Solving Problems With Data

Nick Eubank

Jan 22, 2024

CONTENTS

This is the beginning of a textbook by Nick Eubank.

Few fields have shown as much promise to address the world's problems as data science. At the same time, however, recent years have also made clear that today's global challenges will not be met by simply "throwing data science at the problem" and hoping things will work out. Even in business, where many assume that Artificial Intelligence is a sure ticket to profits, a major recent study found only > 11% of businesses that had piloted or employed Artificial Intelligence had reaped a sizeable return on their AI investments.

How, then, should a burgeoning data scientist approach this field full of such promise but also so many pitfalls? And why have so many data science endeavors failed to deliver on their promise?

The answer lies — at least in significant part — in a failure to provide students with a systematic approach to bringing the techniques learned in statistical modeling and machine learning courses to bear on real-world problems. Data science curricula usually begin with coding, statistics, and model evaluation techniques. All too often, however, that's where they stop. But while the hardest part of data science *classes* is often fitting a model well or getting a good AUC score, the hardest part of being an effective *professional* data scientist is ensuring that the models being fit and the results being interpreted actually solve the problem that motivated you (or your stakeholder) in the first place.

This book is designed to fill this gap between neatly curated classroom exercises and the messiness of the real world. It will provide a unified perspective on how the perspectives and statistical tools learned in other courses complement one another, and *when* different approaches to data science are most appropriate to use. More importantly, though, it provides a framework for understanding your goals as a data scientist, and how to achieve them.

Note: Is this book for me?

You'd be forgiven, serious data scientist, for flipping through this book and finding yourself thinking "Hmmm… I don't see many equations in this book. Is this really for me, the serious data scientist?" But worry not.

This book may not be the *most* helpful resource when it comes to preparing for technical interviews (though the detailed discussion of Causal Questions in later chapters likely would be). And your impression is correct — this book contains more case studies than proofs. But don't be fooled — this is not a "light and fuffy" book that waves vaguely in the direction of statistical concepts so you can discuss them at cocktail parties.

This book *takes as given* that you've already been introduced to statistical inference and machine learning, and you feel comfortable with the core concepts of implementing and evaluating stats and ML models (evaluating a model's AUC, cross-validation, hypothesis testing, train-test sample splits, etc.). Those concepts will be treated as "assumed knowledge."

This book is about what comes *before* and *after* you faithfully fit a model to a dataset in a robust manner. By *before*, I mean that it covers how you decide what questions to answer, what data to collect, and what models to consider using. And by *after*, I mean we will discuss how one evaluates whether a result is likely to generalize, whether a model is safe to deploy, and where to go from there.

It is, in other words, about everything you need to know *beyond* the purely technical. And while that may be the part of data science that doesn't feel as exciting, the ability to reason about problems and think critically about the appropriate use of data science tools is what will get you promoted after you ace that technical interview. And that same ability to think critically is also what will prevent you from being replaced by the next generation of auto-ML tools or Large Language Models.

Part I

Part

CHAPTER

ONE

SOLVING PROBLEMS WITH DATA

1.1 Introduction

Few fields have shown as much promise to address the world's problems as data science. Today, data science is being used to develop climate models to improve our understanding of global climate change and mitigate its effects. It is being used in medicine to speed drug discovery, improve the quality of x-rays and MRIs, and to ensure that patients receive appropriate medical care. Data science is used in courtrooms to fight for fair elections and electoral maps, and by data journalists to document and communicate to readers the injustices prevalent in our criminal justice system and issues in policing. Data science also enables new technologies that have huge potential to improve our lives. Autonomous drones are delivering blood and medical supplies to rural health clinics from Rwanda to North Carolina, and driver-aid features continue to make progress in reducing the over 30,000 traffic deaths and millions of injuries that occur in the US alone every year. Nearly no facet of business has gone untouched by the recent revolution in data analytics, from song and movie recommendation engines on Netflix, Spotify, and Apple's App Store to the use of personalized, targeted advertisements used to ensure businesses can make the most of their advertising revenue.

At the same time, however, recent years have also made clear that today's global challenges will not be met by simply "throwing data science at the problem" and hoping things will work out. Even in business, where many assume that Artificial Intelligence is a sure ticket to profits, a major recent study found only 11% of businesses that had piloted or employed Artificial Intelligence had reaped a sizeable return on their AI investments. In recent years we've also seen near endless examples of data science tools reinforcing racial and gender inequities, like algorithms discriminating against female job candidates at Amazon, prioritizing White patients over Black patients for kidney transplants and preventative care, and being more likely to incorrectly identify Black defendants than White defendants as being a "danger to society" when providing risk assessments to judges deciding on pre-trial release, bail and sentencing. And even companies like Meta's own research have shown its algorithms drive political polarization and division among users, and push users into extremist groups.[1]

How, then, should a burgeoning data scientist approach this field full of such promise but also so many pitfalls? And why have so many data science endeavors failed to deliver on their promise?

The answer lies — in significant part — in a failure to provide students with a systematic approach to bringing the techniques learned in statistical modeling and machine learning courses to bear on real-world problems. Data science curricula usually begin with coding, statistics, and model evaluation techniques. All too often, however, that's where they stop. But while the hardest part of data science *classes* is often fitting a model well or getting a good AUC score, the hardest part of being an effective *professional* data scientist is ensuring that the models being fit and the results being interpreted actually solve the problem that motivated you (or your stakeholder) in the first place.

This book is designed to fill this gap between neatly curated classroom exercises and the messiness of the real world. It will provide a unified perspective on how the perspectives and statistical tools learned in other courses complement one

[1] Recent reporting by the Wall Street Journal has shown that Facebook's own research has confirmed what many outside experts have long argued: the way its recommendation engines prioritize content that results in "user engagement" (clicks, shares, comments) ends up promoting partisan, polarizing, sensationalist, or extreme content. In addition, their own research has also shown that group recommendations are contributing to extremism. According to one internal presentation, "64% of all extremist group joins are due to our recommendation tools" like *Groups You Should Join* and other discovery tools.

another, and *when* different approaches to data science are most appropriate to use. More importantly, though, it provides a framework for understanding your goals as a data scientist, and how to achieve them.

Note: Is this book for me?

You'd be forgiven, serious data scientist, for flipping through this book and finding yourself thinking "Hmmm… I don't see many equations in this book. Is this really for me, the serious data scientist?" But worry not.

This book may not be the *most* helpful resource when it comes to preparing for technical interviews (though the detailed discussion of Causal Questions in later chapters likely would be). And your impression is correct — this book contains more case studies than proofs. But don't be fooled — this is not a "light and fuffy" book that waves vaguely in the direction of statistical concepts so you can discuss them at cocktail parties.

This book *takes as given* that you've already been introduced to statistical inference and machine learning, and you feel comfortable with the core concepts of implementing and evaluating stats and ML models (evaluating a model's AUC, cross-validation, hypothesis testing, train-test sample splits, etc.). Those concepts will be treated as "assumed knowledge."

This book is about what comes *before* and *after* you faithfully fit a model to a dataset in a robust manner. By *before*, I mean that it covers how you decide what questions to answer, what data to collect, and what models to consider using. And by *after*, I mean we will discuss how one evaluates whether a result is likely to generalize, whether a model is safe to deploy, and where to go from there.

It is, in other words, about everything you need to know *beyond* the purely technical. And while that may be the part of data science that doesn't feel as exciting, the ability to reason about problems and think critically about the appropriate use of data science tools is what will get you promoted after you ace that technical interview. And that same ability to think critically is also what will prevent you from being replaced by the next generation of auto-ML tools or Large Language Models.

1.2 Solving Problems with Data: An Overview

The remainder of this chapter provides an overview of the key concepts of this book. All concepts discussed here will also be covered in greater detail in future readings. To help readers understand those more detailed readings in their proper context, however, it is important you first get an overall sense of the approach to data science being advocated in this book.

1.2.1 Specifying the Problem

The first step in solving any problem is *always* to carefully specify the problem. While this may seem trivial, properly articulating the core problem one seeks to address can be remarkably difficult. Moreover, because everything you will do *after* articulating your problem is premised on having correctly specified your objective, it is *the* greatest determinant of the success of your project. The most sophisticated, efficiently executed, high precision, high recall model in the world isn't worth a lick of good if the results it generates don't solve the problem you or your stakeholder need solved.

Specifying your problem not only ensures that your subsequent efforts are properly directed, but it can also radically simplify your task. Many times problems only *appear* difficult because of how they are presented. As Charles Kettering, Head of Research at General Motors from 1920 to 1947 once said, "A problem well stated is a problem half solved."

How do you know if you've "clearly articulated the problem," and how should you go about refining your problem statement with your stakeholder? Those are topics we will discuss in detail in the coming chapters, as well as strategies for using data to help inform this process through iterative refinement of your understanding of the contours of the problem space.

Note: Throughout this book, I will frequently use the term "stakeholder" to refer to the person whose problem that you, the data scientist, is seeking to address. I use this term because, as a young data scientist, you will often be in the position of having to use your data science skills to help someone else. Thus your stakeholder may be your manager, your CEO, or someone at another company you are advising.

However, if you're lucky enough to *not* be directly answerable to someone else, either because you work for yourself or because you're in a field that gives you substantial autonomy like academia, you can simply think of your "stakeholder" as yourself.

If you're interested in developing a consumer-facing product (e.g., you're an independing developer whose thinking of creating a new data-science-based web app), you may also find it useful to think of your customer of the stakeholder, since very few products are successful if they don't solve a problem customers face.

1.2.2 Solving Problems Through Answering Questions

Once we have successfully articulated our problem, we must then figure out how to solve it. As data scientists, we are somewhat restricted in the types of solutions to which we have access; nobody hires a data scientist to call donors to raise funds for cancer research, for example, or invent a new semiconductor manufacturing technique. Rather, as we will explore in detail in this book, all data science models and algorithms can be fundamentally understood as instruments for **answering questions** about the world using quantitative methods.

Answering Questions

All data science models and algorithms can be fundamentally understood as instruments for **answering questions** about the world using quantitative methods.

In light of that fact, we can reframe the challenge of a data scientist from the more amorphous task of just "figuring out how to solve the problem" to the more concrete "figure out what question, if answered, would make it easier to solve this problem."

Once we've articulated a question to answer we can turn to choosing the best tool for generating an answer. But it is worth emphasizing this point — it is only at this stage of our project—not at the beginning!—that we start thinking about what statistical method, algorithm, or model to use.

Our job as data scientists is never to just grab the trendiest tool for a given type of question. Rather, we must recognize and evaluate the strengths and weaknesses of different tools available to us *in the context of the specific problem we are seeking to address.*

1.2.3 Types of Questions

While this may seem an impossible task given the sheer multiplicity of data science methods available today, nearly all data science questions we may wish to answer fall into one of three categories:[2]

- Exploratory Questions: Questions about large-scale patterns in the data.
  - Useful for understanding the problem space better and prioritizing subsequent efforts.
- Passive Prediction Questions: Questions about likely outcomes for individual observations or entities.
  - Useful for targeting individuals for additional attention or automating certain tasks.
- Causal Questions: Questions about the consequences of actions or interventions being considered.

[2] Careful readers may notice that these categories do not include *should questions*, which are sometimes referred to as "proscriptive" or "normative" questions. As we will discuss in detail in an upcoming reading, that is because while data science is an amazing tool for characterizing the world around us, it cannot, on its own, answer questions about how the world *should* be. Answering "should questions" requires evaluating the desirability of different possible states of the world, and that can only be done with reference to a system of values, making them inherently subjective. Data science can help us predict the *consequences* of different courses of action, but it cannot tell us whether those consequences make a given course of action *preferable*.

  - Useful for deciding on appropriate courses of action.

Each of these can play a different but important role in solving problems, and any effort to answer a question of each type will raise similar issues that need to be considered. Thus, by recognizing the *class* of questions we are seeking to answer, we can significantly narrow both the set of data science tools that are appropriate to consider and provide a short list of common considerations to think through.

1.3 Question Types and Their Uses

Understanding these three types of questions — both in terms of how they can be used to help solve problems, and also in terms of the challenges inherent in answering them — is a key objective of this book. Here is a brief introduction to each type of question.

1.3.1 Exploratory Questions

Once you have settled on a problem you wish to address, the next step is often to use data science to better understand the contours of the problem in order to better prioritize and strategize your efforts. As data scientists, our best strategy for this type of investigation is to ask questions about general patterns related to your problem — what I call *Exploratory Questions.*

Why is this necessary? Well, as we'll discuss in a future reading on "stakeholder management," you would be *shocked* at how often stakeholders have only a vague sense of the patterns surrounding their problem. This makes refinement of your problem statement (and thus prioritization of your subsequent efforts) impossible. So before you get too far into any data science project, it's important to ask Exploratory Questions to improve your understanding of how best to get at your problem.

To illustrate, suppose a company hired you because they were having trouble recruiting enough high-quality employees. You *could* ask for their HR data and immediately try to train a neural network to… well, I'm not even sure what you'd want to train it to do right off that bat! And that's a big part of the problem. Getting more high-quality employees is a very general problem, and you could imagine addressing it in any number of ways — you could try and get more people to apply for the job in the first place, you could try and get a *different type* of candidate to apply then is currently applying, you could try and get more high-quality people who are given job offers to accept those offers, or you could help try to increase the number of people who are hired who turn out to be successful hires! But which should you do first?

To help answer this question, we can start by asking a series of Exploratory Questions that, when answered, will aid in your efforts to solve your stakeholder's problem:

- How many job applications are you receiving when you post a job?
- What share of your current job applicants are of high quality?
- If your current applicants come from different sources (online ads, services like Indeed, outreach to colleagues for recommendations, etc.), what share of job applicants from each of these sources are of high quality?
- What share of employees you try to hire accept your offer?
- What share of employees you do hire turn out to be successful employees?

Suppose, for example, only 10% of applicants who receive job offers accept. Then clearly that would seem a place where intervention would be likely to substantially increase the number of high-quality employees being hired. If, by contrast, 95% of applicants accept offers, then that is clearly not a place where you would want to focus.

Similarly, if most applicants are high quality and there just aren't enough of them, then you would probably want to focus your efforts on increasing the number of people who apply in the first place. But if only 2% of applicants seem appropriate to the company, then maybe focus should be put on changing *who* is applying for positions with an eye towards increasing the average quality of applicants.

Answering these questions will likely not, on its own, make it clear exactly where to focus your efforts. Your stakeholder may look at the fact that only 2% of applicants are appropriate and say "That's fine — we have so many applications that the *absolute number* of quality applicants is actually high enough, and it's easy to filter out the bad applicants." But these are numbers you can bring back to your stakeholder to discuss and use to zero in on the specific facet of their problem that is most amenable to an impactful solution.

Generating answers to these types of Exploratory Questions doesn't have the same "coolness factor" as using expensive GPUs to train deep learning models. But it is precisely this type of analysis that will help ensure that when if you *do* later run up a giant bill renting GPUs, at least that money will have been spent addressing a part of your stakeholder's problem that matters.

Note: The term *Exploratory Data Analysis* (EDA) is often used in statistics courses to describe the process of poking around in a new data set before fitting a more complicated statistical model. Answering Exploratory Questions will often use some of the same tools used for EDA, but "answering Exploratory Questions" is **not** synonymous with EDA.

As it is commonly used, EDA (as the name implies) is often focused on getting to know a piece of *data*. It entails learning what variables are in a dataset, how they are coded, and *sometimes* also looking at general patterns in a given dataset (*we will discuss the conceptual issues surrounding EDA in more detail in a later reading*.).

Answering Exploratory Questions, by contrast, will often be far more involved. Answering an important Exploratory Question may require you to actively seek out new datasets, merge data from different sources together, and maybe even do novel data collection.

Moreover, where EDA is often viewed as just a box to check on the road to model fitting, answering Exploratory Questions can often be an end goal in-and-of itself.

How do we answer Exploratory Questions? As we'll discuss in later chapters, at times Exploratory Questions can be answered with simple tools, like scatter plots, histograms, and the calculation of summary statistics like means and medians. Other times, however, it may require more sophisticated methods, like clustering or other unsupervised machine learning algorithms that can, say, identify "customer-types" in a large dataset of customer behavior.

Regardless of the tool used, however, the goal is always to identify patterns in the data that are salient to understanding your stakeholder's problem.

1.3.2 Passive Prediction Questions

Answering Exploratory Questions helps you to prioritize your efforts and improve your understanding of your stakeholder's problem. Often you will even bring the answers you generate to Exploratory Questions back to your stakeholder and use them to refine your problem statement in an iterative loop. But what then? As all data science tools are fundamentally tools for answering questions, we return to asking "What question, if answered, would help solve my problem?"

Many problems can be solved if we can answer questions about the future outcomes or behaviors of *individual entities* (people, stocks, stores, etc.). This type of question may take the form "Given this new customer's behavior on my website, are they likely to spend a lot over the next year?" or "Given the symptoms of this patient and their test results, how likely are they to develop complications after surgery?" I term these questions about outcomes for individual entities *Passive Prediction Questions* as they are questions about what is likely to happen if we do not intervene in some way (i.e., if we remain passive).

Because Passive Prediction Questions are questions about individual entities, they don't necessarily have one "big" answer. Rather, Passive Prediction Questions are answered by fitting or training a model that can take the characteristics of an individual entity as inputs (e.g., this patient is age 67, has blood pressure of 160/90, and no history of heart disease) and spitting out an answer *for that individual* (given that, her probability of surgical complications is 82%).

This differentiates Passive Prediction Questions from Exploratory Questions. Being questions about general patterns in the world, Exploratory Questions will tend to have discrete answers. If a company gives you their hiring data, you can answer the question "What share of employees you try to hire end up accepting your offer?"

With Passive Prediction Questions, by contrast, the first question to ask is often one of feasibility: “Given data on new customer behavior on my website, **can I** predict how much they are likely to spend a lot over the next year?” But you then answer that question by training a model that can answer the question you really care about for any given customer: “Given this new customer’s behavior on my website, are they likely to spend a lot over the next year?”

This ability to make predictions about future outcomes is obviously of tremendous use to stakeholders as it allows them to tailor their approach at the individual level. A hospital that can predict which patients are most likely to experience complications after surgery can allocate their follow-up care resources accordingly. A business that knows which customers are more likely to be big spenders can be sure that those customers are given priority by customer care specialists.

But the meaning of the term “Prediction” in Passive Prediction Questions extends beyond “predicting the future”. Passive Prediction Questions also encompass efforts to predict how a third party *would* behave or interpret something about an individual if given the chance.

For example, suppose our hospital stakeholder wanted to automate the reading of mammograms, so rural hospitals without full-time radiologists could give patients diagnoses more quickly (or, more cynically, pay fewer radiologists).[3] We can think of this process of reading mammograms as answering the question “if a radiologist looked at this particular scan, would they conclude the patient had cancer?”

The value of this type of prediction to stakeholders is likely also self-evident, as it opens the door for automation and scaling of tasks that would otherwise be too costly or difficult for humans. Indeed, answering this question is the type of task for which machine learning has become most famous. Spam filtering amounts to answering the question “If the user saw this email, would they tag it as spam?” Automated content moderation amounts to answering “Would a Meta contractor conclude the content of this photo violates Facebook’s Community Guidelines?” Indeed, even Large Language Models (LLMs) like chatGPT, Bard, and LLaMA can be understood in this way, as we will discuss later.

Note: Thinking of training an algorithm to read a mammogram as “predicting how a radiologist *would* interpret the mammogram if given the chance” may seem a little strange, but this framing is both more accurate and more conceptually useful than other ways of thinking about these models. That’s because many data science problems are solved using a practice called *supervised machine learning* in which a statistical model is “trained” using data that a human has already analyzed. Any real mammogram analyzing algorithm, for example, is likely to be trained using examples of mammograms that human radiologists had reviewed and labeled as either containing suspicious abnormalities or not.

But a critical feature of this supervised machine learning approach is that the model is not actually being being taught to “find cancer” *per se*; it is being taught to emulate the behavior of the human radiologists who labelled the training data. Or, expressed differently, the model is being trained to answer the question “If one of the radiologist who labelled my training data looked at this scan, would they diagnose the patient with cancer?”

This distinction is subtle, but it is important because it helps us to understand why any model we train in this way will inherit all of the biases and limitations of the radiologists who created the data used to train the algorithm. If, for example, our radiologists were less likely to see cancer in denser breast tissue, that bias would also be inherited by the algorithm.

(We call the inevitable existance of some difference between between what we *want* the algorithm to do — in this case, detect cancer — and *what it is actually being trained to do* — predict how a radiologist would interpret the scan — an “alignment problem.”)

The big differentiator between Exploratory Questions and Passive Prediction Questions is that Exploratory Questions are questions about *general patterns* in the data, while Passive Prediction Questions are questions about *individual observations or entities*.

It is worth emphasizing that this is a distinction *in purpose*, not necessarily in the statistical tools that are most appropriate to the task. A linear regression, for example, may be used for answering either type of question, but in different ways. To answer an Exploratory Question, we might look at the coefficients in a linear regression to understand the partial correlations between variables in the data. To answer a Passive Prediction Question, we might only look at the predicted values from the regression model.

[3] Mammograms are x-rays of breast tissue used for the detection of breast cancer.

But even if the same *type* of model can be used for both purposes, how one *evaluates* the model depends entirely on the purpose to which it is being put. When answering an Exploratory Question through the interpretation of regression coefficients, the size of the standard errors on the coefficients is critical. When making predictions, by contrast, one may not care about the coefficients of a model at all! So long as the R-squared is high enough (and other diagnostics seem good), one can simply use the predicted values the regression generates without ever looking "inside the box."

As such, there's no simple mapping between statistical or machine learning methods and the type of questions you aim to answer. However, *in general*, Passive Prediction Questions are most commonly the domain of methods that fall under the label "supervised machine learning," which encompasses everything from linear regression to neural networks.

1.3.3 Causal Questions

The word "passive" in "Passive Prediction Questions" is there because many data science problems entail predicting what outcomes would occur absent intervention. For example, when answering the question "Given their case history, how likely is this patient to experience post-surgical complications?" we don't actually want to know how likely they are to experience complications — we want to know how likely they would be to experience complications *if the status quo prevails and our behavior doesn't change.* Our hope, after all, is that by learning that a certain patient is likely to experience complications we can act to prevent that outcome!

Causal Questions, by contrast, are questions about predicting the *effect* of actions *we may choose to take*. Causal Questions arise when stakeholders want to do something — buy a Superbowl ad, change how the recommendation engine in their app works, authorize a new prescription drug — but they fear the action they are considering may be costly and not actually work. In these situations, stakeholders will often turn to a data scientist in the hope that the scientist can "de-risk" the stakeholder's decision by providing guidance on the likely effect of the action *before* the action is undertaken at full scale.

Causal Questions, therefore, take the form of "What is the effect of an action X on an outcome Y?"—or more usefully, "If I do X, how will Y change?". Nearly anything can take the place of X and Y in this formulation: X could be something small, like changing the design of a website, or something big, like giving a patient a new drug or changing a government regulation. Y, similarly, could be anything from "how long users stay on my website" or "how likely are users to buy something at my store" to "what is the probability that the patient survives".

In my view, Causal Questions are perhaps the hardest to answer for two reasons. The first is that when we ask a Causal Question, we are fundamentally interested in *comparing* what our outcome Y would be in two states of the world: the world where we do X, and the world where we don't do X. But as we only get to live in one universe, we can never perfectly know what the value of our outcome Y would be in *both* a world where we do X and one where we don't do X—a problem known as the **Fundamental Problem of Causal Inference** (causal inference is just what people call the study of how to answer Causal Questions).

But the second reason is Causal Questions land on the desk of data scientists when a stakeholder wants to know the likely consequences of an action *before they actually undertake the action at full scale.* This may seem obvious, but it bears repeating — not only is answering Causal Questions hard because we never get to measure outcomes in both a universe where our treatment occurs and also a universe where it does not (the Fundamental Problem of Causal Inference), but answering Causal Questions is *also* hard because stakeholders want to know about the likely consequences of an action they aren't ready to actually undertake!

As a result, the job of a data scientist who wants to answer a Causal Question is to design a study that not only measures the effect of a treatment but also does so in a setting that is enough like the context in which the stakeholder wants to act that any measured effect will generalize to the stakeholder's context.

1.4 Bringing It All Together

In this introductory chapter alone, we've already covered a substantial amount of material. We've discussed the importance of problem articulation, the idea that the way data scientists solve problems is by answering questions, and the three types of questions data scientists are likely to encounter.

It's easy to see how this framework might result in a sequential development of a project. First, a hospital comes to you concerned about the cost of surgical complications. So you:

1. Work with them to more clearly define the problem ("Surgical complications are extremely costly to the hospital and harm patients. We want to reduce these complications in the most cost-effective manner possible.")
2. You answer some Exploratory Questions ("Are all surgical complications equally costly, or are there some we should be most concerned about?").
3. You develop a model to answer a Passive Prediction Question ("Given data in patient charts, can we predict which patients are most likely to experience complications?") so the hospital can marshal its limited nursing resources more effectively.
4. The hospital then comes back to you to ask the Causal Question "Would a new program of post-discharge nurse home visits for patients identified as being at high risk of complications reduce complications?"

In reality, however, while it is important that some steps come before others (if you don't start by defining your problem, where do you even start?), real projects are never so linear. The reality is that you will constantly find yourself moving back and forth between different types of questions, using new insights gained from answering one question to refine your problem statement and articulate new questions.

Nevertheless, by using this framework as a starting point, and using this taxonomy to help you recognize (a) the type of question you are asking, and (b) the reason you are seeking to answer a given question even when iterating through a project, you will see tremendous gains in your ability to please your stakeholders by staying focused on the problems they need addressed.

1.5 Reading Reflection Questions

At the end of many readings in this book you will find a set of "reading reflection questions." As the name implies, these are questions meant to help readers reflect on what they've read, as well as to draw attention to key points from the chapter.

- What is the purpose of this book? What problem in data science education does it aim to address?
- What is **the** most important task for a data scientist hoping to successfully help their stakeholder?
- In the view of this book, all data science tools are tools for doing what? Do you agree?
- What are the three types of questions a data scientist is likely to encounter? What is the primary purpose of each type of question?
- Does one always move through the questions presented here in the same order?

CHAPTER

TWO

STAKEHOLDER MANAGEMENT & SOLVING THE RIGHT PROBLEM

In Douglas Adams' comedic sci-fi classic *Hitchhiker's Guide to the Galaxy*, a race of hyperintelligent pandimensional beings set out to build a massive supercomputer the size of a city to solve the mysteries of the cosmos once and for all. When they turned on the computer, named Deep Thought, they announced that:

> "The task we have designed you to perform is this. We want you to tell us… the Answer!"
>
> "The Answer?" said Deep Thought.
>
> "The Answer to what?"
>
> "Life!" urged one designer.
>
> "The Universe!" said another.
>
> "Everything!" they said in chorus.
>
> Deep Thought paused, then answered, "Life, the Universe, and Everything. There is an answer. But," Deep Thought added, "I'll have to think about it."

Seven and a half million years later, when Deep Thought had *finally* finished its calculations, the descendants of those designers assembled to learn the result of their ancestors' work.

> "Er …good morning, O Deep Thought," said Loonquawl [one descendants] nervously, "do you have … er, that is …"
>
> "An answer for you?" interrupted Deep Thought majestically. "Yes. I have."
>
> The two [descendants] shivered with expectancy. Their waiting had not been in vain.
>
> "There really is one?" breathed Phouchg [the other descendant].
>
> "There really is one," confirmed Deep Thought.
>
> "To Everything? To the great Question of Life, the Universe and Everything?"
>
> "Yes. […] Though I don't think," added Deep Thought, "that you're going to like it."
>
> […]
>
> "All right," said the computer, and settled into silence again.
>
> The two fidgeted.
>
> The tension was unbearable.
>
> "Forty-two," said Deep Thought, with infinite majesty and calm.
>
> "Forty-two!" yelled Loonquawl. "Is that all you've got to show for seven and a half million years' work?"
>
> "I checked it very thoroughly," said the computer, "and that quite definitely is the answer. I think the problem, to be quite honest with you, is that you've never actually known what the question is."

> “But it was the Great Question! The Ultimate Question of Life, the Universe and Everything,” howled Loonquawl.
>
> “Yes,” said Deep Thought with the air of one who suffers fools gladly, “but what actually is it?”
>
> A slow stupefied silence crept over the men as they stared at the computer and then at each other.
>
> “Well, you know, it’s just Everything … everything …” offered Phouchg weakly.
>
> “Exactly!” said Deep Thought. “So once you do know what the question actually is, you’ll know what the answer means.”[1]

In addition to establishing the premise for one of the greatest comedic science fiction novels in human history,[2] I feel this passage perfectly exemplifies the three things that cause most data science projects to fail.

The first is that we are often so enamored with technology that we have absolute faith that if we just throw our problems at it, it will solve them for us. But it won’t. Without our thoughtful, critical guidance, it can’t. It never has, and it never will.

The second is that the *reason* this doesn’t work is that if we don’t actually figure out what it is we hope the technology will do for us, and make sure that what we’re asking for will actually solve a real problem, technology couldn’t care less. It will do what it has been asked to do — no more and no less. Garbage in, garbage out.

But the third reason data science projects fail that is exemplified in this passage is the most subtle. In this passage, we can see that Deep Thought recognizes the idiocy of the request it has been given by Loonqual and Phouchg, it doesn’t do anything about it. And that, unfortunately, is the cardinal sin committed by most young data scientists — they fail to recognize that helping the stakeholder properly specify their problem is a core part of the job.

And that, unfortunately, is the cardinal sin committed by most young data scientists — they fail to recognize that helping the stakeholder properly specify their problem is a core part of the job.

At this point, you may be thinking “well, isn’t that *their* problem? They’re the ones who asked me to do the wrong thing?” And… yes, in some sense it is. But it’s also yours. Once you leave the classroom, you will no longer be evaluated on the complexity of your model or the aesthetics of your visualizations—you’ll be evaluated on whether you’ve made your stakeholder’s life better. And even if it was the stakeholder who originally misspecified their need, if you fail to correct that error and deliver a result that doesn’t help your stakeholder, then that’s all that will be remembered.

In this chapter, we will discuss the concept of “stakeholder management:” specific ways you can work with your stakeholder to help refine and improve your mutual understanding of the problem you are seeking to solve before you—like the Deep Thought computer in *Hitchhiker’s Guide to the Galaxy*—spend weeks dutifully grinding away to solve a misspecified problem, only to deliver a result to their stakeholder that turns out to not be as helpful as expected.

Not someone who has an obvious stakeholder with whom you can have this type of conversation? Well, stick with me — many of the suggested questions and conversational strategies detailed below are ones I’ve often used in conversation with myself in my academic research, and I assure you they work almost as well when talking to the voices in your head as with another person.

[1] Yes, I recognize that it is wildly indulgent to open a chapter with such a long epigraph. But it’s my book, and if there’s anything to be indulgent about its quotes from *Hitchhiker’s Guide to the Galaxy*, damn it!

[2] Not least for being the only 5-book trilogy of which I am aware!

2.1 Problem Refinement & Stakeholder Management

How then should you — the young data scientist — go about ensuring your efforts are well spent?

There are (sorry) no hard and fast rules for how to work with your stakeholder to better articulate the problem you are seeking to solve. If there were, there would probably be a lot fewer problems in the world, since refining and re-articulating problems is often a major part of what results in them being solved. As Charles Kettering, Head of Research at General Motors from 1920 to 1947 once said, “A problem well stated is a problem half solved.”

> “A problem well stated is a problem half solved.”
>
> - Charles Kettering, Head of Research at General Motors.

Nevertheless, here are some guiding principles to bear in mind. Read these, reflect on these, but most importantly, *review* them from time to time as you begin new data science engagements!

2.1.1 Step 0: Recognize Your Role

If you remember nothing else from this chapter, please remember this: helping your stakeholder better understand their problem is a core part of the job.

Because most stakeholders are older and domain experts in their field, young data scientists tend to err on the side of deference. It is important to be respectful of your stakeholder’s experience and to use their domain expertise, but it is important to also recognize that data science is about *pairing* domain expertise with computational methods and quantitative insights, and neither you nor your stakeholder is likely to have expertise in *both* the substantive domain in question *and* cutting edge quantitative methods. Indeed, if they did, they probably wouldn’t be hiring you![3] So don’t hesitate to speak up! Ask questions, raise concerns, and while you should do so with *some* humility, have confidence in your own expertise.

2.1.2 Step 1: Don’t Assume Your Stakeholder Knows What They Need

A corollary to Step 0 is to not assume your stakeholder understands what they need. So when I say “helping your stakeholder understand their problem is a core part of the job,” I don’t only mean that it’s part of your job *if the stakeholder admits to deep uncertainty about their problem*.” Odds are your stakeholder will come to you with a strong statement of what they think they want, but you should take that as a starting point for discussion, not your mandate.

This is particularly true if your stakeholder comes to you with really specific technical suggestions. Often you will be approached by a stakeholder who, rather than laying out a problem, announces they would like you to do X using some data science tool Y. Occasionally the stakeholder doing this knows exactly what they’re talking about, and you should use Y to do X.

More often, however, you’re dealing with a stakeholder with just enough knowledge to be dangerous (and to drop buzzwords), but not enough to know how best to solve their problem.

Most people ask data scientists for help because they don’t know much about data science (or, worse, they *think* they know about data science but don’t). Again, different rules apply if you’re at Google or Apple, but in most contexts, it’s a good idea to treat implementation details provided by the client as a red herring. Focus on the stakeholder’s *needs*. Only get into implementation details once you feel you understand the problem well.

[3] Obviously there are exceptions to this — if you work for a mature tech company like Google or Meta, you may very well end up working under a manager who knows sides of a problem significantly better than you. In my experience, however, is circumstance is the exception, not the rule.

Focus on the stakeholder's *needs*. Only get into implementation details once you feel you understand the problem well.

2.1.3 Step 2: Abstract the Problem

So how do you help your stakeholder better understand *their* problem?

If I could offer only one piece of advice on how to approach a sticky problem, it would be this: rephrase the problem in a more general manner that abstracts away from the specifics. It's difficult to overstate how often a "unique" sticky problem becomes very straightforward once you realize it's a special case of a more general type of problem, or once you realize that your stakeholder has (often unknowingly) introduced constraints to the problem that aren't actually constraints.

Perhaps my favorite example of this comes from a talk given by Vincent Warmerdam at PyData 2019.

The World Food Program (WFP) is a global leader in food aid provision. As Vincent tells the story — which he reports having heard at an Operations Research Conference — the WFP was struggling with an extremely difficult data science problem: how best to get food from the places it was being grown/stored to the people who needed it most. Essentially, the WFP would receive reports of needs from communities facing food insecurity. One community might report a need for bread and beef, while another might request lentils and meat. The WFP would compile these requests and then set about trying to determine the most efficient way to meet these needs.

This type of logistics problem is an example of a notoriously difficult problem (essentially a version of the Traveling Salesman Problem, which is NP-Complete, if that means anything to you) that companies like FedEx and UPS buy supercomputers to address. But this particular problem was made extra challenging by all the different types of food the WFP was trying to provide communities.

What the WFP realized was that they didn't actually need to provide bread to the village asking for bread. See, humans don't need *bread* to avoid starvation — they need a certain number of calories, a certain amount of protein, and a handful of other nutrients.[4] So when a village asks for bread, rice, or wheat, you can instead think of them asking for carbohydrates. And a village asking for beef or beans is actually asking for protein and iron. So by simply abstracting the task from "How best can we meet all these food requests?" to "How best can we meet the nutritional needs indicated by these requests?" the WFP was able to *dramatically* reduce the number of constraints being imposed on the logistical optimization problem WFP needed to solve, making its task *far* simpler.

2.1.4 Step 3: Ask Questions (Especially Quantitative Ones!)

Be sure to ask a lot of questions of your stakeholder. In particular, I would suggest two types: questions about what success would look like, and questions about the problem itself.

Questions About Success

Getting a sense of where the goalposts are for your stakeholder will both help you know what to target and also help you better understand your stakeholder's understanding of the problem. Make sure to ask questions like:

- How are you measuring the problem? What would you measure to help you know if you were successful in solving the problem?
- How big, in quantitative terms, is this problem?
- How much would you need the current situation to change to call this a success?

[4] As I understand it, calcium, iron, vitamins A, B1, B2, C, and niacin.

Questions About the Problem

The more you know about your client's needs the better, so ask anything that comes to mind. If the client can answer your question, it will help you better understand the situation; if the client can't answer your question you may find that they are suddenly really interested in knowing the answer, and you immediately have some of your first Exploratory Questions to try to resolve.

In the example of the company that wanted to improve recruitment of high-quality employees in the introduction of this book, I suggested that some of the first exploratory questions you might want to investigate would be things like:

- How many job applications are you receiving when you post a job?
- What share of your current job applicants are of high quality?
- What share of employees you try to hire accept your offer?
- What share of employees you do hire turn out to be successful employees?

These are all questions that I would ask my stakeholder in one of our first meetings.

