# Prepare Your Data for Tableau

A Practical Guide to the Tableau Data Prep Tool

—

Tim Costello
Lori Blackshear

Apress®

# Prepare Your Data for Tableau

## A Practical Guide to the Tableau Data Prep Tool

**Tim Costello**
**Lori Blackshear**

Apress®

# *Prepare Your Data for Tableau: A Practical Guide to the Tableau Data Prep Tool*

Tim Costello
Euless, TX, USA

Lori Blackshear
Fort Worth, TX, USA

ISBN-13 (pbk): 978-1-4842-5496-7
ISBN-13 (electronic): 978-1-4842-5497-4
https://doi.org/10.1007/978-1-4842-5497-4



Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Susan McDermott
Development Editor: Laura Berendson
Coordinating Editor: Jessica Vakili

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail rights@apress.com, or visit http://www.apress.com/rights-permissions.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at http://www.apress.com/bulk-sales.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/978-1-4842-5496-7. For more detailed information, please visit http://www.apress.com/source-code.

Printed on acid-free paper

*Tim's Dedication:*

*I dedicate this book to my tribe.*

*To my Dad for his strength*

*To Nancy for patience*

*To Lori for right next steps*

*To Lisa for communication*

*To Kara for love*

*To Waverly for acceptance*

*To April, my muse*

*To Katie for determination and true grit*

*To Trevor for knowing how to pick a room*

*And finally, to my brother Chad. I wrote this book for you.*

*Lori's Dedication:*

*First and foremost, to my co-author Tim Costello, without whom I would never have had the courage to imagine a book and would have never had the guts to embark on our current Brindle Solutions LLC adventure.*

*My husband Dan Blackshear, who has provided the stability and support without which the risks would have been too great.*

*My children Jasmine, Travis, and Colby who likely believe in me more than I believe in myself.*

*Last, but not least, to my grandmother Virginia Joan Hardman who has told me my whole life that I should write a book. (I don't think this is the type she had in mind.)*

*Special Thanks*

*Our special thanks to Wendy Zhou and baby Vina. Wendy is an old and special friend. She is one of the brightest stars in the Tableau community and someone I respect and admire. Lori and I were thrilled that she chose to tech edit this book. What we didn't know was that she would be editing the final draft of the last pages from a hospital bed while she was in labor!*

*We welcome baby Vina to the world with all our love and best wishes.*

*Baby Vina: We look forward to meeting you in person and someday having you sign a copy of this book!*

# Table of Contents

# About the Authors

**Tim Costello** is a senior data architect focused on the data warehouse life cycle, including the design of complex ETL (Extract, Transform, Load) processes, data warehouse design, and visual analytics with Tableau. He has been actively involved with Tableau for almost 10 years. He founded the Dallas/Fort Worth Tableau user group. He has delivered hundreds of Tableau classes online and in person all over the United States and Canada.

When Tim isn't working with data, he is probably peddling his bicycle in circles around DFW airport in Dallas, Texas. He aspires to be a long-distance rider and enjoys going on rides ranging over several days and hundreds of miles at a time.

**Lori Blackshear** is a senior business process architect and expert at facilitating meaningful and productive communication between business and technology groups. She has deep experience in healthcare (human and veterinary), software development, and research and development in support of emergency services.

Lori served as a paramedic in Fort Worth, Texas, and Nashville, Tennessee, before shifting careers to helping people solve problems with data. When Lori isn't pondering business processes, she is active in the Fort Worth Civic Orchestra (violin) and the East Fort Worth Community Jazz band (tenor saxophone).

# About the Technical Reviewer

**Wendy Zhou** is a data visualization pioneer in ASB Bank who is passionate about leveraging data and emerging technology to tell fascinating stories. She tailors and facilitates Tableau training for her clients and colleagues, in a way that everyone can enjoy the peace and love of data with less drama.

She is a client-centric strategist with the heart of a revolutionary artist. She has been traveling with her young family while simultaneously working and solving data world problems for her clients from government to healthcare, Fortune 500 to rocket start-ups in Canada, New Zealand, and China.

She has a heart of giver, served as TUG leadership in Montreal, Canada, and Auckland, New Zealand, from 2013 to 2016.

She is a mother to three beautiful children and lives in Long White Cloud New Zealand. She loves cooking (but not baking!). She is also a YouTube-taught hairdresser (check out her FB page Wendy's Barber Shop) and enjoys many activities from crocheting to violin playing.

# Foreword

You've picked up a book on data, so that marks you as someone who had to take data and turn it into useful information.

Wouldn't it be wonderful if your data came to you complete, in the right format, and without errors? Unfortunately, the world isn't perfect, and neither is the data that you have at your disposal. You must concern yourself with the condition and completeness of the data. You also have to find out about the who, what, where, when, why, how, and how much questions related to the data before you can trust it. That requires some curiosity.

One of the most curious people that I've had the pleasure to meet is Tim Costello. I met Tim around 10 years ago after I gave a speech on data visualization to a group of data architects at SQL Saturday event. I was demonstrating a tool that made it fun and easy to transform data into understandable data visualizations. The audience was totally disinterested, except for Tim. After my talk he asked me myriad questions about the tool. Later at the speaker's dinner, Tim sat next to me and grilled me for another 2 hours about the tool, how I found it, what I had done with it, etc., etc. Tim has curiosity.

I haven't read Tim's book (yet) but I will. I think you can count on Tim. His curiosity about the world has led him to write this book and that's good enough for me to read it when I can get my hands on a copy.

What might some of the who, what, where, when, why, how, and how much questions be pertaining to your data?

1. Who: Who gave it to you, and who are you going to be doing the analysis for?

2. What: What do you or other people need to know? What is the condition of the data? What do you know about the data source that the data came from?
3. Where: Where is the data generated, and where is it going to be used?
4. When: When will you get the data (timeliness), when was the information generated, and when do you need to supply the information?
5. Why: Why is this data needed, why will it be useful (or not), why do you want to undertake the effort to make it useful and understandable, and why will people receive value from the information?
6. How: How will you get the data, how can you improve the data, and how can you automate its production and cleaning?
7. How much: How much time and effort will be required to process the data into useful information? How much effort will you be willing to put into the project of making this data into useful information.

The truth about all raw data is that it almost never comes in a form that is immediately useful. It's almost always the case that most of your time will be spent repairing, restructuring, and cleaning your raw data before you can even begin to turn it into useful information for people to use. Any tool, idea, or process that can help you do that reliably will help you be more effective at this crucial step in analysis and presentation.

This book may give you two to three good ideas that you can apply to your work. Maybe one of these ideas will shift your perspective or provide an insight that helps you deal with your raw data with a better understanding and with an improved process. Let Tim's curiosity be your guide.

—Dan Murray, Director of Strategic Innovation, InterWorks.

Dan has nearly 30 years of business experience as a CFO/CIO/COO/VP Planning/VP Operations and decided to join InterWorks to provide the best possible data visualization solutions available. He strives to give clients high-value solutions that can be implemented quickly, incrementally, and without significant risk. Drawing from his extensive experience, Dan wrote the acclaimed Tableau guidebook *Tableau Your Data!* His contributions to the Tableau community have earned him the title of Tableau Zen Master.

CHAPTER 1

# What Is ETL?

ETL is an acronym for Extract/Transform and Load. ETL is the engine that drives your visual analytics. Without ETL you would be limited to the shape (existing layout) of the data you start with. In many cases this is OK, but as you progress in visual analytics, you will find yourself wanting to combine data sources, combine fields within data sources, or maybe even break some individual fields out into multiple fields. You might want to replace some abbreviations or codes out into the values they represent. You will certainly want to know the "shape" of your data, so you can make judgments about its trustworthiness. These are all parts of ETL, and they are all things that the Tableau Data Prep tool excels at.

Let's start by breaking down the parts of a standard ETL operation and look at what each part might mean to us and why we should care.

## Extract

In the Extract phase, we are focused on connecting to and joining data. This is where we will explore the types of joins we can use and how each could affect the overall shape of our data. This might seem like a simple topic, but it will be one of the things we spend the most time on in this book. Joining your data correctly and understanding the implications of each choice in this step are critical and can have a huge impact on the data you ultimately export.


T. Costello and L. Blackshear, *Prepare Your Data for Tableau*,
https://doi.org/10.1007/978-1-4842-5497-4_1

## Transform

In this step we roll up our sleeves and get hands on with our data. First, we will do some simple auditing to see what we are working with, spot holes in our data, and make notes about things we want to change. Next, we begin to explore the concept of "shaping" our data. You will read a lot in this book about the shape of data and how it plays a critical role in creating an output that sets you up for success in Tableau. When I refer to the shape of data, I'm talking about a couple things. At a high level, we make decisions like do we want a few columns and a lot of rows or maybe we want a lot of columns and a few rows (to pivot or not to pivot our data). Then, we will look at cleaning our data. I think you will be impressed with the variety and power of tools we have to work with in this step. We will also look at combining fields to create a new field, separating the content of one field into two or more new fields, as well as aggregating and grouping fields over multiple records. We will explore applying business rules to control which parts of our data make it into the final output or in some cases we might even choose to create multiple outputs for different audiences based on the same core data. As you can see, there is a lot to cover in this phase. These choices will significantly affect how we can work with the data in Tableau, and we will explore each in depth in this section.

## Load

The final step might seem trivial, but there are some cool things we can do in this stage. We will ultimately end up with a complete data set ready for analysis in Tableau, yes … but we can also create intermediate data sources that could be extremely handy for testing our process and in some cases maybe even useful as data sources in their own right.

Now that we have a general idea of where we are going, let's dive right in and start playing with some data!

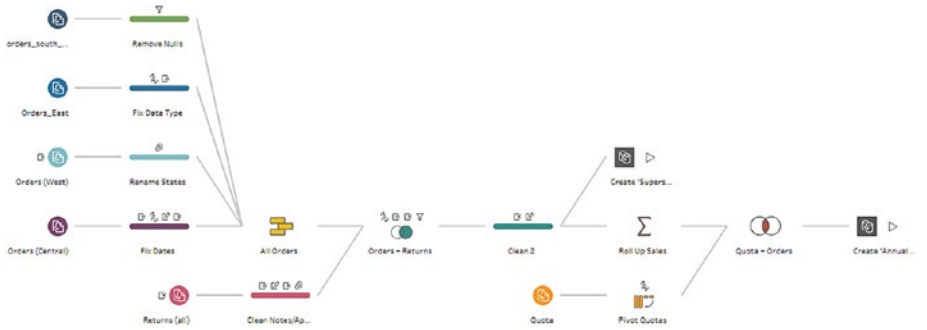



***Figure 1-1.*** *A taste of what's to come …*

CHAPTER 2

# About the Demo Data

My goal in this book is to walk through the ETL process from beginning to end, building a new data set from several public data sources I found in a morning of searching online. The first step in any ETL process is to get familiar with the data you will be working with. This chapter introduces you to the data that we will use throughout the rest of this book.

## ZTCA to Census Tract

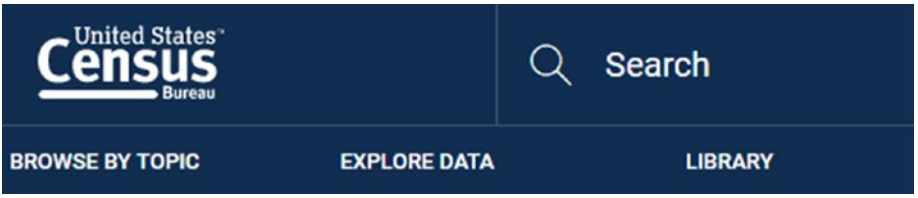



***Figure 2-1.*** *US Census*

This is an interesting data source (but then, I think all of them are interesting!). The purpose of this file is to give us a crosswalk (or data mapping) between ZTCA codes and Census Tract details.

To understand this file, we first need to understand its parts. ZTCA is an acronym for Zip Code Tabulation Area. This is a new code created by the Census in 2000 and continued in 2010. There is a semi-strong relationship between a ZTCA and a USPS Zip Code, but in the real world, this relationship is not to be trusted in all cases. The ZTCA was determined


T. Costello and L. Blackshear, *Prepare Your Data for Tableau*,
https://doi.org/10.1007/978-1-4842-5497-4_2

by looking at the majority of zip codes in use in each census block. A ZTCA is intended to overcome difficulties in precisely defining land area covered by each zip code. Zip codes can be tricky to work with because they can cross state, county, and census boundaries. We will use this file to connect data at the census tract level (average life expectancy by census tract) with school district details by ZTCA. In addition to the ZTCA code, you will find the 2010 state, county, and tract FIPS codes as well as a field called GEOID that concatenates these three values together (quite convenient for us as we will use GEOID to link to another file!).

For more information on ZTCA codes, check out the Wikipedia link here: https://en.wikipedia.org/wiki/ZIP_Code_Tabulation_Area.

The ZTCA file I include in this book was modified to include descriptive headers on row. The original source file is available here: www2.census.gov/geo/docs/maps-data/data/rel/zcta_tract_rel_10.txt.

## US_A.CSV


***Figure 2-2.*** *usaleep: Neighborhood Life Expectancy Project*

This file comes to us courtesy of the USALEEP program and the Center for Disease Control. USALEEP records estimates of life expectancy at birth for almost every census tract in the country. This is a really interesting program, and I encourage you to check out its page and download the original data (www.cdc.gov/nchs/nvss/usaleep/usaleep.html#life-expectancy).

# GEOCORR Education Data by State

For this data source, I generated 51 files (one for each state in the United States plus the District of Columbia) with school district and a few demographic fields for every ZTCA in the country. For those interested in downloading fresh data, I should mention that it IS possible to download one file that covers every ZTCA, but where is the fun in that?

If you want to download these files (or the one master file) yourself, you can see the settings I selected for configuration of the files in the following screenshots.

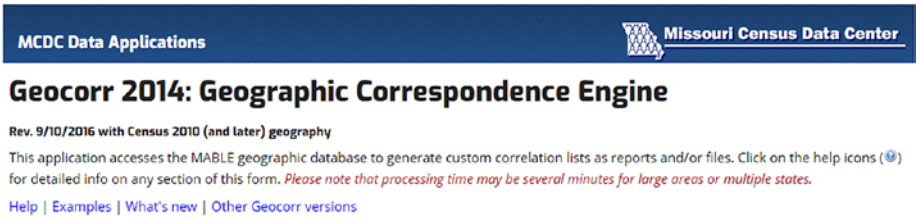



***Figure 2-3.*** *Geocorr download tool at the Missouri Census Data Center*

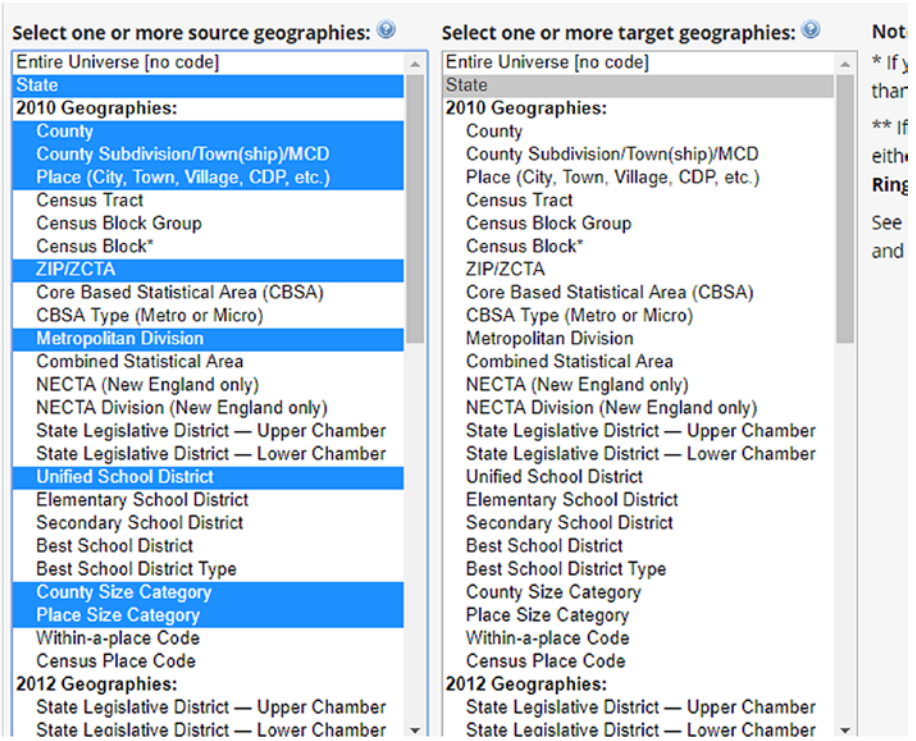



***Figure 2-4.*** *This screenshot shows the configurations I selected for output (for each state)*

# US Population Density and Unemployment by Zip Code


***Figure 2-5.*** *The SplitWise Blog*

These next data sources come to you from a somewhat unlikely source. I found them while researching this book and couldn't help but include them. They have some interesting detail, and they should prove useful for practicing joins to multiple data sources. The source? A blog for an app that helps people share expenses.

If you want to learn more, check out the SplitWise Blog, specifically the following link for an interesting discussion on working with census data:

https://blog.splitwise.com/2014/01/06/free-us-population-density-and-unemployment-rate-by-zip-code/.

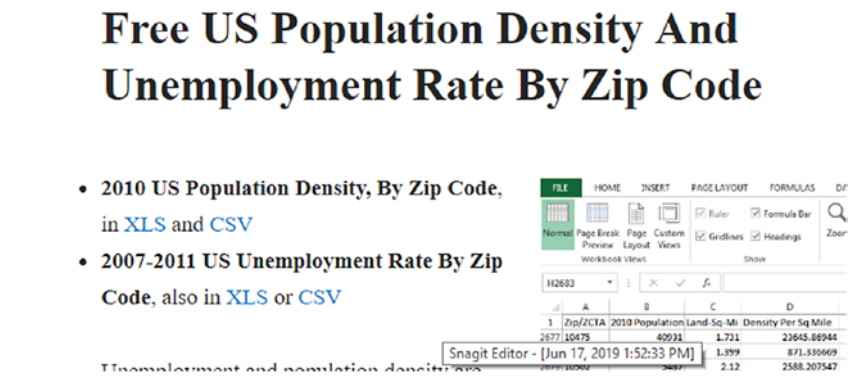



***Figure 2-6.*** *US Population Density and Unemployment Rate*

---

**Note** Not all the data sources described in this chapter made it into the exercises in this book, but I think they are interesting data to experiment with, so I included them in the source files for you to explore on your own.

---

# PART I

# Extract

We begin the journey from source data to fully prepped analytics-ready data with the Extract phase. In this section we will look at the types of data we can connect to, ways we can join different data sources, and some of the things we can do to (hopefully) avoid shooting ourselves in the foot by becoming more aware of common mistakes and how to avoid them.

We will begin by exploring the different files and server-based data sources we can connect to and talk through some of the implications of working with data sourced from different systems.

Next, we will look at a join type you might not have used before, the union join. I'm confident that after this chapter, you will find the UNION join to be a handy way out of a lot of common data integration scenarios.

Once we've conquered the union join, we will take a close look at the left, right, and inner and outer joins. We will discuss the strengths and limitations of each and begin auditing our data as we create joins to make sure we are getting the results we hope for.

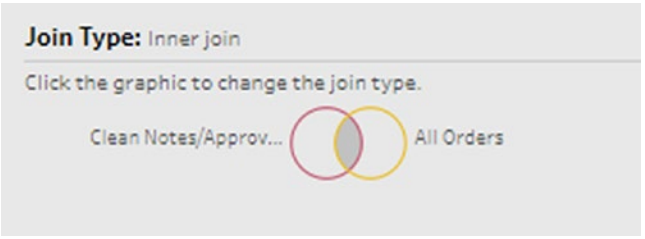



***Figure 1.*** *Join Type control in the Data Prep Builder*

# CHAPTER 3

# Connecting to Data

Tableau offers a lot of ways to connect to data, and I do mean a LOT. One of the challenges in writing this book is that nearly every month Tableau releases connections to new data sources. Today I count 6 file-based data connections and 36 server-based data sources! In addition to these built-in connections, you can connect to a wide variety of data sources not yet currently supported with native connections via an ODBC or a JDBC connection. With all these options, you should be able to connect to almost any kind of data you might encounter!

We will spend the rest of this book looking at file-based data sources (specifically text-based files), so let's start here with a look at the file-based data sources.

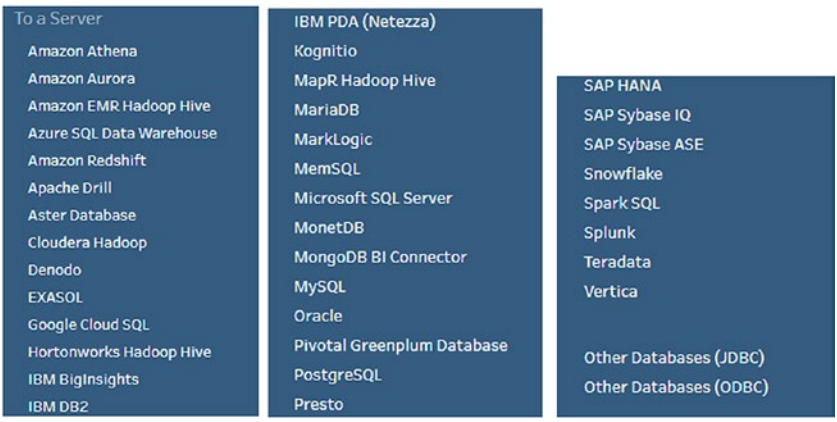



***Figure 3-1.*** *Native data sources in Data Prep Builder 2019.2.2*


T. Costello and L. Blackshear, *Prepare Your Data for Tableau*,
https://doi.org/10.1007/978-1-4842-5497-4_3

As you can see, there are a lot of connections to choose from. If you look closely, you'll find on premises data servers like Oracle, SQL Server, MySQL, PostgreSQL, Teradata, and Vertica. You'll also see cloud-based data sources well represented with sources like Amazon Athena/Aurora/EMR Hadoop Hive/Redshift and Azure SQL Data Warehouse. Last but not least, there is a nice variety of Hadoop-based data connections including Cloudera Hadoop and Hortonworks Hadoop Hive.

Connecting to most of the server-based data sources is fairly straightforward. For the most part, all you'll need is the name of the server you want to connect to and login credentials to connect to that server. In the following example, I will connect to a local instance of SQL Server utilizing Windows Authentication. Connecting to most of the data sources in the server-based list of built-in connections will be a lot like this exercise.

# Working with Server-Based Data Sources

## Connecting to SQL Server

1. From the main screen in Data Prep Builder, click the plus symbol next to "Connections"

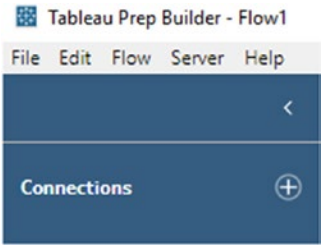



***Figure 3-2.*** *New connection*

2. Scroll down the list and select Microsoft SQL Server


***Figure 3-3.*** *Select a server-based data source*

3. Enter the name of your server and (optionally) the name of your database

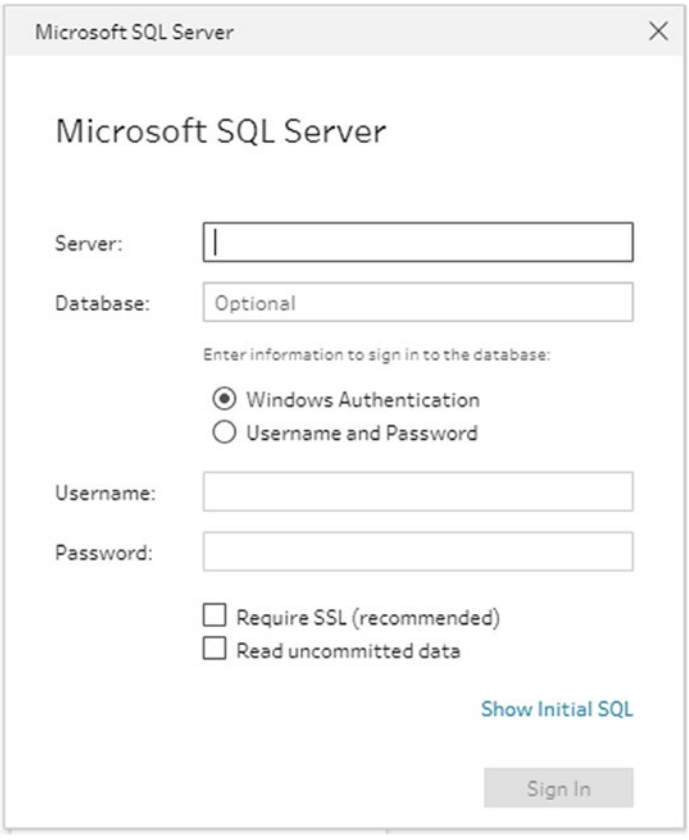



***Figure 3-4.*** *Configure connection to SQL Server*

4. If you are using Windows Authentication to connect to SQL Server, you can click Sign In now. If you are using a Username and Password, you need to enter them before you can click Sign In.

5. If you didn't enter a database name in the connection screen (I usually don't), you will need to pick a database next. Click the drop-down arrow next to Select Database to pick from a list of databases that you have access to.

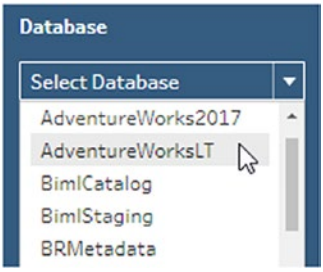



***Figure 3-5.*** *Select a database*

That's all there is to connecting to a SQL Server database! I've connected to many of the native connections in the Data Prep Builder, and for the most part they are very similar.

## Initial SQL (Running SQL on Connection)

One interesting feature of many of these native connections is the ability to execute an initial SQL statement when the connection is opened. To configure this (on connections that support it), click the Show Initial SQL link bottom center of the connection dialog. (See Figure 3-4.)

This opens the Initial SQL configuration. From here you can enter a SQL statement that will be executed every time the connection is opened. This can be handy if you want to create objects on SQL Server prior to starting

your data flow, execute stored procedures, change security context in SQL Server … the sky's the limit. Well, the sky and the permissions granted to your SQL Server login. I recommend using extreme caution when exploring this feature of the Data Prep builder. As they say, with great power comes great responsibility. Let's not anger your DBA. She is someone we want to keep happy, and executing unwise SQL on pre-execute might not end well.

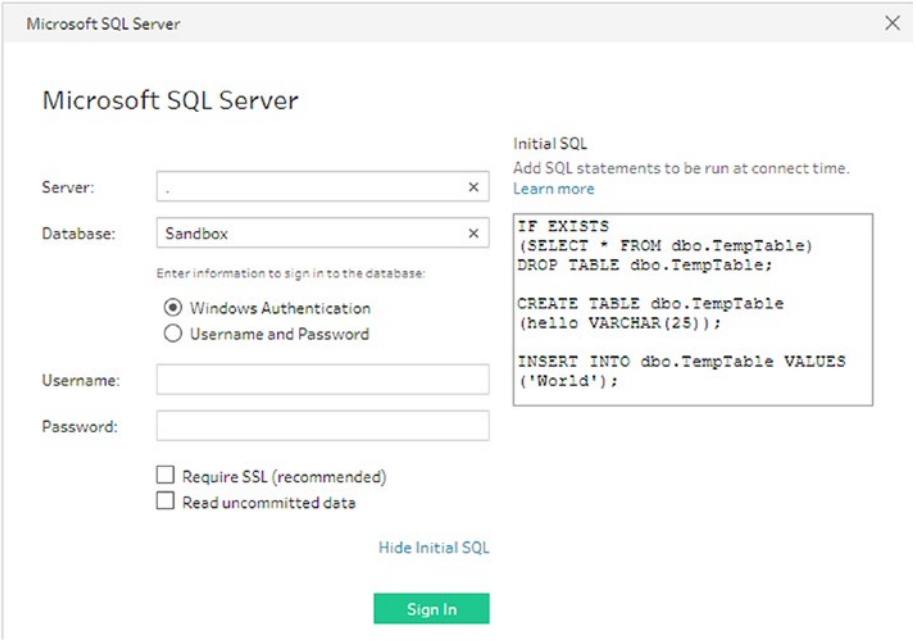



***Figure 3-6.*** *Initial SQL configuration*

In Figure 3-6 you will see an initial SQL statement configured that will check for the existence of a table called TempTable in the SandBox database. If it finds that table, it will delete it. Next it will create a new table called TempTable and insert into it one value. The results are displayed in Figure 3-7.


***Figure 3-7.*** *Results of Initial SQL execution on SQL Server*

Here's the SQL statement I used for that example for those of you following along at home.

***Listing 3-1.*** Check for table, drop and create, insert into table from Initial SQL

```
IF EXISTS (SELECT * FROM dbo.TempTable)
      DROP TABLE dbo.TempTable;

CREATE TABLE dbo.TempTable (hello VARCHAR(25));
INSERT INTO dbo.TempTable VALUES ('World');
```

And here is a SQL statement pulled from the Data Prep Builders online help that can change the security context for this connection (assuming the credentials you provided on login allow this).

***Listing 3-2.*** Execute AS statement for Initial SQL

```
EXECUTE AS USER = [TableauServerUser] WITH NO REVERT;
```

In Listing 3-2 you will need to change the value [TableauServerUser] to the name of a named user in SQL Server that has the permission level you want to use for this connection. I STRONGLY recommend you restrict the named user account to read only. For the most part (with only extremely rare exceptions), you will never want to change existing data stored in SQL Server from the Data Prep Builder. Please refer to my earlier warnings against committing shenanigans in the database, and consult your DBA if you have any questions.

# Working with Tableau Data Extracts

Working with Tableau Extracts (tde and Hyper files) is something of a good news/bad news story. The bad news (I'm a DBA, I always start with the bad news) is that when you work with an extract, the Data Prep Builder will unpack the Extract or expand the Hyper file. This can result in the sudden requirement of a LOT of temp space and system resources. The following example is pulled straight from the Data Prep Builders online help.

(https://onlinehelp.tableau.com/current/prep/en-us/prep_connect.htm)

… an extract file with 18 columns and 1.2 million rows that is 360MB (8.5GB uncompressed) may need up to 32GB RAM, 16-core, and 500GB of disk space available to support the file when it is unzipped.

This might seem like an extreme example, but I've seen bigger extracts in the wild. The takeaway here is if you are working with big extracts, you will need big hardware.

The good news is that you now have the ability to get data OUT of your Extract (or Hyper) files thanks to the Data Prep Builder. In the past, there was simply no way to get data out of an extract. It was strictly a one-way street. Data goes in, data does not come out (except to be used within Tableau Builder). Now you can simply add an output step and you are ready to export data from your .tde or .hyper file. We'll go into more detail on the output step later in the book.

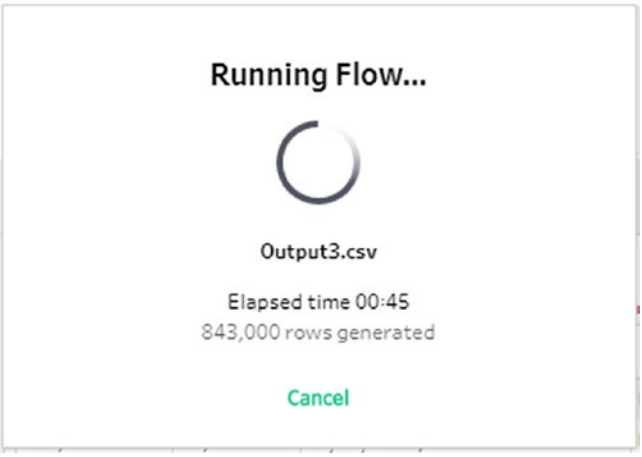



***Figure 3-8.*** *Output from .tde or Hyper file!*

# Working with File-Based Data Sources

The Data Prep Builder lets you connect to all the common file-based data sources. In the following section, we will take a quick look at what's involved in connecting to a Microsoft Access database, Microsoft Excel spreadsheet, PDF files, and text files.

## Connecting to Microsoft Access

Connecting to a Microsoft Access database is as easy as browsing to the file and selecting it. You might have to enter a logon ID and Password if the Access database you are connecting to requires them. The one thing I'd like to point out here is that this is a file-based connection, so that file could be in use by someone else when you try to connect to it. This isn't

a problem with server-based data sources, but with any file-based data source, you run the risk of being blocked from accessing the file if it is currently in use by another user.

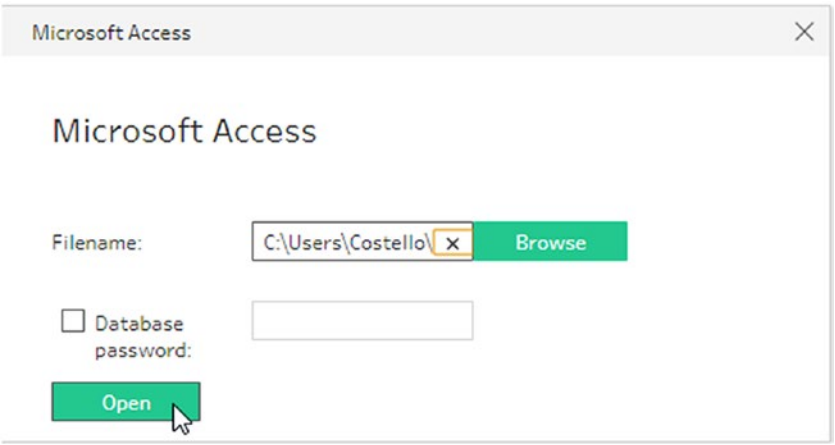



***Figure 3-9.*** *Connecting to a Microsoft Access file*

Once you are connected to your Microsoft Access database, you will be offered a list of tables and views available in that database. From here this connection will behave just like any other connection. Note: There is no option to execute initial SQL statements for this connection.

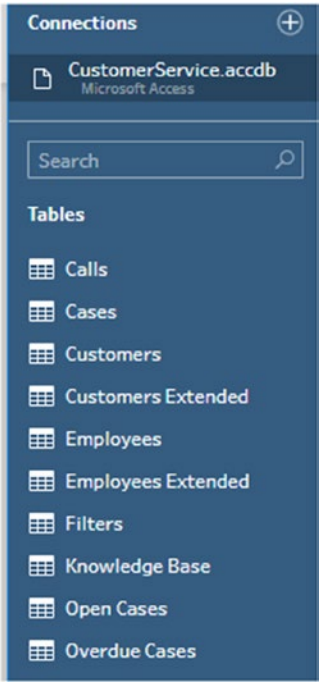



***Figure 3-10.*** *Select a table (or view) from a Microsoft Access database*

## Connecting to Microsoft Excel

I often say that Microsoft Excel is the most used database on the planet. This might sound like an exaggeration, but I stand by it. In my career I can't recall three projects I've worked on that didn't include data from an Excel file. I strongly suspect you will be using this native connection frequently.

Fortunately, it's dead easy. Simply browse to your Excel file and you are off to the races! In the previous section, I warned that the Data Prep Builder can complain to you if you try to connect to an Access database that is in use at the time you want to connect to it. I have not had this experience when connecting to Excel files.

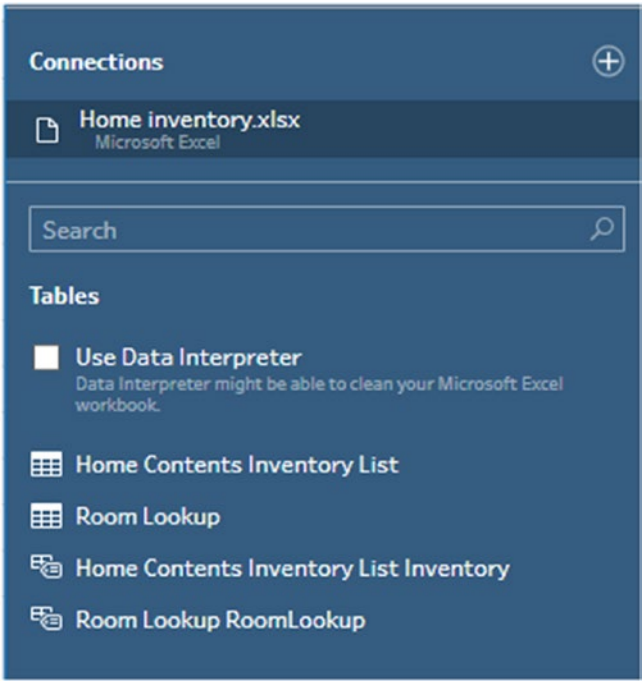



***Figure 3-11.*** *Select a "table" in a Microsoft Excel file*

The thing I LOVE about Excel is that it acts as if it were a database. When you connect, you will be presented with a list of "Tables." These are the worksheets in your Excel workbook. Notice in Figure 3-11 that there are four objects in the Tables list. Two of them (Home Contents Inventory List and Room Lookup) are worksheets in the workbook we selected. The last two items (Home Contents Inventory List Inventory and Room LookupLookup) are named ranges. OK … I think that's cool. This gives us the ability to design tables within a workbook, and within those tables we can designate named ranges that themselves act as tables. That creates a LOT of opportunities for working with data inside and outside of the Tableau ecosystem.

