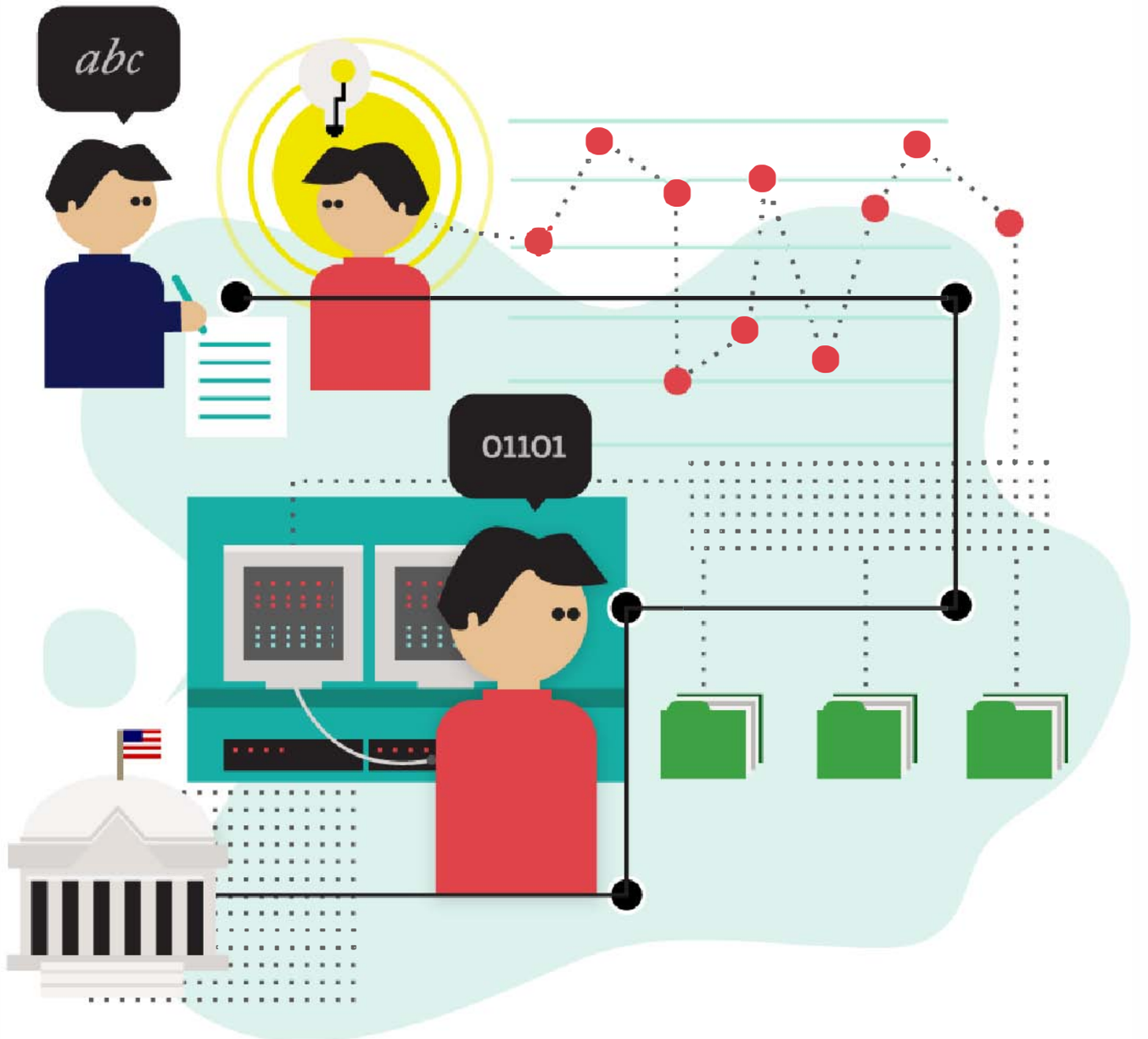
Chapter 5. Understanding Data



Once you've got your data, what do you do with it? What should you look for? What tools should you use? This section opens with some ideas on improving your data literacy, tips for working with numbers and statistics, and things to bear in mind while working with messy, imperfect and often undocumented datasets. We go on to learn about how to get stories from data, data journalists' tools of choice, and how to use data visualization to give you insights into the topic you're looking at.

Become Data Literate in Three Simple Steps

Just as literacy refers to "the ability to read for knowledge, write coherently, and think critically about printed material," data-literacy is the ability to consume for knowledge, produce coherently, and think critically about data. Data literacy includes statistical literacy, but also understanding how to work with large datasets, how they were produced, how to connect various datasets, and how to interpret them.

Figure 5-1. Digging into data (photo by JDHancock, http://www.flickr.com/photos/jdhancock/3386035827/)

Poynter's News University offers math classes for journalists, in which reporters get help with concepts such as percentage changes and averages. Interestingly enough, these concepts are being taught simultaneously near Poynter's offices, in Floridian schools to fifth grade pupils (age 10-11), as the curriculum attests.

That journalists need help in math topics normally covered before high school shows how far newsrooms are from being data-literate. This is a problem. How can a data journalist make use of a bunch of numbers on climate change if she doesn't know what a confidence interval means? How can a data reporter write a story on income distribution if he cannot tell the mean from the median?

A reporter certainly does not need a degree in statistics to become more efficient when dealing with data. When faced with numbers, a few simple tricks can help her get a much better story. As Max Planck Institute professor Gerd Gigerenzer says, better tools will not lead to better journalism if they are not used with insight.

Even if you lack any knowledge of math or stats, you can easily become a seasoned data-journalist by asking 3 very simple questions.

1. How was the data collected?

Amazing GDP growth

The easiest way to show off with spectacular data is to fabricate it. It sounds obvious, but data as commonly commented upon as GDP figures can very well be phony. Former British ambassador Craig Murray reports in his book, *Murder in Samarkand*, that growth rates in Uzbekistan are subject to intense negotiations between the local government and international bodies. In other words, it has nothing to do with the local economy.

GDP is used as the number one indicator because governments need it to watch over their main source of income —VAT. When a government is not funded by VAT, or when it does not make its budget public, it has no reason to collect GDP data and will be better off fabricating them.

Crime is always on the rise

"Crime in Spain grew by 3%," writes El Pais. Brussels is prey to increased crime from illegal aliens and drug addicts, says RTL. This type of reporting based on police-collected statistics is common, but it doesn't tell us much about violence.

We can trust that within the European Union, the data isn't tampered with. But police personnel respond to incentives. When performance is linked to clearance rate, for instance, policemen have an incentive to report as much as possible on incidents that don't require an investigation. One such crime is smoking pot. This explains why drug-related crimes in France increased fourfold in the last 15 years while consumption remained constant.

What you can do

When in doubt about a number's credibility, always double-check, just as you'd have if it had been a quote from a politician. In the Uzbek case, a phone call to someone who's lived there for a while suffices ("Does it feel like the country is 3 times as rich as it was in 1995, as official figures show?").

For police data, sociologists often carry out victimization studies, in which they ask people if they are subject to crime. These studies are much less volatile than police data. Maybe that's the reason why they don't make headlines.

Other tests let you assess precisely the credibility of the data, such as Benford's law, but none will replace your own critical thinking.

2. What's in there to learn?

Risk of Multiple Sclerosis doubles when working at night

Surely any German in her right mind would stop working night shifts after reading this headline. But the article doesn't tell us what the risk really is in the end.

Take 1,000 Germans. A single one will develop MS over his lifetime. Now, if every one of these 1,000 Germans worked night shifts, the number of MS sufferers would jump to 2. The additional risk of developing MS when working in shifts is 1 in 1,000, not 100%. Surely this information is more useful when pondering whether to take the job.

On average, 1 in every 15 Europeans totally illiterate

The above headline looks frightening. It is also absolutely true. Among the 500 million Europeans, 36 million probably don't know how to read. As an aside, 36 million are also under 7 (data from Eurostat; http://bit.ly/eurostat-numeracy).

When writing about an average, always think "an average of what?" Is the reference population homogeneous? Uneven distribution patterns explain why most people drive better than average, for instance. Many people have zero or just one accident over their lifetime. A few reckless drivers have a great many, pushing the average number of accidents way higher than what most people experience. The same is true of the income distribution: most people earn less than average.

What you can do

Always take the distribution and base rate into account. Checking for the mean and median, as well as mode (the most frequent value in the distribution) helps you gain insights in the data. Knowing the order of magnitude makes contextualization easier, as in the MS example. Finally, reporting in natural frequencies (1 in 100) is way easier for readers to understand than using percentage (1%).

3. How reliable is the information?

The sample size problem

"80% dissatisfied with the judicial system", says a survey reported in Zaragoza-based Diaro de Navarra. How can one extrapolate from 800 respondents to 46 million Spaniards? Surely this is full of hot air.

When researching a large population (over a few thousand), you rarely need more than a thousand respondents to achieve a margin of error under 3%. It means that if you were to retake the survey with a totally different sample, 19 times out of 20, the answers you'll get will be within a 3 percentage points interval of the value you would have found, had you asked every single person.

Drinking tea lowers the risk of stroke

Articles about the benefits of tea-drinking are commonplace. This short item in Die Welt saying that tea lowers the risk of myocardial infarction is no exception. Although the effects of tea are seriously studied by some, many pieces of research fail to take into account lifestyle factors, such as diet, occupation, or sports.

In most countries, tea is a beverage for the health-conscious upper classes. If researchers don't control for lifestyle factors in tea studies, they tell us nothing more than "rich people are healthier—and they probably drink tea."

What you can do

The math behind correlations and error margins in the tea studies are certainly correct, at least most of the time. But if researchers don't look for co-correlations (e.g., drinking tea correlates with playing sports), their results are of little value.

As a journalist, it makes little sense to challenge the numerical results of a study, such as the sample size, unless there are serious doubts about it. However, it is easy to see if researchers failed to take relevant pieces of information into account.

