

# CS 513: Final Project Report

By: Tao Li (taol4), Wanlin Yang(wanliny2)

## 1. Introduction and Overview

Farmers markets are an important way for citizens to eat healthy fresh food while supporting small family farms. The US Farmers Market dataset lists markets selling agricultural products. This data contains information about all the registered farmers markets in the United States. Users can access brief information, such as geographical coordinate, state, county, website, etc, and product categories of each farmers market in this dataset.

The original dataset has a total of 8817 rows and 59 columns. Each row of data is an instance of a market, while one market may have multiple rows. Each column is an attribute of a market, including market name, webset, categories of product, season time, last update time, etc. Some attribute data could be missing for a market instance.

This dataset is a thorough statistics of farmers markets all over the country, and valuable information for data analysis. However, the raw data in the csv file is not well formatted and very difficult to be processed by script languages. So our goal of this project is to do data cleaning using OpenRefine. We will also construct a cleaning workflow and check its integrity constraint with SQLite.

## 2. Initial Dataset Assessment & Use Case Discussion

### 2.1 Structure and Content of dataset

The latest UC Farmers Market dataset file can be exported from <https://www.ams.usda.gov/local-food-directories/farmersmarkets>. We can open the csv file directly or import it into OpenRefine.

- The dataset contains **8817** market information with **59** different columns or attributes.
- Each row of the dataset contains information of a market instance.
- Columns can be classified into several categories:

Column/Attribute	Data Type	Category	Description
FMID	integer	Identification	Identifier of each farmers market
MarketName	string	Identification	Name of each market
Website	string	Information	Website of each market
Facebook	string	Information	Facebook of each market
Twitter	string	Information	Twitter of each market
Youtube	string	Information	Youtube of each market
OtherMedia	string	Information	Any other media except those mentioned above
street	string	Location	Street
city	string	Location	City
County	string	Location	County
State	string	Location	State
zip	string	Location	Zip Code
Season1Date	date	Seasonality	Date of Season 1
Season1Time	time	Seasonality	Time of Season 1
Season2Date	date	Seasonality	Date of Season 2

Season2Time	time	Seasonality	Time of Season 2
Season3Date	date	Seasonality	Date of Season 3
Season3Time	time	Seasonality	Time of Season 3
Season4Date	date	Seasonality	Date of Season 4
Season4Time	time	Seasonality	Time of Season 4
x	numeric	Location	The longitude of the market
y	numeric	Location	The latitude of market
Location	string	Location	The description of the market location that the market is actually located in
Credit,WIC,WICcash,SFMNP ,SNAP	binary	Payment	Payment Accepted
Organic	binary	Product	Is that organic of the products
Bakedgoods,Cheese,Crafts, Flowers,Eggs,Seafood,Herbs ,Vegetables,Honey,Jams,Maple,Meat,Nursery,Nuts,Plants ,Poultry,Prepared,Soap,Tree s,Wine,Coffee,Beans,Fruits, Grains,Juices,Mushrooms,PetFood,Tofu,WildHarvested	binary	Product	Product available
updateTime	time	System	Update time

## 2.2 Quality issues

### **Problem 1: Too many missing data**

It's obvious that lots of information is missing in the dataset. For example, few markets have Youtube links, or content of Season 2 to 4.

For future data analysis, those columns are not necessary and we can remove the columns with too many missing data in our cleaning.

### **Problem 2: Bad formatting for some attributes**

Season dates are not in the same format. They mostly follow DD/MM/YYYY format, but some are in Month DD, YYYY format. We can also notice that the data is not well formatted. In the "updateTime" column, some instances have a specific date and time of the last update, but others only have the year.

We should make the time expression into the same format to make the content recognizable by script languages.

### **Problem 3: Inconsistent name or values**

In the "MarketName" column, the same market may have multiple formats of name. For example, the three expressions represent one market: Rochester Downtown Farmers Market, Downtown Rochester Farmers Market, Downtown Rochester Farmers' Market. Clustering the market names will solve the quality issue.

## 2.3 Use cases

The data file is pretty informative and well organized overall, although there are some unformed cells and missing values. With the raw data, several use cases can be satisfied:

- List available products for each market.
- Given FMID, find the corresponding market and its information.
- How many payment methods are available for each market.

For further data analysis using script languages, the raw data is not enough. Lots of tasks can only be performed after data cleaning:

- Locate each Farmers Market on one map using longitude and latitude, which should be well formatted before importing to map.
- Sort Market instances from the oldest to latest based on updateTime. Every date data should be in a standard format.
- List the duration of season1 of markets by start date and end date. The start and end date need to be extracted from one column in raw data.

### 3. Data Cleaning Process and Details

OpenRefine, (formerly called Google Refine), is one of standalone open source applications which is used for data cleaning, transformation and data wrangling. It is similar to spreadsheet applications (and can work with spreadsheet file formats); however, it behaves more like a database.

In this project, we are going to use OpenRefine to understand, clean and transform the Farmers market datasets to a clean version by cleaning and clustering each of the attributes or columns.

First, let's load the data from local computer and create a project "Farmers\_Market\_Data", the data will be showing as the following tabular format:

The screenshot shows the OpenRefine interface with the following details:

- URL: 127.0.0.1:3333/project?project=1652760395673
- Project Name: OpenRefine Farmers\_Market\_Data
- Row Count: 8816 rows
- Columns: FMID, MarketName, Website, Facebook, Twitter, Youtube, OtherMedia
- Facet / Filter and Undo / Redo buttons are visible at the top left.
- A sidebar on the left titled "Using facets and filters" provides instructions on how to use facets and filters to select subsets of data.
- The main table displays 10 rows of data, with the 11th row partially visible.
- Each row contains the following data:

FMID	MarketName	Website	Facebook	Twitter	Youtube	OtherMedia
1. 1018261	Caledonia Farmers Market Brandywine - Danville	<a href="https://sites.google.com/site/caledoniafarmersmarket/">https://sites.google.com/site/caledoniafarmersmarket/</a>	<a href="https://www.facebook.com/Danville.VT.Farmers.Market/">https://www.facebook.com/Danville.VT.Farmers.Market/</a>			
2. 1018318	Stearns Homestead Farmers' Market	<a href="http://www.StearnsHomestead.com">http://www.StearnsHomestead.com</a>	StearnsHomesteadFarmersMarket			
3. 1009364	105 S Main Street Farmers Market	<a href="http://thetownofsmile.wordpress.com/">http://thetownofsmile.wordpress.com/</a>				
4. 1010691	10th Street Community Farmers Market					<a href="http://agrimissouri.com/mo-grow-type=mo-grow&amp;ID=275">http://agrimissouri.com/mo-grow-type=mo-grow&amp;ID=275</a>
5. 1002454	112st Madison Avenue					
6. 1011100	12 South Farmers Market	<a href="http://www.12southfarmersmarket.com">http://www.12southfarmersmarket.com</a>	12_South_Farmers_Market	@12southfmstmk		@12southfmstmk
7. 1009845	125th Street Fresh Connect Farmers' Market	<a href="http://www.125thStreetFarmersMarket.com">http://www.125thStreetFarmersMarket.com</a>	<a href="https://www.facebook.com/125thStreetFarmersMarket">https://www.facebook.com/125thStreetFarmersMarket</a>	<a href="https://twitter.com/FarmMarket125th">https://twitter.com/FarmMarket125th</a>		Instagram -> 125thStreetFarm
8. 1005586	12th & Brandywine Urban Farm Market		<a href="https://www.facebook.com/pages/12th-Brandywine-Urban-Farm-Community-Garden/253769446091860">https://www.facebook.com/pages/12th-Brandywine-Urban-Farm-Community-Garden/253769446091860</a>			<a href="https://www.facebook.com/del">https://www.facebook.com/del</a>
9. 1008071	14&U Farmers' Market		<a href="https://www.facebook.com/14UfarmersMarket">https://www.facebook.com/14UfarmersMarket</a>	<a href="https://twitter.com/14UfarmersMkt">https://twitter.com/14UfarmersMkt</a>		
10. 1012710	14th & Kennedy Street Farmers Market		<a href="https://www.facebook.com/14KennedyFarmersMarket">https://www.facebook.com/14KennedyFarmersMarket</a>	14KenFM		instagram:14kenfm

