

Google openrefine

Google refine

A powerful tool dealing with messy data

A GUIDE TO transform and deal with data



TABLE OF CONTENTS

1. INTRODUCTION
2. Data transformation

Chapter 1

INTRODUCTION

[OpenRefine](#) (formerly [Google Refine](#)) is a powerful tool for working with messy data: cleaning it; transforming it from one format into another; extending it with web services; and linking it to databases like Freebase.

Even journalists with little database expertise should be using Refine to organize and analyze data; it doesn't require much more technical skill than clicking through a webpage.

So normally, before attempting any calculations or analysis in Excel, Google Fusion Tables or visualization in Tableau, we often use [Google Refine](#) to clean the data.

Chapter 1

INTRODUCTION

What is “Messy data”?

“Messy data” refers to data that’s riddled with inconsistencies, either because of human error or poorly designed record systems. So, a column that contains a city’s name may hold values such as “New York”, “new york,” “New york city” and etc.

These inconsistencies can wreak havoc when trying to perform analysis on the data, so they have to be addressed before starting any analysis.

Hence, Google Refine, will help us do data cleaning before analyzing it.

Chapter 1

INTRODUCTION

Download and Installation

Step 1:

Please go to the website to install **Google Refine** on your computer.

<http://openrefine.org/download.html>

Remarks: If you are using Mac, please click “Mac Kit”; if you are using Windows, please click “Windows Kit”.

OpenRefine Core

Google Refine 2.5 - Stable version

- **Windows kit**, Download, unzip, and double-click on *google-refine.exe*. If you're having issues with the above, try double-clicking on *refine.bat* instead.
- **Mac kit**, Download, open, drag icon into the Applications folder and double click on it. **NOTE:** If you have issues installing Refine on Mac, please refer to [issue 590](#) - OpenRefine 2.5 for mac support java 6 and 7 only
- **Linux kit**, Download, extract, then type *./refine* to start. **NOTE:** OpenRefine 2.5 for linux support java 6 and 7 only

Chapter 1

INTRODUCTION

Importing your data

Step 2:

Open the application on your computer, and it will be open in the browser, click “Create Project” and “choose file” from your computer. After uploading the file, click “Next”.

Download the data from this website:

<https://drive.google.com/file/d/0Bx8m5CZw-gtgcVZzLVBYQ0l5VTQ/view?usp=sharing>

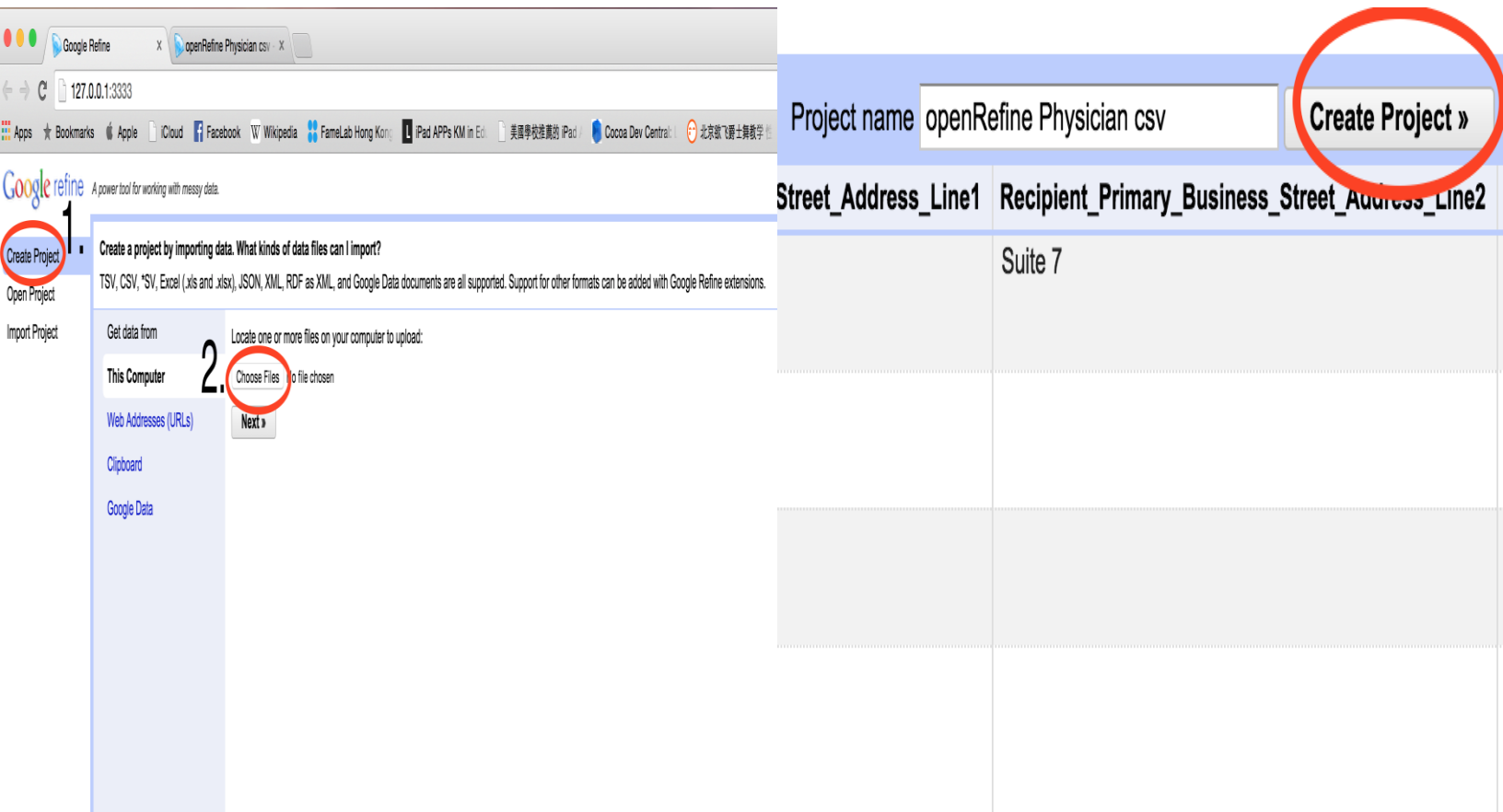
(There is a problem to view it, but no problem to download it)

Chapter 1

INTRODUCTION

Importing your data

See the example below and then click “create project” on the right top corner.



