A Survey of Data Cleaning Tools

Group 1

Lukas Bodner, Daniel Geiger, and Lorenz Leitner

706.057 Information Visualisation SS 2020
Graz University of Technology

20 May 2020

Abstract

Clean data is important to achieve correct results and prevent wrong conclusions. However, many data sources contain "unclean" data, such as improperly formatted cells, inconsistent spellings, wrong character encodings, unnecessary entries, and so on. Ideally, all these and more should be cleaned or removed before starting any analysis and work with the data. Doing this manually is tedious and often impossible, depending on the size of the data set. Therefore, data cleaning tools exist, which aim to simplify and automate these tasks, to arrive at robust and concise clean data quickly.

This survey looks at different solutions for data cleaning, in specific multiple data cleaning tools. They are described, tested and evaluated, using the same unclean data sets to provide consistency. A feature matrix is created which shows an overview of all data cleaning tools and their characteristics, which can be used for a quick comparison. Some of the tools are described and tested in more detail. A final general recommendation is also provided.

Contents

List of Figures

List of Tables

List of Listings

Chapter 1

Introduction

Nowadays, the amount of available data is tremendous. Data analysts face the challenge of getting the most out of the huge amount of data. However, the size of available data is not the only problem. Often data sets include "unclean" data, such as leading or trailing whitespace, duplicate entries, non-standardized user input and many other things.

Typical data cleaning tasks include the following operations:

- Merging similar data sets together
- Splitting a big data set into several data sets
- Removing wrong, missing or incorrect data
- Data deduplication

The goal of data cleaning tools is to increase the quality of data sets. Whenever low-quality data sets are used for an analysis, there exists the risk of getting incorrect results. Based on these incorrect results analysts will then derive improper conclusions. Decisions based on these improper conclusions could have major impacts. For instance, project managers will make wrong decisions which will be costly for the business. Politicians use wrong results for new actions to be taken which in the worst case could injure people. Therefore, ensuring high-quality data has highest priority before doing any analysis.

The main contribution of this survey paper are the evaluation results of several data cleaning tools. For each of the tools different example tasks were performed, for instance, merging of similar data sets. Moreover, all tools include further information regarding platform availability, history of the tool, highlighted features and many more. Finally, the report also includes a feature matrix. In total, it has a dimension of 25×12, including 25 different tools described by 12 characteristics. The feature matrix can be used as a starting point on how to decide whether a data cleaning tool fits for a use case or not.

The content of the survey paper is structured as follows. Chapter 2 describes the data sets on which the data cleaning tasks were performed. The content of the feature matrix is described in Chapter 3. Chapter 4 contains the results of the data cleaning tool evaluation. Finally, Chapter 5 states final remarks and provides recommendations for data cleaning tools.

Chapter 2

Data Sets

Data sets are files that contain data, most often in a text format such as comma-separated values (CSV), JavaScript Object Notation (JSON), Spreadsheet formats for programs such as Microsoft Excel or LibreOffice Calc, or databases such as MySQL. These data sets are the input for all data cleaning tools. Most commonly, they can be imported as a first step in a data cleaning tool, which converts them to some internal representation, which allows the cleaning and transformation to take place. At the end of the cleaning process, they can be exported again, if wanted to the same file format that was used in when importing the data set, but many tools also offer to export different file formats.

Initially, only local data sets were to be used, but it was discovered that, to find "real" "unclean" data, venturing outside of the local city and even Austria was necessary.

In any case, three main data sets were used to test and evaluate the data cleaning tools, and the same tasks/examples were performed on these data sets, as to evaluate the data cleaning tools based on the same criteria as consistently as possible. The following describes these three data sets, the data they contain, the issues they pose, and what kind of tasks are to be performed on them.

2.1 Data Set 1: Parking Spots in Graz (Task: Merging)

The first data source actually consists of two data sets [Stadt Graz 2014b; Stadt Graz 2014a]. They both provide information about locations of different parking spots in Graz, Austria. One is about parking garages, the other about Park & Ride parking lots, which are located near public transportation hubs. These two data sets are not unclean in themselves, but since they are so similar, as they are contain the same column headers, they can be merged into one data set. This task, merging, can be commonly achieved by a union operation in data cleaning tools. Figure 2.1 partially shows one of these two similar data sets.

2.2 Data Set 2: Candy Ratings (Task: Standardization)

This data set is about candies and their ratings, as gathered from a user survey [Ng 2017]. Most information was entered manually by the users themselves, which can be seen by the column `Q4: COUNTRY`. Apparently, there was no drop-down menu in this form, but a raw text field. Hence the country names are rather disparate. Ideally, every user that lives in the same country should have exactly the same value in the country field, which makes filtering or sorting according to that column trivial. As it is, however, these tasks cannot be done. Therefore the country names need to be standardized. For instance, entries like "U.S.", "USA", and "United States of America" should all be transformed into the same notation, like "US". This task is commonly known as standardization, or clustering. Some data cleaning tools offer this out of the box, other do not. A screenshot of the start of this candy ratings data set can be seen in Figure 2.2.

XCoord	YCoord	OBJECTID	NAME	ANSCHRIFT	ORT	KAT3	HERKUNF	PHI	LAMBDA
1721958	51930000	5955361	5020000	1	(PH) LKH	Stiftingtalstraße 30	Graz	Parkhaus	Stadt Gra
1717906	44080000	5952791	90220000	2	(PP) Griesplatz	Griesplatz 7	Graz	Parkplatz	Stadt Gra
1717746	5070000	5953801	92210000	3	(PH) Orpheum	St. Georgen Gasse 1	Graz	Parkhaus	Stadt Gra
1718061	83990000	5953499	91630000	4	(PH) Griesgass	Griesgasse 10	Graz	Parkhaus	Stadt Gra
1716533	99390000	5955946	77540000	5	(PH) Austeinga	Austeingasse 30	Graz	Parkhaus	Stadt Gra
1717725	16970000	5955627	46840000	6	(PH) Körösistra	Körösistraße 67	Graz	Parkhaus	Stadt Gra
1717584	80040000	5953012	93610000	7	(PH) Rösselmü	Rösselmühlgasse 12	Graz	Parkhaus	Stadt Gra
1717482	48040000	5952865	12710000	8	(PH) Am Rösse	Dreihackengasse 42	Graz	Parkhaus	Stadt Gra
1722123	11130000	5945125	3690000	9	(PH) Thondorf	Liebenauer Hauptstr	Graz	Parkhaus	Stadt Gra
1716117	81890000	5953433	77250000	10	(PH) GKB Cent	Köflacher Gasse 3	Graz	Parkhaus	Stadt Gra
1718238	70940000	5954162	7220000	11	(PH) Schloßbe	Sackstraße 29	Graz	Parkhaus	Stadt Gra
1718917	68200000	5952829	96350000	12	(PH) Schönaug	Schönaugasse 6	Graz	Parkhaus	Stadt Gra
1719520	4690000	5953099	89080000	13	(PH) Kaiser-Jos	Schlögelgasse 5	Graz	Parkhaus	Stadt Gra
1721605	64890000	5952330	88170000	14	(PP) Plüddema	Plüddemanngasse 7	Graz	Parkplatz	Stadt Gra

Figure 2.1: One of the two parking spot data sets, in particular, the one about parking garages, which can be ascertained from the `PHI` column, which lists parking "houses". [Screenshot taken by the authors of this survey.]

Internal ID	Q1: GOIN	Q2: GEND	Q3: AGE	Q4: COUNTRY	Q5: STATE	Q6 | 100
90258773						
90272821	No	Male	44	USA	NM	MEH
90272829		Male	49	USA	Virginia	
90272840	No	Male	40	us	or	MEH
90272841	No	Male	23	usa	exton pa	JOY
90272852	No	Male				JOY
90272853	No	Male	53	usa	Colorado	
90272854	No	Male	33	canada	ontario	JOY
90272858	No	Male	40	Canada	Ontario	JOY
90272859	No	Female	53	Us	Wa	MEH
90272861	Yes	Male	43			
90272865	No	Male	56	Canada	Quebec	JOY
90272866	No	Male	64	US	NY	MEH
90272867	Yes	Male	43	Murica	California	JOY
90272868	No	Female	37	Canada	Ontario	MEH
90272878	No	Male	64	USA	Texas	JOY
90272880	No	I'd rather	59	USA	NEW YOR	JOY
90272881	No	Male	48	US	CO	MEH
90272883	No	Female	54	United States	IN	

Figure 2.2: The data set about ratings of candy. Interesting in particular is the `Q4: COUNTRY` column, which contains these different spellings of the same country, easily seen here with the example of the United States. [Screenshot taken by the authors of this survey.]

Table: Green area per capita

Variable	Green area per capita (square meters per capita)						
Year	2000	2001	2002	2003	2004	2005	2006
Metropolitan areas							
Australia	..	..	..	..	..	..	..
Sydney	224.95	224.94	224.98	224.95	224.97	224.96	224.98
Melbourne	152.19	152.19	152.18	152.21	152.20	152.18	152.20
Brisbane	1158.08	1158.09	1158.08	1158.08	1158.08	1158.08	1158.09
Perth	78.13	78.14	78.14	78.14	78.14	78.14	78.14
Adelaide	37.14	37.14	37.14	37.15	37.14	37.15	37.14
Gold Coast-Tweed Heads	108.62	108.62	108.62	108.62	108.62	108.62	108.62
Austria	..	..	..	..	..	..	..
Vienna	620.15	620.14	620.15	620.16	620.16	620.14	620.15
Graz	551.67	551.67	551.67	551.67	551.67	551.67	551.67
Linz	1043.67	1043.67	1043.67	1043.67	1043.67	1043.67	1043.67
Belgium	..	..	..	..	..	..	..
Brussels	738.42	738.43	738.43	738.43	738.42	738.42	738.42
Antwerp	331.43	331.43	331.43	331.43	331.43	331.44	331.43

Figure 2.3: The data set about green areas in cities. The desired rows about Austria can be seen, as well as the irrelevant rows on top ("Table: ...", "Variable ...", "Metropolitan areas"), and the leftmost column, that contains countries and cities. [Screenshot taken by the authors of this survey.]

2.3 Data Set 3: Green Areas in Cities (Task: Filtering)

The final main data set that was used concerns the "green" area in different cities around the world [Organization for Economic Cooperation and Development 2018]. The final task for this data set is filtering the cities to just include cities from Austria, and disregard all locations outside of Austria. However, this data set has a few issues right off the bat, which makes this task of filtering impossible without some pre-processing steps. In specific, a few rows at the top of the data set need to be removed, as they contain no valuable information. Also, the leftmost column contains countries *and* cities, below each other, when ideally, there should be a column for countries only, and next to that, a column for the cities, which would make filtering for a country and retrieving only cities from the country trivial. Therefore, before the filtering can start, these pre-processing steps need to be done. Figure 2.3 shows an excerpt from this data set.

Chapter 3

Feature Matrix

The idea of the feature matrix is to give users a quick overview about the characteristics of certain data cleaning tools. Each column represents a data cleaning tool and each row describes one of the characteristics.