Note: Stakeholder Meetings It is always good to go into meetings with your stakeholder with a clear sense of your objectives — what you hope to communicate, and what information and feedback you need to get before the meeting ends. When your stakeholder is someone you don't get to meet with regularly, it's good practice to detail these objectives and provide them — in writing — to your stakeholder in advance of your meeting. This will not only ensure that you and your teammates are on the same page (as you will all have reviewed the document before sending it to your stakeholder), but also ensure that your stakeholder has adequete time to reflect on any questions or issues you wish to raise.

When it comes to your *first* meeting, however, this practice can feel impractical as you may feel so uncertain about the project that you only know the first few questions you want to ask.

But even in a first meeting, preparation is key. Rather than laying out the new issues you wish to raise and questions you want answered, for a first meeting it's helpful to write out a full *tree* of lines of inquiry you may wish to propose. In other words, for every question you wish to pose to your stakeholder, try to anticipate some likely responses they make provide, then write down a few followup questions to ask if they provide one of those responses.

Time with your stakeholder is *precious*, especially early in a project, make the most of that face time through preparation.

2.1.5 Step 4: Propose Questions You Might Answer

As a data scientist, answering questions about the world is the instrument you have to solve problems. So once you think you have a sense of your stakeholder's needs, turn around and propose a handful of questions and ask them if answering those questions would help solve their problem.

This is important because many people have only a vague sense of what they are likely to get as a "deliverable" from the data scientist. They usually have a vague sense that they will get some type of magic machine (a "magic model" or "magic algorithm") that will just make their problem go away. By concretely framing your deliverable as the answer to a question (or a model that would answer a specific question for each entity like a customer or patient that it encounters), you can get much more valuable feedback before you dive into a problem.

Make Your Questions Specific and Actionable

In developing your questions, it is important to make them specific and actionable. A specific and actionable question makes it very clear what you need to do next. For example, suppose an international aid organization told you they were worried that urbanization in Africa, Asia, and Latin America was impacting efforts to reduce infant mortality. Some examples of specific, actionable questions are: “Is infant mortality higher among recent migrants to urban centers, controlling for income?” or “Are the causes of infant mortality among recent migrants to urban centers different from those living in rural areas?” Reading those questions, you can probably immediately think of what data you’d need to collect, and what regressions you’d want to run to generate answers to those questions.

Vague questions would be “Is urbanization impacting efforts to reduce infant mortality?”, or “Does urbanization affect infant mortality?” Note that when you read these, they don’t seem to obviously imply a way forward.

Perhaps the best way to figure out if your question is answerable is to write down what an answer to your question would look like. Seriously – try it. Can you write down, on a piece of paper, the graph, regression table, or machine learning diagnostic statistics (complete with labels on your axes, names for variables, etc.) that would constitute an answer to your question? If not, it’s probably too vague.

2.1.6 Step 5: Iterate

And here’s the last but perhaps most important step: **iterate.** Bring your work back to your stakeholder as often as possible.

Many stakeholders find the idea of data science mysterious and abstract and will struggle to understand what is and is not feasible. By bringing them intermediate results, the whole process will start to become more concrete for the stakeholder, and it will help them provide you with better feedback.

The way this book is organized suggests a natural flow from problem articulation to answering Exploratory Questions to prioritize efforts, to answering Passive-Prediction Questions to target individuals for extra attention or automate tasks, and finally to Causal Questions to better understand the effects of that extra attention/automation. In reality, however, a good data scientist is always coming back to the stakeholder, updating their plan, and jumping back in the sequence when new questions arise.

2.2 What Solving the Wrong Problem Looks Like

Our discussion up to this point has been a little abstract, so to illustrate what it means to “mis-specifying a problem.” The details of this example are fictitious, but the underlying logic of this example is not; indeed, the insight illustrated by this example is central to one of the biggest pivots in how people think about online advertising.

You have been hired by the advertising division of a fictitious national pizza chain—let’s call it Little Papa Dominos (LPD). LPD spends a *lot* on online advertising, but their resources aren’t being deployed as effectively as they could be. They spend more than most of their competitors, and yet their online sales are lagging.

After consulting industry groups and online advertising experts, they discovered that the rate at which people click their ads (their ads’ *click-through rate*, or CTR) is well below the industry average.

To address the problem, they’ve hired you—a newly minted Data Scientist—to improve the CTR of their ads. They give you a large budget, access to all the cloud computing resources you need, and even a small staff.

“Well,” you reason, “maybe the problem is that our ads aren’t being shown to the right people. After all, it seems unlikely that any ad for pizza—no matter how appealing—is likely to draw a click if it’s shown to a 75-year-old at 7 am.” So you set out to build a statistical model to answer the question, “given a user’s demographics and online behavior, how likely are they to click on one of LPDs ads?” If you can answer that, you figure, LPD can prioritize buying the ad spots for the types of users most likely to click on their ads.

To develop that model, you use your budget to run your ads on different sites and at different times. You then use that data (and those glorious cloud computing resources) to train a machine learning model that predicts whether someone will click on one of your ad based on the user's demographics and ad placement. You try out a few different models, tune the model parameters, and eventually settle on a neural network model with extremely high precision *and* recall. Hooray!

LPD uses the model to target users likely to click their ads, and almost immediately the CTR of their ads increases 5-fold! Not only that, but the share of people who click on ads that go on to buy a pizza has also increased. Everyone congratulates you, and you move on to the next project feeling very smug.

A few months later, though, you are called into a meeting with the LPD advertising team and the company's Chief Financial Officer. They've been looking over the numbers, and despite the huge rise in CTR, they seem to be getting fewer online orders than before you arrived. CTR rates are up, but somehow it isn't generating greater profits.

Can you figure out what went wrong?

OK, this is the place in most books where the authors ask you that question, and you look up at the ceiling for a minute, shrug, and then read on.

But I'm really, *really* serious about this: close your laptop, stand up, set a 5-minute timer on your phone, and go for a walk. Ponder this example. See if you can figure out what's going on. This is *precisely* the kind of problem you will soon face as a professional data scientist, so why not practice trying to think through the problem?

2.2.1 Solving The Wrong Problem

So what happened?

The reason an increased click rate wasn't making LPD richer is that LPD's problem was never the fact they had a low CTR; LPD's *real* problem was that they weren't getting a lot of orders online. And because Little Papa Domino's problem wasn't a low CTR, being able to answer the question "How likely is a given user to click on an ad" *didn't actually solve their real problem*.

What question, if answered, would have helped solve their problem? "Given a user's demographics and online behavior, *how much more likely are they to buy a pizza* from LPD if we show them an ad?"

Or, expressed more succinctly, LPD *thought* their problem was that their ads weren't *getting clicks,* but really their problem was that their ads weren't *driving increased sales*.

The difference is subtle, but critically important: someone clicking an ad doesn't make Little Papa Dominos any money. Someone clicking an ad *and ordering a pizza* doesn't necessarily make LPD any money. Why? Because they may be someone who would have bought a pizza from LPD anyway, whether you showed them an ad or not. The person who was already thinking of ordering a pizza from LPD is *precisely* the type of person your algorithm may have targeted, and who may have clicked the ad to save themselves a Google search!

But the person LPD *wants* to show an ad to isn't the person who was already thinking of ordering a pizza from LPD, it's the person who was thinking of a pizza but wasn't sure who to order it from, or the person who wanted dinner but didn't know what to get. They may be less likely to click the ad than the person who was about to Google "Little Papa Dominos," but their precisely the type of user who is more likely to buy a pizza from LPD as a result of seeing an ad than they would have been otherwise.

Counter-Factual Advertising

Lest you think this example is contrived, it's not. The realization that the goal of ads isn't to maximize clicks but rather to induce the largest possible change in purchasing behavior is one of the most important ideas in online advertising. It has had a huge impact on how online advertising works, and how people evaluate the success of ad campaigns.

Indeed, this is why companies like Meta and Google are so eager to track user behavior across apps and websites. When Meta and Google can "follow" users after they've clicked an ad, they can evaluate ad performance based not on clicks but on customer behavior. When paired with their ability to show ads to some users and not to others and track both groups as they move around the web, Meta and Google can see whether users who see the ads are more likely to make purchases than those who don't. This allows them to estimate the true effect of ads on sales, data they use to improve ad targeting *and* justify higher prices to advertisers.

2.3 Next Up: Types of Questions

Having established the importance of first articulating the problem one seeks to solve, we will shortly turn to developing our understanding of the three types of questions introduced in the first chapter of this book.

First, though, a quick digression into understanding the historical context of data science. This may feel like an odd topic to talk through in a technical data science text, but as we'll see understanding how we got to where we are today is key to successfully navigating modern data science.

2.4 Reading Reflection Questions

- Why should you care if your stakeholder misspecifies their problem?

CHAPTER

THREE

WHAT *IS* DATA SCIENCE: AN HISTORICAL PERSPECTIVE

Given how often the term "data science" gets thrown around, you would be excused for thinking that the meaning of the term was clearly understood. The reality, however, is that if you were to ask ten people working in the field you will almost certainly get ten different descriptions of what it is and what they do.

Part of that is deliberate obfuscation—data science is *so* trendy that everyone wants to claim that what they're doing is data science in order to woo venture capitalists or to win research grants. Indeed, it has been said (half-joking, half-seriously): "Data science is *Artificial Intelligence* when you're raising money, *Machine Learning* when you're hiring, and it's *Logistic Regression* when you actually have to get the job done.

But the ambiguity that surrounds the term "data science" is also the result of the fact that data science is not a mature discipline in the way that computer science, economics, or mechanical engineering are mature disciplines. And, as a young data scientist, that immaturity is important for you to understand, as it is both the source of some of the most exciting opportunities and also some of the biggest challenges you will face.

3.1 The Organization of Academia, Data Science, and You

To explain what the term data science means in practice, we have to start by discussing a bit of the inside-baseball[1] of how academia operates. This may feel esoteric, but it's important to understand because the way academia is organized has shaped the professional training — and thus the language and thought patterns — of most people you will encounter in the data science space. Understanding academia better, as a result, will not only help you understand the material you are exposed to in data science classes better, but also help you relate to your future peers and colleagues.

The idea that academia is deeply fragmented often surprises students, and understandably so. Universities *love* to pay lip service to the importance of interdisciplinarity and are quick to highlight successful interdisciplinary collaborations. But successful interdisciplinary collaborations are so notable precisely because they are the exception, not the rule. The reality is that academic research is starkly divided into disciplinary silos (e.g., computer science, statistics, political science, economics, and engineering). This isn't because researchers aren't *interested* in interdisciplinary collaborations, but rather that their professional imperatives push them to focus their attention on the priorities and language of their own departments and disciplines.[2]

Thus, while the past several decades have seen an unprecedented emergence of new methods across all of academia, the lack of intellectual cross-pollination across academic silos has resulted in disciplines failing to take full advantage of discoveries from other disciplines. Over time, each discipline has developed a perspective on computational methods that emphasizes its own intellectual priorities.

To illustrate, suppose we were interested in using patient data to reduce heart attacks. A computer scientist looking at this problem might use their discipline's methods to *predict* which patients are most likely to experience a heart attack in the

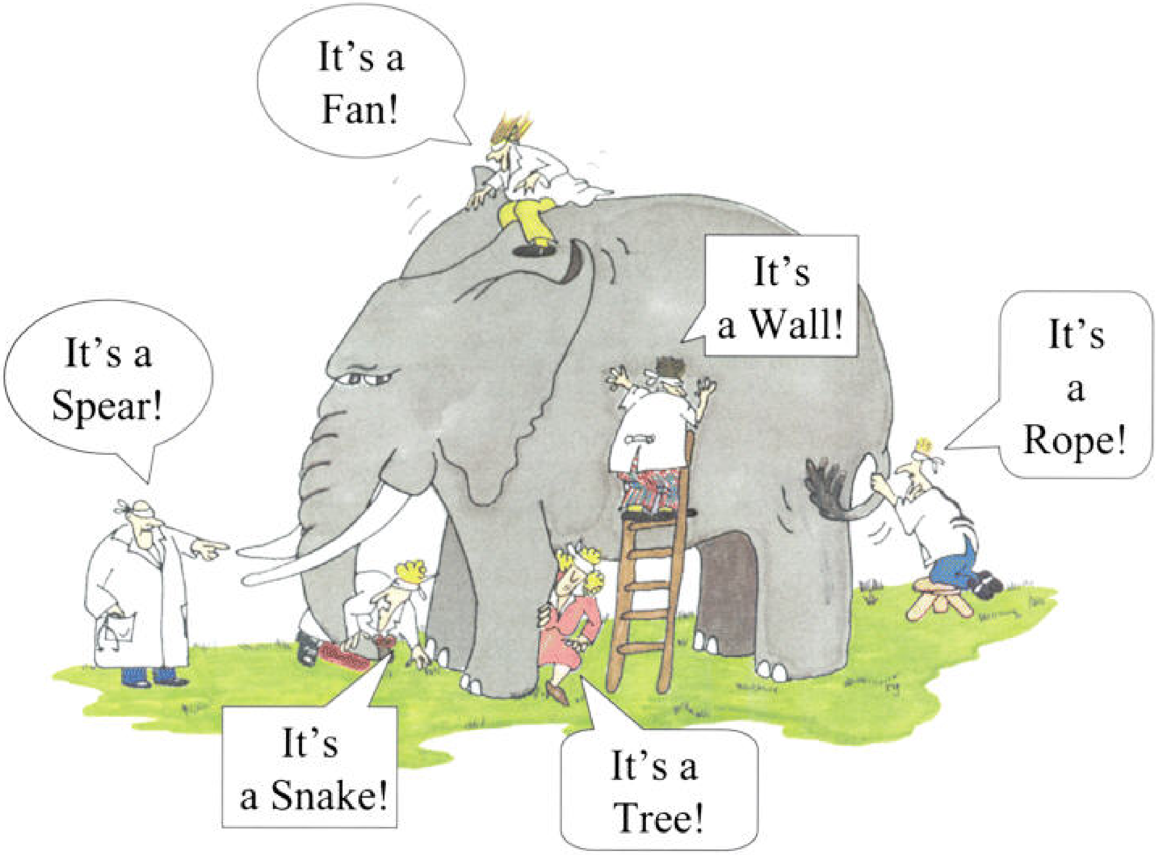
[1] "Inside baseball" refers to the discussion of the idiosyncracies and details of how an institution or system operates internally, something that is often not of interest to people who aren't part of the system.

[2] Nearly all university faculty are hired by established departments like statistics or economics, faculty submit their research to journals specific to their discipline, those journals in turn ask fellow members of the discipline to evaluate their work for publication, and promotions and tenure reviews are managed by the faculty in a faculty member's own department.

future using current patient data; a social scientist might focus on trying to understand the *effect* of giving patients a new drug on heart attack risk; and a statistician might focus on understanding *how confident* we should be in the conclusions reached by the computer scientist and social scientist.

This fragmentation has also resulted in a fragmentation of *language* around data science methodologies. Disciplines often come up with different terminology for the same phenomena, adding another layer of difficulty to efforts to work across departmental silos.

The result is a situation analogous to the Buddhist parable of the blind men and the elephant, wherein a group of blind people come upon an elephant, and upon laying hands on different parts of the elephant, they come to different conclusions about what lies before them. The person touching the tail declares "we have found a rope!", while the person touching the leg declares "we have found a tree!"



(*Note*: Not sure of original source of this image. Found it here, but need to figure out rights prior to anything about this becoming commercial! Lots of pics in public domain if needed, but not blindfolded scientists.)

And yet, as the poet John Godfrey Saxe wrote in his poem *The Blind Men and the Elephant* about this parable many centuries later:

> And so these men of Indostan, Disputed loud and long, Each in his own opinion Exceeding stiff and strong, Though each was partly in the right, And all were in the wrong!

In recent years, however, there has been a growing appreciation of what can be gained from pulling together the insights that have been developed in different fields, despite the challenges of language and professional imperatives to such collaborations. And, at least amongst those who are serious about the development of data science as a discipline and not just a buzzword to use when raising money, is the promise of data science: to unify the different perspectives and methods for analyzing data. Or, to put it more succinctly: to finally see the whole elephant.

While the field is making progress towards "seeing the elephant as a whole," however, as a result of this fragmented origin story, *most* people you will encounter in the world doing data science were trained in one of these academic silos. That means that depending on who you are working with and how they were trained, you may find your future colleagues using terms you've never heard before. And when that happens, it's important to remember that while that *may* be because they're talking about a concept you've yet to encounter, it may also simply be because they're using different language for something you know. Similarly, you may also find senior colleagues unfamiliar with concepts that seem basic to you simply because you were exposed to perspectives that were alien to your colleague's academic silo at the time they were trained. Indeed, given that data science education *is* finally becoming more unified, you should probably expect to learn a lot of ideas that even your more senior colleagues (or rather, especially your more senior colleagues!) were never exposed to.

And therein also lies some of the greatest opportunities. Precisely because of this intellectual fragmentation, there are *lots* of opportunities for taking insights from one intellectual silo and using them to solve problems in another — a kind of "intellectual arbitrage," if you will.

3.2 The Data Analyst / Software Engineering Distinction

In addition to this broader intellectual fragmentation, the world of data science also often feels oddly fragmented around the way people use the tools of data science.

One model of data science is what we will call the "data analyst" approach. Data scientists doing this type of work often collect data to answer specific questions—what is the effect of expanded government health insurance subsidies on mortality? what type of customer should we target with our new advertising campaign?. As a result, when they write code, they write it to be run against a specific set of data to answer a specific question.

The other model is what we will call the "software engineering" approach. Data scientists doing this type of work write software they plan to *deploy* to thousands or millions of users. This is the type of work that gets embedded in the apps on your phone, or that generates your movie recommendations at Netflix. As a result, when these data scientists write code, they are writing more sophisticated and generalizable programs.

To be clear, most data scientists do at least a bit of both types of work—data analysts may often write small programs or packages to aid in types of analysis they do a lot, and software engineers have to prototype and test new programs before they write a version that can be deployed broadly. But most people will eventually choose to specialize in one direction or another, and when you see data science resources in the world—especially ones about programming for data science—bear in mind that depending on *your* proclivities towards on approach or another, not all resources will be well suited to your interests.

I also want to draw attention to this distinction because it's remarkable how dismissive most data scientists will be of the "other" type of data science, and I want to encourage you to both (a)not be so tribal yourself (both flavors of data science have their place, and help solve real world problems!), and (b) not be too surprised when you encounter people with irrationally strong opinions about which approach is the "right" approach to doing data science.

