Introduction

Data is nothing new.

People have been quantifying and tabulating things for centuries.

However, while writing for FlowingData, my website on design, visualization, and statistics, I’ve seen a huge boom in just these past few years, and it keeps getting better.

Improvements in technology have made it extremely easy to collect and store data, and the web lets you access it whenever you want.

This wealth in data can, in the right hands, provide a wealth of information to help improve decision making, communicate ideas more clearly, and provide a more objective window looking in at how you look at the world and yourself.

A significant shift in release of government data came in mid-2009, with the United States’ launch of Data.gov.

It’s a comprehensive catalog of data provided by federal agencies and represents transparency and accountability of groups and officials.

The thought here is that you should know how the government spends tax dollars.

Whereas before, the government felt more like a black box.

A lot of the data on Data.gov was already available on agency sites scattered across the web, but now a lot of it is all in one place and better formatted for analysis and visualization.

The United Nations has something similar with UNdata; the United Kingdom launched Data.gov.uk soon after, and cities around the world such as New York, San Francisco, and London have also taken part in big releases of data.

The collective web has also grown to be more open with thousands of Application Programming Interfaces (API) to encourage and entice developers to do something with all the available data.

Applications such as Twitter and Flickr provide comprehensive APIs that enable completely different user interfaces from the actual sites.

API-cataloging site ProgrammableWeb reports more than 2,000 APIs.

New applications, such as Infochimps and Factual, also launched fairly recently and were specifically developed to provide structured data.

At the individual level, you can update friends on Facebook, share your location on Foursquare, or tweet what you’re doing on Twitter, all with a few clicks on a mouse or taps on a keyboard.

More specialized applications enable you to log what you eat, how much you weigh, your mood, and plenty of other things.

If you want to track something about yourself, there is probably an application to help you do it.

With all this data sitting around in stores, warehouses, and databases, the field is ripe for people to make sense of it.

The data itself isn’t all that interesting (to most people).

It’s the information that comes out of the data.

People want to know what their data says, and if you can help them, you’re going to be in high demand.

There’s a reason that Hal Varian, Google’s chief economist, says that statistician is the sexy job of the next 10 years, and it’s not just because statisticians are beautiful people.

(Although we are quite nice to look at in that geek chic sort of way.)

Visualization

One of the best ways to explore and try to understand a large dataset is with visualizatio numbers into a visual space and let your brain or your readers’ brains find the patterns.

W that.

You can often find stories that you might never have found with just formal statistica John Tukey, my favorite statistician and the father of exploratory data analysis, was w statistical methods and properties but believed that graphical techniques also had a plac strong believer in discovering the unexpected through pictures.

You can find out a lot ab by visualizing it, and a lot of the time this is all you need to make an informed decision story.

For example, in 2009, the United States experienced a significant increase in its un rate.

In 2007, the national average was 4.6 percent.

In 2008, it had risen to 5.8 percent.

By 2009, however, it was 9.8 percent.

These national averages tell only part of the story generalizing over an entire country.

Were there any regions that had higher unemployment rates than others?

Were there any regions that seemed to be unaffected?

The maps in Figure I-1 tell a more complete story, and you can answer the preceding questions after a glance.

Darker-colored counties are areas that had relatively higher unemployment rates, whereas the lighter-colored counties had relatively lower rates.

In 2009, you see a lot of regions with rates greater than 10 percent in the west and most areas in the east.

Areas in the Midwest were not hit as hard (Figure I-2).

Figure I-1: Maps of unemployment in the United States from 2004 to 2009

Figure I-2: Map of unemployment for 2009

You couldn’t find these geographic and temporal patterns so quickly with just a spreadsheet, and definitely not with just the national averages.

Also, although the county-level data is more complex, most people can still interpret the maps.

These maps could in turn help policy makers decide where to allocate relief funds or other types of support.

The great thing about this is that the data used to produce these maps is all free and publicly available from the Bureau of Labor Statistics.

Albeit the data was not incredibly easy to find from an outdated data browser, but the numbers are there at your disposal, and there is a lot sitting around waiting for some visual treatment.

The Statistical Abstract of the United States, for instance, exists as hundreds of tables of data (Figure I-3), but no graphs.

That’s an opportunity to provide a comprehensive picture of a country.

Really interesting stuff.

I graphed some of the tables a while back as a proof of concept, as shown in Figure I-4, and you get marriage and divorce rates, postal rates, electricity usage, and a few others.

The former is hard to read and you don’t get anything out of it other than individual values.

In the graphical view, you can find trends and patterns easily and make comparisons at a glance.

News outlets, such as The New York Times and The Washington Post do a great job at making data more accessible and visual.

They have probably made the best use of this available data, as related stories have come and passed.

Sometimes data graphics are used to enhance a story with a different point of view, whereas other times the graphics tell the entire story.

Graphics have become even more prevalent with the shift to online media.

There are now departments within news organizations that deal only with interactives or only graphics or only maps.

The New York Times, for example, even has a news desk specifically dedicated to what it calls computer-assisted reporting.

These are reporters who focus on telling the news with numbers.

The New York Times graphics desk is also comfortable dealing with large amounts of data.

Visualization has also found its way into pop culture.

Stamen Design, a visualization firm well known for its online interactives, has provided a Twitter tracker for the MTV Video Music Awards the past few years.