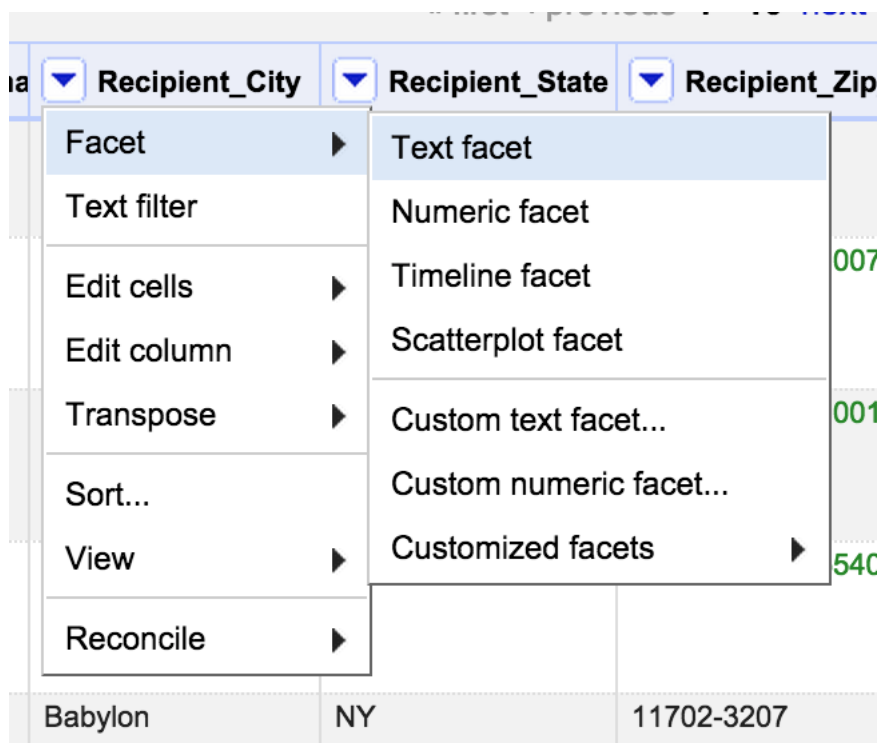
Chapter 1

INTRODUCTION

Faceting data-Text facet

Step 3:

The data includes the information of physicians and recipients and we would like to see the recipient's city. Find the column named "Recipient_City", click the arrow->facet->Text facet.



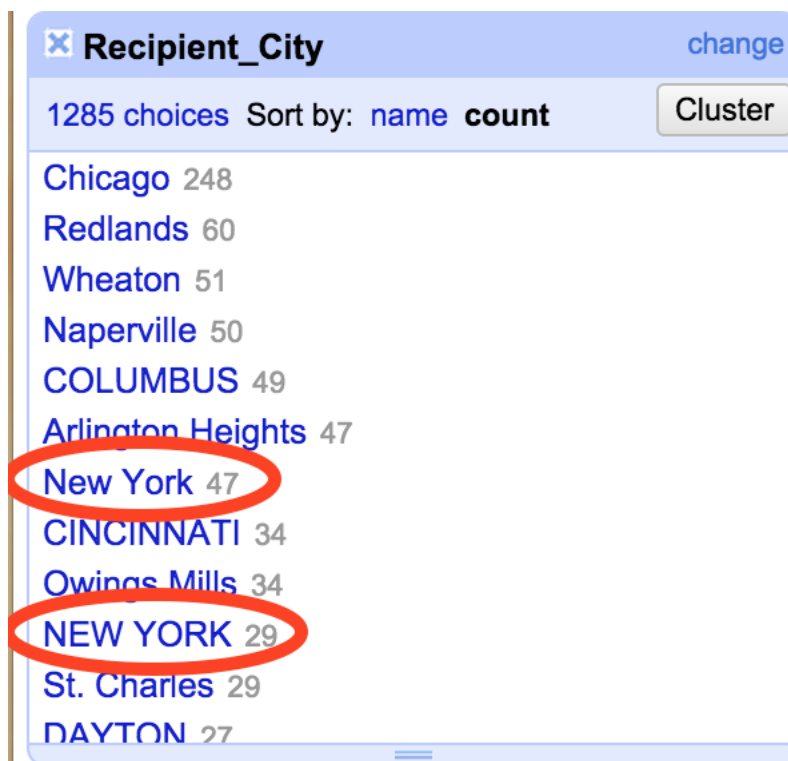
Chapter 1

INTRODUCTION

Faceting data-Text facet

Step 4:

On the left side of your webpage, click “sort by count”, you will see most of the recipients are from Chicago. And also 47 from “**New York**” and 29 from “**NEW YORK**”, which represents the same city.



