Preparing Data - Cleaning Data with OpenRefine

In this video I'm going to show you how to use OpenRefine and its cool features to clean data sets. Go ahead and go to openrefine.org and click on "Download". You can download the version that applies to your operational system. There is a Windows version, a Mac version and a Linux version. So when you download OpenRefine you can open it, it has its diamond shaped logo. If you're on Windows it's going to be a black screen that's going to open. Don't worry about that. Leave the black screen open, it's important to run the application.

So I'm gonna be using Sarah Cohen's great tutorial about how to clean a data set using OpenRefine. We use the same data set on the cloud data prep tool and we're gonna be using a similar but different approach to clean this data set with OpenRefine. And remember, this is the data set that Sarah, that at the time she was a New York Times, and she gave this presentation at the computer assisted, Computer Assisted Reporting Conference at the NICAR conference in Denver in 2016. I watched a class and it was a great class, a great workshop. And she built this data set out of Medicaid plans in the state of New York and trying to decide which company to do the investigation on.

And she built, like every sheet on the spreadsheet was a month and then she built like this big, big, big spreadsheet with all of the files together in one. And then she used OpenRefined to clean it, especially regular expressions. We're gonna be using a couple today but we're not going to be using all the time.

So but this is, this is the spreadsheet and what we're really want is to get like the, a column with the plan names, the county's total enrollment, a month column and a year column. That's something that looks like this. Plan, county, enrollment, month and year. And to do that, we're gonna use OpenRefine. We can also check Sarah's tutorial here, it's pretty cool. It has some of the more advanced features of OpenRefine. We're gonna get familiar with this.

This is the interface so, when you open OpenRefine, OpenRefine runs on your computer. It's not running on the web. Everything that you do stays on your computer. You don't have to be connected to, connected to the Internet to use OpenRefine. Although it uses your web browser, but notice that it uses a different address. So this address points to your local machines. Everything is running locally and you notice that when you open OpenRefine, it has an interface that allows you to create a project so you can choose from a variety of sources here, or even Google data. The clipboard you can paste data, you can put URLs, you can get files from a computer, you can open existing projects, you can import projects from other files. Also there are like languages settings.

So we're gonna go ahead and create a project. This project we created from the Excel file that Sarah created. So we're going to choose the file here. It's data from this computer and OpenRefine opens almost all database files formats there is. So you don't have to worry about it. DSV, CSV, SV files, Excel, JSON, XML, RDF files and even Google data documents are supported. So it's pretty extensive, so give it a try.

So we're gonna open this. Then I click next and then it's going to give me a preview of everything here. I'm going to just give a name for this project, "New York Medicaid plans". And, it gives you a preview so that you can take a look at if everything is all right, if the

encoding is correct. It asks you, it asks you here to "parse data as" an Excel file so you identified already that it was an Excel file. So we're gonna leave it like that. There are order formats here. The thing that we're going to uncheck here is to parse the first line as a column headers. Sometimes that is the case but this is not the case here. As we see the first row is not the header row. So, we're not going to parse the first line as a header row and then we'll create columns here.

So everything looks good. We're going to create the project now and get familiar with the interface on OpenRefine. So everything you do in OpenRefine happens in this interface. You have here on the left a filter and a facet menu, I'll explain that in a bit. And then here at the right, you always have here in bold the number of records that you're working on at the moment. Right.

You can also switch to rows and records. We're gonna be using rows. Those are rows that you usually see in files like this. So, for example if we see here, that's 6664 that is the same number of rows that we're seeing here. You also see the number of rows is gonna be showing when I put in 50. You can open projects from here, we can at any time export your project anything that you're seeing here. That's something else, it can be an Excel file, an HTML table, or a CSV file or whatever. And you can also navigate the file here, next, first and last. And all of the options in OpenRefine, they are here on top of the columns so anything that you need to do you will do to a specific column, or you can also apply to all of the columns if you need that.

So the first thing that we're gonna do here, since we want to transform this messy, messy data set into a tidy data set like this, the first thing that we're gonna do is to create a year column here. And to create the year column we're going to need to extract this number here from this column and then add it to a new column. So, the way we're gonna do this is we're going to click here on this arrow. We're going to click on "edit column". And then "add column based on this column". And then there's a new window where you can do transformations here. So, and we're going to we're going to be using regular expression. So I'm going to name this new column "year" and here I'm gonna just type "find" so that it will find this thing that I want here. So what I'm using here basically is to, starting the regular expression with the forward slash and ending the regular expression with a forward slash. And then what I'm writing in here is like find me. The four digit number that is in the value, and then return it to me which is this.

If we delete this it gives us a list. In computer programming usually the first item of a list is the position zeros or it starts with zero. So I'm just saying give me the position zero of the list of results that you're showing. So this is like just a regular expression to extract the year from this column and then I click "ok" and then it creates a new column with the year.

Now I want to extract a month to here but the month is a little bit more complicated. The month will be "edit column", "add column based on this column", and what I'm going to use here it's a little bit more complicated but it's it's fairly easy to understand. So I also start with the forward slash and I end with the forward slash. Look here, it is NYS, it is a word and then digits so NYS, a word and then digits. What I added here in between, which is these brackets, and then I have a comma and a space, brackets, comma and then a space, that I'm telling I'm telling OpenRefine to say look, find NYS and then I don't care if after NYS comes a comma or a space any number of times. This is what the plus means.