Each year Stamen designs something different, but at its core, it shows what people are talking about on Twitter in real-time.

When Kanye West had his little outburst during Taylor Swift’s acceptance speech in 2009, it was obvious what people thought of him via the tracker.

Figure I-3: Table from the Statistical Abstract of the United States

Figure I-4: A graphical view of data from the Statistical Abstract of the United States

At this point, you enter a realm of visualization less analytical and more about feeling.

The definition of visualization starts to get kind of fuzzy.

For a long time, visualization was about quantitative facts.

You should recognize patterns with your tools, and they should aid your analysis in some way.

Visualization isn’t just about getting the cold hard facts.

Like in the case of Stamen’s tracker, it’s almost more about the entertainment factor.

It’s a way for viewers to watch the awards show and interact with others in the process.

Jonathan Harris’ work is another great example.

Harris designs his work, such as We Feel Fine and Whale Hunt, around stories rather than analytical insight, and those stories revolve around human emotion over the numbers and analytics.

Charts and graphs have also evolved into not just tools but also as vehicles to communicate ideas—and even tell jokes.

Sites such as GraphJam and Indexed use Venn diagrams, pie charts, and the like to represent pop songs or show that a combination of red, black, and white equals a Communist newspaper or a panda murder.

Data Underload, a data comic of sorts that I post on FlowingData, is my own take on the genre.

I take everyday observations and put it in chart form.

The chart in Figure I-5 shows famous movie quotes listed by the American Film Institute.

It’s totally ridiculous but amusing (to me, at least).

Find more Data Underload on FlowingData at <http://datafl.ws/underload>

So what is visualization? Well, it depends on who you talk to.

Some people say it’s strictly traditional graphs and charts.

Others have a more liberal view where anything that displays data is visualization, whether it is data art or a spreadsheet in Microsoft Excel.

I tend to sway more toward the latter, but sometimes find myself in the former group, too.

In the end, it doesn’t actually matter all that much.

Just make something that works for your purpose.

Whatever you decide visualization is, whether you’re making charts for your presentation, analyzing a large dataset, or reporting the news with data, you’re ultimately looking for truth.

At some point in time, lies and statistics became almost synonymous, but it’s not that the numbers lie.

It’s the people who use the numbers who lie.

Sometimes it’s on purpose to serve an agenda, but most of the time it’s inadvertent.

When you don’t know how to create a graph properly or communicate with data in an unbiased way, false junk is likely to sprout.

However, if you learn proper visualization techniques and how to work with data, you can state your points confidently and feel good about your findings.

Figure I-5: Movie quotes in graph form

Learning Data

I got my start in statistics during my freshman year in college.

It was a required introductory course toward my unrelated electrical engineering degree.

Unlike some of the horror stories I’ve heard, my professor was extremely enthusiastic about his teaching and clearly enjoyed the topic.

He quickly walked up and down the stairs of the lecture hall as he taught.

He waved his hands wildly as he spoke and got students involved as he walked by.

To this day, I don’t think I’ve ever had such an excited teacher or professor, and it’s undoubtedly something that drew me into the area of data and eventually what led to graduate school in statistics four years later.

Through all my undergraduate studies, statistics was data analysis, distributions, and hypothesis testing, and I enjoyed it.

It was fun looking at a dataset and finding trends, patterns, and correlations.

When I started graduate school though, my views changed, and things got even more interesting.

Statistics wasn’t just about hypothesis testing (which turns out isn’t all that useful in a lot of cases) and pattern-finding anymore.

Well, no, I take that back.

Statistics was still about those things, but there was a different feel to it.

Statistics is about storytelling with data.

You get a bunch of data, which represents the physical world, and then you analyze that data to find not just correlations, but also what’s going on around you.

These stories can then help you solve real-world problems, such as decreasing crime, improving healthcare, and moving traffic on the freeway, or it can simply help you stay more informed.

A lot of people don’t make that connection between data and real life.

I think that’s why so many people tell me they “hated that course in college” when I tell them I’m in graduate school for statistics.

I know you won’t make that same mistake though, right? I mean, you’re reading this book after all.

How do you learn the necessary skills to make use of data?

You can get it through courses like I did, but you can also learn on your own through experience.

That’s what you do during a large portion of graduate school anyway.

It’s the same way with visualization and information graphics.

You don’t have to be a graphic designer to make great graphics.

You don’t need a statistics PhD either.

You just need to be eager to learn, and like almost everything in life, you have to practice to get better.

I think the first data graphics I made were in the fourth grade.

They were for my science fair project.

My project partner and I pondered (very deeply I am sure) what surface snails move on the fastest.

We put snails on rough and smooth surfaces and timed them to see how long it took them to go a specific distance.

So the data at hand was the times for different surfaces, and I made a bar graph.

I can’t remember if I had the insight to sort from least to greatest, but I do remember struggling with Excel.

The next year though when we studied what cereal red flour beetles preferred, the graphs were a snap.

After you learn the basic functionality and your way around the software, the rest is quite easy to pick up.

If that isn’t a great example of learning from experience, then I don’t know what is.

Oh, and by the way, the snails moved fastest on glass, and the red flour beetles preferred Grape Nuts, in case you were wondering.

This is basic stuff we’re talking about here, but it’s essentially the same process with any software

or programming language you learn. If you’ve never written a line of code, R, many statisticians’

computing environment of choice, can seem intimidating, but after you work through some examples,

you start to quickly get the hang of things. This book can help you with that.