The screenshot shows a web interface for a data visualization tool. At the top, there is a tab labeled 'Recipient_City' with a 'change' button. Below the tab, it says '1285 choices' and 'Sort by: name count'. There is a 'Cluster' button. The list of cities and their counts is as follows:

City	Count
Chicago	248
Redlands	60
Wheaton	51
Naperville	50
COLUMBUS	49
Arlington Heights	47
New York	47
CINCINNATI	34
Owings Mills	34
NEW YORK	29
St. Charles	29
DAYTON	27

Chapter 1

INTRODUCTION

Clustering data

Step 4:

To solve this problem, we could use “cluster” function on the top right corner.

1. Click “cluster”, you will see the same city name will be clustered together;
2. Click “select all”;
3. Click “merge selected & close”.

Cluster & Edit column "Recipient_City"

This feature helps you find groups of different cell values that might be alternative representations of the same thing. For example, the two strings "New York" and "new york" are very likely to refer to the same concept and just have capitalization differences, and "Gödel" and "Godel" probably refer to the same person. [Find out more ...](#)

Method Keying Function **261 clusters found**

Cluster Size	Row Count	Values in Cluster	Merge?	New Cell Value
3	9	<ul style="list-style-type: none">CORAL GABLES (5 rows)Coral Gables (3 rows)Coral gables (1 rows)	<input type="checkbox"/>	CORAL GABLES
3	251	<ul style="list-style-type: none">Chicago (248 rows)CHICAGO (2 rows)chicago (1 rows)	<input type="checkbox"/>	Chicago
3	4	<ul style="list-style-type: none">St. Louis Park (2 rows)ST LOUIS PARK (1 rows)St Louis Park (1 rows)	<input type="checkbox"/>	St. Louis Park
2	15	<ul style="list-style-type: none">ALBUQUERQUE (13 rows)Albuquerque (2 rows)	<input type="checkbox"/>	ALBUQUERQUE
2	3	<ul style="list-style-type: none">MONROEVILLE (2 rows)Monroeville (1 rows)	<input type="checkbox"/>	MONROEVILLE
2	4	<ul style="list-style-type: none">Mobile (3 rows)MOBILE (1 rows)	<input type="checkbox"/>	Mobile
2	16	<ul style="list-style-type: none">Greensboro (15 rows)GREENSBORO (1 rows)	<input type="checkbox"/>	Greensboro

Choices in Cluster

Rows in Cluster

Average Length of Choices

Length Variance of Choices

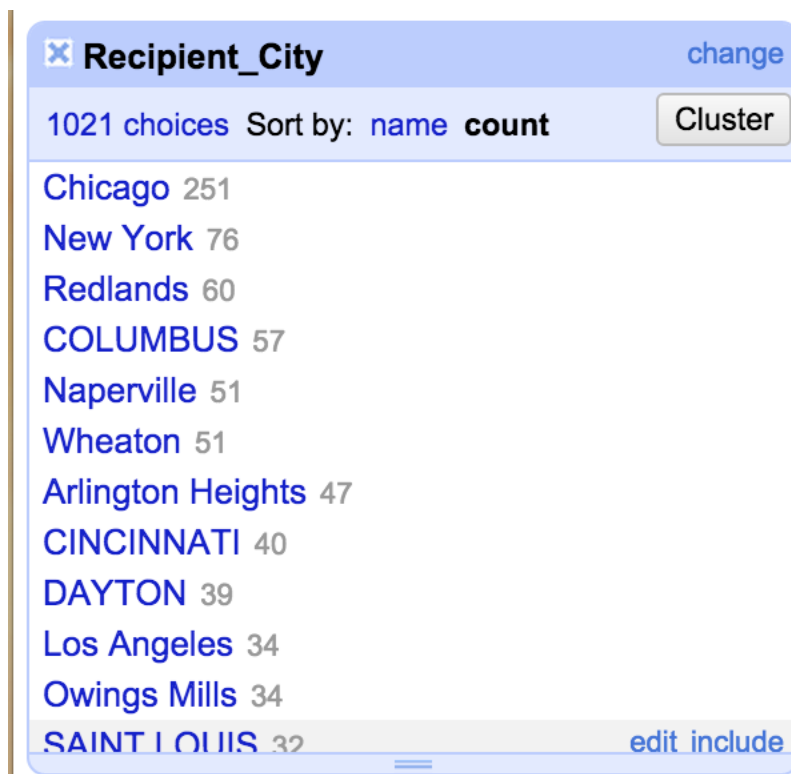
Select All **Deselect All** **Merge Selected & Re-Cluster** **Merge Selected & Close** **Close**

Chapter 1

INTRODUCTION

New Problem Occurs

Then you could see the new lists on the left side. But new **problem** occurs that the names of some cities are in upper case while some of them are in lower case. To overcome this, we use another method.



Recipient_City		change
1021 choices	Sort by: name count	Cluster
Chicago	251	
New York	76	
Redlands	60	
COLUMBUS	57	
Naperville	51	
Wheaton	51	
Arlington Heights	47	
CINCINNATI	40	
DAYTON	39	
Los Angeles	34	
Owings Mills	34	
SAINT LOUIS	32	edit include

Chapter 1

INTRODUCTION

Duplicate the column

Step 5:

Back to the “Recipient_City” column, click the arrow->Edit column->Add column based on this column

10 rows

Prima	Recipient_City	Recipient_State	Recipient_Zip_C	Recipient_
			91766-2007	United States
			10075	United States
				ates
				ates
	Babylon	NY		ates
				ates
	Minneapolis	MI		ates
	Dallas	TX	73390-9057	United States

Facet

Text filter

Edit cells

Edit column

Transpose

Sort...

View

Reconcile

Split into several columns...

Add column based on this column...

Add column by fetching URLs...

Add columns from Freebase ...

Rename this column

Remove this column

Move column to beginning

Move column to end

Move column left

Move column right

Chapter 1

INTRODUCTION

Duplicate the column

Step 5:

1. Enter new column name “city_clean”;
2. Change “value” to “toUppercase(value)”, you will see the value originally and see the new value on the right of it.
3. Then click “OK”.

