Day 2: On Weights and Clusters

Peter Hull

Design-Based Regression Inference
Spring 2024

Outline

1. Heterogeneous Treatment Effects

2. Clustered Standard Errors

Whose Treatment Effect is it Anyway?

- On Monday we contrasted design vs. outcome-model strategies in a constant-effect world (i.e. with a causal model of $y_i = \beta x_i + \varepsilon_i$)
  - Of course the real world is messier: more realistic is $y_i = \beta_i x_i + \varepsilon_i$ (or more complicated forms of effect heterogeneity)

Whose Treatment Effect is it Anyway?

- On Monday we contrasted design vs. outcome-model strategies in a constant-effect world (i.e. with a causal model of $y_i = \beta x_i + \varepsilon_i$)
  - Of course the real world is messier: more realistic is $y_i = \beta_i x_i + \varepsilon_i$ (or more complicated forms of effect heterogeneity)
  - Can think about what the regression/IV estimand equals in such models

Whose Treatment Effect is it Anyway?

- On Monday we contrasted design vs. outcome-model strategies in a constant-effect world (i.e. with a causal model of $y_i = \beta x_i + \varepsilon_i$)
  - Of course the real world is messier: more realistic is $y_i = \beta_i x_i + \varepsilon_i$ (or more complicated forms of effect heterogeneity)
  - Can think about what the regression/IV estimand equals in such models
- Today we'll see another difference: how design-based vs. model-based regression/IV weigh together heterogeneous effects
  - Bottom line: design avoids recent concerns over "negative weights"...

Whose Treatment Effect is it Anyway?

- On Monday we contrasted design vs. outcome-model strategies in a constant-effect world (i.e. with a causal model of $y_i = \beta x_i + \varepsilon_i$)
  - Of course the real world is messier: more realistic is $y_i = \beta_i x_i + \varepsilon_i$ (or more complicated forms of effect heterogeneity)
  - Can think about what the regression/IV estimand equals in such models

- Today we'll see another difference: how design-based vs. model-based regression/IV weigh together heterogeneous effects
  - Bottom line: design avoids recent concerns over "negative weights"...
  - ... at least as long as you don't have multiple treatments!

Why So Negative?

- A recent TWFE literature (e.g. de Chaisemartin and D'Haultfoeuille '20; Goodman-Bacon '21; Borusyak et al. '23) shows that some regressions identify $\beta = E[\psi_i \beta_i]/E[\psi_i]$ for possibly negative ψ_i

Why So Negative?

- A recent TWFE literature (e.g. de Chaisemartin and D'Haultfoeuille '20; Goodman-Bacon '21; Borusyak et al. '23) shows that some regressions identify $\beta = E[\psi_i \beta_i]/E[\psi_i]$ for possibly negative ψ_i
  - We'll term these ψ_i "ex-post" weights, for reasons you'll see shortly

Why So Negative?

- A recent TWFE literature (e.g. de Chaisemartin and D'Haultfoeuille '20; Goodman-Bacon '21; Borusyak et al. '23) shows that some regressions identify $\beta = E[\psi_i \beta_i]/E[\psi_i]$ for possibly negative ψ_i
  - We'll term these ψ_i "ex-post" weights, for reasons you'll see shortly

- Why is this a concern? The possibility of *sign reversals*:
  - Even if all β_i are positive, β could wind up negative (or vice versa) if ψ_i and β_i are correlated
  - The literature proposes alternative specifications/procedures that avoid negative weighting or allow for custom-built weights

Why So Negative?

- A recent TWFE literature (e.g. de Chaisemartin and D'Haultfoeuille '20; Goodman-Bacon '21; Borusyak et al. '23) shows that some regressions identify $\beta = E[\psi_i \beta_i]/E[\psi_i]$ for possibly negative ψ_i
  - We'll term these ψ_i "ex-post" weights, for reasons you'll see shortly

- Why is this a concern? The possibility of *sign reversals*:
  - Even if all β_i are positive, β could wind up negative (or vice versa) if ψ_i and β_i are correlated
  - The literature proposes alternative specifications/procedures that avoid negative weighting or allow for custom-built weights

- It turns out that such ψ_i also arise in design-based specifications, and they can also be negative
  - But sign reversals are impossible in design-based specs: then we also have $\beta = E[\phi_i \beta_i]/E[\phi_i]$ for "ex-ante" ϕ_i which are always non-negative

Simple Setup

- Suppose a researcher estimates by OLS:

$$y_i = \beta x_i + w_i'\gamma + e_i$$

for some outcome y_i, treatment x_i, and vector of controls w_i

Simple Setup

- Suppose a researcher estimates by OLS:

  $$y_i = \beta x_i + w_i' \gamma + e_i$$

  for some outcome y_i, treatment x_i, and vector of controls w_i

- To interpret β, we consider a linear-effect causal model:

  $$y_i = \beta_i x_i + \varepsilon_i$$

  with heterogeneous effects β_i and untreated potential outcomes ε_i

  - Note: for binary x_i this is the more familiar $y_i = (y_i(1) - y_i(0))x_i + y_i(0)$

Simple Setup

- Suppose a researcher estimates by OLS:

  $$y_i = \beta x_i + w_i'\gamma + e_i$$

  for some outcome y_i, treatment x_i, and vector of controls w_i

- To interpret β, we consider a linear-effect causal model:

  $$y_i = \beta_i x_i + \varepsilon_i$$

  with heterogeneous effects β_i and untreated potential outcomes ε_i

  - Note: for binary x_i this is the more familiar $y_i = (y_i(1) - y_i(0))x_i + y_i(0)$

- Assume appropriate asymptotics for OLS to consistently estimate:

  $$\beta = \frac{E[\tilde{x}_i y_i]}{E[\tilde{x}_i^2]} = \frac{E[\tilde{x}_i x_i \beta] + E[\tilde{x}_i \varepsilon_i]}{E[\tilde{x}_i^2]}$$

  where $\tilde{x}_i$ are residuals from the population regression of x_i on w_i

Two Paths to Avoiding Omitted Variables Bias

- $E[\tilde{x}_i \varepsilon_i] = 0$ under either one of two assumptions:

Two Paths to Avoiding Omitted Variables Bias

- $E[\tilde{x}_i \varepsilon_i] = 0$ under either one of two assumptions:

  ASSUMPTION 1: $E[\varepsilon_i \mid x_i, w_i] = w_i' \gamma$

  - Untreated potential outcomes are linear in controls, given treatment
  - E.g. parallel trends, where i indexes unit-period pairs in a panel and w_i includes unit and time dummies