I say this because that’s how I learned. I remember when I first got into more of the design aspects

of visualization. It was the summer after my second year in graduate school, and I had just gotten the

great news that I was going to be a graphics editor intern at The New York Times. Up until then,

graphics had always been a tool for analysis (with the occasional science fair bar graph) to me, and

aesthetics and design didn’t matter so much, if at all. Data and its role in journalism didn’t occur to

me.

So to prepare, I read all the design books I could and went through a guide on Adobe Illustrator

because I knew that’s what The New York Times used. It wasn’t until I actually started making

graphics though when I truly started learning. When you learn by doing, you’re forced to pick up

what is necessary, and your skills evolve as you deal with more data and design more graphics.

How to Read This Book

This book is example-driven and written to give you the skills to take a graphic from start to finish.

You can read it cover to cover, or you can pick your spots if you already have a dataset or visualization in mind.

The chapters are organized so that the examples are self-contained.

If you’re new to data, the early chapters should be especially useful to you.

They cover how to approach your data, what you should look for, and the tools available to you.

You can see where to find data and how to format and prepare it for visualization.

After that, the visualization techniques are split by data type and what type of story you’re looking for.

Remember, always let the data do the talking.

Whatever way you decide to read this book, I highly recommend reading the book with a computer in front of you, so that you can work through examples step-by-step and check out sources referred to in notes and references.

You can also download code and data files and interact with working demos at <http://www.wiley.com/visualizethis> and <http://book.flowingdata.com>.

Just to make things completely clear, here’s a flowchart in Figure I-6 to help you figure what spots to pick.

Have fun! Figure I-6: Where to start reading this book

# Telling Stories with Data

Think of all the popular data visualization works out there—the ones that you always hear in lectures or read about in blogs, and the ones that popped into your head as you were reading this sentence.

What do they all have in common? They all tell an interesting story.

Maybe the story was to convince you of something.

Maybe it was to compel you to action, enlighten you with new information, or force you to question your own preconceived notions of reality.

Whatever it is, the best data visualization, big or small, for art or a slide presentation, helps you see what the data have to say.

# Chapter 2: Handling Data

Before you start working on the visual part of any visualization, you actually need data.

The data is what makes a visualization interesting.

If you don’t have interesting data, you just end up with a forgettable graph or a pretty but useless picture.

Where can you find good data?

How can you access it?

When you have your data, it needs to be formatted so that you can load it into your software.

Maybe you got the data as a comma-delimited text file or an Excel spreadsheet, and you need to convert it to something such as XML, or vice versa.

Maybe the data you want is accessible point-by-point from a web application, but you want an entire spreadsheet.

Learn to access and process data, and your visualization skills will follow.

## Gather Data

Data is the core of any visualization.

Fortunately, there are a lot of places to find it.

You can get it from experts in the area you’re interested in, a variety of online applications, or you can gather it yourself.

Provided by Others

This route is common, especially if you’re a freelance designer or work in a graphics department of a larger organization.

This is a good thing a lot of the time because someone else did all the data gathering work for you, but you still need to be careful.

A lot of mistakes can happen along the way before that nicely formatted spreadsheet gets into your hands.

When you share data with spreadsheets, the most common mistake to look for is typos.

Are there any missing zeros?

Did your client or data supplier mean six instead of five?

At some point, data was read from one source and then input into Excel or a different spreadsheet program (unless a delimited text file was imported), so it’s easy for an innocent typo to make its way through the vetting stage and into your hands.

You also need to check for context.

You don’t need to become an expert in the data’s subject matter, but you should know where the original data came from, how it was collected, and what it’s about.

This can help you build a better graphic and tell a more complete story when you design your graphic.

For example, say you’re looking at poll results.

When did the poll take place?

Who conducted the poll?

Who answered?

Obviously, poll results from 1970 are going to take on a different meaning from poll results from the present day.

Finding Sources

If the data isn’t directly sent to you, it’s your job to go out and find it. The bad news is that, well, that’s more work on your shoulders, but the good news is that’s it’s getting easier and easier to find data that’s relevant and machine-readable (as in, you can easily load it into software).

Here’s where you can start your search.

Search Engines

How do you find anything online nowadays? You Google it. This is a no-brainer, but you’d be surprised how many times people email me asking if I know where to find a particular dataset and a quick search provided relevant results.

Personally, I turn to Google and occasionally look to Wolfram|Alpha, the computational search engine.

|  |
| --- |
| See Wolfram|Alpha at <http://wolframalpha.com>.  The search engine can be especially useful if you’re looking for some basic statistics on a topic. |

Direct from the Source

If a direct query for “data” doesn’t provide anything of use, try searching for academics who specialize in the area you’re interested in finding data for.

Sometimes they post data on their personal sites.

If not, scan their papers and studies for possible leads.

You can also try emailing them, but make sure they’ve actually done related studies.

Otherwise, you’ll just be wasting everyone’s time.

You can also spot sources in graphics published by news outlets such as The New York Times.

Usually data sources are included in small print somewhere on the graphic.

If it’s not in the graphic, it should be mentioned in the related article.

This is particularly useful when you see a graphic in the paper or online that uses data you’re interested in exploring.

Search for a site for the source, and the data might be available.

This won’t always work because finding contacts seems to be a little easier when you email saying that you’re a reporter for the so-and-so paper, but it’s worth a shot.

