# IS590DCO: Data Cleaning Final Project Report Boiled or mashed? That is a question

Xiaoliang Jiang[1], Yi Sun[2]

[12]*School of Information Sciences, University of Illinois at Urbana Champaign*

[1]xjiang36@illinois.edu[2]yisun5@illinois.edu

## 1. Abstract

Mashed potatoes and Boiled potatoes are very popular dishes in the US. to distinguish their differences among different people or different locations, time or other attributes, our group planned to visualize the data and implement data mining processing on the Menu data set. We used several data cleaning tools, data visualization tools and scientific workflow tools to finish our project. Finally, we found that there is no significant different from those two dishes based on the Menu data set.

## 2. Introduction

Mashed potatoes and Boiled potatoes are very popular dishes in the US, even among the world. The recipes of mashed potatoes started appearing from 1747 with an entry in The Art of Cookery, and the boiled potatoes have a longer history. As in general, they are very similar with each other, our research group is very interested in whether different people have different preference on these two dishes based on different locations and whether there is price difference between two dished over time. What's more, we are also interested in whether or not we can make a prediction on which dishes will be sell based on some attributes of the restaurant.

To answer these questions we are interested in, we decide to look at the New York Public Library's Menu data. The New York Public Library's (NYPL) restaurant menu collection is one of the largest in the world, which contains approximately 45,000 menus dating from the 1840s to the present, used by researchers, historians, novelists and chefs (NYPL, 2017). However, the data quality constrains the usability of this data which has different types of data quality problems, including spelling error, missing data and text formatting, etc. Therefore, a well-defined data cleaning processing is needed before searching for the greatest treasures in this data set. In order to answer our questions, we do the data cleaning processings separately by individual question and let them fit for users' operations, decision making and planning (Redman, 2013).

In addition, to trace our data-flow and let others understand our working process better, we want to use a workflow tool to show some important processing among our workings. According to what we learned from class, YesWorkflow enables scientists to annotate existing scripts with special comments that reveal the computational modules and dataflows otherwise implicit in these scripts (McPhillips, Song, & Kolisnik, 2015). Therefore, we also want to include YesWorkflow in our processing.

In conclusion, our first goal is to answer the questions we mentioned: Do people from different states have preference difference on potato dishes? Any significant difference between price for mashed potatoes and boiled potatoes? Can we make any prediction on which dishes will be sell based on some attributes of the restaurant? Our second goal is cleaning the data set to let it fit for our operations. Our third goal is using YesWorkflow to generate some diagrams to help us manage our working and express our ideas better to other researchers.

## 3. Method

To achieve our goals, we used several tools throughout each stage of our project, including data cleaning tools, data visualization tools, and the Workflow tool.
For data cleaning, we used Openrefine and R.

OpenRefine, formerly called Google Refine and before that Freebase Gridworks, is a standalone open source desktop application for data cleanup and transformation to other formats (OpenRefine, 2017).

R is a free software environment for statistical computing and graphics (R, 2017). We also used Rstudio to implement R scripts.RStudio makes R easier to use. It includes a code editor, debugging & visualization tools (RStudio, 2017).

For data visualization, we adopted plotly and Tableau. Plotly creates leading open source tools for composing, editing, and sharing interactive data visualization via the Web (Plotly, 2017). with intuitive drag & drop products. In addition, Tableau is a data visualization software which can combine multiple views of data to get richer insight without programming (Tableau, 2017).

Then, the YesWorkflow help us generate the workflow diagram. The YesWorkflow enables scientists to annotate existing scripts with special comments that reveal the computational modules and dataflows otherwise implicit in these scripts (McPhillips, Song, & Kolisnik, 2015).

We also used a R extension to generate YesWorkflow annotations in R scripts.I made this special extension of ‘strcode’ in my independent study project which can help users to generate YesWorkflow annotations in their R scripts with a well-defined user interface. In addition, users can also generate RDF file in YesWorkflow namespace (idaks, 2017) or ProvONE ontology (Cuevas-Vicenttín, Ludäscher, & Missier, 2014) automatically by a summary funciton.

In general, we used all of the tools above to finish our project.

### 3.1 Data acquisition and exploration

Since this data is open to public, we simply downloaded the zip file from the NYPL website. There are four comma separated files, named Menu.csv, MenuItem.csv, Menupage.csv and Dish.csv. After acquiring data, we tried to load the data into OpenRefine for a brief overview. However, the processing time of OpenRefine in loading data was unexpectedly long. With the help of online searching tool, we found some suggested solution to this issue, that is, increasing the memory in use for OpenRefine. Unfortunately, boosting up the memory did not help in our case. Thus we turned to Excel. By loading data into Excel, we had an basic idea of these four datasets as following form.

Here below are code we tried:

boost REFINE_MEMORY=8000M in refine.ini.
set REFINE_MEMORY in refine.bat.

