Fundamentals of Data Science

Stephyn G. W. Butcher, Ph. D.

21 January 2023

Table of contents

Preface

I once had a student ask, "why do you teach linear regression? I understand the historical relevance but nobody uses that today, right?" Another student asked, "what does probability have to do with data science?" These questions are related because underlying them is a fundamental misunderstanding about what data science is. Everybody who is anybody wants to do Deep Learning on TensorFlow. But that isn't data science.

Data science is about data and science, it is not about algorithms. Specifically, it is the application of the methods of science to data to improve organization outcomes (usually in a business to increase business value: increase revenue, decrease costs, or create an asset).

And this is the problem with assigning an O'Reilly book as a textbook. It may teach you about library X and algorithms A, B, and C but it won't teach you principles that you can apply with any library and any algorithm. In fact, the principles should be able to tell you whether or not the problem is even a data science problem at all, and why you should start with linear regression, even though it is over two hundred years old at this point.

These course notes are an attempt, developed over the last few years, to teach you exactly that.

We start with the **Data Science Process** which is grounded in the Scientific Method. It emphasizes identifying appropriate problems (not all business problems are data science problems…if you have a crappy user experience or horrible customer service, "AI" will not save your company). We talk about a framework, "Context - Need - Outcome - Vision" for *communication* with stakeholders. Communication is actually the most important skill for a data scientist. The best thing you can do at this stage is to have a well-defined problem or question, a hypothesis, and a measure of success.

The next chapters will focus on some foundational tools. This starts with **Systems Theory** as a way of framing domain knowledge. Domain knowledge is critical to being a successful data scientist. If you want to solve your organization's problems, you have to understand how the organization understands its world, what things are, how do things work? System Theory gives you a general tool for framing domain knowledge as a *process* in which the organization finds itself.

Next, we move to **Probability**. We may understand what the process does in a qualitative sense, but in a quantitative sense, we don't always get the same result. For example, not everyone that puts something in their shopping cart in an online store actually buys it. The outcome of this process is thus *uncertain*. Probability is our tool for thinking about and modeling uncertainty.

You can think of it this way. The context of any given business problem is like a building. We know, qualitatively, what goes on inside the building. That's our domain knowledge (some of it may be wrong!). Data is like windows in the building. But there aren't always windows where we want them...and the lights aren't always on...and sometimes the blinds are half closed. When we look in the windows, we don't see everything that's going on inside that building, it's a partial picture *and we don't always see the same thing through the same window*. That's uncertainty. Now you have to take that partial, changing, imperfect information and make business decisions with it. How can you do this in the most skillful way?

The next brick in our foundation is **Visualization**. At least since the 70s, there have been some solid experimental results in how humans interpret these scratches and squiggles. We will concentrate on static charts and tables because those are the most common visualization. They find their way into emails, screenshares on Zoom, PDF reports, and PowerPoint slidedecks.

The chapter on **Data** introduces a grab bag of topics. We will talk about getting and cleaning data, databases, and data types, to name a few of those topics. The typical data scientist spends more than 80% of their time on *ETL* (extract, transform, and load). Here we differentiate between "ETL in Large", which results in a *data warehouse* and "ETL in the Small", which results in *tidy data*, ready for analysis. This chapter could easily expand into its own course and this set of notes so we will only cover the basics.

With **Exploratory Data Analysis** (EDA), we start to delve into analytics. EDA is often criticized as "just playing with data". We avoid this trap by using our business problem as a guide and following a rigorous framework for EDA. EDA is followed by **Mathematical Distributions**. We start talking about the simplest of models, *Null* or *Constant* models that don't use any features. We also show how we can model the distribution of a data and use that distribution to ask interesting questions about our problem domain. You can consider this chapter to introduce "Level I" modeling.

Finally, we start to talk about **Statistical Inference**. We take the Bayesian approach to statistical inference because it works the way people generally think statistical inference works. If EDA results in a description or prediction of our (business) process, statistical inference tells us how much we should believe that result, given an uncertain world.

With the conclusion of this chapter, we have finished a sort of "Level I" data science. It is often surprising to people to learn that this is all that many businesses require. They do not need deep learning. Instead, they need someone to frame relevant business questions, to identify the proper data to answer that question, and to create a description or prediction that answers the question, and determine how believable it is.

In the next three chapters, we develop **Linear Models** such as linear regression and logistic regression. Both of these models are regularly included in the lists of algorithms that data scientists should know; they are not merely historical curiosities. They're simple and effective models that are easy to build and interpret. Unless there are strong contraindications, start with these. With the introduction of more sophisticated models, we start to talk about **Model Evaluation**. This concludes “Level II”.

Having spent a lot of time talking about algorithms, we return to the heart of data science…science, and discuss some common analytical pitfalls. These pitfalls go by various names and the chapter is called **Effects, Biases, and Paradoxes**. Examples include the Simpson Paradox where an effect exists in the aggregate but not among the parts. The most famous example of this paradox arose in a court case against UC Berkeley. In the aggregate, it appeared that UC Berkeley discriminated against women. However, when you looked at the results by academic department, they did not.

This is a lot of material to cover in a semester and we will not do it justice. We may not cover your favor topic enough or at all. We may not cover what you are exactly doing in your data job right now. However, we will develop a solid foundation that will serve you well in the coming years, regardless of the vicissitudes of libraries or particular machine learning algorithms.

One final important note… These course notes are generated from Juypter Notebooks for both computation and *math*. Where relevant (obviously not in the Preface!), you should be running the corresponding Jupyter Notebook and executing the code. Data science is not about code (it’s about communication) but in those cases where seeing the code run will be helpful (or where making changes to the code would be informative), you should do so. Additionally, because these notes include a lot of code (and sometimes data as well), don’t be incredibly alarmed at the size of the book.

About the Author

Stephyn Butcher is currently Principal Software Engineer for Data Science, Machine Learning, and Artificial Intelligence at Gerson Lehrman Group. He has worked previously as “Data Chef” (PXY Data), Data Products Engineer (Appriss Safety), Lead Data Scientist (Clubhouse Software), Data Products Tech Lead (LivingSocial), and Software Engineer/Scientist (Mercury Analytics) and has over 20 years of experience in technology. Before embarking on a career in tech, he was an economist and expert witness specializing in labor relations. Stephyn holds a BA in Economics (California State University, Sacramento), MA in Economics (The American University, Washington, DC), MS in Computer Science (Johns Hopkins University, Baltimore, Maryland), and Ph. D. in Computer Science (Johns Hopkins University, Baltimore, Maryland). He is also an ordained priest in the Katagiri roshi lineage of the Soto school of Zen Buddhism.

Acknowledgments

- Stephyn Butcher's data science students over the semesters: Fall 2013, Fall 2014, Fall 2015, Fall 2016, Spring 2017, Fall 2017, Spring 2018, Fall 2018, Spring 2019, Fall 2019, Spring 2020, Summer 2020, Fall 2020, Spring 2021, Summer 2021, Fall 2021, Spring 2022, Summer 2022, Fall 2022.
- Spotters of Errata

Copyright

1 Introduction

> Vizzini: “Inconceivable!”
>
> Montoya: “You keep using that word. I do not think it means what you think it means.” *The Princess Bride* (1987)

In this chapter we try to get our bearings. The term *data science* is bandied about quite a bit but it’s still not clear what it actually is and how it differs from anything else: is it same or different from machine learning, statistics, data analysis? Is that even the right question? And what is a *data scientist*? Is that merely someone who does data science? If so, then we really need to figure out what data science is or our definitions are circular. And what about *data analyst*, *machine learning engineer*, *data engineer*? How do those jobs differ from data scientist? How are they the same? And what is *business intelligence*?

1.1 Differing Definitions

Let’s look at a few definitions circulating on the Internet. The big data site “Big Data Made Simple” published a list of 14 definitions some time ago. (As a sign of the times the site has has been rebranded itself to “Slaves to the Algo”, an artificial intelligence site. We’ll address both *big data* and *artificial intelligence* later.)

1. “There’s a joke running around on Twitter that the definition of a data scientist is ’a data analyst who lives in California,” — Malcolm Chisholm

This definition is funny and like everything funny, it has a kernel of truth. It’s not exactly clear what the difference between a data analyst and a data scientist might be. One suggestion has been that data analysts’ work is retrospective while data scientists’ work is prospective. That is, data analysts report about the past (“sales were up 10% last quarter”) and data scientists predict the future (“sales will be up 10% next quarter”). This seems kind of reasonable until one looks at the details: data scientists definitely report and data analysts have always forecasted.

2. “A data scientist is that unique blend of skills that can both unlock the insights of data and tell a fantastic story via the data,” — DJ Patil

This quote is interesting because DJ Patil and Jeff Hammerbacher are credited with having coined the term “data scientist”. Additionally, DJ Patil served as the first Chief Data Scientist of the United States (2015-2017). The quote includes some interesting ideas that are often highlighted in discussions about data science: *insights* and *story*.

What’s an insight? An insight is something we didn’t understand before, or something we didn’t understand completely or correctly. We thought one thing and another thing turned out to be true. We thought most of our business came from the South, during the Summer, but it actually comes mostly from the North, during the Winter. Additionally, the insight must be conveyed clearly (maybe not always “fantastically”)…let us just say that *communication* is key. You can discover as many insights as you like (if you discover any at all) but if you can’t communicate your discoveries to others, what good are they?

> 3. “Data scientists are involved with gathering data, massaging it into a tractable form, making it tell its story, and presenting that story to others,” — Mike Loukides

This quote hits two additional themes: *gathering data* and *massaging data*. By “massaging” we don’t mean something nefarious (ie, not “cooking the books”). Instead, we are turning unstructured and ill-structured data into **tidy data** that we can correctly analyze (“making it tell its story”). We see echoes of other themes as well.

> 4. “A data scientist is a rare hybrid, a computer scientist with the programming abilities to build software to scrape, combine, and manage data from a variety of sources and a statistician who knows how to derive insights from the information within. S/he combines the skills to create new prototypes with the creativity and thoroughness to ask and answer the deepest questions about the data and what secrets it holds,” — Jake Porway

This quote reiterates a common idea: a data scientist is a cross between a programmer and statistician. As a result, a data scientist can quickly build *data products* and *generate insights*. This seemed true at the start (a mere 10 years or so ago). Jeff Hammerbacher’s team at Facebook included Ph. D. (“real”) scientists from diverse fields (one was an oceanographer) and software engineers. “Data scientist” was the generic term that was used to describe these people who were often doing different things. A scientist (geologist) might suggest some mathematical concept and a software engineer might develop code that could implement it.

This doesn’t seem completely true anymore for a variety of reasons. First, the “field” has matured and there is now a deeper division of labor. We have data scientists but we also have data engineers and machine learning engineers as well as data analysts and “BI” (business intelligence). Data engineers are generally responsible for building and maintaining data infrastructures (“data pipelines”). Machine learning engineers are generally responsible for data products. However, it might be better to think of these as data engineering and machine learning and note that some people (“data engineers”) will do *only* that and others, especially at smaller or non-tech companies, will have to do it all. They might have any job title (mine is “Principal Software Engineer”).

5. Data scientists are “analytically-minded, statistically and mathematically sophisticated data engineers who can infer insights into business and other complex systems out of large quantities of data,” — Steve Hillion

There’s a lot to like here. It repeats some themes (“analytically mind”) and adds some new ones, with a special emphasis on *business* systems.

6. “A data scientist is someone who blends, math, algorithms, and an understanding of human behavior with the ability to hack systems together to get answers to interesting human questions from data,” — Hilary Mason

This definition is interesting because it’s the first to sort of bring something in from the left hand side of the equation that isn’t data. The other definitions start with data. A data scientist gathers data, massages data, analyzes data, interprets data, reports on data. Here, data is not the starting point. We start with “understanding human behavior” and we end up with “answers to interesting *human* questions” and data, math, and algorithms are the means by which this is accomplished. This shows promise.

7. Data scientist is a “change agent.” “A data scientists is part digital trendspotter and part storyteller stitching various pieces of information together.” — Anjul Bhambhri

This is garbage, hype, hot trash. There was a time when some thought data scientists were the broad defenders and discoverers of truth in the organization. Some even suggested that they should not be in Product or Engineering but Finance where they would be “protected” from the political shenigans common to all organizations. The problem, of course, is that Finance is not immune to such political shenigans and, in fact, is often the source of them. As we will see, Finance might be the worst place to home a data science team.

8. “The definition of “data scientist” could be broadened to cover almost everyone who works with data in an organization. At the most basic level, you are a data scientist if you have the analytical skills and the tools to ‘get’ data, manipulate it and make decisions with it.” — Pat Hanrahan

And this definition hits the nail on the head…don’t all scientists use data to make descisions? Doesn’t everyone who needs to use data to answer questions and support decisions do “data science”? Hmm. I do think there is a difference between a data scientist, a geologist, and a CFO with an Excel spreadsheet. But it doesn’t mean there aren’t also things that they have in common.

9. “By definition all scientists are data scientists. In my opinion, they are half hacker, half analyst, they use data to build products and find insights. It’s Columbus meet Columbo – starry eyed explorers and skeptical detectives.” — Monica Rogati.

Same as above. Isn’t this what all scientists do? However, there seem to be some additional trends here. First, we have “real” scientists (geologists, climate scientists, ecologists) who are studying “data science” to be better geologists, climate scientists, and ecologists. Second, we have geologists, climate scientists, and ecologists (and political scientists, oceanographers), leaving their fields to become “data scientists” in Silicon Valley. Why? What’s going on here?

With regard to the first, there seems to be a general idea that if you want to use applied machine learning in biology (for example), you need to learn data science because that’s where such skills are taught. With regard to the second, the salaries for data scientists are simply larger than those for most geologists, ecologists, and political scientists.

10. “A data scientist is someone who can obtain, scrub, explore, model and interpret data, blending hacking, statistics and machine learning. Data scientists not only are adept at working with data, but appreciate data itself as a first-class product.” — Daniel Tunkelang

We have echoes of other definitions here. What does it mean for data to be a “first-class product”?

I think this falls mainly under the rubric of data engineering. Data is a “first class” *product* in an organization if its storage, manipulation, provenance, and access *for the purposes of analytics* is not an after-thought. Still, there are many data scientists who work at organizations (especially non-tech organizations) where data is not first class in this sense. That is, the data is still viewed as *application data* and access is often tricky and or painful. This may still be true even if organizations have dashboards.

11. An ideal data scientist is “someone who has the both the engineering skills to acquire and manage large data sets, and also has the statistician’s skills to extract value from the large data sets and present that data to a large audience.” — John Rauser

Here we have recurring themes: gathering data, massaging data, interpreting data and reporting insights.

12. Data scientist is “someone who can bridge the raw data and the analysis – and make it accessible. It’s a democratising role; by bringing the data to the people, you make the world just a little bit better,” — Simon Rogers

This is an interesting twist on the idea of "democratizing" data. Usually, "democratizing" data means making data available to everyone in an organization so that they can answer their own questions instead of submitting requests to the data science team. Here, the spin is that data scientists are the ones doing the democratization…not of the raw data, *but of the meaning* or…insights.

Like most things, there is a spectrum here. You can easily find YouTube videos that show you how to repair a broken faucet. However, you might not be able to quickly identify that the faucet is irreparable. And you might not want install the plumbing in an addition or new house. Plumbers are specialists.

This continuum exists everywhere. At one end, we have some basic budgeting in Excel and at the other end we have the financial admistration of a multinational corporation. At one end, we have first aid and at the other end we have brain surgery or a quadruple bypass. So, yes, everyone should be data literate (like first aid)…but, in my experience, people without adequate training in statistics can cause more harm than good. Data science encompasses a great deal of specialized knowledge…specialized knowledge that this text aims to provide.

The Dunning-Kruger Effect is especially relevant here: people without training cannot judge exactly how insufficient their knowledge actually is. Or, as Neil de Grasse Tysons put it, it's possible to know enough about a subject to think you're right but not enough to know you're wrong.

13. "A data scientist is an engineer who employs the scientific method and applies data-discovery tools to find new insights in data. The scientific method—the formulation of a hypothesis, the testing, the careful design of experiments, the verification by others—is something they take from their knowledge of statistics and their training in scientific disciplines. The application (and tweaking) of tools comes from their engineering, or more specifically, computer science and programming background. The best data scientists are product and process innovators and sometimes, developers of new data-discovery tools," — Gil Press

This definition brings in something new: *the scientific method* and knowledge about the scientific process. With this knowledge we understand that we need to know if we have *experimental* data or *observational* data and that this makes a huge difference in what we're trying to do and limits what we can learn or conclude.

14. "A data scientist represents an evolution from the business or data analyst role. The formal training is similar, with a solid foundation typically in computer science and applications, modeling, statistics, analytics and math. What sets the data scientist apart is strong business acumen, coupled with the ability to communicate findings to both business and IT leaders in a way that can influence how an organization approaches a business challenge. Good data

> scientists will not just address business problems, they will pick the right problems that have the most value to the organization," — IBM researchers

This definition recognizes that data scientist is an evolution of the data analyst role, ie., they are on a continuum. It outlines the broad training...the kind of training you're engaged in right now (or will soon be): programming, statistics, analytics, modeling. But it adds one more element to the list: *business acumen*. And it finishes with the punch line:

> Good data scientists will not just address business problems, they will pick the right problems that have the most value to the organization

1. picking the right *business* problems.
2. adding the most *value* to the organization.

1.2 Better Call Venn

One of the first attempts to describe "Data Science" originated with Drew Conway and his Venn Diagram of Data Science. In many ways, it's still a good place to start any discussion about Data Science.

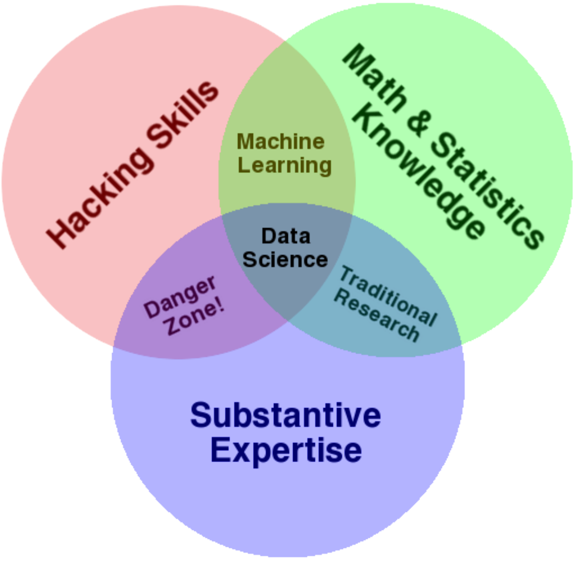


Figure 1.1: Drew Conway's Venn Diagram of Data Science

In Conway's model, there are three areas that combine to form Data Science: programming, math & statistics and substantive expertise.