Part II

Types of Questions

CHAPTER FOUR

DESCRIPTIVE V. PROSCRIPTIVE QUESTIONS

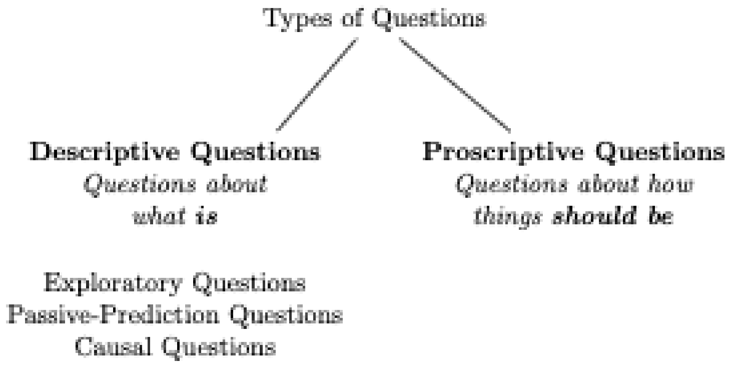
In the chapters that follow, we will discuss Exploratory, Passive-Prediction and Causal in detail. First, though, we must discuss another important concept: the distinction between descriptive and prescriptive questions.

Descriptive Questions are questions about the state of the world and include all the questions and examples we've covered so far in this book. "What kinds of users are clicking our ads?" and "Do high-income and low-income countries emit similar amounts of carbon dioxide?" are examples of Descriptive Questions. And because Descriptive Questions are questions about objective reality,[1] they have right and wrong answers (at least in principle. It may be hard to evaluate whether a given attempt to calculate the answer is actually right or wrong).

But Descriptive Questions are not the only type of question you will come across in your career.

Proscriptive Questions are questions about how the world *should be*, not how it actually is. "Should higher income and lower income countries be expected to meet the same carbon emission reduction standards?" or "Do high-income countries have a moral obligation to provide tuberculosis drugs to developing countries for free (or at cost)?" are both examples of Proscriptive Questions.

Unlike Descriptive Questions, Proscriptive Questions don't have correct answers. That's because answers to Proscriptive Questions require evaluating the *desirability* of possible outcomes, which can only be done in the context of a moral/ethical system of values. And as there is no "correct" system of values (in the sense that there is no single universally accepted system of morality), there can be no right or wrong answers to Proscriptive Questions, even in principle.



Note: The terms "Proscriptive" and "Descriptive" are commonly used for these concepts in the natural sciences, but different academic silos sometimes use different terms. Social scientists, for example, tend to prefer the terms "Positive" in place of Descriptive and "Normative" instead of Proscriptive. These are only difference in nomenclature, however, not substantive meaning.

[1] If you know enough epistemology to object to me asserting the existence of an "objective reality," then I assume you can also understand the point I'm trying to get across in this chapter and will forgive me this philosophical slight.

The focus of this book is on Descriptive Questions. This is not because Proscriptive Questions are unimportant — indeed, one can easily make the argument that they are *more* important than Descriptive Questions. Moreover, as we will discuss in future chapters, they will arise frequently in your career as a data scientist. No, the reason that Descriptive Questions are the focus of this book is that those are the only questions data science tools can answer, and thus answering Descriptive Questions is the domain in which the data scientist has a clear comparative advantage.

Now, to be clear, none of this is to mean that the answers you generate as a data scientist will not have a *bearing* on how people answer Proscriptive Questions. Data science would be a very dull field indeed if it could not speak to the ethical issues of our day. Data science is powerful precisely because it can inform how we answer Proscriptive Questions by helping us understand the relevant stakes. Data science tools can help decision-makers understand the likely *consequences of different courses of action*, information that can help people make *informed* decisions about what outcomes they feel are most desirable. To illustrate, let's consider a few vignettes.

4.1 Opioid Reductions

Suppose you have been hired by a medical regulatory board concerned about the rise in opioid overdoses. They are debating whether they *should* (there's that magic word!) make it harder for patients to get opioids. Fundamentally, however, they worry that while restrictions on opioids may reduce overdoses and addiction, they may also prevent some patients with very real pain conditions from getting the care they need.

Why are they stuck? Well, there may be two causes:

1) They may be unsure of the relative moral weight to give preventing overdoses versus ensuring appropriate patient access to opioids, and/or

2) they may also be unsure about *how much* opioid regulations that reduce overdoses by a certain amount would limit access for patients in need.

The first of these questions is a pure Proscriptive Question — if you could prevent one overdose death at the expense of preventing 10 patients in pain from getting the opioids they need, would you accept that trade-off?

But the second is actually a Descriptive Question that you — the data scientist — *can* answer! You could study policies that have been implemented in the past and come up with a rigorous estimate of how much opioid regulations that reduce overdoses also reduce access for patients in need. You could also evaluate different kinds of policies to figure out which is most efficient — maybe some policies (like not allowing any opioid prescriptions at all) are good at stopping overdose deaths but also *really* limit appropriate access, while other policies are similarly good at reducing overdoses but have a much smaller effect on limiting access.

4.2 The Example of Carbon Emissions

A profoundly difficult Proscriptive Question in debates over carbon reduction is whether developing countries should be held to the same emission reduction targets as more developed countries. On the one hand, developing countries like China and India are the source of most current growth in carbon emissions, and so policies that do not apply to developing countries are unlikely to prevent many of the worst climate change outcomes. On the other hand, these countries produce radically less carbon *per capita* than Europe or the United States, and the industrial growth creating those emissions has been a major factor in lifting billions of people out of extreme poverty.

Hard choices indeed! How does one weigh the improvements in the quality of life of those in extreme poverty against the possible consequences of even greater climate catastrophes?

While that question, in part, is a Proscriptive Question that no regression can answer, data scientists *can* bring data to bear on this question indirectly by helping everyone understand the potential consequences of different carbon targets for developing countries, as well as the feasibility of different strategies for carbon reduction. A data scientist could, for example:

- Evaluate the effectiveness of different messages politicians in the US and Europe could use to convince their constituents to support greater carbon reduction targets,
- Quantify the magnitude of the effect on global warming caused by different emissions targets for developing countries to help politicians in developing countries weigh the poverty-reducing benefits of carbon-intensive industrialization against the likely direct effect of flooding, droughts, or more severe storms on their own citizens, or
- Estimate the cost-effectiveness of developed countries sharing lower emissions industrial technologies with developing countries to ameliorate the tradeoff between poverty reduction and emissions.

In each of these cases, the data scientist is only answering Descriptive Questions, but in doing so they are helping everyone better understand the consequences of their decisions, and in doing so (hopefully) help the world to make more informed decisions about the trade-offs they are making.

4.3 Recap

Answering Descriptive Questions — questions about how the world is or would be in different scenarios — is the core competency of the data scientist. In the chapters that follow, we will explore in detail three different kinds of Descriptive Questions: Exploratory, Passive-Prediction, and Causal Questions.

While these are the only types of questions that data science tools can answer directly, it is important for you, the data scientist, to also recognize when you encounter Proscriptive Questions — that is, questions about how the world *should* be, or what we *ought* to do. These questions can only be answered with respect to a system of values, and as such, do not have right or wrong answers, and cannot be answered by statistical means. Nevertheless, as a data scientist, you are well-prepared to help others (and yourself!) make more informed choices when they decide how to answer Proscriptive Questions for themselves.

CHAPTER FIVE

EDA: THE MOST PERNICIOUS TERM IN DATA SCIENCE

In our next reading, we will turn our attention to *Exploratory Questions*. First, however, it is important to have a candid discussion about what I feel is one of the most problematic concepts in data science education: *Exploratory Data Analysis* or *EDA*.

The problem with the term Exploratory Data Analysis is that, if you asked most data scientists what it means, they probably couldn't actually give you a straight answer. If you pressed them further, they would probably say something like "when you look at your data before you start fitting your models."

While the idea that data scientists should "get to know their data" before fitting a model is well-meaning (you *absolutely* should!), the ubiquitous but uncritical use of the term has given young data scientists the sense that the undirected poking at data is worthy of a capitalized three world title, complete with a universally recognized acronym.

This is problematic because *any* activity that involves data but lacks a clear motivation is doomed to be unending and unproductive. Data science has emerged precisely because our datasets are far too complex for us to understand directly; indeed, I would argue that the job of a data scientist can be summed up, in part, as a person who identifies **meaningful** patterns in our data and makes them comprehensible.

But therein lies the problem — without a clear motivation for *why* the data scientist is poking at their data, what makes a pattern meaningful is undefined. And without a clear purpose from which a concept of meaningfulness can be derived, there is no end to the ways one can slice and dice the data with no way of knowing when to stop or what is useful.

I would argue that what most people call Exploratory Data Analysis (EDA) can actually be decomposed into three activities.

The first activity people call EDA is what I call "learning the structure of your *dataset*" (emphasis on learning about your *dataset*, not using your data to learn about the world). This consists of answering questions about your dataset like "what constitutes a single observation in this dataset?," "what variables are included in this dataset?," "how many observations are there?," "how are variables coded?," and "what population is represented in this data?" These are questions about *the specific dataset* you are working with, *not* the real world, and answers are likely to be found in the dataset documentation and through basic tools for data introspection.[1]

The second activity that often falls under the label EDA is what I call "validating your dataset." It's a poor data scientist who takes the validity of their data on blind faith, so when faced with a new dataset, one should begin with a few "sanity checks" just to make sure things look reasonable. Does the number of observations seem reasonable given what you know about how the data was collected and who is supposed to be represented in the data? If there are date variables in the data, does their range match what should be in this data? And given the specifics of the data, does the range of variables make sense? For example, if you have data on registered voters 18 and over, you should probably check that the age variable has a minimum value of 18 and a maximum value of something sensible (e.g., not 225).

The third and final activity people call EDA is… everything one does with the data before they fit a statistical or machine learning model. This is the second major reason that I feel the very concept of EDA has had a pernicious influence on data science — it implicitly devalues anything done with data that doesn't entail a complicated model as "lesser" or "just a stop on the way towards the "real" analysis," when nothing could be further from the truth.

[1] In `pandas`, this would be things like `df.columns` to see what variables are in the data, `df.info()` to get a sense of how data is being represented and the number of rows, and simple tools for tabulating unique values like `df["first column"].value_counts()`.

This type of data analysis — looking at summary statistics, calculating distributions of variables, computing tabulations and cross-tabulations of different things to improve one's understanding of the world — is categorically different from "learning the structure of your data," because it is inquiry in the service of better understanding the world, not the structure of your dataset. But it is *not* categorically different analyzing data using statistical models, not just because in many cases generating cross-tabulations or calculating group averages are essentially equivalent to using a statistical method like linear regression, but also because they are both examples of the same enterprise: attempting to answer questions about the world using data in the service of solving problems.

And just as one cannot properly fit or tune a model without a clear sense of the question one is seeking to answer and how that answer is meant to be used, nor can one know what cross-tabulations to compute without having a sense of purpose to make clear what constitutes "meaningfulness."

Note: "But I do EDA all the time without a clear question!" I hear you cry. "Sometimes I just want to see what patterns there are in the data." To you I say: you may not have realized you had questions in mind, but most of your data explorations have been *implicitly* motivated by a sense of questions you thought might relate to your stakeholder's problem.

Perhaps you were looking at a store's retail sales data and decided to see how sales volumes varied by customer age or gender. That may not seem obviously question-motivated, but I put it to you that you had in mind that those are customer demographics to which the store could target advertising or product stocking decisions. And had someone suggested "why don't you look at how sales volumes vary by customer birth month or whether their name starts with a letter in the first half of the alphabet," you would have looked at them funny and said "why on Earth would I do that?"

But the problem with approaching your data with *implicit* motivations is that (a) it's hard to reflect on them or evaluate whether they rest on solid assumptions about the stakeholder problem, and (b) without an explicit goal, there's no way to know when you've reached your destination, making it *really* easy to get lost in the data.

5.1 Recap

Despite its ubiquity, few data scientists could actually tell you what constitutes Exploratory Data Analysis (EDA). Moreover, some of what people might call EDA in practice — answering questions about the world without complex modeling — should not be called EDA, but rather… well, that's just data science.

So in this book, we will acknowledge the important (but distinct!) goal of two purposeful activities often called EDA:

- Learning the structure of your dataset (what constitutes a unit of observation, what variables are in the dataset),
- Validating your dataset (does the data pass the sniff test? Does it exhibit the basic properties you would expect given what it claims to be?)

But I will *not* use the term EDA itself, and when I differentiate between data science enterprises, I will do so by emphasizing differences in the *end goals* of those activities (answering Exploratory Questions, Passive-Prediction Questions, or Causal Questions), not the methods used to achieve those ends.

CHAPTER

SIX

USING EXPLORATORY QUESTIONS

The hardest part of a data science project is often properly articulating the problem we wish to solve. That's because properly specifying a problem requires *understanding* the problem well enough to state it, and often we call issues "problems" precisely because we don't really understand them!

Enter *Exploratory Questions*. Exploratory Questions are questions designed to elicit information about our problem space and aid us in prioritizing our efforts and refining our goals. Exploratory Questions are questions about broader patterns in the world. In their simplest form, they can be answered by simple summary statistics or plots. When more complicated or related to more subtle and contingent patterns, they are likely to be answered through unsupervised machine learning algorithms or the tools of *statistical inference*, such as regressions and generalized linear models. When used for answering Exploratory Questions, the emphasis of regression or generalized linear models is on what the model coefficients can tell us about how different factors may co-vary in the world. This is distinct from how these same tools may be used to answer Passive-Prediction Questions, however, where the emphasis is on the predicted values these models can generate.

Of the three classes of questions we detail in this book, answering Exploratory Questions often (though not always) requires the least technical sophistication, and as a result, Exploratory Questions often get the least respect. But because of their critical role in improving our understanding of our objectives, learning to ask and answer Exploratory Questions will have a huge influence on your effectiveness as a data scientist.

In this reading, we will discuss how to *use* Exploratory Questions to guide your work. Then, in the next reading, we will discuss the challenges inherent to *answering* Exploratory Questions.

6.1 Using Exploratory Questions to Prioritize Efforts

Exploratory Questions are questions about the underlying patterns that characterize our world. Given that, answers to Exploratory Questions should make you feel like you understand the contours of the problem you seek to solve better. More than anything else, then, Exploratory Questions help data scientists prioritize their subsequent efforts and investigations.

6.1.1 Code Optimization

If this feels too abstract, let's use a small example of a problem you've probably already come across (and if not, probably should have!) in the classroom to make it more concrete: learning to write performant (i.e., fast) code.

Data science is full of computationally intensive tasks that, if approached incorrectly, can leave a data scientist staring at their computer for hours, days, or even weeks (if they allow it). As a result, most data scientists will go through a phase in their development when they start constantly worrying about how to make every line of code they write as fast as possible. They bend over backward to write unnatural, unreadable code to ensure that they aren't wasting a single CPU clock cycle.

The problem with this is that humans have *incredibly* bad intuition about what tasks take a computer a long time. It turns out that even in programs that take huge amounts of time to run, it is often the case that *most* of the program's runtime is taken up by a single function or loop. As a result, programmers who fixate on ensuring every line of code they write is optimized for speed end up not only wasting their *own* time, but also writing code that is less natural, harder to maintain, and more likely to contain errors for effectively no benefit.

Indeed, no less a figure than Donald Knuth, one of the greatest programmers in history and author of the famous *The Art of Computer Programming*, famously said of this trying to optimize each line of code at the time it is being written ("premature optimization"):

> The real problem is that programmers have spent far too much time worrying about efficiency in the wrong places and at the wrong times; **premature optimization is the root of all evil (or at least most of it) in programming.** [emphasis added]

So what is a programmer interested in performance to do? First, write code in as natural a way as possible. Then, *if* the result is code that is slower than they would like, ask the exploratory question: "What lines of code are contributing most to this program taking so long to run?" And only then, once the programmer has identified the problematic parts of their code, optimize it for performance.

How is this question answered accomplished? Programmers use tools called *profilers* that dip into a running program every few milliseconds to see what functions are currently running. Then after the program has finished running, it reports how often each part of the program code was found to be running, giving the user a sense of the overall distribution of time spent running different parts of the code.

6.1.2 Picking the Right Target

If the preceding example feels too niche — you want to be a data scientist, after all, not a software engineer! — let's consider a different example. Suppose you've been hired by a new non-profit interested in helping reduce energy use in buildings in the United States. They know that fixed structures (factors, stores, houses, etc.) are responsible for a huge share of US energy consumption, and are interested in figuring out how to drive down that energy use by helping building owners improve the energy efficiency of their buildings (by providing information on things like government subsidies for efficiency improvements and the potential value of energy efficient windows, better heating and cooling, etc.).

You *could* start out by trying to build a fancy supervised machine learning model that tried to predict the energy use of every building in the US based on infrared satellite data and weather information. Indeed, that may even be what you were asked to do! (See our discussion of how *stakeholders will often have somewhat wild ideas of what is feasible and what would help most.*).

But given this is a new non-profit, it sounds like their real need is probably to figure out how to target their efforts to be most effective. So maybe we should step back and start by trying to answer a few Exploratory Questions that would help the organization decide where to focus its attention:

- What *type* of buildings (industrial, residential, commercial) consume the most power in the US?
  - The answer to this question can help you prioritize the *types* of buildings on which to focus your efforts. For example, if industrial or commercial buildings only represent a few percent of all energy consumed by buildings, you don't need to worry about addressing their needs!
- In what *region* of the US are buildings consuming the most power?
  - If most energy is being consumed in a specific area, perhaps the non-profit should start by focusing its efforts regionally.
- Is there a *region* of the US where buildings are generating the most CO2?
  - Not all power is created equal when it comes to climate change! Maybe buildings in California consume a lot of energy, but because they have cleaner power plants, those buildings are indirectly generating less CO2 than those in states in the US South?
- Does the *average energy use per building* vary by region or building type?

  - If the non-profit plans to approach building owners, it may be easier to have an impact working with a few owners of large buildings than lots of residential homeowners. But of course, that also depends on the answer to our previous question about what types of buildings are using the most power/generating the most CO2!

- In what season is most building energy consumed? Is more energy consumed by heating or AC needs, or do the two use similar amounts of power?
  - Again, this may impact both the regions the non-profit may wish to focus on, and also the types of efficiency retrofits they may wish to prioritize.
- Where is power most expensive?
  - Building owners are most likely to be interested in efficiency retrofits when power is expensive.

While answering these questions is likely to require some significant detective work, and may require some thoughtful data wrangling, none require deeply sophisticated statistical machinery. But that doesn't mean answering these questions wouldn't provide **huge** value to the stakeholder.

6.1.3 Collecting, Merging, and Creating New Data

Once you start articulating these questions, you can start to see that there is some important data science to do; that's because the answers to these questions may not all point in the same direction, and so the non-profit likely needs someone to be able to evaluate how these different factors co-vary, and the relative magnitude of different trade-offs (e.g., if fewer buildings use a lot of power in the US South than California, but the US South is using coal power instead of renewable energy, where should the non-profit focus?).

And that, fundamentally, is what Exploratory Questions are about: understanding the patterns and distribution of features you care about in the world, and using that information to better understand the problem you want to solve.