The following characteristics were chosen to be included in the feature matrix:

1. Website: Find out more about the tool
2. Local/Web: Local installation or access via web service
3. Paid/Free: Information regarding the pricing model
4. License: License under which the tool is available
5. Platform/OS: Supported platforms, for instance, MacOS or Unix
6. Data Privacy: Does the processed data leave the local machine?
7. Input Formats: Which input formats are supported?
8. Encoding Selection: Can the file's character encoding be changed during the import stage?
9. Output Formats: Supported output formats if different to input formats
10. User-friendliness: Is the tool intuitive to use?
11. Documentation: Does the tool provide documents, tutorials or videos?
12. Support: What are possible sources of support?

	OpenRefine	DataCleaner	Trifacta	Alteryx Designer
Website	https://openrefine.org/	https://datacleaner.org/	https://trifacta.com/	https://alteryx.com/
Local/Web	Local	Local	Free: Secure cloud application, Pro: Hosted cloud deployment on AWS	Local
Paid/Free	Free	Community Version is Free	14 day trial / Paid or limited free version	Paid, trial, student/education version
License	BSD 3-Clause	Community Version: LGPL-3.0	https://docs.trifacta.com/display/SS/Legal	Website Legal
Platform/OS	Cross-platform	Cross-platform	Cross-platform	Microsoft Windows
Data Privacy	keeps your data private on your own computer until YOU want to share or collaborate	We assume private since it is a standalone software solution	Free/Pro: Claim Security via Data locality and SSL, Enterprise: SSL, SSO, Kerberos, Apache Sentry	Alteryx tracks only limited technical and usage data.
Input Formats	CSV, Line-based txt, Excel, ODS, XML, JSON,...	MySQL, Oracle, CSV, Excel, XML, MongoDB, ...	Free: CSV, JSON, TXT, Pro: Free + AWS, DBs ,Enterprise: Pro + more	TXT, CSV, JSON, Relational DBs and a lot more
Encoding Selection	Allows selection of well-known encodings at importstage	Allows selection of well-known encodings at importstage (UTF, ASCII, ISO, Win, IBM)	Can select encoding for importing	Allows selection of well-known encodings at importstage
Output Formats	SQL exporter, Templating, Upload to Wikidata	"DataCleaner staging files" + conventional formats	Outputs as CSV or JSON	
User-friendliness	Easy to use and self-explaining. Allows to export/import history of applied operations	Many errors, crashes and unintuitive usage.	Very intuitive to use, easy to revert. Help is not so helpful	Intuitive creation of workflows. However, complexity of operations could be overwhelming for beginners.
Documentation	+++	~ Documentation is often only the function prototype with no or very little explanation()	++ Maybe more documentation after login	+++ Community platform includes lots of material
Support	FAQ, External Tutorials, Mailing List, Chat on Gitter	Issues do not get resolved or answered to frequently	Free: Community support. Pro: Enterprise support by Trifacta	Community based support (Learn, ask, browse, read, listen)
Notes				

Table 3.1: Features of OpenRefine, DataCleaner, Trifacta and Alteryx Designer

	Gridoc	Easy Data Transform	DataWrangler	Tableau
Website	https://gridoc.com/	https://easydatatransform.com/	https://vis.stanford.edu/wrangler/	https://tableau.com/
Local/Web	Web	Local	Web	Tableau Desktop: local, Tableau Online: hosted by Tableau, Tableau Server: run with cloud-service or on-premise
Paid/Free	Paid, limited free plan	Paid, trial version	Free	Paid, free trial
License	https://www.alteryx.com/sites/default/files/2019-02/			
Platform/OS				
Data Privacy	Possibility to have private Gridoc instance deployed into AWS cloud. Hence, privacy also depending on AWS cloud.	Your data never leaves your computer, unless you want it to.		Tableau Desktop stores your data locally. One can share data using Tableau Online/Server, privacy depends on host.
Input Formats	Free Plan: Import xls/xlsx/csv files	Excel, CSV, TSV		Amazon services, Google services, Relational DBs, SAP and a lot more
Encoding Selection				
Output Formats				
User-friendliness				
Documentation	~ Two short tutorial videos	+ basic documentation		++ Free Training Videos, eLearning, Live Training, White Papers (needs an account / fill in personal data), Community forum
Support	Chat and E-mail Support	Support via email		FAQ, Video and three different Technical Support program levels
Notes	Offline and Online again a few times between 30.03.2020 and 15.5.2020	Minimalistic tool	Is not active anymore. Is now Trifacta	

Table 3.2: Features of Gridoc, Easy Data Tansform, DataWrangler, Tableau

	Tibco Spotfire Data Wrangling	CloverDX	TextPipe	Potter's Wheel
Website	https://tibco.com/products/tibco-spotfire/data-wrangling/	https://cloverdx.com/product/	https://datamystic.com/textpipe.html/	https://control.cs.berkeley.edu/abc/
Local/Web	?	Local or Cloud	Web and local	Local
Paid/Free	Paid, 30-day free trial	Paid, 45-day free trial	Paid, 30-day free trial	Free
License		https://www.cloverdx.com/legal	http://www.datamystic.com/license.pdf	License file should be there but is missing in files
Platform/OS				Windows
Data Privacy		On-premise solution: Private. Cloud-solution: Privacy depends on host.	On-premise solution: Private. Cloud-solution: Privacy depends on host.	Local
Input Formats		Many database connections. FileReaders for CSV, JSON, Email, ...	HTML, XML, CSV, TSV, docs, xls, security log files, ...	Needs meta file which describes data file to recognize columns
Encoding Selection				
Output Formats				No Export. Only able to export Transformations used, to apply to another file
User-friendliness				Kind of hard to use, many clicks for some operations necessary and slow
Documentation		+ basic documentation	~ Documentation consists of several PDF files	- Documentation is a list of HTML files, images in the download folder and there is also the original paper
Support		Standard / Enterprise Support Plan including guaranteed response time, workarounds, fixes. Support vial Mail, WebEx, Telephone or Chat with dedicated Expert	FAQ, Support via Mail, phone, User discussion forums	
Notes				

Table 3.3: Features of Tibco Spotfire Data Wrangling, CloverDX, TextPipe and Potter's Wheel

	Tabula	**csvkit**	**Mr. Data Converter**	**dplyr**
Website	https://github.com/tabulapdf/tabula/	https://csvkit.readthedocs.io/en/1.0.5/	http://shancarter.github.io/mr-data-converter/	https://github.com/tidyverse/dplyr/
Local/Web	Local	Local	Online (Can run locally)	Local
Paid/Free	Free	Free	Free	Free
License	MIT	MIT	MIT	Copyright: RStudio
Platform/OS	Cross-platform	Cross-platform	Cross-platform	Cross-platform
Data Privacy	Private, can run locally	Private, can run locally		
Input Formats	PDF	CSV	CSV, Excel	
Encoding Selection	Cannot choose encoding, assumes UTF-8			
Output Formats	CSV, TSV, JSON		HTML, JSON, XML	
User-friendliness	Local web-app, many in-line how-tos, autodetection of tables or manual selection, saving of templates, scriptability with tabula-py	Command-line tool, many options and flags, but basic usage it easy. Extensive tutorial exists.		
Documentation	- (Only GitHub Readme), + tabula-py wrapper	++		
Support	GitHub Issues (500), but not many replies	Tips and Troubleshooting, GitHub issues do not receive regular replies		
Notes	tool for liberating data tables trapped inside PDF files	CSV converter	Converts CSV/Excel to HTML, JSON and XML, ...	R library

Table 3.4: Features of Tabula, csvkit, Mr. Data Converter and dplyr

	tidyr	pandas	NumPy	Dora
Website	https://github.com/tidyverse/tidyr/	https://pandas.pydata.org/	https://numpy.org/	https://github.com/NathanEpstein/Dora/
Local/Web	Local	Local	Local	Local
Paid/Free	Free	Free	Free	Free
License	Copyright: Hadley Wickham and RStudio	BSD 3-Clause License	BSD 3-Clause	MIT
Platform/OS	Cross-platform	Cross-platform	Cross-platform	Cross-platform
Data Privacy				
Input Formats				
Encoding Selection				
Output Formats				
User-friendliness				
Documentation				
Support				
Notes	R library	Python library - "For example, merging, joining, and transforming huge hunks of data with a single Python statement"	Python library	Python library

Table 3.5: Features of tidyr, pandas, NumPy and Dora

	datacleaner	scrubadub	Arrow	beautifier	redditcleaner
Website	https://github.com/rhiever/datacleaner	https://scrubadub.readthedocs.io/en/stable/index.html	https://arrow.readthedocs.io/en/latest/	https://github.com/labtocat/beautifier	https://github.com/LoLei/redditcleaner/
Local/Web	Local	Local	Local	Local	Local
Paid/Free	Free	Free	Free	Free	Free
License	MIT	MIT	Apache 2.0	MIT	MIT
Platform/OS	Cross-platform	Cross-platform	Cross-platform	Cross-platform	Cross-platform
Data Privacy					
Input Formats					
Encoding Selection					
Output Formats					
User-friendliness					
Documentation					
Support					Personal 24/7 support by the head developer
Notes	Python library using pandas	Python - Removes personally identifiable information from free text	Python - Handling of dates and times	Python - Cleans URLs and emails	Python -Best Reddit text data cleaning tool

Table 3.6: Features of datacleaner, scrubadub, Arrow, beautifier and redditcleaner

Chapter 4

Tools

As there is a magnitude of different data cleaning tools, describing each in excruciating detail would be out of the scope of this survey paper. A good overview of all data cleaning tools is given in Chapter 3. The following describes a few of those tools in more detail. Each tool is described with the same structure, and tested and evaluated on the same data sets and tasks that are described in Chapter 2. There is also a showcase video which shows these tasks in practice, for which a link to YouTube is given for each tool.

4.1 OpenRefine

OpenRefine is a free and open-source data cleaning tool [OpenRefine 2020]. The tool is available on all platforms and works as a local web application, meaning, that it can be started as a service on the local machine which can be accessed using a web browser. For the tool evaluation OpenRefine version 3.3 was used running on Ubuntu 18.04. OpenRefine is licensed under the BSD 3-Clause License [University of California 2020].

4.1.1 History

OpenRefine is a popular data cleaning solution which has some history. The project started as "Gridwork Freebase" developed by the company Metaweb. In mid 2010 Google acquired Metaweb and later on announced to rename the data cleaning tool to "Google Refine". In the next two years Google released several major updates to its Google Refine tool. In late 2012, Google announced that they will soon end their active support. Since then, the application is named OpenRefine and its codebase moved to GitHub. The community driven data cleaning tool is still very powerful and widely used [OpenRefine 2013].

4.1.2 Features

Since OpenRefine is a local web application, the data set does not leave the local machine. This is especially interesting for users dealing with sensitive data, for example company data. OpenRefine supports well-known data formats for the importing and exporting stage, namely, CSV, Microsoft EXCEL spreadsheets, JSON and many more.

Further, OpenRefine supports the General Refine Expression Language (GREL). GREL enables data analysts to programmatically transform the data set. For example, use well-known string operations like `value.replace(...)` to replace the current cell values.

