Chapter 1: Preparing for Data Analysis

“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

We know there is a natural allure to data science and everyone wants to achieve that sexy mystique surrounding security data analysis (oh yeah, we know). So we will begin this chapter by talking about the various elements that come together to create that mystique that everyone is seeking. It doesn’t really matter whether the skills are present in a single person or a team of people. Each skill should have at least some representation in order to build a successful function we are calling security data analysis. It would also be fair to call it security data science, as it’s more than just data analysis. It is combining the domain expertise of information security with computer science and statistics.

We are trying to not get hung up on terms like “data science” here (and that may be evident as we may throw labels around a bit too loosely for some), but instead we want to focus on one simple task: learning from data. The serious field of data science is arising because we’ve got a lot more options to learn from data than just classic statistics. We are living in a time where massive computing power is ubiquitous and that power opens up possibilities and combinations that R. A. Fisher just didn’t have in the first part of the twentieth century. We want to exploit this power not just for classic statistics but also for disciplines arising as a direct result of that computing power like data mining, machine learning and data visualization. Within security data science, we want to draw together statistics, computing power and visualization techniques around the data we are able pull from our environment. And all of that begins not with products or tools, but with our own skills and abilities.

Before we get to the skills though, there are a couple underlying personality traits we see in data analysts that want to discuss: 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 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 we are able to turn a light on in an otherwise dark room, when we can describe some phenomenon or point in a direction to travel and that’s what we’re after. We are uncovering those tiny moments of enlightenment hidden in plain site for those who know where to look.

Once those pearls are discovered and polished, they must be shared with others to really have value. Otherwise, we will have constructed a house that nobody lives in. But it’s not so easy as just pointing at all of the work and saying, “see!” We have to step back and think of the best way to communicate our discovery. The complexity in both our systems and analysis make it difficult to convey the results in a way that everyone sees what we have discovered. Often times it takes a combination of words, numbers and pictures to communicate the insights in data and even then some people will take away nothing and others will take away too much. But there is still a skill in there for being able to condense complexity into a paragraph, table and/or graphic. Having the ability communicate will be invaluable. It’s not enough to collect the data, make sense of it and gather those rubies to share with others. We need to develop the gift of gab and tap into the power of stories and wrap it all around the truth we’ve uncovered. Only then can we bring the value from the data to where it is desperately needed within information security.

While we could spend the entire book to create an exhaustive list of skills to be a good security data scientist[[1]](#footnote-1), we are going to cover the following skills/domains that a data scientist will benefit from knowing within information security: **domain expertise** (setting and maintaining 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) and **visualization** (communicating the results effectively).

These are the skills that make up a functional data analysis effort. Smaller and simpler efforts may seek a single individual to cover all these skills (with varying degrees of success). However, as the workload 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, where 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 contributes a significant and important piece to the workings of data science.

Domain Expertise

Saying that a data scientist needs domain expertise should go without saying and it may seem obvious when we lay it out like this, but this cannot be talked about enough. There are some important points to discuss about the benefits and pitfalls of bringing domain expertise into data analysis (or are we bringing data analysis into 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 resist or even push away data science. Either it is seen as a perceived threat, as if data analysis can replace a domain expert (it decidedly cannot) or perhaps there is a resistance when analyses challenges long-held beliefs that have been solidified into conventional wisdom. But despite the challenges, domain expertise is beneficial in the beginning, middle and end of data analyses and nowhere is that more obvious then when we focus on attempting to create a good research question to ask of data.

Creating a good research question, as we’ll see when we dive into the stages of data analysis, is setting true north for the entire duration of the analysis. That research question 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 and how this wraps into the goals of the business. Attempting to develop the research question without good knowledge of the environment may lead to superficial and/or irrelevant findings and result in chasing ghosts and will generally doom the analysis before it even starts. Therefore knowledge of information security is very important when forming that research question. But creating a good research question is also a learned skill because the questions need to be asked in a way that the data can answer. Asking questions about hypothetical constructs like “how much risk…” while good overall questions, are horrible questions to ask of data.

Once the purpose of the analysis is set and the data is gathered, domain expertise also brings the data and analysis into context. Many problems can arise during the analysis without an understanding of what the data represents and the relationships within them. 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 by a freshly minted CISSP. Regardless, if the domain expert and data scientist are not skills within the same person, countless back and forth communications (and perhaps some frustrations) can be generated here. The data scientist needs to learn from the domain expert about the characteristics within the data that affect the analysis and the outcomes. And the analyst needs to communicate any and all assumptions made during the analysis (there are always some assumptions), because without the intuition of a domain expert, the assumptions may miss some of the subtleties that come natural to information security professionals.

