cobalt

Covariate Balance Tables and Plots

Pierce & Megha

April 1, 2020

Introduction to cobalt

- Developed by Noah Greifer
    - PhD student, UNC Chapel Hill
- Standardized balance measures
    - different methods for conditioning
    - different packages available for conditioning (optmatch, MatchIt, CBPS, ebal, WeightIt, twang)
- Tables and beautiful plots
- Can be used with clustered data, multiply imputed data, continuous treatment

Review of Steps used in Pre-Processing

1. Estimate balancing scores (e.g., propensity scores)

2. Condition on balancing scores

    - Weighting
    - Stratification
    - Matching

3. Check the balance on covariates- compare balance before and after conditioning

4. Iterate

5. Estimate average treatment effect after conditioning

Balance Evaluation: Why?

Balance is crucial

- We are trying to emulate a randomized experiment
- Conditional on true propensity scores, distribution of observed covariates is independent of treatment (Austin, 2011)
- After conditioning on estimated propensity scores, need to check balance
- If balance is inadequate, effect estimate may be biased

Balance Evaluation

- Balance evaluation is very important but…
- Applied studies rarely report balance evaluation. Evaluation is often inappropriate (Greifer, 2017; Austin, 2009).
- Over 66% of applied studies that used propensity scores assessed balance using statistical tests (Thoemmes & Kim, 2011).
- However, statistical tests are not recommended for balance evaluation.
    - Interest in sample not population (Stuart, 2008; Stuart, 2010; Austin, 2011)
    - Reduced power (Imai et al., 2008)
    - `cobalt` does not provide statistical tests

Criteria

- Standardized difference in means (for continuous covariates)
- Variance ratios (for continuous)
- Raw difference in proportions (for binary)
- Love plots, density plots, bar plots

Standardized Difference in Means (Continuous)

Standardized difference in means: the estimate of mean difference (before/after conditioning) divided by standard deviation of the covariate from the unadjusted sample (Greifer, 2017; Rosenbaum & Rubin, 1983; Austin, 2009).

- Stuart (2008) recommended use of standard deviation from unadjusted sample even when checking balance after conditioning
    - Compare mean differences in unadjusted and adjusted sample- the denominator is the same
- sd in treated group (default for ATT), untreated group (ATU), or pooled sd (ATE)
- Threshold of .1 is recommended by Stuart et al. (2013).

Variance Ratio (Continuous)

- Use SMDs to compare center of distributions but also important to compare variance- spread of distributions (Greifer, 2017)
- **Variance ratio**: ratio of variances of continuous covariates in treated and untreated groups
    - Ratios closer to 1 indicate variances of the two groups are similar (Greifer, 2017)
    - Recommended thresholds are 0.5 and 2 (Rubin, 2001)
    - In `cobalt`, the larger variance is in the numerator

Weighted Variance Calculation

Formula used in `cobalt` to calculate variances after conditioning if weights are involved:

$$\boldsymbol{s}_w^2 = \left(\frac{\sum_{i=1}^n w_i}{\left(\sum_{i=1}^n w_i\right)^2 - \sum_{i=1}^n w_i^2} \right) \sum_{i=1}^n w_i (x_i - \bar{x}_w)^2$$

Here w_i is weight for person i (from weighting or matching), x_i is value of covariate for person i and $\bar{x}_w$ is the weighted mean of x within each treatment group.

This formula is recommended by Austin (2008) and Austin & Stuart (2015).

Raw Difference in Proportions (Binary)

- For binary covariates, raw differences in proportions between treated and untreated groups (before/ after conditioning) are used to evaluate balance (Greifer, 2017).
    - Already on the same scale
- No variance ratios- variance of binary variables derived from proportion so ratios do not provide new information (Greifer, 2017)

Interactions and Squared Terms

Assess balance on two-way interactions and squared terms (Rubin, 2001; Austin, 2009; Stuart, 2010)

- Interactions because joint distributions should be similar
- Comparing means of squared terms (for continuous predictors) equivalent to comparing variances of treatment and control group (Austin & Stuart, 2015)

Effective Sample Size

Effective Sample Size - "a measure of the sample size a non-weighted sample would have to have to achieve the same level of precision as the weighted sample" (Greifer, 2017; Ridgeway et al., 2016)

$$ESS = \frac{\left(\sum_{i=1}^{n} w_i\right)^2}{\sum_{i=1}^{n} w_i^2}$$

- Proportionally larger weights lead to:
  - Lower ESS
  - Larger variance of weighted mean
  - Loss of precision
- `cobalt` calculates ESS

cobalt demonstration

Libraries

```r
# install.packages("cobalt")
library(cobalt)
```

```r
library(tidyverse)
library(MatchIt)
```

Data

```
Algebra_dat <-
  read_csv("8th-grade-Algebra-data.csv") %>%
  mutate(Locale = factor(Locale,
                         levels = c("R","S","U"),
                         labels = c("Rural","Suburban","Urban")))

Algebra_dat_org <- Algebra_dat
```

Formula

- There is a function in `cobalt` that takes in the outcome and a data frame or tibble containing the covariates and creates a formula based on that. It doesn't seem to have an easy way to add interaction terms or polynomial terms though.

```
# dataset with covariates
covs <- Algebra_dat %>%
  select(Math, SES, Locale)
```

```
f_lin <- f.build("D", covs)
f_lin
```

```
## D ~ Math + SES + Locale
## <environment: 0x7fb46c36b558>
```

Weighting: Calculations

Here, we are just estimating the propensity scores and calculating ATT weights.

```
# fitting propensity score model
ps_logit <- glm(f_lin, data = Algebra_dat, family = "binomial")

# estimating propensity scores
Algebra_dat$ps <- predict(ps_logit, type = "response")

# calculate the weights - ATT weighting by odds of treatment
Algebra_dat <- Algebra_dat %>%
  mutate(att_wt = D + (1 - D) * ps/(1-ps))
```

Weighting: Balance

Just a table with standardized difference in means after adjustment. `cobalt` normalizes the weights automatically.

```
bal.tab(f_lin,    # formula
        data = Algebra_dat, # data
        weights = "att_wt", # weights
        method = "weighting", # method
        m.threshold = .1) # threshold to judge
```

```
## Balance Measures
##                    Type Diff.Adj        M.Threshold
## Math             Contin.   0.1390 Not Balanced, >0.1
## SES              Contin.   0.1829 Not Balanced, >0.1
## Locale_Rural      Binary  -0.0398     Balanced, <0.1
## Locale_Suburban   Binary  -0.0288     Balanced, <0.1
## Locale_Urban      Binary   0.0686     Balanced, <0.1
##
## Balance tally for mean differences
##                    count
## Balanced, <0.1         3
## Not Balanced, >0.1     2
##
## Variable with the greatest mean difference
##  Variable Diff.Adj        M.Threshold
##       SES   0.1829 Not Balanced, >0.1
##
## Effective sample sizes
##            Control Treated
## Unadjusted 390.000     610
## Adjusted    57.511     610
```

