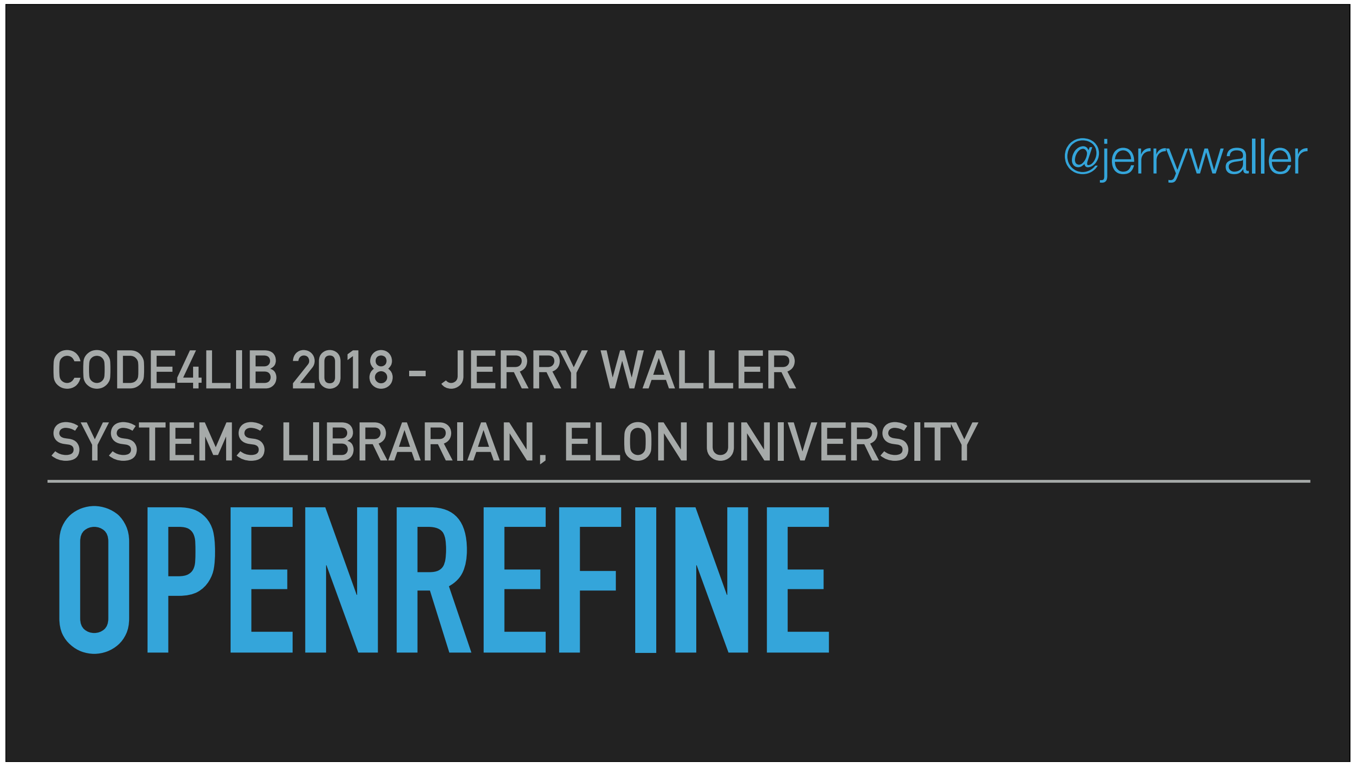
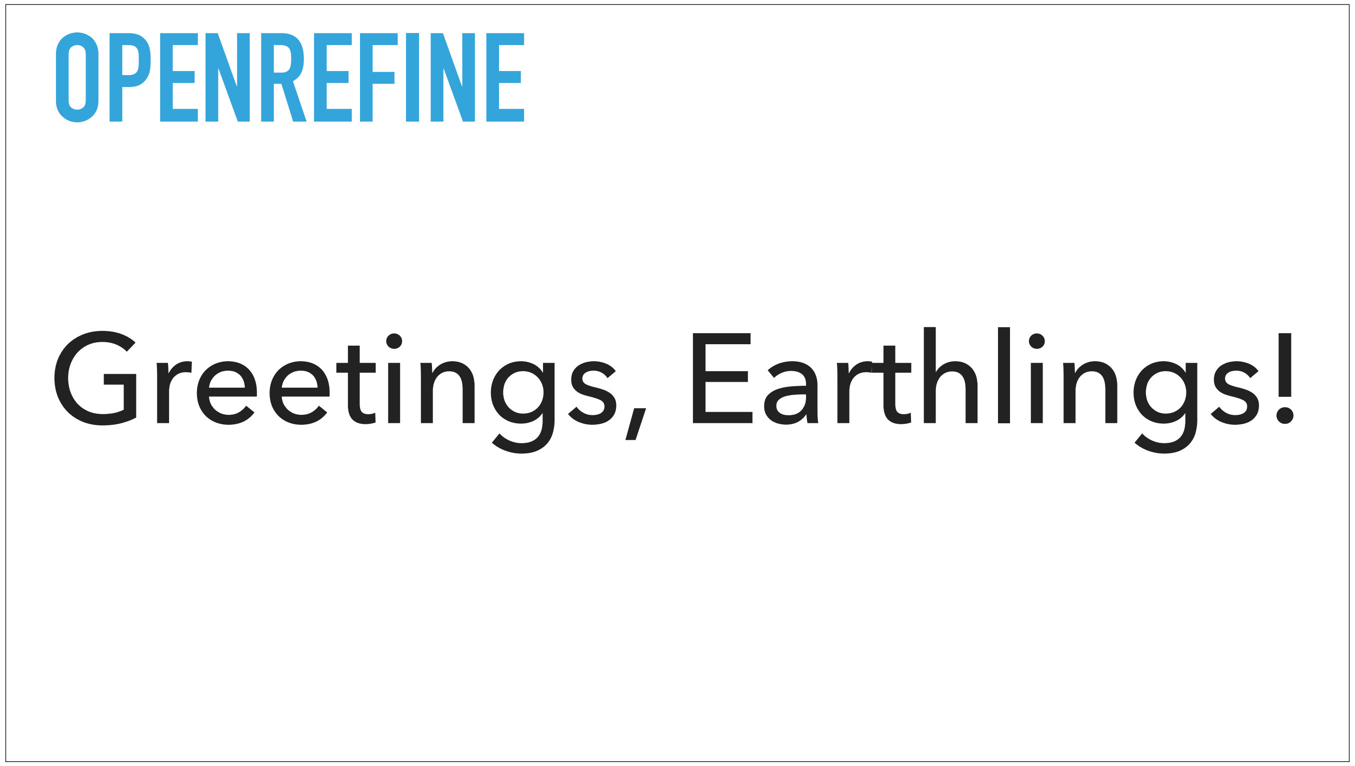


@jerrywaller

CODE4LIB 2018 - JERRY WALLER
SYSTEMS LIBRARIAN, ELON UNIVERSITY

OPENREFINE

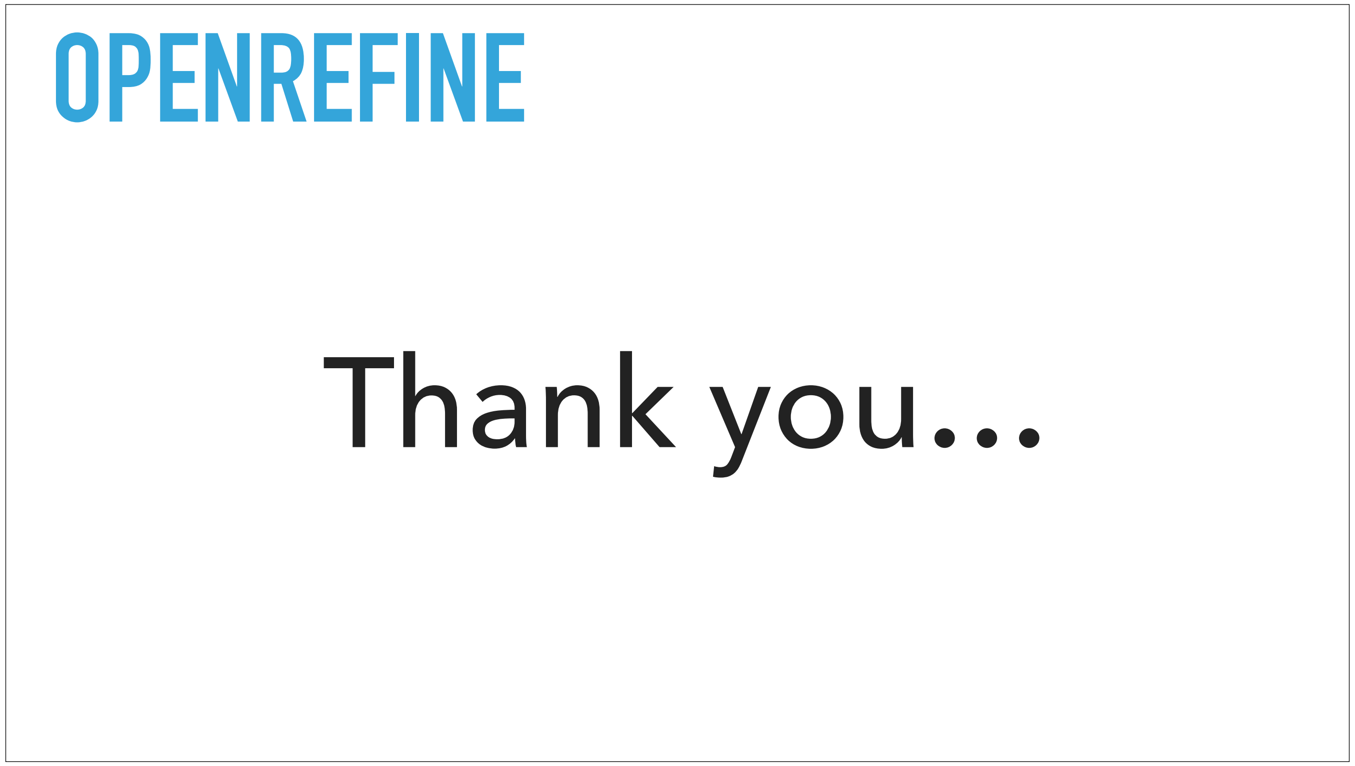
Begin.



OPENREFINE

Greetings, Earthlings!

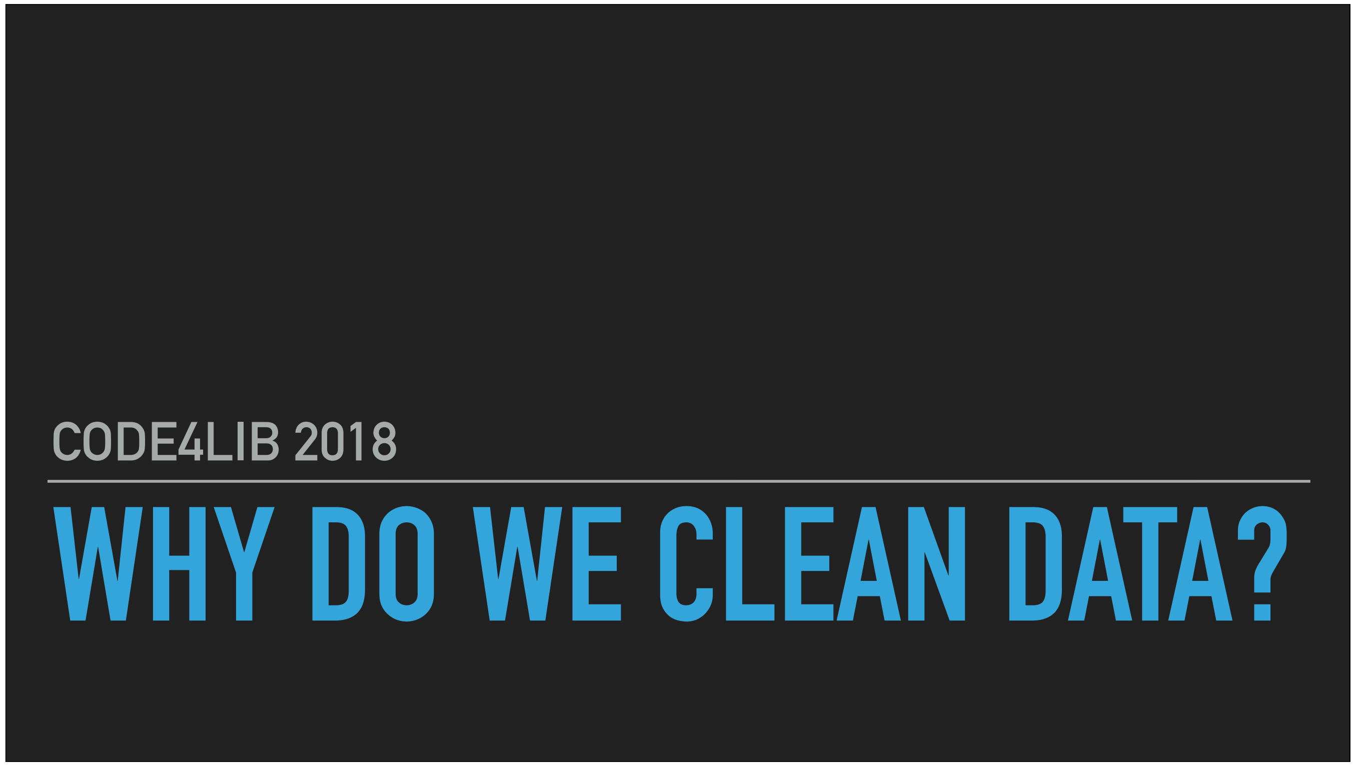
I’m Jerry Waller
Systems Librarian Elon University, and I’d like to…



OPENREFINE

Thank you…

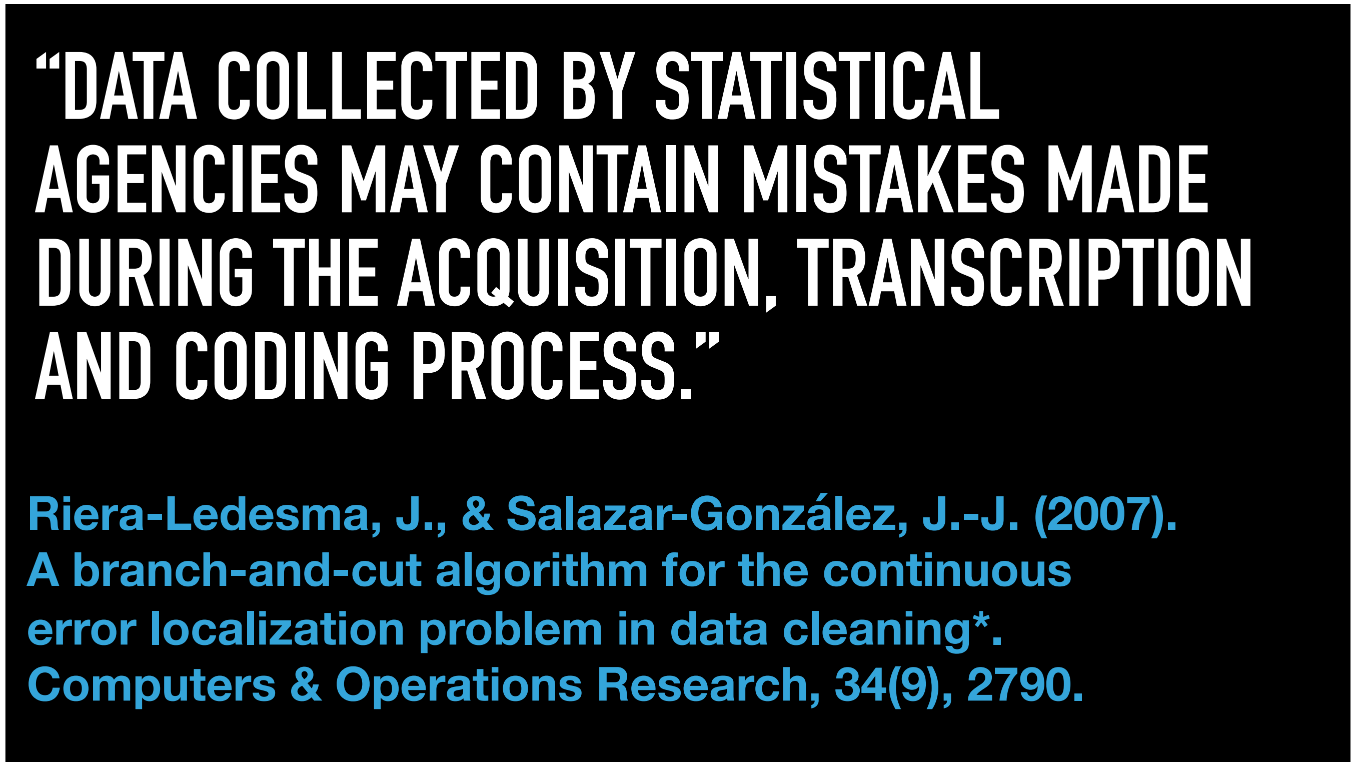
it’s humbling :)



CODE4LIB 2018

WHY DO WE CLEAN DATA?

Begin.



“DATA COLLECTED BY STATISTICAL AGENCIES MAY CONTAIN MISTAKES MADE DURING THE ACQUISITION, TRANSCRIPTION AND CODING PROCESS.”

Riera-Ledesma, J., & Salazar-González, J.-J. (2007).
A branch-and-cut algorithm for the continuous
error localization problem in data cleaning*.
Computers & Operations Research, 34(9), 2790.

Take 7 seconds to read this.
We are neither consistently accurate nor consistently correct when it comes to entering data.
In short, we are imperfect.

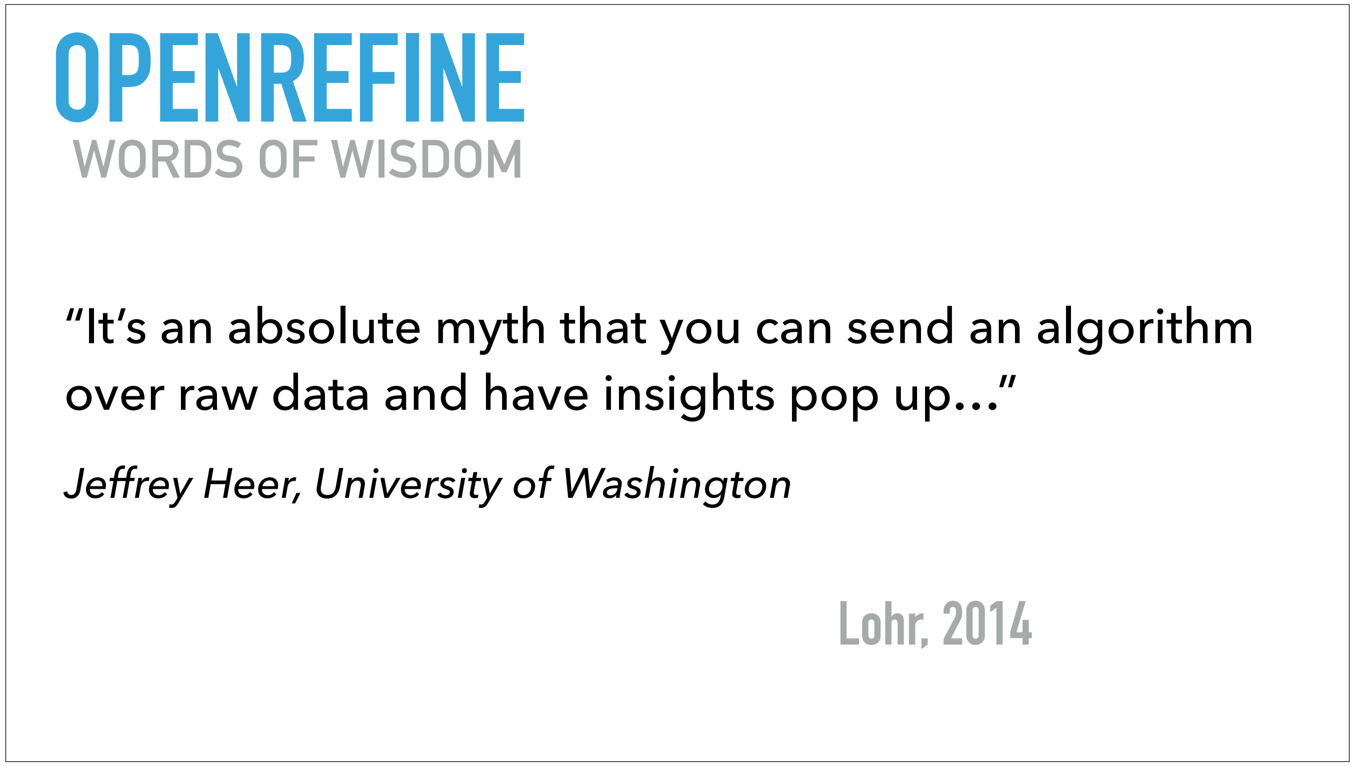
OPENREFINE

Steve Lohr
New York Times
August, 2014

For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights

By STEVE LOHR • AUG. 17, 2014

A sensational headline that underscores the amount of work that goes into data analysis.
I worked retail. I've cleaned my share of toilets.
PPD and biostats. my perspective on this is: Without good data, no good results.
I thought cleaning the data presented its own interesting set of challenges.
The photo features Monica Rogati, Vice President for Data Science at Jawbone



OPENREFINE

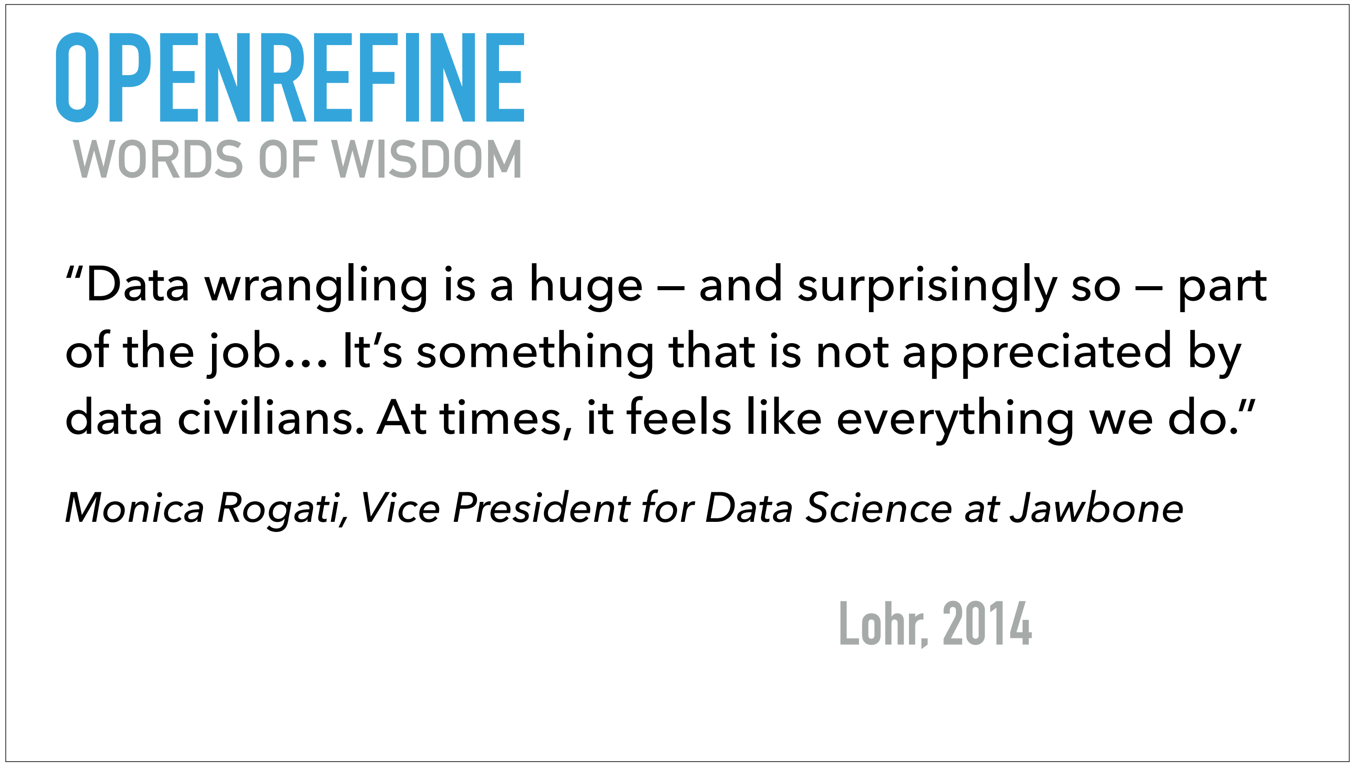
WORDS OF WISDOM

> "It's an absolute myth that you can send an algorithm over raw data and have insights pop up…"

Jeffrey Heer, University of Washington

Lohr, 2014

Hey Siri, clean this data for me.
Unsurprisingly, this will fail.
Turns out cleaning data is a lot of work.



OPENREFINE

WORDS OF WISDOM

“Data wrangling is a huge – and surprisingly so – part of the job… It’s something that is not appreciated by data civilians. At times, it feels like everything we do.”

Monica Rogati, Vice President for Data Science at Jawbone

Lohr, 2014

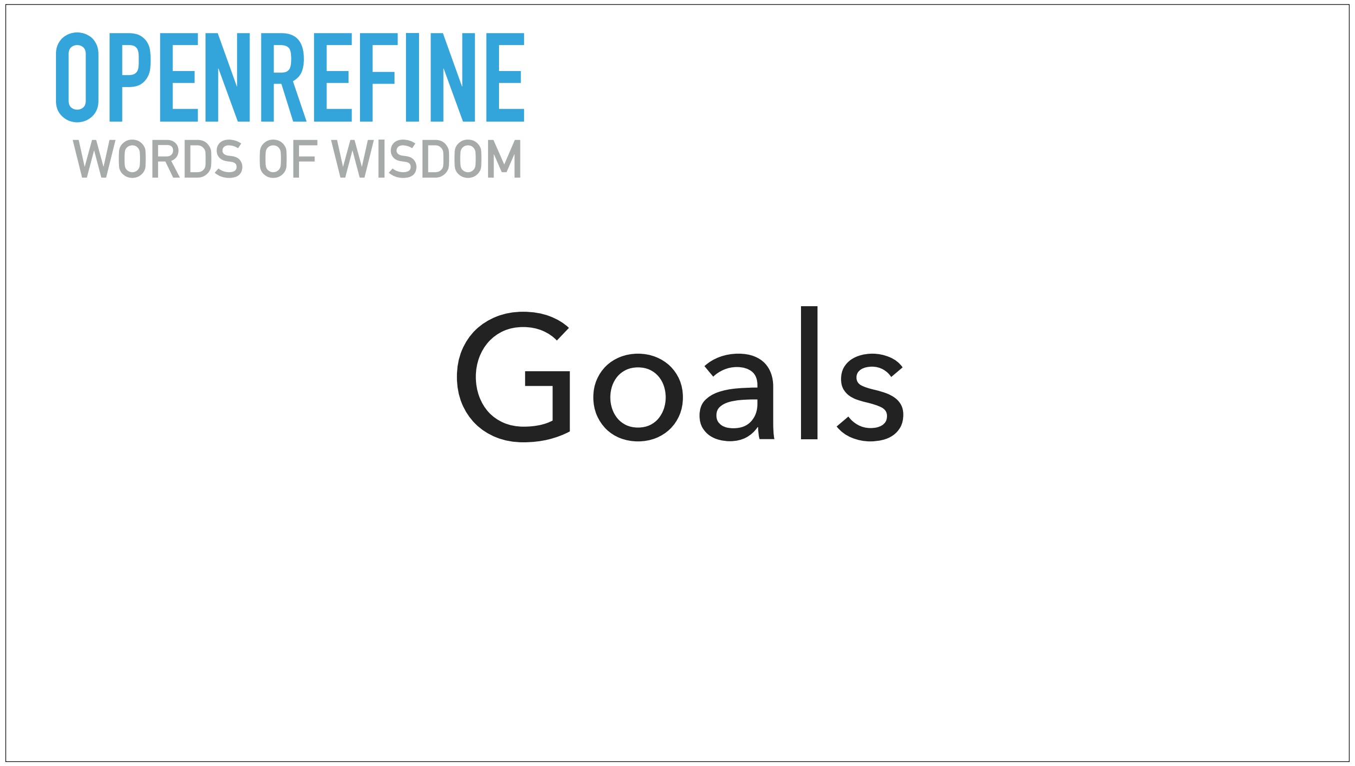
What this boils down to is that:

OPENREFINE

WORDS OF WISDOM

The data conform to a standard.

We need to do the best we can to ensure the data we analyze is as standardized as possible.
Cleaning, wrangling, or munging (whatever term you wish) the data is one of those steps.



OPENREFINE

WORDS OF WISDOM

Goals

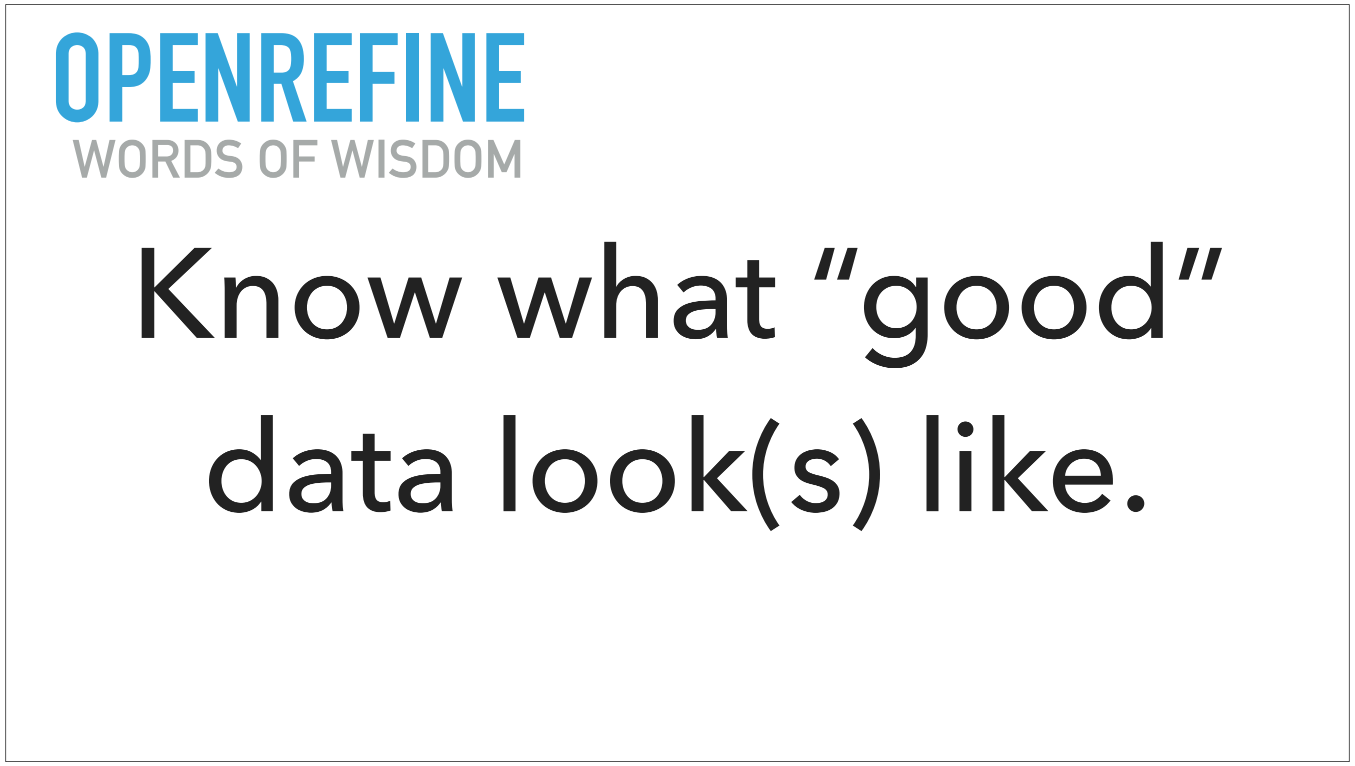
goals of today’s workshop:
foster a mindset of data stewardship.

OPENREFINE

WORDS OF WISDOM

Know your data.

In order to establish or conform to a standard, you must be familiar with your data.



OPENREFINE

WORDS OF WISDOM

Know what “good” data look(s) like.

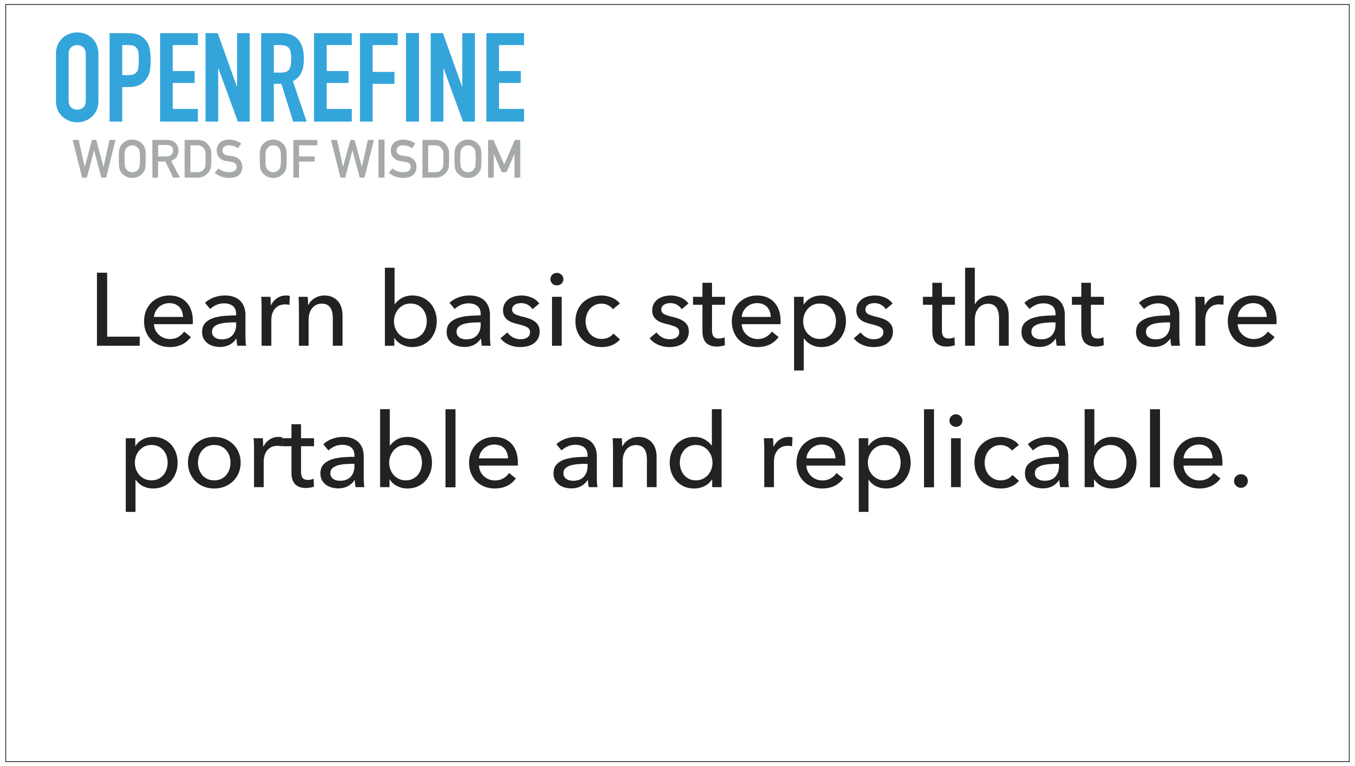
This does not mean memorizing every element

OPENREFINE

WORDS OF WISDOM

Malformed
vs.
Well-formed

what it means is knowing the difference between well-formed and malformed data

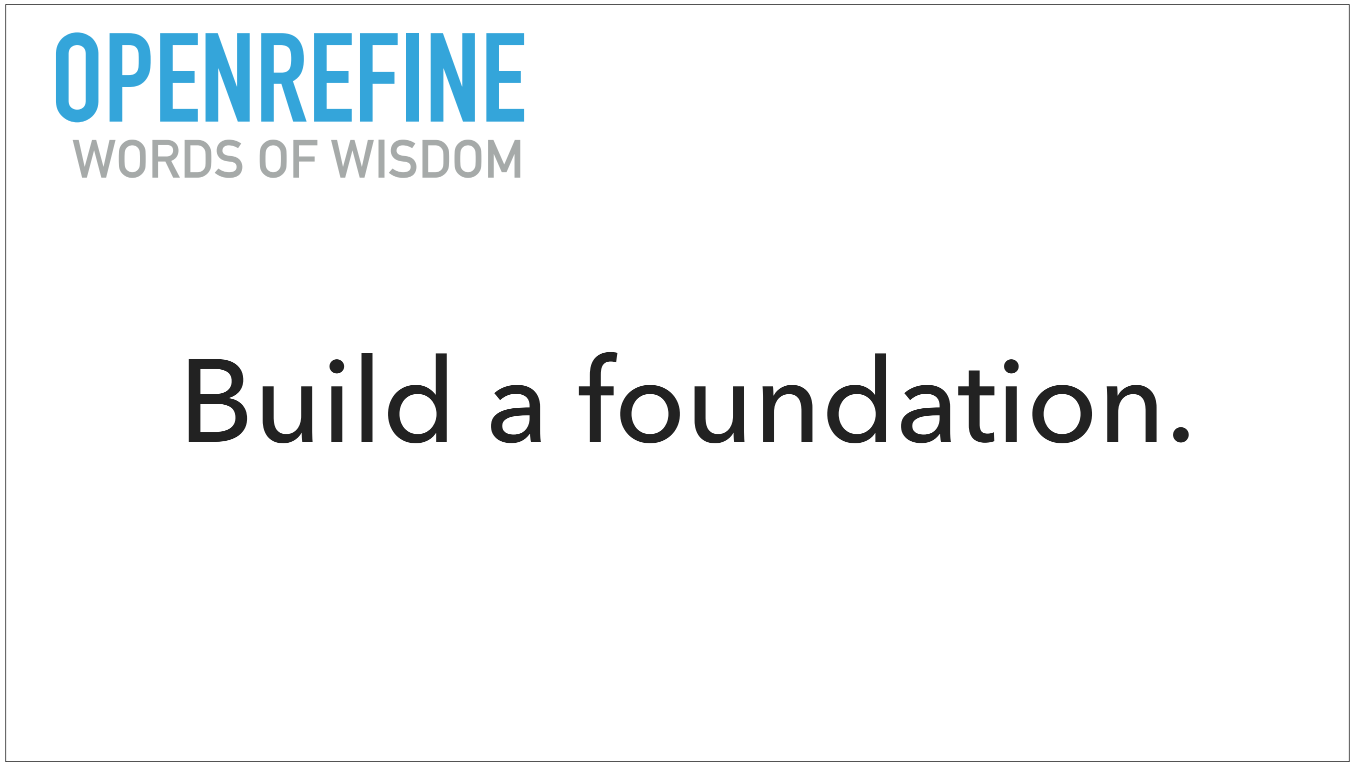


OPENREFINE

WORDS OF WISDOM

Learn basic steps that are portable and replicable.

I’m going to talk through the steps needed to get going in openrefine.
this may not make any sense at first, but we’ll practice.



OPENREFINE

WORDS OF WISDOM

Build a foundation.

It’s my experience that you’re better served learning how to do the basics.
We only have a few hours, and there’s no way to cover everything.
I want you learn enough so that you feel confident to go back to it later.

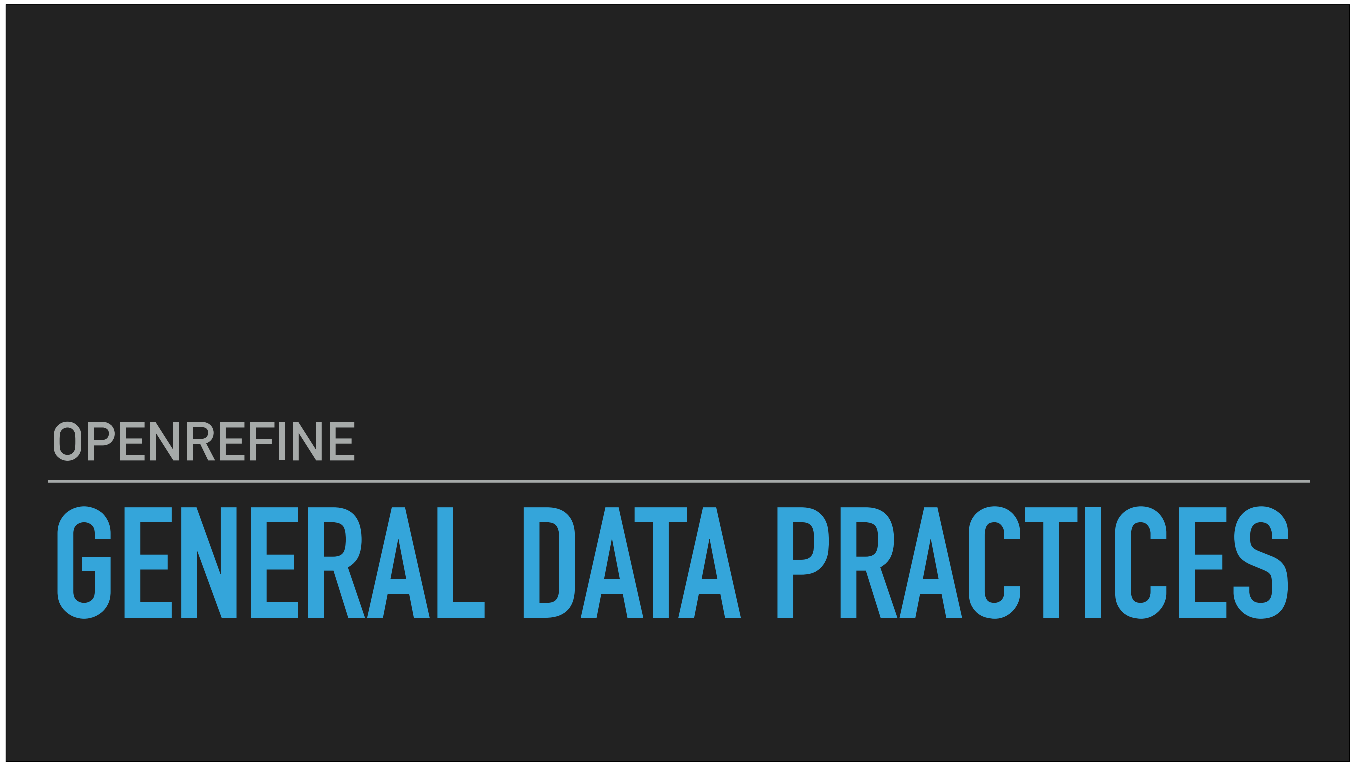
OPENREFINE

WORDS OF WISDOM

- Strategies.
- It’s a *lot* of work.

Fun work! And your boss will love you.

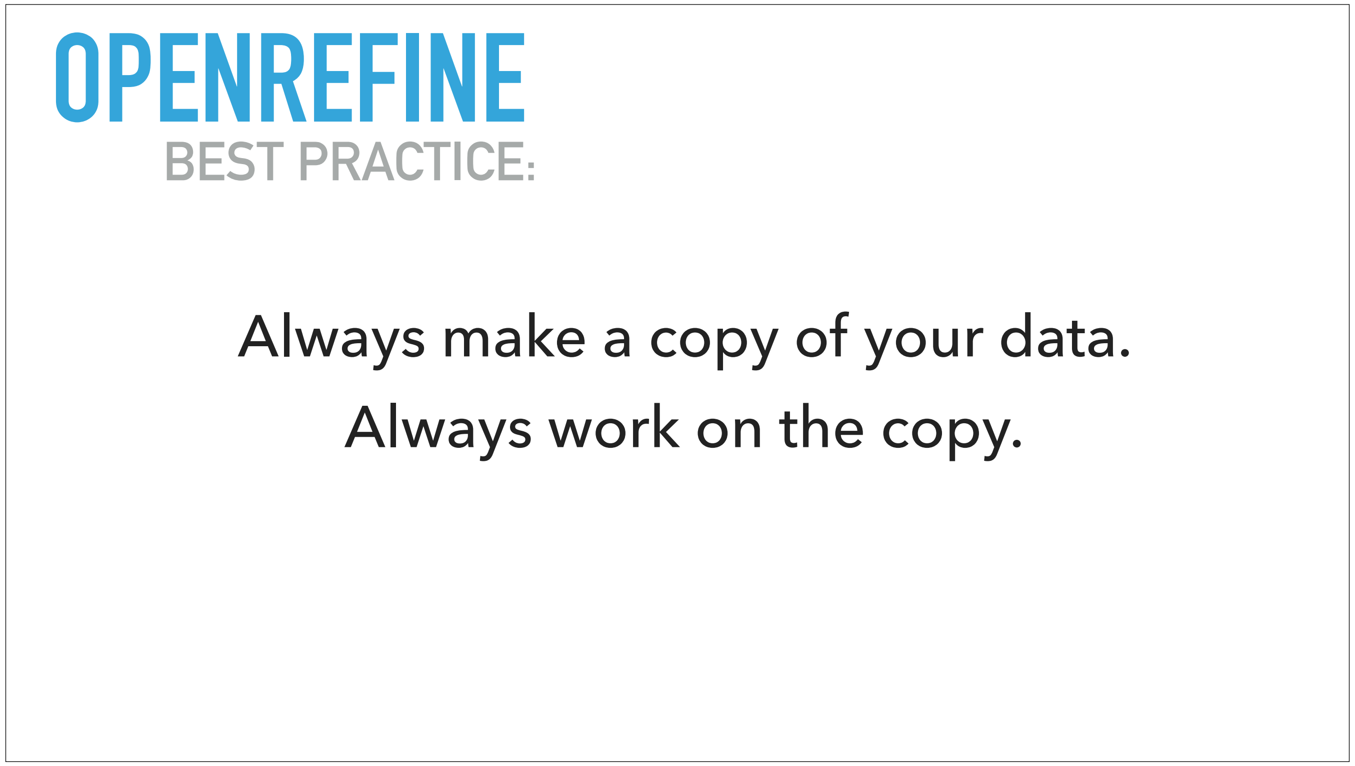
Ideally, I want you to take away a better idea of the strategies you can use and
an understanding of the amount of work required for standardizing and cleaning datasets.



OPENREFINE

GENERAL DATA PRACTICES

As we get started, there are some tips, caveats, and the like I’d like to get out of the way.



OPENREFINE

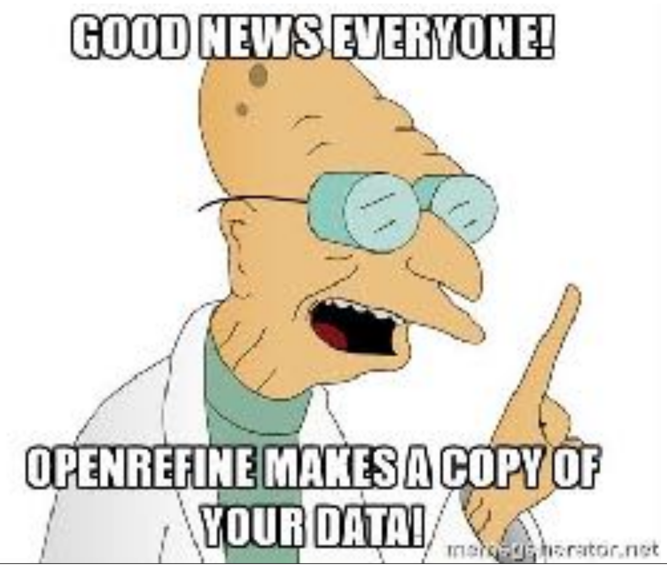
BEST PRACTICE:

Always make a copy of your data.

Always work on the copy.

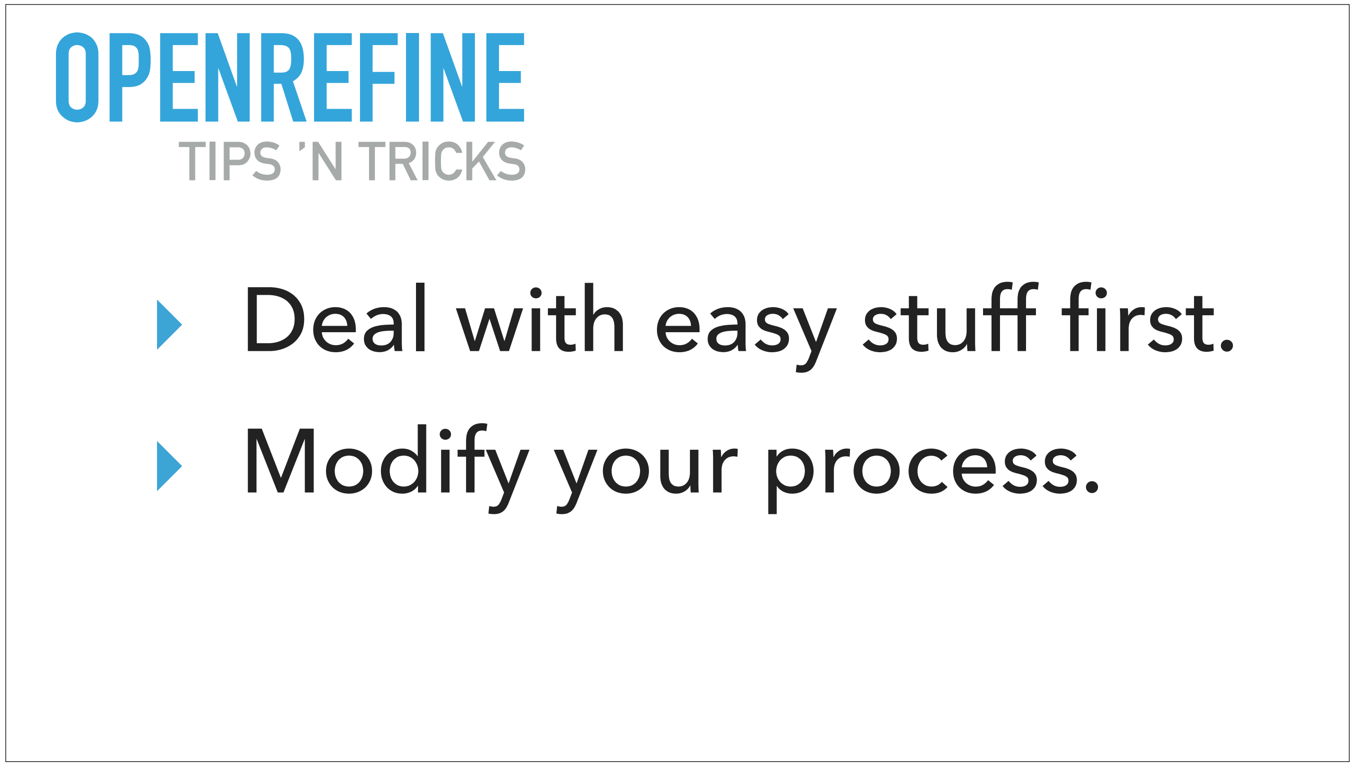
Good news everyone!
OpenRefine makes a copy of the data.

OPENREFINE



Copyright 20th Century Fox

Good news everyone!
OpenRefine makes a copy of the data.

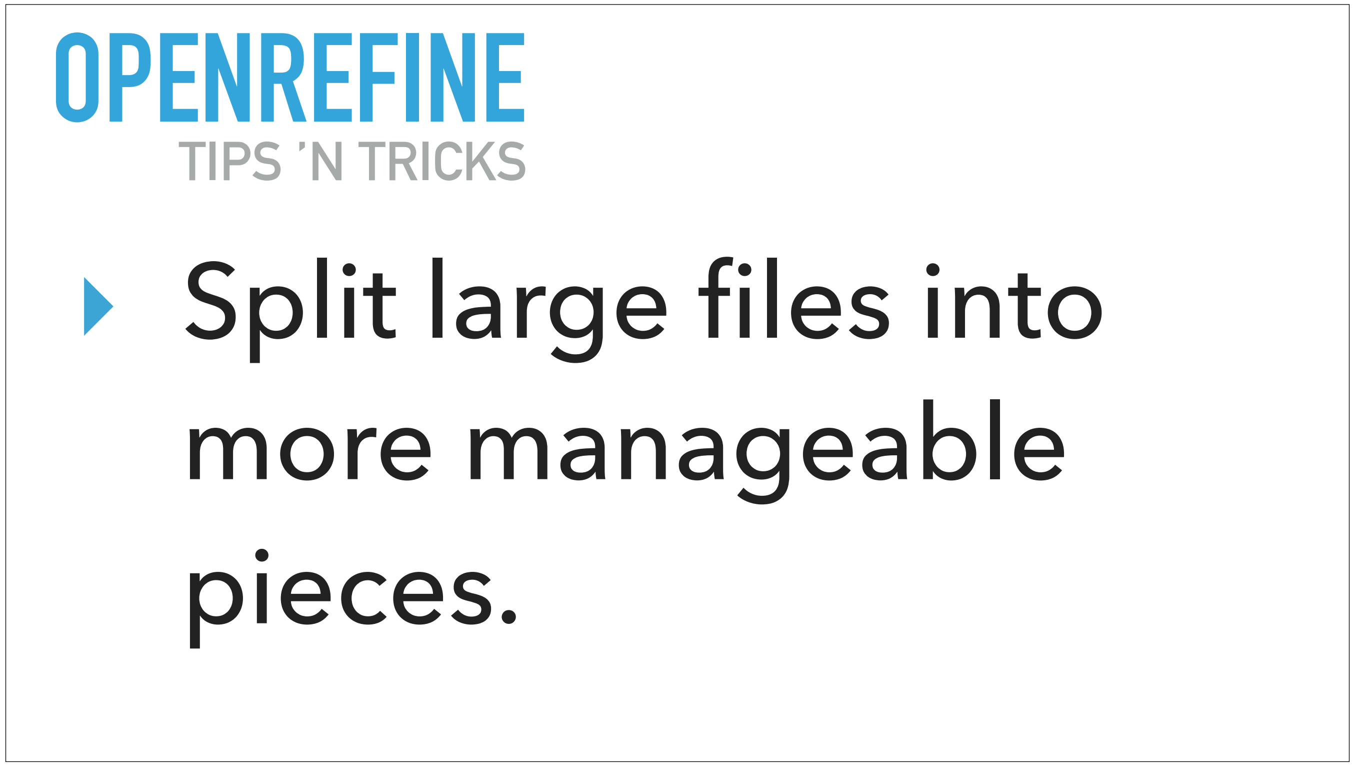


OPENREFINE

TIPS ’N TRICKS

- Deal with easy stuff first.
- Modify your process.

As you iterate through your data, you get to Know Your Data.
Deal with the easier stuff first, then
you can modify your process for more complicated aspects.

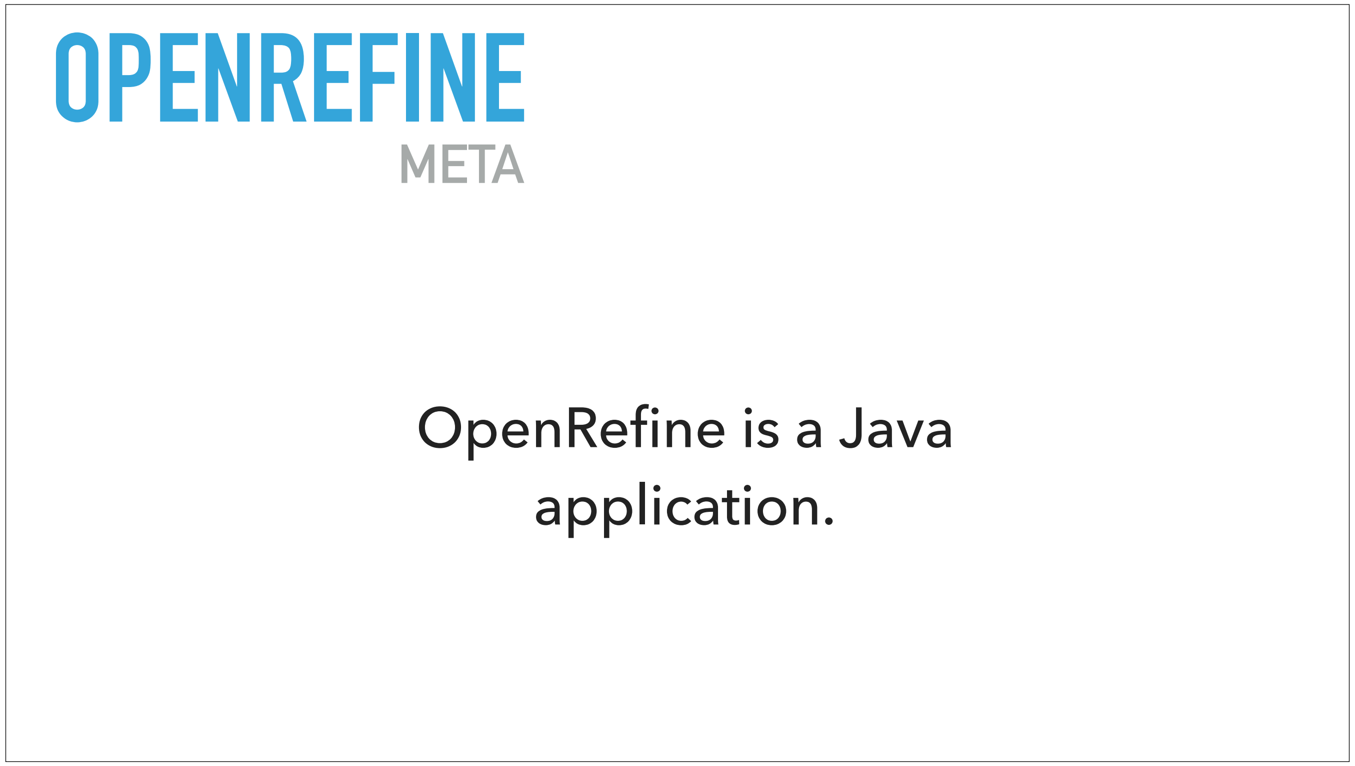


OPENREFINE

TIPS 'N TRICKS

- Split large files into more manageable pieces.

If a large file is slowing down the system…
large files should be split into more manageable pieces.
refine your process with smaller, or even sample datasets.

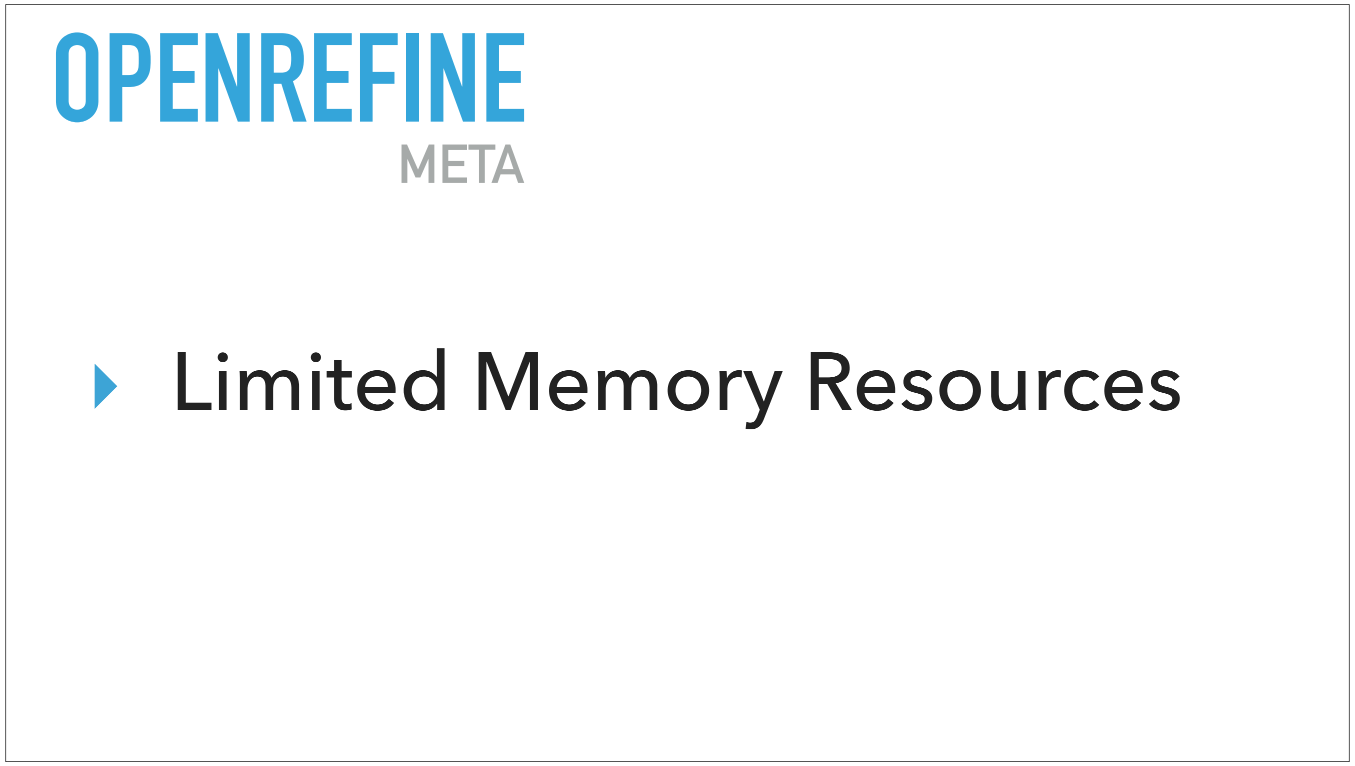


OPENREFINE

META

OpenRefine is a Java application.

