Chapter 2: Beginning with a Question

“To most people statistics means plugging numbers into an advanced calculator that spits out values, without much thought involved. Those people don’t work with data.” (“a life in statistics: Nathan Yau”, significance magazine).

It is natural to assume that the first step in data analysis is getting the data to analyze but data analysis is never performed for its own sake. It is always performed within a larger context and understanding that context is the key to a successful data analysis. But before we get into what makes data analysis good, we want to discuss what skills contribute to a good data analysis. We know there is a natural allure to data science and everyone wants to achieve that sexy mystique (oh we know) surrounding data analysts, so we will begin this chapter by talking about how we achieve that mystique. There are multiple disciplines and skills that come together in the science of data analysis and we will walk through each one.

The two most important personality traits of a data analyst are curiosity and communication. Working with data can at times be a bit like how we imagine archeology: spending hour after hour with small tools in the hope of uncovering even the tiniest of insights in the dirt. So it is with data analysis: pearls of wisdom are nestled deep within a mound of 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 worth every moment.

Once those pearls are discovered and cleaned up, they must be communicated to others. Make no mistake, the complexity of both our environment and analysis is difficult to convey to others. Often times it takes a combination of words, numbers and pictures to communicate the insights in data and that’s the personality trait that’s helpful, being able to condense complexity into a paragraph, table and/or graphic. Curiosity is especially important because doing data analysis well requires gaining and maintaining a broad mixture of skills. It takes some dedication and motivation to develop the other skills needed and with a healthy dose of curiosity the skills seem to develop effortlessly in an individual.

While it may be difficult to create an exhaustive list of skills to be a good data analyst[[1]](#footnote-1), we are going to cover the following skills/domains that data analysts benefit from knowing within information security: **domain expertise** (setting and maintaining purpose to the analysis), **data management** (being able to prepare, store and access data), **programming** (the glue that connects data to the analysis), **statistics** (to avoid being lied to by data) and **visualization** (communicating the analysis graphically).

Not all of these skills are required to be present in a single individual. While smaller shops may seek a single individual to cover these areas, as the work load increases it is entirely possible (and a lot more feasible) to spread these tasks across several individuals (or even individual teams). Wherever these skills come from, when they come together we create an environment where data analysis has its best chance at success. It’d be easy for us to label each one of these skills as the most important, but in reality, the whole is greater than the sum of its parts. Each of these contribute a significant piece of the puzzle.

Domain Expertise

Saying that a data scientist needs domain expertise should go without saying. It may seem obvious when we lay it out like this, but this cannot be talked about enough and there are some important points to discuss about the benefits and pitfalls within domain expertise. On one hand, domain expertise sets the context and purpose for data analysis and can help prevent or identify spurious results in the analysis. On the other hand, domain experts may push away data science, either as a perceived threat to their authority or because it may challenge long-held beliefs, viewed as conventional wisdom.

Creating a good research question, as we’ll see in the next section, is one of if not the most critical steps in any data-driven analysis. A good research question in information security requires an understanding of how systems work (and how they can break), how the attackers think and act and the tools they’ll be using and deploying. Attempting to develop the research question without good knowledge of the environment may lead to superficial and/or irrelevant findings. Therefore knowledge of information security is very important when forming that research question. But the research question also needs to be asked in a way that the data can answer. Asking questions about hypothetical constructs like “how much risk…” do not lend themselves well to data-driven analysis (but we’ll discuss research questions later in this chapter). The point here is that just domain expertise without knowledge of good data collection practices can result in chasing ghosts and doom the analysis before it even starts. Therefore, good data analysis requires knowledge of both information security and statistics so it has a valid purpose and is set up to make an impact.

Once the question is set and the data is gathered, domain expertise also brings context to data and analysis. Without an understanding of what the data represents and the relationships within them, problems can arise during the data munging and the analysis. At best, the lack of context may just waste time, at worst it could produce spurious results that would fail even a basic sniff test. Regardless, if the domain expert and data scientist is not the same person, countless back and forth communications (and sometimes some frustrations) can be generated here as the data scientist learns from the expert about the characteristics within the data that affect the analysis and the outcomes.