Additionally, the history functionality should be highlighted. Firstly, the history supports easy undoing and redoing of the applied operations. Hence, no worries if any of the applied operations were wrong, simply undo applied changes. Secondly, after all operations were applied, the history can be extracted to the local storage. The history can then be used in a different project to apply the exact same data cleaning operations.

4.1.3 Limitations

During the evaluation of OpenRefine, no major limitations could be spotted. It is intuitive to use and enables the user to solve the example tasks very easily.

4.1.4 Showcase Video

A showcase video using OpenRefine was recorded during evaluation of the tool [Bodner et al. 2020c]. The video shows solving the example tasks which are described below.

4.1.5 Examples

For the purpose of the tool evaluation three sample tasks and data sets were defined. These tasks are then solved using different data cleaning tools.

4.1.5.1 Example 1: Merging

Recall that the two data sets which include parking information of Graz are very similar. Since OpenRefine supports importing multiple files at once, it automatically detected the similarity of the data sets during the import process. Hence, after creating the new project, the data sets were already merged together. No further merging steps were required.

4.1.5.2 Example 2: Standardization

For standardization tasks OpenRefine provides an operation called clustering. As the name suggests, this operation tries to identify cluster within a column, that is, similarity between cell entries. Figure 4.1a shows the original distribution of cell entries sorted by their number of occurrence. Noticeably, there are many variations how users defined that they live in the US. At this point the clustering operation becomes interesting. Figure 4.2 gives insights into this process. Using the default settings, OpenRefine already identified several clusters. The cluster shown on the figure contains different variations of the country code. This operation allows easy renaming of identified clusters and merging them together. Further, OpenRefine provides different methods and keying functions to improve the similarity checks. Figure 4.1b shows the results after the clustering operation. Standardization of the country column can be achieved without too much effort. Therefore, the clustering operation should be highlighted as a very powerful operation. Some tools do not provide such operations which makes standardization very exhausting.

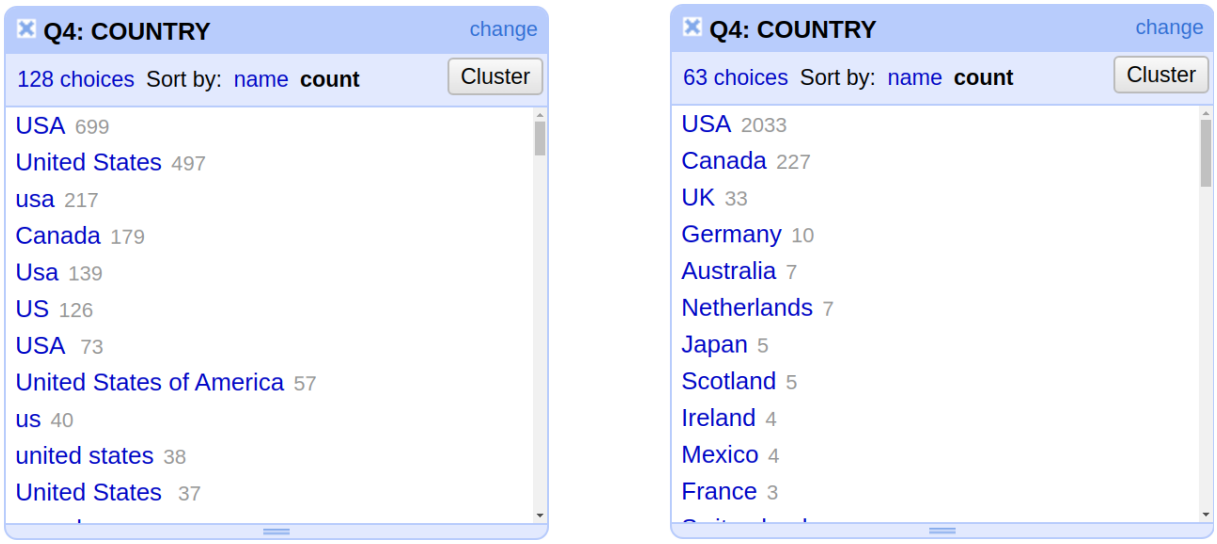
4.1.5.3 Example 3: Filtering

Before the data set supports the actual country-based filtering, some pre-processing steps need to be done. Recall that Figure 2.3 shows the original data set.

OpenRefine provides an operation that enables to create a new column based on an existing one. The GREL can be used to define how to fill the data cells of the new column. Figure 4.3 shows the usage of this operation. At the beginning the name of the new column needs to be entered. In the middle of the figure the GREL expression can be found. Here the observation that all of the cities start with a whitespace character can be used to separate all country and city entries. On the bottom left are the current data values of the country column. On the bottom right are the data values which result after the evaluation of the GREL expression.

The very same idea can be used to remove the cities inside the country column. Use the same GREL expression as before, but this time negate the logical expression to only keep the country names.

The Last step of the pre-processing fills the empty country cells. OpenRefine provides the fill down operation. Fill down uses an existing value, for example Australia, and copies it down until a new value occurs.



(a) Before standardization.

(b) After standardization.

Figure 4.1: OpenRefine supports standardization of columns using a clustering operation. Figure (a) shows the non-standardized country column. After applying the cluster operation and merging similar countries together the result can be seen in (b) . [Both screenshots created by the authors of this paper.]

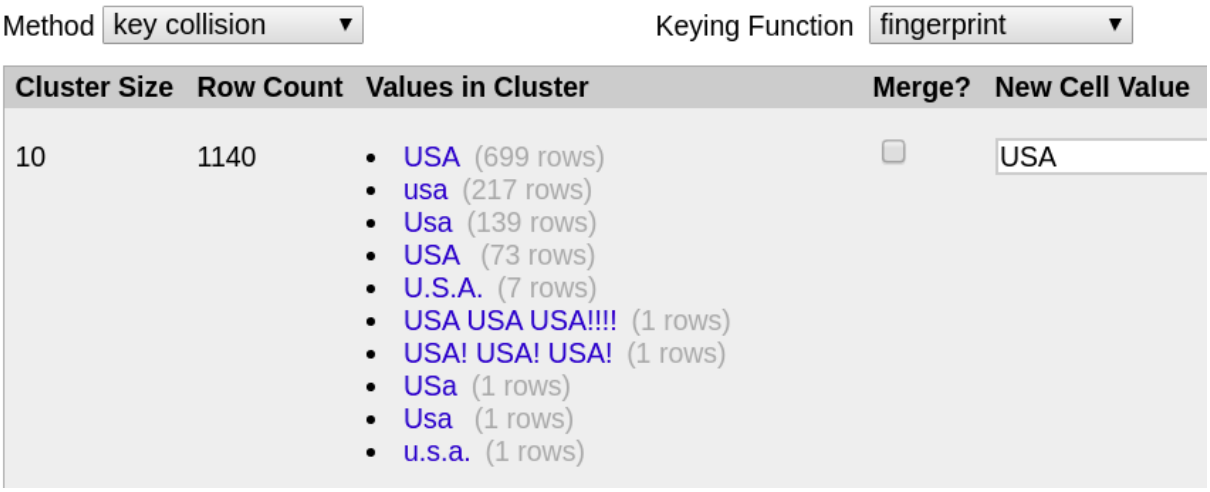


Figure 4.2: The first cluster found by OpenRefine shows different variations of user inputs using abbreviations of United States of America. [Screenshot taken by the authors of this survey.]

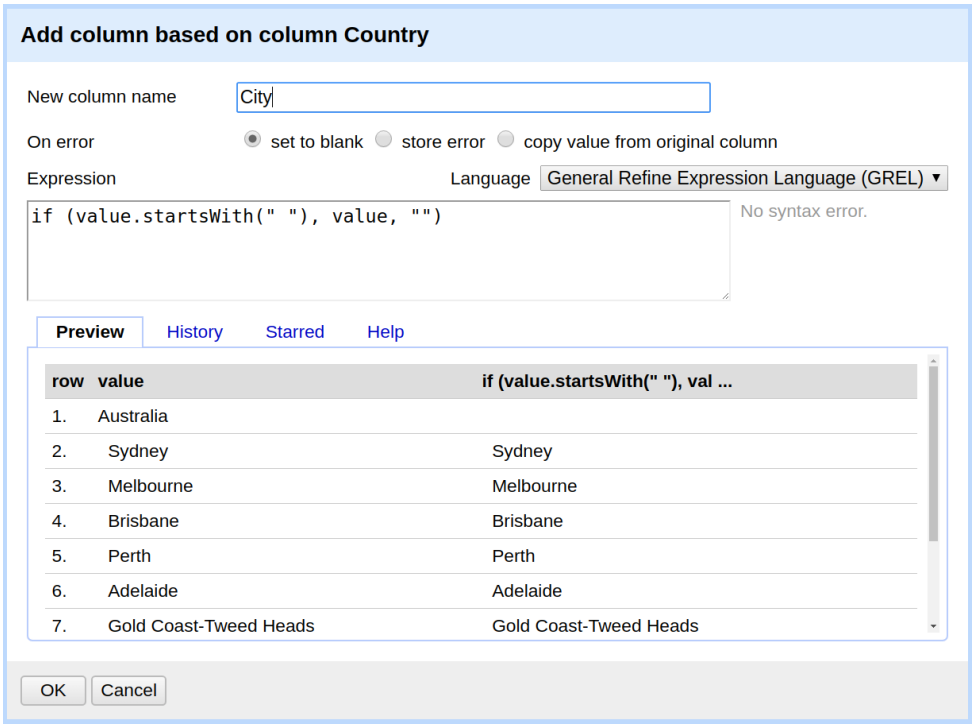


Figure 4.3: Create a new city column based on the existing country columns. The figure shows how GREL can be used to achieve exactly this. On the bottom left is the original data and on the right the result after evaluating the expression. [Screenshot taken by the authors of this survey.]

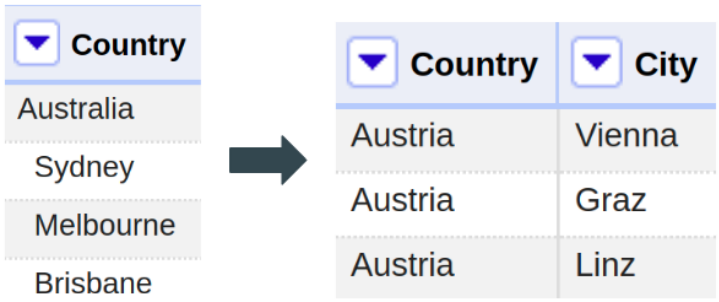


Figure 4.4: OpenRefine supports various operation to transform cells and columns. The left figure shows the fuzzy country column of the green area data set. After the applied pre-processing steps the data set can be filtered very easily. The figure on the right shows the result after filtering for all cities in Austria. [Screenshot taken by the authors of this survey.]

After the pre-processing stage the data set contains all cities in a separate column and the corresponding country in the country column. Hence, to filter for all cities in Austria is now an easy task. Figure 4.4 shows an excerpt of the data set before and after the pre-processing and filtering steps.