*Figure 1. Changed settings in Openrefine to boost its function.*

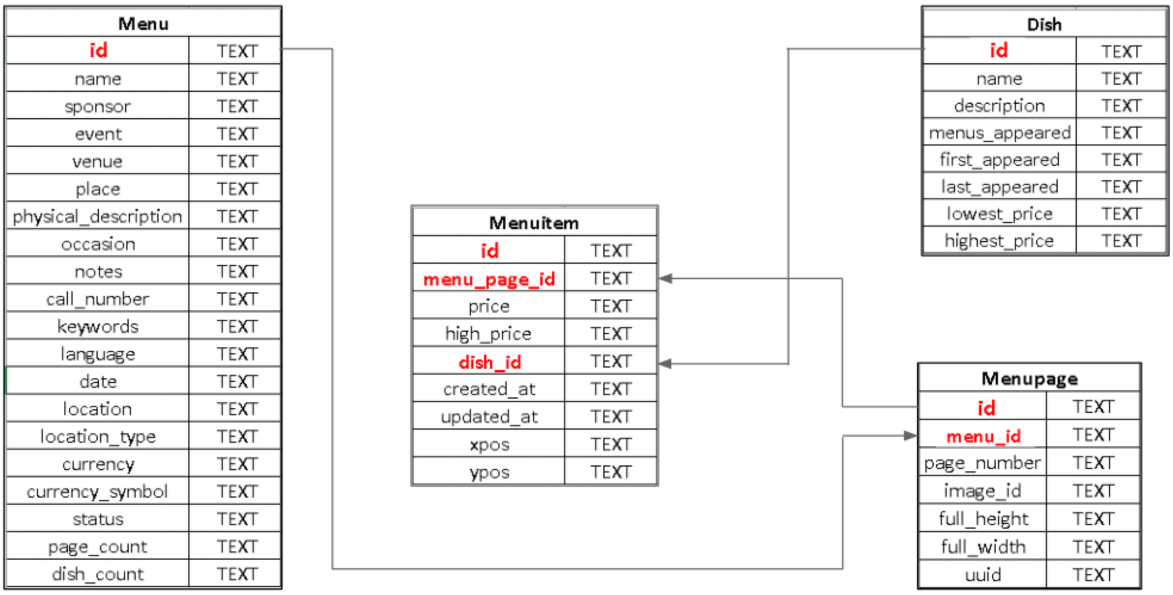
| Dataset name | Columns | Rows | Size |
|---|---|---|---|
| Menu.csv | 17545 | 20 | 3.08MB |
| Menupage.csv | 66937 | 7 | 4.55MB |
| MenuItem.csv | 1333287 | 9 | 113MB |
| Dish.csv | 424509 | 9 | 25.3MB |

*Table 1. Information of the data sets.*

According to the form above, the MenuItem is the largest dataset with 1333287 columns and 9 rows. We also found some issues that may cause problems when computing if they were not taken care of by the data cleaning process, such as the mixed-use of upper/lower case letters, misspelled names, ununified data formats and so on in multiple columns.

## 3.2 Data cleaning and preprocessing

### 3.2.1 Joining tables



| Menu | |
|---|---|
| id | TEXT |
| name | TEXT |
| sponsor | TEXT |
| event | TEXT |
| venue | TEXT |
| place | TEXT |
| physical_description | TEXT |
| occasion | TEXT |
| notes | TEXT |
| call_number | TEXT |
| keywords | TEXT |
| language | TEXT |
| date | TEXT |
| location | TEXT |
| location_type | TEXT |
| currency | TEXT |
| currency_symbol | TEXT |
| status | TEXT |
| page_count | TEXT |
| dish_count | TEXT |

| Menuitem | |
|---|---|
| id | TEXT |
| menu_page_id | TEXT |
| price | TEXT |
| high_price | TEXT |
| dish_id | TEXT |
| created_at | TEXT |
| updated_at | TEXT |
| xpos | TEXT |
| ypos | TEXT |

| Dish | |
|---|---|
| id | TEXT |
| name | TEXT |
| description | TEXT |
| menus_appeared | TEXT |
| first_appeared | TEXT |
| last_appeared | TEXT |
| lowest_price | TEXT |
| highest_price | TEXT |

| Menupage | |
|---|---|
| id | TEXT |
| menu_id | TEXT |
| page_number | TEXT |
| image_id | TEXT |
| full_height | TEXT |
| full_width | TEXT |
| uuid | TEXT |

*Figure 2. Information of tables and the connection between them.*

Based on our questions, we need information from all four datasets. Our first thought was to join four tables together. We took a further look into each table and found out that the Menupage, Menu and Dish table could be linked to the MenuItem table by some ID attributes. Here below shows the links we found to connect each table:

From here, we firstly used SQLite3 to join the four tables by the links we found above. Then, we eliminated the columns that are not related to our questions. After this step, we created 2 new datasets, named Mashed.csv with 6083 instances and Boiled.csv with 5901 instances. Both datasets has 11 attributes including 4 id columns that we used to join tables. The actual code we used is attached in the appendix section.

### 3.2.2 Basic cleaning

As we mentioned before, there were issues in string values in multiple column that we thought might be problematic when analyzing. We used OpenRefine to further clean and reformat our new datasets. Text Faceting was the main method we used not only to see the big picture of the values in columns, but also to filter out the possible clusters of values we want to change in bulk.

Firstly, for the column Dish_name in each dataset, the mix-used upper/lower case letters could be reviewed and fixed by using Text Facet and then cluster method in Openrefine. OpenRefine did a good job on discovering clusters that may represent the same thing but are not exactly the same in spelling due to the reason of mis-entering raw data. For each dataset, the final result in this column we want is all capitalized and unified spelling of the dish name, i.e. MASHED POTATOES and BOILED POTATOES. What's nice about OpenRefine in this operation is that in the pop-up clustering operation window we can select all clusters and manually modify the string value. However, it is sometimes risky if people didn't have enough prior knowledge of the meaning of some string values. Therefore, we did a brief check of all the clusters by eyeballing before we continue to merge all clusters.

We saw the same problematic issue in columns called event and place as well. This time, we found out that it might be easier if we get rid of the mixed-used upper/lower case. Thus we decided to first transform all values into uppercase and then applied the Text Facet to see what values could be clustered. The clustering process in this step was a little different than what we did for the Dish_name column, since we didn't have an unique expected value as before. We applied the clustering multiple times utilizing different combination of method and key functions until there were no further clusters available. The method we used was key collision and the key functions were fingerprint and methaphone3. Same as before, we selected and merge clusters by eyeballing.

Lastly, we modified the date column from string values to date format using the built-in method in OpenRefine. One of the reasons we re-formatted the date value is that it's easier to extract the date value, such as year only, for our further study. Luckily we had the data in string value that could be easily recognized and transformed by OpenRefine. Up to now we finished the basic cleaning of both datasets.

