***Online Appendix**: "When good balance goes bad: A discussion of common pitfalls when using entropy balancing"*

Appendix: Comparing Entropy Balancing (EB), Propensity Score Matching (PSM), and Ordinary Least Squares (OLS) Regression

1. Overview

The goal of this appendix is to use a simulated dataset with a handful of variables to illustrate how three common analysis tools weight observations to construct a counterfactual. In doing so, this appendix builds on illustrations by Hainmueller (2012) and McMullin and Schonberger (2020) designed to illustrate how entropy balancing assigns observational weights to eliminate covariate imbalance across a treatment and control sample of interest.

We begin by generating data containing a treatment indicator *t*, a single covariate *X*, and an outcome variable *Y*. Using this simulated dataset, we use three analysis approaches to identify a control sample used to estimate a treatment effect. These approaches are:

Analysis Approach	Data Preprocessing Approach	Estimation Approach
OLS	None (ordinary least squares)	Regression (equal weighted over full sample)
PSM and OLS	Estimate propensity scores and match treatment and control sample observations on propensity scores without replacement to produce a matched sample	Regression (equal weighted over propensity score matched samples)
Entropy Balancing and Weighted OLS	Entropy balancing to identify weights for each observation in the control sample such that the weighted control sample and treated sample have similar covariate distributions	Regression (equal weights for treated sample and entropy balancing weights for control sample over full sample)

These three approaches typify research designs accounting researchers commonly use. In the interest of brevity, we do not examine related matching approaches in the simulation (e.g., coarsened exact matching [CEM]), as Morgan and Winship (2015) document that PSM outperforms a number of other matching approaches.

We provide Stata code in blocks of yellow and Stata output and figures in blocks of blue.

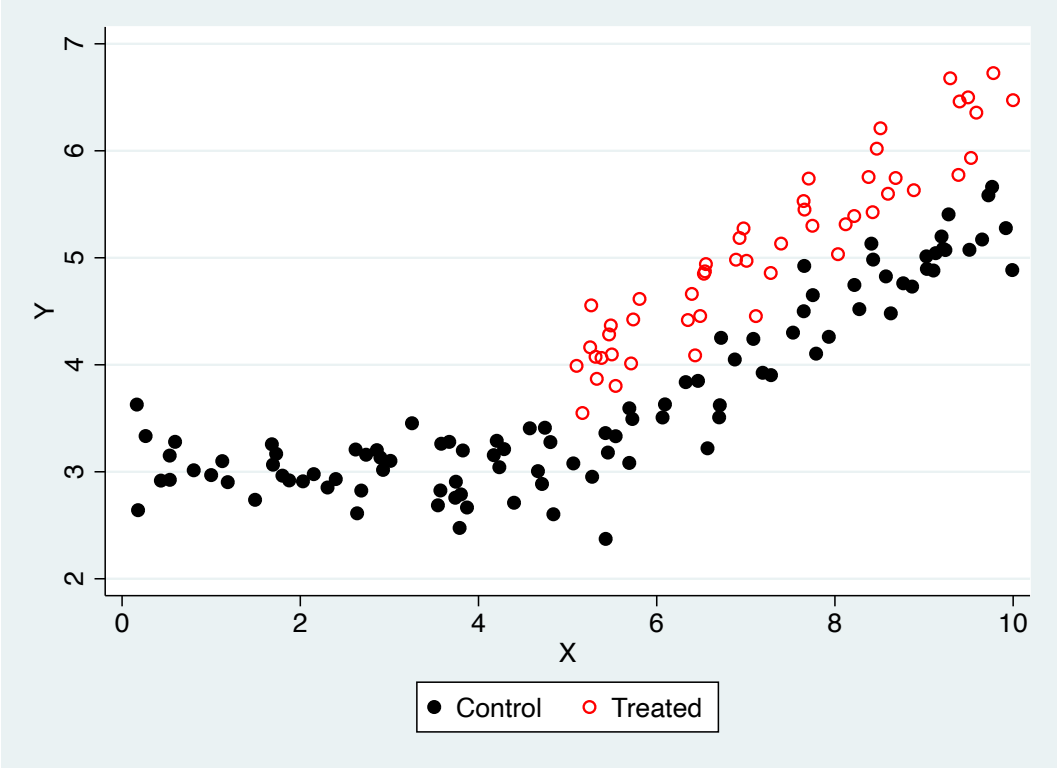
2. Update Stata, Install Packages, and Create Data

We first update Stata and install the entropy balancing package.