Add column based on column Recipient_City


New column name

city_clean

On error

☒ set to blank ☐ store error ☐ copy value from original column

Expression

Language Google Refine Expression Language (GREL) 

toUppercase(value)

No syntax error.

Preview

History

Starred

Help

row	value	toUppercase(value)
1.	Pomona	POMONA
2.	New York	NEW YORK
3.	New York	NEW YORK
4.	Minneapolis	MINNEAPOLIS
5.	Babylon	BABYLON
6.	Minneapolis	MINNEAPOLIS
7.	Dallas	DALLAS

OK Cancel

Chapter 1

INTRODUCTION

Duplicate the column

Then you will get a column of city names in uppercase. And you could delete the original column “Recipient_City”.

One thing good about the refine is you could “Undo” the operation you’ve done to the data if you’ve do something wrong.

Create new column city_clean based on column Recipient_City by filling 4785 rows with `grel:toUppercase(value)` [Undo](#)

0 25 50 rows				
Recipient_Primary	Recipient_City	city_clean	Recipient_State	Recipient_Zip
7	Pomona	POMONA	CA	91766-20

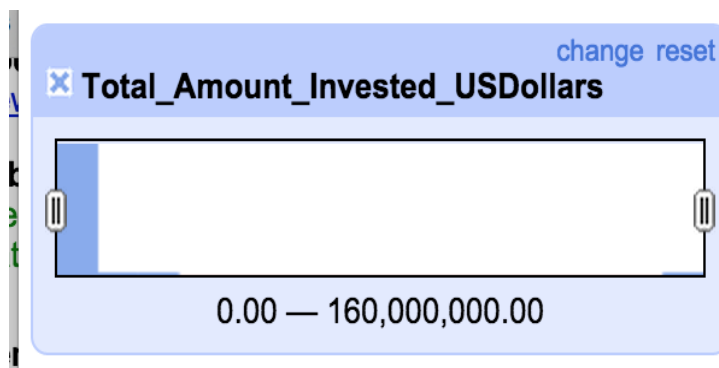
Chapter 1

INTRODUCTION

Facet data-Numeric facet

Step 6:

Find “Total_Amount_Invested_USDollars”, click the arrow->Facet->Numeric facet. Then you could see the result on the left side of the webpage like below.



Chapter 1

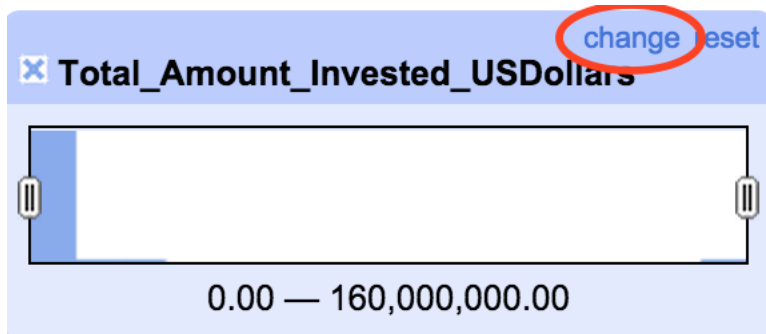
INTRODUCTION

Facet data-Numeric facet

Step 6:

As the range is so large, we need to modify it for better understanding.

1. Click “change” on the top right corner;
2. Change “value” to “value.log()”, you will see the result on the right.
3. Click “OK”.



Expression: `value.log()` (circled in red). Language: Google Refine Expression Language (GREL). No syntax error.

Preview table:

row	value	value.log()
1.	25000	4.3979400086720375
2.	1548471	6.18990307623597
3.	1548471	6.18990307623597
4.	240964	5.3819521638645185
5.	5260	3.7209857441537393
6.	36818	4.56606019338704
7.	2240000	5.5940507071004005

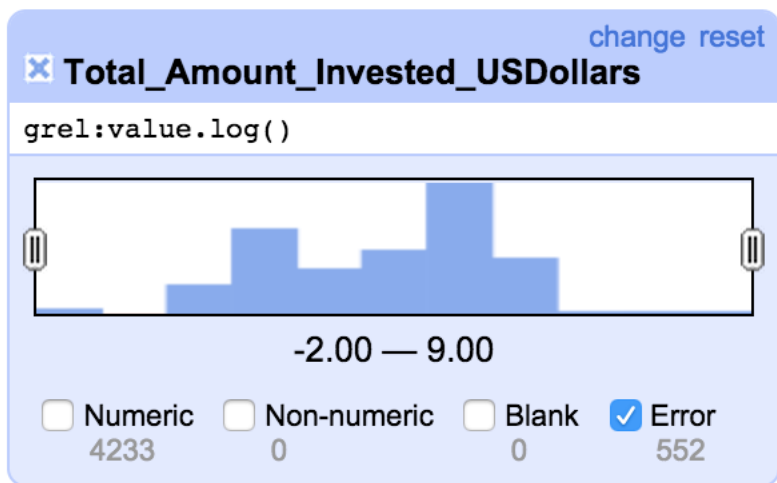
OK (circled in red) Cancel

Chapter 1

INTRODUCTION

Facet data-Numeric facet

Then you will see the result like below: the range will change to from -2 to 9. We could see that there are errors, if we uncheck the numeric, we will see like below. The values are 0, which are correct.