## Connecting to PDF Files

Connecting to PDF files is one of the most interesting, and sometimes frustrating, things you might do with the Data Prep Builder. Interesting because most business users have a wide variety of reference files saved as PDF. Bringing that data into your analytics can open some fun analytic opportunities. Frustrating because connecting to and using PDF files tends to be a tedious process that involves a lot of work in the cleaning step. The more you work with PDF files, the more familiar you will become with the split tools, the data cleanup tools in general, and the UNION join specifically. We will explore these steps in much greater detail as we progress through this book.

## Connecting to Text Files

I left this section for last because we will basically spend the rest of this book learning the things we can do and how to do them with text files. To set the stage, a text file might be any file that you can read with a standard text editor in Windows or on a Mac. I am a Windows user and my text editor of choice is Notepad++. You might have a different favorite text editor, but that's ok. Any file you can open and review in a text editor will be accessible from the text file connection. Examples of common text extensions (in Windows) are .txt, .csv, .tab, and .tsv.

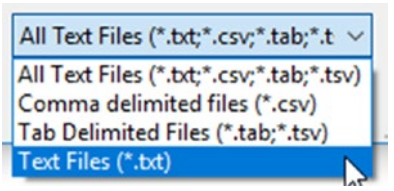



***Figure 3-12.*** *Text file types*

If you have a text file that is a different extension (maybe .dat or .rpt), you can try to open it by typing an asterisk followed by a period and the file extension you want. The Data Prep Builder will then present you with a list of all files in your selected directory with that extension (see Figure 3-13). I say try, because the Data Prep Builder won't necessarily open a file just because it shows it to you in this dialog. In Figure 3-13 I have asked it for a list of all .bak files. These are SQL Server backup files. Most definitely NOT text files. The best-case scenario in this case is Tableau refuses to open the file, the worst case is the file you selected could be corrupted and the Data Prep Builder will most likely crash. That is, as we say in Texas, No Bueno.

# Summary

We've covered a lot of ground in this section. At the end of the day, connecting to data is straightforward. For the most part, if you know the location of a file-based data source, you will be able to connect to it. If you know the Server Name and have valid credentials for a server-based data source, you should have no trouble connecting.

We spent some time here looking at how to execute initial SQL statements or SQL statements that will be executed on a server-based data source when the connection opens. These initial SQL statements give you great opportunities (for good and for evil). I urge you to learn more about them and discuss your plans to use them with your DBA. Once you've done that legwork, I think you will be pleased at the power this feature puts in your hands.

Next, we reviewed connecting to Extracts (and Hyper files) as data sources. The two main takeaways here are that connecting to a large extract could require a HUGE amount of system resources and that the Data Prep Builder makes it easy to get data OUT of an Extract. This is a feature I've had many, many clients ask for. I'm excited that it's now built into Data Prep.

We close out this section with a brief overview of connecting to the most common file-based data sources. The rest of this book will focus on ETL as we explore connecting to data (*Extract*), cleaning up and changing data in the pipeline (*Transform*), and saving data for analytics (*Load*) with data from multiple text files.

CHAPTER 4

# UNION Joins

In this chapter we will dive deep into the workings of the UNION join in the Data Prep Builder. To set the stage, let's talk about what a UNION join does. A very common data scenario might involve receiving multiple text files from multiple locations within your company. Each text file contains the same columns of data, but the rows themselves will be unique. Our goal in this scenario would be to somehow combine all these individual files into one master file that contains data for every location. You might have worked through this scenario before. The brute force solution would be to open each file and copy/paste its contents into one master file. This is a reasonable solution when you are working with a small handful of files or maybe you are only doing this one time. It's less reasonable when you have hundreds of files that you receive multiple times a week. In those cases, the UNION join will save the day!

Let's kick things off with an example. In this exercise we will combine the contents of five files so that we can work with that data as if it were all saved in one file.

## Exercise 4.1: Union Join

1. Open the Data Prep Builder
2. Click the plus symbol next to Connections to open a new connection


T. Costello and L. Blackshear, *Prepare Your Data for Tableau*,
https://doi.org/10.1007/978-1-4842-5497-4_4

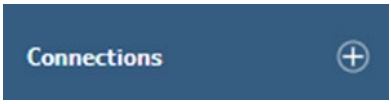



***Figure 4-1.*** *Click for new connection*

3. Select Text file

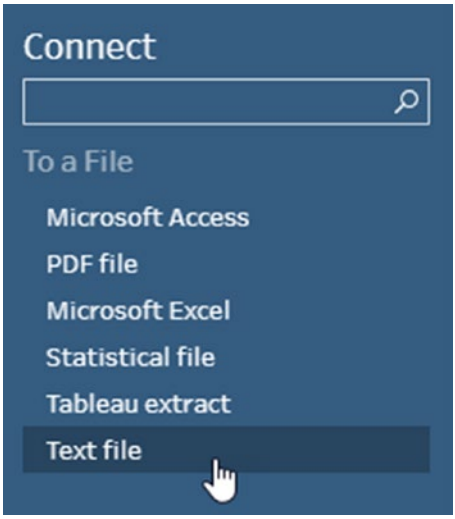



***Figure 4-2.*** *Select a text file connection*

4. Click and drag your mouse over all the files in the Exercise 4.1 folder to select them

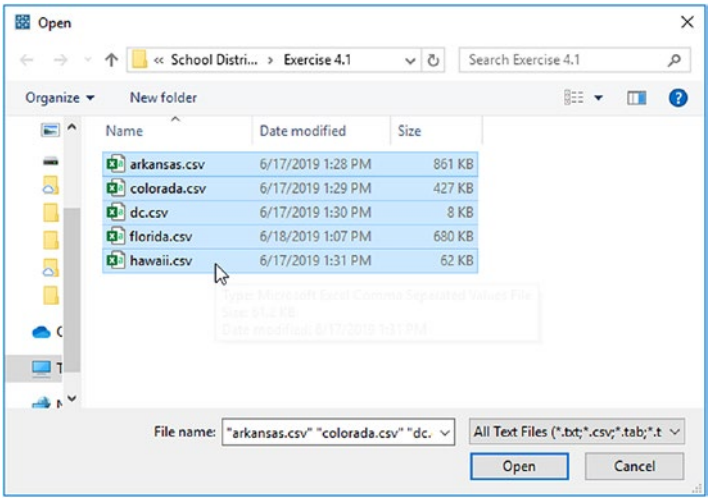



***Figure 4-3.*** *Click and drag mouse to select all files*

5. Click Open to connect to the files you've selected

The Data Prep Builder will open a new data flow and drop the five files you selected onto the design canvas for you. It will also open the input review section. This is an area where we will look at in a moment. For now, close it by clicking any white space in the area near the steps for your connections. I've marked an area you might click in the following screenshot.

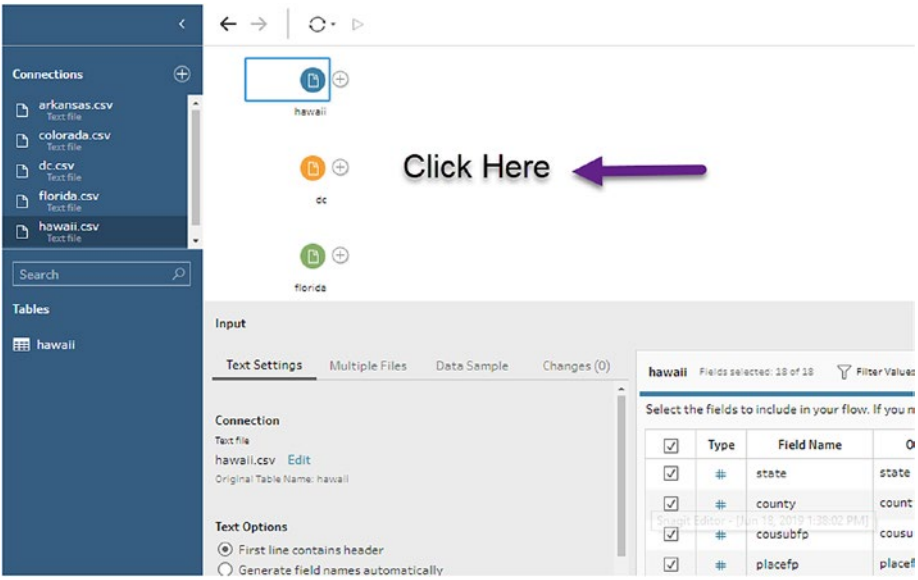



***Figure 4-4.*** *Click in a blank section of the design surface to close review section*

---

**Note** You can open the review section by clicking a connection step. You can close it again by clicking in white space near (not on) a connection step. Go ahead and give this a try. You will frequently toggle this review section open and closed as we work through this book.

---

6. Your design canvas should now look something like Figure 4-5. It's OK if the steps in my screenshot are in a different order and different color from what's on your screen.

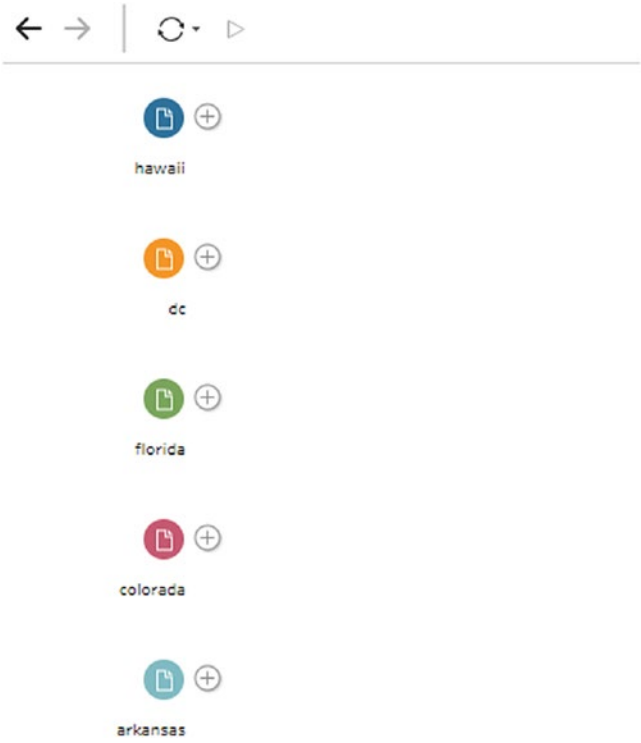



***Figure 4-5.*** *Design canvas ready for UNION join*

7. Click the plus symbol next to any of the steps for your text file connections and select Add Union. (In the preceding screenshot, I selected Florida because it's in the middle. You can pick any of the connections and this exercise will work.)


***Figure 4-6.*** *A Union has been added to the Florida connection*

8. Click and drag any connection step over the Union 1 step. When you do this, you will see some drop zones appear as light orange boxes around the Union 1 step. Drop the connection step you are dragging onto the box marked Add.


***Figure 4-7.*** *Dragging another connection onto the "Add" drop zone of the Union step*

9. You should now have lines connecting two of the text file connections to the Union step (your screen might look slightly different but should be similar to Figure 4-8)

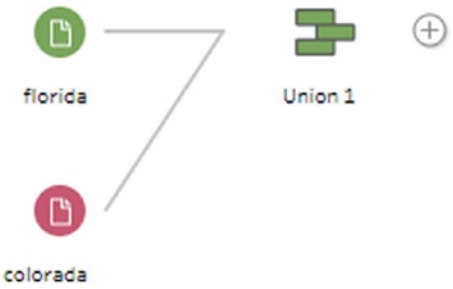



***Figure 4-8.*** *Two text file connections joined to a Union step*

10. Repeat step 8 (drag a connection icon and drop it in the Add drop zone of Union 1) for each of the remaining connections.

At this point you should have five text connections with lines connecting each to the Union 1 step we created on step 7.

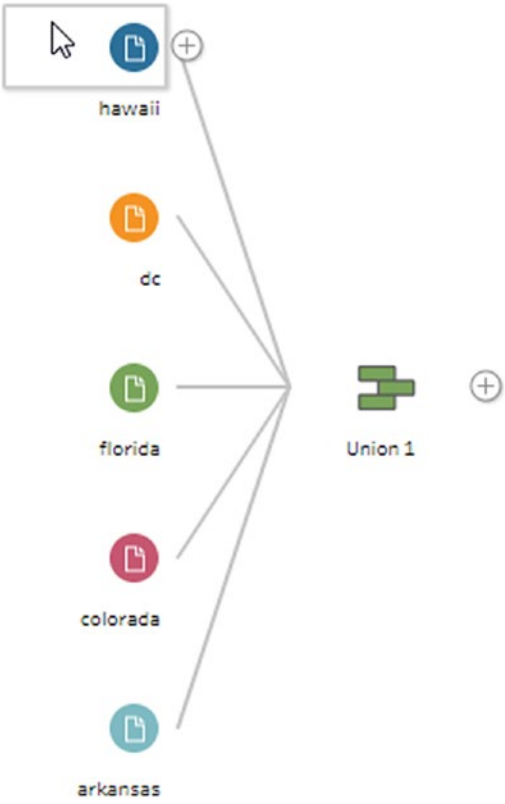



***Figure 4-9.*** *Five connections joined to the same Union step*

When you've completed this exercise, you will probably be really comfortable with closing the review section between each change. Every time you click an icon (or more correctly a step), you will move the focus to that step, and in response the review section for that step will open to give you an idea what is going on in that step and let you make changes.

Click the step for the Hawaii connection. When the review section opens, scroll down and review your options in the Text Options section of the review pane (far left). Here you will see places where you can configure if your input file has a column header on the first line or if the Data Prep Builder should generate headers for you. You can also change the delimiter

for this input, the text qualifier, the character set, and the locale. You will probably only rarely change any of these settings, still it's good to know where they are.

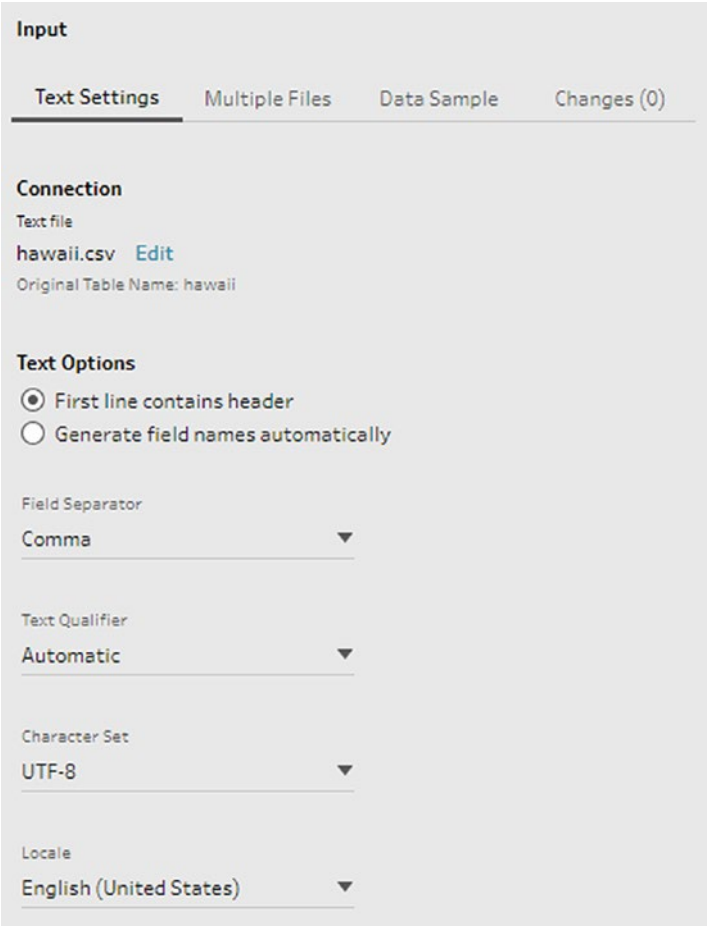



***Figure 4-10.*** *Text settings for the Hawaii Input step*

Next let's toggle over to the Data Sample tab. This section can be very helpful when working with large input files. By default, the Data Prep Builder will try to make a best guess as to the number of rows it should sample. In most cases, this best guess is reasonable. In some cases,

you might want to adjust the sample size. If the sample is too small, you might not get a real feel for the nature of your data in the preview panes, and in some cases the Builder might make bad guesses as to data types or poor suggestions for things it can fix for you. We'll get to those topics soon, for the moment it's just important to know that this is the place where you can change how the data sample is generated and how many rows of data you want the Builder to sample. Note: Sometimes, with very large data sets, you might even want to reduce the sample size to increase performance.

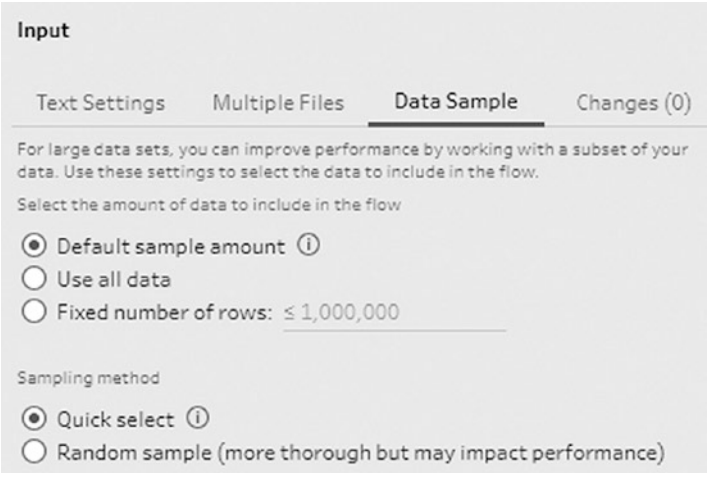
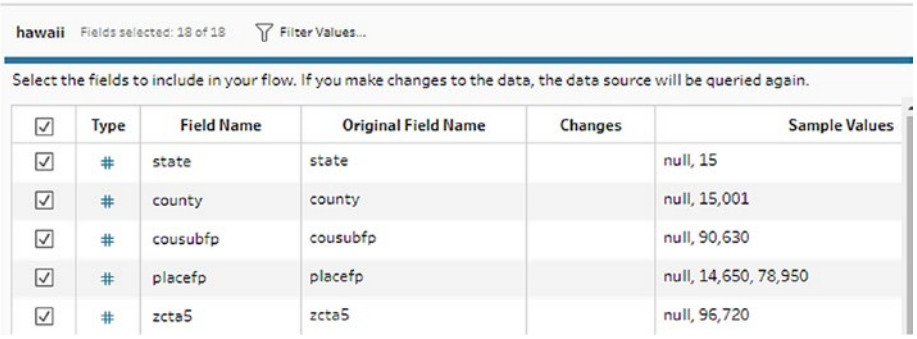



***Figure 4-11.*** *Data Sample pane of the Input review screen for a text file step*

# The Data Preview Pane

Next, let's move a little more to the right on the screen to review the data preview for this text file connection step. In Figure 4-12 you will see a part of this preview for the Hawaii connection.


hawaii Fields selected: 18 of 18 Filter Values...

Select the fields to include in your flow. If you make changes to the data, the data source will be queried again.

| ☑ | Type | Field Name | Original Field Name | Changes | Sample Values |
|---|---|---|---|---|---|
| ☑ | # | state | state | | null, 15 |
| ☑ | # | county | county | | null, 15,001 |
| ☑ | # | cousubfp | cousubfp | | null, 90,630 |
| ☑ | # | placefp | placefp | | null, 14,650, 78,950 |
| ☑ | # | zcta5 | zcta5 | | null, 96,720 |

***Figure 4-12.*** *Preview of text file connection step*

There are a lot of neat things you can do in this preview. Starting with the checkboxes on the left side of the preview pane. By default, every field in the connection will come into your data flow. Uncheck a box next to a field and that field will not come into your data flow. Click the icon in the Type column, and you can change the data type of the column you are working with. In most cases, Builder does a good job of guessing data types, but it's still worthwhile to review each field and make sure you agree with the data type associated it.

# Data Types

I think it's a good idea to stop here and look at how the data types are labeled. Some of these labels might be unfamiliar. We'll walk through them together.

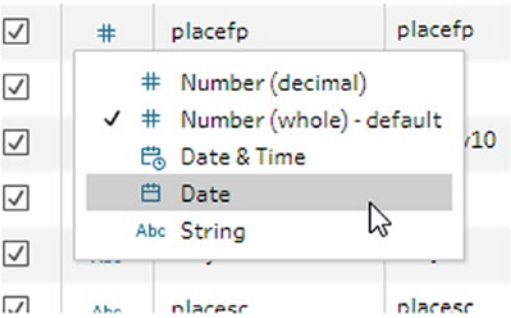



***Figure 4-13.*** *Tableau Data Prep data types*

The first two data types should be easy to grasp. Numbers with a decimal and numbers without. This is a place where you could potentially run into a data sample size problem. Imagine a scenario where you have a text file with pricing details. For the first 5m rows, pricing is all in whole numbers (no decimals). Then in the last 500 records, pricing includes decimal values. In this case, the sample size and type of sample might not catch this change, and the field might be set to Number (whole) rather than Number(decimal). If something like this happens, you could update your sample size to include all records in your data source, or you might set the sample method to pull random records. Alternately, you could just manually set the data type to Number(decimal). As with many powerful tools, there is often more than one way to solve a problem.

Date & Time and Date are like Number (whole) and Number (decimal) in that they capture different levels of detail. If you know you won't be capturing data at the level of hours, minutes, or seconds, you are safe in selecting Date (without time).

The last data type is String. A String is simply a string of alpha-numeric characters. These characters could be letters, numbers, or symbols. The rule here is that if you think you will want to use a field as a label and you don't plan on using that field in any calculations, save it as a String. For example, you might have employee IDs in the format of 5-digit numbers. Prep will usually recognize ID fields based on their field names, but

sometimes it misses. If you see a field that you know is an ID, but it is stored as a Number (whole), you should probably change it to a String. This will make it a dimension when it gets into Tableau and will make using that field as a label or header much easier.

# Changes

I am a big fan of the way Prep tracks changes. Let's try a little exercise to see what I'm talking about. Click any of the field names and make a change. Add a character, rename the field, anything you like. Now change the data type for the same field. You can pick any data type to change to. You should now see a small icon for each change that you made in the Change column. If you hover over that icon, you can see exactly how this column was changed. Now right-click each icon that you see in the Change column and select Edit. This will take you to the property you changed, and you can change it again. Finally, right-click any remaining Change icons and select Remove. This will undo your change.

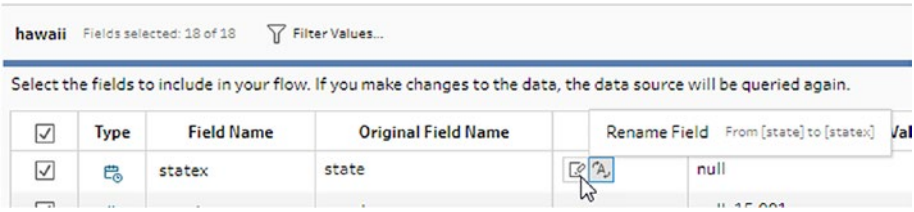



***Figure 4-14.*** *Change tracking in the review pane*

# Reviewing the Union Step

Up to this point we have focused on reviewing an Input step. The focus of this chapter is the Union step, so let's take a look at what we can see when we review that step.

If you still have a review pane for an Input step open, close it by clicking in white space on the design canvas. Next click the Union 1 step to review its configuration.

Scroll down in the Settings tab (far left side of the Review pane) and you will find the mismatched fields section. In a perfect world, this section won't even be visible. What we are looking at here is the way our five input files line up with each other. If every input file has exactly the same field name, they will all line up perfectly and you won't see a mismatched fields section. Unfortunately, we don't live in a perfect world. Fortunately, Prep anticipated this (extremely common) problem and gives us some easy tools to fix it.

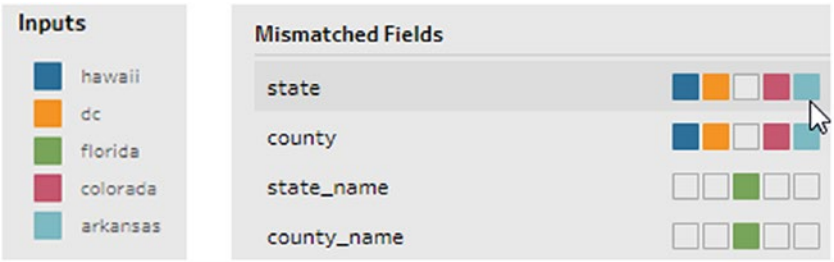



***Figure 4-15.*** *Mismatched fields in a Union step with Inputs color key*

A glance at the mismatched fields section will show you that we have a problem. Notice how there is a hole (unfilled boxes) in the middle column for state and county and the only filled boxes are for the state_name and county_name fields. Someone changed our field names in one or more of our input files. Drat! I hate when that happens.

Fortunately, this is an easy fix. At the top of the Union results pane, you will find a checkbox for Show Only Mismatched Fields. Click this box and all the matched fields will be momentarily hidden, so we can focus on only the mismatched fields.


***Figure 4-16.*** *Union Results showing only mismatched fields*

With matched fields hidden, we now see a list of mismatched fields. If you look at the color key to the left, you can quickly determine that our problem is isolated to one input file, Florida. What we want to do now is tell Prep which fields we want to combine and which field name in each combination we want to keep. To do this, click and drag the field name you don't want to keep and drop it on top of the field name you want to combine it with. This maps the two field names to one field and tells Prep what to call that new field. In this case we will click and drag the state_name field and drop it on top of the state field. Next, we will click and drag the county_name field and drop it on top of the county field. If that worked out, we should have no fields left visible in the review pane. This is because we now have no mismatched fields. If you uncheck Show only mismatched fields, all the fields will come back into view. Scroll all the way to the right in the field list, and at the end you will find the state and county fields, newly merged and now fully populated. Notice that these fields now have five color bands under the field name. This indicates that all five source steps are represented in this field.

Finally, scroll all the way back to the right and look at the first name in the field list. This is a field name that doesn't exist in any of the source steps we are working with. Prep gives us this field to record the name of the source file each record was sourced from. This kind of metadata (data about data) will prove extremely useful when we are auditing or troubleshooting our work in Prep.

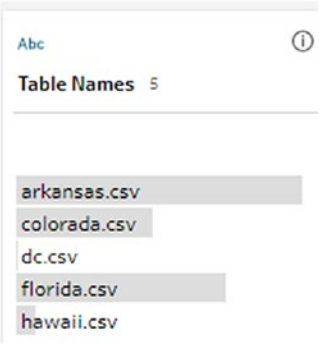



***Figure 4-17.*** *Table names for tables that provide input to Union step*

# Union Join the Easy Way

Now that we've spent a bit of time getting to know the Union step, let's look at another way to get the same result. An easier way.

In many cases, when you want to union data together, you will have all the data that should be brought together in the same folder, the same spreadsheet (even if on different worksheets), or tables in the same database. In cases like this, there is an easier way to get a Union of your data. A way that requires a lot fewer manual steps.

In Exercise 4.2, we will work through an example of using a wildcard union in a text file connection to bring together the content of 51 text files.

# Exercise 4.2

1. In the Data Prep Builder, click Connections and then select Text file
2. Browse to the location you have saved your exercise files associated with this book
3. Select the folder for Exercise 4.1

4. Select any text file (for this example I selected alabama.csv)
5. Click the tab for Multiple Files in the review pane

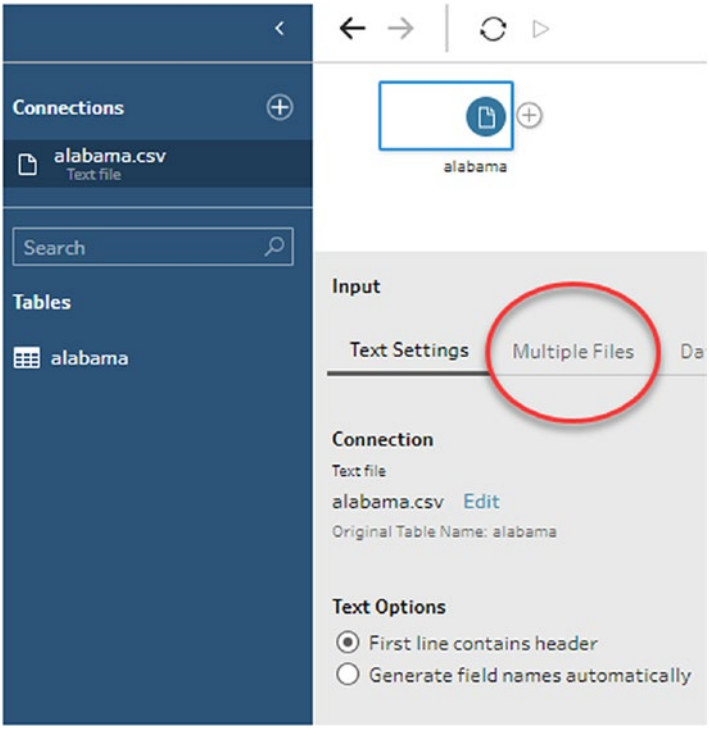



***Figure 4-18.*** *Click Multiple Files in the review pane*

6. Before continuing, look at the metadata review section of the review pane (the area with details like field names, data types). Make a note of the number of fields in the file you are starting with.

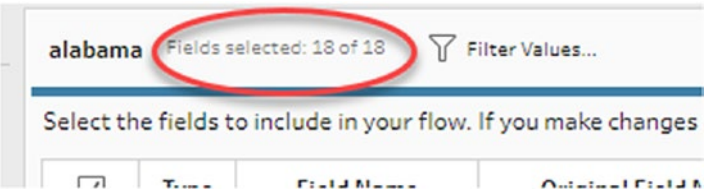



***Figure 4-19.*** *Note the number of fields you start with before doing a wildcard union*

7. Click Wildcard union

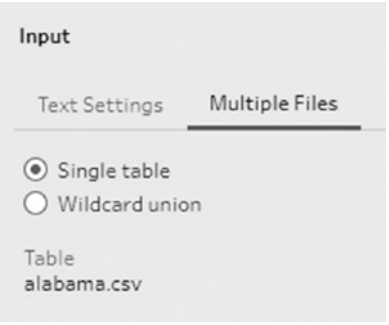



***Figure 4-20.*** *Select Wildcard union*

8. Verify that all the files you want included in the wildcard union are in the list

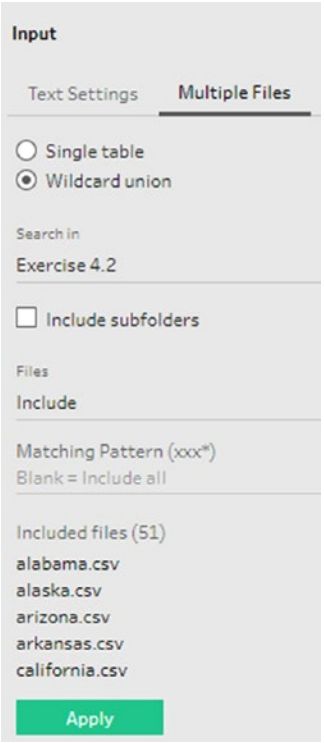



***Figure 4-21.*** *Validate included files (number and file names)*

9. Click Apply

That was easy! In just a few steps, we created a connection that contains the results of a wildcard union of 51 source files!

Before we move on, I want to call out a few things worth taking note of. About halfway down the Multiple Files tab, you will see a checkbox for Include subfolders. Checking this will bring the contents of the folder you have selected and all of its subfolders into the wildcard union results. This is extremely powerful; you could potentially process dozens or hundreds of folders with a single click!

Also notice, just under the word Include a place where you can enter a matching pattern. If you have a scenario where you have a lot of files comingled in the same folder tree, but each grouping of files has a distinct

pattern in its name, you could use this section to focus on just those files. In the example we are working on, go to the text entry area just below the label Matching Pattern (xxx∗) and enter m∗.

If you do this, the list of files to be included in the wildcard union will be reset to only those files that begin with m.

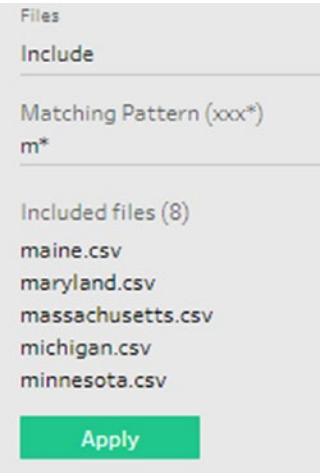



***Figure 4-22.*** *Restricting files with a matching pattern*

I think you will agree that when you have a lot of similar files or spreadsheets, it is much easier to bring them together with a wildcard union.

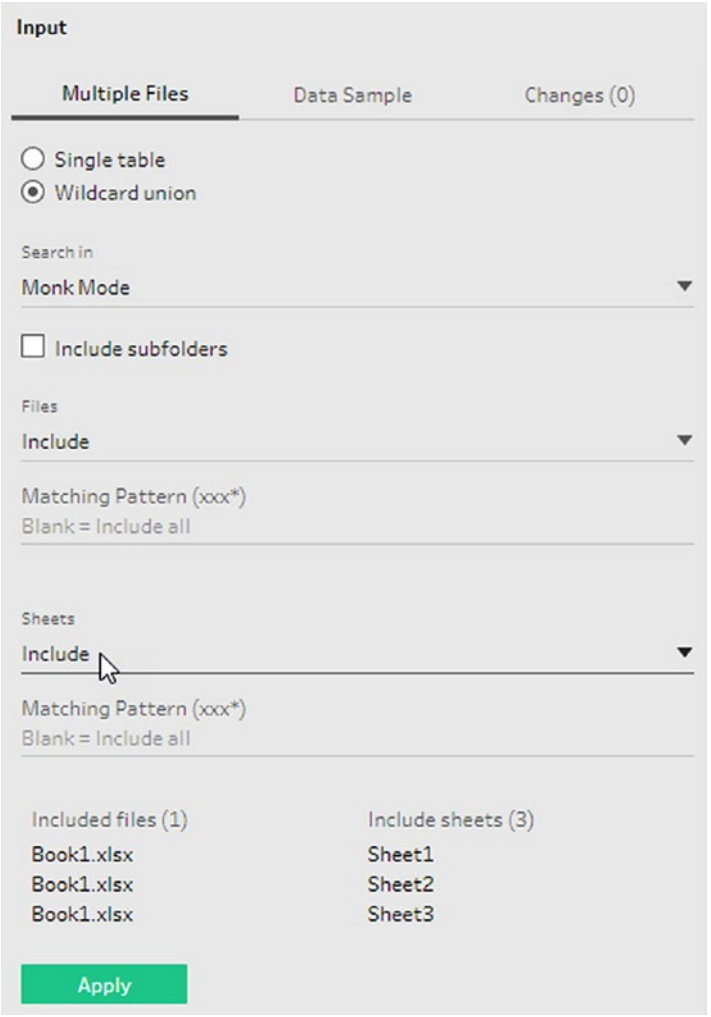



***Figure 4-23.*** *A wildcard union with an Excel data source lets you include (and filter) multiple Excel files in the same folder AND multiple worksheets in each workbook!*

The wildcard union seems pretty great, but there is a catch. If your files don't all have exactly the same structure (column names), they won't line up correctly for the wildcard union. Remember when I asked you to look at the number of fields in step 6 of Exercise 4.2? The observant reader will

have noted that our field count increased by three. We expected our field count to go up by one (we always get a new field that contains the name of the source file). The additional two fields we picked up were from one or more files that have field names that differ from the source file we started with. In this case we started with alabama.csv. Somewhere in our 51 files, there is at least one file that has two additional fields. Let's get to the bottom of this mystery.

The following exercise picks up from where you left off in Exercise 4.2. If you set a Matching Pattern in the Multiple Files tab, clear it and click Apply again before continuing.

# Exercise 4.3

1. Click in a blank space on the design canvas to close the review pane
2. Right-click the label alabama
3. Click Rename Step and rename it to All States
4. Click the plus symbol next to All States and select Add Step

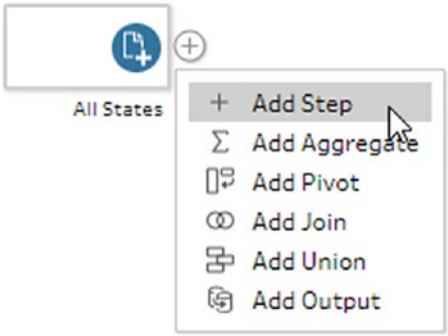



***Figure 4-24.*** *Add Step*

5. You should now have a review pane open on your screen with details for the Clean step you just added. Scroll to the far right of the display of columns.

At the end of the display of columns, you should see the File Paths column. This is the column Prep gave us to record the file name each record was sourced from. Next to File Paths, you should see state_name and county_name.

