

## **CS 513 Final Project**

**Name: Pranav Velamakanni netid: pranavv2**

**Name: Tarik Koric netid: koric1**

Overview of data: Tarik

Data cleaning python: Tarik

Application python: Pranav

Open Refine: Pranav

SQL: Tarik

Yesworkflow: Pranav

Paper: Both

## Overview and initial assessment of the dataset

The data chosen to be cleaned and analyzed was farmer's market data across the entire United States including territories of Puerto Rico and the U.S Virgin Islands. The unedited data originally has 8812 rows with 59 columns all in one csv file. The data contains much of the information you would expect to see for farmers market which includes the market name, social media information, address information, types of items sold at the market, time and date of operating hours, and time the information was last updated. A high overview of the data shows much of the information is fairly structured and relatively easy to read. There are however many distinctive problems with the data that would be a problem for further analysis.

All of the Market names are shown to be present in the data but are not always written in the same manner. Some punctuation and capitalization for similar names is different which can be difficult for more precise analysis. Social media information seems to be only present if the market itself has any. A lot of this information is null since its assuming many don't have a digital presence. Address information is also not all updated exactly, some markets have coordinates but do not have city, county, zip code, or state information. This is later discussed on how exactly the data was added in the cleaning process. Information about times and dates are very difficult to read with most of the times for different days of the week all put in the same column. Having the data spread out for each day of the week and time would have been a lot easier to read and better for analysis to handle. Coordinate data using longitude and latitude all seem to be populated which is important to use to be able to locate the markets on a map even without exact address information. Payment information as well as food sold is shown with either Y as yes, N as no, or blank for null values. Some columns have dashes as null values which should be ignored or changed to null. Finally all update times seem to be accurate as well. However any column that has a date is not in a consistent date format. Much of the cleaning process will involve changing the dates to a suitable date time format for easier analysis.

The data use cases identified was to be able to easily find a farmers market close to oneself by address and be able to look up times for that markets open hours, while also being able to identify the items that could be found at the specific market. Other higher level cases would show what days of the week are most open, number of markets per state, and the different items sold at each market showing which ones are most present from greatest to least. One of the main goals was to completely fill out all the address information for each market that had missing data for city, county, state and zip code information. This is essential to identifying where the actual market is located. Other information like having different times for each day of the week for each season of the farmers market is also essential to be able to show. Having all the date time formats in the same manner and formatting all the strings in a consistent matter will also be done.

The dataset in itself has enough clean data that already allows for some analysis. The coordinate data could be put into a tableau map that would show all the places where a market

exists in a geographical format, since almost every row has coordinate values. The items found in each market is also in the necessary format and can be analyzed further right away. Social media is something that is very difficult since many of these markets need to be online anyway to have something. If they only advertise locally then there is no need for this data. For data analysis it is not very useful. Also there is a column that describes the location venue of the actual market. This column is missing lots of values and someone would physically need to describe every farmers market in the dataset to fill in these values. Something that would take lots of time and effort to do.

## **Data cleaning with Python and OpenRefine**

The dataset consists of 59 columns with various data types. The dataset had several missing values which include, location information which is critical to identify where a particular farmers market was located. It also includes values with different formats that make it hard to parse a particular entry in the data.

The data cleaning and pre-processing tasks have been performed by the following tools:

- Python
- Open Refine

A command-line based script (app.py included in the workspace) has been created that can accept the original Farmers Market dataset (<https://www.ams.usda.gov/local-food-directories/farmersmarkets>) as a CSV and perform the required cleaning steps before exporting a new CSV with the processed dataset.

Data cleaning for this dataset involved removing columns, splitting columns into several other columns to ensure that data is organized and accessible easily. Additional steps like modifying formats especially for dates in string formats to ensure that they can be converted to a date-time object. Filling missing values was also performed for location information like zip codes, county, city, and states to ensure consistency and query efficiency.

