ETL Project

Kevin M. Holbrook and Jonas Thorrud

Date: 5-7-2019

Question the data could be used to answer: Is the number of and location of Marijuana dispensaries in California dependent on the Average Income of the area?

Data sources:

1. From the California secretary of state: <https://aca5.accela.com/bcc/customization/bcc/cap/licenseSearch.aspx>. A complete listing of all Marijuana related businesses in the state.   
   Data: license types, Owner(s), contact information, Business entity type, business address.  
     
   Format: CSV  
     
   Transformation Needed:  
   a. The contact information is provided in a format that needed to be parsed out. The information is likely provided by the business owners application, but much of the information was likely not a required field. For example, some businesses may have a website vs others that do not list one. This meant that parsing the data is not a straight forward as simply splitting the columns at each space.  
   b. The address field needed to be parsed out into separate columns  
   c. Zip codes: Many came with an extra 4 digits at the end. These needed to be removed  
   d. Other typical transformations like dropping NaN, renaming columns, etc.
2. Latitude and Longitude  
   This data was acquired from google maps API using the data previously extracted from the secretary of state (step 1).

Format: Pandas DataFrame extracted by reverse geo location through Google maps API

1. Census.gov  
   This data was acquired from the US census. Specifically we targeted income (through tax returns) by Zip code.

Format: excel

Transformations:  
Due to getting copious amounts of errors reading the spreadsheet in pandas, this one had to be manipulated in excel. Mostly we needed to remove columns, update number formats and save as csv. Once this was completed, we re-imported into Pandas to calculate the per capita zip code incomes using pandas group-by functions.

Databases:

The final data frames were loaded into SqLite and Mongo. We decided to use SqLite ease of querying and Mongo because we wanted to learn it and because it’s fast!

1. SqLite

1. Mongo  
   There were a few issues with Mongo. Because we entered several sources it was difficult to figure out how to query the databases and join the tables. Although we joined within Pandas, this would be a step to overcome in the future as being alke to join is imperative to analyzing the data. The solution appears to be to pull the data into Pandas and then perform the join.

Results: