Getting Started with Google Refine (now OpenRefine)

Start by downloading the .zip file located at: Refine

This download link is currently active, although the project is migrating to github.

Check with the projects main website if there is questions as to how to download and install from github. The site can be found at:

openrefine.org

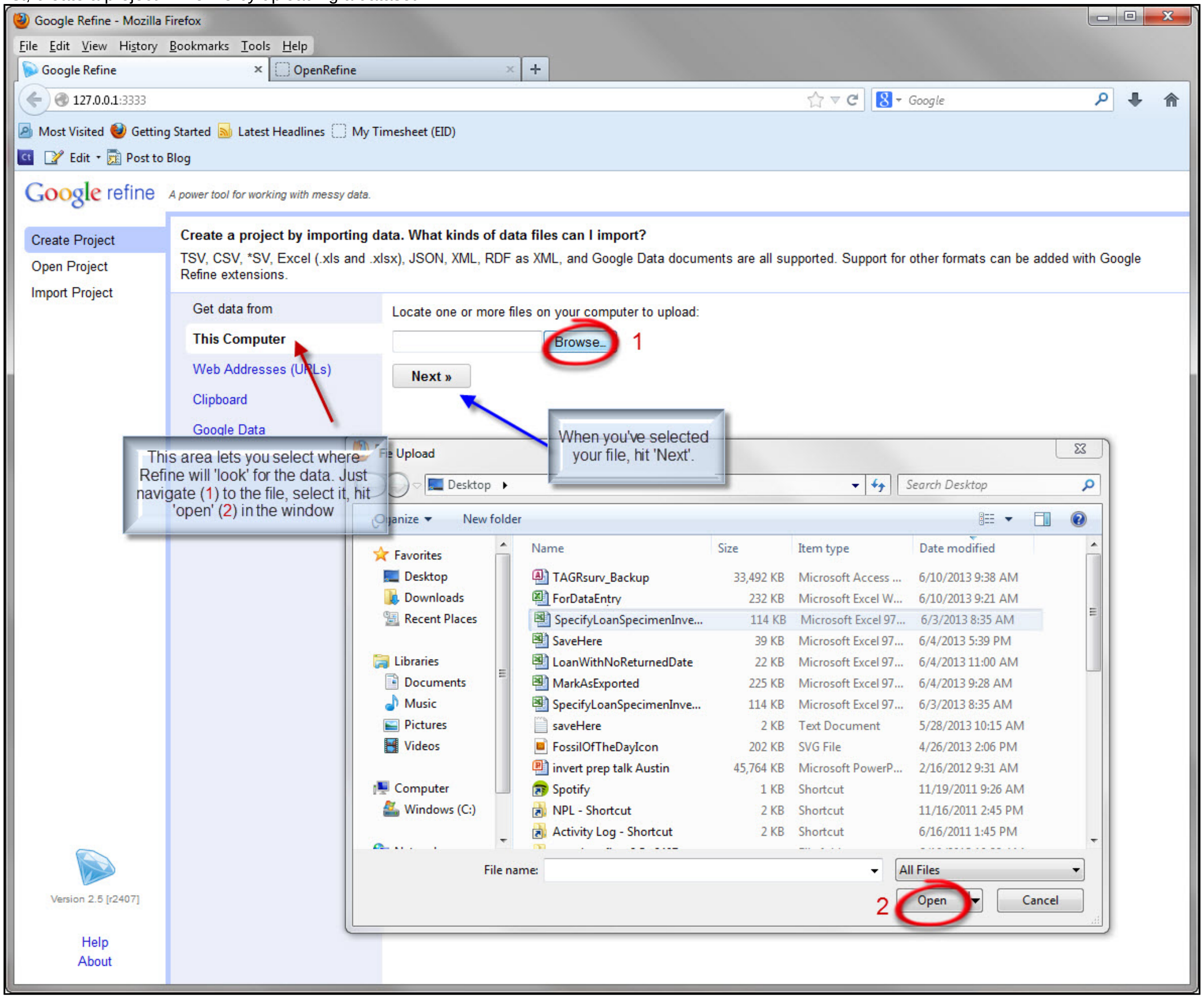
(best viewed on Firefox web browser)

Save the file to your desktop, and then double-click on the file *google-refine.exe*

This will launch Refine in a browser window. If it does not automatically launch, paste this address into the address bar of your web browser: http://127.0.0.1:3333/

Refine will work with many types of files. Because our end goal is to upload into Specify, keeping everything in spreadsheet form is preferred. Refine will work with both .xls and .xlsx files.

First, create a project in Refine by uploading a dataset.

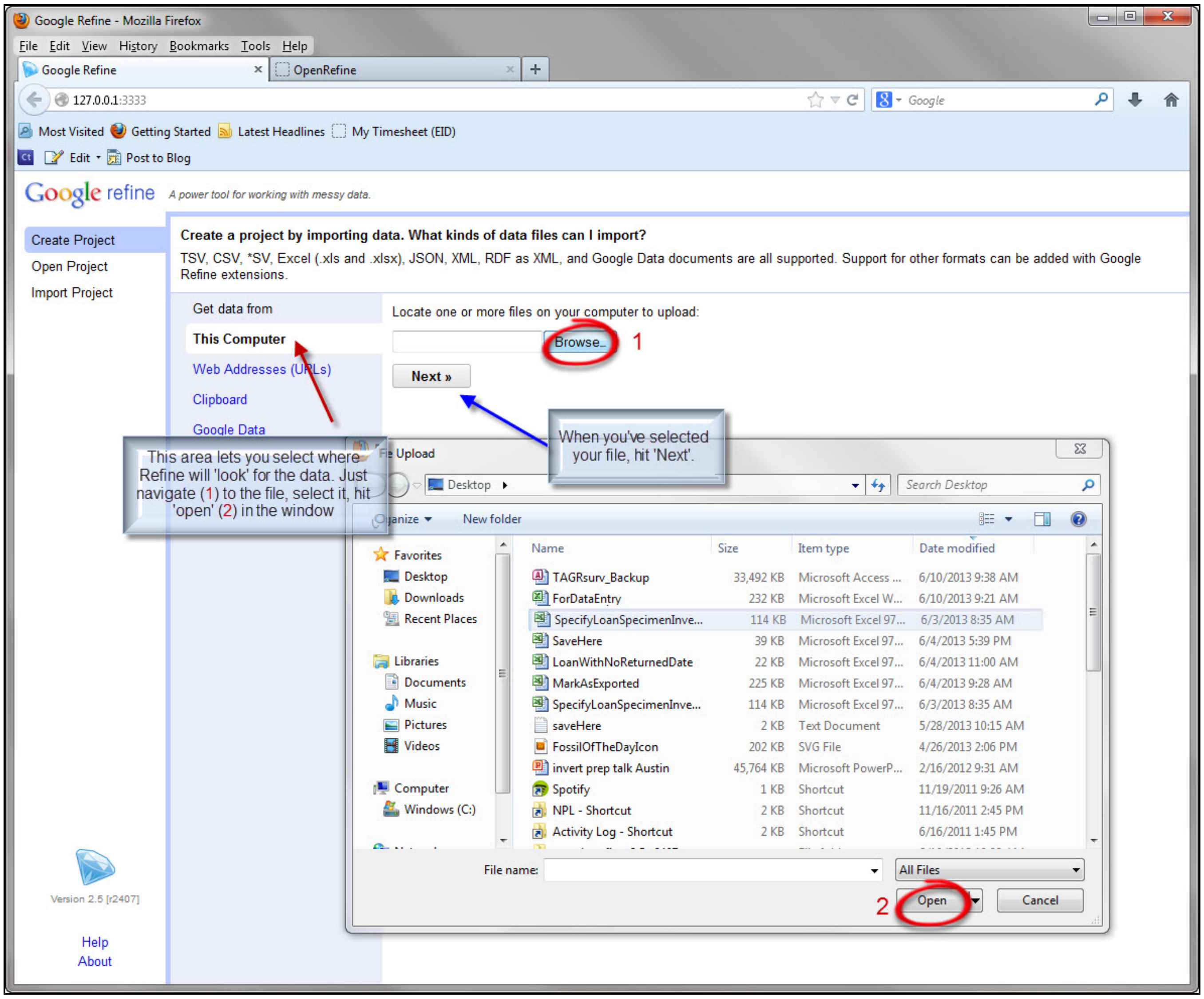


Create a project by importing data. What kinds of data files can I import?
TSV, CSV, *SV, Excel (.xls and .xlsx), JSON, XML, RDF as XML, and Google Data documents are all supported. Support for other formats can be added with Google Refine extensions.

