



# Propensity score matching with clustered data in Stata

Bruno Arpino Pompeu Fabra University bruno.arpino@upf.edu https://sites.google.com/site/brunoarpino

11<sup>th</sup> Spanish Stata Conference 2018 Barcelona, 24 de octubre de 2018

# Outline

- Brief intro to Propensity Score Matching (PSM) for estimation of causal effects in observational studies
- PSM with **clustered** (multilevel, hierarchical) data
- PSM in Stata
  - Available routines
  - How to implement PSM with clustered data

Do-file and dataset to replicate the analyses in these slides can be found at: <u>https://sites.google.com/site/brunoarpino/software</u>

## Motivating case study (1/3)

- Goal: estimating the causal effect of doing homeworks on mathematical proficiency
- We use a subset of the National Education Longitudinal Study of 1988 (NELS-88), a nationally representative, longitudinal study of 8th graders in 1988 in the US
- Our data is a subsample of the original full NELS-88 dataset provided by Kraft and de Leeuw (1998)

## Motivating case study (2/3)

- **Treatment**: T = 1 for students that spend at least 1 hour doing math homeworks per week; 0 otherwise
- **Outcome**: Y, is the score on a math test
- The dataset contains 260 students from 10 schools and several potential confounders on both students (X) and schools (Z)

## Motivating case study (3/3)

- Selection mechanism: what are the factors influencing time spent doing homework (that may also influence math proficiency)? For the sake of illustration we only consider:
- Individual-level: ses (a standardised continuous measure of family socio-economic status), male (1 = male; 0 = female) and white (1 = white; 0 = other race)
- **School-level**: *public* (1 = public schools; 0 = private)

## **Potential outcome framework**

- Let T be the binary treatment indicator:
  - = 1 at least 1 hour doing math homeworks per week;
  - = 0 otherwise
- Let Y(1) and Y(0) denote the potential outcomes, i.e. math score we would observe if students were assigned to the treatment or control group, respectively
- Causal estimand of interest: ATT = E[Y(1) Y(0) | T = 1]
- Y(0) is always unobserved for treated students (T = 1)

# Propensity score (PS) methods

- Identifying assumptions:
  - Y(1), Y(0)  $\perp$  T | X, Z (unconfoundedness)
  - 0 < P (T=1|X, Z) < 1 (overlap)
- PS:  $e(X) \equiv Pr\{T = 1 | X, Z\} = E\{T|X, Z\}.$
- Rosenbaum and Rubin (1983):
  - the propensity score is a balancing score, i.e., X,  $Z \perp T \mid e(X, Z)$
  - if unconfoundedness holds, then Y(1), Y(0)  $\perp$  T | e(X, Z)
- These results justify matching / stratification / weighting on e(X, Z) instead than on (X, Z)

#### **PSM** as a two-step procedure

- Design phase: match similar treated and control individuals to make them as similar as possible in terms of (X, Z)
- Outcome phase: estimate causal effects on the matched data
- It reduces model dependence (extrapolation; Drake, 1993)
- It increses objective causal inference (Rubin, 2008)
- Matching as a data pre-processing (Ho et al., 2007)

### **Clustered data structures**

- Very common in many fields (patients into hospitals, individuals into geographical areas, students into schools)
- Few methodological and applied works exist in the case of clustered data
- In clustered data bias can arise from omitted individual and/or cluster-level confounders
- How should we apply PS methods to these data?
- How can we exploit knowledge on clusters' memberships?

## Existing studies with clustered data

- Arpino and Mealli (2011)
  - Show the benefit of using random or fixed effects models for the estimation of the propensity score to reduce the bias due to unmeasured cluster-level variables in PS matching (PSM)
  - Focus on high number of small clusters
- Thoemmes and West (2011) and Li et al (2013) considered stratification and re-weighting using PS, respectively

# Arpino and Cannas (2016)

- Unbalanced data structure with both big and small clusters
- We compare different approaches:

Approaches	PS model	Matching	
Naïve (NV)	Single-level logit	Pooled	
Within (W)	Single-level logit	Within-cluster	
Preferential (PW)	Single-level logit	"Preferential" within-cluster	
Random-effects (RE) Random-effect logit		Pooled	
Fixed-effects (RE) Fixed-effect logit		Pooled	

• R package: CMatching

### «Naïve» approach

• It ignores the clustered structure in both PS estimation:

$$\text{logit} (e_{ij}) = \alpha_0 + X_{ij}\beta \qquad (1)$$

• and matching

$$A_{rj} = \{kj' \in I_0 : \hat{e}_{kj'} = \min_{kj' \in I_0} \left| \hat{e}_{rj} - \hat{e}_{kj'} \right| < 0.25 \hat{\sigma}_e \} \quad (2)$$

We use one-to-one nearest neighbor matching within a caliper of 0.25 standard deviations of the estimated PS (with replacement)

# **Estimating the ATT**

• The matched dataset is built as the subset of treated and control units that have been matched:

$$M = \{rj : A_{rj} \neq \emptyset\} \cup \left\{\bigcup_{rj} A_{rj}\right\} \quad (3)$$

• and the ATT is estimated on this set using:

$$A\hat{T}T = \frac{1}{card(M)} \left\{ \sum_{r_j \in I_1 \cap M} \left( Y_{r_j} - \sum_{k_{j'}} Y_{k_{j'}} w(r_j, k_{j'}) \right) \right\}$$
(4)

# «Within» approach

 Uses the same PS model than method A (2) but adjusts for clustering in the implementation of the matching that is forced to be within-cluster:

$$A_{rj} = \{kj' \in I_0 : \hat{e}_{kj'} = \min_{kj' \in I_0} \left| \hat{e}_{rj} - \hat{e}_{kj'} \right| < 0.25 \hat{\sigma}_e; \ j = j'\}$$
(5)

 Automatically guarantees that all cluster-level variables are perfectly balanced. But, balance of individual-level variables could be worse than with the "Naïve" approach. Also the no. of unmatched units will be higher.

## «Preferential» approach

- Tries to combine the benefits of the previous two approaches ("Naïve" and "Within").
- Starts by searching control units within-cluster (according to (5)). If none is found, control units are searched in other clusters (according to (2)).
- It is expected to improve the balancing of cluster-level variables with respect to the "Naïve" approach and reduces the loss of units compared to the "Within" approach.

## «Random-effects» and «Fixedeffects» approaches

• They keep clustering into account in the estimation of the propensity score:

$$\text{logit } (e_{ij}) = \alpha_j + X_{ij}\beta \qquad (6)$$

by estimating cluster-specific random or fixed intercepts, respectively (Arpino and Mealli, 2011).

#### **Simulation results (1/2)**



#### Simulation results (2/2)



Note:  $\beta z$ , overall sampe size, etc. are kept fixed. Z is unobserved.

