MS&E 228 (Winter 2026) — Lecture 2 Notes (Student Handout)
Identification by Conditioning (Outcome Regression)

Vasilis Syrgkanis
Stanford University

January 22, 2026

Readings.

- *Applied Causal Inference Powered by ML and AI*, Ch. 5;
- (Optionally) Hernán & Robins, *What If*, Chs. 2–4.

These notes are written in a "chapter" style but follow the lecture flow closely. The goal is to give you something you can read after class that reproduces the narrative, definitions, examples, and calculation steps we covered.

1 From RCTs to conditional randomization

Last lecture, we introduced the potential outcomes framework, defined causal estimands such as the average treatment effect

$$\text{ATE} \;:=\; \mathbb{E}[Y(1) - Y(0)],$$

and saw why randomized experiments are the gold standard: in an RCT,

$$(Y(0), Y(1)) \perp\!\!\!\perp D \quad \Longrightarrow \quad \text{ATE} = \mathbb{E}[Y \mid D = 1] - \mathbb{E}[Y \mid D = 0].$$

Today's lecture makes a conceptually small but practically huge move: we will now **allow the assignment probability to depend on observed covariates** X. This generalization is essential for analyzing a wide range of real-world scenarios where simple randomization does not apply. This happens in many settings:

- in healthcare trials via *conditional/stratified randomization* or *enrichment* (assignment differs across biomarker strata);
- in tech platforms via *algorithmic assignment rules* (e.g. bandit-style experimentation), where randomization depends on user-level features;
- in observational studies, where assignment is not designed by us at all, but we can still hope it is "as-if randomized" after conditioning on measured confounders.

The guiding idea for this entire lecture is the principle of *stratification*:

Compare treated and control among comparable units (same X), then average across the population.

Quick Check

In a stratified trial, the study is still "randomized." Why can the pooled difference $\mathbb{E}[Y \mid D = 1] - \mathbb{E}[Y \mid D = 0]$ fail to equal the ATE when assignment probabilities vary across strata?

Goals for today

In this lecture we

1. Formalize *conditional ignorability* and *overlap/positivity*.
2. Derive the *g-formula* for identification of the ATE and ATT and connect it to outcome regression.
3. Learn why *post-treatment* variables are *bad controls* X (birth-weight paradox).
4. Understand why simple stratification analysis is not applicable when we are faced with continuous covariates X or high dimensional covariates and motivates flexible outcome regression models (and later ML).

2 Healthcare example: PrecISE and stratified assignment

To ground the conditional randomization setting in a real-world context, let's consider a clinical trial example. The PrecISE program (Precision Interventions for Severe and/or Exacerbation-Prone Asthma)[1] is a precision medicine effort that evaluates multiple candidate therapies for severe asthma under a master protocol. A defining feature of precision medicine trials is that baseline biomarkers and phenotypes are used to:

- define subgroups where therapies are expected to be more effective,
- prioritize or enrich enrollment/assignment in those subgroups,
- and more generally make assignment probabilities depend on baseline X.

For our purposes, PrecISE is a clean motivating example because **the assignment mechanism is explicitly tied to observed baseline variables**. To be more concrete, the trial investigated the efficacy of six drugs. For each of these drugs, there exists a biomarker, which is believed based on prior evidence to moderate the efficacy of the drug and to lead to better outcomes (see Table 1). These biomarkers are not mutually exclusive. The trial defined a protocol which essentially increase the probability of assignment of a drug if the corresponding biomarker was observed for a patient. The motivation for such a design was to potentially increase the statistical power of measuring the effect of the prior believed "designated" drug.

2.1 Pedagogical simplification: discrete strata with different propensities

To make the identification logic transparent, let's simplify the PrecISE trial to a pedagogical setting with only two drugs $D \in \{0, 1\}$ (corresponding to some novel *treatment* and a *control* or baseline treatment) and with a small number of baseline biomarker groups $X \in \{\text{High}, \text{Moderate}, \text{Low}\}$, where the names are indicative of the prior belief of magnitude of the novel drug, i.e., believed to be

Intervention	A priori best subgroup	Prevalence
Imatinib	Eos < 300 cells/μl	62%
Clazakizumab	IL-6 > 3.1 pg/μl	33%
Itacitinib	Eos ≥ 300 cells/μl or FeNO > 20 ppb	57%
Cavosonstat	Genotypes	64%
Broncho-Vaxom	Eos ≥ 300 cells/μl	38%
Medium Chain Triglycerides (MCT)	FeNO ≥ 15 ppb	64%

Table 1: Drug and corresponding biomarker designating favorable results for each drug, as well as prevalence of the biomarker in the population.

Biomarker group X	$\mathbb{P}(D = 1 \mid X = x)$	$\mathbb{P}(D = 0 \mid X = x)$	$\mathbb{P}(X = x)$
High	0.70	0.30	0.35
Moderate	0.55	0.45	0.40
Low	0.10	0.90	0.25

high responders, believed to be moderate responders, believed to be low responders. Moreover, we will allow the treatment assignment probabilities to differ across groups.
The key takeaway from this setup is not the particular numbers, but what they imply about the data-generating process: treated units are drawn disproportionately from groups with higher treatment probability. In this scenario, what goes wrong with a naive comparison between treated and control, i.e., $\mathbb{E}[Y \mid D = 1] - \mathbb{E}[Y \mid D = 0]$?

- The treated arm is enriched with a larger fraction of the a-priori high-responder groups.
- A simple *difference in means* between treated and control will therefore be biased upward (it conflates better outcomes due to patient mix with the causal effect of treatment).

2.2 The numeric example: within-group effects and the correct weighted average

Let's see how this confounding problem plays out numerically. For example, suppose that we get the following values at the end of the trial:

X Biomarker group	$\mathbb{P}(X{=}x)$ Mass	$\bar{Y}_{1,x}$ Avg. treated outcome	$\bar{Y}_{0,x}$ Avg. control outcome	$\delta(x)$ Group effect
High	0.35	11.0	9.5	1.5
Moderate	0.40	10.0	9.2	0.8
Low	0.25	9.1	9.0	0.1

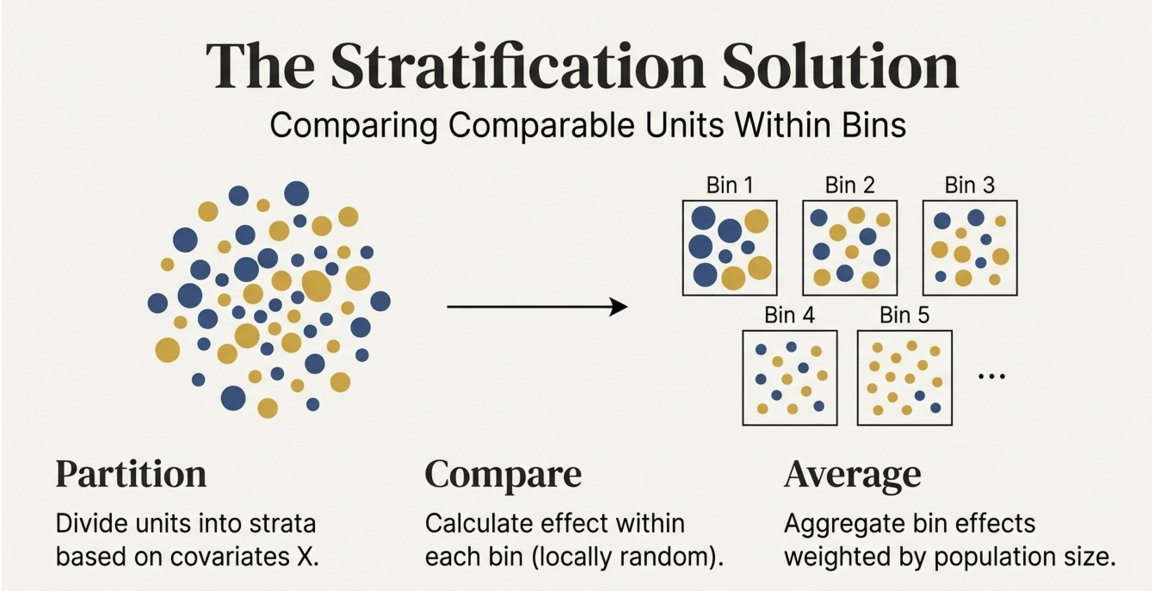
where, within each group, we defined

$$\bar{Y}_{1,x} \coloneqq \mathbb{E}[Y \mid D = 1, X = x], \qquad \bar{Y}_{0,x} \coloneqq \mathbb{E}[Y \mid D = 0, X = x], \qquad \delta(x) \coloneqq \bar{Y}_{1,x} - \bar{Y}_{0,x}.$$

Correct stratified ATE. To correctly calculate the ATE, we must return to the core principle of this lecture:

Compare treated and control among comparable units (same X), then average across the population.

[1] Israel et al. (2021, *JACI*) and Ivanova et al. (2020, *J. Biopharm. Stat.*); see PubMed 33667479 and 32941098.



Let's apply that to this example. We know that within each group the treatment and control was randomized based on some biased coin flip. Therefore, within each group the treated population looks statistically the same as the control population (it might just be that one is larger than the other). Hence, the group effect is a valid measurement of the causal effect within the group. Therefore, if we want to estimate the average effect, we should just average the group effects, weighted by the mass of each group:

$$\begin{aligned}
\text{ATE} &= \sum_x \mathbb{P}(X = x)\,\delta(x) \\
&= \underbrace{0.35}_{\text{mass High}} \cdot \underbrace{1.5}_{\text{effect High}} + \underbrace{0.40}_{\text{mass Moderate}} \cdot \underbrace{0.8}_{\text{effect Moderate}} + \underbrace{0.25}_{\text{mass Low}} \cdot \underbrace{0.1}_{\text{effect Low}} \\
&= 0.525 + 0.320 + 0.025 \\
&= 0.870.
\end{aligned}$$

The following code snippet shows how to implement this stratified estimation approach when the covariates are discrete:

Listing 1: Stratified ATE Estimation for Discrete Covariates

```
def estimate_ate_stratified(Y, D, X):
    """Estimate ATE via stratification for discrete covariates.

    Args:
        Y (np.array): Array of observed outcomes.
        D (np.array): Array of treatment indicators (0 or 1).
        X (np.array): Array of discrete stratum labels.

    Returns:
        float: The estimated Average Treatment Effect (ATE).
    """
    strata = np.unique(X)
    ate = 0.0
    for x in strata:
```

```
        mask = (X == x)
        # Within-stratum effect
        delta_x = np.mean(Y[(mask) & (D == 1)]) - np.mean(Y[(mask) & (D == 0)])
        # Weight by stratum proportion
        ate += np.mean(mask) * delta_x
    return ate
```

Why the pooled difference uses the wrong weights. Let's see how this number compares to a more naive approach that would have simply compared the average outcome of the treated vs the average outcome of the control units. The pooled difference is

$$\mathbb{E}[Y \mid D = 1] - \mathbb{E}[Y \mid D = 0],$$

and it implicitly averages $\bar{Y}_{1,x}$ and $\bar{Y}_{0,x}$ using *different* weights:

$$\mathbb{E}[Y \mid D = 1] = \sum_x \mathbb{P}(X = x \mid D = 1)\, \bar{Y}_{1,x}, \qquad \mathbb{E}[Y \mid D = 0] = \sum_x \mathbb{P}(X = x \mid D = 0)\, \bar{Y}_{0,x}.$$

When $\mathbb{P}(D = 1 \mid X = x)$ differs across x, the treated and control compositions $\mathbb{P}(X = x \mid D = 1)$ and $\mathbb{P}(X = x \mid D = 0)$ differ from each other and from $\mathbb{P}(X = x)$.

