Notebook

September 24, 2019

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 dataframe bus, there are multiple columns that contains data used to describe the location of the business. There are addresses, city, state, postal code, latitude and logitude in the data frame, which is kind of redundant. In addition, there are missing values like the third data in the phone number column in the bus data frame.

0.0.1 Question 2b

With this information, you can address the question of granularity. Answer the questions below.

1. What does each record represent (e.g., a business, a restaurant, a location, etc.)?

2. What is the primary key?
3. What would you find by grouping by the following columns: `business_id`, `name`, `address` each individually?

Please write your answer in the markdown cell below. You may create new cells below your answer to run code, but **please never add cells between a question cell and the answer cell below it.**

1. Each record prepresent a business.
2. The primary key is the business id.
3. By grouping with business id, I would find all the records about a certain business in a specific location. By grouping with the name, I would find all the records about a certian business, but the result might contain businesses with the same name but in mutiple locations. This is especialy for chain stores like Starbucks Coffee. When grouping with business id, I would only get records of a specific Starbucks like the Starbucks on 1800 IRVING ST. But if I group with name, I would get the records of all the Starbucks in the data frame. By grouping with address, I would get records of the stores in the same location. This demands all records has specific locations, and stores needs to be as specific as possible, like stores on the same floor should also have different addresses.

0.1 3: Zip Codes

Next, let's explore some of the variables in the business table. We begin by examining the postal code.

0.1.1 Question 3a

Answer the following questions about the `postal code` column in the `bus` data frame?
1. Are ZIP codes quantitative or qualitative? If qualitative, is it ordinal or nominal? 1. What data type is used to represent a ZIP code?

Note: ZIP codes and postal codes are the same thing.

1. ZIP codes are qualitative and they are nomial. ZIP codes are just numbers used to categorize, label, and identify locations, doing calculations like addition on them has no specific meanings.
2. ZIP codes are represented with strings.

0.1.2 Question 3c : A Closer Look at Missing ZIP Codes

Let's look more closely at records with missing ZIP codes. Describe why some records have missing postal codes. Pay attention to their addresses. You will need to look at many entries, not just the first five.

Hint: The `isnull` method of a series returns a boolean series which is true only for entries in the original series that were missing.

Some records have addresses recorded as "OFF THE GRID" like MOBI MUNCH, INC, and some records may have other data missing like missing phone number, missing latitude and etc.

If we were doing very serious data analysis, we might indivdually look up every one of these strange records. Let's focus on just two of them: ZIP codes 94545 and 94602. Use a search engine to identify what cities these ZIP codes appear in. Try to explain why you think these two ZIP codes appear in your dataframe. For the one with ZIP code 94602, try searching for the business name and locate its real address.

94545 is a ZIP code in Alameda, and 94602 is a ZIP code in Oakland. Both Alameda and Oakland are close to San Francisco, so it is possible that when collecting data, these two places' businesses are also recorded in the data frame. For the one with ZIP code 94602, the name of the business is ORBIT ROOM, and the real address of it is 1900 Market St, San Francisco, CA 94102.

0.1.3 Question 5b

Next, let us examine the Series in the `ins` dataframe called `type`. From examining the first few rows of `ins`, we see that `type` takes string value, one of which is `'routine'`, presumably for a routine inspection. What other values does the inspection `type` take? How many occurrences of each value is in `ins`? What can we tell about these values? Can we use them for further analysis? If so, how?

The Series type also take 'complaint' as one of the values. In ins, there are only one 'complaint' and 14221 'routine's. I assume that the type of inspection depends on the reason why there's an inspection. For 'routine', this is just a routine inspection, and for 'complaint', the inspection occours because there's complaints so that an extra inspection is needed. We can use them in further analysis as this type can somehow indicates the rating or the quality of the business, as complaints only appears when there're problems in the business like bad hygiene situations.

Now that we have this handy `year` column, we can try to understand our data better.

What range of years is covered in this data set? Are there roughly the same number of inspections each year? Provide your answer in text only in the markdown cell below. If you would like show your reasoning with codes, make sure you put your code cells **below** the markdown answer cell.

The range of years is from 2015 to 2018. The number of inspections each year is not roughly the same. The number of inspections in each year is shown below, and we can see that the numbers are not roughly the same.

0.1.4 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.