Click the number 12 in state_name. You should see something like what is in Figure 4-25. Only florida.csv is selected in the File Paths column. This tells us that the values we selected only exist in the florida.csv file. Now click null (above 12 in the same column). Now you will see every file EXCEPT florida.csv is highlighted. You can repeat this exercise by clicking the values in the county_name column. In either case you will find these columns NULL in every file except florida.csv and that florida.csv is the only file that contains these columns.

This tells us that someone changed field names on us and that the only file affected is florida.csv. When we were doing a manual Union step, we had a tool that could map misnamed fields to each other but we have no such tool when we are doing a wildcard union. In this case we can still merge the two fields together in the same way we did when the misnamed fields button told us which fields to combine but we don't have that automatic way to identify the fields. Fortunately, it's not a terrible burden to manually identify fields that should be combined. Once identified, simply drag the field you want to combine onto the field you want to keep, and you are good to go!

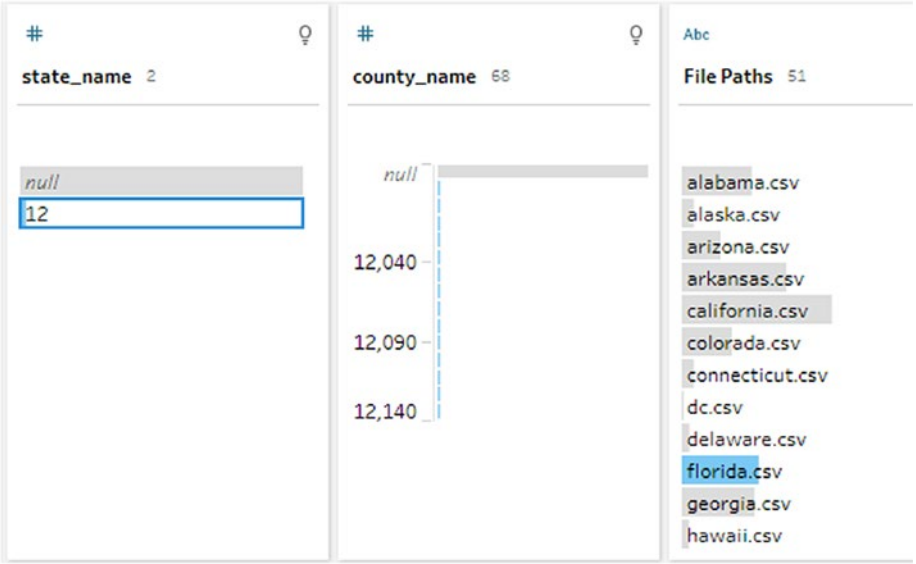



***Figure 4-25.*** *Investigating mismatched fields in a wildcard union v*

# Which Is Better?

We've spent quite a bit of time looking at how you can union data together in Prep. We walked through the Union step, and then we did a deep dive on the wildcard Union. Which is better? When should I use one over the other?

In general, I let the data decide. If all the data I want to union is in the same folder or set of folders or the same database, I lean toward the wildcard union. It's a little riskier in that it won't tell me if I have mismatched fields, but it is a lot faster. In some cases, I want to union data that comes from multiple sources, like tables in a database and data in .csv files. In those cases, I will have to use manual union steps to bring it all together. The thing to remember with the union step (as opposed to wildcard unions) is that the union step only accepts up to ten inputs. If you have 20 inputs, you would need to union 10 of them in one step, union 10

more in a second union step, and then union the 2 steps together in a third union. That's a lot of unions! (See Figure 4-26.)

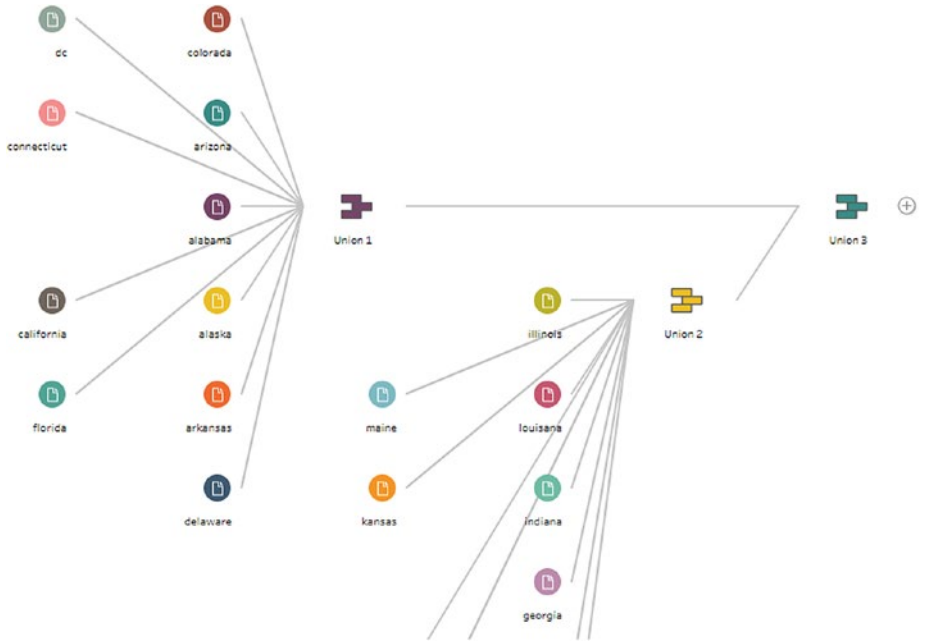



***Figure 4-26.*** *Multiple Union steps to get around ten input limits on each Union step*

# Summary

This has been a long chapter. We covered a lot of ground. The thing I want you to take away from this chapter is that you have options. As a consultant I frequently see people struggling to work with data spread over lots of files or tables (or both) that would be well served with a simple Union. Remember that the Union step gives you a lot of flexibility and you have built-in tools that recognize unmatched fields and make it easy to map them together. Wildcard unions are simply incredible when you want to bring together large numbers of data sources that all share the same format

(all files are Excel, csv, or all data stored in tables in the same database). Be cautious with the wildcard union, and remember that it won't warn you about unmatched fields. When doing a wildcard union, it pays to spend a little extra time auditing your data (we will get to that topic soon I promise!).

CHAPTER 5

# Joins

I've been a database guy for going on 20 years now. For at least the last 15 years I've been explaining joins. It's a tricky topic. When writing queries in SQL, joins can get out of hand quickly. One of the things that impresses me the most about the Tableau products are how effective they are in giving the end user the ability to do complex things while hiding the complexity from that user. The visual join designer is the best example of this genius. In this chapter we will dive deep into LEFT, RIGHT, INNER, and OUTER joins. We will look at joins that return records that match and joins that return only unmatched records, and we'll do all of this without ever writing a single line of SQL. If you already know how to write a query, I think you will be pleasantly surprised at the visual tools Prep gives us; if you don't know a thing about queries and have no interest in them, I think you will be thrilled at the power and flexibility this tool puts in your hands.

There are four main join types: left, right, inner, and outer. I'm going to give you a brief definition of each type here. Go ahead and read through this section; don't worry if it doesn't make a lot of sense up front. It will all fall into place when we get to the first exercise.

## What Is a Table?

Before we can talk about joins, we must understand what we are joining. In this book I define the concept of a table very loosely. That's one of the great things about Prep. It treats all data sources as if they were tables


T. Costello and L. Blackshear, *Prepare Your Data for Tableau*,
https://doi.org/10.1007/978-1-4842-5497-4_5

in a database. Once you have connected to your data and that data is in memory and available to Prep, we see it as a table. Your data might start out as a table in a database, a table in the cloud, a worksheet in a spreadsheet, or a simple comma-separated file. It doesn't matter, Prep sees them all as data sources and treats them all as tables.

Now, having said that there are some things to keep in mind. Prep prefers simple data. Data organized as column headers and row after row of data are the best. In some cases, we can work around fancy headers and summary lines and even images, but that is not ideal. Simple is better.

When we start talking about joins, we will be talking a lot about the things we are joining on. In order to join to data sources in Prep, we need a common element in both data sources (both sides of the join). Ideally this common element will be some sort of numeric ID field, but in many cases, we will join on text values. It's always better to join on a numeric ID when possible as this will give us the best possible performance, but real-world data is not always perfect and sometimes we have to be flexible.

## Equijoins

We think of joins in terms of how the two tables being joined are connected. No matter what kind of join we are talking about, the two tables must be connected on one or more common fields. I'm going to restrict the focus of this chapter to what we call equijoins or joins where we say a value on one side of the join equals a value on the other side of the join. There are also non-equijoins, joins where the left side does not equal the right, or the left side is lesser than or greater than the right. These joins ARE allowed in Prep, but I STRONGLY recommend against using them until you have mastered the equijoin. The times you will need a non-equijoin are extremely rare, and those cases are outside the scope of this book. Honestly, non-equijoins melt my brain, and I've been doing this a long, long time.

# Join Types

There are four primary join types: left, right, inner, and outer. In addition to these types, Prep gives us an additional layer it calls unmatched queries. We will explore unmatched queries at the end of this chapter.

# Inner Joins


***Figure 5-1.*** *Inner join*

Inner joins are the most performant (fastest) joins. They are also the most restrictive. An inner join returns only those records where the join value exists on both sides of the join. Inner joins are the default join type in Prep.

# Left Joins


***Figure 5-2.*** *Left join*

Left joins return all records from the left side of the join and only those records that match the join value from the right side of the join.

# Right Joins


***Figure 5-3.*** *Right join*

Right joins are a lot like left joins except they return all records from the right side of the join and only those records that match the join value from the left side of the join.

Simple, right? Well … no. I often refer to right joins as the evil twin of the left join. They look exactly alike, but they wreak havoc and cause despair until they are identified. Why am I so down on right joins? Because in my career I've written a million left joins and, in that time, I've used a right join maybe two times. Right joins do exactly as they say they will do; the problem is they don't do what left joins do and 99 times out of a hundred you will expect them to act like left joins. This can lead to unexpected results, and unexpected results lead to long nights of debugging data flows. I tend to avoid right joins.

# Outer Joins


***Figure 5-4.*** *Outer join*

Outer joins are another join type that you won't see a lot, but unlike right joins they can be extremely useful (in some cases). The outer join returns ALL records from the left side of the join AND all records from the right

side of the join. This will result in a lot of NULL values in your data, but you won't lose any records.

We have now set the stage for some hands-on practice with joins. Before we do that, let's step back and talk through a couple things you want to think about when choosing a join.

## The Shape of Your Data

I like to always start with the end in mind. In the case of joins, it's useful to consider what you want your data to look like (the shape of your data) before you begin.

If you have multiple data sources that all have the same columns and you simply want to combine them, so the results have the same columns but include the combined rows from all sources you want to use a union join.

If you want to make a wider record, or a record that is some combination of fields from data source A and data source B, you want to use one of the other joins. These joins will make your record wider by adding new fields. This is what I think of when I refer to shaping your data. You started with a record that was shaped with ten fields. You finish with a record that includes all the original ten fields but adds five new fields. You have reshaped your record with a join.

Union joins do not (usually) change the shape of your record, instead they add additional rows. The other joins do change the shape of your data but do not add any additional rows of data. Union joins make your data deeper (more rows). Other joins make your data wider (more columns).

---

**Note** There is an exception where left, right, and outer joins can add additional records to your result set. If you have a one to many relationship between the left and right sides of your join, you could

get additional records in your result set. For example, if you have one record for an item in inventory and you join to a data set of sales where that one record might be represented multiple times (a widget in inventory sells ten times), you could get a record in your results for each sale. That new record would have the same details from inventory and unique details associated with each sale.

---

# Exercise 5.1: Joins

In this exercise we will practice joining two tables stored as named ranges in an Excel workbook.

1. Open the Data Prep Builder
2. Open a new connection to an Excel file
3. Browse to the location you saved your exercise files for this book
4. Select SimpleLinking.xlsx from the Exercise 5.1 folder
5. Notice that you have three objects to choose from. Sheet1 is a worksheet. Table1 and Table2 are both named ranges. Prep treats named ranges within worksheets in Excel as if they were standalone tables.

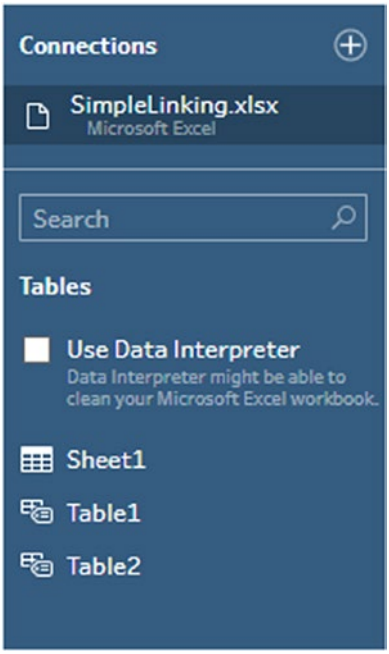



***Figure 5-5.*** *Named ranges are available as tables when connected to an Excel workbook*

6. Double-click Table1 and Table2 to add them to the design canvas

7. Click an empty spot on the design canvas to set focus there and close the review pane

   Click and hold over Table2, drag Table2 over Table1, and let it go on the orange drop zone labeled Join.

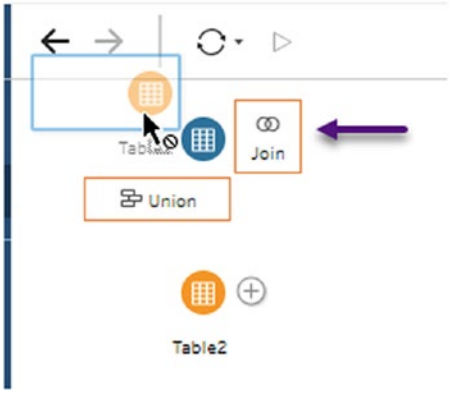



***Figure 5-6.*** *Click, hold, and drag Table1 over Table2. Drop it on the orange box marked "Join"*

Congratulations, you just joined two data sources together! Now let's take a look at what we have done. If you followed along, your screen should now look like Figure 5-2. Note: I broke the image up so it would fit better on the page.

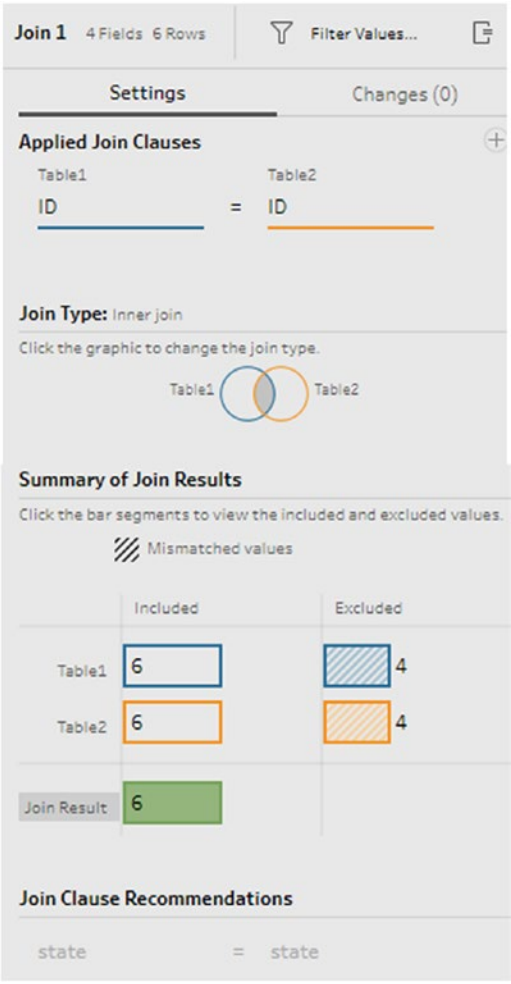



***Figure 5-7.*** *Review pane showing join configuration controls*

Let's take some time and explore the different options available here and see how Prep is helping us to anticipate the results of our join.

First, and perhaps most importantly, we should make sure we are in control of which table is on the right side of our join and which is on the left side. If you have read along to this point and you still have a lingering question about "sides," look at the Venn diagram under the heading "Join

Type" in Figure 5-2 (or on your review pane when you are looking at a join in Prep). Notice that we have two overlapping circles. One is to the left of the other. When we write queries in SQL, one table is on the left side of an equation, and the other table is on the right. Here Prep is making this easier for us by simply displaying the join as a Venn diagram. Interacting with this diagram will change the type of join we have from inner (the default) to left, right, or outer. We'll get to that in a moment.

Notice that Table1 is on the left side of the Venn diagram and Table2 is on the right? This is because we dragged the connection step for Table2 and dropped it on Table1.

---

The connection step you drop a connection on to will be the **left** side of the join.

---

Next, go back and redo Exercise 5.1, but this time drag Table1 onto Table2. Before you begin, don't forget to delete Join1.

This time you will find that Table2 will be on the left side of the join. I strongly encourage you to always try to use left joins. As you get more practiced, it will be easier to anticipate the results of your joins if you are consistent in this way.

One last time delete Join1 and redo Exercise 5.1; this time do it as it was originally written. This will bring us back to a state where Table1 is on the left and Table2 is on the right. If you don't reset in this way, what you see on your screen might be different from the screenshots you will see in the rest of this chapter.

Next, let's make a change to our join clause that will make it easier to follow along with the changes that we are about to make. Click ID under Table1 below the heading "Applied Join Clauses" (see Figure 5-3).

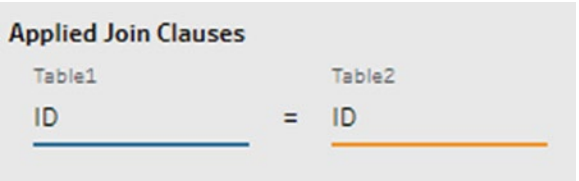



***Figure 5-8.*** *Changing the join clause*

Select State under Table1 and then select State under Table2. This will change the join so that instead of joining on the ID field, it will now join on the state field. This will make it a little easier for us to see the results of the changes we are about to make. In most cases, if you have an ID field that you can link on, that's what you would want to do.

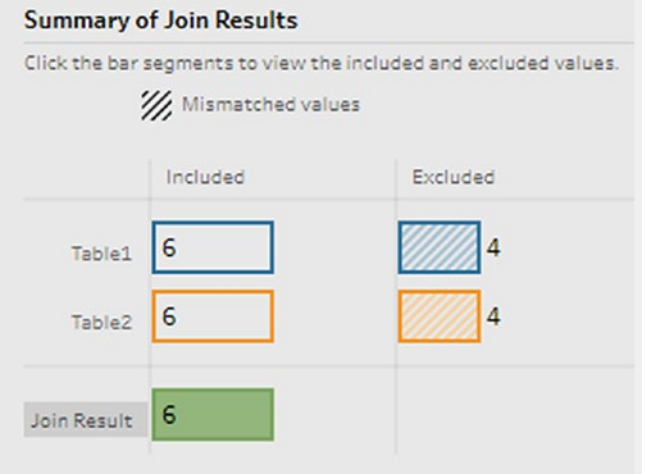



***Figure 5-9.*** *Summary of Join Results*

Let's look at the Summary of Join Results (as seen in Figure 5-4). I find this display to be extremely useful. It helps me see the results of my join at a summary level. In this case we are doing an inner join (the default join type). If you recall, the inner join will only return records where the join field contains the same value on both sides of the join. In this case we can see that each side of our join has six records in common (included) and

each has four records that are distinct (excluded). Table1 (the blue boxes) has ten records, six of which match Table2 and four of which do not match Table2. We see the same thing in Table2 (the orange boxes).

Now click the center of the blue circle on the left side of the Venn diagram. (See Figure 5-5.)


***Figure 5-10.*** *Click the center of the blue box on the left side of the Venn diagram to change the join from an inner join to a left join*

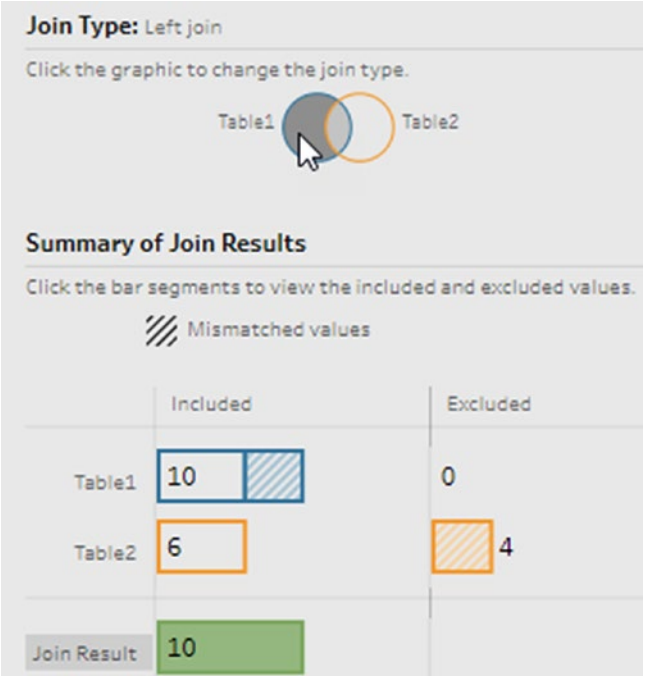



***Figure 5-11.*** *Summary of Join Results for a left join*

Notice how the Summary of Join Results has changed? Remember that a left join returns all records from the left side of the join (Table1 in this case) and only those records that match on the state field in the right side of the join (Table2). Now we see that Table1 (the blue boxes) will return ten records (all the records in this table) and Table2 (the orange boxes) will return six records (only six records in Table2 match records in Table1).

Click your mouse in the center of the Venn diagram circle for Table1 (blue circle) again, and you will return the join type to an inner join (the default). This will reset the Summary of Join Results back to what you see in Figure 5-4.

Click your mouse in the circle on the right side of the Venn diagram (Table2), and you will change the join type from an inner join to a right join.

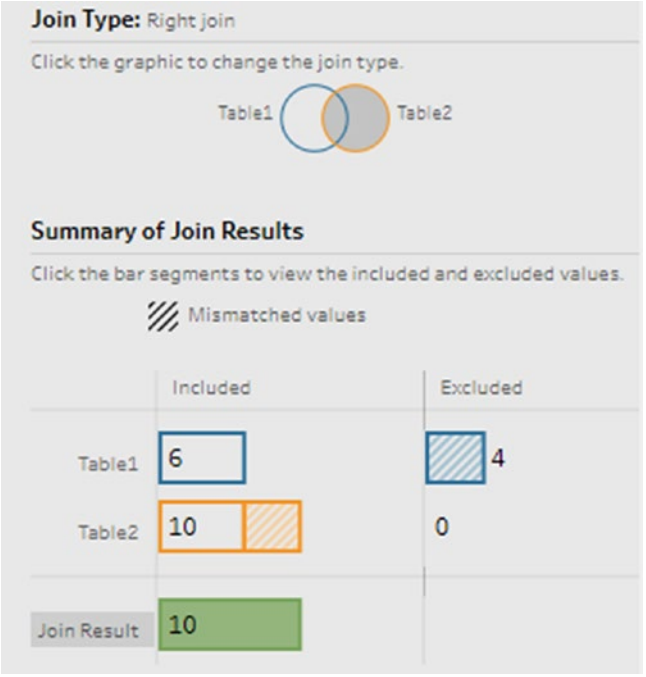



***Figure 5-12.*** *Summary of Join Results for a right join*

At this point, I urge you to take some time and study the differences between Figures 5-7 (left join) and 5-8 (right join). One thing to note is how the colors in this pane all work to help you focus on the two tables involved in the join. In the Venn diagram, Table1 is blue; Table1 is still blue in the Summary of Join Results. Table2 is orange throughout these summaries. I find that to be very helpful.

Remember that Figure 5-6 is a left join. In that summary you will see Table1 returns ten records (blue box under the included column). Figure 5-7 is a right join. It returns ten records in the orange box under the included column. You want to become very comfortable with these summaries. They will clue you in to problems with your joins and sometimes to problems with your data.

Another great tool for validating your data is hiding, just out of site in Figure 5-7. Click the Join Result button (green in Prep). This will bring up a summary view of the data that will be returned from both tables. Here you can see fields returned by Table1 (all records) and the fields returned by Table2 (only records matching on state value).

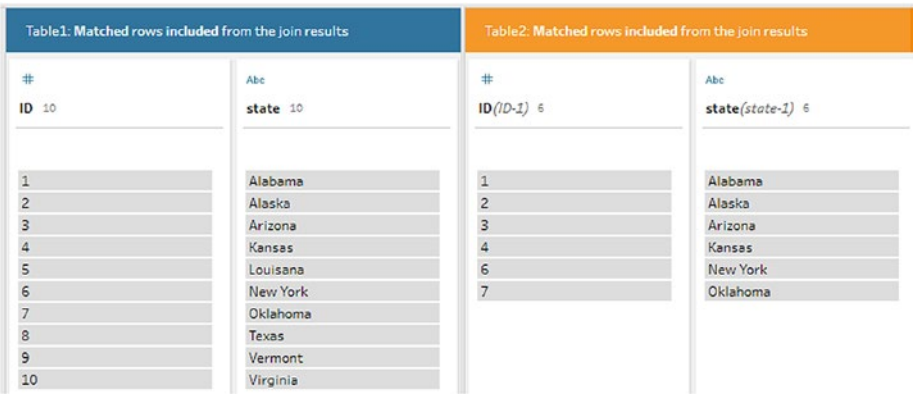


Table1: **Matched rows included** from the join results

| # ID 10 | Abc state 10 |
|---|---|
| 1 | Alabama |
| 2 | Alaska |
| 3 | Arizona |
| 4 | Kansas |
| 5 | Louisana |
| 6 | New York |
| 7 | Oklahoma |
| 8 | Texas |
| 9 | Vermont |
| 10 | Virginia |

Table2: **Matched rows included** from the join results

| # ID(ID-1) 6 | Abc state(state-1) 6 |
|---|---|
| 1 | Alabama |
| 2 | Alaska |
| 3 | Arizona |
| 4 | Kansas |
| 6 | New York |
| 7 | Oklahoma |

***Figure 5-13.*** *Summary of records returned by left join*

# Missing Data

A very common issue I've seen in work my customers share with me is missing data. We might have two data sources and assume that each should match the other on its key column (join column) 100%. For example, imagine a scenario with sales and shipping data. In our scenario we will pretend that every item in inventory ships at least once a month. If we were to try to join an inventory worksheet with a monthly shipping worksheet, we would hope that every item in the inventory would be returned and every item in the shipping summary would be returned. This would confirm that every item in inventory was shipped in that month. In this case it would be natural to go with the default join type (inner join). An inner join would give us all the details from inventory and all the details from shipping. If we had 1000 records in inventory and 1000 records in shipping, we would hope to find 1000 in the blue box under included for inventory and 1000 in the orange box under included for shipping. The real world however is not so kind to us. We would probably see something like 600 in both boxes. This would indicate that we have a match between itemID in inventory and shipping for 600 records. That leaves 400 records that either did not ship or shipped but do not exist in inventory. We will dig deep on the topic of identifying missing data for the remainder of this chapter.

Let's look at the SimpleLinking.xls spreadsheet we have been working with in this chapter. Let's imagine that the state names represent snow globes. Snow globes manufactured with a scene particular to each state. We can think of Table1 as our inventory data and Table2 as our shipping data. I've included a screenshot of the actual data for those that don't have Excel handy as follows.

| Inventory | Item |
|---|---|
| ID | state |
| 1 | Alabama |
| 2 | Alaska |
| 3 | Arizona |
| 4 | Kansas |
| 5 | **Louisana** |
| 6 | New York |
| 7 | Oklahoma |
| 8 | **Texas** |
| 9 | **Vermont** |
| 10 | **Virginia** |

| Shipping | Item |
|---|---|
| ID | state |
| 1 | Alabama |
| 2 | Alaska |
| 3 | Arizona |
| 11 | **Florida** |
| 12 | **Indiana** |
| 4 | Kansas |
| 6 | New York |
| 7 | Oklahoma |
| 14 | **Oregon** |
| 13 | **Pennsylvania** |

***Figure 5-14.*** *Source data*

Now we have two data sets that we want to join and make available for analytics. Our problem is that we have some items in inventory that have never shipped and some items that have shipped that do not exist in inventory. This is easy to see in Figure 5-8, but in the real world, it can be less easy to root out this kind of problem. Fortunately, we have some join options that might help us get this project on track.

## Finding Missing Records

I'm going to walk you through this next part slowly, and there will be lots of screenshots. Frankly, this was one of the more complex things I have done on a regular basis, and now that I have Prep in my toolbox, this job has become almost routine. There are a couple steps to master, but they are not difficult, and once you have them down, you'll be able to produce results that would have been a lot harder without Prep. Let's begin.

We will start with an inner join between Table1 (inventory) and Table2 (shipping). We walked through the steps to get an inner join between these tables in Exercise 5.1. Once you have the inner join, I want you to look to the upper right side of your review pane, right next to the Search form.

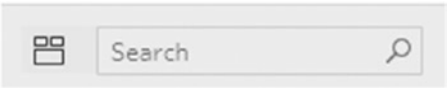



***Figure 5-15.*** *Toggle view button*

Click the small button that looks like a rectangle with two squares above it. This will toggle the preview pane into site (or more into site depending on the size of your monitor).

Next click the Table1 and Table2 boxes under Included in the Summary of Join Results section.

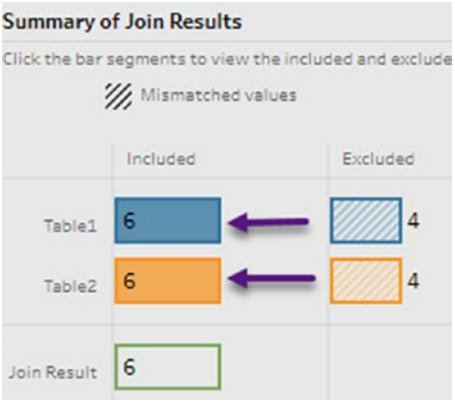



***Figure 5-16.*** *Click the Table1 and Table2 boxes*

This will display the records that will be returned by an inner join between these two tables (as seen in Figure 5-11).

Table1: **Matched** rows included from the join results

| # ID | Abc state |
|---|---|
| 1 | Alabama |
| 2 | Alaska |
| 3 | Arizona |
| 4 | Kansas |
| 6 | New York |
| 7 | Oklahoma |

Table2: **Matched** rows included from the join results

| # ID(ID-1) | Abc state(state-1) |
|---|---|
| 1 | Alabama |
| 2 | Alaska |
| 3 | Arizona |
| 4 | Kansas |
| 6 | New York |
| 7 | Oklahoma |

***Figure 5-17.*** *Rows included by inner join*

Next, go back to your Join Type Venn diagram and click the left circle, then click the right circle, and finally click the wedge between the left and right circles. This will change your join from an outer join to a return unmatched join.


***Figure 5-18.*** *Unmatched only*

This is pretty special folks. I have not seen anything like this in a tool before. To see what we have done, go down to the Summary of Join Results (as seen in Figure 5-10), and click the included boxes for Table1 and Table2. This should change your summary of results to show the records that exist in each table that do not exist in the other table.

| Table1: **Unmatched** rows **included** from the join results | | Table2: **Unmatched** rows **included** from the join results | |
|---|---|---|---|
| # **ID** | Abc **state** | # **ID** *(ID-1)* | Abc **state** *(state-1)* |
| *null* | *null* | 11 | Florida |
| *null* | *null* | 12 | Indiana |
| *null* | *null* | 14 | Oregon |
| *null* | *null* | 13 | Pennsylvania |
| 8 | Texas | *null* | *null* |
| 10 | Virginia | *null* | *null* |
| 5 | Louisana | *null* | *null* |
| 9 | Vermont | *null* | *null* |

***Figure 5-19.*** *Summary of unmatched records*

## Bringing It All Together

Next, we are going to bring it all together by adding a step and two calculated fields. We haven't talked about these much yet in this book, but I think it will be fun to peak ahead a bit and see what's coming.

Click any white space on your design canvas to close the Join review pane. Next click the plus symbol next to Join1 and select Add a Step.

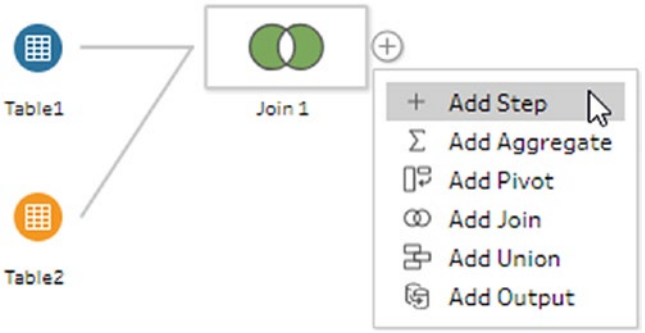
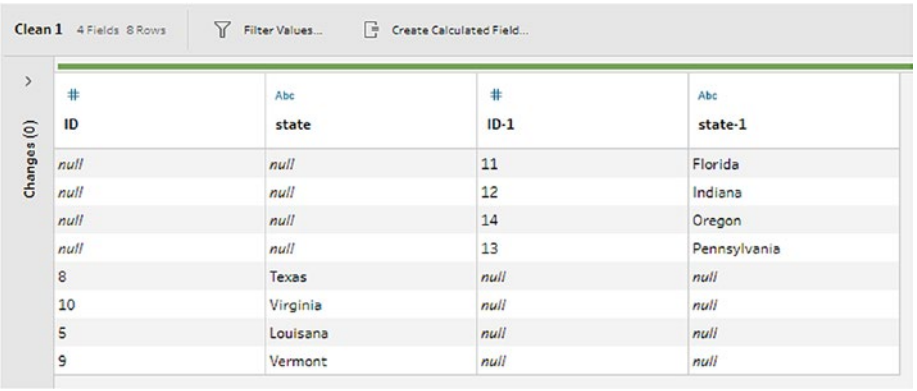



***Figure 5-20.*** *Add a step*

You will now have a new step on your design canvas called Clean1, and you should be looking at the review pane for that new step.




| # ID | Abc state | # ID-1 | Abc state-1 |
|---|---|---|---|
| null | null | 11 | Florida |
| null | null | 12 | Indiana |
| null | null | 14 | Oregon |
| null | null | 13 | Pennsylvania |
| 8 | Texas | null | null |
| 10 | Virginia | null | null |
| 5 | Louisana | null | null |
| 9 | Vermont | null | null |

***Figure 5-21.*** *Review pane for new cleaning step*

This should look familiar. It's basically the same thing we saw in the preview pane in the join we just created (see Figure 5-12). Here's where we are going to get creative and create a couple calculated fields. If you aren't familiar with calculated fields from working in Tableau builder, just follow along, we will cover them in more detail later in the book.

Begin by clicking Create Calculated Field.

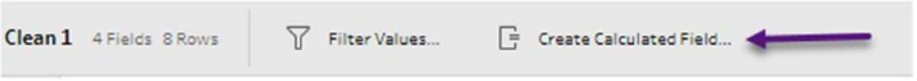



***Figure 5-22.*** *Create a calculated field*

Edit your calculated field to match the screenshot and click Save.

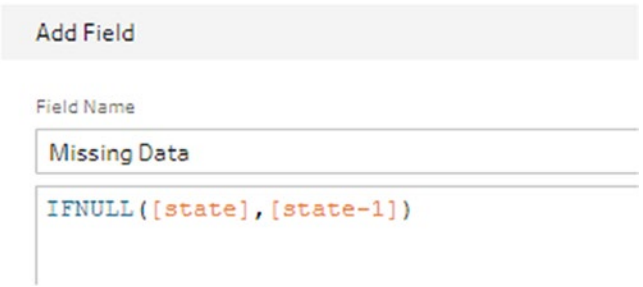



***Figure 5-23.*** *Edit calculated field*

```
IFNULL([state],[state-1])
```

Next, create another calculated field based on the following screenshot and code and click Save.

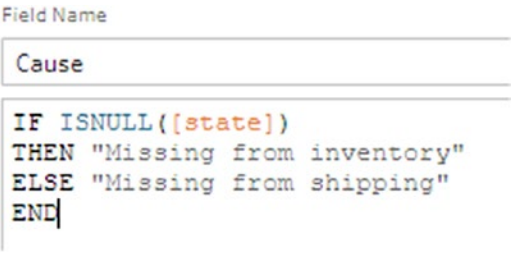



***Figure 5-24.*** *Definition for Cause calculated field*

```
IF ISNULL([state])
THEN "Missing from inventory"
ELSE "Missing from shipping"
END
```

Congratulations! You've created a join that returns items missing between two tables and calculated fields to consolidate your results and label them!