To make this fully explicit, use Bayes' rule. First compute the overall treatment rate:

$$\begin{aligned}
\mathbb{P}(D = 1) &= \sum_x \mathbb{P}(D = 1 \mid X = x)\, \mathbb{P}(X = x) \\
&= 0.70 \cdot 0.35 \;+\; 0.55 \cdot 0.40 \;+\; 0.10 \cdot 0.25 \\
&= 0.245 + 0.220 + 0.025 \\
&= 0.490,
\end{aligned}$$

so $\mathbb{P}(D = 0) = 0.510$. Then

$$\mathbb{P}(X = x \mid D = 1) = \frac{\mathbb{P}(D = 1 \mid X = x)\, \mathbb{P}(X = x)}{\mathbb{P}(D = 1)},$$

$$\mathbb{P}(X = x \mid D = 0) = \frac{\mathbb{P}(D = 0 \mid X = x)\, \mathbb{P}(X = x)}{\mathbb{P}(D = 0)}.$$

Numerically:

$$\mathbb{P}(\text{High} \mid D = 1) = \frac{0.70 \cdot 0.35}{0.49} = 0.50, \qquad \mathbb{P}(\text{High} \mid D = 0) = \frac{0.30 \cdot 0.35}{0.51} \approx 0.2059,$$

$$\mathbb{P}(\text{Moderate} \mid D = 1) = \frac{0.55 \cdot 0.40}{0.49} \approx 0.4490, \qquad \mathbb{P}(\text{Moderate} \mid D = 0) = \frac{0.45 \cdot 0.40}{0.51} \approx 0.3529,$$

$$\mathbb{P}(\text{Low} \mid D = 1) = \frac{0.10 \cdot 0.25}{0.49} \approx 0.0510, \qquad \mathbb{P}(\text{Low} \mid D = 0) = \frac{0.90 \cdot 0.25}{0.51} \approx 0.4412.$$

Therefore,

$$\begin{aligned}
\mathbb{E}[Y \mid D = 1] &= 0.50 \cdot 11.0 + 0.4490 \cdot 10.0 + 0.0510 \cdot 9.1 \\
&\approx 5.50 + 4.490 + 0.464 \\
&\approx 10.454,
\end{aligned}$$

$$\begin{aligned}
\mathbb{E}[Y \mid D = 0] &= 0.2059 \cdot 9.5 + 0.3529 \cdot 9.2 + 0.4412 \cdot 9.0 \\
&\approx 1.956 + 3.247 + 3.971 \\
&\approx 9.174.
\end{aligned}$$

So the naive pooled difference is

$$\mathbb{E}[Y \mid D = 1] - \mathbb{E}[Y \mid D = 0] \approx 10.454 - 9.174 = 1.281,$$

which differs from the correct stratified ATE 0.870.

Remark

Takeaway. When treatment probabilities vary across strata, the pooled treated mean and pooled control mean average over *different covariate mixtures*. Conditioning fixes this by forcing "apples-to-apples" comparisons within X.

3 A Tech example: batch Thompson sampling and continuous X

Let's now examine a completely different scenario from the tech sector, which still fits into the general paradigm of conditional randomization but is qualitatively very different in nature. Viewing many settings where conditional randomization arises motivates the need for an abstract mathematical definition of the problem and a solution that would be applicable in a variety of domains, which we will cover in the next section.

Digital experimentation platforms often use algorithmic assignment rules. A common example is (batch) Thompson sampling for an A/B test.[2] For the purpose of the lecture we will simplify the scenario, while maintaining the essence of the problem that we want to highlight. Here is a stylized description of a variant of Thompson sampling. Let W denote raw user features. Each day the platform maintains a belief about the reward difference between options A (equiv. 1) and B (equiv. 0) for a user with characteristics W:

$$Y(1) - Y(0) \mid W \sim \mathcal{N}(\mu(W), \sigma^2(W)).$$

This belief is typically constructed from data the platform collected in prior days. For each user that arrives that day, the platform draws a random variable from the belief

$$Z \sim \mathcal{N}(\mu(W), \sigma^2(W)),$$

and assigns A if $Z > 0$ (set $D = 1$) and B otherwise (set $D = 0$).

