CHAPTER 7 GETTING YOUR DATA IN SHAPE

In the previous chapter, we learned how to use integrity checks to uncover flaws in our data. In this chapter we will learn some tricks that we can use to scrub some of the most common data dirt. We will also learn some techniques for transforming our data, so they are better prepared for analysis and visualization.

We will use Excel to perform most of our data cleaning and transforming tasks, but we will also use OpenRefine and a PDF conversion service.

Excel and other spreadsheet programs have built-in features and functions that can help whip data into shape. For example, we can use Excel's Text to Columns tool to **parse**—or split—data that are stored in one column into multiple columns. This is useful when we have a column for full names that are stored like, "Doe, Jane". Using **Text to Columns**, we could carve this into one column for the last name, and another column for the first name. Excel also has a Remove Duplicates tool that allows us to remove rows that hold data that are repeated. In addition, we can write formulas in our Excel spreadsheets to carve out data and then assemble them. This is handy when we have dates stored as text, such as "20150101" for January 1, 2015.

OpenRefine (which runs on Windows, Mac and Linux computers) is an open-source program from Google that lets users quickly run data integrity checks, then clean flawed data. Its creators call it a power tool for working with messy data (http://openrefine.org).

Cometdocs and Zamzar are two online converters that allow us to extract Excel files from PDF tables.

Then there are even more-powerful and more-complex tools for data cleaning that are outside the scope of this book. More-advanced data users will use database managers, such as Microsoft Access or MySQL, to clean data using string functions. String functions are code that can be inserted into the **Structured Query Language** (SQL) that these database managers execute. For instance, someone who's adept with SQL can write queries to rearrange dates into a proper format and put them into a new column. Data users who are even more advanced write scripts using programming languages like **Python**, **Ruby**, PHP or **Perl** to clean and transform data sets. These programming languages allow users to process huge amounts of data more quickly than they could using traditional programs like database managers and spreadsheets. In addition, a number of companies offer data cleaning services and software.

For a deeper look at data cleaning, check out the *Bad Data Handbook* (McCallum, 2012) or *Best Practices in Data Cleaning* (Osborne, 2013), which is geared to working with primary data collected for the purposes of research.

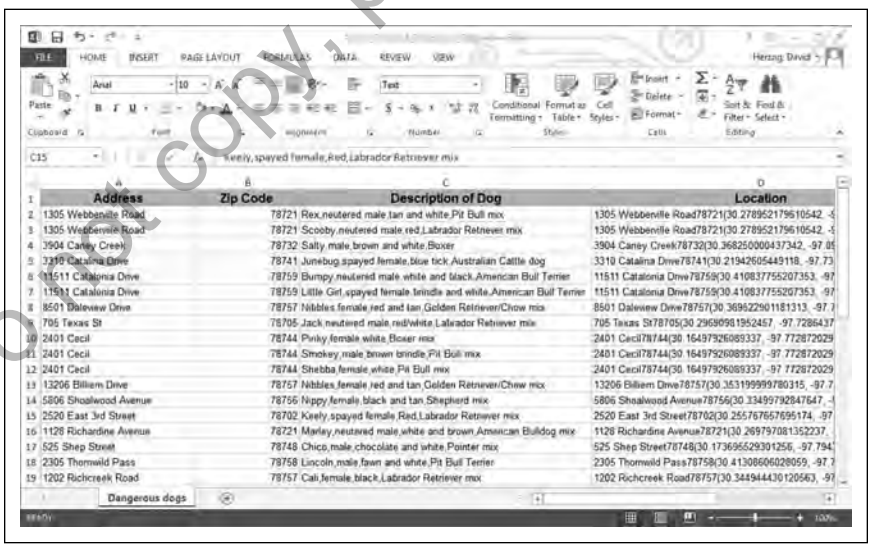
There's plenty that we can do, though, with Excel, PDF converters and OpenRefine, so we'll start to tackle some of the most common challenges now.

COLUMN CARVING

Government agencies often stuff too much data into one column, which makes it difficult for us to work with them. In Chapter 5, we took a look at political campaign contribution data from the Missouri Ethics Commission that stores all the identifying data about contributors into one column. This is an extreme example of dirty data. Someone with advanced data-cleaning skills might be able to parse these data elements into separate fields, but that would take a lot of work. As shown in that chapter, the commission now provides these data with the data broken down into multiple columns that users can work with better.

Most column parsing challenges are far simpler, such as the one in our spreadsheet of dangerous dogs that have been reported to animal control authorities in the city of Austin, Texas. The city's Animal Services Office allows people to fill out forms asserting that a dog is dangerous or has bitten another animal (AustinTexas.gov, n.d.). The original file has been modified for this exercise. The file is called Austin_Declared_Dangerous_Dogs.xlsx and has 40 rows, including one for the column headers. (You should have your data notebook open so you can record any activities.)

We see four columns, one each for the street address, zip code, description of the dog and the dog's location, which includes latitude and longitude points that someone



