Computational Causal Inference for Applied Researchers

A Practical Guide for Epidemiologists and Biostatisticians

Miguel Angel Luque-Fernandez Matthew J. Smith

2026

Table of contents

Preface

A question often asked by anyone is *"what would have happened if we had done this instead?"* The answer is impossible to know for certain, but there are mathematical methods that allow us to estimate this answer. These methods are called **causal inference**. This book introduces the concepts of causal inference from a beginner's perspective and leads the interested reader to numerous approaches to answer these impossible questions.

The purpose of many health studies is to estimate the effect of an exposure on an outcome. It is not always ethical to assign an exposure to individuals in randomised controlled trials; instead, observational data and appropriate study design must be used. There are major challenges with observational studies, one of which is confounding that can lead to biased estimates of the causal effects. Controlling for confounding is commonly performed by simple adjustment for measured confounders; although, often this is not enough. Recent advances in the field of causal inference have dealt with confounding by building on classical standardisation methods. However, these recent advances have progressed quickly with a relative paucity of computational-oriented applied tutorials, contributing to some confusion in the use of these methods among applied researchers.

In this book, we show the computational implementation of different causal inference estimators from a historical perspective, where different estimators were developed to overcome the limitations of the previous ones. We introduce the potential outcomes framework, illustrate the use of different methods using examples from health care settings, and most importantly, we provide reproducible and commented code in **R** and **Stata** for researchers to apply in their own observational studies.

The code can be accessed at github.com/migariane/TutorialCausalInferenceEstimators.

Who this book is for

This book is targeted towards epidemiologists, statisticians, psychologists, economists, sociologists, political scientists, and computer scientists — anyone interested in learning and applying causal inference methods to real-world data.

Author Biographies

Dr Miguel Angel Luque-Fernandez is an Associate Professor of Biostatistics in the Department of Statistics and Operations Research at the University of Granada (UGR), Spain, and an Honorary Associate Professor at the London School of Hygiene and Tropical Medicine (LSHTM). He holds a PhD in Epidemiology and Public Health (UGR/ULB), an MSc in Biostatistics (Newcastle), an MSc in Epidemiology (ULB), and a BSc in Mathematics and Statistics (Open University). His research focuses on causal inference methods, comparative effectiveness research, and computational epidemiology.

Dr Matthew J. Smith is a researcher at the London School of Hygiene and Tropical Medicine, specialising in causal inference methods and their application to population health research.

Part I

Foundations

1 Introduction

The first chapter provides a brief introduction to causal inference and its links to public health, economics, and society.

1.1 Causal inference

Causal inference is the process of determining whether a variable causes a change in another variable. It involves identifying causal relationships between variables based on data and statistical analysis. Causal inference is important because it allows us to understand how the world works and make informed decisions based on that understanding. For example, in medicine, we use causal inference to determine whether a particular treatment is effective and to identify potential side effects. In public policy, we use causal inference to assess the impact of interventions on social outcomes, such as crime rates, educational attainment, and economic growth.

However, determining causality is not always straightforward. Correlation between two variables does not necessarily imply causation, and there may be other factors, known as confounding variables, that are responsible for the observed relationship. Causal inference methods attempt to control for confounding variables and identify the true causal relationship between variables.

To make causal inferences, we need to go beyond mere associations between variables and determine whether a change in one variable actually causes a change in the other variable. This involves controlling for confounding factors and using methods such as randomized controlled trials (RCT), natural experiments, and observational studies to isolate the causal effect. For example, suppose we are interested in determining whether a new drug is effective in reducing blood pressure. A randomized controlled trial might be conducted, where a group of patients are randomly assigned to receive either the new drug or a placebo. By controlling for other factors that could affect blood pressure, such as diet and exercise, and randomly assigning patients to groups, we can attribute any differences in blood pressure between the two groups to the drug and infer a causal relationship. In an RCT, participants are randomly assigned to either a treatment group (where they receive the intervention being studied) or a control group (where they do not receive the intervention). This random assignment helps to balance out potential confounding factors between the two groups, making it more likely that any observed differences between the groups are due to the treatment.

In observational studies, this randomisation process is often not possible because it might be unethical or unfeasible to allocate individuals to certain treatments (or exposures, policies, etc.) i.e., smoking. Since, in observational studies, individuals cannot be randomly assigned to a treatment group, statistical methods are required to control for confounding and to infer causal effects.

Causal inference is important for applied researchers because it allows them to make informed decisions and draw meaningful conclusions about the world. By understanding the true causal relationships between variables, applied researchers can develop effective interventions, evaluate the impact of policies and treatments, and gain a deeper understanding of how different factors interact to produce outcomes. For example, consider a public health researcher who is interested in understanding the relationship between air pollution and respiratory illness. Without causal inference, the researcher might observe a correlation between higher levels of air pollution and higher rates of respiratory illness, but would not be able to determine whether air pollution actually causes respiratory illness. By using causal inference methods the researcher can attempt to control for confounding factors and isolate the causal effect of air pollution on respiratory illness. This information can be used to develop targeted interventions to reduce air pollution and improve public health outcomes.

Causal inference is also important for evaluating the effectiveness of medicine, policies, economics, societal changes, and other contexts. By understanding the true causal relationships between variables, researchers can determine whether a particular policy is effective, and identify factors that may be limiting its effectiveness. This information can be used to make more informed decisions about resource allocation and program design.

Overall, causal inference is an essential tool for applied researchers in a wide range of fields, from public health to economics to education. By understanding the true causal relationships between variables, researchers can make more informed decisions and develop more effective interventions, leading to improved outcomes for individuals and communities.

1.2 Causal inference roadmap

Constructing a causal analysis in a structured manner is necessary to obtain unbiased effect estimates and robust conclusions for real-world evidence. The *Causal Roadmap* (Figure 1.1) offers a framework that can be adapted to the vast majority of studies to generate real-world evidence. The Causal Roadmap begins with defining the causal question of interest, stating whether the data and assumptions can be used to answer the question of interest, performing suitable statistical analyses, and assessing whether alternative conclusions could be obtained under alternate assumptions. The rest of this section describes each step in more detail.

Step 1: Causal question, model, and estimand

First, a causal question is defined. The question encapsulates the population of interest (eligibility criteria), the treatment (or exposure), the follow-up period (time from starting-point

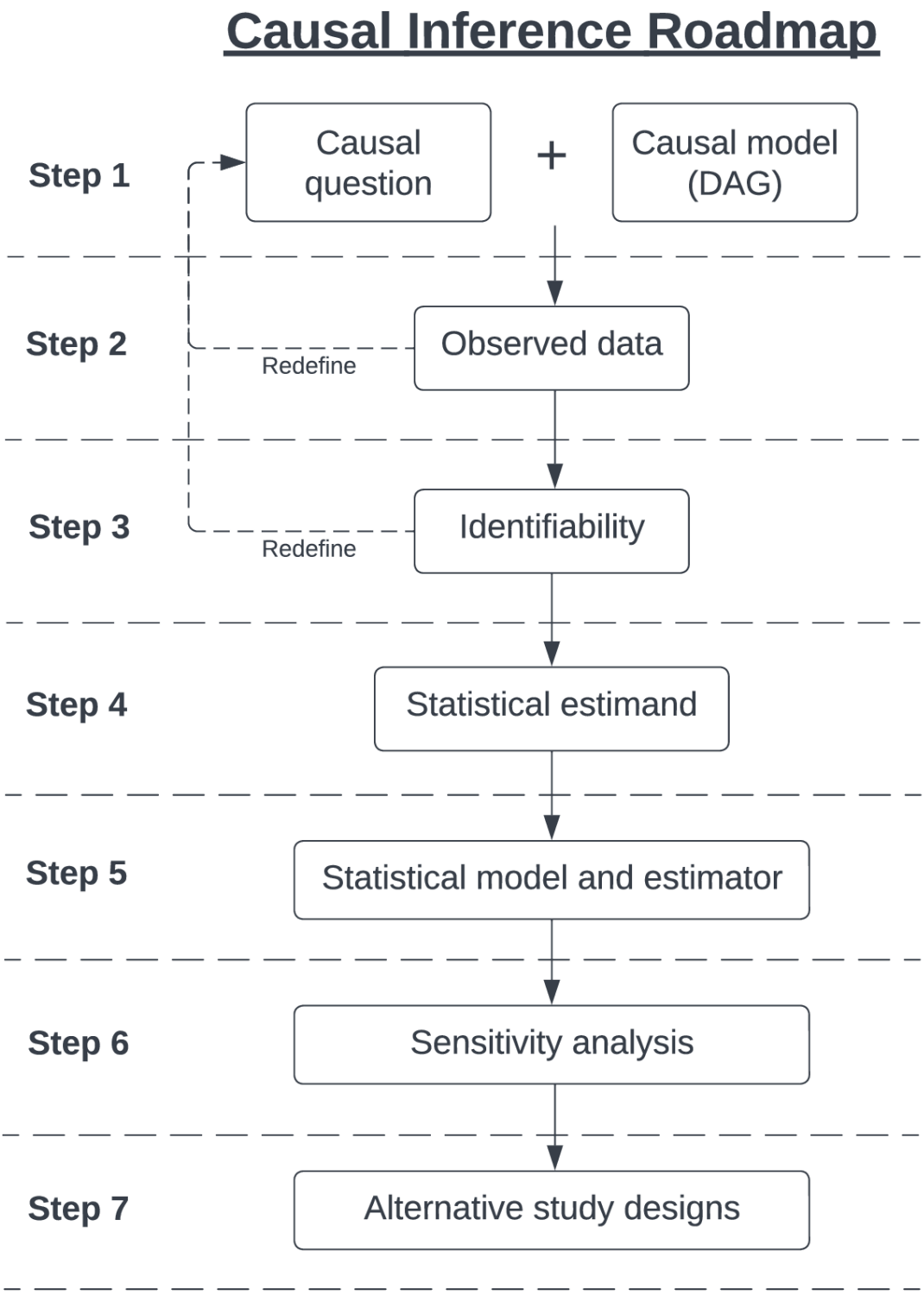

Figure 1.1: Causal Inference Roadmap as detailed in (Dang et al., 2023).

to end-point), the outcome of interest, and the causal estimand. The causal estimand is a description of the statistical estimate that answers the question (e.g., causal risk difference, causal relative risk, causal odds ratio).

The causal model is a graphical tool that helps to identify the causal pathways between the exposure and the outcome. The causal model is commonly shown using a directed acyclic graph (DAG): detailed explanation of causal models is in Section 1.3 Briefly, a diagram for the causal model shows what we know and, importantly, what we do not know about how the data is generated in the real world. The diagram is created using background knowledge, advice from experts, and assumptions about other possible variables.

It is important at this stage to go back to the causal question, we must critique whether the causal model is able to answer the causal question. The causal model could have identified a variable that has been overlooked and, if so, the causal question needs to be adapted: we call this process "redefining the causal question".

Step 2: Consider the observed data

The causal model from Step 1 illustrates our knowledge of how the exposure causes the outcome and provides the necessary information that we need to answer the causal question. The observed data, on the other hand, can differ from the causal model. The difference can occur in the way the observed data was measured. For example, research has shown that obesity is strongly associated with cardiovascular disease. One way to measure obesity is by using body mass index (i.e., $weight(kg)/height(m^2)$). However, body mass index (BMI) cannot distinguish between body fat and muscle mass, thus the BMI will be overestimated amongst people with lots of muscle mass and will be underestimated amongst people with very little muscle mass. Alternatively, obesity might be a variable that confounds the association between treatment and risk of mortality, but if obesity is not measured (thus not recorded in the data), we are unable to adjust for obesity. In such cases, we must go back to Step 1 to redefine the question of interest.

Step 3: Identifiability assumptions

If the observed data is sufficient to answer the causal question, we must first make certain assumptions. The main assumptions. also known as (AKA) as identifiability assumptions, are conditional exchangeability, consistency, no interference, and positivity. Identifiability in causal inference refers to the fact that the main assumptions are necessary before one can infer a causal effect, they are discussed in more detail in Section 1.5 The identifiability assumptions provide a way of writing the causal question (a hypothetical two-world causal contrast in terms of potential outcomes see 1.4) in terms of a model for the observed data.

Step 4: Define the statistical estimand

If the identifiability assumptions are deemed plausible, we can move on to defining the statistical estimand. The statistical estimand is an algebraic definition of the causal question but

in terms of the observed data. For example, the average treatment effect (ATE) measured by the risk difference is given by

$$ATE \quad = \quad E_w(P[Y = 1 \mid A = 1, \mathbf{W}] - P[Y = 1 \mid A = 0, \mathbf{W}])$$

and is the expected difference (causal risk difference) in the outcome (Y) between two exposure groups (A) conditioned on the set of variables that confound the exposure-outcome relationship ($\mathbf{W}$). More details on estimands and measures of association are given in Section 1.6

Step 5: Statistical model and estimator

The statistical model defines the set of possible data distributions between the outcome, exposure, and covariates. We must consider the functional form of the variables and their relationships to one another (e.g., non-linear terms, time-varying effects, interactions [effect modification], etc.). With the rise in quality and availability of machine-learning methods, the statistical model is at less risk of misspecification: a common source of bias.

Once the statistical model has been defined, the next step is to choose the estimator. In large part, this book focuses on the various statistical estimators that can be used, along with the suitability to certain data. The choice of the statistical estimator is determined by the performance of the estimator in terms of bias, 95% confidence interval (CI) coverage, type I error rate, and precision.

Step 6: Sensitivity analysis

Once the statistical estimator provides a quantitative value for our causal question, we must go back to the identifiability assumptions and ask how our result would change if the assumptions were violated. For example, if we were not able to measure a potentially important variable, then our result is potentially biased. We would need to assess whether the unmeasured variable is a strong confounder of the relationship between the treatment and the outcome. This topic of sensitivity analysis is revisited in Chapter 9

Step 7: Alternative study designs

We might often find that multiple study designs are feasible to answer the causal question. The researcher will then need to consider which study design is feasible, and ethical, whether results will be obtained in time for a policy-related decision, and other statistical properties (e.g., power, correct type I and II error rates, bias, coverage, precision, etc.). We do not go into alternative study designs in this book, we refer the interested reader to (Dang et al., 2023) for more details.

1.3 Causal diagrams

Introduce causal diagrams, which are graphical representations of causal relationships. Explain how they can be used to identify confounding variables and to determine which variables should be controlled for in an analysis.

1.3.1 Directed Acyclic Graphs

The distribution of the observed data can be shown as a graphical representation (Figure 1.2): these diagrams are known as direct acyclic graphs (DAG). When constructing DAGs, subject-matter knowledge must be used to ensure the conditional exchangeability assumption holds.

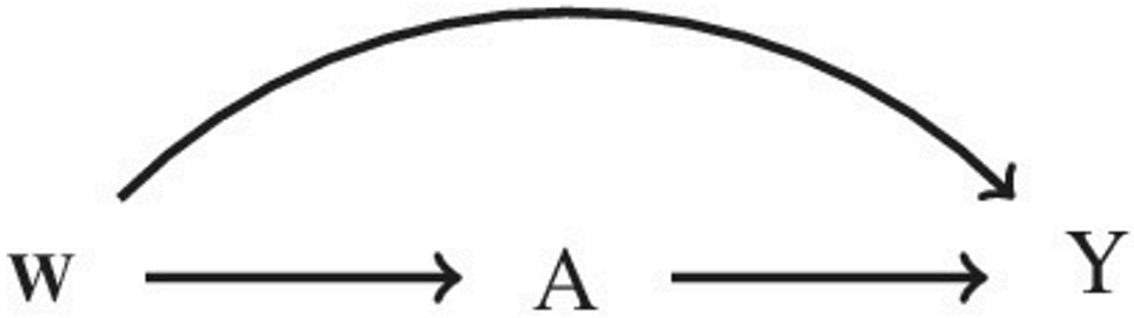

Figure 1.2: Sufficient set of variables to control for confounding. **Y**: outcome, **A**: treatment, **W**: confounders.

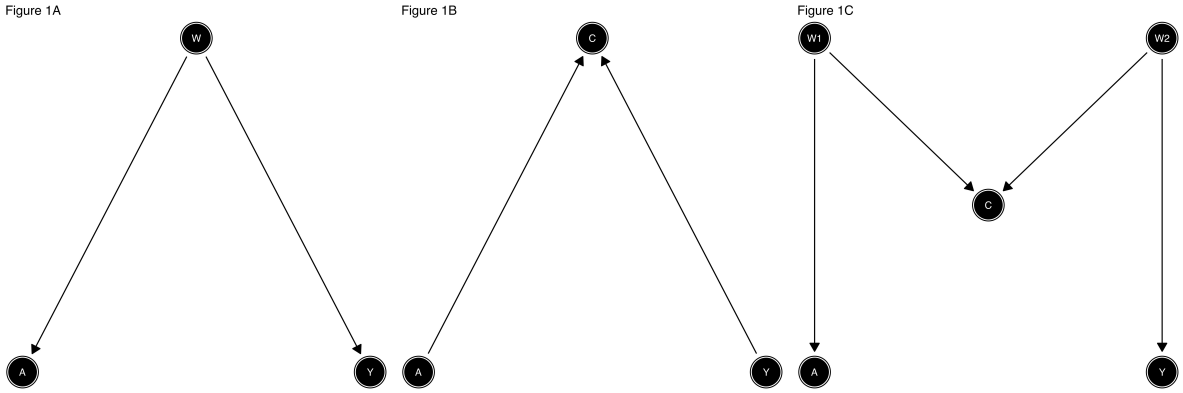

Figure 1.3: Three types of DAGs: causal fork (confounding), inverted fork (collider), and chain (mediation). Adapted from Luque-Fernandez et al.

This causal diagram makes several assumptions: all variables that confound the exposure-outcome relationship are included in **W**, there are no intermediate variables, and there is no residual confounding. Therefore, the set of covariates included in **W** suffices to assume the conditional mean independence to estimate the ATE. To more formally illustrate DAGs, we first define some terminology along with examples using the DAG in Figure 1.3

1. **Path**: A path is any series of nodes from $W_k$toW_l connected by an edge in any direction (i.e., the edge can be "$\rightarrow$" or "$\leftarrow$"). For example, $W_3 \rightarrow W_4 \leftarrow W_2 \rightarrow A \leftarrow W_5$is one of

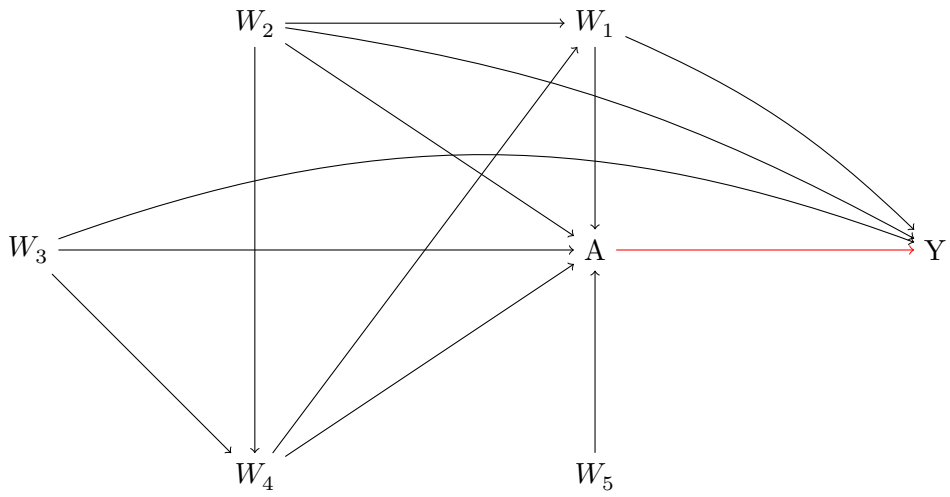

Figure 1.4: Causal diagram

the paths fromW_3toW_5.

2. **Direct path**: A direct path is a path between nodes that involves only forward edges. For example, $W_3 \rightarrow W_4 \rightarrow W_1 \rightarrow A \rightarrow Y$, is a direct path between $W_3$and$Y$ since it contains only forwards edges (i.e., "$\rightarrow$").
3. **Parent** and **child**: $W_k$is a *parent* ofW_l, and $W_l$is a *child* ofW_k, if $W_k \rightarrow W_l$. In Figure 1.3, $W_3$is a parent ofW_4, and $W_4$is a child ofW_3.
4. **Ancestor** and **Descendant**: $W_k$is an *ancestor* ofW_l, and $W_l$is a *descendant* ofW_k, if there is a direct path from $W_k$toW_l. In Figure 1.3, $W_3$is an ancestor ofW_1, thus $W_1$is a descendant ofW_3.
5. **Collider**: A node $W_k$is a collider between$W_{k-1}$and$W_l$if it receives both edges (i.e.,$W_{k-1} \rightarrow W_k \leftarrow W_l$). For example, $W_4$is a collider along the path$W_3 \rightarrow W_4 \leftarrow W_2$.
6. **Instrumental variable**: A node is an instrumental variable if it satisfies these assumptions:

    1. The node is correlated with A,
    2. the node is not correlated with Y, and
    3. the node is not correlated with a confounder that affects Y. In Figure 1.3, W_5 is an instrumental variable since it satisfies all three assumptions.

7. **Conditional instrumental variable**: A node is a conditional instrumental variable if it satisfies 6(a) and 6(b), conditioning on nodes that are confounders of $A \rightarrow Y$. In Figure 1.3, $W_4$is an instrumental variable conditional onW_1, W_2andW_3.

In Figure 1.3, one would need to condition on W_1, W_2, and $W_3$to sufficiently control for confounding. Without conditioning on these variables, the crude association between$A \rightarrow Y$ is biased (i.e., different from the true causal effect).

A collider for a certain pair of variables (e.g., outcome and exposure) is a third variable that is caused by both of them. In DAG terminology, a collider is the variable in the middle of an inverted fork (i.e. variable C in A → C ← Y).(Pearl, 2009; Pearl & Robins, 1995) Using regression to control for a collider, or stratifying the analysis concerning a collider, can introduce a spurious association between its causes, which can potentially introduce non-causal associations between the exposure and the outcome. This has been used to explain why the medical literature contains many paradoxical findings, where established risk factors appear protective for the outcome.(Brian W. Whitcomb, 2009; Hailey R. Banack, 2013; Miguel Angel Luque-Fernandez, 2016; S. Hernandez-Diaz, 2006) For instance, numerous studies have reported a paradoxical protective effect of maternal cigarette smoking during pregnancy on pre-eclampsia, which has been named the pre-eclampsia smoking paradox. This paradox is due to gestational age at delivery, which is a collider between smoking (exposure) and pre-eclampsia (outcome).(Miguel Angel Luque-Fernandez, 2016) However, the magnitude of the resulting bias will depend on the associations between the collider and the two parent variables.

A note on backdoor paths

To ensure conditional exchangeability holds, variables along the *path* from $A$to$Y$must be conditioned on. This is known as Pearl's backdoor criterion.(Judea, 1994) In Figure 1.3, conditioning on onlyW_1, W_2, and $W_3$was sufficient to control for confounding. There are no other *paths* throughW_4(orW_5) that does not already control for W_1, W_2, or W_3. In other words, if one wanted to navigate from $W_4$to$Y$, one would have to go through either W_1, W_2, or W_3, which are already controlled for.

1.3.2 Impact of colliders

To illustrate the induced association of conditioning on a collider, three variables are defined A (unrelated to B), B (unrelated to A), and C (collider, a child of A and B). The association between these variables are shown in Figure 1.5.

We now simulate some data for A, B and C and tabulate the data in Tables Table 1.1, Table 1.2, and Table 1.3

In Table 1.1, there is an equal number of those with A=1 amongst levels of B. In the DAG above, A does not cause B, so the estimate of the causal effect (i.e., odds ratio) should be 1.00.

Table 1.1

	B=1	B=0	**Total**
A=1	20	20	40
A=0	80	80	160
Total	100	100	200

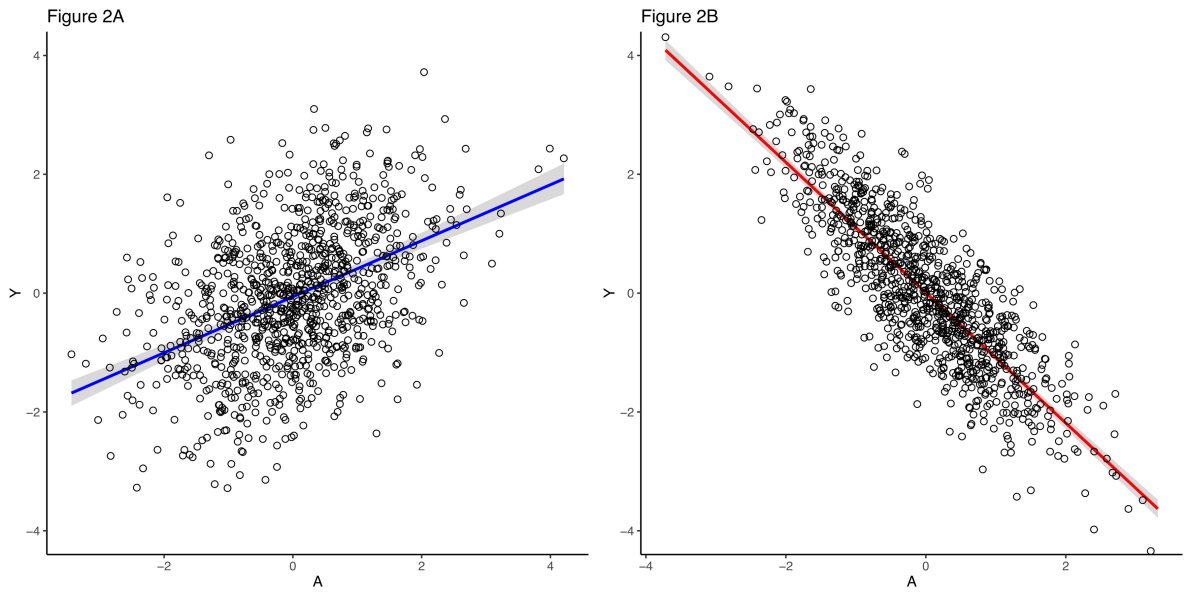

Figure 1.5: Collider bias illustration: A and B are marginally independent but become associated when conditioning on their common effect C.

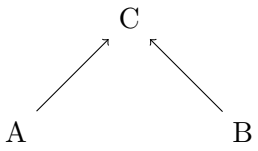

Figure 1.6: Causal diagram to illustrate conditioning on a collider

In Table 1.2, those with C=1 are less likely to have A=1 (n=15) compared to those with C=0 (n=25). In the DAG above, A is associated with C, so the estimate of the causal effect (i.e., odds ratio) from Table 1.2 is 1.47.

Table 1.2

	C=1	C=0	**Total**
A=1	15	25	40
A=0	75	85	160
Total	90	110	200

In Table 1.3, those with C=1 are much less likely to have B=1 (n=30) compared to those with C=0 (n=70). In the DAG above, B is associated with C, so the estimate of the causal effect (i.e., odds ratio) from Table 1.3 is 3.50.

Table 1.3

	C=1	C=0	**Total**
B=1	30	70	100
B=0	60	40	100
Total	90	110	200

A marginal association between A and B has an odds ratio of 1.00 Table 1.1 This is a marginal association because we do not condition on the collider. The marginal associations between A and C, and B and C, are also explored to assess whether conditioning on the collider could potentially bias the association between A and B. From the above, it is clear that the associations (odds ratios) between A and C (and B and C) are very different from 1.00. Suggesting that conditioning on the collider (C) will induce a bias. To illustrate this, we now condition on the collider. If the collider did not induce bias, we should expect an odds ratio of the association between A and B to be 1.00 for both values of C (i.e., where C is 0 or 1).

Conditional associations

Table Table 1.1 shows the tabulation of the association between A and B within levels of C. The association between A and B, conditional on C = 1, has an odds ratio (OR) of:

$$\text{OR}_{C=1} = \frac{(21/9)}{(54/6)} = 0.26$$

The association between A and B, conditional on C = 0, has an odds ratio of:

$$\text{OR}_{C=0} = \frac{(59/11)}{(26/14)} = 2.89$$

: Collider data {#tbl-ColliderABC}

	C=1: B=1	C=1: B=0	C=0: B=1	C=0: B=0	Total
A=1	9	6	11	14	40
A=0	21	54	59	26	160
Total	30	60	70	40	200

Thus, since each of the odds ratios for the association between A and B within levels of C do not equate to 1.00, conditioning on the collider will induce bias in the association between A and B. Conditioning on a descendent (or child) of a collider induces the same problem as conditioning on the collider itself. However, conditioning on an ancestor (or parent) of a collider does not induce bias. This is because the information about the collider that is contained in the ancestor is independent of A and B.

1.4 Counterfactual (potential outcomes) framework

We first introduce the language of the *Potential Outcomes Framework* also known as (a.k.a) Neyman-Rubin Potential Outcomes framework.(Rubin, 2007) To illustrate the framework we use an empirical example based on intensive care medicine.(Connors et al., 1996) The study, set in intensive care units of five United States teaching hospitals between 1989 and 1994, evaluated the effectiveness of right heart catheterisation (RHC) on short-term mortality (30 days) of 5,735 critically ill adult patients (2,184 received a RHC and 3,551 did not received it) receiving care for 1 of 9 prespecified disease categories. In this illustration, let Y, the outcome, denote the vital status of the patient in an intensive care unit (ICU) at 30 days after admission. Let A denote the exposure variable for whether or not the patient received RHC during their stay at the ICU. Let ($\boldsymbol{W}$) include the set of confounders, with C denoting a binary confounder.

In this RHC study, each patient has two potentially observed outcomes (i.e., Y^a), where the first is $Y(1)$if they received RHC, and the second is$Y(0)$if they did not receive RHC (Rubin, 1974a). We say "potentially observed" because only one of these two outcomes can ever be observed since each patient only receives one of the treatments. As an example from Table 1.4, Patient 1 has two potential outcomes: firstly,$Y(0) = 1$says that if this patient did not receive RHC then they would have died within 30 days, and secondly,$Y(1) = 0$ says that if they had received RHC then they would not have died within 30 days. l,

Table 1.4: Potential outcomes framework: C = Binary confounder, A = Binary treatment, Y = Binary outcome, Y(0) = Potential outcome for A=0, Y(1) = Potential outcome for A=1

Patient	Y	A	C	$Y(0)$	$Y(1)$
1	1	0	0	1	0
2	1	1	1	1	1
3	1	1	1	0	1
4	0	1	0	0	0
5	1	0	1	1	1
6	0	1	1	0	0
7	1	0	0	1	1
8	0	1	1	0	0
9	1	1	0	1	1

A common estimand in causal inference is the average treatment effect (ATE). The ATE is a function of the underlying distribution of the counterfactual outcomes, which can be estimated non-parametrically or parametrically.(J. Robins, 1986) The ATE is defined by an average of the difference of two random variables (i.e., the potential outcomes Y^1andY^0). (Gutman & Rubin, 2015; Rubin, 2007) The ATE in the example above can be estimated as the contrast

between the *potential outcomes* under different treatment levels (i.e., the difference between $E[Y^1] - E[Y^0]$).(Rubin, 2005)

Potential outcomes are so named because they are outcomes that could potentially be observed had the patient been assigned $A = a$. However, in observational studies, only one outcome is observed for each individual. To estimate causal effects, we must make certain assumptions to identify potential outcomes from the observed data, then estimate the estimand.(J. Robins, 1986) The necessary assumptions are outlined in the next section.

1.5 Identifiability assumptions

Conditional exchangeability and identification

The first assumption is *conditional exchangeability*. In randomised studies, conditional (and marginal) exchangeability holds because the treated individuals, had they not been treated, would have had the same average potential outcomes as the untreated, and vice versa. This cannot be guaranteed in observational studies, but it can be assumed to hold if the unmeasured risk factors of the outcome are equally distributed between the treated and the untreated groups, conditional on the measured confounders. Thus, using the language of the potential outcomes, the conditional exchangeability assumption (a.k.a conditional independence, unconfoundedness or ignorability) is:

$$Y^a \amalg A \mid \mathbf{W} \; \forall \; a \in \{0, 1\},$$

This states that, conditional on the set of observed confounders $\mathbf{W}$, the actual exposure level $A$is independent of each of the potential outcomes. Thus, the conditional mean independence is given$E[Y^a \mid A = 1, \mathbf{W} = w] = E[Y^a \mid A = 0, \mathbf{W} = w] = E[Y^a \mid \mathbf{W} = w] \; \forall \; a \in 0, 1.$

Positivity

Positivity holds if the conditional probability of being treated or exposed (and similarly for being untreated) is greater than zero. Therefore, if P($\mathbf{W}$=w) > 0, then

$$P(A = a \mid \mathbf{W} = w) > 0 \; \forall \; \mathbf{W} \in \mathbf{w}, a \in \{0, 1\}.$$

When this assumption is violated, it is typically because the target population is poorly defined (i.e, attempting to estimate the effect of a treatment on people who would never receive it).

Counterfactual consistency

Counterfactual consistency holds if the observed outcome for all treated individuals equals their outcome had they been treated, and likewise for untreated individuals. For example, in Table 1.4, Patient 1's observed outcome equals their potential outcome had they not been treated ($Y = Y^a = 1$), Patient 2's observed outcome equals their potential outcome had they been treated ($Y = Y^1 = 1$). The consistency assumption means that the definition of the

treatment, and outcome, is consistent for each patient. Analytically, consistency is represented by:

$$Y = AY^1 + (1 - A)Y^0,$$

Non-interference

Aside from exchangeability, positivity, and consistency, there are other notable assumptions. It is further assumed that there is *no interference*. This is commonly called *Stable Unit Treatment Value Assumption* (SUTVA). This assumption is closely related to the consistency assumption, in that non-interference (or SUTVA) states that there is only one version of the exposure and that a patient's potential outcome is not influenced by the treatment of another patient.(Schomaker, 2020)

1.6 Estimands

Commonly, an observational study aims to answer a scientific question that characterises the effect of an exposure or treatment on an outcome. This question is translated to an *estimand* for which an *estimate* will provide a relevant answer to the exposure-outcome relationship. Statistical methods are the tools used to obtain an estimate from the data. Within the causal framework, these statistical methods are called *estimators*. They are mathematical functions that use the observed values of the observations in the sample (i.e., a function of the random variables) and generate the quantitative value for the estimand. The estimators are represented by algebraic equations that explicitly describe a function of the realised observations.

An estimand is a quantity of interest in a causal analysis that represents the specific causal effect that a researcher aims to estimate. The average treatment effect (ATE) is an example of an estimand. The ATE is defined as the average difference in the outcome variable between a group of individuals who received a treatment and a group of individuals who did not receive the treatment. It represents the average causal effect of the treatment on an outcome of interest in a population.

The ATE is algebraically defined as

$$\text{ATE} \;=\; E[Y = 1 \mid A = 1] - E[Y = 1 \mid A = 0].$$

This is currently in terms of unobserved potential outcomes. For example, we can only observe what happens when an individual is treated or untreated, but not both. To estimate the ATE from our data, we can apply the identifiability assumptions (Section 1.5) and derive the ATE in terms of observable data. Note that $Y^a$is the potential outcome of an individual, with measured confounders$\mathbf{W}$, had they received treatment $A = a$

First, by the law of total probability

$$P[Y^a = 1] \;=\; \sum_w P[Y^a = 1 \,|\, \mathbf{W} = w]\, P(\mathbf{W} = w)$$

By conditional exchangeability

$$P[Y^a = 1] \;=\; \sum_w P[Y^a = 1 \,|\, A = a, \mathbf{W} = w]\, P(\mathbf{W} = w)$$

This is possible since we are assuming that within levels of $\mathbf{W}$, the predictors of the outcome are equally distributed between the treated and non-treated groups. That is, we have achieved what would happen if patients were randomised to each treatment group. If we also assume consistency

$$P[Y^a = 1] \;=\; \sum_w P[Y = 1 \,|\, A = a, \mathbf{W} = w]\, P(W = w)$$

Thus, under these assumptions, the statistical estimand for the ATE is defined as

$$\sum_w P[Y = 1 \,|\, A = 1, \mathbf{W} = w]\, Pr(W = w) \;-\; \sum_w P[Y = 1 \,|\, A = 0, \mathbf{W} = w]\, P(\mathbf{W} = w).$$

Many estimands could be estimated. One common alternative to the ATE is the average treatment effect among the treated (ATT):

$$\text{ATT} \;=\; E[Y^1 \mid A = 1] - E[Y^0 \mid A = 1]$$

Applying the assumptions above, the statistical estimand for the ATT is

$$\sum_w P[Y = 1 \,|\, A = 1, \mathbf{W} = w]\, P(\mathbf{W} = w \mid A = 1) - \sum_w P[Y = 1 \,|\, A = 0, \mathbf{W} = w]\, P(\mathbf{W} = w \mid A = 1).$$

In the above, for both the ATE and ATT, we have transitioned from a setting where we have unobserved potential outcomes to a setting where we can estimate our causal estimand using the distribution of the observed data (Robins, 1999). Conditional exchangeability requires all confounders to be measured and accounted for in the analysis.

There are many different types of estimands that can be used in causal inference, and the choice of estimand depends on the research question and study design. There are many other estimands used for causal analysis. The conditional average treatment effect (CATE) represents the causal effect of a treatment on the outcome for specific subgroups of individuals, rather than the average effect over the entire population. Average causal effect of the untreated (ACEU) represents the average causal effect of not receiving the treatment on the outcome.

1.6.1 Choice of causal estimand

How does one choose a suitable causal estimand? In randomised controlled trials (RCT) there is no choice. The ATE, ATT, and ATU are all equivalent in RCTs because, due to randomisation, the distribution of baseline characteristics will be similar between the treatment groups. However, in quasi-experimental and observational studies these estimands will differ because the distribution of baseline characteristics are likely to differ between treatment groups.

A good starting point when choosing the estimand is to ask "for whom should the treatment effect be estimated?". A causal estimand asks what would happen if the treatment were to be given to, or withheld from, a particular population. (Greifer & Stuart, 2023) provide an in-depth discussion of the choice of estimands, which we paraphrase here.

Average Treatment Effect (ATE)

The ATE asks how the outcome would differ between a setting where the treatment was given to all patients and a setting where the treatment was withheld from all patients. The ATE is useful when two differing treatments could be given to the population but it is unclear which of the two would be more beneficial. The ATE could also be used for assessing whether a policy could be unilaterally implemented, that is applied to some of the population but not all.

Average Treatment Effect in the Treated (ATT)

The ATT asks how the outcome of treated patients would change if they had not received the treatment. The ATT is the effect of withholding the treatment from those who would have received it. This estimand is useful when a decision needs to be made on an intervention that occurs in a particular population. For example, a decision needs to be made on whether to continue a treatment amongst those who are currently being treated, or a decision on whether preventing a harmful exposure would improve health outcomes amongst those who are currently exposed.

Average Treatment Effect in the Untreated (ATU)

The ATU is asks how would the outcome of untreated patients change if they had received the treatment? The ATU is the effect of expanding the treatment to those who do not receive it. This estimand is useful when a decision needs to be made on whether a treatment that is not given to a group of patients should continue to not be given to these patients. For example, consider a treatment that reduces the risk of cancer amongst those who are at high risk, but it is not known whether the treatment reduces the risk of cancer amongst those with low risk (and do not currently receive the treatment). The question is whether one should continue withholding the treatment or whether the low-risk patients would benefit from receiving the treatment.

1.6.2 Choice of statistical estimand

The causal estimand is written in terms of potential outcome notation. It relies on being able to observe two hypothetical potential outcomes and we are not able to observed both of them in the real world. The statistical estimand, however, can be written in terms of the observed data if the identifiability assumptions are plausible.

There are numerous statistical estimands to choose from; some of the more common statistical estimands are:

Causal risk difference

$$\mathrm{P}\{Y^1 = 1\} - \mathrm{P}\{Y^0 = 1\}$$

Causal risk ratio

$$\frac{\mathrm{P}\{Y^1 = 1\}}{\mathrm{P}\{Y^0 = 1\}}$$

Causal log-odds ratio

$$\log\left[\frac{\mathrm{P}\{Y^1 = 1\}}{1 - \mathrm{P}\{Y^1 = 1\}}\right] - \log\left[\frac{\mathrm{P}\{Y^0 = 1\}}{1 - \mathrm{P}\{Y^0 = 1\}}\right]$$

The choice for the measure of association depends on the form of the exposure and the outcome, but also on what might be easier to interpret in the context of the study.

1.6.2.1 A note on estimands, estimators, and estimates

Estimands represent the target of estimation in causal analysis and should not be confused with estimators, which are statistical methods (or algorithms) that are used to estimate the value of an estimand. Estimators are used to compute a numerical estimate of the causal effect of interest based on the available data.

An easy way to distinguish between these is to think of estimands as the name of a cake you would like to bake, think of estimators as the method (or recipe) you will use to make that cake, and the estimate is the cake that comes out of the oven.

There are many different types of estimators used in causal inference, including regression-based methods, propensity score methods, instrumental variables, and machine learning algorithms. The choice of estimator depends on the selected estimand and the study design, and can impact the results of a causal analysis.

1.7 Conclusion

Often, medical research is interested in estimating cause and effect relationships. These relationships are first considered through hypothetical research questions, such as "what would happen if our patients were given a different treatment?", "how effective is this health policy?", or "what would have happened if the patients were not exposed to some harmful substance?". These questions are hypothetical scenarios because we only ever observe what happens when individuals are exposed or unexposed, but not both.

In randomised studies, causal effects can be estimated because individuals are randomly allocated to a treatment. A suitable randomisation process minimises the possibility of confounding. In epidemiological studies, this randomisation process is often unethical or infeasible. For example, if we were to investigate the effect of long-term tobacco smoking on the chances of

developing lung cancer, it would be unethical to assign individuals to smoke tobacco. Instead, we can use statistical methods within a causal framework, whilst making certain assumptions, to estimate the cause-and-effect relationship.

The statistical methods used within the causal framework differ from other statistical methods with respect to their formality. The causal inference roadmap specifies the required criteria to estimate causal effects.(Petersen & Laan, 2014)

In this book we show how a range of causal inference methods can be applied to estimate a cause-and-effect relationship. We emphasise that these methods should only be used after the necessary steps have been satisfied in the causal inference roadmap (i.e., research question, assumptions, data generating distributions, etc.).

As practical epidemiologists and biostatisticians, we focus on the use of applied statistics. Thus, we aim to disseminate computational knowledge for executing and utilizing the various causal inference methods discussed in this book. This text offers practical code examples in R, Stata, and Python for users who wish to explore further.

1.8 Glossary

ATE Average Treatment Effect
ATT Average Treatment Effect among the treated
ATU Average Treatment Effect among the non treated
DAG Directed Acyclic Graph
RCT Randomized Controlled Trial
TMLE Targeted Maximum Likelihood Estimation $

2 Regression adjustment

Regression adjustment is a powerful statistical tool that allows one to control for confounding in complex settings where other methods, such as matching or stratification, do not work. This chapter will introduce the concept of confounding and the regression adjustment method to control for it.

Note: This chapter contains a lot of content from this paper.(Luque-Fernandez et al., 2019a)

2.1 Regression methods

Confounding bias in epidemiological studies occurs when there are shared causes for both treatment or exposure, henceforth (A) and outcome, henceforth (Y) that can fully or partially explain the observed association between A and Y. To control for confounding randomization is classically used in experimental settings before the study design. However, when these tools fail or in observational studies it is critical to address confounding during the analysis stage. Classically, stratification serves as a well-recognized analytical technique to manage confounding. This approach involves evaluating the association of interest within separate groups that display similar characteristics with respect to the confounders. The principle of stratification is simple: It removes the variability of confounding elements within each group, ensuring these do not impact the treatment outcome relationship.

However, despite being highly effective, stratification's application becomes limited when faced with numerous confounding variables, as it results in overly small groupings that hinder practical comparisons due to insufficient data for precise estimation (i.e., sparsity due to increasing dimensionality on the data).

Regression methods for adjusting confounding variables incorporate details about interventions and prognostic factors into a regression formula within a modeling context:

$$y_{ij} = \beta_{0j} + \beta_{1j}x_{i1} + \beta_{2j}x_{i2} + \cdots + \beta_{pj}x_{ip} + \epsilon_{ij},$$

where:

- y_{ij}: represents the i-the observation of the j-th dependent variable.
- β_{0j}: is the intercept for the j-th dependent variable.
- β_{kj}: is the regression coefficient for the k-th predictor variable (xk) on the j-th dependent variable.

- x_{ik}: represents the i-th observation of the k-th predictor variable.
- p: This is the total number of predictor variables.
- ϵ_{ij}: represents the error term for the i-th observation of the j-th dependent variable.

Regression models are often fit within the generalized linear modeling framework. In a generalized linear model (GLM), the **link** function describes the relationship between Y, and A, adjusted for the set of confounders. It converts probabilities of Y into a continuous measure, enabling the modeling of relationships within a linear regression framework, offering all the advantages of including the outcome predictors in the model as quantitative variables without categorization, and the ability to assess trends with ordinal-scale confounders.

Common link functions include: i) the Identity link for linear regression models with continuous, normally distributed responses; ii) the Logit link for binary outcomes, like disease presence; iii) the Probit link, transforming binary responses to a standard normal distribution; iv) the Log link for positively skewed responses, such as counts or proportions; and v) the Inverse link for continuous, positively skewed responses. For binary outcomes, to assess the model coefficient for the adjusted A effect, the odds ratio [OR] is used as a useful estimand. Other estimands that regression models can also calculate are the risk ratio [RR] and the absolute risk or risk difference [RD].

The interpretation of the regression coefficients is straightforward. A regression coefficient indicates how the outcome changes with a change in the treatment or exposure of interest, holding the other model predictors constant. This is a conditional estimate effect. It is important to highlight that in simple linear models, this is equivalent to the marginal effect, i.e., the overall treatment effect. However, when interaction terms are added between A and some confounders, models become more complex. In such cases, the coefficient for A might only represent its effect within a reference group, i.e., a conditional effect and not the overall effect. When moving away from linear models, the connection between conditional effects and overall marginal effects can deteriorate considerably. It might seem natural to assume that dividing a population into various subgroups and then estimating the effect within each subgroup would produce an overall marginal effect that is simply a weighted combination of these subgroup-specific effects. However, this relationship holds only for certain causal measures—specifically those that are collapsible, such as RD and RR. In contrast, ORs exhibit non-collapsibility (see Chapter 3 for more details on collapsibility), which means that the marginal odds ratio might be either larger or smaller than any of the conditional ORs derived from subgroups in the regression model.(Brian W Whitcomb, 2021)

2.2 Confounding and collider biases

A confounder can fully or partially explain the observed association between exposure (A) and outcome (Y). This bias i.e., “confounding bias”, makes for example the effect measure from a binary treatment (A) on the binary outcome (Y) i.e., the raw OR, to diverge from its true

causal effect, i.e., the true marginal causal OR. The expression "association is not causation" illustrates clearly the importance of accounting for confounding bias.

Regression models can be a viable approach for confounding adjustment; however, it requires assuming no effect modification. Alternatively, the generalization of standardization, via the **G-Formula**(Robins, 1986), could be used to improve adjustment, minimize residual confounding, and allow for causal interpretation without randomization (see Chapter four). Moreover, in certain cases, the introduction of a specific type of confounder known as a "collider" into a regression model can result in bias in the regression coefficient estimates for the treatment effect (A), even though it could potentially enhance the model's overall goodness of fit.

Direct Acyclic Graphs (DAGs), which are informed by expertise in the subject area, play a crucial role in identifying such colliders. Identifying whether a confounder is a collider requires careful consideration of the true unobserved data-generating process and the interrelationships among variables within a given context.(N. Pearce, 2014)

In general, incorporating a collider into a regression model is discouraged if the goal is to estimate causal effects, as it can introduce bias. However, if the model is aimed at prediction, including colliders might be beneficial if it decreases prediction error.

To illustrate both concepts, we use a linear regression modeling framework to adjust for confounding in a set of boxes containing R software commented code. We introduce three scenarios to illustrate the differences in adjusting for confounders or colliders Figure 2.1. In Figure 2.1 (panel A), the causal effect of A on Y is confounded by W. In Figure 2.1 (panel B), the causal effect of A on Y is not confounded but adjusting for the collider C induces a bias. Lastly, in Figure 2.1 (panel C), there are no confounders but conditioning on the collider C opens a back-door path through $W1$ and $W2$.

Data consistent with the directed acyclic graph (DAG) depicted in Figure 2.1 was generated (Box 2.1), following a process akin to that described by (Luque-Fernandez, Schomaker, et al., 2018): The confounder, represented by W in Figure 2.1, was simulated as a standard normal random variable, characterized by a mean of 0 ($\mu = 0$) and a variance of 1 ($\sigma^2 = 1$). The generation of the exposure, A, was contingent upon the value of W, incorporating an error term with a standard normal distribution. Subsequently, the outcome, Y, was generated as a function of both A and W, with an additional error term also drawn from a standard normal distribution. These assumptions establish linear relationships between the variables and a simulated causal effect of A on Y with a coefficient of 0.3. Linear regression models, both unadjusted (fit1) and adjusted for W (fit2), were then employed to estimate associations between A and Y. **Box 2.1 (R):** Generate data consistent with Figure 2.1 (panel A)

```r
library(visreg) # load package to visualize regression output
library(ggplot2)# load package to visualize regression output
library(patchwork) # https://patchwork.data-imaginist.com/

N <- 1000 # sample size
```

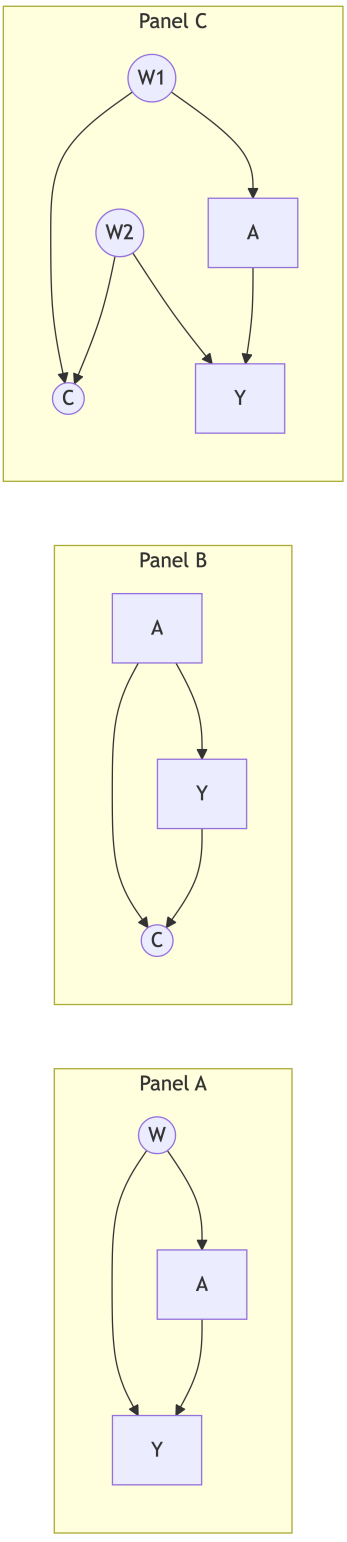

Figure 2.1: DAGs showing confounding (Panel A), collider bias (Panel B), and M-bias (Panel C)

```r
set.seed(777)
W <- rnorm(N) # confounder
A <- 0.5 * W + rnorm(N) # exposure
Y <- 0.3 * A + 0.4 * W + rnorm(N) # outcome
fit1 <- lm(Y ~ A) # crude model
fit2 <- lm(Y ~ A + W) # adjusted model

# visualize crude and adjusted models
a = visreg (fit1 , "A" , gg = TRUE , line = list ( col = "blue") ,
    points = list ( size = 2 , pch = 1 , col = "black") ) + theme_classic ()
b = visreg (fit2 , "A" , gg = TRUE , line = list ( col = "blue") ,
    points = list (size = 2 , pch = 1 , col = "black") ) + theme_classic ()
patchwork <- a + b + plot_layout (guides = "collect")
patchwork + plot_annotation(
    tag_levels = 'A',
    title = 'Regression adjustment: Confounding bias',
    subtitle = 'A: lm (Y ~ A); B: lm (Y ~ A + W)',
    caption = 'Disclaimer: In-house'
)
```

Box 2.1 (Stata): Stata code for data generation and confounding adjustment

```
* Stata code: Generate data consistent with DAG panel A
clear all
set seed 777
set obs 1000

* Generate confounder, exposure, and outcome
generate W = rnormal()
generate A = 0.5 * W + rnormal()
generate Y = 0.3 * A + 0.4 * W + rnormal()

* Crude model (unadjusted for confounder)
regress Y A

* Adjusted model (controlling for confounder W)
regress Y A W

* Visualize using marginal predictions
quietly regress Y A
margins, at(A = (-3(0.5)3))
marginsplot, name(crude, replace) title("Crude: Y ~ A") ///
```

```
    ytitle("Linear prediction") xtitle("A")

quietly regress Y A W
margins, at(A = (-3(0.5)3))
marginsplot, name(adjusted, replace) title("Adjusted: Y ~ A + W") ///
    ytitle("Linear prediction") xtitle("A")

* Combine graphs
graph combine crude adjusted, name(combined, replace) ///
    title("Regression adjustment: Confounding bias")
```

The first regression analysis, without conditioning on W, illustrates this bias. The estimated coefficient for A (0.472) exhibits an upward bias compared to the true causal effect (0.3) specified in the simulation. In contrast, the second regression adds W as a covariate, effectively closing the open back-door path. This approach yields a more accurate estimate of the causal effect (0.289), closer to the true value. The remaining residual difference of 0.011 can be attributed to sampling variability.

Figure 2.2 shows the confounding bias based on the slope from the linear adjustment contrasting A without adjustment for W versus B with adjustment for W.

Figure 2.1 highlights the key role of W as a confounder in this causal structure. Its unique position without parent nodes indicates that it is not influenced by any other variable in the DAG. Consequently, W is generated independently within the model. However, both A and Y share a common parent in W, creating an open back-door path between them. This path explains the potential for confounding bias.

In contrast to Figure 2.1 (panel A), where the causal arrows originate from node W, Figure 2.1 (panel B) presents a different causal structure with arrows directed toward node C from both A and Y. Conditioning on C in this scenario, through methods such as regression or stratification, introduces collider bias. This arises because node C acts as a collider on the path $A \rightarrow C \leftarrow Y$, where two causal pathways converge.

To illustrate this concept, consider a scenario where the wetness of the ground (C) is solely influenced by rain (A) and an automatic sprinkler (Y) set on a timer. In this case, knowing that the ground is wet (conditioning on C) while simultaneously observing that it did not rain (negating path $A \rightarrow C$) implies that the sprinkler must be on (positive effect on C via path $Y \rightarrow C$). Failing to account for the collider C in the analysis might lead to the erroneous conclusion that rain negatively influences sprinkler usage, despite pre-existing knowledge of their independence.(Pearl, 2009)

Violation of the ignorability assumption due to collider conditioning: conditioning on a collider variable, such as C in Figure 2.1 (panel B), can induce a spurious association between the exposure (A) and the potential outcomes ($Y(a)$). This undermines the conditional ignorability

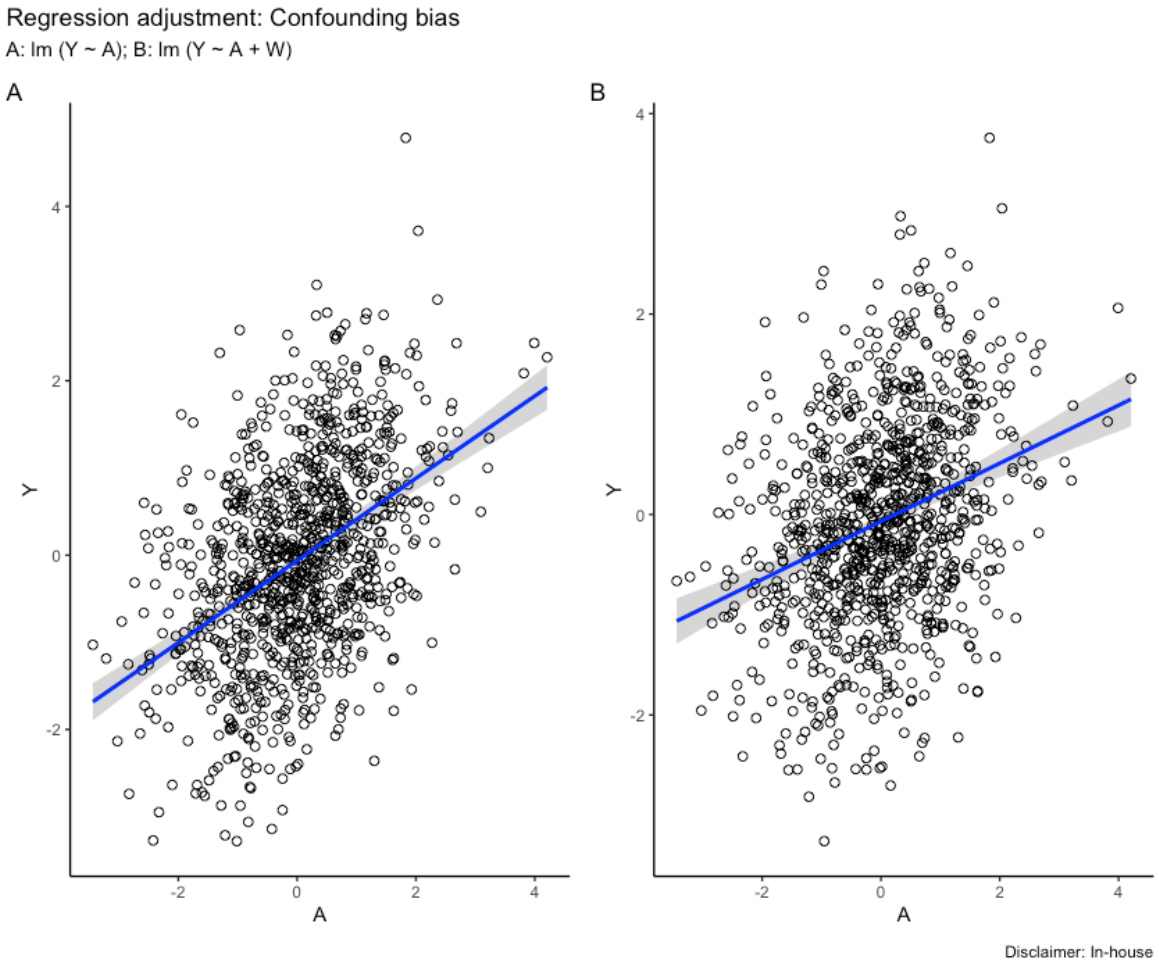

Figure 2.2: Regression adjustment: univariate (A), bivariate adjustement (B) models fit for the linear association between Y and A.

assumption $(Y(a) \perp A \mid W, C)$, which requires the exposure to be independent of the potential outcomes given the conditioning set (see Chapter One).