#### 3.1 Identification attributes

##### MarketName

- Trim leading and trailing whitespace

8816 rows							Extensions: Wikidata ▾
Show as: <a href="#">rows</a> <a href="#">records</a>			Show: 5 10 25 50 rows		« first < previous <b>1 - 10</b> next > last »		
▼ All	▼ FMD	▼ MarketName	▼ Website	▼ Facebook	▼ Twitter	▼ Youtube	▼ OtherMedia
1. 1018261	Facet	▶ /sites.google.com/site/aledoniafarmersmarket/	https://www.facebook.com/Danville VT Farmers Market/				
	Text filter						
2. 1018318	Edit cells	▶ Transform...	StearnsHomesteadFarmersMarket				
	Edit column	▶ Common transforms	Trim leading and trailing whitespace				
3. 1009364	Transpose	▶ Fill down	Collapse consecutive whitespace				
	Sort...	Blank down	Unescape HTML entities				
4. 1010691	View	▶ Split multi-valued cells...	Replace Smart quotes with ascii				http://agrimissouri.com/mo-grow-type=mo-grown&ID=275
	Reconcile	▶ Join multi-valued cells...	To titlecase				
5. 1002454			To uppercase				
6. 1011100	12 South Farmers Market	http://Cluster and edit...	To lowercase	@12southfrmmskt			@12southfrmmskt
			To number				
7. 1009845	125th Street Fresh Connect Farmers' Market	http://www.125thStreetFarmersMarket.co	To date				Instagram--> 125thStreetFarm
			To text				
8. 1005586	12th & Brandywine Urban Farm Market		To null				https://www.facebook.com/del
			To empty string				
9. 1008071	14&U Farmers' Market			https://www.facebook.com/14UFarmersMarket	https://twitter.com/14UFarmersMkt		
10. 1012710	14th & Kennedy Street Farmers Market			https://www.facebook.com/14KennedyFarmersMarket/	14KenFM		instagram:14kenfm

- Collapse consecutive whitespace

8816 rows					Extensions: Wikidata		
Show as: rows records		Show: 5 10 25 50 rows	« first < previous 1 - 10 next > last »				
▼ All	▼ FMID	MarketName	Website	Facebook	Twitter	Youtube	OtherMedia
1.	1018261	Facet	► /sites.google.com/site/aledoniafarmersmarket/	https://www.facebook.com/Danville.VT.Farmers.Market/			
		Text filter					
2.	1018318	Edit cells	► Transform...	StearnsHomesteadFarmersMarket			
		Edit column	► Common transforms	Trim leading and trailing whitespace			
3.	1009364	Transpose	► Fill down	Collapse consecutive whitespace			
4.	1010691	Sort...	Blank down				
		View	► Split multi-valued cells...	Unescape HTML entities			http://agrimissouri.com/mo-grow-type-mo-grown&ID=275
5.	1002454	Reconcile	► Join multi-valued cells...	Replace Smart quotes with ascii			
6.	1011100	12 South Farmers Market	► Cluster and edit...	To titlecase			
			► Replace	To uppercase	@12southfrmsmkt		@12southfrmsmkt
7.	1009845	125th Street Fresh Connect Farmers' Market	► http://www.125thStreetFarmersMarket.co...	To lowercase			
				To number	@125thStreetFarmersMarket	https://twitter.com/FarmMarket125th	Instagram-> 125thStreetFarmersMarket
8.	1005586	12th & Brandywine Urban Farm Market		To date			
				To text			
9.	1008071	14&U Farmers' Market		To null	b-Brandywine-769448091860		
				To empty string			https://www.facebook.com/del...
10.	1012710	14th & Kennedy Street Farmers Market					
					https://www.facebook.com/14KennedyFarmersMarket/	https://twitter.com/14UFarmersMkt	
						14KenFM	instagram:14kenfm

- Use text facet and cluster by using **key collision** method and **fingerprint** keying function.

8816 rows      Extensions: Wikidata ▾

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

All	FMID	MarketName	Website	Facebook	Twitter	Youtube	OtherMedia
1.	1018261	Caledonia Farmers Market Association -	<a href="https://sites.google.com/site/caledoniamarkets/">https://sites.google.com/site/caledoniamarkets/</a>	<a href="https://www.facebook.com/Danville.VT.Farmers.Market/">https://www.facebook.com/Danville.VT.Farmers.Market/</a>			

**Cluster & Edit column "MarketName"**

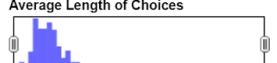
This feature helps you find groups of different cell values that might be alternative representations of the same thing. For example, the two strings "New York" and "new york" are very likely to refer to the same concept and just have capitalization differences, and "Gödel" and "Godel" probably refer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

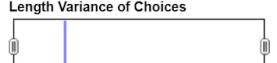
Method key collision      Keying Function fingerprint      230 clusters found

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 type="checkbox"/> Main Street Farmers 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 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 type="checkbox"/> Rochester Downtown Farmers Market
3	3	<ul style="list-style-type: none"> <li>Harrison Farmer's Market (1 rows)</li> <li>Harrison Farmers Market (1 rows)</li> <li>Harrison Farmers' Market (1 rows)</li> </ul>	<input type="checkbox"/> Harrison Farmer's Market
3	4	<ul style="list-style-type: none"> <li>Goshen Farmers Market (2 rows)</li> <li>Goshen Farmer's Market (1 rows)</li> <li>Goshen Farmers' Market (1 rows)</li> </ul>	<input type="checkbox"/> Goshen Farmers Market
2	4	<ul style="list-style-type: none"> <li>Irvington Farmers Market (2 rows)</li> </ul>	<input type="checkbox"/> Irvington Farmers Market

# Choices in Cluster  
  
2 — 4

# Rows in Cluster  
  
2 — 34

Average Length of Choices  
  
18 — 71

Length Variance of Choices  
  
0 — 2.5

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

We found one problem that the farmer market name has different ways: Farmer's, Farmers, Farmers' . And it's hard to let the system make auto clustering to choose the consistent names.