Understanding the security 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. For example, with data we can show that many threat actors employ various techniques to target valid credentials. But the analysis stops there, it is only with domain expertise that we can attempt to make sense of that. Knowing how systems work it’s easy to see that logging into a system with valid credentials is much easier than having to exploit a weakness on each system. Seeing the cause in the pattern of stolen credentials is much easier with experience then it is with data.

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 a threat, trying to replace their work and experience with models and formulas. This objection is not only misplaced and misinformed, but also counter-productive. Statistics and all the related fields only have value in some other context and then only 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. Being an experienced *security analyst* does not make a person an experienced *security data analyst*. As information security professionals, we should understand that data analysis is a supporting role, and as a data analyst, we should understand that we need to be in that supporting role. We should not approach a statistician to present on identity management any more then we should approach an information security practitioner to present on unsupervised clustering algorithms.

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) Whether we agree that particular finding or not is irrelevant. The point here is that as a data scientist, we must be open to the possibility that long held “facts” may turn out to be just an assumption which solidified in our culture over time. Unfortunately, these nuggets of conventional wisdom may be deep-rooted and convincing people otherwise may require a lot of patience (as well some good data analysis).

The opening quote to the chapter from Josh Billings sums up this point rather well: “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.” This over-confidence also plays into our example on assigning meaning and significance to analysis results. Even though we may able to explain the results of the analysis (in our case, why credentials are targeted by attackers), but we should be careful on jumping to conclusions. It is easy to see the wrong connections and attribute the wrong meaning in hindsight and knowing that is half the battle.

type="caseStudy"

Smoking is unhealthy, right?

For years, science and statisticians debated the relationship between smoking and lung cancer. Through the 1940’s and 1950’s cases of epidermoid carcinoma of the lung were on the rise and medical experts sought to understand why. Statistical studies starting in the 1950’s would show correlation between smoking and lung cancer, but they often had flaws. These flaws were not errors in anyway, the flaws were present because the real world presented imperfect data and the researchers did the best they could to compensate for the imperfect data (remember the real world is messy). R. A. Fisher (who was often shown smoking on his pipe) was an outspoken opponent of those studies and would put considerable effort into dissecting and refuting the techniques and conclusions in those studies. His personal belief was being expressed through his expertise in statistics to the point where he even accused researchers of manipulating their data.

Finally, in 1959, Jerome Cornfield and several other researchers took a step back to conduct a meta-analysis, which is analysis done by looking at the combination of several other studies (a tactic Nate Silver would apply to the 2012 U.S. presidential elections with great success). They showed how the aggregate results of all the other studies provided overwhelming evidence that smoking causes lung cancer. Even though each study was flawed in some way, they were flawed in different ways and the aggregate had a consistency that was enough to quell any uncertainty. It would take years for this to permeate into the culture, but this paper was the tipping point. Fisher died in 1962 and the debate on the causal effect of smoking on lung cancer slowly fell silent.

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 that fact.

This is a cold-hard lesson in data science: data comes to us in a wide range of formats, states and overall quality. The data may be embedded in unstructured or 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 frustrating format known as XML. 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, this is where the ability to whip together will provide more functionality, flexibility and efficiency in the long run. 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 manipulate it into whatever formats the analysis software or we would want. There is certainly a large collection of data conversion tools available (appendix A) 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 the data in our world, not vice versa.

Most every modern language will support basic data manipulation tasks, but the scripting languages (python, R, perl) appear to be used more often in data analysis then their compiled counter parts (Java, C). However, the programming language is somewhat irrelevant, as the end results (and a happy analyst) is way more important then picking any “best” language. Whatever gets the job done with the least amount of effort 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) 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. Though spreadsheets have their own sets of challenges and drawbacks, they also have some benefits. If the amounts of data are not too large or complex 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 over and over, moving to a structured 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 data sets or quick tasks, but trust us, you will (more often then you think) have to go back to the source data and re-clean it. Either another day of log files needs to be processed, or we realize we should have pulled another field from the source data, or (gasp) we identified an error in the cleaning process. Something, somewhere (and possibly repeatedly), will cause us to go back to the source and repeat the data cleaning and conversion. Leveraging a spreadsheet means a lot more clicking, while writing a script enables an easy, flexible and consistent execution of the cleaning process each time it runs.

