

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/dlv27sg8l9h6dv0oouqjdv8od7i9jjrf>
- farmersMarket_location.csv - <https://uofi.box.com/s/dx197g8g9k40bggtvfc0a2i3e7888ro>
- farmeresmarkets_payments.csv - <https://uofi.box.com/s/c7f5zkccix6dg4dhlidklnqf74rz877b>
- farmersmarkets_products.csv - <https://uofi.box.com/s/c9rfr41vbq6jutofoj1ymhh6dso3jymI>

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.

Cluster & Edit column "MarketName"

This feature helps you find groups of different cell values that might be alternative representations of the same thing. For example, the two strings "New York" and "new york" are very likely to refer to the same concept and just have capitalization differences, and "Gödel" and "Godel" probably refer to the same person. [Find out more ...](#)

Method: key collision Keying Function: fingerprint 209 clusters found

Cluster Size	Row Count	Values in Cluster	Merge?	New Cell Value
3	12	<ul style="list-style-type: none">Main Street Farmers Market (10 rows)Main Street Farmer's Market (1 rows)Main Street Farmers' Market (1 rows)	<input checked="" type="checkbox"/>	Main Street Farmers Market
3	5	<ul style="list-style-type: none">Rochester Downtown Farmers Market (3 rows)Downtown Rochester Farmers Market (1 rows)Downtown Rochester Farmers' Market (1 rows)	<input checked="" type="checkbox"/>	Rochester Downtown Farmers
3	3	<ul style="list-style-type: none">Harrison Farmer's Market (1 rows)Harrison Farmers Market (1 rows)Harrison Farmers' Market (1 rows)	<input checked="" type="checkbox"/>	Harrison Farmer's Market
3	4	<ul style="list-style-type: none">Goshen Farmers Market (2 rows)Goshen Farmer's Market (1 rows)Goshen Farmers' Market (1 rows)	<input checked="" type="checkbox"/>	Goshen Farmers Market
3	5	<ul style="list-style-type: none">Irvington Farmers Market (3 rows)Irvington Farmer's Market (1 rows)Irvington Farmers' Market (1 rows)	<input checked="" type="checkbox"/>	Irvington Farmers Market
3	4	<ul style="list-style-type: none">Northfield Farmers' Market (2 rows)Northfield Farmer's Market (1 rows)Northfield Farmers Market (1 rows)	<input checked="" type="checkbox"/>	Northfield Farmers' Market

Choices in Cluster

2 — 3

Rows in Cluster

2 — 12

Average Length of Choices

13 — 71

Length Variance of Choices

0 — 2.5

Select All Unselect All Export Clusters Merge Selected & Re-Cluster Merge Selected & Close Close

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.

OpenRefine FarmersMarket [Permalink](#)

Facet / Filter Undo / Redo 13 / 14 **8687 rows**

Show as: rows records Show: 5 10 25 50 rows

Filter:

Extract... Apply...

1. Create project

2. Text transform on 392 cells in column MarketName: value.trim()

3. Text transform on 43 cells in column MarketName: value.replace(/s+/, '')

3. Mass edit 653 cells in column MarketName

4. Remove column Website

5. Remove column Facebook

6. Remove column Twitter

7. Remove column Youtube

8. Remove column OtherMedia

9. Remove column Season2Date

10. Remove column Season2Time

11. Remove column Season3Date

12. Remove column Season3Time

13. Remove column Season4Date

14. Remove column Season4Time

State	zip	Season1Date	Season1Time	Season4Time	x	y	Location	Credit	WIC
Vermont	05828	06/14/2017 to 08/30/2017	Wed: 9:00 AM-1:00 PM;	Facet	140337	44.411036		Y	Y
Ohio		06/24/2017 to 09/30/2017	Sat: 9:00 AM-1:00 PM;	Text filter	7339387	41.3748009		Y	N
South Carolina	29682			Edit cells					N
Missouri	64759	04/02/2014 to 11/30/2014	Wed: 3:00 PM-6:00 PM;Sat: 8:00 AM-1:00 PM;	Edit column					N
New York	10029	July to November	Tue:8:00 am - 5:00 pm;Sat:8:00 am - 8:00 pm;	Transpose					N
Tennessee	37204	05/05/2015 to 10/27/2015	Tue: 3:30 PM-6:30 PM;	Sort...					N
New York	10027	06/10/2014 to 11/25/2014	Tue: 10:00 AM-7:00 PM;	View					N
Delaware	19801	05/16/2014 to 10/17/2014	Fri: 8:00 AM-11:00 AM;	Reconcile	-75.534460	39.742117	On a farm from: a barn, a greenhouse, a tent, a stand, etc	N	N
District of Columbia	20009	05/03/2014 to 11/22/2014	Sat: 9:00 AM-1:00 PM;		-77.0320505	38.9169984	Other	Y	Y
District of Columbia	20011	04/09/2016 to 11/19/2016	Sat: 9:00 AM-1:00 PM;		-77.0334486	38.9559783		Y	Y

