



Appendix A

Data Integrity and Cleaning

Introduction: If you've pulled your data off the web, or extracted it from a PDF, or obtained records that were entered sloppily by hand, you may find the formatting seems hopelessly mangled. If your data has been subject to optical character recognition (OCR)—converting the pixels of documents scanned as images into text—the imperfect process has likely introduced numerous more errors into the process. The excitement a journalist may feel when opening a dataset for the first time can quickly turn to disappointment when a mangled set of text and numbers with no clear structure appears on screen.

Some lines may end before others. Columns may be uneven and contain different types of data. Information that should be in its own column can be mashed together with other data.

This is “dirty data,” and cleaning it can easily become the most time-consuming and aggravating part of any data journalism project and one that can seriously undermine the quality of the journalism that results, if not completed correctly.

The precision of the techniques used in data journalism can create an inherent overconfidence. Sure, your Excel spreadsheet calculates percentages effortlessly, but if the underlying information is garbled, the analysis you perform is meaningless. In any journalistic work, it's important to think about the different ways the source data might have become dirty.

Remember that by the time a journalist gets a copy of a data set, it has already been handled by other humans, who, as we know, are entirely fallible and prone to errors. Much data is born when someone—maybe a government clerk or a parking control officer or a restaurant inspector—sits down in front of his or her computer and creates a record in a database.

Different databases constraint the way their users can enter data. Some may restrict entries in fields to a set of options in drop-down menus—perhaps choosing from one of several dozen municipal parking infractions or, from another, the race of a homicide victims. That means the data will at least be consistent in formatting, even if some selections may be made in error.

Other databases allow free form entry of long blocks of text to describe, for example, the extent and location of mouse droppings in a trendy restaurant. Commonly, most databases allow free text input of at least an address—a data chunk that is frequently subject to errors and inconsistencies in formatting.

After the initial creation of the data, many other humans will have gotten their thick fingers on it before it reached the reporter's hard drive. They could have reformatted, edited, updated, sliced, diced, severed, truncated and generally munged it in ways that could never be expected.

Sometimes the errors and inconsistencies will be readily apparent when the data is displayed on-screen. Someone may have entered street names in the addresses in all uppercase text while someone else used upper and lowercase. Street names such as "St. Martin" may variously appear as "Saint Martin" or "Saint-Martin." These small inconsistencies can affect the analysis of, for example, the street in the city with the most parking tickets. Excel would tabulate these as three different streets, not one.

Other errors may be harder to detect at first. An analysis of incidents in a database that is grouped by year or month may show that certain time periods appear to have been omitted from the data. If any area of town known for traffic problems doesn't show up in a database of parking tickets, there's probably something wrong with the source records. Or there may be very large values in the data, different from most, such as large numbers of fatalities or large dollar values. These outliers, as they are called, could point to important stories, or they may be mistakes.

These are the challenges of dirty data, and this appendix is all about finding and eliminating it.

Part 1. Thinking about the integrity and possible problems with your data.

In Chapter 2, we defined data for our purposes as structured information arrayed in rows and columns to facilitate analysis. That structured nature makes it remarkably easy to reorder, filter and summarize the data, to see what patterns and trends emerge. It's an incredibly powerful and flexible reporting and research tool.

But if we're not careful, all that flexibility and power can make trip us up. If we don't know how and where to look for them, we may miss errors and inconsistencies in the data that could undermine our analysis or make it just plain wrong. Such errors and inconsistencies will not always be obvious just by looking at the data on the screen.

The first thing to keep in mind is that there is nothing particularly special about information stored in a table of data. While it is organized in a systematic way, it is still just information collected by humans or automated systems built by humans. Those humans make a whole series of choices when collecting the data, choices about what information to collect and what to take a pass on, and about how to record the information that is collected, what categories to create, and so on. Those choices directly affect the analysis you will be able to do, and influence the conclusions you will reach.

For example, data on dispatch and arrival times of emergency responders such as paramedics and firefighters might seem on the surface to be precise and unimpeachable. A time is given for the time the personnel were dispatched to the scene and for when they arrived; it might be down to the

second. But how does it get into the database? A dispatch time might be entered automatically by a computer system when the dispatch message goes out, but how is the arrival recorded? If it relies on the emergency personnel reporting their arrival times by radio, they might be delayed doing so, preferring to deal with a life-threatening emergency first and reporting the time of arrival later. The resulting data could be inaccurate some or all of the time, but you'd never know that just looking at it.

Even if data is collected accurately, collection practices or definitions may change over time, with more or different data collected in subsequent months or years. Fields may be added to a dataset, but not populated. Or assumptions underlying data may change so that information in a field may not be strictly comparable to data added to the same field years earlier.

Therefore, some of the first questions you need to ask yourself when working with anything but the simplest data are who collected it, why did they collect it, and how did they go about it? The more you understand the data, and its potential strengths and limitations, the more useful and accurate you can make your analysis.

Understanding how the data was collected can also reveal where the data may be incomplete or in error. For example, if data is first entered on paper forms and later entered into a database by clerks, there is potential for transcription error, even if the data entry system uses safeguards such as dropdown lists of possible entries in a data field. Is the data double checked after it is entered? If so, does the same person who entered it check it, or another? The forms themselves could be filled out incorrectly, or the wrong information entered. Even information entered through handheld terminals is only as good as the choices made by the person entering it. In some cases, information may be gathered by automated means, such as a parking ticket terminal that automatically generates a location from GPS coordinates, but even then, machines can be improperly calibrated or malfunction. There are just so many ways bad data can get into a system. The very fact that data is collected and organized by people, and the systems they create, makes it vital that you understand how they go about the work.

Errors can also occur when data is released to a journalist. An agency may inadvertently leave out some rows from a dataset, rendering it incomplete.

There are steps you can take to root out such errors. One of the most important is to compare summary details of the data with publicly available figures. As an example, does the total value of contributions in a political donations dataset add up to the figures in official reports? Does the total number of bylaw infractions in a municipal dataset equal the total reported in information given to city council? If the numbers aren't close, you'll want to figure out why. Do the addresses of highly ticketed parking ticket locations reported in a database match up with real locations that you can find? Do the totals you get from summary analysis of a database make sense to people who are familiar with the sector you are investigating? A simple and effective technique is to run the results of your analysis past the agency that gave you the data. This can reveal not only errors in your analysis, but errors made by the agency in providing the data.

Another way to look for errors is by sorting different data columns. Such a sort can reveal unlikely large or small values, such as salaries in the millions in a small city or zero dollar values for contracts. Of course, if the values turn out to be valid, you might have an even better story.

Yet another step is to talk to outside groups that also deal with the data or the subject area, such as lobby groups, public interest organizations, academics, labour unions, and the like. Such individuals can help you determine if what you are seeing in the data makes sense, as well as help you identify likely weaknesses in the data collection, organization, or presentation.

The key goal is to become as expert as you can about the data and the subject area. This can go a long way toward identifying hidden weaknesses and invisible errors. At the very least you need to understand what the errors might be; and in the most extreme instance, you may decide that a dataset is simply unreliable. In that case, your story might be about how the data is collected, rather than on an analysis of the contents.

Part 2, cleaning up inconsistent and dirty data.

Other errors and inconsistencies will be more apparent, the data more obviously dirty. Data in some fields may be missing some of the time; entries in a text field may be spelled inconsistently or have leading or trailing spaces that make Phoenix with a space after it and Phoenix without a space after it look like different cities to the computer; dates or numbers may be in text fields, making numeric analysis impossible; data in a column in a spreadsheet may have **mixed formatting**, such as a mix of properly formatted dates and text entries that just look like dates; data may even end up in the wrong columns, leaving a mess that requires extensive manipulation before it can be analyzed. These kinds of problems need to be corrected before a useful and reliable analysis can be done. We're going to use the rest of this appendix to introduce you to a number of techniques that can be used to correct such errors.

There is no single tool for cleaning up data. In fact, a lot of cleaning can be accomplished using tools that we have already seen, such as spreadsheets and relational database programs. There is also a specialized tool called Open Refine that can take some of the tedium out of the job. And for the toughest problems, regular expressions can transform unorganized text into neat rows and columns. We'll be looking at each one.

Finding the problems

The first step is a bit of sleuthing, to find problems in your data. This work is most easily done in a spreadsheet or database program, though you can also explore your data in the Open Refine application we will discuss a little later.

If your file is already in a spreadsheet worksheet, or database table, go ahead and open it. If it's in a format such as delimited text (csv, tab delimited, etc) or JSON, you'll want to import it into a spreadsheet or database table first. If you want to convert a JSON file to an Excel table, a great tool that's easy to use can be found at <https://konklone.io/json/>. Just paste your JSON into the top window, and CSV appears in the bottom window. It's not perfect, but when it works more quickly than writing a custom script in a language such as Python.

We'll use Excel for the examples here, but SQL queries can be used just as easily for these tasks. We'll assume you've already read chapter 4 and its associated online tutorials.

Let's take a look at the Excel file of expenditures on credit cards issued by Halifax Regional Municipality, downloadable from the companion website to *The Data Journalist*.

ID	Line #	Stmt Date	Trans Date	Merchant Name	Comp Code	Cost Cent	G/L	Order #	Name	Reversal	Amount	Sales Tax	PST	Type
1	1	7 20140203	20140108	WAL-MART#3636	HROP		0	6404	RP9AMORO		121.87	15.9	0	Mastercard
3	2	8 20140203	20140112	WAL-MART#3021	HROP		0	6404	RP9AMORO		118.85	15.5	0	Mastercard
4	3	9 20140203	20140114	PIER 1 IMPORTS #1197	HROP		0	6404	RP9AMORO		31.05	4.05	0	Mastercard
5	4	10 20140203	20140114	GUY'S FRENCHYS	HROP		0	6401	RP9AMORO		203.41	25.36	0	Mastercard
6	5	11 20140203	20140114	WAL-MART #3636	HROP		0	6404	RP9AMORO		28.65	3.74	0	Mastercard
7	6	12 20140203	20140123	ATLANTIC SUPERSTORE #3	HROP		0	6404	RP9AMORO		39.67	2.25	0	Mastercard
8	7	18 20140203	20131129	HYPER PROMOTIONS	HROP		0	6399	RP9HXHPO		232.3	30.3	0	Mastercard
9	8	19 20140203	20140104	MICHAELS #3955	HROP		0	6404	RP9RX000		19.54	2.55	0	Mastercard
10	9	20 20140203	20140106	CDN TIRE STORE #00224	HROP		0	6705	RP9RX000		49.05	6.4	0	Mastercard
11	10	21 20140203	20140113	TARGET CANADA T3731	HROP		0	6404	RP9HX001		133.11	17.36	0	Mastercard
12	11	27 20140203	20140108	MICHAELS #3955	HROP		0	6404	RP9FX000		52.29	6.82	0	Mastercard
13	12	62 20140203	20140108	COLES 221	HROP		0	6404	RP9AMORO		35.68	1.7	0	Mastercard
14	13	63 20140203	20140108	WAL-MART #3636	HROP		0	6404	RP9AMORO		356.3	41.48	0	Mastercard
15	14	64 20140203	20140110	CAPITAL SAFE LOCK SER	HROP		0	6404	RP9AMORO		350.95	45.78	0	Mastercard
16	15	87 20140203	20140109	RONA PIERCEYS HALIF 46	HFRM		0	304	COHOON, GORDON		4.12	0	0	Mastercard
17	16	101 20140203	20140110	AIR CAN 0142128883354	HROP		0	800	INNESS HANN, JOANNE	Y	-880.73	0	0	Mastercard
18	17	102 20140203	20140110	AIR CAN 0142128884596	HROP		0	800	INNESS HANN, JOANNE	Y	-880.73	0	0	Mastercard
19	18	119 20140203	20140110	SUBWAY	HROP		0	6919	RP9YAW03		14.95	1.95	0	Mastercard
20	19	120 20140203	20140110	CINEPLEX 5145	HROP		0	6919	RP9YAW03		30	0	0	Mastercard
21	20	273 20140203	20140102	SOBEYS #644 QPS	CENT		0	304	POTERI, STUART		3.83	0	0	Mastercard
22	21	275 20140203	20140106	CDN TIRE STORE #00465	CENT		0	304	POTERI, STUART		35.03	0	0	Mastercard
23	22	277 20140203	20140109	BIG ERIC'S INC	CENT		0	304	POTERI, STUART		46.5	0	0	Mastercard
24	23	279 20140203	20140114	FILTRATION PLUS LTD.	CENT		0	304	POTERI, STUART		52.96	0	0	Mastercard
25	24	281 20140203	20140115	PARTS FOR TRUCKS INC	CENT		0	304	POTERI, STUART		36.29	0	0	Mastercard
26	25	283 20140203	20140120	SOBEYS #644 QPS	CENT		0	304	POTERI, STUART		4.5	0	0	Mastercard
27	26	287 20140203	20140127	BIG ERIC'S INC	CENT		0	304	POTERI, STUART	Y	-3	0	0	Mastercard