## 3.3 Data visualization

Our data is complex, and it is huge. Because of the way the human brain processes information, using charts or graphs to visualize large amounts of complex data is easier than poring over spreadsheets or reports.

### 3.3.1 Heatmap

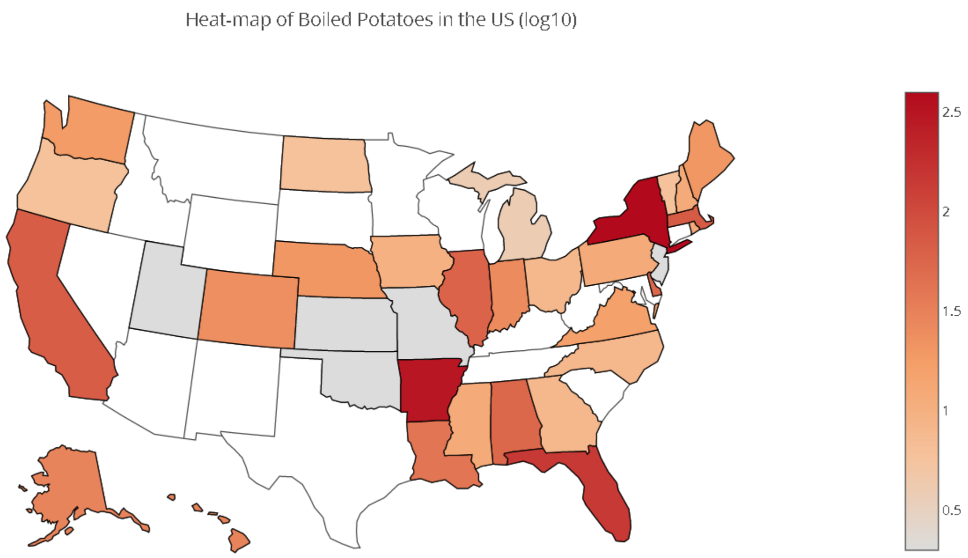


*Figure 3. Heat-map of Boiled Potatoes.*

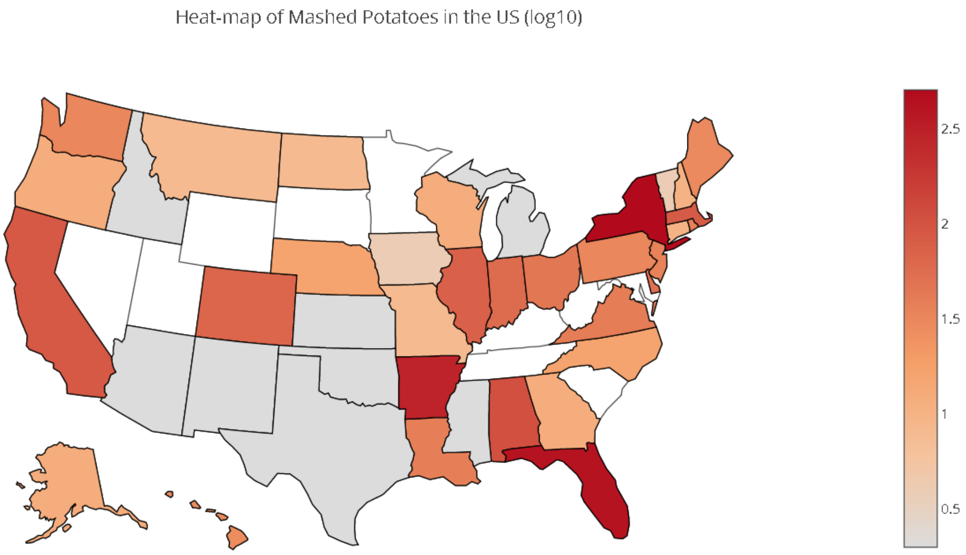


*Figure 4. Heat-map of Mashed Potatoes.*

To answer the first question we brought up, the first useful tool came into our mind was using Tableau to create a heatmap within the US map so as to see the potential differences of dish preference among states. However, when plotting the data onto US map according to states in Tableau requires the location longitude and latitude. We don't have these data for each menu in our raw data that could be used. Thus we turned to plotly, which is more friendly with location details when plotting.

In our datasets, the column place contains the detailed location information of the restaurant that owned the menu. As we can see from the raw data that some of the locations are very specific to detail but some are not. Besides, there are some restaurants that are outside of the United States. These are the questions we need to consider when deriving data that we are going to use in the data visualization process.

To further clean the data, first we want to exclude all entries that are outside the United States so that the data could fit in the US map. Second, the state information is detailed enough for the purpose of showing the differences among states. Our first try was to use OpenRefine to find out the keywords about the state information. However, OpenRefine was good at finding the keywords but we could not figure out the way to extract the keyword and its related information and to save them into separate column. We found the same issue in the keyword value extracting process in other columns. Therefore we decided to use R instead for the following steps in data extraction.

We extracted the state information from the place column and then saved it as state abbreviation, i.e. NY for New York state and IL for Illinois. Then we counted the frequency of the dish appearance in each state and applied log transformation because we would like the data to be less skewed. We applied the same method to both datasets and created the heatmap in plotly accordingly.