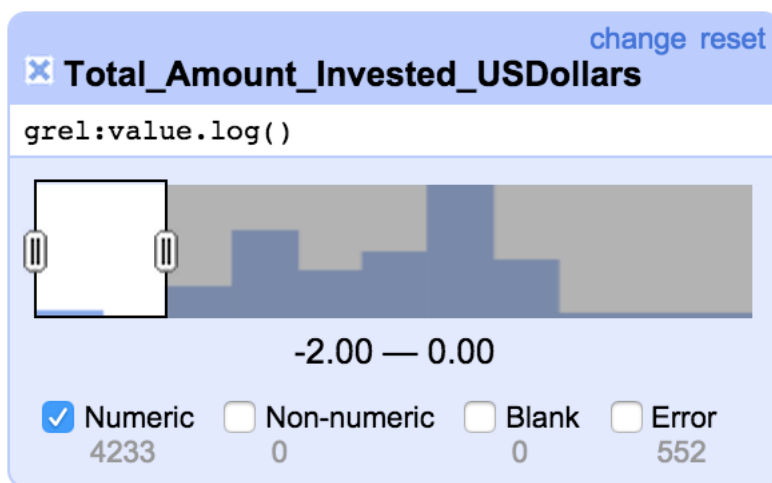
Total_Amount_In
0
0
0
0
edit
0
0
0
0
0
0

Chapter 1

INTRODUCTION

Facet data-Numeric facet

And the range includes negative numbers. If we change the range from -2 to 0, and uncheck the error, we will see the value are 0.1, which is less than 1, we know that $\log(0 \sim 1)$ is negative, and they're correct, too.



Total_Amount_In
0.01
0.01
0.01
0.01
edit 0.01
0.01
0.01
0.01

Chapter 2

Data Transformation

Download original webpage data

Besides the messy data, the data might just not be in the form that you want. And using Google Refine, you can quickly transform it into the form that you do want.

Go to the webpage to get the data!

https://en.wikipedia.org/wiki/Filmfare_Award_for_Best_Actress#List_of_winners_and_nominees

- 1.You need to scroll down to find “List of winners and nominees”;
- 2.Click the “edit” button beside it;
- 3.Copy the codes and paste them in any text editor in your computer;
- 4.Save the file and name it “List of winners”.

Chapter 2

Data Transformation

Import the data

Import the data like the chapter 1 said. You may see the data like this after uploading it. Before cleaning it, we need to do something. If your data does not start with the “===1950s===”, you need to ignore the first “a number” rows to ignore them. Uncheck “store blank rows” and “store blank cells as nulls”.

Column 1
1. ===1950s===\
2. * '''1953 [[Meena Kumari]] \'96 '''[[Baiju Bawra (film) Baiju Bawra]]''' as '''Gauri'''\
3. \
4. * '''1954 [[Meena Kumari]] \'96 '''[[Parineeta (1953 film) Parineeta]]''' as '''Lalita'''\
5. \
6. * '''1955 [[Kamini Kaushal]] \'96 '''[[Biraj Bahu]]''' as '''Biraj Chakravorty'''\
7. ** [[Geeta Bali]] \'96 '''[[Vachan]]''' as Kamla\
8. ** [[Meena Kumari]] \'96 '''[[Azaad (1955 film) Azaad]]''' as Shobha\
9. \
10. * '''1956 [[Nutan]] \'96 '''[[Seema (1955 film) Seema]]''' as '''Gauri'''\
11. \

Parse data as

Fixed-width field text files

Line-based text files

CSV / TSV / separator-based files

PC-Axis text files

JSON files

RDF/N3 files

XML files

Open Document Format spreadsheets (.ods)

RDF/XML files

Character encoding

Column widths:

Column names:

☒ Ignore first

8

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

☐ Store blank rows

☐ Store blank cells as nulls

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

Update Preview

comma separated numbers

optional, comma separated

Chapter 2

Data Transformation

Delete unnecessary rows

1. Choose Column 1, then click “Text filter”;
2. Type “===” to find all rows with “===”, you will see 7 matching rows;
3. Choose “All”, click “Edit rows” then “Remove all matching rows” to delete them aall.

The screenshot illustrates the steps to delete unnecessary rows in a data table. On the left, a table with 10 rows is shown. A context menu for 'Column 1' is open, with 'Text filter' selected. On the right, the table is filtered to show '7 matching rows (330 total)' based on the filter 's===\' applied to 'Column 1'. A secondary menu is open over the 'Edit rows' option, showing 'Remove all matching rows' as the final action.

All	Column 1
1.	
2.	\96
3.	
4.	\96
5.	
6.]] \96
7.	chan]
8.	[[Aza
9.	
10.	eema

7 matching rows (330 total)

Show as: **rows** records Show: 5 10 25 50 rows

All	Column 1
	s===\'

Star rows
Unstar rows
Flag rows
Unflag rows
Remove all matching rows

Chapter 2

Data Transformation

Transform Function

1. Choose “Column 1”, click “Edit cells” and then “Transform”;
2. Modify the Expression like below, you will see the different appearance of the data from below, then click “OK”.

Custom text transform on column Column 1

Expression

Language Google Refine Expression Language (GREL)

```
value.replace("''", "")
```

No syntax error.

Preview

History

Starred

Help

row	value	value.replace("''", "")
1.	* "'1953 [[Meena Kumari]] \96 "[[Baiju Bawra (film) Baiju Bawra]]'" as "'Gauri'"	* 1953 [[Meena Kumari]] \96 "[[Baiju Bawra (film) Baiju Bawra]]" as Gauri
2.	* "'1954 [[Meena Kumari]] \96 "[[Parineeta (1953 film) Parineeta]]'" as "'Lalita'"	* 1954 [[Meena Kumari]] \96 "[[Parineeta (1953 film) Parineeta]]" as Lalita
3.	* "'1955 [[Kamini Kaushal]] \96 "[[Biraj Bahu]]'" as "'Biraj Chakravorty'"	* 1955 [[Kamini Kaushal]] \96 "[[Biraj Bahu]]" as Biraj Chakravorty
4.	** [[Geeta Bali]] \96 "[[Vachan]]" as Kamla	** [[Geeta Bali]] \96 "[[Vachan]]" as Kamla
5.	** [[Meena Kumari]] \96 "[[Azaad (1955	** [[Meena Kumari]] \96 "[[Azaad (1955

On error

- ☒ keep original
☐ set to blank
☐ store error

☐ Re-transform up to times until no change

OK

Cancel

Chapter 2

Data Transformation

Add Column to see the winner

1. Do you still remember how to add column based on this column we described in chapter 1? You need to click “Edit column” then choose “add column based on this column”.
2. Modify like the screenshot below, pay attention to the name and the expression.