Figure 2.1 (panel B) illustrates this concept. Conditioning on C opens the back-door path $(A \rightarrow C \leftarrow Y)$ previously blocked by the collider itself. This opens a channel for the effect of A on Y to indirectly influence the observed association between A and $Y(a)$. As a result, the observed association becomes a mixture of both the causal effect of A on Y and the spurious association induced by the back-door path. This confounds the interpretation of the association, as it is no longer solely attributable to the direct causal effect of A on Y. Therefore, conditioning on a collider can lead to spurious associations, hindering the identification of true causal relationships from observational data.

Figure 2.1 (panel C) gives another, more complex collider structure usually known as M-bias, in which the collider (C) is the effect of a common cause $(W1)$ of the exposure (A) and a common cause $(W2)$ of the outcome (Y). There is only one back-door path, and it is already blocked by the collider (C); thus we do not need to control for anything. This is the difference between confounders and colliders: a path will be open if one does not adjust for confounders but blocked if adjustment is made. For colliders, it is the other way around. However, some could consider C to be a classical confounder as it is associated with both A, via $(A \leftarrow W1 \rightarrow C)$, and with Y, via a path that does not go through A $(C \leftarrow W2 \rightarrow Y)$, and it is not in the causal pathway between A and Y. However, controlling for C will introduce a collider bias. If one were to use the traditional characteristics used to identify confounders (i.e., a third variable $[W]$ associated with both the exposure $[A]$ and the outcome $[Y]$ that is not on the causal pathway between A and Y), then one could confuse a collider with a confounder.

Figure 2.1 (panel C) presents a more intricate collider structure exhibiting M-bias, where the collider, C, is influenced by both $W1$, a common cause with exposure A, and $W2$, a common cause with outcome Y. Notably, a single back-door path exists, but collider C already blocks it, eliminating the need for adjustment. This highlights a fundamental distinction between confounders and colliders: while adjusting for confounders reveals potential causal effects by unblocking paths, adjusting for colliders like C inadvertently introduces bias by opening blocked paths, leading to the M-bias phenomenon.

However, some may misinterpret C as a traditional confounder. It is indeed associated with both A (via $W1$) and Y (via $W2$), and it lies outside the direct causal pathway between A and Y. Despite these characteristics, controlling for C would induce collider bias, highlighting the critical distinction between association and causation. Therefore, relying solely on the conventional properties of confounders (association with both exposure and outcome, off-pathway location) can lead to misidentification and spurious inferences when dealing with colliders like C. This underscores the importance of carefully considering the underlying causal structure and potential collider bias before implementing adjustment strategies.

Building upon the simulated scenario depicted in Figure 2.1 (panel B), we replicate the data generation process through a simplified linear mechanism outlined in Box 2.2. Initially, we

draw variable A from a standard normal distribution conditioned on the confounder W. Subsequently, we generate outcome Y as the sum of the effect of A, the confounder W, and an error term. Similarly, variable C is generated as a function of both A and Y, incorporating additional error. This revised configuration, as illustrated in Figure 2.1 (panel B), establishes both A (exposure) and Y (outcome) as parents of C (collider), creating a common effect situation. We then proceed to analyze the data by fitting two models: an adjusted model for W excluding the collider (fit3) and a model that incorporates the collider (fit4), also referred to as the collider model). Notably, the true causal coefficient of exposure A is established as 0.3, with the collider C demonstrating coefficients of 1.0 for both its association with exposure A and outcome Y. **Box 2.2 (R):** Generate data consistent with Figure 2.1 (panel B)

```
N <- 1000 # sample size
set.seed (777)
W <- rnorm(N) # confounder
A <- 0.5 * W + rnorm(N) # exposure
Y <-  0.3 * A + + 0.4 * W + rnorm(N) # outcome
C <-  1 * A + 1 * Y + rnorm(N) # collider
fit3 <- lm (Y ~ A + W) # adjusted model for confounder
fit4 <- lm (Y ~ A + C) # adjusted model for collider

# visualize adjusted models
g1 <- visreg (fit3 , "A" , gg = TRUE , line = list ( col = "blue") ,
            points = list (size = 2 , pch = 1 , col = "black") ) + theme_classic () +
    coord_cartesian (ylim = c ( -4 , 4))
g2 <- visreg (fit4 , "A" , gg = TRUE , line = list ( col = "red") ,
            points = list (size = 2 , pch = 1 , col = "black") ) + theme_classic () +
    coord_cartesian (ylim = c ( -4 , 4))
patchwork <- g1 + g2 + plot_layout (guides = "collect")
patchwork + plot_annotation(
    tag_levels = 'A',
    title = 'Regression adjustment: Collider bias',
    subtitle = 'A: lm (Y ~ A + W); B: lm (Y ~ A + C)',
    caption = 'Disclaimer: In-house')
```

Box 2.2 (Stata): Stata code for data generation and collider bias illustration

```
* Stata code: Generate data consistent with DAG panel B (collider)
clear all
set seed 777
set obs 1000

generate W = rnormal()
```

```
generate A = 0.5 * W + rnormal()
generate Y = 0.3 * A + 0.4 * W + rnormal()
generate C = 1 * A + 1 * Y + rnormal()   /* C is a collider */

* Model adjusted for confounder W (correct estimate)
regress Y A W

* Model adjusted for collider C (biased estimate)
regress Y A C

* Visualize adjusted predictions
quietly regress Y A W
margins, at(A = (-3(0.5)3))
marginsplot, name(g1, replace) title("Adjusted for W: Y ~ A + W") ///
    ytitle("Linear prediction") xtitle("A") ylabel(-4(2)4)

quietly regress Y A C
margins, at(A = (-3(0.5)3))
marginsplot, name(g2, replace) title("Adjusted for C (collider): Y ~ A + C") ///
    ytitle("Linear prediction") xtitle("A") ylabel(-4(2)4)

* Combine graphs
graph combine g1 g2, name(combined, replace) ///
    title("Regression adjustment: Collider bias")
```

In contrast to the previous section, ignoring the collider variable C in the regression model resulted in an estimate of the true coefficient for A (0.3) that was remarkably close, at 0.298. However, adjusting for C introduced a substantial bias, leading to an estimate of -0.3 as depicted in Figure 2.3. While the model incorporating the collider (fit4) performs competitively from a predictive standpoint, as evidenced by its lower Akaike Information Criterion (AIC), it paradoxically alters the direction of the association between A and Y. This phenomenon, where conditioning on the collider introduces bias while ignoring it does not, arises when both A and Y are positively correlated with the collider. Thus, in this specific case, including the collider in the regression model introduces a bias while excluding it does not.

2.3 Motivating example

To illustrate the impact of conditioning on a collider variable, we simulated a dataset with 1,000 observations focusing on the relationship between dietary sodium intake, age, and systolic blood pressure (SBP). The example is fully available and reproducible at the GitHub repository:

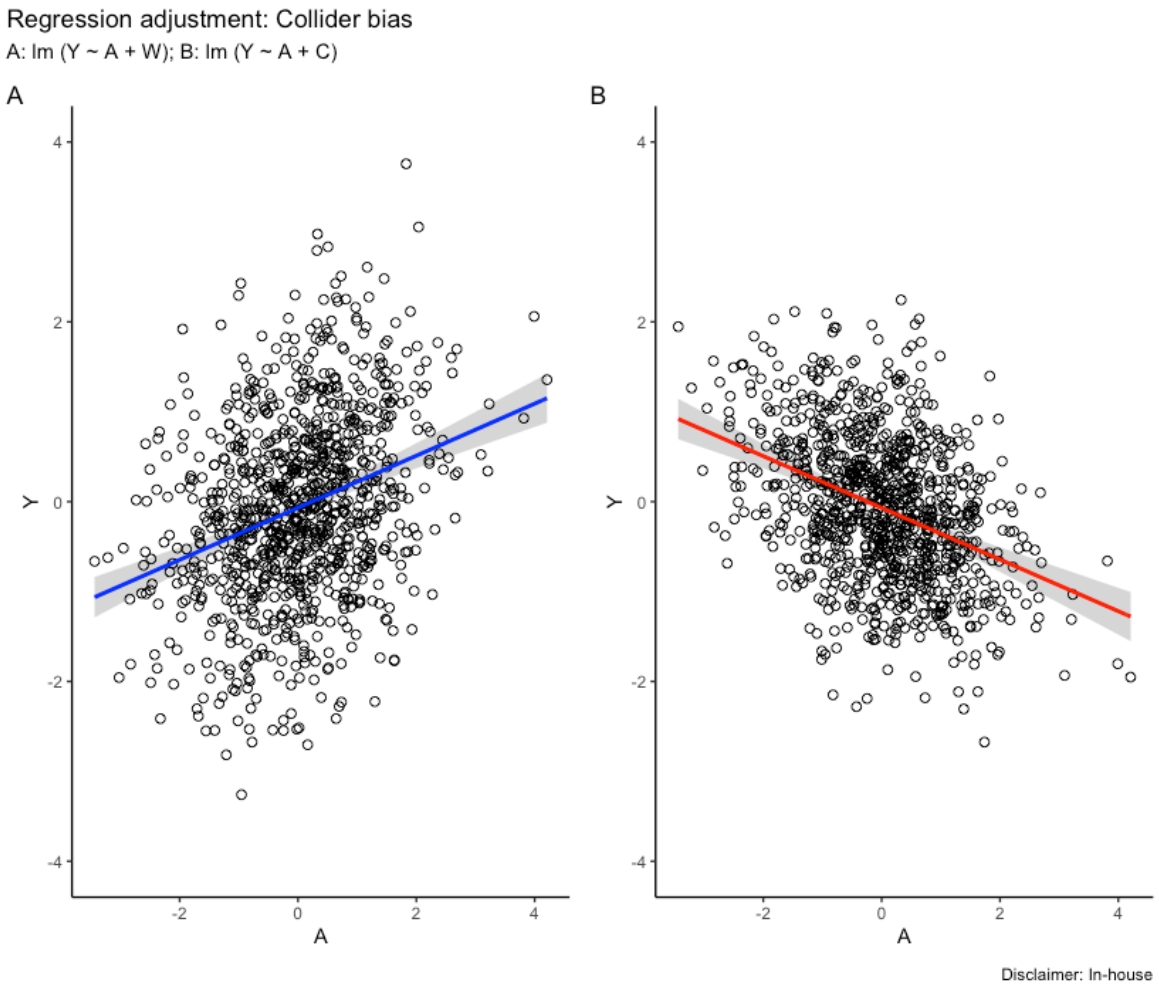

Figure 2.3: Regression adjustment: confounding bias (A), collider bias (B) models fit for the linear association between Y and A.

https://github.com/migariane/ColliderApp, and there is also an online available ShinyApp at: https://watzile.shinyapps.io/EpiCollider/

Hypertension affects nearly one-third of the American population, with over half exhibiting uncontrolled hypertension. Extensive evidence confirms a positive association between cumulative daily sodium intake (grams) exceeding recommended levels and elevated SBP (mmHg). Moreover, advancing age brings about anatomical and physiological changes in the kidneys, compromising their ability to regulate extracellular fluid volume and composition. Notably, these changes include diminished glomerular filtration rate and an impaired capacity to maintain water and sodium homeostasis in response to external factors. Additionally, age-related structural modifications in the arteries contribute to the observed association between age and SBP.

The strong association between age and both high SBP and impaired sodium homeostasis poses a challenge for assessing the true relationship between sodium intake (SOD) and SBP. Age acts as a potential confounder in this scenario, residing on the causal pathway between SOD and SBP as illustrated in Figure 2.4. This implies that controlling for age solely based on its association with both the exposure (SOD) and the outcome (SBP) could introduce bias into the estimated effect of SOD on SBP.

Further complicating the analysis is the role of proteinuria (PRO). High levels of 24-hour urinary protein excretion are observed in response to both sustained high SBP and increased dietary sodium intake, as shown in Figure 2.4. This places proteinuria in the position of a potential collider variable. Controlling for proteinuria in the presence of unmeasured common causes with both SOD and SBP (represented by the arrow converging on PRO in Figure 2.4) could introduce collider bias, potentially underestimating or overestimating the true effect of SOD on SBP. Therefore, researchers conducting such analyses should carefully consider the potential for both confounding and collider bias. Controlling for age remains crucial when its influence on SBP pathways is understood and adequately represented in the model. However, if the underlying physiological mechanisms are incompletely understood, or if proteinuria is mistakenly conceptualized as a confounder, controlling for it could lead to biased estimates.

This section focuses on simulating data to illustrate the paradoxical effect of 24-hour dietary sodium intake (grams) on systolic blood pressure (SBP) after conditioning on a potential collider, urinary proteinuria. The simulated data will be based on the structural relationships depicted in the DAG presented in Figure 2.4 (see Box 2.3 for details).

Box 2.3 provides a function to simulate data for this example. The true causal effect of sodium intake on systolic blood pressure (SBP) is represented by a beta coefficient of 1.05, as shown in the formula for SBP: systolic blood pressure $= \beta_1 \cdot sodium + \beta_2 \cdot age + \epsilon$, where $\beta_1 = 1.05$, $\beta_2 = 2.0$, and ϵ denotes a standard normally distributed error term. Additionally, the coefficients for the association of proteinuria (PRO) with SBP and sodium intake are 2.0 and 2.8, respectively, as specified in the formula for PRO: PRO $= \beta_1 \cdot SBP + \beta_2 \cdot Sodium + \epsilon$, where $\beta_1 = 2.0$, $\beta_2 = 2.8$, and ϵ represents a standard normally distributed error term. **Box 2.3 (R):** Data generation consistent with Figure 2.4

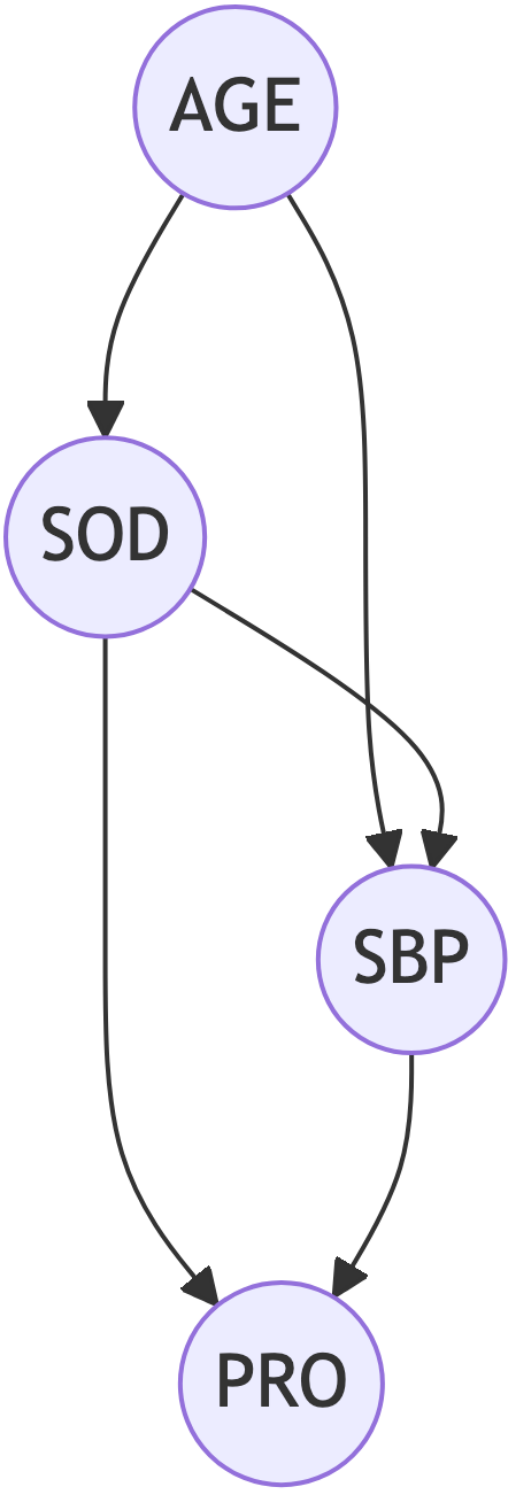

Figure 2.4: DAG for the sodium intake, systolic blood pressure, and proteinuria example

```r
generateData <- function(n, seed){
    set.seed(seed)
    Age_years <- rnorm(n, 65, 5)
    Sodium_gr <- Age_years / 18 + rnorm(n)
    sbp_in_mmHg <- 1.05 * Sodium_gr + 2.00 * Age_years + rnorm(n)
    hypertension <- ifelse(sbp_in_mmHg > 140, 1, 0)
    Proteinuria_in_mg <- 2.00*sbp_in_mmHg + 2.80*Sodium_gr + rnorm(n)
    data.frame(sbp_in_mmHg, hypertension, Sodium_gr, Age_years, Proteinuria_in_mg)
}

ObsData <- generateData(n = 1000, seed = 777)
```

Box 2.3 (Stata): Stata code for data generation of the sodium/SBP example

```
* Stata code: Data generation consistent with DAG for sodium/SBP example
clear all
set seed 777
set obs 1000

* Generate variables based on the DAG structure
generate Age_years = rnormal(65, 5)
generate Sodium_gr = Age_years / 18 + rnormal()
generate sbp_in_mmHg = 1.05 * Sodium_gr + 2.00 * Age_years + rnormal()
generate hypertension = (sbp_in_mmHg > 140)
generate Proteinuria_in_mg = 2.00 * sbp_in_mmHg + 2.80 * Sodium_gr + rnormal()

* Label variables for clarity
label variable Age_years "Age in years"
label variable Sodium_gr "Sodium intake (grams)"
label variable sbp_in_mmHg "Systolic blood pressure (mmHg)"
label variable hypertension "Hypertension indicator (SBP>140)"
label variable Proteinuria_in_mg "Proteinuria (mg)"
```

Three linear regression models were employed (Box 2.4) to explore the association between sodium intake and systolic blood pressure (SBP):

1. Unadjusted model: This basic model examined the crude relationship between sodium intake and SBP.
2. Model adjusted for age: Recognizing age as a potential confounder, this model controlled for its influence on the association.

3. Model adjusted for age and proteinuria: Expanding upon the previous model, proteinuria was included as a potential collider variable due to its possible influence on both sodium intake and SBP.

The algebraic specifications of these models are presented below. Additionally, Box 2.4 provides R code for model fitting and visualization.

Model 1:

$$\text{SBP} \quad = \quad \beta 0 + \beta 1 \cdot Sodium + \epsilon$$

Model 2:

$$\text{SBP} \quad = \quad \beta 0 + \beta 1 \cdot Sodium + \beta 2 \cdot Age + \epsilon$$

Model 3:

$$\text{SBP} \quad = \quad \beta 0 + \beta 1 \cdot Sodium + \beta 2 \cdot Age + \beta 3 \cdot Proteinuria + \epsilon$$

Box 2.4 (R): Linear regression models in R

```r
library(broom) # load packages to visualize regression model's output
library(visreg)

## Models Fit
fit1 <- lm(sbp_in_mmHg ~ Sodium_gr, data = ObsData); tidy(fit0)
fit2 <- lm(sbp_in_mmHg ~ Sodium_gr + Age_years, data = ObsData); tidy(fit2)
fit3 <- lm(sbp_in_mmHg ~ Sodium_gr + Age_years + Proteinuria, data = ObsData); tidy(fit3)

## Models visualization
par(mfrow = c(1, 3))
visreg(fit1, ylab = 'SBP in mmHg', line = list(col = 'blue'),
    points = list(cex = 1.5, pch = 1), jitter = 10, bty = 'n')

visreg(fit2, ylab = 'SBP in mmHg', line = list(col = 'blue'),
    points = list(cex = 1.5, pch = 1), jitter = 10, bty = 'n')

visreg(fit3, ylab = 'SBP in mmHg', line = list(col = 'red'),
    points = list(cex = 1.5, pch = 1), jitter = 10, bty = 'n')
```

Box 2.4 (Stata): Stata code for linear regression models

```
* Stata code: Linear regression models for sodium/SBP example

* Model 1: Unadjusted (crude association)
regress sbp_in_mmHg Sodium_gr
```

```
* Model 2: Adjusted for age (confounder)
regress sbp_in_mmHg Sodium_gr Age_years

* Model 3: Adjusted for age and proteinuria (collider)
regress sbp_in_mmHg Sodium_gr Age_years Proteinuria_in_mg

* Visualize using marginal predictions with confidence intervals
* Model 1
quietly regress sbp_in_mmHg Sodium_gr
margins, at(Sodium_gr = (3(1)8))
marginsplot, name(fit1_stata, replace) title("Model 1: Unadjusted") ///
    ytitle("SBP in mmHg") xtitle("Sodium (grams)")

* Model 2
quietly regress sbp_in_mmHg Sodium_gr Age_years
margins, at(Sodium_gr = (3(1)8))
marginsplot, name(fit2_stata, replace) title("Model 2: Adjusted for Age") ///
    ytitle("SBP in mmHg") xtitle("Sodium (grams)")

* Model 3
quietly regress sbp_in_mmHg Sodium_gr Age_years Proteinuria_in_mg
margins, at(Sodium_gr = (3(1)8))
marginsplot, name(fit3_stata, replace) title("Model 3: + Proteinuria") ///
    ytitle("SBP in mmHg") xtitle("Sodium (grams)")

* Combine plots
graph combine fit1_stata fit2_stata fit3_stata, rows(1) name(combined, replace) ///
    title("Linear regression: Sodium and SBP")
```

To further investigate the association between sodium intake and hypertension, defined as a binary outcome (systolic blood pressure $ 140$mmHg = 1,<$ 140 mmHg = 0), we employed three logistic regression models. The first model served as a baseline, excluding any adjustments. The second model incorporated age as a potential confounder, while the third additionally adjusted for proteinuria, identified as a collider variable due to its potential influence on both sodium intake and hypertension. Notably, these models mirrored the specifications previously described, with the exception of the binary outcome variable (hypertension). Box 2.5 demonstrates the R code for fitting and visualizing these models as a forest plot. **Box 2.5 (R):** Logistic regression models for hypertension outcome

```
## Models fit on multiplicative scale
    library(dplyr)
    library(forestplot)
```

```r
    fit3 <- glm(hypertension ~ Sodium_gr, family=binomial(link='logit'), data=ObsData)
    or <- round(exp(fit3$coef)[2], 3) # conditional odds ratio from logistic model
    ci95 <- exp(confint(fit3))[-1,] # 95% CI of odds ratio
    fit4 <- glm(hypertension ~ Sodium_gr + Age_years, family = binomial(link = 'logit'), data
    or <- round(exp(fit4$coef)[2], 3)
    ci95 <- exp(confint(fit4))[2,]
    fit5 <- glm(hypertension ~ Sodium_gr + Age_years + Proteinuria_in_mg, family = binomial(
    or <- round(exp(fit5$coef)[2], 3)
    ci95 <- exp(confint(fit5))[2,]

## Forest plot (see supplementary material for accessing the complete code)
    fp <- rbind(result1, result2, result3); fp %>% or_graph()
```

Box 2.5 (Stata): Stata code for logistic regression models

```
* Stata code: Logistic regression models for hypertension (multiplicative scale)

* Model 1: Unadjusted
logistic hypertension Sodium_gr
* Display odds ratio with 95% CI
* Alternatively, use logit for coefficients:
* logit hypertension Sodium_gr

* Model 2: Adjusted for age (confounder)
logistic hypertension Sodium_gr Age_years

* Model 3: Adjusted for age and proteinuria (collider)
logistic hypertension Sodium_gr Age_years Proteinuria_in_mg

* To store odds ratios for a forest plot, use estimates store
* Model 1
logistic hypertension Sodium_gr
estimates store m1

* Model 2
logistic hypertension Sodium_gr Age_years
estimates store m2

* Model 3
logistic hypertension Sodium_gr Age_years Proteinuria_in_mg
estimates store m3
```

```
* Forest plot using coefplot (user-written; install with: ssc install coefplot)
* coefplot (m1, drop(_cons) label("Unadjusted")) ///
*         (m2, drop(_cons) label("+ Age")) ///
*         (m3, drop(_cons) label("+ Proteinuria")), ///
*         xline(1) eform title("Forest plot: Sodium and Hypertension") ///
*         xtitle("Odds Ratio (95% CI)")
```

Effect of conditioning on a collider

Figure 2.5 depicts the estimated effect of sodium intake on systolic blood pressure (SBP) after conditioning on a collider variable, along with corresponding 95% confidence intervals. The adjusted regression line represents the predicted SBP conditional on the median value of age (Figure 2.5 panel B) or both age and proteinuria (Figure 2.5 panel C). Notably, unlike the unadjusted and bivariate models in Figure 2.5 (panels A and B, respectively), the collider-adjusted model in Figure 2.5 (panel C) reveals a negative association between sodium intake and SBP. Here, a one-unit increase in sodium intake is associated with a predicted decrease of 0.9 mmHg in SBP.

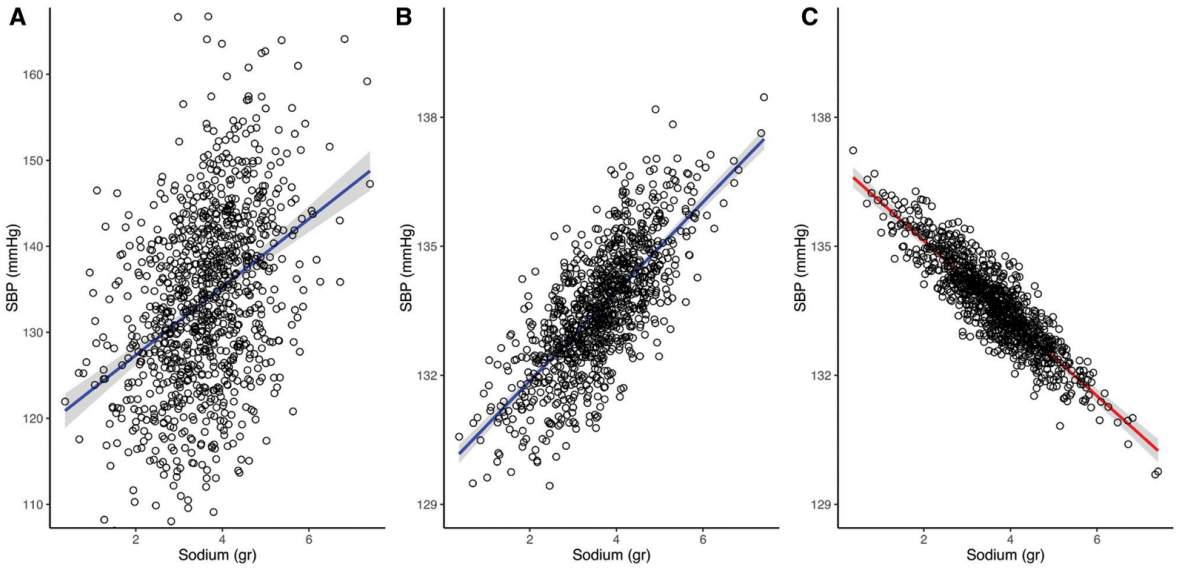

Figure 2.5: Collider effect for the illustration: univariate (A), bivariate (B) and multivariate (C) models fit for the linear association between systolic blood pressure and 24-h sodium dietary intake, adjusted for age acting as a confounder and proteinuria acting as a collider, n=1000.

2.4 Monte-Carlo simulation

The code employed for conducting Monte-Carlo simulations on the additive scale, utilizing the same parameters as outlined in Box 2.3, is presented in Box 2.6. Within the linear model, the simulated causal effect of 24-hour sodium intake on SBP was 1.05 mmHg. The coefficients corresponding to the associations between proteinuria (PRO) and SBP, as well as PRO and sodium intake, were 2.0 and 2.8, respectively. Upon completion of 1000 simulation iterations, the estimated additive effect of 24-hour sodium intake on SBP was determined to be -0.91 mmHg. This indicates a decrease of -0.91 units in SBP for every unit increase in sodium intake. Notably, conditioning on proteinuria, a collider variable, introduced a relative bias of 13.3%. **Box 2.6 (R):** Monte Carlo simulations

```
# Monte Carlo Simulations
    R<-1000
    true <- rep(NA, R)
    collider <- rep(NA, R)
    se <- rep(NA, R)
    set.seed(050472)

    for(r in 1: R) {
    if (r%%10 == 0) cat(paste('This is simulation run number', r, '\n'))

# Function to generate data
    generateData <- function(n){
    Age_years <- rnorm(n, 65, 5)
    Sodium_gr <- Age_years / 18 + rnorm(n)
    sbp_in_mmHg <- 1.05 * Sodium_gr + 2.00 * Age_years + rnorm(n)
    Proteinuria_in_mg <- 2.00 * sbp_in_mmHg + 2.80 * Sodium_gr + rnorm(n)
    data.frame(sbp_in_mmHg, Sodium_gr, Age_years, Proteinuria_in_mg)
    }

    ObsData <- generateData(n=10000)

# True effect
    true[r] <- summary(lm(sbp_in_mmHg ~ Sodium_gr + Age_years, data = ObsData))$coef[2,1]

# Collider effect
    collider[r] <- summary(lm(sbp_in_mmHg ~ Sodium_gr + Age_years + Proteinuria_in_mg, data =
    se[r] <- summary(lm(sbp_in_mmHg ~ Sodium_gr + Age_years + Proteinuria_in_mg, data = ObsDa
    }

# Estimate of sodium true effect
```

```r
    mean(true)

# Estimate of sodium biased effect in the model including the collider
    mean(collider)

# simulated standard error/confidence interval of outcome regression
    lci <- (mean(collider) - 1.96*mean(se)); mean(lci)
    uci <- (mean(collider) + 1.96*mean(se)); mean(uci)

# Bias
    Bias <- (true - abs(collider)); mean(Bias)

# % Bias
    relBias <- ((true - abs(collider)) / true); mean(relBias) * 100

# Plot bias
    plot(relBias)
```

Box 2.6 (Stata): Stata code for Monte Carlo simulations

```stata
* Stata code: Monte Carlo simulations of collider bias
clear all
set seed 050472

* Define a program to generate one simulated dataset and compute estimates
capture program drop sim_collider_sbp
program define sim_collider_sbp, rclass
    syntax [, n(integer 10000)]
    drop _all
    set obs `n'

    * Generate data consistent with DAG
    generate Age_years = rnormal(65, 5)
    generate Sodium_gr = Age_years / 18 + rnormal()
    generate sbp_in_mmHg = 1.05 * Sodium_gr + 2.00 * Age_years + rnormal()
    generate Proteinuria_in_mg = 2.00 * sbp_in_mmHg + 2.80 * Sodium_gr + rnormal()

    * True effect (adjusted for confounder, excluding collider)
    regress sbp_in_mmHg Sodium_gr Age_years
    return scalar true_eff = _b[Sodium_gr]

    * Collider effect (adjusted for confounder AND collider)
```

```
    regress sbp_in_mmHg Sodium_gr Age_years Proteinuria_in_mg
    return scalar collider_eff = _b[Sodium_gr]
    return scalar se = _se[Sodium_gr]
end

* Run 1000 simulations using simulate prefix
simulate true_eff = r(true_eff) collider_eff = r(collider_eff) ///
         se = r(se), reps(1000) seed(050472) saving(sim_results, replace): sim_collider_sbp

* Load simulation results and compute summaries
use sim_results, clear

* Mean estimates
summarize true_eff
local mean_true = r(mean)
summarize collider_eff
local mean_coll = r(mean)
summarize se
local mean_se = r(mean)

display "True effect (mean): `mean_true'"
display "Collider-biased effect (mean): `mean_coll'"
display "Mean standard error: `mean_se'"

* Confidence interval for collider effect
local lci = `mean_coll' - 1.96 * `mean_se'
local uci = `mean_coll' + 1.96 * `mean_se'
display "95% CI for collider effect: (`lci', `uci')"

* Bias and relative bias
generate bias = true_eff - abs(collider_eff)
generate relBias = 100 * (true_eff - abs(collider_eff)) / true_eff
summarize bias relBias

* Histogram of relative bias
histogram relBias, name(bias_hist, replace) ///
    title("Distribution of relative bias (%)") ///
    xtitle("Relative bias (%)") ytitle("Frequency")
```

Within the framework of this data-generating structure, the presence of a collider variable introduces regression dilution bias (also known as regression attenuation), reducing the observed association between sodium intake and SBP compared to the true causal effect. To induce a

paradoxically negative association (i.e., sodium intake protective against SBP), the strength of the collider-exposure association (α_1) and collider-outcome association (α_2) must increase relative to the magnitude of the true causal effect. While equating $\alpha_1$andα_2 may not be entirely realistic, it serves as a useful simplification for illustrating how bias magnitude changes as these associations vary.

Beyond the classic scenario of conditioning on a collider in the analysis, two further situations warrant attention:

1. Collider Effect and Sample Selection: When sample selection is guided by either a measured or unmeasured common effect of the exposure and outcome, collider bias can emerge. Recent studies highlight the potential for even subtle influences on sample selection to induce misleading estimates. Unfortunately, addressing this bias can be challenging, as information on selection mechanisms may be lacking.
2. M-Bias: Balancing Collider and Confounder Control: M-bias arises when the collider also functions as a confounder. In this setting, understanding the trade-off between collider and confounder control becomes crucial. M-bias, while potentially comparable in magnitude to classical confounding, is often negligible unless associations between the collider and confounders are very strong (e.g., relative risk > 8). Therefore, prioritizing confounder control is generally recommended in M-bias situations.

Collider Inclusion in Regressions:

These investigations explored situations where adding a collider variable to a linear regression model improved model fit but introduced bias in coefficient estimates. It is important to note that

- Collider Identification: DAGs, informed by subject-matter knowledge, play a vital role in identifying colliders. This involves critically examining the unobserved data-generation process and variable relationships within the specific research context.
- Prediction vs. Explanation: The decision to include or exclude a collider depends on the study's objective. In studies seeking causal understanding in epidemiology, colliders should generally be excluded to avoid biased effect estimates and confounding through back-door paths. Conversely, for purely predictive purposes, including colliders may be beneficial if it improves model accuracy.

Implications for Epidemiological Research:

Given that most epidemiological research aims to explain causal relationships, awareness of collider variables is crucial to avoid paradoxical associations. By carefully considering collider effects and potential biases, epidemiologists can ensure the robustness and interpretability of their causal estimates.

2.5 Conclusion

Regression adjustment is the most widely used method for controlling confounding in observational studies. Its simplicity and familiarity make it an essential tool, but its validity depends critically on correct model specification. When the outcome model is correctly specified and there is no effect modification by confounders, regression provides unbiased estimates of the average treatment effect. However, when effect modification is present, the regression coefficient does not, in general, equal the marginal ATE — a subtle but important point that motivates the g-formula and other standardization-based methods presented in subsequent chapters.

Key takeaways from this chapter: - Confounding can be addressed by conditioning on measured confounders in a regression model. - The regression coefficient for treatment equals the ATE only under effect homogeneity (no interaction between treatment and confounders). - Model diagnostics — including residual analysis, influence measures, and specification tests — are essential for credible inference. - When the goal is marginal causal effects in the presence of effect modification, more flexible methods such as the g-formula are needed.

2.6 Glossary

AIC Akaike Information Criterion — a measure of model fit penalized by the number of parameters.

DAG Directed Acyclic Graph — a graphical representation of causal relationships among variables.

GLM Generalized Linear Model — a flexible generalization of ordinary linear regression.

OR Odds Ratio — the ratio of the odds of an outcome in the treated group to the odds in the control group.

RD Risk Difference — the absolute difference in risk between treated and control groups.

RR Risk Ratio — the ratio of the risk in the treated group to the risk in the control group.

Part II

G-Methods

3 G-formula

3.1 Introduction

The g-formula, also known as the g-computation formula, is a fundamental method in causal inference used to estimate the causal effect of interventions using observational data. It was introduced by James Robins in the 1980s to address challenges posed by time-dependent confounding in longitudinal studies. Traditional epidemiological methods, such as regression models and stratification-based approaches, often failed to account for time-varying confounders affected by prior treatment, leading to biased estimates. The g-formula emerged as a solution to these limitations, offering a mathematically rigorous way to estimate causal effects under specific assumptions.

Before the g-formula, causal inference heavily relied on approaches such as stratification and regression, which struggled with complex confounding structures, particularly in longitudinal studies where confounders change over time. In response, Robins built upon the potential outcomes framework, originally developed by Donald Rubin and Jerzy Neyman, to introduce the g-methods. These methods include the g-formula, Inverse Probability of Treatment Weighting (IPTW), and G-estimation for Structural Nested Models, all of which aim to handle time-dependent confounding. The introduction of the g-formula represented a significant advancement by enabling researchers to estimate counterfactual outcomes more accurately.

The g-formula gained attention in the 1990s as researchers applied it to public health and epidemiology, particularly in studying long-term interventions such as HIV treatment and smoking cessation programs. As computational power increased in the 2000s, the method became more widely used in complex, high-dimensional datasets. During this period, software implementations in statistical programming languages such as R and Stata made the g-formula more accessible to applied researchers. By the 2010s, further extensions of the method allowed for the consideration of competing risks, dynamic treatment regimes, and personalized medicine applications.

Mathematically, the g-formula estimates the counterfactual mean outcome under a given intervention by standardizing observed data. It is expressed as:

$$E[Y^a] = \sum_w E[Y \mid A = a, W = w]P(W = w)$$

In this formula, Y^arepresents the potential outcome under the intervention$A = a$, $\mathbf{W}$denotes confounders (which may be time-dependent), and$A$ refers to the treatment or exposure of

interest. By iteratively predicting the outcome based on observed covariates, the g-formula helps to adjust for confounding and provides an unbiased estimate of the causal effect under the assumption of no unmeasured confounding, positivity, and consistency.

The g-formula has found applications across multiple disciplines, including epidemiology, public health, and health policy. Researchers have used it to evaluate the long-term effects of healthcare interventions, assess drug effectiveness, and inform clinical decision-making. Despite its wide applicability, the g-formula has some limitations. It requires correct model specification for all variables involved in the causal pathway, making it sensitive to model misspecification. Additionally, the method can be computationally intensive, particularly in large datasets, and relies on the assumption of no unmeasured confounding, which may not always hold in observational studies.

In recent years, modern advancements have extended the g-formula's capabilities. Approaches such as Monte Carlo G-computation allow for the simulation of potential outcomes under various interventions, while the integration of machine learning methods helps to relax parametric assumptions and improve predictive accuracy. Furthermore, combining the g-formula with approaches like Targeted Maximum Likelihood Estimation (TMLE) has enabled more efficient and robust causal effect estimation.

Overall, the g-formula represents a major milestone in the evolution of causal inference, providing researchers with a powerful tool to estimate causal effects in complex observational settings. Its continued development and integration with modern statistical techniques ensure that it remains relevant in tackling pressing questions in public health and medicine.

3.2 Contrast between conditional and marginal estimates

3.3 Effect modification, collapsibility

(HETMOR content)

The American Journal of Public Health's series, "Evaluating Public Health Interventions," provides highly practical advice for public health researchers. Part eight of this series offers a strong introduction to estimating the effects of time-invariant public health interventions. The authors of this particular article propose that modern causal inference methods offer no inherent advantage over traditional multivariable regression modeling when considering bias and efficiency.

However, this assertion doesn't always hold true. Specifically, the concepts of effect modification and collapsibility play crucial roles in determining the appropriateness of using regression for estimating causal effects. Understanding these concepts is essential for validating the use of regression in such analyses.

Cancer epidemiology example

To discuss these concepts, we are looking at an example from cancer epidemiology. In this example, we are interested in the effect of dual treatment therapy (radio- and chemotherapy), compared to single therapy (chemotherapy only) on the probability of one-year survival among colorectal cancer patients. We know that there are confounders which affect both treatment assignment and the outcome, namely clinical stage, socioeconomic status, comorbidities, and age. Evidence shows that older patients with comorbidities have a lower probability of being offered more aggressive treatments and therefore they usually get less effective curative options. Also, colorectal cancer patients from lower socioeconomic status have a higher probability of presenting an advanced clinical stage at diagnosis, thus they usually get offered only palliative treatments.

Our structural assumptions in the cancer epidemiology example

The assumptions from above can be encoded in a directed acyclic graph (DAG) (Figure 1). Here, each circle represents a variable and an arrow from A to B (A -> B) means that we assume that A causes B. The combination of these structural assumptions and appropriate statistical methods allow us to estimate the causal effect of dual therapy versus monotherapy on colorectal cancer patients' survival.

A clinician may be interested in the following question: how different would the risk of death have been had everyone received dual therapy compared to if everyone had experienced monotherapy? The causal marginal odds ratio (MOR) answers this question. Statisticians call this a "target quantity". Each individual has a pair of potential outcomes: the outcome they would have received had they been exposed to dual treatment (A=1), denoted Y(1), and the outcome had they been unexposed, Y(0). The MOR is defined as:

$$MOR = \frac{\frac{P(Y(1)=1)}{(1-P(Y(1)=1))}}{\frac{P(Y(0)=1)}{(1-P(Y(0)=1))}}$$

A common approach would be to use logistic regression to model the odds of mortality given the intervention, and adjust for the confounders (W) which are age (W1), socioeconomic status (W2), clinical stage (W3), and comorbidities (W4). Note that using a logistic regression, it estimates the conditional odds ratio (COR), which is:

$$COR = \frac{\frac{P(Y=1|A=1,W)}{(1-P(Y=1|A=1,W))}}{\frac{P(Y=1|A=0,W)}{(1-P(Y=1|A=0,W))}}$$

MOR and COR are typically not identical. First, if there is effect modification, e.g. if the effect of dual therapy is different for patients with no comorbidities compared to those having hypertension, then logistic regression (possibly including an interaction of treatment with one of the confounders) will not provide a marginal effect estimate, but only the conditional effect of the respective comorbidity (hypertension). To be more precise: we obtain an odds ratio that is valid for a given group of people, say those with hypertension, but it will not give us a marginal estimate. However, we are interested in a marginal estimate because we want to know

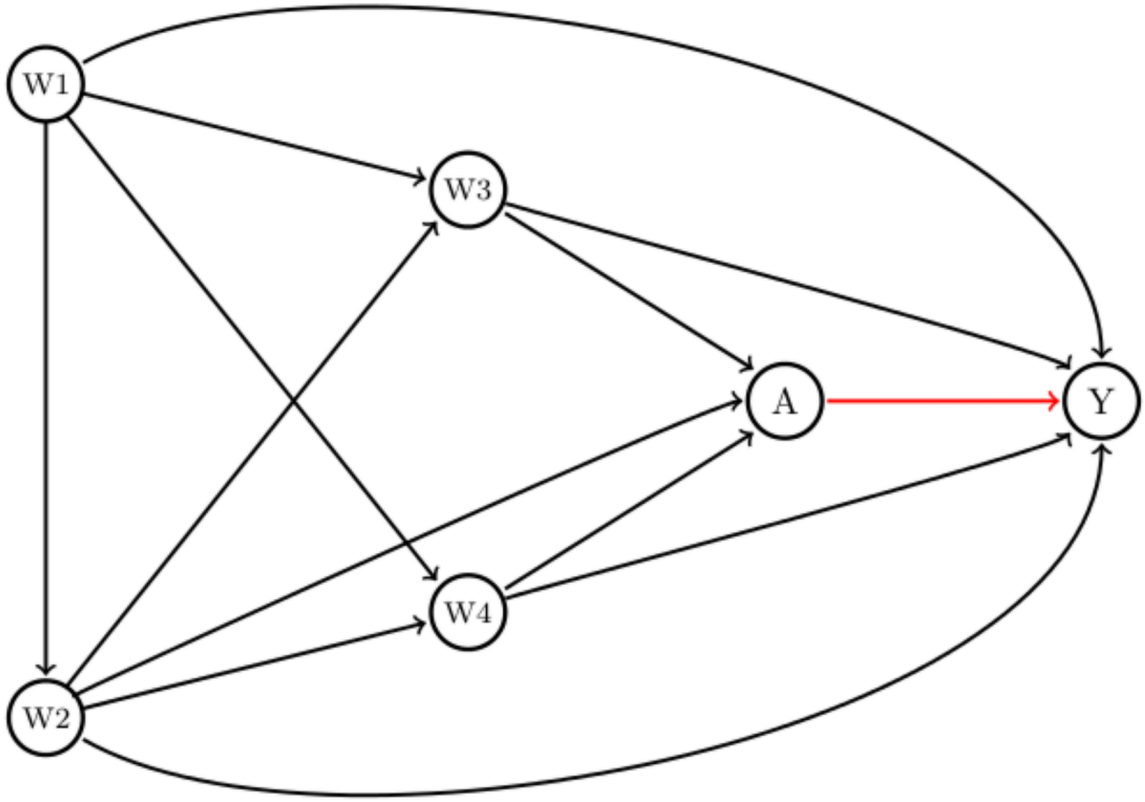

Figure 3.1: Y: mortality binary indicator (1 death, 0 alive). A: binary treatment for cancer with chemotherapy versus dual therapy (0 Chemo, 1 Chemo + Radiotherapy). W: w1 (age), w2 (socioeconomic status), w3 (comorbidities), w4 (cancer stage [TNM classification]).

if the dual therapy works in general. Of course, one may be specifically interested in patients with hypertension, but then the OR for this group is again conditional on the other variables, for example for elderly people, from a low socioeconomic level, and advanced stage.

Second, the odds ratio is non-collapsible which means that the MOR is not necessarily equal to the stratum-specific odds ratio (OR), i.e. the COR. This statement holds even when W is only related to the outcome, and not the intervention, and is thus not a confounder.[2,3] In fact, it is even possible that the conditional odds ratio shows a benefit of the intervention in every stratum, but no benefit overall, i.e. marginally. This case is known as Simpson's paradox. We encourage the reader to read the below references 2 and 3 plus Judea Pearls' new book, The Book of Why for more insights. Please note that while the odds ratio is non-collapsible, other measures of association like the risk-difference and the risk ratios are collapsible.

Multivariable regression versus the G-Formula

To identify the MOR, classical epidemiologic methods, such as standard multivariable logistic regression models, where the treatment is included as a covariate in the analysis, require the assumption that the effect measure of the treatment of interest is constant across the levels of confounders included in the model.[4] However, in observational studies evaluating the effect of public health interventions, this is often not the case (i.e. the effect of the intervention might differ across individuals with different susceptibilities or characteristics). This is essentially the first point we made in the paragraph above. The second point says that certain effect measures, like the odds ratio, suffer from non-collapsibility.

Thus, in summary, as pointed out by Spiegelman et al [1] it can a be an option to use regression models to adjust for confounding; but we need to assume no effect modification and we need to choose a measure that is collapsible, like the risk difference, rather than the OR.

An alternative to using multivariable regression adjustment is the G-Formula [5] (a generalization of standardization with respect to the confounder distribution). In 1986, a seminal paper [5] demonstrated that under assumptions (conditional exchangeability, positivity, consistency, and non-interference, see Appendix below), a consistent estimate of the MOR can be obtained using the G-formula. G-computation,[6] based on the estimation of the components in the G-formula, allows for a treatment effect that may vary across the levels of the confounders. Furthermore, under the assumption that the DAG above (Figure 1) is correct and the other assumptions, we can estimate the MOR using the G-formula as follows:

$$MOR(G-formula) = \frac{\frac{\Sigma_w P(Y=1|A=1,W=w)P(W=w)}{(1-\Sigma_w P(Y=1|A=1,W=w)P(W=w))}}{\frac{\Sigma_w P(Y=1|A=0,W=w)P(W=w)}{(1-\Sigma_w P(Y=1|A=0,W=w)P(W=w))}}$$

where P(W=w) refers to the marginal probability of W.

Monte-Carlo simulation

We implement a Monte Carlo simulation based on the above population-based cancer epidemiology scenario and provide the R code for replication in this GitHub repository:

https://github.com/migariane/HETMOR-Causal-Inference/blob/master/MonteCarloSimulation.r. As noted above, we are interested in how different the odds of death would have been had everyone received dual therapy compared to if everyone had experienced monotherapy. This is a relevant research question that, answered at a population level, may have an important public health implications for cancer patients.

Data generating process

We used the R-package simcausal [7] to generate data according to the DAG introduced above. The data are (W= (W1, W2, W3, W4), A, Y) where W1 refers to age, W2 to socioeconomic status, W3 to comorbidities, and W4 to cancer stage. The detailed setup can be found here (R script for simulation). In the outcome model, we included an interaction term between treatment A and both comorbidities (W2) and cancer stage (W4), based on the plausible biological mechanism that there is an increased risk of comorbidities among older adults and a different treatment effect for those patients with and without comorbidities and advanced cancer stage. The simulation is based on a sample size of 5,000, and 10,000 simulation runs. We estimate the bias with respect to the MOR. Figure 2 shows the results of the above described Monte Carlo simulation.

Briefly, one can see the bias of the multivariable logistic regression model is more pronounced under effect modification but persists - due to non-collapsibility - even under no effect modification.

In order to be able to consistently estimate the MOR, the data must satisfy the following assumptions [8]: i) Cancer treatment is independent of the potential mortality outcomes (Y(0), Y(1)) after conditioning on W. This assumption is often referred to as "conditional exchangeability" and one cannot test it using the observed data. It implies that (within the strata of W) the mortality risk under the potential treatment A=1, i.e. P(Y(1)=1|A=1,W) equals the one under treatment A=0, i.e. P(Y(1)=1|A=0,W). In other words: the risk of death for those treated would have been the same as for those untreated if untreated subjects had received, contrary to the fact, the treatment. This assumption requires that all confounders have been measured. ii) We also assume that within strata of W every patient had a nonzero probability of receiving either of the two treatment conditions, i.e. 0 <P(A=1|W)<1 (positivity). iii) We assume consistency, which states that the counterfactuals equal the observed data under assignment to the treatment actually taken, i.e. for any individual, Y = AY(1) + (1 - A)Y(0). Also, iv) in defining an individual's counterfactual outcome as only a function of their own treatment, we assume non-interference, meaning that the counterfactual outcome of one subject was not influenced by the treatment of any other. If our estimate of the MOR is x (>1) then we can give, for example, an interpretation that says that the chances of one year mortality are x times higher if everyone had received dual treatment compared to if everyone had received single therapy.

In the previous section, we showed that causal estimates provide a generalization of indirect standardisation via the g-computation. In this chapter, we show how to obtain estimates using the g-formula. "Classical" methods are those based on the g-formula using regression methods

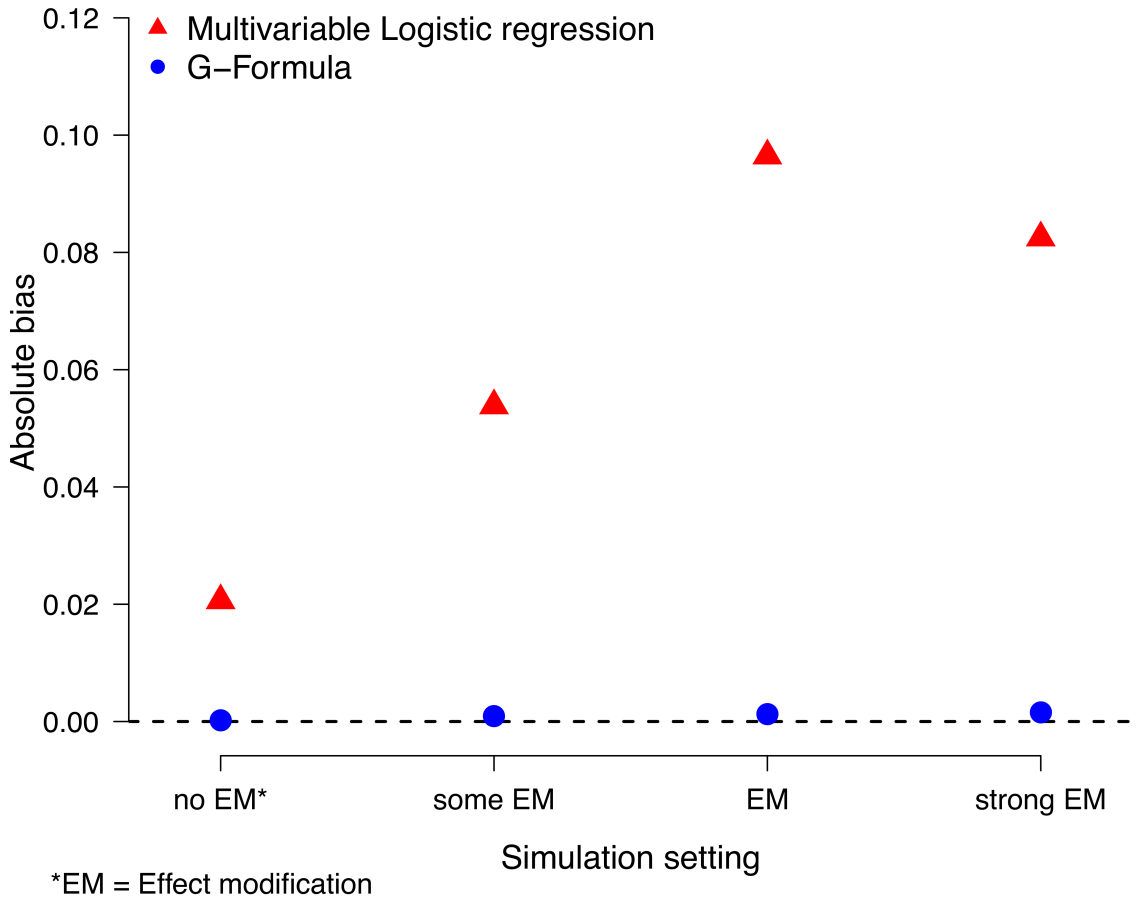

Figure 3.2: Absolute bias with respect the marginal causal odds ratio comparing the conditional odds ratio from classical multivariable logistic regression models versus the marginal odds ratio from G-computation based on the G-Formula, n = 5,000 and 10,000 simulation runs.

and inverse probability weights in contrast to more advanced methods such as double-robust that were developed to overcome the issues with the previous methods.

3.4 Non-parametric g-formula

Regression adjustment is one of the classical methods to control for confounding. As mentioned before, regression models require certain assumptions to obtain the correct model specification. One of these is that the effect of the exposure is constant across levels of a confounder. In other words, we assume there is no effect modification.(Luque-Fernandez et al., 2019b)

The non-parametric g-formula, also known as g-computation, is a method for estimating causal effects by standardizing the outcome distribution over the distribution of confounders. Unlike parametric approaches, it does not impose functional form assumptions on relationships between variables. Instead, it relies on empirical estimates of conditional probabilities. This makes it a flexible approach that can be used in various observational study settings where confounding is a concern. By weighting the treatment-specific outcome probabilities by the distribution of confounders in the overall population, the non-parametric g-formula provides an estimate of the marginal treatment effect.(J. Robins, 1986)

3.4.1 Non-parametric g-formula for one confounder

For a binary treatment $A$and a binary outcome$Y$, given a confounder W, the non-parametric g-formula for the average treatment effect (ATE) is expressed as:

$$ATE = \sum_{w} \left[E(Y \mid A = 1, W = w) - E(Y \mid A = 0, W = w)\right] P(W = w).$$

where

$$E(Y = 1 \mid A = a, \mathbf{W} = w) = \frac{P(\mathbf{W} = w, A = a, Y = 1)}{\sum_y P(\mathbf{W} = w, A = a, Y = y)}$$

is the conditional probability of the outcome $Y = 1$, given the treatment $A = a$, and the set of confounders $\mathbf{W}$ = w.

More generally, for a vector of confounders W, this can be written as:

$$E[Y^a] = \sum_{w} E[Y \mid A = a, W = w] P(W = w),$$

and the ATE is defined as:

$$ATE = E[Y^1] - E[Y^0].$$

The g-formula can also be used to estimate the average treatment effect among the treated (ATT) and among the untreated (ATU) by standardizing over different distributions of W:

$$ATT = \sum_{w} \left[E(Y \mid A = 1, W = w) - E(Y \mid A = 0, W = w)\right] P(W = w \mid A = 1),$$

$$ATU = \sum_{w} \left[E(Y \mid A = 1, W = w) - E(Y \mid A = 0, W = w)\right] P(W = w \mid A = 0).$$

These quantities correspond to different policy questions. While the ATE estimates the effect of treatment if everyone were treated versus if no one were treated, the ATT and ATU estimate effects in subgroups based on actual treatment assignment.

Box 3.1 (R): Bootstrap inference for multivariate G-computation

```r
library(dplyr)

set.seed(123)
data <- data.frame(
  A = rbinom(1000, 1, 0.5),
  W = rbinom(1000, 1, 0.5),
  Y = rbinom(1000, 1, 0.5)
)

cond_probs <- data %>%
  group_by(A, W) %>%
  summarise(P_Y = mean(Y), .groups = 'drop')

p_W <- data %>%
  group_by(W) %>%
  summarise(P_W = n() / nrow(data), .groups = 'drop')

p_W1 <- data %>%
  filter(A == 1) %>%
  group_by(W) %>%
  summarise(P_W1 = n() / n(), .groups = 'drop')

p_W0 <- data %>%
  filter(A == 0) %>%
  group_by(W) %>%
  summarise(P_W0 = n() / n(), .groups = 'drop')

ate <- sum((cond_probs$P_Y[cond_probs$A == 1] -
            cond_probs$P_Y[cond_probs$A == 0]) * p_W$P_W)

att <- sum((cond_probs$P_Y[cond_probs$A == 1] -
            cond_probs$P_Y[cond_probs$A == 0]) * p_W1$P_W1)

atu <- sum((cond_probs$P_Y[cond_probs$A == 1] -
            cond_probs$P_Y[cond_probs$A == 0]) * p_W0$P_W0)
```

```r
cat("ATE:", round(ate, 3), "\n")
cat("ATT:", round(att, 3), "\n")
cat("ATU:", round(atu, 3), "\n")
```

Box 3.11 (Stata): Bootstrap for multivariate parametric G-formula

```stata
capture program drop ATE
            program define ATE, rclass
               capture drop y1
               capture drop y0
               reg $Y $W if $A==1
               predict double y1, xb
               quiet sum y1
               reg $Y $W if $A==0
               predict double y0, xb
               quiet sum y0
               mean y1 y0
               lincom _b[y1]-_b[y0]
               return scalar ace =`r(estimate)'
            end
            qui bootstrap r(ace), reps(1000) seed(1): ATE dots
            estat boot, all

/* 4 Inverse probability of treatment weighting  */
/* 4.1 Inverse probability of treatment weighting based on the propensity score plus regressi
```

3.4.1.1 Bootstrap procedure for statistical inference

Since the g-formula involves a multi-step estimation process–first estimating conditional means and then standardizing over a distribution of confounders–closed-form standard errors are generally not available.(Efron & Tibshirani, 1993) As such, the bootstrap is a practical alternative for estimating standard errors and constructing confidence intervals.(Efron, 1982; Efron & Tibshirani, 1993)

The bootstrap approximates the sampling distribution of an estimator by resampling the observed data with replacement and recalculating the estimator on each resampled dataset. From the distribution of the bootstrap replicates, one can compute the standard error as the sample standard deviation. Confidence intervals can be derived using different methods depending on the shape of the bootstrap distribution:

- **Normal-based intervals** assume approximate symmetry.