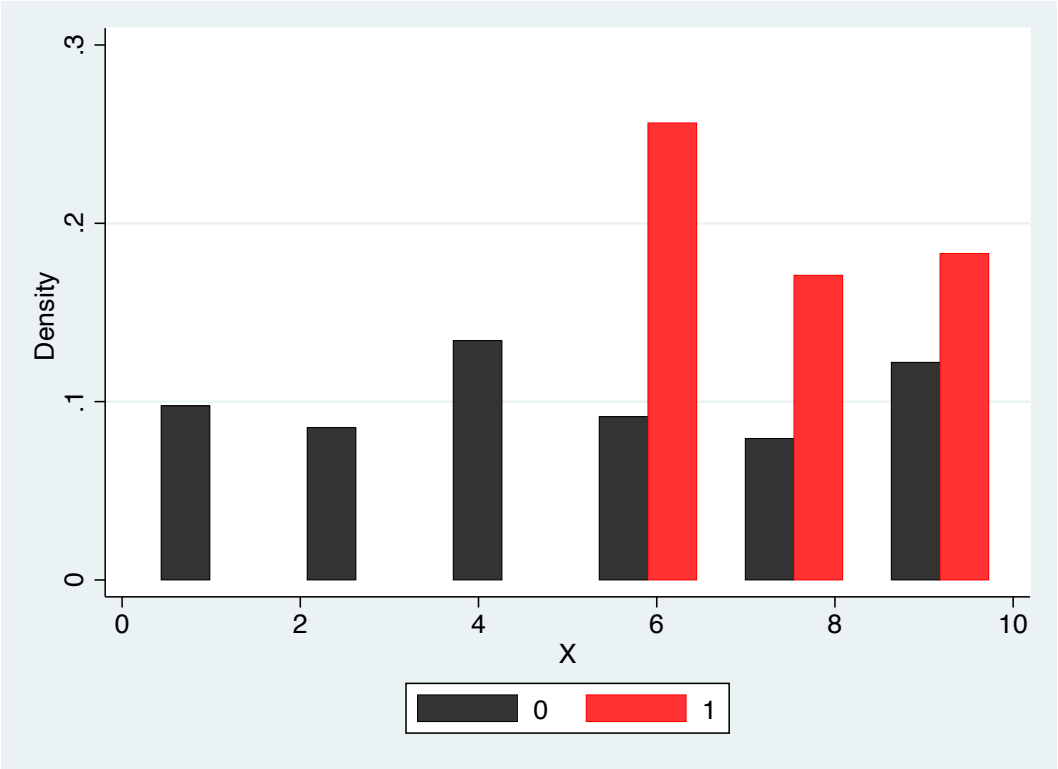
```
*-------------------------------------------------------------------------------*
* Update and Install software
*-------------------------------------------------------------------------------*
update all
ssc install ebalance, replace all
ssc install psmatch2, replace all
```

We next create 150 observations (50 treated and 100 control, denoted by a treatment indicator *t*) with one covariate *X*, constructed as a uniform random variable over the interval (0, 10) for the control sample (with $t = 0$) and over the interval (5, 10) for the treatment sample (with $t = 1$). As a result of this definition of *X* over a smaller interval for the treatment sample, treatment and control samples will display covariate imbalance. We also generate an outcome variable *Y* which is a function of the *X* covariate value as well as the treatment indicator *t*. This code also visualizes the data via a scatter plot of *Y* and *X* grouped by treatment status.

```
*-------------------------------------------------------------------------------*
* Create data to analyze
*-------------------------------------------------------------------------------*
*---> Set seed and initialize dataset
clear
set seed 123456
set obs 150
gen t = 1 if _n <= 50
replace t = 0 if t == .

*---> generate a covariate where there are control observations with
*     covariate values where no treatment observations exist
gen X = runiform(0,10) if t == 0
replace X = runiform(5,10) if t == 1

*---> generate a dependent variable that is a function of X and t:
*     A treatment effect of 1
*     A nonlinear relation between Y and X for control observations.
gen Y = .5 + .5*X + 1*t + rnormal(0,0.25)
replace Y = 3 + rnormal(0,0.25)  if X < 5

*---> plot the data to view what we've created
graph twoway (scatter Y X if t==0, leg(label(1 "control")) mcolor(black)) (scatter Y
X if t==1, leg(label(2 "treated")) mcolor(red) mfc(none))

*--->  Graph covariate imbalance across treated and control
graph box X, over(t)
byhist X, by(t) bin(6) tw1(color(red)) tw2(color(blue)) density
```



Using this code, we also plot a histogram of X separately by treatment status to depict covariate imbalance across the treated and control samples:



3. Analysis Approaches

In this section we employ three common research designs to the simulated data to assess how each method weights observations in constructing an estimate of the treatment effect.

