Matching

EC 607, Set 7

Edward Rubin
Spring 2020

Prologue

Schedule

Last time

- The conditional independence assumption: $(\mathbf{Y}_{0i}, \mathbf{Y}_{1i}) \perp\!\!\!\perp \mathbf{D}_i \big| \mathbf{X}_i$
- Omitted variable bias
- Good *vs.* bad controls

Today

- Return first round of project proposals.
- Matching estimators (*MHE* 3.2 and Cameron and Trivedi 25.4).

Upcoming

- Admin: Assignment(s)
- No midterm
- Next round of the project proposal

Matching

Matching

The gist

Remember the **conditional independence assumption**[†] in a setting—*i.e.*, treatment is as-good-as random conditional on a known set of covariates?

Matching estimators take us at our word.

If we really believe $(\text{Y}_{1i},\ \text{Y}_{0i}) \perp\!\!\!\perp \text{D}_i | \text{X}_i$, then we can just calculate a bunch of treatment effects conditional on X_i, *i.e.*,

$$\tau(x) = E[\text{Y}_{1i} - \text{Y}_{0i} \mid \text{X}_i = x]$$

The idea: Estimate a treatment effect only using observations with (nearly?) identical values of X_i. The CIA buys us causality within these groups.

[†] AKA "selection on observables"

Matching

Goals

Let's return to **the fundamental problem of causal inference** for a moment.

1. We want/need to know $\tau_i = \mathbf{Y}_{1i} - \mathbf{Y}_{0i}$.
2. We cannot simultaneously observe *both* $\mathbf{Y}_{1i}$ *and* $\mathbf{Y}_{0i}$.

Most empirical strategies boil to strategies to estimate $\mathbf{Y}_{0i}$ for treated individuals—the unobservable counterfactual for the treatment group.

Matching is no different.

We match untreated observations to treated observations using $\mathbf{X}_i$, *i.e.*, calculate a $\widehat{\mathbf{Y}_{0i}}$ for each $\mathbf{Y}_{1i}$, based upon "matched" untreated individuals.

Matching

More formally

We want to construct a counterfactual for each individual with $\mathrm{D}_i = 1$.

The counterfactual for i should only use individuals that match X_i.

Let there be N_T treated individuals and N_C control individuals. We want

- N_T sets of weights
- with N_C weights in each set: $w_i(j)$ $(i = 1, \ldots, N_T; j = 1, \ldots, N_C)$

Assume $\sum_j w_i(j) = 1$. Our estimate for the counterfactual of treated i is

$$\widehat{\mathrm{Y}_{0i}} = \sum_{j \in (D=0)} w_i(j) \mathrm{Y}_j$$

Matching

More formally

If our estimated counterfactual for treated individual i is

$$\widehat{\text{Y}_{0i}} = \sum_j w_i(j)\text{Y}_j$$

then our estimated treatment effect (for individual i) is

$$\hat{\tau}_i = \text{Y}_{1i} - \widehat{\text{Y}_{0i}} = \text{Y}_{1i} - \sum_j w_i(j)\text{Y}_j$$

$\therefore$ a generic matching estimator for the treatment effect on the treated is

$$\hat{\tau}_M = \frac{1}{N_T} \sum_{i \in (\text{D}=1)} \left(\text{Y}_{1i} - \widehat{\text{Y}_{0i}} \right) = \frac{1}{N_T} \sum_{i \in (\text{D}=1)} \left(\text{Y}_{1i} - \sum_{j \in (D=0)} w_i(j)\text{Y}_j \right)$$

Matching

Weight for it[†]

So all we need is those weights and we're done.[††]

Q Where does one find these handy weights?

A You've got options, but you need to choose carefully/responsibly.

E.g., if $w_i(j) = \frac{1}{N_C}$ for all (i, j), then we're back to a difference in means.
This weighting doesn't abide by our conditional independence assumption.

The plan Choose weights $w_i(j)$ that indicate ***how close*** $\mathbf{X}_j$ is to $\mathbf{X}_i$.

[†] 🤦 [††] Plus an interesting, policy-relevant setting with valid conditional independence. And data.

Matching

Proximity

Our weights $w_i(j)$ should be a measure of ***how close*** $\mathbf{X}_j$ is to $\mathbf{X}_i$.

If $\mathbf{X}$ is **discrete**, then we can consider equality, *i.e.*, $w_i(j) = \mathbb{I}(\mathbf{X}_i = \mathbf{X}_j)$, scaling as necessary to get $\sum_j w_i(j) = 1$.

Matching

Proximity

Our weights $w_i(j)$ should be a measure of ***how close*** $\mathbf{X}_j$ is to $\mathbf{X}_i$.

If $\mathbf{X}$ is **continuous**, then we need *proximity* rather than *equality*.

Nearest-neighbor matching chooses the single closest control observation using the Euclidean distance between $\mathbf{X}_i$ and $\mathbf{X}_j$, *i.e.*,

$$\mathbf{d}_{i,j} = \left(\mathbf{X}_i - \mathbf{X}_j\right)' \left(\mathbf{X}_i - \mathbf{X}_j\right)$$

- $\hat{\tau}_i = \mathbf{Y}_{1i} - \mathbf{Y}_{0j}^i$, where $\mathbf{Y}_{0j}^i$ is i's nearest neighbor in the control group.
- **Estimator:** $\hat{\tau}_M = \frac{1}{N_T} \sum_i \hat{\tau}_i$
- Produces causal estimates if CIA is valid *and* we have sufficient overlap.
- Suffers from arbitrary choices of units.

Matching

Proximity

Our weights $w_i(j)$ should be a measure of ***how close*** $\mathbf{X}_j$ is to $\mathbf{X}_i$.

If $\mathbf{X}$ is **continuous**, then we need *proximity* rather than *equality*.

Nearest-neighbor matching with Mahalanobis distance chooses the single closest control using Mahalanobis distance between $\mathbf{X}_i$ and $\mathbf{X}_j$, *i.e.*,

$$\mathbf{d}_{i,j} = (\mathbf{X}_i - \mathbf{X}_j)' \Sigma_X^{-1} (\mathbf{X}_i - \mathbf{X}_j)$$

where Σ_X^{-1} is the covariance matrix of $\mathbf{X}$.

- **Estimator:** $\hat{\tau}_M = \frac{1}{N_T} \sum_i \hat{\tau}_i$ where $\left(\hat{\tau}_i = \mathbf{Y}_{1i} - \mathbf{Y}_{0j}^i \right)$
- Produces causal estimates if CIA is valid *and* we have sufficient overlap.
- Does not suffer from arbitrary choices of units.

Matching

More neighbors?

Why limit ourselves to a **single** "best" match?

If we're going to let a function/algorithm choose the *nearest* match, can't we also let the function/algorithm choose *how many* matches?

Furthermore, if $N_C \gg N_T$, it we're throwing away *a lot* of information.

We could instead use this information and be more efficient.

Matching

More neighbors!