How does Conway describe these?

Let's look first at "hacking skills":

> For better or worse, data is a commodity traded electronically; therefore, in order to be in this market you need to speak hacker. This, however, does not require a background in computer science—in fact—many of the most impressive hackers I have met never took a single CS course. Being able to manipulate text files at the command-line, understanding vectorized operations, thinking algorithmically; these are the hacking skills that make for a successful data hacker.

While it is true that there may be impressive hackers who have never taken a single CS course, I have also seen them reinvent the wheel (this is actually a recurring theme in Silicon Valley and the "Digital Native" Culture it has spawned…if it's not on InstaFaceTwitTokApp, did it really happen?). There is a sort of strange double standard here. Data scientists are saying "no, you need a professional" while at the same time looking at CS problems and saying "I don't need any training for that!". Learning about data structures and algorithms (especially graph and string algorithms) is incredibly important to the practicing data scientist.

Beyond that, however, you simply need to know how to write good code. At a minimum, good code is well organized across files, uses good naming conventions, and is readable. It has been tested and has error handling. It is under version control. The days of one off scripts than run from a developer's laptop are, hopefully, over.

This is not to say that data in different places, in different formats, of different quality, doesn't sometimes require inventive solutions. As with anything else, it is a continuum. But it's really just a hop, skip, and a jump from a script meant to be run once on your laptop to one that's run every ten minutes on a production server.

Servers world-wide are littered with scripts that were never meant to make it into production. (And, I kid you not, at one company where I worked, we had production code that ran from someone's laptop). This makes such "personal" or "one off" scripts a slippery slope. A one off becomes scripted, gets attached to `cron` and then goes into production all without error handling or tests. You can think of exploratory data analysis as a version of "test first" applied to data, so you should do the same thing with your code: if you can find a simple way to test your code, do so. There's nothing quite like having to explain to the CEO that the new campaign about to launch is misguided because there was a "bug" in the script (I don't think this issue gets enough attention). I think it's also true that the shift towards roles such as "machine learning *engineer*" and "data *engineer*" are indications that companies want people who know how to program well. Of course, the reverse is still true…people hire "data scientists" when they just want software engineers with some analytic acumen.

ℹ Note

The issue of programming *language* often rears its ugly head in these discussions. Specifically, what language should data scientists on a team use? Should they all use

the same language? The main data science programming languages are Python, R, and Julia. Perl is sometimes used as well, especially by data scientists coming from academia. This doesn't mean you can't use other languages…these are the most common/popular.
One school of thought is that each data scientist should use the language with which they are the most comfortable. The other–and the one with which I am in agreement–is that everyone on the team should use the same language.
The rationale for the first view seems reasonable: we want every data scientist to be as productive as possible, right away, and that means they can use the language they're comfortable with. But it is short sighted. What happens when (not "if") that one person who loved Perl leaves 20 Perl scripts behind when they leave for a better job? What happens if a critical analysis is done in R but that data scientist goes on parental leave? So, although this text will use Python, you should be prepared to use whatever language your team uses.
There is a second dimension to this question as well, what programming language gets used in *production* but we will have that discussion later.

Conway continues:

> Once you have acquired and cleaned the data, the next step is to actually extract insight from it. In order to do this, you need to apply appropriate math and statistics methods, which requires at least a baseline familiarity with these tools. This is not to say that a PhD in statistics in required to be a competent data scientist, but it does require knowing what an ordinary least squares regression is and how to interpret it.

There's more to it than that, but we get his drift. This "more" includes linear algebra, optimization, probability, and statistics. There isn't otherwise a special "data science math" (at least to my mind; I bring this up only because I've been asked about "data science math" by past students; there's just math and some of it is very useful for data science). However, I can safely say that when you're done with this text, you will know what ordinary least squares regression is and how to interpret it. And this is a lot more important than knowing that, "OLS is BLUE".

ℹ Note

If you don't understand the statement above yet, that's fine. "OLS is BLUE" very often the first thing one proves in a Math/Stats or Econometrics course. It stands for, Ordinary Least Squares (regression) is the Best Linear Unbiased Estimator". That's nice (maybe) but it doesn't tell you how to build a regression model.

Next, Conway describes "Substantative Expertise":

> In the third critical piece—substance—is where my thoughts on data science diverge from most of what has already been written on the topic. To me, data plus math and statistics only gets you machine learning, which is great if that is what you are interested in, but not if you are doing data science. Science is about discovery and building knowledge, which requires some motivating questions about the world and hypotheses that can be brought to data and tested with statistical methods. On the flip-side, substantive expertise plus math and statistics knowledge is where most traditional researcher falls. Doctoral level researchers spend most of their time acquiring expertise in these areas, but very little time learning about technology. Part of this is the culture of academia, which does not reward researchers for understanding technology. That said, I have met many young academics and graduate students that are eager to bucking that tradition.

For many years, I interpreted "substantive expertise" to mean *domain knowledge* and after a recent re-reading of this short blog post, I have changed my mind and, to a certain extent, this becomes where I take issue with the Data Science Venn Diagram.

What is *actually* included in substantive expertise is important. We can add this,

> Science is about discovery and building knowledge, which requires some motivating questions about the world and hypotheses that can be brought to data and tested with statistical methods.

to our growing list of important points. But I don't know if it's where Conway was situated in the sciences (he was a political scientist) or where he went to school but many of the sciences have always *been* technologically advanced. Studying economics in the 90s, I used SPSS and SAS to gather, massage, and analyze data. I had physicist friends who wrote detailed simulations in Fortran during the same period. Maybe there's just a limit to argument by Venn diagram.

1.3 What about Statistics?

Remember the definition of data science above?

> 10. "A data scientist is someone who can obtain, scrub, explore, model and interpret data, blending hacking, statistics and machine learning. Data scientists not only are adept at working with data, but appreciate data itself as a first-class product." — Daniel Tunkelang

Here's the definition of *statistics* from the Oxford Dictionary of Statistical Terms:

> Statistics is the discipline that concerns the collection, organization, analysis, interpretation, and presentation of data.

Hmm. That sounds awfully familiar.

The economist Hal Varian said,

> I keep saying the sexy job in the next ten years will be statisticians. People think I'm joking, but who would've guessed that computer engineers would've been the sexy job of the 1990s?

And he goes on to describe what he means by "statistician":

> The ability to take data – to be able to understand it, to process it, to extract value from it, to visualize it, to communicate[. I]t's going to be a hugely important skill in the next decades, not only at the professional level but even at the educational level for elementary school kids, for high school kids, for college kids. Because now we really do have essentially free and ubiquitous data. So the complimentary scarce factor is the ability to understand that data and extract value from it. I think statisticians are part of it, but it's just a part. You also want to be able to visualize the data, communicate the data, and utilize it effectively. But I do think those skills – of being able to access, understand, and communicate the insights you get from data analysis – are going to be extremely important. Managers need to be able to access and understand the data themselves. — Hal Varian, Flowing Data

So why are we talking about "data scientist" and not "statistician"? Would any of you be interested in "data science" if it was called "statistics"? You wouldn't be the first person to wonder why statistics isn't the "science of data" and why we have this new term, "data science". In fact, some people claim there is no difference. Nate Silver–a very famous Bayesian statistician–says that data science *is* just statistics. (By the way, 10 years is up but I would argue that's a good thing).

I think Allen Downey is a bit closer to the truth:

> Having read The Theory That Would Not Die and The Lady Tasting Tea recently, I suggest the following conjecture:
>
> The term "data scientist" has been created to describe what people want from a statistician, but which many statisticians fail to provide because statistics, as a field, spent too much time in the 20th century on problems in philosophy of science, and theoretical mathematical problems, and not enough time on practical applications and the use of computation to analyze data. As a result, many graduates from statistics programs have a set of skills that is not a good match for what the market wants them to do. This market vacuum is being filled by "data science." That's my theory, which is mine. — Allen Downey

This divide between "data scientist" and "statistician" is also related to the "Two Cultures" theory of machine learning and statistics. Simon Blomberg said,

> machine learning is statistics minus any checking of models and assumptions – Brian D. Ripley Two Cultures

To which Bayesian statistician Andrew Gelman responded,

> In that case, maybe we should get rid of checking of models and assumptions more often. Then maybe we'd be able to solve some of the problems that the machine learning people can solve but we can't!

Although "machine learning" as such grew out of computer science, many of the same principles and algorithms are simply called "pattern recognition" in other engineering specialties. In fact, Bishop's textbook on machine learning is called Machine Learning and Pattern Recognition to cater to both audiences.

1.3.1 Business Statistics

If you listen to the hype, organizations *need* data science today or they're going to fall hopelessly behind their competitors. There's tons of data and they need data scientists to uncover the business value and actionable insights in that data in order to gain–or maintain–a competitive advantage. But, honestly, companies have had data and analysts to pour through that data for decades. Also, if you don't have data *engineers* to make the data available (or data scientists with data engineering training), your data scientists are going to have nothing to do but twiddle their thumbs.

If we take a step back and look at the history of Business Informatics ("BI"), the application of statistics to business processes is not new. A business statistics course is often required of business majors in universities worldwide.

Here's a definition of Business Statistics:

> Business statistics takes the data analysis tools from **elementary statistics** and applies them to business. For example, estimating the probability of a defect coming off a factory line, or seeing where sales are headed in the future. Many of the tools used in business statistics are built on ones you've probably already come across in basic math: mean, mode and median, bar graphs and the bell curve, and basic probability. Hypothesis testing (where you test out an idea) and regression analysis (fitting data to an equation) builds on this foundation.
>
> Basically, the course is going to be practically identical to an elementary statistics course. There will be slight differences. The questions will have a business feel, as opposed to questions about medicine, social sciences or other non-business subjects. Data samples will likely be business-oriented. Some subjects usually found in a basic stats course (like multiple regression) might be downplayed or omitted entirely in favor of more analysis of business data.

I think the only difference between the "business statistics" described here and data science is the breadth and depth of tools available to data scientists. Data scientists have the ability to go beyond elementary statistics and use complex statistical and machine learning models. But this is a difference of degree, not kind. As we shall see, most (possibly all) machine learning algorithms are fundamentally the same as something as simple as a measure of central tendency like the mean. It's better to think of statistics and machine learning as a continuum of modeling choices, from simple to complex.

So don't look down your nose at "elementary" statistics. Simple models are almost always the best starting place whenever you look to solve a business problems or answer business questions. It is an unfortunate and overlooked truth by those wishing to use the latest, shiny algorithms that most businesses can get by with models based on means, sums, and rates. The *hard* part is the data: getting it and organizing it, understanding the processes behind it.

Business statistics is where statistics started! The application of elementary statistics to good decision making began well over a hundred years ago. In fact, the "theorical" statisticians of the early 20th century were actually trying to solve practical problems in industry. Most famously, Gosset used statistics to improve the brewing process at Guinness in the early 1900's. In order to do so, he developed an entire theory around working with small samples and published them under the pseudonym, "A. Student" (and we got "Student's t-distribution as a result). R. A. Fisher, a famous statistician and contemporary of Gosset's, spent part of his career analyzing data from agricultural field trials to determine the best fertilizer. Why wasn't this"data science"? Actually, I think it was…and there is an implicit echo here of Downey's observation: statistics lost its way.

Let's look at a more modern example, McDonald's:

> When you're as large as we are, we can't run the business on simple gut instinct. We rely heavily on all kinds of statistical data to help us determine whether our products are meeting customer expectations, when products need to be updated, and much more. The cost of making an educated guess is simply too great a risk.
> – Wade Thomas in Business Statistics: A Decision-Making Approach

That sounds like data science to me.

And if we google the term "data science for X" where X is "fertilizer analysis" we find Monsanto has far surpassed Fisher's work, when X is beer, we have Empirical Brewery Brews with (Data) Science, and when X is food products, we get, well, McDonald's in 7 Uses of Big Data in Food and Beverages Industry.

It's important to note that we don't mean to imply that data science is only for *for-profit* businesses. But even today, non-profits need to operate like businesses. They have a mission, they have a product or service. It needs to meet the needs of the population the organization serves. The organization needs to attract donors, sponsors, patrons, and volunteers. The donors need to feel like they're getting something out of their donations. That is, non-profits (and governments) have measurable goals and problems that *may* be solvable with data and computation, just as for-profits do.

1.4 Defining Data Science

Why all the hand wringing? If we're going to spend time learning how to do something, I think we should probably have general agreement about what that something is. Additionally, anyone trying to find a job as a data scientist or doing data science will quickly be confronted by a mish-mash of job descriptions, some of which have only the most tenuous relationship to each other. I have long since given up the idea that "data science is what data scientists do" because most organizations don't know what data scientists should reasonably be expected to do and a data scientist may end up doing more than data science.

Here's a real example that takes this hyperbole to the extreme:

> [Company] is looking for a Data Scientist eager to use advanced analytical, machine learning and data transformation techniques as a means to develop practical tools and analyses that can help solve complex business problems; transforming volumes of data into actionable information. In some instances, you will be using languages like R or Python to employ newer techniques like tree ensembles, neural nets, and clustering to solve long- standing questions. **In other instances, it will be your responsibility to come up with solutions for problems that have not yet been identified.** In order to do so, the Data Scientist will have to be confident that they can solve a wide- range of problems using a variety of techniques—some known, some new, some yet to be created.

My favorite bit is the part I've emphasized in boldface: basically, you must solve unidentified problems. How is that even possible?

Foster Provost in Data Science for Business defines *data science* as fundamental concepts and *data mining* as the tools. He quite correctly states that data science is not tools any more than biology is about test tubes. A data science text that focuses on Pandas, Spark, Scikit-Learn, and MongoDB misses the point entirely (although you still need to learn how to use those "test tubes"…and the test tubes change!). Here is a sampling of four of his 12 concepts:

- Extracting useful knowledge from data to solve business problems can be treated systematically by following a process with reasonably well-defined stages.
- From a large mass of data, information technology can be used to find informative descriptive attributes of entities of interest.
- If you look too hard at a set of data, you will find something – but it might not generalize beyond the data you're looking at.
- Formulating data mining solutions and evaluating the results involves thinking carefully about the context in which they are used.

There are about a dozen or so in his book. Still, I'm not sure that 12 maxims truly constitute "data science" but they might certainly inform the discussion.

1.5 Centrality of Domain Knowledge

Domain knowledge has had a very strange relationship to data science. Back in the Old Days (tm), at data science conferences like STRATA, there would be roundtables entitled, "Is Domain Knowledge Necessary?". There would always be someone who had done well on a Kaggle competition who would stand up and say, "I solved the Breast Cancer Problem with a Support Vector Machine model and I don't know anything about breast cancer. Therefore, domain knowledge is not necessary.". What they really meant to say was, "if someone hands you a CSV file and tells you the problem, domain knowledge isn't necessary". As we saw from the different definitions of data science above, *gathering and massaging the data* was a recurring theme. If you don't have domain knowledge, how do you know what data to gather? How do you know to massage it into the right form? So we know at least *that* much can't be true.

If we comb the definitions above, however, we that there is something else lurking in the background:

> unlock the insights of data
>
> derive insights from the information within [the data].
>
> infer insights into business and other complex systems out of large quantities of data.
>
> they use data to … find insights.
>
> interpret data
>
> extract value from large data sets
>
> find new insights in data

We even see it in Conway's Venn Diagram:

> extract insight from [data].

It's in Varian's quote as well:

> understand [data]. … access, understand, and communicate insights you get from data analysis.

It's not the *insights*, although that's important, it's an unspoken assumption that *the insights are in the data*. In short, there is a view that *data is sufficient*.

This view is symptomatic of a larger problem and I think it gets to the crux of the matter. There is a rather popular misception that *the data can speak for itself*. This is wrong. The data cannot possibly speak for itself because data always exists in context.

In the "real" sciences, domain knowledge consists of the theories, concepts, and rules of thumb that are generally accepted by practitioners. Acquiring this domain knowledge–along with the ability to add to it–is generally what it means to become a physicist, a biologist or an environmental ecologist and then *do* physics, biology, or ecology. At the fringes, this knowledge is the object of active research but it generally consists of the things everybody "knows". This domain knowledge gives shape to the creation of new questions, new hypotheses, the design of experiments and the interpretation of results. New evidence can cause a revision in this domain knowledge.

Unfortunately, even in the "regular" sciences, scientists who put method (statistics) ahead of domain knowledge end up in trouble. In 2005, Ioannidis published the watershed paper, "Why Most Published Research Findings Are False", wherein he detailed results showing that many (perhaps not really "most") research findings fail to replicate. The publication of the paper has lead to the identification of a general Replication Crisis that appears to be plaguing all sciences. The Replication Crisis has lead many to suggest that domain knowledge needs to be put back in the driver's seat. This is neatly summarized by Richard McElreath's slogan,

> Science before Statistics.

This contrasts with the view that data scientists do not need domain knowledge at all. They can just be get the data, apply a machine learning algorithm to it, and that's that because they have feature selection algorithms. But we're discovering that it simply doesn't work that way. Those models end up failing in production in spectacular ways or, even if they succeed, they hide implicit prejudices hidden in the data used to build them.

What both of these problems have in common is a failure to heed the age-old dictum:

> Correlation is not Causation.

and its corrollary:

> Prediction is not Causation, either.

What the Kaggle competitions are missing is the step where the variables in the data were selected as broadly relevant to the problem at hand. And so this means that you don't start out by running correlation analyses for the variables in your data set but instead map out what you think you know about the variables and how they interact. We will talk more about this in later chapters. Therefore, we follow Judea Pearl is holding a strong line against the idea that data science is simply the application of machine learning algorithms to (business?) data.