That said open refine is a java app. I have a general dislike of Java, but it is what it is.

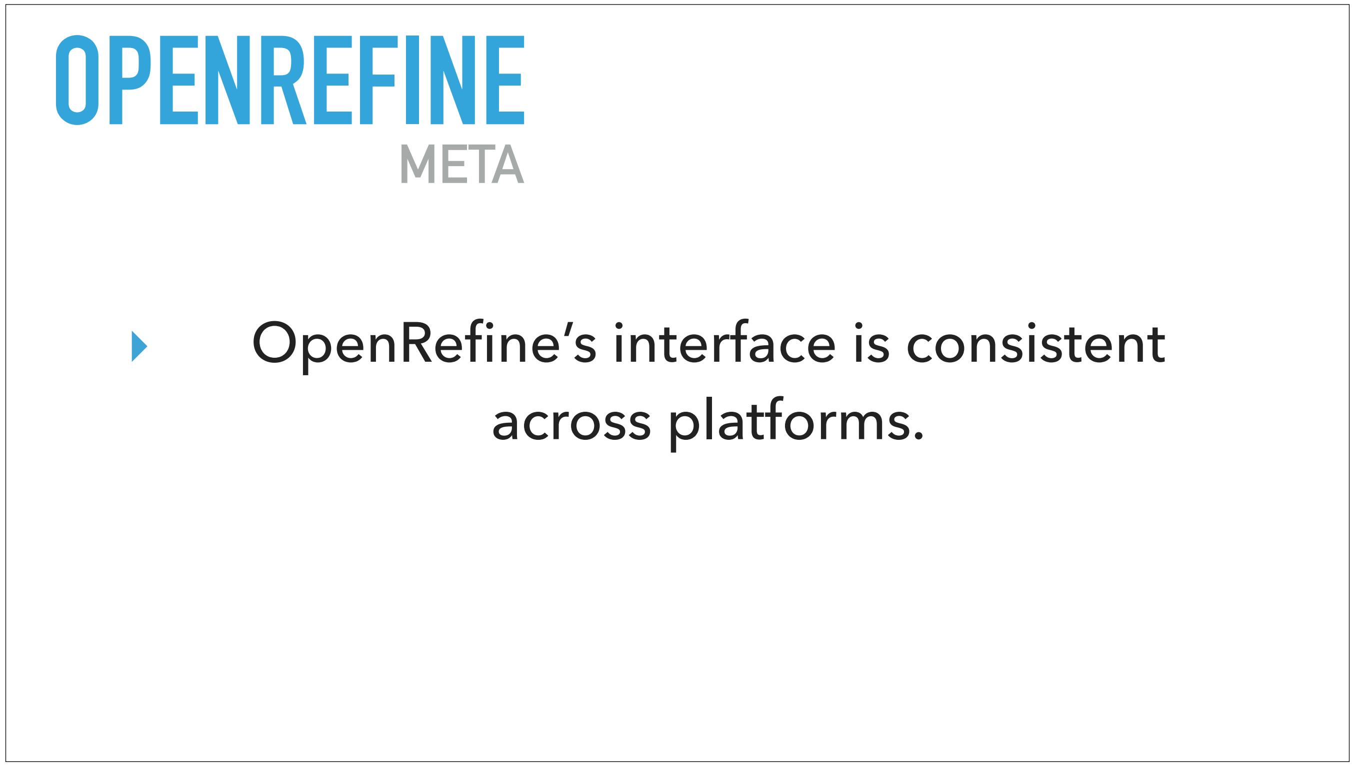


OPENREFINE

META

- Limited Memory Resources

A downside to this java implementation is ^^^
Remember those large files I mentioned a moment ago

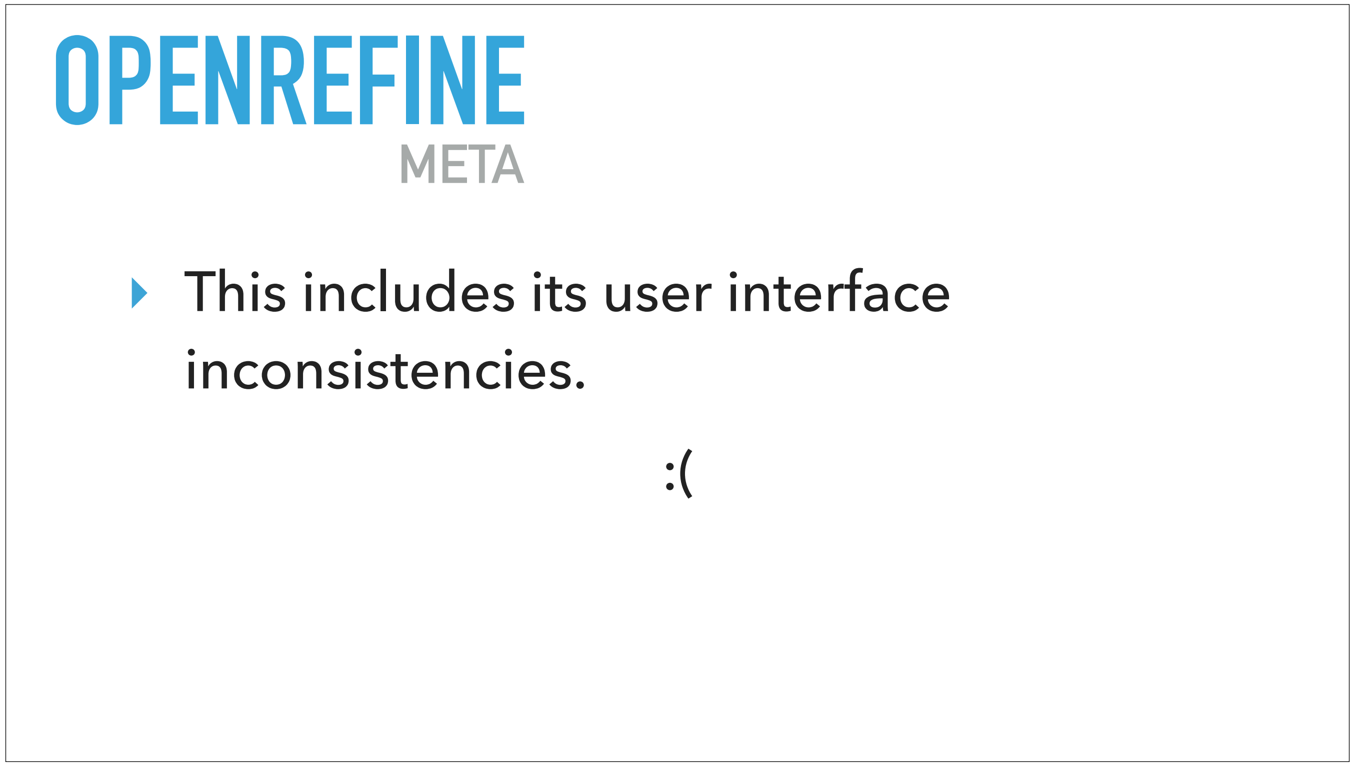


OPENREFINE

META

- OpenRefine’s interface is consistent across platforms.

On the plus side:^^^^
Regardless of what operating system you use, OpenRefine provides a consistent interface. We’ll be toggling back and forth between an interactive session and the slides when it’s best to do so.



OPENREFINE

META

- This includes its user interface inconsistencies.

:(

Unfortunately there are several UX inconsistencies, some of which we will cover.

delete or move?

OPENREFINE

TIPS ’N TRICKS

Iterate.

Wash, rinse, repeat.

OPENREFINE
META

Ready, Player One?

Wash, rinse, repeat.

OPENREFINE

EXERCISE 1

OPENREFINE

EXERCISE 1

Create a project folder for your exercises.

Create the first project.
Discuss the options.

OPENREFINE

EXERCISE 1

OpenRefine's Gemstone Icon

It's a gem! The crown jewel!

OPENREFINE

EXERCISE 1

There’s no place like

127.0.0.1:3333

OPENREFINE

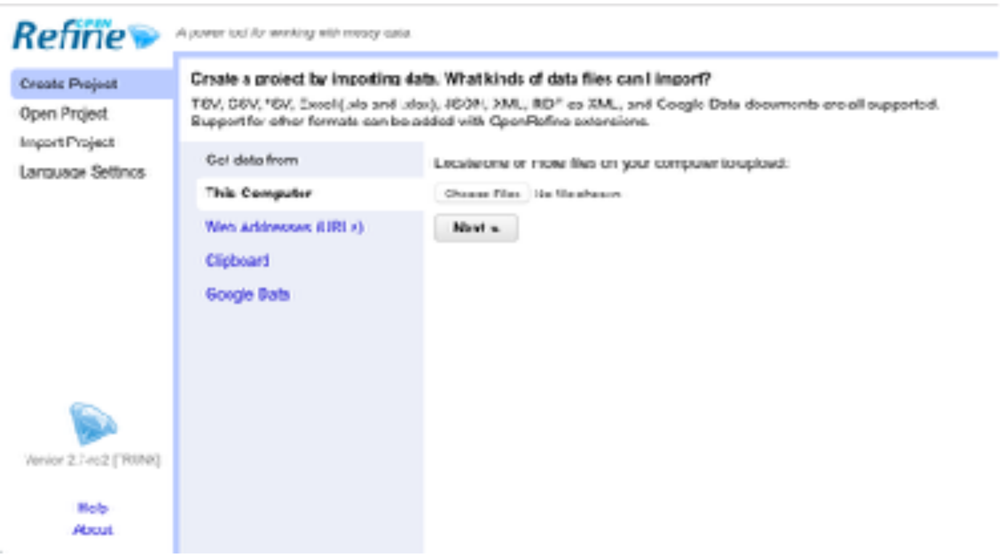
EXERCISE 1

The Interface.

Even small, well-formed datasets can be wrangled.

OPENREFINE

EXERCISE 1



A power tool for working with messy data

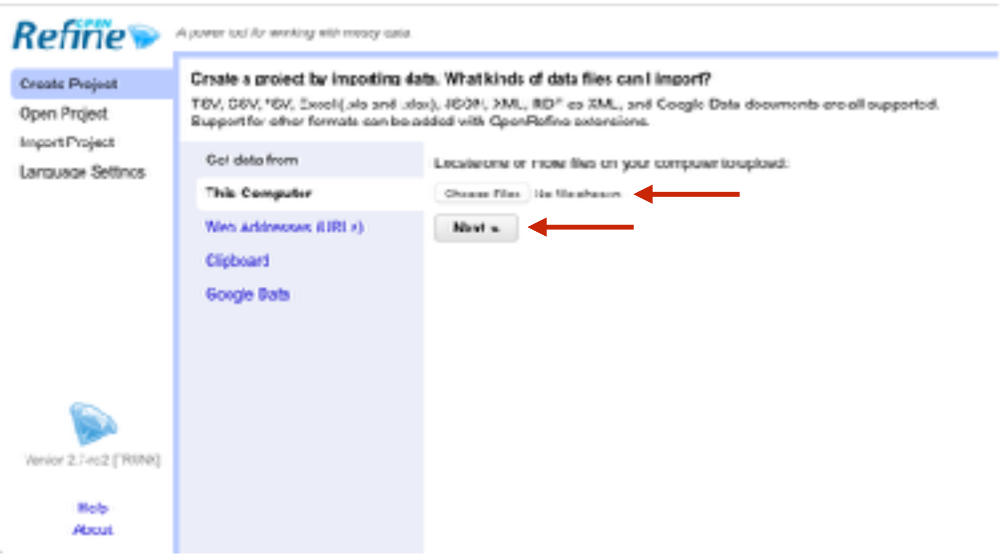
Create a project by importing data. What kinds of data files can I import?

TSV, CSV, *SV, Excel (.xls and .xlsx), JSON, XML, RDF as XML, and Google Data documents are all supported. Support for other formats can be added with OpenRefine extensions.

Locate one or more files on your computer to upload:

This is the opening screen, where you declare the “intent”, whether opening or creating or importing.

OPENREFINE
EXERCISE 1



Create a project by importing data. What kinds of data files can I import?

TSV, CSV, *SV, Excel (.xls and .xlsx), JSON, XML, RDF as XML, and Google Data documents are all supported. Support for other formats can be added with OpenRefine extensions.

Locate one or more files on your computer to upload:

1. Click “Choose Files”3
2. From your project directory select 01-excercise.tsv
3. Click—of course—“Next”

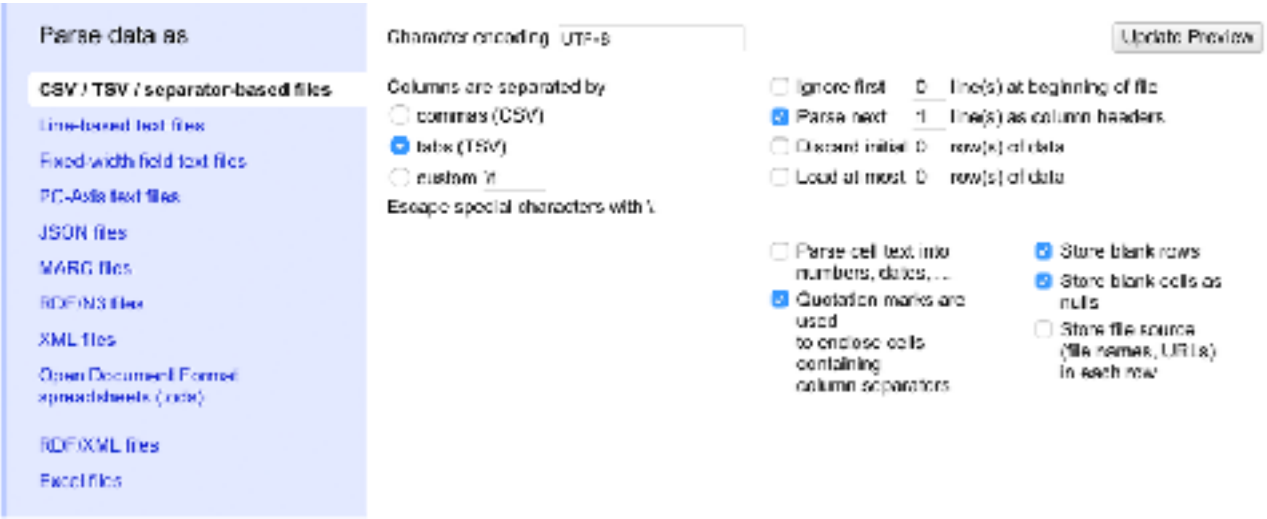
OPENREFINE

EXERCISE 1

1. Click “Choose Files”
2. From your project directory select 01.tsv
3. Click “Next”

OPENREFINE

EXERCISE 1



Parse data as

- CSV / TSV / separator-based files
- Line-based text files
- Fixed-width field text files
- PC-Axis text files
- JSON files
- MARC files
- RDF/N3 files
- XML files
- Open Document Format spreadsheets (.ods)
- RDF/XML files
- Excel files

Character encoding UTF-8

Update Preview

Columns are separated by

○ commas (CSV)

● tabs (TSV)

○ custom \t

Escape special characters with \

□ Ignore first 0 line(s) at beginning of file

☑ Parse next 1 line(s) as column headers

□ Discard initial 0 row(s) of data

□ Load at most 0 row(s) of data

□ Parse cell text into numbers, dates, ...

☑ Quotation marks are used to enclose cells containing column separators

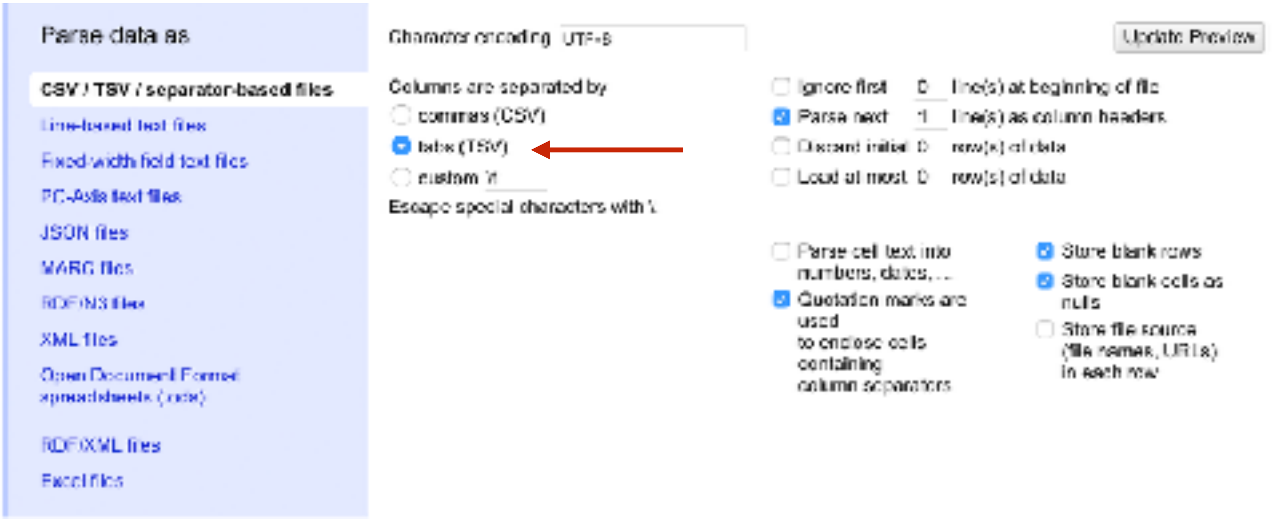
☑ Store blank rows

☑ Store blank cells as nulls

□ Store file source (file names, URLs) in each row

A Note on the “Parse Data As…” column:
Some formats are obvious; they're either explicitly named or formatted in a way that you recognize. In other, less obvious, cases try out other options to see if they provide you with a preview that makes sense to you. You're not going to destroy the original dataset. The worst thing that can happen is that you crash OpenRefine and have to restart it.

OPENREFINE
EXERCISE 1



Parse data as

- **CSV / TSV / separator-based files**
- Line-based text files
- Fixed-width field text files
- PC-Axis text files
- JSON files
- MARC files
- RDF/N3 files
- XML files
- Open Document Format spreadsheets (.ods)
- RDF/XML files
- Excel files

Character encoding UTF-8

Columns are separated by

○ commas (CSV)
● tabs (TSV)
○ custom \t

Escape special characters with \

□ Ignore first 0 line(s) at beginning of file
☑ Parse next 1 line(s) as column headers
□ Discard initial 0 row(s) of data
□ Load at most 0 row(s) of data

□ Parse cell text into numbers, dates, ...
☑ Quotation marks are used to enclose cells containing column separators

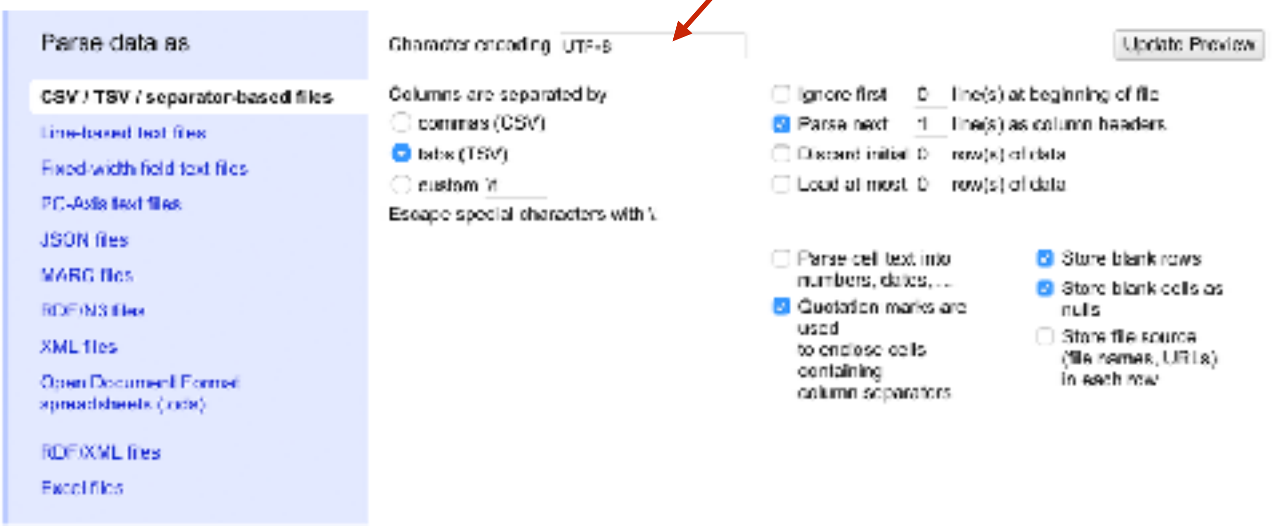
☑ Store blank rows
☑ Store blank cells as nulls
□ Store file source (file names, URLs) in each row

OpenRefine is pretty good at guessing the delimiters. In this case, it has correctly chosen tabs for a tab-delimited file with an extension of .TSV

OPENREFINE

EXERCISE 1

THIS IS NOT A TEXT BOX.
CECI N'EST PAS UNE ~~PIPE~~ BOÎTE DE TEXTE.



Parse data as

- CSV / TSV / separator-based files
- Line-based text files
- Fixed-width field text files
- PC-Axis text files
- JSON files
- MARC files
- RDF/N3 files
- XML files
- Open Document Format spreadsheets (.ods)
- RDF/XML files
- Excel files

Character encoding UTF-8

Update Preview

Columns are separated by
○ commas (CSV)
● tabs (TSV)
○ custom \t
Escape special characters with \

□ Ignore first 0 line(s) at beginning of file
☑ Parse next 1 line(s) as column headers
□ Discard initial 0 row(s) of data
□ Load at most 0 row(s) of data

□ Parse cell text into numbers, dates, ...
☑ Quotation marks are used to enclose cells containing column separators
☑ Store blank rows
☑ Store blank cells as nulls
□ Store file source (file names, URLs) in each row

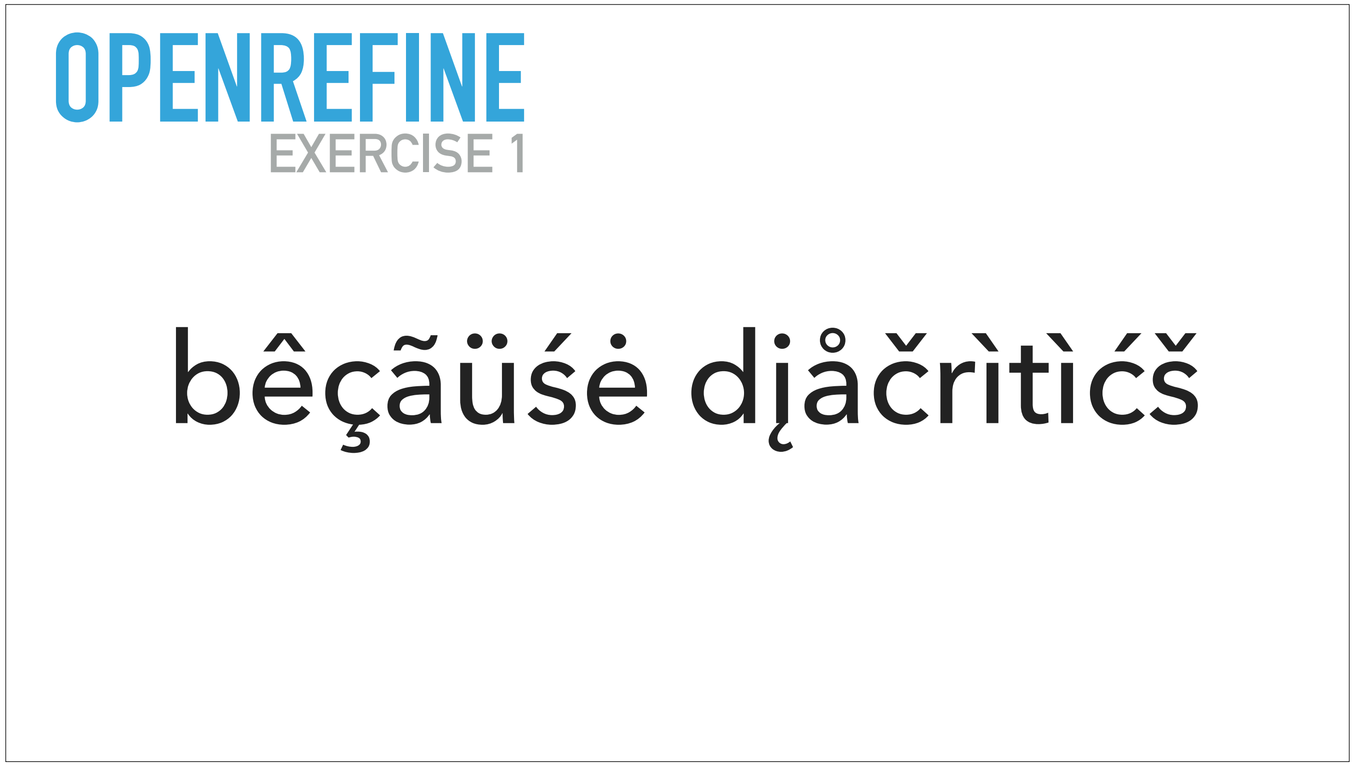
You should almost always click in the “Character Encoding” box (which is really a button) and choose UTF-8.

OPENREFINE

EXERCISE 1

UTF-8

When importing, define your encoding as utf-8
Because diacritics!

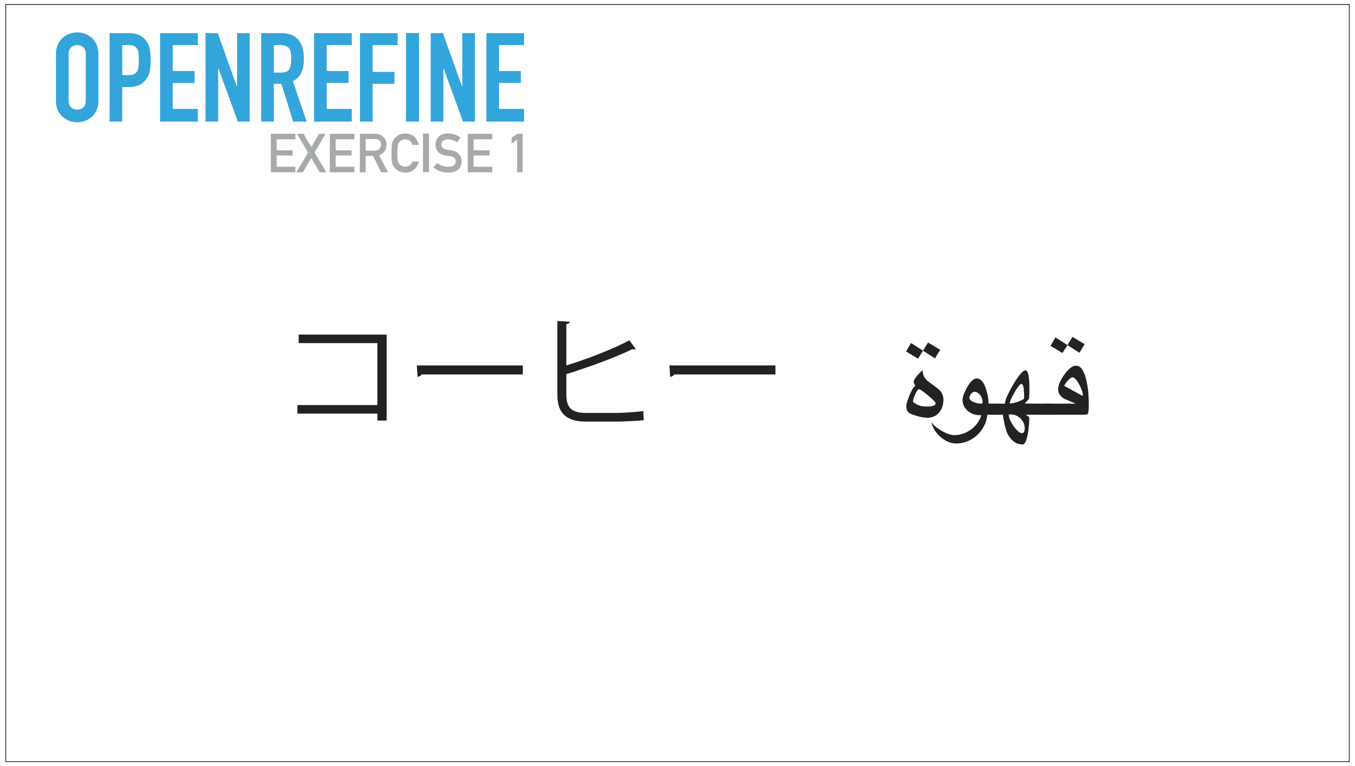


OPENREFINE

EXERCISE 1

bêçãüśė dįåčrìtìćš

Because diacritics! Interdisciplinary, multi-linguistic.