The following tasks are performed in the python script:

1. Split season dates into 2 separate columns, start date and end date for each season. This is primarily done to ensure that all values of these columns follow the same format which makes it easier to perform queries on the database to fetch specific information like the start dates of various farmers' markets. This process is repeated for all 4 seasons resulting in 8 new columns, 2 columns for each season.
2. Split times based on the day of the week for each season. The open times for a market are split into 7 columns based on the day of the week. The original string for each season time value is split and then sorted based on the day before placing the

corresponding value in the dataset. This step is performed for all 4 seasons in the dataset resulting in 28 new columns, 7 columns for each season. This step further enhances the query efficiency of the dataset, by splitting the values by the day of the week, we can obtain open times for a specific day for a specific market without parsing the full string from the original columns for every query.

3. Converting months to dates for the new start and end date columns created for each season. The format for this is not standard across all values. Some of them are provided with the name of the month followed by the date and some of them are provided with the full date including the year. To address this discrepancy, a standard date format is used across all fields. For values with missing year data, the year from the UpdateTime column is used since that year reflects the last time the data has been refreshed in the dataset. For values with a missing day of the month, the first day is used. This function is applied to all start and end date columns for all seasons. This step ensures that these values can be converted to a valid date data type which enables queries targeting a specific time frame.
4. Remove unused columns. This step involves removing all unused columns in the dataset, this includes the 8 season dates and time columns, the OtherMedia, and location columns. The 8 season columns are used in the previous steps and are split into separate columns, the other 2 unused columns consist of missing and duplicate values that do not contribute to adding value to the dataset. This step further aids in condensing the dataset to only contain relevant information.
5. Replace missing values with None. The dataset contains a significant amount of missing values marked with a "-" string. These strings make it challenging to distinguish between a valid value and a missing value. So all these missing values are converted to Python's None type. This particular value is present in the Organic column of the dataset. This ensures that missing values have a valid none type, this helps to sort values and optimize queries by ignoring these values.
6. Convert timestamps with 12 PM to 12. This change has been performed to avoid an issue that caused type conversions from string to DateTime objects to fail. All noon times have been modified in all 9 columns which contain date information.
7. Convert month to date. The updateTime column contains date information with the abbreviated month as the name, this hinders query efficiency and it is not consistent across all columns of the dataset containing date information. A dictionary mapping each month to a valid date was used to modify all values of this column to comply with the same format used in the dataset.
8. Convert all not a number (NaN) values to Python None type. This step modifies all the missing data points in the dataset to a None type object. The original NaN is imported as a string which makes it harder to parse, ignore, and perform queries on. By converting it a none type, the SQL database is able to identify these values and ignore them in queries when required.
9. Add missing zip codes using the coordinates. The dataset contains missing zip code values which can be replaced by other location data provided in the dataset like the latitude and longitude coordinates. Using a library that provides this information, all missing zip codes have been added using these coordinates. This further ensures

consistency in the dataset and also improves search of nearby farmers market based on a certain zip code.

10. Add missing locations using the zip code. The dataset contains missing county, city, and state values. With all missing zip codes filled in using the step above, the missing location information can be obtained using this zip code. This ensures consistency and queries targeting a specific country, city, or state yield all available results.

Season1Date	Season1 Start Date	Season1 End Date
06/14/2017 to 08/30/2017	2017-06-14T00:00:00Z	2017-08-30T00:00:00Z
06/24/2017 to 09/30/2017	2017-06-24T00:00:00Z	2017-09-30T00:00:00Z
NaN	NaN	NaN
04/02/2014 to 11/30/2014	2014-04-02T00:00:00Z	2014-11-30T00:00:00Z
July to November	2012-07-01T10:38:22Z	2012-11-01T10:38:22Z
05/05/2015 to 10/27/2015	2015-05-05T00:00:00Z	2015-10-27T00:00:00Z
06/10/2014 to 11/25/2014	2014-06-10T00:00:00Z	2014-11-25T00:00:00Z
05/16/2014 to 10/17/2014	2014-05-16T00:00:00Z	2014-10-17T00:00:00Z
05/03/2014 to 11/22/2014	2014-05-03T00:00:00Z	2014-11-22T00:00:00Z
04/09/2016 to 11/19/2016	2016-04-09T00:00:00Z	2016-11-19T00:00:00Z

Season1Data split into 2 columns, modified date string format before being converted to a standard date-time object.