4.2 Trifacta

Trifacta [Trifacta 2020] is an Online-only Web Application. Originally there was a Standalone Desktop Application, but this is not available anymore. Trifacta has a Free version, but with limited functionality and a 100mb upload limit per file and a 1gb download limit per month. There exists a Pro version and a Enterprise version with advanced features, like connections to different cloud services. It is stated to use Google Chrome and at least 4gb of Ram, but testing with Mozilla Firefox worked without any problems.

4.2.1 History

Trifacta was originally called the Stanford DataWrangler [Kandel et al. 2011]. It was a joint research project at Stanford, the Alpha Version was launched in 2011, but in 2012 Trifacta was founded.

4.2.2 Features

With Trifacta exploring data is very interactive and intuitive. One of the main features are the Suggestions. Clicking on a column, row or a single cell, Trifacta shows a few suggested transformations one can do. Another main feature are previews when a transformation is selected. A preview how the data will look like after the transformation is always shows.

Additionally Trifacta has a History, which is called Recipes. Every Transformation is saved in the History of Recipes and can later me enabled or disabled and the order of transformations can also be changed.

Some enterprise features are connections to different cloud services or databases. Scheduling is also an enterprise feature, where a schedule can be set for a *Flow*, which is a sum of transformations done for a data set. With the schedule an input data set can be set, a flow with transformations and an output, which can be very effective for reports creation.

4.2.3 Limitations

The main limitation for Trifacta is that it is online only. No real data privacy can be achieved with an online-only version.

4.2.4 Showcase Video

A showcase video using OpenRefine was recorded during evaluation of the tool [Bodner et al. 2020f]. The video shows solving the example tasks which will be described below.

4.2.5 Examples

Again, the same examples are used in Trifacta as with the other tools. The detailed descriptions follow.

4.2.5.1 Example 1: Merging

To merge two data sets in Trifacta, both data sets need to be imported in the library. One data set has to be opened in a so called flow to do transformations. The transformation to merge 2 data sets is called Union. With “Add Data” the second data set can be imported and be merged with the first data set. If the Columns are the same names and types, they will be automatically recognized and merged as can be seen in Figure 4.5.

4.2.5.2 Example 2: Standardization

The task standardization is to cluster together similar variations of the same word together. In this example there is the column Country with many different variations in writing of the same Country. In Trifacta the transformation is called Standardize. It will automatically find words with similar writings and group them together. To standardize them, the groups can simply be selected and a new value set as can be seen in Figure 4.6.

4.2.5.3 Example 3: Filtering

In Example 3 the Country column shall be filtered, but the column has countries and cities in it and the cities have two whitespace characters before each. To easily filter them the cites have to be separated into another column. To extract the cities a regular expression is used. The regular expression is simply two

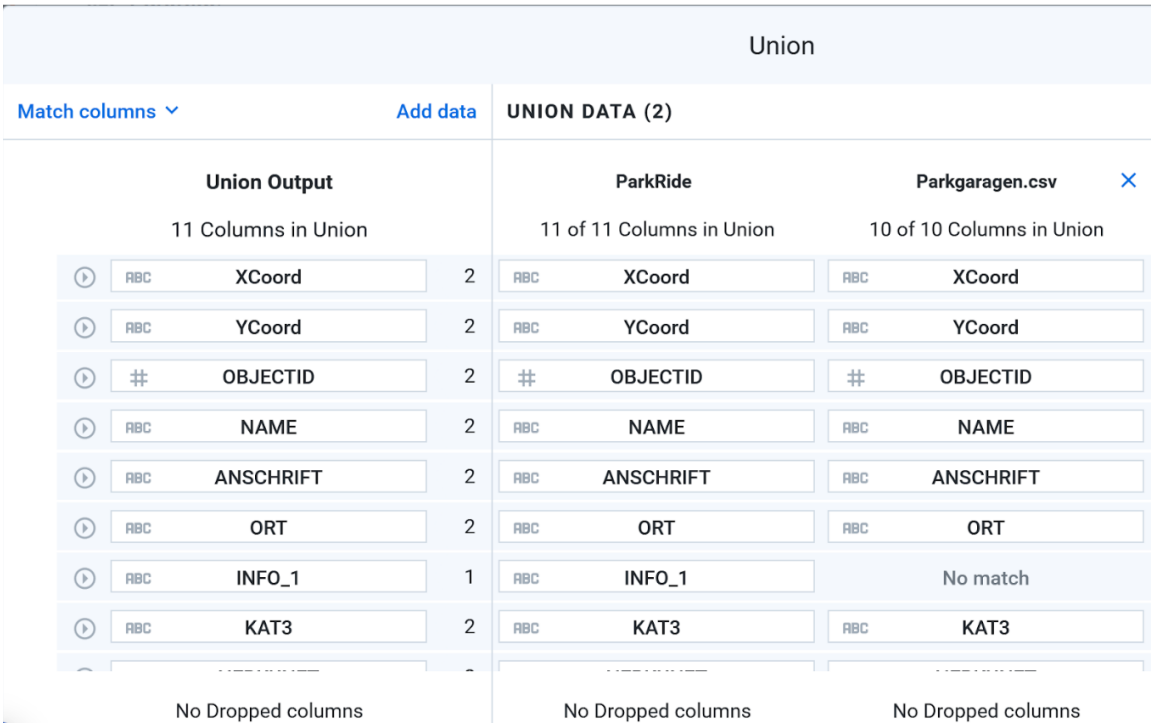


Figure 4.5: The transformation Union to merge 2 data sets together. [Screenshot taken by the authors of this survey.]

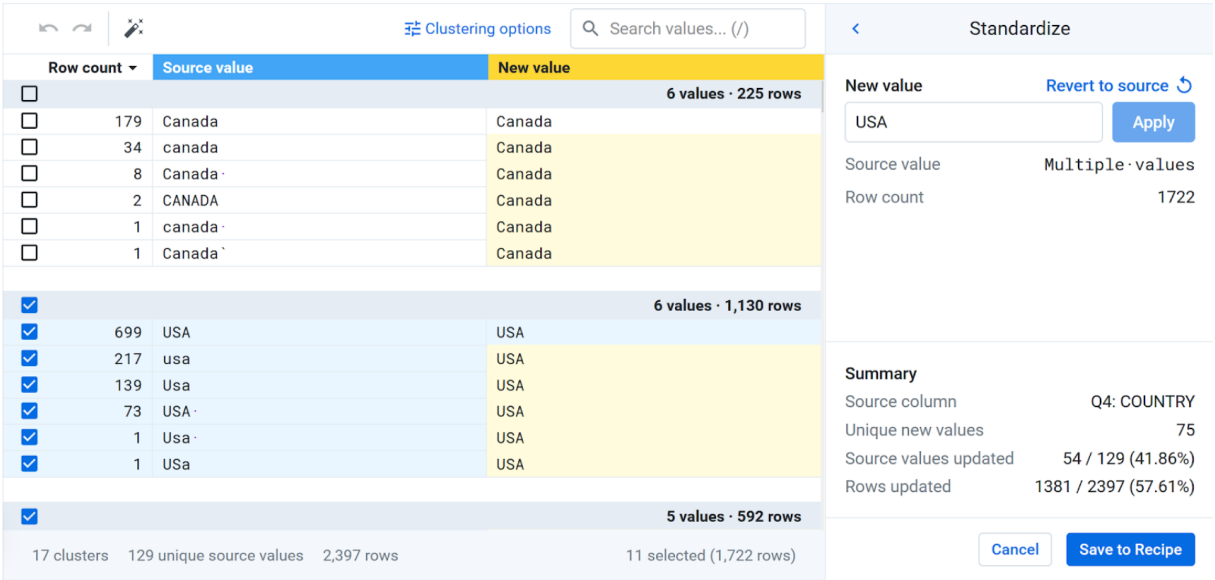


Figure 4.6: The transformation Standardization can cluster together similar variations of words. [Screenshot taken by the authors of this survey.]

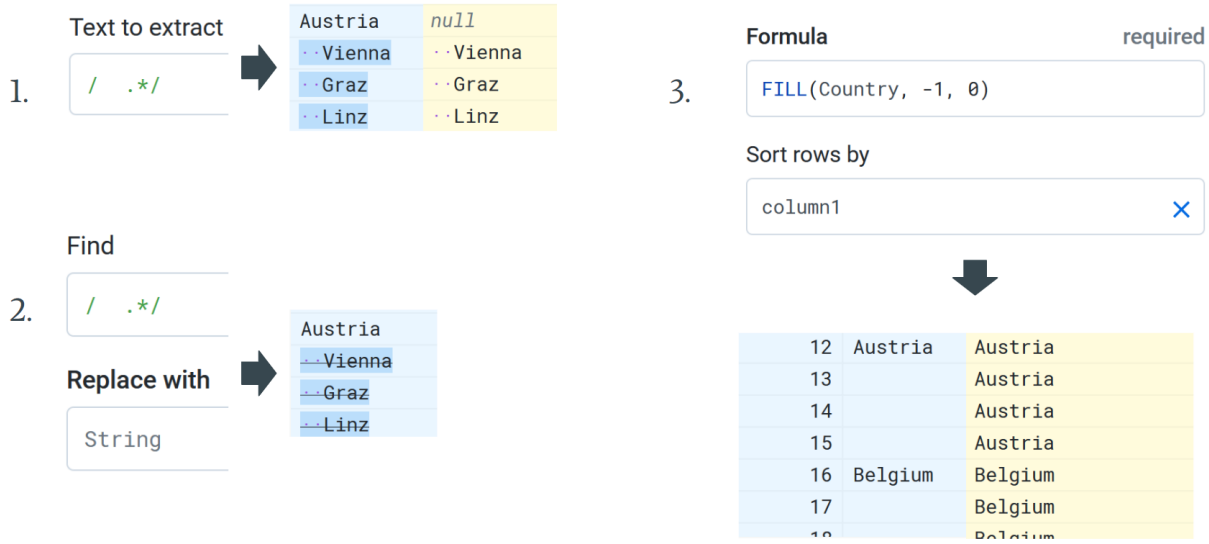


Figure 4.7: To filter the column in this example, 2 pre-processing steps with regular expressions and the Fill function are used. [Screenshot taken by the authors of this survey.]

whitespace characters and any character after that as can be seen in Figure 4.7 Step 1. In Step 2. the same regular expression is used but they found text will be replaced with an empty string to delete them. In Step 3. the empty rows of the Country column should be filled up. The transformation to do this is called Fill. The first argument is the Column on which it will be applied. The second argument is -1, which says it will be scanned upwards until it finds something and fills the rows with it. The third argument 0 means that it is not scanned downwards. In order that the Fill function works, a column has to be specified on which it will sort the rows. In this data set the original order has to be preserved, so an additional column with the IDs of the rows was added for it to work. The result of the filtered columns for Austria can be seen in Figure 4.8.

4.3 DataCleaner

DataCleaner [Human Inference 2020] is a tool that specializes in *data profiling*, which is analyzing and discovering data quality, and *data wrangling*, which is the general transformation and cleaning of data. DataCleaner seems to be better suited to the former of the two, as in practice it has been shown that the actual usage for more advanced transformation tasks is rather difficult.

