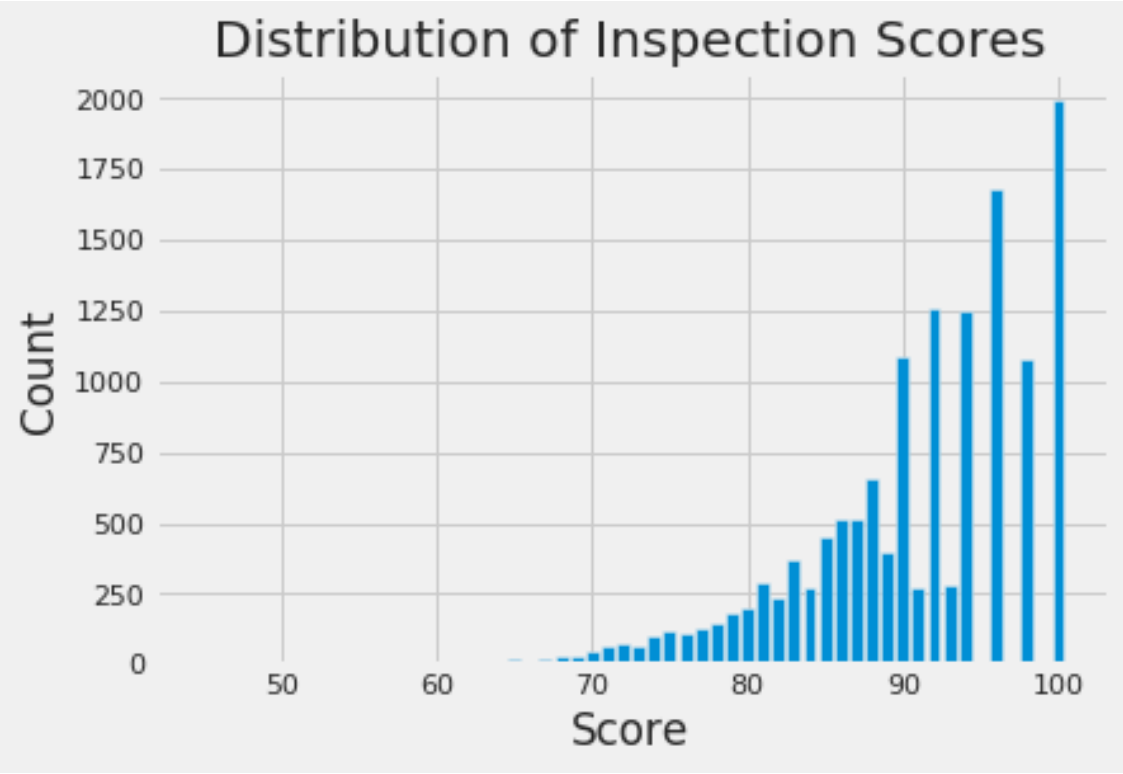
```
plt.bar
plt.xlabel
plt.ylabel
plt.title
```

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 [ ]:

In [79]:
```
score_counts = ins['score'].value_counts()
plt.bar(score_counts.keys(), score_counts)
plt.xlabel('Score')
plt.ylabel('Count')
plt.title('Distribution of Inspection Scores')
```

Out[79]: Text(0.5, 1.0, 'Distribution of Inspection Scores')