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. 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.

… Data were uploaded manually without sufficient quality control. Spreadsheet-based calculations were conducted with insufficient controls and frequent formula and code changes were made.

In the report, they labeled the excel-based model as “error prone” and “not easily scalable”. As with any complex system, catastrophe requires multiple failures[[2]](#footnote-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.

Throughout this book, we will have examples in a few different programming languages and serve as examples of what’s possible and how we may go about solving specific problems within data analysis. Excel will be used in a few examples because it is fairly ubiquitous and many people attempt to leverage it already for data analysis. Hopefully we can raise awareness around the limit of spreadsheets and offer some other avenues to pursue for data analysis. But if the analyst has built up some spreadsheet skills, nothing will beat it for quick analyses (at least initially).

After the data is ready for analysis, we can continue to benefit from understanding how to program as many of the languages we mentioned have robust data analysis features built into (or onto) the language. 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 a comparable data analysis environment as well. In some cases we can even combine the function in the same script. We can write one script to grab the source data, manipulate and clean it, then run the analysis on it and display the results.

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 some examples in later chapters.

Data Management

If there was one skill we may be able to skimp on, it’s data management, but we would only be able to skimp on it for a while. Within information security (as well as most other disciplines) our data can quickly multiply. If we don’t learn to manage it, the strain of ever-expanding data will take its toll on our efficiency and effectiveness. As we mentioned, we can leverage spreadsheets for the simple analyses. However, we will quickly outgrow that stage and we must move up 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 it still may not be necessary.

At some point we reach a tipping point, either through the complexity of the data or the volume of data and moving to a more robust data management solution is going to be inevitable. There is a misconception that the large relational databases of yesteryear are 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 help make the analysis more efficient and scalable. Once data management skills become more natural, their benefit can be used on even the smallest of projects. We’ve installed a local database and imported our data for some smaller one-time projects.

When we talk about data management skills, we naturally focus in on 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. Data management is also about managing the quality and integrity of the data. We want to be sure the data we are working with are not inadvertently modified or corrupted, either through misconfiguration of a tool or rushing through the cleaning and conversion a bit to quick. Whatever the reason, it doesn’t hurt to have some checks that keep an eye on data quality and integrity, especially over long-term data analysis efforts (e.g. metrics). It’s the like the concept of unit tests while writing code. We may want to automate some integrity checking of data after any new import or conversion.

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. But let’s take a step back for some context first. There seems to be a pattern repeating in our history: 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 from a malicious user. 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 definitely seeing the more emphasis on functionality and less on the security (though their security would still kick the security of the early UNIX systems). Most of the new data management (NoSQL/Hadoop) platforms were not designed with many of the security policies or compliance requirements of most enterprise networks. The result is a distributed computing platform with some difficult security challenges. 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.

type="reference"

I am thinking we pull in Adrian Lane and/or Mortman here (or at least reference some of Adrian’s work on big data security. Idenfity a “top 5” list of things to consider in data security.

Statistics

Perhaps we are a little biased here, but picking up some skills around statistics will improve most 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 (did we go too far with that one?) Statistics, despite classes with that general name is actually a collection of skills and focus areas (much like information security is evolving into multiple disciplines). Statistics has evolved (and continues to evolve) to meet the deceptively simple-looking task of learning by observing and measuring, but that turns out to be a pretty deep well to drink from. It’s now a multi-faceted profession that has touched most every field of science.

There are a few reasons to learn more about working with data. First, Even though data never lies, it is far too easy to be tricked by the data. We, as heuristic beings, are capable of pulling out patterns and meaning from a complicated environment. Our ability to see subtle connections and patterns helps us on a daily basis. However, that skill can also mislead and we may think we see patterns and connections where none exist. Understanding statistics can raise awareness of this and the tactics can help minimize incorrect conclusions. Secondly, even though we just said data never lies, the way it’s collected can create deceptive data. As an example, asking opinions of those around us may mistakenly confirm our own opinion since we naturally surround ourselves with like-minded people (we’ll talk about sample bias later on). Finally, statistics can provide some handy methods for extracting the story (or stories) from the data. Statistical methods will help us uncover meaning that may be hidden to the naked or untrained eye. We go through a whole lot of trouble to collect and prepare the data, we want to be able to learn as much as we can from the data and the field of statistics has evolved for that specific purpose.