	Address	Zip Code	Description of Dog	Location
2	1305 Webberville Road	78721	Rex,neutered male,tan and white,Pit Bull mix	1305 Webberville Road78721(30.278952179610542, -9
3	1305 Webberville Road	78721	Scooby,neutered male,red,Labrador Retriever mix	1305 Webberville Road78721(30.278952179610542, -9
4	3904 Carley Creek	78732	Salty,male,brown and white,Boxer	3904 Carley Creek78732(30.368250000437342, -97.05
5	3310 Catalina Drive	78741	Junebug,spayed female,blue tick,Australian Cattle dog	3310 Catalina Drive78741(30.21942605449118, -97.73
6	11511 Catalonia Drive	78759	Bumpy,neutered male,white and black,American Bull Terrier	11511 Catalonia Drive78759(30.410837755207353, -97
7	11511 Catalonia Drive	78759	Little Girl,spayed female,brindle and white,American Bull Terrier	11511 Catalonia Drive78759(30.410837755207353, -97
8	8501 Daleview Drive	78757	Nibbles,female,red and tan,Golden Retriever/Chow mix	8501 Daleview Drive78757(30.369622901181313, -97.7
9	705 Texas St	78705	Jack,neutered male,red/white,Labrador Retriever mix	705 Texas St78705(30.296909811952457, -97.728643
10	2401 Cecil	78744	Pinky,female,white,Boxer mix	2401 Cecil78744(30.16497926089337, -97.772872029
11	2401 Cecil	78744	Smokey,male,brown brindle,Pit Bull mix	2401 Cecil78744(30.16497926089337, -97.772872029
12	2401 Cecil	78744	Shebba,female,white,Pit Bull mix	2401 Cecil78744(30.16497926089337, -97.772872029
13	13206 Billiem Drive	78757	Nibbles,female,red and tan,Golden Retriever/Chow mix	13206 Billiem Drive78757(30.353199999780315, -97.7
14	5806 Shoalwood Avenue	78756	Nippy,female,black and tan,Shepherd mix	5806 Shoalwood Avenue78756(30.33499792847647, -9
15	2520 East 3rd Street	78702	Keely,spayed female,Red,Labrador Retriever mix	2520 East 3rd Street78702(30.255767657695174, -97
16	1128 Richardine Avenue	78721	Marley,neutered male,white and brown,American Bulldog mix	1128 Richardine Avenue78721(30.269797081352237,
17	525 Shep Street	78748	Chico,male,chocolate and white,Pointer mix	525 Shep Street78748(30.173695529301256, -97.794
18	2305 Thornwild Pass	78758	Lincoln,male,fawn and white,Pit Bull Terrier	2305 Thornwild Pass78758(30.413086060280059, -97.7
19	1202 Richcreek Road	78757	Cali,female,black,Labrador Retriever mix	1202 Richcreek Road78757(30.344944430120563, -97

Source: Retrieved from Data.austintexas.gov.

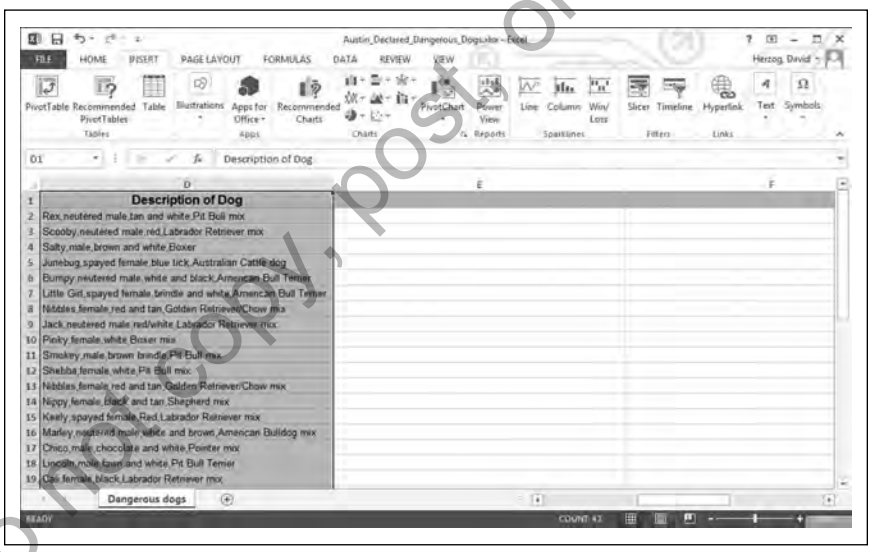
Note: Austin declared dangerous dog data. Note the Description of Dog column has four pieces of data, separated by a comma.

could use to create a webmap. The Description of Dog column has a lot of different bits of data mashed together: the dog's name, sex, color and breed. Each piece of data is separated by a comma, a pattern that is important to note before using the Text to Columns tool.

Let's get ready by saving a copy of the spreadsheet with a name other than the original, so we preserve the original copy. We will work with the new file.

If we use Text to Columns, it will overwrite our original data in Column C. We want to preserve that for reference, so insert four columns to the right of Column C. We'll need four columns because we're going to parse out four data items. Right-click on Column D and pick Insert from the popup menu four times. You should have blank columns from D through G.

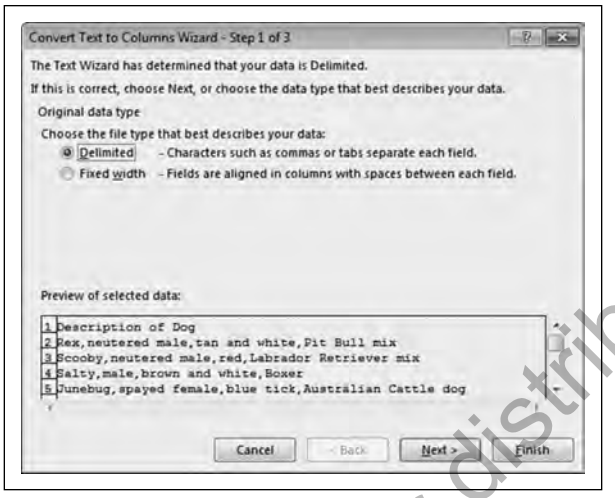
Practice safe computing by copying the contents of Column C into Column D: Right-click on C and pick Copy from the popup menu, then right-click on D and pick Paste. Your spreadsheet should look like this with Column D highlighted:



Source: Retrieved from Data.austintexas.gov.

Note: Description of Dog column, ready for parsing.

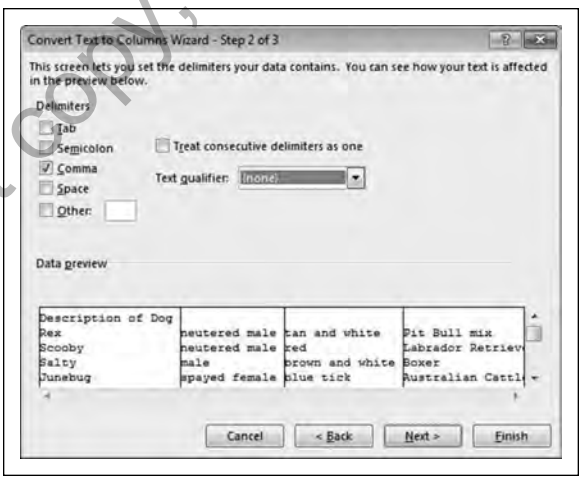
Now we're ready to tell Excel to perform its magic. Click on the Data tab and then the Text to Columns button. Excel launches a Wizard that walks us through carving up the data in three steps. In the first step, tell Excel the correct data type, which is delimited.



Source: Microsoft Excel for Windows 2013.

Note: Step 1 of the Text to Columns Wizard for delimited data.