3.1. OLS Regression Only

By far, the most common approach accounting researchers to adjust for covariates is a multiple regression. In this approach, researchers estimate an ordinary least squares (OLS) regression over all available observations without missing data for the variables in their model. This approach treats all observations with an equal weight.

```
*-------------------------------------------------------------------------------*
* Approach 1 – Regression only
*-------------------------------------------------------------------------------*
*---> Estimate Regression
. regress Y t X
```

This produces the following output:

```
      Source |       SS           df       MS      Number of obs   =       150
-------------+----------------------------------   F(2, 147)       =    338.88
       Model |  146.342981         2  73.1714905   Prob > F        =    0.0000
    Residual |   31.740546       147  .215922082   R-squared       =    0.8218
-------------+----------------------------------   Adj R-squared   =    0.8193
       Total |  178.083527       149  1.19519146   Root MSE        =    .46467

------------------------------------------------------------------------------
           Y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           t |   .8165832   .0866383     9.43   0.000     .6453657    .9878007
           X |   .2897373   .0152053    19.05   0.000      .259688    .3197865
       _cons |   2.163907   .0909312    23.80   0.000     1.984206    2.343608
------------------------------------------------------------------------------
```

We see that the imbalance and nonlinearity result in a downward bias for the estimated coefficient on the treatment indicator *t* (0.817 coefficient vs 1.000 for an unbiased estimate) and on *X* (0.290 coefficient vs 0.500 for an unbiased estimate). This bias in both estimates derives from the assumption of a constant linear treatment effect across all observations in the sample, resulting in excess weight applied to control sample observations where $X < 5$ and the treatment effect is zero.

3.2. PSM and Regression

Accounting researchers' efforts to achieve covariate balance between treatment and control samples have emphasized propensity score matching (e.g., Rosenbaum and Rubin 1983; Armstrong, Jagolinzer, and Larcker 2010; Lawrence, Minutti-Meza, and Zhang 2011).[1] In this

[1] Accounting researchers also employ alternative matching methods, including matching treated observations to control observations on a small subset of covariates using nearest neighbor matching within a pre-specified group such as industry (e.g., Kothari, Leone, and Wasley 2005), using coarsened exact matching to facilitate matching on several covariates (e.g., DeFond, Erkens, and Zhang 2017), or matching on a distance metric such as the generalized Mahalanobis distance (e.g., Diamond and Sekhon 2005).

approach, researchers estimate a propensity score model where the dependent variable is the treatment indicator and the independent variables are a set of covariates, compute propensity scores (p-scores) as the fitted values from this model, and then match treated observations to control observations with similar p-scores. With these subsamples identified, researchers estimate a regression over all matched observations. This approach results in unmatched observations receiving a zero weight and matched observations typically receiving a positive, integer weight. When matching is done without replacement, this weight will be one.

```
*-------------------------------------------------------------------------------*
*  Approach 2 – PSM - Caliper distance with replacement
*-------------------------------------------------------------------------------*
*---> estimate Pscore
psmatch2 t X, norepl descending caliper(0.03) logit

*---> Examine '_weight'; weight variable created by psmatch2
tab _weight t
```

This code produces the following output:

First, it returns the estimated propensity score model. Fitted values from this model (p-scores) are used to construct a distance metric between observations (i.e., differences in p-scores). Matching typically proceeds by identifying a control sample observation for each treatment sample observation where the distance metric is the smallest and less than some threshold, often referred to as a caliper.

```
Logistic regression                               Number of obs     =        150
                                                  LR chi2(1)        =      22.67
                                                  Prob > chi2       =     0.0000
Log likelihood = -84.140984                       Pseudo R2         =     0.1187

------------------------------------------------------------------------------
           t |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           X |   .3517719   .0827706     4.25   0.000     .1895445    .5139992
       _cons |   -2.89988   .5829605    -4.97   0.000    -4.042462   -1.757299
------------------------------------------------------------------------------
```