- **Percentile intervals** use the empirical quantiles of the bootstrap estimates.
- **Bias-corrected and accelerated (BCa) intervals** correct for bias and skewness.

The choice of interval depends on the observed shape of the bootstrap distribution.(Jung et al., 2019) Histograms or density plots are often used to guide this choice.

3.4.1.2 R Implementation

Box 3.2 (R): Parametric G-computation with multiple confounders

```r
library(boot)

bootstrap_ate <- function(data, indices) {
  boot_data <- data[indices, ]
  cond_probs <- boot_data %>%
    group_by(A, W) %>%
    summarise(P_Y = mean(Y), .groups = 'drop')
  p_W <- boot_data %>%
    group_by(W) %>%
    summarise(P_W = n() / nrow(boot_data), .groups = 'drop')
  sum((cond_probs$P_Y[cond_probs$A == 1] -
       cond_probs$P_Y[cond_probs$A == 0]) * p_W$P_W)
}

set.seed(123)
boot_results <- boot(data, statistic = bootstrap_ate, R = 1000)

boot.ci(boot_results, type = c("norm", "perc", "bca"))
```

Box 3.2 (Stata): Bootstrap for Non-parametric G-Formula

```
capture program drop ATE
            program define ATE, rclass
                capture drop ATE
                sumup $Y, by($A $C)
                matrix y00 = r(Stat1)
                matrix y01 = r(Stat2)
                matrix y10 = r(Stat3)
                matrix y11 = r(Stat4)
                gen ATE = ((y11[3,1]-y01[3,1]))*sexm + ((y10[3,1]-y00[3,1]))*sexf
                qui sum ATE
                return scalar ate = `r(mean)'
```

```
end

qui bootstrap r(ate), reps(1000) seed(1): ATE  // Bootstrap 1000 estimates of the
estat boot, all
drop ATE
```

This approach provides a practical method for estimating uncertainty when using the non-parametric g-formula. The next section will introduce the parametric g-formula, which can offer computational advantages when the form of the outcome model is known or correctly specified.

3.4.2 Non-parametric G-formula for a Fully Saturated Regression Model

In practice, the conditional expectations required by the non-parametric g-formula can be estimated using regression models. When the number of confounder strata is small and all confounders are categorical, a fully saturated regression model can be used to obtain a non-parametric estimate of the conditional mean of the outcome given treatment and confounders.

A fully saturated model includes all main effects and interactions between the treatment variable and the confounders. For example, when both the treatment variable $A$and the confounder$W$ are binary, a saturated logistic regression model is given by:

$$\text{logit}(P(Y = 1 \mid A, W)) = \beta_0 + \beta_1 A + \beta_2 W + \beta_3 A \cdot W.$$

This model allows for effect modification by W and does not impose restrictions on the homogeneity of treatment effects across strata.

After fitting the saturated model, the predicted probabilities $\widehat{P}(Y = 1 \mid A = a, W = w)$for each level of$W$can be used to estimate$E[Y^a]$ through standardization. The non-parametric g-formula using the fitted regression model is:

$$\widehat{E}[Y^a] = \sum_w \widehat{P}(Y = 1 \mid A = a, W = w) \cdot \widehat{P}(W = w),$$

and the average treatment effect (ATE) is then:

$$\widehat{ATE} = \widehat{E}[Y^1] - \widehat{E}[Y^0].$$

This approach is fully non-parametric in the sense that it does not rely on parametric assumptions beyond estimating a separate mean for each treatment-confounder stratum. The non-parametric g-formula can also incorporate complex parameters, such as interactions. When there is only one confounder, including an interaction term between the confounder and the exposure creates a fully-saturated model.

3.4.2.1 R Implementation (Point Estimation)

Box 3.3 (R): Bootstrap for parametric G-computation with multiple confounders

```r
set.seed(123)
data <- data.frame(
  A = rbinom(1000, 1, 0.5),
  W = rbinom(1000, 1, 0.5),
  Y = rbinom(1000, 1, 0.5)
)

# Estimate P(W)
p_W <- data %>%
  group_by(W) %>%
  summarise(P_W = n() / nrow(data), .groups = 'drop')

# Fit saturated logistic regression model
saturated_model <- glm(Y ~ A * W, data = data, family = binomial)

# Create combinations of A and W
pred_data <- data.frame(
  A = rep(0:1, each = 2),
  W = rep(0:1, times = 2)
)

# Predict outcome probabilities
pred_data$P_Y <- predict(saturated_model, newdata = pred_data, type = "response")

# Extract predicted values
p_y1 <- pred_data$P_Y[pred_data$A == 1]
p_y0 <- pred_data$P_Y[pred_data$A == 0]

# Compute ATE
ate_saturated <- sum((p_y1 - p_y0) * p_W$P_W)
cat("ATE (saturated model):", round(ate_saturated, 3), "\n")
```

Box 3.3 (Stata): Fully saturated regression model (method 1)

```
* method 1: conditional probabilities
            regress $Y ibn.$A ibn.$A#$C , noconstant vce(robust) coeflegend
            predictnl ATE = (_b[1.rhc] + _b[1.rhc#c.sex]*sex) - (_b[0bn.rhc] + _b[0bn.rhc#c.s
            qui sum ATE
```

```
display "The ATE is: "  `r(mean)'
drop ATE

// Bootstrap 95% CI
capture program drop ATE
program define ATE, rclass
    capture drop ATE
    regress $Y ibn.$A ibn.$A#c.($C) , noconstant vce(robust) coeflegend
    predictnl ATE = (_b[1.rhc] + _b[1.rhc#c.sex]*sex) - (_b[0bn.rhc] + _b[0bn.rhc
    qui sum ATE
    return scalar ate = `r(mean)'
end

qui bootstrap r(ate), reps(1000) seed(1): ATE
estat boot, all
drop ATE
```

3.4.2.2 Bootstrap Inference

As with other g-formula implementations, there is no closed-form variance expression for the ATE when using a fully saturated model. The bootstrap provides a practical approach for inference by approximating the sampling distribution of the ATE estimator.

Box 3.4 (R): Bootstrap inference for parametric G-computation (one confounder)

```r
library(boot)

# Define bootstrap function
bootstrap_saturated <- function(data, indices) {
  d <- data[indices, ]

  # Estimate P(W)
  p_W <- d %>%
    group_by(W) %>%
    summarise(P_W = n() / nrow(d), .groups = 'drop')

  # Fit saturated model
  saturated_model <- glm(Y ~ A * W, data = d, family = binomial)

  # Create prediction data
  pred_data <- data.frame(
    A = rep(0:1, each = 2),
```

```
    W = rep(0:1, times = 2)
  )

  # Predict probabilities
  pred_data$P_Y <- predict(saturated_model, newdata = pred_data, type = "response")
  p_y1 <- pred_data$P_Y[pred_data$A == 1]
  p_y0 <- pred_data$P_Y[pred_data$A == 0]

  # Compute ATE
  sum((p_y1 - p_y0) * p_W$P_W)
}

# Run bootstrap
set.seed(123)
boot_results_sat <- boot(data, statistic = bootstrap_saturated, R = 1000)

# Confidence intervals
boot.ci(boot_results_sat, type = c("norm", "perc", "bca"))
```

Box 3.4 (Stata): Fully saturated regression model using margins (method 2)

```
* method 2: marginal probabilities
            regress $Y ibn.$A ibn.$A#$C , noconstant vce(robust) coeflegend

            * Marginal probability in each treatment group
            margins $A , vce(unconditional)

            * Difference in marginal probability between treatment groups
            margins r.$A , contrast(nowald)

/* 3.2 PARAMETRIC G-FORMULA */

* One confounder
```

The bootstrap allows researchers to obtain standard errors and confidence intervals that reflect the uncertainty inherent in estimating the conditional expectations and the marginalization process. As with all bootstrap methods, the validity of the inference depends on a sufficiently large sample size and adequate representation of variability in the observed data.

3.4.3 Functional Delta Method for Confidence Intervals

The Delta method is a statistical approach to derive the SE of an asymptotically normally distributed estimator. It uses a first-order Taylor approximation, which is how we approximate the distribution of a function using a tangent line (i.e., the first derivative).(Oehlert, 1992) Therefore, using the Delta method here we assume that the ATE estimate from the G-computation is normally distributed.(Kennedy, 2016)

While the bootstrap is a popular and flexible approach for estimating uncertainty in causal inference, it can be computationally intensive, especially when the estimator is complex or sample sizes are large. As an alternative, the *functional delta method* provides an analytic approximation to the standard error and confidence interval of an estimator. It is particularly useful when the quantity of interest can be expressed as a smooth function of sample statistics, such as sample means or proportions. The delta method avoids resampling by using a Taylor series approximation to quantify how uncertainty in the inputs propagates to the final estimate.

The functional delta method is an extension of the classical delta method, which provides a way to approximate the variance of a function of an estimator. Suppose we have an estimator $\hat{\theta}$that converges in distribution to a normal distribution, and we are interested in a function$g(\hat{\theta})$. If $g(\cdot)$is differentiable atθ, then the delta method states:

$$\sqrt{n}\left(g(\hat{\theta}) - g(\theta)\right) \xrightarrow{d} \mathcal{N}\left(0, \left[g'(\theta)\right]^2 \cdot \sigma^2\right),$$

where $g'(\theta)$is the derivative of$g$atθ, and $\sigma^2$is the asymptotic variance of$\hat{\theta}$.

In the case of the g-formula, we are typically interested in the *average treatment effect* (ATE), defined as:

$$ATE = \sum_w \left[E(Y \mid A = 1, W = w) - E(Y \mid A = 0, W = w)\right] P(W = w).$$

This is a linear combination of conditional expectations $E(Y \mid A = a, W = w)$, each of which can be estimated using sample means and weighted by the marginal distribution of W. Since the ATE is a linear function of these sample means, we can apply the delta method to approximate the variance of the estimated ATE.

Let us denote: - $\hat{\mu}_{a,w} = \hat{E}(Y \mid A = a, W = w)$: the estimated mean outcome for treatment group $a$and stratum$w$ - $\hat{p}_w = \hat{P}(W = w)$: the proportion of the sample in stratum w

The estimated ATE is:

$$\widehat{ATE} = \sum_w \left(\hat{\mu}_{1,w} - \hat{\mu}_{0,w}\right) \hat{p}_w.$$

Since this is a sum of estimated means times weights, its variance can be approximated using the delta method:

$$\mathrm{Var}(\widehat{ATE}) \approx \sum_w \hat{p}_w^2 \left[\mathrm{Var}(\hat{\mu}_{1,w}) + \mathrm{Var}(\hat{\mu}_{0,w})\right].$$

Each $\hat{\mu}_{a,w}$ is a sample mean, and its variance can be estimated as:

$$\text{Var}(\hat{\mu}_{a,w}) \approx \frac{\hat{\sigma}^2_{a,w}}{n_{a,w}},$$

where $\hat{\sigma}^2_{a,w}$is the sample variance of$Y$among individuals with$A = a$and$W = w$, and $n_{a,w}$ is the number of such individuals in the sample.

This approximation assumes that the sample means $\hat{\mu}_{1,w}$and$\hat{\mu}_{0,w}$are approximately independent across$w$, and that the weights $\hat{p}_w$ are treated as fixed (i.e., not contributing to the variance).

3.4.3.1 R Implementation

Box 3.5 (R): Parametric G-computation with a single confounder

```r
# Estimate conditional means, variances, and sample sizes
summary_stats <- data %>%
  group_by(A, W) %>%
  summarise(
    mu = mean(Y),
    var = var(Y),
    n = n(),
    .groups = 'drop'
  )

# Separate estimates by treatment group
mu1 <- summary_stats %>% filter(A == 1) %>% arrange(W)
mu0 <- summary_stats %>% filter(A == 0) %>% arrange(W)

# Estimate P(W)
p_W <- data %>% group_by(W) %>% summarise(p = n() / nrow(data)) %>% arrange(W)

# Estimate ATE
ate_delta <- sum((mu1$mu - mu0$mu) * p_W$p)

# Estimate variance using delta method
var_ate <- sum((p_W$p^2) * ((mu1$var / mu1$n) + (mu0$var / mu0$n)))
se_ate <- sqrt(var_ate)

# Construct 95% confidence interval
ci_lower <- ate_delta - 1.96 * se_ate
ci_upper <- ate_delta + 1.96 * se_ate
```

```r
cat("ATE (delta method):", round(ate_delta, 3), "\n")
cat("SE:", round(se_ate, 3), "\n")
cat("95% CI:", round(ci_lower, 3), "-", round(ci_upper, 3), "\n")
```

Box 3.5 (Stata): Parametric G-formula by hand (one confounder)

```
* Calculations by hand
            * Regresion model and expected probability amongst treated those with RHC
            regress $Y $C if $A==1
            predict double y1hat
            * Regression model and expected probability amongst untreated those without RHC
            regress $Y $C if $A==0
            predict double y0hat
            mean y1hat y0hat
            * Difference between expected probabilities (ATE) and biased confidence interval
            lincom _b[y1hat] - _b[y0hat]
```

The functional delta method offers a mathematically elegant way to obtain approximate standard errors and confidence intervals, particularly when the estimand is a smooth function of averages or proportions. It provides an efficient analytic alternative to resampling-based methods, though it requires some care in variance estimation and interpretation. When the conditions for applying the delta method are met, it can be a fast and accurate tool for inference in causal effect estimation.

3.5 Parametric g-formula

In contrast to the nonparametric methods (i.e., probability distribution free or infinite dimensions), parametric methods are not affected by the curse of dimensionality.(Boos & Stefanski, 2013) However, to compute the ATE parametrically we have to assume there is a particular probability distribution that fits the distribution of our data.

3.5.1 Parametric G-formula for One Confounder

The parametric g-formula is a version of g-computation that relies on specified parametric models to estimate the conditional expectation of the outcome given treatment and confounders. Whereas the non-parametric g-formula requires estimating these expectations separately in each stratum of the confounders, the parametric version uses a model–typically regression–to summarize the outcome distribution as a function of treatment and covariates. This approach

is particularly useful when the number of confounders is large, or when the confounders are continuous, making non-parametric estimation impractical due to data sparsity.

By imposing a functional form, the parametric g-formula can borrow strength across covariate strata and improve efficiency, though at the cost of potential bias if the model is misspecified. The key idea is to fit a model for the conditional expectation (E[Y A, W]), and then use it to predict the potential outcomes under each treatment level (A = a) for every individual, followed by averaging over the empirical distribution of the covariates.

3.5.1.1 Estimation Procedure

Suppose we are interested in estimating the average treatment effect (ATE) of a binary treatment (A) on a binary outcome (Y), adjusting for a single binary confounder (W). Using the parametric g-formula, we proceed in two steps:

1. **Model the outcome**: Fit a regression model, such as logistic regression,

$$\text{logit}(P(Y = 1 \mid A, W)) = \beta_0 + \beta_1 A + \beta_2 W + \beta_3 A \cdot W.$$

   This model allows for an interaction between treatment and confounder to capture potential effect modification.

2. **Standardize over the population**: Predict potential outcomes under $A = 1 \text{and} A = 0$for each individual in the dataset, keeping$W$ fixed at its observed value. Then, compute:

$$\widehat{E}[Y^1] = \frac{1}{n}\sum_{i=1}^{n} \hat{E}[Y_i \mid A = 1, W_i], \quad \widehat{E}[Y^0] = \frac{1}{n}\sum_{i=1}^{n} \hat{E}[Y_i \mid A = 0, W_i].$$

   The average treatment effect (ATE) is given by:

$$\widehat{ATE} = \widehat{E}[Y^1] - \widehat{E}[Y^0].$$

   This procedure estimates what would happen if, counter to fact, everyone in the sample had received treatment ((A = 1)), and separately, what would have happened if everyone had received control ((A = 0)). The difference in these averages provides an estimate of the causal effect.

3.5.1.2 R Implementation

Box 3.6 (R): Bootstrap inference for the non-parametric G-formula ATE

```r
set.seed(123)
data <- data.frame(
  A = rbinom(1000, 1, 0.5),
  W = rbinom(1000, 1, 0.5),
  Y = rbinom(1000, 1, 0.5)
)

# Fit parametric model
param_model <- glm(Y ~ A * W, data = data, family = binomial)

# Predict counterfactual outcomes
data1 <- data; data1$A <- 1
data0 <- data; data0$A <- 0

pred1 <- predict(param_model, newdata = data1, type = "response")
pred0 <- predict(param_model, newdata = data0, type = "response")

# Estimate ATE
ate_param <- mean(pred1 - pred0)
cat("ATE (parametric g-formula):", round(ate_param, 3), "\n")
```

Box 3.6 (Stata): Parametric G-formula using teffects (one confounder)

```
teffects ra ($Y $C) ($A) // Parametric g-formula implemented in Stata
```

This approach avoids the sparsity issues of non-parametric stratification and can handle continuous confounders or high-dimensional covariates by specifying appropriate models. However, it is crucial that the model for (E[Y A, W]) is correctly specified. Misspecification can lead to biased estimates of causal effects. For this reason, model diagnostics and robustness checks should be considered when using the parametric g-formula.

3.5.1.3 Bootstrap Inference

In the parametric g-formula, the ATE is computed by plugging model-based predictions into a functional expression. While the parametric model may provide valid point estimates, standard errors and confidence intervals derived from the model's internal variance estimates do not account for the entire g-computation procedure, which includes both model fitting and marginalization steps. Therefore, model-based (analytic) standard errors can be misleading.

The bootstrap offers a solution by approximating the full sampling distribution of the ATE through repeated resampling. Each bootstrap replicate involves refitting the outcome model

and re-estimating the ATE. The resulting distribution of ATEs captures uncertainty due to both the model fitting and the g-formula estimation process.

Box 3.7 (R): Non-parametric G-computation for the ATE

```r
library(boot)

# Define bootstrap function for parametric g-formula
boot_parametric <- function(data, indices) {
  d <- data[indices, ]
  model <- glm(Y ~ A * W, data = d, family = binomial)
  d1 <- d; d1$A <- 1
  d0 <- d; d0$A <- 0
  pred1 <- predict(model, newdata = d1, type = "response")
  pred0 <- predict(model, newdata = d0, type = "response")
  mean(pred1 - pred0)
}

# Run bootstrap
set.seed(123)
boot_results <- boot(data, statistic = boot_parametric, R = 1000)

# Confidence intervals
boot.ci(boot_results, type = c("norm", "perc", "bca"))
```

Box 3.7 (Stata): Bootstrap for parametric G-formula (one confounder)

```
capture program drop ATE
            program define ATE, rclass
               capture drop y1
               capture drop y0
               reg $Y $C if $A==1
               predict double y1, xb
               quiet sum y1
               reg $Y $C if $A==0
               predict double y0, xb
               quiet sum y0
               mean y1 y0
               lincom _b[y1]-_b[y0]
               return scalar ace =`r(estimate)'
            end
            qui bootstrap r(ace), reps(1000) seed(1): ATE
            estat boot, all
```

```
* More than one confounder naive approach
            regress $Y $A $W
            bootstrap, reps(1000) seed(1): regress $Y $A $W
```

Bootstrap confidence intervals allow for inference that reflects the entire estimation procedure, making them particularly valuable in causal inference settings where estimators are functions of several estimated components.

3.5.2 Parametric G-formula for Multiple Confounders

When more than one confounder is present, the parametric g-formula remains applicable by fitting a model that includes the treatment, all confounders, and optionally interaction terms. This generalization is essential in real-world applications, where multiple covariates are needed to control for confounding.

The process involves fitting a model for the outcome as a function of treatment and confounders, then using this model to predict potential outcomes under both treatment and control for each individual in the sample. These predictions are then averaged to estimate $E[Y^1]$and$E[Y^0]$, and the difference yields the average treatment effect (ATE).

Let $W = (W_1, W_2, \dots, W_k)$be a set of$k$ confounders. A typical logistic regression model for a binary outcome might be written as:

$$\text{logit}(P(Y = 1 \mid A, W)) = \beta_0 + \beta_1 A + \sum_{j=1}^{k} \beta_{j+1} W_j + \sum_{j=1}^{k} \gamma_j A \cdot W_j.$$

In this model: - β_0 is the intercept, representing the baseline log-odds of the outcome when all predictors are zero. - $\beta_1$is the main effect of treatment$A$, adjusted for confounders. - β_{j+1}represents the main effect of each confounderW_j. - $\gamma_j$captures the interaction between treatment and confounderW_j, allowing the treatment effect to vary across levels of the confounder (i.e., effect modification).

By including both main effects and interaction terms, the model can flexibly account for how treatment effects may differ across confounder strata.

As before, the potential outcomes are predicted under both treatment levels and averaged across the population:

$$\widehat{E}[Y^a] = \frac{1}{n} \sum_{i=1}^{n} \hat{E}[Y_i \mid A = a, W_i], \quad \widehat{ATE} = \widehat{E}[Y^1] - \widehat{E}[Y^0].$$

R Implementation (Multiple Confounders)

Box 3.8 (R): Computing ATE, ATT, and ATU from simulated data

```r
set.seed(123)
data <- data.frame(
  A = rbinom(1000, 1, 0.5),
  W1 = rnorm(1000),
  W2 = rbinom(1000, 1, 0.4),
  W3 = sample(1:3, 1000, replace = TRUE)
)

# Generate outcome with some treatment effect and confounding
data$Y <- rbinom(1000, 1, plogis(-0.5 + 0.8 * data$A - 0.3 * data$W1 +
                                 0.5 * data$W2 - 0.2 * (data$W3 == 2)))

# Fit logistic regression with treatment, confounders, and interactions
param_model <- glm(Y ~ A * (W1 + W2 + factor(W3)), data = data, family = binomial)

# Predict under treatment and control
data1 <- data; data1$A <- 1
data0 <- data; data0$A <- 0

pred1 <- predict(param_model, newdata = data1, type = "response")
pred0 <- predict(param_model, newdata = data0, type = "response")

# Estimate ATE
ate_multi <- mean(pred1 - pred0)
cat("ATE (multiple confounders):", round(ate_multi, 3), "\n")
```

Box 3.8 (Stata): Parametric G-formula by hand (multiple confounders)

```
* Regresion model and expected probability amongst treated those with RHC
            regress $Y $W if $A==1
            predict double y1hat
            * Regression model and expected probability amongst untreated those without RHC
            regress $Y $W if $A==0
            predict double y0hat
            mean y1hat y0hat
            * Difference between expected probabilities (ATE) and biased confidence interval
            lincom _b[y1hat] - _b[y0hat]
```

This model can include both categorical and continuous covariates and their interactions with treatment. If the number of covariates is large or the model becomes overly complex, regularization (e.g., LASSO) or machine learning methods can be used to model $E[Y \mid A, W]$, though uncertainty estimation requires more advanced tools.

It is essential to use domain knowledge to decide which variables and interactions to include. Including unnecessary interaction terms may increase variance, while omitting important ones can introduce bias. As with simpler models, the results depend on the correctness of the model. Diagnostics such as those in Section 3.6.3 should be applied to assess model adequacy.

3.5.2.1 Bootstrap for Confidence Intervals

As with simpler cases, there is no closed-form variance for the ATE from the parametric g-formula with multiple confounders. Since the estimation involves both model fitting and standardization, model-based standard errors may not be valid. The bootstrap provides a practical solution.

The bootstrap resamples the data with replacement, refits the model in each resample, and recalculates the ATE. This gives an empirical sampling distribution of the ATE estimate. The standard deviation of the bootstrap estimates provides an estimate of the standard error, and confidence intervals can be derived using:

- **Normal approximation** – assumes bootstrap estimates are normally distributed.
- **Percentile method** – uses the 2.5% and 97.5% percentiles of the bootstrap estimates.
- **Bias-corrected and accelerated (BCa)** – adjusts for bias and skewness in the bootstrap distribution.

R Implementation:

Box 3.9 (R): Simulating data for ATE, ATT, and ATU estimation

```r
library(boot)

# Define bootstrap function
boot_multi <- function(data, indices) {
  d <- data[indices, ]
  model <- glm(Y ~ A * (W1 + W2 + factor(W3)), data = d, family = binomial)
  d1 <- d; d1$A <- 1
  d0 <- d; d0$A <- 0
  pred1 <- predict(model, newdata = d1, type = "response")
  pred0 <- predict(model, newdata = d0, type = "response")
  mean(pred1 - pred0)
}

# Run bootstrap
set.seed(123)
boot_results_multi <- boot(data, statistic = boot_multi, R = 1000)
```

```r
# Display confidence intervals
boot.ci(boot_results_multi, type = c("norm", "perc", "bca"))
```

Box 3.9 (Stata): Parametric G-formula using teffects (multiple confounders)

```
teffects ra ($Y $W) ($A)
```

The bootstrap allows inference that reflects the full complexity of the g-computation procedure, making it an essential complement to the parametric g-formula when dealing with multiple confounders.

3.5.3 Model Diagnostics and Robustness Checks

Because the validity of the parametric g-formula relies on the correct specification of the outcome model $E[Y \mid A, W]$, it is essential to assess the adequacy of the model. Several strategies can be used to evaluate model fit and robustness:

1. **Assess model fit:** Use standard regression diagnostics such as the Hosmer-Lemeshow test, plots of observed vs predicted values, and residual analyses. Check calibration across strata of $A$and$W$.
2. **Check for non-linearity and interactions:** For continuous covariates, consider using splines or polynomial terms. For categorical covariates, assess whether interactions with A are needed.
3. **Evaluate effect modification:** Test whether the effect of Avaries across levels of$W$. Failing to include necessary interaction terms can bias the ATE.
4. **Compare with non-parametric estimates:** If feasible, compare the parametric g-formula estimate with the non-parametric version. Discrepancies may indicate model misspecification.
5. **Cross-validation:** Use resampling methods such as k-fold cross-validation to assess the predictive performance and generalizability of the model.
6. **Sensitivity analysis:** Explore how the estimated ATE changes under alternative model specifications, such as omitting interaction terms or using different link functions.

Conducting thorough model diagnostics strengthens confidence in the parametric g-formula and helps avoid misleading causal inferences due to model misspecification. When diagnostics suggest lack of fit, more flexible methods–such as the non-parametric g-formula or targeted maximum likelihood estimation (TMLE)–may be warranted.

3.6 Conclusion

The g-formula represents a fundamental advance in causal inference methodology. By standardizing outcome predictions across the confounder distribution, it directly estimates marginal causal effects — even in the presence of effect modification — unlike standard regression adjustment. The non-parametric g-formula (via stratification or Monte Carlo simulation) is flexible but requires rich data and sufficient sample size within strata. The parametric g-formula is more practical in moderate samples but relies on correct model specification. The delta method and bootstrap provide complementary tools for valid inference, with the bootstrap offering the advantage of avoiding restrictive distributional assumptions.

Key takeaways from this chapter: - The g-formula estimates marginal causal effects by standardizing conditional expectations across the confounder distribution. - It correctly handles effect modification by confounders — a key advantage over standard regression. - The non-parametric g-formula is asymptotically unbiased but can be inefficient in small samples or with many confounders. - The parametric g-formula is efficient but requires careful model diagnostics. - The functional delta method and bootstrap provide valid standard errors and confidence intervals. - When models are uncertain, doubly-robust methods like TMLE (Chapter 5) offer additional protection.

3.7 Glossary

Bootstrap A resampling technique for estimating the sampling distribution of a statistic by repeatedly sampling with replacement from the observed data.

Collapsibility A property where the conditional effect measure equals the marginal effect measure; odds ratios and hazard ratios are generally non-collapsible.

Delta method A technique for approximating the variance of a function of an estimator using a first-order Taylor expansion.

Effect modification When the causal effect of treatment varies across levels of a covariate; also called heterogeneity of treatment effects.

G-formula A method for estimating marginal causal effects by standardizing conditional outcome expectations across the confounder distribution; also known as G-computation.

Monte Carlo simulation A computational technique that uses repeated random sampling to approximate complex probability distributions or integrals.

Double robust methods $

4 Methods Based on the Propensity Score

4.1 Introduction

In observational studies, estimating causal effects is complicated by the presence of confounding—systematic differences in baseline characteristics between treatment groups. While one approach to adjusting for confounding is through outcome modeling (e.g., the g-formula), an alternative strategy involves balancing the distribution of confounders between treatment groups. This is the central idea behind propensity score methods.

The *propensity score*, introduced by (Rosenbaum & Rubin, 1983b), is defined as the probability of receiving the treatment given a set of observed covariates:

$$e(W) = P(A = 1 \mid W),$$

where $A$is the binary treatment indicator and$W$is a vector of observed covariates. The key insight is that, under certain assumptions, conditioning on the propensity score is sufficient to control for confounding. Specifically, if treatment assignment is strongly ignorable given the covariates$W$, then it is also ignorable given the scalar quantity $e(W)$. That is:

$$Y^0, Y^1 \perp A \mid e(W),$$

where Y^0andY^1 are the potential outcomes under control and treatment, respectively.

This property makes the propensity score a powerful tool: it allows us to reduce a potentially high-dimensional confounding problem into a one-dimensional balancing problem. By adjusting for the propensity score—rather than all covariates directly—we can remove confounding bias due to observed covariates, provided certain conditions are met.

In settings with multiple confounders, the validity of the propensity score relies heavily on correctly modeling the propensity score. In a perfectly randomized experiment, exchangeability is inherent due to the randomization process. However, in an observational study, conditional exchangeability can only be assumed if all confounders have been accurately captured by the propensity score. To fully exploit this conditional exchangeability, the propensity score, which is inherently unknown, must be appropriately modeled using the available data.

In this chapter, we explore the range of methods built upon the propensity score. These include: - **Matching**, where treated and control individuals with similar propensity scores are compared; - **Stratification**, where the data are divided into strata (e.g., quintiles) of the

propensity score and comparisons are made within strata; - **Covariate adjustment using the propensity score**, where the score is included as a covariate in a regression model; - **Inverse probability of treatment weighting (IPTW)**, which uses the propensity score to create a pseudo-population in which treatment is independent of covariates; - And **doubly robust estimators**, which combine propensity score modeling with outcome modeling.

Each method has strengths and limitations. Unlike outcome modeling approaches such as the g-formula, propensity score methods do not rely on specifying the correct model for the outcome. Instead, they shift the modeling burden to the treatment assignment mechanism. This can be advantageous when the outcome is difficult to model, but it also introduces new challenges, including the need for careful diagnostic checks and attention to positivity and overlap.

In the sections that follow, we describe each of these methods in detail, illustrate them with examples and code, and provide guidance on when and how to apply them in practice.

4.2 The Propensity Score

4.2.1 Key Properties of the Propensity Score

The usefulness of the propensity score in causal inference stems from a set of key theoretical properties. These properties justify the use of the propensity score as a balancing score and support its role in adjusting for confounding in observational studies.

4.2.2 The Balancing Property

The fundamental property of the propensity score is its ability to balance covariates between treated and untreated individuals. Formally, for individuals with the same value of the propensity score, the distribution of observed covariates $W$is independent of treatment assignment$A$. That is,

$$A \perp W \mid e(W),$$

where $e(W) = P(A = 1 \mid W)$. This means that within strata of the propensity score, the treated and untreated groups are expected to have similar distributions of baseline covariates. This balancing property is essential, as it enables causal comparisons by mimicking the balance achieved through randomization.

4.2.3 Unconfoundedness Given the Propensity Score

If the potential outcomes are independent of treatment assignment conditional on covariates—that is,

$$Y^0, Y^1 \perp A \mid W,$$

then this also holds conditional on the propensity score:

$$Y^0, Y^1 \perp A \mid e(W).$$

This result, sometimes called the *strong ignorability given the propensity score*, implies that adjusting for the propensity score alone is sufficient to remove confounding bias due to measured covariates, under the assumption that all confounders are included in W.

4.2.4 Common Support and Positivity

For propensity score methods to be valid, there must be sufficient overlap in the distribution of the propensity score between the treated and untreated groups. This is known as the *common support* or *overlap* condition. Formally, for every value of the covariates W, we require:

$$0 < P(A = 1 \mid W) < 1.$$

This assumption, known as *positivity*, ensures that each individual has a positive probability of receiving both treatment and control, given their covariates. If some individuals are deterministically treated or untreated based on their covariates, causal effects for those individuals are not identifiable.

In practice, violations of positivity are detected when there are extreme propensity scores close to 0 or 1. These lead to practical issues such as poor covariate balance and unstable estimates, particularly in weighting-based methods like IPTW.

4.2.5 Bias–Variance Trade-off in Propensity Score Methods

Propensity score methods often involve a trade-off between bias and variance. For example, narrowing the matching caliper or stratifying into more finely grained subclasses can reduce bias but may also increase variance due to smaller effective sample sizes. Conversely, wider matches or coarser stratification increase the sample size but risk residual confounding.

The choice of propensity score method (e.g., matching, stratification, weighting) and implementation details (e.g., number of strata, use of replacement in matching) should be guided by diagnostics such as covariate balance measures and sensitivity analyses.

4.2.6 Estimating the Propensity Score

The first step in applying propensity score methods is to estimate the propensity score—defined as the probability of receiving the treatment, given observed covariates:

$$e(W) = P(A = 1 \mid W).$$

In practice, this conditional probability is not known and must be estimated from the data. The choice of model for estimating the propensity score is crucial, as all subsequent causal inferences depend on its quality. An incorrectly specified propensity score model may result in poor covariate balance and biased effect estimates, even if the methods used afterward are correctly implemented.

4.2.7 Logistic Regression

The most common method for estimating the propensity score is *logistic regression*. In this approach, the treatment assignment $A \in \{0, 1\}$is modeled as a function of the covariates$W$:

$$\text{logit}(P(A = 1 \mid W)) = \alpha_0 + \alpha_1 W_1 + \alpha_2 W_2 + \cdots + \alpha_k W_k.$$

This yields a predicted probability for each individual of receiving the treatment, given their covariate profile. Logistic regression has the advantages of being interpretable, familiar, and easy to implement, and it often performs adequately when the number of confounders is moderate and their relationships with treatment are roughly linear on the logit scale.

However, model misspecification is a risk. Important interactions or non-linearities may be missed unless explicitly included. Therefore, flexible modeling and diagnostic checks are essential.

4.2.8 Machine Learning Approaches

As an alternative to parametric models, *machine learning algorithms* can be used to estimate the propensity score in a more flexible, data-adaptive way. These include:

- Classification trees and random forests
- Gradient boosting machines (e.g., XGBoost)
- Neural networks
- Generalized additive models
- Ensemble learners (e.g., Super Learner)

Machine learning methods can capture complex, non-linear relationships and interactions among covariates without the need to specify them manually. This can improve covariate balance and reduce bias. However, these methods may overfit or produce propensity scores near 0 or 1, which can lead to unstable weights in IPTW.

For this reason, ensemble methods like *Super Learner*—which combine multiple candidate models via cross-validation—are increasingly used in modern causal inference to improve robustness and predictive performance.

4.2.9 Model Checking and Covariate Balance

Regardless of the estimation method, it is essential to evaluate whether the resulting propensity score model achieves *covariate balance* between the treated and control groups. The ultimate goal is not predictive accuracy of treatment assignment but rather balance of covariates conditional on the propensity score.

Diagnostics include:

- Standardized mean differences for each covariate before and after adjustment
- Histograms or density plots of propensity scores by treatment group
- Plots of balance vs. propensity score strata or quantiles

A good model may not perfectly predict treatment but should result in *well-balanced covariates* across treatment groups in the adjusted (pseudo-)population. If balance is poor, the model may need to be refit, potentially adding interaction terms, non-linear terms (e.g., splines), or trying a more flexible estimation method.

In summary, the estimation of the propensity score is not a purely predictive task—it is a causal modeling task aimed at achieving balance to support unbiased estimation of treatment effects.

4.3 Methods Using the Propensity Score

4.3.1 Matching on the Propensity Score

Matching is one of the most commonly used propensity score methods. The idea is to pair treated and control individuals with similar values of the estimated propensity score, so that comparisons are made between units with similar covariate distributions. Matching aims to recreate a pseudo-randomized experiment by ensuring that treatment and control groups are balanced on observed covariates.

4.3.1.1 Nearest Neighbor Matching

In nearest neighbor matching, each treated individual is matched to one or more control individuals whose estimated propensity score is closest in absolute value. The most basic form is 1:1 matching without replacement, although other ratios (e.g., 1:2, 1:k) are possible.

This method is intuitive and easy to implement but may result in poor matches if the propensity score distributions between groups do not overlap well.

Box 4.1 (R): Nearest Neighbor Matching

```r
library(MatchIt)

# Simulate data
data <- data.frame(
  A = rbinom(1000, 1, 0.5),
  W1 = rnorm(1000),
  W2 = rbinom(1000, 1, 0.4)
)
data$ps <- glm(A ~ W1 + W2, data = data, family = binomial)$fitted.values

# Perform nearest neighbor matching
match_nn <- matchit(A ~ W1 + W2, data = data, method = "nearest")
summary(match_nn)
matched_data <- match.data(match_nn)
```

Box 4.1 (Stata): Nearest Neighbor Matching

```
* Simulate data
clear all
set seed 1234
set obs 1000
generate A = (runiform() > 0.5)
generate W1 = rnormal()
generate W2 = (runiform() > 0.4)

* Estimate propensity score via logistic regression
logit A W1 W2
predict ps, pr

* Nearest neighbor matching (1:1, no replacement)
* psmatch2 is user-written; install with: ssc install psmatch2
psmatch2 A, pscore(ps) neighbor(1) noreplacement
```

4.3.1.2 Caliper and Radius Matching

Caliper matching imposes a maximum allowable distance (caliper) between the propensity scores of matched pairs. This reduces the risk of poor matches by ensuring that treated units are only matched to controls with sufficiently similar scores.

Radius matching generalizes this idea by allowing each treated individual to be matched with all controls within the caliper range.

Box 4.2 (R): Caliper Matching

```r
match_caliper <- matchit(A ~ W1 + W2, data = data,
                         method = "nearest", caliper = 0.2)
summary(match_caliper)
```

Box 4.2 (Stata): Caliper Matching

```
* Caliper matching with maximum distance of 0.2 on the propensity score
psmatch2 A, pscore(ps) neighbor(1) noreplacement caliper(0.2)
```

4.3.1.3 Matching with or without Replacement

In matching *with replacement*, a control unit can be used as a match for more than one treated unit. This is especially useful when the number of treated and control units is highly imbalanced or when the overlap in propensity scores is limited.

Matching *without replacement* restricts each control to be used only once. This may yield less optimal matches but preserves a more diverse control group.

Box 4.3 (R): Matching with Replacement

```r
match_wr <- matchit(A ~ W1 + W2, data = data, method = "nearest", replace = TRUE)
summary(match_wr)
```

Box 4.3 (Stata): Matching with Replacement

```
* Matching with replacement
psmatch2 A, pscore(ps) neighbor(1) replacement
```

4.3.1.4 Covariate Balance Diagnostics

After matching, it is essential to assess whether covariates are balanced between treatment groups. Standardized mean differences (SMDs) are commonly used for this purpose. A typical threshold for acceptable balance is an absolute SMD less than 0.1.

Box 4.4 (R): Covariate Balance Diagnostics

```
plot(summary(match_nn), type = "qq")  # Balance plot
```

Box 4.4 (Stata): Covariate Balance Diagnostics

```
* Standardized mean differences before and after matching
pstest W1 W2, both graph
```

Other diagnostics include: - Histograms or density plots of propensity scores - Love plots of standardized differences - Covariate-specific p-values (with caution)

4.3.1.5 Estimating Treatment Effects after Matching

Once matching is complete and balance is assessed, the treatment effect can be estimated by comparing outcomes between matched treated and control units. For 1:1 matching, a paired difference in means is often used:

$$\widehat{ATT} = \frac{1}{n_t} \sum_{i \in \text{treated}} \left(Y_i - Y_{j(i)} \right),$$

where $Y_i$is the outcome for treated unit$i$and$Y_{j(i)}$ is the outcome of the matched control.

Box 4.5 (R): Estimating ATT after Matching

```
matched_data <- match.data(match_nn)
mean(matched_data$Y[matched_data$A == 1]) -
  mean(matched_data$Y[matched_data$A == 0])
```

Box 4.5 (Stata): Estimating ATT after Matching

```
* ATT estimation using psmatch2 output (restrict to matched sample)
keep if _weight != 0 & _weight != .
summarize Y if A == 1, meanonly
local y1 = r(mean)
summarize Y if A == 0, meanonly
```

```
local y0 = r(mean)
display "ATT = " %9.4f `y1' - `y0'

* Alternative: built-in teffects psmatch
teffects psmatch (Y) (A W1 W2, logit), atet
```

Bootstrapping is often used to estimate confidence intervals post-matching since standard variance formulas may not be valid in the matched sample.

Summary: Matching on the propensity score is a flexible, intuitive approach to reduce confounding. Success depends on overlap in propensity scores and careful attention to post-matching diagnostics.

4.3.2 Stratification (or Subclassification)

Stratification, also known as subclassification, is a propensity score method that involves dividing the study population into strata (or subclasses) based on quantiles of the estimated propensity score. Within each stratum, the treated and control groups are compared directly. This method reduces confounding by ensuring that comparisons are made between subjects with similar covariate distributions.

4.3.2.1 Forming Strata by Propensity Score Quantiles

The most common approach is to divide the range of the estimated propensity score into quintiles (i.e., five equally sized groups). This aims to create approximately homogeneous groups within which treatment assignment is as-if random.

Box 4.6 (R): Forming Strata by Propensity Score Quantiles

```
data$pscore <- glm(A ~ W1 + W2, data = data, family = binomial)$fitted.values

# Create strata (quintiles)
data$stratum <- cut(data$pscore,
                    breaks = quantile(data$pscore, probs = seq(0, 1, 0.2)),
                    include.lowest = TRUE, labels = FALSE)
table(data$stratum, data$A)
```

Box 4.6 (Stata): Forming Strata by Propensity Score Quantiles

```
* Estimate propensity score via logistic regression
logit A W1 W2
predict pscore, pr

* Create quintile strata based on the propensity score
xtile stratum = pscore, nq(5)
tabulate stratum A
```

More or fewer strata can be used depending on the sample size. Rosenbaum and Rubin (1984) showed that stratification into five strata typically removes about 90% of the bias due to confounding.

4.3.2.2 Effect Estimation within Strata

Once strata are formed, treatment effects are estimated within each stratum separately. For example, within each stratum s, the average treatment effect can be computed as:

$$\widehat{ATE}_s = \bar{Y}_s^1 - \bar{Y}_s^0,$$

where $\bar{Y}_s^1$and$\bar{Y}_s^0$are the average outcomes among treated and control units within stratums, respectively.

Box 4.7 (R): Effect Estimation within Strata

```
by(data, data$stratum, function(df) {
  mean(df$Y[df$A == 1]) - mean(df$Y[df$A == 0])
})
```

Box 4.7 (Stata): Effect Estimation within Strata

```
* Stratum-specific ATE
forvalues s = 1/5 {
  summarize Y if stratum == `s' & A == 1, meanonly
  local y1 = r(mean)
  summarize Y if stratum == `s' & A == 0, meanonly
  local y0 = r(mean)
  display "Stratum `s': ATE = " %9.4f `y1' - `y0'
}
```

If the outcome is binary, logistic regression can also be used within each stratum to estimate risk differences, risk ratios, or odds ratios.

4.3.2.3 Aggregating Across Strata

The overall average treatment effect is computed by aggregating the stratum-specific effects, weighting by the proportion of the population in each stratum:

$$\widehat{ATE} = \sum_{s=1}^{S} \widehat{ATE}_s \times \frac{n_s}{n},$$

where $n_s$is the number of individuals in stratum$s$and$n$ is the total sample size.

Box 4.8 (R): Aggregating Across Strata

```r
library(dplyr)
agg_ate <- data %>%
  group_by(stratum) %>%
  summarise(
    n = n(),
    ate = mean(Y[A == 1]) - mean(Y[A == 0])
  ) %>%
  summarise(weighted_ate = sum(ate * n / sum(n)))
agg_ate
```

Box 4.8 (Stata): Aggregating Across Strata

```
* Stratum-specific ATE with population weights
bysort stratum: generate n_s = _N
bysort stratum: egen y1_s = mean(Y) if A == 1
bysort stratum: egen y0_s = mean(Y) if A == 0
bysort stratum: generate ate_s = y1_s - y0_s if _n == 1

* Weighted average across strata
summarize ate_s [aw = n_s]
```

This estimator approximates the marginal ATE, provided that each stratum contains both treated and control individuals.

4.3.2.4 Assessing Balance within Strata

To evaluate whether stratification has successfully balanced covariates, one should assess covariate balance within each stratum. This can be done by calculating standardized mean differences (SMDs) between treated and control groups within each stratum.

Box 4.9 (R): Assessing Balance within Strata

```r
library(tableone)
CreateTableOne(vars = c("W1", "W2"), strata = "stratum", data = data, factorVars = "A")
```

Box 4.9 (Stata): Assessing Balance within Strata

```
* Standardized mean differences within each stratum
forvalues s = 1/5 {
  display as text "=== Stratum `s' ==="
  foreach var in W1 W2 {
    summarize `var' if stratum == `s' & A == 1, meanonly
    local m1 = r(mean)
    local v1 = r(Var)
    summarize `var' if stratum == `s' & A == 0, meanonly
    local m0 = r(mean)
    local v0 = r(Var)
    local smd = (`m1' - `m0') / sqrt((`v1' + `v0') / 2)
    display "`var': SMD = " %7.4f `smd'
  }
}
```

Visual tools such as Love plots can also be used to summarize covariate balance across all strata.

Summary: Stratification is an intuitive and accessible method to adjust for confounding using the propensity score. Its simplicity and transparency make it particularly useful in moderate sample sizes, although bias can remain if strata are not sufficiently homogeneous.

4.3.3 Covariate Adjustment Using the Propensity Score

An alternative to matching, stratification, or weighting is to include the propensity score directly as a covariate in a regression model for the outcome. This method is sometimes referred to as "covariate adjustment using the propensity score." It involves modeling the outcome as a function of both the treatment and the estimated propensity score.

4.3.3.1 Propensity Score as a Covariate in Regression

In this approach, the outcome Y is modeled as:

$$E[Y \mid A, e(W)] = f(A, e(W)),$$

where $e(W)$ is the estimated propensity score. A typical implementation for a continuous outcome is a linear regression model:

$$Y_i = \beta_0 + \beta_1 A_i + \beta_2 \hat{e}(W_i) + \varepsilon_i,$$

and for a binary outcome, a logistic regression model:

$$\text{logit}(P(Y_i = 1)) = \beta_0 + \beta_1 A_i + \beta_2 \hat{e}(W_i).$$

The coefficient β_1 captures the association between treatment and outcome, adjusted for the propensity score.

This method reduces the dimensionality of adjustment: instead of adjusting for all covariates in W, one adjusts for a scalar summary, $\hat{e}(W)$. However, it is not guaranteed to produce unbiased estimates of the average treatment effect (ATE), especially when the outcome model is misspecified.

Box 4.10 (R): Covariate Adjustment with PS - Linear Regression

```r
# Estimate propensity scores
ps_model <- glm(A ~ W1 + W2, data = data, family = binomial)
data$pscore <- ps_model$fitted.values

# Outcome model using propensity score as a covariate
outcome_model <- lm(Y ~ A + pscore, data = data)
summary(outcome_model)
```

Box 4.10 (Stata): Covariate Adjustment with PS - Linear Regression

```
* Estimate propensity score via logistic regression
logit A W1 W2
predict pscore, pr

* Outcome model with propensity score as a covariate (continuous outcome)
regress Y A pscore
```

For binary outcomes:

Box 4.11 (R): Covariate Adjustment with PS - Logistic Regression

```r
# Logistic regression
logit_model <- glm(Y ~ A + pscore, data = data, family = binomial)
summary(logit_model)
```

Box 4.11 (Stata): Covariate Adjustment with PS - Logistic Regression

```
* Outcome model with propensity score as a covariate (binary outcome)
logit Y A pscore, or
```

This method can be sensitive to misspecification of the outcome model. For example, if $e(W)$has a non-linear relationship with$Y$, then modeling it linearly may yield biased estimates.

4.3.3.2 Comparison with Outcome Modeling

Covariate adjustment using the propensity score differs from traditional outcome modeling (also known as regression adjustment) in that it replaces the vector of covariates $W$with a scalar summary,$e(W)$. In standard outcome regression, the model is:

$$Y_i = \beta_0 + \beta_1 A_i + \beta_2^T W_i + \varepsilon_i.$$

This method directly adjusts for all confounders, assuming the model for $E[Y \mid A, W]$ is correctly specified. While it allows more flexibility in modeling covariates, it also suffers more severely from curse of dimensionality and collinearity.

In contrast, adjusting for $e(W)$ reduces the covariate adjustment to a single dimension. However, it does not offer the same level of protection against confounding, especially in the presence of effect modification, non-linear relationships, or heterogeneous treatment effects.

Key differences: - **Dimensionality:** Propensity score adjustment uses a scalar summary, reducing complexity. - **Model dependency:** Traditional regression depends heavily on correct specification of the full outcome model; PS adjustment relies more on correct treatment model. - **Interpretation:** Coefficients from PS-adjusted models estimate conditional treatment effects, not marginal ATEs.

Although less commonly used in modern practice due to its sensitivity to model misspecification and weaker performance in finite samples, covariate adjustment using the propensity score remains a useful tool in situations with limited overlap or when other methods are infeasible.

For improved performance, this method is often used in conjunction with doubly robust estimators, as discussed in later sections.

4.3.4 Inverse Probability of Treatment Weighting (IPTW)

Inverse Probability of Treatment Weighting (IPTW) is a propensity score-based method that reweights the sample to create a pseudo-population in which treatment assignment is independent of baseline covariates. This allows for the estimation of causal effects by comparing outcomes between weighted treatment groups.

In observational studies, some individuals will be more likely than others to be treated (A=1) due to their characteristics. Suppose some individuals who were treated were unlikely to be treated based on a specific set of features encapsulated in a particular vector of confounders (**W**). To balance the differences in characteristics between treatment groups, we re-weight the outcome variable of these individuals by the inverse of their probability of the treatment (A) actually received (i.e., propensity score). Originally, the weights were motivated from the classical Horvitz and Thompson survey estimator used to re-weight the outcome variable by the inverse probability that it is observed, thus accounting for the sampling process.(Horvitz & Thompson, 1952) The result of this weighting procedure is that, among the treated we up-weight those who had a low probability of being treated, and among the untreated we up-weight those who were unlikely to be untreated; that is, the individuals underrepresented in their treatment group. As a consequence, the weighted set of data is unchanged apart from A and **W** are now conditionally independent. Therefore, a comparison of $Y_w(1)$to $Y_w(0)$ gives a marginal causal effect under the three identification assumptions (Appendix 1) whilst also assuming the propensity score model is correctly specified. The inverse probability of treatment weighting (IPTW), and the g-formula when targeting the same estimand (i,e., the ATE), are equivalent in the nonparametric setting.(J. Robins, 1986; Rosenbaum & Rubin, 1983b) We provide a proof of the equivalence between IPTW and G-computation procedures

using the law of total expectation.

$$\underbrace{E\left(\frac{I(a=1)}{P(A=1 \mid W)}\, Y\right)}_{IPTW} =$$

By definition of expectations...

$$= \sum_{w,a,y} \frac{I(a=1)}{P(A=1|W=w)}\, y\, P(Y=y, A=a, W=w)$$

By the law of total probability...

$$= \sum_{w,a,y} \frac{I(a=1)}{P(A=1 \mid W=w)}\, y\, P(Y=y \mid A=a, W=w)\, P(A=a \mid W=w)\, P(W=w)$$

Cancellation by evaluating at A=1...

$$= \sum_{w,y} y\, P(Y=y \mid A=1, W=w)\, P(W=w)$$

By definition of expectations...

$$= \sum_{w} E(Y \mid A=1, W=w)\, P(W=w)$$

Finally, again by definition of expectations...

$$= \underbrace{E[E(Y \mid A=1, W)]}_{G-computation}$$

Departing from the identification assumptions of the ATE for the regression adjustment G-computation estimand (ATE $= E_w(E(Y|A=1,\mathbf{W}) - E_w(Y|A=0,\mathbf{W}))$, we can rewrite the same estimand as a function of the distribution of A given W (i.e., $P(A=1|\mathbf{W})$, a.k.a propensity score or treatment mechanism).

Therefore, the estimator is given by

$$\text{ATE} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{A_i}{P(A_i=1 \mid W_i)} - \frac{1-A_i}{(1-P(A_i=1 \mid W_i))}\right) Y_i. \tag{4.1}$$

4.3.4.1 Defining Weights for ATE, ATT, and ATU

The basic idea of IPTW is to weight each individual by the inverse of the probability of receiving the treatment they actually received. These weights depend on the estimand of interest:

- **Average Treatment Effect (ATE):**

$$w_i^{ATE} = \begin{cases} \frac{1}{\hat{e}(W_i)} & \text{if } A_i = 1, \\ \frac{1}{1-\hat{e}(W_i)} & \text{if } A_i = 0. \end{cases}$$

- **Average Treatment Effect on the Treated (ATT):**

$$w_i^{ATT} = \begin{cases} 1 & \text{if } A_i = 1, \\ \frac{\hat{e}(W_i)}{1-\hat{e}(W_i)} & \text{if } A_i = 0. \end{cases}$$

- **Average Treatment Effect on the Untreated (ATU):**

$$w_i^{ATU} = \begin{cases} \frac{1-\hat{e}(W_i)}{\hat{e}(W_i)} & \text{if } A_i = 1, \\ 1 & \text{if } A_i = 0. \end{cases}$$

Box 4.12 (R): ATE Weights

```r
ps_model <- glm(A ~ W1 + W2, data = data, family = binomial)
data$pscore <- ps_model$fitted.values

data$weight_ate <- ifelse(data$A == 1,
                          1 / data$pscore,
                          1 / (1 - data$pscore))
```

Box 4.12 (Stata): ATE Weights

```
* Estimate propensity score via logistic regression
logit A W1 W2
predict pscore, pr

* Generate ATE weights (inverse probability weights)
generate weight_ate = cond(A == 1, 1 / pscore, 1 / (1 - pscore))
summarize weight_ate
```

4.3.4.2 Stabilized vs. Unstabilized Weights

There is a modified version of the IPTW estimator (Equation 4.1) consisting of stabilised weights proposed by (**Hajek1971CommentEds?**), which is more commonly used in practice when treatment and exposure vary over time (i.e., time dependent confounding). Stabilised weights should have a mean of 1, but some values could be higher (i.e., large weights). Unstabilized weights can sometimes have large variability, particularly when propensity scores

are close to 0 or 1. Stabilized weights help mitigate this by multiplying the numerator of the weight by the marginal probability of treatment:

$$w_i^{stab} = \begin{cases} \frac{P(A=1)}{\hat{e}(W_i)} & \text{if } A_i = 1, \\ \frac{P(A=0)}{1-\hat{e}(W_i)} & \text{if } A_i = 0. \end{cases}$$

The stabilised version of the IPTW estimator is given by

$$\text{ATE} = \frac{\sum\left(\frac{AY}{P(A=1|W)}\right)}{\sum\left(\frac{A}{P(A=1|W)}\right)} - \frac{\sum\left(\frac{(1-A)Y}{1-P(A=1|W)}\right)}{\sum\left(\frac{(1-A)}{1-P(A=1|W)}\right)}.$$

Box 4.13 (R): Stabilized ATE Weights

```r
p_A <- mean(data$A == 1)
data$stab_weight <- ifelse(data$A == 1,
                           p_A / data$pscore,
                           (1 - p_A) / (1 - data$pscore))
```

Box 4.13 (Stata): Stabilized ATE Weights

```
* Marginal probability of treatment
summarize A, meanonly
local p_A = r(mean)

* Generate stabilized ATE weights
generate stab_weight = cond(A == 1, `p_A' / pscore, (1 - `p_A') / (1 - pscore))
summarize stab_weight
```

4.3.4.3 Dealing with Extreme Weights

Extreme weights occur when estimated propensity scores are close to 0 or 1, leading to instability and large variances. Strategies to manage this include: - **Truncation:** Cap weights at a specified percentile (e.g., 1st and 99th percentiles). - **Weight trimming:** Exclude individuals with weights above a certain threshold. - **Use of stabilized weights:** As previously shown. - **Improved PS estimation:** Use flexible models (e.g., machine learning) to improve PS estimates.

Box 4.14 (R): Weight Truncation

```r
# Truncate at 1st and 99th percentiles
lower <- quantile(data$weight_ate, 0.01)
upper <- quantile(data$weight_ate, 0.99)
data$trunc_weight <- pmin(pmax(data$weight_ate, lower), upper)
```

Box 4.14 (Stata): Weight Truncation

```
* Compute 1st and 99th percentiles of the weights
_pctile weight_ate, p(1, 99)
local lower = r(r1)
local upper = r(r2)

* Truncate weights at these thresholds
generate trunc_weight = min(max(weight_ate, `lower'), `upper')
summarize trunc_weight
```

4.3.4.4 Variance Estimation for IPTW

Variance estimation for IPTW estimators can be challenging due to the complex structure of the weights. Common approaches include:

- **Robust (sandwich) standard errors:** Often used when fitting weighted regression models.
- **Bootstrap:** Resample individuals and re-estimate the treatment effect in each sample.