Click Next and it's time to set more options. We need to tell Excel that the delimiter is a comma, so make sure only that box is checked in the Delimiters section. We have no text qualifiers (single or double quotation marks to denote text), so change that to None. In the preview area, Excel draws lines where the column breaks will go, based on the position of the comma delimiters.

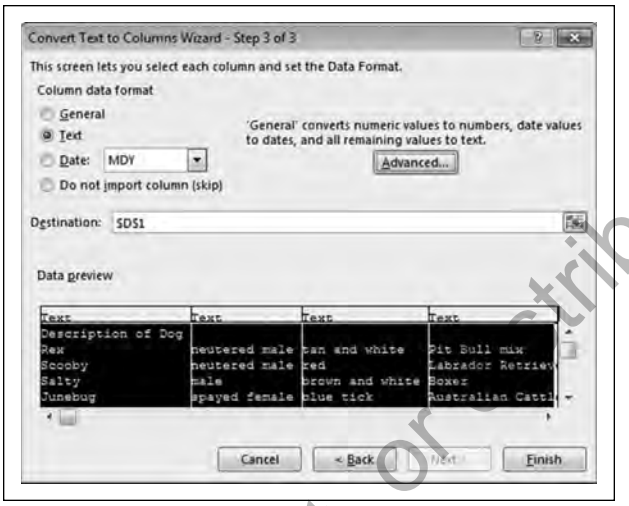


Source: Microsoft Excel for Windows 2013.

Note: Step 2 of the Text to Columns Wizard for delimited data.

Click Next and we're in the last step of our wizard, where we set the data types for the four columns that we are creating. Change all of these from General to Text. Do that by selecting all of the headers, which say General, and changing the Column data format to Text. Click Finish and Excel carves up the data into four columns. If you get a warning

about overwriting existing data, click OK. That's just Excel's way of telling you that the data in Column D will be modified. (This is why we saved the original in Column C.)



Source: Microsoft Excel for Windows 2013.

Note: Step 3 of the Text to Columns wizard for delimited data.

Source: Retrieved from Data.austintexas.gov.

Note: Parsed Description of Dog column. Note that each of the four pieces of data is stored in its own column.

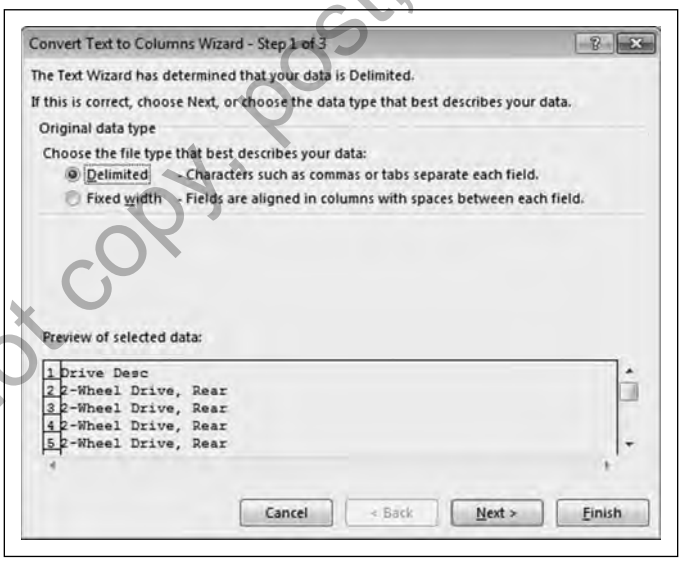
So Column D holds the name of the dog, Column E the sex and whether it's been neutered or spayed, F the color and G the breed. Now that we've placed the breed of the dog in its own column, we could create an Excel pivot table that tells us which breed is the most common on the dangerous dogs list. Let's practice safe computing

by adding some labels that make sense: Change the contents of cell D1 to "name", E1 to "sex", F1 to "color" and G1 to "breed". Now save the file and close it because we are done with it for now.

Carving up data using Text to Columns usually is more complicated than this because of the way government agencies store data. Get the Excel file fuel_economy2013.xlsx from the book website and open it. Make a working copy using File | Save As. This file from the U.S Department of Energy and the Environmental Protection Agency lists fuel economy data for vehicles that are sold in the United States. The data are in 1,166 rows, including one for the headers. Let's scroll over to column AC, which provides descriptive data about each vehicle's drive system. The first line tells us that the drive system for one particular variant of the Aston Martin V8 Vantage is "2-Wheel Drive, Rear". That's actually two pieces of data: the drive system is two-wheel and those two wheels are the rear ones. We could use the comma and space that follows it to carve this into two distinct columns.

Using Insert, create two new columns, one for the number of wheels and the other to designate which wheels propel the vehicle. Copy the contents of AC and paste them into AD. Make sure AD is highlighted and start the Text to Columns Function.

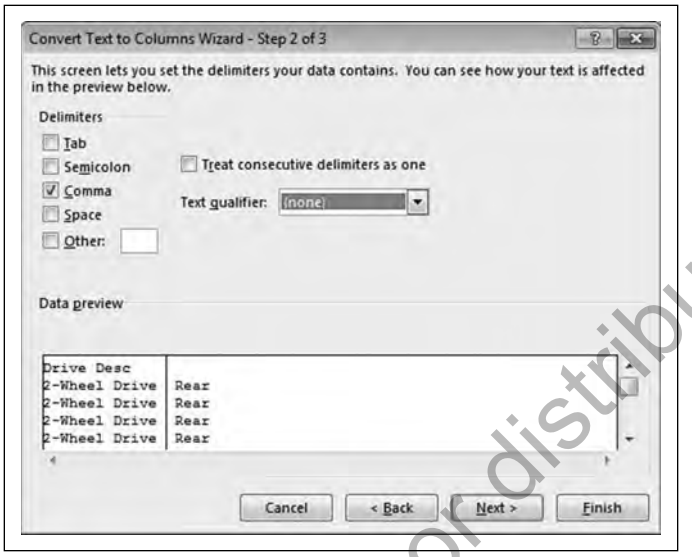
Choose delimited in the first screen of the wizard and pick Next.



Source: Microsoft Excel for Windows 2013.

Note: Step 1 in the Text to Columns wizard for delimited data.