# Implementing matching in Stata

#### • **psmatch2** (Leuven and Sianesi 2003)

- PSM and covariate matching
- severalalgorithms (nn and caliper matching (with and w/o replacement), kernel, radius, local linear matching
- common support plots (psgraph) and covariate imbalance testing (pstest)
- standard errors obtained using bootstrap methods or variance approximation
- **nnmatch** (Abadie, Drukker, Herr, and Imbens 2004)
  - nearest neighbour matching with different distance metrics (replacement allowed)
  - allows **exact matching** (or as close as possible) on a subset of variables
  - allows for bias correction
  - sample or population variance, with or w/o assuming a constant treatment effects

# Implementing matching in Stata

#### • teffects (built-in)

- PSM (some of the features of psmatch2). It does not offer balance checks
- covariate matching (nnmatch)
- it calculates standard errors that take into account that propensity scores are estimated. Theoretical results for clustered data are not yet available
- **kmatch** (Jann, 2017)
  - PSM and covariate matching (nn, kernel, ridge)
  - several options for optimal bandwidth selection; exact matching; bias adjustment
  - tools for common support and balance diagnostics
- **Cem** (lacus, King and Porro 2008)
  - coarsened exact matching
- There is no command designed specifically for clustered data

# **PSM in Stata with clustered data**

Approaches	PS model	Matching
Naïve (NV)	Single-level logit ( <b>logit</b> )	Pooled ( <b>psmatch2</b> ; nnmatch)
Within (W)	Single-level logit ( <b>logit</b> )	Within-cluster (cycle on <b>psmatch2</b> ; nnmatch with exact option)
Preferential (PW)	Single-level logit ( <b>logit</b> )	" <b>Preferential</b> " within-cluster (ad hoc procedure based on <b>psmatch2</b> or nnmatch)
Random-effects (RE)	Random-effect logit (e.g., <b>xtmelogit</b> )	Pooled (psmatch2; nnmatch)
Fixed-effects (RE)	Fixed-effect logit (e.g., <b>logit +</b> clusters' dummies)	Pooled (psmatch2; nnmatch)

Outcome analysis should account for clustering (robust se)