Two Paths to Avoiding Omitted Variables Bias

- $E[\tilde{x}_i \varepsilon_i] = 0$ under either one of two assumptions:

  ASSUMPTION 1: $E[\varepsilon_i \mid x_i, w_i] = w_i' \gamma$

  - Untreated potential outcomes are linear in controls, given treatment
  - E.g. parallel trends, where i indexes unit-period pairs in a panel and w_i includes unit and time dummies

  ASSUMPTION 2: $E[x_i \mid \varepsilon_i, \beta_i, w_i] = w_i' \lambda$

  - Treatment is conditionally mean-independent of potential outcomes, with a linear *expected treatment* $E[x_i \mid w_i]$ (e.g. the propensity score)
  - E.g. a stratified experiment, where x_i is randomly assigned within strata dummied out in w_i
  - Note we're conditioning on *both* ε_i and β_i, ruling out "selection on gains" (will relax with IV version soon)

Two Paths to Avoiding Omitted Variables Bias

- $E[\tilde{x}_i \varepsilon_i] = 0$ under either one of two assumptions:

  ASSUMPTION 1: $E[\varepsilon_i \mid x_i, w_i] = w_i' \gamma$

  - Untreated potential outcomes are linear in controls, given treatment
  - E.g. parallel trends, where i indexes unit-period pairs in a panel and w_i includes unit and time dummies

  ASSUMPTION 2: $E[x_i \mid \varepsilon_i, \beta_i, w_i] = w_i' \lambda$

  - Treatment is conditionally mean-independent of potential outcomes, with a linear *expected treatment* $E[x_i \mid w_i]$ (e.g. the propensity score)
  - E.g. a stratified experiment, where x_i is randomly assigned within strata dummied out in w_i
  - Note we're conditioning on *both* ε_i and β_i, ruling out "selection on gains" (will relax with IV version soon)

- The second assumption yields a design-based OLS specification

  - Stronger (sufficient) condition: $x_i \mid (\varepsilon_i, \beta_i, w_i) \overset{iid}{\sim} G(w_i)$

Ex-Post Weights

- Since $E[\tilde{x}_i \varepsilon_i] = 0$, the OLS estimand has an average-effect representation under either assumption:

$$\beta = \frac{E[\psi_i \beta_i]}{E[\psi_i]}, \qquad \psi_i = \tilde{x}_i x_i$$

Ex-Post Weights

- Since $E[\tilde{x}_i \varepsilon_i] = 0$, the OLS estimand has an average-effect representation under either assumption:

$$\beta = \frac{E[\psi_i \beta_i]}{E[\psi_i]}, \qquad \psi_i = \tilde{x}_i x_i$$

- But the ex-post weights ψ_i are generally non-convex: $E[\tilde{x}_i] = 0$, so $\tilde{x}_i$ must take on both positive and negative values

Ex-Post Weights

- Since $E[\tilde{x}_i \varepsilon_i] = 0$, the OLS estimand has an average-effect representation under either assumption:

$$\beta = \frac{E[\psi_i \beta_i]}{E[\psi_i]}, \qquad \psi_i = \tilde{x}_i x_i$$

- But the ex-post weights ψ_i are generally non-convex: $E[\tilde{x}_i] = 0$, so $\tilde{x}_i$ must take on both positive and negative values
  - E.g. if $x_i > 0$ then i with low values of x_i (the effective control group) will always receive negative ex-post weight

Ex-Post Weights

- Since $E[\tilde{x}_i \varepsilon_i] = 0$, the OLS estimand has an average-effect representation under either assumption:

$$\beta = \frac{E[\psi_i \beta_i]}{E[\psi_i]}, \qquad \psi_i = \tilde{x}_i x_i$$

- But the ex-post weights ψ_i are generally non-convex: $E[\tilde{x}_i] = 0$, so $\tilde{x}_i$ must take on both positive and negative values
  - E.g. if $x_i > 0$ then i with low values of x_i (the effective control group) will always receive negative ex-post weight
  - This can lead to sign reversals: e.g. $\beta < 0$, despite $\beta_i > 0$

Ex-Post Weights

- Since $E[\tilde{x}_i \varepsilon_i] = 0$, the OLS estimand has an average-effect representation under either assumption:

$$\beta = \frac{E[\psi_i \beta_i]}{E[\psi_i]}, \qquad \psi_i = \tilde{x}_i x_i$$

- But the ex-post weights ψ_i are generally non-convex: $E[\tilde{x}_i] = 0$, so $\tilde{x}_i$ must take on both positive and negative values
  - E.g. if $x_i > 0$ then i with low values of x_i (the effective control group) will always receive negative ex-post weight
  - This can lead to sign reversals: e.g. $\beta < 0$, despite $\beta_i > 0$

- The ex-post weights are the end of the story for β under Assumption 1. But in design-based specifications we can take one more step
  - In experiments, who is in the effective control group is *random*. Before treatment is drawn, everyone expects the same weight!

Ex-Ante Weights

- Using the law of iterated expectations, we can also write:

$$\beta = \frac{E[E[\psi_i \mid w_i, \beta_i]\beta_i]}{E[E[\psi_i \mid w_i, \beta_i]]} \equiv \frac{E[\phi_i \beta_i]}{E[\phi_i]}$$

  for ex-ante weights $\phi_i = E[\tilde{x}_i x_i \mid w_i, \beta_i]$

Ex-Ante Weights

- Using the law of iterated expectations, we can also write:

  $$\beta = \frac{E[E[\psi_i \mid w_i, \beta_i]\beta_i]}{E[E[\psi_i \mid w_i, \beta_i]]} \equiv \frac{E[\phi_i \beta_i]}{E[\phi_i]}$$

  for ex-ante weights $\phi_i = E[\tilde{x}_i x_i \mid w_i, \beta_i]$

  - Under Assumption 1, this need not help: i.e. if treatment is deterministic in the unit/time FE in w_i, then $\phi_i = \psi_i$

Ex-Ante Weights

- Using the law of iterated expectations, we can also write:

  $$\beta = \frac{E[E[\psi_i \mid w_i, \beta_i]\beta_i]}{E[E[\psi_i \mid w_i, \beta_i]]} \equiv \frac{E[\phi_i \beta_i]}{E[\phi_i]}$$

  for ex-ante weights $\phi_i = E[\tilde{x}_i x_i \mid w_i, \beta_i]$

  - Under Assumption 1, this need not help: i.e. if treatment is deterministic in the unit/time FE in w_i, then $\phi_i = \psi_i$

- But under Assumption 2, $\phi_i = Var(x_i \mid w_i, \beta_i)$ which is non-negative!