— *Nicolas Kayser-Bril, Journalism++*

Tips for Working with Numbers in the News

- The best tip for handling data is to enjoy yourself. Data can appear forbidding. But allow it to intimidate you and you'll get nowhere. Treat it as something to play with and explore and it will often yield secrets and stories with surprising ease. So handle it simply as you'd handle other evidence, without fear or favor. In particular, think of this as an exercise in imagination. Be creative by thinking of the alternative stories that might be consistent with the data and explain it better, then test them against more evidence. "What other story could explain this?" is a handy prompt to think about how this number, this obviously big or bad number, this clear proof of this or that, might be nothing of the sort.
- Don't confuse skepticism about data with cynicism. Skepticism is good; cynicism has simply thrown up its hands and quit. If you believe in data journalism (and you probably do, or you wouldn't be reading this book), then you must believe that data has something far better to offer than the lies and damned lies of caricature or the killer facts of swivel-eyed headlines. Data often gives us profound knowledge, if used carefully. We need to be neither cynical nor naive, but alert.
- If I tell you that drinking has gone up during the recession, you might tell me it's because everyone is depressed. If I tell you that drinking is down, you might tell me it's because everyone is broke. In other words, what the data says makes no difference to the interpretation that you are determined to put on it, namely that things are terrible one way or the other. If it goes up, it's bad; if it goes down, it's bad. The point here is that if you believe in data, try to let it speak before you slap on your own mood, beliefs, or expectations. There's so much data about that you will often be able to find confirmation of your prior beliefs if you simply look around a bit. In other words, data journalism, to me at least, adds little value if you are not open-minded. It is only as objective as you strive to make it, and not by virtue of being based on numbers.
- Uncertainty is OK. We associate numbers with authority and certainty. Often as not, the answer is that there is no answer, or the answer may be the best we have but still wouldn't hit a barn door for accuracy. I think we should say these things. If that sounds like a good way of killing stories, I'd argue that it's a great way of raising new questions. Equally, there can often be more than one legitimate way of cutting the data. Numbers don't have to be either true or false.
- The investigation is a story. The story of how you tried to find out can make great journalism, as you go from one piece of evidence to another—and this applies in spades to the evidence from data, where one number will seldom do. Different sources provide new angles, new ideas, and richer understanding. I wonder if we're too hung up on wanting to be authoritative and tell people the answer—and so we miss a trick by not showing the sleuthing.
- The best questions are the old ones: is that really a big number? Where did it come from? Are you sure it counts what you think it counts? These are generally just prompts to think around the data, the stuff at the edges that got squeezed by looking at a single number, the real-life complications, the wide range of other potential comparisons over time, group or geography; in short, context.

— *Michael Blastland, freelance journalist*

Basic Steps in Working with Data

There are at least three key concepts you need to understand when starting a data project:

- Data requests should begin with a list of questions you want to answer.

- Data often is messy and needs to be cleaned.
- Data may have undocumented features.

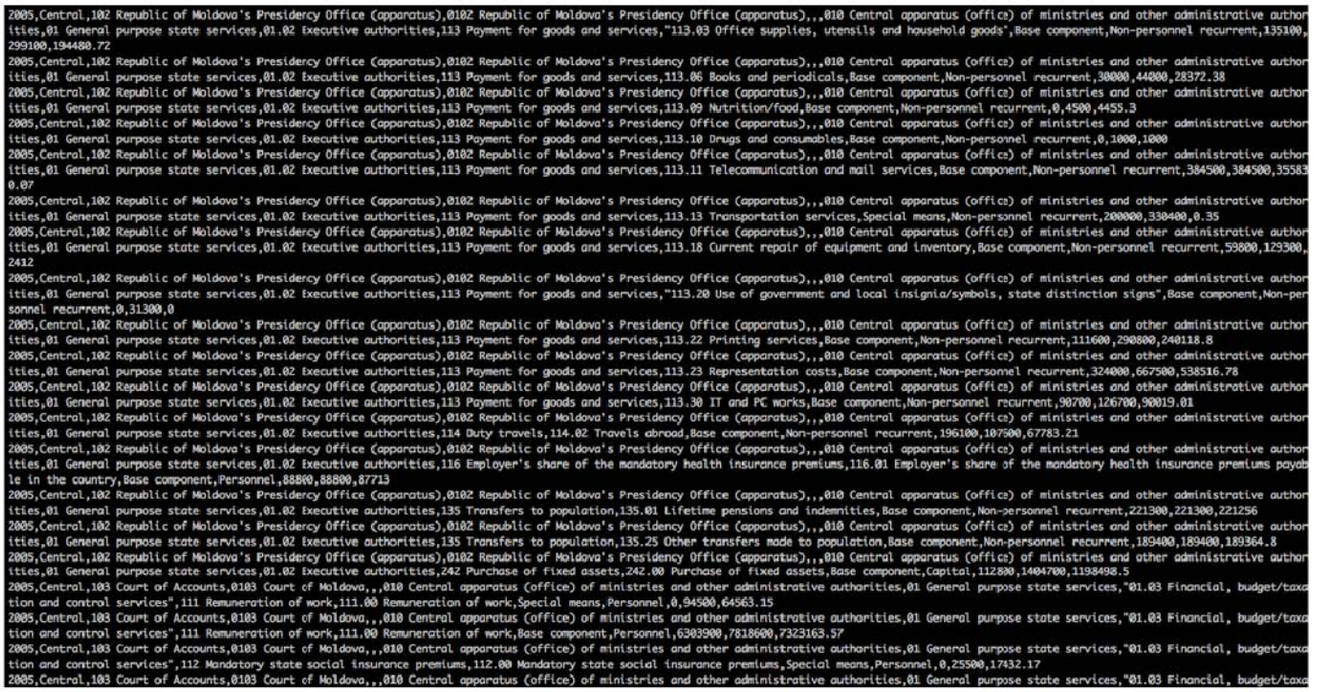


Figure 5-2. Messy data

Know the Questions You Want to Answer

In many ways, working with data is like interviewing a live source. You ask questions of the data and get it to reveal the answers. But just as a source can only give answers about which he has information, a dataset can only answer questions for which it has the right records and the proper variables. This means that you should consider carefully what questions you need to answer even before you acquire your data. Basically, you work backward. First, list the data-evidenced statements you want to make in your story. Then decide which variables and records you would have to acquire and analyze in order to make those statements.

Consider an example involving local crime reports. Let's say you want to do a story looking at crime patterns in your city, and the statements you want to make involve the times of day and the days of a week in which different kinds of crimes are most likely to happen, as well as what parts of town are hot spots for various crime categories.

You would realize that your data request has to include the date and the time each crime was reported, the kind of crime (murder, theft, burglary, etc.) as well as the address of where the crime occurred. So Date, Time, Crime Category, and Address are the minimum variables you need to answer those questions.

But be aware that there are a number of potentially interesting questions that this four-variable dataset *can't* answer, like the race and gender of victims, or the total value of stolen property, or which officers are most productive in making arrests. Also, you may only be able to get records for a certain time period, like the past three years, which would mean you couldn't say anything about whether crime patterns have changed over a longer period of time. Those questions may be outside of the planned purview of your story, and that's fine. But you don't want to get into your data analysis and suddenly decide you need to know what percentage of crimes in different parts of town are solved by arrest.

One lesson here is that it's often a good idea to request *all* the variables and records in the database, rather than the subset that could answer the questions for the immediate story. (In fact, getting all the data can be cheaper than getting a subset, if you have to pay the agency for the programming necessary to write out the subset.) You can always subset the data on your own, and having access to the full dataset will let you answer new questions that may come up in your reporting and even produce new ideas for follow-up stories. It may be that confidentiality laws or other policies mean that some variables, such as the identities of victims or the names of confidential informants, can't be released. But even a partial database is much better than none, as long as you understand which questions the redacted database can and can't answer.

Cleaning Messy Data

One of the biggest problems in database work is that often you will be using data for analysis reasons that has been gathered for bureaucratic reasons. The problem is that the standard of accuracy for those two is quite different.

For example, a key function of a criminal justice system database is to make sure that defendant Jones is brought from the jail to be in front of Judge Smith at the time of his hearing. For that purpose, it really doesn't matter a lot if Jones' birth date is incorrect, or that his street address is misspelled, or even if his middle initial is wrong. Generally, the system still can use this imperfect record to get Jones to Smith's courtroom at the appointed time.

But such errors can skew a data journalist's attempts to discover the patterns in the database. For that reason, the first big piece of work to undertake when you acquire a new dataset is to examine how messy it is and then clean it up. A good quick way to look for messiness is to create frequency tables of the categorical variables, the ones that would be expected to have a relatively small number of different values. (When using Excel, for instance, you can do this by using Filter or Pivot Tables on each categorical variable.)

Take "Gender," an easy example. You may discover that your Gender field includes any of a mix of values like these: Male, Female, M, F, 1, 0, MALE, FEMALE, etc., including misspellings like "Femal". To do a proper gender analysis, you must standardize—decide on M and F, perhaps—and then change all the variations to match the standards. Another common database with these kinds of problems are American campaign finance records, where the Occupation field might list "Lawyer," "Attorney," "Atty," "Counsel," "Trial Lawyer," and any of a wealth of variations and misspellings; again, the trick is to standardize the occupation titles into a shorter list of possibilities.

Data cleanup gets even more problematic when working with names. Are "Joseph T. Smith", "Joseph Smith," "J.T. Smith," "Jos. Smith," and "Joe Smith" all the same person? It may take looking at other variables like address or date of birth, or even deeper research in other records, to decide. But tools like Google Refine can make the cleanup and standardization task faster and less tedious.

Dirty Data

Thanks to the generally strong public records laws in the United States, getting data here isn't as big a problem as it can be in many other countries. But once we get it, we still face the problems of working with data that has been gathered for bureaucratic reasons, not for analytic reasons. The data often is "dirty," with values that aren't standardized. Several times I have received data that doesn't match up to the supposed file layout and data dictionary that accompanies it. Some agencies will insist on giving you the data in awkward formats like .pdf, which have to be converted. Problems like these make you appreciate it when you do get an occasional no-hassle dataset.

— Steve Doig, Walter Cronkite School of Journalism, Arizona State University

Data May Have Undocumented Features

The Rosetta Stone of any database is the so-called data dictionary. Typically, this file (it may be text or PDF or even a spreadsheet) will tell you how the data file is formatted (delimited text, fixed width text, Excel, dBase, etc.), the order of the variables, the names of each variable, and the datatype of each variable (text string, integer, decimal, etc.) You will use this information to help you properly import the data file into the analysis software you intend to use (Excel, Access, SPSS, Fusion Tables, any of various flavors of SQL, etc.)

The other key element of a data dictionary is an explanation of any codes being used by particular variables. For instance, Gender may be coded so that "1=Male" and "0=Female." Crimes may be coded by your jurisdiction's statute numbers for each kind of crime. Hospital treatment records may use any of hundreds of 5-digit codes for the diagnoses of the conditions for which a patient is being treated. Without the data dictionary, these datasets could be difficult or even impossible to analyze properly.

But even with a data dictionary in hand, there can be problems. An example happened to reporters at the Miami Herald in Florida some years ago when they were doing an analysis of the varying rates of punishment that different judges were giving to people arrested for driving while intoxicated. The reporters acquired the conviction records from the court system and analyzed the numbers in the three different punishment variables in the data dictionary: amount of prison time given, amount of jail time given, and amount of fine given. These numbers varied quite a bit amongst the judges, giving the reporters' evidence for a story about how some judges were harsh and some were lenient.

But for every judge, about 1-2 percent of the cases showed no prison time, no jail time, and no fine. So the chart showing the sentencing patterns for each judge included a tiny amount of cases as "No punishment," almost as an afterthought. When the story and chart was printed, the judges howled in complaint, saying the Herald was accusing them of breaking a state law that required that anyone convicted of drunk driving be punished.

So the reporters went back to Clerk of the Court's office that had produced the data file and asked what had caused this error. They were told that the cases in question involved indigent defendants with first-time arrests. Normally they would be given a fine, but they had no money. So the judges were sentencing them to community service, such as cleaning litter along the roads. As it turned out, the law requiring punishment had been passed after the database structure had been created. So all the court clerks knew that in the data, zeros in each of the prison-jail-fine variables meant community service. However, this *wasn't* noted in the data dictionary, and therefore caused a Herald correction to be written.