Season1Time	Season1Time Mon	Season1Time Tue	Season1Time Wed	Season1Time Thu	Season1Time Fri	Season1Time Sat	Season1Time Sun
Wed: 9:00 AM-1:00 PM;	NaN	NaN	9:00 AM-1:00 PM	NaN	NaN	NaN	NaN
Sat: 9:00 AM-1:00 PM;	NaN	NaN	NaN	NaN	NaN	9:00 AM-1:00 PM	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Wed: 3:00 PM-6:00 PM;Sat: 8:00 AM-1:00 PM;	NaN	NaN	3:00 PM-6:00 PM	NaN	NaN	8:00 AM-1:00 PM	NaN
Tue:8:00 am - 5:00 pm;Sat:8:00 am - 8:00 pm;	NaN	8:00 am - 5:00 pm	NaN	NaN	NaN	8:00 am - 8:00 pm	NaN
Tue: 3:30 PM-6:30 PM;	NaN	3:30 PM-6:30 PM	NaN	NaN	NaN	NaN	NaN
Tue: 10:00 AM-7:00 PM;	NaN	10:00 AM-7:00 PM	NaN	NaN	NaN	NaN	NaN
Fri: 8:00 AM-11:00 AM;	NaN	NaN	NaN	NaN	8:00 AM-11:00 AM	NaN	NaN
Sat: 9:00 AM-1:00 PM;	NaN	NaN	NaN	NaN	NaN	9:00 AM-1:00 PM	NaN
Sat: 9:00 AM-1:00 PM;	NaN	NaN	NaN	NaN	NaN	9:00 AM-1:00 PM	NaN

Splitting the Season time column into 7 columns based on the day of the week.

The following tasks are performed in OpenRefine:

1. Cluster all values in the Market Names column to ensure that duplicates with the same name follow the same format. This step ensures that queries focusing on a specific market produce all possible results matching the name.
2. Trim whitespaces of all columns of the dataset. This step is performed to ensure consistency in the dataset and avoids whitespaces causing queries to miss certain values.
3. Market name, city, state, county columns are converted to a title case format. This enhances the presentation of the data and ensures that these values can be searchable.
4. Season date start, end, update time columns are converted to a date object. This data type gets identified by SQL and lets users perform queries targeting a specific date or a range.


**Cluster & Edit column "MarketName"**

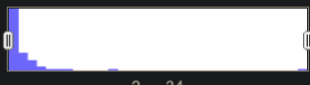
This feature helps you find groups of different cell values that might be alternative representations of the same thing. For example, the two strings "New York" and "new york" are very likely to refer to the same concept and just have capitalization differences, and "Gödel" and "Godel" probably refer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


Method key collision Keying Function fingerprint 239 clusters found

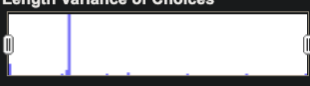
Cluster Size	Row Count	Values in Cluster	Merge?	New Cell Value
4	13	<ul style="list-style-type: none"> <li>Main Street Farmers Market (10 rows)</li> <li>MAIN STREET FARMERS MARKET (1 rows)</li> <li>Main Street Farmer's Market (1 rows)</li> <li>Main Street Farmers' Market (1 rows)</li> </ul>	<input checked="" type="checkbox"/>	Main Street Farmers Market
4	4	<ul style="list-style-type: none"> <li>Irvington Farmer's Market (1 rows)</li> <li>Irvington Farmers Market (1 rows)</li> <li>Irvington Farmers Market (1 rows)</li> <li>Irvington Farmers' Market (1 rows)</li> </ul>	<input checked="" type="checkbox"/>	Irvington Farmer's Market
3	3	<ul style="list-style-type: none"> <li>Wakefield Farmer's Market (1 rows)</li> <li>Wakefield Farmers Market (1 rows)</li> <li>Wakefield Farmers Market (1 rows)</li> </ul>	<input checked="" type="checkbox"/>	Wakefield Farmer's Market
3	4	<ul style="list-style-type: none"> <li>Columbus Farmers Market (2 rows)</li> <li>Columbus Farmers' Market (1 rows)</li> <li>columbus farmers market (1 rows)</li> </ul>	<input checked="" type="checkbox"/>	Columbus Farmers Market
3	5	<ul style="list-style-type: none"> <li>Rochester Downtown Farmers Market (3 rows)</li> <li>Downtown Rochester Farmers Market (1 rows)</li> <li>Downtown Rochester Farmers' Market (1 rows)</li> </ul>	<input checked="" type="checkbox"/>	Rochester Downtown Farmer's Market

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

# Choices in Cluster:  2 — 4

# Rows in Cluster:  2 — 34

Average Length of Choices:  14 — 71

Length Variance of Choices:  0 — 2.5

Open Refine edit of Market names.