Define the reduced covariates

$$X \coloneqq (\mu(W), \sigma(W)).$$

For any user with belief parameters μ, σ, we can write $Z = \mu + \sigma\varepsilon$ with $\varepsilon \sim \mathcal{N}(0, 1)$. Then probability of assigning $D = 1$ (propensity) can be expressed as:

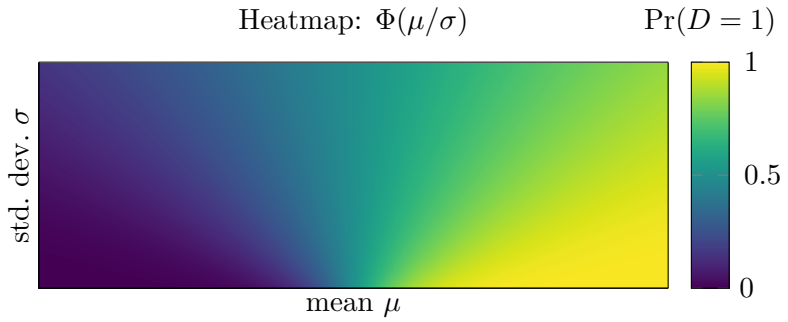
$$\begin{aligned}
\mathbb{P}(D = 1 \mid \mu, \sigma) &= \mathbb{P}(Z > 0 \mid \mu, \sigma) \\
&= \mathbb{P}\left(\varepsilon > -\frac{\mu}{\sigma}\right) \\
&= \Phi\left(\frac{\mu}{\sigma}\right),
\end{aligned}$$

where Φ is the standard normal CDF.

This setting carries the same high level property of conditional randomization as the healthcare example. Conditional on $X = (\mu, \sigma)$, the only remaining randomness in D is the random seed

[2] See, e.g., *Using a Multi-Armed Bandit with Thompson Sampling to Identify Responsive Dashers*

ε. And this random seed has nothing to do with the potential outcome processes, nor with the covariates X. It is a fully independent random seed. From this perspective, this setting also has the property that conditional on X the treatment can be viewed as stemming from a randomized trial. Albeit, unlike in the healthcare example, X is now continuous: you should not expect to see many repeated exact covariate profiles. So we cannot form small tables of strata and compute within-cell means. At best we can visualize the propensity $\mathbb{P}(D = 1 \mid \mu, \sigma)$ as a 2D surface or heatmap.



Despite this fact, we will next define a general setting and a general identification recipe that encompasses both examples and generalizes the lectures mantra.

Compare treated and control among comparable units (same X), then average across the population.

Albeit in this general recipe, we will no longer be able to implement the solution using simple tables and simple weighted averages. We will have to rely on some form of outcome regression modeling.

4 Assignment model and conditional ignorability

A compact way to formalize "as-if randomized conditional on X" is by assuming that the assigned treatment D is determined by the equation:

$$D = f(X, \varepsilon), \qquad \varepsilon \perp\!\!\!\perp \{(Y(0), Y(1)), X\}.$$

Intuitively, ε is the random seed (e.g., coin flip) and is unrelated to outcomes.

A key consequence of this assignment model is the property of conditional ignorability (also called conditional exogeneity):

$$(Y(0), Y(1)) \perp\!\!\!\perp D \mid X. \qquad \text{(conditional ignorability)}$$

In words: once we fix the covariates X, treated and control units are comparable.

The formal proof why the conditionally randomized assignment model implies conditional ignorability is simple:

Proof.

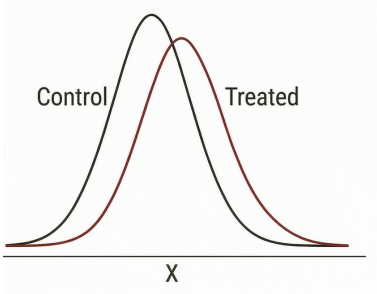
$$\begin{aligned} &\varepsilon \perp\!\!\!\perp \{(Y(0), Y(1)), X\} \\ &\quad \Rightarrow \quad \varepsilon \perp\!\!\!\perp (Y(0), Y(1)) \mid X \\ &\quad \Rightarrow \quad f(X, \varepsilon) \perp\!\!\!\perp (Y(0), Y(1)) \mid X \\ &\quad \Rightarrow \quad D \perp\!\!\!\perp (Y(0), Y(1)) \mid X. \end{aligned}$$

□

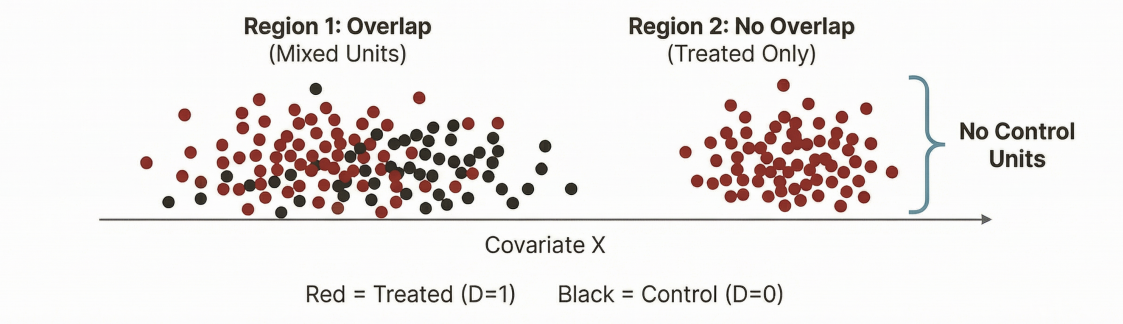
5 The identification recipe: assumptions ⇒ identification formulas

We now formalize the "compare within X, then average" recipe in this broad setting. We have already seen that conditional randomization implies the key property of conditional ignorability. Does this suffice to identify causal effects? Conditioning alone is not enough; we also need a support condition. We say *overlap* (or positivity) holds if

$$0 < \mathbb{P}(D = 1 \mid X = x) < 1 \qquad \text{for all } x \text{ in the support of } X. \qquad \text{(overlap/positivity)}$$



Interpretation: for each covariate profile that occurs in the population, we observe *both* treated and control units with non-zero probability. This is a support condition. For instance, if $\mathbb{P}(D = 0 \mid X = x) = 0$, then we cannot learn what $Y(0)$ looks like at x from data, no matter how many samples we have. To even have a chance to identify $\mathbb{E}[Y(0) \mid X = x]$ using observables, we need to at least observe some outcomes under control for units that look like x. Note that if $\mathbb{P}(D = 0 \mid X = x) = 0$ then even the conditional expectation $\mathbb{E}[Y \mid D = 0, X = x]$ is undefined from data, since the conditioning event is not in the support of the distribution.