Finally, an understanding of the domain is essential in interpreting and assigning significance to the results. As we’ll see later in the book, statistics can only show correlation and can never prove causation. Causal relationships can only be established by an expert in the environment, and then only based on their understanding, observations and experience.

While we’d struggle to understand data without expertise in information security, it’s that same expertise that can present a challenge to good data analysis. Aside from any political or power struggle that may be come up, some people may see data science as silly threat, trying to replace their work and experience with models and formulas. This objection is not only misplaced, but also counter-productive. Statistics and all the related fields only have value in context and then as a supporting role. We need to mentally split expertise within an environment (like how we protect information systems) from the expertise of data analysis. We should not approach a statistician to present on identity management any more then we should approach an information security practitioner to collect and analyze data.

There is one more pitfall we have to discuss and it’s a bit of a challenge to write about knowing that the target audience for this book is the experts in the field of information security. There are times when expertise is built on some assumptions that appear logical, but later prove to be false upon closer inspection. Which is just a fancy way to say that experts can be wrong and even the possibility of that can cause friction. For example, we often hear that passwords should always be of a certain length and pull from multiple characters sets, but is this good advice? Florêncio and Herley from Microsoft Research collected data from 75 different websites and concluded a restrictive password policy “causes considerable inconvenience for negligible security improvement.” (http://research.microsoft.com/pubs/132623/WhereDoSecurityPoliciesComeFrom.pdf) As a data scientist, we must not only have awareness that these assumptions exist and are sometimes deep-rooted in the culture but that we must also have the confidence to challenge the conventional wisdom that has been built up in the industry over the years.

Programming Skills

As much as we’d like to portray data science as a glamorous pursuit of truth and knowledge, honestly it can get a little messy. Okay, that’s an understatement. Working with data is a lot more uncertain and messy then people think and unfortunately the mess usually appears early on when we collect and prepare the data. This is something that many classes in statistics never prepare their students for. The professors hand out rather nice and neat data sets ready to be imported into the analysis tool du jour. But once we leave the comfort of the classroom we quickly realize that the world is a disorganized and messy place and data are a reflection of the world.

This is the first cold-hard lesson in data science: data comes to us in a wide range of formats, states and overall quality causing us to spend a healthy portion of our time cleaning and preparing the data for analysis. This is where the ability to whip together a script comes in very handy. Learning even basic programing skills opens up a whole range of possibilities when we are working with data. It frees us to accept any form of data and munge it into whatever format we (or the analysis software) would want. There is certainly a large collection of data conversion tools available that can come in handy, but they certainly cannot anticipate or handle everything we will come across. To be really effective while working with data, we need to adapt to our data, not vice versa.

Our data may be embedded in semi-structured log files or maybe it needs to be scraped from a website, or in really bad cases, data comes in an overly complex and thoroughly confusing format known as an XML document. Somehow this data must be collected, coaxed and massaged into a format that supports further analysis. While this could be done with a lot of patience and a text editor, learning a programming language is way more efficient in the long run.

Most every modern language will support basic data manipulation tasks, but the scripting languages (python, R, perl) are used more often in data analysis then their compiled counter parts (Java, C). The programming language is somewhat irrelevant, as the results and a happy analyst matter way more then picking a “best” language. Whatever gets the job done is the best language to use. We prefer using Python for the cleaning and converting data (or perhaps some Perl if we’re feeling nostalgic, or want some unique package) and then R and/or Python for the analysis and visualization. Learning web-centric languages like HTML, CSS and JavaScript will help create interactive visualizations for the web, as we’ll see in chapter 7, but web languages are not typically involved in the preparation and analysis.

There is a tool worth mentioning, which we will label as a “gateway tool” between a text editor and programming known as the spreadsheet (MS Excel, OpenOffice Calc). It allows non-programmers to do some amazing things and get some quick and accessible results. While spreadsheets have their own sets of challenges and draw-backs, they also have some benefits. If the amount of data are not too large and the task is not deciding the future of the world economy (see case study), then excel may be the best tool for the job. We would strongly suggest seeing excel as a temporary solution. It does well at quick one-shot tasks. But if there is a repeating analytic task or model that is used consistently, moving to a programming language is highly recommended.

