A Visual Introduction to Propensity Score Analysis with R

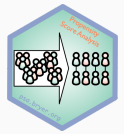
New York Open Statistical Programming Meetup

Jason Bryer, Ph.D.

Last updated: November 14, 2023

Agenda

- Overview of causal inference
  - Counterfactuals
  - Randomized control trials
  - Rubin's Causal Model
- Propensity Score Analysis in Three Phases
  - Phase I: Estimate propensity scores and check balance
  - Phase II: Estimate causal effects
    - Matching
    - Stratification
    - Weighting
  - Phase III: Check sensitivity to unobserved confounders
    - Sensitivity analysis
    - Bootstrapping PSA
- Advanced topics in PSA
  - Matching for Non-Binary Treatments
  - Multilevel PSA



Getting Started in R

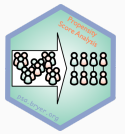
Installing the `psa` package from Github with `dependencies = 'Enhances'` should install all the packages we will use in this workshop.

```
install.packages('devtools')
remotes::install_github('jbryer/psa', build_vignettes = TRUE, dependencies = 'Enhances')
```

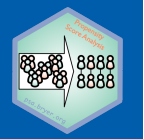
You can download an R script to work through the commands in this slide deck at:

https://github.com/jbryer/psa/blob/master/Slides/Intro_PSA.R

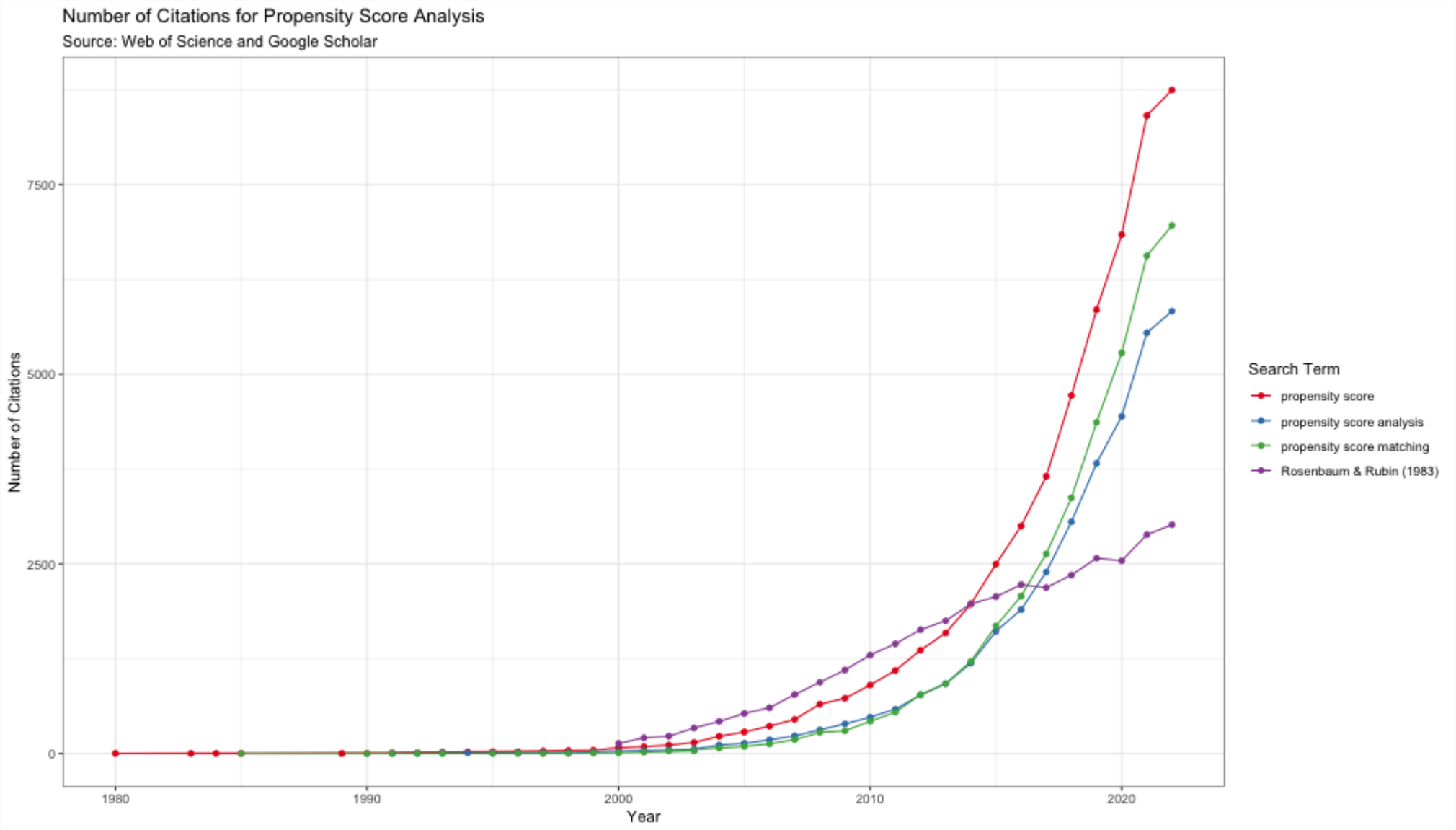
Other resources are available on Github here: https://github.com/jbryer/psa



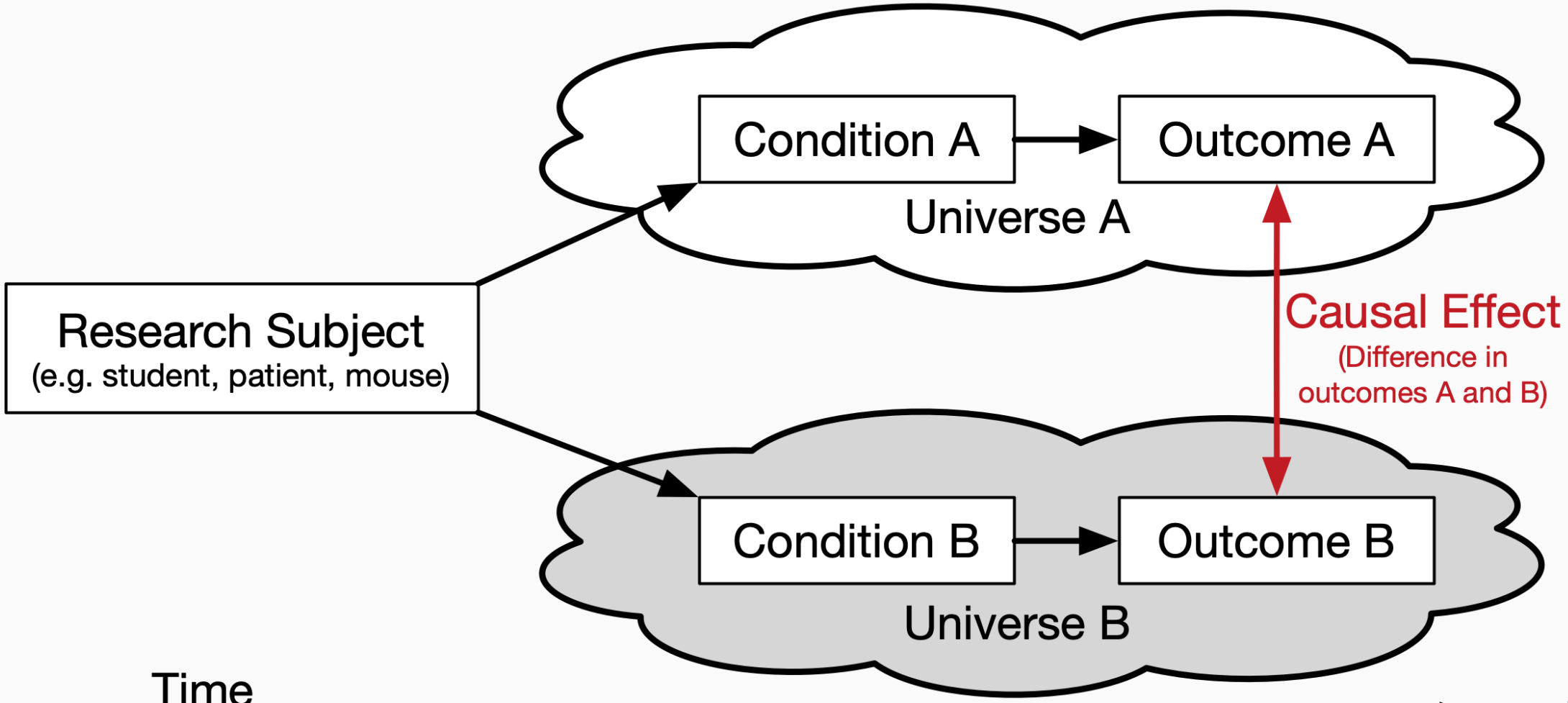
Overview of Causal Inference



Popularity of Propensity Score Analysis



Counterfactuals



The only difference between universes A and B is conditions A and B and, potentially, outcomes A and B.