Ex-Ante Weights

- Using the law of iterated expectations, we can also write:

  $$\beta = \frac{E[E[\psi_i \mid w_i, \beta_i]\beta_i]}{E[E[\psi_i \mid w_i, \beta_i]]} \equiv \frac{E[\phi_i \beta_i]}{E[\phi_i]}$$

  for ex-ante weights $\phi_i = E[\tilde{x}_i x_i \mid w_i, \beta_i]$

  - Under Assumption 1, this need not help: i.e. if treatment is deterministic in the unit/time FE in w_i, then $\phi_i = \psi_i$

- But under Assumption 2, $\phi_i = Var(x_i \mid w_i, \beta_i)$ which is non-negative!
  - $E[\tilde{x}_i x_i \mid w_i, \beta_i] = E[\tilde{x}_i^2 \mid w_i, \beta_i] + E[\tilde{x}_i \mid w_i, \beta_i] w_i' \lambda = Var(x_i \mid w_i, \beta_i) + 0$

Ex-Ante Weights

- Using the law of iterated expectations, we can also write:

$$\beta = \frac{E[E[\psi_i \mid w_i, \beta_i]\beta_i]}{E[E[\psi_i \mid w_i, \beta_i]]} \equiv \frac{E[\phi_i \beta_i]}{E[\phi_i]}$$

  for ex-ante weights $\phi_i = E[\tilde{x}_i x_i \mid w_i, \beta_i]$

  - Under Assumption 1, this need not help: i.e. if treatment is deterministic in the unit/time FE in w_i, then $\phi_i = \psi_i$

- But under Assumption 2, $\phi_i = Var(x_i \mid w_i, \beta_i)$ which is non-negative!
  - $E[\tilde{x}_i x_i \mid w_i, \beta_i] = E[\tilde{x}_i^2 \mid w_i, \beta_i] + E[\tilde{x}_i \mid w_i, \beta_i] w_i' \lambda = Var(x_i \mid w_i, \beta_i) + 0$

- Hence: sign reversals cannot occur in design-based OLS specifications

Interpretation

- Even if we formulate a design-based regression in terms of constant effects, the estimand is still reasonable under heterogeneous effects
  - Not necessarily true for outcome models (makes sense: we were just modeling ε_i! But additional models on β_i need not help)

Interpretation

- Even if we formulate a design-based regression in terms of constant effects, the estimand is still reasonable under heterogeneous effects
    - Not necessarily true for outcome models (makes sense: we were just modeling ε_i! But additional models on β_i need not help)

- With the stronger design assumption of $x_i \mid (\varepsilon_i, \beta_i, w_i) \overset{iid}{\sim} G(w_i)$, the ex ante weights become identified: $\phi_i = Var(x_i \mid w_i, \beta_i) = Var(x_i \mid w_i)$
    - C.f. earlier results in Angrist (1998), Angrist and Krueger (1999), etc

Interpretation

- Even if we formulate a design-based regression in terms of constant effects, the estimand is still reasonable under heterogeneous effects
  - Not necessarily true for outcome models (makes sense: we were just modeling ε_i! But additional models on β_i need not help)

- With the stronger design assumption of $x_i \mid (\varepsilon_i, \beta_i, w_i) \overset{iid}{\sim} G(w_i)$, the ex ante weights become identified: $\phi_i = Var(x_i \mid w_i, \beta_i) = Var(x_i \mid w_i)$
  - C.f. earlier results in Angrist (1998), Angrist and Krueger (1999), etc
  - Could inverse-weight by $\widehat{Var}(x_i \mid w_i)$ to estimate unweighted $E[\beta_i]$

Interpretation

- Even if we formulate a design-based regression in terms of constant effects, the estimand is still reasonable under heterogeneous effects
    - Not necessarily true for outcome models (makes sense: we were just modeling ε_i! But additional models on β_i need not help)

- With the stronger design assumption of $x_i \mid (\varepsilon_i, \beta_i, w_i) \overset{iid}{\sim} G(w_i)$, the ex ante weights become identified: $\phi_i = Var(x_i \mid w_i, \beta_i) = Var(x_i \mid w_i)$
    - C.f. earlier results in Angrist (1998), Angrist and Krueger (1999), etc
    - Could inverse-weight by $\widehat{Var}(x_i \mid w_i)$ to estimate unweighted $E[\beta_i]$

- Of course, the ϕ_i-weighted estimand may not be most of interest!
    - If $Cov(\phi_i, \beta_i) \approx 0$, we'll still get something close to $E[\beta_i]$
    - Otherwise, ϕ_i-weighting has desirable efficiency properties (Goldsmith-Pinkham et al. 2024)
    - Large class of alternative propensity-score-based estimators for other estimands under the stronger design assumption

General Setting

- Borusyak and Hull (2024) extend ex ante / ex post weights to:
  1. A more general causal model: potential outcomes $y_i(x)$ and $y_i = y_i(x_i)$
  2. IV: design-based assumption is then $E[z_i \mid y_i(\cdot), w_i] = w_i'\lambda$

General Setting

- Borusyak and Hull (2024) extend ex ante / ex post weights to:
  1. A more general causal model: potential outcomes $y_i(x)$ and $y_i = y_i(x_i)$
  2. IV: design-based assumption is then $E[z_i \mid y_i(\cdot), w_i] = w_i'\lambda$

- For convex ex-ante weights in IV we require first-stage *monotonicity*: that x_i is non-decreasing in z_i for all units regardless of $y_i(\cdot)$
  - C.f. earlier results in Imbens and Angrist ('94, '95), Angrist et al. ('00)

General Setting

- Borusyak and Hull (2024) extend ex ante / ex post weights to:
  1. A more general causal model: potential outcomes $y_i(x)$ and $y_i = y_i(x_i)$
  2. IV: design-based assumption is then $E[z_i \mid y_i(\cdot), w_i] = w_i'\lambda$

- For convex ex-ante weights in IV we require first-stage *monotonicity*: that x_i is non-decreasing in z_i for all units regardless of $y_i(\cdot)$
  - C.f. earlier results in Imbens and Angrist ('94, '95), Angrist et al. ('00)
  - Ex post weights are still potentially non-convex under monotonicity

General Setting

- Borusyak and Hull (2024) extend ex ante / ex post weights to:
    1. A more general causal model: potential outcomes $y_i(x)$ and $y_i = y_i(x_i)$
    2. IV: design-based assumption is then $E[z_i \mid y_i(\cdot), w_i] = w_i'\lambda$