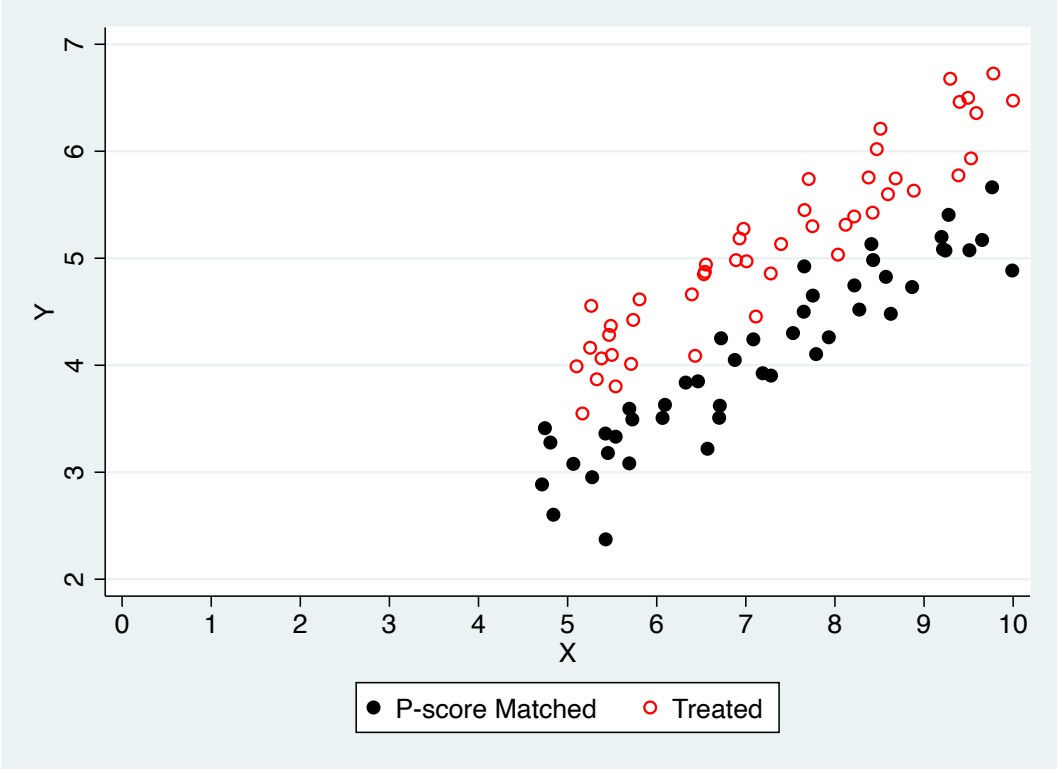
Next, this code describes the resulting weights `_weight` from matching without replacement after sorting the data in descending order, and requiring a maximum difference between propensity scores of 0.03. Only 46 of the 50 treatment observations are matched via this process. This matched control sample is used as the counterfactual.

```
 psmatch2: |
 weight of |
   matched |           t
  controls |         0          1 |     Total
-----------+----------------------+----------
         1 |        46         46 |        92
-----------+----------------------+----------
     Total |        46         46 |        92
```

Next we plot the data in these two samples.

```
*---> SCATTER PLOT
*plot scatter plot of matched
graph twoway  (scatter Y X if t==0 & _weight<., leg(label(1 "P-score Matched"))
mcolor(black%30) xlabel(0(1)10) ylabel(2(1)7)) (scatter Y X if t==1 & _weight<.,
leg(label(2 "Treated")) mcolor(red) mfc(none))
```

Examining the scatter plot of *Y* and *X* of the matched observations, we can see the procedure discarded all control observations with values of *X* less than approximately 5, consistent with a valid match between treatment and control observations based on the single covariate.



We next estimate the regression after performing propensity score matching without replacement by only including those observations with a non-missing weight in the estimation.

```
*---> Estimate Regression
regress Y t X if _weight != .
```

This produces the following output:

```
      Source |       SS           df       MS      Number of obs   =        92
-------------+----------------------------------   F(2, 89)        =    466.92
       Model |  80.6372937         2  40.3186468   Prob > F        =    0.0000
    Residual |  7.68517487        89  .086350279   R-squared       =    0.9130
-------------+----------------------------------   Adj R-squared   =    0.9110
       Total |  88.3224685        91  .970576577   Root MSE        =    .29385
```

```
------------------------------------------------------------------------------
           Y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           t |   .9634928   .0613409    15.71   0.000     .8416098    1.085376
           X |   .5089889   .0200039    25.44   0.000     .4692415    .5487362
       _cons |   .4322711   .1499198     2.88   0.005     .1343837    .7301584
------------------------------------------------------------------------------
```

This analysis demonstrates that using propensity score matching to identify a matched sample eliminates bias in the regression estimate of the coefficients on *t* (0.963 vs 1.000 for an unbiased estimate) and on *X* (0.509 vs 0.500 for an unbiased estimate).