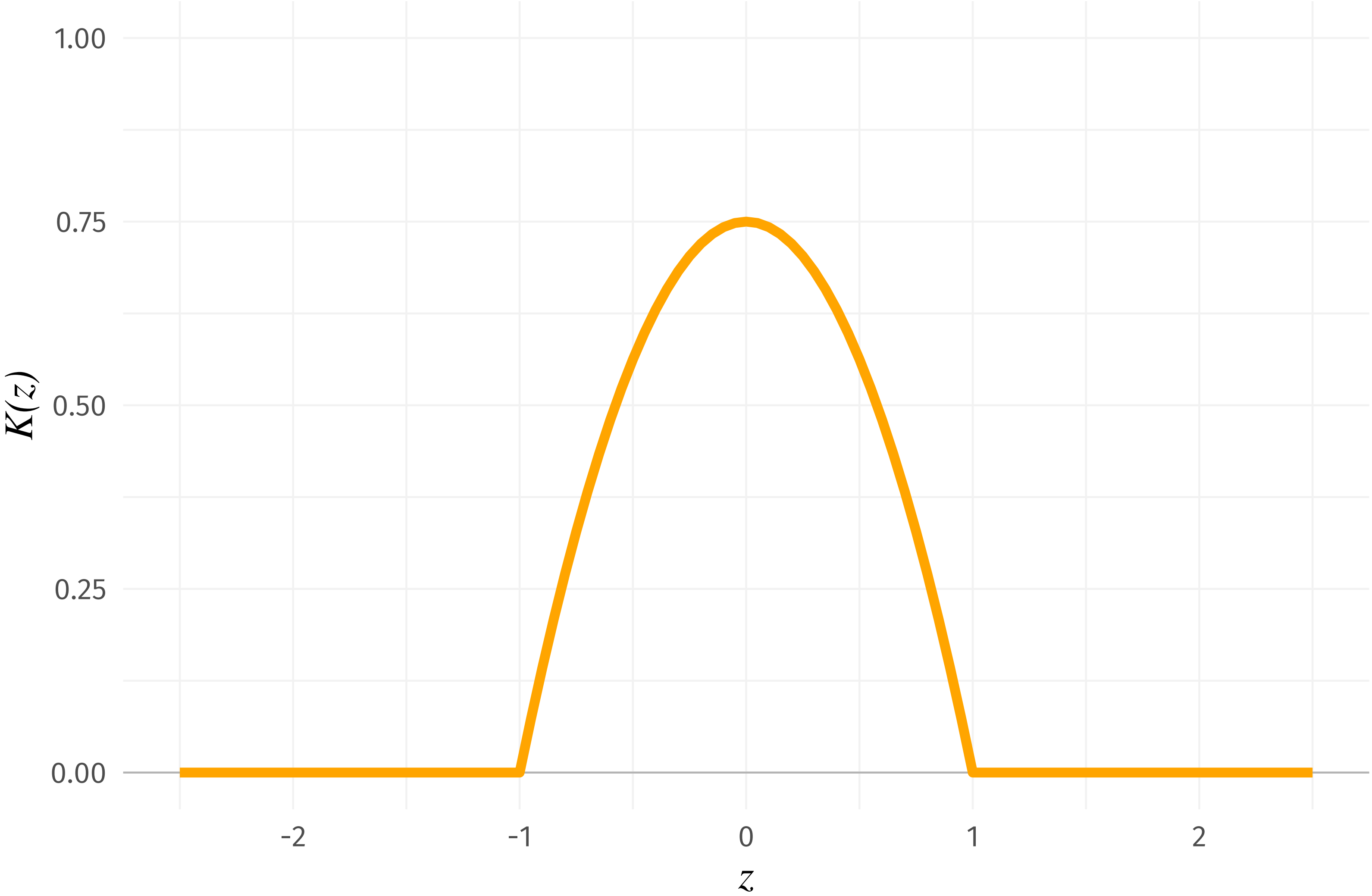
Kernel matching gives positive weight to all control observations within some **bandwidth** h, with higher weight for closer matches determined by some **kernel function** $K(\cdot)$,

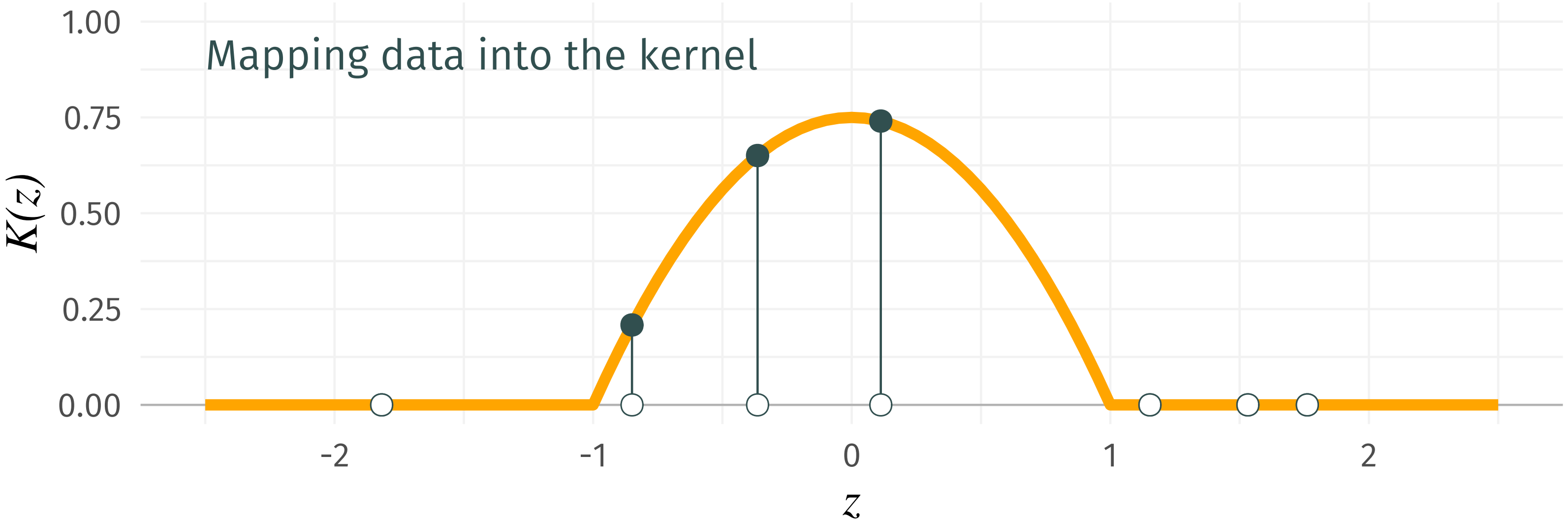
$$w_i(j) = \frac{K\left(\frac{\mathrm{X}_j - \mathrm{X}_i}{h}\right)}{\sum_{j \in (D=0)} K\left(\frac{\mathrm{X}_j - \mathrm{X}_i}{h}\right)}$$

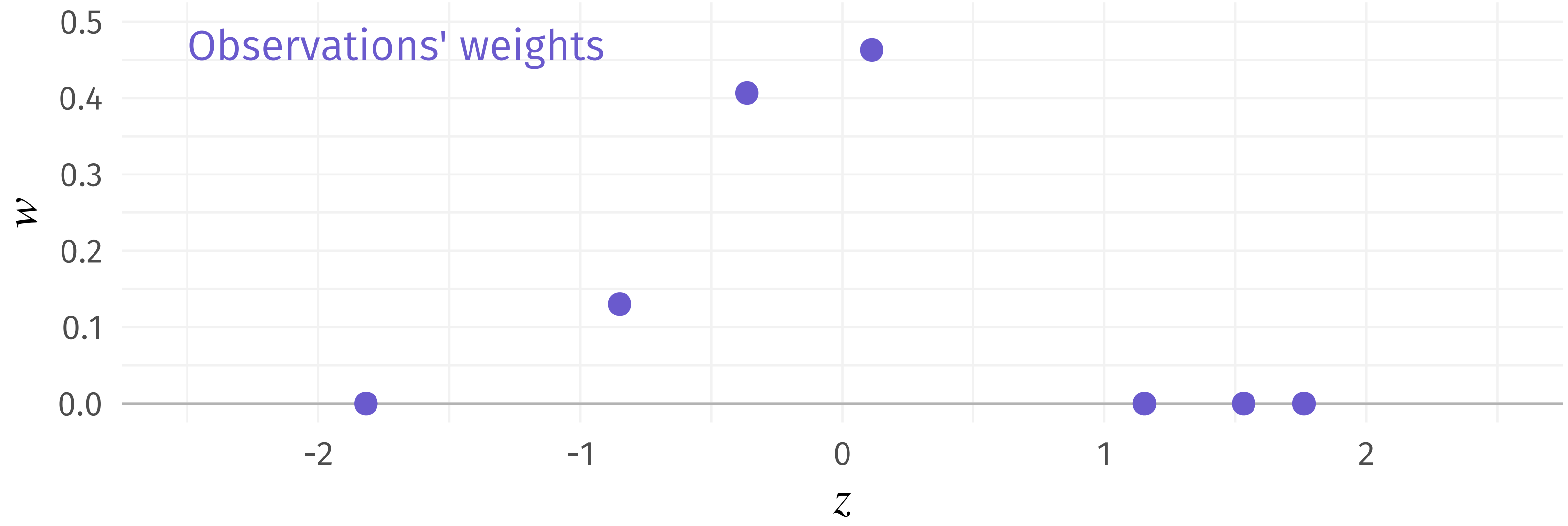
Example The *Epanechnikov kernel* is defined as