- For convex ex-ante weights in IV we require first-stage *monotonicity*: that x_i is non-decreasing in z_i for all units regardless of $y_i(\cdot)$
    - C.f. earlier results in Imbens and Angrist ('94, '95), Angrist et al. ('00)
    - Ex post weights are still potentially non-convex under monotonicity

- Framework is general, allowing for "formula" IVs (e.g. shift-share)
    - We'll see more about this in Friday's class

Multiple Treatments: Contamination Bias

- Goldsmith-Pinkham et al. (2024) generalize things in a different direction: hetFX weighting for regressions w/multiple treatments
  - Unfortunately the picture is a bit less rosy for design here

Multiple Treatments: Contamination Bias

- Goldsmith-Pinkham et al. (2024) generalize things in a different direction: hetFX weighting for regressions w/multiple treatments
  - Unfortunately the picture is a bit less rosy for design here

- The coefficient on treatment j estimates the sum of two terms:
  1. A weighted average of treatment j's effects, with convex weights in design-based specifications ✓

Multiple Treatments: Contamination Bias

- Goldsmith-Pinkham et al. (2024) generalize things in a different direction: hetFX weighting for regressions w/multiple treatments
    - Unfortunately the picture is a bit less rosy for design here

- The coefficient on treatment j estimates the sum of two terms:
    1. A weighted average of treatment j's effects, with convex weights in design-based specifications ✓
    2. A non-convex combination of effects from other treatments k ("contamination bias") X

Multiple Treatments: Contamination Bias

- Goldsmith-Pinkham et al. (2024) generalize things in a different direction: hetFX weighting for regressions w/multiple treatments
  - Unfortunately the picture is a bit less rosy for design here

- The coefficient on treatment j estimates the sum of two terms:
  1. A weighted average of treatment j's effects, with convex weights in design-based specifications ✓
  2. A non-convex combination of effects from other treatments k ("contamination bias") X

- They derive alternative estimators which avoid contamination bias while maintaining the nice efficiency properties of OLS weighting
  - Ultimately, becomes an empirical question of how important bias is

Example: Project STAR

- Krueger (1999) studies the STAR RCT, which randomized 12k students in 80 public elementary schools in Tennessee (!) to one of 3 classroom types:
    1. Regular-sized (20-25 students) – Control
    2. Small (13-17 students) – Treatment 1
    3. Regular-sized with a teaching aide – Treatment 2

Example: Project STAR

- Krueger (1999) studies the STAR RCT, which randomized 12k students in 80 public elementary schools in Tennessee (!) to one of 3 classroom types:
  1. Regular-sized (20-25 students) – Control
  2. Small (13-17 students) – Treatment 1
  3. Regular-sized with a teaching aide – Treatment 2

- Kids were randomized within schools, so the propensity of assignment to each treatment varied by school
  - Krueger thus estimates: $TestScore_i = \alpha_{school(i)} + \beta_1 D_{i1} + \beta_2 D_{i2} + \varepsilon_i$

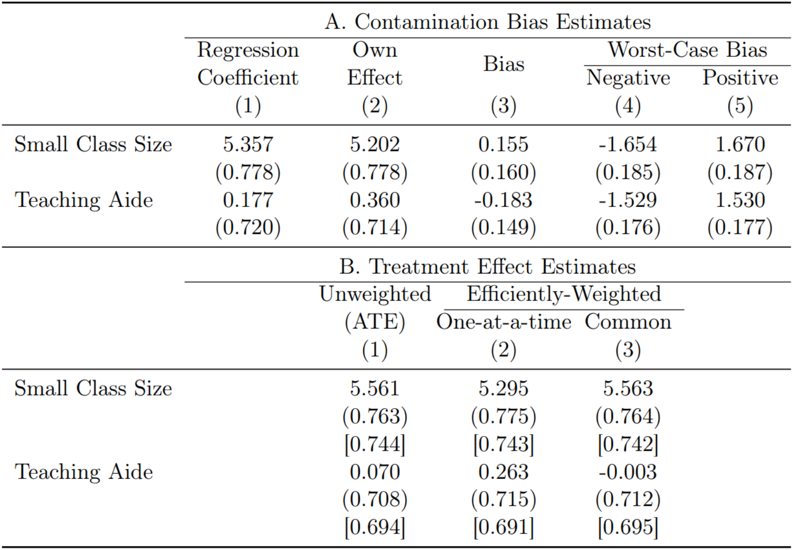
Example: Project STAR

- Krueger (1999) studies the STAR RCT, which randomized 12k students in 80 public elementary schools in Tennessee (!) to one of 3 classroom types:
  1. Regular-sized (20-25 students) – Control
  2. Small (13-17 students) – Treatment 1
  3. Regular-sized with a teaching aide – Treatment 2

- Kids were randomized within schools, so the propensity of assignment to each treatment varied by school
  - Krueger thus estimates: $TestScore_i = \alpha_{school(i)} + \beta_1 D_{i1} + \beta_2 D_{i2} + \varepsilon_i$

- We find significant *potential* for contamination bias: lots of treatment effect heterogeneity and variation in contamination weights
  - But actual contamination bias is minimal: $Corr(effects, weights) \approx 0$

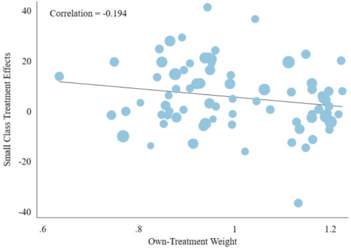
Project STAR, Revisited



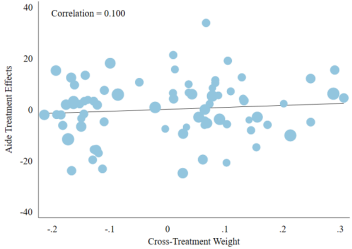
	A. Contamination Bias Estimates				
	Regression Coefficient	Own Effect	Bias	Worst-Case Bias	
				Negative	Positive
	(1)	(2)	(3)	(4)	(5)
Small Class Size	5.357	5.202	0.155	-1.654	1.670
	(0.778)	(0.778)	(0.160)	(0.185)	(0.187)
Teaching Aide	0.177	0.360	-0.183	-1.529	1.530
	(0.720)	(0.714)	(0.149)	(0.176)	(0.177)

	B. Treatment Effect Estimates		
	Unweighted	Efficiently-Weighted	
	(ATE)	One-at-a-time	Common
	(1)	(2)	(3)
Small Class Size	5.561	5.295	5.563
	(0.763)	(0.775)	(0.764)
	[0.744]	[0.743]	[0.742]
Teaching Aide	0.070	0.263	-0.003
	(0.708)	(0.715)	(0.712)
	[0.694]	[0.691]	[0.695]