A command-line application has been built that performs the above-discussed cleaning steps and exports the processed data as a new CSV. The tool also supports a flag to convert an existing CSV to a SQL database. The usage is as follows.

To install the dependencies required by the application, execute the following:

**`pip install uszipcode pandas`**

To clean a new CSV, download the Farmers Market data from <https://www.ams.usda.gov/local-food-directories/farmersmarkets>, and execute the following command from the workspace. A new file is created with the processed data.

- **`python app.py --file /Users/pranav/Downloads/Export.csv --clean`**

With this cleaned dataset, the OpenRefine recipe can be applied (file attached as recipe.json) in the project workspace. After this step, the final dataset can be exported as a database using the following:

- **`python app.py --file Processed.csv ---to-database`**

## Developing a relational schema

The data was loaded into an SQL database file (.db) using the python sqlite3 library. The file that is loaded is the cleaned csv file after it has been cleaned in Open Refine. Once the database file is measured appropriate queries can be run using a GUI like tool called sqlite studio. The integrity constraints checked for the database is to first check that each market has its own unique id (FMID) this ensure that there are no duplicates. Once that is done then the database is checked for markets with the same name, street, city, and state. This shows that a market was input into the database twice. The query will update the database based off of the newest update time and delete the duplicate values. The final integrity constraints look at season dates and times. Because there are 4 different seasons to input if a season column has the same dates and times as another season column then the larger season number would be left blank given this is duplicate data. This is done for all seasons and all data.

FMID	MarketName	Website	Facebook
1	1010241 Caledonia Farmers Market Association - Danville	https://sites.google.com/site/caledoniafarmersmarket/	Facebook https://www.facebook.com/DanvilleVT-Farmers-Market/
2	1010181 Swains Homestead Farmer Market	http://www.SwainsHomestead.com	SwainsHomesteadFarmersMarket
3	1000044 100th Street Farmers Market	http://thefoodtrust.org/farmers-markets/market/100th-street	N/A
4	1000464 100th Street Community Farmers Market	N/A	N/A
5	1000464 100th Street Community Farmers Market	N/A	N/A
6	1011001 12 South Farmers Market	http://www.12southfarmersmarket.com	12_South_Farmers_Market
7	1000101 120th Street Fresh Connect Farmers' Market	https://www.facebook.com/120thstreetfarmersmarket	https://www.facebook.com/pages/120th-Street-Farmers-Market/
8	1000088 12th & Randolph Urban Farm Market	https://www.facebook.com/pages/12th-Randolph-Urban-Farm-Market/	https://www.facebook.com/140FarmersMarket
9	1000071 14th & Kennedy Street Farmers Market	https://www.facebook.com/14thandkennedystreetfarmersmarket	https://www.facebook.com/14thandkennedystreetfarmersmarket
10	1001707 14th & Kennedy Street Farmers Market	https://www.facebook.com/14thandkennedystreetfarmersmarket	https://www.facebook.com/14thandkennedystreetfarmersmarket
11	1010702 170 Farm Stand	https://www.facebook.com/CommunityFoodAction/	https://www.facebook.com/CommunityFoodAction/
12	1010702 170th Street Greenmarket	https://www.grownc.org/greenmarkets/manhattan/170th-street	https://www.facebook.com/ManhattanGreenmarkets/
13	1000071 17th Ave Market	http://www.17thave.org/17thave-market	https://www.facebook.com/17thAveMarket/
14	1010704 17th Street Farmers Market	http://www.17thstreetfarmersmarket.com	https://www.facebook.com/17thStreetFarmersMarket/
15	1000088 18th and Christian Farmers Market	http://thefoodtrust.org/farmers-markets/market/18th-Christian	N/A
16	1000088 18th Street Farmers Market	N/A	N/A
17	1000088 18th Street Farmers Market	N/A	N/A
18	1000088 18th Street Farmers Market	N/A	N/A
19	1000088 18th Street Farmers Market	N/A	N/A
20	1000088 18th Street Farmers Market	N/A	N/A
21	1000088 18th Street Farmers Market	N/A	N/A
22	1000088 18th Street Farmers Market	N/A	N/A
23	1000088 18th Street Farmers Market	N/A	N/A
24	1000088 18th Street Farmers Market	N/A	N/A
25	1000088 18th Street Farmers Market	N/A	N/A
26	1000088 18th Street Farmers Market	N/A	N/A
27	1000088 18th Street Farmers Market	N/A	N/A
28	1000088 18th Street Farmers Market	N/A	N/A
29	1000088 18th Street Farmers Market	N/A	N/A
30	1000088 18th Street Farmers Market	N/A	N/A
31	1000088 18th Street Farmers Market	N/A	N/A
32	1000088 18th Street Farmers Market	N/A	N/A
33	1000088 18th Street Farmers Market	N/A	N/A
34	1000088 18th Street Farmers Market	N/A	N/A
35	1000088 18th Street Farmers Market	N/A	N/A
36	1000088 18th Street Farmers Market	N/A	N/A
37	1000088 18th Street Farmers Market	N/A	N/A
38	1000088 18th Street Farmers Market	N/A	N/A
39	1000088 18th Street Farmers Market	N/A	N/A
40	1000088 18th Street Farmers Market	N/A	N/A
41	1000088 18th Street Farmers Market	N/A	N/A
42	1000088 18th Street Farmers Market	N/A	N/A