However, you don't have to start from scratch. Your organization will have a body of theories and concepts, rules of thumb, but covering a much smaller domain than, say, physics. There will be things like "women purchase more than men", "Fridays are always have the lowest sales", and "promocodes increase revenue in the long run". Not all of it will have been collected scientifically–that's where you come in (for example, at least in one case, it turns out that promocodes only increase revenue in the short run, by giving 10% off in February to people who were going to pay full price in March). As a data scientist, you will become an expert on this domain knowledge, verifying some theories and discarding others. A data scientist at Walmart is a "Walmartologist", a data scientist as Apple is an "Apple-ologist" and a data scientist at Starbucks is a "Starbucksologist". The key to the successful practice of data science is the marriage of general skills (communication, data acquisition and analysis, statistics, machine learning, programming) to a specific, often very circumscribed, but small, domain.

Unfortunately, you won't get domain knowledge from this book (although a book on general business principles and marketing concepts won't hurt). It is much more likely, however, that you will get this information from domain experts, stakeholders, and Josephine, who doesn't appear on the org chart, "but has been here forever years and knows everything". However, it's worth noting that I have recently started to see data scientist job postings that require experience in a given field such as human resources or healthcare.

Despite selling a "Data First" perspective for the last ten plus years, O'Reilly has gotten on board as well. In What is Causal Inference?, Hugo Bowne-Anderson and Mike Loukides say,

> One of the most dangerous myths of the past two decades was that the sheer volume of data we have access to renders causality, hypotheses, the scientific method, and even understanding the world obsolete. … In the "big data" limit, we don't need to understand mechanism, causality, or the world itself because the data, the statistics, and the at-scale patterns speak for themselves.

And then the punchline:

> We're coming out of a hallucinatory period when we thought that the data would be enough. It's still a concern how few data scientists think about their data collection methods, telemetry, how their analytical decisions (such as removing rows with missing data) introduce statistical bias, and what their results actually mean about the world. And the siren song of AI tempts us to bake the biases of historical data into our models. We are starting to realize that we need to do better.

In other words, data is a shadowy, often times imperfect reflection of the world, it isn't the world.

Figure 1.2: Data is the reflection not the thing reflected

The purpose of data science is to take existing domain knowledge + data and create new, better, or corrected domain knowledge, "insights".

Okay, that's the purpose, what is the definition?

Max Shron's definition of Data Science in Thinking with Data drives this point home:

> "To me, data plus math and statistics only gets you machine learning, which is great if that is what you are interested in, but not if you are doing data science. Science is about discovery and building knowledge, which requires some motivating questions about the world and hypotheses that can be brought to data and tested with statistical methods."

This is very good quote for several reasons.

1. It may very well be that you are only interested in math and statistics, machine learning. If so, you may find that data science is not your cup of tea.
2. It emphasizes *science*.
3. It emphasizes that you do *not* start with data, a point we started to make above.

So, for Shron, there is this sense of doing actual science, of applying the scientific method, on the problems and data available in an organization, in the context of everyday life. He then goes into his actual definition of data science:

> Data science is the application of math and computers to solve problems that stem from a lack of knowledge, constrained by the small number of people with any interest in the answers.

Data science is an *applied* science. It is not the quest for information for its own sake. Instead, you should always be working on solving a business problem or answering a business question.

If I appear to be belaboring this point, it is because I want to make sure there is no room for misunderstanding. The angel you should keep on your shoulder throughout this book is that data science is not synonymous with machine learning, it is the application of the tools of science to everyday problems. The question should not be, “what is the latest version of this algorithm? Is there a better one?” but “if I pulled a data set off the internet right now, could I apply this method to it?”. So data science is just science with the *priviso* that only a few people have any interest in insights that comprise some company’s understanding of its churn rate as opposed to, say, the general theory of relativity or a cure for cancer (or, in 2021, COVID).

John W. Forman in Data Smart says,

> Data science is the transformation of data using mathematics and statistics into valuable insights, decisions, and products.

which is also good.

However, you don’t start with data, you start with a question or a problem. Otherwise, your data science team will not fare much better than the infamous Underwear Gnomes:

1. Hire data scientist.
2. ???
3. Profit

You start with people, the domain knowledge and their problems and you end with communication and increased understanding. Even in the not so grand exploration of a variable, “daily sales”, Data Science is not *just* the rote execution of code and plotting charts. You should always have a reason for why you did something and document it. Communication is central.

The confusion surrounding data science and the job title “data scientist” has gotten so bad that the Harvard Data Science Initiative has stepped in to define a standard, Toward Foundations for Data Science and Analytics: A Knowledge Framework for Professional Standards. The comic XKCD has this to say about standards:

HDSI’s definition of data science is:

> Using Data to achieve specified goals by designing or applying computational methods for inference or prediction.

I am not sure why Data is capitalized *a la* German but I do take Issue with the Definition for a Number of Reasons:

1. I’m not entirely keen on including “designing” in the definition of data science when “designing” already resides in other fields: statistics, computer science, machine learning. Because many (most?) of these algorithms are data-driven, it’s not clear what that

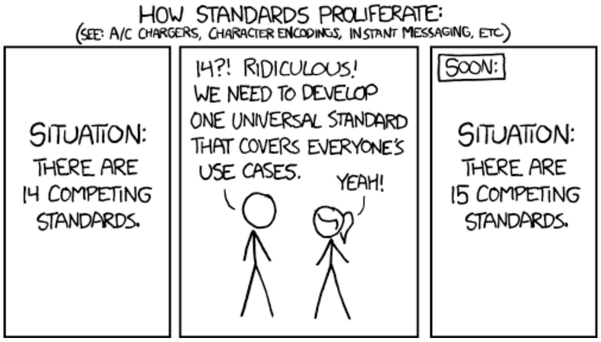

HOW STANDARDS PROLIFERATE:
(SEE: A/C CHARGERS, CHARACTER ENCODINGS, INSTANT MESSAGING, ETC.)
SITUATION: THERE ARE 14 COMPETING STANDARDS.
14?! RIDICULOUS! WE NEED TO DEVELOP ONE UNIVERSAL STANDARD THAT COVERS EVERYONE'S USE CASES.
YEAH!
SOON:
SITUATION: THERE ARE 15 COMPETING STANDARDS.

Figure 1.3: Standards

means. Again, is everything that studies machine learning algorithms “data science”? No.

2. What about those computational methods that don’t use data as usually defined. For example, you might build an expert system to solve a pressing problem. Yes, that expert system includes information about all its rules but they weren’t learned from data. If “data” means all information whatsoever then every algorithm uses data and is thus data science.
3. Not all data science ends up as inference or prediction. There is visualization for understanding and entertainment. Are these not data science?

It’s quite a pickle, isn’t it?

Others have tried to improve on the Venn Diagram, here’s a candidate:

I’m sorry, this isn’t a data scientist, this is a data science *team*.

So what is our working definition of data science? While there are going to be exceptions, I believe the definition with the broadest applicability includes:

1. Communication - the identification of problems and questions in support of a goal, iterative communication on the progress towards reaching that goal, and the communication of the solution or answers. Problems are solved and questions answered because someone wants to achieve something, often decision making, but sometimes entertainment or understanding.
2. Data - problems and questions arise in the context of a real world processes. Data are our observations about those processes and the raw material for our analyses.
3. Modeling - whether it is statistical or machine learning, the central focus of data science is to build models of real world processes to answer questions or solve problems. It is no accident that most statisticians who say data science is just machine learning are *Bayesian* statisticians. Bayesian statistics emphasizes building models of data.

In a sentence, something like,

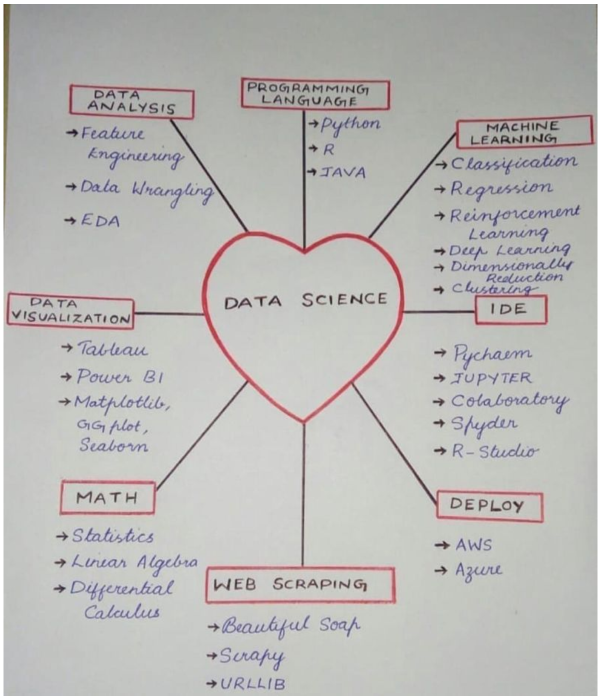


Figure 1.4: Data Science Heart

> "data science is the application of computational modeling to data to solve problems or answer questions in support of a goal (decision making, understanding, entertainment)".

The definition sort of works backwards. The goal is the purpose. Above we stated that the *purpose* of data science is to turn domain knowledge into new, better, or correct domain knowledge. Data science itself involves the application of computational modeling to data. Why? To solve problems with our domain knowledge. Not for its own sake, though, to aid with decision making, understanding, and (possibly) entertainment.

This definition also emphasizes something important: data scientist is a supporting role. It also implicitly includes a subtle recognition that *not everything a data scientist does is data science*. They may *also* do software engineering, management, analytics, artificial intelligence, machine learning, etc.

1.6 Data and Science

At its root, data science is about data and science and not about algorithms *per se*. It's odd how many people get into data science (even "machine learning") who don't actually like data or want to work with data. This is problematic for anyone making the career choice to become a data scientist because automation is coming and the automation is coming first to the machine learning part. We already have some degree of "AutoML" and for some companies it is their entire business model (for example, DataRobot).

Here is one such lament on Twitter:

> One of the biggest failures I see in junior ML/CV engineers is a complete lack of interest in building data sets. While it is boring grunt work I think there is so much to be learned in putting together a dataset. It is like half the problem. – Katherine Scott

and another:

> for my last few ML projects the complexity hasn't been in the modeling or training; it's been in input preprocessing. find myself running out of CPU more than GPU & in one project i'm actually unsure how to optimize the python further (& am considering c++ for one piece) – Matt Kelcey

and yet another:

> Just a personal anecdote, but, in the past 2 years, % of any given project: + that involves ML: 15% + that involves moving, monitoring, and counting data to feed ML: 85%" – Vicki Boykis

and these tweets are coming from very machine learning-centric folks, imagine what data scientists might say.

> ⚠ Warning
>
> If you don't like working with data, cleaning it, parsing it, exploring it–even on questions that don't necessarily interest you deeply on a personal level–data science might not be the field for you.

How does one explain that we are educating data professionals who don't like data? I think there are several reasons for this. First, these efforts don't get much hype or attention. We might know Apple is using such-and-such an algorithm and that they've had a lot of success with it, and Pinterest may make their Github repositories public and that project is so very shiny, and Google may share their latest NLP APIs, but we rarely see all the effort that went into data that made these algorithms work. Neither do we see the actual business questions or results. Because we can only see the "tip of the iceberg", we can only talk about the tip of the iceberg.

At one company I worked at, there was zero interest in data infrastructure or data improvement. As a result, we fought the same data battles across (siloed) teams. And while we could come up with workarounds but the workarounds were almost always project specific and when we started a new project there was always a sense of deja vu. The issue was that while projects could be linked to increases in TPV (total purchase value), data improvement could not. Additionally, the workarounds were "proof" that data improvement wasn't necessary despite the fact they were imperfect and projects fell well short of their potential.

Second, many data scientists come out of computer science and the algorithmic outlook of computer science is not appropriate to data science. Swetta Jha writes,

> Data vs. method centrism: Scientists are data driven, while computer scientists are algorithm driven. Real scientists spend enormous amounts of effort collecting data to answer their question of interest. They invent fancy measuring devices, stay up all night tending to experiments, and devote most of their thinking to how to get the data they need. By contrast, computer scientists obsess about methods: which algorithm is better than which other algorithm, which programming language is best for a job, which program is better than which other program. The details of the data set they are working on seem comparably unexciting.

And this probably hits the nail on the head. In computer science, we fetishize algorithmic comparison and evaluation, very often on standardized data sets without any concern for the data set itself. This is how you get new Ph. D.'s (It worked for me!). But it's not good data science.

1.6.1 Scientific Method

The other piece of the data science puzzle is the science part, specifically, the **Scientific Method**. Data science is ultimately about applying the scientific method to business problems. You may have learned about the Scientific Method in grammar school. There are always those projects where you grow a seed in some blue solution and they walk you through the steps. One of the many versions of those steps is:

1. Question/Problem
2. Hypothesize
3. Predict
4. Test
5. Analyze

In data science, we do this at all levels:

1. A single variable in EDA
2. A pairwise EDA
3. Building a model to answer some question or solve a problem.

and we iterate in order to get better answers. This will be a recurring theme in everything we do throughout the book.

1.7 Big Data

Since we have emphasized data so much, at this point, you might be wondering. What about big data? There so much hype surrounding “Big Data Science”, that Dan Ariely quipped:

> Big data is like teenage sex: everyone talks about it, nobody really knows how to do it, everyone thinks everyone else is doing it, so everyone claims they are doing it... – Dan Ariely

The results in Analyzing the Analyzers seem to back Ariely up. Although “Data Science” and “Big Data” are often spoken of in the same breath and usually just a breathlessly, the survey showed that most data scientists don’t work with anything like “big data” on a regular basis:

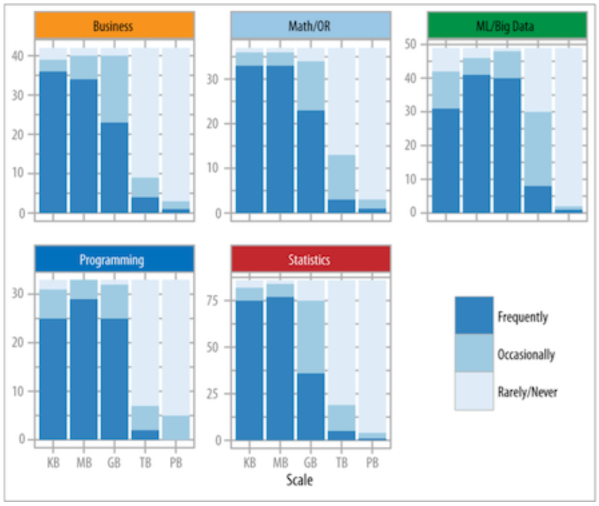


Figure 1.5: Data Set Sizes

Part of the problem here is that the concept of “Big Data” has drifted. And now some people are saying that it’s all just “data” now–there’s no such thing as “big data” and a lot of that has to do with technological improvements.

For a good discussion of the issues, read Don’t use Hadoop - your data isn’t that big by Chris Stucchio. It starts with a funny–but all too common story–of a client that wanted to use Hadoop on 600Mb of data. The upshot: if your data is under 1TB, modern computers (8+ cores, SSD, 16Gb of memory in a *laptop*), databases (even SQLite!), and hard drives (external hard drives come in 4TB models) and Python are more than sufficient for your analytics needs. This doesn’t necessarily hold for your *production* needs.

1.8 Data Science Case Studies

In order to get a feel for Data Science, we're going to present a few case studies in Data Science. As we go through the main chapters of the book, you should come back here and see how much more you understand about what went into these examples of Data Science.

As you read or watch each the following examples of Data Science, take note of anything that interests you. You should also answer the following questions:

1. What question where they trying to answer?
2. How did they approach the question?
3. Where did they get the data and what technique did they use?

1.8.1 Signet Bank

> What can be gained from classification? There are many iconic stories of how forward thinking companies anticipating business issues before they arrive – and then take action. My favorite is story **Signet Bank**, whose credit card division was unprofitable, due to "bad" customer defaults on loans and "good" customers being lost to larger financial institutions who could offer better terms and conditions. The answer, revolutionary at the time, was to apply classification to their customer data. They separated the "Bad" from the "Good", cut the "Bad" ones loose and nurtured the "Good" ones with offers and incentives. Today, we know them as **Capital One**. Data Science Foundations – Classification and Regression

But this isn't the whole story. Signet Bank didn't have data with which to develop a classification algorithm so it had to generate it. How did it generate the data? By offering customers *random terms*. They accepted losses for 5 years in order to collect data from which they could learn what made a good customer that stayed and what made a bad customer who you shouldn't give a credit card to in the first place.

1.8.2 Target

This particular story received national attention. Basically, a father called Target to complain that his daughter was receiving targeted maternity coupons and it was distressing because his daughter wasn't pregnant. The manager apologized. A few days later, the father called back. His daughter was indeed pregnant. *He* apologized.

How did Target know that his daughter was pregnant? Well, they didn't *know* but they had a good inkling. Target collects data on all of its customers, assigning to each of them a "guest id". Looking at actual maternity purchases, could you back up and see what did this person purchased last month? Two months ago? Three?

> As [Target statistician Andrew] Pole's computers crawled through the data, he was able to identify about 25 products that, when analyzed together, allowed him to assign each shopper a "pregnancy prediction" score. More important, he could also estimate her due date to within a small window, so Target could send coupons timed to very specific stages of her pregnancy. One Target employee I spoke to provided a hypothetical example. Take a fictional Target shopper named Jenny Ward, who is 23, lives in Atlanta and in March bought cocoa-butter lotion, a purse large enough to double as a diaper bag, zinc and magnesium supplements and a bright blue rug. There's, say, an 87 percent chance that she's pregnant and that her delivery date is sometime in late August.

You can find out more by reading How Target Figured Out A Teen Girl Was Pregnant Before Her Father Did.

1.8.3 Walmart

Walmart also collects information on their customers. In 2004, as Hurricane Frances was about to make landfall, Walmart started analyzing the purchasing patterns of customers prior to a previous hurricane, Charley.

> The experts mined the data and found that the stores would indeed need certain products – and not just the usual flashlights. "We didn't know in the past that strawberry Pop-Tarts increase in sales, like seven times their normal sales rate, ahead of a hurricane," Ms. Dillman said in a recent interview. "And the pre-hurricane top-selling item was beer." Thanks to those insights, trucks filled with toaster pastries and six-packs were soon speeding down Interstate 95 toward Wal-Marts in the path of Frances. Most of the products that were stocked for the storm sold quickly, the company said.