This area lets you select where Refine will 'look' for the data. Just navigate (1) to the file, select it, hit 'open' (2) in the window

When you've selected your file, hit 'Next'.

Name	Size	Item type	Date modified
TAGRsurv_Backup	33,492 KB	Microsoft Access ...	6/10/2013 9:38 AM
ForDataEntry	232 KB	Microsoft Excel W...	6/10/2013 9:21 AM
SpecifyLoanSpecimenInve...	114 KB	Microsoft Excel 97...	6/3/2013 8:35 AM
SaveHere	39 KB	Microsoft Excel 97...	6/4/2013 5:39 PM
LoanWithNoReturnedDate	22 KB	Microsoft Excel 97...	6/4/2013 11:00 AM
MarkAsExported	225 KB	Microsoft Excel 97...	6/4/2013 9:28 AM
SpecifyLoanSpecimenInve...	114 KB	Microsoft Excel 97...	6/3/2013 8:35 AM
saveHere	2 KB	Text Document	5/28/2013 10:15 AM
FossilOfTheDayIcon	202 KB	SVG File	4/26/2013 2:06 PM
invert prep talk Austin	45,764 KB	Microsoft PowerP...	2/16/2012 9:31 AM
Spotify	1 KB	Shortcut	11/19/2011 9:26 AM
NPL - Shortcut	2 KB	Shortcut	11/16/2011 2:45 PM
Activity Log - Shortcut	2 KB	Shortcut	6/16/2011 1:45 PM



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

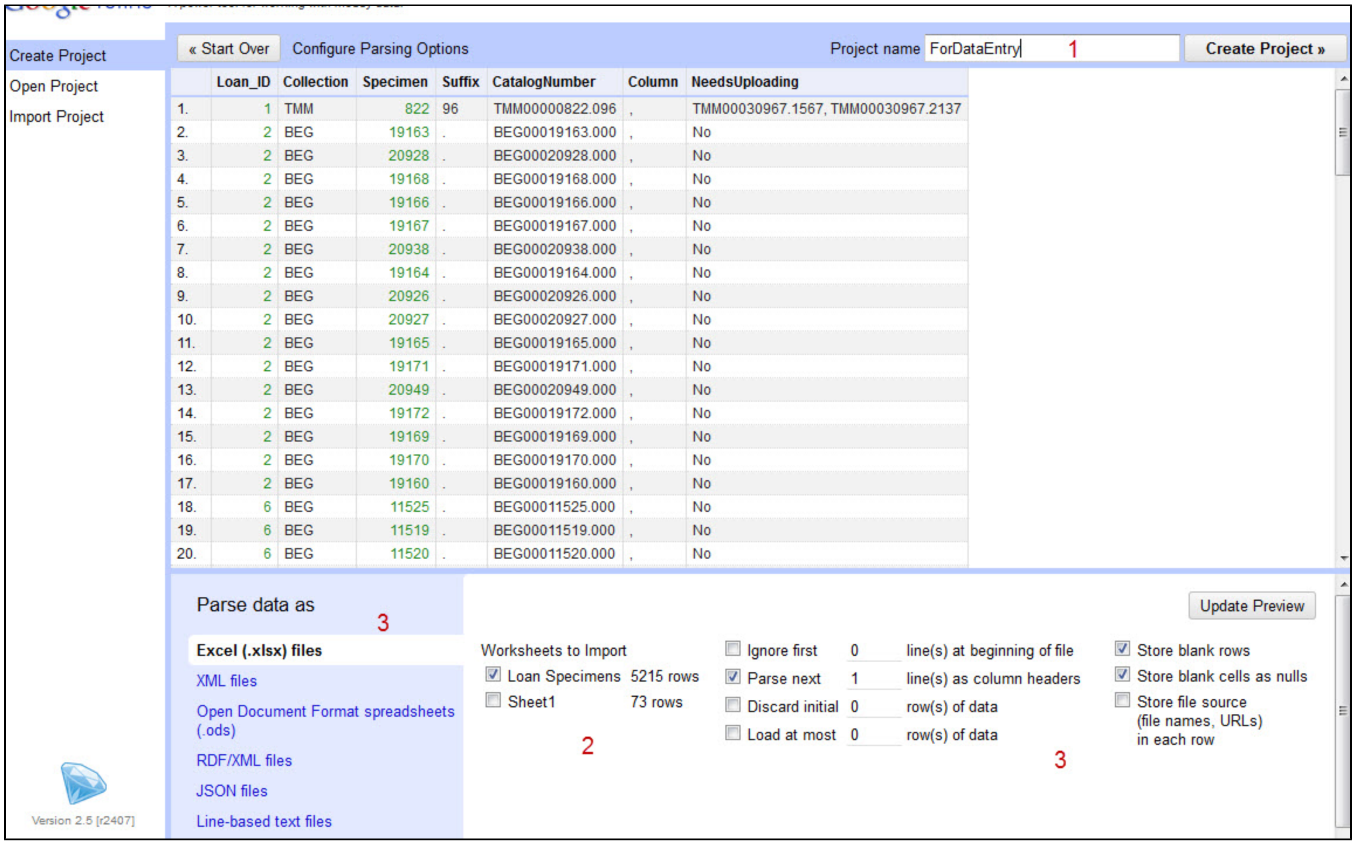
TSV, CSV, *SV, Excel (.xls and .xlsx), JSON, XML, RDF as XML, and Google Data documents are all supported. Support for other formats can be added with Google Refine extensions.

This area lets you select where Refine will 'look' for the data. Just navigate (1) to the file, select it, hit 'open' (2) in the window

When you've selected your file, hit 'Next'.

Name	Size	Item type	Date modified
TAGRsurv_Backup	33,492 KB	Microsoft Access ...	6/10/2013 9:38 AM
ForDataEntry	232 KB	Microsoft Excel W...	6/10/2013 9:21 AM
SpecifyLoanSpecimenInve...	114 KB	Microsoft Excel 97...	6/3/2013 8:35 AM
SaveHere	39 KB	Microsoft Excel 97...	6/4/2013 5:39 PM
LoanWithNoReturnedDate	22 KB	Microsoft Excel 97...	6/4/2013 11:00 AM
MarkAsExported	225 KB	Microsoft Excel 97...	6/4/2013 9:28 AM
SpecifyLoanSpecimenInve...	114 KB	Microsoft Excel 97...	6/3/2013 8:35 AM
saveHere	2 KB	Text Document	5/28/2013 10:15 AM
FossilOfTheDayIcon	202 KB	SVG File	4/26/2013 2:06 PM
invert prep talk Austin	45,764 KB	Microsoft PowerP...	2/16/2012 9:31 AM
Spotify	1 KB	Shortcut	11/19/2011 9:26 AM
NPL - Shortcut	2 KB	Shortcut	11/16/2011 2:45 PM
Activity Log - Shortcut	2 KB	Shortcut	6/16/2011 1:45 PM

Once Refine has finished verifying your data, it gives you an intermediary screen that allows you to name your project (1), select which worksheets get imported (2), and some data-handling options (3). Select "Create Project" when you are satisfied with dataset.