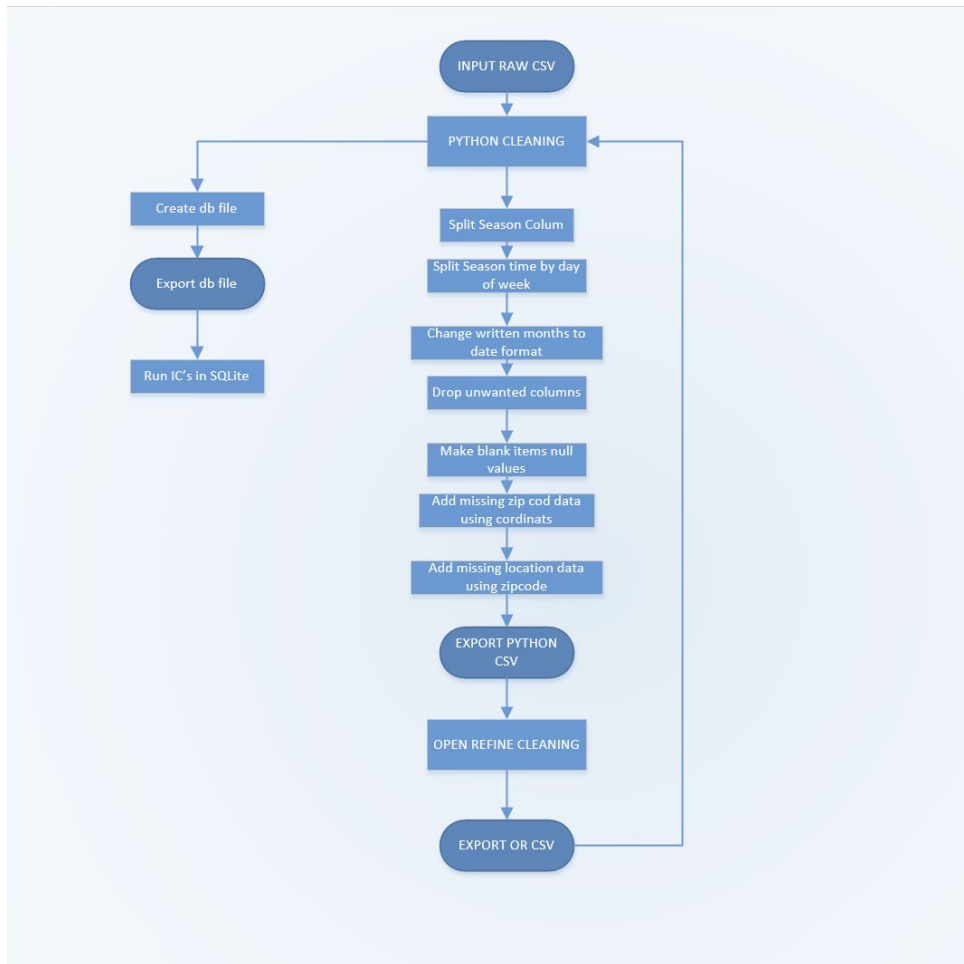
FMID	MarketName	Website	Facebook	Twitter	Youtube	street	city	County	State	zip	x	y	Crawl	USC	WDCash	SPREP	SNAP	Organic	Bakedgood	Cheese	Crafts	Flowers	Eggs	Seafood	Herbs
1	1010241 Caledonia Farmers Market Association - Danville	https://sites.google.com/site/caledoniafarmersmarket/	Facebook https://www.facebook.com/DanvilleVT-Farmers-Market/	SwainsHomesteadFarmersMarket	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