Common sense says that things like toilet paper, milk, eggs and bread are always the first to go in an emergency but discovering an increased demand for Pop-Tarts and Beer is definitely interesting. We also see another example of how insights lead to decision making. It is neither knowledge for knowledge's sake nor an opportunity to apply the latest machine learning algorithm.

What kinds of things are we going to learn about purchasing patterns from the lockdowns associated with the COVID-19 pandemic?

You can find out more by reading What Wal-Mart Knows About Customers' Habits.

1.8.4 Obama Campaign

The first Obama campaign was already "data driven".

> In the 2008 presidential election, Obama's targeters had assigned every voter in the country a pair of scores based on the probability that the individual would perform two distinct actions that mattered to the campaign: casting a ballot and supporting Obama. These scores were derived from an unprecedented volume of ongoing survey work. For each battleground state every week, the campaign's call centers conducted 5,000 to 10,000 so-called short-form interviews that quickly gauged a voter's preferences, and 1,000 interviews in a long-form version that was more like a traditional poll. To derive individual-level predictions, algorithms trawled for patterns between these opinions and the data points the campaign had assembled for every voter—as many as one thousand variables each, drawn from voter registration records, consumer data warehouses, and past campaign contacts.

By the re-election campaign, however, they had grown more sophisticated and moved to models of individual behavior. They wanted to find *persuadable* voters. While the Romney campaign tried to match the Obama campaign's data science team, they never got beyond the question, "is all this advertising working?". The Romney team was one-tenth the size of Obama's.

As we all know, the application of data science to elections is a controversial topic following the 2016 election and the Brexit referendum. Both Cambridge Analytica and Facebook have come under increasing scrutiny for their different roles in the application of data to those elections and data "breeches". Events like these (as well as Walmart and Target) are bringing ethical concerns to the fore. The most visible result of this concern was the passage and implementation of the European Union's General Data Protection Regulation GDPR. Many believe something like GDPR will come to the United States and international firms have started general GDPR compliance as a matter of convenience.

You can find out more by reading How President Obama's campaign used big data to rally individual voters.

1.8.5 Identifying Place

Foursquare is a mobile app that allows you to share location information with friends by "checking in". Blake Shaw of Foursquare made a presentation at Data Gotham 2012 that showcased the interesting questions that could be answered with this kind of location data. He first started by animating the check-in data for New York City (Manhattan actually). This showed interesting but not unexpected patterns of people checking in at work and transportation hubs, eateries for breakfast, work and then eateries for lunch, work, then eateries and home for dinner and the rest of the night.

The kinds of questions that can be asked of this data include: 1. What is a place? He showed check-ins that identified “Central Park” and “JFK Airport”. Note that this is people checking-in and saying where they were. This process could be used to define a “place”. 2. How do check-ins vary over time? Next he shows charts that show check-ins over time for three places. These charts show how these places exist in time relative to their check-ins. The coffee shop has a different pattern than the casual eatery which is still different than the more formal restaurant. However, there are interesting patterns…the eatery is busy late night on Fridays and Saturdays and the restaurant is busy at midday on Sundays…they serve brunch. 3. What places should be recommended for a weekend? It turns out that stadiums, flea markets, dim sum, and pool halls are much more popular on the weekends. 4. What are the characteristics of a neighborhood? For this analysis he compares Soho to the East Village and shows how there are more check-ins at offices in Soho and more check-ins at bars in the East Village. 5. What neighborhoods are similar? He demonstrates using this same information to identify neighborhoods that are similar.

All of this is interesting information. It appears that Foursquare has finally started monetizing this data but that is the central challenge. You can have a lot of interesting data and frame questions about it but are the answers something you can act on? This is not to say that data can’t be used for something other than to make money but the goal is usually some sort of insight that influences decisions. These decisions might just make people happier.

You can find out more by watching Big Data and The Big Apple.

1.8.6 Pruning Trees in NYC

This next case study is interesting because it was done by someone at Media6Degrees, Brian Delessandro, during “donated” time. The idea of donated time has made some headway into some companies. Basically, it’s an opportunity to use company resources to give back to the community. It’s a bit like open source, except you are doing analysis. Although Delessandro is a work-a-day data scientist interested in his company’s bottom line, Brian’s decision to contribute was inspired by Jeff Hammerbacher’s observation that,

> The best minds of my generation are thinking about how to make people click ads…and that sucks.

and the desire to contribute to the social bottom line. As a side note, Jeff Hammerbacher left Cloudera to work in data science for medicine, specifically, depression.

The particular study that Delessandro was involved looked at the efficacy of “block pruning” in New York City. During a typical day, there are very few fallen branches and trees in New York City. This makes the typical workload easy to plan for in terms of staff and equipment. During storms, however, there is typically a significant increase in fallen branches and trees. This debris creates a public safety hazard (and inconvenience if they’re blocking your road).

Block pruning involves sending crews to specific locations throughout NYC to trim trees to make the aftermath of a storm look like a typical day. The question is, does this work?

The best way to answer this question would be to pick blocks at random separating them out into control (no block pruning) and treatment (block pruning) and monitor the blocks of the course of the next year or so. However, as in many medical trials, this approach creates an ethical problem because the City is purposely *not* trimming trees in certain areas and allowing hazardous conditions to exist. This rules out a traditional A/B test.

However, it turns out that NYC has a census of trees in the City ("The Urban Forest") that contains information about every tree in the city: species, age, location, etc. They also have historical data on every maintenance action on every tree. After aggregating this data to the block level, Brian was able use this *observational* data to build a model that related treatment (block pruning) to the outcome (work orders in the following year).

This particular study is interesting because of the presence of *confounding factors*. A confounding factor is one that affects both the treatment and the outcome. For example, blocks with more trees are more likely to have large tree and are thus more likely to get pruned. Additionally, the crews have latitude in doing "surgical" pruning as they travel the city. The original results actually show that pruning makes the city more hazardous! Using a method designed to control for confounding factors, the data did show that pruning reduced future hazards by 13%.

Observational studies like this are very common in situations where controlled testing would create ethical problems. For example, epidemiologists determined that cigarette smoking is hazardous not because they did an A/B test, randomly assigning people to control and treatment groups (and forcing the treatment group to smoke 1, 2, 3 packs a day). Instead, they were able to use observational data to determine that there was a strong causal link.

This research is also an excellent demonstration of why you must have domain knowledge when doing data science. You have to know where your data comes from and the actual, real world process that generates it.

My favorite quote of the presentation,

> Data scientist salaries are good but they're not country club good.

You can find out more by watching Change the World in a Day.

1.8.7 The Information Architecture of Medicine is Broken

This research by Ben Goldacre represents some interesting data science detective work in the field of medicine. The cruz of his research is this. He was interested in the efficacy of a certain anti-depressant ("the drug"). In the US, drugs are only approved by the FDA after they have undergone a certain number of clinical trials proving their efficacy. We will have a lot to say about "efficacy" later but in the case of medicine this goes by the standard of "clinically significant results". For depression there is a survey that sufferers take that determines a depression score. The clinically significant result here is a lowering of one or more points on the scale.

As you might imagine, the result of one study might show that the drug lowers depression by 1.2 points on average. Another study might show that that the drug lowers depression by 0.2 points on average. Still another study might show that the drug lowers depression by 1.3 points. With a sufficient number of studies, we can plot these points as a *histogram*. By the Central Limit Theorem, this histogram should be relatively bell-shaped or have a Normal distribution. What Goldacre found was that the actual plot looked like a bell curve cut in half, showing only the good results. How is this possible?

As previously mentioned, in order for a medication to be approved by the FDA it must undergo a series of clinical trials. The catch is that pharmaceutical companies can start trials and stop them when the results don't seem to be going in their favor. They do not need to report the existence of those failed trials. This explains why the results are not normally distributed. As a result, Goldacre and others have tried to start capturing information about all clinical trials.

You can find out more by watching The Information Architecture of Medicine is Broken.

1.8.8 Love in the Time of Data

Daniel Chapsky is a Data Scientist at Snap Interactive, which produces a dating app called AYI ("Are You Interested?"). AYI is a recommendation "engine" (model) built on top of Facebook's social graph and other information.

One of the interesting things about this presentation is the use of data-driven personas. Personas are very common in user experience/user interface (UX/UI) design. By adopting a particular persona ("manager", "tech lead", "data scientist") designers can go through an application and see how well the application serves the needs of that particular persona. AYI developed data driven personas of their customers using clustering. The presentation talks about three different personas, "Rupert", "Pam", and "Blane", who each want very different things from a dating site and interact with its features in different ways.

This case study also demonstrates some central problems in working with data. The most central here is feature engineering. How do we take the data available from Facebook and external sources and turn it into actual features in a recommendation engine? For example, Facebook interests can be noisy for some applications (for example, "Domino's Pizza", which is probably not a good dating interest) and too specific as well (for example, specific artists instead of "Jazz"). AYI tackled these problems by using external data to generate higher level *taxonomies* or labels.

Another aspect of feature engineering involves using the social network. Certain personas ("Pam") are more likely to respond to a message if they have a friend in common (a feature). They are also more likely to have a lot of friends (a feature). Using the social network and AYI membership, AYI is able to better match "Pam" with people she is more likely to respond to.

Finally, in a dating app, attractiveness is going to play some kind of role. Through exploring their data they noticed that women are picky regardless of age and men are pickier as they get older. They then layered these inferences (common interests, friends of friends, pickiness) onto the social network and created a recommendation engine.

Note that since this presentation, Snap Interactive rebranded to PeerStream. They are one of the largest social media companies with chat, video and dating apps that "piggyback" on the social network. It appears that AYI became "FirstMet". And for you GenXers out there, they have "0ver50" as well. One has to wonder how they're faring in a post-Cambridge Analytica world.

You can find out more by watching Love in the Time of Data.

1.8.9 Booz/Allen/Hamilton Data Science

This is a set of case study presentations by Booz/Allen/Hamilton (BAH) given at the Data Works Maryland Meetup. During the introduction, BAH discussed how they organize their data science teams. Since BAH is a consultancy, it is a little bit different than other companies and more project driven. For any given data science project, they assign a software engineer, data scientist and domain expert. Instead of trying to find a single unicorn with all the skills, they build unicorn teams. There is some interesting information in the presentation that we will return to but our interest here is in the lightning talks showcasing case studies in data science.

Malaria kills an estimated 600,000 people a year and nearly 2 million are infected. Poor countries are hit especially hard. Africa has an interesting mix of good and bad infrastructure. In terms of good infrastructure, Africa has a well developed cell tower and mobile phone system.

There was a malaria outbreak at a teak plantation. The typical response is to treat the local bodies of water with chemicals to kill the mosquitos but that didn't seem to be working this time. Someone made the observation that a large percentage of the workforce of this plantation was composed of migrant workers. Accessing anonymized cell phone data, they were able to create a map of where all of these people had been over the previous weeks. It turned out that many of them had traveled near Lake Victoria, hundreds of miles away. When Lake Victoria was treated, the malaria outbreak at the teak planation stopped.

Vehicle theft is a major problem in metropolitan areas both in terms of property loss and public safety. Using crime data and plotting it on a map of San Francisco, a data science team was able to identify several hotspots in the city for vehicle theft. Concentrating on a single hotspot they saw that the hotspot was surrounded by three parks that made foot access (to and from) very easy. Although one might think most vehicle thefts occur in the wee hours of the morning, looking at the data over time for this hotspot, they identified 9-10pm as the peak hours for crime.

Using this information as a model, the city deployed police to these specific locations at the specific times indicated by the model. In response, there was a shift in the hotspots.

Cancer is a major health problem throughout the world for both poor and wealthy countries. For those cancers with successful treatments, differential access to medicines can mean the difference between high and low mortality rates. In this particular case, the medicine used to treat the specific type of cancer (neither the medicine nor cancer were named) is the result of a biological process. Biological processes are difficult to control and have quite a bit of variation. The goal of the company was to see if data could be used to decrease the variability of the process and increase the yield, making more medicine available and lowering the cost.

The available data was mostly time series: records of sensor measurements (pressure, temperature, etc.) over time as the process ran. There was 10 years worth of the data, which measured in a several terabytes. The team's theory was that runs with similar profiles–the movements of measurements over time–would have similar yields. In order to measure similarity for time series, they used *dynamic time warping*. In the end they were able to identify the conditions that lead to less variable, larger yields.

Unfortunately, these videos were removed in late 2018.

Hopefully these case studies will have given you a flavor for what can be done with data science.

1.9 Data Science Use Cases

Kaggle used to publish and maintain an interesting list of Data Science Use Cases. The only way to access it now is via the Internet Archive Wayback Machine. I'm not entirely sure why they stopped it, it's not clear that anyone thought the list was exhaustive but maybe it just got to be too much work. I think the list is still suggestive and I encourage you to look through the list, to familiarize yourself with the types of problems that can be solved because this, too, is part of what a data scientist should know.

1.9.1 Data Science Use Cases by Function

Marketing

- Predicting Lifetime Value (LTV)

  - what for: if you can predict the characteristics of high LTV customers, this supports customer segmentation, identifies upsell opportunties and supports other marketing initiatives
  - usage: can be both an online algorithm and a static report showing the characteristics of high LTV customers

- Wallet share estimation
  - working out the proportion of a customer's spend in a category accrues to a company allows that company to identify upsell and cross-sell opportunities
  - usage: can be both an online algorithm and a static report showing the characteristics of low wallet share customers

- Churn
  - working out the characteristics of churners allows a company to product adjustments and an online algorithm allows them to reach out to churners
  - usage: can be both an online algorithm and a statistic report showing the characteristics of likely churners

- Customer segmentation
  - If you can understand qualitatively different customer groups, then we can give them different treatments (perhaps even by different groups in the company). Answers questions like: what makes people buy, stop buying etc
  - usage: static report

- Product mix
  - What mix of products offers the lowest churn? eg. Giving a combined policy discount for home + auto = low churn
  - usage: online algorithm and static report

- Cross selling/Recommendation algorithms/
  - Given a customer's past browsing history, purchase history and other characteristics, what are they likely to want to purchase in the future?
  - usage: online algorithm

- Up selling
  - Given a customer's characteristics, what is the likelihood that they'll upgrade in the future?
  - usage: online algorithm and static report

- Channel optimization
  - what is the optimal way to reach a customer with certain characteristics?

  - usage: online algorithm and static report

- Discount targeting
  - What is the probability of inducing the desired behavior with a discount
  - usage: online algorithm and static report
- Reactivation likelihood
  - What is the reactivation likelihood for a given customer
  - usage: online algorithm and static report
- Adwords optimization and ad buying
  - calculating the right price for different keywords/ad slots

Sales

- Lead prioritization
  - What is a given lead's likelihood of closing
  - revenue impact: supports growth
  - usage: online algorithm and static report
- Demand forecasting

Logistics

- Demand forecasting
  - How many of what thing do you need and where will we need them? (Enables lean inventory and prevents out of stock situations.)
  - revenue impact: supports growth and militates against revenue leakage
  - usage: online algorithm and static report

Risk - Credit risk

- Treasury or currency risk
  - How much capital do we need on hand to meet these requirements?
- Fraud detection
  - predicting whether or not a transaction should be blocked because it involves some kind of fraud (eg credit card fraud)
- Accounts Payable Recovery
  - Predicting the probably a liability can be recovered given the characteristics of the borrower and the loan

- Anti-money laundering
    - Using machine learning and fuzzy matching to detect transactions that contradict AML legislation (such as the OFAC list)

Customer support

- Call centers
    - Call routing (ie determining wait times) based on caller id history, time of day, call volumes, products owned, churn risk, LTV, etc.
- Call center message optimization
    - Putting the right data on the operator's screen
- Call center volume forecasting
    - predicting call volume for the purposes of staff rostering

Human Resources

- Resume screening
    - scores resumes based on the outcomes of past job interviews and hires
- Employee churn
    - predicts which employees are most likely to leave
- Training recommendation
    - recommends specific training based of performance review data
- Talent management
    - looking at objective measures of employee success

1.10 Data Science Use Cases By Vertical

Some of these are just the faintest wisp of an idea, still, they should be suggestive.