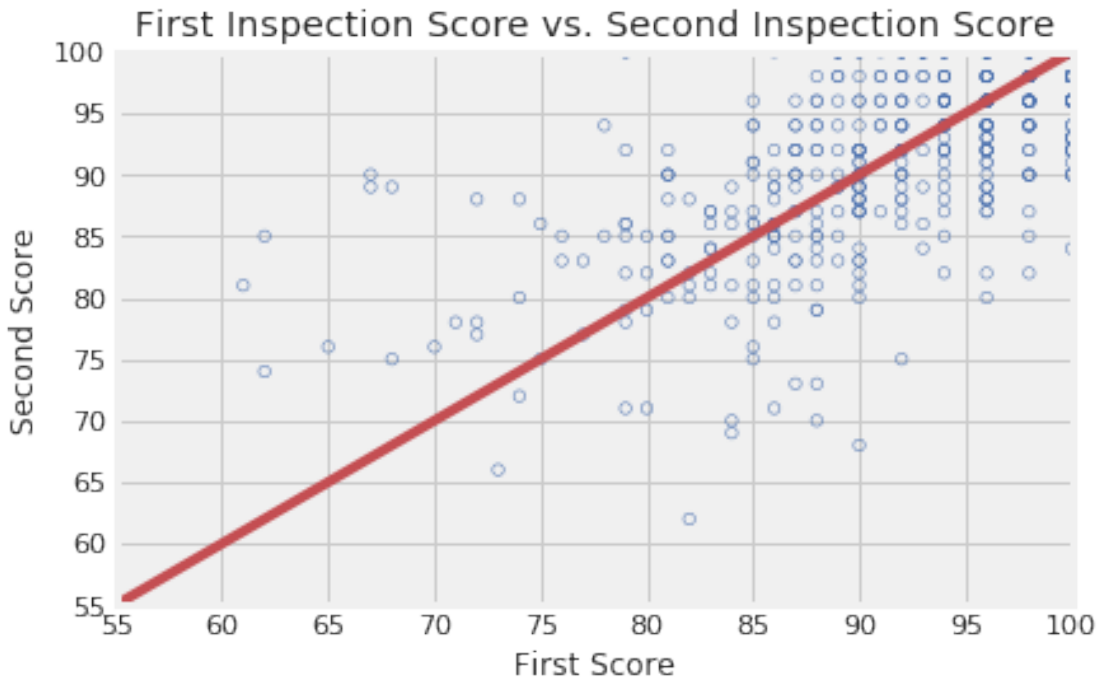


0.1.1 Question 6b

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 distribution is unimodal, not symmetric, and is left skewed. There are gaps in the upper values, from mid to high 90s, and the bar around 98 is much shorter than those around it. The gaps would imply that points were taken of the scores in increments greter than 1, like 2, 5, etc.

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:

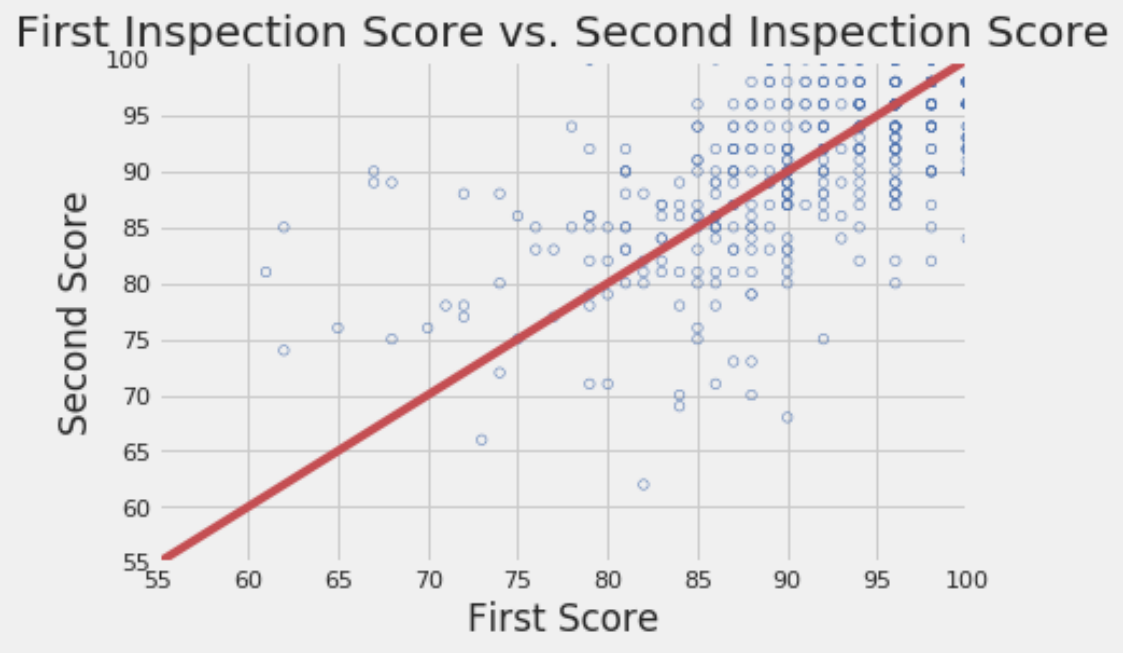
`plt.scatter` plots a set of points. Use `facecolors='none'` and `edgecolors=b` to make circle markers with blue borders.