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')`

Custom text transform on column street

Expression Language General Refine Expression Language (GREL)

`value.replace(/[%@#!.?:;,"]/, '').replace(/-\\[\\]\\\\/ , '').replace("&", 'AND')` No syntax error.

Preview History Starred Help

row	value	value.replace(/[%@#!.?:;,"]/, ...
1.	null	Error: replace expects 3 strings, or 1 string, 1 regex, and 1 string
2.	6975 Ridge Road	6975 Ridge Road
3.	106 S. Main Street	106 S Main Street
4.	10th Street and Poplar	10th Street and Poplar
5.	112th Madison Avenue	112th Madison Avenue
6.	3000 Granny White Pike	3000 Granny White Pike
7.	400 West 405th Street and Adam Clayton Powell Jr Blvd	400 West 405th Street and Adam Clayton Powell Jr Blvd

On error ☒ keep original ☐ set to blank ☐ store error ☐ Re-transform up to 10 times until no change

OK Cancel

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

Facet / Filter Undo / Redo 17 / 17 **8687 rows**

Show as: **rows** records Show: 5 10 25 50 rows « first < pre

Filter:

Extract... Apply...

0. Create project

1. Text transform on 392 cells in column MarketName: value.trim()

2. Text transform on 43 cells in column MarketName: value.replace(/s+/, ' ')

3. Mass edit 653 cells in column MarketName

4. Remove column Website

5. Remove column Facebook

6. Remove column Twitter

7. Remove column Youtube

8. Remove column OtherMedia

9. Remove column Season2Date

10. Remove column Season2Time

11. Remove column Season3Date

12. Remove column Season3Time

13. Remove column Season4Date

FMID	MarketName	street	city	County	State	zip	Season1Date	Season1Time	x
1018261	Caledonia Farmers Market Association - Danville			Caledonia	Vermont	05828	06/14/2017 to 08/30/2017	Wed: 9:00 AM-1:00 PM;	-72.140337
1018318	Stearns Homestead Farmers' Market						06/24/2017 to 08/30/2017	Sat: 9:00 AM-1:00 PM;	-81.7339387
1009364	106 S. Main Street Farmers Market								2.8187
1010691	10th Steet Community Farmers Market								4.2746191
1002454	112st Madison Avenue								3.9493
1011100	12 South Farmers Market	3000 Granny White Pike	Nashville						6.790709
1009845	125th Street Fresh Connect Farmers' Market	163 West 125th Street and Adam Clayton Powell Jr Blvd	New York	New York	New York	10011			3.9482477
1005586	12th & Brandywine Urban Farm Market	12th AND Brandywine Streets	Wilmington	New Castle	Delaware	19801	05/16/2014 to 10/17/2014	Fri: 8:00 AM-11:00 AM;	-75.534460
1008071	14&U Farmers'	1400 U	Washington	District of	District of	20009	05/03/2014 to 08/30/2014	Sat: 9:00 AM-1:00 PM;	-77.0320505

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

Cluster & Edit column "street"

This feature helps you find groups of different cell values that might be alternative representations of the same thing. For example, the two strings "New York" and "new york" are very likely to refer to the same concept and just have capitalization differences, and "Gödel" and "Godel" probably refer to the same person. [Find out more ...](#)