Weighting: Balance

Adding variance ratios and balance measures for unadjusted sample.

```
b_w1 <- bal.tab(f_lin,
                data = Algebra_dat,
                weights = "att_wt",
                method = "weighting",
                disp.v.ratio = TRUE, # ask for var ratios
                un = TRUE) # asked for unadjusted sample stats

b_w1
```

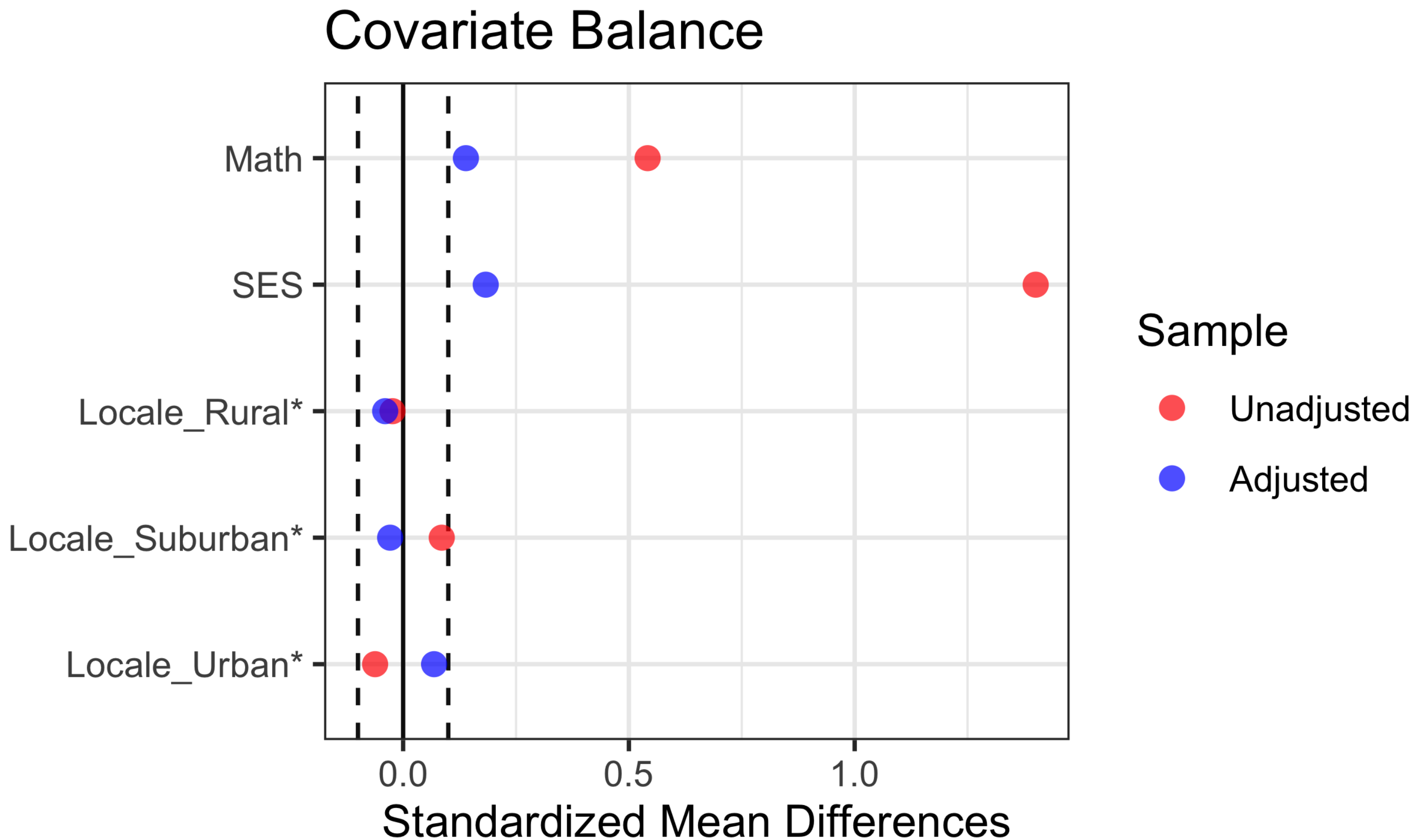
```
## Balance Measures
##                      Type Diff.Un V.Ratio.Un Diff.Adj V.Ratio.Adj
## Math              Contin.  0.5415     1.0440   0.1390      0.9415
## SES               Contin.  1.4009     1.2191   0.1829      1.0423
## Locale_Rural       Binary -0.0235             -0.0398
## Locale_Suburban    Binary  0.0855             -0.0288
## Locale_Urban       Binary -0.0620              0.0686
##
## Effective sample sizes
##            Control Treated
## Unadjusted 390.000     610
## Adjusted    57.511     610
```

Weighting: Love Plot

- Way to visualize results from balance evaluation
- Named after Dr. Thomas E. Love

```
love.plot(b_w1,   # add the table here
          threshold = .1, # threshold
          colors = c("red", "blue"), # change the colors
          size = 3, # size of dots
          alpha = .7,  # transparency
          stars = "raw") + # change the x axis label
  theme_bw()
```