Configure Parsing Options

Project name: ForDataEntry

	Loan_ID	Collection	Specimen	Suffix	CatalogNumber	Column	NeedsUploading
1.	1	TMM	822	96	TMM00000822.096	,	TMM00030967.1567, TMM00030967.2137
2.	2	BEG	19163	.	BEG00019163.000	,	No
3.	2	BEG	20928	.	BEG00020928.000	,	No
4.	2	BEG	19168	.	BEG00019168.000	,	No
5.	2	BEG	19166	.	BEG00019166.000	,	No
6.	2	BEG	19167	.	BEG00019167.000	,	No
7.	2	BEG	20938	.	BEG00020938.000	,	No
8.	2	BEG	19164	.	BEG00019164.000	,	No
9.	2	BEG	20926	.	BEG00020926.000	,	No
10.	2	BEG	20927	.	BEG00020927.000	,	No
11.	2	BEG	19165	.	BEG00019165.000	,	No
12.	2	BEG	19171	.	BEG00019171.000	,	No
13.	2	BEG	20949	.	BEG00020949.000	,	No
14.	2	BEG	19172	.	BEG00019172.000	,	No
15.	2	BEG	19169	.	BEG00019169.000	,	No
16.	2	BEG	19170	.	BEG00019170.000	,	No
17.	2	BEG	19160	.	BEG00019160.000	,	No
18.	6	BEG	11525	.	BEG00011525.000	,	No
19.	6	BEG	11519	.	BEG00011519.000	,	No
20.	6	BEG	11520	.	BEG00011520.000	,	No

Parse data as

- **Excel (.xlsx) files**
- XML files
- Open Document Format spreadsheets (.ods)
- RDF/XML files
- JSON files
- Line-based text files

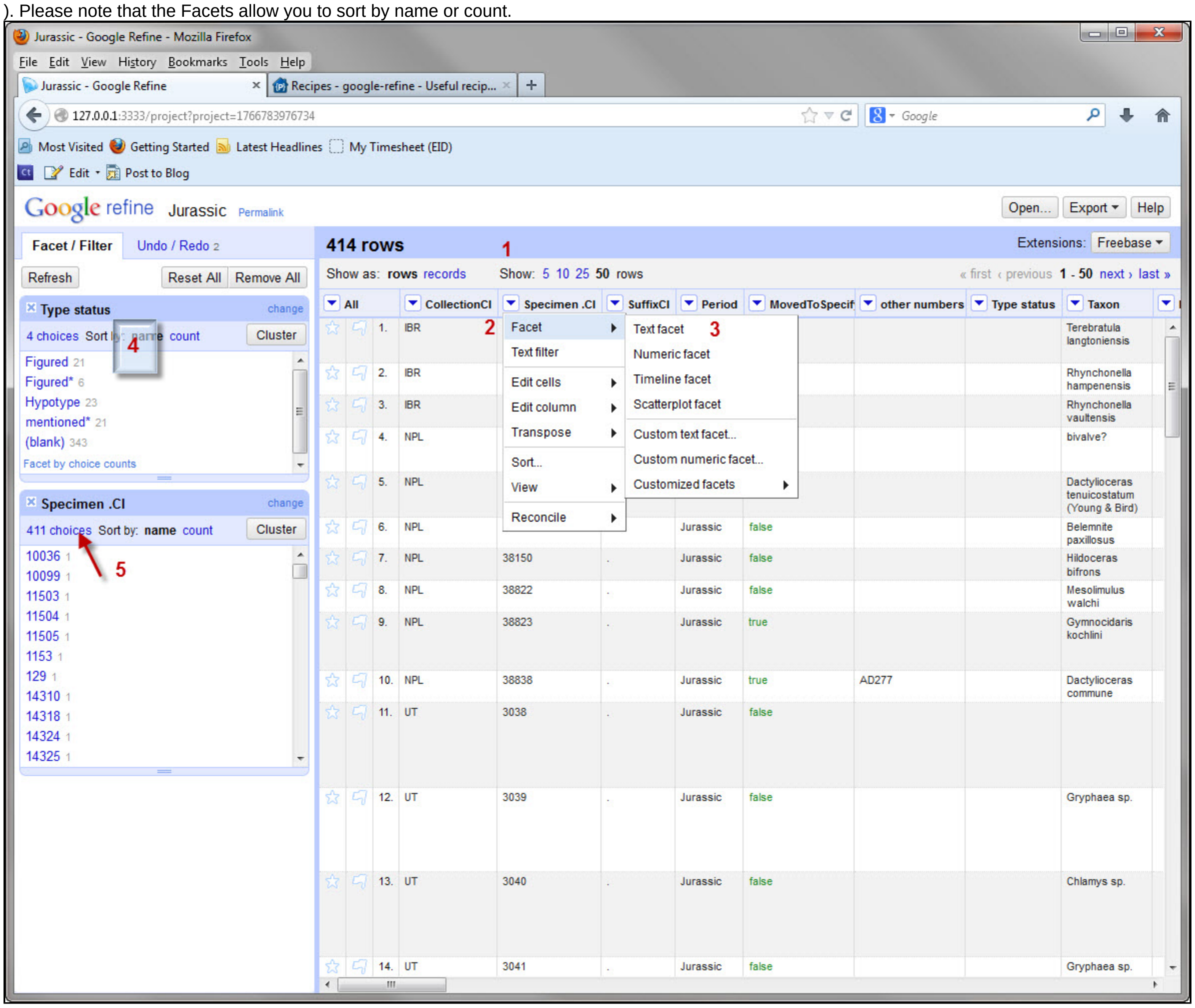
Worksheets to Import
- ✓ Loan Specimens 5215 rows
- ☐ Sheet1 73 rows

- ☐ 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

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

Refine, as a default, displays 10 rows. You can have it display up to 50 but not more (1). Refine is not a tool for modifying data within cells one at a time. It is best used for dealing with whole swaths of data. Refine does that by a tool called 'facet' (2), which is an option you find by clicking on the down-arrow on which ever column you wish to facet. Faceting data is like a filter for selecting data that meets a certain criteria- it can be a word, length of an entry, or just lumping data into how many times it occurs. You can also facet many rows at once, to get a very precise set of data which you can then act on. In the example below, the facet was set to 'text facet' (3). Faceting the column this way shows the data in the cells (4), and how many times that data is used .

The column 'Type status' (4) will only have a handful of variety (4 choices in this case). Something like 'Collection Number' would have many- 411 choices (
). Please note that the Facets allow you to sort by name or count.