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 and if we want to apply a conversion to a row of data, we have to click to select the row and apply a conversion. This works for small or quick data sets, but trust us, you will (more often then you think) have to go back to the source data and re-clean it. Either you got another day of log files, or you realized you could pull another data point from the source data, or you identified an error in the process. Something, somewhere will cause you to go back to the source and repeat the data cleaning and conversion. Leveraging a spreadsheet means a lot more clicking, while write a script to convert enable consistent and easy re-execution of the process.

type="caseStudy"

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” (<http://files.shareholder.com/downloads/ONE/2532388207x0x628656/4cb574a0-0bf5-4728-9582-625e4519b5ab/Task_Force_Report.pdf>) in which they investigate the loss of $6 billion in trades. In an appendix they have this:

“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. In addition, many of the tranches were less liquid, and therefore, the same price was given for those tranches on multiple consecutive days, leading the model to convey a lack of volatility. While there was some effort to map less liquid instruments to more liquid ones (i.e., calculate price changes in the less liquid instruments derived from price changes in more liquid ones), this effort was not organized or consistent.”

“…the model was approved despite observed operational problems. The Model Review Group noted that the VaR computation was being done on spreadsheets using a manual process and it was therefore “error prone” and “not easily scalable.” …” (p. 105)

“CIO’s implementation of the model was flawed. CIO relied on the model creator, who reported to the front office, to operate the model. Data were uploaded manually without sufficient quality control. Spreadsheet-based calculations were conducted with insufficient controls and frequent formula and code changes were made. Inadequate information technology resources were devoted to the process. Contrary to the action plan contained in the model approval, the process was never automated.”

As with any complex system, catastrophe requires multiple failures[[2]](#footnote-2). We cannot point to their use of a difficult-to-understand spreadsheet as the primary cause, but certainly it appears to have been a contributory factor.

Throughout this book, we will have examples in a few different languages, but don’t think data analysis is limited to these languages or tools. They serve as an example of what’s possible and how we may go about solving specific problems within data analysis. Excel makes the list because it is fairly ubiquitous these days and many people attempt to leverage it already for data analysis. Python and R make the list because they are open source efforts and free to download and they are incredibly robust languages with many convenient features that make data analysis pleasurable.

One last note, programming is not just for data preparation, though we end up investing a lot of time and energy at that stage. Many of the languages we mentioned have robust data analysis features built into (or onto) the language. So the same programming language we used for data preparation can be used for the analysis itself. For example, the R language was developed by statisticians for performing data analysis. Python, with the addition of packages like NumPy, SciPy and Pandas offers a rich and comparable data analysis environment as well.

But we’re not done there, just preparing and analyzing the data is not enough, we also need to communicate our results and one of the most effective methods for that is data visualization (of which we devote several chapters to here). Again, Excel has the ability to produce graphics and with judicial modification of the default settings, good visualization can be done with Excel. However, in our opinion, flexibility and detail in data visualization is best achieved through programming. Both Python and R have some feature-rich packages for generating and exporting data visualization and we will cover that in later chapters.

Data Management

If there was one skill we may be able to skimp on, it’s data management, but we would do so at a high cost. Within information security our data points can quickly multiply and if we don’t learn to manage our data the strain of the ever-expanding data sets will take its toll on our efficiency and effectiveness. When we start out with simple data sets, we can leverage spreadsheets and text editors. We outgrow that stage quickly though and we must move into programing languages and simple formats like comma-separated value (CSV) files. At this point, we may see some benefits by moving our data into a database, but we haven’t quite reached that painful point.

There comes a tipping point, either in the complexity of the data or the amount of data that we must move to a more robust infrastructure for our data. Let’s not be fooled here, the relational databases of yesteryear were reserved for the biggest of our projects, but that is no longer a helpful mindset. Many of the database systems we discuss in Chapter 6 can be installed on a desktop and simply help make the analysis more efficient and repeatable. There have been analysis efforts that we’ve installed a local database and imported our data for a one-time project. No longer are databases so unwieldy that they can’t help with the smaller efforts as well.

When we talk about data management skills, we naturally jump to databases. We want to have enough knowledge to install a Mongo or CouchDB, dump our data in and leverage that for our analysis. However data management is more than databases. It also requires some work on data quality and data integrity. We want to be sure the data we are working with is not inadvertently modified or corrupted in some way. Also, we work in information security and we’d be negligent if we didn’t talk about that for a bit here.