Statistics is not just a collection of tools, it is a collection of toolboxes each with their own set of tools. We can begin with descriptive statistics, which attempt to simplify the data we collect into a few “descriptive” measurements of the whole. Anytime we simplify something we lose detail and this is where visualization can really serve us well. We could do descriptive visualizations that do not lose much detail and yet are accessible and meaningful. Another challenge with descriptive statistics is it only describes the data we collect. Inferential statistics helps us go beyond just describing our observations and enables us to make statements about a larger population given a smaller representative sample from that population. With “representative” being a key word there, statistics will teach us about the “design of experiments” to help us gather data correctly so we have confidence in the samples we gather. This book will focus on these topics, but beyond that we have two relatively new additions of data mining and machine learning. 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 will 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, especially since we work within networks of complex systems and an intelligent and adaptive adversary attempting to bypass our defenses. The best tools in the toolbox to limit the chance of an error in our complex environment are the combination of experience and data. Even with the combination though, errors will (and do) occur, but by applying the rigor and methods within statistics, when can reduce the frequency of those errors and be in a much better position to learn from the mistakes when they occur.

Having now built up statistics on a pedestal, we should point out that we can learn a lot from data without advanced statistical techniques. We briefly mentioned “descriptive visualization” above. Looking around at many of visualizations out there, they are not built on complicated mathematics, but are just describing some set of data and showing the relationships therein. There is no doubt that we can improve our ability to secure our information assets with descriptive statistics and visualizations. There are a whole slew of processes and tools out there designed to aggregate and visually communicate the relationships and meaning of categorical and quantitative data. All it takes is the patience to ask a question, gather the evidence, make sense of it and communicate it out.

Visualization (a.k.a. Communication)

Speaking of communicating it out, the final skill is what we are labeling “visualization” but really it is about the skill of communication. There are multiple ways to classify the types of visualizations out there, but for our discussion we want to talk about two general types of visualization, which are separated by who we want to read and interpret the visualization. The distinction we are making here is quite simple: 1) visualizing for ourselves or 2) everyone else. For example, Figure 2.2 shows four common plots used to diagnose the fit of a linear regression model and let’s face it, these are quite ugly and we would not include these in our next presentation slide deck. They serve to inform the analyst and the R code to read in the data, run a linear regression and generate the diagnostic plots is so unbelievably simple (of course, after we have prepared all the data it’s simple), we’ll include it here:

# read in "data frame" of Internet Users and Bot Infections

zeroaccess <- read.csv("zerogeo.csv", header=T)

# contains the data like this:

# id state population intUsers bots

# 4 California 37350092 29758896 3878

# 31 New York 19746813 16091772 3856

# 34 Ohio 11663946 8949773 2581

# run linear regression on internet users to bot infections

users <- lm(zeroaccess$bots~zeroaccess$intUsers)

# set the graph to be 2x2

par(mfrow=c(2,2))

# plot it

plot(users)

Figure 2.2 Diagnostic plots for regression model of bot infections to Internet users [FILENAME 793725c02f002]

These graphs are generated as a way to understand certain relationships and attributes of the model, but these graphics raise some eyebrows around entries 4, 31 and 34 (California, New York and Ohio respectively), but especially that fourth entry. This may indicate any number of things, but figure 2.2 is just an example of creating visualizations for the purpose of analysis and not for an external audience. We will use this type of visual to see into the data to detect anomalies, relationships or other aspects of the data for the purpose of understanding it during the analysis. Very little effort is spent on making these pretty or presentable since they are meant to be a part of the analysis, not the result.

The other type of visualization exists to explain the data or results of an analysis to others. These are typically much prettier as we are producing these visuals as a communication tool.

INSERT MAP of ZERO ACCESS INTERNET USERS HERE?

Once last bit worth talking about, and that is the power of stories. There is a meme in data visualization stories that our graphics should tell a story. Anyone blessed with the gift of the gab will be able to espouse on the power of a good story. Stories provide context and a narrative that can personalize the message. Stories can touch our deepest held beliefs or inspire us into action. When we create visuals that tell a story, and that story is based the evidence we’ve discovered in the data, we can create a message that is interesting, meaningful and memorable.