<pre>66 67 68 69 69 69 69 69 69 69 69 69 69 69 69 69</pre>	match	ing.do ×	
<pre>70 *** If not installed the following packages should be installed *** soc install pantch, replace 71 ** soc install numatch, replace 72 ** soc install numatch, replace 73 * Sstimate the propensity score model 74 ************************************</pre>	69		
<pre>11 ** socinstall pamatch2, replace 2** socinstall numatch, replace 3************************************</pre>	70	*** If not installed the following packages should be installed	
<pre>72 *** soc install numatch, replace 74 75 76 77 77 77 77 77 77 77 77 77 77 77 77</pre>	71	*** ssc install psmatch2, replace	
<pre>Naive Naive N</pre>	72	*** ssc install nnmatch, replace	
<pre>74 75 76 78 78 78 79 78 79 78 79 78 79 78 7 7 7 7</pre>	73		
<pre>75 ************************************</pre>	74		
<pre>76 ************************************</pre>	75	*****	
<pre>77 ***********************************</pre>	76	**************************************	
<pre>79 * Estimate the propensity score model 1 logit treat ses male white public 1 1 reat ses male white public 1 2 * estimate the propensity scores for each unit 3 predict pscorel, pr 4 4 5 *** implement PSM with psmatch2 6 * nearest neighbor with caliper, replacement and common support 7 * caliper is set to 0.25 standard deviations of the ps 9 5 sum pscorel 9 5 sum pscorel 9 5 sum pscorel 9 5 spantch2 treat, pscore(pscorel) outcome(math) common caliper(`=scalar(cal)') 5 5 * model based ATT estimate (as above) and cluster standard error 9 7 reg math treat [fweight=_weight], cluster(schid) 9 5 * balance check 10 pstest ses male white public, both graph 11 5 * implement PSM with numatch 12 * balance check 13 * balance check 14 * arrest neighbor with replacement 15 * caliper and common support cannot be imposed 15 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5</pre>	77	***************************************	
<pre>79 * Estimate the propensity score model 10 logit treat ses male white public 1 1 20 * estimate the propensity scores for each unit 21 * estimate the propensity scores for each unit 22 * estimate the propensity scores for each unit 23 predict pscorel, pr 24 25 *** implement PSM with psmatch2 26 * nearest neighbor with caliper, replacement and common support 27 * caliper is set to 0.25 standard deviations of the ps 28 * so, we first need to estimate the standard deviation of the ps 29 sum pscorel 20 scular cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps 29 * psm using psmatch2 29 psmatch2 treat, pscore(pscorel) outcome(math) common caliper(`=scalar(cal)') 29 * model based ATT estimate (as above) and cluster standard error 29 reg math treat [fweight=_weight], cluster(schid) 29 * balance check 20 pstest ses male white public, sum both 20 pstest ses male white public, both graph 20 *** implement PSM with namatch 21 * narest neighbor with replacement 22 * caliper and common support cannot be imposed 23 * caliper and common support (cannot be imposed 24 ************************************</pre>	78		
<pre>00 logit treat ses male white public 11 * estimate the propensity scores for each unit predict pscorel, pr 4 *** implement PSM with psmatch2 * nearest neighbor with caliper, replacement and common support * caliper is set to 0.25 standard deviations of the ps * so, we first need to estimate the standard deviation of the ps * so, we first need to estimate the standard deviation of the ps * so, we first need to estimate the standard deviation of the ps * so, we first need to estimate the standard deviation of the ps * so, we first need to estimate the standard deviation of the ps * so are standard deviation of the ps * so are standard deviation of the ps * psm using psmatch2 psmatch2 treat, pscore(pscorel) outcome(math) common caliper('=scalar(cal)') * model based ATT estimate (as above) and cluster standard error reg math treat [fweight=_weight], cluster(schid) * balance check pstest ses male white public, sum both pstest ses male white public, both graph *** implement PSM with nmatch * caliper and common support cannot be imposed * caliper and common support cannot be imposed * caliper and common support cannot be imposed * so are standard explained common support cannot be imposed * so are standard explained common support cannot be imposed * so are standard explained common support cannot be imposed * so are standard explained common support cannot be imposed * so are standard explained common support cannot be imposed * so are standard with numben * caliper and common support cannot be imposed * so are standard explained common support cannot be imposed * so are standard with numben * caliper and common support cannot be imposed * so are standard explained common support cannot be imposed * so are standard explained common support cannot be imposed * so are standard with numben * caliper and common support cannot be imposed * so are standard explained common support cannot be imposed * so are standard explained common support cannot be imposed * so are standard explained common suport cannot be imposed</pre>	79	* Estimate the propensity score model	
<pre>* estimate the propensity scores for each unit predict pscorel, pr *** implement PSM with psmatch2 * nearest neighbor with caliper, replacement and common support * caliper is set to 0.25 standard deviations of the ps * so, we first need to estimate the standard deviation of the ps sum pscorel scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps * psm using psmatch2 psmatch2 treat, pscore(pscorel) outcome(math) common caliper(`=scalar(cal)') * model based ATT estimate (as above) and cluster standard error reg math treat [fweight=_weight], cluster(schid) * balance check pstest ses male white public, sum both pstest ses male white public, sum both pstest ses male white public, both graph ** implement PSM with nnmatch * caliper and common support cannot be imposed ** caliper and common support cannot be imposed ************************************</pre>	80	logit treat ses male white public	
<pre>82 * estimate the propensity scores for each unit 83 predict pscore1, pr 84 85 *** implement FSM with psmatch2 86 * nearest neighbor with caliper, replacement and common support 87 * caliper is set to 0.25 standard deviations of the ps 88 * so, we first need to estimate the standard deviation of the ps 99 sum pscore1 90 sum pscore1 91 scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps 93 * psm using psmatch2 94 psmatch2 treat, pscore(pscore1) outcome(math) common caliper(`=scalar(cal)') 95 * 96 * model based ATT estimate (as above) and cluster standard error 97 reg math treat [fweight=_weight], cluster(schid) 99 * balance check 99 states sem ale white public, sum both 91 pstest ses male white public, both graph 92 **** implement PSM with nmmatch * nearest neighbor with replacement * caliper and common support cannot be imposed 93 * caliper and common support cannot be imposed 94 * caliper and common support cannot be imposed 95 * caliper and common support canno</pre>	81		
<pre>83 predict pscore1, pr 84 *** implement PSM with psmatch2 * nearest neighbor with caliper, replacement and common support * caliper is set to 0.25 standard deviations of the ps 87 97 98 99 90 sum pscore1 90 sum pscore1 91 scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps 92 93 * psm using psmatch2 94 psmatch2 treat, pscore(pscore1) outcome(math) common caliper('=scalar(cal)') 95 96 * model based ATT estimate (as above) and cluster standard error 97 reg math treat [fweight=weight], cluster(schid) 99 * balance check 99 * balance check 99 * balance heck 99 * timplement PSM with numatch 10 * nearest neighbor with replacement 10 * caliper and common support cannot be imposed 10 * caliper and common support cannot be imposed 11 * caliper and common support cannot be imposed 12 ** caliper and common support cannot be imposed 13 ** caliper and common support cannot be imposed 14 ** caliper and common support cannot be imposed 15 * caliper and common support cannot be imposed 15 * caliper and common support cannot be imposed 15 * caliper and common support cannot be imposed 15 * caliper and common support cannot be imposed 16 * caliper and common support cannot be imposed 17 * caliper and common support cannot be imposed 18 ****</pre>	82	* estimate the propensity scores for each unit	
<pre>%4 %*** implement PSM with psmatch2 % nearest neighbor with caliper, replacement and common support % caliper is set to 0.25 standard deviations of the ps % so, we first need to estimate the standard deviation of the ps % sum pscore1 % scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps % % psm using psmatch2 % psmatch2 treat, pscore(pscore1) outcome(math) common caliper(`=scalar(cal)') % % model based ATT estimate (as above) and cluster standard error % reg math treat [fweight=_weight], cluster(schid) % % balance check % pstest ses male white public, sum both % pstest ses male white public, both graph % % *** implement PSM with nnmatch % nearest neighbor with replacement % caliper and common support cannot be imposed % % % % % % % % % % % % % % % % % % %</pre>	83	predict pscore1, pr	
<pre>#** implement PSM with psmatch2 * nearest neighbor with caliper, replacement and common support * caliper is set to 0.25 standard deviations of the ps * so, we first need to estimate the standard deviation of the ps sum pscore1 scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps * psm using psmatch2 psmatch2 treat, pscore(pscorel) outcome(math) common caliper(`=scalar(cal)') * model based ATT estimate (as above) and cluster standard error reg math treat [fweight=_weight], cluster(schid) * balance check pstest ses male white public, sum both pstest ses male white public, both graph *** implement PSM with nnmatch * caliper and common support cannot be imposed </pre>	84		
