Final project walk-through

1. Overview and initial assessment (page or half page without screen shots)

Schema, content(variables), check quality problem

Use case: what is the goal of data cleaning?

Limits or boundaries maybe (ignoring specific columns)

Some missing information that could have been useful

1. Data cleaning with OpenRefine

Provenance information

Provide openrefine operation history(json file)

Cluster some duplicate column names

What and why you are doing it.

Include screen shots

Converting data format, splitting columns, merge columns included

1. (Optional) Applying more suitable solutions (R, Python)

Tools that could be much useful (happy to see extra exercises)

Individually too much efforts

1. Develop a relational database schema ()

Create schema to load the data into SQLite

Integrity constraints

Explanation could be very short

1~2 integrity constraints

Execute queries and results

1. Create a workflow model

Selling the product to customers (work as an explanation)

Diagram the work you’ve done for cleaning the data

1. (Optional) Develop provenance queries (Datalog /DLV)

1. Dataset overview and Initial assessment

1.1 Dataset

US Farmers Market dataset is lists of markets that sell agricultural products directly to customers at a common, recurrent physical location. It is originally designed to provide customers information about farmers market such as market locations, directions, operating times, media sources, products sell, and more.

The dataset contains 8687 market information with 59 different attributes.

FMID: (integer) Identifier of each farmers market

MarketName: (string) Name of each market

Website: (string) Website address of each market

Facebook: (string) Facebook address of each market

Twitter: (string)Twitter address of each market

Youtube: (string) Youtube address of each market

OtherMedia: (string) Any other media address except those mentioned above

street: (string) Specific position that market is located in.

city: (string)city that market is located in

County: (string)County of the market

State: (string)State of the market

zip: (integer) zip code of the market

Season1Date: (date)

Season1Time: (time)

x: (numeric) longitude of the market

y: (numeric) latitude of market

Location: (string) description of the market location that the market is actually located in, such as private parking lot, specific buildings

Credit~WildHarvested: (binary) Whether some of products or payment methods are available

updateTime: (date) the date of information updateds

1.2 Quality Issue

Is data clean enough for use cases?

Dataset contains lots of null data. Mostly, it lacks media attributes such as website, facebook and others. Although some location data like street, city are missing, it can be replaced by using x,y data which are assumed to be longitude and latitude of the market. Too much season date and time are missing, which seem to be almost useless. Also, information about what kind food are available in the market are missing largely.

Session date are not in the same format. They mostly follow DD/MM/YYYY format, but some are in Month DD, YYYY format.

Almost 95% of Session2 to session 4 date and time data are missing.

County text needs to be clustered, it contains both lowercase and uppercase.

1.3 Use Case

Real use case

How much of each payments methods such as credit, WIC and others in overall market places are available.

List of products available in each market.

Analyzing the number of farmers markets in each city or states

The number of farmers markets per 10,000 people in each city or states

Hypothetical use case

After going through some cleaning process, this dataset can be used for:

Create a map that give exact locations of each Farmers market with some of information

* With the use of latitude and longitude data, some of the street locations should be reformatted or replaced.

In what period of time the farmers market session1 open most or least

* Session1 date should be reformatted in same format such as DD/MM/YYYY.

The number of farmers markets per 10,000 people in each county

-> county data needs to be clustered into same values.

2. Data Cleaning with OpenRefine

With OpenRefine, data will be clustered if they are in similar text, or reformatted to keep consistency of data. Firstly, all column data should be trimmed and collapsed if they have consecutive white spaces. Next, county, city, States names are inconsistent. Some of them are in uppercase while others are not. They need to be converted into same format by clustering function. Some of SessionDate columns contain various formats which should be fixed with the use of regular expression.