SQLStudio application used for checking market.db file.



## Workflow

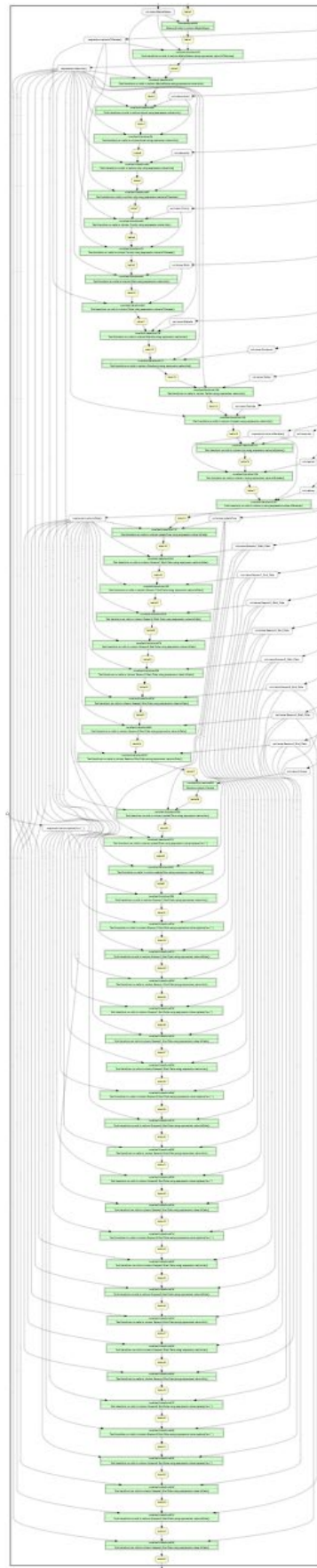
The workflow of the processes involved in cleaning the original dataset using Python and Open Refine before converting to a database is shown below.



In the flowchart above, the original farmers market CSV is passed into the Python program that performs a list of functions discussed in the previous section. This resulting CSV from the Python application is passed into OpenRefine which then performs a set of cleaning procedures. This final processed CSV contains all the optimizations designed for this dataset. To perform queries and obtain results, this CSV is passed into the same Python application with a special argument that converts the CSV to a SQL database that can be read by SQLite where queries can be executed to obtain results. The dependencies for this application and the steps to execute this have been mentioned in the section above.

The workflow diagram for the Open Refine steps was computed using YesWorkflow as shown below. The processed CSV from the Python script is imported to Open Refine where the

provided recipe is applied, the steps performed here are discussed above. The resulting CSV contains the final processed dataset.