### 3.3.2 Line plot

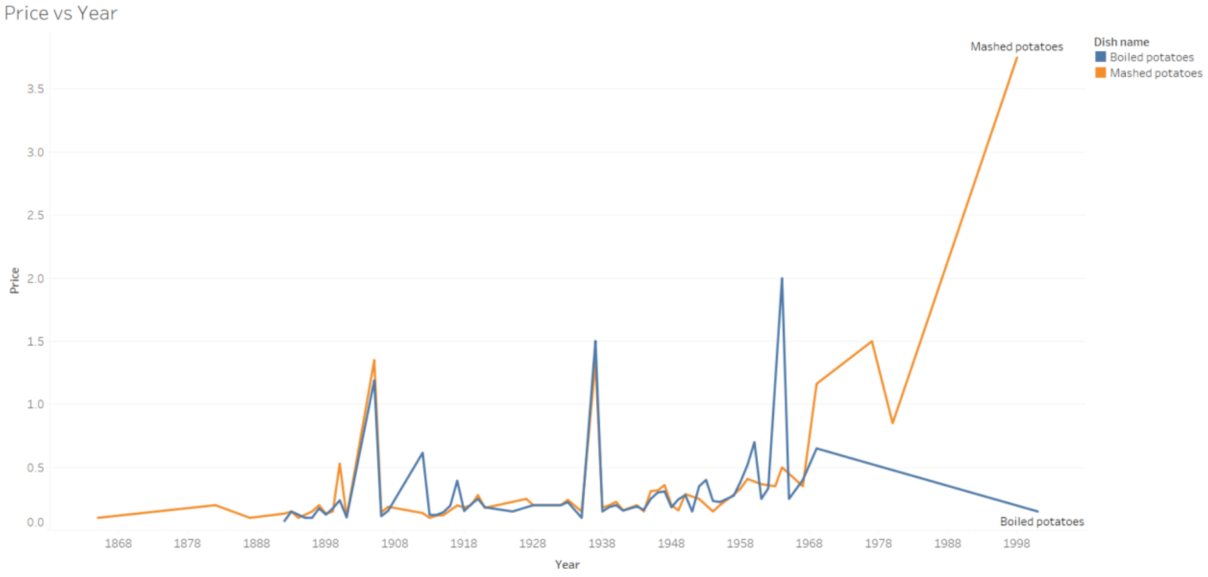


*Figure 5. Line plot of Price vs Year*

Our second question is to see if there is significant difference in price of these two potato dishes over year. We combined two tables by rows and decided to omit all NA values in both price and date column. For the price column, we omit the NA values because those values may impact the plot. Similary, we ommited NA values in date column. After the elimination, we were left with one dataset with 3 columns and 3810 rows. Loading this data into Tableau we created the line plot with X axis as Year and Y axis as Price. In the graph, the blue line represents the boiled potatoes and the yellow line represents the mashed potatoes.

## 3.4 Data mining

After combined the two datasets as we mentioned in the former section, we also did another data cleaning processing to generate a sub-dataset for data mining.In this step, originally, we import the bm.csv data which contains 11981 instances and 16 columns. And then we select five columns event, physical_description, state, price and Dish_name to create a sub-dataset. The former four attributes serve as attribute and the last one is category. We did data cleaning processing for each attribute, for the event we clustered them into 5 main categories by using different keywords: BREAKFAST (keyword: BREAK), LUNCH (keyword: LUN), DINNER (keyword: DIN),  null value named UNKNOWN and other value named DAILY. Similarly, for the physical_description, we clustered it into 6 main categories: FOLDER (keyword: FOLD), BOOKLET (keyword: BOOKLET), BROADSHEET (keyword: BROADSHEET), CARD (keyword:CAR), null value named UNKNOWN and others named OTHER. Then, we removed all null value in state, and there are 3760 instances and 5 columns remained. Similarly, we removed all null value in price, and finally we got 462 instances and 5 columns.

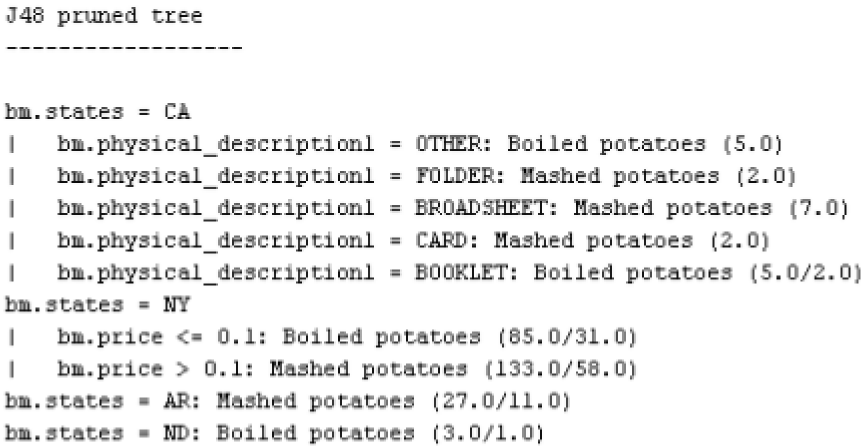
For data mining processing, we randomly sampled the data into two different data set: training set bmtrain.csv (362 instances and 5 columns) and testing set bmtest.csv (100 instances and 5 columns). Then we can load them into Weka and started to do the data mining.

| Algorithm name | Accuracy | Recall | Precision | F-Measure |
|---|---|---|---|---|
| Naive Bayes | 47% | 0.470 | 0.515 | 0.441 |
| ZeroR | 57% | 0.570 | 0.325 | 0.414 |
| IBk | 56% | 0.560 | 0.565 | 0.562 |
| JRip | 62% | 0.620 | 0.617 | 0.591 |
| Logistic Regression | 49% | 0.490 | 0.457 | 0.460 |
| J48 | 63% | 0.630 | 0.624 | 0.616 |

*Table 2. Results of different algorithms.*

In the Weka, we loaded these two datasets and change their attribute types in Weka and set Dish_name as class, then saved them as bmtrain.arff and bmtest.arff files. The arff is the suffix name of the dataset which can be generated from Weka and used in Weka. Then, after loaded these two arff files, we tried several popular algorithms to do the classification. The algorithms include Naive Bayes, ZeroR, IBk, JRip, Logistic Regression, and J48. The results of these algorithms are as the form above.

