DATA + JOURNALISM

A Story-Driven Approach to Learning Data Reporting

Mike Reilley and Samantha Sunne

Data + Journalism

Taking a hands-on and holistic approach to data, *Data + Journalism* provides a complete guide to reporting data-driven stories.

This book offers insights into data journalism from a global perspective, including datasets and interviews with data journalists from countries around the world. Emphasized by examples drawn from frequently updated sets of open data posted by authoritative sources like the FBI, Eurostat and the US Census Bureau, the authors take a deep dive into data journalism's "heavy lifting" – searching for, scraping and cleaning data. Combined with exercises, video training supplements and lists of tools and resources at the end of each chapter, readers will learn not just how to crunch numbers but also how to put a human face to data, resulting in compelling, story-driven news stories based on solid analysis.

Written by two experienced journalists and data journalism teachers, *Data + Journalism* is essential reading for students, instructors and early career professionals seeking a comprehensive introduction to data journalism skills.

Mike Reilley teaches data and digital journalism at the University of Illinois at Chicago. He founded the digital resources site JournalistsToolbox.org in 1996 and continues to update it today for the Society of Professional Journalists.

Samantha Sunne teaches data and digital techniques to journalists at universities and other groups including the Society of Professional Journalists, Hunter College-CUNY and Investigative Reporters and Editors. She is a long-time journalism freelancer specializing in data and investigative stories.

Data + Journalism

A Story-Driven Approach to Learning Data Reporting

Mike Reilley and Samantha Sunne

Routledge
Taylor & Francis Group
NEW YORK AND LONDON

Designed cover image: © Illustration by Billy O'Keefe

First published 2023
by Routledge
605 Third Avenue, New York, NY 10158

and by Routledge
4 Park Square, Milton Park, Abingdon, Oxon, OX14 4RN

Routledge is an imprint of the Taylor & Francis Group, an informa business

Library of Congress Cataloging-in-Publication Data
Names: Reilley, Mike, 1965– author. | Sunne, Samantha, 1991– author.
Title: Data + journalism: a story-driven approach to learning data reporting / Mike Reilley and Samantha Sunne.
Other titles: Data [plus] journalism
Description: New York: Routledge, 2022. |
Includes bibliographical references and index.
Identifiers: LCCN 2022028643 | ISBN 9781032226125 (hardback) | ISBN 9781032225913 (paperback) | ISBN 9781003273301 (ebook)
Subjects: LCSH: Journalism—Data processing. | Data mining.
Classification: LCC PN4784.E5 R45 2023 |
DDC 070.4/0285—dc23/eng/20220924
LC record available at https://lccn.loc.gov/2022028643

ISBN: 9781032226125 (hbk)
ISBN: 9781032225913 (pbk)
ISBN: 9781003273301 (ebk)

DOI: 10.4324/9781003273301

Typeset in Goudy
by codeMantra

Access the companion website: http://dataplusjournalism.com

To my parents, who taught me never to give up. And to my wife, Isabella, who helped me beat cancer.

—Mike

To my mom, who always said I could be a writer.

—Samantha

Contents

Tables

Figures

Acknowledgments

We would like to thank many people who helped us with the writing and publishing of this book. Being a first-time author can be challenging, but a great support team and cooperation from many professional journalists made it much easier. We're grateful to our publisher, Routledge, and the team of Lizzie Cox, Hannah McKeating and Priscille Biehlmann, who helped us navigate the process. Their patience and sense of humor were greatly appreciated.

We also want to thank David Cuillier of the University of Arizona, for lending us his expertise on freedom of information and public records around the world. Lise Olsen, an investigative journalist with the *Texas Observer* and author of the book *Code of Silence*, shared nearly three decades of experience writing data-driven stories. Heather Cherone, the Chicago politics reporter at WTTW, shared how to work data reporting into daily online and TV news coverage.

Alvin Chang, head of visuals and data for *The Guardian US*, gave us some keen insights into designing graphics, and Lynn Walsh of Trusting News kept us on the cutting edge of building trust with readers through data stories. John Walton, data journalism editor at BBC News' data journalism team, and Andy Boyle, director of product engineering at the *Chicago Sun-Times*, shared their innovative graphics and databases for international and local audiences.

Mike would like to thank his department chair at the University of Illinois–Chicago, Zizi Papacharissi, for her encouragement and longtime support not only for this textbook but also for moving data journalism to the core of University of Illinois–Chicago's journalism offerings.

Samantha would like to thank the community at Investigative Reporters and Editors, who elevate journalism in every field, and give a salute to the data journalists who made this whole universe possible, including Steve Doig, Sarah Cohen, Brant Houston and Cheryl Phillips, among others. Thank you to Carlie Procell, Lukas Udstuen and Lucia Walinchus for their technical expertise; Jian Chung Lee for individual projects; and Marissa DeCuir of Books Forward for advice.

Introduction

The Power of Data Storytelling

In 1892, nearly 75 years before data journalism found its way into newsrooms, Ida B. Wells returned to Memphis from a promotional tour for her newspaper, the *Memphis Free Speech*. She found the city in peril – a white mob had lynched three Black men after a conflict between a white man and a Black man had escalated for a few days.

Wells began to research why those and many other lynchings happened – by using what we know today as data reporting techniques.

She went to the places where the lynchings happened, compiling data from newspapers and firsthand interviews. She counted not just the lynchings but also how many had occurred due to an accusation of a Black man attacking or harassing a white woman. This provided the foundation for a series of articles and editorials that she published in the *Free Speech*. In her autobiography, *Crusade for Justice*, Wells later wrote, "They had committed no crime against white women. This is what opened my eyes to what lynching really was."

After publishing a series of articles and editorials using the data, Wells was issued a warning from those she exposed: If she returned, she would be lynched. After a mob in Memphis trashed her newspaper office, friends in New York implored her not to go back – and she didn't (Figure 0.1).

While Wells didn't have RStudio, SQLite or glitzy mapping software to tell her story, she understood one of the fundamentals of data journalism – in order to show the scale of a problem, just start counting.

Her story, portrayed in an audio mini-documentary on the Center for Documentary Studies' Scene on Radio, clearly shows the impact of what we know today as data journalism. It tells complicated stories more clearly than relying on words, quotes and anecdotes alone. Data lends credibility to stories and unearths disturbing trends and corruption. Today, it helps the reader understand a story on a deeper level through analyses, maps, charts and databases.

Wells' boots-on-the-ground reporting, combined with her data analysis, lifted her story to a different level. That's the goal of this textbook – to elevate your reporting by taking a different approach. Data reporting techniques have existed for more than a century as journalists have collected and counted information for stories. But with the emergence of technology in the latter half of the 20th century, computer-assisted reporting and what we know today as data journalism became embedded in newsrooms large and small.

DOI: 10.4324/9781003273301-1

Figure 0.1 Ida B. Wells.

"Data journalism has been important – crucial – for more than 30 years now," said Lise Olsen, an investigative reporter and editor at the *Texas Observer* whose work has appeared in Inside Climate, NBC, *Houston Chronicle*, and in documentaries on A&E and CNN. "Without knowing how to analyze data, a journalist must go nearly blind into the world of reporting.

"So much data is digital, that if a reporter doesn't know how to interpret numbers, statistics or recognize basic formats, he or she will miss crucial opportunities or be far too easily fooled by spin masters and fake news."

What Is Data Journalism?

> *What makes it data journalism isn't the form, it's the starting point in a data source we corralled, cleaned and interpreted.*
>
> – Melissa Bell, Vox Media publisher

Those words, written by Bell in 2015, best summarize *Data + Journalism*'s approach. Data journalism is the use of data and statistical analysis to uncover, better explain and/or provide context to a news story. Some data stories take

months to report and publish, while others can be turned around in a day or two. Either way, reporters, editors and designers follow a process to tell those stories.

Specialized data journalism skills focus on acquiring, cleaning, analyzing and visualizing data, a process we follow in this book (Figure 0.2):

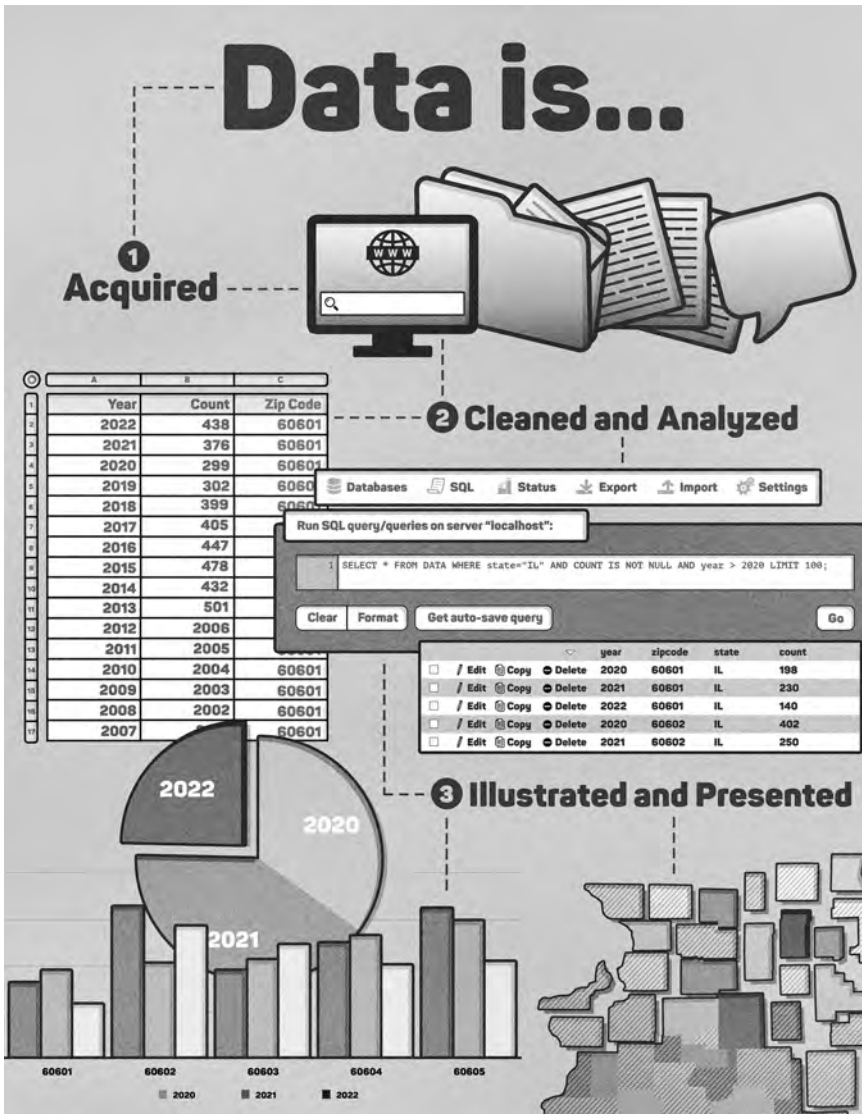


Figure 0.2 The data reporting process (Illustration/Billy O'Keefe).

- **Acquiring data** includes knowing what public records exist and how to obtain them, as well as seeking numbers suitable for spreadsheets or databases.
- **Cleaning data** involves software and human intelligence to standardize spelling, punctuation and formatting to obtain accurate counts.
- **Analyzing data** may be accomplished with spreadsheets, programming languages, database management systems or visual representations to find commonalities or outliers.
- **Presenting data** involves building maps, interactive charts or graphics to enable audiences to comprehend the analysis or personalize the data through interactives.

This book follows that reporting process, offering the skills and exercises journalists need to practice the craft. It is much like a Swiss Army knife – it exposes you to many aspects of data journalism, ranging from coding to point-and-click tools. You'll work with RStudio and SQL to analyze data. You'll scrape data from websites, social media and PDFs using code and browser-based tools. You learn how to visualize data with tools such as Datawrapper and Flourish.

For decades, journalism has been taught as a trade, and this book helps readers learn by working with real-world data – datasets that are updated regularly by government agencies and other authoritative sources.

This book serves as the latest chapter in the ever-evolving data journalism landscape. Our approach pays homage to the fundamentals of the past with a look to technologies of the present and the future. Moreover, it covers not just how to crunch numbers but also how to put a human "face" to data, resulting in compelling news stories based on solid analysis. It also explores how to use and build databases, create visualizations with datasets and build trust with readers through transparency, accuracy and sound research.

How important is a human element in a data story? Alvin Chang, head of data and visuals at *The Guardian US*, said each source gives him a slightly sharper view of the issue he's writing about. But he often wonders if his biases led him to talk to a self-selecting group of people. "What if I just needed to talk to more people? What if I'm missing the larger story?

"It forces us to test the theses of our stories using the scientific method, rather than just through our individual experiences," Chang said. "Sometimes it reveals human experiences that we weren't previously aware of. And ultimately, it helps us be more curious and empathetic journalists."

On one level, data journalism is shorthand for database- or data-driven journalism, where journalists find stories and create conclusions by analyzing large datasets. It is also closely associated with investigative journalism and overlaps with data visualization, as it requires close collaboration between journalists and digital and design specialists to find ways to present data through interactives, written stories, databases, videos, maps and charts.

Data-driven journalism applies to nearly any beat or topic that journalists cover, said Andy Boyle, former director of Product Engineering at the *Chicago Sun-Times*.

"If you have a data-oriented frame of mind, it will help you in everything you do, not just in your journalism career, but in your life," he said. "Everything is data, and the quicker you understand that, the faster you'll be able to find stories.

"It's just another skill in your toolbox, but an important one that lifts your work to greater heights, gives it more of a scientific or academic rigor, and makes your stories potentially be even more powerful."

Heather Cherone, the politics reporter at WTTW News, the PBS station in Chicago, uses data reporting on a regular basis on her coverage of city hall and the state legislature. She says data skills are "crucial for all journalists."

"The world we live in is awash in data," she said. "Our job is to help make sense of that raw information by translating it into clear knowledge that informs the lives of our readers and viewers.

"To do that, journalists have to know how to find that data, which often takes old-fashioned reporting. It is also important to analyze that data without fear or favor and without a desired conclusion."

Ethics, Trust and Data

Lynn Walsh is the co-founder and assistant director of Trusting News, which teaches journalists how to be more open and build trust with readers. Part of her job is to help data journalists learn how to be more transparent in their reporting and explain the process we cover in this book. In Chapter 11, we explore some of those transparency techniques with Walsh and other journalists.

Walsh, a former investigative TV news producer, said journalists often don't explain that data (or good data) on a specific topic or issue doesn't exist. This is something readers and viewers typically want to know but are rarely told.

For example, during the first few months of the COVID-19 pandemic, journalists were criticized and accused of hiding information related to how many people were recovering from the virus. Readers and viewers were saying journalists were hiding this information because they wanted to focus on the number of infections and deaths instead of the more positive stories of recovery.

"As journalists, we are limited to what we can report on and provide data about," Walsh said.

"If the data doesn't exist, it doesn't mean we are hiding information, it means we don't have it to share. Yes, in some cases, we can create our own databases and in some cases we do, but that is not always possible or feasible and that can take a lot of resources."

Technical skills can be acquired – not without effort – but with time people can learn the technical side of working with data. But far less time is spent teaching and developing the skills needed to find story ideas – and good story ideas are the true currency of all forms of journalism.

In reality, the federal, state and local health departments were not tracking the information, or doing a poor job of it. Explaining these limitations around data

can be helpful to users. Explaining how we get data (freedom of information or public records) and the limits of the data – what it does not include because it was not collected by the agency – is also helpful, Walsh said.

"We should realize a lot of people do not know what FOI is or that the general public has a right to access certain information," she said. "Adding media literacy elements explaining how public records work can also be helpful to the audience.

"If we do not explain these elements of our reporting, people will make assumptions that we are purposefully trying to hide data, push a certain agenda, etc. Also, if you do create your own database and collect your own data, explain why you had to do this, as the data didn't exist, why you thought it was important to invest the time in doing this and how you did it."

Impact of Data Journalism

Data journalism walks hand in hand with investigative reporting, which is why so many data reporting skills are taught at annual conferences such as Investigative Reporters and Editors (IRE) and the National Institute for Computer-Assisted Reporting (NICAR). Data and public records lend credibility to investigative stories and reaffirm one of journalism's basic premises: Show, don't tell.

Just how valuable are these investigative stories? In his book, *Democracy's Detectives*, author James T. Hamilton did a cost-and-benefit analysis of several investigative stories. He showed that for every $1 a media outlet spent on an investigative story, the net policy of benefits to the public was as much as $287.

Good data-driven stories do more than win awards. They effect change at many levels of government and society as a whole.

History of Data Journalism

While Ida B. Wells didn't have a laptop or smartphone to do her work, she took to heart the concept of looking for patterns and then writing about it. But it wasn't until more than a half century later that newsrooms began to adopt data journalism.

In his 2021 article, "The History of Data Journalism" for *The Data Journalism Handbook* website, University of Illinois professor Brant Houston outlined the evolution of what we know today as data journalism and computer-assisted reporting. "Many practitioners date the beginning of computer-assisted reporting and data journalism to 1952 when the CBS network in the United States tried to use experts with a mainframe computer to predict the outcome of the presidential election," Houston wrote. "That's a bit of a stretch, or perhaps it was a false beginning because they never used the data for the story."

Houston traced the true origins of data journalism – and one of its key founders – to the Detroit riots in the summer of 1967. Dozens were killed and more than 1,000 people were injured in the violence, but some Detroit newspaper reporters dug deeper into the cause of the riots in the wake of what occurred.

Among those reporters was Philip Meyer, who used a social science approach to gathering data and telling the story of the riots in a new way. In doing so, Meyer pioneered what we know today as data journalism, eventually writing the book *Precision Journalism*, which has been revised several times as *New Precision Journalism*. The book is a mainstay in data journalism and social science college courses.

The evolution of desktop computers, software and the Internet opened new opportunities in the 1980s and 1990s. Spreadsheets helped reporters analyze city budgets and build large databases that could be maintained over years. New digital tools and phone apps emerged, making it easier to gather data. Drones, sensors and other technologies offered new opportunities. In the last decade, tools such as RStudio and code such as Python gave data journalists the chance to ask questions of data like never before.

Free data visualization tools like Flourish, Datawrapper, Infogr.am and many others help reporters build interactive charts and graphics with nothing more than a spreadsheet. Journalists have more ways to tell stories than ever before. Arc-GIS, Carto.com, MapBox, Open Street Map, Google Earth and MyMaps give readers a birds-eye view of an issue.

Data journalism skills are essential in today's newsrooms, according to academic research. Surveys show that many professional journalists want to improve their abilities to work with data. The demand for data journalism instruction exists both in the journalism industry and others, and in the United States and abroad.

John Walton, data journalism editor at BBC News' data journalism team, said journalists need to be able to interpret data, have a solid understanding of statistics and crucially they need to be able to communicate what they have found in the data to readers who may never peek within the tiny cells of a spreadsheet. News organizations unable to deploy data journalists can't really hope to report on the modern world in all its rich complexity. It's as simple as that.

But Walton also cautioned that data journalists shouldn't focus too much on the technical side of the craft and that journalism itself should always come first. What really counts is shoe-leather reporting, the curiosity to seek out stories, the imagination to see the opportunities for storytelling using data and the doggedness to stick with a complicated investigation, no matter where it leads.

"A journalist can be equipped with all the technical skills in the world," Walton said, "but if they don't have ideas for stories they will always be beaten to the punch by someone with a good idea for a story but only a fairly basic grasp of how to sort a spreadsheet."

In this book, you will learn not just how to crunch numbers but also how to humanize data, resulting in compelling news stories based on solid analysis.

It features training videos, tip sheets and links to resources for finding, cleaning and analyzing data. The DataPlusJournalism.com website includes additional

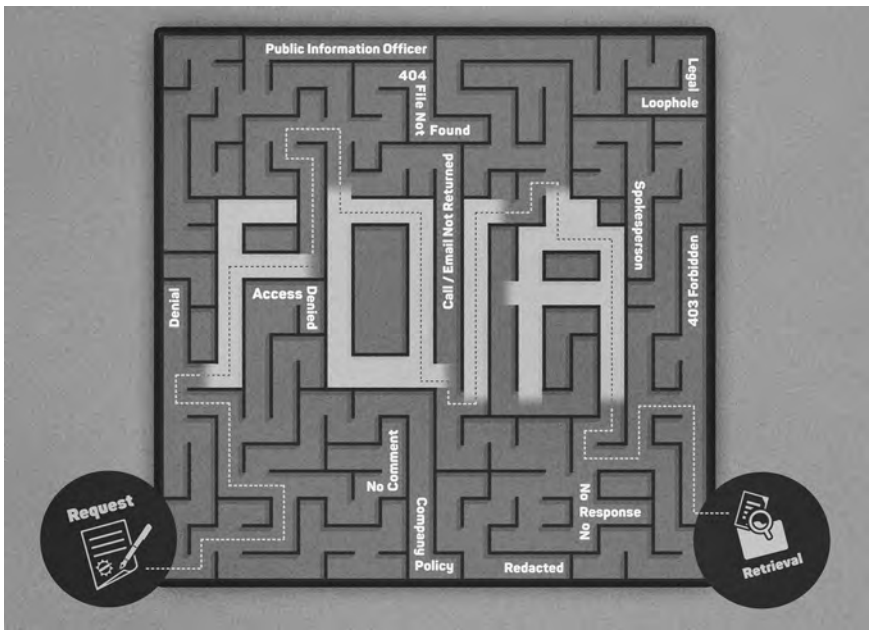
exercises, diversity in data, changes in data journalism and software and many tips, tricks and resources. Chapter and exercise updates can also be found on the blog.

Footnotes

Scene on Radio, More Truth https://www.sceneonradio.org/s4-e11-more-truth/
Vox, What Is Data Journalism? https://www.vox.com/2015/2/4/7975535/what-is-data-journalism
Sage, Journalism & Mass Communication Educator https://bit.ly/sagejmce
American Press Institute, How to Teach Data Journalism in Journalism Schools https://bit.ly/dataapiarticle
Nieman Lab, Journalists Know They Need to Get Better with Data and Statistics, But They Have a Long Way to Go https://bit.ly/niemanlabdata
The Data Journalism Handbook, The History of Data Journalism http://bit.ly/datajhistory
Ida B. Wells Papers, University of Chicago Library https://bit.ly/idabwellsdata

1 Acquiring Data

Mike Reilley



Pearl Zhu, author of the "Digital Master" book series, wrote that we are "moving slowly into an era where big data is the starting point, not the end." This is true, and it begs a question: Where do we really *start* with acquiring the data we need for a news story?

There are many avenues to pursue: federal, regional, and local data portals; contacting a government agency; doing original research to build a dataset and doing online searches. And once we locate the data, how can we capture it in a format that helps us tell the story?

DOI: 10.4324/9781003273301-2

The RTI Rating reports that freedom of information (FOI) laws, on a global scale, are relatively new. As of 2022, 134 countries have laws on the books, more than 100 of which only adopted laws in the past 22 years.

Sweden has the oldest FOI laws, dating back to a 1766 law that gives Swedes the right to gain government documents. Some dispute this, saying China was ahead of the curve several centuries earlier, but Sweden is widely considered the first.

Other countries lagged behind until the mid-20th century, with the US, France, Japan, Israel, Colombia and 22 other countries all adopting FOI laws during the last century. Russia, Switzerland, Germany, Mexico and Argentina have followed suit, but the efficacy of those laws depends on the courts and governing bodies in those countries.

Using FOI Laws

University of Arizona professor David Cuillier has taught public records reporting for decades, has studied public records denials and has testified before Congress on public records laws. He has trained thousands of journalists on how to obtain records and use them as concrete data in their reporting.

"Public records provide credible, substantive 'proof' for story angles that sources have a difficult time refuting," he said. "They avoid being spun by officials."

For example, when the *Miami Herald* in 2017 heard that juvenile detention there was a rough place, they knew they wouldn't get much confirmation from officials. But in obtaining surveillance video through public records laws, they showed the public actual brawls among detainees, often egged on by guards. The story, "Fight Club," was a 2018 Pulitzer Prize finalist.

Likewise, when a broadcaster heard the Boise, Idaho, mayor was using the city credit card for buying personal items, the city denied it. But after reviewing that surveillance video from a public records request, the KBCI-TV station showed the mayor and his cronies hauling loot into his office, caught red-handed.

"The mayor was fired and imprisoned for that and other shenanigans," Cuillier said. "That is the power of public records."

This chapter focuses on obtaining datasets through public records searches, data portals, FOI requests and how to scrape data from a web page or a PDF using a variety of tools and techniques.

* * *

Making a Public Records Request

While the FOI's impact varies by country, all FOI laws tend to have requirements for filing, and agencies are given a specific amount of time to respond. No matter where they are working, it is the reporter's responsibility to look up the particular laws in their country, state, province, etc.

When you cannot find datasets through searches or a data portal, you must turn to government agencies to find them. While datasets are increasingly available

online, many are still kept hidden deep in government offices at the federal, state and local levels, for a variety of reasons. So reporters must contact public information officers (PIOs) to request the documents.

Typically, this can be handled through a routine phone call, email or visit to the agency. But if those avenues aren't fruitful, reporters must use a Freedom of Information law to request the public document.

According to the National Archives, FOI (5 U.S.C. 552, as amended) provides any person with the statutory right to request information from executive branch agencies of the US government. This right of access is subject to nine statutory FOI exemptions, which provide agencies the authority to withhold records in whole or in part. FOIA requesters may appeal any such withholding, or other adverse decision, back to the agency and may also file a lawsuit to seek redress in federal court. Before going to court, requesters are encouraged to contact the agency's FOIA Public Liaison at any time for assistance and to utilize mediation services offered by the Office of Government Information Services (OGIS).

Journalists aren't the only ones who file FOI requests. Lawyers do, and private and public businesses, including banks, file them. So do everyday citizens who are curious about what is going on in their communities. But for journalists, FOI and other "sunshine" laws, which can vary by state and country, can greatly enhance the depth and scope of reporting. Backing up a story with public documents lends a high level of credibility to reporting.

* * *

Pro Tip 

When you start on a beat, go to the agency office or website that you're covering. Study the forms that the agency collects. What kinds of data are they gathering from the public? How are they organizing it, and what formats do they keep it in (Excel, Word, etc.)? Check the city, county, state or federal data portal to see if the information is readily available. Is some of it withheld? Find out why. This also serves as a road map for when you need to pull data later on. You'll already know what information is being collected.

* * *

There are many FOI form letters available with a basic Google search. You can download them as Word or Google documents and simply fill out the form and send to the agency by either email or registered US mail. But there are some free websites that help reporters not only write the letters but also track them. iFOIA.org from the Reporters Committee for Freedom of the Press is one of the best, as well as the Student Press Law Center Public Records Letter Generator.

Once you set up a free account on iFOIA.org, you can use several pull-down menus to select the agency and letter you want to use. When filling out the letter, be sure to be very specific about the records you want and what format you want them sent as (Excel, Word, shapefile, etc.).

Be specific about the type of record you want, what dates the records cover and what the topic is. Simply asking for "all of the mayor's email correspondence" is too broad. However, requesting the mayor's emails over the past three years discussing the public funding of a new bridge with the city's CFO is more specific and gives the agency a better road map to find the records. This typically cuts down on denials.

Listing a specific file format in the request is especially important; otherwise, the agency can send the documents as PDFs, which make it harder to extract data, though we'll explore some tools to do that later in the chapter. It's also wise to stipulate what costs you would cover for file transfer or, if need be, photocopying (sounds old school, we know). Make sure you have the name and title of the PIO and the correct mailing and email addresses on the letter.

"In many states, if a requester wants the data electronically, the agency must provide it in that format – can't convert to PDFs and print out on paper," Cuillier said. "Also, in some states, such as Arizona, they are required to provide the metadata, if requested."

The form letters stipulate that the agency has 30 days or less to respond to the request. iFOIA.org will notify you of when those 30 days are over, which is particularly helpful if the reporter is managing dozens of requests at once.

After the 30 days, the agency may fulfill the request, deny it or ask for more information from you. If denied, immediately file an appeal letter, which is available through iFOIA.org. If denied, an agency must list a specific state or federal statute backing the denial rather than citing a broad "internal policy" – a popular excuse.

"Agencies are always figuring out ways to game the system," Cuillier said.

"Using privacy apps like Signal has been going on for some years. Another scam is having files hosted on private servers or with a nonprofit, and then claiming the files aren't in their possession. The most common tactics are simply ignoring requests, inflating copy fees, or making up reasons for denial. The only way to stop it is to sue the agency, and unfortunately that does not happen enough."

Government officials can be clever in hiding their work from public records law. Aides to Washington, D.C., Mayor Muriel Bowser used WhatsApp, the Facebook-owned messaging app, to skirt public records laws in 2019. And in 2020, NCAA officials used a third-party platform to hide discussions about COVID-19 and the football season.

Cuillier said agencies create all kinds of reasons they can't deliver on a request. If they say the software can't export the data, then reporters should ask what company made it, contact that company and ask why their software can't export data.

"The company will say it certainly can and explain how, or link them up with the agency to help them export," he said. "I haven't encountered data yet that can't be exported."

He said some agencies will say they keep their data with a private vendor, or in a system created through a vendor, and that the whole system is proprietary, even the data dictionary, and can't be released because of an agreement.

"That is bogus," he said. "Agencies can't make agreements with companies that don't follow state public records law."

Pursuing public records will test your patience, but the payoff is worth it. Being persistent, especially after a denial, is key, Cuillier said.

"Keep at it until you get the records," he said. "Look at it as a maze, where you run into a dead-end, but just maneuver around until you get to the cheese.

"It takes persistence, tenacity and sometimes a little psychology. Don't take officials' denials at face value – check with records experts. Find out what other agencies do (peer pressure is effective at getting records). Team up with other journalism organizations. Always appeal denials (a third of the time it will kick them loose). And triangulate your records work with people sourcing – they go hand-in-hand. Sometimes you are better off getting the records leaked to you, rather than wasting time in court."

Appealing a Request Denial

If an agency denies your request for public records, you should appeal the denial immediately. Many agencies will fulfill the request immediately as the appeals letter is a sign that you mean business and won't go away until you get the record.

Although it varies by country, most appeals must be submitted within 30 days of the denial. Ask the PIO to review your FOI request and denial decision. Give reasons why you think the denial was wrong, cite any federal or local laws that apply and refer to any communication you've had with the agency in the past. And be sure to include a copy of your request and denial in the appeal document.

> For more data journalism tips, tricks, exercises and an archive of data portals, visit the Data + Journalism blog at http://dataplusjournalism.com.

Most agencies have 20 working days to respond to the appeal, though that time may vary by agency and country. Keep in mind that the agency might take a bit longer to decide on the appeal, and they must notify you of that delay.

Some government agencies, such as the US Department of Health and Human Services, post steps on how to appeal an FOI denial. Make sure you check with the agencies you cover for specific criteria they may have to prevent any further delays.

When denied and thwarted by a hostile agency/official, ramp up the pressure and make it harder for them to say "no" than to say "yes." That means writing stories about the denial, quoting experts, the elected officials and average people

affected. Survey surrounding agencies to show that the recalcitrant agency is an outlier (peer pressure is extremely effective). Submit 10 more requests about their expense reports, disciplinary records, claims against the agency, emails about your requests and anything else that might expose wrongdoing so common in agencies run by ego-driven, defensive and arrogant officials. During all this, keep your cool and take the high road.

Todd Wallack of WBUR in Boston has a great strategy when a government agency rejects one of his FOI requests. He immediately files an appeal letter with the agency and then files a separate FOI request requesting the internal emails from the agency officials discussing why his initial public records request was denied. Wallack says this accomplishes two things:

1. Prompts the agency to take the initial request more seriously and disclose the documents
2. The emails help him better understand the denial process

When you receive a denial or if your appeal is denied, there are many organizations you can turn to for help. The Society of Professional Journalists (SPJ) helps journalists all over the world in their pursuit of public records. SPJ's FOI committee plays a watchdog role on agencies and has dozens of resources to help.

The Reporters Committee for Freedom of the Press provides a variety of resources on public records and open meetings laws, also known as sunshine laws, right to know laws or FOIAs. These laws – primarily focused on the US – provide a legal basis for access to government records and meetings, with certain exceptions, and are used by reporters and news media organizations to inform the public on the workings of the government.

The organization has a FOIA Wiki on submitting requests, exemptions, administrative appeals and most other topics related to the federal FOIA. It also has a free iFOIA online tool to create, file and track federal, state or local public records requests. The tool not only helps you file the request letter to the proper agency but also sends you alerts for when the agency must respond and seamlessly provides appeals letters (Figure 1.1).

Figure 1.1 iFOIA.org's letter generator.

RCFP's Open Government Guide is a collection of every US state's open records and open meetings laws. Each state is arranged according to a standard outline, which helps journalists compare the laws in various states on specific topics.

Like RCFP, MuckRock helps you file, track and share public records requests. The site offers great tips on filing FOI requests as well as a state-by-state FOI laws guide.

Another useful resource is the National Freedom of Information Coalition, a nonprofit, nonpartisan organization of state and regional affiliates representing 39 states, commonwealths, territories and districts. The organization provides transparency education and guidance, offers financial support for FOI litigation, provides a state-by-state resource guide and undertakes evidence-based research projects.

Five Tips for Getting Public Records

There are so many strategies in acquiring public records, but here are five from David Cuillier that work for a lot of journalists internationally:

1. Get to know the records officers. Go to coffee or lunch with them (pay your own way) and ask what their job entails. Ask what bugs them about requesters. Ask for tips in requesting records. Ask what records they have seen that few people request and could be really interesting. If they and other record custodians see you as a real person, then they are more likely to help you and offer you further records that you might not have known about.
2. Hone your letter with specific keywords for agencies to use in their search. More and more agencies are searching for records electronically, so if you can provide keywords that will find what you need, it will speed things up. One of the fastest-increasing reasons for denial today is the response, "We do not have records responsive to your request." In other words, they couldn't find what you needed because they didn't have the right keywords.
3. Invest time into finding the existence of records. It is essential to know what records agencies have to craft a specific letter. That means scouring agency websites for references to reports, records and data. It means looking at forms that agencies have people fill out (that information likely feeds into a monster database that can be requested). Look at retention lists and public record logs. Go to agencies and talk to workers about what they do all day – and how they record that activity.
4. Find an hour each week to set aside as your records time. Turn off email and the phone and spend the time submitting a public records request and following up on pending requests. This will get agencies used to you, so when you come to them with sensitive requests, they won't recoil. Also, if you get amazing records out of just half of those requests, you will have 26 kick-butt stories produced every year, helping your news organization, your career and, most importantly, your community.

Using Data Portals

Data portals have emerged over the past decade as a rich resource for finding datasets for stories. It's also a great place to find story ideas.

Many governments have data portals – vast online catalogs where you can browse or search for open data on diverse subjects. The US government, for instance, has Data.gov, an index of hundreds of thousands of datasets posted by myriad agencies.

International organizations have data libraries, too. The data portals of the United Nations, World Health Organization and World Bank include datasets on topics ranging from education and economic indicators to crime and social trends.

Our World in Data is a free, open-source portal containing nearly 3,300 charts across 300 topics, including a deep COVID-19 data archive. The charts are easily embeddable into a web page, and the site offers an easy data download so you can build your own charts with the datasets.

The portal features datasets on the environment, the economy, census/population, industry and employment and public health, among other topics. It also includes a newsletter digest that updates when new datasets are added to the site (Figure 1.2).

Journalists in Europe often tap into Eurostat, the statistical office of the European Union (EU). The site produces European statistics in partnership with National Statistical Institutes and other national authorities in the EU member states and includes the statistical authorities of the European Economic Area (EEA) countries and Switzerland.

The Open Knowledge Foundation has assembled a list of about 600 data portals around the world. It may take some searching, but you can find relevant datasets by entering keywords and sorting the results by date.

Figure 1.2 Our World in Data's homepage.

Many government agencies and municipalities store thousands of routine datasets online and accessible to the public. These relieve PIOs from having to chase down data requests from reporters, lawyers and anyone else requesting the documents.

Read More: Glossary 

Journalists love to use jargon when working with data. Even the most basic terms can get confusing when not explained clearly.

In this book, we often refer to the word "import," which equates to "upload." We're importing or uploading the files to the web.

The term "export" equates to "download." We're exporting/downloading the data from the portal.

We'll define more terms throughout the book.

Portals give journalists the chance to inspect datasets and download them by simply hitting a button. Many contain routine data – city repairs, county or regional health data, national crime data, etc. They also include shapefiles that include states, provinces, bus and train lines, political districts, wards and districts. These shapefiles are critical for building maps.

Reporters covering local and regional news download datasets from city, county and state data portals. For example, the City of Chicago data portal contains city budget data, crime statistics over the past two decades, restaurant inspections, public transportation ridership data and many other data that are useful for reporters looking for public-records driven stories.

Data portals are relatively easy to find. Newsrooms often compile lists of them on shared documents, and there are many search tools beyond a basic Google search – for example, "Dublin Ireland data portal" – that can help you find what you're looking for.

If a Google search produces no results, the US Data Portal Github page provides links to dozens of state, county and city portals in one page. DataPortals.org has a searchable map of nearly 600 data portals from around the world. Better Data Portal offers search across Socrata data portals, and Statista provides clean datasets from all over the world for a fee.

* * *

Basic Search

Working from their Stanford dorm rooms in the mid-1990s, graduate students Larry Page and Sergey Brin built a search engine that used links to determine the importance of individual pages on the World Wide Web. They called this search engine Backrub.

Thankfully, that search engine evolved by 1998 and was renamed to Google, the go-to search engine we use in everyday life. But the search engine also has many tools and features that help data journalists quickly locate datasets.

Google's Powerful (but Hidden) Advanced Search Tool

It can be hard to find, but Google has an advanced search interface that lets you specify even more exactly what you're looking for: Words or phrases to include or exclude from the results, number ranges, the domain or site and other criteria.

You can find this tool by clicking on "Settings" in the lower right corner of Google's main search page. With the advanced interface, you might search for documents that have been posted on federal government websites in the past year about student debt, loans and tuition at colleges and universities around the world, etc. Simply use the various fields to isolate your search.

* * *

Google Search Operators

Dan Russell, a senior research scientist at Google, has compiled a list of dozens of search operators to help you find the information you want. Russell also offers a free course on Power Searching with Google.

Filetype

This operator helps you search the web for a specific type of file: .xlsx, .ppt, .pdf. Type this into a Google search field and see the results:

Filetype:xlsx Sydney Australia positive COVID-19 cases

The Sydney and New South Wales government health data pages typically show up on the first page of search results, along with a page of Sydney COVID-19 data from Our World in Data.

Site

Search within a specific website for information on sublevel pages. This is particularly helpful with government websites – and some media websites – that have poor search engines built in. Use Google to work around it by typing this into the Google search field:

Site:cdc.gov SARS

The result is a deep list of SARS-related pages useful to journalists: About page, fact sheet, FAQ, historical timeline, etc. (Figure 1.3).

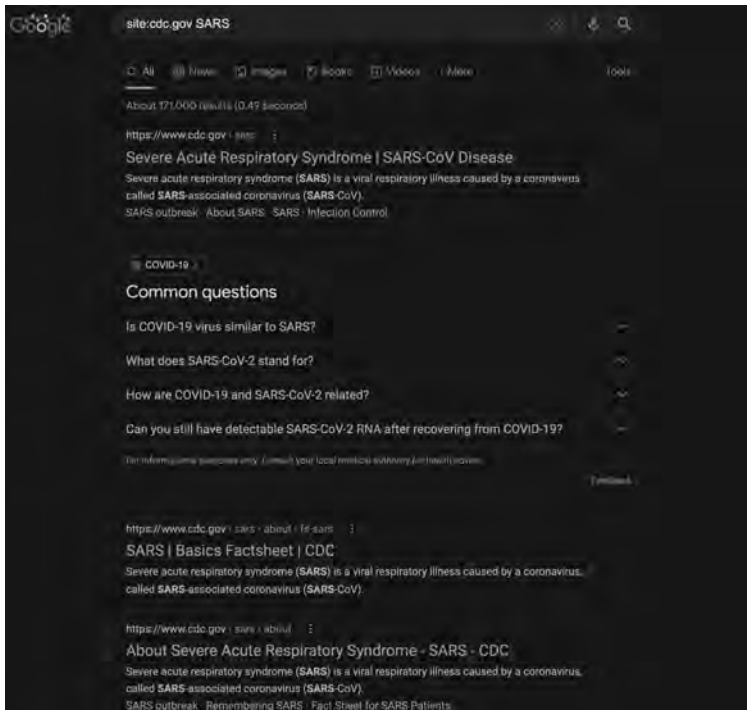


Figure 1.3 Site: operator search on the CDC.gov site for SARS.

There are many other useful search operators that Russell recommends:

Site minus site. A search like [site:nyc.gov -site:www.nyc.gov] will give you sites in NYC.gov that do NOT begin with WWW. That's handy for finding subdomains within a particular site, which you can then use **site:** to search.

Stars in site search. A search like [site:*.law.*.edu] will find all of the EDU sites with ".law" in the domain name. Also try: [site:*.nyc.gov] to match all of the NYC.gov sites with a subdomain. Also: [site:*.nasa.* inurl:education] gives lots of good clues about education sites at NASA.

* * *

Google's Dataset Search Engine

In 2018, Google launched its Dataset Search, a database of millions of datasets posted by academic researchers, nonprofit organizations, companies and government agencies (Figure 1.4). To have a dataset appear in the database, developers

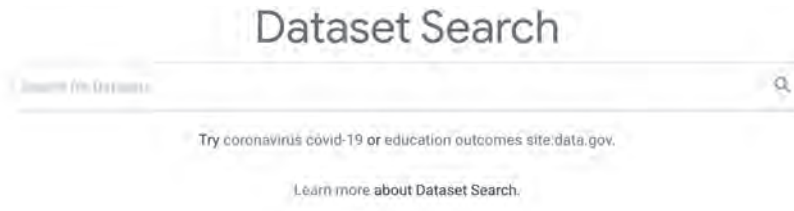


Figure 1.4 Google Dataset Search interface.

must format the data and provide detailed information about the dataset to Google's satisfaction.

Google's reasoning: Datasets are easier to find when they include clear and consistent supporting information, such as the data's description, source and file format. Those details can help journalists determine whether a dataset is worth downloading, thus saving hours of research time.

As you type keywords into the Dataset Search box, Google suggests possible datasets. Moreover, you can filter the research results by when the data was posted, its format and whether you must pay to download the data. Just use the filters tab at the top of the interface.

The tabs down the left side of the interface allow you to move to various datasets. As you click each tab, the description of the dataset – provided by the data's producers – appears on the right side of the interface. This includes the format, any special permissions (most are public records or available under Creative Commons licenses) and details of the dataset methodology.

It's important to note that Google doesn't fact-check the datasets. It merely guides you to the datasets and provides information about them so you can vet the data yourself. It's an incredibly efficient tool for finding clean datasets on deadline.

Journalists also can have their datasets listed in the search engine by completing the steps outlined on the tool's developer page.

* * *

Exercise

1. Go to Google.com and type in "US mass shootings data," and see what appears on the first screen. Are you getting datasets in your results? Try it with other countries. You likely will get news stories and blog posts, not datasets.
2. Now type this search operator text into Google.com: "filetype:xlsx US mass shootings". You should get a more robust search result featuring datasets from Kaggle and Stanford Libraries.

3. Now go to the Google Dataset Search tool, and type in US mass shootings. Use the toolbar down the left side to view the different datasets. Details of each dataset are featured on the right. Your first result is typically the best result, even better than using the search operator in step 2.

Video: How to use Google Dataset Search
https://www.youtube.com/watch?v=dxMretoIA3Q

Other Search Options

While Google is a valuable search tool for data journalists, there are other useful tools for searching the Internet for datasets, including:

- Bing, Microsoft's search engine
- Yahoo!
- Some journalists prefer DuckDuckGo or Startpage because they vow to protect your privacy by not tracking your search history. In China, which has banned Google, the most popular search engine is Baidu; in Russia, it's Yandex.

Academic Studies and Expert Sources

With any data story, it's important to seek outside experts who can help you understand your subject. Find them early in your reporting process as they can help guide you to good datasets and assist in the cleaning process.

Remember, someone somewhere studies the subject you're writing about. Ask them basic questions, and get their input through every step of the process. Academics have developed standard methodologies to study all kinds of things. You don't need to reinvent them. You'll end up with a better analysis and a measure of credibility you could never achieve on your own.

Research by scholars with deep expertise also can make your news stories more authoritative. But peer-reviewed journal articles may be too long and dense for reporters on tight deadlines. The Shorenstein Center on Media, Politics and Public Policy at the Harvard Kennedy School created Journalist's Resource, which curates and summarizes academic research relevant to newsworthy issues.

The website's database contains thousands of summaries of academic and governmental research on topics from economics and the environment to politics and social issues. The summaries translate jargon and abstruse statistics into everyday language. Journalist's Resource also includes tip sheets for reporters, including data journalism.

In addition, two specialty search engines – Google Scholar and Microsoft Academic – can help you find scholarly research. Scholar lets you search academic journals and case law and filter it by publication year, etc. Thousands of

academic journal articles are listed there, and many of their authors have contact pages that show a list of all of their publications. Each page has a "follow" button that alerts you when the academic publishes or edits an article. This is particularly helpful when writing about an ongoing issue and you want to keep up with the academic's work.

Google Scholar also is good for finding expert sources. Simply click on the author's name if hotlinked to go to his or her bio page (Figure 1.5). There, you'll find past publications, areas of expertise and more. Hitting the blue Follow button next to the bio subscribes you to that author so you receive an email when the author publishes.

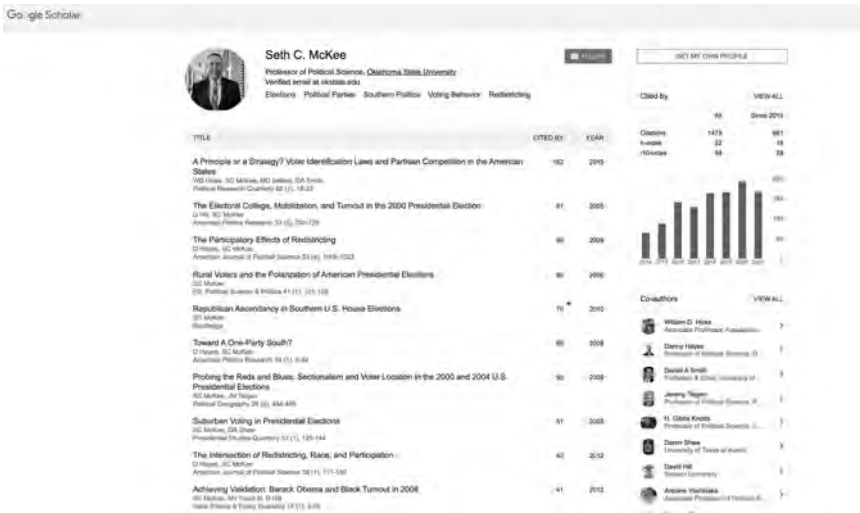


Figure 1.5 Google Scholar author page.

The authors typically have a link to a homepage or contact information, so reporters can reach out to them directly. If there's no link, simply Google the author's name and find the contact page.

You also can look for knowledgeable sources in online databases such as ExpertiseFinder and DiverseSources.org, which feature underrepresented voices about science, health and the environment. National Public Radio features a database of diverse expert sources organized by state, and Sources of Color assists both journalists and public relations (PR) pros with finding and pitching diverse experts.

The Journalist's Toolbox has an entire page devoted to expert source databases, including a deep archive of diverse experts.

Cision, the company that operates PR Newswire, has two services to connect experts with journalists:

- Help a Reporter Out – Journalists can send their story idea to HARO, and Cision will forward it to a network of media relations practitioners at universities, think tanks and other institutions. The PR staff then look for in-house experts who might contact the reporter.
- ProfNet – With ProfNet, journalists not only can pitch their stories to prospective sources but also sign up for email alerts about experts available to discuss timely topics in the news.

Hacking a Web Address to Find Archived Data

Many times, government agencies don't maintain an easy-to-navigate archive of their records on the web, providing only the most current dataset. To find older datasets, you can hack the web address of the most recent dataset as agencies have consistent URLs for naming files.

For instance, the US Postal Service publishes its quarterly performance report on its website. The data table shows the percentage of mail delivered on time in each postal district. Here is the page of quarter 3 data from 2020 (Figure 1.6).

Here is the website address for that table. Note that we boldfaced the fiscal year and quarter:

https://about.usps.com/what/performance/service-performance/fy2020-q3-single-piece-first-class-mail-quarterly-performance.html

To find quarter 4 data, simply change the q3 in the URL to q4:

https://about.usps.com/what/performance/service-performance/fy2020-q4-single-piece-first-class-mail-quarterly-performance.html

You also can change the year and the quarter to find more pages:

https://about.usps.com/what/performance/service-performance/fy2021-q1-single-piece-first-class-mail-quarterly-performance.html

This can be an effective way to sort through government sites and other poorly navigated websites to find data buried deep in the archives.

Evaluating the Information You Find Online

The rise in the volume and speed of data production might be overwhelming for some journalists, especially when they don't typically use large datasets for research and storytelling. But the urgency and eagerness to make use of data, and the technology available to process it, should not distract us from our underlying quest for accuracy.

Testing and verifying data is one of the most challenging parts of data reporting. It's also one of the most important, so don't rush the process. Build time into your reporting process to check and recheck your data.

1. One verification method is to compare datasets from different sources. For example, compare the homicide statistics compiled by local police with that

District	Overnight	Two-Day	Three-To-Five-Day
	Percent On Time	Percent On Time	Percent On Time
Capital Metro Area	**N/A**	**92.0**	**82.2**
Atlanta	N/A	92.1	83.5
Baltimore	N/A	88.9	73.6
Capital	N/A	88.9	81.4
Greater South Carolina	N/A	95.1	85.3
Greensboro	N/A	93.2	83.2
Mid-Carolinas	N/A	93.4	81.7
Northern Virginia	N/A	91.6	84.8
Richmond	N/A	91.0	80.3
Eastern Area	**N/A**	**93.6**	**83.6**
Appalachian	N/A	94.7	84.8
Central Pennsylvania	N/A	93.5	82.1
Kentuckiana	N/A	94.6	83.3
Northern Ohio	N/A	92.3	82.8
Ohio Valley	N/A	92.5	83.7
Philadelphia Metro	N/A	91.8	83.3
South Jersey	N/A	92.8	82.0
Tennessee	N/A	94.4	83.6
Western New York	N/A	95.2	84.0
Western Pennsylvania	N/A	95.7	86.7
Great Lakes Area	**N/A**	**90.2**	**80.7**
Central Illinois	N/A	91.8	82.0
Chicago	N/A	88.7	80.0
Detroit	N/A	73.5	64.7
Gateway	N/A	92.6	83.9
Greater Indiana	N/A	93.6	84.0
Greater Michigan	N/A	91.8	81.9
Lakeland	N/A	94.0	83.7
Northeast Area	**N/A**	**88.3**	**73.8**
Albany	N/A	93.6	83.0
Caribbean	N/A	90.0	67.5
Connecticut Valley	N/A	90.0	77.8
Greater Boston	N/A	90.9	77.8
Long Island	N/A	86.6	72.2
New York	N/A	68.4	53.6
Northern New England	N/A	94.0	76.5
Northern New Jersey	N/A	87.1	72.2
Triboro	N/A	74.5	60.9
Westchester	N/A	88.2	74.4

Figure 1.6 US Postal Service performance data.

of state police, logs kept by media outlets, etc. What doesn't add up? What slips through the cracks in one dataset may appear in another.

2. It's also important to check with the PIO or data experts in a government office or organization about what criteria are used in building the dataset. Data experts in the open records office are particularly helpful, because they work with the data all the time and compile reports for journalists who request the data. Ask them to explain what the fields mean.

Some data developers post these criteria in a Github page or in their Google Dataset Search post. You can compare these criteria among various datasets. For instance, some organizations classify a mass shooting as having three or more victims. Others use 10 or more. Some 13 or more. Those criteria greatly adjust the number of mass shootings in the database. It's important to explain those criteria to the reader and how it compares to other datasets.

Remember, data can be bad for lots of reasons. Sometimes, the records are wrong or incomplete. Other times, the actual methods for collecting the data produce inaccuracies. If the data are wrong, so is the story.

3. If your data is in a spreadsheet, do a quick sort and filter on the data, skills you will learn in Chapters 5 and 6 of this book. Sorting and filtering will expose any data missing from the sheet. Contact the PIO or source of the data to fill in the gaps.
4. Verify all conversions – Celsius/Fahrenheit, miles/kilometers, milligrams/micrograms, etc. Check your units. Don't confuse parts per million and parts per billion. Make sure your verbal descriptions are correct.
5. Keep a data diary that will help track of how you're analyzing your data. Keep a log of any changes or corrections you're making. List where you got the data, when, the software and steps you took to get it, clean it and analyze it. List it as a process. It's helpful in finding mistakes or replicating the process for future stories. A simple Google Doc will suffice.
6. Verify your data with actual, on-the-ground reporting. Does your data describe a place? Go there. Does it describe a person? Talk to them.

* * *

Tools and Resources from This Chapter

RTI Rating https://www.rti-rating.org/
FOIA Wiki https://foia.wiki/wiki/Main_Page
iFOIA.org https://www.ifoia.org/
Student Press Law Center Public Records Letter Generator https://splc.org/lettergenerator/
MuckRock https://www.muckrock.com/
Better Data Portal http://www.betterdataportal.com/
Data.gov https://www.data.gov/
United Nations Data https://data.un.org/
World Health Organization https://www.who.int/data/gho
World Bank https://data.worldbank.org/
Statista https://www.statista.com/
Our World in Data https://ourworldindata.org/
Eurostat https://ec.europa.eu/eurostat
Open Knowledge Foundation: DataPortals.org http://dataportals.org/

Google Advanced Search https://www.google.com/advanced_search
Power Searching with Google https://coursebuilder.withgoogle.com/sample/course?use_last_location=true
Google Dataset Search https://datasetsearch.research.google.com/
Video: How to Use Google Dataset Search https://www.youtube.com/watch?v=dxMretoIA3Q
Google Dataset Search Developers Page https://developers.google.com/search/docs/advanced/structured-data/dataset
Yahoo! https://www.yahoo.com/
DuckDuckGo https://duckduckgo.com/
StartPage https://www.startpage.com/
Baidu http://www.baidu.com/
Yandex https://yandex.ru/
Bing https://www.bing.com/
Expertise Finder https://expertisefinder.com/
DiverseSources.org https://diversesources.org/
NPR Sources https://training.npr.org/sources/
Sources of Color https://sourcesofcolor.com/
Journalist's Toolbox Expert Sources https://www.journaliststoolbox.org/category/expert-sources/
Help a Reporter Out (HARO) https://www.helpareporter.com/
ProfNet https://profnet.prnewswire.com/ProfNetHome/Profnet-Journalists.aspx
US Postal Service Quarterly Performance Report https://about.usps.com/what/performance/service-performance/fy2020-q3-single-piece-first-class-mail-quarterly-performance.html

* * *

Public Records and Search Tools

Beyond what we've covered in this chapter, here are some more helpful tools for finding data and public records.

Google search operators https://support.google.com/websearch/answer/2466433?hl=en
FEC Itemizer from ProPublica https://projects.propublica.org/itemizer/
Itemizer allows you to browse electronic campaign finance filings from the Federal Election Commission to see individual contributions and expenditures reported by committees raising money for federal elections. As of October 2018, these filings include Senate candidate or Senate party committees, which previously filed their reports on paper.
The FOIA Machine https://www.foiamachine.org/
Automate your FOIA requests.
Amazon's IRS Form 990 Filings https://registry.opendata.aws/irs990/

Machine-readable data from certain electronic 990 forms filed with the IRS from 2011 to present are available for anyone to use via Amazon S3.

Government Attic https://governmentattic.org/

Provides electronic copies of thousands of interesting federal government documents obtained under the Freedom of Information Act.

Journalist's Toolbox search tools page https://www.journaliststoolbox.org/category/search-engines/

Journalist's Toolbox public records and FOIA pages https://www.journaliststoolbox.org/category/public-records/

Social Searcher https://www.social-searcher.com/

Collates postings on social media networks.

* * *

Footnotes

FreedomInfo.org http://www.freedominfo.org/about-us/

Statista, Where Do Freedom of Information Laws Exist https://www.statista.com/chart/17879/global-freedom-of-information-laws/

National Freedom of Information Coalition, National Freedom of Information Laws https://www.nfoic.org/international-foi-laws/

Miami Herald, Fight Club https://www.miamiherald.com/news/local/community/miami-dade/article209052374.html

HHS.gov, How to File a FOIA Appeal https://www.hhs.gov/foia/faqs/how-do-i-appeal-a-denial/index.html

NPR, DC Officials Use WhatsApp https://www.npr.org/local/305/2019/10/09/768529012/d-c-officials-using-whats-app-for-city-business-may-skirt-open-records-laws?t=1570705980449

The Journalist's Resource https://journalistsresource.org/

The Journalist's Resource: Know Your Research https://journalistsresource.org/type/know-your-research/ An archive of tip sheets for journalists, including data journalism.

