Week 5: Matching and Regression

PLSC 30600 - Causal Inference

Previously

- Identification under **conditional ignorability**
    - Treatment assignment is independent of the potential outcomes *conditional* on observed covariates
    - "Selection-on-observables"
- With discrete and low-dimensional covariates, simple nonparametric estimator:
    - Weighted average of CATEs across strata
    - With continuous/higher-dimensional covariates, often need *some* parametric assumptions.
- Can we adjust for a single scalar?
    - Yes: the propensity score: $e(x) = P(D_i = 1|X_i = x)$
    - IPTW: Weight each unit by the inverse probability of receiving the treatment it received.

This week

- Can we construct a "weighting" estimator that doesn't rely on a parametric model for the outcome?
  - Yes: Matching!
  - Problem: (inexact) matching is biased (though typically less biased than *failing* to adjust for confounding).
- What if we modelled the outcome instead?
  - Regression estimators!: $\hat{E}[Y_i(0)|X]$ and $\hat{E}[Y_i(1)|X]$
- Combining estimators
  - Regresssion to correct for bias in matching
  - Regression + IPTW: "doubly-robust" augmented IPTW

Matching

Imputation estimators

- We want to estimate the sample average treatment effect

$$\tau = \frac{1}{N} \sum_{i=1}^{N} Y_i(1) - Y_i(0)$$

- If we could directly observe $Y_i(1)$ and $Y_i(0)$, we could just plug them into the expression above.
    - We can't...but what if we could construct an *estimator* for each $Y_i(1)$ and $Y_i(0)$ and then plug *those* in.
- Consider $Y_i(1)$.
    - If $D_i = 1$, we can just plug in Y_i
    - If $D_i = 0$, we'll have to come up with some way of *imputing* $Y_i(1)$ from the rest of the data.
- If treatment is completely ignorable, a good (unbiased) estimate is just the average of Y_i in the control group
- If treatment is not completely ignorable, we'll need to somehow use the X_i

Imputation estimators

- In general, a lot of estimators that we use can be written as imputations of the individual potential outcomes

$$\hat{\tau} = \frac{1}{N}\sum_{i=1}^{N} \widehat{Y_i(1)} - \widehat{Y_i(0)}$$

- One intuitive imputation approach to adjust for X_i is to simply impute, for each treated unit, the opposite potential outcome using the control units with the most "similar" values of X_i
  - Do the same among control units (imputing using the "most similar" treated observations).
- These are **matching** estimators

Matching estimators

- How do we define what "close" or "similar" means?
- One approach: choose a *distance metric*
    - Let Q_{ij} denote the distance between the covariates X_i and X_j between units i and j

- Common metrics:

    - *Exact*: $Q_{ij} = 0$ if $X_i = X_j$ and $Q_{ij} = \infty$ if $X_i \neq X_j$
    - *Standardized Euclidean*:

$$Q_{ij} = \sqrt{\sum_{k=1}^{K} \frac{(X_{ik} - X_{jk})^2}{s_k}}$$

    - *Mahalanobis*:

$$Q_{ij} = \sqrt{(X_i - X_j)' S^{-1} (X_i - X_j)}$$

where S is the sample covariance matrix.

Matching with or without replacement

- Should we let units matched to one observation be allowed to be matched again?
- Advantages
  - Bias reduction - we always pick the closest matches.
  - Order of matching doesn't matter
- Challenges:
  - Possibly greater variance (e.g. only one treated unit is "close" to many controls)
- Here, we'll analyze matching **with** replacement.

Nearest-neighbor matching

- For a treated unit with $D_i = 1$, we impute the potential outcomes as:

$$\widehat{Y_i(1)} = Y_i$$

$$\widehat{Y_i(0)} = \frac{1}{M} \sum_{j \in \mathcal{J}_M(i)} Y_j$$

  where $\mathcal{J}_M(i)$ is the set of M closest matches to i among the control observations.

- Do the same for the controls (but impute $Y_i\hat(1)$ using matched treated units)
- We can think of matching as a kind of weighting estimator that assigns a weight of $1 + \frac{K_M(i)}{M}$ to each unit.

$$\tau_M^{\hat{m}} = \frac{1}{N} \sum_{i=1}^{N} (2D_i - 1)\left(1 + \frac{K_M(i)}{M}\right) Y_i$$

ATE or ATT?

- In many settings where we might want to use matching, we have a handful of treated units and many controls.
  - Easy to find a good match for each treated unit
  - *Hard* to find a good match for each control.
- So instead of trying to estimate the ATE, we could try to estimate the ATT instead -- using the controls *only* to impute.

$$\tau^{\hat{m}}_{\text{ATT}} = \frac{1}{N_t} \sum_{i:D_i=1} Y_i - \widehat{Y_i(0)}$$

- **Intuition**: Matching as a form of "pruning" -- many controls will have $K_M(i) = 0$
  - We're throwing away observations! But with good reason.
- ATT in an observational study is often the more policy-relevant quantity
  - e.g.: How were the incomes of people who *actually* received a particular social service improved?

Properties of the simple matching estimator

- Unless matching is exact, Abadie and Imbens (2006) show that matching exhibits a bias.

$$B_M = \frac{1}{N}\sum_{i=1}^{N}(2D_i - 1)\left[\frac{1}{M}\sum_{m=1}^{M}\mu_{1-D_i}(X_i) - \mu_{1-D_i}(X_{\mathcal{J}_m(i)})\right]$$

where $\mu_1(X_i) = E[Y_i(1)|X_i]$ and $\mu_0(X_i) = E[Y_i(0)|X_i]$ are the conditional expectations.

- **Intuitively** - the bias term captures the differences in the conditional expectation function between observation i's covariates and the covariates of the M matches in $\mathcal{J}_m(i)$.
  - When matching is exact, X_i and all of the X_js of the matched units are identical
  - When matching is inexact, we have this **matching discrepancy**
- But does this bias go away in large samples?
  - With many continuous covariates, not fast enough - the rate of convergence of the bias term is slower than that of the sampling variance (the simple matching estimator is not $\sqrt{n}$-consistent).
  - This means our asymptotic approximations for the variance will be poor even in large samples.

Simulation to show the bias

- Let's construct a simulation with confounding. Start with $K = 8$ i.i.d. covariates $X_1, X_2, \ldots X_K$ each distributed $\mathcal{N}(0, 1)$.
- Treatment probability is modeled as a logit

$$\log\left(\frac{e(X_i)}{1 - e(X_i)}\right) = \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \ldots + \beta_k X_k$$

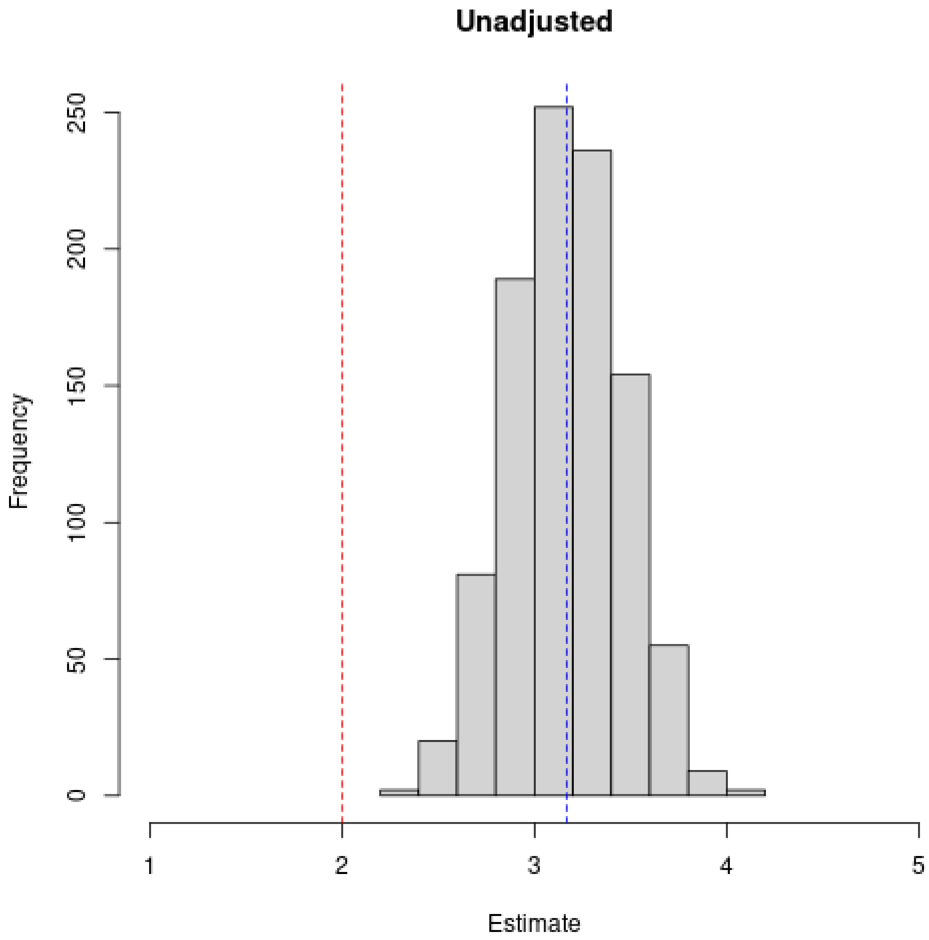
with assumed coefficients $\beta_k = \frac{1}{k}$