Healthcare

- Claims review prioritization
    - payers picking which claims should be reviewed by manual auditors
- Medicare/medicaid fraud
    - Tackled at the claims processors, EDS is the biggest & uses proprietary tech

- Medical resources allocation
    - Hospital operations management
    - Optimize/predict operating theatre & bed occupancy based on initial patient visits
- Alerting and diagnostics from real-time patient data
    - Embedded devices (productized algos)
    - Exogenous data from devices to create diagnostic reports for doctors
- Prescription compliance
    - Predicting who won't comply with their prescriptions
- Physician attrition
    - Hospitals want to retain Drs who have admitting privileges in multiple hospitals
- Survival analysis
    - Analyse survival statistics for different patient attributes (age, blood type, gender, etc) and treatments
- Medication (dosage) effectiveness
    - Analyse effects of admitting different types and dosage of medication for a disease
- Readmission risk
    - Predict risk of re-admittance based on patient attributes, medical history, diagnose & treatment

Consumer Financial - Credit card fraud - Banks need to prevent, and vendors need to prevent

Retail (FMCG - Fast-moving consumer goods) - Pricing - Optimize per time period, per item, per store - Was dominated by Retek, but got purchased by Oracle in 2005. Now Oracle Retail. - JDA is also a player (supply chain software)

- Location of new stores
    - Pioneerd by Tesco.
    - Dominated by Buxton.
    - Site Selection in the Restaurant Industry is Widely Performed via Pitney Bowes.
- Product layout in stores
    - This is called "plan-o-gramming"
- Merchandizing
    - when to start stocking & discontinuing product lines

- Inventory Management (how many units)
  - In particular, perishable goods
- Shrinkage analytics
  - Theft analytics/prevention
- Warranty Analytics
  - Rates of failure for different components
    - And what are the drivers or parts?
  - What types of customers buying what types of products are likely to actually redeem a warranty?
- Market Basket Analysis
- Cannibalization Analysis
- Next Best Offer Analysis
- In store traffic patterns (fairly virgin territory)

Insurance - Claims prediction - Might have telemetry data - Claims handling (accept/deny/audit), managing repairer network (auto body, doctors) - Price sensitivity - Investments - Agent & branch performance - DM, product mix

Construction - Contractor performance - Identifying contractors who are regularly involved in poor performing products - Design issue prediction - Predicting that a construction project is likely to have issues as early as possible

Life Sciences - Identifying biomarkers for boxed warnings on marketed products - Drug/chemical discovery & analysis - Crunching study results - Identifying negative responses (monitor social networks for early problems with drugs) - Diagnostic test development - Hardware devices - Software - Diagnostic targeting (CRM) - Predicting drug demand in different geographies for different products - Predicting prescription adherence with different approaches to reminding patients - Putative safety signals - Social media marketing on competitors, patient perceptions, KOL feedback - Image analysis or GCMS analysis in a high throughput manner - Analysis of clinical outcomes to adapt clinical trial design - COGS optimization - Leveraging molecule database with metabolic stability data to elucidate new stable structures

Hospitality/Service - Inventory management/dynamic pricing - Promos/upgrades/offers - Table management & reservations - Workforce management (also applies to lots of verticals)

Electrical grid distribution - Keep AC frequency as constant as possible

Manufacturing - Sensor data to look at failures - Quality management - Identifying out-of-bounds manufacturing - Visual inspection/computer vision - Optimal run speeds - Demand forecasting/inventory management - Warranty/pricing

Travel - Aircraft scheduling - Seat mgmt, gate mgmt - Air crew scheduling - Dynamic pricing - Customer complain resolution (give points in exchange) - Call center stuff - Maintenance optimization - Tourism forecasting

Agriculture - Yield management (taking sensor data on soil quality - common in newer John Deere et al truck models and determining what seed varieties, seed spacing to use etc

Mall Operators - Predicting tenants capacity to pay based on their sales figures, their industry - Predicting the best tenant for an open vacancy to maximise over all sales at a mall

Education - Automated essay scoring

Utilities - Optimise Distribution Network Cost Effectiveness (balance Capital 7 Operating Expenditure) - Predict Commodity Requirements

Other - Sentiment analysis - Loyalty programs - Sensor data - Alerting - What's going to fail? - De duplication - Procurement

1.11 Data Use Cases That Need Fleshing Out

Procurement - Negotiation & vendor selection - Are we buying from the best producer

Marketing - Direct Marketing - Response rates - Segmentations for mailings - Reactivation likelihood - RFM - Discount targeting - FinServ - Phone marketing - Generally as a follow-up to a DM or a churn predictor - Email Marketing - Offline - Call to action w/ unique promotion - Why are people responding- How do I adjust my buy (where, when, how)? - "I'm sure we are wasting half our money here, but the problem is we don't know which ad" - Media Mix Optimization - Kantar Group and Nielson are dominant - Hard part of this is getting to the data (good samples & response vars)

Healthcare

- CRM & utilization optimization
- Claims coding
- Forumlary determination and pricing
- How do I get you to use my card for auto-pay? Paypal? etc. Unsolved.
- Finance
  - Risk analysis
  - Automating Excel stuff/summary reports

1.12 Conclusion

At a high level, we can say that data science is concerned with the application of the scientific method to business problems, taking domain knowledge and ignorance and creating new domain knowledge. Although there isn't complete agreement after that, there are enough commonalities in the different definitions to define a broad set of skills that comprise *doing* data science. These skills include "hard" skills like programming, statistics, and machine learning as well as "soft" skills like communication and domain expertise. For the purposes of this text, we will define data science to be:

> the application of modeling to data to solve problems or answer questions in support of decision making

There are three main components here: problems/questions/decision making, data, and modeling.

We also presented a number of Data Science case studies. By the end of this text, you should have a greater understanding of what went into each of these case studies and be able to do similar analyses.

1.13 Review

1. What is the definition of statistics?
2. Why is it difficult to define data science?
3. What is the difference between scientists and computer scientists?
4. Is data science about big data only?
5. What are the steps of the scientific method?
6. What is the working definition of data science for this text?

1.14 Exercises

1. In Analyzing the Analyzers, the authors identify four different kinds of "data scientists". Skim through the report. If you plan on becoming a Data Scientist, what kind of Data Scientist will you be?
2. Because of the interdisciplinary nature of Data Science, (ideal) Data Scientists are described as having a T-Shaped skill set. Looking at the skill sets documented in *Analyzing the Analyzers* above, what breadth skills do you think are lacking and what skill would you like to develop as your "depth" skill?
3. Are there any interesting Data Science Use Cases in your function/industry? It's interesting to note that many of these use cases can be found in a typical business statistics book.

4. Find five data scientist job listings on an employment site. What's do they have in common? What's different? (You probably shouldn't do this at work…).
5. Find your own case study for data science (not just applied machine learning or "Artificial Intelligence"). Can you find one for one of the Data Science Use Cases?

2 Data Science Process

In this chapter we will discuss **The Data Science Process**. The Data Science Process is a framework that identifies the key steps to completing a data science project (or, at least, one iteration of a data science project). This Process includes identifying problems in your organization by talking to various stakeholders, obtaining and scrubbing data, data exploration, modeling and finally "reporting" your results in some form that affects the operation of your organization. We will use the **Green Cloud** model of Data Science in these notes. However, because different organizations have differing degrees of data science experience, we will also talk about **stages of data science** and the data science pyramid put forth by Russell Journey.

We will then look at the first step in the Green Cloud model of Data Science, **ASK**, and explore Max Shron's **CoNVO**. CoNVO stands for **Context**, **Need**, **Vision**, and **Outcome**. All good data science projects include these elements, at least implicitly. Finally, we will talk a bit about data science teams and data culture as the larger context in which the data science process unfolds.

2.1 The Data Science Process

Unfortunately, mere algorithmic cleverness is not sufficient to ensure that a data science project succeeds. If you can program a deep learning neural network using a redstone computer in Minecraft, but you cannot properly identify a problem with business value in your organization, your data science team will ultimately fail. If you can conduct statistical analyses but you cannot write clearly about your results, your results will go unused and your insights unheeded (here we interpret "reporting" quite broadly to include surfacing your results either as a notebook, a webapp, or moving the model into production).

In the Introduction, we defined Data Science as:

> the application of modeling to data to solve problems or answer questions in support of decision making

and we emphasized that data science was more related to applying the tools and mindset of science to everyday problems than with "working with data", whatever that might mean. We also emphasized that you will come to an organization with general skills but they will have their own, specific problems. Biologists spend years learning the ins and outs of biology;

you will have weeks to become an expert in your organization's processes, terms, and domain knowledge.

The Data Science *Process* is an idealized view of the steps needed to go from the start to the "end" of a Data Science project. These steps include things we generally think about (getting the data, building models) and those we do not (talking to people about the problem we're supposed to solve). The quotation marks around *end* signify that such projects rarely truly end.

I think the two most important guiding principles in Data Science are:

1. **Communication** - solve the right problem for the right people and talk about expectations and results constantly. Explain things clearly. You need to be good at both written and oral communication.
2. **Simplicity** - follow the agile development dictum and start with the simplest thing that could possibly work. Focus on solving problems that add business value and not algorithms.

A quick search for "Data Science Process" will bring up any number of diagrams that differ in both fundamental and subtle ways. If we search on Wikipedia (the final authority on everything, right?), we see a typical diagram of the "Data Science Process":

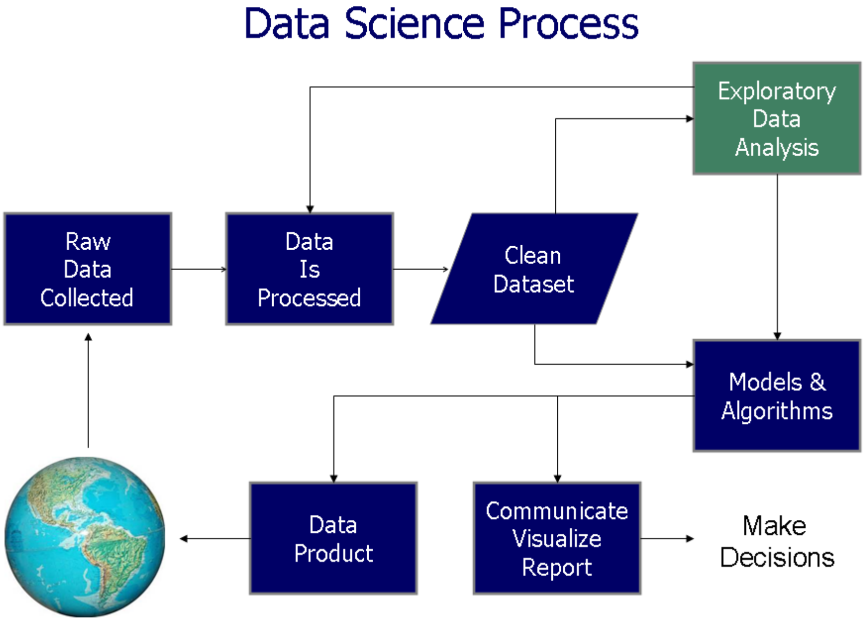


Figure 2.1: Basic Data Science Process

Source

This is a very typical rendering of the Data Science Process. It starts with the collection of raw data which is then processed. This processed data is cleaned into a form suitable for exploration. As you explore the data, errors and shortcoming may be detected in which case you go back to processing data. When the data is cleaned sufficiently, you progress to models and algorithms. Depending on the use case, the results of your modeling are either deployed as a data product (for example, a recommendation algorithm) or reported back to stakeholders who use the insights discovered to make decisions.

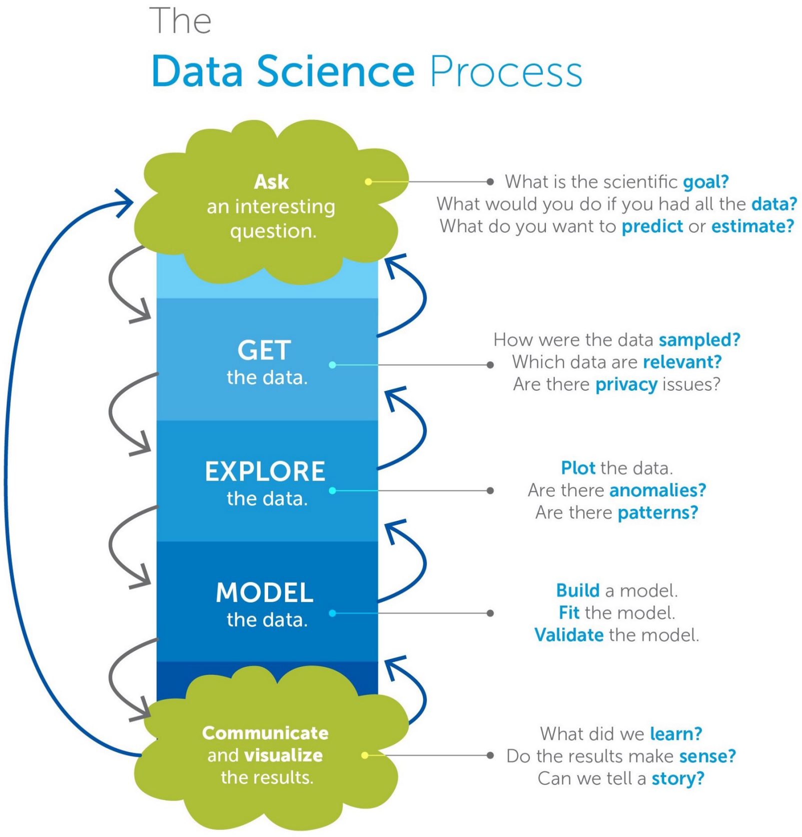
This characterization is fine as far as it goes but it falls a bit short by starting with the data. **This gives rise to unreasonable expectations on all sides.** First, stakeholders and decision makers think that just throwing data at "Data Scientists" will make money. Second, data scientists, perhaps with little training outside of academia, think that just sifting through data for six months will make a valuable contribution to the organization.

This is the path to "nothing appears to succeed and everyone is laid off, Data Science is hype." and harkens back to those nefarious and misguided Underwear Gnomes from *South Park*. As Forrester reports, 99% of companies think data science is an important discipline to develop yet only 22% of companies are seeing significant business value from data science. I have personally witnessed the dissolution of several data science teams and heard of several companies that either laid off entire teams of data scientists or were not hiring data scientists until they had "figured out" their data science story.

The Green Cloud model is more explicit about the starting point for the Data Science process: ASK - asking a question (or asking for problems to solve). Specifically, it *asks* you to identify a goal for the data science project. It also *asks* you to think about what you would do if you had all the data at hand. We also *ask* what kind of problem we're going to solve: prediction or estimation (which I would call "explanation"). More generally, we need to talk about our ideal solution to the problem and what kind of model it might entail. Although not shown here, we may need to address any ethical and or legal issues associated with solving this problem. We will talk about ways of framing the ASK step later in this chapter.

The GET step involves obtaining data, which may be inside or outside (or both) the organization. We also need to identify which data are relevant. This is also the place to address privacy issues related to the data (as opposed to the problem or question). "Extract, Transform, and Load" (ETL), "Data Munging", and "Data Wrangling" are often associated with the GET step. Data Scientists often spend 80% of their time in this step and the next.

The EXPLORE step involves looking at the data and trying to identify anomalies and problems. This step is often associated with Exploratory Data Analysis (EDA). The first pass at EDA might involve looking for anomalies and errors in the data. When such problems are discovered, we often have to go back to the GET step. The second pass at EDA will concentrate on visualization and the identification of patterns that will help us understand the data and make our modeling more effective.


The
Data Science Process
Ask an interesting question.
What is the scientific goal?
What would you do if you had all the data?
What do you want to predict or estimate?
GET the data.
How were the data sampled?
Which data are relevant?
Are there privacy issues?
EXPLORE the data.
Plot the data.
Are there anomalies?
Are there patterns?
MODEL the data.
Build a model.
Fit the model.
Validate the model.
Communicate and visualize the results.
What did we learn?
Do the results make sense?
Can we tell a story?

Derived from the work of Joe Blitzstein and Hanspeter Pfister, originally created for the Harvard data science course http://cs109.org/.

Figure 2.2: Green Cloud Model

The MODEL step is where we build representations of the data that help us solve the actual problem or answer the question. Models can be simple, if we want to know what our current churn rate is, or complex, if we need to identify entities in images. Simple models using averages, rates, and linear models are often all that is needed here, for most organizations, especially at the start. You can iterate later.

The REPORT step is where we communicate our results back to the organization. Here we try to connect the insights we have discovered back to the decision makers in the organization. That reporting can be through a report, an email, a notebook, a webapp, or a dashboard.

As indicated by the arrows, we will very often have to go back and revisit previous steps. There are some very "tight" loops in the GET, EXPLORE, and MODEL steps because, as we will see, your modeling may reveal deficiencies in the data you have at which point you need to go back and apply different transformation or get new data, all of which will need to be explored.

At the end, we may have more questions or problems that need solving and the process starts all over again. As we will see, it's best to keep focused on a single problem and "solve" it then come back to the questions and problems raised.

2.2 Side Note: Data Products

There are a ton of definitions for data products these days:

1. the data itself may be someone's "data product" (especially if it has a slick UI or API on top of it).
2. anything and everything (reports, visualizations, emails) that results from data analysis is a "data product".

I can see why #1 might be a "data product". I wish there were a better/different name. For #2, I think people just want to give a cool name to reports and visualizations. For me, a data product represents the integration of a model into the software platform/infrastructure of an organization. The canonical example is a recommendation "engine" that might be used to personalize a customer's email or the webpage they're visiting. It might also be the surfacing of a churn score in the UI of a customer service representative. Although the latter is starting to go by the ill-considered and horrible name, "reverse ETL". That is, if I read data from Salesforce into a data warehouse, that is "ETL". If I take that data, run a model on it to predict churn and write that score back to Salesforce, that last step is "reverse ETL".

One could easily include such "data products" in the REPORT step. However, I generally identify the creation of data products with *software engineering*. This doesn't mean, however, that some data scientists won't be doing some of this software engineering (I did). I don't think data scientists only do data science (otherwise, all machine learning is data science and, again, that's weird to me). It might be more likely, especially in larger organizations, that they'll have to closely coordinate with team members or co-workers who do those things.

My perspective on this topic is the result of several years of working on–and deploying–recommendation engines. I worked at a company that wanted to develop a recommendation "engine" to personalize customer emails. The Data Science team set to work on developing this algorithm, following the steps above. And, after a while, a candidate was found. The hard part was done, right?

No.

It's nice to have a model there in your Jupyter notebook but if that model needs to be integrated into the software platform, to actually affect real world emails and–and this is important–be fault tolerant like the rest of the production software, there are even more steps:

1. storing the recommendation model (if the system needs to train the model as well, this is even more complicated).
2. reading in the current inventory
3. reading in user preferences
4. applying the algorithm to the features for each user
5. storing the actual recommendations
6. surfacing the results to the emailer, which used the generic offerings in the event of a failure.
7. recording the recommendations and user interactions.
8. A/B analysis of the results.

People severely underestimate what's required to put something like this into production beyond trying to get a "good" model in a Jupyter notebook. As we can see, deploying a model as a "data product" requires coordination between teams and some sort of intrastructure. I'm not entirely willing to call this coordination and infrastructure, "reporting". But there *is* a reporting step in here; I just didn't realize it for years.

Returning to our story, after the first recommendation engine was deployed, every day for a month or more, the VP of Email (and others) would come to my desk and ask, "why is my email this way?" I would spend *hours* going over the inventory, preferences, and email for that specific person, trying to determine what the model had done and why and explain it to them. Still, what they really meant to say was, "I didn't like this item or that item that got recommended and I don't think your algorithm works."

Years later at a different company, I worked on a project to identify duplicate records in a database. Basically, a duplicate record is where we're fairly sure that two or more records are for the same person but they may contain slightly differing personal information (different email or phone number, name changes, etc.). But here, we were pre-emptive. When the algorithm was "done", we built a UI to show how it detected duplicate records and on what basis. The stakeholders could search all the duplicate records, see why the model thought they were duplicates, and simulate merging the records so they could see the result. Being able to "touch" the model that otherwise seemed entirely abstract and hidden away in the database gave them confidence in what we were doing.