Weighting: Love Plot



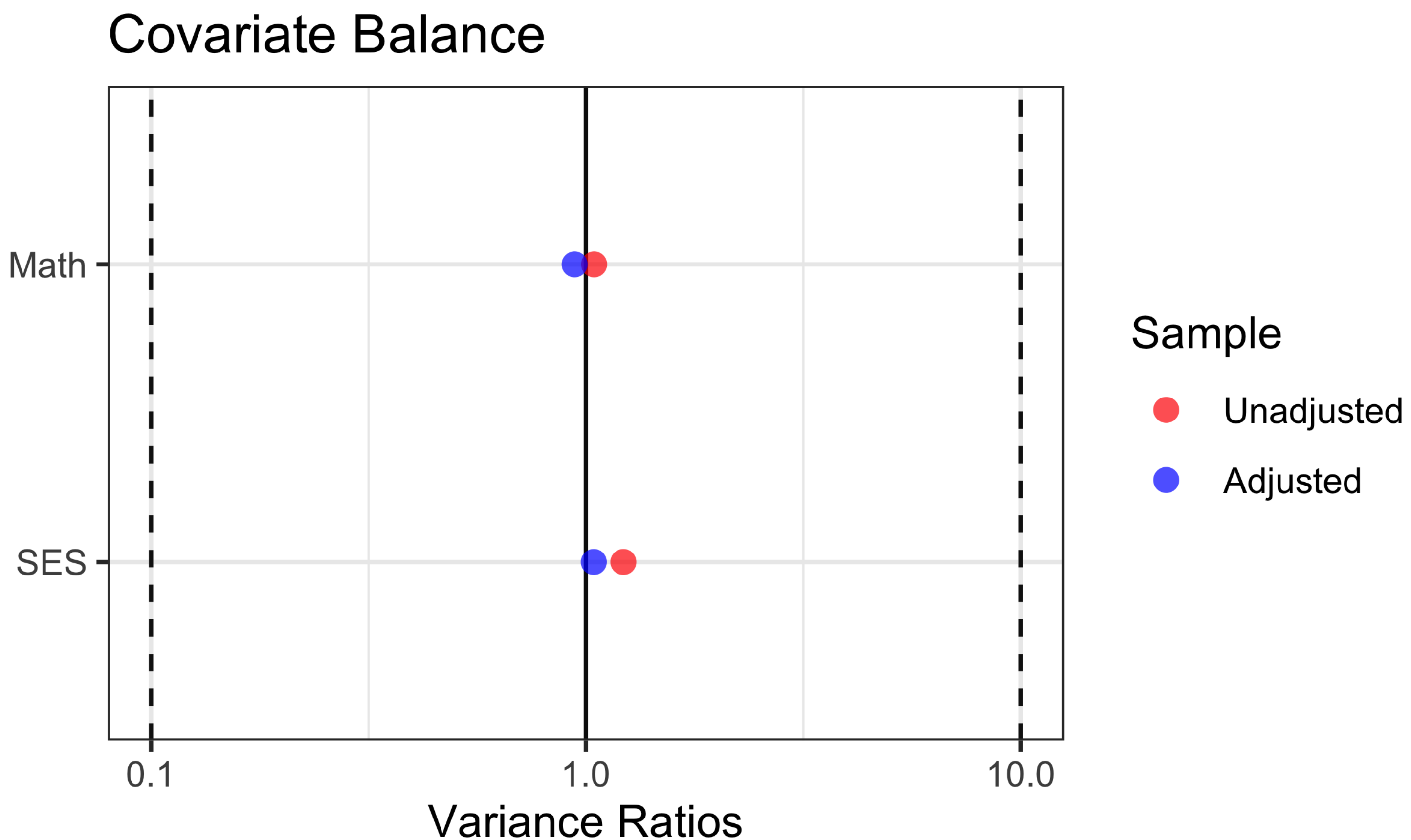
Weighting: Love Plot Default

```
love.plot(b_w1,
          threshold = .1,
          stars = "raw")
```

Weighting: Love Plot for Variance Ratios

```r
love.plot(b_w1,
          threshold = .1,
          colors = c("red", "blue"),
          size = 3,
          alpha = .7,
          stat = "variance.ratios", # just ask for var ratios here
          stars = "raw") +
  theme_bw()
```

Weighting: Love Plot for Variance Ratios

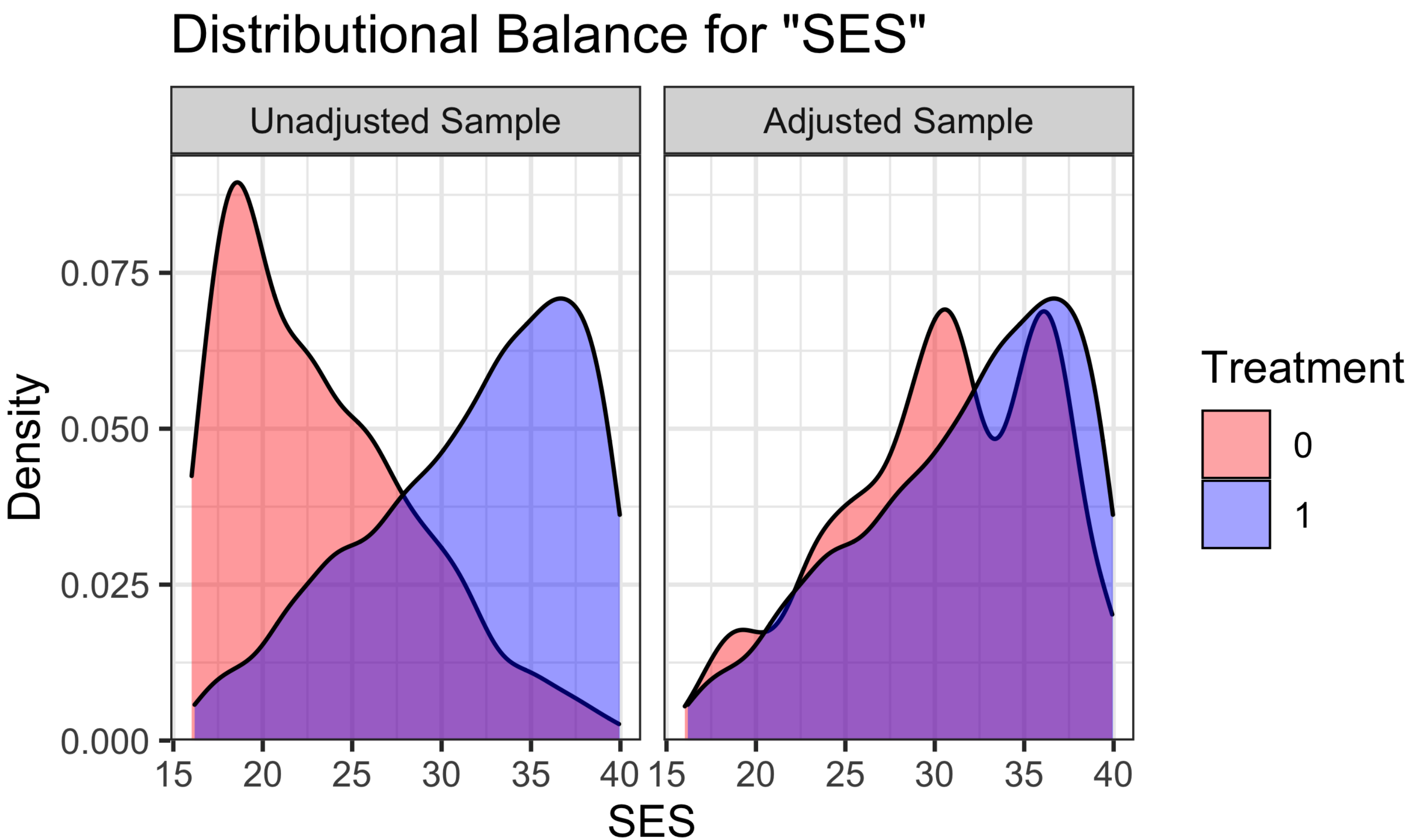


Weighting: Density Plot (SES)