The lesson in this case is to always ask the agency giving you data if there are any undocumented elements in the data, whether it is newly created codes that haven't been included in the data dictionary, changes in the file layout, or anything else. Also, always examine the results of your analysis and ask "Does this make sense?" The Herald reporters were building the chart on deadline and were so focused on the average punishment levels of each judge that they failed to pay attention to the scant few cases that seemed to show no punishment. They should have asked themselves if it made sense that all the judges seemed to be violating state law, even if only to a tiny degree.

— *Steve Doig, Walter Cronkite School of Journalism, Arizona State University*

Mixed Up, Hidden and Absent Data

I remember a funny situation where we tried to access the Hungarian data on EU farm subsidies: it was all there—but in an excessively heavy PDF document and mixed up with data on national farm subsidies. Our programmers had to work for *hours* before the data was useful.

We also had a pretty interesting time with data about EU fish subsidies, which national payment agencies in all 27 Member States are obliged to disclose. Here's an excerpt from a report we wrote on the topic: "In the United Kingdom, for example, the format of the data varies from very user-friendly HTML search pages to PDF overviews or even lists of recipients in varying formats hidden away at the bottom of press releases. All this is within just one member state. In Germany and Bulgaria, meanwhile, empty lists are published. The appropriate headings are there but without any data."

— *Brigitte Alfter, Journalismfund.eu*

The £32 Loaf of Bread

A story for Wales On Sunday about how much the Welsh government is spending on prescriptions for gluten-free products, contained the headline figure that it was paying £32 for a loaf of bread. However, this was actually 11 loaves that cost £2.82 each.

The figures, from a Welsh Assembly written answer and a Welsh NHS statistics release, listed the figure as cost per prescription item. However, they gave no additional definition in the data dictionary of what a prescription item might refer or how a separate quantity column might define it.

The assumption was that it referred to an individual item—e.g., a loaf of bread—rather than what it actually was, a pack of several loaves.

No one, neither the people who answered the written answer nor the press office, when it was put to them, raised the issue about quantity until the Monday after the story was published.

So do not assume that the background notes for government data will help explain what information is being presented or that the people responsible for the data will realize the data is not clear even when you tell them your mistaken assumption.

Generally newspapers want things that make good headlines, so unless something obviously contradicts an interpretation, it is usually easier to go with what makes a good headline and not check too closely and risk the story collapsing, especially on deadline.

Figure 5-3. Prescriptions for gluten-free bread costing Welsh taxpayers £32 (WalesOnline)

But journalists have a responsibility to check ridiculous claims, even if it means that this drops the story down the news list.

— *Claire Miller, WalesOnline*

Start With the Data, Finish With a Story

To draw your readers in, you have to be able to hit them with a headline figure that makes them sit up and take notice. You should almost be able to read the story without having to know that it comes from a dataset. Make it exciting and remember who your audience is as you go.

One example of this can be found in a project carried out by the Bureau of Investigative Journalism using the EU Commission's Financial Transparency System. The story was constructed by approaching the dataset with specific queries in mind.

We looked through the data for key terms like "cocktail," "golf," and "away days." This allowed us to determine what the Commission had spent on these items and raised plenty of questions and storylines to follow up.

But key terms don't always give you what you want—sometimes you have to sit back and think about what you're really asking for. During this project we also wanted to find out how much commissioners spent on private jet travel but as the dataset didn't contain the phrase "private jet" we had to get the name of their travel providers by other means. Once we knew the name of the service provider to the Commission, "Abelag," we were able to query the data to find out how much was being spent on services provided by Abelag.

With this approach, we had a clearly defined objective in querying the data—to find a figure that would provide a headline; the color followed.

Another approach is to start with a blacklist and look for exclusions. An easy way to pull storylines from data is to know what you shouldn't find in there! A good example of how this can work is illustrated by the collaborative EU Structural Funds project between the Financial Times and the Bureau of Investigative Journalism.

We queried the data, based on the Commission's own rules about what kinds of companies and associations should be prohibited from receiving structural funds. One example was expenditure on tobacco and tobacco producers.

By querying the data with the names of tobacco companies, producers, and growers, we found data that revealed that British American Tobacco was receiving €1.5m for a factory in Germany.

As the funding was outside the rules of Commission expenditure, it was a quick way to find a story in the data.

You never know what you might find in a dataset, so just have a look. You have to be quite bold and this approach generally works best when trying to identify obvious characteristics that will show up through filtering (the biggest, extremes, most common, etc.).

— *Caelainn Barr, Citywire*

Data Stories

Data journalism can sometimes give the impression that it is mainly about presentation of data—such as visualizations that quickly and powerfully convey an understanding of an aspect of the figures, or interactive searchable databases that allow individuals to look up places like their own local street or hospital. All this can be very valuable, but like other forms of journalism, data journalism should also be about stories. So what are the kinds of stories you can find in data? Based on my experience at the BBC, I have drawn up a list, or "typology," of different kinds of data stories.

I think it helps to bear this list below in mind, not only when you are analyzing data, but also at the stage before that, when you are collecting it (whether looking for publicly available datasets or compiling freedom of information requests).

Measurement

> The simplest story; counting or totaling something: "Local councils across the country spent a total of £x billion on paper clips last year." But it's often difficult to know if that's a lot or a little. For that, you need context, which can be provided by:
>
> Proportion
> "Last year local councils spent two-thirds of their stationery budget on paper clips."
>
> Internal comparison
> "Local councils spend more on paper clips than on providing meals-on-wheels for the elderly."
>
> External comparison
> "Council spending on paper clips last year was twice the nation's overseas aid budget."

There are also other ways of exploring the data in a contextual or comparative way:

Change over time
> "Council spending on paper clips has trebled in the past four years."

"League tables"
> These are often geographical or by institution, and you must make sure the basis for comparison is fair (e.g., taking into account the size of the local population). "Borsetshire Council spends more on paper clips for each member of staff than any other local authority, at a rate four times the national average."

Or you can divide the data subjects into groups:

Analysis by categories
> "Councils run by the Purple Party spend 50% more on paper clips than those controlled by the Yellow Party."

Or you can relate factors numerically:

Association
> "Councils run by politicians who have received donations from stationery companies spend more on paper clips, with spending increasing on average by £100 for each pound donated."

But, of course, always remember that correlation and causation are not the same thing.

So if you're investigating paper clip spending, are you also getting the following figures?

- Total spending to provide context?
- Geographical/historical/other breakdowns to provide comparative data?
- The additional data you need to ensure comparisons are fair, such as population size?
- Other data that might provide interesting analysis to compare or relate the spending to?

— *Martin Rosenbaum, BBC*

Data Journalists Discuss Their Tools of Choice

Psssss. That is the sound of your data decompressing from its airtight wrapper. Now what? What do you look for? And what tools do you use to get stuck in? We asked data journalists to tell us a bit about how they work with data. Here is what they said:

At the Guardian Datablog, we really like to interact with our readers and allowing them to replicate our data journalism quickly means they can build on the work we do and sometimes spot things we haven't. So the more intuitive the data tools, the better. We try to pick tools that anyone could get the hang of without learning a programming language or having special training and without a hefty fee attached.

We're currently using Google products quite heavily for this reason. All the datasets we tidy and release are available as a Google Spreadsheet, which means people with a Google account can download the data, import it into their own account and make their own charts, sort the data and create pivot tables, or they can import the data into a tool of their choice.

To map data, we use Google Fusion tables. When we create heat maps in Fusion, we share our KML shape files so that readers can download and build their own heat maps—maybe adding extra layers of data onto the Datablog's original map. The other nice feature of these Google tools is that they work on the many platforms our readers use to access the blog, such as their desktop, their mobile, and tablets.

In addition to Google Spreadsheets and Fusion, we use two other tools in our daily work. The first is Tableau, to visualize multi-dimensional datasets; and the second is ManyEyes, for quick analysis of data. None of these tools are perfect, so we continue to look for better visualization tools that our readers will enjoy.

— Lisa Evans, *the Guardian*

Am I ever going to be a coder? Very unlikely! I certainly don't think that all reporters need to know how to code. But I do think it is very valuable for them to have a more general awareness of what is possible and know how to talk to coders.

If you're starting out, walk, don't run. You need to persuade your colleagues and editors that working with data can get you stories that you wouldn't otherwise get and that it's well worth doing. Once they see the value of this approach, you can expand into doing more complex stories and projects.

My advice is to learn Excel and do some simple stories first. Start out small and work your way up to database analysis and mapping. You can do so much in Excel—it's an extremely powerful tool and most people don't even use a fraction of its functionality. If you can, go on a course on Excel for journalists, such as the one offered by the Centre for Investigative Journalism.

With respect to interpreting data: don't take this lightly. You have to be conscientious. Pay attention to detail and question your results. Keep notes on how you're processing the data and keep a copy of the original data. It is easy to make a mistake. I always do my analysis two or three times practically from scratch. Even better would be to get your editor or someone else to analyze the data separately and compare the results.

— Cynthia O'Murchu, *Financial Times*

The ability to write and deploy complex software as quickly as a reporter can write a story is a pretty new thing. It used to take a lot longer. Things changed thanks to the development of two free/open source rapid development frameworks: Django and Ruby on Rails, both of which were first released in the mid-2000s.

Django, which is built on top of the Python programming language, was developed by Adrian Holovaty and a team working in a newsroom—the Lawrence Journal-World in Lawrence, Kansas. Ruby on Rails was developed in Chicago by by David Heinemeier Hansson and 37Signals, a web application company.

Though the two frameworks take different approaches to the "MVC pattern," they're both excellent and make it possible to build even very complex web applications very quickly. They take away some of the rudimentary work of building an app. Things like creating and fetching items from the database, and matching URLs to specific code in an app are built into the frameworks, so developers don't need to write code to do basic things like that.

While there hasn't been a formal survey of news app teams in the U.S., it is generally understood that most teams use one of these two frameworks for database-backed news apps. At ProPublica, we use Ruby on Rails.

The development of rapid web server "slice" provisioning services like Amazon Web Services also took away some of what used to make deploying a web app a slow process.

Apart from that, we use pretty standard tools to work with data: Google Refine and Microsoft Excel to clean data; SPSS and R to do statistics; ArcGIS and QGIS to do GIS; Git for source code management; TextMate, Vim and Sublime Text for writing code; and a mix of MySQL, PostgreSQL and SQL Server for databases. We built our own JavaScript framework called "Glass" that helps us build front-end heavy apps in JavaScript very quickly.

— Scott Klein, *ProPublica*

Sometimes the best tool can be the simplest tool—the power of a spreadsheet is easy to underestimate. But using a spreadsheet back when everything was in DOS enabled me to understand a complex formula for the partnership agreement for the owners of The Texas Rangers—back when George W. Bush was one of the key owners. A spreadsheet can help me flag outliers or mistakes in calculations. I can write clean-up scripts and more. It is a basic in the toolbox for a data journalist.