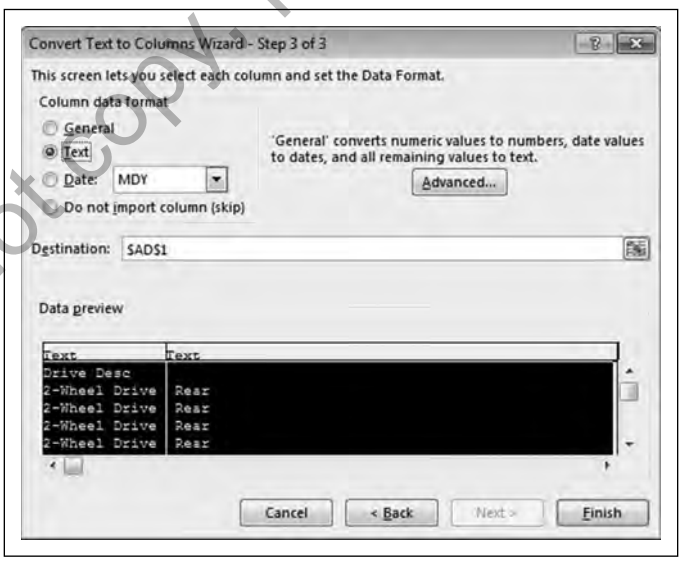
Set the delimiter to Comma only and make sure text qualifier is None. Excel gives a preview of how it's going to parse the data. Look closely and you will see that Excel is including the spaces on either side of the comma as text. That's OK for now. We'll fix that later.



Source: Microsoft Excel for Windows 2013.

Note: Step 2 in the Text to Columns wizard for delimited data.

Click Next to move on to the third and final step. Change both of these columns to Text and then Finish.



Source: Microsoft Excel for Windows 2013.

Note: Step 3 in the Text to Columns wizard for delimited data.

This is great—Excel has carved the data up in Columns AD and AE. Label AD "NumWheels" (for number of wheels) and AE "DriveWheels".

AC	AD	AE
Drive Desc	**Drive Desc**	
2-Wheel Drive, Rear	2-Wheel Drive	Rear
2-Wheel Drive, Rear	2-Wheel Drive	Rear
2-Wheel Drive, Rear	2-Wheel Drive	Rear
2-Wheel Drive, Rear	2-Wheel Drive	Rear
All Wheel Drive	All Wheel Drive	
2-Wheel Drive, Rear	2-Wheel Drive	Rear
2-Wheel Drive, Rear	2-Wheel Drive	Rear
2-Wheel Drive, Rear	2-Wheel Drive	Rear
2-Wheel Drive, Rear	2-Wheel Drive	Rear
2-Wheel Drive, Rear	2-Wheel Drive	Rear
All Wheel Drive	All Wheel Drive	

Source: Department of Energy.

Note: Data parsed, but with extraneous spaces.

Our next step is to remove those extraneous spaces using Excel. We want to remove these because the spaces technically are data (one of the basic ASCII characters) and could throw off any analysis. Column AE has leading spaces in the cells. We have "Rear" or "Front" for the values, when we really want no leading spaces in them. Excel's TRIM function can help here. TRIM removes any leading and trailing spaces, as well as duplicated spaces inside a text entry.

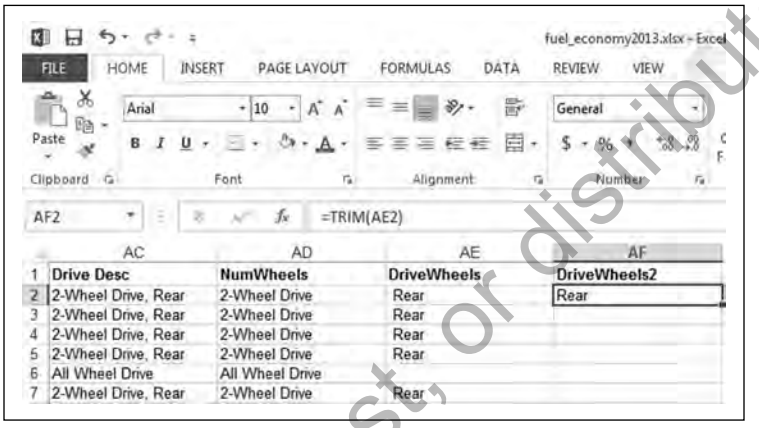
Insert a new column to the right of AE and make sure it is formatted as General, or else your formula will appear, instead of "Rear" or "Front". Format the column by highlighting it, then clicking the Home tab. Select General from the options in the drop-down list.

	AC	AD	AE	AF
1	**Drive Desc**	**NumWheels**	**DriveWheels**	
2	2-Wheel Drive, Rear	2-Wheel Drive	Rear	
3	2-Wheel Drive, Rear	2-Wheel Drive	Rear	
4	2-Wheel Drive, Rear	2-Wheel Drive	Rear	
5	2-Wheel Drive, Rear	2-Wheel Drive	Rear	
6	All Wheel Drive	All Wheel Drive		
7	2-Wheel Drive, Rear	2-Wheel Drive	Rear	
8	2-Wheel Drive, Rear	2-Wheel Drive	Rear	
9	2-Wheel Drive, Rear	2-Wheel Drive	Rear	
10	2-Wheel Drive, Rear	2-Wheel Drive	Rear	
11	2-Wheel Drive, Rear	2-Wheel Drive	Rear	

Source: Department of Energy.

Note: New column in fuel efficiency data.

In cell AF2, enter the trim formula: "=TRIM(AE2)". This says, Trim the contents of cell AE2. Hit Enter and you should see the new, trimmed version. Copy this for all of your cells in this column by double-clicking on the square at the bottom right of your cursor box.



	AC	AD	AE	AF
1	**Drive Desc**	**NumWheels**	**DriveWheels**	**DriveWheels2**
2	2-Wheel Drive, Rear	2-Wheel Drive	Rear	Rear
3	2-Wheel Drive, Rear	2-Wheel Drive	Rear	
4	2-Wheel Drive, Rear	2-Wheel Drive	Rear	
5	2-Wheel Drive, Rear	2-Wheel Drive	Rear	
6	All Wheel Drive	All Wheel Drive		
7	2-Wheel Drive, Rear	2-Wheel Drive	Rear	

Source: Department of Energy.