<pre>% nearest neighbor with caliper, replacement and common support * caliper is set to 0.25 standard deviations of the ps % so, we first need to estimate the standard deviation of the ps % scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps % scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps % scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps % scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps % scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps % scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps % scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps % scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps % scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps % scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps % scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps % scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps % scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps % scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps % scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps % scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps % transform the standard prore (scalar(cal)') % * model based ATT estimate (as above) and cluster standard error % reg math treat [fweight=_weight], cluster(schid) % * balance check % pstest ses male white public, sum both % pstest ses male white public, both graph % * marest neighbor with replacement % * caliper and common support cannot be imposed % * caliper and common support cannot be imposed % * caliper and common support cannot be imposed % * caliper and common support cannot be imposed % * caliper and common support cannot be imposed % * caliper and common support cannot be imposed % * caliper and common support cannot be imposed * * caliper</pre>	85	*** implement PSM with psmatch2	
<pre>* caliper is set to 0.25 standard deviations of the ps * so, we first need to estimate the standard deviation of the ps sum pscorel scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps * psm using psmatch2 psmatch2 treat, pscore(pscorel) outcome(math) common caliper(`=scalar(cal)') * model based ATT estimate (as above) and cluster standard error reg math treat [fweight=_weight], cluster(schid) * balance check pstest ses male white public, sum both pstest ses male white public, both graph *** implement PSM with nnmatch * nearest neighbor with replacement * caliper and common support cannot be imposed</pre>	86	* nearest neighbor with caliper, replacement and common support	
<pre>* so, we first need to estimate the standard deviation of the ps sum pscore1 scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps * psm using psmatch2 psmatch2 treat, pscore(pscore1) outcome(math) common caliper(`=scalar(cal)') * model based ATT estimate (as above) and cluster standard error reg math treat [fweight=_weight], cluster(schid) * balance check pstest ses male white public, sum both pstest ses male white public, both graph *** implement PSM with nnmatch * nearest neighbor with replacement * caliper and common support cannot be imposed</pre>	87	* caliper is set to 0.25 standard deviations of the ps	
<pre>89 90 sum pscore1 91 scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps 93 * psm using psmatch2 94 psmatch2 treat, pscore(pscore1) outcome(math) common caliper(`=scalar(cal)') 95 96 * model based ATT estimate (as above) and cluster standard error 97 reg math treat [fweight=_weight], cluster(schid) 99 * balance check 00 pstest ses male white public, sum both 11 pstest ses male white public, both graph 22 *** implement PSM with nnmatch 4 * nearest neighbor with replacement 5 * caliper and common support cannot be imposed</pre>	88	* so, we first need to estimate the standard deviation of the ps	
<pre>90 sum pscorel 91 scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps 92 93 * psm using psmatch2 94 psmatch2 treat, pscore(pscorel) outcome(math) common caliper(`=scalar(cal)') 95 * model based ATT estimate (as above) and cluster standard error 97 reg math treat [fweight=_weight], cluster(schid) 98 * balance check 99 * balance check 99 * balance check 99 pstest ses male white public, sum both 91 pstest ses male white public, both graph 92 *** implement PSM with nnmatch 93 *** implement PSM with nnmatch 94 * nearest neighbor with replacement 95 * caliper and common support cannot be imposed</pre>	89		
<pre>91 scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps 92 93 * psm using psmatch2 9 psmatch2 treat, pscore(pscorel) outcome(math) common caliper(`=scalar(cal)') 95 96 * model based ATT estimate (as above) and cluster standard error 97 reg math treat [fweight=_weight], cluster(schid) 98 99 * balance check 99 * balance check 90 pstest ses male white public, sum both 10 pstest ses male white public, both graph 03 *** implement PSM with nnmatch 04 * nearest neighbor with replacement * caliper and common support cannot be imposed</pre>	90	sum pscorel	
<pre>92 93 * psm using psmatch2 94 psmatch2 treat, pscore(pscorel) outcome(math) common caliper(`=scalar(cal)') 95 96 * model based ATT estimate (as above) and cluster standard error 97 reg math treat [fweight=_weight], cluster(schid) 99 * balance check 00 pstest ses male white public, sum both 11 pstest ses male white public, both graph 02 03 *** implement PSM with nnmatch 14 * nearest neighbor with replacement 15 * caliper and common support cannot be imposed</pre>	91	scalar cal = $r(sd)*0.25$ // caliper = 1/4 of standard deviation of the ps	
<pre>93 * psm tsing psmatch2 94 psmatch2 treat, pscore(pscorel) outcome(math) common caliper(`=scalar(cal)') 95 96 * model based ATT estimate (as above) and cluster standard error 97 reg math treat [fweight=_weight], cluster(schid) 98 9 * balance check 00 pstest ses male white public, sum both 11 pstest ses male white public, both graph 02 03 **** implement PSM with nnmatch 04 * nearest neighbor with replacement 05 * caliper and common support cannot be imposed</pre>	92		
<pre>94 psmatch2 treat, pscore(pscorel) outcome(math) common caliper(=scalar(cal)') 95 96 * model based ATT estimate (as above) and cluster standard error 97 reg math treat [fweight=_weight], cluster(schid) 98 99 * balance check 00 pstest ses male white public, sum both 01 pstest ses male white public, both graph 02 03 **** implement PSM with nnmatch 04 * nearest neighbor with replacement 05 * caliper and common support cannot be imposed</pre>	93	* psm using psmatch2	
<pre>95 96 * model based ATT estimate (as above) and cluster standard error 97 reg math treat [fweight=_weight], cluster(schid) 98 99 * balance check 00 pstest ses male white public, sum both 01 pstest ses male white public, both graph 02 03 *** implement PSM with nnmatch 04 * nearest neighbor with replacement 05 * caliper and common support cannot be imposed</pre>	94	psmatch2 treat, pscore(pscore1) outcome(math) common caliper('=scalar(cal)')	
<pre>% model based ATT estimate (as above) and cluster standard error % reg math treat [fweight=_weight], cluster(schid) % % balance check % pstest ses male white public, sum both % pstest ses male white public, both graph % *** implement PSM with nnmatch % nearest neighbor with replacement % caliper and common support cannot be imposed % % *** implement support cannot be imposed % % *** implement support cannot be imposed % % *** implement support cannot be imposed % % % % % % % % % % % % % % % % % % %</pre>	95	t and l based more set instants (as shows) and shows a standard source	
<pre>97 reg math treat [rweight], cluster(schid) 98 99 * balance check 00 pstest ses male white public, sum both 01 pstest ses male white public, both graph 02 03 *** implement PSM with nnmatch 04 * nearest neighbor with replacement 05 * caliper and common support cannot be imposed</pre>	96	* model pased ATT estimate (as apove) and cluster standard error	
<pre>98 99 * balance check 00 pstest ses male white public, sum both 01 pstest ses male white public, both graph 02 03 *** implement PSM with nnmatch 04 * nearest neighbor with replacement 05 * caliper and common support cannot be imposed</pre>	97	reg math treat [Iweight=_weight], cluster(schid)	
<pre>99 balance check 00 pstest ses male white public, sum both 01 pstest ses male white public, both graph 02 03 *** implement PSM with nnmatch 04 * nearest neighbor with replacement 05 * caliper and common support cannot be imposed</pre>	90	t balance sheek	
01 pstest ses male white public, both graph 02 03 *** implement PSM with nnmatch 04 * nearest neighbor with replacement 05 * caliper and common support cannot be imposed	99 100	<pre>&gt; Dalance check pstost sos malo white public, sum both</pre>	
02 03 *** implement PSM with nnmatch 04 * nearest neighbor with replacement 05 * caliper and common support cannot be imposed	101	pstest ses male white public, sum both	
<pre>03 *** implement PSM with nnmatch 04 * nearest neighbor with replacement 05 * caliper and common support cannot be imposed</pre>	102	poteot oes mare white public, both graph	
04 * nearest neighbor with replacement 05 * caliper and common support cannot be imposed	102	*** implement PSM with numatch	
05 * caliper and common support cannot be imposed	104	* nearest neighbor with replacement	
current and common support cannot be imposed	105	* caliper and common support cannot be imposed	
	105	cariper and common support cannot be imposed	

