CS513: Theory and Practice of Data Cleaning – Final Project

Individual Submission: Jonathan Chang

NOTE:

Input dataset:

- farmersmarkets.csv - https://uofi.box.com/s/6j68z6yukfb3msrjniovo0hsk7pp1lsg

Output dataset(s):

- farmersmarkets_output.csv - https://uofi.box.com/s/dlv27sg8l9h6dv0ouqjdv8od7i9jjrf
- farmersMarket_location.csv - https://uofi.box.com/s/dx197g8g9k40bggtnvfc0a2i3e7888ro
- farmeresmarkets_payments.csv - https://uofi.box.com/s/c7f5zkccix6dg4dhlidklnqf74rz877b
- farmersmarkets_products.csv - https://uofi.box.com/s/c9rfr41vbq6jutofoj1ymhh6dso3jyml

1. Introduction and Overview

In my data cleaning project, I explore the US Farmers Market dataset from the USDA Website: https://www.ams.usda.gov/local-food-directories/farmersmarkets. As defined by Wikipedia, a farmers' market is "a physical retail marketplace intended to sell foods directly by farmers to consumers." The dataset is a directory listing of the various farmers markets in the United States, and includes information such as social media accounts, market location, accepted payments, and agricultural products sold.

1.1 Project Use Case

Given this dataset and my interest in the modernization of payment methods, I think an interesting use case to explore would be identifying the adoption of credit card usage. We could do this by either some SQL queries and in the end, by creating a map that portrays the acceptance of credit cards by state (maybe percentage of markets that accept credit cards).

1.2 Other Potential Use Cases (dataset "clean enough")

Without (or with very little) additional cleaning, these are just a sample of some of the possible use cases possible with our dataset.

- We could determine the most and least popular products that tend to be sold by farmers' markets by summing the existence of 'Y' for each product's column. We could also do this across certain states or zip codes.
- We could also determine the most popular type of payment options accepted by farmers markets in general - cash, credit, food stamps or vouchers, etc...

- We could explore competition within certain zip codes by looking at the density or count of farmers' markets in certain zip codes.

1.3 Unrealistic Use Cases (dataset will never be good enough)

- Any use cases involving dates past Season1 will simply not be able to be supported as really only Season1 is populated. So, for example, it would not be possible to compare the dates and times for which a farmers' market is opened across seasons.
- Detailed analysis of social media options for the farmers markets is also highly unlikely due to missingness. For instance, Youtube, Twitter, and Other Media columns have around 90% missing values. If some of these columns were better populated with links, then a web-scraping pipeline could potentially be developed to augment the current dataset.

1.3 Data Cleaning Goals

Given my use case, the goal of my data cleaning exercises is primarily to reduce errors and ensure consistency and reliability of the 'state' column. For instance, we initially observe 53 unique values which indicates that there may be several misspellings or some other type of error. I also want to make sure that the x and y columns which correspond to longitude and latitude, are in good shape

However to improve the overall fitness of the dataset and allow it to be usable for some of the other potential use cases, I will generally seek to improve the consistency, reliability, and accuracy of all the columns with the help of OpenRefine. As we have done earlier in the class in OpenRefine on the Airbnb dataset, we will generally seek to perform some of the following, similar operations:

- trim and collapse white spaces
- ensure proper typing (e.g. for numbers and dates)
- correct misspellings by facets and clustering
- ensure consistency of certain columns by specifying the case
- delete some of the irrelevant columns

2. Initial Assessment of the dataset

2.1 Structure and Contents

There are 8687 total observations and 59 columns in this dataset which are described below. The provided html report was generated via a python package called pandas_profiling, and allows us to observe some basic, preliminary statistics such number of rows and columns, cardinality, missing values, correlations, etc... as well as the overall schema of the dataset.

FMID - 7 digit integer that uniquely identifies each farmers' market
MarketName - a string containing the name of the farmers' market

Website, Facebook, Twitter, Youtube, OtherMedia - a string containing URL or other information that identifies the social media site
street, city, County, State, zip - strings that contain data corresponding to the column name that identifies the location of the farmers' market
Season1Date, Season1Time, Season2Date, Season2Time, Season3Date, Season3Time, Season4Date, Season4Time - date fields representing the start date and end date for the given farmers' market or the times in which the farmers' markets are opened
x, y - latitude and longitude coordinates
location - a string describing the location of the farmers' market
Credit, WIC, WICcash, SFMNP, SNAP - Y/N (boolean) character to indicate whether or not a given payment method is accepted
Organic, Bakedgoods, Cheese...PetFood, Tofu, WildHarvested (30 columns) - Y/N (boolean) column to indicate whether or not a given product is offered

There is a provided ER diagram in Section 4 that depicts how we chose to separate our dataset into various tables, the relationship between them, and the schema we created.

2.2 Quality Issues (narrative)

Using the groups above that describe the dataset contents, we describe some of the quality issues that exist in the dataset from a precursory glance. Many of these will be targeted for cleaning via OpenRefine.

For the social media columns (Website, Facebook, Twitter, Youtube, OtherMedia), most of the rows appear to be missing, and sometimes, in lieu of an URL, a string is provided. The string could be a Facebook username or Twitter handle, but the representation is not uniform. The location columns that together comprise an address may have some missing values and basically don't contain all 5 components of the address. There may also be leading/trailing white spaces that need to be trimmed, or case conversions that need to be performed, in order to standardize and clean the address data.

