1

The Journey to Data-Driven Security

"It ain't so much the things we don't know that get us into trouble. It's the things we know that just ain't so."

Josh Billings, Humorist

This book isn't really about data analysis and visualization.

Yes, almost every section is focused on those topics, but being able to perform good data analysis and produce informative visualizations is just a means to an end. You never (okay, rarely) analyze data for the sheer joy of analyzing data. You analyze data and create visualizations to gain new perspectives, to find relationships you didn't know existed, or to simply discover new information. In short, you do data analysis and visualizations to learn, and that is what this book is about. You want to learn how your information systems are functioning, or more importantly how they are failing and what you can do to fix them.

The cyber world is just too large, has too many components, and has grown far too complex to simply rely on intuition. Only by augmenting and supporting your natural intuition with the science of data analysis will you be able to maintain and protect an ever-growing and increasingly complex infrastructure. We are not advocating replacing people with algorithms; we are advocating arming people with algorithms so that they can learn more and do a better job. The data contains information, and you can learn better with the information in the data than without it.

This book focuses on using real data—the types of data you have probably come across in your work. But rather than focus on huge discoveries in the data, this book focuses more on the process and less on the result. As a result of that decision, the use cases are intended to be exemplary and introductory rather than knock-your-socks-off cool. The goal here is to teach you new ways of looking at and learning from data. Therefore, the analysis is intended to be new ground in terms of technique, not necessarily in conclusion.

A Brief History of Learning from Data

One of the best ways of appreciating the power of statistical data analysis and visualization is to look back in history to a time when these methods were first put to use. The following cases provide a vivid picture of "before" versus "after," demonstrating the dramatic benefits of the then-new methods.

Nineteenth Century Data Analysis

Prior to the twentieth century, the use of data and statistics was still relatively undeveloped. Although great strides were made in the eighteenth century, much of the scientific research of the day used basic descriptive statistics as evidence for the validity of the hypothesis. The inability to draw clear conclusions from noisy data (and almost all real data is more or less noisy) made much of the scientific debates more about opinions of the data than the data itself. One such fierce debate[1] in the nineteenth century was between two medical professionals in which they debated (both with data) the cause of cholera, a bacterial infection that was often fatal.

The cholera outbreak in London in 1849 was especially brutal, claiming more than 14,000 lives in a single year. The cause of the illness was unknown at that time and two competing theories from two researchers emerged. Dr. William Farr, a well-respected and established epidemiologist, argued that cholera was caused by air pollution created by decomposing and unsanitary matter (officially called the ***miasma*** theory). Dr. John Snow, also a successful epidemiologist who was not as widely known as Farr, put forth the theory that cholera was spread by consuming water that was contaminated by a "special animal poison" (this was prior to the discovery of bacteria and germs). The two debated for years.

Farr published the "Report on the Mortality of Cholera in England 1848–49" in 1852, in which he included a table of data with eight possible explanatory variables collected from the 38 registration districts of London.

[1] And worthy of a bona fide Hollywood plot as well. See `http://snowthemovie.com/`

In the paper, Farr presented some relatively simple (by today's standards) statistics and established a relationship between the average elevation of the district and cholera deaths (lower areas had more deaths). Although there was also a relationship between cholera deaths and the source of drinking water (another one of the eight variables he gathered), he concluded that it was not nearly as significant as the elevation. Farr's theory had data and logic and was accepted by his peers. It was adopted as fact of the day.

Dr. John Snow was passionate and vocal about his disbelief in Farr's theory and relentless in proving his own. It's said he even collected data by going door to door during the cholera outbreak in the Soho district of 1854. It was from that outbreak and his collected data that he made his now famous map in Figure 1-1. The hand-drawn map of the Soho district included little tick marks at the addresses where cholera deaths were reported. Overlaying the location of water pumps where residents got their drinking water showed a rather obvious clustering around the water pump on Broad Street. With his map and his passionate pleas, the city did allow the pump handle to be removed and the epidemic in that region subsided. However, this wasn't enough to convince his critics. The cause of cholera was heavily debated even beyond John Snow's death in 1858.

The cholera debate included data and visualization techniques (long before computers), yet neither had been able to convince the opposition. The debate between Snow and Farr was re-examined in 2003 when statisticians in the UK evaluated the data Farr published in 1852 with modern methods. They found that the data Farr pointed to as proof of an airborne cause actually supported Snow's position. They concluded that if modern statistical methods were available to Farr, the data he collected would have changed his conclusion. The good news of course, is that these statistical methods are available today to you.

Twentieth Century Data Analysis

A few years before Farr and Snow debated cholera, an agricultural research station north of London at Rothamsted began conducting experiments on the effects of fertilizer on crop yield. They spent decades conducting experiments and collecting data on various aspects such as crop yield, soil measurements, and weather variables. Following a modern-day logging approach, they gathered the data and diligently stored it, but they were unable to extract the full value from it. In 1919 they hired a brilliant young statistician named Ronald Aylmer Fisher to pore through more than 70 years of data and help them understand it. Fisher quickly ran into a challenge with the data being confounded, and he found it difficult to isolate the effect of the fertilizer from other effects, such as weather or soil quality. This challenge would lead Fisher toward discoveries that would forever change not just the world of statistics, but almost every scientific field in the twentieth century.

What Fisher discovered (among many revolutionary contributions to statistics) is that if an experiment was designed correctly, the influence of various effects could not just be separated, but also could be measured and their influence calculated. With a properly designed experiment, he was able to isolate the effects of weather, soil quality, and other factors so he could compare the effects of various fertilizer mixtures. And this work was not limited to agriculture; the same techniques Fisher developed at Rothamsted are still used widely today in everything from medical trials to archaeology dig sites. Fisher's work, and the work of his peers, helped revolutionize science in the twentieth century. No longer could scientists simply collect and present their data as evidence of their claim as they had in the eighteenth century. They now had the tools to design robust experiments and the techniques to model how the variables affected their experiment and observations.