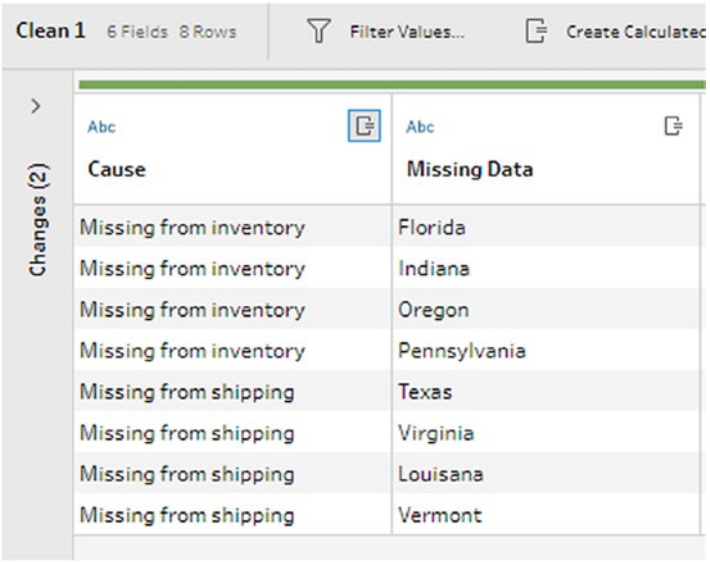



***Figure 5-25.*** *Calculated fields added in cleaning step*

## But Wait, There's More!

There's one last tool I want to draw your attention to before we move away from our discussion of joins. Let's go back to the Join review pane by clicking the join step on the design canvas. Scroll down to the settings pane. Notice at the bottom there's a section marked Join Clause Recommendations? This is where Prep offers suggestions that could in many cases make your join better.

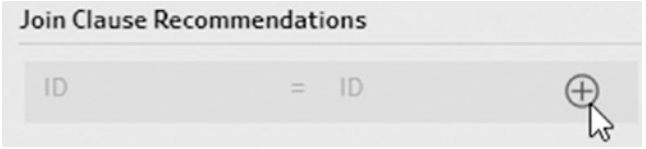



***Figure 5-26.*** *Join Clause Recommendations*

In some cases, Prep will suggest ways to make your joins better. In this case Prep is telling us that having a join on the ID helps. I think it's making this suggestion because joins on numeric fields are more performant than joins on text fields. If we click the plus symbol next to this recommendation (see Figure 5-20), Prep will add this recommendation to our join for us.


***Figure 5-27.*** *Join Clause Recommendations added an additional join on ID*

I would go a step further and recommend that you now remove the state join from the Applied Join Clauses. It is now redundant, and in cases where you might have a data entry problem (misspelled state), it could even cause a problem.

Closing out this section, I recommend that you always look at the Join Clause Recommendations and take a moment to manually review the Applied Join Clauses before moving away from the Join review pane. In most cases Prep will offer perfectly reasonable joins, but occasionally things do go sideways. The way you configure your joins will dramatically impact the performance and accuracy of your data pipeline in the Data Prep Builder.

## Summary

We've covered a lot in this chapter. Joining tables (data sources) is one of the easiest things to do in Prep AND one of the easiest ways to get yourself into trouble in Prep. Fully understanding the differences between union, inner, left, right, and outer joins will make all the difference.

Pay careful attention to the type of join you create (remember, the source step you drop a connection ON TO will be the LEFT side of the join). Always review the Applied Join Clauses and the Join Clause Recommendations. Remember that while joins that return matching records are the default (and by far the most common) joins, you can also configure your join to return unmatched records. This is an often-overlooked feature of the join designer that can bring to light problems that would otherwise be hard to identify.

Mastering these concepts will make joining data sources easy and let you focus on the heart of your data flow, the data cleaning, grouping, aggregating, and pivoting steps. We look at those topics next!

# PART II

# Transform

The Transform steps are at the heart of our work in the Data Prep Builder. We have successfully connected to our data, now we make that data our own. Here is a peak at what's to come …

## Audit

The first step in working with data is to know your data. In this section we look at how to spot patterns in our data and what those patterns might mean.

## Clean

Here we rename fields, alias the content of fields, break some data out into new fields, and consolidate others from multiple to single fields. Prep has a lot to offer in this step, and we will take advantage of all of it!

## Group and Replace

The group and replace functionality in Data Prep are perhaps the most powerful elements of this entire tool. This section explores the various options you have for grouping your data and shows you how to control what goes into each group.

## Aggregate

It is not uncommon to receive data at a very fine level of detail and want to aggregate that data at a higher level of detail for analytics. For example, you might receive data that monitors machinery with readings captured multiple times every second. You might want to create an aggregated version of that data with details summarized every hour. In those cases, it can make sense to do that aggregation in Prep and avoid a lot of reaggregation in Tableau Builder. We will look at when you might want to aggregate and how to make that happen in this chapter.

## Pivot

Pivoting gives us another chance to reshape our data to make it easier to work with in Tableau Builder. Some things are just a lot harder to build in Tableau if the data isn't shaped in a way that supports our vision. In this chapter we will look at common patterns required for analytics and how Pivot can be the perfect tool to prep data to support those needs.

# CHAPTER 6

# Audit

All data is dirty data until proven otherwise. I've lived my life as a database guy holding this rule close and repeating it often. It's never let me down. In this chapter we will look at our data with fresh eyes and new tools. The Data Prep Builder is good at a lot of things, it is **great** at auditing data. Learn these tools and patterns and your work will come together with ease. Avoid them at your peril. (Queue dramatic background music …)

Ok. I hope you're still with me. Yes, that opening seems a bit ominous. I've dealt with a lot of dirty data in my life. It's made me a bit untrusting (to say the least). What I DO trust are the tools Prep puts in my hands to get a grip on a new data source. They are simply wonderful. That put a smile on my cranky database guy face. Let's start with an exercise.

## Exercise 6.1: Connect to Data

---

**Note** From this point forward, the exercises will have less detail on some steps if those steps were included in previous exercises.

---

1. Open a new text file connection
2. Browse to the folder where you saved the exercises for this book


T. Costello and L. Blackshear, *Prepare Your Data for Tableau*,
https://doi.org/10.1007/978-1-4842-5497-4_6

3. Browse to the Exercise 6.1 folder
4. Browse to the ZTCA to Census Track folder
5. Select the file zcta_tract_rel_10.csv
6. Open another new text file connection
7. Browse to the School District Data MCDC folder
8. Select the file ohio.csv
9. In the input review pane for the Ohio connection, click Multiple Files
10. Add wildcard union to the rest of the csv files in the same folder to this connection

---

(Review Chapter 4: Unions if you get stuck here.)

---

11. Click a blank spot on the design canvas to close the input review pane
12. Right-click the Ohio connection and rename it to "All States"
13. Right-click the zcta_tract_rel_10 connection and rename it to "ZTA – Tract"
14. Drag the All States connection and drop it on the Join drop zone for the ZTA – Tract connection
15. Change the Applied Join Clause to join on the field ZTA5 in both tables

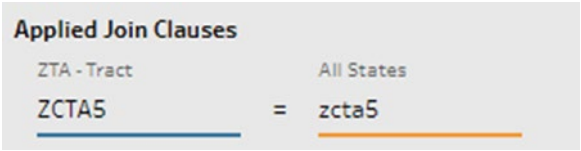



***Figure 6-1.*** *Update Applied Join Clauses to join on ZTA5 in both tables*

16. Close the Join review pane (see step 11)


***Figure 6-2.*** *All States and ZTA – Tract joined with inner join on ZCTA5*

We have now set the stage to begin auditing our data. Before we continue, I'd like to point out a new thing introduced in the previous exercise. After we established each connection, we renamed them by right-clicking and selecting Rename Step. I encourage you to get in the habit of naming your steps with short descriptive names as soon as you create them. It makes working with complex data flows a lot easier when each step has a simple, intuitive name.

While we are looking at the right-click menu for these steps, please notice the option marked Refresh. We can refresh our connection to a data source individually by right-clicking its step and choosing Refresh, or we can click the drop-down menu by the refresh icon at the top of the design canvas and selecting the connection to refresh. We can also refresh all connections by simply clicking the refresh icon.

***Figure 6-3.*** *Refresh*

---

Refreshing a connection brings any changes that have occurred in the data source since we connected to it into our data flow.

---

Many of the things we are about to do can be done in the Source step, but I prefer to do them in a new step. I find it to be easier to work with a data flow when changes are made in a new step rather than adding them into the connection step.

Let's continue by adding a step to look at the auditing tools built into Prep.

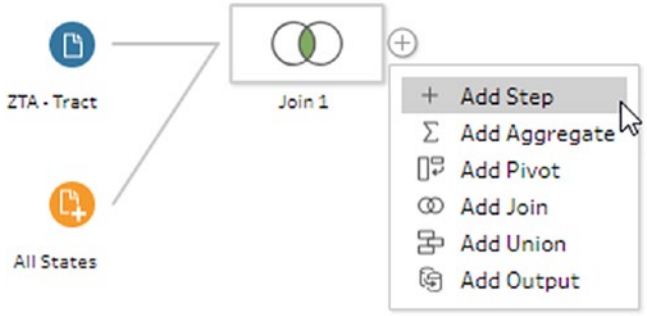



***Figure 6-4.*** *Add a step*

Take a moment before continuing to explore the review pane for the step we just added.

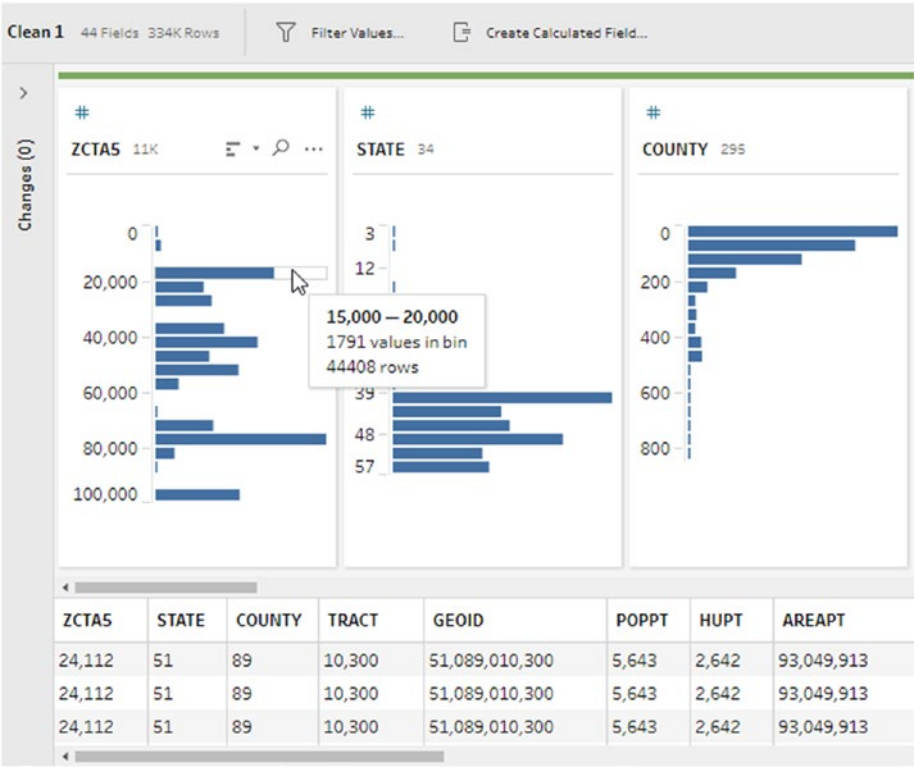



***Figure 6-5.*** *New step review pane*

You will find the tools (visualizations) you see here in a lot of the review panes you encounter in Prep. I like looking through them here because they are easier to navigate in the review than they might be in another step that has more going on in it (like the Source step).

Scan left and right, and you will see that every field from both connections is visible for review in the upper part of this screen. If the field is numeric, you will find a histogram showing your groupings (bins) for the values found in that field and bars to indicate the number of rows in each bin.

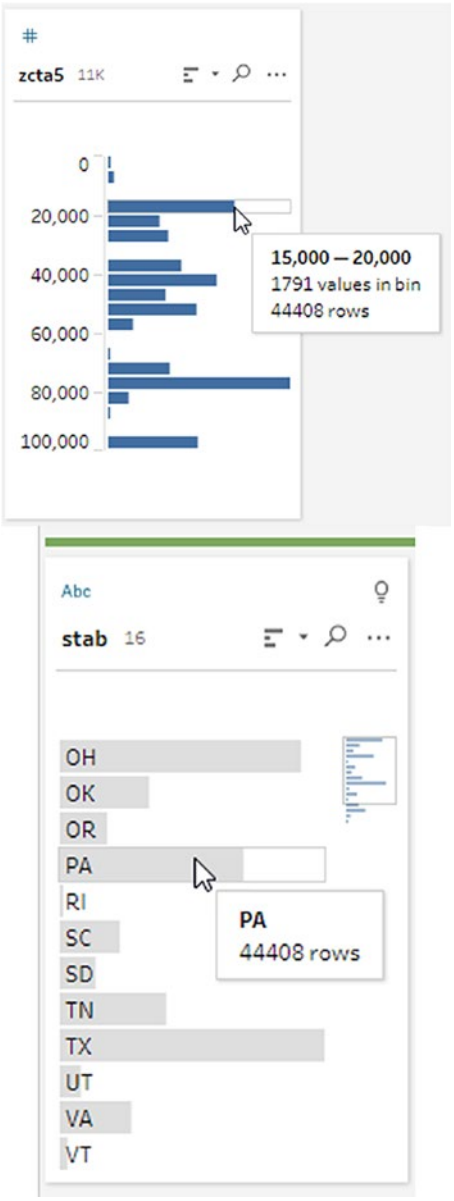



***Figure 6-6.*** *Histogram of numeric values and list of text values*

If the field contains text, you will see a list of the distinct values found in that field with a bar over each value that indicates the number of rows that value exists in.

At the top of each of these cards, you will find a small icon to sort the view. Clicking once will sort the bars descending, the next click sorts them ascending, and a third click returns the bars to an unsorted state. You can also filter the view in any of these cards by clicking the magnifying glass and entering a value to filter on. If any part of the value you enter exists in this field, it will be returned in the updated display. Searching for T in the stab field (see Figure 6-6) would filter the view in the card to only those values that contain the letter T.

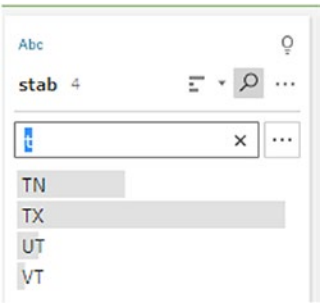



***Figure 6-7.*** *Summary review card filtered to values that contain T*

These displays are extremely helpful for getting a gut check on your data. I've discovered missing data at this point in my analysis many times. These visualizations can clue you in to holes in your data very quickly. In Figure 6-6 it is clear that Texas is a major contributor to this data set. Looking at this chart can quickly call into question a data set that has a lot less (or more) records associated with an expected value than you are expecting.

While these charts do give us a nice high-level overview of the data set, I'm sure some of you are thinking "well yes Tim, but I want to see the details …." Have no fear, Prep has you covered there as well.

Click the toggle view button at the top of the summary viz to toggle the detail viz into view. It is located on the top of the viz next to the search bar.


***Figure 6-8.*** *Toggle details button*

This will toggle into view the data source details. Here you will be able to see every record and every field available to this step. Next, toggle the Summary back into view, and let's look at how we can filter the details to only the records we want to validate.

With the summary view open, scroll to the right until you find the field named stab. See Figure 6-9.

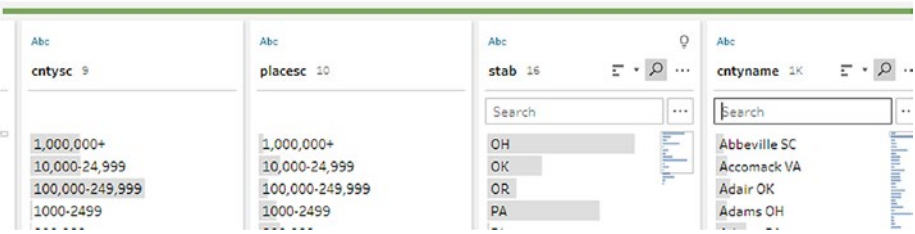



***Figure 6-9.*** *Select cntysc, placesc and stab*

Click the header for the field cntysc, and then shift+click the header for the field stab. This will multiselect the fields cntysc, placesc, and stab. You will know they are all selected because they will be outlined in blue.

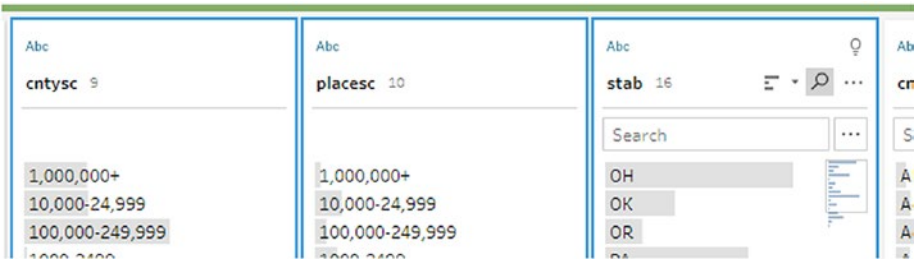



***Figure 6-10.*** *Multiselect fields in the summary view*

Click the ellipsis symbol (…) in any of the selected cards and select Keep Only.

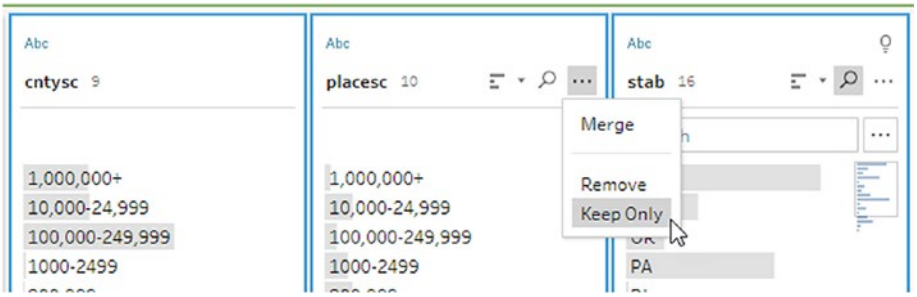



***Figure 6-11.*** *Filter summary view to only selected fields*

You will notice that the icon for this step on the design canvas now has a change icon over it. We will go into making and tracking changes in much more detail in the next chapter.

We went through this effort to limit the fields in our summary to make the next point clearer. The cards in the summary view act just like visualizations in a Tableau dashboard. Each of them displays important information on their own, and each of them can filter the others to give us even more insight when we are auditing our data.

To see this in action, click the bar for OH in the stab field. You will now see the bar on the histograms for the other two visible fields has updated to show the percentage of values in those binsis associated with our choice

(stab = OH). Give it a try for yourself. For those of you not following along at home, see Figure 6-12.

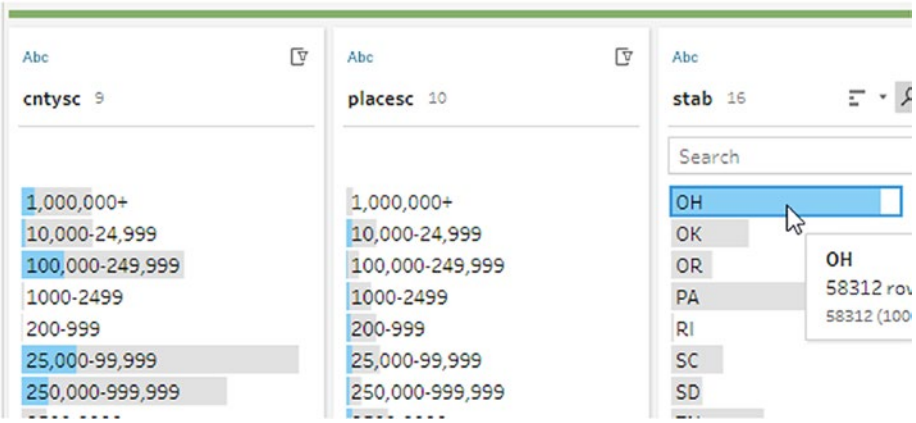



***Figure 6-12.*** *Selecting a value in once summary card filters the other summary cards*

Next, ctrl+click the 25,000–99,999 bar in the cntysc card. Control clicking this additional card will select the value you just clicked without losing the selection you made in the stab field.

Now click the summary/detail toggle (see Figure 6-8). You should see that the details view has also been filtered to only the fields you selected and only the values within those fields that you selected.

| Abc cntysc | Abc placesc | Abc stab |
|---|---|---|
| 25,000-99,999 | Not in a place | OH |
| 25,000-99,999 | Not in a place | OH |
| 25,000-99,999 | Not in a place | OH |
| 25,000-99,999 | Not in a place | OH |
| 25,000-99,999 | Not in a place | OH |

***Figure 6-13.*** *Detail review has been filtered to selections made in the summary cards*

# Summary

That might seem like a lot of steps, but I think when you try this with your own data sources, you will find this process to be intuitive. Prep makes it easy for us to gut check our data to ensure that it matches our expectations at the macro and micro levels. The summary review cards give us a great sense for the distribution of our data, and the detail view lets us focus in on the record level to make sure things line up as expected. I encourage you to try this with your own data. Pick a data source you feel is "good data" and put it through its paces. You might be surprised to find it isn't as clean as you previously imagined!

CHAPTER 7

# Cleaning

This is the chapter I wrote this book for. These are the tools I use the most in my day-to-day work with Tableau Data Prep. In this chapter we will make our first real changes to our data, cleaning up field names and data types, as well as making the contents of some fields more consistent and easier for our analysts to use. We will consolidate some fields, clean "garbage" from some fields, and in general make dirty data usable.

If you are reading this and you find yourself thinking "well … MY data isn't dirty," I have some sad news for you. ALL data is dirty, it's just a matter of who finds the problems first. My experience has been that my life is a lot less stressful if I'm the one spotting (and resolving) data cleanliness issues rather than having the consumer of my analytics raising those same issues down the road!

## Exercise 7.1: Add Step

Let's kick this chapter off with an exercise to get familiar with the Clean step. Before starting this exercise, connect to the DirtyData.csv file in the Exercise 7.1 files. You've done this step a few times, so I'll spare you the walk through. If you have skipped ahead to this chapter, you might want to go back and have a look at Chapter 3 for a refresher on connecting to data.


T. Costello and L. Blackshear, *Prepare Your Data for Tableau*,
https://doi.org/10.1007/978-1-4842-5497-4_7

1. Click the plus symbol to the right of the dirtyData connection

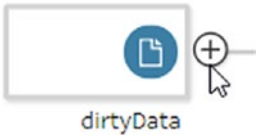



***Figure 7-1.*** *Add a step*

2. Click "Add Step"

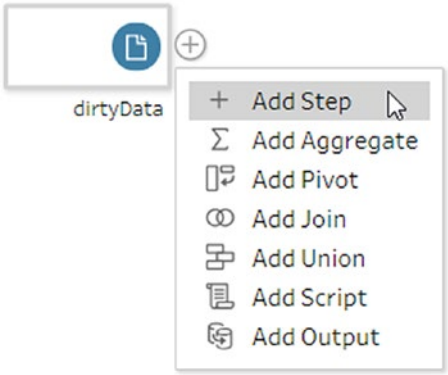



***Figure 7-2.*** *Adding a new step part 2*

3. Rename the new step by double-clicking "Clean 1" and giving the step a more descriptive name. It doesn't really matter what you call it, but my thinking is that this is a good habit to develop and best to start with the first step!

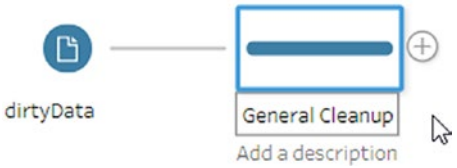



***Figure 7-3.*** *Rename new step*

You might have noticed that in this first short exercise we added a generic step, there is no step explicitly called "Clean." This is because this generic step lets you do so many things that at the end of the day it's easier to just think of it as a step, rather than a specific action.

What can we do here? Well … a lot. Here's a brief rundown of the cleaning steps, most of which we will explore in this chapter.

1. Make Uppercase
2. Make Lowercase
3. Remove Letters
4. Remove Numbers
5. Remove Punctuation
6. Trim Spaces
7. Remove Extra Spaces
8. Remove All Spaces

In addition to these actions, we can also do any of the following:

1. Rename fields
2. Alias contents of fields
3. Change data types
4. Assign role
5. Create calculated fields

6. Apply filters
7. Remove fields
8. Review changes

That's a lot of functionality for one step! Let's get back to the example we started earlier in this chapter and continue with some cleaning steps.

# Exercise 7.2: General Cleanup

In this exercise we will walk through a lot of the cleaning options listed in the preceding section to try to create a cleaner version of this data source that we might build on in future steps. We will continue with the data flow we started in Exercise 7.1.

## Merge and Clean

1. Select the cards for fields "name a" through "name d". You can do this by holding the Ctrl key on your keyboard and clicking each field or by clicking the card for "name a" and then shift+clicking the card for "name d". You will know you have all cards selected when each is outlined in blue.

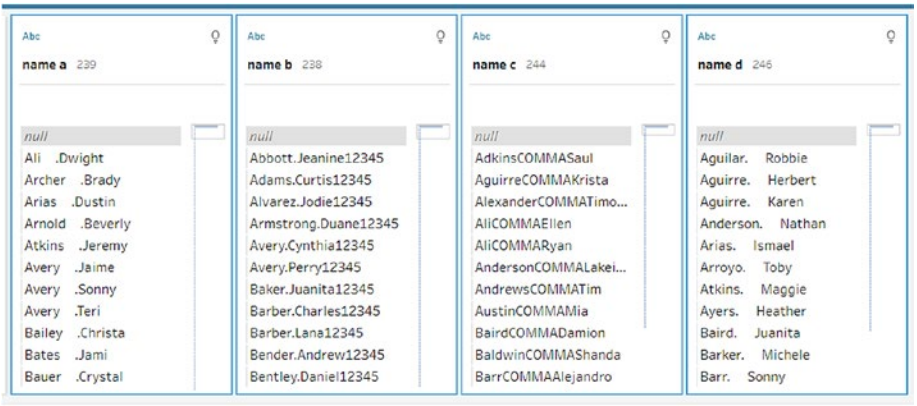



***Figure 7-4.*** *Select "name a" through "name d" cards*

---

**Note** You should only merge fields when the final result would be one valid (non-NULL) value in that field after the merge. Take a look at Figure 7-5 and notice that each record is either NULL or has a value. When we combine these fields into one merged field, the fields that have a value will overwrite the fields that are NULL to create a single field with only one value on each row.

---

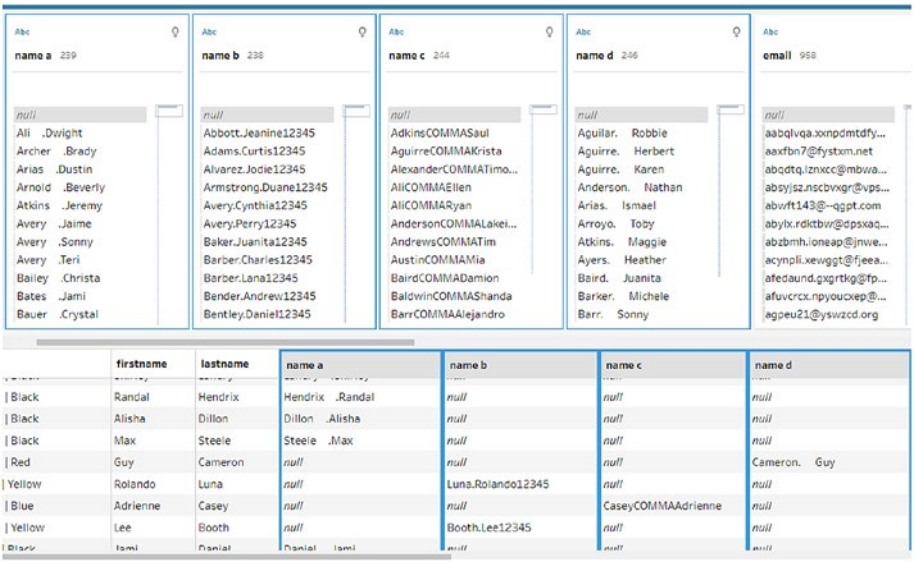



***Figure 7-5.*** *Verify that merged fields will contain only one non-NULL value per record*

2. Right-click one of the cards you have selected ("name a" – "name d") and select "Merge"

Two things will happen at this point. First, the four cards you selected will merge into one new card with the name "name a" at the beginning of your stack of cards, and second you will now see a change log has been created for you to review the work you have done in this step.

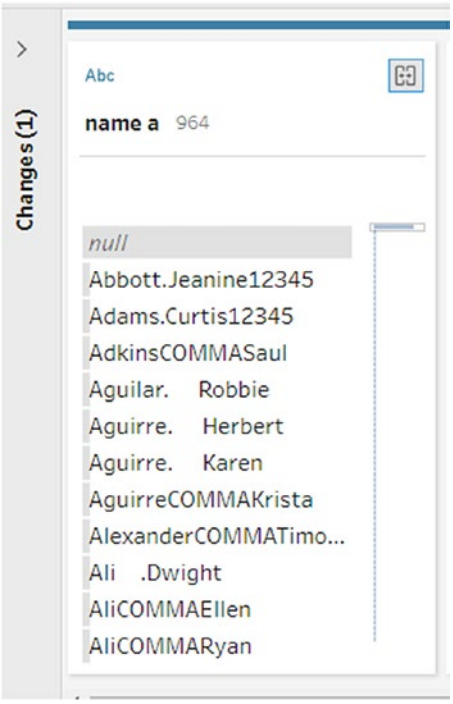



***Figure 7-6.*** *New card and change log*

Go ahead and open up the change log by clicking the label marked "Changes(1)" that you will find running diagonally along the side of the "name a" card.

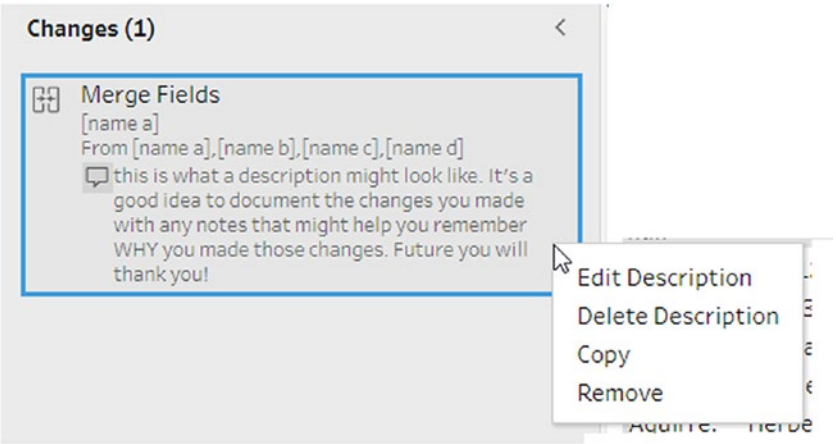



***Figure 7-7.*** *Change log*

There are a couple things to note here. First, each change has its own icon. We just added a Merge to four fields. The newly created field now shows the same icon in the header of its card that you see on this change step. This is a nice visual hint at the card level that one or more change has affected any given card. This change log will update with details for every change you make. Every step has a change log. You can undo any change by simply deleting it from the change log. You can go to the card associated with any change by double-clicking a change in the change log.

The change log also gives you some details on exactly what changed and lets you add a note (of up to 200 characters) to give you or those that follow you some context as to WHY you made a given change. I urge you to get in the habit from the start of adding notes to every step, every change. They will be invaluable when you come back to your work months or years later.

Next let's clean up the contents of this field. I'm going to walk you through a couple of the steps I might take if I were doing this cleanup. One of the great (and sometimes terrible) things about the Prep tool is there are a LOT of ways to accomplish most tasks. There are as many ways to do any given task as there are people to do the work. These examples are intended to give you a direction in which to start, not as a rulebook that must be followed.

Before we begin making changes to the data in the "name a" field, let's give that field a better name. Double-click over "name a" and give this field a better name. I'm going to call mine "Full Name".

## Make the Data More Consistent

1. Click the ellipsis (…) next to your newly renamed field name

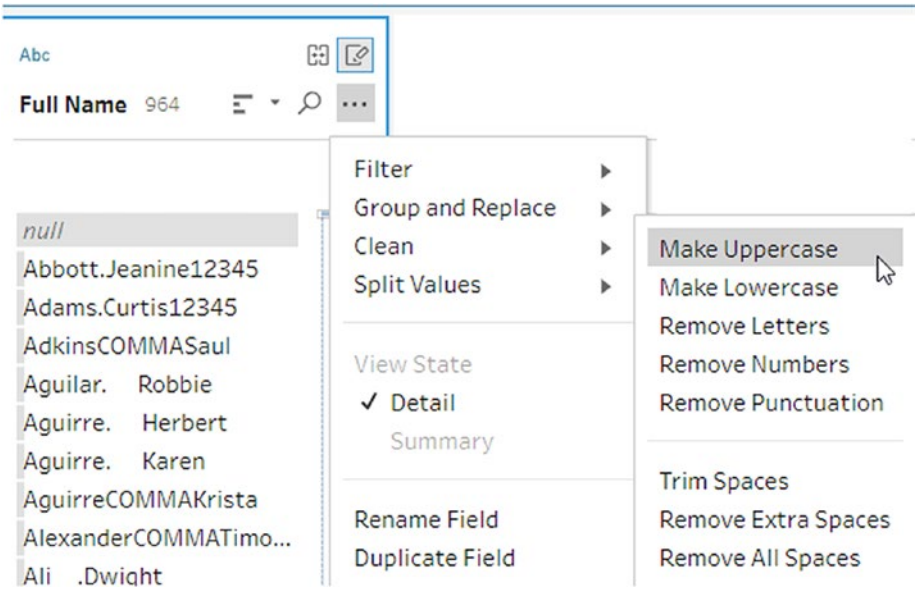



***Figure 7-8.*** *Cleaning options*

Some of these options are fairly obvious and we won't need them in this case. Make Uppercase and Make Lowercase speak for themselves, I think. Remove Letters will remove any alphabetic characters from your fields, Remove Numbers removes all numbers, and Remove punctuation … well, you get the idea. These are useful options when you have extra characters in your field that you want to simply get rid of.

We will move down to the next group and select Remove All Spaces to clear out all spaces from our field (spaces before, after, and inside our data in this field).

2. Click “Clean”
3. Click “Remove All Spaces”
4. Click “Remove Numbers”

   Next we’ll clean up a weird data entry problem that somehow inserted the word COMMA between first and last names in this field, rather than the comma character. We will create a calculated field to do this task for us.

5. Right-click the card for “Full Name” and select “Create Calculated Field”

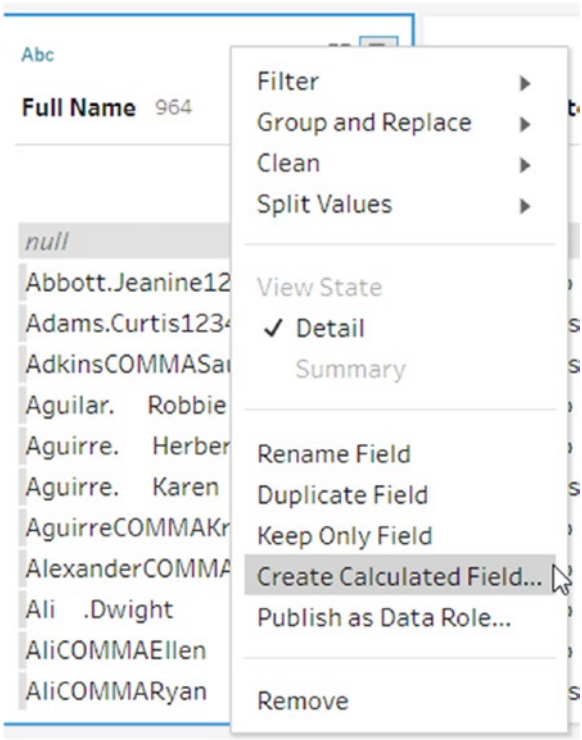



***Figure 7-9.*** *Create calculated field*

I love calculated fields. I could write a whole book about them. The power available in this tool is staggering. You can do a lot of cool things with a calculated field. If you aren't familiar with how (and when) to write calculated fields, I encourage you to get out on the Web and do a search for Tableau Calculated Fields. You will find a lot of great information out there.

For this example, we are going to simply search this field and replace every instance of the word "COMMA" with an actual comma.