OPENREFINE

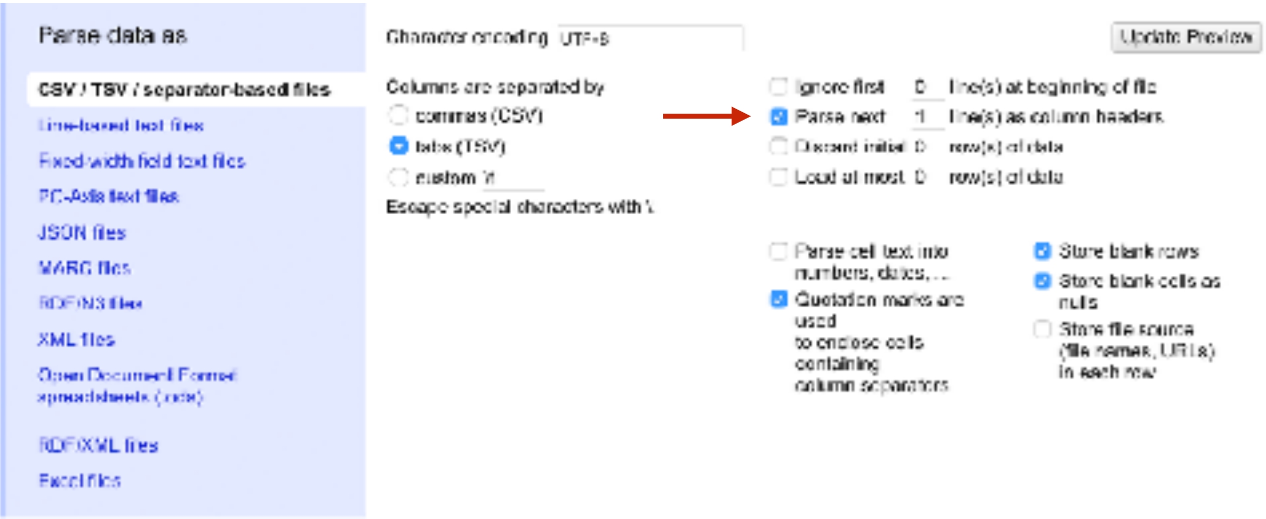
EXERCISE 1

コーヒー　قهوة

Because non-Roman characters

OPENREFINE

EXERCISE 1



Parse data as

- CSV / TSV / separator-based files
- Line-based text files
- Fixed-width field text files
- PC-Axis text files
- JSON files
- MARC files
- RDF/N3 files
- XML files
- Open Document Format spreadsheets (.ods)
- RDF/XML files
- Excel files

Character encoding UTF-8

Columns are separated by

○ commas (CSV)

● tabs (TSV)

○ custom \t

Escape special characters with \

□ Ignore first 0 line(s) at beginning of file

☑ Parse next 1 line(s) as column headers

□ Discard initial 0 row(s) of data

□ Load at most 0 row(s) of data

□ Parse cell text into numbers, dates, ...

☑ Quotation marks are used to enclose cells containing column separators

☑ Store blank rows

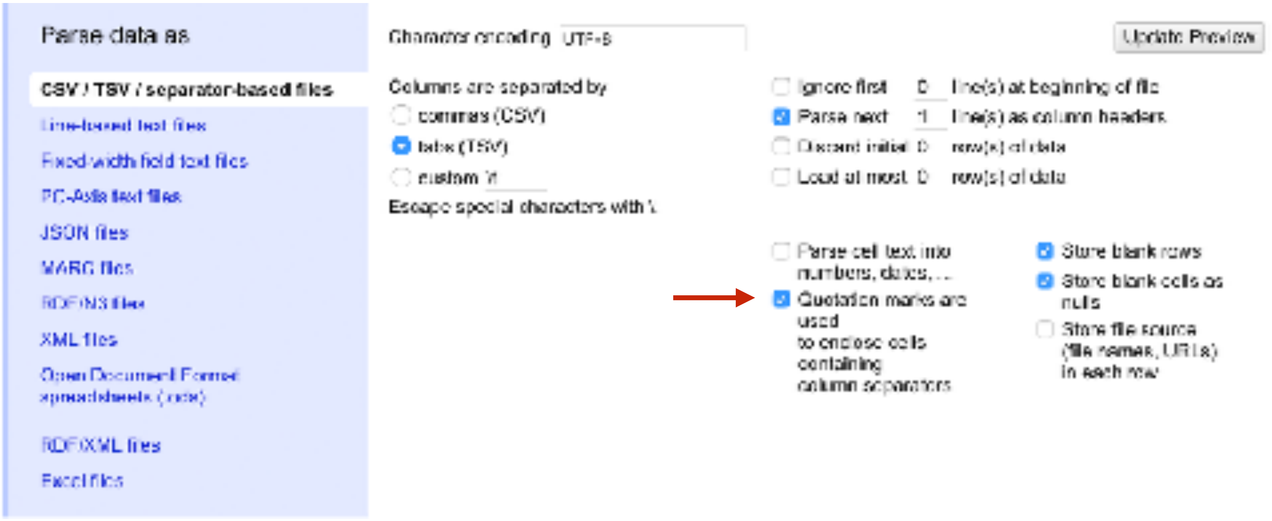
☑ Store blank cells as nulls

□ Store file source (file names, URLs) in each row

OpenRefine typically defaults to parsing the first line as column headers, which will work in Exercise 1. This will not always be the case.

OPENREFINE

EXERCISE 1



Character encoding UTF-8

Columns are separated by
○ commas (CSV)
● tabs (TSV)
○ custom \t
Escape special characters with \

☐ Ignore first 0 line(s) at beginning of file
☑ Parse next 1 line(s) as column headers
☐ Discard initial 0 row(s) of data
☐ Load at most 0 row(s) of data

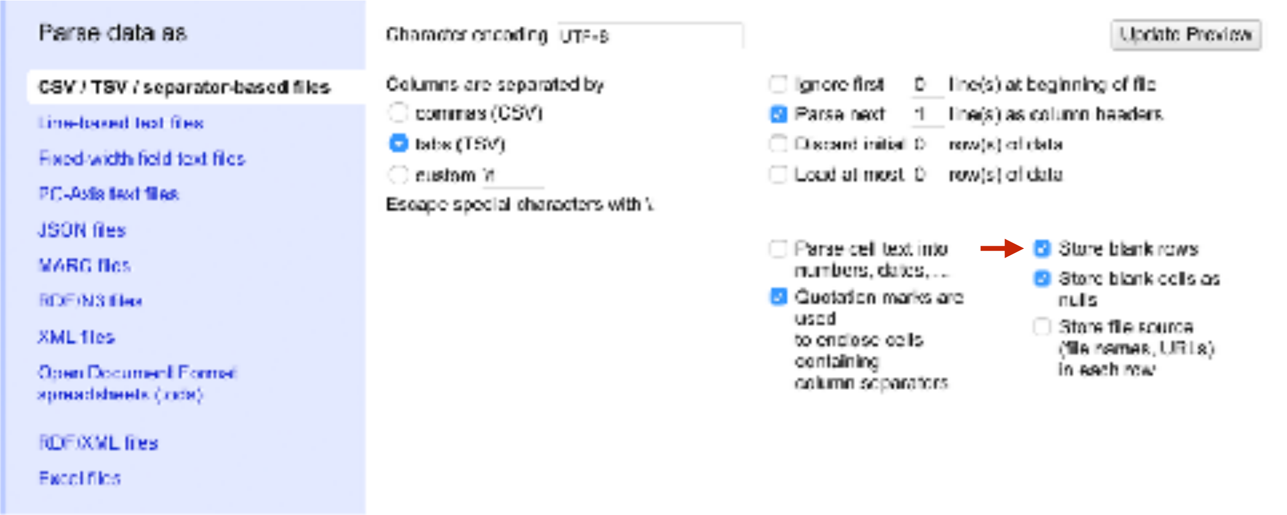
☐ Parse cell text into numbers, dates, ...
☑ Quotation marks are used to enclose cells containing column separators

☑ Store blank rows
☑ Store blank cells as nulls
☐ Store file source (file names, URLs) in each row

OpenRefine typically defaults assuming that quotation marks are used to contain column separators.
I don’t like to use this option unless I know it won’t break the data.
In the case of Exercise 1, we will turn this option off.

OPENREFINE

EXERCISE 1



Character encoding UTF-8

Columns are separated by
○ commas (CSV)
● tabs (TSV)
○ custom \t
Escape special characters with \

□ Ignore first 0 line(s) at beginning of file
☑ Parse next 1 line(s) as column headers
□ Discard initial 0 row(s) of data
□ Load at most 0 row(s) of data

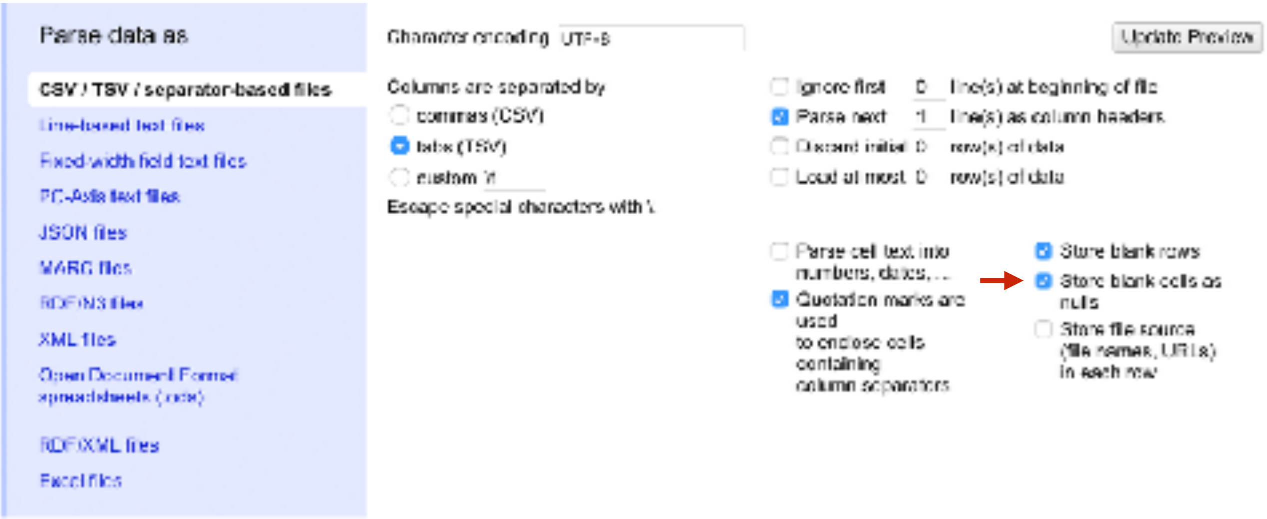
□ Parse cell text into numbers, dates, ...
☑ Quotation marks are used to enclose cells containing column separators

→ ☑ Store blank rows
☑ Store blank cells as nulls
□ Store file source (file names, URLs) in each row

Storing blank rows is, in my personal experience, seldom used.
The possible exception is when a blank row separates individual records.
we'll get to that later, but for now we'll turn it off.

OPENREFINE

EXERCISE 1



Character encoding UTF-8

Columns are separated by
○ commas (CSV)
● tabs (TSV)
○ custom \t
Escape special characters with \

□ Ignore first 0 line(s) at beginning of file
☑ Parse next 1 line(s) as column headers
□ Discard initial 0 row(s) of data
□ Load at most 0 row(s) of data

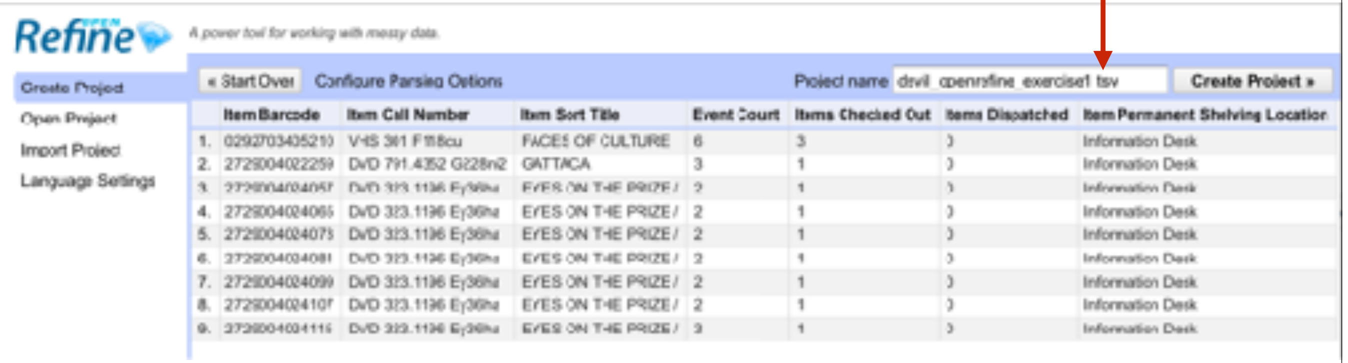
□ Parse cell text into numbers, dates, ...
☑ Quotation marks are used to enclose cells containing column separators

☑ Store blank rows
☑ Store blank cells as nulls
□ Store file source (file names, URLs) in each row

Storing blank cells as nulls may be an option if you’re going to export the data into a relational database management system like MySQL. Otherwise, it shouldn’t hurt to turn it off.

OPENREFINE

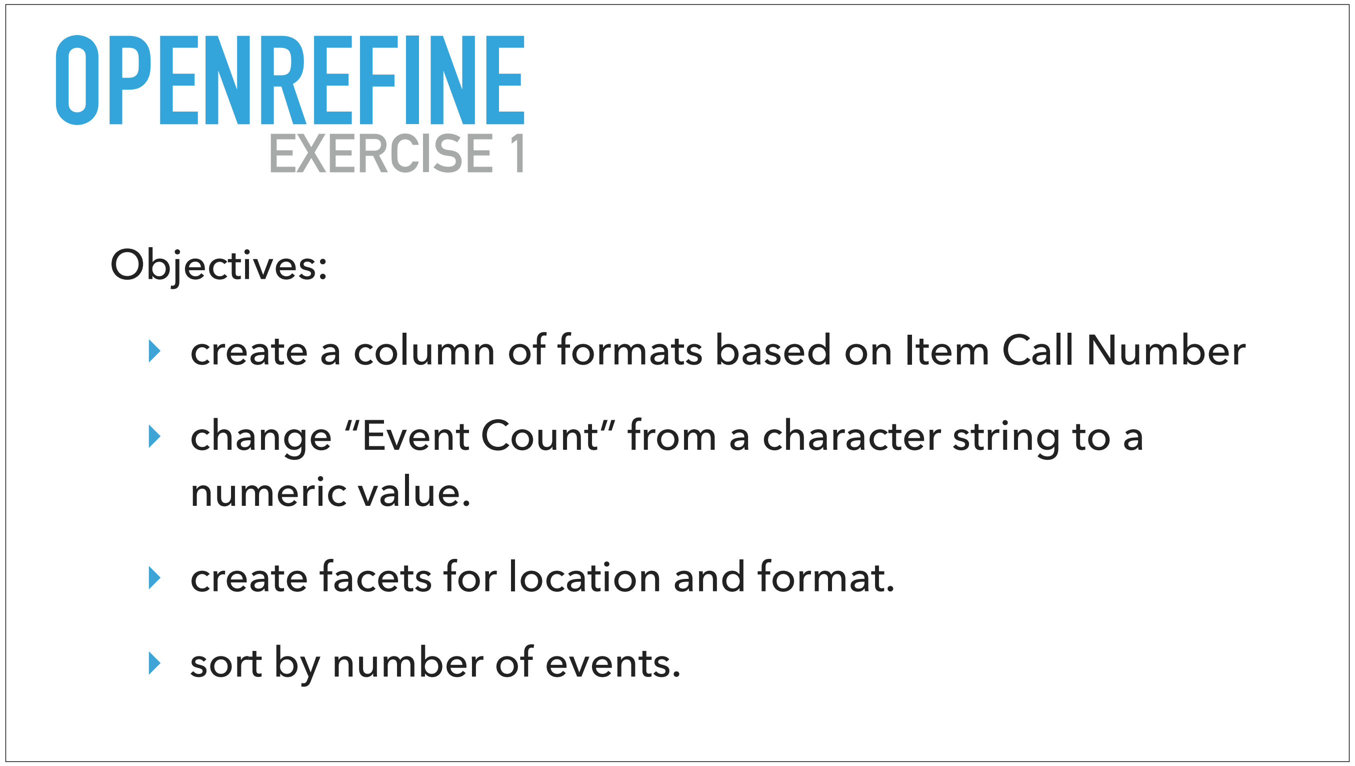
EXERCISE 1



Project name: dvd_openrefine_exercise1 tsv

	Item Barcode	Item Call Number	Item Sort Title	Event Count	Items Checked Out	Items Dispatched	Item Permanent Shelving Location
1.	029270340521	VHS 301 F118cu	FACES OF CULTURE	6	3	0	Information Desk
2.	272900402225	DVD 791.4352 G228n2	GATTACA	3	1	0	Information Desk
3.	272900402405	DVD 323.1196 Ey36ha	EYES ON THE PRIZE /	2	1	0	Information Desk
4.	272900402406	DVD 323.1196 Ey36ha	EYES ON THE PRIZE /	2	1	0	Information Desk
5.	272900402407	DVD 323.1196 Ey36ha	EYES ON THE PRIZE /	2	1	0	Information Desk
6.	272900402408	DVD 323.1196 Ey36ha	EYES ON THE PRIZE /	2	1	0	Information Desk
7.	272900402409	DVD 323.1196 Ey36ha	EYES ON THE PRIZE /	2	1	0	Information Desk
8.	272900402410	DVD 323.1196 Ey36ha	EYES ON THE PRIZE /	2	1	0	Information Desk
9.	272900402411	DVD 323.1196 Ey36ha	EYES ON THE PRIZE /	2	1	0	Information Desk

The preview looks good. The headers look good.
The data in the preview looks well-formed and consistent.
By default, OpenRefine keeps the filename as project name, including the extension minus punctuation, but you can call it whatever makes sense to you.



OPENREFINE

EXERCISE 1

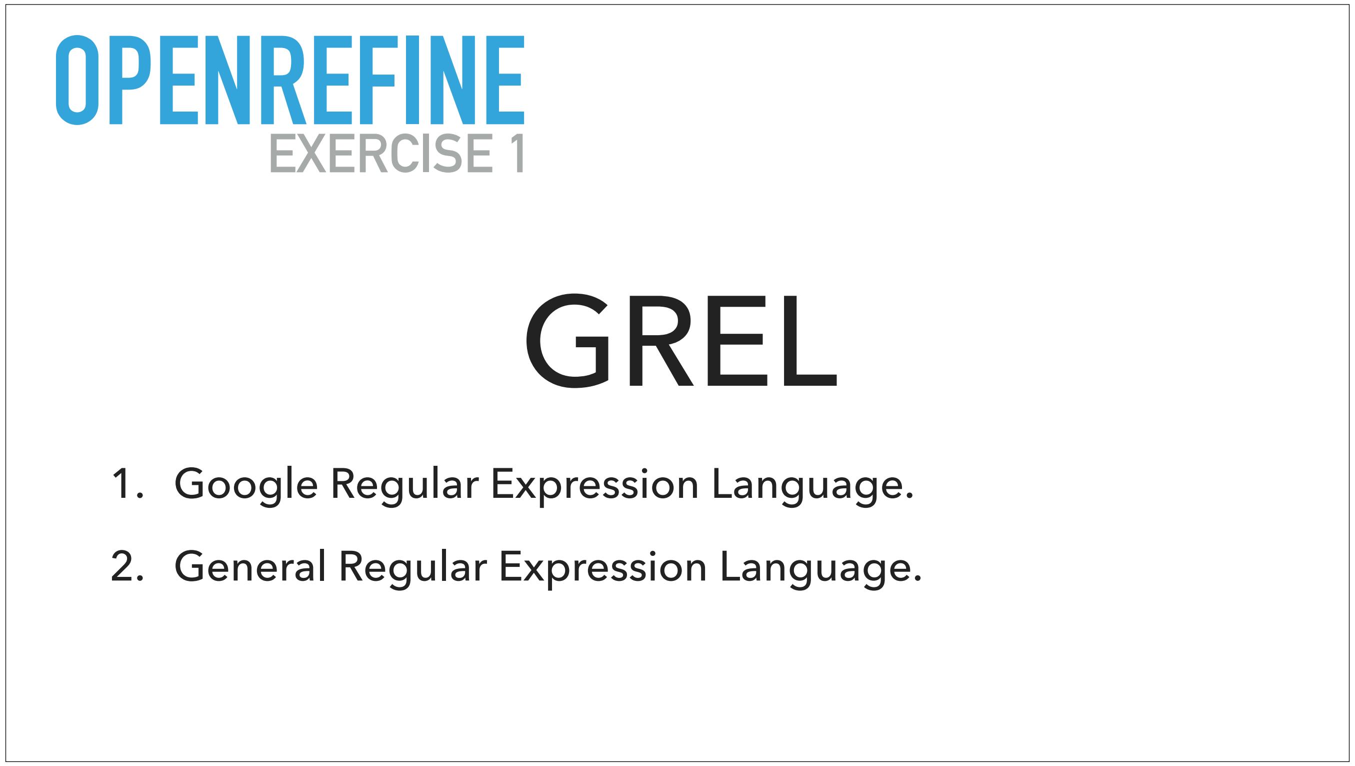
Objectives:

- create a column of formats based on Item Call Number
- change “Event Count” from a character string to a numeric value.
- create facets for location and format.
- sort by number of events.

Bonus: change the case of Item Sort Title from all-caps to Title Case.

```
        substring(value,0,4)
```

Close this tab when done.



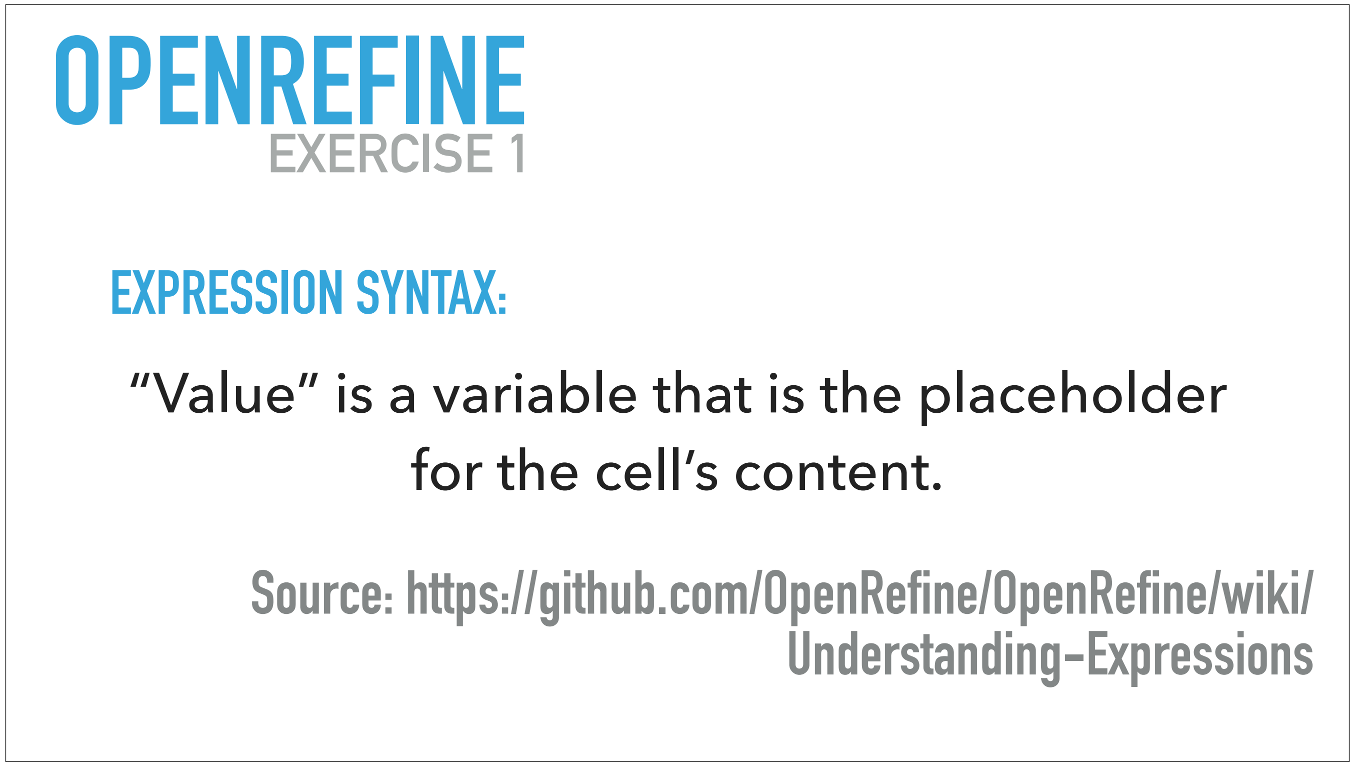
OPENREFINE

EXERCISE 1

GREL

1. Google Regular Expression Language.
2. General Regular Expression Language.

Question for class: how many of you have used regular expressions?
Don’t sweat it if you don’t know it. They are a powerful tool, but they don’t always make sense.
For the sake of consistency, going to try to use “expression” throughout the workshop.



OPENREFINE

EXERCISE 1

EXPRESSION SYNTAX:

“Value” is a variable that is the placeholder for the cell’s content.

Source: https://github.com/OpenRefine/OpenRefine/wiki/Understanding-Expressions

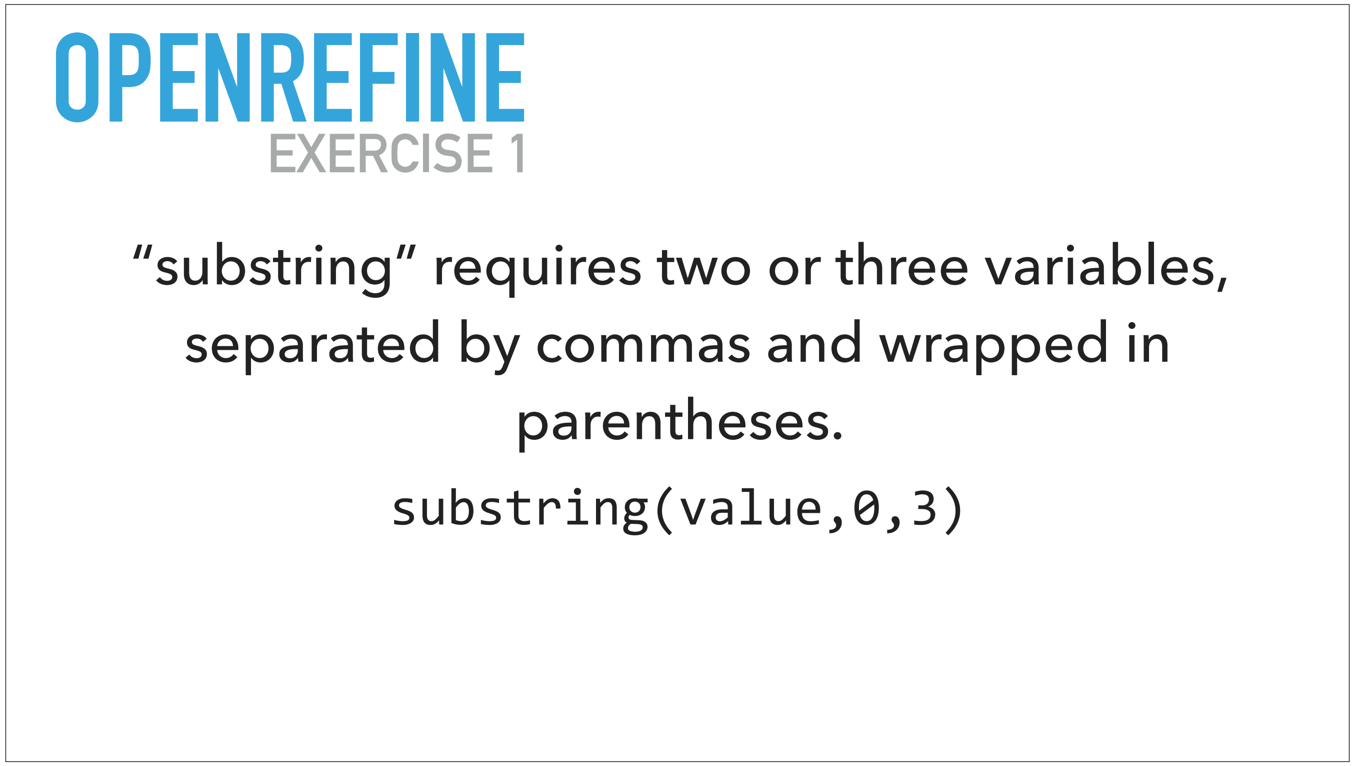
the literal string “value” in an expression is always shorthand for the actual value in the cell.