- Outcome is linear w/ same coefficients β_k and a constant treatment effect of 2

$$Y_i = 2D_i + \mathbf{X}\beta + \epsilon_i$$

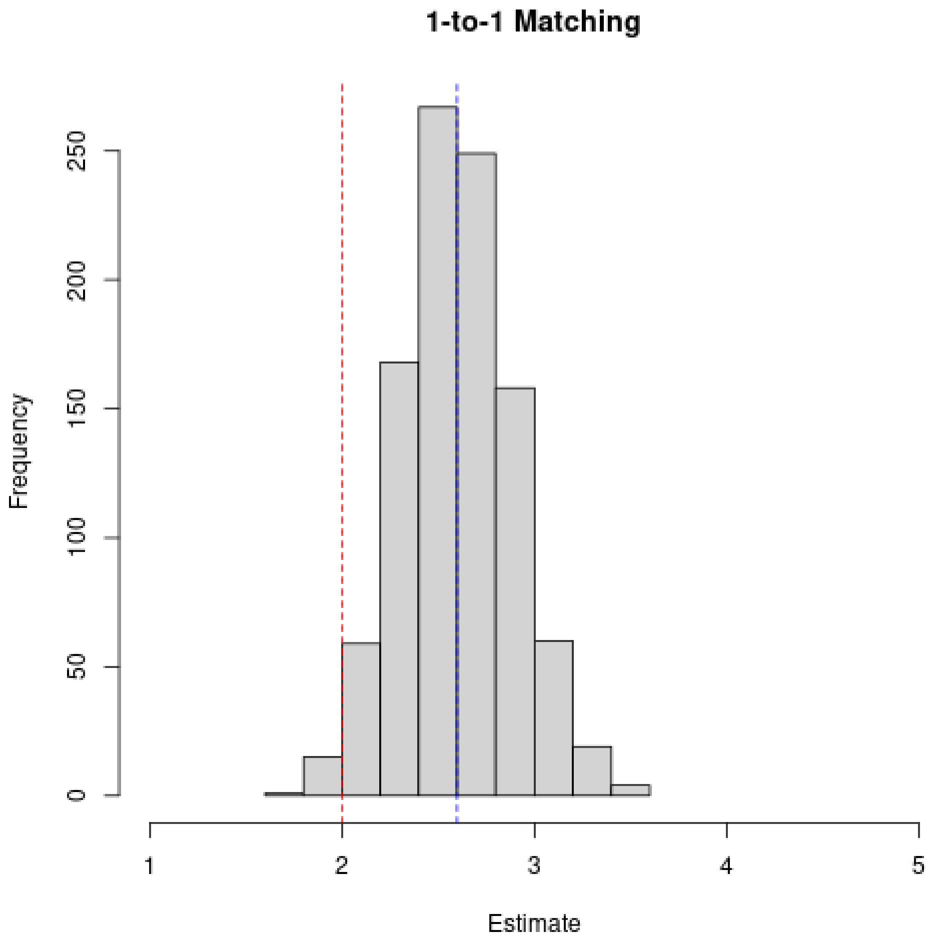
Simulation

- First, our unadjusted simple difference-in-means estimator



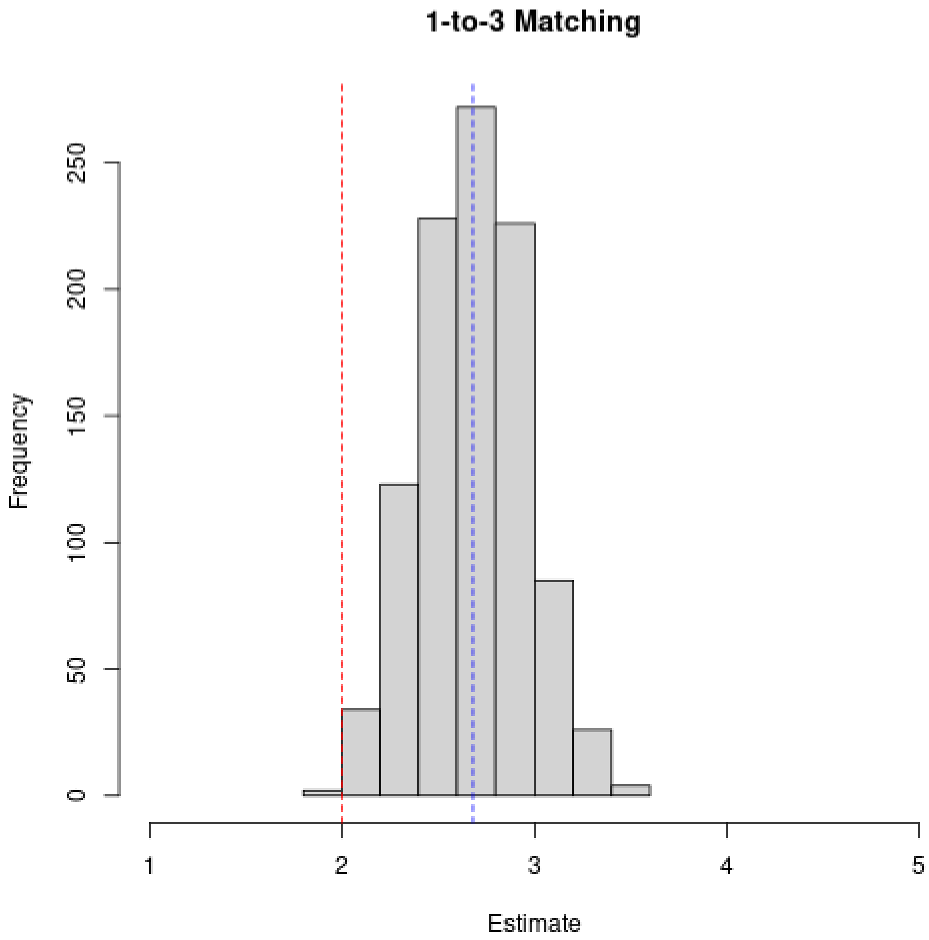
Simulation

- Now, the 1-to-1 matching estimator



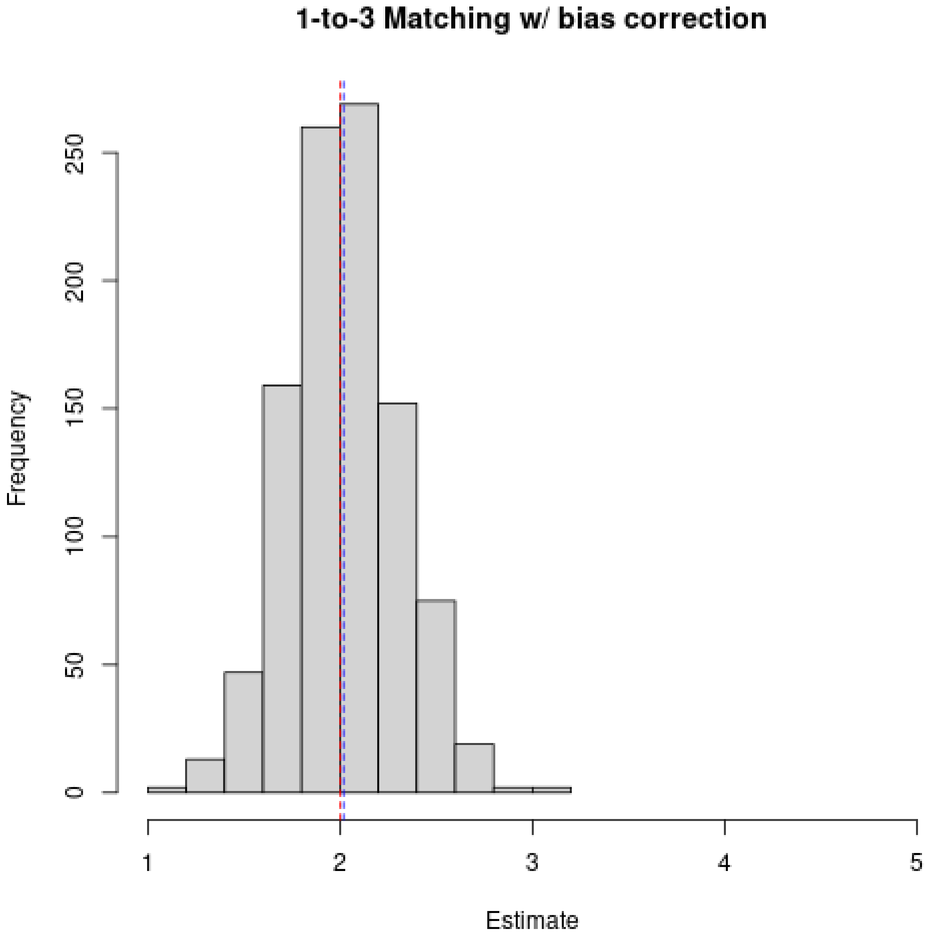
Simulation

- How about 1-to-3 matching?



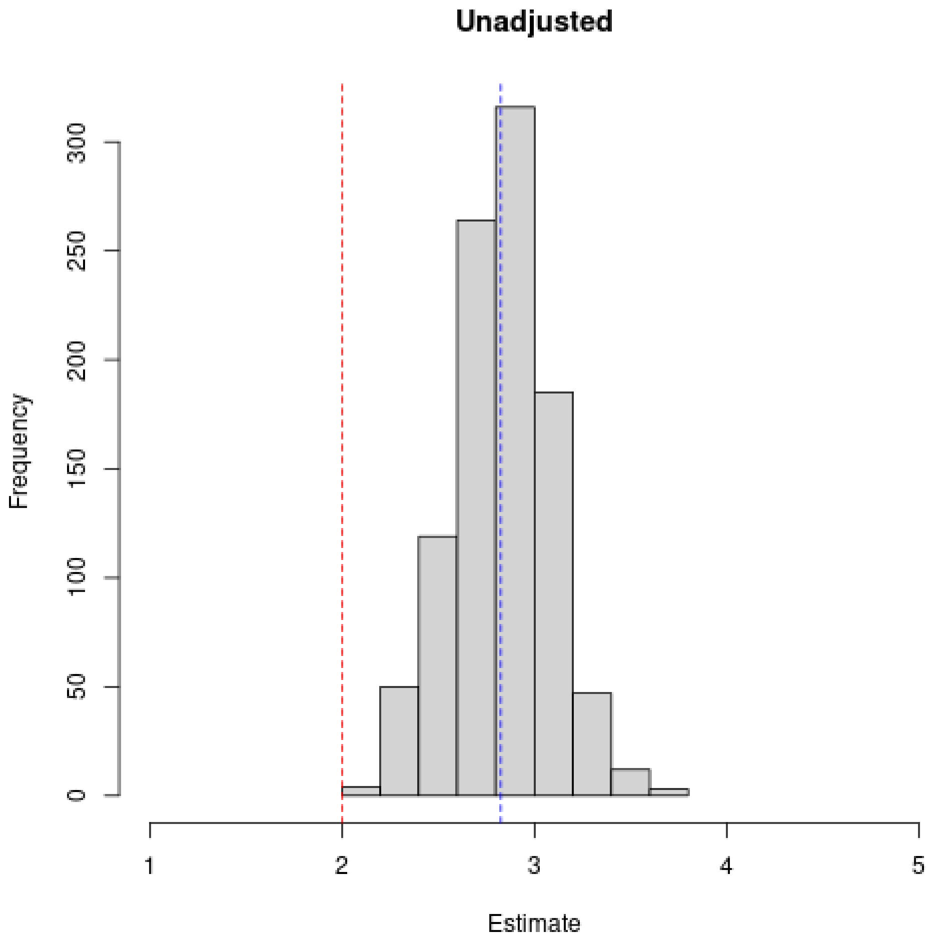
Simulation

- Now, what if we estimate the bias correction (using a regression estimator)