The Randomized Experiment

Considered to be the *gold standard* for estimating causal effects.

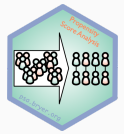
- Effects can be estimated using simple means between groups, or blocks in randomized block design.
- Randomization presumes unbiasedness and balance between groups.

However, randomization is often not feasible for many reasons, especially in educational contexts.

The **strong ignorability assumption** states that:

$$(Y_i(1), Y_i(0)) \perp\!\!\!\perp T_i | X_i = x$$

for all X_i.

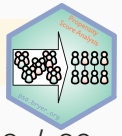


RCT Example

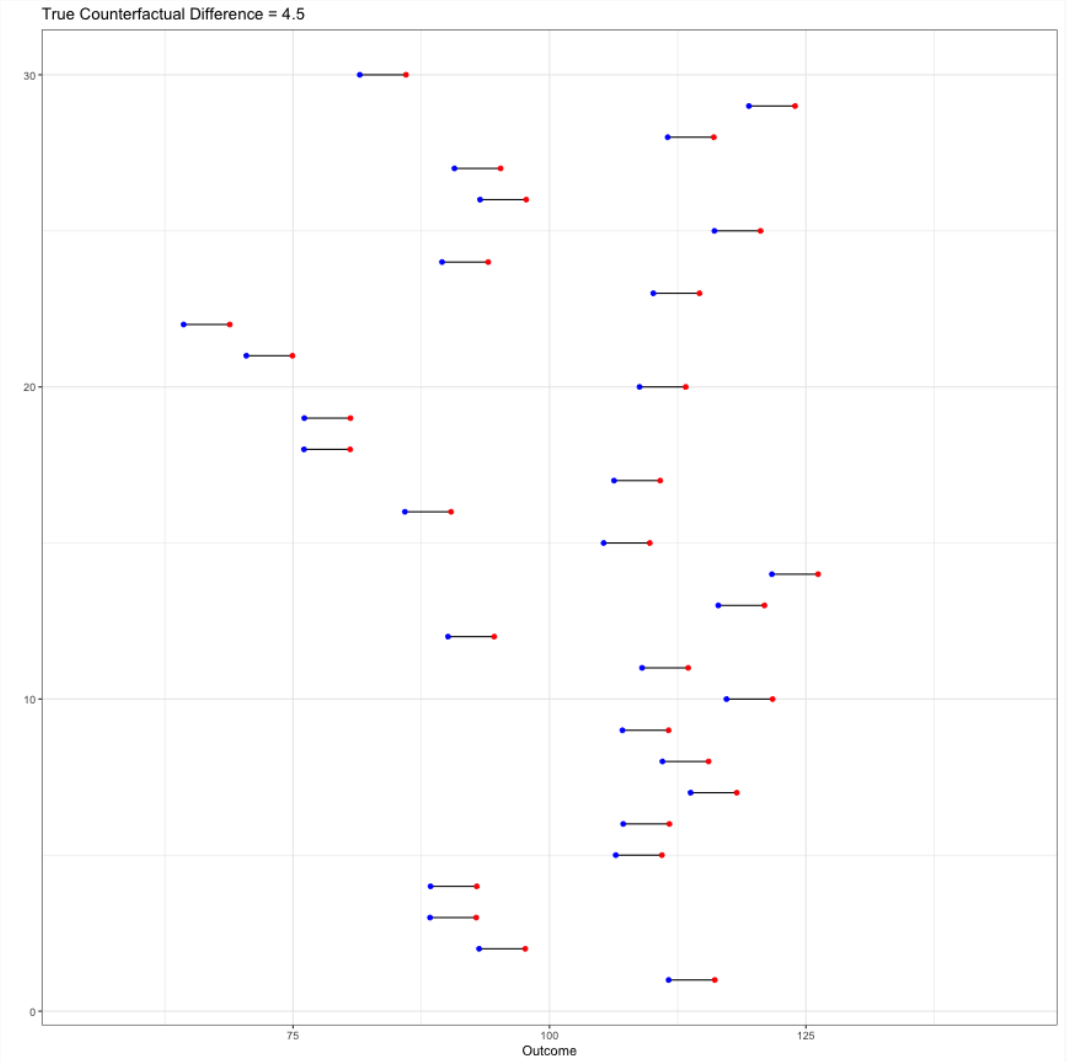
```
set.seed(2112)
pop.mean <- 100
pop.sd <- 15
pop.es <- .3
n <- 30
thedata <- data.frame(
    id = 1:30,
    center = rnorm(n, mean = pop.mean, sd = pop.sd),
    stringsAsFactors = FALSE
)
val <- pop.sd * pop.es / 2
thedata$placebo <- thedata$center - val
thedata$treatment <- thedata$center + val
thedata$diff <- thedata$treatment - thedata$placebo
thedata$RCT_Assignment <- sample(c('placebo', 'treatment'), n, replace = TRUE)
thedata$RCT_Value <- as.numeric(apply(thedata, 1,
                    FUN = function(x) { return(x[x['RCT_Assignment']]) }))
head(thedata, n = 3)
```

```
##   id    center   placebo treatment diff RCT_Assignment RCT_Value
## 1  1 113.86506 111.61506 116.11506  4.5      treatment 116.11506
## 2  2  95.38746  93.13746  97.63746  4.5      treatment  97.63746
## 3  3  90.60380  88.35380  92.85380  4.5      treatment  92.85380
```

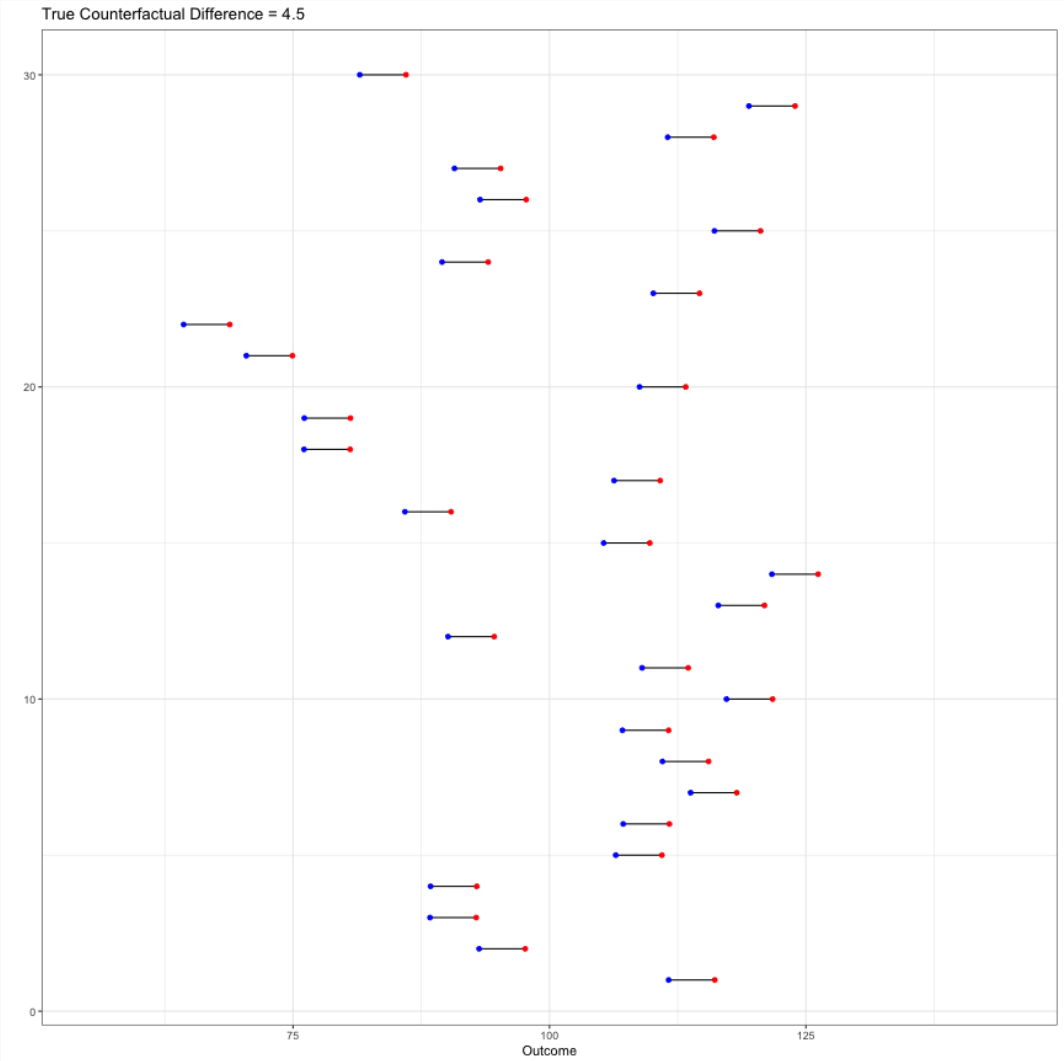
```
tab.out <- describeBy(thedata$RCT_Value, group = thedata$RCT_Assignment, mat = TRUE, skew = FALSE)
```