Universities

As a graduate student, I frequently make use of the academic resources available to me, namely the library.

Many libraries have amped up their technology resources and actually have some expansive data archives.

A number of statistics departments also keep a list of data files, many of which are publicly accessible.

Albeit, many of the datasets made available by these departments are intended for use with course labs and homework.

I suggest visiting the following resources:

Data and Story Library (DASL) (<http://lib.stat.cmu.edu/DASL/>)—An online library of data files and stories that illustrate the use of basic statistics methods, from Carnegie Mellon

Berkeley Data Lab (<http://sunsite3.berkeley.edu/wikis/datalab/>)—Part of the University of California, Berkeley library system

UCLA Statistics Data Sets ([www.stat.ucla.edu/data/](http://www.stat.ucla.edu/data/))—Some of the data that the UCLA Department of Statistics uses in their labs and assignments

5

General Data Applications

A growing number of general data-supplying applications are available.

Some applications provide large data files that you can download for free or for a fee.

Others are built with developers in mind with data accessible via Application Programming Interface (API).

This lets you use data from a service, such as Twitter, and integrate the data with your own application.

Following are a few suggested resources:

Freebase ([www.freebase.com](http://www.freebase.com))—A community effort that mostly provides data on people, places, and things. It’s like Wikipedia for data but more structured. Download data dumps or use it as a backend for your application.

Infochimps (<http://infochimps.org>)—A data marketplace with free and for-sale datasets. You can also access some datasets via their API.

Numbrary (<http://numbrary.com>)—Serves as a catalog for (mostly government) data on the web.

AggData (<http://aggdata.com>)—Another repository of for-sale datasets, mostly focused on comprehensive lists of retail locations.

Amazon Public Data Sets (<http://aws.amazon.com/publicdatasets>)—There’s not a lot of growth here, but it does host some large scientific datasets.

Wikipedia (<http://wikipedia.org>)—A lot of smaller datasets in the form of HTML tables on this community-run encyclopedia.

Topical Data

Outside more general data suppliers, there’s no shortage of subject-specific sites offering loads of free data.

Following is a small taste of what’s available for the topic of your choice.

Geography

Do you have mapping software, but no geographic data? You’re in luck.

Plenty of shapefiles and other geographic file types are at your disposal.

* TIGER ([www.census.gov/geo/www/tiger/](http://www.census.gov/geo/www/tiger/))—From the Census Bureau, probably the most extensive detailed data about roads, railroads, rivers, and ZIP codes you can find
* OpenStreetMap ([www.openstreetmap.org/](http://www.openstreetmap.org/))—One of the best examples of data and community effort
* Geocommons ([www.geocommons.com/](http://www.geocommons.com/))—Both data and a mapmaker
* Flickr Shapefiles ([www.flickr.com/services/api/](http://www.flickr.com/services/api/))—Geographic boundaries as defined by Flickr users

Sports

People love sports statistics, and you can find decades’ worth of sports data.

You can find it on Sports Illustrated or team organizations’ sites, but you can also find more on sites dedicated to the data specifically.

* Basketball Reference ([www.basketball-reference.com/](http://www.basketball-reference.com/))—Provides data as specific as play-by-play for NBA games.
* Baseball DataBank (<http://baseball-databank.org/>)—Super basic site where you can download full datasets.
* databaseFootball ([www.databasefootball.com/](http://www.databasefootball.com/))—Browse data for NFL games by team, player, and season.

World

Several noteworthy international organizations keep data about the world, mainly health and development indicators. It does take some sifting though, because a lot of the datasets are quite sparse. It’s not easy to get standardized data across countries with varied methods.

* Global Health Facts ([www.globalhealthfacts.org/](http://www.globalhealthfacts.org/))—Health-related data about countries in the world.
* UNdata (<http://data.un.org/>)—Aggregator of world data from a variety of sources
* World Health Organization ([www.who.int/research/en/](http://www.who.int/research/en/))—Again, a variety of health-related datasets such as mortality and life expectancy
* OECD Statistics (<http://stats.oecd.org/>)—Major source for economic indicators
* World Bank (<http://data.worldbank.org/>)—Data for hundreds of indicators and developer-friendly

Government and Politics

There has been a fresh emphasis on data and transparency in recent years, so many government organizations supply data, and groups such as the Sunlight Foundation encourage developers and designers to make use of it.

Government organizations have been doing this for awhile, but with the launch of data.gov, much of the data is available in one place.

You can also find plenty of nongovernmental sites that aim to make politicians more accountable.

* Census Bureau ([www.census.gov/](http://www.census.gov/))—Find extensive demographics here.
* Data.gov (<http://data.gov/>)—Catalog for data supplied by government organizations. Still relatively new, but has a lot of sources.
* Data.gov.uk (<http://data.gov.uk/>)—The Data.gov equivalent for the United Kingdom.
* DataSF (<http://datasf.org/>)—Data specific to San Francisco.
* NYC DataMine (<http://nyc.gov/data/>)—Just like the above, but for New York.
* Follow the Money ([www.followthemoney.org/](http://www.followthemoney.org/))—Big set of tools and datasets to investigate money in state politics.
* OpenSecrets ([www.opensecrets.org/](http://www.opensecrets.org/))—Also provides details on government spending and lobbying.

Data Scraping

Often you can find the exact data that you need, except there’s one problem.