Method **key collision** Keying Function **fingerprint** **8 clusters found**

Cluster Size	Row Count	Values in Cluster	Merge?	New Cell Value
3	3	<ul style="list-style-type: none"> 5TH AND MAIN STREET (1 rows) 5TH STREET AND MAIN STREET (1 rows) MAIN STREET AND 5TH STREET (1 rows) 	<input checked="" type="checkbox"/>	5TH AND MAIN STREET
3	3	<ul style="list-style-type: none"> DOWNTOWN MAIN STREET (1 rows) MAIN STREET DOWNTOWN (1 rows) MAIN STREET- DOWNTOWN (1 rows) 	<input checked="" type="checkbox"/>	DOWNTOWN MAIN STREET
2	2	<ul style="list-style-type: none"> 555 14TH STREET WEST (1 rows) 555 WEST 14TH STREET (1 rows) 	<input checked="" type="checkbox"/>	555 14TH STREET WEST
2	2	<ul style="list-style-type: none"> 1ST NORTH ST (1 rows) NORTH 1ST ST (1 rows) 	<input checked="" type="checkbox"/>	1ST NORTH ST
2	2	<ul style="list-style-type: none"> 124TH STREET AND 5TH AVENUE/ MARCUS GARVEY PARK (1 rows) MARCUS GARVEY PARK 124TH STREET AND 5TH AVENUE (1 rows) 	<input checked="" type="checkbox"/>	124TH STREET AND 5TH AVE
2	2	<ul style="list-style-type: none"> 3RD AND MAIN ST (1 rows) MAIN ST AND 3RD (1 rows) 	<input checked="" type="checkbox"/>	3RD AND MAIN ST

Choices in Cluster: 2 — 3

Rows in Cluster: 2 — 3

Average Length of Choices: 12 — 47

Length Variance of Choices: 0 — 3.3000000000000003

Select All Unselect All Export Clusters Merge Selected & Re-Cluster Merge Selected & Close Close

Cluster & Edit column "street"

Use

This feature helps you find groups of different cell values that might be alternative representations of the same thing. For example, the two strings "New York" and "new york" are very likely to refer to the same concept and just have capitalization differences, and "Gödel" and "Godel" probably refer to the same person. [Find out more ...](#)

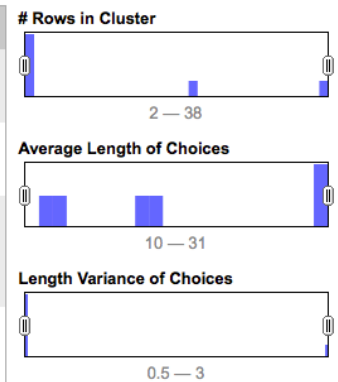
Method key collision

Keying Function ngram-fingerprint

Ngram Size 2

6 clusters found

Cluster Size	Row Count	Values in Cluster	Merge?	New Cell Value
2	38	<ul style="list-style-type: none"> MAIN STREET (37 rows) MAIN STREEET (1 rows) 	<input checked="" type="checkbox"/>	MAIN STREET
2	2	<ul style="list-style-type: none"> 12TH STREET (1 rows) 12THSTREET (1 rows) 	<input checked="" type="checkbox"/>	12TH STREET
2	2	<ul style="list-style-type: none"> EAST SIDE OF COURT HOUSE SQUARE (1 rows) EAST SIDE OF COURTHOUSE SQUARE (1 rows) 	<input checked="" type="checkbox"/>	EAST SIDE OF COURT HOU
2	2	<ul style="list-style-type: none"> WEST SIDE OF COUNTY COURT HOUSE (1 rows) WEST SIDE OF COUNTY COURTHOUSE (1 rows) 	<input checked="" type="checkbox"/>	WEST SIDE OF COUNTY COL
		Browse this cluster		
2	2	<ul style="list-style-type: none"> COMMUNITY CENTER (1 rows) UNITY COMMUNITY CENTER (1 rows) 	<input type="checkbox"/>	COMMUNITY CENTER
2	22	<ul style="list-style-type: none"> COURTHOUSE SQUARE (20 rows) COURT HOUSE SQUARE (2 rows) 	<input checked="" type="checkbox"/>	COURTHOUSE SQUARE



Select All Unselect All

Export Clusters

Merge Selected & Re-Cluster

Merge Selected & Close

Close

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.