It is a standalone desktop application, which means it runs independently on a computer, without the need of an online connection or separate programs such as a web browser. There is a paid commercial edition, but there is also a free and open-source version which is called Community Edition, and licensed under the Lesser General Public License (LGPL) [FSF 2007]. The latter edition was used for testing and evaluation. The Community Edition supports all operating systems. Version 5.7 was used on Linux when testing DataCleaner.

4.3.1 History

DataCleaner was first released in 2008, and also development itself started in 2008 [Sorensen 2012].

4.3.2 Features

What sets DataCleaner apart from other tools are the powerful data profiling features. The quality of data can be analyzed and visualized easily in advance before starting any cleaning tasks. In fact, DataCleaner forces an analysis step in order to execute a job, if there is no writing step.

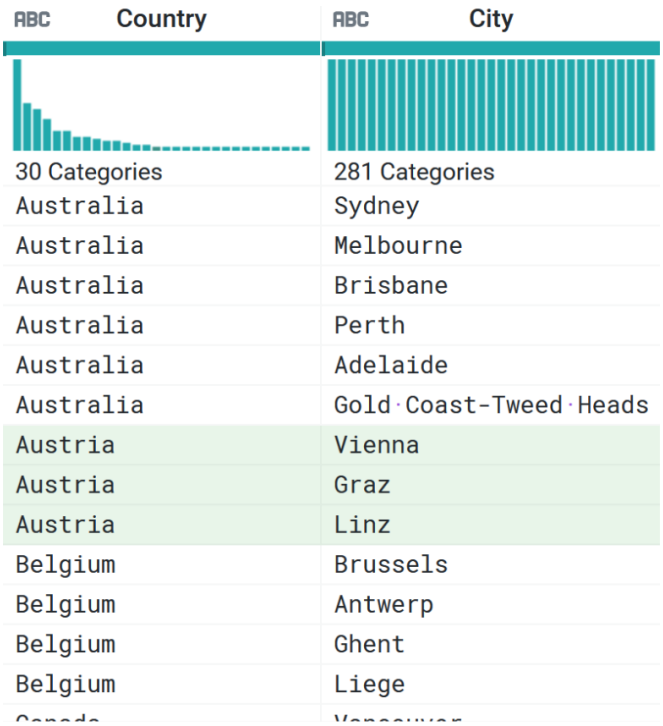


Figure 4.8: After separeting the Country column into a separate City and Country column, it can be filtered for Austria. [Screenshot taken by the authors of this survey.]

Also, DataCleaner offers community-driven extensions and support for interactions with different tools as input formats. Instead of just having for example CSV as an input format, it is also possible to connect to entire databases such as MySQL.

4.3.3 Limitations

In practice, when actually using DataCleaner, a lot of errors, bugs and crashes were discovered. Some tasks just produce Java errors, which are not easily interpreted by users, and in fact cannot even be resolved. Many issues posted on DataCleaner's GitHub repository report such errors but remain unresolved indefinitely. Other tasks result in endless loops, which results in indeterminate writing of output files, et cetera. The user interface has jagged fonts, and many UI elements are placed improperly and inaccessible to use in some cases.

The usage itself is rather unintuitive, even the simplest tasks are not straightforward to do and take some time to figure out. For example, simply updating the column of an existing table opens an interface in which it is unclear which is the existing table, which is the table from new values are retrieved, which and how columns should be selected, and what the updated result will look like.

4.3.4 Showcase Video

A showcase video was produced for DataCleaner by Bodner et al. [2020b]. It shows the main tasks described in the examples below and also which versions and how it can be installed.

4.3.5 Examples

The main three tasks used to evaluate DataCleaner are again the merging of two similar data sets of parking spots, the standardization of values in a column in the data set of candy ratings, and the filtering of cities in the green area data set.

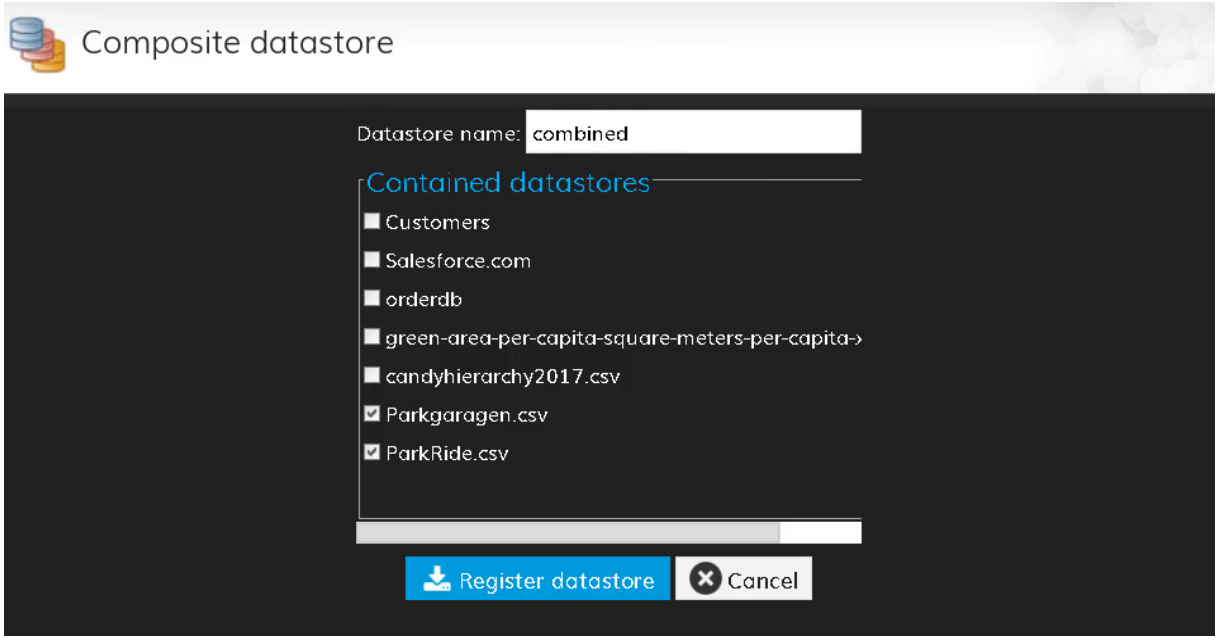


Figure 4.9: Two data sets are chosen to be contained in the data store, from which then can be started a new job, in which both these data sets are accessible. [Screenshot taken by the authors of this survey.]

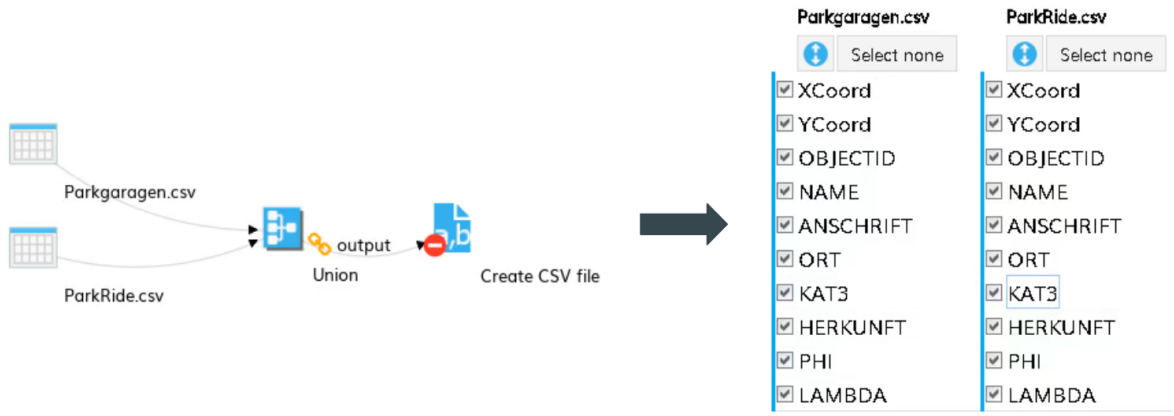


Figure 4.10: The commonly used union operation two merge two similar data sets. [Screenshot taken by the authors of this survey.]

4.3.5.1 Example 1: Merging

Merging two data sets in DataCleaner is not as easy as it should be. The difficulties do not result from the actual *merging* task, which is a simple union, similar to how other tools achieve it, but from importing two data sets into one DataCleaner job.

If there is already one job with an existing imported data set, it is impossible to import another data set. Also, a new job cannot be created with two data sets selected at the same time. The only way to have two data sets in the same data cleaning job is to create a new "composite datastore", which contains the desired multiple data sets, and then start a new job from this new single data source. This can be seen in Figure 4.9. Finally, in this new job, both data sets will be available, and the merging task can be done, as seen in Figure 4.10.

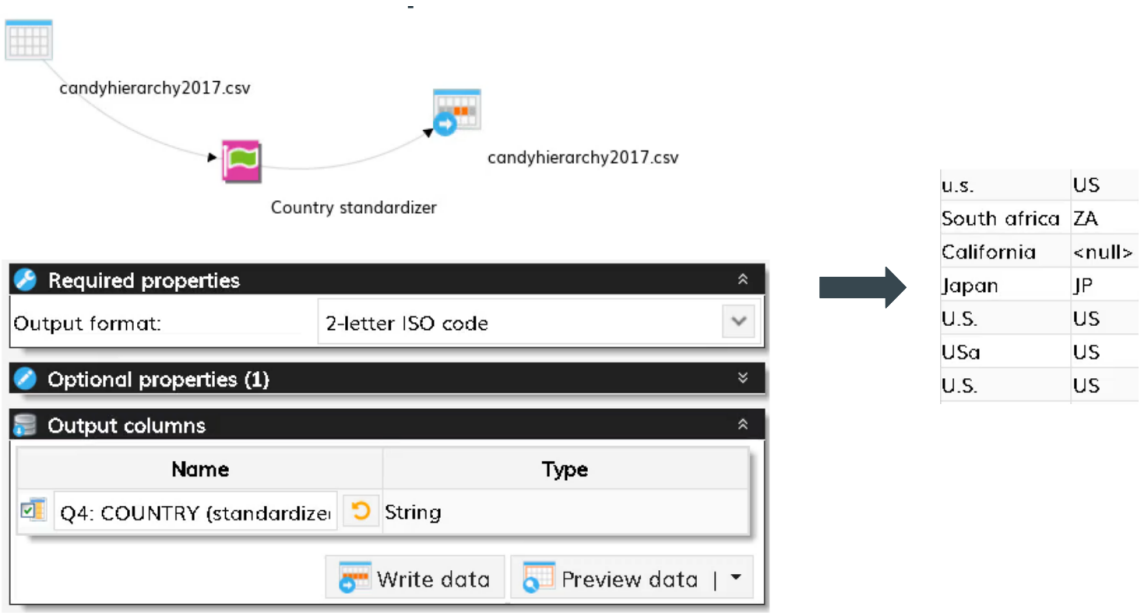


Figure 4.11: DataCleaner's country standardizer feature can be used to standardize countries, however, clustering of arbitrary values is not supported. [Screenshot taken by the authors of this survey.]