- Harrison Farmer's Market** (1 rows)
  - Harrison Farmers Market** (1 rows)
  - Harrison Farmers' Market** (1 rows)
- 
- Goshen Farmers Market** (2 rows)
  - Goshen Farmer's Market** (1 rows)
  - Goshen Farmers' Market** (1 rows)
- 
- Irvington Farmers Market** (2 rows)
  - Irvington Farmer's Market** (1 rows)
  - Irvington Farmers' Market** (1 rows)
- 
- Northfield Farmers' Market** (2 rows)
  - Northfield Farmer's Market** (1 rows)

- Custom normalization by using replace characters

In order to have consistent names, we plan to apply our custom normalizations first before applying the clustering.

Custom text transform on column MarketName

Expression: value.replace("Farmers'","Farmers").replace("Farmer's","Farmers")

Language: General Refine Expression Language (GREL) ▾ No syntax error.

Preview	History	Starred	Help
3. 100 S. Main Street Farmers Market	100 S. Main Street Farmers Market		
4. 10th Street Community Farmers Market	10th Street Community Farmers Market		
5. 112st Madison Avenue	112st Madison Avenue		
6. 12 South Farmers Market	12 South Farmers Market		
7. 125th Street Fresh Connect Farmers' Market	125th Street Fresh Connect Farmers Market		
8. 12th & Brandywine Urban Farm Market	12th & Brandywine Urban Farm Market		
9. 14&U Farmers' Market	14&U Farmers Market		
10. 14th & Kennedy Street Farmers Market	14th & Kennedy Street Farmers Market		

On error:  keep original  set to blank  store error

Re-transform up to  10 times until no change

OK Cancel

- Now, we do the Cluster again by using **key collision** method and **fingerprint** keying function.
- This time, we see less problems, and it's easier to manually review all the clusters and choose the consistent names.

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 52 clusters found

Cluster Size	Row Count	Values in Cluster	Merge?	New Cell Value
2	3	• Oxford Farmers Market (2 rows) • OXFORD FARMERS MARKET (1 rows)	<input checked="" type="checkbox"/>	Oxford Farmers Market
2	2	• Dearborn Farmers & Artisans Market (1 rows) • Farmers & Artisans Market Dearborn (1 rows)	<input checked="" type="checkbox"/>	Dearborn Farmers & Artisans Mar
2	5	• Midtown Farmers Market (4 rows) • Mid-Town Farmers Market (1 rows)	<input checked="" type="checkbox"/>	Midtown Farmers Market
2	2	• Athens Farmers Market, L.L.C. (1 rows) • Athens Farmers Market, LLC (1 rows)	<input checked="" type="checkbox"/>	Athens Farmers Market, L.L.C.
2	13	• Main Street Farmers Market (12 rows) • MAIN STREET FARMERS MARKET (1 rows)	<input checked="" type="checkbox"/>	Main Street Farmers Market
2	2	• Morgan County Farmers Market Association (1 rows) • Morgan County farmers' Market Association (1 rows)	<input checked="" type="checkbox"/>	Morgan County Farmers Market A
2	4	• Columbus Farmers Market (3 rows)	<input checked="" type="checkbox"/>	Columbus Farmers Market

# Rows in Cluster: 2 — 34

Average Length of Choices: 18 — 70

Length Variance of Choices: 0 — 2.5

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

- Use text facet and cluster again by using key collision method and ngram-fingerprint, except those have distinct different names.

## 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  Keying Function  Ngram Size  16 clusters found

Cluster Size	Row Count	Values in Cluster	Merge?	New Cell Value
2	2	• Nashville F.A.R.M. II (1 rows) • Nashville F.A.R.M. III (1 rows)	<input type="checkbox"/>	Nashville F.A.R.M. II
2	3	• Northport Farmers Market (2 rows) • North Port Farmers Market (1 rows)	<input checked="" type="checkbox"/>	Northport Farmers Market
2	2	• El Dorado Farmers Market (1 rows) • Eldorado Farmers Market (1 rows)	<input checked="" type="checkbox"/>	El Dorado Farmers Market
2	5	• Eastside Farmers Market (3 rows) • East Side Farmers Market (2 rows)	<input checked="" type="checkbox"/>	Eastside Farmers Market
2	2	• Old Town Farmers Market (1 rows) • Old-Town Farmers Market (1 rows)	<input checked="" type="checkbox"/>	Old Town Farmers Market
2	2	• St. John Farmers Market (1 rows) • St.John Farmers Market (1 rows)	<input checked="" type="checkbox"/>	St. John Farmers Market
2	2	• East Point Farmers Market (1 rows) • Eastpoint Farmers Market (1 rows)	<input checked="" type="checkbox"/>	East Point Farmers Market

# Rows in Cluster  
  
2 — 5

Average Length of Choices  
  
21 — 42

Length Variance of Choices  
  
0 — 1.5

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

## 3.2 Information attributes

### Website|Facebook|Twitter|Youtube|OtherMedia

- Trim leading and trailing whitespace
- By using GREL, change 'n/a' or 'none' values to blank.

Custom text transform on column Facebook

Expression `value.replace(/[\n\r]/g, '')` Language  No syntax error.

Preview History Starred Help

row	value	value.replace(/[\n\r]/g, '')
1.	<a href="https://www.facebook.com/Danville.VT.Farmers.Market/">https://www.facebook.com/Danville.VT.Farmers.Market/</a>	<a href="https://www.facebook.com/Danville.VT.Farmers.Market/">https://www.facebook.com/Danville.VT.Farmers.Market/</a>
2.	StearnsHomesteadFarmersMarket	StearnsHomesteadFarmersMarket
3.	null	Error: replace expects 3 strings, or 1 string, 1 regex, and 1 string
4.	null	Error: replace expects 3 strings, or 1 string, 1 regex, and 1 string
5.	null	Error: replace expects 3 strings, or 1 string, 1 regex, and 1 string
6.	12_South_Farmers_Market	12_South_Farmers_Market
7.	<a href="https://www.facebook.com/125thStreetFarmersMarket">https://www.facebook.com/125thStreetFarmersMarket</a>	<a href="https://www.facebook.com/125thStreetFarmersMarket">https://www.facebook.com/125thStreetFarmersMarket</a>

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

OK Cancel

## Facebook

Specially for Facebook, we also want to keep the url consistent as we found a lot of the values have the alias without facebook.com prefix. We plan to update that to keep data consistent.