2 Searching the Deep Web

Samantha Sunne

Searching the web is an inveterate skill among reporters, but online information goes much deeper than what appears in a Google search. Being a data reporter means digging beyond a basic search and into the many databases and repositories of information available on the deep web.

The "deep web" is a term for all of the data that is not indexed by search engines – that is, not considered by Google and others as a potential search result. It includes things like Facebook profiles, records in a searchable database or the time and date of a YouTube video. It is not the "dark web," which is a highly anonymized, hard-to-reach part of the web that is not accessible through regular web browsers.

As surprising as it may seem, the "deep web" makes up most of the content on the web. This chapter will share examples of these kinds of "deep web" data sources and how to find similar ones for your stories.

But the best way to find data is the same as finding any other type of source: shoe-leather reporting. Often, your search for data will begin with an online query, but it won't stop there.

Journalism vets often recommend calling up sources or stakeholders – like the chief of police, the state auditor's office or an industry rep – and ask them about what data is collected and where. Better yet, stage a broad conversation where you can ask about data as well as other sources like documents, other human sources and topics to look into. Building a stable of sources, and establishing rapport with them, is as important in data reporting as it is in any other part of journalism.

This traditional form of reporting has the added benefit of being able to check your data, your theories and your findings. You can use expert sources, government officials, public relations representatives and other stakeholders to confirm your findings, get their opinion on them or at least get a feel for whether they sound reasonable. We'll talk about this more in Chapter 4, Cleaning Data.

In addition, databases, despite having "data" in the name, are often in a format that is not ready to be downloaded or analyzed. Usually, there are web pages, PDF documents, search bars and other doorways that create additional steps for the data reporter. Future chapters will address how to obtain and prepare this data for analysis, but for now, we will focus on how to locate them.

DOI: 10.4324/9781003273301-3

Pro Tip

When searching an online database, you can sometimes get it to return *all* of the records by leaving the search bar blank. You can also try entering wildcards like "*" and "%%". These tricks don't work with every database, but they are always worth trying.

Government Databases

A big part of the "deep web" is databases, which are often searchable from within a web page. The databases themselves may pop up in an online search, but the records they contain won't.

For example, the International Monetary Fund (IMF), a financial institution joined by almost every country in the world, has an online data portal including the organization's widely used World Economic Factbook.

The Factbook includes downloadable Excel files with economic data on dozens of countries, such as Canada's gross domestic product growth in 2019 (1.9 percent) or Kazakhstan's unemployment rate in 2020 (4.9 percent).

The Factbook itself may come up in a search engine, but the data points won't, because they are inside an Excel file hosted on that page. These data points are part of the so-called deep web.

Government open data portals, like Data.gov from the US, were discussed in Chapter 1 (Figure 2.1). Other useful databases include court records, campaign finance disclosures, crime maps, patents, health inspections, city budgets and many, many more. For a particular story or beat, it's most effective to ask human sources for ideas on where to find data.

How to Determine If a Government Source Is Reliable

In the data journalism field, government databases are considered more reliable than other sources because they are subject to laws like public information and anti-corruption rules. This means the government has more incentive to publish accurate and comprehensive information. But it doesn't mean a database is reliable just because it came from a government.

In 2013, a team of Russian journalists made a discovery that later would have an enormous impact on elections around the world. A group of online provocateurs were working together to sow discord in online groups around Russian (and later, American) elections. The group, it turned out, was being partly funded by the Russian government, which the journalists discovered through the government's data on financial contracts.

Another unreliable government database, though for different reasons, is the Federal Bureau of Investigation's (FBI) national dataset on crimes in the US.

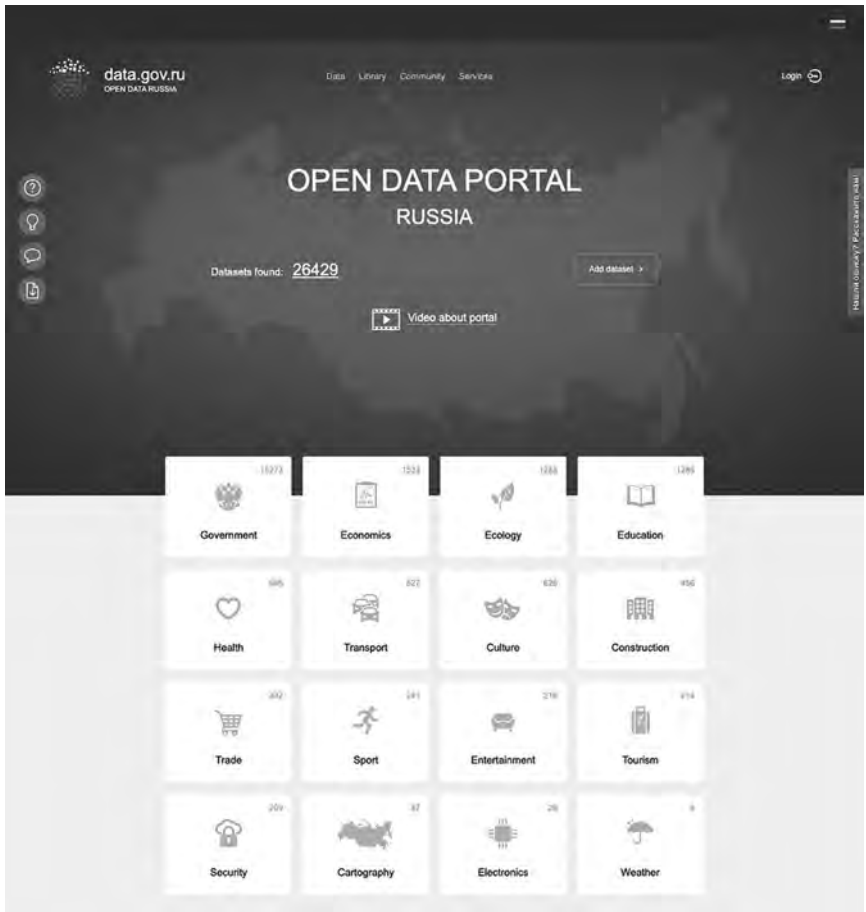


Figure 2.1 data.gov.ru, Russia's open data portal.

Because the FBI collects this information from individual cities, police departments and sheriff's offices, any error that is in their collection will be carried over to the national data. It faces the additional pitfall of errors being introduced in transfer, data being categorized differently or not being submitted at all.

The report "provides a wealth of information about crimes, but it is only as good as the agencies that report data," reporter Mark Fazlollah wrote in a guide for Investigative Reporters and Editors.

The FBI and the Russian data portal are examples of government databases that can be tricky but insightful sources nonetheless. In general, you should evaluate a dataset's reliability by fact-checking and cross-checking it with other sources. These can be human, like an expert source, or nonhuman, like another

dataset. When you have created your findings, always do bulletproofing, which we will address in Chapter 4.

Proxy Sources

If you can't find a proper dataset for your story, you can sometimes use something called a proxy source. This is data that is similar or is expected to correlate with the data you are looking for.

For example, Google Trends shows trends in Google searches. These can show, for instance, that searches for a political candidate rose dramatically after a debate. This does not mean that the political candidate became more popular, or even technically, more discussed, but it does show that they were more googled.

It can be useful to use proxy data in a story, but you should be clear what the data is actually representing. You may also want to note in your story that actual data was not available.

Exercise 1 

Let's explore crime data with the FBI Crime Data Explorer: https://crime-data-explorer.app.cloud.gov/

1. Open the Crime Data Explorer page and navigate to the Crime Data Explorer.
2. Using the Location dropdown, filter for your state. If you are not in the US, choose a state you are interested in.
3. In the Agency Select box, start typing the name of your city, university or a law enforcement agency you are interested in.
4. In Year Select, choose 2020.
5. Scroll down to look at the crime trends by state and agency. You can see data on the total numbers of crimes as well as subcategories.
6. Try to verify this data by finding the same information from the law enforcement agency itself. You can do this by looking at the agency's website, calling their public information officer or filing a public records request for the data.

The agency's data may or may not match what is in the FBI data. When in doubt, it is always best to get data from the original source, in this case, your local law enforcement agency.

Examples

These are a few examples of "deep web" data sources around the world. As a data journalist, you will probably need to find more localized datasets, which you can find through interviews and Internet searches of your own.

Data Portals	
Data.gov	The US's federal open data portal is a hub for agencies and departments from different levels of government.
data.go.jp	Many countries and states have their own open data portals. This one, from Japan, collects data from government agencies as well as some nongovernmental groups.
UNdata	The United Nations' data hub can be filtered down by topic and country.
Court Documents	
PACER	PACER is the US government's portal for federal court records. The federal court is different from city, state, appellate and other court systems.
International Court of Justice	The United Nations tries war crimes in this international court, with records dating back to 1946.
EUR-Lex	The Court of Justice of the European Union enforces laws established by the European Union and publishes court records dating back to the 1970s.
Legislation	
United Nations Treaty Series	This is a non-comprehensive collection of international treaties collected by the United Nations.
Library of Congress	Federal US legislation is tracked by this library, including bill sponsors and amendments.
Financial Disclosures	
US Federal Election Commission (FEC)	The FEC is the body responsible for tracking campaign finance disclosures for federal political campaigns in the US.
US Securities and Exchange Commission (SEC)	The SEC tracks financial information filed by companies that are being traded on the stock market, including their revenue and value.
IMF Data	The International Monetary Fund's portal includes international economic sources like the Direction of Trade Statistics.

Nongovernmental Databases

Sometimes, the best place to find data is from a nongovernmental party like an industry group, nonprofit or advocacy organization. This is typically because no government decided to collect the data or make it public. An example is the Berkeley Earth Surface Temperatures dataset, which contains hundreds of years of land and ocean temperature data. No governmental group has collected or published such a comprehensive inventory of historical temperatures, which is influential in climate change research.

These third parties can be think tanks, tech companies, academic departments, news outlets and more. However, each of these sources may have their own agenda, which should be taken into account both in writing your story and in assessing the data.

In many cases, you should be more skeptical of third-party sources than of government sources. Government releases are usually subject to mandated reporting, such as freedom of information (FOI) legislation, that enforce accurate and publicly available information. With a few exceptions, private groups don't face the same rules.

Nevertheless, these groups may be the best – or only – places to find data for your story. And in some cases, they are more accurate or comprehensive than the government that monitors the issue.

How to Determine If a Nongovernmental Source Is Reliable

The Berkeley dataset is one of many climate change-related datasets available online. The Independent Petroleum Association of America, an industry group of American oil and gas companies, publishes the *United States Petroleum Statistics* (Figure 2.2). The World Resource Institute, which aims to reduce worldwide carbon emissions, maintains a website called ClimateWatch.

Figure 2.2 The *United States Petroleum Statistics*, published by the Independent Petroleum Association of America.

Both of these groups have implicit biases, on either side of the climate crisis. This doesn't necessarily mean that they are being false or misleading, but it could affect which data they collect and how they present it.

So how can we determine which sources to use and how trustworthy the information is? The answer is the same as any other journalism source: due diligence and backgrounding. One way to do this is to read the group's About page. Sometimes you can find their funders, tax forms, donors or other background information.

For example, the Berkeley Earth Surface Temperatures is published by Berkeley Earth, a nonprofit that claims to provide neutral, fact-checked data on climate change. It shares its mission, funders and source data on the group's website, which all serve as indicators of trustworthiness. It has also been cited as a source by the European Environment Agency and the *New York Times*, meaning those outlets most likely did their own research and found the group to be credible.

It's also smart to search the group online, including in news sources. You can even post on a journalism forum with questions. Always look at your sources through a skeptical journalistic lens.

When in doubt, it's best to share this potential bias with your audience. This can be a simple statement like, "ClimateWatch, which advocates for a reduction in carbon emissions, collected this data from private companies." This clarifies your source and allows your audience to make their own decisions about how credible it is.

Exercise 2 

1. Open the *United States Petroleum Statistics* (shortlink: https://bit.ly/petroleumstatistics), a report containing data on oil rigs from 2017.
2. First, look at the data format and its source. Its first page indicates it was published by the Independent Petroleum Association of America, and the end of the URL indicates it is a PDF.
3. On the report, look at Table 1, "Exploration Activity." You may be able to copy and paste this table into a spreadsheet program, but it is easier with a tool like Tabula. We will address how to scrape data out of PDFs in the next chapter.
4. Describe two to three other sources where you could potentially get this data. Think: What government agency would track oil rigs? Would it be federal or state? Are there any advocacy groups or nonprofits that would do this as well?

When using nongovernmental data sources, it's often important to ask yourself whether it is the best (or only) place to find and use the data you are looking for. In Chapter 3, we will learn how to actually extract and use this report data in a technical sense.

Examples

Less Neutral Sources	
Political Organizations	Lobbying groups sometimes publish their own resources. Examples include the National Rifle Association's (NRA) database of State Gun Laws and the International Labour Organization's ILOSTAT hub.
Advocacy Groups	Advocacy groups are organizations that explicitly support a cause. Some create resources such as ClimateWatch by the World Resources Institute.
Stakeholders	A stakeholder is any person or institution directly involved in, or affected by, the issue at hand. A list of World Cup competitors is maintained by FIFA, which is the organizing committee and therefore an active stakeholder.
Industry Groups	Companies or private groups sometimes form to track information pertaining to a certain industry, such as the National Retail Foundation's State of Retail.
Think Tanks	Think tanks are organizations that produce research and recommendations on a certain topic. Some advocate for one point of view, like the Heritage Foundation, and others claim to be nonpartisan, such as the Brookings Institution.
Professional Associations	A professional association is a group that offers membership within a certain industry. This includes the American Bar Association (ABA), which offers an enormous amount of legal research on its website.
More Neutral Sources	
News Organizations	News outlets sometimes publish and maintain their own databases, such as ICIJ's Offshore Leaks and ProPublica's Dollars for Docs.
Universities	Many universities around the world track and publish data, like the UK Data Archive from the University of Essex.
Academic Papers	Scholarly papers can be an excellent source of data, as long as they are not funded by a stakeholder. One example is "Assessing the Performance of Freedom of Information" by the Columbia Law School.
Scientific Papers	Peer-reviewed papers also tend to be trustworthy sources, as long as they are not funded by stakeholders. An example is "Global Trends in Lifespan Inequality" by the Centre d'Estudis Demogràfics.
Transparency Organizations	Some groups simply advocate for transparency overall. These include Charity Navigator, the Sunlight Foundation and OpenCorporates.

Social Media

In addition to data collected by public and private entities, you can find a lot of data on people, companies, news topics and more from social media.

A good tool to have in your toolbox, no matter which platform you are using, is the syntax for advanced searches. "Syntax" is a word meaning the set of words and phrases you can use in a certain language.

Chapter 1 covered some of this syntax in Google. We learned how to search within websites with Google's "site:" filter, find files with "filetype:", exclude results with the minus sign (–) and find near-results with the asterisk (*) wildcard. These search operators are also available on other search engines and within social media sites themselves.

Many of these sites have advanced search pages, where you can click and type your search terms. But, in general, it's better to learn the syntax so that you can write your own queries. For example, on Twitter, searching "@WhiteHouse" returns any tweet that tags the White House account, but "from:WhiteHouse" returns only posts tweeted *by* that account.

Exercise 3

Here, we will find a news-relevant tweet with Twitter's advanced search syntax. The site has its own Advanced Search page, but it is only available to users logged in to their Twitter accounts.

1. In a web browser, type in "twitter.com/whitehouse". You do not need to have a Twitter account.
2. In the search bar, type "from:whitehouse 'build back better act' since:2020–01-01".
3. Click on "Latest" in the toolbar, below the search bar, to sort the tweets in chronological order.

This search takes advantage of three advanced search operators (Figure 2.3). "From:" limits the results to only tweets posted by the US White House's official Twitter account. The quotes around the phrase "build back better act" limit the tweets to ones mentioning this exact phrase, as opposed to individual mentions of "build" or "act." And the date filter, "since:", limits the tweets to ones posted after the date in the search.

Twitter offers a full list of search operators on its support site.

Figure 2.3 Twitter advanced search.

Because platforms and search engines often use different search operators, sometimes it is easier or more effective to do a search on Google instead. To do this, you will need to understand URLs and how they work. We will discuss URLs more in Chapter 3, Scraping Data.

Let's look at the URL of this Reddit post:

https://www.reddit.com/r/news/comments/ru92dd/tornadoes_from_rare_supercell_caused_damage_in/

Post IDs

In social media, posts, users, pages, locations and other criteria are often given IDs. For example, you can find the ID of your Facebook account by navigating to your profile and looking at the URL. It should look something like this:

"https://www.facebook.com/profile.php?**id=1156612157**"

Once you know the ID, you can use it in tasks like scraping and backgrounding.

The URL contains several different usable parts: "reddit.com", which is the website; "/r/news/", which is the "news" subreddit, a subsection of the site; "/comments/", indicating the content is a post; "ru92dd", a unique ID for the post; and the beginning of the post title.

Exercise 4

1. In Google, type the following search: "site:reddit.com/r/news inurl:/comments/ Coronavirus vaccine". This should return a list of Google results that are all r/news posts about COVID vaccines.
2. In your search, change "vaccine" to "conspiracy." How does this change your results?
3. Try adding quotes ("") around "Coronavirus vaccine". This should limit your results to that exact phrase.

By looking at how Reddit constructs its URLs, we were able to use Google's advanced search to find posts from a subreddit that mention a certain phrase. If a social media site's search feature doesn't help you do what you want to do, you can often use this workaround.

Pro Tip

You can create a Google Alert for these searches and save them to your Google account. This way, when a new result for your search is indexed by Google, you will receive an email in your inbox.

Google Alerts are not instantaneous – they typically update once a day – but they can be a good way to monitor for tweets or new posts if you can't do that within the social media site itself.

How to Determine If a Social Media Source Is Reliable

Social media companies are not typically required to disclose their data. For example, Discord may make an announcement of how many users it has, but it is not legally required to.

One exception is "public" companies – meaning those that are being traded on the stock market. In the US, when a company "goes public," it starts being regulated by the SEC, meaning it has to disclose certain information every year. Other countries have similar laws and regulations, so it may be helpful to look up what local requirements are, similar to researching FOI laws.

Because it is legally required, and a company can be fined for incorrect or misleading information, the reports are often considered a more reliable source than numbers published on a web page. There have, of course, been instances of companies lying in legally mandated reports, but the consequences are much higher.

On the other hand, getting data from a press release or announcement can be much easier and doesn't necessarily need to be disbelieved. As a journalist, it's always smart to state your source and sometimes even explain why one is more trustworthy than another.

Pro Tip

When googling, you can use the site filter with only the domain, in order to get any website on that domain. For example, "site:.gov" will return only government websites, while "site:.edu" will return educational institutions.

Examples

In addition to Google and advanced search pages, there are many third-party tools for analyzing social media content. Beware that many of these tools, especially if they're free, are in danger of falling out of date, moving behind a paywall or disappearing entirely.

That's why it's important to know the fundamentals, like search syntax, so you can keep up with whichever tools are available. The most foolproof way to navigate it is to ask your fellow journalists (or even just Google!) for the most recent active tools.

* * *

Facebook	
Who Posted What	This bare-bones site locates Facebook posts within a specific time range, location or profile.
Sow Search	Another simple page that offers a search of public Facebook posts, pages, profiles, videos and events by location and date.
CrowdTangle	This Facebook-owned browser extension lets users track posts and content being shared across different social networks, like Facebook, Twitter and Instagram.
Twitter	
Politwoops	ProPublica, a nonprofit news organization, created this database of tweets deleted by newsworthy politicians.
followerwonk	followerwonk searches bios, followers, profile names and other criteria to analyze a Twitter user or connection.
Social Bearing	This media analysis tool shows tweets, hashtags, photos and other Twitter content based on a keyword search.
LinkedIn	
LinkedIn Advanced Search	This social network for business connections offers a robust search feature that will filter by name, location, connection and current and past employment.
PhantomBuster	This is a tool meant to help influencers and marketers scrape data out of LinkedIn, Twitter and other social media sites. You can export spreadsheets of LinkedIn users, companies and more.

Other Social Networks	
InVID	InVID analyzes YouTube uploads and other content for metadata including location, upload time and thumbnails.
Snapchat Snap Map	The Snap Map is a widely used feature showing recent Snapchat broadcasts by user and location.
redditsearch.io	This tool by PushShift lets you filter Reddit searches by user, comment, subreddit, date and more.
Telegram Telegago	Telegago searches Telegram, one of the most popular messaging apps in the world, for messages, channels, contacts, bots and more.
Discord Disboard	Disboard searches public servers, or groups, on Discord, an application for group texts, video chat and streaming.
VKontakte Vk.watch	This paid, Russian-language tool searches users on VKontakte, one of the most popular social networks in Russia.
ClubhouseDB	This is a database of users and clubs on Clubhouse, a quickly growing social network popular among celebrities and public figures in the US.
Gravatar Email Checker	Search for someone's email address to see if they have a photo or avatar uploaded in Gravatar, a WordPress-based profile picture site.
SMAT	This tool created by open knowledge advocates searches for phrases on Gab, Telegram, Parler and smaller social networks.

Pro Tip

Social media searches will only return public content – that is, posts or profiles that are not switched to private or only visible to certain circles. The "deep web" does not mean a user is "hacking" into private information – just information that does not necessarily come up on search engines.

* * *

Tech Products and Archives

Introduction

A lot of data is collected by tech companies, and most of it is not shared. Some of this is in the "deep web" – the data that is not indexed by search engines – and some of it is simply stored in the companies' private servers. Still, companies do voluntarily share some information, and it can be a good catchall for other datasets or ways to find source information.

These include nonprofits like Wikipedia, tech products like Google Scholar and websites by data enthusiasts like data.world (Figure 2.4). Much data can also be found in past versions of pages, like the Internet Archive and Google Cache.

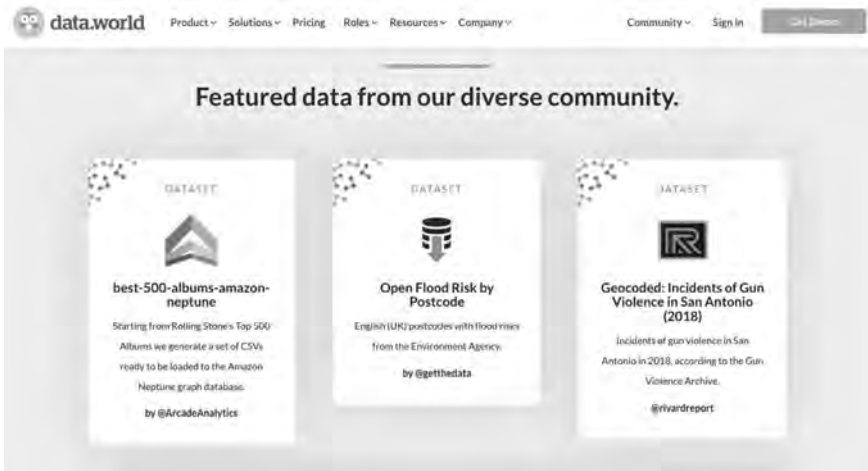


Figure 2.4 data.world featured datasets.

Exercise 5

The Internet Corporation for Assigned Names and Numbers (ICANN) is a nonprofit that maintains databases of many of the touchstones of the Internet, like IP addresses and domain names. One of them, WHOIS, contains the name, phone number, physical address and email address of whoever a website is registered to.

Since 2018, much of this information has been private, but a third-party research group archived many websites' registration in a tool called the WHOIS History Search: https://whois-history.whoisxmlapi.com/lookup.

1. In a web browser, open the WHOIS History Search. You may need to create a free account.
2. In the WHOIS History Search, type in a website you are reporting on or interested in.
3. Look at the most recent record. It most likely shows the registrant as a pseudonym or company.
4. Scroll down to an earlier year, like 2017. For many sites, the registration will show the site's registrant, LLC company, phone number and address.

WHOIS data can be an excellent way to background websites, track down sources or find connections between entities.

How to Determine If a Tech Source Is Reliable

Data provided by private tech companies and nonprofits should be evaluated along the same lines as nongovernmental sources. Again, remember to background the source (and ideally, the source's source) and keep a skeptical mind. They don't have mandated reporting and potentially may have a bias.

Examples

Transparency Initiatives	
Wikipedia	Wikipedia is widely considered a credible source for detailed lists and tables of data. Each entry links to its source for the information.
Data Commons	Data Commons, supported by Google, is an "open knowledge" repository that brings many open datasets into one place.
Internet Archive	The Internet Archive is a nonprofit repository of billions of web pages, as well as other media like books and audio files.
Data Collectors	
Statista	Statista is a private company offering an enormous collection of data on topics and sources around the world. Some of the sources for datasets are hidden behind a paywall.
Our World in Data	Our World in Data is a global data repository owned by a nonprofit and run by a research team at Oxford University.
Kaggle	Kaggle is a crowdsourced repository where millions of data enthusiasts upload datasets. It is owned by Google.
data.world	data.world bills itself as a collaborative data community and has been used by the Associated Press to share data with member organizations. You can search for datasets or share your own.
Data Products	
Google Scholar	Google Scholar is an excellent place to find academic and scientific papers as well as court records. Some of these papers include their research data in full.
Google Trends	Google Trends tracks the billions of Google searches performed every day and displays them as trendlines.
Westlaw	Westlaw is a legal information database owned by Thomson Reuters. It offers a large archive of laws, case records and court transcripts.
LexisNexis	LexisNexis offers legal records as well as an enormous collection of public records, such as property deeds and tax assessments.

Create Your Own Database

If you can't find or access the database you are looking for, you may want to create your own. One advantage is that you can later publish the dataset, creating useful resources for your audience. Chapter 11, Ethics, Trust, Transparency and Posting Data Online, will cover this in more detail.

The first step is to determine whether it's worth the effort. Downloading a dataset from the web is one thing, but creating your own involves a lot of time, effort and planning. That's in addition to the research, due diligence and bulletproofing that you always do.

For the technology side, make sure you are separating out values and variables as much as possible. For a table of the FIFA World Cup, it might make sense to enter one winner as "Italy 1934." But what if, later, you want to be able to show a smaller list of all the 1934 competitors? Or all the years that Italy won? From a computational standpoint, it would be better to enter "Italy" and "1934" in separate columns.

Especially if you are creating your own file, always keep a copy of raw data files somewhere else. These should be accessible somewhere on your computer, like in a folder for the story, but should be entirely untouched from when you downloaded them. You can give them a name like "FIFA raw data."

* * *

Exercise 6

We're going to practice planning a story on the most successful soccer teams in the world. FIFA, the organization that plans the international soccer World Cup, has a list of the winners on its website dating back to 1930. Each year gets a web page with a list of the top four winners, looking something like this (Figure 2.5):

Figure 2.5 FIFA's list of World Cup finalists in 1938.

This website would be a good candidate for scraping – computationally extracting the data – but for now we will form a plan for collecting the data manually.

1. In a word document, write an outline for a story on which countries have the most successful teams. Rely on your earlier journalism training: What is the most important aspect to the story? What information does the audience *most* want to know?
2. Write a lede with the letters "TK" (to come) to note where the data and findings will go. For example, "France won TK World Cups from 1930 to 2022."
3. Create a list of the information you would need to find this out.
4. Next, create a list of rows, separating out each variable, so you can run the proper analysis. There are several ways to design your table, but here are two examples (Tables 2.1 and 2.2).

Table 2.1 A table of World Cup winners by year

Year	*First Place*	*Second Place*	*Third Place*	*Fourth Place*
1930	Uruguay	Argentina	US	Yugoslavia
1934	Italy	Czechoslovakia	Germany	Austria

Table 2.2 A table of World Cup winners by country

Country	*Year*	*Placement*
Uruguay	1930	1
Argentina	1930	2
US	1930	3

As you can see, it can be a lot of work to manually enter this data. Each of these tables has advantages and disadvantages when it comes to the ease of your analysis, which will come later.

Pro Tip

Many journalists also recommend creating a "notes" or "miscellaneous" field, typically all the way to the right in the spreadsheet. You don't even need to know what it is for yet – but it will act as a catchall for annotations, reminders, attributions or unanswered questions.

Make sure to create it as a separate column so that you aren't combining values and comments in the same cell.

* * *

Examples

Once you have designed your database and found your source, there are many ways to collect the data.

Manual Entry

Manual data entry – the method we used in the FIFA exercise – can be one of the most time-consuming but also one of the most precise ways to collect and store data. This can be anything from one reporter keeping track of their findings in a spreadsheet to a team of journalists creating databases in advanced programs.

Two good tools for this method are Google Sheets and Airtable. Both are free, or have free tiers, and offer sophisticated features for multiple users to store, tag and organize data.

Past data journalists have also taken advantage of resources like students, civic data enthusiasts and short-term hires for data entry. In 2021, Reuters won a Pulitzer Prize for a project on qualified immunity for police. Part of the project relied on students at Stanford distilling thousands of court records into a database (Figure 2.6).

Crowdsourcing

Crowdsourcing means soliciting data or volunteer work from the public. Survey tools, like Google Forms and Survey Monkey, can make the technological aspect of this data collection easy.

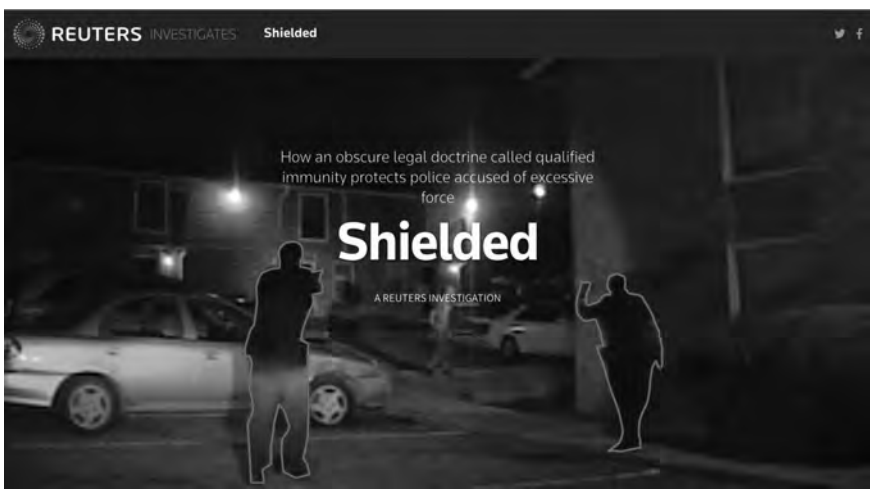


Figure 2.6 The 2021 Pulitzer Prize for Explanatory Reporting award winning project, Shielded.

Crowdsourcing is often considered one aspect of "engagement reporting" – making the audience more of a player in a story's reporting than a passive recipient. In 2017, ProPublica published a call for submissions with the title "Do You Know Someone Who Died or Nearly Died in Childbirth? Help Us Investigate Maternal Health." The result was a widely read story on childbirth-related deaths and injuries in the US.

Combining Existing Data

Sometimes, data is available, but it doesn't quite suit your purposes. For a story on wildfires in California, for example, you could collect data from the National Interagency Fire Center, the California Department of Forestry and Fire Protection, the University of California – Riverside or other sources. Each of these will have different advantages and disadvantages to their data collection.

Because each of them tracks and records data on wildfires in a different way, combining them might mean manually entering them in a spreadsheet. If they have matching columns, you can join them using a tool called Structured Query Language, to be discussed in Chapter 8.

Footnotes

IMF Data https://www.imf.org/en/Data
World Economic Outlook https://www.imf.org/en/publications/weo
World Economic Outlook Update January 2021 https://www.imf.org/en/Publications/WEO/weo-database/2020/October
Global Investigative Journalism Network, How They Did It: The Real Russian Journalists Who Exposed the Troll Factory in St. Petersburg https://gijn.org/2018/03/26/real-russian-journalists-exposed-troll-factory-st-petersburg/
Hidden Crimes: UCR Data, and What's Not There https://www.ire.org/product/tipsheet-3313/
FBI Crime Data Explorer https://crime-data-explorer.app.cloud.gov/pages/home
Data.gov Data.gov
Open Data Portal Russia Data.gov.ru
UNdata https://data.un.org
PACER https://pacer.uscourts.gov/
Search | International Court of Justice https://www.icj-cij.org/en/advanced-search
International Criminal Court https://www.icc-cpi.int/Pages/cases.aspx
United Nations Treaty Series https://treaties.un.org/pages/UNTSOnline.aspx?id=3&clang=_en
Library of Congress https://www.congress.gov
US Federal Election Commission https://www.fec.gov/data/
US Securities and Exchange Commission https://www.sec.gov/dera/data
Berkeley Earth Surface Temperatures http://berkeleyearth.org/
United States Petroleum Statistics https://www.ipaa.org/economics/
Petroleum Statistics shortlink https://bit.ly/petroleumstatistics

ClimateWatch https://www.climatewatchdata.org/
Independent Petroleum Association of America https://www.ipaa.org/
State Gun Laws https://www.nraila.org/gun-laws/state-gun-laws/
ILOSTAT https://ilostat.ilo.org
FIFA World Cup https://www.fifa.com/tournaments/mens/worldcup
National Retail Federation https://nrf.com/
State of Retail https://nrf.com/topics/economy/state-retail
About Heritage https://www.heritage.org/about-heritage/impact
About Us | The Brookings Institution https://www.brookings.edu/about-us/
American Bar Association https://www.americanbar.org
Offshore Leaks Database https://offshoreleaks.icij.org
Dollars for Docs Data (2017–2018) https://www.propublica.org/datastore/dataset/dollars-for-docs
UK Data Archive https://www.data-archive.ac.uk/
Assessing the Performance of Freedom of Information https://www.sciencedirect.com/science/article/abs/pii/S0740624X10000614
Global Trends in Lifespan Inequality https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0215742
Charity Navigator https://www.charitynavigator.org/
Sunlight Foundation https://sunlightfoundation.com/
OpenCorporates https://opencorporates.com
Twitter Advanced Search https://twitter.com/search-advanced?lang=en
Rules and filtering: Standard v1.1 https://developer.twitter.com/en/docs/twitter-api/v1/rules-and-filtering/search-operators
Tornadoes from Rare Supercell Caused Damage in Georgia r/news https://www.reddit.com/r/news/comments/ru92dd/tornadoes_from_rare_supercell_caused_damage_in/
Who Posted What https://whopostedwhat.com
Sow Search https://www.sowsearch.info
CrowdTangle https://www.crowdtangle.com/
Politwoops https://projects.propublica.org/politwoops/
Followerwonk https://followerwonk.com
Social Bearing https://socialbearing.com
LinkedIn Advanced Search https://www.linkedin.com/search
PhantomBuster https://phantombuster.com/
InVID Project https://www.invid-project.eu/
Snap Map https://map.snapchat.com/
Reddit Finder https://archivesort.org/redditfinder
Telegago https://cse.google.com/cse?&cx=006368593537057042503:efxu7xprihg#gsc.tab=0
Public Discord Servers https://disboard.org/servers
Vk.watch https://vk.watch/
Clubhouse Database https://clubhousedb.com
Gravatar Email Checker https://en.gravatar.com/site/check/
SMAT https://www.smat-app.com/

WHOIS History Search https://drs.whoisxmlapi.com/whois-history
Wikipedia https://en.wikipedia.org/wiki/Main_Page
Data Commons https://datacommons.org/
Internet Archive https://archive.org/
Statista https://www.statista.com/
Our World in Data https://ourworldindata.org/
Kaggle https://www.kaggle.com/
data.world https://data.world
Google Scholar https://scholar.google.com/
Google Trends https://trends.google.com/trends/
Westlaw https://legal.thomsonreuters.com/en/westlaw
LexisNexis https://www.lexisnexis.com/en-us/gateway.page
Shielded https://www.reuters.com/investigates/section/usa-police-immunity/
How Stanford Students Helped with a Pulitzer Prize-Winning Project https://news.stanford.edu/report/2021/06/15/stanford-students-helped-pulitzer-prize-winning-project/
Do You Know Someone Who Died or Nearly Died in Childbirth? Help Us Investigate Maternal Health https://www.propublica.org/getinvolved/help-propublica-and-npr-investigate-maternal-mortality

3 Scraping Data

Mike Reilley and Samantha Sunne

Reporters often grouse when government officials fulfill a public information request by sending a PDF or a web page with data embedded in tables. This is often done to frustrate reporters, even after they've specifically asked for the data to be sent in a spreadsheet, Word doc or other usable format.

Appealing to get the data in the requested format can take days, even weeks. And government agencies also may charge for that.

But there are many free tools to solve this problem and extract data from web pages and pesky PDFs.

Data scraping is the process of extracting information from a source file into a spreadsheet, when it's more difficult than simply clicking "Download." You can "scrape" data from a website, PDF, image or other document. It's an efficient way to get data and, in some cases, to channel that data to another website.

Lena Groeger, a journalist, designer and developer with ProPublica, often must scrape data to build interactive graphics, databases and other data-driven projects.

"A web page is basically a bunch of stuff that gets downloaded," she tells journalists she trains on scraping.

"Really, that's it. Slightly more technically, it's a bunch of different files that get downloaded – some text files and maybe some images. The best way to see the "behind the scenes" of a webpage is to use a tool called the web inspector to actually look at these files."

The Web Inspector is a panel available in every browser that lets you navigate through the components that go into rendering the web page you see. It also has several features to help achieve better performance, find bugs, check mobile views and more (Figure 3.1).

For example, go to WashingtonPost.com, then right-click and select "Inspect Element." You can now look at the contents of the files making up the page, including the underlying HTML, the Cascading Style Sheets (CSS) that style the page and much more. Viewing a page with a Web Inspector also shows code for tables <table>, table rows <tr> and table data <td> that can be scraped into a Google Sheet.

A "table" is a word you will come across often in your data journalism journey. A table is generally referred to as anything with rows and columns. You'll find tables

DOI: 10.4324/9781003273301-4

Figure 3.1 WashingtonPost.com viewed through the Web Inspector on Google Chrome.

in PDFs, spreadsheets, pieces of paper and many other sources. In this case, it is a very specific HTML element that is identifiable by the <table> tag.

Data scraping gives journalists extra artillery in acquiring data, which we explored in Chapter 1. If seeking data from a source doesn't work, try downloading or scraping it on a computer (Figure 3.2).

Typically, journalists scrape data to move it into a spreadsheet so they can manipulate and analyze it to find patterns and develop story ideas. This data often explores assumptions or theories journalists have about an issue or story idea. We'll refine the storytelling process in later chapters, but first, we must master the basics of obtaining data from the web and from PDFs.

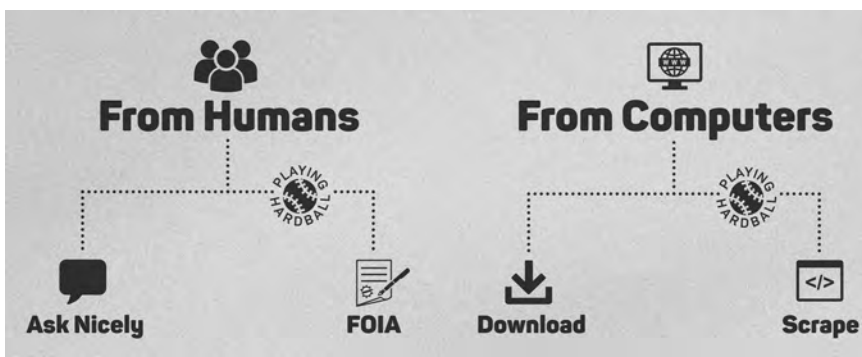


Figure 3.2 Obtaining data from sources diagram (Illustration/Billy O'Keefe).

Scraping from the Web

At ProPublica, Groeger creates influential and award-winning infographics by combining programming knowledge with visual design and human-centered storytelling. In order to build these incredible data visualizations, Groeger must first find the datasets that power those graphics. Sometimes, she'll simply download a file from a web portal. But other times the data is nested in web tables without a download button. So how can she extract it? This is a subset of data scraping called "web scraping."

There are many tools at her disposal, but one she often reaches for is a simple scraping formula: =IMPORTHTML("URL", "ELEMENT", NUMBER OF ELEMENT ON PAGE)

Many of us have used spreadsheet formulas before to solve a math problem. If you are new to formulas or functions, we will cover them more in Chapter 5.

This formula tells your Google Sheet to go to the web and scrape a page located at a specific uniform resource locator (URL). You'll select the HTML element to scrape (usually a table) and where that element appears on the web page. For this last part, you'll often type 0 (zero) as it tells the sheet to go to the first table on the page.

An HTML element is an individual component of a much longer "HTML document" – aka, the web page you are looking at. Elements are recognizable by their distinctive <> tags. That is, if you find "<table>" in the HTML of a web page, that means you are looking at the beginning of a table element. It ends with a "</table>" tag.

The scraping formula works because web tables have what are called "nested elements," elements that are embedded inside other elements. So within an HTML

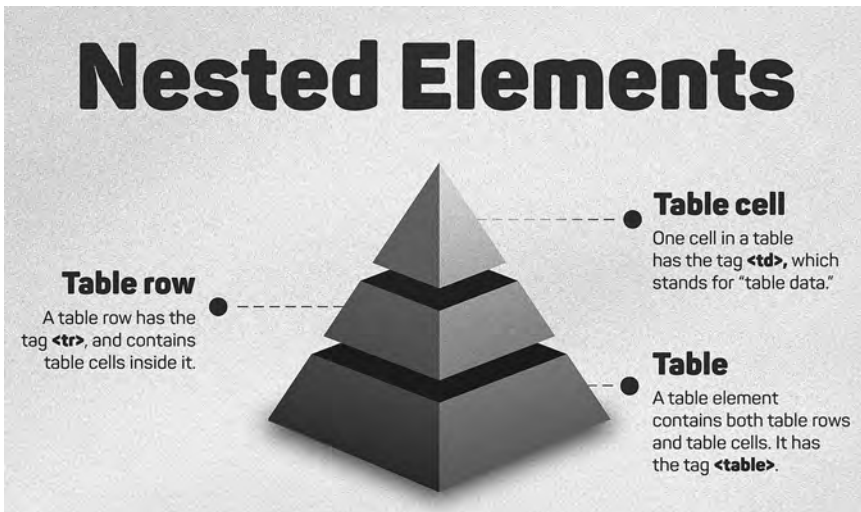


Figure 3.3 Nested Tables diagram (Illustration/Billy O'Keefe).

<table> element, you'll find a table row <tr>, and within those rows, you'll find table data <td> code that creates the small cells that include the data we seek. Because of this nested format, we can identify and scrape a table element and all of the elements inside of it (Figure 3.3).

The scraping formula also provides another feature helpful to journalists. Once linked to a web page, the formula continues to update the spreadsheet as the web page is updated. For instance, if you scrape an election results page on election night, the results will continue to update in the sheet as the web page is updated, up to once a day. This is an excellent option for reporters tracking government pages that update daily, weekly, monthly, etc.

To do analysis or to save a snapshot of the data, you must make a copy of the dataset and work off that copy. Google Sheets won't allow you to work on the scraped page. Once you start editing, you'll get an error message.

* * *

With this kind of scraping, the first step is to understand the URL you are going to scrape from: https://www.fdic.gov/resources/resolutions/bank-failures/failed-bank-list/. Web scraping, and this method of web scraping in particular, requires very specific URLs to work. In this case, we are looking at a website called "fdic" on a government domain, that is, ending in ".gov". We are looking at a specific web page on that website called "failed-bank-list", and it is in a section called "resources" and various subsections.

It's important to understand the difference between websites and web pages, especially with scraping. If you don't cite this specific URL – for instance, if you try to scrape a table on "fdic.gov" – you will get very different results.

Exercise 1

The best way to learn how to use the scraping formula is to test it out on some datasets. In this book, we will use Google Sheets for a variety of data tasks, because it is free and only needs to be run in a web browser. If you do not have a Google Drive account, you can create one for free.

For this exercise, we're going to scrape a table of failed banks tracked by the Federal Deposit Insurance Corporation (FDIC). The FDIC is a US government agency that insures banks so that people feel comfortable depositing their money there. If you look through the list, you can see that several small banks in various US states have closed their doors over the years. This web page lists closures going back to the year 2000.

1. To get started, open a Google Sheet by going to your browser and typing “sheets.new” into the URL bar and hitting return. A spreadsheet will appear. If you need to, sign into your Google Drive account or create a new one.
2. Label your spreadsheet: Click in the upper left and name it “FDIC scrape.”
3. Next, type in this formula. Do not copy and paste, as this is likely to cause errors.

   =IMPORTHTML("URL", "ELEMENT", NUMBER OF ELEMENT ON PAGE)

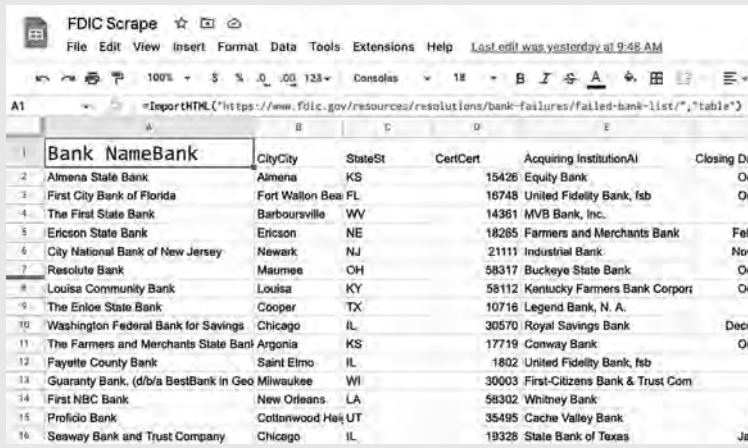
4. AddthisURLtotheformula:https://www.fdic.gov/resources/resolutions/bank-failures/failed-bank-list/
5. Enter “table” as the element. Be careful not to delete any code, and don’t leave a space after the web address when pasting it in.
6. Now add the table number (in this case a 0 as you’re starting at the top of the first page). It should look like this:

   =IMPORTHTML("https://www.fdic.gov/resources/resolutions/-bank-failures/failed-bank-list/", "table", 0)

7. Press enter. If you get an error message, review the steps above, copy and paste your error message into Google or see the Troubleshooting Common Spreadsheet Errors section in Chapter 4.

   ***Note:** Depending on your Google Sheets language settings, the delimiter in the function could be “,” or “;” but is usually a comma.

8. If done correctly, the entire table of banks should appear in your spreadsheet. Once you have scraped the page, do a slow scroll to make sure you haven’t missed any data or there are no garbles (Figure 3.4).
9. Now you’re ready to edit, sort and filter. There’s just one problem: If you go to type on the screen after you scrape, your data disappears. This is because the spreadsheet is linked to the web page, as discussed earlier. When the web page updates, so does the sheet. That prevents us from editing.
10. To remedy the issue, highlight all the data on the screen and copy it. One way to highlight everything without scrolling down is by pressing Ctrl+A or Cmd+A. Then click on the Plus Sign in the lower left corner of the Google Sheet to open a new tab. Click on the “Sheet 2” language on the tab to rename it “EDITS” or whatever you want.



	A	B	C	D	E	
1	Bank NameBank	CityCity	StateSt	CertCert	Acquiring InstitutionAI	Closing Dat
2	Almena State Bank	Almena	KS	15426	Equity Bank	Oct
3	First City Bank of Florida	Fort Walton Bea	FL	16748	United Fidelity Bank, fsb	Oct
4	The First State Bank	Barboursville	WV	14361	MVB Bank, Inc.	
5	Ericson State Bank	Ericson	NE	18265	Farmers and Merchants Bank	Febr
6	City National Bank of New Jersey	Newark	NJ	21111	Industrial Bank	Nove
7	Resolute Bank	Maumee	OH	58317	Buckeye State Bank	Oct
8	Louisa Community Bank	Louisa	KY	58112	Kentucky Farmers Bank Corpor	Oct
9	The Enloe State Bank	Cooper	TX	10716	Legend Bank, N. A.	
10	Washington Federal Bank for Savings	Chicago	IL	30570	Royal Savings Bank	Decen
11	The Farmers and Merchants State Bank	Argonia	KS	17719	Conway Bank	Oct
12	Fayette County Bank	Saint Elmo	IL	1802	United Fidelity Bank, fsb	
13	Guaranty Bank, (d/b/a BestBank in Geo	Milwaukee	WI	30003	First-Citizens Bank & Trust Com	
14	First NBC Bank	New Orleans	LA	58302	Whitney Bank	
15	Proficio Bank	Cottonwood Hei	UT	35495	Cache Valley Bank	M
16	Seaway Bank and Trust Company	Chicago	IL	19328	State Bank of Texas	Jan

Figure 3.4 Scraping with ImportHTML.

11. To paste the data into the new sheet, click on cell A1 and then click Edit > Paste Special > Values only. Presto! Your data appears in the new sheet and is now editable. You still have your original scraped sheet in the first tab, which will update up to once a day.
12. Now you have your dataset in a format you can edit. But there's another issue (who said scraping was easy, right?): It appears that the date format changed to a number that looks something like a ZIP code. This sometimes happens when cutting and pasting dates that use both letters and numbers.

Pro Tip

You always, always, always work off a COPY of your dataset in case you mess something up. You may want to save an offline copy by clicking File > Download > Microsoft Excel.

It's an easy fix. Just click on the letter C to highlight the column, then go to the pull-down menu at top of the page and select Format > Number > Date, and it will reformat the dates for you.

You have probably noticed that scraping the data directly from a website came with some garbles – for instance, the header row contains repeated terms, like "CityCity". Watch for these small glitches as you scrape and move data, and check it after each step in the process. We'll talk about this more in Chapter 4.

Now your dataset is ready to be sorted, filtered and analyzed. There will be more spreadsheet work and more scraping in Chapters 5, 6 and 9.

Video: Watch how to scrape web pages using the =IMPORTHTML formula as well as scrape PDFs with Tabula. Tabula is a free, secure tool you can download to your desktop from http://tabula.technology.

- Video: Scraping in Google Sheets and Tabula: https://bit.ly/sheetscrape

Exercise 2

Here are some more pages you can practice scraping with this formula. Just plug one of these web addresses into the formulas below.

Ireland Trolley Ward Watch https://www.inmo.ie/Trolley_Ward_Watch

National Interagency Fire Center: US Wildfires https://www.nifc.gov/fire-information/statistics/wildfires

1. Type "sheets.new" into the browser window.
2. Paste both links into the area where it says URL in the formulas below.

   =IMPORTHTML("URL", "table", 0)
   =IMPORTHTML("URL", "table", 0)

3. Copy the formula and paste into cell A1 in your sheet, and then repeat the steps for making a copy of the data.

When You Have Multiple Tables on a Page

Some web pages have several tables, like this example from Washington, D.C.'s, COVID-19 testing site. We're looking for the table that appears fourth on the page and focuses on the addresses of testing centers. To scrape only this table, you must tweak the table number at the end of the formula to 3. It's not 4, as your table numbers start with 0. Here's the formula to try:

=IMPORTHTML("https://coronavirus.dc.gov/testing", "table", 3)

This table would be good for a map or list in a story, as it contains the addresses of the testing locations.

Scraping More Than Tables

Besides IMPORTHTML, Google Sheets offers a few other formulas for scraping data from the web.

- IMPORTRANGE: Imports a range of cells from a specified spreadsheet.
- IMPORTFEED: Imports an RSS or ATOM feed.
- IMPORTDATA: Imports data at a given URL in comma-separated value (CSV) or tab-separated value (TSV) format.

You can also use a more flexible formula called IMPORTXML. It's very similar to IMPORTHTML but with a much more complex section to it:

=IMPORTXML("URL", "XPath")

What makes this formula flexible is the second parameter, the XPath. A "parameter" is the variable you are entering into a formula so the formula knows what to do. The IMPORTXML formula has two parameters: a URL and an XPath.

An XPath is like an address to a very, very specific point on the web. Think of it like a physical address. Let's say a mailman is walking down the street, delivering letters and packages all over the city. He reaches into his bag and pulls out an envelope with the name "Mary Contrera" on it. How does he know where it goes?

The mailman needs to know not just what city Mary lives in but what road she lives on and where her building lies on the road. Sometimes he needs to know the number of an apartment within that building or the floor it's on. Luckily, the envelope has an address on it, like this:

421 Crescent Drive, Apt. 4, Maughantown, NY 80123

This tells the mailman that the letter is going to the building numbered 421, on the street "Crescent Drive," in the town of Maughantown in the state of New York. This address is read from smallest to largest.

Right away, you might notice some not-insignificant leaps of logic here. For instance, when the mailman reads the mailing address, he actually reads the zip code first – the last part. A ZIP code itself is made up of multiple codes designating areas within areas.

> For more data journalism tips, tricks and exercises, visit the Data + Journalism blog at http://dataplusjournalism.com

And this address has an apartment number between the street name and city name, further complicating the procedure. Mail deliverers and computers alike are trained to read these addresses and sort the mail accordingly. With this method, the US Postal Service is able to sort and deliver more than 173 million pieces of mail every day.

The postal address system makes that possible, and so does the XPath system on the web. Just like the mailman, a web browser needs to know what to show you, and it needs to process billions or even trillions of these requests each and every day.

Let's say you're doing a story on the Mountain National Bank and want to know what state it's in. How does the computer know to show you that, and not, say, the state of the Mountain Heritage Bank? It doesn't inherently know where this information is, just like the mailman doesn't automatically know where Mary lives.

The answer is the data's XPath, which looks something like this:

```
//table//tr[56]/td[3]
```

This might seem confusing at first, but that's only because we're accustomed to reading physical addresses, designed for humans, and not digital addresses, designed for computers. But we can break it into parts and decipher it the same way.

Unlike a mailing address, an XPath is read from largest to smallest. This one says, "on this page, in the table, in table row 57, in table column 3."

If you find that cell in the human-targeted version of the FDIC website, you'll find your answer – the Mountain National Bank is in TN, or Tennessee (Figure 3.5).

So how can we use XPaths to get our hands on some data? The answer is to learn to construct your own. This can be easy or hard, depending on the complexity of your data and the source code.

Exercise 3

Later in this book, we'll explore some of the more complicated aspects of using source code, but for now, let's use this simple example to scrape the failed bank page.

In your Google Sheet, use the + sign at the bottom of the window to open a new tab within your Sheet. Rename it to "IMPORTXML scrape" or something similar. In the first cell (A1), type this formula:

```
=IMPORTXML("https://www.fdic.gov/resources/resolutions/bank-failures/failed-bank-list", "//table//tr[56]/td[3]")
```

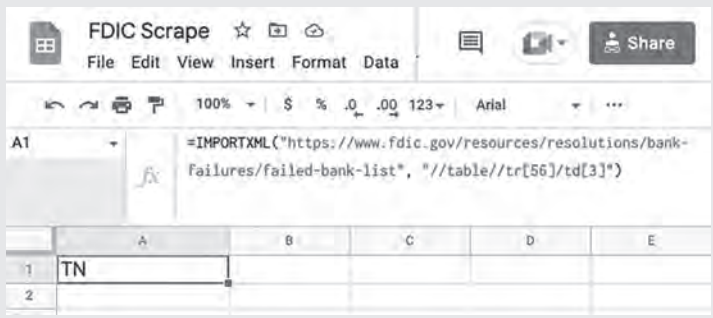


Figure 3.5 Scraping specific data with ImportXML.

If you hit enter, the sheet should populate with exactly one piece of data: The state where the bank is located. That's because the XPath we gave it, the second parameter in the formula above, points to this exact piece of data on a long page full of data. Again, if you get an error message, review the steps above or refer to the Troubleshooting section in Chapter 4.

Exercise 4

You can edit the XPath to show other bits of data, for instance, the state of the Mountain Heritage Bank. If you look at the web page, the Mountain Heritage Bank is the 167th row in the table, so you would change "tr[56]" to "tr[167]".

Creating XPaths

That leads us to the next question: How do you create your XPath or find the XPath of an existing piece of data? This chapter will explain two possible ways. Each has its own benefits and drawbacks.

One way is to read through the source code. You can view the source code of a web page by right-clicking on a white space and selecting "View Source."

You have already learned the basics of nested HTML elements. Simply string those together, with slashes in between, to narrow down the code until you get to just the data you want, like this.

```
//div[3]//ul[2]/li[1]/a[1]
```

Another way is to open the data you want in the Web Inspector. As we discussed earlier in this chapter, you can open an element in the Web Inspector by right-clicking on it and selecting "Inspect element." Once the Web Inspector panel has opened, you can right-click again on the element within the code panel.

It should give you a number of options including Copy. Hovering over Copy should give you the option "Copy XPath." You can copy and paste that XPath straight into your formula (remember to put it inside quotes).

But there is a drawback to this cheat code: Often, the XPath copied from the Web Inspector contains extra elements and bits of code that Google Sheets struggles to read. This is one of the many road bumps that exist when you try to work with source code across different programs.

The way over this hurdle is to simply paste in your copied XPath and troubleshoot from there. You can cut out HTML elements you're unfamiliar with, like "tbody." You can use the "View Source" tool to see the source code and determine which elements your data is nested in. You can use your critical thinking to figure out extraneous elements (e.g., does your XPath really need to say "div/div/div/div/div/"?).