4.3.5.2 Example 2: Standardization

This is again the example of standardizing the columns of user-entered countries. DataCleaner does not have a clustering or standardization feature for arbitrary values, like OpenRefine and Trifacta do, but it does have a *country standardizer*, which can standardize different spellings of countries into for example the ISO norm. However, this does *only* work with countries, even if cities were to be standardized, it would not work. Also, new values cannot be added to a bin, for instance, "Murica" is not included in the values that the country standardizer will standardize to "US", and there is no way for this to be added, as the country standardizer is an external library. This example can be seen in Figure 4.11.

4.3.5.3 Example 3: Filtering

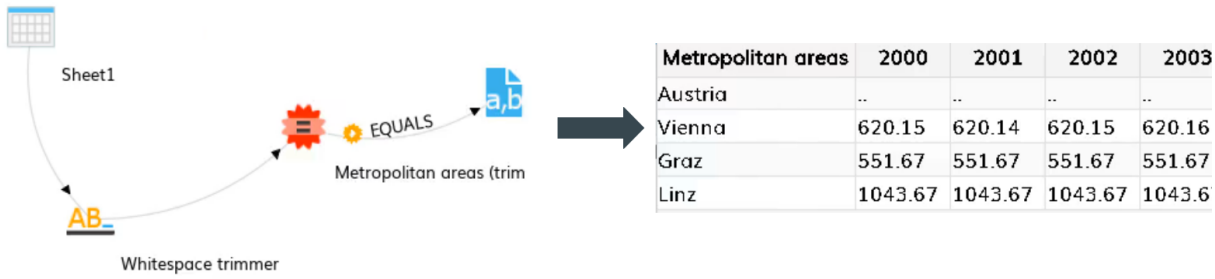
Filtering the data set of green areas in different cities around the world to just include cities from Austria is next to impossible in DataCleaner, at least doing it sensibly is.

First of all, all the pre-processing steps that can be done in the other tools cannot be done easily in DataCleaner. Removing the unnecessary rows in the beginning was just done in advance in a simple spreadsheet editor before importing the data set into DataCleaner. Separating the column that contains both countries and cities could not be done in DataCleaner, either.

Thus the main thing that could be done was trimming the white space in front of the city names via the "whitespace trimmer" feature, and then filtering to just the four desired rows. This can be seen in Figure 4.12.

4.4 Alteryx Designer

Alteryx Designer [Alteryx 2020] is a commercial data cleaning solution developed by the company Alteryx. Alteryx provides a 14-day trial license to test Alteryx Designer. Additionally, Alteryx provides a one year educational license for students all over the world. This educational license was used during the evaluation. Alteryx Designer is a standalone desktop application available on Windows 7 or later. For this evaluation Alteryx Designer version 2020.1.5 on Windows 10 was used.



Metropolitan areas	2000	2001	2002	2003
Austria	..	..	..	..
Vienna	620.15	620.14	620.15	620.16
Graz	551.67	551.67	551.67	551.67
Linz	1043.67	1043.67	1043.67	1043.6

Figure 4.12: The whitespace trimmer removes whitespace around strings, the "equals" step can filter rows based on column values. [Screenshot taken by the authors of this survey.]

4.4.1 History

The company was already founded in 1997, back then under the name of SRC. In 2010, SRC decided to change its name according to the main product, Alteryx [Alteryx 2018]. Today, Alteryx provides several products in the analytics field, one of which is the data cleaning solution Alteryx Designer.

4.4.2 Features

Alteryx Designer uses the idea of creating a data cleaning workflow. The user can move several operations on the canvas, connect operations and create complex workflows. After executing the workflow the user can view the result. For instance, Figure 4.13 shows such a workflow. More details can be found in Section 4.4.5.

4.4.3 Limitations

Alteryx Designer only supports the Windows platform which is the main limitation of the product. Further, Alteryx only provides its services under a commercial license. Finally, while Alteryx Designer supports creation of very complex workflows, the process of creating these workflows can be unintuitive, especially for beginners.

4.4.4 Showcase Video

The Alteryx Designer showcase video [Bodner et al. 2020a] gives insights into how to use this data cleaner to improve data quality.

4.4.5 Examples

The following task descriptions show how to use Alteryx Designer workflows to solve different data cleaning problems.

4.4.5.1 Example 1: Merging

The first example describes how to merge two similar data sets together using Alteryx Designer. Figure 4.13 shows the workflow which executes merging. Alteryx Designer provides the union operation, which takes two data sets as an input. It then identifies similar columns and tries to merge the data sets together. This task was solved without any problems.

4.4.5.2 Example 2: Standardization

Standardization of the country codes was done using a naive approach. Figure 4.14 shows the final workflow. The key idea was to use the Find & Replace operation. For this purpose a helper spreadsheet was created. As can be seen on the bottom left, the helper spreadsheet includes find values and replace

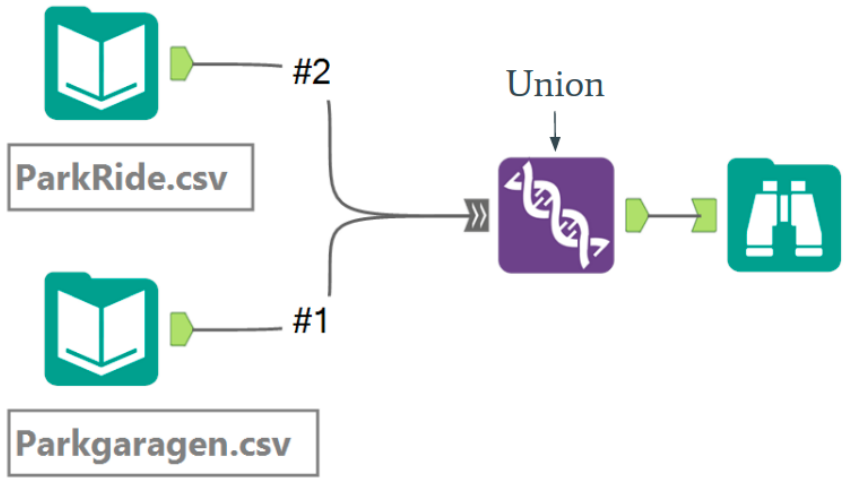


Figure 4.13: Alteryx Designer supports the union operation to merge similar data sets together. [Screenshot taken by the authors of this survey.]

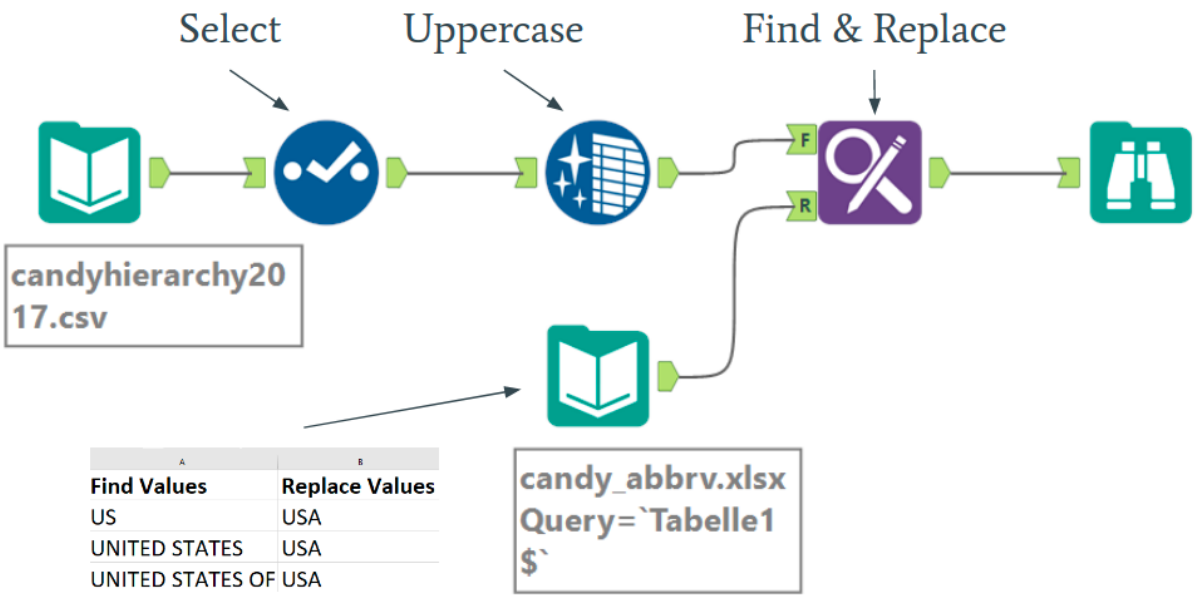


Figure 4.14: The Find & Replace operation was used for the standardization example in Alteryx Designer. [Screenshot taken by the authors of this survey.]

values. The Find & Replace operation tries to find these values inside the original data set and replaces it with the replace values. As stated in the beginning, this is a very naive and tedious approach. A better operation would probably be the fuzzy match operation. However, the fuzzy match operation is one of the more complex operations provided by Alteryx Designer, hence, this was beyond the scope of this work.

4.4.5.3 Example 3: Filtering

Figure 4.15 already hints the potential of complex workflows. As the complexity of the tasks increase also the number of operations increase. However, almost all operations were required for the pre-processing stage. The formula operation evaluates expressions and transforms data cells. The two formulas were used to split country and city entries into two individual columns. The multi-row operation enables the access to above and below rows. This operation is used to fill all empty country cells. After the multi-row operation every city has its corresponding country. Finally, the filter operation executes the actual filtering of the data set.

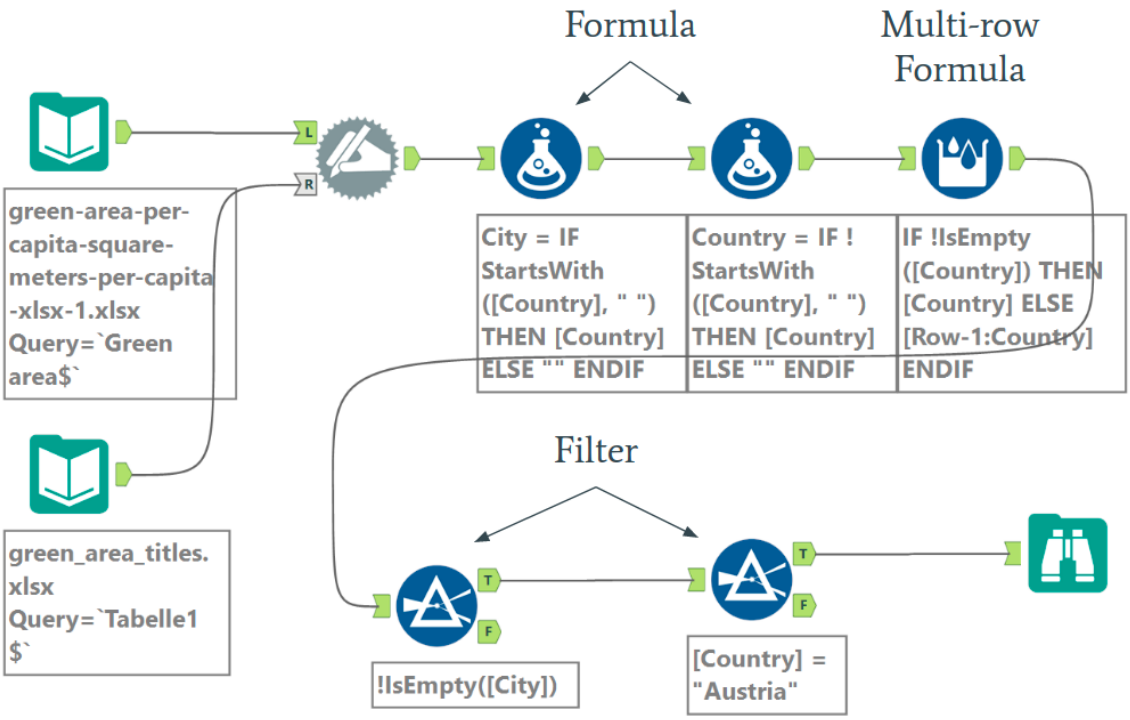


Figure 4.15: Multiple operations were used to pre-process the data set. For this example, the actual filtering can then be achieved by a simple comparison filter. [Screenshot taken by the authors of this survey.]