3.3. Entropy Balancing and Weighted Regression

While propensity score matching typically improves covariate balance, a more recently developed method of reweighting control sample observations, entropy balancing (Hainmueller 2012), virtually eliminates covariate imbalance with a simple, one-line Stata command. McMullin and Schonberger (2020) apply entropy balancing in the setting of estimating abnormal accruals to provide a detailed discussion of the advantages of entropy balancing in addressing non-linear relations with underlying covariates in order to eliminate (or at least reduce) bias in estimated treatment effects. Further, the authors describe several limitations of entropy balancing in empirical-archival research settings with features that are common in accounting research. We build on their discussion in the following illustration.

The first step in performing entropy balancing is to run the ebal command. This iterative algorithim creates a new variable `_webal`, which when used to weight the control sample results in near perfect covariate balance. The option `target(3)` sets the balance conditions required to be met for the algorithm to cease adjusting the control sample weights, which in this case requires the first three moments of the *X* covariate distribution to be the same (within a default tolerance) across the treated and weighted control sample.

```
*-------------------------------------------------------------------------------*
* Approach 3 - Entropy Balancing and Weighted Regression
*-------------------------------------------------------------------------------*
*--->run ebalance
ebalance t X, target(3)
```

This command produces the following output:

```
Data Setup
Treatment variable:   t
Covariate adjustment: X (1st order). X (2nd order). X (3rd order).


Optimizing...
Iteration 1: Max Difference = 6186.94578
Iteration 2: Max Difference = 2276.21682
Iteration 3: Max Difference = 837.550557
Iteration 4: Max Difference = 308.317911
Iteration 5: Max Difference = 113.657737
Iteration 6: Max Difference = 42.0557396
Iteration 7: Max Difference = 15.8045807
Iteration 8: Max Difference = 6.60675262
```

```
Iteration 9: Max Difference = 3.50860125
Iteration 10: Max Difference = 1.94948744
Iteration 11: Max Difference = .790783647
Iteration 12: Max Difference = .196287767
Iteration 13: Max Difference = .016841045
Iteration 14: Max Difference = .000139034
maximum difference smaller than the tolerance level; convergence achieved


Treated units: 50      total of weights: 50
Control units: 100     total of weights: 50


Before: without weighting

             |               Treat               |             Control
             |      mean   variance   skewness |      mean   variance   skewness
-------------+---------------------------------+---------------------------------
           X |      7.25      2.243      .1802 |      5.14      8.323     .02468


After:  _webal as the weighting variable

             |               Treat               |             Control
             |      mean   variance   skewness |      mean   variance   skewness
-------------+---------------------------------+---------------------------------
           X |      7.25      2.243      .1802 |      7.25      2.243      .1802
```

The first panel provides summary statistics that allow us to assess pre-adjustment covariate balance. The second panel shows the summary statistics after weighting the control sample using weights identified by the entropy balancing algorithm. While covariate balance was achieved, we next examine the weights created to achieve the balance.

```
*---> Analysis examining entropy balancing weights
. tabstat _webal, statistics(mean, max, N) by(t)
. gsort - t _webal
. list _webal if inrange(_n, 1, 10) & t == 0 | inrange(_n, _N-10+1, _N)
```

This produces the following output:

```
Summary for variables: _webal
     by categories of: t

       t |      mean       min       max         N
---------+--------------------------------------
       0 |        .5  1.27e-18  1.562386       100
       1 |         1         1         1        50
---------+--------------------------------------
   Total |  .6666667  1.27e-18  1.562386       150
------------------------------------------------
```

From this output we can see that the entropy balancing algorithm retains all 100 control and all 50 treatment observations. This output also shows that the treatment sample observations retain a

weight of one and that continuous weights assigned to control sample observations range from nearly zero (1.27e-18) to 1.56.