So I don't care if there is a comma here, or a space, or a double space because of typos in this dataset you can get all of these cases. So it doesn't really matter if it's a comma or a space any number of times. Get me the word that's happening here, right in the middle. That's why I'm putting this in parentheses. This is to capture a group. And I put it here so that we can understand this expression better. Right.

So we, delete this to understand what's going on here. So what's going on here is that I'm capturing here in green just the word January, right. Parentheses in regular expression means to capture whatever is inside. So I'm capturing the word that's happening between this space or comma that comes before four digits. That happens after a space or a comma after the NYS. So I'm getting the word in the middle. That's what I'm doing here in OpenRefine and then it captures here, January. Which is exactly what I need and I'm going to rename this column "month".

Now I need to fill down all of the values here because it is, it found January in 2009. But I also have 2010 here. If I go to the last you see that there are other years here, 2009 November. And it keeps going to other months and years as well. As you can see, here December 2013. So we need to feel down this so that it gets all of the values from month and year going down. So we're gonna do that and it's very, very easy to do that in OpenRefine. So just click here in month, for example. "Edit cells" and then you select "Fill down", that's it. Then you do the same here. "Edit cells", and then you select "fill down" and the same happened for a year.

Now we're going to start some cleaning and to do that we're gonna do some filtering which is pretty cool because then you can delete based on the filtering. So the first thing that we're gonna do is to just to delete the rows that have total here. So we're gonna create here a text filter then we're going to type "total". And then it'll show me here all of the rows, 74, that have the word total. And then we're gonna click here in "all", and then "edit rows" now "remove all matching rows" and then remove seventy four rows. Then we clear the filter and it shows back.

So we're gonna do this a couple of times more so that, to have a better cleaning so we're going to remove blank cells for example so we're gonna do "facets", "customized facets", and then "facet by blank". So facets are ways to identify categories inside of your column. So by creating a facet you're trying to identify cells that have the same value so we're going to group them and then you can visualize the categories. So we're going to, we're gonna do by blank and then we're going to do by text so that you can see better what I mean.

So I'm going to create a "facet by blank" which means a null or empty string and then it tells me that 842 rows in here they are blank meaning that they are empty or there no end value. So you're gonna click here and it gives me like the 842 matching rows and I want to delete all of these rows because they all, they're not required for the final datasets. I'm just gonna remove all of these rows and I'm going to clear the facet here.

Again I have the county and enrollment here. I'm going to go ahead and create filters. To remove that and enrollment. I'm going to remove all matching rows here. And now, what I want to do is also to fill down the names of the plans, right? Because they all belong to the same, the same plans here. So what I'm gonna do is "edit cells", also fill down. Here, and remove the total, here from this column too. All right. Remove all matching rows. It's

looking much much better but still there are some we can do facet by blank here as well. See that. You can remove all of those. And then clear this.

All right so it's looking much, much, much better. Yeah so we can rename this column now "edit column". Name this column, wanted to call "Plans". This one we're gonna rename for "county". And this one "total". Now another thing that you could do is to use a very very powerful. So we can create a text facet here to see what the plans are and then we can categorize them. We can also here see the counties creating a text facet to see what counties are listed here and how many times the counties are listed.

So you can see like New York counties the one that is listed the the highest amount of times. You can also do that. For plans, so for plans, you can check whether the plans are spelled correctly or you can, you know, create like. See here "Eddy Seniorcare" senior care that is together and this is separated. There is a very powerful function in OpenRefine called cluster. And when you click on cluster it tries to identify spelling mistakes and uses different methodologies to do that, and I'll let you explore. You can explore the methodologies here.

Also there's methods and key functions but what it does is it tries to find similarities in names that were spelled almost fully almost the same. So Eddy Seniorcare for example this one and you can go and browse this cluster to see what's going on here. And you see which rows its its meaning right. So all of these are at Eddy Seniorcare. They're all tied together, but we're looking for the ones that are very similar. And we want them to correct the spelling. So for example in this one we have Eddy Senior Care and Eddie Seniorcare spell together. We want the one that we want is Eddy Senior Care spelled separately, so we're gonna just merge this one and then "merge selected and re-cluster" and using, so one 171 cells that were edited in this, so in a split second it edited 171 cells fixing the type of mistakes. So you can, you can change here the methodologies and see which one would work better for you but this is also a powerful tool to fix typing mistakes.

So from now if this is looking okay to you can export the table to any format that you want, CSV or Excel file or TSV, so you can even export the project to open in other versions of OpenRefine. So I'm just going to export this as a CSV. And then it asked me to download to the CSV file. Like any file that you would download on Internet. So here it is, the final CSV file. And see it's ready to use. So this is a quick intro to OpenRefine. Go ahead and go to openrefine.org/download and download the version that's best for your operational system and go cleaning. And any doubts that you guys have, just go ahead and post in the forums and we'll see each other there.