It’s not all in one place or in one file.

Instead it’s in a bunch of HTML pages or on multiple websites.

What should you do? The straightforward, but most time-consuming method would be to visit every page and manually enter your data point of interest in a spreadsheet.

If you have only a few pages, sure, no problem.

What if you have a thousand pages? That would take too long—even a hundred pages would be tedious.

It would be much easier if you could automate the process, which is what data scraping is for.

You write some code to visit a bunch of pages automatically, grab some content from that page, and store it in a database or a text file.

|  |
| --- |
| Note  Although coding is the most flexible way to scrape the data you need, you can also try tools such as Needlebase and Able2Extract PDF converter.  Use is straightforward, and they can save you time. |

Example: Scrape a Website

The best way to learn how to scrape data is to jump right into an example.

Say you wanted to download temperature data for the past year, but you can’t find a source that provides all the numbers for the right time frame or the correct city.

Go to almost any weather website, and at the most, you’ll usually see only temperatures for an extended 10-day forecast.

That’s not even close to what you want.

You want actual temperatures from the past, not predictions about future weather.

Fortunately, the Weather Underground site does provide historic temperatures; however, you can see only one day at a time.

To make things more concrete, look up temperature in Buffalo. Go to the Weather Underground site and search for BUF in the search box.

This should take you to the weather page for Buffalo Niagara International, which is the airport in Buffalo (see Figure 2-1).

|  |
| --- |
| Visit Weather Underground at http://wunderground.com. |

Generalizing the Example

Although you just scraped weather data from Weather Underground, you can generaliz with other data sources. Data scraping typically involves three steps:

* + - 1. **Identify the patterns.**
      2. **Iterate.**
      3. **Store the data.**

In this example, you had to find two patterns.

The first was in the URL, and the sec web page to get the actual temperature value.

To load the page for a different day in 2 month and day portions of the URL.

The temperature value was enclosed in the sixth o class in the HTML page.

If there is no obvious pattern to the URL, try to figure out how of all the pages you want to scrape.

Maybe the site has a site map, or maybe you can go t search engine.

In the end, you need to know all the URLs of the pages of data.

After you find the patterns, you iterate.

That is, you visit all the pages programmatically, load them, and parse them.

Here you did it with Beautiful Soup, which makes parsing XML and HTML easy in Python.

There’s probably a similar library if you choose a different programming language.

Lastly, you need to store it somewhere.

The easiest solution is to store the data as a plain text file with comma-delimited values, but if you have a database set up, you can also store the values in there.

Things can get trickier as you run into web pages that use JavaScript to load all their data into view, but the process is still the same.

## Formatting Data

|  |
| --- |
| What I Learned about Formatting  When I first learned statistics in high school, the data was always provided in a nice, rectangular format.  All I had to do was plug some numbers into an Excel spreadsheet or my awesome graphing calculator (which was the best way to look like you were working in class, but actually playing Tetris).  That’s how it was all the way through my undergraduate education.  Because I was learning about techniques and theorems for analyses, my teachers didn’t spend any time on working with raw, preprocessed data.  The data always seemed to be in just the right format.  This is perfectly understandable, given time constraints and such, but in graduate school, I realized that data in the real world never seems to be in the format that you need.  There are missing values, inconsistent labels, typos, and values without any context.  Often the data is spread across several tables, but you need everything in one, joined across a value, like a name or a unique id number.  This was also true when I started to work with visualization.  It became increasingly important because I wanted to do more with the data I had.  Nowadays, it’s not out of the ordinary that I spend just as much time getting data in the format that I need as I do putting the visual part of a data graphic together.  Sometimes I spend more time getting all my data in place.  This might seem strange at first, but you’ll find that the design of your data graphics comes much easier when you have your data neatly organized, just like it was back in that introductory statistics course in high school. |

Data Formats

Different visualization tools use different data formats, and the structure you use varies by the story you want to tell.

So the more flexible you are with the structure of your data, the more possibilities you can gain.

Make use of data formatting applications, and couple that with a little bit of programming know-how, and you can get your data in any format you want to fit your specific needs.

The easy way of course is to find a programmer who can format and parse all of your data, but you’ll always be waiting on someone.

This is especially evident during the early stages of any project where iteration and data exploration are key in designing a useful visualization.

Honestly, if I were in a hiring position, I’d likely just get the person who knows how to work with data, over the one who needs help at the beginning of every project.

Various data formats, the tools available to deal with these formats, and finally, some programming, using the same logic you used to scrape data in the previous example are described next.

Delimited Text

Most people are used to working with data in Excel.

This is fine if you’re going to do everything from analyses to visualization in the program, but if you want to step beyond that, you need to familiarize yourself with other data formats.

The point of these formats is to make your data machine-readable, or in other words, to structure your data in a way that a computer can understand.

Which data format you use can change by visualization tool and purpose, but the three following formats can cover most of your bases: delimited text, JavaScript Object Notation, and Extensible Markup Language.

JavaScript Object Notation (JSON)

This is a common format offered by web APIs.

It’s designed to be both machine- and human-readable; although, if you have a lot of it in front of you, it’ll probably make you cross-eyed if you stare at it too long.

It’s based on JavaScript notation, but it’s not dependent on the language.

There are a lot of specifications for JSON, but you can get by for the most part with just the basics.

JSON works with keywords and values, and treats items like objects.