According to the result, the rule-based algorithms showed better performances. It is reasonable, since most attributes in our dataset are nominal, the rule-basedd algorithms can show the relationships within the attributes best. For example, here is an exaple of J48 Prune Tree example:



```
J48 pruned tree
------------------

bm.states = CA
|   bm.physical_description1 = OTHER: Boiled potatoes (5.0)
|   bm.physical_description1 = FOLDER: Mashed potatoes (2.0)
|   bm.physical_description1 = BROADSHEET: Mashed potatoes (7.0)
|   bm.physical_description1 = CARD: Mashed potatoes (2.0)
|   bm.physical_description1 = BOOKLET: Boiled potatoes (5.0/2.0)
bm.states = NY
|   bm.price <= 0.1: Boiled potatoes (85.0/31.0)
|   bm.price > 0.1: Mashed potatoes (133.0/58.0)
bm.states = AR: Mashed potatoes (27.0/11.0)
bm.states = ND: Boiled potatoes (3.0/1.0)
```

*Figure 6. A sub-ruleset of J48 algorithm.*

According to this sub-rulesets, we can found that the algorithm set several rules based on this dataset. Take the NY states as an example (line 7-9), if the states is NY, if the price of dishes is no higher than 0.1, it probably belongs to Boiled potatoes, however, if the price is higher than 0.1, it is more like Mashed potatoes instead of Boiled potatoes. Detailed information about our project can be found in our Github repository which can be found in the Appendix.

So, based on the results of the algorithms, we did not find significantly high accuracy rate which means that the Boiled potatoes and Mashed potatoes are hard to be distinguished by existing attributes. Therefore, we cannot answer the question three based on current research. It is hard to distinguish them with each other right now.

## 3.5 Workflow Tracking

### 3.5.1 Main YesWorkflow Graph

As we discussed before, the scientific workflow management systems can help users to deal with complex computational pipelines from modular building blocks, executing the resulting automated workflows, and recording the provenance of data products resulting from workflow runs (McPhillips, Song, & Kolisnik, 2015). YesWorkflow is such a set of software tools that aim to provide many of the benefits of scientific workflow systems.