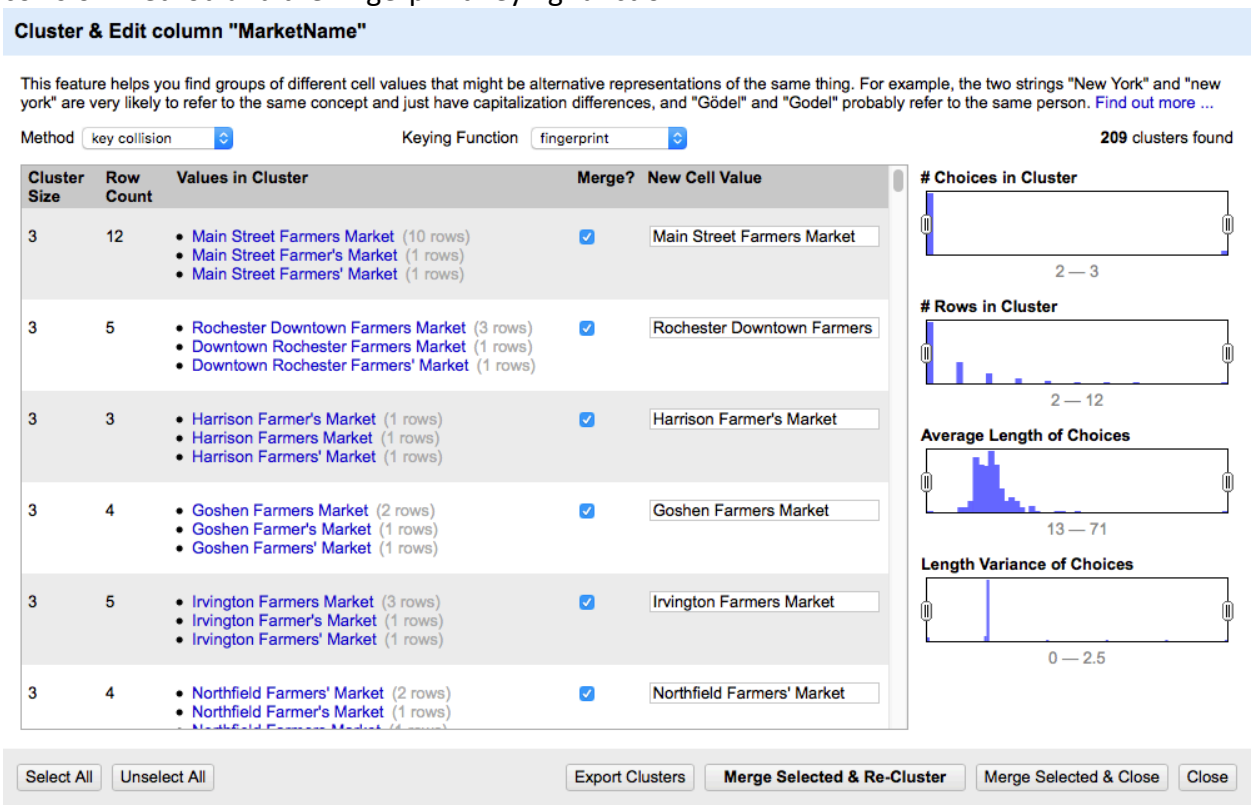
Next, for the dates and times, we see that only Season1 tends to be populated. The values are fairly inconsistent as well - some dates are represented using mm/dd/yyyy and some are represented using month name. I've also noticed some date ranges that don't contain the end date. The Season1Time column is also inconsistent. Also, the x and y columns could be better labeled as latitude and longitude, and even the Location column is somewhat poorly because it appears to be a description about the location.

Meanwhile, for the boolean columns that contain Y/N values, we also see '-' values which could probably be better represented by a null value. In another words, we want the column to be truly boolean with only 'Y' or 'N'.
Finally, for the updateTime column, we only receive year for some of the records, while others contain the full date time. Also, some of the records contain the month name as opposed to the number. Again, we will use OpenRefine to correct some of the data quality issues.

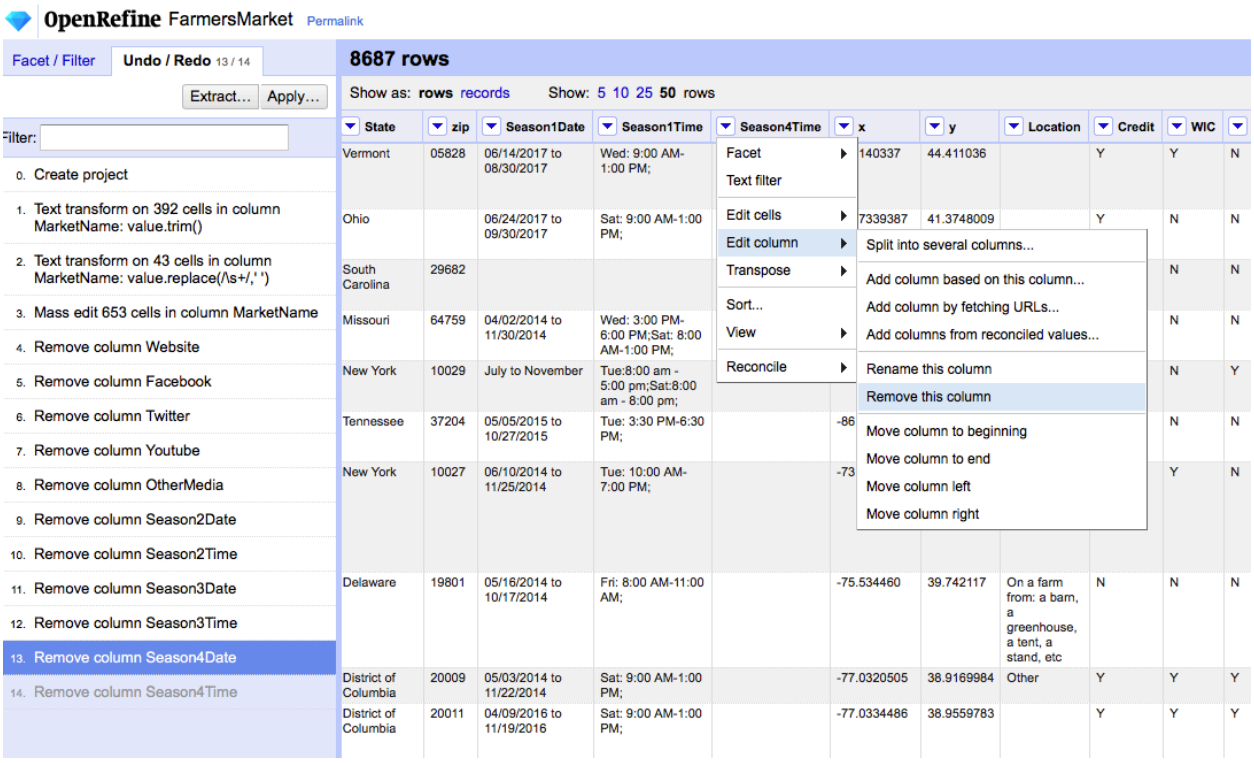
3. Data Cleaning methods and process

OpenRefine

Step 1. We begin with the MarketName column by first trimming the leading and trailing whitespace and then collapsing any consecutive whitespaces. Then we use a text facet and clustering in order to group similar MarketNames together. As seen below, we used the key collision method and the fingerprint keying function.