5.1 Identifying the ATE: the g-formula

Having defined conditional ignorability and positivity, we are now ready to prove our key identification formulas. First, a key step is showing that conditional mean counterfactuals are identified.

Proposition 1. *If conditional ignorability and overlap hold, then for $d \in \{0, 1\}$,*

$$\mathbb{E}[Y(d) \mid X] = \mathbb{E}[Y \mid D = d, X].$$

Proof.

$$\begin{aligned} \mathbb{E}[Y(d) \mid X] &= \mathbb{E}[Y(d) \mid D = d, X] && \text{(by conditional ignorability; overlap ensures } \mathbb{P}(D = d \mid X) > 0) \\ &= \mathbb{E}[Y \mid D = d, X] && \text{(by consistency: } Y = Y(d) \text{ when } D = d). \end{aligned}$$

□

Given this we can immediately show identifiability of the conditional average treatment effect (CATE), defined as:

$$\mathrm{CATE}(X) \;\coloneqq\; \mathbb{E}[Y(1) - Y(0) \mid X].$$

Proposition 2. *Under ignorability + overlap,*

$$\mathrm{CATE}(X) = \mathbb{E}[Y \mid D = 1, X] - \mathbb{E}[Y \mid D = 0, X].$$

Averaging over the population distribution of X yields the **g-formula for the ATE**:

Theorem 1. *Under ignorability + overlap*

$$\mathrm{ATE} = \mathbb{E}[\mathbb{E}[Y \mid D = 1, X] - \mathbb{E}[Y \mid D = 0, X]] \qquad \text{(g-formula)}$$

Proof.

$$\begin{aligned}
\mathrm{ATE} &= \mathbb{E}[Y(1) - Y(0)] \\
&= \mathbb{E}\big[\mathbb{E}[Y(1) - Y(0) \mid X]\big] && \text{(law of iterated expectations)} \\
&= \mathbb{E}\big[\mathrm{CATE}(X)\big] \\
&= \mathbb{E}[\mathbb{E}[Y \mid D = 1, X] - \mathbb{E}[Y \mid D = 0, X]]\,.
\end{aligned}$$

□

5.2 Identifying the ATT and one-sided assumptions

The average treatment effect on the treated (ATT) is

$$\mathrm{ATT} \coloneqq \mathbb{E}[Y(1) - Y(0) \mid D = 1].$$

ATT often requires weaker assumptions to be identified than ATE, because among treated units we observe $Y(1)$ directly and only need to impute $Y(0)$. For this reason, it is sufficient to assume one-sided versions of ignorability and overlap:

One-sided ignorability: $Y(0) \perp\!\!\!\perp D \mid X$.

One-sided overlap: $\mathbb{P}(D = 0 \mid X = x) > 0$ for all x with $\mathbb{P}(X = x \mid D = 1) > 0$.

Remark

One-sided overlap allows for regions of X where units are never treated. The reason is that we only care to estimate the average effect for units that have some positive probability of treatment. Thus we don't care if there are regions were we don't have any information about $Y(1)$. We don't care to average the effect for these regions.

Remark

One-sided ignorability allows for some confounding in the assigned treatment. It only states that treated and control populations are comparable in terms of how they would have performed under control. But they are allowed to be statistically different in terms of how they would have performed under treatment. For instance, if there are unobserved factors U that are un-correlated with the baseline response $Y(1)$, but are correlated with the individual effect $Y(1) - Y(0)$, then these unobserved factors could be influencing the assigned treatment. In a conditionally randomized assignment setting that we are working with in this section, this weakening of the assumption has no bite, since the two-sided ignorability also holds. However, this weakening can be powerful in observational settings that we discuss in the next section.

Under these conditions, we can prove the following **g-formula for the ATT**:

Theorem 2. *Under one-sided conditional ignorability + one-sided overlap*

$$\begin{aligned} \mathrm{ATT} &= \mathbb{E}[Y \mid D=1] - \mathbb{E}[\mathbb{E}[Y \mid D=0, X] \mid D=1] \\ &= \mathbb{E}[Y - \mathbb{E}[Y \mid D=0, X] \mid D=1] \qquad \text{(ATT g-formula)} \end{aligned}$$

Remark

Practical meaning. To estimate ATT, you only need to model the *control* outcome regression $\mathbb{E}[Y \mid D=0, X]$ and then predict counterfactual untreated outcomes for treated units, subtract them from the observed outcome and average among treated units.