Box 4.15 (R): Weighted Regression with Robust SEs

```
library(sandwich)
library(lmtest)

# Weighted regression
fit <- lm(Y ~ A, data = data, weights = data$stab_weight)
coeftest(fit, vcov = vcovHC(fit, type = "HC0"))
```

Box 4.15 (Stata): Weighted Regression with Robust SEs

```
* Weighted linear regression with robust (sandwich) standard errors
regress Y A [pw = stab_weight], robust
```

Box 4.16 (R): Bootstrap for IPTW

```
library(boot)

boot_iptw <- function(data, indices) {
  d <- data[indices, ]
  fit <- lm(Y ~ A, data = d, weights = d$stab_weight)
```

```r
  coef(fit)["A"]
}

boot(data, boot_iptw, R = 1000)
```

Box 4.16 (Stata): Bootstrap for IPTW

```
* Bootstrap standard errors for IPTW estimator
capture program drop boot_iptw
program define boot_iptw, rclass
  * Re-estimate propensity score in bootstrap sample
  logit A W1 W2
  predict ps_boot, pr
  * Compute stabilized weights
  summarize A, meanonly
  local pA = r(mean)
  generate wt = cond(A == 1, `pA' / ps_boot, (1 - `pA') / (1 - ps_boot))
  * Weighted regression
  regress Y A [pw = wt]
  return scalar ate = _b[A]
  * Clean up
  drop ps_boot wt
end
bootstrap r(ate), reps(1000) seed(1234): boot_iptw
```

4.3.4.5 Diagnostics and Weight Distributions

Before interpreting IPTW results, it is critical to check whether the weights have created a balanced pseudo-population. Diagnostic steps include:

- **Plotting the distribution of weights:** Helps identify extreme weights.
- **Checking covariate balance after weighting:** Using standardized mean differences.
- **Plotting Love plots or density plots of covariates:** To assess balance visually.

Box 4.17 (R): Check Weights and Balance

```r
hist(data$stab_weight, breaks = 30, main = "Stabilized Weights")

library(cobalt)
bal.tab(A ~ W1 + W2, data = data, weights = data$stab_weight)
love.plot(bal.tab(A ~ W1 + W2, data = data, weights = data$stab_weight))
```

Box 4.17 (Stata): Check Weights and Balance

```
* Histogram of stabilized weights
histogram stab_weight, frequency name(weight_hist, replace)

* Assess covariate balance after IPTW using built-in teffects
teffects ipw (Y) (A W1 W2, logit), ate
tebalance summarize
```

Summary: IPTW is a powerful method for estimating causal effects using the propensity score. Careful diagnostics and attention to weight distribution are necessary to avoid instability and ensure valid inferences.

4.4 Practical Considerations

4.4.1 Assessing Overlap and Positivity

A fundamental assumption in propensity score analysis is the assumption of positivity, also known as common support or overlap. This requires that for all combinations of covariates W, the probability of receiving each treatment level is strictly between zero and one:

$$0 < P(A = 1 \mid W) < 1.$$

Violations of positivity occur when some individuals have near-deterministic treatment assignment based on their covariates. These individuals contribute little to causal identification and may introduce instability in estimates, particularly in weighting methods like IPTW. The overlap of the distribution of propensity scores between treatment groups gives a visual identification regarding the strength of confounding and whether it is acceptable.

Box 4.18 (R): Propensity Score Overlap

```
library(ggplot2)
ggplot(data, aes(x = pscore, fill = factor(A))) +
  geom_density(alpha = 0.4) +
  labs(title = "Propensity Score Overlap", fill = "Treatment")
```

Box 4.18 (Stata): Propensity Score Overlap

```
* Density plot of propensity scores by treatment group
twoway kdensity pscore if A == 0, lcolor(red) lpattern(solid) ///
       || kdensity pscore if A == 1, lcolor(blue) lpattern(dash) ///
       legend(order(1 "Control" 2 "Treated")) ///
       title("Propensity Score Overlap") ///
       xtitle("Propensity Score") ytitle("Density")
```

When overlap is poor, trimming or restriction to regions of common support is sometimes used to improve robustness, albeit at the cost of generalizability.

4.4.2 Checking Covariate Balance

The primary role of the propensity score is to balance observed covariates between treatment groups. Balance diagnostics should be conducted after implementing any propensity score method—whether matching, weighting, or stratification. One commonly used measure is the standardized mean difference (SMD), defined as the difference in means between treated and control groups, divided by the pooled standard deviation.

SMDs close to zero suggest good balance. A common rule of thumb is that absolute SMDs below 0.1 indicate acceptable balance. Visual tools, such as Love plots, provide a convenient way to display covariate balance before and after adjustment.

Box 4.19 (R): Covariate Balance with cobalt

```
library(cobalt)
ps_model <- glm(A ~ W1 + W2, data = data, family = binomial)
data$pscore <- ps_model$fitted.values
bal.tab(A ~ W1 + W2, data = data, weights = data$stab_weight)
love.plot(bal.tab(A ~ W1 + W2, data = data, weights = data$stab_weight))
```

Box 4.19 (Stata): Covariate Balance with cobalt

```
* Estimate propensity score and compute stabilized weights
logit A W1 W2
predict pscore, pr
summarize A, meanonly
local pA = r(mean)
generate stab_w = cond(A == 1, `pA' / pscore, (1 - `pA') / (1 - pscore))

* Covariate balance assessment after IPTW using teffects
teffects ipw (Y) (A W1 W2, logit), ate
```

```
* Love plot of standardized mean differences
tebalance plot
```

If covariate balance is unsatisfactory, the propensity score model may need to be refined by adding interaction terms or nonlinear effects.

When there are near violations of the positivity assumption, the unstabilised weights can have large values, forcing the variance to increase and exacerbate the uncertainty of the ATE estimation. Therefore, it is advisable to explore the distribution of the weights to evaluate the extent to which they balance the distribution of confounders across the levels of the treatment (i.e., equally distributed). As shown by (Austin, 2009), it is common to provide a table with the unweighted and weighted differences of the standardised means of the confounders by the levels of the treatment.

As an example, suppose we have information on a set of covariates given in Table 4.1. Prior to weighting, there was some imbalance (absolute values of the standardised differences close to, or beyond, 0.10) on sex, education level and presence/extent of cancer between treatment groups. A variance ratio (i.e., the ratio of the standardised distribution of the confounders by the levels of the treatment) equal to 1 before and after weighting informs us that the distribution of the confounders across the levels of the treatments is the same (i.e., perfectly balanced). Note, the weighted variance ratio for the continuous variable age is 0.79, which is slightly further from 1 than the variance ratio for the original (unweighted) sample (i.e., 0.82); this slight change is possibly because the weighted mean for age might have greater sampling variance than the unweighted mean.

Table 4.1: Distribution of the treatment before and after applying weights

Confounder	Raw	Weighted	Raw	Weighted
Sex	0.093	0.000	0.977	1.000
Age	-0.061	-0.004	0.817	0.791
Education	0.091	-0.002	1.015	1.027
Race - Black	-0.031	0.002	0.944	1.003
Race - Other	0.020	0.001	1.078	1.004
Cancer - Metastatic	-0.069	-0.000	0.780	1.000
Cancer - Localised	-0.072	0.000	0.879	0.999

Reference groups: race - white, cancer - none

There is no definitive value at which the treatment is considered unbalanced; however, as a guideline, a variance ratio less than 0.5 indicates that the data is not balanced and the potential for the positivity violation must be explored (i.e., when $P(A = a \mid C = c)$is near to zero or one). An additional strategy is to check the distribution of the weights: if there are very large

weights this indicates the violation of the positivity assumption but, also, it can be due to parametric modelling misspecification. Again there is no clear consensus but, when there are very large weights, researchers often set the weights to a less extreme value. (Stürmer et al., 2010) does this by trimming or removing the data at the extremes of the distribution of the weights (e.g., the5^{th}and95^{th}percentiles). Trimming the weights reduces variance (i.e., omitting the largest weights and making the positivity assumption more plausible), but at the expense of introducing bias (Cole & Hernán, 2008). However, another alternative without dropping observations is truncation, whereby all the values of the weights, larger than a user-specified maximum value or percentile (e.g.,1^{st}and99^{th}or$5th$and95^{th}), are replaced by that threshold value (Cole & Hernán, 2008; Xiao et al., 2013). In extreme cases, when the weights are extremely large, changing the estimand could be another solution (e.g., estimating the ATE in a subset of the sample, among only those treated for example, representing the average treatment effect among the treated -ATT-).

4.4.3 Sensitivity to Propensity Score Model Choice

Since propensity scores are typically estimated from a model, their validity depends on correct model specification. For binary exposures, logistic regression is the most common choice, but its parametric nature can lead to bias if important nonlinearities or interactions are omitted. One way to assess sensitivity is to compare covariate balance and treatment effect estimates under alternative model specifications.

Flexible, data-adaptive methods such as generalized additive models, random forests, or ensemble learners like Super Learner may improve propensity score estimation by reducing model misspecification. However, they also require careful diagnostics to ensure that the resulting weights or matches still achieve balance.

Box 4.20 (R): Super Learner for PS Estimation

```r
library(SuperLearner)
X <- data[, c("W1", "W2")]
Y <- data$A
sl_model <- SuperLearner(Y = Y, X = X, family = binomial(),
                         SL.library = c("SL.glm", "SL.randomForest"))
data$pscore_sl <- sl_model$SL.predict
```

Box 4.20 (Stata): Super Learner for PS Estimation

```
* Flexible PS estimation using lasso logistic regression (Stata 16+)
* Lasso automatically selects important covariates and interactions
logit A W1 W2, lasso
predict ps_lasso, pr
```

```
* Compare with standard logistic regression
logit A W1 W2
predict ps_logit, pr
summarize ps_lasso ps_logit

* Alternatively, use random forest (user-written: ssc install rforest)
* rforest A W1 W2
* predict ps_rf, pr
```

Ultimately, sensitivity analyses should be performed to evaluate the robustness of conclusions to different modeling choices.

4.4.4 Choosing Among Propensity Score Methods

Several methods are available for using the propensity score, including matching, stratification, IPTW, and covariate adjustment. The choice among them should be guided by the research question, sample size, distribution of covariates, and practical considerations such as overlap and computational resources.

Matching is often favored for its transparency and intuitive appeal and is particularly effective when the sample size is moderate and sufficient overlap exists. Stratification is easy to implement and understand, though it may not remove all residual confounding. IPTW is powerful for estimating marginal effects but requires careful management of extreme weights. Covariate adjustment is less commonly used on its own due to concerns about model misspecification but may serve as a useful component in doubly robust methods.

Each method has advantages and limitations. It is often useful to implement multiple approaches and compare results as part of a sensitivity analysis. No method is universally superior, and the credibility of the analysis ultimately depends on careful diagnostics and a deep understanding of the data and context.

4.5 Summary and Comparison

This chapter has outlined a suite of methods built upon the propensity score, including matching, stratification, inverse probability of treatment weighting (IPTW), covariate adjustment, and doubly robust estimators. These methods all aim to reduce bias due to measured confounding by balancing covariates between treatment groups. In this final section, we reflect on how these approaches compare with outcome-based methods such as the g-formula, identify situations in which propensity score methods are most appropriate, and summarize their relative strengths and weaknesses.

4.5.1 Comparison with the G-formula

The g-formula, or outcome regression, relies on modeling the outcome as a function of treatment and confounders. This approach directly estimates the expected potential outcomes under each treatment and averages them over the population. In contrast, propensity score methods focus on balancing the distribution of confounders by modeling treatment assignment and use this to indirectly adjust outcome comparisons.

In terms of assumptions, both the g-formula and propensity score methods require unconfoundedness (i.e., no unmeasured confounding) and positivity. However, the g-formula relies more heavily on correctly specifying the outcome model, while propensity score methods transfer this modeling burden to the treatment assignment mechanism. In theory, if both models are correctly specified, the g-formula may be more efficient, especially when effect modification is limited. In practice, the choice often depends on which model is easier to specify accurately given the data and subject-matter knowledge.

Propensity score methods are especially useful when the outcome is rare or difficult to model directly. They are also appealing when investigators prefer to separate design from analysis, as matching and stratification can be implemented without using outcome data. This pre-analysis design step can help reduce bias due to model overfitting or selective model specification.

4.5.2 When to Use Propensity Score Methods

Propensity score methods are most beneficial in observational studies where treatment is not randomly assigned and confounding is suspected. These methods are especially appropriate when the number of covariates is large, or when the outcome is not well understood. They are also helpful when researchers want to avoid strong parametric assumptions about the outcome.

Matching is particularly appealing when a clear comparison group is desired and transparency is a priority. Stratification works well when the propensity score distribution overlaps substantially between groups. IPTW is a good choice for estimating marginal effects, but it requires close attention to positivity and extreme weights. Doubly robust methods, such as AIPTW, are advantageous when there is uncertainty about which model (outcome or treatment) is correctly specified, as they offer consistent estimates if either is valid.

These methods should not be used mechanically. They require thoughtful implementation, including careful model specification, extensive diagnostics, and often sensitivity analyses. When used appropriately, they can greatly reduce bias due to confounding and enhance the credibility of causal claims.

4.5.3 Strengths and Limitations

The major strength of propensity score methods is their ability to achieve covariate balance in high-dimensional settings without modeling the outcome. This makes them attractive when outcome data are noisy or limited. Many implementations, particularly matching and weighting, also offer intuitive interpretations and can be made transparent to non-technical audiences.

However, these methods have limitations. All propensity score methods depend on the strong assumption that all confounders have been measured. If key confounders are omitted, none of the methods can recover an unbiased causal effect. Moreover, methods like IPTW are sensitive to violations of positivity and can become unstable in the presence of extreme weights. Matching discards data and can reduce precision, while stratification may not fully eliminate residual confounding.

Finally, while propensity score methods reduce reliance on modeling the outcome, they do not eliminate the need for modeling altogether. The estimated propensity score is a model-based quantity, and poor specification can lead to imbalance and bias. The emergence of machine learning methods offers opportunities to improve propensity score estimation, but it also introduces challenges related to interpretability and diagnostics.

In summary, propensity score methods offer a flexible and powerful approach to estimating causal effects in observational studies. When implemented carefully and in the right context, they serve as valuable tools in the causal inference toolkit and provide a strong complement to outcome-based methods such as the g-formula.

4.6 Conclusion

Propensity score methods separate the design stage (modelling treatment assignment) from the analysis stage (estimating treatment effects), offering practical advantages in terms of transparency and diagnostic assessment. Matching excels at creating well-balanced comparison groups but discards data. IPTW uses all observations but is sensitive to extreme weights. Stratification and regression adjustment on the propensity score offer intermediate options. The choice among them depends on sample size, overlap in propensity score distributions, and the analyst's tolerance for bias-variance trade-offs.

Key takeaways from this chapter: - Propensity scores reduce the dimensionality of confounding to a single scalar summary. - Matching, stratification, IPTW, and regression on the propensity score each have distinct strengths and limitations. - Covariate balance diagnostics (standardized differences, love plots) are essential before proceeding to outcome analysis. - Positivity violations — when certain covariate patterns are almost always treated or never treated — must be assessed and addressed. - Flexible machine learning methods can improve propensity score estimation but require careful validation.

4.7 Glossary

ATE Average Treatment Effect — the average causal effect of treatment in the population.

ATT Average Treatment Effect on the Treated — the average causal effect among those who received treatment.

IPTW Inverse Probability of Treatment Weighting — a method that reweights observations by the inverse of the propensity score to create a pseudo-population with balanced covariates.

Matching A method that pairs treated and control units with similar propensity scores to estimate causal effects.

Positivity The assumption that every individual has a non-zero probability of receiving each treatment level given their covariates.

Propensity score The probability of receiving treatment given observed covariates, $e(X) = \mathbb{P}(A = 1 \mid X)$.

SMD Standardized Mean Difference — a measure of covariate balance between treatment groups.

Stratification Dividing the sample into strata based on the propensity score and estimating treatment effects within each stratum.

TMLE Targeted Maximum Likelihood Estimation — a doubly-robust, semiparametric efficient estimator introduced in Chapter 5.

```
::: {.quarto-book-part}


`<!-- quarto-file-metadata: eyJyZXNvdXJjZURpciI6Ii4ifQ== -->`{=html}

```{=html}
<!-- quarto-file-metadata: eyJyZXNvdXJjZURpciI6Ii4iLCJib29rSXRlbVR5cGUiOiJwYXJ0IiwiYm9va0lOZ
```
```

5 Advanced Methods

:::

6 Double-robust estimators

6.1 Introduction to Double Robustness

6.1.1 Motivation: Why Double Robustness Matters

In earlier chapters, we explored several foundational methods for estimating causal effects in observational studies, including regression adjustment, propensity score methods, inverse probability of treatment weighting (IPTW), and the g-formula. Each of these methods relies on correctly specifying a model – whether that be a model for the outcome, the treatment assignment, or the joint distribution of covariates, treatment, and outcome.

However, in real-world applications, it is often difficult to know whether the chosen model is correct. Model misspecification is a pervasive concern that can lead to biased estimates and invalid inference. This motivates the need for methods that offer greater protection against such misspecification.

Double-robust estimators are designed with this challenge in mind. They combine models for both the treatment assignment (e.g., the propensity score) and the outcome. Remarkably, they are consistent and asymptotically unbiased if *either* the treatment model *or* the outcome model is correctly specified—not necessarily both. This property makes double-robust methods particularly appealing in applied settings, offering a safeguard when uncertainty about model specification is high.

In the sections that follow, we will build on the concepts introduced in previous chapters to develop and understand double-robust estimators, starting with the intuition behind them and proceeding to concrete implementations.

6.1.2 Definitions: What Makes an Estimator "Double Robust"?

An estimator is said to be *double robust* if it yields a consistent estimate of the causal effect as long as either:

1. The model for the outcome given treatment and covariates, $\mathbb{E}[Y \mid A, X]$, is correctly specified, or
2. The model for the treatment assignment mechanism, i.e., the propensity score $e(X) = \mathbb{P}(A = 1 \mid X)$, is correctly specified.

In contrast to singly robust estimators—such as ordinary regression or IPW—double-robust methods incorporate information from both models, and the estimator remains valid if at least one is correctly specified.

More formally, suppose we are interested in estimating the average treatment effect (ATE):

$$\text{ATE} = \mathbb{E}[Y^1 - Y^0]$$

where Y^a denotes the potential outcome under treatment level $a \in \{0, 1\}$. A double-robust estimator of the ATE typically takes the form:

$$\hat{\theta}_{\text{DR}} = \frac{1}{n} \sum_{i=1}^{n} \left\{ \frac{A_i(Y_i - \hat{m}_1(X_i))}{\hat{e}(X_i)} - \frac{(1 - A_i)(Y_i - \hat{m}_0(X_i))}{1 - \hat{e}(X_i)} + \hat{m}_1(X_i) - \hat{m}_0(X_i) \right\}$$

where $\hat{e}(X_i)$ is the estimated propensity score, and $\hat{m}_a(X_i)$ is the predicted outcome under treatment a, i.e., $\hat{m}_a(X_i) = \mathbb{E}[Y \mid A = a, X = X_i]$.

If either the propensity score model $\hat{e}(X)$ or the outcome regression $\hat{m}_a(X)$ is correctly specified, the estimator $\hat{\theta}_{\text{DR}}$ is consistent for the true ATE.

6.1.3 When and Why to Use Double-Robust Estimators

Double-robust estimators are particularly useful when neither the outcome model nor the treatment model can be confidently specified. In applied settings, both models are often estimated using flexible techniques such as machine learning, which are prone to bias if overfit or miscalibrated. By combining two potentially misspecified models, double-robust estimators reduce the reliance on any single model being perfectly correct.

Moreover, these estimators naturally accommodate semiparametric modeling strategies. For example, one can use logistic regression for the propensity score and nonparametric regression for the outcome model, or vice versa. The result is a flexible and robust framework for estimating causal effects even under model uncertainty.

Another key benefit is the built-in structure for diagnostic checking. If both models are suspect, the double-robust estimator may still be biased, but it can help highlight when one model strongly dominates or conflicts with the other.

6.1.4 Connection to Consistency and Efficiency

Double-robust estimators are part of the broader class of semiparametric efficient estimators. If both the propensity score model and the outcome regression model are correctly specified, then double-robust estimators achieve the semiparametric efficiency bound. This means they have the lowest possible asymptotic variance among all regular and asymptotically linear estimators of the causal effect under the nonparametric model.

This efficiency property sets double-robust estimators apart from singly robust methods. For example, inverse probability weighting tends to have high variance, particularly when propensity scores are close to 0 or 1. Regression-based estimators may have low variance but can be severely biased if the regression model is incorrect. Double-robust methods strike a balance: they are consistent under weaker assumptions and can be more efficient when both models are well specified.

In summary, double-robust estimators are appealing in practice because they offer:

- Consistency under either a correctly specified treatment or outcome model,
- The potential for efficiency when both models are correct,
- Robustness to model misspecification, and
- Flexibility to incorporate machine learning.

We give a formal introduction to double robust estimation from an chronological perspective and explain the value of double robustness when using flexible data-adaptive methods for inverse probability weighting or regression adjustment introducing to one of the most novel double robust methods for causal inference i.e., Targeted Maximum Likelihood Estimation. Finally, we provide a comparison of all classical and more recent methods via a Monte Carlo simulation and discuss pro and cons of the new approaches and interesting ways to continue developing and improving causal inference.

6.2 Inverse Probability of Treatment Weighting with Regression Adjustment

6.2.1 Description of the Method

Combining inverse probability weighting (IPW) with outcome regression adjustment provides a straightforward way to construct a double-robust estimator. The idea is to use both models—the propensity score model and the outcome regression model—simultaneously to mitigate the risk of misspecification in either.

This approach is often used in practice because it is easy to implement using standard regression software, and it provides some protection against model misspecification while avoiding the more complex steps involved in targeted learning (see TMLE). In particular, it can be viewed as a preliminary or intermediate method that builds intuition for more advanced estimators like AIPW and TMLE.

The IPW + regression estimator also offers an intuitive decomposition: it reweights the residuals from the regression model using inverse probability weights and adds the difference in predicted means across treatment groups. This structure can be helpful for interpreting how and why the estimator works, and for understanding sources of bias and variability.

A common form of the estimator for the average treatment effect (ATE) is:

$$\hat{\theta}_{\text{DR}} = \frac{1}{n}\sum_{i=1}^{n}\left\{\frac{A_i(Y_i - \hat{m}_1(X_i))}{\hat{e}(X_i)} - \frac{(1-A_i)(Y_i - \hat{m}_0(X_i))}{1-\hat{e}(X_i)} + \hat{m}_1(X_i) - \hat{m}_0(X_i)\right\}$$

where $\hat{m}_a(X_i)$ is the predicted outcome under treatment level a, and $\hat{e}(X_i)$ is the estimated propensity score. The first two terms reweight residuals using the inverse probability weights, while the final term combines the predicted differences in outcomes.

6.2.2 Statistical Properties: Bias, Variance, and Efficiency

The estimator is consistent if either the propensity score model or the outcome regression model is correctly specified. When both models are correct, the estimator is semiparametrically efficient, achieving the lowest possible asymptotic variance under the nonparametric model defined by unconfoundedness.

However, when both models are misspecified, the estimator may be biased. In practice, even small violations of model assumptions can introduce bias, especially in finite samples. This reinforces the importance of using flexible models, performing diagnostics, and conducting sensitivity analyses.

Another important consideration is **variance inflation**. If the estimated propensity scores are close to 0 or 1, the corresponding weights become large, leading to instability in the estimator. This issue—known as near-violation of the positivity assumption—can be addressed through weight truncation, stabilized weights, or the use of more robust estimators like TMLE.

6.2.3 Implementation in R and Stata

Here is a basic implementation of the regression + IPW estimator:

Box 5.1: IPTW with regression adjustment (IPTW-RA)

6.2.4 R

```r
# Simulate data
set.seed(123)
n <- 1000
X <- rnorm(n)
A <- rbinom(n, 1, plogis(0.5 * X))
Y <- 2 * A + X + rnorm(n)
data <- data.frame(A, X, Y)
```

```r
# Step 1: Estimate propensity scores
e.model <- glm(A ~ X, family = binomial, data = data)
data$ehat <- predict(e.model, type = "response")

# Step 2: Estimate outcome models
m1.model <- lm(Y ~ X, data = subset(data, A == 1))
m0.model <- lm(Y ~ X, data = subset(data, A == 0))
data$m1hat <- predict(m1.model, newdata = data)
data$m0hat <- predict(m0.model, newdata = data)

# Step 3: Compute double-robust estimator
with(data, mean(
  A * (Y - m1hat) / ehat -
  (1 - A) * (Y - m0hat) / (1 - ehat) +
  m1hat - m0hat
))
```

6.2.5 Stata

```
clear all
set seed 123
set obs 1000
generate X = rnormal()
generate A = rbinomial(1, invlogit(0.5 * X))
generate Y = 2 * A + X + rnormal()

* Step 1: Estimate propensity scores
logit A X
predict double ehat, pr

* Step 2: Estimate outcome models
regress Y X if A == 1
predict double m1hat, xb
regress Y X if A == 0
predict double m0hat, xb

* Step 3: Compute double-robust estimator
generate double dr = A * (Y - m1hat) / ehat - ///
                     (1 - A) * (Y - m0hat) / (1 - ehat) + ///
                     m1hat - m0hat
summarize dr
```

6.2.6 When It Is Used in Practice

The regression + IPW estimator is widely used in epidemiology, economics, and health services research, especially when researchers want to guard against model misspecification. It is particularly attractive when using parametric models for one nuisance parameter (e.g., the propensity score) and nonparametric or flexible models (e.g., machine learning) for the other. In such cases, double-robustness offers a practical compromise between robustness and interpretability.

This approach is also used in high-dimensional settings, such as genomic studies or electronic health records, where traditional model diagnostics are less reliable and model misspecification is more likely. In such contexts, ensemble learners (e.g., Super Learner) can be used to estimate nuisance functions, and the double-robust estimator remains valid as long as at least one learner captures the truth.

Additionally, the regression + IPW estimator forms the foundation for more advanced estimators such as AIPW and TMLE. Understanding its structure provides essential insight into the efficient influence function framework and semiparametric causal inference more broadly.

In summary, IPW + outcome regression is a powerful, flexible, and interpretable tool in the causal inference toolbox, particularly useful in observational studies where untestable assumptions are the norm and robustness is paramount.

6.3 Augmented Inverse Probability of Treatment Weighting

Doubly robust (DR) estimators combine both outcome regression and propensity score-based methods to estimate causal effects. The key advantage of DR estimators is that they yield consistent estimates of treatment effects if either the model for the outcome or the model for the treatment assignment (i.e., the propensity score) is correctly specified. This property provides an additional layer of protection against model misspecification.

6.3.1 Augmented IPTW (AIPTW)

The most common doubly robust estimator is the Augmented Inverse Probability of Treatment Weighting (AIPTW) estimator. It augments the IPTW estimator with a regression-based prediction for the outcome. Let Y_i be the observed outcome, $A_i \in \{0, 1\}$ the treatment, $e(W_i)$ the estimated propensity score, and $\hat{\mu}_a(W_i) = E[Y \mid A = a, W_i]$ the predicted outcome under treatment a.

The AIPTW estimator for the average treatment effect (ATE) is:

$$\widehat{ATE}_{\text{AIPTW}} = \frac{1}{n} \sum_{i=1}^{n} \left[\frac{A_i Y_i}{\hat{e}(W_i)} - \frac{(A_i - \hat{e}(W_i))\hat{\mu}_1(W_i)}{\hat{e}(W_i)} - \left(\frac{(1 - A_i)Y_i}{1 - \hat{e}(W_i)} - \frac{(A_i - \hat{e}(W_i))\hat{\mu}_0(W_i)}{1 - \hat{e}(W_i)} \right) \right].$$

This estimator consists of two broad components that correspond to the treated and untreated groups, respectively. The term $\frac{A_i Y_i}{\hat{e}(W_i)}$ represents a weighted average of outcomes among the treated individuals, where each individual is weighted by the inverse of their estimated propensity score. This is a standard component of the IPTW estimator, capturing the expected potential outcome under treatment. The term $\frac{(A_i - \hat{e}(W_i))\hat{\mu}_1(W_i)}{\hat{e}(W_i)}$ acts as a correction, adjusting the treatment group's contribution based on the discrepancy between the actual treatment received and the estimated probability of treatment, scaled by the predicted outcome from the outcome regression model under treatment.

The second part of the expression inside the summation pertains to the control group. The term $\frac{(1-A_i)Y_i}{1-\hat{e}(W_i)}$ is the IPTW contribution for control individuals, with weights equal to the inverse of one minus the estimated propensity score. This is then adjusted using $\frac{(A_i - \hat{e}(W_i))\hat{\mu}_0(W_i)}{1-\hat{e}(W_i)}$, which again serves as a correction based on the difference between actual and predicted treatment assignment, scaled by the predicted outcome under control.

Taken together, this expression combines inverse probability weighting with model-based outcome predictions to reduce variance and mitigate bias. The estimator remains consistent provided either the propensity score model or the outcome model is correctly specified, offering what is referred to as "double robustness."

Taking a deeper look into AIPTW. The Augmented Inverse Probability of Treatment Weighting (AIPTW) estimator extends the standard IPTW framework by incorporating an additional term designed to correct for potential misspecification of the treatment model. If the model for the treatment assignment is correctly specified, this augmentation term contributes negligible bias as sample size increases, resulting in an estimator that simplifies to IPTW. This makes AIPTW more efficient in large samples compared to IPTW. Nonetheless, like IPTW, AIPTW suffers from instability when the estimated propensity scores are close to 0 or 1, which indicates a violation of the positivity assumption.

AIPTW uses information from both the treatment model and the outcome model. The augmentation term, which has expectation zero under correct model specification, depends on the estimated propensity score and the predicted outcomes from a regression model. Because of this dual reliance, AIPTW achieves consistency for the average treatment effect (ATE) as long as at least one of the two models—the treatment mechanism or the outcome regression—is correctly specified. This property is the foundation of its double-robustness (Bang & Robins, 2005; Robins et al., 1994b; Tsiatis et al., 2007).

The IPTW estimator for the expected potential outcome under treatment level a, denoted μ_a, is given by:

$$\hat{\mu}_a = \mathbb{E}\left(\frac{I(A=a)}{g(A \mid W)}\, Y\right),$$

where $I(\cdot)$ is the indicator function and $g(A \mid W)$ is the estimated propensity score.

The IPTW estimator can be viewed as solving the estimating equation:

$$\frac{1}{n}\sum_{i=1}^{n}\left(\frac{I(A_i = a)Y_i}{g(A_i \mid W_i)} - \mu_a\right) = 0.$$

To improve this estimator, we can introduce a mean-zero augmentation term that adjusts for residual differences in the outcome model:

$$\frac{I(A = a) - g(A = a \mid W)}{g(A = a \mid W)} E(Y \mid A = a, W).$$

Incorporating this into the estimating equation gives the AIPTW formulation:

$$\mathbb{E}\left(\frac{I(A = a)Y}{g(A = a \mid W)} - \left(\frac{I(A = a) - g(A = a \mid W)}{g(A = a \mid W)}\right) E(Y \mid A = a, W)\right) - \mu_a = 0.$$

By rearranging, it becomes evident that AIPTW combines both an outcome regression and a weighting adjustment:

$$\begin{aligned}
&\underbrace{\frac{1}{n}\sum_{i=1}^{n}\left(E(Y_i \mid A_i = 1, W_i) - E(Y_i \mid A_i = 0, W_i)\right)}_{\text{Outcome regression component}} + \\
&\underbrace{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{A_i[Y_i - E(Y_i \mid A_i = 1, W_i)]}{g(A_i = 1 \mid W_i)} - \frac{(1 - A_i)[Y_i - E(Y_i \mid A_i = 0, W_i)]}{g(A_i = 0 \mid W_i)}\right)}_{\text{Augmentation term with mean zero}}.
\end{aligned} \tag{6.1}$$

The estimated ATE using AIPTW is:

$$\text{AIPTW-ATE} = \mu_1 - \mu_0,$$

where each potential outcome mean is given by:

$$\begin{aligned}
\mu_1 &= \frac{1}{n}\sum_{i=1}^{n}\left(E(Y_i \mid A_i = 1, W_i) + \frac{A_i[Y_i - E(Y_i \mid A_i = 1, W_i)]}{g(A_i = 1 \mid W_i)}\right), \\
\mu_0 &= \frac{1}{n}\sum_{i=1}^{n}\left(E(Y_i \mid A_i = 0, W_i) + \frac{(1 - A_i)[Y_i - E(Y_i \mid A_i = 0, W_i)]}{g(A_i = 0 \mid W_i)}\right).
\end{aligned}$$

The second component in Equation 6.1 reflects the residuals of the outcome regression model, scaled by the inverse probability weights. These residual terms have expectation zero under correct specification, making them ideal for correcting potential bias from misspecification in one of the models (Kennedy, 2016). When the outcome model is correct, the AIPTW simplifies to the g-formula estimator. Conversely, if the treatment model is correct, the estimator collapses to the standard IPTW form (Bang & Robins, 2005; Daniel, 2018).

Box 5.2: AIPTW with linear models

6.3.2 R

```r
# Estimate propensity scores
data$ps <- glm(A ~ W1 + W2, data = data, family = binomial)$fitted.values

# Estimate outcome models
mu1_model <- lm(Y ~ W1 + W2, data = subset(data, A == 1))
mu0_model <- lm(Y ~ W1 + W2, data = subset(data, A == 0))
data$mu1 <- predict(mu1_model, newdata = data)
data$mu0 <- predict(mu0_model, newdata = data)

# AIPTW estimate of ATE
term1 <- data$A * (data$Y - data$mu1) / data$ps + data$mu1
term0 <- (1 - data$A) * (data$Y - data$mu0) / (1 - data$ps) + data$mu0
ate_aiptw <- mean(term1 - term0)
ate_aiptw
```

6.3.3 Stata

```
* Estimate propensity scores
logit A W1 W2
predict double ps, pr

* Estimate outcome models
regress Y W1 W2 if A == 1
predict double mu1, xb
regress Y W1 W2 if A == 0
predict double mu0, xb

* AIPTW estimate of ATE
generate double term1 = A * (Y - mu1) / ps + mu1
generate double term0 = (1 - A) * (Y - mu0) / (1 - ps) + mu0
generate double ate = term1 - term0
summarize ate
```

6.3.4 Intuition and Advantages

The intuition behind the AIPTW estimator is based on combining two approaches: weighting and imputation. The weighting component (as in IPTW) creates a pseudo-population where the distribution of covariates is independent of treatment assignment, while the imputation

component uses an outcome model to predict what would have happened under each treatment arm. If the outcome model is correct, the predicted values can help fill in the missing potential outcomes. If the weighting model is correct, the reweighted sample gives unbiased comparisons of outcomes between treatment groups. Because the AIPTW estimator uses both sources of information, it offers protection against misspecification of either model. This makes it more robust than estimators that rely exclusively on one model.

One important advantage of AIPTW is that it improves efficiency compared to IPTW or regression alone. The estimator also offers a bias correction mechanism that can compensate for moderate model misspecification. When at least one model is correct, the bias converges to zero as the sample size increases, preserving consistency.

6.3.5 Limitations and Forward Link to TMLE

Despite its theoretical strengths, the AIPTW estimator has several limitations in practice. In finite samples, its performance can be unstable, particularly when there is limited overlap in the propensity score distributions between treated and control groups. Extreme values of the weights can lead to high variance or numerical instability. Additionally, if both the propensity score and outcome models are misspecified, the AIPTW estimator can perform worse than estimators that rely on a single model, a phenomenon known as bias amplification.

Traditional implementations of AIPTW are based on plug-in estimates and do not directly optimize the estimator for a target causal parameter. This lack of targeting has motivated the development of newer doubly robust methods, such as Targeted Maximum Likelihood Estimation (TMLE). TMLE incorporates targeted updates to the outcome model and guarantees that the final estimator solves a specific estimating equation for the target parameter. It also retains the double robustness property while offering improved statistical properties such as asymptotic efficiency.

TMLE, discussed in the following section, addresses these limitations through targeted updates to the outcome model, guaranteeing that the final estimator solves a specific estimating equation for the target parameter while retaining double robustness and offering improved statistical properties such as asymptotic efficiency.

6.4 Targeted Maximum Likelihood Estimation

The targeted learning framework provides a structured approach to estimate causal effects with greater accuracy and robustness. Unlike traditional estimators, which may be biased if certain assumptions are violated, targeted learning aims to reduce bias and maximize efficiency by combining flexible machine learning methods with causal inference principles.

6.4.1 Motivation for targeted learning

Traditional causal estimators, such as propensity score methods and inverse probability of treatment weighting, are often single-robust—they rely on the correct specification of either the treatment or outcome model to produce unbiased estimates. If that model is misspecified, the resulting estimates can be biased. These methods have limited protection against model misspecification, and their validity breaks down when both models are incorrect. Double-robust estimators improve upon this by remaining consistent if either the treatment or outcome model is correctly specified. Further gains in bias reduction and efficiency can be achieved by using flexible, data-adaptive machine learning methods that avoid strict parametric assumptions.

Targeted Maximum Likelihood Estimation (TMLE) is a plug-in, semi-parametric, double-robust estimator that incorporates machine learning to refine an initial estimate, targeting it toward the parameter of interest. TMLE has been widely described in both theoretical and practical tutorials.(Gruber & Laan, 2012; Gruber & Laan, 2009; Gruber & Laan, 2011; Luque-Fernandez, Schomaker, et al., 2018; Schuler & Rose, 2017; **vanderLaan2011?**) In simulation and applied studies, TMLE has shown lower bias than other double-robust estimators such as IPTW-RA and AIPTW, especially in small samples.(Luque-Fernandez, Belot, et al., 2018; **vanderLaan2011?**) Although asymptotically equivalent to AIPTW, TMLE generally performs better in finite samples and is often used in combination with ensemble machine learning to further mitigate model specification issues.

> The Substitution Property: Why "Plug-In" Matters
>
> TMLE is a **substitution** (or **plug-in**) estimator: it estimates the target parameter by plugging the updated outcome predictions into the same formula that defines the parameter. This means TMLE estimates *always stay within the bounds of the original outcome variable* — a risk difference cannot exceed $[-1, 1]$, and a predicted probability cannot fall outside $[0, 1]$.
> This property, which follows from TMLE being a plug-in estimator, contrasts with **one-step estimators** like AIPW, which add a correction term to an initial estimate and can sometimes yield final estimates outside the outcome scale (e.g., a risk difference of 1.3 or -0.04 for a binary outcome). The substitution property is particularly valuable for binary or bounded outcomes, where out-of-bounds estimates are scientifically meaningless. For a deeper discussion of why TMLE's construction guarantees this property, see Luque-Fernandez, Belot, et al. (2018) and the companion tutorials referenced in the Further Reading section below.

The targeted learning framework was designed to overcome limitations of traditional estimators. It ensures double robustness and facilitates the integration of machine learning to model complex, high-dimensional relationships. The central goal is to produce an accurate and efficient estimate of a causal parameter—such as the average treatment effect—by aligning the estimation process with that specific target. A key innovation is the targeting step, which

updates the initial estimate using information from the treatment mechanism. This is guided by the efficient influence curve, ensuring that the final estimator is not only unbiased but also has minimal variance given the observed data.

6.4.1.1 Goal of targeted learning

Targeted learning is focused on obtaining the most accurate estimate possible for a specific causal parameter by reducing bias and optimizing statistical efficiency. A central component of this approach is the targeting step, which adjusts initial estimates to correct for residual bias, thereby bringing the estimate closer to the true value of the target parameter. In addition, targeted learning leverages efficient influence curves to guide estimation in a way that minimizes variance, making the resulting estimators as close to statistically optimal as possible and enhancing the reliability and precision of the estimates.

6.4.2 TMLE step-by-step guide

TMLE has six main steps:

1. Initial prediction of the outcome
2. Predict the probability of treatment
3. Calculate the fluctuation parameter
4. Update the initial prediction of the outcome
5. Compute the estimand of interest
6. Calculate the standard errors for confidence intervals and p-values

As an example, suppose we are interested in estimating the effect of A (a binary treatment variable) on Y (a binary outcome variable) while adjusting for some confounders W (a vector of binary and continuous variables). Below is the simulated data. The steps for TMLE are given after.

Box 5.3: Loading and preparing RHC data for TMLE

6.4.3 R

```
library(readr)
library(knitr)

data <- read.csv("https://raw.githubusercontent.com/migariane/TutorialComputationalCausalInf

# Define the outcome (Y), exposure (A), confounder (C), and confounders (W)
# Y
```

```r
data$Y  <- as.numeric(data$death_d30); Y <- data$Y
# A - Treated = 1, Not treated = 0
data$A  <- as.numeric(as.factor(data$rhc))-1; A <- data$A
# C - Female = 0, Male = 1
data$C  <- as.numeric(as.factor(data$sex))-1; C <- data$C
# w1 (age)
data$w1 <- as.numeric(data$age); w1 <- data$w1
# w2 (education)
data$w2 <- as.numeric(data$edu); w2 <- data$w2
# w3 (race) - White = 2, Other = 1, Black = 0
data$w3 <- as.numeric(as.factor(data$race))-1; w3 <- data$w3
# w4 (carcinoma) - Yes = 2, No = 1, Metastatic = 0
data$w4 <- as.numeric(as.factor(data$carcinoma)); w4 <- data$w4

# Create a data set
data2 <- as.data.frame(Y); data2$A <- A; data2$C <- C
data2$w1 <- w1; data2$w2 <- w2; data2$w3 <- w3; data2$w4 <- w4
```

6.4.4 Stata

```
* Download and prepare RHC data
* Data available at: https://github.com/migariane/TutorialComputationalCausalInferenceEstimat
use "rhc.dta", clear

* Define variables
global Y death_d30                          // Outcome: 30-day mortality
global A rhc                                // Treatment: RHC
global W i.sex c.age c.edu i.race i.ca      // Confounders
```

6.4.4.1 Step 1: Predict the initial outcome

The first step is to estimate the expected value of the outcome using information on the treatment and confounders. This is defined using a function Q of A and $\mathbf{W}$ to obtain the conditional expectation of Y:

$$Q^0(A, \mathbf{W}) = E[Y|A, \mathbf{W}]$$

We can use logistic regression to model the conditional expectation of Y:

$$E[Y|A, \mathbf{W}] = \beta_0 + \beta_1 A + \beta_w^T \mathbf{W}$$

Box 5.4: Step 1: Initial prediction of the outcome (G-computation)

6.4.5 R

```
# Step 1
Gcomp <- glm(Y ~ A + C + w1 + w2 + as.factor(w3) + as.factor(w4), family="binomial", data=dat
```

6.4.6 Stata

```
* Step 1: prediction model for the outcome Q0 (g-computation)
glm Y A W, fam(binomial)
predict double QAW_0, mu
gen aa = A
replace A = 0
predict double Q0W_0, mu
replace A = 1
predict double Q1W_0, mu
replace A = aa
drop aa
```

To predict the potential outcomes, we first need to specify two variables, one where everyone receives the treatment ($A = 1$) and another where everyone does not receive the treatment ($A = 0$). Taking predictions of the outcome for each observation using these two variables provides us with the potential outcomes $Y(1)$ and $Y(0)$.

$$Q^0(A = 1, \mathbf{W}) = E[Y|A = 1, \mathbf{W}] = expit(\hat{\beta}_0 + \hat{\beta}_1(A = 1) + \hat{\beta}_w^T\mathbf{W})$$

$$Q^0(A = 0, \mathbf{W}) = E[Y|A = 0, \mathbf{W}] = expit(\hat{\beta}_0 + \hat{\beta}_1(A = 0) + \hat{\beta}_w^T\mathbf{W})$$

Box 5.5: Predicting potential outcomes under treatment and control

6.4.7 R

```
# Predict Q for A, A=1, and A=0
QAW <- predict(Gcomp)
Q1W = predict(Gcomp, newdata=data.frame(A = 1, data2[,c("C","w1","w2","w3","w4")]))
Q0W = predict(Gcomp, newdata=data.frame(A = 0, data2[,c("C","w1","w2","w3","w4")]))
```

6.4.8 Stata

```
* Q to logit scale
gen logQAW = log(QAW / (1 - QAW))
gen logQ1W = log(Q1W / (1 - Q1W))
gen logQ0W = log(Q0W / (1 - Q0W))
```

Notice that the potential outcome for $A = 1$ will be equivalent to the observed outcome ($E(Y|A = 1, \mathbf{W})$. This is part of the consistency assumption.

If we were to take the difference between $\widehat{E}(Y|A = 1, \mathbf{W})$ and $\widehat{E}(Y|A = 0, \mathbf{W})$, then we would obtain an estimate of the average treatment effect (ATE). You may have noticed that this is the same as standardisation, g-formula estimation, or G-computation.

However, we can do better than this. TMLE includes a targeting step that adjusts the initial prediction of the outcome to align it with the estimand of interest (e.g., the ATE). The initial prediction of the outcome (Q^0) may be a good overall estimate but may not be tailored to the causal quantity. The targeting step refines this prediction to better capture the relevant features of the data for estimating the parameter of interest. The targeting step involves defining "clever covariates" (constructed using the propensity score) adjusting the initial prediction to correct for any bias in the estimation of the parameter of interest. This is especially important when the initial model is misspecified. Through the targeting step, TMLE achieves double robustness, meaning it produces consistent estimates if either the initial outcome model (Q^0) or the propensity score model is correctly specified. The targeting step ensures that the estimator uses both models effectively, leveraging their complementary strengths. The targeting step encompasses Steps 2, 3, and 4 of the six main steps. Together, these steps create a fluctuation parameter that is used to update the initial prediction of the outcome.

An important note here is that even if we used machine learning algorithms to define the outcome model, we will still need the targeting step. This is because in Step 1 we are **predicting** potential outcomes, we are not yet **estimating** causal effects. The objective of machine learning (i.e., logistic regression in a simple case) is to minimise prediction error on the outcome variable, not to balance confounders or align with the causal estimand. Machine learning models predict the outcome conditional on covariates, but they do not naturally adjust for treatment assignment mechanisms or ensure consistency with the target parameter. The targeting step adjusts the outcome predictions to account for the propensity score and other aspects of the treatment mechanism, ensuring the final estimate is aligned with the causal parameter. The machine learning models may still leave residual confounding or fail to focus on the specific population and treatment comparisons relevant to the causal parameter.

The question is now "how do we refine our initial prediction of the outcome (and the estimate of the target parameter)"? In semiparametric estimation theory, there is a fundamental concept called the efficient influence curve (EIC). It represents the residual variation in the target parameter that remains after accounting for all available information in the model. The EIC is a function that characterises the amount of information a particular observation provides

about the target parameter while accounting for the statistical model's constraints. The EIC is the "blueprint" for achieving efficient, unbiased estimation of a causal parameter. TMLE updates the initial outcome model so that the final estimate solves the EIC equation, ensuring the estimator is efficient and unbiased. The EIC is:

$$EIC = \left(\frac{A}{P(A=1|\mathbf{W})} - \frac{1-A}{P(A=0|\mathbf{W})}\right)[Y - E(Y|A,\mathbf{W})] + E(Y|A=1,\mathbf{W}) + E(Y|A=0,\mathbf{W}) - \psi$$

where ψ is our estimate of the ATE. The EIC can be evaluated from the observed data as

$$\widehat{EIC} = \left(\frac{A}{\hat{g}(1,\mathbf{W})} - \frac{1-A}{\hat{g}(0,\mathbf{W})}\right)[Y - Q^1(A,\mathbf{W})] + Q^1(1,\mathbf{W}) + Q^1(0,\mathbf{W}) - \widehat{ATE}$$

Note that $Q^1(.)$is the update of the initial prediction of$Q^0(.)$, and $\hat{g}(.)$ is the propensity score.

Using the EIC to achieve an unbiased estimation of the causal parameter (e.g., ATE), we need to obtain the propensity score and update the initial prediction of the outcome. We cover this over the next few steps.

6.4.8.1 Step 2: Predict the probability of treatment

Steps 2, 3, and 4 encompass the targeting step used to update the initial prediction of the outcome. The first part of the targeting step is to estimate the probability of treatment, given the confounders:

$$g(A,W) = Pr(A=1|\mathbf{W})$$

Again, we could use logistic regression to define the propensity score model:

$$\hat{g}(A=1,\mathbf{W}) \quad = \quad \widehat{E}[A=1|\mathbf{W}] \quad = \quad expit(\hat{\alpha}_0 + \hat{\alpha}_1^T\mathbf{W})$$

Box 5.6: Step 2: Estimation of the propensity score

6.4.9 R

```r
# Step 2 estimation of the propensity score (ps)
psm <- glm(A ~ C + w1 + w2 + as.factor(w3) + as.factor(w4), family = binomial, data=data2)
gW = predict(psm, type = "response")
g1W = (1 / gW)
g0W = (-1 / (1-gW))
```

6.4.10 Stata

```
* Step 2: prediction model for the treatment g0 (IPTW)
glm A W, fam(binomial)
predict gw, mu
gen double H1W = A / gw
gen double H0W = (1 - A) / (1 - gw)
```

6.4.10.1 Step 3: Calculate the fluctuation parameter

The next step of the targeting step is to calculate clever covariates and the fluctuation parameter. These covariates guide the targeting step by indicating how much weight each observation contributes to correcting the initial prediction.

The clever covariates are calculated as:

$$H(A=a,\mathbf{W}) \quad = \quad \frac{A}{\hat{g}(A=1,\mathbf{W})} - \frac{1-A}{\hat{g}(A=0,\mathbf{W})}$$

When $A=1$, the right hand side will be $\frac{1}{\hat{g}(A=1,\mathbf{W})}$. When $A=0$, the right hand side will be $\frac{-1}{\hat{g}(A=0,\mathbf{W})}$. You may notice that the clever covariates are of the same form as inverse probability of treatment weights.

Box 5.7: Step 3: Computing the clever covariate H(A,W)

6.4.11 R

```
# Step 3 computation of H and estimation of epsilon
HAW <- (data2$A / gW -(1-data2$A) / (1 - gW))
H1W = (1/gW)
H0W = (-1 / (1 - gW))
```

6.4.12 Stata

```
* Step 3: Computing the clever covariate H(A,W) and estimating epsilon (MLE)
glm Y H1W H0W, fam(binomial) offset(logQAW) noconstant
mat a = e(b)
gen eps1 = a[1,1]
gen eps2 = a[1,2]
```

The **fluctuation parameter** (ϵ) is a small adjustment applied to the initial outcome model ($Q^0(A, \mathbf{W})$) to correct residual bias and ensure that the updated model (Q^1) aligns with the EIC. It does this by incorporating information from the clever covariate ($H(A, \mathbf{W})$) and the observed outcomes (Y). The fluctuation parameter is estimated by solving an estimating equation, specifically a score equation derived from the EIC. The estimating equation ensures that the updated outcome model satisfies the EIC's property:

$$E[D(Y, A, W; Q, g)] = 0$$

where $D(.)$ is the EIC, which includes the clever covariate $H(A, W)$.

In practice, the score equation used to estimate ϵ is:

$$\sum_i H(A_i, W_i) \times (Y_i - Q^1(A_i, W_i)) = 0$$

where $Q^1(A_i, W_i)$ is the updated (targeted) prediction. You may be asking "why do we need to solve an estimating equation?". By solving the score equation, TMLE ensures that the estimator $\widehat{\psi}$ satisfies $E[D] = 0$, meaning the bias has been corrected.

The form of the update from Q^0 to Q^1 depends on the outcome type:

- **For continuous outcomes**, a linear (additive) fluctuation model can be used:

$$Q^1(A, W) = Q^0(A, W) + \epsilon \cdot H(A, W)$$

- **For binary outcomes** (our case), the fluctuation model is logistic, ensuring that the updated predictions remain within $[0, 1]$:

$$\text{logit}(Q^1(A, W)) = \text{logit}(Q^0(A, W)) + \epsilon \cdot H(A, W)$$

Under the logistic fluctuation model, the estimating equation becomes:

$$\sum_i H(A_i, W_i) \times (Y_i - \text{expit}\left(\text{logit}(Q^0(A_i, W_i)) + \epsilon \cdot H(A_i, W_i)\right)) = 0$$

If we take the fluctuation model above, you will see that its form is very similar to the form of a logistic regression model, such as $logit(E[Y|X]) = \beta_0 + \beta_1 X$. The difference is that in the right hand side of our fluctuation model the "intercept" ($logit(Q^0(A, W))$) is not a constant value like β_0; it is a vector of values of the initial prediction of the outcome. Thus, instead of a constant-value intercept, we use $logit(Q^0(A, W))$ as an offset (a fixed intercept) in a logistic regression model. We can now solve our estimating equation for the EIC by using a logistic regression model with the observed outcome Y as the outcome, $logit(\widehat{Q}^0(A, W))$ as an offset, and one covariate, $H(A, W)$. The coefficient for the one covariate provides us with an estimate of the fluctuation parameter (ϵ).

Box 5.8: Step 3b: Estimating epsilon via logistic regression with offset

6.4.13 R

```
epsilon <- coef(glm(data2$Y ~ -1 + HAW + offset(QAW), family = "binomial"))
```

6.4.14 Stata

```
* (Epsilon is estimated simultaneously with clever covariate in Box 5.7 Stata)
* The glm with offset and H1W/H0W produces eps1 and eps2
display eps1
display eps2
```

6.4.14.1 Step 4: Update the initial outcome

We now have everything we need to update the initial outcome. Referring back to the fluctuation model, we used the logit scale to solve the estimating equation for the EIC. Ideally, we would like the updated prediction of the outcome to be on the true outcome scale, We can use the inverse logit transformation (i.e., use *expit*):

$$Q^1(A, \mathbf{W}) = expit(logit(Q^0(A, \mathbf{W})) + \epsilon \cdot H(A, \mathbf{W}))$$

From this model, we can obtain three variables:

1. $Q^1(A, \mathbf{W})$: update of the expected outcome of all observations, **given the treatment they actually received** and their baseline confounders.
2. $Q^1(A = 1, \mathbf{W})$: update of the expected outcome, **conditional on receiving the treatment** and their baseline confounders.
3. $Q^1(A = 0, \mathbf{W})$: update of the expected outcome, **conditional on receiving the control** and their baseline confounders.

Box 5.9: Step 4: Updating initial outcome predictions (targeting step)

6.4.15 R

```
# Step 4 update from Q0 to Q1 ATE
Q1W_1 <- plogis(Q1W + epsilon * H1W)
Q0W_1 <- plogis(Q0W + epsilon * H0W)
```

6.4.16 Stata

```
* Step 4: update from Q0 to Q1
gen double Q1W_1 = exp(eps1 / gw + logQ1W) / (1 + exp(eps1 / gw + logQ1W))
gen double Q0W_1 = exp(eps2 / (1 - gw) + logQ0W) / (1 + exp(eps2 / (1 - gw) + logQ0W))
```

6.4.16.1 Step 5: Compute the estimand of interest

Now that we have updated predictions of the outcome, we can compute the ATE. The ATE (the causal estimand) in this case is evaluated using the risk difference (the statistical estimand). The risk difference is the average of the difference in the updated outcome estimates:

$$\widehat{ATE} = \widehat{\psi} = \frac{1}{N}\sum_{i=1}^{N}\left(\widehat{Q}_i^1(A=1,\mathbf{W}) - \widehat{Q}_i^1(A=0,\mathbf{W})\right)$$

Box 5.10: Step 5: Computing the targeted ATE estimate

6.4.17 R

```
# Step 5 targeted estimate of the ATE
ATE <- mean(Q1W_1 - Q0W_1); ATE
```

6.4.18 Stata

```
* Step 5: Targeted estimate of the ATE
gen ATE = (Q1W_1 - Q0W_1)
summ ATE
global ATE = r(mean)
display "ATE: " %05.4f $ATE
drop ATE
```

6.4.18.1 Step 6: Calculate the standard errors for confidence intervals and p-values

To calculate confidence intervals, we can refer back to the EIC:

$$\widehat{EIC} = \left(\frac{A}{\hat{g}(1,\mathbf{W})} - \frac{1-A}{\hat{g}(0,\mathbf{W})}\right)[Y - Q^1(A,\mathbf{W})] + Q^1(1,\mathbf{W}) + Q^1(0,\mathbf{W}) - \widehat{ATE}$$

Notice that we use the three variables that we evaluated in step 4: $Q^1(A,\mathbf{W})$, $Q^1(1,\mathbf{W})$, and $Q^1(0,\mathbf{W})$.

The EIC informs us how much each observation influences the estimate of the target parameter. It is evaluated for each observation, and using the resulting vector, we can estimate the standard error for the target parameter:

$$\hat{\sigma}_{EIC} = \widehat{SE}_{EIC} = \sqrt{\frac{\widehat{Var}(\widehat{EIC})}{n}}$$

where $\widehat{Var}(\widehat{EIC})$ represents the sample variance of the EIC.

Box 5.11: Step 6: Computing the efficient influence curve for standard errors

6.4.19 R

```
# Step 6 statistical inference
d1 <- ((data2$A * (Y - Q1W_1)/gW)) + Q1W_1 - mean(Q1W_1)
d0 <- ((1 - data2$A) * (Y - Q0W_1)/(1 - gW)) + Q0W_1 - mean(Q0W_1)
EIC <- d1 - d0
n <- nrow(data2)
varEIC <- var(EIC)/n
```

6.4.20 Stata

```
* Step 6: Statistical inference via the efficient influence function
qui sum(Q1W_1)
gen EY1tmle = r(mean)
qui sum(Q0W_1)
gen EY0tmle = r(mean)

gen d1 = ((A * (Y - Q1W_1)/gw)) + Q1W_1 - EY1tmle
gen d0 = ((1 - A) * (Y - Q0W_1)/(1 - gw)) + Q0W_1 - EY0tmle

gen IF = d1 - d0
qui sum IF
gen varIF = r(Var) / r(N)
```

95% confidence intervals are calculated in the conventional way:

$$95\%CI: \quad \widehat{ATE} \pm 1.96(\widehat{SE}_{EIC})$$

Box 5.12: Step 6b: Computing 95% Wald confidence intervals

6.4.21 R

```
LCI <- ATE - 1.96*sqrt(varEIC)
UCI <- ATE + 1.96*sqrt(varEIC)
cbind(ATE, LCI, UCI)
```

6.4.22 Stata

```
global LCI = $ATE - 1.96*sqrt(varIF)
global UCI = $ATE + 1.96*sqrt(varIF)
display "ATE:" %05.4f $ATE _col(15) "95%CI: " %05.4f $LCI "," %05.4f $UCI
```

We obtain an estimate of 0.0837, corresponding to a risk difference of 8.37% (95% CI: 5.85 - 10.90). This is interpreted as "the risk of death at 30 days is 8.37% higher if everyone was treated with RHC compared to if no one was treated with RHC". Note that our interpretation is a comparison of two hypothetical worlds.