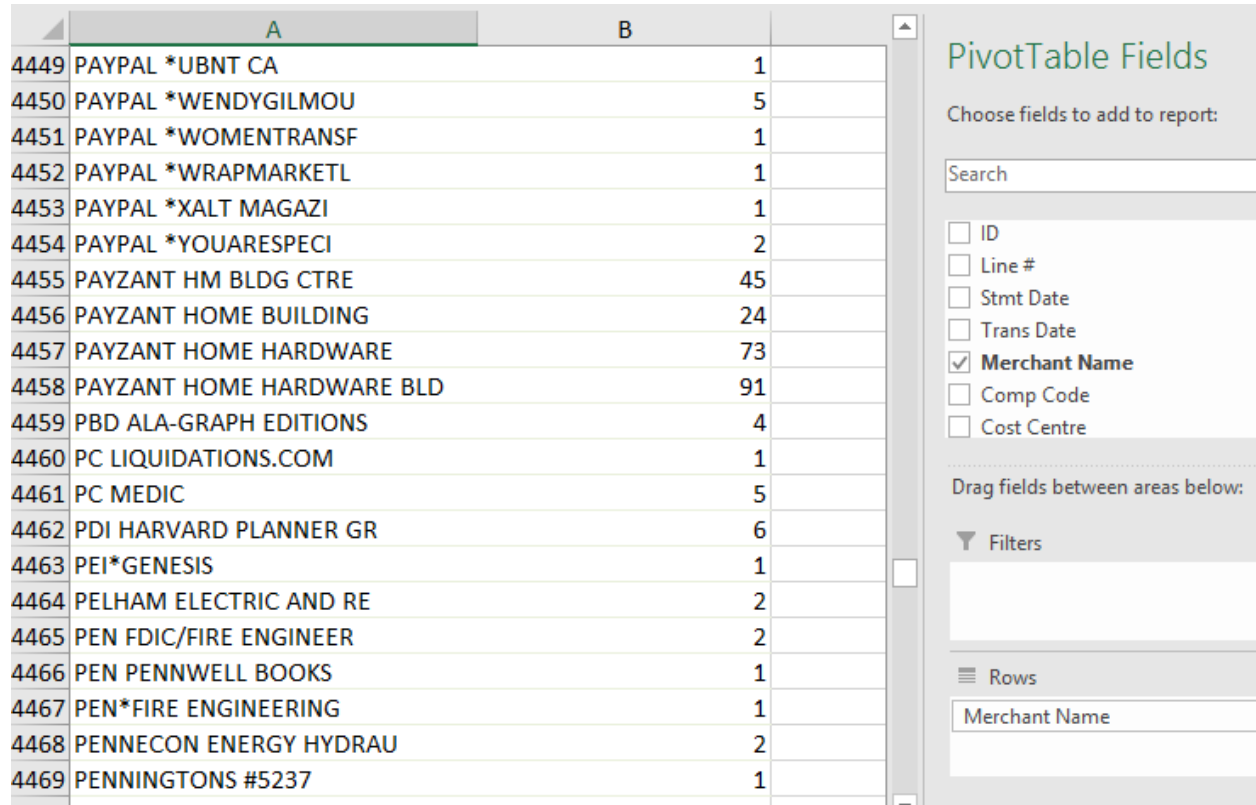
The data doesn't look too bad at first glance, but even a cursory look shows one issue; the two dates are in text format without any separators between the year, month and day. You can tell they are in text format because the data is aligned to the left side of the cell. We'd probably like to get the dates into a proper date format so we can do things such as extract the month or year using date functions.

We can also see a few of what appear to be inconsistencies in names in the Merchant Name field. A quick way to determine if there are many such inconsistencies is to sort the column in ascending order.

D	E	F	G
20140828	PAYZANT HM BLDG CTRE	HROP	W183
20140901	PAYZANT HM BLDG CTRE	HROP	R735
20140924	PAYZANT HM BLDG CTRE	HROP	R742
20141028	PAYZANT HM BLDG CTRE	HROP	R715
20141021	PAYZANT HM BLDG CTRE	HROP	R735
20141030	PAYZANT HM BLDG CTRE	HROP	R740
20141029	PAYZANT HM BLDG CTRE	HROP	R742
20141106	PAYZANT HM BLDG CTRE	HROP	0
20141106	PAYZANT HM BLDG CTRE	HROP	0
20141106	PAYZANT HM BLDG CTRE	HROP	0
20141126	PAYZANT HM BLDG CTRE	HROP	C330
20141120	PAYZANT HM BLDG CTRE	HROP	F160
20141106	PAYZANT HM BLDG CTRE	HROP	R735
20141113	PAYZANT HM BLDG CTRE	HROP	R827
20130715	PAYZANT HOME BUILDING	HROP	R741
20130729	PAYZANT HOME BUILDING	HROP	R742
20130712	PAYZANT HOME BUILDING	HROP	R742
20130722	PAYZANT HOME BUILDING	HROP	R742
20130710	PAYZANT HOME BUILDING	HROP	W182
20130801	PAYZANT HOME BUILDING	HROP	0
20130806	PAYZANT HOME BUILDING	HROP	R735
20130806	PAYZANT HOME BUILDING	HROP	R735
20130806	PAYZANT HOME BUILDING	HROP	R735
20130812	PAYZANT HOME BUILDING	HROP	R742
20130819	PAYZANT HOME BUILDING	HROP	R742
20130821	PAYZANT HOME BUILDING	HROP	R742
20130809	PAYZANT HOME BUILDING	HROP	W182
20130809	PAYZANT HOME BUILDING	HROP	W182
20130807	PAYZANT HOME BUILDING	HROP	W185
20131105	PAYZANT HOME BUILDING	HROP	D550

As we scroll down, we can definitely see inconsistencies. Such inconsistencies will make summary analysis difficult because each unique spelling in the field will be treated as if it were a different merchant.

An even more effective way of looking for this kind of problem is to create a pivot table, counting how many unique instances there are.



The screenshot shows an Excel spreadsheet with two columns, A and B. Column A contains merchant names, and column B contains their respective counts. The PivotTable Fields task pane on the right is open, showing the 'Merchant Name' field selected for the report.

	A	B
4449	PAYPAL *UBNT CA	1
4450	PAYPAL *WENDYGILMOU	5
4451	PAYPAL *WOMENTRANSF	1
4452	PAYPAL *WRAPMARKETL	1
4453	PAYPAL *XALT MAGAZI	1
4454	PAYPAL *YOUARESPECI	2
4455	PAYZANT HM BLDG CTRE	45
4456	PAYZANT HOME BUILDING	24
4457	PAYZANT HOME HARDWARE	73
4458	PAYZANT HOME HARDWARE BLD	91
4459	PBD ALA-GRAPH EDITIONS	4
4460	PC LIQUIDATIONS.COM	1
4461	PC MEDIC	5
4462	PDI HARVARD PLANNER GR	6
4463	PEI*GENESIS	1
4464	PELHAM ELECTRIC AND RE	2
4465	PEN FDIC/FIRE ENGINEER	2
4466	PEN PENNWELL BOOKS	1
4467	PEN*FIRE ENGINEERING	1
4468	PENNECON ENERGY HYDRAU	2
4469	PENNINGTONS #5237	1

PivotTable Fields
Choose fields to add to report:
Search
 ID
 Line #
 Stmt Date
 Trans Date
 Merchant Name
 Comp Code
 Cost Centre
Drag fields between areas below:
Filters
Rows
Merchant Name

Now we can see that there are four different spellings of the Payzant hardware store, plus how many there are.

If you are unsure how to create a pivot table, see Chapter 4, as well as the tutorial on pivot tables on the companion website.

We've already got two cleaning tasks to do. Let's keep looking for more problems.

It's always a good idea to sort any numeric fields, including currency fields, to see if there are any unlikely or obviously errant values in them. So let's sort the Amount field in our credit card data. We'll start with ascending order.

	K	L	M	N
	Reversal	Amount	Sales Tax	PST
RIE H	Y	-1,763.48	0	
	Y	-1,342.02	-175.05	
L	Y	-1,288.00	-168	
	Y	-1,207.50	-157.5	
	Y	-1,149.94	-132.29	
JE	Y	-1,072.64	0	
NNNE	Y	-1,062.39	0	
	Y	-1,037.88	-135.38	
RIE H	Y	-1,035.20	0	
RIE H	Y	-1,035.20	0	
	Y	-1,025.12	0	
	Y	-998.77	-130.27	
	Y	-998.05	-130.18	
.AS	Y	-977.5	-127.5	
	Y	-975.91	-112.27	
	Y	-959.41	-125.14	
	Y	-942.02	-122.87	
	Y	-940.36	-122.66	

We see some negative values here, but notice that they seem to be paired with the Reversal column. These appear to be refunds, and the values look plausible. Let's sort the other way, in descending order.

	E	F	G	H	I	J	K	L	M	N	O
1	Merchant Name	Comp Code	Cost Centre	G/L	Order #	Name	Reversal	Amount	Sales Tax	PST	Type
2	Merchant Name	Comp Code	Cost Centre	G/L	Order #	Name		Amount	Sales Tax		Visa
3	TIM HORTONS H/O QTH	HR0P	F141	6938		HOLLETT, ROY		11,880.00	0	0	Mastercard
4	TIM HORTONS H/O	HR0P	F141	6938		HOLLETT, ROY		11,880.00	0	0	Visa
5	IMP SOLUTIONS	HR0P	R102	6204		KAISER, JOSEPH		9,337.54	1,217.94	0	Visa
6	ADVANTAGE WIRELESS	HR0P	A421	6204		CURRIE, TAMMY		8,890.65	1,159.65	0	Visa
7	IMP SOLUTIONS	HR0P	A743	6706		KAISER, JOSEPH		8,718.15	1,137.15	0	Mastercard
8	CDW CANADA	HR0P	A743	6706		KAISER, JOSEPH		7,912.00	1,032.00	0	Mastercard
9	CDW CANADA	HR0P	A743	6706		KAISER, JOSEPH		7,237.48	944.02	0	Mastercard
10	ADVANTAGE WIRELESS	HR0P	F190	6711		CURRIE, TAMMY		7,136.08	930.79	0	Visa

This turns up a duplicated header row. Not a huge deal, but worth eliminating to get the data into the best shape.

Alright, let's use another file to look at a couple of more problems that can crop up. It's called `misalignedcolumns.xlsx`, and it's also downloadable from the companion site.

1	A	B	C	D	E	F	G
ID	Contract Date	Vendor Name	Description of Work	Contract Value			
2	1	2005-04-01	OTIS CANADA INC	859 - Other Business Services not Elsewhere Specified	\$17,027,229.55		
3	2	2005-04-08	THYSSENKRUPP ELEVATOR (CANADA) LTD.	639 - Institutional Buildings	\$7,016,872.41		
4	3	2008-04-22		THE ARCOPI GROUP / GERSOVITZ MOSS	423 - Engineering Consultants - Other	\$76,300,367.96	
5	4	2008-04-22	PRIESTMAN NEILSON & ASSOCIATES LTD	423 - Engineering Consultants - Other	\$627,462.51		
6	5	2009-04-01		OTIS CANADA INC	859 - Other Business Services not Elsewhere Specified	\$518,671.21	
7	6	2009-04-01	OTIS CANADA INC	859 - Other Business Services not Elsewhere Specified	\$90,212.78		
8	7	2009-04-01	OTIS CANADA INC	859 - Other Business Services not Elsewhere Specified	\$410,544.95		
9	8	2009-04-01		OTIS CANADA INC	859 - Other Business Services not Elsewhere Specified	\$273,195.39	
10	9	2009-04-01	OTIS CANADA INC	859 - Other Business Services not Elsewhere Specified	\$32,835.87		

Again, a quick glance shows an issue right away; some of the data columns aren't aligned properly. This can happen when importing CSV files that may have missing or extra delimiters; it can also happen with data scraped from the web, extracted from PDF files, or pasted from HTML tables. If we sort the Contract Value column in descending order, we can get a sense of how many rows may be affected.