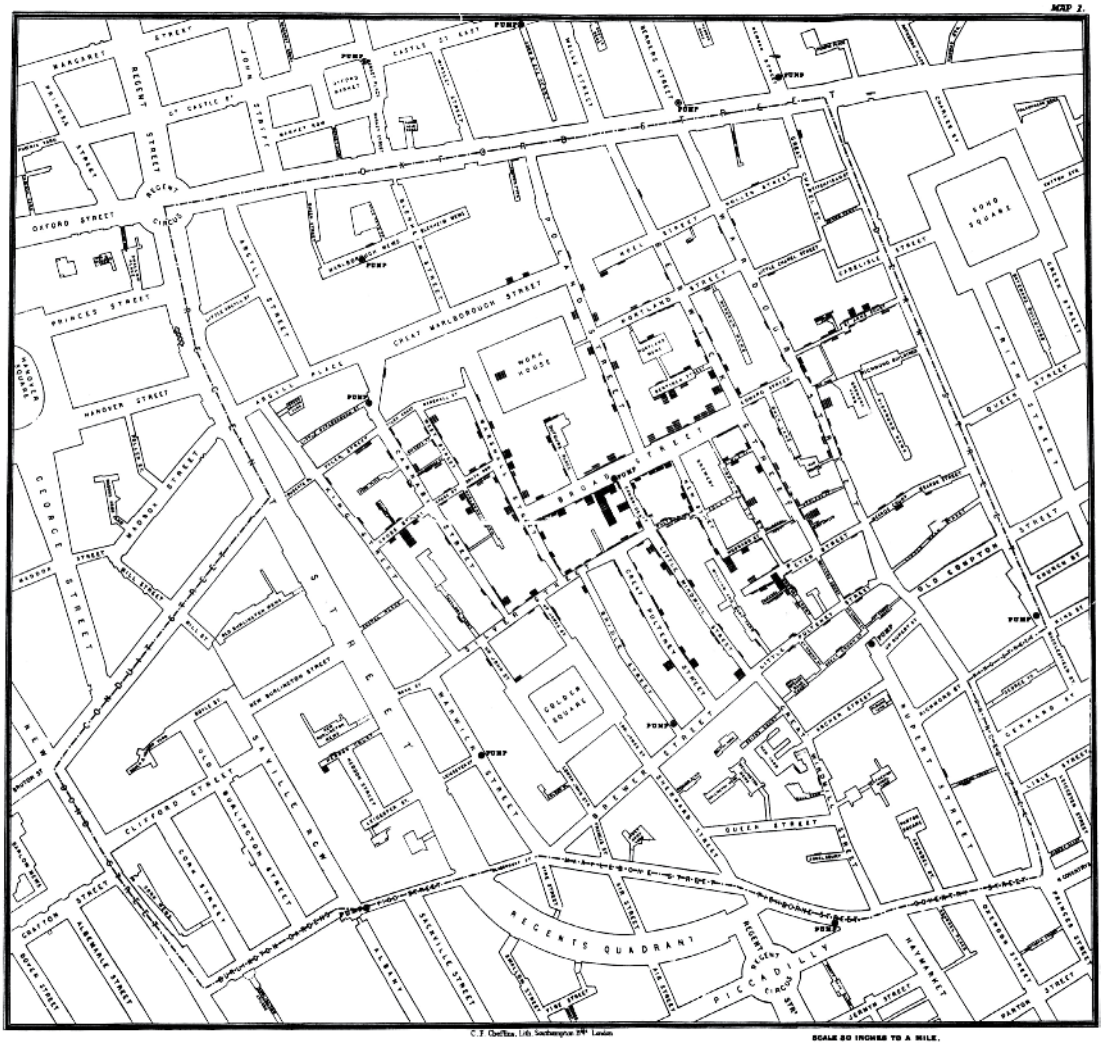

Figure 1-1 *Hand-drawn map of the areas affected by cholera*

At this point, the world of science included statistical models. Much of the statistical and science education focused on developing and testing these models and the assumptions behind them. Nearly every statistical problem started with the question—"What's the model?"—and ended with the model populated to allow description and even prediction using the model. This represented a huge leap forward and enabled research never before possible. If it weren't for computers, the world would probably still consider these techniques to be modern. But computers are ubiquitous and they have enabled a whole new approach to data analysis that was both impossible and unfathomable prior to their development.

Twenty-First Century Data Analysis

It's difficult to pull out any single person or event that captures where data analysis is today like Farr and Fisher captured the previous stages of data analysis. The first glimpse at what was on the horizon came

from John Tukey, who wrote in 1962 that data analysis should be thought of as different from statistics (although analysis leveraged statistics). He stated that data analysis must draw from science more than mathematics (can you see the term "data science" in there?). Tukey was not only an accomplished statistician, having contributed numerous procedures and techniques to the field, but he was also an early proponent of visualization techniques for the purpose of describing and exploring the data. You will come back to some of Tukey's work later in this chapter.

Let's jump ahead to a paper written in 2001 by Leo Breiman, a statistician who focused on machine learning algorithms (which are discussed in Chapter 9). In the paper he describes a new culture of data analysis that does not focus on defining a data model ***of nature*** but instead derives an algorithmic model ***from nature***. This new culture has evolved within computer science and engineering largely outside (or perhaps alongside) traditional statistics. New approaches are born from the practical problems created by the information age, which created large quantities of complex and noisy data. The revolutionary idea that Breiman outlined in this paper is that models should be judged on their predictive accuracy instead of validating the model with traditional statistical tests (which are not without value by the way).

At face value you may think of testing "predictive accuracy" by gathering data today and determining how it predicts the world of tomorrow, but that's not what the idea is about. The idea is about splitting the data of today into two data sets, using the first data set to generate (or "train") an algorithm and then validating (or "test") its predictive accuracy on the second data set. To increase the power of this approach, you can iterate through this process multiple times, splitting the data into various training and test sets, generating and validating as you go. This approach is not well suited to small data sets, but works remarkably well with modern data sets.

There are several main differences between data analysis in the modern information age and the agricultural fields of Rothamsted. First, there is a large difference in the available sample size. "Classic" statistical techniques were largely limited by what the computers of the day could handle ("computers" were the people hired to "compute" all day long). With generally smaller samples, generating a training and test was impractical. However, modern environments are recording hundreds of variables generated across thousands of systems. Large sample sizes are the norm, not the exception.

Second, for many environments and industries, a properly designed experiment is unlikely if not completely impossible. You cannot divide your networks into control and test groups, nor would you want to test the efficacy of a web application firewall by only protecting a portion of a critical application. One effect of these environmental limits is a much higher noise-to-signal ratio in the data. The techniques of machine learning (and the related field of data mining) have evolved with the challenges of modern data in mind.

Finally, knowledge of statistics is just one skill of many that contributes to successful data analysis in the twenty-first century. With that in mind, the next section spends some time looking at the various skills and attributes that support a good data analysis.

Gathering Data Analysis Skills

We know there is a natural allure to data science and everyone wants to achieve that sexy mystique surrounding security data analysis. Although we have focused on this concept of data analysis so far, it takes more than just analytic skills to create the mystique that everyone is seeking. You need to combine statistics and data analysis with visualization techniques, and then leverage the computing power and mix with a healthy dose of domain (information security) knowledge. All of this begins not with products or tools but with your own skills and abilities.