Type status change

4 choices Sort by: name count Cluster

Figured	21
Figured*	6
Hypotype	23
mentioned*	21
(blank)	343

Facet by choice counts

Specimen .CI change

411 choices Sort by: name count Cluster

10036	1
10099	1
11503	1
11504	1
11505	1
1153	1
129	1
14310	1
14318	1
14324	1
14325	1

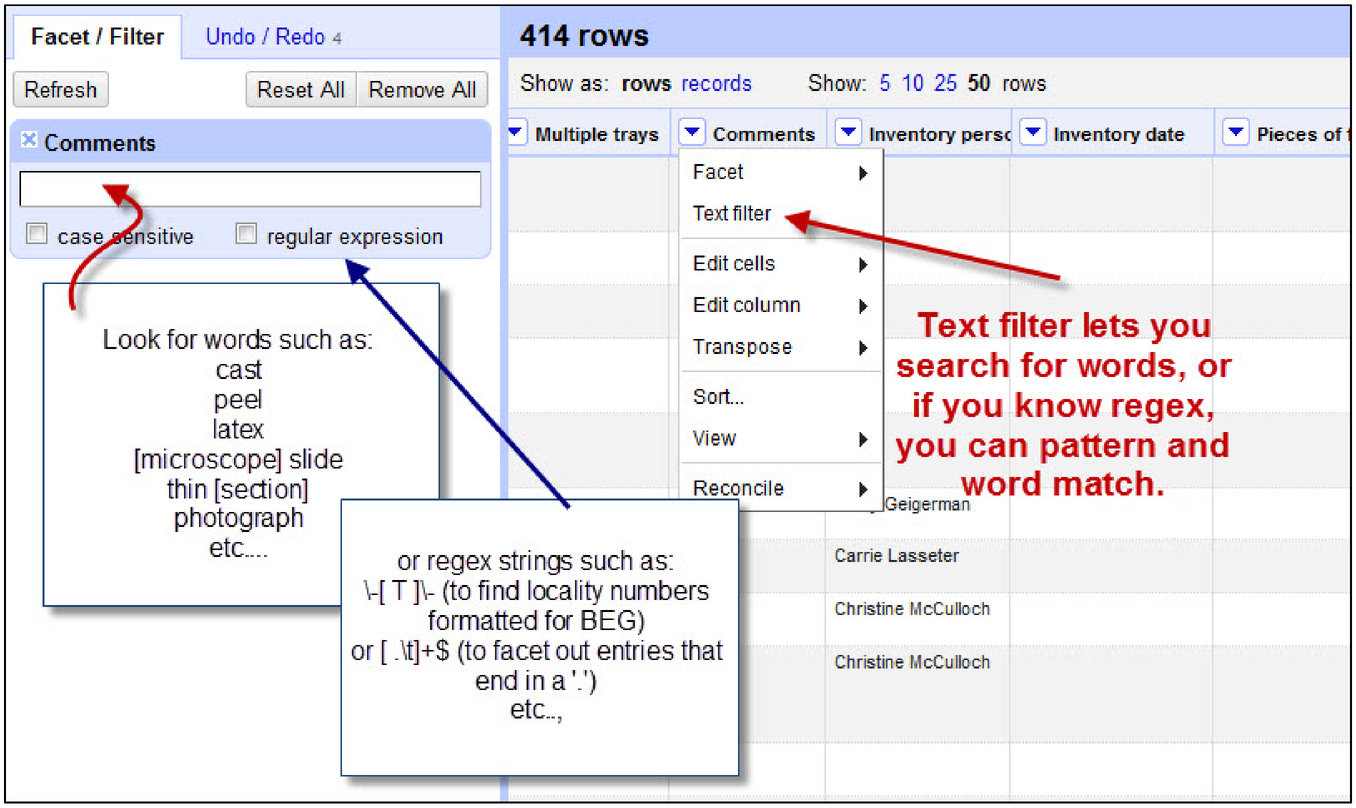
414 rows

Show as: rows records Show: 5 10 25 50 rows « first ‹ previous 1 - 50 next › last »

All	CollectionCI	Specimen .CI	SuffixCI	Period	MovedToSpecif	other numbers	Type status	Taxon
1.	IBR							Terebratula langtoniensis
2.	IBR							Rhynchonella hampenensis
3.	IBR							Rhynchonella vaultensis
4.	NPL							bivalve?
5.	NPL							Dactylioceras tenuicostatum (Young & Bird)
6.	NPL			Jurassic	false			Belemnite paxillosus
7.	NPL	38150	.	Jurassic	false			Hildoceras bifrons
8.	NPL	38822	.	Jurassic	false			Mesolimulus walchi
9.	NPL	38823	.	Jurassic	true			Gymnocidaris kochlini
10.	NPL	38838	.	Jurassic	true	AD277		Dactylioceras commune
11.	UT	3038	.	Jurassic	false			
12.	UT	3039	.	Jurassic	false			Gryphaea sp.
13.	UT	3040	.	Jurassic	false			Chlamys sp.
14.	UT	3041	.	Jurassic	false			Gryphaea sp.

Taking a closer look at the "Type Status" facet we see many entries that won't upload into Specify. There are 21 entries that read "mentioned*", and six that say "Figured*". By clicking on the 'edit' option that appears when we hover over that selection, the edit box appears and we can change all 6 entries at once. The number of choices now becomes 3, and the number of entries that say 'Figured' has gone from 21 to 27. We can do the same for the entries 'mentioned*'. Specify expects to see 'referred', not 'mentioned*', so we use this same process to change those records. There are many other columns where this can be done, also- Building (adding 'Building' to 122 and 33), you can also use 'edit' to add things to the blanks, such as adding 'dry' to all the blank entries in 'prep type' and so on. Facet by name also helps identify typos and misspelling ('Texas' has 300 entries, while Texsa has 4)

Faceting by words can help mine data out of comments fields. From the drop down on the column you are going to be mining, select 'Facet' then 'Customized Facets' and from that sub menu, select 'Word facet'. You can also facet for patterns (like lat/long entries in hours, min, seconds) using regex.