Balance plots to evaluate similarities in univariate distributions of a covariate in treated and untreated groups. For continuous covariates, we look at density plots.

```
bal.plot(f_lin,
         data = Algebra_dat,
         weights = "att_wt",
         method = "weighting",
         var.name = "SES", # ask for a particular variable SES
         which = "both", # ask for both unadjusted and adjusted samp
         colors = c("red", "blue")) +
  theme_bw()
```

Weighting: Density Plot (SES)

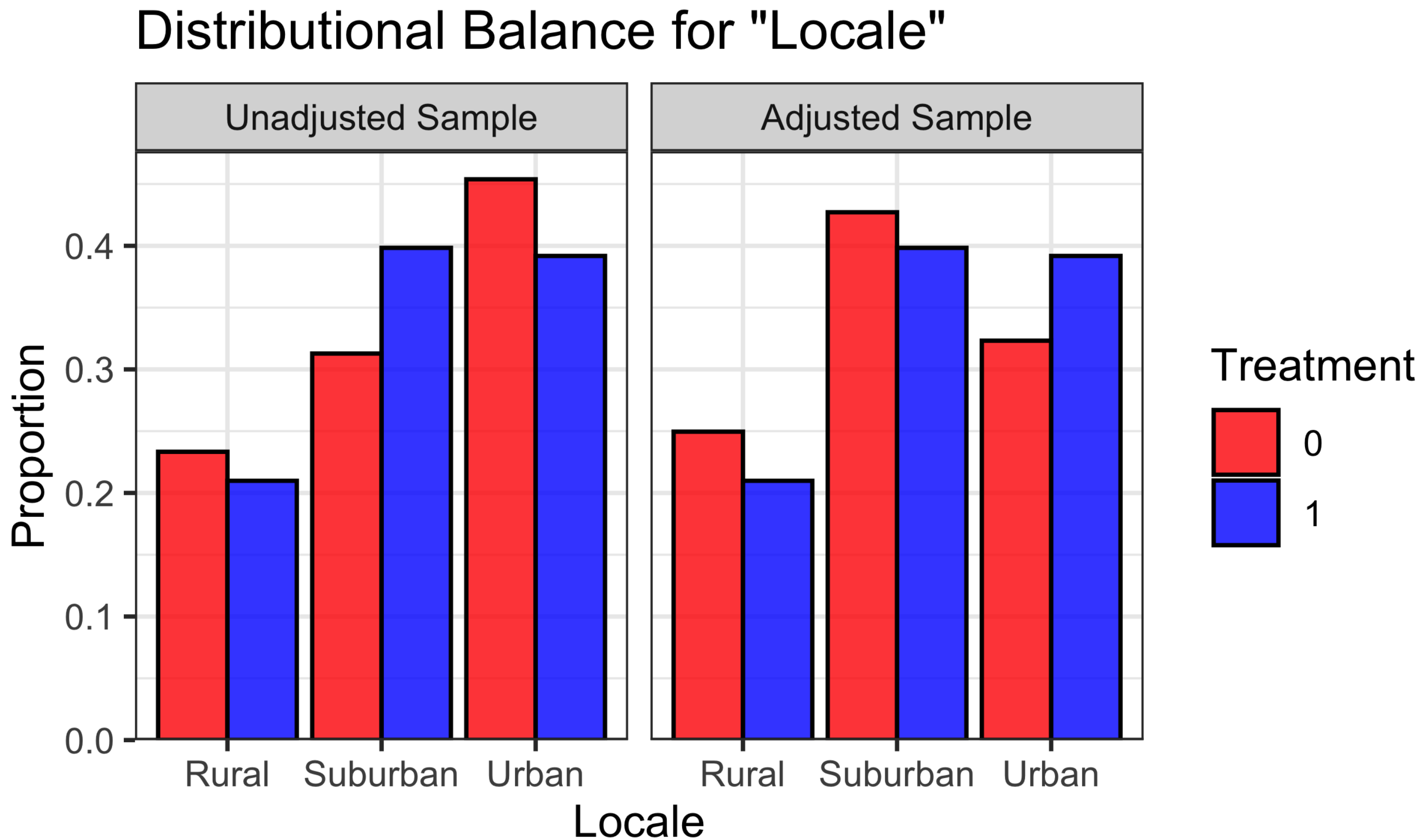


Weighting: Bar Plot (Locale)

If the covariate is binary, `bal.plot` will create bar plots.

```
bal.plot(f_lin,
         data = Algebra_dat,
         weights = "att_wt",
         method = "weighting",
         var.name = "Locale",
         which = "both",
         alpha = .2,
         position = "stack", #alpha and position don't get passed to
         colors = c("red", "blue")) +
  theme_bw()
```

Weighting: Bar Plot (Locale)



Weighting: Int and Sq Terms

Balance should be evaluated for two-way interactions and squared terms of continuous covariates (Stuart, 2010). We can add `int = TRUE` and `poly = 2` as arguments. Note that the output presents variance ratios of squared terms which doesn't mean what it should.