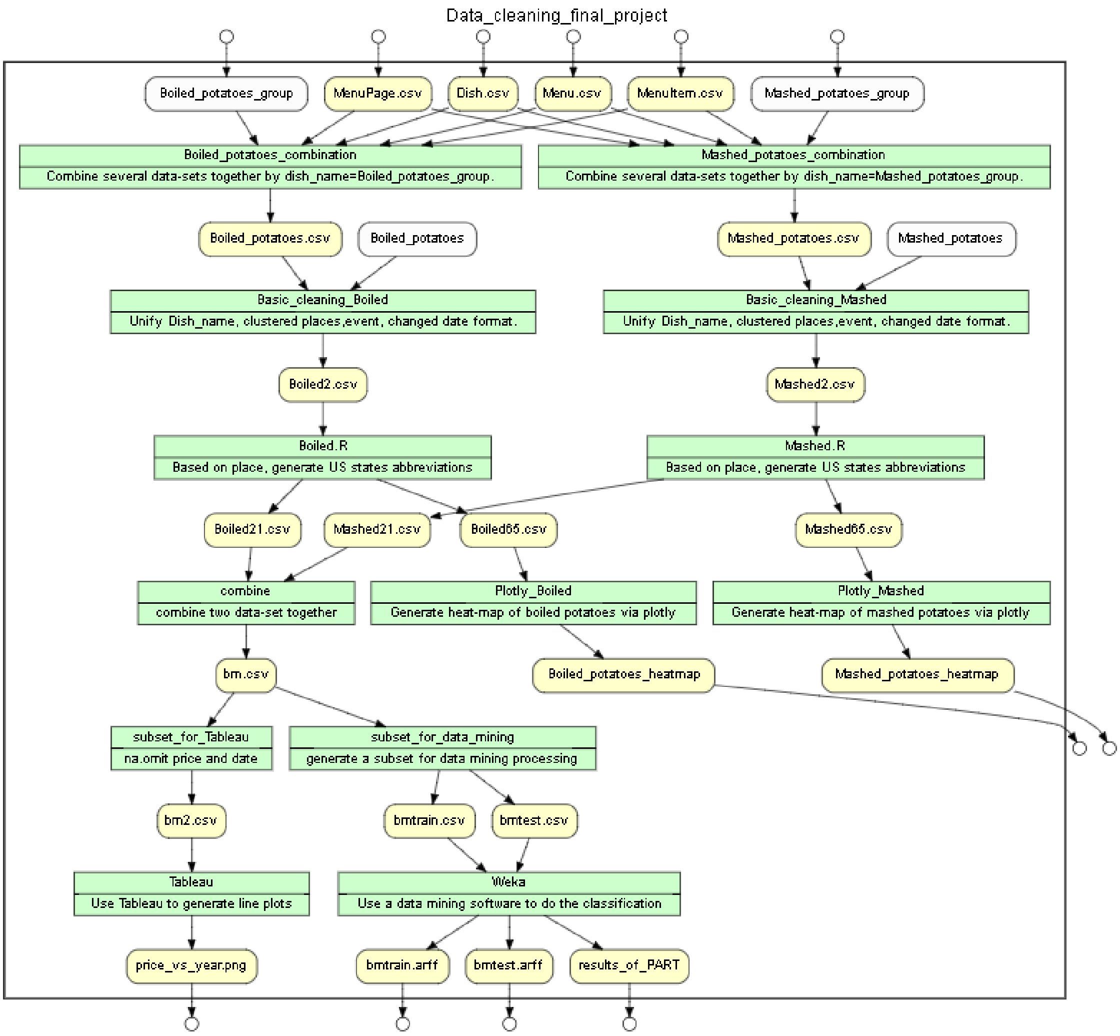

Data_cleaning_final_project
Boiled_potatoes_group
MenuPage.csv
Dish.csv
Menu.csv
MenuItem.csv
Mashed_potatoes_group
Boiled_potatoes_combination
Combine several data-sets together by dish_name=Boiled_potatoes_group.
Mashed_potatoes_combination
Combine several data-sets together by dish_name=Mashed_potatoes_group.
Boiled_potatoes.csv
Boiled_potatoes
Mashed_potatoes.csv
Mashed_potatoes
Basic_cleaning_Boiled
Unify Dish_name, clustered places,event, changed date format.
Basic_cleaning_Mashed
Unify Dish_name, clustered places,event, changed date format.
Boiled2.csv
Mashed2.csv
Boiled.R
Based on place, generate US states abbreviations
Mashed.R
Based on place, generate US states abbreviations
Boiled21.csv
Mashed21.csv
Boiled65.csv
Mashed65.csv
combine
combine two data-set together
Plotly_Boiled
Generate heat-map of boiled potatoes via plotly
Plotly_Mashed
Generate heat-map of mashed potatoes via plotly
bm.csv
Boiled_potatoes_heatmap
Mashed_potatoes_heatmap
subset_for_Tableau
na.omit price and date
subset_for_data_mining
generate a subset for data mining processing
bm2.csv
bmtrain.csv
bmtest.csv
Tableau
Use Tableau to generate line plots
Weka
Use a data mining software to do the classification
price_vs_year.png
bmtrain.arff
bmtest.arff
results_of_PART

*Figure 7. YesWorkflow diagram for our processing*

In this project, to help other researchers understand our methods and processing, we use YesWorkflow Editor to generate YesWorkflow Diagram (YesWorkflow Editor, 2017) to represent the whole processing and some key step.

Originally, we have four data sets as what we discussed above. Firstly, we use Openrefine to combine those four data sets together and generate two different data sets Mashed.csv (6083 rows and 11 columns) and Boiled.csv (5901 rows and 11 columns). The numbers of instances in these two dataset are similar which let the following processing feasible and reasonable. Secondly, we used Openrefien to finish some basic data cleaning processing. In addition, we use R to generate different sub data for the different data visualization tasks. By using Boiled.R and Mashed.R, we created two small subset to generate the heatmap as we mentioned before. Moreover, by bm.R we generated a bm.csv to draw the plot of price vs price for both dishes. Finally, we also generate two new dataset bmtest.csv and bmtrain.csv to do the data mining in Weka.

### 3.5.1 Main YesWorkflow Graph

We also generate a sub-diagram for one of the most complex processing ”creating the subset for data mining.” which is in the R scripts bm.R. The following diagram shows what we did for additional data cleaning in 3.4 Data mining preprocessing section. In this section, we used a special R extension of “strcode” to generate YesWorkflow annotations in Rstudio. It contains a user interface to help users generate the special comments in their scripts. It is the outcome of my independent study in this semester. Detailed information about the code can be found in the appendix section at the end of this report.

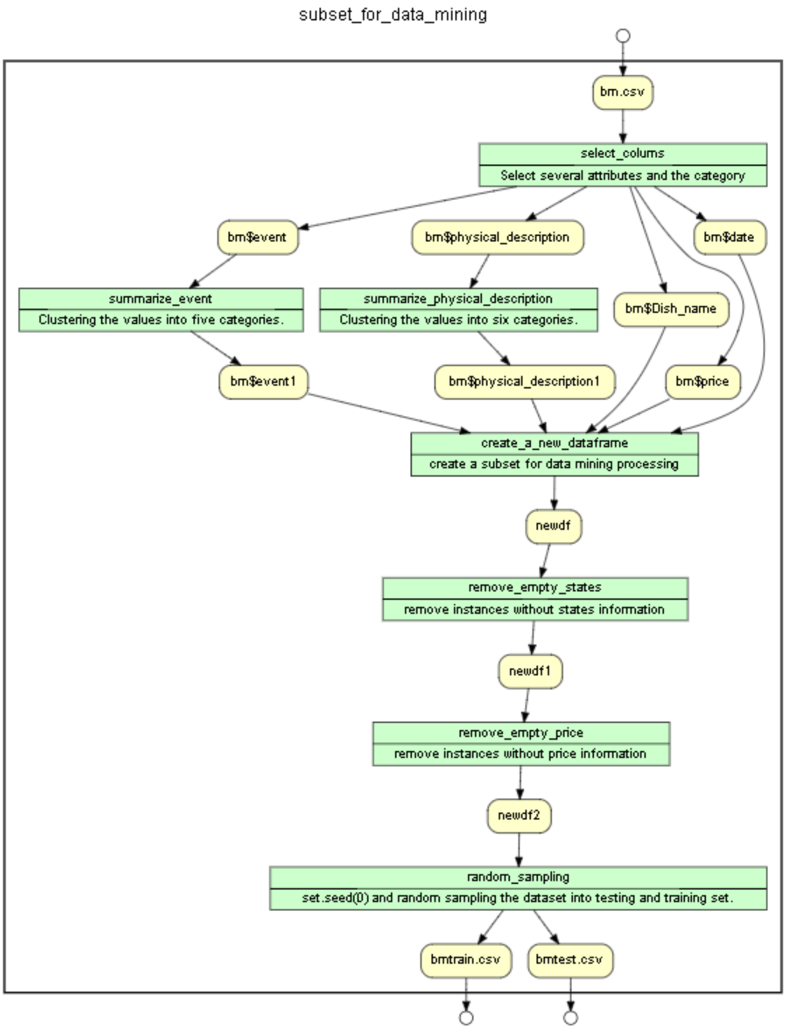

subset_for_data_mining
bm.csv
select_colums
Select several attributes and the category
bm$event
bm$physical_description
bm$date
summarize_event
Clustering the values into five categories.
summarize_physical_description
Clustering the values into six categories.
bm$Dish_name
bm$event1
bm$physical_description1
bm$price
create_a_new_dataframe
create a subset for data mining processing
newdf
remove_empty_states
remove instances without states information
newdf1
remove_empty_price
remove instances without price information
newdf2
random_sampling
set.seed(0) and random sampling the dataset into testing and training set.
bmtrain.csv
bmtest.csv