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

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. 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 setting a goal and creating one or more well thought out **research questions**. A well-prepared research question may be one of the biggest pitfalls in data analysis as many efforts skip this step. With one or more well-formed research questions, we identify the data that may answer that question and start the **data collection** process and then 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, we should seek **feedback** from the analyses as a method to improve our skill and accuracy with each analysis effort.

Remember, the purpose of data collection is to learn from our environment and that can be done with or without data with varying degrees of success. At some level each of these stages will always be done. For example, think of the largest breach you can remember. Most security professionals know these stores and can relate them as supporting points. As we think of these, the research question may be glossed over or something poor like “how does this relate to me?” Which opens us up to drawing convenient and unfocused conclusions. The data collection process becomes the various stories and news articles we happen to come across. The data preparation is a mental operation where we attempt to logically line up the facts. For the analysis, we mentally compare and correlate the stories to our own environment. The result is conclusions that emphasize the emotional and favor the simple. Framing famous breach events like an analysis effort, we can see how it may be easy to arrive at some spurious conclusions and either over-react or under-react to the data. While the “gut-feel” approach serves us well on a daily basis, there is a point where our ability to accurately see the patterns and pull meaning is corrupted by the complexity of the event. Complexity will silently overtake our intuition and leave us drawing overly simplified conclusions or worse, conclusions that are completely wrong. These are good examples that stories are powerful tools and are often no match for rigours data collection and analysis.

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 and prepare for 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 convenient answers in the data or worse, we may end up answering a question nobody was asking 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 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 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 analyzing data and does not help to inform us about our environment in any meaningful way.

Figure 2.1 The outcome of a poor research question [793725c02f001]

But we are also in a unique position as information security practitioners to build for future data analysis. We often are involved at some level in I.T. project and we should be approaching those efforts with an analytic mindset. 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. It’s natural to build a system and alert when there is a failure, but there is a huge difference between showing when a system fails and how a system fails. When it comes to learning from our environment, we are rarely 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 the applications and systems generate this data via log files or other artifacts rather than trying to generate and collect the data after it’s rolled out. 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?” We don’t stop at how much spam is not blocked, whatever that number is it will have no contextual meaning (nobody can internalize the difference between ten-thousand or twelve- thousand spam emails). What we want to know is why kind of an impact does spam have on employee productivity? While “productivity” may be a challenge to measure directly we can flip that around and just assume it is impossible to be productive when employees are reading and deleting spam. Therefore, what we really want to measure is time as it is directly spent dealing with unfiltered spam.

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. We really don’t care that 17,642 emails were blocked at the perimeter or that 95% of spam is successfully blocked and it really doesn’t matter how much email on the Internet is spam. Even though the purpose of this analysis is to assess the effectiveness of the spam filtering system, we want to collect a measurement of employee time as an indicator of filtering effectiveness. 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. Either way, we’ve identified what analysis would be useful, not what type of data would be useful to analyze.

Steps to Creating a Good Research Question

Creating a good research question is relatively straight forward, but requires a bit of discipline. 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. If we go back to the spam example, maybe we realized the spam filter hasn’t been updated in a number of years, or we are seeing more spam in our own inbox then we’d like. 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). Otherwise we end back up at Figure 2.1. Once we took the time to develop a good research question, it was obvious that the data we had was not the data we needed to answer the question we need answered. We need to have the flexibility to focus on a set of questions that are interesting and informative and not be tied to a single (possibly myopic) data source.  
 Once we get this potpourri of questions we should pare down the questions to a single question, setting the context and purpose. This will serve as the over-arching goal of the analysis and help guide any decisions we need to make during the analysis. But this isn’t quite the research question yet. Going back to our spam example, we may at this point be asking “Is our spam filtering effective?” or in the case of information security we may have many questions here that begin “how much risk…”

Now comes the important part, forming one or more research questions 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 collected large amounts of data, spent countless hours developing and discussing baseball metrics, he sees his role is “to find questions about baseball that have objective answers.” He arrived at the same conclusion as academics and researchers have promoted for generations: forming a good research question is the cornerstone of good analysis. We don’t spend our time seeking the right data to analyze; we spend our time seeking the right questions to answer. This may seem subtle, but it’s a powerful distinction that will save a lot of time and analyses. But do not mistake this for ignoring the data, we will still want to identify and understand all of our data sources (we will focus on this in Chapter 3).