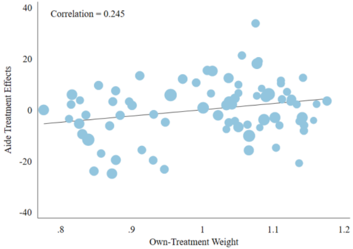
STAR Regression Weights vs. Treatment Effects



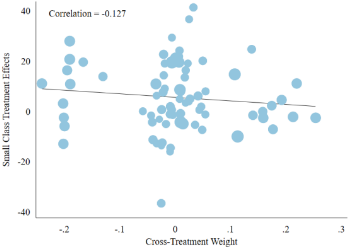
Panel A: Small Class
Own-Treatment Weight



Panel B: Aide
Cross-Treatment Weight



Panel C: Aide
Own-Treatment Weight



Panel D: Small Class
Cross-Treatment Weight

Outline

1. Heterogeneous Treatment Effects✓

2. Clustered Standard Errors

OLS Asymptotics: Review

- Where do SEs come from? OLS $\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$ can be rewritten:

$$\sqrt{N}(\hat{\beta} - \beta) = \left(\frac{\mathbf{X}'\mathbf{X}}{N}\right)^{-1} \left(\frac{\mathbf{X}'\boldsymbol{\varepsilon}}{\sqrt{N}}\right)$$

  where $\mathbf{Y} = \mathbf{X}\beta + \boldsymbol{\varepsilon}$ stacks observations of the population regression

OLS Asymptotics: Review

- Where do SEs come from? OLS $\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$ can be rewritten:

$$\sqrt{N}(\hat{\beta} - \beta) = \left(\frac{\mathbf{X}'\mathbf{X}}{N}\right)^{-1} \left(\frac{\mathbf{X}'\boldsymbol{\varepsilon}}{\sqrt{N}}\right)$$

  where $\mathbf{Y} = \mathbf{X}\beta + \boldsymbol{\varepsilon}$ stacks observations of the population regression

  - Under rather mild conditions (a LLN), $\frac{\mathbf{X}'\mathbf{X}}{N} \xrightarrow{p} E\left[\frac{1}{N}\sum_i X_i X_i'\right]$

OLS Asymptotics: Review

- Where do SEs come from? OLS $\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$ can be rewritten:

$$\sqrt{N}(\hat{\beta} - \beta) = \left(\frac{\mathbf{X}'\mathbf{X}}{N}\right)^{-1}\left(\frac{\mathbf{X}'\boldsymbol{\varepsilon}}{\sqrt{N}}\right)$$

  where $\mathbf{Y} = \mathbf{X}\beta + \boldsymbol{\varepsilon}$ stacks observations of the population regression

  - Under rather mild conditions (a LLN), $\frac{\mathbf{X}'\mathbf{X}}{N} \xrightarrow{p} E\left[\frac{1}{N}\sum_i X_i X_i'\right]$
  - W/slightly stronger conditions (a CLT), $\frac{\mathbf{X}'\boldsymbol{\varepsilon}}{\sqrt{N}} \Rightarrow \mathrm{N}(0, Var(\frac{1}{\sqrt{N}}\sum_i X_i \varepsilon_i))$

OLS Asymptotics: Review

- Where do SEs come from? OLS $\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$ can be rewritten:

$$\sqrt{N}(\hat{\beta} - \beta) = \left(\frac{\mathbf{X}'\mathbf{X}}{N}\right)^{-1} \left(\frac{\mathbf{X}'\boldsymbol{\varepsilon}}{\sqrt{N}}\right)$$

  where $\mathbf{Y} = \mathbf{X}\beta + \boldsymbol{\varepsilon}$ stacks observations of the population regression

  - Under rather mild conditions (a LLN), $\frac{\mathbf{X}'\mathbf{X}}{N} \xrightarrow{p} E\left[\frac{1}{N}\sum_i X_i X_i'\right]$
  - W/slightly stronger conditions (a CLT), $\frac{\mathbf{X}'\boldsymbol{\varepsilon}}{\sqrt{N}} \Rightarrow \mathrm{N}(0, Var(\frac{1}{\sqrt{N}}\sum_i X_i \varepsilon_i))$

- This gives our general asymptotic approximation for OLS: $\hat{\beta} \approx \beta^*$ for

$$\beta^* \sim \mathrm{N}(\beta, \frac{V}{N}), \quad V = E\left[\frac{1}{N}\sum_i X_i X_i'\right]^{-1} Var\left(\frac{1}{\sqrt{N}}\sum_i X_i \varepsilon_i\right) E\left[\frac{1}{N}\sum_i X_i X_i'\right]^{-1}$$

OLS Asymptotics: Review

- Where do SEs come from? OLS $\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$ can be rewritten:

$$\sqrt{N}(\hat{\beta} - \beta) = \left(\frac{\mathbf{X}'\mathbf{X}}{N}\right)^{-1}\left(\frac{\mathbf{X}'\boldsymbol{\varepsilon}}{\sqrt{N}}\right)$$

  where $\mathbf{Y} = \mathbf{X}\beta + \boldsymbol{\varepsilon}$ stacks observations of the population regression

  - Under rather mild conditions (a LLN), $\frac{\mathbf{X}'\mathbf{X}}{N} \xrightarrow{p} E\left[\frac{1}{N}\sum_i X_i X_i'\right]$
  - W/slightly stronger conditions (a CLT), $\frac{\mathbf{X}'\boldsymbol{\varepsilon}}{\sqrt{N}} \Rightarrow \mathrm{N}(0, Var(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i))$

- This gives our general asymptotic approximation for OLS: $\hat{\beta} \approx \beta^*$ for

$$\beta^* \sim \mathrm{N}(\beta, \frac{V}{N}), \quad V = E\left[\frac{1}{N}\sum_i X_i X_i'\right]^{-1} Var\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i\right) E\left[\frac{1}{N}\sum_i X_i X_i'\right]^{-1}$$

- SEs come from $\hat{V} = \left(\frac{1}{N}\sum_i X_i X_i'\right)^{-1} \widehat{Var}\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i\right)\left(\frac{1}{N}\sum_i X_i X_i'\right)^{-1}$

Getting to the Meat of the “Sandwich Estimator,” $\hat{V}$

- Key q: how do we form the variance estimate $\widehat{Var}\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i\right)$?