The screenshot shows a data processing interface with a table containing 8816 rows. The columns are FMID, MarketName, Website, Facebook, Twitter, YouTube, and OtherMedia. Two rows are visible:

FMID	MarketName	Website	Facebook	Twitter	YouTube	OtherMedia
1. 1018261	Caledonia Farmers Market Association - Danville	https://sites.google.com/site/caledoniafarmersmarket/	https://www.facebook.com/Danville.VT.Farmers.Market/			
2. 1018318	Stearns Homestead Farmers Market	http://www.StearnsHomestead.com	StearnsHomesteadFarmersMarket			

A custom text transform dialog is open for the Facebook column. The expression is:

```
value = value.replace("http://", "https://")
if(value.startswith("https://")): return value
else: return "https://www.facebook.com/" + value
```

The preview shows the transformed values:

Original	Transformed
5. null	Error: Traceback (most recent call last): File "<string>", line 2, in <temp_450080237> AttributeError: 'NoneType' object has no attribute 'replace'
6. 12_South_Farmers_Market	https://www.facebook.com/12_South_Farmers_Market
7. https://www.facebook.com/125thStreetFarmersMarket	https://www.facebook.com/125thStreetFarmersMarket
8. https://www.facebook.com/pages/12th-Brandywine-Urban-Farm-Community-Garden/253769448091860	https://www.facebook.com/pages/12th-Brandywine-Urban-Farm-Community-Garden/253769448091860
9. https://www.facebook.com/14UFarmersMarket	https://www.facebook.com/14UFarmersMarket
10. https://www.facebook.com/14KennedyFarmersMarket/	https://www.facebook.com/14KennedyFarmersMarket/

On error options:  keep original,  set to blank,  store error.

OK Cancel

## Twitter

Similarly for Twitter, we also need to normalize and unify the format. We found 4 patterns in the Twitter data: @XXX, #XXX, XXX, <https://twitter.com/XXX>. We applied the python code to unify the formats to <https://twitter.com/XXX>.

The screenshot shows a data processing interface with a table containing 8816 rows and a custom text transform dialog for the Twitter column. The expression is:

```
value = value.replace("http://", "https://")
if(value.startswith("@")): return "https://twitter.com/" + value[1:]
elif(value.startswith("#")): return "https://twitter.com/" + value[1:]
elif(value.startswith("http://")): return value
else: return "https://twitter.com/" + value
```

The preview shows the transformed values:

Original	Transformed
6. @12southfrmsmkt	https://twitter.com/12southfrmsmkt
7. https://twitter.com/FarmMarket125th	https://twitter.com/https://twitter.com/FarmMarket125th
8. null	Error: Traceback (most recent call last): File "<string>", line 2, in <temp_651374877> AttributeError: 'NoneType' object has no attribute 'replace'
9. https://twitter.com/14UFarmersMkt	https://twitter.com/https://twitter.com/14UFarmersMkt
10. 14KenFM	https://twitter.com/14KenFM

On error options:  keep original,  set to blank,  store error.

OK Cancel

### 3.3 Location attributes

#### Street|City|State

- Trim leading and trailing white space.
- Collapse consecutive white spaces.
- To Title Case.

#### City:

- Use text facet and cluster by using key collision method and fingerprint keying function.

**Cluster & Edit column "city"**

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 10 clusters found

Cluster Size	Row Count	Values in Cluster	Merge?	New Cell Value
2	10	• St. Louis (8 rows) • St Louis (2 rows)	<input checked="" type="checkbox"/>	St. Louis
2	2	• St Augustine (1 rows) • St. Augustine (1 rows)	<input checked="" type="checkbox"/>	St. Augustine
2	2	• Land O Lakes (1 rows) • Land O' Lakes (1 rows)	<input checked="" type="checkbox"/>	Land O' Lakes
2	2	• Wheeling (1 rows) • Wheeling, (1 rows)	<input checked="" type="checkbox"/>	Wheeling
2	16	• Indianapolis (15 rows) • Indianapolis, (1 rows)	<input checked="" type="checkbox"/>	Indianapolis
2	5	• Greenwood (4 rows) • Greenwood Greenwood (1 rows)	<input checked="" type="checkbox"/>	Greenwood
2	2	• Mt Airy (1 rows) • Mt. Airy (1 rows)	<input checked="" type="checkbox"/>	Mt. Airy

# Rows in Cluster  
2 — 16

Average Length of Choices  
5.5 — 14

Length Variance of Choices  
0 — 5

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

- Use text facet and cluster again by using key collision method and ngram-fingerprint, except those have distinct different names.

**Cluster & Edit column "city"**

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	17 clusters found
Cluster Size	Row Count	Values in Cluster	Merge?	New Cell Value	
2	3	• - (2 rows) • O (1 rows)	<input type="checkbox"/>		
2	2	• Le Roy (1 rows) • Leroy (1 rows)	<input checked="" type="checkbox"/>	Le Roy	
2	5	• Northport (4 rows) • North Port (1 rows)	<input checked="" type="checkbox"/>	Northport	
2	4	• La Crosse (3 rows) • Lacrosse (1 rows)	<input checked="" type="checkbox"/>	La Crosse	
2	3	• La Grange (2 rows) • Lagrange (1 rows)	<input checked="" type="checkbox"/>	La Grange	
2	3	• Delmar (2 rows) • Del Mar (1 rows)	<input checked="" type="checkbox"/>	Delmar	
2	3	• East Hampton (2 rows) • Easthampton (1 rows)	<input checked="" type="checkbox"/>	East Hampton	

**# Rows in Cluster**  
  
 2 — 7

**Average Length of Choices**  
  
 1 — 13

**Length Variance of Choices**  
  
 0 — 0.5

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

## Zip

- Trim leading and trailing white space.
- Convert any string values to blank using GREL.
- [GREF] Unify the format. (United States zip code follows NNNNN or NNNNN-NNNN format)

**Custom text transform on column zip**

Expression  
`value.replace(/^\d{1,4}\$|^\d{5}-\d{4}\$/,"")`

Language General Refine Expression Language (GREL) No syntax error.

Preview History Starred Help

row	value	value.replace(/^\d{1,4}\\$ ^\d{5}-\d{4}\\$/,"")
1.	05828	05828
2.	null	Error: replace expects 3 strings, or 1 string, 1 regex, and 1 string
3.	29682	29682
4.	64759	64759
5.	10029	10029
6.	37204	37204
7.	10027	10027

On error  keep original  set to blank  store error

Re-transform up to 10 times until no change

OK Cancel

## x|y

- Trim leading and trailing white space.

- Convert the values to number

x	y	Location	Credit	WIC	WICcash	SFMNP	SNAP	
-72.140335	44.411037		Y	Y	N	Y	N	Y
-81.73394	41.374802		Y	N	N	Y	N	-
-82.818703	34.804199		Y	N	N	N	N	-
-94.27462	37.495628		Y	N	N	N	N	-

#### Location:

- Remove the column as not useful for our analysis and too sparse with no useful information.

## 3.4 Seasonality

The data provides the attributes Date and Time for Season 1, Season 2, Season 3, Season 4. However, The Season 2|3|4 values are pretty sparse with nearly all empty values there. So, we consider removing these attributes for our analysis.

Season1Date	Season1Time	Season2Time	Season3Date	Season3Time	Season4Date	Season4Time
06/14/2017 to 08/30/2017	Wed: 9:00 AM-1:00 PM;	Facet ▾				
		Text filter				
06/24/2017 to 09/30/2017	Sat: 9:00 AM-1:00 PM;	Edit cells ▾				
		Edit column ▾	Split into several columns...			
		Transpose ▾	Join columns...			
04/02/2014 to 11/30/2014	Wed: 3:00 PM-6:00 PM; Sat: 8:00 AM-1:00 PM;	Sort... ▾	Add column based on this column...			
July to November	Tue: 8:00 am - 5:00 pm; Sat: 8:00 am - 8:00 pm;	View ▾	Add column by fetching URLs...			
05/05/2015 to 10/27/2015	Tue: 3:30 PM-6:30 PM;	Reconcile ▾	Add columns from reconciled values...			
06/10/2014 to 11/25/2014	Tue: 10:00 AM-7:00 PM;		Rename this column			
			Remove this column			
			Move column to beginning			
			Move column to end			
			Move column left			

#### Season1Date

- Trim leading and trailing white space.
- Collapse consecutive white spaces.