If you were to convert JSON data to comma-separated values (CSV), each object might be a row.

As you can see later in this book, a number of applications, languages, and libraries accept JSON as input.

If you plan to design data graphics for the web, you’re likely to run into this format.

|  |
| --- |
| Visit http://json.org for the full specification of JSON. You don’t need to know every detail of the format,  but it can be handy at times when you don’t understand a JSON data source. |

Extensible Markup Language (XML)

XML is another popular format on the web, often used to transfer data via APIs.

There are lots of different types and specifications for XML, but at the most basic level, it is a text document with values enclosed by tags.

For example, the Really Simple Syndication (RSS) feed that people use to subscribe to blogs, such as FlowingData, is actually an XML file, as shown in Figure 2-7.

The RSS lists recently published items enclosed in the <item></item> tag, and each item has a title, description, author, and publish date, along with some other attributes.

XML is relatively easy to parse with libraries such as Beautiful Soup in Python.

You can get a better feel for XML, along with CSV and JSON, in the sections that follow.

Formatting Tools

Just a couple of years ago, quick scripts were always written to handle and format data.

After you’ve written a few scripts, you start to notice patterns in the logic, so it’s not super hard to write new scripts for specific datasets, but it does take time.

Luckily, with growing volumes of data, some tools have been developed to handle the boiler plate routines.

Google Refine

Google Refine is the evolution of Freebase Gridworks.

Gridworks was first developed as an in-house tool for an open data platform, Freebase; however, Freebase was acquired by Google, therefore the new name.

Google Refine is essentially Gridworks 2.0 with an easier-to-use interface (Figure 2-8) with more features.

It runs on your desktop (but still through your browser), which is great, because you don’t need to worry about uploading private data to Google’s servers.

All the processing happens on your computer.

Refine is also open source, so if you feel ambitious, you can cater the tool to your own needs with extensions.

When you open Refine, you see a familiar spreadsheet interface with your rows and columns.

You can easily sort by field and search for values.

You can also find inconsistencies in your data and consolidate in a relatively easy way.

For example, say for some reason you have an inventory list for your kitchen.

You can load the data in Refine and quickly find inconsistencies such as typos or differing classifications.

Maybe a fork was misspelled as “frk,” or you want to reclassify all the forks, spoons, and knives as utensils.

You can easily find these things with Refine and make changes.

If you don’t like the changes you made or make a mistake, you can revert to the old dataset with a simple undo.

Getting into the more advanced stuff, you can also incorporate data sources like your own with a dataset from Freebase to create a richer dataset.

If anything, Google Refine is a good tool to keep in your back pocket.

It’s powerful, and it’s a free download, so I highly recommend you at least fiddle around with the tool.

|  |
| --- |
| Download the open-source Google Refine and view tutorials on how to make the most out of the tool at   * <http://code.google.com/p/google-refine/> (old) * <https://github.com/OpenRefine> (new) |

Mr. Data Converter

Often, you might get all your data in Excel but then need to convert it to another format to fit your needs.

This is almost always the case when you create graphics for the web.

You can already export Excel spreadsheets as CSV, but what if you need something other than that? Mr. Data Converter can help you.

Mr. Data Converter is a simple and free tool created by Shan Carter, who is a graphics editor for The New York Times.

Carter spends most of his work time creating interactive graphics for the online version of the paper.

He has to convert data often to fit the software that he uses, so it’s not surprising he made a tool that streamlines the process.

It’s easy to use, and Figure 2-9 shows that the interface is equally as simple.

All you need to do is copy and paste data from Excel in the input section on the top and then select what output format you want in the bottom half of the screen.

Choose from variants of XML, JSON, and a number of others.

Figure 2-9: Mr.

Data Converter makes switching between data formats easy.

The source code to Mr. Data Converter is also available if you want to make your own or extend.

|  |
| --- |
| Try out Mr. Data Converter at <http://www.shancarter.com/data_converter/> (outdated) or download the source on github at <https://github.com/shancarter/Mr-Data-Converter> to convert your Excel spreadsheets to a  web-friendly format.  Note. New loc of website <https://shancarter.github.io/mr-data-converter/> |

Mr. People

Inspired by Carter’s Mr. Data Converter, The New York Times graphics deputy director Matthew Ericson created Mr. People.

Like Mr. Data Converter, Mr. People enables you to copy and paste data into a text field, and the tool parses and extracts for you.

Mr. People, however, as you might guess, is specifically for parsing names.

Maybe you have a long list of names without a specific format, and you want to identify the first and last names, along with middle initial, prefix, and suffix.

Maybe multiple people are listed on a single row.

That’s where Mr. People comes in.

Copy and paste names, as shown in Figure 2-10, and you get a nice clean table that you can copy into your favorite spreadsheet software, as shown in Figure 2-11.

Like Mr. Data Converter, Mr. People is also available as open-source software on github.

|  |
| --- |
| Use Mr. People at <http://people.ericson.net/> or download the Ruby source on github to use the name parser  in your own scripts: <http://github.com/mericson/people>. |

Spreadsheet Software

Of course, if all you need is simple sorting, or you just need to make some small changes to individual data points, your favorite spreadsheet software is always available.

Take this route if you’re okay with manually editing data.

Otherwise, try the preceding first (especially if you have a giganto dataset), or go with a custom coding solution.