OpenRefine FarmersMarket Permalink

Facet / Filter Undo / Redo 36 / 37

8687 rows

Show as: rows records Show: 5 10 25 50 rows

Extensions: Wikidata

Filter:

- 23. Text transform on 2 cells in column city: value.replace(/s+/,'')
- 24. Text transform on 8547 cells in column city: value.toUpperCase()
- 25. Mass edit 5 cells in column city
- 26. Text transform on 24 cells in column city: value.replace("-", "")
- 27. Mass edit 63 cells in column city
- 28. Text transform on 126 cells in column County: grel.value.replace(/%#?;:;"/, "").replace(/-|_|\\|/,"").replace("&", "AND")
- 29. Text transform on 0 cells in column County: value.trim()
- 30. Text transform on 0 cells in column County: value.replace(/s+/,'')
- 31. Text transform on 8140 cells in column County: value.toUpperCase()
- 32. Mass edit 17 cells in column County
- 33. Text transform on 0 cells in column State: grel.value.replace(/%#?;:;"/, "").replace(/-|_|\\|/,"").replace("&", "AND")
- 34. Text transform on 0 cells in column State: value.trim()
- 35. Text transform on 0 cells in column State: value.replace(/s+/,'')
- 36. Text transform on 8687 cells in column State: value.toUpperCase()

FMID	MarketName	street	city	County	State	zip	Season1Date	Season1Time	x	y
1018261	Caledonia Farmers Market Association - Danville		DANVILLE	CALEDONIA	VERMONT	05828	06/14/2017 to 08/30/2017	Wed: 9:00 AM-1:00 PM;	-72.140337	44.411036
1018318	Stearns Homestead Farmers' Market	6975 RIDGE ROAD	PARMA	CUYAHOGA	OHIO		06/24/2017 to 09/30/2017	Sat: 9:00 AM-1:00 PM;	-81.7339387	41.3748009
1009364	106 S. Main Street Farmers Market	106 S MAIN STREET	SIX MILE		SOUTH CAROLINA	29682			-82.8187	34.8042
1010691	10th Steet Community Farmers Market	10TH STREET AND POPLAR	LAMAR	BARTON	MISSOURI	64759	04/02/2014 to 11/30/2014	Wed: 3:00 PM-6:00 PM;Sat: 8:00 AM-1:00 PM;	-94.2746191	37.4956280
1002454	112st Madison Avenue	112TH MADISON AVENUE	NEW YORK	NEW YORK	NEW YORK	10029	July to November	Tue: 8:00 am - 5:00 pm;Sat: 8:00 am - 8:00 pm;	-73.9493	40.7939
1011100	12 South Farmers Market	3000 GRANNY WHITE PIKE	NASHVILLE	DAVIDSON	TENNESSEE	37204	05/05/2015 to 10/27/2015	Tue: 3:30 PM-6:30 PM;	-86.790709	36.118370
1009845	125th Street Fresh Connect Farmers' Market	163 WEST 125TH STREET AND ADAM CLAYTON POWELL JR BLVD	NEW YORK	NEW YORK	NEW YORK	10027	06/10/2014 to 11/25/2014	Tue: 10:00 AM-7:00 PM;	-73.9482477	40.8089533
1005586	12th & Brandywine Urban Farm Market	12TH AND BRANDYWINE STREETS	WILMINGTON	NEW CASTLE	DELAWARE	19801	05/16/2014 to 10/17/2014	Fri: 8:00 AM-11:00 AM;	-75.534460	39.742117
1008071	14&U Farmers' Market	1400 U STREET NW	WASHINGTON	DISTRICT OF COLUMBIA	DISTRICT OF COLUMBIA	20009	05/03/2014 to 11/22/2014	Sat: 9:00 AM-1:00 PM;	-77.0320505	38.9169984
1012710	14th & Kennedy Street Farmers Market	5500 COLORADO AVENUE NW	WASHINGTON	DISTRICT OF COLUMBIA	DISTRICT OF COLUMBIA	20011	04/09/2016 to 11/19/2016	Sat: 9:00 AM-1:00 PM;	-77.0334486	38.9559783
1016782	175th Street Greenmarket	175TH STREET BETWEEN WADSWORTH AND BROADWAY	NEW YORK	NEW YORK	NEW YORK	10033	06/29/2017 to 11/22/2017	Tue: 8:00 AM-6:00 PM;	-73.938049	40.846354

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.