Even with these road bumps, you can see how IMPORTXML is a much more flexible formula than IMPORTHTML. IMPORTHTML will only pull in a whole table (or a list), but XML can pull in just part of a table, the whole table itself or even more. Instead of one table cell, you could give the address for a whole column, or even a descriptor, like all the text that is bold.

Exercise 5

Let's try using IMPORTXML to scrape another specific kind of data – a whole category.

Earlier, we changed an XPath to the 22nd row by changing the number in the element "tr[]." We could instead collect all of the rows by deleting that limiter entirely.

//table/td[3]

Changing your formula to this XPath should result in the spreadsheet importing all of the rows of data, but only the fourth column.

Scraping with Code

Now we're going to experiment with a slightly more ambitious way to scrape: Writing our own code! Here is the simple code script we are going to run, using a Python library called Pandas.

```
import pandas
table    =    pandas.read _ html("https://www.fdic.gov/resources/
resolutions/bank-failures/failed-bank-list/")
output _ table = table[0]
output _ table.to _ csv("failed _ banks.csv")
```

Luckily, there are only four steps. Let's read through them:

1. "Import pandas" installs the Pandas Python library. Usually, when writing code, you'll need to install add-ons to make it work. In Python, these are called "libraries," but they go by different names in different languages.
2. The second line imports the HTML where our table is located. Once again, make sure you are using the exact URL and not just the website as a whole.
3. The third line identifies which table we want, similar to the third parameter in the IMPORTHTML function. This example is easy because there is only one table on that page. We use the number "0" because that's often how a computer counts – the first instance of something is considered number zero.
4. Lastly, we convert the HTML table into a CSV and give our file a name.

Exercise 6

Let's try it. Later in Chapter 9, we will install Python on our computers and create our own programs, but for now, we'll just use a Google Drive tool called Colaboratory.

Colaboratory is basically Google Docs for code. First, you'll need to enable it as an add-on for Google Drive. You can do this in Drive by clicking New > More > Connect more apps and search for "Colaboratory."

Once you have it installed, in your Google Drive, click New > More > Google Colaboratory. This will create a Colaboratory file, called a "notebook." Name your notebook "FDIC scrape."

Then, type each of the four steps into your notebook (Figure 3.6). You can create new lines by clicking "+ Code" at the top. Once the four lines are in the coding interface, click Runtime > Run all to execute it. A file will appear in the Files section, which looks like a small folder in the left-hand sidebar.

Figure 3.6 Google Colaboratory.

You have now obtained data through websites, spreadsheet formulas and writing your own code.

For more data journalism tips, tricks and exercises, visit the Data + Journalism blog at http://dataplusjournalism.com

Pro Tip: Legal Issues and Data Scraping

Much of the data that journalists scrape is public record, like government databases. But some datasets are proprietary and are protected by copyright law, including many in academic research.

Always check for usage rights, and reach out to the authors if you're unsure if you can use the dataset without paying a fee. Google Dataset Search is a good tool for this as you can filter out the paid websites in your search.

Scraping with a Browser Extension

Besides the scraping formulas, there are many other ways to scrape web pages. One of the easiest to use is Scraper, which is available for free on this Github page: http://mnmldave.github.io/scraper/

Once you install the extension in your browser, a small icon with a putty knife appears in the plug-ins area at the top of your browser. There's no need to click on it. The installation has turned your browser into a scraping machine.

It's important to note that this tool works differently than the formulas. It is a "one-time" scrape, similar to copy-pasting the data. It doesn't continually scrape the web page into the sheet, like the Google Sheets formula does.

Exercise 7 

Practice the Google Chrome scraper extension with these datasets. It's a snap. Just highlight the row you want to scrape, right-click on it (Control+click on a Mac) and the data appears in a dialog box. Just hit the "Copy to Clipboard" button, and paste it into your spreadsheet. It works with Google Sheets or Excel.

FDIC Failed US Banks List https://www.fdic.gov/resources/resolutions/bank-failures/failed-bank-list/

ESPN: 2019 Major League Baseball Attendance http://www.espn.com/mlb/attendance/_/year/2019

Scraping Documents

There's a running joke that if you want to tick off journalists, just email them a dataset in PDF format. PDFs are the scourge of the data journalism world as they provide valuable data but in a format that's useless. You cannot sort, filter or analyze the data in a PDF, so you must first extract it. The same applies to Microsoft Word, plain text files and other documents.

When requesting public documents, journalists should always ask for them in the electronic format they want to use, like an Excel sheet. But the records often arrive as a PDF.

Adobe Acrobat Pro allows users to convert PDFs into spreadsheets, Word docs or other usable formats. But many newsrooms don't have the software as it's very expensive.

There are many free tools that will easily scrape data from both "native" PDFs – such as Excel or Word documents saved in a PDF format – and "scanned" PDFs that have been scanned from old paper records.

Tabula is a free software you can download to your Mac or PC that scrapes tables out of PDFs. It was created by journalists and is wildly popular among investigative journalists because it's secure. Rather than loading the PDF to a live website like PDFtoExcel.com, they can scrape the file right on their desktop, which is much more secure.

Tabula doesn't work on scanned or image PDFs. But what it can do is identify and scrape tables throughout a whole document, even if it's many pages. Just download the software at Tabula.technology to get started (Figure 3.7).

Figure 3.7 Tabula scraping interface.

Exercise 8

Let's use Tabula to scrape a table out of a PDF with state voter registration statistics from the Nevada Secretary of State's office.

1. To get started, download the PDF to your computer from this folder: https://bit.ly/scrapepdfs
2. Now open Tabula from the applications folder on your computer. It should open as a tab in a web browser.

3. In the Tabula interface, select the blue "Browse" button, and select the Nevada PDF you downloaded. The name of the file will appear in the text field.
4. Hit the gray "Import" button next to the field to import the PDF.
5. Your PDF will appear in the interface. Click the gray "Autodetect Tables" button above the PDF. This will turn your table pink.
6. Now hit the green "Preview & Export Data" button in the upper right corner. This will convert your PDF into a table.
7. In the export format dropdown menu, select CSV – for a CSV file – and hit the export button.
8. Your spreadsheet will download to your computer. Then you can open the file in Excel or Google Sheets to clean and format the data.

Exercise 9

Let's repeat the steps from the previous scraping exercise; only this time we'll scrape the file on North Carolina COVID-19 outbreaks located in this folder: https://bit.ly/scrapepdfs. The North Carolina data is a much larger file and will take longer to scan and download. Repeat the steps in Exercise 8 to scrape and open it on your computer.

Exercise 10

Another method to scrape is to use simple websites. These tend to be very simple and are therefore easy to use but not as powerful or customizable as spreadsheet formulas or downloadable software.

PDFtoExcel.com is a free tool that lets you directly load a PDF into this browser-based tool for a free scrape. It will convert native PDFs as well as scanned PDFs by using optical character recognition (OCR) software built into it.

A word of caution: Be careful with scanned documents and OCR. Characters often get misread. A 3 can look like an 8 when converted, or a 7 may appear as a 2. After scraping, make sure to carefully review the spreadsheet to make sure the data is clean.

1. Go to this folder: https://bit.ly/scrapepdfs to download the scanned PDF from the Clinton Foundation's 2003 Form 990.
2. Open PDFtoExcel.com and hit the red upload button.

3. Select your PDF, and it will upload. You'll need to wait up to a minute as it converts, even for a smaller file.
4. Once it's converted, a red button titled "Free Download" appears. Hit that button, and your spreadsheet will download to your computer. You can clean and format it in Excel or Google Sheets.

The free version of PDF to Excel works fine for smaller documents such as the one you just scraped. It offers a $5 monthly version that reduces wait time and allows you to process a larger batch of files.

Another drawback to PDFtoExcel is security. You're loading a document to the live web, which means it's vulnerable. If you need to scrape a sensitive document, use Tabula.

* * *

Video: Watch how to scrape PDFs with Tabula
https://bit.ly/videoscrape

Exercise 11

Scrape Data from Google Finance

Business and tech reporters keep a close eye on the stock exchange. Google Finance (https://www.google.com/finance/) lets you not only build a free portfolio of stocks to track, but you can also use formulas in Google Sheets to scrape real-time and historical stock prices and other data.

To get started, open our practice spreadsheet (shortlink: https://bit.ly/scrapefinance), and make a copy of it by going to File > Make a Copy.

Note that there are two tabs at the bottom of the sheet. One for our first two exercises and a third for the single stock focus exercise.

These exercises will walk you through how to set up your own spreadsheet and write specific formulas. You also can consult this list of Google Finance scraping formulas: https://bit.ly/scrapeformulas

Scraping Real-Time Stock Data into a Google Sheet

1. Open sheets.new in a browser window.

2. In Row 1, create a header. Type "symbol" in cell A1, "price" in B1, "pe" in C1 and "price52" in D1. The phrase "pe" means price earnings ratio, and "price52" is the 52-week high price.
3. In Column 1 under "symbol," type some stock symbols in rows 2–4:
   goog
   vz
   nke
   f

   Those are the symbols for Google, Verizon, Nike and Ford. Those symbols and cells are what the Sheet will use to pull the correct data from Google Finance.
4. Now you're ready to scrape data into each of the cells:
   In cell B2, type: =googlefinance(A2,B1) and hit return for the price
   In cell C2, type: =googlefinance(A2,C1) and hit return for the price earnings ratio
   In cell D2, type: =googlefinance(A2,D1) and hit return for the 52-week high price
   Repeat these steps for the other cells, changing A2 to A3, A4, A5, etc.
   These functions will auto-update over time so you'll see current prices.

How to Scrape Historical Stock Prices

1. In the same spreadsheet, type: =GOOGLEFINANCE(a4, "price","12/09/2020", today(), "daily") in cell E1 to get daily ending prices since Dec. 9, 2020 (or adjust date/price category as you see fit).
2. Type: =GOOGLEFINANCE(a3, "price","12/09/2020", today(), "weekly") in cell G1 to get end-of-week prices since Dec. 9, 2020 (or adjust date/price category as you see fit).
3. These functions will auto-update over time so you'll see current prices.

Focusing a Sheet onto One Stock

1. Open a Google Sheet by typing sheets.new into the browser field.
2. Set up a sheet to look like Figure 3.8, and then in cell B2 under Value, type this formula and hit return:

   =GOOGLEFINANCE("AAPL", A2)

   Then grab the blue square in the lower right of the cell and drag down: It will populate the sheet with data from Apple stock, and it'll update as the market changes.

Formulas: To get help on any of the formulas, click on the blue button on the cell when you type in =GOOGLEFINANCE. You also can see them when you hit return after typing in Googlefinance, and you can find many of them listed here: https://support.google.com/docs/answer/3093281?hl=en

	A	B
1	Field	Value
2	Name	
3	Price	
4	Low	
5	High	
6	Change	
7	Changepct	
8	Low52	
9	High52	
10	Volume	
11	Volumeavg	
12	Marketcap	
13	PE	
14	EPS	
15	Shares	
16	Beta	

Figure 3.8 Google Finance stock scraping spreadsheet.

Training Video: How to Scrape Google Finance. Review how to do these exercises in this short video: https://bit.ly/gfinancevideo

Exercise 12

Scraping Comments from Social Media

We will explore social media scraping with code in Chapter 9, but a quick and easy way to extract comments, tweets and more from social media posts is to use a browser-based tool called ExportComments.com. You can scrape up to 100 comments on a post for free, and there are pricing options for larger data pulls. Prices range from $11 for three

days of use up to $200 for a monthly business plan for larger scrapes. Only the paid accounts require a sign-in.

The tool works for Facebook, Twitter, Instagram, YouTube, TikTok and many other social channels. It's easy to use. Just take the link from the top level of the social media post, paste it into the field on the interface and hit Start Export Process. After a minute or two, you'll get an interface for you to download the scrape and export in several formats, including Excel and CSV files. Once the comments are in a spreadsheet, it's easier to sort and filter and look for trends. You also can do a search for specific keywords to see how often they appear in the comments.

Security is always a concern when pulling data from the web. Although the posts are public, you may not want a record of your scrape on the Export Comments site. It offers a "private export" button underneath the list of social channels to increase privacy.

Let's put the tool to the test by pulling the 100 latest comments from a post on the National Rifle Association's official Twitter account. Type this URL into the ExportComments.com search field (or search for another Twitter post and try that; Figure 3.9):

https://twitter.com/NRA/status/1490059049316065281

Figure 3.9 ExportComments.com Interface.

Once the URL is in the field, you get several options that let you narrow the export to just followers or those followed by the account you selected. Or you can check the box asking for nested comments.

Hit the Start Export Process button, and wait a couple of minutes for your download interface to appear (Figure 3.10):

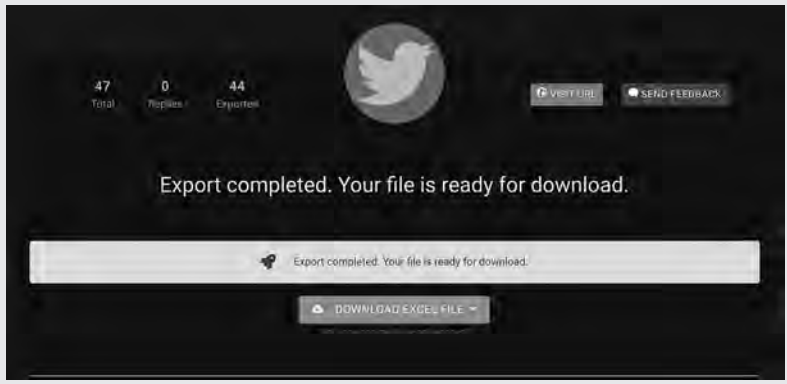


Figure 3.10 Download interface on ExportComments.com.

Now just hit the green download button, and the file will download. The pull-down menu gives you several export file format options.

Besides extracting comments, the tool also helps you pull hashtagged comments and other data from the social media accounts. Export Comments isn't perfect, but it can be very handy for quick analysis on social media reactions to controversial breaking news stories, such as the Charlottesville protests, George Floyd protests, mask mandates and others.

Resources for Getting Data

Freedom of Information (FOI) Resources

Besides iFOIA.org, there are many other free tools and guides available to help you find public records:

SPJ FOIA Guide for Professional Journalists https://www.spj.org/foi-guide-pros.asp

SPJ FOI Toolkit https://www.spj.org/foitoolkit.asp

Student Press Law Center Public Records Letter Generator https://splc.org/lettergenerator/

Journalist's Toolbox Public Records and FOI Resources https://www.journalistst oolbox.org/category/public-records/

MuckRock FOIA 101 Guides https://www.muckrock.com/project/foia-101-tips-and-tricks-to-make-you-a-transparency-master-234/

* * *

Data Scraping Tools

Here are some other data scraping tools to help with deadline reporting. Most of them are free. Find more data scraping tools in Journalist's Toolbox: https://www.journaliststoolbox.org/category/data-scraping/

Import.io https://www.import.io/

If your data is behind a login, inside an image, or if you need to interact with a website, this paid tool has you covered. Once you enter a web page, you simply point and click on the items of interest, and Import.io will learn to extract them into your dataset. Once extractors are fully trained, they can be set to run on a schedule over multiple different web pages, creating large datasets ready for transformation, analysis and integration into your applications and internal systems.

Scrapy https://scrapy.org/

The "Scraping with Code" section of this chapter mentioned Python libraries – essentially add-ons to the language of Python itself. Scrapy is a fast, open-source and collaborative library for extracting data from websites.

BeautifulSoup https://pypi.org/project/beautifulsoup4/

BeautifulSoup is another Python library for pulling data out of HTML and XML files. It's one of the most popular tools in Python for scraping pages on the web.

Outwit Hub https://www.outwit.com/

Outwit is a desktop app that can identify HTML elements on a web page and scrape them. The free version lets you download 100 rows at a time, and it can handle pagination – finding and scraping rows of data on multiple pages.

ParseHub https://www.parsehub.com/

A desktop app that can identify and scrape elements and sub-elements. The free version lets you scrape 200 pages at a time.

CometDocs https://www.cometdocs.com/

This tool not only converts PDFs into Excel files but also provides the ability to host documents. It is one of the most consistently accurate conversion tools outside of Adobe Acrobat, which is a paid tool.

Online OCR https://www.onlineocr.net/

Good for scraping smaller files.

XML Grid https://xmlgrid.net/xml2text.html

Have data buried in an XML file? You can use this CSV file scraper to extract it.

Google Keep https://keep.google.com/

Google's note-taking app lets you export text out of an image. Just click on "Grab Image Text" and pop it into the text of your note. Google Pinpoint (https://journaliststudio.google.com/pinpoint/about) and MacOS Monterey can also do this with text in images. iOS phone software also can extract small amounts of text from an image by clicking on the yellow lines around the text.

Tools Used in This Chapter

Google Sheets http://sheets.google.com
Google Dataset Search https://datasetsearch.research.google.com/
Google Chrome Data Scraper Plug-in http://mnmldave.github.io/scraper/
Tabula.technology https://tabula.technology/
PDF to Excel https://www.pdftoexcel.com/
Folder to download PDFs https://drive.google.com/drive/folders/17WF-GhDbbawCfbU-RsEKGHMuTWs7PpTc

* * *

Footnotes

Washington DC COVID-19 Testing Sites https://coronavirus.dc.gov/testing
Our World in Data: COVID-19 Vaccinations by Country https://ourworldindata.org/covid-vaccinations#source-information-country-by-country
Pew Research Center: Religious Composition by Country, 2010-2050 https://www.pewforum.org/2015/04/02/religious-projection-table/
Ireland Trolley Ward Watch https://www.inmo.ie/Trolley_Ward_Watch
Association of Food and Drug Officials Directory https://www.afdo.org/directories/dslo/results/?q=Georgia&unifyfda=1&bystate=1&selected_facets=area_exact:%22100%22
FDIC Failed Banks List https://www.fdic.gov/bank/individual/failed/banklist.html
National Interagency Wildfire Center Wildfires Data https://www.nifc.gov/fireInfo/fireInfo_stats_totalFires.htm

4 Cleaning Data

Samantha Sunne

In 2016, the movie *Spotlight* won the Academy Award for Best Picture, the highest award in the American film industry. The movie also won praise from journalists for its accurate portrayal of the painstaking process in the *Boston Globe*'s investigation of the Catholic Church child abuse scandal. The *Globe*'s investigative team uncovered evidence by interviewing stakeholders, reviewing public records and compiling data from printed church directories.

In one scene, the team huddles around a phone, asking a researcher for his opinion on its findings. "Does that sound right to you?" the editor asks. "In terms of scale?"

This is an example of data fact-checking, or as data journalists would have it, "bulletproofing." It's one of the most vital steps to a valid and ethical data analysis for journalism and often includes a subset of data work called "cleaning."

In *Spotlight*, the expert's advice bolsters the team's findings and also provides them with new clues. Data journalists recommend doing these kinds of "data integrity checks" during and after the analysis itself.

Data from government databases, papers and websites is often "dirty," meaning it is unorganized, incorrect, incomplete or contains other flaws that prevent accurate analysis. This chapter addresses tactics for finding those flaws, fixing them and making sure your analysis is ready to produce the best journalism possible.

Many errors will come from your source data, meaning you need to clean it after scraping or importing it. It's most prudent (and in some cases unavoidable) to clean your data at this point, to avoid errors and headaches later.

At the end of this chapter, you will find a list of common errors in spreadsheets and code and likely solutions for them.

How to Find Errors

As with other aspects of data journalism, sometimes the most important part of your task is the old-fashioned reporting. Data journalists almost always recommend bulletproofing data analysis with a series of interviews, cross-references and "gut checks."

Pulitzer Prize–winning data journalist Jaimi Dowdell said that exciting findings, especially, need to be fact-checked. "If you get an amazing analysis, look

DOI: 10.4324/9781003273301-5

first to see if you've made a mistake," she advised as a trainer for Investigative Reporters and Editors.

Sometimes, data cleaning can be quite simple. For instance, with the table of failed banks we scraped in Chapter 3, the header row contains some doubled-up terms, like "CityCity". If there is only a cell where there is a problem, it's easiest to simply click on the cell and type the correct value manually, that is, "City".

A deep dive, like the kind Dowdell specializes in, can involve a long list of checking and cleaning tasks. The Center for Public Integrity, a pioneering news nonprofit, shared the process for its story "Power Trips." In order to investigate tens of thousands of financial documents on congressional staffers, the team:

1. Scanned paper reports and converted them to PDFs,
2. Scraped the data from PDFs and imported it to a computer program,
3. Checked the resulting 30,000 rows of data against source documents,
4. Standardized alternate spellings and misspellings of names,
5. Cross-referenced reports with other documents, like calendars,
6. Identified and flagged missing information,
7. Filtered the data for a relevant time span, and
8. Interviewed congressional staffers on potential ethics violations.

And this was all before doing the data analysis itself.

As you go through your importing and analysis, keep these overarching questions in mind. You don't necessarily need to do each of these checks for every dataset you ever use, but use them to identify the types of problems likely to occur.

Save the original version. In a completely separate file, store the dataset exactly as you obtained it from the source. Give this file a name like "raw dataset," and do your analysis on a copy of it.

Background your data. Anytime you obtain data from an outside source, you should background it the same way you would an interview subject. Who or what is the source? Is it a primary source, or did they collect it from somewhere else? Could they have a bias or a blind spot that would skew your analysis?

Find the methodology. How was the data collected? Was it aggregated from other sources, like the Federal Bureau of Investigation's (FBI) Crime Data Explorer? Was it typed in by hand or scanned from a document? What errors could have occurred in the collection or the aggregation?

Keep a data diary. Create a detailed list of all the actions you took when working with the data, including importing, cleaning and filtering. Include the full text of formulas and queries. This can also take the form of saved query files in SQL and documentation in code. We will explore data diaries more in Chapter 11.

Double-check. Does the row count in your spreadsheet match the one in the source data? Do a "spot check" – a random sampling of data points to check

against the source. If you find problems, you may want to try importing again with a different program.

Run tests. Use formulas like SUM() and COUNTA() to check for problems. Make quick data visualizations, or use the MAX() and MIN() formulas to look for outliers. Do your double-checking in a completely separate program, like a calculator, rather than the same spreadsheet.

Cross-reference. Has anyone else published stories, reports or analyses of the same or similar data? What did they find? Are your findings similar? Can you find the same data from a different source?

Don't allow for confirmation bias. Are you interpreting the data in a way that supports your story, or are you being truly neutral? Can someone else in the newsroom, who is unfamiliar with the source or the story, proofread it? If the data doesn't support your theory – or outright disproves it – be honest with your readers.

Keep your critical thinking hat on. Does this seem like the correct number of records for such a dataset? For a poll or a scientific paper, how big is the sample? Does the finding fall within the margin of error? (See Chapter 12 for more tips on using polling data.)

Find the data dictionary. A data dictionary, or record layout, is a guide to the field names in a dataset. They are commonly found in very large datasets, like FBI data. This helps especially with field names that are vague or jargony.

Consult human experts. Sometimes, this is the most important part of your data cleaning to-do list. You should always check your assumptions and confirm your understanding with the source of the data. Expert sources can also provide insight or confirm your findings. For an especially large or novel analysis, you may want to check your entire methodology. Ask whether there is anything you're missing, or anything else you should look into.

If major problems arise, and you think you can't clean or vet the data enough to write the story ethically, it's best not to use it at all.

Confounding Variables

A "confounding variable" isn't a variable that's especially confusing – it's a variable that might affect your findings but isn't visible in the dataset. It's also called a "lurking variable."

Consider a 2018 CBS story about new Starbucks cafes making home values rise. Are home values actually going up because it's easier to get a cup of coffee? Or are there other factors at play?

Remember to keep these possibilities in mind when implying any correlation – or stricter yet, causation – between variables. You can read more on correlation and causation in Chapter 6.

Finding Empty Values

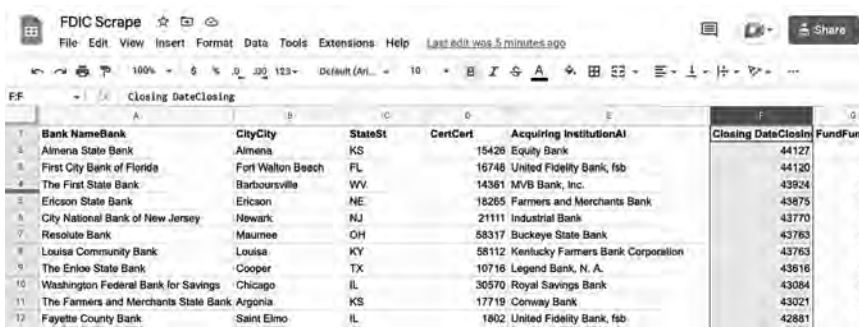
As we will learn in Chapter 5, Basic Spreadsheets, a single empty cell can bring down an entire data analysis, making it one of the first errors a data journalist looks for. Blank rows can truncate filters, skew averages or trigger error messages. Some programs will auto-populate empty cells with the value "0," which has another meaning entirely.

In some contexts, these are called "blank fields" or "null values." There are several methods for finding these. In a spreadsheet program, you can use formulas like ISBLANK() or IFERROR(). Other programs, like SQL Server, use functions like ISNULL(). Microsoft Excel has a special copy-down command to fill in missing values that are missing but implied. In Google Sheets, you can jump to the last populated row using the keyboard shortcut Ctrl+Down/Cmd+Down.

Another option is Conditional Formatting – formatting cells based on the value inside them. This tool is controlled via point-and-click buttons in the formatting menu and can help you find null values as well as outliers and patterns. For a small enough dataset, scrolling through is always an option!

Macros

In Chapter 3, Scraping Data, we already accomplished some cleaning when we scraped a table of failed banks. When we copy-pasted the scraped values into a new tab, the dates were converted into a five-digit number that looks something like this (Figure 4.1):



	Bank NameBank	CityCity	StateSt	CertCert	Acquiring InstitutionAI	Closing DateClosin	FundFund
2	Almena State Bank	Almena	KS	15426	Equity Bank	44127	10
3	First City Bank of Florida	Fort Walton Beach	FL	16748	United Fidelity Bank, fsb	44120	10
4	The First State Bank	Barboursville	WV	14361	MVB Bank, Inc.	43924	10
5	Ericson State Bank	Ericson	NE	18265	Farmers and Merchants Bank	43875	10
6	City National Bank of New Jersey	Newark	NJ	21111	Industrial Bank	43770	10
7	Resolute Bank	Maumee	OH	58317	Buckeye State Bank	43763	10
8	Louisa Community Bank	Louisa	KY	58112	Kentucky Farmers Bank Corporation	43763	10
9	The Enloe State Bank	Cooper	TX	10716	Legend Bank, N. A.	43616	10
10	Washington Federal Bank for Savings	Chicago	IL	30570	Royal Savings Bank	43084	10
11	The Farmers and Merchants State Bank	Argonia	KS	17719	Conway Bank	43021	10
12	Fayette County Bank	Saint Elmo	IL	1802	United Fidelity Bank, fsb	42881	10

Figure 4.1 The date serial number format.

These are sometimes called "date serial numbers," and they are a valid format for a computer to analyze. It's not so readable for humans, though, so we used the formatting toolbar in Google Sheets to convert it back to a "Month/Date/Year" format. This is an example of data cleaning.

Now, we will automate that formatting using a tool called a "macro." A macro is a recorded action that a spreadsheet program can then run automatically.

Exercise 1

We will try creating a macro in our table of failed banks we scraped in Chapter 3.

1. Open the table you created in Chapter 3, and navigate to the tab that contains only values. (Earlier, we copy-pasted these values so that they don't change – Google Sheets executes the scraping function every day, so the first tab is liable to change.)
2. Highlight cell F2 and click Extensions > Macros > Record macro. This will pop open a box with a blinking red light indicating that Google Sheets is recording our actions. Check the radio button for "Use relative references."
3. Still highlighting cell F2, click Format > Number > Date, or a specific date format like "MM-DD-YYYY". Click the green Save button. Give your macro a name, like "Format Date".
4. Highlight the rest of Column F and click Extensions > Macros > Format Date. This will format the rest of the cells according to the action we performed in cell F2.

Macros are just one of many ways to clean, rearrange or reformat data. Different methods will work better depending on your data and the problems you are facing (Figure 4.2).

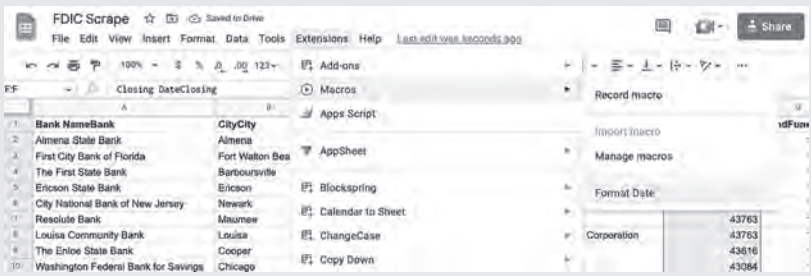


Figure 4.2 The Macro menu in Google Sheets.

Regular Expressions

We will continue experimenting with cleaning by using a technique called Regular Expressions. Regular Expressions, or RegEx, is a syntax that can be used to identify patterns of text and numbers in spreadsheets. It also works in programming languages like Python and R.

Table 4.1 Useful RegEx Phrases

\w	A lowercase "w" in RegEx stands for "word character," that is, characters that are not punctuation. Underscores (_) are counted as word characters, but other punctuation marks are not. The "w" command can help with removing extraneous punctuation marks, like "[" and "<", that sometimes come with data scraped from the web. It is helpful because it can return both numbers and letters.
\d	A lowercase "d" stands for "digit" and can be useful when extracting only numbers from a line of text. For example, extracting street numbers from a list of addresses.
\	A backslash (\) indicates that you are using RegEx to find matching patterns, not search for the text itself. Using just "w" in the RegExExtract() formula would return only the letter "w", but "\w" would return any alphanumeric character. If listed before a punctuation mark, a backslash functions as an "escape." This means adding a backslash before a character like a period ("\.") indicates you want to look for an actual period, not use it as a wildcard (see below).
.	A period (.) is a wildcard, meaning it will match any character. This can be useful if you don't know which characters are going to occur in between the ones you want, such as phone numbers that include parentheses and ones that don't.
[1–5]	Instead of the letter "d", you can specify a list of numbers you are looking for, like 1 through 5. The phrase "[0–9]" would return any numeric character.
[a–z]	The same is true for letters. "[c–t]" will return the first character that matches a letter between those letters, or you can use "[a–z]" to mean any letter in the alphabet. RegEx is case sensitive, so "[a–z]" will return lowercase letters, while "[A–Z]" will return uppercase.
+	A plus sign (+) indicates you want to extract all characters identified in your command. Searching for "\w" would return the first alphanumeric character, but "\w+" returns the first as well as all of the ones that come after it.

Just like with programming and functions in general, it's best not to try to memorize all of the syntax in Regular Expressions (Table 4.1). Instead, use support sites and instructions like the one in Google Sheets to guide your RegEx creations.

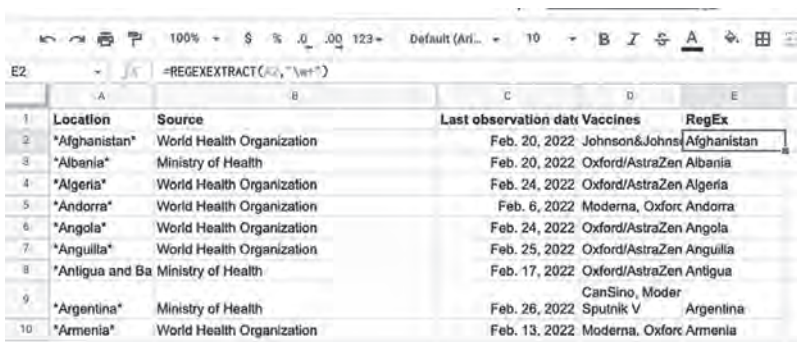
* * *

Exercise 2

For this method, we will use a dataset of Coronavirus vaccines available in 2021 (shortlink: https://bit.ly/vaccinetable). The data comes from Our World in Data, a website run by a research team at Oxford University.

Because this data was also scraped from the web, the sheet contains some extraneous characters and mismatched data formats. First, we will remove the asterisks in Column A using RegEx.

1. Make a copy of the vaccine table (shortlink: https://bit.ly/vaccinetable).
2. Give Column E the name "RegEx".
3. In cell E2, type the following formula and press enter.
   =REGEXEXTRACT(A2, "\w+")
4. Copy the formula down to apply to the rest of your data.



	A	B	C	D	E
1	Location	Source	Last observation date	Vaccines	RegEx
2	*Afghanistan*	World Health Organization	Feb. 20, 2022	Johnson&Johns	Afghanistan
3	*Albania*	Ministry of Health	Feb. 20, 2022	Oxford/AstraZen	Albania
4	*Algeria*	World Health Organization	Feb. 24, 2022	Oxford/AstraZen	Algeria
5	*Andorra*	World Health Organization	Feb. 6, 2022	Moderna, Oxfor	Andorra
6	*Angola*	World Health Organization	Feb. 24, 2022	Oxford/AstraZen	Angola
7	*Anguilla*	World Health Organization	Feb. 25, 2022	Oxford/AstraZen	Anguilla
8	*Antigua and Ba	Ministry of Health	Feb. 17, 2022	Oxford/AstraZen	Antigua
9	*Argentina*	Ministry of Health	Feb. 26, 2022	CanSino, Moder Sputnik V	Argentina
10	*Armenia*	World Health Organization	Feb. 13, 2022	Moderna, Oxfor	Armenia

Figure 4.3 The REGEXEXTRACT() formula.

Pro Tip

Keyboard shortcuts are just one way to "copy down" your formula to apply to other rows. Another method is to hover over the bottom-right corner of the function cell until the cursor becomes a thin black plus sign. (The cursor's appearance may vary by program.)

Click and drag the plus sign down to the bottom row of your data. For spreadsheets containing many rows, this is not efficient, but it can be convenient if you have only a few rows.

This is a very simple Regular Expressions syntax that returns only alphanumeric characters. The formula REGEXEXTRACT() in Google Sheets returns characters that match the queries you are writing in RegEx (Figure 4.3). We will learn more about functions in Chapter 5.

Spreadsheet Formulas

REGEXEXTRACT() is just one of the many, many formulas in Google Sheets that can help you with your data cleaning. We're going to experiment with another one called SPLIT() that separates parts of a cell's value based on a delimiter. A "delimiter" is the character that defines where the columns begin and end.

In the case of our vaccine table, we can see that Our World in Data used commas to separate the list of available vaccines in each country. So our SPLIT() formula should look something like this:

=SPLIT(D2, ", ")

The SPLIT() formula has two parameters: the cell to split and the delimiter. If you recall from Chapter 3, a "parameter" is a variable inside a formula that tells a formula what to do. You need to wrap the comma in quotes so that the program knows it is a piece of text to search for (Figure 4.4).

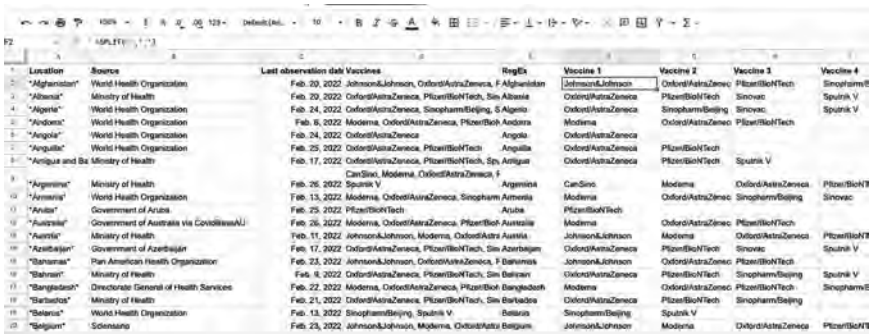


Figure 4.4 The SPLIT() formula.

Exercise 3

1. Give Column F the name "Vaccine 1".
2. In cell F2, type the following formula and press enter.
   =SPLIT(D2, ", ")
3. This should populate Row 2 of Column F, and additional columns, with the vaccines listed in cell D2.
4. Rename Columns G, H and I with the titles "Vaccine 2", "Vaccine 3" and "Vaccine 4".
5. Copy your formula down to the rest of the rows in Column F.

The SPLIT() formula creates a new column for every vaccine in the list.

Pro Tip

Most Google Sheets formulas only populate the columns they are located in. SPLIT() populates additional columns to the right, which can be very powerful, but also dangerous, as it could overwrite data located to the right. You could also run into an error message refusing to overwrite existing data.

If your cell contains more than four or five values to split up, it's best to use a different cleaning method. Using SPLIT() to populate dozens of columns will make your spreadsheet harder to use, not easier. At that point, a journalist would likely split values into rows, not columns.

Make sure you understand which delimiter the source data is using, because that can have enormous effects on the integrity of your data.

For instance, we can also see Our World in Data used slashes (/) to indicate multiple names for the same vaccine. In other datasets, the slash could be the delimiter. Pipes (|), semicolons (;) and tabs are other common delimiters you may come across.

Text Editors

Many programs have Find and Replace functions, but with very large datasets, it can be easier to use a text editor like Microsoft Word, Sublime Text, TextEdit or many other free programs.

They can be an especially good option for finding small bits of text that occur very often in your dataset. This could be double spaces that should be single or semicolons that should be commas.

Text editors usually navigate data by "strings" – bits of grouped text. Because strings can contain numbers, letters, punctuation and spaces, all of which are counted as characters, try to think in terms of "strings" rather than words, phrases or other terms that we would use verbally.

For example, "AstraZeneca " and "AstraZeneca" are both text strings, but the first is one character longer. Unlike humans, many text editors and data programs will consider them to be two separate values.

In addition, many text editors (and some spreadsheet programs) use different font colors to help the user distinguish bits of text and code. For example, in Google Sheets, formula names are shown in green and parameter values in black.

OpenRefine

As a reporter, you will often come across columns of names, dates, places and people. This creates many possibilities for misspellings, alternate spellings, abbreviations and acronyms.

As we have learned, a data analysis program will usually read "Barack Obama", "B. Obama" and "President Barack Obama" as completely different entities, even though we know they are the same person.

In addition, human-entered or human-edited data is especially prone to incorrect or inconsistent entries. Christopher Groskopf, the author of the "Quartz Guide to Bad Data," once encountered a dataset with more than 250 different spellings of the word "Chihuahua."

Programs like OpenRefine are designed to combat this very problem. OpenRefine uses a collection of linguistic analysis techniques to identify and "cluster" rows containing similar values or ones that OpenRefine thinks are related, like "Central Intelligence Agency" and "CIA." You can then batch edit these values or rename them.

Let's try this out by cleaning this dataset of people who donated to Obama's run for the US House of Representatives in 2000 (shortlink: https://bit.ly/obama2000).

When opening this dataset, you will find that it is a comma-separated values (CSV) file containing 571 rows. It contains the names and addresses of donors, as well as how much they donated, and some additional data recorded by the Federal Election Commission (FEC).

Exercise 4

1. Download the Obama donors dataset to your computer (shortlink: https://bit.ly/obama2000).
2. Download the free OpenRefine desktop app from OpenRefine.org. Launch the app, which will open as a new tab in your web browser.
3. Import the obama2000.csv dataset by uploading it from your computer and clicking Next. In the import wizard, make sure the comma is selected as the delimiter. To finish importing, click "Create Project."
5. Scrolling over to the "contributor_city" column, it appears that "Chicago" is misspelled in one of the first rows.
6. We will create a "text facet" to see if any other cities are misspelled, and rename them. Open the drop-down menu in the "contributor_city" column, and choose Facet > Text facet. This will create a pop-up panel showing the various cities and the number of rows they occur in (Figure 4.5).
7. Click "Cluster" in the panel to open a new pop-up window where you can batch edit the city names. In the top-left corner, change "key collision" to "nearest neighbor."
8. Type "Chicago" into the boxes for New Cell Value.

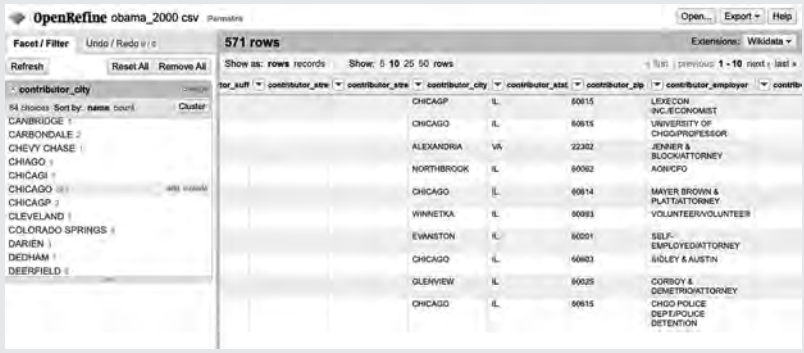


Figure 4.5 OpenRefine text facet.

9. In the "Merge?" column, check the boxes for the five spellings of Chicago, and click Merge Selected & Close. The cities should now be renamed to consistent spellings (Figure 4.6).

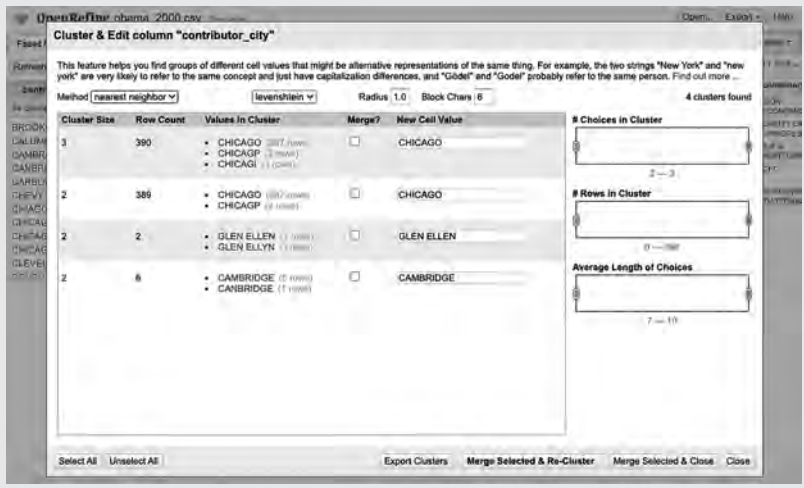


Figure 4.6 The OpenRefine cluster window.

There are also multiple spellings of Glen Ellyn and Cambridge. With these, you should do some external research to figure out which is the correct spelling.

This is just one tool that is possible with OpenRefine, and one small step in verifying and cleaning this dataset, but batch editing values in this way can save you a lot of time.

Troubleshooting Common Spreadsheet Errors

Below is a list of common errors and popular solutions for them. Remember there are many potential solutions, varying by program and project aim, and you may need to do some digging of your own.

Leading Zeros	Leading zeros are common in text values that are read as numerals. For instance, the ZIP code for Jersey City, New Jersey, is "07097." Some programs would read that as the number 7,097, which is incorrect.
	You can avoid this problem by designating a text format while importing or wrapping the values in quotes to indicate they are text strings.
Spaces	On the flip side, some imported values will contain extra characters you don't want – like trailing spaces at the end of a text string. Remember that a computer counts every character in a text string, meaning they can affect your functions and outputs.
	One way to remove leading or trailing spaces is the TRIM() function. However, this will not delete white space characters in the middle of text, like a double space.
Duplicate Rows	Some datasets will have duplicate rows – specifically, rows that are identical in every column. It can take some digging to find out why the duplicate exists. It's possible that it is a valid record that should be included in the analysis, or perhaps it should be removed.
	Duplicates can be hidden with filters or removed with tools like OpenRefine. Google Sheets also has a data cleaning feature that removes duplicate rows based on matching columns.
Incorrect Formats	As we discussed in Chapter 3, importing data can be one of the toughest parts of a data project, as the process can be prone to errors. For example, "5556934563" could be a phone number, a unique identifier or an amount over 5 billion. Each of these data types will be interpreted differently in analysis.
	Many spreadsheet and database programs will have import processes that let you specify the data format before completing the import. You can also change the format after the fact using the formatting toolbar in a spreadsheet program or queries in a tool like SQL.
Incorrect Range	Double-check the ranges in your formula bar, conditional formatting and filters to make sure they contain all the data you need. If the filter doesn't contain the whole dataset, rows will be separated from their data in other columns.
	This is another reason empty rows or cells can trip up your analysis – often programs will stop a filter or a formula when it reaches the first empty row.
	In Google Sheets, cells are in a filter when they have a thin green line around them. When in doubt, you should always be able to open a pop-up interface to control your filter and specify the range there.
	To select all the data in your spreadsheet, click on a populated cell and select all (Ctrl+A or Cmd+A). To highlight every single cell in the sheet, not just the ones that contain data, click on an empty cell and select all from there.

(*Continued*)

Delimiters	A CSV is a "comma-separated values" file, meaning the data columns are separated by a comma. Another common format is TSV, for "tab-separated values." But delimiters – also called "separators" – can technically be any character. Additionally, CSVs can contain data where a comma is *not* a delimiter – like in the value "4,392." Most programs will be adept at interpreting these values, but occasionally, you will see an unusual separator, like a pipe (|). Watch out for these options in import wizards. If the delimiter also occurs within values, you can format that column in a formatting toolbar or wrap it in quotes.
Rounded Digits	In a spreadsheet program, you can often make decimal points visible or invisible using the formatting bar. This doesn't affect the value itself – just how it is presented to you. In a few cases, it may be that one program actually *converted* a value into one with fewer digits – that is, rounded it up or down. For example, Facebook shows the number of comments on a post as "3.4K" rather than 3,454. This can affect your final analysis, so it's best to watch for these rounded numbers while still in the acquisition phase.
International Punctuation	This textbook uses American English for spreadsheet formulas and other inputs. Some keyboards and programs will use different syntax according to their native language. For example, in Germany, Google Sheets uses a semicolon (;) rather than a colon (,) to separate parameters. If you use a computer or a keyboard that is not English, consider looking up the specific syntax and punctuation for your language and region.

Troubleshooting Common Programming Errors

Error messages are unfortunately common in both spreadsheets and programming, as we will discover in Chapter 9. Many programs will helpfully list the line number where your error occurs, which can at least help you narrow down the potential culprits.

Researching your own solutions to programming problems is going to be vital. Search engines like Google and forums like StackOverflow will be an essential resource. In her book *Mining Social Media: Finding Stories in Internet Data*, data journalist Lam Thuy Vo claims that researching errors is an integral part of becoming a programmer.

"One day we might have to analyze text-based reactions, and another day we might be looking at thousands of images," she wrote. "To be a good coder, in other words, means that you have to be a *resourceful* coder, one who knows how to look for and ask for help solving any problems you encounter."

This is a list of some of the most common road bumps, which often come down to a simple letter or punctuation mark. You will be surprised at how often the most intimidating of errors are solved by the simplest of solutions!

Missing or Incorrect Characters	Perhaps the most common error of all is a wrong or missing character. For instance, an XPath might use square brackets ([]) to offset an identifying class, not parentheses, as one might assume.
	Read carefully through both your error message (if you have one) and your code to see if you can find the missing piece. If not, a search engine is the easiest way to narrow it down.
Language Variation	Each programming language (and other tools like SQL and RegEx) uses its own syntax. The syntax is the series of words, punctuation and "grammar" that rules the language.
	For example, RegEx is case sensitive, meaning it considers "A" and "a" to be two different values, but others wouldn't. Some programs allow spaces between parameters in a function, and others don't.
	Overall, it is best to be consistent. If you use capitalization for a value in the beginning, continue using that format throughout.
Copy and Paste	We don't recommend copy-pasting formulas from a website or an ebook. Some programs fail to correctly interpret fonts and characters pasted from other programs. Quotes are the usual culprit.
	The best practice is to always type in formulas and values by hand.
Quotes	A common road bump is that some programs use single quotes (') and others use double ("). Some use both.
	If any of your functions are not running, an easy first troubleshooting step is to delete all the quotes and retype them in manually. If that doesn't work, manually replace single quotes with double quotes, or vice versa.
Parentheses	Parentheses and other characters that are used to offset parts of code, like brackets ({}), can get confusing, especially when working with nested functions.
	Carefully check each parenthesis to see if it is paired with a beginning or an ending parenthesis. If you have too many on one side, the function will not work.
	Programs like Google Sheets and text editors like Sublime Text will also help by highlighting matching parentheses or making them a different color.
Changed Source Code	When your script is communicating with another program, like a scraper or an API, it is dependent on that other code. For example, if a website uses the tag <h1> to enhance text, but later changes it to <h2>, a scraper will suddenly find no values matching the <h1> tag.
	If your scraper stops working entirely, you may need to look at the source to see if anything has changed.
Conflicting Language Versions	In Chapter 9, we will talk about downloading, installing and updating versions of programming languages. Because they need to be downloaded and installed, your languages could be out of date, or you could need a certain version in order to use a certain library.
	Your error message will likely show something along these lines. If the message refers to a version, it's best to search for that message in a search engine, whether or not you think you have that version installed. The version you have or want could be in a different file path or virtual environment.

(*Continued*)

Install Failures	Installing languages can be tough, but libraries even tougher. Libraries that were made by individuals, rather than published by an official company, might require some extra steps. Use the library's website, Github page or search engine results to find the extra steps you need to successfully install the package.
Conflicting File Paths and Directories	The file path of your programming language, script files and source data all matter. The computer will look for libraries and files in the directory you specify or the one you are currently in. If it cannot find a file that it needs to run a script, like a dependency, it won't run. Use the Command Line to find the location of your files and notebooks. One trick to identify a file path is to simply drag and drop a file into the CLI from the computer's file interface, like the MacOS Finder or Windows File Explorer.
Messy Source Data	Source data is often unlabeled or labeled in a way that isn't helpful for our purposes. To tackle this, you will have to get creative in how you identify, export and import the data. It might also mean doing additional manual identification, like inserting tags in a text editor, after exporting it.
Reference Error	This is a common error that typically means you are calling a variable without defining it first. If you ever get an error message containing "ref" or "reference," it's likely that you need to define the variable before your script can move on. When writing code, you will often find yourself having to define things you thought were implied. Remember, when writing a brand-new script, it doesn't know that an apple is a fruit, or even that it's an object. It just knows that "apple" is a five-character text string. Computers also read scripts vertically, in almost all cases. This means the variable needs to be defined above the line where the error occurs. Search with Ctrl+F or Cmd+F for the variable to find where it should have been defined.

Footnotes

Bulletproofing Your Stories https://slideplayer.com/slide/1452564/

Power Trips Archives https://publicintegrity.org/topics/politics/congress/power-trips/

ThisIsHowMuchaStarbucksAddstothePriceofaNearbyHomehttps://web.archive.org/web/20210614022911/https://www.cbsnews.com/news/starbucks-makes-property-values-jump-study-shows/

Coronavirus (COVID-19) Vaccinations https://ourworldindata.org/covid-vaccinations

Regex cookbook – Top 10 Most wanted regex https://web.archive.org/web/20210723025002/https://medium.com/factory-mind/regex-cookbook-most-wanted-regex-aa721558c3c1

REGEXEXTRACT https://support.google.com/docs/answer/3098244?hl=en

Quartz Guide to Bad Data https://qz.com/572338/the-quartz-guide-to-bad-data/

OpenRefine https://openrefine.org/

Obama Donors Dataset https://drive.google.com/file/d/1f4Pa3gftUQE4bs8gpqrbhp1JsV-E3hTx/view?usp=sharing

Obama Donors Shortlink https://bit.ly/obama2000

StackOverflow https://stackoverflow.com/

Mining Social Media: Finding Stories in Internet Data http://socialdata.site/

5 Basic Spreadsheets

Mike Reilley

A bridge collapse in Genoa, Italy, in the summer of 2018 left 43 people dead and the media around the world scrambling for localized story angles about crumbling infrastructure in their country. They need to look no further than a couple of spreadsheets.

In the US, MSNBC aired a story shortly after the collapse that was based on a recent study of US bridge inspections from the American Road & Transportation Builders Association (ARTBA).

The story touched on the study's highlights: 54,259 US bridges were "structurally deficient," and the bridges in the study were on average 67 years old. The article also revealed which states had the most and fewest structurally deficient bridges.

But that piece only scratched the surface about the condition of the nation's bridges. It was based on only one study and cited just a few statistics. That may be fine for a short national cable TV report, but closer examination using basic data analysis would yield more meaningful information – and potential story ideas – particularly for local news outlets.

For that, you need a spreadsheet. Microsoft Excel or Google Sheets works best, unless you're among the Mac-loving loyalists to Apple's Numbers software. Downloading or scraping data from a website and then sorting, filtering and analyzing that information is one of the foundational blocks of data journalism.

Spreadsheets seem imposing to many journalists. Many of us became writers to avoid math courses in high school and college, right? But even the simplest spreadsheet skills can produce newsworthy results and drive your reporting in new ways.

This chapter focuses on how to download and analyze a spreadsheet to find data points for news stories.

* * *

Video: Follow along with this exercise by watching this video: http://bit.ly/bridgeanalysisvideo

DOI: 10.4324/9781003273301-6

Exercise 1

Getting Started: Loading Data into a Spreadsheet

Every year, the US Federal Highway Administration posts data about bridge inspections. The National Bridge Inventory database rates bridges as being in Good, Fair or Poor condition. By sorting and filtering the data in a spreadsheet, you can determine which states have the most bridges and highest percentage of bridges in each category. You also can create a new category – Fair-Poor – to group the subpar bridges together.

We'll show you how to do this using the 2019 NBI data. But later, you can apply these skills to the data for the most recent year.

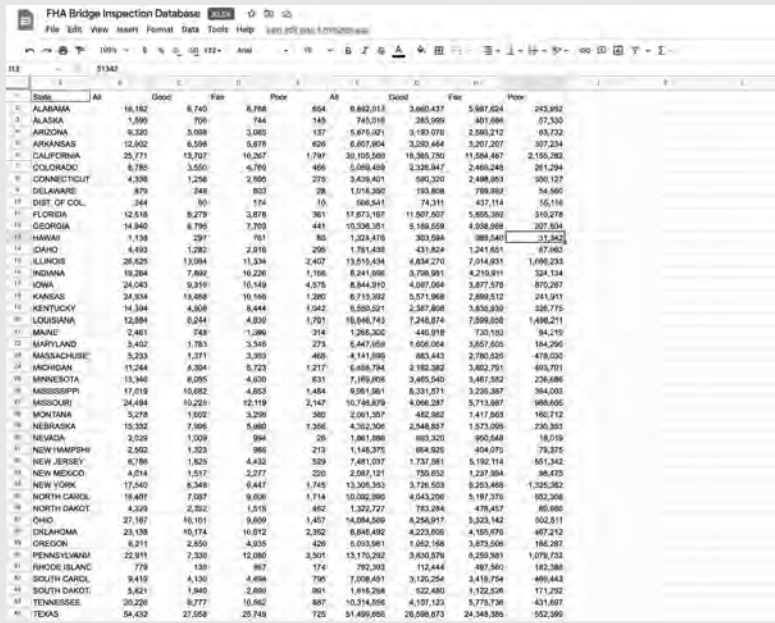
We're using Google Sheets for this exercise, but if you prefer Excel, you can follow the same steps, though the tools you'll use are named a little differently. You'll develop a preference over time.

1. Download the NBI dataset from here: https://bit.ly/ch5spreadsheet. Use the arrow or three vertical dots in the upper-right-hand corner to download it to your computer, and then drag and drop that file into a Google Drive folder.

   Or you can use the "Open With" pull-down menu at the top of the page, select Google Sheets and then go to the File menu and select "Make a Copy."

   Pro tip: To move around quickly on a large dataset, hold down the CMD (Apple) key and use the directional arrows to move up-down and side-to-side. On a PC, hold down the Control key and use the arrows.

