Data Wrangling Report

For all dataset, I first read in the full dataset then extracts the columns that are needed. I renamed the columns for easier understanding. All area/city name in each dataset are converted to title case for future merging purpose. All zip code are converted into string for future merging purpose as well. All datasets use .drop_duplicates() during the data wrangling process. I worked on three dataset separately until achieved the desired results before merging them into one.

In the Yelp dataset, I found records that are not in US. Initially, I thought they are invalid state names. It turns out those state names are ISO 3166-2 code but without country code attached to the front. I manually construct a list with valid state abbreviation to eliminate the ones that are not in US. I also found few cities that are outside of US later when I was manually filling in missing population value in the main dataset, which also were removed. There are few outside of US records found in zip code data, which are removed along with decommissioned zip codes. Since census data is from US government census, there's no data that is outside of US.

The state name in census dataset is full state name, for future merging purpose, I first created a state full name and abbreviation lookup table by merging the self-constructed dataset with distinct state names extracted from census state column, then merged back to census dataset using left-join on 'state'. I removed words from the census data, like "Borough", "Town" and "City", and leave the area name only to match as many records. After few tries, I realized if I gradually removed the words throughout merging processes, I would get the best merging result, meaning I would have more matches comparing to I removed the words all at once.

After examining the Yelp data, I constructed a list with categories that I think are food related, like "Food", "Bars", "Bakeries", "Bakeries", etc. I used the list to extract out food business related records. Since .isna() function can't find null string, I created a is_empty function to help find empty string in all string columns. There is one city value and many zip code values missing. I was able to fill the missing city. Next, I merged Yelp data with zip code data to find missing or incorrect state name. In addition, latitude and longitude columns are used to help fill in missing city and state names; correct city, state and zip code value; and verify records.

Many city names are unified before merging with census data. For example, any city name with Mt. or Mt are changed to Mount and N or N. are changed to North. There were about 1000 records with population value is null after merging. I manually go through the records and correct them until between 50-100 records left, which I was unable to find the population values. During the process, I cross-checked between census data and zip code data, along using a website that can convert latitude and longitude into actual address. In the case where zip code is missing, I only use the website. I changed the data mainly in two ways. One is using business_id where only single record or only few other ones with the same

correct value, like business_id '110iMPMPEEjFlf8HKVq84g' and '1jdE-PeiQHvL8165vebWrw'
both has incorrect city name that are supposed to be “Charlotte”; or the city name exists but in another state, like “Pittsburg” exists in California but with state is PA the city name is actually “Pittsburgh”. Besides city name, this method is also applied to change incorrect zip code and state name. The other way changes city name in the case where the city name is misspelled and it doesn’t exist in another state; or in the case where a zip code has multiple area names in the census dataset, thus, I changed the city name in Yelp data to match. Last, I removed the ones that I was unable to find population number from the result dataset.