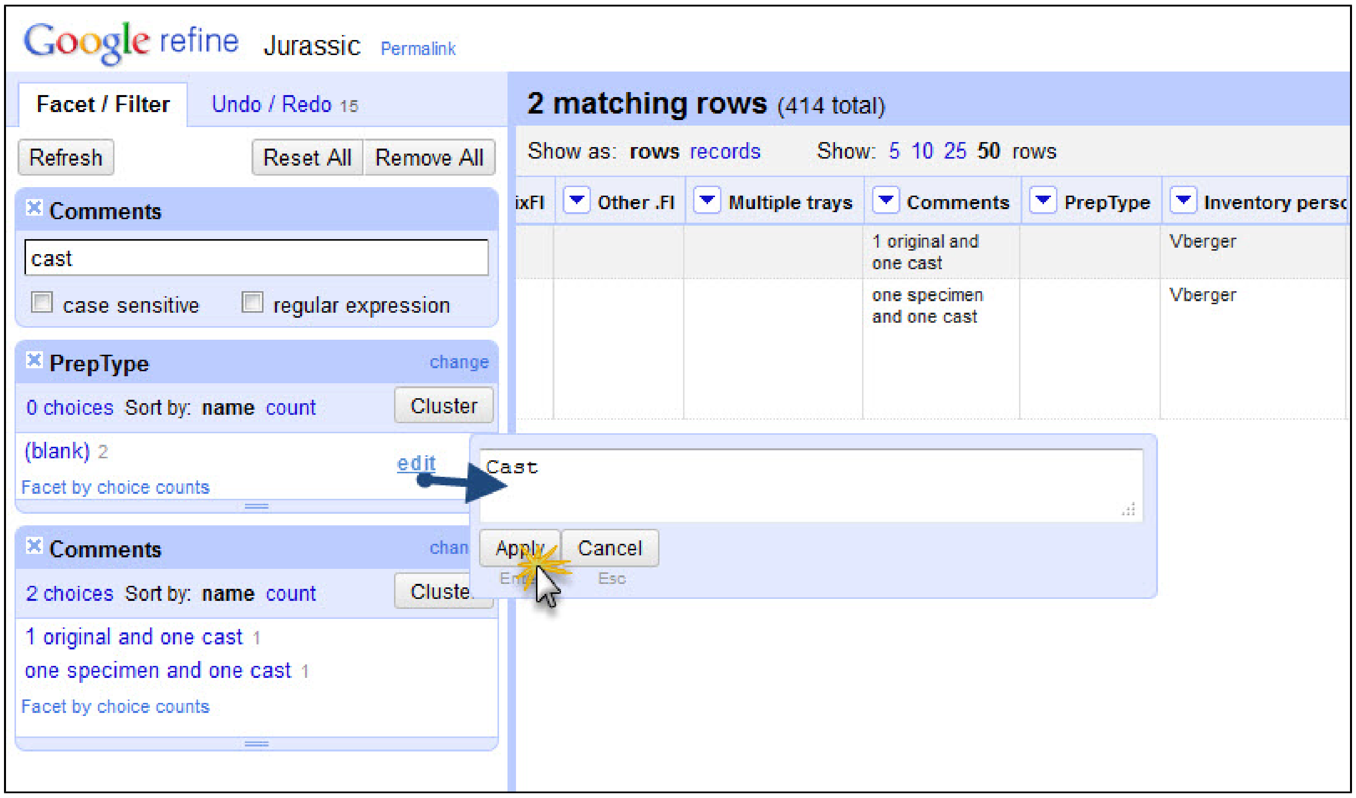


Look for words such as:
cast
peel
latex
[microscope] slide
thin [section]
photograph
etc....

or regex strings such as:
-[T]\- (to find locality numbers formatted for BEG)
or [.\t]+$ (to facet out entries that end in a '.')
etc..,

Text filter lets you search for words, or if you know regex, you can pattern and word match.

Moving the data is easy- faceting the Comments column for entries with the word 'cast' shows us which entries were noted in their inventory as being 'casts'. We can see in the Text Facet of the Comments column that the word 'cast' is being used to mean a cast of a specimen. We can then see the records in the PrepType Text Facet is blank for those records. Edit the blank cells in bulk using the edit option in the facet display box. Change (blank) to Cast, and apply the change. This process can be done on any type of data.



When preparing a spreadsheet for upload to Specify, there are many instances where you'll need to combine data from many columns into one, or transpose data from one column into another. Refine makes this very simple, using the 'Transform' option. For example, we are putting the previous storage location information into the Comments section, and renaming this column 'Inventory Remarks' (the field name in Specify). Renaming the columns can be done via the 'Edit Column' drop down, but taking the information from 3 fields and adding it to another field is a little trickier. For this, we will need to combine a few tricks we got from Refine Recipes. The recipe for Merging columns is:

cells["col1"].value + ", " + cells["col2"].value

cells = each cell in whatever is inside the [].

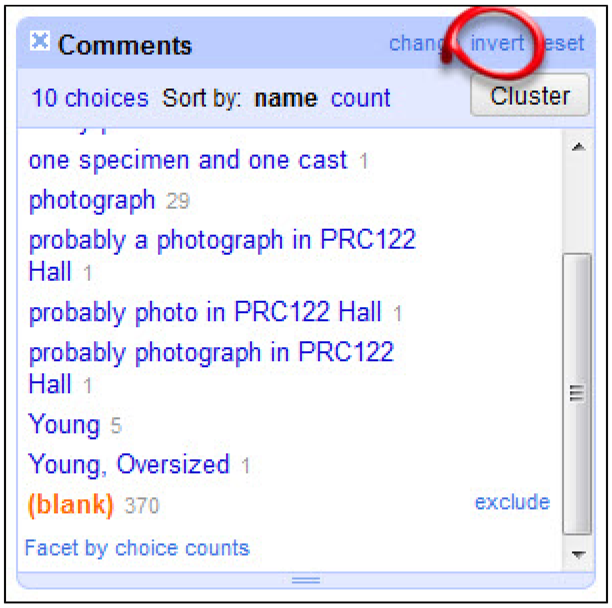
"col1" = the name of first column you want to combine

.value = tells Refine to get the exact value of the cells.

+ = combine this value with
", " = telling Refine to add a comma and a space
+ = combine this value with
cells = (same as above)
"col2" = the name of the second column you want to combine
.value = tells Refine to get the exact value of the cells.

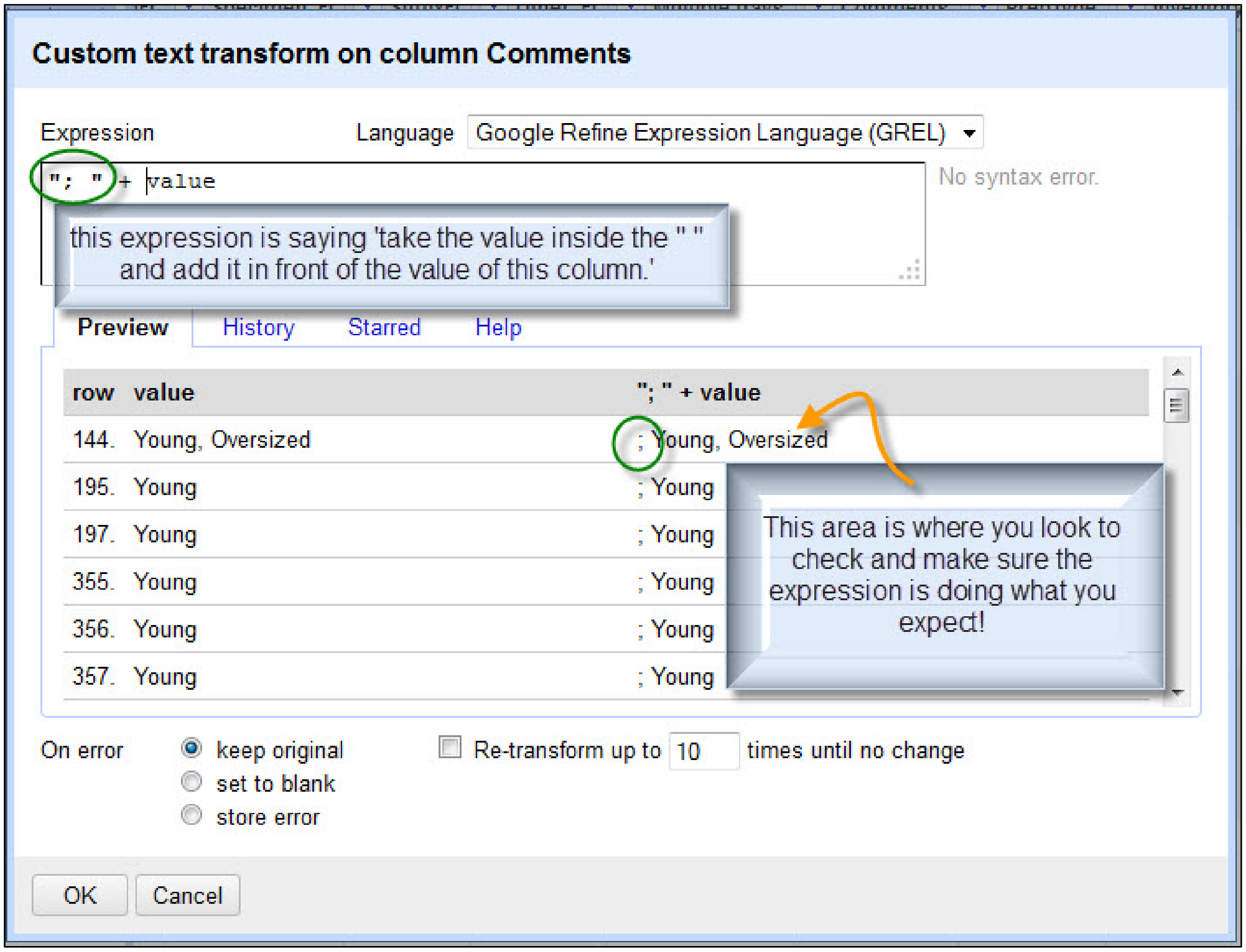
So, it's a coded sentence that is telling Refine "for each cell of this column, take that value and add a ", " to it. Then, take the cells in this other column and add the values to behind the ", ".