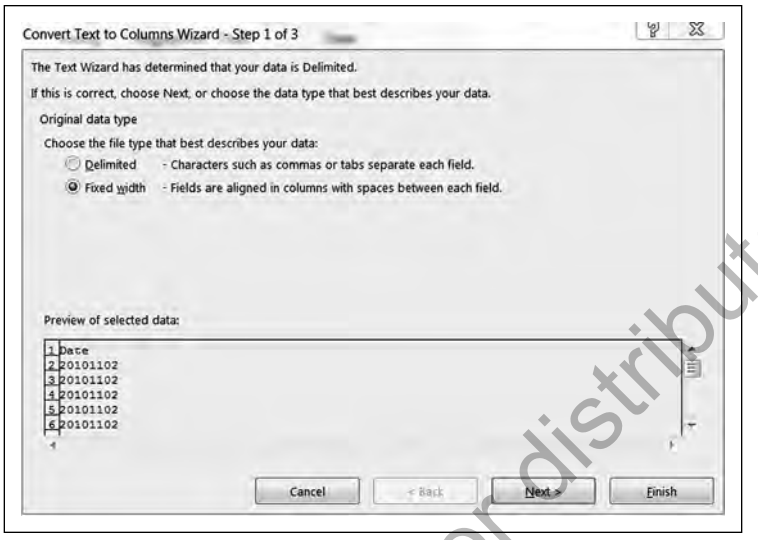
Note: Trim function.

These are just two examples of how you can carve data in Excel using Text to Columns. Sometimes we do the opposite and combine the contents of multiple columns into one. For that, we turn to concatenation.

CONCATENATE TO PASTE

We're going to use **concatenation** to put election date data into a proper form. Open the voterturnout.xlsx Excel file, which holds data about voter turnout by precinct in Boone County, Missouri, during the November 2, 2010, general election. The spreadsheet has columns for the precinct or voting district, number of people who voted, number registered and date. All of the dates are recorded as "20101102". Our goal is to turn all of those to 11/2/2010.

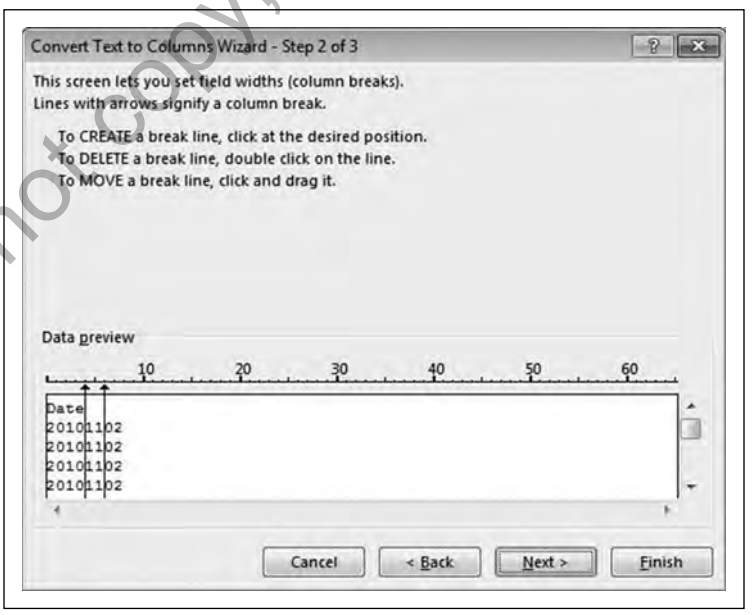
We'll use the Text to Columns tool to carve the date and then use concatenation to put the pieces back together. Practice safe computing by copying the contents of Column D into Column E, which we will parse. Highlight E and pick the Text to Columns button from the data menu. That launches a familiar wizard to help. Our data are not delimited this time, so punch the Fixed width button. We can tell that the dates are fixed width because the characters for year, month and day are in consistent locations.



Source: Microsoft Excel for Windows 2013.

Note: Step 1 in Text to Columns for fixed-width text.

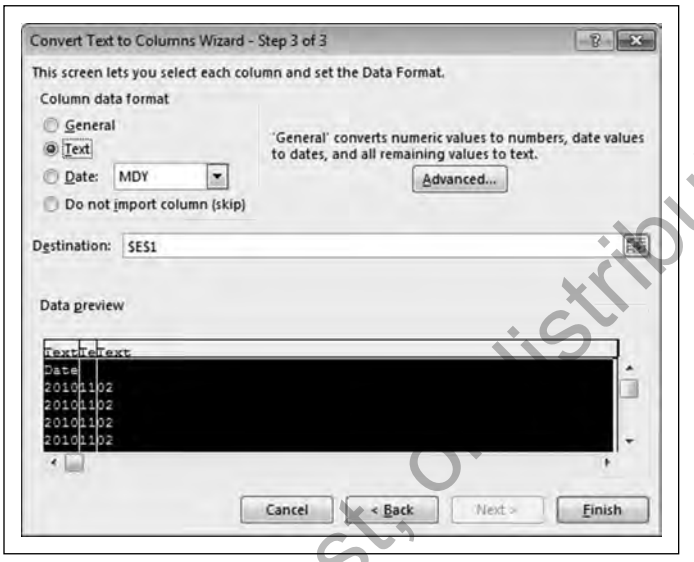
Click Next and we're at Step 2. In this screen, we create the column dividers by clicking where we'd like them to go inside the Data preview area. Don't worry if you put a break in the wrong position—Excel allows you to move or remove it. Create column breaks at positions 4 and 6, just like this.



Source: Microsoft Excel for Windows 2013.

Note: Step 2 in Text to Columns for fixed-width text.

Click Next and go to the final step. Make sure all of the new columns are set to text and click Finish.



Source: Microsoft Excel for Windows 2013.

Note: Step 3 in Text to Columns for fixed-width text.

Success! Excel has put the year in Column E, month in F and date in G. Label those three columns accordingly and save a working copy of the file with File | Save as.

	Precinct	Voting	Registered	Date	Date		
2	1A	360	1030	20101102	2010	11	02
3	1B	234	420	20101102	2010	11	02
4	1C	708	1159	20101102	2010	11	02
5	1D	474	1136	20101102	2010	11	02
6	1E	75	2704	20101102	2010	11	02
7	1F	162	1011	20101102	2010	11	02
8	1G	373	759	20101102	2010	11	02
9	1I	163	1026	20101102	2010	11	02
10	2A	629	957	20101102	2010	11	02
11	2B	503	1171	20101102	2010	11	02

Source: Boone County, Missouri, Clerk.