materil	ing.do ×									<b>•</b>
69									/	^
70	*** If not installed the following packages should be installed									
71	*** ssc install psmatch2, replace									
72	*** ssc install nnmatch, replace									
73										
74										
75	***************************************	****								
76	**************************************	****								
77	***************************************	* * * *								
78		. sum pscore	=1							
79	* Estimate the propensity score model	-								
80	logit treat ses male white public	Variable	e   01	os Me	an St	d. Dev.	Min	Ν	Max	
81										
82	* estimate the propensity scores for each unit	pscore	L 2	.49230	77.1	.889558 .	1679723	.87831	179	
83	predict pscore1, pr	-	I							
84		. scalar ca	L = r(sd) * 0.3	25 // calipe	r = 1/4	of standar	d deviati	on of t	the ps	
									-	
85	*** implement PSM with psmatch2									
85 86	<pre>*** implement PSM with psmatch2     * nearest neighbor with caliber, replacement and common support</pre>									
85 86 87	<pre>*** implement PSM with psmatch2   * nearest neighbor with caliber, replacement and common support   * caliper is set to 0.2<sup>5</sup> standard deviations of the ps</pre>	. * psm usii	ng psmatch2							
85 86 87 88	<pre>*** implement PSM with psmatch2  * nearest neighbor with caliber, replacement and common support  * caliper is set to 0.25 standard deviations of the ps  * so, we first need to estimate the standard deviation of the ps</pre>	. * psm usin . psmatch2 t	ng psmatch2 treat, pscore	e(pscorel) o	utcome (m	ath) commo	n caliper	(`=scal	lar(cal)')	
85 86 87 88 89	<pre>*** implement PSM with psmatch2  * nearest neighbor with caliber, replacement and common support  * caliper is set to 0.25 standard deviations of the ps  * so, we first need to estimate the standard deviation of the ps</pre>	. * psm usin . psmatch2	ng psmatch2 treat, pscore	e(pscorel) o	utcome (m	ath) commo	n caliper	(`=scal	lar(cal)')	
85 86 87 88 89 90	<pre>*** implement PSM with psmatch2  * nearest neighbor with caliber, replacement and common support  * caliper is set to 0.25 standard deviations of the ps  * so, we first need to estimate the standard deviation of the ps sum pscore1</pre>	. * psm usin . psmatch2 f	ng psmatch2 treat, pscore	e(pscorel) o	utcome(m	Controls	n caliper	(`=sca]	lar(cal)')	T-st.
85 86 87 88 89 90 91	<pre>*** implement PSM with psmatch2  * nearest neighbor with caliber, replacement and common support  * caliper is set to 0.25 standard deviations of the ps  * so, we first need to estimate the standard deviation of the ps sum pscore1 scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps</pre>	. * psm usin . psmatch2 f	ng psmatch2 treat, pscord iable San	e(pscorel) o mple Tr	utcome(m eated	controls	n caliper Differ	(` <b>=sca</b> ] ence	lar(cal)') S.E.	T-st.
85 86 87 88 90 91 92	<pre>*** implement PSM with psmatch2  * nearest neighbor with caliber, replacement and common support  * caliper is set to 0.25 standard deviations of the ps  * so, we first need to estimate the standard deviation of the ps sum pscorel scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps</pre>	. * psm usin . psmatch2 ( Var:	ng psmatch2 treat, pscorr iable Sau	a(pscorel) o mple Tr	eated	Controls	n caliper Differ	(`=sca] ence 8712	Lar(cal)') S.E.	T-st.
85 86 87 88 90 91 92 93	<pre>*** implement PSM with psmatch2  * nearest neighbor with calibor, replacement and common support  * caliper is set to 0.25 standard deviations of the ps  * so, we first need to estimate the standard deviation of the ps sum pscore1 scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps * psm using psmatch2</pre>	. * psm usin . psmatch2 f	ng psmatch2 treat, pscorr iable Sau math Unmat	a(pscorel) o mple Tr ched 55.8	utcome (m eated 59375 59375	Controls 46.8787879	n caliper Differ 8.9805	(`=sca] ence 8712 4375	Lar(cal)') S.E. 1.26608652 1.81218926	T-st. 7.
85 86 87 88 90 91 92 93 94	<pre>*** implement PSM with psmatch2  * nearest neighbor with calibor, replacement and common support  * caliper is set to 0.25 standard deviations of the ps  * so, we first need to estimate the standard deviation of the ps sum pscore1 scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps * psm using psmatch2 psmatch2 treat, pscore(pscore1) outcome(math) common caliper(`=scalar(cal)')</pre>	. * psm usin . psmatch2 f	ng psmatch2 treat, pscorr iable San math Unmat	a(pscorel) o mple Tr ched 55.8 ATT 55.8	utcome (m eated 59375 59375	ath) commo Controls 46.8787875 51.421875	n caliper Differ 8.9805 4.	(`=sca] ence 8712 4375	lar(cal)') S.E. 1.26608652 1.81218926	T-st. 7. 2.
85 86 87 88 90 91 92 93 94 95	<pre>*** implement PSM with psmatch2  * nearest neighbor with caliber, replacement and common support  * caliper is set to 0.25 standard deviations of the ps  * so, we first need to estimate the standard deviation of the ps sum pscorel scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps * psm using psmatch2 psmatch2 treat, pscore(pscorel) outcome(math) common caliper(`=scalar(cal)')</pre>	. * psm usin . psmatch2 + Var:	ng psmatch2 treat, pscorr iable Sar math Unmat	e(pscorel) o mple Tr ched 55.8 ATT 55.8	utcome (m eated 59375 59375	Controls 46.8787879 51.421875	n caliper Differ 8.9805 4.	(`=scal ence 8712 4375	lar(cal)') S.E. 1.26608652 1.81218926 atimated.	T-st 7. 2.
85 86 87 88 90 91 92 93 94 95 96	<pre>*** implement PSM with psmatch2  * nearest neighbor with calibor, replacement and common support  * caliper is set to 0.25 standard deviations of the ps  * so, we first need to estimate the standard deviation of the ps sum pscore1 scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps * psm using psmatch2 psmatch2 treat, pscore(pscore1) outcome(math) common caliper(`=scalar(cal)') * model based ATT estimate (as above) and cluster standard error</pre>	. * psm usin . psmatch2 * Var: Note: S.E. (	ng psmatch2 treat, pscorr iable Sar math Unmate does not tak	e(pscorel) o mple Tr ched 55.8 ATT 55.8 e into accou	utcome (m eated 59375 59375 nt that	ath) commo Controls 46.8787875 51.421875 the proper	n caliper Differ 8.9805 4. sity scor	(`=scal ence 8712 4375 e is es	lar(cal)') S.E. 1.26608652 1.81218926 stimated.	T-st 7. 2.
85 86 87 88 90 91 92 93 94 95 95 96 97	<pre>*** implement PSM with psmatch2  * nearest neighbor with calibor, replacement and common support  * caliper is set to 0.25 standard deviations of the ps  * so, we first need to estimate the standard deviation of the ps sum pscorel scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps * psm using psmatch2 psmatch2 treat, pscore(pscorel) outcome(math) common caliper(`=scalar(cal)') * model based ATT estimate (as above) and cluster standard error reg math treat [fweight=_weight], cluster(schid)</pre>	. * psm usin . psmatch2 * Var: Note: S.E. (	ng psmatch2 treat, pscorr iable Sar math Unmate does not take	e(pscorel) o mple Tr ched 55.8 ATT 55.8 e into accou	utcome (m eated 59375 59375 nt that	Controls Controls 46.8787875 51.421875 the proper	n caliper Differ 8.9805 4. sity scor	(`=scal ence 8712 4375 e is es	Lar(cal)') S.E. 1.26608652 1.81218926 stimated.	T-st 7. 2.
85 86 87 88 90 91 92 93 94 95 95 96 97 98	<pre>*** implement PSM with psmatch2  * nearest neighbor with calibor, replacement and common support  * caliper is set to 0.25 standard deviations of the ps  * so, we first need to estimate the standard deviation of the ps sum pscorel scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps * psm using psmatch2 psmatch2 treat, pscore(pscorel) outcome(math) common caliper(`=scalar(cal)') * model based ATT estimate (as above) and cluster standard error reg math treat [fweight=_weight], cluster(schid)</pre>	. * psm usin . psmatch2 * Var: Note: S.E. (	ng psmatch2 treat, pscorr iable Sar math Unmate does not take psmatch2:	e(pscorel) o mple Tr thed 55.8 ATT 55.8 e into accou	utcome (m eated 59375 59375 nt that	Controls Controls 46.8787875 51.421875 the proper	n caliper Differ 8.9805 4. sity scor	(`=scal ence 8712 4375 e is es	Lar(cal)') S.E. 1.26608652 1.81218926 stimated.	T-st 7. 2.
85 86 87 90 91 92 93 94 95 94 95 96 97 98 99	<pre>*** implement PSM with psmatch2  * nearest neighbor with calibor, replacement and common support  * caliper is set to 0.25 standard deviations of the ps  * so, we first need to estimate the standard deviation of the ps sum pscorel scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps * psm using psmatch2 psmatch2 treat, pscore(pscorel) outcome(math) common caliper(`=scalar(cal)') * model based ATT estimate (as above) and cluster standard error reg math treat [fweight=_weight], cluster(schid) * balance check</pre>	. * psm usin . psmatch2 * Var: Note: S.E. ( psmatch2: Treatment	ng psmatch2 treat, pscorr iable Sar math Unmate does not take psmatch2: Common support	e(pscorel) o mple Tr thed 55.8 ATT 55.8 e into accou	utcome (m eated 59375 59375 nt that	Controls Controls 46.8787875 51.421875 the proper	n caliper Differ 8.9805 4. sity scor	(`=scal ence 8712 4375 e is es	Lar(cal)') S.E. 1.26608652 1.81218926 stimated.	T-st 7. 2.
85 86 87 90 91 92 93 94 95 96 97 98 99	<pre>*** implement PSM with psmatch2  * nearest neighbor with calibor, replacement and common support  * caliper is set to 0.25 standard deviations of the ps  * so, we first need to estimate the standard deviation of the ps sum pscorel scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps * psm using psmatch2 psmatch2 treat, pscore(pscorel) outcome(math) common caliper(`=scalar(cal)') * model based ATT estimate (as above) and cluster standard error reg math treat [fweight=_weight], cluster(schid) * balance check pstest ses male white public, sum both</pre>	. * psm usin . psmatch2 * Var: Note: S.E. ( psmatch2: Treatment assignment	ng psmatch2 treat, pscorr iable Sau math Unmate does not take psmatch2: Common support On support	e(pscorel) o mple Tr thed 55.8 ATT 55.8 e into accou	utcome (m eated 59375 59375 nt that	Controls Controls 46.8787875 51.421875 the proper	n caliper Differ 8.9805 4. sity scor	(`=scal ence 8712 4375 e is es	Lar(cal)') S.E. 1.26608652 1.81218926 stimated.	T-st 7. 2.
85 86 87 88 90 91 92 93 94 95 96 97 98 99 90 00	<pre>*** implement PSM with psmatch2  * nearest neighbor with caliber, replacement and common support  * caliper is set to 0.2° standard deviations of the ps  * so, we first need to estimate the standard deviation of the ps sum pscorel scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps * psm using psmatch2 psmatch2 treat, pscore(pscorel) outcome(math) common caliper(`=scalar(cal)') * model based ATT estimate (as above) and cluster standard error reg math treat [fweight=_weight], cluster(schid) * balance check pstest ses male white public, sum both pstest ses male white public, both graph</pre>	. * psm usin . psmatch2 * Var:	ng psmatch2 treat, pscor iable Sau math Unmate does not take psmatch2: Common support On suppor	e(pscorel) o mple Tr ched 55.8 ATT 55.8 e into accou	utcome (m eated 59375 59375 nt that	Controls Controls 46.8787875 51.421875 the proper	n caliper Differ 8.9805 4. sity scor	(`=scal ence 8712 4375 e is es	Lar(cal)') S.E. 1.26608652 1.81218926 stimated.	T-st 7. 2.
85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 90 00 01	<pre>*** implement PSM with psmatch2  * nearest neighbor with caliber, replacement and common support  * caliper is set to 0.25 standard deviations of the ps  * so, we first need to estimate the standard deviation of the ps sum pscore1 scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps * psm using psmatch2 psmatch2 treat, pscore(pscore1) outcome(math) common caliper(`=scalar(cal)') * model based ATT estimate (as above) and cluster standard error reg math treat [fweight=_weight], cluster(schid) * balance check pstest ses male white public, sum both pstest ses male white public, both graph</pre>	. * psm usin . psmatch2 * Var: Var: Note: S.E. of psmatch2: Treatment assignment	ng psmatch2 treat, pscore iable Sau math Unmate does not take psmatch2: Common support On support	e(pscorel) o mple Tr ched 55.8 ATT 55.8 e into accou Total	utcome (m eated 59375 59375 nt that	Controls Controls 46.8787875 51.421875 the proper	n caliper Differ 8.9805 4. sity scor	(`=scal ence 8712 4375 e is es	Lar(cal)') S.E. 1.26608652 1.81218926 stimated.	T-st 7. 2.
85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 90 00 101	<pre>*** implement PSM with psmatch2  * nearest neighbor with caliber, replacement and common support  * caliper is set to 0.25 standard deviations of the ps  * so, we first need to estimate the standard deviation of the ps sum pscore1 scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps * psm using psmatch2 psmatch2 treat, pscore(pscore1) outcome(math) common caliper(`=scalar(cal)') * model based ATT estimate (as above) and cluster standard error reg math treat [fweight=_weight], cluster(schid) * balance check pstest ses male white public, sum both pstest ses male white public, both graph *** implement PSM with nnmatch *** implement PSM with nnmatch</pre>	. * psm usin . psmatch2 * Var: Var: Note: S.E. of psmatch2: Treatment assignment Untreated Treated	ng psmatch2 treat, pscore iable Sar math Unmate does not take psmatch2: Common support On suppor 132 128	e(pscorel) o mple Tr ched 55.8 ATT 55.8 e into accou Total 132 128	utcome (m eated 59375 59375 nt that	Controls Controls 46.8787875 51.421875 the proper	n caliper Differ 8.9805 4. sity scor	(`=scal ence 8712 4375 e is es	Lar(cal)') S.E. 1.26608652 1.81218926 stimated.	T-st 7. 2.
85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103	<pre>*** implement PSM with psmatch2  * nearest neighbor with caliber, replacement and common support  * caliper is set to 0.25 standard deviations of the ps  * so, we first need to estimate the standard deviation of the ps sum pscore1 scalar cal = r(sd)*0.25 // caliper = 1/4 of standard deviation of the ps * psm using psmatch2 psmatch2 treat, pscore(pscore1) outcome(math) common caliper(`=scalar(cal)') * model based ATT estimate (as above) and cluster standard error reg math treat [fweight=_weight], cluster(schid) * balance check pstest ses male white public, sum both pstest ses male white public, both graph *** implement PSM with nnmatch  * nearest neighbor with replacement </pre>	. * psm usin . psmatch2 * Var: Var: Note: S.E. of psmatch2: Treatment assignment Untreated Treated	ng psmatch2 treat, pscore iable Sar math Unmate does not take psmatch2: Common support On support 0n suppor 132 128	e(pscorel) o mple Tr ched 55.8 ATT 55.8 e into accou Total 132 128	eated 59375 59375 nt that	Controls Controls 46.8787875 51.421875 the proper	n caliper Differ 8.9805 4. sity scor	(`=scal ence 8712 4375 e is es	Lar(cal)') S.E. 1.26608652 1.81218926 stimated.	T-st 7. 2.