D	E	F	G
Description of Work	Contract Value		
BRITISH COLUMBIA SAFETY AUTHORITY	881 - Construction Services	\$389,771.57	
CHARLOTTE COUNTY REFRIGERATION LT	881 - Construction Services	\$11,849.18	
RENOVATION MARC CLEROUX.INC	881 - Construction Services	\$29,509.95	
H.M.T. MECHANICAL	881 - Construction Services	\$10,509.00	
NESBITT ENGINEERING LTD	881 - Construction Services	\$11,789.29	
CARMICHAEL ENGINEERING LTD.	881 - Construction Services	\$39,547.74	
WISNIA INC.	881 - Construction Services	\$11,113.55	
FLYNN CANADA LTD.	881 - Construction Services	\$226,868.25	
LUMECH PLUMBING & HEATING LTD	881 - Construction Services	\$43,787.50	
LES CONSTRUCTIONS GILGA LTÉE	881 - Construction Services	\$18,275.28	
AVONDALE CONSTRUCTION LIMITED	881 - Construction Services	\$8,779,501.10	
HONEYWELL LIMITED / HONEYWELL LIMI	881 - Construction Services	\$104,824.86	
KUDLIK CONSTRUCTION LTD.	881 - Construction Services	\$181,335.00	
SBL ELECTRIC INC.	881 - Construction Services	\$19,362.55	
ALLIANCE ENGINEERING & CONSTRUCTI	881 - Construction Services	\$14,469.65	
OTIS CANADA INC	859 - Other Business Services not Elsewhere Specified	\$518,671.21	
	859 - Other Business Services not Elsewhere Specified	\$273,195.39	

It seems quite a few.

If you switch to the second worksheet in this file, Numbers as Text, you can immediately see an issue with the Contract Value column; the values are in text format. This is a similar problem to that which we saw with the dates in the credit card data, but the solution is different. If we tried to sort on these dollar values, they would sort in alphabetical, rather than numeric format:

	D	E	F
	499 - O-Professional Services not Elsewhere Specified	\$13,216.38	
COMPANY	658 - Electric Lighting, Distribution and Control Eqpt	\$13,224.00	
	1316 - Roads, Highways and Airport Runways	\$13,256,004.04	
	493 - Interpretation Services	\$13,288.80	
INC.	656 - Heating, Air-Conditioning and Refrigeration Eqpt	\$13,332.50	
	460 - Protection Services	\$13,338.97	
TD.	852 - Real Estate Services	\$13,359.55	
	3259 - Miscellaneous Expenditures not		

As you can see, \$13,256,004.04 appears in the middle of values of about \$13,000, because the sort is done as if the numbers were words.

Perhaps of even greater concern, you can't do math with numbers entered as text.

There are other, more subtle, problems you can face. For example, you might find supposedly unique identifiers, or even entire rows, duplicated. Or you may find a unique person or business has more than one unique identifier.

Fixing the problems

These are pretty typical problems that you may face with data, though the set of examples here is far from exhaustive. Let's turn to some methods we can use to take this dirty data, and run it through the laundry. For each problem, we're going to show you one or more ways you can fix it.

In this Excel sheet of government contracts, some of the cells are misaligned with the columns. If you'd like to practice, you can download the sheet [contracts.xlsx](#) from the companion website.

TYr

	A	B	C	D	E
1	Contract Date	Vendor Name	Description of Work	Contract Value	
2	2005-04-01	OTIS CANADA INC	859 - Other Business Services not Elsewhere Specified	\$17,027,229.55	
3	2005-04-08	THYSSENKRUPP ELEVATOR (CANADA) LTD.	639 - Institutional Buildings	\$7,016,872.41	
4	2008-04-22		THE ARCOP GROUP / GERSOVITZ MOSS	423 - Engineering Consultants - Other	\$76,300,367.96
5	2008-04-22	PRIESTMAN NEILSON & ASSOCIATES LTD	423 - Engineering Consultants - Other	\$627,462.51	
6	2009-04-01	OTIS CANADA INC		859 - Other Business Services not Elsewhere Specified	\$518,671.21
7	2009-04-01	OTIS CANADA INC	859 - Other Business Services not Elsewhere Specified	\$90,212.78	
8	2009-04-01	OTIS CANADA INC	859 - Other Business Services not Elsewhere Specified	\$410,544.95	
9	2009-04-01	OTIS CANADA INC		859 - Other Business Services not Elsewhere Specified	\$273,195.39
			859 - Other Business Services not		

If you tried to sort or summarize the data, for example creating an Excel pivot table of total contract values by Description of Work, you'd get incomplete or illogical results with so dozens of work description entries in the contract value field, and the corresponding contract values off beyond the table to the right. Fortunately, the fix is simple. We'll use Excel to sort the table to bring all of the misaligned cells together, then shift data to the left to fill blank cells.

To begin, the first thing you should do is add a new column A with a sequential ID number. To do that, insert a new column by highlighting column A, right clicking (Command click on a Mac) and choosing Insert from the context-sensitive menu. Once you have added the new column, manually type 1 and 2 into cells A2 and A3, and then fill down to the bottom of the sheet by highlighting cells A2 and A3, then placing the mouse in the bottom right hand corner of A3 and either double clicking, or dragging down.

A	B	C
ID	Contract Date	Vendor Name
1	2005-04-01	OTIS CANADA INC
2	2005-04-08	THYSSENKRUPP ELEVATOR (CAI
	2008-04-22	
	2008-04-22	PRIESTMAN NEILSON & ASSOCI
	2009-04-01	
		OTIS CANADA INC
	2009-04-01	OTIS CANADA INC
7	2009-04-01	OTIS CANADA INC

This will fill the sequence to the bottom.

	A	B	
1	ID	Contract Date	Vendor Name
2026	2025	2016-03-27	ANDREW C. DA
2027	2026	2016-03-27	
2028	2027	2016-03-28	JEAN MARCHA
2029	2028	2016-03-28	
2030	2029	2016-03-29	ARI financial
2031	2030	2016-03-30	Rogers Media I
2032	2031	2016-03-30	
2033	2032	2016-03-30	MDELCC
2034	2033	2016-03-30	
2035	2034	2016-03-31	Axxys 3469051
2036	2035	2016-04-04	OPEN TEXT COI
2037	2036	2016-04-28	SUPRA MAINTÉ
2038			
2039			

This step is important as it will allow us to reset the sheet to its original order, if we need to do so, once we have cleaned up the misaligned cells.

The next step is to sort the column with the blank cells, which is column C. Place your mouse anywhere in the column, then right click (command click on a Mac) and choose Sort on the Data ribbon (data menu on older versions of Excel for Mac).

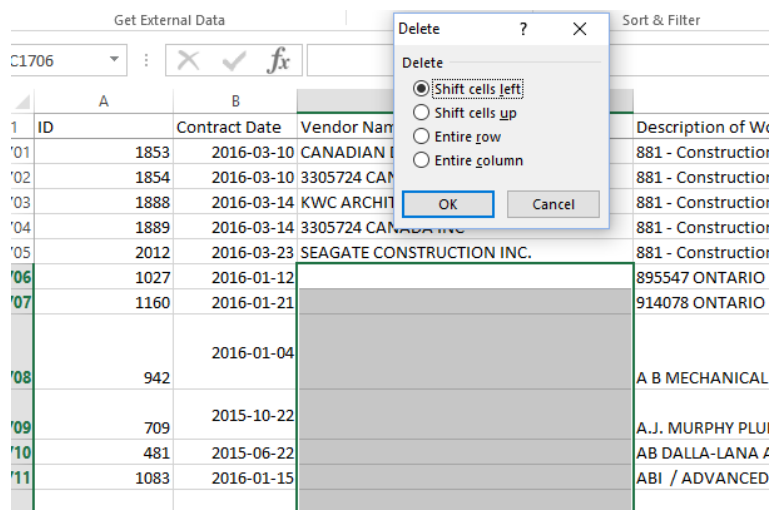
The screenshot shows the Microsoft Excel interface with the 'DATA' ribbon selected. The 'Sort' button is highlighted, and a context menu is open over cell C2. The context menu shows 'Sort A to Z' and 'Lowest to highest' options. The spreadsheet data is visible below the ribbon, showing columns A, B, and C with various contract dates and vendor names.

	A	B	C
1	ID	Contract Date	Vendor Name
2	1	2005-04-01	OTIS CANADA INC
3	2	2005-04-08	THYSSENKRUPP ELEVATOR (CANADA) LTD.
4	3	2008-04-22	
5	4	2008-04-22	PRIESTMAN NEILSON & ASSOCIATES LTD
6	5	2009-04-01	OTIS CANADA INC
7	6	2009-04-01	OTIS CANADA INC

This will sort the column in ascending order. You could also choose descending order. The goal is to sort the sheet so all of the blank cells end up either at the top or bottom of the sheet.

1	ID	Contract Date	Vendor Name	Description o
1701	1853	2016-03-10	CANADIAN DEMOLITION LIMITED	881 - Constru
1702	1854	2016-03-10	3305724 CANADA INC	881 - Constru
1703	1888	2016-03-14	KWC ARCHITECTS INC.	881 - Constru
1704	1889	2016-03-14	3305724 CANADA INC	881 - Constru
1705	2012	2016-03-23	SEAGATE CONSTRUCTION INC.	881 - Constru
1706	1027	2016-01-12		895547 ONTA
1707	1160	2016-01-21		914078 ONTA
1708	942	2016-01-04		A B MECHANI
1709	709	2015-10-22		A.J. MURPHY
1710	481	2015-06-22		AB DALLA-LAI
1711	1083	2016-01-15		ABI / ADVAN
1712	1253	2016-01-26		ABI / ADVAN
1713	1806	2016-03-08		ABI / ADVAN
1714	1934	2016-03-17		ABI / ADVAN4

With the blank cells highlighted, right click on the highlighted cells, and choose Delete. Excel will ask if you want to shift the cells to the left, down, right or up. Choose left to shift everything to the right toward the left, realigning the misaligned cells with the correct columns.



Now, re-sort the sheet by your ID column, and the sheet will return to its original order, with all the misaligned cells corrected.

	A	B	C	D
1	ID	Contract Date	Vendor Name	Description of Work
2	1	2005-04-01	OTIS CANADA INC	859 - Other Business Service Elsewhere Specified
3	2	2005-04-08	THYSSENKRUPP ELEVATOR (CANADA) LTD.	639 - Institutional Buildings
4	3	2008-04-22	THE ARCOP GROUP / GERSOVITZ MOSS	423 - Engineering Consultar
5	4	2008-04-22	PRIESTMAN NEILSON & ASSOCIATES LTD	423 - Engineering Consultar
6	5	2009-04-01	OTIS CANADA INC	859 - Other Business Service Elsewhere Specified
7	6	2009-04-01	OTIS CANADA INC	859 - Other Business Service Elsewhere Specified
8	7	2009-04-01	OTIS CANADA INC	859 - Other Business Service Elsewhere Specified
9	8	2009-04-01	OTIS CANADA INC	859 - Other Business Service Elsewhere Specified
10	9	2009-04-01	OTIS CANADA INC	859 - Other Business Service Elsewhere Specified
11	10	2009-04-01	OTIS CANADA INC	859 - Other Business Service Elsewhere Specified
12	11	2009-04-01	OTIS CANADA INC	859 - Other Business Service Elsewhere Specified

Fixing dates in odd text formats.

You'll recall when we looked at the credit card data, that the dates in the table were entered as text, such as 20150101, in year, month, day order. While Excel is forgiving and will try to convert numbers stored as text to actual numbers before doing math, in this case it will treat the numbers as numbers in the millions, rather than as dates, producing the wrong answers. Database programs will generate errors if you try to do any math on text.