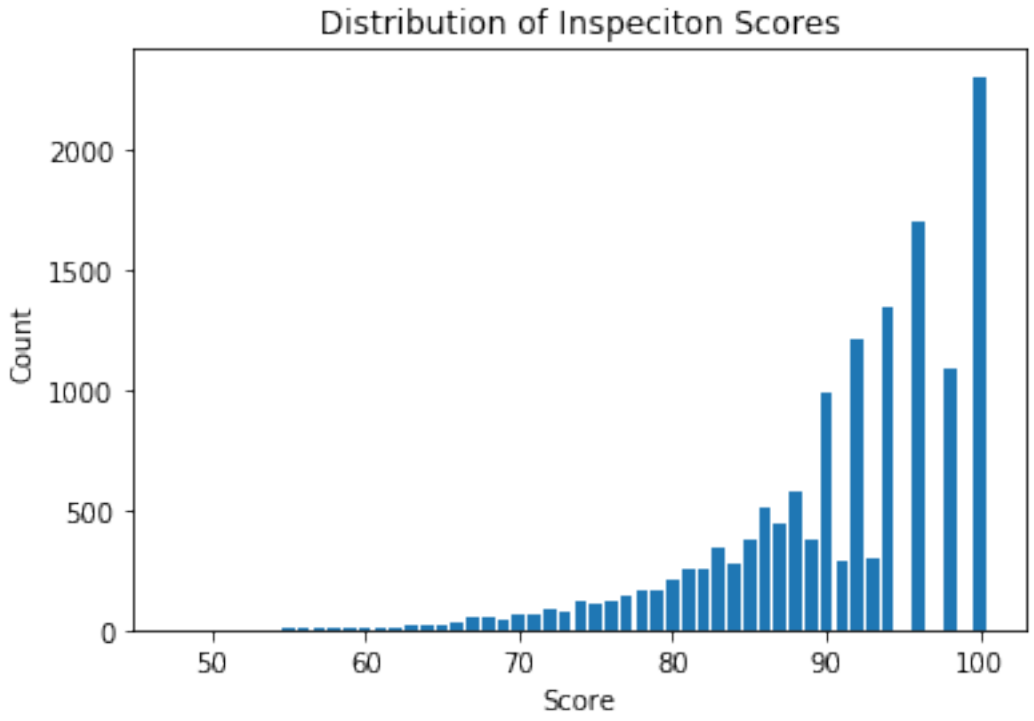
It should look like the image below. It does not need to look exactly the same (e.g., no grid), but make sure that all labels and axes are correct.

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 [47]: score_count = ins.groupby("score")["new_date"].count().to_frame().rename(columns={"new_date":
         plt.bar(score_count.index.tolist(), score_count["counts"].tolist())
         plt.xlabel("Score")
         plt.ylabel("Count")
         plt.title("Distribution of Inspeciton Scores")
```

```
Out[47]: Text(0.5, 1.0, 'Distribution of Inspeciton Scores')
```



0.1.5 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 anamolous values. Are there any unusual features of this distribution? What do your observations imply about the scores?

From the plot we can see that this distribution is asymmetric and it skews to the left. There are gaps in the score distributions, especially between 90 and 100. The maximum score is 100, and there are a lot of them. Most of the scores are in the 90-100 range. There are no unusual features of this distribution, and my observations about the scores implies that the scores are tend to be high in the 90-100 range, which means that most of the restaurants are in good condition and receives good scores in inspections.

Using this data frame, identify the restaurant with the lowest inspection scores ever. Head to yelp.com and look up the reviews page for this restaurant. Copy and paste anything interesting you want to share.

The restaurant with the lowest inspection scores ever is a restaurant called DA CAFE on 407 CLEMENT ST with the business_id 86647.

Something interesting: The reviews for this restaurant is acutally not that bad.

Now, create your scatter plot in the cell below. It does not need to look exactly the same (e.g., no grid) as the above sample, but make sure that all labels, axes and data itself are correct.

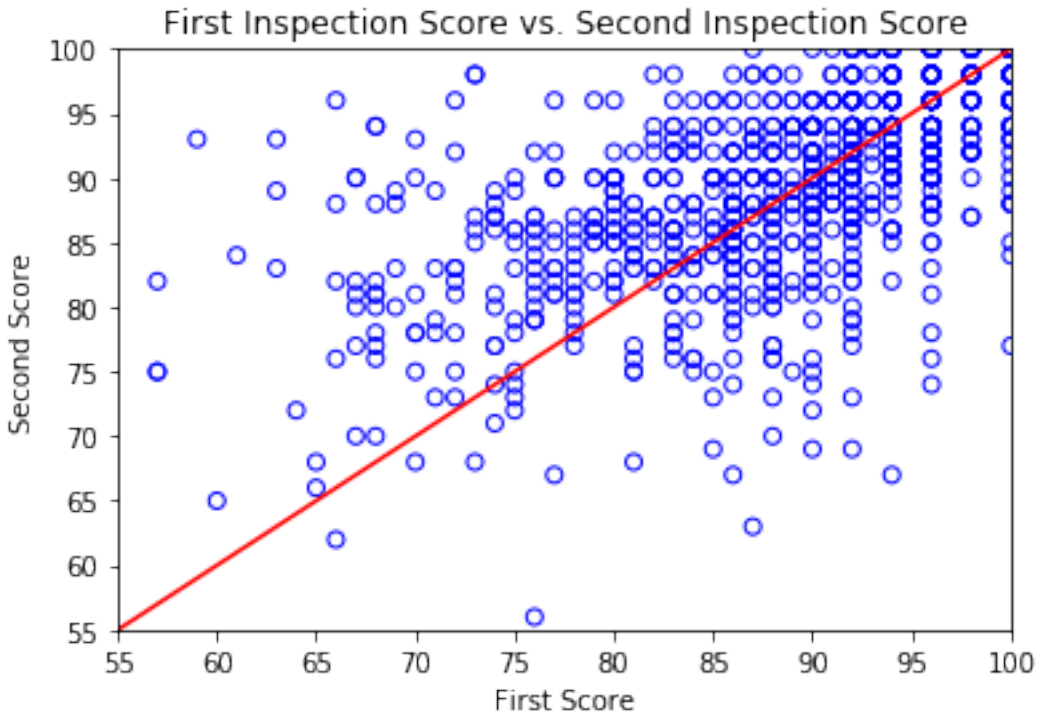
Key pieces of syntax you'll need: + `plt.scatter` plots a set of points. Use `facecolors='none'` to make circle markers. + `plt.plot` for the reference line. + `plt.xlabel`, `plt.ylabel`, `plt.axis`, and `plt.title`.

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.

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