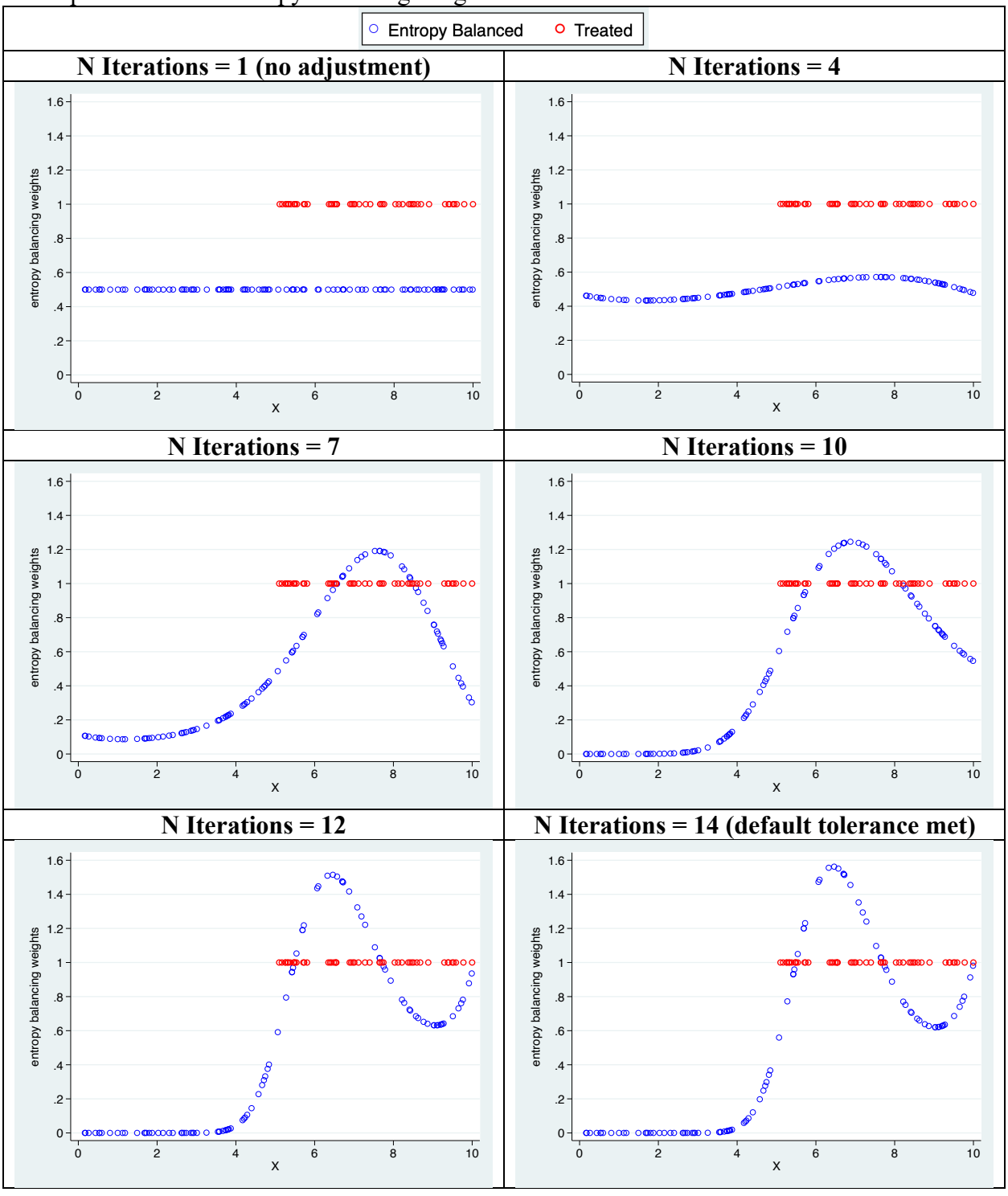
```
     +-----------+
     |    _webal |
     |-----------|
141. | 1.3520483 |
142. | 1.4549744 |
143. | 1.4727891 |
144. | 1.4858081 |
145. | 1.5137473 |
     |-----------|
146. | 1.5182191 |
147. | 1.5204759 |
148. | 1.5512371 |
149. | 1.5558077 |
150. | 1.5623861 |
     +-----------+
```

This list of the top 10 weights shows that no single observation was assigned an excessively high weight.