OPENREFINE

EXERCISE 1

Expressions use camelCase

case sensitive, and camel case sensitive :)



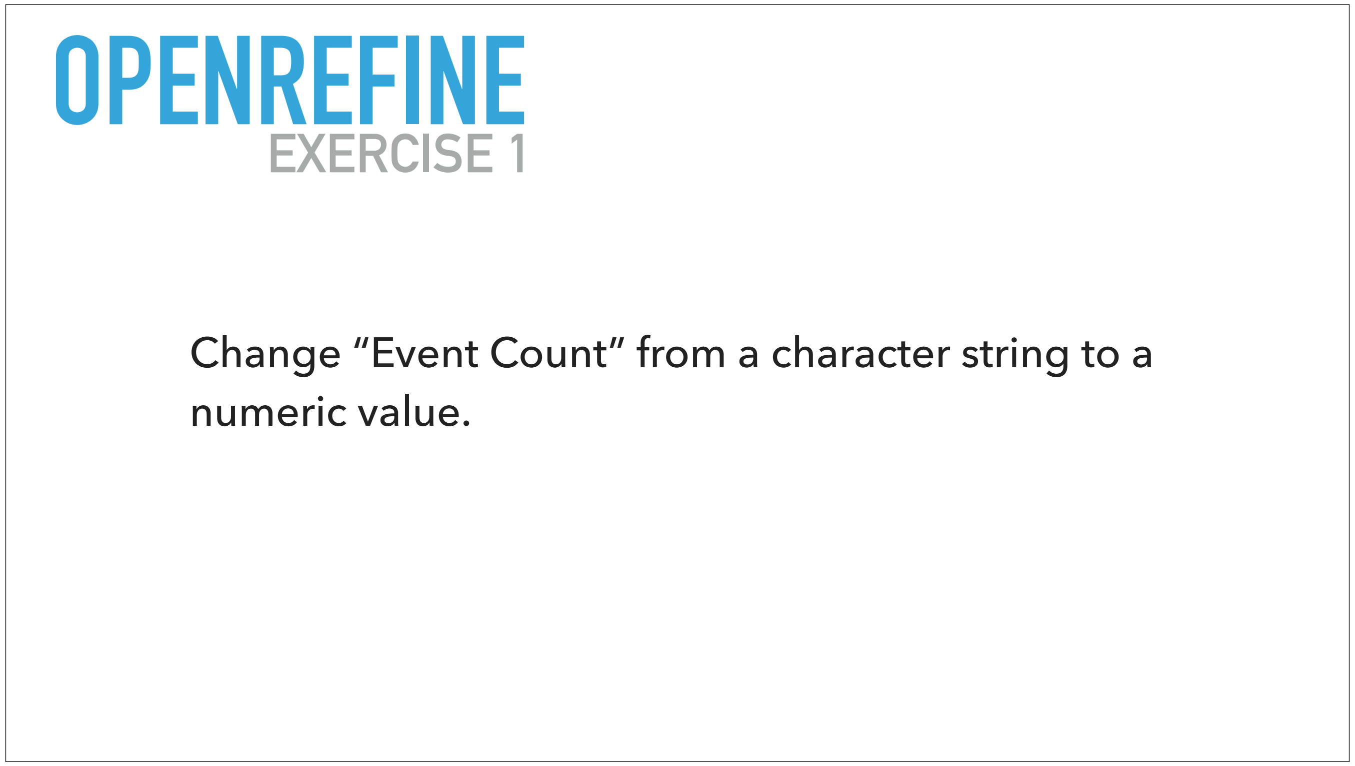
OPENREFINE

EXERCISE 1

“substring” requires two or three variables, separated by commas and wrapped in parentheses.

```
substring(value,0,3)
```

case sensitive, and camel case sensitive :)
A positive number for the third variable tells open refine to capture the string from the start variable through that number of characters.
A negative number for the third variable counts backwards from the end of the string.

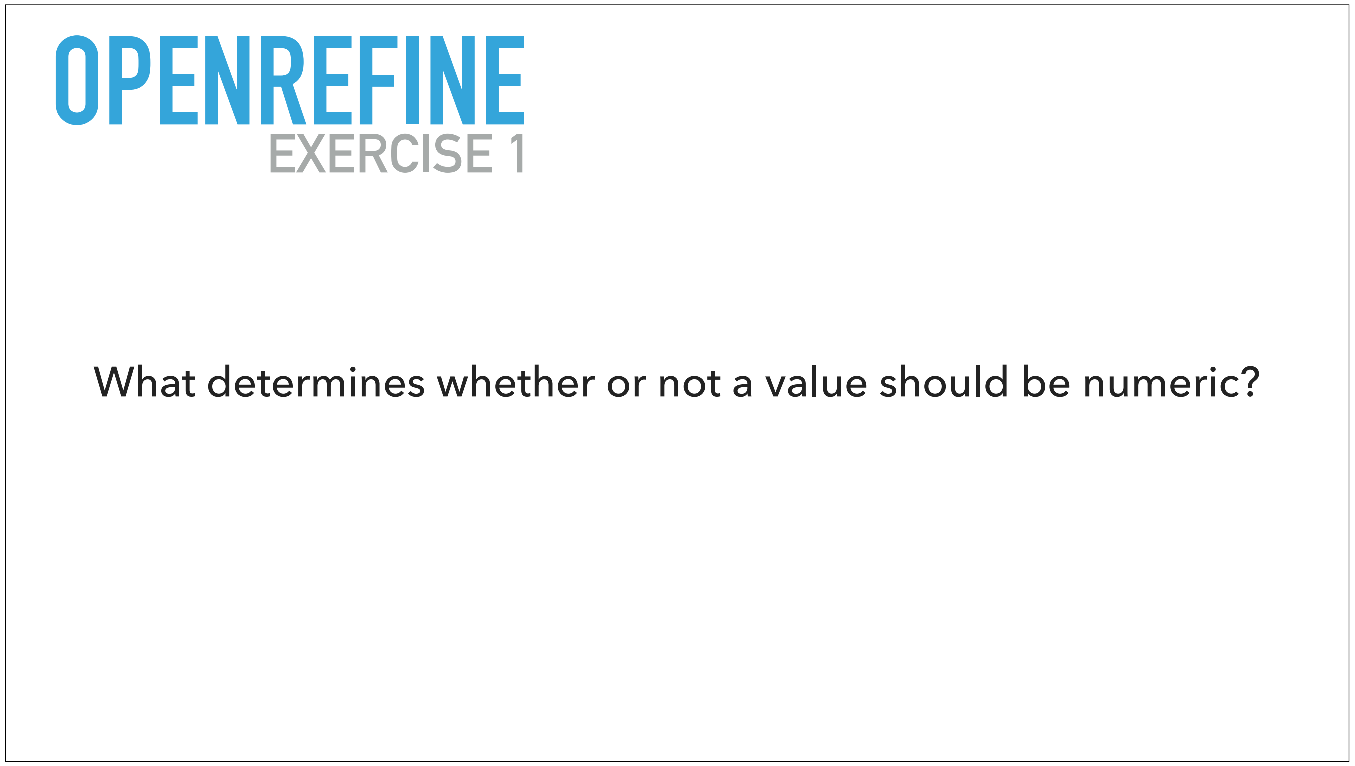


OPENREFINE

EXERCISE 1

Change “Event Count” from a character string to a numeric value.

Two ways to do it.



OPENREFINE

EXERCISE 1

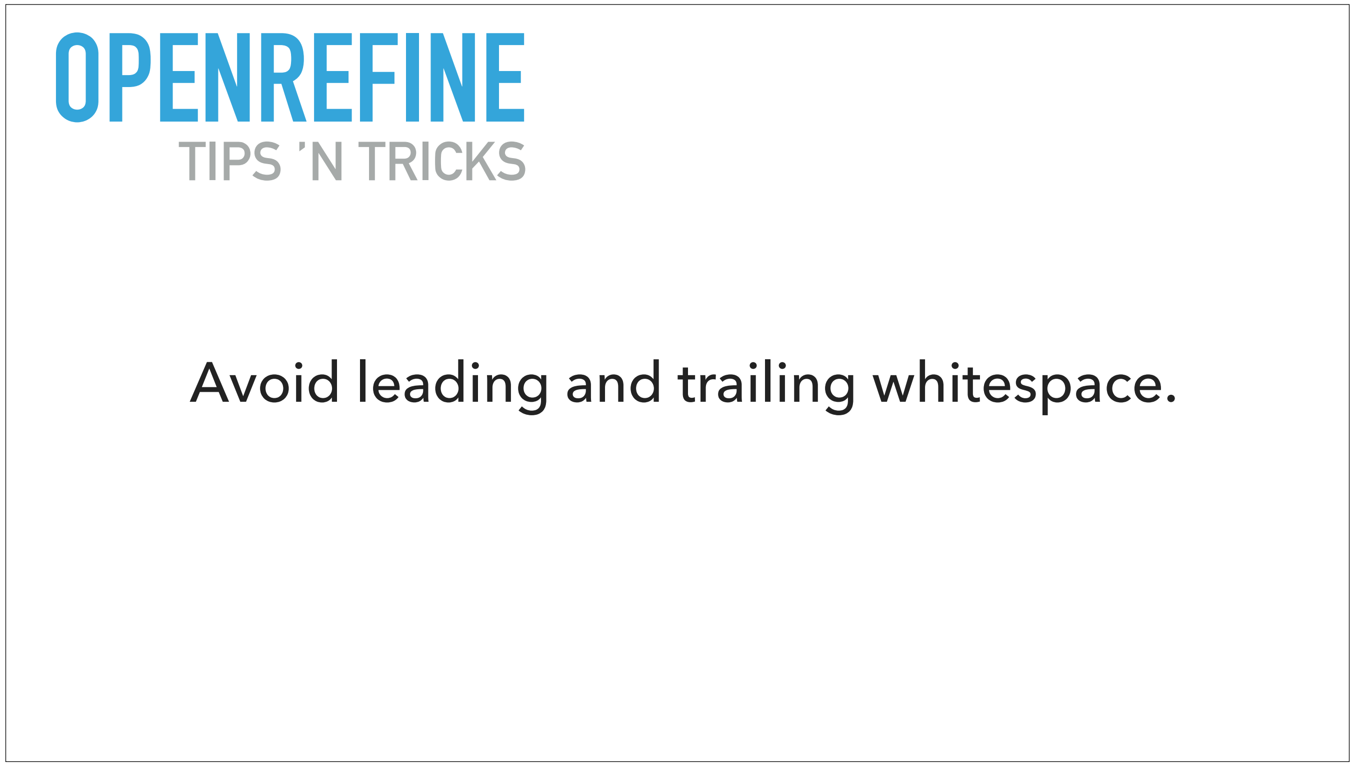
What determines whether or not a value should be numeric?

What determines whether a value is numeric or not?

OPENREFINE

EXERCISE 1

Are you going to perform any mathematical functions on it?

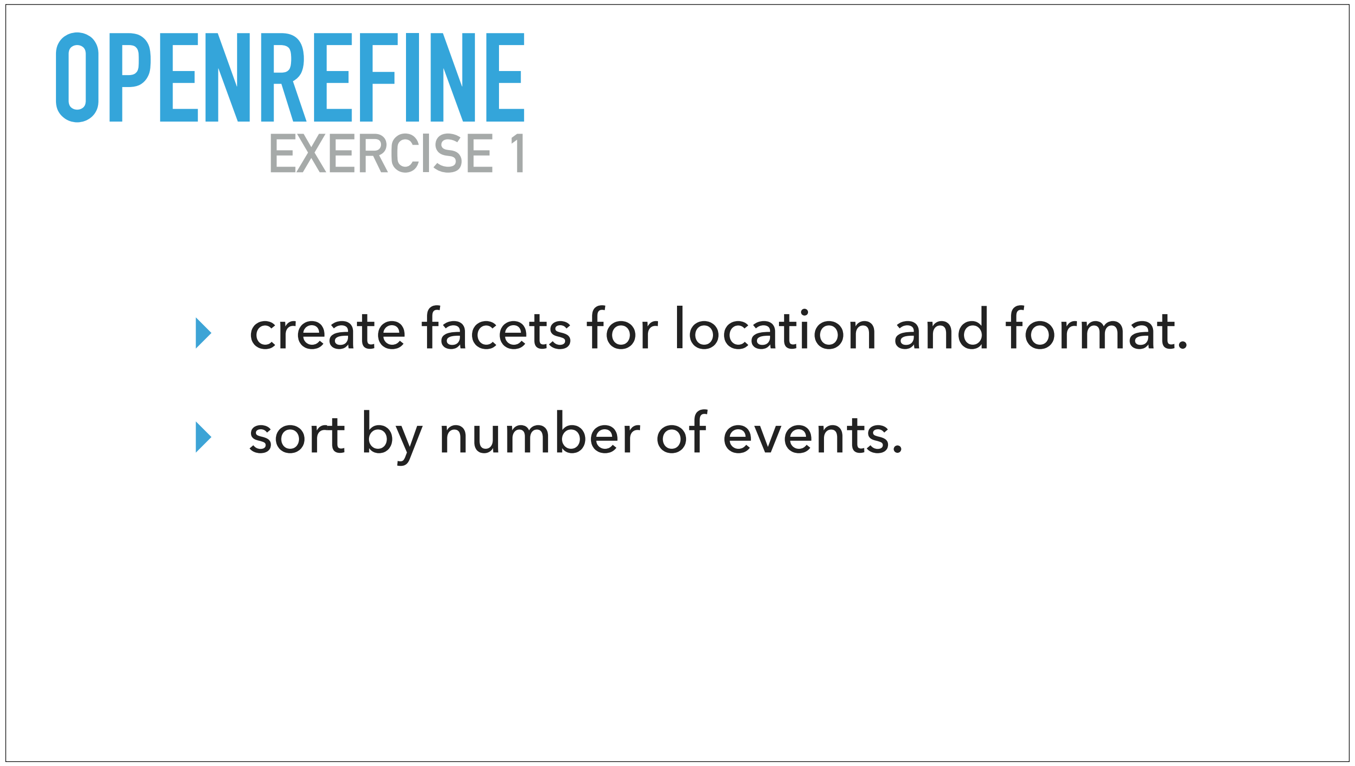


OPENREFINE

TIPS ’N TRICKS

Avoid leading and trailing whitespace.

It can muck with your data.

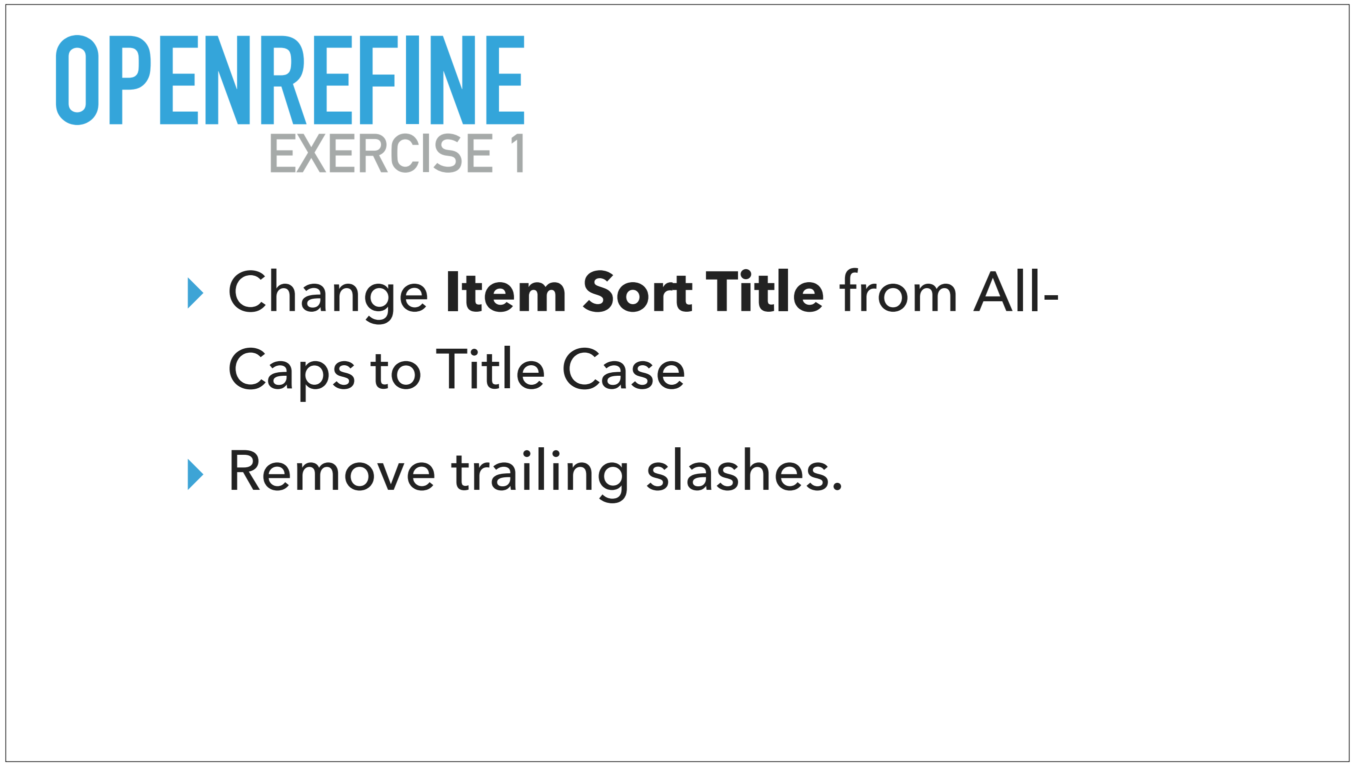


OPENREFINE

EXERCISE 1

- create facets for location and format.
- sort by number of events.

A facet lets you view and edit a subset of the data without affecting the entire dataset.



OPENREFINE

EXERCISE 1

- Change **Item Sort Title** from All-Caps to Title Case
- Remove trailing slashes.

OpenRefine doesn’t know what prepositions and articles are, so those will get the title case treatment as well.

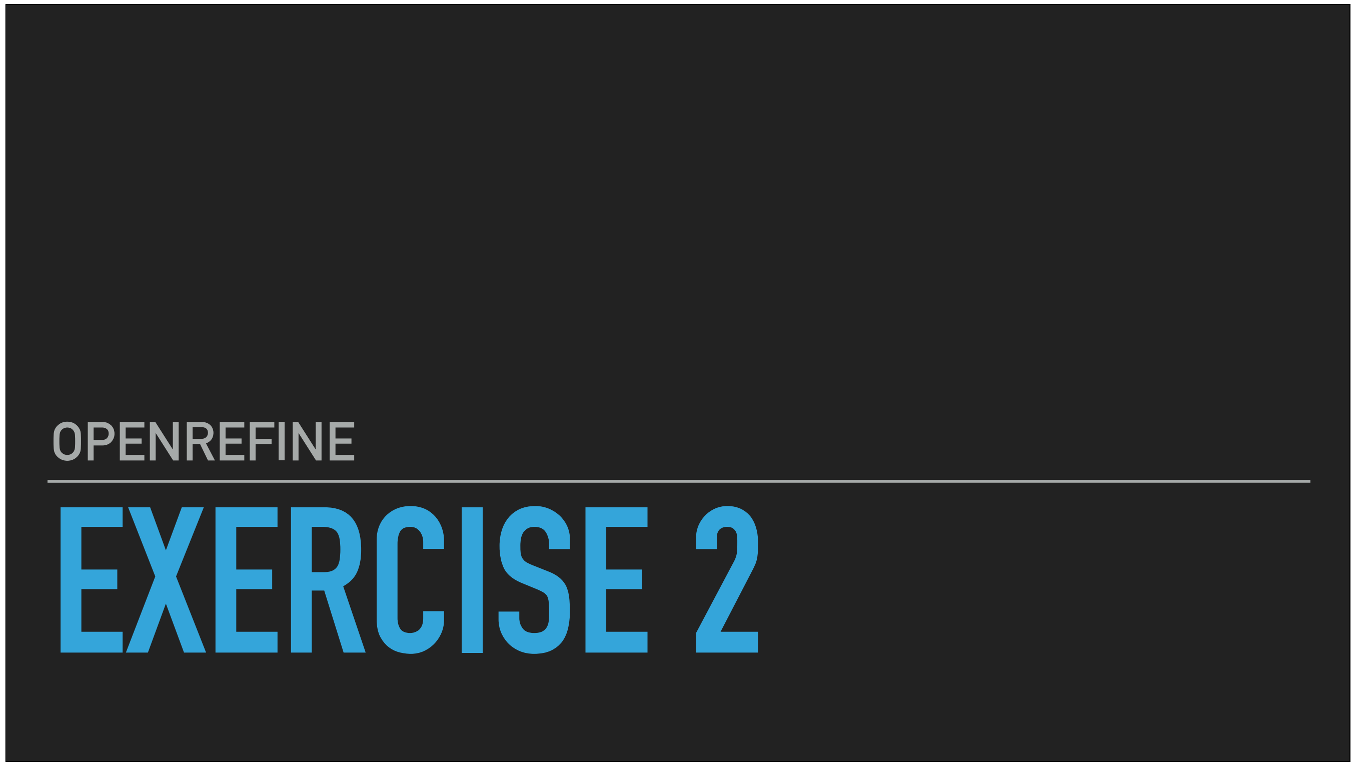
By creating a filter on “Item Sort Title” for ‘/‘, we can Transform the cells to delete the trailing slash.

OPENREFINE

EXERCISE 1

Bonus Level!

Create a column with the Call Number *sans* format.



OPENREFINE

EXERCISE 2

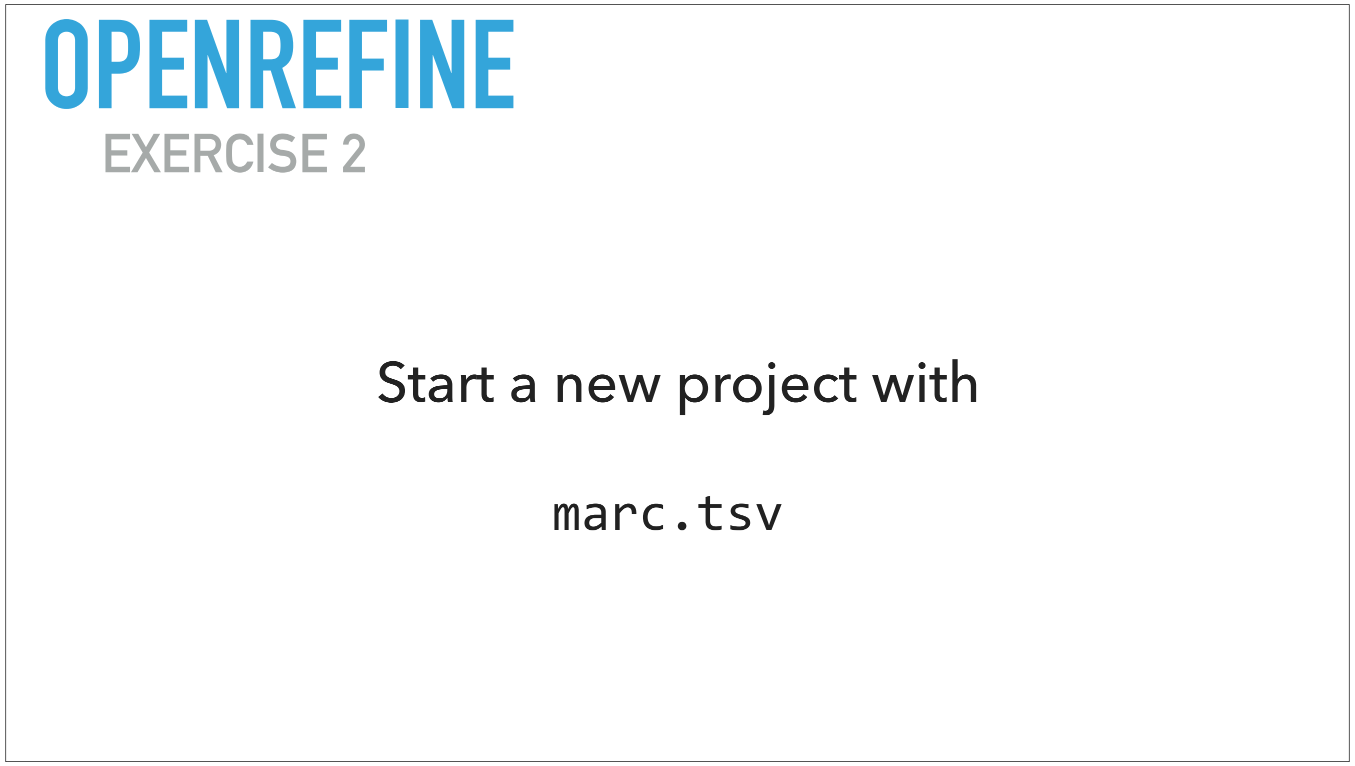
MARC records are some of the craziest data conglomerations that I’ve ever seen. Even if you are not a cataloger (I am not), there is much to be learned from the manipulation of MARC records in OpenRefine.

OPENREFINE

EXERCISE 2

Rows? Records?

develop an awareness of which view you are in.
OpenRefine typically defaults to rows



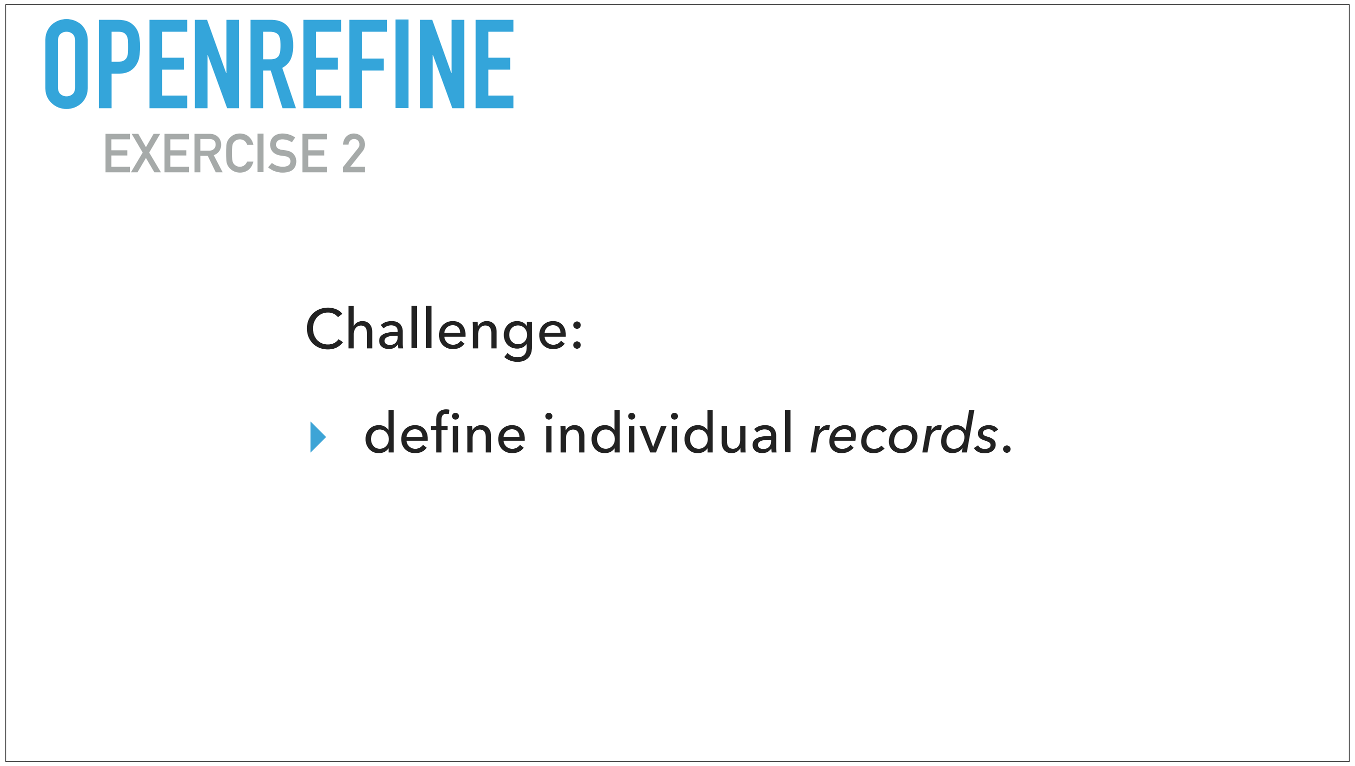
OPENREFINE

EXERCISE 2

Start a new project with

```
marc.tsv
```

try it on your own or in groups, but DON’T click “create”



OPENREFINE

EXERCISE 2

Challenge:

- define individual *records*.

This is a little tricky, but isn’t so bad once you get the gist.

OPENREFINE

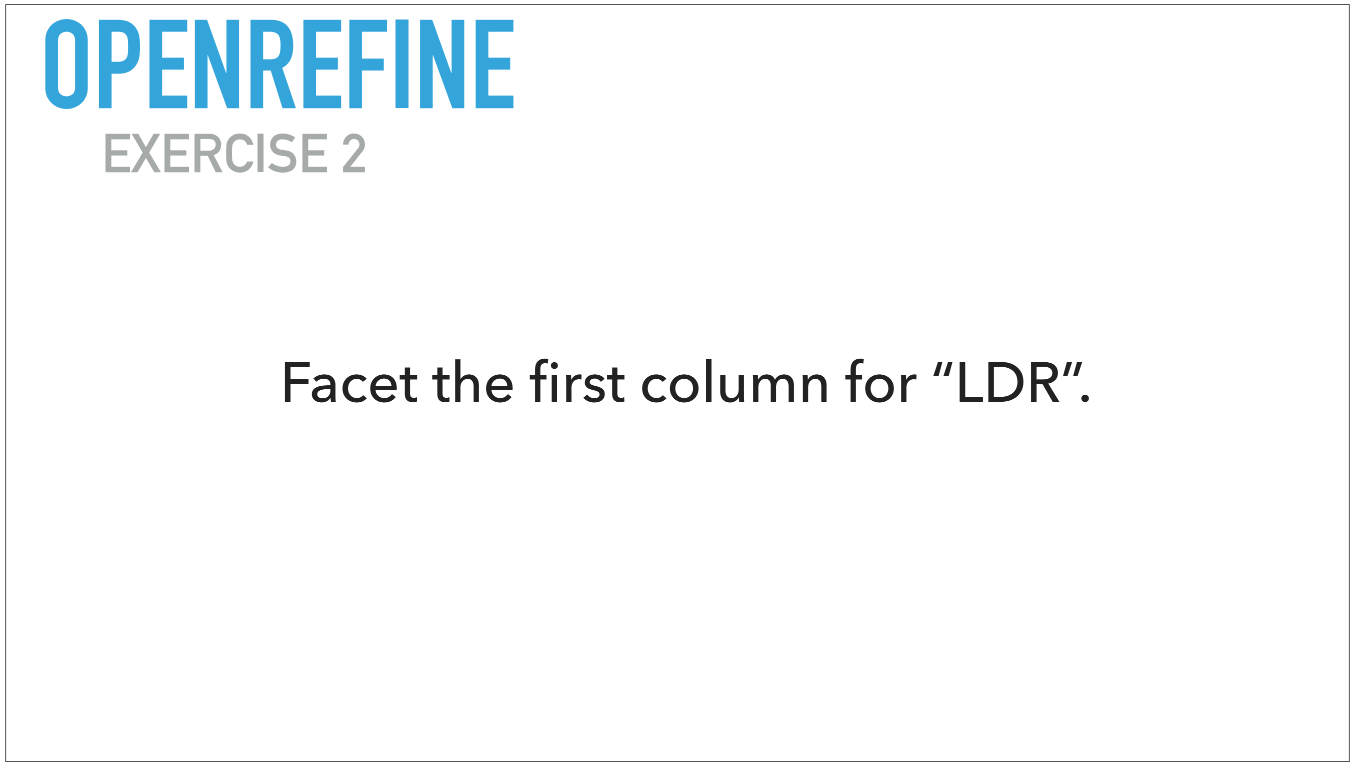
EXERCISE 2

Take me to your LDR.