2.1 MarketName

- Trim leading and trailing white space

- Collapse consecutive white spaces

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

- Use text facet and cluster again by using key collision method and ngram-fingerprint, except those have distinct different names such as Arkansas Home Grown Market II and Arkansas Home Grown Market III

2.2 Website

- Trim leading and trailing white space

2.3 Facebook

- Trim leading and trailing white space

- Collapse consecutive white spaces

- By using GREL, change ‘n/a’ or ‘none’ values to blank

- Reformat data that starts with ‘@data’ or ‘#data’ to <https://facebook.com/data> for better access and consistency by using python/jython

2.4 Twitter

- Trim leading and trailing white space

- Eliminate ‘N/A’ or ‘none’ string values by using clustering function

- Reformat data that starts with ‘@data’ or ‘#data’ to <https://twitter.com/data> for better access and consistency by using python/jython

2.5 Youtube

- Trim leading and trailing white space

- Eliminate ‘N/A’ or ‘none’ string values by using clustering function

2.6 OtherMedia

- Trim leading and trailing white space

- Collapse consecutive white spaces

2.7 street

- Trim leading and trailing white space

- Collapse consecutive white spaces

2.8 city

- Trim leading and trailing white space

- Collapse consecutive white spaces

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

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

2.9 zip

- Convert any string values to blank using GREL

- United States zip code follows NNNNN or NNNNN-NNNN format, check if the format is right with GREL

2.10 Season1Date

- Trim leading and trailing white space

- Collapse consecutive white spaces

- For better use and clear view, split into two columns: one is start date “Season1StartDate”, and the another is end date “Season1EndDate”

2.11 Season1StartDate

- Clean separator ‘to’ to blank

- Change the date type to date, and its format becomes YYYY.MM.DD

2.12 Season1EndDate

- Change the date type to date, and its format becomes YYYY.MM.DD

2.13 Season2,3,4Date and Season2,3,4Time

- Considering that about 95% of these columns are blank, drop the columns for decreasing sparsity of data

2.14 x

- Trim leading and trailing white space

2.15 y

- Trim leading and trailing white space

2.16 Organic (all binary data)

- Replace “-“ string to blank

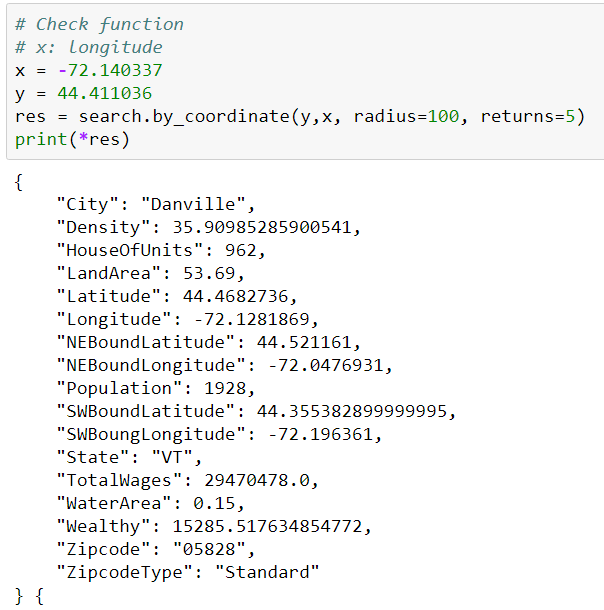
2.17 updatedTime

- Convert ‘Month DD YYYY time’ format to ‘MM/DD/YYYY time’ format for data consistency by using python/jython

3. Applying more suitable solutions(Python)

Albeit OpenRefine is a great tool for cleaning data, there are some limitations. For Farmers Market data, the tool cannot help filling some of missing zip code. And by using ‘uszipcode‘ package in python along with latitude, and longitude variables (which are ‘x’, ’y’), zip code can be filled.

Based on closest match from given latitude, and longitude, when city name of searched zip code and city name of the data matches, the data zip code is assigned with matched zip code. In case there are no matches for city names, zip code is given according to the closest distance.



Out of 961 missing zip code, 934 cases are filled. Remain 27 data is also missing x, y values, which makes it hard to find zip code.

- convert x,y columns to numeric values

- convert updateTime column to date values

- Trim leading and trailing white space to each columns

- Collapse consecutive white spaces for each column

- Cluster county data

- Session1Date reformatting into DD/MM/YYY

- Drop session2date session2time session3date session3time session4date session4time

- check zip code formatting with regular expression

4. Develop Relational Database Schema

SQLite integrity constraint

there should not be same FMID or null FMID

website format follows certain format

Website, Facebook, Twitter Youtube should not be same as OtherMedia or each other

No zipcode should be missing when x,y column data are present

zipcode check