Add column based on column Column 1

New column name

On error



set to blank



store error



copy value from original column

Expression

Language

Google Refine Expression Language (GREL)



```
not(value.startsWith("**"))
```

No syntax error.

Preview

History

Starred

Help

row	value	not(value.startsWith("**"))
1.	* 1953 [[Meena Kumari]] \96 "[[Baiju Bawra (film) Baiju Bawra]]" as Gauri\	true
2.	* 1954 [[Meena Kumari]] \96 "[[Parineeta (1953 film) Parineeta]]" as Lalita\	true
3.	* 1955 [[Kamini Kaushal]] \96 "[[Biraj Bahu]]" as Biraj Chakravorty\	true
4.	** [[Geeta Bali]] \96 "[[Vachan]]" as Kamla\	false
5.	** [[Meena Kumari]] \96 "[[Azaad (1955 film) Azaad]]" as	false

OK

Cancel

Chapter 2

Data Transformation

Add the “Year” Column

1. Then you will see a new column with values “true” or “false” occurs, facet it with text facet and click “true” to isolate the winners.
2. Click “add column based on this column” again and modify like below.

Add column based on column Column 1

New column name

On error



set to blank



store error



copy value from original column

Expression

Language

Google Refine Expression Language (GREL)



value[1,6]

No syntax error.

Preview

History

Starred

Help

row value

value[1,6]

1.	* 1953 [[Meena Kumari]] \96 "[[Baiju Bawra (film) Baiju Bawra]]" as Gauri\	1953
2.	* 1954 [[Meena Kumari]] \96 "[[Parineeta (1953 film) Parineeta]]" as Lalita\	1954
3.	* 1955 [[Kamini Kaushal]] \96 "[[Biraj Bahu]]" as Biraj Chakravorty\	1955
6.	* 1956 [[Nutan]] \96 "[[Seema (1955 film) Seema]]" as Gauri\	1956
7.	* 1957 [[Nargis Dutt]] \96 "[[Mother India]]" as Radha\	1957
8.	* 1958 [[Vyjayanthimala]] \96 "[[Sadhna]]" as Champabai / Rajani\	1958
11.	* 1959 [[Nutan]] \96 "[[Qudrat (1959 film) Qudrat]]" as Qudrat\	1959

OK

Cancel

Chapter 2

Data Transformation

Delete year from the original column

1. Choose “Column 1”, click “Edit cells” then “Transform”;
2. Modify the expression like below. You will see the result.

Expression

Language

Google Refine Expre

```
value.substring(7)
```

▼ All	▼ Column 1	▼ Year	▼ Is Winner
★	1. [[Meena Kumari]] \96 "[[Baiju Bawra (film) Baiju Bawra]]" as Gauri\	1953	true
★	2. [[Meena Kumari]] \96 "[[Parineeta (1953 film) Parineeta]]" as Lalita\	1954	true
★	3. [[Kamini Kaushal]] \96 "[[Biraj Bahu]]" as Biraj Chakravorty\	1955	true
★	6. [[Nutan]] \96 "[[Seema (1955 film) Seema]]" as Gauri\	1956	true
★	7. [[Nargis Dutt]] \96 "[[Mother India]]" as Radha\	1957	true
★	8. [[Vyjayanthimala]] \96 "[[Sadhna]]" as Champabai / Rajani\	1958	true
★	11. [[Nutan]] \96 "[[Sujata (1959 film) Sujata]]" as Sujata\	1959	true
★	14. [[Bina Rai]] \96 "[[Ghunghat (1960 film) Ghunghat]]" as Parvati / Jamuna\	1960	true
★	17. [[Vyjayanthimala]] \96 "[[Gunga Jumna]]" as Dhanno\	1961	true
★	20. [[Meena Kumari]] \96 "[[Sahib Bibi Aur Ghulam]]" as Chhoti Bahu\	1962	true

Chapter 2

Data Transformation

Deal with nominees (not winners)

1. Choose “false” in the “Is Winner” facet;
2. Choose “Column 1”, “Edit cells” then “Transform”;
3. Modify like below and you will see the result.

Expression

Language

Google Refine Expression

value.substring(2)

No s

193 matching rows (256 total)

Show as: rows records

Show: 5 10 25 50 rows

« first < pre

All		Column 1	Year	Is Winner
☆	🔍	4. [[Geeta Bali]] \96 "[[Vachan]]" as Kamla\		false
☆	🔍	5. [[Meena Kumari]] \96 "[[Azaad (1955 film) Azaad]]" as Shobha\		false
☆	🔍	9. [[Meena Kumari]] \96 "[[Sahara (1958 film) Sahara]]" as Leela\		false
☆	🔍	10. [[Vyjayanthimala]] \96 "[[Madhumati]]" as Madhumati / Madhavi / Radha\		false
☆	🔍	12. [[Mala Sinha]] \96 "[[Dhool Ka Phool]]" as Meena Khosla\		false
☆	🔍	13. [[Meena Kumari]] \96 "[[Chirag Kahan Roshni Kahan]]" as Ratna\		false
☆	🔍	15. [[Madhubala]] \96 "[[Mughal-e-Azam]]" as [[Anarkali]]\		false
☆	🔍	16. [[Nutan]] \96 "[[Chhalia]]" as Shanti\		false
☆	🔍	18. [[Padmini]] \96 "[[Jis Desh Men Ganga Behti Hai]]" as Kammo\		false
☆	🔍	19. [[Saira Banu]] \96 "[[Junglee]]" as Rajkumari 'Raj'\		false

Chapter 2

Data Transformation

Deal with nominees (not winners)

1. Move the “Is Winner” facet on the left side, you will see that the nominees’ rows are missing years.
2. To deal with this, we can just choose the “Year” column, click “Edit cells” then “Fill down” because the nominees’ year are the same with the winners’.