- In order to analyze the date range of start and end for this season. We plan to split the data to 2 columns: Season1StartDate, Season1EndDate

8816 rows

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

	city	County	State	zip	Season1Date	Season1Time	x	y	Location	Credit	WIC	WICcash	SFMNP	Other
	Danville	Caledonia	Vermont	05828	06/14/2017 to 08/30/2017	Wed: 9:00 AM-1:00 PM;	-72.140335	44.411037		Y	Y	N	Y	N
ge	Parma	Cuyahoga	Ohio		06/24/2017 to 09/30/2017	Sat: 9:00 AM-1:00 PM;	-81.73394	41.374802		Y	N	N	Y	N
ain	Six Mile		South Carolina	29682			-82.818703	34.804199		Y	N	N	N	N
et	Lamar	Barton								N	N	N	N	N
ar	New York	New York								N	Y	Y	N	
te	Nashville	Davidson								N	N	N	Y	
et	New York	New York								Y	N	Y	Y	
en	Wilmington	New Castle								N	N	N	Y	
ne	Washington	District Of								Y	Y	Y	Y	

**Split column Season1Date into several columns**

**How to Split Column**

by separator  
 Separator   regular expression

Remove this column

by field lengths  
  
List of integers separated by commas, e.g., 5, 7, 15

**After Splitting**

Guess cell type  
 Remove this column

**OK** **Cancel**

For the new splitted columns:

- Trim leading and trailing white space.
- Apply the transform to date
- Rename the column to Season1StartDate, Season1EndDate

	County	State	zip	Season1StartDate	Season1EndDate	Season1Time	x	y	Location	Other
	Caledonia	Vermont	05828	2017-06-14T00:00:00Z	2017-08-30T00:00:00Z	Wed: 9:00 AM-1:00 PM;	-72.140335	44.411037		Y
	Cuyahoga	Ohio		2017-06-24T00:00:00Z	2017-09-30T00:00:00Z	Sat: 9:00 AM-1:00 PM;	-81.73394	41.374802		Y
		South Carolina	29682				-82.818703	34.804199		Y
	Barton	Missouri	64759	2014-04-02T00:00:00Z	2014-11-30T00:00:00Z	Wed: 3:00 PM-6:00 PM; Sat: 8:00 AM-1:00 PM;	-94.27462	37.495628		Y
rk	New York	New York	10029	July	November	Tue: 8:00 am - 5:00 pm; Sat: 8:00 am - 8:00 pm;	-73.949303	40.7939	Private business parking lot	N
le	Davidson	Tennessee	37204	2015-05-05T00:00:00Z	2015-10-27T00:00:00Z	Tue: 3:30 PM-6:30 PM;	-86.79071	36.11837		Y
rk	New York	New York	10027	2014-06-10T00:00:00Z	2014-11-25T00:00:00Z	Tue: 10:00 AM-7:00 PM;	-73.94825	40.808952	Federal/State government building grounds	Y

After converting the Season1StartDate and Season1EndDate to date format, we found that formats like "July" and "November" are not convertible, and the year is missing in those cells.

**Facet / Filter** Undo / Redo 52 / 58

Refresh Reset All Remove All

**Season1StartDate** change reset

2010-03-31 17:00:00 — 2020-11-06 16:00:00

Time 4976 Non-Time 734 Blank 3106 Error 0

**Season1StartDate** change

14 choices Sort by: name count Cluster

- April 51
- August 5
- December 4
- February 1
- January 111
- July 52
- June 238
- June 23 1
- March 5
- May 246
- November 12
- Oct 4

**734 matching rows (8816 total)**

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

city	County	State	zip	Season1StartDa	Season1EndDa	Season1Time
New York	New York	New York	10029	July	November	Tue:8:00 am - 5:00 pm;Sat:8:00 am - 8:00 pm;
Minneapolis	Hennepin	Minnesota	55413	June	September	Wed:3:00 pm - 7:00 pm;
Cedar Rapids	Linn	Iowa	52401	May	October	Tue:4:00 PM - 6:00 PM;Sat:7:30 AM - 12:00 PM;
Abbotsford	Clark	Wisconsin	54405	May	October	Tue:1:30 PM - 5:30 PM;
Albany	Dougherty	Georgia	31702	May	December	Sat:8:00 AM - 2:00 PM;
Harrisville	Alcona	Michigan	48740	June	September	Sat:10:00 am - 1:00 pm;
Altus	Jackson	Oklahoma	73521	June	October	Tue:5:00 pm - 8:00 pm;Fri:5:00 pm - 8:00 pm;
Amery	Polk	Wisconsin	54001	June	October	Mon:3:00 pm - 6:00 pm;
Anchorage	Anchorage	Alaska	99501	May	September	Sat:10:00 am - 6:00 pm;sun:10:00 am - 6:00 pm;

For those cells that are not convertible to date, we assume the year is the same as the year of corresponding updateTime.

For each cell:

- Convert the month name to numeric format.
- Join the updateTime column and only keep the year.
- Convert the column to date.

**Facet / Filter** Undo / Redo 69 / 70

Extract... Apply...

Filter: Season1StartDate

59. Mass edit 238 cells in column Season1StartDate

60. Mass edit 1 cells in column Season1StartDate

61. Mass edit 5 cells in column Season1StartDate

62. Mass edit 246 cells in column Season1StartDate

63. Mass edit 12 cells in column Season1StartDate

64. Mass edit 4 cells in column Season1StartDate

65. Mass edit 3 cells in column Season1StartDate

66. Mass edit 1 cells in column Season1StartDate

67. Text transform on 1 cells in column Season1StartDate: value.toDate()

68. Text transform on 733 cells in column Season1StartDate: gret:value.split(",") [1].split(".").[0]+""+value.split(",")[0]

69. Text transform on 733 cells in column Season1StartDate: gret:value.split(",") [1].split(".").[0]+""+value.split(",")[0]

**733 matching rows (8816 total)**

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

city	County	State	zip	Season1StartDa	Season1EndDa	Season1Time	x	y	Credit	WIC	WI
New York	New York	New York	10029	Facet	Member	Tue:8:00 am - 5:00 pm;Sat:8:00 am - 8:00 pm;	-73.949303	40.7939	N	N	Y
Minneapolis	Hennepin	Minnesota	55413	Text filter			-93.259102	45.004398	N	Y	Y
Cedar Rapids	Linn	Iowa	52401	Edit cells	Transform...						
Abbotsford	Clark	Wisconsin	54405	Edit column	Common transforms	Trim leading and trailing whitespace					
Albany	Dougherty	Georgia	31702	Transpose		Collapse consecutive whitespace					
Harrisville	Alcona	Michigan	48740	Sort...		Unescape HTML entities					
Altus	Jackson	Oklahoma	73521	View		Replace Smart quotes with ascii					
Amery	Polk	Wisconsin	54001	Reconcile		To titlecase					
Anchorage	Anchorage	Alaska	99501			To uppercase					
Angels Camp	Calaveras	California	95222			To lowercase					
						To number					
						To date					
						To text					
						To null					
						To empty string					

Then we can follow the same process for the Season1EndDate column.