Replace

Find:

☐ case insensitive ☐ whole word ☐ regular expression

Leave blank to add the replacement string after each character.

Check "regular expression" to find special characters (new lines, tabulations...) or complex patterns.

Replace with:

☐ use \n for new lines, \t for tabulation, \\n for \n, \\t for \t.

If "regular expression" option is checked and finding pattern contains groups delimited with parentheses, \$0 will return the complete string matching the pattern, and \$1, \$2... the 1st, 2d... group.

OK

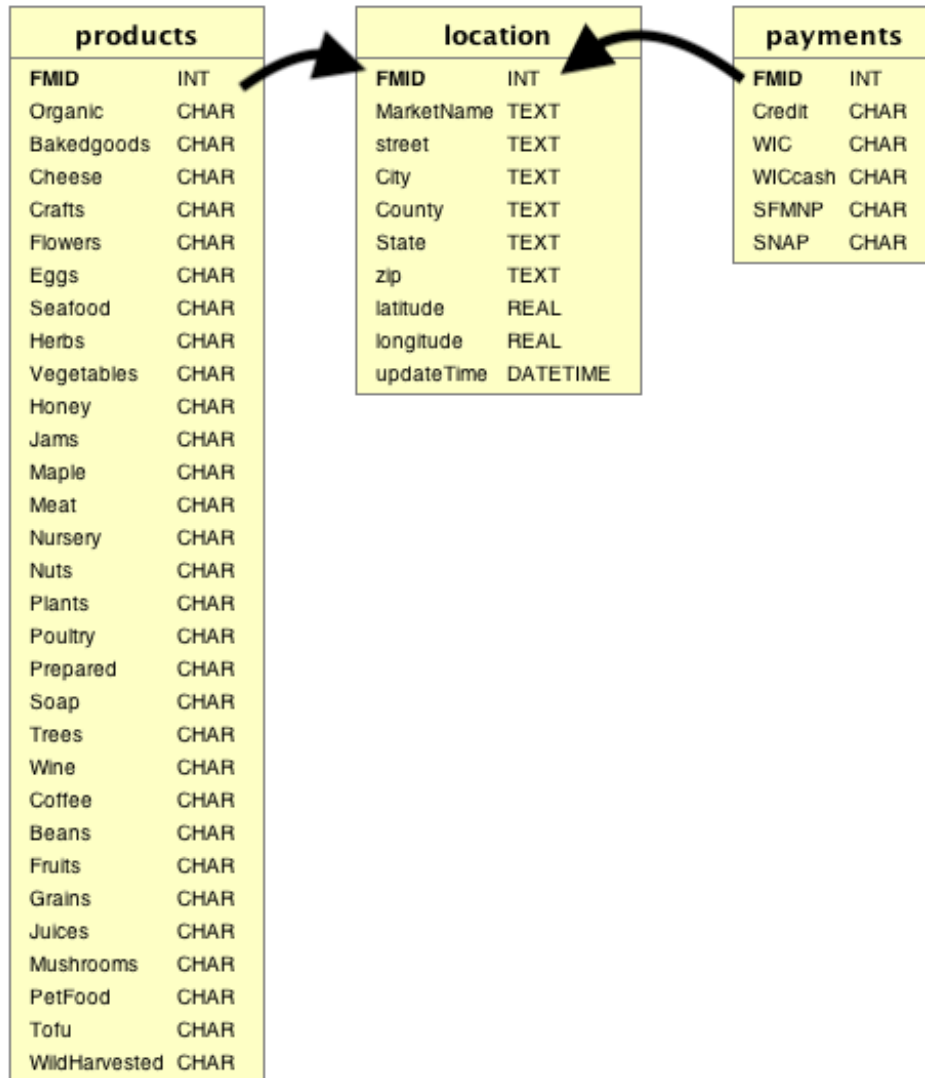
Cancel

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.