Focusing the Purpose of the Analysis

There is one more area to explore with regards to preparing for the data and that is how the question can be created to set the focus on the analysis. There are three broad types of analyses. **Explanatory** analysis attempts to uncover what happened or is happening and is looking at the reality of the environment. **Normative** analysis attempts to define the best or most optimal approach to how things should be done and is attempting to define an environment that may or may not match the current one. **Exploratory** analysis is a little different than both of those is that the purpose is intentionally undefined and it’s main purpose to explore the data and flip the research question around. Rather than seeking a good question, we explore what answers are in the data and see if those answer any good questions.

Most of what people think of when they think data analysis is explanatory and it’s what we will mostly focus on in this book. Analyzing how much time is lost reading and deleting spam is an explanatory analysis; we want to explain an element of employee time in our current environment. But even within explanatory analysis we may want a sharper focus, maybe we just want to keep it simple and simply describe what we observe. Perhaps we want to do some comparative analysis and compare between two commercial products, network segments or applications.

Normative analysis attempts to define the best “normal” way something should perform or be done. Normative analysis may be a goal from the start, or it may turn out to be a natural extension of explanatory analysis. In the spam example, a logical follow up question is to ask how much time spent on spam is too much (or how much is acceptable). In which case we may want to compare the costs involved with reducing spam against the cost of the time spent dealing with spam, but the purpose of that analysis is to define an optimal balance that we should strive for.

On Risk

We would be remiss if we didn’t address the relationship of data analysis to risk analysis. Most everything we do in information security is related to this nebulous concept of risk. Whether explicit or implicit, security practitioners are constantly making judgments about what practice is good, what vulnerabilities must be fixed and with what urgency, or defining what policies and controls are essential to our security. These are all risk statements and intended to treat risk to some normative level.

But any research question that sets out to measure risk is going to put the analysis itself at risk. If we want to focus on technical risk[[5]](#footnote-5) we may create a repository of help-desk (or security operations) tickets to help inform our estimation around the frequency of events and put some effort into collecting data around the impact after a breach occurs. It is astonishing how little data is collected that could reduce our uncertainty around objective measurements of risk.

Let’s talk through a simple example. Suppose there is a proposal to expose an interactive menu for the company cafeteria to the Internet. While this may raise all sorts of questions around controls, suppose the proposal is to require authentication with the corporate username and password. The general consensus among the security wonks is the single factor represents a risk and should be made into two factors. How much risk does single factor authentication represent? One valid research question may be to ask how often we see attempts and successes in brute forcing single factor authentication. As luck would have it we can find some very good sources for that data since there is an instance of MS OWA, and it has been leveraging single factor authentication for years. In order to shed light on the probable frequency of events involving single factor authentication we could dig into a similar service and its history.

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, perhaps we’ve created a repository of data sources, or we know it’s sitting in some 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 begin collecting data within an application or system. Or perhaps we need to put together a survey to extract data from people. How the data is collected is quite important as it often may set limits on *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, just as it is infeasible to count all the email on the Internet. 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 the influence of sample bias because it can silently throw off the results and measuring how much bias is present has proven to be a challenge. With sample error on the other hand, if we have a representative sample, we can estimate the amount of sample error and account for its effect.

Sample bias occurs when the sample is *not representative* of the larger population (see the case study from the 1938 U.S. elections) and is typically caused by a systemic flaw in the selection method. Most non-random selection processes (we’ll get into randomness later) will either over-represent or under-represent some subset of the population and simply gathering more samples (with the same flawed sampling method) will not help since the flaw is in selection method, not the sample size. 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. Either way, there may be an underlying pattern in the respondents that reduce our confidence that the samples are representative of the larger population.

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, as we cannot force a person to respond to a survey. Hospitals are often limited to only collecting data on and studying the patients in their hospital. And we cannot pick a random sample of drivers to not wear a seatbelt and crash anymore then we can choose a random sample of organizations and force them suffer a breach so we can study the safety effects of adverse effects. We will never completely remove sample bias, but we can take some carefully designed steps to reduce it’s impact on our analysis.

Sample error on the other hand, is not really a mistake or “error” as the name implies, it is just trying to describe the random variation in the system we are observing. Since this variation is an attribute of the samples in our sample size, we can reduce our sample error by increasing the sample size. 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

A statistics professor walked into the lecture hall and asked all of her students to stand up and pull out a coin. Once they were all standing she instructed them to flip their coin and if they got a tails they should sit down. On the first flip, about half of the students sat down. She repeated this and again half of those standing sat down. She repeated this until finally there was one student standing and she looked at the student as said, “You just flipped a coin and got seven heads in row, please stay standing, tell us your name and share your secret so that others may do the same next time.”