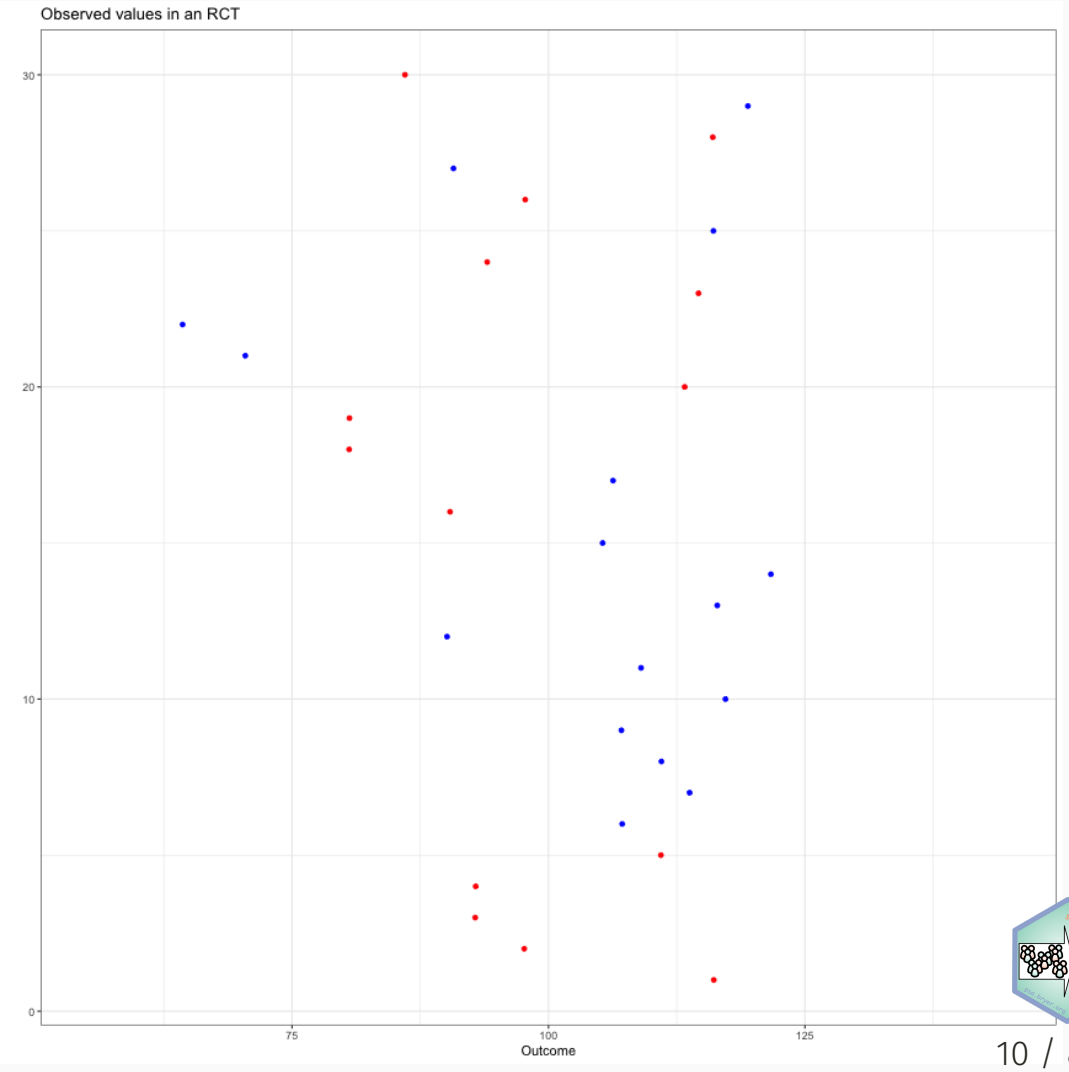


True Counterfactual

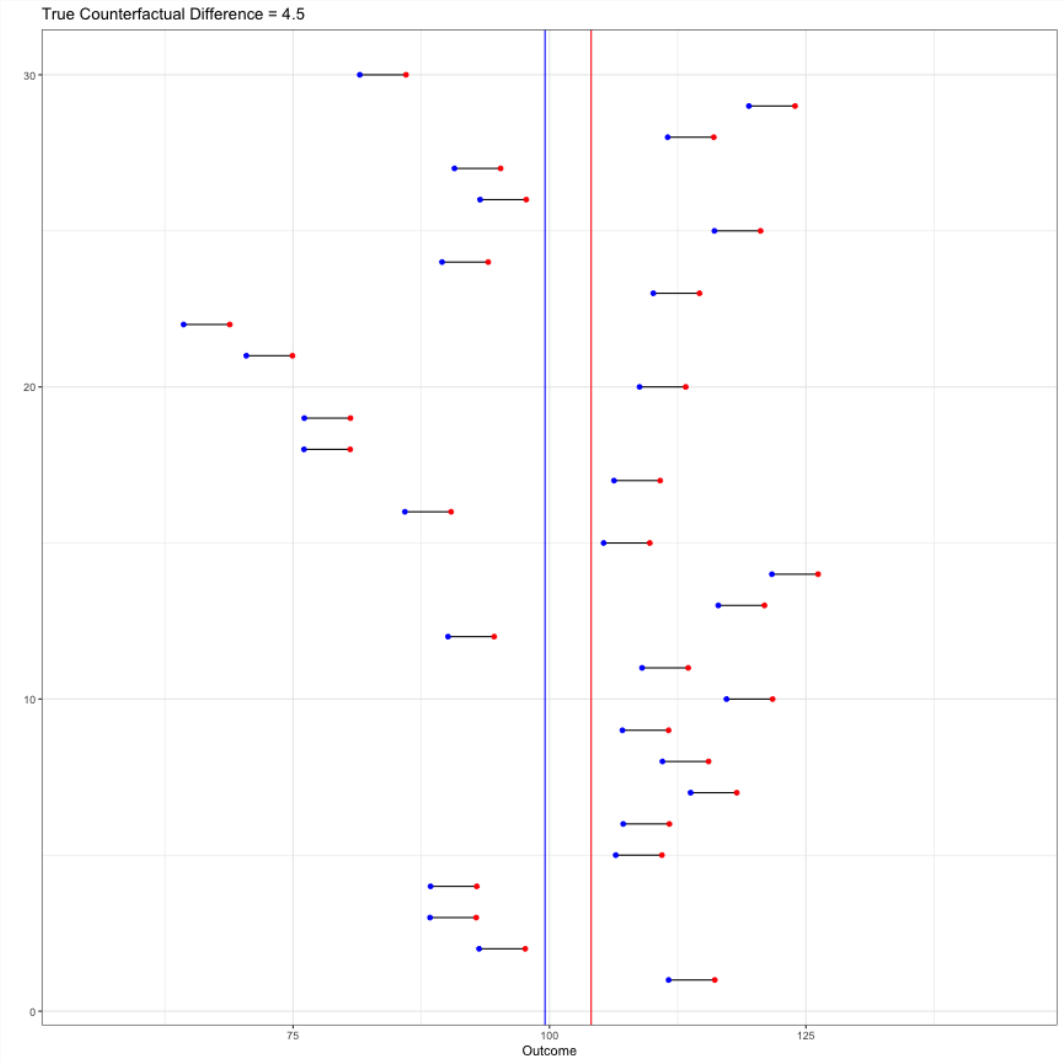


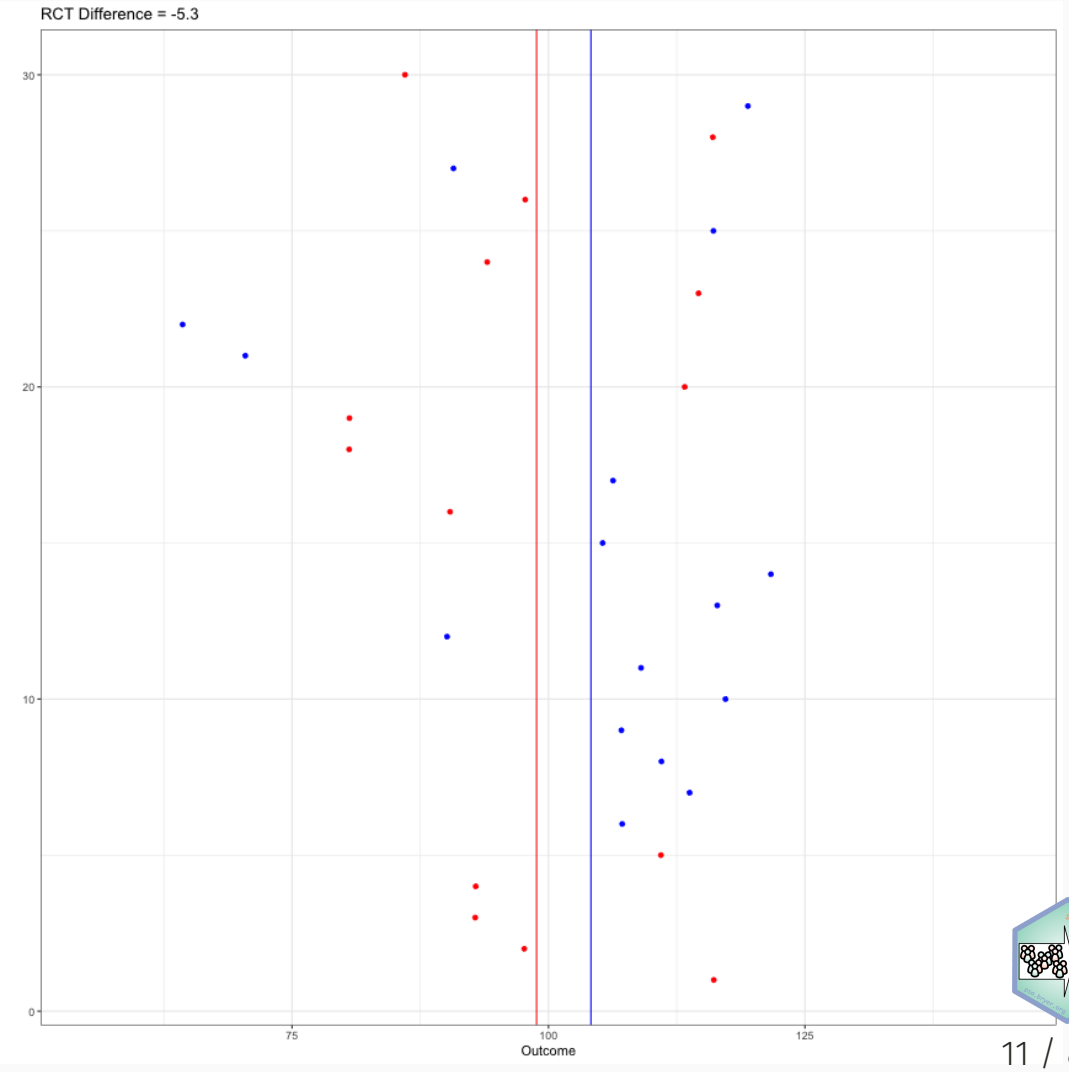
True Counterfactual (left) vs. One RCT (right)



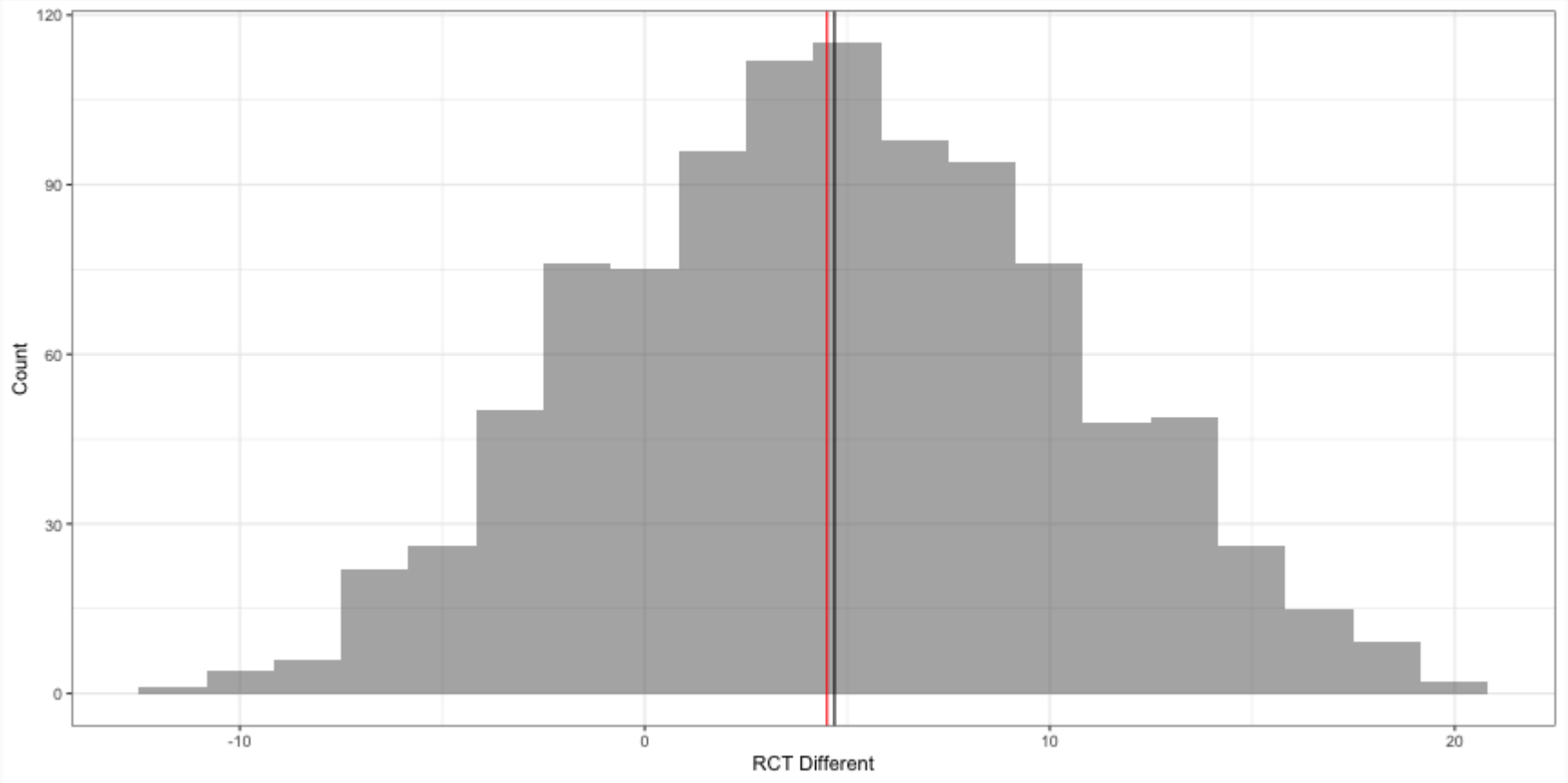


True Counterfactual (left) vs. One RCT (right)





Distribution of Differences from 1,000 RCTs



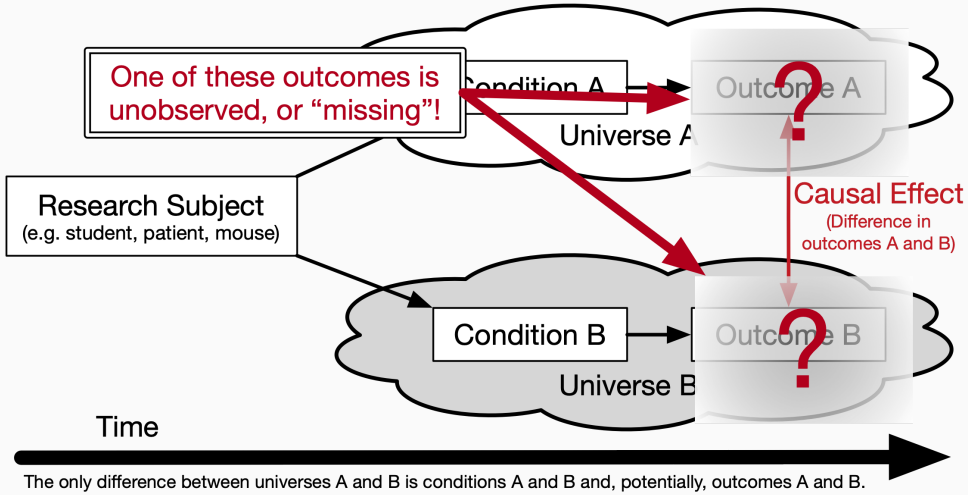
Rubin's Causal Model

- The causal effect of a treatment is the difference in an individual's outcome under the situation they were given the treatment and not (referred to as a counterfactual).

$$\delta_i = Y_{i1} - Y_{i0}$$

- However, it is impossible to directly observe δ_i (referred to as *The Fundamental Problem of Causal Inference*, Holland 1986).