Before getting to the skills, there are a couple underlying personality traits we see in data analysts that we want to discuss: curiosity and communication. Working with data can at times be a bit like an archeological dig—spending hour after hour with small tools in the hope of uncovering even the tiniest of insights. So it is with data analysis—pearls of wisdom are nestled deep within data just waiting to be discovered and presented to an eagerly awaiting audience. It is only with that sense of wonder and curiosity that the hours spent cleaning and preparing data are not just tolerable, but somehow exciting and worth every moment. Because there is that moment, when you're able to turn a light on in an otherwise dark room, when you can describe some phenomenon or explain some pattern, when it all becomes worth it. That's what you're after. You are uncovering those tiny moments of enlightenment hidden in plain sight if you know where to look.

Once you turn that light on, you have to bring others into the room for the discovery; otherwise, you will have constructed a house that nobody lives in. It's not enough to point at your work and say, "see!" You have to step back and think of the best way to communicate your discovery. The complexity present in the systems and the analysis makes it difficult to convey the results in a way that everyone will understand what you have discovered. Often times it takes a combination of words, numbers, and pictures to communicate the data's insights. Even then, some people will take away nothing, and others will take away too much. But there is still a need to condense this complexity into a paragraph, table, or graphic.

Although we could spend an entire book creating an exhaustive list of skills needed to be a good security data scientist, this chapter covers the following skills/domains that a data scientist will benefit from knowing within information security:

- **Domain expertise**—Setting and maintaining a purpose to the analysis
- **Data management**—Being able to prepare, store, and maintain data
- **Programming**—The glue that connects data to analysis
- **Statistics**—To learn from the data
- **Visualization**—Communicating the results effectively

It might be easy to label any one of these skills as the most important, but in reality, the whole is greater than the sum of its parts. Each of these contributes a significant and important piece to the workings of security data science.

Domain Expertise

The fact that a data scientist needs domain expertise should go without saying and it may seem obvious, but data analysis is only meaningful when performed with a higher purpose in mind. It's your experience with information security that will guide the direction of the analysis, provide context to the data, and help apply meaning to the results. In other words, domain expertise is beneficial in the beginning, middle, and end of all your data analysis efforts.

And Why Expertise Shouldn't Get in the Way

We are probably preaching to the choir here. If you are reading this book, it is probably safe to assume that you have domain expertise and see value in moving toward a data-driven approach in information security. Therefore, rather than spend the effort discussing the benefits of domain expertise in data analysis, this

section covers some objections you might encounter as other domain experts (or skeptical leadership) are brought into the data analysis effort.

People are smarter than models. There are those who hold the opinion that people will always outperform algorithms (or statistics, or models) and there is some truth to this. Teaching a machine, for example, to catch a fly ball is remarkably challenging. As Kahneman and Klein point out in their 2009 paper titled *Conditions for Intuitive Expertise: a Failure to Disagree,* however, determining when people will outperform algorithms is heavily dependent on the environment of the task. If the environment is complex and feedback is delayed or ambiguous, algorithms will generally and relatively consistently outperform human judgment. So, the question then becomes, how complex is the security of the information systems and how clear is the feedback? When you make a change or add a security control, how much feedback do you receive on how well it is actually protecting the information asset?

The result is that information security occurs in a very complex environment, but that doesn't mean you put all your eggs in the algorithm basket. What it does mean is that you should have some healthy skepticism about any approach that relies purely on human judgment, and you should seek ways to augment and support that expertise. That's not to compare algorithms to human judgment. It's not wise to set up an either-or choice. You do, however, want to compare human judgment combined with algorithms and data analysis against human judgment alone. You do not want to remove the human element, but you should be skeptical of unsupported opinion. In a complex environment, it is the combination of human intuition and data analysis that will produce the best results and create the best opportunity for learning and securing the infrastructure.

It's just lying with statistics. This expresses a general distrust in statistics and data analysis, which are often abused and misused (and in some cases flat out made up) for the sake of serving some ulterior motive. In a way, this distrust is grounded in a collective knowledge of just how easy it is to social-engineer people. However, you are in a different situation since your motive is to learn from the data. You are sitting on mounds of data that hold information and patterns just waiting to be discovered. Not leveraging data analysis because statistics are misused is like not driving a car because they are sometimes used as get-away vehicles. You need to be comfortable with adding statistics to your information security toolkit.

This is not to say that data analysis is infallible. There may be times when the analysis provides the wrong answer, perhaps through poor data collection, under-trained analysts, a mistake in the process, or simply using Excel (couldn't resist). But what you should see is simply fewer mistakes when you apply the rigor of data analysis combined with your expertise. Again, the key is combining data analysis and expertise.

This ain't rocket science. This statement has two insinuations. First, it says that whatever the problem is you're trying to solve, you should be able to solve it with common sense. But this concern goes back to the first point, which is thinking that people outperform algorithms consistently and a group of people around a conference table looking at a complex environment can solve the (complex) problem without the need for data analysis. But as we discussed, you should pull a chair up to the conference table for the data analysis because you are generally better off with it than without it.

The second implication of the statement is that data analysis is too complicated and will cost too much (in time, money, or resources). This view is simply misinformed and the objection is more likely to be a concern about an uncomfortable change in practices than a concern about time spent with data analysis. Many of the tools are open source (if the organization is averse to open source, there are plenty of commercial solutions out there as well) and the only real commitment is in the time to learn some of the basic techniques and methods in this book. The actual analysis itself can be fairly quick, and with the right combination of tools and experience, it can be done in real time.

We don't have the data. An alternate form of this objection is saying that we don't have actuarial-quality data (which is more prevalent when you start talking about risk analysis). Data detractors argue that anything less than perfect data is worthless and prevents you from creating well-designed experiments. This statement is untrue and quite harmful. If you were to wait around for perfect data, you would always be waiting and many learning opportunities would be missed. More importantly and to the heart of this objection, you don't ***need*** perfect data. You just need methods to learn from the messy data you do have. As Douglas Hubbard wrote in 2010 in his book *How to Measure Anything,* "The fact is that we often have more data than we think, we need less data than we think, and getting more data through observation is simpler than we think." So, generally speaking, data for security analysis absolutely exists; often times it is just waiting to be collected. You can, with a few alterations, collect and accurately analyze even sketchy data. Modern data analysis methods have evolved to work with the noisy, incomplete, and imperfect data you have.