This also demonstrates one of the key ways that one answers Exploratory Questions: by collecting and merging datasets that had not previously been pulled together. Sometimes this data collection requires no more than finding people who already have the data you need, getting it, and finding a way to merge different data sources (e.g., data on power plant CO2 emissions and data on building energy use), while in other situations this will entail building new datasets yourself by doing things like using Natural Language Processing to make collections of documents (contracts, patient files, public records) analyzable systematically.

CHAPTER

SEVEN

ANSWERING EXPLORATORY QUESTIONS

In the last reading, we discussed how Exploratory Questions are used by data scientists to help stakeholders better understand their problems and to prioritize subsequent investigations. In this reading, we turn to the questions of what *answering* Exploratory Questions effectively entails.

7.1 The Three-Part Goal

Whether one uses simple summary statistics (means and medians), plots, or more sophisticated algorithms from the domains of statistical inference and unsupervised machine learning, answering Exploratory Questions always boils down to the same challenge:

Creating (1) understandable summarizations (2) of meaningful patterns in the data, (3) and ensuring they are faithful representations of the data.

What is meant by these three components exactly? Let's take each in turn.

7.1.1 Understandable Summarizations

> Creating **(1) understandable summarizations** (2) of meaningful patterns in the data, (3) and ensuring they are faithful representations of the data.

Answering Exploratory Questions effectively is all about taking large datasets that, in their raw form, are effectively incomprehensible to humans and summarizing the patterns in that data in a way that can be understood. These summaries of patterns in the data may take many forms — summary statistics, regression coefficients, plots, etc. — but all, when done well, have a similar goal: to represent the salient aspects of data in a way that is accessible to the human mind.

Professionals from different disciplines often use different terminology to describe this process of summarization. Some like to refer to it as "separating the signal (the thing that's important) from the noise (all the other variation that doesn't matter)," others talk about "dimensionality reduction" (basically linear algebra speak for summarization), while still others may talk about "modeling the underlying data generating process that gave rise to the observed data." Regardless of the terminology one uses, however, these all boil down to the same thing: filtering and discarding the variation the data scientist deems to be irrelevant to make it easier to see and understand the variation deemed important.

The importance of researcher discretion in deciding what variation to discard as noise and what variation to foreground as "important" is one of the defining challenges of answering Exploratory Questions. Other types of questions — like Passive Prediction Questions — often involve using more mathematically sophisticated modeling tools, and consequently are viewed as more challenging. In my experience, however, learning to understand the stakeholder's problem context *and* the variation in a data set well enough to exercise this discretion effectively is actually one of the things young data scientists struggle with most. It requires both good domain knowledge to understand what is *meaningful* (as we will discuss below), and also for the data scientist to spend a lot of time exploring the data thoughtfully and from different perspectives. This is a hard skill to learn,[1] but with intentionality, patience, and practice, it is a talent that once learned will helps set you apart from the average pytorch-jockey.

Summarizations created to answer Exploratory Questions can differ radically in their ambition. At one end of the spectrum are simple summary statistics, like means, median, and standard deviations. These seek to provide a simple characterization of a single feature of a single variable. Slightly more ambitious are various forms of plots — like histograms (which are substantially richer than the aforementioned summary statistics) or scatter plots and heatmaps (which provide substantial granularity and communicate information about the relationship between different variables). The most ambitious efforts make use of multivariate regressions and unsupervised machine learning algorithms to model what they call the *Data Generating Process* (DGP) — the actual physical or social processes that gave rise to the data you observe, and which (hopefully) can be represented in a relatively parsimonious manner, much as the relatively simple laws of physics give rise to the orbits of the planets and the complexity of life.

To illustrate what I mean by trying to deduce something about the data-generating process, suppose you are a medical researcher interested in a poorly understood disease like Chronic Fatigue Syndrome (CFS). It is generally agreed that CFS is more of a label for a constellation of symptoms than an understood physical ailment, and you have a hypothesis that the symptoms of CFS aren't actually caused by a single biological dysfunction, but rather that multiple distinct biological dysfunctions give rise to similar symptoms that we have mistakenly grouped under this same umbrella term. In other words, you think that the data-generating process that gives rise to patients diagnosed with Chronic Fatigue Syndrome consists of two distinct diseases.

You're fortunate enough to have detailed patient data on people diagnosed with the condition, but it's impossible for you to just look at these gigabytes of thousands of patient records and "see" any meaningful patterns. You need a way to filter out irrelevant data to identify the "signal" of these two conditions. To aid you in this question, you decide to ask "If you were to group patients into two groups so that the patients in each cluster looked as similar as possible, but patients in different clusters looked as *dissimilar* as possible, how would you group these patients?"

This, you may recognize, is precisely the question clustering algorithms (a kind of unsupervised machine learning algorithm) are designed to answer! So you apply your clustering algorithm to the patient data and get back a partition of the patients into two distinct groups. This, in and of itself, doesn't constitute a particularly *understandable* summarization of your data, but it provides a starting point for trying to investigate *diagnostically and biologically relevant* differences that exist between these populations. If one cluster included more patients reporting fatigue when doing any exercise, while another cluster reported they felt better when they exercised, but felt a high level of baseline fatigue that didn't respond to sleep, that might suggest that the *data-generating process* for these patients was actually driven by two different biological processes. *And* it gives you a great starting point to prioritize your subsequent investigations into what might explain these differences!

7.1.2 Meaningful Patterns

> Creating (1) understandable summarizations **(2) of meaningful patterns in the data,** (3) and ensuring they are faithful representations of the data.

Inherent in creating any summarization is exercising discretion over what variation is relevant (signal) and what variation is not (noise). But just as one person's trash may be another person's treasure, so too may one person's signal be another person's noise, depending on their goals! Crucially, then, the data scientists' guiding star when deciding what is important is whether certain variation in the data is *meaningful to the stakeholder's problem*.

As data scientists, we are blessed with an abundance of tools for characterizing different facets of our data. These range from the simple — means, standard deviations, and scatter plots — to the profoundly sophisticated, like clustering algorithms, principal component analyses, and semi-parametric generalized additive models.

Regardless of the specific methods being employed, however, none of these tools can really tell us whether the patterns they identify are meaningful, and that's because what constitutes a meaningful pattern depends on the problem the stakeholder is seeking to address and the context in which they're operating.

[1] Although I am far from convinced that the discipline has tried particularly hard to teach it (*see my screed against "EDA"*).

To illustrate the importance of context, suppose you are hired by a hospital to learn what can be done to reduce antibiotic-resistant infections. So you grab data on the various bacteria that had been infecting patients and write a web scraper and Natural Language Processing pipeline to systematically summarize all available research on the cause of these antibiotic-resistant bacteria. Your work is *amazing*, seriously top of the line, and after two months you conclude that in most cases, the cause of antibiotic resistance in the bacteria infecting patients is… the use of antibiotics in livestock.

Now, that analysis may not be *wrong* — you have properly characterized a pattern in the data — but it isn't a pattern that's meaningful to your stakeholder, who has no ability to regulate the livestock industry. That pattern might be meaningful to someone else — like a government regulator — but in this context, with this stakeholder, it just isn't helpful. The features of the data that are important, in other words, depend on what we may be able to do in response to what we learn. And there's no summary statistic, information criterion, or divergence metric that can evaluate whether a pattern of this type is *meaningful*.

7.1.3 Faithful Representations

> Creating (1) understandable summarizations (2) of meaningful patterns in the data, **(3) and ensuring they are faithful representations of the data.**

What do you means, medians, standard deviations, linear regressions, logistic regressions, generalized additive models (GAMs), singular value decomposition (SVD), principal component analyses (PCAs), clustering algorithms, and anomaly detection algorithms all have in common?

Answer: unless your dataset is extremely degenerate, you can point *any* of these tools at your data and they will return a relatively easy-to-understand characterization of the structure of your data.

At first, that may seem extremely exciting. But if you think about it a little longer you will realize the problem: all of these are designed to give you a relatively understandable summary of radically different properties of your data, and even though they will all provide you with a result, these results can't all possibly be faithful representations of the dominant patterns in your data.

To illustrate the point, suppose I told you that in one university math course, the average grade was a B-. You might infer that students were doing pretty well! But now suppose I told you that in a different university math course, 20% of the students had gotten a 0 on the midterm and on the final—you would probably infer something was going seriously wrong in that class. And yet those two statistics could both be true of the same class—the only difference is what patterns in the data *I*, the data scientist, have decided are meaningful to communicate to you, the reader.

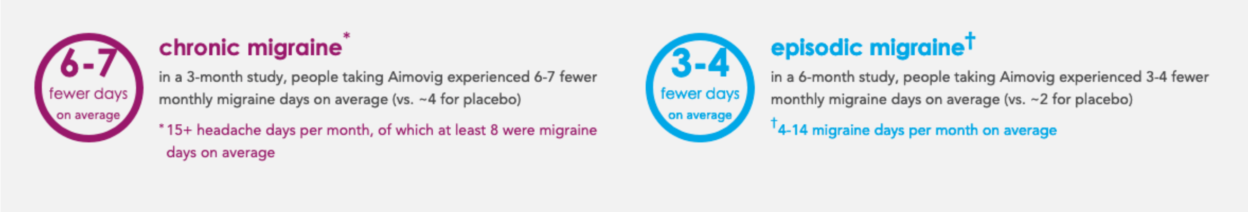
The example of the math class in which the average grade was a B- and 20% of the students were failing also illustrates one of the great dangers of tools for data summarization: they are so eager to please, they will *always* provide you with an answer, whether that answer is meaningful or not. I think most readers would agree that learning that the average grade in the class was a B- actually misleads more than it informs (since for the class to have an average grade of 80% and a 20% fail rate, the grade distribution would need to be something like 20% 0's and 80% 100's). Indeed, it's worth emphasizing that while hearing "the average grade is a B-" makes the reader think that most kids are doing ok-ish, the reality is that *no one* in the class is doing ok-ish! They're either doing horribly or terrifically!

Less that feel like a contrived example, consider the case of Aimovig, a drug authorized by the FDA in 2018 for treating chronic migraines that was heralded as a "game changer."

To get Aimovig authorized, the pharmaceutical companies developing (Amgen and Novartis) had to run a clinical trial in which a random sample of people with chronic migraines was given Aimovig (the treatment group) and a random sample was a placebo (the control group). Patients in the clinical trial self-reported how their migraine frequency changed when in the trial, and the effectiveness of Aimovig was then evaluated by comparing the decrease in self-reported migraines for those taking Aimovig (on average, a decrease of 6-7 migraines a month) to the decrease in self-reported migraines for those taking a placebo (on average, a decrease of 4 migraines a month).[2] This difference of 2-3 migraines a month — called the "Average Treatment Effect" of the trial — was found to be positive and statistically significant, and so the drug

[2] A placebo is a "fake" treatment given to patients in clinical trials. Despite not being biologically active — placebos are often simple saline or sugar pills — most patients on placebos see their condition improve when dealing with subjective conditions, like pain.

was authorized. Indeed, if you see an ad for Aimovig, you'll probably see the average effect of the drug reported in the same way:



6-7 fewer days on average

chronic migraine*

in a 3-month study, people taking Aimovig experienced 6-7 fewer monthly migraine days on average (vs. ~4 for placebo)

*15+ headache days per month, of which at least 8 were migraine days on average

3-4 fewer days on average

episodic migraine†

in a 6-month study, people taking Aimovig experienced 3-4 fewer monthly migraine days on average (vs. ~2 for placebo)

†4-14 migraine days per month on average

That's great! Chronic migraines can be a crippling disability, so any improvement in treatment is exciting. But you would be excused for asking why people were getting *so* excited about what seems like a relatively small reduction in migraines.

The answer, as it turns out, is that almost nobody experiences this "average effect." Instead, *most* people who take Aimovig see little to no benefit, but *some* (depending on your criteria, something like 40%) see their migraine frequency fall by 50% or more. Amgen and Novartis don't yet know how to identify who will benefit and who will not before they try the drug, and we don't allow drug companies to "move the goalposts" after a clinical trial has already started by changing the way they plan to measure the effectiveness of a drug (for fear they will hunt through the data till they find a spurious correlation that makes it look like the drug works when it really doesn't), so this average effect remains the only statistic that Amgen and Novartis are allowed to report in their advertising.

But if you're a *doctor* or a *patient*, it seems clear that this simple average effect — a reduction of 2-3 migraines a month — really does not provide a *faithful* summary of the underlying variation.

7.1.4 But... I Thought Unsupervised Machine Learning Always Found The "Best"

"Fine," I hear you say, "that makes sense for simple summary statistics. Those are computed by simple formulas. But what about unsupervised machine learning algorithms or generalized additive models? Those use numerical optimization to find the *best* answer!"

Well… yes and no. As you may recall, in the first chapter of the book I posited that all data science algorithms are just fancy tools for answering questions, and even the most sophisticated unsupervised machine learning algorithms are no exception. While it is true that the machinery that underlies these algorithms is much more sophisticated than the formula we use for calculating a variable's average, it is important to not attribute too much intelligence to these tools.

Underlying any unsupervised machine learning algorithm is a simple formula that takes as input whatever parameters the algorithm gets to choose (be those factor loads in a PCA model, or the assignment of observations to clusters in a clustering algorithm) and returns as output a single number. Often this number is called "loss," and the function is called a "loss function," but occasionally different terminology will be used.

One way to think of the job of an unsupervised machine learning algorithm is to pick the parameter values that minimize this loss function. A clustering algorithm for example, may try and assign observations to clusters to maximize the similarity of observations within a cluster (say, by minimizing the sum of squared differences between the values of certain variables for all observations within a cluster) while also maximizing the differences between observations in different clusters (say, by maximizing the sum of squared differences between the values of certain variables for all observations *not* in the same cluster).

But another way to say that is that the job of an unsupervised machine learning algorithm (or any algorithm, really) is to find the parameter values (coefficients in a regression, observation assignments for a clustering algorithm) that answer the question "If my goal is to minimize [whatever the loss function your specific algorithm seeks to minimize], how should I do it?" But while they are likely to find the best way to accomplish that goal given the parameters they control, they will do so *whether or not the "best" solution is actually a "good" solution!* Point a clustering algorithm at any data and ask it to split the data into 3 clusters, and it will pick the best way to split the data into three clusters, even if the three clusters are *almost* indistinguishable. In other words, clustering algorithms assign observations to clusters… even when there's no real clustering of the data! Dimensionality reduction algorithms will always tell you a way to drop dimensions, and anomaly detection algorithms will always find (relative) outliers.

Moreover, just because your clustering algorithm finds what it thinks is the best solution doesn't mean there isn't a *substantively* very different solution that was *just* a little less good it hasn't told you about.

It's up to you, the data scientist, to evaluate whether the answers these algorithms provide to relatively myopic questions give a meaningful picture of the data.

7.1.5 Myopic Tools

This last point is illustrative of a more general point: data science tools are incredibly powerful at finding answers to questions of the form "If my goal is to minimize X, how should I do it?" type questions — answers you may have never figured out in millions of years! — but their power lies in figuring out the best way to accomplish an articulated goal, *not* in figuring out what goal to pursue.

This is true at both the macro level (doesn't make sense to look for clusters in my data?) and also at the micro level (when assigning observations to clusters, how do I measure success?). Hidden inside nearly all algorithms you use are a handful of baked-in choices you may not even realize are being made for you. Take clustering, for example. In general, when clustering observations, one has two objectives: maximize the similarity of observations within each cluster and maximize the *dis*similarity of observations in different clusters. But what you might not have thought about very much is that there's an inherent tension between these two objectives — after all, the best way to maximize the similarity of observations within each cluster is to only assign observations to the same cluster if they are identical (a choice that creates lots and lots of very small clusters). And the best way to maximize *dis*similarity between clusters is to only put *really really* different observations in different clusters (resulting in a few really big clusters). So how is your clustering algorithm balancing these two considerations? Is the algorithm's choice of how to balance them in any way a reflection of the balance that makes the most sense in the context of your stakeholder's problem? (I'll give you a hint — the algorithm sure can't answer that question, so you'd better be able to!)

Discretion: it's everywhere, and you're exercising it, whether you realize it or not.

CHAPTER

EIGHT

INTERNAL VERSUS EXTERNAL VALIDITY

It is at this point I have to come clean about having employed a… small indirection. At the top of this reading, I introduced the idea that answering Exploratory Questions boiled down to creating (1) understandable summarizations (2) of meaningful patterns, and (3) ensuring those summaries faithful represent the data. But that three-part objective is actually only one-half of answering an Exploratory Question. More specifically, those are the three components of ensuring high *internal validity* when answering an Exploratory Question. But to generate a truly useful answer to an Exploratory Question, your analysis must also have high *external validity*.

Essentially, *internal validity* is a measure of how well you have analyzed *the data you have*, while *external validity* is how well you expect the answer you generated from that data to generalize to your stakeholder's context. Internal and external validity arise when answering *any* data science question, and so these two concepts are ones that we will return to time and again in this book.

8.1 Interval v. External Validity: An Example

To illustrate what is meant by these terms, suppose you've been hired by a specialty grocery chain interested in opening new stores. They have a good sense of their customer base ()

To illustrate the difference, suppose a new video streaming service sent out an e-mail offering new users a deal on subscriptions, and then measured the difference in sign-up rates between the users who got the deal and users who just got a generic e-mail with information about the service.

The **internal validity** of the study is the degree to which the study accurately measured the causal effect of the offer on signup rates. Internal validity hinges on things we've talked about a lot in class, like whether the people who received the deal had the same potential outcomes as the people who got the generic email.

The **external validity** of the study, by contrast, is about whether we think the estimated effect is the same effect we would see if we tried to send out a similar email to recruit customers to an *established* streaming service (instead of a new one), or if we tried to use a similar offer to recruit people to a new *music* streaming service.

All studies are subject to both types of concerns, and as we'll discuss below, there are often trade-offs between internal and external validity, especially in causal research.

8.2 External Validity

External validity is fundamentally about the *generalizability* of a study: whether the causal estimate found in a study is likely to also be a good guess for the causal effect in a different context.

External validity is one of the most important things to think about as a *consumer* of other people's research, because when you read other people's research, you're usually doing so because you're looking for information you can use to address a specific problem you face. In these situations, it's critical that you always ask yourself: are the results from this study likely to also be valid in the context of my problem?

Of course, when asking about the external validity of a study, we have to specify the setting to which we want to generalize its results. A study that looks at how Duke undergraduates' consumer behavior changes when faced with different types of ads on google may have good external validity in terms of its generalizability to other elite Univerities like Emory, Vanderbilt, or UNC. But it might not generalize to the US population as a whole.

This means that external validity is different from internal validity in an important way: when faced with the same facts about a study, everyone should *generally* agree on the internal validity of a study, but the external validity of a study really depends on how you want to use the results.