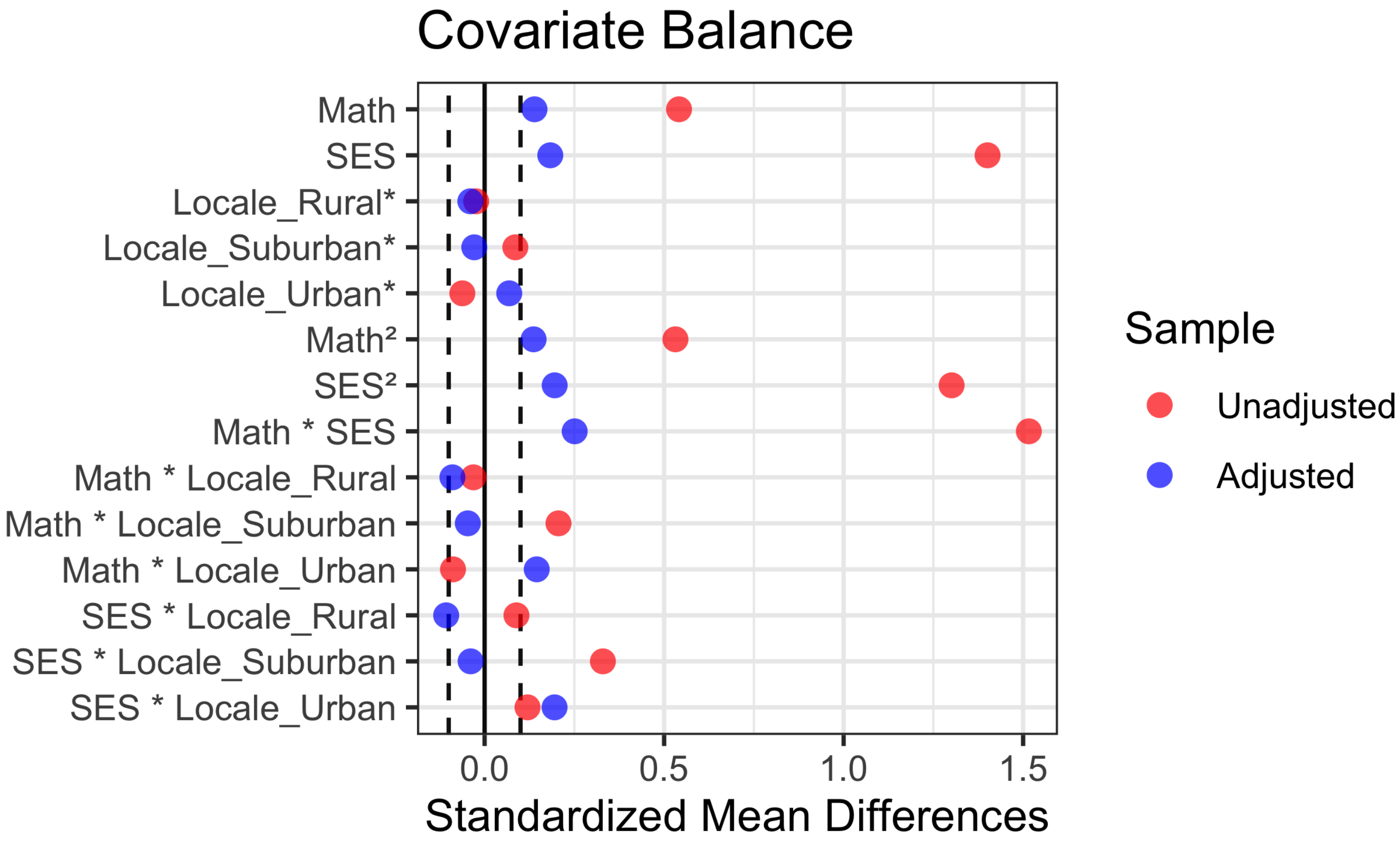
```
b_w2 <- bal.tab(f_lin,
                data = Algebra_dat,
                weights = "att_wt",
                method = "weighting",
                int = TRUE,  # add interaction terms
                poly = 2, # add sq terms
                disp.v.ratio = TRUE,
                un = TRUE)

b_w2
```

```
## Balance Measures
##                              Type Diff.Un V.Ratio.Un Diff.Adj V.Ratio.Adj
## Math                      Contin.  0.5415     1.0440   0.1390      0.9415
## SES                       Contin.  1.4009     1.2191   0.1829      1.0423
## Locale_Rural               Binary -0.0235            -0.0398            
## Locale_Suburban            Binary  0.0855            -0.0288            
## Locale_Urban               Binary -0.0620             0.0686            
## Math²                     Contin.  0.5315     1.1252   0.1364      0.9713
## SES²                      Contin.  1.3007     1.6729   0.1951      1.0783
## Math * SES                Contin.  1.5159     1.5552   0.2509      1.3351
## Math * Locale_Rural       Contin. -0.0311     1.0174  -0.0897      0.8926
## Math * Locale_Suburban Contin.  0.2058     1.2359  -0.0468      0.9861
## Math * Locale_Urban       Contin. -0.0881     1.0437   0.1452      1.0940
## SES * Locale_Rural        Contin.  0.0886     1.6501  -0.1071      0.8389
## SES * Locale_Suburban     Contin.  0.3295     1.9538  -0.0389      1.0024
## SES * Locale_Urban        Contin.  0.1195     1.7533   0.1948      1.3097
##
## Effective sample sizes
##            Control Treated
## Unadjusted 390.000     610
## Adjusted    57.511     610
```

Int & Sq Terms: Love Plot

```
love.plot(b_w2,
          threshold = .1,
          colors = c("red", "blue"),
          size = 3,
          alpha = .7,
          stars = "raw") +
  theme_bw()
```

Int & Sq Terms: Love Plot



Iterate...

If balance is not adequate, respecify propensity score model (e.g., add interactions, squared terms) and assess balance again.

Stratification: Calculation

```
# Quintiles based on treatment group
ATT_quint <- with(Algebra_dat, quantile(ps[D==1], seq(0,1,0.2)))

Algebra_dat$quintile <- cut(Algebra_dat$ps, ATT_quint,
                            labels = c("A","B","C","D","E"),
                            include.lowest = TRUE)
```

Stratification: Balance

Specify the subclass and method:

```
b_s1 <- bal.tab(f_lin,
                data = Algebra_dat,
                subclass = "quintile",  # add the variable that defi
                method = "subclassification", # specify method
                disp.subclass = TRUE,
                disp.v.ratio = TRUE,
                un = TRUE)

b_s1
```

```
## Balance by subclass
##  - - - Subclass A - - -
##                      Type Diff.Adj V.Ratio.Adj
## Math              Contin.   0.1174      1.0884
## SES               Contin.   0.2457      1.0510
## Locale_Rural       Binary   0.0172
## Locale_Suburban    Binary   0.0130
## Locale_Urban       Binary  -0.0302
##
##  - - - Subclass B - - -
##                      Type Diff.Adj V.Ratio.Adj
## Math              Contin.  -0.0692      1.0314
## SES               Contin.   0.1215      1.0080
## Locale_Rural       Binary   0.0193
## Locale_Suburban    Binary   0.0468
## Locale_Urban       Binary  -0.0661
##
##  - - - Subclass C - - -
##                      Type Diff.Adj V.Ratio.Adj
## Math              Contin.   0.2728      0.7489
## SES               Contin.  -0.0194      0.6920
## Locale_Rural       Binary  -0.0893
## Locale_Suburban    Binary  -0.2013
## Locale_Urban       Binary   0.2905
##
##  - - - Subclass D - - -
##                      Type Diff.Adj V.Ratio.Adj
```