But we will fall off the edge of the world. There is one last point to consider and it's not so much an objection to data analysis, but an obstacle in data analysis. When you are seen as a domain expert, you are expected to provide answers with confidence. The conflict arises when confidence is confused with certainty. Data analysis requires just enough self-awareness and humility to create space for doubt in the things you think you know. Even though you may confidently state that passwords should be so many characters long with a certain amount of complexity, the reality is you just don't know where the balance is between usability and security. Confidence needs to be balanced with humility and the ability to update your beliefs based on new evidence. This obstacle in data analysis is not just limited to the primary analyst. Other domain experts involved in the analysis will have to come face to face with their own humility. Not everyone will want to hear that his or her world isn't flat.

Programming Skills

As much as we'd like to portray data science as a glamorous pursuit of truth and knowledge, as we've said, it can get a little messy. Okay, that's an understatement. Working with data is a great deal more uncertain and unkempt than people think and, unfortunately, the mess usually appears early on when you're attempting to collect and prepare the data. This is something that many classes in statistics never prepare their students for. Professors hand out rather nice and neat data sets ready to be imported into the analysis tool *du jour*. Once you leave the comfort of the classroom, you quickly realize that the world is a disorganized and chaotic place and data (and its subsequent analyses) are a reflection of that fact.

This is a cold, hard lesson in data science: Data comes to you in a wide range of formats, states, and quality. It may be embedded in unstructured or semi-structured log files. It may need to be scraped from a website. Or, in extreme cases, data may come in an overly complex and thoroughly frustrating format known as XML. Somehow, you must find a way to collect, coax, combine, and massage what you're given into a format that supports further analysis. Although this could be done with a lot of patience, a text editor, and judicious use of summer interns, the ability to whip together a script to do the work will provide more functionality, flexibility, and efficiency in the long run. Learning even basic programming skills opens up a whole range of possibilities when you're working with data. It frees you to accept multiple forms of data and manipulate it into whatever formats work best with the analysis software you have. Although there is certainly a large collection of handy data conversion tools available, they cannot anticipate or handle everything you will come across. To be truly effective while working with data, you need to adapt to the data in your world, not vice versa.

AES-256-Bit Keys Are Twice as Good as AES-128, Right?

One natural assumption about AES-256-bit keys is that because they are twice as long as AES-128-bit keys, they are twice as secure. We've been around information security people when they force a project to use 256-bit keys because they are "twice as good." Well, let's look into the math. First, you are talking about bits here, and although 256 bits is twice as many bits as 128, 256-bit keys actually have 2^{128} ***times*** more keys. Break out your slide rules and work through an exercise to try to answer a simple question: If you had access to the world's fastest super-computer, how many 128-bit keys could you crack?

The world's fastest super computer (at the time of this writing) is the ***Tianhe-2*** in China, which does around 34 petaflops (34×10^{15} floating point operations) per second. If you assume it takes one operation to generate a key and one operation to test it (this is an absurd and conservative assumption), you can test an amazing 17×10^{15} keys per second. But a 128-bit key has 3.4×10^{38} possibilities, which means after a full year of cracking 128-bit keys, you will have exhausted 1.6×10^{-13} percent of the key space. Even if you run the super-computer for 1,000 years, you will only have searched 0.0000000000016 percent of all the possible keys (and spent a fortune on electricity).

To put this simply, ***the probability of brute-force cracking a 128-bit key is already so infinitesimally small that you could easily round off that probability to zero***. But let's be professional here and say, "Moving from a 128-bit key to a 256 is moving the probability from really-super-duper-infinitesimally-small to really-super-duper-infinitesimally-small x 2^{128}."

Any modern language will support basic data manipulation tasks, but scripting languages such as Python and R appear to be used slightly more often in data analysis than their compiled counterparts (Java and C). However, the programming language is somewhat irrelevant. The end results (and a happy analyst) are more important than picking any "best" language. Whatever gets the job done with the least amount of effort is the best language to use. We generally flip between Python (pandas) and R for cleaning and converting data (or perhaps some Perl if we're feeling nostalgic) and then R or pandas for the analysis and visualization. Learning web-centric languages like HTML, CSS, and JavaScript will help create interactive visualizations for the web, as you'll see in Chapter 11, but web languages are not typically involved in the preparation and analysis of data.

There is a tool worth mentioning in this section—the "gateway tool" between a text editor and programming—known as the ***spreadsheet*** (such as Microsoft Excel or OpenOffice Calc). Spreadsheets allow non-programmers to do some amazing things and get some quick and accessible results. Although spreadsheets have their own sets of challenges and drawbacks, they also have some benefits. If the data is not too large or complex and the task is not deciding the future of the world economy (see the following sidebar), Excel may be the best tool for the job. We strongly suggest seeing Excel as a temporary solution. It does well at quick one-shot tasks. But if you have a repeating analytic task or model that is used repeatedly, it's best to move to some type of structured programming language.

As a cleaning tool, spreadsheets seem like a very good solution at first (especially for those who have developed some skill with them). But spreadsheets are event-driven, meaning they work through clicking, typing, and dragging. If you want to apply a conversion to a row of data, you have to click to select the row and apply a conversion. This works for small data sets or quick tasks, but trust us, you will (more often than

you think) have to go back to the source data and re-clean it. Another day of log files needs to be processed, or you realize you should have pulled another relationship from the source data, or (gasp) you identify an error in the cleaning process. Something, somewhere, and probably more than once, will cause you to go back to the source and repeat the data cleaning and conversion. Leveraging a spreadsheet means a lot more clicking. Writing a script, on the other hand, enables an easy, flexible, and consistent execution of the cleaning process each time it runs.

The Limits of Spreadsheets

On January 16th, 2013, J.P. Morgan issued a report to shareholders titled "Report of JPMorgan Chase & Co. Management Task Force Regarding 2012 CIO Losses" (full citation in Appendix B) in which they investigate the loss of $6 billion in trades. They perform a detailed examination of the breakdown and describe the spreadsheet as a contributory factor. "During the review process, additional operational issues became apparent. For example, the model operated through a series of Excel spreadsheets, which had to be completed manually, by a process of copying and pasting data from one spreadsheet to another." They uncovered a huge challenge with spreadsheets, which is the consistency and integrity of the computations made in the data. "Data were uploaded manually without sufficient quality control. Spreadsheet-based calculations were conducted with insufficient controls and frequent formula and code changes were made." They continue on and label the Excel-based model as "error prone" and "not easily scalable." As with any complex system, catastrophe requires multiple failures.[2] We cannot point to their use of an "error-prone" spreadsheet as the primary cause, but certainly it appears to have contributed in the loss of $6 billion.