That said, my favorite tools have even more power—SPSS for statistical analysis and mapping programs that enable me to see patterns geographically.

— Cheryl Phillips, *The Seattle Times*

I'm a big fan of Python. Python is a wonderful open source programming language that is easy to read and write (e.g., you don't have to type a semi-colon after each line). More importantly, Python has a tremendous user base and therefore has plugins (called packages) for literally everything you need.

I would consider Django as something rarely needed by data journalists. It is a Python web application framework—that is, a tool to create big, database-driven web applications. It is definitely too heavyweight for small interactive infographics.

I also use QGis, which is an open source toolkit providing a wide range of GIS functionality needed by data journalists who deal with geodata every now and then. If you need to convert geospatial data from one format into another, then QGis is what you need. It can handle nearly every geodata format out there (Shapefiles, KML, GeoJSON, etc.). If you need to cut out a few regions, QGis can do this as well. Plus there is a huge community around QGis, so you find tons of resources like tutorials out in the web.

R was created mainly as a scientific visualization tool. It is hard to find any visualization method or data wrangling technique that is not already built into R. R is a universe in its own, the mecca of visual data analysis. One drawback is that you need to learn (yet another) programming language, as R has its own language. But once you have taken the initial climb on the learning curve, there's no tool more powerful than R. Trained data journalists can use R to analyze huge datasets that extend the limits of Excel (for instance, if you have a table with a million rows).

What's really nice about R is that you're able to keep an exact "protocol" of what you're doing with the data throughout the entire process—from reading a CSV file to generating charts. If the data changes, you can regenerate the chart using one click. If someone is curious about the integrity of your chart, you can show the exact source, which allows everyone to recreate the exact chart on their own (or maybe find the mistakes you made).

NumPy + MatPlotLib is kind of a way of doing the same thing in Python. It's an option if you're already well trained in Python. In fact, NumPy and MatPlotLib are two examples of Python packages. They can be used for data analysis and data visualization, and are both limited to static visualizations. They cannot be used to create interactive charts with tooltips and more advanced stuff.

I'm not using MapBox, but I've heard it is a great tool if you want to provide more sophisticated maps based on OpenStreetMap. It allows you, for instance, to customize the map styles (colors, labels, etc). There's also a companion of MapBox, called Leaflet. Leaflet is basically a higher level JavaScript library for mapping that allows you to easily switch between map providers (OSM, MapBox, Google Maps, Bing, etc.).

RaphaelJS is a rather low-level visualization library that allows you to work with basic primitives (like circles, lines, text), and to animate them, add interactions, etc. There's nothing like a ready-to-use bar chart in it, so you have to draw a set of rectangles yourself.

However, the good thing about Raphael is that everything you create will also work in Internet Explorer. That's not the case with many other (amazing) visualization libraries like d3. Sadly, so many users are still using IE and no newsroom can afford to ignore 30% of their users.

Besides RaphaelJS, there's also the option of creating a Flash fallback for IE. That is basically what The New York Times is doing. This means that you have to develop each application twice.

I'm still not convinced about the "best" process of shipping visualization for IE and modern browsers. Often I find that RaphaelJS applications can run horribly slow on IE, around ten times slower than they run in Flash using modern browsers. So Flash fallbacks might be a better option if you want to provide high-quality animated visualizations for all users.

— Gregor Aisch, *Open Knowledge Foundation*

My go-to tool is Excel, which can handle the majority of CAR problems and has the advantages of being easy to learn and available to most reporters. When I need to merge tables, I typically use Access, but then export the merged table back into Excel for further work. I use ESRI's ArcMap for geographic analyses; it's powerful and is used by the agencies that gather geocoded data. TextWrangler is great for examining text data with quirky layouts and delimiters, and can do sophisticated search-and-replace with regular expressions. When statistical techniques like linear regression are needed, I use SPSS; it has a friendly point-and-click menu. For really heavy lifting, like working with datasets that have millions of records that may need serious filtering and programmed variable transformations, I use SAS software.

— Steve Doig, *Walter Cronkite School of Journalism*

Our tools of choice include Python and Django for hacking, scraping, and playing with data, and PostGIS, QGIS, and the MapBox toolkit for building crazy web maps. R and NumPy + MatPlotLib are currently battling for supremacy as our kit of choice for exploratory data analysis, though our favorite data tool of late is homegrown: CSVKit. More or less everything we do is deployed in the cloud.

— Brian Boyer, *Chicago Tribune*

At La Nacion we use:

- Excel for cleaning, organizing and analyzing data;
- Google Spreadsheets for publishing and connecting with services such as Google Fusion Tables and the Junar Open Data Platform;
- Junar for sharing our data and embedding it in our articles and blog posts;
- Tableau Public for our interactive data visualizations;
- Qlikview, a very fast business intelligence tool to analyze and filter large datasets;
- NitroPDF for converting PDFs to text and Excel files; and
- Google Fusion Tables for map visualizations.

— Angélica Peralta Ramos, *La Nacion (Argentina)*

As a grassroots community without any technical bias, we at Transparency Hackers use a lot of different tools and programming languages. Every member has it's own set of preferences and this great variety is both our strength and our weakness. Some of us are actually building a "Transparency Hacker Linux Distribution," which we could live-boot anywhere and start hacking data. This toolkit has some interesting tools and libraries for handling data like Refine, RStudio and OpenOffice Calc (usually an overlooked tool by savvy people, but really useful for quick/small stuff). Also, we've been using Scraperwiki quite a lot to quickly prototype and save data results online.

For data visualization and graphs, there are a lot of tools we like. Python and NumPy are pretty powerful. A few people in the community have been playing with R, but at the end of the day I still think Javascript plotting graph libs like d3, Flot, and RaphaelJS end up being used in the majority of our projects. Finally, we've been experimenting a lot with mapping, and Tilemill has been a really interesting tool to work with.

— Pedro Markun, *Transparência Hacker*

Using Data Visualization to Find Insights in Data

Visualization is critical to data analysis. It provides a front line of attack, revealing intricatestructure in data that cannot be absorbed in any other way. We discover unimagined effects, and we challenge imagined ones.

— William S. Cleveland (from Visualizing Data, *Hobart Press)*

Data by itself, consisting of bits and bytes stored in a file on a computer hard drive, is invisible. In order to be able to see and make any sense of data, we need to visualize it. In this section I'm going to use a broader understanding of the term *visualizing*, that includes even pure textual representations of data. For instance, just loading a dataset into a spreadsheet software can be considered as data visualization. The invisible data suddenly turns into a visible "picture" on our screen. Thus, the question should not be whether journalists need to visualize data or not, but which kind of visualization may be the most useful in which situation.

In other words: when does it makes sense to go beyond the table visualization? The short answer is: *almost always*. Tables alone are definitely not sufficient to give us an overview of a dataset. And tables alone don't allow us to immediately identify patterns within the data. The most common example here are geographical patterns that can only be observed after visualizing data on a map. But there are also other kinds of patterns, which we will see later in this section.

Using Visualization to Discover Insights

It is unrealistic to expect that data visualization tools and techniques will unleash a barrage of ready-made stories from datasets. There are no rules, no "protocol" that will guarantee us a story. Instead, I think it makes more sense to look for "insights," which can be artfully woven into stories in the hands of a good journalist.

Every new visualization is likely to give us some insights into our data. Some of those insights might be already known (but perhaps not yet proven), while other insights might be completely new or even surprising to us. Some new insights might mean the beginning of a story, while others could just be the result of errors in the data, which are most likely to be found by visualizing the data.

In order to make finding insights in data more effective, I find the process discussed in Figure 5-4 (and the rest of this section) to be very helpful.