Formatting with Code

Although point-and-click software can be useful, sometimes the applications don’t quite do what you want if you work with data long enough.

Some software doesn’t handle large data files well; they get slow or they crash.

What do you do at this point? You can throw your hands in the air and give up; although, that wouldn’t be productive.

Instead, you can write some code to get the job done.

With code you become much more flexible, and you can tailor your scripts specifically for your data.

Now jump right into an example on how to easily switch between data formats with just a few lines of code.

Example: Switch Between Data Formats

This example uses Python, but you can of course use any language you want.

The logic is the same, but the syntax will be different.

(I like to develop applications in Python, so managing raw data with Python fits into my workflow.) Going back to the previous example on scraping data, use the resulting wunder-data.txt file, which has dates and temperatures in Buffalo, New York, for 2009.

The first rows look like this:

This is a CSV file, but say you want the data as XML in the following format:

Put Logic in the Loop

If you look at the code to convert your CSV file to JSON, you should notice the if-else statement in the for loop, after the three print lines.

This checks if the current iteration is the last row of data.

If it isn’t, don’t put a comma at the end of the observation.

Otherwise, you do.

This is part of the JSON specification.

You can do more here.

You can check if the max temperature is more than a certain amount and create a new field that is 1 if a day is more than the threshold, or 0 if it is not.

You can create categories or flag days with missing values.

Actually, it doesn’t have to be just a check for a threshold.

You can calculate a moving average or the difference between the current day and the previous.

There are lots of things you can do within the loop to augment the raw data.

Everything isn’t covered here because you can do anything from trivial changes to advanced analyses, but now look at a simple example.

Going back to your original CSV file, wunder-data.txt, create a third column that indicates whether a day’s maximum temperature was at or below freezing.

A 0 indicates above freezing, and 1 indicates at or below freezing.

Like before, read the data from the CSV file into Python, and then iterate over each row.

Check each day and flag accordingly.

This is of course a simple example, but it should be easy to see how you can expand on this logic to format or augment your data to your liking.

Remember the three steps of load, loop, and process, and expand from there.

## Wrapping Up

This chapter covered where you can find the data you need and how to manage it after you have it.

This is an important step, if not the most important, in the visualization process.

A data graphic is only as interesting as its underlying data.

You can dress up a graphic all you want, but the data (or the results from your analysis of the data) is still the substance; and now that you know where and how to get your data, you’re already a step ahead of the pack.

You also got your first taste of programming.

You scraped data from a website and then formatted and rearranged that data, which will be a useful trick in later chapters.

The main takeaway, however, is the logic in the code.

You used Python, but you easily could have used Ruby, Perl, or PHP.

The logic is the same across languages.

When you learn one programming language (and if you’re a programmer already, you can attest to this), it’s much easier to learn other languages later.

You don’t always have to turn to code.

Sometimes there are click-and-drag applications that make your job a lot easier, and you should take advantage of that when you can.

In the end, the more tools you have in your toolbox, the less likely you’re going to get stuck somewhere in the process.

Okay, you have your data.

Now it’s time to get visual.

# Chapter 3: Choosing Tools to Visualize Data

## Out-of-the-Box Visualization

In the last chapter, you learned where to find your data and how to get it in the format you need, so you’re ready to start visualizing.

One of the most common questions people ask me at this point is “ What software should I use to visualize my data?” Luckily, you have a lot of options.

Some are out-of-the-box and click-and-drag.

Others require a little bit of programming, whereas some tools weren’t designed specifically for data graphics but are useful nevertheless.

This chapter covers these options.

The more visualization tools you know how to use and take advantage of, the less likely you’ll get stuck not knowing what to do with a dataset and the more likely you can make a graphic that matches your vision.

Options

The out-of-the-box solutions are by far the easiest for beginners to pick up.

Copy and paste some data or load a CSV file and you’re set.

Just click the graph type you want—maybe change some options here and there.

Microsoft Excel

You know this one. You have the all-familiar spreadsheet where you put your data, such as in Figure 3-1.

Then you can click the button with the little bar graph on it to make the chart you want.

You get all your standard chart types (Figure 3-2) such as the bar chart, line, pie, and scatterplot.

Some people scoff at Excel, but it’s not all that bad for the right tasks.

For example, I don’t use Excel for any sort of deep analyses or graphics for a publication, but if I get a small dataset in an Excel file, as is often the case, and I want a quick feel for what is in front of me, then sure, I’ll whip up a graph with a few clicks in everyone’s favorite spreadsheet program.

|  |
| --- |
| **Graphs Really Can Be Fun**  The first graph I made on a computer was in Microsoft Excel for my fifth grade science fair project.  My project partner and I tried to find out which surface snails moved on the fastest.  It was ground-breaking research, I assure you.  Even back then I remember enjoying the graph-making.  It took me forever to learn (the computer was still new to me), but when I finally did, it was a nice treat.  I entered numbers in a spreadsheet and then got a graph instantly that I could change to any color I wanted—blinding, bright yellow it is. |

This ease of use is what makes Excel so appealing to the masses, and that’s fine.

If you want higher quality data graphics, don’t stop here.

Other tools are a better fit for that

Google Spreadsheets

Google Spreadsheets is essentially the cloud version of Microsoft Excel with the familiar spreadsheet interface, obviously (Figure 3-3)

Many Eyes

Many Eyes is an ongoing research proje application that enables you to upload yo interactive visualization tools.