*Figure 8. YesWorkflow of creating sub-dataset for data mining.*

# 4. Conclusion

## 4.1 Findings

From the data visualizatio results, we conclude some findings to our questions. First of all, by looking at two heatmaps, we didn’t see big differences in preferences of potato dishes among most of the states in east and west coast in general. However, we do spot some minor differences in some states in midwest. States like Montana, Minnesota, Wisconsin, Iowa and Michigan tend to prefer boiled potatoes a little more than smashed potatoes while states like Arizona, New Mexico, Texas and Missouri tend to prefer mashed potatoes a little more than boiled potatoes. This finding may lead to a further study of why the difference of preference occured.

Secondly, from the line plot we can see some interesting findings. In general, we can see that two lines are changing in the similar trend, especially in the time span from 1892 to 1969. There is no significant change of price differences between two potato dishes. However, the obvious unusual peak of yellow line and drop of blue line at the right end of the graph deserves a further investigation in the raw dataset, as well as the extension to the left of yellow line. Tracing back to the data we found out that from 1969 to 2015, we only have 3 data points in mashed potato data, which results in the weird peak in yellow line. Similarly, there are only 2 data points in boiled potato data, thus the blue line seemed to drop. Before 1892 there is no data point for boiled potato data so the yellow line, representing mashed potato data, seemed to be extended to the left.

Over all, the difference in price over years and dishes preferences among states are not as obvious as our first thought. But there are some interesting findings that we could dig in further, and relates the findings to other fields that may provide hints to the answers.

## 4.2 Limitation

There are several limitations in our projects. Due to the limited attributes we selected, there are no significant differences can be found in our results between these two different dishes. In addition, limited attributes also constrain the accuracy of the data mining result. In the future research, more attributes may be needed to get a better result.

Moreover, YesWorkflow is a powerful tool sets which not only can generate workflow diagram but also support prospect and retrospect queries. We may include these powerful functions in our future work to analyze our working processing.

## 5. References

Cuevas-Vicenttín, V., Ludäscher, B., & Missier, P. (2014, March 27). *ProvONE: A PROV Extension Data Model for Scientific Workflow Provenance*. From ProvONE: A PROV Extension Data Model for Scientific Workflow Provenance: http://vcvcomputing.com/provone/provone.html

*idaks*. (2017). From Github: https://github.com/idaks/DataONE-Prov-Summer-2017

McPhillips, T., Song, T., & Kolisnik, T. (2015). YesWorkflow: A User-Oriented, Language-Independent Tool for Recovering Workflow Information from Scripts. *International Journal of Digital Curation, 10*(1746-8256). doi:http://dx.doi.org/10.2218/ijdc.v10i1.370

NYPL. (2017). *NYPL Labs*. From NYPL Labs: http://menus.nypl.org/about

*OpenRefine*. (2017). From Wikipedia: https://en.wikipedia.org/wiki/OpenRefine

*Plotly*. (2017). From Plotly: https://plot.ly

*R*. (2017). From The R Project for Statistical Computing: https://www.r-project.org/

Redman, T. C. (2013, December 30). Data Driven: Profiting from Your Most Important Business Asset. *Harvard Business Press*.

*RStudio*. (2017). From RStudio: https://www.rstudio.com/

*Tableau*. (2017). From Tableau: https://www.tableau.com/

*YesWorkflow Editor*. (2017). From YesWorkflow Editor: http://absflow.westus.cloudapp.azure.com

# 6. Appendix

## 6.1 Github Repository

Here is the Github repository of our project. Detailed information about the R scripts, graph, generated tables and the data mining results can be found here:
https://github.com/XiaoliangJiang/FALL17IS590DC_FinalProject/tree/master

## 6.2 SQLite3 code

### 6.2.1 SQLite3 code of Mashed potatoes

Code 1, selecting Boiled potatoes instances:

```
SELECT Dish.id as Dish_id, Dish.name as Dish_name, Menuitem.menu_page_id,
Menuitem.price, Menu.id AS Menu_id, Menupage.id AS Menupage_id, Menu.event,
Menu.place, Menu.physical_description, Menu.notes, Menu.date FROM Dish JOIN
Menuitem JOIN Menu JOIN Menupage ON (Dish.name='Boiled potatoes' OR
Dish.name='Boiled Potatoes' OR Dish.name='Boiled Potatoes (2)' OR Dish.name='New
Boiled Potatoes (2)' OR Dish.name='BOILED POTATOES' OR Dish.name='boiled potatoes'
OR Dish.name='Boiled Potatoes.' OR Dish.name='New boiled potatoes' OR
Dish.name='Boiled Potatoes ' OR Dish.name='Boiled Potatoes (2)' OR Dish.name=' Boiled
Potatoes') and Dish.id=Menuitem.dish_id AND Menuitem.menu_page_id=Menupage.id AND
Menupage.menu_id=Menu.id;
```