$$K(z) = \frac{3}{4}\left(1 - z^2\right) \times \mathbb{I}(|z| < 1)$$

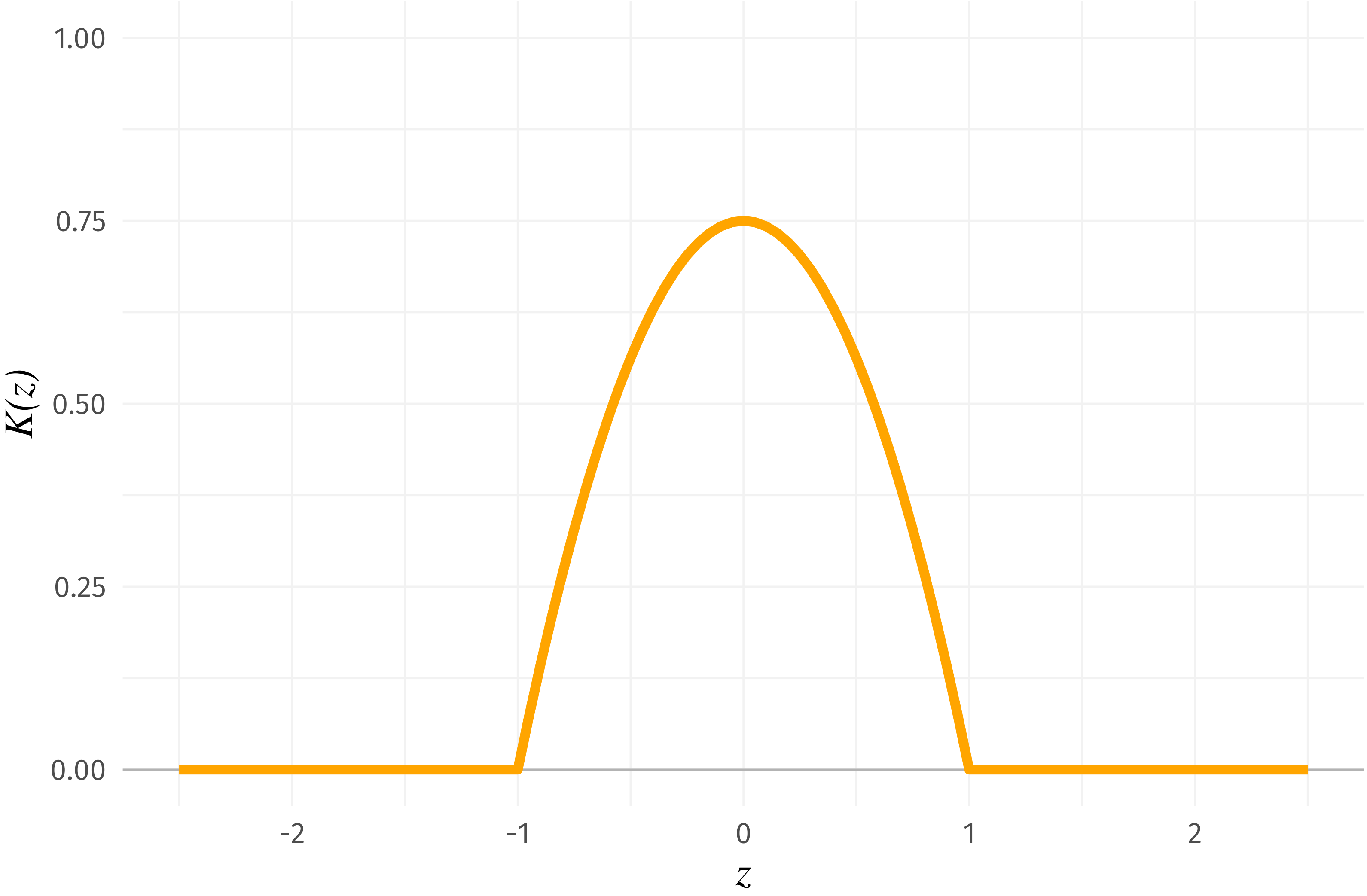
The Epanechnikov kernel $K(z) = \frac{3}{4}\left(1 - z^2\right) \times \mathbb{I}(|z| < 1)$