[2] See Richard Cook's "How Complex Systems Fail" for a brief and wonderful discussion of this topic:
`http://www.ctlab.org/documents/How%20Complex%20Systems%20Fail.pdf`

After the data is ready for analysis, you can continue to benefit from understanding how to program. Many of the languages mentioned here have robust data analysis features built into (or onto) them. For example, statisticians developed the R language specifically for the purpose of performing data analysis. Python—with the addition of packages like NumPy, SciPy, and pandas—offers a rich and comparable data analysis environment. But, preparing and analyzing the data is not enough. You also need to communicate your results, and one of the most effective methods for that is data visualization (covered in several chapters of this book). Again, Excel can produce graphics. With judicial modification of the default settings, you can get good visualization with Excel. However, in our opinion, flexibility and detail in data visualization are best achieved through programming. Both Python and R have some feature-rich packages for generating and exporting data visualization. In many cases, however, you can combine all these steps and functions in the same script. You can write one script to grab the source data, manipulate/clean it, run the analysis on it, and then visualize the results.

Data Management

If there is one skill you can hold off on learning, it's data management, but you can put it off only for a while. Within information security (as well as most other disciplines), your data can quickly multiply. If you

don't learn to manage it, the strain of ever-expanding data will take its toll on efficiency and effectiveness. As mentioned, you can leverage spreadsheets for the simple analyses. You will quickly outgrow that stage and should be resolved to expanding your repertoire to programming languages and simple formats like comma-separated value (CSV) files. At this point, you may see some benefits by moving your data into a database, but it still may not be necessary.

As the data repository grows, you reach a tipping point, either through the complexity of the data or the volume of data. Moving to a more robust data management solution becomes inevitable. There is a misconception that the large relational databases of yesteryear are reserved for the biggest projects, and that is not a helpful mindset. Many of the database systems discussed in Chapter 8 can be installed on a desktop and make the analysis more efficient and scalable. Once your data management skills become more natural, such skill can benefit even the smallest projects. We've installed a local database and imported the data even for some smaller one-time projects.

When discussing data management skills, we naturally focus on databases. You want to have enough knowledge to install a relational or NoSQL database to dump the data in and leverage it for analysis. However, data management is more than databases. Data management is also about managing the quality and integrity of the data. You want to be sure the data you are working with isn't inadvertently modified or corrupted. It doesn't hurt to have some checks that keep an eye on data quality and integrity, especially over long-term data analysis efforts (metrics). It's like the concept of unit tests in software development where the smallest piece of testable code in an application is isolated from the larger body of code and checked to determine whether it behaves exactly as expected. You may want to automate some data integrity checking after any new import or conversion, especially when the data analysis has sufficient efficacy to be performed regularly and used as a metric or control.

Finally, we work in information security, and we'd be negligent if we didn't talk about the security of the data for a bit here. Take a step back for some context first. There seems to be a pattern repeating: Some passionate need drives a handful of geniuses to work their tails off to produce an elegant solution, but the security of their system is not their primary concern; meeting the functional need is. As an example, when the UNIX platform was first developed it was intended to be a shared (but closed) platform for multiple users who use the platform for programs they would write. As a result, most of the authentication and permissions were constructed to protect the system from unintentional errors in their programs, and not from malicious users.[3] The point here is that "young" technology typically places an emphasis on functionality over security.

With the fast-paced and passionate push of the current data revolution, we are definitely seeing more emphasis on functionality and less on security. New data management platforms such as Hadoop and NoSQL environments were designed to solve a data problem and were not designed (initially) with many of the security policies or compliance requirements of most enterprise networks (though they are quickly learning). The result is a distributed computing platform with some difficult security challenges. The authentication and security features are far better than the early days of UNIX; they typically do not compare to the security and features of the more established relational databases. We won't focus too much on this point, but whatever data management platform is chosen, don't assume the security is built in.

---

[3] For an example of the focus on functionality and preventing error over stopping misuse, early authentication systems would store the user passwords in a clear text file. See Morris and Thompson, 1979 (full reference in Appendix B) for a discussion.

Statistics

Perhaps we are a little biased here, but picking up some statistics skills will improve almost every aspect of your life. Not only will it change the way to see and learn from the world around you, but it will also make you more interesting and probably even a bit more attractive to those around you. Seriously, though, statistics (we are discussing it as a single skill here) is a very broad topic and quite a deep well to drink from. We use the term to describe the varied collection of techniques and methods that have evolved (and continue to evolve) to attempt to learn from data. These skills include the classic statistical approaches as well as newer techniques like data mining and machine learning. Luckily, you can learn from the successes and mistakes of the generations of rather brilliant people who have worked with data very similar to ours, even if their calculations were performed with pencil and paper versus silicon circuits. Regardless of your personal belief in the utility of statistics and data analysis, when it comes to information security, there is a vast amount of evidence showing its significant influence and benefit to almost every other field of science.

Aside from the obvious "learning from data" approach, there are a few perhaps more subtle reasons to focus on improving your statistics skills:

- **Even though data never lies, it is far too easy to be tricked by it**—As heuristic beings, we are capable of pulling out patterns and meaning from the world around us. The ability to see subtle connections and patterns is usually helpful, and people use that skill on a daily basis. However, that skill can also mislead you, and you may think you see patterns and connections when none exist. A good understanding of statistics can raise awareness of this, and its tactics can help minimize incorrect conclusions.
- **Even though we just said that data never lies, the way it's generated and collected can create deceptive conclusions**—Consider that asking for the opinions of those around us may mistakenly confirm our own opinions, because we naturally surround ourselves with like-minded people. Data itself may not be deceptive, but it's quite easy to think the data means something it does not, as in the story of the 1936 election polling (see the following sidebar).

Statistics is not just a collection of tools; it is a collection of toolboxes each with their own set of tools. You can begin with descriptive statistics, which attempt to simplify the data into numbers that describe aspects of the data. For example, you can calculate the center of the data by calculating the mean, mode, or median; you can describe how spread out the data is with the standard deviation; you can explain the symmetry of the data with skew; and you can describe the width of peak with the kurtosis. However, any time you simplify the data, you lose some level of detail and this is where visualization can serve you well. With visualizations, you create a single representation, or message, that can contain and communicate every data point, without simplification. Think of this type of visualization as being a "descriptive visualization" since it is doing nothing more than simply describing the data to its viewers.

Aside from the challenge of oversimplifying, descriptive statistics is also limited to describing only the data that you collect. It is not correct to simply scan a few systems, calculate the mean number of vulnerabilities, and announce that the statistic describes all the systems in the environment. Inferential statistics helps you go beyond just describing the observations and enables you to make statements about a larger population given a smaller representative sample from that population. The key word there is "representative." Statistics teaches you about the "design of experiments" (thanks to Fisher and his peers) and this will help you gather data so that you reduce the probability of being misled by it. You want to have confidence that the samples you collect are representative of the whole. That lesson has been learned many times in the past by a good number of people.