In another project, we were merging job histories. We might have an old job history for someone and obtain an updated one. They may have added, deleted, condensed, or expanded jobs and positions. Again, when the algorithm was done, we built a UI so that people could see what a job history merge would look like and what the end result would be.

What I would like to have been able to tell my earlier self is that we should have had a UI that allowed stakeholders to see what their emails would have been using the existing formula and what they would be using the recommendation engine and why each item was there. And this surfacing of the model to stakeholders is definitely part of the REPORT step. But putting the model into production is an entirely different thing…

2.3 Side Note: Data

We're going to talk about data a lot in this text. Organizations have always had databases and we already mentioned that Gosset was trying to improve brewing using statistics well over a 100 years ago so so you might ask yourself, why are Data and Data Science such a big thing *now*?

The flippant answer is that no MBA has ever understood what their business statistics course was trying to tell them. And in a world of specialization, this is, perhaps, not unexpected and perhaps it was never realistic. After all, MBAs and Business Majors take accounting classes as well but they still hire CPAs. Maybe now they've just gotten around to hiring statisticians. A better answer, I think, hinges on four factors that converged over the last decade (although they started earlier than that):

1. Ubiquitous computing.
2. Cheap data storage.
3. Logging.
4. Success of Machine Learning.

Ubiquitous computing is huge for a number of reasons. Your iPhone has more computational power than the original Cray. Your laptop is beyond what anyone could even dream of a few decades ago. Remember Bill Gates' famous quote, "There's no reason anyone would ever want more than 540k of RAM". This means that organizations–and individuals–have more computing power at their disposal than ever before to run simulations, learn a programming language, do machine learning, etc. It used to be very expensive and time consuming to do any of these things.

Cheap data storage means we can just save everything and look at it later. My original Mac Plus had an external hard drive with 20Mb (that's right, *megabytes*) of storage. We routinely download applications and files larger than 20Mb. This has possible legal and ethical ramifications which we should not fail to recognize and discuss but in general, this is a great boon to organizations.

Logging is very important for the rise of data science, which is a bit weird because it's more or less a side effect of the rise of the internet. That is, nobody planned for logging to be used they way it's come to be used. We just wanted to make sure our webapps worked or be able to find the problem when they didn't. Instead, by logging every interaction on a webserver, we were inadvertently taking measurements of human behavior that we could never have done before. Let's compare two scenarios.

In a brick-and-mortar grocery store, you go up and down the aisles the way you want, you look at products, you pick some up, some are put in the cart and others are put back. In the days before UPCs, you just checked out. In an online store, all of that is controlled and audited. You are shown a particular order of products. Your search for specific products is logged. Every page you visit is logged. Anything you put into your cart is logged and everything you take out of your cart is logged. Even for non-ecommerce situations, there are important differences between old style computer systems and ones with logging. For example, in old style computer systems, when you update your address in a database, the old address is typically gone. When you update your marital status, the insurance company's database forgets that you were ever single.

The logs, on the other hand, remember everything. They are records of event streams that are much more interesting. Consider a log that records everything you look at on Amazon and the cost of doing the same thing at the local Safeway.

This is actually why the canonical bank account example in Object Oriented Programming is so misleading. Banks keep records of *transactions.* Your account balance is a derived value from those transactions (although there are cached starting points that they work from, normally coinciding with statement periods).

Finally, the broad success and availability of Machine Learning algorithms is finally making it possible to include them in everyday projects. This wasn't usually the case with Artificial Intelligence. Of course, data science isn't *just* machine learning and machine learning may not even be the most important part of data science.

2.4 The Data Mining Perspective

In the previous chapter we talked about the possibility that "data science" is just a fancy name for an already existing field. We gave the example of statistics and, more probably, business statistics. As it turns out, business statistics might not be the only claimant to the data science throne. Although data *dredging* or *fishing* started out as pejorative terms for "just trying to find things" in the data, **database mining** and eventually **data mining** came to be organized on more principled grounds especially in the KDD (knowledge discovery in databases) community.

The Wikipedia entry on Data Mining notes that KDD summarizes the steps of data mining as follows:

1. Selection
2. Preprocessing
3. Transformation
4. Data Mining
5. Interpretation and evaluation

The Cross-Industry Standard Process for Data Mining (CRISP-DM) defines six phases:

1. Business understanding
2. Data understanding
3. Data preparation
4. Modeling
5. Evaluation
6. Deployment

Data Mining for Dummies elaborates on these phases as follows:

1. Business understanding

    - identify business goals
    - assess the situation (context)
    - define the data mining goals
    - produce a project plan

2. Data understanding

    - gather data
    - describe data
    - explore the data
    - verify data quality

3. Data preparation

    - select data
    - clean data
    - construct data
    - integrate data
    - format data

4. Modeling

    - select techniques

- design tests
- build models
- assess models

5. Evaluation

    - evaluate results
    - review the process
    - determine the next steps

6. Deployment

    - plan deployment (integrate models/discoveries into use)
    - report final results
    - review final results

While these steps are on the same continuum as the "Green Cloud" model of Data Science, they do slice them a bit differently. We will revisit many of these over the course of this book.

2.5 Different Views on the Stages of Data Science

Data Science teams, projects, and even models can go through different stages. Do we have the data for this project? Do we have the expertise on this data science team? What kind of model do we need for this problem? There are lot of options and the *right* option is influenced by a host of factors. These factors may include:

1. *The appropriateness of a technique or algorithm to the problem.* Some approaches really are overkill. If a stakeholder wants to know about purchase rates, you don't need to train a deep neural network. On the other hand, if you have an application that is heavily dependent on "machine" vision and recognizing shapes in images, you may very well need to start with the state of the art.
2. *The availability of data for the problem.* We often talk about how much data is produced everyday but we don't often stop to consider if any of the data is actually useful or if the right data is being collected. You may have a problem for which your organization is not collecting data (and needs to start) or has been collecting the wrong data (and needs to fix or improve the data being collected).
3. *Historical approaches to the problem in your organization.* In the past, a financial analyst has always just calculated your performance metrics but now you want to get more sophisticated. You may want to change the metrics, start calculating confidence intervals for them and even projecting them. On the other hand, you may come to a problem that has already seen great improvements from linear regression and your task is to see if a new model or new features will improve performance. On *new* problems, you should

start simply but on *historical* problems, you have to take into account the work that has gone before.

4. *The maturity of the data science efforts in the organization.* The maturity of data science in the organization affects your choices in a variety of ways.

With fledgling data science teams, you often need to concentrate on breadth. With so many different problems, the return on investment (ROI) to data science is often highest at the start. That is moving from a non-data driven approach to a data driven approach is often has the biggest impact on the problem. Moving to more sophisticated models can bring incremental increases in returns but often not as big as the first models.

I saw this first hand at Company X. We struggled with building complicated models we "knew" should work but didn't. At one point we switched to a very simple model and got a huge lift over the "conventional" practice (in data science, "lift" is the differential increase in something like review; you will often see "lift" mean "effort" in product management circles. They don't mean the same thing at all!). And although we spent months looking for better models, most of the time they were only better than the first model by small amounts. Because of the effort involved in switching models, we required a new "winning" model to have a lift of at least 3%. We never replaced the simple model. So we concentrated on breadth, bringing data science to other parts of the company in order to get those large initial payoffs.

As the data science efforts in an organization matures, there can be more room for specialization. In "One Data Science Job Doesn't Fit All", Elena Grewal talks about this at AirBnB:

> We started off as the "A-team" — Analytics team — with the first hire an "Analytics Specialist." In 2012, I was hired as a "Data Scientist." Later, we hired a "Data Architect," to tackle data quality, then a "Data Analytics Specialists" to help solve gaps in data access and tools. Then we saw additional needs in machine learning so we hired "Machine Learning Data Scientists." These title evolutions were both reactions to team needs and also to the competitive landscape. We became the "Data Science" function in 2015, though we still use "A-team" because it's fun and has a history we value.

She goes on to explain how Data Science has evolved into three tracks: Analytics, Algorithms, and Inference. The *Analytics* track is focused on asking questions and surfacing answers in the form of dashboards. The *Algorithms* track is focused on applying machine learning to improve business products. The *Inference* track is focused on improving decision making and measuring the impact of initiatives.

It is interesting to observe that in her post, Grewal notes that people in the Analytics track often feel like they are undervalued relative to those in the Algorithms track even though they make substantial contributions. At Company Z, where I worked, we had a similar tension when the Business Intelligence (BI) team paired up with the Data Science team. I often think of the Analytics and Inference tracks as "BI++" (after C++). At Company Y, there was a similar division between those who worked on Inference and those who worked on Algorithms. I am aware that this division exists at many other organizations as well.

AirBnB is relatively mature, however. In an organization just getting started with data science, it may either focus just on Inference or have a small team that does Inference, Algorithms, and Inference. This is somewhat biased towards internet companies. Non-tech companies may not have an Algorithms focus at all.

2.5.1 Data Science Stages as Civilization

Ganes Kesari talks more about the evolution of Data Science teams in “What are the 3 Stages where Data Science Teams Fail?”. First, he echos a sentiment that will be a major theme throughout this text:

> Going by any analyst estimate, hundreds of billions are being thrown in by companies to solve problems with data. The key ask is to draw actionable insights that can drive business decisions. Building of predictive models is the top-of-mind recall with the very mention of the word, ‘analytics’.
>
> However, considerable business value in data science comes with the right application of exploratory analytics or statistical techniques. While AI or Deep learning have their rightful place, they are not silver bullets for Business ROI to every data problem.

But he continues, discussing the evolution of data science teams from “Makeshift Camp”:

> Similarly, too much of preparation for an ideal mix of skills could lead to analysis-paralysis. Onboard generalists, people who can cover many of the needed skills in analytics (say statistics, programming, and experimental design), even if only to a limited depth. The need is for survivors who flourish in scarcity, wear many hats and instill dynamism to solve any given challenge.

to “Thatched House”:

> Having won small victories with the initial team and established a purpose, the data science team can start fanning out into adjacent use cases. Slowly expand the scope of problems addressed and deepen partnership with users. Initial pilots can now mature into longer initiatives spanning a quarter or year.
>
> Showcase enhanced ROI to justify the next level of investment needed. While things may start to work in one’s favour, avoid over-committing in this interim stage. Start specialising by investing in few deeper areas (say Sales analytics, NLP), while continuing to be shallow and get job done in others (say design).

to “Palatial Home”:

> As an evolved entity, the data science team is essentially a mature business unit now. With specialised domain expertise and grasp over all key data science skill areas, the team is now ready to handle sufficiently complex problems, across a wide breadth of areas.
>
> No longer faced with existential challenges, the team's mandate can be deeply woven into long-term business objectives of the stakeholders. Teams could be structured with a vertical alignment, or as technical centres of excellence alias horizontals, or maybe along a hybrid, matrix structure which goes in between.

I have seen firsthand (and heard many times second-hand) stories about data science teams that did not survive their existential challenge. If your organization does not currently have a data science team or if you're part of a business unit spinning up their own data science capabilities, do not necessarily look to the Facebooks and AirBnBs of *now* for inspiration but instead find out how their Data Science teams started out. This is what makes advice from the current thought leaders of Facebook, Google, and others bad advice. They forget how they got to where they are now.

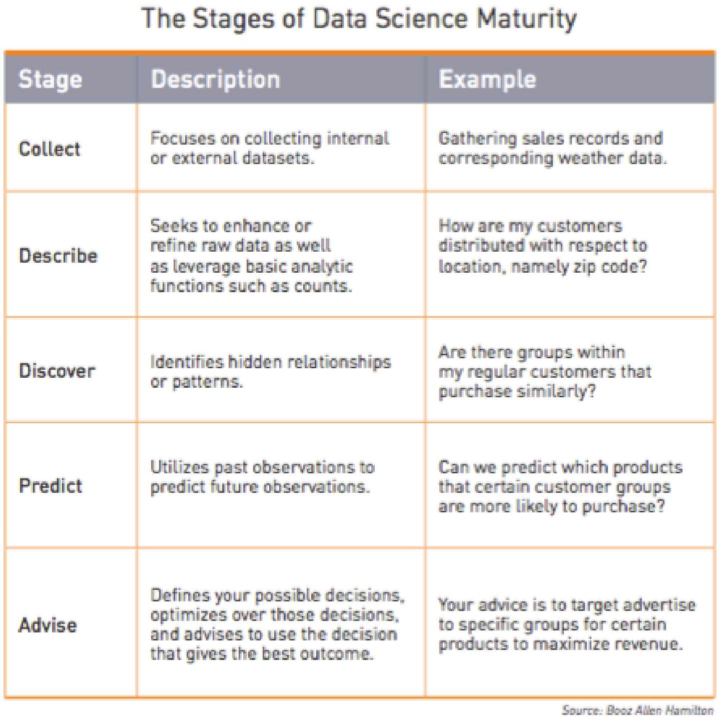
2.5.2 Booz Allen Hamilton: Levels of Data Science

Booz Allen Hamilton describes these Stages of Data Science maturity in a slightly different but useful way (Figure 2.3). They even go down one level...the first state of data science is collecting data and having it available. This is another landmine in the rush to having data science at an organization. If you do not have the data or it is not accessible to the data science team, your data scientists will be twiddling their thumbs. And while data scientists *should* have the basic skills needed to obtain data from internal databases and external APIs, a lack of data and or data infrastructure can severely limit what the team is capable of accomplishing. You cannot hope to long endure if you have a automatically generated weekly report that runs from Jane's laptop and requires constant supervision.

The first stage of data science is **Collect**. The organization needs to collect (internal) or obtain (external) the data. The provenance of the data needs to be maintained (where did this data come from?). Additionally, in this stage you are going to need the pipelines that move data from place to place. At first these might just be data scientists with the requisite skills. As the effort matures, however, this can evolve into a data infrastructure that moves data from production and external sources to data warehouses. The provided example is also illustrative of the idea of data *enrichment*. In this case, we want to see if there's a relationship between sales (internal data) and weather (external data). The weather data *enriches* the sales data.

The second level, **Describe**, was alluded to above. This is always where you start. It is foundational not only because of Exploratory Data Analysis but also because simple rate calculations–with or without data enrichment–can be very illuminating.

The third level, **Discover**, involves either visualization or clustering.



The Stages of Data Science Maturity

Stage	Description	Example
Collect	Focuses on collecting internal or external datasets.	Gathering sales records and corresponding weather data.
Describe	Seeks to enhance or refine raw data as well as leverage basic analytic functions such as counts.	How are my customers distributed with respect to location, namely zip code?
Discover	Identifies hidden relationships or patterns.	Are there groups within my regular customers that purchase similarly?
Predict	Utilizes past observations to predict future observations.	Can we predict which products that certain customer groups are more likely to purchase?
Advise	Defines your possible decisions, optimizes over those decisions, and advises to use the decision that gives the best outcome.	Your advice is to target advertise to specific groups for certain products to maximize revenue.

Source: Booz Allen Hamilton

Figure 2.3: Levels of Data Science

The fourth level, **Predict**, involves anything that's a simple as linear and logistic regression all the way to random forests and neural networks and, finally, deep learning. We will spend a lot of time on the simpler, *interpretable* approaches in this text.

The fifth level, **Advise**, bring the previous levels together along with A/B testing and experimental design.

Note that these stages can apply to specific *projects* as well as the entire data science effort of an organization.

How might the Stages of Data Science Maturity look in practice?

Imagine you're a data scientist for a specialty bakery with a retail outlet. A stakeholder comes to you and asks, "how busy are we on a typical day?". Your first thought would probably be, "why do you want to know this?". Not because you don't think the person shouldn't know it (or that the person should already know it) but because the use to which the information will be put plays a large role in determining how to answer it. It might also help you determine exactly what they mean by "typical" and "busy". "busy" might mean dollars per day or customers per day or donuts per day. Does "typical" mean Monday through Friday or September versus June?

Let's say that the someone from Finance just wants to know how many donuts we sell on average per day for a slide deck the CFO is putting together. Assuming that the data exists in electronic form somewhere *and you can access it* (not always the best assumption), this is a the **describe** stage of data science maturity. Much of the time, these simple data science problems involve a short conversation, a quick SQL query, and an email reply. Still, they can make up a large proportion of the work that data scientists do. They can also be the kinds of things that evolve into dashboards. Someone somewhere decides that they need to be able to see the number of donuts sold everyday, at a moment's notice.

When the CEO makes their presentation, someone from a different department hears the CFO's presentation and is interested in how this relates to ordering. We don't really want to have more than a month's worth of ingredients on hand at any given time, given the carrying cost and risk of spoilage. They ask you to work with them to figure out how average sales of donuts might work into ordering ingredients. Something subtle has happened here. We have moved from describe to **predict**, because we're not just describing what happened in the past but estimating what might happen in the future.

You work together and decide that for now the average donuts sold per day over the last month is a good enough indicator of how much flour, eggs, milk, yeast, sugar, etc., to buy for the coming weeks and months. This is a simple constant model using the average (mean) number of donuts sold.

After a while (or a week), they come back and ask if you can help them determine how many donuts they should make per day. If we make too few, we lose customers. If we make too many, we have leftovers that are sold as "day old" donuts, which cannabilizes sales of fresh donuts. Anything leftover is thrown away.

First, you suggest donating the excess to a homeless shelter. Next, you ask whether making too few donuts has the same cost as making too many donuts. After some discussion, everyone agrees that making too few donuts costs more than making too many donuts. Why? On days we have too few donuts, we lose customers and can't attract additional customers. On days we have too many donuts, we still recoup some costs. This implies we shouldn't use the mean to estimate how many donuts we should make on any given day because we'll have too few donuts 50 percent of the time and too many donuts the other 50 percent of time. After additional discussion, everyone agrees that we should only run out of donuts at most once per week or 1/5, 20 percent of the time.

What do we use instead of the mean? We need to build a distributional model of how many donuts we sell per day and find the 80th percentile of donuts sold. This is a slightly more complex predictive model.

After a few weeks, both models (the ingredient model and the donuts-to-make model) might need some refinement. The baker notices that during some parts of the year, we start accumulating ingredients and in other parts of the year we run short, requiring us to buy additional ingredients at higher prices. You look into the data and see that donut sales are noticably cyclical: people buy more donuts in the winter and fewer donuts in the summer. Instead of using the last month's donut average donut sales, you build a **regression model** that uses additional information about the upcoming month, season, and current sales levels to estimate how much of each ingredient to buy.