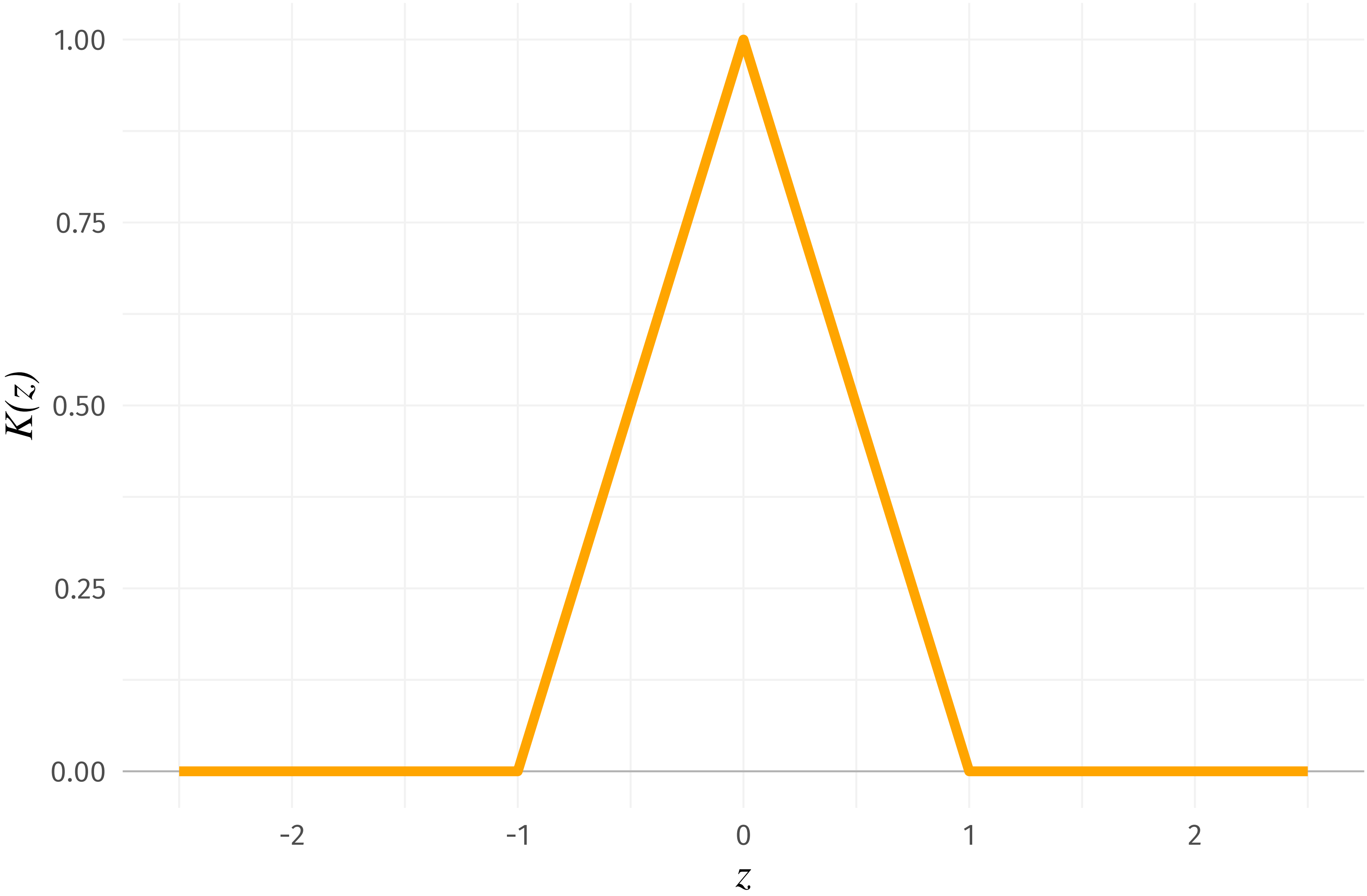

Mapping data into the kernel
$K(z)$
1.00
0.75
0.50
0.25
0.00
-2
-1
0
1
2
z


Observations' weights
w
0.5
0.4
0.3
0.2
0.1
0.0
-2
-1
0
1
2
z

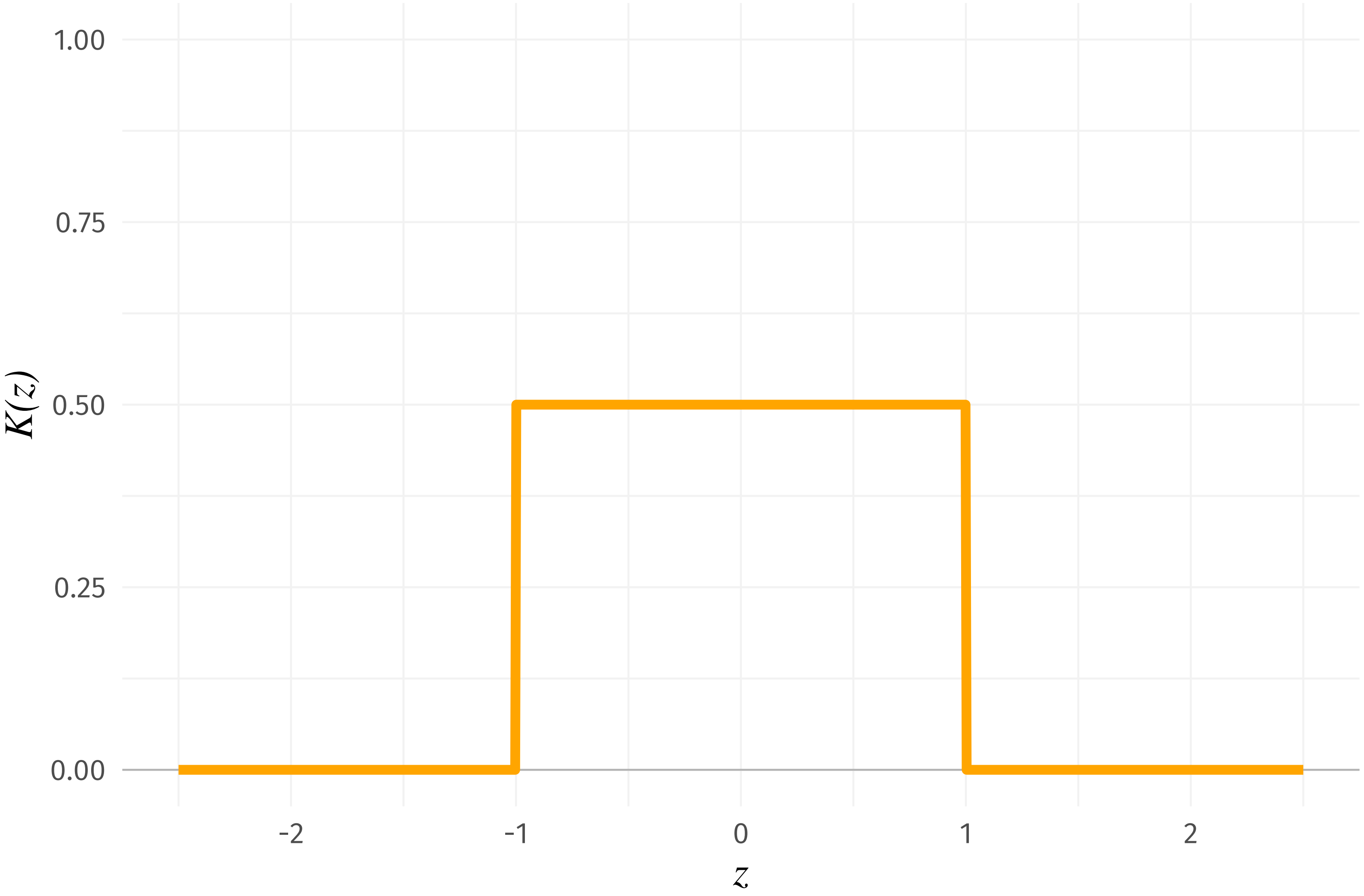
The Epanechnikov kernel $K(z) = \frac{3}{4}\left(1 - z^2\right) \times \mathbb{I}(|z| < 1)$



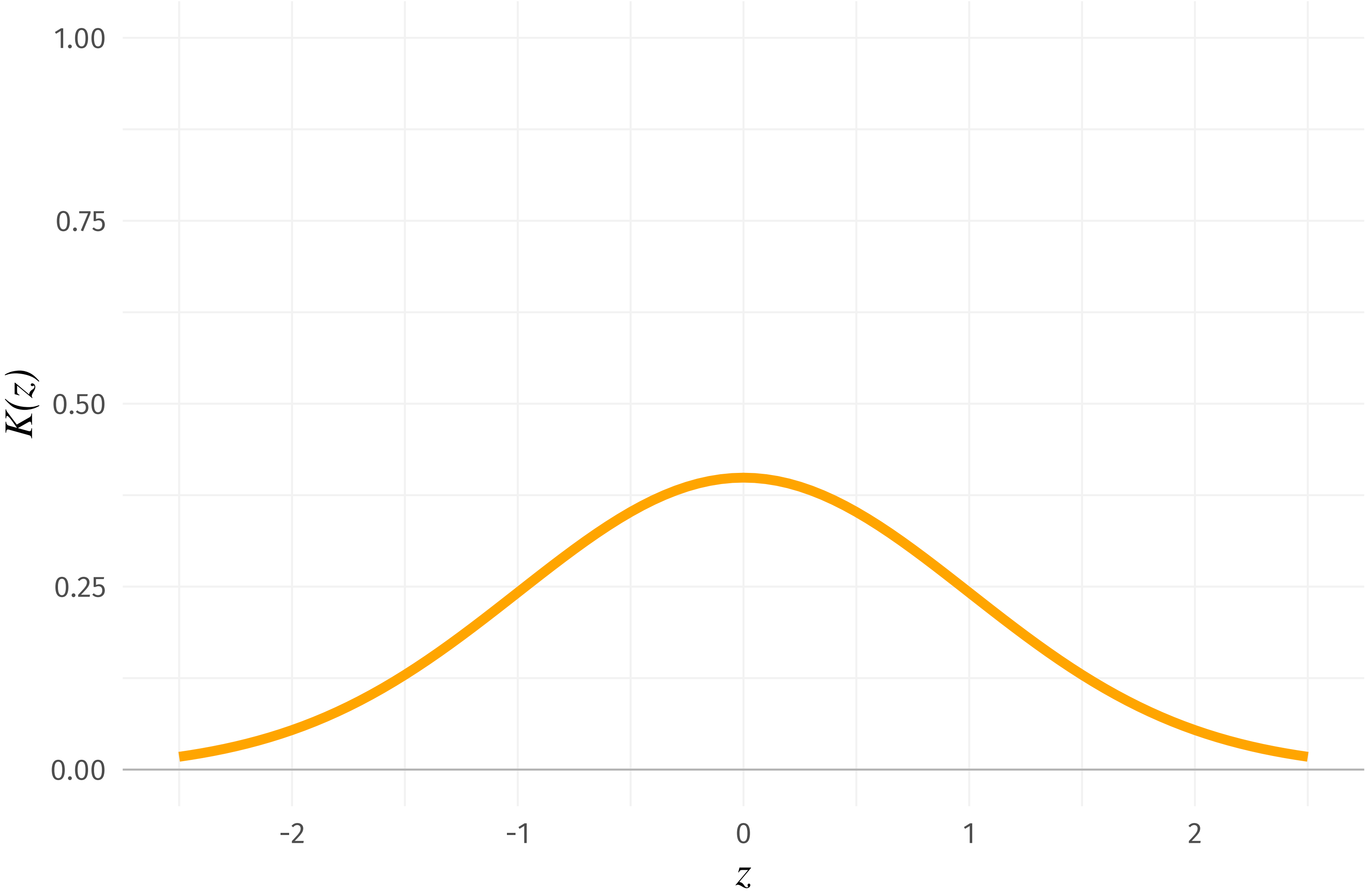
The Triangle kernel $K(z) = (1 - |z|) \times \mathbb{I}(|z| < 1)$



The Uniform kernel $K(z) = \frac{1}{2} \times \mathbb{I}(|z| < 1)$



The Gaussian kernel $K(z) = (2\pi)^{-1/2} \exp\left(-z^2/2\right)$



Kernels

Aside

Kernel functions are good for more than just matching.

You will most commonly see/use them smoothing out densities—providing a smooth, moving-window average.

E.g., `R`'s (`ggplot2`'s) smooth, density-plotting function `geom_density()`.

`geom_density()` defaults to `kernel = "gaussian"`, but you can specify many other kernel functions (including `"epanechnikov"`).

You can also change the `bandwidth` argument. The default is a bandwidth-choosing function called `bw.nrd0()`.

Matching

Adding neighbors

As we add more neighbors—either moving from 1 to $n > 1$ or increasing our bandwidth—we potentially increase the efficiency of our estimator.

We need to **be careful not to add *too many* controls** for each treated i.

CIA requires that we're actually conditioning on the observables—it does not allow us to take a simple average across all control observations.

Matching

The curse of dimensionality[†]

It turns out kernel- and bandwidth-selection are not our biggest enemies.