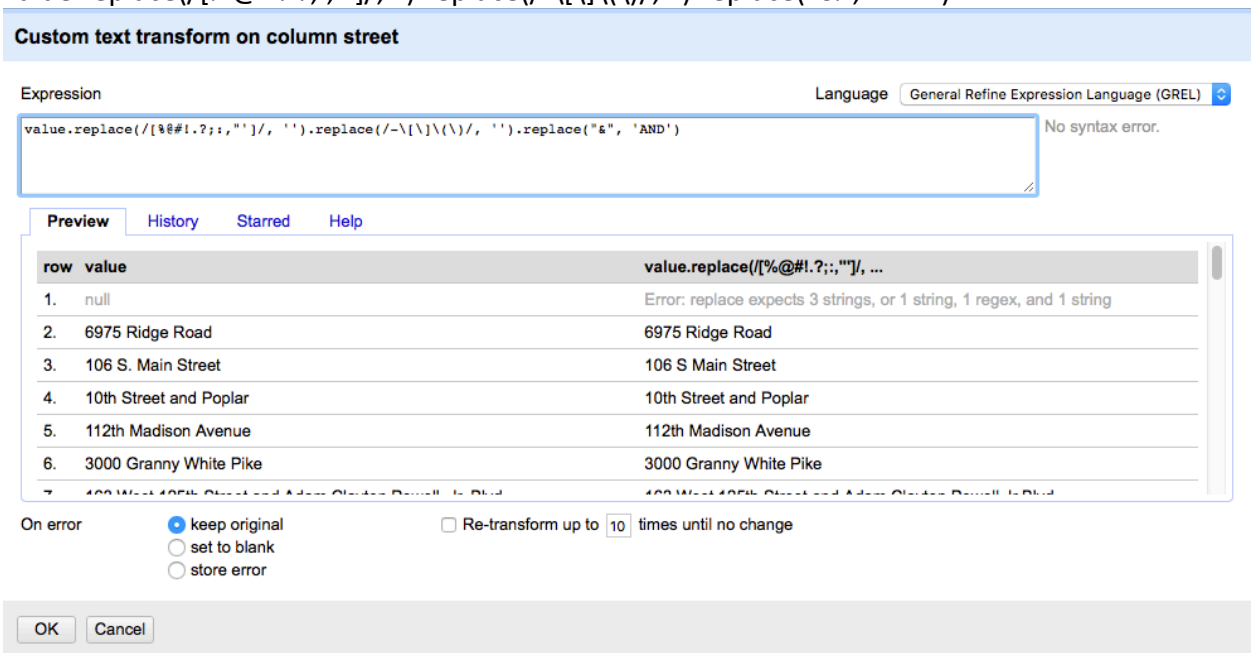


Step 2. Next, we remove some of the columns that are irrelevant to both our main use case and other potential use cases. We had decided that the social media data quality was very poor and so we delete the following columns: Website, Facebook, Twitter, Youtube, OtherMedia. We also remove the time and date columns for Season2 onwards because there is very little data for these.

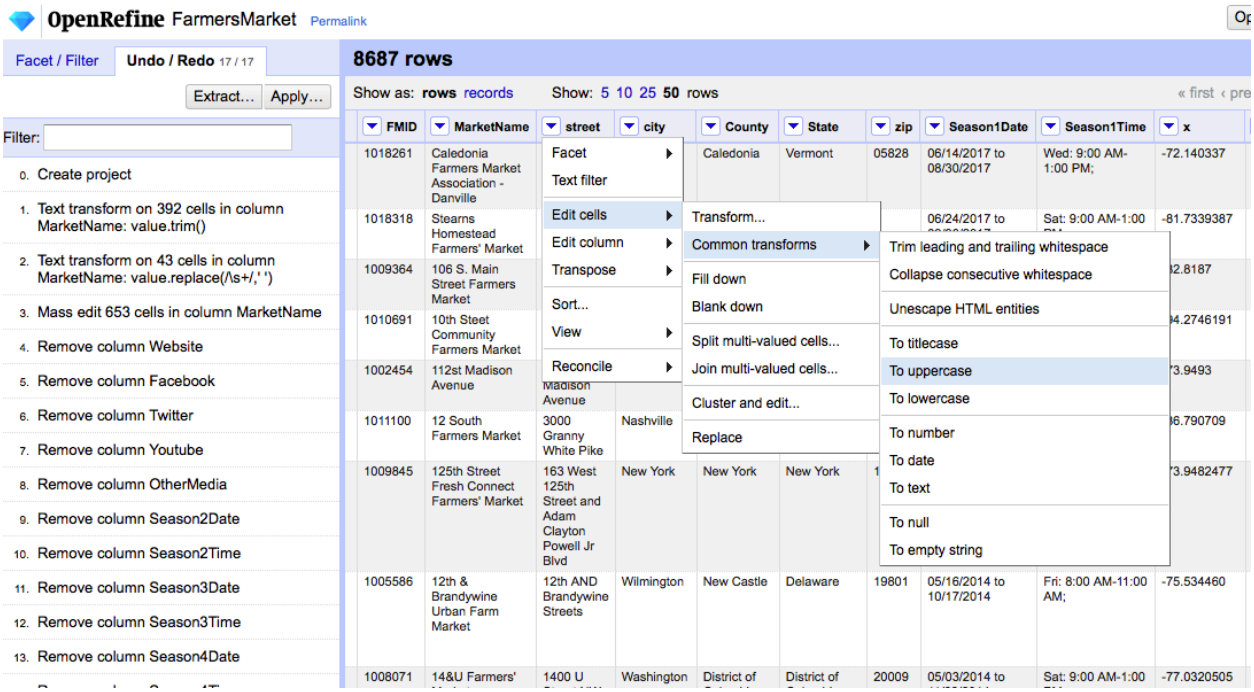


Step 3. Then, we focus on the location columns - street, city, County, State, and zip.

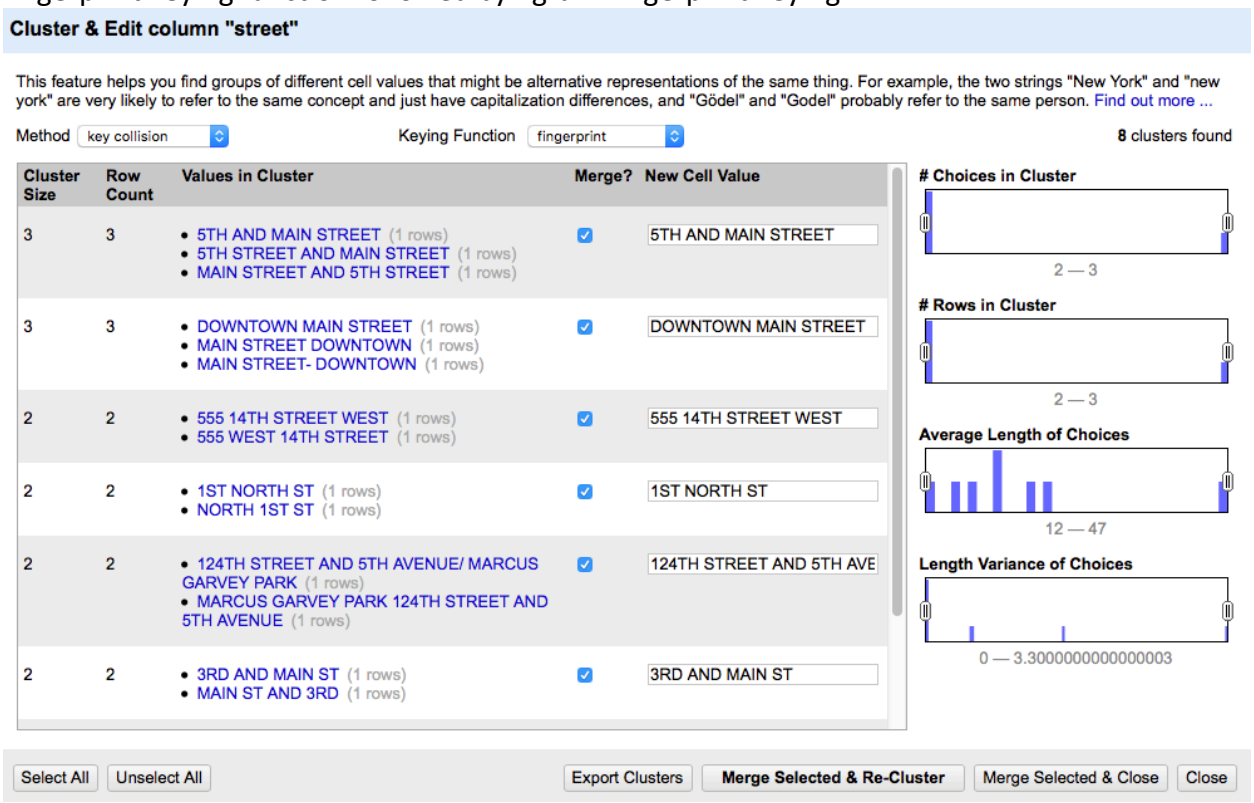
For street, we use the following GREL expression to remove any special characters and substitute the ampersand with 'AND':
value.replace(/[%@#!.?;:,"']/, '').replace(/-\[\]\(\)/, '').replace("&", 'AND')