When Data Deceives

The magazine ***Literary Digest*** ran a large public opinion poll in an attempt to predict the 1936 presidential race. They gathered names from a variety of sources, including the telephone directory, club memberships, and magazine subscriptions. They ended up with more than 2 million responses and predicted a clear winner: Alfred Landon (for those not up on their American history, the Democratic candidate, Theodore Roosevelt, won that election, carrying 46 states). The problem with the ***Literary Digest*** poll began long before a single response was collected or counted. Their trouble began with where they went looking for the data. Remember the year was 1936 and the great depression in the United States hadn't let up yet. Yet, they polled people with phones, club memberships, and magazine subscriptions. They systematically polled the middle and upper class, which generally leaned toward Landon, and arrived at an answer that was mathematically correct and yet completely wrong.

The data did not lie. If they wanted to know which presidential candidate would get the most votes among Americans with a phone, club membership, or magazine subscription, the data told an accurate story. However, they weren't looking for that story. They wanted to know about all registered voters in the United States, but through their selection of sources they introduced bias into their sample and drew meaning from the data that simply did not exist.

The fact that they had an unprecedented 2 million responses did not help improve the accuracy of their poll. Gathering more data with the same systemic flaw just generates a larger sample with the bias. To drive that point home, in the same 1936 election, a young man named George Gallup had gathered a relatively small sample of just 50,000 voters but he applied a much more representative sampling method and correctly predicted Franklin Roosevelt as the winner of the 1936 elections. The ***Literary Digest*** closed its doors a few years later, but Gallup, Inc. is now an international organization, still conducting surveys and gathering data.

You should always approach statistics with a healthy degree of respect and humility. As you slide more and more into the depths of applied mathematics, you'll realize how easy it is to find meaning where none exists (technically called a ***type I error***). But what is more important to understand here is that this error can occur with or without data. Even before you fill a single cell in an Excel spreadsheet, you can make this mistake. The best tools in the toolbox are designed to limit the chance of these types of errors, but statistics alone is not enough. You need the combination of experience and data to decrease the chance of being misled. Errors can and will occur even with this combination, but you can reduce the frequency of these errors by applying the rigor and methods within statistics. Such rigor will place you in a much better position to learn from mistakes when they do occur.

Having built up the application of statistics on a pedestal, we should point out that you can learn a lot from data without advanced statistical techniques. Recall the "descriptive visualization" mentioned previously. Take some time to look around at many of visualizations out there; they are generally not built from statistical models, but describe some set of data and show the relationships therein. Snow's map of the areas around the water pump on Broad Street in Figure 1-1 did not involve logistic regression or machine learning; this map was just a visual description of the relationship between address and deaths. There is no doubt that you can improve your ability to secure your information assets with simple statistical methods and descriptive visualizations. All it takes is the patience to ask a question, gather the evidence, make sense of it, and communicate it to others.

Visualization (a.k.a. Communication)

The final skill is ***visualization***, but really it is about communication. There are multiple ways to classify the types of visualizations out there, but for this discussion we want to talk about two general types of visualization, which are separated by who you want to read and interpret the visualization. The distinction we make here is quite simple: 1) visualizing for ourselves, or 2) everyone else.

For example, Figure 1-2 shows four common plots, which are automatically generated by R's `lm()` function (for linear regression) and they are used to diagnose the fit of a linear regression model (which you'll run in Chapter 5). Let's face it; these plots are quite ugly and confusing unless you've learned how to read them. We would not include these in our next presentation to the Board of Directors. This type of visualization serves to provide information to the analyst while working with the data, or in this case about a data model.

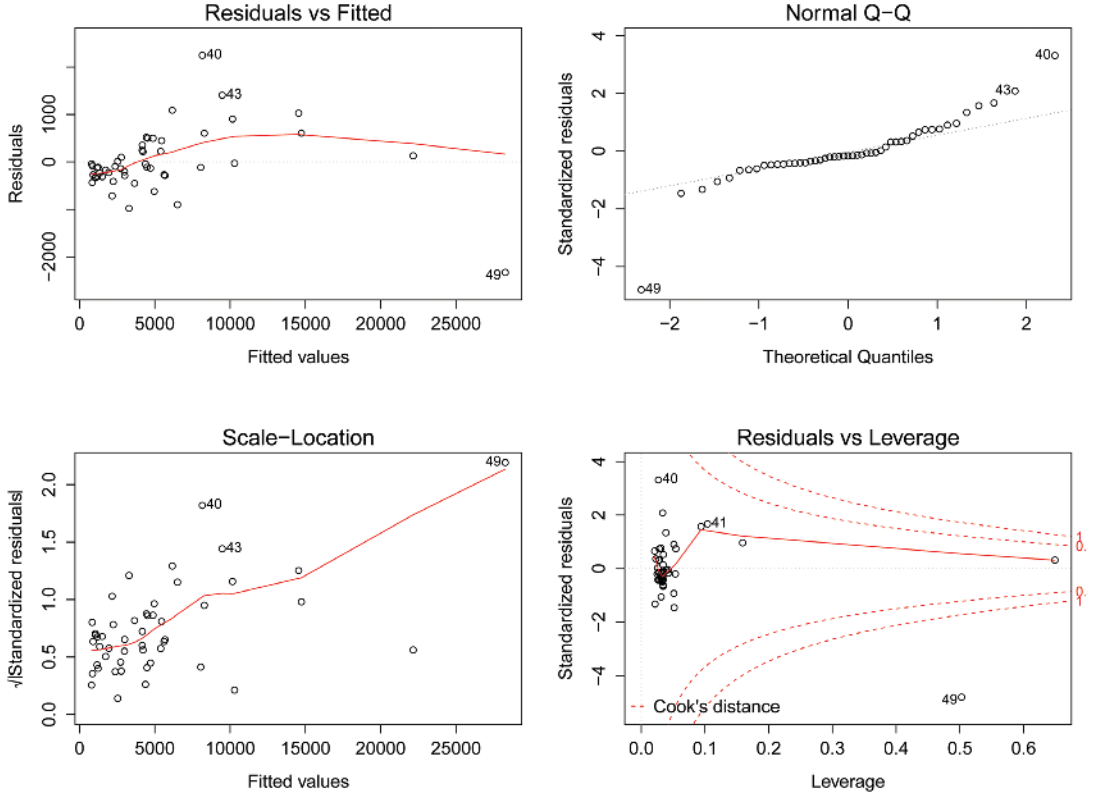

FIGURE 1-2 *Diagnostic plots for regression model of bot infections*

These graphs are generated as a way to understand certain relationships and attributes of the model. They communicate from the data to the analyst and are used to visually inspect for anomalies, strength of relationships, or other aspects of the data for the purpose of understanding it better. Very little effort is spent on making these attractive or presentable since they are part of the analysis, not the result.