256 rows						
Show as: rows records		Show: 5 10 25 50 rows			« first < pre	
▼ All		▼ Column 1			▼ Year	▼ Is Winner
★	🗨	1.	[[Meena Kumari]] \96 "[[Baiju Bawra (film) Baiju Bawra]]" as Gauri\		Facet	▶
★	🗨	2.	[[Meena Kumari]] \96 "[[Parineeta (1953 film) Parineeta]]" as Lalita\		Text filter	
★	🗨	3.	[[Kamini Kaushal]] \96 "[[Biraj Bahu]]" as Biraj Chakraverty\			
★	🗨	4.	[[Geeta Bali]] \96 "[[Vachan]]" as Kamla		Transform...	
★	🗨	5.	[[Meena Kumari]] \96 "[[Azaad (1955 film) Azaad]]" as Azaad		Common transforms	▶
★	🗨	6.	[[Nutan]] \96 "[[Seema (1955 film) Seema]]" as Seema		Fill down	
★	🗨	7.	[[Nargis Dutt]] \96 "[[Mother India]]" as Mother India		Blank down	
★	🗨	8.	[[Vyjayanthimala]] \96 "[[Sadhna]]" as Sadhna		Split multi-valued cells...	
★	🗨	9.	[[Meena Kumari]] \96 "[[Sahara (1958 film) Sahara]]" as Sahara		Join multi-valued cells...	
★	🗨	10.	[[Vyjayanthimala]] \96 "[[Madhumati]]" as Madhumati		Cluster and edit...	

Chapter 2

Data Transformation

Separate columns

In this part, we need to separate the first column into 3 separate columns: actress, film and character.

1. Choose “Column 1”, “Edit column” then “split into several columns”.

2. Type into the separator like below, pay attention to the **space at the beginning and the end**. If the separator are not these, just copy the separator between the actress and the film. Then type in “2” columns, and click “OK”.

Split column Column 1 into several columns

How to Split Column

- ☒ by separator
Separator ☐ regular expression
Split into columns at most (leave blank for no limit)

- ☐ by field lengths

List of integers separated by commas, e.g., 5, 7, 15

After Splitting

- ☒ Guess cell type
☒ Remove this column

OK

Cancel

Chapter 2

Data Transformation

Separate columns

1. Choose the new column you added before;
2. Still choose “Edit column”-> “split into several columns”;
3. For this time, you need to type in “**space-**>**as->space**”, and still 2 columns to separate.

Split column Column 1 2 into several columns

How to Split Column

- ☒ by separator
- Separator ☐ regular expression
- Split into columns at most (leave blank for no limit)