We can use some Excel string functions, to extract out the elements of the dates, then reassemble them in a layout Excel recognizes as a date.

To begin, let's insert a new column to the right of the Stmt Date column, and format it as a date column in year, month, day format (e.g. 2016-10-30).

Next, we'll write a formula using string functions in the first data cell in the new column:

	C	D
	Stmt Date	Stmt Date
1	20130301	=LEFT(C2,4)&"-"&MID(C2,5,2)&"-"&RIGHT(C2,2)
2	20130301	MID(text, start_num, num_chars)
3	20130301	
4	20130301	
5	20130301	
6	20130301	

The formula uses the LEFT function to grab the first four characters of the date, joins or concatenates that with a hyphen, then uses the MID function to grab the fifth and sixth characters of the date, and again concatenates that with a hyphen, and finally uses the RIGHT function to grab the last two characters, representing the day. The & character is the concatenation operator in Excel. You can also write this, if you prefer, as:

Stmt Date	Stmt Date	T
20130301	=CONCAT(LEFT(C2,4),"-",MID(C2,5,2),"-",RIGHT(C2,2))	2
20130301	CONCAT(text1, [text2], [text3], [text4], [text5], ...)	2
20130301		2
20130301		2
20130301		2
20130301		2

The result is the same.

Now, we'll copy the formula down the column and go through the same steps for the Trans Date column.

	A	B	C	D	E	F
1	ID	Line #	Stmt Date	Stmt Date	Trans Date	Trans Date
2	1	7	20140203	2014-02-03	20140108	2014-01-08
3	2	8	20140203	2014-02-03	20140112	2014-01-12
4	3	9	20140203	2014-02-03	20140114	2014-01-14
5	4	10	20140203	2014-02-03	20140114	2014-01-14
6	5	11	20140203	2014-02-03	20140114	2014-01-14
7	6	12	20140203	2014-02-03	20140123	2014-01-23
8	7	18	20140203	2014-02-03	20131129	2013-11-29
9	8	19	20140203	2014-02-03	20140104	2014-01-04
10	9	20	20140203	2014-02-03	20140106	2014-01-06
11	10	21	20140203	2014-02-03	20140113	2014-01-13
12	11	27	20140203	2014-02-03	20140108	2014-01-08
13	12	62	20140203	2014-02-03	20140108	2014-01-08
14	13	63	20140203	2014-02-03	20140108	2014-01-08

We now have properly formatted dates, though in one of its quirks, Excel is still displaying the new columns aligned to the left, as if they were text. A quick bit of math, however, shows they are actually dates. If you really want to align them to the right, you can use the Align Right icon in the Alignment area of the Excel home ribbon.

This work could easily be done in a database program, using string functions to update a new Date format column with the reorganized contents of the original date column.

Another common problem in Excel is having a number column that is formatted as text. If you are lucky, correcting this will be a simple matter of changing the number format of the column to a number type, such as currency, or Number. But sometimes this has no effect, and the column remains stubbornly text despite your best efforts. Here, we see the same data we used in the last example. We've tried to sort on the contract value column, but it sorts in alphabetical order.

ID	Contract Date	Vendor Name	Description of Work	Contract Value
1505	2016-02-12	FAROS CONSTRUCTION LIMITED	630 - Office Buildings	\$10000
1712	2016-03-01	GENTARA REAL ESTATE L.P.	601 - Marine Installations	\$10000.47
1822	2016-03-08	ATLANTICA MECHANICAL SERVICES	630 - Office Buildings	\$10010.52
115	2013-04-01	XEROX CANADA LTD.	533 - Rentl Machinery, Off Furnitre / Fixtres & Other Eqpt	\$10053.59
2010	2016-03-23	FLAGSHIP CONSTRUCTION LIMITED	639 - Institutional Buildings	\$1007803.05
1121	2016-01-19	STEWART WEIR MACDONALD LTD	852 - Real Estate Services	\$10080
1664	2016-02-26	TRADUCTEURS REUNIS S.C.C. (LES)	494 - Transl. Serv. - Operating Expenses	\$101178
910	2015-12-29	CANADA LANDS COMPANY CLC LIMITED	881 - Construction Services	\$10118.73
1335	2016-02-01	CBCL LIMITED	422 - Engineering Consultants - Construction	\$10146
1423	2016-02-06	4 OFFICE AUTOMATION LTD.	671 - Other Office Equipment	\$10170
1027	2016-01-12	895547 ONTARIO LTD.	630 - Office Buildings	\$101922.3

One trick that will often work to solve this problem is to multiply the reluctant column by 1.

First, change the cell formats to Number, currency or whatever type you want. To do this, highlight the whole column, right click (or Command click on a Mac), then choose Format Cells from the context-sensitive menu. In the Number tab of the Format Cells dialogue, pick the number format you want.

Now, in a new column to the right, in the cell to the right of the first number, enter a formula to multiply the cell by 1.

	E	F	G	H
Contract Value	Value2			
\$10000	=E2*1			
\$10000.47				
\$10010.52				
\$10053.59				
\$1007803.05				
\$10080				

More often than not, this will cure the problem, as we see here.

	E	F	G
Contract Value	Value2		
\$10000	\$10,000.00		
\$10000.47			
\$10010.52			
\$10053.59			

Now, fill the formula to the bottom of the sheet, and you have numbers where before you had text.

D	E	F
Description of Work	Contract Value	Value2
630 - Office Buildings	\$10000	\$10,000.00
601 - Marine Installations	\$10000.47	\$10,000.47
630 - Office Buildings	\$10010.52	\$10,010.52
533 - Rentl Machinry, Off Furnitre / Fixtres & Other Eqpt	\$10053.59	\$10,053.59
639 - Institutional Buildings	\$1007803.05	\$1,007,803.05
852 - Real Estate Services	\$10080	\$10,080.00
494 - Transl. Serv. - Operating Expenses	\$101178	\$101,178.00
881 - Construction Services	\$10118.73	\$10,118.73
422 - Engineering Consultants - Construction	\$10146	\$10,146.00

Fixing inconsistent values

Probably the most time-consuming task in data cleaning is eliminating inconsistencies in spelling within fields in a data table. Poor data entry protocols will often result in different people entering the same information in different ways. You might see “California” also entered as “CA” or “Calif.”. In a database or spreadsheet program, the three values will be seen as completely different entities, even though we as users know from experience that the three refer to the same place.

In a small dataset, it’s realistic to go through and correct the entries by hand. But as soon as you have many hundreds, thousands, hundreds of thousands or even millions of rows to fix, you need something a little more powerful. This kind of cleaning can be accomplished in different ways, and you can either use a spreadsheet or database program, or a special tool called Open Refine, originally developed by Google, but now an open source project.

For our example data, we’ll use data on use of credit cards issued to employees of Halifax Regional Municipality in Halifax, Nova Scotia. You can download the data from the companion website.

There are many spelling issues in the vendor field. For example, for Air Canada, the ticket number is always listed, and Air Canada is spelled two ways, both of which would make it difficult to sum up the total spent with Air Canada.

E	F	G
Trans Date	Trans Date	Merchant Name
20130815	2013-08-15	AIR CAN 0143971138892
20130815	2013-08-15	AIR CAN 0143971138894
20130904	2013-09-04	AIR CAN 0143971357006
20131002	2013-10-02	AIR CAN 0143971666128
20131002	2013-10-02	AIR CAN 0143971666132
20131003	2013-10-03	AIR CAN 0143971666163
20131019	2013-10-19	AIR CAN 0143971666163
20131023	2013-10-23	AIR CAN 0143971766587
20140113	2014-01-13	AIR CAN 0143972384093
20140118	2014-01-18	AIR CAN 0143972384094
20140325	2014-03-25	AIR CANADA 0140851268724
20140221	2014-02-21	AIR CANADA 0140851305555
20140109	2014-01-09	AIR CANADA 0142129429074
20140110	2014-01-10	AIR CANADA 0142129447254
20140110	2014-01-10	AIR CANADA 0142129505068
20140113	2014-01-13	AIR CANADA 0142129540454
20140114	2014-01-14	AIR CANADA 0142129604745
20140117	2014-01-17	AIR CANADA 0142129604745
20140114	2014-01-14	AIR CANADA 0142129607919

We'll look at three ways of fixing the data, using Excel, using a database program, and using Open Refine. We'll start with Excel, our familiar Swiss Army Knife from Chapter 4.

As a first step, let's add a new column to our worksheet, directly to the right of the existing Merchant Name field. We can call it Clean_Merchant. Next, sort the original Merchant Name field in ascending order. Your sheet should end up something like this:

	G	H	I
	Merchant Name	Clean_Merchant	Comp Code
.3-08-21	#0514 220 COBEQUID RD		HROP
.3-02-06	#0705 1830 ST MARGARE		HROP
.3-04-18	#0724 5450 INGLISH ST		HROP
.3-04-18	#0724 5450 INGLISH ST		HROP
.3-08-23	#1983 220 COBEQUID RD		HROP
.3-07-11	#1995 280 LACEWOOD DR		HROP
.3-09-04	#2344 817 SACKVILLE		HROP
.3-09-04	#2344 817 SACKVILLE		HROP
.3-07-22	#2575 575 MAIN		HROP
.4-01-08	#274 SPORT CHEK COMBO		HROP
.4-01-06	#274 SPORT CHEK COMBO		HROP

As an example, let's scroll down to the entries for Air Canada. In the new, blank column, enter AIR CANADA in the cell directly to the right of the first Air Canada entry, AIR CAN 0140851268724.

8	AIM*OLD HOUSE JRNL SUB		LIBY
1	AINO AB		HROP
1	AINS INC		HROP
0	AIR & SPACE MAG		LIBY
8	AIR CAN 0140851268724	AIR CANADA	HROP
5	AIR CAN 0142117418907		HROP
6	AIR CAN 0142117458306		HROP
6	AIR CAN 0142117466253		HROP
1	AIR CAN 0142117679529		HROP
3	AIR CAN 0142117758918		HROP
3	AIR CAN 0142117758919		HROP
3	AIR CAN 0142117759996		HROP

Now, copy the new AIR CANADA entry down until you reach the last Air Canada entry in the original Merchant Name column.

	G	H	I	
	Merchant Name	Clean_Merchant	Comp Code	Cost
4	AIR CANADA 0144623404316	AIR CANADA	HROP	
0	AIR CANADA 0144623404316	AIR CANADA	HROP	A8
4	AIR CANADA 0144623404317	AIR CANADA	HROP	
3	AIR CANADA 0144623688345	AIR CANADA	HROP	
3	AIR CANADA 0144623688410	AIR CANADA	HROP	
3	AIR CANADA 0144623848514	AIR CANADA	HROP	
5	AIR CANADA 0144624000274	AIR CANADA	HROP	F1
7	AIR CANADA 0144624000274	AIR CANADA	HROP	
2	AIR CANADA 0144624196285	AIR CANADA	HROP	
2	AIR CANADA 0145865850672	AIR CANADA	HROP	
2	AIR CANADA 0145865850673	AIR CANADA	HROP	
4	AIR CANADA 0372342062601	AIR CANADA	HROP	R6
4	AIR CANADA 0372342062614	AIR CANADA	HROP	R6
7	AIR CANADA ON BOARD CA	AIR CANADA	HROP	
2	AIR LIQUIDE CANADA INC		HROP	R7
4	AIR LIQUIDE CANADA INC		HROP	W
3	AIR LIQUIDE CANADA INC		HROP	R7
2	AIR LIQUIDE CANADA INC		HROP	R7

The ability of Excel to quickly fill down values makes this an effective method for cleaning worksheets with many inconsistencies of the same name. You only need to enter cleaned entries for those names for which there are inconsistent spellings. To add the remaining entries to the new column, sort the sheet by the new column so all the blank cells in it are at the top or bottom, then highlight and copy all of the remaining original entries and paste them into the remaining blank cells in the new column.