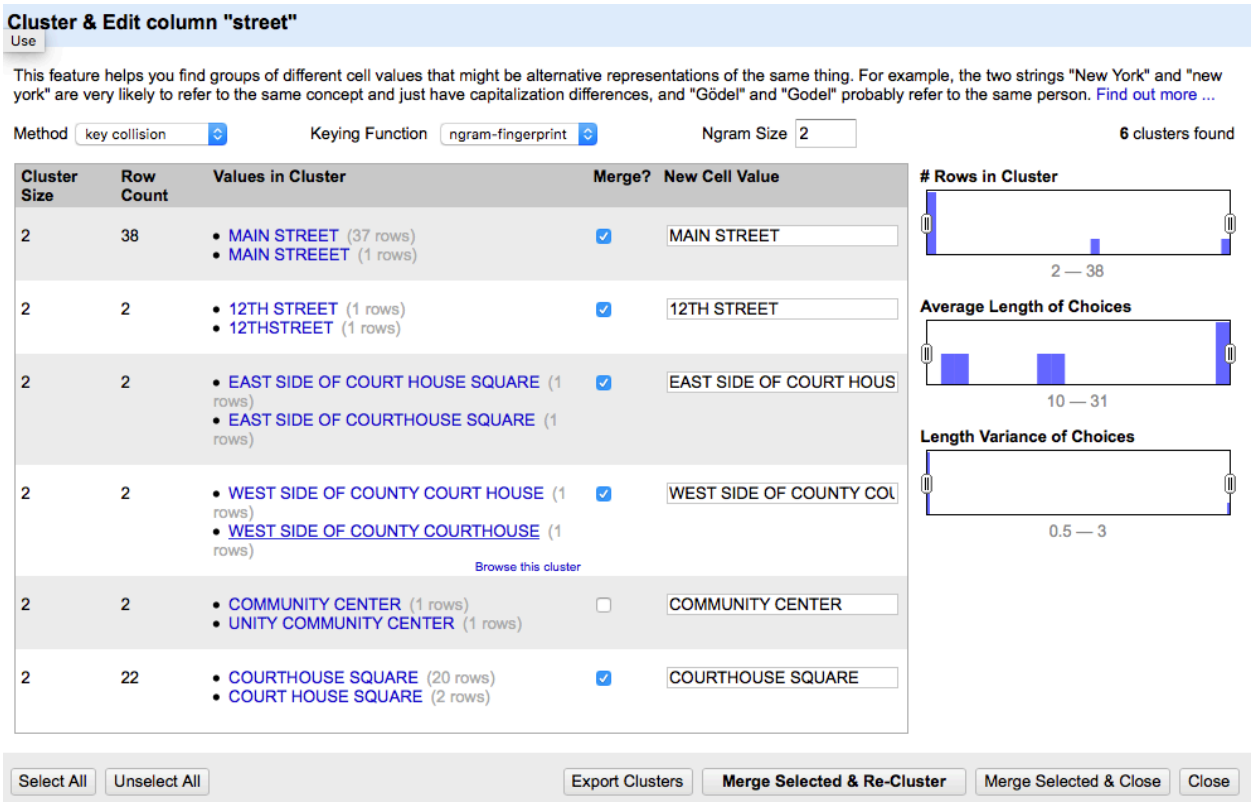


Then, we trim the leading and trailing whitespace and collapsed any consecutive whitespaces and then convert to uppercase.

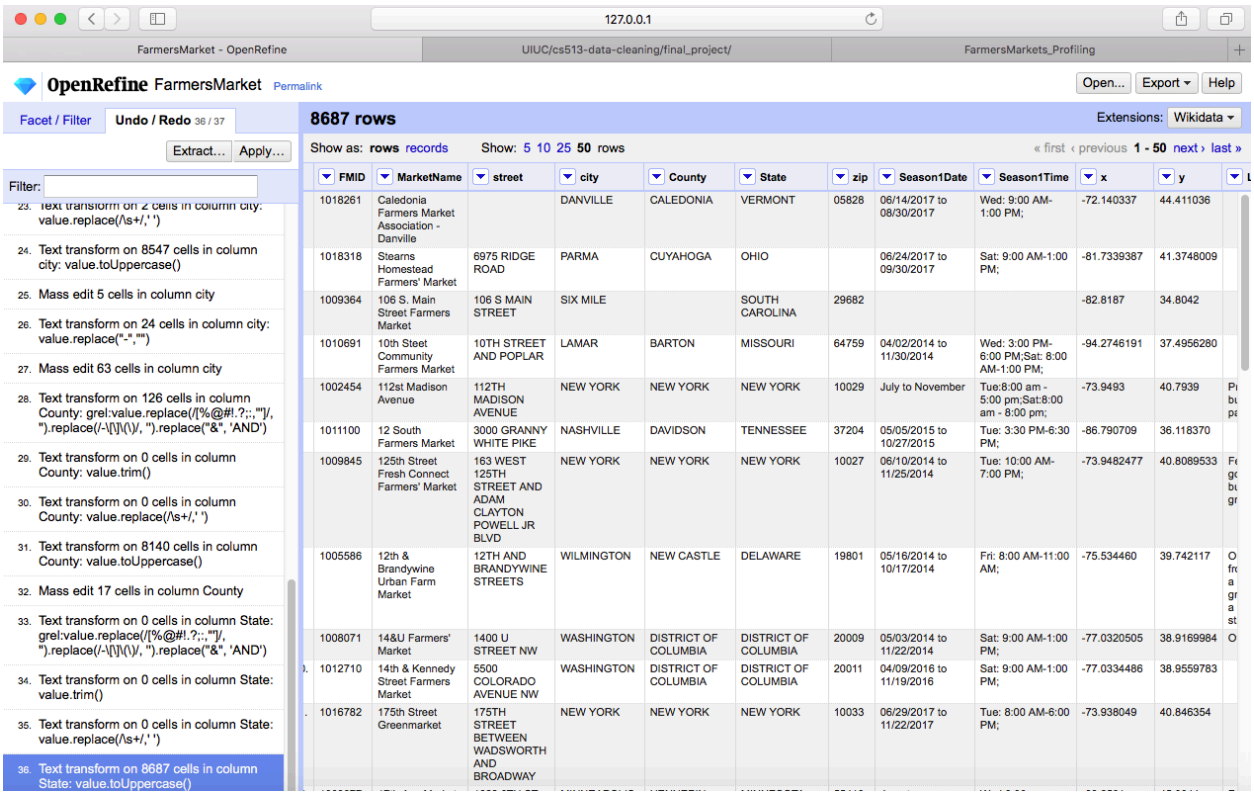


Then, we begin to merge any logical clusters together using the key collision method and fingerprint keying function followed by ngram-fingerprint keying.