- Rubin frames this problem as a "missing data problem" (see Rubin, 1974, 1977, 1978, 1980, and Holland, 1986).



Propensity Score Analysis

The propensity score is the "conditional probability of assignment to a particular treatment given a vector of observed covariates" (Rosenbaum & Rubin, 1983, p. 41). The probability of being in the treatment:

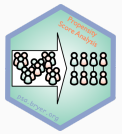
$$\pi(X_i) \equiv Pr(T_i = 1|X_i)$$

The balancing property under exogeneity:

$$T_i \perp\!\!\!\perp X_i \mid \pi(X_i)$$

We can then restate the **ignorability assumption** with the propensity score:

$$(Y_i(1), Y_i(0)) \perp\!\!\!\perp T_i \mid \pi(X_i)$$



Treatment Effects

The average treatment effect (ATE) is defined as:

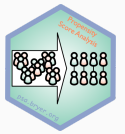
$$E(r_1) - E(r_0)$$

where $E(.)$ is the expectation in the population. For a set of covariates, X, and outcomes Y where 0 denotes control and 1 treatment, we define ATE as:

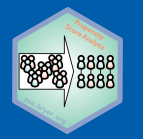
$$ATE = E(Y_1 - Y_0|X) = E(Y_1|X) - E(Y_0|X)$$

As we will see later there are alternative treatment effects (estimands) we can estimate instead of ATE.

What Rosenbaum and Rubin (1983) proved in their seminal paper is that the propensity score is a univariate representation of the multivariate matrix. As we will see later, two observations with very similar propensity scores will look similar across all the observed covariates.



Propensity Score Analysis in Three Phases



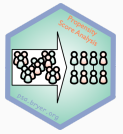
Simulated Example

We will simulate a dataset with three covariates, `x1` and `x2` which are continuous and `x3` which is categorical. The assumed treatment effect is 1.5.

```
n <- 500
treatment_effect <- 1.5
X <- mvtnorm::rmvnorm(
    n,
    mean = c(0.5, 1, 0),
    sigma = matrix(c(2, 1, 1,
                     1, 1, 1,
                     1, 1, 1),
                     ncol = 3) )
dat <- tibble(
    x1 = X[, 1],
    x2 = X[, 2],
    x3 = X[, 3] > 0,
    treatment = as.numeric(- 0.5 +
                               0.25 * x1 +
                               0.75 * x2 +
                               0.05 * x3 +
                               rnorm(n, 0, 1) > 0),
    outcome = treatment_effect * treatment +
        rnorm(n, 0, 1)
)
```

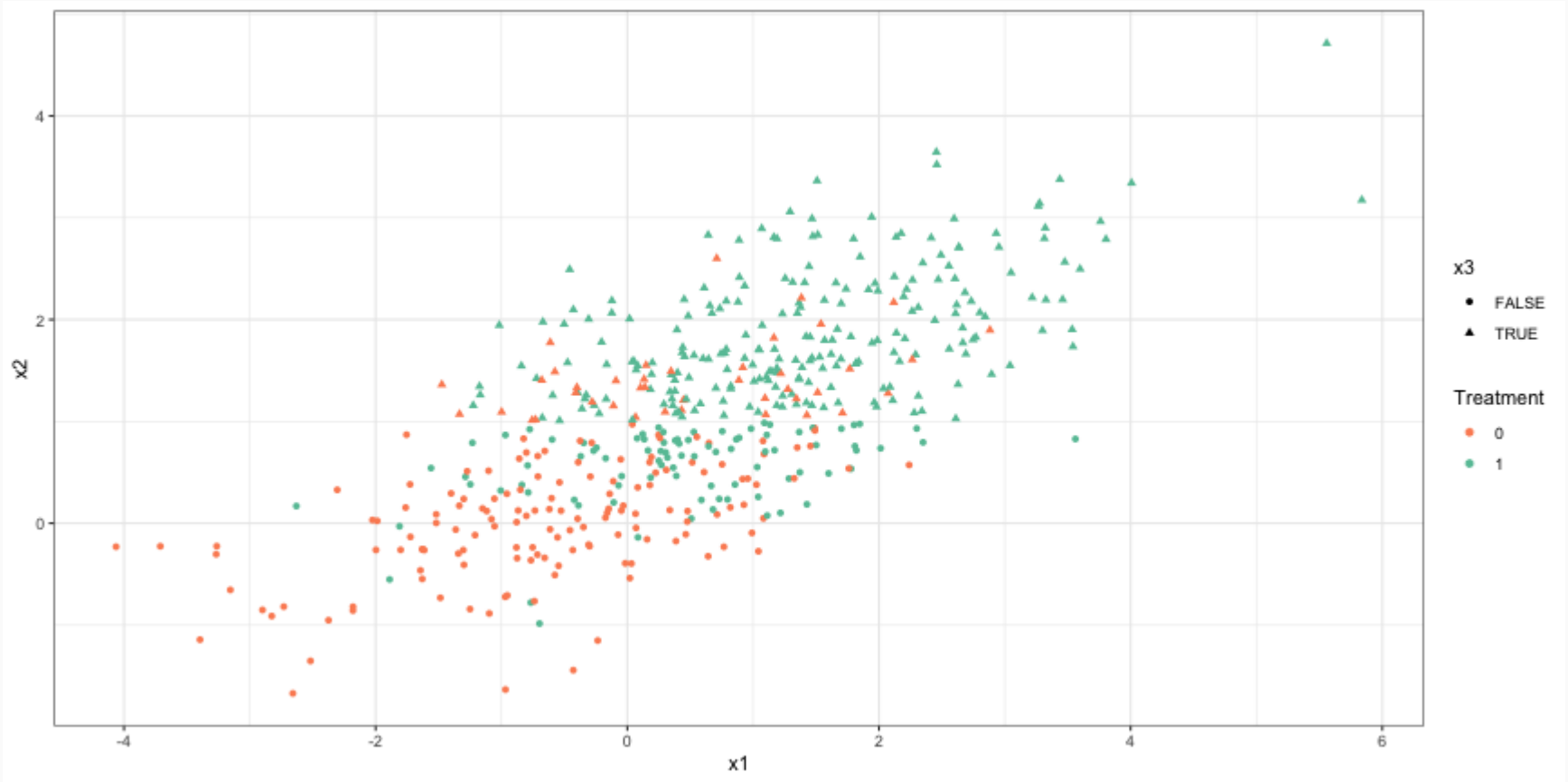
```
head(dat, n = 6)
```

```
## # A tibble: 6 × 5
##       x1    x2 x3    treatment outcome
##    <dbl> <dbl> <lgl>     <dbl>   <dbl>
## 1  1.35  0.744 FALSE         0  1.46
## 2  0.149 1.55  TRUE          0 -0.924
## 3  2.47  2.39  TRUE          1 -0.0527
## 4  2.29  1.66  TRUE          1  1.05
## 5  2.93  2.85  TRUE          1  0.721
## 6 -0.867 0.125 FALSE         0  0.723
```

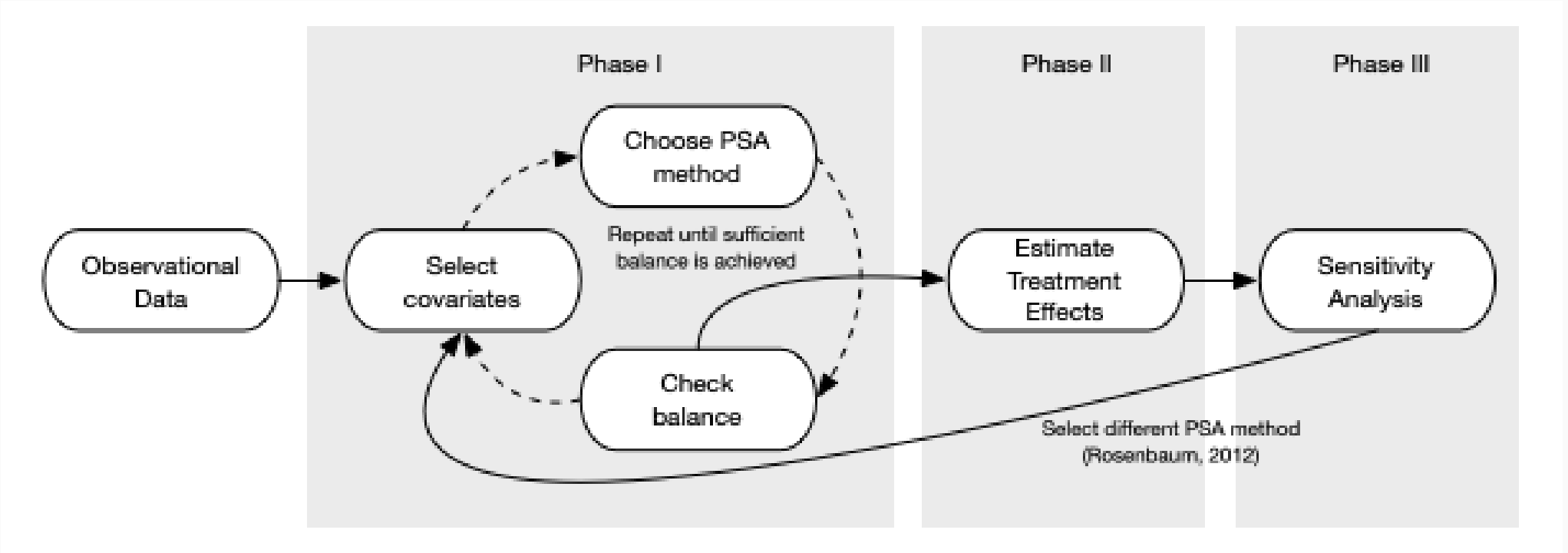


Scatterplot

```
ggplot(dat, aes(x = x1, y = x2, shape = x3, color = factor(treatment))) +
    geom_point() + scale_color_manual('Treatment', values = cols)
```