FileEditViewInsertCellKernelWidgetsHelp

TrustedPython 3

Commands for sqllite

```
In [5]: 1 %%bash
2 sqlite3 ./farmersmarkets.db
3
4 .header on
5 .mode csv
6
7 CREATE TABLE location(
8     "FMID" INT PRIMARY KEY NOT NULL,
9     "MarketName" TEXT,
10    "street" TEXT,
11    "City" TEXT,
12    "County" TEXT,
13    "State" TEXT,
14    "zip" TEXT,
15    "latitude" REAL,
16    "longitude" REAL,
17    "updateTime" DATETIME
18 );
19 .import /Users/jchang16/UIUC/cs513-data-cleaning/final_project/CleanedData/farmersmarkets_location.csv location
20
21 CREATE TABLE payments(
22     "FMID" INT PRIMARY KEY NOT NULL,
23     "Credit" CHAR(1),
24     "WIC" CHAR(1),
25     "WICcash" CHAR(1),
26     "SFMNP" CHAR(1),
27     "SNAP" CHAR(1)
28 );
29 .import /Users/jchang16/UIUC/cs513-data-cleaning/final_project/CleanedData/farmersmarkets_payments.csv payments
30
31 CREATE TABLE products(
32     "FMID" INT PRIMARY KEY NOT NULL,
33     "Organic" CHAR(1),
34     "Bakedgoods" CHAR(1),
35     "Cheese" CHAR(1),
36     "Crafts" CHAR(1),
37     "Flowers" CHAR(1),
```