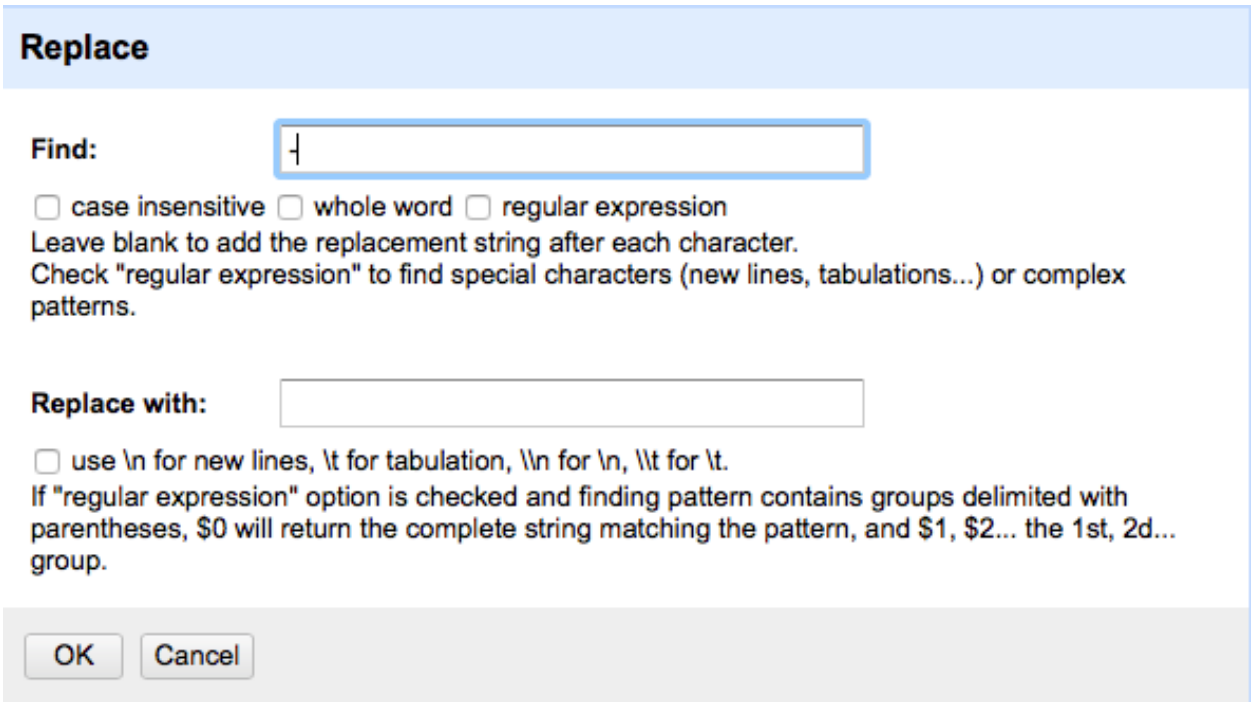
We go through this exact same process (remove special characters, trim and collapse whitespace, convert to uppercase, clustering) for the city, County, and State columns. After this process, we see that the address information is much cleaner and more consistent.



Step 4. Now we move to Season1Date and Season1Time. We decided to just remove these columns because they are not relevant to our current use case. (Of course, we could split Season1Date into 2 columns for a starting and ending date, but we would also have to figure out how we want to represent the rows where a month is given)

Step 5. Important to our analysis later, are the x and y columns, which we rename to latitude and longitude respectively, and then convert to numeric. We remove the Location column which is not helpful for our purposes, and is generally blank. Because, our analysis is dependent on the Credit column, we make an additional

Step 6. For some finishing touches, we remove the occurrence of "-" in the Organic column, so that missing values are just left blank.