Given that we only have one covariate in this simulated example, we can compare the weight assigned to each observation and its value of *X* to understand how the weights are iteratively adjusted. To do this we use the following code create a scatter plot of `_webal` and `X` after each of the 14 iterations required to achieve covariate balance within the default tolerance of 0.015.

```
*---> Plot covariate X and entropy balance created variable '_webal'
graph twoway (scatter _webal X if t==0, leg(label(1 "Entropy Balanced"))
mcolor(blue) mfc(none) mlw(vthin) ysc(r(0 1.6)) ylabel(0(.2)1.6, angle(0) grid) )
(scatter _webal X if t==1, leg(label(2 "Treated")) mcolor(red) mfc(none))
```

These plots show the entropy balancing weights after N iterations.[2]



[2] We achieve this series of plots by stopping ebalance from iterating by increasing the tolerance. (E.g., setting tolerance to 7000 (i.e., adding the option `tol(7000)`) results in no adjustment, setting it to 5000 results in 2 iterations of adjusting the weights, 2000 results in 3 iterations, etc.) This tolerance corresponds to the Max Difference parameter reported with each iteration. See the accompanying Stata `do` file for the remaining tolerances we specify to stop the algorithm after each of the N iterations.

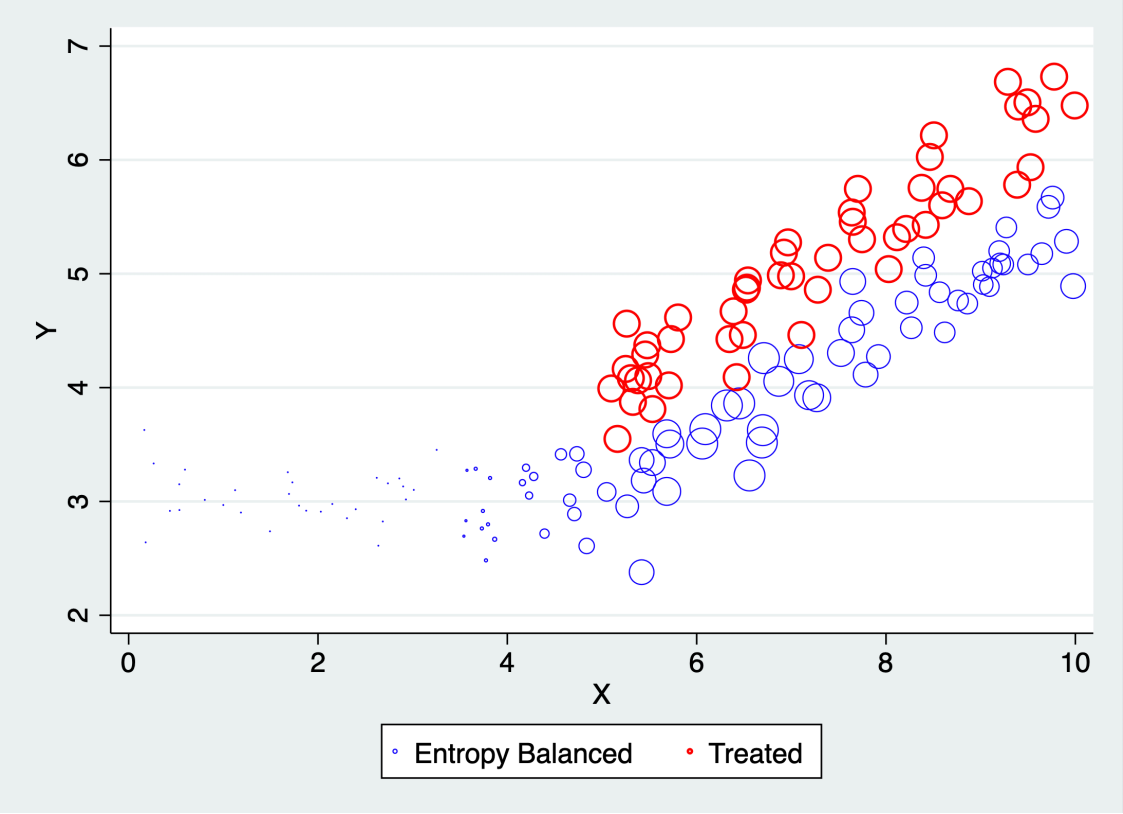
We direct the reader interested in the theoretical underpinning of this entropy optimization algorithm to Kapur and Kesavan (1992) or Mattos and Veiga (2004). Hainmuller (2012) provides these references for more detail on the iterative algorithm he employs in this technique.

Examining these plots, we see the iterative algorithm decreased weights assigned to observations with an X value less than 4 to be nearly zero. The remainder of the weights were assigned following a continuous function that spans the range of X values in the treatment sample.