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
```

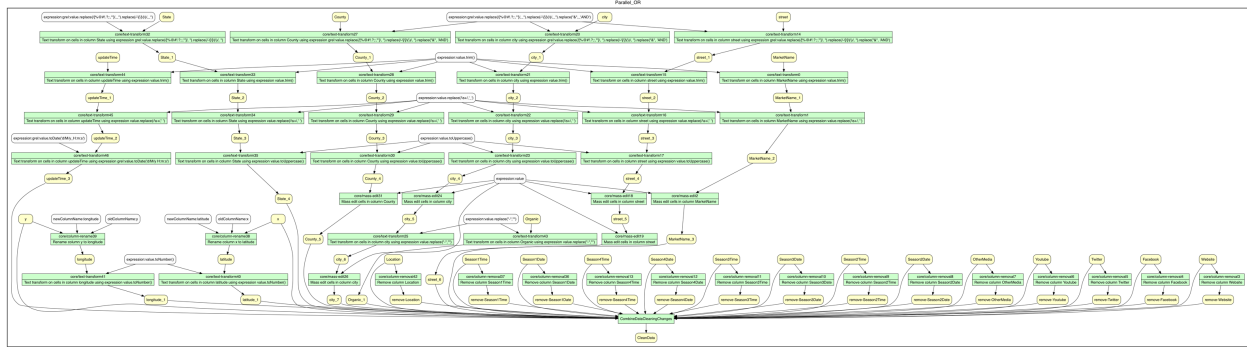
```
or2yw -i farmersmarkets_OperationHistory.json -o farmersmarkets_parallel.yw -t parallel
```

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

<i>FMID</i>	no change
<i>MarketName</i>	392 leading/trailing whitespaces trimmed; 43 whitespace collapsed; 653 clustered
<i>Website, Facebook, Twitter, Youtube, OtherMedia</i>	columns removed
<i>street, city, County, State, zip</i>	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:
<i>Season1Date, Season1Time, Season2Date, Season2Time, Season3Date, Season3Time, Season4Date, Season4Time</i>	columns removed
<i>x, y</i>	columns renamed to latitude and longitude, and 8658 cells converted to Numeric
<i>location</i>	column removed
<i>Credit, WIC, WICcash, SFMNP, SNAP</i>	no change

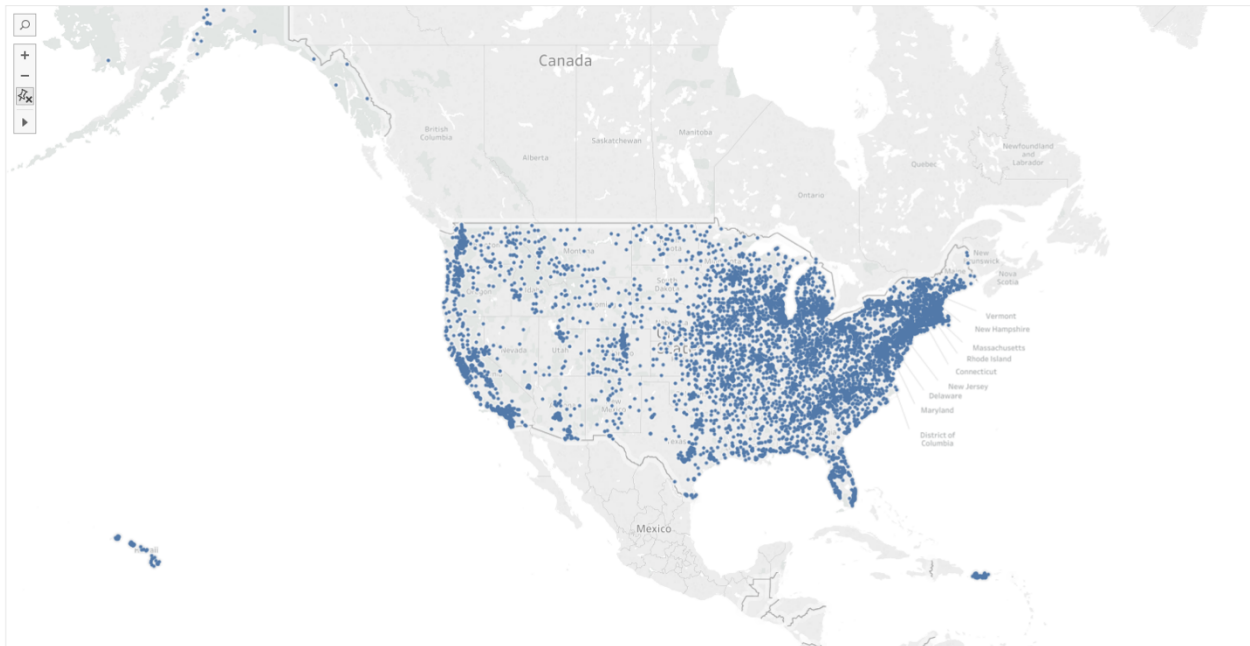
<i>Organic, Bakedgoods, Cheese...PetFood, Tofu, WildHarvested (30 columns)</i>	5043 cells had "-" replaced in Organic column
