MIDS W205

|  |  |  |  |
| --- | --- | --- | --- |
| **Lab #** | 10 | **Lab Title** | OpenRefine -- Introduction |
| **Related Module(s)** | 10 | **Goal** | Get you started on OpenRefine and Edit Distance |
| **Last Updated** | 9/27/15 | **Expected duration** | 60 minutes |

## Introduction

OpenRefine is an open source tool for working with bad data. In this Lab we will give you a quick tour of how you can use it to clean data. If you want a more comprehensive tutorial you can follow any of the tutorials listed in the resources section.

We will be using two data sets one from with earthquake data and one with customer complaint data. The first data set is the eq2015 data set which data about earthquakes of magnitude 3 or more during the first 6 months of 2015. You can download the data set here . You can fine a data attribute glossary here The second data set contains customer complaints, you can download that data set here . Please answer the following questions by using OpenRefine.

OpenRefine is to a large extent menu driven. But it also allows you to use a language for doing certain types of transformations.

# Instructions, resources and prerequisites

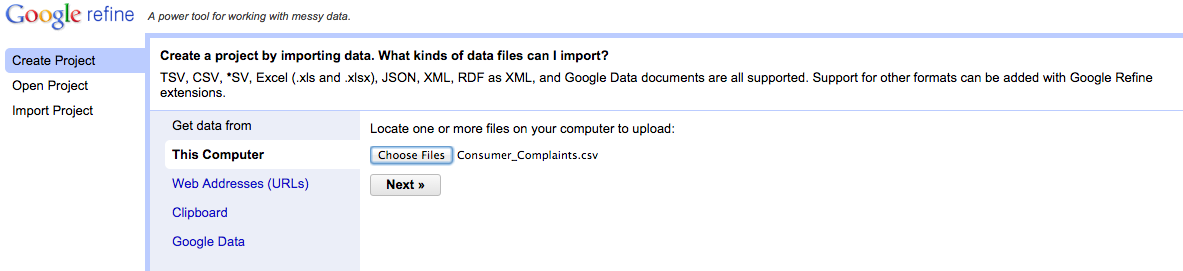
|  |  |
| --- | --- |
| **Resource** | **What** |
| http://openrefine.org/ | This is where you download OpenRefine. |
| http://arcadiafalcone.net/GoogleRefineCheatSheets.pdf | A short description of OpenRefine commands. |
| http://enipedia.tudelft.nl/wiki/OpenRefine\_Tutorial | Another tutorial on OpenRefine . |
| http://davidhuynh.net/spaces/nicar2011/tutorial.pdf | Another tutorial on OpenRefine . |
| http://schoolofdata.org/handbook/recipes/cleaningdatawithrefine/ | Programming guide for the Spark Context object. Here you can find actions available on the Spark Contexts. |
| https://github.com/OpenRefine/OpenRefine/wiki/GeneralRefineExpressionLang  uage | GREL is the language used in OpenRefine for data refinements. This is a reference guide for the GREL language. |
| http://earthquake.usgs.gov/eart | Explanation of the Earthquake data. |

Cleaning Data with OpenRefine.