`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.

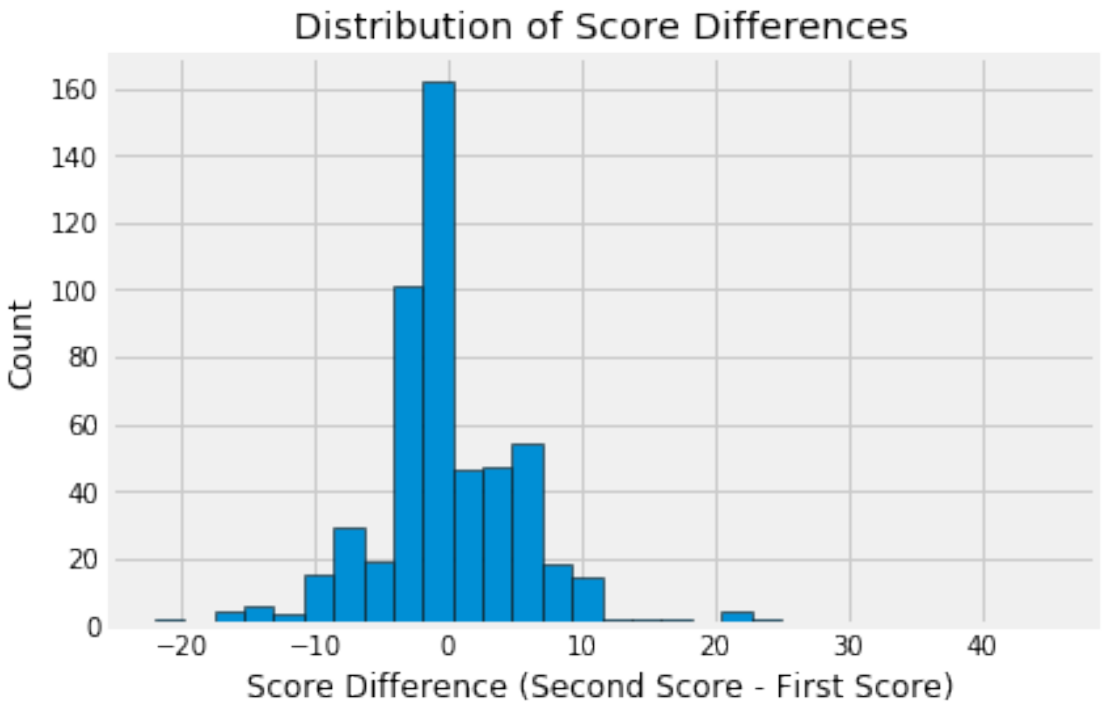
```
In [87]: first, second = zip(*scores_pairs_by_business['score_pair'])
         plt.scatter(first, second, s = 20, facecolors = 'none', edgecolors = 'b')
         plt.plot([55,100],[55,100],'r-')
         plt.xlabel('First Score')
         plt.ylabel('Second Score')
         plt.axis([55,100,55,100])
         plt.title("First Inspection Score vs. Second Inspection Score");
```


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:

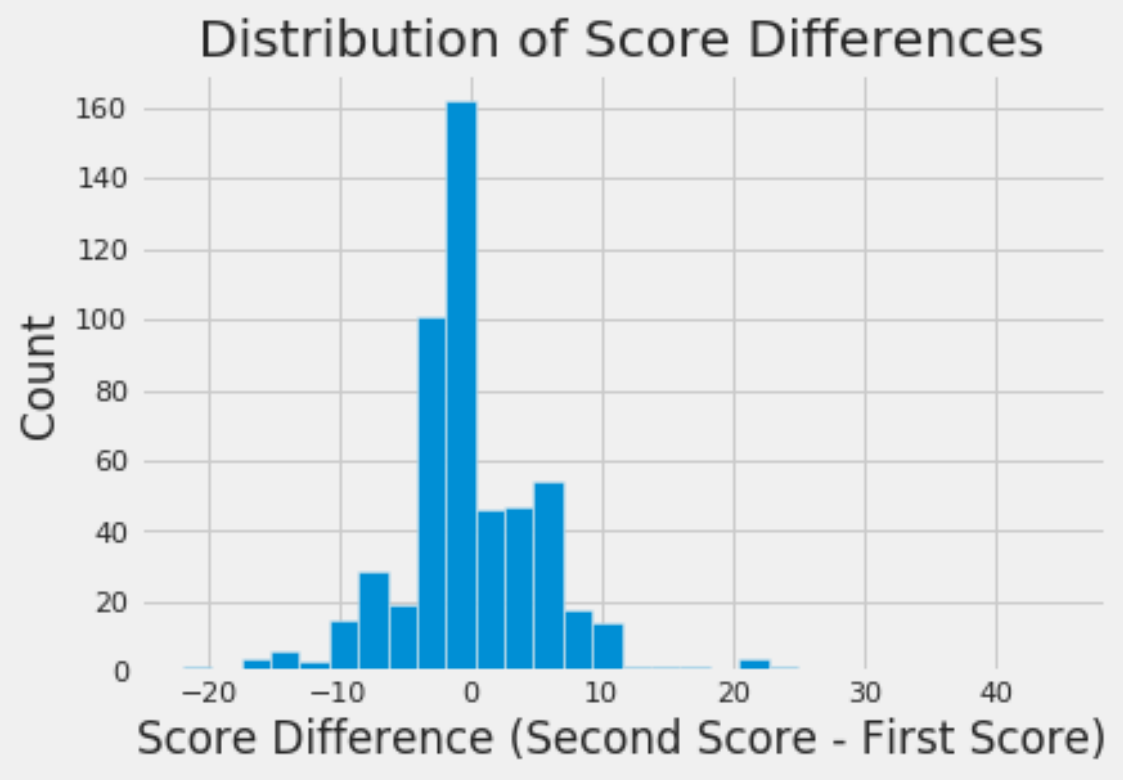


Hint: Use `second_score` and `first_score` created in the scatter plot code above.