The other type of visualization exists to communicate from the analyst to others and serves to explain the story (or the lack of a story) the analyst uncovered in the data. These are typically intended to be attractive and carry a clear message, as it is a communication tool for non-analysts. Figure 1-3 (which you'll learn to generate in Chapter 5) is derived from the same data as Figure 1-2 but is intended for a completely different audience. Therefore, it is cleaner and you can pull a message for each of the 48 continental states from this one picture.

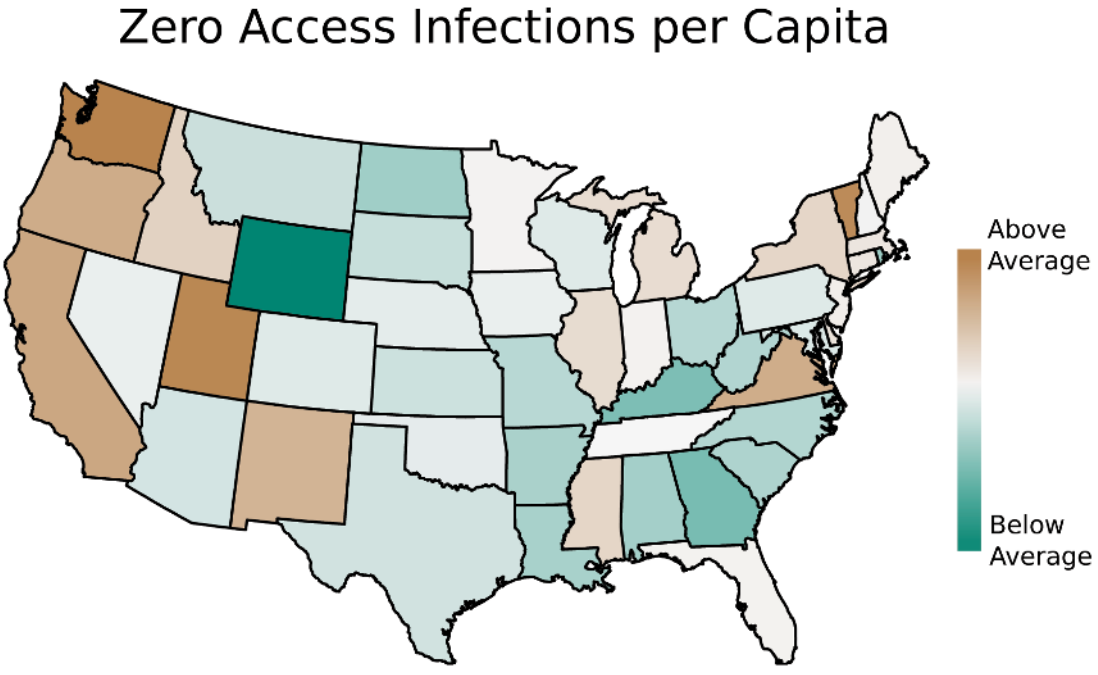

Figure 1-3 *Visualization for communicating density of ZeroAccess bot infections*

Combining the Skills

You need some combination of skills covered in this chapter in order to make the analysis run smoother and improve what you can learn from the data. Although we may have portrayed these skills as belonging to a single person, that is not required. As the data grow and the demands for analysis become more embedded into the culture, spreading the load among multiple individuals will help lighten the load. Moreover, if you are just beginning to build your security data science team, you may be setting yourself up for an impossible task if you try to find even one individual with all these skills. Take the time to talk through each of these points with candidates to ensure there is at least some element of each of the skills discussed here.

Centering on a Question

While we consider data analysis to be quite fun, it is never performed for its own sake. Data analysis is always performed within a larger context and understanding that context is the key to successful data analysis. Losing sight of that context is like running a race without paying attention to where the finish line is. You want to have a good concept of what you're trying to learn from the data. Therefore, every good data analysis project begins by setting a goal and creating one or more ***research questions***. Perhaps you have come across a visualization or research and thought, "Yeah, but so what?" That reaction is probably caused by the lack of a well-prepared research question in the analysis. Remember, the purpose of data analysis is to learn from the environment; learning can be done with or without data (with varying degrees of success). Creating and following a good research question is a component of ***good learning***, not just of good data analysis. Without a well-formed question guiding the analysis, you may waste time and energy seeking convenient answers in the data, or worse, ***you may end up answering a question that nobody was asking in the first place***.

For example, Figure 1-4 shows the amount and categories of spam blocked at an organization during a given month. Thanks to the logs generated by an email filtering system, it is entirely possible to collect and show this information. However, the questions this data answers (and whatever subsequent actions it may drive) are of little interest to the typical organization. It's hard to imagine someone looking at this graphic and thinking, "Let's understand why travel spam was up in December." Outcomes like those shown in Figure 1-4 are the result of poor question selection or skipping a question altogether—data analysis for the sake of analyzing data, which does not help to inform anyone about the environment in any meaningful way.

A good research question around spam might be, "How much time do employees spend on spam that is not blocked by the spam filter?" Just counting how much spam is blocked has little value since it will have no contextual meaning (nobody can internalize the effective difference between 1,000 and 5,000 spam emails). What you want to know is the impact spam has on employee productivity. Although "productivity" may be a challenge to measure directly, you can flip that around and just assume it is impossible to be productive when employees are reading and deleting spam. Therefore, what you really want to measure is the time employees spend dealing with unfiltered spam.

Now that you've framed the question like this, it's clear that you can't look to the spam filter logs to answer this spam-related question. You really don't care that thousands of emails were blocked at the perimeter or even what proportion of spam is blocked. With a research question in hand, you now know to collect a measurement of employee time. Perhaps you can look for logs from the email clients of events when users select the "mark as spam" option. Or perhaps, it's important enough to warrant running a short survey in which you select a sample of users and ask them to record the amount of spam and time spent going through it for some limited period of time. Either way, the context and purpose of the analysis is being set by the research question, not by the availability of data.