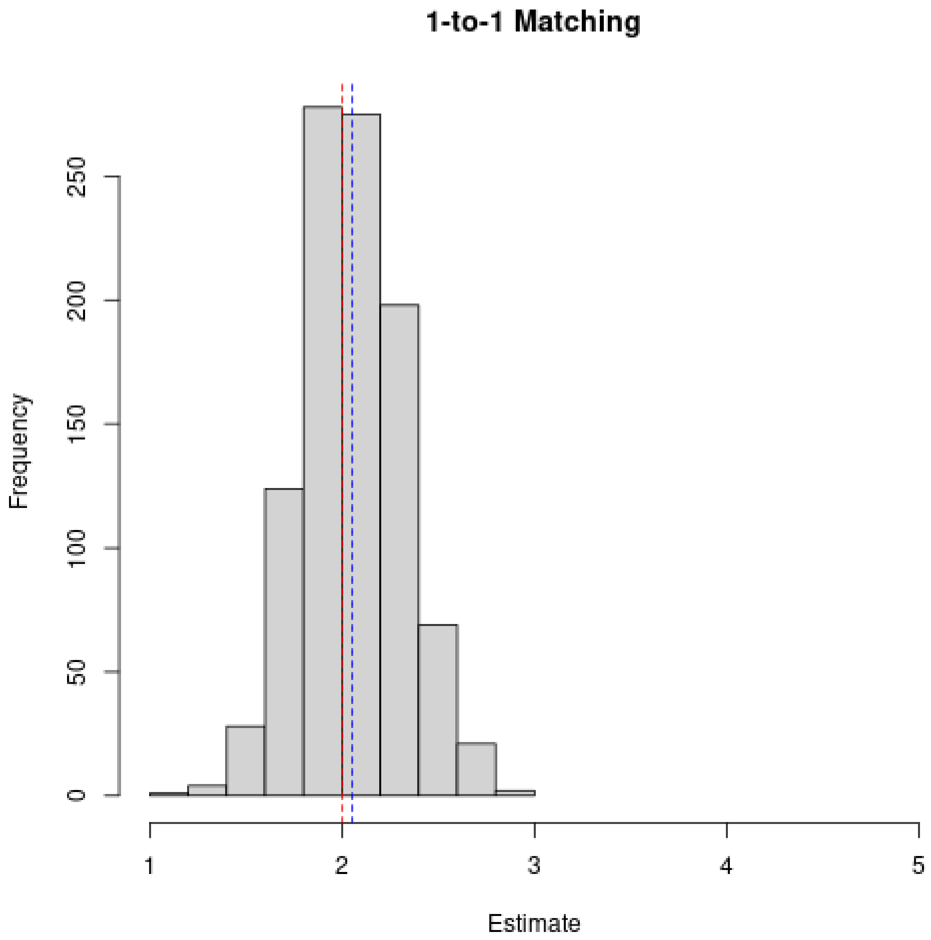
Simulation to show the bias

- What if we just had 1 covariate?



Simulation to show the bias

- Matching bias is a **dimensionality** problem!



Bias-corrected matching

- Instead of substituting in just the average in the matches, Abadie and Imbens (2006) propose a "bias-corrected" imputation
- For $D_i = 1$

$$\widehat{Y_i(1)} = Y_i$$

$$\widehat{Y_i(0)} = \frac{1}{M} \sum_{j \in \mathcal{J}_M(i)} (Y_j + \hat{\mu_0}(X_i) - \hat{\mu_0}(X_j))$$

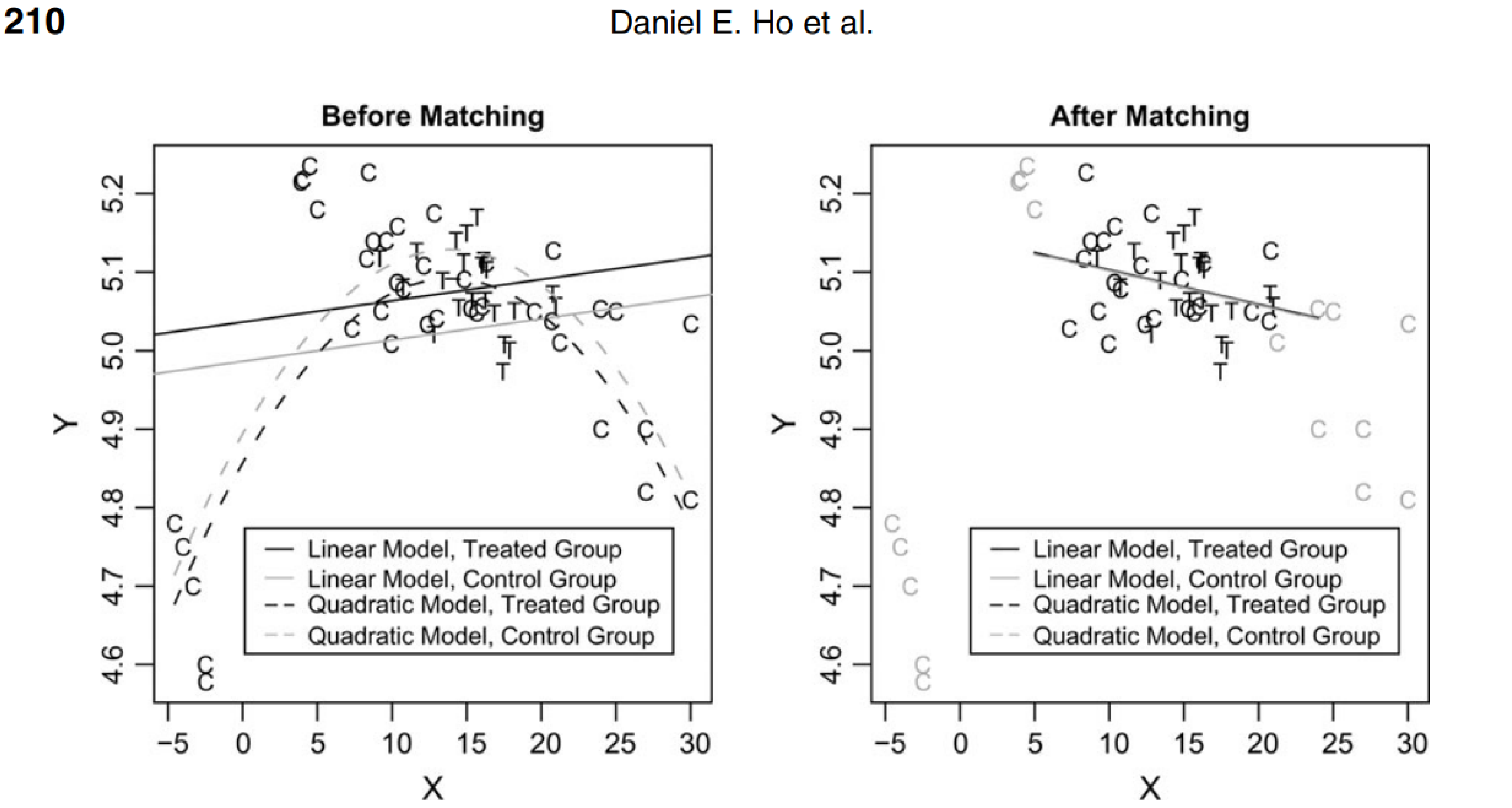
- For $D_i = 0$

$$\widehat{Y_i(0)} = Y_i$$

$$\widehat{Y_i(1)} = \frac{1}{M} \sum_{j \in \mathcal{J}_M(i)} (Y_j + \hat{\mu_1}(X_i) - \hat{\mu_1}(X_j))$$

- **Intuition** -- We combine regression and matching! Regression models adjust for the residual imbalance that matching doesn't solve while matching helps limit the consequences of regression model misspecification.

Matching as pre-processing



210 Daniel E. Ho et al.

Variance estimation

- Unfortunately the standard pairs bootstrap doesn't work for variance estimation (even in the case with zero asymptotic bias) (Abadie and Imbens, 2008)
    - **Intuition**: The regular bootstrap does not preserve the distribution of match counts (the weights) $K_M(i)$.
- Otsu and Rai (2017) show that a *weighted* bootstrap based on the linearized form of the bias-corrected matching estimator *will* work since it conditions on the number of times a unit is matched
- Alternatively, Abadie and Imbens (2006) derive the asymptotic variance of the with-replacement nearest neighbor matching estimator + provide an estimator.
    - `Matching` package implements this estimator.
- In the case of post-matching inference when matching **without replacement**, Abadie and Spiess (2020) show that matching induces dependence within matched sets
    - Solution: Clustered standard errors, clustered on matched set.

Other matching methods

- *Optimal* matching
    - Minimize the **total distance** between treated and the set of chosen (matched) control units
    - Can improve over greedy "nearest-neighbor" matching when matching **without replacement**
- *Full* matching
    - Instead of matching 1-to-1 (or 1-to-many), create subclasses with at least 1 treated and control
    - Minimize the within-subclass distances (optimal matching)
- *Genetic* matching (Diamond and Sekhon, 2013)
    - Find the S^{-1} matrix in the Mahalanobis distance that optimizes some criterion of balance between treated and control groups
    - Essentially trying to find optimal "weights" to put on covariates in the matching algorithm to achieve some global optimum of balance.
    - Use a "genetic" algorithm to search for this optimum (non-linear optimization problem)
- In general, *matching* is just another technique to try to achieve *balance* on the covariates between the treated and control groups
    - To some extent being superseded by other approaches to weighting that don't rely on distance metrics between observations.

Example: Keele et. al. (2017)

- Do minority candidates drive minority voter turnout?
  - **Keele, Shah, White and Kay (2017, JOP) "Black Candidates and Black Turnout: A Study of Viability in Louisiana Mayoral Elections"**
  - Examine mayoral elections in Louisiana from 1988 to 2011 and compare differences in Black turnout among elections with a Black candidate and elections without a Black candidate (all-white candidate slate)
- **Identification challenge**: Black candidates are not randomly assigned to elections!
  - Strategic entry: Black candidates run in districts with larger Black population shares.

Example: Keele et. al. (2017)

- Read in the data

```
turn <- haven::read_dta("assets/match-all.dta")
```

- What's the naive difference-in-means?