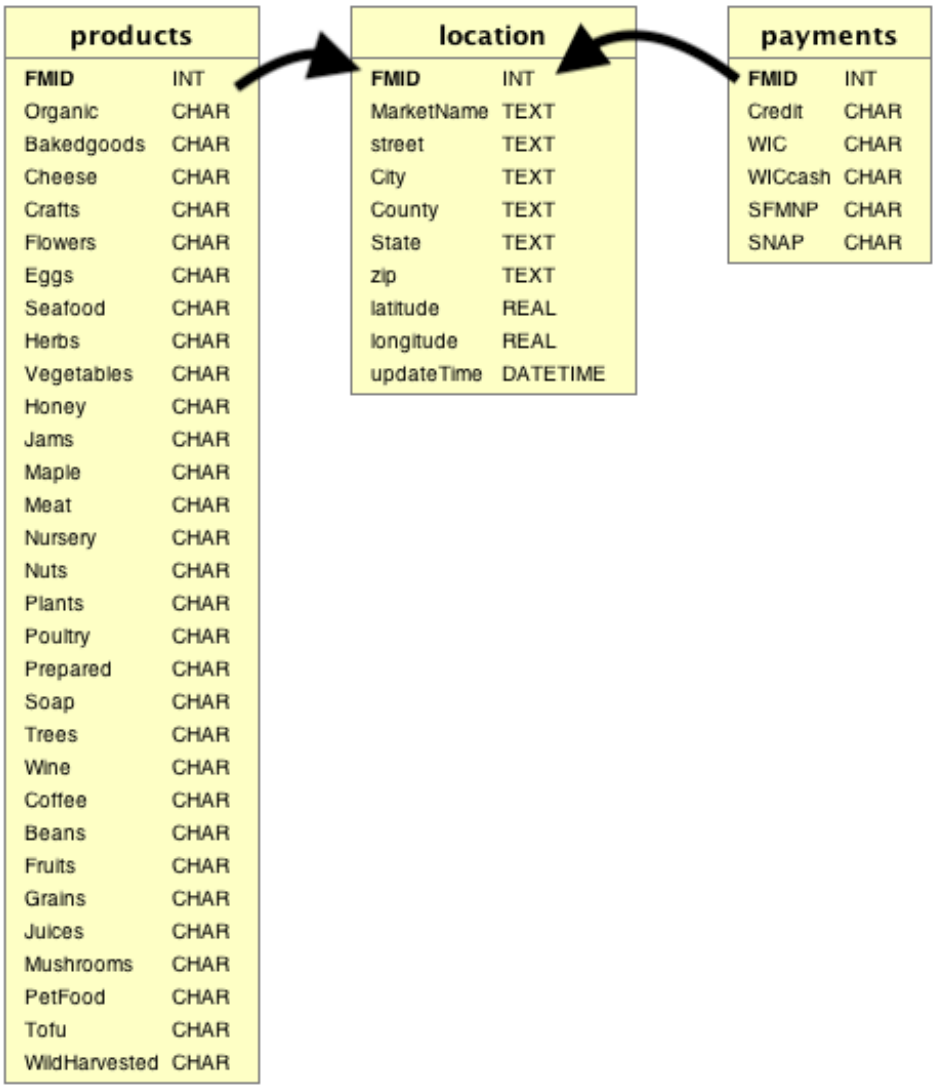


We also converted the values in the updateTime column to ISO format using the GREL expression: *value.toDate('d/M/y H:m:s')* after trimming and collapsing whitespace.

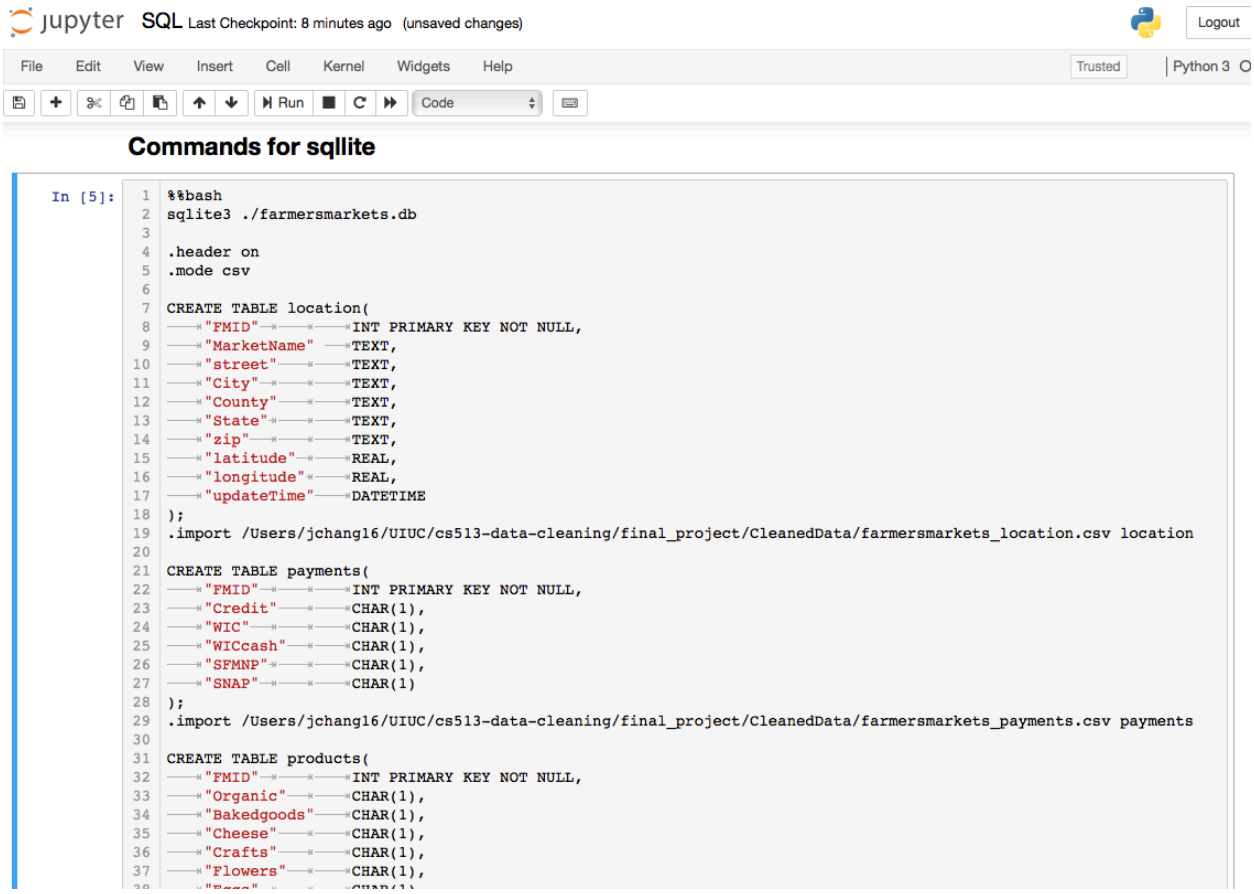
4. Data Cleaning Results

4.1 Relational Database Schema

The following Entity Relationship shows the schema we developed for our dataset. We broke our cleaned dataset into 3 separate tables (found in CleanedData): location, payments, and products, with the FMID as the primary key for all of them. This ER diagram was generated using DBVisualizer after loading the separate tables into sqllite.



We loaded the 3 tables (in CleanedData) into sqlite, using the SQL/SQL.ipynb Jupiter notebook, and the commands are also contained in SQL/sqlite_commands.txt.



Then, we develop a few integrity constraints which we ran in SQL/SQL.ipynb notebook.

- check that FMID is an appropriate primary key: non-null and unique
- Ensure that data for my use case is non-null (specifically latitude, longitude, state, credit)
- latitude must be in [0,90] and longitude should be [-180, 180]
- Every FMID has single address (street, City, County, State, zip) if it exists

Integrity Constraints

1. Check that FMID is an appropriate primary key: non-null and unique

Looks good.

In [9]:
```
%%sql
SELECT * FROM location where FMID IS NULL OR FMID = '';
SELECT COUNT(distinct FMID) from location;
```

```
 * sqlite:///./farmersmarkets.db
Done.
Done.
```

Out[9]:

COUNT(distinct FMID)
8687

2. Ensure that data for my use case is non-null. Turns out that we have 29 rows where latitude and longitude information are missing.