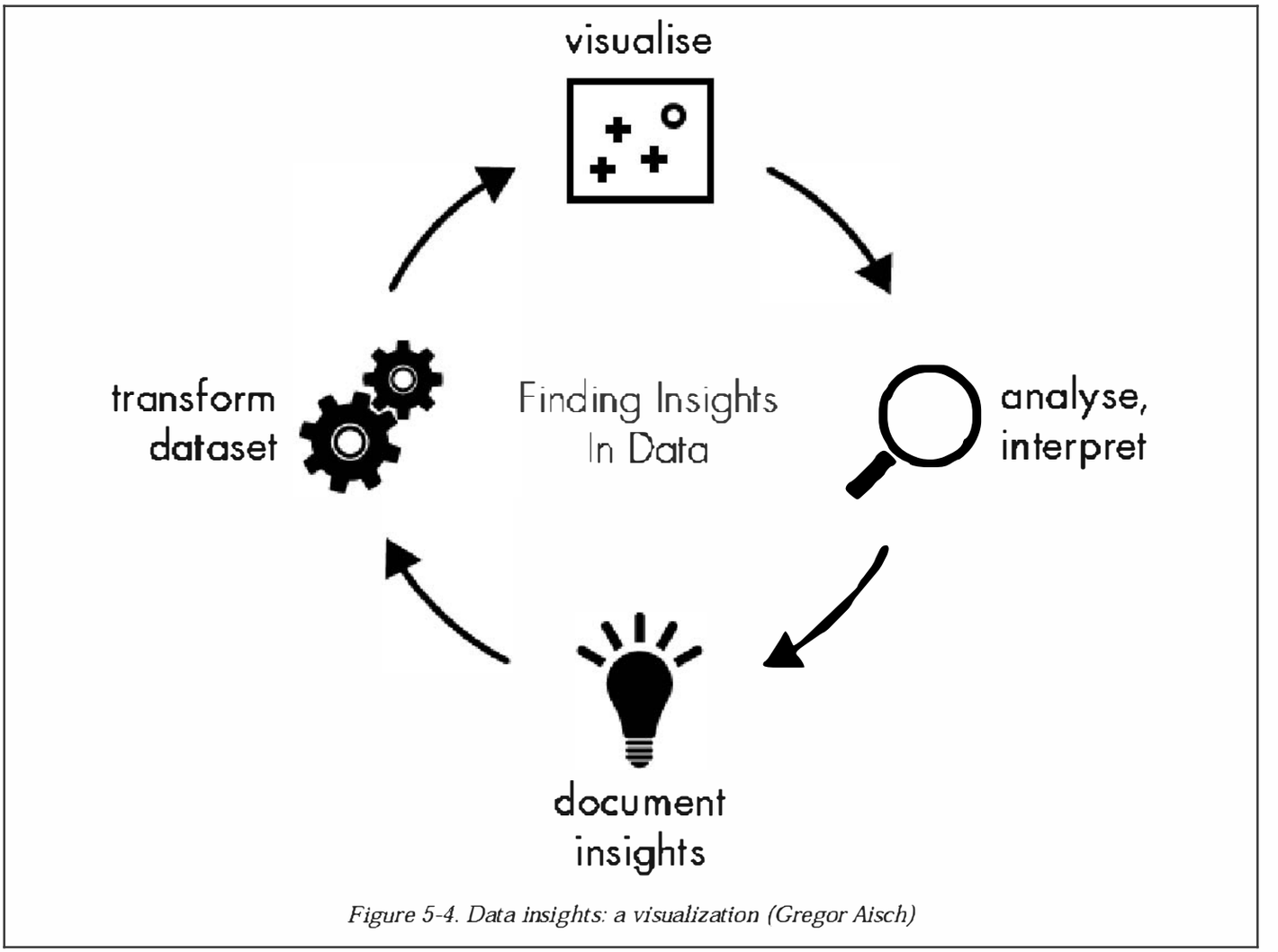
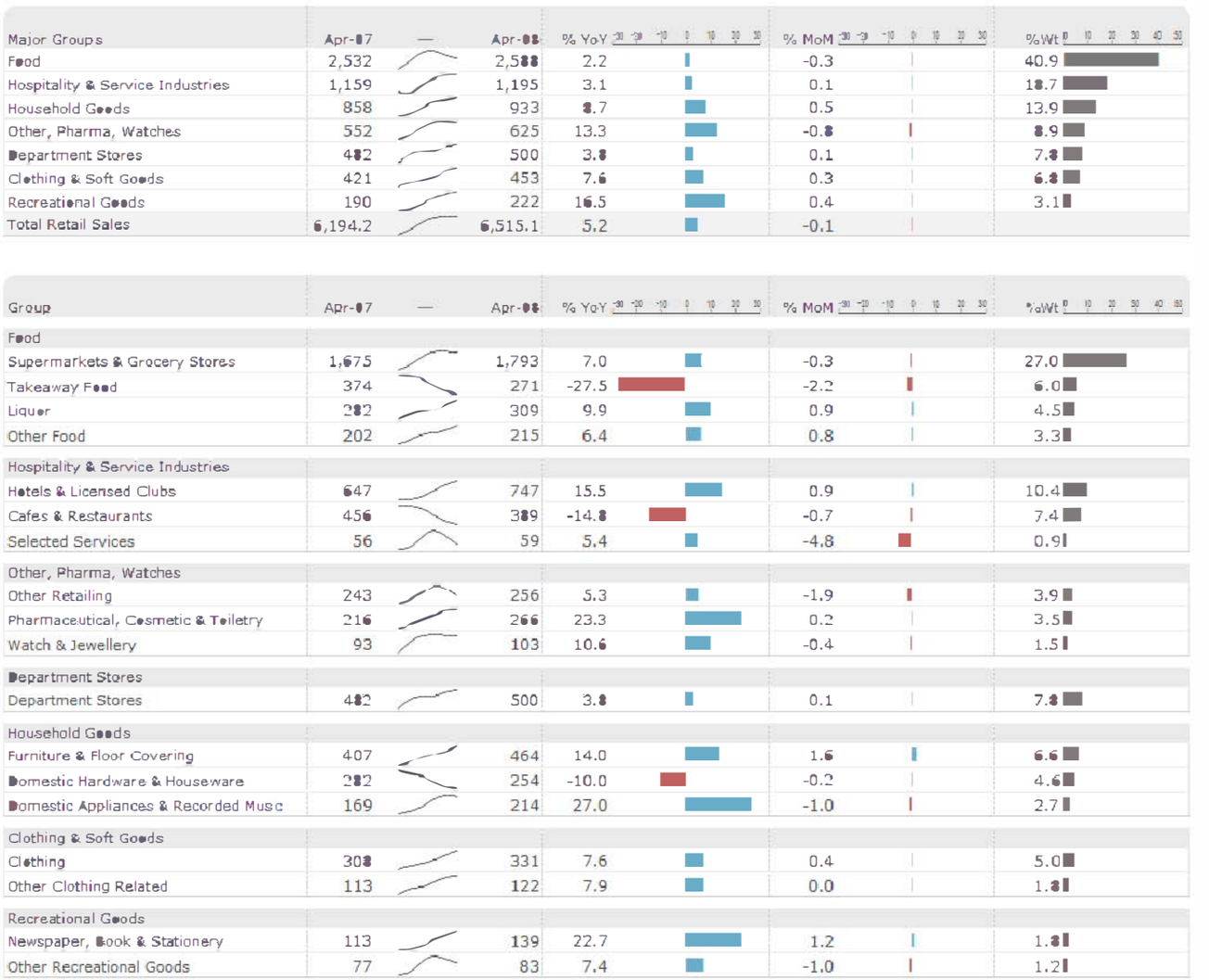


Figure 5-4. Data insights: a visualization (Gregor Aisch)

Learn how to visualize data

Visualization provides a unique perspective on the dataset. You can visualize data in lots of different ways.

Tables are very powerful when you are dealing with a relatively small number of data points. They show labels and amounts in the most structured and organized fashion and reveal their full potential when combined with the ability to sort and filter the data. Additionally, Edward Tufte suggested including small chart pieces within table columns—for instance, one bar per row or a small line chart (since then also known as a sparkline). But still, as mentioned earlier, tables clearly have their limitations. They are great to show you one-dimensional outliers like the top 10, but they are poor when it comes to comparing multiple dimensions at the same time (for instance, population per country over time).



Major Groups	Apr-07	Apr-08	% YoY	% MoM	%Wt
Food	2,532	2,588	2.2	-0.3	40.9
Hospitality & Service Industries	1,159	1,195	3.1	0.1	18.7
Household Goods	858	933	8.7	0.5	13.9
Other, Pharma, Watches	552	625	13.3	-0.8	8.9
Department Stores	482	500	3.8	0.1	7.8
Clothing & Soft Goods	421	453	7.6	0.3	6.8
Recreational Goods	190	222	16.5	0.4	3.1
Total Retail Sales	6,194.2	6,515.1	5.2	-0.1	

Group	Apr-07	Apr-08	% YoY	% MoM	%Wt
Food					
Supermarkets & Grocery Stores	1,675	1,793	7.0	-0.3	27.0
Takeaway Food	374	271	-27.5	-2.2	6.0
Liquor	282	309	9.9	0.9	4.5
Other Food	202	215	6.4	0.8	3.3
Hospitality & Service Industries					
Hotels & Licensed Clubs	647	747	15.5	0.9	10.4
Cafes & Restaurants	456	389	-14.8	-0.7	7.4
Selected Services	56	59	5.4	-4.8	0.9
Other, Pharma, Watches					
Other Retailing	243	256	5.3	-1.9	3.9
Pharmaceutical, Cosmetic & Toiletry	216	266	23.3	0.2	3.5
Watch & Jewellery	93	103	10.6	-0.4	1.5
Department Stores					
Department Stores	482	500	3.8	0.1	7.8
Household Goods					
Furniture & Floor Covering	407	464	14.0	1.6	6.6
Domestic Hardware & Houseware	282	254	-10.0	-0.2	4.6
Domestic Appliances & Recorded Music	169	214	27.0	-1.0	2.7
Clothing & Soft Goods					
Clothing	308	331	7.6	0.4	5.0
Other Clothing Related	113	122	7.9	0.0	1.8
Recreational Goods					
Newspaper, Book & Stationery	113	139	22.7	1.2	1.8
Other Recreational Goods	77	83	7.4	-1.0	1.2

Figure 5-5. Tips from Tufte: sparklines (Gregor Aisch)

Charts, in general, allow you to map dimensions in your data to visual properties of geometric shapes. There's much written about the effectiveness of individual visual properties, and the short version is this: color is difficult, position is everything. In a scatterplot, for instance, two dimensions are mapped to the to the x- and y-position. You can even display a third dimension to the color or size of the displayed symbols. Line charts are especially suited for showing temporal evolutions, while bar charts are perfect for comparing categorical data. You can stack chart elements on top of each other. If you want to compare a small number of groups in your data, displaying multiple instances of the same chart is a very powerful way (also referred to as small multiples). In all charts you can use different kinds of scales to explore different aspects in your data (e.g., linear or log scale).

In fact, most of the data we're dealing with is somehow related to actual people. The power of maps is to reconnect the data to our very physical world. Imagine a dataset of geolocated crime incidents. Crucially, you want to see *where* the crimes happen. Also maps can reveal geographic relations within the data (e.g., a trend from North to South, or from urban to rural areas).

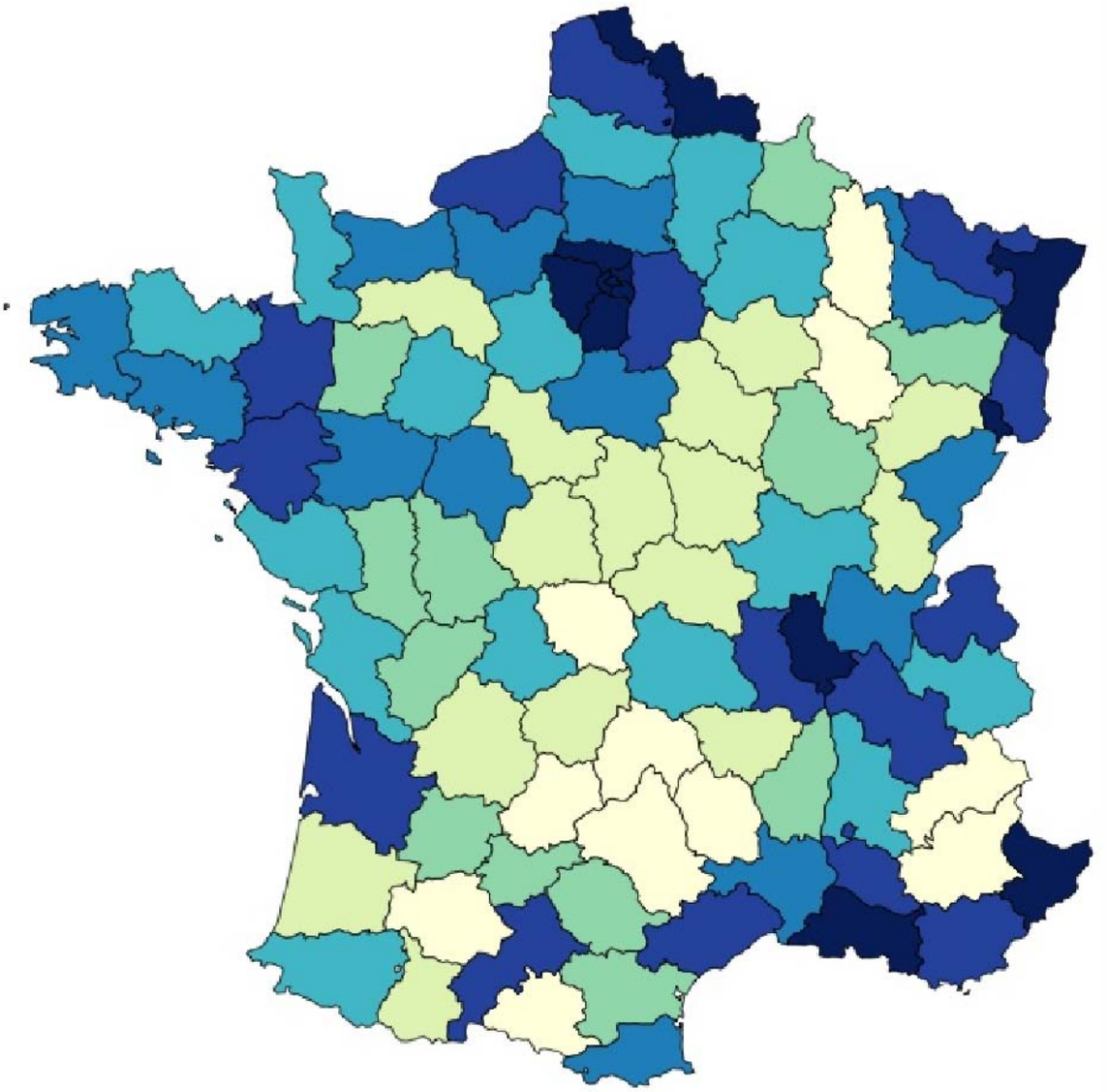


Figure 5-6. Choropleth map (Gregor Aisch)

Speaking of relations, the fourth most important type of visualization is a graph. Graphs are all about showing the interconnections (edges) in your data points (nodes). The position of the nodes is then calculated by more or less complex graph layout algorithms which allow us to immediately see the structure within the network. The trick of graph visualization in general is to find a proper way to model the network itself. Not all datasets already include relations, and even if they do, it might not be the most interesting aspect to look at. Sometimes it's up to the journalist to define edges between nodes. A perfect example of this is the U.S. Senate Social Graph, whose edges connect senators that voted the same in more than 65% of the votes.