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 through descriptive visualizations. Simply gathering up data, counting, comparing and describing it visually can be enough to answer simple research questions (at least 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.

Once we outgrow simply describing or visualizing our observations, we step into the world probability and probabilistic statements. There are a number of reasons for this. First, we simply cannot gather all of the data, most of the time it is infeasible, expensive or impossible. Therefore, we gather a smaller sample of observations and make inferential statements about the larger population. Election polling is a great example of this. A well-constructed survey of just a thousand people can represent the opinions of millions with a relatively high degree of accuracy. That “relatively high degree” qualification is key. Anytime we measure less than the whole, we can only make probabilistic statements so results of election polls will generally include a “margin of error”. The term *error* here is not synonymous with *mistake*, but represents the natural fluctuations in our observations (also called “observational error”) that occur when we measure less than everything.

Secondly, we have natural fluctuations in the system we are measuring. If we want to record how much time employees spend dealing with spam, of course the amount of spam will fluctuate from day to day and we should expect this. Even in a deterministic system like a computer (that generally does the same task the same way every time), we could expect variations in its functioning especially when multiple components and interconnections are involved.

Lastly, we may have fluctuations in our ability to accurately measure. Going back to our spam example, if we ask people to record their time manually, we could expect some people to forget to record some time or misrepresent their time simply for any number of reasons. This uncertainty will also lead us to probabilistic statements.

A common approach in data analysis (and much of science) is falsification, where we don’t attempt to prove something is true, but simply attempt to show the opposite is false. In the spam example, if we’d only take action if employees spent more than an hour on spam per week. We would want to set up the analysis to disprove that employees spend less than an hour. It may seem a little odd at first, but it’ll make sense. We test to see if systems and applications are secure by testing their insecurity. If we have a lack of evidence showing insecurity, we feel reasonably good about it’s security. We cannot prove secure, but we can prove insecure and that’s the same approach in much of data analysis. In order to prove two things are equal, we disprove their inequality.

Communication / Visualization

The analysis isn’t complete when we get a result from the analysis we need to communicate that result. Looking at the typical communication process we have a sender (the analyst) with a message (the result, meaning or “story” in the data). The sender encodes the message into a format to be sent through some medium or channel. Sending the message through email may create a much different message than through an interactive presentation. The recipient (the audience we want to communicate to) will get the message from the channel and decode it, attempting to reconstruct the meaning from the sender. It doesn’t really matter if the channel is a paragraph of text, a table of numbers or a complex visualization, the important aspect of this stage of data analysis is a successful communication. This stage benefits from pulling experience from behavioral psychology, economics and some cognitive science (especially when it comes to visualizations). But the important thing here is that we don’t focus on simply making the coolest dataviz here, we want to focus on successfully communicating the data. If we focus on the communication and it means an intricate interactive visualization, let’s dig into that. However, if we can summarize our analysis with a one-sentence email of “The employees do not spend too much time dealing with spam.” then that’s good too. Focus on creating the message that represents the data and put it in a format the recipient will decode accurately.

Exploratory Data Analysis

There is another approach to data analysis that is quite useful (and quite common), but can easily be misapplied and misunderstood. In this method we start with data and jump into it, digging around and seeking answers without any specific research question in mind. We do this to understand the data, uncover any underlying structure in the data, or detect anomalies, outliers or important variables (among other tasks). And the important thing about this process is that we still want to circle back around to the research question. This is why we presented the traditional approach first, we will always end up back at the research question.

John Tukey, in addition to coining the term “bit” and creating the box plot, is often attributed as the pioneer of exploratory data analysis. He describes it as “actively incisive rather than passively descriptive, with real emphasis on the discovery of the unexpected.” He also wrote, “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.” The process emphasizes descriptive statistics and data visualization as methods to see into the data and distributions.

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)
5. Paul Slovic wrote in a 2001 paper called “The risk game” in which he defines risk as “a game in which the rules must be socially negotiated within the context of a speciﬁc problem.” When we call out “technical risk” we are simply acknowledging the limited scope of assessments targeting frequency and impact. [↑](#footnote-ref-5)