Getting to the Meat of the "Sandwich Estimator," $\hat{V}$

- Key q: how do we form the variance estimate $\widehat{Var}\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i\right)$?

- In *iid* data, we know $Var\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i\right) =$

Getting to the Meat of the "Sandwich Estimator," $\hat{V}$

- Key q: how do we form the variance estimate $\widehat{Var}\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i\right)$?

- In *iid* data, we know $Var\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i\right) = \frac{1}{N}\sum_i Var(X_i\varepsilon_i) =$

Getting to the Meat of the “Sandwich Estimator,” $\hat{V}$

- Key q: how do we form the variance estimate $\widehat{Var}\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i\right)$?

- In *iid* data, we know $Var\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i\right) = \frac{1}{N}\sum_i Var(X_i\varepsilon_i) = E[X_i X_i'\varepsilon_i^2]$

Getting to the Meat of the “Sandwich Estimator,” $\hat{V}$

- Key q: how do we form the variance estimate $\widehat{Var}\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i\right)$?

- In *iid* data, we know $Var\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i\right) = \frac{1}{N}\sum_i Var(X_i\varepsilon_i) = E[X_iX_i'\varepsilon_i^2]$
  - This suggests $\widehat{Var}\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i^2\right) = \frac{1}{N}\sum_i X_iX_i'\hat{\varepsilon}_i$, which leads to our usual heteroskedasticity-robust estimator

Getting to the Meat of the “Sandwich Estimator,” $\hat{V}$

- Key q: how do we form the variance estimate $\widehat{Var}\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i\right)$?

- In *iid* data, we know $Var\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i\right) = \frac{1}{N}\sum_i Var(X_i\varepsilon_i) = E[X_iX_i'\varepsilon_i^2]$
  - This suggests $\widehat{Var}\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i^2\right) = \frac{1}{N}\sum_i X_iX_i'\hat{\varepsilon}_i$, which leads to our usual heteroskedasticity-robust estimator

- The motivation for alternative estimators comes from the possibility that $X_i\varepsilon_i$ and $X_j\varepsilon_j$ may be correlated for $i \neq j$
  - Generally, $Var\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i\right) = \frac{1}{N}\sum_i Var(X_i\varepsilon_i) + 2\sum_{i,j\neq i} Cov(X_i\varepsilon_i, X_j\varepsilon_j)$

Getting to the Meat of the "Sandwich Estimator," $\hat{V}$

- Key q: how do we form the variance estimate $\widehat{Var}\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i\right)$?

- In *iid* data, we know $Var\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i\right) = \frac{1}{N}\sum_i Var(X_i\varepsilon_i) = E[X_iX_i'\varepsilon_i^2]$
    - This suggests $\widehat{Var}\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i^2\right) = \frac{1}{N}\sum_i X_iX_i'\hat{\varepsilon}_i$, which leads to our usual heteroskedasticity-robust estimator

- The motivation for alternative estimators comes from the possibility that $X_i\varepsilon_i$ and $X_j\varepsilon_j$ may be correlated for $i \neq j$
    - Generally, $Var\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i\right) = \frac{1}{N}\sum_i Var(X_i\varepsilon_i) + 2\sum_{i,j\neq i} Cov(X_i\varepsilon_i, X_j\varepsilon_j)$
    - But we can't allow for arbitrary cross-sectional correlations, since then we couldn't guarantee $\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i$ converges ...

Getting to the Meat of the "Sandwich Estimator," $\hat{V}$

- Key q: how do we form the variance estimate $\widehat{Var}\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i\right)$?

- In *iid* data, we know $Var\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i\right) = \frac{1}{N}\sum_i Var(X_i\varepsilon_i) = E[X_i X_i' \varepsilon_i^2]$
  - This suggests $\widehat{Var}\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i^2\right) = \frac{1}{N}\sum_i X_i X_i' \hat{\varepsilon}_i$, which leads to our usual heteroskedasticity-robust estimator

- The motivation for alternative estimators comes from the possibility that $X_i\varepsilon_i$ and $X_j\varepsilon_j$ may be correlated for $i \neq j$
  - Generally, $Var\left(\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i\right) = \frac{1}{N}\sum_i Var(X_i\varepsilon_i) + 2\sum_{i,j\neq i} Cov(X_i\varepsilon_i, X_j\varepsilon_j)$
  - But we can't allow for arbitrary cross-sectional correlations, since then we couldn't guarantee $\frac{1}{\sqrt{N}}\sum_i X_i\varepsilon_i$ converges ...
  - We need to zero out some covariances to make progress

Cluster-Robust Estimators

- Suppose we can partition observations into clusters, $c(i) \in 1, \ldots, C$
  - To ease notation, suppose equal sizes: $|i : c(i) = c| = N/C \equiv T$

Cluster-Robust Estimators

- Suppose we can partition observations into clusters, $c(i) \in 1, \ldots, C$
  - To ease notation, suppose equal sizes: $|i : c(i) = c| = N/C \equiv T$
  - With $N = CT$, OLS can be rewritten: $\sqrt{N}(\hat{\beta} - \beta) = \left(\frac{\mathbf{X}'\mathbf{X}}{N}\right)^{-1} \cdot \left(\frac{\mathbf{X}'\boldsymbol{\varepsilon}}{\sqrt{CT}}\right)$

Cluster-Robust Estimators

- Suppose we can partition observations into clusters, $c(i) \in 1, \ldots, C$
  - To ease notation, suppose equal sizes: $|i : c(i) = c| = N/C \equiv T$
  - With $N = CT$, OLS can be rewritten: $\sqrt{N}(\hat{\beta} - \beta) = \left(\frac{\mathbf{X}'\mathbf{X}}{N}\right)^{-1} \cdot \left(\frac{\mathbf{X}'\boldsymbol{\varepsilon}}{\sqrt{CT}}\right)$
- Define $Q_c = \frac{1}{\sqrt{T}} \sum_{i:c(i)=c} X_i \varepsilon_i$ and note that $\frac{\mathbf{X}'\boldsymbol{\varepsilon}}{\sqrt{CT}} = \frac{1}{\sqrt{C}} \sum_c Q_c$