Cleaning using SQL

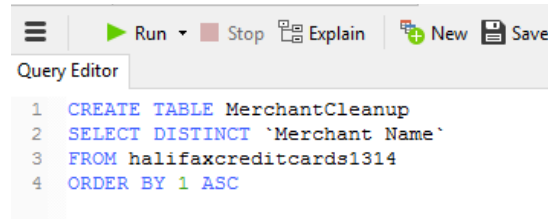
You can also do bulk data cleaning using SQL in a database program. If you have only a limited number of entries to clean, you can follow essentially the same process outlined above for Excel, adding a new column and then populating it with new values. The easiest way to do that is to write an UPDATE query to update the new column WHERE certain values exist in the original column. For example:

```
UPDATE HfxCreditCards
SET Clean_Merchant = "AIR CANADA"
WHERE Merchant_Name LIKE 'AIR CAN%'
```

For a refresher on SQL syntax for UPDATE queries, see Chapter 5.

If you have a lot of entries to fix, it can be more efficient to create an intermediary, lookup table containing all of the unique entries in the original column, add a new blank column to the intermediary table and clean it up, and finally joining the intermediary table back to the main table using the values in the original field for the join, updating the original table.

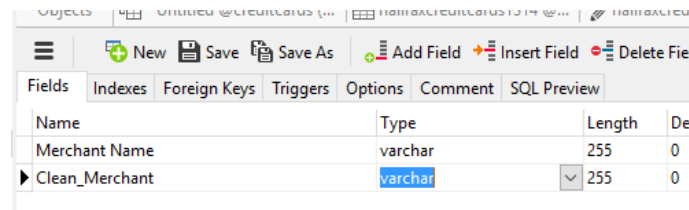
Let's walk through the process with our credit card data. First we create the intermediary table. Here is the query, shown in the Navicat front end for MySQL:



```
1 CREATE TABLE MerchantCleanup
2 SELECT DISTINCT `Merchant Name`
3 FROM halifaxcreditcards1314
4 ORDER BY 1 ASC
```

Remember that if you are running this query in Access, the SQL is slightly different. See Chapter 5 of *The Data Journalist* for a refresher.

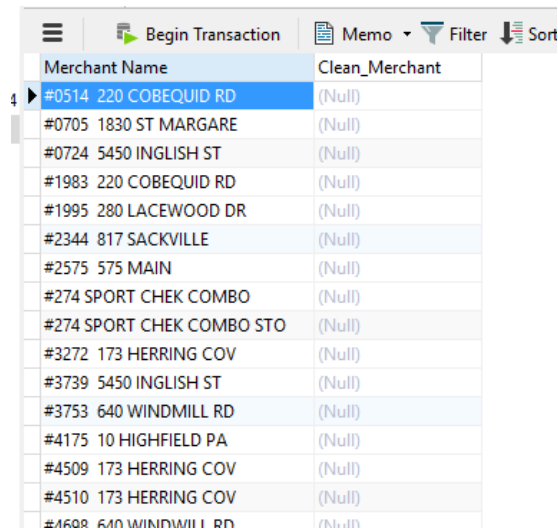
The next step is to modify the new table to add a new column for the clean entries. Here we are adding the new field in Navicat:



Name	Type	Length	De
Merchant Name	varchar	255	0
Clean_Merchant	varchar	255	0

If you are unsure how to modify a table, see the tutorials on creating tables and adding data, also available on the companion website for *The Data Journalist*.

The new table now has two fields:



Merchant Name	Clean_Merchant
#0514 220 COBEQUID RD	(Null)
#0705 1830 ST MARGARE	(Null)
#0724 5450 INGLISH ST	(Null)
#1983 220 COBEQUID RD	(Null)
#1995 280 LACEWOOD DR	(Null)
#2344 817 SACKVILLE	(Null)
#2575 575 MAIN	(Null)
#274 SPORT CHEK COMBO	(Null)
#274 SPORT CHEK COMBO STO	(Null)
#3272 173 HERRING COV	(Null)
#3739 5450 INGLISH ST	(Null)
#3753 640 WINDMILL RD	(Null)
#4175 10 HIGHFIELD PA	(Null)
#4509 173 HERRING COV	(Null)
#4510 173 HERRING COV	(Null)
#4608 640 WINDMILL RD	(Null)

Now, we can begin to update the values in the Clean_Merchant field. We could do it manually, but that could take a long time. By using SQL UPDATE queries, we can update many rows simultaneously, and by making only small changes to the query each time, we can move the process along quickly.

For example, to update the almost 500 AIR CAN and AIR CANADA entries, we can use this query, as seen in Navicat:

```

1 UPDATE merchantcleanup
2 SET Clean_Merchant = "AIR CANADA"
3 WHERE `Merchant Name` LIKE "AIR CAN%"
4

```

By using the LIKE operator, we capture all of the entries that begin with AIR CAN, with this result:

Merchant Name	Clean_Merchant
AIR CAN 0143970844563	AIR CANADA
AIR CAN 0143970844564	AIR CANADA
AIR CAN 0143970844582	AIR CANADA
AIR CAN 0143971138892	AIR CANADA
AIR CAN 0143971138894	AIR CANADA
AIR CAN 0143971357006	AIR CANADA
AIR CAN 0143971666128	AIR CANADA
AIR CAN 0143971666132	AIR CANADA
AIR CAN 0143971666163	AIR CANADA
AIR CAN 0143971766587	AIR CANADA
AIR CAN 0143972384093	AIR CANADA
AIR CAN 0143972384094	AIR CANADA
AIR CANADA 0140851268724	AIR CANADA
AIR CANADA 0140851305555	AIR CANADA
AIR CANADA 0142129429074	AIR CANADA
AIR CANADA 0142129447254	AIR CANADA
AIR CANADA 0142129505068	AIR CANADA
AIR CANADA 0142129540454	AIR CANADA
AIR CANADA 0142129604745	AIR CANADA
AIR CANADA 01421296697010	AIR CANADA

Similarly, we could update the six entries for Best Buy, seen in this image:

BENTLEY #83097	(Null)
BERGMAN CONCRETE	(Null)
BEST BUY #912	(Null)
BEST BUY #979	(Null)
BEST BUY.CA # 899	(Null)
BEST BUY.CA # 900	(Null)
BEST WESTERN CHOCOLATE LA	(Null)
BEST WESTERN DARTMOUTH	(Null)
BEST WESTERN NEWARK AI	(Null)
BEST WESTERN PLUS DART	(Null)
BEST WESTERN RENFREW I	(Null)
BESTBUY.CA #898	(Null)
BESTBUYCANADA.CA #898	(Null)
BETTER BUY SPORTS	(Null)
BETTER SAFE THAN SORRY	(Null)
BEVLO PRODUCTS INCORPORAT	(Null)
BEY'S FAST FUEL # 4511	(Null)
BF*MUSICIANSFRIEND	(Null)
BIG BELLY SOLAR INC	(Null)

With this query:

```
Run Stop Explain New Save Save
Query Editor
1 UPDATE merchantcleanup
2 SET Clean_Merchant = "BEST BUY"
3 WHERE `Merchant Name` LIKE "BEST%BUY%"
4
```

By using the % wildcard between BEST and BUY we capture all variations of Best Buy, those with and without a space between the two words. Remember that Microsoft Access SQL in its default mode uses the * wildcard instead of the % wildcard.

Merchant Name	Clean_Merchant
BERGMAN CONCRETE	(Null)
BEST BUY #912	BEST BUY
BEST BUY #979	BEST BUY
BEST BUY.CA # 899	BEST BUY
BEST BUY.CA # 900	BEST BUY
BEST WESTERN CHOCOLATE LA	(Null)
BEST WESTERN DARTMOUTH	(Null)
BEST WESTERN NEWARK AI	(Null)
BEST WESTERN PLUS DART	(Null)
BEST WESTERN RENFREW I	(Null)
BESTBUY.CA #898	BEST BUY
BESTBUYCANADA.CA #898	BEST BUY
BETTER BUY SPORTS	(Null)
BETTER SAFE THAN CORRV	(Null)

Continue cleaning until you have provided clean versions of all entries for which there was more than one variation. You can leave entries that have only one spelling as is as we will take care of those in the next step.

Once you have finished populating the Clean_Merchant field, you can copy all of the entries for which there was only one spelling to the new column using this syntax:

```
UPDATE merchantcleanup
SET Clean_Merchant = `Merchant Name`
WHERE Clean_Merchant IS NULL
```

This query will populate the Clean_Merchant field with the original value but only when there isn't already a value in Clean_Merchant. The cleanup table is now ready to use.

Merchant Name	Clean_Merchant
BEELER SECURITY SERVICE	BEELER SECURITY
BELL	BELL
BELL ALIANT BAYERS LAK	BELL ALIANT
BELL ALIANT BEDFORD CO	BELL ALIANT
BELL ALIANT BEDFORD HW	BELL ALIANT
BELL ALIANT BEDFORD HWY	BELL ALIANT
BELL ALIANT BURNSIDE B	BELL ALIANT
BELL ALIANT BURNSIDE BROW	BELL ALIANT
BELL ALIANT DARTMOUTH	BELL ALIANT
BELL ALIANT DARTMOUTH SUP	BELL ALIANT
BELL ALIANT MICMAC MAL	BELL ALIANT
BELL ALIANT MICMAC MALL	BELL ALIANT
BELL ALIANT SCOTIA SQU	BELL ALIANT
BELL ALIANT SCOTIA SQUARE	BELL ALIANT
BELL ALIANT SHOPPING C	BELL ALIANT
BELL ALIANT SHOPPING CTR	BELL ALIANT
BELL ALIANT SUPERSTORE	BELL ALIANT
BELL CANADA (OB)	BELL CANADA (OB)
BELL EXPRESSVU	BELL EXPRESSVU
BELL FUND IN SBT RADIO	BELL FUND IN SBT RADIO
BELL FUNDSIN SBT RADIO -	BELL FUND IN SBT RADIO
BELL MOBILITE RADIO	BELL MOBILITY
BELL MOBILITE RADIO (POS)	BELL MOBILITY

The final step is to update the original table. First, add a new column to the original credit card table, and call it Clean_Merchant. It likely makes most sense to put this new column directly to the right of the existing merchant table.

Now, use a SQL UPDATE query to update the new Clean_Merchant column in the original table with the values from the Clean_Merchant field in the cleanup table.

```
UPDATE Halifaxcreditcards1314 a, merchantcleanup b
SET a.Clean_Merchant = b.Clean_Merchant
WHERE a.`Merchant Name` = b.`Merchant Name`
```

The new, clean field can now be used for queries.

Trans Date	Trans Date1	Merchant Name	Clean_Merchant	Comp Code
20130528	2013-05-28	BELL ALIANT SCOTIA SQU	BELL ALIANT	HROP
20130502	2013-05-02	BELL ALIANT SCOTIA SQU	BELL ALIANT	HROP
20130604	2013-06-04	BELL ALIANT SCOTIA SQU	BELL ALIANT	HROP
20130715	2013-07-15	BELL ALIANT SCOTIA SQU	BELL ALIANT	HROP
20130726	2013-07-26	BELL ALIANT SCOTIA SQU	BELL ALIANT	HROP
20130709	2013-07-09	BELL ALIANT SCOTIA SQU	BELL ALIANT	HROP
20130708	2013-07-08	BELL ALIANT SCOTIA SQU	BELL ALIANT	HROP
20130821	2013-08-21	BELL ALIANT SCOTIA SQU	BELL ALIANT	HROP
20130828	2013-08-28	BELL ALIANT SCOTIA SQU	BELL ALIANT	HROP
20130815	2013-08-15	BELL ALIANT SCOTIA SQU	BELL ALIANT	HROP
20130911	2013-09-11	BELL ALIANT SCOTIA SQU	BELL ALIANT	HROP
20131016	2013-10-16	BELL ALIANT SCOTIA SQU	BELL ALIANT	HROP
20140530	2014-05-30	BELL ALIANT SCOTIA SQUA	BELL ALIANT	HROP
20140620	2014-06-20	BELL ALIANT SCOTIA SQUA	BELL ALIANT	HROP
20140801	2014-08-01	BELL ALIANT SCOTIA SQUA	BELL ALIANT	HROP
20141003	2014-10-03	BELL ALIANT SCOTIA SQUA	BELL ALIANT	HROP
20130604	2013-06-04	BELL ALIANT SHOPPING C	BELL ALIANT	HROP
20131116	2013-11-16	BELL ALIANT SHOPPING C	BELL ALIANT	HROP
20131118	2013-11-18	BELL ALIANT SHOPPING C	BELL ALIANT	HROP
20141027	2014-10-27	BELL ALIANT SHOPPING CT	BELL ALIANT	HROP
20130724	2013-07-24	BELL ALIANT SUPERSTORE	BELL ALIANT	HROP
20140813	2014-08-13	BELL CANADA (OB)	BELL CANADA (OB)	HROP
20141027	2014-10-27	BELL CANADA (OB)	BELL CANADA (OB)	HROP

Cleaning with Open Refine

You can download Open Refine from openrefine.org. It requires Java to be installed on your computer (note that Java is not the same as JavaScript).