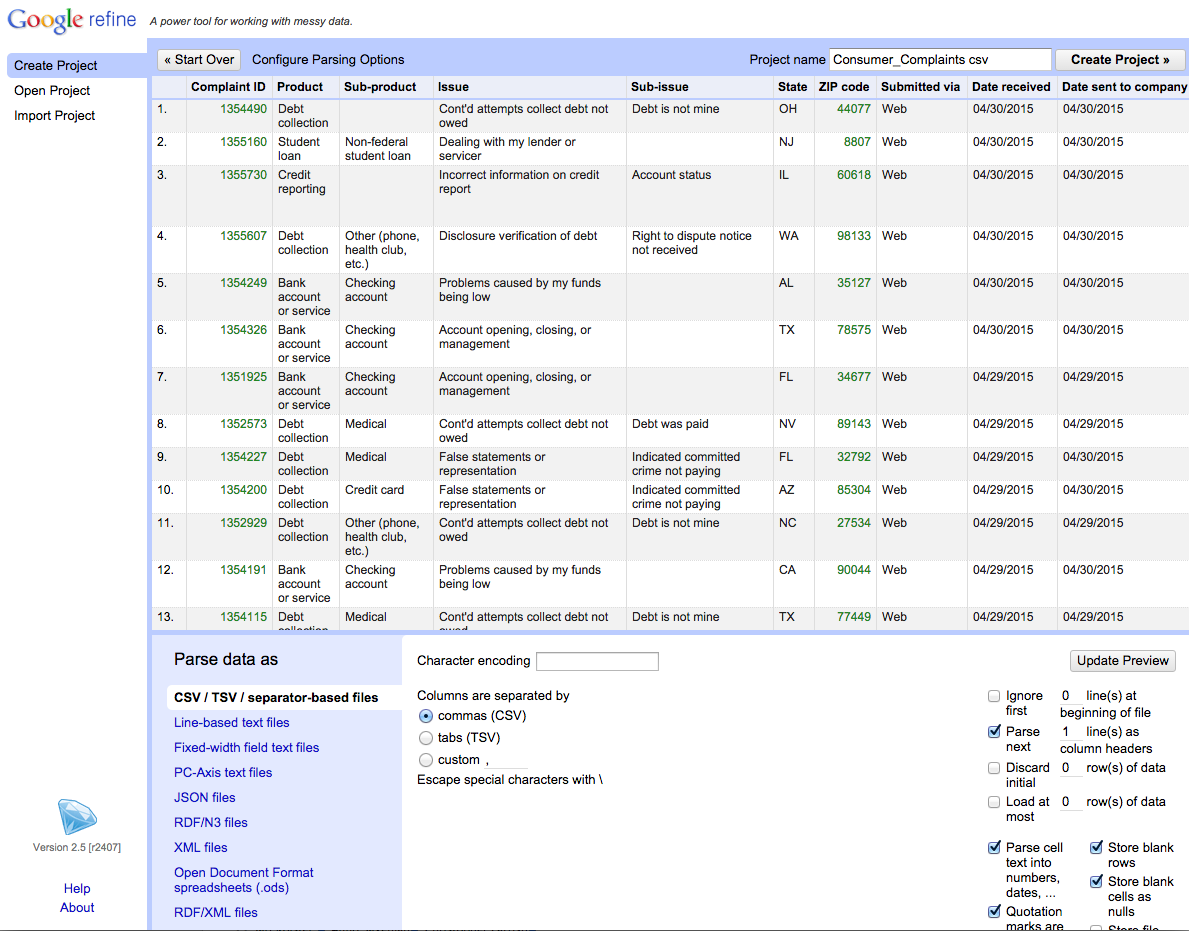
# Step-1. Wrangling the Customer Complaints Data

## Uploading data

After you started OpenRefine you can pick a data set. For this first step choose the Customer Complaints Data set.

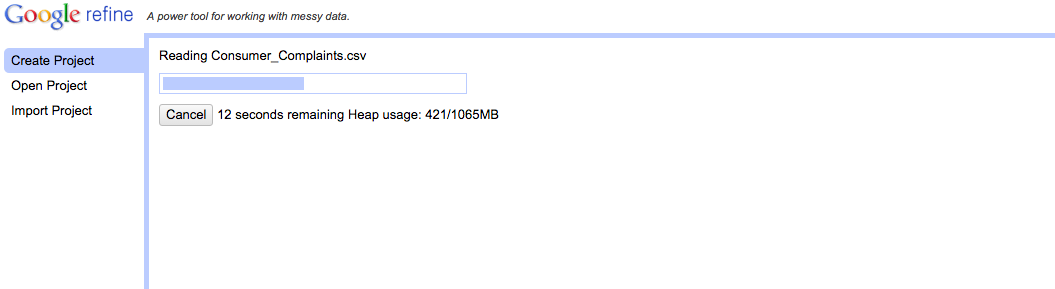


Once the data is read you can inspect it. In this case it looks ok. But lets say that it would have been tab separated, then OpenRefine would not have read it correctly. You have the opportunity to look at the data here and confirm it is ok. Ion this case we think it looks good and we click the “Create Project” button.