(I’m so, so sorry)

Copyright 20th Century Fox

Next, since the leader marks the beginning of an individual marc record…



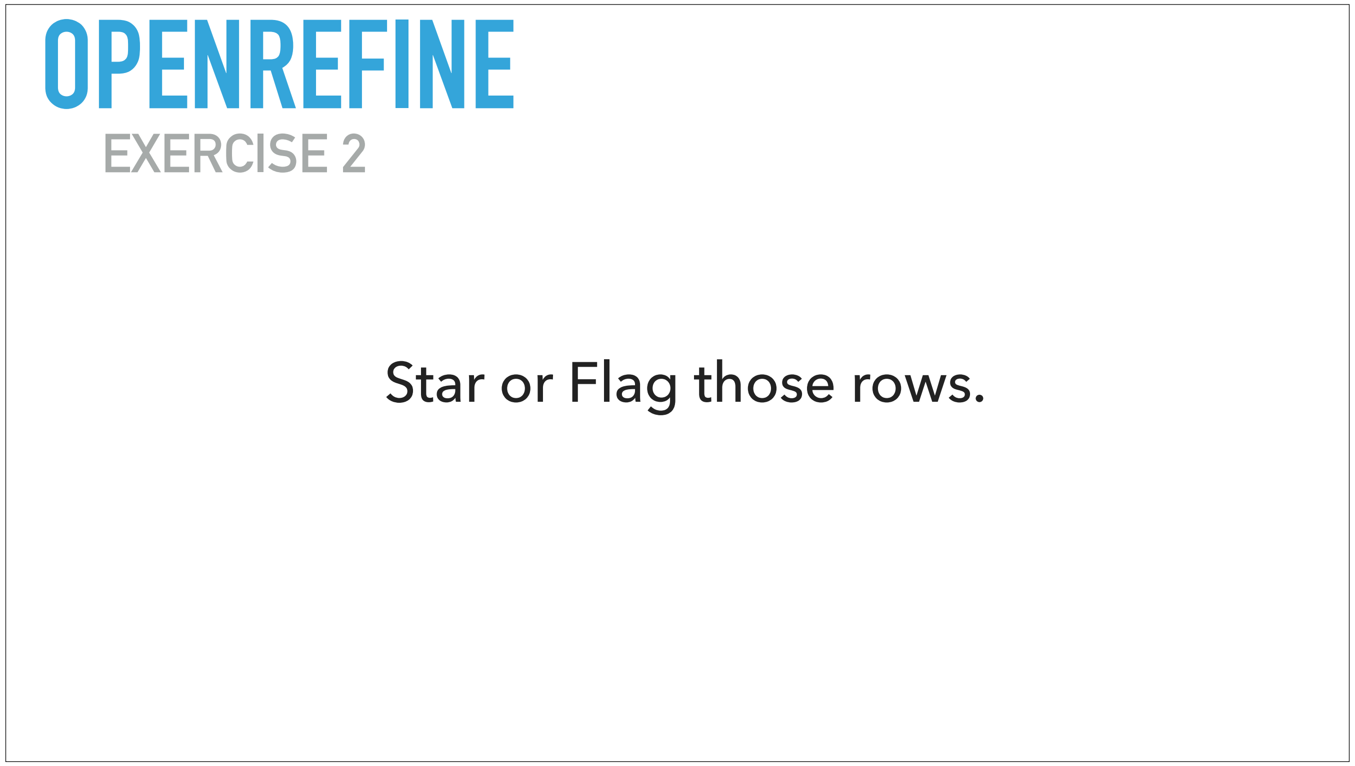
OPENREFINE

EXERCISE 2

Facet the first column for “LDR”.

we’re going to facet the first column and isolate the rows that start with “LDR”

This brings up the point: what if you’re not familiar with a dataset that you’ve been asked to clean? Remember “Know your Data”: this includes asking someone who might know more about the format.

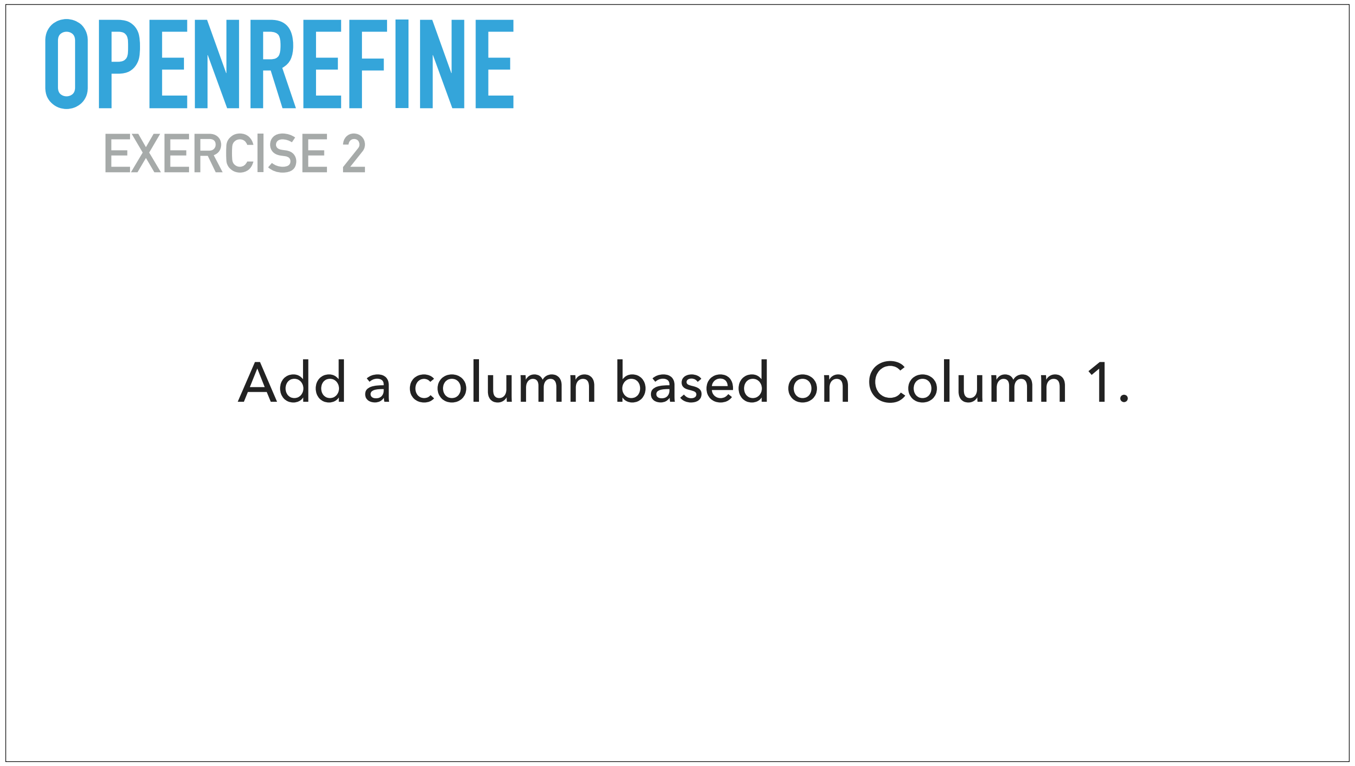


OPENREFINE

EXERCISE 2

Star or Flag those rows.

then star those rows.

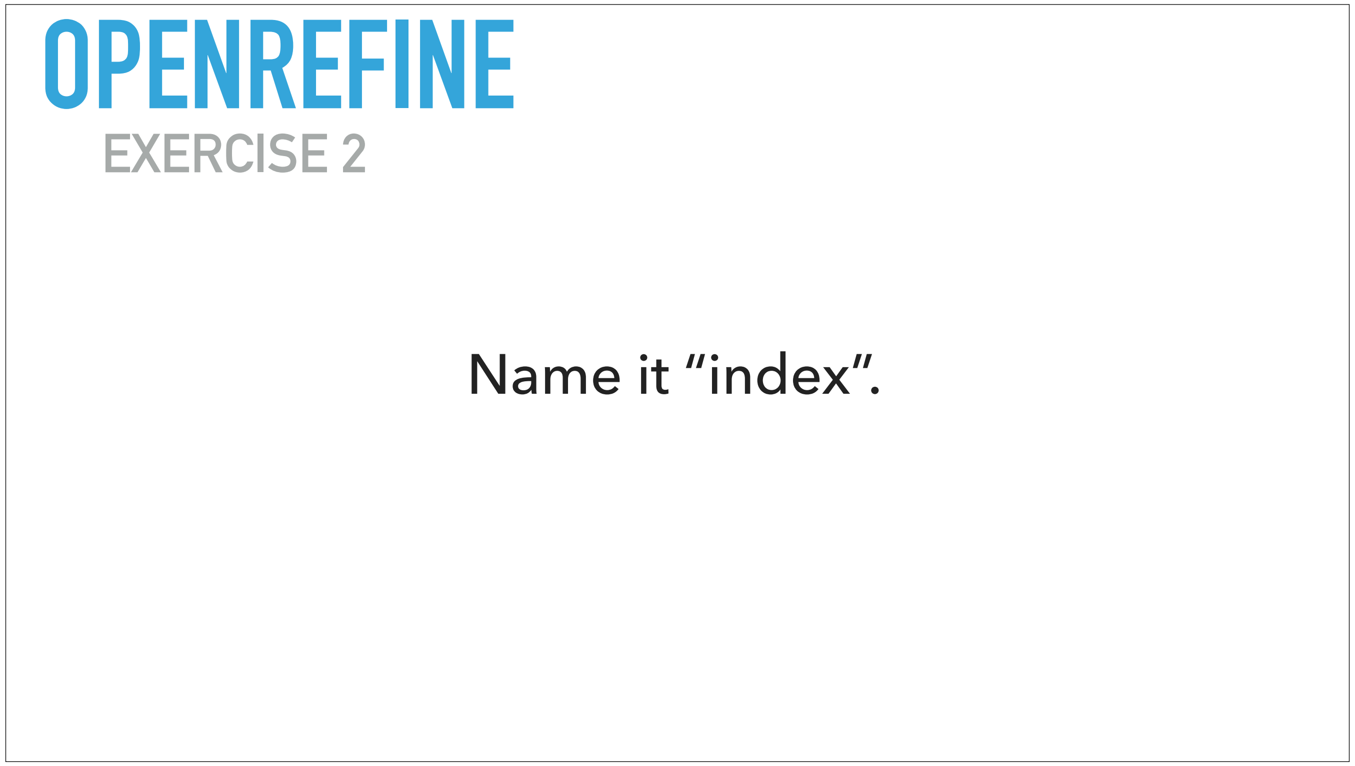


OPENREFINE

EXERCISE 2

Add a column based on Column 1.

I’ll add a new column based on the col containing the “LDR” value.

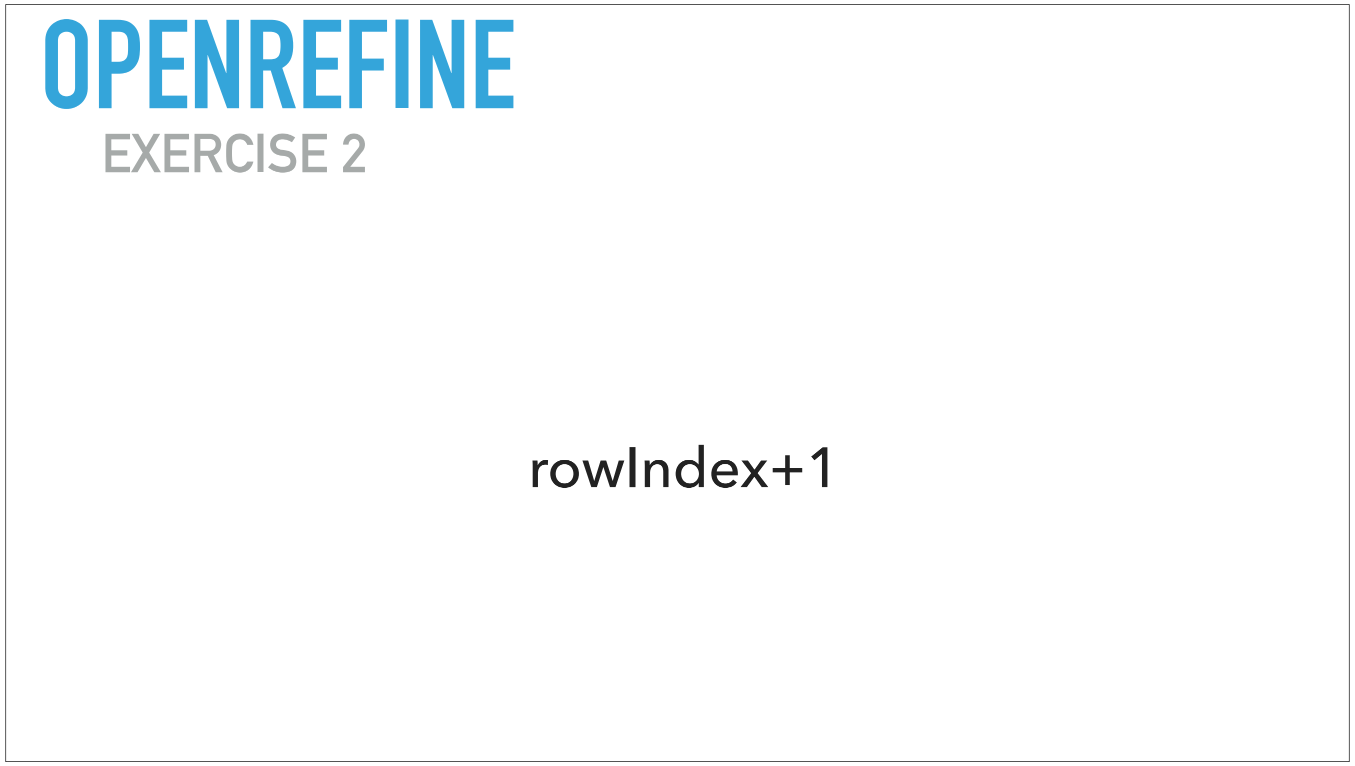


OPENREFINE

EXERCISE 2

Name it “index”.

I’m going to name the new column.
I like to call mine “index”.

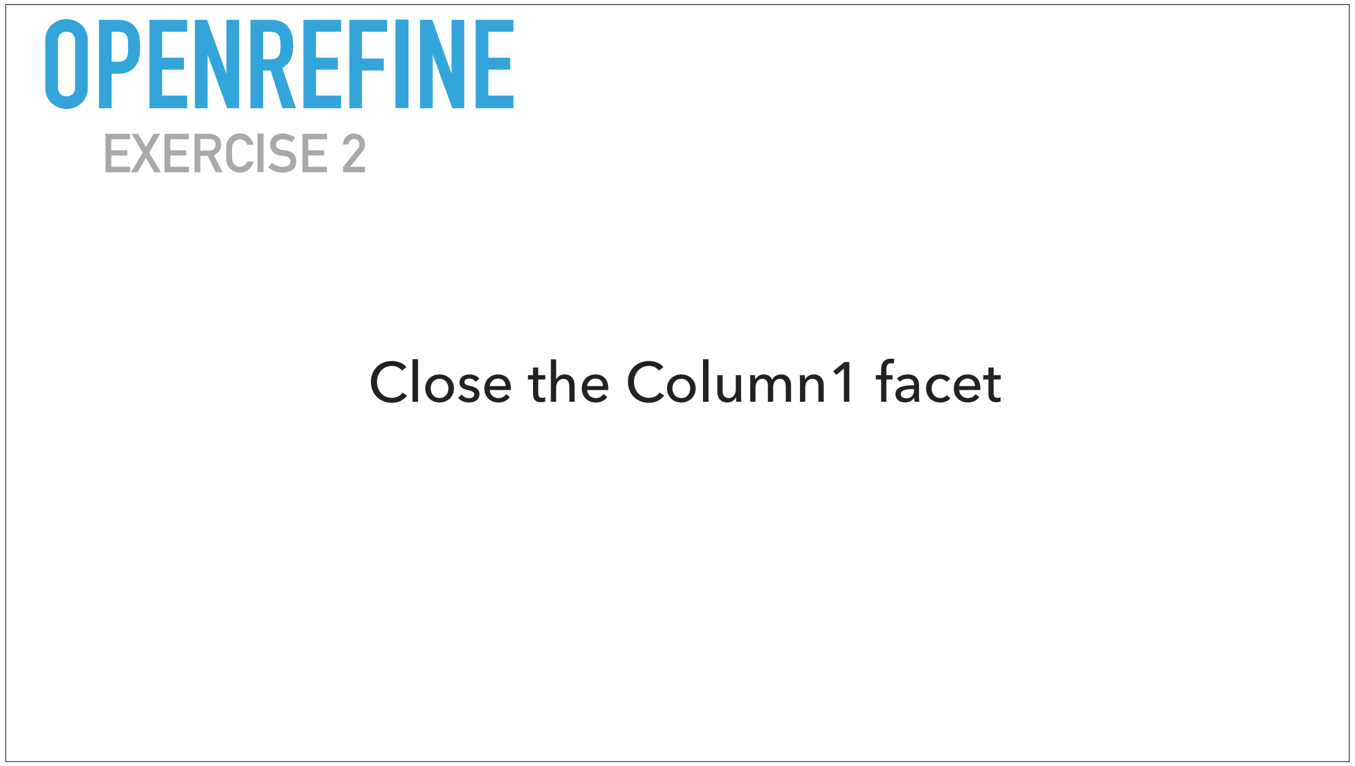


OPENREFINE

EXERCISE 2

rowIndex+1

We use the formula aka GREL syntax rowIndex+1 to create cell values
Why +1? because OpenRefine, like many programs (and programming languages) is zero-based

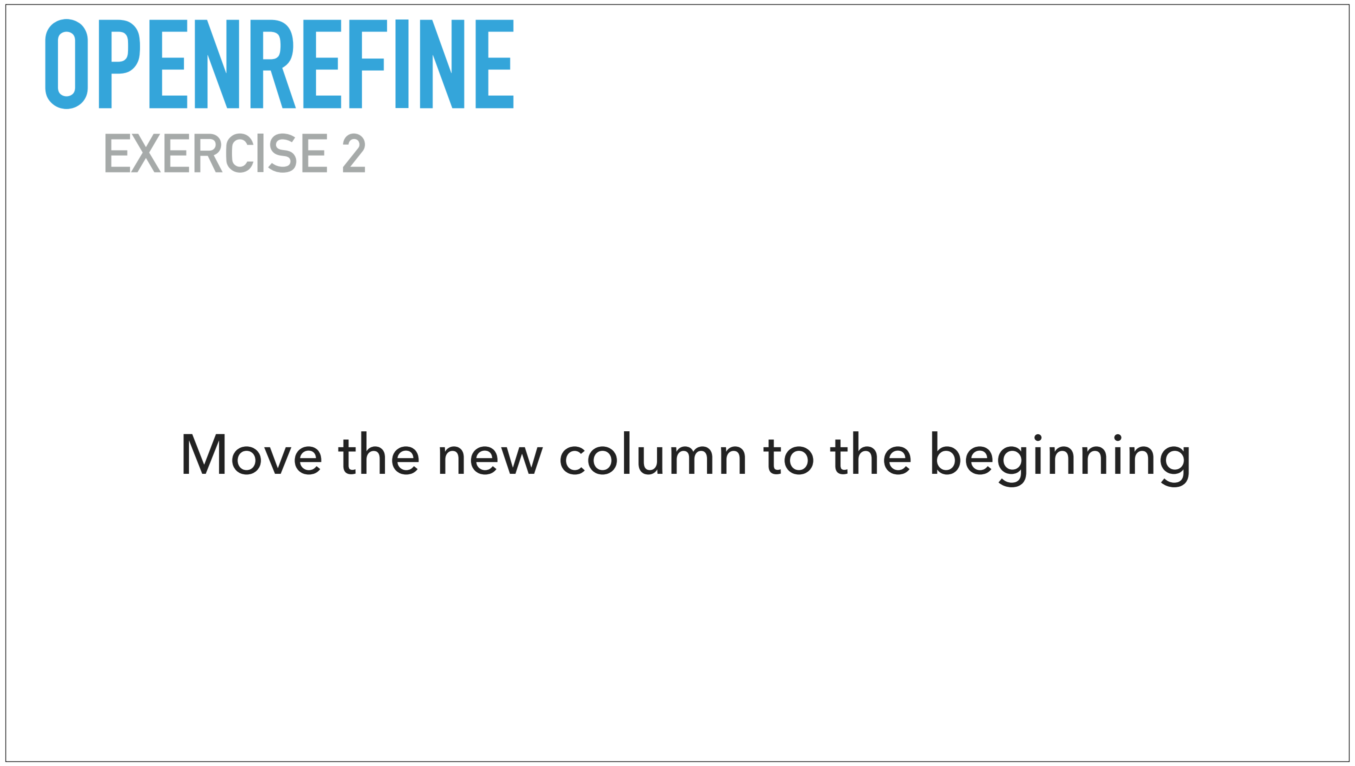


OPENREFINE

EXERCISE 2

Close the Column1 facet

Once that’s complete, close the facets dialog to show all the rows.

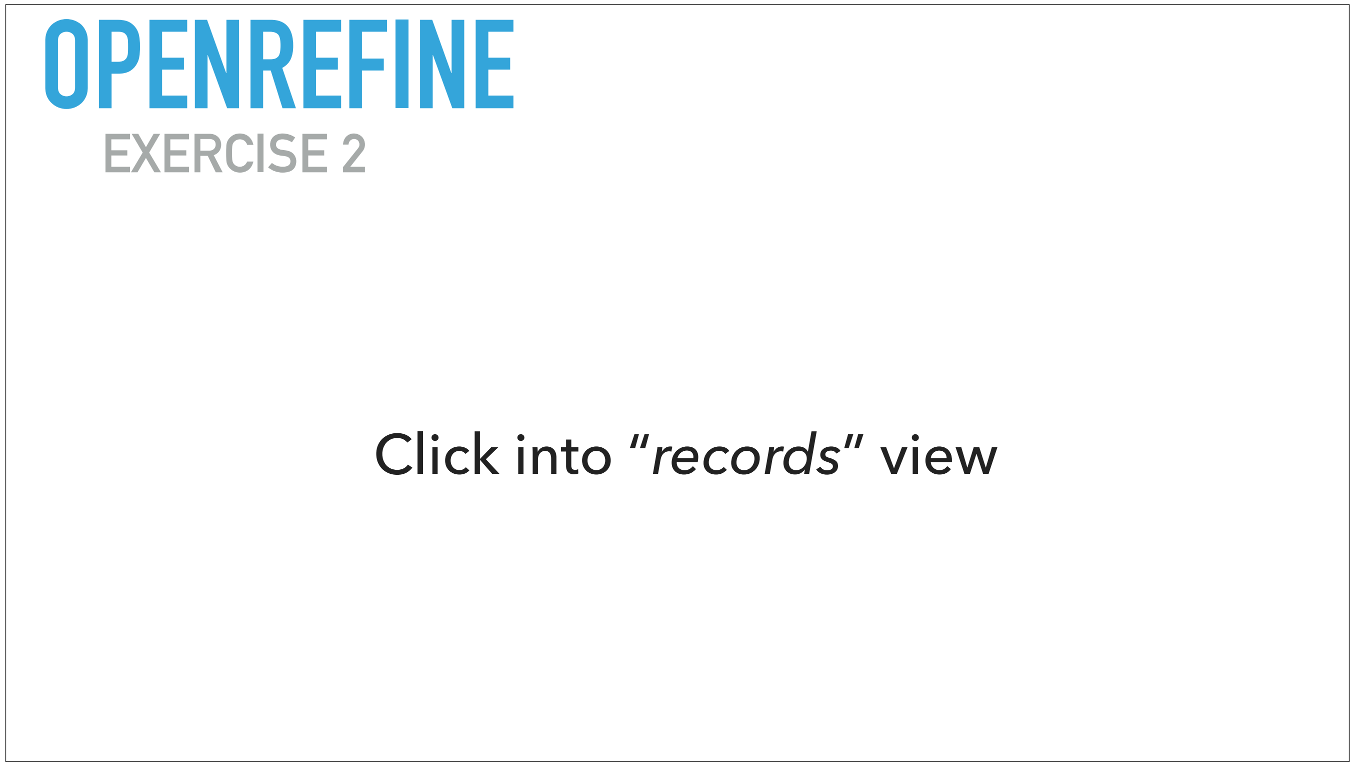


OPENREFINE

EXERCISE 2

Move the new column to the beginning

We’ll make the new column “key” the first column in the dataset.



OPENREFINE

EXERCISE 2

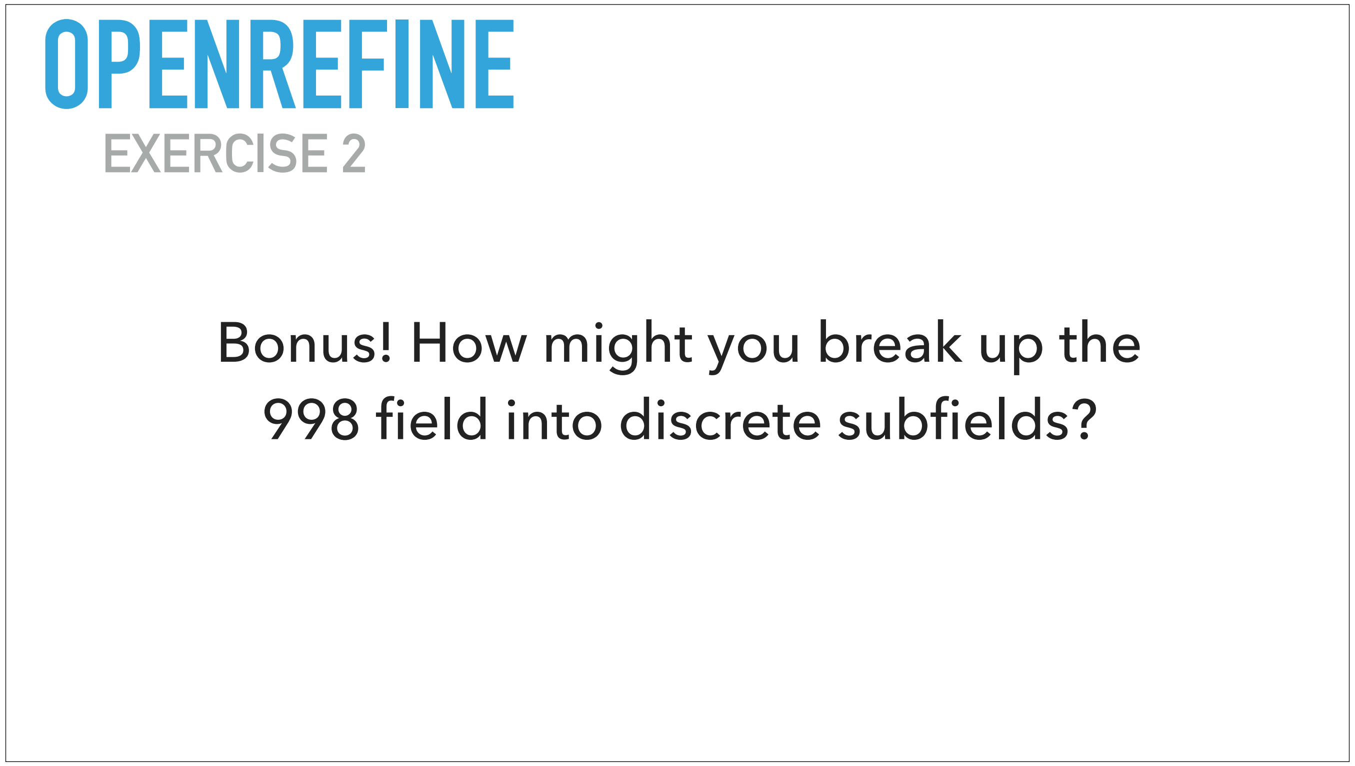
Click into “*records*” view

to verify that OpenRefine identifies individual marc records,
click into the records view.

OPENREFINE

EXERCISE 2

You’ve got distinct records.



OPENREFINE

EXERCISE 2

Bonus! How might you break up the 998 field into discrete subfields?

Tell the ILS migration story. Explain that the 998 has “special data” useful for migrations.

OPENREFINE

EXERCISE 3

Something a bit more complicated. Perhaps even a little scary. It’s okay, I’m the Virgil to your Dante.