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 [88]: diff = np.array(second) - np.array(first)
         plt.hist(diff, bins = 30)
         plt.title("Distribution of Score Differences")
         plt.xlabel("Score Difference (Second Score - First Score)")
         plt.ylabel("Count");
```


Distribution of Score Differences
Count
160
140
120
100
80
60
40
20
0
−20
−10
0
10
20
30
40
Score Difference (Second Score - First Score)

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?

If scores tend to improve from first to second, the points should be above the line of best fit in scatterplot of 7c. However, there this is not the case. The majority of the points are clearly not over the line, its split more evenly. This suggests that the average restaurant will not improve its inspection score from the first to second inspection.

0.1.4 Question 7f

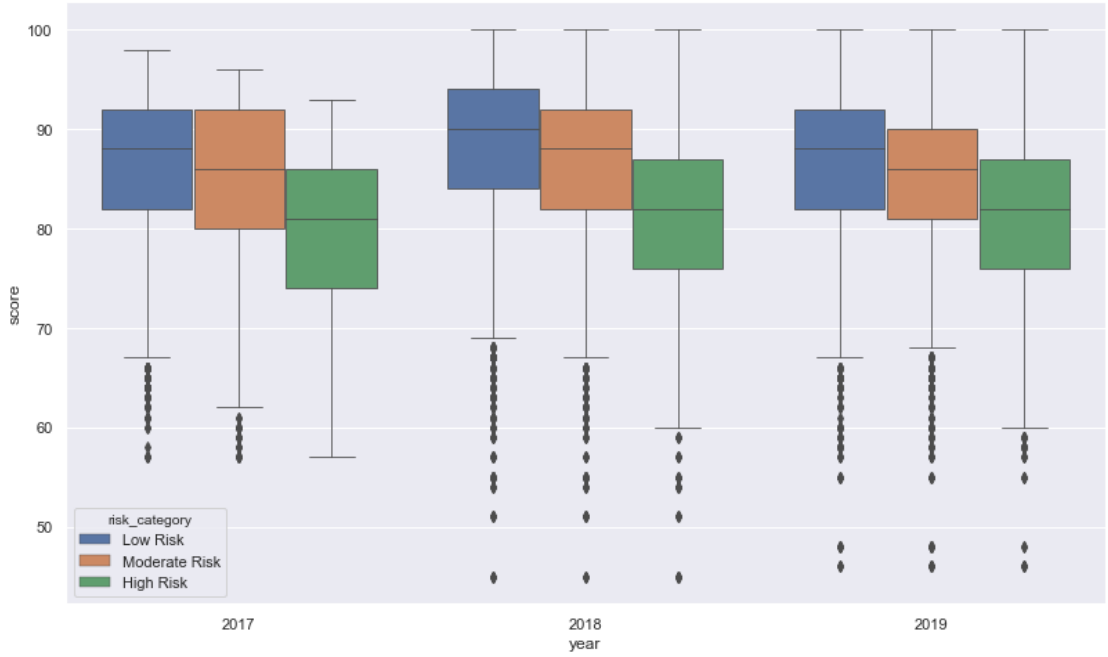