As the dimension of $\mathbf{X}$ expands (matching on more variables), it becomes **harder and harder to find a nice, close control** for each treated unit.

We need a way to shrink the dimensionality of $\mathbf{X}$.

[†] I'm not sure if this is a title for Harry Potter or Indiana Jones... crossover anyone?

Propensity-score methods

Propensity-score methods

Setup

Let's begin with two assumptions—one old and one new.

1. **Conditional independence:** $\left(\text{Y}_{0i},\ \text{Y}_{1i}\right) \perp\!\!\!\perp \text{D}_i | \text{X}_i$

2. **Overlap:** $0 < \Pr\!\left(\text{D}_i = 1 \mid \text{X}_i\right) < 1$

We can estimate an average treatment effect by conditioning on X_i.

However, overlap may fail if the dimensions of X are large and N is finite.

Propensity scores provide a solution to this mess.

Propensity-score methods

The magic

It turns out that if $(\text{Y}_{0i}, \text{Y}_{1i}) \perp\!\!\!\perp \text{D}_i | \text{X}_i$, then we actually only need to match/condition on $p(\text{X}_i) = E[\text{D}_i | \text{X}_i]$.

$p(\text{X}_i)$ is the propensity score, the probability of treatment given X_i.

Propensity-score theorem If $(\text{Y}_{0i}, \text{Y}_{1i}) \perp\!\!\!\perp \text{D}_i | \text{X}_i$, then $(\text{Y}_{0i}, \text{Y}_{1i}) \perp\!\!\!\perp \text{D}_i | p(\text{X}_i)$.

This theorem extends our CIA to a one-dimensional score, avoiding the curse of dimensionality.

Propensity-score methods

Theorem If $(\mathrm{Y}_{0i},\, \mathrm{Y}_{1i}) \perp\!\!\!\perp \mathrm{D}_i | \mathrm{X}_i$, then $(\mathrm{Y}_{0i},\, \mathrm{Y}_{1i}) \perp\!\!\!\perp \mathrm{D}_i | p(\mathrm{X}_i)$.

Proof

$$\Pr\left[\mathrm{D}_i = 1 \middle| \mathrm{Y}_{0i},\, \mathrm{Y}_{1i},\, p(\mathrm{X}_i)\right]$$

$$= E\left[\mathrm{D}_i \middle| \mathrm{Y}_{0i},\, \mathrm{Y}_{1i},\, p(\mathrm{X}_i)\right]$$

$$= E\left[E\left(\mathrm{D}_i \middle| \mathrm{Y}_{0i},\, \mathrm{Y}_{1i},\, p(\mathrm{X}_i),\, \mathrm{X}_i\right) \middle| \mathrm{Y}_{0i},\, \mathrm{Y}_{1i},\, p(\mathrm{X}_i)\right]$$

$$= E\left[E\left(\mathrm{D}_i \middle| \mathrm{Y}_{0i},\, \mathrm{Y}_{1i},\, \mathrm{X}_i\right) \middle| \mathrm{Y}_{0i},\, \mathrm{Y}_{1i},\, p(\mathrm{X}_i)\right]$$

Propensity-score methods

Theorem If $(\mathrm{Y}_{0i},\, \mathrm{Y}_{1i}) \perp\!\!\!\perp \mathrm{D}_i | \mathrm{X}_i$, then $(\mathrm{Y}_{0i},\, \mathrm{Y}_{1i}) \perp\!\!\!\perp \mathrm{D}_i | p(\mathrm{X}_i)$.

Proof

$$\Pr\left[\mathrm{D}_i = 1 \middle| \mathrm{Y}_{0i},\, \mathrm{Y}_{1i},\, p(\mathrm{X}_i)\right] = \cdots = E\left[E\left(\mathrm{D}_i \middle| \mathrm{Y}_{0i},\, \mathrm{Y}_{1i},\, \mathrm{X}_i \right) \middle| \mathrm{Y}_{0i},\, \mathrm{Y}_{1i},\, p(\mathrm{X}_i) \right]$$

$$= E\left[E\left(\mathrm{D}_i \middle| \mathrm{X}_i \right) \middle| \mathrm{Y}_{0i},\, \mathrm{Y}_{1i},\, p(\mathrm{X}_i) \right]$$

$$= E\left[p(\mathrm{X}_i) \middle| \mathrm{Y}_{0i},\, \mathrm{Y}_{1i},\, p(\mathrm{X}_i) \right]$$

$$= p(\mathrm{X}_i)$$

$$\therefore\ (\mathrm{Y}_{0i},\, \mathrm{Y}_{1i}) \perp\!\!\!\perp \mathrm{D}_i | \mathrm{X}_i \implies (\mathrm{Y}_{0i},\, \mathrm{Y}_{1i}) \perp\!\!\!\perp \mathrm{D}_i | p(\mathrm{X}_i) \quad ✔$$

Propensity-score methods

Intuition

Q What's going on here?

$\mathbf{X}_i$ carries way more information than $p(\mathbf{X}_i)$, so how can we still get conditional independence of treatment by only conditioning on $p(\mathbf{X}_i)$?

$\mathtt{A}_1$ Conditional independence of treatment isn't about extracting all of the information possible from $\mathbf{X}_i$. We actually only care about creating a situation in which $\mathbf{D}_i|$something is independent of $(\mathbf{Y}_{0i},\ \mathbf{Y}_{1i})$.

$\mathtt{A}_2$ Back to our main concern: **selection bias**. People select into treatment. If $\mathbf{X}$ says two people were equally likely to be treated, and if $\mathbf{X}_i$ explains all of selection (CIA), then there cannot be selection between these two people.

Propensity-score methods

Estimation

So where do propensity scores come from?

We estimate them—and there are a lot of ways to do that.

1. Flexible (*i.e.*, interactions) logit specification
2. Kernel regression (remember kernel functions?)
3. Many others—machine learning, series-logit estimator, *etc.*

Q Can we just use plain OLS (linear probability model)?

A Sort of. Think about FWL. This route is going to be the same as a regression conditioning on $\mathbf{X}_i$.

Propensity-score methods

Estimation

From *MHE* (p. 83)

Question

> A big question here is how to best model and estimate $p(\mathbf{X}_i)$...

Answer

> The answer to this is inherently application-specific. A growing empirical literature suggests that a logit model for the propensity score with a few polynomial terms in continuous covariates works well in practice...

Propensity-score methods

Application

So you have some estimated propensity scores $\hat{p}(\mathrm{X}_i)$. What next?

Option 1 Conditioning via regression

Option 1a Use a **regression to condition** on $p(\mathrm{X}_i)$, *i.e.*,

$$\mathrm{Y}_i = \alpha + \delta \mathrm{D}_i + \beta p(\mathrm{X}_i) + u_i \tag{1a}$$

Option 1b If we think treatment effects are heterogeneous and may covary with $\mathbf{X}$, then we might want to also **interact** treatment with $p(\mathrm{X}_i)$, *i.e.*,

$$\mathrm{Y}_i = \alpha + \delta_1 \mathrm{D}_i + \delta_2 \mathrm{D}_i p(\mathrm{X}_i) + \beta p(\mathrm{X}_i) + u_i \tag{1b}$$

Propensity-score methods

Heterogeneity with regression

Let's think a bit more about heterogeneous treatment effects in this setting.

$$\begin{aligned} \text{Y}_{0i} &= \alpha + \beta \text{X}_i + u_i \\ \text{Y}_{1i} &= \text{Y}_{0i} + \delta_1 + \delta_2 \text{X}_i \end{aligned}$$

i.e., the treatment effect depends upon X_i.

$$\begin{aligned} \text{Y}_i &= \text{D}_i \text{Y}_{1i} + \left(1 - \text{D}_i \right) \text{Y}_{0i} \\ &= \text{D}_i \bigg(\text{Y}_{0i} + \delta_1 + \delta_2 \text{X}_i \bigg) + \left(1 - \text{D}_i \right) \text{Y}_{0i} \\ &= \text{Y}_{0i} + \delta_1 \text{D}_i + \delta_2 \text{D}_i \text{X}_i \\ &= \alpha + \delta_1 \text{D}_i + \delta_2 \text{D}_i \text{X}_i + \beta \text{X}_i + u_i \end{aligned}$$

Propensity-score methods

Heterogeneity

This final equation

$$\mathrm{Y}_i = \alpha + \delta_1 \mathrm{D}_i + \delta_2 \mathrm{D}_i \mathrm{X}_i + \beta \mathrm{X}_i + u_i$$

suggests that we want $p(\mathrm{X}_i)$ *and* $\mathrm{D}_i p(\mathrm{X}_i)$, *i.e.*,

$$\mathrm{Y}_i = \alpha + \delta_1 \mathrm{D}_i + \delta_2 \mathrm{D}_i p(\mathrm{X}_i) + \beta p(\mathrm{X}_i) + u_i \tag{1b}$$

which yields

1. a **group-specific treatment effect** $\delta_1 + \delta_2 p(\mathrm{X}_i)$ for each X_i
2. an **average treatment effect** $\delta_1 + \delta_2 \overline{p}(\mathrm{X}_i)$

Propensity-score methods

More flexibility

We motivated propensity scores with a desire to reduce dimensionality and estimate/choose/assume fewer parameters.