### Season1Time

- Trim leading and trailing white space.
- Collapse consecutive white spaces.
- The data contains one or multiple times, each with the weekday, start time and end time. In our analysis, we plan to keep the raw format. If any scenario is needed, we may need to process this attribute for better analysis.

## 3.5 Payments & Products

### Organic:

Better to replace “-” to empty

row	value	value.replace("-", "")
1.	Y	Y
2.	-	
3.	-	
4.	-	
5.	-	
6.	Y	Y
7.	Y	Y

On error:  keep original  set to blank  store error

Re-transform up to 10 times until no change

OK Cancel

All the rest of values contain binary or boolean values. No need to clean these fields.

After the data cleaning, we can use the export feature to export projects and data results into various formats.

### 3.6 UpdateTime

The simplest way to format the updateTime column is convert values to date format directly.

red	<input type="checkbox"/> Soap	<input type="checkbox"/> Trees	<input type="checkbox"/> Wine	<input type="checkbox"/> Coffee	<input type="checkbox"/> Beans	<input type="checkbox"/> Fruits	<input type="checkbox"/> Grains	<input type="checkbox"/> Juices	<input type="checkbox"/> Mushrooms	<input type="checkbox"/> PetFood	<input type="checkbox"/> Tofu	<input type="checkbox"/> WildHarvested	<input type="checkbox"/> updateTime	
	Y	Y	N	Y	Y	Y	N	N	Y	Y	N	Facet Text filter	0/2017 43:57 PM	
	Y	N	N	N	N	Y	N	N	N	Transform...			Edit cells Edit column Transpose Sort... View Reconcile	1/2017 5:01 PM 3 28/2014 9:46 AM 10/2012 10:38:22 AM
						Trim leading and trailing whitespace Collapse consecutive whitespace				Fill down Blank down				
	Y	N	N	N	N	Unescape HTML entities Replace Smart quotes with ascii				Split multi-valued cells... Join multi-valued cells... Cluster and edit...				
	Y	N	N	N	N	To titlecase To uppercase To lowercase				Replace				
	Y	N	N	Y	N	To number To date To text				N	N	N	4/7/2014 4:32:01 PM	
	Y	N	Y	Y	N	To null To empty string				N	N	N	4/3/2014 3:43:31 PM	
	N	N	N	N	N					N	N	N	4/5/2014 1:49:04 PM	
	N	N	N	N	Y	Y	Y	N		N	N	N	7/20/2016 11:16:24 AM	
	N	N	N	Y	N	Y	N	N		Y	N	N		

But there are several cells that OpenRefine is unable to convert. The problem is that cells with 12PM are not convertible.

For the unconvertible cells:

- Remove PM appendix.

- Trim leading and trailing white space.
- Convert to date format.

## 4. Results

In this section, we will firstly export SQLite database from OpenRefine, then do integrity constraint analysis with SQLite scripts.

### 4.1 Relational Database Schema

OpenRefine is able to specify the SQL Type of each column before exporting the sql file.

The screenshot shows the 'SQL Exporter' interface in OpenRefine. It displays a table of columns and their corresponding SQL types. The columns listed are city, County, State, zip, Season1StartDate, Season1EndDate, Season1Time, and x. The types assigned are VARCHAR, VARCHAR, VARCHAR, INT, DATE, DATE, VARCHAR, and VARCHAR respectively. Each column has an 'Apply All' checkbox checked. At the bottom, there are buttons for 'Select All', 'De-select All', and checkboxes for 'Output empty row (i.e. all cells null)', 'Ignore facets and filters and export all rows', and 'Trim Column Names'.

Column	Type	Action
city	VARCHAR	Apply All
County	VARCHAR	Apply All
State	VARCHAR	Apply All
zip	INT	Apply All
Season1StartDate	DATE	Apply All
Season1EndDate	DATE	Apply All
Season1Time	VARCHAR	Apply All
x	VARCHAR	Apply All

The FMID and zip column is INT format. Season1StartDate, Season1EndDate, are updateTime are DATE format. x and y are NUMERIC format. Others are VARCHAR format.

The following is the schema generated by SQL Exporter of OpenRefine:

```
CREATE TABLE Farmers_Market_Data (
    FMID INT NULL,
    MarketName VARCHAR(255) NULL,
    Website VARCHAR(255) NULL,
    Facebook VARCHAR(255) NULL,
    Twitter VARCHAR(255) NULL,
    Youtube VARCHAR(255) NULL,
    OtherMedia VARCHAR(255) NULL,
    street VARCHAR(255) NULL,
    city VARCHAR(255) NULL,
    County VARCHAR(255) NULL,
```

```
State VARCHAR(255) NULL,
zip INT NULL,
Season1StartDate DATE NULL,
Season1EndDate DATE NULL,
Season1Time VARCHAR(255) NULL,
x NUMERIC NULL,
y NUMERIC NULL,
Credit VARCHAR(255) NULL,
WIC VARCHAR(255) NULL,
WICcash VARCHAR(255) NULL,
SFMNP VARCHAR(255) NULL,
SNAP VARCHAR(255) NULL,
Organic VARCHAR(255) NULL,
Bakedgoods VARCHAR(255) NULL,
Cheese VARCHAR(255) NULL,
Crafts VARCHAR(255) NULL,
Flowers VARCHAR(255) NULL,
Eggs VARCHAR(255) NULL,
Seafood VARCHAR(255) NULL,
Herbs VARCHAR(255) NULL,
Vegetables VARCHAR(255) NULL,
Honey VARCHAR(255) NULL,
Jams VARCHAR(255) NULL,
Maple VARCHAR(255) NULL,
Meat VARCHAR(255) NULL,
Nursery VARCHAR(255) NULL,
Nuts VARCHAR(255) NULL,
Plants VARCHAR(255) NULL,
Poultry VARCHAR(255) NULL,
Prepared VARCHAR(255) NULL,
Soap VARCHAR(255) NULL,
Trees VARCHAR(255) NULL,
Wine VARCHAR(255) NULL,
Coffee VARCHAR(255) NULL,
Beans VARCHAR(255) NULL,
Fruits VARCHAR(255) NULL,
Grains VARCHAR(255) NULL,
Juices VARCHAR(255) NULL,
Mushrooms VARCHAR(255) NULL,
PetFood VARCHAR(255) NULL,
Tofu VARCHAR(255) NULL,
WildHarvested VARCHAR(255) NULL,
updateTime DATE NULL
);
```

We can then create the database from this schema through command:

```
sqlite3 Farmers_Market_Data.db < Farmers_Market_Data.sql
```

## 4.2 Integrity Constraints

There are several integrity problems that should be checked. After this section, we will fix some logical problems in this dataset.