Stratification: Across Subclasses

```
## Balance measures across subclasses
##                    Type Diff.Un V.Ratio.Un Diff.Adj V.Ratio.Adj
## Math             Contin.  0.5415     1.0440   0.0612      1.0376
## SES              Contin.  1.4009     1.2191   0.1537      0.9329
## Locale_Rural      Binary -0.0235            -0.0603
## Locale_Suburban   Binary  0.0855            -0.0212
## Locale_Urban      Binary -0.0620             0.0815
##
## Sample sizes by subclass
##            A   B   C   D   E  All
## Control  258  62  18   6   2  390
## Treated  122 122 122 122 122  610
## Total    380 184 140 128 124 1000
```

Stratification: Love Plot Fail

```
love.plot(b_s1)
```

```
## Error in is_not_null(facet): object 'facet' not found
```

Matching: Balance

Matching without replacement

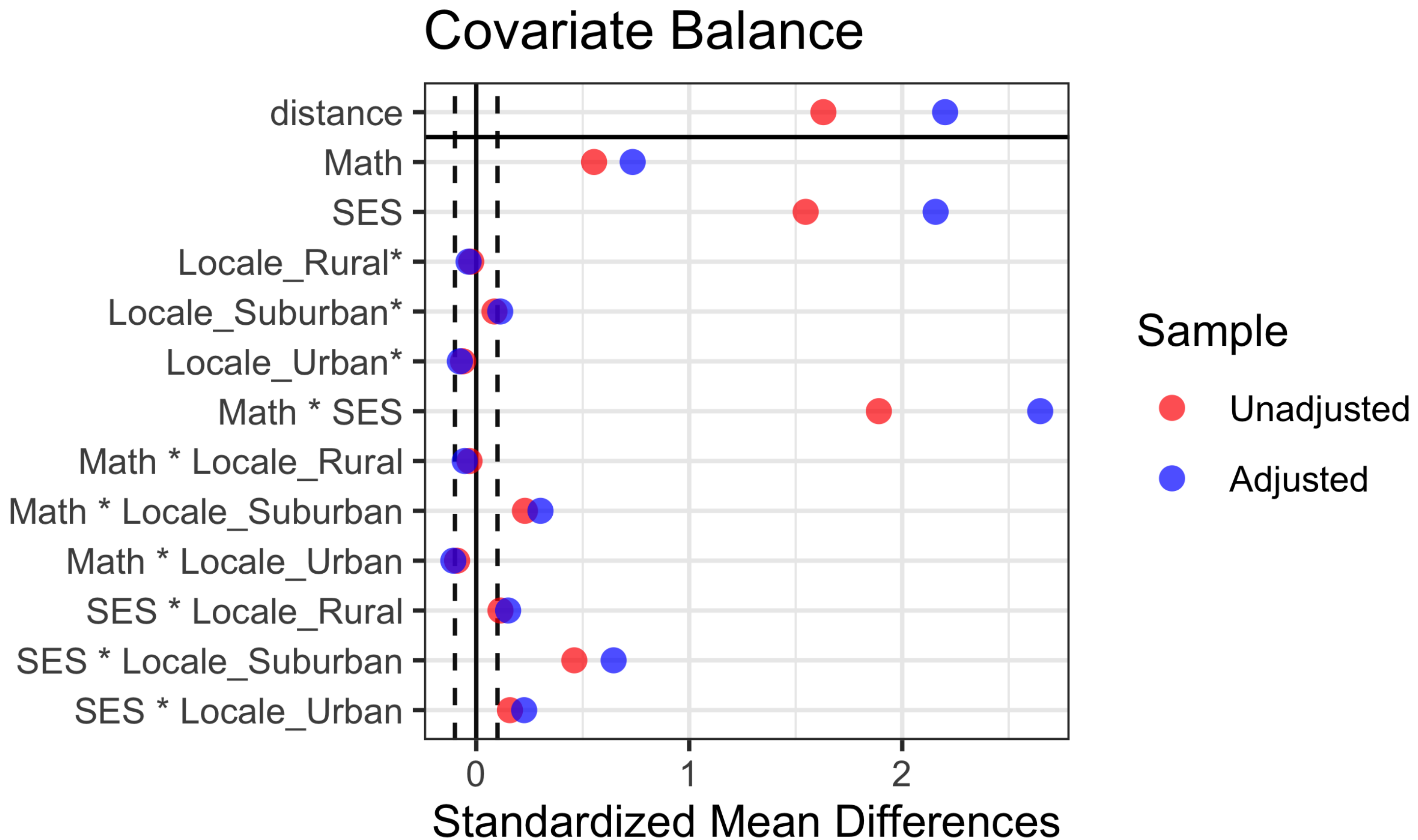
- More treated than untreated units- discards treated units
- Depends on order of treated units
- Starts with treated units with highest propensity scores
  - Throws out those with lower- even though better match

```
Algebra_dat_org <- as.data.frame(Algebra_dat_org)

m_out <- matchit(D ~ Math + SES + Locale , data = Algebra_dat_org,
                  method = "nearest", distance = "logit")

b_m1 <- bal.tab(m_out,    # the output of matchit
                un = TRUE,
                int = TRUE)
```

Matching: Love Plot

```r
love.plot(b_m1,
          threshold = .1,
          colors = c("red", "blue"),
          size = 3,
          alpha = .7,
          stars = "raw") +
  theme_bw()
```

Matching: Love Plot



Better Match

- Matching with replacement- match an untreated unit more than once
- Caliper- maximum tolerated difference- prevents matching with whatever nearest unit available

```
better_match <- matchit(D ~ Math + SES + Locale,
                        data = Algebra_dat_org,
                        distance = "logit",
                        method = "nearest",
                        replace = TRUE,
                        caliper = 0.1)
```