Note: Fixed-width text parsed.

Label Column H as "Date2", which is where we'll put the restructured date that we'll build with concatenation, which uses the ampersand (&) character.

In cell H2 enter "=F2&"/"". This tells Excel to take the contents of cell F2 (or 11 for the month of November) and tack on a slash. We need to put the slash in double quotation marks, so Excel treats it as a text character. We get "11/" for a result, which is a good start.

Next we will tack on the date, by building on the formula "=F2&"/"&G2". Now we see "11/02".

As the last step, will add another slash and the year: "=F2&"/"&G2&"/"&E2". Now we have "11/02/2010" in our cell. Copy this for all of the cells in Column H and format it as a short date. Save the file and close it.

DATE TRICKS

Excel offers other functions that transform data stored in dates. These are handy for generating a month or a year. Open the disaster_declarations.xlsx Excel file and we'll see how this works. This file, originally downloaded from the federal government's Data.gov portal, holds data about more than 4,100 major disasters declared since 1953. Under federal law, state governors may request a declaration of a major disaster and receive federal assistance (Federal Emergency Management Agency, n.d.). In Column G, we have a declaration date, but not a month or year. So it would be difficult if we wanted to analyze the data from either of those dimensions.

Excel's YEAR and MONTH function can extract these data. We'll use another function to generate the day of week, by name, for the declaration dates.

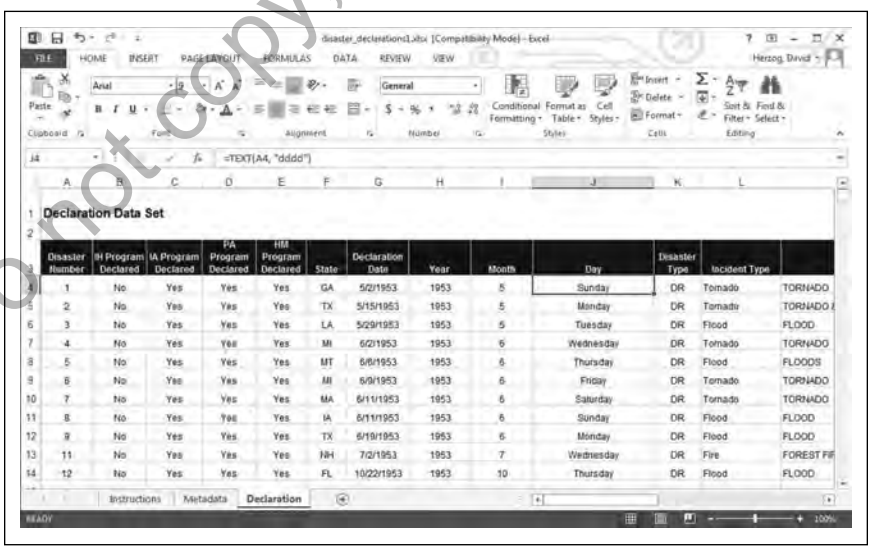
Insert three columns between columns G and H, so we have a place to put the year, month and day of the week. Label these "Year", "Month" and "Day". Format these columns as General.

In cell H4, enter "=Year(G4)" and "1953" appears. Copy this formula for all of the cells.

In cell I4, enter "=Month(G4)" and "5" appears for May. Copy this formula for all of the cells.

In cell J4, enter "=TEXT(A4, "dddd")" and Sunday appears. Copy this formula for all of the cells. The TEXT function, in this case, returns a text value from the data stored in A4. dddd tells Excel to generate the day using its full name. Use ddd to generate an abbreviation.

Use File | Save As to save a working copy of your file and close it.



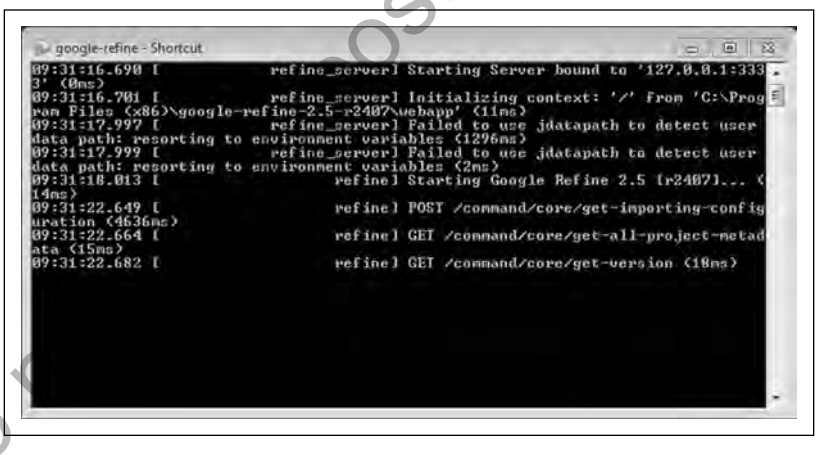
Source: Federal Emergency Management Agency.

Note: Year, month and day extracted from date in Federal Emergency Management Agency disaster declaration data set.

POWER SCRUBBING WITH OPENREFINE

OpenRefine, a free and open-source program, can help for many data cleaning challenges. (Find the program for Windows, Mac and Linux operating systems here: https://github.com/OpenRefine/OpenRefine/wiki/Installation-Instructions.) We'll use OpenRefine to perform one of the most common, frustrating and time-consuming tasks: standardizing data entries. In Chapter 5, we learned how poor data-entry controls can lead to messy data. Such is the case in this Excel file of campaign contributions released by the Missouri Ethics Commission named mo12contribs.xlsx. Open the file and note that we have 19,867 rows, including one for the headers. Each row represents a campaign contribution made by a person, company or political committee. We can take an educated guess that Column J, which has the city of the contributor, might have a lot of typos.

Start OpenRefine. In Windows, the program launches a command-line window, which you might be familiar with if you have ever done any programming. Then it opens the program inside your default browser—here Google Chrome. Note that the address that Chrome is pointed to is http://127.0.0.1:3333, which refers to your machine. In other words, the browser is working locally, not over the Internet.



Source: OpenRefine.