2. Develop the habit of always working with a copy of the original data. By clicking on the tab of this worksheet, you can duplicate it. You might do this repeatedly during your analysis, naming each sheet whenever you make a significant change to the data. That way, you will have an audit trail of your work.
3. The first row of any dataset you work with should be labels of what each column of data represents (Figure 5.1). It should never start with just the data. In this case, there are two series of labels: Bridge Counts and Bridge Area (in meters). For this exercise, we'll focus on the Bridge Counts. Lock your column headings in place so that as you scroll down, you can always see what each column



FHA Bridge Inspection Database

File Edit View Insert Format Data Tools Help

State	All	Good	Fair	Poor	All	Good	Fair	Poor
ALABAMA	16,162	8,740	6,768	654	9,892,013	3,660,437	5,987,624	243,952
ALASKA	1,595	706	744	145	745,016	285,999	401,686	57,330
ARIZONA	8,320	5,098	3,085	137	5,675,021	3,193,078	2,586,212	63,732
ARKANSAS	12,902	6,596	5,878	626	6,607,904	3,293,464	3,207,207	307,234
CALIFORNIA	25,771	13,707	10,267	1,797	39,105,580	19,385,730	11,584,467	2,155,282
COLORADO	8,785	3,550	4,769	466	5,089,469	2,328,947	2,469,248	261,294
CONNECTICUT	4,338	1,258	2,805	275	3,439,401	580,320	2,498,953	360,127
DELAWARE	879	248	603	28	1,516,392	193,808	769,992	54,560
DIST. OF COL.	344	90	174	10	596,541	74,311	437,114	55,116
FLORIDA	12,518	8,279	3,878	361	17,573,167	11,507,507	5,855,382	310,278
GEORGIA	14,940	6,795	7,703	441	10,336,351	5,189,559	4,938,968	207,504
HAWAII	1,138	297	761	80	1,324,476	363,594	929,540	31,342
IDAHO	4,493	1,282	2,916	295	1,791,488	431,824	1,241,551	67,982
ILLINOIS	26,825	13,094	11,334	2,407	13,510,434	6,834,270	7,014,931	1,660,233
INDIANA	19,284	7,892	10,226	1,166	8,241,986	3,798,951	4,219,911	524,134
IOWA	24,043	9,319	10,149	4,575	5,844,910	4,097,064	3,877,578	870,267
KANSAS	24,934	13,468	10,166	1,280	6,713,392	5,571,968	2,899,512	241,911
KENTUCKY	14,394	4,806	8,444	1,042	6,560,521	2,367,808	3,836,939	326,775
LOUISIANA	12,884	6,244	4,830	1,701	10,946,743	7,248,874	7,999,808	1,498,211
MAINE	2,461	748	1,399	314	1,266,302	440,918	730,153	94,215
MARYLAND	5,402	1,783	3,346	273	5,447,159	1,606,064	3,857,605	184,296
MASSACHUSE	5,233	1,371	3,393	468	4,141,895	883,443	2,780,526	478,030
MICHIGAN	11,244	4,304	5,723	1,217	6,658,794	2,182,382	3,882,791	693,701
MINNESOTA	13,346	6,085	4,630	631	7,169,806	3,465,540	3,467,582	236,686
MISSISSIPPI	17,019	10,682	4,853	1,484	9,961,961	8,331,571	3,236,387	394,003
MISSOURI	24,484	10,221	12,119	2,147	10,746,879	4,996,287	5,713,987	986,605
MONTANA	5,278	1,607	3,299	380	2,061,357	482,982	1,417,663	160,712
NEBRASKA	15,332	7,996	5,980	1,356	4,362,306	2,548,857	1,573,095	230,353
NEVADA	2,029	1,009	994	26	1,861,898	893,320	950,548	18,019
NEW HAMPSHI	2,502	1,323	966	213	1,146,375	664,925	404,075	79,375
NEW JERSEY	6,786	1,825	4,432	529	7,481,037	1,737,981	5,192,114	551,342
NEW MEXICO	4,014	1,517	2,277	220	2,087,121	750,652	1,237,994	98,475
NEW YORK	17,540	8,348	8,447	1,745	13,305,353	3,726,503	8,253,466	1,325,382
NORTH CAROL	19,401	7,087	9,659	1,714	10,092,990	4,043,206	5,197,376	852,308
NORTH DAKOT	4,329	2,332	1,515	482	1,322,727	783,284	478,457	60,986
OHIO	27,187	16,101	9,609	1,457	14,084,569	8,258,917	5,323,142	502,511
OKLAHOMA	23,138	10,174	10,512	2,352	8,646,492	4,223,606	4,155,676	467,212
OREGON	8,211	2,850	4,935	426	5,093,961	1,662,166	3,873,506	186,287
PENNSYLVANI	22,911	7,330	12,880	3,501	13,170,282	3,830,579	8,259,981	1,079,732
RHODE ISLANC	779	139	867	174	792,303	112,444	497,560	182,388
SOUTH CAROL	9,419	4,130	4,494	795	7,008,451	3,120,254	3,419,754	469,443
SOUTH DAKOT	5,621	1,940	2,880	801	1,816,298	522,480	1,122,526	171,292
TENNESSEE	20,226	8,777	10,862	587	10,314,556	4,107,123	5,775,736	431,697
TEXAS	54,432	27,958	25,749	725	51,499,886	26,598,873	24,348,386	552,399

Figure 5.1 Bridge inspections database in Google Sheets.

means. In Google Sheets, you do this by going to View > Freeze > 1 row.

4. Do a slow scroll through your dataset and make sure the set is complete and there are no empty or garbled cells. Data entry with some government entities can be spotty. Make sure your set passes initial inspection before starting any analysis. If data is missing or garbled, contact the public information officer for that government agency to get the issue cleared up.

Interviewing Your Dataset

ProPublica's Derek Willis builds incredible data visualizations and databases on all kinds of social and economic issues. He knows his way around a spreadsheet, so when he talks about them, you listen closely. Willis uses a popular approach to spreadsheet analysis: He interviews data the same way he would a person.

"Interviewing is a skill. I've heard this repeatedly as a college student, at journalism conferences and in newsrooms," Willis wrote in a July 2014 *MediaShift* article. "Not all of us are born with the ability to form penetrating questions or to extract crucial details from sources. Like any skill, interviewing takes practice and preparation."

Willis argues that

> journalists who wouldn't consider themselves "data journalists" already have the necessary foundation for asking good questions of data. Just as you would background a person you were interviewing for a story, data has its own history. Interviewing data takes some practice, but the benefits for journalists are plentiful. You'll learn how to find the holes in data (and there are always holes), how to avoid misinterpretations and how to find better stories.

For more data journalism tips, tricks and exercises, visit the Data + Journalism blog at http://dataplusjournalism.com

Think about it: Our first week in journalism school, we were taught to ask these basic questions when interviewing for a news story:

- Who?
- What?
- When?
- Where?
- Why?
- How/how much?

So what if you were writing a story about a city budget? You might ask the city manager, the mayor or the city treasurer questions like:

- How much did the budget increase?
- Who got the biggest raise? Whose pay was cut?
- Why was the police budget cut? How much?
- Where is the increased funding for schools coming from? (Hint: Bet your property taxes went up!)

The same questions apply when sorting and filtering your data. Look over the header labels, and start asking yourself some questions:

- How many bridges in my state were in good condition? That's easy, just look in the corresponding cell.

- What percentage of bridges in my state were in poor condition? You'll need to use a spreadsheet formula (gasp!) to calculate that. Don't worry, we'll show you how.
- What percentage of bridges in my state were in fair-poor condition? We'll need to add cells together and divide by the total number of bridges in my state.
- And finally, where did my state rank among all states in poor bridges? Fair? Fair-poor?
- Who had the most poor bridges? Who had the least?

Once you start this analysis, you'll develop a list of data points that will help guide your story. When I teach this approach to students, I have them type the data points into a Google Doc as they do the analysis so they won't forget them. In the end, they have a list of data and an outline for a great story in the process.

This interview approach works well for datasets of all sizes. It's particularly helpful in teaching college students or professional students who might feel overwhelmed by even a smaller dataset. At the very least, it's an excellent way to think through a complex sheet of data in a way we can all grasp as journalists.

Sunne applied the interviewing data approach to a KBIA story about federal inspections and violations at an animal sanctuary in Columbia, Missouri. (shortlink: https://bit.ly/sunnestory) She found a downloadable USDA database of animal welfare checks and simply filtered the welfare checks for Missouri.

"I had just graduated from the University of Missouri, and sorted them in order of who had the most violations," she said.

> Lo and behold, the dirtiest and most dangerous exotic animal location in the state was right there in my college town, a couple miles south of the mall. So that was a pretty easy, obvious story that I ended up producing for the NPR station there.

Exercise Quick Challenge: Calculation Percentage

According to the ARTBA bridge inspections study, 54,259 of 612,677 US bridges are rated "structurally deficient." How would we figure that as a percentage for our readers? And how would we write it?

Answer: Divide 54,259 by 612,667 to get 8.8 percent. This can easily be done on your phone calculator app. We'll show you how to do it on a spreadsheet later in this exercise.

Analyzing Bridge Inspections: Calculate Percentages in Google Sheets

Once your data is in the Google Sheet, use these steps to calculate the percentage of fair bridges, poor bridges and fair-poor bridges for each state. Let the spreadsheet do the heavy lifting for you by using the formulas below to calculate the percentages.

Think of your spreadsheet like the grid on the game Battleship you played as a kid: Cell J2 contains some data. Is it a hit or a miss? How do I use it? Same with cell K20, C13, etc.

Now you're ready to do some analysis.

1. Row 1 should be the labels of each column. Every row below that should contain state data, starting with Alabama.
2. Now go to cell J1, and type the words Percentage Fair. In cell K1, add the words Percentage Poor. And in cell L1, type the words Percentage Fair-Poor.
3. In column J, in the Alabama row: Figure the percentage of fair bridges by pasting this formula into cell J2: =D2/B2

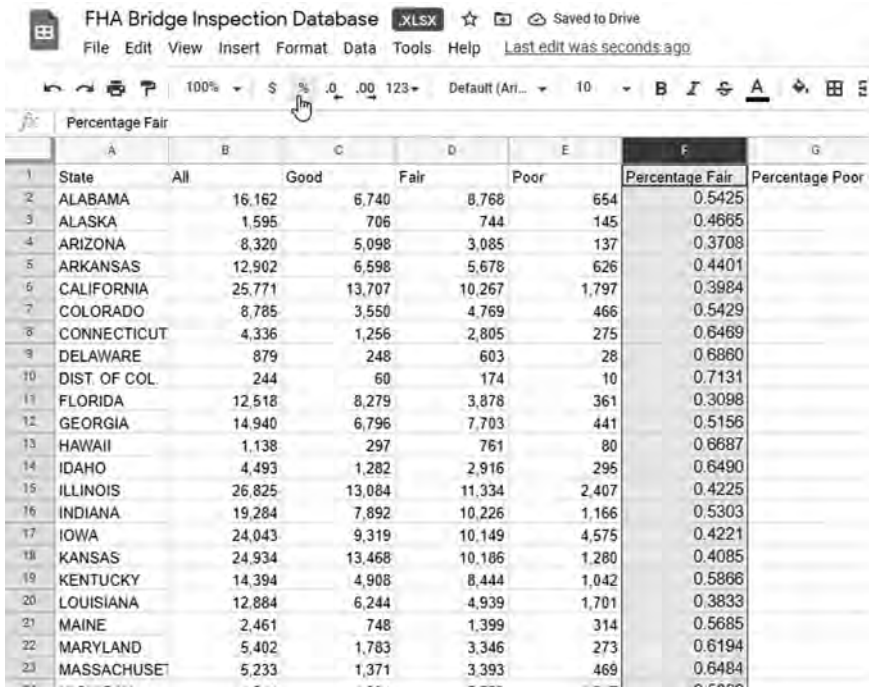
   * While you have cell J2 highlighted, hit CMD-C (Control-C on a PC) to copy it and then drag the cursor to the bottom of the J row, so all of the column is highlighted. Then hit the Paste button (CMD-V on Mac, CTRL-C on PC) and voila! It will calculate all of the percentages for each state/row.

   Troubleshooting: If the column doesn't calculate properly, just hit CMD-Z (Mac) or CTRL-Z (PC) to back out of the calculations and repeat the steps. It should work after a couple of attempts.
4. With the J column highlighted, click the decrease decimal point button a few times to move the decimal point over a few places on the data so it's down to three decimals. (See Figure 5.2. The hand is over the button, two to the left of the $ button.)

   Pro Tips

   a. You can skip the decimal step by hitting the percentage sign button (%) to convert the decimals into a percentage, and it automatically rounds off the decimals for you. You'll need to repeat these steps for every column of data.
   b. Another way to convert to percentages: Click on the J column at the top to highlight the whole column, then right click (Control+Click on Mac) to get the pull-down menu and select Format Cells. On the next interface, select Percentages. It will convert the decimals into a percentage with two decimal points.
5. In column K, in the Alabama row: Figure the percentage of poor bridges by pasting this formula into cell K2: =E2/B2

   * Then copy that formula and select all of column K and hit paste. It will calculate all of the percentages for each state/row.
6. In column L, in the Alabama row: Figure the percentage of fair/poor bridges by pasting this formula into cell L2: =(D2+E2)/B2



	A	B	C	D	E	F	G
1	State	All	Good	Fair	Poor	Percentage Fair	Percentage Poor
2	ALABAMA	16,162	6,740	8,768	654	0.5425	
3	ALASKA	1,595	706	744	145	0.4665	
4	ARIZONA	8,320	5,098	3,085	137	0.3708	
5	ARKANSAS	12,902	6,598	5,678	626	0.4401	
6	CALIFORNIA	25,771	13,707	10,267	1,797	0.3984	
7	COLORADO	8,785	3,550	4,769	466	0.5429	
8	CONNECTICUT	4,336	1,256	2,805	275	0.6469	
9	DELAWARE	879	248	603	28	0.6860	
10	DIST. OF COL.	244	60	174	10	0.7131	
11	FLORIDA	12,518	8,279	3,878	361	0.3098	
12	GEORGIA	14,940	6,796	7,703	441	0.5156	
13	HAWAII	1,138	297	761	80	0.6687	
14	IDAHO	4,493	1,282	2,916	295	0.6490	
15	ILLINOIS	26,825	13,084	11,334	2,407	0.4225	
16	INDIANA	19,284	7,892	10,226	1,166	0.5303	
17	IOWA	24,043	9,319	10,149	4,575	0.4221	
18	KANSAS	24,934	13,468	10,186	1,280	0.4085	
19	KENTUCKY	14,394	4,908	8,444	1,042	0.5866	
20	LOUISIANA	12,884	6,244	4,939	1,701	0.3833	
21	MAINE	2,461	748	1,399	314	0.5685	
22	MARYLAND	5,402	1,783	3,346	273	0.6194	
23	MASSACHUSET	5,233	1,371	3,393	469	0.6484	

Figure 5.2 The decrease decimal and percent buttons in Google Sheets.

In essence, you're adding the fair and poor columns together and dividing by the total number of bridges.

* Then copy that formula down column L.

7. Before sorting we're going to insert what is called a "data moat" between the totals at the bottom of the sheet and all of the states and districts. Just click on the Totals row, go to the Insert button at the top menu and select Insert/Rows/Insert Row Above.
8. Now we're going to sort by column L and order them from highest to lowest percentage. Now drag your cursor from cell A1 across the data you want to sort, excluding, of course, the Totals row (see Figure 5.3).
9. Now go to the Data pull-down at the top of the menu and select Sort Range, and then select Advanced range sorting options (see Figure 5.4).
10. In the Sort menu, make sure to check the Data Has Header Row box and change the column pull-down menu to Percentage Fair (column L) and the order Z to A, and then hit the Sort button (see Figure 5.5).
11. Now your sheet is sorted by state/Puerto Rico (be mindful of totals in row 34) in order of highest to lowest Fair-Poor condition bridges. Now you're ready to answer the questions below in the Challenge section.

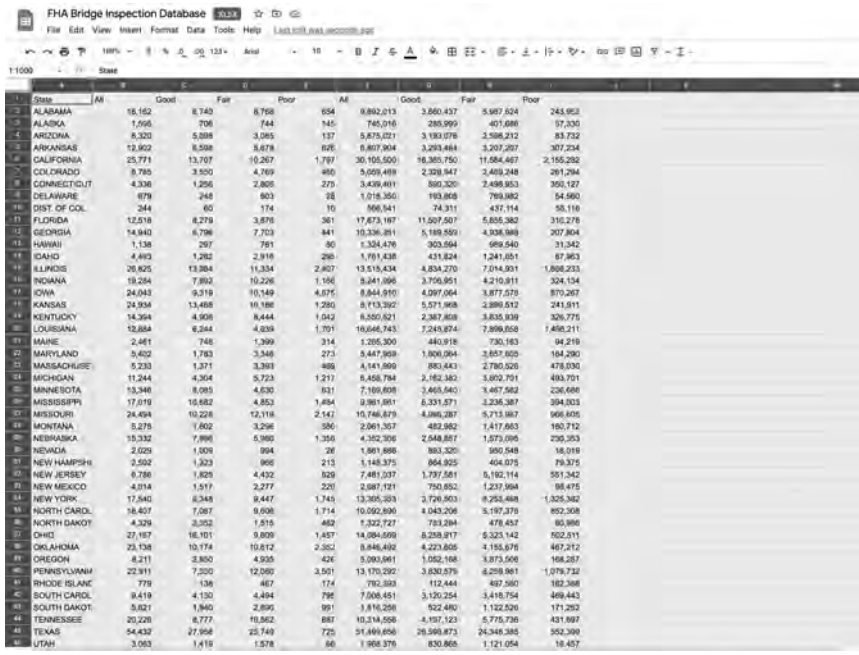


Figure 5.3 Highlighting data in the sheet prior to sorting.

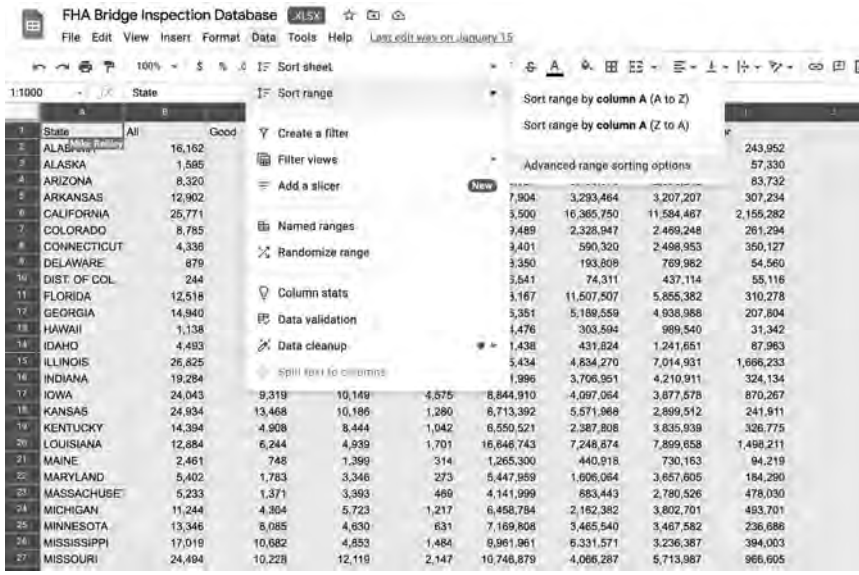


Figure 5.4 Sort Range in the Data pull-down menu.

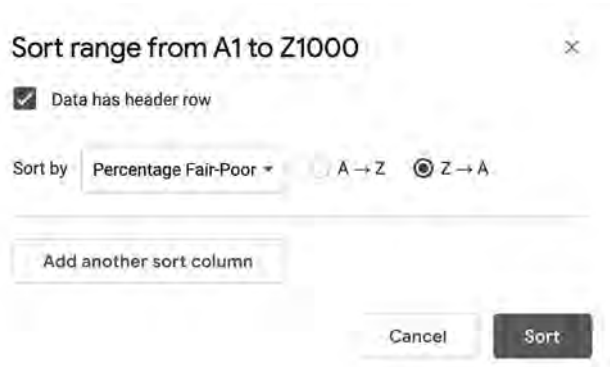


Figure 5.5 Sort Range interface.

You can also repeat these sorting steps for columns J (Percentage Fair) and K (Percentage Poor) to see where your state ranks in those categories.

Be sure to write your data findings out in a Google Doc as you sort. Your story will begin to write itself.

Exercise Challenge: Answer These Questions

1. Which three states had the most fair/poor bridges by percentage? (Be mindful that Guam, Puerto Rico and the US Virgin Islands are not states.)
2. Which three states had the least?
3. Where does Illinois rank among all states with fair/poor bridges? (Be mindful that Row 34 is a Totals column and you have a header in row 1, as well as nonstates in the list.)
4. Where does your state rank on the list? Be sure to save these answers as you'll use them to write a short story for an exercise in Chapter 7, Writing a Data Story.

Answers

1. Most: Rhode Island, District of Columbia and West Virginia. (Listing Puerto Rico is fine, but it's not a state.)
2. Least: Florida, California and Mississippi
3. 34th. It's 37th in the sheet, but take into account the header row, Puerto Rico, which is not a state, and that Totals appear in row 34.

   Pro tip: To avoid the Totals row appearing in the sort, insert a blank row between Wyoming and Totals before sorting. This is called a "data moat," much like a moat around a castle. It's designed to keep totals away from other rows of data in budgets and other datasets. Also make sure to select just the data you want to sort above the moat, or the totals will get mixed into the list.

On Your Own: Analyze Bridges by Area

You can repeat these percentage formulas and sort for the columns showing the Bridge Area data – the length of the bridges in each state. This can yield some interesting and different results.

For example, Iowa is bordered by two very large bridges over the Missouri and Mississippi Rivers that are considerably larger than the Sutliff Bridge over the tiny Cedar River. California also has many large, expansive bridges.

Compiling the percentages of poor or fair bridges by area will produce different results from states that have a large number of poor bridges that are smaller in area. Try it on your own, and compare the results. It will give the reader a more complete picture of the issue.

Pro Tip 

If you need to create a new Google Sheet from scratch, simply type Sheets.new into a browser window. Name the new sheet in the upper-left corner, and then go to File/Move to and save it to a specific folder in your Google Drive.

Filters and Datasets

In the fall of 2020, the Big Ten Conference made a controversial decision to start the football season more than a month later than most Power Five conferences. The conference based its decision on research about COVID-19 possibly causing heart disease in players. The decision was lambasted by the media and drew outrage from athletic officials, coaches, players and fans.

The plan backfired. The Big Ten started the season later, right as COVID-19 cases began to spike. Several games were canceled as players and coaches got sick. But just how bad was COVID-19 on the Big Ten campuses?

As the season drew to a close, Chris Katsaros and Charles Tharpe, two reporters for RedLineProject.org based in Chicago, examined the number of cases on each campus. They used filters and basic mathematical formulas to determine which campuses had the most cases and which had the most per 1,000 students. The latter data point was important as the Big Ten has a wide range of student enrollment: Ohio State has more than 61,000 students, while Northwestern has just over 22,000. Enrollment obviously impacted the chances for an outbreak and had to be factored into the equation.

They did it by loading this spreadsheet into Google Sheets: http://bit.ly/bigtencovid19. The dataset has three tabs at the bottom:

1. Big Ten Daily COVID-19 cases, which we'll use for filtering exercises.
2. Big Ten COVID-19 cases per 1,000 students, which we'll use for math calculations.
3. The answer key for the cases per 1,000 students.

Video: Watch this video on compiling rates with the same data used in this exercise: http://bit.ly/covidratesvideo

Getting Started

Once you have downloaded the spreadsheet and opened it in Google Drive, click on the Big Ten Daily COVID-19 cases in the bottom-left tab and inspect the data (see Figure 5.6). Use the CMD (CTRL on PC) arrow up/down keys to move from the top to the bottom of the dataset, and do a slow scroll through some of the sections to see data for the various schools. Note that Column F is a running total for positive cases, not positive tests per day.

The daily data are grouped by university. While useful, the sheet has 1,894 rows and is cumbersome to use. It would be much easier if you could examine just the rows for each school, right? That's why you should use filters, which help journalists isolate specific data in a large set.

The name of the school appears in Column A, so if we can isolate that, we can easily inspect each school's data.

Big Ten Daily Covid-19 XLSX

File Edit View Insert Format Data Tools Help Last edit was on July 16

A1 University

	A	B	C	D	E	F	G	H
1	University	City	State	Date	Tests	Confirmed Cases		
2	Illinois	Champaign	Illinois	7/6/2020	99	0		
3	Illinois	Champaign	Illinois	7/7/2020	335	0		
4	Illinois	Champaign	Illinois	7/8/2020	933	2		
5	Illinois	Champaign	Illinois	7/9/2020	1652	3		
6	Illinois	Champaign	Illinois	7/10/2020	2382	8		
7	Illinois	Champaign	Illinois	7/11/2020	2382	8		
8	Illinois	Champaign	Illinois	7/12/2020	2382	8		
9	Illinois	Champaign	Illinois	7/13/2020	3051	10		
10	Illinois	Champaign	Illinois	7/14/2020	3679	12		
11	Illinois	Champaign	Illinois	7/15/2020	4217	18		
12	Illinois	Champaign	Illinois	7/16/2020	4834	25		
13	Illinois	Champaign	Illinois	7/17/2020	5559	28		
14	Illinois	Champaign	Illinois	7/18/2020	5559	28		
15	Illinois	Champaign	Illinois	7/19/2020	5559	28		
16	Illinois	Champaign	Illinois	7/20/2020	6399	31		
17	Illinois	Champaign	Illinois	7/21/2020	6906	34		
18	Illinois	Champaign	Illinois	7/22/2020	7538	37		

Figure 5.6 Big Ten positive COVID-19 cases by university.

1. Go to the Data pull-down menu at the top of the spreadsheet and select Create a Filter from the menu (see Figure 5.7).

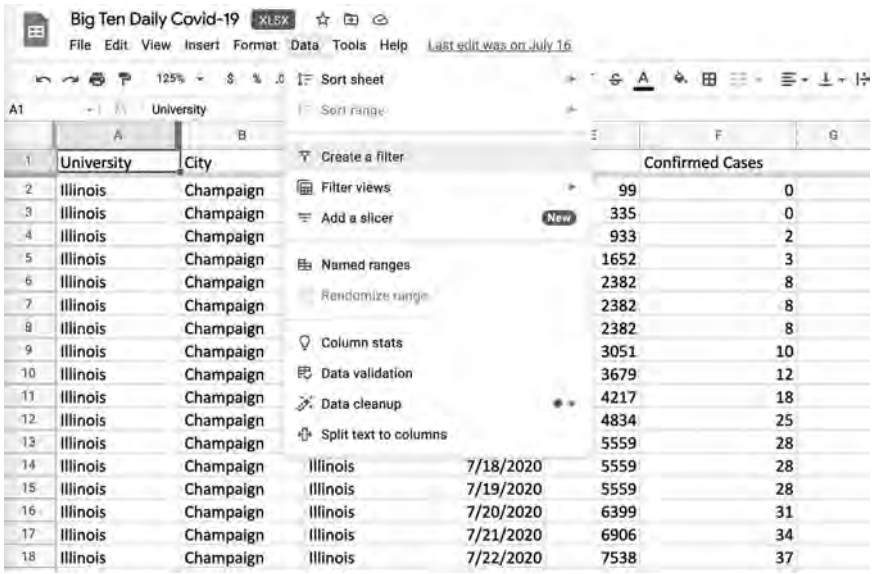


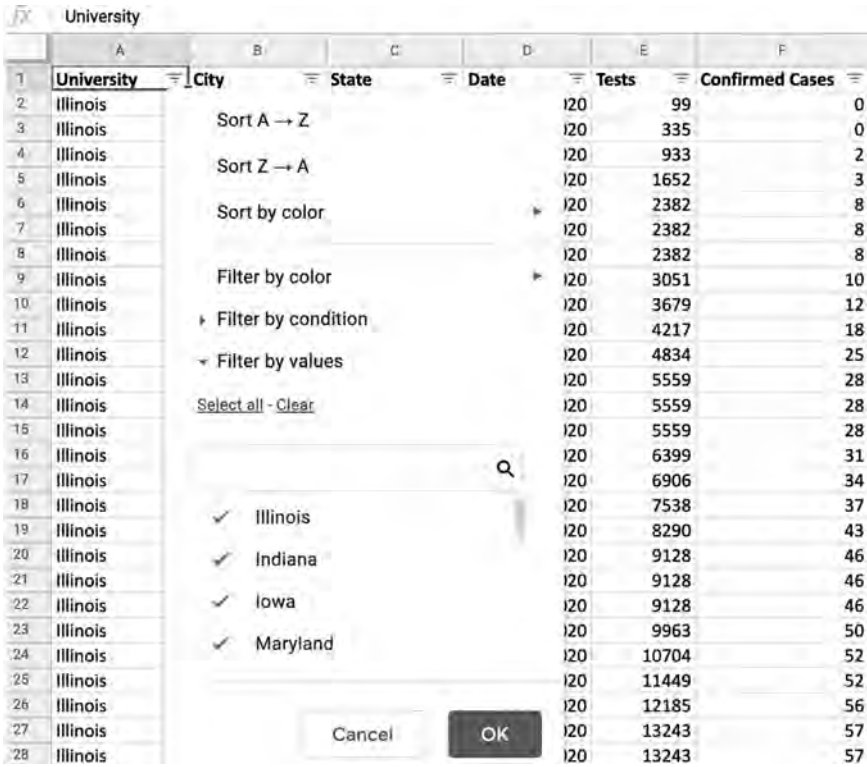
Figure 5.7 Data/Create a Filter pull-down menu.

2. Once you create the filter, small triangular icons appear in your header row next to the words in each cell (see Figure 5.8).

	A	B	C	D	E	F
1	University	City	State	Date	Tests	Confirmed Cases
2	Illinois	Champaign	Illinois	7/6/2020	99	0
3	Illinois	Champaign	Illinois	7/7/2020	335	0
4	Illinois	Champaign	Illinois	7/8/2020	933	2
5	Illinois	Champaign	Illinois	7/9/2020	1652	3
6	Illinois	Champaign	Illinois	7/10/2020	2382	8
7	Illinois	Champaign	Illinois	7/11/2020	2382	8
8	Illinois	Champaign	Illinois	7/12/2020	2382	8
9	Illinois	Champaign	Illinois	7/13/2020	3051	10

Figure 5.8 Filters.

3. Click on the filter next to University in cell A1. In the interface, hit the Clear button, and then click on Illinois to select it. Then hit the green OK button (see Figure 5.9).



	University	City	State	Date	Tests	Confirmed Cases
2	Illinois			120	99	0
3	Illinois			120	335	0
4	Illinois			120	933	2
5	Illinois			120	1652	3
6	Illinois			120	2382	8
7	Illinois			120	2382	8
8	Illinois			120	2382	8
9	Illinois			120	3051	10
10	Illinois			120	3679	12
11	Illinois			120	4217	18
12	Illinois			120	4834	25
13	Illinois			120	5559	28
14	Illinois			120	5559	28
15	Illinois			120	5559	28
16	Illinois			120	6399	31
17	Illinois			120	6906	34
18	Illinois			120	7538	37
19	Illinois			120	8290	43
20	Illinois			120	9128	46
21	Illinois			120	9128	46
22	Illinois			120	9128	46
23	Illinois			120	9963	50
24	Illinois			120	10704	52
25	Illinois			120	11449	52
26	Illinois			120	12185	56
27	Illinois			120	13243	57
28	Illinois			120	13243	57

Figure 5.9 Filter pull-down.

4. The spreadsheet will isolate only the 147 rows for the University of Illinois in Champaign. You can then sort and reorder the spreadsheet by cases or tests to begin asking questions of the data, much like we did in sorting the bridges database.

The advantage of using filters is you can extract the isolated data by copying it to a new sheet, tab or saving it as its own sheet. The original dataset is still there. All you have to do is go back to the Data pull-down menu and select Turn off Filter and the complete dataset will return.

We'll do more filtering and sorting in Chapter 6, Advanced Spreadsheets and R.

Figuring Rates

Rates help level the playing field when making comparisons. They have been very helpful in calculating COVID-19 data by comparing rates in large cities to those in smaller towns. They're also used when comparing homicides between larger and smaller cities.

Spreadsheet formulas are extremely helpful in calculating math problems, both basic and complex. We'll apply some of those formulas to our second exercise: Big Ten Cases per 1,000 students. Click on that tab in the Big Ten exercise spreadsheet you have opened, and inspect the data.

We'll use formulas to total the positive cases, enrollment and the average positive cases per school in the Totals row at the bottom of the spreadsheet. The average will be helpful in seeing which schools are above or below the average.

Google Sheets Formulas

Google compiled a deep list of formulas (functions), so you don't have to memorize them. Bookmark this to keep it handy. You'll use it a lot: https://bit.ly/googlesheetsform

Then we'll figure the rate for each school by dividing the positive cases per school by the enrollment and normalizing it to 1,000 students. Normalizing the data gives readers scale and helps them better understand the impact on each school rather than just looking at raw numbers.

1. Click in cell B17 and type this formula: =sum(b2:b15) and hit return (Figure 5.10). As you finish typing, the answer should pop up in blue just above the formula. You should get 41,422 positive cases. Remember, think of your spreadsheet as a game of Battleship: You want to summarize cells B2 to B15. These formulas work in rows as well, which we discovered in the bridge inspections exercise.

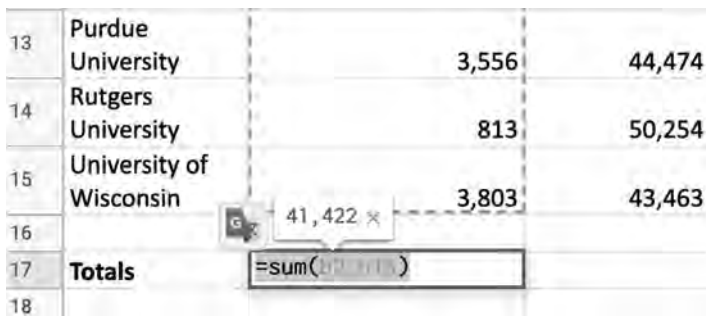
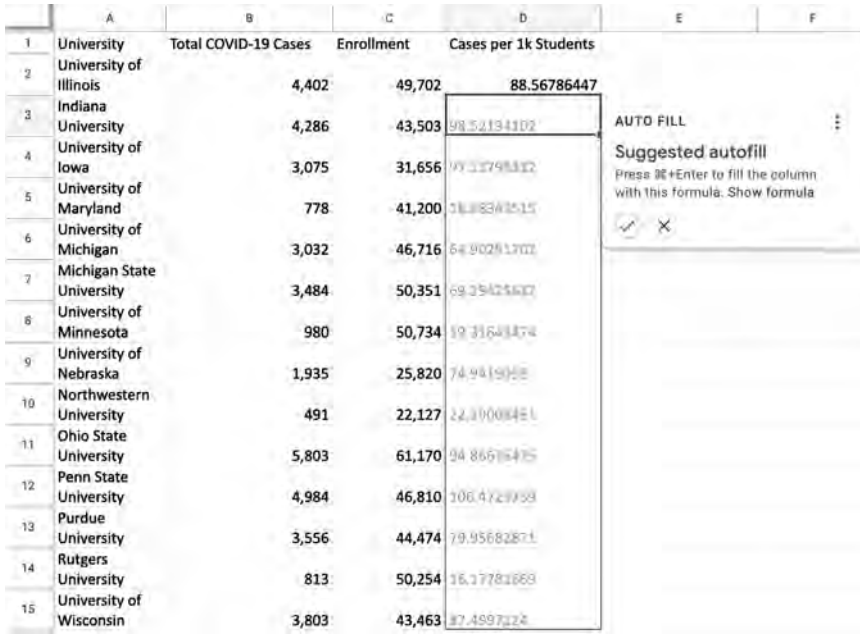


Figure 5.10 Sum formula.

2. Now click in cell C17 and type this formula =sum(c2:c15) and hit return.
3. You now have two data points for your story: During the fall football season, Big Ten schools had 41,422 positive COVID-19 cases, while their total enrollments stood at 607,980. Not earth-shattering, but it's a good start.

4. Now let's make a new column. In cell D1, type "Cases per 1k students".
5. Click in cell D17 and type in this formula: =b17/c17*1000. You should get 68.1. (Tip: Use the decrease decimal button in the top toolbar to round down the number to one-tenth.) We now know that the average number of positive cases per 1,000 students on a Big Ten campus in fall 2020 was 68.1. This number will be helpful when we calculate and sort the cases per 1,000 for each school.
6. Now go to cell D2 and type this formula: =b2/c2*1000 and hit return. You should get 88.56, and Google Sheets will give you the option to autofill the rest of the cells to get the totals for each row. Click the green checkmark to do so (see Figure 5.11).

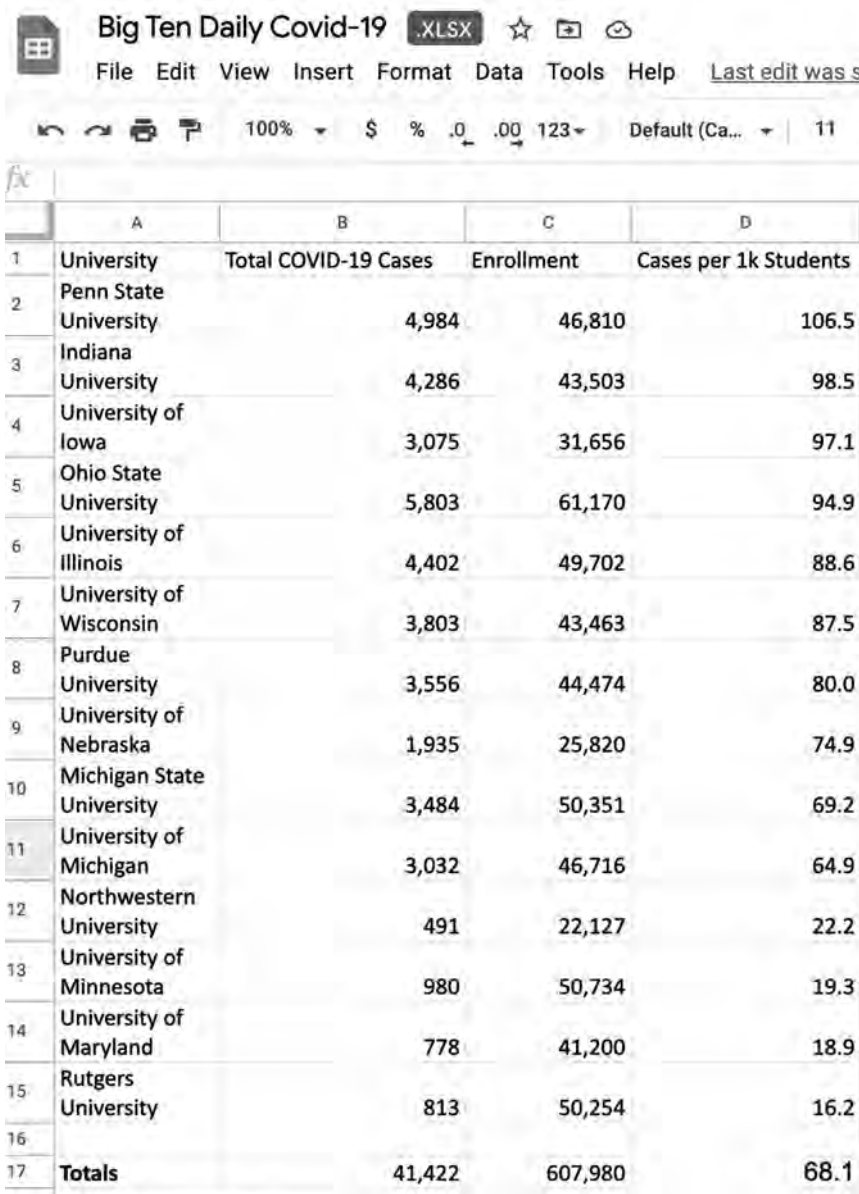


	A	B	C	D	E	F
1	University	Total COVID-19 Cases	Enrollment	Cases per 1k Students		
2	University of Illinois	4,402	49,702	88.56786447		
3	Indiana University	4,286	43,503	98.52134102		
4	University of Iowa	3,075	31,656	97.13798832		
5	University of Maryland	778	41,200	18.88349515		
6	University of Michigan	3,032	46,716	64.90251701		
7	Michigan State University	3,484	50,351	69.2542562		
8	University of Minnesota	980	50,734	19.31648474		
9	University of Nebraska	1,935	25,820	74.9419058		
10	Northwestern University	491	22,127	22.19008451		
11	Ohio State University	5,803	61,170	94.86616475		
12	Penn State University	4,984	46,810	106.4723559		
13	Purdue University	3,556	44,474	79.95682871		
14	Rutgers University	813	50,254	16.17781669		
15	University of Wisconsin	3,803	43,463	87.4997124		

Figure 5.11 Autofill columns.

7. Click the decrease decimal point button a few times to round off the data to one-tenth. Then highlight only rows 1 through 15, and sort the data Z to A by the "Cases per 1k students" column.

   You'll see Penn State surges past Ohio State as the leader in positive cases per 1,000 students. Indiana and Iowa also are near the top, the two Michigan Schools hovered right at the average and Minnesota, Maryland and Rutgers had the lowest rates (Figure 5.12).



	A	B	C	D
1	University	Total COVID-19 Cases	Enrollment	Cases per 1k Students
2	Penn State University	4,984	46,810	106.5
3	Indiana University	4,286	43,503	98.5
4	University of Iowa	3,075	31,656	97.1
5	Ohio State University	5,803	61,170	94.9
6	University of Illinois	4,402	49,702	88.6
7	University of Wisconsin	3,803	43,463	87.5
8	Purdue University	3,556	44,474	80.0
9	University of Nebraska	1,935	25,820	74.9
10	Michigan State University	3,484	50,351	69.2
11	University of Michigan	3,032	46,716	64.9
12	Northwestern University	491	22,127	22.2
13	University of Minnesota	980	50,734	19.3
14	University of Maryland	778	41,200	18.9
15	Rutgers University	813	50,254	16.2
16				
17	**Totals**	41,422	607,980	68.1

Figure 5.12 The sorted sheet.

Turning the Data into a Story

After Katsaros and Tharpe finished filtering, sorting and calculating the rates, they interviewed expert sources and did some additional reporting for the story. They used the data to create a small database of Big Ten university cases and slick animated graphics in Google Flourish, which you'll learn to use in a later chapter.

The story, published in December 2020, highlighted not just the total number of cases but the rate per 1,000 students at each school. Normalizing the data gave a clearer picture of what was happening on Big Ten campuses. Ohio State dropped from first (total cases) to second behind Penn State in cases per 1,000 students. In an ironic twist, Ohio State beat Northwestern, which had the fewest cases, in the Big Ten Championship game. One stat that didn't appear in the box score: Not one Northwestern football player tested positive for COVID-19 the entire season.

We'll explore more writing approaches in Chapter 7.

* * *

Exercise 2

City Budget

For our next exercise, we're going to analyze a fictitious city budget. Download or make a copy of the exercise here: https://bit.ly/citybud.

Note the budget has department revenues and expenses. We're going to compile the percentage change between last year's and this year's budgets for each department's revenues and expenses. Then we will compile the percentage of the total budget each department is getting. Finally, we'll double-check the totals that the city has given us to make sure they are correct. Note that the budget has a data moat between the totals and each department row.

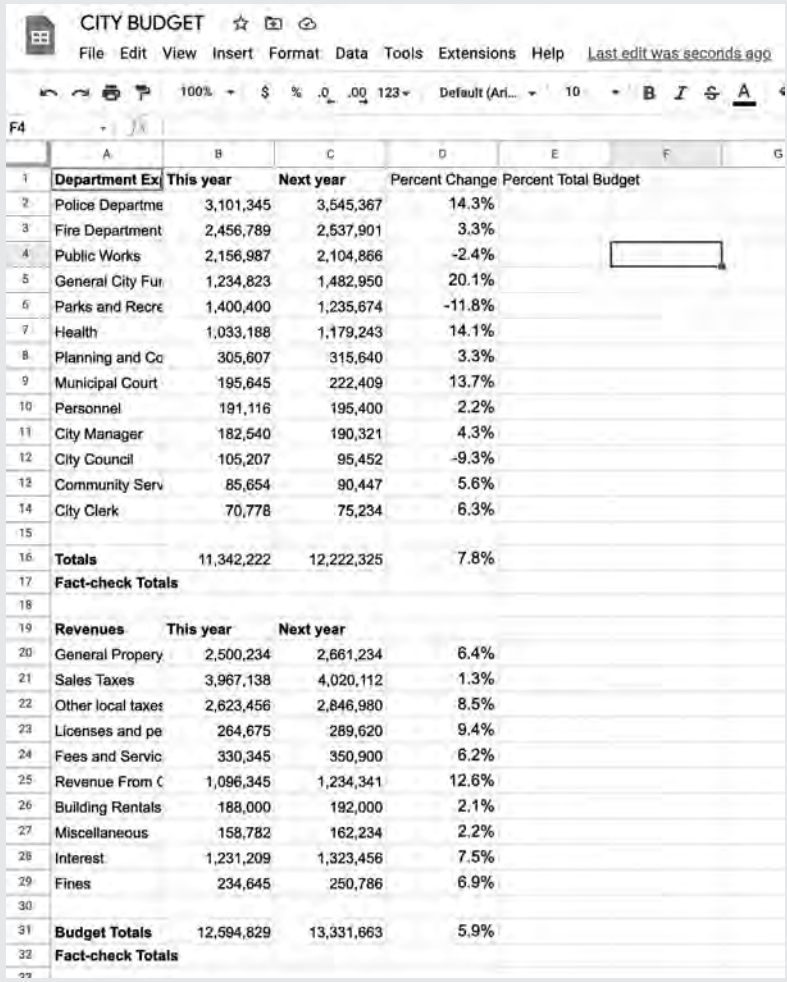
Follow these steps, and use the formulas below. Apply what you learned on previous exercises to fill in the columns. Be prepared to answer some questions at the end of the exercise.

1. **Column D: Figure percent change**

   Start by labeling cell D1: Percent Change.

   The formula you'll use is =(new budget-old budget)/old budget (this is also known as the NOO acronym). So type this formula into cell D2 and drag the cell border down or hit Autofill to fill the rest of the column: =(c2-b2)/b2

   Delete any errors in the blank rows, then highlight all of column D and click on the % button in the toolbar and click the decimal left button next to it once to move the percentage over to one decimal point. Your spreadsheet should look like this when finished (Figure 5.13):



CITY BUDGET

File Edit View Insert Format Data Tools Extensions Help Last edit was seconds ago

F4

	A	B	C	D	E	F	G
1	**Department Ex**	**This year**	**Next year**	**Percent Change**	**Percent Total Budget**		
2	Police Departme	3,101,345	3,545,367	14.3%			
3	Fire Department	2,456,789	2,537,901	3.3%			
4	Public Works	2,156,987	2,104,866	-2.4%			
5	General City Fur	1,234,823	1,482,950	20.1%			
6	Parks and Recre	1,400,400	1,235,674	-11.8%			
7	Health	1,033,188	1,179,243	14.1%			
8	Planning and Co	305,607	315,640	3.3%			
9	Municipal Court	195,645	222,409	13.7%			
10	Personnel	191,116	195,400	2.2%			
11	City Manager	182,540	190,321	4.3%			
12	City Council	105,207	95,452	-9.3%			
13	Community Serv	85,654	90,447	5.6%			
14	City Clerk	70,778	75,234	6.3%			
15							
16	**Totals**	11,342,222	12,222,325	7.8%			
17	**Fact-check Totals**						
18							
19	**Revenues**	**This year**	**Next year**				
20	General Propery	2,500,234	2,661,234	6.4%			
21	Sales Taxes	3,967,138	4,020,112	1.3%			
22	Other local taxes	2,623,456	2,846,980	8.5%			
23	Licenses and pe	264,675	289,620	9.4%			
24	Fees and Servic	330,345	350,900	6.2%			
25	Revenue From C	1,096,345	1,234,341	12.6%			
26	Building Rentals	188,000	192,000	2.1%			
27	Miscellaneous	158,782	162,234	2.2%			
28	Interest	1,231,209	1,323,456	7.5%			
29	Fines	234,645	250,786	6.9%			
30							
31	**Budget Totals**	12,594,829	13,331,663	5.9%			
32	**Fact-check Totals**						

Figure 5.13 City budget percent change column in Google Sheets.

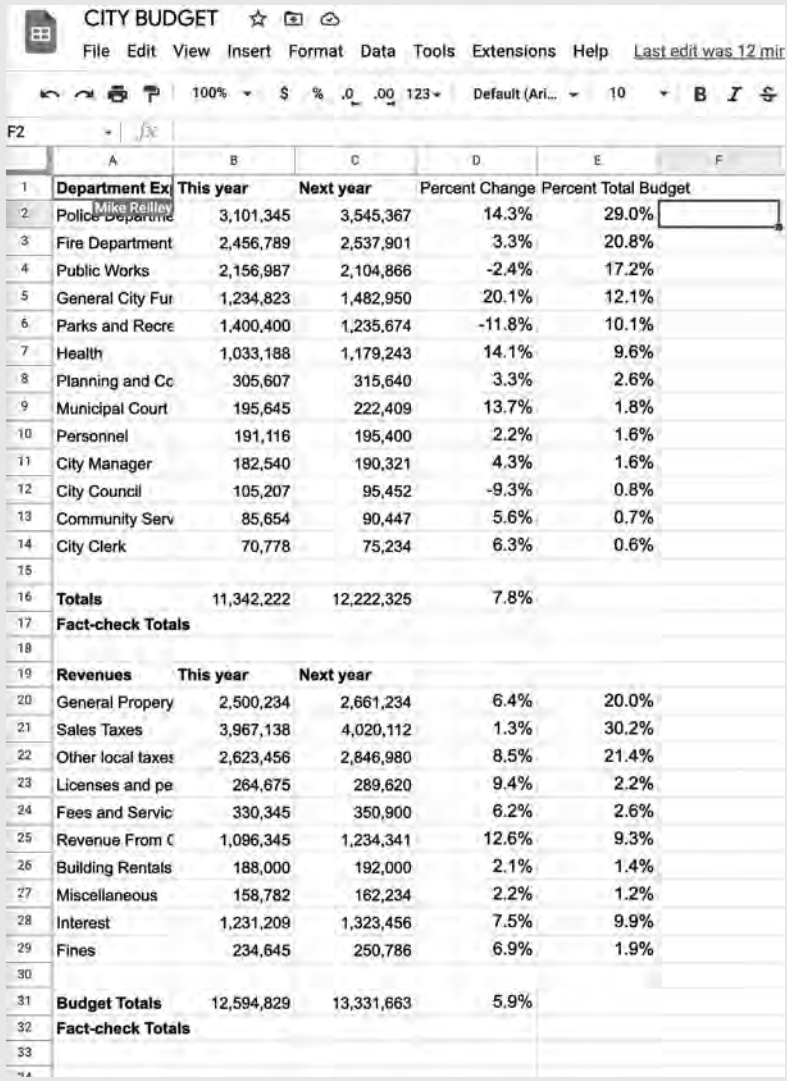
2. Percentage Total Budget column

In order to figure the percentage of total budget each department gets, we have to anchor the cells in C16 and C31, the totals, to get the correct total.

Start by labeling cell E1: "Percentage Total Budget." In cell E2, paste this formula, and fill the cells under it by dragging the cursor or using autofill. Be sure to stop at E14, the end of the expenses section: =c2/c16

In cell E20, paste this formula, and repeat the steps above that you used to fill the expenses section: =c20/c31

Your spreadsheet should look like this (Figure 5.14):



	A	B	C	D	E	F
1	**Department Ex**	**This year**	**Next year**	**Percent Change**	**Percent Total Budget**	
2	Police Departme	3,101,345	3,545,367	14.3%	29.0%	
3	Fire Department	2,456,789	2,537,901	3.3%	20.8%	
4	Public Works	2,156,987	2,104,866	-2.4%	17.2%	
5	General City Fur	1,234,823	1,482,950	20.1%	12.1%	
6	Parks and Recre	1,400,400	1,235,674	-11.8%	10.1%	
7	Health	1,033,188	1,179,243	14.1%	9.6%	
8	Planning and Cc	305,607	315,640	3.3%	2.6%	
9	Municipal Court	195,645	222,409	13.7%	1.8%	
10	Personnel	191,116	195,400	2.2%	1.6%	
11	City Manager	182,540	190,321	4.3%	1.6%	
12	City Council	105,207	95,452	-9.3%	0.8%	
13	Community Serv	85,654	90,447	5.6%	0.7%	
14	City Clerk	70,778	75,234	6.3%	0.6%	
15						
16	**Totals**	11,342,222	12,222,325	7.8%		
17	**Fact-check Totals**					
18						
19	**Revenues**	**This year**	**Next year**			
20	General Propery	2,500,234	2,661,234	6.4%	20.0%	
21	Sales Taxes	3,967,138	4,020,112	1.3%	30.2%	
22	Other local taxes	2,623,456	2,846,980	8.5%	21.4%	
23	Licenses and pe	264,675	289,620	9.4%	2.2%	
24	Fees and Servic	330,345	350,900	6.2%	2.6%	
25	Revenue From C	1,096,345	1,234,341	12.6%	9.3%	
26	Building Rentals	188,000	192,000	2.1%	1.4%	
27	Miscellaneous	158,782	162,234	2.2%	1.2%	
28	Interest	1,231,209	1,323,456	7.5%	9.9%	
29	Fines	234,645	250,786	6.9%	1.9%	
30						
31	**Budget Totals**	12,594,829	13,331,663	5.9%		
32	**Fact-check Totals**					
33						

Figure 5.14 City Budget percentage of total budget column in Google Sheets.

3. Fact-checking budget totals

City officials added a Totals row under each budget to summarize all department expenses and revenues for each year. You always need to fact-check this figure as there can be errors. Use the =SUM formula we used earlier in the chapter to calculate the totals.

In Cell B17, type: =sum(b2:b14)
In Cell C17, type: =sum(c2:c14)

In Cell B32, type: =sum(b20:b29)
In Cell C32, type: =sum(c20:c29)

Do the totals match the totals the city gave you? It looks like only one total was correct. Here's what your sheet should look like (Figure 5.15):

CITY BUDGET

File Edit View Insert Format Data Tools Extensions Help Last edit

100% $ % .0 .00 123 Default (Ari... 10

H14

	A	B	C	D	E
1	**Department Exp**	**This year**	**Next year**	Percent Change	Percent Total Budget
2	Police Departme	3,101,345	3,545,367	14.3%	29.0%
3	Fire Department	2,456,789	2,537,901	3.3%	20.8%
4	Public Works	2,156,987	2,104,866	-2.4%	17.2%
5	General City Fur	1,234,823	1,482,950	20.1%	12.1%
6	Parks and Recre	1,400,400	1,235,674	-11.8%	10.1%
7	Health	1,033,188	1,179,243	14.1%	9.6%
8	Planning and Co	305,607	315,640	3.3%	2.6%
9	Municipal Court	195,645	222,409	13.7%	1.8%
10	Personnel	191,116	195,400	2.2%	1.6%
11	City Manager	182,540	190,321	4.3%	1.6%
12	City Council	105,207	95,452	-9.3%	0.8%
13	Community Serv	85,654	90,447	5.6%	0.7%
14	City Clerk	70,778	75,234	6.3%	0.6%
15					
16	**Totals**	11,342,222	12,222,325	7.8%	
17	**Fact-check Tota**	12,520,079	13,270,904		
18					
19	**Revenues**	**This year**	**Next year**		
20	General Propery	2,500,234	2,661,234	6.4%	20.0%
21	Sales Taxes	3,967,138	4,020,112	1.3%	30.2%
22	Other local taxes	2,623,456	2,846,980	8.5%	21.4%
23	Licenses and pe	264,675	289,620	9.4%	2.2%
24	Fees and Servic	330,345	350,900	6.2%	2.6%
25	Revenue From (	1,096,345	1,234,341	12.6%	9.3%
26	Building Rentals	188,000	192,000	2.1%	1.4%
27	Miscellaneous	158,782	162,234	2.2%	1.2%
28	Interest	1,231,209	1,323,456	7.5%	9.9%
29	Fines	234,645	250,786	6.9%	1.9%
30					
31	**Budget Totals**	11,242,765	13,331,663	18.6%	
32	**Fact-check Tota**	12,594,829	13,331,663		

Figure 5.15 Budget totals a fact-check totals rows on the city budget in Google Sheets.

Budget Challenge Questions

1. You notice that the revenue and expenses totals in the columns don't match the sum of all of the departments. Something is amiss. As a reporter writing this city budget story, what would you do?
2. How would you write the story about this budget? What are the biggest changes or key departments to focus on? Summarize your analysis in a few short paragraphs.

Challenge Answers

1. Don't assume the totals the city gave you are wrong. The error could occur in the individual department numbers. A few typos, anything. A good reporter would contact the city treasurer or other official handling the budget and trace back the error. It could be a reporting error or a typo or something could be amiss.
2. There are a few angles. I would start with finding out why the police and health departments have double-digit expense increases. More hires? A lawsuit settlement? Why the change? And it looks like the parks and recreation department spent nearly 12 percent less than the previous year. Was a major project eliminated? Other cuts?

As for revenues, the big increase came in the "revenue from other sources" category. What were they? And are these revenues expected to carry over into future budgets that the city can count on?

* * *

Resources

Try some of these tools and primers to extend your knowledge of spreadsheets.

Google Sheets vs. Microsoft Excel: The Differences Are Disappearing https://bit.ly/sheetsexcel

Journalists argue this in newsrooms across the country. Sheets are better for collaboration on multi-staff investigative projects. And there are fewer differences between the software compared to just a few years ago.

Data journalism training: Beginner Excel https://sites.google.com/view/mj-basic-data-academy/home

Mary Jo Webster of the Minneapolis Tribune offers a great beginning Excel tutorial.

Journalist's Toolbox Public Records http://bit.ly/jtbpublicrecords

Search dozens of public records and data portals to find spreadsheets to sort, filter and develop into stories.

Investigative Reporters and Editors https://www.ire.org/resources/

IRE offers dozens of case studies, exercises and online training with spreadsheets.