match	ing.do ×							•
69								^
70	*** If not installed the following packages should be installed							
71	*** ssc install psmatch2, replace							
72	*** ssc install nnmatch, replace							
73								
74								
75	***************************************	***						
76	**************************************	***						
77	***************************************	***						
78								
79	* Estimate the propensity score model							
80	logit treat ses male white public							
81								
82	* estimate the propensity scores for each unit							
83	predict pscorel, pr							
84	the implement POM with sematable							
85	*** implement PSM with psmatch2							
80	<pre>^ nearest neighbor with callper, replacement and common support t callers is set to 0.25 standard deviations of the pa-</pre>							
87	* callper is set to 0.25 standard deviations of the ps							
00	* so, we first need to estimate the standard deviation of the ps							
09								
90 01	sum pscorer scalar cal = $r(cd) \pm 0.25 / (calinor = 1/4 of standard domination of the ns$							
92	scalar car = r(su)~0.25 // callper = 1/4 or scandard deviation of the ps	* model base	d ATT estimat	te (as above	) and cl	uster stand	dard error	
02	* nem using nematch?	reg math tre	at [iweight=	_weight], cl	uster(sc	hid)		
9.5 Q.1	<pre>psm using psmatchz nematch2 treat _ necore(necore1) outcome(math) common caliner(`=scalar(cal)')Li </pre>	near regress	ion			Number of	Fobs =	256
95	psmatchz creat, pscore(pscore) outcome(math) common carper( =scarar(car) )	incur regress	1011			F(1, 9)	=	4.39
96	* model based ATT estimate (as above) and cluster standard error					Prob > F	=	0.0657
97	reg math treat [fweight= weight], cluster(schid)					R-squared	= £	0.0460
98						Root MSE	=	10.146
99	* balance check							
100	pstest ses male white public, sum both			(Std.	Err. ad	justed for	10 clusters	in schid)
101	pstest ses male white public, both graph			Robust				
102		math	Coef.	Std. Err.	t	P> t	[95% Conf.	Intervall
103	*** implement PSM with nnmatch			200. 211.	5		[500 SONT.	lineertarj
104	* nearest neighbor with replacement	treat	4.4375	2.118416	2.09	0.066	35469	9.22969
105	* caliper and common support cannot be imposed	_cons	51.42188	3.535302	14.55	0.000	43.42447	59.41928