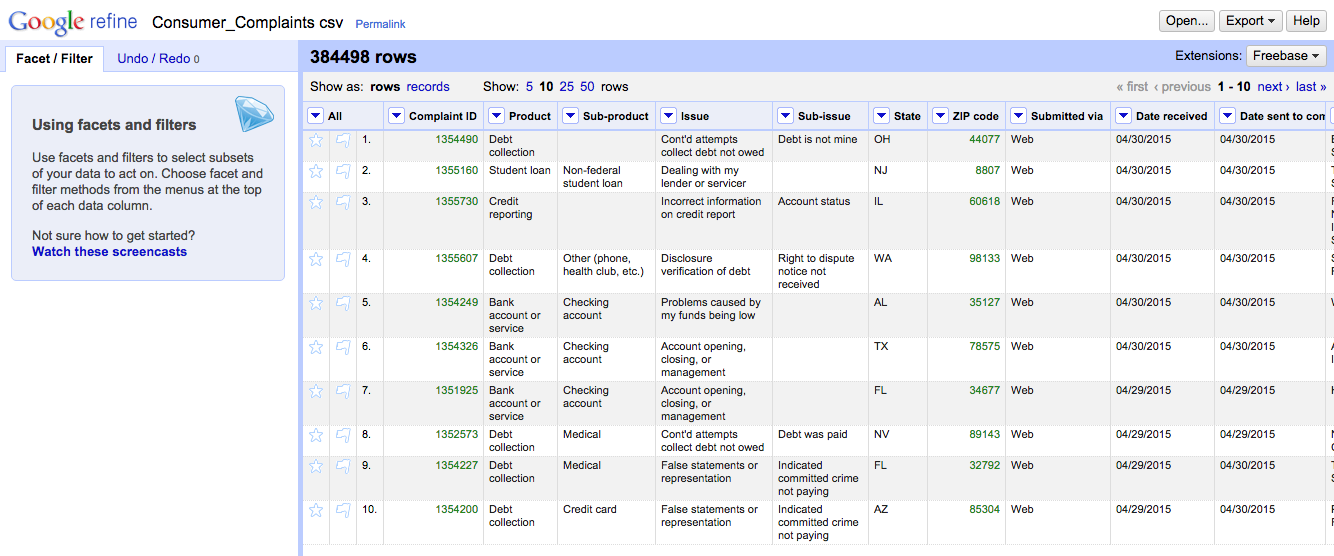


## Creating a project

Creating the project can take a little time.



Once the project is created you can see that it has 384498 rows.



● A1: How many rows are missing value in the state column? Explain how you came up

with the number?

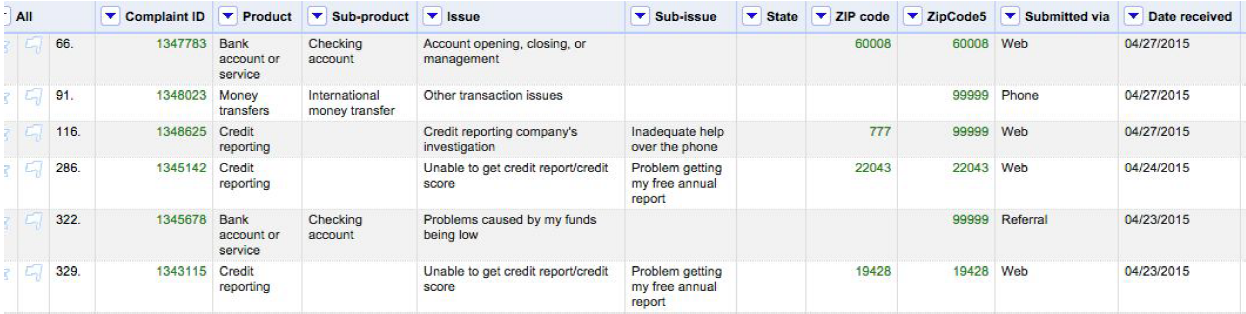
● A2: How many rows with missing zip codes do you have?

● A3: Clean up the zip code column. Create a new column called “ZipCode5” with all zip

codes that contains 5 digits preserved. All other rows should have the zip code 99999.

You should have the same type for all cells in the created column.

Example of result:



● A4: If you consider all zip codes less than 99999 valid zip codes. How many valid and

invalid zip codes do you have respectively.

# Step-2. Cleaning up eq2015 Data.

Upload the data

● A5: For column “nst” fill in missing values.

● A6: Clean up the place column so that it has state or country name depending on what is

in the text.

● A7: From the column “updated” extract the Date without time into a new column called

“eventdate”

● A8: Run cluster en edit on “location” column. Run nearest neighbor and levenshtein

distance. Answer the following questions:

○ Does it make sense to merge detected values?

○ Why or why not?

● A9: Try to do nearest neighbor clustering on “place’ column.

○ What happens?

○ Explain why it is happening.

Step-3