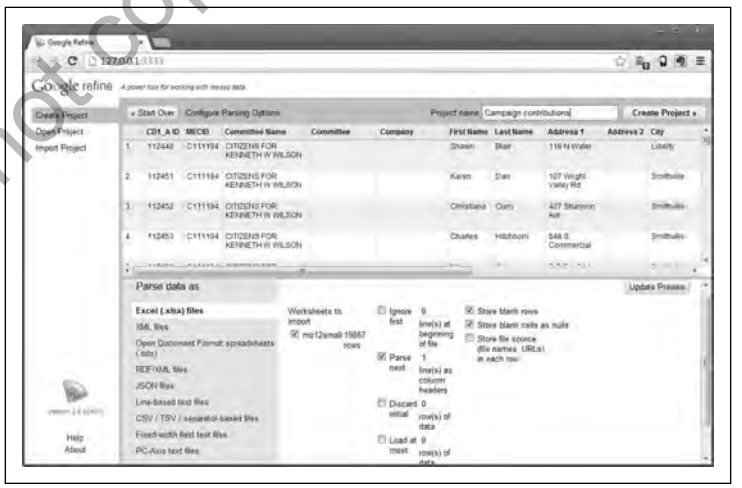
Note: OpenRefine command-line window. OpenRefine opens this window upon startup on Windows machines.

Source: OpenRefine.

Note: OpenRefine create project screen.

Let's create a project in Refine, so we can clean the data. Click Create Project | Get data from | This Computer and find the mo12contribs.xlsx file. Click Next and Refine takes us to the next step for loading the file.

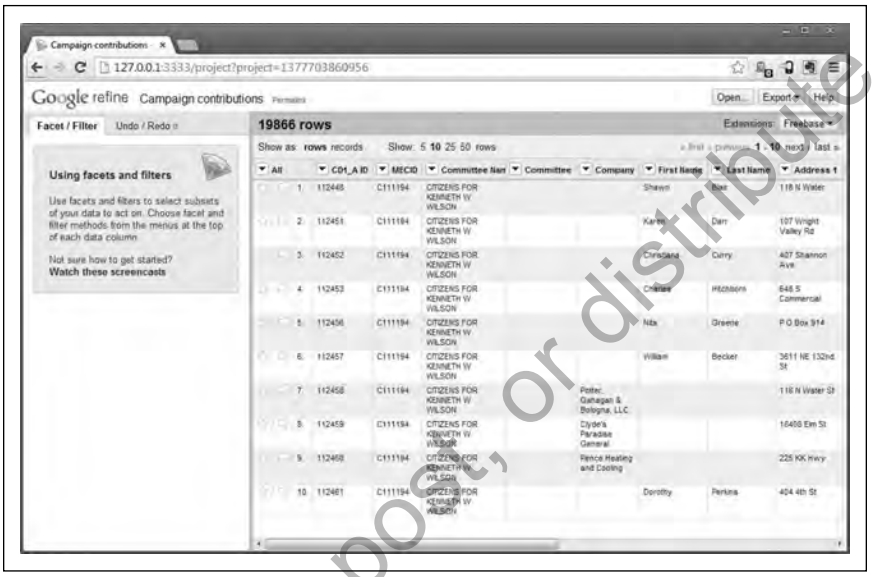
Change the project name to "Campaign contributions" and make sure that the option to parse next line as column header at the bottom is checked. This tells Refine to use the first row from the Excel file as column headers. Note that the top of the screen previews our data.



Source: OpenRefine.

Note: OpenRefine project preview screen.

Click Create project and Refine loads the data. By default, it shows only 10 rows at a time. You can change that if you like. Working with data in Refine typically takes two steps: detecting data problems, then correcting them. One of the most common ways to detect the flaws is by creating **Facets**, or views of your data that are very much like spreadsheet pivot tables.



Source: Missouri Ethics Commission.

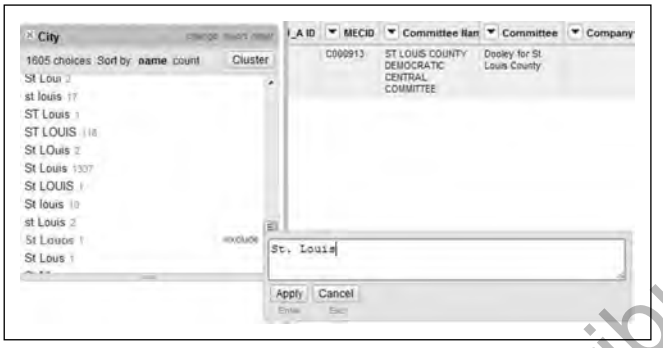
Note: OpenRefine project screen, data successfully loaded.

Create a Facet for the City column by clicking on its drop-down arrow, then selecting Facet / Text facet from the menu. Scroll down the results and we can see, for example, that the city of St. Louis (proper spelling) has been misspelled many different ways. We could click on the edit line for each misspelling and manually change the city names to our proper spelling, but that would take a whole lot of time. Fortunately, Refine has some powerful tools that will allow us to clean up these misspelled city names en masse.

That's where Clustering can help. **Clustering** uses algorithms designed to detect text values that might be the

Source: Missouri Ethics Commission.

Note: Text facet for City column. Note that the results are very similar to those created by an Excel pivot table.

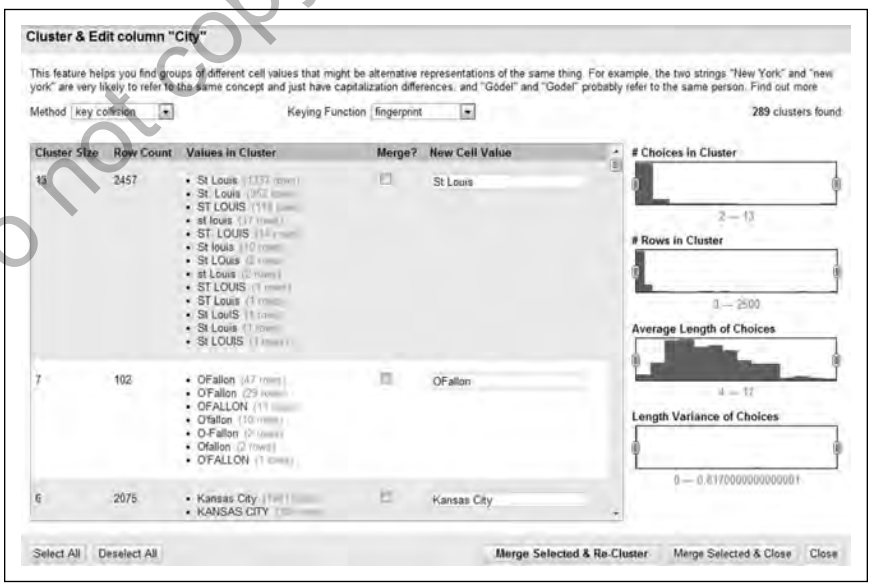


Source: Missouri Ethics Commission.