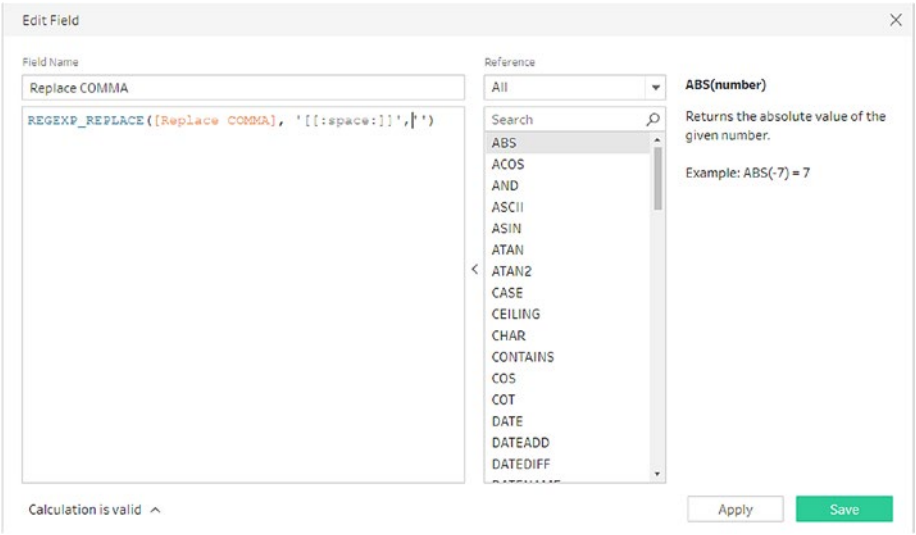



***Figure 7-10.*** *Replace COMMA*

```
Code:  REPLACE([Full Name],'COMMA', ',')
```

I want to call out a couple things to notice in Figure 7-10. First, ALWAYS name your calculated fields. You don't need to be elaborate here, but a short descriptive name will be greatly appreciated by whomever follows you in supporting this Prep flow.

Next, notice the right side of the Calculated Field dialog? In it you see a list of functions (All functions in this case but if you hit the drop-down next to All, you can filter the list to only Date, Logical, Number, String, or Type Conversion functions). To the right of the functions list, you will find a description of the REPLACE function with an example of how it could be used. Any time you click a function in the functions list, the help will update to show details for that function. If you use a function in your calculated field, you can click that functions name in your calculation, and the help will update to show details for the function you have selected.

If you don't see the list of functions and the help details, you might need to click the arrow on the right border of your calculated field editor to expand it to show these details.

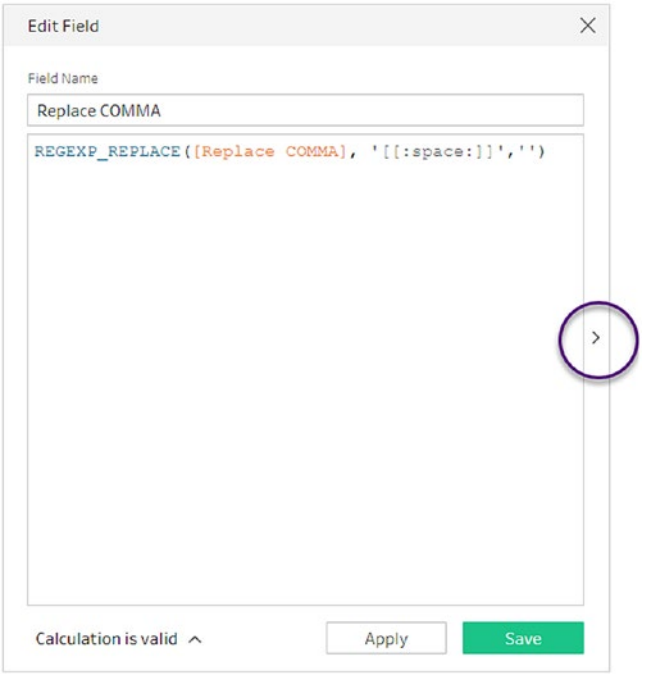



***Figure 7-11.*** *Expand Calculated Field editor to show functions and help*

6. Click Save

Next, let's take a look at our data. There are two things to notice. First, when we created our calculated field in the preceding step, it added a new card to our stack. This new card was named "Replace Comma". This card now contains the new field we just created. Second, you will find that some names are now separated with a period and others with a comma. Let's fix that.

7. Right-click the header of the “Replace Comma” field and select “Create Calculated Field”
8. Name your calculated field Last Name, First Name
9. Type the following formula into your calculated field

```
Code:  REPLACE([Replace Comma],'.',',')
```

10. Click Save

We now have a nice, clean version of our field. We no longer need some of the intermediate fields that were created to get here. Let’s delete them.

11. Right-click the card for “Replace Comma” and select “Remove”
12. Right-click the card for “Full Name” and select “Remove”

# Split

We’re off to a great start with this cleanup! We’ve consolidated four fields into one; we removed numbers, spaces, and characters. Next we’ll move on to the record locator and look at how we can make one field that contains multiple values easier to work with in our analytics.

The problem you will find in the “recordlocator” field is a common one. This field contains four discrete pieces of information, but unfortunately all that detail is locked in one field. It would be tricky to filter our data in a dashboard with this field. How would you filter to just

the Yellow records? How would you get a count of only those records with the value Yes? Simple, we break this field out into four new fields, each of which can be filtered with or independently of the others.

## Exercise 7.3: Split

We continue with the data flow started in Exercise 7.1.

1. Click the ellipses on the top of the "recordlocator" field
2. Select Split Values
3. Select Automatic Split
4. Rename recordlocator – Split 1 to EmpID
5. Rename recordlocator – Split 2 to Active
6. Rename recordlocator – Split 3 to DeptID
7. Rename recordlocator – Split 4 to FavoriteColor

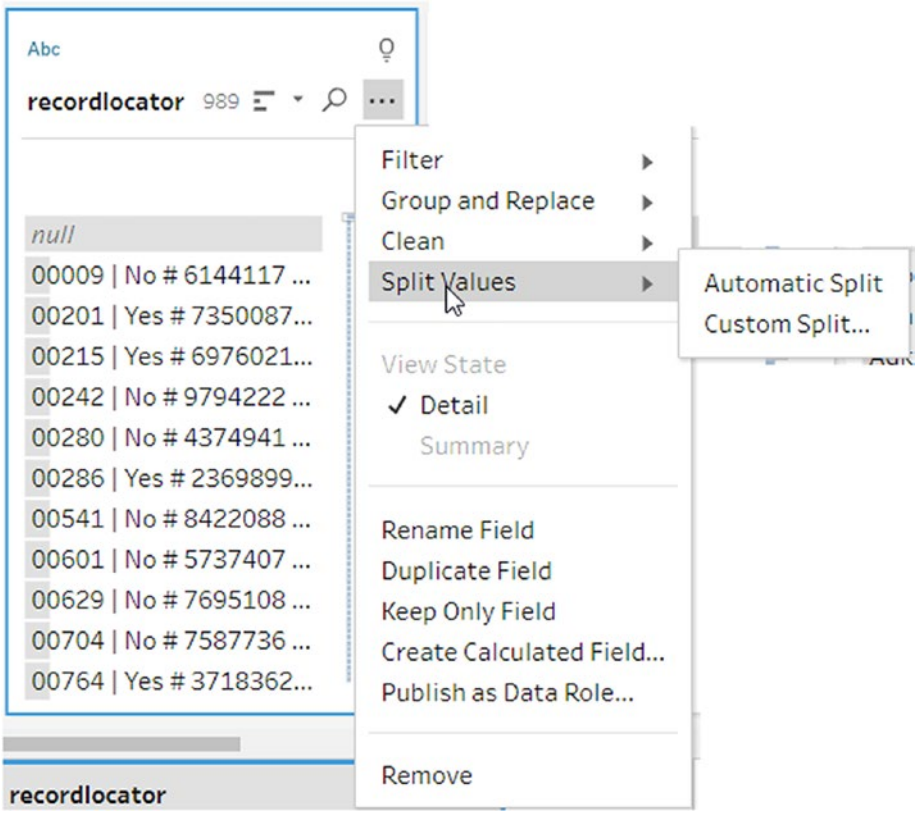



***Figure 7-12.*** *Automatic Split*

And … that's pretty much it. Anticlimactic? Maybe. But wow, you gotta admit that's cool! In just a couple clicks, we broke out data into four new fields, data that was separated with two different characters (the # and the |). If we had wanted to, we could have done a custom split. In some cases, we will indeed need to do a custom split, but in many cases, Prep is smart enough to identify the way data is separated within a field and adjust automatically.

If we had needed to do a custom split, we would have the option to split on any separator and to split out all groups present in the field or only the first, last, first n fields (where n is any number between 1 and 10) or the

last n fields. That should give you just about enough flexibility to handle any kind of split scenario you are likely to encounter!

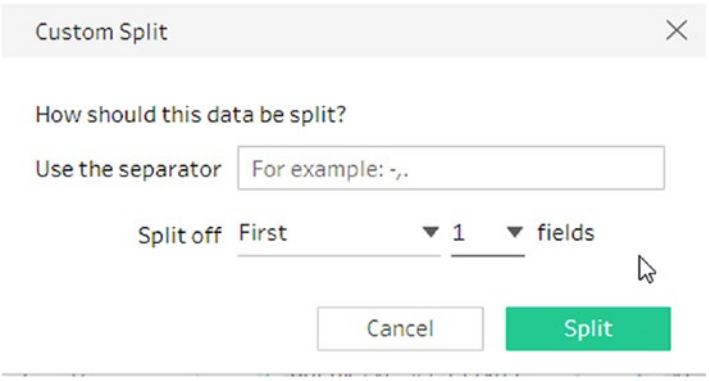



***Figure 7-13.*** *Custom Split*

One last note before we move away from the Split. If you scan through your fields, you will find that the original recordlocator card is still in your stack. In some cases, you might want to bring that original combined field into your analytics. In other cases, you will want to simply drop it at this point, so it won't be in your way. You can remove this field from the General Cleanup step (and from the data flow overall) by right-clicking it and selecting "Remove".

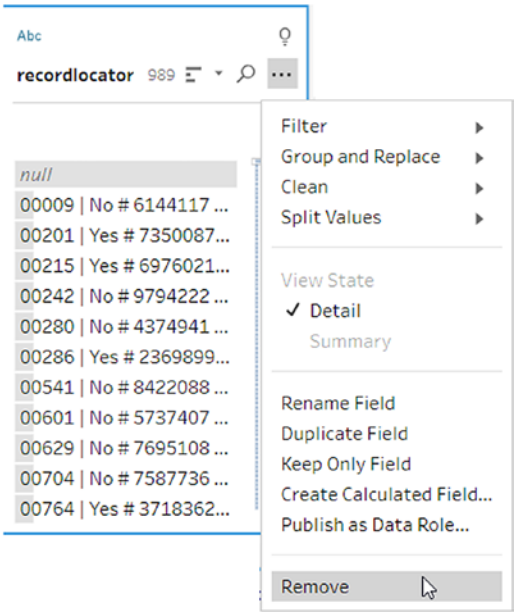



***Figure 7-14.*** *Remove field*

# Recommendations

Riding high on our success with splitting fields with the Automatic Split, we next move on to another tool built into Prep that makes our work ridiculously easy. The recommendations menu.

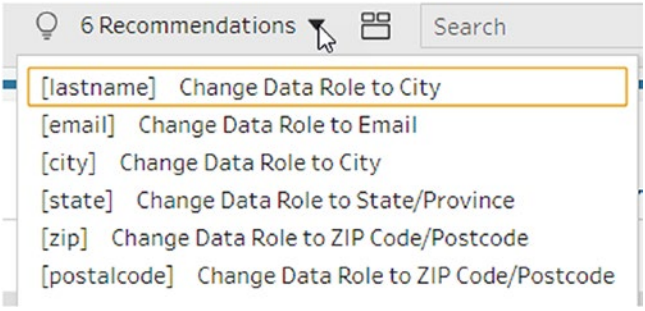



***Figure 7-15.*** *Recommendations*

Recommendations are Prep's way of giving us a nudge in the right direction when we are cleaning our data. If you click the drop-down menu next to Recommendations in the General Cleanup step, you will see six recommendations. The recommendations are listed in the sequence of the fields in the data source. In this case, the first field with a recommendation is "lastname". The recommendation here is to change the data role to City. As much as I'd like to tell you that Prep is never wrong, in this case it is clearly off base. I think we can safely ignore this first recommendation.

The next recommendation is to change the data role for the email field to Email. This IS a recommendation that we should act on. To make this change, we have a couple choices. We can click that item in the recommendations list; we can scroll to the email field and click the "Abc" label in the top left corner of its card and select the Email data role. A third option would be to click the lightbulb icon in the upper right corner of the email card. Here we will see a recommendation to change the data role to Email with a short note telling us what this means and a preview of some of the values in this field. If we select Apply, it will apply this recommendation for us and change the role of this field to Email.

At this point, you might be wondering what a Role is. Tableau has several built-in Roles that help it know how to treat some fields by default. For example, if you were in Tableau builder and you were to double-click a field called State that has the Role of State/Province, Tableau will automatically generate a map at that level of detail for you. Whenever possible, you want any field that can be assigned to a role to be assigned before you output the data to Tableau for analytics, it just makes everything a lot easier.

Let's continue with the example I just described and look into one of the more exciting features of Roles.

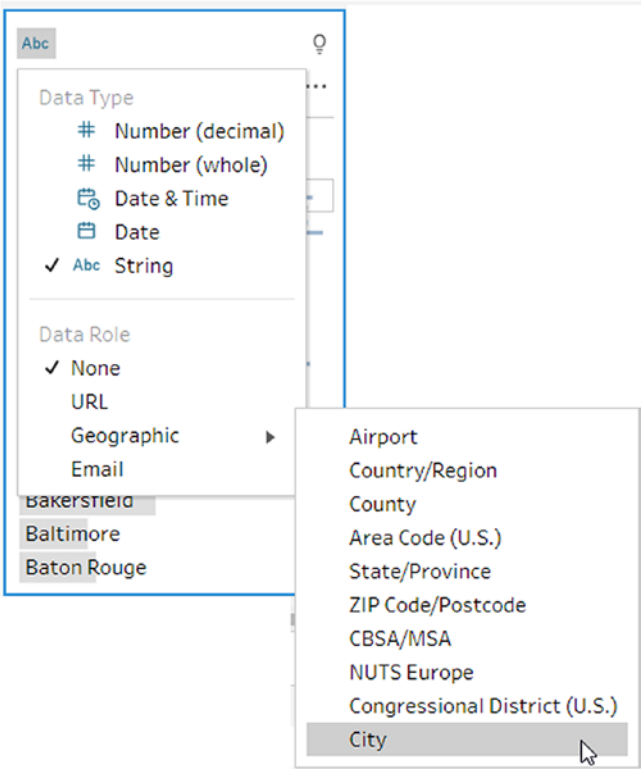



***Figure 7-16.*** *Assign Geographic Role*

8. Click the "Abc" icon in the top right corner of the "city" card
9. Click Geographic under Data Role
10. Click City

If you are following along, you will have just defined the contents of this field as a Geographic State/Province. This is good, but the point of this chapter is that dirty data happens. How much can we trust that all the values in this field are valid States (or Provinces)? In my opinion, not much. Let's do a little digging to see if we have a problem.

11. Click the ellipses (…) in the upper right corner of the city card

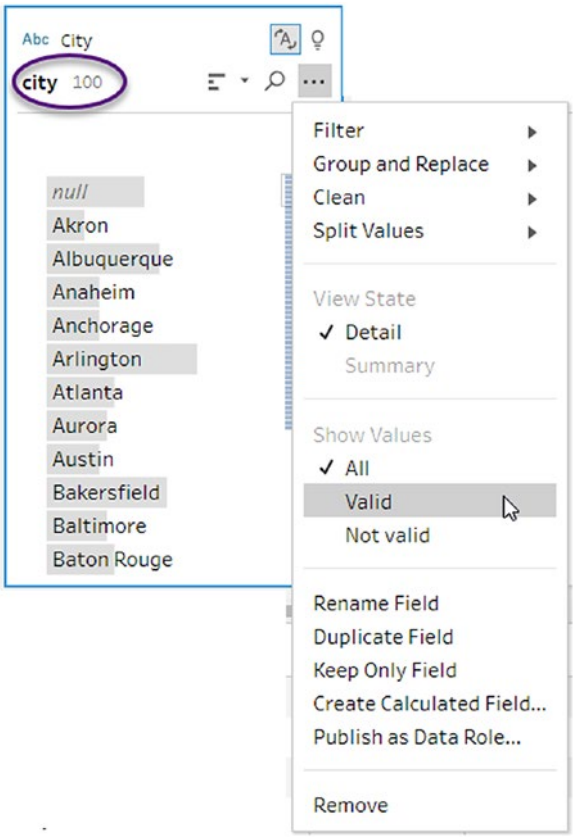



***Figure 7-17.*** *Validate data conforms to Role*

Before we continue, take note of the number 100 next to City in Figure 7-17. This indicates that there are 100 unique values in this field.

12. Click Valid

You will now see a filter called "Valid" has been applied to this field. In the background, Prep has compared every value in this field with its internal list of known Cities.

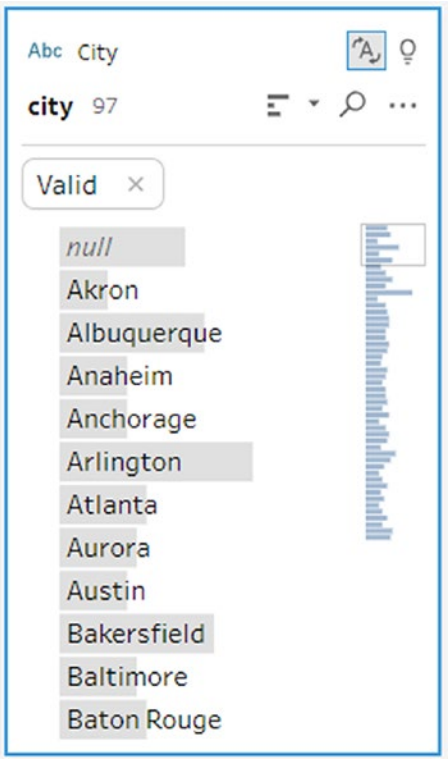



***Figure 7-18.*** *Filter to Valid cities*

The observant reader will have noticed that our count of unique cities has dropped from 100 to 97 when we filter to only Valid cities. This tells me we have a problem.

13. Click the x on the Valid filter to remove it
14. Click the ellipses and select "Not Valid" under Show Values

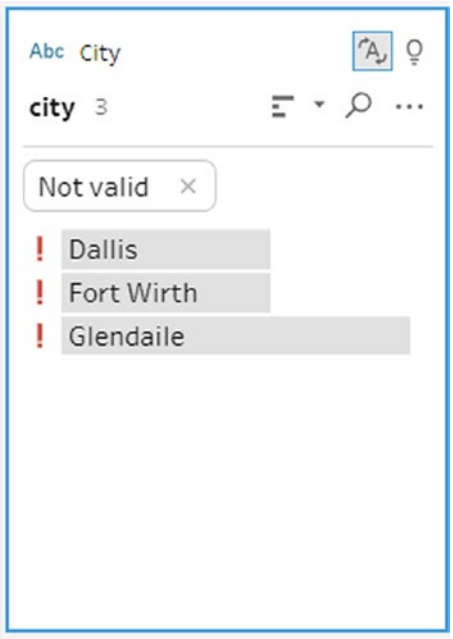



***Figure 7-19.*** *Invalid cities*

We have now filtered the contents of this field to three unique values that will not map correctly in Tableau Builder (they are not recognized by Tableau as valid cities, so they will not be associated with a valid latitude/longitude). Further, we can tell this problem affects 11 rows. Hover your mouse over each value, and you will see a tooltip showing you the total number of rows that share the value you are inspecting.

To correct this problem, simply double-click each value and type in a valid city.

15. Edit "Dallis" to "Dallas"
16. Edit "Fort Wirth" to "Fort Worth"
17. Edit "Glendaile" to "Glendale"
18. Click the x on the "Not Valid" filter to remove it from this card

As you made each edit, you will have noticed that the value you corrected dropped from the list. That is because the corrected value is no longer invalid. When you finished and cleared the "Not Valid" filter, you would be left with a list of 100 valid cities. You can verify this by applying the "Valid" filter again to see you have 100 valid cities.

To close out this section, we will make a last pass through our data cards and clean up our field names and data types.

19. Scroll through the data cards, and for each card, ask yourself if you like the field name presented

In many cases you will find field names are all lower case and have no spaces between words. We are preparing data for analytics, so at this point it is OK to give each field a name that will be more friendly to the consumers of our analytics. Rename all your fields so that words have spaces between them, acronyms are spelled out (whenever possible without creating long field names), and each word is capitalized.

20. Double-check your data types

At this stage we want to make sure every field is represented with the correct data type. You might find some fields like empID that are set as integers when in fact you would never want to include them in any mathematical operations. If you find fields like this, it is best to set them as strings. They will likely only be used as filters and labels in Tableau, and it will be easier for your analysts to work with them this way.

21. Delete fields you know will not be necessary for analytics

We've already done this a time or two, but removing fields is key to crafting a useful data source. There is a field toward the end of your card stack called "delete me". Go ahead and remove this field now.

## Handle NULL Values

NULL values are everywhere. Before we can discuss how you might handle NULL values, we should briefly explore what exactly NULL means. NULL is a special term used in many databases to indicate the lack of a value. A field marked NULL has no value at the time the record was last updated, but it could at some future time be updated to contain a value. NULL does not mean 0, missing, unknown, TBD, Invalid … it could be interpreted as any of those things, but strictly speaking NULL really means that a value could exist here but at this time that value is not present.

Why do we care about NULL? Because it can be a real pain when we try to aggregate data. In text fields it can make our labels and filters look strange. In numeric fields NULL will not be considered in any aggregate value. For example, if we want the average of a field over 100 records, but in 20 records that field is NULL, we would in fact only get the average of the 80 non-NULL records in a calculation involving that field.

With all of that in mind, one of the more important steps in cleaning data is making choices regarding the handling of NULL values. In many cases we will leave a NULL as it is. In some cases, we will want to replace the NULL marker with some value of our choosing. For example, one of the fields we created earlier in this chapter is named FavoriteColor. This field contains four values (Black, Blue, Red, Yellow) and NULL. As this is a text field, it is possible it would be used as a filter. We might not want NULL to be an option on that filter. Instead, the business stakeholders might prefer to see the value “Unknown” in cases where a record is not marked as Black, Blue, Red, or Yellow. We can replace NULL with Unknown by double-clicking NULL in the list of values in the FavoriteColor card and updating it to Unknown.


***Figure 7-20.*** *Handling NULL values*

# Filtering Records

The last topic we will address in the cleanup phase is filtering out records that we do not want to keep from our source data. In many cases, dirty data means that there are entire records in our source files that must be eliminated.

Scroll through the data cards and find the DeptID field we created in our Split earlier in this chapter. Let's imagine that the business stakeholders have informed us that any record with a NULL DeptID is invalid and should not be included in our analytics. Before we filter this record out of our results, let's look at what sort of impact this filter will have. Hover your mouse over the word NULL at the top of the list of values in the DeptID card. You will see that 12 records in this data set have NULL for DeptID.

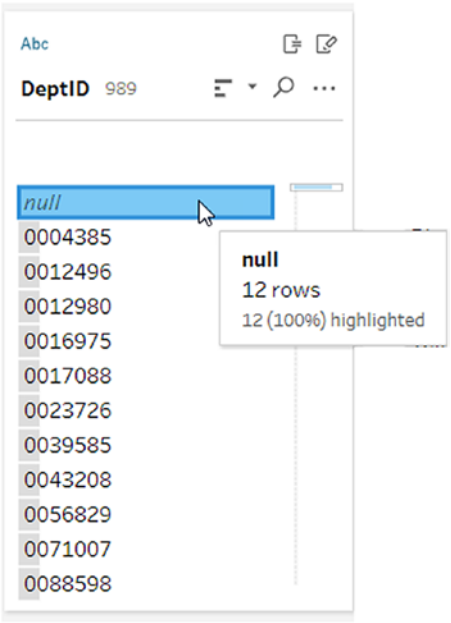



***Figure 7-21.*** *12 NULL records*

Also take note of the total number of records overall in our data at this time. You can find this information at the top left side of the audit pane (1K Rows).

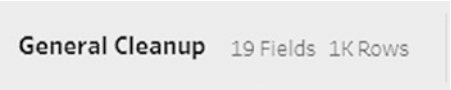



***Figure 7-22.*** *1K Rows*

Now you can right-click the NULL value at the top of the values list in the DeptID card and select "Exclude".

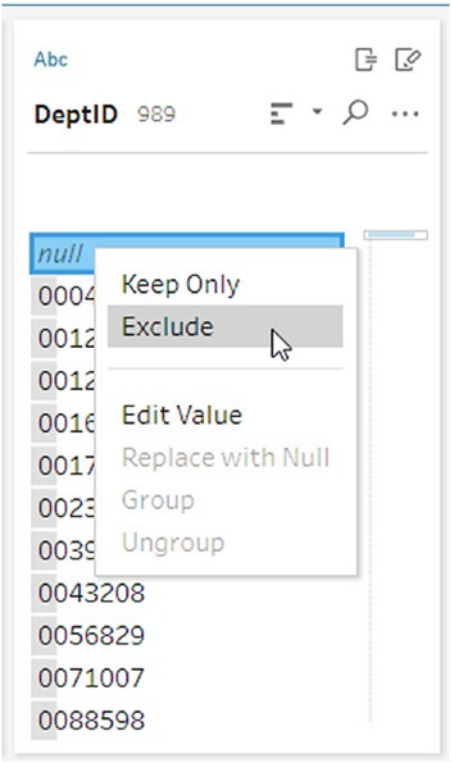



***Figure 7-23.*** *Exclude NULL to filter records*

A quick glance at the record count will now verify that we have filtered 12 records from our data set.

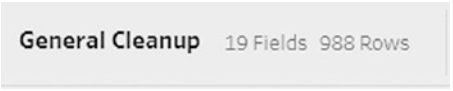



***Figure 7-24.*** *988 Rows*

# Summary

This has been a long chapter with lots of important information. To summarize it I am including a checklist of the things I focus on to clean every data source I encounter. These items are in no particular order, but all should be considered in every project.

- Rename any fields that are cryptic, are abbreviations, or lack capitalization for first letter of each word
- Verify all data types are correct
- Verify all fields that can be assigned to a Role have been
    - Apply the Valid and Not Valid filters to each field assigned to a role, and correct any invalid records that you can. If you can't correct a value, note it for follow-up with the business stakeholders.
- Split fields that contain multiple data points in one field
- Rename, set data types, and set the role for any fields created by Split
- Remove fields that are no longer needed
- Filter rows that are not required for analytics

# CHAPTER 8

# Group and Replace

Group and Replace is, in my opinion, one of Preps' most powerful tools. With it you can clean up some very bad data very quickly. I've been working with data for almost 20 years, and I've never seen anything quite like it in a tool like this.

To start this chapter off, let's open a new connection to the dirtyData file we worked through in the last chapter. I'll leave that to you; if you have trouble, you can refer to Chapter 3 for a refresher on connecting to data.

We will begin by adding a generic step just like we did in the last chapter. In a way, this short chapter is very much an extension of our conversation on cleaning dirty data. I broke it out to a chapter on its own because I want you to focus on this functionality apart from the other cleaning steps. This step is significantly more powerful and worth a little extra attention.


T. Costello and L. Blackshear, *Prepare Your Data for Tableau*,
https://doi.org/10.1007/978-1-4842-5497-4_8

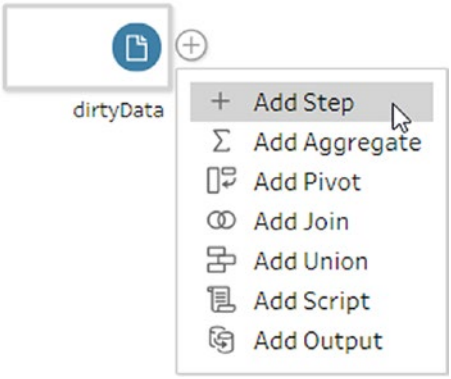



***Figure 8-1.*** *Add a step*

First, rename the step you just created from Clean 1 to Group Words. Renaming steps is tedious and requires attention, but it is critical for maintainability and will save you considerable frustration down the line.

Next, scroll through the cards until you get to the last field; it is called "word." This field contains 201 distinct values. The values in this field are a list of commonly misspelled words in the English language. We will use the Group and Replace tool to fix the data in this field, so misspelled words are grouped under correctly spelled words. This is basically magic. Of course, it's not REALLY magic. There are some sophisticated fuzzy matching algorithms at play in the background, but to my eye the results are nothing short of incredible. Let me show you what I'm talking about.

# Manual Selection

In keeping with the age-old tradition, I will start with the most difficult solution first, and we'll work our way to the easiest to implement solutions.

1. Click the ellipses on the upper right corner of the "word" card, and select Group and Replace
2. Click Manual Selection

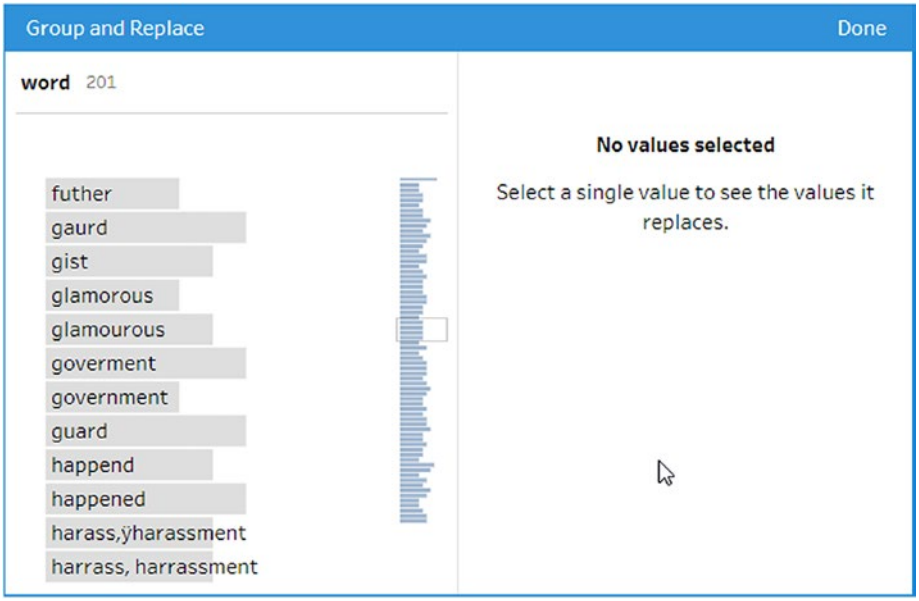



***Figure 8-2.*** *Group and Replace – Manual Selection*

3. Click government
4. Ctrl+click government

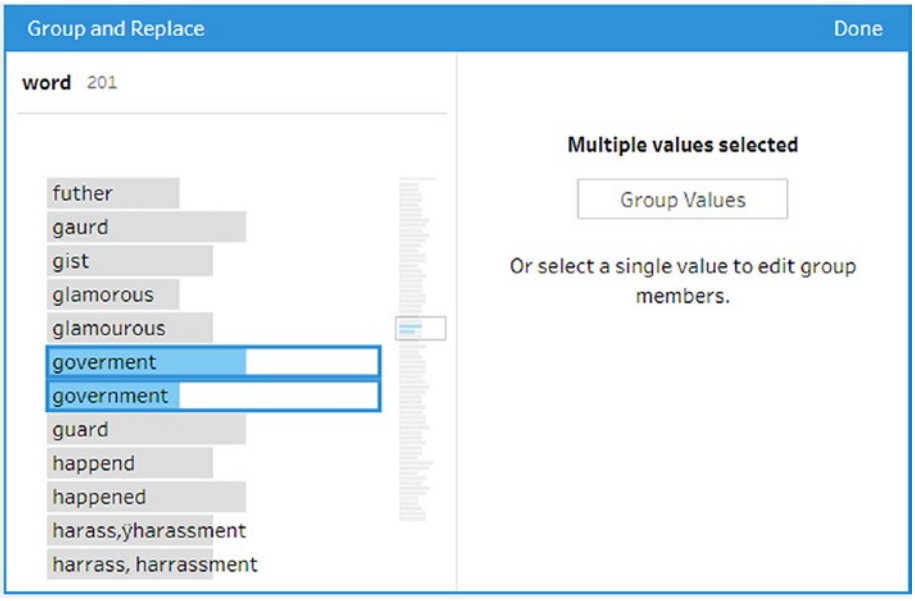



***Figure 8-3.*** *Grouping with manual selection*

5. Click Group Values

You have just created a manual grouping of the two values "goverment" and "government". You can tell you have created a group because now your word list contains the value for "government" only and it has a paperclip icon indicating that this is a group. Clicking the grouped label shows you the members contained in that group.

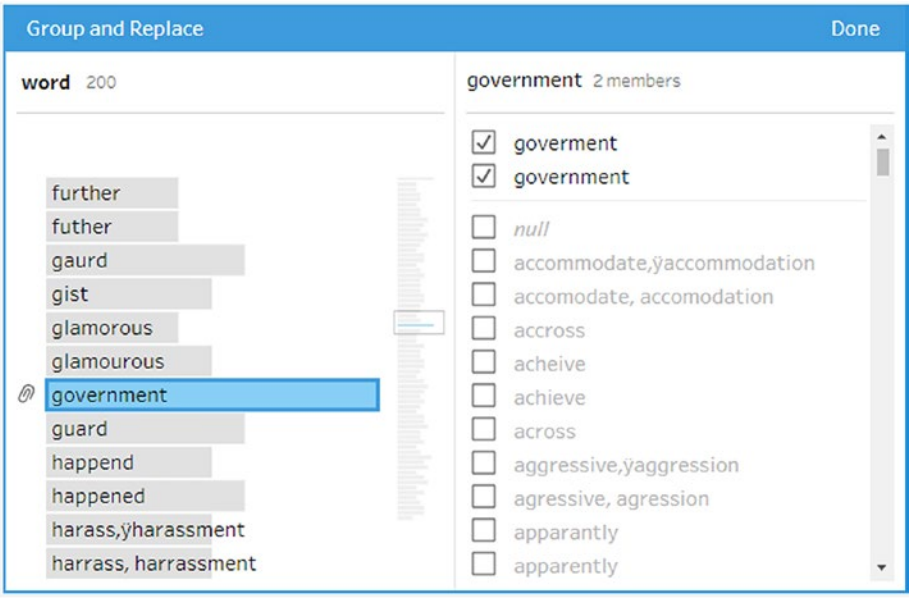



***Figure 8-4.*** *Government with two members*

---

The easiest way to add members to a group and control which member will be the name for that group is to select your group name from the list on the left and your group members from the list on the right.

If you select all your group members from the list on the left, the LAST value clicked when selecting members for a group will be the value that determines the name for the group.

---

# Pronunciation

The next three grouping options are a little more automated. This can be good, especially with large data sets, but it can also limit your ability to control members of each group.

The first of the more automated approaches we will look at is the Pronunciation grouping.

1. Go to the change log, and delete any changes you might have made in this step up to this point
2. Click the ellipses in the word card
3. Select Group and Replace
4. Select Pronunciation
5. Scroll to the top of the word list

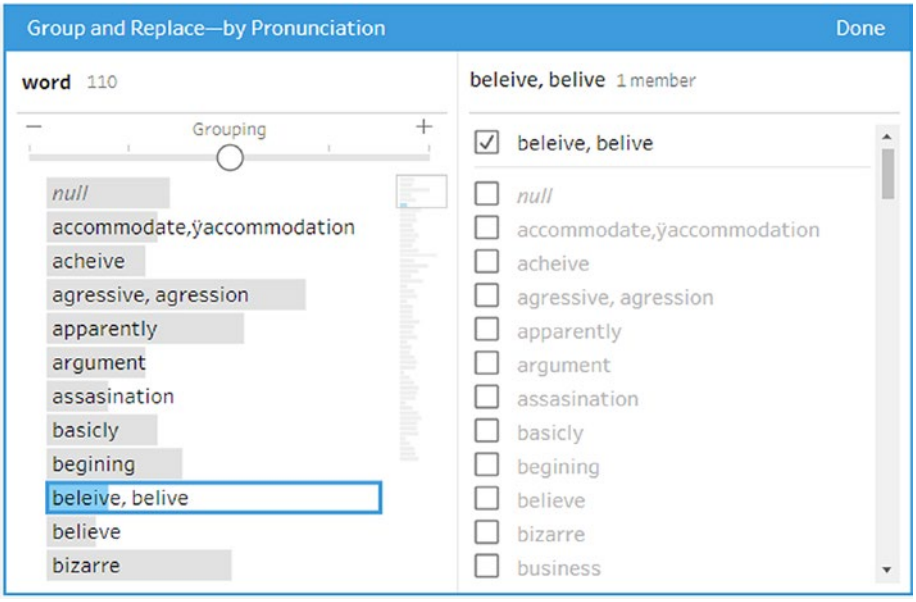



***Figure 8-5.*** *Pronunciation*

This grouping algorithm attempts to automatically create groupings based on common pronunciation of the words to be grouped. In Figure 8-5 you can see that, at the default setting beleive and belive have been grouped together, but believe is not a part of that group.

You can now adjust the grouping by interacting with the slider at the top of the values list.