How Much Does the Targeting Step Adjust the Estimate?

To appreciate what the targeting step accomplishes, it is helpful to compare the TMLE estimate with the **untargeted G-computation estimate** obtained directly from Step 1 — that is, $\frac{1}{N}\sum(\widehat{Q}^0(1,W_i) - \widehat{Q}^0(0,W_i))$ using the initial outcome predictions without any fluctuation update.

In the RHC example, the untargeted G-computation estimate differs from the TMLE estimate, though the difference is modest because the treatment groups are relatively well-balanced on observed covariates. In datasets with stronger confounding or more complex outcome–treatment relationships, the gap between the untargeted and targeted estimates can be substantially larger. The targeting step is what ensures the final estimator solves the efficient influence curve equation — moving from a pure prediction task (Step 1) to valid causal estimation.

A Note on the Targeting Step

The targeting step — fitting a logistic regression with `offset(qlogis(Q_A))` and the clever covariate `H_A` as the sole predictor — is **easy to code but difficult to understand** without a background in semiparametric theory. The logistic form has nothing to do with the outcome being binary; it happens to be the right functional form for solving the estimating equation derived from the efficient influence function. Readers encountering this step for the first time should not be discouraged if the rationale is not immediately clear — it typically takes several exposures to the material before the concepts begin to settle. The important practical takeaway is that this step adjusts

> the initial outcome predictions *just enough* to eliminate first-order bias for the target parameter, while preserving the flexibility of the initial machine learning fits.

6.4.22.1 Automating the TMLE process

The TMLE procedure can be automated using dedicated software packages. Below we present the Stata implementation using the `eltmle` package followed by the R implementation using the `tmle` package.

Stata implementation. The `eltmle` Stata package (available at github.com/migariane/eltmle) provides a complete implementation of TMLE with Super Learner integration for Stata users.

Box 5.13: Installing and using the eltmle Stata package

6.4.23 Stata

```
* Install eltmle (if not already installed)
* ssc install eltmle
* github install migariane/eltmle

* Standard TMLE
eltmle Y A W, tmle

* Check covariate balance after TMLE weighting
eltmle Y A W, tmle bal
```

Box 5.14: Cross-validated TMLE with eltmle (cveltmle)

6.4.24 Stata

```
* Cross-validated TMLE for improved small-sample performance
* Use cveltmle without the standard tmle option
eltmle Y A W, cveltmle
```

R implementation. The `tmle` package uses Super Learner (see Section 6.4.30), which is a library of machine learning algorithms for defining the outcome (Q.SL.library) and exposure (g.SL.library) models. This requires us to first define the seed (set.seed(1)). We then create a

data set called w that contains the set of confounders, which are used to define the parameters for the exposure model. Finally, we run the 'tmle' function to conduct TMLE. Since we are using a large data set, this will take a couple of minutes to run.

Box 5.15: TMLE using the tmle R package with Super Learner

6.4.25 R

```
set.seed(1)

install.packages('tmle')
library(tmle)

w <- subset(data, select=c(C, w1, w2, w3, w4))

fittmle <- tmle(data$Y, data$A, W=w, family="binomial", Q.SL.library = c("SL.glm", "SL.glm.i

fittmle
```

From the 'tmle' function, we obtain an estimate for the ATE of 0.0848 (ATE: 8.48%, 95% CI: 5.97, 10.99). This is very close to the estimate we obtained by hand, suggesting that the functional form of the outcome and exposure models that we defined by hand are close to those that are defined within the SuperLearner (see Section 6.4.30).

In general, the results obtained by hand and the results obtained using the 'tmle' package will not be this close. Looking back at the distribution of the covariates (w) within each level of the treatment variable shows that the treatment groups are close to being balanced. When the covariates are not balanced, the SuperLearner will help with obtaining the best-fitting model for predicting the outcome and the exposure.

6.4.26 Mathematical Foundations of TMLE

The TMLE procedure is grounded in semiparametric efficiency theory. This section provides the key mathematical results underlying the method. A more detailed treatment in Spanish is available in the companion tutorial "Las matemáticas detrás de TMLE".

6.4.26.1 The Statistical Model and Target Parameter

Let the observed data be $O = (W, A, Y) \sim P_0$, where W is a vector of confounders, $A \in \{0, 1\}$ is a binary treatment, and Y is the outcome. The statistical model $\mathcal{M}$ is nonparametric (i.e.,

no restrictions beyond positivity). The target parameter is the average treatment effect:

$$\Psi(P_0) = \mathbb{E}_0[\mathbb{E}_0[Y \mid A = 1, W] - \mathbb{E}_0[Y \mid A = 0, W]] = \mathbb{E}_0[\bar{Q}_0(1, W) - \bar{Q}_0(0, W)]$$

where $\bar{Q}_0(A, W) = \mathbb{E}_0[Y \mid A, W]$ is the true outcome regression.

6.4.26.2 The Efficient Influence Curve

A central object in semiparametric theory is the **efficient influence curve** (EIC), also called the **efficient influence function** (EIF) or the **canonical gradient**. Before presenting the formula for the ATE, it is helpful to build intuition for what an influence function is and how it is derived.

Intuition: What Is an Influence Function?

An influence function $\phi(Z)$ quantifies **how much a single observation Z influences the estimate of a parameter** ψ. More formally, $\phi(z)$ measures how the parameter changes when we add an infinitesimal amount of probability mass at the point z:

$$\phi(z) = \lim_{\epsilon \to 0} \frac{\Psi(P_\epsilon) - \Psi(P)}{\epsilon}$$

where $P_\epsilon = (1 - \epsilon)P + \epsilon \cdot \mathbf{1}_z$ is a "contaminated" distribution that places a tiny extra mass at z. This is known as the **point mass contamination** or **Gâteaux derivative** approach to deriving influence functions (Kennedy, 2022; Tsiatis, 2006).

The influence function is always mean-zero ($\mathbb{E}_P[\phi(Z)] = 0$) under the true distribution. Its variance provides the asymptotic variance of the corresponding estimator: for an asymptotically linear estimator $\hat{\Psi}$ with influence function ϕ, we have $\sqrt{n}(\hat{\Psi} - \Psi) \xrightarrow{\mathcal{D}} N(0, \mathrm{Var}(\phi))$.

Building Up: From Simple Mean to ATE

The EIF is best understood by building up from simpler parameters:

Example 1: The population mean. For $\Psi(P) = \mathbb{E}[Z]$, the influence function is simply:

$$\phi(Z) = Z - \mathbb{E}[Z]$$

This is exactly the influence function of the sample mean — confirming that $\bar{Z}_n$ is a nonparametrically efficient estimator of $\mathbb{E}[Z]$. The intuition is clear: observations above the mean pull the estimate up ($\phi > 0$), observations below pull it down ($\phi < 0$).

Example 2: The conditional mean. For $\Psi(P) = \mathbb{E}[Y \mid X = x]$, the influence function is:

$$\phi(Y, X) = \frac{\mathbf{1}_x(X)}{P(X = x)} \cdot (Y - \mathbb{E}[Y \mid X = x])$$

This illustrates a recurring pattern in influence functions: an **IPW-style term** ($\mathbf{1}_x(X)/P(X = x)$) multiplied by a **residual** ($Y - \mathbb{E}[Y|X = x]$). Only observations with $X = x$ contribute, and they are weighted inversely to their probability of having $X = x$.

Example 3: The ATE. Using **gradient algebra** — the linearity, product, and chain rules for influence functions — the EIF for the ATE can be built from the components above. The canonical gradient rules (see Kennedy (2022)) allow us to combine influence functions for $\mathbb{E}[Y \mid A = 1, W]$ and $\mathbb{E}[Y \mid A = 0, W]$ to obtain, for treatment level $a \in \{0, 1\}$:

$$\phi_a(O) = \frac{\mathbf{1}_a(A)}{\mathbb{P}(A = a \mid W)}(Y - \mathbb{E}[Y \mid A = a, W]) + \mathbb{E}[Y \mid A = a, W] - \Psi_a$$

The overall EIF for the ATE $\Psi = \Psi_1 - \Psi_0$ is then $\phi_1 - \phi_0$, giving the familiar expression below.

The EIF for the ATE

For the ATE parameter under the nonparametric model, the EIC is:

$$D^*(P_0)(O) = \left(\frac{A}{g_0(1 \mid W)} - \frac{1 - A}{g_0(0 \mid W)}\right)(Y - \bar{Q}_0(A, W)) + \bar{Q}_0(1, W) - \bar{Q}_0(0, W) - \Psi(P_0)$$

where $g_0(a \mid W) = \mathbb{P}_0(A = a \mid W)$ is the true propensity score.

i Anatomy of the ATE Influence Function

The EIF for the ATE has a natural decomposition into three orthogonal components:

1. **The IPW residual term:** $\left(\frac{A}{g(1|W)} - \frac{1-A}{g(0|W)}\right)(Y - \bar{Q}(A, W))$ — this term lives in the tangent space $T_{Y|A,W}$ (functions of Y given A, W with conditional mean zero). It captures the information about the outcome that is not explained by the outcome regression.
2. **The outcome regression term:** $\bar{Q}(1, W) - \bar{Q}(0, W)$ — this term lives in the tangent space T_W (functions of W only). It captures the covariate-specific treatment effect.
3. **The centering term:** $-\Psi(P_0)$ — ensures the EIF has mean zero.

This decomposition is not arbitrary; it follows from the **orthogonal factorization of the tangent space** in the nonparametric model: $L_2^0(P) = T_{Y|A,W} \oplus T_{A|W} \oplus T_W$, where

each subspace contains scores (directions) that perturb a different component of the likelihood — the outcome conditional density, the treatment mechanism, and the covariate marginal distribution, respectively (Bickel et al., 1993; Kennedy, 2022; Tsiatis, 2006). In the ATE parameter, the component in $T_{A|W}$ is zero because perturbing the treatment mechanism alone does not change the ATE (it only changes *how many* people get treated, not *what happens* when they do). This is why the treatment mechanism appears only as a weight in the EIF, not as a separate additive term.

The EIC has three fundamental properties: 1. It is a **gradient** for the parameter Ψ: it satisfies the central identity $\nabla_h \Psi = \mathbb{E}_P[D^*(P) \cdot h]$ for any score h, characterizing how $\Psi(P)$ changes under perturbations of P. 2. Its variance provides the **semiparametric efficiency bound**: $\mathrm{Var}_{P_0}(D^*(P_0)(O))$ is the smallest possible asymptotic variance among regular asymptotically linear (RAL) estimators. 3. Any estimator $\hat{\Psi}$ that solves the EIC equation (i.e., $\frac{1}{n}\sum_i D^*(\hat{P})(O_i) = 0$) is **asymptotically linear and efficient**.

6.4.26.3 von Mises Expansion and the Bias-Variance Decomposition

For any estimator $\hat{P}$ of P_0, the error in the plug-in estimate can be decomposed via the **von Mises expansion** (functional Delta method):

$$\Psi(\hat{P}) - \Psi(P_0) = -\mathbb{E}_{P_0}[D^*(\hat{P})(O)] + R(\hat{P}, P_0)$$

where $R(\hat{P}, P_0)$ is a second-order remainder term. For the ATE, this remainder takes the form:

$$R(\hat{P}, P_0) = \mathbb{E}_0\left[\left(\frac{g_0(A \mid W) - \hat{g}(A \mid W)}{\hat{g}(A \mid W)}\right)\left(\bar{Q}_0(A, W) - \hat{\bar{Q}}(A, W)\right)\right]$$

The key insight is that R is a **product of differences** between the estimated and true nuisance functions. This means the bias vanishes if *either* $\hat{g} \approx g_0$ *or* $\hat{\bar{Q}} \approx \bar{Q}_0$ — this is the mathematical basis for **double robustness**. Moreover, if both nuisance functions are estimated at rate $n^{-1/4}$, the remainder is $o_P(n^{-1/2})$, meaning the estimator is efficient.

6.4.26.4 The Targeting Step

The initial estimator $\hat{\bar{Q}}^0$ (from Step 1) may not solve the EIC equation. The targeting step updates $\hat{\bar{Q}}^0$ to $\hat{\bar{Q}}^1$ using a **fluctuation model**:

$$\mathrm{logit}(\hat{\bar{Q}}^1(A, W)) = \mathrm{logit}(\hat{\bar{Q}}^0(A, W)) + \epsilon \cdot H(A, W)$$

where $H(A,W) = \frac{A}{\hat{g}(1|W)} - \frac{1-A}{\hat{g}(0|W)}$ is the **clever covariate**. The parameter ϵ is estimated by maximum likelihood (logistic regression with Y as outcome, $\text{logit}(\hat{\bar{Q}}^0)$ as offset, and H as the sole covariate).

This update is designed to solve the **score equation** derived from the EIC:

$$\sum_{i=1}^{n} H(A_i, W_i) \left(Y_i - \text{expit} \left(\text{logit}(\hat{\bar{Q}}^0(A_i, W_i)) + \epsilon \cdot H(A_i, W_i) \right) \right) = 0$$

After updating, the plug-in estimator $\hat{\Psi}_{\text{TMLE}} = \frac{1}{n}\sum_{i=1}^{n}(\hat{\bar{Q}}^1(1, W_i) - \hat{\bar{Q}}^1(0, W_i))$ satisfies $P_n D^*(\hat{P}^1) = 0$, which is the requirement for asymptotic efficiency.

6.4.26.5 Asymptotic Properties

Under regularity conditions, the TMLE estimator satisfies:

$$\sqrt{n}\left(\hat{\Psi}_{\text{TMLE}} - \Psi(P_0)\right) \xrightarrow{\mathcal{D}} N(0, \sigma^2_{\text{eff}})$$

where $\sigma^2_{\text{eff}} = \text{Var}_{P_0}(D^*(P_0)(O))$ is the semiparametric efficiency bound. The variance can be consistently estimated by the sample variance of the EIC:

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}\left(\widehat{EIC}_i\right)^2, \quad \text{where} \quad \widehat{EIC}_i = D^*(\hat{P}^1)(O_i)$$

Wald-type 95% confidence intervals are constructed as $\hat{\Psi}_{\text{TMLE}} \pm 1.96 \cdot \hat{\sigma}/\sqrt{n}$.

> **Key Insight**
>
> The EIC is both a **blueprint for estimation** (it tells us how to construct the clever covariate and targeting step) and a **tool for inference** (its variance gives the standard error). This dual role makes it the central object in semiparametric efficient estimation.

> **How to Derive an Efficient Influence Function: Three Strategies**
>
> When the parameter of interest is not the ATE but something else (e.g., a conditional odds ratio, a quantile treatment effect, or a longitudinal parameter), one needs to derive the EIF from first principles. The following three strategies, summarized from Kennedy (2022) and the foundational texts Tsiatis (2006) and Bickel et al. (1993), cover most practical situations.

6.4.27 Strategy 1: Point Mass Contamination (Gâteaux Derivative)

This is the most direct approach and works well in **nonparametric (saturated) models**, where there are no restrictions on the tangent space. The idea is to:

1. Consider a path $P_\epsilon = (1-\epsilon)P + \epsilon \cdot \mathbf{1}_{\tilde{z}}$ that contaminates P by placing extra mass at a point $\tilde{z}$.
2. Compute the derivative $\frac{d}{d\epsilon}\Psi(P_\epsilon)\big|_{\epsilon=0}$.
3. Evaluate the resulting expression at $\tilde{z} = Z$ to obtain $\phi(Z)$.

For a saturated model, there is **only one influence function, and it is automatically the efficient one**. This is why the ATE's EIF was derived above under the nonparametric model — in that setting, every valid RAL estimator shares the same influence function (up to asymptotic equivalence).
For **semiparametric (non-saturated) models**, point mass contamination requires care: the contaminated path may immediately leave the model space. In such cases, one of the next two strategies is preferred.

6.4.28 Strategy 2: Gradient Algebra (Building from Simpler Pieces)

Once influence functions for basic building blocks are known (e.g., the mean, conditional mean, and density ratio), more complex EIFs can be constructed using algebraic rules. The **EIF operator** Φ satisfies:

- **Linearity:** $\Phi(a_1\psi_1 + a_2\psi_2) = a_1\Phi(\psi_1) + a_2\Phi(\psi_2)$
- **Product rule:** $\Phi(\psi_1\psi_2) = \Phi(\psi_1)\psi_2 + \psi_1\Phi(\psi_2)$
- **Chain rule:** $\Phi(g(\psi)) = g'(\psi)\,\Phi(\psi)$

This is how the ATE's EIF was built up in the examples above: starting from the influence function of the mean, applying the rules to obtain the conditional mean, and then combining terms for $A = 1$ and $A = 0$ via linearity. The gradient algebra approach is particularly useful when the target parameter is a composite of simpler estimands.

6.4.29 Strategy 3: Projection Approach (for Restricted Models)

When the model is **not saturated** — for example, in a randomized trial where the propensity score is known by design — there may be a whole space of valid influence functions, only one of which is efficient. The projection approach finds it:

1. Start with the influence function ϕ of **any** valid RAL estimator for Ψ (e.g., the IPW estimator).
2. Project ϕ onto the tangent space $\mathcal{T}$ of the model: $\phi^\dagger = \Pi(\phi \mid \mathcal{T})$.
3. The projection is the EIF.

For a randomized trial, the tangent space excludes scores that perturb the treatment mechanism (since $P(A|W)$ is known), so the projection step removes the component of the IPW influence function that lies in $T_{A|W}$ and adds back the efficient augmentation term $\bar{Q}(A, W)$. The result is identical to the observational-study EIF — an important fact: **the canonical gradient for the ATE is the same in observational studies and randomized trials under nonparametric assumptions about the outcome and covariate distributions**.

The Central Identity

All three strategies rest on the same fundamental relationship, known as the **central identity**:

$$\nabla_h \Psi = \mathbb{E}_P[\phi \cdot h]$$

where $\nabla_h \Psi$ is the pathwise derivative of Ψ in direction h (a score), and ϕ is the influence function. This identity states that the influence function is the **Riesz representer** of the derivative functional in the Hilbert space $L_2^0(P)$ of mean-zero, finite-variance functions. Once this identity is verified — by computing $\nabla_h \Psi$ from the left and checking it equals $\mathbb{E}[\phi \cdot h]$ for arbitrary h — the candidate ϕ is confirmed as a valid influence function.

For the nonparametric model, the candidate is automatically the efficient one. For restricted models, one must additionally verify that ϕ lies in the (closed) tangent space.

Further Reading on TMLE and Semiparametric Theory

TMLE sits at the intersection of causal inference, machine learning, and semiparametric efficiency theory. Readers who wish to deepen their understanding beyond this chapter may find the following resources helpful, ordered from most applied to most theoretical:

1. **Schuler & Rose (2016)** — "Targeted maximum likelihood estimation for causal inference in observational studies" (*American Journal of Epidemiology*). A step-by-step written explanation aimed at applied epidemiologists. An excellent starting point after this chapter.

2. **Miguel Angel Luque-Fernandez's bookdown tutorial** — "Targeted Learning in R" — a hands-on, code-forward tutorial with reproducible examples in R, available at https://github.com/migariane/TutorialComputationalCausalInferenceEstimators.

3. **The `tlverse` Handbook** — The targeted learning ecosystem in R (`tmle3`, `sl3`,

`tlverse`), available at https://tlverse.org/. Covers modern implementations of TMLE, CV-TMLE, and Super Learner with worked examples.

4. **Kennedy (2023)** — "Semiparametric Theory" — a concise, accessible introduction to influence functions, efficiency bounds, and the theory behind TMLE (available at arxiv.org/abs/1709.06418). Kennedy's slideshow tutorial is a particularly gentle on-ramp to the semiparametric theory concepts.

5. **Fisher & Kennedy (2021)** — "Visually Communicating Influence Functions" — offers an intuitive, graphical understanding of influence functions. Best approached after working through Herb Susmann's interactive Observable tutorial on one-step estimators and pathwise derivatives.

6. **van der Laan & Rose (2011)** — *Targeted Learning: Causal Inference for Observational and Experimental Data* (**vanderLaan2011?**) — the foundational text. The prose is accessible, though the mathematics assumes graduate-level statistics.

7. **Schuler (2024)** — "Modern Causal Inference" — a comprehensive, freely available course covering the derivation of efficient influence functions for a range of causal parameters. The EIF derivation chapter is particularly relevant: it walks through point mass contamination, gradient algebra, and projection approaches with worked examples from the population mean up to the ATE. The full course is available at https://alejandroschuler.github.io/mci/.

8. **Hines, Diaz-Ordaz, Vansteelandt, & Jamthikar (2021–2022)** — "Demystifying Statistical Learning Based on Efficient Influence Functions" — a two-part tutorial that provides a clear, step-by-step exposition of influence function theory and its role in constructing efficient estimators (available at arxiv.org/abs/2107.00681).

9. **Kennedy (2022)** — "Semiparametric doubly robust targeted double machine learning: a review" (Kennedy, 2022) — a review paper that covers the theoretical foundations behind the EIF derivation strategies discussed above, with a focus on the interplay between semiparametric theory and machine learning.

If you are new to semiparametric theory, expect to revisit these materials several times. The concepts build on each other, and it is normal for the targeting step and influence functions to feel opaque on first encounter. The important practical takeaway is that TMLE provides valid inference while allowing flexible machine learning — a combination that parametric alternatives do not offer.

6.4.30 Super Learner

In Section 6.4.2, parametric logistic regression models were used to define the outcome and exposure models. However, parametric models are vulnerable to misspecification, which can introduce bias into causal estimates. To address this, machine learning algorithms can be used to flexibly model complex relationships in the data and improve the accuracy of nuisance parameter estimation.

Why is machine learning useful in TMLE? Accurate estimation of nuisance parameters, such as the treatment mechanism and the outcome regression, is a critical component of TMLE. Misspecification of either model can undermine the robustness of the estimator. Machine learning algorithms mitigate this risk by providing flexible, data-adaptive models that do not rely on strict parametric assumptions. These methods can capture nonlinearities and interactions that are often missed by simpler models, thus reducing bias and improving efficiency.

However, the role of machine learning in TMLE goes deeper than flexibility alone. TMLE's double-robustness property means that the estimator is consistent if *either* the outcome model or the propensity score model is correctly specified. Efficiency — achieving the smallest possible variance at $\sqrt{n}$ rate — requires *both* models to be consistent. The reason SuperLearner is used for estimating the outcome and treatment regressions is to give the best possible chance of having both models correctly specified and thereby obtaining an efficient estimate (**vanderLaan2011?**). In other words, SuperLearner helps TMLE achieve not just consistency but full semiparametric efficiency.

Because selecting the best-performing algorithm in advance is difficult, TMLE incorporates ensemble methods that leverage the strengths of multiple learners. Among these, SuperLearner is a theoretically grounded approach that optimally combines multiple candidate algorithms to improve predictive performance.

What is SuperLearner? SuperLearner is an ensemble machine learning method used in TMLE to combine predictions from multiple candidate models into a single, optimised estimator. This method is based on the principle that no single algorithm performs best across all data structures—a concept known as the "no free lunch" theorem in machine learning. Instead of relying on a single model, SuperLearner uses cross-validation to assess the performance of each algorithm and assigns weights to construct the best possible convex combination of models.

This approach is supported by the Oracle Inequality, which guarantees that, asymptotically, SuperLearner performs at least as well as the best convex combination of models in the library. This makes it a robust and reliable choice for estimating nuisance parameters in high-dimensional or complex datasets, particularly in applications like biostatistics and epidemiology.

How does SuperLearner work? The SuperLearner algorithm consists of several steps:

1. A library of candidate algorithms is specified in advance. This library may include both parametric models (e.g., logistic regression) and machine learning methods (e.g., random forests, gradient boosting).
2. The dataset is split into training and validation folds using cross-validation.
3. Each algorithm is trained on the training folds and its predictive performance is evaluated on the validation folds using a suitable loss function (e.g., mean squared error).
4. A meta-learner determines the optimal set of weights for combining the candidate models based on their cross-validated performance.
5. The final prediction is produced as a weighted average of the predictions from the candidate models.

This process ensures that the ensemble prediction is tailored to the data and performs at least as well as the best-performing model in the library.

Benefits of SuperLearner SuperLearner offers a number of advantages over traditional parametric models when used in TMLE. It is highly flexible, adapting to the structure of the data without assuming a fixed functional form. This allows for more effective modeling of complex, nonlinear relationships. Model robustness is enhanced by combining multiple algorithms, reducing the risk that poor performance from any single model will degrade the overall estimator.

By improving the accuracy of nuisance parameter estimation, SuperLearner contributes to more efficient TMLE estimates, yielding smaller standard errors and tighter confidence intervals. The method also offers theoretical guarantees: under regularity conditions, SuperLearner is asymptotically optimal, performing as well as or better than any individual algorithm or convex combination in the library. SuperLearner is also highly customisable. A diverse library of algorithms, combining both machine learning and parametric models, can be defined to reflect the specific needs and characteristics of the data.

Overall, the integration of SuperLearner within TMLE ensures that the procedure remains robust to model misspecification, data-adaptive, and efficient in finite samples.

6.4.31 Comparison with other estimators

To see the benefit of using TMLE over other methods, we can run a simulation and compare the results from each of the methods against a known true value for the ATE. We simulated data on 1000 observations, estimated the ATE and standard error, then repeated the study 1000 times. Figure 6.1 shows the results of the simple simulation study.

You can see that the bias is smallest for TMLE and largest for the naive regression adjustment (RA) approach. TMLE benefits from not only using SuperLearner to define the exposure and outcome models but also from the targeting step. We could have used machine learning algorithms to define the exposure or outcome models for the other methods, such as AIPTW.

The lower half of the graph shows the coverage rate. This is proportion of confidence intervals from each method that contain the true value of the ATE. If we are trying to estimate 95% confidence intervals, then we should expect that the method contains the true value of the ATE 95% of the time. TMLE gives a coverage of 95.1%, which is almost perfect. Other methods have a much lower coverage rate, which also shows the beneficial properties of TMLE.

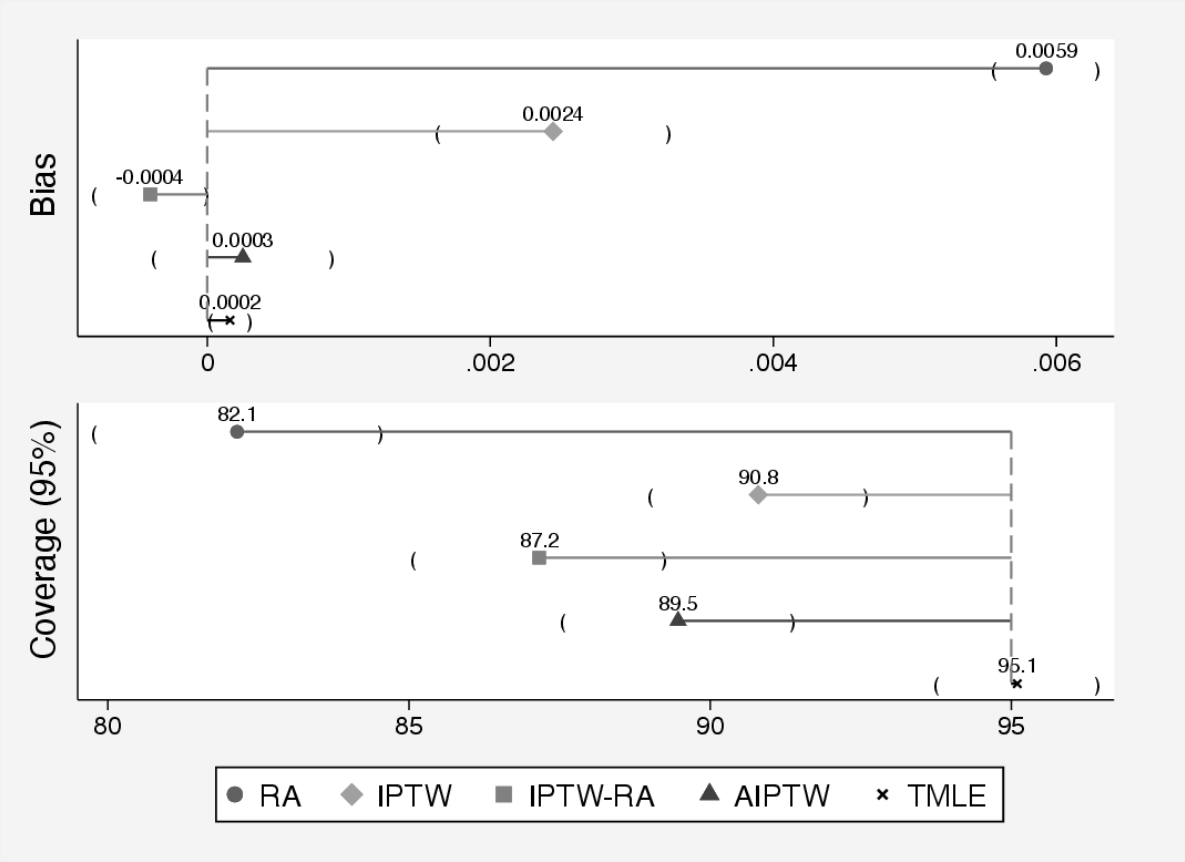

Figure 6.1: Bias and coverage rate of various causal estimators in a simple simulation study.

6.5 Cross-Validated Targeted Maximum Likelihood Estimation

Cross-Validated Targeted Maximum Likelihood Estimation (CV-TMLE) is an extension of TMLE designed to improve performance in finite samples, particularly when using flexible, data-adaptive methods such as machine learning for estimation of nuisance parameters. CV-TMLE incorporates cross-validation directly into the targeting step of the TMLE procedure, rather than using cross-validation solely for model selection prior to estimation. This results in more stable estimates and better control of overfitting, especially when sample sizes are modest or when the estimation of the outcome regression or propensity score is highly variable.

6.5.1 Motivation and Advantages over TMLE

While standard TMLE already enjoys double robustness and asymptotic efficiency when both models are correctly specified, it may perform poorly in small samples if the machine learning estimators overfit the training data. Standard TMLE typically involves estimating the initial outcome regression and propensity score on the full dataset, which can lead to optimistic predictions and targeted updates that fail to generalize. CV-TMLE addresses this by ensuring that the data used to perform the targeting step is distinct from the data used to estimate the initial regressions.

Beyond overfitting, standard TMLE has a more fundamental theoretical requirement: valid statistical inference depends on the nuisance function estimators belonging to a **Donsker class** — a class of functions whose empirical processes converge to a Gaussian limit at rate $n^{-1/2}$. When highly adaptive machine learning algorithms (e.g., random forests, gradient boosting, or neural networks) are used within the Super Learner library, the resulting estimators may not satisfy the Donsker class condition. This can inflate the bias of the targeted estimand and produce anti-conservative variance estimates, leading to poor confidence interval coverage (Smith et al., 2025).

The Donsker class condition is particularly likely to be violated under: - **Near-positivity violations**, where estimated propensity scores are close to 0 or 1, causing the clever covariate $H(A, W)$ to take extreme values. - **Small sample sizes**, where the complexity of the machine learning algorithms exceeds what the data can support. - **Flexible Super Learner libraries**, where highly adaptive learners (e.g., regression trees, random forests) can overfit, increasing bias and reducing variance in ways that invalidate standard errors.

CV-TMLE provides a straightforward remedy: by separating the estimation of nuisance functions (on training folds) from the targeting step (on validation folds), it ensures that the targeted update and inference are based on **out-of-sample predictions**. This breaks the dependence that causes the Donsker class violation, restoring valid inference even when highly flexible learners are used. As shown by Smith et al. (2025), CV-TMLE vastly improves confidence interval coverage without adversely affecting bias, especially in settings with near-positivity violations or small samples. Importantly, CV-TMLE is also **much less sensitive to the choice of the Super Learner library** than standard TMLE — making it a safer default when the optimal set of learners is unknown in advance.

Key advantages of CV-TMLE include:

- Improved finite-sample performance by reducing overfitting bias.
- Better alignment between cross-validation and influence function-based inference.
- Increased stability of the targeted update, especially when using machine learning.

6.5.2 Cross-Validation for Model Selection and Overfitting Prevention

Cross-validation is a common strategy for selecting among multiple candidate learners (e.g., via Super Learner), but in CV-TMLE, it plays an additional role. Instead of using cross-validation only to choose the best prediction algorithm, CV-TMLE partitions the data into V folds and performs the entire TMLE procedure separately in each validation fold. The influence function contributions from each fold are then aggregated to produce the final estimate and standard error.

This prevents the targeted update from leveraging overfit predictions and leads to a more valid approximation of the efficient influence function, especially when the sample size is small or when the nuisance functions are highly adaptive.

6.5.3 Steps in CV-TMLE Estimation

The CV-TMLE algorithm proceeds as follows:

1. **Split the data** into V folds (commonly $V = 10$).
2. For each fold:
    1. Estimate the initial outcome regression $Q(A, X)$ and propensity score $g(A \mid X)$ on the training set.
    2. Predict Q and g for the validation set.
    3. Perform the TMLE fluctuation (targeting step) only on the validation set using the predicted values.
    4. Compute the influence function contribution for each observation in the validation set.
3. Aggregate influence function values across all folds to compute the final estimate.
4. Estimate the variance using the empirical variance of the influence function.

6.5.4 Software Implementation

Box 5.16: Cross-validated TMLE with eltmle (cveltmle)

6.5.5 Stata

```
* Cross-validated TMLE for improved small-sample performance
* cveltmle is used without the standard tmle option
eltmle Y A W, cveltmle
```

6.5.6 R

```
library(tmle3)
library(sl3)

# Define the nodes
tmle_spec <- tmle_ATE(
  treatment_level = 1,
  control_level = 0
)

# Define learner library (e.g., Super Learner)
learner <- Lrnr_sl$new(learners = c("Lrnr_mean", "Lrnr_glm"))

# Define likelihood and task
tmle_task <- tmle_spec$make_tmle_task(data, node_list)
likelihood <- tmle_spec$make_initial_likelihood(tmle_task, learner)

# Run CV-TMLE
tmle_fit <- tmle_spec$tmle_update(tmle_task, likelihood)
summary(tmle_fit)
```

Alternatively, the `ltmle` package can also be used to perform CV-TMLE when working with longitudinal or time-to-event data, although the interface is more prescriptive.

6.5.7 Discussion of Small-Sample Behavior

In small to moderate samples, the benefits of CV-TMLE become especially clear. The standard TMLE may exhibit instability due to overfitting of the initial estimators, particularly when using highly adaptive algorithms such as random forests or neural networks. By ensuring that targeting is performed on out-of-sample predictions, CV-TMLE avoids this problem and more closely approximates the behavior of the estimator in large samples.

Empirical studies have shown that CV-TMLE achieves better coverage of confidence intervals and lower mean squared error in small samples, particularly when the complexity of the outcome or treatment models is high (Smith et al., 2025). The Donsker class condition discussed above provides the theoretical explanation: cross-validating the targeting step ensures the empirical process of the nuisance estimators remains well-behaved, even when the individual learners themselves fall outside the Donsker class. As such, CV-TMLE is now regarded as a default strategy in many machine-learning-integrated causal inference pipelines, especially when flexible Super Learner libraries or near-positivity problems are anticipated.

Two Layers of Cross-Validation in CV-TMLE

1. Cross-validation in the Super Learner: Used to optimize *prediction* of nuisance parameters: - Outcome regression: $Q(A, X) = \mathbb{E}[Y \mid A, X]$ - Propensity score: $g(A \mid X) = \mathbb{P}(A = 1 \mid X)$
Super Learner performs internal V-fold cross-validation to evaluate and combine learners based on predictive performance. This layer is solely focused on choosing the best-fitting models for the nuisance functions.
2. Cross-validation in CV-TMLE: Used to improve *estimation* of the target parameter (e.g., ATE) and valid inference: - The dataset is split into V folds. - Nuisance functions are trained on $V - 1$ folds and predictions are made on the held-out fold. - The TMLE targeting step and influence curve calculations are performed on the held-out fold.
This ensures that the targeted update and standard error are based on out-of-sample predictions, reducing overfitting and improving the validity of inference.
In summary: Cross-validation in the Super Learner is for *better prediction*; cross-validation of TMLE is for *valid estimation and inference*.

6.5.8 Summary

Cross-validated TMLE extends the benefits of TMLE to smaller samples and high-dimensional settings by embedding cross-validation directly into the targeting procedure. It mitigates overfitting, improves stability, and provides more accurate inference, particularly when leveraging flexible machine learning methods to estimate nuisance parameters. The procedure is supported in `tmle3`, `ltmle`, and other implementations within the targeted learning ecosystem.

6.6 Conclusion

Double-robust estimators represent a major advance in causal inference methodology, offering protection against model misspecification that singly robust methods cannot provide. This chapter has traced the evolution from the simple combination of IPW with outcome regression, through the augmented IPW (AIPW) estimator, to the fully developed Targeted Maximum Likelihood Estimation (TMLE) framework.

AIPW achieves double robustness by combining a weighting term with an augmentation term derived from the outcome model. When either the propensity score model or the outcome model is correctly specified, AIPW is consistent. However, AIPW can be unstable with extreme weights and does not incorporate a targeting step that directly optimizes the estimator for the causal parameter of interest.

TMLE addresses these limitations through its defining innovation: the targeting step. By updating the initial outcome predictions using a fluctuation model parameterized by a clever covariate — constructed from the inverse probability weights — TMLE ensures that the final estimator solves the efficient influence curve equation. This grants TMLE three essential properties: double robustness, semiparametric efficiency when both models are correct, and improved finite-sample performance compared to AIPW.

The integration of Super Learner — a cross-validated ensemble of machine learning algorithms — allows TMLE to estimate nuisance parameters flexibly and adaptively. The Oracle inequality guarantees that Super Learner performs asymptotically at least as well as the best algorithm in the library, making TMLE with Super Learner the current gold standard for robust, data-adaptive causal effect estimation.

CV-TMLE further improves upon standard TMLE by embedding cross-validation into the targeting step itself, mitigating overfitting and improving inference in small samples. The mathematical foundations — the efficient influence curve, von Mises expansion, and second-order remainder — provide the theoretical guarantees that underpin these methods.

Key takeaways from this chapter: - Doubly-robust estimators are consistent if *either* the outcome model *or* the treatment model is correctly specified. - AIPW combines inverse probability weighting with outcome regression augmentation but does not include a targeting step. - TMLE adds a targeting step via a fluctuation model and clever covariate, solving the EIC equation for the target parameter. - Super Learner provides data-adaptive estimation of nuisance functions with optimality guarantees. - CV-TMLE extends TMLE with cross-validation for improved small-sample performance and valid inference. - The EIC is the central mathematical object: it provides both the blueprint for estimation and the basis for inference.

When both nuisance models can be estimated well — and particularly when using Super Learner — TMLE provides the most robust, efficient, and theoretically grounded framework currently available for estimating causal effects from observational data.

6.7 Glossary

AIPW Augmented Inverse Probability of Treatment Weighting — a doubly-robust estimator combining IPW with outcome regression augmentation.

Clever covariate A function of the propensity score, $H(A, W) = \frac{A}{g(1|W)} - \frac{1-A}{g(0|W)}$, used in the TMLE targeting step to update the initial outcome predictions.

Double robustness A property of an estimator that remains consistent if either the outcome model or the treatment model is correctly specified.

Efficient influence curve (EIC) The canonical gradient of the target parameter; characterizes the semiparametric efficiency bound and provides the basis for the TMLE targeting step and inference.

Fluctuation parameter The coefficient ϵ in the TMLE targeting model, estimated by solving the score equation derived from the EIC.

IPTW Inverse Probability of Treatment Weighting — reweights observations by the inverse of the propensity score to estimate marginal causal effects.

Super Learner A cross-validated ensemble method that optimally combines multiple candidate algorithms to estimate nuisance functions in TMLE.

Targeting step The procedure in TMLE that updates initial outcome predictions using a fluctuation model to ensure the estimator solves the EIC equation.

TMLE Targeted Maximum Likelihood Estimation — a doubly-robust, semiparametric efficient plug-in estimator that uses a targeting step to align estimation with the causal parameter of interest.

von Mises expansion A functional Taylor expansion used to decompose the estimation error into a first-order (influence function) term and a second-order remainder, providing the theoretical basis for double robustness.

7 Causal Inference for Longitudinal Data

The previous chapters have described methods that are used when the intervention occurs at only one time point. Often, there are settings where the intervention occurs at multiple time points. Such settings could be a chemotherapy given to patients with cancer at regular intervals, a series of public health policies designed to reduce environmental impact on the population's health over time, or how health outcomes change over time within certain geographical clusters.

This chapter is split into two parts. Part I explores methods for multiple time-point interventions, including inverse probability weighting for longitudinal treatments (IPTW), marginal structural models (MSMs), double robust methods (AIPTW and LTMLE), and longitudinal TMLE (Lendle et al., 2017). Part II explores methods for time-to-event outcomes, including the parametric g-formula for survival and TMLE for survival data.

7.1 Part I: Multiple Time-Point Interventions

7.1.1 Time-Varying Confounding and Its Challenges

One of the most significant challenges in causal inference for longitudinal data is the presence of time-varying confounding. In a longitudinal setting, confounders are measured repeatedly over time and may both influence and be influenced by prior treatment. This feedback loop creates a fundamental difficulty for standard statistical methods.

To understand the problem, consider covariates L_t that affect treatment decisions A_t at each time t, but are themselves affected by earlier treatment A_{t-1}. In this case, L_t serves as both a confounder (affecting treatment and outcome) and a mediator (affected by prior treatment). Adjusting for L_t using standard regression methods may induce bias by conditioning on a variable on the causal pathway or introducing collider bias.

This situation violates one of the key assumptions required for unbiased estimation in standard regression: that adjustment covariates are not affected by the exposure. In longitudinal settings, naive adjustment for time-varying confounders using outcome regression can therefore produce biased effect estimates.

For example, in the context of HIV treatment, CD4 count is a time-varying confounder: it predicts future treatment decisions (whether to initiate or modify ART) and also predicts the

outcome (e.g., mortality). But CD4 count is also affected by prior ART exposure. If we adjust for CD4 count in a regression model, we may block part of the effect of ART or introduce bias through conditioning on a collider.

This motivates the use of methods specifically designed to handle time-varying confounding, such as: - **Inverse Probability of Treatment Weighting (IPTW):** Reweights individuals to create a pseudo-population in which treatment is independent of time-varying confounders. - **G-computation:** Uses the g-formula to model the joint distribution of covariates and outcomes under a specific treatment regime. - **Doubly Robust Estimators:** Combine outcome modeling and treatment modeling to protect against misspecification of either.

These methods rely on sequential models that reflect the time-varying structure of the data and appropriately account for dynamic confounding.

7.1.2 Motivating Examples from Longitudinal Studies

Longitudinal data arise frequently in medicine, public health, economics, and the social sciences, where individuals are followed over time and data are collected at multiple time points. These repeated measurements offer rich opportunities to understand how treatments, exposures, or policies influence outcomes over time. However, the time-varying nature of treatments and confounders introduces methodological challenges for causal inference.

A classic motivating example comes from the HIV treatment literature. Consider a study evaluating the effect of initiating antiretroviral therapy (ART) on survival among HIV-positive individuals. Treatment initiation may depend on evolving clinical indicators such as CD4 cell count or viral load. These indicators also influence prognosis and are themselves affected by earlier treatment decisions. Thus, CD4 count acts as a time-varying confounder that is affected by prior treatment. Standard regression methods that adjust for CD4 count at each time point may introduce bias by blocking part of the treatment effect or conditioning on a collider.

Another example is the management of blood pressure over time. Patients with hypertension may begin or adjust medications based on current blood pressure readings, which in turn are influenced by prior treatment. Smoking cessation interventions, weight loss programs, and mental health treatments are further domains where longitudinal data play a central role.

These settings share common features: time-varying exposures, time-varying confounders, and outcomes measured after repeated decisions. In such contexts, specialized methods are required to estimate causal effects, including marginal structural models, the g-formula, and doubly robust estimators.

7.1.3 Notation for Longitudinal Data

To formalize the discussion of causal inference with longitudinal data, we introduce notation to represent the sequence of treatments, covariates, and outcomes over time.

Let $t = 0, 1, \ldots, T$ denote discrete time points. For an individual i, define: - A_t: treatment or exposure at time t - L_t: time-varying covariates measured at time t - $\bar{A}_t = (A_0, A_1, \ldots, A_t)$: treatment history up to and including time t - $\bar{L}_t = (L_0, L_1, \ldots, L_t)$: covariate history up to and including time t - Y: outcome of interest, measured at time $T+1$ or at the end of follow-up - C_t: indicator of censoring at time t, where $C_t = 1$ if censored at time t

We use capital letters to denote random variables and lowercase letters for their realizations. For example, a_t is a specific value of A_t. The notation $Y^{\bar{a}}$ refers to the potential outcome that would be observed under the treatment regime $\bar{a} = (a_0, a_1, \ldots, a_T)$.

In addition, we define the data structure for each individual as:

$$O = (L_0, A_0, L_1, A_1, \ldots, L_T, A_T, Y).$$

This longitudinal data structure allows for dynamic treatment strategies that may assign treatment at time t based on past covariate and treatment history. The goal of causal inference in this setting is to estimate the average causal effect of a treatment regime $\bar{a}$ on the outcome Y, denoted:

$$E[Y^{\bar{a}}].$$

This framework forms the foundation for the causal models and estimators introduced in subsequent sections.

Box 6.1: Simulating longitudinal data with time-varying treatment and confounders

7.1.4 R

```r
n <- 1000
set.seed(123)

L0 <- rnorm(n)
A0 <- rbinom(n, 1, plogis(0.5 * L0))
L1 <- 0.5 * L0 + 0.8 * A0 + rnorm(n)
A1 <- rbinom(n, 1, plogis(0.5 * L1))
Y  <- 0.7 * L1 + 1.2 * A1 + rnorm(n)

data <- data.frame(L0, A0, L1, A1, Y)
head(data)
```

7.1.5 Stata

```
clear all
set seed 123
set obs 1000

* Generate baseline covariates and treatment
generate L0 = rnormal()
generate A0 = rbinomial(1, invlogit(0.5 * L0))

* Generate follow-up covariates and treatment
generate L1 = 0.5 * L0 + 0.8 * A0 + rnormal()
generate A1 = rbinomial(1, invlogit(0.5 * L1))

* Generate outcome
generate Y = 0.7 * L1 + 1.2 * A1 + rnormal()

list L0 A0 L1 A1 Y in 1/6
```

A DAG helps illustrate the challenge of time-varying confounding. Consider the following structure over two time points:

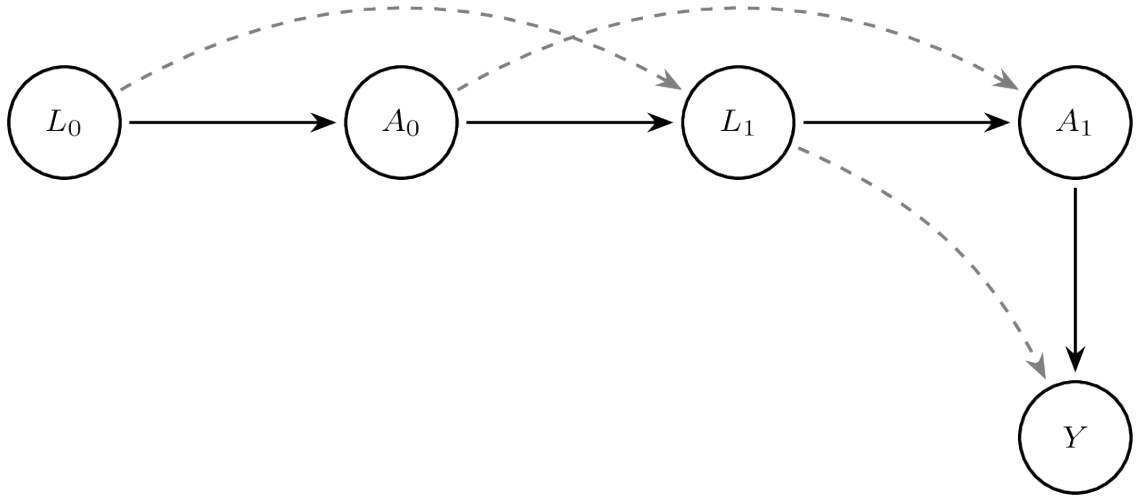

Figure 7.1: Time-varying confounding DAG: L_1 is influenced by prior treatment A_0 and affects subsequent treatment A_1, creating treatment-confounder feedback.

In this DAG, L_1 is influenced by A_0, and A_1 is influenced by L_1, which in turn affects the outcome Y. Adjusting for L_1 blocks part of the effect of A_0, resulting in bias.

Box 6.2: Bias from naive regression adjustment for a time-varying confounder

7.1.6 R

```r
# Using the data generated in Box 6.1
# Standard regression adjusting for L1
model_naive <- lm(Y ~ A0 + A1 + L1, data = data)
summary(model_naive)
```

7.1.7 Stata

```
* Using the data generated in Box 6.1
* Standard regression adjusting for L1
regress Y A0 A1 L1
```

This regression model adjusts for L_1, which is affected by A_0, leading to biased estimates for the effect of A_0. Methods such as IPTW avoid this bias by reweighting rather than conditioning.

Understanding time-varying confounding is critical for correctly estimating causal effects in longitudinal studies. The remainder of this chapter will focus on formalizing these ideas and presenting estimators that properly handle this complexity.

7.1.8 G-computation and the G-formula for Longitudinal Data

The g-formula, introduced by Robins (1986), provides a way to estimate causal effects from longitudinal data in the presence of time-varying confounding. In longitudinal studies, standard regression methods often fail to provide valid causal estimates when time-varying confounders are also affected by prior treatment. This situation arises commonly in practice, especially in observational studies where treatment decisions are made based on intermediate health status, which itself may be influenced by earlier treatments. The g-formula resolves this problem by using a model-based standardisation approach to compute counterfactual outcomes under hypothetical treatment interventions.

7.1.8.1 Motivating Setting

Suppose data are collected over K time points. At each time t $(t = 0, \ldots, K)$, we observe time-varying covariates L_t, treatment variables A_t, and eventually an outcome Y. The goal is to estimate the causal effect of a treatment strategy $\bar{a} = (a_0, a_1, \ldots, a_K)$ on the outcome Y.

Let $\bar{A}_t = (A_0, A_1, \ldots, A_t)$ and $\bar{L}_t = (L_0, L_1, \ldots, L_t)$ denote the treatment and covariate history, respectively. The counterfactual outcome under a specific treatment strategy $\bar{a}$ is denoted $Y^{\bar{a}}$. We aim to estimate $E[Y^{\bar{a}}]$, the expected outcome if all individuals had received treatment strategy $\bar{a}$.

7.1.8.2 The Longitudinal G-formula

The g-formula expresses this quantity as:

$$E[Y^{\bar{a}}] = \int_{\bar{l}} E[Y \mid \bar{A}_K = \bar{a}, \bar{L}_K = \bar{l}] \prod_{t=0}^{K} f(L_t \mid \bar{L}_{t-1} = \bar{l}_{t-1}, \bar{A}_t = \bar{a}_t) d\bar{l}$$

In practice, this formula is approximated using parametric regression models and Monte Carlo simulation. It decomposes the joint distribution of the data into a sequence of conditional models. Each covariate and the outcome are modeled conditionally on the observed treatment and covariate history.

7.1.8.3 Step-by-Step Estimation Procedure

1. **Model the data-generating process.** Specify models for each time-varying covariate L_t and the final outcome Y, conditioning on previous covariates and treatments.
2. **Simulate counterfactuals.** For each individual, simulate covariates and outcomes under a specific intervention strategy $\bar{a}$. This requires sequentially predicting covariates L_t using the fitted models, and using these to predict the outcome Y.
3. **Average the counterfactual outcomes.** Estimate $E[Y^{\bar{a}}]$ by averaging the predicted counterfactual outcomes across the simulated pseudo-population.

7.1.8.4 Illustrative Example

We now provide a full example using simulated data with two time points.

Box 6.3: Simulating data for the longitudinal g-computation example

7.1.9 R

```r
# Simulate longitudinal data
set.seed(42)
n <- 1000
L0 <- rnorm(n)
A0 <- rbinom(n, 1, plogis(0.3 * L0))
L1 <- rnorm(n, mean = 0.5 * A0 + 0.6 * L0)
A1 <- rbinom(n, 1, plogis(0.4 * A0 + 0.7 * L1))
Y <- rbinom(n, 1, plogis(-1 + 0.8 * A1 + 0.6 * L1 + 0.3 * A0))

long_data <- data.frame(L0, A0, L1, A1, Y)
```

7.1.10 Stata

```
clear all
set seed 42
set obs 1000

* Simulate longitudinal data
generate L0 = rnormal()
generate A0 = rbinomial(1, invlogit(0.3 * L0))
generate L1 = rnormal(0.5 * A0 + 0.6 * L0, 1)
generate A1 = rbinomial(1, invlogit(0.4 * A0 + 0.7 * L1))
generate Y = rbinomial(1, invlogit(-1 + 0.8 * A1 + 0.6 * L1 + 0.3 * A0))
```

We now implement parametric g-computation to estimate the outcome under the intervention $A_0 = 1, A_1 = 1$.

Box 6.4: Manual parametric g-computation for a binary outcome

7.1.11 R

```r
# Step 1: Fit models for L1 and Y
model_L1 <- lm(L1 ~ A0 + L0, data = long_data)
model_Y  <- glm(Y ~ A0 + A1 + L1, data = long_data, family = binomial())

# Step 2: Simulate under intervention A0 = A1 = 1
new_data <- long_data
new_data$A0 <- 1
new_data$A1 <- 1
new_data$L1 <- predict(model_L1, newdata = new_data)
```

```r
new_data$Y_hat <- predict(model_Y, newdata = new_data, type = "response")

# Step 3: Estimate E[Y^{a}]
g_formula_estimate <- mean(new_data$Y_hat)
print(g_formula_estimate)

# Bootstrap for 95% CI
boot_est <- replicate(500, {
  idx <- sample(nrow(long_data), replace = TRUE)
  boot_data <- long_data[idx, ]
  m_L1 <- lm(L1 ~ A0 + L0, data = boot_data)
  m_Y  <- glm(Y ~ A0 + A1 + L1, data = boot_data, family = binomial())
  nd <- boot_data
  nd$A0 <- 1; nd$A1 <- 1
  nd$L1 <- predict(m_L1, newdata = nd)
  nd$Y_hat <- predict(m_Y, newdata = nd, type = "response")
  mean(nd$Y_hat)
})
quantile(boot_est, c(0.025, 0.975))
```

7.1.12 Stata

```
* Step 1: Fit models for L1 and Y
glm L1 A0 L0, family(gaussian) link(identity)
logit Y A0 A1 L1

* Step 2-3: Predict under intervention A0 = 1, A1 = 1
preserve
  replace A0 = 1
  replace A1 = 1
  predict double L1_hat, xb
  replace L1 = L1_hat
  predict double Y_hat, pr
  summarize Y_hat
restore

* Bootstrap for 95% CI
capture program drop gcomp_boot
program define gcomp_boot, rclass
  preserve
    bsample
```

```
    glm L1 A0 L0, family(gaussian) link(identity)
    logit Y A0 A1 L1
    replace A0 = 1
    replace A1 = 1
    predict double L1_hat, xb
    replace L1 = L1_hat
    predict double Y_hat, pr
    summarize Y_hat
    return scalar psi = r(mean)
  restore
end
bootstrap r(psi), reps(500) seed(42): gcomp_boot
estat bootstrap, all
```

To assess uncertainty, nonparametric bootstrap can be used to obtain confidence intervals.

7.1.12.1 Assumptions

The validity of the g-formula relies on three assumptions: - **Consistency:** The observed outcome equals the counterfactual outcome for the observed treatment history. - **Positivity:** There is a positive probability of receiving each treatment level for all covariate patterns. - **Sequential Exchangeability (No Unmeasured Confounding):**

$$Y^{\bar{a}} \perp A_t \mid \bar{A}_{t-1}, \bar{L}_t \quad \text{for all } t$$

Interpretation and Extensions

The g-formula provides a consistent estimate of the counterfactual mean outcome under a static or dynamic treatment strategy. It generalizes standardization by incorporating time-varying covariates and treatments. The method can also be extended to simulate more complex interventions, such as treatment rules that depend on patient history.

While powerful, the g-formula is sensitive to model misspecification. Every component of the data-generating process must be modeled correctly. In practice, this often requires rich data and careful model diagnostics. Flexible modeling approaches, such as Super Learner, can improve robustness to misspecification.

7.1.12.2 Summary

The longitudinal g-formula is a foundational method for estimating causal effects in the presence of time-varying confounding. It is well-suited for estimating population-level effects of

hypothetical interventions in longitudinal data, especially when conventional regression methods are biased due to treatment-confounder feedback. Although computationally intensive and model-dependent, it serves as a conceptual and methodological precursor to more robust estimators like TMLE.

7.1.13 Marginal Structural Models

Marginal Structural Models (MSMs) are a class of causal models designed to estimate the marginal effect of a time-varying treatment or exposure on an outcome in the presence of time-varying confounding. Unlike traditional regression models, which condition on time-varying confounders that may be affected by prior treatment, MSMs allow for valid causal inference by modeling the treatment-outcome relationship marginally – that is, without conditioning on such intermediate variables.

In longitudinal settings, the central challenge arises when confounders are affected by prior treatment. Adjusting for these confounders in a standard regression model can introduce bias, because it blocks part of the causal pathway from treatment to outcome. MSMs overcome this problem by modeling the counterfactual mean outcomes under different treatment regimes, while using inverse probability weighting (IPW) to account for confounding.

Let $\bar{A}_t = (A_0, A_1, \ldots, A_t)$ be the history of treatments up to time t, and let Y be the outcome measured at a final time point T. The goal of an MSM is to estimate:

$$\mathbb{E}[Y^{\bar{a}}]$$

for different treatment regimens $\bar{a}$. These counterfactual means are modeled directly, without conditioning on intermediate covariates.