Better Match: Balance Table

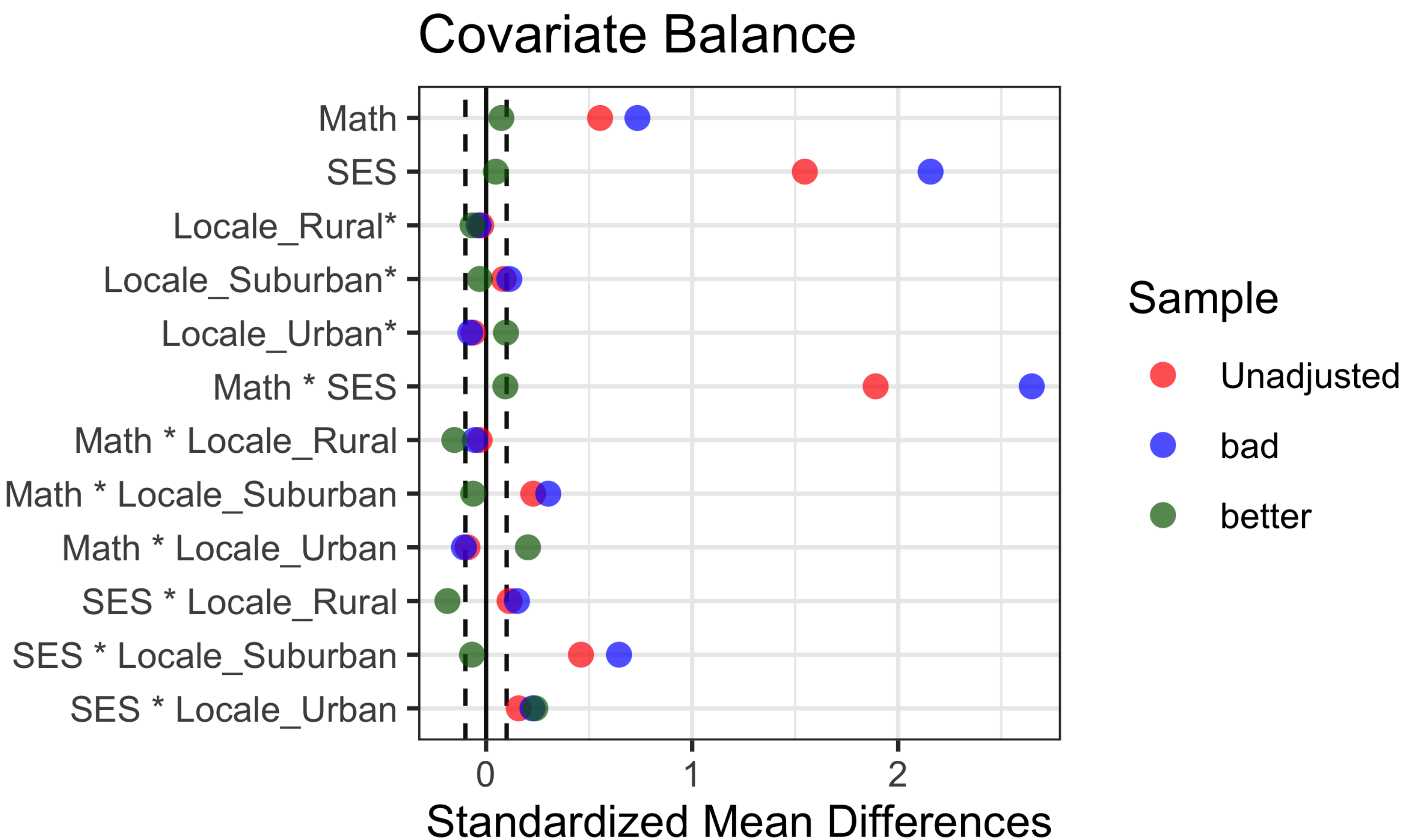
- We can compare the two different matching methods
- The `get.w` function will extract weights from `matchit` results
- Specify data frame with weights and specify method as "weighting"

```
b_m2 <- bal.tab(f_lin,
                data = Algebra_dat,
                weights = data.frame(    # create a data frame with we
                    bad = get.w(m_out),
                    better = get.w(better_match)),
                method = "weighting",
                int = TRUE)
```

Better Match: Love Plot

```
love.plot(b_m2,
          colors = c("red", "blue", "darkgreen"),
          size = 3,
          alpha = .7,
          threshold = .1,
          stars = "raw") +
  theme_bw()
```