* * *

Footnotes

BBC, Italy Bridge Collapse: Genoa Death Toll Reaches 43 https://bbc.in/3G7ctKe

MSNBC, Almost 1 in 10 US Bridges Needs Repairing https://on.msnbc.com/3IGq8tq

American Road & Transportation Builders Safety Report https://artbabridgereport.org/

National Bridge Inventory – Bridge Inspection Safety Database https://bit.ly/ch5spreadsheet

MediaShift, Take an Interviewing Approach to Find Stories in Data http://mediashift.org/2014/07/take-an-interviewing-approach-to-find-stories-in-data/

KBIA, Columbia Animal Sanctuary Regularly Cited by USDA Inspectors https://www.kbia.org/environment/2016-05-11/columbia-animal-sanctuary-cited-regularly-by-usda-inspectors#stream/0

Bureau of Transportation: County Transportation Statistics https://data.bts.gov/Research-and-Statistics/County-Transportation-Profiles/qdmf-cxm3

The Red Line Project, Thousands of Bridges Graded 'Structurally Deficient' http://redlineproject.org/poorbridges.php

Google Sheets Functions List https://support.google.com/docs/table/25273?hl=en

Journalist's Toolbox Training Video: Determining Rates in Google Sheets https://bit.ly/toolbxorates

Google Sheets: Big Ten School COVID-19 Positivity Rates http://bit.ly/bigtencovid19

The Red Line Project, Saturday Night Football During COVID-19, Less Beer, More Masks 2020 http://redlineproject.org/big_ten_covid.php

6 Advanced Spreadsheets and R

Samantha Sunne

Introduction

So far, we have learned the basics of spreadsheet operations – creating one, establishing data types and filtering for relevant data. In this chapter, we will expand our repertoire of formulas and learn additional point-and-click tools that make analysis in a spreadsheet easy.

Throughout this book, we will continue to use Google Sheets. Sheets differs from Microsoft Excel, probably the world's most famous spreadsheet program, in a few key ways.

First, it is based entirely online, unlike Excel, which only connects to the Internet to sync files across Microsoft products. Google Sheets is free with a Google account, but only a limited version of Excel is free with a Microsoft account.

Excel is a paid desktop app, which means it tends to have more robust tools and processing power. For example, at the time of writing, Excel could handle billions of data points, while Google Sheets could handle up to 10 million.

This doesn't take into account the reality of manipulating millions of rows over an Internet connection. Excel can work offline thanks to its desktop app, while Google Sheets' offline functionality is limited.

There are also free spreadsheet desktop apps, like OpenOffice, though they may also offer limited functionality compared to the big two. If you work with so much data that your spreadsheet program starts to fail or freeze, we will learn about even more powerful programs in Chapter 8.

In the end Google Sheets and Excel have similar functionalities, at least for the average data user, and they become more and more alike as time goes by. Throughout your reporting career, you should utilize whichever program makes the most sense for you.

In this chapter, we will continue to analyze the dataset of COVID-19 cases in Big Ten universities from Chapter 5 (you can download it or make a copy here: http://bit.ly/bigtencovid19).

DOI: 10.4324/9781003273301-7

Referencing Tabs

We learned how to write cell references, like B2 – what about tab references? In Google Sheets, spreadsheet tabs each have a name, even if that name is just something like "Sheet1." If you want to refer to a cell or range within that sheet, you can use an exclamation point (!) between the tab name and cell range, like this: "Sheet1!A2:D10."

If your tab has a name with spaces, it should be wrapped in single quotes, like this: "'Big Ten Daily Covid-19 Cases'!A2:D10."

You most likely will only need to reference tabs if you're running calculations across multiple tabs. But the ability to refer to different tables will come in handy especially when we write SQL queries in Chapter 8.

Formulas

Formulas, also called functions, are one of the key tools in the spreadsheet toolbox. We have already used several of them for scraping, cleaning, fact-checking and analysis.

In addition to the ones we have already learned, these are some of the most common and useful formulas you will encounter (Table 6.1). These are for Google Sheets, but other programs will have similar functions with slightly different names and limitations.

Table 6.1 Common Formulas

AVERAGE()	AVERAGE() returns the average of a list of numbers, also sometimes called a "mean." The program calculates it by tallying the sum of values and dividing it by the number of values in the list. You can use this function on individual cells or a range or both. It will ignore any text values.
MEDIAN()	The median is different from the average in that it returns the "middle" number between the highest and lowest. If there are two equal numbers in the middle, the program will calculate the average of those two. Just like AVERAGE(), the MEDIAN() formula only works on numbers, but it can take a list of ranges or cells. You can learn more about the difference between the median and the average in Chapter 12.
MODE()	The mode is the number that occurs most often in a list. Once again, the MODE() function will only work on numerical values but can take ranges and individual cells. See Chapter 12 for more ideas on why you might want to find the mode in a list.
COUNTA()	The COUNTA() function, as its name suggests, counts values in a range or list. For instance, if your dataset has 1,102 rows, but some are empty, you could use COUNTA(A1:A1102) to see how many actually contain values. Like the other functions, COUNTA() can intake ranges, cells and individual values.

(*Continued*)

Table 6.1 Continued

COUNTIF()	COUNTIF() is a more advanced version of COUNT(): It will only count values meeting a certain condition. For instance, if you wanted to count the values above 10,000 in a list, you could use COUNTIF(A2:A1102,">10000").
	COUNTIF() can count text values. It also takes two parameters: the range to search and the condition to check against.
SUMIF()	SUMIF() is similar in that it adds up the cells meeting the criteria, as opposed to counting them.
	Like COUNTIF(), SUMIF() has two parameters: the range to search and the criteria. Unlike COUNTIF(), it only works on numerical values. You can also use it to calculate sums in one column based on another column.
LEFT()	LEFT() is a simple formula that returns a certain number of characters from the left hand side of a value. For example, if cell A2 has the value "Australia," the formula LEFT(A2,5) would return "Austr".
	It can be useful for cleaning or isolating bits of data, the same as we did with the SPLIT() function.
RIGHT()	RIGHT() extracts a certain number of characters from the right hand side of a value.

Hiding Columns

After writing formulas, it can be helpful to hide the column the formula was working on, to keep your spreadsheet tidy. You can hide rows and columns in most spreadsheet programs by right-clicking on the row or column and selecting "Hide." This is different from deleting and even technically different from filtering.

The program only makes the columns invisible, and you can re-expand them later. In Google Sheets, you would do this by clicking on the small left- and right-pointing arrows in the column bar.

Just like with filters, make sure you don't forget which columns or rows you hid. Often the program will indicate that there are columns hidden by including small arrows.

Concatenate

In Chapter 4, we learned the SPLIT() formula to separate individual vaccines that were listed in the same column. The CONCATENATE() formula is kind of the opposite: It pulls values from different columns and puts them in the same cell.

Let's say we're making a map of COVID-19 cases at Big Ten universities, and we want the city and state to be in the same column. We can do that using the CONCATENATE() formula.

Exercise 1

1. Open your spreadsheet of Big Ten Covid cases from Chapter 5, and make sure you're on the first tab, "Big Ten Daily Covid-19 Cases."
2. Give Column G the name "City and State."
3. In cell G2, type this formula and press enter:
   =CONCATENATE(B2, C2)
   That gives us a confusing result–the city and state are squished together as one long text string. Let's use another feature of CONCATENATE(), which is to specify additional characters or strings to add. The usual way to write a city and state in American English would be "City Name, State Name." We'll make the spreadsheet emulate that by adding ", " in between the two values.
4. In cell G2, edit your formula to add a third parameter, like this:
   =CONCATENATE(B2, ", ", C2)
5. Copy down your formula to apply to the rest of the rows.
6. Click on the letter B to highlight Column B. Hold down the Shift key, and click on the letter C to also highlight Column C.
7. Right-click on the column names and select "Hide columns".

G2 =CONCATENATE(B2, ", ",C2)

	A	D	E	F	G
1	University	Date	Tests	Confirmed Cases	City and State
2	Illinois	7/6/2020	99	0	Champaign, Illinois
3	Illinois	7/7/2020	335	0	Champaign, Illinois
4	Illinois	7/8/2020	933	2	Champaign, Illinois
5	Illinois	7/9/2020	1652	3	Champaign, Illinois
6	Illinois	7/10/2020	2382	8	Champaign, Illinois
7	Illinois	7/11/2020	2382	8	Champaign, Illinois
8	Illinois	7/12/2020	2382	8	Champaign, Illinois
9	Illinois	7/13/2020	3051	10	Champaign, Illinois
10	Illinois	7/14/2020	3679	12	Champaign, Illinois
11	Illinois	7/15/2020	4217	18	Champaign, Illinois
12	Illinois	7/16/2020	4834	25	Champaign, Illinois
13	Illinois	7/17/2020	5559	28	Champaign, Illinois
14	Illinois	7/18/2020	5559	28	Champaign, Illinois
15	Illinois	7/19/2020	5559	28	Champaign, Illinois
16	Illinois	7/20/2020	6399	31	Champaign, Illinois
17	Illinois	7/21/2020	6906	34	Champaign, Illinois
18	Illinois	7/22/2020	7538	37	Champaign, Illinois
19	Illinois	7/23/2020	8290	43	Champaign, Illinois

Figure 6.1 The Concatenate formula.

Now we have a new column combining columns B and C (Figure 6.1). If you were to actually use a mapping tool, you may need to use the Paste Special > Values technique we learned in Chapter 3 to input it in a mapping tool. Paste Special > Values pastes the values as plain text rather than as a formula.

Pro Tip

While typing in your Concatenate formula, Google Sheets may suggest the shorter-named CONCAT() function. This is just a minimized version of concatenate, and it can only combine two values. If you want to add a third element, like the comma and space characters, you will need to use the full CONCATENATE() function.

IF Statements

When you're writing formulas in spreadsheets, and not quite writing your own code yet, IF statements are one of the most powerful tools in your toolbox. It's basically like writing a conditional sentence right into your spreadsheet cell.

Here is the IF formula:

=IF(CELL meets CRITERIA, then THIS, otherwise THAT)

It makes more sense when you see it in action. Using our COVID spreadsheet as an example:

=IF(F2>20, "High case rate", "Low case rate")

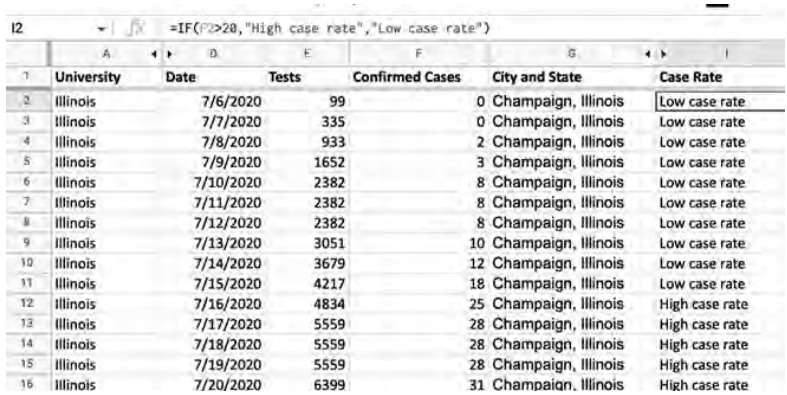
You can read the function from left to right, as you would a sentence. If the number of confirmed cases at a school (F2) is greater than 20, the spreadsheet will print the phrase "High case rate." If it is *not* greater than 20, it will print "Low case rate" (Figure 6.2). And that's it!

You can also write nested IF functions. If you recall from Chapter 3, when we talked about HTML, "nested" means to wrap one function inside another. A nested IF statement might look like this:

=IF(F2=0, "No cases", IF(F2>20, "High case rate", "Low case rate"))

This adds a third parameter. If F2 is zero, the spreadsheet will write "No cases." If F2 is not zero, then the computer will turn to the second IF function: The same one we wrote above. This provides three possibilities for what the cell will populate with, all based on values in Column F.

As you can see, IF statements are a useful tool but are still limited. For instance, what if you wanted more than three possible outcomes? You could technically keep writing nested IF statements within IF statements within IF statements – but that gets messy! Once you get to that point, it's better to use another method.



I2 =IF(F2>20,"High case rate","Low case rate")

	A	D	E	F	G	I
1	University	Date	Tests	Confirmed Cases	City and State	Case Rate
2	Illinois	7/6/2020	99	0	Champaign, Illinois	Low case rate
3	Illinois	7/7/2020	335	0	Champaign, Illinois	Low case rate
4	Illinois	7/8/2020	933	2	Champaign, Illinois	Low case rate
5	Illinois	7/9/2020	1652	3	Champaign, Illinois	Low case rate
6	Illinois	7/10/2020	2382	8	Champaign, Illinois	Low case rate
7	Illinois	7/11/2020	2382	8	Champaign, Illinois	Low case rate
8	Illinois	7/12/2020	2382	8	Champaign, Illinois	Low case rate
9	Illinois	7/13/2020	3051	10	Champaign, Illinois	Low case rate
10	Illinois	7/14/2020	3679	12	Champaign, Illinois	Low case rate
11	Illinois	7/15/2020	4217	18	Champaign, Illinois	Low case rate
12	Illinois	7/16/2020	4834	25	Champaign, Illinois	High case rate
13	Illinois	7/17/2020	5559	28	Champaign, Illinois	High case rate
14	Illinois	7/18/2020	5559	28	Champaign, Illinois	High case rate
15	Illinois	7/19/2020	5559	28	Champaign, Illinois	High case rate
16	Illinois	7/20/2020	6399	31	Champaign, Illinois	High case rate

Figure 6.2 IF statements.

IF Error

One common use of IF statements is to avoid pesky error messages. For example, if you are running a conditional on a cell that is empty, Google Sheets will sometimes populate your cell with "#DIV/0", meaning it tried to run a formula on a nonexistent piece of data.

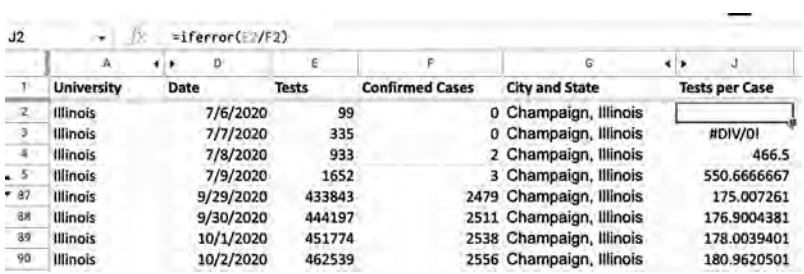
This can be confusing, so you can avoid them altogether using a subset of Google Sheets IF functions called IFERROR(). It looks like this:

=IFERROR(FORMULA is an error, then THIS)

The simplicity of IFERROR() is that you don't actually need to specify an outcome. If there is an error in the formula or calculation, IFERROR() will simply leave the cell blank. If you think your formula is liable to prompt errors, you can wrap it in an IFERROR() formula, like this:

=IFERROR(A2/B2)

For example, if we tried to calculate the number of tests per confirmed cases, the first two rows would return errors. Wrapping that calculation in an IFERROR() formula would replace these errors with simple blank rows (Figure 6.3).



J2 =iferror(E2/F2)

	A	D	E	F	G	J
1	University	Date	Tests	Confirmed Cases	City and State	Tests per Case
2	Illinois	7/6/2020	99	0	Champaign, Illinois	
3	Illinois	7/7/2020	335	0	Champaign, Illinois	#DIV/0!
4	Illinois	7/8/2020	933	2	Champaign, Illinois	466.5
5	Illinois	7/9/2020	1652	3	Champaign, Illinois	550.6666667
87	Illinois	9/29/2020	433843	2479	Champaign, Illinois	175.007261
88	Illinois	9/30/2020	444197	2511	Champaign, Illinois	176.9004381
89	Illinois	10/1/2020	451774	2538	Champaign, Illinois	178.0039401
90	Illinois	10/2/2020	462539	2556	Champaign, Illinois	180.9620501

Figure 6.3 IFError formula.

Nested Functions

Mid

Another useful nested formula to know is the SEARCH() function. As we learned earlier, the LEFT() function pulls a certain number of characters from the left and RIGHT() from the right. But what if we don't know how many characters to pull, or we want to start somewhere in the middle? The MID() function does that. It takes three parameters:

=MID(CELL to search, POSITION to start at, NUMBER OF CHARACTERS to extract)

For example, this formula in our COVID spreadsheet would return the first five letters of the name of the university. We know it would return the first five because we ordered the formula to start at character number 1.

=MID(A2, 1, 5)

Let's say we want to chart the number of COVID cases by month and need a column showing the month that each count was taken in. We could use =LEFT(D2,1) to extract just the first character from the date.

This would give us the number 7 in Row 2, meaning July, but what about October? In Row 89, we need to pull two characters, not one. But if we use =LEFT(D2,2), we get a slash (/) in some rows! How frustrating.

One answer is the SEARCH() function.

Pro Tip

If you don't know what to enter for MID's third parameter, the number of characters to extract, one hack is to give a large number like 100 to extract all characters. But, even if you're reasonably confident that all your data is less than 100 characters, you should scroll through to make sure. While you can make reasonable assumptions as you go along, never publish your assumptions without doing at least a cursory check like we learned in Chapter 4.

Search

The SEARCH() function can be confusing at first, but it is useful.

It returns the position of a character or string in a cell. For instance, if we have a column of expenses, the formula SEARCH("$", A2) would return the number "1" because the expense starts with a dollar sign. If the cell doesn't contain a dollar sign, SEARCH() would return the "#VALUE!" error message.

Like MID(), the SEARCH() function takes three parameters:

=SEARCH(for CHARACTER, in CELL, starting at POSITION)

For example, in our COVID spreadsheet, cell A2 contains the word "Illinois." This formula would return the number 4:

=SEARCH("i", A2,3)

The letter "i" is actually the first letter in the cell, meaning it has position 1. But because the starting position is 3, SEARCH() counts the next "i" after that, which is position 4. If you plan to start at the first character, it is fine to omit the third parameter entirely.

You can see how the SEARCH() function can be powerful, but also a bit confusing. It's best to remember that computers think differently than us, and we are on their turf. If you are running into errors, it also never hurts to use a search engine.

* * *

Exercise 2 

Let's use MID(), SEARCH() and nested functions to make a "Month" column in our Big Ten spreadsheet.

1. First, give Column H the name "Month." In cell H2, we will start with our MID() formula.

   The first two parameters are rather easy: We know the Date is in cell D2, and we know we want to start from the beginning. So our formula looks like this:

   =MID(D2,1,?)

   The third parameter is a question mark (?) because we don't know how many characters to extract. In July, we want one, but in October, two.

   Let's use the SEARCH() function to figure it out. As we know, the SEARCH() function returns the position of a character or string of characters. How can it help us isolate the month? By looking for a slash (/).
2. Give Column I the name "Slash Position." In cell I2, enter this SEARCH() function, and copy it down to the rest of the rows:

   =SEARCH("/", D2,1)

   Cell I2 should populate with the number 2, because the slash was the second character in D2. In cell H89, it should show 3. Progress!

   Now, let's put our SEARCH and MID tools together.

3. In cell H2, replace the question mark with your SEARCH formula, creating a nested formula. Copy it down to apply to the rest of the rows.

   =MID(D2,1,SEARCH("/", D2,1))

   Well–it kind of worked. We can see that H2 returned the number 7, and H89 returned 10, but they also included the slash. That isn't going to look very good on a chart.

   Because the MID formula takes a number as its third parameter (the number of characters to extract), that SEARCH function is actually telling the computer a number. And that number is one too many, as we can see by the long list of numbers containing slash marks after them. Let's remove it with a simple math equation: subtraction.
4. Add the text "-1" to your third parameter to reduce the position by one. It should go after the parentheses creating the SEARCH function but before the parentheses closing the MID function.

   =MID(D2,1,SEARCH("/", D2,1)-1)

   And voila! Copying this rather confusing-looking formula down the rest of our sheet returns a simple number for each month (Figure 6.4). No extra slash marks and no cutoff months, either.

H2 =mid(D2,1,search("/",D2,1)-1)

	A	D	E	F	G	H
1	University	Date	Tests	Confirmed Cases	City and State	Month
2	Illinois	7/6/2020	99	0	Champaign, Illinois	7
3	Illinois	7/7/2020	335	0	Champaign, Illinois	7
4	Illinois	7/8/2020	933	2	Champaign, Illinois	7
5	Illinois	7/9/2020	1652	3	Champaign, Illinois	7
87	Illinois	9/29/2020	433843	2479	Champaign, Illinois	9
88	Illinois	9/30/2020	444197	2511	Champaign, Illinois	9
89	Illinois	10/1/2020	451774	2538	Champaign, Illinois	10
90	Illinois	10/2/2020	462539	2556	Champaign, Illinois	10
91	Illinois	10/3/2020	466390	2565	Champaign, Illinois	10
92	Illinois	10/4/2020	470764	2569	Champaign, Illinois	10
93	Illinois	10/5/2020	481906	2617	Champaign, Illinois	10
94	Illinois	10/6/2020	492275	2642	Champaign, Illinois	10
95	Illinois	10/7/2020	502055	2663	Champaign, Illinois	10

Figure 6.4 The nested Search and Mid functions.

Pro Tip

When working with nested functions, make sure you have the correct number of parentheses. Sometimes you can type too many, or a program will autofill one for you.

You can check by clicking on each parenthesis and making sure it has a pair at either end of the formula. Programs like Google Sheets will also try to help you by highlighting the punctuation marks in bold or different font colors.

These nested functions may look daunting, but as with so many other skills in this book, they will become less so the more you use them. In some cases, the REGEXEXTRACT() formula we learned in Chapter 4 may be a better method for extracting parts of a text string.

Pivot Tables

A pivot table is a fantastic way to organize your data and get straight to the point. Its key function is that it groups together values in order to create a summary. This is similar to the GROUP BY syntax we will learn in Chapter 8 but with the advantage of point-and-click tools.

Grouping values is an extremely common task you will find yourself doing, so it's best to get comfortable with pivot tables now. One hurdle is that the interface for creating one can be confusing, just like the name ("pivot" refers to "pivoting" the rows and columns to make new combinations).

We will learn to make one in Google Sheets, but each program will, of course, have its own buttons and steps.

Transposing

Switching the places of the rows and columns is called Transposing, and it's a useful feature, even if it's somewhat uncommon. Transposing is different from "pivoting," because it does not group any rows together: It simply rotates the entire table by 90 degrees, so to speak.

In Google Sheets, the easiest way to transpose is to highlight all the data, copy it to your clipboard and then, on a new tab, click Paste special > Transposed.

Exercise 3

We're going to find out which months had the most COVID cases, using a pivot table.

1. Create a new pivot table by clicking Insert > Pivot table.
2. In the pop-up window, confirm your data range. It should be your full dataset, consisting of several columns and more than 1,000 rows.
3. Check the radio button for New Sheet and click Create. This will open the pivot table editor in a new tab.

In the pivot table editor, we have four options to work with: Rows, Columns, Values and Filters. Clicking the Add button will create a new instance of that category for *every* instance of that variable. For instance, clicking Add > University in the Columns section would create 14 new columns, because there are 14 universities in the Big Ten Conference (Figure 6.5).

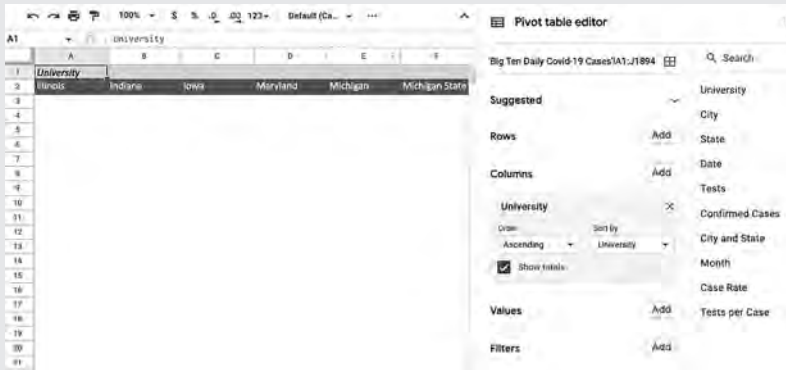


Figure 6.5 The pivot table editor.

So, before creating a pivot table, think: What would my ideal table look like? You can even sketch it out on a napkin. If you end up with far too many columns, or the wrong values in your rows, you can click the X in the variable box to delete it.

Exercise 4

In this case, our ideal table would have a row for every month and a value showing the number of COVID-19 cases that month. Let's make it!

1. In the pivot table editor, click the Add button in the Rows section, and select Month. The pivot table will create a row for every month.
2. The values we want to see are the number of COVID-19 cases. So in the Values section, click Add > Confirmed Cases.
3. Right away, the table creates a SUM of the total COVID-19 cases per month. This is the key to a pivot table.
4. The next step is sorting, which is managed in the Rows section.
5. In the Month variable box, choose the drop-down for Confirmed Cases and the drop-down for Descending.

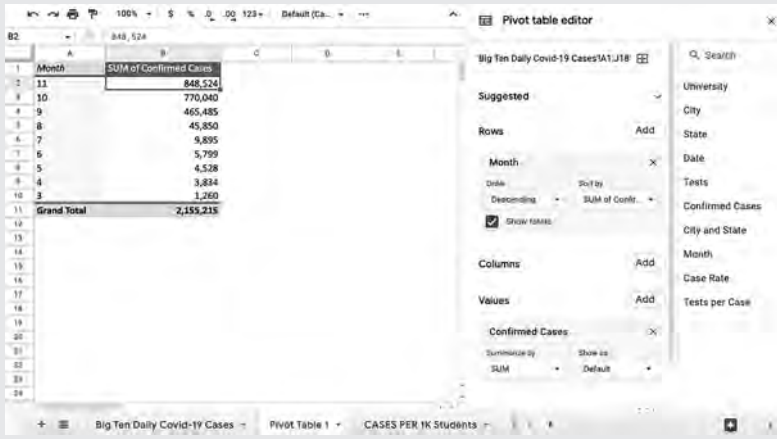


Figure 6.6 The finished pivot table.

Our finished table should have a month in every row, followed by the sum of confirmed cases that month (Figure 6.6). It should be sorted in order of COVID-19 cases, from largest to smallest.

Thanks to this table-of-a-table, we can see that COVID-19 cases were highest in the autumn months.

As with many of the tools addressed in this book, this is only the start of pivot tables. You can choose additional variables in the Row section to create subsections, or drag a column to the Filter section to filter for only rows meeting a certain criteria. You can Count or Average the case numbers instead of Sum them.

Try experimenting to find which school had the most cases or other conclusions you might want to use for your story.

Correlation and Causation

It may be tempting to say that COVID-19 cases spiked due to students returning to campus, or because of the cold weather, but we don't actually know for sure.

This is the difference between correlation and causation. "Correlation" means two values are related, while "causation" means that one *caused* the other. This is notoriously hard to establish with data alone, and it's not ethical to imply causation without doing additional research.

One way around this is to share that cases were highest in the fall and also share opinions from experts that it may be due to students returning to campus. This way, you are accurately conveying your findings, as well as possible reasons why, without overstating them.

Another method is to be clear with your audience that you don't actually have enough evidence of causation, like this: "COVID-19 cases were highest in the fall, but the data did not make it clear why."

If you aren't comfortable enough, you can simply avoid mentioning correlation or causation at all: "COVID-19 cases were highest in September, October and November." You also don't want to imply causation and later turn out to be wrong!

Analysis in R

You can also take your analysis skills to the next level by writing code. We will learn more about how to write and deploy code in Chapter 9, but one of its main advantages is that it can be used for analyzing very large sets of data. This chapter will offer a short introduction to doing spreadsheet-type analysis in a more in-depth program.

R, despite its extremely short name, is the name of a programming language. Like other languages, it can be downloaded to your computer and used to run functions and write scripts. In this case, we will simply use R in a browser-based tool called RStudio Cloud.

The RStudio desktop app is a graphical user interface (GUI) for R, meaning a point-and-click program for writing and running R code. RStudio is free to download and use, but it requires the user to install the R language on their computer, which can lead to other complications (more on this in Chapter 9).

RStudio Cloud avoids this by running all of its work in the cloud. It is not capable of quite as much computing power as a desktop app – let alone straight through a Command Line Interface – but it is more than enough for the purposes of this initial journalistic analysis.

RStudio is meant for all kinds of R coding projects, not just analysis, so its appearance can be customized extensively. Because we are importing data and writing analysis commands, we will start with three main panels, or "panes." These are the Console, Environment and Files panes (Figure 6.7).

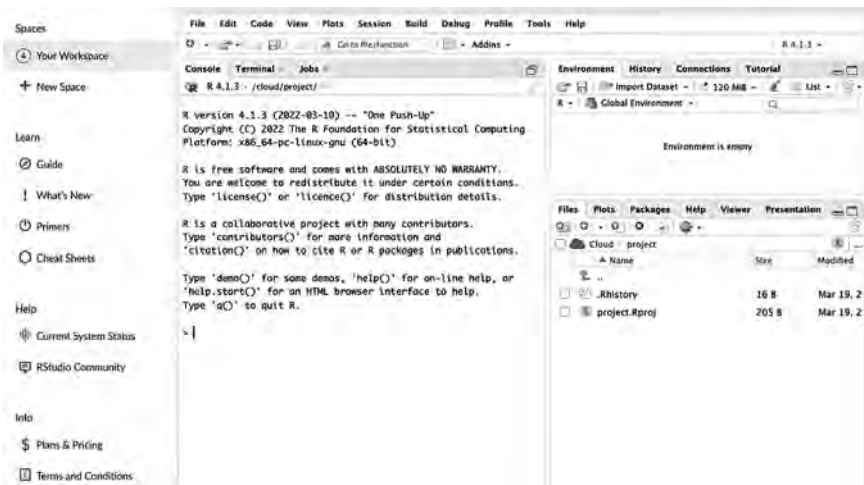


Figure 6.7 The RStudio Cloud panes.

Exercise 5 

We will learn how to do the same tasks we did in Google Sheets, like importing and filtering, in R. First, we will open the GUI and create a new project file.

1. In a web browser, go to RStudio.Cloud and create a free account.
2. Create a new project by clicking New Project > New RStudio Project in the top-right corner of the workspace.
3. In the top toolbar, change the project name to "big_ten_covid." In many programs, but especially in programming, it's best to avoid spaces and instead use underscores (_) in just about everything from functions to file names.
4. In one pane, make sure the Console tab is selected. This is where we will write our commands. Set another pane to display the Environment and another to show Files. The Environment shows the tables we are working with.

You can experiment with other panes and views, like History, which shows your past commands, and Packages, which shows add-on tools you can install.

Importing

Now that we have created an RStudio project, we can begin our analysis.

* * *

Exercise 6

1. Download the COVID-19 spreadsheet (the first tab) as a CSV by clicking File > Download > Comma Separated Values. Once it has downloaded, rename the file to "Big-Ten-Covid-Cases.csv." This name will make it easier to reference in R.
2. Upload the CSV to RStudio Cloud by clicking the "Upload files to server" button in the Files pane. It looks like a white square with an up arrow.
3. Use the upload wizard to select the COVID-19 file. Once uploaded, you should see it listed in your Files panel.
4. Next, import it into your project by clicking File > Import dataset > From text (base).

5. In the import wizard, make sure the data preview looks correct and that the separator is a comma.
6. Click Import. You should now see a preview of your table in a new pane.

When importing, RStudio offers the user several options for specifying their data (Figure 6.8). These include the separator (the delimiter), the character encoding, whether the data uses single or double quotes and whether it contains a header row.

Generally speaking, if you are unsure about importing options, it's best to leave the default selection as is. If problems arise, refer back to Chapter 4 for troubleshooting and common solutions. Using a web search to research your own solution is always an option.

One thing that the import wizard does not address is data types. Remember, these are formats for columns, such as dates, numbers and text. Later on, we will learn how to see and edit these data types.

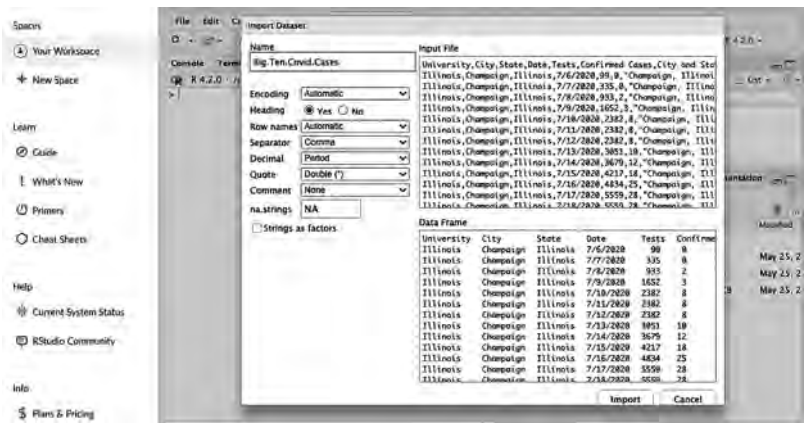


Figure 6.8 The RStudio import wizard.

Summaries

As we said in the Introduction, R can be used for very large datasets. Often, these datasets are too big for a computer to even attempt to display (imagine trying to skim millions of rows!). Instead, these more advanced data programs will display overviews, or summaries, of your data tables.

Now that we have imported our data, let's pull up a summary to make sure that it looks correct.

* * *

Exercise 7

1. In the Console pane, type the command "head(Big.Ten.Covid. Cases)" and press enter. This will return a list of the first few rows of your dataset.
2. Next, type "nrow(Big.Ten.Covid.Cases)" and press enter. This will return the number of rows (Figure 6.9).

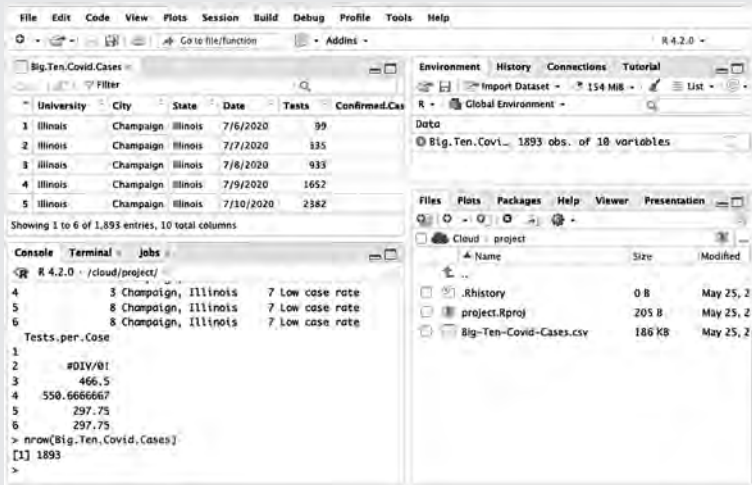


Figure 6.9 R summary functions.

These are called summary functions, and they are used to get a sense of the data you are working with before you start executing commands on it. After importing, you should be thinking critically about your dataset and potential problems it could have. Summary functions help you make sure the row count and other factors seem at least in the ballpark of correct.

They are also useful for the "gut checks" we talked about in Chapter 4. For example, if you open the original CSV in a text editor, does it show the same number of rows as the answer to the nrow() function?

RStudio is a GUI, which means it offers buttons and menus as an alternative to typing out commands. For instance, if you scroll up in your Console pane, you may see the function read.csv(). This is the function that ran when we selected the "Upload files to server" button in the Files pane.

These buttons and drop-down menus may feel more comfortable because they are what we are familiar with, from web browsers to Google Sheets. As you progress in your data journey, you will become more comfortable with writing your own functions and commands.

Packages

In order to do this analysis in RStudio, we have to install an add-on called a "package." These packages are extremely common when working with programming languages, and we will learn more about them in Chapter 9. In the case of RStudio, they can be found in the Packages tab of the Files pane (Figure 6.10).

We'll start by installing a package called tidyverse.

* * *

Exercise 8

1. In the Files pane, navigate to the Packages tab, and click the Install button.
2. In the pop-up window, search for "tidyverse" in the Packages field.
3. Leave all other options on the default and click Install. You will see the Console pane fill with text as RStudio installs the package.
4. Once it has installed, check the radio button next to tidyverse in the Packages tab to load it into your project.

   Now you have the tidyverse package installed and are ready to use the RStudio program to write queries.

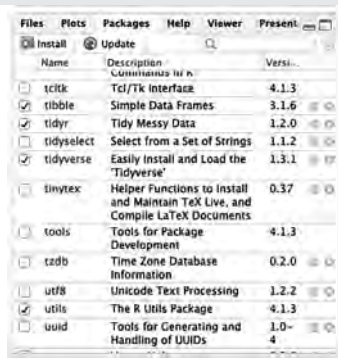


Figure 6.10 The RStudio Packages pane.

Queries

Now let's put our formula skills to use with some analysis queries. One of the most important things to understand from this book is that the same tasks can be done in different programs, to different scales, with much of the same language.

The tidyverse package offers many of the same functions you're used to seeing in other contexts, like Select, Filter and Count. Take this simple R function:

```
filter(Big.Ten.Covid.Cases, "Confirmed.Cases">100)
```

As you may have guessed, thanks to your new data skills, this command is telling RStudio to filter the COVID-19 dataset for rows where there were more than 100 confirmed cases.

There are many options at your fingertips, but we are going to explore one advantage that R has over spreadsheets: the ability to create an entirely new table from a function. This is done by typing the name of a new table, then typing a reverse arrow (<-) and then the conditions for the filter.

```
high_case_days <- filter(Big.Ten.Covid.Cases, "Confirmed.
Cases">100)
```

Running that command will give us a whole new table called high_case_days. This table should appear in your Environment pane alongside your original table (Figure 6.11).

You can then use your new high_case_days table when writing new functions, like this:

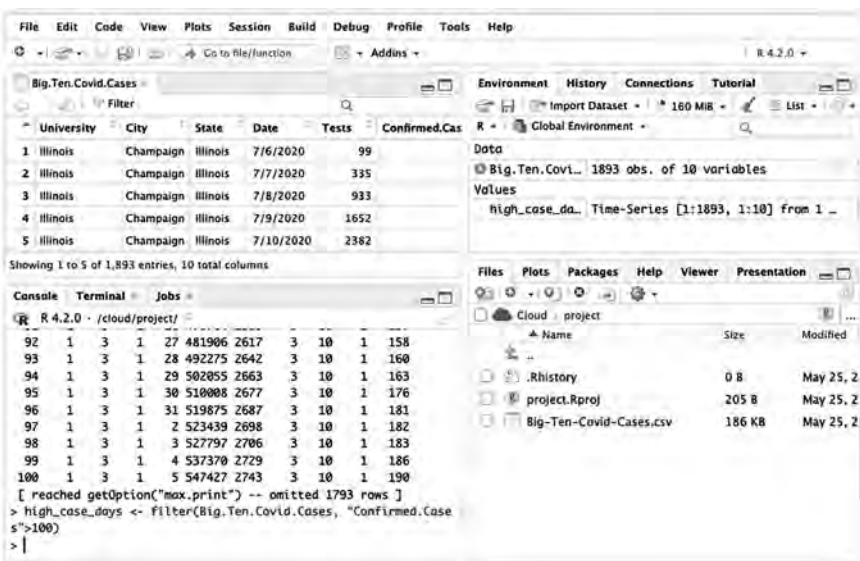


Figure 6.11 Filtering in R.

```
filter(high _ case _ days, "State"=="Illinois")
```

Let's try it using our own COVID-19 table.

Exercise 9

1. In the Console pane of RStudio, create a filter query that would limit your data to a useful subset. For instance, COVID-19 cases at a certain university or low case rate universities in a certain state.
2. Use the commands learned above to create this filter, and assign it to a new table. Give your table a specific name, but do not use spaces.
3. Press enter to execute this query in the Console. You should see your new filtered table appear in the Environment pane.

For more data journalism tips, tricks and exercises, visit the Data + Journalism blog at http://dataplusjournalism.com

* * *

This method – using queries to create entirely new tables – is just a small taste of what R can do. We will explore the powers of programming languages further in Chapter 9.

Footnotes

Google Sheets https://www.google.com/sheets/about/
Microsoft Excel https://www.microsoft.com/en-us/microsoft-365/excel
SQLServer https://www.microsoft.com/en-us/sql-server/sql-server-downloads
OpenOffice https://www.openoffice.org/
Big Ten Daily COVID-19 shortlink http://bit.ly/bigtencovid19
R https://www.r-project.org/
RStudio Cloud https://rstudio.cloud/
RStudio https://www.rstudio.com/
Tidyverse https://www.tidyverse.org/

7 Writing a Data Story

Mike Reilley

Lise Olsen has worked on more than 100 data-driven investigative stories in a career that spans more than 30 years. She said she's most proud of stories that have "helped people in one way or another," and her multipart series over the years have changed laws and inspired reforms in three different states.

At the *Houston Chronicle*, Olsen was part of a team that produced "Abuse of Faith," which identified Southern Baptist preachers, missionaries and other workers who had committed sexual assault or other forms of sex abuse. In many cases, the church moved problematic pastors from one church to another, they found, including a missionary who had abused his own children. The team persuaded many victims who'd never given interviews to talk to them and posted their powerful stories online, along with videos. They also created a national database of convicted offenders. In the aftermath, the Southern Baptist Convention was forced to proceed with reforms.

The team opened the six-part series with a human-centered approach to a very sensitive story:

> *Thirty-five years later, Debbie Vasquez's voice trembled as she described her trauma to a group of Southern Baptist leaders.*
>
> *She was 14, she said, when she was first molested by her pastor in Sanger, a tiny prairie town an hour north of Dallas. It was the first of many assaults that Vasquez said destroyed her teenage years and, at 18, left her pregnant by the Southern Baptist pastor, a married man more than a dozen years older.*
>
> *In June 2008, she paid her way to Indianapolis, where she and others asked leaders of the Southern Baptist Convention and its 47,000 churches to track sexual predators and take action against congregations that harbored or concealed abusers. Vasquez, by then in her 40s, implored them to consider prevention policies like those adopted by faiths that include the Catholic Church.*

In Seattle, Olsen and Lewis Kamb solved cold cases as part of a *Seattle Post-Intelligencer* series that looked into Washington State's missing and unidentified deaths. They built databases from paper records on missing persons from every county in Washington and also researched every unidentified cold case victim statewide at a time when the Green River serial killings remained unsolved. After

DOI: 10.4324/9781003273301-8

posting the series online, several long-unidentified murder victims were identified almost immediately. More than a year later, Olsen got a phone call from Australia that led to the identification of a mother and child murder victim who'd gone unidentified since 1987.

The story again opened with a human-centered anecdotal lead:

> *KELSO – Jai Prasad's tears mix with the cold drip of the day as he stares at an anonymous patch of ground near the crest of a hilltop in the misty Columbia River Valley.*
>
> *He has no proof, yet. But he believes that before him lies the unmarked grave of his youngest sister, a shy Fiji Islands village girl who came to the United States as an immigrant bride, saw the birth of her first child and then disappeared from a cheap downtown apartment in Eugene, Ore., nearly 20 years ago.*
>
> *He utters her name. Raj Mati. He promises to take her home to Fiji. He pushes down his glasses and wipes away his tears.*
>
> *"It really breaks my heart to leave you here alone," he says.*

In Texas, a data-driven series Olsen did in 2020 involved analyzing Superfund sites nationwide and showing which were most vulnerable to climate change. The "Super Threats" series, a collaboration with Inside Climate, NBC and *The Texas Observer*, showed that some particularly dangerous sites, based on the Environmental Protection Agency's (EPA) own data, faced ongoing triple threats from flooding, hurricanes and rising sea levels.

The analysis for this series again spawned multiple investigative stories that often led with a human-centered angle and then moved into the data:

> *BARRETT, Texas – Fred Barrett thought he'd wait out Hurricane Harvey at his home in this town outside Houston, founded by his great-grandfather in 1889. He prepared for heavy rain, wind and flooding.*
>
> *But when the murky brown San Jacinto River jumped its banks, flooding Barrett's neighbors and an ominous cluster of four hazardous waste Superfund sites nearby, Barrett worried the catastrophic 2017 storm could fill his community with deadly toxins.*
>
> *The most notorious of the sites, the San Jacinto Waste Pits, was smashed by 16 feet of water that undermined a concrete cap covering the site's toxic contents, washing dioxin downriver. A dive team from the Environmental Protection Agency later found the potent human carcinogen in river sediment at 2,300 times the agency's standard for cleanup.*

All three of Olsen's projects had immediate and long-term impacts – and also illustrate how posting compelling data on the web can yield leads, sources and spin-off story ideas. That crowdsourcing approach has only become a more powerful tool over the years with the growth of social media.

For more data journalism tips, tricks and exercises, visit the Data + Journalism blog at http://dataplusjournalism.com

Her stories also shared one other common characteristic – crisp, clear writing, combining numbers with a human-driven focus to show how the data impacted *people*. She and many other data journalists are the first to tell you: Great data stories don't *read* like data stories. They are stories about how issues impact human beings.

In this chapter, you'll explore how to focus your data stories with a human-centered narrative approach, how to simplify your writing with just a few numbers per paragraph, how to scale the data to make sense to the reader and how to find key data points that can be buried in reports and databases. Earlier chapters explore how to research, scrape, analyze and draw data points from datasets. Now let's see how to turn it into stories that make a difference.

Summary Paragraphs and Making Numbers Relatable

In December 2016, the *Charleston (West Virginia) Gazette-Mail* reporters wrote a series of stories called "Painkiller Profiteers" on the state's rising number of prescription drugs. In the fifth paragraph of the in-depth story, the reporter, Eric Eyre, wrote a summary paragraph for the ages.

> *In six years, drug wholesalers showered the state with 780 million hydrocodone and oxycodone pills, while 1,728 West Virginia families fatally overdosed on those two painkillers, a Sunday Gazette-Mail investigation found.*
>
> *The unfettered shipments amount to 433 pain pills for every man, woman and child in West Virginia.*

In two sentences, Eyre encapsulated the issue using three key data points, the last of which he calculated by dividing the 780 million pills by the state's population. More importantly, the 433 pain pills per resident brought the story into a crystal-clear focus and scaled the issue in an alarming way.

Eyre's approach is also used in sports infographics. For example, when Kansas City Chiefs quarterback Patrick Mahomes signed a 10-year, $503 million contract extension – at the time the largest in the National Football League's (NFL) history – ESPN and CBS Sports built digital graphics to help readers better understand just how much $503 million is.

CBS Sports' infographic based its analysis on his career averages and 17-game regular-season schedule. It focused on five key data points with Mahomes' contract:

- $49,300 per minute
- $83,500 per attempt
- $126,700 per completed pass
- $1.21 million per touchdown pass
- $2.96 million per game

ESPN's infographic spread the contract out over time:

- $50.3 million per year average
- $137,808 per day
- $5,742 per hour
- $96 per minute
- $1.60 per second

Both sets of data help any sports fan getting by on a livable wage relate to the contract. Mahomes makes $1.60 for taking a breath, and $5,742 per hour, more than many people make in a month or two. Mahomes makes more than $49,000 – an annual salary for some – for every *minute* he's on the field.

Helping the reader better understand the data can be a tricky balancing act for reporters, particularly when writing a summary paragraph. As with Eyre's story, a summary paragraph – also known as a "nut graph" as it tells the story in a nutshell – typically appears between paragraphs three to five, depending on the length of the piece. For magazine stories, it can be lower. It typically follows an anecdotal or straight news lead and contains one to three data points spread over two or three sentences.

Human-Centered Reporting: Putting a Face on a Data Story

Cognitive psychologist Jerome Bruner, quoted by *Forbes* in 2015, said readers are 22 times more likely to remember a fact when it has been wrapped in a story. Why? Because stories are memorable. People relate to them, and they inspire emotions and reactions from the reader. We tell stories every day. We hear them. So put the data into the context of a story, and the reader can better understand.

Adopting a narrative structure, a data story typically starts with an anecdote on the current issue, uses data and other anecdotal support that builds up to the key findings and usually ends with a "call to action" that gives the readers choices at the end of the story: Read a related story, how to get help, comment, answer a poll, etc.

A great example of this approach came in a June 2018 *Arizona Republic* story by Arizona State University student journalists Nate Fain, Daniel Perle and Veronica Graff. They wrote an in-depth piece about a group of NFL players' lawsuit against the league over its poor handling of ongoing concussions with current and former players. The story used several data points that weren't presented until the summary paragraphs. Instead, the reporters focused the lead on how the concussions affected former Arizona Cardinals players, using those local anecdotes to set the tone for the story:

> *Mark Maddox is left with fragments of what should be his fondest memories: He's lost the details of a family trip to Disney World, and forgotten most of what happened in the three Super Bowls he played in. Sometimes, he watches video of his own games to help jog a memory impaired by too many hits to the head.*
>
> *Tyronne Stowe once anchored the Cardinals' defense. Nowadays, he often forgets where he's going when he's driving on Arizona highways. He forgets that*

he's babysitting his grandchild. Stowe recently covered the walls of his office with photos to remind himself of his days as a hard-hitting NFL linebacker who for years went head-to-head with opponents – literally.

Derek Kennard wears hearing aids, thanks to his history of concussions. He suffers from anxiety attacks when he's in any crowded room and has to leave. Just two years ago, Kennard lost his last job as a guidance counselor at Grand Canyon University because he couldn't remember his duties.

Once that issue was established, it was time to introduce the data and show how widespread the issue was. How many other players were impacted? How big is the lawsuit? Following the lead were three paragraphs summarizing the broader issue and scale of the suit:

To most Americans, even to most football fans, the 2011 lawsuit against the National Football League for concussion-related injuries is all about numbers – 5,000 former players, a $1 billion initial settlement, scores of lawyers.

But for the nine former Cardinals players interviewed for this article, those numbers are irrelevant. What matters most is whether, or when, or how soon memory loss might become dementia, then a death too early for men once idolized for their physical prowess.

All told, there are 157 former Cardinals who played for the team since it moved to Arizona in 1988 who are among the 5,000 men who joined in the lawsuit, according to a database analysis of the lawsuit's plaintiffs and all former Cardinals players. Those 157 represent 21 percent of everyone who's played for the team since 1988. Another 109 players from the franchise's days in St. Louis also joined the suit.

Their story followed a basic structure common in journalism called "The Hero's Journey" (Figure 7.1), where the protagonist(s) faces a challenge, goes to resolve it and then returns to normalcy or at least a satisfying end in most cases.

Olsen's aforementioned articles as well as the *Arizona Republic*'s NFL concussions story follow the "Hero's Journey" narrative. For instance, in the NFL story, the heroes are the players suing the league. Their challenges are the fallout from the concussions. We see atonement as they continue on the journey with the lawsuit (which the NFL later settled with the players), and the story is eventually brought back full circle to end with the players. While this is not a "one-size-fits-all" model for writing data stories, it's a common approach for when reporters are trying to show how the issue impacts people while also weaving the data into the story.

There's a running joke among data journalists not to make the reader "numb" with numbers – don't overwhelm them with data. While data can be an interesting way to tell the story, journalism is most often about those being impacted by the data: People. And who's reading the stories? People.

"We connect with those we can relate to," said Andy Boyle, former director of product engineering at the *Chicago Sun-Times*.

Figure 7.1 Structuring a data story in narrative form. (Illustration/Billy O'Keefe).

"It's harder to relate to an opening paragraph showing that a data analysis found 80% of a certain group of people were impacted by something nefarious. But if you instead open it with a *person* who's been impacted by this nefarious thing, as an example of the larger issue, the reader can now connect with this person's struggle."

Boyle's next step: Explain what the data you found shows. Then show more people from this community. Then you explain the choices that were made that led to this situation. And hopefully, you can explain what some of the potential situations are.

Olsen's approach is to tell "human stories above all." Instead of using a lot of numbers, a good reporter will know exactly how to interpret and describe the data, she said. A reporter can use numbers to describe and bolster investigative findings. It's also important for reporters to provide the context and importance (or limitations) of the numbers in clear and compelling explanatory or declarative sentences.

Usually, the fewer numbers you can use, the better. Choose the numbers that will have the most impact, and then use the others in graphics or charts. Use

the numbers to write – and to find people and examples that best illustrate the problem. Humans, and how we deal with one another, are almost always the real part of the story, Boyle said.

"Behind most data analyses are human decisions that led to whatever you found," he said. "That's a policy decision. Or it's a lack of enforcement of something. Or people turning a blind eye. Or just deciding they don't give a damn. They may *say* they give a damn, but if the data shows otherwise, who cares what they say?

"When you're writing about something using data, always find people representative of that data. That'll help make your story not only better, but more relatable."

Olsen uses the interviewing data approach to building stories. Ask questions of the data the same you would a source: Who, what, when, where, how and how much. Then use those queries – or the analysis of a spreadsheet – to gain valuable information. Ask the data good questions. Then, let the "responses" and the numbers you get to your questions guide you to make good decisions about whom to interview or who is typical of the trend you're trying to describe and the additional questions to ask.

For example, when Olsen analyzed deaths in oil fields nationwide for the *Houston Chronicle*, she could say with confidence that Texas workers had the highest death rate of any state at the time she did her research – and that the risks seemed to be rising in the drilling boom. She also could say that nationwide, Nabors Drilling had by far the most overall deaths even compared to other large drillers. And because she had done her homework, she could write stories about workers with that important context. Those conclusions – from the data – shaped the stories that she did in that project, Peril in the Oil Patch.

She picked Texas workers and Texas families to profile in writing the story for Houston and Texas audiences. But that's how a California oil worker's allegations of a cover-up of a nearly fatal accident by Nabors became part of the series – Olsen had the important context that Nabors was the nation's leader in drilling deaths.

Seven Common Angles for Data Stories

Paul Bradshaw's Online Journalism Blog outlines seven common angles for data journalism stories. Consider these when you're starting to research a data-driven story and when you start writing it.

1. **Scale:** How big is an issue?
2. **Change/stasis:** This is going up/down/not improving.
3. **Outliers/ranking:** The best/worst/where we rank on a list.
4. **Variation:** Postcode lotteries and distributions.
5. **Exploration:** Tools, simulators, analysis – and art.
6. **Relationships/debunking:** How are things connected? Or are they not connected? Track networks and flows of power and money.
7. **Problems and solutions:** Concerns over data, missing data, etc.

Writing Data Stories on a Beat

In 2021, Cherone wrote a story for the web and reported for TV a story about City Council committee spending overseen by Alderman Carrie Austin (34th Ward), who was then under indictment.

Cherone's reporting relied on data from Chicago's annual comprehensive financial report, which is a dense 236-page document. To begin the process, she looked at the actual spending by each of the City Council's 19 committees in 2020 and compared that to what each committee was authorized to spend by the 2020 city budget. Then she compared the actual spending of each committee with the spending by Austin's committee.

That data showed the disparity clearly, which meant she could start talking to committee chairs and other sources, like Mayor Lori Lightfoot and Austin. Eleven days later, Austin resigned as committee chair. The fact that Austin's committee spent so much more than other committees and did so much less than all of them is newsworthy on its own. But the fact that Austin was also under indictment for bribery is the key.

Cherone opened her story with a hard news lead:

> *The Chicago City Council committee led by indicted Ald. Carrie Austin (34th Ward) spent $191,500 in 2020, while meeting just three times without advancing a single piece of substantive legislation or pressing officials on how the city can do a better job ensuring lucrative contracts can benefit firms owned by women or Black, Latino or Asian Chicagoans.*
>
> *More than 45 days after Austin was indicted on charges of bribery and lying to federal officials, Mayor Lori Lightfoot, who picked Austin to lead the Committee on Contracting Oversight and Equity, has yet to call for Austin to relinquish her position.*

Note that in the lead, Cherone focused on the money – taxpayer dollars – while reporting that the committee rarely met and didn't accomplish much. By doing so, she followed one of her key rules for writing data-driven stories: Isolate numbers in short, tight sentences and paragraphs. She typically uses just one statistic per sentence and just one or two sentences per paragraph.

"For example, Chicago's police budget is $1.9 billion, but at one point was $1.7 billion, so I'm either going to use those two numbers or I'm simply going to calculate the percentage change and what it changed to," she said. "It's essentially the same statistic, but it's just presented in a way that gives people the right context.

"I think what is problematic is if in that one sentence you're saying the police budget went up 3 percent of the city's total budgeted $16.7 billion, then you're dividing the reader's attention in a sense."

When writing for the web, specifically mobile, a paragraph should never be longer than three sentences in order to maintain clarity and readability on those devices. Spreading the data throughout the story and balancing it with human-centered anecdotes, quotes and context help increase reader understanding of the issue, Cherone said.

Declutter Your Data Stories

Numbers can be messy and hard to read in small type. Here are some ways to declutter your writing by simplifying numbers and presenting them in a concise way:

- 29,912 can be rounded up: Nearly 30,000.
- Round off percentages, too: 62.2 percent is 62 percent; 79.9 percent is 80 percent; 75 percent is three-fourths or three in four.
- 12.55 percent of women and 25 percent of men can be one in eight women and one in four men.
- Instead of a 100 percent increase, say it doubled. Instead of a 200 percent increase, say it tripled. It's easier for the reader to understand.
- Break data into bullet-point lists and pullout boxes like this.
- Could the data be better presented in a chart, infographic or database? Does it *need* to be written?

Source: Working Safely with Statistics, presentation by Jennifer LaFleur and Holly Hacker, NICAR 2022 Conference

"Many readers' eyes glaze over at the sight of numbers, and they are very likely to stop reading and head over to TikTok and never return," she said. "Talking about a person rather than a percentage can make an issue come alive for readers, and make it clear what is at stake."