- latitude and longitude
- state
- credit

In [10]:
```
%%sql
SELECT *
FROM location loc
WHERE loc.latitude IS NULL
        OR loc.latitude = ''
        OR loc.longitude IS NULL
        OR loc.longitude = ''
        LIMIT 5;
```

```
 * sqlite:///./farmersmarkets.db
Done.
```

Out[10]:

FMID	MarketName	street	City	County	State	zip	latitude	longitude	updateTime
2000001	Center for Design Practice - Mobile Farmers Market				MARYLAND				2013-01-01T05:00:00Z
1011689	Charlotte Regional Farmers Market	1801 YORKMONT ROAD	CHARLOTTE	MECKLENBURG	NORTH CAROLINA	28217			2015-11-09T01:23:36Z

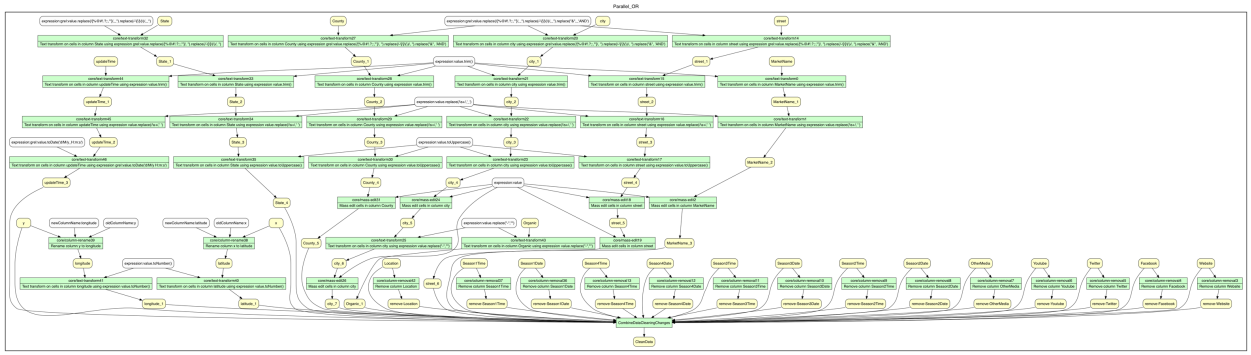
4.2 Workflow Model

Thanks to the or2yw tool, and the instructions found at https://pypi.org/project/or2ywtool/, we are able to create a workflow model of our data cleaning process very efficiently. All we needed was the operations history json file from our OpenRefine Data Cleaning steps, and we are ready to go with a few commands.
First, we do a pip install of the or2yw tool:
pip install or2ywtool

Then, we generate both a serial and parallel yw file
or2yw -i farmersmarkets_OperationHistory.json -o farmersmarkets_serial.yw
or2yw -i farmersmarkets_OperationHistory.json -o farmersmarkets_parallel.yw -t parallel

Finally, we install graphviz:
brew install graphviz

and then generate the model using the following commands:
or2yw -i farmersmarkets_OperationHistory.json -o farmersmarkets_parallel.png -ot png -t parallel



or2yw -i farmersmarkets_OperationHistory.json -o farmersmarkets_linear.png -ot png

4.3 Overview of Changes

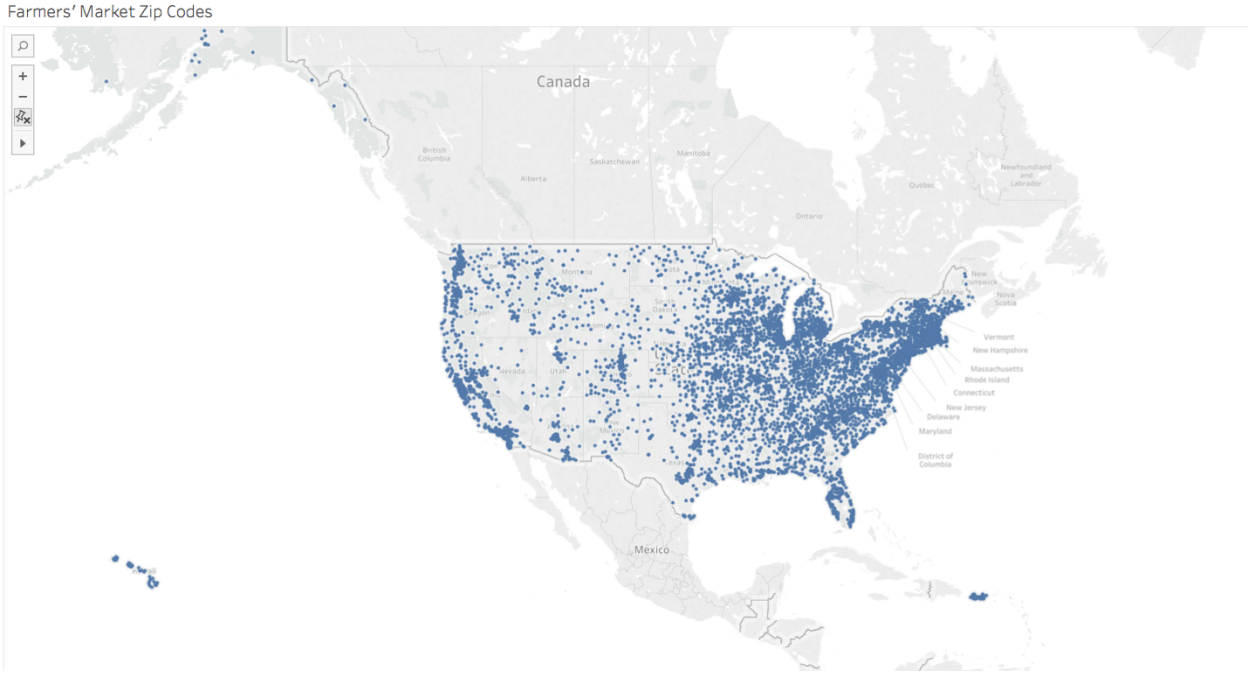
FMID	no change
MarketName	392 leading/trailing whitespaces trimmed; 43 whitespace collapsed;653 clustered
Website, Facebook, Twitter, Youtube, OtherMedia	columns removed
street, city, County, State, zip	street: 3175 cells had special characters removed/replaced; 305 had whitespaces trimmed, and 108 had whitespace collapsed, and 8301 were converted to uppercase, and 84 total were clustered city: 917 had whitespaces trimmed and 2 had whitespace collapsed; 68 total clustered cells clustered, 24 had "-" replaced, County: 126 cells had punctuation/special characters removed or replaced; 8140 were converted to uppercase State: all converted to uppercase zip:
Season1Date, Season1Time, Season2Date, Season2Time, Season3Date, Season3Time, Season4Date, Season4Time	columns removed
x, y	columns renamed to latitude and longitude, and 8658 cells converted to Numeric
location	column removed
Credit, WIC, WICcash, SFMNP, SNAP	no change

Organic, Bakedgoods, Cheese...PetFood, Tofu, WildHarvested (30 columns)	5043 cells had "-" replaced in Organic column
updateTime	219 cells had whitespace collapsed; 8384 cells changed to ISO date

