

cobalt

## Covariate Balance Tables and Plots

Pierce & Megha

November 9, 2017

# 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)
  - `coba1t` 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), control group (ATC), 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:

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

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)

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

- Proportionally larger weights lead to lower ESS (Notes from Tuesday)
- Large variance of weighted mean
- Loss of precision
- `cobalt` calculates ESS

# cobalt demonstration

# Libraries

The CRAN version does not work with tibbles but the development version does (but then major functions don't run). Here, we are using the CRAN version.

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

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

# Data

If you use readr or dplyr, make sure to convert the tibble to a dataframe if using the CRAN version of cobalt.

```
Algebra_dat <-  
  read_csv("datasets/8th-grade-Algebra-data.csv") %>%  
  mutate(Locale = factor(Locale, levels = c("R", "S", "U"), labels = c(  
  
# convert to data frame  
Algebra_dat <- Algebra_dat_org <- data.frame(Algebra_dat)
```

# Formula

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

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

```
## D ~ Math + SES + Locale  
## <environment: 0x7fe14ec0db70>
```



# 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, data = Algebra_dat, weights = "att_wt",  
        method = "weighting", estimand = "ATT",  
        m.threshold = .1)
```

```

## 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", estimand = "ATT",  
                disp.v.ratio = TRUE, un = TRUE))
```

## ## 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

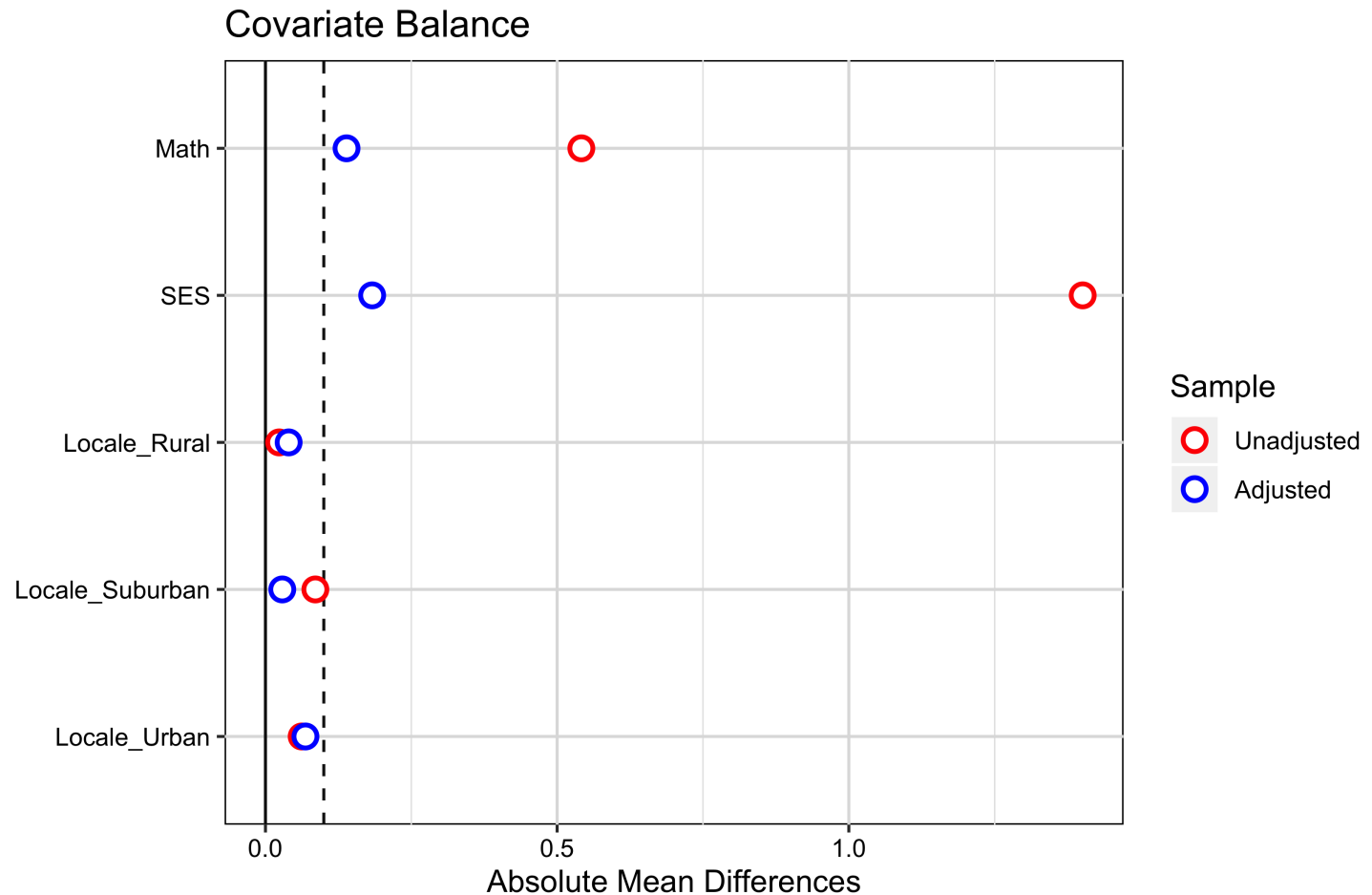
##		Control	Treated
##	Unadjusted	390.000	610
##	Adjusted	57.511	610

# Weighting: Love Plot

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

```
love.plot(b_w1, threshold = .1, colors = c("red", "blue"),  
          size = 1.5)
```

# Weighting: Love Plot

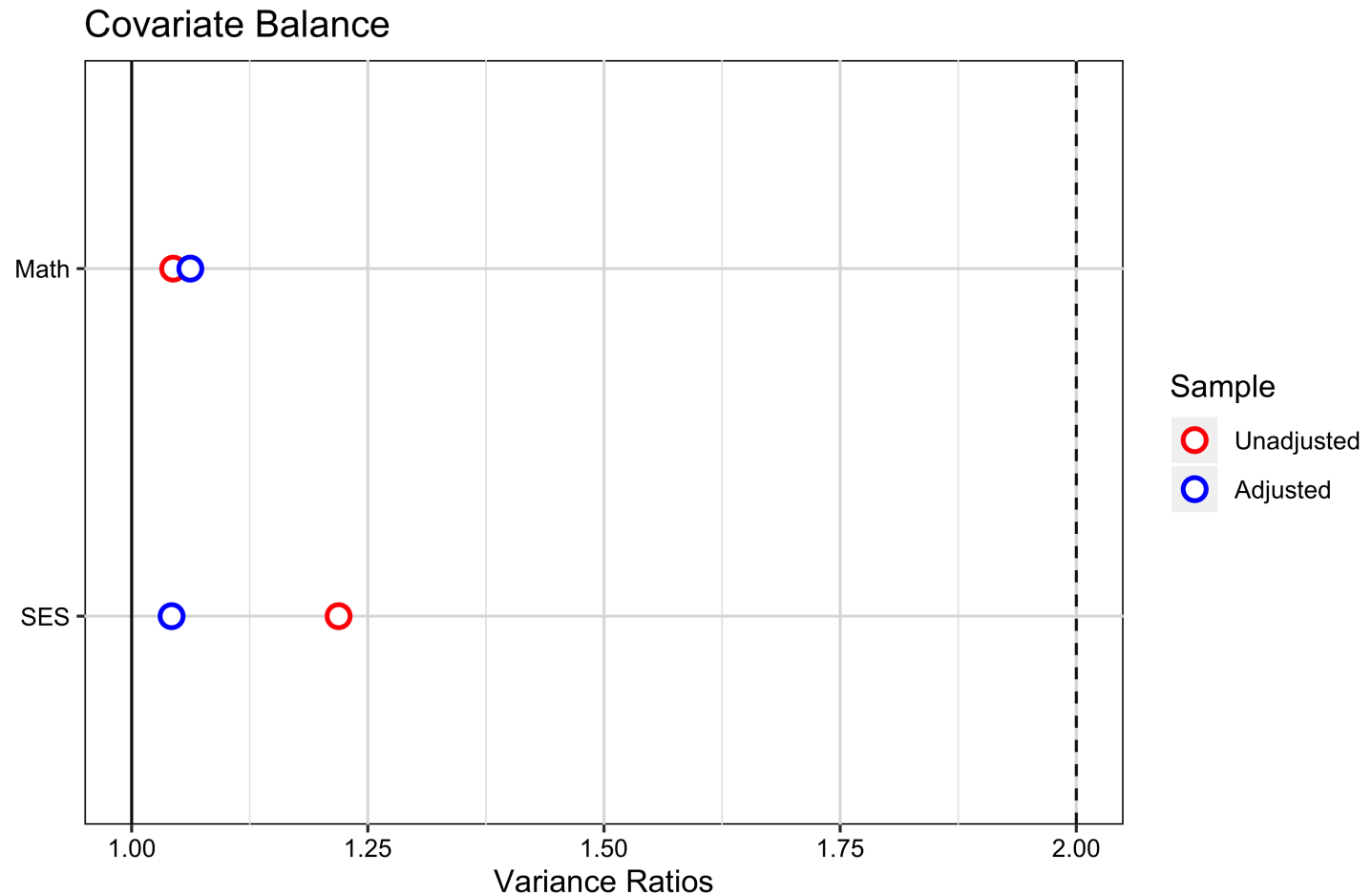


# Weighting: Love Plot for Variance Ratios

```
love.plot(b_w1, threshold = 2, colors = c("red", "blue"),  
          size = 1.5, stat = "variance.ratios")
```



# 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", estimand = "ATT",  
        var.name = "SES", which = "both") +  
  scale_fill_manual(values = c("red", "blue")) +  
  theme_minimal()
```

# Weighting: Density Plot (SES)

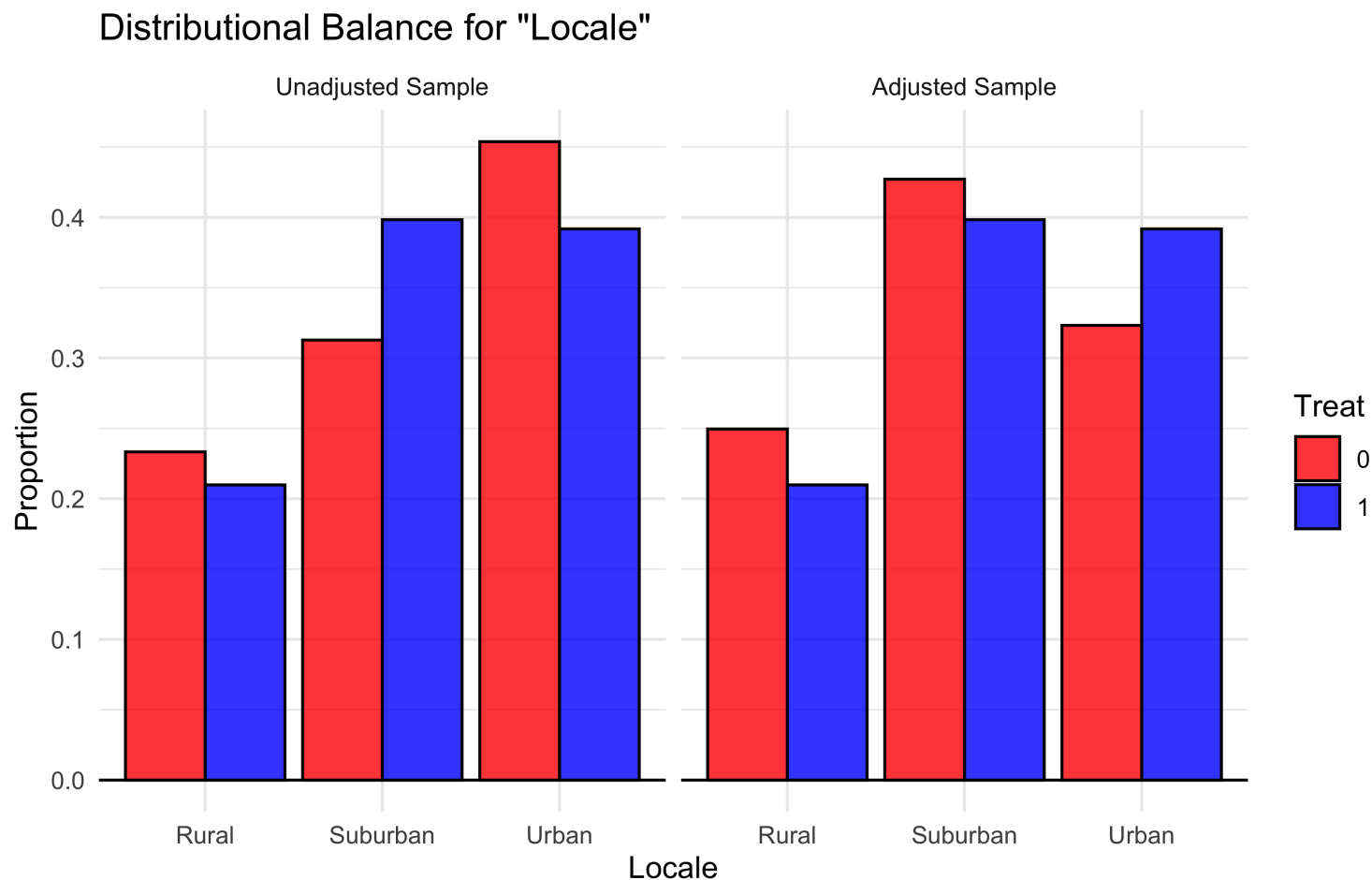
```
## Scale for 'fill' is already present. Adding another scale for 'fill',  
## which will replace the existing scale.
```

# 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", estimand = "ATT",  
         var.name = "Locale", which = "both") +  
  scale_fill_manual(values = c("red", "blue")) +  
  theme_minimal()
```

# 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` as an argument.

```
(b_w2 <- bal.tab(f_lin, data = Algebra_dat, weights = "att_wt",  
  method = "weighting", int = TRUE,  
  estimand = "ATT", disp.v.ratio = TRUE, un = TRUE))
```

## ## 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	
## 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

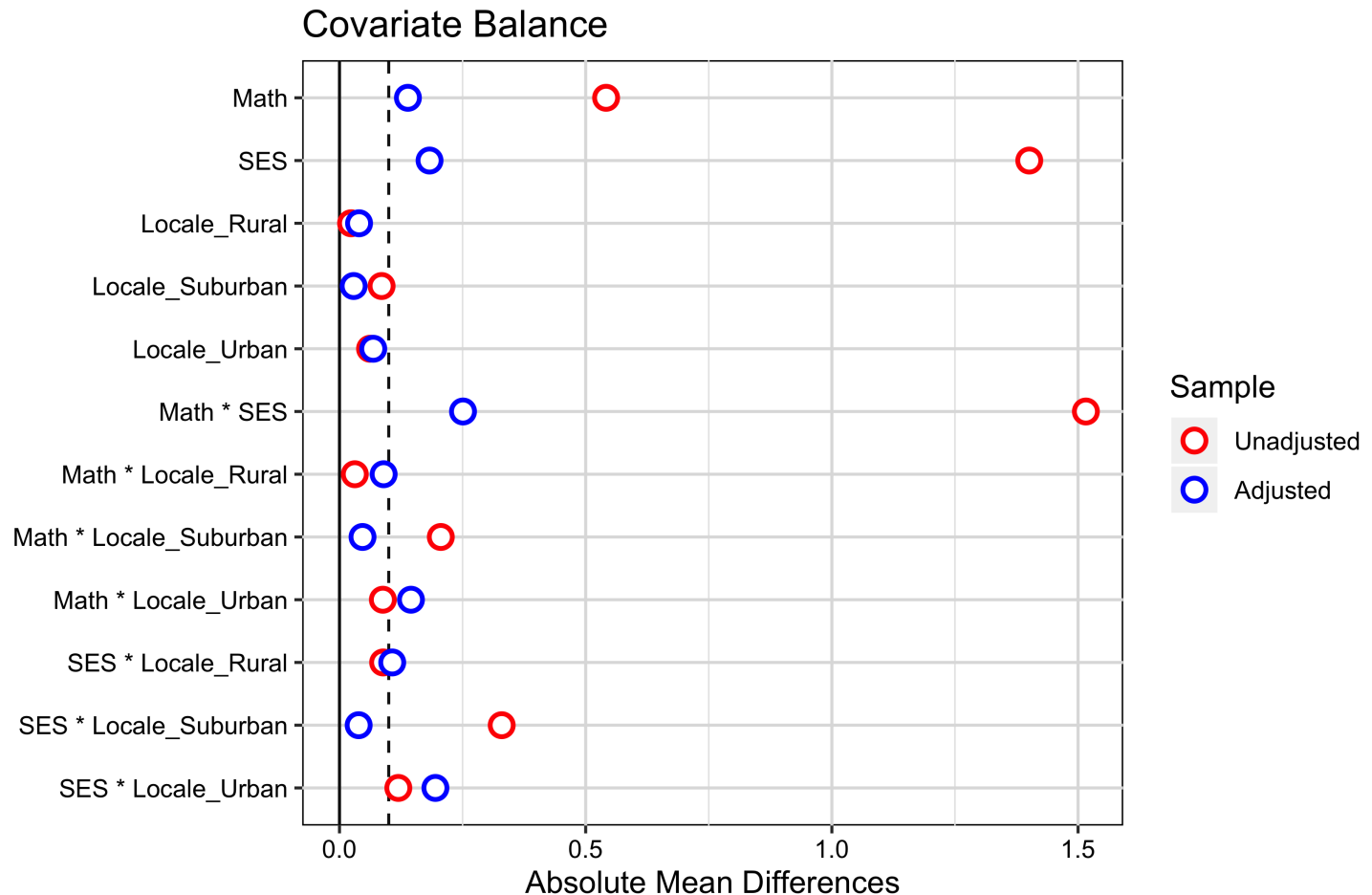
##	Control	Treated
## Unadjusted	390.000	610
## Adjusted	57.511	610

# Int & Sq Terms: Love Plot