John Walton, data journalism editor at BBC News, said one pitfall of data-driven stories is that reporters assume that the reader loves the data as much as the "nerdy journalist who's spent hours and hours analyzing, cleaning and visualizing it.

"It is also easy for the data journalist to fall into the habit of assuming that the audience knows the subject being written as well as the data journalist does," Walton said. "It's much more likely that they don't, and they are also likely to have a lower tolerance for reading number after number in a news report too."

Use numbers sparingly if you can, it's not always easy, but remember it's your job to digest and simplify your work for the reader, and to communicate it succinctly.

One way to keep readers' interest from figure fatigue is to make the numbers relevant or, to put a face on them. Make the inhuman figures relevant in human terms and the reader is more likely to stay with you.

One of Walton's tricks of the trade: Boil large budget figures down to figures per person. What does the 3 percent city sales tax increase mean to the taxpayers?

"Even better, if you're able to make the data relevant to people by their age, income, location, gender or ethnicity then this is going to have much more impact and relevance to them than a simple national average figure might," he said.

Reporting and Writing for a Broadcast Audience

As she did with the Austin story, Cherone sometimes has to produce broadcast stories for WTTW to complement her online pieces. With only three minutes to work with as opposed to thousands of words, she must shift her approach to a more conversational tone and be more selective in the data she shares and explains on-air. A short bullet list of key stats may flash on the screen as she discusses the story with the news anchor.

"I know the questions that the anchor is going to ask me so that I can be sure that we're heading in the right direction," she said. "Many times, you can only tell them why it matters and maybe one important supplementary piece of information and then that's it. It's frustrating but it is just the nature of TV."

Cherone said she believes data can point journalists to the most compelling and illustrative people to interview for an article, and a data-infused summary graph can give the story authority and credibility. It's important to know how to integrate data into a story and how to characterize and present numbers and statistics in ways that readers can readily understand. The general rule of thumb: Journalists should use only the most salient numbers in their story and allow readers to drill deeper by posting data online and creating graphics.

Data journalism gives reporters the opportunity to "zoom out" and look at an issue on a larger scale. For instance, you notice more potholes showing up on the roads in your neighborhood – this is a potential story idea. You talk to some people whose cars were damaged by the potholes. You talk to some area tire and auto repair shops and, sure enough, business is booming this year.

That makes for a good *start* to a story, but you need data to give the readers the entire picture. Enter data reporting. Most likely, your city, county or province tracks pothole complaints and repairs. Pull the dataset, clean it and start sorting and filtering. Ask questions of the data: Which ZIP code, neighborhood or area has the most potholes? Which has the least? Are certain areas of the city slow to be repaired? If so, why? Compare this year's pothole data to past years. Is there an increase or decrease?

Cherone used this "zoom out" approach when she wrote a WTTW News story on Illinois' ban on evictions during the COVID-19 pandemic in November 2021. Her first challenge came with finding the data. She contacted the Chicago Department of Housing, which said it didn't maintain data on evictions. She eventually reached the Office of Cook County Chief Judge Tim Evans, which took seven days to get her the data.

"Then it was a straightforward matter of comparing the number of evictions in October 2021 with the number of evictions in October 2020 and October 2019," she said.

"It was important to have [two] years of data to have a point of comparison before the COVID-19 pandemic; while stay-at-home orders and the ban on evictions designed to stop the spread of COVID-19 were in place; and after the ban and stay-at-home orders had been lifted."

The analysis broadened the scope of the story both in scale and time. She wrote:

> *Evictions rebounded significantly in October 2021 as compared with October 2020, when the restrictions were in place as the second wave of COVID-19 swept Chicago and Illinois. Cook County judges approved only 322 evictions in October 2020, as compared with 1,866 in October 2021, according to the data*
>
> *In Chicago, 1,278 households were evicted from their homes in October 2021, along with 42 businesses, according to the data. In suburban Cook County, an additional 566 households were evicted, along with 22 businesses, according to the data.*

* * *

Alvin Chang, head of data and visuals at *The Guardian US*, said data is "merely a structured collection of stories."

"So that means you can do two things with it: You can look at the dataset as a whole and see the big trends. But you can also zoom into one data point and learn more about it. If it's a dataset of people, you can call them; if it's a dataset of places, you can visit a location."

Beyond the data analysis, you can use the data to visualize the problem. A simple bar chart would track the number of pothole repairs and complaints annually over a five-year stretch. This approach will be covered in-depth in Chapter 10, but a map of those complaints and repairs, layered over a neighborhood or ZIP code shapefile, helps readers visualize patterns (Figure 7.2) that would be difficult to see in a written piece.

Chang said there's another dimension that reporters can forget about: The context. When reporters are neck-deep in a dataset, it's easy to forget that this isn't the only information available. There are other datasets, other articles, other studies that can contribute to our understanding of the story. This is true for writing and visualizing a story, which we'll explore more in Chapter 10.

"They can help us explain *why* something is happening in the data," he said.

"A good example is visualizing racial segregation in neighborhoods, which can lead people to assume that there was some kind of natural sorting of racial groups. But understanding the larger context will reveal that residential segregation was an engineered phenomena born out of racist housing policies – and missing that story would be journalistic malpractice."

* * *

Figure 7.2 A layered map of 2014 Chicago pothole repairs by neighborhood. The map was built in Google MyMaps.

Writing Exercise

In Chapter 5, you analyzed the 2019 Federal Highway Administration National Bridge Inventory database. You figured the percentage of good, fair and fair-poor graded bridges per state and sorted them to see where various states rank.

Remember the questions you answered in the Chapter 5 exercises about states with the most and least in each category. Where does your state rank? Using the data points from that Chapter 5 analysis, write a three-to-four paragraph story for a local audience. Use a hard news lead focusing on just one key finding from your analysis, and then expand to more detail. Focus your lead there, and then support with a few paragraphs analyzing the high/low totals. There's an example for Illinois at the end of this chapter.

Journalism students at the University of Illinois at Chicago turned this assignment (using the 2017 database) into final project stories examining the conditions of Illinois and, more specifically, Illinois bridges. They compiled the percentages, sorted and filtered to find where Illinois ranked nationally for fair and poor bridges and then analyzed other studies to map where many of the troubled bridges are in the state.

The stories came with an interesting twist: The week after the first story was published on RedLineProject.org in December 2018, a bridge over Lake Shore Drive at the Chicago River, one of the city's busiest and ranked "structurally

deficient," buckled and shut down northbound lanes for two days. This snarled traffic and proved that, to play off a cliche, the numbers don't lie.

* * *

Writing Exercise Example

Illinois ranked 34th nationally in the number of bridges rated in fair-to-poor condition, according to an analysis of the 2019 Federal Highway Administration National Bridge Inventory database.

The data showed that 51.2 percent of the state's 26,825 bridges were in fair-poor shape. The study also showed that 9 percent were rated in poor condition.

Rhode Island led all states with 82 percent of its bridges in fair-poor condition, but the smaller state also had only 779 bridges. The District of Columbia (80 percent) and West Virginia (75.4 percent) weren't far behind.

Florida had the fewest fair-poor rated bridges with 33.9 percent, followed by Mississippi with 37.2 percent.

* * *

Footnotes

Houston Chronicle, Abuse of Faith https://www.houstonchronicle.com/local/investigations/abuse-of-faith/

Seattle Post-Intelligencer, Unmarked Graves May Hold His Sister, Niece https://www.seattlepi.com/local/article/Unmarked-graves-may-hold-his-sister-niece-1213531.php

Inside Climate News, Superfund, Super Threats https://insideclimatenews.org/project/super-threats/

Charleston (West Virginia) Gazette-Mail, Painkiller Profiteers https://www.wvgazettemail.com/news/legal_affairs/drug-firms-poured-780m-painkillers-into-wv-amid-rise-of-overdoses/article_99026dad-8ed5-5075-90fa-adb906a36214.html

Working Safely With Statistics, presentation by Jennifer LaFleur and Holly Hacker, NICAR 2022 Conference

Forbes, A Good Presentation Is about Data and a Story https://www.forbes.com/sites/kateharrison/2015/01/20/a-good-presentation-is-about-data-and-story/?sh=4360736b450f

Arizona Republic, AZCentral.com, Former Arizona Cardinals Players among Thousands in NFL Concussion Lawsuit https://www.azcentral.com/story/sports/nfl/cardinals/2018/06/09/former-arizona-cardinals-players-among-thousands-nfl-concussion-suit/592331002/

DataCamp, Seven Tricks for Better Data Storytelling https://www.datacamp.com/blog/seven-tricks-for-better-data-storytelling-part-ii

Houston Chronicle, Peril in the Oilpatch https://www.houstonchronicle.com/local/peril-in-the-oil-patch/

Online Journalism Blog https://onlinejournalismblog.com/

WTTW, City Council Committee Led by Indicted Ald. Austin Spends More, Does Less Than Nearly All Others https://news.wttw.com/2021/08/16/city-council-committee-led-indicted-ald-austin-spends-more-does-less-nearly-all-others

WTTW, Evictions Jump after Ban Ends, but Tsunami Fails to Materialize in Chicago, Cook County: Data https://news.wttw.com/2021/11/23/evictions-jump-after-ban-ends-tsunami-fails-materialize-chicago-cook-county-data

The Red Line Project, Thousands of Illinois Bridges Graded Structurally Deficient http://redlineproject.org/poorbridges.php

8 SQL

Samantha Sunne

Spreadsheet skills like sorting, filtering and formulas can get you very far in journalism. Some data journalists even say that 80 percent of data journalism can be done with these initial skills. But as you begin to work with larger and more complex datasets, you may want to use a more advanced method like SQL.

SQL stands for "structured query language," and it's a syntax that can be used in many different programs. It's an especially popular option for people who need to join multiple tables together or handle datasets so big they would make a spreadsheet program break down.

Technically, spreadsheet programs like Google Sheets and Microsoft Excel can process SQL, but it's not what they are designed to do. Excel can run SQL queries by connecting to other software programs, and Google Sheets has a powerful QUERY() function.

But just because you can work with millions of rows in Google Sheets doesn't mean you should! You will find your functions stalling and your screens freezing. Spreadsheet programs are designed to handle smaller tables, even if "small" in this case means a few million rows.

* * *

Pro Tip 

How is SQL pronounced? In conversation, it's usually referred to as "sequel" but can also be spelled out as S-Q-L. There are a variety of programs that use SQL in their name. And just like different dialects of the same language can exist in the same country, the preferred pronunciation of some tech terms can also vary depending on who you ask.

Both the programs and the syntax can be referred to either way. But the more you use these particular programs, the more you will become familiar with the lingo, making you sound more like an experienced data journalist.

* * *

DOI: 10.4324/9781003273301-9

To trade up, many people use programming languages like Python or desktop applications called "database managers," both of which can use SQL. To learn how it works, we will practice a story on the wealthiest cities in California.

We'll start with a dataset (shortlink: https://bit.ly/incometable) that shows the population and median household income for every ZIP code in California. It comes from the 2010–2014 American Community Survey, a five-year demographics collection program by the US Census Bureau.

Unfortunately, the dataset records the ZIP code and the population in the same column, separated by a slash (/). It also doesn't list the names of places or towns, just their ZIP codes. Most journalism audiences will want to see the name of the town.

We will use SQL to clean, organize and analyze this data and ultimately merge it with another dataset to find a story on the richest locales in the Golden State.

Queries

When working with SQL, the keyword is in its middle name: Query.

In technical terms, a SQL query is a complete instruction in SQL syntax that can be executed by a computer. They are also called "SQL statements." In conversational terms, a query is like a question asked to a computer rather than a human being.

SQL is made up of individual keywords, commands and operators (Table 8.1). To run queries, you'll need to understand a few of these keywords, but just like with sorting and filtering in spreadsheets, you can dive deep into a database with just a few basic commands.

For example, to ask, "What was the population of Istanbul in 2006?" you might write this query:

```
SELECT population
FROM istanbul
WHERE year = 2006
```

The words at the beginning of each line – SELECT, FROM and WHERE – are called "commands." Each line is a "clause," and the equal sign is an "operator." All of these fit under the umbrella of SQL "keywords," and together they form a SQL query. Some programs, but not all, require a query to end in a semicolon (;).

Importing

SQL is a common language found in many programs and even programming languages across the digital world. So how, and where, should you use it?

It's safest to write SQL in programs that can save your queries, outputs and tables so that you don't lose them, or lose track of what you've done. (This is one reason to keep data diaries, which we learned about in Chapter 4.)

Table 8.1 Common SQL Keywords

Commands	
SELECT	SELECT tells the computer which data to retrieve from a table. You can request all columns, just a few columns, or limit it to criteria like the ones below. The columns will appear in the order they are entered in the SELECT command, separated by commas. To run a calculation on your data, like SUM, you still need to start your query with the SELECT command. You can think of it as asking the computer to "show you" the sum.
FROM	FROM tells the computer which table you are retrieving data from. In the example above, the name of the table is "istanbul." The FROM command will be especially important to understand when we start to retrieve data from multiple datasets in the same query. This is one of the features that makes SQL stand out from spreadsheets.
WHERE	The WHERE command functions similarly to its use in English. You can use it as a filter so that the query only returns data meeting certain criteria. In the WHERE clause above, we only want to see the population where the year is "2006." The WHERE clause must always be placed after the FROM clause, but before some other commands, like ORDER BY.
ORDER BY	ORDER BY is equivalent to sorting in a spreadsheet. You can use it to arrange rows alphabetically, chronologically, numerically and so on. The ORDER BY clause will default to ascending order. If you want the opposite (large to small), you can specify that by adding the DESC keyword.
GROUP BY	GROUP BY is a little more complex. This command groups together rows that have the same value in one of the table's columns, similar to the pivot tables we used in Chapter 6. This command is especially useful for calculating totals, such as the total population per state, or finding unique values in a column.
HAVING	HAVING is a command that is almost identical to the WHERE command. The difference is that HAVING can filter groups, while WHERE can only filter rows. If you use the GROUP BY command to create groups of rows and want to filter them by a certain condition, you will need to use a HAVING clause.
LIMIT	This command limits the number of rows to return. It can be helpful when working with large datasets, or if you know your query will return too many rows for you to look at comfortably. For example, the command "LIMIT 10" would return only the first ten rows matching the criteria in your query.
OFFSET	OFFSET is a command that begins a query at a row other than the default. OFFSET is mostly used for declaring a header row.
DISTINCT	DISTINCT is not technically a command itself, but it is an argument you can add to a SELECT command. It limits the output to unique values within a column. This tool is especially useful if you can't see the whole dataset and want to know all the values in a column. In a spreadsheet program like Google Sheets, it is similar to the Values that appear in the Filter drop-down menu for a column.

(*Continued*)

Table 8.1 Continued

Functions	
COUNT()	COUNT() is a mathematical function in SQL that counts the number of rows meeting certain criteria. COUNT can be used for numerical values as well as text. For instance, you could count the number of rows containing the word "istanbul." This can be especially helpful for data integrity checks. To find the number of rows in your dataset, you could use COUNT(*), meaning "count all."
SUM()	SUM is another mathematical function. Just like its spreadsheet counterpart, it returns the sum of the values you enter as parameters. SUM will only work on numerical values, not text, and null values (empty cells) will be considered 0.

One option is a database manager, like SQL Server, MySQL or Microsoft Access. These programs are very popular among companies that record and analyze huge amounts of information, like sales databases. They are powerful but can be expensive.

There are some free options, like DB Browser, MySQL Workbench and SQLite Manager. And beyond that, there are some hybrid programs for working with data, like Tableau and Microsoft Power BI, that can perform both spreadsheet and SQL functions.

You can also use SQL in programming languages like Python and R. RStudio, which we used in Chapter 6, is capable of running SQL queries and has a relatively easy-to-use interface. In Chapter 9, Scraping Social Media, we will tackle installing and writing code.

All of these are viable options for running SQL queries. The question is, how big is your data, how big is your budget and what are you trying to accomplish?

To avoid having to install a language or GUI, we will use a website called DB Fiddle. DB Fiddle is an online sandbox – a self-contained environment that lets you execute functions and test code, without it affecting other parts of your computer. This allows us to see the outputs of various SQL statements without installing other programs or paying for a more powerful application.

Unlike database managers, DB Fiddle will not store queries or outputs, meaning it is not the best choice for a bigger or more complex analysis. Because it's a sandbox, it also does not have a very sophisticated import system. Instead, we will simply copy-paste our values into the site.

* * *

Exercise 1

1. Download this comma-separated value (CSV) file of incomes in California (shortlink: https://bit.ly/incometable), and open it in a text editor, such as Sublime Text. Do not use a spreadsheet program like Excel, because that can introduce new formatting. Highlight all of the data and copy it to your clipboard.
2. Open DB-Fiddle.com and click the Text to DDL button. This is where we will paste our data.
3. In the pop-up window, name the table "income_data." The name of the table will be very important later, so make sure you name it this exact text string, including the underscore and without quotes around it.
4. Paste the copied values into the Formatted Text field.
5. Click the Preview DDL button to get a glimpse at how your data looks as it is being imported.
6. Click Append to Schema to load it into the sandbox.

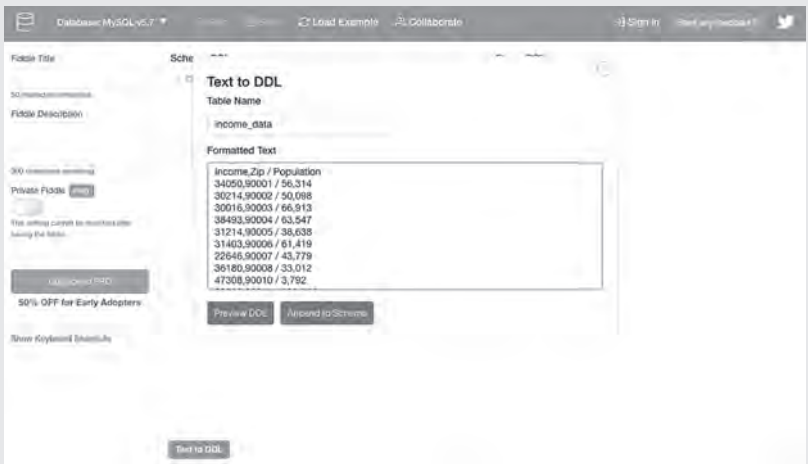


Figure 8.1 Importing via the text to DDL window in DB fiddle.

As you paste in the data, the two fields should be separated by a comma (Figure 8.1). The second column is called "Zip / Population" and contains a slash in the middle of it. DB Fiddle should intake the income column as an integer and the ZIP code column as a "varchar."

These are two data types found in SQL. As we have learned from other programs, it's important for the computer to understand what kind of data is in each column. After all, you can't run a SUM() calculation on a list of names. Only a list of numbers.

SQL data types can sometimes vary by program, but integer and varchar are two of the most common types. An integer, sometimes shortened to Int, is a number.

A varchar datapoint is a text string made of "various characters." A varchar column, just like a text column in a spreadsheet, can contain letters, numbers or even entire phrases, all considered one datapoint. Because it is text, a varchar column should be wrapped in quotes.

Pro Tip 

You may have noticed that DB Fiddle includes more text at the top of the schema panel, including the words CREATE TABLE. With many SQL programs, you will simply import your data without needing to create a table, although typing it manually is an option, with commands like INSERT.

* * *

Filtering

Whenever you want to view the results of a query in the same place you are writing it, you will use the SELECT command. SELECT can also be thought of as a filter, because you are telling the computer which columns or calculations you want to see. Everything else will be hidden.

If you're not ready to filter yet, and want to see all the columns in a table, you can type the following clause to mean "show all."

```
SELECT *
```

The asterisk (*) is a common wildcard in many search syntaxes, meaning it can take the place of other characters.

Even if you are writing a calculation, like COUNT(), you will use the SELECT command to display the output. For example, a good query to keep in your pocket for "gut checks" is to count the null values (empty cells) in a dataset. That query would look something like this:

```
SELECT COUNT(*)
FROM income_data
WHERE income IS NULL
```

In DB Fiddle, we can see that our "Zip / Population" column is wrapped in quotes, meaning it was imported as text. As we saw in Chapter 4, values sometimes need to be wrapped in quotes so that a program doesn't think the slash (or even the space) is a delimiter. The first column, "income," is only one word and so doesn't necessarily need quotes.

Therefore, a SELECT command for both columns would look like this:

```
SELECT Income, 'Zip / Population'
```

Cleaning and importing may not be the most fun part of a data analysis, but it's always important to understand how your program is interpreting and storing your data. Skipping these steps can lead to disaster!

* * *

Pro Tip

If you get error messages right away, quotation marks are usually to blame. If DB Fiddle claims the Zip column can't be found, try copying and pasting the field name (including quotes) from the SQL Schema panel.

Alternately, try deleting the quotation marks in the SQL Schema panel, where we pasted the data, or deleting and retyping them. Fonts can have different types of single quotation marks (opening, closing or neutral), and sometimes software programs will only accept a neutral (not slanted) quotation mark.

* * *

Now that we have imported our data into DB Fiddle, let's practice filtering by writing a query.

DB Fiddle will return the first 10 rows of our income table in a new Results panel but only the income column (Figure 8.2). This is an example of two filters at work: the SELECT command, which retrieves columns, and the LIMIT command, which restricts the number of rows.

Exercise 2

1. In the Query SQL panel of DB Fiddle, type the following SELECT statement:

```
SELECT income
FROM income_data
LIMIT 10
```

2. Click Run at the top of the screen.

Figure 8.2 The simple SELECT statement.

Table 8.2 Common Search Operators

LIKE	The LIKE operator in SQL works similarly to the word "like" in English. It matches partial strings, making it useful for finding data if you don't know exactly what the value is, or if you want to return a collection of similar values. For example, the operator "LIKE 'dog%'" would return rows that contain "dog" or "dogs." It is often used in conjunction with wildcards like the percent sign (%). Without a LIKE operator, a WHERE command for a text string will only return exact matches for that string. If you are using the LIKE operator on a text string, it needs to be wrapped in quotes.
AND	AND adds another condition to a WHERE clause. It is useful for limiting results beyond what you could filter with just one search condition. The AND operator can combine searches for different columns and data types. For example, you could filter for rows where the median household income is above a certain amount, and the ZIP code is a certain number, combining text and numerical filters.
OR	The AND operator narrows a search, but the OR operator widens it. A query containing the OR operator will return any row that matches either condition in a WHERE clause, casting a much wider net. In the example above, the query would return any row with a certain income *or* a certain ZIP code.

Table 8.2 Continued

IS NULL	This book has touched many times on the dangers of empty rows or null values. The method to find these in SQL is the condition IS NULL. It is most often used in a WHERE clause and can be complemented by its opposite, IS NOT NULL.
_	An underscore (_) is a wildcard – a character that stands in for other characters. An underscore will only match a single character. That is, a search for "do_" will return "dog" but not "dogs." It is most useful when searching for values that you know probably exist, like an ID number.
%	A percent sign (%) is a wildcard that will match any number of characters, making it a much broader search option. A search for "d%" would return "dog" as well as "dogs." If you're looking for rows that contain a term anywhere in it, you can enter it between two percent signs, like "%dog%." Importantly, a percent sign will also return values where there is no character in that location, but it will not return null values.
*	An asterisk (*) is a wildcard meaning "all" and is a common sight in SQL statements. For example, "SELECT *" means "select all columns," while "COUNT(*)" means "count all rows."
=	Unlike a wildcard, the equals sign (=) operator identifies a perfect match. This can be used for both strings and numbers. For example, the clause "WHERE breed = 'chihuahua'" would only return rows that correctly spelled the word "chihuahua." If you're not confident in your source's spelling skills, as we saw in Chapter 4, Cleaning Data, you may want to instead use a wildcard, like "WHERE breed LIKE 'chi%'".
!= or < >	The not-equal-to operator is the opposite of the equal sign, meaning it will return any value that does *not* meet the search condition. Some SQL programs will require the user to enter either "!=" or "< >" even though they mean the same thing. In many programs, you can add an exclamation mark (!) in front of a mathematical operator to mean "not." That is, ">5" means "greater than 5," and "!>5" means "not greater than 5."
>, <, >=, <=	These operators hearken back to elementary math and can only be used on numerical values. The angle bracket that broadens away from the value, that is, ">5", means "greater than," while its reverse means "less than." Adding an equal sign to these operators makes them "greater than or equal to" or "less than or equal to."
BETWEEN	The BETWEEN operator saves you some time when limiting results to a range between two values. For example, the clause "WHERE amount BETWEEN 5 AND 10" is the same as "WHERE amount > = 5 AND amount < = 10". The operator is inclusive, meaning it will also return values matching 5 and 10.

Next, let's try filtering with the WHERE command. WHERE is a much more influential filter because it can work with search operators to limit rows to certain criteria (Table 8.2). These searches are referred to as the "WHERE clause."

So far, we have used the SELECT command to limit columns and the LIMIT command to limit rows. Now, let's use the WHERE command to filter our results for something much more interesting.

It's important to use the exact terminology for both commands and criteria. For example, if a table is named "income_data", with an underscore, but the FROM command calls from "income data", with a space, the query would produce an error message or simply not run at all.

Exercise 3

1. In DB Fiddle, change your SELECT clause to an asterisk, meaning "all columns."

```
SELECT *
FROM income _ data
LIMIT 10
```

2. Next, add a WHERE clause so that your final query looks like this:

```
SELECT *
FROM income _ data
WHERE income>40000
LIMIT 10
```

3. Click Run to see a new output in your Results panel. The output should now only show incomes above $40,000.
4. Next, add an AND operator to your WHERE clause, to make the following query:

```
SELECT *
FROM income _ data
WHERE income>40000
      AND income<60000
LIMIT 10
```

5. Click Run. Now, the output is limited to incomes that are both above $40,000 and below $60,000.
6. Lastly, add a LIKE operator to your WHERE clause and click Run.

```
SELECT *
FROM income _ data
WHERE income>40000
```

```
        AND income<60000
        AND `Zip / Population` LIKE '90%'
    LIMIT 10
```

7. Bonus Question: How would you rewrite the query to pull incomes above $100,000?

With the LIKE operator, we need to use quotes (') around the values because they are text strings. It limits the results to ZIP codes starting with "90", while the mathematical operators limit the results to incomes between $40,000 and $60,000 (Figure 8.3).

Figure 8.3 The AND and LIKE operators.

Nested Queries

Another way to filter is to write a query inside another query, just like we wrote functions inside of other functions in a spreadsheet. These are called "nested queries" or "subqueries," and a simple version looks like this:

```
SELECT `Zip / Population`
FROM income_data
WHERE income > (
      SELECT AVG(income)
      FROM income_data)
```

Nested queries can also be used to connect two tables, which we will discuss later in this chapter. However, they can start to get confusing and may not be the best method for filtering down your data.

Some SQL programs are case sensitive – meaning you need to use the correct capitalization for both commands and criteria for it to work. Even if your program isn't case sensitive, the norm is to write your commands in all capital letters. This helps them stand out and is a generally accepted practice, even if a program doesn't necessarily need it to run.

Another general practice is to write each command on its own line, with clauses indented below it. Many programs will happily execute queries all written out in one long line. But giving each clause its own line, and writing the command in all capital letters, makes it easier for the human to read.

We have now limited our results to certain ZIP codes, and to a certain income range, but that's not all that we need to know. We want to find the highest incomes as well as their cities – not just their ZIP code. And unfortunately for us, the ZIP codes are stored in a text string along with the population. So to continue our analysis, we will need to do a bit of cleaning.

Cleaning

SQL may not be the most efficient cleaning method at your fingertips, but it is capable. If a dataset contains many errors, like misspellings or empty rows, you may want to first run it through a cleaning method like spreadsheet functions, OpenRefine, or the other tools we talked about in Chapter 4. But if the data is relatively organized, editing some data in SQL may be the smartest choice. In the SQL world, this kind of work is called "data manipulation" because you are changing the data values themselves, not sorting or analyzing them.

One way to keep track is an "alias" – a new or alternate name for a column. Aliases are especially useful when creating calculated columns, like totals or averages. They are established with the AS operator in a SELECT clause, like this:

```
SELECT AVG(population) AS avg_pop
```

We're going to clean the confusing "ZIP / Population" column by extracting the ZIP code and putting it in a new column called "ZIP." SQL uses many of the same functions you find in spreadsheet programs, including LEFT().

If you recall from Chapter 6, the LEFT() function extracts a certain number of characters from the left-hand side of the specified column. In SQL, the column is usually the first parameter, and the number of characters is the second.

As you get more used to writing SQL, it will become safer to omit the original column and SELECT only your newly created field. But if you are unsure, it's fine to retrieve the original column as well, so you can see them side by side.

Figure 8.4 The LEFT() function in SQL.

Exercise 4

1. In DB Fiddle, delete your query and write a new one:

```
SELECT income,
   'ZIP / Population',
   LEFT('ZIP / Population',5)
   AS ZIP
FROM income _ data
LIMIT 10
```

2. Click Run.

Here, we are selecting three columns: the income, the Zip / Population, and a new column called ZIP. The ZIP column contains only the first five characters of the "ZIP / Population" column (Figure 8.4).

At this point, it would be smart to store this output as a new table. The table would have three columns, one of which is a simple five-character ZIP code. Unfortunately, DB Fiddle doesn't allow us to store queries or tables, so in our practice sessions, we will need to keep establishing the calculated column in every query.

We've taken an important step: Separating the ZIP code from the area's population. But what towns are these? Journalism audiences don't know the ZIP codes of every town in California. For that, we'll have to venture into one of the biggest advantages of SQL: Joins.

Joining

Joining datasets is one of the main reasons journalists use SQL instead of other analysis methods. While you can technically use two datasets at once in a spreadsheet program, SQL has dedicated tools for this and can handle much larger sets of data.

Luckily for us, the Census has another dataset of Postal Service ZIP code areas, including the city and county names. We will write a query to "join" those two datasets, using the ZIP code as a guide.

Unless there are errors, the extracted ZIP codes in the income table should exactly match the ZIP codes in the postal code table. This means we can use the WHERE command with the equal sign operator to find a perfect match.

In a Join, these are often called "keys." The ZIP in the income table is the "primary key," and the ZIP in the postal table is the "foreign key." The values must match exactly for the Join to work, which is why we extracted the ZIP code using the LEFT() function.

In a JOIN query, you use the SELECT command to choose columns, but this time you can choose columns from different datasets. If they happen to have the same field name, you can use a period to identify the source table, like this:

```
SELECT income_data.ZIP, postal_data.ZIP
FROM income_data, postal_data
```

That simple SELECT command would show two columns of ZIP codes – one from the income_data table and another from the postal_data table. If the field name is unique, you do not need to use a period to specify the table it's from.

* * *

Pro Tip

To run a join, you don't need to SELECT the joined column – it is working as a kind of glue in the WHERE clause. But to be sure your queries are executing correctly, you can retrieve all of the columns including the keys. Some columns will be redundant, because the primary key will match the foreign key, but you can be sure that your Join is operating correctly.

* * *

Let's execute a simple SELECT statement to see a Join in action.

To keep track of the ZIP codes from different tables, we will use aliases. Let's call the ZIP from the income table "income_zip" and the ZIP from the postal code table "postal_zip".

* * *

Exercise 5

1. Download this CSV file of California ZIP codes (shortlink: https://bit.ly/ziptable) and open it in a text editor. We can see there are four columns, with one named "zip" and one named "city." (Remember, exact column names are important.)
2. Import it into DB Fiddle using the instructions in Exercise 1. Name the table "postal_data." It should now appear as a second dataset in the Schema SQL panel.
3. In the Query SQL panel, write the following query:

```
SELECT income,
    LEFT(`Zip / Population`,5) AS income_zip,
    postal_data.zip AS postal_zip,
    city
FROM income_data, postal_data
WHERE LEFT(`Zip / Population`,5) = postal_data.zip
LIMIT 10
```

3. Click Run and look at the output.

Here we selected four different columns: the income, the ZIP code from the income table, the matching ZIP code from the postal table, and the city from the postal data. This is an example of how a Join is powerful. Now that we have joined these datasets, we can see the place names for these incomes, not just their ZIP codes (Figure 8.5).

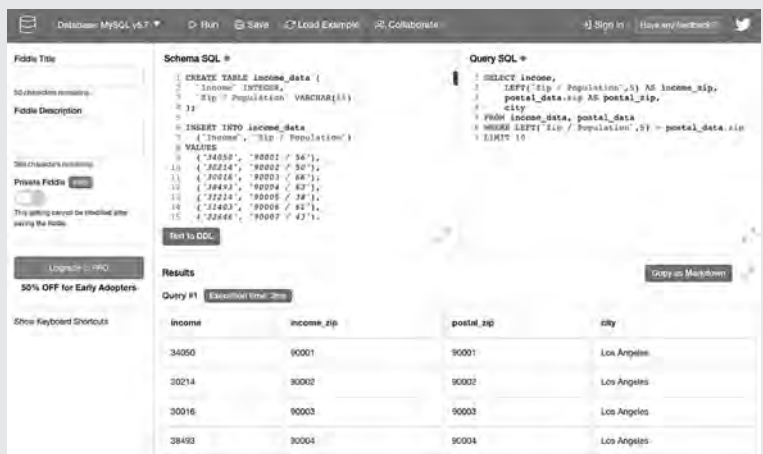


Figure 8.5 Joining ZIP codes from multiple datasets.

Inner and Outer Joins

There is another kind of join called an "Outer Join," which returns rows even if they don't match a key. As you learn more SQL, you might come across terms like "Left Join," "Right Join" and even "Cross Join."

The type that we did is called an "Inner Join," and it limits the output to rows where the primary key matches the foreign key. Inner Joins are the most common and the most likely to be used in a data journalism analysis.

So far, we have accomplished a lot of cleaning and analysis with SQL. We identified the cities in these ZIP codes and limited the results by income. But we still have a challenge ahead of us. Most large cities, like Los Angeles, have several or even dozens of ZIP codes – but journalism audiences would likely think of them as the same city. We will move on to another powerhouse in the SQL syntax: Grouping.

Grouping

The GROUP BY command groups together rows with matching values, similar to pivot tables in spreadsheet programs. (Refer back to Chapter 6 for more on pivot tables.)

One good use of this is the postal code dataset. Instead of dozens of rows for ZIP codes in Los Angeles, we could group them into one row called "Los Angeles."

There is one downside to this method: Since we are compressing all of those rows into one, we can no longer see all of the individual ZIP codes or incomes.

This means we will have to use a calculated column rather than simply calling the income column. Calculated columns are usually created using SQL functions like COUNT() and SUM(). We will use the AVG() function to find the average income in each city.

Exercise 6

1. Enter "AVG(income)" in your SELECT clause to create a calculated column of income averages.
2. Second, add "city" as a second column. Because we are grouping by city, it makes sense to see each city listed in our output.
3. Keep the FROM and WHERE clauses the same, since we are using the same tables and the same joining criteria.
4. After the WHERE clause, add a GROUP BY clause to group the rows by city. It is not necessary to specify "postal_data.city" with a period, because there is only one column called "city".

5. Keep the LIMIT 10 clause intact to avoid outputting more rows than you need. This can slow down your computer. The final query should look like this (Figure 8.6):

```
SELECT AVG(income), city
FROM income _ data, postal _ data
WHERE LEFT(`Zip / Population`,5) = postal _ data.zip
GROUP BY city
LIMIT 10
```

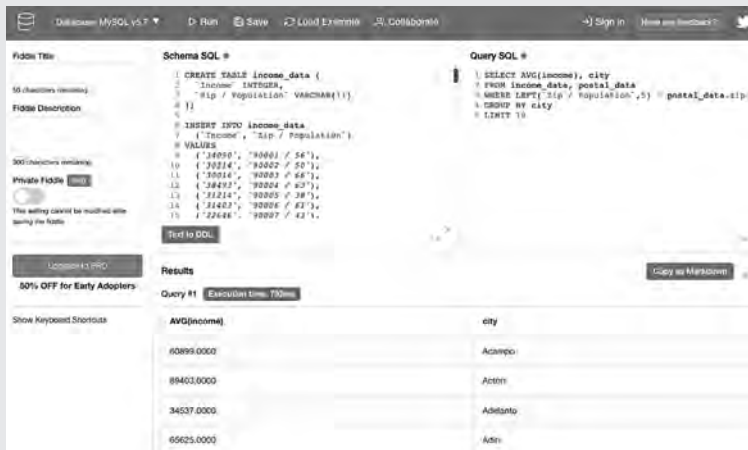


Figure 8.6 Selecting the average income and city.

We can now see that, for instance, the average income in Acampo, California, is around $60,000.

To make our query less unwieldy, we omitted the ZIP code columns from our SELECT clause. Because they are listed in the WHERE clause, we know they are working to join the two datasets, but we don't actually need to show them.

Now we have a list of ten average incomes grouped together by city. We're almost there! The last step is simply to sort those incomes from highest to lowest.

Sorting

Sorting is a rather simple task within SQL, typically with the ORDER BY command.

Exercise 7

1. Add an ORDER BY clause after your GROUP BY clause, to create this full query (Figure 8.7):

```
SELECT AVG(income), city
FROM income_data, postal_data
WHERE LEFT(`Zip / Population`,5) = postal_data.zip
GROUP BY city
ORDER BY AVG(income) DESC
LIMIT 10
```

Figure 8.7 Ordering by average income.

Remember that ORDER BY defaults to arranging values in ascending order, which is why we included the DESC keyword.

* * *

Pro Tip

You can also sort multiple columns with the ORDER BY clause. This option comes into play if several rows have the same value in one column. For example, this clause would sort alphabetically by country and then by city:

```
ORDER BY country, city
```

* * *

And with that, we have our story! Atherton is the wealthiest city in California, at least by this measurement. The other nine cities provide some good context and additional information for the reader.

We used SQL for spreadsheet-type functions, like filtering and sorting, as well as more advanced techniques like joining and grouping. In addition to this, SQL programs can process much more data than your average spreadsheet.

* * *

For more data journalism tips, tricks and exercises, visit the Data + Journalism blog at http://dataplusjournalism.com

* * *

Footnotes

How to Get Started with Data Journalism in Your Newsroom https://www.americanpressinstitute.org/publications/reports/strategy-studies/how-to-get-started-data/
SQL Server https://www.microsoft.com/en-us/sql-server/sql-server-downloads
MySQL https://www.mysql.com/
Oracle https://www.oracle.com/index.html
Microsoft Access https://www.microsoft.com/en-us/microsoft-365/access
DB Browser https://sqlitebrowser.org/
SQLite https://sqlite.org/index.html
Tableau https://www.tableau.com/
Microsoft PowerBI https://powerbi.microsoft.com/en-us/
California Household Income https://drive.google.com/file/d/10FmnemIAbDZlu4tlBSi5D_TTvoS_3GZ6/view?usp=sharing
California Household Income shortlink https://bit.ly/incometable
2010–2014 American Community Survey https://www.census.gov/programs-surveys/acs/technical-documentation/table-and-geography-changes/2014/5-year.html
MySQL Workbench https://www.mysql.com/products/workbench/
SQLite Manager https://chrome.google.com/webstore/detail/sqlite-manager/njognipnngillknkhikjecpnbkefclfe
QUERY() function https://support.google.com/docs/answer/3093343?hl=en
DB Fiddle https://www.db-fiddle.com/
ZIP Code Tabulation Areas https://www.census.gov/programs-surveys/geography/guidance/geo-areas/zctas.html
California ZIP Codes https://drive.google.com/file/d/1cvym2cbUTRo4VqniVNDabAu2PWZEVqng/view?usp=sharing
California ZIP codes shortlink https://bit.ly/ziptable

9 Scraping Social Media

Samantha Sunne

Introduction

In 2015, two tenacious Associated Press reporters were facing a problem. Illinois state lawmaker Aaron Schock was facing accusations of corruption and bribery, and he wasn't returning their calls.

In an attempt to dig in to some of these accusations, the reporters turned to what was then an unusual source: Instagram.

"The AP extracted location data associated with each image then correlated it with flight records," Jack Gillum and Stephen Braun later explained in their story. "The AP identified at least one dozen flights worth more than $40,000 on donors' planes since mid-2011."

Gillum and Braun's story, using Instagram metadata, revealed a pattern of flashy and ethics-violating behavior by the lawmaker. It partly inspired a congressional inquiry and also served as an introduction to social media investigations for many in the journalism industry.

These days, using Instagram as a source wouldn't be considered unusual. Journalists as well as their audiences are aware of how much data everyone is constantly releasing via social media networks.

We have scraped data from the web, but in this chapter, we will learn that many front-end tools, like web browsers and mobile apps, are not the only way to interact with data. Combine that mentality with a "data frame of mind," and you are well on your way to exposing secrets on the Internet.

When Gillum and Braun pulled location data from Instagram posts, they were accessing "metadata," which has a rather circular definition: Metadata is data that describes other data. It's easier to think of metadata in real-world examples.

For instance, look at this Instagram post by the World Health Organization (WHO) (shortlink: https://bit.ly/whoinstagram). The post itself is the "data." But this data, in turn, carries its own collection of data, like the number of likes, the day it was uploaded, the user ID of the uploader and much more (Figure 9.1). Oftentimes, when you are scraping or backgrounding something on the web, you are digging up the metadata.

DOI: 10.4324/9781003273301-10

Figure 9.1 WHO Instagram post

Digging into Code

Metadata is everywhere, but if you're looking at a web browser, an easy way to find it is by opening the "source code." Source code is the collection of code that creates whatever you are looking at, whether that's the HTML behind a web page or the JavaScript behind a "subscribe" button. Often, source code can help you achieve something that the point-and-click tools of a website cannot.

Exercise 1

Right-click on the image in this WHO Instagram post (shortlink: https://bit.ly/whoinstagram) and try to choose "Save Image As." The option isn't there. That's because Instagram doesn't want to let you download images through its website. Instead, we will obtain the image through the website's source code.

Right-click on the page's white space, and this time select "View Source" or "Page Source." This opens up the source code in a new tab. This page may look intimidating, but that's just because it's meant for a computer to read, not a human. The easiest way to navigate it is to search for recognizable, human-readable parts. If you scroll down long enough, you will find elements you recognize, like the WHO's caption for this post.

This is called "looking under the hood," and it's not the most efficient way to navigate a page's source code. But it does offer a window into the complicated goings-on behind a web page. An easier way to explore it is the Web Inspector, which we learned about in Chapter 3. Every browser has a Web Inspector functionality, though sometimes it goes by different names, like "developer tools." We will continue using Google Chrome.

Right-click again on the image in the Instagram post, and this time, choose "Inspect" or "Inspect Element." Now, the Web Inspector will pop open in a panel. It highlights parts of the code so that you can see where exactly each part of the web page comes from (Figure 9.2).

Because we clicked "Inspect" on the image, it should highlight the image's HTML element within the pop-up panel. You may need to expand the section with the drop-down arrow, or click on nearby parts of code, in order to find it.

You can also use the Web Inspector in the reverse order. In the Inspector's toolbar, select the cursor button to make sure you have the "inspect element" function enabled. Then you can click on different parts of the web page, and the Inspector will highlight them in the code accordingly.

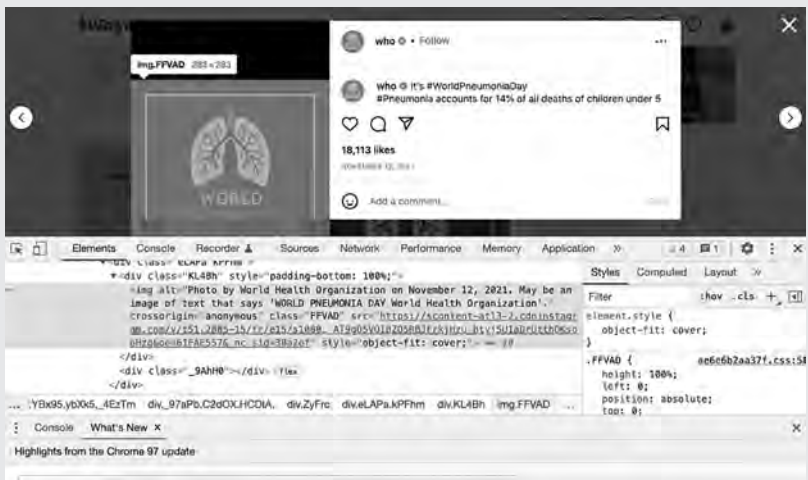


Figure 9.2 The Web Inspector "inspect element" tool.

In our Instagram post, the HTML containing the image should include the letters "src," meaning "source," and a long URL. This is the link to the image's real home on the web. Open it in a new tab to see the raw image, and now you can save it! This is an example of how you can use source code to be more than a passive web surfer.

Exercise 2

Now let's obtain the image using another function of the Web Inspector – the Network panel. The Network panel shows you all of the different calls your browser is making to download site resources located on different servers.

Click on the Network tab to see a list of resources the page is using. If this tab is empty, refresh the page using your web browser. Then you can watch as the Network panel loads all the various elements and requests it uses to create the page (Figure 9.3).

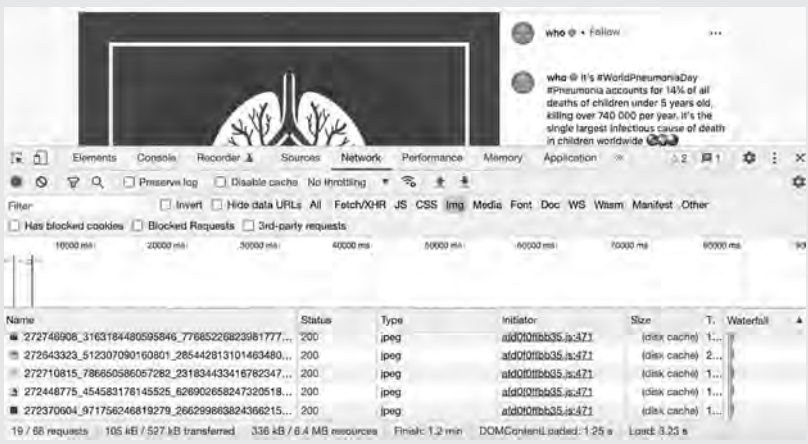


Figure 9.3 The Web Inspector Network panel.

Sort the results by Type to find the images – they will usually be png or jpg files. If you click on each one, you should find the raw image that we opened in Exercise 1.

Another option is to sort by file size. Some journalists have even had success finding entire databases through these Network panels.

* * *

Pro Tip

As we discussed in Chapter 1, journalists are typically going after public information like government databases. In those cases, you don't need to worry about breaking laws or website terms of service.

But if a site has the information behind a paywall, under copyright, or after a CAPTCHA, those are clues that you may be violating its terms of service or even the law. Some sites also restrict the number of requests you can make per session, although that is often to avoid overloading their servers.

If in doubt, it is always best to reach out to the person or organization that publishes or maintains the data. In the journalism field, resources like the Student Press Law Center offer guidance on these topics.

Programming Languages

In Chapter 3, we briefly discussed programming languages – tools you can use across many different platforms. R, Python, JavaScript and C++ are all examples of different programming languages.

There are hundreds of languages in existence, but only a handful tend to surface as the most common. Github, a popular website for sharing and editing code, publishes a list of the languages most commonly used on its site (Figure 9.4). On Github, you can find scripts of code that other people have written and made freely available. This is referred to as "open-source" code.

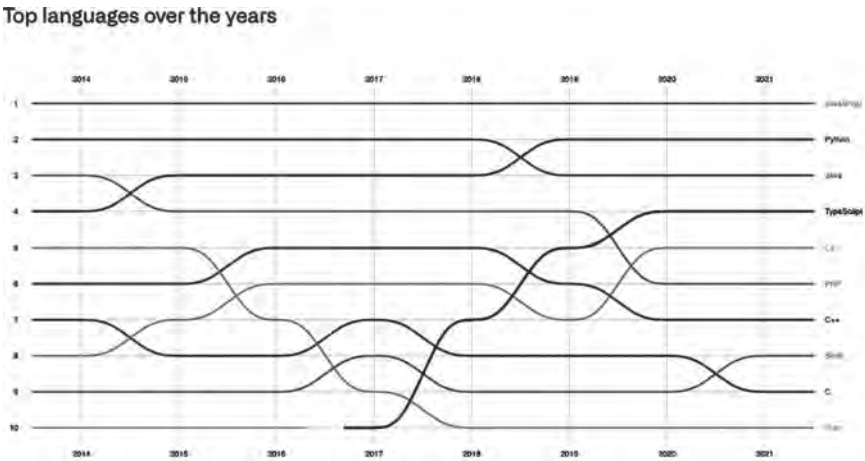


Figure 9.4 Popular coding languages on Github.

So, which language should you learn? To begin with, you should assess which would best meet your needs. For instance, R is popular in data journalism because of its large-scale data analysis tools. JavaScript is popular for interactive items or animations. Both of these tasks are doable in Python.

If nothing else, you may want to simply look for a sample script of what you are trying to accomplish – say, scraping the HTML of a website. The same task may be easier for you in one language than another.

In addition, languages like Python have active communities online that can give you support. You may know someone in person, a friend or newsroom coworker, who has some familiarity with a language. All of these are valid reasons for picking a language.

This book will mainly rely on Python, which is one of the most popular languages in data journalism. We also will not be delving too deeply into writing code: Software engineering is an enormous field in its own right, and we will only be learning enough to accomplish our tasks, like scraping metadata. Sometimes, the data is so hard to get that it's not worth the time and effort it takes to write a scraper. In those cases, it's better to request the data through a human source like a public records request or a public information officer – even if you need to negotiate with them.

Coding Challenges

Writing code tends to have a higher learning curve, but it also pays higher dividends. You are free from the limitations of third-party tools, but you also need to manually create your own.

Instead of downloading tweets with Twitonomy, for instance, you can create your own tweet downloader using the Twint Python library.

A "Python library" is a set of code that someone wrote in an existing programming language to accomplish specific tasks and shared with the world. Sometimes prepared bits of code are also referred to as "packages" or "modules."

Table 9.1 Useful Libraries

pandas	Pandas is an extraordinarily useful Python library for data analysis. It turns tasks like parsing HTML and merging large data files into something as simple as a few lines of code.
BeautifulSoup	BeautifulSoup has spent years as a popular Python library for scraping web pages into spreadsheets. It helps complete the intermediary step of "parsing" source code into chunks that can then be separated into rows and columns.
pdftotext	Pdftotext is a tool run entirely in the Command Line. It scrapes text and data out of PDFs and can handle up to 20 documents at a time.
CSV	CSV is the name of a file format that can be imported into spreadsheet programs, but in this case, it's also the name of a Python library for creating (or "writing") CSV files.
t	t is a Ruby gem for accessing Twitter data. It can perform bulk Twitter operations like following and unfollowing from the command line.
dplyr	dplyr is an R package that makes it easier to do the kind of data analysis you're used to doing in SQL and spreadsheet programs.

You can find a list of useful libraries at the end of this chapter (Table 9.1). With almost every script, you will need to install them as add-ons to the language you installed on your computer, which is why you will sometimes see them referred to as "dependencies" or "requirements."

In this chapter, we will use code to access, download and analyze data from social media sites.

Writing functions and scripts in programming languages, rather than in a platform like Google Colab, comes with several hurdles. As with many things, it's best to take these hurdles one at a time, until you can at least run a function without getting an error message.

Though it can be daunting to learn to code, it is entirely possible to learn enough to aid your reporting!

* * *

Pro Tip 

When writing code, you are inevitably going to encounter error messages or simply be unable to run a script. Chapter 4 of this book contains troubleshooting options for some common problems, but the most foolproof method is to simply copy the entire text of the error message and paste it into Google.

Most of the time, you'll find someone else has posted the same query and received a response. If you can't find your specific problem, try adding keywords like the name of the programming language.

Coding Terminology

Many of these coding tasks come with an enormous amount of vocabulary that may be unfamiliar to you. While reading this chapter, and googling your error messages, you may come across phrases like "command line interface," "Python interpreter," "virtual environment" and other daunting terms.

Don't attempt to memorize these concepts and phrases right off the bat. It's not necessary to know all of these keywords, and even professional software engineers often find themselves looking up definitions and syntax.

If you are interested in learning to code further, you may want to become more well-versed in this vocabulary. But because we are learning technology to aid journalism, and not the other way around, we won't go out of our way to learn the many different ways to install, store and run code. You will find a detailed glossary at the end of this chapter (Table 9.2). But remember, there is no shame in googling!

The Command Line

The Command Line is an application that lets you communicate with the computer via simple lines of text. It's sometimes referred to as the "console," the "terminal" or the CLI, which stands for Command Line Interface. Often, you will write a script in a program like a text editor, but then run it using the Command Line.

On a Mac, the CLI is often accessed through an application called the Terminal. On Windows, it is called the Command Prompt.

As mentioned earlier in the chapter, writing code offers many advantages, one them being the ability to write your own applications. The downside is that you will need to manually set up tasks that happen automatically in third-party tools.

Things like exporting a dataset, or importing it into a spreadsheet, are tasks that happened automatically when we scraped data using the Google Sheets IMPORTXML function. But when you're writing your own function, you'll need to manually write code to export the scraped HTML elements into a list that you can then import into a spreadsheet.

Installing Languages and Libraries

When learning to code, one of the biggest challenges can be the first step: Installing the necessary tools onto your computer. While you have probably heard of programming languages like Python, it may or may not be something you need to download and install.

The Python website has instructions for downloading and updating the Python language. Some scripts and libraries work better with earlier versions of Python, in which case you might want to use something like a virtual environment. See the glossary at the end of this chapter for guidance on opening and running Python.

In addition, in order to use Python or another language for a data journalism task, you will most likely need to install additional tools inside it, like libraries. Usually, you will install or load these libraries inside the script you are writing. These add-ons are sometimes called "dependencies" or "requirements."

One of the biggest hurdles to programming can be just getting it set up on your computer. So have patience!

Pro Tip

When programmers share prewritten code, it often contains "documentation" – comments or notations between lines of code that help other programmers understand what the script is doing.

As you write your own code, you should strive to maintain your own documentation, or "docs" for short. This not only helps you keep track of your own work but also makes it reproducible, meaning others can reuse your code to scrape their own websites. Writing documented, reproducible code contributes to the "open web" and makes it easier for everyone to learn and use programming.

You can find many examples of "open-source" code and documentation on Github, the code-sharing website.

* * *

Exercise 3 

For this exercise, we will download and install the Python language onto our computers. If you have an Apple computer, or if you have written code before, you may already have Python ready to go. Here is how to find out: open the CLI and simply type the word "python" (Figure 9.5).

On Mac computers, the Command Line is accessed via an application called Terminal. If you can't find the application in Finder, just search for it using the MacOS Spotlight Search. On Windows, the application is called the Command Prompt, or cmd.exe. The easiest way to find it is by searching for "cmd" in the Start Menu.

1. Open the CLI. It should start with a screen showing the names of your computer and your user.
2. Type the word "python" and press enter.
3. If the CLI returns a series of text that contains the word "Python" followed by a number, like 2.7.10, that means you have Python installed. Type "quit()" and press enter to exit the Python interface.
4. If it returns an error message, that means you do not have Python installed. In a web browser, go to python.org and follow the instructions to download the latest version.

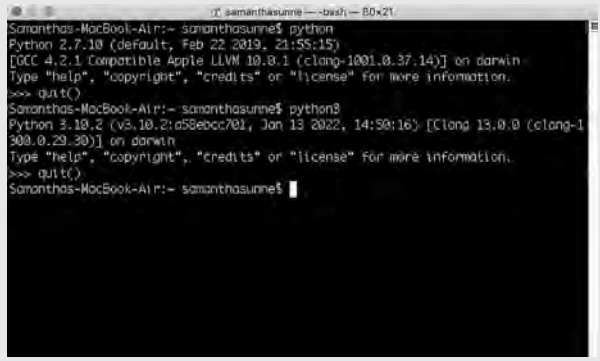


Figure 9.5 Python in a MacOS terminal.

Scraping with Code

Now, we will use Python to scrape live data from social media websites. Use the glossary at the end of this chapter and Chapter 4, Cleaning Data, to troubleshoot your progress. First, we will scrape a Twitter user's full history of tweets using a Python library called Twint. We're going to take a look at how often the US White House tweets about China.

On a regular web browser, you would have to scroll indefinitely to load all of a user's tweets, which can number in the millions. Twitter doesn't give you an option to download all of this data from its website, but you can create an application to download your own.

If, at the end, you find you are unable to execute the file, you may want to repeat the script in the browser-based Google Colab platform. See Chapter 3 for instructions on how to use Google Colab.

File Paths

With programming, it's important to keep all of your scripts and code files in the correct folders. To do that, you'll need to write directories. The directory, or file path, is programming-speak for a folder or a location on your computer.

It's often written out as "user/folder/subfolder" on a Mac and "user\folder\subfolder" on Windows. Sometimes you will need to manually type out a directory in order to execute code.

One easy way to move around folders in the Command Line is to use the "cd" command. This stands for "Change Directory," and you can type "cd [folder name]" to move into any folder as long as you are currently located in the folder that contains it. In this exercise, we will use the Desktop to make navigation as simple as possible (Figure 9.6).