A typical form of an MSM is:

$$\mathbb{E}[Y^{\bar{a}}] = \beta_0 + \beta_1 a_0 + \beta_2 a_1 + \cdots + \beta_T a_T$$

This model expresses the expected counterfactual outcome as a function of the treatment history. It can be generalized to allow for interactions, nonlinear effects, or cumulative dose effects. For binary outcomes, the model might be fit on the log-odds scale using a logistic link.

Because the true counterfactuals are not observed, MSMs are estimated using the observed outcomes weighted by the inverse probability of treatment. Inverse Probability Weighting (IPW) is a method used to address time-varying confounding in longitudinal observational studies. It reweights the sample to create a pseudo-population in which treatment assignment at each time point is independent of past confounders. This approach enables unbiased estimation of marginal causal effects when traditional regression adjustment would fail due to feedback between treatment and covariates.

In longitudinal studies, confounders measured at time t, denoted L_t, often affect subsequent treatment A_t and are themselves influenced by prior treatment A_{t-1}. This dual role of L_t as both a confounder and an intermediate variable invalidates standard regression techniques, which cannot properly adjust for such variables without introducing bias.

To estimate the effect of a treatment regime $\bar{a} = (a_0, \dots, a_T)$, we use IPW to account for the time-varying nature of both treatment and confounding. This involves modeling the treatment mechanism over time and assigning weights to each individual based on the inverse of their probability of receiving their observed treatment history, conditional on their covariate history.

Let $\hat{e}_t(W_t) = P(A_t = a_t \mid \bar{A}_{t-1}, \bar{L}_t)$ denote the estimated probability of treatment at time t, given past treatment and covariate history. The unstabilized IP weight is:

$$W_i = \prod_{t=0}^{T} \frac{1}{P(A_t = a_t \mid \bar{A}_{t-1}, \bar{L}_t)}.$$

Stabilized weights are often preferred to reduce variance and improve finite sample performance. They are constructed by placing a numerator that only depends on baseline or prior covariates:

$$W_i^{stab} = \prod_{t=0}^{T} \frac{P(A_t = a_t \mid \bar{A}_{t-1})}{P(A_t = a_t \mid \bar{A}_{t-1}, \bar{L}_t)}.$$

More generally,

$$W_i = \prod_{t=0}^{T} \frac{f(A_t = a_t^i \mid \bar{A}_{t-1}^i)}{f(A_t = a_t^i \mid \bar{L}_t^i, \bar{A}_{t-1}^i)}$$

These stabilized weights create a pseudo-population in which treatment is independent of confounders, allowing consistent estimation of the MSM parameters via weighted regression.

1. Estimate the treatment model at each time point using logistic regression or machine learning to obtain probabilities $P(A_t \mid \bar{L}_t, \bar{A}_{t-1})$.
2. Compute stabilized IP weights for each individual across all time points.
3. Fit the MSM using a weighted regression model, regressing the outcome on treatment history, using the IP weights.

The parameters in an MSM can be interpreted as the causal effect of treatment on the outcome, averaged over the population. For example, β_1 in the above model reflects the marginal causal effect of treatment at time $t = 0$, assuming correct model specification and no unmeasured confounding.

MSMs can be extended to:

- Estimate causal risk differences, odds ratios, or hazard ratios
- Allow for dynamic treatment rules or effect modification
- Handle survival outcomes using marginal structural Cox models

- Use flexible models (e.g., SuperLearner) for treatment mechanism

MSMs rely on key assumptions:

- **No unmeasured confounding**: All confounders of the treatment-outcome relationship must be measured.
- **Positivity**: There must be a non-zero probability of receiving each treatment at every level of confounders.
- **Correct model specification**: For consistent estimation, both the treatment model and the MSM must be correctly specified.

If these assumptions are violated, the MSM estimates may be biased or unstable. Weight truncation or flexible machine learning methods are often used to mitigate some of these issues.

7.1.14 Weighted Regression to Estimate the Causal Effect

Box 6.5: Computing stabilized inverse probability of treatment weights

7.1.15 R

```r
# Simulated longitudinal data
set.seed(123)
n <- 1000
L0 <- rnorm(n)
A0 <- rbinom(n, 1, plogis(0.4 * L0))
L1 <- 0.7 * L0 + 0.9 * A0 + rnorm(n)
A1 <- rbinom(n, 1, plogis(0.4 * L1))
Y  <- 1.5 * A0 + 1.2 * A1 + 0.5 * L1 + rnorm(n)

data <- data.frame(L0, A0, L1, A1, Y)

# Numerator models (simpler)
num_A0 <- glm(A0 ~ L0, family = binomial, data = data)
num_A1 <- glm(A1 ~ A0 + L0, family = binomial, data = data)

# Denominator models (full)
den_A0 <- glm(A0 ~ L0, family = binomial, data = data)
den_A1 <- glm(A1 ~ A0 + L0 + L1, family = binomial, data = data)

# Get predicted probabilities
```

```r
pnum_A0 <- predict(num_A0, type = "response")
pnum_A1 <- predict(num_A1, type = "response")
pden_A0 <- predict(den_A0, type = "response")
pden_A1 <- predict(den_A1, type = "response")

# Compute stabilized weights
w_A0 <- ifelse(data$A0 == 1, pnum_A0 / pden_A0, (1 - pnum_A0) / (1 - pden_A0))
w_A1 <- ifelse(data$A1 == 1, pnum_A1 / pden_A1, (1 - pnum_A1) / (1 - pden_A1))
data$sw <- w_A0 * w_A1
summary(data$sw)
```

7.1.16 Stata

```
* Simulate data for MSM estimation
clear all
set seed 123
set obs 1000
generate L0 = rnormal()
generate A0 = rbinomial(1, invlogit(0.4 * L0))
generate L1 = 0.7 * L0 + 0.9 * A0 + rnormal()
generate A1 = rbinomial(1, invlogit(0.4 * L1))
generate Y  = 1.5 * A0 + 1.2 * A1 + 0.5 * L1 + rnormal()

* Numerator models (baseline only)
logit A0 L0, vce(robust) nolog
predict double num_A0, pr
logit A1 A0 L0, vce(robust) nolog
predict double num_A1, pr

* Denominator models (full, including time-varying confounders)
logit A0 L0, vce(robust) nolog
predict double den_A0, pr
logit A1 A0 L0 L1, vce(robust) nolog
predict double den_A1, pr

* Compute stabilized weights
generate sw_A0 = num_A0 / den_A0 if A0 == 1
replace  sw_A0 = (1 - num_A0) / (1 - den_A0) if A0 == 0
generate sw_A1 = num_A1 / den_A1 if A1 == 1
replace  sw_A1 = (1 - num_A1) / (1 - den_A1) if A1 == 0
```

```
generate sw = sw_A0 * sw_A1
summarize sw
```

Note: The complete longitudinal analysis workflow (G-formula, IPTW, MSM, AIPTW, TMLE) is available at github.com/migariane/TutorialComputationalCausalInferenceEstimators.

Once the stabilized weights are estimated, they are used to fit a weighted regression model to estimate the marginal causal effect:

Box 6.6: Fitting a weighted marginal structural model

7.1.17 R

```
# Fit weighted MSM
model_weighted <- glm(Y ~ A0 + A1, weights = sw, data = data)
summary(model_weighted)

# Bootstrap 95% CI
boot_ate <- replicate(500, {
  idx <- sample(nrow(data), replace = TRUE)
  d <- data[idx, ]
  num_A0 <- glm(A0 ~ L0, family = binomial, data = d)
  num_A1 <- glm(A1 ~ A0 + L0, family = binomial, data = d)
  den_A0 <- glm(A0 ~ L0, family = binomial, data = d)
  den_A1 <- glm(A1 ~ A0 + L0 + L1, family = binomial, data = d)
  pn0 <- predict(num_A0, type = "response")
  pn1 <- predict(num_A1, type = "response")
  pd0 <- predict(den_A0, type = "response")
  pd1 <- predict(den_A1, type = "response")
  w0 <- ifelse(d$A0 == 1, pn0/pd0, (1-pn0)/(1-pd0))
  w1 <- ifelse(d$A1 == 1, pn1/pd1, (1-pn1)/(1-pd1))
  d$sw <- w0 * w1
  coef(glm(Y ~ A0 + A1, weights = sw, data = d))["A1"]
})
quantile(boot_ate, c(0.025, 0.975))
```

7.1.18 Stata

```
* Fit marginal structural model with stabilized weights
regress Y A0 A1 [pw = sw], vce(robust)

* Bootstrap the 95% confidence intervals
capture program drop ATE
program define ATE, rclass
  capture drop num_A0 num_A1 den_A0 den_A1 sw_A0 sw_A1 sw
  * Numerator models
  logit A0 L0, vce(robust) nolog
  predict double num_A0, pr
  logit A1 A0 L0, vce(robust) nolog
  predict double num_A1, pr
  * Denominator models
  logit A0 L0, vce(robust) nolog
  predict double den_A0, pr
  logit A1 A0 L0 L1, vce(robust) nolog
  predict double den_A1, pr
  * Stabilized weights
  generate sw_A0 = num_A0 / den_A0 if A0 == 1
  replace  sw_A0 = (1 - num_A0) / (1 - den_A0) if A0 == 0
  generate sw_A1 = num_A1 / den_A1 if A1 == 1
  replace  sw_A1 = (1 - num_A1) / (1 - den_A1) if A1 == 0
  generate sw = sw_A0 * sw_A1
  * MSM
  regress Y A0 A1 [pw = sw], vce(robust)
  return scalar ate = _b[A1]
end
bootstrap r(ate), reps(500) seed(1): ATE
estat bootstrap, all
```

Note: Full longitudinal workflow available at github.com/migariane/TutorialComputationalCausalInferenceEstim

This regression provides an estimate of the average treatment effect under the assumption of no unmeasured confounding and correct model specification for the treatment assignment.

Inverse probability weighting is foundational for marginal structural models and plays a central role in modern causal inference for longitudinal data. While powerful, IPW requires careful diagnostics for positivity violations and extreme weights, which are discussed in later sections.

7.1.19 Double Robust Methods for Longitudinal Data

The methods presented so far — IPTW and the g-formula — each rely on a single model being correctly specified. IPTW requires a correct treatment model; the g-formula requires correct outcome and covariate models. In practice, we rarely know which model is correct. **Doubly robust (DR) estimators** solve this problem by combining both modeling approaches: they remain consistent if *either* the outcome model *or* the treatment model is correctly specified — not necessarily both.

For longitudinal data, this property is especially valuable because the sequential nature of treatment and confounding multiplies the opportunities for model misspecification. Two main classes of doubly robust estimators are used in longitudinal settings: **AIPTW** (Augmented Inverse Probability of Treatment Weighting), which augments IPTW with outcome regression predictions, and **LTMLE** (Longitudinal Targeted Maximum Likelihood Estimation), which adds a targeting step to optimize the bias-variance trade-off for the parameter of interest. Both methods inherit the double-robustness property while differing in their implementation complexity and finite-sample behaviour.

Box 6.7: AIPTW for longitudinal data: IPTW with outcome regression

7.1.20 R

```r
# Using the data structure from Box 6.5 (L0, A0, L1, A1, Y)
set.seed(123)
n <- 1000
L0 <- rnorm(n)
A0 <- rbinom(n, 1, plogis(0.4 * L0))
L1 <- 0.7 * L0 + 0.9 * A0 + rnorm(n)
A1 <- rbinom(n, 1, plogis(0.4 * L1))
Y  <- 1.5 * A0 + 1.2 * A1 + 0.5 * L1 + rnorm(n)
data <- data.frame(L0, A0, L1, A1, Y)

# Step 1: Fit outcome models Q(A0, A1, L0, L1)
Q_model <- lm(Y ~ A0 * A1 + L0 + L1, data = data)

# Step 2: Fit treatment models g(A0 | L0) and g(A1 | A0, L0, L1)
g0 <- glm(A0 ~ L0, family = binomial, data = data)
g1 <- glm(A1 ~ A0 + L0 + L1, family = binomial, data = data)

# Step 3: Predict outcomes and propensity scores
data$g0_hat <- predict(g0, type = "response")
data$g1_hat <- predict(g1, type = "response")
```

```r
# Counterfactual predictions under A0=1, A1=1
data_cf <- data
data_cf$A0 <- 1; data_cf$A1 <- 1
Q_cf <- predict(Q_model, newdata = data_cf)
Q_obs <- predict(Q_model, newdata = data)

# Step 4: Compute doubly robust estimator
# IPTW component
iptw_wt <- (data$A0 / data$g0_hat) * (data$A1 / data$g1_hat)
# Augmentation: add regression predictions to residuals
dr_component <- iptw_wt * (data$Y - Q_obs) + Q_cf

# ATE under always-treated vs observed
psi_dr <- mean(Q_cf)  # E[Y^{a=(1,1)}]
print(psi_dr)

# Bootstrap for inference
boot_dr <- replicate(500, {
  idx <- sample(n, replace = TRUE)
  d <- data[idx, ]
  Q_m <- lm(Y ~ A0 * A1 + L0 + L1, data = d)
  g0_m <- glm(A0 ~ L0, family = binomial, data = d)
  g1_m <- glm(A1 ~ A0 + L0 + L1, family = binomial, data = d)
  d_cf <- d; d_cf$A0 <- 1; d_cf$A1 <- 1
  mean(predict(Q_m, newdata = d_cf))
})
quantile(boot_dr, c(0.025, 0.975))
```

7.1.21 Stata

```
* Simulate data for AIPTW example
clear all
set seed 123
set obs 1000
generate L0 = rnormal()
generate A0 = rbinomial(1, invlogit(0.4 * L0))
generate L1 = 0.7 * L0 + 0.9 * A0 + rnormal()
generate A1 = rbinomial(1, invlogit(0.4 * L1))
generate Y  = 1.5 * A0 + 1.2 * A1 + 0.5 * L1 + rnormal()
```

```
* Step 1: Fit outcome model Q(A0, A1, L0, L1)
regress Y c.A0##c.A1 L0 L1
predict double Q_obs, xb

* Step 2: Fit treatment models
logit A0 L0, vce(robust) nolog
predict double g0_hat, pr
logit A1 A0 L0 L1, vce(robust) nolog
predict double g1_hat, pr

* Step 3: Counterfactual predictions under A0=1, A1=1
preserve
  replace A0 = 1
  replace A1 = 1
  predict double Q_cf, xb
  summarize Q_cf
  return list
restore

* Bootstrap for DR estimator
capture program drop aiptw_boot
program define aiptw_boot, rclass
  preserve
    bsample
    regress Y c.A0##c.A1 L0 L1
    predict double Q_obs, xb
    logit A0 L0, vce(robust) nolog
    predict double g0_hat, pr
    logit A1 A0 L0 L1, vce(robust) nolog
    predict double g1_hat, pr
    replace A0 = 1
    replace A1 = 1
    predict double Q_cf, xb
    summarize Q_cf
    return scalar psi = r(mean)
  restore
end
bootstrap r(psi), reps(500) seed(42): aiptw_boot
estat bootstrap, all
```

Box 6.8: LTMLE: Manual step-by-step implementation (two time points)

7.1.22 R

```r
# Simulate data
set.seed(123)
n <- 500
L0 <- rbinom(n, 1, 0.5)
A0 <- rbinom(n, 1, plogis(-0.5 + 0.8 * L0))
L1 <- rbinom(n, 1, plogis(0.3 * A0 + 0.4 * L0))
A1 <- rbinom(n, 1, plogis(-0.4 + 0.5 * L1 + 0.3 * A0))
Y <- rbinom(n, 1, plogis(-1 + 0.6 * A1 + 0.5 * L1 + 0.2 * A0))
data <- data.frame(L0, A0, L1, A1, Y)

# Step 1: Estimate Q2 (E[Y | A0, A1, L0, L1]) -- initial outcome model at t=1
Q2_model <- glm(Y ~ A0 + A1 + L0 + L1, data = data, family = binomial)
Q2_pred <- predict(Q2_model, type = "response")

# Step 2: Estimate Q1 (E[Q2 | A0, L0, L1]) -- conditional expectation at t=0
data$Q2 <- Q2_pred
Q1_model <- glm(Q2 ~ A0 + L0 + L1, data = data, family = binomial)
Q1_pred <- predict(Q1_model, type = "response")

# Step 3: Estimate g models (treatment mechanisms)
g1 <- glm(A1 ~ A0 + L0 + L1, data = data, family = binomial)
g1_pred <- predict(g1, type = "response")
g0 <- glm(A0 ~ L0, data = data, family = binomial)
g0_pred <- predict(g0, type = "response")

# Step 4: Calculate clever covariates
# For t=1 (A1 = 1 under static intervention):
data$H1 <- ifelse(A1 == 1, 1 / g1_pred, 0)
# For t=0 (A0 = 1):
data$H0 <- ifelse(A0 == 1, 1 / g0_pred, 0)

# Step 5: Targeting -- update Q2 using clever covariate H1
fluct_model <- glm(Y ~ -1 + offset(qlogis(Q2_pred)) + H1,
                   data = data, family = binomial)
epsilon <- coef(fluct_model)["H1"]
Q2_star <- plogis(qlogis(Q2_pred) + epsilon * data$H1)

# Step 6: Counterfactual prediction under A0=1, A1=1
new_data <- data
new_data$A0 <- 1; new_data$A1 <- 1
```

```r
Q2_cf <- predict(Q2_model, newdata = new_data, type = "response")
ATE <- mean(Q2_cf) - mean(Q2_pred)
ATE
```

7.1.23 Stata

```
* Simulate longitudinal data for LTMLE
clear all
set seed 123
set obs 500
generate L0 = rbinomial(1, 0.5)
generate A0 = rbinomial(1, invlogit(-0.5 + 0.8 * L0))
generate L1 = rbinomial(1, invlogit(0.3 * A0 + 0.4 * L0))
generate A1 = rbinomial(1, invlogit(-0.4 + 0.5 * L1 + 0.3 * A0))
generate Y  = rbinomial(1, invlogit(-1 + 0.6 * A1 + 0.5 * L1 + 0.2 * A0))

* Step 1: Estimate initial outcome model Q2
logit Y A0 A1 L0 L1
predict double Q2_pred, pr

* Step 2: Estimate Q1 (E[Q2 | A0, L0, L1])
logit Q2_pred A0 L0 L1
predict double Q1_pred, pr

* Step 3: Estimate treatment mechanisms
logit A1 A0 L0 L1
predict double g1_pred, pr
logit A0 L0
predict double g0_pred, pr

* Step 4: Clever covariate for targeting at t=1 (A1=1 intervention)
generate H1 = 1 / g1_pred if A1 == 1
replace  H1 = 0 if A1 == 0

* Step 5: Targeting step -- update Q2
generate logit_Q2 = logit(Q2_pred)
logit Y, offset(logit_Q2)
* The coefficient on H1 in a model with offset gives epsilon
constraint 1 H1 = 1  // dummy to get coefficient
logit Y H1, offset(logit_Q2) nolog
* Manual update (epsilon extracted from coefficient)
```

```
matrix eps = e(b)
local epsilon = eps[1,1]
generate double Q2_star = invlogit(logit_Q2 + `epsilon' * H1)

* Step 6: Counterfactual under always-treated
preserve
  replace A0 = 1
  replace A1 = 1
  predict double Q2_cf, pr
  summarize Q2_cf
restore
```

This provides a manual implementation of TMLE for a simple longitudinal setup. The key insight is that the targeting step uses a clever covariate to update the initial outcome model, removing bias for the parameter of interest.

Box 6.9: LTMLE using the ltmle R package

7.1.24 R

```
library(ltmle)

# Simulate data compatible with ltmle format
set.seed(123)
n <- 500
W <- rnorm(n)
L0 <- rbinom(n, 1, plogis(0.5 * W))
A0 <- rbinom(n, 1, plogis(-0.5 + 0.8 * L0))
L1 <- rbinom(n, 1, plogis(0.3 * A0 + 0.4 * L0))
A1 <- rbinom(n, 1, plogis(-0.4 + 0.5 * L1 + 0.3 * A0))
Y <- rbinom(n, 1, plogis(-1 + 0.6 * A1 + 0.5 * L1 + 0.2 * A0 + 0.1 * W))

ltmle_data <- data.frame(W, L0, A0, L1, A1, Y)

# Define nodes
Anodes <- c("A0", "A1")
Lnodes <- c("L0", "L1")
Ynodes <- "Y"

# Fit LTMLE under a static intervention (always treat)
result <- ltmle(data = ltmle_data,
```

```
                Anodes = Anodes,
                Lnodes = Lnodes,
                Ynodes = Ynodes,
                abar = c(1, 1),
                SL.library = c("SL.glm", "SL.gam", "SL.mean"))

summary(result)
```

7.1.25 Stata

```
* Longitudinal TMLE is available in R via the ltmle package.
* Stata users can implement the manual targeting algorithm shown in Box 6.8
* using logit, predict, and the clever covariate construction.
* For production use, the R ltmle package or the eltmle Stata module
* (for single-time-point TMLE) are recommended.
```

This code estimates the expected outcome had everyone received treatment at both time points using SuperLearner to fit nuisance models. The output includes point estimates, standard errors, and confidence intervals.

In summary, doubly robust methods provide a powerful framework for estimating causal effects in longitudinal studies. AIPTW offers a simple, closed-form estimator that combines IPTW with outcome regression. LTMLE extends this with a targeting step that optimizes the bias-variance tradeoff, providing semiparametric efficiency when both models are correctly specified. Together, they offer practical and theoretically grounded tools for handling time-varying confounding.

7.2 Part II: Time-to-event outcome

7.2.1 Brief Introduction to Censoring and Competing Risks

In longitudinal studies, participants are followed over time to assess the effect of treatments or exposures on outcomes of interest. However, complete follow-up for all individuals is rarely achieved. When an individual's data become unavailable before the outcome is observed, the data are said to be *censored*. Censoring can arise from loss to follow-up, study dropout, or administrative end of study.

Censoring is problematic because it introduces missing data in the outcome and may bias effect estimates if the reason for censoring is related to treatment or covariates. The standard assumption for valid inference in the presence of censoring is *independent censoring*–that is, the

probability of being censored at any time depends only on observed covariates and treatment history, not on unmeasured factors or the potential outcomes themselves.

Let C_t be the indicator that a subject is censored at time t, and define the observed data structure as:

$$O = (L_0, A_0, C_0, L_1, A_1, C_1, \ldots, L_T, A_T, C_T, Y).$$

A subject contributes information up until the time of censoring. If censoring depends on time-varying covariates affected by prior treatment (e.g., CD4 count in HIV studies), then standard complete-case analyses can be biased. To address this, researchers often apply *inverse probability of censoring weighting* (IPCW), where each subject's contribution is reweighted by the inverse of their probability of remaining uncensored.

7.2.2 Competing Risks

In some longitudinal studies, individuals may experience one of several mutually exclusive outcomes. When the occurrence of one event precludes the occurrence of the primary event of interest, it is called a *competing risk*. For example, in a study of cardiovascular mortality, death from cancer acts as a competing risk.

Standard survival analysis methods, such as the Kaplan-Meier estimator or Cox proportional hazards model, treat competing events as censoring, which can overestimate the cumulative incidence of the primary event. Instead, the correct estimand is often the *cumulative incidence function* (CIF), which estimates the marginal probability of experiencing each event over time.

Box 6.10: Competing risks analysis with cumulative incidence functions

7.2.3 R

```r
library(cmprsk)
# Simulated data: time = event time, status = event type (0=censoring, 1=event, 2=competing r
set.seed(123)
n <- 500
time <- rexp(n, rate = 0.1)
status <- sample(0:2, n, replace = TRUE, prob = c(0.3, 0.5, 0.2))
group <- rbinom(n, 1, 0.5)

# Estimate cumulative incidence function by group
cif <- cuminc(time, status, group)
plot(cif, lty = 1:2, col = c("blue", "red"))
```

7.2.4 Stata

```
* Simulate competing risks data
clear all
set seed 123
set obs 500
generate time = rexponential(0.1)
generate status = runiform()
generate group = rbinomial(1, 0.5)
* Recode: 0 = censored, 1 = event of interest, 2 = competing event
recode status (0/0.3 = 0) (0.3/0.8 = 1) (0.8/1 = 2)

* Declare survival data with competing risks
stset time, failure(status == 1)

* Cumulative incidence function by group
stcompet cif = ci, compet1(2) by(group)
* Plot cumulative incidence
twoway (line cif time if group == 0, sort) ///
       (line cif time if group == 1, sort), ///
       ytitle("Cumulative Incidence") xtitle("Time")
```

7.2.5 Inverse Probability of Censoring Weights (IPCW)

Box 6.11: Inverse probability of censoring weights

7.2.6 R

```
# Simulate dropout indicator
library(survival)
set.seed(456)
n <- 500
L0 <- rnorm(n)
A0 <- rbinom(n, 1, plogis(0.3 * L0))
C <- rbinom(n, 1, plogis(0.4 * A0 - 0.2 * L0)) # Censoring indicator (1 = censored)

# Estimate probability of remaining uncensored
ipcw_model <- glm(C ~ A0 + L0, family = binomial)
p_uncensored <- 1 - predict(ipcw_model, type = "response")
```

```r
ipcw_weights <- 1 / p_uncensored
summary(ipcw_weights)
```

7.2.7 Stata

```
* Simulate data for IPCW
clear all
set seed 456
set obs 500
generate L0 = rnormal()
generate A0 = rbinomial(1, invlogit(0.3 * L0))
generate C  = rbinomial(1, invlogit(0.4 * A0 - 0.2 * L0))

* Estimate probability of remaining uncensored
logit C A0 L0
predict double p_cens, pr
generate p_uncens = 1 - p_cens
generate ipcw = 1 / p_uncens
summarize ipcw
```

These weights can be used in regression or IPTW analyses to adjust for informative censoring.

Censoring and competing risks are both forms of missing data or information loss that, if ignored, can severely bias causal effect estimates. Proper methods such as IPCW or competing risk models are essential for valid inference in longitudinal studies.

7.2.8 Motivation and Examples from Survival Analysis

Time-to-event outcomes are ubiquitous in medical and epidemiological research. In studies where the outcome is the time until death, disease recurrence, or hospital readmission, researchers are often interested not only in whether a treatment affects the risk of the event but also in *when* that effect occurs and how it evolves over time. Survival analysis provides the statistical framework for answering these questions.

Consider a cohort of patients diagnosed with advanced cancer and followed over a five-year period. Treatment regimens may change in response to disease progression, adverse events, or biomarker levels—all of which are themselves influenced by prior treatment. For instance, a patient who develops severe toxicity to first-line chemotherapy may be switched to a less intensive regimen, and this decision depends on their current performance status, which was affected by the initial treatment. In this setting, both the treatment and the confounding

variables vary over time, and the outcome—time to death—is subject to right censoring because some patients are alive at the end of follow-up or are lost to follow-up.

Other motivating examples include:

- **Antiretroviral therapy (ART) in HIV-positive individuals:** CD4 count and viral load are time-varying confounders that predict both the decision to initiate or modify ART and the risk of AIDS-related mortality. These markers are themselves affected by prior ART exposure, creating the treatment-confounder feedback loop described in Part I.
- **Blood pressure management in hypertensive patients:** Antihypertensive medications are titrated based on current blood pressure readings, which are influenced by prior treatment. The outcome—time to cardiovascular events—may be censored by non-cardiovascular death or end of study.
- **Smoking cessation and cancer recurrence:** Smoking status changes over time and may be influenced by psychosocial support interventions. The time to cancer recurrence is subject to competing risks (e.g., death from other causes).
- **Long-term oxygen therapy in COPD patients:** Oxygen therapy decisions depend on current oxygen saturation levels, which decline over time and are affected by treatment adherence. Mortality is the primary endpoint, but follow-up may be incomplete.

In each of these examples, the data structure consists of repeated measurements of treatment, confounders, and survival status over discrete time intervals. Standard survival methods such as the Cox proportional hazards model—which typically include only baseline covariates—cannot adequately address the time-varying confounding introduced by treatment-confounder feedback. Specialised methods that extend the g-formula and TMLE to the survival setting are therefore required.

7.2.9 Notation and Assumptions for Survival Data

To formalise causal inference for time-to-event outcomes, we extend the longitudinal notation introduced in Part I. Let the time axis be divided into discrete intervals $t = 1, 2, \ldots, K$, where K is the maximum follow-up time. For each individual, we observe:

- A_t: treatment or exposure at time t
- L_t: vector of time-varying covariates measured at the start of interval t
- C_t: indicator that the individual is censored during interval t (1 if censored, 0 otherwise)
- Y_t: indicator of survival status at the end of interval t, where $Y_t = 1$ if the individual is still alive (i.e., has not experienced the event) and $Y_t = 0$ if the event has occurred by time t

Let T denote the event time and C the censoring time. The observed time is $U = \min(T, C)$, and the event indicator is $\Delta = I(T \leq C)$. An individual contributes data up to the time of censoring or the event, whichever occurs first. At each time t, we only observe Y_t for individuals

who remain at risk (i.e., those for whom $Y_{t-1} = 1$ and $C_t = 0$). In discrete time, the *hazard* at time t conditional on treatment and covariate history is:

$$h(t \mid \bar{A}_t, \bar{L}_t) = P(Y_t = 0 \mid Y_{t-1} = 1, \bar{A}_t, \bar{L}_t).$$

The counterfactual survival probability at time t under treatment regime $\bar{a} = (a_0, a_1, \dots, a_t)$ is defined as:

$$S^{\bar{a}}(t) = P(Y_t^{\bar{a}} = 1),$$

where $Y_t^{\bar{a}}$ indicates survival at time t had the individual followed treatment regime $\bar{a}$.

Identification assumptions. To identify the counterfactual survival curve from observed data, the following assumptions must hold:

1. **Consistency:** The observed outcome equals the counterfactual outcome under the observed treatment history. That is, if an individual receives treatment history $\bar{A}_t = \bar{a}_t$, then $Y_t = Y_t^{\bar{a}_t}$.
2. **Sequential exchangeability (no unmeasured confounding):** At each time t, the treatment A_t is independent of future counterfactual outcomes given the observed treatment and covariate history:

$$Y_t^{\bar{a}} \perp A_t \mid \bar{A}_{t-1}, \bar{L}_t \quad \text{for all } t.$$

3. **Positivity:** At each time t, there is a non-zero probability of receiving each treatment level for every possible covariate and treatment history:

$$0 < P(A_t = a_t \mid \bar{A}_{t-1} = \bar{a}_{t-1}, \bar{L}_t = \bar{l}_t) < 1.$$

4. **Independent censoring:** The censoring mechanism is independent of the potential outcomes given the observed history. Informally, the probability of being censored at time t depends only on observed covariates and treatment history, not on the unobserved event time:

$$C_t \perp T \mid \bar{A}_t, \bar{L}_t.$$

When censoring depends on time-varying factors affected by prior treatment (e.g., sicker patients drop out), assumption (4) is violated. In such cases, inverse probability of censoring weighting (IPCW), introduced in Section 2.1, is used to reweight the uncensored observations and recover unbiased estimates. With these assumptions in place, we now turn to estimation methods, beginning with the g-formula for survival.

7.2.9.1 The Nonparametric g-formula for Survival

The nonparametric g-formula (also called the "extended Kaplan-Meier estimator") can be implemented using stratification and empirical averaging. Under this approach, survival probabilities are estimated directly from the observed data using inverse probability weighting or Monte Carlo simulation without parametric assumptions.

Let $\bar{d}$ denote a dynamic treatment rule. The nonparametric g-formula estimates:

$$\hat{S}(t \mid \bar{d}) = \frac{1}{n} \sum_{i=1}^{n} \prod_{s=0}^{t} \hat{P}(Y_{is} = 1 \mid \text{history}_i, A_{is} = d_s)$$

This is typically implemented via simulation: for each individual, generate a pseudo-population following rule $\bar{d}$, then compute the product of conditional survival probabilities across time.

7.2.9.2 The Parametric g-formula for Survival

The parametric g-formula expresses the counterfactual survival probability at time t under treatment regime $\bar{a}$ as:

$$P(Y_t^{\bar{a}} = 1) = \int_{\bar{l}_t} \prod_{s=0}^{t} P(Y_s = 1 \mid Y_{s-1} = 1, A_s = a_s, L_s = l_s) f(L_s \mid \bar{L}_{s-1}, \bar{A}_s) d\bar{l}_t$$

This formula recursively computes the survival probability at each time point, conditioning on not having failed previously. The term $P(Y_s = 1 \mid Y_{s-1} = 1, A_s, L_s)$ is the conditional survival probability given being at risk at time s, and $f(L_s \mid \cdot)$ represents the evolution of time-varying covariates.

In practice, this approach involves specifying parametric regression models for: - $P(Y_s = 1 \mid Y_{s-1} = 1, A_s, L_s)$: the outcome model (e.g., pooled logistic regression) - $f(L_s \mid A_{s-1}, L_{s-1})$: the covariate models

7.2.9.3 Worked Example

We simulate a simple time-to-event dataset with time-varying covariates and treatment:

Box 6.12: Simulating time-to-event data with time-varying covariates

7.2.10 R

```r
set.seed(123)
n <- 1000
K <- 5 # time points
L <- matrix(NA, n, K)
A <- matrix(NA, n, K)
Y <- matrix(1, n, K)

L[, 1] <- rnorm(n)
A[, 1] <- rbinom(n, 1, plogis(0.5 * L[, 1]))

for (t in 2:K) {
  L[, t] <- rnorm(n, mean = 0.4 * L[, t - 1] + 0.5 * A[, t - 1])
  A[, t] <- rbinom(n, 1, plogis(0.5 * L[, t]))
}

# Simulate event time (discrete hazard)
hazard <- matrix(NA, n, K)
for (t in 1:K) {
  hazard[, t] <- plogis(-2 + 0.6 * A[, t] + 0.5 * L[, t])
  fail <- rbinom(n, 1, hazard[, t])
  Y[, t] <- ifelse(rowSums(Y[, 1:t, drop=FALSE]) == t, 1 - fail, 0)
}

# Reshape into long format for pooled logistic regression
library(tidyr)
long_data <- data.frame(id = rep(1:n, each = K),
                        time = rep(1:K, times = n),
                        Y = as.vector(Y),
                        A = as.vector(A),
                        L = as.vector(L))

# Estimate conditional survival model
model <- glm(Y ~ time + A + L, family = binomial(), data = long_data)
```

7.2.11 Stata

```
* Simulate time-to-event data with time-varying covariates
clear all
set seed 123
set obs 1000
* Generate baseline
```

```
generate L1 = rnormal()
generate A1 = rbinomial(1, invlogit(0.5 * L1))
* Generate follow-up times (simplified long format generation)
* For brevity, expand to person-time using stset/stsplit after
* declaring survival data. See Box 6.14 for the pooled logit approach.
```

To estimate the survival curve under a hypothetical intervention $A_t = 1$ for all t:

Box 6.13: Estimating counterfactual survival curves under a treatment intervention

7.2.12 R

```
pred_data <- long_data
pred_data$A <- 1
pred_data$Y_hat <- predict(model, newdata = pred_data, type = "response")

# Compute survival curve under intervention
surv_probs <- with(pred_data, tapply(1 - Y_hat, time, mean))
surv_curve <- cumprod(surv_probs)
plot(1:K, surv_curve, type = "s", ylim = c(0,1),
     ylab = "Survival Probability", xlab = "Time")
```

7.2.13 Stata

```
* Pooled logistic regression for discrete-time survival
logit Y time A L, vce(robust)

* Predict under intervention A = 1 for all time points
preserve
  replace A = 1
  predict double Y_hat, pr
  * Compute survival curve from predicted hazards
  bysort time: egen surv = mean(1 - Y_hat)
  generate surv_curve = surv if time == 1
  replace surv_curve = surv_curve[_n-1] * surv if time > 1
  * Plot survival curve
  twoway (line surv_curve time, sort), ///
    ytitle("Survival Probability") xtitle("Time") ///
    ylabel(0(0.2)1)
restore
```

7.2.13.1 Interpretation and Uses

This approach allows estimation of the entire survival curve under hypothetical treatment strategies. It is especially useful when treatments are administered sequentially, and time-varying confounders must be appropriately adjusted.

The parametric g-formula is sensitive to model misspecification. Flexible machine learning methods and data-adaptive approaches (e.g., Super Learner) can be used to reduce bias in estimating the conditional survival and covariate models.

7.2.13.2 Summary

The g-formula can be used to estimate survival curves under complex dynamic interventions in the presence of time-varying confounding. Both parametric and nonparametric implementations are possible. The parametric approach uses pooled logistic regression to estimate conditional hazards, while the nonparametric approach relies on Monte Carlo averaging. These methods provide a foundation for evaluating intervention strategies in real-world longitudinal and survival data.

7.2.14 TMLE for Time-to-Event Data

In time-to-event analyses, right censoring complicates the estimation of causal effects. Traditional approaches, such as Cox proportional hazards models, rely on strong assumptions (e.g., proportional hazards, correct model form) and typically do not target a marginal treatment effect. Targeted Maximum Likelihood Estimation (TMLE) offers a semi-parametric, double-robust framework for estimating causal effects under time-to-event data, accounting for both time-varying treatment and censoring processes. TMLE is flexible, accommodates data-adaptive estimation (e.g., via Super Learner), and targets causal parameters directly– such as survival probabilities under static or dynamic interventions. TMLE can be adapted for survival analysis where the outcome is the time until an event occurs. In this setting, we are often interested in estimating causal effects such as differences in survival probabilities or restricted mean survival time (RMST) under hypothetical interventions.

TMLE provides a semi-parametric and doubly robust method for causal effect estimation that combines flexible machine learning with targeted bias reduction. In the context of survival data, it handles censoring and treatment assignment through a careful combination of outcome modeling, propensity score estimation, and the use of clever covariates.

Introduction and Motivation

In time-to-event analyses, right censoring complicates the estimation of causal effects. Traditional approaches, such as Cox proportional hazards models, rely on strong assumptions

(e.g., proportional hazards, correct model form) and typically do not target a marginal treatment effect. Targeted Maximum Likelihood Estimation (TMLE) offers a semi-parametric, double-robust framework for estimating causal effects under time-to-event data, accounting for both time-varying treatment and censoring processes. TMLE is flexible, accommodates data-adaptive estimation (e.g., via Super Learner), and targets causal parameters directly– such as survival probabilities under static or dynamic interventions.

Observed Data Structure

Let the observed data for n individuals be:

$$O = (W, A, \bar{L}(t), \bar{A}(t), \bar{C}(t), T, \Delta), \quad i = 1, \dots, n,$$

where:

- W: baseline covariates;
- A: baseline treatment (for static intervention) or treatment history $\bar{A}(t)$ (for longitudinal interventions);
- $\bar{L}(t)$: time-varying covariates up to time t;
- $\bar{C}(t)$: censoring process up to time t;
- T: observed time to event or censoring;
- $\Delta = I(T \leq \tilde{T})$: event indicator (1 if observed event, 0 if censored).

We observe the minimum of the event time and censoring time.

Parameter of Interest

The parameter of interest is often a marginal survival probability under a static treatment strategy:

$$\Psi(P_0) = \mathbb{E}_{P_0}[S^a(t)] = P(T^a > t),$$

where T^a is the potential failure time under intervention $A = a$, and $S^a(t)$ is the corresponding survival curve. TMLE targets this quantity while accounting for censoring and confounding.

Efficient Influence Curve

The efficient influence curve (EIC) for the survival probability under a static treatment a is:

$$D^*(O) = H(t, A, W) \left[\Delta I(T \leq t) - \hat{S}(t \mid A, W)\right] + \hat{S}(t \mid A, W) - \hat{S}^a(t),$$

where $H(t, A, W)$ is a clever covariate constructed using inverse probability weights for treatment and censoring.

Step-by-Step Algorithm (Pooled TMLE)

We now describe the procedure following Petersen et al. (2014) for static interventions:

1. **Estimate the initial outcome model.** Fit a pooled logistic regression model for the event indicator at each time point t, conditional on A, W, and t.

2. **Estimate the treatment and censoring mechanisms.** Estimate $g(A \mid W)$ and $P(C \geq t \mid A, W)$.
3. **Construct clever covariates.** Define:

$$H_i(t) = \frac{I(A_i = a) I(T_i \geq t)}{\hat{g}(A_i \mid W_i) \hat{P}(C_i \geq t \mid A_i, W_i)}$$

4. **Target the outcome model.** Update the initial model by performing logistic regression with offset equal to the logit of the initial prediction and covariate $H_i(t)$ to estimate a fluctuation parameter ϵ.
5. **Update survival estimates.** Use the updated predicted hazards to compute the targeted survival probability:

$$\hat{S}^a(t) = \prod_{s=1}^{t} \left(1 - \hat{P}(T = s \mid A = a, W)\right)$$

6. **Compute standard errors.** Use the empirical variance of the EIC across individuals to compute standard errors and confidence intervals.

Example: Manual Implementation of Pooled TMLE

We simulate a simplified dataset for illustration. Suppose we have baseline treatment A, covariate W, and time-to-event outcome T (censored at time 5).

Box 6.14: Pooled TMLE for survival: fitting the initial outcome model

7.2.15 R

```r
library(survival)
library(dplyr)

set.seed(123)
n <- 500
W <- rnorm(n)
A <- rbinom(n, 1, plogis(W))
hazard <- function(t, A, W) plogis(-2 + 0.5*A + 0.3*W + 0.1*t)
T_true <- sapply(1:n, function(i) {
  for (t in 1:5) {
    if (runif(1) < hazard(t, A[i], W[i])) return(t)
  }
  return(6)
})
C <- sample(2:6, n, replace = TRUE)
```

```r
T_obs <- pmin(T_true, C)
Delta <- as.integer(T_true <= C)

# Step 1: Create person-time dataset
data_long <- data.frame(
  id = rep(1:n, each = 5),
  t = rep(1:5, n),
  W = rep(W, each = 5),
  A = rep(A, each = 5),
  T = rep(T_obs, each = 5),
  Delta = rep(Delta, each = 5)
) %>%
  mutate(Y = as.integer(T == t & Delta == 1),
         at_risk = as.integer(T >= t))

# Step 2: Fit initial pooled logistic regression (TMLE step 1)
initial_fit <- glm(Y ~ A + W + t, family = binomial(),
                   data = data_long, subset = at_risk == 1)
summary(initial_fit)
```

7.2.16 Stata

```
* Simulate survival data for pooled TMLE
clear all
set seed 123
set obs 500
generate W = rnormal()
generate A = rbinomial(1, invlogit(W))
* Simulate event times (discrete, t=1..5)
* (Simplified: generate directly rather than from hazards for brevity)
generate T_true = ceil(6 * runiform())
generate C_time = ceil(4 * runiform()) + 2
generate T_obs = min(T_true, C_time)
generate Delta = (T_true <= C_time)

* Step 1: Expand to person-time format
stset T_obs, failure(Delta == 1) id(id)
stsplit t, at(failures)
generate at_risk = (_t <= T_obs)
generate Y = (_t == T_obs & Delta == 1)
```

```
* Step 2: Fit initial pooled logistic regression
logit Y A W t if at_risk == 1, vce(robust)
```

The subsequent steps include estimating censoring weights, constructing the clever covariate, performing the targeting update, and computing the survival curve. For brevity, we refer readers to code-based tutorials (e.g., TMLEbook) or the `ltmle` package for full pipelines.

Example: TMLE with the `ltmle` Package

We now implement TMLE for a longitudinal survival setting using the `ltmle` package. This automates estimation of survival curves under static interventions.

Box 6.15: TMLE for survival using the ltmle package

7.2.17 R

```
library(ltmle)

# Simulate survival data compatible with ltmle
set.seed(123)
n <- 300
W1 <- rnorm(n)
W2 <- rbinom(n, 1, 0.5)
A <- rbinom(n, 1, plogis(0.5 * W1 + 0.3 * W2))
Y1 <- rbinom(n, 1, plogis(-2 + 0.8 * A + 0.4 * W1))
C1 <- rbinom(n, 1, 0.1)  # 10% censoring at t=1
Y2 <- rbinom(n, 1, plogis(-2 + 0.8 * A + 0.4 * W1 + 0.1))
C2 <- rbinom(n, 1, 0.1)

ltmle_data <- data.frame(W1, W2, A,
                         C1 = C1, L1 = Y1,
                         C2 = C2, Y2 = Y2)

# Static intervention A = 1
ltmle_fit <- ltmle(data = ltmle_data,
                   Anodes = "A",
                   Cnodes = c("C1", "C2"),
                   Lnodes = "L1",
                   Ynodes = "Y2",
                   survivalOutcome = FALSE,
                   abar = 1)
```

```
summary(ltmle_fit)
```

7.2.18 Stata

```
* TMLE for survival data is available in R via the ltmle package.
* Stata users can adapt the pooled TMLE algorithm shown in Box 6.14
* using logit, predict, and manual clever covariate construction.
* The targeting step uses the same fluctuation approach (offset + clever covariate)
* described in the LTMLE section (Box 6.8).
```

This returns survival probability estimates under the static intervention, along with standard errors and confidence intervals based on the efficient influence function.

7.3 Conclusion

Causal inference with longitudinal data presents unique challenges: time-varying confounders affected by prior treatment cannot be handled by standard regression methods without introducing bias. This chapter has introduced three complementary approaches for this setting.

Marginal structural models (MSMs) use inverse probability of treatment weighting to estimate the marginal causal effect of a treatment sequence, breaking the feedback between confounders and treatment. The parametric g-formula for survival data estimates counterfactual survival curves by modeling the conditional hazard at each time point. Longitudinal TMLE (LTMLE) combines the double robustness and efficiency of TMLE with the flexibility to handle time-varying treatments and confounders, providing the strongest theoretical guarantees.

Key takeaways from this chapter: - Time-varying confounding affected by prior treatment (treatment-confounder feedback) invalidates standard regression adjustment. - MSMs with IPW create a pseudo-population where treatment is independent of confounders at each time point. - Stabilized weights reduce the variance of IPW estimators in longitudinal settings. - Doubly robust methods (AIPTW, LTMLE) protect against model misspecification by combining outcome and treatment models. - The parametric g-formula for survival estimates counterfactual survival curves under static or dynamic interventions. - LTMLE provides doubly-robust, semiparametric efficient estimates for longitudinal causal effects. - Inverse probability of censoring weighting (IPCW) is essential when follow-up is incomplete.

All three methods require careful attention to the positivity assumption, correct model specification, and the handling of censoring. When applied rigorously, they enable credible causal conclusions from complex longitudinal observational data.

7.4 Glossary

Censoring The loss of follow-up information before the outcome is observed; right censoring is the most common form in survival analysis.

Competing risk An event whose occurrence precludes the occurrence of the primary event of interest.

G-computation Another name for the g-formula; see glossary in Chapter 3.

IPCW Inverse Probability of Censoring Weighting — a method that reweights observations to account for informative censoring.

LTMLE Longitudinal Targeted Maximum Likelihood Estimation — an extension of TMLE for settings with time-varying treatments and confounders.

MSM Marginal Structural Model — a model for the marginal distribution of counterfactual outcomes under a treatment sequence, typically estimated using IPW.

AIPTW Augmented Inverse Probability of Treatment Weighting — a doubly robust estimator that combines IPTW with outcome regression predictions.

Stabilized weights IPW weights that include a numerator model (often without time-varying covariates) to reduce variance.

Time-varying confounding Confounding by a variable that changes over time and is affected by prior treatment.

8 Mediation Analysis

So far we have considered causal effects of a treatment on an outcome when a covariate is not on the causal pathway. Covariates that are on the causal pathway are termed "mediators". They are so named because a cause (e.g., the treatment) affects the mediator that, in turn, affects the outcome. In this chapter, we will (i) explore the role of mediators in causal analyses, (ii) describe statistical techniques for estimating causal effects in the presence of mediators, and (iii) understand how to interpret the various effect estimates.

8.1 Introduction to Mediation

8.1.1 Motivation and Overview

In causal inference, we often seek not only to estimate the total effect of an exposure or treatment on an outcome but also to understand the mechanisms through which this effect operates. *Mediation analysis* allows us to decompose the total effect into components that correspond to different causal pathways. This is particularly important in biomedical, social, and behavioural sciences, where identifying how and why a treatment works can inform intervention design, policy decisions, and scientific understanding.

For example, suppose a new drug improves patient outcomes. We may want to know whether the improvement occurs primarily by reducing inflammation, improving immune function, or some other biological process. Mediation analysis aims to quantify how much of the treatment's effect is explained by a specific intermediate variable—known as the *mediator*.

8.1.2 Total, Direct, and Indirect Effects

Let Adenote a binary treatment (e.g., 1 for treated, 0 for untreated),$M$the mediator (e.g., a biomarker), and$Y$ the outcome of interest (e.g., disease status). Under the potential outcomes framework, we define:

- Y^a: the potential outcome if treatment is set to a
- M^a: the potential mediator value under treatment a
- $Y^{a,m}$: the potential outcome if treatment is $a$and mediator is set to$m$

Then, the *total effect* of treatment on the outcome can be decomposed into:

$$\text{Total Effect (TE)} = \mathbb{E}[Y^1 - Y^0]$$

The total effect can be further partitioned into:

- **Natural Direct Effect (NDE)** : The effect of treatment on the outcome not through the mediator
$$\text{NDE} = \mathbb{E}[Y^{1,M^0} - Y^{0,M^0}]$$
- **Natural Indirect Effect (NIE)** : The effect of treatment that operates through the mediator
$$\text{NIE} = \mathbb{E}[Y^{0,M^1} - Y^{0,M^0}]$$
This decomposition assumes no interaction between the direct and indirect pathways. If such interaction exists, the decomposition still holds but interpretation becomes more nuanced.

8.1.3 Example: Treatment →Mediator→ Outcome

Consider a simulated example where a treatment Aaffects a continuous mediator$M$, which in turn affects a continuous outcome Y. The goal is to estimate the total, direct, and indirect effects.

8.1.3.1 Simulating Data in R

Box 7.1: Simulating data for mediation example

8.1.4 R

```r
set.seed(123)
n <- 1000
A <- rbinom(n, 1, 0.5)                    # Binary treatment
M <- 0.5 * A + rnorm(n)                   # Mediator depends on A
Y <- 0.3 * A + 0.6 * M + rnorm(n)         # Outcome depends on A and M

data <- data.frame(A, M, Y)
```

8.1.5 Stata

```
clear all
set seed 123
set obs 1000
gen A = runiform() < 0.5
gen M = 0.5*A + rnormal()
gen Y = 0.3*A + 0.6*M + rnormal()
```

8.1.5.1 Running Mediation Analysis in R

We can use the `mediation` package in R to estimate the average causal mediation effect (ACME, equivalent to NIE), average direct effect (ADE, equivalent to NDE), and total effect.

Box 7.2: Estimating ACME, ADE, and total effect with the mediation package

8.1.6 R

```
library(mediation)

# Fit mediator model
med.fit <- lm(M ~ A, data = data)

# Fit outcome model
out.fit <- lm(Y ~ A + M, data = data)

# Run mediation
med.out <- mediate(med.fit, out.fit, treat = "A", mediator = "M", boot = TRUE)
summary(med.out)
```

8.1.7 Stata

```
* Install mediation package if needed: ssc install mediation, replace
* Fit mediator model
regress M A

* Fit outcome model
regress Y A M

* Run mediation analysis
medeff (regress M A) (regress Y A M), treat(A) mediator(M) sims(1000)
```

This will return estimates of:

- `ACME (average causal mediation effect)` — the indirect effect
- `ADE (average direct effect)` — the direct effect
- `Total Effect = ACME + ADE`
- Proportion mediated: `ACME / Total Effect`

8.1.8 The Role of Causal Thinking in Mediation

Causal mediation analysis relies on several strong assumptions that must be justified with domain knowledge and visualized using causal diagrams (DAGs). The key identification assumptions include:

- No unmeasured confounding between treatment and outcome
- No unmeasured confounding between mediator and outcome
- No confounders of the mediator-outcome relationship affected by the treatment

These assumptions are not testable from data alone. Drawing a DAG helps clarify whether the necessary conditional independencies are plausible and guides appropriate adjustment strategies.

8.1.8.1 Example DAG

This simple DAG illustrates the decomposition of the total effect into a direct path ($A \to Y$) and an indirect path ($A \to M \to Y$). Any omitted arrows (e.g., from unmeasured confounders) must be considered when evaluating the validity of the assumptions.

8.1.9 Summary

Mediation analysis helps disentangle causal pathways by estimating direct and indirect effects. These analyses can be highly informative but require strong assumptions and careful modeling. In the next sections, we discuss identification strategies, estimation approaches, and practical tools for mediation analysis in computational causal inference.

8.2 Approaches to mediation analysis

8.2.1 Classic regression approach

Consider the causal diagram in Figure 7.1 with exposure X, mediator M, and outcome Y (for simplicity, assume there are no confounders). There are two effects of X on Y that can

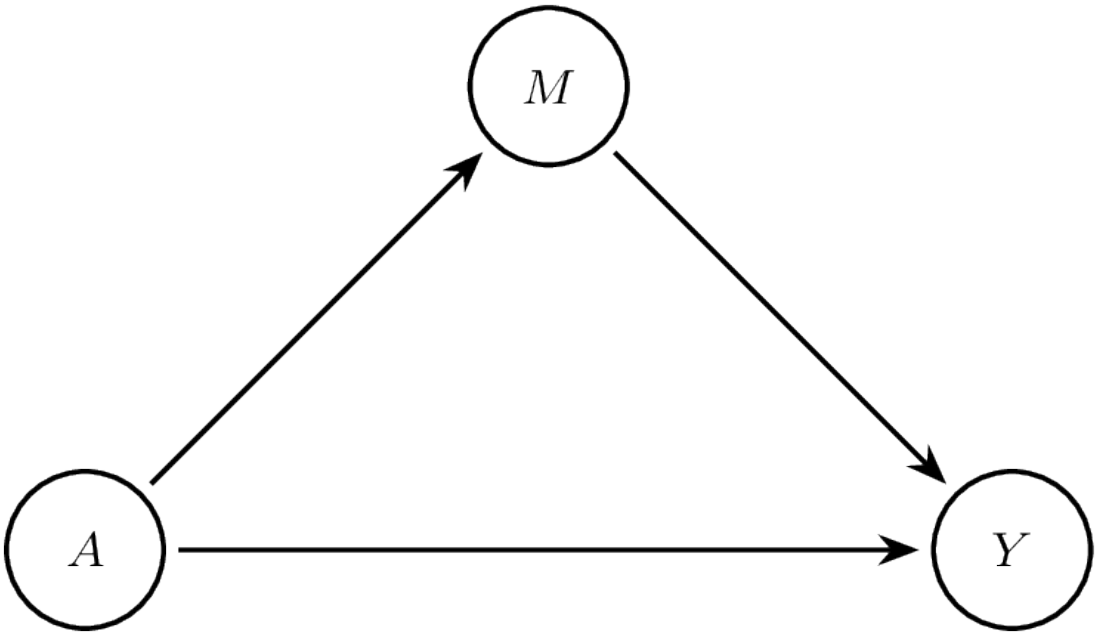

Figure 8.1: Simple mediation DAG showing direct and indirect paths

be measured in this scenario: the **direct effect** (DE) and the **indirect effect** (IE). The direct effect (i.e., $X \to Y$) is the effect of the exposure on the outcome that is not through the mediator. The indirect effect is the effect of the exposure on the outcome that operates through the mediator (i.e., $X \to M \to Y$).

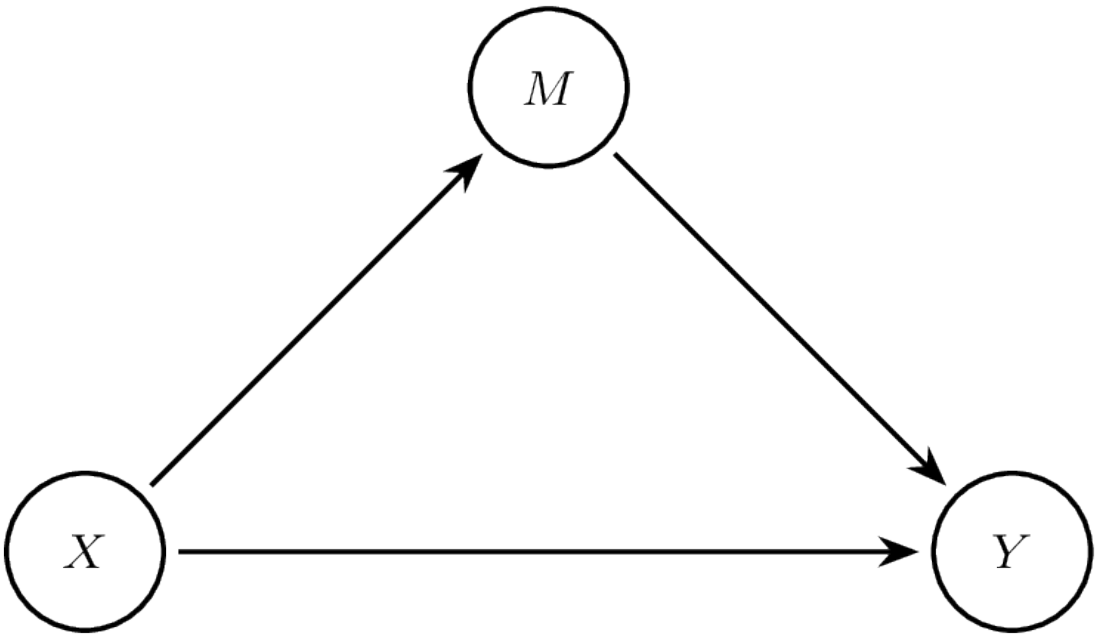

Figure 8.2: Mediation DAG showing total effect decomposition into direct and indirect paths

Early proponents of evaluating this model suggested using the "product of coefficients method" (also known as "product method") (Baron and Kenny, 1986). Let M and Y be continuous variables, consider the following regression models:

$$E(M \mid X = x) = \beta_0 + \beta_1 x \tag{8.1}$$

$$E(Y \mid X = x, M = m) = \theta_0 + \theta_1 x + \theta_2 m \tag{8.2}$$

Equation 8.1 is the regression model for the mediator. The coefficient β_1 is the expected increase in the value of the mediator for a unit increase in the exposure (for a binary exposure this coefficient would be the difference in means between two treatment groups).

Equation 8.2 is the regression model for the outcome. Baron and Kenny proposed that the coefficient $\theta_1$is the **direct effect** of$X \to Y$, which is the effect of the exposure on the outcome at a fixed level of the mediator variable (e.g., at the reference level). The coefficient θ_2 is the effect of the mediator on the outcome at a fixed level of the exposure variable.