There seems to be a pattern in technology: a passionate need drives a handful of geniuses to work their tail 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[[3]](#footnote-3). As a result, most of the authentication and permissions were constructed to protect the system from unintentional errors in their programs, and not a malicious user. Of course as the technology evolves and grow, security usually comes with more experience (and failures).

This focus-on-function is the same thing we are seeing in our current “data revolution” and new brand of NoSQL databases and data management tools. While the authentication and security features are far better then the early days of UNIX, they typically do not compare to the security and features of the more established relational databases. We will not dwell on this point though. Just keep in mind that data management is just as much about data security and it is making the data accessible and usable.

Statistics

Think of statistics as a collection of skills we can focus on, because it turns out to be a pretty deep well to drink from. We begin with descriptive statistics (describing the data we collect) and move into inferential statistics (making inferential statements about a larger population from a smaller sample). But we also want to be careful how we plan the analysis and collect the data. Statistics will teach us about the “design of experiments” to ensure we aren’t putting more faith into the data then the collection method warrants. Lastly we have two relatively new additions of data mining and machine learning which we won’t be able to cover in much detail in this book. But as computers have evolved, rather than applying inferential statistics to enable people to understand the data, we are now applying algorithms to teach computers to understand the data and do some amazing things like finding intricate patterns or classifications that the human brain would struggle to find.

We should also approach statistics with a healthy degree of respect and humility. As we slide more and more into the depths of applied mathematics, we begin to realize that it’s easy to look at data and assume there is meaning where none exists (technically called a type I error). But what’s really important is that this error can occur with or without data, especially when we work with networks of complex systems and an intelligent and adaptive adversary attempting to affect the confidentiality, integrity or availability of our information. One of the best tools in the toolbox to limit this error is the combination of experience and data. Even with the combination though, errors do occur, but more often the not, statistics will help us not be fooled by our environment.

Visualization (a.k.a. Story Telling)

The final skill is what we are labeling “visualization” but really it is about the skill of communication. Being able to encode the results and meaning of data analysis into a message that people can understand is another critical skill in the process. Typically this does involve producing one more graphics, perhaps interactive, that can help convey that message.

Combining the Skills

The skills we have listed here are what we want in order to make the analysis run smoother and reduce the chances we are misled by the data. While we may have portrayed these skills as being in a single person, that is not a requirement. As the data stores grow and the demands for analysis gets more embedded into the culture, spreading the load among multiple experts in maybe one or two of these skills will help lighten the load. And if you are in the position of having to hire for this type of role: finding all of these skills in a single person may be a bit hard to find. Take the time to talk through each of these points with candidates though and just be sure there is at least some element of each of the skills we talked through here.

Stages of Analysis

“My job was to find questions about baseball that have objective answers, that’s all that I do, that’s all that I’ve done.”

-- Bill James, Sabermetrician

Now that we’ve looked at the skills that contribute to a good data analysis program, it’s time we turn our attention to the analysis itself. Just jumping in and grabbing data is like running a race without knowing where the finish line is. We want to have a good concept of what we’re trying to learn from the data. Therefore, every good data analysis project begins by creating one or more well thought out **research questions**. A well-prepared research question may be one of the biggest challenges in data analysis. Once we understand the research question, we identify the data that may answer that question and start the **data collection** process. Once we have all the data, we may need to spend quite a bit of time in **data preparation** and getting it readyfor analysis. Then, of course, we will have to do the **data analysis** and attempt to answer the research question. Once we’ve completed the analysis, we have to communicate our answers either through words, tables or **data visualization.** As a final step, check for any **feedback** from the analysis as a method to improve with each iteration.