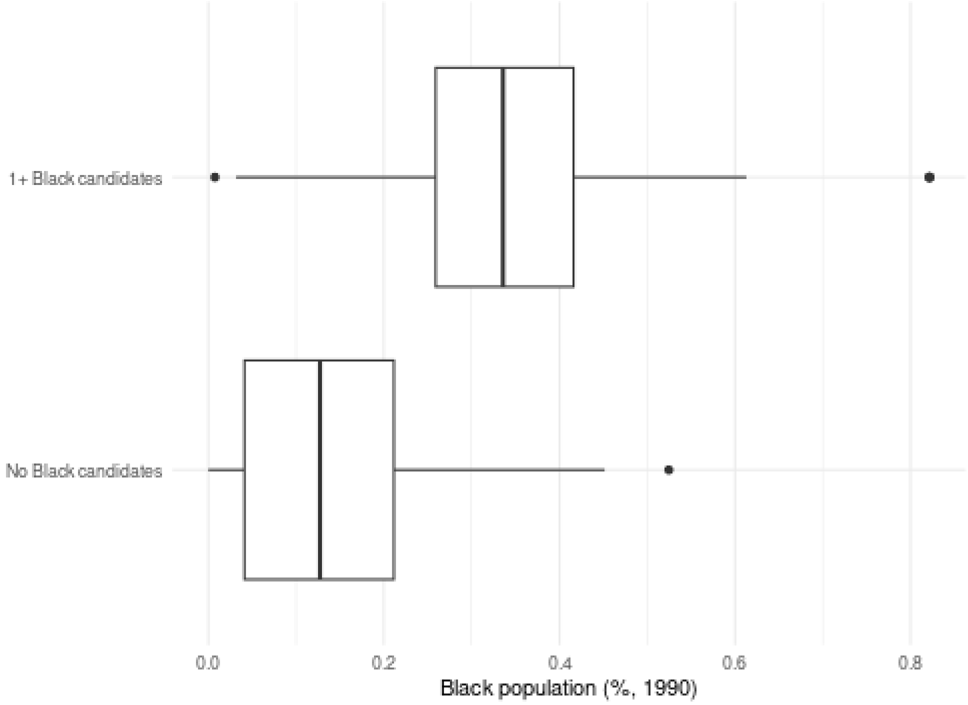
```
lm_robust(black_turnout ~ black, data=turn)
```

```
##              Estimate Std. Error t value  Pr(>|t|) CI Lower CI Upper   DF
## (Intercept)     44.0      0.946   46.47 1.88e-252    42.12     45.8 1004
## black            7.7      1.230    6.26  5.71e-10     5.28     10.1 1004
```

Example: Keele et. al. (2017)

- Visualize the confounding!

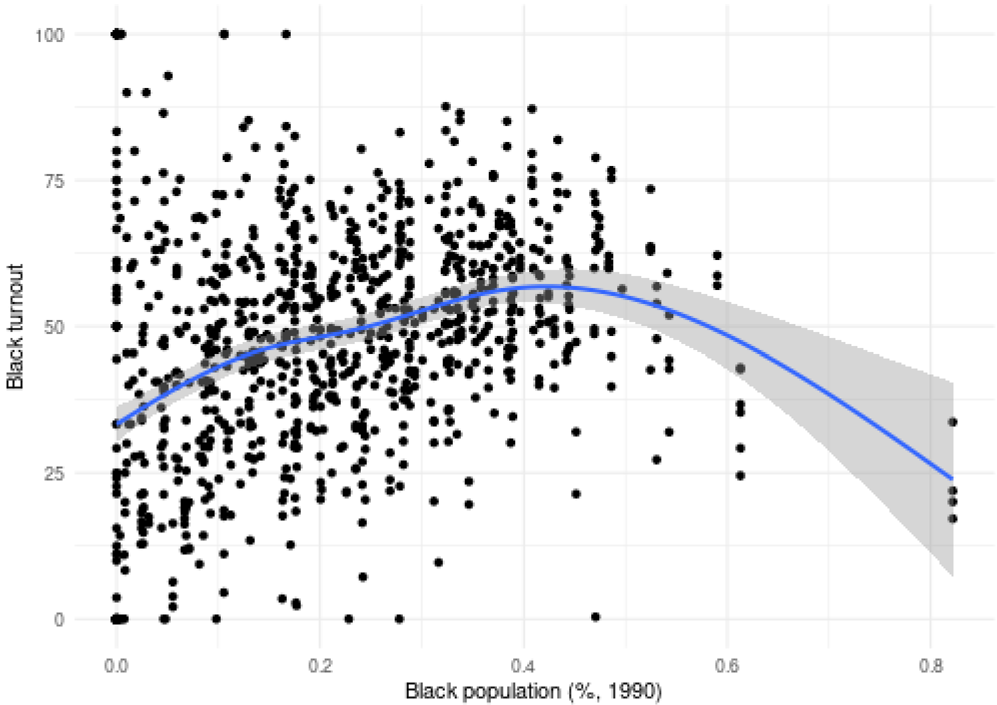
```
turn %>% ggplot(aes(x=blackpop_pct1990, y=as.factor(black))) + geom_boxplot(orientation = "y")
  labs(x = "Black population (%, 1990)", y="") + theme_minimal()
```



Example: Keele et. al. (2017)

- Visualize the confounding!

```
turn %>% ggplot(aes(y=black_turnout, x=blackpop_pct1990)) + geom_point() + geom_smooth() +
  labs(x = "Black population (%, 1990)", y="Black turnout") + theme_minimal()
```



Example: Keele et. al. (2017)

- **Design**: Selection-on-observables with lots of covariates to adjust for:
    - Population, pct. Black, pct. College degree, pct. High school, pct. Unemployed, median income, pct. below poverty line, home rule charter
- We'll focus on replicating their matching approach for **general** election turnout.
    - They also look at **runoff** elections where they believe selection-on-observables is more plausible.
- We'll use standard 1-to-1 nearest neighbor matching
    - The paper itself actually uses a variant of optimal matching that minimizes the total sum of treated-control distances subject to constraints on the covariate-level imbalances.

Example: Keele et. al. (2017)

- We'll implement the Mahalanobis distance 1-to-3 matching estimator

```r
match_results <- Matching::Match(Y = turn$black_turnout, Tr = turn$black,
                         X = turn %>% dplyr::select(year, pop90, blackpop_pct1990,
                                                    college_pct, hs_pct,
                                                    unemp, income, poverty, home),
                         M=1 , Weight = 2, estimand = "ATT")
                        # Weight = 2 = Mahalanobis distance
```

Example: Keele et. al. (2017)

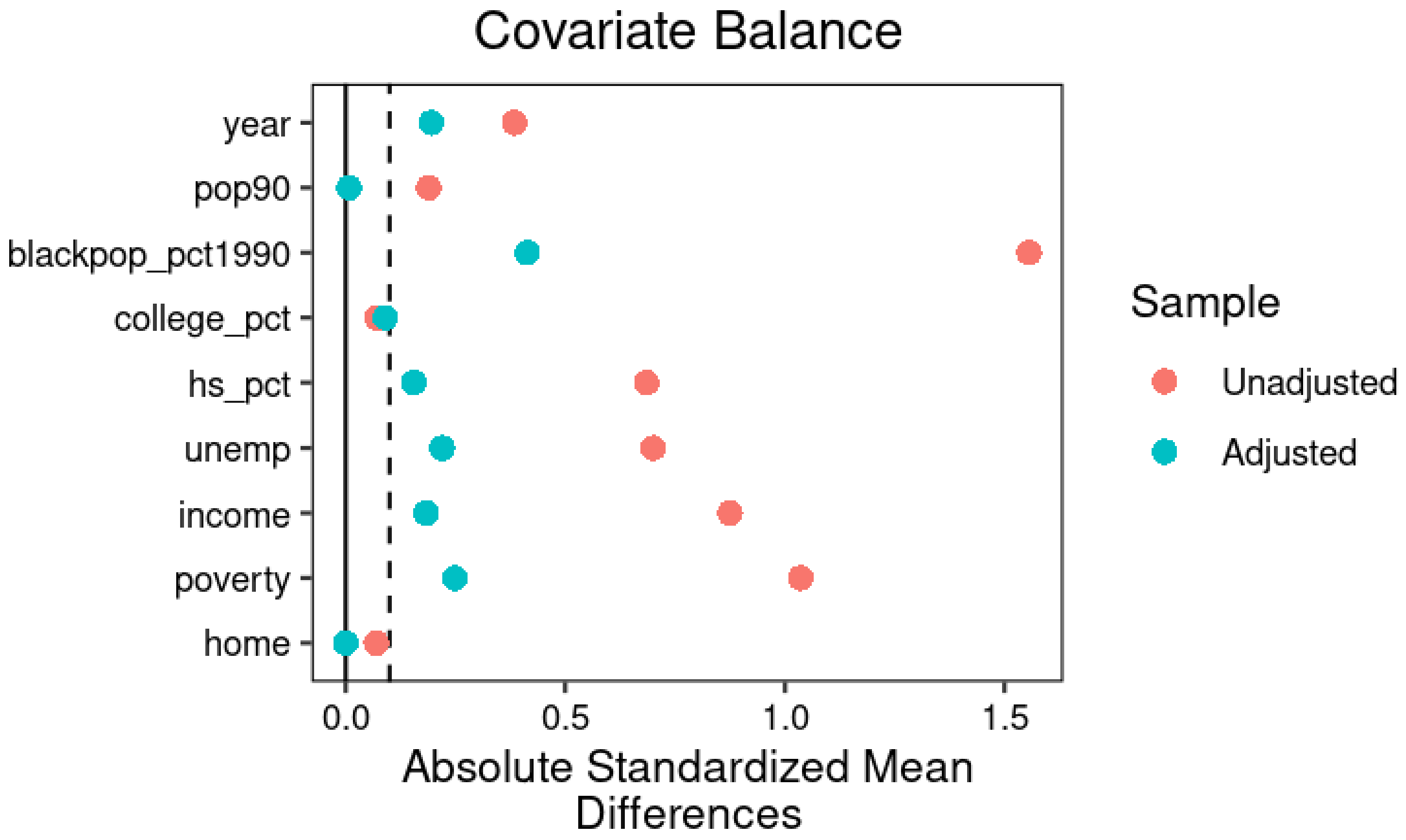
- Results

```
summary(match_results)
```

```
##
## Estimate...  0.48066
## AI SE......  1.3318
## T-stat.....  0.36093
## p.val......  0.71815
##
## Original number of observations..............  1006
## Original number of treated obs...............  356
## Matched number of observations...............  356
## Matched number of observations  (unweighted).  371
```

Example: Keele et. al. (2017)

```
library(cobalt)
cobalt::love.plot(match_results, treat = turn$black, covs = turn %>% dplyr::select(year, pop90,
                              abs=T, binary="std", thresholds= c(m=.1))
```