8.2.1 External Validity Considerations

There are many reasons that the results of a study may not generalize to a new context. Here are a handful of the most common issues to bear in mind:

The study population may be different from the population in the new context.

Almost by definition, the entities in the new context will be different from the entities in the original study (even if we're working with the same people, we're looking at them at a different time). But the key question for external validity is whether the entities in the new context are different *in a way that would impact their response to a given treatment*.

It's not hard to think of reasons that different populations may respond differently to a given treatment. For example, suppose a company finds ads for luxury cars increase sales among rich people in New York. It's hard to imagine that the same ad run in a poor neighborhood in Detroit would have the same effect.

As you think about population differences, make sure you consider not only standard demographic attributes (age, gender, wealth, education), but also cultural or social differences. Many issues businesses deal with – especially advertising and brand image – may be culturally specific, and so may not generalize to all communities.

This may all seem obvious as you read it, but using unrepresentative samples in research and medicine, then making recommendations for the general public is a huge problem in the real world.

White men are massively over-represented in medical trials, for example. Unsurprisingly, this means that when the results of those trials are generalized to the population as a whole, we suddenly discover (SURPRISE) that the predicted results didn't always hold for women or people of color! (e.g. drug doses set for men are often too high for women; some heart drugs work great for White men, but often interact poorly with a gene common in Asians and Pacific Islanders; and Multiple sclerosis turns out to be drive by a different mutation in Black patients than European descendants).

And for the longest time, psychology research was based almost entirely on studies conducted using student volunteers. But of course, students at elite universities are not a representative population – they're disproportionately Western, Educated, from Industrialized, Rich, and Democratic countries (they're WEIRD). And as a result, our academic model of human behavior is really just a model of a bunch of WEIRD kids.

Unrepresentative training data is also one of the reasons that so many machine learning algorithms are just plain racist (this isn't causal inference, but it's the same idea) – if you train a facial recognition algorithm using predominantly white faces, turns out that they will either not see Black faces, or worse, mis-identify people of color (which is a really bad thing when those algorithms are being used by the police).

So while internal validity issues may seem more sophisticated and thus interesting, don't overlook the importance of these kinds of external validity issues!

The treatment might differ between study and new context

A study may declare that it has measured the effect of billboard ads on sales, or an infinite scroll on engagement. But it's always important to remember that while we may interprete studies in these general terms, the reality is that that billboard study probably measured the effect *of a specific set of billboard ads* on sales, and the infinite scroll study looked at the effect of infinite scroll *in a specific app*.

So always be careful to think about what *exactly* the treatment in a study was, and whether its likely to generalize to the case you study about.

There may be scaling effects

Often times when we're thinking about external validity, we're not just thinking re-using a treatment or intervention; we're thinking about scaling them up.

But an intervention that works on a few people / is only in place for a short period may not be a perfect model for what happens when that same intervention is applied at scale or permanently. For example, the returns to showing people a TV ad about your company for the first time is probably not the same as the returns to airing that ad the 1,000,000th time. Or sales from selling a special product at one store for a limited time may not be a good indicator of the sales you would see if your "special product" were available everywhere all the time.

People may also respond differently to an intervention when it gets big or becomes permanent. To illustrate, I'd like to tell a story about a famous experiment in India (paper).

Rural health clinics in India have a huge problem: nurse absenteeism. To try and address the problem, in the late 2000s an NGO (along with some MIT economists) decided to see if they could fix the problem. The NGO started keeping track of when nurses clocked in and out, and then shared the information with the government, who then applied fines or punishments to nurses who weren't showing up for work.

Initially, the intervention was successful, leading to very large increases in attendance (doubling it in fact!) after a few months. But as nurses came to realize this wasn't just a little study but actually something that was going to be around for a while, they mobilized politically, and soon administrators were allowing nurses to claim an increasing number of "exempt days", avoiding punishment. And so sure enough, nurses stopped coming to work, and absenteeism had returned to pre-intervention levels 16 months after the program began.

This is an example of what economists call a "general equilibrium" effect – when we introduce a treatment to the world, the world responds. But often these responses don't happen in small trials the same way they do when policies go big, creating serious generalizability problems.

Relatedly: if you are a public policy person or an economic development person, I cannot recommend this paper by Angus Deaton and Nancy Cartwright enough for discussing the limitations of RCTs for learning about the effects of policy or nature of social processes. It's a long, very thoughtful paper, but it's really, *really* good.

CHAPTER

NINE

PASSIVE-PREDICTION QUESTIONS

When most people hear the term "machine learning," what they think of is the ability of computers to answer Passive-Prediction Questions: "which patients are likely to experience complications from surgery if we don't do anything?", "which people applying for life insurance are healthy enough we should issue them a policy?", or "which job applicants would make good employees (and thus, which job applicants should we interview)?" And indeed, the ability of data scientists to answer Passive-Prediction Questions is one of our most useful skills.

However, answering this type of question is also one of the easiest ways to get in trouble as a data scientist. Why? Just as you can always calculate a summary statistic or get a result from an unsupervised machine learning model when trying to answer an Exploratory Question, you can also always get predicted values from a statistical model. But with Passive-Prediction Questions—*unlike* with Exploratory Questions—you can't fully check the validity of your answer to a Passive-Prediction Question with data you currently have. That's because, *by definition*, the reason you are trying to answer a Passive-Prediction Question is that you want to predict something that you don't currently know!

9.1 Flavors of Passive-Prediction Questions

There are two flavors of Passive-Prediction Questions:

- predicting something that has yet to occur ("which patients going in for surgery are likely, in the future, to experience complications?"), and
- predicting something that *could* occur but actually won't ("if a radiologist had looked at this mammogram, would they conclude the patient had cancer?").

The first category of passive prediction—predicting something that has yet to occur—is the most intuitive, and is the type of passive prediction that accords best with the normal meaning of the term "predict." But the second favor of passive prediction—in which we try and predict what someone *would* do—is also very important, as it underlies efforts at automation. Spam detection, image classification, autocomplete, and self-driving cars are all examples of situations where we train a model by showing it examples of how a person *would* do something, so the model can predict what a person would do when faced with new data and emulate that behavior itself.

And just as there are two flavors of passive-prediction, so too are there two corresponding use cases for answering Passive-Prediction Questions:

- Identifying individual entities for follow-up, and
- Automating data classification to make hard-to-work-with data (images, medical scans, text) simpler

9.2 Differentiating Between Exploratory and Passive-Prediction Questions

If you have not felt a little confused about the distinction between Exploratory and Passive-Prediction Questions previously, there's a good chance you find yourself struggling with that issue here, and for understandable reasons.

In many cases, one can easily imagine how the same analysis might constitute an answer to *either* an Exploratory or Passive-Prediction Question. For example, predicting which patients are likely to experience complications from surgery using a logistic regression could constitute the answer a Passive-Prediction Question, but it could also answer Exploratory Questions like "what hospitals have the highest surgery complication rates?" or "what type of surgeries have the highest complication rates?"

The confusion lies in the fact that the distinction between these types of questions isn't related to the statistical machinery you might use to answer the question, but rather what we are trying to accomplish, and thus how we might evaluate the success of a given statistical or machine learning model.

With Passive-Prediction Questions, our interest is in the values that get spit out of a model for each entity in the data. When answering a Passive-Prediction Question, the *only* thing we care about is the quality of those predictions, and so we evaluate the success of a model that aims to answer a Passive-Prediction Question by the quality of those predictions (using metrics like AIC, AUC, R-Squared, Accuracy, Precision, Recall, etc.). Thus, when using a logistic regression to answer a Passive-Prediction Question, we don't actually care about what factors are being used to make our predictions, just that they improve the predictions. Our interest is only the quality of our predicted values, and a good model is one that explains a substantial portion of the variation in our outcome.

With Exploratory Questions, our interest is in improving our understanding of the problem space, not in making precise predictions for each entity in our data. Thus, in the example of logistic regression, our interest is in the factors on the "right-hand side" of our logistic regression and how they help us understand what shapes outcomes, not the exact accuracy of our predictions. A good model, in other words, doesn't actually have to explain a large share of variation at the level of individual entities, but it does have to help us understand our problem space.

For example, a model that looked at the relationship between individuals' salaries and their age, education, and where they live might tell us a *lot* about the importance of a college degree to earnings (which we could see by the large and statistically significant coefficient on having a college degree), even if it only explains a small amount of overall variation in salaries (e.g., the R-Squared might only be 0.2).

This distinction also has important implications when working with more opaque supervised machine learning techniques, like deep learning, random forests, or SVMs. These techniques are often referred to as "black boxes" because exactly how different impute factors relate to the predictions that the model makes is impossible to understand (in other words, it's like the input data is going into a dark box we can't see into, and then predictions are magically popping out the other side). These models can be very useful for answering Passive-Prediction Questions, as they can accommodate very unusual, non-linear relationships between input factors and predicted values, but because these relationships are opaque to us, the data scientist, they don't really help us understand the problem space.

9.3 When Are Our Predictions Valid?

Because passive-prediction is fundamentally about making predictions about things that are not-yet-seen, making predictions is one of the more precarious things a data scientist can do. But that doesn't mean that we are helpless when it comes to determining how confident we should be in our predictions, and when and where we think our predictions will be reliable. In particular, as data scientists, we have a great many tools for evaluating how well our model fits the data we *already have* (a concept known as *internal validity*), and ways of thinking critically about the contexts in which using a given model to make predictions are appropriate (a concept known as *external validity*).

9.3.1 Internal Validity

Of all the places where data science is fragmented, none is more evident than in how data scientists evaluate how effectively we think a model is representing our data.

The first data science perspective on evaluating the internal validity of a model comes from the field of statistics. Statisticians have approached evaluating model fit with, unsurprisingly, methods based on the idea of random sampling and the properties of statistical distributions. They make assumptions about the distributions underlying data and use those to derive theoretically-motivated metrics. That's the origin of statistics like Akaike Information Criterion (AIC), Bayesian Information Criterion (BIC), as well as the emphasis on the validity of the standard errors assigned to factors on the right-hand side of the regression.

When computer scientists were first developing their own machine learning techniques… I'm editorializing a little here, but I think it's safe to say that initially they either didn't *know about* a lot about these metrics, or they thought that they could do a better job investing their own. So they developed the "split-train-test" approach to model evaluation: they split their data into two parts, train their model on part of the data, then test how well the model is able to predict the (known) outcomes in the test dataset.

Of course, over time these two fields have largely converged in adopting one another's methods, and some—like cross-validation—live comfortably in the middle. But if you're ever wondering why, when you get to a machine learning class, it seems like everything you learned in stats has been abandoned (or end up in a stats class and have the opposite experience), it's largely an artifact of parallel development of methods of model evaluation in computer science and statistics departments.

9.3.2 External Validity

Where *internal validity* is a measure of how well a model captures the meaningful variation in the data we already have, *external validity* is a measure of how well we think that our model is likely to perform when faced with new data.

The external validity of a model, it is important to emphasize, is specific to the context in which a model is being used. A model will generally have very *high* external validity when used to answer Passive-Prediction Questions in a setting that is very similar to the setting from which the data used to train the model was collected, but *low* external validity when applied in a very different setting.

There are a range of factors that can determine external validity, such as whether a model is being used to answer Passive-Prediction Questions about:

- the *same population* from which the training data was drawn. The patterns in data from one country will often differ from patterns in data from another country, for example.
- the *same time period* from which the training data was drawn. Consumer behavior may vary across seasons, and many patterns in data change over longer timespans.
- the *same parameter ranges* as those in the training data. Statistical and machine learning models are designed to fit the data they can see as well as possible. However, while nearly all models will generate predictions about outcomes when given inputs that weren't in the data used to fit a model because they weren't trained with access to data of this type, their guesses are unlikely to be particularly meaningful.

To illustrate, consider the two models in the figure below (source)—one a linear fit, and one a higher-order polynomial. Both model the data similarly *in the range for which data is available* but make very different predictions at values of `x` below 0 or above 2.

FIX COMMENTED SECTIONS

CHAPTER

TEN

CAUSAL QUESTIONS: THE THEORY

In our previous readings, we learned how answering different types of questions can help us better understand the world around us. By answering Exploratory Questions, we can better understand the contours of our problem — where our problem is most acute, whether there are groups who have figured out how to get around the problem on their own, etc. This, in turn, can help us prioritize our subsequent efforts. And by answering Passive Predictive Questions, we can help identify individual entities — patients, customers, products, etc. — to whom we may wish to pay extra attention or recommend certain products. And when answering Passive Predictive Questions, we can also automate tasks by predicting how a person or more complicated process *would* have classified an entity.

In both cases, however, answering these questions only helps us better understand the world *around* us. But to the extent to which, as data scientists, we want to intervene to directly address problems, we are rarely interested in just knowing about the world around us — we want to *act* on the world, and wouldn't be great if data science could provide us with a set of tools designed to help us predict the *consequences* of our actions?

Enter *Causal Questions*. Causal Questions ask what *effect* we can expect from *acting* — that is, actively *manipulating* or *intervening* — in the world around us in some way. For example, if we pay to show an ad to a specific customer, what will the *effect* of that choice be the likelihood they buy something on our website? Or if we chose to give a new drug to a patient, what will the *effect* of that choice be on their disease?

Because of their potential to help us understand the future consequences of our actions, it should come as no surprise that the ability to answer Causal Questions is of *profound* interest to everyone from companies to doctors and policymakers. At the same time, however, it may also come as no surprise that answering Causal Questions is an inherently challenging undertaking.

In this reading, we will discuss what it means to measure the effect of an action X (administering a new drug to a patient, showing an ad to a user) on an outcome Y (patient survival, or customer spending). This section will, at times, feel a little abstract and woo-woo, but please hang in there. Answering Causal Questions is as much about critical thinking as it is statistics, and the concepts introduced here will prove crucial to your ability to be effective in this space.

Then in our next reading, we will term from the more abstract to the concrete and discuss where Causal Questions arise in practice, and the workflow one goes about answering them.

10.1 What Does It Mean for X to Cause Y?

To understand what it means to answer a Causal Question, and why answering Causal Questions is intrinsically hard, we must start by taking a step back to answer the question: "what do we mean when we say some action X *causes* a change in some outcome Y?"

Seriously, what do we mean when we say "X causes Y?" Try and come up with a definition!

While this question may *seem* simple, it turns out that this question has been the subject of serious academic debate for hundreds of years by philosophers no less famous than David Hume. Indeed, even today there is still debate over how best to answer this question.

In this course, we will make use of the *Counterfactual Model of Causality* (sometimes called the Neyman-Rubin causal model). In plain English, it posits that for “doing X to cause Y”, it must be the case that if we do X, then Y will occur, and if we did not do X, then Y would not occur. This is by far the most used definition of causality today, and yet remarkably, it only emerged in the 20th Century and was only really fleshed out in the 1970s. Yeah… that recently.

Counterfactual Model of Causality

For it to be the case that doing X causes Y”, it must be the case that if we do X, then Y will occur, and if we did not do X, then Y would not occur.

10.2 Measuring the Effect of X on Y

At first blush, this definition may seem simple. But its simplicity belies a profoundly difficult practical problem. See, this definition relies on *comparing* the value of our outcome Y in two states of the world: the world where we do X, and the world where we don’t do X. But as we only get to live in one universe, we can never perfectly know what the value of our outcome Y would be in *both* a world where we do X and one where we don’t do X for a given entity at a given moment in time. As such, we can **never** directly measure the causal effect of X on Y for a given entity (say, a given patient or customer) at a given moment in time — a problem known as the **Fundamental Problem of Causal Inference** (causal inference being what people call the practice of answering Causal Questions).

To illustrate, suppose we were interested in the effect of taking a new drug (our X) on cancer survival (our Y) for a given patient (a woman named Shikha who arrived at the hospital on June 18th 2022). We can give her the drug and evaluate whether she is still alive a year later, but that alone can’t tell us whether the new drug *caused* her survival according to our counterfactual model of causality — after all, if she survives maybe she would have survived even without the drug! To actually know the effect of the drug on Shikha *by direct measurement,* we would have to be able to measure her survival both in the world where we gave her the drug *and* the world where we did not and compare outcomes.

Since we can never see both states of the world — the world where we undertake the action whose effect we want to understand and the world where we don’t — almost everything we do when trying to answer Causal Questions amounts to trying to find something we *can* measure that we think is a *good approximation* of the state of the world we can’t actually see.

A quick note on vocabulary: by convention, we refer to the action whose effect we want to understand as a “treatment,” and the state of the world where an entity receives the treatment as the “treated condition.” Similarly, we refer to the state of the world where an entity does *not* receive the treatment as the “control condition.” We use this language even when we aren’t talking about medical experiments or even experiments at all. We also refer to the state of the world we cannot observe as the “counterfactual” of the world we can observe — so the world where Shikha does not get the cancer drug is the *counterfactual condition* to the world where Shikha does get the drug.

It’s at this point most people start throwing out “but what about…“‘s, and that’s good! You should be — that’s exactly the kind of thinking you have to do when trying to answer Causal Questions. For example, “what about if we measured the size of Shikha’s tumor before she took the drug and compared it to the size of her tumor after? If the tumor got smaller as soon as she started the drug, then surely the drug caused the tumor to shrink!”

Maybe! Implicitly, what you have done is asserted that you think that the size of Shikha’s tumor before we administered the drug is a good approximation for what you think the size of Shikha’s tumor *would have been* had we not given her the drug.

But this type of comparison will always fall short of the Platonic ideal given by our definition of causality. Yes, Shikha’s tumor *may* have stayed the same size if we had not given her the drug (in which case the size of the tumor before she took the drug would be a good approximation), but it is also possible that regardless of whether we’d given her the drug, her cancer would have shrunk on its own.[1]

[1] The fact that diseases naturally change over time on their own is known as a disease’s “natural history.”

According to the Counterfactual Model of Causality, we could only ever *know* if taking the drug caused a decrease in tumor size if we could both administer the drug and observe the tumor *and also observe a parallel world in which the same person at the same moment in time was not given the drug for comparison*. And since we can never see this parallel world — the *counterfactual* to the world we observe — the best we can do is come up with different, imperfect tricks for *approximating* what might have happened in this parallel world, like comparing the tumor size before and after we administer the drug, imperfect though that may be.

So does that mean we're doomed? Yes and no. Yes, it *does* mean that we're doomed to never be able to take the exact measurements that make it possible to directly answer a Causal Question. But no, that doesn't mean we can't do anything — in the coming weeks, we will learn about different strategies for approximating counterfactual conditions, and in each case we will learn about what *assumptions* must be true for our strategy to provide a valid answer to our Causal Question. By making the assumptions that underlie each empirical strategy explicit, we will then be able to evaluate the plausibility of these assumptions.

