Rebecca Lewis
October 6, 2019
DSC 540
Mid-Term Project

The midterm project uses a dataset retrieved from Kaggle over a million beer reviews. [1] For time saving measures, I used a subset of this data to preform the assignment. The first step was to review the fields and create some meaningful descriptors for the headers since a separate data dictionary was not available. I worked through each objective for the assignment separately and then reorganized the code to make it more efficient. I combined the date formatting, key assignment, and fuzzy matching in one function to clean the data since all three tasks needed to iterate through the records. To implement fuzzy matching, I grouped the beers by the beer_style into one of the following categories: Lager, Ale, IPA, Stout, Bock, Pilsner, Other. With each transformation performed, I created a new element in the list to store the information with a column heading with the prefix 'derived'. Once the data was cleaned, I was able to test for missing values and duplicates. Based on these tests, the data was relatively clean.

I encountered the following challenges and worked through them:

1. Review Time: The review time was in relative time format. By reviewing other's work with the dataset in Kaggle, I discovered the number represented seconds and could be converted using the datetime library. [2]
2. My first attempt at finding a unique key was by using the brewery id, beer id and reviewer. However, this produced duplicates because a reviewer reviewed the same beer multiple times. To create a better key, I appended the review time integer to the key and stored it as a new element in the list.
3. When reviewing the distribution of data in the dataset, I experienced some challenges with the conditional statements used in the textbook counting data as the wrong datatype. My float numbers were not being recognized by isdigit, some of the text strings contained forward slashes so they were getting categorized as dates, and basic text was being categorized as unknown. To correct these issues, I created a custom is_number() function that attempted to convert the value to a float. If it was successful, the function returned true, otherwise it returned false. I created a similar function to test for datetime. To prevent strings from being flagged as unknown, I removed the "elif" criteria and just used "else" so if none of the other options were true, then it would count it as text.

Overall, I found working with lists and dictionaries to perform cleanup and exploratory data analysis challenging. I would most likely place the data in a dataframe to perform this type of work in the real world. With that being said, I do appreciate the practice in expanding my comfort zone. One thing that I couldn't figure out was how to find with this list/dictionary structure was the min and max value of each numerical data point. I like to check the numerical statistics for each field to determine outliers.

[1] Beer Reviews. Retrieved October 6, 2019 from https://kaggle.com/rdoume/beerreviews
[2] Recommending Beers. Retrieved October 6, 2019 from https://kaggle.com/fabiancpl/recommending-beers