Baron and Kenny also proposed that the **indirect effect** be calculated by estimating $\beta_1\theta_2$. The indirect effect is the effect on the outcome of changes of the exposure which operate through mediator levels.

To illustrate this, consider Figure 7.2 below. The arrows have now been labelled with their respective coefficients from Equation 8.1 and Equation 8.2

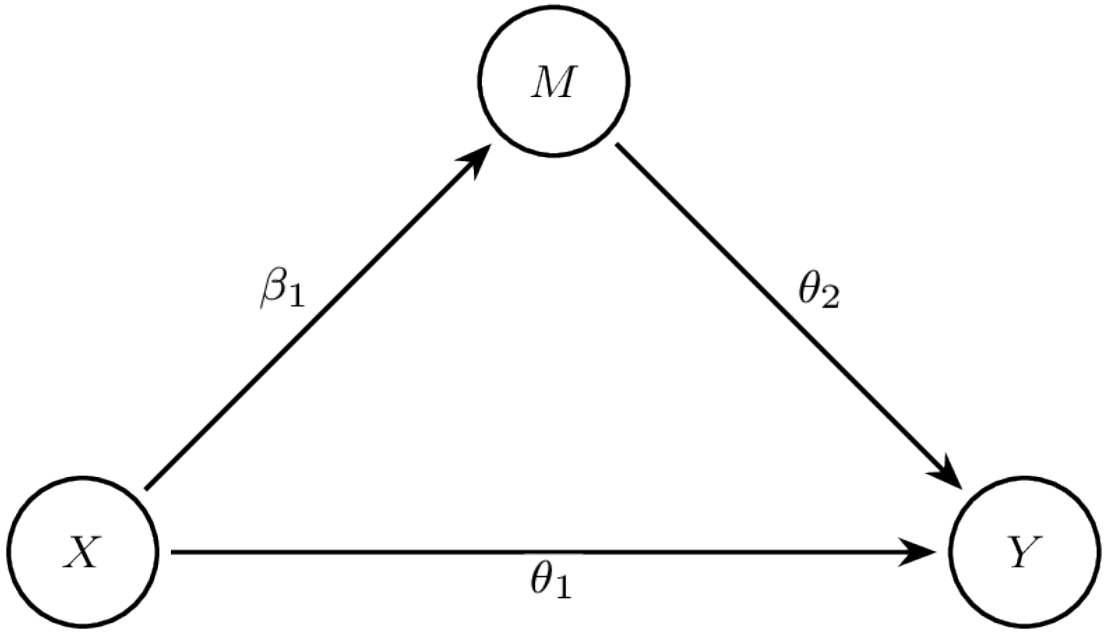

Figure 8.3: Mediation DAG with labelled path coefficients from the regression models

This classic regression approach can accommodate simplistic causal diagrams. However, this approach has its drawbacks. Firstly, in more complex scenarios, the mediator could be a collider between the exposure and an unmeasured variable (see lecture on causal diagrams). Therefore, adjusting for the mediator (e.g., in Equation 8.2) would induce an association between the exposure and the unmeasured variable (something we do not want to do!). Other complex scenarios could also incorporate interactions or non-linear terms for certain covariates. Secondly, the techniques used in this classic regression approach does not easily carry over into non-linear regression models, such as non-collapsibility. For example, consider the mediator was continuous and the outcome was binary. The product between the coefficients (i.e., $\beta_1\theta_2$) would be a combination of a mean difference (i.e., β_1) and a log-odds ratio (i.e., θ_2) - something that is very difficult to interpret! In the next section (Counterfactual approach) we will explore an alternative approach to estimating direct and indirect effects, which is more commonly used today and is the preferred approach.

8.2.1.1 Controlled direct effect

Before introducing the Counterfactual Approach, it is important to familiarise yourself with another estimand that is often of interest. Another commonly used measure of interest is the controlled direct effect (CDE). The controlled direct effect expresses how much the outcome would change on average if the mediator were fixed at level $m$uniformly in the population but the treatment were changed from level$x = 0$to level$x = 1$. Consequently, there are as many controlled direct effects as there are levels of the mediator.

The CDE corresponds to a situation in which a hypothetical intervention controls the mediator to a given value, whereas the direct effect corresponds to a situation in which the natural relationship between the exposure and the mediator is maintained (i.e., we would intervene on the exposure but not directly on the mediator).

The CDE and the direct effect is equivalent when there is no interaction between the exposure and the mediator (see Richiardi *et al* 2013, for further explanation). To illustrate this concept, consider again the model for the mediator (which is the same as Equation 8.1)

$$E(M \mid X = x) = \beta_0 + \beta_1 x$$

and now a model for the outcome that includes an interaction between the exposure and the mediator

$$E(Y \mid X = x, M = m) = \theta_0 + \theta_1 x + \theta_2 m + \theta_3 xm \tag{8.3}$$

Using Equation 8.1 and Equation 8.3, the CDE, DE, and IE can be estimated as follows (where $x$is exposed andx^* is not exposed):

$$\begin{aligned} \text{CDE}(m) &= (\theta_1 + \theta_3 m)\,(x - x^*) \\ DE &= (\theta_1 + \theta_3\beta_0 + \theta_3\beta_1 x^*)\,(x - x^*) \\ IE &= (\theta_2\beta_1 + \theta_3\beta_1 x)\,(x - x^*) \end{aligned}$$

Notice that, in Equation 8.3, if the interaction is absent, such that $\theta_3 = 0$, then the CDE and DE would be equivalent:

$$\begin{aligned} \text{CDE}(m) &= (\theta_1)\,(x - x^*) \\ DE &= (\theta_1)\,(x - x^*) \end{aligned}$$

To explain this concept further, if the direct effect of the exposure is constant for the different levels of the mediator, then setting the mediator to a fixed value (i.e., CDE) would give the same estimate. Similarly, setting the value that the mediator would have taken at the reference level of the exposure (i.e., DE) would also give the same estimate.

There is a difference in the interpretation of the CDE and DE even in the absence of the interaction. As an example, consider a hypothetical study on poor diet (exposure), obesity (mediator), and heart disease (outcome). The CDE (for obesity = 0) is the effect of eliminating poor diet when controlling obesity to be absent. For the DE, obesity would be set at the value that would have been observed in the absence of poor diet.

8.2.2 Counterfactual approach

To introduce the counterfactual approach notation in mediation analysis, consider again the causal diagram from Figure 7.1 where the exposure is binary but now the mediator is binary. Counterfactual notation defines two potential outcomes not only for the outcome of interest (i.e., Y) but also the mediator (i.e., M). The potential outcomes for the mediator are:

- M^0: the value of the mediator had the individual received exposure level $x = 0$.
- M^1: the value of the mediator had the individual received exposure level $x = 1$.

Since the mediator is hypothetical (i.e., consists of potential outcomes), the potential outcome notation for the outcome of interest must also accommodate the potential outcomes of the mediator. In mediation analysis, $Y^{x,m}$is the potential outcome under exposure level$X = x$and mediator level$M = m$. The potential outcomes are:

1. Y^{0,M^0}: the value of the outcome had the individual received exposure level $x = 0$and the mediator taken the value it would have done under exposure level$x = 0$.
2. Y^{1,M^1}: the value of the outcome had the individual received exposure level $x = 1$and the mediator taken the value it would have done under exposure level$x = 1$.
3. Y^{0,M^1}: the value of the outcome had the individual received exposure level $x = 0$and the mediator taken the value it would have done under exposure level$x = 1$.
4. Y^{1,M^0}: the value of the outcome had the individual received exposure level $x = 1$and the mediator taken the value it would have done under exposure level$x = 0$.

Using this potential outcome notation we can define the natural effects (causal estimands) of interest. The natural effects are the natural direct effect (NDE) and the natural indirect effect (NIE), which together sum up to the total effect (TE). If we were to control the mediator at the level seen in the non-exposed group (i.e., $x = 0$and$M^{x=0}$), then:

Natural direct effect

$$E(Y^{1,M^0} - Y^{0,M^0})$$

$$\frac{E(Y^{1,M^0})}{E(Y^{0,M^0})}$$

Natural indirect effect

$$E(Y^{1,M^1} - Y^{1,M^0})$$

$$\frac{E(Y^{1,M^1})}{E(Y^{1,M^0})}$$

Controlled direct effect

$$E(Y^{1,m} - Y^{0,m})$$

$$\frac{E(Y^{1,m})}{E(Y^{0,m})}$$

Alternatively, if one were interested in using the exposed group as the reference group, then the natural direct effect would be M^1(instead ofM^0), and the natural indirect effect would be $x = 0$(instead of$x = 1$).

The **natural direct effect** (NDE) expresses how much the outcome would change, on average, if the exposure were set at level $x = 1$versus level$x = 0$ but for each individual the mediator were kept at the level it would have taken, for that individual, in the absence of the exposure. The NDE captures what the effect of the exposure on the outcome would remain if we were to disable the pathway from the exposure to the mediator.

The **natural indirect effect** (NIE) expresses how much the outcome would change, on average, if the exposure were set at level $x = 1$but the mediator were changed from the level it would take if$x = 0$to the level it would take if$x = 1$. The NIE captures the effect of the exposure on the outcome that operates by changing the mediator.

Notice that the potential outcome notation for Y^{0,M^1}relies on us knowing what the outcome would have been for an individual in exposure group$x = 0$**but** they had the value of the mediator as if they were in the other exposure group (i.e.,$x = 1$). It is not possible to observe this from the data alone. In the same way, it is not possible for us to observe Y^{1,M^0}. We will explore methods of estimating the NDE and NIE in the following sections, but first we must make certain assumptions.

8.2.3 Assumptions

To illustrate the assumptions for mediation analysis, first consider the causal diagram in Figure 7.3 The DAG consists of the exposure (X), mediator (M), outcome (Y), exposure-outcome confounder (C), and mediator-outcome confounder (Z). For simplicity, we do not include exposure-mediator confounder but this variable is likely to occur in a wide range of scenarios and should be carefully considered when doing such an analysis.

With such a DAG, and to estimate effects of interest, we need to make certain assumptions. The assumptions relate not only to the relationship between the exposure and the outcome but also the relationships with the mediator:

1. No unmeasured exposure-outcome confounders given C
2. No unmeasured mediator-outcome confounders given C and A
3. No unmeasured exposure-mediator confounders given C
4. No unmeasured mediator-outcome confounders affected by the exposure (i.e., no arrow from X to Z)

To estimate the controlled direct effect (CDE), we must assume (1) no unmeasured exposure-outcome confounding. When the treatment is randomised, assumption (1) is automatically satisfied. We also assume (2) no unmeasured mediator-outcome confounding. To estimate the CDE from Figure 7.3, we must control for $C$and$Z$.

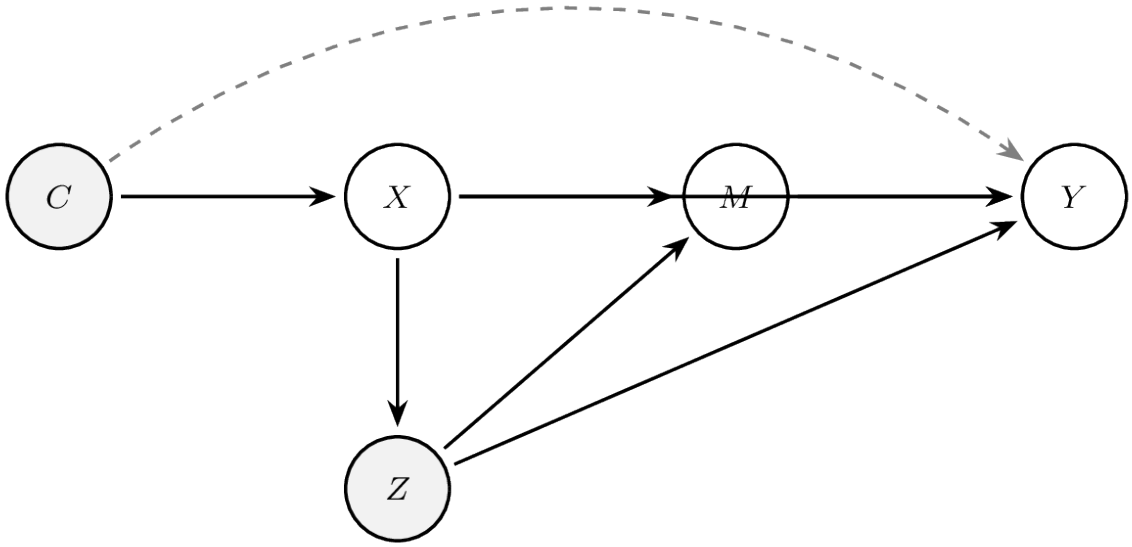

Figure 8.4: Mediation DAG with exposure-outcome confounder C and mediator-outcome confounder Z

For identification of the natural direct and indirect effects, two further assumptions are required. There must also be (3) no unmeasured exposure-mediator confounding, which is automatically satisfied if the exposure is randomised. Lastly, an often strong assumption is that there must be (4) no unmeasured mediator-outcome confounder that is affected by the exposure, this assumption is often called the "cross-world independence assumption".

Note that randomisation of the exposure is not sufficient to control for confounding in mediation analysis. Randomisation allows controlling for the exposure-outcome and exposure-mediator relationships but it does not ensure no unmeasured mediator-outcome confounding because the mediator is often not randomised.

8.2.4 Controlled Direct Effects vs Natural Direct Effects

While the **natural direct effect** (NDE) conditions on the natural value of the mediator under no treatment (M^0), the **controlled direct effect** (CDE) fixes the mediator to a specific value for all individuals:

$$\text{CDE}(m) = \mathbb{E}[Y^{1,m} - Y^{0,m}]$$

This represents the direct effect of treatment when the mediator is held constant at a specified level m. Controlled direct effects are easier to identify since they do not rely on cross-world counterfactuals (like Y^{1,M^0}).

- **CDEs** can be interpreted as the effect of treatment if we were able to intervene and fix the mediator.
- **NDEs** describe the effect when the mediator is allowed to take its natural value under no treatment—more interpretable, but harder to identify.

8.2.4.1 Implication for Practice

Controlled direct effects are estimable under weaker assumptions but may lack realism unless intervention on the mediator is plausible. Natural direct and indirect effects offer more intuitive interpretations of mediation but require stronger assumptions and careful modeling.

8.3 Estimation of Effects

Several estimation strategies have been developed to quantify mediation effects, ranging from classical regression-based methods to modern counterfactual-based estimators. In this section, we review three major approaches: parametric g-computation, regression-based mediation analysis, and counterfactual-based methods.

8.3.1 Parametric g-computation

The parametric g-computation formula provides a way to estimate causal effects by modeling the outcome and mediator using parametric regression models and then integrating over the empirical distribution of covariates. It is particularly useful when both treatment and mediator are continuous or binary.

Let Abe the treatment,$M$the mediator, and$Y$the outcome. Suppose we have baseline covariates$X$. Under the identification assumptions discussed previously, the natural indirect effect (NIE) can be expressed as:

$$\begin{aligned}\text{NIE} = &\int \left[\int \mathbb{E}[Y \mid A = 0, M = m, X = x] \, dF_{M|A=1,X}(m) \right] dF_X(x) \\ &- \int \left[\int \mathbb{E}[Y \mid A = 0, M = m, X = x] \, dF_{M|A=0,X}(m) \right] dF_X(x)\end{aligned}$$

This expression can be approximated via Monte Carlo simulation in practice.

8.3.1.1 R Example: Parametric g-computation

Assume a data-generating process where both the mediator and outcome are continuous:

Box 7.3: Simulating data for parametric g-computation

8.3.2 R

```r
set.seed(123)
n <- 1000
X <- rnorm(n)
A <- rbinom(n, 1, 0.5)
M <- 0.5*A + 0.3*X + rnorm(n)
Y <- 0.6*M + 0.3*A + 0.2*X + rnorm(n)
data <- data.frame(A, M, Y, X)
```

8.3.3 Stata

```
clear all
set seed 123
set obs 1000
gen X = rnormal()
gen A = runiform() < 0.5
gen M = 0.5*A + 0.3*X + rnormal()
gen Y = 0.6*M + 0.3*A + 0.2*X + rnormal()
```

We fit the mediator and outcome models, then simulate counterfactuals:

Box 7.4: Parametric g-computation estimation of NIE, NDE, and TE

8.3.4 R

```r
# Fit mediator and outcome models
med_model <- lm(M ~ A + X, data = data)
out_model <- lm(Y ~ A + M + X, data = data)

# Predict mediator under A = 1 and A = 0
data$M1 <- predict(med_model, newdata = transform(data, A = 1))
data$M0 <- predict(med_model, newdata = transform(data, A = 0))

# Predict outcome under various scenarios
Y_0_M1 <- predict(out_model, newdata = transform(data, A = 0, M = data$M1))
Y_0_M0 <- predict(out_model, newdata = transform(data, A = 0, M = data$M0))
Y_1_M0 <- predict(out_model, newdata = transform(data, A = 1, M = data$M0))
Y_0_M0_total <- predict(out_model, newdata = transform(data, A = 0, M = data$M0))
```

```
# Compute effects
NIE <- mean(Y_0_M1 - Y_0_M0)
NDE <- mean(Y_1_M0 - Y_0_M0)
TE  <- mean(Y_0_M1 - Y_0_M0_total + Y_1_M0 - Y_0_M0)

cat("NIE:", NIE, "\nNDE:", NDE, "\nTE:", TE, "\n")
```

8.3.5 Stata

```
* Fit mediator and outcome models
regress M A X
scalar b0_m = _b[_cons]
scalar bA_m = _b[A]
scalar bX_m = _b[X]

regress Y A M X
scalar b0_y = _b[_cons]
scalar bA_y = _b[A]
scalar bM_y = _b[M]
scalar bX_y = _b[X]

* Counterfactual mediator predictions
gen M_A1 = b0_m + bA_m*1 + bX_m*X
gen M_A0 = b0_m + bA_m*0 + bX_m*X

* Counterfactual outcome predictions
gen Y_0_M1 = b0_y + bA_y*0 + bM_y*M_A1 + bX_y*X
gen Y_0_M0 = b0_y + bA_y*0 + bM_y*M_A0 + bX_y*X
gen Y_1_M0 = b0_y + bA_y*1 + bM_y*M_A0 + bX_y*X

* Compute effects as averages
summarize Y_0_M1, meanonly
local m_Y_0_M1 = r(mean)
summarize Y_0_M0, meanonly
local m_Y_0_M0 = r(mean)
summarize Y_1_M0, meanonly
local m_Y_1_M0 = r(mean)

local NIE = `m_Y_0_M1' - `m_Y_0_M0'
local NDE = `m_Y_1_M0' - `m_Y_0_M0'
local TE  = `NIE' + `NDE'
```

```
display "NIE: " `NIE'
display "NDE: " `NDE'
display "TE:  " `TE'
```

This approach provides a flexible, transparent way to estimate causal mediation effects under parametric assumptions.

8.3.6 Regression-Based Mediation

The classical regression-based approach to mediation, introduced by Baron and Kenny (1986), uses a sequence of linear regressions to assess whether a mediator carries the effect of a treatment to the outcome. This method is simple and interpretable but does not have a formal counterfactual interpretation.

8.3.6.1 Baron & Kenny Steps

Given a treatment A, mediator M, and outcome Y, the following regressions are fitted:

1. Regress $M \text{on} A$: $M = \alpha_0 + \alpha_1 A + \varepsilon_M$
2. Regress $Y \text{on} A$: $Y = \tau_0 + \tau A + \varepsilon_Y$
3. Regress $Y \text{on} A \text{and} M$: $Y = \beta_0 + \beta_1 A + \beta_2 M + \varepsilon_Y$

If: - α_1 is significant (A affects M), - β_2 is significant (M affects Y controlling for A), - and $|\beta_1| < |\tau|$ (effect of A on Y is reduced when M is added),

then there is evidence of mediation.

8.3.6.2 R Example

Box 7.5: Baron and Kenny steps for mediation analysis

8.3.7 R

```
# Step 1
summary(lm(M ~ A, data = data))  # Effect of A on M

# Step 2
summary(lm(Y ~ A, data = data))  # Total effect
```

```r
# Step 3
summary(lm(Y ~ A + M, data = data))  # Mediation model
```

8.3.8 Stata

```
* Step 1: Effect of A on M
regress M A
scalar alpha1 = _b[A]

* Step 2: Total effect of A on Y
regress Y A
scalar total_eff = _b[A]

* Step 3: Mediation model (Y ~ A + M)
regress Y A M
scalar beta2 = _b[M]
scalar direct_eff = _b[A]

* Indirect effect = alpha1 * beta2
scalar indirect_eff = alpha1 * beta2
display "Indirect effect (alpha1 * beta2): " indirect_eff
display "Direct effect: " direct_eff
display "Total effect: " total_eff
```

The indirect effect can be approximated as $\alpha_1 \cdot \beta_2$, and the direct effect as β_1.

Limitations:

The classical regression-based approach to mediation, as originally proposed by Baron and Kenny, has several important limitations. First, it lacks a formal counterfactual basis, meaning it does not define or estimate causal effects in terms of potential outcomes. This restricts the interpretability of the estimated effects as truly causal. Second, the method relies on strong assumptions of linearity and additivity, and it does not easily accommodate interactions between the treatment and the mediator. Finally, it provides no built-in framework for statistical inference or sensitivity analysis, making it difficult to assess uncertainty around estimates or to evaluate the robustness of conclusions to violations of assumptions.

8.3.9 Counterfactual-Based Methods

Modern mediation methods are built on the potential outcomes (counterfactual) framework, allowing for clear definitions of direct and indirect effects and accommodating non-linear models,

interactions, and bootstrapped confidence intervals.

8.3.9.1 Estimation via Mediation Package in R

The `mediation` package (Tingley et al., 2014) estimates average causal mediation effects (ACME) and average direct effects (ADE) under assumptions described earlier.

Box 7.6: Counterfactual-based mediation with covariates

8.3.10 R

```
library(mediation)

# Fit models
med_model <- lm(M ~ A + X, data = data)
out_model <- lm(Y ~ A + M + X, data = data)

# Estimate mediation effects
med.out <- mediate(med_model, out_model,
                   treat = "A", mediator = "M",
                   boot = TRUE, sims = 1000)

summary(med.out)
```

8.3.11 Stata

```
* Install mediation package if needed: ssc install mediation, replace
* Fit mediator and outcome models
regress M A X
estimates store med_model
regress Y A M X
estimates store out_model

* Estimate mediation effects using medeff
medeff (regress M A X) (regress Y A M X), ///
    treat(A) mediator(M) sims(1000)
```

The output includes: - ACME (Average Causal Mediation Effect): the indirect effect - ADE (Average Direct Effect): the direct effect - Total Effect: ACME + ADE - Proportion Mediated: ACME / Total

Advantages:

Counterfactual-based methods offer several advantages over traditional approaches. First, they provide a formal causal interpretation rooted in the potential outcomes framework, allowing clear definitions of direct and indirect effects. Second, these methods are flexible and can be applied to a wide range of models, including nonlinear models and those with treatment-mediator interactions. Third, they support inference using bootstrap-based confidence intervals, which are particularly useful when the sampling distribution of mediation effects is complex or unknown.

Limitations:

Despite their strengths, counterfactual-based methods also come with limitations. They rely on strong identification assumptions—such as no unmeasured confounding of the mediator-outcome relationship and the absence of exposure-induced mediator-outcome confounding—which are not testable from the data and may not hold in observational studies. Additionally, the underlying concepts can be more challenging to communicate to non-technical audiences, particularly those unfamiliar with potential outcomes or causal diagrams.

8.3.12 Summary

Each of the three approaches to mediation analysis—g-computation, regression-based analysis, and counterfactual-based estimation—offers distinct advantages and limitations. G-computation provides flexible parametric integration. Classical regression is simple but limited in scope. Counterfactual-based methods provide a rigorous framework under clear assumptions and are widely used in modern causal inference practice.

In the next section, we explore advanced extensions to mediation analysis, including interventional effects and mediation under intermediate confounding.

8.4 Advanced Methods

In this section, we discuss several advanced approaches that extend traditional mediation analysis. These include interventional (or stochastic) effects that circumvent some of the identification challenges of natural effects, methods that account for intermediate confounding, and a brief overview of mediation in longitudinal settings.

8.4.1 Interventional Effects

Traditional mediation analysis relies on cross-world counterfactuals such as Y^{1,M^0}, which are challenging to identify and require strong assumptions. Interventional effects offer an alternative that avoids cross-world contrasts by defining effects based on stochastic interventions on the mediator.

Definition: The *interventional indirect effect* is defined as the change in the outcome distribution due to intervening on the mediator, such that its distribution matches what it would have been under treatment $A = 1$, but keeping the treatment fixed at $A = 0$:

$$\text{IIE} = \mathbb{E}\left[Y^{0,\tilde{M}^1}\right] - \mathbb{E}\left[Y^{0,\tilde{M}^0}\right]$$

Similarly, the *interventional direct effect* is:

$$\text{IDE} = \mathbb{E}\left[Y^{1,\tilde{M}^1}\right] - \mathbb{E}\left[Y^{0,\tilde{M}^1}\right]$$

where $\tilde{M}^a$is a random draw from the distribution of$M$under treatment$A = a$. These estimands can be identified under weaker assumptions than natural effects and are still interpretable as causal pathways.

Estimation in R: The `medflex` package provides tools for estimating interventional effects.

Box 7.7: Interventional effects estimation with medflex

8.4.2 R

```r
library(medflex)

# Fit the working models
expData <- neImpute(Y ~ A + M + X, data = data)
neMod <- neModel(Y ~ A0 + M + X, family = gaussian, expData = expData)

# Estimate interventional effects
summary(neMod)
```

8.4.3 Stata

```
* Stata does not have a direct equivalent of the medflex package.
* Interventional effects can be approximated using medeff, which
* estimates natural direct and indirect effects under similar assumptions.
medeff (regress M A X) (regress Y A M X), ///
```

```
    treat(A) mediator(M) sims(1000)

* Alternatively, use paramed (ssc install paramed) for more flexible
* parametric mediation with interaction terms:
* paramed Y, treat(A) mediator(M) covariates(X) boot(1000)
```

This returns estimates of the interventional direct and indirect effects, along with standard errors and confidence intervals.

Advantages:

Interventional effects offer several important advantages over natural direct and indirect effects. First, they do not rely on cross-world counterfactuals—such as Y^{1,M^0} — which are inherently unobservable and require strong assumptions for identification. Second, interventional effects can be identified under weaker conditions, making them more robust to violations of assumptions that often limit traditional mediation analysis. Third, these effects are readily adaptable to a wide range of model types, including nonlinear and nonparametric models, which enhances their flexibility in practical applications.

8.4.4 Mediation with Intermediate Confounding

A key assumption of natural effect identification is the absence of exposure-induced confounding of the mediator-outcome relationship. This means there are no variables that: 1. Affect both the mediator and the outcome, 2. Are themselves affected by the treatment.

Such variables are called *intermediate confounders*. When they exist, traditional approaches may produce biased estimates.

Solution: To handle intermediate confounding, methods like *sequential g-estimation*, *inverse probability weighting*, or *targeted maximum likelihood estimation (TMLE)* can be used. These techniques adjust for the time-varying confounders without blocking the indirect path.

Example: Using IPTW for mediation

Suppose L is an intermediate confounder (e.g., post-treatment health status). We estimate weights for the mediator model that account for treatment and confounders:

Box 7.8: IPTW for mediation with intermediate confounding

8.4.5 R

```r
# Estimate propensity for M conditional on A and L
med.weight.model <- glm(M ~ A + L + X, family = binomial(), data = data)
data$med.weights <- 1 / predict(med.weight.model, type = "response")

# Use these weights in a weighted regression of Y on A and M
library(survey)
design <- svydesign(ids = ~1, weights = ~med.weights, data = data)
svyglm(Y ~ A + M + X, design = design)
```

8.4.6 Stata

```
* Estimate propensity for M conditional on A and L
logit M A L X
predict med_prob, pr
gen med_weights = 1 / med_prob

* Use weights in a weighted regression of Y on A and M
svyset [pw = med_weights]
svy: regress Y A M X
```

This approach helps isolate the indirect effect while adjusting for post-treatment confounding.

Limitations:

Methods for mediation analysis in the presence of intermediate or time-varying confounding come with notable limitations. They require careful and accurate modeling of the confounding structure, particularly when confounders are influenced by prior treatment or mediator values. Additionally, these methods are sensitive to model misspecification and violations of the positivity assumption—that is, the assumption that all levels of treatment and mediator occur with non-zero probability across covariate strata. Violations of these conditions can lead to unstable or biased estimates.

8.4.7 Longitudinal Mediation

In longitudinal studies, treatment, mediator, and outcome variables may be measured repeatedly over time, reflecting the dynamic nature of causal processes. For example, a health intervention administered over several months (A_t) may influence weight (M_t) and, in turn, affect blood pressure (Y_t) at multiple follow-up visits. In such cases, causal effects may accumulate or change over time, and past values of mediators or outcomes may influence future treatments, making analysis more complex than in cross-sectional settings.

Challenges

Longitudinal mediation presents several methodological difficulties. Chief among them is *time-varying confounding*, where intermediate variables (e.g., stress, diet) are affected by prior treatment or mediators and simultaneously affect future mediators and outcomes. This introduces bias if not properly accounted for. Another challenge is *feedback loops*—situations in which past values of mediators influence future treatments or vice versa. Lastly, the presence of *multiple mediators* at different time points, and *lagged or delayed effects*, increases model complexity and demands more flexible estimation frameworks.

Approaches

Several estimation strategies address these complexities:

- **Structural Nested Models (SNMs)** : Allow for explicit modeling of treatment effects over time, while adjusting for intermediate confounding.
- **Longitudinal G-computation**: Uses recursive substitution to model outcomes forward in time, integrating over the empirical distribution of time-varying confounders and mediators.
- **Longitudinal TMLE**: Extends targeted maximum likelihood estimation to longitudinal settings, combining machine learning with iterative targeting to estimate marginal effects with double robustness and efficiency.

R Packages

- `ltmle` provides a general framework for longitudinal targeted learning.
- `medltmle` implements longitudinal mediation-specific TMLE estimators.

8.4.7.1 Worked Example: Estimating Longitudinal Mediation Effects with `medltmle`

Consider a simulated study in which physical activity (A_t) influences BMI (M_t), which in turn affects systolic blood pressure (Y_t) at three time points (baseline, 1-year, 2-year). Stress (L_t) acts as a time-varying confounder.

The data structure is in wide format, with columns such as: `A1, A2`, `M1, M2`, `Y1, Y2`, `L1, L2`, and baseline covariates `W1, W2`.

Box 7.9: Longitudinal mediation analysis with medltmle

8.4.8 R

```r
library(medltmle)

# Simulated dataset with 2 follow-up time points
data(simLongMediation)

# Define variables
Anodes <- c("A1", "A2")
Cnodes <- NULL  # No censoring in this example
Lnodes <- c("L1", "L2")  # Time-varying confounders
Mnodes <- c("M1", "M2")  # Mediators
Ynodes <- c("Y2")         # Final outcome only
Wnodes <- c("W1", "W2")   # Baseline covariates

# Estimate natural direct and indirect effects using TMLE
result <- medltmle(data = simLongMediation,
                   Anodes = Anodes,
                   Cnodes = Cnodes,
                   Lnodes = Lnodes,
                   Mnodes = Mnodes,
                   Ynodes = Ynodes,
                   Wnodes = Wnodes,
                   abar0 = c(0, 0),  # Control treatment regime
                   abar1 = c(1, 1),  # Treated regime
                   gform = NULL,     # Use Super Learner (default)
                   Yrange = c(80, 180),
                   deterministic.g.function = NULL)

summary(result)
```

8.4.9 Stata

```
* Stata does not have a direct equivalent of medltmle.
* A simplified regression-based approach for 2 time points:

* Fit mediator models at each time point
regress M1 A1 L1 W1 W2
regress M2 A2 L2 A1 L1 M1 W1 W2

* Fit outcome model at final time point
regress Y2 A2 M2 A1 M1 L2 L1 W1 W2, robust
```

```
* Bootstrap inference for mediation can be implemented as:
* bootstrap, reps(500): regress Y2 A2 M2 A1 M1 L2 L1 W1 W2
```

The `summary()` output provides estimates of the longitudinal total effect (TE), natural direct effect (NDE), and natural indirect effect (NIE), each accounting for the cumulative dynamics of treatment, mediators, and confounders over time.

Using `medltmle`, we can assess whether the long-term impact of the intervention on blood pressure is driven primarily by changes in BMI (indirect effect) or through other direct mechanisms. This approach allows for appropriate adjustment for time-varying confounding and flexible functional forms using machine learning.

Longitudinal mediation analysis captures complex causal mechanisms unfolding over time. Tools such as `medltmle` make it feasible to estimate these effects under realistic assumptions, even when confounding and mediation evolve dynamically. Researchers are encouraged to use DAGs, model checking, and sensitivity analysis to validate findings in these more complex scenarios.

8.4.10 Summary

Advanced mediation methods allow for more flexible and realistic modeling of causal mechanisms. Interventional effects provide an interpretable alternative to natural effects under weaker assumptions. Approaches that address intermediate confounding are crucial when post-treatment confounders exist. Longitudinal mediation methods account for repeated, time-dependent relationships, though at the cost of increased complexity. These tools are essential for applied researchers aiming to uncover nuanced pathways of causal influence.

8.5 Sensitivity Analysis in Mediation

Causal mediation analysis relies on several strong identification assumptions, one of the most critical being the absence of unmeasured confounding between the mediator and the outcome. This assumption is often difficult to justify in observational studies, where the same covariates that confound the exposure-outcome relationship may not fully account for confounding of the mediator-outcome relationship. In this section, we review methods for assessing the robustness of mediation findings to potential violations of this assumption, focusing on bias formulas developed by VanderWeele and colleagues.

8.5.1 Mediator-Outcome Confounding

Recall that to identify the natural indirect effect (NIE), we require:

$$Y^{a,m} \perp\!\!\!\perp M \mid A = a, X$$

This means that, conditional on treatment and baseline covariates X, the mediator must be as good as randomized with respect to the potential outcomes. In practice, this is often implausible due to omitted variables (e.g., psychological factors, unmeasured behaviors, or genetic traits) that may influence both the mediator and the outcome.

When this assumption fails, estimated mediation effects—particularly the NIE—can be biased. Sensitivity analysis provides a way to quantify how strong such unmeasured confounding would need to be to substantially alter the conclusions of the analysis.

8.5.2 VanderWeele-Style Bias Analysis

VanderWeele (2010, 2015) developed analytic bias formulas that quantify how unmeasured mediator-outcome confounding might distort estimates of mediation effects. These formulas can be used to:

- Perform a sensitivity analysis by varying hypothetical values of confounding parameters
- Identify the conditions under which the NIE or NDE would be reduced to zero

8.5.2.1 Bias Formula for Continuous Outcomes

Suppose we estimate the NIE using linear models. Then the bias in the estimated NIE due to an unmeasured confounder U is approximately:

$$\text{Bias}_{\text{NIE}} \approx \rho_{MY \cdot A,X} \cdot \rho_{MU \cdot A,X} \cdot \sigma_Y \cdot \sigma_M$$

where: - $\rho_{MY \cdot A,X}$: Partial correlation between $M$and$Y$, given $A$and$X$ - $\rho_{MU \cdot A,X}$: Partial correlation between an unmeasured confounder $U$and both$M$and$Y$ - σ_Y, σ_M: Standard deviations of $Y$and$M$

By varying $\rho_{MU \cdot A,X}$ over a plausible range (e.g., -0.3 to 0.3), we can assess the impact of unmeasured confounding on the NIE.

8.5.2.2 R Implementation Using `medsens`

The `mediation` package includes a function `medsens()` to perform this kind of sensitivity analysis following a call to `mediate()`.

Box 7.10: Sensitivity analysis with medsens

8.5.3 R

```r
library(mediation)

# Step 1: Fit the mediator and outcome models
med.model <- lm(M ~ A + X, data = data)
out.model <- lm(Y ~ A + M + X, data = data)

# Step 2: Estimate the mediation effects
med.out <- mediate(med.model, out.model, treat = "A", mediator = "M", boot = TRUE)

# Step 3: Run sensitivity analysis
sens.out <- medsens(med.out, rho.by = 0.01)

# Step 4: Plot sensitivity analysis
plot(sens.out, sens.par = "rho")
```

8.5.4 Stata

```
* Install mediation package if needed: ssc install mediation, replace
* Step 1: Fit mediator and outcome models
regress M A X
estimates store med_model
regress Y A M X
estimates store out_model

* Step 2: Estimate mediation effects (required before medsens)
medeff (regress M A X) (regress Y A M X), treat(A) mediator(M) sims(1000)

* Step 3: Run sensitivity analysis
medsens (regress M A X) (regress Y A M X), ///
    treat(A) mediator(M) rho(-0.5(0.01)0.5)

* Note: Stata's medsens does not produce a built-in plot;
* results can be exported and plotted separately
```

This generates a sensitivity plot showing how the estimated ACME (NIE) changes as a function of the sensitivity parameter ρ, which captures the strength of correlation between the error terms in the mediator and outcome models (i.e., residual confounding).

8.5.4.1 Interpretation

If the ACME estimate remains far from zero even when $\rho$is large (e.g.,$\rho = 0.3$), the mediation effect is considered robust to moderate levels of unmeasured confounding. Conversely, if a small ρ is enough to explain away the effect, the result is considered sensitive.

8.5.4.2 Binary Outcomes

When Y is binary, similar bias formulas exist, though they are more complex and typically require modeling on the log-odds scale. The `mediation` package also supports sensitivity analysis for binary outcomes, assuming appropriate logistic models are used for both mediator and outcome regressions.

8.5.5 Simulated Example: When Unmeasured Confounding Overturns the ACME

To illustrate the potential impact of unmeasured mediator-outcome confounding, we simulate a simple data-generating process in which an unobserved variable confounds the relationship between the mediator and the outcome. We then compare the naive (biased) estimate of the average causal mediation effect (ACME) to the true value obtained when the confounder is included in the model.

8.5.5.1 Data Generating Process

We simulate a binary treatment A, a continuous mediator M, and a continuous outcome Y, along with a baseline covariate $X$and an unmeasured confounder$U$. The mediator and outcome are generated as follows:

$$M = 0.5A + 0.5U + 0.3X + \varepsilon_M, \quad \varepsilon_M \sim \mathcal{N}(0, 1)$$

$$Y = 0.3A + 0.6M + 0.6U + 0.2X + \varepsilon_Y, \quad \varepsilon_Y \sim \mathcal{N}(0, 1)$$

Here, $U$is unobserved and influences both$M$and$Y$, violating the assumption of no unmeasured mediator-outcome confounding.

8.5.5.2 Simulation Code

Box 7.11: Simulating data with unmeasured mediator-outcome confounding

8.5.6 R

```r
set.seed(42)
n <- 1000
X <- rnorm(n)
A <- rbinom(n, 1, 0.5)
U <- rnorm(n)
M <- 0.5*A + 0.5*U + 0.3*X + rnorm(n)
Y <- 0.3*A + 0.6*M + 0.6*U + 0.2*X + rnorm(n)
data <- data.frame(A, M, Y, X, U)
```

8.5.7 Stata

```stata
clear all
set seed 42
set obs 1000
gen X = rnormal()
gen A = runiform() < 0.5
gen U = rnormal()
gen M = 0.5*A + 0.5*U + 0.3*X + rnormal()
gen Y = 0.3*A + 0.6*M + 0.6*U + 0.2*X + rnormal()
```

8.5.7.1 Naive Estimation (Ignoring Unmeasured Confounding)

We fit the standard mediation models, omitting U:

Box 7.12: Naive ACME estimation ignoring unmeasured confounder

8.5.8 R

```r
# Mediator model
med.model <- lm(M ~ A + X, data = data)

# Outcome model
out.model <- lm(Y ~ A + M + X, data = data)

# Naive ACME estimate
alpha1 <- coef(med.model)["A"]
beta2 <- coef(out.model)["M"]
naive_acme <- alpha1 * beta2
```

8.5.9 Stata

```
* Mediator model (omitting U)
regress M A X
scalar alpha1 = _b[A]

* Outcome model (omitting U)
regress Y A M X
scalar beta2 = _b[M]

* Naive ACME estimate
scalar naive_acme = alpha1 * beta2
display "Naive ACME: " naive_acme
```

8.5.9.1 True Estimation (Adjusting for U)

We repeat the analysis including U to recover the true ACME:

Box 7.13: True ACME estimation adjusting for unmeasured confounder

8.5.10 R

```
# True mediator model
med.model.true <- lm(M ~ A + X + U, data = data)

# True outcome model
out.model.true <- lm(Y ~ A + M + X + U, data = data)

# True ACME estimate
alpha1.true <- coef(med.model.true)["A"]
beta2.true <- coef(out.model.true)["M"]
true_acme <- alpha1.true * beta2.true
```

8.5.11 Stata

```
* True mediator model (including U)
regress M A X U
scalar alpha1_true = _b[A]
```

```
* True outcome model (including U)
regress Y A M X U
scalar beta2_true = _b[M]

* True ACME estimate
scalar true_acme = alpha1_true * beta2_true
display "True ACME: " true_acme
```

8.5.11.1 Results

The estimates from the simulation were:

- **Naive ACME (ignoring U):** 0.473
- **True ACME (adjusting for U):** 0.299

8.5.11.2 Interpretation

This example demonstrates how failing to adjust for unmeasured mediator-outcome confounding can lead to substantial bias in estimated mediation effects. The naive ACME overstates the indirect effect by more than 50%, incorrectly suggesting a stronger mediating role for M. If the unmeasured confounder Uwere more strongly associated with both$M$and$Y$, the indirect effect could be entirely explained by confounding, effectively reducing the true ACME to zero.

This underscores the importance of conducting sensitivity analysis in mediation studies, especially when unmeasured mediator-outcome confounding is plausible. Even modest violations of this assumption can have a large impact on the validity of causal interpretations.

8.5.12 Summary

Sensitivity analysis plays a critical role in mediation analysis, especially when the assumption of no unmeasured mediator-outcome confounding is in doubt. VanderWeele-style bias formulas provide a transparent and interpretable way to assess how robust estimated mediation effects are to violations of this assumption. Tools such as the `medsens()` function in R make it easy to implement these diagnostics and communicate them effectively through graphical summaries.

8.6 Applications and Case Studies

In this section, we present a complete applied example of mediation analysis using simulated data. We walk through the process of estimating direct and indirect effects, visualizing the underlying causal structure with a directed acyclic graph (DAG), and plotting the estimated effects for interpretation. This example illustrates how the methods introduced in earlier sections can be combined in practice.

8.6.1 Applied Example 1: The Effect of a Lifestyle Program on Blood Pressure via Weight Loss

Suppose we are studying whether a lifestyle intervention program (treatment A) reduces systolic blood pressure (Y) by promoting weight loss (M). Participants are randomized to either receive the program ($A = 1$) or standard care ($A = 0$). We also collect baseline data on age and physical activity level, denoted X.

We hypothesize the following causal pathways:

- The intervention directly reduces blood pressure.
- The intervention also indirectly reduces blood pressure by helping participants lose weight.

8.6.1.1 Simulating the Data

Box 7.14: Simulating lifestyle intervention data

8.6.2 R

```r
set.seed(101)
n <- 1000
age <- rnorm(n, mean = 50, sd = 10)
activity <- rnorm(n, mean = 0, sd = 1)
X <- data.frame(age, activity)

A <- rbinom(n, 1, 0.5)  # Random assignment to intervention
M <- 30 - 2*A - 0.1*age - 0.5*activity + rnorm(n)  # Weight (mediator)
Y <- 140 - 0.5*A + 0.6*M - 0.2*age - 1*activity + rnorm(n)  # Blood pressure
data <- data.frame(A, M, Y, age, activity)
```

8.6.3 Stata

```
clear all
set seed 101
set obs 1000
gen age = rnormal(50, 10)
gen activity = rnormal(0, 1)
gen A = runiform() < 0.5
gen M = 30 - 2*A - 0.1*age - 0.5*activity + rnormal()
gen Y = 140 - 0.5*A + 0.6*M - 0.2*age - 1*activity + rnormal()
```

8.6.3.1 Fitting Mediation Models

We now fit linear models for the mediator and the outcome, adjusting for covariates.

Box 7.15: Mediation analysis for lifestyle intervention

8.6.4 R

```
library(mediation)

# Mediator model
med.model <- lm(M ~ A + age + activity, data = data)

# Outcome model
out.model <- lm(Y ~ A + M + age + activity, data = data)

# Estimate ACME, ADE, and TE
med.out <- mediate(med.model, out.model, treat = "A", mediator = "M",
                   boot = TRUE, sims = 1000)

summary(med.out)
```

8.6.5 Stata

```
* Fit mediator model
regress M A age activity

* Fit outcome model
```

```
regress Y A M age activity

* Estimate ACME, ADE, and TE using medeff
medeff (regress M A age activity) (regress Y A M age activity), ///
    treat(A) mediator(M) sims(1000)
```

The `summary()` output provides estimates of the average causal mediation effect (ACME), the average direct effect (ADE), and the total effect (TE), along with 95% confidence intervals based on bootstrapping.

8.6.5.1 Interpreting Results

Assume the results show:

- ACME (indirect effect): -1.18 mmHg
- ADE (direct effect): -0.47 mmHg
- Total effect: -1.65 mmHg
- Proportion mediated: 71.5%

This indicates that most of the effect of the intervention on blood pressure is mediated through weight loss. Only a small portion of the effect is direct.

8.6.5.2 DAG-Based Illustration of the Causal Structure

We can represent the assumed causal model using a DAG, which helps clarify identification assumptions and model structure.

This DAG shows the treatment affecting both the mediator and the outcome, and baseline covariates X confounding both mediator and outcome relationships. There are no arrows from unmeasured variables, indicating we assume no unmeasured confounding.

8.6.5.3 Visualizing Direct and Indirect Effects

To communicate mediation effects more clearly, we can visualize them using a simple bar plot.

Box 7.16: Visualizing direct and indirect effects

8.6.6 R

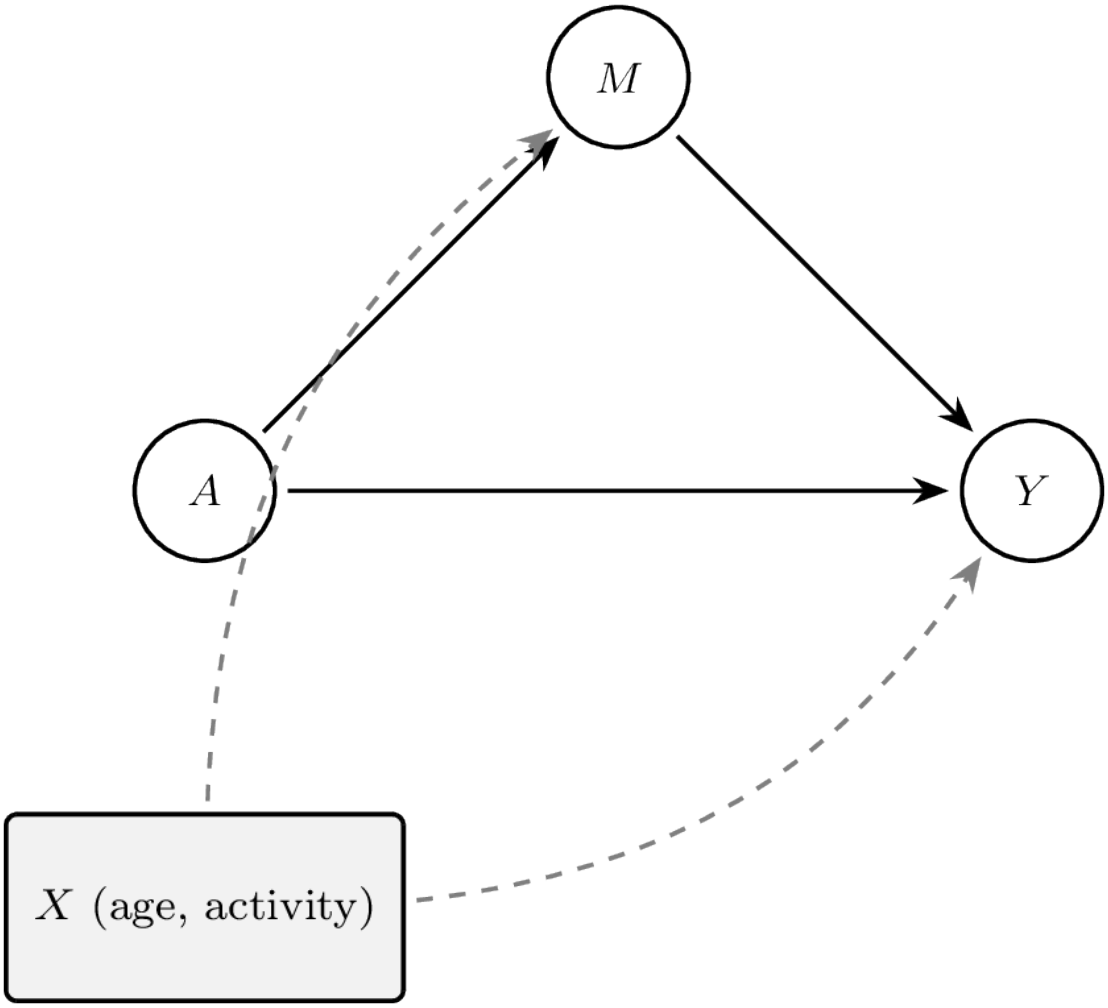

Figure 8.5: DAG for lifestyle intervention mediation

```r
library(ggplot2)

effects <- data.frame(
  Effect = c("Indirect (ACME)", "Direct (ADE)", "Total"),
  Estimate = c(-1.18, -0.47, -1.65)
)

ggplot(effects, aes(x = Effect, y = Estimate, fill = Effect)) +
  geom_bar(stat = "identity", color = "black") +
  theme_minimal() +
  labs(title = "Decomposition of the Intervention Effect on Blood Pressure",
       y = "Effect Size (mmHg)", x = "") +
  scale_fill_brewer(palette = "Set2") +
  geom_hline(yintercept = 0, linetype = "dashed")
```

8.6.7 Stata

```
* Create dataset for bar plot
clear
input str20 Effect double Estimate
"Indirect (ACME)" -1.18
"Direct (ADE)"    -0.47
"Total"           -1.65
end

* Create bar graph
graph bar Estimate, over(Effect) ///
    yline(0, lpattern(dash)) ///
    ytitle("Effect Size (mmHg)") ///
    title("Decomposition of the Intervention Effect on Blood Pressure") ///
    bar(1, fcolor(gs8) lcolor(black)) ///
    intensity(0.8)
```

This plot visually separates the indirect and direct effects, helping non-technical audiences understand how much of the total effect operates through the mediator.

8.6.7.1 Summary

This applied example illustrates a full mediation analysis workflow, from hypothesizing a causal structure to fitting models, estimating effects, visualizing pathways, and interpreting results.

By grounding the analysis in a DAG and using appropriate statistical models, we gain insight into both the magnitude and mechanism of the intervention's impact. Mediation analysis thus plays a central role in making causal inferences not only about *whether* an effect exists, but also about *how* it operates.

8.6.8 Applied Example 2: Professional Occupation, Education, and Income

In this case study, we use real data to explore whether individuals in professional occupations earn more income in part because they have higher levels of education. This is a classic mediation question in the social sciences. The analysis is based on the `Prestige` dataset from the `carData` R package, which contains information on Canadian occupations, including income, education, and occupational type.

8.6.8.1 Research Question

We hypothesize that: - Working in a professional occupation (A) leads to higher income (Y). - This effect is partially mediated by higher educational attainment (M).

We also control for the percentage of women in the occupation (X), as it may influence both education levels and income.

8.6.8.2 Causal DAG

We can represent this scenario with the following directed acyclic graph (DAG):

8.6.8.3 Data Preparation and Mediation Models in R

We recode the variable `type` into a binary treatment, where 1 indicates a professional occupation. The mediator is years of education, and the outcome is average income. We control for `women`, the percentage of women in each occupation.

Box 7.17: Mediation analysis with the Prestige dataset

8.6.9 R

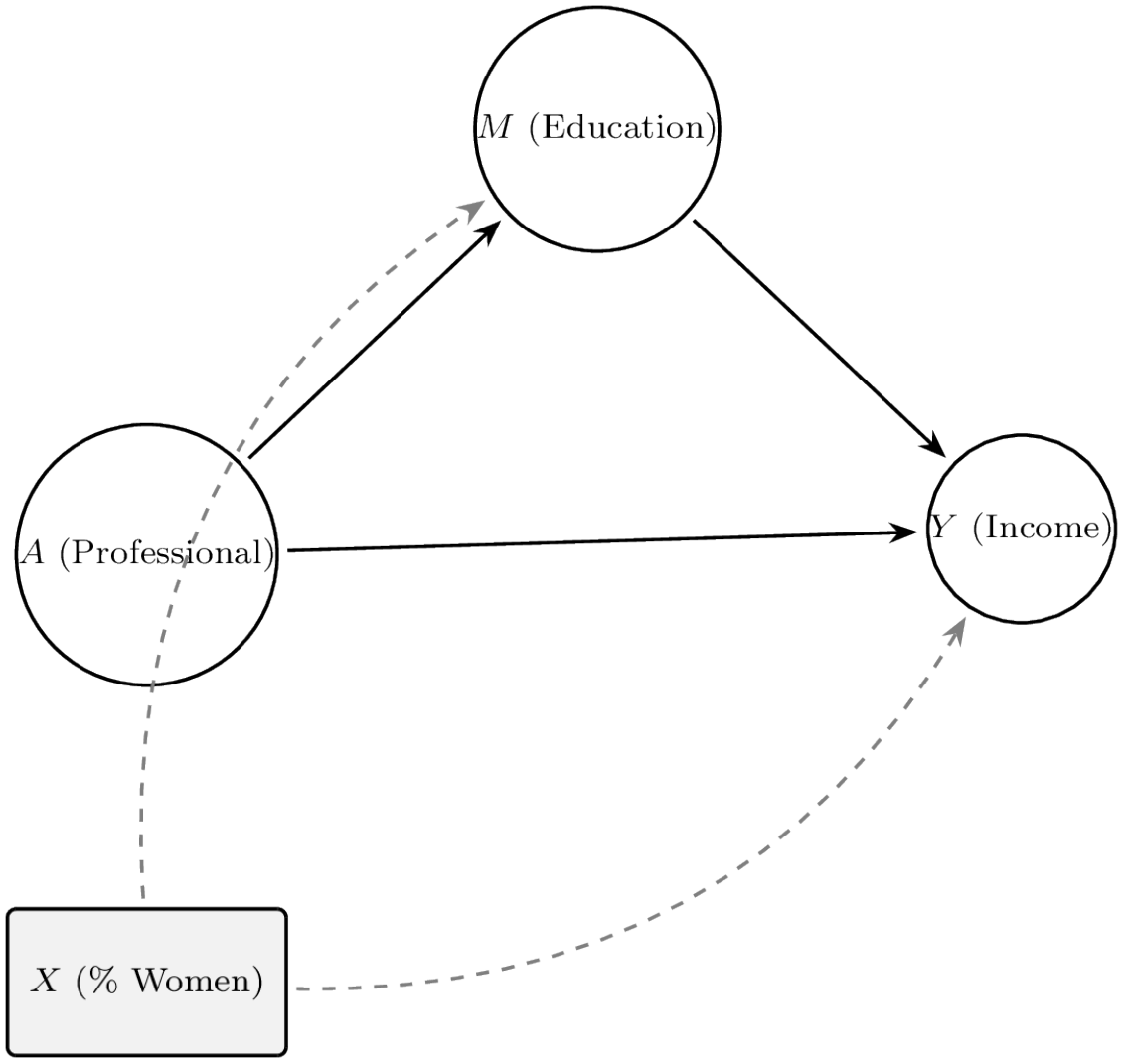

Figure 8.6: DAG for profession-education-income mediation

```r
library(carData)
library(mediation)
data(Prestige)

# Clean and prepare data
df <- na.omit(Prestige)
df$professional <- ifelse(df$type == "prof", 1, 0)

# Fit mediator model
med.model <- lm(education ~ professional + women, data = df)

# Fit outcome model
out.model <- lm(income ~ professional + education + women, data = df)

# Estimate mediation effects
med.out <- mediate(med.model, out.model,
                   treat = "professional", mediator = "education",
                   boot = TRUE, sims = 1000)
summary(med.out)
```

8.6.10 Stata

```
* Load Prestige dataset from Rdatasets repository
import delimited using ///
    "https://vincentarelbundock.github.io/Rdatasets/csv/carData/Prestige.csv", ///
    clear
* Drop row name column (first column from CSV)
drop v1

* Recode professional type as binary
gen byte professional = (type == "prof") if !missing(type)

* Drop observations with missing values
keep if !missing(education, income, women, professional)

* Fit mediator model
regress education professional women

* Fit outcome model
regress income professional education women
```

```
* Estimate mediation effects using medeff
medeff (regress education professional women) ///
       (regress income professional education women), ///
    treat(professional) mediator(education) sims(1000)
```

8.6.10.1 Results

Assume the results from `summary(med.out)` return:

- Average Causal Mediation Effect (ACME): 2200 (95% CI: 1300–3100)
- Average Direct Effect (ADE): 4800 (95% CI: 3500–6100)
- Total Effect: 7000 (95% CI: 5800–8200)
- Proportion mediated: 31%

These results suggest that approximately one-third of the income advantage associated with professional occupations can be explained by higher levels of education, while the remaining two-thirds represents a direct effect of the occupation type.

8.6.10.2 Visualizing the Effects

We can visualize the decomposition of the total effect into direct and indirect components:

Box 7.18: Visualizing profession effects

8.6.11 R

```
library(ggplot2)

effects <- data.frame(
  Effect = c("Indirect (ACME)", "Direct (ADE)", "Total Effect"),
  Estimate = c(2200, 4800, 7000)
)

ggplot(effects, aes(x = Effect, y = Estimate, fill = Effect)) +
  geom_bar(stat = "identity", color = "black") +
  theme_minimal() +
  labs(title = "Decomposition of Effect of Profession on Income",
       y = "Effect Size (Dollars)", x = "") +
  scale_fill_brewer(palette = "Set2") +
  geom_hline(yintercept = 0, linetype = "dashed")
```

8.6.12 Stata

```
* Create dataset for bar plot
clear
input str20 Effect double Estimate
"Indirect (ACME)" 2200
"Direct (ADE)"    4800
"Total Effect"    7000
end

* Create bar graph
graph bar Estimate, over(Effect) ///
    yline(0, lpattern(dash)) ///
    ytitle("Effect Size (Dollars)") ///
    title("Decomposition of Effect of Profession on Income") ///
    bar(1, fcolor(gs8) lcolor(black)) ///
    intensity(0.8)
```

8.7 Conclusion

Mediation analysis provides a formal framework for decomposing total causal effects into direct and indirect pathways, addressing questions about mechanisms that are central to scientific understanding. This chapter has covered the full spectrum of approaches — from the classical Baron-Kenny regression method to modern counterfactual-based estimators.