Proof.

$$\mathrm{ATT} = \mathbb{E}[Y(1) - Y(0) \mid D=1] = \mathbb{E}[Y \mid D=1] - \mathbb{E}[Y(0) \mid D=1]$$

$$\begin{aligned} \mathbb{E}[Y(0) \mid D=1] &= \mathbb{E}[\mathbb{E}[Y(0) \mid X, D=1] \mid D=1] && \text{(by tower rule)} \\ &= \mathbb{E}[\mathbb{E}[Y(0) \mid X] \mid D=1] && \text{(by one-sided ignorability)} \\ &= \mathbb{E}[\mathbb{E}[Y(0) \mid X, D=0] \mid D=1] && \text{(by one-sided (ignorability + overlap))} \\ &= \mathbb{E}[\mathbb{E}[Y \mid D=0, X] \mid D=1] && \text{(by consistency)} \end{aligned}$$

□

6 Observational studies: the same formulas, harder assumptions

In observational studies, the g-formula remains algebraically correct under the same assumptions of conditional ignorability and overlap. The difference is epistemic: in a designed experiment, conditional ignorability can be argued from the assignment mechanism; in observational data, it becomes a **no unmeasured confounding** assumption.

6.1 No unmeasured confounding (untestable)

Conditional ignorability

$$(Y(0), Y(1)) \perp\!\!\!\perp D \mid X,$$

is often summarized as **no unmeasured confounding**. Concretely, we assume there is no unobserved U that both influences treatment assignment and predicts potential outcomes after conditioning on X. This assumption is generally *untestable from the dataset alone*. Diagnostics like balance checks and overlap plots can reveal warning signs, but they cannot certify the absence of unmeasured confounding.

6.2 Homework example: NHEFS (smoking cessation → weight change)

In the NHEFS homework example:

- D: indicator for quitting smoking (versus continuing),
- Y: weight change over follow-up,

- X: baseline covariates (age, sex, baseline weight, smoking intensity, health measures, etc.).

Conditional ignorability means: *among people with the same baseline profile X, quitting is as-if random relative to potential weight changes.* This is a strong assumption; it cannot be proven from the dataset alone.

Discussion

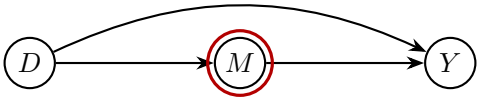
Overlap diagnostics can show when the conditioning approach is implausible (e.g. no controls for some treated covariate profiles). Why can they never *prove* no unmeasured confounding?

7 Bad controls: why "control for everything" can fail

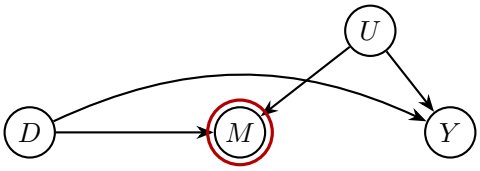
A recurring temptation in applied work in observational studies is to use the "kitchen sink" strategy: include every available variable as a control. Causal inference requires more care, because conditioning on the wrong variables can change the estimand or introduce bias.

For instance, **conditioning on post-treatment variables can induce bias** and it is always a safe strategy to never condition on post-treatment variables. Here are two major sources of bias when conditioning on post-treatment variables:

(1) Mediator: conditioning blocks part of the effect. If M lies on the causal pathway $D \to M \to Y$, conditioning on M removes the indirect effect through M.

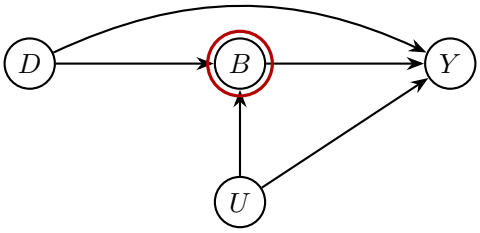


(2) Collider: conditioning opens a spurious path. If M is a collider ($D \to M \leftarrow U$), conditioning on M can induce dependence between D and U, which then biases comparisons if $U \to Y$.



7.1 Birth-weight paradox: intuition and a simple linear illustration

Let's look at a real world empirical example that is a classic in epidemiology and is known as the **Birth-weight paradox**. Empirically, smoking increases infant mortality. Yet conditioning on low birth weight can create a reversal where smoking appears protective within the low-birth-weight group. This is a concrete instance of collider bias due to conditioning on a post-treatment variable. In this scenario, the post-treatment variable B (birth weight) is affected by D (smoking) and also by competing risks U, with Y (infant mortality) downstream:



The intuition of collider bias is the following: among low-birth-weight infants, smokers are more likely to have low birth weight *because of smoking*, so they are less likely to have other hidden risks U. Non-smokers with low birth weight are more likely to have severe competing risks U. So conditioning on B and comparing treatment vs. control, we compare:

- smokers with relatively *fewer* competing risks (U low)
- to non-smokers with relatively *more* competing risks (U high),

which can make smoking appear "protective" within the low-B group.

We can also illustrate this mathematically through a simple linear model. Suppose that all relationships are linear:

$$\begin{aligned} Y &\coloneqq D + B + \kappa U + \varepsilon_Y, \\ B &\coloneqq D + U + \varepsilon_B, \\ D &\coloneqq \varepsilon_D, \end{aligned}$$

with independent standard normal noise variables and $U \sim \mathcal{N}(0,1)$. Conditioning on (B, D) implies

$$\mathbb{E}[Y \mid B, D] = D + B + \kappa\, \mathbb{E}[U \mid B, D],$$

Moreover, from the B equation, we can write $B - D = U + \varepsilon_B$. Thus $B - D$ is a noisy gaussian signal of the variable U, which is also drawn from a gaussian. Thus a simple posterior calculation yields that: $\mathbb{E}[U \mid D, B] = (B - D)/2$, i.e. if $D = 0$ (non-smoker), then we have higher belief about the value of U (existence of other competing risks), as compared to when $D = 1$ (smoker). Bottom line:

$$\mathbb{E}[Y \mid B, D] = \left(1 - \frac{\kappa}{2}\right) D + \left(1 + \frac{\kappa}{2}\right) B.$$

If $\kappa > 2$, the conditional coefficient on D becomes negative, capturing the qualitative reversal.