First things first though. You are going to want to add a separator in front of any data that is already in the Comments Column. To do this, we'll be using facets to act ONLY on the cells with data in them, and the Transform command to add a character in front of the comments. To do this, we create a facet on the comments section, hover over the (blank) and click on the "include" option. Then select the 'invert' option near the top of the facet title.



one specimen and one cast	1
photograph	29
probably a photograph in PRC122 Hall	1
probably photo in PRC122 Hall	1
probably photograph in PRC122 Hall	1
Young	5
Young, Oversized	1
(blank)	370

This 'flips' the selection, and now you should see only the cells with data in them. From the Comments drop down, select Edit Cells and from the sub menu, select Transform. A new window will appear. Adding information is very simple. Make sure the Language is set to the default value, Google Refine Expression Language (GREL). We are going to add a semi-colon and a space before the existing data.


Custom text transform on column Comments

Expression: `"; " + value`

Language: Google Refine Expression Language (GREL)

No syntax error.

this expression is saying 'take the value inside the " " and add it in front of the value of this column.'

row	value	"; " + value
144.	Young, Oversized	; Young, Oversized
195.	Young	; Young
197.	Young	; Young
355.	Young	; Young
356.	Young	; Young
357.	Young	; Young

This area is where you look to check and make sure the expression is doing what you expect!

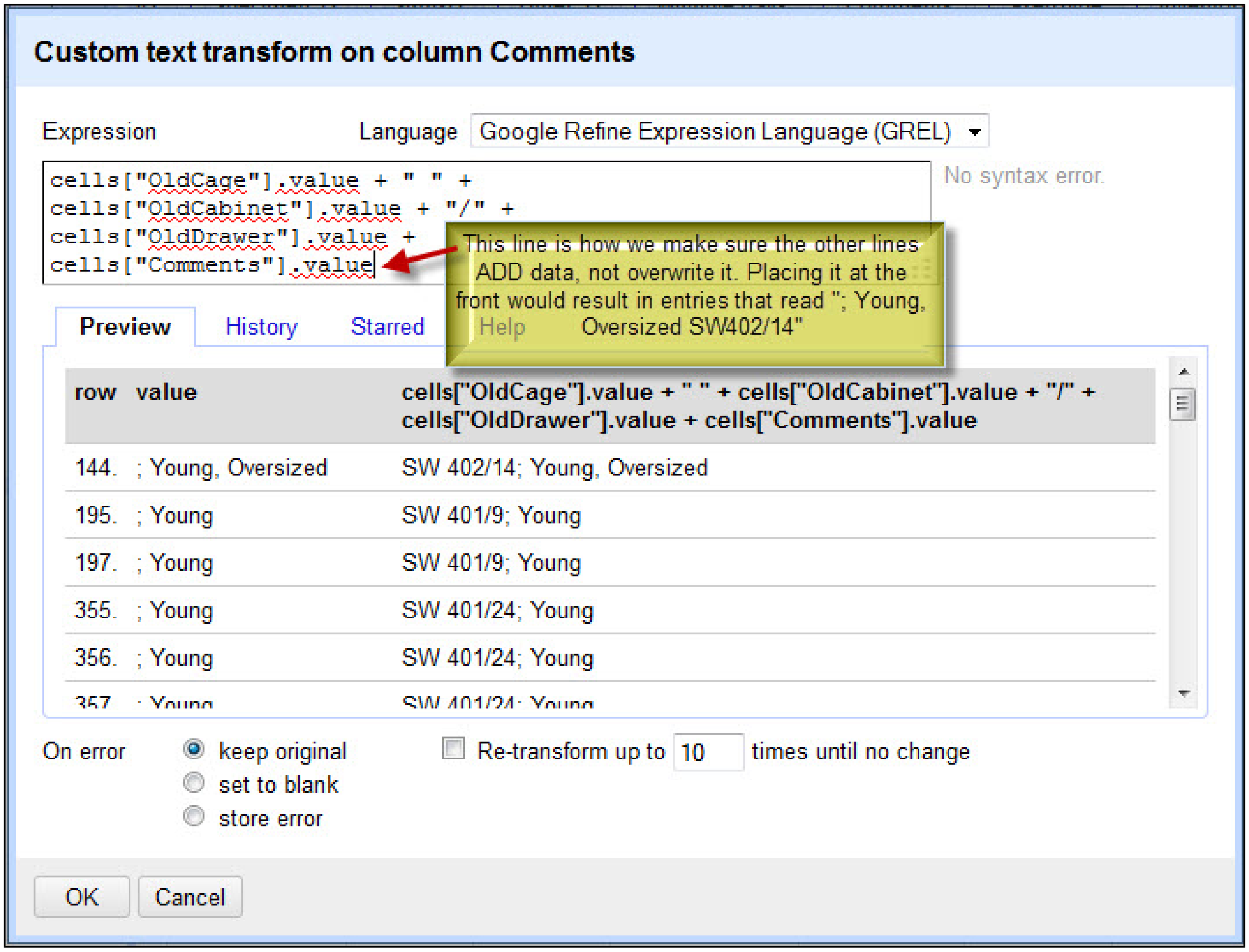
On error: [✓ keep original] set to blank / store error

☐ Re-transform up to 10 times until no change

Now that we've done that, we can add the previous storage location without ending up with statments in the field that read "SW 269/9Young, Oversized"!

Remember, when we are ADDING data to another column, you have to tell Refine to also add the data from the cells in the column you are working in.

In practice, it looks like this:



Custom text transform on column Comments

Expression Language Google Refine Expression Language (GREL)

```
cells["OldCage"].value + " " +
cells["OldCabinet"].value + "/" +
cells["OldDrawer"].value +
cells["Comments"].value
```

No syntax error.

This line is how we make sure the other lines ADD data, not overwrite it. Placing it at the front would result in entries that read "; Young, Oversized SW402/14"

row	value	cells["OldCage"].value + " " + cells["OldCabinet"].value + "/" + cells["OldDrawer"].value + cells["Comments"].value
144.	; Young, Oversized	SW 402/14; Young, Oversized
195.	; Young	SW 401/9; Young
197.	; Young	SW 401/9; Young
355.	; Young	SW 401/24; Young
356.	; Young	SW 401/24; Young
357.	; Young	SW 401/24; Young

On error [✓ keep original] set to blank store error

☐ Re-transform up to 10 times until no change

The same 'recipe' is the foundation for how we take our old database numbering style and make it Specify compatible. But first, we have to use the recipe for padding with 0's.

The recipe is:

```
"0000"[0,4-value.length()] + value
```

The recipe says to take the value, add as many 0's as are needed to make the field have 4 values stored in it. This will add 4 0's to our value. If we need more, we increase the number of 0's to what we need, then change the number inside the brackets to match.