Open Refine can work with several conventional data formats, including delimited text files, Excel spreadsheet files, JSON, XML and others.

You start Open Refine on a Windows PC by double clicking on the `openrefine.exe` executable file. On a Mac, drag Open Refine into the Applications folder, then double click on it. As long as the required Java version is present on your system, you will see a terminal window open, followed by your default web browser. Open Refine behaves like a web server that runs on your machine. You access it through a browser.

To begin, you need to create a project. For this example, we will open the file `HalifaxCreditCards1314.xlsx`, available for download from the Data Journalist companion site. It's

an Excel file of use of credit cards by staff of Halifax Regional Municipality in 2013 and 2014. Make sure you keep a backup copy of your original data.

Download the file, and then click on Browse to traverse your file system to find the file.

When you click on Next, Open Refine will work for a while, then present you with a screen like this.

ID	Line #	Stmt Date	Stmt Date 2	Trans Date	Trans Date 2	Merchant Name	Comp Code	Cost Ce
1.	1	7 20140203	2014-02-03T00:00:00Z	20140108	2014-01-08T00:00:00Z	WAL-MART #3636	HROP	
2.	2	8 20140203	2014-02-03T00:00:00Z	20140112	2014-01-12T00:00:00Z	WAL-MART #3021	HROP	
3.	3	9 20140203	2014-02-03T00:00:00Z	20140114	2014-01-14T00:00:00Z	PIER 1 IMPORTS #1197	HROP	
4.	4	10 20140203	2014-02-03T00:00:00Z	20140114	2014-01-14T00:00:00Z	GUY'S FRENCHYS	HROP	
5.	5	11 20140203	2014-02-03T00:00:00Z	20140114	2014-01-14T00:00:00Z	WAL-MART #3636	HROP	
6.	6	12 20140203	2014-02-03T00:00:00Z	20140123	2014-01-23T00:00:00Z	ATLANTIC	HROP	

In this dialogue, you get a preview of what your data looks like, as well as some options as to how Open Refine should handle, or parse, the data. In our case, we will leave the options as they are, as Open Refine has correctly identified that it is an Excel file, and we do want the first line to be treated as a column header line. WE also don't want to ignore or discard any rows, or limit the number of rows to be loaded.

When we are certain of the settings, we'll click on Create Project at the top right of the dialogue. It may take a few moments for Open Refine to complete this task. When it's done, you'll see the top 5,10,25 or 50 rows of the data, depending on the option you choose.

HalifaxCreditCards1314.xlsx [Permalink](#) Open... Export Help

o / Redo **52526 rows** Extensions: undefined

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

All	ID	Line #	Stmt Date	Stmt Date 2	Trans Date	Trans Date 2	Merchant Name	Comp Code	Cost Centre	G/L	Order #	
1.	1	7	20140203	2014-02-03T00:00:00Z	20140108	2014-01-08T00:00:00Z	WAL-MART #3636	HROP		0	6404	RP9AMORO
2.	2	8	20140203	2014-02-03T00:00:00Z	20140112	2014-01-12T00:00:00Z	WAL-MART #3021	HROP		0	6404	RP9AMORO
3.	3	9	20140203	2014-02-03T00:00:00Z	20140114	2014-01-14T00:00:00Z	PER 1 MPORTS #1197	HROP		0	6404	RP9AMORO
4.	4	10	20140203	2014-02-03T00:00:00Z	20140114	2014-01-14T00:00:00Z	GUY'S FRENCHYS	HROP		0	6401	RP9AMORO
5.	5	11	20140203	2014-02-03T00:00:00Z	20140114	2014-01-14T00:00:00Z	WAL-MART #3636	HROP		0	6404	RP9AMORO
6.	6	12	20140203	2014-02-03T00:00:00Z	20140123	2014-01-23T00:00:00Z	ATLANTIC SUPERSTORE #3	HROP		0	6404	RP9AMORO
7.	7	18	20140203	2014-02-03T00:00:00Z	20131129	2013-11-29T00:00:00Z	HYPER PROMOTIONS	HROP		0	6399	RP9HXHP0
8.	8	19	20140203	2014-02-03T00:00:00Z	20140104	2014-01-04T00:00:00Z	MICHAELS #3955	HROP		0	6404	RP9RX000
9.	9	20	20140203	2014-02-03T00:00:00Z	20140106	2014-01-06T00:00:00Z	CDN TIRE STORE #00224	HROP		0	6705	RP9RX000
10.	10	21	20140203	2014-02-03T00:00:00Z	20140113	2014-01-13T00:00:00Z	TARGET CANADA T3731	HROP		0	6404	RP9HX001
11.	11	27	20140203	2014-02-03T00:00:00Z	20140108	2014-01-08T00:00:00Z	MICHAELS #3955	HROP		0	6404	RP9FX0K0
12.	12	62	20140203	2014-02-03T00:00:00Z	20140108	2014-01-08T00:00:00Z	COLES 221	HROP		0	6404	RP9AMORO
13.	13	63	20140203	2014-02-03T00:00:00Z	20140108	2014-01-08T00:00:00Z	WAL-MART #3636	HROP		0	6404	RP9AMORO

As we discussed earlier, one of the big issues with this dataset is that the vendor name column contains many variations for the same vendor. If we want to do accurate and/or complete summary calculations, we need to correct that. OpenRefine is perfectly suited for the task.

We'll use the text facet tool, which we open by clicking on the arrow at the top of the column we wish to clean, in this case merchant name, and choosing Text facet, as seen below.

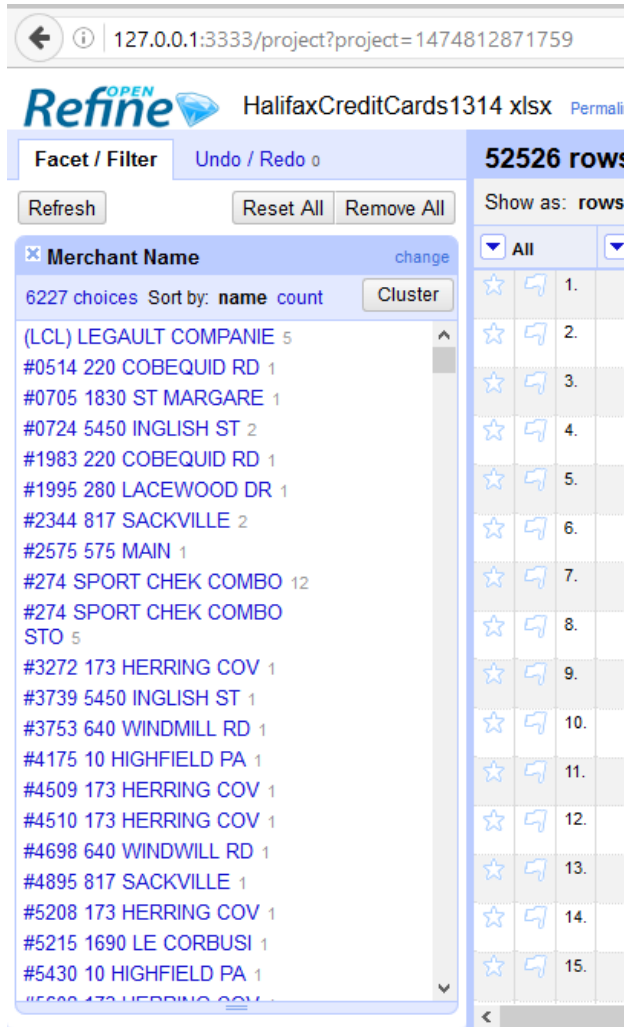
Extensio

« first < previous 1

Date 2	Merchant Name	Comp Code	Cost Centre	G
IT00:00:00Z	Facet	Text facet		64
IT00:00:00Z	Text filter	Numeric facet		64
IT00:00:00Z	Edit cells	Timeline facet		64
IT00:00:00Z	Edit column	Scatterplot facet		64
IT00:00:00Z	Transpose	Custom text facet..		64
IT00:00:00Z	Sort...	Custom Numeric Facet..		64
IT00:00:00Z	View	Customized facets		64
IT00:00:00Z	Reconcile	OP	0	63
IT00:00:00Z	MICHAELS #3955	HROP	0	64
IT00:00:00Z	CDN TIRE STORE #00224	HROP	0	67

You may receive an error message if you have a large number of rows. You can choose to increase the number of distinct items that can be shown, although the more you choose, the slower Open Refine may work.

In our case, we see we have 6227 choices, or distinct values, in that column.



The screenshot shows the Open Refine web interface. At the top, the browser address bar displays '127.0.0.1:3333/project?project=1474812871759'. The Open Refine logo is visible, along with the file name 'HalifaxCreditCards1314.xlsx'. The interface includes a 'Facet / Filter' section with 'Undo / Redo' and 'Permalink' options. A 'Refresh' button and 'Reset All' / 'Remove All' buttons are present. The main area shows a facet for 'Merchant Name' with 6227 choices, sorted by name and count. A 'Cluster' button is available. The right sidebar shows '52526 rows' and 'Show as: rows'. A list of merchant names and their counts is displayed, such as '(LCL) LEGAULT COMPANIE 5', '#0514 220 COBEQUID RD 1', '#0705 1830 ST MARGARE 1', '#0724 5450 INGLISH ST 2', '#1983 220 COBEQUID RD 1', '#1995 280 LACEWOOD DR 1', '#2344 817 SACKVILLE 2', '#2575 575 MAIN 1', '#274 SPORT CHEK COMBO 12', '#274 SPORT CHEK COMBO STO 5', '#3272 173 HERRING COV 1', '#3739 5450 INGLISH ST 1', '#3753 640 WINDMILL RD 1', '#4175 10 HIGHFIELD PA 1', '#4509 173 HERRING COV 1', '#4510 173 HERRING COV 1', '#4698 640 WINDWILL RD 1', '#4895 817 SACKVILLE 1', '#5208 173 HERRING COV 1', '#5215 1690 LE CORBUSI 1', and '#5430 10 HIGHFIELD PA 1'.

What Open Refine does here is it groups the values in the chosen column, in this case Merchant Name. This is exactly the same operation as Excel performs when it creates a pivot table or an SQL database query does when you use GROUP BY or DISTINCT. See chapters 4 and 5 respectively for a discussion of Excel and database programs.

You will notice that Open Refine also provides you with a count of how many times that entry appears in the column.

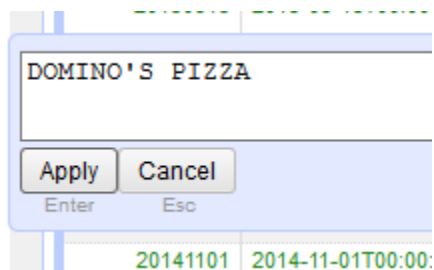
Since our problem is to get rid of multiple variations of the same name, and this is easy with Open Refine. Let's say we wanted to clean up the various entries for Dominos Pizza. Depending on how we wish to proceed, we either need to narrow down to three variations, one for each store number, or just one. This would depend on whether we want to be able to look at each store as a separate entity, or Domino's Pizza as one overall vendor. Let's go with the latter, Domino's as a single entity.



If you hover your mouse over one of the choices, you will see you are given the option to either Edit or Include the entry.



We'll choose "edit." This will open a small window that allows us to edit the entry, so it just reads DOMINO'S PIZZA.



After making the change, and ensuring you have deleted the trailing space following PIZZA, click apply. You'll see the first entry has been changed.