Note: Manually editing a misspelling.

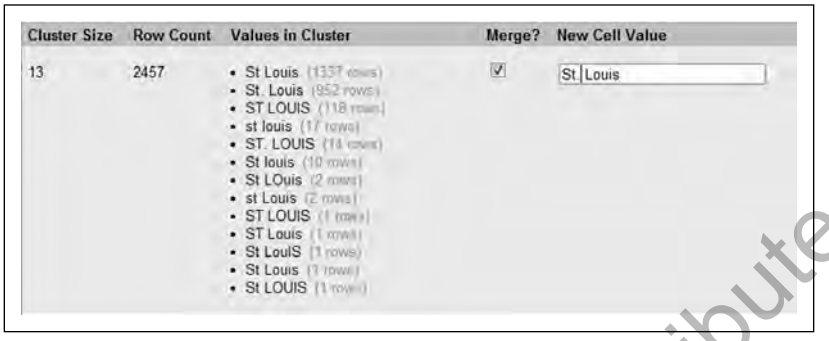
same. Go ahead and click the Cluster button on the Facet box for City. Refine employs two methods for clustering—key collision and nearest neighbor. Each of those methods has multiple Keying Functions. By default, Refine clusters by key collision/fingerprint. For details about how these algorithms work, see the documentation at https://github.com/OpenRefine/OpenRefine/wiki/Clustering-In-Depth.

Using key collision/fingerprint, Refine shows us how it has clustered city names, including many St. Louis variants at the top of the list. Let's look closely: Refine says the Cluster Size is 13, or that it found 13 different values for clustering here. The Row Count is 2,457. Under Values in Cluster, we see the 13 different spellings. Merge? asks whether we want to transform these values into something else. We do, so check this box. In the New Cell Value box, enter our proper spelling of St. Louis.



Source: Missouri Ethics Commission.

Note: Clustering in OpenRefine.



Source: Missouri Ethics Commission.

Note: Detail of clustering for variations of St. Louis.

Click Merge Selected & Re-Cluster and Refine **standardizes** all of those St. Louis misspellings. We see that we have a whole bunch more for O'Fallon, Kansas City, Lee's Summit and so on. If we were really using this column for analysis or visualization, we'd want to go through each of these clusters, then apply all of the other methods and keying functions. Close the clustering box.

Let's export our cleaned data as an Excel file by picking Export | Excel at the top right. Refine creates a file in the older (.xls) format that has the same name as our project.

This is just a quick look at how Refine can detect and correct errors. Refine can also split and merge values in columns. It even has its own scripting language called GREL, which allows you to write code for cleaning data.

Close Refine by closing your browser window, then close the window with the command line terminal. If you forget to do the latter, Refine will continue to run in the background.

EXTRACTING DATA FROM PDFS

Let's tackle one last challenge that you might face while gathering data: converting data tables inside PDF files into Excel files. Many government agencies publish data tables trapped inside these PDFs. In order to work with the tables, we must first liberate them. Open the PDF called fy2012_US_gov_net_cost.pdf. This one-page table, which documents the net operating costs of federal agencies, was extracted from the larger Financial Report of the U.S. Government for Fiscal Year 2012 that was released by the Department of the Treasury. We could try copying and pasting the data into Excel, but that would not preserve the column layout. So we'll instead need to use an online converter that will turn the PDF file into an Excel table. Cometdocs (at www.cometdocs.com) and Zamzar (at www.zamzar.com) are two proven services. Sometimes one does better than the other, depending on how the PDF is structured. So try both in real-life situations. (Note that these conversions will not work with PDFs that were created by scanning other documents. You would need to use OCR—or Optical Character Recognition—software to convert those.)

Go to Cometdocs and upload the PDF. Then drag the icon for the uploaded PDF onto the Convert button and pick Excel (xls) from the options list. Enter your email address and click Convert. Now check your email for a message from Cometdocs with the download link.

Open the Excel file and you'll see Cometdocs did a good job converting the PDF. The only glitch is that some agencies, like Health and Human Services, take up more than one line. We could clean that manually in the file if we needed to.

9	(Gain)/Loss					
10	from					
11		Gross		Earned		Changes in	Net
12	(In billions of dollars)	Cost		Revenue	Subtotal	Assumptions	Cost
13	Department of Health and Human						
14	Services		924.0	67.8	856.2	0.3	856.5
15	Social Security Administration		825.4	0.3	825.1	-	825.1
16	Department of Defense		784.7	56.0	728.7	70.4	799.1
17	Department of Veterans Affairs		213.6	4.1	209.5	149.3	358.8
18	Interest on Treasury Securities Held by						
19	the Public		245.4	-	245.4	-	245.4
20	Department of Agriculture		161.0	12.0	149.0	-	149.0
21	Office of Personnel Management		48.2	19.1	29.1	98.9	128.0
22	Department of Labor		107.3	-	107.3	-	107.3
23	Department of Transportation		79.0	0.8	78.2	-	78.2
24	Department of Housing and Urban						
25	Development		74.5	1.5	73.0	-	73.0
26	Department of Energy		60.8	4.3	56.5	-	56.5
27	Department of Homeland Security		58.2	9.9	48.3	0.4	48.7
28	Department of Education		62.7	20.0	42.7	-	42.7
29	Department of Justice		38.9	1.3	37.6	-	37.6

Source: Department of the Treasury.

Note: Data extracted from a PDF into an Excel file by Cometdocs.

For students and professionals working with data, cleaning is an essential part of the process. This chapter showed how you can get started to solve some of the most vexing problems with Excel, OpenRefine and online PDF converters.

Now that we know how to check and clean our data, we can move on to analyzing it to discover meaning.

ON YOUR OWN

Use OpenRefine to continue cleaning the campaign contributions data file. Clean the spellings of at least 10 cities. Use all of the algorithms available under clustering. Which one worked the best? Why?