The donuts-to-make model needs refinement because the cashiers have noticed we run out of donuts more often on Monday and have more unsold donuts on Fridays. This might indicate that we need a different distributional model for each day of the week.

You can see where this is going. At some point, someone wants to start a **loyalty program** to improve **customer lifetime value** (CLV). Everyone who belongs gets a "baker's dozen" of donuts. Once we have a loyalty program, we want to start doing **recommendations** for donuts and other pastries. We have been so successful with our modeling that HR wants a model to help them fill out the schedule…

2.5.3 Agile Data Science Pyramid

Agile Data Science is a wonderful book by Russell Journey that is hard to classify. It takes a practical and hands-on approach to building an entire data science project iteratively–as you expect from something with the term "Agile" in the title. It applies the "Agile" philosophy to doing data science. The Manifesto for Agile Software Development states:

> We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

> Individuals and interactions over processes and tools Working software over comprehensive documentation Customer collaboration over contract negotiation Responding to change over following a plan
>
> That is, while there is value in the items on the right, we value the items on the left more.

Taking these four times together, the general idea was that software engineers would consult directly with end users to build software with the features they actually wanted to do the work they needed to do. The software would be built iteratively starting with the basics and enhanced as additional "user stories" were uncovered and implemented.

Journey wants to apply this method to data science. As a result, some of the focus shifts. For example, data science as science does not necessarily have predetermined results that can be managed in a predictable way. Although it is widely recognized that building software is not like building cars, you can still get an approximate estimate from a programmer as to when a new feature will be added. I have been asked by product managers when I thought I would find a more successful model! This isn't much different than asking when I'll find that cure for cancer.

As a result, data scientists need to be very communicative about their process and progress. If you shoe horn data science into a software engineering model, the stakeholders will often be frustrated. I remember being the only data scientist at a smaller company and having to report each morning at the "stand up" (meeting), what I had done and what I was doing. While the software engineers had different bugs and features every day, I was almost always working on the same thing. The method of reporting didn't match the work. However, there were also some lessons I hadn't yet learned about communicating progress over results. Journey formalizes the iterative application of data science to a problem in the Agile Data Science Pyramid:

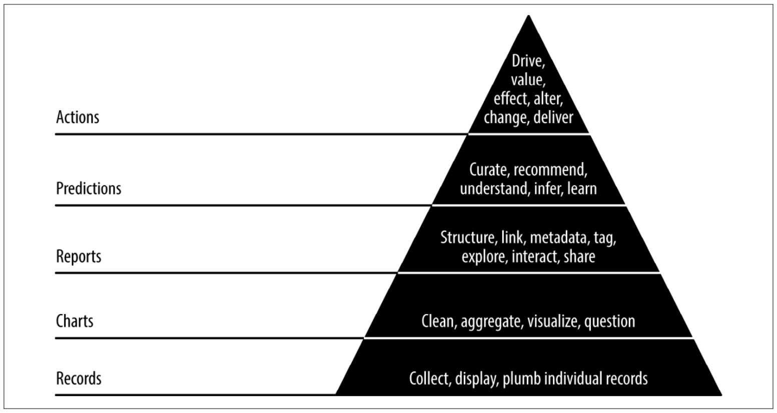

Drive, value, effect, alter, change, deliver
Actions
Curate, recommend, understand, infer, learn
Predictions
Structure, link, metadata, tag, explore, interact, share
Reports
Charts
Clean, aggregate, visualize, question
Records
Collect, display, plumb individual records

Figure 2.4: Agile Data Science Pyramid

With start with the base, **records**. In this case, you are building the infrastructure for data scientists and stakeholders to get access to the raw data themselves. This is related to the *GET* step above.

The next layer involves using **charts** to display aggregate information. This layer is followed by **reports**. These layers are related to the steps in the Data Science Process, both ASK and EXPLORE. In *charts*, we produce visualizations to generate questions and problems. In *reports*, we are moving away from unstructured exploration to structured linking and reporting.

The next layer is to add **predictions**. Here we are taking the previous layers and moving towards insights and recommendations for action. This fits well with the MODEL step of the Green Cloud process. The final layer is *actions* in which the previous layer leads to actual changes in organization strategy and tactics or is integrated into company products.

There are a lot of different ways of characterizing the Data Science process as well as the evolution of Data Science teams and projects. For most of the text we will stick with the Green Cloud model except where the other models have an interesting perspective as we delve more deeply into the details. We'll start here with the ASK step.

2.6 Step: ASK

So we start at the start. I assume that our organization has been collecting data as part of its regular business process and perhaps online presence. It might have done some marketing surveys. If not, there will be scant little for data *scientists* to do and they need to be hiring some data *engineers*. Let's assume that "having data" is a solved problem for now…although our organization might not have all the *right* data.

In the introduction to their Data Science uses cases, Kaggle provides a good set of starting questions for any data science project:

1. What problem does it solve and for whom?
2. How is it being solved today?
3. How could a different solution (data driven) beneficially affect business?
4. What are the data inputs and where do they come from?
5. What are the outputs and how are they consumed (predictive model and a report? recommender and an email? statistical analysis and a dashboard?)
6. Is this a revenue leakage ("save us money") or a revenue growth ("makes us money") problem?

We can rephrase and revise No. 6 to be a bit more explicit: does this make us money, save us money, or *create an asset*. These are great prompts for the ASK step and are likely to lead to better conversations. Additionally, I would that the question or problem posed should be meet the general characteristics of a scientific question: is it falsifiable? For example, "Increase customer satisfaction" is not a falsifiable problem or question. Not only must have you have data but you also need to arrive at a *metric* with which to gauge success.

There are aspects of these questions embedded in CRISP-MD Phase 1, "Business Understanding". This Phase includes steps such as identifying the business goals, assessing the situation, defining the data mining goals, and producing a project plan.

In addition to the CRISP-DM phases, there is often a discussion of the "9 Laws of Data Mining". Only the first two are relevant here (although we'll have cause to look at the others in later chapters).

- 1st Law of Data Mining - "Business Goals Law"

Business objectives are the origin of every data mining solution. Specifically, data mining is *not* about technology but is a process. The heart of that process is business objectives.

- 2nd Law of Data Mining - "Business Knowledge Law"

Business knowledge is central to every phase of the data mining process. We saw the phases earlier. In the current context,

> - Business understanding [Phase 1] must be based on business knowledge, and so must the mapping of business objectives to data mining goals. (This mapping is also based on data knowledge data mining knowledge).

2.7 Max Shron's CoNVO

In what follows, we will elaborate on the ASK step a bit more by describing a framework proposed by Max Shron Thinking with Data which also has a handy mnemonic, "CoNVO". The goal of this framework is to make sure we pick problems and set expectations about the solution appropriately. I like the CoNVO approach because it includes mock ups for the solution and thus plans for the final step in the Green Cloud, REPORT.

It can also be used to track progress. Anytime you start to diverge from what was agreed upon during the "CoNVO", it's time for another "CoNVO". That being said, I have only ever used this framework *implicitly*. I once brought up the idea of using it explicitly and got a bunch of blank stares. Apparently not everyone reads the same books I do! Don't be pedantic about it. You should think about how to get the same information, even if you can't be explicit about it.

As we stated previously, an approrpriate *data science* question is one that solves a business problem when answered. One way to make sure you have an appropriate question is to follow the CoNVO framework. **CoNVO** stands for **Co**ntext, **N**eeds, **V**ision and **O**utcome.

- **Context** - What is the context of the need? Who are the stakeholders and other interested parties?
- **Need** - What is the organizational need which requires fixing with data?
- **Vision** - What is going to be required and what does success look like?
- **Outcome** - How will the result work itself back into the organization?

For any Data Science project, you should know the CoNVO even if you don't explicitly use those terms. Keep refining the CoNVO as you delve into the data.

The key idea to keep in mind is that we're not saying "we need to do a logistic regression to predict customers who might turnover". This is too much detail and assumes too much about the actual problem ("is turnover really a or *the* problem?"), where the data might come from ("do we even have the data we need?"?, where the results of the model go ("where does this turnover score go? Who reads it? What do they do if it's 'high' "?). Additionally, it doesn't do any good to predict turnover if your customer service is crappy, or your product sucks, or your website is unusable. We're start out by asking the right questions.

In the book, Shron describes a situation faced by a university:

> Suppose that a university is interested in starting a pilot program to offer special assistance to incoming college students who are at risk of failing out of school (**Context**). What it needs is a way to identify who is at risk and find effective interventions (**Need**). We propose to create a predictive model of failure rates and to assist in the design of an experiment to test several interventions (**Vision**). If we can make a case for our ideas, the administration will pay for the experiments to test it; if they are successful, the program will be scaled up soon after (**Outcome**).

So we have a Context that includes the identification of a problem: some incoming college students are at risk of failing out of school and the university is interested in starting a pilot program. You can think of the Context as the background to the problem or question.

The Need is to identify who is at risk and what interventions are effective. The Need is the actual problem or question. If someone asks you, "what's the actual problem?", for any given data science project, you should be able to give a short explanation that is basically, the Need. Notice the problem doesn't say anything about the solution. The need isn't "we need a dashboard". The need *might* be "we need to identify the operational status of the network at any given moment."

The Vision is a proposal to create a predictive model of failure rates and to design experiments to test different interventions. We haven't specified the kind of model yet (regression, random forest, etc.) We have just mapped out the general goals. This step also includes mockups of what the inputs and the outputs might be. In this case, it might be a score from 1 to 10. We also have the experimental design and the identification of interventions and how those results will be communicated. The Vision here might include example tables.

The Outcome is to communicate the findings and if they are robust, scale up the program. This short paragraph does skip over more details. When you describe your Outcome, you should also describe how *exactly* people will receive the risk score, how they will interpret it, and what they will do, what decisions they will make as a result of the score. Let's dig into this all a bit more deeply.

We can actually map the Kaggle questions above onto the CoNVO:

1. What problem does it solve (**Need**) and for whom (**Context**)?
2. How is it being solved today (**Need**)?
3. How could a different solution (data driven) beneficially affect business (**Outcome**)?
4. What are the data inputs and where do they come from (**Vision**)?
5. What are the outputs (**Vision**) and how are they consumed (**Outcome**)?
6. Is this a revenue leakage ("save us money") or a revenue growth ("makes us money") problem (**Context**)?

so we can see that the CoNVO is just a different way of getting at these key parts of the ASK step.

With regard to the specific elements, most importantly, the Need is *not* a dashboard or Hadoop or deep learning. A data product or tool is a potential solution; it is not a Need. Shron notes, "A data science need is a problem that can be solved with knowledge, not a lack of a particular tool." The solution is never "the CEO needs Tableau". At Company Z, I built a dashboard from scratch that included logging so that I could see the most used features. That's right, I went meta with my data science. When people said "we need this", I could verify which features were and weren't being used. If a feature wasn't being used, I could ask, "why?".

It's worth noting, however, that if your project has already been in progress for sometime, many of these questions will already be answered for you. While this might constrain your available options, you should feel free to investigate other possibilities.

Shron's approach of starting with a Need instead of the *data* is in stark opposition those who emphasize "playing with data". As more and more data science projects fail, I feel like the "give data scientists data to explore and they'll find you money" is becoming a caricature...or maybe not because I continue to hear of projects failing. In any case, this probably seemed more radical at the time. Because we start with a Need (and not the solution) as we identify a Vision (a potential solution) we are not constraining ourselves by the data. Of course, our Vision may be too grandiose at the start, especially once we find out what data we actually have or what data we can actually afford. And this matches nicely with the Green Cloud process where we ASK the question or pose the problem and we imagine we have all the data we need. Then reality sets in and we have to GET the data we can.

There's a bit more to say about Vision. In Shron's framework, the Vision is conveyed through **Mockups** and **Argument Sketches**, either singly or together. These ideas borrow heavily from software engineering and design, especially website design.

Let's take Argument Sketches first because they're a bit easier to explain. Has someone ever tried to persuade you using evidence? For example, you might read something that says:

> We should build our new factory in Georgia because the workforce is growing, state and local governments are giving tax breaks to companies chosing Georgia, and it will reduce labor and transportation costs to our markets in the southern United States.

An Argument Sketch is persuasive. It has no actual numbers in it (although see below). The aim here is to identify the argument we want to make by writing it out and thus making it concrete. We're getting everyone on the same page about what the goal of the analysis is.

One might think, aren't we putting the cart before the horse? How do we reach conclusions *before* we've done any research? However, we aren't saying that the argument sketch will ultimately be true. We're laying out our argument to *guide our analysis*. In the context of the Green Cloud Model, this is really just a more business-y form of "make a hypothesis" and "what would you do if you had all the data?".

Still, it might be safer to stick with **Mockups**. There are two kinds (at least) of Mockups: textual and visual.

A textual mockup is simply a template into which we will insert adjectives and numbers. Consider the same basic Argument Sketch but as a Mockup (or Template):

> The workforce in Georgia is (growing|shrinking) (% in 2021). State and local governments have given $M dollars in tax breaks to companies like ours. Labor costs would be (reduced|increased) by %. Transportation costs to markets in the Southern US would be (reduced|increased) by %.

Here we have indicated the kind of facts we interested in, that will sway our decision to build our next factory in Georgia or not. The prose indicates the facts that we will find persuasive (workforce growth, tax breaks, labor costs, and transportation costs) and leave blanks for the actual values. This is a **Textual Mockup**. You could do a PowerPoint-ish version as well:

We are (not) building our new factory in Georgia…

- (Growing|shrinking) workforce (% in 2021)
- (No) Tax breaks (\$M last year)
- (Lower|higher) labor costs (%) and
- (Lower|higher) transportation costs to the South (%)

This is just a template. But we are indicating the key points that will go into our decision and highlighting the information (evidence) we need to get. (I find the difference between Argument Sketches and Textual Mockups to be so slight that I would include Textual Mockups in with Argument Sketches).

The **Visual Mockup** is a picture instead of prose. It is a low level sketch of a possible result of all our work usually something like a chart or charts, dashboard, spreadsheet, email or report. Mockups are common in the design of User Experience or User Interface in software and they can have the same function here. In fact, you may have seen the infamous *Lorem ipsum* text by accident (or on purpose). Here's a wireframe for a YouTube-like website:

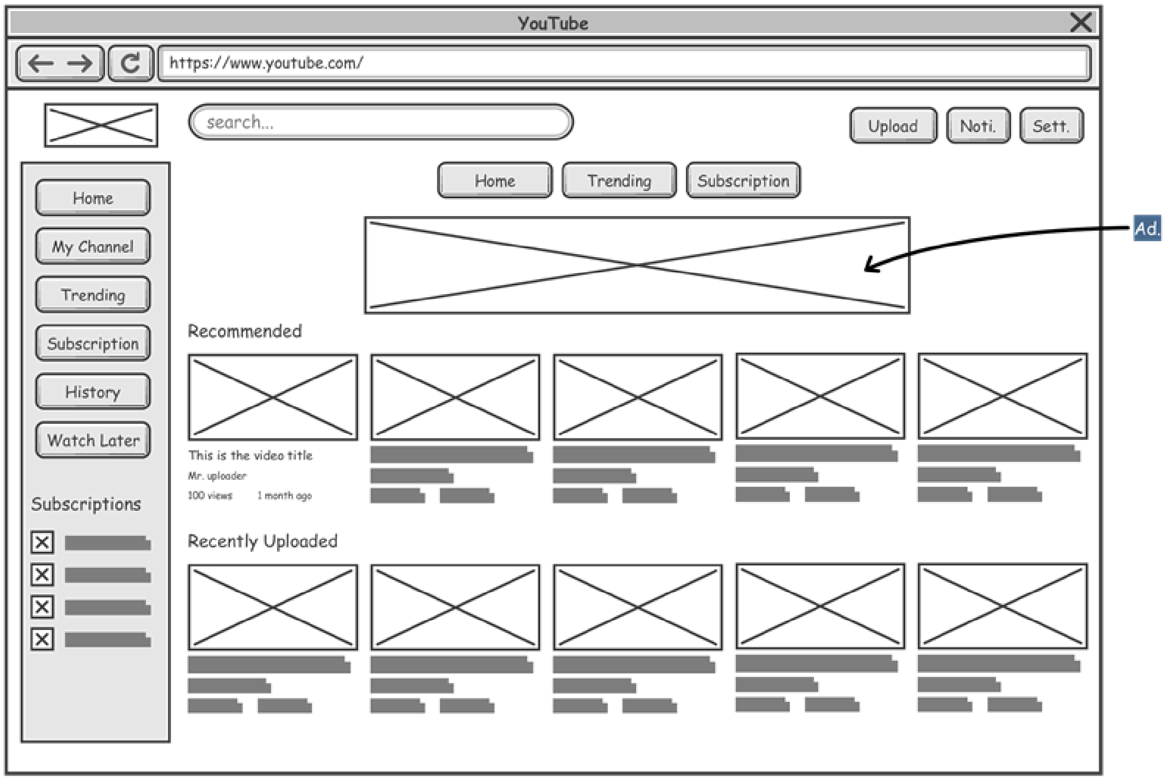
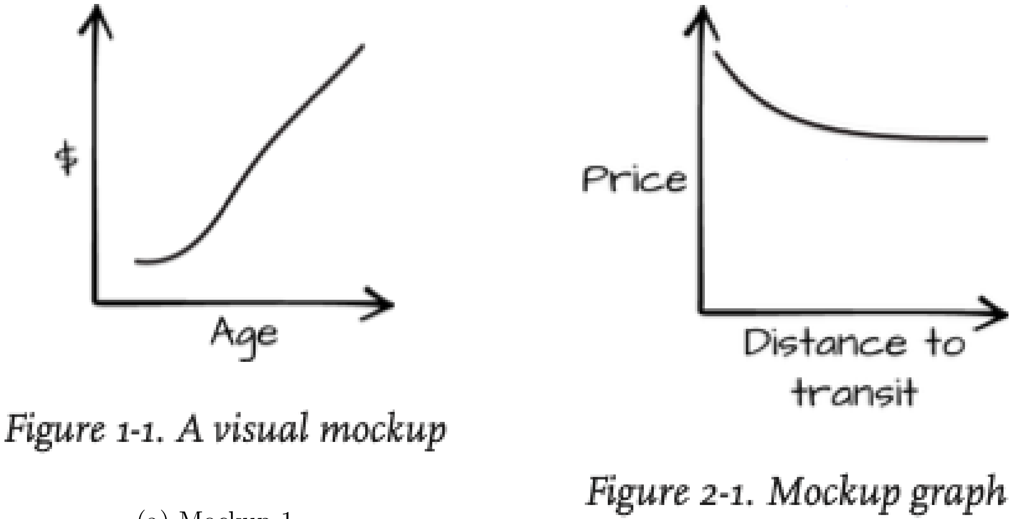


Figure 2.5: Website Wireframe

For our purposes, a Mockup is a wireframe of tables, charts, and statistics that the report will include. If a stakeholder says, "I need a chart of sales by minute". You can draw a chart or charts of sales by the minute. You can draw the axes, the tick marks with labels, and an example line. The goal is to get the stakeholder to think about what the request really means. After you literally drawing them a quick sketch of what their request actually is. As a result, they may realize that, no, they actually only need sales per week. Shron doesn't have many examples of actual Visual Mockups but here are a few of them:

They aren't complicated. It could be something you sketch on a whiteboard.