Analyze and interpret what you see

Once you have visualized your data, the next step is to learn something from the picture you created. You could ask yourself:

- What can I see in this image? Is it what I expected?
- Are there any interesting patterns?
- What does this mean in the context of the data?

Sometimes you might end up with a visualization that, in spite of its beauty, might seem to tell you nothing of interest about your data. But there is almost always *something* that you can learn from any visualization, however trivial.

Document your insights and steps

If you think of this process as a journey through the dataset, the documentation is your travel diary. It will tell you where you have traveled to, what you have seen there, and how you made your decisions for your next steps. You can even start your documentation before taking your first look at the data.

In most cases when we start to work with a previously unseen dataset, we are already full of expectations and assumptions about the data. Usually there is a reason why we are interested in that dataset that we are looking at. It's a good idea to start the documentation by writing down these initial thoughts. This helps us to identify our bias and reduces the risk of misinterpretation of the data by just finding what we originally wanted to find.

I really think that the documentation is the most important step of the process—and it is also the one we're most likely to tend to skip. As you will see in the example below, the described process involves a lot of plotting and data wrangling. Looking at a set of 15 charts you created might be very confusing, especially after some time has passed. In fact, those charts are only valuable (to you or any other person you want to communicate your findings) if presented in the context in which they have been created. Hence you should take the time to make some notes on things like:

- Why have I created this chart?
- What have I done to the data to create it?
- What does this chart tell me?

Transform data

Naturally, with the insights that you have gathered from the last visualization, you might have an idea of what you want to see next. You might have found some interesting pattern in the dataset which you now want to inspect in more detail.

Possible transformations are:

Zooming
: To have look at a certain detail in the visualization

Aggregation
: To combine many data points into a single group

Filtering
: To (temporarily) remove data points that are not in our major focus

Outlier removal
: To get rid of single points that are not representative for 99% of the dataset.

Let's consider that you have visualized a graph, and what came out of this was nothing but a mess of nodes connected through hundreds of edges (a very common result when visualizing so-called *densely connected networks*). One common transformation step would be to filter some of the edges. If, for instance, the edges represent money flows from donor countries to recipient countries, we could remove all flows below a certain amount.

Which Tools to Use

The question of tools is not an easy one. Every data visualization tool available is good at something. Visualization and data wrangling should be easy and cheap. If changing parameters of the visualizations takes you hours, you won't experiment that much. That doesn't necessarily mean that you don't need to learn how to use the tool. But once you learned it, it should be really efficient.

It often makes a lot of sense to choose a tool that covers both the data wrangling and the data visualization issues. Separating the tasks in different tools means that you have to import and export your data very often. Here's a short list of some data visualization and wrangling tools:

- Spreadsheets like LibreOffice, Excel or Google Docs
- Statistical programming frameworks like R (r-project.org) or Pandas (pandas.pydata.org)
- Geographic Information Systems (GIS) like Quantum GIS, ArcGIS, or GRASS
- Visualization Libraries like d3.js (mbostock.github.com/d3), Prefuse (prefuse.org), or Flare (flare.prefuse.org)
- Data wrangling tools like Google Refine or Datawrangler
- Non-programming visualization software like ManyEyes or Tableau Public (tableausoftware.com/products/public)

The sample visualizations in the next section were created using R, which is kind of a Swiss Army knife of (scientific) data visualization.

An Example: Making Sense of US Election Contribution Data

Let us have look at the US Presidential Campaign Finance database, which contains about 450,000 contributions to US presidential candidates. The CSV file is 60 megabytes and way too big to handle easily in a program like Excel.

In the first step I will explicitly write down my initial assumptions on the FEC contributions dataset:

- Obama gets the most contributions (since he is the president and has the greatest popularity).
- The number of donations increases as the time moves closer to election date.
- Obama gets more small donations than Republican candidates.

To answer the first question, we need to *transform* the data. Instead of each single contribution, we need to sum the total amounts contributed to each candidate. After *visualizing* the results in a sorted table, we can confirm our assumption that Obama would raise the most money:

Candidate	Amount ($)
Obama, Barack	72,453,620.39
Romney, Mitt	50,372,334.87
Perry, Rick	18,529,490.47
Paul, Ron	11,844,361.96
Cain, Herman	7,010,445.99
Gingrich, Newt	6,311,193.03
Pawlenty, Timothy	4,202,769.03
Huntsman, Jon	2,955,726.98
Bachmann, Michelle	2,607,916.06
Santorum, Rick	1,413,552.45
Johnson, Gary Earl	413,276.89
Roemer, Charles E. *Buddy* III	291,218.80
McCotter, Thaddeus G	37,030.00

Even though this table shows the minimum and maximum amounts and the order, it does not tell very much about the underlying patterns in candidate ranking. Figure 5-7 is another view on the data, a chart type that is called a "dot chart," in which we can see everything that is shown in the table *plus* the patterns within the field. For instance, the dot chart allows us to immediately compare the distance between Obama and Romney, and Romney and Perry, without needing to subtract values. (Note: the dot chart was created using R. You can find links to the source code at the end of this chapter).

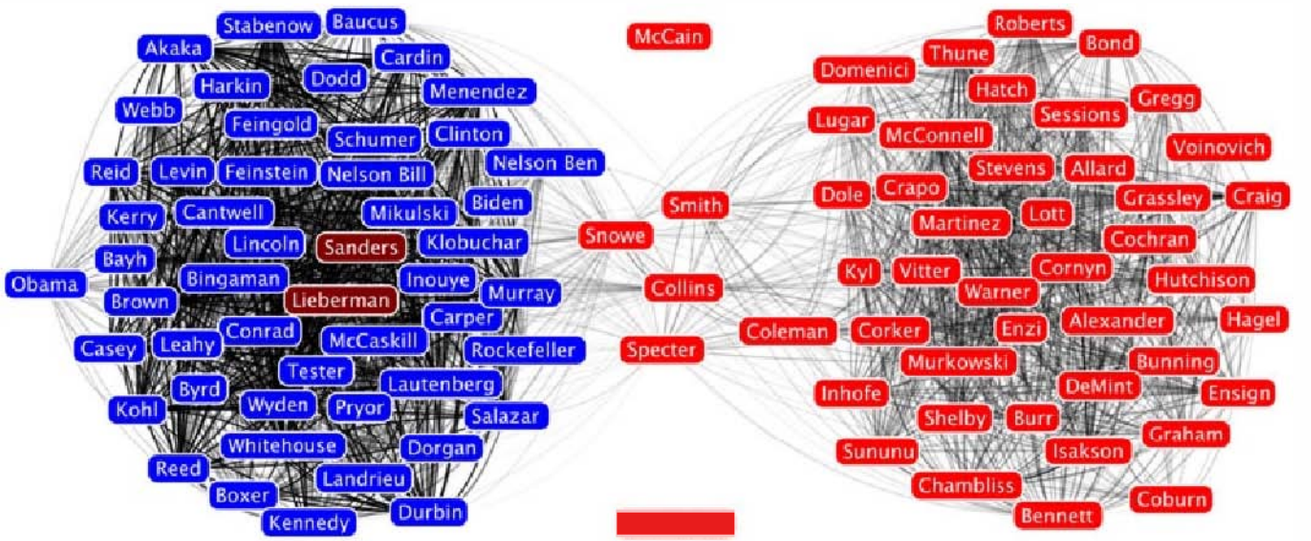


Figure 5-7. Visualizations to spot underlying patterns (Gregor Aisch)

Now, let us proceed with a bigger picture of the dataset. As a first step, I *visualized* all contributed amounts over time in a simple plot. We can see that almost all donations are very, very small compared to three really big outliers. Further investigation reveals that these huge contributions are coming from the "Obama Victory Fund 2012" (also known as Super PAC) and were made on June 29th ($450k), September 29th ($1.5mio), and December 30th ($1.9mio).

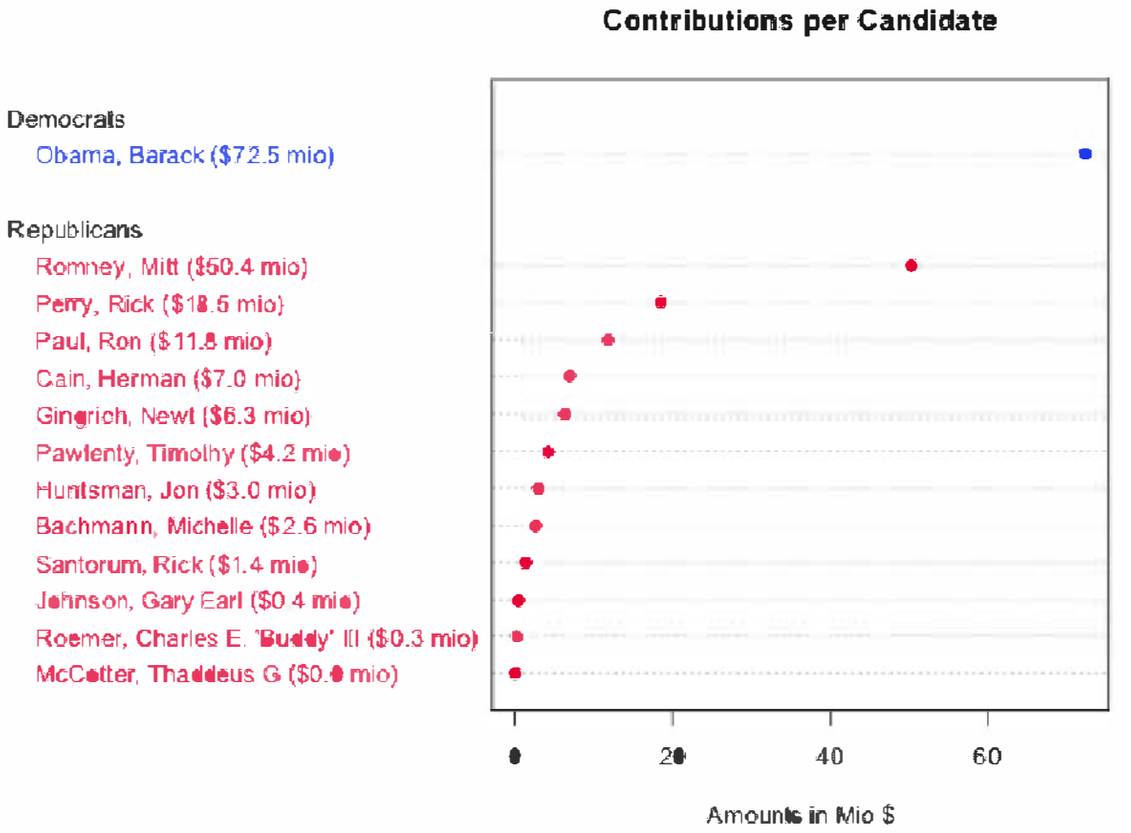


Figure 5-8. Three clear outliers (Gregor Aisch)

While the contributions by Super PACs alone is undoubtedly the biggest story in the data, it might be also interesting to look beyond it. The point now is that these big contributions disturb our view on the smaller contributions coming from individuals, so we're going to remove them from the data. This transform is commonly known as outlier removal. After visualizing again, we can see that most of the donations are within the range of $10k and -$5k.