Example: Keele et. al. (2017)

- What happens if we force **exact** matching on year?

```r
match_results2 <- Matching::Match(Y = turn$black_turnout, Tr = turn$black,
                         X = turn %>% dplyr::select(year, pop90, blackpop_pct1990,
                                                    college_pct, hs_pct,
                                                    unemp, income, poverty, home),
                         exact = c(T, F, F, F, F, F, F, F, F),
                         M=1 , Weight = 2, estimand = "ATT")
                        # Weight = 2 = Mahalanobis distance
```

Example: Keele et. al. (2017)

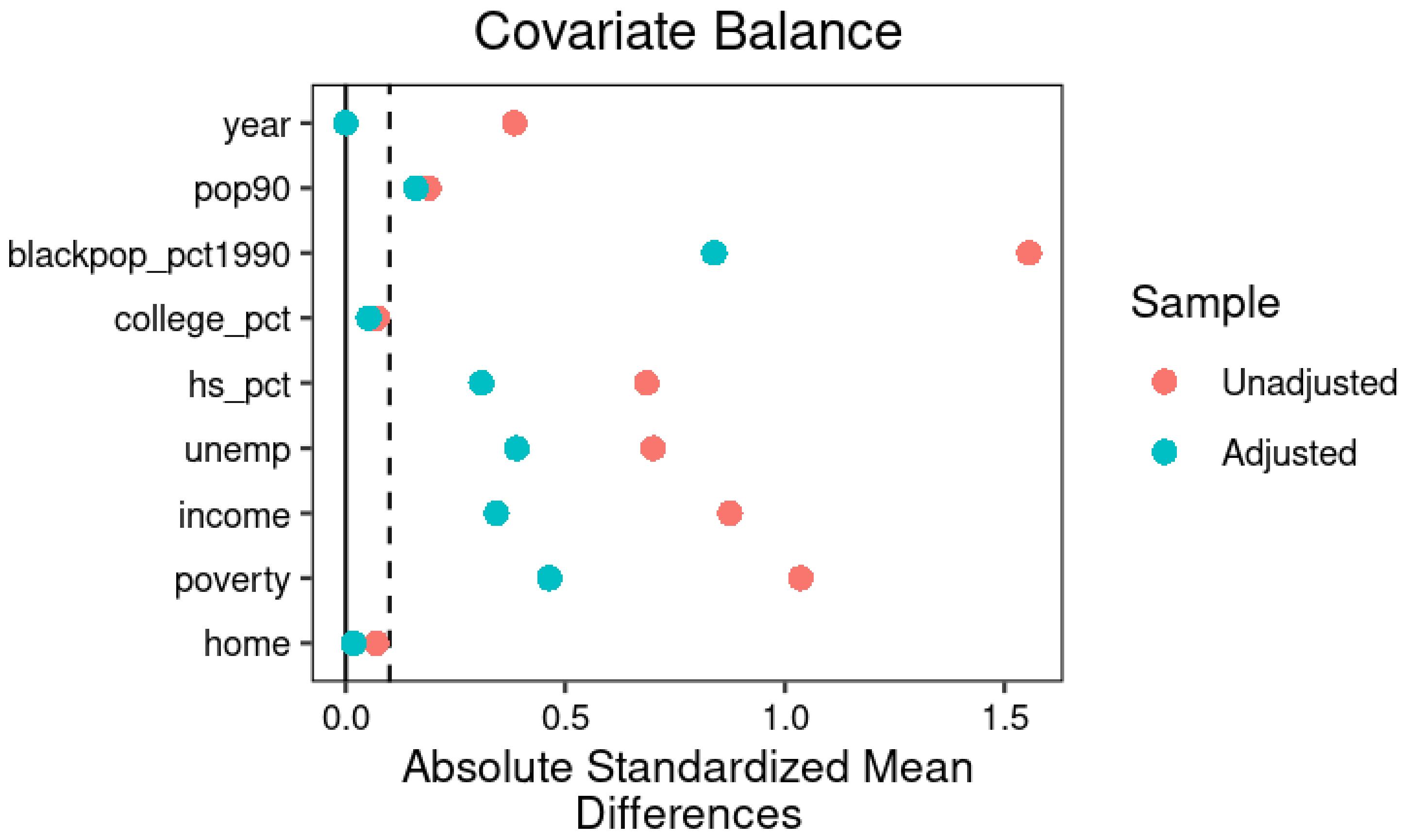
- Results

```
summary(match_results2)
```

```
##
## Estimate...  5.698
## AI SE......  1.4385
## T-stat.....  3.9612
## p.val......  7.4571e-05
##
## Original number of observations..............  1006
## Original number of treated obs...............  356
## Matched number of observations...............  356
## Matched number of observations  (unweighted).  358
##
## Number of obs dropped by 'exact' or 'caliper'  0
```

Example: Keele et. al. (2017)

```r
cobalt::love.plot(match_results2, treat = turn$black, covs = turn %>% dplyr::select(year, pop9
                                     abs=T, binary="std", thresholds= c(m=.1))
```



Example: Keele et. al. (2017)

- Now, what if we add in the bias-correction (the regression)

```
match_results_bc <- Matching::Match(Y = turn$black_turnout, Tr = turn$black,
                         X = turn %>% dplyr::select(year, pop90, blackpop_pct1990,
                                                    college_pct, hs_pct,
                                                    unemp, income, poverty, home),
                         M=1 , Weight = 2, estimand = "ATT", BiasAdjust = T)
                        # Weight = 2 = Mahalanobis distance
```

Example: Keele et. al. (2017)

```
summary(match_results_bc)
```

```
##
## Estimate...  -1.1
## AI SE......  1.3559
## T-stat.....  -0.81122
## p.val......  0.41724
##
## Original number of observations..............  1006
## Original number of treated obs...............  356
## Matched number of observations...............  356
## Matched number of observations  (unweighted).  371
```

Summary

- Matching is a useful tool for reducing covariate imbalance between treated and control groups in a selection-on-observables design
    - **Intuition**: Group together treated and control units with "similar" covariate values
    - Does not depend on any model for the treatment or the outcome
- However, matching is not a universal panacea even if we buy selection-on-observables
    - Still have residual imbalance due to imperfect matches.
    - Matching in high-dimensional space is tricky.
- **Combining** matching and regression
    - Matching is commonly framed as a "pre-processing" step prior to regression to avoid regression imputations that are far from the data.

Regression

Agnostic Regression

- Classical approaches to the linear regression model focus on justifying inference under a particular parametric model

$$Y_i|X_i \sim \mathcal{N}(\beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \dots, \sigma^2)$$

- OLS is BLUE under the Gauss-Markov assumptions. It's the MLE under a normal outcome model.
- But we rarely believe these
  - Homoskedasticity is almost never true
  - Outcomes are binary, count, etc...
- Alternative: Linear regression is the Best Linear Predictor (BLP) of the conditional expectation function (CEF)

Linear Regression

- The regression population parameter β is the solution to the following optimization problem

$$\beta = \arg\min_b E[(Y_i - X_i' b)^2]$$

- We'll estimate it from our sample using OLS:

Justifying linear regression

- One justification for linear regression is when the true population CEF $E[Y_i|X_i]$ is actually linear. In that case,

$$E[Y_i|X_i] = X_i'\beta = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \dots$$

- In this setting, our OLS estimator is unbiased and consistent for the true CEF
- When is the CEF linear?
  - Guaranteed when the model is **saturated**.
- A saturated regression model is one where the number of parameters ($\beta_1, \beta_2, \dots$) equals the number of unique levels of X_i

Fully saturated model

- Consider a model with two binary covariates X_{i1} and X_{i2}.
- Four possible unique values:
  - $E[Y_i|X_{1i} = 0, X_{2i} = 0] = \alpha$
  - $E[Y_i|X_{1i} = 1, X_{2i} = 0] = \alpha + \beta$
  - $E[Y_i|X_{1i} = 0, X_{2i} = 1] = \alpha + \gamma$
  - $E[Y_i|X_{1i} = 1, X_{2i} = 1] = \alpha + \beta + \gamma + \delta$
- The CEF can be written as:

$$E[Y_i|X_{i1}, X_{i2}] = \alpha + \beta X_{i1} + \gamma X_{2i} + \delta X_{1i} X_{2i}$$

- The CEF is linear by construction! Each level of $E[Y_i|X_{1i}, X_{2i}]$ is estimated separately by taking the mean.
  - Note that the *outcome distribution doesn't matter*! Binary outcome? Still a linear **CEF**! Count outcome? Still a linear **CEF**!

Justifying linear regression

- The second justification is that even if the true CEF is not linear, linear regression provides a "best" linear approximation. Why? Recall that the regression parameters solve the optimization problem:

$$\beta = \arg\min_b E[(Y_i - X_i'b)^2]$$

- Among linear approximations to the CEF (ones that have the form $X_i'\beta$), linear regression gives us the approximation that minimizes the mean squared error to the true CEF. In other words

$$\beta = \arg\min_b E[(E[Y_i|X_i] - X_i'b)^2]$$

- So we don't have to *believe* linearity is true to use linear regression - we're still getting some sort of approximation
  - But the approximation *might* be bad, especially when the true CEF is very non-linear.