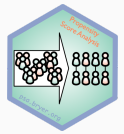
Steps for Implementing Propensity Score Analysis



Propensity score methods

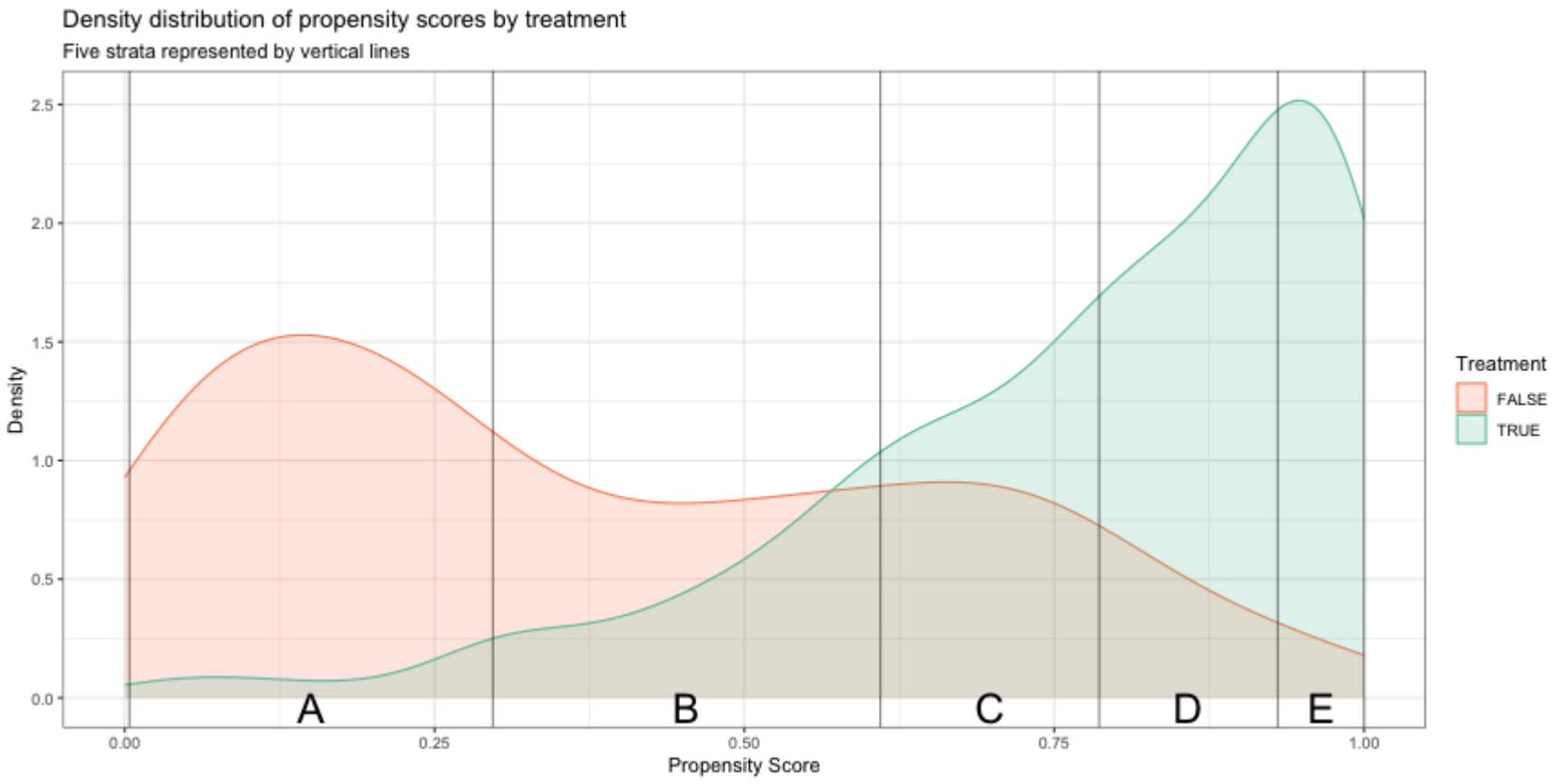
There are three major approaches for conducting PSA:

- **Stratification** Treatment and comparison units are divided into strata (or subclasses) so that treated and comparison units are similar within each strata. Cochran (1968) observed that creating five subclassifications (stratum) removes at least 90% of the bias in the estimated treatment effect.
- **Matching** - Each treatment unit is paired with a comparison unit based upon the pre-treatment covariates.
- **Weighting** Each observation is weighted by the inverse of the probability of being in that group.



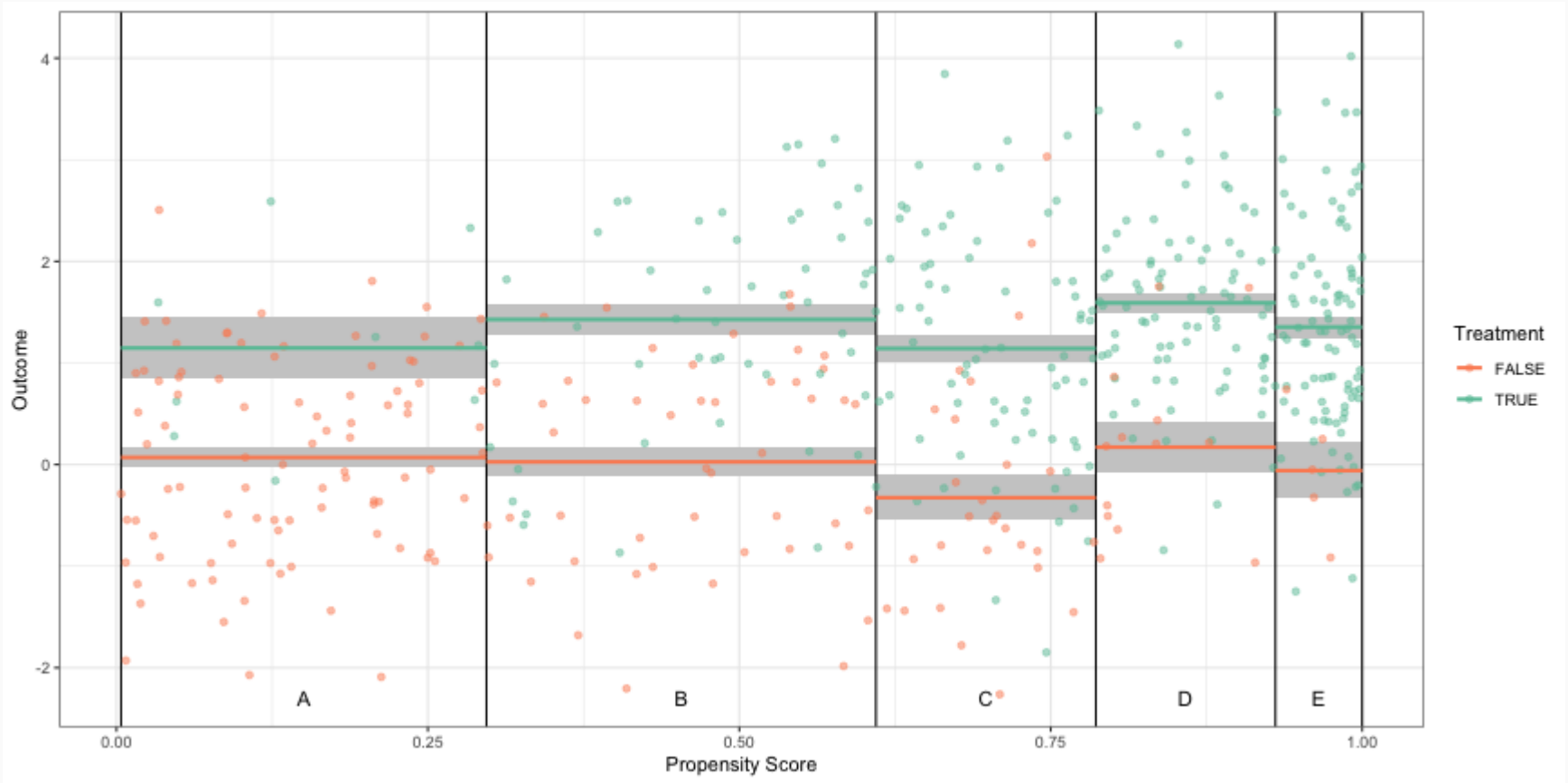
Stratification

Stratification involves dividing (or stratifying) the observations into subgroups based upon the propensity score. Here, we used quintiles on the propensity scores where were estimated using logistic regression. For classification trees the stratum is determined by the leaf nodes.