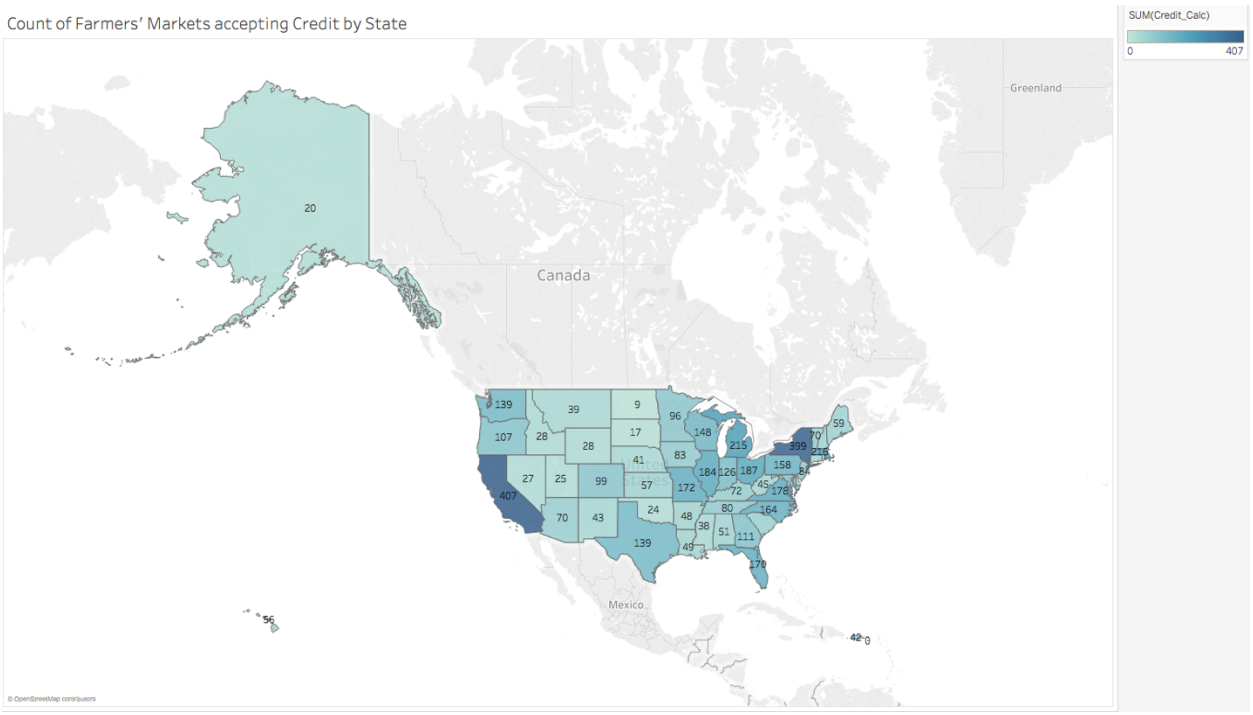
5. Conclusions and Future Work

5.1 Summary

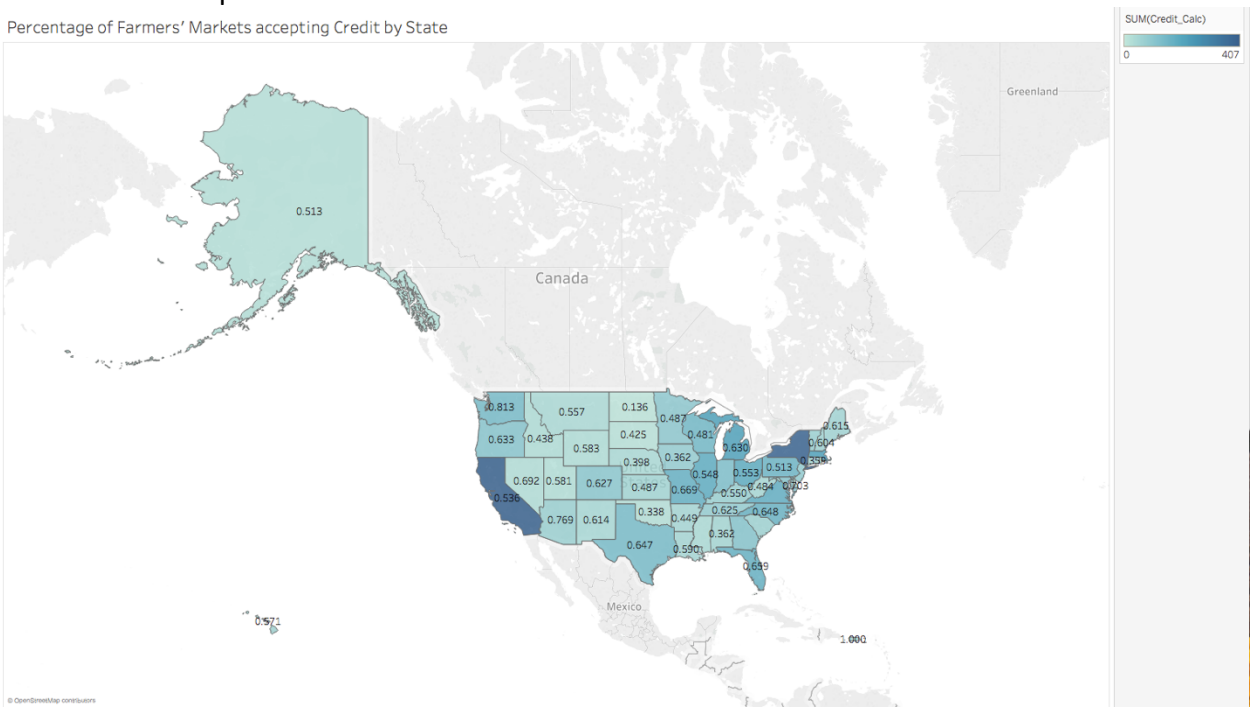
After our data cleaning exercise in OpenRefine, we are finally able to dive into our use cases. Our use case was to explore the adoption of credit card usage of the farmers' markets in our dataset, and so we run the cleaned dataset through Tableau to give ourselves a few views . The first one plots the zip codes corresponding to the farmers' markets, and it is essentially like a density plot that allows us to see that some of the more densely populated regions, for example in the northeast, have many farmers' markets, while the midwest and Alaska appears to be sparser.



Next, we see exactly how many farmers' markets each state has, and it is no surprise that more heavily populated states such as California and New York are shaded darker.



However, a more appropriate view that allows us to understand how much each state has adopted/accepted credit card usage is below. Here, we depict the percentage of farmers' markets that accept credit cards. The color intensity is the same as in the previous view which allows us to compare the overall number of farmers' markets between states.



Based off of the provided descriptions in this report, it should be very clear to the client what we have changed from the original dataset, as well as the challenges we faced in dealing with certain quality issues, and how we chose to resolve them.

5.2 Next Steps

In the first section of my project, I had listed a few other use cases that our farmers' market data supports, and with additional time, they could certainly be explored. For instance, we could delve into the specific product offerings of the farmers' markets and their distributions by region. We could also look into combinations of product offerings to determine whether people could get all their shopping done at specific farmers' market locations. We could also look into the season1Date to get a sense for how long some of these farmers' markets have been around for. There are many more questions that could be answered, and with some more time, the Tableau visualizations or dashboards could be advanced. It was nice that the dataset included location data that allowed us to utilize the map plot, but other visualization types could be used. Predictive models could also potentially built (e.g. predict whether or not credit card is accepted based on location and product offerings), but some additional work with OpenRefine or Python might be necessary to further prepare the dataset.