If a restaurant's score improves from the first to the second inspection, how would this be reflected in the histogram of the difference in the scores that you made in question 7d? What do you oberve from the plot? Are your observations consistent with your expectations? Explain your observations in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

If scores improve from first to second, the histogram should favor positive values. However, the distribution is centered around 0. This points to the average restaurant not improving from the first to second inspection, just like the scatterplot, which is consistent with my expectations. The spread is relatively large, from around -20 to 20, but those values are few, with the majority of the data within -10 t0 10. The distribution has long tails due to these values, which also increase the deviations from the peak.

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!



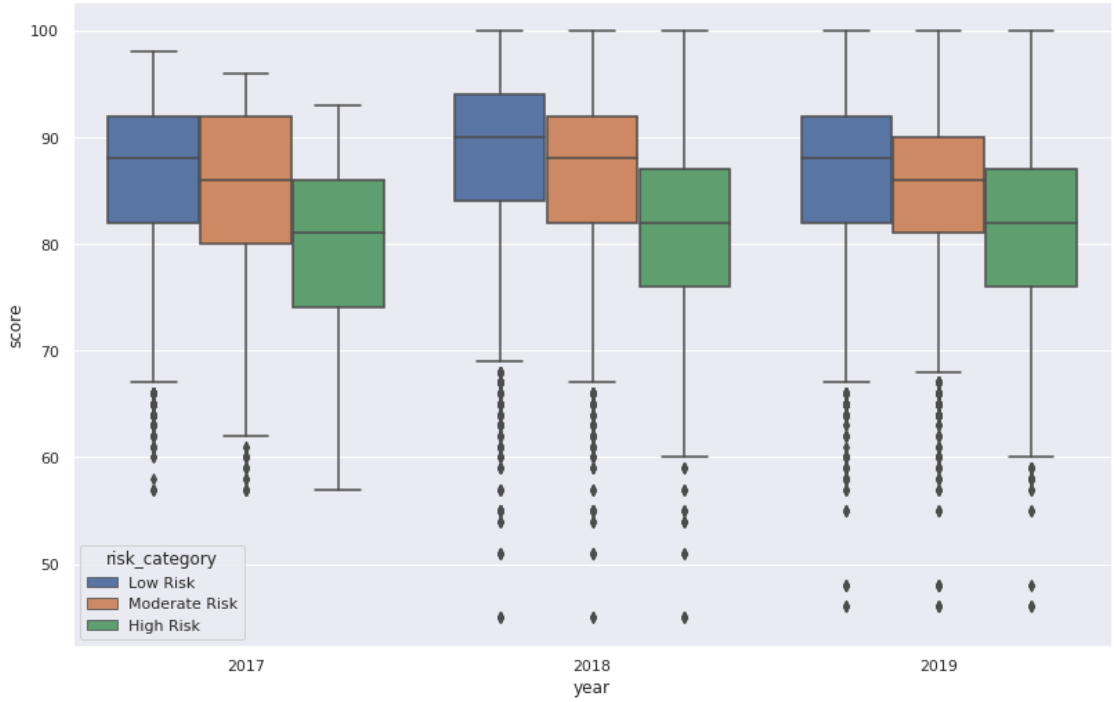
Hint: Use `sns.boxplot()`. Try taking a look at the first several parameters. The documentation is linked here!

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