In the example of Shikha, for example, we know that our comparison of tumor size before taking the drug to tumor size after taking the drug is only valid if her tumor *would not have gotten smaller without the drug*. This is something we can't measure directly, but we can look to other patients with similar tumors, or the history of her tumor size to evaluate how often we see tumors get smaller at the rate observed after she took the drug. If it's very rare for these types of tumors to ever get smaller, than we can have more confidence that a decrease in tumor size was the result of the drug.

We are also sometimes in a position to be more proactive than our effort to answer Causal Questions. Rather than trying to make inferences from the world around us using what is termed "observational data" (data that was generated through a process we did not directly control, a process we only "observe"), we can sometimes generate our own data through randomized experiments.

Randomized experiments — perhaps the most familiar tool for answering Causal Questions — are also just another way of approximating the unobservable counterfactual condition. In a randomized experiment — also known as "Randomized Control Trials (RCTs)", or "A/B Tests" whether you're hanging out with statisticians, doctors, or web developers — participants are assigned to either receive the treatment (the treatment group) or not (the control group) based on the flip of a coin, a roll of a die, or more commonly a random number generator on a computer. Provided we have enough participants, the Law of Large Numbers then promises that, *on average*, the people assigned to the control group will (probably) be "just like" the people assigned to the treatment group in every possible way (save being treated). Subject to a few other assumptions we'll discuss in great detail later, that means that the outcomes of the control group — being just like the treatment group *on average* — will be a good approximation of what *would* have happened to the treatment group in a world where they did not receive the treatment.

Randomized experiments are not a silver bullet, however. The validity of experimental comparisons still rests on a number of assumptions, many of which cannot be directly tested. For example, we can never be entirely sure that when we randomly assigned people to control and treatment groups, the process was truly random, or that we ended up with people who were similar in both groups (the law of large numbers only promises that getting similar groups becomes *more likely* as the size of the groups increases, not that it will happen with certainty!). Moreover, conducting a randomized experiment requires working in a context where the researcher can control everything, and that can sometimes generate results that may not generalize to the big messy world where you actually want to act.

10.3 So where does that leave us?

For many data scientists, this will feel *profoundly* dissatisfying. Many people come to data science because of the promise that it will provide direct answers to questions about the world using statistics. But because of the Fundamental Problem of Causal Inference, this will never be possible when answering Causal Questions. Rather, the job of a data scientist answering Causal Questions is a lot like the job of a detective trying to solve a crime — your task is to determine what *probably* happened at a crime scene. You can gather clues, collect forensic evidence, and interview suspects, all in an effort to come up with the *most likely* explanation for a crime. But no matter how hard you try, you can't go back in time to witness the crime itself, so you will never be able to be entirely sure if you are right or not.

But just as we investigate and prosecute crimes despite our inability to ever be 100% certain an arrested suspect is guilty, so too must businesses and governments make decisions using the best available evidence, even when that evidence is imperfect. But it is our job, as data scientists, to help provide our stakeholders with the best available evidence, and also to help them understand the strength of the evidence we are able to provide.

10.4 Why Passive-Prediction Is Not Enough

At this point, it is worth pausing to reflect on a question it may not have occurred to you to ask above — if answering Causal Questions is usually about *predicting* what would happen if we were to act on the world in a certain way, then how/why is it different from answering the kind of Passive-Prediction Questions we discussed previously?

There are a number of different ways one can frame the answer to this question, but the one I like most for Data Scientists is that when answering a Passive-Predictive Question, we can usually achieve our goals simply by identifying *correlations* that we think are likely to persist into the future. For example, suppose we run the maintenance department for a rental car company. The fact that a car whose *Check Engine* light is on is a car that is likely to break down if it isn't taken to a mechanic is enough for us to identify cars in trouble! Obviously, the *Check Engine* light isn't *causing* the cars to break down, but it doesn't have to to be useful.

But when seeking to answer Causal Questions, we wish to go beyond just identifying cars in trouble, and instead predict what might happen to cars if we *chose to act* in different ways. This requires going beyond simple Passive-Prediction because, in choosing to act, we are asking about how things might turn out in a world where we are behaving differently than we are currently — in other words, we are no longer being passive.

Thus, in a sense, answering Causal Questions is therefore *always* an example of "out-of-sample extrapolation" or "out-of-sample prediction", because by definition we are saying we want to know what happens in a world where at least one major agent — us! — changes their behavior. And indeed, there's a very real sense in which that's what we *mean* by a causal relationship: a relationship between our actions and an outcome that would persist even if we change our behavior!

What's a situation where a correlation is sufficient for Passive Prediction but not answering a Causal Question? Well, let's go back to our example of the rental car maintenance manager — suppose rather than using *Check Engine* lights to identify cars that needed more attention, the manager decided to just cut the cables that run to all the *Check Engine* lights! After all, the cars that are breaking down all have their *Check Engine* light on, and the cars that don't have their *Check Engine* lights almost never break down! So why not just disable the *Check Engine* lights on all these cars so they stop breaking down?

Now that we've been clear about what we mean when we ask "does X cause Y?", we can now understand why this is a perfect example of why correlation does not always imply causation.

Fundamentally, the manager is asking "would cutting the cables to the *Check Engine* lights prevent our cars from breaking down?" For that to be true, we know that in an ideal universe, we would want to compare a car on the verge of breaking down that has its *Check Engine* light intact to that same car in a world where we cut the *Check Engine* light — then we can see if there is a difference in whether these cars break down!

But this is *not* the data our manager has turned to draw their conclusion — rather, they are comparing cars with their *Check Engine* lights on and cars without their *Check Engine* lights on. And it turns out that cars *without* their *Check Engine* lights on are not a good approximation for the cars *with* their *Check Engine* lights on because the cars without the light on are different from the cars with the light on in ways that matter for the likelihood of breaking down (they have engine problems!) *other* than the *Check Engine* light!

Depending on what classes you may have taken in the past, you may have heard these differences referred to as "confounders" or "omitted variables" — those are just different words or ways of talking about the same idea! Confounders or omitted variables are just different words for features that are different between the "treated" and "untreated" observations being examined that the untreated observations are bad approximations of the counter-factual condition for the treated observations!

10.5 Next Steps

In this reading, we learned — in an intuitive sense — why answering Causal Questions is inherently hard. But this explanation, while accurate, is a little informal to be rigorous. In the readings that follow, we will be introduced to the *Potential Outcomes Framework* — the formal statistical framework that underlies the Neyman-Rubin Counterfactual Model of Causality. This framework will help us reason more systematically about how and when methods like randomized experiments, linear regression, matching, and differences-in-differences can help us answer Causal Questions.

But first, in the interest of not losing perspective on the forest for the trees, a discussion of *how* Causal Questions are used in practice.

CHAPTER

ELEVEN

CAUSAL QUESTIONS IN PRACTICE

In our last reading, we learned a little about what it means to measure a causal effect, and why it is inherently difficult. That is a topic we will return to shortly, as understanding why measuring causal effects is hard is key to being able to measure them effectively. But first, take a moment to discuss how Causal Questions come up and are addressed in practice to help contextualize the more technical readings that will follow.

11.1 When Do Causal Questions Come Up?

Causal Questions arise when stakeholders want to *do* something — buy a Superbowl ad, change how the recommendation engine in their app works, authorize a new prescription drug — but they fear the action they are considering may be costly and not actually work. In these situations, stakeholders will often turn to a data scientist in the hope that the scientist can "de-risk" the stakeholder's decision by providing guidance on the likely effect of the action *before* the action is undertaken at full scale.

Usually, the action the stakeholder is considering will not have been pulled out of a hat. Rather, a stakeholder will generally pose a Causal Question because they have some reason to suspect a given course of action may be beneficial. Indeed, Causal Questions often emanate from patterns discovered when answering Exploratory or Passive Predictive Questions.

11.1.1 Where Causal Questions Come From: An Example

For example, suppose the Chief of Surgery at a major hospital is interested in reducing surgical complications. They begin by asking "What factors predict surgical complications?" (a Passive Predictive Question) and develop a predictive model that allows the hospital to identify patients who are likely to have issues so that caretakers can provide additional support to these patients during recovery.

In the course of developing this model, the Chief discovers that one of the strongest predictors of surgical complications is patient blood pressure — patients with high blood pressure are substantially more likely to experience complications than those with normal blood pressure.

This leads the Chief to wonder about whether they could reduce surgical complications if they treated patients with high blood pressure with pressure-reducing medications prior to surgery. In other words, the Chief Surgeon wants to know "What effect treating patients with high blood pressure would have on surgical complication rates?"

But rather than just starting to give all patients with high blood pressure new drugs (and delay their surgeries while the drugs take effect), the Chief wants *you* to provide a more rigorous answer to their question. After all, high blood pressure may be *causing* the complications (and thus the medicine may help), but it could also be that high blood pressure isn't the *cause* of the complications, but rather the *symptom* of a third factor that causes both high blood pressure *and* complications that makes people with high blood pressure different from those with low blood pressure — like leading an overly stressful life. The Chief doesn't need to know whether high blood pressure is the *cause* of complications or just a "warning light" that identifies people at risk to use that information for directing additional support to those patients during recovery; but it *does* matter for determining whether it makes sense to delay surgeries to teach patient high blood pressure!

This is, of course, just one example, and it's not hard to imagine others. Perhaps your online retailer stakeholder has noticed that one of your competitors has stopped showing customer reviews in the search results, for example, so they suspect it must be improving sales for your competitor and want to know if it would work for your site too! But the point is that Causal Questions generally don't appear out of the blue, but rather because someone has noticed a pattern in the world and wants to act on it. Thus, many Causal Questions may actually take the form of hypotheses or hunches that your stakeholder wants investigated.

11.2 The Two-Fold Challenge of Causal Questions

In our last reading, we discussed how answering Causal Questions is difficult in part because measuring the effect of any action on any outcome is a definitionally difficult endeavor. But answering Causal Questions is also difficult for a more practical — less epistemological — reason: risk aversion!

As we noted above, stakeholders generally turn to data scientists because they want to know the likely consequences of an action *before they actually undertake the action at full scale.* This may seem obvious, but it bears repeating — not only is answering Causal Questions hard because we never get to measure outcomes in both a universe where our treatment occurs and also a universe where it does not (the Fundamental Problem of Causal Inference), but *also* because stakeholders want to know about the likely consequences of an action they aren't ready to actually undertake!

As a result, the job of a data scientist who wants to answer a Causal Question is to design a study that not only measures the effect of a treatment, but also does so in a setting that is enough like the context in which the stakeholder wants to act that any measured effect will generalize to the stakeholder's context.

We call these two objectives of a study *internal validity* (how well the study answers the Causal Question *in the setting the study is conducted*) and *external validity* (how well the results of the study generalize to the context the stakeholder cares about). And to provide value to a stakeholder, a data scientist's analysis must have both.

11.2.1 Internal and External Validity: An Example

To illustrate, suppose you work for a medical device company in Boston that wants the US Food and Drug Administration (FDA) to authorize a new cochlear implant your company has developed (a partially surgically implanted device for helping those with certain types of hearing loss regain hearing). Before authorizing the device, the FDA wants to be sure that it's safe and effective — in other words, it wants to know what the *effect* of authorizing the device for patients throughout the United States would be on patient health.

Your job, therefore, is to conduct a study that (a) convincingly measures the effect of the device on patients (has high internal validity), *and* (b) does so in a way that convinces the FDA that the findings from *your study* are likely to be the same as what would be seen if the device were being used across the United States (has external validity to the context the FDAs cares about).

In medical trials, internal validity is usually ensured by conducted a randomized experiment — referred to as a Randomized Control Trial (RCT) in medical circles — according to a set of FDA requirements. We'll discuss what features must be present for us to have confidence in the results of a randomized experiment soon, but they are things like making sure that the people in the control group look like the people in the treatment group in terms of things we can measure (age, gender, etc.) to help us feel confident that when people were randomly assigned to control and treatment groups, we didn't end up in a really unlikely situation where, purely by chance, only men ended up in control group and only women ended up in treatment group.

External validity, by contrast, comes from things like *who* is enrolled in the trial. The average age of children getting cochlear implants is between 2 and 3, so if your study only included children between 12 and 18 months of age, the FDA may worry that the results of the study would not *generalize* to the US population as a whole.

In the context of a clinical trial, this issue of external validity may seem easy to address — just get a sample of people who "look like" the US population (when applying for US FDA approval)! Historically, however, women[1] and minorities have been underrepresented in clinical trial participants.[2] Moreover, the people designing clinical trials often limit enrollment to participants who, aside from the specific condition being treated, are healthy to avoid complications. This reduction in complications may increase the *internal* validity, but as many patients face more than one health challenge, it may reduce external validity.

Outside of drug or medical device trials, however, external validity can much harder to establish. For example, the functionality of many internet services and apps depends on network effects — testing out a new social feature on Instagram by making it available to only a handful of users in a randomized trial (an A/B test, in the language of tech companies) may not give you a meaningful sense of how the feature would be used if it was visible to all users. And the way that bank customers use a new budgeting app in the context of a two-week study may not be indicative of how they would use it over the long run when the feature is no longer new.

11.2.2 External Validity To *What*

Throughout this text, we will refer to whatever course of action the stakeholder is actually considering pursuing as the "stakeholder's context." As a result, when discussing the external validity of a study, we will implicitly be referring to its external validity *to the stakeholder's context.* But it is worth emphasizing that while the internal validity of a study is a single things — you have some level of confidence that the study measured the effect they set out to measure — external validity is *relative*. A study conducted in a hospital in Denver, Colorado may have good external validity from the perspective of a doctor in a Pheonix, Arizona hospital, but that same study may not have very good external validity from the perspective of a doctor at a hospital in Pune, India. So always remember that external validity is not an absolute property of an analysis, but a property that is *relative* to the context to which one wishes to generalize the results.

11.3 The Causal Question Work Flow

Before we dive into the technical details of answering Causal Questions, it's worth starting with a high-level overview of how data scientists approach answering Causal Questions.

11.3.1 Identify Relevant Previous Studies

Once a Causal Question has been posed, the next step is **to identify any research that has already been done** that may help answer your causal question. It's hard to overstate how often this step is overlooked by data scientists, but it's *such* a no-brainer once you think of it! There's no reason to spend days or weeks trying to design a study to answer a question if someone else has already put the time and money into doing it for you!

If your stakeholder is somebody who works in public policy or medicine, then the first place to look for previous studies is in academic medical or policy journals. But don't assume that if you aren't working on a medical or public policy question that you won't be able to find an answer to your question in academic or pseudo-academic publications — lots of data scientists present research done at private companies at ``industry" conferences like the MIT Conference on Digital Experimentation (CODE@MIT) or the NetMob Cellphone MetaData Analysis Conference!

And if you are at a company, ask around! Someone at your own company may have looked into a similar question before, and talking to them could save you a lot of effort.

[1] In 1977, the FDA actually banned enrollment of women of "childbearing potential" from Phase 1 and Phase 2 clinical trials in the interest of avoiding birth defects.

[2] This seems likely to be due, in part, to hesitancy to enroll in clinical trials by individuals aware of past abuses of minority patients, as in the Tuskegee Syphilis Study.

11.3.2 Evaluate Previous Studies

If you do find studies, then for each study you will have to ask yourself two questions:

- **Did the study authors do a good job of answering the Causal Question? *in the context they were studying*?**
- **Do I believe that the *context* in which the study was conducted is similar enough to my own context that their conclusions are relevant to me?**

This first question is about the *internal validity* of the study, and we'll talk at length about how to evaluate that in the context of causal inference in the coming weeks. The second question is about the *external validity* (i.e., the generalizability) of the study to your context. There are lots of extremely well-conducted studies in the world that may be seeking to answer the same question as you, but if, for example, they investigated the effect of a new drug in *young* patients, and your hospital only treats very old patients, you may not be comfortable assuming their results are good predictors for what might happen in your hospital.

11.3.3 Plan A New Study

If you were unable to find any studies that answer your Causal Question satisfactorily (either on their own or in combination), then it may be time to do a study of your own!

When most people think about answering Causal Questions, their minds immediately jump to randomized experiments. Randomized experiments are *often* the best strategy for trying to answer Causal Questions, but they are not always the best choice.

Studies designed to answer Causal Questions can be divided into roughly two types: experimental studies and observational studies.

In an experimental study, a researcher has control over everything that happens in the study, including who enrolls in the study and also who in the study gets assigned to the treatment group and who gets assigned to the control group. Nearly all clinical trials, A/B tests where the version of a website or app users see is randomly determined, and field experiments where, say, voters are randomly assigned to receive different types of mailers from political campaigns to measure their effect on voter turnout are all examples of "experimental studies."

In an observational study, by contrast, researchers use data from a context where the researchers did not control who was treated and who was not. This includes data from public opinion surveys, data on user behavior and demographics, or census data.

(We say studies can be divided into roughly two types because some studies fall into a category sometimes called "quasi-experimental." In these studies, researchers were not in control over who was treated and who was not, but they have some reason for thinking that *something in the world* — like a chance storm, or a draft lottery — caused who was treated and who was not to be determined randomly. But these types of studies tend to be more relevant for academics than applied data scientists, and evaluating them is incredibly difficult, so we will largely ignore them in this text.)

While it is sometimes believed that only experimental studies can generate valid answers to Causal Questions, this is *unequivocally untrue*, as is the slightly more generous version of this claim, that experimental studies always constitute the best form of evidence for answering Causal Questions. As we will explore in *great* detail in the coming days, the validity of conclusions drawn from *both* experimental and observational studies rests on whether a number of fundamentally untestable assumptions hold. As a result, both types of studies are capable of providing meaningful answers to causal questions *and* of being deeply misleading.

Moreover, while experimental studies often (but not always) have greater *internal validity* (they are often better able to ensure that they have measured the true causal effect in the lab), this often comes at the expense of lower *external validity*, because ensuring the researchers can control who is treated and who is not requires operating the study take place in a highly monitored, often artificial and unrealistic setting. Observational studies, by contrast, are often based on data collected in the real world, and as a result may yield answers that tell us more about what is likely to happen in our own real-world application, even if they have somewhat lower internal validity.

11.4 Wrapping Up and Next Steps

Hopefully, this reading has given you a better sense of *how* Causal Questions are used to solve stakeholder problems, and when and where they come up in the life of a practicing data scientist. In the readings that follow, we will turn first to the details of the *Potential Outcomes Model*, a rigorous, formal statistical framework for understanding the Counterfactual Model of Causality. This framework will not only provide a presentation of the Counterfactual Model of Causality that may be appealing to those who draw intuition from mathematical formalism, but also machinery that we can use to evaluate how much confidence we can have in answers generated using different methods of answering Causal Questions — including both experimental and observational studies.

Part III

Data Science in Practice

CHAPTER

TWELVE

BACKWARDS DESIGN

Backwards Design is a way of developing an efficient strategy for completing a new data science project, and in my view it is one of the most important skills of a professional data scientist.