8 Operationalizing the g-formula: outcome regression

Identification expresses causal estimands in terms of conditional expectations. When covariates X are discrete and take a small number of values, these conditional expectations can be replaced by simple sub-group averages. However, beyond this setting we need to estimate conditional expectations by training predictive outcome regression models. The following figure summarizes the typical workflow.

8.1 Outcome regression functions and plug-in estimators

Define

$$g(d, x) \;\coloneqq\; \mathbb{E}[Y \mid D = d, X = x].$$

Given i.i.d. data $\{(Y_i, D_i, X_i)\}_{i=1}^n$ we first fit $\hat{g}$ using some regression approach. In this lecture, we will mostly focus on linear regression, but you can think of your favorite (ML) regression approach. A regression or a conditional expectation is nothing but a pure prediction problem. It asks us to predict the outcome Y from the treatment D and the controls X.

In practice, there are two common approaches to fitting these regression models:

- **Single interacted model (S-learner):** fit one model $\hat{g}$ for $g(d, x) = \mathbb{E}[Y \mid D = d, X = x]$ predicting the outcome Y using both the treatment D and controls X as input features (if your model is linear, then advisably include interactions between treatment and features of X, to allow for treatment effect heterogeneity).
- **Two-model (T-learner):** first write: $g(d, x) = d \cdot m_1(x) + (1 - d) \cdot m_0(x)$ and fit two regression models; a model $\hat{m}_1$ predicting the outcome from X using only treated units and a model $\hat{m}_0$ predicting the outcome from X using control units.

After fitting such predictive models, we can calculate an estimate of the ATE using simple empirical averages over the data. Note that the two modeling approaches are semantically equivalent, in that $g(d, x) = m_d(x)$, but they can lead to different estimation strategies.

$$\widehat{\text{ATE}} = \frac{1}{n} \sum_{i=1}^{n} \Big(\hat{g}(1, X_i) - \hat{g}(0, X_i) \Big) = \frac{1}{n} \sum_{i=1}^{n} \Big(\hat{m}_1(X_i) - \hat{m}_0(X_i) \Big),$$

$$\widehat{\text{ATT}} = \frac{1}{n_1} \sum_{i: D_i = 1} \Big(Y_i - \hat{m}_0(X_i) \Big), \qquad n_1 := \sum_{i=1}^{n} \mathbb{I}\{D_i = 1\}.$$

This plug-in approach is the simplest way to operationalize the g-formula. While effective, it has limitations, and we will discuss more advanced techniques (such as doubly robust methods) in later lectures on in the course.

Listing 2: T-Learner: ATE Estimation via Outcome Regression

```python
def t_learner(Y, D, X, reg):
    """Estimate ATE using the T-learner (two separate models).

    Args:
        Y (np.array): Array of observed outcomes.
        D (np.array): Array of treatment indicators (0 or 1).
        X (np.array): Covariate matrix.
        reg: A regression model with fit(X, y) and predict(X) methods.

    Returns:
        float: The estimated Average Treatment Effect (ATE).
    """
    from sklearn.base import clone

    # Fit separate models for treated and control
    m1 = clone(reg).fit(X[D == 1], Y[D == 1])
    m0 = clone(reg).fit(X[D == 0], Y[D == 0])

    # Predict for all units and average the difference
    return np.mean(m1.predict(X) - m0.predict(X))
```

Listing 3: S-Learner: ATE via Single Model

```python
def s_learner(Y, D, X, reg):
    """Estimate ATE using the S-learner (single model with treatment).

    Args:
        Y (np.array): Array of observed outcomes.
        D (np.array): Array of treatment indicators (0 or 1).
        X (np.array): Covariate matrix.
        reg: A regression model with fit(X, y) and predict(X) methods.

    Returns:
        float: The estimated Average Treatment Effect (ATE).
    """
    # Augment features with treatment indicator
    X_aug = np.column_stack([D, X])
    model = reg.fit(X_aug, Y)

    # Predict under D=1 and D=0 for all units
    X_treated = np.column_stack([np.ones(len(X)), X])
    X_control = np.column_stack([np.zeros(len(X)), X])

    return np.mean(model.predict(X_treated) - model.predict(X_control))
```

Listing 4: ATT Estimation via Outcome Regression

```python
def estimate_att_regression(Y, D, X, reg):
    """Estimate ATT using outcome regression on controls.

    Args:
        Y (np.array): Array of observed outcomes.
        D (np.array): Array of treatment indicators (0 or 1).
        X (np.array): Covariate matrix.
        reg: A regression model with fit(X, y) and predict(X) methods.

    Returns:
        float: The estimated Average Treatment Effect on the Treated (ATT).
    """
    # Fit model on control units only
    m0 = reg.fit(X[D == 0], Y[D == 0])

    # Predict counterfactual for treated, subtract from observed
    return np.mean(Y[D == 1] - m0.predict(X[D == 1]))
```

8.2 Connection to adjustment via linear regression

A classical special case of this workflow is to use linear regression to model the conditional expectation function $\mathbb{E}[Y \mid D, X]$. If one assumes that:

$$\mathbb{E}[Y \mid D, X] = \beta_0 + \alpha D + \gamma^\top \phi(X),$$

where $\phi(X)$ are some hard-coded engineered features of the raw covariates X (e.g. polynomials, splines, bins, interactions among covariates). Then note that

$$m_1(x) - m_0(x) = \alpha, \qquad \Rightarrow \qquad \text{ATE} = \alpha.$$

Thus, under the correct linear specification of the conditional expectation function, the OLS coefficient on D is exactly the g-formula plug-in estimate.