To visualize the weights assigned by entropy balancing after the final iteration, we construct the scatter plot of the covariates Y and X for all observations in the sample, weighted by `_webal`.

```
*---> SCATTER PLOT
* Ensure Stata produces consistent sized dots based on weights across both groups
* See https://www.stata.com/statalist/archive/2008-08/msg00987.html
expand 2
replace X = . if _n>(_N/2)
recode t (1=0) (0=1) if X==.

*plot scatter plot of Y and X based on entropy balancing weights (_webal)
graph twoway (scatter Y X if t==0 [aw=_webal], leg(label(1 "Entropy Balanced"))
mcolor(blue) mfc(none) mlw(vthin) msize(*.25)) (scatter Y X if t==1 [aw=_webal],
leg(label(2 "Treated")) mcolor(red) mfc(none) msize(*.25))
```

This generates the following plot:



We see that all observations are retained after entropy balancing. This approach also down-weighted control observations where there was no overlap with treatment sample (i.e., X values below 5), assigning a weight of effectively zero to observations with X values below 4.

The estimation of the regression after executing the entropy balancing procedure is done using the `pweight` option and the `_webal` variable.

```
*---> Estimate Regression
. regress Y t X [pweight=_webal]
```

This produces the following output:

```
(sum of wgt is   1.0000e+02)

Linear regression                               Number of obs     =        150
                                                F(2, 147)         =     456.37
                                                Prob > F          =     0.0000
                                                R-squared         =     0.9099
                                                Root MSE          =     .28543

------------------------------------------------------------------------------
             |               Robust
           Y |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
           t |   .9686133   .0574659    16.86   0.000     .8550472    1.082179
           X |   .5085347   .0188487    26.98   0.000     .4712853    .5457841
       _cons |   .4256848   .1463198     2.91   0.004     .1365227    .7148468
------------------------------------------------------------------------------
```

Similar to the PSM approach, this demonstrates that entropy balancing is effective in eliminating bias in the estimation of the coefficients on t (0.967 vs. 1.000 for an unbiased estimate) and on X (0.509 vs. 0.500 for an unbiased estimate). While these estimates retain all observations in the dataset, the impact of those control observations that are not similar to the treated observations is effectively zero. This implies that the counterfactual is constructed by weighting the observations in the control samples using the entropy balance weights and that this counterfactual is similar to the treated sample.

References

Armstrong, C., A. Jagolinzer, and D. Larcker. 2010. "Chief Executive Officer Equity Incentives and Accounting Irregularities." *Journal of Accounting Research* 48 (2): 225-271.

DeFond, M., D. H. Erkens, and J. Zhang. 2017. "Do Client Characteristics Really Drive the Big N Audit Quality Effect? New Evidence from Propensity Score Matching." *Management Science* 63 (11): 3531–3997.

Diamond, A., and J. S. Sekhon. 2005. "Genetic Matching for Estimating Causal Effects: A General Multivariate Matching Method for Achieving Balance in Observational Studies." *Review of Economics and Statistics*, no. 0.

Hainmueller, J. 2012. “Entropy Balancing for Causal Effects: A Multivariate Reweighting Method to Produce Balanced Samples in Observational Studies.” *Political Analysis* 20 (1): 25–46.

Kapur, J. N., and H. K. Kesavan. 1992. *Entropy Optimization Principles with Applications*. Springer.

Kothari, S. P., A. J. Leone, and C. E. Wasley. 2005. “Performance Matched Discretionary Accrual Measures.” *Journal of Accounting & Economics* 39 (1): 163–97.

Lawrence, A., M. Minutti-Meza, and P. Zhang. 2011. “Can Big 4 versus Non-Big 4 Differences in Audit-Quality Proxies Be Attributed to Client Characteristics?” *The Accounting Review* 86 (1): 259–86.

Mattos, R., and A. Veiga. 2004. “Entropy Optimization: Computer Implementation of the Maxent and Minexent Principles.” Working Paper. Universidade Federal de Juiz de Fora, Brazil.

McMullin, J. L., and B. Schonberger. 2020. “Entropy-Balanced Accruals.” *Review of Accounting Studies* 25 (1): 84–119.

Morgan, S. L., and C. Winship. 2015. *Counterfactuals and Causal Inference*. Cambridge University Press.

Rosenbaum, P. R., and D. B. Rubin. 1983. “The Central Role of the Propensity Score in Observational Studies for Causal Effects.” *Biometrika* 70 (1): 41–55.