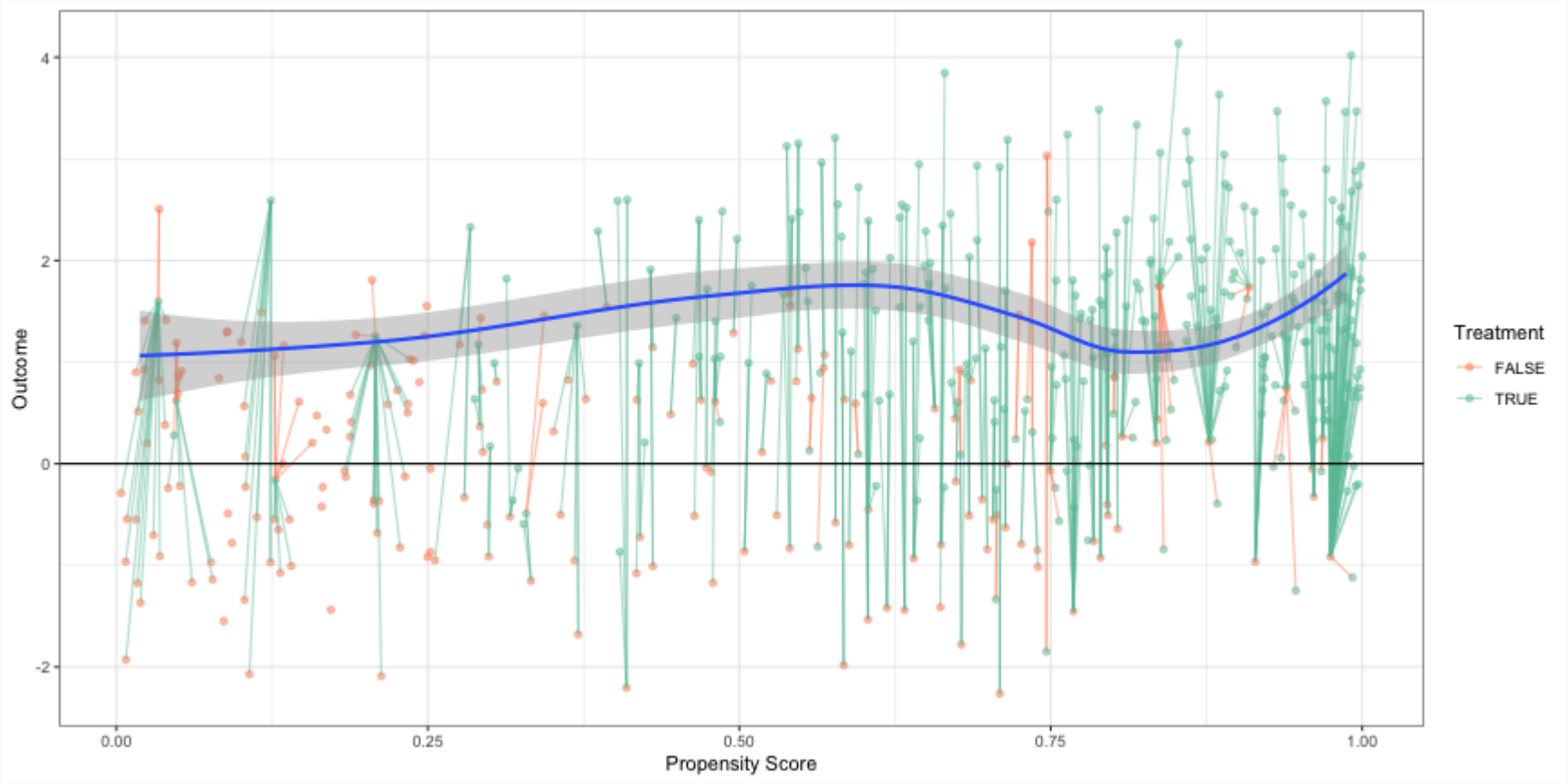
Stratification (cont.)

Independent sample tests (e.g. *t*-tests) are conducted within each stratum and pooled to provide an overall estimate.



Matching

Dependent sample tests (e.g. *t*-tests) are conducted using match pairs to provide a treatment.



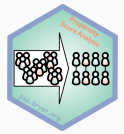
Matching Methods

There are many choices and approaches to matching, including:

- Propensity score matching.
- Limited exact matching.
- Full matching.
- Nearest neighbor matching.
- Optimal/Genetic matching.
- Mahalanobis distance matching (for quantitative covariates only).
- Matching with and without replacement.
- One-to-one or one-to-many matching.

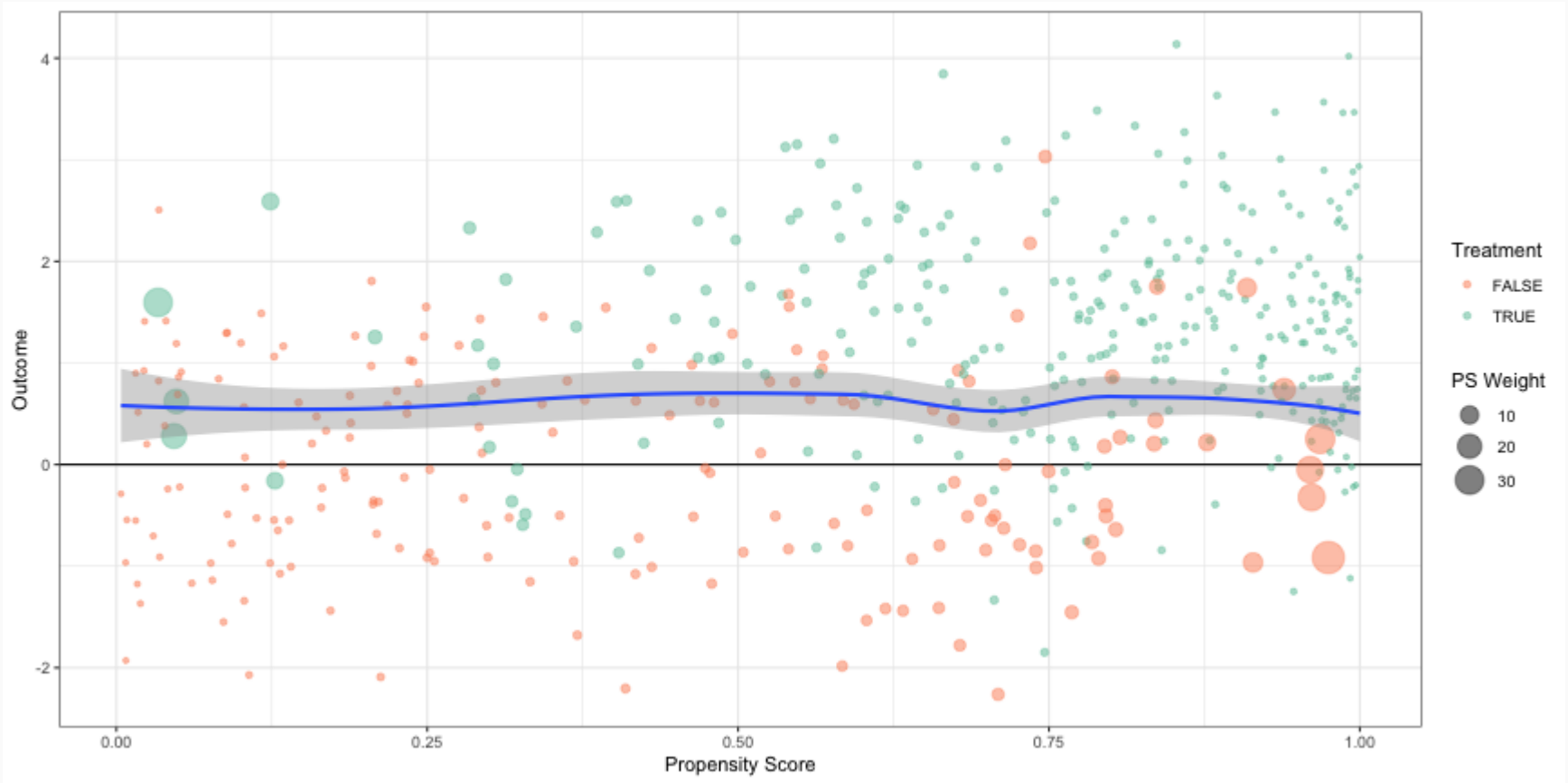
Which method should you use?