```
In [57]: first_score = list(zip(*scores_pairs_by_business["score_pair"]))[0]
         second_score = list(zip(*scores_pairs_by_business["score_pair"]))[1]
         plt.scatter(first_score, second_score, facecolors='none', edgecolors='b')
         plt.plot(np.arange(55, 100, 0.1), np.arange(55, 100, 0.1), color = "r")
         plt.xlim(55, 100)
         plt.ylim(55, 100)
         plt.xlabel("First Score")
         plt.ylabel("Second Score")
         plt.title("First Inspection Score vs. Second Inspection Score")

Out[57]: Text(0.5, 1.0, 'First Inspection Score vs. Second Inspection Score')
```



0.1.6 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:

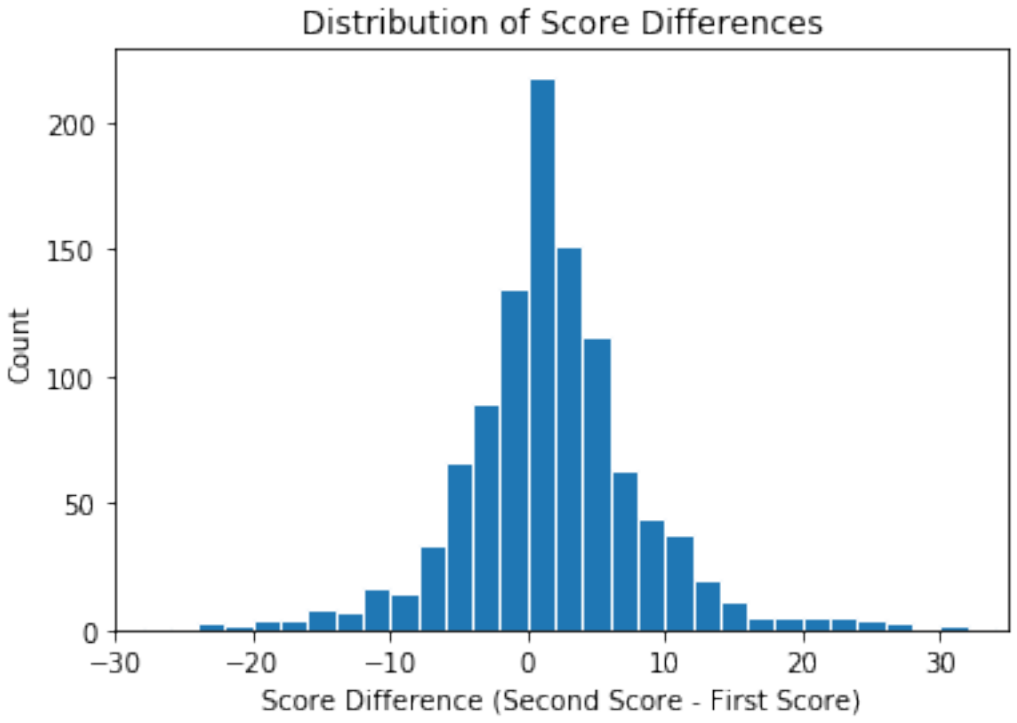
Hint: Use `second_score` and `first_score` created in the scatter plot code above.

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 [58]: first_score = np.array(first_score)
         second_score = np.array(second_score)
         diff_score = second_score - first_score
         plt.hist(diff_score, bins = np.arange(-30, 35, 2), edgecolor = "w")
         plt.xlim(-30, 35)
         plt.xlabel("Score Difference (Second Score - First Score)")
         plt.ylabel("Count")
         plt.title("Distribution of Score Differences")
```

```
Out[58]: Text(0.5, 1.0, 'Distribution of Score Differences')
```



0.1.7 Question 7e

If a restaurant's score improves 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 see?

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 see?

If a restaurant's score improves from the first to the second inspection, then in question 7c the scattered points would all be above the red line.

If a restaurant's score improves from the first to the second inspection, then in question 7d the histogram would only have bins on the right of 0. In other words, there will be no negative x values.