Adding $p(\text{X}_i)$ and $\text{D}_i p(\text{X}_i)$ as covariates in a linear regression doesn't quite exhaust our potential for flexible/nonparametric estimation.

Propensity-score methods

Blocking

Option 2 Block (stratify) on propensity scores.

1. Divide the range of $\hat{p}(\mathrm{X}_i)$ into K blocks (*e.g.*, 0.05-wide blocks).

2. Place each observation into a block via its $\hat{p}(\mathrm{X}_i)$.

3. Calculate $\hat{\tau}_k$ for each block via difference in means.

4. Average the $\hat{\tau}_k$ using their shares of the sample, *i.e.*,

$$\hat{\tau}_{\text{Block}} = \sum_{k=1}^{K} \hat{\tau}_k \frac{N_{1k} + N_{0k}}{N}$$

Note Blocking is similar to NN/kernel matching using $p(\mathrm{X}_i)$ as distance.

Propensity-score methods

Choosing blocks

Blocking on propensity scores requires defining defining blocks.

One common route involves some iteration.

1. **Choose blocks**.

2. Check the **balance of the covariates** within each block.[†]

   - If covariates are not balanced, then split your blocks and repeat.

   - If covariates are balanced, then stop.

[†] Keep multiple-hypothesis testing in mind. With many covariates and many blocks, you are bound to find statistically significant relationships—even if you are balanced in truth.

Propensity-score methods

Overlap

Blocking emphasizes our overlap assumption, *i.e.*, $\mathbf{0} < \mathbf{Pr}(\mathrm{D}_i | \mathrm{X}_i) < \mathbf{1}$.

If a block contains zero treated/control units, we cannot calculate $\hat{\tau}_k$.

Caution Logit can hide violations—it forces $\mathbf{0} < \hat{p}(\mathrm{X}_i) < \mathbf{1}$.

Common practice Empirically enforce overlap:

- Drop control units with $\hat{p}(\mathrm{X}_i)$ below the minimum propensity score in the treatment group.
- Drop treated units with $\hat{p}(\mathrm{X}_i)$ above the maximum propensity score in the control group.

Propensity-score methods

Weighting

Option 3 Weight observations by the inverse propensity score.

Q How does weighting by $1/\hat{p}(\mathrm{X}_i)$ make sense?

A Consider our old (likely biased) friend the difference in means, *i.e.*,

$$\hat{\tau}_{\text{Diff}} = \overline{\mathrm{Y}}_{\mathrm{T}} - \overline{\mathrm{Y}}_{\mathrm{C}} = \frac{\sum_i \mathrm{D}_i \mathrm{Y}_i}{\sum_i \mathrm{D}_i} - \frac{\sum_i \left(1 - \mathrm{D}_i\right) \mathrm{Y}_i}{\sum_i \left(1 - \mathrm{D}_i\right)}$$

which we've discussed is biased due to selection into treatment, *i.e.*,

$$E[\mathrm{Y}_{0i} | \mathrm{D}_i = 1] \neq E[\mathrm{Y}_{0i}]$$

Propensity-score methods

Weighting, justified

Suppose we know $p(\mathrm{X}_i)$ and we weight each **treated** individual by $1/p(\mathrm{X}_i)$

$$E\left[\frac{\mathrm{D}_i\mathrm{Y}_i}{p(\mathrm{X}_i)}\right] = E\left[\frac{\mathrm{D}_i\left(\mathrm{D}_i\mathrm{Y}_{1i} + (1-\mathrm{D}_i)\mathrm{Y}_{0i}\right)}{p(\mathrm{X}_i)}\right] = E\left[\frac{\mathrm{D}_i\mathrm{Y}_{1i}}{p(\mathrm{X}_i)}\right]$$

$$= E\left(E\left[\frac{\mathrm{D}_i\mathrm{Y}_{1i}}{p(\mathrm{X}_i)}\,\middle|\,\mathrm{X}_i\right]\right) = E\left(\frac{E[\mathrm{D}_i \mid \mathrm{X}_i]\,E[\mathrm{Y}_{1i} \mid \mathrm{X}_i]}{p(\mathrm{X}_i)}\right)$$

$$= E\left(\frac{p(\mathrm{X}_i)\,E[\mathrm{Y}_{1i} \mid \mathrm{X}_i]}{p(\mathrm{X}_i)}\right) = E\Big(E[\mathrm{Y}_{1i} \mid \mathrm{X}_i]\Big) = E[\mathrm{Y}_{1i}]$$

Similarly, weighting **control** individuals by $1/(1-p(\mathrm{X}_i))$ yields

$$E\left[\frac{(1-\mathrm{D}_i)\mathrm{Y}_i}{1-p(\mathrm{X}_i)}\right] = E[\mathrm{Y}_{0i}]$$

Propensity-score methods

Weighting: The estimator

Thus, we can estimate an unbiased treatment effect via

$$\hat{\tau}_{p\text{Weight}} = \frac{1}{N} \sum_{i=1}^{N} \left[\frac{\text{D}_i \text{Y}_i}{p(\text{X}_i)} - \frac{(1 - \text{D}_i \text{Y}_i)}{1 - p(\text{X}_i)} \right]$$

Intuition We're trying to overcome selection bias, *i.e.*, treated individuals were more likely to be treated as a function of $\mathbf{X}_i$—producing higher $p(\mathbf{X}_i)$.

We want to get back to *as-good-as random* variation in treatment.

So we upweight (**1**) **treated** individuals with low $p(\text{X}_i)$ and (**2**) **control** observations with high $p(\text{X}_i)$.

Propensity-score methods

Weighting: The example

Suppose for some individual i, $p(\mathbf{X}_i) = 0.80$.

This propensity score says someone with this set of $\mathbf{X}_i$ was four-times more likely to be **treated** than **control**.

Our weights fix this imbalance for each $\mathbf{X}_i$.

- If i is **treated**, then her weight is $1/p(\mathbf{X}_i) = 1/0.80 = 1.25$
- If i is **control**, then her weight is $1/(1 - p(\mathbf{X}_i)) = 1/(1 - 0.80) = 5$

And guess what $5/1.25$ is... 4! This weighting scheme gets us back to equal representation for each set of $\mathbf{X}_i$.

Propensity-score methods

Weighting: Last issue

Practical issue Nothing guarantees $\sum_i \hat{p}(\mathrm{X}_i) = 1$.

Solution Normalize weights by their total sum.

Applying the normalized (and estimated) propensity scores

$$\hat{\tau}_{p\text{Weight}} = \sum_{i=1}^{N} \frac{\dfrac{\mathrm{D}_i \mathrm{Y}_i}{\hat{p}(\mathrm{X}_i)}}{\sum_i \dfrac{\mathrm{D}_i}{\hat{p}(\mathrm{X}_i)}} - \sum_{i=1}^{N} \frac{\dfrac{(1-\mathrm{D}_i)\mathrm{Y}_i}{1-\hat{p}(\mathrm{X}_i)}}{\sum_i \dfrac{(1-\mathrm{D}_i)}{1-\hat{p}(\mathrm{X}_i)}}$$

Hirano, Imbens, and Ridder (2003) suggests this estimator is efficient.

Propensity-score methods

Why choose one?

There's nothing special about weighted averages—regression can weight.

Thus, a **regression-based estimate**

$$\text{Y}_i = \alpha + \text{X}_i\beta + \tau\text{D}_i + u_i$$

with **weights**

$$w_i = \sqrt{\frac{\text{D}_i}{\hat{p}(\text{X}_i)} + \frac{(1 - \text{D}_i)}{1 - \hat{p}(\text{X}_i)}}$$

offers a *doubly robust* property—you have two chances to be right: $p(\text{X}_i)$ or the regression specification.