Better Match: Love Plot

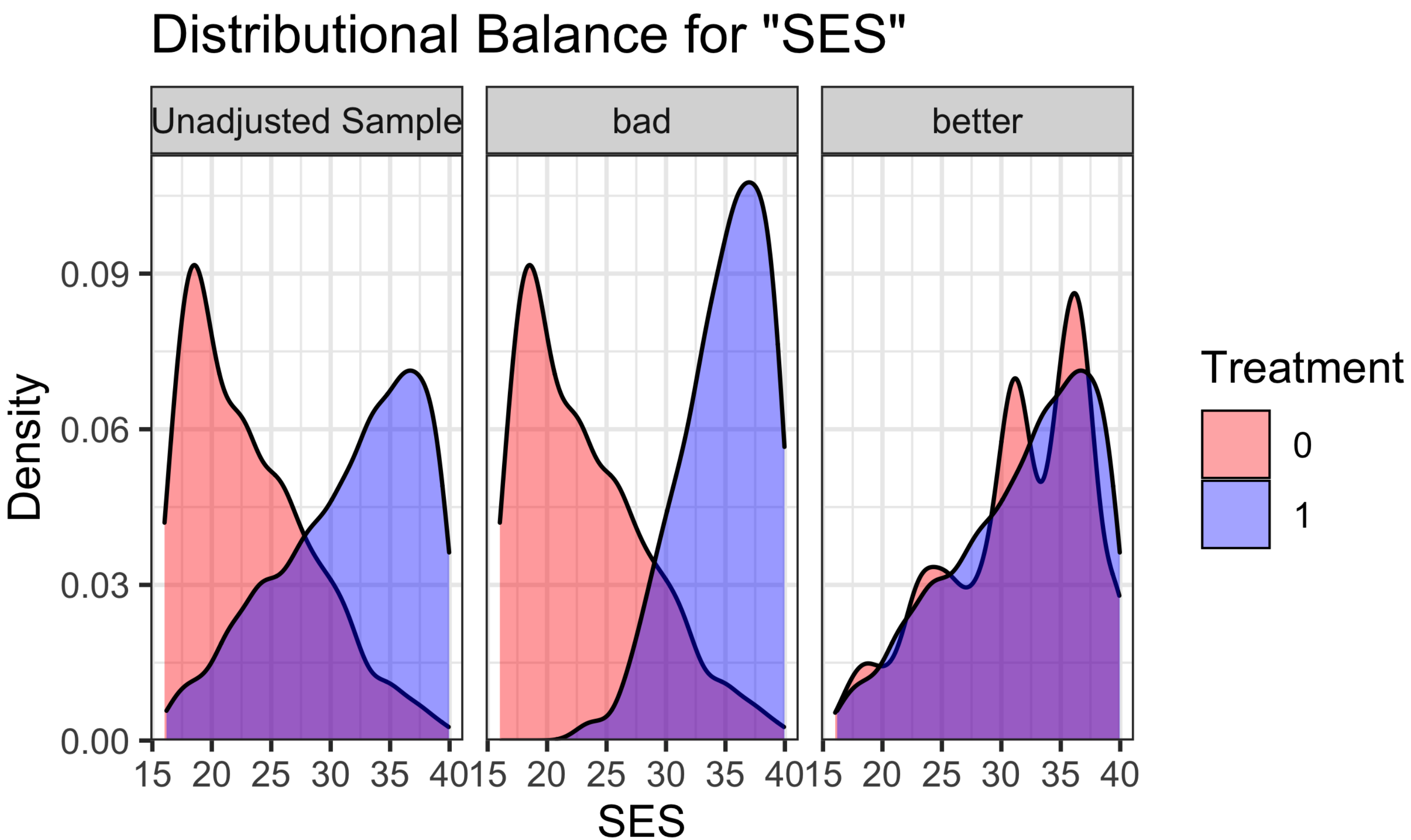


Better Match: Density Plots

Can compare the two matching methods in `bal.plot` too. Here creating density plots for SES.

```
bal.plot(f_lin,
         data = Algebra_dat,
         weights = data.frame(  # create a data frame with weights
           bad = get.w(m_out),
           better = get.w(better_match)),
         method = "weighting",
         var.name = "SES", # specify var name
         colors = c("red", "blue"),
         which = "both",
         alpha = .2) + # alpha doesn't work
  theme_bw()
```

Better Match: Density Plots



Matching: Effective Sample Size

```
b_m2 <- bal.tab(f_lin,
                data = Algebra_dat,
                  weights = data.frame(
                    bad = get.w(m_out),
                    better = get.w(better_match)),
                method = "weighting",
                int = TRUE)

b_m2
```

Matching: Effective Sample Size

```
## Balance Measures
##                               Type Diff.bad Diff.better
## Math                       Contin.   0.7345      0.0742
## SES                        Contin.   2.1569      0.0475
## Locale_Rural                Binary  -0.0359     -0.0656
## Locale_Suburban             Binary   0.1128     -0.0311
## Locale_Urban                Binary  -0.0769      0.0967
## Math * SES                 Contin.   2.6481      0.0935
## Math * Locale_Rural        Contin.  -0.0540     -0.1540
## Math * Locale_Suburban     Contin.   0.3019     -0.0625
## Math * Locale_Urban        Contin.  -0.1075      0.2032
## SES * Locale_Rural         Contin.   0.1500     -0.1875
## SES * Locale_Suburban      Contin.   0.6452     -0.0683
## SES * Locale_Urban         Contin.   0.2251      0.2392
##
## Effective sample sizes
##        Control Treated
## All    390.000     610
## bad    390.000     390
## better  33.093     610
```

References

Ali, M. S., Groenwold, R. H. H., Pestman, W. R., Belitser, S. V., Roes, K. C. B., Hoes, A. W., … Klungel, O. H. (2014). Propensity score balance measures in pharmacoepidemiology: a simulation study. Pharmacoepidemiology and Drug Safety, 23(8), 802–811. https://doi.org/10.1002/pds.3574

Austin, P. C. (2008). A critical appraisal of propensity-score matching in the medical literature between 1996 and 2003. Statistics in Medicine, 27(12), 2037–2049. http://doi.org/10.1002/sim.3150

Austin, P. C. (2009). Balance diagnostics for comparing the distribution of baseline covariates between treatment groups in propensity-score matched samples. Statistics in Medicine, 28(25), 3083–3107. http://doi.org/10.1002/sim.3697

Austin, P. C. (2011). An Introduction to Propensity Score Methods for Reducing the Effects of Confounding in Observational Studies. Multivariate Behavioral Research, 46(3), 399–424. http://doi.org/10.1080/00273171.2011.568786

References

Austin, P. C., & Stuart, E. A. (2015). Moving towards best practice when using inverse probability of treatment weighting (IPTW) using the propensity score to estimate causal treatment effects in observational studies. Statistics in Medicine, 34(28), 3661–3679. http://doi.org/10.1002/sim.6607

Greifer, N. (2020). cobalt: Covariate Balance Tables and Plots. R package version 3.0.0.

Imai, K., King, G., & Stuart, E. A. (2008). Misunderstandings between experimentalists and observationalists about causal inference. Journal of the royal statistical society: series A (statistics in society), 171(2), 481-502.

Ridgeway, G., McCaffrey, D., Morral, A., Burgette, L., & Griffin, B. A. (2016). Toolkit for Weighting and Analysis of Nonequivalent Groups: A tutorial for the twang package. R Vignette. RAND. Retrieved from https://CRAN.R-project.org/package=twang/vignettes/twang.pdf

Rubin, D. B. (2001). Using Propensity Scores to Help Design Observational Studies: Application to the Tobacco Litigation. Health Services and Outcomes Research Methodology, 2(3-4), 169–188. http://doi.org/10.1023/A:1020363010465

References

Stuart, E. A. (2008). Developing practical recommendations for the use of propensity scores: Discussion of “A critical appraisal of propensity score matching in the medical literature between 1996 and 2003” by Peter Austin, Statistics in Medicine. Statistics in Medicine, 27(12), 2062–2065. http://doi.org/10.1002/sim.3207

Stuart, E. A. (2010). Matching Methods for Causal Inference: A Review and a Look Forward. Statistical Science, 25(1), 1–21. http://doi.org/10.1214/09-STS313

Stuart, E. A., Lee, B. K., & Leacy, F. P. (2013). Prognostic score-based balance measures can be a useful diagnostic for propensity score methods in comparative effectiveness research. Journal of Clinical Epidemiology, 66(8), S84. http://dx.doi.org/10.1016/j.jclinepi.2013.01.013

Thoemmes, F. J., & Kim, E. S. (2011). A Systematic Review of Propensity Score Methods in the Social Sciences. Multivariate Behavioral Research, 46(1), 90–118. http://doi.org/10.1080/00273171.2011.540475