The origina large datasets as groups—therefore the na find interesting points in the data quicker or more efficiently or find things in the data that you would not have found on your own? Although social data analyses never caught on with Many Eyes, the tools can still be useful to the individual.

Most traditional visualization types are available, such as the line graph (Figure 3-7) and the scatterplot (Figure 3-8).

One of the great things about all the visualizations on Many Eyes is that they are interactive and provide a number of customization options.

The scatterplot, for example, enables you to scale dots by a third metric, and you can view individual values by rolling over a point of interest.

Many Eyes also provides a variety of more advanced and experimental visualizations, along with some basic mapping tools.

A word tree helps you explore a full body of text, such as in a book or news article.

You choose a word or a phrase, and you can see how your selection is used throughout the text by looking at what follows.

Figure 3-9, for example, shows the results of a search for right in the United States Constitution.

Alternatively, you can easily switch between tools, using the same data.

Figure 3-10 shows the Constitution visualized with a stylized word cloud, known as a Wordle.

Words used more often are sized larger.

As you can see, Many Eyes has a lot of options to help you play with your data and is by far the most-extensive (and in my eyes, the best) free tool for data exploration; however, a couple of caveats exist.

The first is that most of the tools are Java applets, so you can’t do much if you don’t have Java installed.(This isn’t a big deal for most, but I know some people, for whatever reason, who are particular about what they put on their computer.)

The other caveat, which can be a deal breaker, is that all the data you upload to the site is in the public domain.

So you can’t use Many Eyes, for example, to dig into customer information or sales made by your business.

Try uploading and visualizing your own data <http://many-eyes.com>.

Tableau Software

Tableau Software, which is Windows-only software, is relatively new but has been growing in popularity for the past couple of years.

It’s designed mainly to explore and analyze data visually.

It’s clear that careful thought has been given to aesthetics and design, which is why so many people like it.

Tableau Software offers lots of interactive visualization tools and does a good job with data management, too.

You can import data from Excel, text files, and database servers.

Standard time series charts, bar graphs, pie charts, basic mapping, and so on are available.

You can mix and match these displays, hook in a dynamic data source for a custom view, or a dashboard, for a snapshot of what’s going on in your data.

Most recently, Tableau released Tableau Public, which is free and offers a subset of the functionality in the desktop editions.

You can upload your data to Tableau’s servers, build an interactive display, and easily publish it to your website or blog.

Any data you upload to the servers though, like with Many Eyes, does become publicly available, so keep that in mind.

If you want to use Tableau and keep your data private, you need to go with the desktop editions.

At the time of this writing, the desktop software is on the pricier side at $999 and $1,999 for the Personal and Professional editions, respectively.

Visit Tableau Software at <http://tableausoftware.com>. It has a full-functioning free trial.

your.flowingdata

My interest in personal data collection inspired my own application, your.flowingdata (YFD).

It’s an online application that enables you to collect data via Twitter and then explore patterns and relationships with a set of interactive visualization tools.

Some people track their eating habits or when they go to sleep and wake up.

Others have logged the habits of their newborn as sort of a baby scrapbook, with a data twist.

YFD was originally designed with personal data in mind, but many have found the application useful for more general types of data collection, such as web activity or train arrivals and departures.

Try personal data collection via Twitter at <http://your.flowingdata.com>.

Trade-Offs

Although these tools are easy to use, there are some drawbacks.

In exchange for click-and-drag, you give up some flexibility in what you can do.

You can usually change colors, fonts, and titles, but you’re restricted to what the software offers.

If there is no button for the chart you want, you’re out of luck.

On the flip side, some software might have a lot of functions, but in turn have a ton of buttons that you need to learn.

For example, there was one program (not listed here) that I took a weekend crash course for, and it was obvious that it could do a lot if I put in the time.

The processes to get things done though were so counterintuitive that it made me not want to learn anymore.

It was also hard to repeat my work for different datasets, because I had to remember everything I clicked.

In contrast, when you write code to handle your data, it’s often easy to reuse code and plug in a different dataset.

Don’t get me wrong.

I’m not saying to avoid out-of-the-box software completely.

They can help you explore your data quickly and easily.

But as you work with more datasets, there will be times when the software doesn’t fit, and when that time comes you can turn to programming.

## Programming

Options

So you decide to get your hands dirty with code—good for you.

A lot of options are freely available.

Some languages are better at performing certain tasks better than others.

Some solutions can handle large amounts of data, whereas others are not as robust in that department but can produce much better visuals or provide interaction.

Which language you use largely depends on what your goals are for a specific data graphic and what you’re most comfortable with.

Some people stick with one language and get to know it well.

This is fine, and if you’re new to programming, I highly recommend this strategy.

Familiarize yourself with the basics and important concepts of code.

Use the language that best suits your needs.

However, it’s fun to learn new languages and new ways to play with data; so you should develop a good bit of programming experience before you decide on your favorite solution.

Python

PHP

Processing

Flash and ActionScript

HTML, JavaScript, and CSS

R

\* Trade-Offs

## Illustration

Options

Adobe Illustrator

Inkscape

Others

Trade-Offs

## Mapping

Options

Google, Yahoo, and Microsoft Maps

ArcGIS

Modest Maps

Polymaps

R

Online-Based Solutions

Trade-Offs

## Survey Your Options

Combining Them

## Wrapping Up