Regression imputation

- We've typically worked with linear regression as a *prediction* problem: estimating $E[Y_i|X_i]$.
- But how do we use it to estimate $E[Y_i(1)|X_i]$ and $E[Y_i(0)|X_i]$?
  - We need our **identification** assumptions to hold!
- Recall that under selection-on-observables

$$E[Y_i(1)] = E_X\Big[E[Y_i(1)|X_i]\Big] = E_X\Big[E[Y_i|X_i, D_i = 1]\Big]$$

$$E[Y_i(0)] = E_X\Big[E[Y_i(0)|X_i]\Big] = E_X\Big[E[Y_i|X_i, D_i = 0]\Big]$$

- So what we need to do to estimate the ATE is:
  1. Fit a regression model in the treated group to estimate $E[Y_i(1)|X_i]$
  2. Fit a regression model in the control group to estimate $E[Y_i(0)|X_i]$
  3. Average the estimates from that model over the sample distribution of X_i

Regression imputation

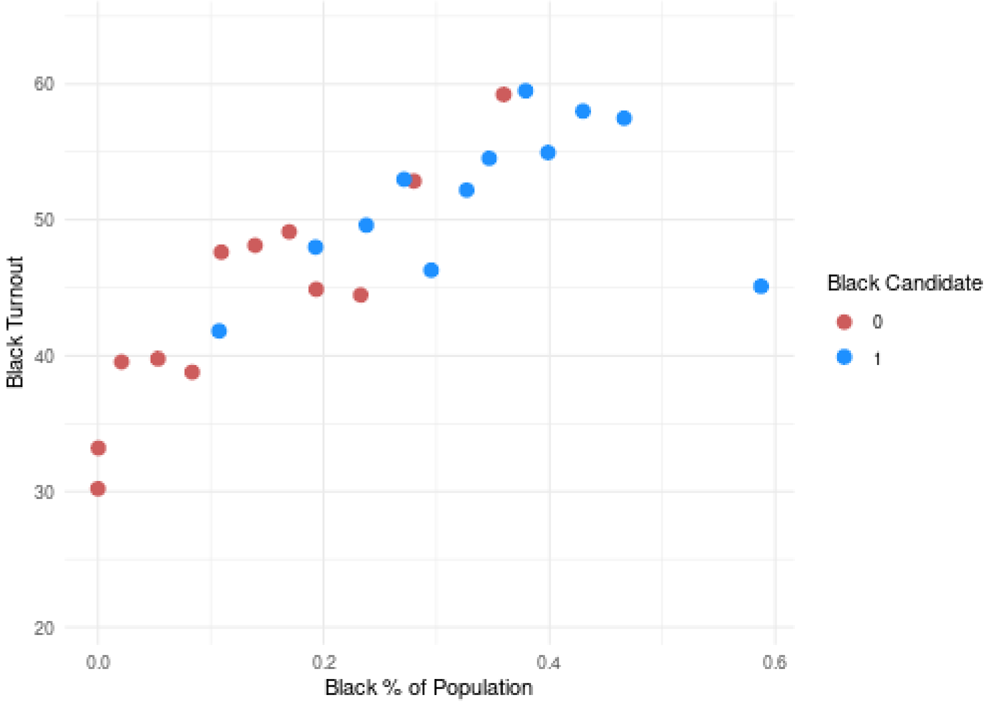
- Regression can be thought of as an **imputation** estimator:

$$\hat{\tau}_{\text{reg}} = \frac{1}{N} \sum_{i=1}^{N} \hat{\mu}_1(X_i) - \hat{\mu}_0(X_i)$$

- $\hat{\mu}_d(X_i) = \hat{E}[Y_i(d)|X_i]$ is our prediction from a regression of Y_i on X_i in either treated or control group
- Note that we have made **no** restrictions on individual treatment effect heterogeneity!
- Unbiased and consistent for the ATE if we've specified the true regression model for $E[Y_i(d)|X_i]$ correctly

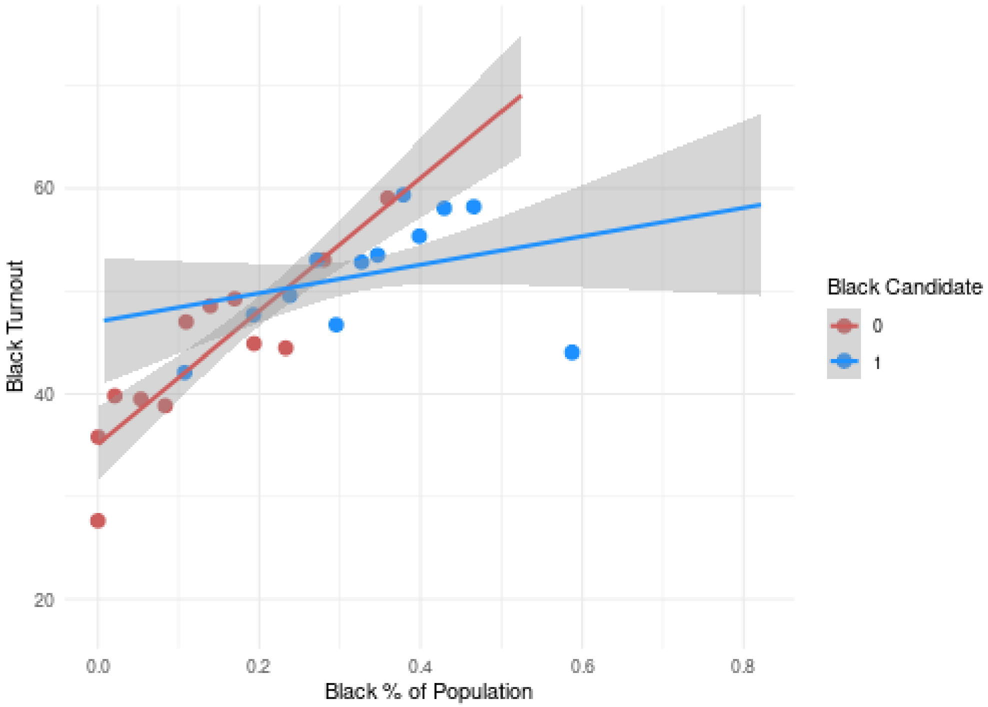
Regression imputation in practice

- Let's take a look again at our Keele et. al. (2017). data. Let's consider first adjusting for a single covariate: black share of the population.
  - This is arguably the biggest confounding story
- Before running any regression, it's very useful to plot the data using a **binned scatterplot**



Regression imputation in practice

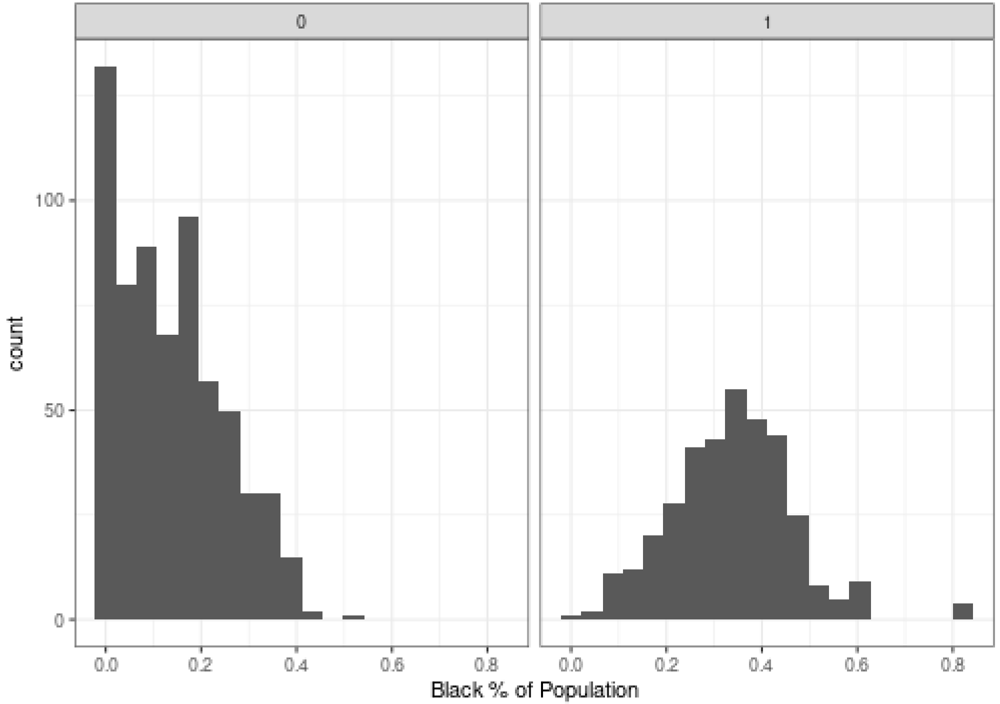
- Now let's fit a linear regression in each treatment condition



- Note the approximation works kinda well in the range where we have a *lot* of data, but is a lot worse at the tails.

Regression imputation in practice

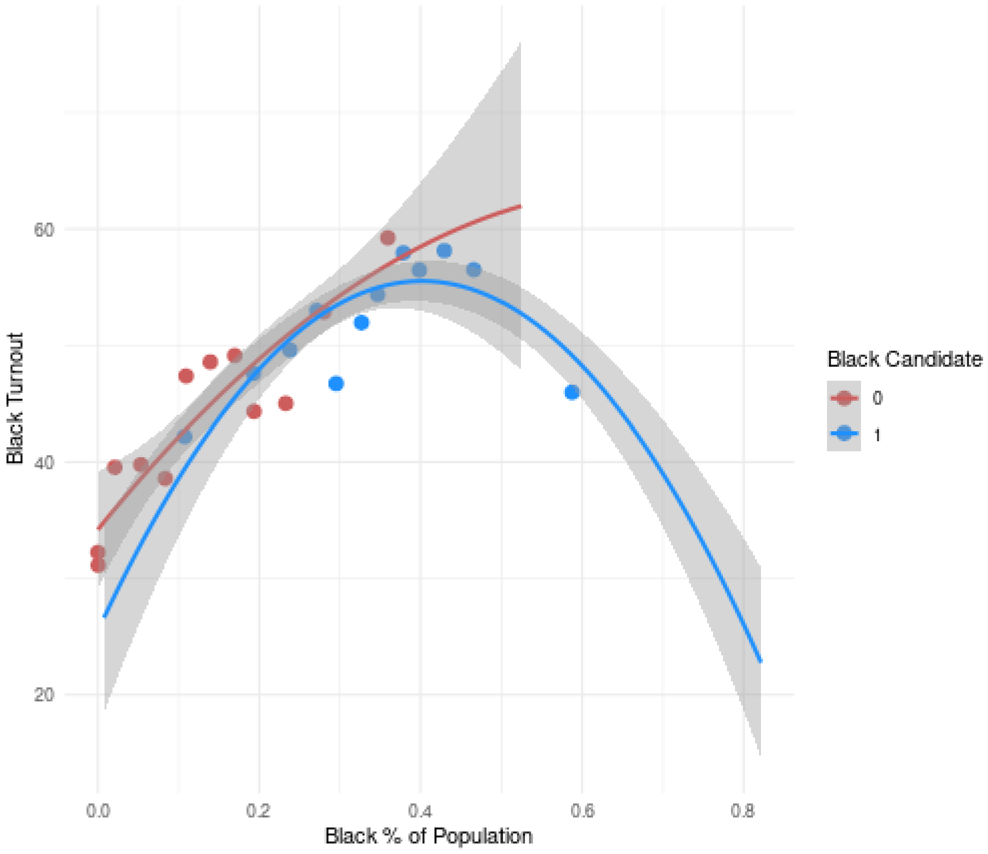
- If we have **zero** overlap between treatment and control in the covariates, our counterfactual predictions will be based on extreme extrapolations -- lots of model sensitivity