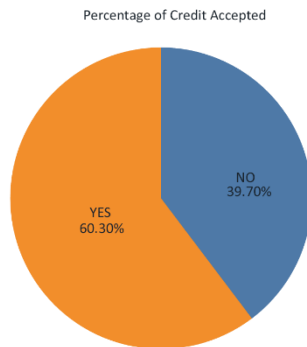


## Conclusions and Data Visualization

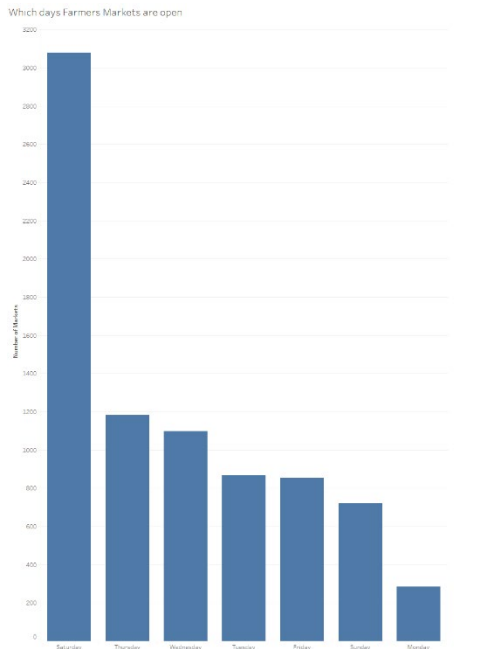
Further work was done to show the final results of the clean data set using tableau as a way to visualize the data for the user.

State	City	Street	Market Name	Website	Credit	Organic	Vegetables	Fruits	Meat
Illinois	Champaign	229 Mattis Ave.	Country Fair Farmers Market In Champaign	Null	N	Null	Y	Y	N
		310-330 N. Neil	The Land Connection Champaign Farmers' Market	<a href="http://thelandconnection.org/market">http://thelandconnection.org/market</a>	Y	Y	Y	Y	Y
		North 1st Street	Farmers Market On Historic North 1st Street	Null	N	N	Y	N	Y
	Urbana	400 South Vine Street	Urbana's Market At The Square	<a href="http://www.urbanainillinois.us/market">http://www.urbanainillinois.us/market</a>	Y	Y	Y	N	Y

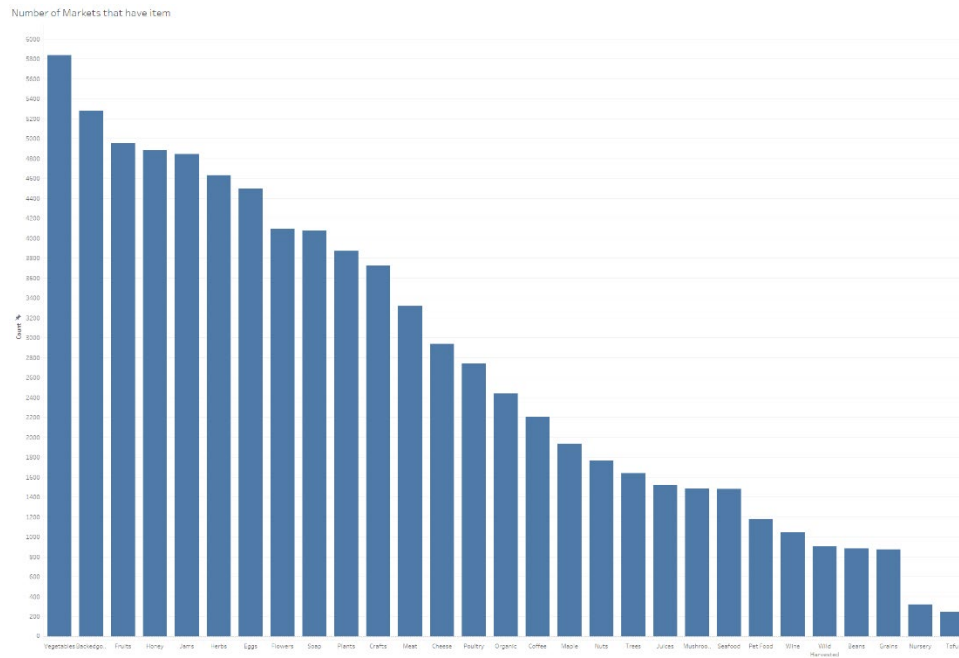
Data for Champaign-Urbana in a tabular form.