DOMAINSATCOST/DOMAINSA	7	20140114
DOMINO'S PIZZA	1	20140114
DOMINO'S PIZZA #10981	3	20140123
DOMINOS PIZZA 10975	2	20131129
DOMINOS PIZZA 10980	1	20140104
DOMINOS PIZZA 10981	1	20140106
DOMTAR DISTRIBUT. GROUP	1	
DOMTAR DISTRIBUTION GR	48	
DOMTAR DISTRIBUTION		

Follow the same steps for the other four DOMINO's entries, being sure to use the same spelling each time. You'll see what were once five different spellings have been collapsed into one, which you will see now has 8 entries.

DOMAINSATCOST	16
DOMAINSATCOST/DOMAINSA	7
DOMINO'S PIZZA	8
DOMTAR DISTRIBUT. GROUP	1
DOMTAR DISTRIBUTION GR	48
DOMTAR DISTRIBUTION GROUP	15
DOMTAR INC ADIVA	

You can use the same procedure to clean up additional entries.

If you want to see which entries appear the most often, you can choose to sort the list by the count by clicking the "count" link at the top of the list of entries. You may choose to concentrate your early cleaning efforts on entries that appear most often.

The screenshot shows the Refine search interface for HalifaxCreditCards13. The search results are filtered by Merchant Name and sorted by count. The top results are:

Merchant Name	Count
CORPORATE EXPRESS	5530
AMAZON.CA	3567
ACCESS NOVA SCOTIA-RMV	1054
CDN TIRE STORE #00041	700
ADVANTAGE WIRELESS	591
WORK AUTHORITY OM	504
INDIGO ONLINE	489
PRINCESS AUTO	485
CDN TIRE STORE #00044	469

You can also clean your data using Open Refine's cluster tool, which will automatically cluster variations that appear similar to one another. This can be an enormous time saver. Begin by clicking on the Cluster button, as seen in the previous illustration, to bring up the clustering dialogue.

Cluster & Edit column "Merchant Name"

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 "GÃ¶del" and "Godel" probably refer to the same person. [Find out more ...](#)

Method: Keying Function: 251 clusters found

Cluster Size	Row Count	Values in Cluster	Merge?	New Cell Value
3	81	<ul style="list-style-type: none"> MARITIME LAWN GARDEN (34 rows) MARITIME LAWN & GARDEN (30 rows) MARITIME LAWN GARDEN (17 rows) 	<input type="checkbox"/>	MARITIME LAWN GARDEN
3	83	<ul style="list-style-type: none"> SOBEYS #622 QPS (44 rows) SOBEYS 622 QPS (38 rows) SOBEYS #622 QPS (1 rows) 	<input type="checkbox"/>	SOBEYS #622 QPS
3	16	<ul style="list-style-type: none"> PAYPAL (10 rows) PAYPAL *PAYPAL (5 rows) PAYPAL PAYPAL (1 rows) 	<input type="checkbox"/>	PAYPAL
3	6	<ul style="list-style-type: none"> LINDT SPRUNGLI (3 rows) LINDT SPRUNGLI (2 rows) LINDT & SPRUNGLI (1 rows) 	<input type="checkbox"/>	LINDT SPRUNGLI
2	16	<ul style="list-style-type: none"> TIM HORTONS #1107# QTH (9 rows) TIM HORTONS 1107 QTH (7 rows) 	<input type="checkbox"/>	TIM HORTONS #1107# QTH
2	29	<ul style="list-style-type: none"> ARTS TROPHY HOUSE (17 rows) 	<input type="checkbox"/>	ARTS TROPHY HOUSE

Choices in Cluster: 2 — 3

Rows in Cluster: 0 — 350

Average Length of Choices: 6 — 25

Length Variance of Choices: 0 — 3.7800000000000002

Open Refine automatically groups similar-looking entries into clusters that you can rename by entering a new, common, value in the New Cell Value box.

Before we do that, though, let's take note of the two dropdown text boxes at the top of the screen, labeled "Method" and "Keying Function." Generally speaking, you should probably stick to the Key Collision method, because the alternative, Nearest Neighbor, requires a great deal of math, and can be very slow (see <https://github.com/OpenRefine/OpenRefine/wiki/Clustering-In-Depth> for an in-depth discussion of this issue). However, it is definitely worth trying out the different keying functions for the Key Collision method. They can produce markedly different results. As an example, switching to cologne phonetic as the keying method is much more effective in collecting together the dozens of entries for Air Canada and Air Can—each one has a different ticket number.

Cluster & Edit column "Merchant Name"

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 "GÃ¶del" and "Godel" probably refer to the same person.

Method: Keying Function:

Cluster Size	Row Count	Values in Cluster	Merge?	New Cell Value
250	289	<ul style="list-style-type: none"> AIR CANADA 0144623151776 (5 rows) AIR CANADA 0142137894777 (4 rows) AIR CANADA 0142141274835 (3 rows) AIR CANADA 0142129604745 (2 rows) AIR CANADA 0142130524159 (2 rows) AIR CANADA 0142131514494 (2 rows) AIR CANADA 0142131514713 (2 rows) AIR CANADA 0142131801224 (2 rows) AIR CANADA 0142131805382 (2 rows) AIR CANADA 0142131891633 (2 rows) AIR CANADA 0142131892020 (2 rows) AIR CANADA 0142132018763 (2 rows) AIR CANADA 0142132446497 (2 rows) AIR CANADA 0142132541006 (2 rows) AIR CANADA 0142133888335 (2 rows) AIR CANADA 0142133888498 (2 rows) AIR CANADA 0142133945847 (2 rows) 	<input type="checkbox"/>	AIR CANADA 0144623151776

Method key collision Keying Function cologne-phonetic

		<ul style="list-style-type: none"> AIR CANADA 0372342062601 (1 rows) AIR CANADA 0372342062614 (1 rows) 		
164	200	<ul style="list-style-type: none"> AIR CAN 0142118322894 (4 rows) AIR CAN 0142119937690 (4 rows) AIR CAN 0142524787081 (3 rows) AIR CAN 0142118616432 (2 rows) AIR CAN 0142118929666 (2 rows) AIR CAN 0142118968501 (2 rows) AIR CAN 0142118983249 (2 rows) AIR CAN 0142119742018 (2 rows) AIR CAN 0142119876169 (2 rows) AIR CAN 0142120012046 (2 rows) AIR CAN 0142120030061 (2 rows) AIR CAN 0142120117281 (2 rows) 	<input type="checkbox"/>	AIR CAN 0142118322894

To consolidate all of these under a single entry, we put the new entry we want into the text box on the right, making sure to enter the same thing, AIR CANADA, for both the AIR CANADA and AIR CAN entries, click the check boxes for each cluster, and then at the bottom of the dialogue click on Merge Selected and Re Cluster (or Merge Selected and Close, if you are done). Here is the dialogue just before we take that final step.

Cluster & Edit column "Merchant Name"

This feature helps you find groups of different cell values that might be alternative representations of the same thing. For example, "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.

Method key collision Keying Function cologne-phonetic

		<ul style="list-style-type: none"> AIR CANADA 0372342062601 (1 rows) AIR CANADA 0372342062614 (1 rows) 			
164	200	<ul style="list-style-type: none"> AIR CAN 0142118322894 (4 rows) AIR CAN 0142119937690 (4 rows) AIR CAN 0142524787081 (3 rows) AIR CAN 0142118616432 (2 rows) AIR CAN 0142118929666 (2 rows) AIR CAN 0142118968501 (2 rows) AIR CAN 0142118983249 (2 rows) AIR CAN 0142119742018 (2 rows) AIR CAN 0142119876169 (2 rows) AIR CAN 0142120012046 (2 rows) AIR CAN 0142120030061 (2 rows) AIR CAN 0142120117281 (2 rows) AIR CAN 0142120117282 (2 rows) AIR CAN 0142120142990 (2 rows) AIR CAN 0142120143124 (2 rows) AIR CAN 0142120297504 (2 rows) AIR CAN 0142120506752 (2 rows) AIR CAN 0142120509233 (2 rows) AIR CAN 0142120509391 (2 rows) AIR CAN 0142121502000 (2 rows) AIR CAN 0142121587062 (2 rows) AIR CAN 0142121885607 (2 rows) AIR CAN 0142122274348 (2 rows) AIR CAN 0142123650006 (2 rows) AIR CAN 0142125425753 (2 rows) 	<input checked="" type="checkbox"/>	AIR CANADA	# Chc # Row Avera Leng

In one step we can clean up almost 500 purchases with Air Canada. If we close the clustering dialogue, and select AIR CANADA in the facet list, we can see that all of the entries have a single value under Merchant Name, making proper analysis possible.

The screenshot shows the Refine tool interface. At the top, it says "Refine OPEN HalifaxCreditCards1314.xlsx Permalink". Below that, there are buttons for "Facet / Filter", "Undo / Redo", "Refresh", "Reset All", and "Remove All". A facet for "Merchant Name" is expanded, showing 5810 choices. "AIR CANADA" is selected with 489 items. The main table shows 489 matching rows (52526 total). The table has columns: Trans Date, Trans Date 2, Merchant Name, and Comp Code. All Merchant Name entries are "AIR CANADA".

Trans Date	Trans Date 2	Merchant Name	Comp Code
20140110	2014-01-10T00:00:00Z	AIR CANADA	HROP
20140110	2014-01-10T00:00:00Z	AIR CANADA	HROP
20140113	2014-01-13T00:00:00Z	AIR CANADA	HROP
20140118	2014-01-18T00:00:00Z	AIR CANADA	HROP
20140205	2014-02-05T00:00:00Z	AIR CANADA	HROP
20140414	2014-04-14T00:00:00Z	AIR CANADA	HROP
20130405	2013-04-05T00:00:00Z	AIR CANADA	HROP
20130409	2013-04-09T00:00:00Z	AIR CANADA	HROP
20130410	2013-04-10T00:00:00Z	AIR CANADA	HROP
20130410	2013-04-10T00:00:00Z	AIR CANADA	HROP

It is important to remember that everything we have been doing we have been doing inside the browser. The original data file has not been altered. As well, if our computer crashes, or our Browser stops responding or closes, we will lose our work.

To generate a permanent copy of our cleaned data, we can either export a copy of the project, which we can reopen later, or we can export as an Excel file, a comma delimited file, an HTML file, or an OpenOffice format spreadsheet. To do this, click on the Export button at the top right of the screen, and choose one of the options.

The screenshot shows the Refine tool interface with the "Export" dropdown menu open. The menu options are: "Export project", "Tab-separated value", "Comma-separated value", "HTML table", "Excel", "ODF spreadsheet", "Triple loader", "MQLWrite", "Custom tabular exporter...", and "Templating...". The background shows a table with columns "Merchant Name" and "Comp Code".

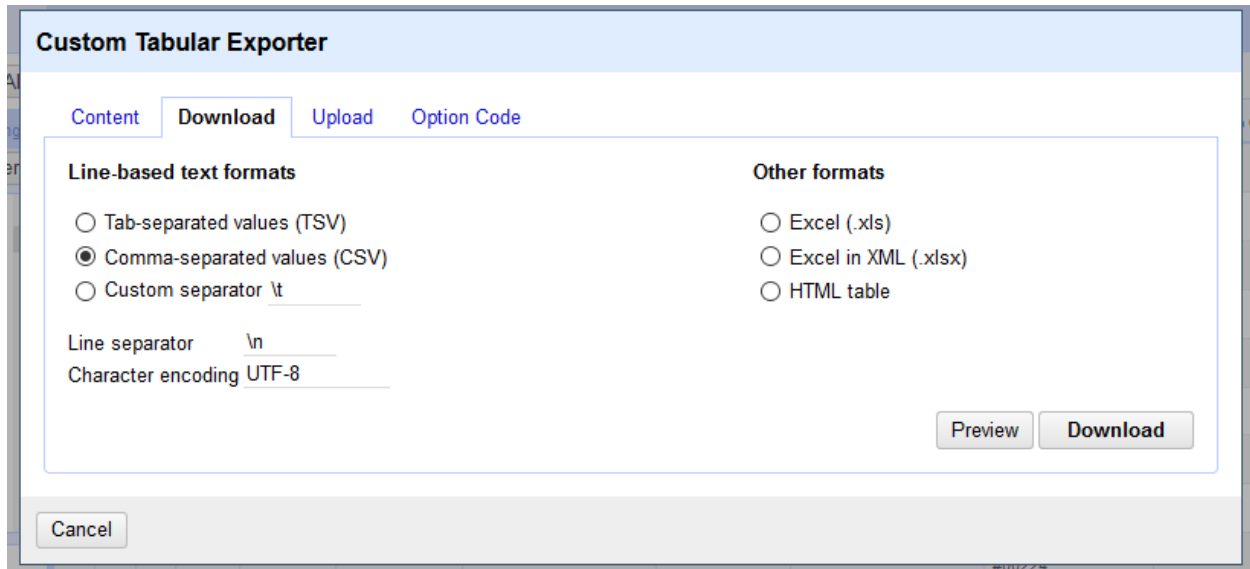
Merchant Name	Comp Code
WAL-MART #3636	HROP
WAL-MART #3021	HROP
PIER 1 IMPORTS #1197	HROP
GUY'S FRENCHYS	HROP
WAL-MART #3636	HROP
ATLANTIC SUPERSTORE #3	HROP
HYPER	HROP

We will export as CSV file because we can then open it in Excel, an SQL database or a GIS program.