- Here it's not too bad

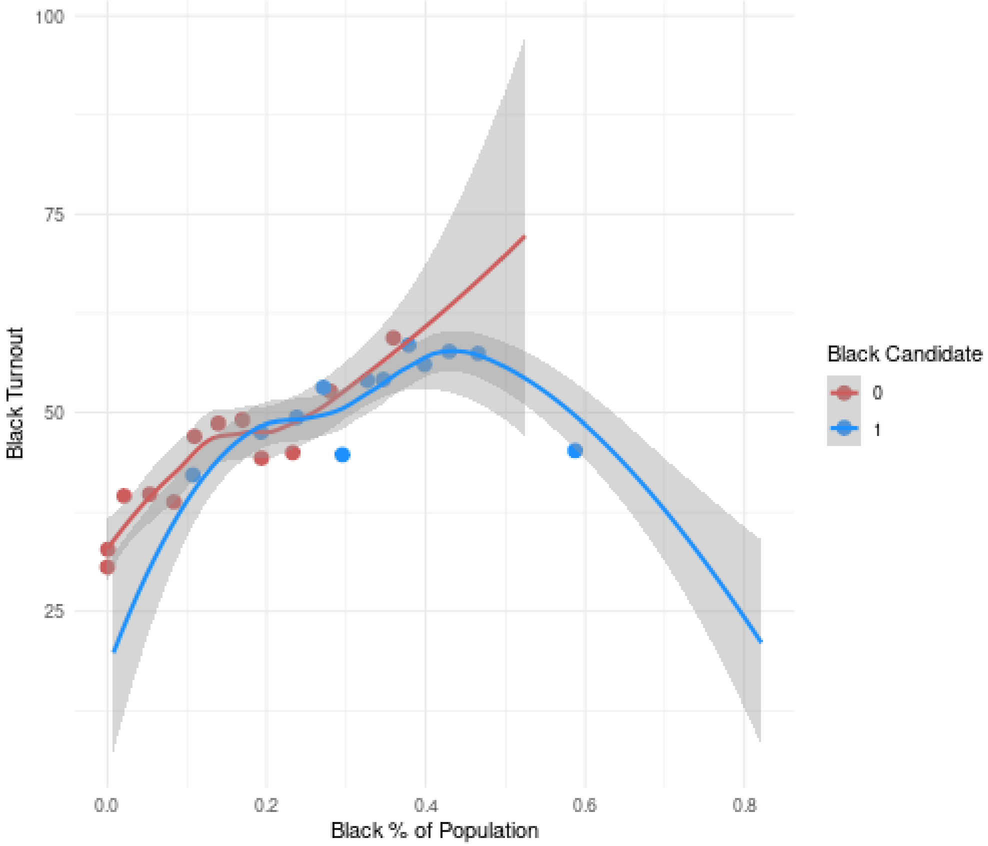
Regression imputation in practice

- What happens if we use a quadratic fit?



Regression imputation in practice

- Local linear regression smoothers are also popular



Regression imputation in practice

- Now let's actually construct our imputation estimator for the ATE using all of the covariates

```
# Fit regression models
mu_1 <- lm_robust(black_turnout ~ pop90 + blackpop_pct1990 + I(blackpop_pct1990^2) +
                    unemp + college_pct + hs_pct + unemp + income + poverty + home +
                    as.factor(year),  data = turn %>% filter(black == 1))
mu_0 <- lm_robust(black_turnout ~ pop90 + blackpop_pct1990 + I(blackpop_pct1990^2) +
                    unemp + college_pct + hs_pct + unemp + income + poverty + home +
                    as.factor(year),  data = turn %>% filter(black == 0))

# Predict onto the sample
turn$y1 <- predict(mu_1, newdata=turn)
turn$y0 <- predict(mu_0, newdata=turn)

# Point estimate
point <- mean(turn$y1) - mean(turn$y0)
point
```

```
## [1] 0.921
```

- Inference?
    - Bootstrap...
    - ...or do this all in one regression!

Regression imputation in practice

- Recall that the Lin (2013) estimator (with full interactions w/ treatment) is essentially fitting two regressions

```
reg_ate <- lm_lin(black_turnout ~ black,
                  covariates = ~ pop90 + blackpop_pct1990 + I(blackpop_pct1990^2) +
                    unemp + college_pct + hs_pct + unemp + income + poverty + home +
                    as.factor(year),
                  data = turn)

tidy(reg_ate) %>% filter(term == "black") %>% dplyr::select(term, estimate, std.error, p.value)
```

```
##    term estimate std.error p.value
## 1 black    0.921      2.25   0.683
```

- The coefficient on treatment captures the difference between these two regression functions averaged over the distribution of the data
    - In other words, the imputation estimator!

Regression with constant effects

- Often you will see researchers estimate a single regression model with the form:

$$E[Y_i|D_i, X_i] = \tau D_i + X_i'\beta$$

- Does τ identify the ATE?
    - We need to assume the model is correct (e.g. no treatment-covariate interactions)
    - **And** we need to assume constant treatment effects!
- Even if the model for the outcome is correct, under effect heterogeneity, the regression coefficient τ is not the ATE.
    - "Regression weighting problem" (Samii and Aronow, 2012)

Regression weighting

- Suppose the potential outcomes can be written as

$$Y_i(d) = Y_i(0) + \tau_i \times d$$

- The ATE is $E[\tau_i] = \tau$

- Suppose we then fit our usual linear regression model

$$Y_i = \alpha + \tau_{\mathrm{R}} D_i + X_i' \beta + \epsilon_i$$

- Will our estimate of $\hat{\tau_{\mathrm{R}}}$ equal $E[\tau_i]$?
    - No!

Frisch-Waugh-Lovell

- An important theorem (that you will spend more time with in Linear Models) is that the coefficients in a multiple linear regression can be written by **partialling out** correlations with the other covariates.
    - The **Frisch-Waugh-Lovell** theorem!
- Consider estimating the following by OLS:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \epsilon_i$$

- The standard least-squares estimator for β_1 can be expressed as the ratio of the sample covariance of X_{i1} and Y_i and the variance of X_{i1}

$$\hat{\beta}_1 = \frac{\widehat{Cov}(Y_i, X_{i1})}{\widehat{Var}(X_{i1})} \to \frac{Cov(Y_i, X_{i1})}{Var(X_{i1})}$$

Frisch-Waugh-Lovell

- Now what happens with a multiple linear regression - how do we write β_1 when the model is

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \epsilon_i$$

- It turns out that β_1 can be written in terms of a bivariate regression

$$\hat{\beta}_1 = \frac{\widehat{Cov}(\tilde{Y}_i, \tilde{X}_{i1})}{\widehat{Var}(\tilde{X}_{i1})} \to \frac{Cov(\tilde{Y}_i, \tilde{X}_{i1})}{Var(\tilde{X}_{i1})}$$

- $\tilde{Y_i}$ is the residual from a regression of Y_i on all of the other covariates (here, just X_{i2})
- $\tilde{X_{i1}}$ is the residual from a regression of X_{i1} on all of the other covariates (X_{i2} here)
- And for β_1, it's equivalent to write it as a regression of just Y_i on $\tilde{X_{i1}}$

Regression weighting

- Under the Frisch-Waugh-Lovell theorem. we can write an expression for the treatment coefficient as

$$\tau_{\text{R}} = \frac{Cov(Y_i, \tilde{D}_i)}{Var(\tilde{D}_i)}$$

where $\tilde{D}_i$ is the residual from a regression of D_i on all of the other covariates.

- In this case, that's just:

$$\tau_{\text{R}} = \frac{Cov(Y_i, D_i - E[D_i|X_i])}{Var(D_i - E[D_i|X_i])}$$

Regression weighting

- We can re-arrange terms (see Samii and Aronow, 2016 for the math) to get an expression for the weights placed on individual τ_i

$$\tau_{\text{R}} = \frac{E[w_i \tau_i]}{E[w_i]}$$

where $w_i = (D_i - E[D_i|X_i])^2$

- **Intuition** - Units whose treatment status is *poorly predicted* by the covariates get more weight.
    - Why? Because OLS is a **minimum-variance** estimator - estimates are more precise for strata where there is more residual variation in D_i
- The typical multiple regression estimator does not recover the ATE under effect heterogeneity.

Regression weighting

```
full_reg <- lm_robust(black_turnout ~ black + pop90 + blackpop_pct1990 + I(blackpop_pct1990^2)
                    unemp + college_pct + hs_pct + unemp + income + poverty + home +
                    as.factor(year), data=turn)
tidy(full_reg) %>% filter(term == "black") %>% dplyr::select(term, estimate, std.error, p.value
```

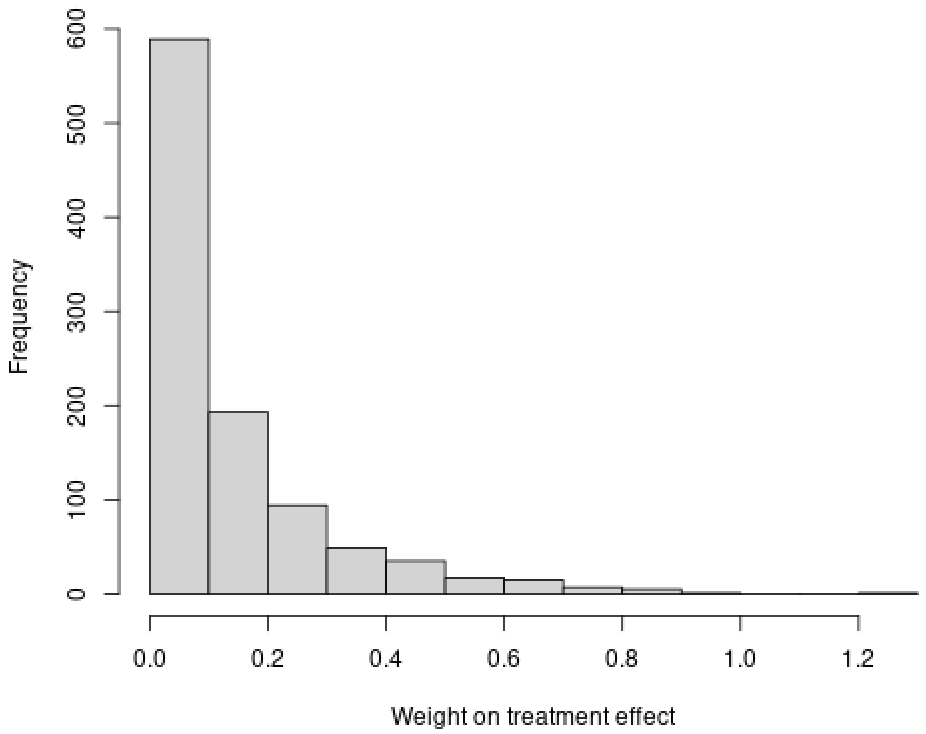
```
##    term estimate std.error p.value
## 1 black     1.87      1.23   0.129
```

```
# Get the weights
resid_reg <- lm_robust(black ~ pop90 + blackpop_pct1990 + I(blackpop_pct1990^2) +
                    unemp + college_pct + hs_pct + unemp + income + poverty + home +
                    as.factor(year), data=turn)
turn$black_resid <- turn$black - resid_reg$fitted.values
# Show FWL working
partial_reg <- lm_robust(black_turnout ~ black_resid, data=turn)
tidy(partial_reg) %>% filter(term == "black_resid") %>% dplyr::select(term, estimate)
```

```
##          term estimate
## 1 black_resid     1.87
```

```
# Weights on individual treatment effects
turn$black_w <- turn$black_resid^2
```