Propensity-score methods

Why choose one? Part two

An alternative, doubly robust method combines propensity-score blocking with regression.

Step 1 For each block k, we run the regression

$$\mathrm{Y}_i = \alpha_k + \mathrm{X}_i \beta_k + \tau_k \mathrm{D}_i + u_i$$

Step 2 Aggregate block-level treatment-effect estimates

$$\hat{\tau} = \sum_{k=1}^{K} \hat{\tau}_k \frac{N_{1k} + N_{0k}}{N}$$

Propensity-score methods

Major requirements

Don't get (too) caught up in the bells and whistles.

We still have two **major** requirements for any of these methods to work.

1. Is the **conditional-independence assumption** true?

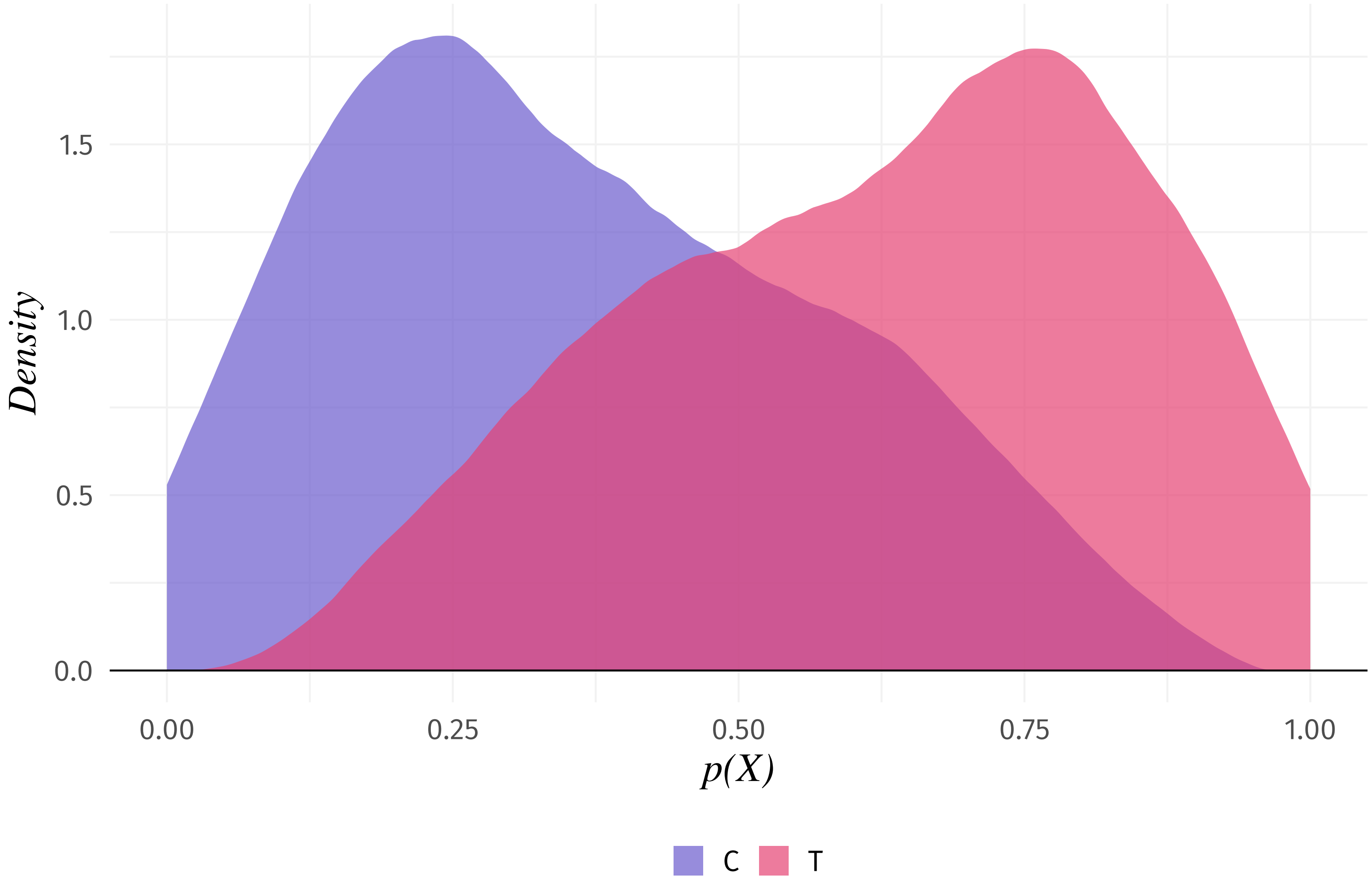
2. Do we have **overlap** between treatment and control units.

We can look for evidence of (**2**) in the data—particularly if we're using propensity-score methods.[†]

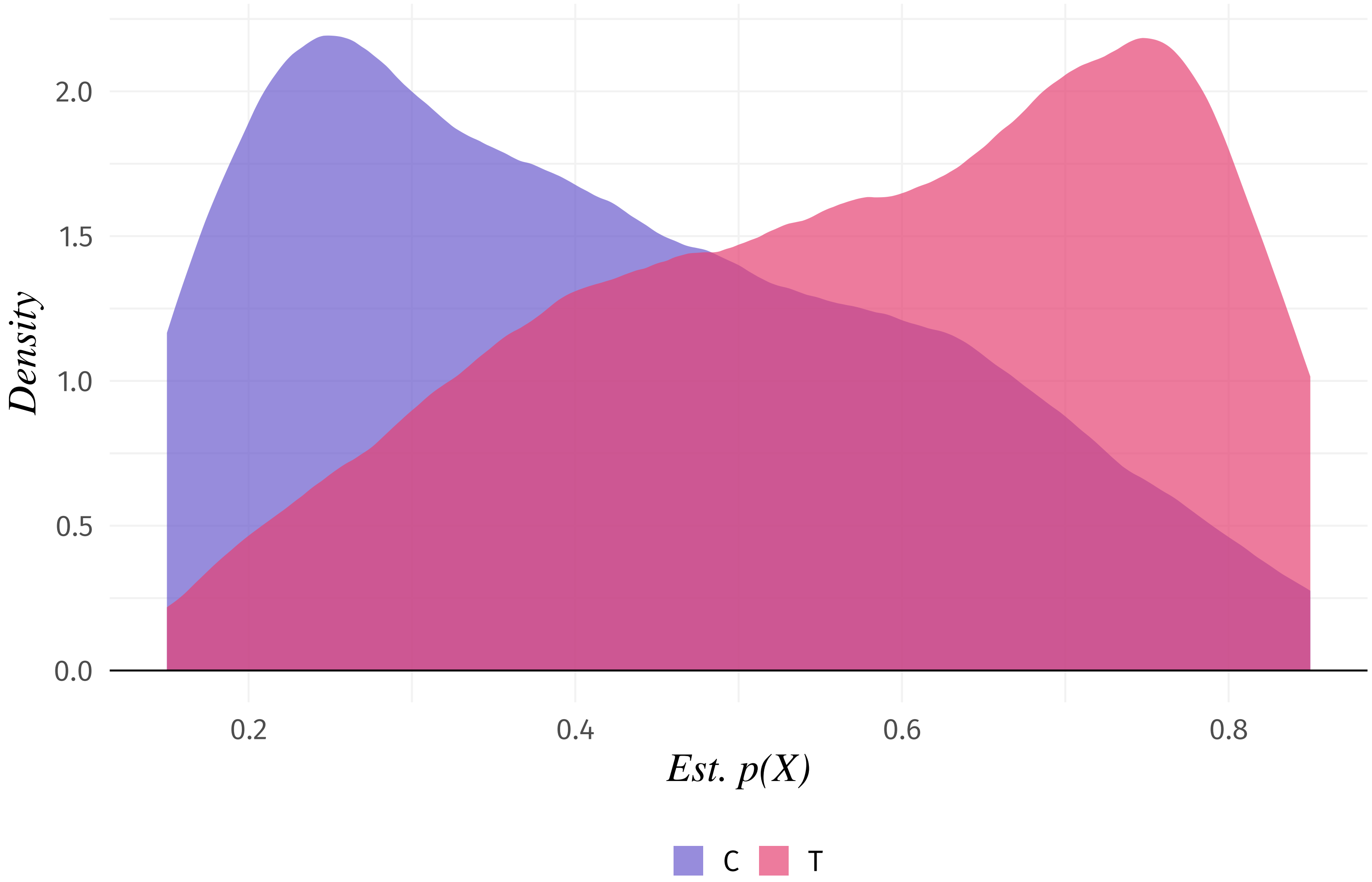
How? Plot the distributions of $p(\mathbf{X}_i)$ for **T** and **C**.