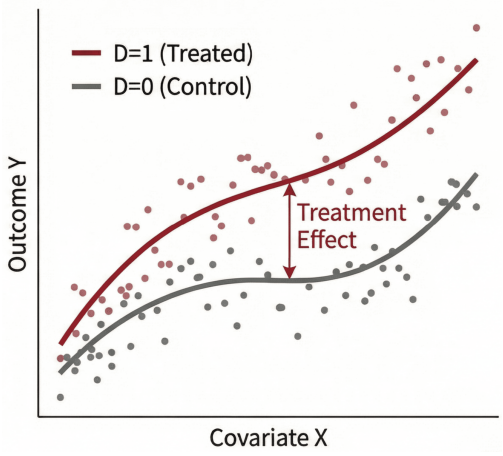
Listing 5: Linear Regression without Interactions (Constant Treatment Effect)

```python
def ate_linear_no_interactions(Y, D, X, phi):
    """Estimate ATE via linear regression without treatment interactions.

    Assumes: E[Y|D,X] = beta_0 + alpha*D + gamma'*phi(X)
    Under this model, ATE = alpha (constant across X).

    Args:
        Y (np.array): Array of observed outcomes.
        D (np.array): Array of treatment indicators (0 or 1).
        X (np.array): Raw covariate matrix.
        phi: Featurizer function phi(X) -> transformed features.

    Returns:
        float: The estimated ATE (coefficient on D).
    """
    from sklearn.linear_model import LinearRegression

    # Build design matrix: [D, phi(X)]
    features = np.column_stack([D, phi(X)])
    model = LinearRegression().fit(features, Y)

    # ATE is the coefficient on D (first covariate)
    return model.coef_[0]
```

The linear specification above is quite restrictive as it essentially imposes that the CATE is the same for all types of units x, i.e. there is no effect heterogeneity. This is a harsh assumption. One way to relax this is to consider a linear specification that constructs interactive features between the treatment D and the covariates X, i.e.,

$$\mathbb{E}[Y \mid D, X] = \beta_0 + \beta^\top \phi(X) + D \cdot (\alpha + \theta^\top \psi(X)),$$

where $\psi(X)$ is another set of hard-coded engineered features from the raw covariates X. This specification allows for some treatment effect heterogeneity.



Then, applying the g-formula, we get

$$\mathrm{CATE}(X) = \alpha + \theta^\top \psi(X), \qquad \mathrm{ATE} = \mathbb{E}[\mathrm{CATE}(X)] = \alpha + \theta^\top \mathbb{E}[\psi(X)].$$

Listing 6: Linear Regression with Interactions (Heterogeneous Treatment Effects)

```python
def ate_linear_with_interactions(Y, D, X, phi, psi):
    """Estimate ATE via linear regression with treatment interactions.

    Assumes: E[Y|D,X] = beta_0 + beta'*phi(X) + D*(alpha + theta'*psi(X))
    Under this model, CATE(x) = alpha + theta'*psi(x).

    Args:
        Y (np.array): Array of observed outcomes.
        D (np.array): Array of treatment indicators (0 or 1).
        X (np.array): Raw covariate matrix.
        phi: Featurizer function phi(X) -> baseline features.
        psi: Featurizer function psi(X) -> effect modifier features.

    Returns:
        tuple: (ATE estimate, CATE function).
    """
    from sklearn.linear_model import LinearRegression

    phi_X = phi(X)
    psi_X = psi(X)

    # Build design matrix: [phi(X), D, D*psi(X)]
    D_col = D.reshape(-1, 1)
    features = np.column_stack([phi_X, D_col, D_col * psi_X])
    model = LinearRegression().fit(features, Y)

    # Extract alpha (coef on D) and theta (coefs on D*psi(X))
    n_phi = phi_X.shape[1]
    alpha = model.coef_[n_phi]
    theta = model.coef_[n_phi + 1:]

    # ATE = alpha + theta' * E[psi(X)]
    ate = alpha + theta @ np.mean(psi_X, axis=0)

    # CATE function
    cate_fn = lambda x: alpha + psi(x) @ theta

    return ate, cate_fn
```

8.3 Why naive stratification does not scale

If X is discrete with a small number of values, we can stratify exactly:

$$\mathrm{ATE} = \sum_x \Big(\mathbb{E}[Y \mid D = 1, X = x] - \mathbb{E}[Y \mid D = 0, X = x]\Big)\mathbb{P}(X = x).$$

But with continuous covariates, we must bin/discretize, which creates a bias–variance tradeoff, and with many covariates the number of strata explodes (curse of dimensionality). This motivates flexible (ML-based) modeling for estimating $m_d(x)$.

Summary

- Conditional ignorability + overlap identify counterfactual conditional means and imply the ATE g-formula:
$$\text{ATE} = \mathbb{E}[\mathbb{E}[Y \mid D = 1, X] - \mathbb{E}[Y \mid D = 0, X]]\,.$$
- One-sided versions identify ATT:
$$\text{ATT} = \mathbb{E}[Y - \mathbb{E}[Y \mid D = 0, X] \mid D = 1]\Big].$$
- In observational studies, ignorability becomes "no unmeasured confounding" (not testable from data alone).
- Post-treatment adjustment can bias estimates (mediators/colliders; birth-weight paradox).
- Outcome regression plug-in estimators operationalize the g-formula; naive stratification does not scale in high dimensions.