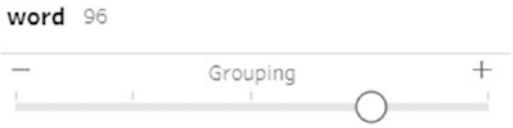



***Figure 8-6.*** *Accuracy slider*

Adjusting this slider to the right will (in general) bring more members into each group; adjusting to the left will usually remove members from groups. You should play with the slider, adjusting it left and right and watching how each move changes the content of the groups themselves. In this case, I found that moving the slider 1 point to the right (to the third of the four hashmarks on the slider scale) gave me the best results. Your results will vary with different input data that you attempt to group.

Once you have found the best level of grouping for your data, you will most likely have to scroll through the grouped values and adjust the names of each group. This can be tedious, but it is better than having to manually group a large set of members in a field.

## Common Characters

The Common Characters grouping offers a little less control over how detailed the groupings are, but it is overall a very accurate option in some cases.

1. Go to the change log, and delete any changes you might have made in this step up to this point
2. Click the ellipses in the word card
3. Select Group and Replace
4. Select Common Characters

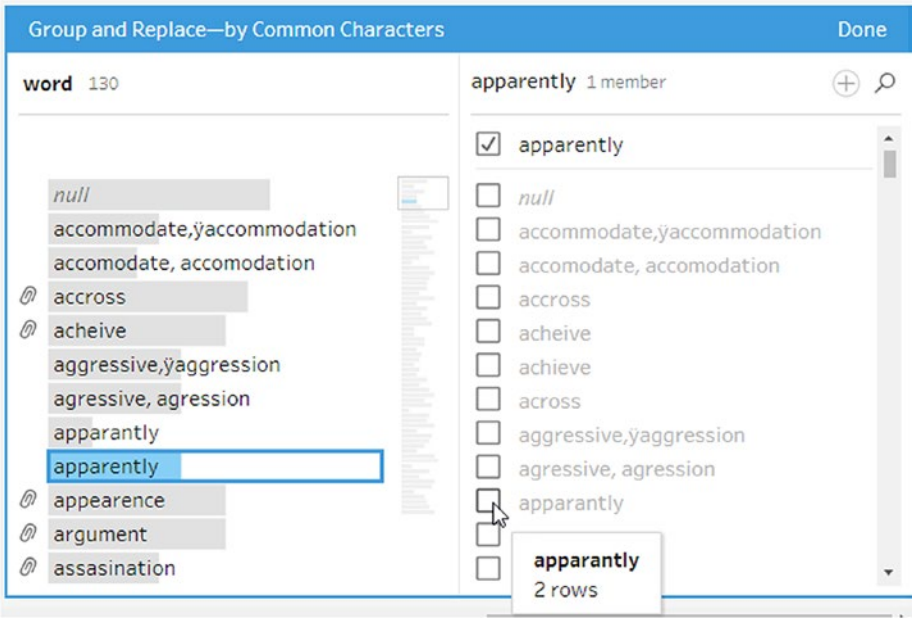



***Figure 8-7.*** *Common Characters*

This grouping method does not give you the slider to adjust accuracy, but it does let you manually add members to a group. Remember, to create a new group, select the member on the left that you want to be the name for your group, and then add members from the list on the right to be included in your group.

## Spelling

The last grouping type available to review is probably the best choice for us based on our data. The Spelling group is accessed just like the others, and once in it, you will find a very common interface. This grouping type gives you a slider control like the Pronunciation Grouping, and you can manually add or remove members to groups here just as you did with the other grouping types.

## Summary

Grouping is an extraordinarily powerful feature and one you should keep in mind when you are cleaning up your data. Having the ability to group our data based on manual groupings, pronunciation, common characters, or spelling makes it possible for us to replace bad values with good values in our data. I have used this feature frequently when cleaning up address information where you might find multiple variations in spelling for commonplace names. There is a lot of power in this feature, and while it can help you clean up dirty data very quickly, you would be well served to slow down when implementing this step and carefully observing the way the groups are assembled and the members of each group the tool creates for you.

# CHAPTER 9

# Aggregate

In a book full of things that are easier than you expected, none are easier than creating aggregated result sets. It's so simple, in fact I debated even giving this topic its own chapter. There really isn't much to do when you aggregate data, but it is something you want to think through before you do.

Why the caution? Why would it be rare to aggregate results in Prep? Simply because nine times out of ten you will want to deliver the most data possible to your analysts and let them create aggregates at design time in Tableau builder.

When would you want to create an aggregate in Prep? I've only found one compelling reason to do so. In some cases, you will have a data source that is very granular. For example, you might have readings from an automated process that runs multiple times per second. There are times when you will want to combine this very granular data with less detailed data (maybe another data source that was recorded once per hour). If you do a join of these two data sources, you will end up with a lot of duplicate records. In this case it is usually better to aggregate the detail record to show one record per hour, grouped by a field common to your less detailed record set. This makes joining the two sets much easier, and the results will be a lot smaller and faster to work with.

The downside to aggregation is that every field that persists after the aggregation must either be grouped or used in some mathematical aggregate like SUM, MIN, MAX, or AVERAGE. Let's work through an example to see what Aggregation is all about.


T. Costello and L. Blackshear, *Prepare Your Data for Tableau*,
https://doi.org/10.1007/978-1-4842-5497-4_9

# Aggregating Data

For this exercise we will be working with census data for the state of Texas. This data is recorded at the level of the State, County, and Zip Code. We will aggregate the results up to the level of the State first and then expand it out to include County-level records.

1. Open a new connection file texas.csv
2. Add an Aggregate step to your workflow

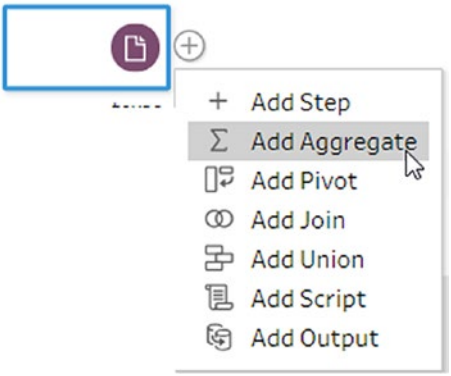



***Figure 9-1.*** *Add Aggregate step*

3. Rename your Aggregate step with a meaningful name. I've named mine "Agg to county".

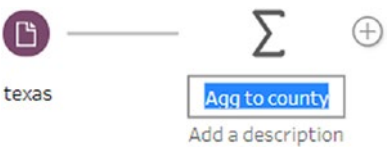



***Figure 9-2.*** *Rename Aggregate step*

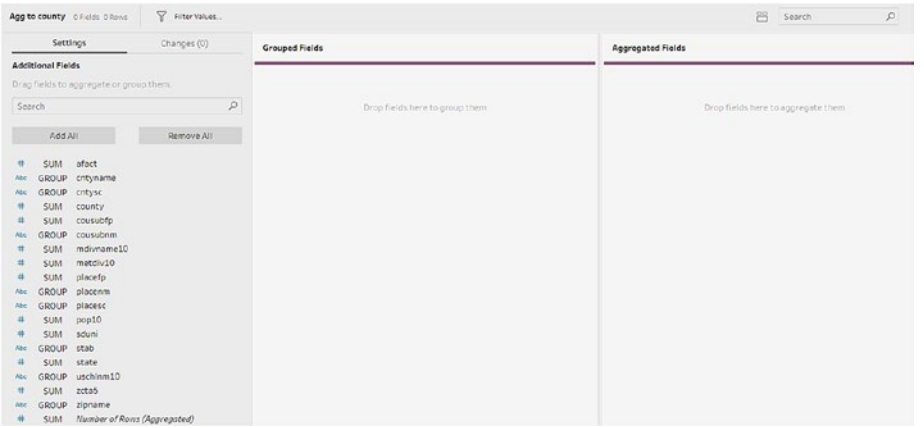



***Figure 9-3.*** *Configure Aggregate step*

4. Drag stab field to Grouped Fields zone
5. Right-click the value "State abbreviation" in the stab card on the Grouped Fields zone, and select "Exclude"
6. Double-click the stab field name in the Grouped Fields zone, and change it to "State Abbreviation"
7. Double-click the cntyname field name in the Grouped Fields zone, and change it to "County Name"
8. Double-click on the pop10 field name in the Aggregate Fields zone, and change it to "Population". Drag cntyname field to Grouped Fields zone.
9. Drag pop10 field to Aggregated Fields zone

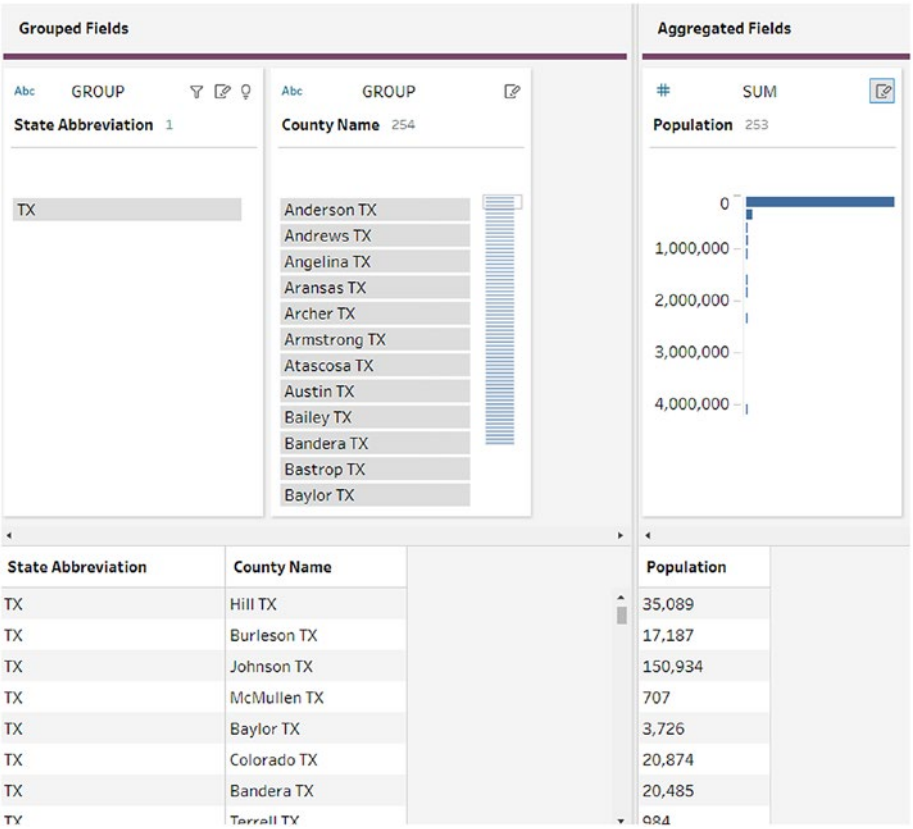



***Figure 9-4.*** *Aggregate Step with Group and Aggregate fields*

And that's really all there is to Aggregation. In the previous exercise, we added a step that reduced 18 fields and 10K rows down to 3 fields and 255 rows. The output from this step will now contain one row for each county in the Texas file with the population for each county.

If we want to further aggregate our results, we could remove the County Name field from the Grouped Fields zone. This would reduce the output from this step to two fields and one row!

## Summary

In this section we explored the Aggregate step. This step is not difficult to implement, but it is one that should be thought through before you use it. In most cases, you will want to share the most detailed record set possible with your analysts and let them aggregate the data as needed to tell the story of the data. In some situations, you will want to aggregate huge quantities of data to make it easier to analyze, or you might want to aggregate data to make it easier to join. As a rule of thumb, it is always better to have the data at the same level of detail on both sides of a join whenever possible.

CHAPTER 10

# Pivoting Data

Pivoting data is one of my favorite uses of the Data Prep tool. I've done this task in a variety of tools over the years, but none were as easy or as intuitive as Prep. As usual, we begin with a short discussion of when and why you might want to pivot (or unpivot) your data.

For me, the single biggest use case for pivoting data involves survey data. In almost all cases, your analytics will benefit from pivoting raw survey data. Why is this the case? Because frequently Tableau just works better when the data you are trying to analyze is oriented on rows rather than columns.

Here is an example. In this chapter we will be working with some crime data I found online. This data is a summary of crimes committed between 1997 and 2016 with the type of crime in columns and the count of crimes on rows.

**Crime in the United States**
by Volume and Rate per 100,000 Inhabitants, 1997–2016

Overview Data Declaration Download Excel

| Year | Population[1] | Violent crime[2] | *Violent crime rate* | Murder and nonnegligent manslaughter | *Murder and nonnegligent manslaughter rate* | Rape (revised definition[3]) | *Rape (revised definition) rate[3]* | Rape (legacy definition[4]) | *Rape (legacy definition) rate[4]* | Robbery | *Robbery rate* | Aggravs assau |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1997 | 267,783,607 | 1,636,096 | 611.0 | 18,208 | 6.8 | | | 96,153 | 35.9 | 498,534 | 186.2 | 1,023, |
| 1998 | 270,248,003 | 1,533,887 | 567.6 | 16,974 | 6.3 | | | 93,144 | 34.5 | 447,186 | 165.5 | 976, |
| 1999 | 272,690,813 | 1,426,044 | 523.0 | 15,522 | 5.7 | | | 89,411 | 32.8 | 409,371 | 150.1 | 911, |
| 2000 | 281,421,906 | 1,425,486 | 506.5 | 15,586 | 5.5 | | | 90,178 | 32.0 | 408,016 | 145.0 | 911, |
| 2001[5] | 285,317,559 | 1,439,480 | 504.5 | 16,037 | 5.6 | | | 90,863 | 31.8 | 423,557 | 148.5 | 909, |
| 2002 | 287,973,924 | 1,423,677 | 494.4 | 16,229 | 5.6 | | | 95,235 | 33.1 | 420,806 | 146.1 | 891, |
| 2003 | 290,788,976 | 1,383,676 | 475.8 | 16,528 | 5.7 | | | 93,883 | 32.3 | 414,235 | 142.5 | 859, |

***Figure 10-1.*** *Crime in the United States*


T. Costello and L. Blackshear, *Prepare Your Data for Tableau*,
https://doi.org/10.1007/978-1-4842-5497-4_10

This is interesting data, but it will be hard to work with in Tableau Builder. Currently there are multiple fields that each contains counts for different crimes. What we really want here is one field that contains all the types of crimes (a dimension) and another field that contains all the counts associated with each crime (a measure). To accomplish this, we need to pivot our data from an orientation on columns to an orientation on rows. The results will be fewer columns and more rows. That might seem confusing at first, but as you work through the examples, I think you'll see what I'm talking about.

# Pivot

For the rest of this chapter, we will be working with a slightly modified version of the data behind the screenshot displayed in Figure 10-1. I removed a few fields to make this data a little easier to work with; otherwise it is as I found it.

1. Create a new connection to the "2016 crime rates from fbi" Excel spreadsheet in your source files

This is the only Excel file I use as a source in this book, and it presents a challenge that you will often find when working with Excel. If you look at the source file, you will see that the first three columns are basically header information. This file was formatted to be read by a person, not by a machine. Prep is happiest when it is presented with source files that are strictly columns and rows of data with no fancy formatting. Unfortunately, that is rarely the state of the source files we receive in Excel.

| | A | B | C | D | E | F | G | H | I |
|---|---|---|---|---|---|---|---|---|---|
| 1 | **Table 1** | | | | | | | | |
| 2 | **Crime in the United States** | | | | | | | | |
| 3 | by Volume and Rate per 100,000 Inhabitants, 1997–2016 | | | | | | | | |
| 4 | Year | Population[1] | **Violent crime[2]** | *Violent crime rate* | Murder and nonnegligent manslaughter | *Murder and nonnegligent manslaughter rate* | Robbery | *Robbery rate* | Aggravated assault |
| 5 | 1997 | 267,783,607 | 1,636,096 | 611.0 | 18,208 | 6.8 | 498,534 | 186.2 | 1,023,201 |
| 6 | 1998 | 270,248,003 | 1,533,887 | 567.6 | 16,974 | 6.3 | 447,186 | 165.5 | 976,583 |
| 7 | 1999 | 272,690,813 | 1,426,044 | 523.0 | 15,522 | 5.7 | 409,371 | 150.1 | 911,740 |
| 8 | 2000 | 281,421,906 | 1,425,486 | 506.5 | 15,586 | 5.5 | 408,016 | 145.0 | 911,706 |
| 9 | 2001[5] | 285,317,559 | 1,439,480 | 504.5 | 16,037 | 5.6 | 423,557 | 148.5 | 909,023 |
| 10 | 2002 | 287,973,924 | 1,423,677 | 494.4 | 16,229 | 5.6 | 420,806 | 146.1 | 891,407 |
| 11 | 2003 | 290,788,976 | 1,383,676 | 475.8 | 16,528 | 5.7 | 414,235 | 142.5 | 859,030 |
| 12 | 2004 | 293,656,842 | 1,360,088 | 463.2 | 16,148 | 5.5 | 401,470 | 136.7 | 847,381 |
| 13 | 2005 | 296,507,061 | 1,390,745 | 469.0 | 16,740 | 5.6 | 417,438 | 140.8 | 862,220 |

***Figure 10-2.*** *Excel source file with formatted titles*

The good news is Tableau has included the Data Interpreter from Tableau Builder in the Prep tool. I have found the Data Interpreter to be an excellent resource for cleaning up Excel files so that I can work with them in Prep (and Builder).

In cases where Prep recognizes that a source file will need to be handled differently because of additional columns and rows of descriptive information above the data, you will be prompted with a checkbox marked "Use Data Interpreter". You will find this option below your new connection and above the list of tables in that connection.

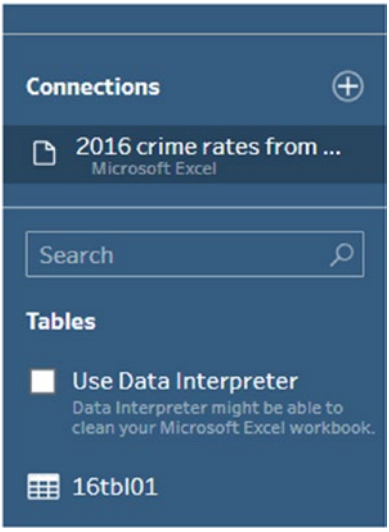



***Figure 10-3.*** *Use Data Interpreter*

2. Click the checkbox to "Use Data Interpreter"

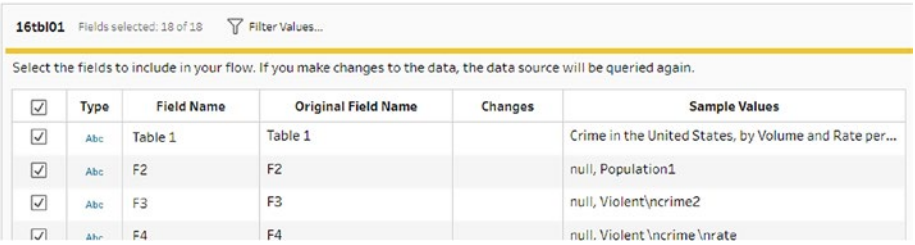
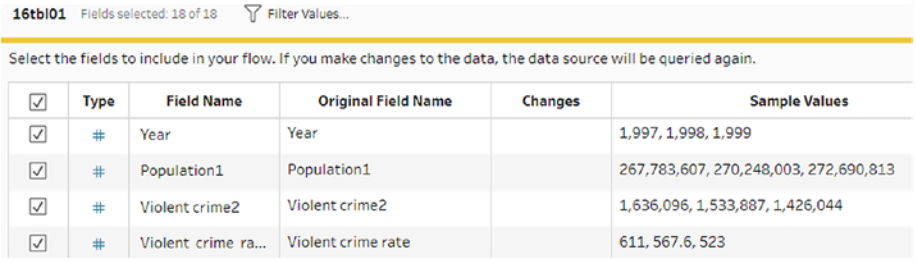



***Figure 10-4.*** *Data source before Data Interpreter*


16tbl01 Fields selected: 18 of 18 Filter Values...

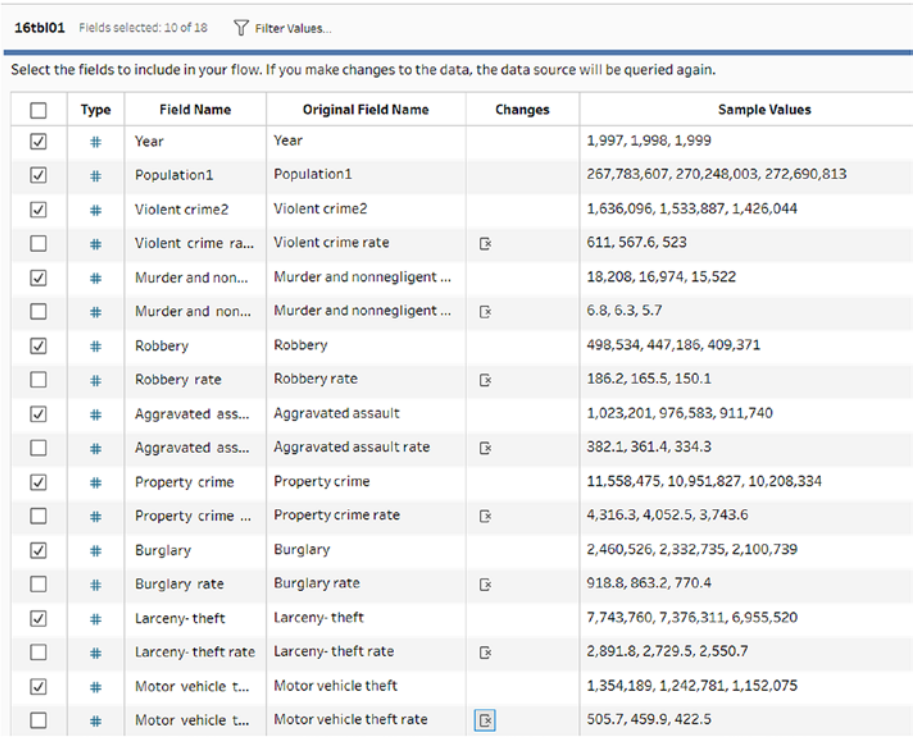
Select the fields to include in your flow. If you make changes to the data, the data source will be queried again.

| | Type | Field Name | Original Field Name | Changes | Sample Values |
|---|---|---|---|---|---|
| ☑ | # | Year | Year | | 1,997, 1,998, 1,999 |
| ☑ | # | Population1 | Population1 | | 267,783,607, 270,248,003, 272,690,813 |
| ☑ | # | Violent crime2 | Violent crime2 | | 1,636,096, 1,533,887, 1,426,044 |
| ☑ | # | Violent crime ra... | Violent crime rate | | 611, 567.6, 523 |

***Figure 10-5.*** *Data source after Data Interpreter*

Look closely at the screenshots in Figure 10-3 (Before Interpreter) and Figure 10-4 (After Interpreter). You will see that before we used the Data Interpreter, Tableau wasn't sure where the column headers were in the data. After Data Interpreter, Tableau was able to correctly identify the columns and rows that contain data. This is a HUGE improvement and saves us having to manually edit our source files or jumping through other hoops simply to access our data.

3. Scan through the filled list in the source connection, and uncheck any field that has the word "rate" in its Field Name (or Original Field Name)



16tbl01 Fields selected: 10 of 18 Filter Values...

Select the fields to include in your flow. If you make changes to the data, the data source will be queried again.

| Type | Field Name | Original Field Name | Changes | Sample Values |
|---|---|---|---|---|
| # | Year | Year | | 1,997, 1,998, 1,999 |
| # | Population1 | Population1 | | 267,783,607, 270,248,003, 272,690,813 |
| # | Violent crime2 | Violent crime2 | | 1,636,096, 1,533,887, 1,426,044 |
| # | Violent crime ra... | Violent crime rate | | 611, 567.6, 523 |
| # | Murder and non... | Murder and nonnegligent ... | | 18,208, 16,974, 15,522 |
| # | Murder and non... | Murder and nonnegligent ... | | 6.8, 6.3, 5.7 |
| # | Robbery | Robbery | | 498,534, 447,186, 409,371 |
| # | Robbery rate | Robbery rate | | 186.2, 165.5, 150.1 |
| # | Aggravated ass... | Aggravated assault | | 1,023,201, 976,583, 911,740 |
| # | Aggravated ass... | Aggravated assault rate | | 382.1, 361.4, 334.3 |
| # | Property crime | Property crime | | 11,558,475, 10,951,827, 10,208,334 |
| # | Property crime ... | Property crime rate | | 4,316.3, 4,052.5, 3,743.6 |
| # | Burglary | Burglary | | 2,460,526, 2,332,735, 2,100,739 |
| # | Burglary rate | Burglary rate | | 918.8, 863.2, 770.4 |
| # | Larceny- theft | Larceny- theft | | 7,743,760, 7,376,311, 6,955,520 |
| # | Larceny- theft rate | Larceny- theft rate | | 2,891.8, 2,729.5, 2,550.7 |
| # | Motor vehicle t... | Motor vehicle theft | | 1,354,189, 1,242,781, 1,152,075 |
| # | Motor vehicle t... | Motor vehicle theft rate | | 505.7, 459.9, 422.5 |

***Figure 10-6.*** *Uncheck all rate fields*

4. Double-click the Field Name Population1 and rename it to Population
5. Double-click the field name “Violent crime2” and rename it “Violent Crime”

In the previous two steps, you took advantage of a relatively new feature of Prep called “Edit Anywhere.” You will find that you can rename fields, change data types, and do many other common data cleanup tasks from almost all the Step types in Prep. I find this to be hugely convenient!

6. Rename the Source step to “Crime Data”
7. Add a Pivot Step

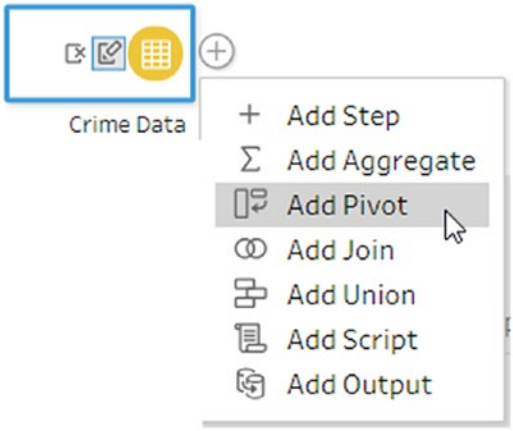



***Figure 10-7.*** *Add Pivot*

8. Rename the Pivot Step to "Pivot on Crimes"

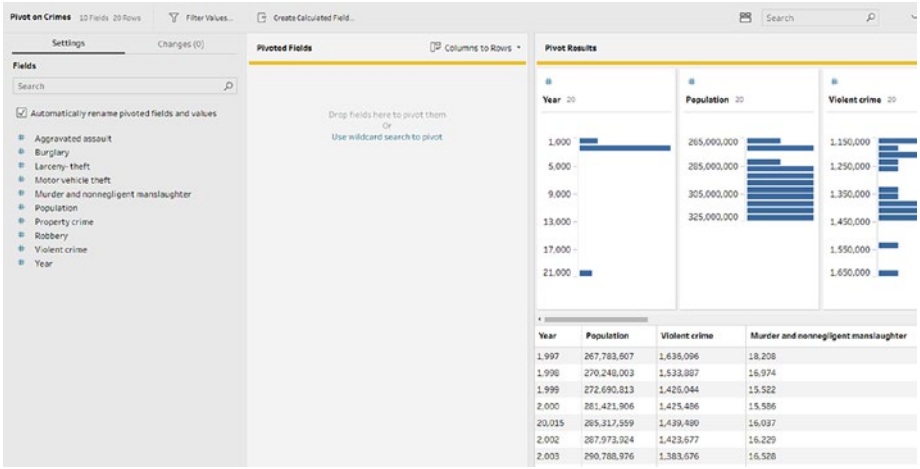



***Figure 10-8.*** *Pivot*

Before we continue, take a look at Figure 10-8, and notice that at the top of the Pivoted Fields drop zone, we see a marker labeled "Columns to Rows". It is here that we can change the behavior of our pivot to either pivot columns to rows or rows to columns. We will explore this feature soon.

9. Drag all the fields EXCEPT Year and Population to the Pivot1 Values drop zone on the Pivoted Fields card

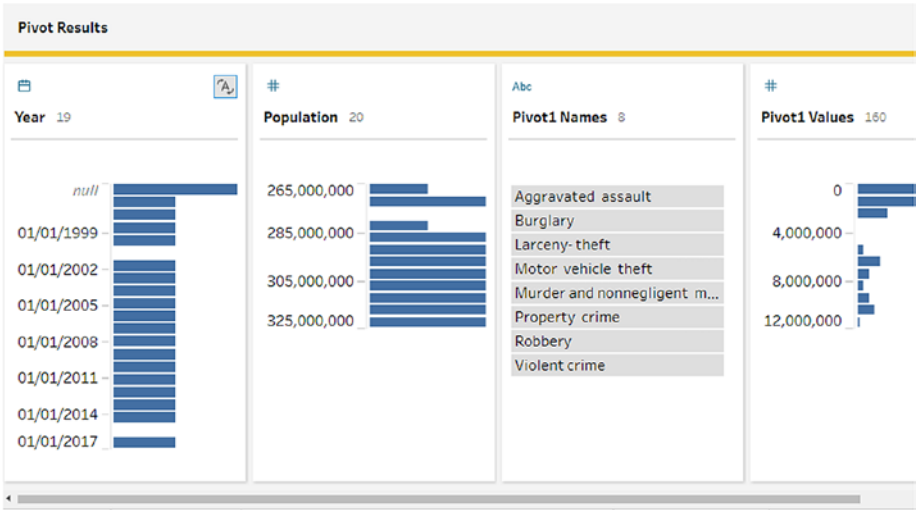

***Figure 10-9.*** *Pivoted Fields card*

10. In the Pivot Results pane, double-click the title of the Pivot1 Names card and rename it to "Crimes"
11. Double-click the title of the Pivot1 Values card and rename it to "Counts"
12. Change the data type of the Year card to Date
13. Click and drag the Year card to the first position in the Pivot Results pane
14. Click and drag the Population card to the second position in the Pivot Results pane

---

**Note** Reordering the position of the fields is not really a requirement for anything in Prep, I just find it sometimes makes it easier to validate the data when it is in an order that makes sense to me.

---




| Year | Population | Pivot1 Names | Pivot1 Values |
|---|---|---|---|
| 01/01/1997 | 267,783,607 | Motor vehicle theft | 1,354,189 |
| 01/01/1997 | 267,783,607 | Violent crime | 1,636,096 |
| 01/01/1997 | 267,783,607 | Robbery | 498,534 |
| 01/01/1997 | 267,783,607 | Property crime | 11,558,475 |
| 01/01/1997 | 267,783,607 | Aggravated assault | 1,023,201 |
| 01/01/1997 | 267,783,607 | Murder and nonnegligent manslaughter | 18,208 |
| 01/01/1997 | 267,783,607 | Burglary | 2,460,526 |

***Figure 10-10.*** *Pivot columns to rows*

Now that we have performed a Pivot of our data, we have a little more context to help us understand what happens when we pivot from columns to rows. In the previous steps, we started with 10 fields and 20 rows, and after the pivot, we will output 4 fields and 160 rows. We have reduced the number of columns, increased the number of rows, and introduced two new columns (Pivot Names and Pivot Values). These two new columns will be the Dimension and Measure that give us the flexibility we want in Tableau. Now, rather than drag ten fields on to columns in Builder, we can simply use the Pivot Names field and get ten values on our viz. This gives us a lot more flexibility and makes our views a lot less complicated to design and interpret.

# UnPivot

1. Create a new Pivot step after the Pivot on Crimes Step created in the previous exercise

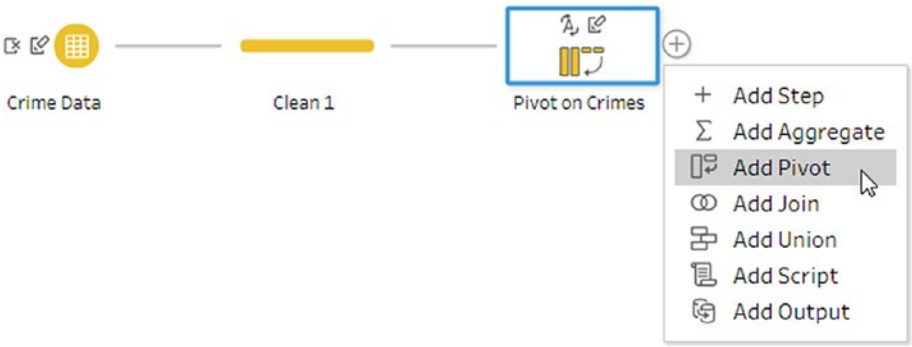



***Figure 10-11.*** *Add another pivot*

In this exercise we will build on the pivot we just created, but this time we will unpivot our data to create an output like the one we started with in our source connection. Of course, this is a bit contrived, but I find it easier to follow what is happening in the pivot and unpivot steps when I am familiar with the data.

2. Change the name of the Pivot2 Step to "UnPivot on Crimes"
3. Change the Pivot Type to Rows to Columns


***Figure 10-12.*** *Change Pivot type to Rows to Columns*

4. Drag Crimes to "Field that will pivot rows to columns"
5. Drag Counts to "Field to aggregate for new columns"

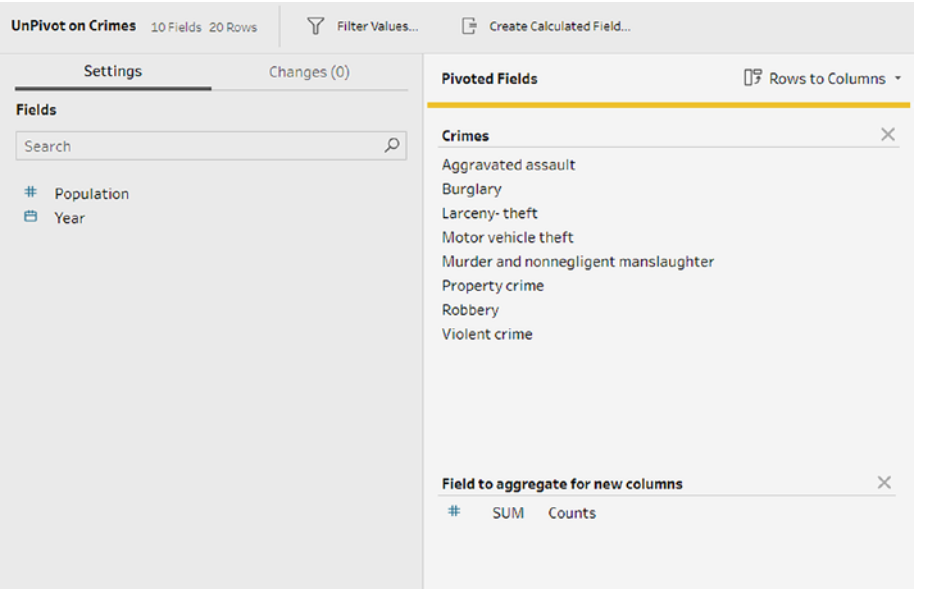



***Figure 10-13.*** *Pivot rows to columns*

Scroll through the Pivot Results, and you will find that this step has succesfully recreated the original shape of our source data; fields that were previously rows are once again columns. When we started the step, we had 4 fields and 160 rows. We now have 10 fields and 20 rows again.

Pivot Results

| # Motor vehicle theft | # Murder and nonnegligent manslaughter | # Violent crime | # Larceny-th |
|---|---|---|---|
| 1,198,245 | 17,309 | 1,435,123 | 6,626,363 |
| 1,246,646 | 16,229 | 1,423,677 | 7,057,379 |
| 723,186 | 14,856 | 1,217,057 | 6,168,874 |
| 1,152,075 | 15,522 | 1,426,044 | 6,955,520 |
| 795,652 | 15,399 | 1,325,896 | 6,338,095 |

***Figure 10-14.*** *Pivot rows to columns*

# Summary

Pivoting rows to columns and columns to rows gives us the flexibility to control the shape of our data.

When our data is oriented on rows, it makes it easier to see column headers as a single dimension in Tableau Builder. This makes creating a chart with a bar for each question in a survey simple to create where it might have been a mess if every question lived in its own dimension.

When our data is oriented on columns, it makes it easier to compare values with each other that would have otherwise been on separate rows. For example, we might pivot rows to columns so we have a column for significant date milestones, and we can do date math in Builder to determine the number of minutes, hours, days, months … between milestones. This could be accomplished with calculated fields (or sometimes table calculations) in Builder, but changing the shape of our data makes this kind of operation much easier.

Sometimes you will need BOTH shapes of data (oriented on columns, oriented on rows). Thanks to Prep it's simple to create alternate outputs that accommodate multiple analytic needs.

# PART III

# Load

Data Prep Builder gives us a lot of flexibility in how we output our data flows for analytics. In this section we will explore saving data to Tableau extracts, Hyper extracts, CSV files, and saved data sources in Tableau Server.