Specify expects to see 3 values for the Collection, 8 values for the specimen number, a decimal then 3 values for the suffix. Remember here that letters follow the .R00 format, and numbers .001. Suffixes like 'T6' would be set to .T06

To make the Collection have 3 digits, paste this expression into the Transform window for the Collection column-value

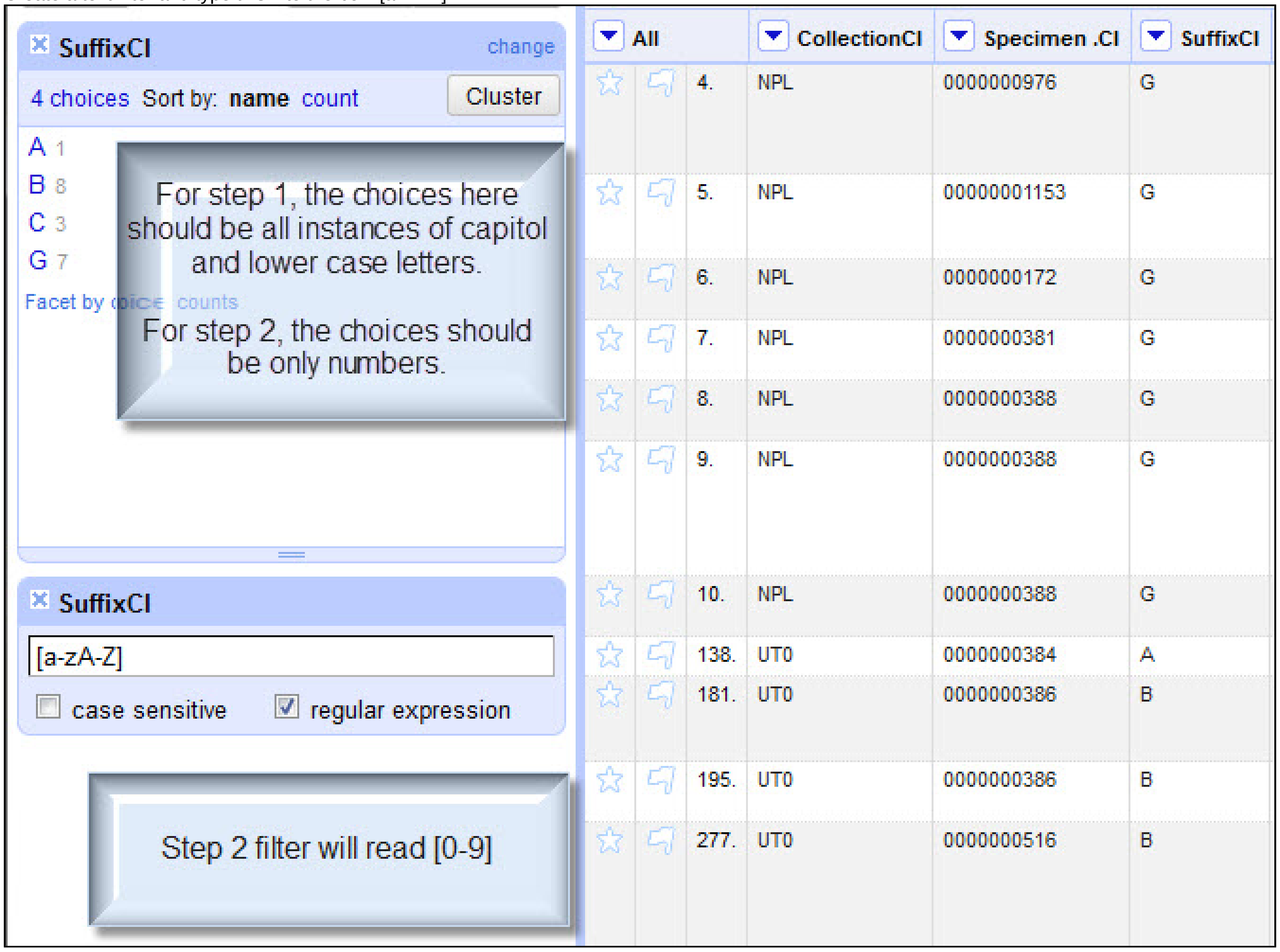
```
value +"000"[0,3-value.length()]
```

To make the Specimen number have 8 digits, paste this expression into the Transform window:

```
"0000000" + value[0,8-value.length()]
```

* Notice how the order has changed a little- the string of 0's is at the start, and the value is added to it. This gives us 0 padding in front of the number. For the Collection, we wanted the 0's to come at the end of the collection acronym so the expression structure was swapped around.

Changing the suffixes is a little more involved. First, facet the column. All the entries for '.' can be bulk edited to "000". Don't put a decimal in just yet. Create a text filter and type this into the box: [a-zA-Z]


SuffixCI
change
4 choices Sort by: name count
Cluster
A 1
B 8
C 3
G 7
Facet by choice counts
For step 1, the choices here should be all instances of capitol and lower case letters.
For step 2, the choices should be only numbers.
SuffixCI
[a-zA-Z]
case sensitive
regular expression
Step 2 filter will read [0-9]

All	CollectionCI	Specimen .CI	SuffixCI
4.	NPL	0000000976	G
5.	NPL	00000001153	G
6.	NPL	0000000172	G
7.	NPL	0000000381	G
8.	NPL	0000000388	G
9.	NPL	0000000388	G
10.	NPL	0000000388	G
138.	UT0	0000000384	A
181.	UT0	0000000386	B
195.	UT0	0000000386	B
277.	UT0	0000000516	B

1) Select the 'regular expression' box. We are telling the text filter to show records that have a-z in them, capitol or lower case letters.

1a) Transform on the filtered column:

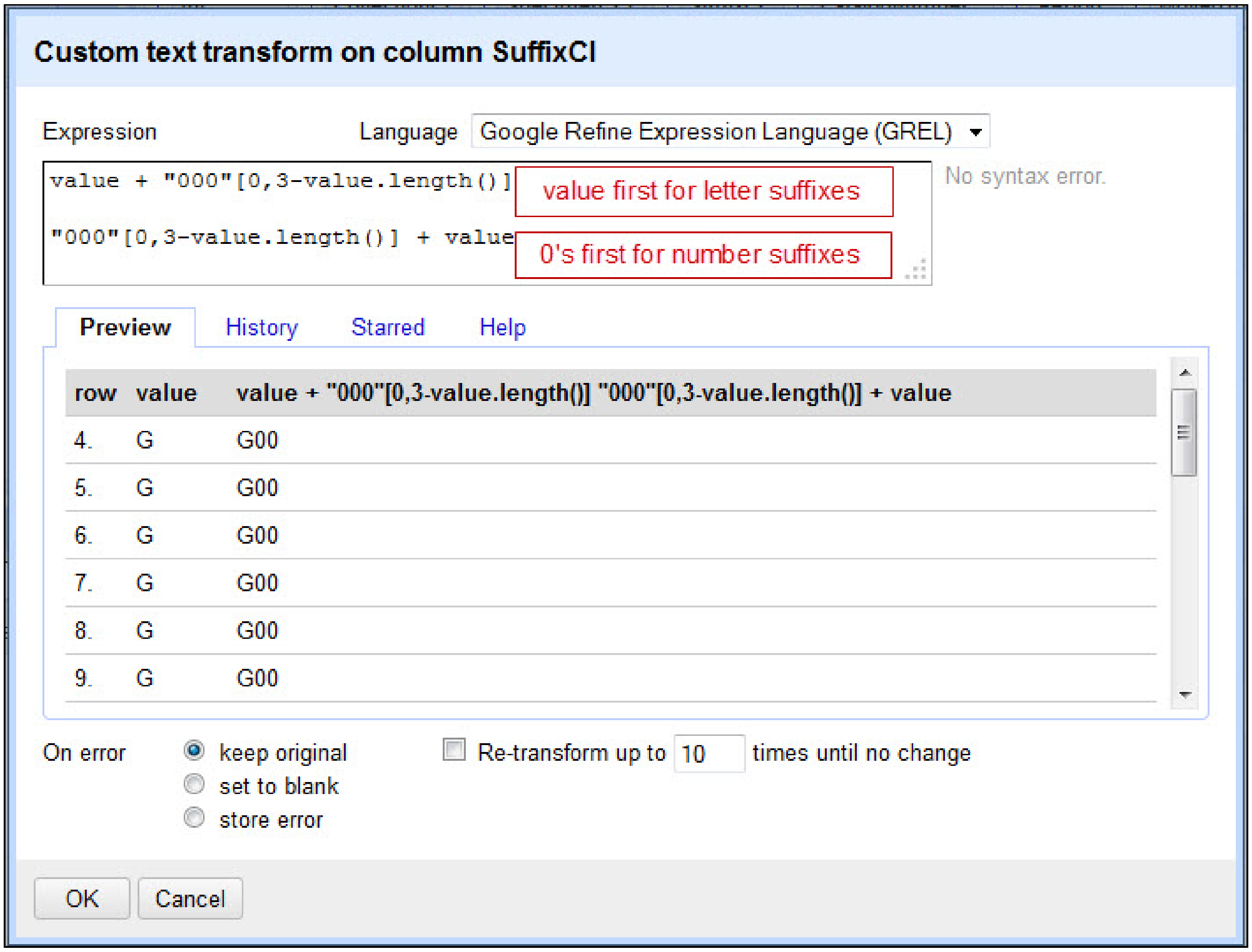
```
value +"000"[0,3-value.length()]
```