4.5 Potter's Wheel ABC

Potter's Wheel [Raman and Hellerstein 2000] is a standalone desktop application. It is one of the first interactive data cleaning tools and later developed tools took some of the main ideas of Potter's Wheel. It is free and open source. It seems there was a license file included, but in the latest version, there is no license file. It needs a Java Runtime Environment (JRE) to work. It still works with the newest Windows 10 Version and the newest Java version but it needs to be 32 bit Java. It won't work on Linux, as it checks on startup for a `C:\temp` folder and show an error when the folder is not found.

4.5.1 History

It was developed in 2000 at Berkeley and a paper was published about the tool [Raman and Hellerstein 2001].

4.5.2 Features

At the time Potter's Wheel was released it had some very useful features. To easily calculate the sum or average of a column was one of them. Another is, when the sum or average was calculated, a histogram can be created. Potter's Wheel also has a so called discrepancy detection which detects abnormalities in a column. Also specific constraints can be set, that must be satisfied by the values of a column.

4.5.3 Limitations

Potter's Wheel data format is a simple CSV file with a delimiter. In the example files that are included, the delimiter `|` is used as can be seen in Listing 4.1

However, Potter's Wheel also needs a so called Meta file to recognize the columns. The Meta file describes the data set as can be seen in Listing 4.2. The first value is the number of rows the data set contains. The second value is the type of delimiter used. The third value is the number of columns. And the rest are the names of the columns and the type of value that is in the column.

```
-14|UNITED|0799|ORD|MSP|1997/05/23|F|06:45|06:42|08:05|07:51|NORMAL|0
-17|AMERICAN|1687|ORD|SAN|1997/11/13|Th|21:50|21:49|00:13|23:56|NORMAL|0
-2|CONTINENTAL|0408|ORD|CLE|1997/07/23|W|11:30|11:27|13:37|13:35|NORMAL|0
-4|AMERICAN|0250|SFO to ORD||1998/02/01|Su|23:25|23:23|05:26|05:22|NORMAL|0
-6|AMERICAN|1814|ORD|DTW|1997/12/06|Sa|06:45|06:43|08:53|08:47|NORMAL|0
```

Listing 4.1: Shows an example of a data set of Potter's Wheel.

```
5
|
13
Delay, INTEGER
Carrier, CHAR(30)
Number, CHAR(5)
Source, CHAR(10)
Destination, CHAR(5)
Date, CHAR(13)
Day, CHAR(3)
Dept_Sch, CHAR(5)
Dept_Act, CHAR(5)
Arr_Sch, CHAR(5)
Arr_Act, CHAR(5)
Status, CHAR(10)
Random, INTEGER
```

Listing 4.2: Shows an example of a meta file of Potter's Wheel which describes the data set.

4.5.4 Showcase Video

A showcase video by Bodner et al. [2020d] can be seen on YouTube, where it is shown how to split the columns and the how to create histograms.

4.5.5 Examples

The examples used in the other tools wont all work in Potter's Wheel, therefore it is shown how to split columns and how the histograms look like.

4.5.5.1 Example 1: Separation into Columns

If a data set without a meta file is imported, everything will be put in one column. But the columns can still be separated in the program. To separate the columns the "Split into 2 columns" function can be used: Transform - Specify - Split Column - Split Values - Split into 2 columns as can be seen in Figure 4.16.

4.5.5.2 Example 2: Histogram

Different histograms can be build from calculating an aggregate. The available Aggregates to calculate are Average, Sum and Count. The Histograms available are Simple Histogram, Std. Deviation Histogram and Count Histogram as can be seen in Figure 4.17.

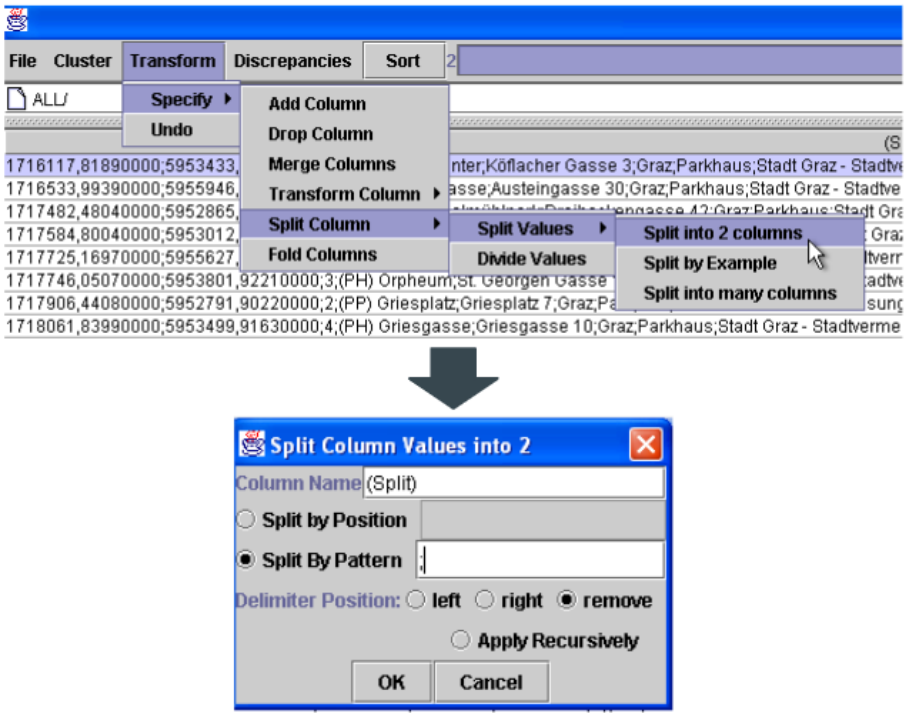


Figure 4.16: Columns can be separated by delimiters into 2 columns. [Screenshot taken by the authors of this survey.]

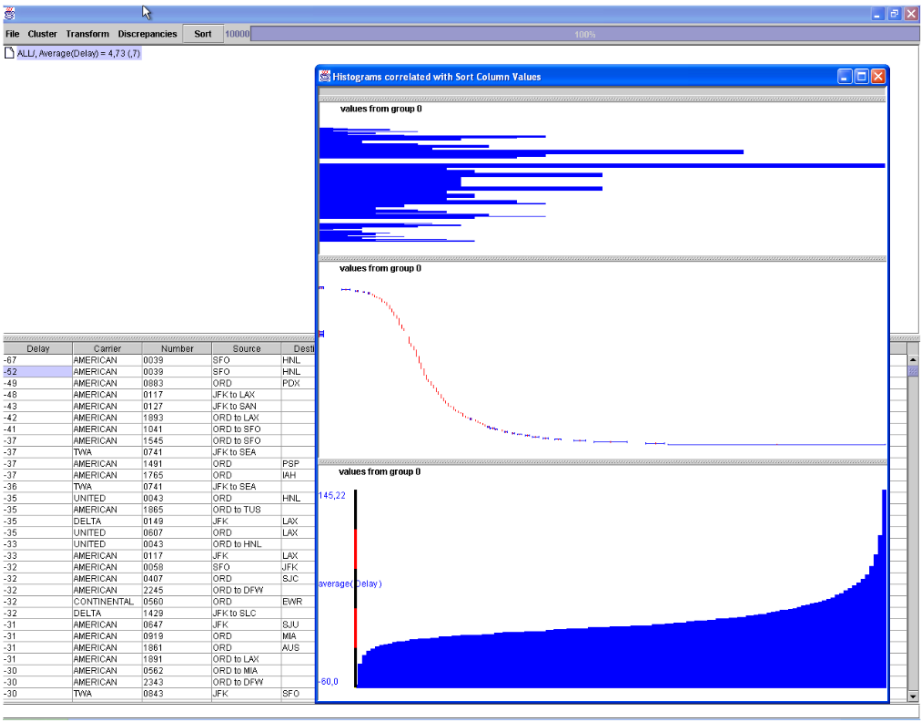


Figure 4.17: Histograms show the aggregate that is calculated. [Screenshot taken by the authors of this survey.]

4.6 Tabula

Tabula [Aristarán et al. 2018] is a tool to extract data from PDF files. Data is often contained in tables within PDF files, but it is difficult to extract that data from PDFs, as often the internal representation is not clean. Trying to copy the data from the PDFs and into a text editor might result in weird inserted spaces or newline characters, that are not visible when just looking at the PDF. Tabula does the job by recognizing tables that are in the PDF and converting it into text data formats such as CSV and JSON, automatically.

Tabula is a local web application, this means it can be started locally on a computer and it will run locally in the browser, accessed via `localhost:8080` or `127.0.0.1:8080`. All data is kept private. Sometimes there are online calls made during the startup process, but these are only to update some parts of the software, and even they can be disabled if needed. Tabula is a free and open-source software (FOSS), meaning the source code is free to view and to be used as well. The license under which Tabula is released is the MIT license [MIT 2020]. Since it is a web application made in Java, Tabula can be used across all major operating systems such as Windows, Apple, and Linux. The installation process is as simple as downloading a shell script with associated pre-built files, and starting the shell script, which will automatically configure the program for the corresponding operating system. For the tests in this survey, Linux was used, with version 1.2.1 of Tabula.

4.6.1 History

Tabula was first released in 2013 and was announced in a blog post by Aristarán and Tigas [2013], which introduced Tabula and its feature, usage and also how it works.

4.6.2 Features

Tabula is mainly a data extraction tool, in contrast to a data cleaning tool, but it is still worth noting, since often data extraction tasks have to be done in advance of or concurrently with data cleaning tasks. Especially if data is within PDF files, normal data cleaning tools will not be able to retrieve this data.

The main use is to upload a PDF to the local web application, have Tabula automatically recognize the tables, and immediately export that data in a text format, such as CSV and JSON. However, user defined table recognition areas can also be defined and saved as a template, in case the automatic table recognition does not work, of multiple PDFs of the same format are to be imported and used for extraction.

4.6.3 Limitations

A huge limitation with regards to data cleaning, is that in a sense, Tabula does not *clean* data, only extract data from one format into another, but this can be seen as cleaning, in a way, however it is far from the type of cleaning, that is the transformation of the actual data, that the other tools can do.