Unfortunately, all of these steps occur with or without data, and often times without much thought. Take for example the largest breach you can think of, we may consider RSA, Sony, Heartland or if we’re really desperate we could go all the way back to the TJX breach. These are very powerful stories and often are effective at stirring up an emotional response. Hearing these stories we naturally jump to causal inferences. Because of the personal nature of the stories and the emotional response, we often skip right over the research question and any thought to good data collection practices. The result is a very poor data analysis effort. In this case the “data” is an anecdotal story, filtered through many levels and the collection method is just about as poor as we can get. Even though we can relate to the breach at a fortune 500 organization, there are still another 499 not in the headlines.

without realizing that we are skipping some critical steps in the

When we here these stories we are naturally making causal inferences. In the case of RSA we could walk away thinking “This is proof that I could be hacked at any moment” If we here a single but powerful story, we may produce a link between the cause of that single incident and all future incidents without realizing that the sample size is one and the data collection method is questionable. Stories are powerful tools.

Preparing the Research Question

A good research question will bring efficiency, purpose and context to the analysis by creating a clear and focused goal for the analysis. Plus, by spending the time to form a good research question, we may also think through the others steps to improve the overall methodological design of the analysis. However, choosing a poorly defined question (or no question at all) could send the analysis off in a tailspin. Without a well-formed question guiding the analysis, we may waste time and energy seeking the first convenient answer in the data and end up answering a question nobody wanted to ask in the first place.

For example, figure 2.1 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 any 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, “we should try to scale back on prescription-selling spam.” Outcomes like figure 2.1 are the result of a poor question selection and/or skipping a question altogether -- it is data analysis for the sake of data analysis and does not help to inform us about our environment in any meaningful way.

Even as we build our I.T. systems and applications we want to have some idea of the types of questions we will want to answer. There is a huge difference between showing when a system fails and how a system fails. Most of the time we are not interested in just the simple fact that something failed, we want to know why, how (and how to avoid it next time) and in the case of information security, by whom. It makes data analysis much easier if we can gather this type of data from the log files, rather than trying to generate and collect the data outside of an application. Thinking of these questions during the development of an application is far more efficient then afterwards.

Let’s continue on with the spam example. A good research question might be, “How much time do employees spend on spam that is not blocked by the spam filter?” Now that we’ve framed the question like this, it’s pretty clear to see that we may not look to our spam filter logs to answer this spam-related question. Perhaps we would look for any logging 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 study in which we select a sample of users and ask them to record amount of spam and time spent going through them for some limited period of time.

According to Lipowski[[4]](#footnote-4), following three relatively simple steps creates a good research question: (1) ask a series of interesting questions, (2) select the best question for research and (3) transform that question into one or more objective research questions. If we haven’t made it clear yet, the more time and effort we put to forming a good research question, the more focused and beneficial the analysis will ultimately prove to be to the organization. Spend some time in this section and get to know it, it will help out in the long run. The overall goal of the analysis may be slightly different then the research questions in the end.

We start forming a research question with ideas or general topics and generate a series of questions from there. These initial questions could stem from observed problems or gaps, perhaps starting as a curiosity or a hunch, or perhaps we just want to question if some long-held belief is still valid. We want to leave the field wide-open at this point and not start with any given data in mind (though see the section below on exploratory data analysis). As we see from the spam example, the topical research question couldn’t be answered by the original data source. We need to have the flexibility to focus on a set of questions that are interesting and informative.  
 Once we get this potpourri of questions we should pare down the questions to a single research question. This will serve as the over-arching goal of the analysis and help guide any decisions we need to make during the analysis. This question should be relatively specific. For example, if we are looking into whether or not to buy <insert use case here?>

Now comes the important part, forming a question that can be answered with data. Notice the opening quote of this section from Bill James (whose work is portrayed in Michael Lewis’ Moneyball). Even though Bill James has developed and discussed countless baseball metrics, as he states in a 2010 lecture titled “*Battling Expertise with the Power of Ignorance*.” Rather then attribute his success to his ability to collect and analyze data, he attributes his success to his ability to form a good question. He spent his time looking for the right question, from which the answer could be pulled from the data. This is an important distinction.

There is one final point to make here and that is the research question can be modified depending on the type of information we’re looking for. We may engage in data analysis to be **descriptive** (describing some phenomenon or event), **comparative** (analyzing the relationship of two or more entities), **explanatory** (looking for contributory or causality for some phenomenon or event) or **normative** (looking for the best “normal” way something should perform or be done). Thinking of these differences may help as the research questions are being formed.