Remember, for letters we want A00 B00 and so on.

2) Now change the a-zA-Z to 0-9, leaving the brackets in place.

2a) Transform on the filtered column:

```
"000"+ value[0,3-value.length()]
```



Custom text transform on column SuffixCI

Expression   Language: Google Refine Expression Language (GREL)

```
value + "000"[0,3-value.length()]

"000"[0,3-value.length()] + value
```

value first for letter suffixes
0's first for number suffixes
No syntax error.
Preview
History
Starred
Help

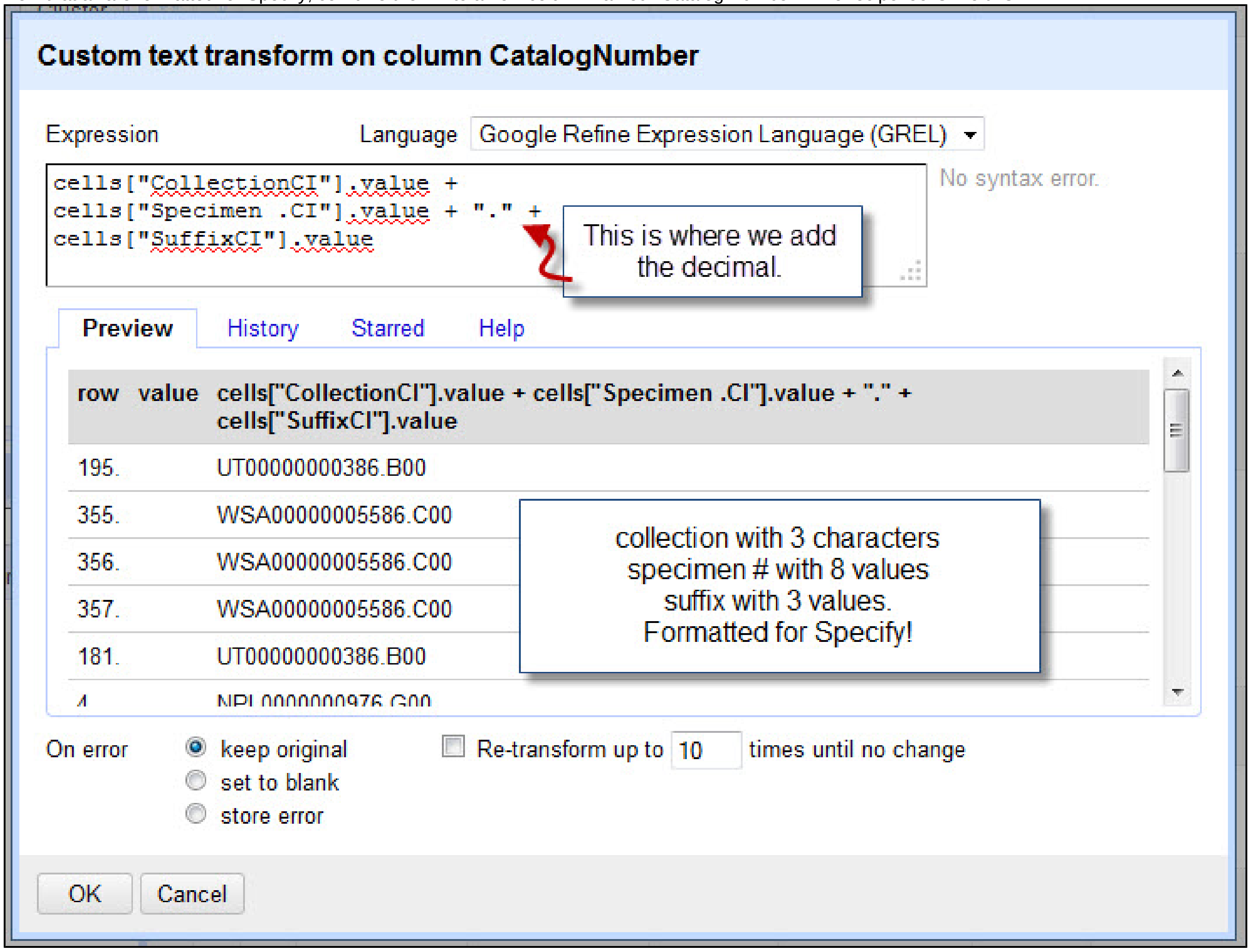
row	value	value + "000"[0,3-value.length()] "000"[0,3-value.length()] + value
4.	G	G00
5.	G	G00
6.	G	G00
7.	G	G00
8.	G	G00
9.	G	G00

On error: [✓ keep original] set to blank store error

☐ Re-transform up to 10 times until no change

You now have all suffixes transformed and properly formatted.

Now that all are formatted for Specify, combine them into a new column named "Catalog Number". The recipe looks like this:



Custom text transform on column CatalogNumber

Expression

Language: Google Refine Expression Language (GREL)

```
cells["CollectionCI"].value +
cells["Specimen .CI"].value + "." +
cells["SuffixCI"].value
```

No syntax error.

This is where we add the decimal.

row	value	cells["CollectionCI"].value + cells["Specimen .CI"].value + "." + cells["SuffixCI"].value
195.		UT00000000386.B00
355.		WSA00000005586.C00
356.		WSA00000005586.C00
357.		WSA00000005586.C00
181.		UT00000000386.B00
4		NPL0000000976.G00

collection with 3 characters
specimen # with 8 values
suffix with 3 values.
Formatted for Specify!

On error: [✓ keep original] set to blank store error

Re-transform up to 10 times until no change

These recipes can be adapted, combined and broken apart to do a variety of actions. Refine is exceptional in that it allows you to undo as much as you need- straight back to the start, if you want.