If you don't have a lot of professional data science experience, it may not be obvious why this is an important skill, or even why I call it a "skill." That's because most data science students' experience with project development comes from classroom exercises or sites like kaggle. These types of projects are excellent opportunities for learning, but it is usually the case that – unbeknownst to the student – these projects have been carefully tailored to have clearly defined goals, and they come with data sets that have been cleaned and filter to provide only relevant variables. This is usually done for good reasons – the instructors design these exercises in a way that focuses student attention on the skills that they are trying to develop (like model selection or model interpretation). But as a result, students often come away with the impression that most of what data scientists do is work with statistical models.

In reality, however, often the most important thing that a data scientist does is (a) develop and articulate a concrete, feasible objective of a data science project, and (b) develop a strategy for achieving that objective efficiently. And Backwards Design is one of the best ways to go about accomplishing both of these goals.

12.1 Overview

As the name implies, the idea Backwards Design is to *start* by clearly defining where you want to end up at the end of the project, and then working backwards to figure out exactly what you need to do to get there. Backwards Design is actually a common project management strategy and a range of different domains, and so you may already be familiar with the strategy and broad terms. In this class, however, we will focus on a five-step strategy for doing Backwards Design in data science:

1. Define the problem you want to solve.
2. Define a *question* that you wish to answer to help you solve this problem.
3. Articulate exactly what an answer to your question would look like.
4. Determine the variables you would need in order to generate that answer.
5. Identify data sets with those variables, and develop a strategy for bringing them together.

12.2 1) Define Your Problem

This first step should be the most straightforward, and yet you will be surprised at how often it is never actually explicitly addressed. People get so excited about the idea of data science that they will often come to you (the data scientist) with the data set and say "do some data science with this!" So the first thing you should always do when starting a data science project is make sure that you can clearly articulate the objective of the project. In addition, you should always make sure that *your stakeholder* agrees with that articulation of the problem you are trying to address! There's nothing worse than spending weeks on a project and then discovering that it's not actually a value to your stakeholder.

Here are a few examples of defined problems:

- We don't know how to reduce mass incarceration.
- My business can't identify potential new customers.
- We don't know who is going to develop Alzheimers, so we can't test early interventions.

12.3 2) Define the Question You Wish to Answer

Although not everyone will agree with us, it is my view the data science is fundamentally the practice of using data to quantifiably answer questions about the world.

For example, when we ran our regressions of birth weight on various demographic variables and whether the mother smoked during pregnancy, those models were answering the question "is maternal smoking associated with lower birth weight (at a statistically significant level after controlling for other confounds)?"

If someone who runs a commerce website runs an A/B test where users visiting the site are randomly assigned to see two versions of a new landing page, and we then track their purchasing behavior, then when we analyze that data statistically what we're doing is answering the question "Which of these designs is more effective at getting customers to buy things?"

And finally we can think of supervised machine learning algorithms as answering two types of question: there's the broad question that you answer by building a model and evaluating it ("can we identify cancer from patient x-rays, and if so, how well?"), and then the narrow question the is answered each time the model is run ("given this x-ray, how likely is this patient to have cancer?"). (There's a small digression on supervised machine learning and this "data science is about answering questions" conceptual framework at the end of this if you're interested).

So the next step in backwards design is to ask yourself: what question, if answered, would help you solve the problem that motivates you?

12.3.1 Defining a Good Question

A key feature of a good question is one that it is *concrete*, *tractable* and *answerable* by a data science project. Your question meets these criteria if it **directly implies how you should approach the data science project, and that approach seems feasible.** If your question is so vague that you don't immediately start thinking about the data you want to collect, it's not a good motivating question.

To illustrate, here are a set of *bad* questions to the three problems described above:

- What policies reduce mass incarceration?
- Can machine learning help me identify potential customers.
- What indicates Alzheimers?

By contrast, here's a set of concrete, tractable, and answerable questions:

- Does the availability of grand juries result in longer sentences? (Grand juries are a pool of citizens prosecutors can ask for guidance on sentencing. In theory they're supposed to hold prosecutors accountable, but in reality its often said that prosectors can shape the information grand juries get so they reach the recommendations that prosecutors wanted to begin with).
- What attributes are common to the customers who buy the most from my business?
- Are there lab results common in patients who later develop Alzheimers (diagnosed post-mortem) that we don't see in patients who don't go on to develop Alzheimers?

For the first set, we've asked questions, but they're so vague that it's not clear how you should approach answering the question. In the second set, by contrast, likely first steps are very clear. For example, the first good question clearly implies we need to find data on sentencing from places with and without grand juries. For the second question, its clear we need data on our current customers, and data from the general population for comparison. And for the last question we clearly want lab data from patients with and without diagnosed Alzheimers!

Moreover, for these answerable questions, you can imagine what the answer to the question will look like: for the first, you could have a regression that regresses sentences on grand jury availability controlling for crime committed; and for the second, you could imagine a table that compares various demographic characteristics of customers to non-customers.

12.3.2 Why is Having a Good Question Important?

- It will save you from getting lost in your data, since it helps you focus you energy.
- Being able to articulate the question you wish to answer allows you to make sure that answering that question will actually help address your motivating problem (/make sure that your stateholder agrees that answering your question will help them). There's *nothing* worse than getting excited, diving into your data, doing lots of work, and then getting a result that doesn't actually help address the problem that motivated you (but it happens *all the time*).

12.4 3) Write Down What An Answer Would Look Like

Seriously. Do it. Not abstractly: I mean draw the graph, figure, table, dataset with columns of predicted values and predictors, or set of model diagnostics you want to generate as a way of answering your question. Literally draw the graph, label the axes, etc.

Why?

- If you can't, then it turns out your question wasn't sufficiently concrete.
- You can then show this to your stakeholder to ensure they think this constitutes an answer that would help them solve their problem (and avoid later being told your work doesn't actually help them)
- It makes it even clearer what steps you need to do next to generate this result.

12.4.1 Falsifiability

OK. So now you're written down what an answer to your question looks like. But there's one other key feature of a good question that we didn't get into above: it should be *falsifiable*, which means that (a) you should be able to articulate a hypothesis about the answer to your question *and* (b) know what a result to your question would look like.

So when writing down what your answer should look like, do the following:

1. State a hypothesis about what you think the answer to your question is likely to be.
2. Draw what an answer to your question would look like *if your hypothesis is true*.
3. Draw what an answer to your question would look like *if your hypothesis is false*.

For example, consider our question about grand juries and sentecing. My hypothesis might be that grand juries result in longer sentences because they insulate prosectors from accountability.

My result, as described above, could be a regression table where I regress sentences on whether a county has a grand jury along with controls for crime committed.

The answer if my hypothesis is true is that we'd have a positive coefficient on the presence of grand juries.

The answer if my hypothesis is false is that we'd have a zero or negative coefficient on the presence of grand juries.

Why do all this? Because it helps ensure that the way we're planning to answer our question will actually answer it by generating different results in different states of the world.

12.5 4) What Data Do You Need?

Congratulations! You've now completed the really hard part of a data science project: defining your goals. Now we turn to the easy stuff.

First, now you know your goal – the result described above – we turn to how we will actually answer our question. So ask yourself:

- What variables do I need to make the result I described above,
- What population do I need represented in that data

So let's think about our business trying to find new customers. Clearly, we need data on (a) customer spending, and (b) the demographics of those same customers.

We also need data on *both* people who spend a lot, and people who don't spend a lot so we can compare these two populations. We could do this either by getting data on all our customers and comparing big spenders from people who don't buy much (if there's a lot of variation in the data on level of spending), or we could compare current customers to the general public (most of whom aren't customers).

12.6 5) Where Can You Get That Data?

OK, now you know the variables you need measured and the populations for whom you need those variables. Now let's figure out:

- Where can I get that data, and
- If the data will come from different datasets, how will I combine them?

In the customer example above, for example, we can start by looking for the data the company already has on its current customers. What's in that data?

We can also look for similar demographic data for non-customers using a resource like the American Community Survey (the annual survey run by the US Census bureau).

If we use government data for our comparison group, we'll then have to make sure we get comparable samples, so we'll want to make sure we can match our observations on things like age, gender, and where people live, so we'll need to make sure we have those variables in both datasets.

12.7 Wrapping Up

Congratulations!

By the time you've done these 5 steps, you've managed to develop a concrete plan for exactly where you'll focus your time, and you've nearly guaranteed that the result you're working to generate will actually be useful in solving the problem that motivates you or your stakeholder. There's still a lot of data wrangling, model selection, etc. ahead, but at least you won't get lost in your data, or do lots of work that ends up no helping anyone!

Want a template for this? Great news! You can download one here!

12.8 A Digression on Supervised Machine Learning

As noted above, we can think of superived machine learning as a tool that answers two types of questions: the first is the broad question of "can we predict [outcome of interest] using [variables we have] and the training set we have access to?", and the second is the more narrow prediction question "for a given set of predictors, what value of [outcome of interest] would the model predict for a given observation"?

The first, I think, is pretty straightforward. But there's a nuance to the second question that's super important to understand: when we ask a supervised machine learning it's prediction for a given observation, what we're fundamentally asking your model is: **"how do you think the entity who labeled the data in your training data set would label this new observation?"**

Because that's all that supervised machine learning does: it develops models that are designed to replicate the behavior that gave rise to the data set due used for training your model. For example, if you train a supervised machine learning algorithm to label pictures with the name of the animal in the picture by feeding it a bunch of pictures that have been labeled by undergraduates at an American university, than what you are training that machine learning algorithm to do is answer the question "how would an American undergraduate label this photo" every time it sees an unlabeled photograph.

Obviously different supervised machine learning algorithms go about trying to answer this question in different ways, and some will be more successful than others depending on the context (which is why we spend so much time studying model selection in machine learning courses), but answering this question is *always* the goal to which they aspire.

This is a bit of a digression, but I think it's an important one: recognizing that this is all supervised machine learning algorithms do is important because it helps you, the data scientist, understand the limitations of supervised machine learning algorithms. For a surprisingly long time, people thought that machine learning algorithms were incapable of harboring racial or sexist prejudices. They are, after all, just built of math, and math can't be racist, can it? And so companies like amazon tried to build supervised machine learning algorithms to help them decide who to hire. The problem, though, is that they trained them using data on which people human employees had decided to hire in the past, and data from subjective employee evaluations that had been made by human supervisors. And because this gave rise to an algorithm that looked at people's resumes and asked itself "what would Amazon's (very human) hiring staff and supervisors have thought of this person?," the algorithm of course inherited all the biases of those humans. And so, OOPS!, when Amazon suddenly realized that "their new recruiting engine didn't like women," they had to abandon the project.

OK, digression on bias in data science complete. For now. :)

CHAPTER THIRTEEN

MAKING DECISIONS USING DATA

(This is something that I know I need to add to my classes but I haven't really managed integrated yet. I talk to my students about that fact that they should just pray at the altar of $p<0.05$, but instead weigh the relative costs and benefits of Type 1 and Type 2 errors in the context in which they are trying to make a decision. But where I really struggled is the fact that you can't directly map P values onto decision theory very easily because of all of the weirdnesses of frequentist P values—e.g. A p-value of 0.05 means that under the null, the odds of a Type 2 is 5%… but that means the ACTUAL odds of a Type 2 error if you have a p-value of 0.05% is pr(null is true) * 0.05. 🤷‍♂.)

CHAPTER

FOURTEEN

WRITING DATA SCIENCE REPORT FOR NON-TECHNICAL AUDIENCES

As a data scientist, you'll often be required to summarize your analyses and present them to non-data scientists. This type of translation of technical analyses to something of use to less-technical audiences is an absolutely critical part of being an effective data scientist – if you don't communicate what you've done to decision makers, it often doesn't matter how rigorous or careful your work has been up to that point.

With that in mind, here is an outline of one strategy for writing for non-technical audiences. Obviously different people may prefer slightly different approaches, but I think this is a good model to start with.

Also, note that this *is* the model I'd like you to use when writing your final report for this class, so there are a few notes that are specific to class expectations!

14.1 Identify your audience

Before you write a single word, you should pause to reflect on exactly who you wish to address with your report, and their background. What follows are general guidelines, but the better you know your audience, the more precisely you can tailor the level of detail in your report.

For this class: At the top of your report, please specify the stakeholder to whom you are addressing your report – a product manager, a legislative aid, a policymaker, etc. This stakeholder should be relevant to your study, but should not be someone with data science training. You may assume they know about basic statistical concepts (means and standard deviations), but no more (no assumed understanding of potential outcomes, the theoretical underpinnings of experiments, specific designs like differences-in-differences, etc.). Obviously this is not something you'd put in a real write-up, but will be helpful for evaluation of your project.

14.2 Introduction / Executive Summary

One of the most important things to remember when writing up an analysis is that the person your writing to has too many things to do, and is **definitely** less interested in your project than you are. With that in mind, it's important that you write and organize your report in a way that catches their attention early and gets them invested so they keep reading. As a result, one generally wants to start with the most important parts of the analysis, then slowly draw back and lay out additional details.

You may have never noticed this before, but this is how most news articles are written: one of the first two or three paragraphs is what's referred to as the "nut graf" (or nutshell paragraph) in which the journalist basically summarizes the entire news article in a single paragraph. In the words of Ken Wells from the Wall Street Journal, the nut graf is "a paragraph that says what this whole story is about and why you should read it. It's a flag to the reader, high up in the story: You can decide to proceed or not, but if you read no farther, you know what that story's about.

Thankfully you probably aren't *so* pressed for time that you have to summarize everything in a single paragraph, but we will follow a similar structure in which we try and give the reader a full summary of why your project is important, how you do your analysis, and broadly what you conclude up front. In particular, I would argue that your introduction / executive summary should be organized as follows:

Identify the problem you wish to address

The first thing to do in any report is *motivate* your analysis – tell us about *why* you need to undertake this project. At this point in the report, keep this relatively brief – the motivation for the project is important, but you don't want to drown the reader in background. This should probably be one-to-two solid paragraphs. But don't draw it out – we can get more into background on the problem later, and you don't want to get bogged down talking about the problem, you want to get to how you're gonna help the reader.

What question will you try to answer, and how will it help you address

Here's the linchpin of the report: announce the question you're seeking to answer in your project *and* make it clear how this will help address the problem you've identified. This transition is where you will either get the reader to buy into the report and read it carefully, or lose their interest.

Summarize your strategy

Now in one to two paragraphs provide an overview of your project, your approach, and a preliminary summary of your results.

In all, you should have covered all this is about one page, maybe a page and a half, and *hopefully* now you've got your reader hooked!

14.3 Background

OK, so at this point you've hopefully caught your readers interest, so now you can circle back and provide any additional background needed to help the reader better understand your motivation or the specific context you are analyzing (if you're looking at a policy change, the details of the policy, the context in which it occurred, the players involved, etc.) The amount of background needed will vary across projects, but whatever you need goes here.

14.4 Your Design

Here's where you lay out how you plan to answer the question you laid out in your summary.

As you do so, bear in mind the difference between your goals in writing to a non-technical stakeholder and your goals when writing to a fellow data scientist (most of your professors).

When writing to a fellow data scientist, you're generally writing to a skeptical audience. Your goal is to try and convince them that you did everything correctly – crossed every t and dotted every i. This is especially true when writing to professors in technical classes, since you're usually trying to demonstrate your mastery of a technical skill, which means communicating very detail.

But a stakeholder reading your analysis is generally someone who mostly decided to put their trust in you when they hired you, and at this point your job is not to convince them every technical nuance of the project is right – by definition, most non-technical audiences wouldn't be able to read a balance table showing that your randomization created balanced samples – but rather to communicate to them **the key take-aways** of the analysis.

That's not to say you don't need to engage with some technical aspects of your project. For example, if you're using a good causal design, it's critical the reader know why your causal research design is better than just looking at observational data in a regular regression (especially since someone else may try and argue with your results using that type of data). And of course they need to know about any limitations of what you've learned. But you don't have to put every bit of due deligience you've done in the main report.

With that in mind, one thing that's crucial to this section if you're doing causal inference is to help the reader **understand why you're using a specific causal design without using technical language.** To do so, you want to lay out *specific, concrete reasons* that just using observational data might lead to erroneous conclusions (e.g. do the same thing you did on the homework assignments / midterm when asked about how people were interpreting observational studies.)

For example, if you are doing an experiment to see how sending people coupons would impact consumer behavior, you want to explain that "we can't just use data on sales from stores that chose to send out coupons to evaluate whether we should be sending out coupons to all our customers because it's possible that the stores that sent out coupons did so precisely because they knew that their customers were struggling financially, and thus needed coupons to be able to afford products. As a result, if we compared sales to customers who got coupons to those who did not, we might inadvertently assume the lower sales to customers who got coupons was the result of the coupons, when in fact it actually reflected the fact that the coupons went to customers who were less well-off financially to begin with."

"But if we run an experiment in which we randomly assign customers to either receive a coupon or not, then we know that on average the people getting coupons will be the same as the people not getting coupons (since who gets coupons is random, and not related to anything like customer income). As a result, we can compare sales to customers who got coupons and those that did not, and infer with confidence that any difference we see is the result of getting coupons, not other differences in the customers with or without coupons."

(See? No discussion of potential outcomes or use of terms like "baseline differences!"

14.5 Your Results

Now it's time for results! As with your design, remember your goal is to emphasize the key take-aways of your analysis, which means both what the data can tell you *and what it can't*. Remember that honest humility is a key part of being a good data scientist – don't over-sell your results!

14.6 Conclusions

Now the final part of the project – quickly recapitulate the problem you wanted to address, the question you sought to answer, the answer you reached, and the implications of this result. In this discussion, make sure you talk a lot about *external validity*: where are these results likely applicable? Where are they not? What other research could be done to learn more? Do you have concrete recommendations?

14.7 Appendix

Remember when I said that in writing to a non-technical stakeholder, you don't have to detail all the nuances of your analysis? Well… that's true. BUT: it's often good to put the details of all the careful analyses of robustness and diagnostic tests you completed in appendices. That way you can reference them in the body of your report (communicating in broad terms that you were careful without boring your reader), but then also include them in case your stakeholder wants to share your report with another data scientist for a second opinion.

So you probably want (**and for this class, should have**) an appendix with things like balance tests, A/A tests, evidence of parallel trends, discussion of why you chose certain sample restrictions, alternate specifications, etc., depending on what's appropriate for your particular research design.

Part IV

Advanced Topics

CHAPTER

FIFTEEN

INTERPRETABLE MODELS

For most applications, perform just as well as fancy pants models. https://arxiv.org/abs/1811.10154 Allow for non-specialists to see what goes into a model to debate ethics (after all, we data scientists have specialized knowledge when it comes to statistical methods, but not ethics)