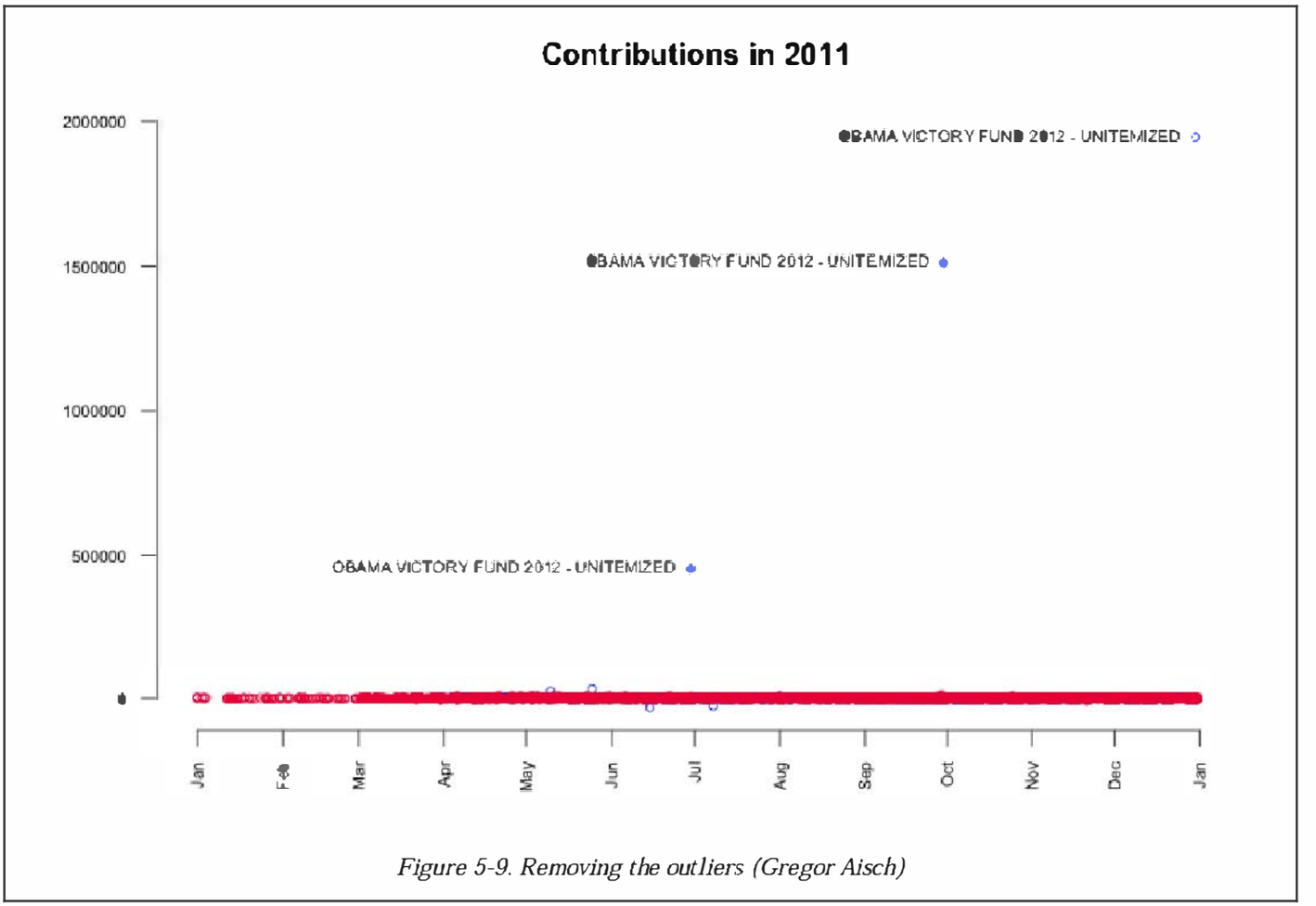


Figure 5-9. Removing the outliers (Gregor Aisch)

According to the contribution limits placed by the FECA, individuals are not allowed to donate more than $2500 to each candidate. As we see in the plot, there are numerous donations made above that limit. In particular, two big contributions in May attract our attention. It seems that they are *mirrored* in negative amounts (refunds) in June and July. Further investigation in the data reveals the following transactions:

- On May 10, *Stephen James Davis*, San Francisco, employed at Banneker Partners (attorney), has donated **$25,800** to Obama.
- On May 25, *Cynthia Murphy*, Little Rock, employed at the Murphy Group (public relations), has donated **$33,300** to Obama.
- On June 15, the amount of **$30,800** was refunded to *Cynthia Murphy*, which reduced the donated amount to **$2500**.
- On July 8, the amount **$25,800** was refunded to *Stephen James Davis*, which reduced the donated amount to $0.

What's interesting about these numbers? The $30,800 refunded to Cynthia Murphy equals the maximum amount individuals may give to national party committees per year. Maybe she just wanted to combine both donations in one transaction, which was rejected. The $25,800 refunded to Stephen James Davis possibly equals the $30,800 minus $5000 (the contribution limit to any other political committee).

Another interesting finding in the last plot is a horizontal line pattern for contributions to Republican candidates at $5000 and -$2500. To see them in more detail, I visualized just the Republican donations. The resulting graphic is one great example of patterns in data that would be invisible without data visualization.

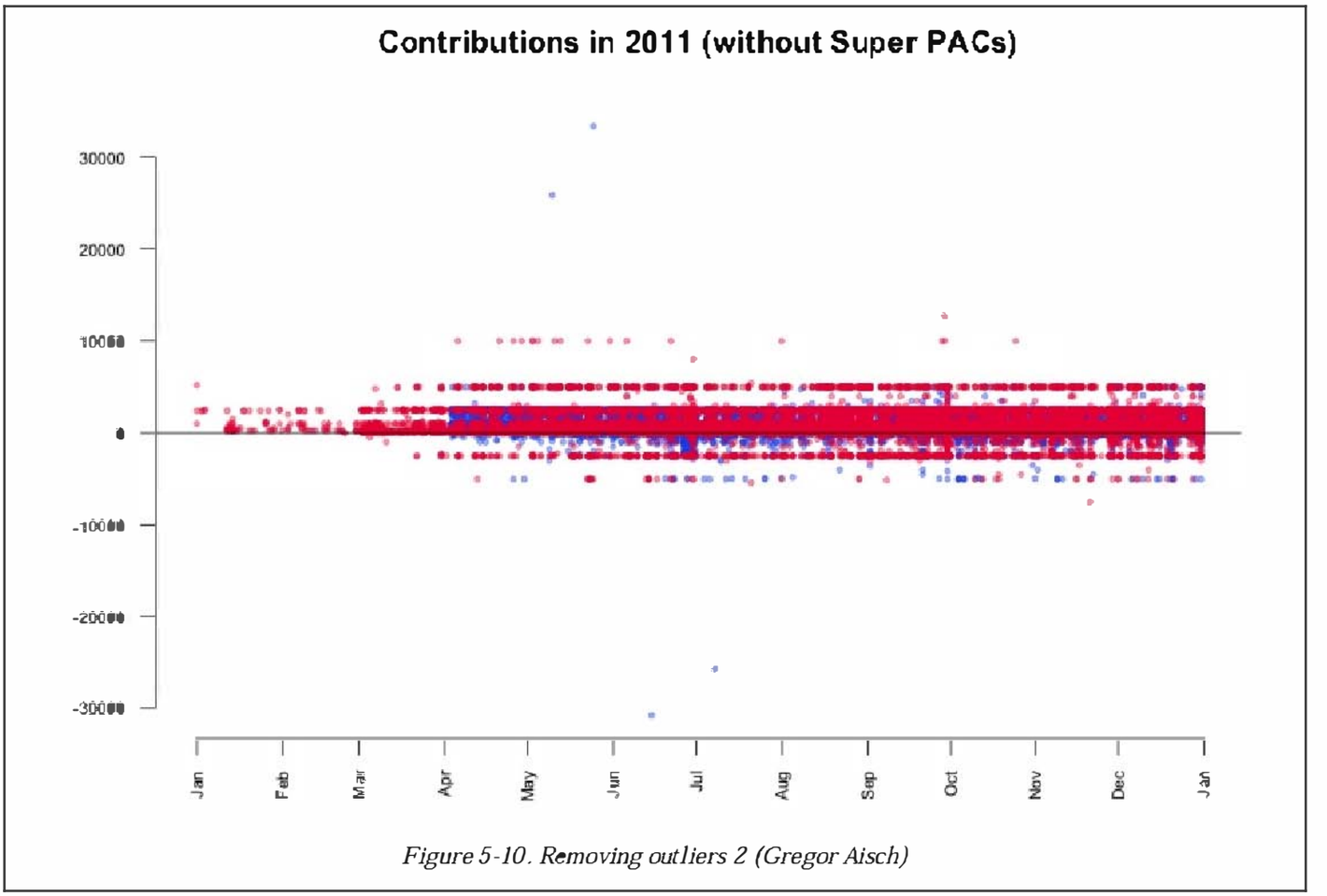


Figure 5-10. Removing outliers 2 (Gregor Aisch)

What we can see is that there are many $5000 donations to Republican candidates. In fact, a look up in the data returns that these are 1243 donations, which is only 0.3% of the total number of donations, but since those donations are evenly spread across time, the line appears. The interesting thing about the line is that donations by individuals were limited to $2500. Consequently, every dollar above that limit was refunded to the donors, which results in the second line pattern at -$2500. In contrast, the contributions to Barack Obama don't show a similar pattern.