OPENREFINE

EXERCISE 3

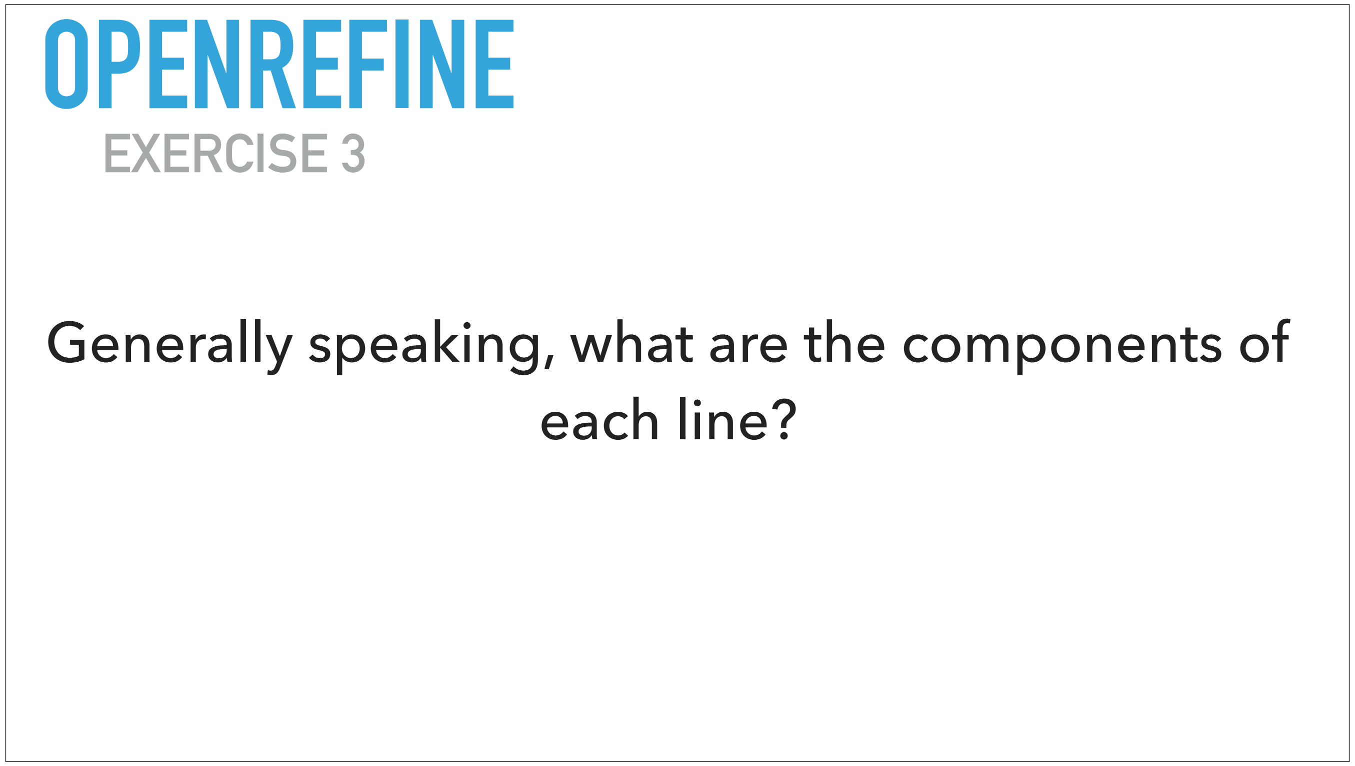
Create a new OpenRefine project with
ezproxy.log

OPENREFINE

EXERCISE 3

Get to know the data.

Let's take a moment to scroll through the data. We're not looking for any specific thing; we're looking at the overall pattern and format of the data. We're checking its consistency and its structure.

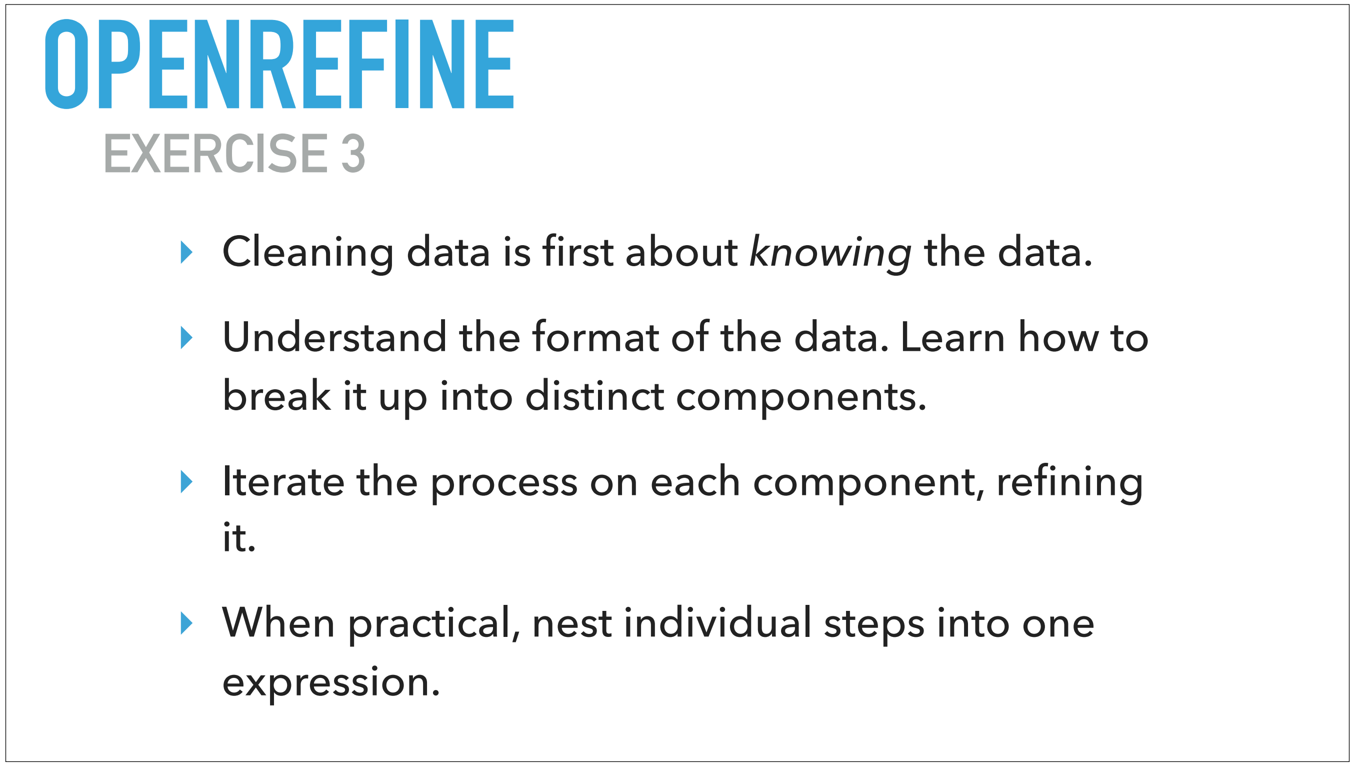


OPENREFINE

EXERCISE 3

Generally speaking, what are the components of each line?

What are the components of each line? The headers are there to help you.

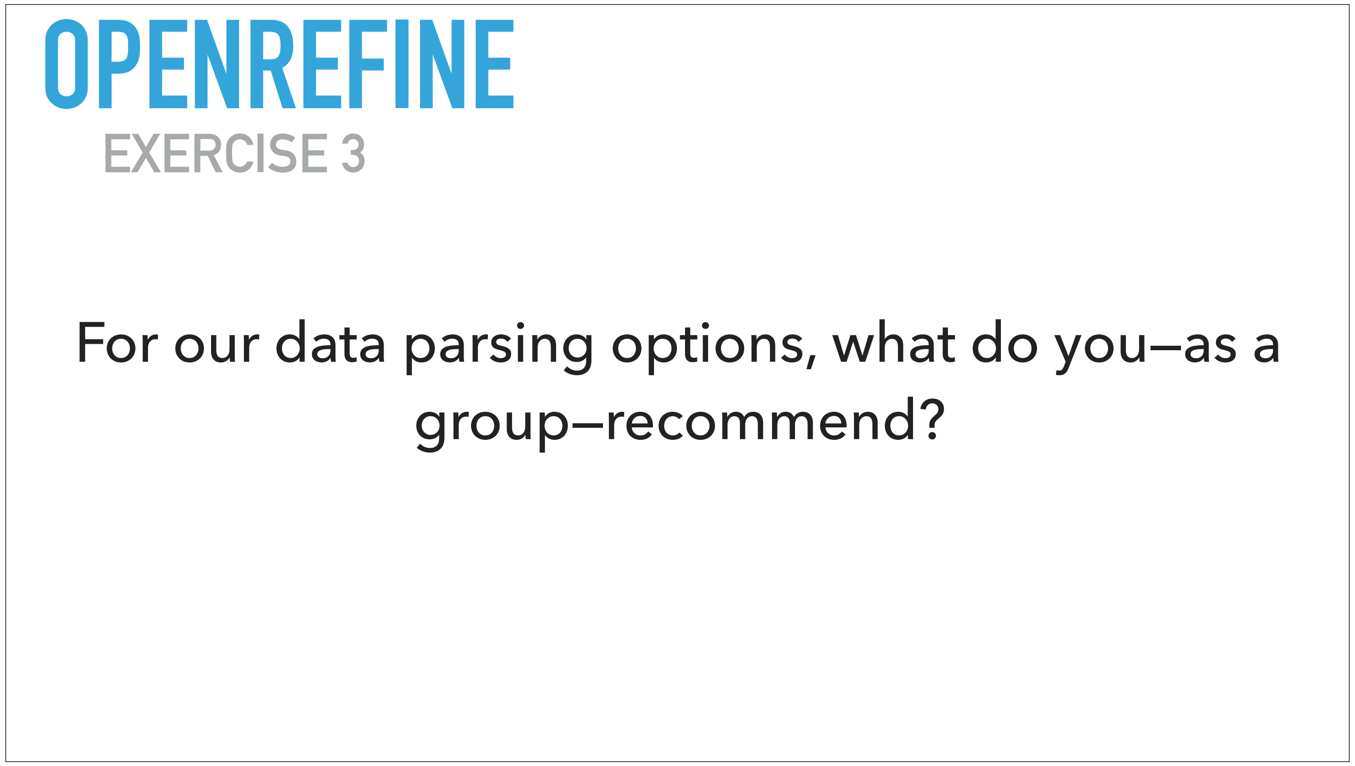


OPENREFINE

EXERCISE 3

- Cleaning data is first about *knowing* the data.
- Understand the format of the data. Learn how to break it up into distinct components.
- Iterate the process on each component, refining it.
- When practical, nest individual steps into one expression.

To reiterate:

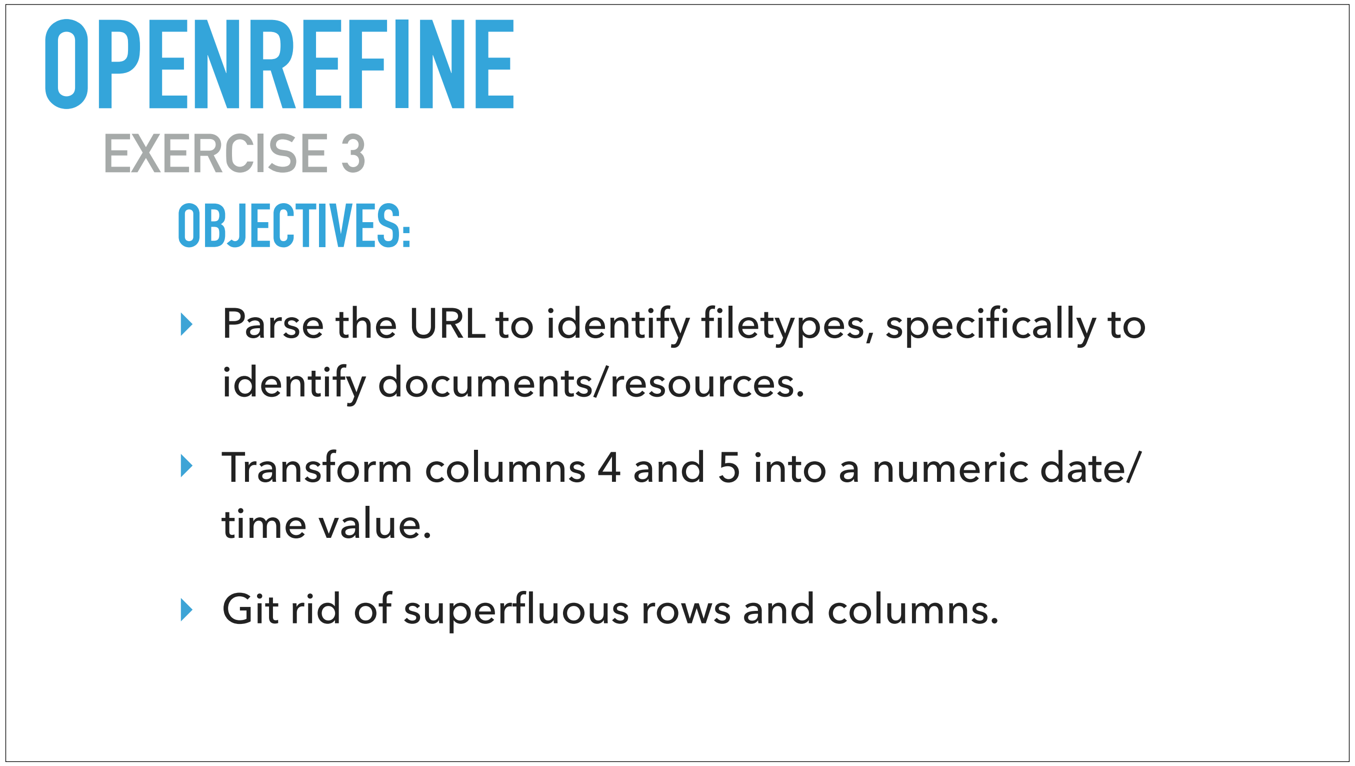


OPENREFINE

EXERCISE 3

For our data parsing options, what do you–as a group–recommend?

Talk amongst yourselves.



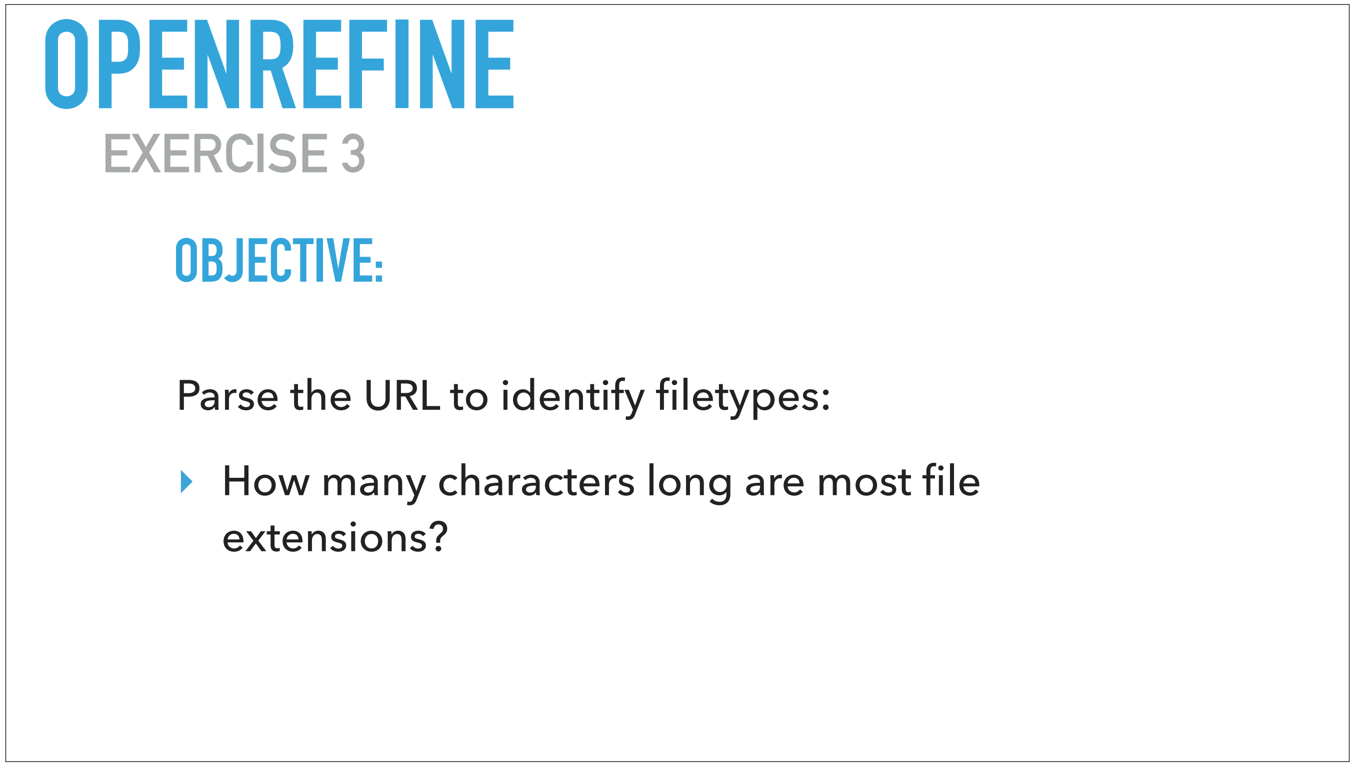
OPENREFINE

EXERCISE 3

OBJECTIVES:

- Parse the URL to identify filetypes, specifically to identify documents/resources.
- Transform columns 4 and 5 into a numeric date/time value.
- Git rid of superfluous rows and columns.

read objectives.
what document/resource is the ultimate goal of a patron’s completed query/search.



OPENREFINE

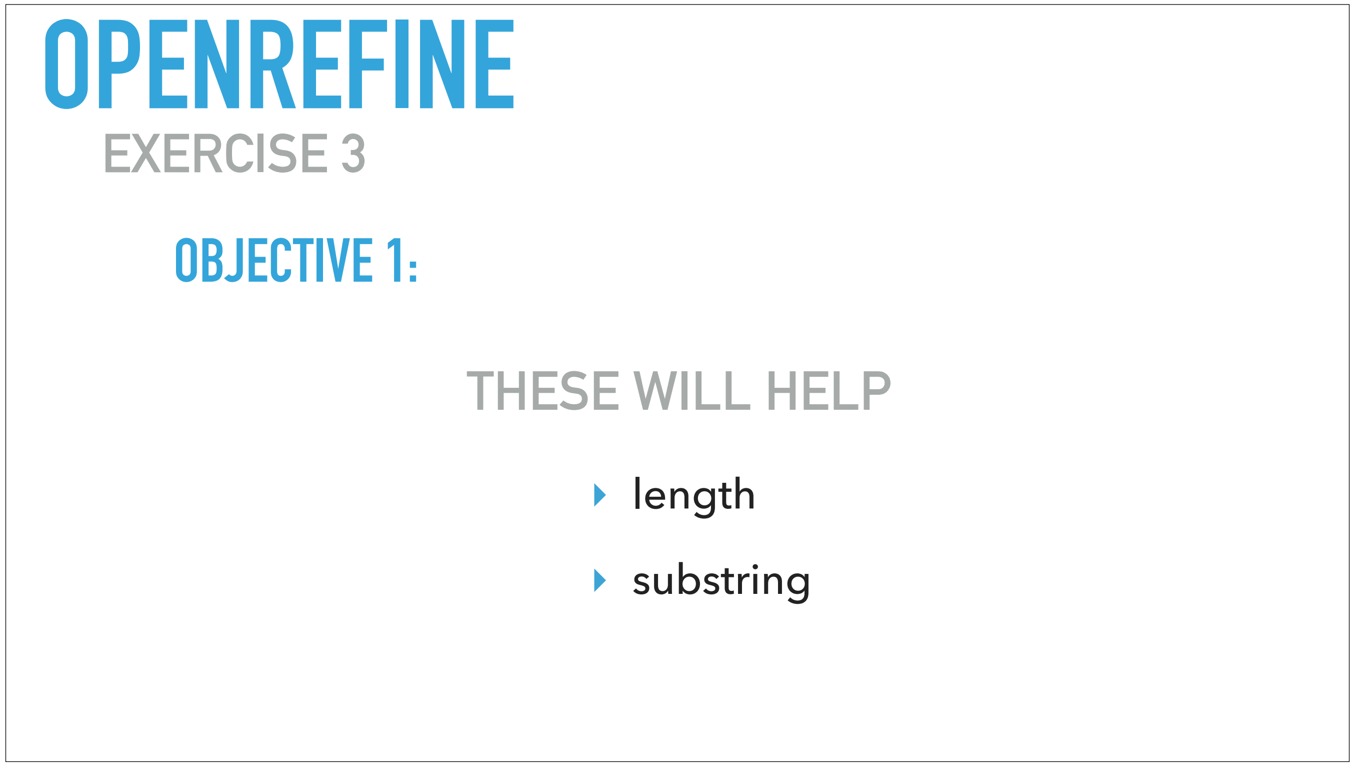
EXERCISE 3

OBJECTIVE:

Parse the URL to identify filetypes:

- How many characters long are most file extensions?

Most file extensions are three characters, preceded by a period or dot.



OPENREFINE

EXERCISE 3

OBJECTIVE 1:

THESE WILL HELP

- length
- substring

1. length(value) to get the number of characters in the url.
2. substring(value,(length(value)-4) to get the last four characters of the URL.
2. create a filter on the new column, using a regular expression to suss out those values that start with a '.'

OPENREFINE

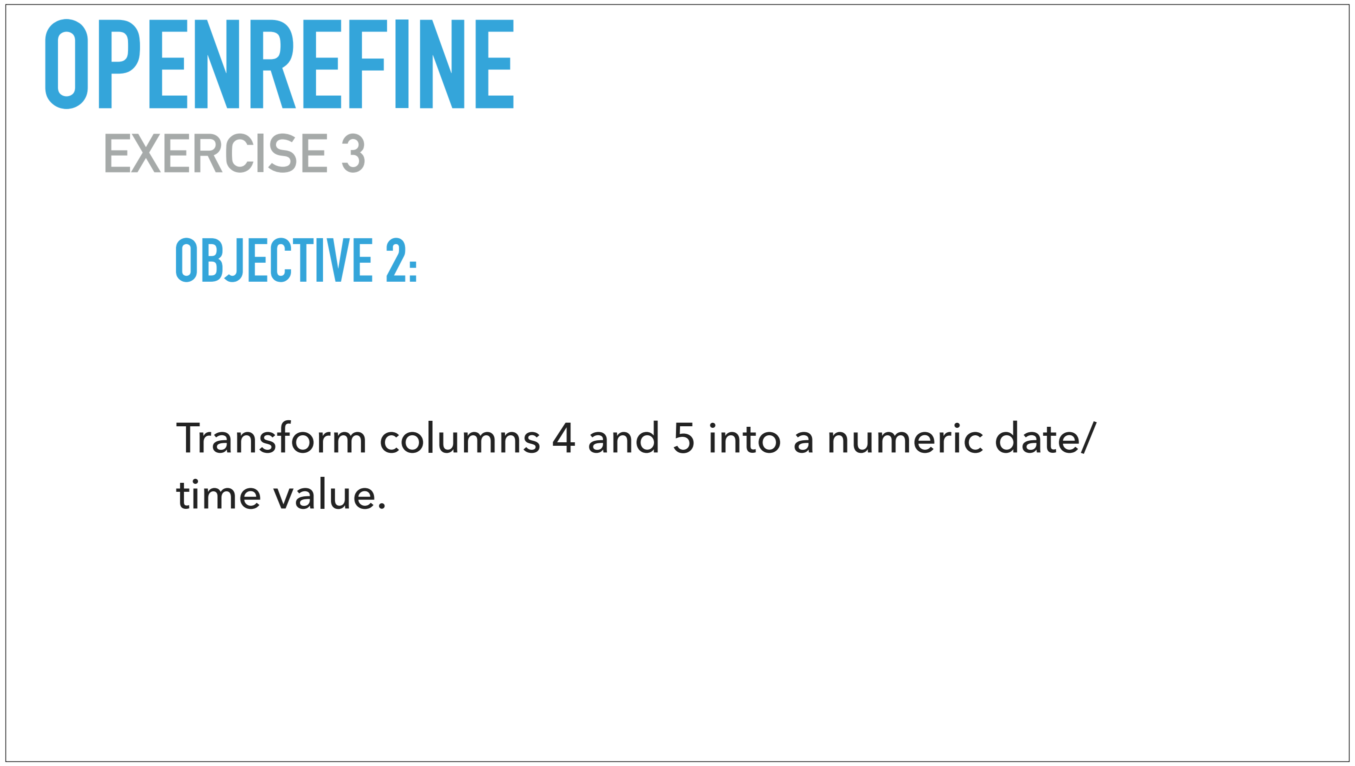
HERE

EXERCISE 3

OBJECTIVE 3:

Filter out the URLs that end in the following extensions:

- .jpg
- .gif
- .css
- .png
- .js
- .ico
- .tiff
- .ttf
- .woff



OPENREFINE

EXERCISE 3

OBJECTIVE 2:

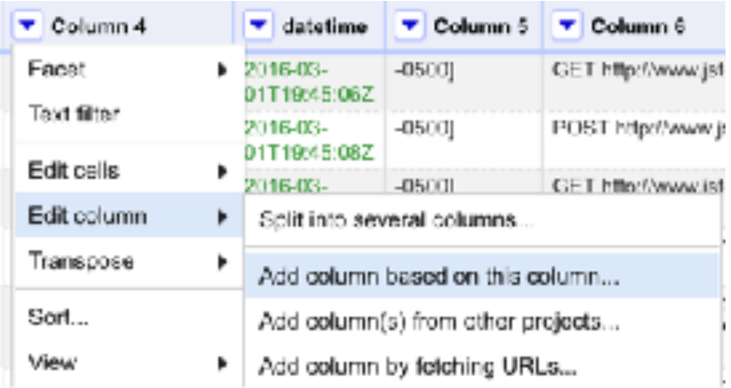
Transform columns 4 and 5 into a numeric date/time value.

I’ll guide you through this one, as well.
First we have to merge, or concatenate the values.

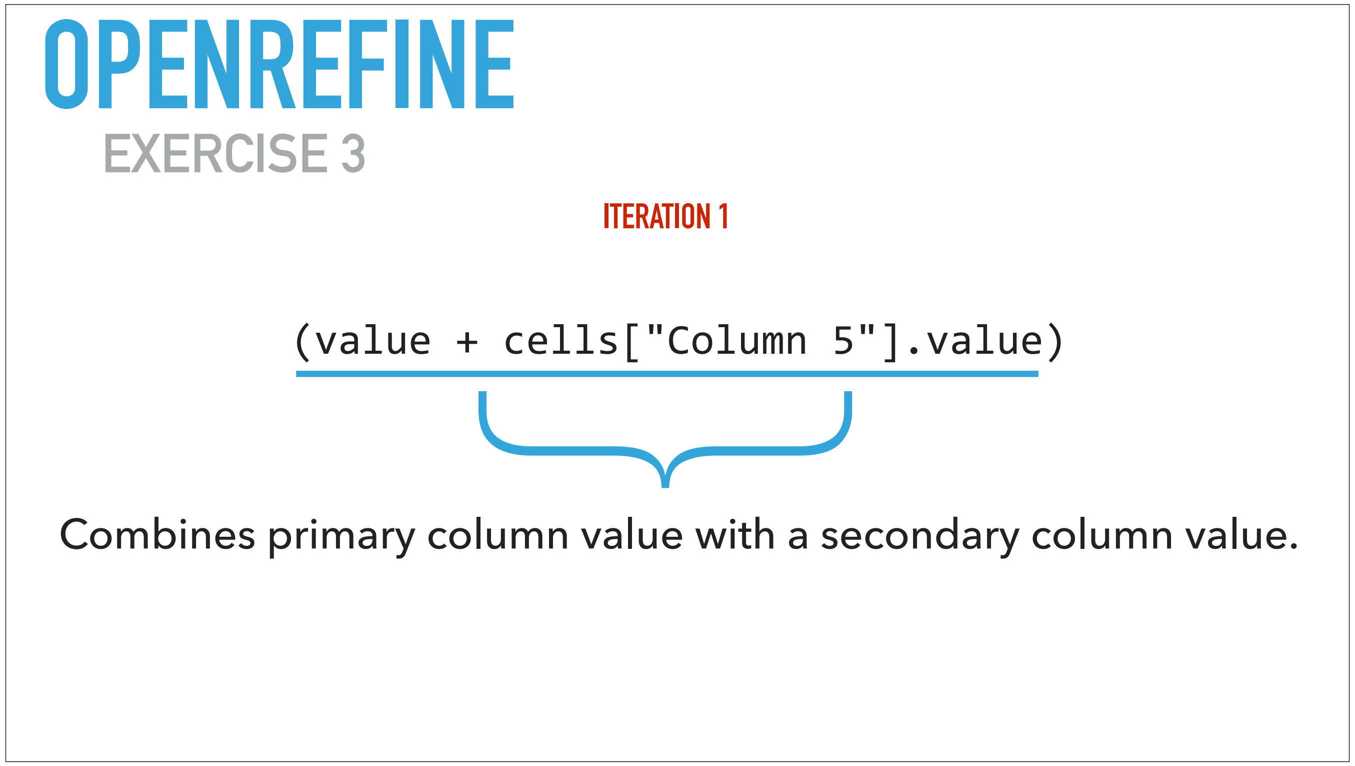
OPENREFINE

EXERCISE 3

OBJECTIVE 2:



Add column based on this column.
Yours will probably look different because I have plugins enabled.



OPENREFINE

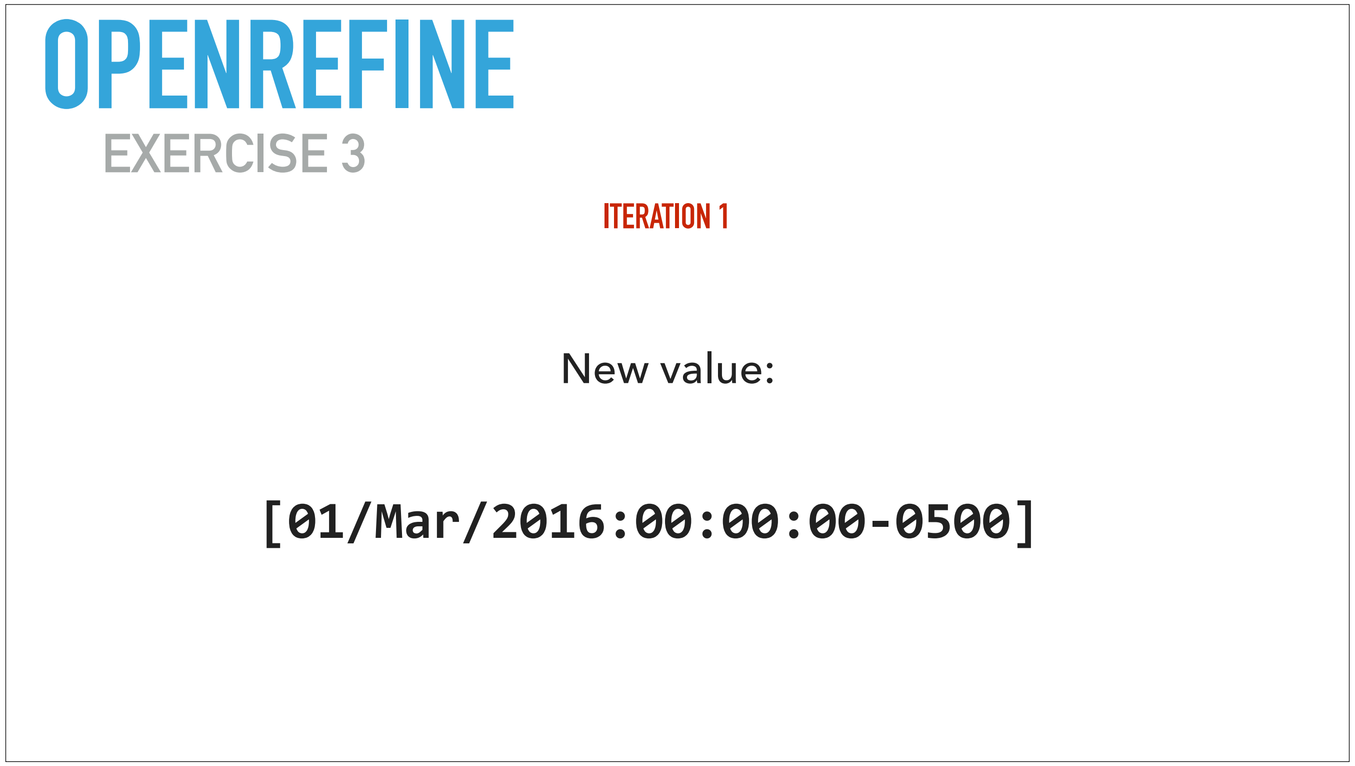
EXERCISE 3

ITERATION 1

```
(value + cells["Column 5"].value)
```

Combines primary column value with a secondary column value.