[†] Checking for overlap in $\mathbf{X}$-space, can be tough as the dimensions of $\mathbf{X}$ expand.

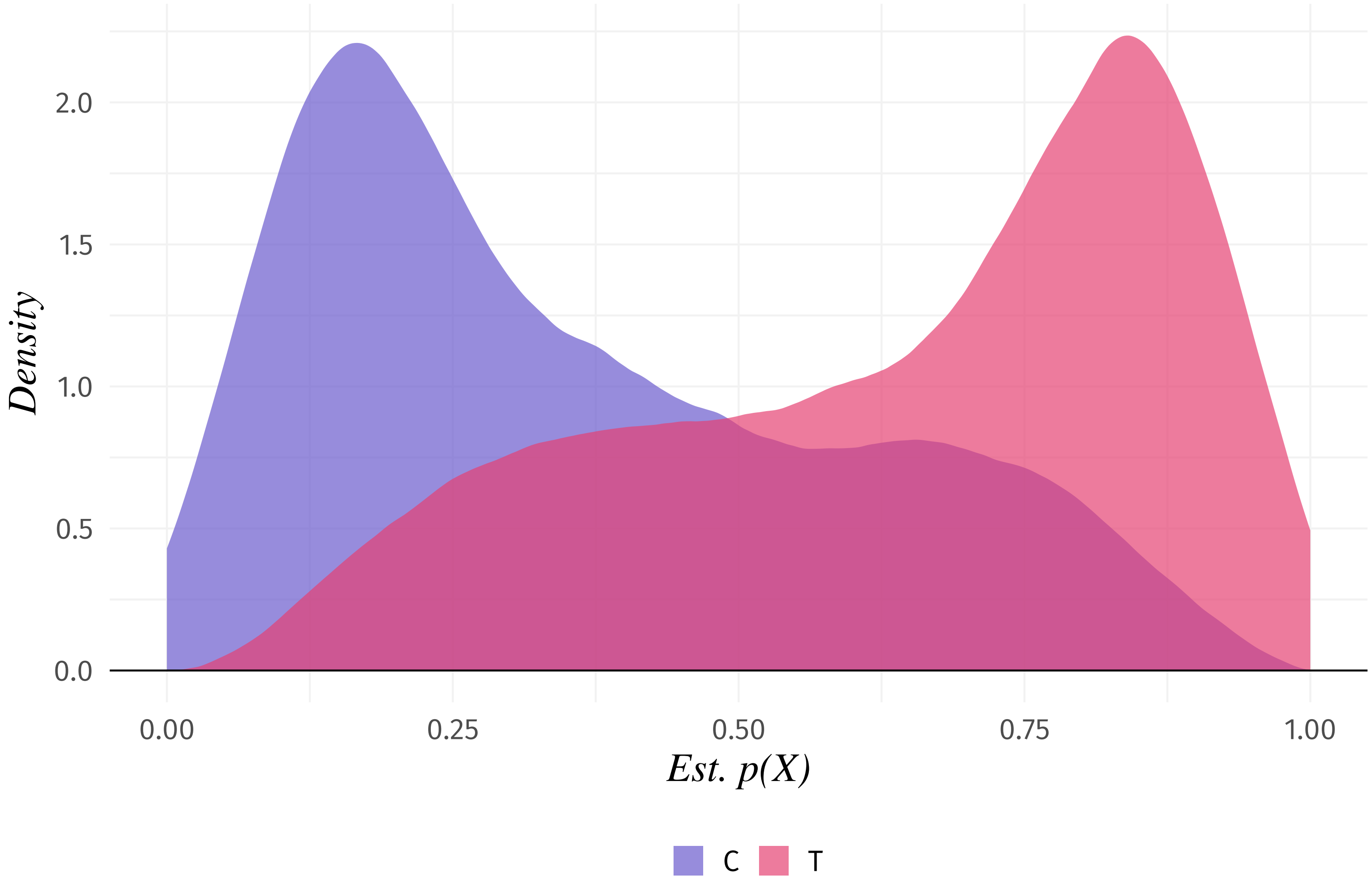
Missing overlap in $p(\mathbf{X}_i)$



Authentic (enforced) overlap in $p(\mathbf{X}_i)$



Logit-based $\hat{p}(\mathbf{X}_i)$ hiding some of the missing overlap in $p(\mathbf{X}_i)$



Overlap in one dimension does not guarantee in two dimensions.

Note Shading denotes **share of treatment:** **white** =0% and **pink**=100%.

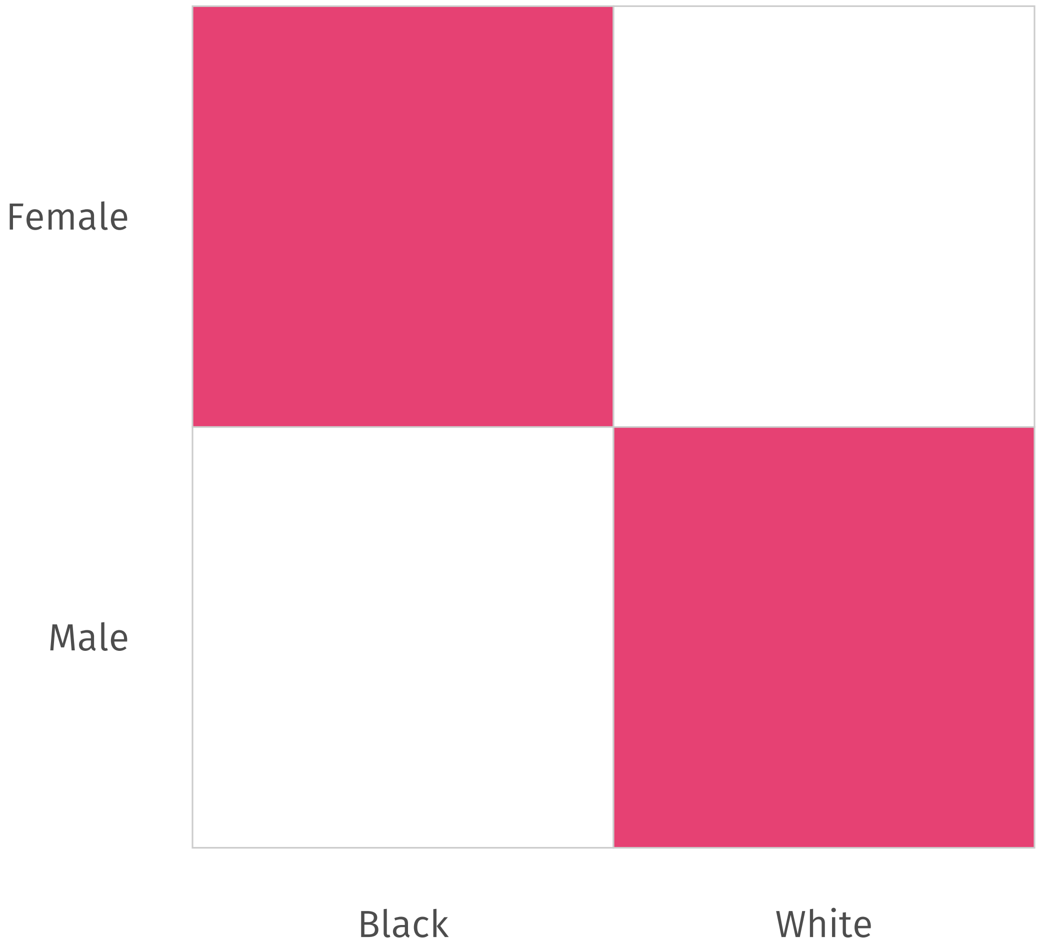


Table of contents

Admin

1. Schedule
2. Follow up

General matching

1. The gist
2. Goals
3. Generic matching
4. Weights
    - Discrete $\mathbf{X}$
    - Nearest neighbor, Euclidean
    - Nearest neighbor, Mahalanobis
    - Kernel matching

Propensity-score methods

1. Setup
2. Propensity-score theorem
    - The magic
    - The proof
    - Intuition
3. Estimation
4. Application
    - Regression
    - Heterogeneity
    - Blocking on $p(\mathbf{X}_i)$
    - Weighting with $p(\mathbf{X}_i)$
    - Doubly robust methods
5. Overlap plots