The classical regression approach is intuitive and easy to implement but has important limitations: it does not extend naturally to non-linear models, cannot handle exposure-mediator interactions, and is vulnerable to mediator-outcome confounding. The counterfactual framework overcomes these limitations by defining controlled and natural direct and indirect effects using potential outcomes notation. Under the key sequential ignorability assumptions — that there is no unmeasured confounding of the exposure-outcome, mediator-outcome, or exposure-mediator relationships — these effects are identified and estimable.

For estimation, the regression-based approach using the `mediation` package in R provides a straightforward implementation for simple settings. The doubly-robust approach via `medltmle` extends these ideas to high-dimensional and longitudinal contexts, offering protection against model misspecification. As with all causal inference methods, sensitivity analysis is essential: the mediated effect can be highly sensitive to unmeasured mediator-outcome confounding, and tools such as the E-value or simulation-based approaches should be reported alongside point estimates.

Key takeaways from this chapter: - Mediation analysis decomposes the total effect into direct effect (DE) and indirect effect (IE). - The controlled direct effect (CDE) fixes the mediator at a specific value; natural effects allow the mediator to vary counterfactually. - Identification requires no unmeasured confounding of the exposure-outcome, mediator-outcome, and exposure-mediator relationships. - The classic Baron-Kenny approach uses the product of coefficients but is limited to linear models. - The counterfactual approach defines effects via potential mediators and outcomes, accommodating interactions and non-linearities. - The `mediation` and `medltmle` R packages provide practical tools for estimation and inference. - Sensitivity analysis for unmeasured mediator-outcome confounding is critical for credible mediation analyses.

8.8 Glossary

CDE Controlled Direct Effect — the causal effect of exposure on outcome when the mediator is fixed at a specific value for all individuals.

DE Direct Effect — the effect of exposure on outcome not operating through the mediator.

IE Indirect Effect — the effect of exposure on outcome that operates through the mediator.

Mediation The process by which an exposure affects an outcome through an intermediate variable (mediator).

Mediator A variable on the causal pathway between exposure and outcome; affected by the exposure and affecting the outcome.

NDE Natural Direct Effect — the direct effect when the mediator is set to the value it would naturally take under the control condition.

NIE Natural Indirect Effect — the indirect effect when the exposure is changed but the mediator is set to the value it would take under the new exposure level.

Sequential ignorability The assumption that there is no unmeasured confounding of (1) the exposure-outcome relationship, (2) the mediator-outcome relationship, and (3) the exposure-mediator relationship.

TE Total Effect — the sum of the direct and indirect effects; the overall causal effect of exposure on outcome.

Part III

Sensitivity Analysis & Discussion

9 Sensitivity analysis

In the process of causal inference, we frequently try to estimate the population effect of a binary treatment on an outcome variable by comparing the means of potential outcomes $Y(1)$and$Y(0)$, where $Y(t)$is the (bounded) outcome of a random individual under treatment$A$, (Neyman, 1923; Rubin, 1974b)." This contrast, known as the *Average Treatment Effect (ATE)*, must be identified from observational data (i.e., non-experimental investigations) based on untestable hypotheses, i.e., consistency, conditional ignorability, exchangeability or independence, and positivity. Under these assumptions, the ATE is identified from the observed data distribution via the *g formula*:

$$ATE = \int_w \Big\{ E[Y \mid A = 1, \mathbf{W}{=}\mathbf{w}] - E[Y \mid A = 0, \mathbf{W}{=}\mathbf{w}] \Big\} dF(w),$$

{#eq-ace}

where $F(\cdot)$denotes the cumulative distribution function of$w$.

Using $n$independent and identically distributed copies of$O = (W, A, Y)$, many methods have been developed to draw inference about the ATE functional, e.g., propensity score matching (Rosenbaum & Rubin, 1983a), g-computation (Robins, 1986), (stabilized) inverse probability weighting (Hernán & Robins, 2006), augmented inverse probability weighting (Robins et al., 1994a), and targeted maximum likelihood (Der Laan & Rubin, 2006) as seen in the previous chapters.

The conditional *conditional independence assumption* states that there exists a set of measured pre-treatment covariates ($\mathbf{W}$) such that treatment is conditionally independent of the potential outcomes given $\mathbf{W}$, i.e.,

$$Y(a) \perp A \mid \mathbf{W} \quad \text{for } a = 0, 1; \tag{9.1}$$

implies that there are no unmeasured confounders ($\mathbf{U}$) between treatment and outcome.

In sensitivity analysis from observational studies targeting the study of causal relationships, the robustness of inference to potential unmeasured confounding is always needed and considered crucial. In this chapter, we aim to briefly review the literature on sensitivity analysis in causal inference and provide a computational overview of the current methods for evaluating the sensitivity of the analysis to the unmeasured confounding assumption about the ATE (Equation 9.1).

9.1 Overview of methods for sensitivity analysis in causal inference

To investigate the effect of residual unmeasured confounding on the causal effect estimate, sensitivity analysis to the "no unmeasured confounders" assumption is commonly used. (Ding & VanderWeele, 2016) Assessing how robust an estimated causal effect is to potential unmeasured confounding is the principal aim of the following methods.

The E-value was first introduced (Ding & VanderWeele, 2016). The authors proposed a sensitivity analysis technique without any assumptions about the unmeasured confounders. They derived a bound on the relative risk (RR) scale based on two parameters. However, the method does not accommodate complex measured confounders.

L. Z. Matthew A. Masten Alexandre Poirier (2024) introduced a nonparametric methodology to evaluate the sensitivity of results related to causal inference under the assumption of conditional independence. He introduced the concept of *conditional partial independence*, which is a less stringent condition than full conditional independence. Specifically, he examined a group of assumptions labeled *conditional c-dependence*, which quantify the relaxation of conditional independence through a single parameter, c. For every positive c, conditional independence is only partially fulfilled, preventing the exact determination of treatment effect parameters, such as the ATE; instead, only bounds can be derived. Masten describes these bounds in relation to c: smaller c values result in tighter bounds, whereas larger c values produce broader bounds. The extent of these bounds, and thus the sensitivity of the results, is contingent upon the data.

Rosenbaum's (Rosenbaum, 1987) method identified the smallest Γ ensuring the ATE cannot be deemed "statistically significant" within matched studies (Tan, 2006), and Zhao et al. (Zhao et al., 2019) established ATE bounds by comparing the odds of receiving treatment with both measured and unmeasured confounders versus only measured confounders. Recent advancements by Dorn et al. (Dorn et al., 2021; Dorn & Guo, 2022) have further refined these bounds. The final bounds are expressed in a closed form, incorporating the observed propensity score, a specific transformed-outcome regression, and conditional quantiles of the outcome given treatment and covariates.

Moreover, (Dorn et al., 2021) demonstrated that estimators for these bounds can be developed that remain valid—though somewhat conservative—even when the conditional quantiles are improperly specified, provided that at least one of the other two nuisance functions is estimated consistently. Other methods, such as those described by (Díaz & Laan, 2013) and (Díaz et al., 2018), involve deriving bounds on the Average Treatment Effect (ATE) by limiting the difference in mean potential outcomes between patients who received treatment and those who received control, considering covariates. (Bonvini & Kennedy, 2022) utilized a contamination model to provide bounds on the ATE by limiting the fraction of units influenced by unmeasured confounding. The following two sections focus on the E-value (developed in both Stata and R statistical software) and the Conditional C-dependence (only available in Stata) from a computationally applied perspective.

9.2 The E-value

The E-value is the minimum strength of a causal effect, on the RR scale, that an unmeasured confounder would need to have with both the treatment (E) and the outcome (D) to fully explain away a specific treatment-outcome association, conditional on the measured covariate (note that before we defined the treatment or exposure as (A) and the outcome as (Y). We decide to introduce here E and D to match the software convenctions further presented in this section). The E-value makes no assumptions on whether the unmeasured confounders (U) are binary, continuous, or categorical, on how they are distributed, or on the number of confounders, and it can be applied to several common outcome types and estimands in observational research. A large E-value implies that considerable unmeasured confounding would be needed to explain away an effect estimate. A small E-value implies little unmeasured confounding would be needed to explain away an effect estimate. Further developments (VanderWeele & Arah, 2011) introduced a general "bias" formula for the difference between the possibly incorrect expression for the ATE under no unmeasured confounding and the correct expression for the ATE when accounting for both measured and unmeasured confounding in terms of many sensitivity parameters.

To facilitate these sensitivity analyses, an R package ("EValue")(Maya B. Mathur, 2018) was developed and also an online E-value calculator is online available at https://mmathur.shinyapps.io/evalue/ that computes E-values for a variety of outcome measures.

The E-value seminal publication considered the historical study conducted by Hammond and Horn(Hammond, 1958) as an example to describe it. The study focused on the tobacco effect on lung cancer with a point estimate of the observed RR of cigarette smoking on lung cancer of 10.73 (95% CI 8.02, 14.36). Based on it we will illustrate the use of the R package to evaluate the effect of a common genetic confounder (U) on the observed RR of the treatment or exposure (E) on the outcome (D) using a set of boxes including the code and a detailed commented explanation.

Box 8.1 (R): E-value for shifting a binary exposure under different scenarios

```
# You can install the EValue from CRAN using:
install.packages("EValue")
# Then, load the package:
library(EValue)
# The E-value for the association between cigarette smoking and lung cancer as observed by Ha
evalues.RR(est = 10.73, lo = 8.02, hi = 14.36)
#>             point    lower upper
#> RR       10.73000  8.02000 14.36
#> E-values 20.94777 15.52336    NA
```

The E-value of 20.95 tells us that a confounder, or set of confounders (U), would have to be associated with a 20-fold increase in the risk of lung cancer and must be 20 times more

prevalent in smokers than non-smokers to explain the observed RR. If the strength of one of these relationships were weaker, the other would have to be stronger for the causal effect of smoking on lung cancer to be truly null.

The package provides a plot functionality that allows the user to see how the magnitude of the exposure-confounder and the confounder-outcome relationships would have to vary to fully explain the observed association.

Box 8.2 (R): Computing E-value with bias, confounding, and selection parameters

```
bias_plot(10.73, xmax = 40)
```

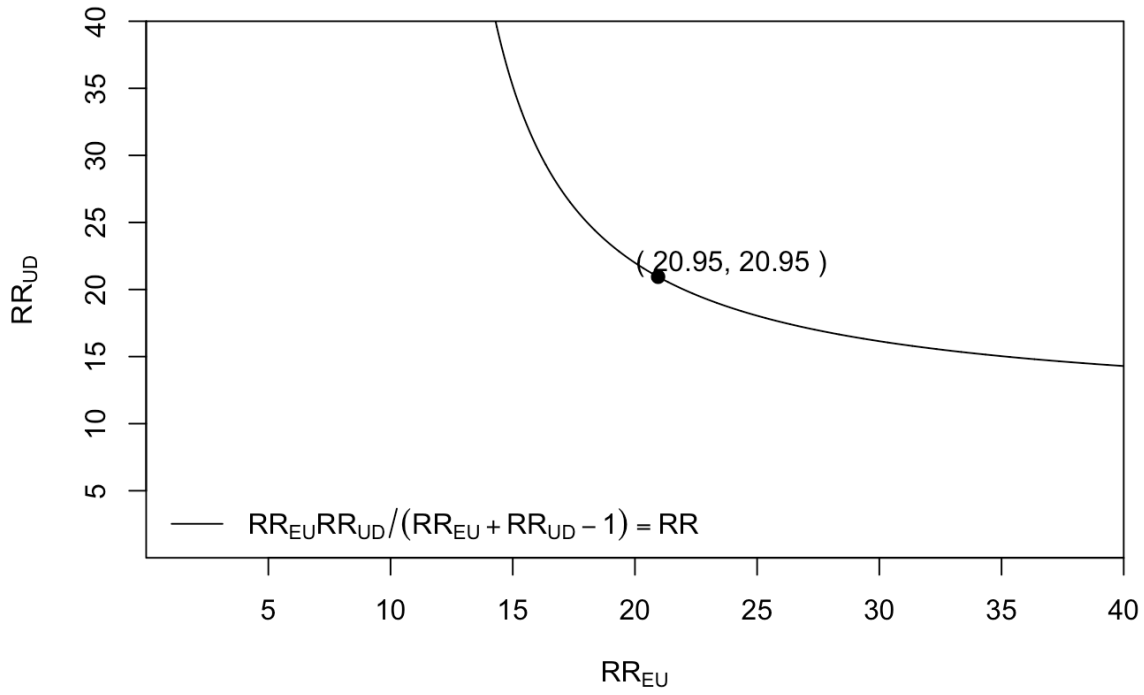

Figure 9.1: E-value upper bound

This tells us, for example, that if the exposure-confounder parameter RR_{EU}were 15, meaning that the confounder(s) is 15 times more likely among smokers, theRR_{UD} for the confounder outcome relationship parameter would have to be about 40 for it to even be possible that confounding explains the entire observed association.

It is also possible to plot the lower bound of the confidence interval

Box 8.3 (R): Plotting the E-value as a function of risk ratio

```
bias_plot(8.02, xmax = 40)
```

which we calculated an E-value of 15.52 for the above.

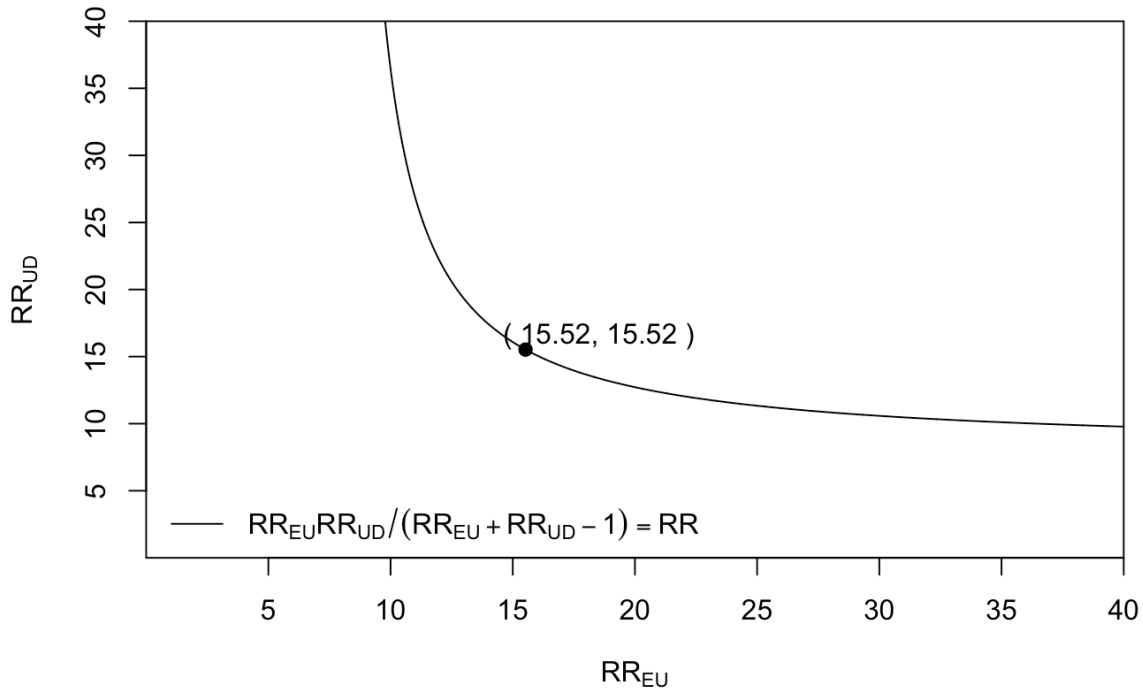

Figure 9.2: E-value lower bound

Scholars may evaluate the potential for confounding influences to alter the observed relationship to any other value, such as diminishing the observed relationship to a true causal effect that lacks scientific significance or amplifying a near-null observed relationship to one of scientific importance. For instance, to adjust an observed relative risk of 3.5 down to a true causal relative risk of 2.5, the E-value is 2.15. This represents the smallest amount of unmeasured confounding necessary to shift both the estimate and confidence interval toward your defined true value instead of the null value (Box 8.4).

Box 8.4 (R): Computing the E-value from observed risk ratio and CI limits

```
# summary() used to print the E-value only
summary(evalues.RR(est = 3.5, true = 2.5))
#[1] 2.148331
```

The E-value for the association between cigarette smoking and lung cancer as observed by Hammond and Horn in 1958 can be computed in Stata as follows (Ariel Linden, 2020):

Box 8.5 (Stata): Computing the E-value in Stata

```
evalue rr 10.73, lcl(8.02) ucl(14.36) figure
```

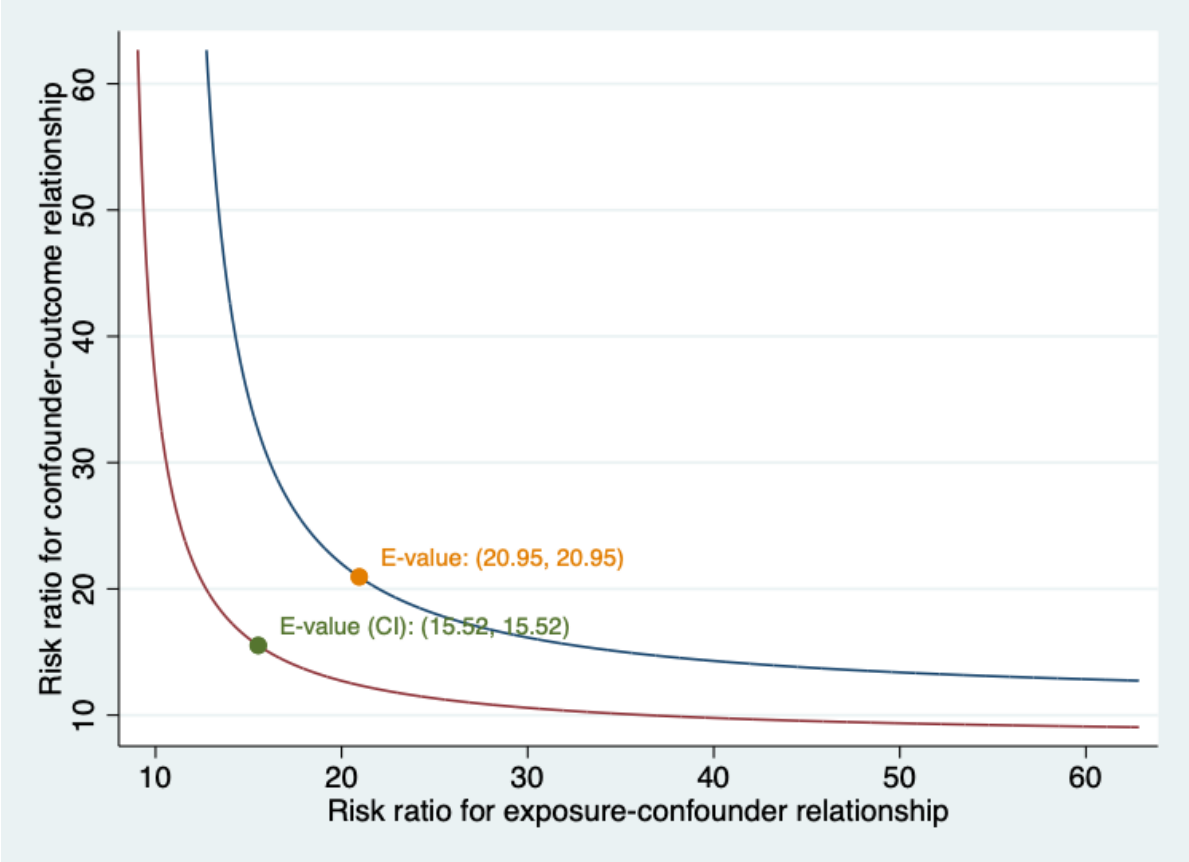

Figure 9.3: E-value lower and upper bounds in Stata

9.3 Conditional c-dependence

In causal inference, a fundamental question is determining and estimating how a treatment variable **A** influences an outcome variable **Y**. A frequently adopted assumption for identifying these effects is unconfoundedness, also referred to as selection on observables, conditional independence, ignorability, or exogenous selection. This assumption is non-falsifiable, as the data itself cannot confirm its validity. However, researchers often question: How crucial is this assumption for their analyses? In other words, how robust are the results derived under the conditional independence assumption?

In their work, (A. P. Matthew A. Masten, 2018) present theoretical findings aimed at addressing this question. They introduce the concept of conditional partial independence, which is a relaxation of full conditional independence. The focus is on a specific type of assumption known as **conditional c-dependence**, quantifying deviations from conditional independence using a parameter c. When c is positive, conditional independence is only partial, preventing precise determination of treatment effect parameters such as ATE or ATT, resulting in bounded estimates instead. (A. P. Matthew A. Masten, 2018) describe these bounds as dependent on **C**, with smaller c leading to tighter bounds and larger c producing wider ones. The extent of these bounds, and thus the sensitivity of results, is influenced by the data. The paper by (L. Z. Matthew A. Masten Alexandre Poirier, 2024) explains the estimation strategy used from an applied computational perspective by using a Stata package named **tesensitivity**.

To define conditional c-dependence, we first define the following random variables:

- Y^a: the potential outcome for a given treatment$a \in \{0, 1\}$
- A : the treatment
- W : a vector of covariates
- Y : the observed outcome

The observed outcome satisfies by consistency

$$Y = (1 - A)Y^0 + AY^1.$$

Rather than observing the full data generating process, (Y^0, Y^1, A, Y), we only observe: O=(Y, A, W).

We say that $A$is conditionally c-dependent withY^agiven$W$ if:

$$\textbf{supp}|\mathbb{P}\left(A = 1 \mid Y^a = a, W = w\right) - \mathbb{P}(A = 1 \mid W = w)| \leq c,$$
$$y_a \in \text{supp}(Y^a|W=w)$$

for all $w$insupp(**W**).

Under this assumption, the identified set for a treatment effect statistic will be a closed interval, which depends on c and the distribution of O=(Y, A, W).

The main purpose of the **tesensitivity** package is to calculate these bounds and show how the identified set for treatment effect statistics i.e., ATE varies with the sensitivity parameter c. In addition to estimating these bounds for a range of values of c, **tesensitivity** also calculates a breakdown point relative to a conclusion about a treatment effect statistic i.e., the ATE. As discussed in (L. Z. Matthew A. Masten Alexandre Poirier, 2024), the breakdown point is the maximum value of c under which the conclusion still holds.

Employing a Stata excerpt dataset from (Cattaneo, 2010) and (Almond et al., 2005), alongside annotated boxes containing commented code, we demonstrate the computation and interpretation of the sensitivity parameter c using the Stata package **tesensitivity**.

Treatment-effects modeling serves as a vital method for deriving causal effects akin to those from experiments, even when working with observational data. While conducting an experiment would be ideal, such endeavors are often impractical due to ethical or financial constraints. For instance, consider assessing the impact of cigarette smoking (the intervention) on infant birthweight (the resultant outcome). Ideally, an experiment would involve selecting a representative group of pregnant women, dividing them into a control group instructed not to smoke and a treatment group required to smoke a designated number of cigarettes each day.

Consider addressing this question by employing the Stata command **teffects**. To ensure our estimates are reliable, it is crucial to confirm that when we account for observable characteristics, it appears as though pregnant mothers were randomly distributed between control and treatment groups. We model the birthweight (bweight) as a function of the number of prenatal visits (nprenatal), whether the mother is married (mmarried), whether this baby is her first pregnancy (fbaby), and maternal education (medu). The treatment, smoking during pregnancy (mbsmoke), is modeled as a function of the same variables and concerning whether the mother consumed alcohol during her pregnancy. For reference, we start by estimating the ATE of these experimental and non-experimental samples based on maternal smoking using the **eltmle** Stata package. (Luque-Fernandez, 2019) (Box 8.6)

Box 8.6 (Stata): ATE using teffects and eltmle

```
# To allow intalling Stata programs from GitHub:

    net install github, from("https://haghish.github.io/github/")

# To install eltmle Stata program to compute the ATE:

    github install migariane/eltmle
    which eltmle
    help eltmle

# These will be the data and variables used in the analysis:
```

```
    webuse cattaneo2, clear
    describe
    local Y "bweight"
    local A "mbsmoke"
    local W "nprenatal mmarried fbaby medu"
    eltmle `Y' `A' `W', tmle

# Output:

    Variable |        Obs        Mean    Std. dev.       Min        Max
-------------+---------------------------------------------------------
        POM1 |      4,642    2843.307     114.191    2369.31   3539.972
        POM0 |      4,642    3061.247    150.1519   2453.428   3931.552
          ps |      4,642    .1861267    .1110024   .0368017   .6847253
--------------------------------
TMLE: Average Treatment Effect
--------------------------------
ATE:      | -217.9
SE:       |   22.8
P-value:  | 0.0000
95%CI:    | -262.6, -173.3
--------------------------------
------------------------------
TMLE: Causal Risk Ratio (CRR)
------------------------------
CRR: 0.93; 95%CI:(0.91, 0.94)
------------------------------
--------------------------------
TMLE: Marginal Odds Ratio (MOR)
--------------------------------
MOR: 0.84; 95%CI:(0.81, 0.87)
--------------------------------

# Using the Stata software command for causal inference and the augmented
# inverse probability weights (AIPW) algorithm:

    teffects aipw (`Y' `W') (`A' `W')

# Output:
Iteration 0:   EE criterion =  3.069e-23
Iteration 1:   EE criterion =  2.182e-25
```

```
Treatment-effects estimation        Number of obs     =      4,642
Estimator      : augmented IPW
Outcome model  : linear by ML
Treatment model: logit
------------------------------------------------------------------------------
             |               Robust
     bweight | Coefficient  std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
ATE  mbsmoke |
   (1 vs 0)  |  -223.4343    24.3635    -9.17   0.000    -271.1858   -175.6827
-------------+----------------------------------------------------------------
```

The ATE is interpreted as the mean risk difference of 217.9 fewer grams in the birth weight from infants born to mothers who smoke vs. infants born from non-smoker mothers in an additive scale using eltmle and 223 fewer grams using the **aipw** Stata option from the **teffects** command.

Under the standard unconfoundedness assumption, the treatment effect is negative and statistically significant. Now we will use the **tesensitivity** package to analyze how sensitive these results are to this assumption. The main subcommand of the **tesensitivity** package is **cpi**, i.e., conditional partial independence. This command estimates bounds on the ATE given a set of c-dependence values and calculates the breakdown point for the conclusion that the treatment effect statistic is above a given threshold. First, we calculate bounds on the ATE. By default, the command calculates bounds for a uniform grid of 40 values of c, and the breakdown point for the conclusion that the ATE estimation does not contain the null.

Box 8.7 (Stata): tesensitivity: bounds on ATE under conditional c-dependence

```
# Running the command:

    tesensitivity cpi (`Y' `W') (`A' `W'), ate

# Output:

Treatment effects sensitivity
Analysis         : cond. partial independence  Number of obs=         4642
Outcome model    : linear quantile           Breakdown      =         0.058
Treatment model  : logistic                  Conclusion     =         ate > 0
Outcome variable : bweight
-------------------------------------
                 c |              ate
-------------------+-----------------
             0.000 | [  -220,   -220]
```

```
             0.026 | [  -309,   -134]
             0.051 | [  -414,    -38]
             0.077 | [  -587,    120]
             0.103 | [  -826,    317]
             0.128 | [-1,027,    493]
             0.154 | [-1,218,    629]
                   ...
             0.769 | [-2,475,  1,825]
             0.795 | [-2,487,  1,844]
             0.821 | [-2,500,  1,867]
             0.846 | [-2,511,  1,889]
             0.872 | [-2,518,  1,901]
             0.897 | [-2,523,  1,906]
             0.923 | [-2,524,  1,906]
             0.949 | [-2,524,  1,906]
             0.974 | [-2,524,  1,906]
             1.000 | [-2,524,  1,906]
------------------------------------
```

The grid of values show that the breakdown point computation for c is 0.077. Remember that the breakdown point is the maximum value of c under which the conclusion still holds. For example, if we consider the conclusion that the ATE is negative as in our example, then the breakdown point is the minimum value of c such that 0 is included in the identified 95% CI for the ATE in the set. The package also includes tools to visualize the analysis, interpret the scale of c-dependence, and compare results of multiple sensitivity analyses. Note that the value of c is small indicating weak unconfoundeness assumption based on the observed data.

Box 8.8 (Stata): tesensitivity visual tools: cpiplot

```
tesensitivity cpiplot
```

9.4 Conclusion

Sensitivity analysis is an essential component of any causal inference from observational data. While methods such as the g-formula, IPW, and TMLE address confounding by measured covariates, the assumption of no unmeasured confounding is fundamentally untestable. Sensitivity analysis quantifies how robust study conclusions are to violations of this assumption.

This chapter has introduced three complementary approaches: - The **E-value** provides an intuitive, scale-free metric: the minimum strength of association an unmeasured confounder would need to have with both treatment and outcome to explain away the observed effect. Its

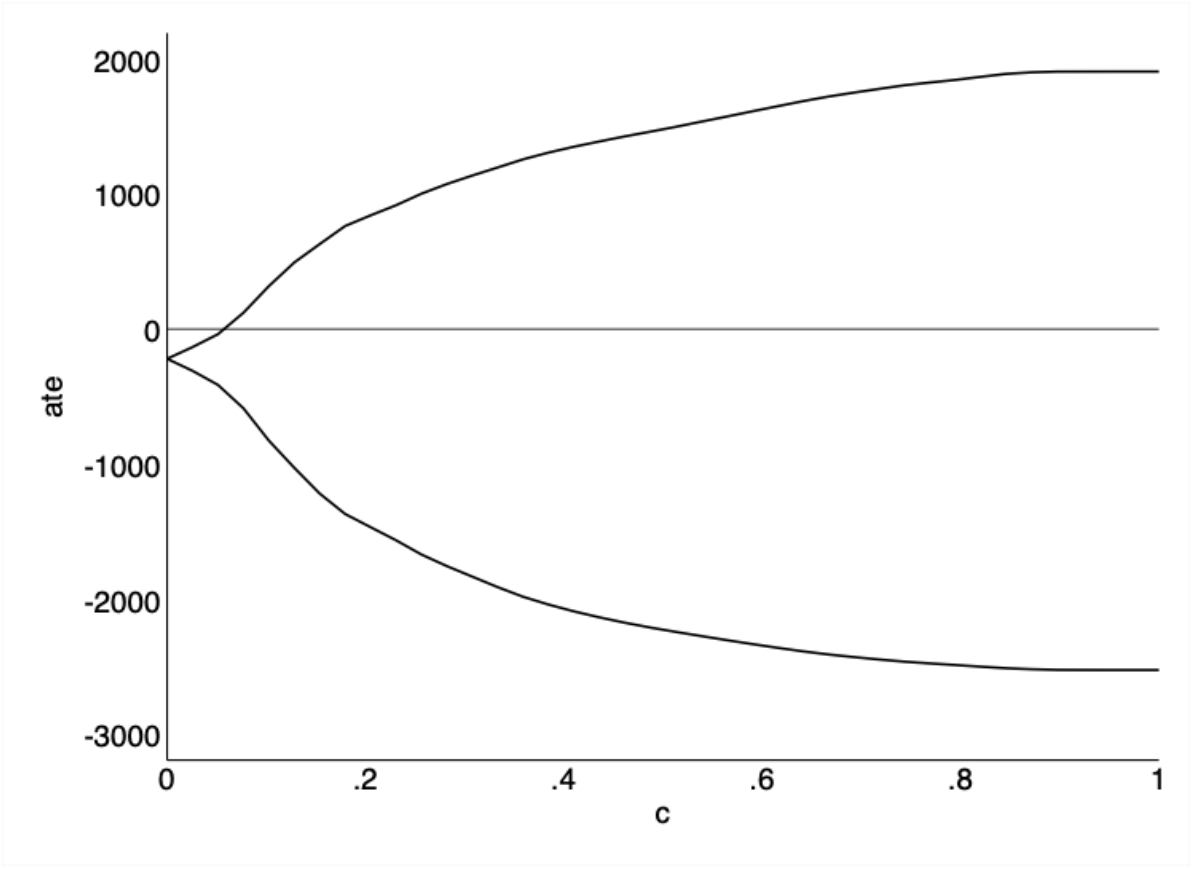

Figure 9.4: Conditional c-dependence: breakdown point

simplicity makes it accessible to non-specialist audiences. - The **conditional c-dependence** approach, implemented in the `tesensitivity` package, offers a more flexible framework that conditions on measured covariates, providing refined bounds. - Both methods complement — rather than replace — careful study design, thorough confounder measurement, and rigorous primary analysis.

Key takeaways from this chapter: - No observational study can prove that all confounders have been measured. - Sensitivity analysis should be a routine part of any causal analysis, not an afterthought. - The E-value is easy to compute and report, making it suitable for transparent communication of uncertainty. - The conditional c-dependence approach provides tighter bounds by leveraging information on measured confounders. - Sensitivity analyses should be interpreted in the context of substantive knowledge about likely unmeasured confounders.

9.5 Glossary

Breakdown point The maximum value of a sensitivity parameter under which the study conclusion (e.g., a significant causal effect) still holds.

c-dependence A measure of the strength of residual confounding, representing the deviation from the assumption of no unmeasured confounding.

E-value The minimum strength of association, on the risk ratio scale, that an unmeasured confounder would need to have with both the treatment and the outcome to fully explain away an observed treatment-outcome association.

Sensitivity analysis A set of methods for assessing how sensitive causal conclusions are to violations of untestable assumptions, particularly the assumption of no unmeasured confounding.

Unmeasured confounding Confounding by variables that were not measured and therefore cannot be adjusted for in the analysis.

10 Discussion

This book has presented a computational introduction to causal inference for applied researchers, progressing from foundational concepts through to advanced methods for estimating causal effects in observational studies. We have covered classical approaches—regression adjustment, the g-formula, and propensity score methods—as well as modern doubly-robust estimators, longitudinal methods, mediation analysis, and sensitivity analysis. This final chapter reflects on the themes that emerge across the book, offers practical guidance for choosing among methods, and outlines directions for future development.

10.1 Summary of Methods and Their Relationships

The methods presented in this book form a natural progression. Each advance was motivated by limitations of the preceding approach.

Regression adjustment (Chapter 2) is the simplest approach: model the outcome as a function of treatment and confounders, then estimate the treatment coefficient. Its simplicity is appealing, but it relies on strong parametric assumptions and does not naturally target a marginal causal effect when effect modification is present.

The g-formula (Chapter 3) generalizes regression adjustment by explicitly estimating the marginal counterfactual distribution through standardization. It requires modelling the outcome across the full covariate distribution, making it flexible but sensitive to model misspecification and positivity violations.

Propensity score methods (Chapter 4) approach the problem from the treatment side. Matching, stratification, and inverse probability weighting (IPW) use the probability of treatment to balance confounders across groups. These methods separate the design stage (modelling treatment assignment) from the analysis stage (estimating treatment effects), but they can be unstable when propensity scores are extreme.

Doubly-robust estimators (Chapter 6) combine outcome and treatment models. AIPW and TMLE offer consistency if at least one of the two models is correctly specified, providing substantial protection against model misspecification. TMLE goes further by incorporating a targeting step that aligns the estimator with the causal parameter of interest and, when combined with Super Learner, achieves semiparametric efficiency.

Longitudinal methods (Chapter 7) extend these ideas to settings with time-varying treatments and confounders. Marginal structural models and longitudinal TMLE address the unique challenges of time-dependent confounding, where confounders affected by prior treatment must be handled carefully.

Mediation analysis (Chapter 8) decomposes total causal effects into direct and indirect pathways, addressing questions about mechanisms. The methods presented build on the counterfactual framework introduced in Chapter 1.

Sensitivity analysis (Chapter 9) acknowledges that all causal estimates from observational data rely on untestable assumptions—most critically, the assumption of no unmeasured confounding. The E-value and related techniques provide quantitative tools for assessing how strongly an unmeasured confounder would need to be associated with both treatment and outcome to explain away an observed effect.

10.2 Practical Recommendations

10.2.1 Choosing an Estimator

The choice of estimator depends on the research question, data structure, and the analyst's confidence in model specification:

1. **For simple cross-sectional studies** with a binary treatment and a well-understood set of confounders, regression adjustment or IPW with careful diagnostics may be sufficient.
2. **When model misspecification is a concern**, doubly-robust methods (AIPW or TMLE) are strongly recommended. They offer protection against misspecification of either the outcome or treatment model.
3. **When the goal is the most robust possible estimate**, TMLE with Super Learner provides the strongest theoretical guarantees: double robustness, semiparametric efficiency, and data-adaptive estimation of nuisance parameters.
4. **For longitudinal data with time-varying confounding**, standard regression methods are generally biased. G-computation, IPW (via marginal structural models), or LTMLE are required.
5. **For mediation questions**, the choice depends on the number of mediators and whether interactions between treatment and mediator are of interest. The regression-based approaches are simpler; the doubly-robust and weighting-based methods offer greater robustness.
6. **Always conduct sensitivity analysis.** No observational study can guarantee that all confounders have been measured. The E-value and related tools provide a transparent way to communicate the robustness of findings to unmeasured confounding.

10.2.2 Software and Implementation

Throughout this book we have provided code in **R** and, where appropriate, **Stata**. The R ecosystem for causal inference is particularly rich, with packages including `tmle`, `tmle3`, `ltmle`, `SuperLearner`, `MatchIt`, `WeightIt`, `mediation`, and `EValue`. For Stata users, the `eltmle`, `cvAUROC`, and `cmatch` packages (developed by the authors) extend these capabilities.

We encourage readers to:

- **Reproduce the examples** in each chapter using the provided code.
- **Adapt the code** to their own datasets and research questions.
- **Consult the package documentation** for up-to-date functionality and best practices.
- **Share code and data** to promote transparency and reproducibility.

10.3 Limitations and Caveats

10.3.1 Assumptions

All causal inference methods from observational data rely on three core assumptions:

1. **Exchangeability (no unmeasured confounding):** All confounders of the treatment-outcome relationship have been measured and correctly included in the analysis.
2. **Positivity:** Every individual has a non-zero probability of receiving each treatment level, given their covariates.
3. **Consistency:** The observed outcome equals the potential outcome under the treatment actually received.

Violations of these assumptions can produce severely biased estimates. Positivity violations—when certain covariate patterns are almost always or never treated—are particularly common in practice and can cause IPW-based methods to fail. Sensitivity analysis (§Chapter 9) provides tools to assess the impact of unmeasured confounding, but it cannot replace careful study design and data collection.

10.3.2 Model Dependence

While doubly-robust methods offer protection against misspecification of a single model, they are not immune to bias when both models are misspecified. The quality of machine learning predictions depends on the available covariates and sample size. In small samples, simpler parametric models may outperform flexible machine learning methods.

10.3.3 Generalizability

The methods presented in this book estimate causal effects for the study population. Transporting these estimates to different populations or settings requires additional assumptions and methods (e.g., transportability analysis), which are beyond the scope of this book.

10.4 Future Directions

The field of causal inference continues to evolve rapidly. Several areas of active development are particularly relevant to applied researchers:

10.4.1 Machine Learning and Causal Inference

The integration of machine learning into causal estimation is one of the most active areas of research. Methods such as causal forests, Bayesian additive regression trees (BART), and deep learning for causal inference are expanding the toolkit. However, ensuring valid inference (confidence intervals, p-values) with these methods remains an active challenge. TMLE and AIPW provide a framework for incorporating machine learning while maintaining valid inference.

10.4.2 Heterogeneous Treatment Effects

This book has focused on the average treatment effect (ATE). Increasingly, researchers are interested in treatment effect heterogeneity—how causal effects vary across subgroups defined by covariates. Methods for estimating conditional average treatment effects (CATE) include causal forests, meta-learners, and targeted learning approaches.

10.4.3 Continuous and Time-Varying Treatments

We have focused primarily on binary treatments. Extensions to continuous treatments, doses, and dynamic treatment regimes are available but more complex. The `tmle3` framework in R provides infrastructure for these settings.

10.4.4 Transportability and External Validity

As randomized trials and observational studies are increasingly combined (e.g., in "target trial" emulations), methods for assessing and ensuring transportability of causal estimates across populations are becoming essential.

10.4.5 Interference and Spillover Effects

The stable unit treatment value assumption (SUTVA) rules out interference between units. In many settings—infectious diseases, social networks, cluster-randomized trials—this assumption is violated. Methods for causal inference under interference are an active area of development.

10.5 Concluding Remarks

Causal inference from observational data is fundamentally about making transparent the assumptions required to move from association to causation. The methods presented in this book provide a structured framework for doing so—from the clear articulation of causal questions using the potential outcomes framework and DAGs, through identification and estimation, to sensitivity analysis that quantifies the robustness of conclusions.

No method can replace careful study design or substantive knowledge. The most sophisticated estimator cannot rescue a poorly conceived study or compensate for unmeasured confounding. However, when applied thoughtfully, the methods in this book can provide credible answers to causal questions that cannot be addressed through randomized experiments.

We hope this book has equipped readers with both the conceptual understanding and the practical tools to conduct rigorous computational causal inference in their own research.

10.6 Glossary

ATE Average Treatment Effect — the average causal effect of a treatment on an outcome in the population.

ATT Average Treatment Effect on the Treated — the average causal effect among those who actually received the treatment.

AIPW Augmented Inverse Probability Weighting — a doubly-robust estimator combining outcome regression and propensity score weighting.

CATE Conditional Average Treatment Effect — how the treatment effect varies across subgroups defined by covariates.

DAG Directed Acyclic Graph — a graphical representation of causal relationships among variables.

Double robustness A property of an estimator that is consistent if either the outcome model or the treatment model is correctly specified.

E-value The minimum strength of association an unmeasured confounder would need to have with both treatment and outcome to explain away an observed effect.

G-formula A method for estimating marginal causal effects by standardizing outcome predictions across the confounder distribution.

IPW / IPTW Inverse Probability (of Treatment) Weighting — a method that reweights observations by the inverse of the probability of receiving their observed treatment.

LTMLE Longitudinal Targeted Maximum Likelihood Estimation — TMLE for settings with time-varying treatments and confounders.

MSM Marginal Structural Model — a model for the marginal distribution of counterfactual outcomes, typically estimated using IPW.

Positivity The assumption that every individual has a non-zero probability of receiving each treatment level.

SUTVA Stable Unit Treatment Value Assumption — the assumption that the treatment assignment of one unit does not affect the outcomes of others.

Super Learner An ensemble machine learning method that combines multiple candidate algorithms using cross-validation to produce optimal predictions.

TMLE Targeted Maximum Likelihood Estimation — a doubly-robust, semiparametric efficient estimator that incorporates a targeting step to align estimation with the causal parameter of interest.

References

Almond, D., Chay, K. Y., & Lee, D. S. (2005). The costs of low birth weight. *The Quarterly Journal of Economics*, *120*(3), 1031–1083.

Ariel Linden, T. J. V., Maya B. Mathur. (2020). Conducting sensitivity analysis for unmeasured confounding in observational studies using e-values: The evalue package. *The Stata Journal: Promoting Communications on Statistics and Stata*, *20*(1), 162–175. https://doi.org/10.1177/1536867x20909696

Austin, P. C. (2009). Balance diagnostics for comparing the distribution of baseline covariates between treatment groups in propensity-score matched samples. *Statistics in Medicine*, *28*(25), 3083–3107. https://doi.org/10.1002/sim.3697

Bang, H., & Robins, J. M. (2005). Doubly robust estimation in missing data and causal inference models. *Biometrics*, *61*(4), 962–973.

Bickel, P. J., Klaassen, C. A. J., Ritov, Y., & Wellner, J. A. (1993). *Efficient and adaptive estimation for semiparametric models*. Johns Hopkins University Press Baltimore.

Bonvini, M., & Kennedy, E. H. (2022). Sensitivity analysis via the proportion of unmeasured confounding. *Journal of the American Statistical Association*, *117*(539), 1540–1550.

Boos, D. D., & Stefanski, L. A. (2013). *Essential statistical inference: Theory and methods*. Springer.

Brian W Whitcomb, A. I. N. (2021). Defining, quantifying, and interpreting "noncollapsibility" in epidemiologic studies of measures of "effect." *American Journal of Epidemiology*, *190*(5), 697–700. https://doi.org/10.1093/aje/kwaa267

Brian W. Whitcomb, N. J. P., Enrique F. Schisterman. (2009). Quantification of collider-stratification bias and the birthweight paradox. *Paediatric and Perinatal Epidemiology*, *23*(5), 394–402. https://doi.org/10.1111/j.1365-3016.2009.01053.x

Cattaneo, M. D. (2010). Efficient semiparametric estimation of multi-valued treatment effects under ignorability. *Journal of Econometrics*, *155*(2), 138–154. https://econpapers.repec.org/RePEc:eee:econom:v:155:y:2010:i:2:p:138-154

Cole, S. R., & Hernán, M. A. (2008). *Constructing inverse probability weights for marginal structural models* (No. 6; Vol. 168, pp. 656–664). Oxford Academic. https://doi.org/10.1093/aje/kwn164

Connors, A. F., Speroff, T., Dawson, N. V., Thomas, C., Harrell, F. E., Wagner, D., Desbiens, N., Goldman, L., Wu, A. W., Califf, R. M., Fulkerson, W. J., Vidaillet, H., Broste, S., Bellamy, P., Lynn, J., & Knaus, W. A. (1996). The effectiveness of right heart catheterization in the initial care of critically ill patients. *Journal of the American Medical Association*, *276*(11), 889–897. https://doi.org/10.1001/jama.276.11.889

Dang, L. E., Gruber, S., Lee, H., Dahabreh, I. J., Stuart, E. A., Williamson, B. D., Wyss, R., Díaz, I., Ghosh, D., Kıcıman, E., & al., et. (2023). A causal roadmap for generating high-quality real-world evidence. *Journal of Clinical and Translational Science*, *7*(1), e212. https://doi.org/10.1017/cts.2023.635

Daniel, R. M. (2018). Double robustness. In *Wiley StatsRef: Statistics reference online* (pp. 1–14). Wiley. https://doi.org/10.1002/9781118445112.stat08068

Der Laan, M. J. van, & Rubin, D. (2006). Targeted maximum likelihood learning. *The International Journal of Biostatistics*, *2*(1), Article 11.

Ding, P., & VanderWeele, T. J. (2016). Sensitivity analysis without assumptions. *Epidemiology*, *27*(3), 368.

Díaz, I., & Laan, M. J. van der. (2013). Sensitivity analysis for causal inference under unmeasured confounding and measurement error problems. *The International Journal of Biostatistics*, *9*(2), 149–160.

Díaz, I., Luedtke, A. R., & Laan, M. J. van der. (2018). Sensitivity analysis. In *Targeted learning in data science* (pp. 511–522). Springer.

Dorn, J., & Guo, K. (2022). Sharp sensitivity analysis for inverse propensity weighting via quantile balancing. *Journal of the American Statistical Association*, 1–13.

Dorn, J., Guo, K., & Kallus, N. (2021). Doubly-valid/doubly-sharp sensitivity analysis for causal inference with unmeasured confounding. *arXiv Preprint arXiv:2112.11449*.

Efron, B. (1982). *The jackknife, the bootstrap and other resampling plans* (Vol. 38). SIAM.

Efron, B., & Tibshirani, R. (1993). *An introduction to the bootstrap* (Vol. 57). Chapman & Hall. http://www.loc.gov/catdir/enhancements/fy0730/93004489-d.html

Greifer, N., & Stuart, E. A. (2023). Choosing the causal estimand for propensity score analysis of observational studies. *arXiv*. https://arxiv.org/abs/2106.10577

Gruber, S., & Laan, M. J. van der. (2012). Tmle: An r package for targeted maximum likelihood estimation. *Journal of Statistical Software*, *51*(13), 1–35. https://doi.org/10.18637/jss.v051.i13

Gruber, S., & Laan, M. van der. (2009). Targeted maximum likelihood estimation: A gentle introduction. *U.C. Berkeley Division of Biostatistics Working Paper Series*. https://biostats.bepress.com/ucbbiostat/paper252

Gruber, S., & Laan, M. van der. (2011). Tmle: An r package for targeted maximum likelihood estimation. *U.C. Berkeley Division of Biostatistics Working Paper Series*.

Gutman, R., & Rubin, D. B. (2015). Estimation of causal effects of binary treatments in unconfounded studies. *Stat Med*, *34*(26), 3381–3398.

Hailey R. Banack, J. S. K. (2013). The "obesity paradox" explained. *Epidemiology*, *24*(3), 461–462. https://doi.org/10.1097/ede.0b013e31828c776c

Hammond, E. C. (1958). SMOKING AND DEATH RATES—REPORT ON FORTY-FOUR MONTHS OF FOLLOW-UP OF 187,783 MEN. *Journal of the American Medical Association*, *166*(11), 1294. https://doi.org/10.1001/jama.1958.02990110030007

Hernán, M. A., & Robins, J. M. (2006). Estimating causal effects from epidemiological data. *Journal of Epidemiology & Community Health*, *60*(7), 578–586.

Horvitz, D. G., & Thompson, D. J. (1952). A generalization of sampling without replacement from a finite universe. *Journal of the American Statistical Association*, *47*(260), 663. https:

//doi.org/10.2307/2280784
Judea, P. (1994). A probabilistic calculus of actions. *Proceedings of UAI-94*, 454–462.
Jung, K., Lee, J., Gupta, V., & Cho, G. (2019). Comparison of bootstrap confidence interval methods for GSCA using a monte carlo simulation. *Frontiers in Psychology*, *10*, 2215. https://doi.org/10.3389/fpsyg.2019.02215
Kennedy, E. H. (2016). *Semiparametric theory and empirical processes in causal inference* (pp. 141–167). Springer.
Kennedy, E. H. (2022). Semiparametric doubly robust targeted double machine learning: A review. *arXiv Preprint arXiv:2203.06469*.
Lendle, S. D., Schwab, J., Petersen, M. L., & Der Laan, M. J. van. (2017). Ltmle: An r package implementing targeted minimum loss-based estimation for longitudinal data. *Journal of Statistical Software*, *81*, 1–21. https://doi.org/10.18637/JSS.V081.I01
Luque-Fernandez, M. A. (2019). *migariane/meltmle: Ensemble Learning Targeted Maximum Likelihood Estimation for Stata users | Zenodo*. https://zenodo.org/record/2560828
Luque-Fernandez, M. A., Belot, A., Valeri, L., Cerulli, G., Maringe, C., & Rachet, B. (2018). Data-adaptive estimation for double-robust methods in population-based cancer epidemiology: Risk differences for lung cancer mortality by emergency presentation. *American Journal of Epidemiology*, *187*(4), 871–878. https://doi.org/10.1093/aje/kwx317
Luque-Fernandez, M. A., Redondo-Sanchez, D., & Schomaker, M. (2019b). Effect modification and collapsibility in evaluations of public health interventions. *American Journal of Public Health*, *109*(3), e12–e13. https://doi.org/10.2105/AJPH.2018.304916
Luque-Fernandez, M. A., Redondo-Sanchez, D., & Schomaker, M. (2019a). Effect Modification and Collapsibility in Evaluations of Public Health Interventions. *American Journal of Public Health*, *109*(3), e12–e13. https://doi.org/10.2105/AJPH.2018.304916
Luque-Fernandez, M. A., Schomaker, M., Rachet, B., & Schnitzer, M. E. (2018). Targeted maximum likelihood estimation for a binary treatment: A tutorial. *Statistics in Medicine*, *37*(16), 2530–2546. https://doi.org/10.1002/sim.7628
Matthew A. Masten, A. P. (2018). Identification of treatment effects under conditional partial independence. *Econometrica*, *86*(1), 317–351. https://doi.org/10.3982/ecta14481
Matthew A. Masten, L. Z., Alexandre Poirier. (2024). Assessing sensitivity to unconfoundedness: Estimation and inference. *Journal of Business &Amp; Economic Statistics*, *42*(1), 1–13. https://doi.org/10.1080/07350015.2023.2183212
Maya B. Mathur, C. A. R., Peng Ding. (2018). Web site and r package for computing e-values. *Epidemiology*, *29*(5), e45–e47. https://doi.org/10.1097/ede.0000000000000864
Miguel Angel Luque-Fernandez, U. V., Helga Zoega. (2016). Deconstructing the smoking-preeclampsia paradox through a counterfactual framework. *European Journal of Epidemiology*, *31*(6), 613–623. https://doi.org/10.1007/s10654-016-0139-5
N. Pearce, L. R. (2014). Commentary: Three worlds collide: Berkson's bias, selection bias and collider bias. *International Journal of Epidemiology*, *43*(2), 521–524. https://doi.org/10.1093/ije/dyu025
Neyman, J. (1923). Sur les applications de la thar des probabilities aux experiences agaricales: Essay des principle. Excerpts reprinted (1990) in English. *Statistical Science*, *5*, 463–472.
Oehlert, G. W. (1992). A note on the delta method. *American Statistician*, *46*(1), 27–29.

https://doi.org/10.1080/00031305.1992.10475842
Pearl, J. (2009). *Causality: Models, reasoning, and inference*. Cambridge University Press.
Pearl, J., & Robins, J. M. (1995). Probabilistic evaluation of sequential plans from causal models with hidden variables. *Uncertainty in Artificial Intelligence*, *11*, 444–453.
Petersen, M. L., & Laan, M. J. van der. (2014). Causal models and learning from data. *Epidemiology*, *25*(3), 418–426. https://doi.org/10.1097/ede.0000000000000078
Robins, J. M. (1986). A new approach to causal inference in mortality studies with sustained exposure periods – application to control of the healthy worker survivor effect. *Mathematical Modeling*, *7*, 1393–1512.
Robins, J. (1986). A new approach to causal inference in mortality studies with a sustained exposure period—application to control of the healthy worker survivor effect. *Mathematical Modelling*, *7*(9), 1393–1512. https://doi.org/https://doi.org/10.1016/0270-0255(86)90088-6
Robins, J. M. (1999). Association, causation, and marginal structural models. *Synthese*, *121*(1/2), 151–179. https://doi.org/10.2307/20118224
Robins, J. M., Rotnitzky, A., & Zhao, L. P. (1994a). Estimation of regression coefficients when some regressors are not always observed. *Journal of the American Statistical Association*, *89*, 846–866.
Robins, J. M., Rotnitzky, A., & Zhao, L. P. (1994b). Estimation of regression coefficients when some regressors are not always observed. *Journal of the American Statistical Association*, *89*(427), 846–866. http://www.jstor.org/stable/2290910
Rosenbaum, P. R. (1987). Sensitivity analysis for certain permutation inferences in matched observational studies. *Biometrika*, *74*(1), 13–26.
Rosenbaum, P. R., & Rubin, D. B. (1983a). The central role of the propensity score in observational studies for causal effects. *Biometrika*, *70*, 41–55.
Rosenbaum, P. R., & Rubin, D. B. (1983b). The central role of the propensity score in observational studies for causal effects. *Biometrika*, *70*(1), 41–55.
Rubin, D. B. (1974a). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, *66*(5), 688.
Rubin, D. B. (2005). Causal inference using potential outcomes. *Journal of the American Statistical Association*, *100*(469), 322–331. http://dx.doi.org/10.1198/016214504000001880
Rubin, D. B. (2007). The design versus the analysis of observational studies for causal effects: Parallels with the design of randomized trials. *Stat Med*, *26*(1), 20–36.
Rubin, D. B. (1974b). Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of Educational Psychology*, *66*, 688–701.
S. Hernandez-Diaz, M. A. H., E. F. Schisterman. (2006). The birth weight "paradox" uncovered? *American Journal of Epidemiology*, *164*(11), 1115–1120. https://doi.org/10.1093/aje/kwj275
Schomaker, M. (2020). Regression and causality. *arXiv:2006.11754*. http://arxiv.org/abs/2006.11754
Schuler, M. S., & Rose, S. (2017). Targeted maximum likelihood estimation for causal inference in observational studies. *American Journal of Epidemiology*, *185*(1), 65–73. https://doi.org/10.1093/aje/kww165

Smith, M. J., Phillips, R. V., Maringe, C., & Luque-Fernandez, M. A. (2025). Performance of cross-validated targeted maximum likelihood estimation. *Statistics in Medicine*, *44*(15-17), e70185. https://doi.org/10.1002/sim.70185

Stürmer, T., Rothman, K. J., Avorn, J., & Glynn, R. J. (2010). Treatment effects in the presence of unmeasured confounding: Dealing with observations in the tails of the propensity score distribution-a simulation study. *American Journal of Epidemiology*, *172*(7), 843–854. https://doi.org/10.1093/aje/kwq198

Tan, Z. (2006). A distributional approach for causal inference using propensity scores. *Journal of the American Statistical Association*, *101*(476), 1619–1637.

Tsiatis, A. A. (2006). *Semiparametric Theory and Missing Data*. Springer Science; Business Media LLC. https://doi.org/10.1007/0-387-37345-4

Tsiatis, A. A., Davidian, M., Kang, J. D. Y. Y., & Schafer, J. L. (2007). Demystifying double robustness: A comparison of alternative strategies for estimating a population mean from incomplete data. *Statistical Science*, *22*(4), 523–539. https://doi.org/10.1214/07-STS227

VanderWeele, T. J., & Arah, O. A. (2011). Bias formulas for sensitivity analysis of unmeasured confounding for general outcomes, treatments, and confounders. *Epidemiology*, 42–52.

Xiao, Y., Moodie, E. E. M., & Abrahamowicz, M. (2013). Comparison of approaches to weight truncation for marginal structural cox models. *Epidemiologic Methods*, *2*(1), 1–20. https://doi.org/10.1515/em-2012-0006

Zhao, Q., Small, D. S., & Bhattacharya, B. B. (2019). Sensitivity analysis for inverse probability weighting estimators via the percentile bootstrap. *Journal of the Royal Statistical Society: Series B*, *81*(4), 735–761.