Explain the Expression.
Congratulations! You’ve just learned how to access values from other cells.



OPENREFINE

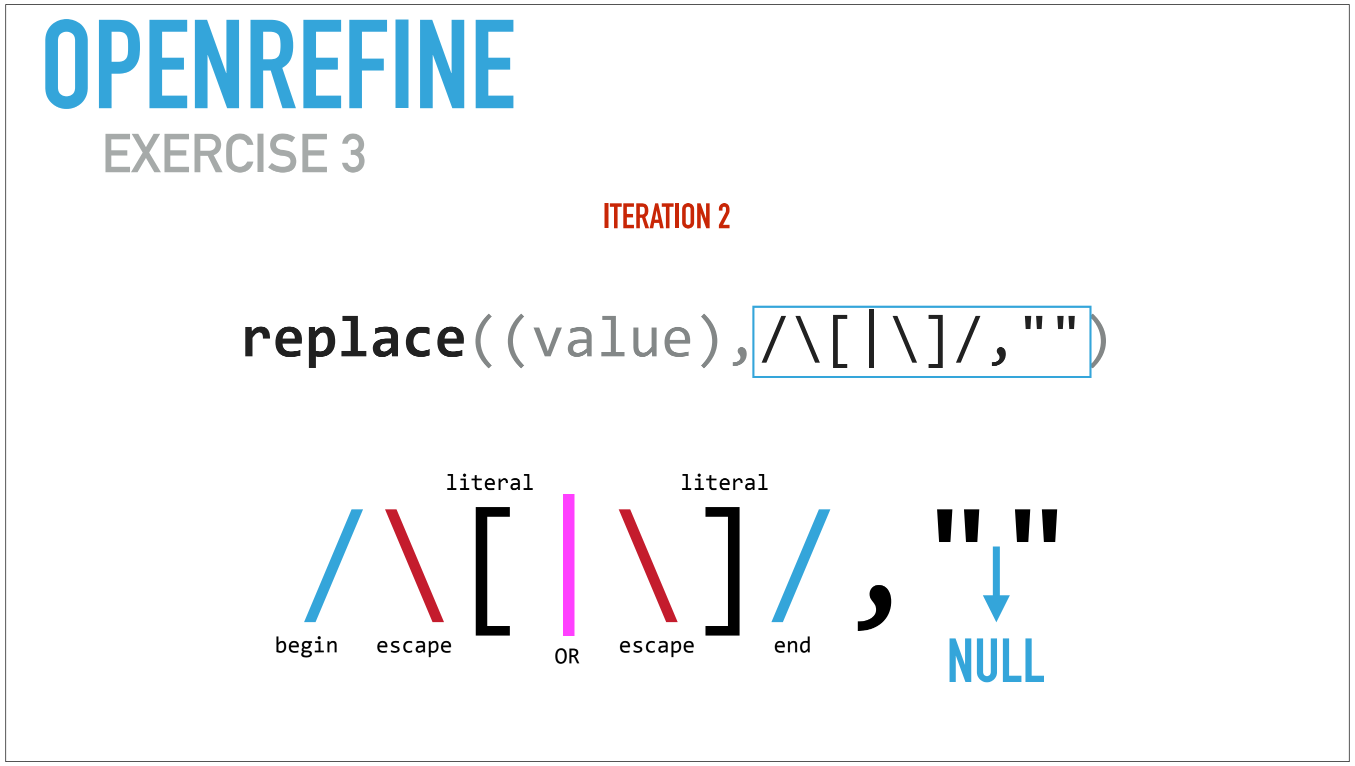
EXERCISE 3

ITERATION 1

New value:

```
[01/Mar/2016:00:00:00-0500]
```

The new column now has this new value.



OPENREFINE

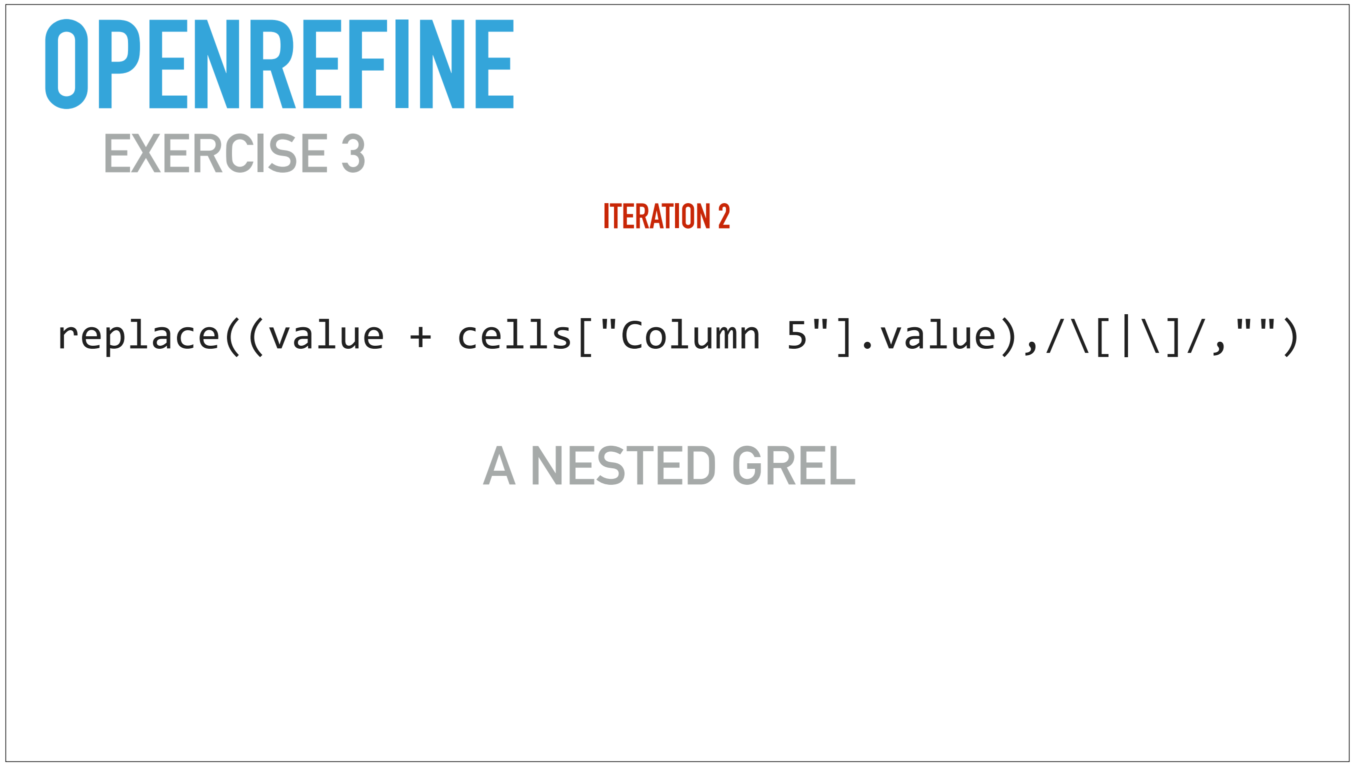
EXERCISE 3

ITERATION 2

```
replace((value),/\[|\]/,"")
```

```
/\[|\]/,""
```

Shield your eyes!



OPENREFINE

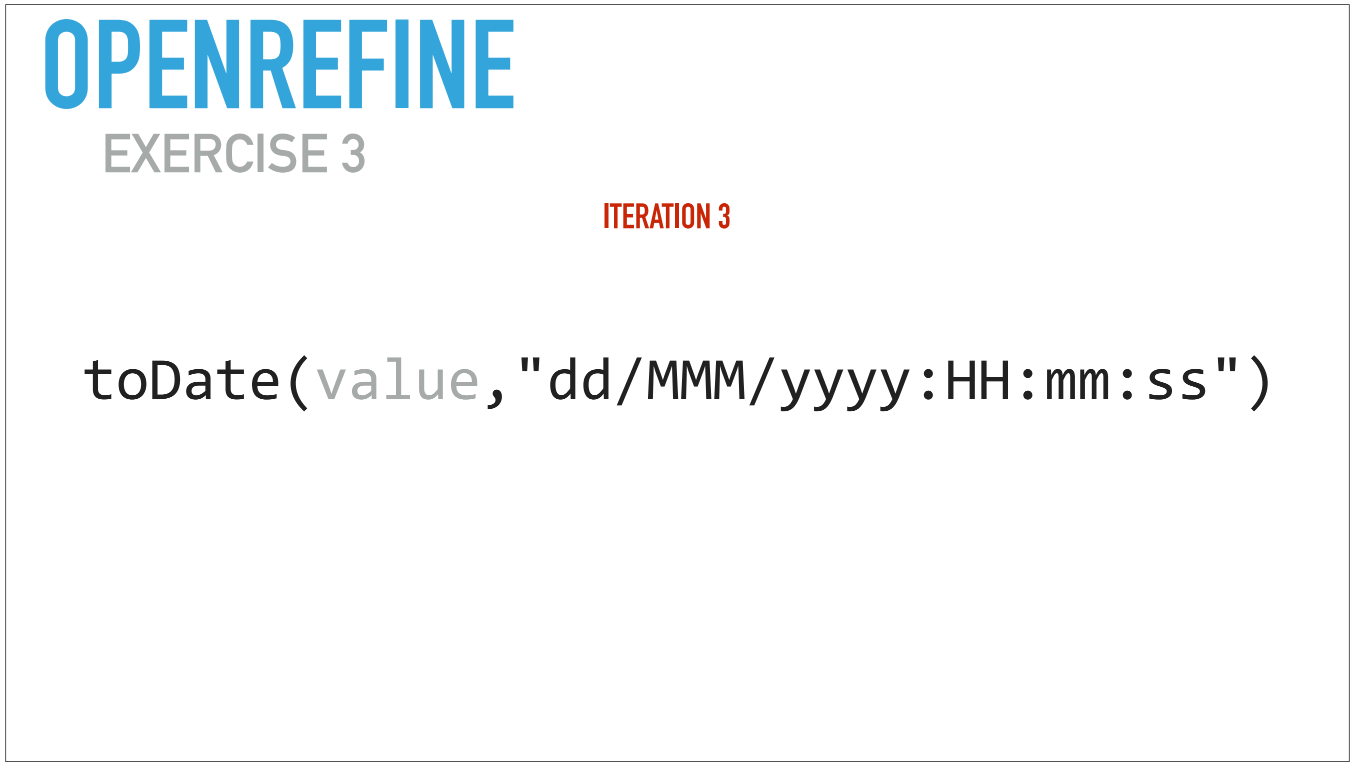
EXERCISE 3

ITERATION 2

```
replace((value + cells["Column 5"].value),/\[|\]/,"")
```

A NESTED GREL

Instead of adding another column, you can nest the GREL.
(The GREL, with the previous two operations nested.)



OPENREFINE

EXERCISE 3

ITERATION 3

```
toDate(value,"dd/MMM/yyyy:HH:mm:ss")
```

Do you want to try this one on your own?
Again, *value* is the previous expression.

OPENREFINE

EXERCISE 3

```
toDate((replace((value + cells["Column 5"].value),/\[|\]/,"")),"dd/MMM/yyyy:HH:mm:ss")
```

Regular expressions don’t look good on slides.

¯_(ツ)_/¯

OPENREFINE

EXERCISE 3

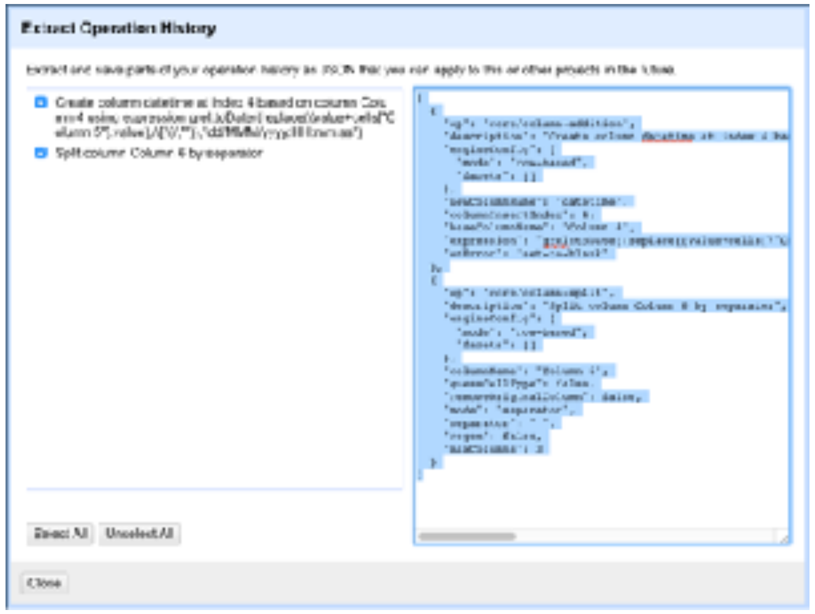
QUESTION:

Is there a way to automate that last objective?

OPENREFINE

EXERCISE 3

Save as .json



Extract Operation History

Extract and save parts of your operation history as JSON that you can apply to this or other projects in the future.

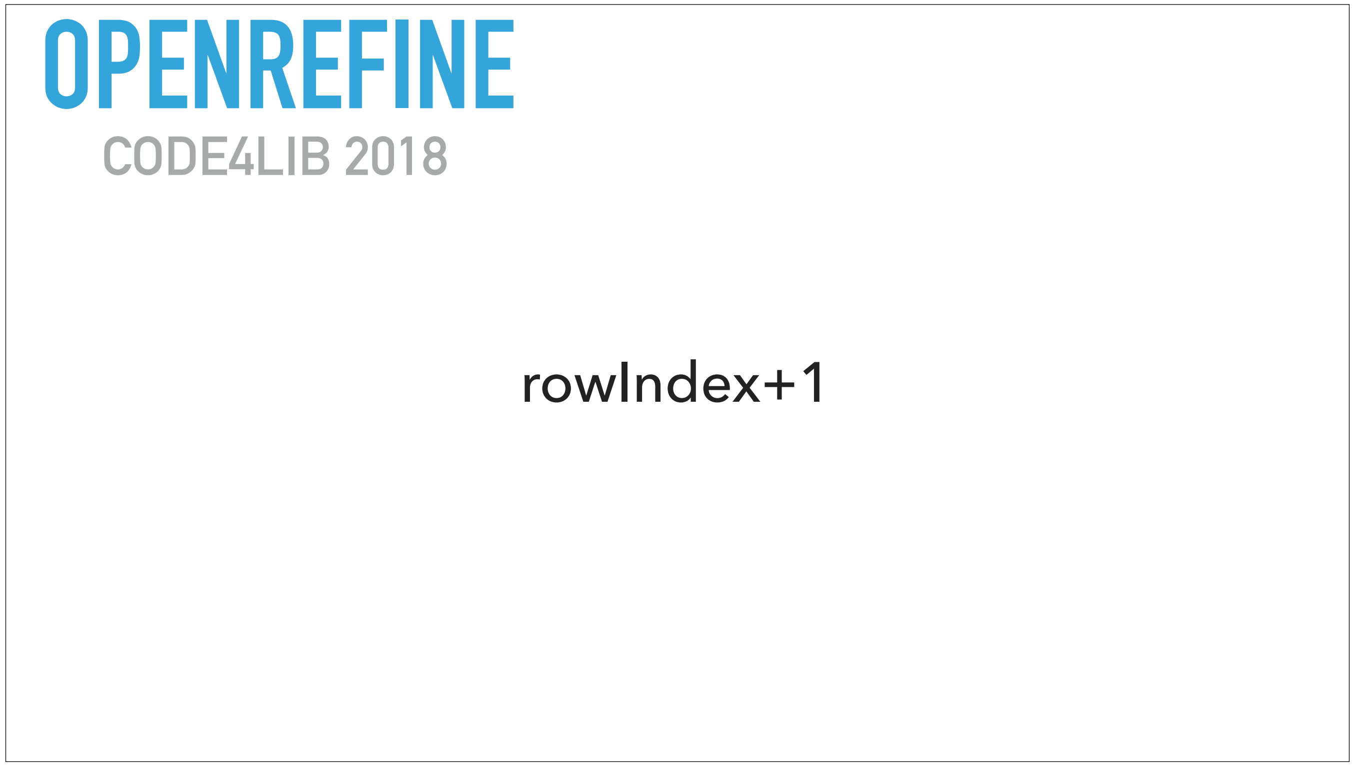
- ☑ Create column datetime at index 4 based on column Column 4 using expression grel:toDate(replace(value+cells["Column 5"].value,/[\/\-"]/,""),"dd/MM/yyyy HH:mm:ss")
- ☑ Split column Column 4 by separator

```
[
  {
    "op": "core/column-addition",
    "description": "Create column datetime at index 4 ba
    "engineConfig": {
      "mode": "row-based",
      "facets": []
    },
    "newColumnName": "datetime",
    "columnInsertIndex": 4,
    "baseColumnName": "Column 4",
    "expression": [illegible]
    "onError": "set-to-blank"
  },
  {
    "op": "core/column-split",
    "description": "Split column Column 4 by separator",
    "engineConfig": {
      "mode": "row-based",
      "facets": []
    },
    "columnName": "Column 4",
    "guessCellType": false,
    "removeOriginalColumn": false,
    "mode": "separator",
    "separator": " ",
    "regex": false,
    "maxColumns": 0
  }
]
```

OpenRefine lets you extract JSON for reuse in future projects.

OPENREFINE

CONCLUSION



OPENREFINE

CODE4LIB 2018

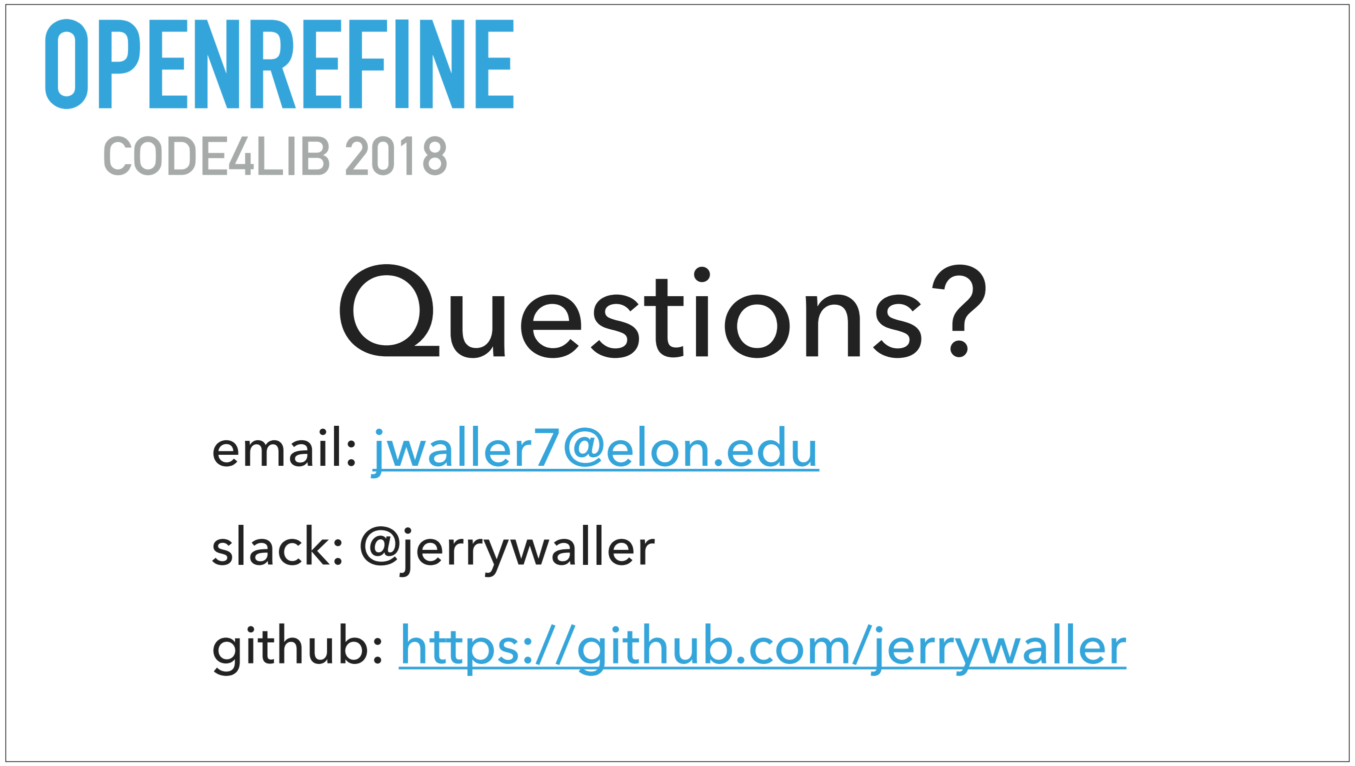
rowIndex+1

use rowIndex+1 to create cell values, because of zero-based numbering.

OPENREFINE

CODE4LIB 2018

Facets let you view and edit a subset of the data without affecting the entire dataset.



OPENREFINE

CODE4LIB 2018

Questions?

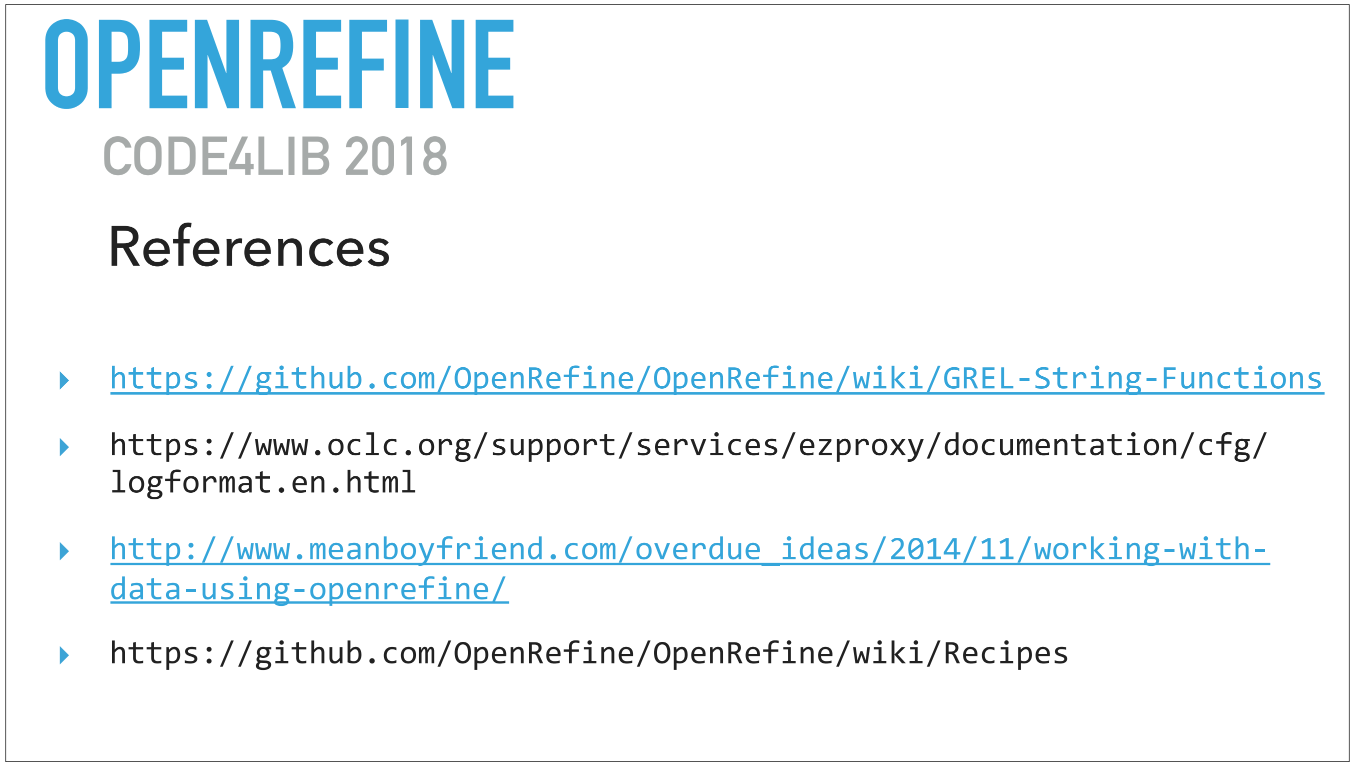
email: jwaller7@elon.edu

slack: @jerrywaller

github: https://github.com/jerrywaller

know your data :)

Any questions?



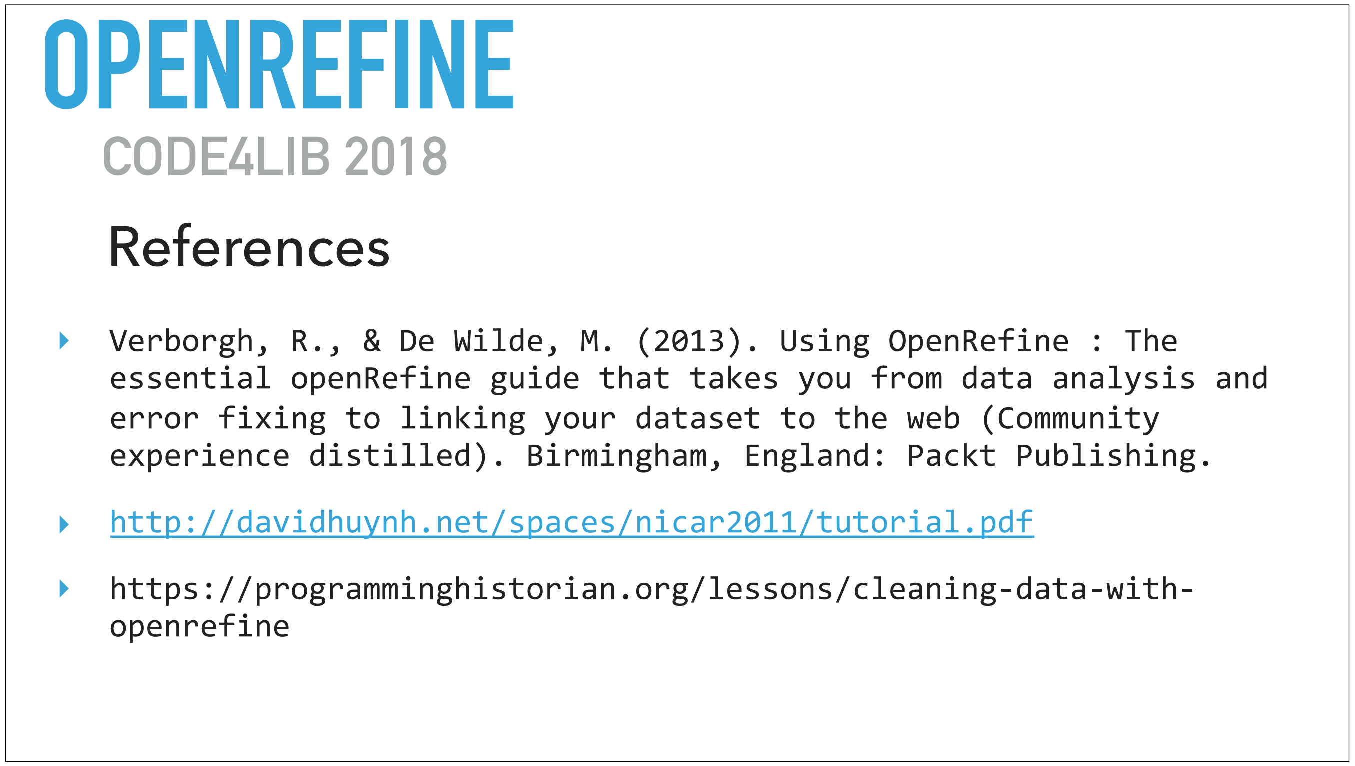
OPENREFINE

CODE4LIB 2018

References

- https://github.com/OpenRefine/OpenRefine/wiki/GREL-String-Functions
- https://www.oclc.org/support/services/ezproxy/documentation/cfg/logformat.en.html
- http://www.meanboyfriend.com/overdue_ideas/2014/11/working-with-data-using-openrefine/
- https://github.com/OpenRefine/OpenRefine/wiki/Recipes

Works cited



OPENREFINE

CODE4LIB 2018

References

- Verborgh, R., & De Wilde, M. (2013). Using OpenRefine : The essential openRefine guide that takes you from data analysis and error fixing to linking your dataset to the web (Community experience distilled). Birmingham, England: Packt Publishing.
- http://davidhuynh.net/spaces/nicar2011/tutorial.pdf
- https://programminghistorian.org/lessons/cleaning-data-with-openrefine

Works cited

CODE4LIB 2018

THANK YOU