```
love.plot(b_w2, threshold = .1, colors = c("red", "blue"),  
          size = 1.5)
```



# 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",  
  method = "subclassification", disp.subclass = TRUE,  
  estimand = "ATT",  
  disp.v.ratio = TRUE, un = TRUE))
```

```
## 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
## Math      Contin.  0.5415      1.0440  0.0612
## SES        Contin.  1.4009      1.2191  0.1537
## Locale_Rural Binary -0.0235              -0.0603
## Locale_Suburban Binary  0.0855              -0.0212
## Locale_Urban Binary -0.0620              0.0815
##
## Sample sizes
##           Subclass A Subclass B Subclass C Subclass D Subclass E
## Control          258          62          18          6          2
## Treated           122          122          122          122          122
## Total             380          184          140          128          124
```

**Does not show variance ratios across subclasses.**

# 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

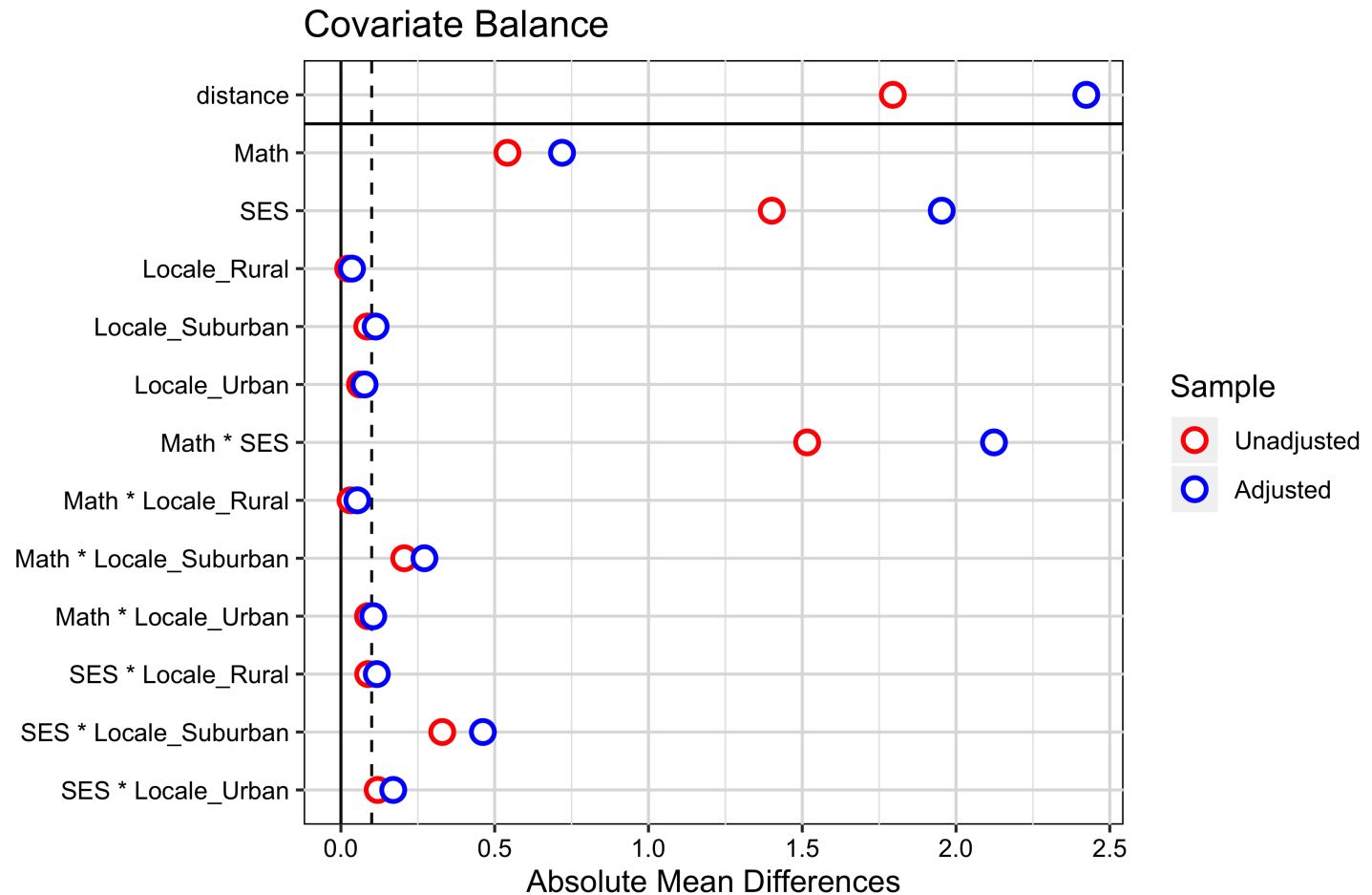
```
m_out <- matchit(D ~ Math + SES + Locale , data = Algebra_dat_org,  
                 method = "nearest", distance = "logit")  
  
b_m1 <- bal.tab(m_out, un = TRUE, estimand = "ATT",  
               int = TRUE)
```

# Matching: Love Plot

```
love.plot(b_m1, threshold = .1, colors = c("red", "blue"),  
          size = 1.5)
```



# 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

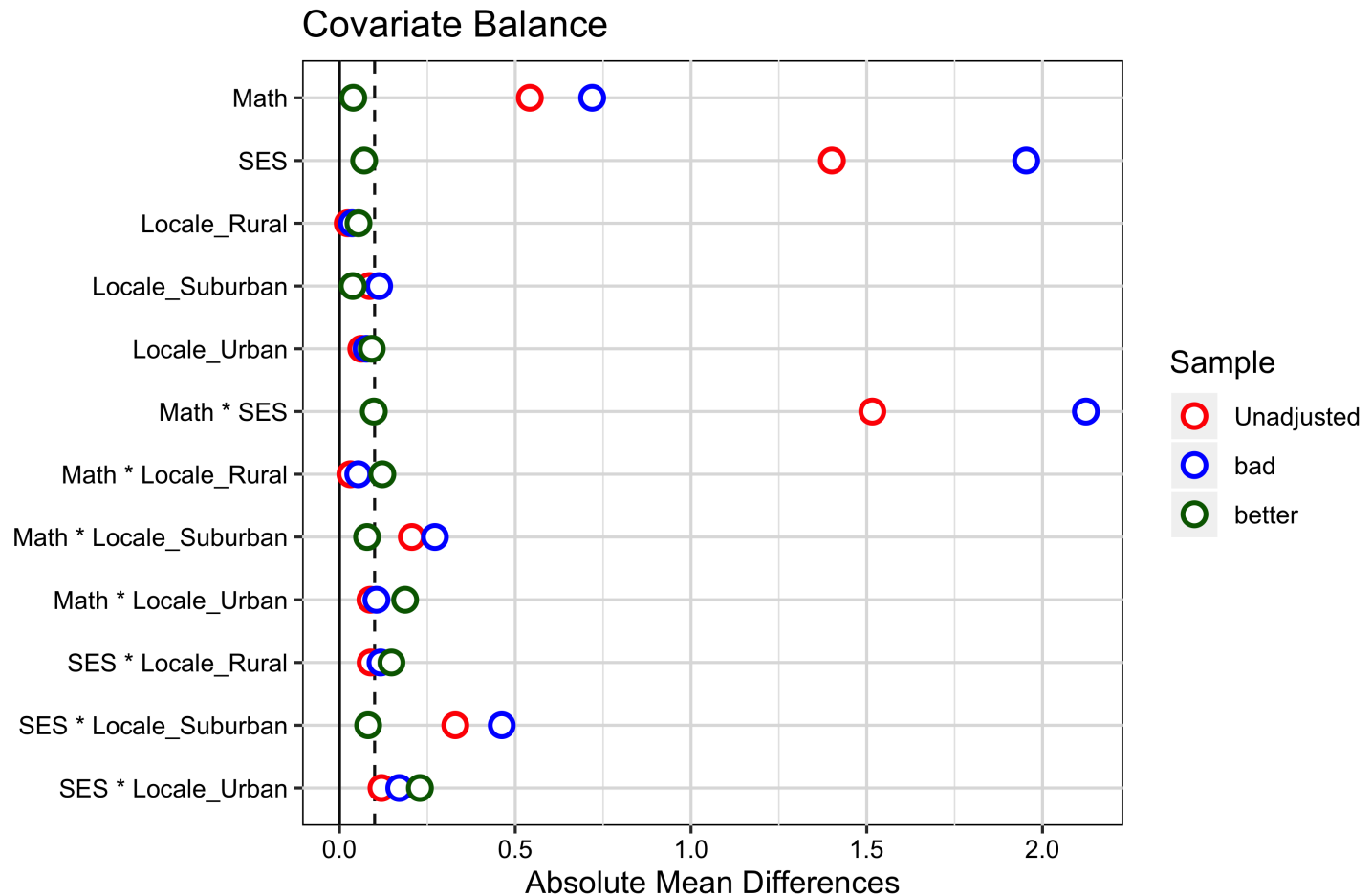
- We can compare the two different matching methods
- The `get.w` function will extract weights from `matchit` results
- Specify data frame with weights and specify method as "weighting"

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

# Better Match: Love Plot

```
love.plot(b_m2, colors = c("red", "blue", "darkgreen"),  
          size = 1.5, threshold = .1)
```

# 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(bad = get.w(m_out),  
                              better = get.w(better_match)),  
         method = "weighting", var.name = "SES", which = "both") +  
  scale_fill_manual(values = c("red", "blue")) +  
  theme_minimal()
```

# Better Match: Density Plots

```
## Scale for 'fill' is already present. Adding another scale for 'fill',  
## which will replace the existing scale.
```

# 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, estimand = "ATT")
```



# Matching: Effective Sample Size

## Balance Measures

##	Type	Diff.bad	Diff.better
## Math	Contin.	0.7189	0.0392
## SES	Contin.	1.9535	0.0702
## Locale_Rural	Binary	-0.0359	-0.0541
## Locale_Suburban	Binary	0.1128	-0.0377
## Locale_Urban	Binary	-0.0769	0.0918
## Math * SES	Contin.	2.1235	0.0977
## Math * Locale_Rural	Contin.	-0.0535	-0.1220
## Math * Locale_Suburban	Contin.	0.2715	-0.0785
## Math * Locale_Urban	Contin.	-0.1052	0.1867
## SES * Locale_Rural	Contin.	0.1167	-0.1478
## SES * Locale_Suburban	Contin.	0.4616	-0.0816
## SES * Locale_Urban	Contin.	0.1700	0.2287

##

## Effective sample sizes

##	Control	Treated
## All	390.000	610
## bad	390.000	390
## better	32.606	610

# matchit Output

```
bal.tab(better_match)
```

```
## Call
## matchit(formula = D ~ Math + SES + Locale, data = Algebra_dat_org,
##         method = "nearest", distance = "logit", replace = TRUE, caliper = 0.1)
##
## Balance Measures
##
##           Type Diff.Adj
## distance      Distance  0.0132
## Math           Contin.  0.0392
## SES            Contin.  0.0702
## Locale_Rural    Binary -0.0541
## Locale_Suburban Binary -0.0377
## Locale_Urban    Binary  0.0918
##
## Sample sizes
##           Control Treated
## All           390      610
## Matched       169      610
## Unmatched     221        0
```

Thank you!!

# 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. (2017). 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*, 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>