We don’t want to ask a question that begins with, “How much risk is there with our authentication mechanisms” because risk is an abstracted concept from data. Now if we wanted to transform that into a good set of objective questions we could form some research questions that choose to describe what we’ve seen: “How often are stolen passwords used in attacks?”

Data Collection

Once we know the questions we want the analysis to answer, it’s time to seek the data that would best answer the questions. Sometimes, the data collection is sometimes relatively straightforward, it’s sitting in a set of log files, or an existing database and all we have to do is grab it. Other times, we may have to create a process to collect the data from systems or we need to put together a survey to extract data from people. Either way *how* we collect the data may determine *what* we can do with or infer from the data.

For example, if we really did want to know (for some strange reason), the proportion of spam emails we block that offer discount prescription drugs we can grab the logs of the spam filter and count up all of the blocked spam (known as the *population*) and then count how many were in the prescription drug category. This method allows us to count and *describe* what we have observed (*descriptive statistics is discussed in chapter 4)*. But what if we wanted to estimate the proportion of prescription drug spam on the Internet as whole? Could we infer that by looking at just our spam data?

To look at answering that, let’s return to the research question we formed in the last section, “How much time do employees spend on spam that is not blocked by the spam filter?” It is infeasible to record all the time each employee spends dealing with spam. But what if we picked out just a handful of employees and understood the time they spend dealing with spam? Would that help us get close to answering our research question? Even though the answer to that question is “yes”, we have to append a whole slew of qualifications on it. This is where one aspect of statistics can help and the key phrase is “design of experiments”. We have our friend from Chapter 1, R. A. Fisher to thank for much of our knowledge in this area who wrote a book on this topic in 1935 (appropriately titled “The design of experiments”). This work gave birth to many of the research tactics used across most every scientific field of study.

Back to our problem though, if we can’t grab all the data, we want to grab data from a sample that is *representative* of the larger population. We will talk about how this works in chapter 5. For now, just know that whether we are talking about a survey or log collection process, we want to be aware of the population we are drawing the data from and how the population is represented in the data we are collecting. When we are collecting a subset of samples from the population there are two concepts we must be aware of and how they influence our results: sample bias and sample error. We try to reduce and even eliminate sample bias, while we account and adjust for sample error.

Sample bias occurs when the sample is *not representative* of the larger population (see the case study from the 1938 U.S. elections) typically caused by a systemic flaw in the selection method. The flaw will either over-represent or under-represent some subset of the population and simply gathering more samples will not remove or even reduce the bias in the data. A classic example is the self-selected survey where we may send out a survey our users to gauge opinion of the enforced password policy. We construct a set of questions and send it to every employee, but only a small proportion respond. This is referred to as voluntary-response bias since those willing to volunteer for the survey are motivated by their strong opinion on the topic. This is closely related to the non-response bias, which emphasizes the part of the population excluded from the survey. Non-response bias describes those people who are unwilling, unable or unmotivated to respond, who are then excluded from the sample.

type="caseStudy"

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 over 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 U.S. hadn’t let up yet and they ended up polling people with phones, club memberships and magazine subscriptions. They systematically polled the middle and upper class, which generally leaned towards Landon, and arrived at an answer that was mathematically correct and yet completely wrong. Through their selection of sources to contact people they introduced bias into their sample. The fact that they had 2 million responses did not reduce that bias; they just had a larger sample with bias.

To add to their embarrassment, at the same time, a young man named George Gallup had gathered a relatively small sample of just 50,000 voters but using a much more representative sampling method and correctly predicted Franklin Roosevelt as the winner of the 1936 elections, which catapulted his name into the spotlight as the Nate Silver of the day.

There are multiple ways bias can creep into our data collection and affect our results. Another form of bias may be introduced in how the questions of surveys are asked or assumptions we make in preparing the data may introduce bias. Again, we may never completely remove the sample bias, but we can take steps to reduce the impact of the sample bias. on our analysis.

As another example, we collect and study breach data. But we have what’s called a convenience sample.

I feel like I’m going down a deep hole here and I won’t get back in time for the end of this chapter.

There are times when sample bias is unavoidable; we cannot study the safety effects of seatbelts by choosing a random sample of people and crashing them with and without seatbelts. Hospitals are limited in their research subjects to those with the problem they are research and have visited their hospital.