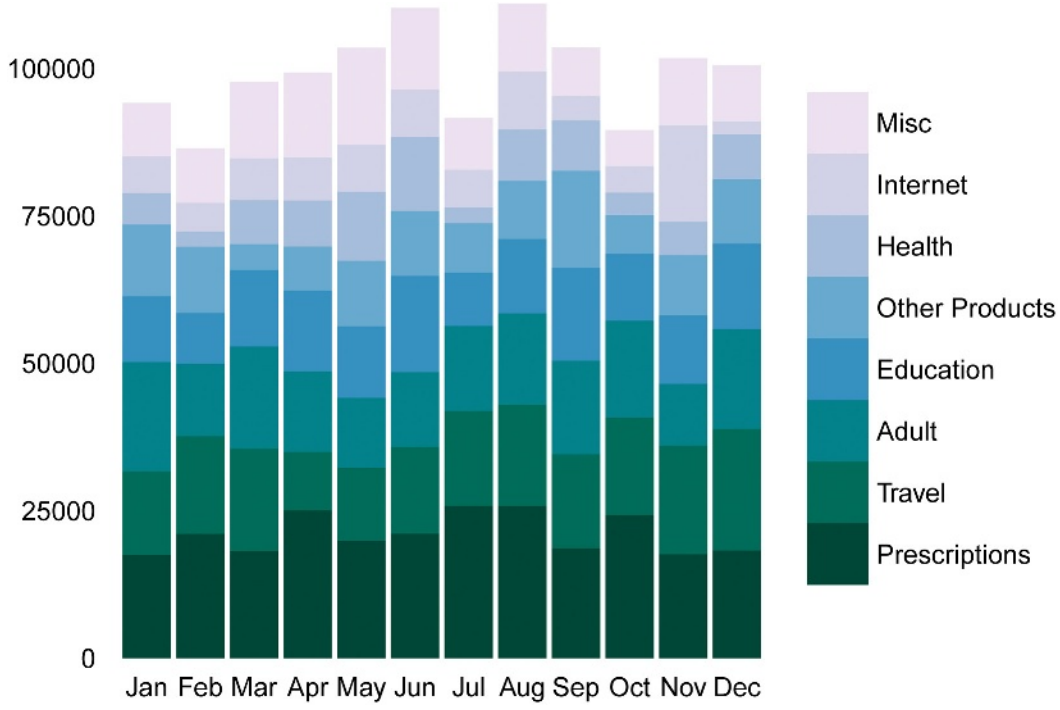

Figure 1-4 *Amount of spam by category—the result of a poor research question*

Creating a Good Research Question

Creating a good research question is relatively straightforward but requires a bit of practice, critical thinking, and discipline. Most research questions will serve as pivot points for a decision or action (or inaction). Knowing the context of the result may also help determine what to collect. Going back to the spam example, maybe you learn there is some tolerance for wasted time. If so, maybe you don't need to how much time is wasted, but just whether the time spent dealing with spam is simply above or below that tolerance. Planning the analysis with that information could change how data is sought or simplify data storage and analysis.

You usually begin with some topic already in mind. Perhaps you are measuring the possible benefit from a technical change or you are trying to protect a specific asset or data type, or simply trying to increase your visibility into a network segment. Even if you just have a general sense of direction, you can begin by coming up with a series of questions or things you'd like to know about it. Once you have a good list of questions, you can whittle those down to one or just a few related questions. Now the fun really begins—you have to make those questions objective.

Consider this simple example. The Human Resources department submits a proposal to post a searchable lunch menu from the company's cafeteria to the Internet. Although this may raise all sorts of questions around controls, processes, and procedures, suppose the core security-oriented decision of the proposal is limited to either allowing authentication with the corporate username and password, or investing in a more expensive two-factor authentication mechanism. You may brainstorm a question like "How much risk does single factor authentication represent?" Or perhaps, "How effective is two-factor authentication?" These types of questions are really nice and squishy for the initial phase of forming a research question, but not well suited to serious analysis. You would struggle to collect evidence of "risk" or "effectiveness" in these questions. So, you must transform them to be more specific and measurable as an approach to inform the decisions or actions in context. Perhaps you start by asking how many services require single-factor versus dual-factor authentication. You might also like to know how many of those services have been attacked, and with what success, and so on. Perhaps you have access to a honey pot and can research and create a profile of Internet-based brute force attempts. Perhaps you can look at the corporate instance of Microsoft Outlook Web Access and create a profile of authentication-based attacks on that asset. These are all good questions that are very answerable with data analysis. They can produce outcomes that can help support a decision.

Exploratory Data Analysis

Now that we've explained how a good data analysis should begin, we want to talk about how things will generally occur in the real world. It'd be great to start each day with a hot, caffeinated beverage, a clear research question, and a bucket of clean data, but in reality you'll usually have to settle for just the hot, caffeinated beverage. Often times, you do start off with data and a vague question like, "Is there anything useful in this data?" This brings us back to John Tukey (remember him from earlier in this chapter?). He pioneered a process he called ***exploratory data analysis***, or EDA. It's the process of walking around barefoot in the data, perhaps even rolling around a bit in it. You do this to learn about the variables in the data, their significance, and their relationships to other variables. Tukey developed a whole range of techniques to increase your visibility into and understanding of the data, including the elegantly simple ***stem and leaf plot***, the ***five-number summary***, and the helpful ***box plot*** diagram. Each of these techniques is explained or used later in this book.

Once you get comfortable with the data, you'll naturally start to ask some question of it. However, and this is important, you always want to circle back and form a proper research question. As Tukey said in his 1977 book, "Exploratory data analysis can never be the whole story." He refers to EDA as the foundation stone and the first step in data analysis. He also said that, "Exploratory data analysis is an attitude, a state of flexibility, a willingness to look for those things that we believe are not there, as well as those we believe to be there." With that in mind, most of the use cases in this book use exploratory analysis. We will take an iterative approach, and you'll learn as you walk around in the data. In the end though, you need to remember that data analysis is performed to find an answer to a question that's worthy of asking.

Summary

The cyber world is just too large, has too many components, and has grown far too complex to simply rely on intuition. Generations of people before us have paved the way; and with a mixture of domain expertise, programing experience and statistics combined with data management and visualization skills, we can improve on our ability to learn from this complex environment through the data it produces.

In the next chapter we will walk you through setting up your data analysis environment, and then proceed into Chapter 3, where you will be guided through a gentle introduction to data analysis techniques.

Recommended Reading

The following are some recommended readings that can further your understanding on some of the topics we touch on in this chapter. For full information on these recommendations and for the sources we cite in the chapter, please see Appendix B.

"Conditions for Intuitive Expertise: A Failure to Disagree" by Daniel Kahneman and Gary Klein—This dense article covers a lot of ground but gets at the heart of when and why you should look for help in complex environments and when your expertise is enough. The references in this paper also provide a good jumping point to answer questions about how people learn.

"How Complex Systems Fail" by Richard Cook—If you are wondering whether or not you are dealing with complexity, this short and brilliant paper looks at qualities of complex systems and how they fail.

***Naked Statistics: Stripping the Dread from Data* by Charles Wheelan**—This is a great introductory book to statistical concepts and approaches, written in an easy-to-consume style and written so that the math is not required (but it is included).