. \* balance check

. pstest ses male white public, sum both

	Unmatched	М	Mean %reduct		t-test		
Variable	Matched	Treated	Control	%bias	bias	t	p> t
ses	U M	.23211 .23211	36947 .20523	65.0 2.9	95.5	5.25 0.22	0.000 0.825
male	U M	.47656 .47656	.53788 .42969	-12.2 9.4	23.6	-0.99 0.75	0.325 0.453
white	U M	.73438 .73438	.7197 .71875	3.3 3.5	-6.5	0.26 0.28	0.792 0.780
public	U M	.59375 .59375	.88636 .60938	-70.5 -3.8	94.7	-5.70 -0.25	0.000 0.799

```
matching.do* ×
110
111
112
113
                                         WITTHIN
114
115
116
      * balance before matching
117
      pstest ses male white public, treated(treat) raw graph
118
119
      qen weight = .
120
      gen att = .
121
122
      egen c = group(schid)
123
      levels c, local(cluster)
124
125

    guietly foreach j of local cluster {

126
          psmatch2 treat if c==`j', pscore(pscore1) outcome(math) caliper(`=scalar(cal)')
              replace weight = weight if c==`i'
127
128
              replace att = r(att) if c==`j'
129
      L }
130
131
      * Estimated ATT (ignoring clustering):
132
      sum att
133
134
      * Model based ATT with se adjusted for clustering
135
      reg math treat [fweight = weight] if weight!=., cluster(schid)
136
137
      * Balance after matching
138
      pstest ses male white public if weight!=., treated(treat) mweight(weight) raw graph
139
          *Note: mweight is need to get appropriate estimates in case some obs are used more than once
          * raw here actually refers to the matched dataset (weight!=.)
140
141
```

. \* Model based ATT with se adjusted for clustering

. reg math treat [fweight = weight] if weight!=., cluster(schid)

Linear regression

Number of obs	=	238
F(1, 9)	=	6.41
Prob > F	=	0.0321
R-squared	=	0.0581
Root MSE	=	10.584

(Std. Err. adjusted for 10 clusters in schid)

math	Coef.	Robust Std. Err.	t	P> t	[95% Conf.	Interval]
treat	5.235294	2.067732	2.53	0.032	.5577587	9.91283
_cons	50.7563	4.093998	12.40	0.000	41.49503	60.01757

. \* balance before matching

. pstest ses male white public, treated(treat) raw graph

	Me	ean		t-t	est
Variable	Treated	Control	%bias	t	p> t
ses male	.23211 .47656	36947 .53788	65.0 -12.2	5.25 -0.99	0.000 0.325
white	.73438	.7197	3.3	0.26	0.792
public	. 59375	.00030	-70.5	-5.70	0.000

\* Balance after matching

. pstest ses male white public if weight!=., treated(treat) mweight (84 missing values generated)

Variable	Me Treated	ean Control	%bias	t-t t	test p> t
ses male white public	.24655 .47059 .7563 .57983	.14807 .43697 .71429 .57983	10.6 6.7 9.6 0.0 <	0.78 0.52 0.73	0.435 0.604 0.465 1.000

key advantage of within matching! (%bias = 0)