Sample error on the other hand, is not really a mistake or error as the name implies, it is just trying to describe (and we will then account for) the random variation in the system we are observing. We can measure this thanks to a wondrous property of samples called the central limit theorem.

Data Preparation

Despite what students see in the classroom, it’s very rare when data comes in the format we need. This is where our skills in programming really come in handy. But even with numerous tools and/or custom programs, this stage still requires a lot of time and effort. Perhaps we need to transform the format or structure of the data, or we may have to correct missing or invalid entries, or perhaps we just want to merge multiple sources in order to answer our research question. While it may be possible accomplish this task with various tools like DataWrangler or Google Refine (see appendix for a full list of data preparation and cleaning tools), they simply cannot account for everything. Leveraging those tools for their strengths and filling their weaknesses with custom scripts reduces the amount of effort we spend at this stage.

One other point worth repeating: it is inevitable that all of the steps in data preparation will have to be repeated. Either we discover an error in the cleaning, or got to the analysis stage and realized we should have included one other data point, or our source data was updated with more entries. Or we just want to repeat the analysis for another days worth of logs. The reasons are plentiful and all point to automating the data preparation step. Most of the data conversion tools do not support automatic re-execution of the step and will require the user to click and drag through the interface for the updated data. This not only gets tedious quickly, it also prevents a conversion to automation, which is nice for the generation of metrics or supporting any type of dashboard interface. For repeating analyses like that, we will need to remove all the touch points we can, so writing a python script to take the raw data source as input and output the final data ready for analysis (or input into metrics or a dashboard) makes a lot of sense.

Data Analysis

Finally we get to the juicy part of the whole process and what a good portion of this book is about. The good news is that we can pull a lot of information from basic descriptive statistics. Simply gathering up data, counting, comparing and describing it can be enough to answer simple research questions (enough to inform a decision). But we want to be careful of overconfidence in what descriptive statistics can do. People often underestimate the variability present in our systems and the actions of our users and adversaries. We want to ,

Writing something here about data analysis, maybe cover descriptive, inferential and then some machine learning and data mining techniques?

Communication / Visualization

We want to talk about how the analysis isn’t over until something useful is doen with the results. Perhaps that a visualization but it could just as easily be an email, a paragraph a table of numbers or even a single number. Somehow we have to communicate the outcome of our analysis.

Feedback

Exploratory Data Analysis

This is a slight variation on the stages where we start with a collection of data and we aren’t exactly clear on the research question, or the question is annoyingly vague “what information can we pull from this log?” In this case we do begin with the data, but not to arrive at a conclusion but to arrive back at a question. In other words, exploratory data analysis helps us understand what questions may be answered by this set of data. But then we should circle back around and form a proper research question and go through the steps outlined in this chapter.

Overall, exploratory analysis may serve to see the following types of relationships:

Exploratory data analysis … employes a variety of techniques to:

* Uncover underlying structure
* Extract important variables
* Detect outliers and anomalies
* Test underlying assumptions
* Develop parsimonious models
* Determine optional factor setetings
* <http://www.unitedbiosource.com/pdfs/webinars/20121031-exploratory-wasiakr.pdf>

1. For example, we may argue that playing a musical instrument helps teach the creative and critical thinking necessary for good analysis. But alas, that did not make it on the broad list of skills we cover here. [↑](#footnote-ref-1)
2. See Richard Cook’s “How Complex Systems Fail” for a brief and wonderful dissection of this topic. http://www.ctlab.org/documents/How%20Complex%20Systems%20Fail.pdf [↑](#footnote-ref-2)
3. <http://www.cse.psu.edu/~tjaeger/cse443-s12/docs/ch4.pdf> and one of the first solutions for the UNIX platform was to simply store the users passwords in a clear text file on the system: <https://info.aiaa.org/tac/isg/SOFTC/Public%20Documents/Technical%20Working%20Groups/Cyber%20Security/Password%20Security%20A%20case%20Study.pdf> [↑](#footnote-ref-3)
4. Paper from Lipowski: <http://www.ashpfoundation.org/MainMenuCategories/ResearchResourceCenter/FosteringYoungInvestigators/AJHPResearchFundamentalsSeries/Developinggreatresearchquestions.aspx> [↑](#footnote-ref-4)