But, by providing these examples ahead of time…before you even do any analysis, before you get any data, you can get the stakeholder's feedback…and for exactly the same reasons that we show wireframes of websites before we build the whole thing: mistakes and misunderstandings



(a) Mockup 1

(b) Mockup 2

Figure 2.6: Mockups

are costly. Or, you can show someone a chart and ask them how it solves their need. You don't have to be confrontational. You're basically just saying, "like this?" Sometimes there will be a realization along the lines of "Oh, no, I just need this number", when you make nebulous ideas *concrete*. Or you and the stakeholder may realize that two charts are required.

Having a good mental library of examples (of Mockups and Argument Sketches) is critical to coming up with a Vision. This library of examples can be acquired by reading widely and experimenting, making note of particularly effective arguments and visualizations.

The importance of the Mockup cannot be overemphasized. People grossly mis-estimate what they think they need to solve their problem and how much time it'll take to solve it. A dashboard does sometimes fulfill a need but an email is easier, faster, and cheaper. You can actually have too much information. If someone says, "I need a dashboard"...draw it on a whiteboard. Draw the charts. Ask them, what do these charts tell you and what decisions will they influence. You don't have to be confrontational but *now* is the time to get clarity. Requests for dashboard are often preceded by the adjective "real time". If it takes weeks to get your suppliers lined up, a real time dashboard is not needed...and can, in fact, be harmful. (Of course, these days you may very well have a team of "BI" experts whose sole job is building dashboards).

2.8 CoNVO Examples

In order to familiarize yourself with the framework, I have included a few examples here of CoNVOs from Shron's book.

2.8.1 Refugee Non-Profit

Context - A nonprofit reunites families that have been separated by conflict. It collects information from refugees in host countries. It visits refugee camps and works with informal networks in host countries. It has built a tool for helping refugees find each other. The decision makers are the CEO and CTO.

So this is the context of the problem, where we find our interesting question.

Need. The non-profit does not have a good way to measure success. It is prohibitively expensive to follow up with every individual to see if they have contacted their families. By knowing when individuals are doing well or poorly, the non-profit will be able to judge the effectiveness of changes to its strategy.

This is the need, the actual problem, and what forms the basis for the interesting question.

Vision. The non-profit trying to measure its successes will get an email of key performance metrics on a regular basis. The email will consist of graphs and automatically generated text.

- **Mockup**. The Mockup would consist of wireframe drawings of the proposed kinds of graphs, showing labels, highlights, and frequency. The metric would be unspecified and very general for now.
- **Argument Sketch**. This may consist of an example of generated text (or the template) that says, "The nonprofit is doing well (poorly) because it has high (low) values for key performance indicators." The actual values and names of the key indicators will be inserted into the generated text. After seeing the key performance indicators, the reader will have a good sense of the state of the non-profit's activites and will be able to make appropriate adjustments.

The Vision is about what a solution looks like without delving into the details. What are we going to do? We're going to identify key performance metrics, generate graphs and descriptive text about them and send them in an email every week to the CEO and CTO. There are two suggestions for this Vision. With the Mockup we spec out an email that shows what it would be like for the system to generate an email that shows something the NGO has never seen before with its current record keeping, hitting an enrollment goal. With the Argument Sketch, we create a template for text that the system might generate.

This might be a little vague at this stage because we haven't identified what the performance indicator is. Suppose we have a performance metric, "percent of families reunited". For the

Mockup, the graph might be the trend of that percent over the last several weeks. If the value crosses a threshold, it might be highlighted in green (or in red if it goes below a certain threshold). The Mockup is an actual example of what this graph would look like…even if you just draw it out by hand on a piece of paper or the whiteboard.

Similarly, the generated text might say,

> We did well this last week because our Reunited Index reached 23%.

or it might say,

> We did poorly this last week because our Reunited Index reached 9%.

The basic principle here is that we want to identify the specific argument or description we plan to make so that we understand what the supporting numbers (and thus calculations and data) need to look like. In either case, the stakeholders do not need to *imagine* what this information might look like. They can see actual examples of this information and acting on it.

It could be as simple as drawing a rough chart of anything on a piece of paper, labeling the axes, giving it to the stakeholder and saying, what if you got this chart in an email, how would that help you make a decision? And they may very well find that it doesn't, at all. Maybe they need a table of the actual numbers. Maybe they need the information only once a week. Maybe they need entirely different information. By providing a concrete example of the end result, stakeholders can better determine if the proposed solution (Vision) actually meets their Need or not.

Have you seen the design shows where people see a house, hate it, and then some designer comes and shows them what can happen with a little remodling? It's like that. People are notoriously bad at the abstract. Give them something concrete.

Outcome. The metrics email for the nonprofit needs to be setup, verified, and tweaked. The sysadmin at the nonprofit needs to be briefed on how to keep the email system running. The CTO and CEO need to be trained on how to read the metrics emails, which will consist of a document writtent to explain it.

It's not enough to generate an email. The result needs to be a data *product* with technical support, education, and training on how to interpret the charts and text. There is no such thing as "intuitive" use.

Using this discussion as an example, go through the following three examples and see how the CoNVO is documented.

2.8.2 Marketing Department

Context. A department in a large company handles marketing for a large shoe manufacturer with an online presence. The department's goal is to convince new customers to try its shoes and to convince existing customers to return again. The final decision maker is the VP of marketing.

Need. The marketing department does not have a smart way to select cities to advertise in. Right now it selects targets based on intuition but it thinks there is a better way. With a better way of selecting cities, the department expects sales to go up.

Vision. The marketing department will get a spreadsheet that can be dropped into the existing workflow. It will fill in some of the characteristics of a city and the spreadsheet will indicate what the estimated value would be.

- **Mockup** - By inputting gender, age skew and performance results for 20 cities, an estimated return on investment is placed next to each potential new market. Austin, Texas is a good place to target based on gender, age skew, performance in similar cities and its total market size.
- **Argument Sketch** - "The department should focus on city X because it is most likely to bring in high value." The definition of high value that we use is substantiated for the following reasons.

Outcome. The marketing team needs to be trained in using the model (or software) in order to have it guide their decisions, and the success of the model needs to be gauged in its effects on sales. If the result ends up being a report instead, it will be delivered to the VP of Marketing, who will decide based on the recommendations of the report which cities will be targeted and relay the instructions to the staff. To make sure everything is clear, there will be a follow-up meeting two weeks and then two months after the delivery.

2.8.3 Media Organization

Context. This news organization produces stories and editorials for a wide audience. It makes money through advertising and through premium subscriptions to its content. The main decision maker for this project is the head of online business.

Need. The media organization does not have the right way to define an engaged reader. The standard web metric of unique daily users doesn't really capture what it means to be a reader of an online newspaper. When it comes to optimizing revenue, growth and promoting subscriptions, 30 different people visiting on 30 different days means something different than 1 person visiting for 30 days in a row. What is the right way to measure engagement that respects these goals?

Vision. The media organization trying to define user engagement will get a report outlining why a particular user engagement metric is the best one, with supporting examples, models that connect that metric to revenue, growth and subscriptions; and a comparison against other metrics.

- **Mockup**. Users who score highly on engagement metric A are more likely to be readers at one, three and six months than users who score highly on engagement metrics B or C. Engagement metric A is also more correlated with lifetime value than other metrics.
- **Argument Sketch**. The media organization should use this particular engagement metric going forward because it is predictive of other valuable outcomes.
- **Outcome**. The report going to the media organization about engagement metrics will go to the head of online business. If she signs off on its findings, the selected user engagement metric will be incorporated by the business analysts into the performance metrics across the entire organization. Funding for existing and future initiatives will be based in part on how they affect the new engagement metric. A follow-up study will be conducted in six months to verify that the new metric is successfully predicting revenue.

compare that to this:

> We will create a logistic regression of web log data using SAS to find patterns in reader behavior. We will predict the probability that someone comes back after visiting the site once.

2.8.4 Advocacy Group

Context. This advocacy group specializes in ferreting out and publicizing corruption in politics. It is a small operation with several staff members who serve multiple roles. They are working with a software development team to improve their technology for tracking evidence of corrupt politicians.

Need. The advocacy group doesn't have a good way to automatically collect and collate media mentions of politicians. With an automated system for collecting media attention, it will spend less time and money keeping up with the news and more time writing it.

Vision. The developers working on the corruption project will get a piece of software that takes in feeds of media sources and rates the chances that a particular politician is being talked about. The staff will set a list of names and affiliations to watch for. The results will be fed into a database, which will feed a dashboard and email alert system.

- **Mockup**. A typical alert is that politician X, who was identified based on campaign contributions as a target to watch, has suddenly shown up on 10 news talk shows.

- **Argument sketch**. We have correctly kept tabs on politicians of interest, and so the people running the anti-corruption project can trust this service to do the work of following names for them.

Outcome. The media mention finder needs to be integrated with the existing mention database. The staff needs to be trained to use the dashboard. The IT person needs to be informed of the existence of the tool and taught how to maintain it. Periodic updates to the system will be needed in order to keep it correctly parsing new sources, as bugs are uncovered. The developers who are doing the integration will be in charge of that. Three months after the delivery, we will follow up to check on how well the system is working.

You should not skip this planning under any circumstances but you may find resistance to such an overt approach (I have). However, you need to attach your efforts to real organizational needs and you need to constantly communicate progress with stakeholders. You *may* encounter some resistance if you're working with people who thought data science was going to turn their pile o' data into a pile o' gold. But even implicitly adhering to this framework will guide expectations.

2.9 Resistance to Data Science

> Resistance is futile. – The Borg
>
> Resistance is not futile. – Jean Luc Picard

I wish it were all beer and skittles but that would be a lie. When you work as a data scientist or just do data science you will sometimes be picking around the organization's sacred cows. There are going to be problems. And sometimes the problem is going to be you.

1. People say they want data science but don't want to give up control.
2. People used to calling the shots may feel threatened.
3. People may not cooperate with you.
4. There may be a lot of folksy wisdom in the organization.

Regardless of how glamorous you think your job is, however, you are not the most important function in the organization.

People say they want data science, they want to be data-driven, they want data scientists…but they may be unwilling to give up the control that entails. Currently, most organizations operate by a simple rule: the highest paid person in the room decides. That doesn't mean that options aren't generated by everyone else but ultimately the highest paid person in the room has the responsibility. Or they operate by passive-aggressive consent. The chair/manager suggests something, everyone is intimidated into following along and when it goes wrong, the "team" is responsible. Problems with "group think" not withstanding, this view is very often in conflict with being data-driven.

I have had several instances of this in my career so far. At Company Y, we were constantly offering promo codes in order to boost sales. Yet the data science team had proven that promo codes only shifted revenue. People didn't buy more, they bought sooner and generally things they were already going to buy from us anyway. They just ended up buying them at a discount.

At Company X, I was hired to do more sophisticated modeling that would be ground breaking for the industry. However, one of the managers wanted to just hack something together from what everyone else had already been doing. The manager won.

But what does it actually mean to be data driven? Even if a company isn't as bad as suggested above, they may still only be *data inspired*. What's the difference? If you (your team, your division, your company) are **data driven**, then you decide *ahead of time* what the decision will be, based on the data. You can think of this as the 7th Step in the Kaggle questions or part of the Outcome in the CoNVO:

7. What will we do if the results are favorable? What will we do if the results are not favorable?

If you are merely **data-inspired**, then you wait until you see the results and mix them with the traditional decision making process. In this situation, having a data science team may or may not even make an actual difference to the company.

It's worth emphasizing that the dichotomy between data-driven and data-inspired isn't a claim that the data speaks for itself. That never happens. It is the simple (in theory) but difficult (in practice) principle that you will decide what to do before you see the results. The results are always a combination of domain theory, the model, and the data.

A regular obstacle to being data driven is that the people you work with may feel threatened. As an embedded function, working with stakeholders and decision makers, you are going to run into some resistance from people who are not used to making decisions this way or doing things differently or just don't want to change their workflow.

One major obstacle is that we reward people for their seeming contribution. It is well known that the vast majority of stock brokers over the long run do not do better than the market. And yet stockbrokers who beat the market in any given year are given bonuses. They are feted. They are slapped on the back. Of course, it's unlikely they'll repeat that performance next year. But a year's a long time and in 365 days, someone else will beat the averages and be crowned victorious instead.

A marketing person may feel slighted that they are not picking the blue button that wins the company millions of dollars (and if it didn't, at least they tried, right? Maybe next time! Lots of bro slaps and beers.). The same is going to be true of designers who work hours on features, colors schemes, etc and then you mention “A/B testing”. Their response is, *what about my artistic vision*?

None of these people is wrong but there may have to be changes in what is valued in the company. The creative contribution in this case is to generate options (whether the button is blue or green) and not necessarily picking the *winning* option. And, honestly, there are cases where A/B testing isn't appropriate. That doesn't mean you shouldn't collect data on the user experience to see which features are *actually* being used and how.

Another obstacle to good data science is that people say they want data science but are not actually willing to be involved. Good data science requires communication and sometimes that means meetings. Talking to you may never hit priority 1. There may always be something more important. It's very easy to put off participating in a process that could take months to see any results…results that cannot even be guaranteed. Don't let this happen. Be determined. You need to talk to these people not only to make sure that your efforts are really solving their problems (and make sure you are solving their problems) but also because they are the source of domain knowledge you need to do your job.

People are going to have beliefs about various aspects of the organization's operations. "Everybody knows that our biggest donors are older women whose husbands have passed away." When the data shows that your non-profit is anomaly, you need to be a bit tactful in revising the prevailing "folksy wisdom".

Unless you are an actual data science consultancy, you are overhead. You do not make the goods and services. You do not sell the goods and services. You are overhead. You may improve the production of goods and services. You may improve the sales of goods and services. Remember this. You are not the *most* important function in your organization. Still, you are *an* important function in your organization.

2.10 Conclusion

The Data Science Process is the general context in which the various data science skills are executed. We start with ASK and find a question or problem that solves business needs. With GET, we obtain data suitable for answering the question or solving the problem. We EXPLORE the data to make sure the data is what we expected and to get a general sense of the data. Next we MODEL the data maybe using deep learning or maybe using something as simple as an average. Finally, we REPORT on our findings to that they can help stakeholders make informed decisions (or they find their way into our products).

Data science teams and projects mature over time. We start with the basics (basic data, basic questions, basic models) often with only a few team members and poor data infrastructure. We look for breadth at the start because the first model is often the one that gives the biggest ROI. Even in a mature data science team, new areas may start out here. The main difference is that you'll have some experience with the various growing pains.

When I worked at Company X, we spent nearly two years working on recommendations for email. After those two years, the process was fairly well established and ironed out. But in the beginning there would be bugs and constant questions like "Why is my email this way?". When we started personalizing the website, we went through the entire process all over again. "Why is the website this way?" We kept having to remind people, "we went through this with email, remember?".

These maturation steps are covered both by the Stages of Data Science and Agile Data Science Pyramid. However, because a project stops at using averages and rates and never moves to logistic regression, decision trees or deep learning, you shouldn't think it was immature. Project should always use the appropriate technique.

Determining the problem to solve and the technique to use can be determined by consulting stakeholders during the ASK step. Although Kaggle suggested a good set of questions to start with at the ASK step, we used Shron's CoNVO framework to further flesh out the kinds of things we should discuss with stakeholders and plan up front.

CoNVO stands for Context, Need, Vision, and Outcome. In talking to stakeholders and identifying a problem, we also identify the *contexts* of the problem and the actual *need* that is to be satisfied. We establish a *vision* for the solution that includes *mockups* and *argument sketches* and establish the *outcome* we want from solving the problem or answering the question.

This is probably the "softest" chapter in the text but arguably the most important. No matter how good your technical skills, if you cannot identify a good and appropriate problem facing your organization and apply an appropriate solution and communicate your results, data science initiatives will constantly experience "existential threats".

2.11 Review

1. What is the Green Cloud model of the data science process?
2. What four factors may influence the stages and maturity of a data science team or project?
3. What are the Booz Allen Hamilton Stages of Data Science?
4. What are the layers in the Agile Data Science Pyramid?
5. How do the Stages of Data Science and the Agile Data Science Pyramid relate to the Green Cloud model?
6. What does Shron's CoNVO stand for and how is it used? Where does it fit into the Green Cloud?
7. The Vision includes Mockups and Argument Sketches. Describe what these two things are.
8. What are the Six Phases of Data Mining described in CRISP-DM?
9. What are the Laws of Data Mining relevant to the ASK step?
10. What's the difference between being Data-Driven and being Data-Inspired?

2.12 Exercises

1. Reverse engineer the application of the Green Cloud to one of the case studies from the previous chapter or one you have found.
2. Reverse engineer the CoNVO for two of the case studies in the previous chapter or ones that you have found.
3. Comparing the various case studies, can you guess at the various organizations' data science maturity?

2.13 Additional Resources

Elena Grewal - One Data Science Job Doesn't Fit All (blogpost)

Ganes Kesari - What are the 3 Stages where Data Science Teams Fail? (blogpost)

Booz Allen Hamilton - The Field Guide to Data Science (website, PDF)

Russell Journey - Agile Data Science (amazon)