Exercise 4

1. On your Desktop, right-click and create a new folder called "python_code".
2. In the Command Line, which we opened in Exercise 3, type "pwd" and press enter. This will return the name of the folder you are currently in.
3. Type "ls" to see a list of files and other folders inside this folder.
4. If "Desktop" is among them, type "cd Desktop" to move to that folder.
5. If "Desktop" is not an option, use the "cd" command to navigate to the folder that contains it. Specifically, you can use "cd.." to move back one folder, until you find the folder that contains the Desktop.
6. Once you are in the Desktop, enter your "python_code" folder by typing "cd python_code".
7. Type "pwd" again to make sure you are in the correct folder.

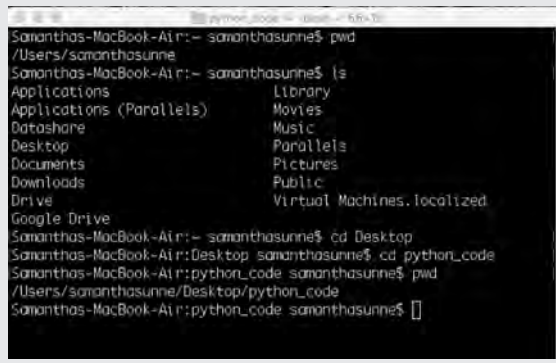


Figure 9.6 Navigating the command line.

Virtual Environments

Next up on our list is to create a virtual environment. A virtual environment lets you install and deploy bits of code, and entire programming languages, within a walled area inside your computer. This helps prevent the code from affecting other parts of your machine.

For example, you may need Python version 2 to run a certain script, because the script was written before Python 3 was released. You could load Python 2 into a virtual environment and run the script in there, without replacing the Python 3 that is installed on your computer. Virtual environments are often referred to with the shorthand "venv."

Exercise 5

1. Make sure you are in your python_code folder using the navigation tools we learned in Exercise 4.
2. Type "python3 -m venv virtualenvironment" and press enter. This will create a virtual environment in that folder.
3. Next, we will "turn on" the virtual environment. On a Mac, type "source virtualenvironment/bin/activate" and press enter. On Windows, type "virtualenvironment\Scripts\activate.bat".

After these steps, you should see the text "(virtualenvironment)" at the beginning of each terminal line. This indicates that we are now operating in a virtual environment.

Installing Dependencies

Our last step, before actually writing code, is to install the libraries we need. Since we are using the Twint library, we can find the install instructions on the Twint website.

Exercise 6

In the Command Line, make sure you are in a virtual environment using the steps in Exercise 5. Type the following commands:

```
install git
git clone https://github.com/twintproject/twint.git
cd twint
pip3 install. -r requirements.txt
```

This will install the Twint library. Your terminal will show each component as it downloads.

We're also going to use a tool called Jupyter Notebooks, which we will use to write and then execute our code. After installing Twint, install Jupyter Notebook by typing the following command into the Command Line and pressing enter:

```
pip install jupyter
```

Once it is done loading, type "jupyter notebook" and press enter.

This will open a notebook in a new tab of a web browser. If you want to close the notebook to continue writing in the terminal, press Ctrl+C. Later, you can open it again by typing "jupyter notebook".

Pro Tip

You will often encounter problems and need to retrace your steps to figure out what went wrong. For example, if you're unable to create a virtual environment, try the command again with "python3" instead of "python." If you're unable to install a library, try it again with "pip3" rather than "pip."

It is always easiest to copy the error message, straight out of the terminal or script, and paste it into Google.

Writing a Script

Here comes the part where we write our code! A Jupyter Notebook is like a midway point between a Command Line and a point-and-click tool. Once a Jupyter Notebook has opened in a browser tab, you can write and run code in "cells."

Exercise 7

In your Jupyter Notebook, click New > Python3 to create a new Python file.

In the first cell of the Jupyter Notebook, type the following text:

```
import twint
import nest _ asyncio
nest _ asyncio.apply()
```

Create a new "cell" by clicking the plus sign (+) in the Jupyter Notebook toolbar. In this second cell, type the following text:

```
c = twint.Config()
c.Username = "whitehouse"
c.Search = "China"
c.Limit = 100
c.Store _ csv = True
c.Output = "white _ house _ china _ tweets.csv"
twint.run.Search(c)
```

This will use Twint to find the most recent 100 tweets by the Twitter account @whitehouse that contain the word "China" and export them into a CSV (Figure 9.7). This CSV will go into the python_code folder that you created. This is one reason why it's important to keep track of which folders contain your code, source data, outputs and work logs.

Figure 9.7 Script and output in Jupyter Notebook.

As stated earlier, there are far too many options in programming to cover in this chapter. If you would like to learn more about how to construct a Twint search, you can find documentation on the library's website.

Scraping with APIs

API stands for Application Programming Interface. It's a way to communicate between two applications via programming, as opposed to something non-programming, like typing in a search term.

APIs most often come in handy in data journalism when reporters are scraping data or creating an automated task. They let the reporter's code communicate with the source automatically.

Creating your own tool to obtain data by talking directly to the source's API may not be as difficult as you think. Take an excerpt of this script by Lam Thuy Vo, who used it to scrape Facebook comments for BuzzFeed News.

```
import json
import datetime
import csv
import time
import ssl
from utils import request_until_succeed, open_csv_w
from secrets import FACEBOOK_APP_ID, FACEBOOK_APP_SECRET
…
# get authentication
access_token = FACEBOOK_APP_ID + "|" + FACEBOOK_APP_SECRET
…
# Construct the URL string
    base = "https://graph.facebook.com/v2.9"
    node = "/%s/comments" % status_id
    fields = "?fields=id,message,like_count,created_time,
             comments, from,attachment"
    parameters="&order=chronological&limit=%s&access_to-
              ken=%s" % \ (num_comments, access_token)
    url = base + node + fields + parameters
…
def scrapeFacebookPageFeedComments(page_id, access_token):
  # with open('%s_facebook_comments.csv' % file_id, 'wb') as
file:
  with  open_csv_w('../output/%s_facebook_comments.csv'  %
file_id) as file:
    w = csv.writer(file)
    w.writerow(["comment_id",    "status_id",    "parent_id",
"comment_message",
       "comment_author","comment_published","comment_likes"])
```

Due to changes in Facebook's API, this script no longer works, but you can see how Thuy Vo used it to download the text and metadata of Facebook comments and export the data into a CSV file. The script starts by importing the required libraries, then constructing custom URLs and then scraping data points from each Facebook comment.

APIs are different from scraping HTML, like we did with IMPORTXML or using Python, like we did with Twint. An API is more like an ongoing hose of information from one application to another.

There are some limitations to them, mainly because developers don't want you to overload their site with requests. The Twitter API, for instance, limits how many tweets you can scrape at one time. You also need an API key, which you must request from the website that runs the API. Twint and csv are two useful libraries for scraping social media data, but there are many, many more.

Table 9.2 Programming Glossary

Programming Language	A programming language is a defined set of words, phrases and syntax that helps humans communicate with computers. Some languages are somewhat similar to human speech, like Python, and some are not, like binary.
Machine-Readable Data	Machine-readable data can be thought of as data that is ready to be read and processed by a computer, with no additional steps necessary. For example, a PDF is not necessarily "machine readable" because it is meant to be printed out and read by humans. A CSV *is* machine readable because it can be imported straight into a spreadsheet.
Structured Data	Data that has been tagged and identified is considered "structured data." This makes it easier and faster for a computer to run functions and can also make the data more easily understood by humans.
Console	When trying out code, the console is where you view your various inputs and outputs. You can test functions by looking at the outputs that get printed in the console before trying to actually scrape and export the data. It might not be smart, for instance, to run a whole scraper on Facebook's API before you're sure the script works.
Python Interpreter	The Python Interpreter is a fancy name for the machine that runs Python on your computer. Running Python commands outside of the Interpreter may not work because your computer doesn't inherently understand every Python function. Rather, you need to enter the Python Interpreter by typing "python" (or the most recent version, like "python3") into your CLI. You know you are in the Interpreter if the CLI shows three brackets (>>>) at the beginning of each line instead of a dollar sign ($).
Python Standard Library	The Standard Library is the set of libraries that come automatically with Python. It includes basic functions like help() and print() so that you don't have to install additional libraries just to get support for your code. But to accomplish many data journalism tasks, you will need to install additional libraries.
Script	A script is a finite collection of code, which usually accomplishes a specific task or set of tasks. A common programming method is to write a script in a word processor, like a text editor, save that file on your computer and then run the script using the CLI.

(*Continued*)

Table 9.2 Continued

Function	A function is a finite set of code, usually within a script, that accomplishes a certain task. For example, in Python, the print() function displays an output in a console, while bespoke functions like "scrapeFacebookPosts()" may be much more complex. In many programming languages, they are written with the syntax "function()", where the parameters of the function are listed inside parentheses that come directly after the name of the function.
Parameter	A parameter is an input in a function. For example, in our function importHTML("URL", "HTML element"), the URL we want to scrape is the first parameter, and the HTML element is the second.
Argument	An argument is the value in a parameter. For example, in the function importHTML("https://example.com/page", "table"), the first argument is the URL "https://example.com/page" and the second argument is the HTML element "table."
Variable	A variable is like a nickname you use to refer to something when writing code. You will have to define what your variable refers to early on in the code in order to use it later. One example is a URL. Instead of typing out a long URL, you can store it as a variable simply called "URL." Then you can tell your function to scrape "URL" rather than a long string starting with "https://".
Define	When you create a function inside a script, it is often called "defining" a function. It can also be called "declaring" or "stating." "Define" usually means an object is being given a name so that a function can call on it later. See "variable" and "reference" in this glossary.
Parse	To parse means to separate bits of data based on labels or indicators. For example, the BeautifulSoup Python library helps users to scrape HTML by parsing out the text into HTML elements.
Object	An object is an entity within a bit of code that can contain many different variables, values and data types. In pandas, for instance, you need to store a dataset as an object before being able to export it to a spreadsheet.
Array	An array is a list of values, but it's not as simple as a human-written list or the HTML "list" element. Arrays help to store many values so that you can tell a function to operate on all of them rather than on each item individually. In addition, some data can only be downloaded and parsed through an array.

Table 9.2 Continued

for loop	A "for loop" is a common programming technique that orders a function to run its task on every item in a list. For example, you could use a for loop to scrape a certain HTML element from every website in a list of websites. You can read a for loop as, "for every item in this list, do this function."
Reference	An empty code script is like a blank slate. Very often, something needs to be defined before it can be used. To a human, you could say something like, "I want to scrape this table." But to a computer, you would first need to "define" the table. When you start running functions, it's common to get an error message called a Reference Error. This means you referred to something without defining it first. Look above the line where the error occurred to find where you should have defined it.
Pagination	Pagination refers to the number of pages that contain the data you're looking for. This is often a challenge in scrapes where a database or list of results is spread out among many sequential webpages.
Call	"Call" often means to execute a bit of code or to send a request to something else. For example, running a function within a script can be "calling a function," while running a script to request data from an API is referred to as an "API call."
Run/Execute	To run or execute a script is to make it actually perform the task it was designed to do. This is different from printing outputs in a console, which you will most likely do before actually running a script. Many people write a script in one program, like a text editor, and then run it using a different program, like the CLI.
Debug	To debug is to test a bit of code for problems – not to actually execute the script. Often a code editing tool will return error messages or print outputs in a console to help you get ready to execute it.
Print	In Python, to "print" something means to display it as an output within a console. For example, running the function print("Hello") will simply return the text "Hello" inside your console. This is different from exporting or creating an end product, like a CSV. Printing can be useful for checking your work as you go along. For example, you could print a variable in your console to make sure the variable is assigned correctly. Other languages, like JavaScript, have similar functions such as console.log().

(*Continued*)

Table 9.2 Continued

Chain	To chain commands means to use one command to accomplish another. Sometimes this method saves time and space, and sometimes it is necessary to run a function at all. A chain is written with a period (.) between two commands. For instance, the chain "pandas.read_html()" tells the computer to run the read_html() function that exists inside the pandas library. Without referring to the pandas library, the Python interpreter wouldn't know what the read_html() function is.
String	A string is a sequence of characters, like the word "January" or the text "04FG8n." While they may seem simple, strings are an important step between understanding human and computer language. For instance, a computer doesn't automatically know that "January" is a month – it just recognizes it as a string of seven characters. Understanding this distinction can be very influential in the success of your parsing and scraping.
HTML	HTML stands for HyperText Markup Language. Despite having "language" in the name, it is not a programming language but rather a markup language – a set of rules and syntax designed to identify and categorize bits of content on a web page.
Log	A log is like a record of activity for a script or part of a script. A log will show you, for instance, the output of a function or an error message it caused. When your script fails, logs are critical to understanding what went wrong.
Read and Write	In the programming sphere, "read" usually means to use the contents of a file without changing it, while "write" means to change it or another file. In this case, "Read" is somewhat analogous to "upload" or "import," while "write" is similar to "download" or "export." For example, to import an HTML file into your script, you might use the function read_html(), but to create a new row in a spreadsheet, you could use writerow().

Find more social media scraping tools on the Journalist's Toolbox: http://bit.ly/scrapingsocial

Footnotes

Associated Press, Politician Used Campaign Money on Private Jets, Massages and a Katy Perry Concert https://web.archive.org/web/2021*/https://apnews.com/article/e2f1f52c3eb34caca7d74e5bf90f27f9

Miami Herald, US Opens Criminal Inquiry of Resigning Illinois Congressman https://web.archive.org/web/20220519104506/https://www.miamiherald.com/news/nation-world/national/article15459164.html

PoliticsNation, Tuesday, February 24, 2015 https://www.nbcnews.com/id/wbna57031116

World Health Organization Instagram https://www.instagram.com/p/CQWGRuQjcW_/

Twitonomy https://www.twitonomy.com/

Twint https://github.com/twintproject/twint

Python Beginners Guide https://wiki.python.org/moin/BeginnersGuide/Download

Python.org python.org

BuzzFace threads.py https://github.com/gsantia/BuzzFace/blob/master/threads.py

BuzzFace: A News Veracity Dataset with Facebook User Commentary and Egos https://www.aaai.org/ocs/index.php/ICWSM/ICWSM18/paper/view/17825/17046

BuzzFace https://github.com/gsantia/BuzzFace

Pandas https://pandas.pydata.org/

BeautifulSoup https://www.crummy.com/software/BeautifulSoup/bs4/doc/

pdftotext https://github.com/jalan/pdftotext

csv https://docs.python.org/3/library/csv.html

t https://github.com/sferik/t

10 Data Visualization

Mike Reilley

When Alvin Chang was a Vox senior reporter in 2017, he started working on a series of stories showing how the American education system exacerbates existing inequities. It was a special project because he could use data to show the extent of the problem, dig into the history that led there and explain the policies that uphold this inequality.

But it wasn't the data that got Chang interested in this story.

Halfway through Chang's project, Tomas Monarrez, then a University of California at Berkeley postdoctoral student, contacted him to say that he had a study that showed that American school districts were drawing attendance zones to make racial segregation worse.

What intrigued Chang was that Monarrez had data on nearly every single public elementary school. But after a few hours of trying to make sense of his work, Chang realized what made this study special was *how* he was determining whether a district was making segregation worse.

Chang saw something in Monarrez's thinking that ultimately drove the story: If every American child went to the public school closest to where they live, schools would be very racially segregated because government policies of the past made our neighborhoods very segregated. But what Monarrez found was that many school districts drew their school zones to make school even more segregated than the underlying neighborhood. They had an opportunity to lessen the existing segregation, but instead they used this power to keep the systemically racist status quo.

The data for that story was crucial, but it is the visual explanations that made the piece work, said Chang, now the head of visuals and data for *The Guardian US*. He knew the story was powerful – but only if he could explain the methodology in a way that conveyed the stakes.

The project included a feature where readers could look up their school district. But by the time they looked it up, Chang wanted them to recognize that their lives were shaped by school segregation. He wanted them to internalize that their school boards actively made a decision to put them in a classroom with kids of the same racial and class backgrounds, which is no different than what our schools were doing before the civil rights movement (Figure 10.1).

DOI: 10.4324/9781003273301-11

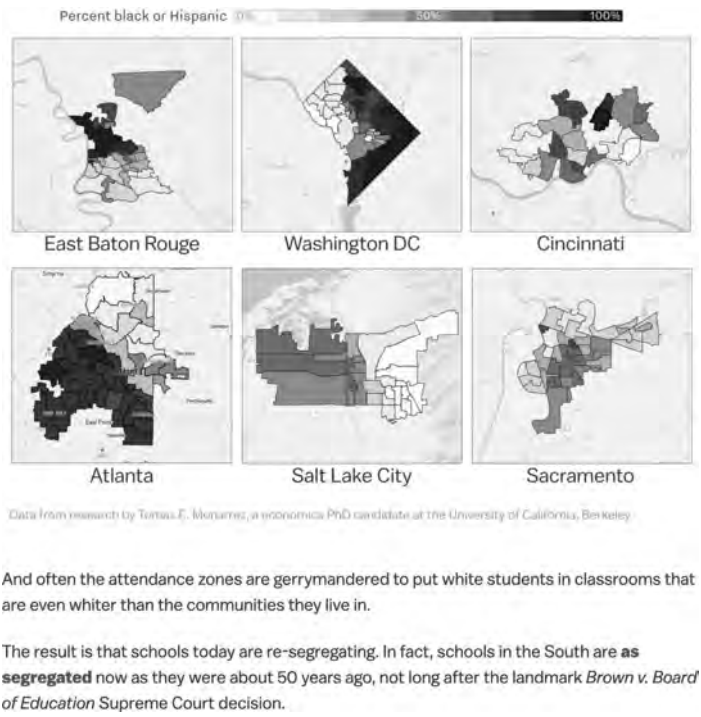


Figure 10.1 How Chang used choropleth maps and Monarrez's data to visualize segregation in six US cities.

Infographics are like glasses, says Alberto Cairo, who teaches data visualization at the University of Miami. If you take your glasses off, everything is blurry, but with your glasses on, everything comes into focus and makes sense.

This chapter discusses various types of interactive and animated charts and maps and how professional data journalists build them. It explains how to know which type of chart or map is right for your data. It explores how to choose the right colors, fonts and the size for your graphic. Using Flourish, Datawrapper and real-world data, we will walk through the process of making a variety of charts and maps with exercises and training videos.

* * *

Building Charts: Types of Charts

A good chart can make readers smarter. They show trends and patterns that might not be obvious when looking at numbers in raw data. But, as Cairo has pointed out, charts can lie. And charts can be confusing or misleading, especially

if the journalist chooses the wrong chart to present the data. There are dozens of types of charts available, so to get started, let's focus some of the basic charts and what data they can visualize.

Bar/Column Chart

Horizontal bar charts – or vertical column charts – feature rectangular bars with heights or lengths proportional to the values that they represent.

They're often used to compare year-to-year figures, percentages, etc., usually over time or between entities. For instance, it can show an increase in campaign contributions to a candidate over a 12-month period or an increase in COVID-19 positive tests over a few weeks.

Line Charts

A line chart displays information as a series of data points called markers (data points) that are connected by straight or curved lines. Line charts are popular for sports data and stock market/finance charts as these show an asset's historical price action that connects a series of data points with a continuous line. The charts show continuous growth or decrease over a scale (typically time).

Line charts and bar charts can be interchangeable with some datasets, but bar charts break out individual points (years, etc.), whereas line charts represent continuous growth and change.

Pie Charts and Treemaps

These are typically used for budgets or breaking down parts of a whole. The "pieces of the pie" must be proportional to the data corresponding with it. So if 10 percent of a city budget is going to the police department, the proportion in the graphic should amass 10 percent of the overall chart (or pie).

Treemaps are a popular and effective way to break down a budget and other data. They are ideal for displaying large amounts of hierarchically structured (-tree-structured) data. The space in the visualization is split up into nested figures, usually rectangles that are sized and ordered by a quantitative variable. Example: Treemap (Shortlink: https://bit.ly/treemapexample).

Interactive and Animated Charts

Interactive charts allow the user to control what they see: for instance, zooming, hovering over a marker and using a search bar or a pulldown menu. It can also enable the exploration of data via the manipulation of chart images, with the color, brightness, size, etc. This gets the reader engaged and *involved* in the data.

Animated charts illustrate data by creating changes and movements in the chart. Changes over time are a great way to use animated graphics, for example:

- 2020 COVID-19 Cases by State and Country (Shortlink: https://bit.ly/covidcaseschart)
- Baseball Home Runs in the Steroid Era https://public.flourish.studio/visualisation/8443792/

Venn Diagrams

Lucidchart defines Venn diagrams as "overlapping circles or other shapes to illustrate the logical relationships between two or more sets of items. Often, they serve to graphically organize things, highlighting how the items are similar and different." Common elements of the sets are represented by the areas of overlap among the circles, which shows relationships or common areas of interest.

Hierarchy and Organizational Charts

According to OrgCharting.com, hierarchical charts are typically used to show roles, ranks, levels or positions of people or things. They are designed in a format that shows the relationships between the entities, with the top of the chart typically kept for the most important or significant part of the system. These diagrams look like organizational charts in some ways, and you can use org chart makers to draw hierarchy charts as well.

There are many other types of charts – timelines, scatterplots, etc. Be sure to look through chart libraries in tools like Flourish, Datawrapper and other tools we explore later in this chapter to find more templates and designs. Many of the templates have "dummy data" in them so you can see how to format the spreadsheet to make the chart work seamlessly.

Visualizing a Story on Deadline

Journalists often think of data as structured numbers, but Chang said he believes text transcripts are some of the best data sources for journalists. The downside is that text analysis often returns murky and inconclusive results. But that was not the case for the Senate testimony of Supreme Court nominee Brett Kavanaugh and the woman accusing him of sexual assault, Christine Blasey Ford.

The morning after the hearing, Chang sat in bed reading through the transcript and noticed the extent to which Kavanaugh was unwilling to answer any questions about the allegations. He knew his biggest challenge for this story was going to be time: The window of interest for this story would be that morning, so he devised a plan that would allow him to finish relatively quickly.

First, he pasted the transcript into a text editor and started to denote the questions Kavanaugh and Ford answered and didn't answer. He read through the transcript several times to make sure his coding was correct.

Next, he visualized the data using a technique he had used in the past: Put the entire transcript onto the page, shrink the text to 1px and color the lines based

on the coding. When he first saw the visual, his jaw dropped: Kavanaugh dodged a huge number of questions, while Ford answered all of them (Figure 10.2).

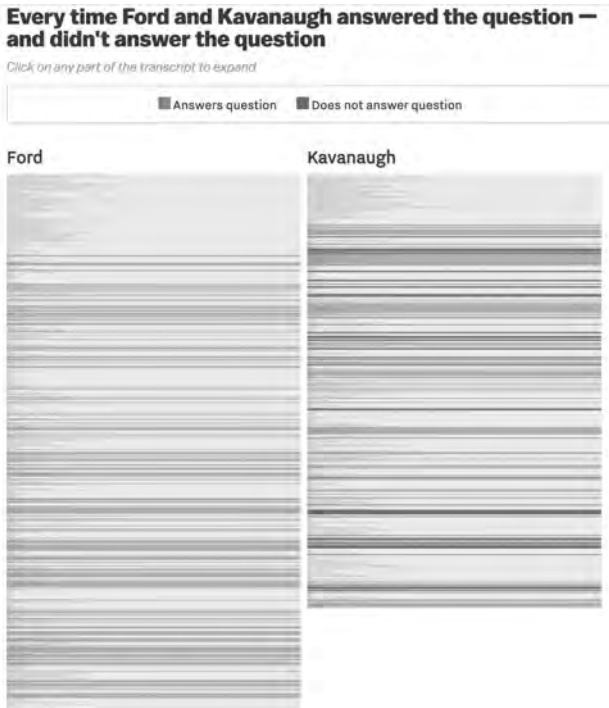


Figure 10.2 Chang's visualization of the Kavanaugh testimony for Vox.

This project was a good reminder that useful data analysis doesn't have to involve complex statistical methods. Chang published the story, took a shower and, when he got back to his computer, the visualization had gone viral and was being shown on several TV news shows.

Andy Boyle, former director of product engineering at the *Chicago Sun-Times*, and his data team tackle the issue of homicides in the Windy City on both a daily and long-term basis, which presents a variety of challenges.

Their Chicago homicide victims database lists the names of everyone killed by another person in the city of Chicago and is updated multiple times a day.

"While these dots on a map represent data points, what they really represent are human beings who've lost their lives," Boyle said. "They represent families, friends, people who feel the pain from the loss of a loved one. That's what we're hoping also gets conveyed when you look at this data, the human impact."

People are killed almost every day in Chicago, and the rates have gone up substantially since 2019. The *Sun-Times* has a team of dedicated reporters who work around the clock covering shootings, homicides and other breaking news events,

Tips for Creating Data Visualizations

From **Alvin Chang, *Guardian US***

- **Remember that data isn't just numbers**; the numbers are descriptions of things in the real world. This is why it's so important to first understand what each column of your data is describing.
- **Your data viz doesn't have to tell people what to think**, but your design decisions should tell people what to look at. Not making that decision is also a decision – but often the worst one, because that means you haven't fully grappled with what the reader should learn from your visualization.
- **Simple charts** are often the best. Bar charts and line charts are great.
- **There are many guidelines** for data visualization, but nothing beats showing your visuals to a bunch of people and asking them if their experience is what you intended. Sometimes, you can follow all the rules and still fail to communicate what you intended.
- **It can feel overwhelming** to think about the technical skills you need to learn in order to make a data visualization. But learning something new can be a wonderful experience, even if the end product isn't exactly what you had hoped to create. If you're willing to be bad at something, that means you're on the path to being good at it.

who spend time filling out a spreadsheet tracking the homicides as we learn about them (Figure 10.3).

Sometimes, Boyle said, they get the information from hearing about it over the police scanners and sometimes through press releases. Sometimes they don't find out until days later, when the body shows up at the Cook County Medical Examiner's Office.

Inaccuracy and discrepancies in how police classify homicide data is another issue. For instance, any homicides that occur on Chicago expressways aren't investigated by Chicago police but the Illinois State Police. So when the city of Chicago reports its homicide statistics, it does not include the homicides on the expressways, even though they occur well within city limits. They also don't always include cases the Medical Examiner's Office has deemed a homicide, but the Chicago Police Department either isn't the investigatory agency or disagrees with the medical examiner's office's findings, Boyle said.

"Our job is to try and give people the most accurate account of how many people were homicide victims within the city's limits," he said. "While our tracker doesn't track every single one – and sometimes we do miss them – our goal is to try and get as many as possible" (Figure 10.4).

The database updates regularly, so within 10 minutes of a new homicide being added, the information flows into the graphic. They try to track as much

Figure 10.3 *Chicago Sun-Times* homicides victims database.

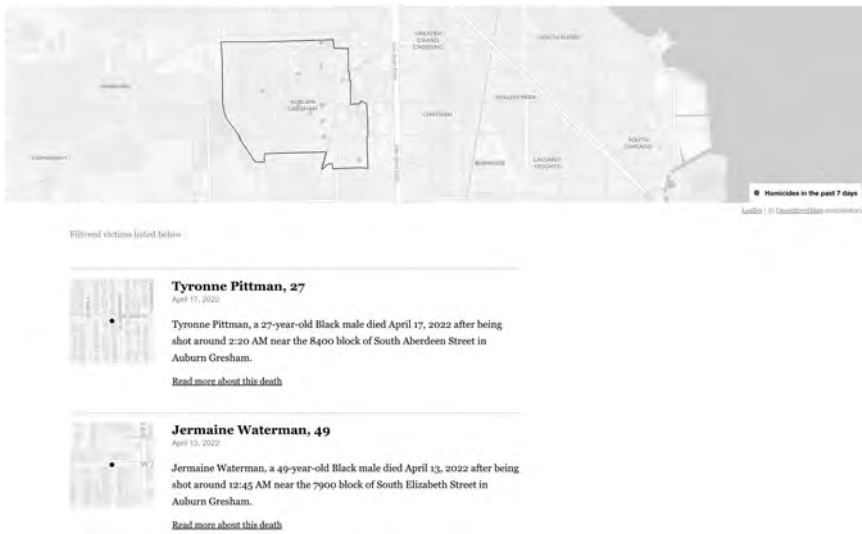


Figure 10.4 *Chicago Sun-Times* homicides database: Profiles of victims in Auburn Gresham neighborhood.

information as possible about the homicide: Where did the incident occur? What was the cause? Shooting? Stabbing? Other? What was the person's ethnicity or race, their age, their gender?

Readers can filter this information, so they can look at individual community areas, which is a Chicago statistical area that's more or less like a large neighborhood grouping (Chicago has 77), or drill down by some of the other information.

Creating Data Graphics for TV

John Walton, BBC News

There are a lot of similarities when working on graphics for broadcast and working on graphics for smartphones. Although broadcast tends toward landscape and smartphones tend toward portrait, both platforms have very limited space.

For simple charts, reducing using larger fonts and writing shorter titles aids reader understanding. You also can make the same basic chart usable for both digital and broadcast platforms when using graphics produced from code.

The BBC Data Team doesn't produce all the charts that get used on the broadcast by any means, but when they have a script set up and can offer colleagues a version of something they have made online, they pass it on for broadcast.

Although online has the advantage of more space, which means graphics can be a little more detailed, broadcast has the great strength of having a human voice to talk viewers through what they're seeing, which can be invaluable to aid understanding.

LEARN MORE: Walton recommends watching videos of data visualization expert Hans Rosling for a master class on how to talk to people about graphics. Rosling has done many TedEd talks on data visualizations, including this one (Shortlink: https://bit.ly/roslingvideo).

Becky Dale and Nassos Stylianou, data journalists on the BBC's Data Team, built a project in 2021 for both the BBC digital and broadcast departments. "Climate Change: World Now Sees Twice as Many Days over 50C" was developed as part of a BBC World Service series of programs and reports looking at climate change.

To create a data story for the series, Dale and Stylianou examined climate data to test the hypothesis that the number of days seeing temperatures of over 50 degrees Celsius (122 degrees Fahrenheit) was rising.

Walton said the guiding principle was to make climate change, which can sometimes be rather technical, readily understood. So, choosing the metric of days experiencing highs of over 50 degrees Celsius was a way of framing the story

in a familiar way rather than expressing climate change as an increase in temperatures against a long-term average above preindustrial levels. While the idea was simple, execution was not. It required the reporters to analyze more than *15 billion* data points.

"Their analysis had not been attempted by any other news organization, or indeed, by climate scientists," Walton said.

> So, Becky and Nassos had to work out a robust methodology to tackle the question. This led to them seeking support and advice from the University of Oxford, Berkeley Earth, Carbon Brief, the University of Reading, the UK Met Office and the Copernicus Climate Change Service.

For more data visualization tips, tricks and exercises, visit the Data + Journalism blog at http://dataplusjournalism.com

They also had to work with file types and data formats specific to climate science. The analysis and visualization were produced in 19 languages, for the BBC World Service, including Arabic, Persian, Punjabi, Chinese and Spanish.

Having delved deeply into a difficult and demanding piece of data analysis, Dale and Stylianou took a step back and thought about how best to convey their findings in a simple and engaging way to a general audience, Walton said.

Working with designers, they decided on a time series displayed on both a map and a bar chart at the same time. The map and the bar chart animate in tandem to show how many days had experienced these extreme temperatures and also to show how many areas around the planet were affected (Figure 10.5).

Besides their work on the *Chicago Sun-Times* daily homicide victims database, Boyle and data visualization developer Jesse Howe also build long-term projects with homicide data. At the end of the calendar year, the *Sun-Times* compiles all of the homicide data for that year and presents a series of graphics with an in-depth story.

Their project at the end of 2021 focused on Chicago's most dangerous neighborhoods. They built a series of interactive choropleth maps that showed homicides by police districts and by neighborhoods. They built a line chart that compared the city's safest and most dangerous neighborhoods for homicides.

They also produced a line chart comparing the last decade of annual homicide data that showed 2021 – with just shy of 800 homicides – ranked as one of the worst in several years. Another line chart underscored another dangerous trend: Homicides peak during the hot, humid summer months in Chicago.

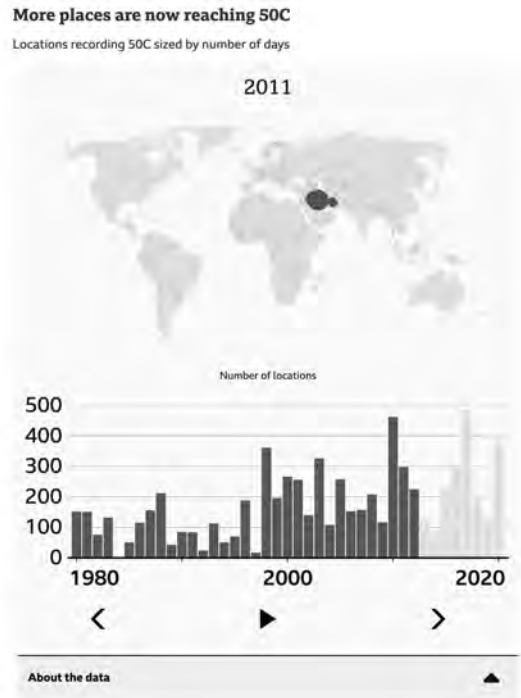


Figure 10.5 BBC time lapse chart on 50-degree Celsius days.

Creating Data Visualizations

Tips from John Walton, BBC News

- **At the start, ask yourself: Will my data visualization tell the story better than words?** If not, ditch it. Graphics take time and effort to make, so it is best to only produce them when they will definitely add something to your story or project. A picture paints a thousand words, and your graphic should do the same.
- **Before you start cleaning**, sorting, filtering and analyzing the data, remember to always keep an original copy for reference, preserved in its pristine and unaltered state.
- **Think hard about your audience and be empathetic.** Will they understand your graphic? Average levels of number literacy may not be as high as you think, and your audience may be less used to reading charts and visualizations than you and your colleagues.

- **Where will people be looking at my data visualization?** Is your audience most likely to be looking at your chart on a phone, desktop or on broadcast? Each platform gives you a different canvas to work on. Mobile and broadcast demand clear and simple graphics, while desktop and print can offer more scope for detail.
- **Sometimes, data may just be too flawed to be of any use.** It may be incomplete, too old, collected poorly and in a partial manner. Whether you want to use data of this kind is a judgment call that can vary project by project.
- **If in doubt, always ask yourself:** "Does writing up or visualizing this data help people's understanding of the topic at hand?" If the answer's a straight "no," then it may be best to find another way to tell that particular story.

Building Charts and Maps

Lena Groeger, an award-winning visual journalist, developed a framework for designing infographics that can pack a punch but still be easily understood by the viewer. She says whenever you are designing an infographic, ask yourself two simple questions:

- What is the audience supposed to understand, visually?
- What is that visual perception supposed to represent?

Take a line chart. When a reader looks at a line moving across an axis, what are they looking for? Quite simply, whether the line goes up and down. Second, what is that up or down line supposed to represent? The chart's title, caption and axis labels make this clear.

An upward-trending line, typically, indicates an increase in whatever variable is named in the chart description. A downward line, of course, represents a decrease. In a real line chart, this line will most likely trend both up and down, but the audience is still able to ingest the information at a glance. That's one of the main reasons behind representing information in a visual form to begin with.

These are not hard-and-fast rules. Some line charts, for instance, use an inverted *Y*-axis, meaning a downward line indicates an *increase* in the variable. But these are rare and can be confusing to the viewer. Even if you do not *have* to model your chart along the most typical methods, it is the simplest way to convey information to the audience.

If you find your own chart confusing, or are looking for the clearest way to convey information, ask yourself: What is the audience seeing at a glance? What are they supposed to take away from that glance? Sometimes this framework can help you figure out which kind of chart to use – a pie chart, a bar chart, etc. Often, this will result in cutting out extraneous information, just like you would cut unnecessary details from a text story.

Cairo wrote in his book, *How Charts Lie*, that to understand a chart, readers must focus on "features that surround the content and support it – the chart's scaffolding – and on the content itself (how the data is represented or encoded)." A chart or map's "scaffolding" can go a long way in building trust with readers, an issue we explore in-depth in Chapter 11. For instance, during the COVID-19 pandemic, social media feeds filled with fake news charts and maps misrepresenting data for various countries, states, etc. Many of these graphics had little or no scaffolding and rarely had a link to source data or a credit. Without those key elements, readers should question the graphic's credibility.

Data journalists must consider form, function and scale when designing charts and maps: The function is the data, conveying the information to the reader. The form is the presentation of the data in pictorial or graphic format. Cairo believes that the questions the designer means for the reader to ponder dictate the form or, at the very least, constrain it to a limited set of choices. Because the function limits the form, the formula for a graphic's success is simple: Bad function (data) = bad form (graphic).

Scale matters as well. A dot on a bubble map must be sized in proportion to the data and other points on the map. The same is true for a bar chart: If data in a bar chart showing COVID-19 positive cases shows China has four times as many cases as the US, the bar for China should be four times as long as the bar for the US. Skewing that, Cairo argues, is to mislead the reader.

"The same way that a long, deep and complex sentence can't be understood in the blink of an eye, charts that display rich and worthwhile information will often demand some work from [readers]," Cairo wrote.

That's why titles, introductions, footers with credits and source information, annotations, legends and scales are so important for charts and maps. They expedite the reader's understanding.

Once you're ready to create a visualization for a story, follow a simple process:

1. Think of the idea: What story do you want to tell?
2. Research the story and pull datasets.
3. Sketch what you want to do with the graphic.
4. Find a template (or a blank graphic) on the data viz software you want to use.
5. Begin your design, prioritizing the most important datasets at the top.
6. Add titles, descriptions, credits and sources to the chart.
7. Publish and export it, or download it as a png image.
8. Post to your site and share over social channels.

Choosing the Right Chart

For journalists who are new to data visualizations, choosing the right chart to represent the data can be a challenge. When do I use a bar chart? A line chart? Is it best as a map? Or do I need a chart at all – is the data better presented as a database, list or written into the story?

These are all questions journalists face when planning charts and infographics for their stories. There are a few approaches to take.

1. Before opening any software, sketch your graphic on scratch paper first. Think through the *X*-axis and *Y*-axis. Use labels. Even write the title. This will bring the chart into focus. This is an important, often-overlooked step that can save time in the long run.
2. Try testing a dataset in the Explorer tab in the lower-right corner of Google Sheets or in the charts tool in Microsoft Excel before loading the data into other software. These spreadsheet visualization tools do a good job of evaluating the data and selecting a chart to match. This is a quick, effortless step that can save you much time down the road.
3. Look at chart-building tools such as Flourish.studio or Datawrapper.de for clues on how to structure a dataset to fit that particular template. Flourish templates all contain "dummy data" – raw data that has been made up – that helps you see the correct format to set up your spreadsheet.
4. If you are just starting out in a newsroom, ask for help from the graphics editor or a veteran data journalist.
5. Try using a visual guide, such as the Financial Times Visual Vocabulary cheat sheet, that is available as a handout, poster and more on Github. There are many other tools for chart selection on the Journalist's Toolbox.

Choosing Colors and Fonts

Gradients and color shades are key to choropleth maps. For example, if you're building a map of COVID-19 positive cases by country, the countries with more cases should be a darker shade than those with fewer cases. This, along with a legend, helps the reader understand the density of cases. The same holds true with a bubble map. The larger the bubble, the higher the density. But the bubble size must be proportional to the data in the other countries, states, cities, etc., so as not to misrepresent the data.

Some of your color choices are simple: Green and earthy tones for environmental graphics, for instance. But try to avoid other color combinations: Red for homicide and other crime statistics as readers identify it with blood, red and green mean Christmas to many readers and red and yellow are identified with McDonalds.

While choosing colors for your visualizations is hard, choosing colorblind-safe colors is harder. In a three-part blog post, Datawrapper offers great advice for taking colorblindness and color weakness into consideration.

For example, a red- and green-blind person has problems distinguishing between red, green and brown, if their brightness is the same. Red and green might look gray to people who are red- and green-colorblind. Same with purple and blue.

Datawrapper, Flourish and many other chart-making tools take this into consideration and offer colorblind options when building charts and maps. The tools can usually be found on the main interface of your graphic workspace.

Chart-Building Tools

There are hundreds of chart and infographic creation tools on the Internet. Many are free, or some cost a nominal fee.

Adobe Illustrator is a vector-graphic building tool that has existed since 1985 and remains popular today. It's costly, but Adobe offers free trials through its Creative Cloud.

RStudio, which we explored in a previous chapter, offers chart-making features as part of its open-source integrated development environment for R. But for newbies, the point-and-click tools are a good start.

Here are a few of the easiest to use:

- Flourish and Datawrapper are among the most versatile and popular tools, and they offer free and paid versions.
- Infogr.am, Venngage and Canva offer basic infographic tools and are very intuitive. The Google Public Data Explorer lets you build graphics by linking directly to datasets, so no coding or spreadsheets are required.
- Tableau Public gives users more versatility by creating multiple graphics and maps from a single dataset and then merging them together into a single interactive "dashboard" that can be filtered or sorted to focus on a specific part of the data (city, ZIP code, etc.). In essence, it combines databases, charts and maps into one interface and offers advanced storytelling features. The tool was popular for building COVID-19 dashboards during the early days of the pandemic.
- **Mobile tools:** Phone and tablet apps such as Chartistic, Icongraph and Viz provide infographic and static chart-making tools. Canva also has a phone app, although the desktop version offers more control over design and is more intuitive to use.

Chart Exercises

We have prepared several chart-building exercises for you on these YouTube videos (shortlink: https://bit.ly/djvideos). Simply select the tool you want to learn, and follow the directions in the video to set up the account, download the data and exercise and produce the chart. All data used in the examples are real, so you can publish and share your work when you are done.

1. Flourish chart
2. Datawrapper bar chart
3. Venngage infographic
4. Tableau Public chart and dashboard
5. Tableau Public: Adding filters to a dashboard

Find all of the training videos at this link: https://bit.ly/djvideos, and experiment more with Flourish charts on this exercise: http://bit.ly/googleflourish

Mapping

Like charts, a great map can tell a story on its own. A well-crafted interactive map distinguishes itself from a print map in many ways. First, the map needs to reward the reader for clicking on it, whether it be a pinpoint, search button, zoom bar or a shape. When readers click, they expect to find more than just a number that we would see on a two-dimensional map. So a good interactive map should contain a combination of the following:

1. Multiple data points about the location
2. Address or longitude–latitude
3. A short paragraph about the location or a link to read more
4. Multimedia, if available: photos, video or audio

A map doesn't need all four elements, but a combination of at least two, to give the reader a deeper understanding of an issue. For instance, simply placing pinpoints of homicide locations on a map of Chicago isn't enough. Layering in a shapefile of Chicago neighborhoods or ZIP codes, making it searchable and adding a background paragraph or link to a news story from each pinpoint create a much more robust story.

Types of Maps

There are dozens of different types of maps that you can use to tell a story, but these are some that are most commonly used by journalists:

- **Pin map:** The most common type of interactive map uses geocoding to assign map coordinate locations and provide data about that location. This is typically done through loading a spreadsheet into a tool such as Google MyMaps, Carto.com or another mapping tool. The map will geocode city names, names of buildings, street addresses, longitude–latitude, postal codes and more.
- **Cluster map:** Also known as a bubble map, a cluster map is an excellent alternative to a pin map that might have several tightly packed markers together in one area, according to Maptive. So instead of a large conglomerate of pins, you get a clean cluster icon that displays key information and corresponds with the number of markers contained in it. Zooming in reveals individual markers that are clickable.
- **Choropleth map:** Popular during the COVID-19 pandemic, these maps shade areas based on a value. For example, a country with more positive COVID-19 cases is shaded darker than a country with fewer. A legend typically shows the gradient scale and range of the data. They're common because they are easy for readers to understand (Figure 10.6).
- **Symbol map:** This map appends symbols that show location or another form of data that can be applied to geographical locations. Tableau Public cites symbol maps as a way to point out the cities that have been hit by hurricanes

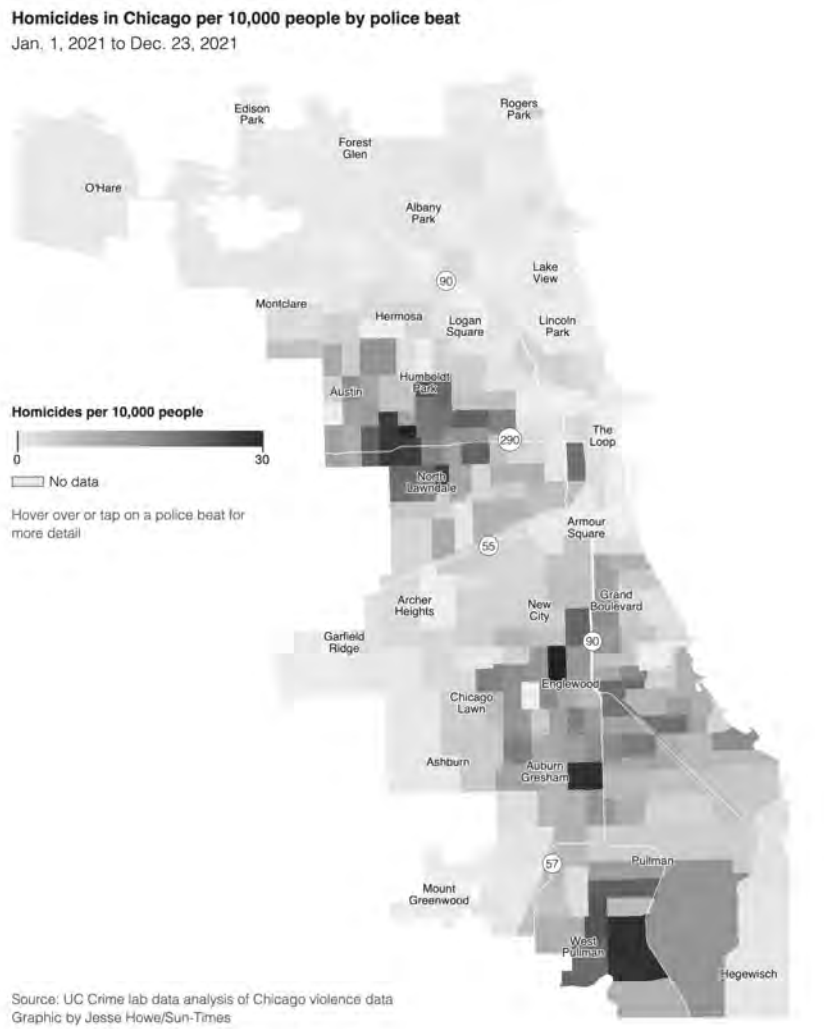


Figure 10.6 *Chicago Sun-Times* choropleth map shows which Chicago police districts have the most homicides per 10,000 people in 2021.

throughout a period of time with each symbol sized (scaled) to indicate the total number of hurricanes.

- **Locator map:** This map shows the location of a particular geographic area within its larger and presumably more familiar context. The map can be used on its own or as an inset for a larger map. For instance, highlighting Ireland in a map of the world to show its location.

- **Heat map:** Unlike choropleth maps, heat maps are not tied to a boundary. Instead, they have a color code using the density of points. If there is a high density of crimes in an area, it shades the area as red. Fewer cases are orange, yellow, etc.

Geographic Information System Maps and Shapefiles

A Geographic Information System, or GIS, is a visual representation of quantifiable data that allows you to layer information on top of the map. These are usually shapefiles, such as Keyhole Markup Language (.KML) or another format that represents a shape such as a state, county, neighborhood, ZIP code or some other geospatial shape. They can be layered behind pinpoints on a map, such as homicides by a specific police precinct. Or they can be shaded to show density.

Building Maps: Tools

The Journalist's Toolbox website lists dozens of free and low-cost mapping tools. Some of the most common mapping tools used by journalists include ArcGIS, the cloud-based mapping software from Esri; Carto.com (great for animated time-lapse maps); Google MyMaps; Open Street Map; MapBox; Maptive; StoryMap JS and Tableau Public, among many others. Flourish and Datawrapper are among the most popular as they have simple choropleth, symbol and locator map templates that are intuitive and easy to use.

MyMaps is one of the most basic tools for map-building. It will handle a spreadsheet of data up to 1,000 rows and geolocate the addresses or other location data in the map. It allows for layering shapefiles and offers nine different base maps. Developers have control over shapes of the pinpoints and colors and can easily integrate text, images and videos into the pinpoints. The exercises in this chapter will get you started with this tool.

* * *

Exercises

We have prepared several map-building exercises for you on these YouTube videos (shortlink: https://bit.ly/djmapvideos). Simply select the tool you want to learn and follow the directions in the video to set up the account, download the data and exercise and produce the chart. All data used in the examples are real, so you can publish and share your work when you are done.

Google MyMaps
Datawrapper choropleth and locator maps
Tableau Public filtered maps
Find all of the training videos and more on this YouTube playlist: https://bit.ly/djmapvideos

Tools Used in This Chapter

Flourish https://flourish.studio/
Datawrapper https://www.datawrapper.de/
Infogra.am https://infogram.com/
Venngage https://venngage.com/
Canva https://www.canva.com/
Google Public Data Explorer https://www.google.com/publicdata/directory
Tableau Public https://public.tableau.com/en-us/s/
Chartistic https://apps.apple.com/us/app/chartistic-charting-app/id1127272574
Icongraph https://www.icongraph.com/
Viz App (iOS only) https://apps.apple.com/gb/app/viz/id678141110

Google MyMaps https://mymaps.google.com/
Open Street Map https://www.openstreetmap.org/
ArcGIS https://www.esri.com/en-us/arcgis/about-arcgis/overview
Carto.com https://carto.com/
Open Street Map https://www.openstreetmap.org/
MapBox https://www.mapbox.com/
Maptive https://www.maptive.com/
StoryMap JS https://storymap.knightlab.com/

* * *

Footnotes

Vox, We Can Draw School Zones to Make Classrooms Less Segregated: This Is How Well Your District Does https://www.vox.com/2018/1/8/16822374/school-segregation-gerrymander-map

AlbertoCairo.com http://albertocairo.com/

Infographics Formula by Alberto Cairo https://mindthegraph.com/blog/infographics-alberto-cairo/

BBC UK, How Many Coronavirus Cases Are There in My Area? https://www.bbc.com/news/uk-51768274

Chartbeat, Most Engaging Stories of 2020 https://blog.chartbeat.com/2020/12/22/most-engaging-stories-2020/

BBC, Climate Change: World Now Sees Twice as Many Days Over 50C https://www.bbc.co.uk/news/science-environment-58494641

Chicago Homicides Victims Database https://graphics.suntimes.com/homicides/

Chicago Sun-Times Year-End Violence Story and Graphics (2021) https://bit.ly/csthomicides

Lena Groeger, What is Wrong with This Chart? http://lenagroeger.s3.amazonaws.com/newschool/WrongWThisChart.pdf

Alberto Cairo, How Charts Lie, pages 24, 34–36, 40 and 47

American Journalism Review: Journalism Professors Used Legos to Teach Super Bowl Data Visualization https://ajr.org/2015/02/02/journalism-professors-used-legos-teach-super-bowl-data-visualization/

Financial Times Visual Vocabulary Cheat Sheet https://github.com/Financial-Times/chart-doctor/tree/main/visual-vocabulary
Journalist's Toolbox: Choosing the Right Chart https://www.journaliststoolbox.org/2021/11/25/choosing-the-right-chart/
Datawrapper: How Your Colorblind and Colorweak Readers See Your Colors https://blog.datawrapper.de/colorblindness-part1/
2022 Chicago City Budget Treemap https://public.flourish.studio/visualisation/8988206/
Lucid Charts: Venn Diagram Tutorial https://www.lucidchart.com/pages/tutorial/venn-diagram
OrgCharting.com: Hierarchy Charts http://www.orgcharting.com/what-is-a-hierarchy-chart/
Journalist's Toolbox: Charts and Infographics https://www.journaliststoolbox.org/2022/05/16/online_journalism/
Tableau Public Dashboard: Chicago COVID-19 Cases by ZIP code http://redlineproject.org/timeline/coviddashboard.php
YouTube: Building a Flourish Chart https://www.youtube.com/watch?v=qD2SBLvmuWE
YouTube: Datawrapper Bar Charts https://www.youtube.com/watch?v=pHm7V9CaKfM
YouTube: Venngage Infographic Exercise https://www.youtube.com/watch?v=a4xOQXFddVU&feature=youtu.be
YouTube: Tableau Public Chart and Dashboard https://www.youtube.com/watch?v=vTedRGIlMGY
YouTube: Tableau Public – Adding Filters https://www.youtube.com/watch?v=QbmEhUNgX9s
Google Flourish Exercises http://bit.ly/googleflourish
Earth Observing System https://eos.com/blog/gis-mapping/
Tableau Public: Symbol Maps https://www.tableau.com/data-insights/reference-library/visual-analytics/geospatial/symbol-maps

11 Ethics, Trust, Transparency and Posting Data Online

Mike Reilley

Ethical decision-making with data stories can be tricky, so let's start with an exercise.

Let's say you are a data journalist at a local news outlet. You just downloaded your state's or region's gun license database from a data portal. As you sort through the spreadsheet, you see that you have addresses of everyone throughout the state/region who has been approved for a gun license (or in states like Illinois, a Firearm ID card). You decide that you could map this dataset and post it to your media outlet's website. You then decide to make the dataset searchable by posting it in a searchable template in Google Flourish or Datawrapper.

You embed both the map and the database in your content management system and publish them. You share them on social channels, post on Reddit and link to it off your site's home page. You go home feeling good that you've provided a great service to your community, right?

Fast-forward to a few weeks later. Your crime reporter stops by your desk and mentions a series of break-ins to homes in your community. The thieves are stealing guns. During questioning, police ask one of the suspects they caught how they knew where to find the right homes. The answer: Your map and searchable database of gun licenses.

This anecdote underscores the importance of thinking through not just the accuracy of data but what your audience can do with the data or visualization once it's published. Will it be used for good? For harm? Could it endanger someone?

These are tough, but necessary, questions to ask when planning data projects, particularly those involving public safety and crime. There are many resources that offer broad guidance when reporting with data. The Society of Professional Journalists (SPJ) provides the oldest and most broad-based code of ethics available online or in downloadable, printable PDFs.

In the case of the gun licenses, the SPJ Code of Ethics addresses a few key areas:

- **Balance the public's need for information** against potential harm or discomfort. Pursuit of the news is not a license for arrogance or undue intrusiveness. In this case, the public's need to know where the gun owners lived was not critical to any story. The map and gun license database had no context. There was no story on an increase in crime in the area, and even then the

DOI: 10.4324/9781003273301-12

need for a database is highly questionable. What public service does it serve? Does the public really need to know this information?

Had the reporter been doing a story on an increase in gun licenses in an area of a city or region, analysis of the gun database would be appropriate. But the results *only* should be shared in the aggregate: "Cook County had a 20 percent increase in gun license applications from 2021 to 2022." By publishing the locations of the gun license applicants, it gave criminals a road map to find them, a classic example of causing harm.

- **Provide context. Take special care not to misrepresent or oversimplify in promoting, previewing or summarizing a story.** With the gun database, a simple phone call to local authorities would provide some much-needed context: Most crimes aren't committed by legal gun owners but by criminals using stolen or black market firearms. By posting the database and map without context, it misrepresented the issue.

Data also can be manipulated and misrepresented just like any fact or quote. Journalists should always show extra care if working with data, as mistakes and ethical issues can arise in ways that are different from other stories. The SPJ Code of Ethics addresses this: *Never deliberately distort facts or context, including visual information. Clearly label illustrations and re-enactments.* While this applies to photojournalists as well, data journalists should take this entry to heart.

The Reader on Data Visualization says that the topic of ethics in data visualization is not something that comes to the fore when we start working. It is rarely the case that one sets out to deceive without altering data. The topic of good ethics in data visualization is very important and it is the duty of the creator to take care of it.

Alberto Cairo, who teaches data journalism at the University of Miami in Florida, wrote a book, *How Charts Lie*, that addresses many ethical issues in presenting data visually. "Charts lie in a variety of ways – displaying inaccurate data, oversimplifying stories and suggesting misleading patterns – or are frequently misunderstood, such as the cone of uncertainty maps shown on TV every hurricane season."

To Cairo, data visualization is where journalism and engineering share a beer. Journalists who build graphics are "information engineers" similar to software developers or industrial designers.

"For us the information is as important as the effectiveness and efficiency of the displays we devise to convey it," Cairo wrote in a 2014 article titled "Ethical Infographics." "When we create a visualization, we're giving information a visual shape, we model it, we sculpt it. There should be a connection between the forms we choose and the tasks that our visualization is intended to facilitate.

"Creating a visualization isn't just an act of journalism, but also of engineering. To think about the ethics of news visualization – a field waiting to be

developed – is to go beyond the grand themes traditionally covered in the literature on journalism ethics and to explore matters of effectiveness and efficiency. Journalists who design visualizations need to address questions related to what they should display and why, but also pay in- creased attention to how they should display it. In other words, visualization designers must think about the structures, styles and graphic forms that let audiences access information successfully in every situation."

This means double-checking the *Y*-axis on a bar or line chart to make sure the scale doesn't misrepresent the data. It means area maps can misrepresent geography and density if not placed in proper context. And it means pausing for a moment to think about the impact the data, visuals or story could have on your readers or viewers."