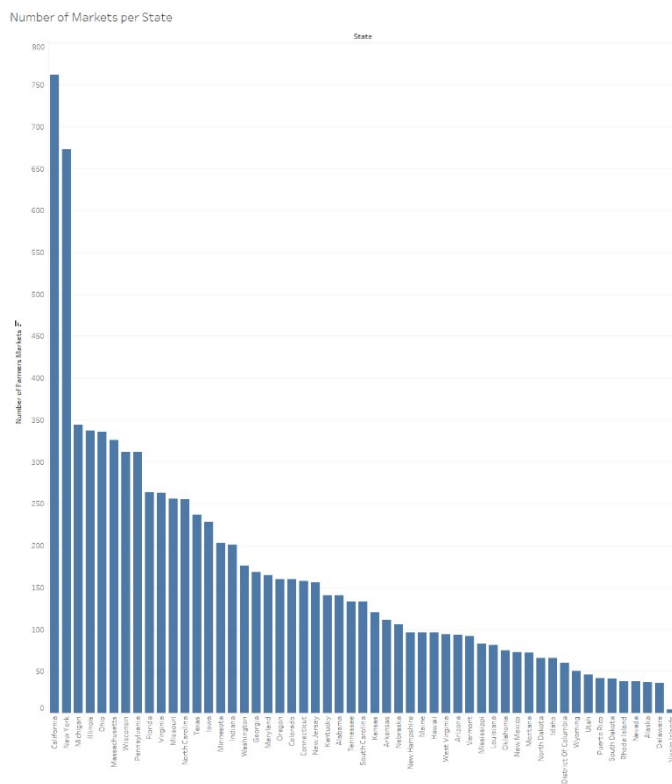
Acceptance of credit cards at the Farmers market.



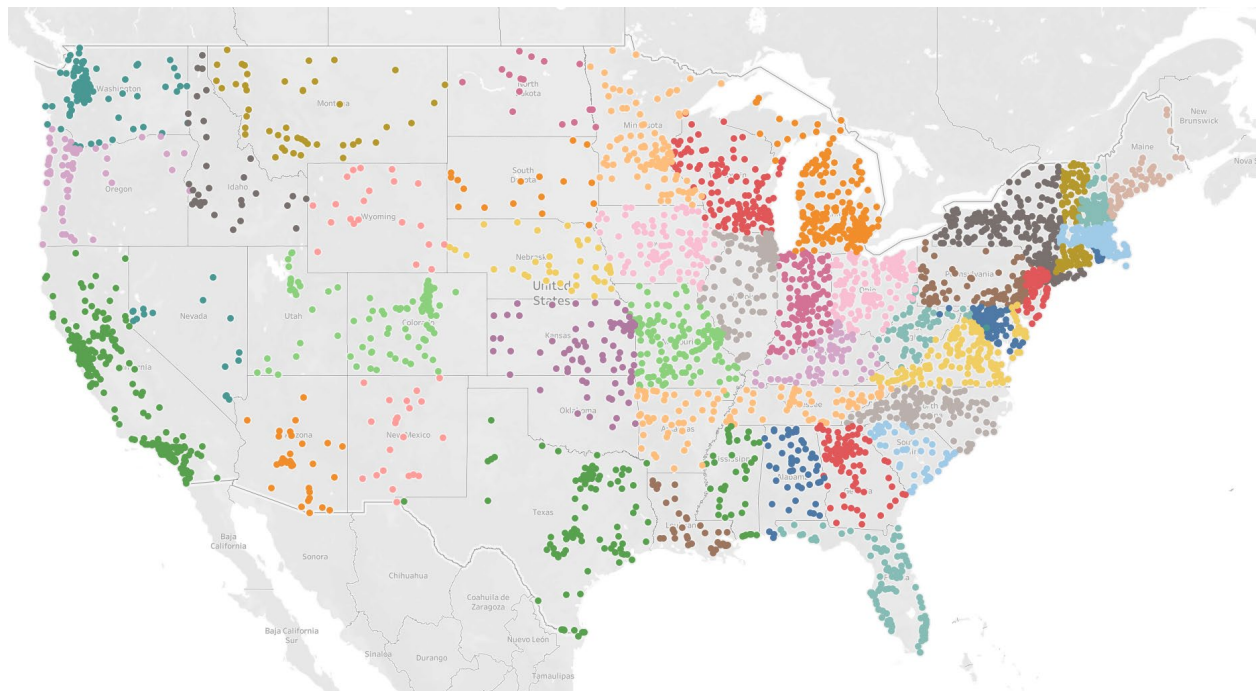
Days of the week that Farmers Market are open.



Count of which items show up at different Farmers Markets.



Number of markets per state.



Location of Farmers Markets.