CHAPTER 11

# Output

The output step is very simple. As this is the last chapter in this book, this will not be the first time you've read of a step in Prep that is simple. Like the others, this step is simple to configure but can provide exceptional value if you think through how and when you want to use it. Yes, we do certainly want to have an output step at the end of every data flow, but that's not the end of the story for this step.

Let's start this chapter with an exercise, but unlike other exercises we've worked through this time, we will begin with one of the sample exercises provided by Tableau. This wouldn't be much of a Tableau book if we never looked at Superstore data, now would it?

## Exercise 11.1: Simple Output

1. Open a fresh instance of Tableau Prep Builder.
2. Click the Superstore sample under Sample Flows on the front page of the Prep.


T. Costello and L. Blackshear, *Prepare Your Data for Tableau*,
https://doi.org/10.1007/978-1-4842-5497-4_11

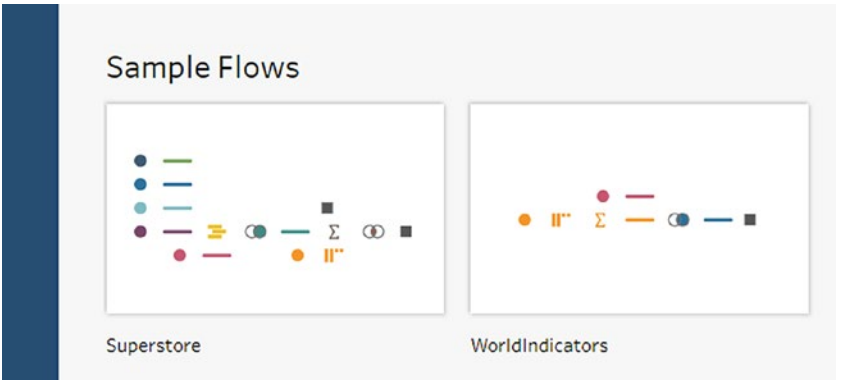



***Figure 11-1.*** *Superstore sample data flow*

You will now have the Superstore Sales data flow open on your desktop. This might look more complex flow than the ones we've played with in this book, but really it's all the same steps we've worked through, just in one flow rather than spread out over many. This data flow is more typical of the work that you will find yourself creating as you progress with Prep.

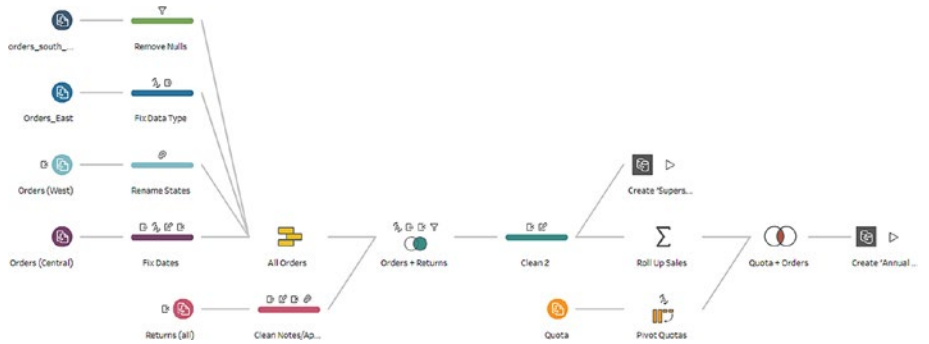



***Figure 11-2.*** *Superstore data flow*

Before we add an output step of our own, let's take a look at one that has been created in this flow and get a feel for how it works.

3. Hover your mouse over the Create "Annual Regional Performance.tde" output step on the far right end of the data flow

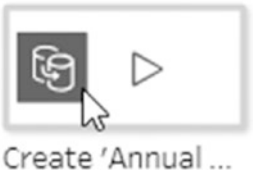


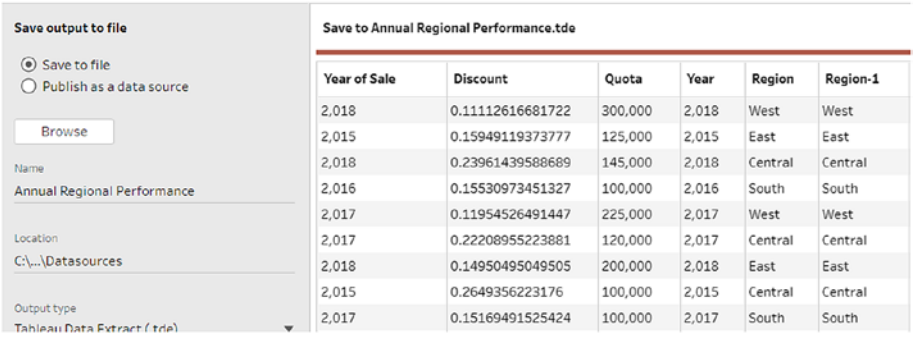

***Figure 11-3.*** *Create "Annual Regional Performance.tde" output step*

Before we continue with this exercise, I want to point out something special about this step. This is a step that you can interact with to run this data flow. If you click the triangle in this steps box, you will run the flow associated with this step. The flow associated with an output step includes all steps before it that are connected to it with a flow line (the gray lines you see in Figure 11-2).

In Figure 11-2 you will note that there are two output steps. Each output step has a control on it to initiate the flow associated with that step. This lets you execute elements of a large flow independently of the others in that flow. In the example we are looking at, there are two outputs. The first output is located just past midway of the data flow and creates a Superstore Sales.tde file. The second output is the one we are looking at in this exercise at the end of the flow. It creates a version of Superstore Sales rolled up to show annual sales.

Clicking the run flow icon on an output step will initiate that flow and output its data to whatever storage location you configured within that output.

4. Click the output step (not the initiate flow icon) for Create "Annual Regional Performance.tde"




| Year of Sale | Discount | Quota | Year | Region | Region-1 |
|---|---|---|---|---|---|
| 2,018 | 0.11112616681722 | 300,000 | 2,018 | West | West |
| 2,015 | 0.15949119373777 | 125,000 | 2,015 | East | East |
| 2,018 | 0.23961439588689 | 145,000 | 2,018 | Central | Central |
| 2,016 | 0.15530973451327 | 100,000 | 2,016 | South | South |
| 2,017 | 0.11954526491447 | 225,000 | 2,017 | West | West |
| 2,017 | 0.22208955223881 | 120,000 | 2,017 | Central | Central |
| 2,018 | 0.14950495049505 | 200,000 | 2,018 | East | East |
| 2,015 | 0.2649356223176 | 100,000 | 2,015 | Central | Central |
| 2,017 | 0.15169491525424 | 100,000 | 2,017 | South | South |

***Figure 11-4.*** *Output step*

The output step gives you a place to configure how you want to save your output and a preview of the output you will save.

Your options to save an output include Save to file and Publish as a data source. Saving to a file will open a file browser that will let you save your output to any local or shared resource you have access to (on your hard drive or your work network).

One thing I found a little confusing the first time I saved a file was the process used to name the file we are saving. By default, Prep offers the name Output. You will pretty much never use that name. To change the name of the output, you must click the Browse button and edit the name in the file browser that pops up. If you don't click the browse button, your output will save to the default location with the default name and the default output type.

---

**Note** The default save location (in Windows) is Documents\My Tableau Prep Repository\Datasources, the default name is output, and the default output type is Tableau Extract (.tde). Only the first output in your flow will save as output.tde, after that any additional outputs you create will increment the default name by one (output.tde, output1.tde, output2.tde …).

---

5. Click the Browse button

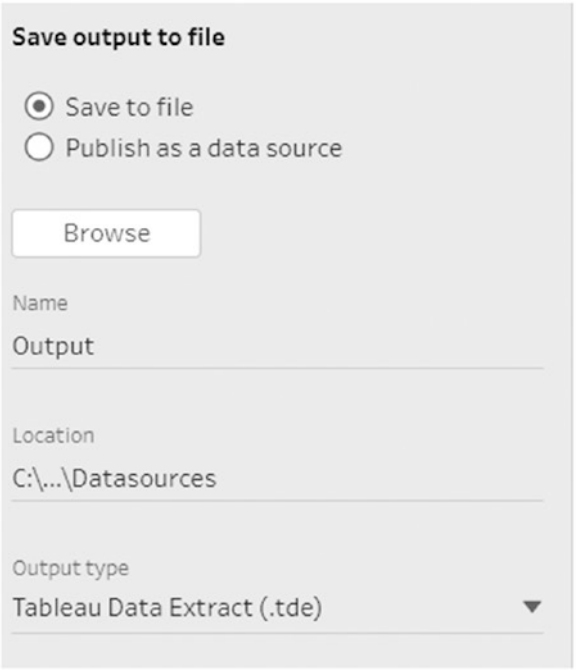



***Figure 11-5.*** *Configure output step*

6. Browse to a location to which you would like to save your output
7. Enter a file name for your output file
8. Click Accept
9. Click the Output type menu and select a file type for your output

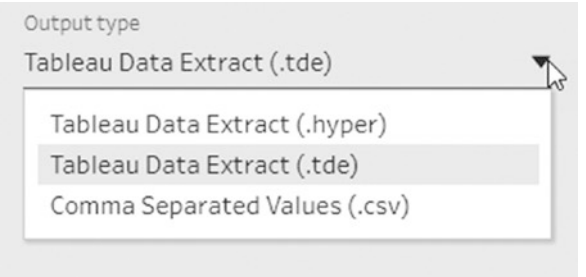



***Figure 11-6.*** *Output types*

# Summary

It's easy to create outputs for your data flow in Tableau Prep. One thing I'd like to call your attention to in the previous sentence is that I said steps, plural. In the example we explored in this chapter, there are two output steps. In some of my data flows, there are six or even ten outputs. I like to create intermediate output steps to give me a place to test my data at various points in the data flow. I like to create multiple outputs from the same source where each is aggregated, filtered, or shaped differently to support multiple audiences in my business community.

A last note on outputs that might not be obvious. I've taught Tableau classes for thousands of people all over the United States and Canada. One of the questions that comes up in almost every class is "how can I get data out of a Tableau extract?" Before Prep there just wasn't any convenient way to get data out of an extract, but now it's as simple as choosing an extract as your source and setting your destination to a csv file. Problem solved!

# APPENDIX A

# Preparing Data IN Tableau

You might be wondering, with a tool like Data Prep why would I want to prepare data in Tableau Builder? Well, there are a couple scenarios that come to mind. You might want to create a quick viz or two with a data source that you don't plan on using again after initial analytics. You might have a data source that you feel is ready for analytics without any prep, or you might have a data source that was already processed in Prep that you want to make minor changes to without going back to the prep phase. In any case, it is not uncommon to clean up and prep data from within Tableau Builder. In fact, until recently Builder was your only option!

In this Addendum I share my go-to checklist for prepping a data source in Tableau Builder, with some notes on each step to help you get started. You won't use every step in this list for every data source, but you should at least walk through the list every time to make sure you've covered all your bases.

## Tableau Builder Data Prep Checklist

- Do I want/need any filters at the connection level?
- Are all field names easy to read with spaces between words and each word capitalized?


T. Costello and L. Blackshear, *Prepare Your Data for Tableau*,
https://doi.org/10.1007/978-1-4842-5497-4

- Should I hide any fields?
- Are there columns that should be rows? (Pivot)
- Do any fields need to be split?
- Are all field values easy to interpret? (Alias codes and acronyms)
- Are all data types correct?
- Are all data roles correct?
- Are all defaults set the way I want them?
- Should I create a custom date?
- Do I want to create any hierarchies to share with this data source?
- Do I want to create any data groupings?
- Do I want to create any sets?
- Are there any fields that would benefit from comments?
- Do I want to organize the Measures/Dimensions with folders?
- Are there any calculated fields that I want to share with this data source?
- Do I want a live connection or an extract?

# Sample Data

We will use one data source for examples throughout this chapter. I found this data on the web site make MakeOverMonday.com. This is a fun web site for exploring new data sets in Tableau. I've learned a lot of cool tricks here over the years: https://www.makeovermonday.co.uk/.

1. Open Tableau Builder.
2. Click Text file under Connect➤To a File

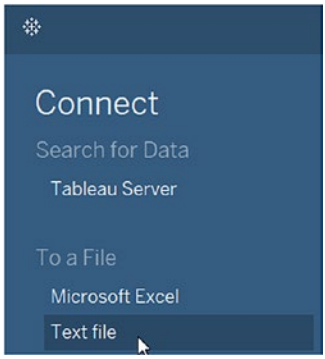



***Figure A-1.*** *Connect to a text file in Builder*

3. Browse to the resource files for this chapter
4. Open "north_america_bear_killings.csv"

# Filter

One of the most important decisions you will make when preparing a new data source for Tableau is whether to apply filters and what kind of filters to apply. In the context of the data source itself, we are only focused on two filter types, the extract filter and the data source filter. To give you some context into how filters are applied, see Figure A-1. This image comes directly from the Tableau online help, and it is one that I refer to often when troubleshooting performance in Tableau dashboards.

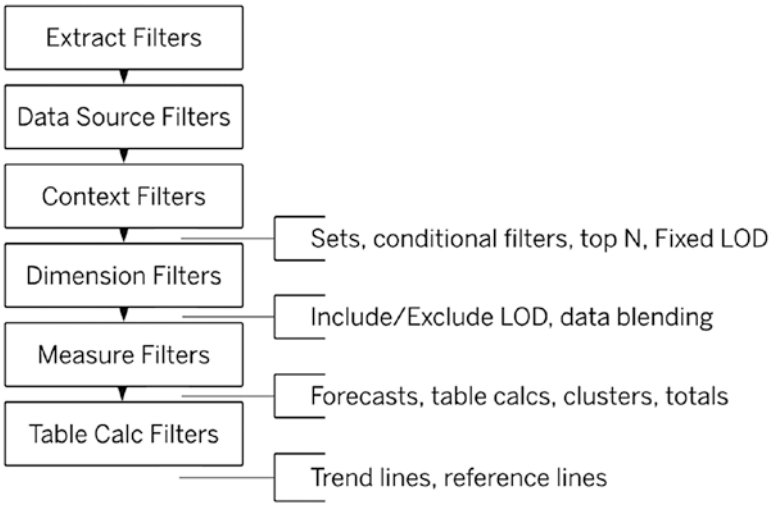



***Figure A-2.*** *Filter order in Tableau*

We will explore extract filters at the end of this chapter when we look at extracts in depth. That leaves us with Data Source filters to consider.

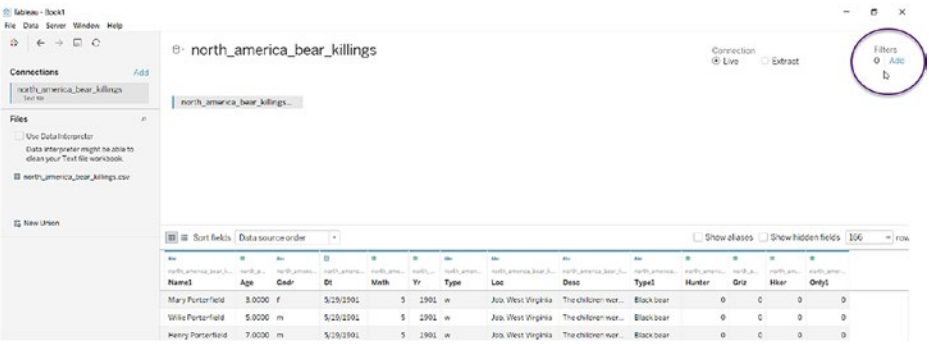



***Figure A-3.*** *Data Source filter*

You will find the count of existing data source filters and the link to add new data source filters at the top right corner of the data source tab (see Figure A-3).

Once clicked, you will be presented with a dialog where you can add a filter. When you click Add, you will be given a list of every field in the connection. For each data type, you will be given slightly different options for configuring the filter. If you've spent any time building in Tableau, this will feel very familiar.

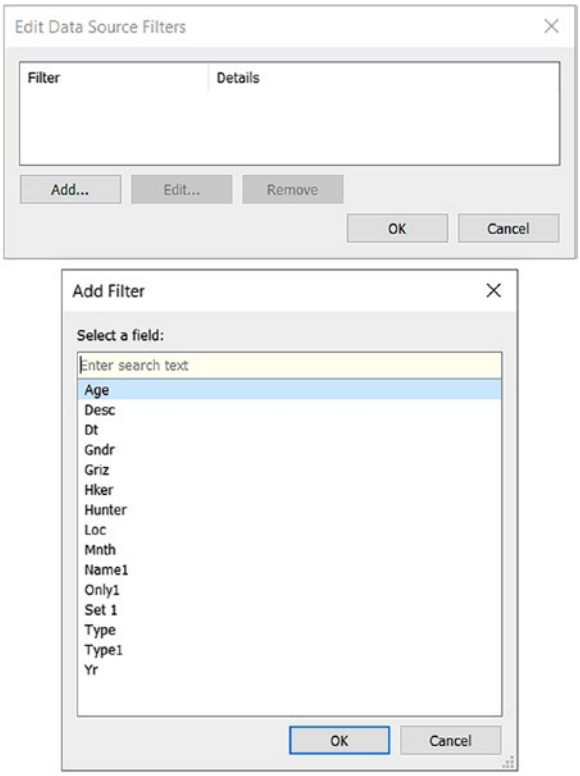



***Figure A-4.*** *Add Data Source filter*

There are many cases where you would want to add a data source filter. If you have a very large data source but you know that you will only want to interact with a small amount of that data, it makes sense to filter before you begin. For example, you might have data from around the world in your connection, but for this data source, you want to limit analytics to the

southern region of the United States. In this case it makes perfect sense to apply a data source filter. Use caution; however, once applied a data source filter can be subtle. Your analysts might or might not realize they are working with a filtered data source. This could cause some confusion if you don't communicate the intended use for each data source clearly.

I like to make the purpose of my connections explicit by including a brief description of any filters applied in the name. You can do this by double-clicking the name of the data source ("north_america_bear_killings" in Figure A-3) and renaming to something that indicates a filter has been applied. In this case I might filter to data from 2015 to 2019 only and rename the connection to "North_America_Bear_Killings_2015-2019".

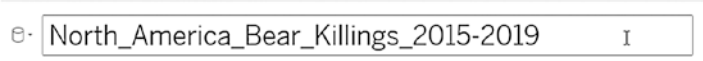



***Figure A-5.*** *Rename connection*

# Hide Fields

After applying filters, hiding fields is one of the biggest improvements you can make to most data sources. Practically every data source includes fields that are not useful for analytics. These might be housekeeping fields from a database (inserted by, inserted on, updated by, updated on), or it could be ID fields that would be meaningless to your business community. To hide a field in a data source, click the drop-down menu in the upper right corner of that field and select "Hide" (see Figures A-6 and A-7). For this section, hide the fields Mnth and Yr. We won't need them as Tableau gives us this level of detail based on the Date field.

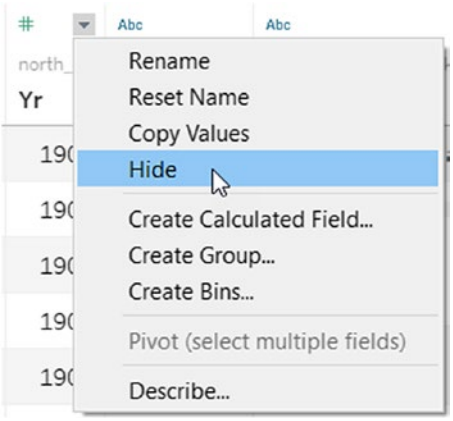



***Figure A-6.*** *Hide Field in Data Source*

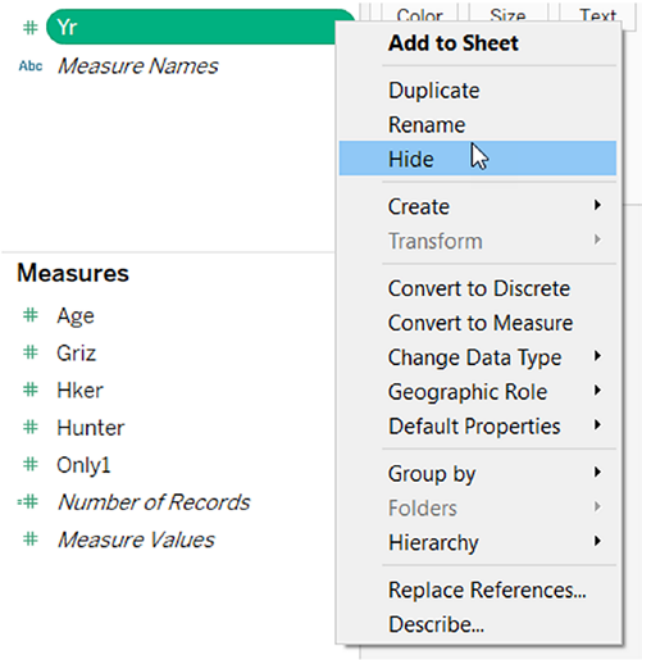



***Figure A-7.*** *Hide Field in Worksheet*

# Rename

Renaming fields is critical for creating a data source that is easy to work with. We could move this step anywhere in the checklist, and it could be accomplished in the data source tab or on a worksheet. I address it first in this list because soon we will want to Pivot a few of our fields, and it makes sense to rename everything before we do that Pivot.

Renaming fields couldn't be easier. Simply double-click a field name and type in the name you prefer. I've "dirtied up" this data source to give you a chance to rename a couple fields. Follow along with the next exercise to rename your fields before we continue, so we can stay in synch as we move through this chapter.

1. Rename Name1 to Name
2. Rename Gndr to Gender
3. Rename Dt to Date
4. Rename Loc to Location
5. Rename Desc to Description
6. Rename Type1 to Type of Bear
7. Rename Griz to Grizzly
8. Rename Hker to Hiker
9. Rename Only1 to Only 1

# Pivot

Many of the steps in this chapter can be completed from the data source tab or from within a worksheet in Builder. I prefer to do most of my work in the worksheet, but Pivot is an example of a task that can only be completed in the data source tab. When we Pivot, we shift some fields

from an orientation on columns to rows. In this case we have four fields at the end of our field list that would be easier to work with if they were all consolidated into two new fields, one field containing the column header and another containing the column value. This might seem confusing at first, but once you've completed the following example, I think it will make more sense …

1. Click the data source tab in Builder (bottom left corner of the Builder window)
2. Scroll through the list of fields. Notice the last four fields in the list (Hunter, Grizzly, Hiker, Only 1)? We want to pivot these values from columns to rows.
3. Click the Hunter column (it should highlight blue when selected)
4. Shift+click the Only 1 column. All columns between Hunter and Only 1 should now be highlighted.

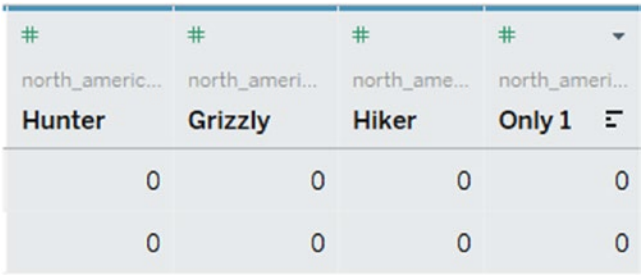



***Figure A-8.*** *Select columns to Pivot*

5. Click the drop-down menu in the top right corner of any of the selected fields, and choose Pivot

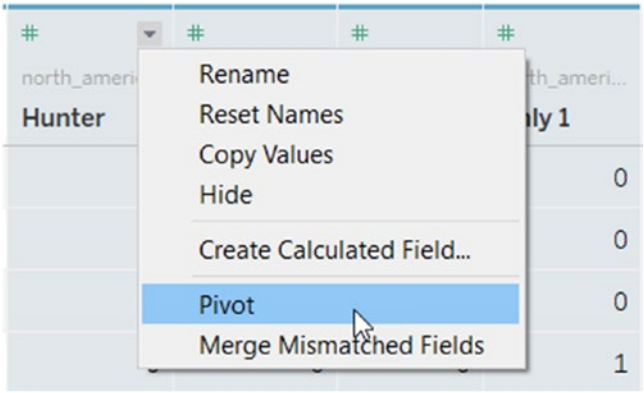



***Figure A-9.*** *Pivot*

You have now pivoted the four columns you selected into 664 rows. This change replaced your original columns with a new column called "Pivot Field Names" that contains the values that were previously field names and "Pivot Field Values" with a column that now contains the value that was previously associated with each column. A simpler way to describe this change is we converted 4 columns and 166 rows into 2 columns and 664 rows.

If this is your first time playing with a Pivot, I encourage you to explore what just happened. This is an extremely powerful tool. There are some visualizations that would be very difficult to create without a pivot of source data.

6. Rename "Pivot Field Names" to "Attack"
7. Rename "Pivot Field Values" to "Count"

# Split

In many cases you will find multiple values saved within one field, often separated with a comma or some other character. Tableau has a great built-in tool that makes it easy to split these values out into multiple fields.

For example, in our data we have a field called Location. This field contains the City and State together in one field. This causes us a couple problems. First, it would be difficult to filter our data to just one city or just one state. More critical in this case, this field will not be recognized for geocoding, so Tableau will not be able to map the city or state from our data. Fortunately, this is easy to fix. Simply click the drop-down menu at the top right corner of the Location field and select "Split". In this case, Tableau will recognize that the Location field contains two pieces of data separated by a comma. It will automatically create two new fields for us called "Location – Split 1" and "Location – Split 2". Go ahead and rename these fields to City and State.

If you have a more complex split scenario, you might try the custom split. It gives you the ability to designate the character on which to split and lets you choose to split out the first, last, or n (where n is any number between 1 and 10) fields. Split is a powerful tool. You might find that you need to run the split a couple times to drill down to exactly what you want (split a field one way and then split the resulting field another way).

In closing, please note that while Split does give us new fields with the split-out contents of our starting field, we still have that starting field available to us for analytics. This original field might be useful as a label in our analytics (in which case we would keep it), or it might be a field we want to hide.

## Alias Contents

Just as we often rename fields, it is very common that we will need to alias the contents of fields.

I find the data source tab a great place to get familiar with a new data source. One of the first things I do when getting to know a new data source is to expand the range of the preview to around 5,000 rows. If my data source has less than 5k rows, it will display all rows in the preview; if it has more, I can usually get a feel for a data set in the first 5k rows.

With enough rows displayed in the preview, I start by looking for fields with cryptic values. If I see fields with acronyms, single letter values, or codes of some sort, I try to determine if analytics would be easier if these records had more intuitive values.

For example, in our data source, there is a field called Type with values w and c. After a quick chat with the business stakeholder responsible for this data, I might learn that w = wild and c = captive. Rather than leave these values as they are and expect the consumers of this data source to know what they mean, I would next set aliases to make their value easier to interpret. To set an alias, click the drop-down menu at the top right corner of the field and select "Aliases …".

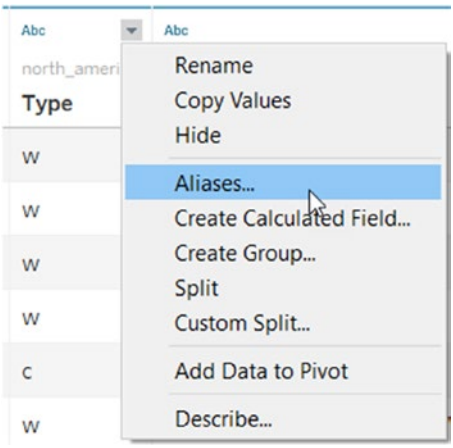



***Figure A-10.*** *Aliases …*

This will open an Alias editor from which you can see the original value of a field, a marker that indicates if an alias has been set and the alias for each field. Make special note of the button at the bottom of this dialog. You can click "Clear Aliases" to remove all Aliases from this field through this button.

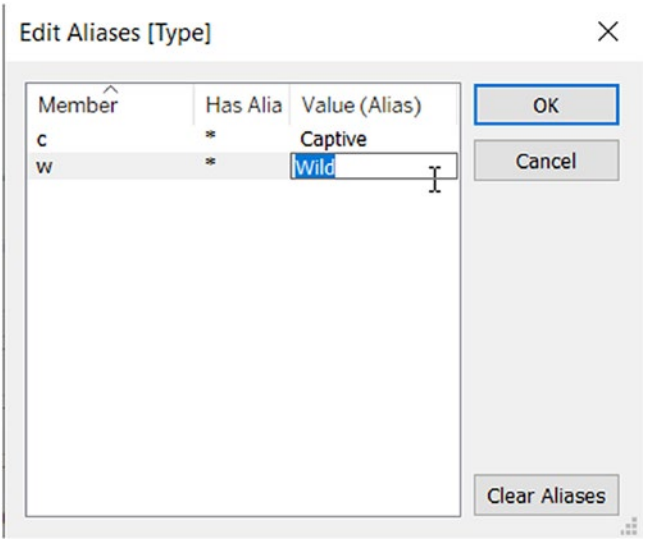



***Figure A-11.*** *Edit Aliases*

Next do the same to set Aliases for the Gender field (set f to Female and m to Male). You will find this does not work. Go ahead and cancel out of the Edit Aliases dialog. We will fix this issue later in this chapter.

---

**Note** Almost nothing you do in Tableau Builder or Tableau Prep will change the source data. There are a couple new features recently introduced in Tableau Builder that could affect change in a source system, but nothing discussed in this book will change your source data. All changes you make in this book are meta-data changes and are saved with the Prep data flow, the Tableau workbook, or the Tableau saved data source.

---

# Data Types

By default, Tableau will try to select the data type that makes the most sense based on a quick look at the data in your connection. I'm not sure exactly how many records are polled to make this determination; if I were to guess, I would think between 500 and 1000 records. The number of records examined can be adjusted in Tableau Prep, potentially making Prep slightly more accurate at picking data types than Builder. In any case, it is smart to review the data types for all fields and make sure you agree with Tableau's choices. At a minimum you want to make sure the data type selected in Tableau agrees with the data type in the database for fields you might use as filters, and data types should be compatible between connections that you want to Blend.

To change the data type of a field in a worksheet, click the data type icon to the left of the field name in the data pane, and pick the data type you want from the drop-down menu.

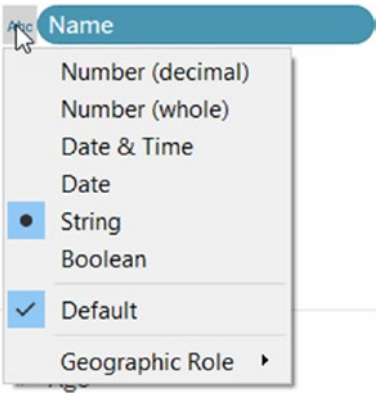



***Figure A-12.*** *Changing a data type in data pane of a worksheet*

# Data Roles

When we think of data roles, we are usually thinking of geographic roles, but recently Tableau has introduced additional roles that can be useful. For example, there is an email role that can be used to ensure that all data in a field conform to an expected definition of an email address.

For this chapter, we will focus on the geographic roles. When you connect to a new data source, Tableau does several things in the background. One thing it does is to look at every field name and match those names in your data to names it knows in an internal database. If it finds names like City, State, and Postal Code, Tableau will compare the contents of your fields to known values in its internal database. When it finds matches, it associates a latitude/longitude pair to your every matching record in your data for that field. This is how Tableau knows how to map the location of Dallas Texas when you have a field called City with the value Dallas in your data.

In our data set, we have two fields that should be mapped to geographic roles that are currently considered as strings by Tableau. Before we map them, let's look at why they were not mapped by default for us. The fields in question are City and State. If you recall, these are the fields that we split in an earlier section of this chapter. Take a look at the data type next to the field name for these fields in the data pane of a worksheet.


***Figure A-13.*** *City field in Tableau*

Notice the = sign next to the Abc icon. This tells us this is a calculated field. If you right-click this field and select Edit, you will see the calculation the Split command created for us when we split this field.

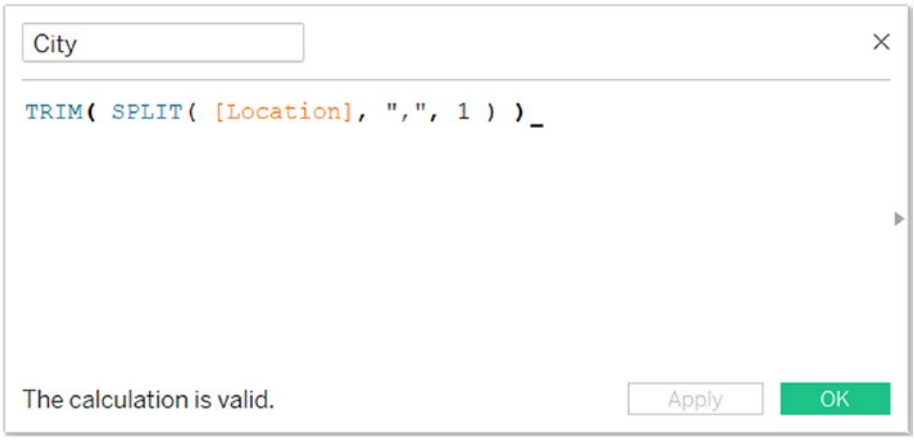



***Figure A-14.*** *Calculated field for City*

The main reason Tableau didn't give us the geocoding for this field when we connected to our data source is that this field didn't exist when we made that connection. We created this field later. Fortunately, this is easy to fix.

Right-click the City field in the data pane on a worksheet, and select Geographic Role➤City (see Figure A-15). Next, do the same for the state field, this time choosing the State/Province Role.

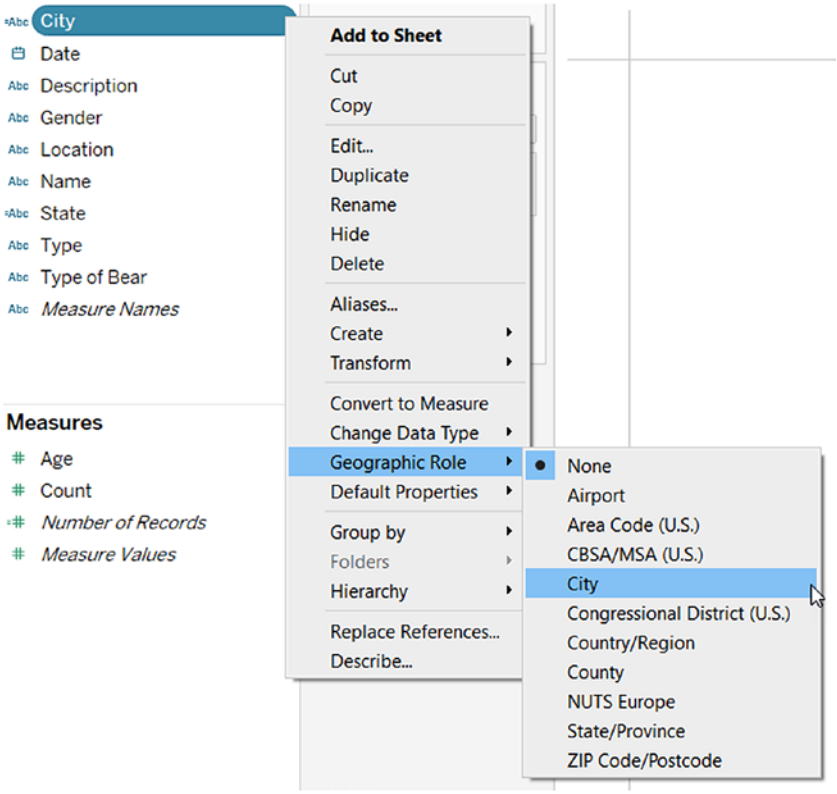



***Figure A-15.*** *Assign Geographic Role of City to the City field*

---

**Note** For the remainder of this chapter, all the changes we make will be done from the data pane on a worksheet in Tableau Builder. Some of these changes COULD be made from the data source tab; it's simply my preference to make these changes on the worksheet.

---

# Defaults

Setting defaults is a great way to avoid changing the same configurations in a field over and repeatedly. My general rule of thumb is that if I find myself changing the same configuration in a field more than twice when I am designing a viz, I stop and set the default instead. Changing a default value for a field changes it everywhere it is already in use and the change will control every future use of that field.

The defaults that we can set for a field are based on the data type of that field. In this section we will set the defaults for a string field and a numeric field.

## Strings

### Colors

Right-click Gender➤Default Properties➤Color. You will be prompted with the edit colors dialog. From here you can pick one of the colors in the Automatic palette, or you can select colors from any other palette or combination of palettes. For this example, Ctrl+click both the f blocks under "Select Data Item" and then click the Pink block under "Select Color Palette." Next do the same for the m blocks. Note, there are two sets of blocks for f and m. We will fix this soon.