The Importance of Transparency and Trust-Building

Marilia Gehrke, a professor at Universidade Federal do Rio Grande do Sul, presented a paper at the 2020 Computation + Journalism Symposium titled "Transparency as a Key Element of Data Journalism." The paper focused on the growing need for transparency in data journalism in Brazil and globally. Gehrke wrote:

> In recent years, especially because of disinformation, transparency has been an important part of journalists' work. Transparent conduct involves opening methods and procedures of reporting. It means, overall, showing the audience how the information was obtained and verified until being published.

She interviewed 36 journalists, most of them data journalists who have been working in newsrooms for more than 10 years.

"A significant part of the respondents believes that transparency must be shown every day to the readers. Among the main reasons to rely on transparency as an important value in their practice is the necessity to increase credibility in journalism and combat disinformation. Furthermore, this research suggests that transparency seems to be connected to objectivity and journalists' ethics, but not necessarily to the news outlets' rules."

Show Your Work

When former *Chicago Sun-Times* Director of Product Engineering Andy Boyle worked for rival newspaper *Chicago Tribune*, their internal newsroom software-oriented storytellers had adopted the mantra "Show Your Work," even printing it on T-shirts they wore in the office. The principle is that a data journalist's work should be public and something that people can scrutinize and test.

"It should be able to face rigor," he said. "If we're saying, 'This is an accurate representation of this data,' then the audience should be able to get the data themselves, analyze it themselves, follow our methods and come to the same conclusions."

In his role at the *Sun-Times*, whenever the team referred to an analysis of data, they shared a direct link to whatever it is they were analyzing, so readers could find the data themselves. Team members also wrote up their methodologies a few times, in case people wanted to follow along and do it themselves.

"If someone wants to say your work is bunk, it's harder for them to do that if they can go step-by-step and come to the same conclusions as you," he said. "Showing your work out in the open lends it much more credibility. Just like we try and use confidential sources only in very limited situations and try and get everyone to speak on the record, you should treat your data work as if you, the journalist, require your own work to be on the record."

"And if you can't explain how you got your analysis, you probably aren't in a good enough position yet to explain to your audience what the information means. Or if you feel you need to hide the way you've come up with your analysis, that's giving people less incentive to trust your work."

There are many ways journalists can give readers a peek behind the curtain and show them how data stories are built. They can post links to the raw datasets used in reporting the story. They can post their research and methodology to a GitHub page. They can link to datasets and the origin source from the footer of their graphics. They can post excerpts of their data diaries as sidebars to their main stories.

Some large data projects provide a box inside the article – jokingly called a "nerd box" – in many newsrooms, or even a footnote briefly explaining the methods. Journalists can also pare down a longer data diary, which we'll explore later in this chapter, into a Twitter thread to better explain how they reported, wrote and visualized the story.

David Eads posted on Vimeo and Twitter a 41-minute video where he and reporters from the Marshall Project walked viewers through their data analysis and visualization process for an award-winning investigation into K-9 police dog bites. He also provided a short story about the reporting process for the project, which the Marshall Project collaborated on with the *Indianapolis Star*, AL.com and the Invisible Institute.

Benefits to making data readily available can be far-reaching. For instance, the *Washington Post* in 2019 examined the prescription opioid epidemic in the US by building a database on the sales of millions of painkillers. The *Post* made the data and its methodology open to the public and encouraged reporters from other media outlets to localize it. Journalists in more than 30 states participated, including a piece examining the opioid crisis in Chicago and Cook County.

That openness helps expose flaws in reporting as well. Alvin Chang, head of visuals and data for *The Guardian US*, was horrified when he realized he made a mistake in a story.

"Not because I got it wrong," he said, "but because I was the only person who realized I got it wrong. It showed me just how much faith people put in data visualizations, as if putting it in chart form somehow makes it more valid."

"My data transparency efforts are often about trying to explain what the data is, what question we asked of the data, how we conducted the analysis and what results it yielded. When my readers talk about my data visualization to their friends, I want them to be able to tell a story about what data was gathered, how it was parsed and what the findings were. I also link to datasets and occasionally share code, but my concept of 'transparency'" is often for the majority of readers who aren't data fluent.

Lynn Walsh is an assistant director of Trusting News, which teaches journalists how to be more open and build trust with readers. A former investigative TV news producer, Walsh said journalistic transparency and "showing your work" are designed to engender trust by showing that journalists are willing to draw attention to matters that might influence their work.

In addition to disclosure, transparency can refer to a broader kind of storytelling around journalism's motives, values and processes. News consumers do not have deep knowledge of how journalism operates, as a 2018 American Press Institute study shows. And when there is a void of understanding, readers are not often giving journalists the benefit of the doubt. Investigations and data-heavy reporting often take time.

When news outlets publish these stories and say they spent months on them, people may wonder why they spent months on that specific topic. Instead of allowing them to wonder and come to their own conclusions (which would most likely be negative), they should explain why they choose to focus on the topic – by tying it to their values and goals – and also be transparent about how they did the reporting. Because these stories are also complicated and involve data, people will have many questions about why journalists chose that specific dataset or spoke to that expert or focused on that element of the data. Therefore, they must explain all of these to their users.

It has become common for journalists to do this by writing a "behind the story" piece to accompany long investigative projects. In these stories, an editor typically explains why a story was written and the work involved, lists team members and answers readers' questions. Those pieces are usually on a separate page from the main story, and often only the most dedicated readers will click through. Walsh and her Trusting News colleague, Joy Mayer, are firm believers in taking advantage of attention where it already exists – in the story itself.

For example, if part of a story says a source wasn't available for comment, journalists could explain how they tried to reach the person. When they introduce an expert source, they could include information about their independence and reliability. When part of a story led a reporter to consult a conflict of interest policy, describe the situation and link to the policy, Walsh said.

But there's a major drawback: All of the extra information can often disrupt a story's flow. Walsh recommends that reporters experiment with what their CMS can do to set off type so it looks different from the narrative story, or use a "nerd box" to highlight or link off to the information. Try not to let internal conventions and comfort zones get in the way of important trust-building opportunities. For example, Walsh's team worked with Gannett's regional-based data and investigations unit that built a WordPress tool that allows pieces of a FAQ to be embedded throughout a text story. The tidbits (dubbed "trust nuggets" internally) made their debut in a story by reporter Lucille Sherman about a lack of oversight of midwives in Oklahoma. The story is part of a larger project about out-of-hospital births.

Walsh said there are two wonderful things about the "trust nuggets":

1. **Each trust nugget offers actual information.** It doesn't just say "to learn why we started this investigation, click here." It gives enough background that the reader learns something about the journalists' values or processes without having to leave the page.
2. **The trust nuggets appear where they're most needed.** They anticipate that where the story introduces a new character, readers might be curious about why that character is included. Or where a dataset is cited, readers might wonder why that was the right data to rely on.

Joy Mayer, Walsh's Trusting News colleague, wrote on Medium about the technology behind the "trust nugget" tool and when transparency is most needed. She pointed out that they work for TV and radio, too, as WUSA built transparency into an investigation of Washington, D.C., police practices.

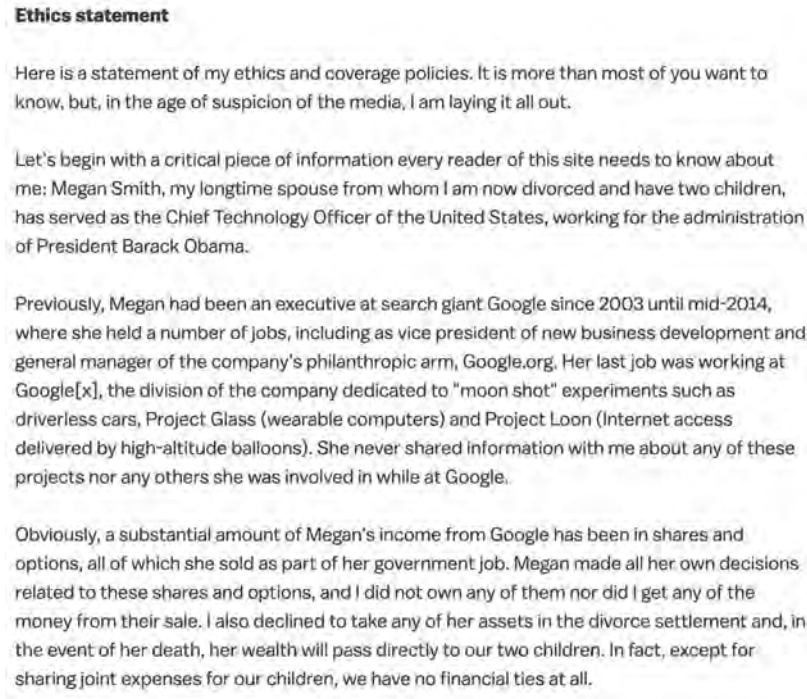
Readers don't mostly show up to your journalism automatically curious about how and why it was put together. They typically care only when the behind-the-scenes information relates to the news they're consuming in the moment, Walsh said.

"Consider how you feel about other institutions you interact with," Walsh said. "When do you care to learn more about the leadership of your local school district, the manager of your local grocery store or the policies of a local nonprofit? Only when they intersect with your interests or your concerns, right?

"You don't just show up at the grocery store wondering about the background of the manager, their corporate philosophy on charitable giving or the values that guide their hiring."

For example, Walsh said it might be appropriate to reveal that the subject of a story has also been a donor to a news organization. Or a journalist's staff bio might reference financial investments, community involvement or family connections to an issue of high interest. Walsh, former president and ethics chair for the SPJ, cited Kara Swisher's Recode bio as an example of great transparency (Figure 11.1).

Chang, at *The Guardian US*, cautioned journalists against placing too much trust in data. One of the flaws of data journalism is that it can't answer every question and is only one tool to describe the world and what's happening in it.



Ethics statement

Here is a statement of my ethics and coverage policies. It is more than most of you want to know, but, in the age of suspicion of the media, I am laying it all out.

Let's begin with a critical piece of information every reader of this site needs to know about me: Megan Smith, my longtime spouse from whom I am now divorced and have two children, has served as the Chief Technology Officer of the United States, working for the administration of President Barack Obama.

Previously, Megan had been an executive at search giant Google since 2003 until mid-2014, where she held a number of jobs, including as vice president of new business development and general manager of the company's philanthropic arm, Google.org. Her last job was working at Google[x], the division of the company dedicated to "moon shot" experiments such as driverless cars, Project Glass (wearable computers) and Project Loon (Internet access delivered by high-altitude balloons). She never shared information with me about any of these projects nor any others she was involved in while at Google.

Obviously, a substantial amount of Megan's income from Google has been in shares and options, all of which she sold as part of her government job. Megan made all her own decisions related to these shares and options, and I did not own any of them nor did I get any of the money from their sale. I also declined to take any of her assets in the divorce settlement and, in the event of her death, her wealth will pass directly to our two children. In fact, except for sharing joint expenses for our children, we have no financial ties at all.

Figure 11.1 Ethics statement from Kara Swisher's Vox Media bio.

"It's not more trustworthy than a human source," Chang said.

> The biggest mistakes I've made are when I tried to use data to answer a question that can't be answered with data, or when I've assumed the data is correct. When you're working with other people who are fluent with data, it's easy to talk through some of these concerns and pivot when necessary.

But what if you are the only data journalist in the room and your editors see data journalism as a magic answer machine? That puts a lot of pressure on you to produce an answer from the data. This is why it's so important to have more data-fluent people in newsrooms, especially in leadership positions. Chang said it's also why it's important to say "no" to editors if the data can't answer the question.

How to Post Data Online

Providing complete datasets gives readers the opportunity to pore over the raw figures journalists used in their story. Journalists have many options when posting data to the web.

Lynn Walsh's Tips for Building Trust with Data Stories

1. **Explain what investigative/data journalism is and isn't.** People have a lot of questions about how and why we choose to cover certain topics and stories. When we do not explain the why and how they make assumptions about it and often assume we are biased. Talk about why you invest time into these stories and how you choose which stories to select. Explaining this publicly can also help hold your team accountable for the type of journalism you want to produce. *The Seattle Times* recently answered these questions and more with a FAQ about its investigative team and how it operates.
2. **Be transparent about your process**, which includes showing your work and linking to the original data and documents. Make sure to include these explanations within the reporting or at least linked to the main story. You want to catch people's attention and curiosity while they are engaging with your content and not expect them to find a separate story and click on it later. Some elements are worth explaining: What data you used and why, how you used the data, how you analyzed the data, who you relied on to be experts and why, what you still do not know or are left wondering about and where the data came from.
3. **Be available to answer questions**, and remember, sometimes questions can be misinterpreted as criticism at first glance. Make sure you build in time to review comments on social media, emails, etc. It's really important that we pay attention to and engage with our users, and one of the best ways to do that is by responding to their comments, answering their questions and asking for their feedback.

   Some of the questions may be very basic, but if we want people to understand the sometimes complicated data, we may need to explain the very basic elements first. Also, sometimes questions are buried in criticism. That negative comment may be based on a misunderstanding of how the reporting process works. So, don't dismiss it. Instead, take time to respond and ask more specific questions to them to get at what they really want to know or do not understand. And if people do not understand elements of the story, take the time to produce FAQs or do a follow-up story that helps them understand.
4. **Make corrections.** We try so hard to not make mistakes, but sometimes they happen. If they do, make the correction quickly and publicly. Do not try to hide it or make the change without acknowledging it. Also, our reporting is done with the best information we have at the time of publication. If you receive new information after the story publishes or someone sees your story and can provide new details, do the update and link to the original story and update the original story.

While not a dedicated data publishing platform, GitHub is a very powerful, capable, free platform on which to publish high-quality data. It's popular among journalists for storing datasets and methodology for large projects, and the documents can be easily searched and downloaded.

Since it's free, any size newsroom – from one staffer to hundreds – can use the platform to share work. Some examples:

- **ProPublica:** https://github.com/propublica
  Links to data and methodology from several big investigative projects. ProPublica also stores its news app and data style guides on GitHub so others can learn from them.

- **The Guardian:** https://github.com/guardian
  Includes data and coding exercises

- **Vox:** https://github.com/voxmedia
  Stores hundreds of projects as well as hiring guides, programming examples and more.

- **BBC:** https://github.com/bbc/
  Offers open-source code used on public facing services, internal services and educational resources.

- **Buzzfeed News:** https://github.com/BuzzFeedNews/everything#guides
  Indexes all of its open-source data, analysis, libraries, tools and guides. It releases not only the reporting methods but also the scripts used to pull, clean, analyze and visualize data.

- **New York Times:** https://github.com/nytimes
  Resources range from data to immersive projects and has a very large repository for its in-depth COVID-19 data.

The easiest way to make a dataset available is to upload it in Google Sheets. You can use the Share settings in the upper-right corner to set it to public viewing and then link to the data from your story or share it over social channels. This gives the audience a chance to inspect the raw data in view-only mode, so it can't be tampered with.

Once your data is posted online as a Google Sheet, you can format it to be searchable in the Google Dataset Search tool, which we explored in Chapter 1. You must follow a set of criteria outlined on the tool's developer page (Figure 11.2) to get the dataset listed in the search algorithm, including details about the dataset, source, methodology and more. It's an excellent way to steer traffic to your data and, possibly, your story.

Some tools, such as Datawrapper, give journalists the option of making the data available through a clickable link in the footer of a graphic. This is a simple, easy way for readers to access the raw data (Figure 11.3).

Figure 11.2 Google Dataset Search posting criteria.

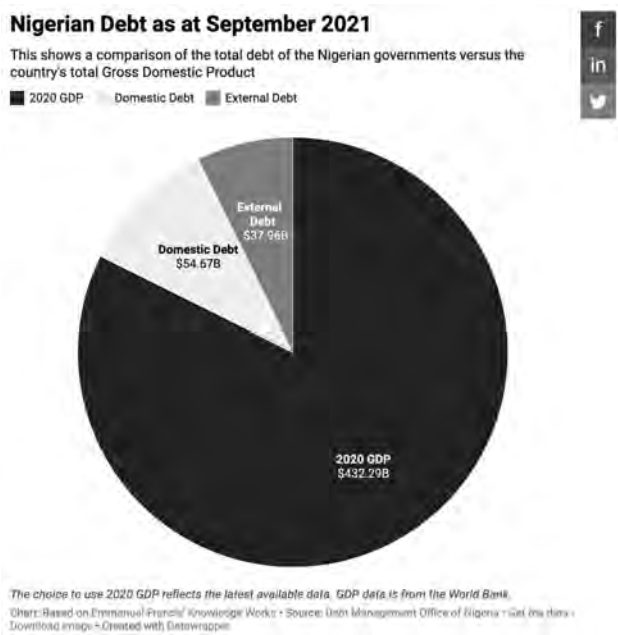


Figure 11.3 This open-source graphic from Datawrapper has a link to the source of the data and a "Get the Data" link to the spreadsheet.

USA TODAY COLLEGE FB COACHES SALARY DATABASE

Type in a school, state or conference to view the data.

Big Ten

Rk	School	Conf	Coach	Scheduled School Pay	Contract Year Pandemic Reduction	Actual School Pay	Total Pay	Total Pandemic Reduction	School Buyout As Of 12/1/20
4	Michigan	Big Ten	Jim Harbaugh	$8,054,000	$268,721	$7,785,279	$8,036,179	$554,584	$6,367,929
12	Ohio State	Big Ten	Ryan Day	$5,748,264	$101,570	$5,646,694	$5,651,694	$236,877	$45,464,910
13	Northwestern	Big Ten	Pat Fitzgerald	–	–	–	$5,218,658	–	–
14	Michigan State	Big Ten	Mel Tucker	$5,057,000	$99,750	$4,957,250	$5,057,250	$266,000	$23,186,513
17	Nebraska	Big Ten	Scott Frost	$5,000,000	$166,667	$4,833,333	$4,833,333	$166,667	$25,375,000
19	Iowa	Big Ten	Kirk Ferentz	$4,900,000	$229,250	$4,670,750	$4,670,750	$393,000	$20,721,250
24	Minnesota	Big Ten	P.J. Fleck	$4,600,000	$318,320	$4,281,680	$4,281,680	$548,320	$18,697,250
28	Wisconsin	Big Ten	Paul Chryst	$4,250,000	$318,750	$3,931,250	$3,983,750	$318,750	$15,328,333
32	Illinois	Big Ten	Lovie Smith	$4,000,000	$200,000	$3,800,000	$3,800,000	$200,000	$2,600,000
33	Indiana	Big Ten	Tom Allen	$3,770,000	$0	$3,770,000	$3,770,000	$50,000	$20,550,000

1 / 2

Figure 11.4 College football coaches salary database made with Flourish.

Flourish's table and database templates make it easy to upload data to the web, make it searchable and embed in a story. For example, the *USA Today* college football coaches' salary spreadsheet was uploaded to Flourish's searchable database template and published in a matter of minutes. Readers can search the database (Figure 11.4) by searching by name, conference or state. Datawrapper and Airtable also offer a searchable table template, and Flourish and Datawrapper have social media sharing buttons so readers can quickly upload links to the databases to send to social channels.

Tableizer is another free tool that can post data quickly to the web. Just cut and paste the data from your spreadsheet into the interface, select the font and header color you want in your table and hit the Tableize It! button. (Figure 11.5). You'll get JavaScript code to embed the table in your site and a sample of what the table will look like.

Figure 11.5 Tableizer with Chicago's annual homicide rate data in it.

Sharing Data on Social Media

Social media channels provide another medium to share data with an audience. Several months into the pandemic, Cherone started tweeting COVID-19 data daily on her @heathercherone account she uses for sharing WTTW news. She pulled data from the Chicago Department of Public Health and weaved in context from her reporting and covering announcements from the mayor's office and other sources.

Cherone "reverse engineered" the daily data updates on the city's dashboard to give readers a clear look at the data Governor JB Pritzker and Chicago mayor Lori Lightfoot were using to base public health decisions.

"I didn't do these daily updates at the start of the pandemic because we didn't have good data right away," Cherone said.

> "I started doing them when the second wave hit in the fall of 2020 and winter of 2021. That was because Pritzker and Lightfoot sent a really clear message about what would trigger increased restrictions for bars and restaurants, or if we would be able to open gyms and schools."

For more examples of how to share data on social media, visit the Data + Journalism blog at http://dataplusjournalism.com

The biggest challenge Cherone faced: "How do I make it readable for people and engaging?" She started by featuring the rolling, seven-day positivity rate for the city, then used arrows for bullet points to work other data into the posts, such as the number of cases, tests, hospitalizations and deaths (Figure 11.6).

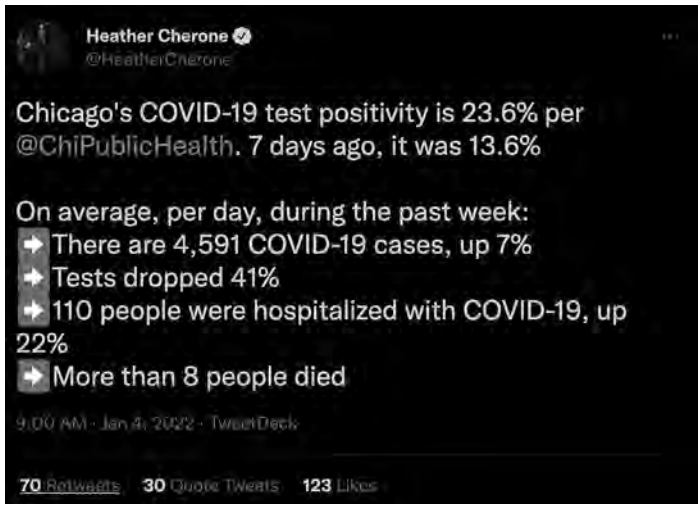


Figure 11.6 Heather Cherone's daily COVID-19 data updates.

"The test positivity was important because the governor was using that to Institute or lift rollbacks and the number of cases was important because the city was using that to either add or remove restrictions," Cherone said.

> "I knew that those two had to be there. Then it became clear that hospitalizations were also going to be especially key in the summer of 2021 because the governor reinstituted a mask mandate because of increasing hospitalizations. I added hospitalizations per day because I think it's really easy again to get lost in the data and percentages and not focus on the fact that this disease was killing two or more people a day."

Readers engaged with the tweets from the start, asking Cherone questions about the figures and for more context. Many questions came in about the confusing death rate, which was presented as a decimal because it was a rolling seven-day rate, so Cherone explained how the data was calculated and provided a link to the data dashboard for readers to review (Figure 11.7).

"A lot of people were really confused about the math involved there so I have a tweet that I did and I just use that one to respond to people saying, 'How could 6.8 have died?'" she said. "I think it's helpful to share the data with people because they care and are trying to make the right decisions for themselves and their families."

Data Diaries and Transparency

When Boyle works on a large project, he usually keeps a text document open where he walks through all of his steps. These "data diaries" can be helpful for journalists – and fellow team members – to retrace their steps on an analysis over months of research.

Figure 11.7 Cherone responds to Twitter followers who have questions about the city's COVID-19 death rate.

In the diary, Boyle addresses questions such as: Where did I download the data from and when? Where did I upload it for analysis? What sort of code did I write to actually do the analysis?

"I try and keep it all step by step, so someone else can go through everything I do – or, more importantly, I can go through the steps again myself – and get the same answers," he said.

> "It's basically the scientific method. Your work should be reproducible. Just like in journalism, we try and explain who told us a fact, or what document we got information from. We try and do the same when it comes to analyzing and sharing data."

Those data diaries can be used for more than internal purposes. They can be reproduced as short sidebar stories or "nerd boxes," which Walsh discussed earlier in this chapter. They also can be reproduced in short bursts on social media, such as a "Twitter thread."

Journalists use social media threads for many purposes: Explaining how a complex data story was reported, sharing a running narrative, live-tweeting, descriptive writing and more. When done right, a thread can engage a social media audience in a very meaningful way.

Twitter threads are an effective way for data journalists to explain the research, reporting and visualization process to readers. If the reporter or designer is keeping a data diary – a Google or Word document with a running narrative of the reporting, writing and visualization process – those diary highlights can easily transform into a Twitter thread. In other words, it's the data journalist's chance to show how the sausage is made.

As a rule, most Twitter threads are around 10–15 tweets, but you can make it shorter if need be. Weaving in links, photos, graphics, etc. helps visualize the thread and increase understanding for the reader. The graphics are particularly important as it's a way to showcase them to the reader and tease the story. Also, think of ways to *engage* the reader in the thread: A Twitter poll or pose a question to the reader at the end of the thread.

For your first tweet, you should always include a link to your story/project and set up the thread as an explanatory/how-the-sausage-is-made process or the lead to your narrative.

These examples from newsrooms show various approaches for writing effective Twitter threads for data stories. Study how each is structured and the conversational tone the author took in explaining the process.

Mary Jo Webster: Minneapolis Star Tribune Justice Denied https://twitter.com/maryjowebster/status/1021424211880013827?s=12

She explains how she did data reporting on an investigative project. Explains what tools she used and how she went about editing the data. Good explanatory journalism and transparency. When you tell how the "sausage gets made," it builds trust with your reader (Figure 11.8).

Figure 11.8 Mary Jo Webster's Twitter thread for Justice Denied.

David Eads: The Marshall Project https://twitter.com/eads/status/1341772612377202688?s=27

He explains how the Marshall Project investigated police K-9 bites with this six-tweet thread that includes a link to the story and an explanatory data video recorded on Zoom.

NY Times: Tulsa's Black Wall Street https://twitter.com/singhvianjali/status/1397302024333627392?s=27

Great visuals tied to this "how-to" of what Tulsa's Black Wall Street looked like before it was destroyed in the Tulsa Massacre.

Karen Hao: How We Made Tech Review Graphics https://twitter.com/_karenhao/status/1185647498418892808?s=12

Similar to Webster's approach that explains technique, software and process to the reader. Great transparency.

BBC News Graphics: Theresa May Resignation in Data/Graphics https://twitter.com/bbcnewsgraphics/status/1131896210644754432?s=12

How the data team covered the British prime minister's resignation.

* * *

Read More

Data and Diversity

How can journalists make their data reporting more inclusive for diverse audiences? Read how to do it on our blog, http://dataplusjournalism.com

Tools Used in This Chapter

Google Dataset Search https://datasetsearch.research.google.com/
Google Dataset Search developers page https://developers.google.com/search/docs/advanced/structured-data/dataset
Google Flourish https://flourish.studio/
Datawrapper https://www.datawrapper.de/
Tableizer https://tableizer.journalistopia.com/
Google's Data GIF Maker https://datagifmaker.withgoogle.com/

* * *

Footnotes

Society of Professional Journalists Code of Ethics https://www.spj.org/ethicscode.asp
The Reader on Data Visualization, Ch. 5 https://mschermann.github.io/data_viz_reader/ethics.html#ref-ethical_infographics

Alberto Cairo, Ethical Infographics https://www.dropbox.com/s/pqgmg02yz0pgju4/EthicalInfographics.pdf
Washington Post Uber Investigation GitHub page https://github.com/comp-journalism/2016-03-wapo-uber
The Marshall Project, Tracking Police K-9 Violence Using Data https://vimeo.com/493807936/2375f9b7d0
The Marshall Project, We Investigated How Police Use Dogs as Weapons. Here's How You Can Do It Too https://www.themarshallproject.org/2020/12/23/we-investigated-how-police-use-dogs-as-weapons-here-s-how-you-can-do-it-too
The Washington Post, Follow the Post's Coverage of the Opioid Epidemic https://www.washingtonpost.com/national/2019/07/20/opioid-files/
The Red Line Project, Chicago, Cook County on Forefront of Opioid Crisis http://redlineproject.org/opioids1.php
The Oklahoman, We're Losing Children https://stories.usatodaynetwork.com/unaccountable/
The Oklahoman, Failure to Deliver https://stories.usatodaynetwork.com/failure todeliver/#
American Press Institute, Americans and the News Media: What They Do – and Don't – Understand about Each Other https://www.americanpressinstitute.org/publications/reports/survey-research/americans-and-the-news-media/
Trusting News, Investigative Team's "Trust Nuggets" Inject Transparency into Long Stories https://medium.com/trusting-news/investigative-teams-trust-nuggets-inject-transparency-into-long-stories-2dc449b3add8
Trusting News, Earn Trust through Explaining the Process of Reporting https://medium.com/trusting-news/earn-trust-through-explaining-the-process-of-reporting-3c0d1eb06397
Vox, Kara Swisher's Recode Bio https://www.vox.com/authors/kara-swisher
Answering Your Questions about How the Seattle Times Does Investigative Journalism https://www.seattletimes.com/seattle-news/times-watchdog/faq-how-the-seattle-times-does-investigative-journalism/
ProPublica GitHub page https://github.com/propublica
ProPublica News App and Data Style Guides https://github.com/propublica/guides
The Guardian GitHub page https://github.com/guardian
Vox GitHub page https://github.com/voxmedia
BBC GitHub page https://github.com/bbc/
Buzzfeed News GitHub page https://github.com/BuzzFeedNews/everything#guides
New York Times GitHub page https://github.com/nytimes
Datawrapper, Nigerian Debt Pie Chart https://www.datawrapper.de/_/3q5YO/
USA Today, College Football Coaches Salary Database https://public.flourish.studio/visualisation/6431056/

12 Math for Journalists: Writing with Numbers

Mike Reilley

A little humor can go a long way in data journalism. A long-running joke among college journalism professors is that many college students flee the math, science and engineering departments for journalism schools because they think they can escape having to learn math.

How wrong they are.

Math calculations are vital to nearly any newsroom beat, whether you're covering city hall, working in sports or the business departments. While spreadsheets often handle the heavy lifting with formulas, journalists still must master some basic math skills to calculate specific data points and fact-check data from other sources, such as a city budget.

WTTW political reporter Heather Cherone's story about Alderman Austin's committee spending, detailed in Chapter 7, was based on a comprehensive 236-page report that would be overwhelming for many journalists. Cherone spent hours poring over the data, isolating the key numbers for the story. Then she stepped away from the story for a few hours to clear her mind, returned and double-checked all the percentage calculations.

Taking that break gave Cherone a chance to look at the data with a fresh perspective before recalculating "because the last thing you want to do is get something wrong and mess up a percentage," she said. "It's mostly for peace of mind because you don't want to worry."

Without reliving a few years of middle-school math, here are some ways we can make calculations easier for reporters, especially on deadlines.

Figuring Percentage Change

Calculating percentage change can be painful but only if you calculate it as a fraction. Fractions aren't fun, as we know from eighth-grade algebra, so it's better to think of a percentage change as a decimal.

For instance, let's say a company's $135,000 marketing budget increases by 5 percent from one year to the next. To figure the new total, simply multiply $135,000 by 1.05 (1.00 plus 0.05). Use a decimal instead of doing subtraction and fractions.

$135,000 × 1.05 = $141,750 for the new budget

DOI: 10.4324/9781003273301-13

How you could write it as a sentence: "The company's marketing budget increased by 5 percent this year to $141,750."

You apply the same approach if the budget has decreased. Let's say a company's marketing budget decreases by 8 percent; multiply it times 0.92 (1.00 minus 0.08) to get the new budget figure.

$227,000 × 0.92 = $208,840 for the new budget

How you could write it as a sentence: "The company's marketing budget decreased by 8 percent this year to $208,840."

What if you have the new and the old values and need to figure out the amount of the percentage change itself? Take the new value and subtract the old one, and then divide by the old one.

University of Missouri professor David Herzog abbreviates the calculation as "(N-O)/O." Since reporters find themselves doing this often, he recommends a phrase to remember it: "Do journalists like doing math? NOO!"

Percent Change and Percentage Points

An increase from 12 percent to 15 percent isn't a 3 percent increase. It's a three percentage point increase. This is important for budget stories as well as election data. Be very careful with percent and percentage points.

Percentage Increases Exceeding 100 Percent

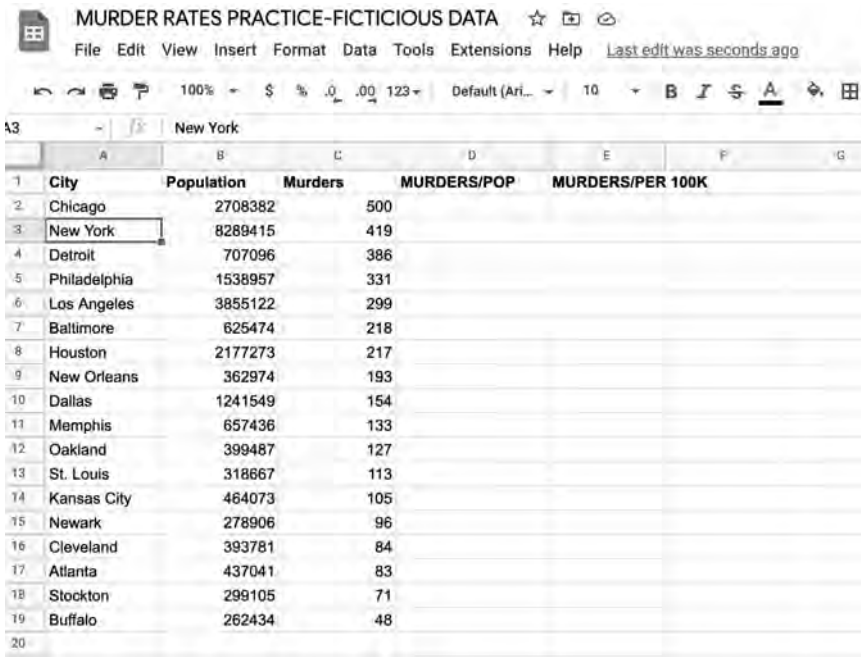
Reporters often make the mistake of miscalculating percentages over 100 percent. There's an easy way to keep it straight. Look what's increasing or decreasing and by how much, not at the percentages.

For example, let's say you're reading a city budget summary, and one of the line items said it increased 300 percent, from $150,000 to $450,000. You check the budget spreadsheet and see that the monetary figures are correct. But is it really a 300 percent increase? It's not. The amount has tripled ($150,000 × 3 = $450,000), but tripling is a 200 percent increase, not 300 percent increase. So it's simpler to tell the readers that the budget has tripled, as stating it as a 200 percent increase can be confusing.

Per Capita and Rates

We covered compiling rates and per capita in the spreadsheet chapters, but it's important to note that percent change in a value tells you only part of the story when you are comparing values for several communities or groups. Another important statistic is each group's *per capita* value. This figure helps you compare values among groups of different sizes.

Rates help readers better understand complex issues such as homicide data or COVID-19 positive cases or vaccinations. They also give a more accurate picture


MURDER RATES PRACTICE-FICTICIOUS DATA

File Edit View Insert Format Data Tools Extensions Help Last edit was seconds ago

A3 New York

	City	Population	Murders	MURDERS/POP	MURDERS/PER 100K
2	Chicago	2708382	500		
3	New York	8289415	419		
4	Detroit	707096	386		
5	Philadelphia	1538957	331		
6	Los Angeles	3855122	299		
7	Baltimore	625474	218		
8	Houston	2177273	217		
9	New Orleans	362974	193		
10	Dallas	1241549	154		
11	Memphis	657436	133		
12	Oakland	399487	127		
13	St. Louis	318667	113		
14	Kansas City	464073	105		
15	Newark	278906	96		
16	Cleveland	393781	84		
17	Atlanta	437041	83		
18	Stockton	299105	71		
19	Buffalo	262434	48		
20					

Figure 12.1 Murders sorted in descending order.

of an issue by looking beyond just the totals for an issue but rather measuring it against another variable, such as population or number of tests administered, etc.

For example, take this fictional dataset of murders for one year in a sample of cities (Shortlink: http://bit.ly/murderrates). When you sort them by column C, the total number of murders, you see that Chicago has the most (500), followed by New York City (419) and Detroit (386) (Figure 12.1).

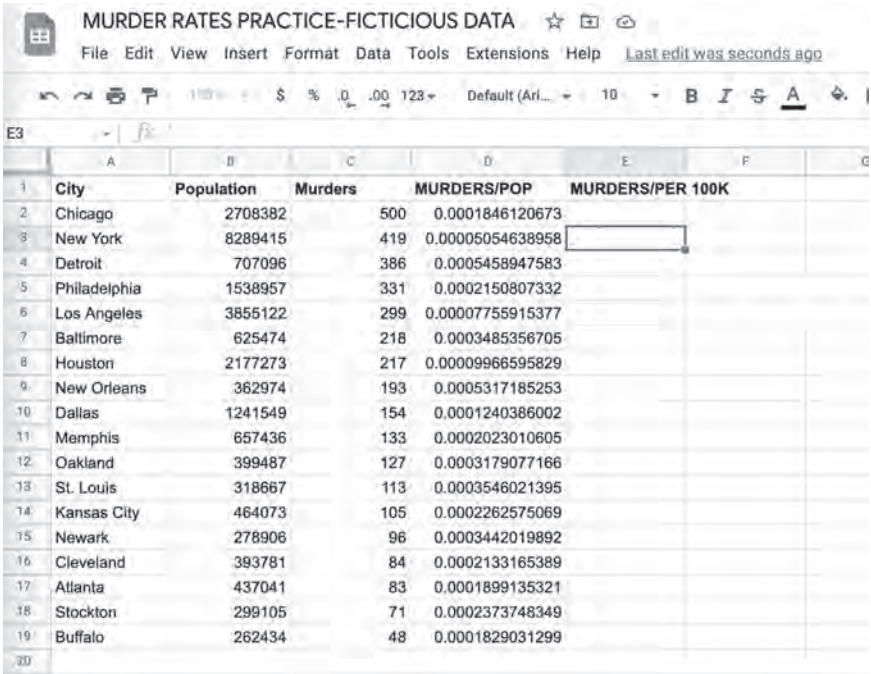
But when you figure population into the equation and sort it, you get a different result.

Start by putting this equation into cell D2 and dragging the handle on the cell down to the bottom to populate the column: =c2/b2 (Figure 12.2).

This compiles murders by population, but the decimal figure would be hard to explain to readers. It becomes easier when we figure the rate per 100,000 residents in cell E2 by typing: =d2*100000 and dragging down to autofill the other columns.

Now highlight the spreadsheet, go to the data menu and select Sort Range/Advanced Sorting Options, check the header row box and sort by descending order (Z to A) on Column E (Murders per 100k).

The next step is to highlight only column E and hit the decrease decimal button to round off the number to one decimal point.



	City	Population	Murders	MURDERS/POP	MURDERS/PER 100K
2	Chicago	2708382	500	0.0001846120673	
3	New York	8289415	419	0.00005054638958	
4	Detroit	707096	386	0.0005458947583	
5	Philadelphia	1538957	331	0.0002150807332	
6	Los Angeles	3855122	299	0.00007755915377	
7	Baltimore	625474	218	0.0003485356705	
8	Houston	2177273	217	0.00009966595829	
9	New Orleans	362974	193	0.0005317185253	
10	Dallas	1241549	154	0.0001240386002	
11	Memphis	657436	133	0.0002023010605	
12	Oakland	399487	127	0.0003179077166	
13	St. Louis	318667	113	0.0003546021395	
14	Kansas City	464073	105	0.0002262575069	
15	Newark	278906	96	0.0003442019892	
16	Cleveland	393781	84	0.0002133165389	
17	Atlanta	437041	83	0.0001899135321	
18	Stockton	299105	71	0.0002373748349	
19	Buffalo	262434	48	0.0001829031299	
20					

Figure 12.2 Figuring murders by population in Google Sheets

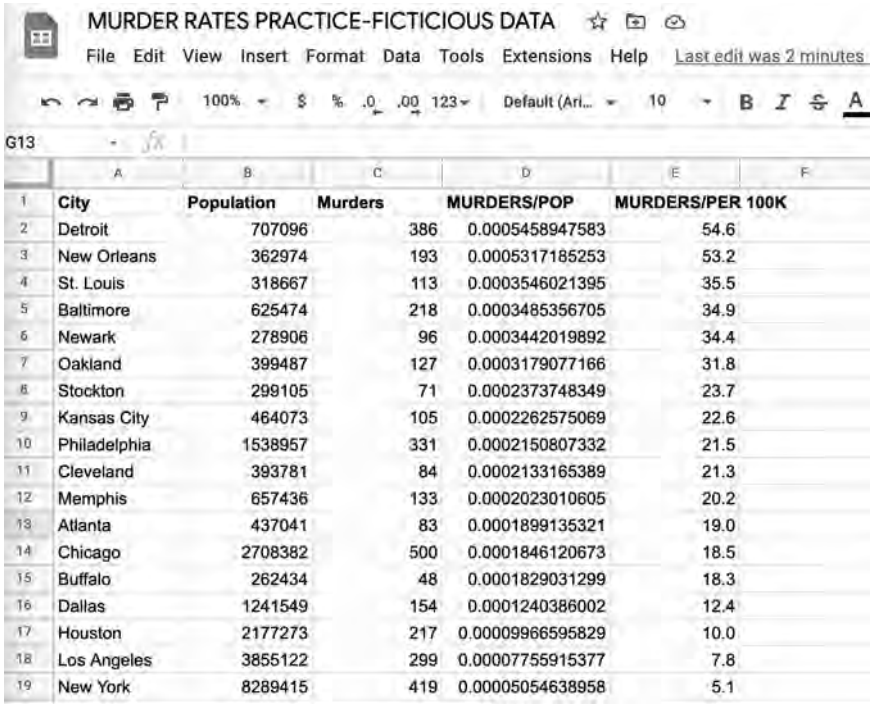
Now your data shows a different result: The cities that have the highest murder rate per 100,000 residents (Figure 12.3).

Now you can write a much more complete story about the murder rates in these cities. Detroit, New Orleans and St. Louis have the highest *murder rates*, while Chicago, New York City and Detroit have the highest *total number* of murders.

Percentage Increase/Decrease Tables

Use this guide as reference for figuring increases and decreases:

- 50 percent increase = increased by half
- 100 percent increase = better to say doubled
- 200 percent increase = better to say tripled
- 300 percent increase = better to say quadrupled


MURDER RATES PRACTICE-FICTICIOUS DATA

File Edit View Insert Format Data Tools Extensions Help Last edit was 2 minutes

100% $ % .0 .00 123 Default (Ari... 10 B I S A

G13

	A	B	C	D	E	F
1	City	Population	Murders	MURDERS/POP	MURDERS/PER 100K	
2	Detroit	707096	386	0.0005458947583	54.6	
3	New Orleans	362974	193	0.0005317185253	53.2	
4	St. Louis	318667	113	0.0003546021395	35.5	
5	Baltimore	625474	218	0.0003485356705	34.9	
6	Newark	278906	96	0.0003442019892	34.4	
7	Oakland	399487	127	0.0003179077166	31.8	
8	Stockton	299105	71	0.0002373748349	23.7	
9	Kansas City	464073	105	0.0002262575069	22.6	
10	Philadelphia	1538957	331	0.0002150807332	21.5	
11	Cleveland	393781	84	0.0002133165389	21.3	
12	Memphis	657436	133	0.0002023010605	20.2	
13	Atlanta	437041	83	0.0001899135321	19.0	
14	Chicago	2708382	500	0.0001846120673	18.5	
15	Buffalo	262434	48	0.0001829031299	18.3	
16	Dallas	1241549	154	0.0001240386002	12.4	
17	Houston	2177273	217	0.00009966595829	10.0	
18	Los Angeles	3855122	299	0.00007755915377	7.8	
19	New York	8289415	419	0.00005054638958	5.1	

Figure 12.3 Figuring murders per 100,000 residents in Google Sheets

Margin of Error in Polls

One of the most common errors political reporters make is to misrepresent polling data leading up to an election. It's critical to convey the flaws in polling to readers and viewers to avoid misleading who's a frontrunner in a race or in a study.

In his book, *A Journalist's Guide to Public Opinion Polls*, G. Evans Witt points out that the most common mistake by political journalists is misstating the nature of one candidate's margin over the other in a well-conducted poll.

Knowing the sampling error margin in a poll you're reporting on is critical, Witt writes, and the figure should be rounded up to the nearest whole number. In other words, a 2.5 percentage point margin of error should be rounded up to 3 percentage points.

Witt cites three situations by using hypothetical polls with a sampling error margin of plus or minus 4 percent:

1. Fifty-two percent of the public says it would vote for Joe Biden for president, while 37 percent say they would vote for Donald Trump. The remainder are undecided.

If the difference between the candidates' percentages is more than twice the sampling error, it's correct to say one candidate leads the other.

2. What if the margin is narrower?

   Biden 44 percent
   Trump 41 percent
   Undecided 15 percent

   This would be a dead heat, too close to call. It would be inaccurate to say that Biden leads Trump as the margin of error is 4 percent, and there are so many undecided voters. The poll only tells you that they're very close in terms of support.

3. If the poll falls between the two previous cases:

   Biden 47 percent
   Trump 41 percent
   Undecided 12 percent

   The margin is six percentage points, and the margin of error is four percentage points, so it would be safe to say Biden has a small lead or leads by a narrow margin. Still, we should be very careful in providing context for the poll and explain to the readers and viewers what the margin of error is.

Besides margin of error, you should closely scrutinize how poll questions are phrased and the sample size, with a particular focus on demographics. The same is true on survey results. Has it captured a large sample and a diverse one in terms of age, ethnicity, gender, religion, political views, geography and other categorizations? Challenge what you find before using it in a story.

The Off by One Error

Let's say you're rushing to write a 500-word story, and you need to get it in at 5 p.m. It's currently 1 p.m. Watching the clock, you write 100 words per hour, reaching a stopping point right on deadline. But, bad news – you're 100 words short!

You've run into a common mathematical trip-up called the "off by one error." It has wide implications in math, programming and law and is sometimes called "the fence-post problem" based on an easy-to-understand allegory.

If someone is building a fence that is 50 feet long, and they need a fence post every 10 feet, how many fence posts do they need?

It's tempting to say five, but the answer is actually six. The yards are distances, or spans, meaning they have a point at each end. If the builder only bought five posts, they would find the last boards hanging into empty space.

This counting process can be deceptively tricky, especially with time spans. Some measurements of time are items, and others are spans. For example, 2011–2014 is three years, because years are counted as distances. But February 11 to 14 is four days, because days are counted as individual items.

How do you know if something is an item or a span? Try to reduce the equation to just one instance. For example, to build a 10-foot fence, two fence posts are needed. That means the fence posts are items, while the fence itself is a span.

You can think of spans as entities that "touch" items. Ergo, the time span of a week – Sunday to Saturday – "touches" seven items, or seven days.

Items and Spans Quiz

1. Sally worked from Monday to Friday and earned $50,000. How much money did she make per day?
2. Mark worked from 1 p.m. to 5 p.m. and earned $50,000. How much did he earn per hour?
3. You are running the SUM() function on the range B1:B4. How many values will the function count?
4. A recent storm blew down street posts, and the city announced it will install new stop signs from 2nd Avenue to 8th Avenue. How many stop signs does the city need to order?

Answers:

1. Sally made $10,000 per day. Days are items, meaning the dollar amount should be divided by the number of items in the list. Monday through Friday "touches" five days.
2. Mark earned $12,500 per hour. Even though the time span of 1 p.m. to 5 p.m. "touches" four hours, hours are counted as distances.
3. The function will count four values, because the values exist in individual cells, or points.
4. The city needs to buy seven stop signs. Even though the distance between 2nd Avenue and 8th Avenue is six blocks, a stop sign is needed for every block that distance touches, which is seven.

Mean, Median and Mode

Mean (average): It's easy to calculate. Just add up all the values in a set of data and then *divide* that sum by the number of values in the dataset.

Median (middle): When you write about "the average worker" this, or "the average household," you don't want to use the mean to describe those situations. You want a statistic that tells you something about the worker or the household in the middle – the median.

It's easy to compile, just line up the values in your set of data, from largest to smallest. The one in the dead center is your median. In the example below, the median is five:

1
2
5
8

12

If it's an even set of numbers, take the two middle numbers, add them and divide by two to determine the median:

1
2
5
6 6 + 5 = 11 11/2 is 5.5
8
12

Mode: The mode is the value in a set that occurs the most often. It's not used often in news stories. But one instance would be showing the most popular example from a list of product prices.

$385 $227 $227 $666 $922
$385 $227 $666 $999 $1,090
$385 $385 $1,090 $999 $666
$227 $667 $385 $921 $1090

This is a bit of a brain tease, but if you look at it long enough, you'll see the mode is $385, as it appears five times in the list. For large lists, it's best to use a spreadsheet formula, which is detailed later in this chapter.

To better understand how mean, median and mode work, let's apply the median, mean and mode to consumer retail purchases. For instance, you're analyzing a local electronics store's purchases for the holiday season. Here's an approach you could take:

- Use the median to describe how much money the typical customer spent at the electronics store. Median also can be used with Census data to best describe household income.

  It's commonly used in real estate to analyze home values in a particular area because it eliminates outliers.
- Use the mean to describe how much money the electronics store collected per customer this season. This figure typically skews larger than the median as it includes outliers, such as customers buying a home computer and others purchasing a set of earbuds.
- Use the mode to describe what was the most common/popular single item or price at the store. For example, an iPad price might be the most popular item sold around the holidays and will appear most often in the store's books.

Basic Google Sheets and Excel Formulas

Why use a calculator when you can let the spreadsheet do the heavy lifting for you?

Figuring Sums

This is the basic formula (Table 12.1). You can adjust cell numbers based on columns/rows/needs: =SUM(B4:B13)

For short lists: =SUM(B4,B5,B6,B7); =SUM(B4+B5+B6+B7). But this formula works best for longer lists: =sum(B4:B700) to add the cells as a larger group.

Or, place your cursor in the first empty cell at the bottom of your list (or any cell, really) and press the plus sign, then click B4, press the plus sign again and click B5 and so on to the end, and then press Enter. Excel adds/totals this list you just "pointed to:" =+B4+B5+B6+B7.

Average (Mean)

=AVERAGE(B4:B13)

Adds the list, divides by the number of values, then provides the average.

Median

=MEDIAN(A2:A6)

Median of the 5 numbers in the range A2:A6. Because there are 5 values, the third is the median.

=MEDIAN(A2:A7)

Median of the 6 numbers in the range A2:A7. Because there are six numbers, the median is the midway point between the third and fourth numbers.

Mode

=MODE(A2:A40)

Shows you the number that occurs most often.

Maximum/Minimum Number in a List

=MAX(B4:B13) returns the highest value in the list.
=MIN(B4:B13) returns the lowest value in the list.

Convert a Fraction into a Decimal

Divide the top number by the bottom number: 5/8 = 0.625, 17/64 = 0.265

Convert a Decimal into a Percentage

Multiply by 100 (or simply move the decimal two places to the RIGHT): 0.421 = 42.1%

Table 12.1 Basic Calculations

– (Minus key)	– (minus)	Use in a formula to subtract numbers or to signify a negative number. Example: =18–12 Example: =24*–5 (24 times negative 5)
× (Multiply key)	* (asterisk; also called "star")	Use in a formula to multiply numbers. Example: =8*3
÷ (Divide key)	/ (forward slash)	Use in a formula to divide one number by another. Example: =45/5

Turn a Percentage into a Decimal

Divide by 100 (or simply move the decimal two places to the LEFT): 122.4% = 1.224

AP Style and Math

Most newsrooms follow *The Associated Press Stylebook* for writing with numbers. The AP's general rule: Spell out any number under 10, and use the numeral for 10 and up. But the AP has many exceptions to its own rule when working with various types of data. Here are a few of those exceptions to focus on:

Temperatures: 7 degrees … minus–5 degrees
Ages: The 9-year-old boy…
Dimensions: 1-foot by 3-foot wide tiles.
Height: 5-foot 10-inch guard … 5-foot-10, 5–10
Weight: 227 pounds, 9 pounds
Speed: 5 mph, not five and not m.p.h.
Time: 2:30 p.m., 9 a.m., not 9:00 AM or PM
Rank: He was my No. 1 choice or first choice. Never use the pound sign # for a ranking. (It can easily be confused for a hashtag.)
Sports: The Bulls beat the Pacers 100–93. Michael Jordan scored 32 points and had 6 assists. Jordan, a 6-foot-6 guard out of North Carolina, has led the Bulls in scoring in seven of 10 games this season.
Monetary figures: Millions and billions are spelled out: $1 million, $1 billion, but not thousands. $1,387,222 is $1.39 million or $1.4 million, depending on how far you need to round up or down. $30,000, not 30,000 dollars. Cents is spelled out: 10 cents, not $0.10.

Estimating Crowd Sizes

You can use MapChecking.com to estimate a crowd size from a past protest or event in your area. Use archived photos as reference to paint the perimeter and estimate the density. Does the attendance estimate you get differ from the once in the coverage? How big is the difference?

Should MapChecking not be available, you can use Google Earth's measure tool to estimate the square yardage or meters of an area and then multiply it by the density. This takes a bit more math work than MapChecking, but it's still a quick and easy way to get an estimate.

You also can estimate a crowd size using a formula shared by Arizona State professor and Pulitzer Prize–winning data journalist Steve Doig:

1. Calculate crowd area in square feet (length × width).
2. Divide by 10 for a loose crowd (people are at arm's length from one another).
3. Divide by 7.5 for a tight crowd (people are more shoulder to shoulder).

> For more tips on how to estimate crowd sizes, visit the Data + Journalism blog at http://dataplusjournalism.com

Math Quiz

1. Springfield's city budget was $27.8 million in 2022. In 2023 it's $33.7 million. What percent change is it? How would you write it in a sentence for the story?
2. Edit this sentence for accuracy, assuming you know the dollar figures are accurate:

   The budget increased 300 percent from 4 billion to 12 billion dollars.
3. Subtract these two percentages: 50–27 percent. What is the answer?
4. The $972,758 budget increased by 8 percent to what dollar figure?
5. Edit this sentence for AP Style:

   From 8:00 am to 1:30 pm, the temperature dropped from 8 to 12 below zero degrees. This was the largest decrease in temperature in the last two and one-half years.
6. You are editing this story and you read the lead and a few paragraphs of the piece. You notice something is amiss. How do you handle this situation?

The United Way on Monday awarded $23,000 in supplemental funds to three organizations.

Later in the story, you spot this sentence: "The supplemental funds will go to the Boy Scouts, $11,000; the Girl Scouts, $9,000; and the Salvation Army, $4,000."

* * *

Math Quiz Answers:

1. $5.9 million increase divided by the original number ($27.8 million) gives you 21.2 percent.

"Springfield's 2023 city budget increased 21.2 percent from last year to a total of $33.7 million." You also could include the 2022 dollar amount as well.

2. "The budget tripled from $4 billion to $12 billion."

   Three hundred percent is quadrupling. If we know the $4 billion and $12 billion are correct, then it's tripling. It's better to use triple than a 200 percent increase. We also fixed the AP Style issue on dollars.
3. The difference is 23 percentage points. Remember, when subtracting percentages, we use percentage points.
4. $972,758 × 1.08 = $1,050,578.64. Or round off to $1.05 million or round up to $1.1 million. The former is more precise than the latter.
5. From 8 a.m. to 1:30 p.m., the temperature dropped from 8 degrees to minus −12 degrees. This was the largest decrease in temperature in the last 2½ years. (Note: Some editors may accept 2.5 years.)
6. The numbers in the second sentence don't add up to $23,000. But which number(s) is wrong? The best thing to do would be to contact the reporter to see if there's a typo or reach out to a United Way official to confirm. Don't assume the dollar figure in the lead is wrong. It could be one of the totals in the second sentence.

* * *

Resources: Math for Journalists

Journalist's Toolbox Math for Journalists https://www.journaliststoolbox.org/category/writing-with-numbers/ Tip sheets, calculators and more

The Journalist's Resource: Statistics for Journalists https://journalistsresource.org/home/statistics-for-journalists/

The Guardian Data Blog https://www.theguardian.com/data

Venngage: Telling Stories with Data https://venngage.com/blog/data-storytelling/

MapChecking Crowd Estimate Tool https://www.mapchecking.com/

Crowd Safety and Crowd Risk Analysis https://www.gkstill.com/Support/crowd-density/625sm/Density6.html Visuals from Dr. Keith Still to measure crowd density.

The Guardian: Trump Inauguration Crowd: Sean Spicer's Claim vs. the Evidence https://www.theguardian.com/us-news/2017/jan/22/trump-inauguration-crowd-sean-spicers-claims-versus-the-evidence

Google Earth https://earth.google.com/

* * *

Footnotes

WTTW, City Council Committee Led by Indicted Ald. Austin Spends More, Does Less than Nearly All Others https://news.wttw.com/2021/08/16/city-council-committee-led-indicted-ald-austin-spends-more-does-less-nearly-all-others

Murder Rates Practice Exercise, Google Sheet https://docs.google.com/spreadsheets/d/1WxUkktVBCrVmt6pPGDxU0Wj1heDTZ2PZ_OpwEwdut5c/edit?usp=sharing

BusinessJournalism.org, Newsroom Math Crib Sheet https://businessjournalism.org/2017/09/newsroom-math-crib-sheet/

Murder Rates Shortlink http://bit.ly/murderrates

* * *

To read the book's addendum, visit the Data + Journalism blog at http://dataplusjournalism.com

Index

Note: **Bold** page numbers refer to tables and *italic* page numbers refer to figures.