```
In [89]: # Do not modify this line
         sns.set()

         ins2vio_vio = pd.merge(vio, ins2vio, how = 'left')
         risk_table = ins2vio_vio.merge(ins, on=['iid'])[['year', 'risk_category','score']]
         drop_2016 = risk_table[ risk_table['year'] == 2016].index
         risk_table.drop(drop_2016, inplace=True)
         risk_table
```

```
plt.figure(figsize = [12, 8])
sns.boxplot(x = 'year', y = 'score', hue = 'risk_category', data = risk_table, hue_order = ['L
```

Out[89]: <matplotlib.axes._subplots.AxesSubplot at 0x7f677d2d25b0>



---

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 [ ]:

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.

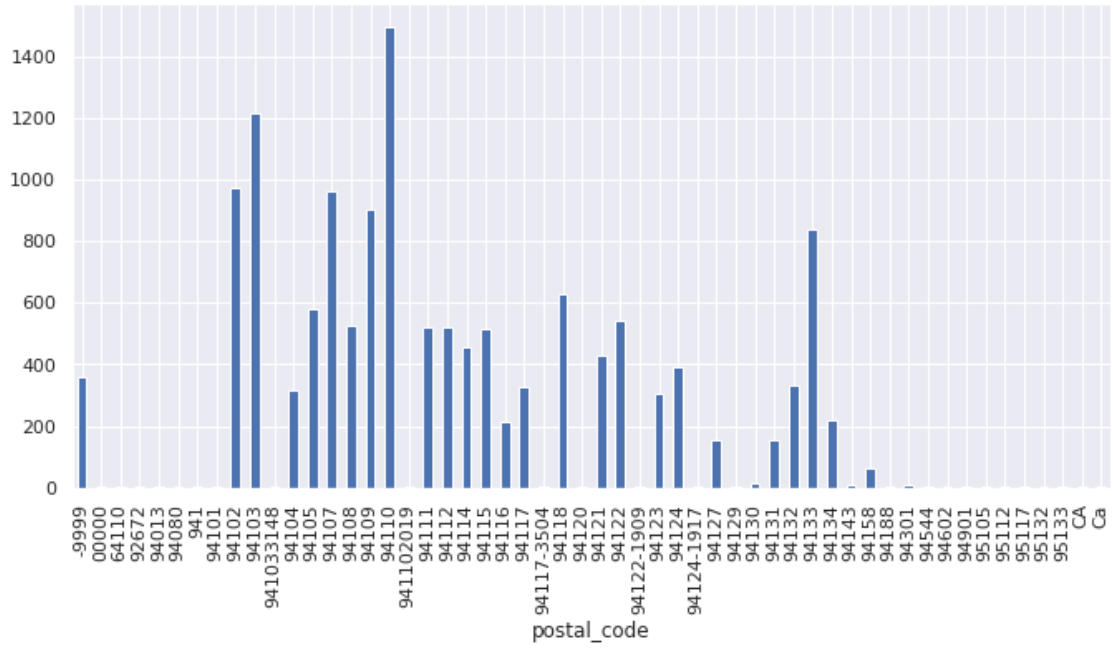
We will lean towards being generous with the grading. We might also either discuss in discussion or post on Piazza some examplar analysis you have done (with your permission)!

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).

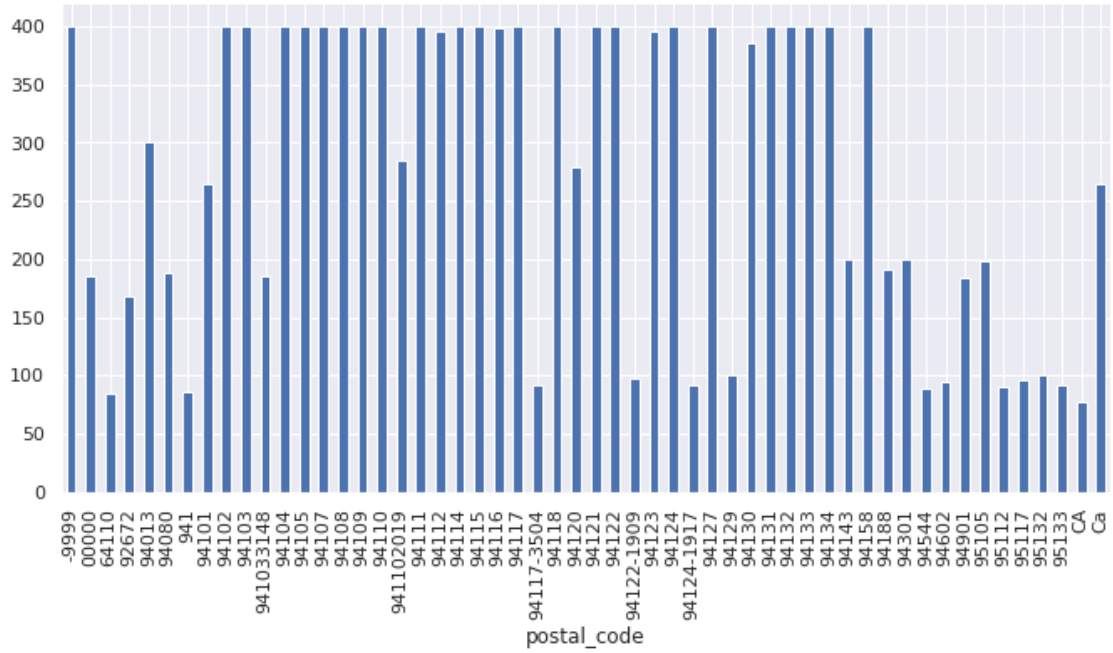
In [92]:
```
# YOUR DATA PROCESSING AND PLOTTING HERE
bus_ins = pd.merge(ins, bus, how = 'left', on = 'bid')

highest_bus_ins_len = bus_ins.pivot_table(index = 'postal_code',
                    columns = 'year',
                    values = 'score',
                    aggfunc = len,
                    fill_value = 0)

highest_bus_ins_len['total'] = highest_bus_ins_len.sum(axis = 1)
highest_bus_ins_len['total'].plot(kind = 'bar', figsize = [10, 5])