### 6.2.2 SQLite3 code of Boiled potatoes

Code 2, selecting Mashed potatoes instances:

```
SELECT Dish.id as Dish_id, Dish.name as Dish_name, Menuitem.menu_page_id,
Menuitem.price, Menu.id AS Menu_id, Menupage.id AS Menupage_id, Menu.event,
Menu.place, Menu.physical_description, Menu.notes, Menu.date FROM Dish JOIN
Menuitem JOIN Menu JOIN Menupage ON (Dish.name='Mashed potatoes' OR
Dish.name='Mashed Potatoes' OR Dish.name='MASHED POTATOES' OR
Dish.name='mashed potatoes' OR Dish.name='MASHED POTATOES.' OR
Dish.name='mashed Potatoes' OR Dish.name='*Mashed potatoes' OR Dish.name='*Mashed
Potatoes' OR Dish.name='Mashed Potatoes .35') and Dish.id=Menuitem.dish_id AND
Menuitem.menu_page_id=Menupage.id AND Menupage.menu_id=Menu.id;
```

## 6.3 YesWorkflow annotations

### 6.3.1 YesWorkflow annotations of main diagram

```
# @begin Data_cleaning_final_project @desc Data Cleaning Final Project
# @in Menu.csv
# @in MenuItem.csv
# @in MenuPage.csv
# @in Dish.csv
# @param Mashed_potatoes_group
# @param Boiled_potatoes_group
# @out price_vs_year.png
# @out Boiled_Potatoes.png @as Boiled_potatoes_heatmap
# @out Mashed_Potatoes.png @as Mashed_potatoes_heatmap
# @out bmtrain.arff
# @out bmtest.arff
# @out results_of_PART

    # @begin Mashed_potatoes_combination @desc Combine several data-sets together by dish_name=Mashed_potatoes_group.
    # @in Dish.csv
    # @in Menu.csv
    # @in MenuItem.csv
    # @in MenuPage.csv
    # @param Mashed_potatoes_group
    # @out Mashed_potatoes.csv
    # @end Prefilter

    # @begin Boiled_potatoes_combination @desc Combine several data-sets together by dish_name=Boiled_potatoes_group.
    # @in Dish.csv
    # @in Menu.csv
    # @in MenuItem.csv
    # @in MenuPage.csv
    # @param Boiled_potatoes_group
    # @out Boiled_potatoes.csv
    # @end Boiled_potatoes_combination

    # @begin Basic_cleaning_Mashed @desc Unify Dish_name, clustered places,event, changed date format.
    # @param Mashed_potatoes
    # @in Mashed_potatoes.csv
    # @out Mashed2.csv
    # @end Basic_cleaning

    # @begin Basic_cleaning_Boiled @desc Unify Dish_name, clustered places,event, changed date format.
    # @param Boiled_potatoes
    # @in Boiled_potatoes.csv
    # @out Boiled2.csv
    # @end Basic_cleaning

    # @begin Boiled.R @desc Based on place, generate US states abbreviations
    # @in Boiled2.csv
    # @out Boiled21.csv
    # @out Boiled65.csv
    # @end Boiled.R

    # @begin Plotly_Boiled @desc Generate heat-map of boiled potatoes via plotly
    # @in Boiled65.csv
    # @out Boiled_Potatoes.png @as Boiled_potatoes_heatmap
    # @end Plotly_Boiled

    # @begin Mashed.R @desc Based on place, generate US states abbreviations
    # @in Mashed2.csv
    # @out Mashed21.csv
    # @out Mashed65.csv
    # @end Mashed.R

    # @begin Plotly_Mashed @desc Generate heat-map of mashed potatoes via plotly
    # @in Mashed65.csv
    # @out Mashed_Potatoes.png @as Mashed_potatoes_heatmap
    # @end Plotly_Mashed

    # @begin combine @desc combine two data-set together
    # @in Boiled21.csv
    # @in Mashed21.csv
    # @out bm.csv
    # @end combine

    # @begin subset_for_Tableau @desc na.omit price and date
    # @in bm.csv
    # @out bm2.csv
    # @end subset_for_Tableau
     # @begin subset_for_data_mining @desc generate a subset for data mining processing
    # @in bm.csv
    # @out bmtrain.csv
    # @out bmtest.csv
    # @end subset_for_data_mining

    # @begin Tableau @desc Use Tableau to generate line plots
    # @in bm2.csv
    # @out price_vs_year.png
    # @end Tableau

    # @begin Weka @desc Use a data mining software to do the classification
    # @in bmtrain.csv
    # @in bmtest.csv
    # @out bmtrain.arff
    # @out bmtest.arff
    # @out results_of_PART
    # @end Weka
# @end Data_cleaning_final_project
```

### 6.3.1 YesWorkflow annotations of diagram about creating subset for data mining

```
# @begin subset_for_data_mining @desc Subset Workflow for data mining
# @in bm.csv
# @out bmtrain.csv
# @out bmtest.csv
   # @begin select_colums @desc Select several attributes and the category
   # @in bm.csv
   # @out bm$Dish_name
   # @out bm$price
   # @out bm$date
   # @out bm$event
   # @out bm$physical_description
   # @end select colums

   # @begin summarize_event @desc Clustering the values into five categories.
   # @in bm$event
   # @out bm$event1
   # @end summarize_event

   # @begin summarize_physical_description @desc Clustering the values into six categories.
   # @in bm$physical_description
   # @out bm$physical_description1
   # @end summarize_physical_description

   # @begin create_a_new_dataframe @desc create a subset for data mining processing
   # @in bm$Dish_name
   # @in bm$event1
   # @in bm$price
   # @in bm$date
   # @in bm$physical_description1
   # @out newdf
   # @end create_a_new_dataframe

   # @begin remove_empty_states @desc remove instances without states information
   # @in newdf
   # @out newdf1
   # @end remove_empty_states

   # @begin remove_empty_price @desc remove instances without price information
   # @in newdf1
   # @out newdf2
   # @end remove_empty_price
```

```
    # @begin random_sampling @desc set.seed(0) and random sampling the dataset into testing and training set.
    # @in newdf2
    # @out bmtrain.csv
    # @out bmtest.csv
    # @end random_sampling

# @end subset_for_data_mining
```