Cluster-Robust Estimators

- Suppose we can partition observations into clusters, $c(i) \in 1, \ldots, C$
  - To ease notation, suppose equal sizes: $|i : c(i) = c| = N/C \equiv T$
  - With $N = CT$, OLS can be rewritten: $\sqrt{N}(\hat{\beta} - \beta) = \left(\frac{\mathbf{X}'\mathbf{X}}{N}\right)^{-1} \cdot \left(\frac{\mathbf{X}'\boldsymbol{\varepsilon}}{\sqrt{CT}}\right)$
- Define $Q_c = \frac{1}{\sqrt{T}} \sum_{i:c(i)=c} X_i \varepsilon_i$ and note that $\frac{\mathbf{X}'\boldsymbol{\varepsilon}}{\sqrt{CT}} = \frac{1}{\sqrt{C}} \sum_c Q_c$
  - If the Q_c clusters are *iid*, a CLT applies: $\frac{1}{\sqrt{C}} \sum_c Q_c \Rightarrow \mathrm{N}(0, Var(Q_c))$
  - E.g. in a balanced panel, could have *iid* series $(X_{c1}\varepsilon_{c1} \ldots, X_{cT}\varepsilon_{cT})$

Cluster-Robust Estimators

- Suppose we can partition observations into clusters, $c(i) \in 1, \ldots, C$
  - To ease notation, suppose equal sizes: $|i : c(i) = c| = N/C \equiv T$
  - With $N = CT$, OLS can be rewritten: $\sqrt{N}(\hat{\beta} - \beta) = \left(\frac{\mathbf{X}'\mathbf{X}}{N}\right)^{-1} \cdot \left(\frac{\mathbf{X}'\boldsymbol{\varepsilon}}{\sqrt{CT}}\right)$
- Define $Q_c = \frac{1}{\sqrt{T}} \sum_{i:c(i)=c} X_i \varepsilon_i$ and note that $\frac{\mathbf{X}'\boldsymbol{\varepsilon}}{\sqrt{CT}} = \frac{1}{\sqrt{C}} \sum_c Q_c$
  - If the Q_c clusters are *iid*, a CLT applies: $\frac{1}{\sqrt{C}} \sum_c Q_c \Rightarrow \mathrm{N}(0, Var(Q_c))$
  - E.g. in a balanced panel, could have *iid* series $(X_{c1}\varepsilon_{c1} \ldots, X_{cT}\varepsilon_{cT})$
- This gives us a new "clustered" variance estimate to plug into $\hat{V}$:

$$\widehat{Var}\left(\frac{1}{\sqrt{N}} \sum_i X_i \varepsilon_i\right) = \frac{1}{C} \sum_c \hat{Q}_c^2, \text{ for } \hat{Q} = \frac{1}{\sqrt{T}} \sum_{i:c(i)=c} X_i \hat{\varepsilon}_i$$

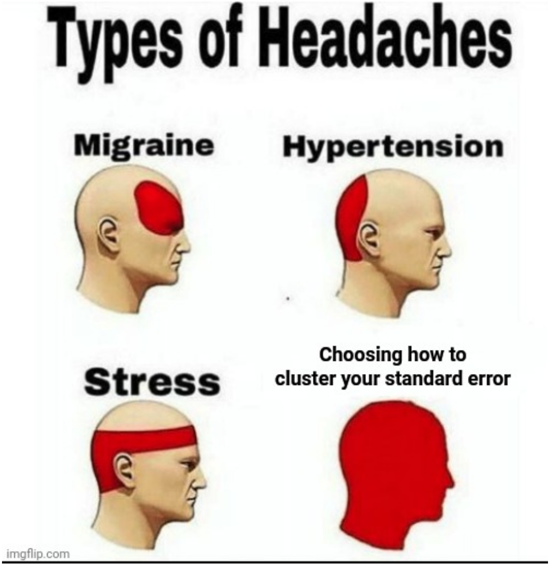
Cluster-Robust Estimators

- Suppose we can partition observations into clusters, $c(i) \in 1, \dots, C$
  - To ease notation, suppose equal sizes: $|i : c(i) = c| = N/C \equiv T$
  - With $N = CT$, OLS can be rewritten: $\sqrt{N}(\hat{\beta} - \beta) = \left(\frac{\mathbf{X}'\mathbf{X}}{N}\right)^{-1} \cdot \left(\frac{\mathbf{X}'\boldsymbol{\varepsilon}}{\sqrt{CT}}\right)$
- Define $Q_c = \frac{1}{\sqrt{T}} \sum_{i:c(i)=c} X_i \varepsilon_i$ and note that $\frac{\mathbf{X}'\boldsymbol{\varepsilon}}{\sqrt{CT}} = \frac{1}{\sqrt{C}} \sum_c Q_c$
  - If the Q_c clusters are *iid*, a CLT applies: $\frac{1}{\sqrt{C}} \sum_c Q_c \Rightarrow \mathrm{N}(0, Var(Q_c))$
  - E.g. in a balanced panel, could have *iid* series $(X_{c1}\varepsilon_{c1} \dots, X_{cT}\varepsilon_{cT})$
- This gives us a new "clustered" variance estimate to plug into $\hat{V}$:

$$\widehat{Var}\left(\frac{1}{\sqrt{N}} \sum_i X_i \varepsilon_i\right) = \frac{1}{C} \sum_c \hat{Q}_c^2, \text{ for } \hat{Q} = \frac{1}{\sqrt{T}} \sum_{i:c(i)=c} X_i \hat{\varepsilon}_i$$

- This is what's going on under the hood when you ", *cluster(c)*"!

Easy, Right?



Source: Khoa Vu (of course)

Design Can Help!

- At an (unhelpfully) high level, the previous results tell us when to cluster i and j together: when we think $Cov(X_i\varepsilon_i, X_j\varepsilon_j) \neq 0$

Design Can Help!

- At an (unhelpfully) high level, the previous results tell us when to cluster i and j together: when we think $Cov(X_i\varepsilon_i, X_j\varepsilon_j) \neq 0$

- With design this may not be too hard to figure out:
  - Suppose $(X_1, \ldots, X_N) \mid (\varepsilon_1, \ldots, \varepsilon_N)$ is mean-zero with $X_i \perp\!\!\!\perp X_j$ whenever $c(i) \neq c(j)$ (e.g. village-level RCT with $c(i)$ giving i's village)

Design Can Help!

- At an (unhelpfully) high level, the previous results tell us when to cluster i and j together: when we think $Cov(X_i\varepsilon_i, X_j\varepsilon_j) \neq 0$

- With design this may not be too hard to figure out:
  - Suppose $(X_1, \dots, X_N) \mid (\varepsilon_1, \dots, \varepsilon_N)$ is mean-zero with $X_i \perp\!\!\!\perp X_j$ whenever $c(i) \neq c(j)$ (e.g. village-level RCT with $c(i)$ giving i's village)
  - Then whenever $c(i) \neq c(j)$:

$$Cov(X_i\varepsilon_i, X_j\varepsilon_j) = E[X_iX_j'\varepsilon_i\varepsilon_j] =$$

Design Can Help!