# YOUR EXPLANATION HERE (in a comment)
# I wanted to investigate the relationship between postal codes and scores.
# To do this, I merged the ins and bus tables on the bid column, then made three different piv
# Each pivot table is identical except for the aggfunc.
# The first table used len, so it shows the number of restaurants scored in a given postal cod
# This table is intersting, as some postal codes had 0 total inspections, while another had ov
# The second table used max, which takes the max score from each year in each postal code.
# This graph was intersting in that some postal codes had totals under 100, while the max is 4
# The last graph used min, which does the same as the max table but takes the min score instea
# This graph had some interesting results, like a postal code whose total min scores from the
```

Out[92]: <matplotlib.axes._subplots.AxesSubplot at 0x7f6765654f10>



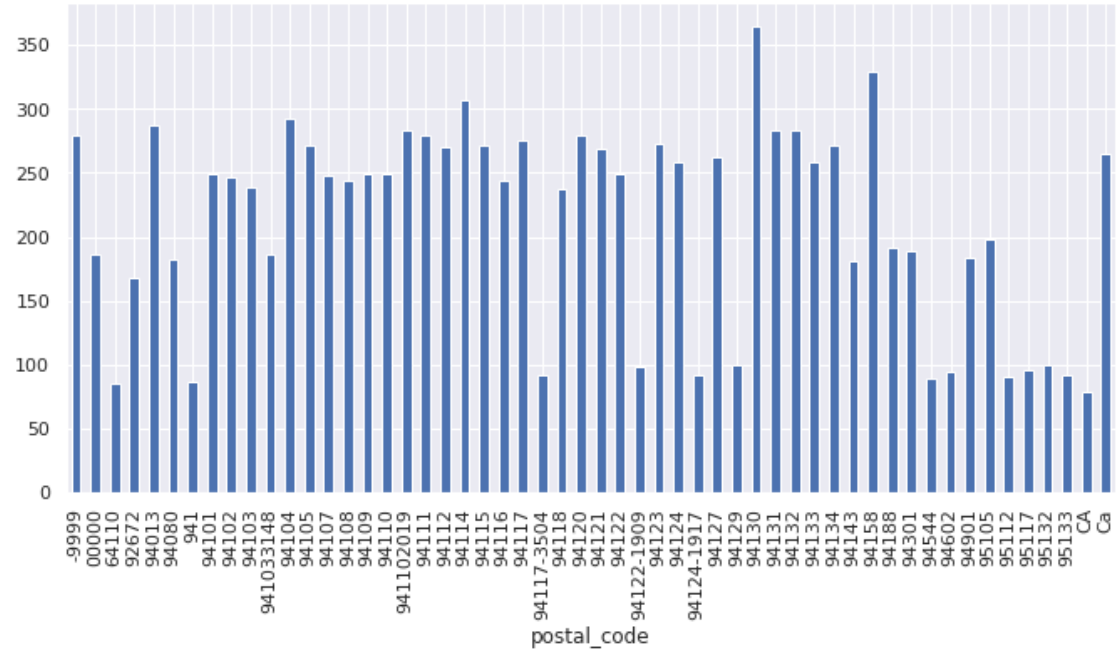
In [93]:
```
highest_bus_ins_max = bus_ins.pivot_table(index = 'postal_code',
                      columns = 'year',
                      values = 'score',
                      aggfunc = max,
                      fill_value = 0)
highest_bus_ins_max['total'] = highest_bus_ins_max.sum(axis = 1)
highest_bus_ins_max['total'].plot(kind = 'bar', figsize = [10, 5])
```

Out[93]: <matplotlib.axes._subplots.AxesSubplot at 0x7f6765bf2df0>



In [94]:
```
highest_bus_ins_min = bus_ins.pivot_table(index = 'postal_code',
                          columns = 'year',
                          values = 'score',
                          aggfunc = min,
                          fill_value = 0)

highest_bus_ins_min['total'] = highest_bus_ins_min.sum(axis = 1)
highest_bus_ins_min['total'].plot(kind = 'bar', figsize = [10, 5])
```

Out[94]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6765bd7310>
```


350
300
250
200
150
100
50
0
-9999
00000
64110
92672
94013
94080
941
94101
94102
94103
941033148
94104
94105
94107
94108
94109
94110
941102019
94111
94112
94114
94115
94116
94117
94117-3504
94118
94120
94121
94122
94122-1909
94123
94124
94124-1917
94127
94129
94130
94131
94132
94133
94134
94143
94158
94188
94301
94544
94602
94901
95105
95112
95117
95132
95133
CA
Ca
postal_code