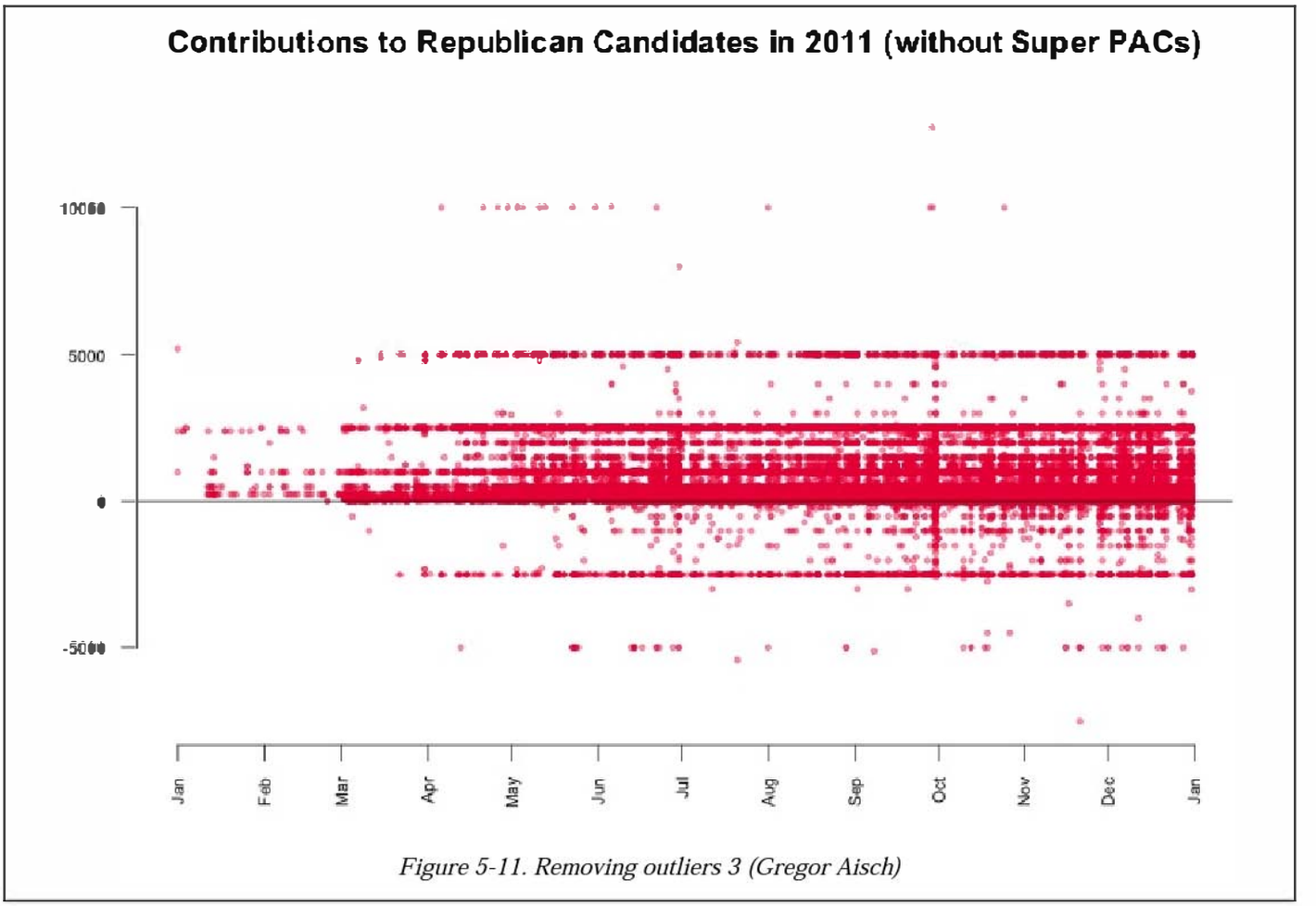


Figure 5-11. Removing outliers 3 (Gregor Aisch)

So, it might be interesting to find out why thousands of Republican donors did not notice the donation limit for individuals. To further analyze this topic, we can have a look at the total number of $5k donations per candidate.

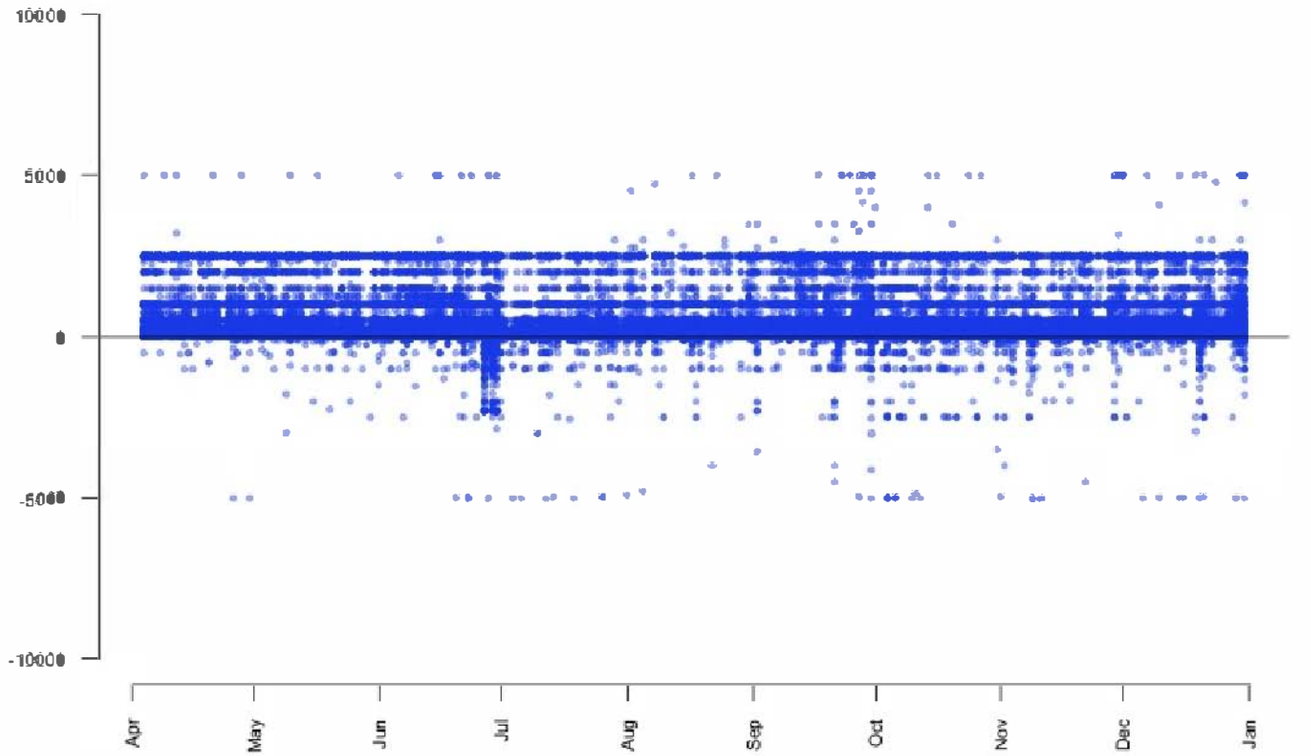


Figure 5-12. Donations per candidate (Gregor Aisch)

Of course, this is a rather distorted view since it does not consider the total amounts of donations received by each candidate. The next plot shows the percentage of $5k donations per candidate.

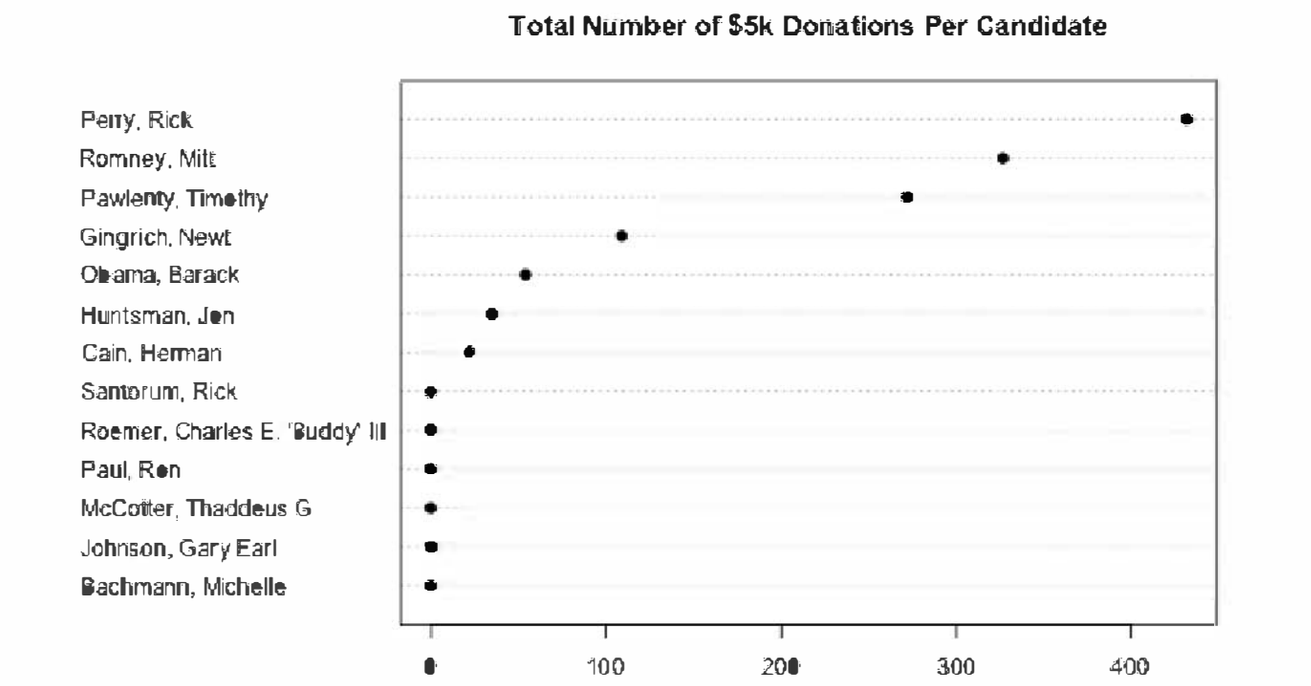


Figure 5-13. Where does the senator's money come from?: donations per candidate (Gregor Aisch)

What To Learn From This

Often, such a visual analysis of a new dataset feels like an exciting journey to an unknown country. You start as a foreigner with just the data and your assumptions, but with every step you make, with every chart you render, you get new insights about the topic. Based on those insights, you make decisions for your next steps and what issues are worth further investigation. As you might have seen in this chapter, this process of visualizing, analyzing and transformation of data could be repeated nearly infinitely.

Get the Source Code

All of the charts shown in this chapter were created using the wonderful and powerful software R. Created mainly as a scientific visualization tool, it is hard to find any visualization or data wrangling technique that is not already built into R. For those who are interested in how to visualize and wrangle data using R, here's the source code of the charts generated in this chapter:

- dotchart: contributions per candidate
- plot: all contributions over time
- plot: contributions by authorized committees

There is also a wide range of books and tutorials available.

— *Gregor Aisch, Open Knowledge Foundation*