- ☐ by field lengths

List of integers separated by commas, e.g., 5, 7, 15

After Splitting

- ☒ Guess cell type
- ☒ Remove this column

OK

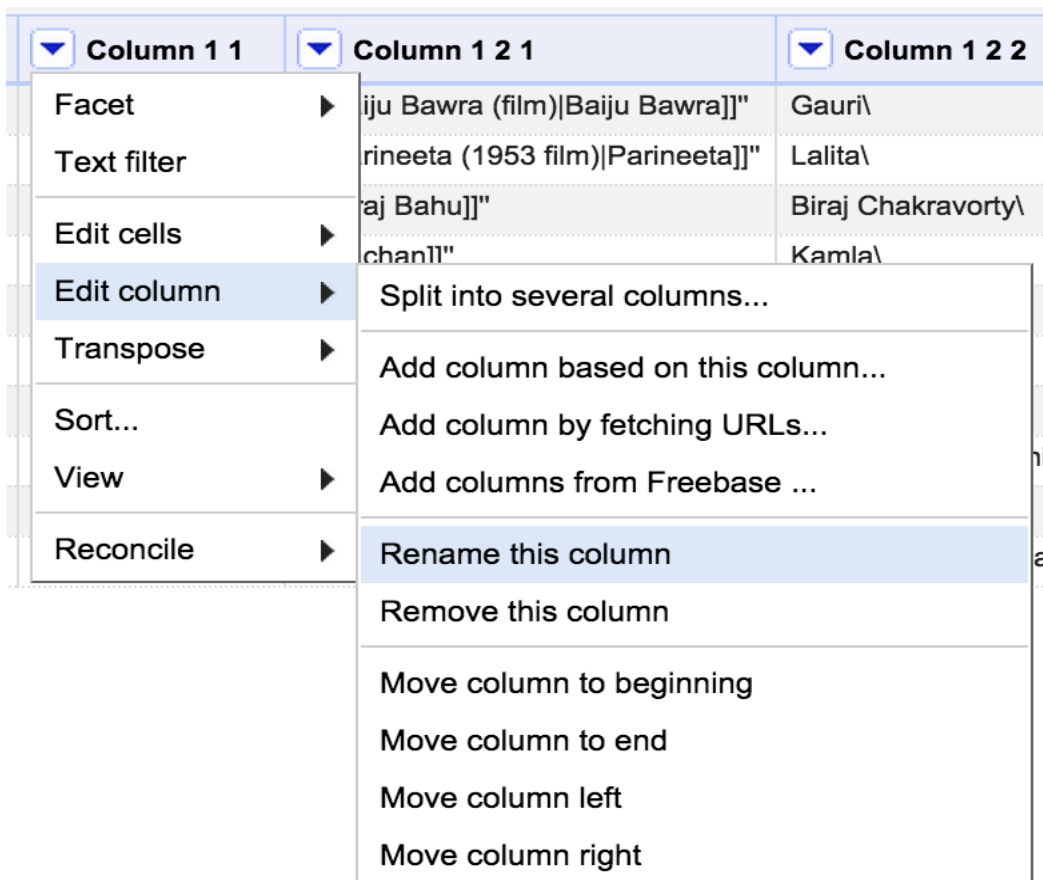
Cancel

Chapter 2

Data Transformation

Rename columns

1. Choose “Edit column”-> “rename this column” to modify each column to “actress”, “film” and “character”.



Chapter 2

Data Transformation

Do something different

We would like to know if the actress is the winner in the actress column. So we do like this:

1. In the actress column, choose “edit cell” and then “transform”;
2. Modify like below, you will see the result. (If there are some errors, just ignore this step due to the data errors).

Custom text transform on column actress

Expression

Language

Google Refine Expression

```
if (cells["Is Winner"].value,value+"(winner)",value)
```

No

Preview

History

Starred

Help

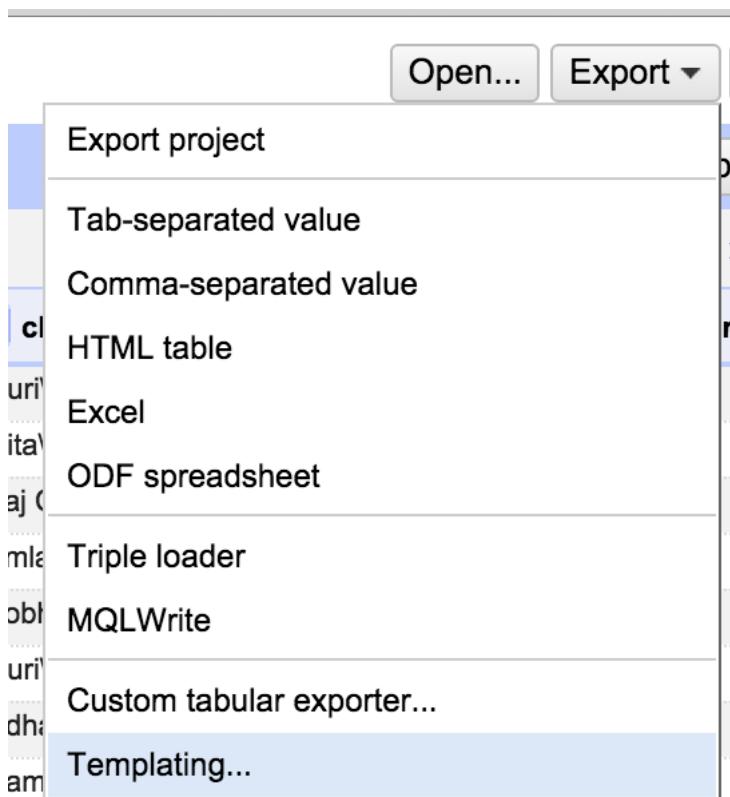
row	value	if (cells["Is Winner"].value,value+"(winner)",value)
1.	[[Meena Kumari]]	[[Meena Kumari]](winner)
2.	[[Meena Kumari]]	[[Meena Kumari]](winner)

Chapter 2

Data Transformation

Export the data

After modifying the data, we need to export it. Click the “export” button then choose “Templating”. Modify the code like below, then click “export”.



Prefix

```
{ |  
|-  
! actress  
! film  
! ...
```

Row Template

```
{{cells["actress"].value}}  
{{cells["film"].value}}  
{{cells["character"].value}}  
{{cells["Year"].value}}
```

Prefix

```
! actress  
! film  
! character  
! year  
|-
```

Row Separator

```
|-
```

Suffix

```
| }
```


Chapter 2

Data Transformation

Export the data

The file will be downloaded automatically, open the file, copy the codes and paste them in the original wiki editor, click “show preview”, you will see a table like this. (The layout may be a little different on your computer due to the different settings).

actress	film	character	year
Meena Kumari	<i>Parineeta</i>	Lalita	1954
Kamini Kaushal	<i>Biraj Bahu</i>	Biraj Chakravorty	1955
Geeta Bali	<i>Vachan</i>	Kamla	1955
Meena Kumari	<i>Azaad</i>	Shobha	1955
Nutan	<i>Seema</i>	Gauri	1956
Nargis Dutt	<i>Mother India</i>	Radha	1957
Vyjayanthimala	<i>Sadhna</i>	Champakabai / Rajani	1958
Meena Kumari	<i>Sahara</i>	Leela	1958
Vyjayanthimala	<i>Madhumati</i>	Madhumati / Madhavi / Radha	1958
Nutan	<i>Sujata</i>	Sujata	1959
Mala Sinha	<i>Dhool Ka Phool</i>	Meena Khosla	1959
Meena Kumari	<i>Chirag Kahan Roshni Kahan</i>	Ratna	1959
Bina Rai	<i>Ghunghat</i>	Parvati / Jamuna	1960
Madhubala	<i>Mughal-e-Azam</i>	Anarkali	1960
Nutan	<i>Chhalia</i>	Shanti	1960
Vyjayanthimala	<i>Gunga Jumna</i>	Dhanno	1961