You can also use the Custom Tabular Exporter, which gives you far more control over the output file. In the content tab, you choose what to export, including which fields and for date/time fields, the format. Pay special attention to the date/time formats. We are choosing the Full Locale format so the dates are exported to the operating system's native full date/time format.

The screenshot shows the 'Custom Tabular Exporter' interface. At the top, there are tabs for 'Content', 'Download', 'Upload', and 'Option Code'. The 'Content' tab is active. Below the tabs, there are two main sections: 'Select and Order Columns to Export' and 'Options for Trans Date 2'. In the 'Select and Order Columns to Export' section, a list of columns is shown with checkboxes: ID (checked), Line # (checked), Stmt Date (unchecked), Stmt Date 2 (checked), Trans Date (unchecked), Trans Date 2 (checked and highlighted), Merchant Name (checked), Comp Code (checked), and Cost Centre (checked). Below this list are 'Select All' and 'De-select All' buttons. In the 'Options for Trans Date 2' section, there are two columns of options. The first column has radio buttons for 'Matched entity's name' (selected), 'Matched entity's ID' (unchecked), and 'Cell's content' (unchecked). The second column has radio buttons for 'ISO 8601, e.g., 2011-08-24T18:36:10+08:00' (unchecked), 'Short locale format' (unchecked), 'Medium locale format' (unchecked), 'Long locale format' (unchecked), 'Full locale format' (selected), and 'Custom' (unchecked). There are also checkboxes for 'Link to matched entity's page' (checked), 'Output nothing for unmatched cells' (checked), 'Use local time zone' (checked), and 'Omit time' (unchecked). A 'Help' link is visible next to the 'Custom' option. At the bottom of the interface, there are checkboxes for 'Output column headers' (checked), 'Output empty rows (ie all cells null)' (checked), and 'Ignore facets and filters and export all rows' (unchecked). A 'Cancel' button is located at the bottom left.

Once you are ready to export your file, click on the download tab to download a file to your computer and Upload to upload a Google spreadsheet or Fusion Table. The download dialogue allows you to determine the type of export file, the type of delimiter for a delimited text file, the line-ending character for a delimited file and the character encoding to be used for text files.



When you have completed your choices, click on Download to initiate the download. The procedure for uploading to Google spreadsheets or Fusion Tables is different. After choosing either Google sheets or Fusion Tables, you will have to give Open Refine permission to access your Google account. At this time, this functionality does not work properly. A workaround is to download as a delimited text file or Excel sheet, then upload to Google Sheets or Fusion Tables through the Google Drive interface.

Open Refine has a number other data manipulation tools that can come in handy when preparing your data for analysis. You can use it as an alternative to a spreadsheet or database TRIM function, to remove leading and trailing whitespace from columns of data, to convert all the text in a column to uppercase, lowercase or title (normal) case, to convert numbers stored as text to numbers, to convert dates stored as text to date format, and numbers and date to text format. To access these functions, click on the arrow at the top of the column you wish to modify, then choose Common transforms and the function you wish to use.

Merchant Name	Merchant2	Comp Code	Cost Centre	G/L	Order #	Narr
Facet	al-mart #3636	HROP	0	6404	RP9AM0R0	PEACE, TERRI
Text filter	al-mart #3021	HROP	0	6404	RP9AM0R0	PEACE, TERRI
Edit cells	Transform...		0	6404	RP9AM0R0	PEACE,
Edit column	Common transforms					
Transpose	Fill down					E,
Sort...	Blank down					E,
View	Split multi-valued cells...					E,
Reconcile	Join multi-valued cells...					ILL
Michaels #3955	Mi					ILL
Cdn Tire Store #00224	Cdn Tire Store #00224	HROP				ILL
Target Canada T3731	Target Canada T3731	HROP				ER,

- Trim leading and trailing whitespace
- Collapse consecutive whitespace
- Unescape HTML entities
- To titlecase
- To uppercase
- To lowercase
- To number
- To date
- To text
- Blank out cells

These tools mostly duplicate functionality you will find in a spreadsheet or database program, but it's always good to have options.

Giving data structure with Regular Expressions

A messy data structure can be cleaned up in Excel, a database program, or Open Refine, but what to do with data that has little or no structure at all? What if it's just a giant blob of text? When you copy data from a webpage, it's not uncommon for all the coding to disappear once you've pasted it into Excel. The delimiters needed to keep the columns separate can get lost in translation and all the fields will appear in the first column.

Or maybe you want to conduct a data analysis on large blocks of texts, such as transcripts of political debates or legal proceedings.

In these cases, you need to create structure to your data where there is none.

To do this, you must find patterns in the text that indicate where a delimiter should appear.

Consider a long list of names and addresses that has no delimiters and nothing to break up the data into fields:

Paul Jackson 148 Bexhill Close London ON N6E 3B1
J David A Jackson 3800 Yonge St Toronto ON M4N 3P7
Dennis Jackson 188 Green Brighton ON K0K 1H0
A D Jackson 10250 Kennedy N Brampton ON
Malcolm Jackson 89 Winchester St Toronto ON M4X 1B1
G Jackson 3130 Council Ring Rd Mississauga ON L5L 1L4
Rod Jackson 445 Simon Fraser Dr Thunder Bay ON P7C 4Z9
James C B Jackson 415 Norfolk St S Simcoe ON N3Y 2W8
Lawrence D A Jackson ON
Robert Jackson 82 Viscount Ave Ottawa ON K1Z 7M9
Paul Jackson ON
D Jackson 19 Wood St Trenton ON K8V 5P6
B Jackson 1545 Rue Deroy Saint-Calixte QC J0K 1Z0

Each street address is a different length and begins with a different number. There's no way to know where to put in a delimiter. You could go through every line and manually insert tabs in the right places, but that isn't practical with a long list.

Instead, you need to find a way to automate this task.

Fortunately, with addresses, there is typically a common pattern: a street number followed by a street name and one of the common suffixes, such as Avenue, Road or Boulevard.

Then, usually, this is followed by the name of the municipality and then a postal code or zip code.

Breaking a blob of address data into fields is such a common data journalism task that makes it worthwhile to learn how to use a powerful set of pattern search tools called Regular Expressions.

Also called RegEx, these functions are supported by commonly used (and open-source) text editors that have long been used by computer programmers and should be in every data journalist's

toolkit. To use Regular Expressions for cleaning up data, you should download and install one of the many open-source text editors that support them. On a Mac, TextWrangler is without peer. On Windows, try [Notepad++](#).

Most of us know how to do a conventional search-and-replace in Microsoft Word or other text-based programs, using CTRL-F (or Command-F on a Mac) to locate a string of text and substitute it with something else.

These functions are typically limited to specific strings of text. If you want to change every reference to “Minister of Defence” to “Minister of Finance,” these work fine.

But imagine the common task of trying to find postal codes within a long file of addresses. You can search for **K2P 3C4**, but that will match only that exact postal code and not others. The power of Regular Expressions is that they can locate patterns of numbers and letters, not specific text.

To find a postal code using RegEx, we just need to write an expression that matches the pattern, which is: a capitalized letter, a number, another capitalized letter, a hyphen or space, then a number, a capitalized letter and another number.

In RegEx, the phrase **[A-Z]** will match for capitalized letter (include the square brackets). Similarly, **[0-9]** will match any digit.

So, to find any postal code using RegEx, we would open up the text in TextWrangler or Notepad++ and search for the expression **[A-Z][0-9][A-Z]-[0-9][A-Z][0-9]**.

Suppose we want to put tabs on either side of the postal codes, so that when we import the text to Microsoft Excel, they will appear in their own column.

If we replaced with only tabs, using the expression `\t`, we would lose the postal code our pattern located. We need to include it in the replace phrase.

But since we don't know the exact postal code we're matching, we need use to a RegEx function called “[backreferences](#)”, which store patterns we've already matched to be recalled later. These are created simply by putting parentheses around all or part of the search pattern, then recalled in the replace expression with `\1`, with the number referring to order in which it was stored.

Imagine that our data is formatted with an empty space in the middle of the postal code, and we want to replace it with a hyphen. We can't replace every space in the document with hyphen, but only those within the pattern of postal codes.

We would search for **([A-Z][0-9][A-Z]) ([0-9][A-Z][0-9])**. This will store the first three characters as one backreference, and then the second three as another.

So let's search for that, then replace with the expression `\t\1-\2\t`.

For every postal code, this expression will replace it with a tab, followed by the first three characters of the original postal code that were stored as a back expression, then the hyphen we want, then the last three characters, and another tab.

If we felt like it, we could have reversed the order of the postal code by replacing it with the phrase `\t\2-\1\t`. That would turn “M4C-5T5” into “5T5-M4C”.

RegEx are particularly powerful matching fuzzy text phrases. Suppose we’re working with a messy list of addresses that had been manually entered by different users. Some wrote out the province as “Ontario”, but others wrote “Ont.”, others “Ont.” or “On.”, and some others typed it as “Ontaroi.”

We can replace all these by telling RegEx to find a capitalized “O” and lower-case “n” and whatever comes after it, and replace the whole mess with the proper “Ontario”.

For this expression, we’ll use a period `.` which is RegEx’s [wildcard](#) version of the asterisk and matches any character or space. We’ll also use a plus sign `+`, which tells the RegEx to find any number of the thing comes before it, and the question mark `?` which tells our expression to stop looking soon as it hits something else we specify—in this case, a space `.`

So, we search for `On.+?` (with a space after the `?`) and replace with `\tOntario\t`. We can throw in a `\r` for a hard return at the end of the expression if that’s the end of the record.

By learning to combine multiple characters and wildcards into Regular Expressions, we can take messy unstructured data and turn it into beautifully structured rows and columns.

It can also take large blocks of text that appear to have little structure and turn them into structured data. They could be used to take, for example, thousands of pages of House of Commons transcripts and transform them into a database of quotes organized by MP, party, data and even subject.

The key to adding structure in large blocks of texts is to find ways that different types of data is indicated. In a transcript of House of Commons debates, it may be indicated by the MP’s name followed by her or her constituency set off with parentheses. Replace the `(` and `)` characters with tabs, then import the data in Excel and the data should be structured to allow sorting by MP, constituency and statement.

The *Ottawa Citizen* used this technique In 2011 when the New Democrat Party MPs were filibustering—speaking at length without giving up the floor—to delay the Conservative government’s legislation that would force Canada Post employees back to work.

The *Citizen* copied the textual transcripts of the parliamentary debates, called Hansard, then added structure by inserting tabs using Regular Expressions.

A quick fun analysis of these data found the one MP who spoke the most and calculated that the total number of words he uttered in the filibusters to be 432,143. If he speech during the filibuster was printed in book form, it would about 55 per cent of the length of the King James Bible and 77 per cent of Tolstoy's War and Peace.

Conclusion

Data analysis is a powerful tool in the journalist's toolkit, but if the data is dirty, the analysis may produce junk results. Taking the time to clean your data can be the difference between an award-winning story and a huge correction. It's an essential part of the process.