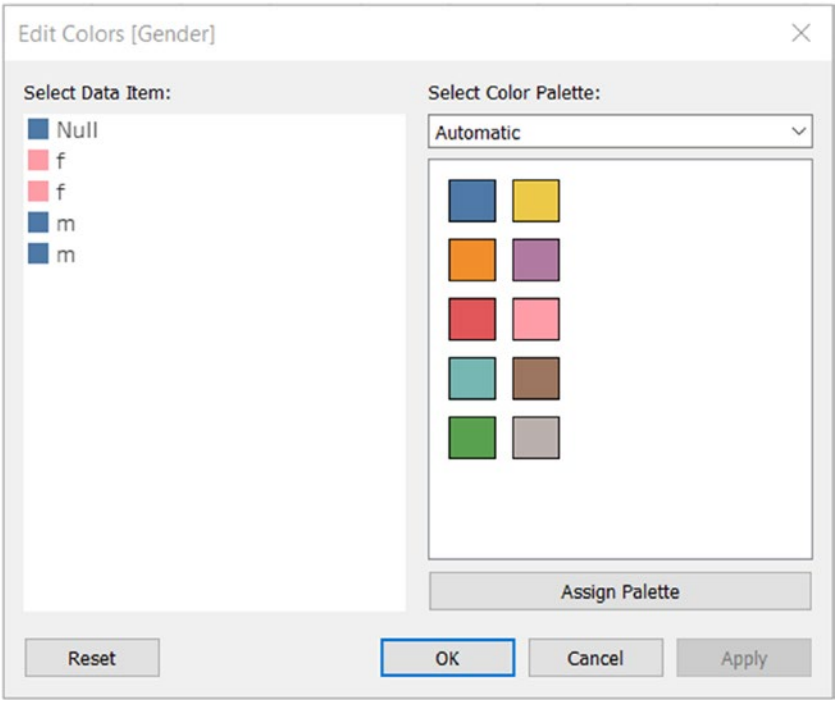



***Figure A-16.*** *Edit Colors dialog*

## Shapes

Next we will set the default for Shape. Right-click the Gender field and select Default Properties and then click Shape.

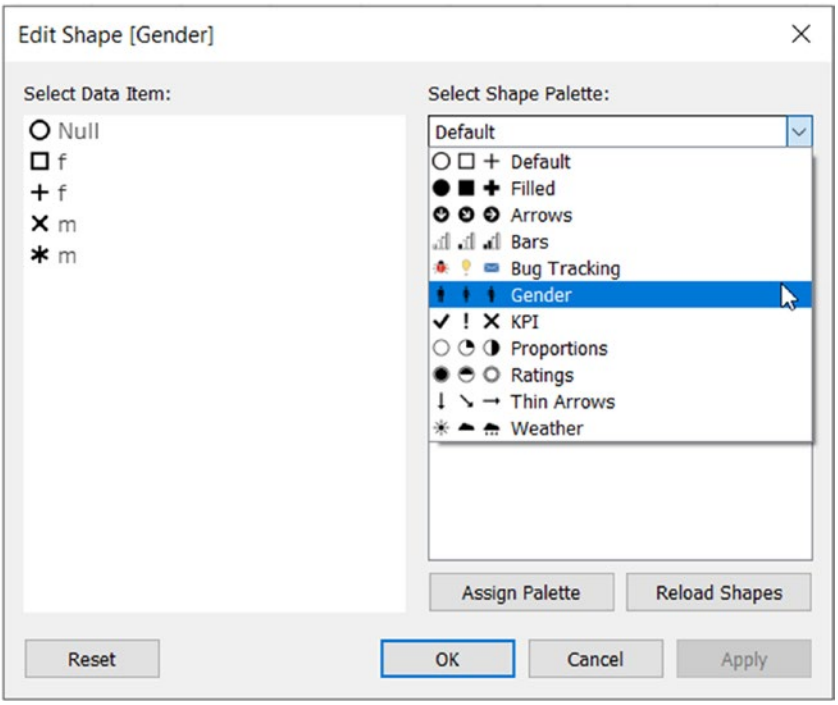



***Figure A-17.*** *Set default for Shape*

This is where you can select default shapes to use for the unique values in your field. You might not use shapes often in your viz, but in some cases you will know before you start that some fields are excellent candidates for KPI style dashboards and setting default shapes and really go a long way to ensure a consistent look and feel in your work.

## Sort

To close out this section, we will look at how to set the default sort on a field. This will be useful for situations where you might want a specific ASC or DESC sort on a field, yes, but personally I go here more often when I want to set a manual sort. For example, I might know that my data will always have five locations, and I want them sorted in a specific sequence (maybe home office first).

Right-click Type of Bear and select Default Properties➤Sort. Next change the "Sort By" to Manual.

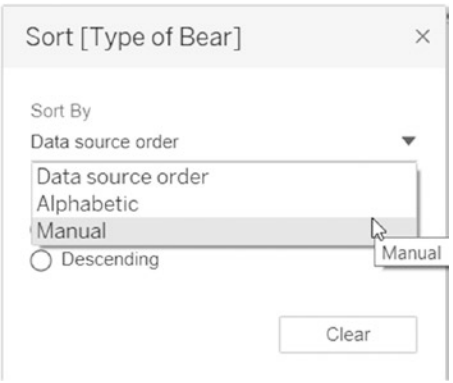



***Figure A-18.*** *Set default sort to Manual*

Now you can either drag the members of this field to whatever sequence you prefer, or you can move them using the buttons on the side of the control.

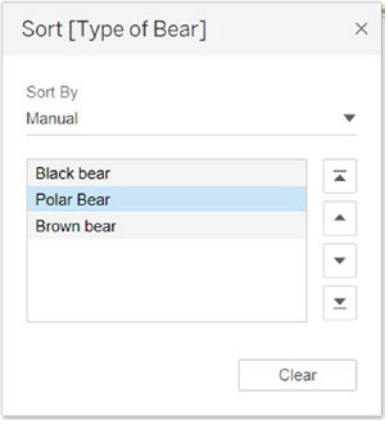



***Figure A-19.*** *Select sort sequence*

# Numbers

We will look at the two most common default values you might set in this section, Number Format and Aggregation.

## Number Format

I've lost track of how many hours I've seen junior developers waste setting and resetting the default number format for the same field in the same data connection in one viz after another. It's so easy to change that you don't even notice when you've changed it 20 times in 20 worksheets in the same workbook. Then you have the new problem of making sure you changed that setting EVERYWHERE. Did you miss one? Did you miss a half dozen? Trust me, it's easy. I've done it. If you take nothing else from this checklist, get in the habit of at least setting THIS default value for every new connection.

Right-click a Measure, select Default Properties, and then click Number Format.

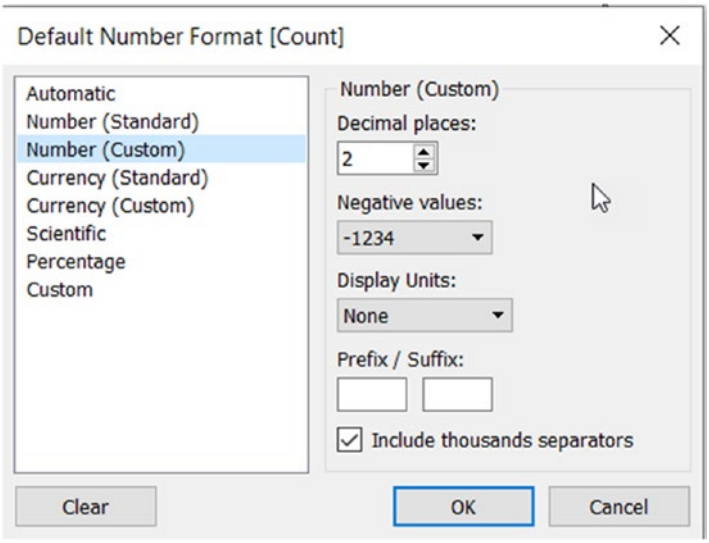



***Figure A-20.*** *Change default number format*

There is no best setting here, but my experience has been that I most commonly go to a Custom number format, remove decimal places, and in some cases (if I'm working with big numbers) change the Display Units to Thousands (K), Millions (M), or Billions (B). These changes make the numbers easier to read in tables, charts, and graphs and in general give the viz a clean look overall. You should explore all the options in this dialog (there are many), so you will be ready to quickly set defaults on all your measures in each new data source.

## Default Aggregation

For this example, we will duplicate a field, rename that field, and then set its default aggregation.

1. Right-click Age and select Duplicate
2. Right-click Age (copy) and rename it to Average Age
3. Right-click Average Age and select Default Properties➤ Aggregation➤Average

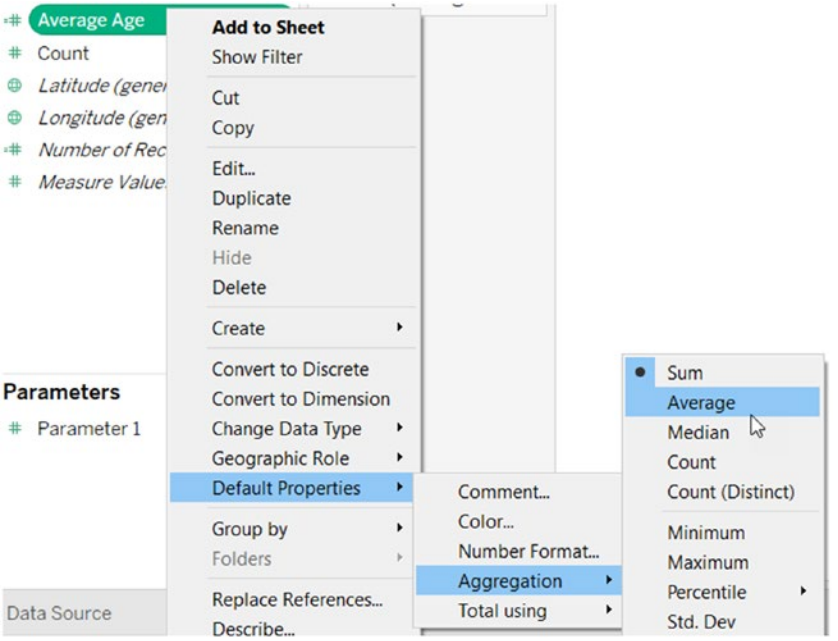



***Figure A-21.*** *Set default aggregation*

We now have a new measure we can use to show Average Age with a simple double click, without having to reset the aggregation after we add it to our viz!

## Dates

The two main date settings I want to call your attention to are the Fiscal Year start and date format. These are settings you would do well to consider for every data source as they can significantly impact the accuracy of your analytics.

## Fiscal Year Start

The Fiscal Year Start is easy to change; unfortunately you can only pick a month for your FY start.

Right-click a date field, select Default Properties➤Fiscal Year Start and then pick a month for your fiscal year start.

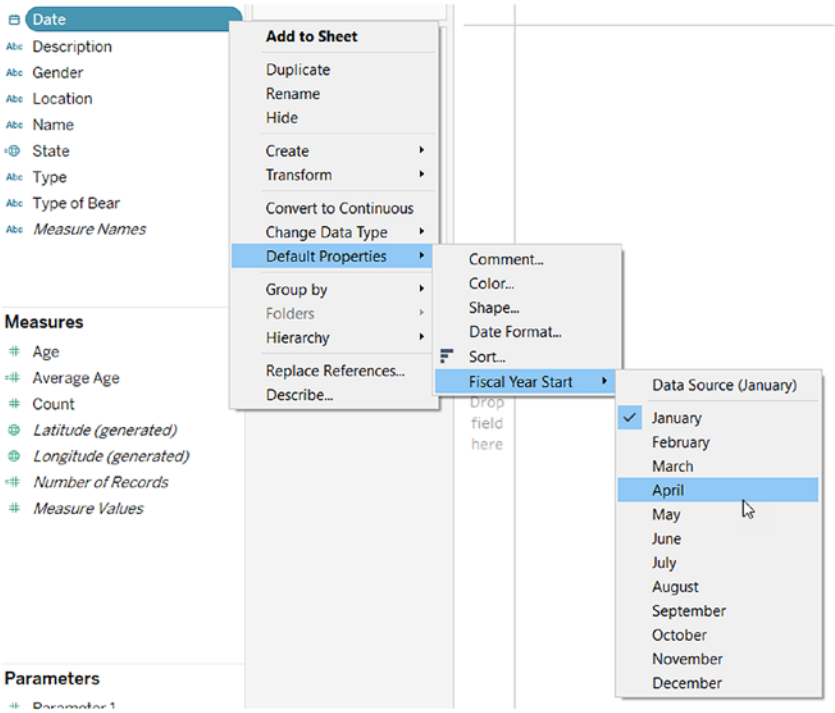



***Figure A-22.*** *Set Fiscal Year Start*

## Custom Date Format

These next two sections cover a topic that just recently came to me in a question on LinkedIn. A connection wanted to have complete control over how a date was formatted in Tableau. I recommended to him that he look into custom date formats (this section); unfortunately he misunderstood

and thought I was referring to custom dates (the next section). It took the two of us two days and a Google hangout to get to the bottom of it!

You can select a custom date format by right-clicking a date and selecting Default Properties ➤ Date Format.

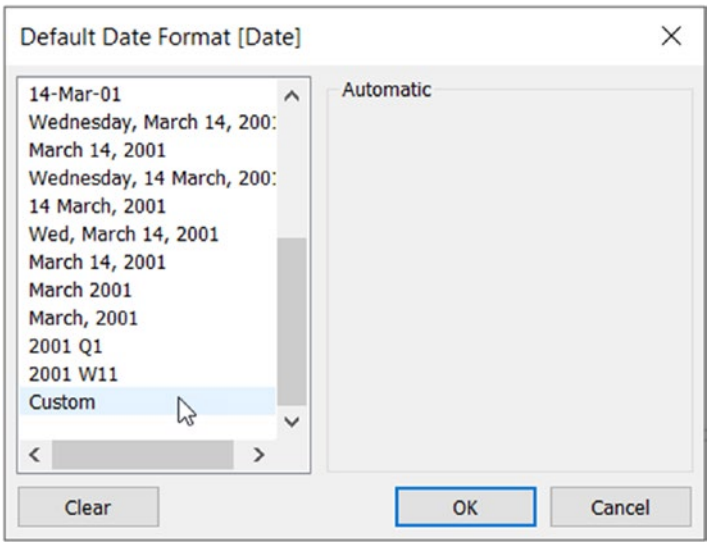



***Figure A-23.*** *Custom Date Format*

This dialog gives you over 20 options (at the time of this writing) as well as a Custom option at the bottom of the list. This Custom option lets you specify any date format you might be able to dream up. For a reference of the date format symbols available for use in a custom date format, go to the search engine of your choice, and search for Tableau Custom Date Formats. You should find a page on the Tableau Help with instructions and a table with all the supported date format symbols.

## Custom Date

Custom dates are quite a bit different from Custom Date Formats. I find Custom Dates to be a feature of Tableau that few take advantage of. This is a shame because they can be super useful!

A custom date is simply a copy of a date field set to a specific date format. I most commonly use a custom date when I want a date set to a specific format, and I want to prevent my audience from drilling into that date on my viz.

Right-click a date field and select Create➤Custom Date.

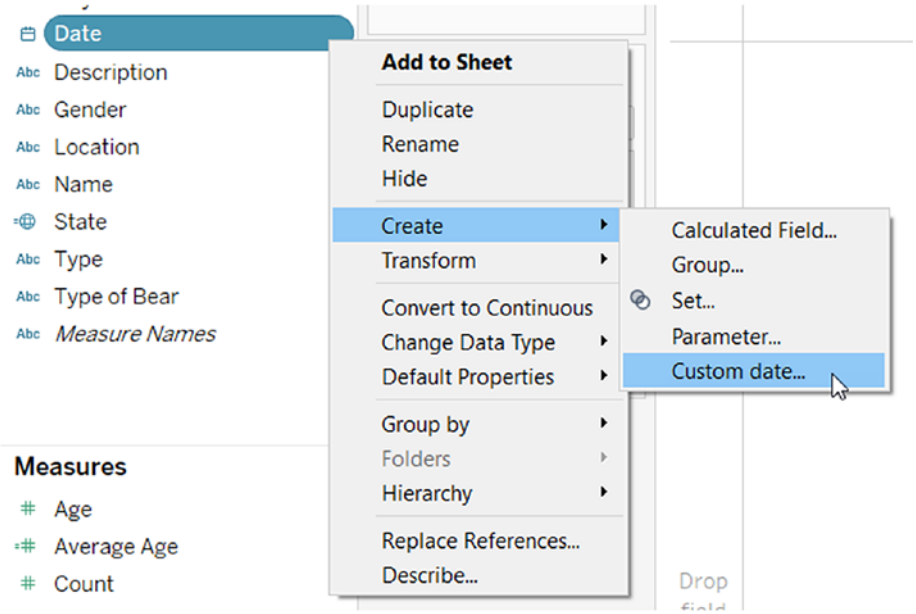



***Figure A-24.*** *Create Custom Date*

You will then be prompted to set a name and select a level of detail for your custom date. Most of the time when I create a custom date, I choose the Month/Year for the format and set the new field to work as a Date Part.

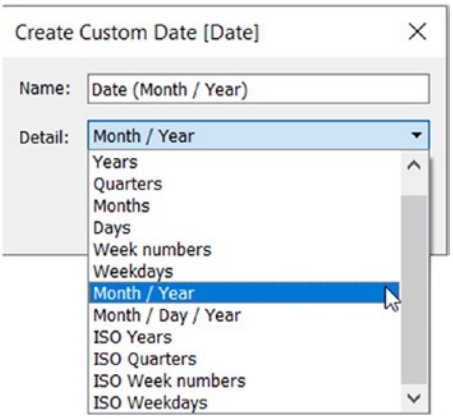



***Figure A-25.*** *Month/Year custom date*

Let's break down what we are doing here. In this example I am locking the format of the custom date I am creating down at the level of the month and year. I have chosen Date Part so this new date will act as a discrete date. Without a custom date, if I wanted to have a discrete Month/Year pill on my viz, I would need to add the date as a Year to my viz, then right-click it, and select More/Custom and select Month/Year.

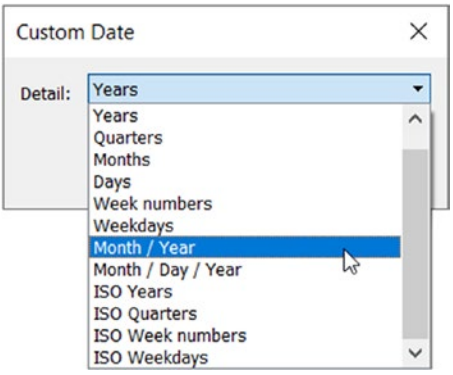



***Figure A-26.*** *Custom Date*

The observant reader will notice that Figure A-26 and Figure A-27 are nearly the same. The difference is that if I don't create a custom date as a new dimension on my data pane, I have to go through these steps every time I want to display a date as a Month/Year (my personal preference in most cases).

# Hierarchies

Hierarchies are a useful way to organize your dimensions in the data pane as well as a great option to give your audience to explore the data. In this exercise we will create a location hierarchy made up of City and State.

1. Drag the State dimension on top of the City dimension, and when you see a solid blue box around the City dimension, drop the State dimension
2. Name your new hierarchy "Location"

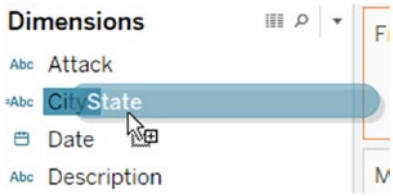



***Figure A-27.*** *Create Location hierarchy*

3. Drag State up and drop it above (not on) City. The member you are moving will move to wherever the black bar is (see Figure A-28)


***Figure A-28.*** *Change order of members in a hierarchy*

Now if you drag your new Location hierarchy on to a viz, you will see the top-level member of that hierarchy appear where you drop it. In our case, if we were to drag Location on to Rows, we would see State, and we would be able to click State to see City.

## Groups

Groups are a handy way to organize your data and to give you a higher-level roll-up than what exists in the source. For example, we might have a data source with ten cities. We could create a group for seven of those cities called "Eastern Region" and another group for the remaining three cities called "Western Region." We would now be able to aggregate our measures based on these new regions, regions that don't exist in our source data.

For this example, we will use groups in another way. We will use them to clean up our Gender field.

1. Right-click Gender
2. Select Create➤Group

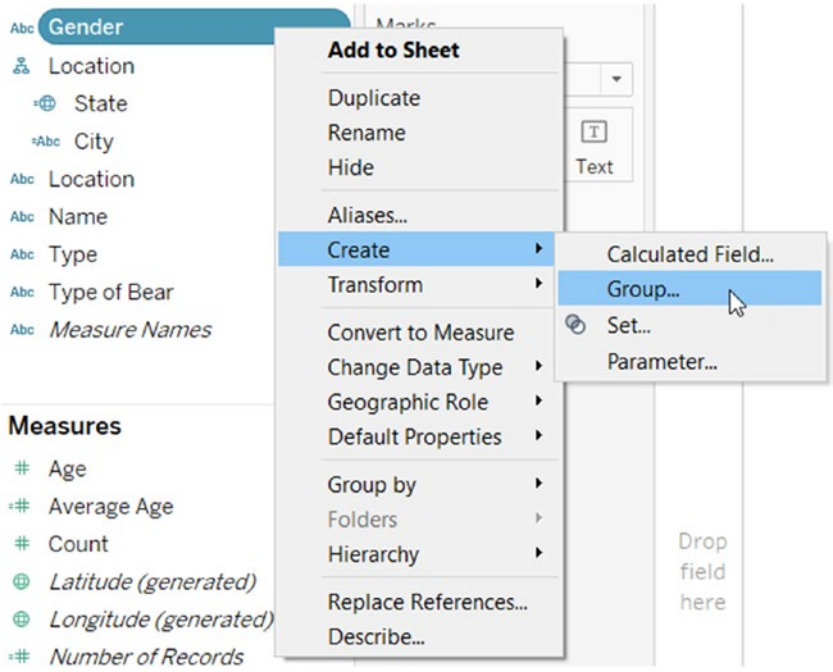



***Figure A-29.*** *Create Group*

3. Click and Ctrl+click the two entries for f
4. Click Group
5. Name your new group “Female”
6. Click and Ctrl+click the two entries for m
7. Click Group
8. Name your new group “Male”
9. Click the checkbox for “Include Other”
10. Click OK

You have now created a new field with three members (Female, Male, Other). If any new records should come into our data set, they will fall into one of these three groups. We can now set the default values for this field

just like any other field and use it in place of the original Gender field in our data source.

# Comments

Comments are one of my favorite meta-data features of Tableau, and one of the features I see used least often by just about everyone else. I LOVE comments, especially when they are well written and brief. A great comment is one that is added to a field that might be confusing either because of its name or its contents. This is a place where we can leave ourselves and those that follow us notes to remind us of all those wonderful details that are fresh on our minds when we created this data set but might be a little harder to recall in six months or six years.

To create a comment, right-click a Measure or Dimension, and select Default Properties➤Comment.

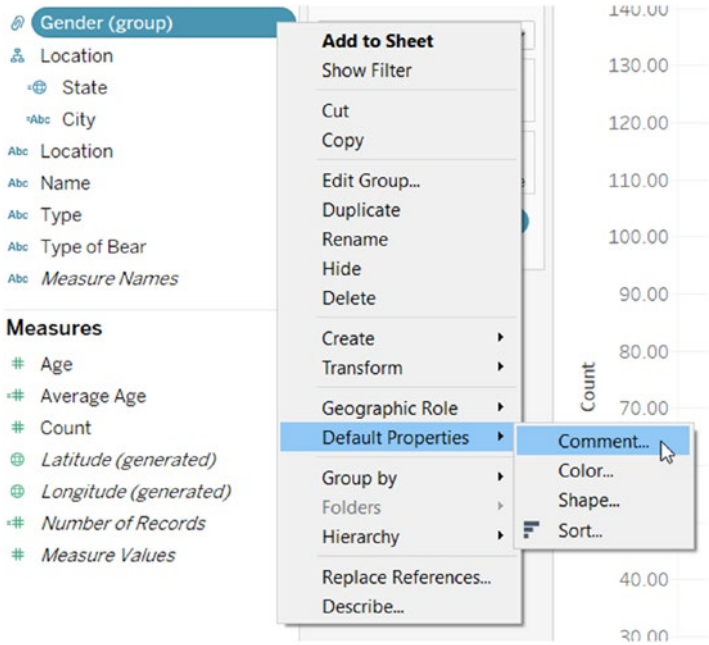



***Figure A-30.*** *Create Comment*

The comment editor is a rich text editor, meaning you can change the font, size, color, and alignment of text in your comment. I recommend keeping your comments readable but as brief as possible without sacrificing clarity.

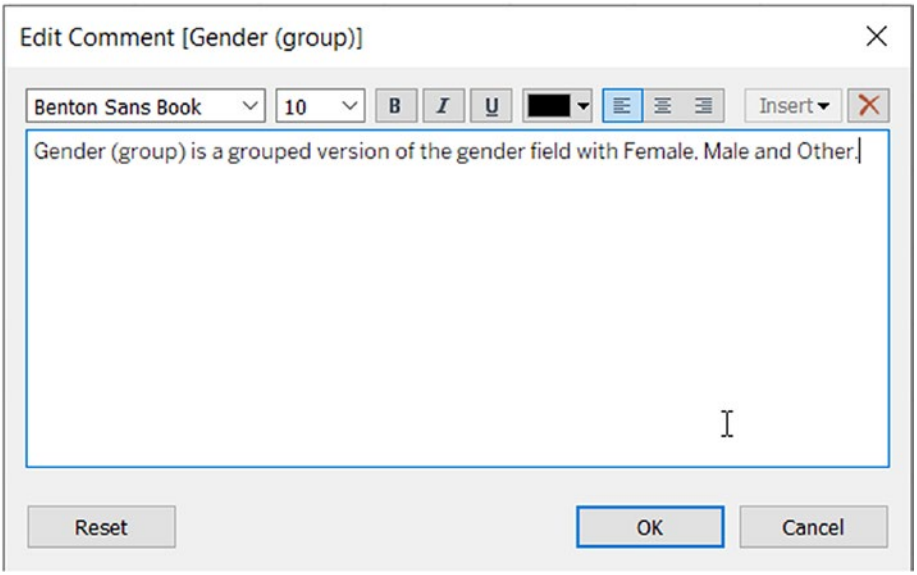



***Figure A-31.*** *Edit Comment*

Once you have created a comment, you can view it by hovering your mouse over the dimension or measure that has a comment.

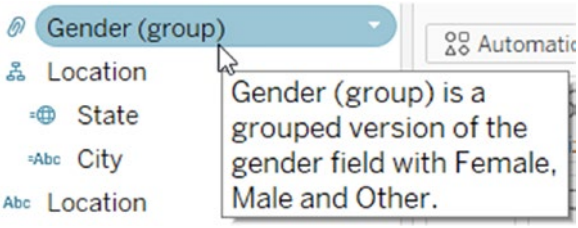



***Figure A-32.*** *Comment*

# Organize Dimensions and Measures

Another often overlooked feature of Tableau Builder is the ability to organize your data pane either by Folder or by Data Source Table. In this case, all our data is from one table, but in many cases your connection will be more complex, and you will see your measures and dimensions organized by the table from which they are drawn. If you prefer to have complete control over where your measures and dimensions are found in the data pane, you can choose to organize by folder.

Right-click an empty part of the data pane and select "Group by Folder."

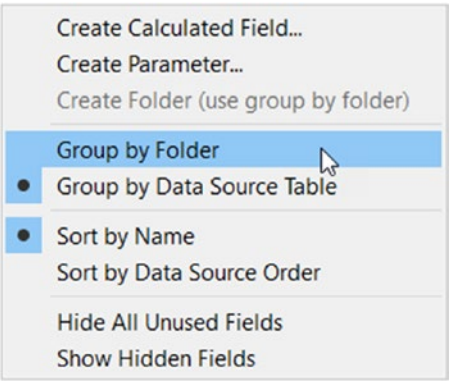



***Figure A-33.*** *Group by Folder*

Next, right-click the same area (or any empty spot in the data pane) and select Create Folder. If you click a blank area in the Measures shelf, you will create a folder on that shelf. If you click a blank area in the Dimensions shelf, you will create a folder for your dimensions.

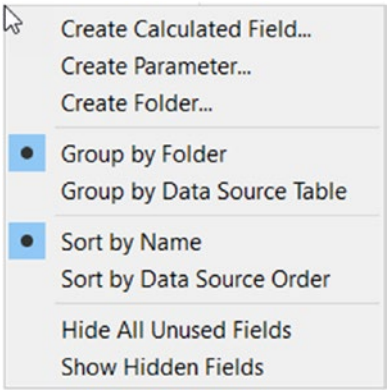



***Figure A-34.*** *Create Folder*

You will be prompted to name your new folder. You can create as many folders as you want. Once you have your folders, you simply drag any measure you want into a folder on the Measures shelf or any dimension you want into a folder on the Dimensions shelf.

## Calculated Fields

If you create a calculated field, that field will be saved as part of your data source. In the example we have been working through, we used the Split function to generate City and state fields. As noted previously, these fields are in fact calculated fields that were generated for us. These calculated fields will be saved with the connection and available to anyone that bases their work on the Saved Data Source or Extract that we create in the next step. Saving calculated fields with your connection is another great way to ensure consistency across your community of Tableau developers.

# Live or Extract

The final section in this chapter is not so much a thing we do to prep our data for Tableau as it is a choice we make as to how we want to save the work we've done up to this point.

The entire point of this chapter is to do the vital steps of cleaning and organizing our data sources in the beginning and then benefiting from this work later. To get that benefit, we must save our work.

We have three choices for saving a fully prepped data source in Tableau. We can publish the connection to Tableau Server, we can create a local saved data source for our use or to share with others in a local network, or we can create an extract (Tableau Data Extract or Hyper) that will contain all the changes we have made as we prepped this data.

## Tableau Data Server

If you have access to Tableau Data Server, saving your connection to the server is hands down your best option. This will create a live connection to your data stored in Tableau Server. Any other developer with access to your Tableau Server (and permissions to your saved connections) will be able to connect to your saved connection and begin working immediately. Any changes pushed up to your saved connection in Tableau Server will cascade out to any work based on that connection the next time those dashboards are opened.

To save a connection to Tableau Server, click Server➤Publish Data Source, and select the data source you wish to publish.

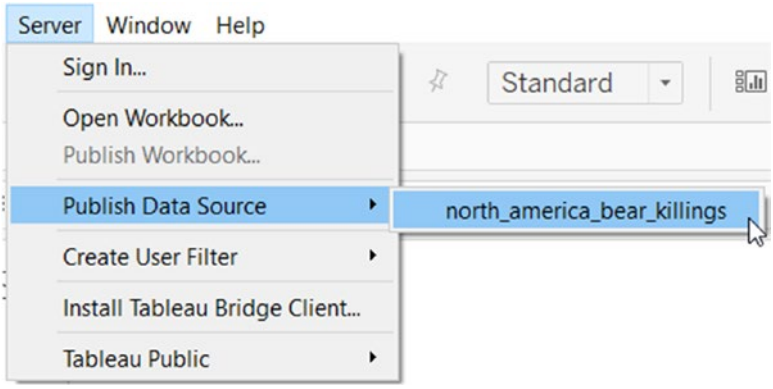



***Figure A-35.*** *Save to Tableau Server*

## Saved Data Sources

Saving a connection to a Saved Data Source can be a viable option for saving and sharing a live connection if you don't have access to Tableau Server. It works basically the same way as saving to Tableau Server, but you will save to a local file resource (your hard drive or a shared drive on your network).

To save a connection to a Saved Data Source, right-click that connection and select "Add to Saved Data Sources."

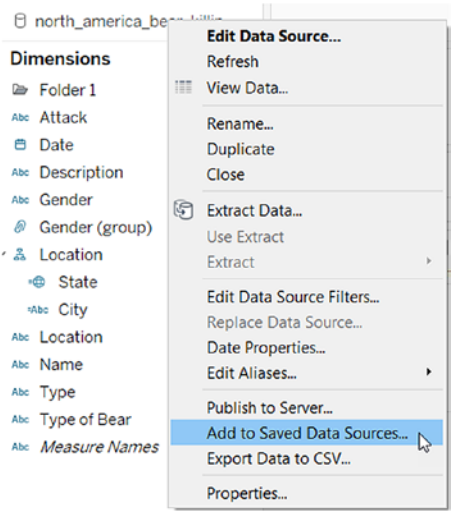



***Figure A-36.*** *Add to Saved Data Sources*

You will then be prompted for a location to which to save your file.

## Save to Extract

Another option is to save your work to a Tableau Extract (either a Tableau Data Extract or Hyper). In this case you will extract your data and save the workbook as a packaged workbook. To share this data source, you should share the workbook.

Extracts are a great way to make your Tableau dashboards faster. In closing out this chapter, I want to share three ways to make extracts faster.

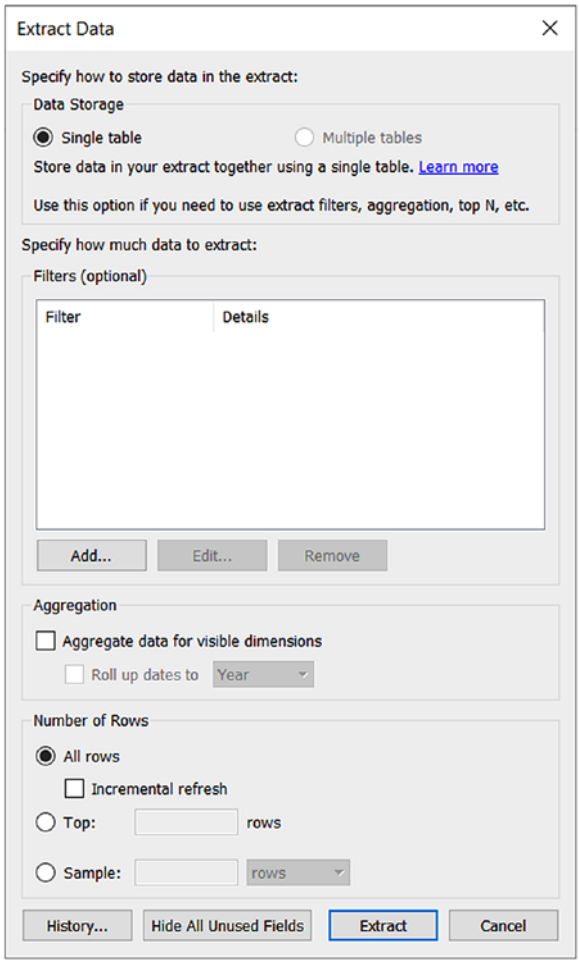



***Figure A-37.*** *Configure extracts*

In Figure A-37 you will note that a large part of configuring a new extract is setting up filters for that extract. Filtering an extract has the same pros and cons as filtering a connection to any source data. On the pro side, you will have faster dashboards if your extract is limited to only the data required for those dashboards. The major con is that any data you filter out of your extract won't be available in your analytics.

Another option for making faster extracts is to select the option to "Aggregate data for visible dimensions." If you select this option, Tableau will look at all the views in your workbook and will aggregate the data in your extract as much as possible based on how you use that data in your work. Let me explain. In a very simple example, you might have a connection to 1 million rows of data from a manufacturing process with multiple records saved per second. In this example you have created one view that aggregates this data to show results hourly. If you were to create an extract from this connection in the workbook described and you check "Aggregate data for visible dimensions," the extract will aggregate the data hourly rather than at the original level of detail (multiple records per second). This will result in a significantly smaller, faster extract. I recommend you leave this option unchecked until you have completed your analytics and then rebuild your extract prior to publishing and enable this option at that time.

A final performance tweak for extracts could be to check the box marked "Hide unused fields." Like the "Aggregate data for visible dimensions" option, I recommend you leave this unchecked until you have completed your views/dashboards and then just prior to sharing your work rebuild your extract with this option enabled. What would happen in this case is that Tableau will hide all unused fields prior to creating the extract. Unused fields are NOT included in the extract, so the width (number of rows) in your extract could potentially be reduced. This option can have a significant impact on extract performance in some cases (cases where a large number of fields are unused or even sometimes when a small number of large fields are unused).

---

**Warning!** If you save a Tableau Extract and then open that extract with a workbook other than the one in which it was created, **YOU WILL NOT BE ABLE TO REFRESH YOUR EXTRACT.**

---

## Summary

This chapter has covered a LOT of ground. In closing I just want to remind you that the work you do here, and yes, it can be a LOT of work …, will pay off. It will absolutely be worth your time and effort to go through these steps and prep your data before beginning analytics. You will know your data better; you will be more aware of its shape and nature, and you will have to do less modification on the fly as you create your viz.

Whether you use Tableau Prep or Tableau Desktop, you now have the tools you need to be successful. Now, as my mentor Dan Murray would say, it's time to dance with the data.

# Index

## A, B



## C



## D



## E




T. Costello and L. Blackshear, *Prepare Your Data for Tableau*,
https://doi.org/10.1007/978-1-4842-5497-4

## F



## G, H



## I



## J, K



## L



## M

## N



## O



## P, Q



## R



## S



## T

## U, V



## W, X, Y, Z