updateTime	219 cells had whitespace collapsed; 8384 cells changed to ISO date

5. Conclusions and Future Work

5.1 Summary

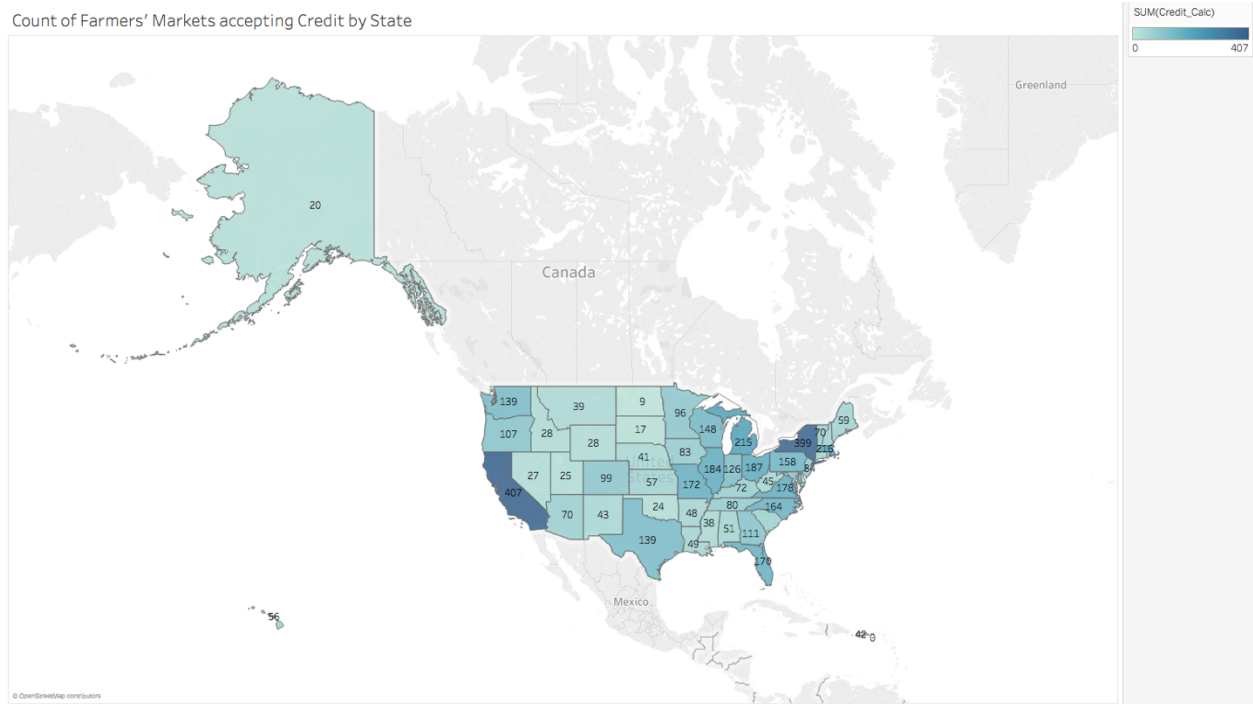
After our data cleaning exercise in OpenRefine, we are finally able to dive into our use cases. Our use case was to explore the adoption of credit card usage of the farmers' markets in our dataset, and so we run the cleaned dataset through Tableau to give ourselves a few views. The first one plots the 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.

Farmers' Market Zip Codes



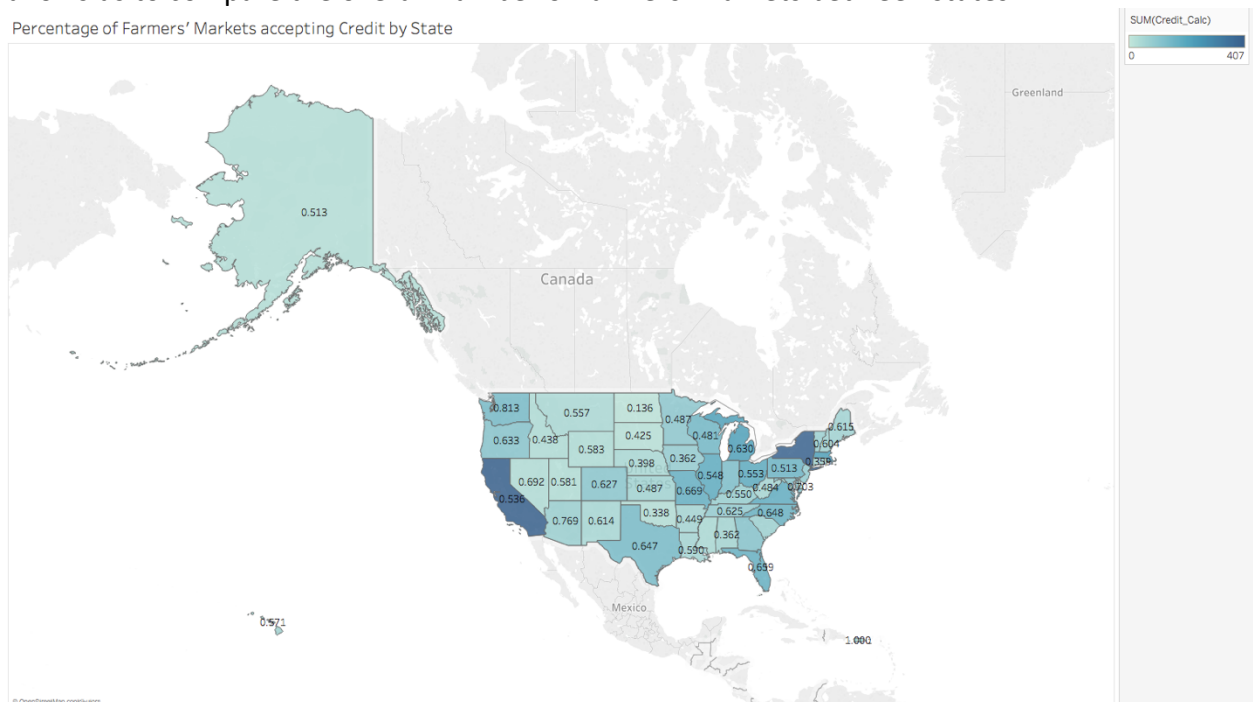
Next, we see exactly how many farmers' markets each state has, and it is no surprise that more heavily populated states such as California and New York are shaded darker.

Count of Farmers' Markets accepting Credit by State



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.

Percentage of Farmers' Markets accepting Credit by State



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 be 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.