### FMID

Since FMID is the identifier of each farmers market, each FMID should be unique in the dataset. We can output duplicates through the command:

```
SELECT FMID
  FROM Farmers_Market_Data
 GROUP BY FMID
 HAVING COUNT(FMID)>1;
```

The output is empty, so there are no duplicates in FMID.

Then we can check whether there are any null FMID:

```
SELECT COUNT(FMID)
  FROM Farmers_Market_Data
 WHERE FMID IS NULL;
```

The output is 0. So there is no null FMID.

### OtherMedia

The column of OtherMedia is the supplement of the columns of Website, Facebook, Twitter, and Youtube. If content in OtherMedia is the same as those in one of the four columns, it is considered as redundant.

We can check whether data in OtherMedia is same as any other or not:

```
SELECT "Website == Othermedia cases ", COUNT(OtherMedia)
  FROM Farmers_Market_Data
 WHERE Website = OtherMedia AND OtherMedia!="";
SELECT "Facebook == Othermedia cases ", COUNT(OtherMedia)
  FROM Farmers_Market_Data
```

```

WHERE Facebook = OtherMedia AND OtherMedia!="";
SELECT "Twitter == Othermedia cases ", COUNT(OtherMedia)
FROM Farmers_Market_Data
WHERE Twitter = OtherMedia AND OtherMedia!="";
SELECT "Youtube == Othermedia cases ", COUNT(OtherMedia)
FROM Farmers_Market_Data
WHERE Youtube = OtherMedia AND OtherMedia!="";

```

The output is:

Website == Othermedia cases	1
Facebook == Othermedia cases	1
Twitter == Othermedia cases	0
Youtube == Othermedia cases	5

Then we can remove the redundant data:

```

UPDATE Farmers_Market_Data
SET OtherMedia = ""
WHERE FMID IN
(SELECT FMID
FROM Farmers_Market_Data
WHERE Website=OtherMedia AND OtherMedia!="");
UPDATE Farmers_Market_Data
SET OtherMedia = ""
WHERE FMID IN
(SELECT FMID
FROM Farmers_Market_Data
WHERE Facebook=OtherMedia AND OtherMedia!="");
UPDATE Farmers_Market_Data
SET OtherMedia = ""
WHERE FMID IN
(SELECT FMID
FROM Farmers_Market_Data
WHERE Twitter=OtherMedia AND OtherMedia!="");
UPDATE Farmers_Market_Data
SET OtherMedia = ""
WHERE FMID IN
(SELECT FMID
FROM Farmers_Market_Data
WHERE Youtube=OtherMedia AND OtherMedia!="");

```

After these steps, duplicated content in OtherMedia is removed, and the updated database is saved in the .db file automatically.

## Season1StartDate, Season1EndDate

In previous cleaning section, we splitted Season1Date into Season1StartDate and Season1EndDate, and formatted them into standard date. We can now use SQLite to and output instances that the start is later than the end date in this database:

```
SELECT FMID, Season1StartDate, Season1EndDate
FROM Farmers_Market_Data
WHERE Season1StartDate>Season1EndDate
AND Season1EndDate!="";
```

The output:

1006039	2011-11-01T00:00:00Z	2011-04-01T00:00:00Z
1004929	2012-11-01T00:00:00Z	2012-03-01T01:00:00Z
1011959	2016-10-01T00:00:00Z	2016-05-07T00:00:00Z
1000865	2012-11-01T00:00:00Z	2012-03-01T01:00:00Z
1000108	2012-11-01T00:00:00Z	2012-03-01T01:00:00Z
1001963	2012-10-01T00:00:00Z	2012-03-01T01:00:00Z
1002273	2012-12-01T01:00:00Z	2012-05-01T00:00:00Z
1003898	2012-09-01T00:00:00Z	2012-04-01T00:00:00Z
1002777	2012-10-01T00:00:00Z	2012-05-01T00:00:00Z
1005022	2011-09-01T00:00:00Z	2011-05-01T00:00:00Z
1001511	2012-11-01T00:00:00Z	2012-04-01T00:00:00Z
1005379	2012-11-01T00:00:00Z	2012-04-01T00:00:00Z
1002027	2011-11-01T00:00:00Z	2011-05-01T00:00:00Z
1003932	2012-11-01T00:00:00Z	2012-05-01T00:00:00Z
1002724	2012-10-01T00:00:00Z	2012-05-01T00:00:00Z
1008935	2013-01-01T00:00:00Z	2012-12-31T00:00:00Z
1006056	2012-11-01T00:00:00Z	2012-04-01T00:00:00Z
1006508	2012-12-01T01:00:00Z	2012-04-01T00:00:00Z

To fix this integrity problem, we can swap the start date and the end date:

```
UPDATE Farmers_Market_Data
SET Season1StartDate=Season1EndDate, Season1EndDate=Season1StartDate
WHERE FMID IN
(SELECT FMID
FROM Farmers_Market_Data
WHERE Season1StartDate>Season1EndDate
AND Season1EndDate!="");
```

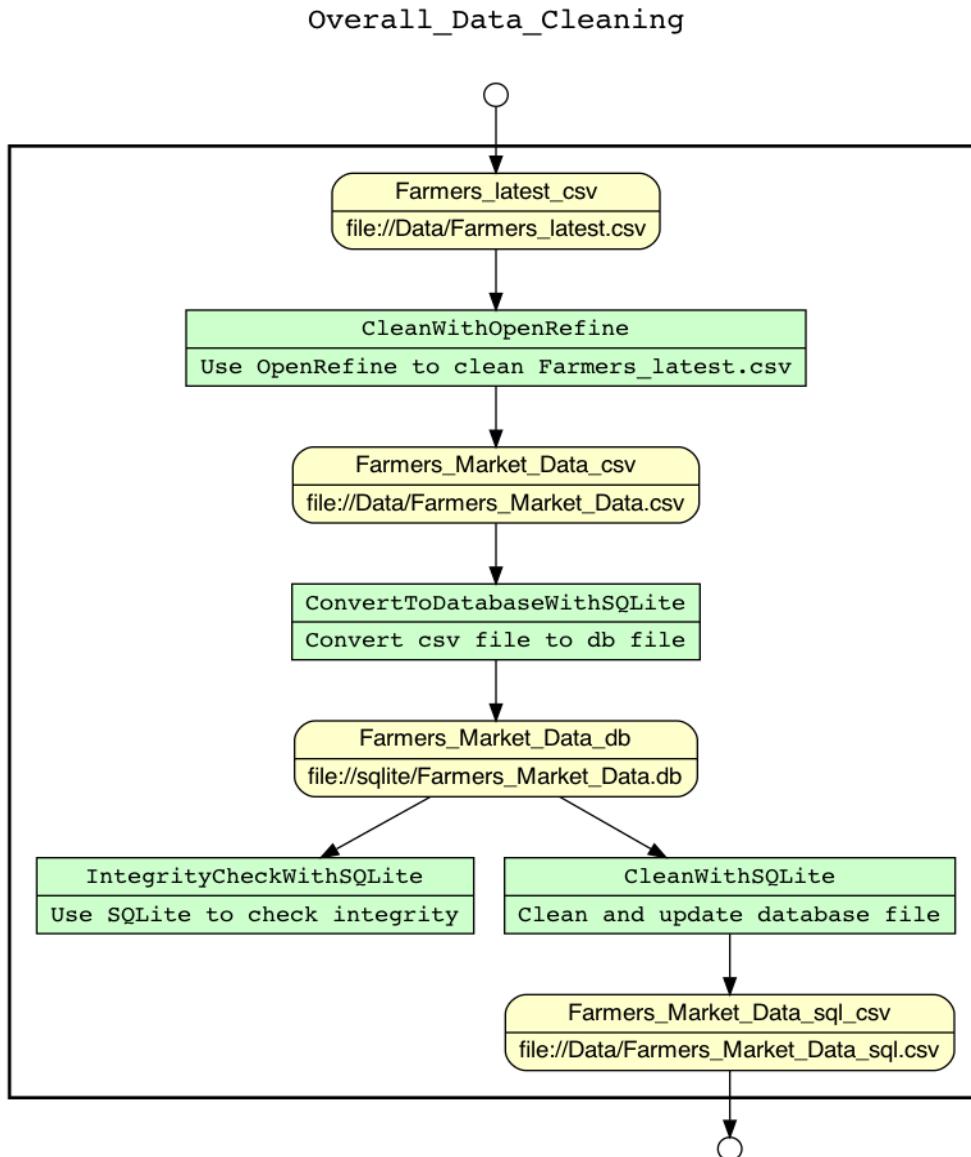
Then the Season1StartDate and Season1EndDate are reasonable in the database.