- At an (unhelpfully) high level, the previous results tell us when to cluster i and j together: when we think $Cov(X_i\varepsilon_i, X_j\varepsilon_j) \neq 0$

- With design this may not be too hard to figure out:
  - Suppose $(X_1, \dots, X_N) \mid (\varepsilon_1, \dots, \varepsilon_N)$ is mean-zero with $X_i \perp\!\!\!\perp X_j$ whenever $c(i) \neq c(j)$ (e.g. village-level RCT with $c(i)$ giving i's village)
  - Then whenever $c(i) \neq c(j)$:

$$Cov(X_i\varepsilon_i, X_j\varepsilon_j) = E[X_iX_j'\varepsilon_i\varepsilon_j] = E[E[X_iX_j' \mid \varepsilon_i, \varepsilon_j]\varepsilon_i\varepsilon_j] =$$

Design Can Help!

- At an (unhelpfully) high level, the previous results tell us when to cluster i and j together: when we think $Cov(X_i\varepsilon_i, X_j\varepsilon_j) \neq 0$

- With design this may not be too hard to figure out:
  - Suppose $(X_1, \ldots, X_N) \mid (\varepsilon_1, \ldots, \varepsilon_N)$ is mean-zero with $X_i \perp\!\!\!\perp X_j$ whenever $c(i) \neq c(j)$ (e.g. village-level RCT with $c(i)$ giving i's village)
  - Then whenever $c(i) \neq c(j)$:

$$Cov(X_i\varepsilon_i, X_j\varepsilon_j) = E[X_i X_j' \varepsilon_i \varepsilon_j] = E[E[X_i X_j' \mid \varepsilon_i, \varepsilon_j]\varepsilon_i \varepsilon_j] = 0$$

Design Can Help!

- At an (unhelpfully) high level, the previous results tell us when to cluster i and j together: when we think $Cov(X_i\varepsilon_i, X_j\varepsilon_j) \neq 0$

- With design this may not be too hard to figure out:
    - Suppose $(X_1, \dots, X_N) \mid (\varepsilon_1, \dots, \varepsilon_N)$ is mean-zero with $X_i \perp\!\!\!\perp X_j$ whenever $c(i) \neq c(j)$ (e.g. village-level RCT with $c(i)$ giving i's village)
    - Then whenever $c(i) \neq c(j)$:

$$Cov(X_i\varepsilon_i, X_j\varepsilon_j) = E[X_iX_j'\varepsilon_i\varepsilon_j] = E[E[X_iX_j' \mid \varepsilon_i, \varepsilon_j]\varepsilon_i\varepsilon_j] = 0$$

    - So we only need to cluster by $c(i)$: the design tells us what to do!

Design Can Help!

- At an (unhelpfully) high level, the previous results tell us when to cluster i and j together: when we think $Cov(X_i\varepsilon_i, X_j\varepsilon_j) \neq 0$

- With design this may not be too hard to figure out:
  - Suppose $(X_1, \ldots, X_N) \mid (\varepsilon_1, \ldots, \varepsilon_N)$ is mean-zero with $X_i \perp\!\!\!\perp X_j$ whenever $c(i) \neq c(j)$ (e.g. village-level RCT with $c(i)$ giving i's village)
  - Then whenever $c(i) \neq c(j)$:

$$Cov(X_i\varepsilon_i, X_j\varepsilon_j) = E[X_i X_j' \varepsilon_i \varepsilon_j] = E[E[X_i X_j' \mid \varepsilon_i, \varepsilon_j]\varepsilon_i\varepsilon_j] = 0$$

  - So we only need to cluster by $c(i)$: the design tells us what to do!

- This leads to the popular (and sometimes misused) heuristic: cluster at the level of treatment / identifying variation
  - See Abadie et al. (2023) for a more complete version of this argument

Where Intuition Can Fall Short: Paired Randomization

- Suppose (as is often done) we pair individuals up by some baseline characteristics, then in each pair c we randomly treat one individual
  - Treatment is at the individual level... so should we just “, r” ?

Where Intuition Can Fall Short: Paired Randomization

- Suppose (as is often done) we pair individuals up by some baseline characteristics, then in each pair c we randomly treat one individual
  - Treatment is at the individual level... so should we just ", r" ?

- de Chaisemartin and Ramirez-Cuellar (2022) show the answer is no: non-clustered SEs will generally be downward-biased (maybe badly)
  - Under constant effects, $E[\hat{V}] = V/2$; severe over-rejection!

Where Intuition Can Fall Short: Paired Randomization

- Suppose (as is often done) we pair individuals up by some baseline characteristics, then in each pair c we randomly treat one individual
  - Treatment is at the individual level... so should we just “, r” ?
- de Chaisemartin and Ramirez-Cuellar (2022) show the answer is no: non-clustered SEs will generally be downward-biased (maybe badly)
  - Under constant effects, $E[\hat{V}] = V/2$; severe over-rejection!
- Paired randomization makes X_i and X_j *negatively* correlated in pairs
  - Clustering by pair solves this; treatment assignment is *iid* across pairs

Where Intuition Can Fall Short: Paired Randomization

- Suppose (as is often done) we pair individuals up by some baseline characteristics, then in each pair c we randomly treat one individual
  - Treatment is at the individual level... so should we just ", r" ?

- de Chaisemartin and Ramirez-Cuellar (2022) show the answer is no: non-clustered SEs will generally be downward-biased (maybe badly)
  - Under constant effects, $E[\hat{V}] = V/2$; severe over-rejection!

- Paired randomization makes X_i and X_j *negatively* correlated in pairs
  - Clustering by pair solves this; treatment assignment is *iid* across pairs
  - Alternatively, you could ", r" with pair fixed effects (and the standard Stata d.f. correction). Why?

Where Intuition Can Fall Short: Paired Randomization

- Suppose (as is often done) we pair individuals up by some baseline characteristics, then in each pair c we randomly treat one individual
  - Treatment is at the individual level... so should we just “, r” ?
- de Chaisemartin and Ramirez-Cuellar (2022) show the answer is no: non-clustered SEs will generally be downward-biased (maybe badly)
  - Under constant effects, $E[\hat{V}] = V/2$; severe over-rejection!
- Paired randomization makes X_i and X_j *negatively* correlated in pairs
  - Clustering by pair solves this; treatment assignment is *iid* across pairs
  - Alternatively, you could “, r” with pair fixed effects (and the standard Stata d.f. correction). Why? Because FE = FD when $T = 2$