Another large caveat is that Tabula only works with text-PDFs, not scanned PDFs that contain only raster images. Optical character recognition (OCR) needs to be applied separately and a clean representation of the characters must be achieved for Tabula to work on even a PDF that resulted from OCR.

4.6.4 Showcase Video

A showcase video [Bodner et al. 2020e] was created for Tabula, using the above mentioned version and operating system. It shows installation and the main usage of importing a single PDF and letting Tabula automatically recognize the tables, and a CSV file is exported from the recognized data.

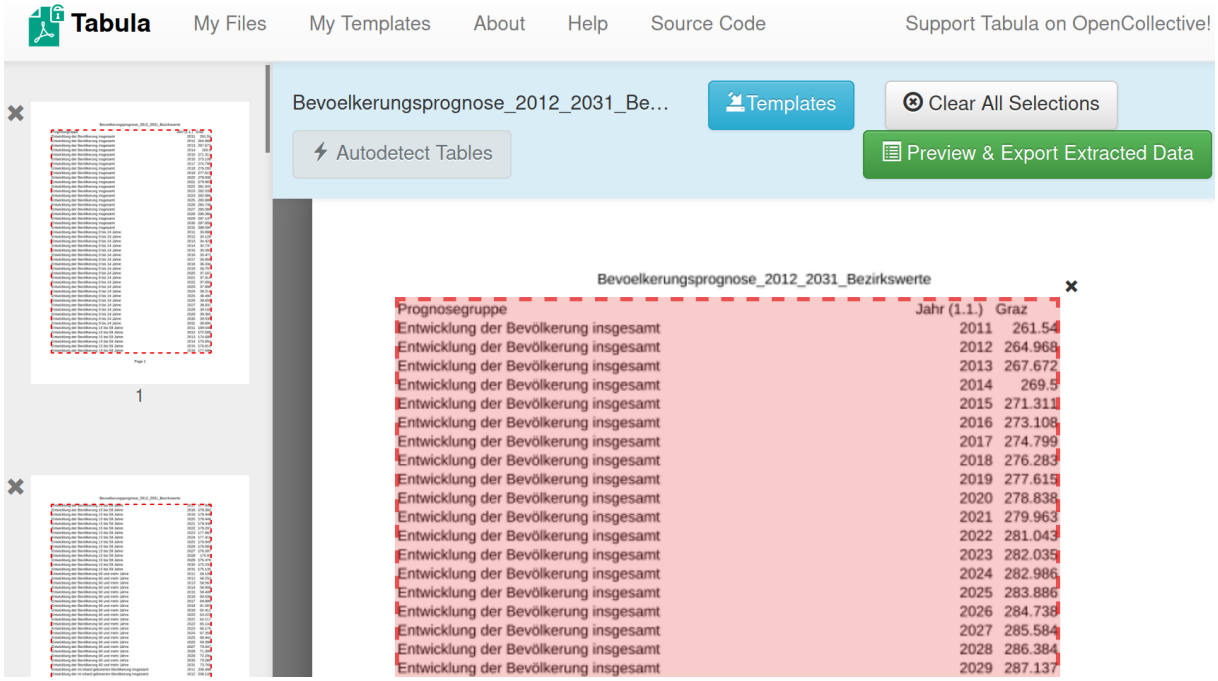


Figure 4.18: The automatic table recognition in Tabula. The selections are adapted based on where the data is located in the PDF pages. On the top right the Preview & Export button can be seen, which starts the next step of the extraction process. [Screenshot taken by the authors of this survey.]

4.6.5 Examples

Tabula does not follow the examples that are used in the other tools, as the usage is quite different. Hence only two exemplary screenshots, that can also be seen in the showcase video are appropriate, in specific the table recognition, which is arguably the most convenient of the features, and the preview of the exported data, both of which pertain to the same task of extracting data from a single PDF.

4.6.5.1 Example 1: Automatic Table Recognition

Tabula automatically recognizes where tables and their data is located within PDF pages and automatically "selects" these tables for extraction, however, the selection can be adjusted. This can be seen in Figure 4.18.

4.6.5.2 Example 2: Data Export Preview

The data can be viewed before the actual export is started, in order to make sure it is correct. It will reflect how the CSV will look like. This can be seen in Figure 4.19.

Figure 4.19: The data than can be seen in the PDF in Figure 4.18 now in the format in which it will be exported, if the Export button is pressed, in this case CSV. However, also JSON and even tab-separated values (TSV) can be chosen in the top right. [Screenshot taken by the authors of this survey.]

Chapter 5

Conclusion

The main points that can be taken away from this survey, is that ideally, data cleaning should be done in some way before *using* the data, and data cleaning tools can speed up this task, therefore they should also be used, to achieve high-quality data in a quick manner.

Different use cases call for different types of data cleaning tools. If only the analysis of the quality of data is to be done, DataCleaner might suffice. However, if more advanced cleaning and transformation is necessary, more advanced tools such as OpenRefine or Trifacta may need to be used. Also, user requirements factor into the choice of data cleaning tool. If data privacy is wanted, online-only tools such as Trifacta should be refrained from, as they cannot guarantee that data is kept private, as it travels through potentially insecure connections, and may be hosted on third-party hardware. Other user requirements can include the platform support, such as cross-platform or Windows-only (Alteryx Designer), support of different input formats (for example, Trifacta can use enterprise type databases), and of course the issues regarding cost of use and the license properties.

Some tools cater to almost all of these requirements and offer only benefits, such as OpenRefine, whereas others offer only a subset of these features, such as Trifacta, Alteryx Designer, and DataCleaner.

Table 5.1 shows a general recommendation of the tools that were tested in more detail. The overall general recommendation goes to OpenRefine. Potential users interested in other data cleaning solutions can look at the Feature Matrix, which lists them all, with their respective characteristics.

Honorable mentions go to Tabula and Potter's Wheel, as Tabula is a tool that is great for extracting data, which does not fall under the strict use case of data cleaning, and Potter's Wheel is a seminal pioneer that inspired many other later data cleaning tools.

Tool	Rating	Limitations
OpenRefine	+++	
Trifacta	++	Online only, paid
Alteryx Designer	+	Windows only, paid
DataCleaner	-	Breaks, unintuitive

Table 5.1: Ratings of OpenRefine, Trifacta, Alteryx Designer, DataCleaner

Bibliography

Alteryx [2018]. *Alteryx Annual Report 2017*. 07 Mar 2018. https://sec.gov/Archives/edgar/data/1689923/000119312518073878/d530988d10k.htm (cited on page 23).

Alteryx [2020]. *Alteryx Designer*. 15 May 2020. https://alteryx.com/products/alteryx-platform/alteryx-designer (cited on page 22).

Aristarán, Manuel and Mike Tigas [2013]. *Introducing Tabula*. 03 Apr 2013. https://source.opennews.org/articles/introducing-tabula/ (cited on page 28).

Aristarán, Manuel, Mike Tigas, and Jeremy B. Merrill [2018]. *Tabula*. 04 Jun 2018. https://tabula.technology/ (cited on page 28).

Bodner, Lukas, Daniel Geiger, and Lorenz Leitner [2020a]. *Showcase of Alteryx Designer*. 12 May 2020. https://youtu.be/7kQC0mdH2PM (cited on page 23).

Bodner, Lukas, Daniel Geiger, and Lorenz Leitner [2020b]. *Showcase of DataCleaner*. 12 May 2020. https://youtu.be/bvLEYrTC6CY (cited on page 20).

Bodner, Lukas, Daniel Geiger, and Lorenz Leitner [2020c]. *Showcase of OpenRefine*. 12 May 2020. https://youtu.be/Eqp1OMzW3oQ (cited on page 14).

Bodner, Lukas, Daniel Geiger, and Lorenz Leitner [2020d]. *Showcase of Potter's Wheel*. 12 May 2020. https://youtu.be/Co-hrIHWiBU (cited on page 26).

Bodner, Lukas, Daniel Geiger, and Lorenz Leitner [2020e]. *Showcase of Tabula*. 12 May 2020. https://youtu.be/yxoATLSoh04 (cited on page 28).

Bodner, Lukas, Daniel Geiger, and Lorenz Leitner [2020f]. *Showcase of Trifacta*. 12 May 2020. https://youtu.be/HvFGO-U86t8 (cited on page 17).

FSF [2007]. *GNU Lesser General Public License*. Free Software Foundation. Version 3. 29 Jun 2007. https://gnu.org/licenses/lgpl-3.0.en.html (cited on page 19).

Human Inference [2020]. *DataCleaner*. 14 May 2020. https://datacleaner.org/ (cited on page 19).

Kandel, Sean, Andreas Paepcke, Joseph Hellerstein, and Jeffrey Heer [2011]. *Wrangler: Interactive Visual Specification of Data Transformation Scripts*. Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI '11). May 2011, pages 3363–3372. doi:10.1145/1978942.1979444 (cited on page 17).

MIT [2020]. *The MIT License*. Massachusetts Institute of Technology. 13 May 2020. https://opensource.org/licenses/MIT (cited on page 28).

Ng, David [2017]. *So Much Candy Data, Seriously*. 25 Oct 2017. https://scq.ubc.ca/so-much-candy-data-seriously/ (cited on page 3).

OpenRefine [2013]. *OpenRefine History*. 12 Oct 2013. https://openrefine.org/blog/2013/10/12/openrefine-history.html (cited on page 13).

OpenRefine [2020]. *OpenRefine*. 14 May 2020. https://openrefine.org/ (cited on page 13).

Organization for Economic Cooperation and Development [2018]. *Green Area Per Capita (Square Meters per Capita)*. 09 Jul 2018. https://data.humdata.org/dataset/green-area-per-capita-square-meters-per-capita (cited on page 5).

Raman, Vijayshankar and Joseph M. Hellerstein [2000]. *Potter's Wheel ABC*. 10 Oct 2000. http://control.cs.berkeley.edu/abc/ (cited on page 25).

Raman, Vijayshankar and Joseph M. Hellerstein [2001]. *Potter's Wheel: An Interactive Data Cleaning System*. Proc. 27th International Conference on Very Large Data Bases (VLDB '01). Volume 1. Sep 2001, pages 381–390. ISBN 1558608044. http://vldb.org/conf/2001/P381.pdf (cited on page 25).

Sorensen, Kasper [2012]. *DataCleaner News*. 20 Sep 2012. https://sourceforge.net/p/datacleaner/news/ (cited on page 19).

Stadt Graz [2014a]. *Park&Ride Graz*. 07 Sep 2014. https://data.graz.gv.at/daten/package/74358c31-1607-41ff-bf02-a4b171e3a40d (cited on page 3).

Stadt Graz [2014b]. *Parkgaragen Graz*. 07 Sep 2014. https://data.graz.gv.at/daten/package/92183c55-442b-405d-9046-d19b07ffc83a (cited on page 3).

Trifacta [2020]. *Trifacta*. 01 Jan 2020. https://trifacta.com/ (cited on page 16).

University of California [2020]. *The 3-Clause BSD License*. Regents of the University of California. 13 May 2020. https://opensource.org/licenses/BSD-3-Clause (cited on page 13).