# Comparing balance: naive vs within PSM

#### naive

#### within

. \* Balance after matching

. pstest ses male white public if weight!=., treated(treat) mweight (84 missing values generated)

Mean         Mean         Mean         t-te           Treated Control         %bias         Variable         Treated Control         %bias         t-te           ses         .23211         .20523         2.9         ses         .24655         .14807         10.6         0.78           male         .47656         .42969         9.4         male         .47059         .43697         6.7         0.52           white         .73438         .71875         3.5         white         .7563         .71429         9.6         0.73           public         .59375         .60938         -3.8         public         .57983         .57983         0.0         0.00							
ses.23211.205232.9ses.24655.1480710.60.78male.47656.429699.4male.47059.436976.70.52white.73438.718753.5white.7563.714299.60.73public.59375.60938-3.8public.57983.579830.00.00	Variable	Mean Treated Control	%bias	Variable	Mean Treated Con	trol %bias	t-test t p> t
	ses male white public	.23211 .20523 .47656 .42969 .73438 .71875 .59375 .60938	2.9 9.4 3.5 -3.8	ses male white public	.24655 .1 .47059 .4 .7563 .7 .57983 .5	4807       10.6         3697       6.7         1429       9.6         7983       0.0	0.78 0.435 0.52 0.604 0.73 0.465 0.00 1.000

. pstest ses male white public, sum

```
* number and proportion of unmatched by cluster
gen unmatched =(weight==.)
45
46
table schid if treat==1, c(rawsum unmatched n treat mean unmatched)
47
* number unmatched / number of treated / % unmatched
40
```

. table schid if treat==1, c(rawsum unmatched n treat mean unmatched)

School ID	rawsum(unmatc~d)	N(treat)	mean(unmatc~d)
7472	0	6	0
7829	2	14	.1428571
7930	0	12	0
24725	0	7	0
25456	0	5	0
25642	0	4	0
62821	2	52	.0384615
68448	5	13	.3846154
68493	0	6	0
72292	0	9	0

# Case study: preferential within PSM

```
*********
153
154
     ********
                            PREFERENTIAL WITHIN
     155
156
157
     * balance before matching
158
     pstest ses male white public, treated(treat) raw graph
159
160
     gen weight pw = .
161
162
     * egen c = group(schid)
     levels c, local(cluster)
163
164
165
    □ quietly foreach j of local cluster {
         psmatch2 treat if c==`j', pscore(pscore1) outcome(math) caliper(`=scalar(cal)')
166
167
            replace weight pw = weight if c==`j'
168
169
170
     psmatch2 treat if ((weight pw== .& treat==1) | (treat==0)), pscore(pscore1) outcome(math) caliper(`=scalar(cal)')
171
172
     replace weight pw = weight pw + weight if (weight pw!=. & weight!=.) // for control units that were already used in the within
173
     replace weight pw = weight if weight pw ==. // both treated and control that were unmatched in within
174
175
176
     * Model based ATT with se adjusted for clustering
177
     req math treat [fweight = weight pw] if weight pw!=., cluster(schid)
178
179
     * Balance after matching
180
     pstest ses male white public if weight pw!=., treated(treat) mweight(weight pw) raw graph
         *Note: mweight is need to get appropriate estimates in case some obs are used more than once
181
182
         * raw here actually refers to the matched dataset (weight2!=.)
102
```

#### **Case study: preferential within PSM**

. \* Model based ATT with se adjusted for clustering

. reg math treat [fweight = weight\_pw] if weight\_pw!=., cluster(schid)

Linear regression

Number of obs	=	256
F(1, 9)	=	5.79
Prob > F	=	0.0395
R-squared	=	0.0564
Root MSE	=	10.493

(Std. Err. adjusted for 10 clusters in schid)

math	Coef.	Robust Std. Err.	t	P> t	[95% Conf.	Interval]
treat	5.109375	2.123337	2.41	0.039	.306053	9.912697
_cons	50.75	3.827984	13.26	0.000	42.0905	59.4095

# Comparing balance: within vs preferential within PSM

#### within

. \* Balance after matching

. pstest ses male white public if weight!=., treated(treat) mweight (84 missing values generated)

#### preferential

- . \* Balance after matching
- . pstest ses male white public if weight\_pw!=., tre
  (71 missing values generated)

Variable	Me Treated	ean Control	%bias	t-t t	test p> t	Variable	Me Treated	ean Control	%bias
ses male	.24655	.14807 .43697	10.6 6.7	0.78 0.52	0.435	ses male	.23211	.15266	8.3 12.5
white public	.7563 .57983	.71429 .57983	9.6 0.0	0.73 0.00	0.465	white public	.73438 .59375	.69531 .60938	8.8 -3.5

### **Case study: preferential within PSM**

. table schid if treat==1, c(rawsum unmatched pw n treat mean unmatched pw)

School ID	rawsum(unmatc~w)	N(treat)	mean(unmatc~w)		
7472	0	6	0		
7829	0	14	0		
7930	0	12	0		
24725	0	7	0		
25456	0	5	0		
25642	0	4	0		
62821	0	52	0		
68448	0	13	0		
68493	0	6	0		
72292	0	9	0		

# **Concluding remarks**

- In the presence of clustered data several approaches can be followed to implement PSM
- "Within" matching works well with big clusters
- "Preferential" within matching is an attractive alternative when all or some clusters are small
- Available routines in Stata can be adapted to clustered data
- Future developments:
  - Standard errors accounting for estimation of PS (as in teffects psmatch2)
  - Within-cluster balance

# (Some) References

- Abadie, A., D. M. Drukker, J. L. Herr, and G. W. Imbens. 2004. Implementing matching estimators for average treatment effects in Stata. Stata Journal 4: 290– 311.
- Arpino B. and Cannas M. (2016) Propensity score matching with clustered data. An application to the estimation of the impact of caesarean section on the Apgar score, *Statistics in Medicine*, 35(12), 2074–2091.
- Arpino B. and Mealli F. (2011) The specification of the propensity score in multilevel studies, *Computational Statistics and Data Analysis*, 55, 1770-1780.
- Imbens, G. W. (2014). Matching methods in practice: Three examples (NBER Working Paper No. 19959). Cambridge, MA: National Bureau of Economic Research
- Imbens, G. W., & Rubin, D. B. (2015). Causal inference in statistics, social, and biomedical sciences. Cambridge University Press.
- Li F, Zaslavsky AM, Landrum MB. (2013) Propensity score weighting with multilevel data. *Statistics in Medicine*, 32(19): 3373–3387.
- Thoemmes FJ, West SG. (2011) The use of propensity scores for nonrandomized designs with clustered data. *Multivariate Behavioral Research*, 46(3): 514–543.



Bruno Arpino Pompeu Fabra University bruno.arpino@upf.edu https://sites.google.com/site/brunoarpino