## 5. Workflow

YesWorkflow aims to provide a number of the benefits of using a scientific workflow management system without having to rewrite scripts and other scientific software. In this project, we are going to use Workflow to create and manage the scripts for data cleaning, and SQL scripts for data modeling and analysis.

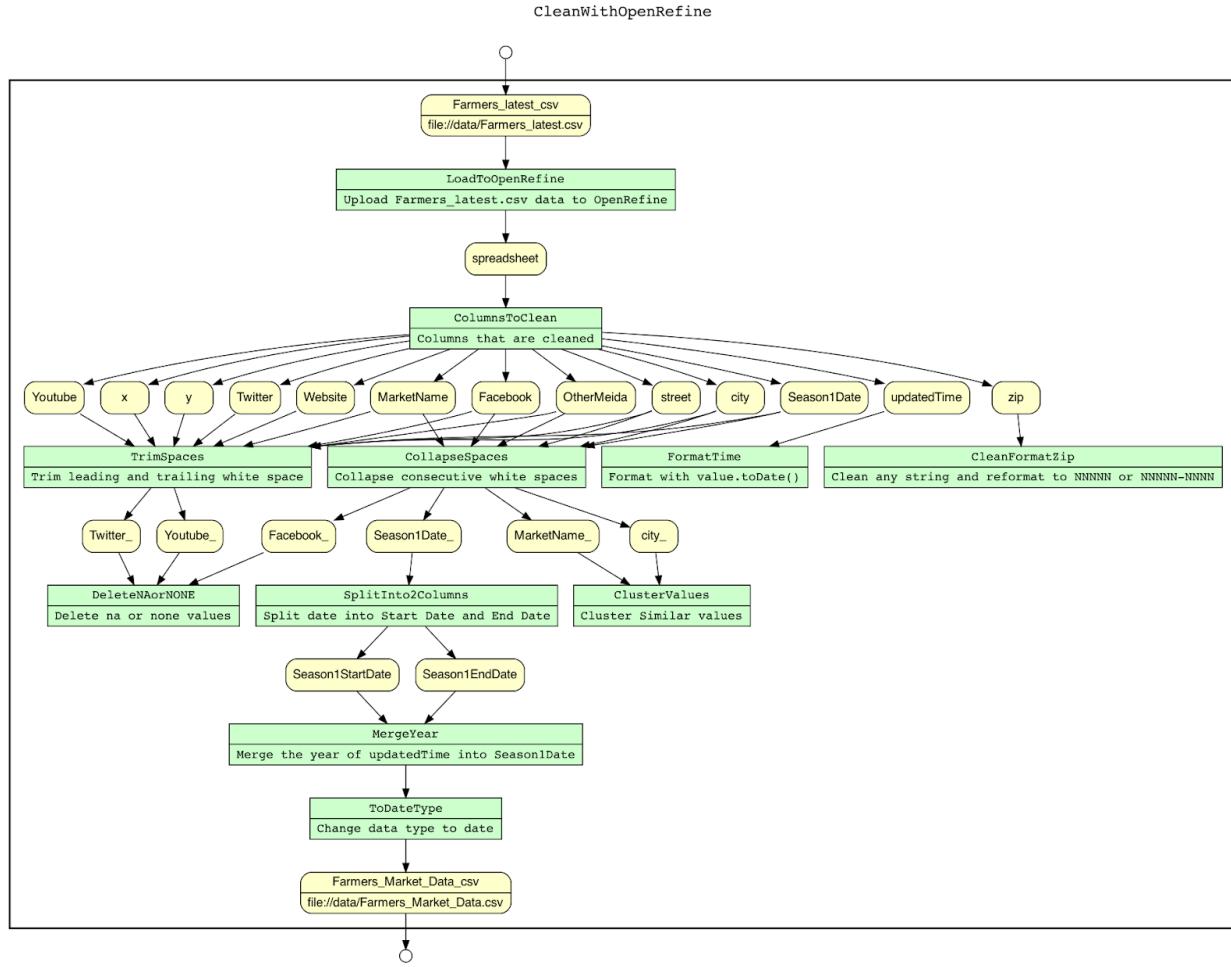
### 5.1 E2E Workflow:

The following is the end-to-end workflow for our project including data cleaning by OpenRefine, Data modeling and querying in sqlite.



## 5.2 Data Cleaning by OpenRefine

In the following workflow, it highlights each part of the data cleaning by OpenRefine for each attribute of the dataset.



## 6. Conclusions and Future Work

In this project, we analyzed the data from Farmers Market datasets which contain very informative attributes but too much noisy, diverse formats and empty information. We first analyzed all these attributes and categorized them into several categories based on the semantic meaning and business scenarios of the attributes to **identification, information, location, Seasonality payment and products** categories.

In order to have high quality data for better analysis, we used OpenRefine to do the data cleaning with text cleaning, and clustering to have clean data and consistent formats. Although

there is still some data not being cleaned or reformatted into a perfect way. One problem is that the data contains too much missing values for several attributes. But overall, the data quality after data cleaning is good enough to support our analysis.

In the data analysis part, we leveraged the SQL to convert the data to structured data with data types and formats. On top of this SQL data, we experimented with diverse analysis on the data to analyze the business insights behind the Farmers Market datasets. In this process, we also double checked the data quality when performing the data analysis.

In the last, we also created the workflow model to track our end-to-end pipeline including the data cleaning, data modeling and analysis.

In the next step, we want to continue the data cleaning using OpenRefine and other tools (For example Python Script, SQL Script) And more data modeling and analysis will also continue to have better understanding and insights.