Regression weighting



Regression overview

- Regression adjusts for X_i by directly modeling $E[Y_i(1)|X_i]$ and $E[Y_i(0)|X_i]$.
  - Linear regression is actually quite flexible -- don't need the full Gauss-Markov assumptions to justify it as a "best approximation" to the CEF.
- **Recommended**
  - Fit two separate models in treatment/control and use them to impute the P.O. for each unit i
  - Can bootstrap to get the sampling variance
  - Or fit a single regression model with all treatment-covariate interactions
- **Careful**
  - Regression estimators aren't great when overlap is poor
  - Regression doesn't necessarily assign a "representative" weight to each observation

Beware the Table 2 fallacy

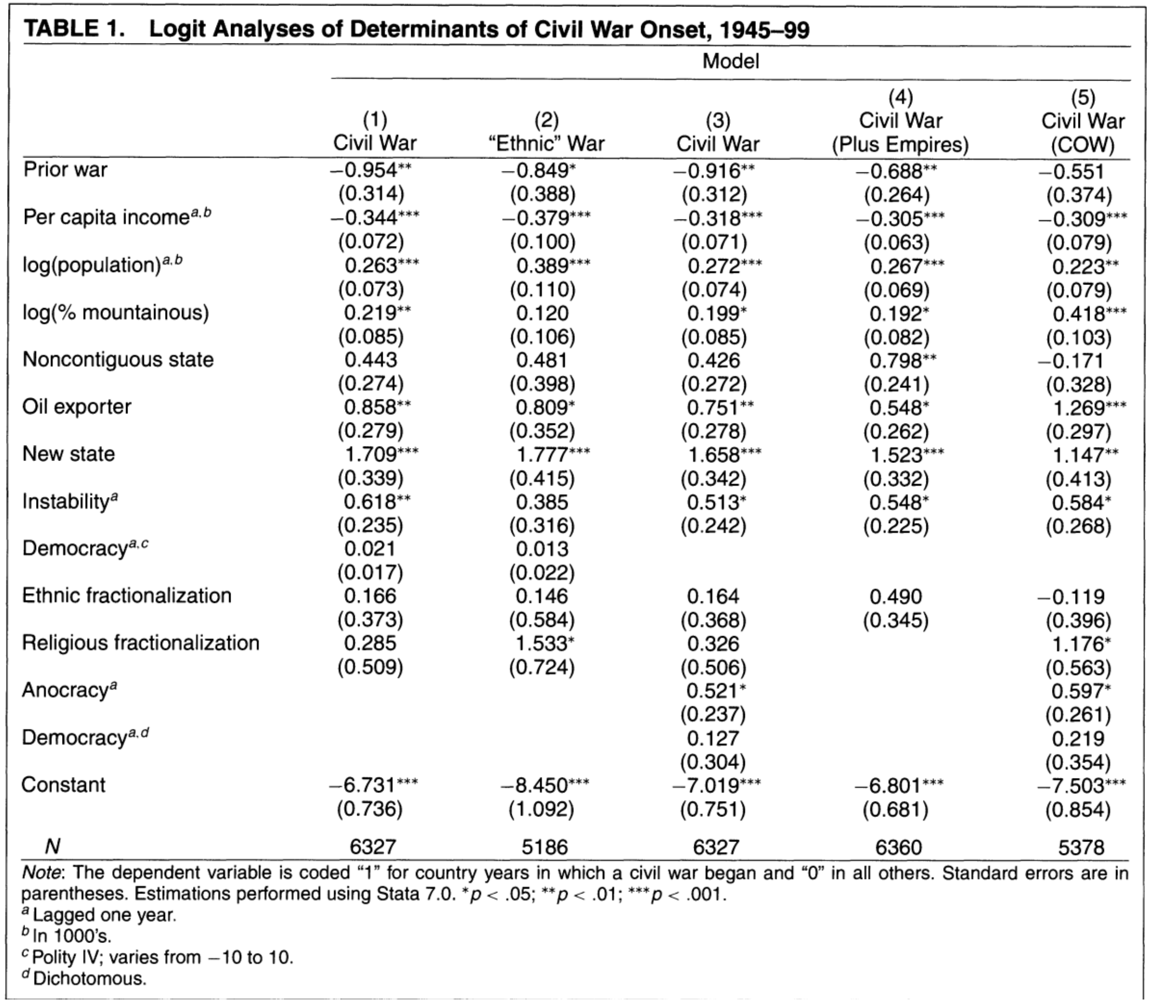


TABLE 1. Logit Analyses of Determinants of Civil War Onset, 1945–99

	Model				
	(1) Civil War	(2) “Ethnic” War	(3) Civil War	(4) Civil War (Plus Empires)	(5) Civil War (COW)
Prior war	−0.954**	−0.849*	−0.916**	−0.688**	−0.551
	(0.314)	(0.388)	(0.312)	(0.264)	(0.374)
Per capita income[a,b]	−0.344***	−0.379***	−0.318***	−0.305***	−0.309***
	(0.072)	(0.100)	(0.071)	(0.063)	(0.079)
log(population)[a,b]	0.263***	0.389***	0.272***	0.267***	0.223**
	(0.073)	(0.110)	(0.074)	(0.069)	(0.079)
log(% mountainous)	0.219**	0.120	0.199*	0.192*	0.418***
	(0.085)	(0.106)	(0.085)	(0.082)	(0.103)
Noncontiguous state	0.443	0.481	0.426	0.798**	−0.171
	(0.274)	(0.398)	(0.272)	(0.241)	(0.328)
Oil exporter	0.858**	0.809*	0.751**	0.548*	1.269***
	(0.279)	(0.352)	(0.278)	(0.262)	(0.297)
New state	1.709***	1.777***	1.658***	1.523***	1.147**
	(0.339)	(0.415)	(0.342)	(0.332)	(0.413)
Instability[a]	0.618**	0.385	0.513*	0.548*	0.584*
	(0.235)	(0.316)	(0.242)	(0.225)	(0.268)
Democracy[a,c]	0.021	0.013			
	(0.017)	(0.022)			
Ethnic fractionalization	0.166	0.146	0.164	0.490	−0.119
	(0.373)	(0.584)	(0.368)	(0.345)	(0.396)
Religious fractionalization	0.285	1.533*	0.326		1.176*
	(0.509)	(0.724)	(0.506)		(0.563)
Anocracy[a]			0.521*		0.597*
			(0.237)		(0.261)
Democracy[a,d]			0.127		0.219
			(0.304)		(0.354)
Constant	−6.731***	−8.450***	−7.019***	−6.801***	−7.503***
	(0.736)	(1.092)	(0.751)	(0.681)	(0.854)
N	6327	5186	6327	6360	5378

Note: The dependent variable is coded “1” for country years in which a civil war began and “0” in all others. Standard errors are in parentheses. Estimations performed using Stata 7.0. *$p < .05$; **$p < .01$; ***$p < .001$.

[a] Lagged one year.

[b] In 1000’s.

[c] Polity IV; varies from −10 to 10.

[d] Dichotomous.

Beware the Table 2 fallacy

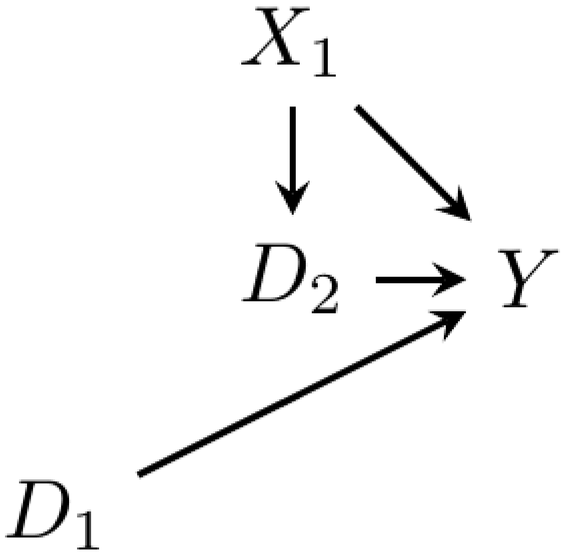
- It is best to design a regression to estimate **one** effect.
- Don't read all the coefficients from a regression as causal!
- **Intuition**: We've selected X_i that affect our treatment D.
  - So the coefficient on X_i in our regression will - by construction - suffer from post-treatment bias since we're controlling for D in that same regression!
  - We also justified our choices of X_i based on what affects D, but the confounders for D are not the same as the confounders for some other X
  - Effect heterogeneity also messes this up -- remember FWL!
- For more, see

> > Westreich, Daniel, and Sander Greenland. "The table 2 fallacy: presenting and interpreting confounder and modifier coefficients." American journal of epidemiology 177.4 (2013): 292-298.

> Hünermund, Paul, and Beyers Louw. "On the Nuisance of Control Variables in Causal Regression Analysis." Organizational Research Methods (2023)

Beware the Table 2 fallacy

- Depending on the DAG, you might get multiple effects out of a single analysis, but it's a **very** specific set of assumptions:



Overview

- All of the methods we've discussed only work if our identification assumption is correct
- **Selection-on-observables**:
    - $\{Y_i(1), Y_i(0)\} \perp\!\!\!\perp D_i | X_i$
- How do we know if X_i is the right set of covariates?
    - Theory! + (some) knowledge of the treatment assignment process.
    - DAGs as a tool for synthesizing theory about known effects
- Once we think selection-on-observables is a plausible **identification strategy**, then we have some choice in our **estimation** approach:
    - Discrete covariates: Just stratify!
    - Concerned about modelling assumptions? Maybe match first?
    - IPTW + Regression: More recent methods combine the two for "double-robustness"
    - Consider less parametric methods for the models (kernels, local linear regression, BART, GAMs)
- **Always look at your data!**