Whichever one gives the best balance!



Weighting

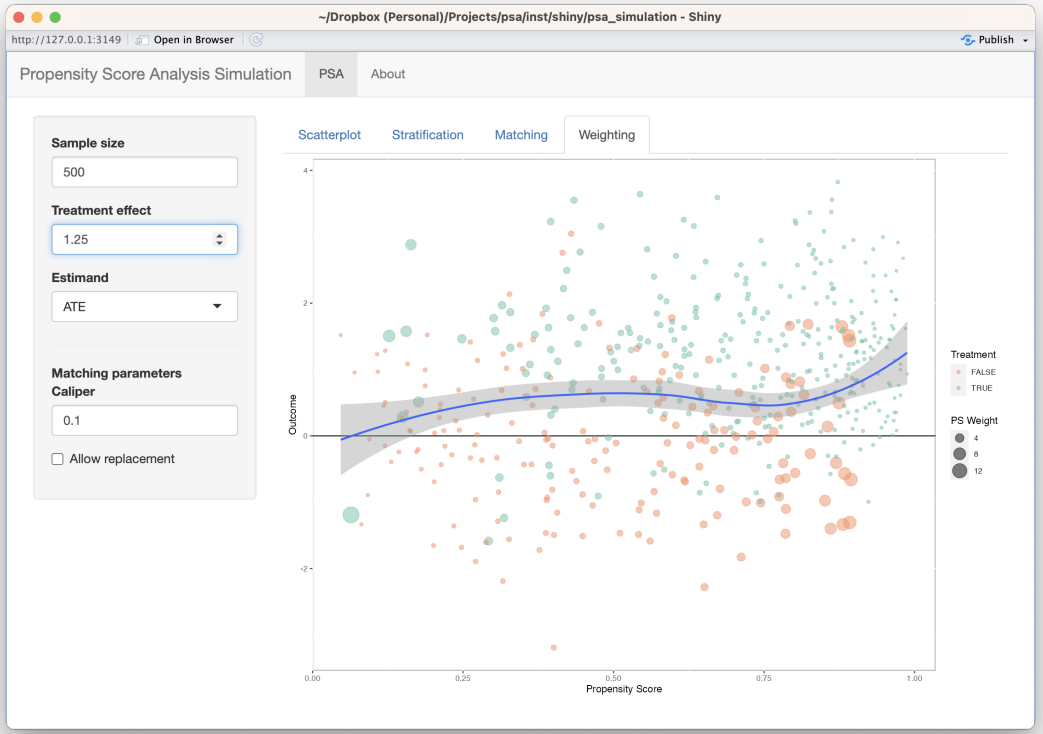
Propensity score weights can be used as regression weights, the specific weights depend on the desired estimand and will be provided in later slides.



Shiny Application

We can explore how these three plots change as the treatment effects change using the `psa::psa_simulation_shiny()` application.

```
psa::psa_simulation_shiny()
```



Phase I: Estimate Propensity Scores

In this example we will use logistic regression to estimate the propensity scores.

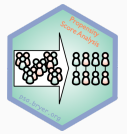
```
lr.out <- glm(
    treatment ~ x1 + x2 + x3,
    data = dat,
    family = binomial(link='logit'))
dat$ps <- fitted(lr.out) # Propensity scores
```

For stratification we will use quintiles to split the observations into five equal groups.

```
breaks5 <- psa::get_strata_breaks(dat$ps)
dat$strata5 <- cut(
    x = dat$ps,
    breaks = breaks5$breaks,
    include.lowest = TRUE,
    labels = breaks5$labels$strata)
```

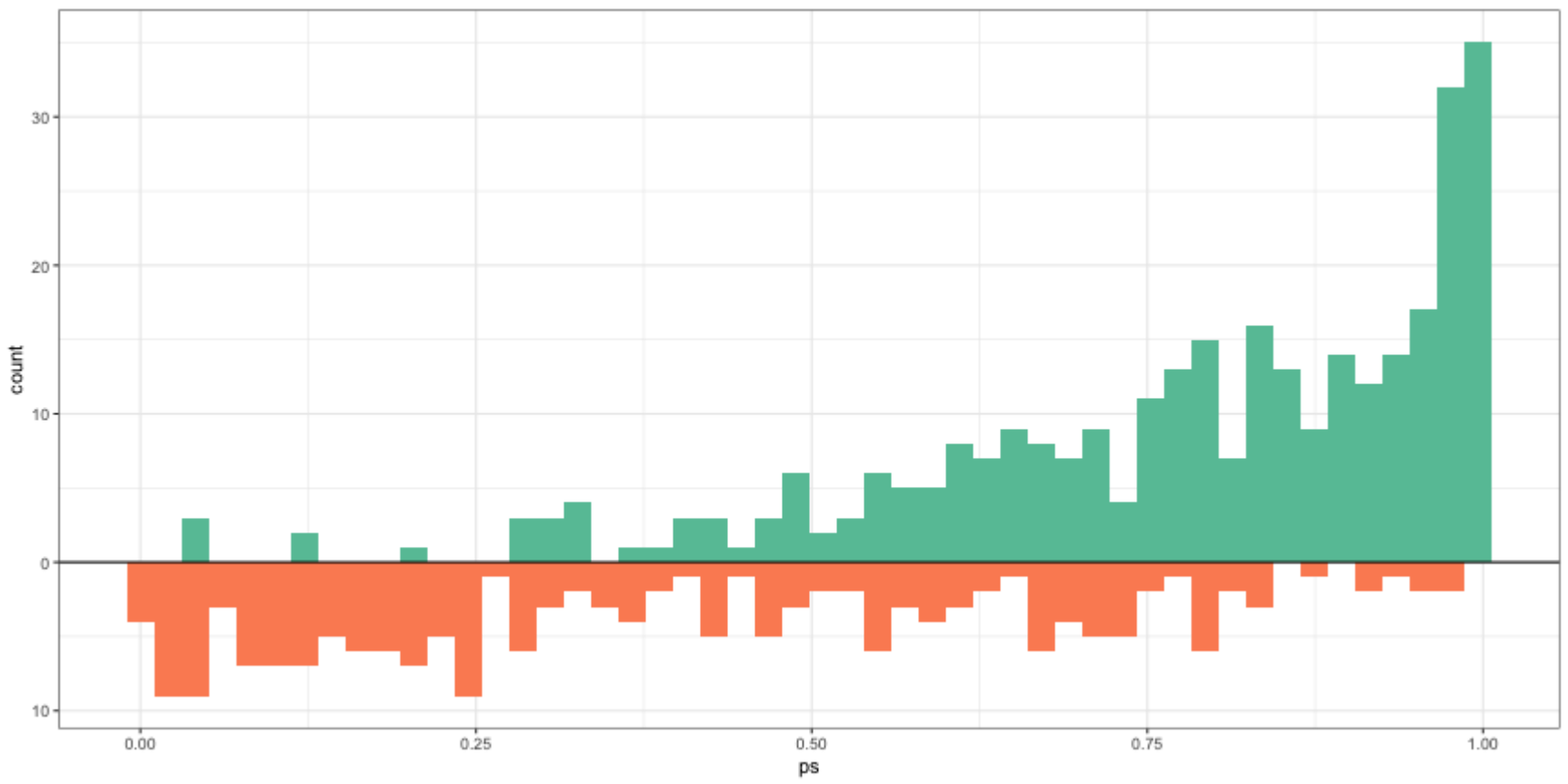
```
summary(lr.out)
```

```
##
## Call:
## glm(formula = treatment ~ x1 + x2 + x3, family = binomial(link = "logit"),
##     data = dat)
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.1006     0.2069  -5.319 1.04e-07 ***
## x1            0.4399     0.1266   3.476  0.00051 ***
## x2            1.9818     0.3404   5.823 5.79e-09 ***
## x3TRUE       -0.7166     0.4087  -1.753  0.07955 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 658.96  on 499  degrees of freedom
## Residual deviance: 432.95  on 496  degrees of freedom
## AIC: 440.95
##
## Number of Fisher Scoring iterations: 5
```



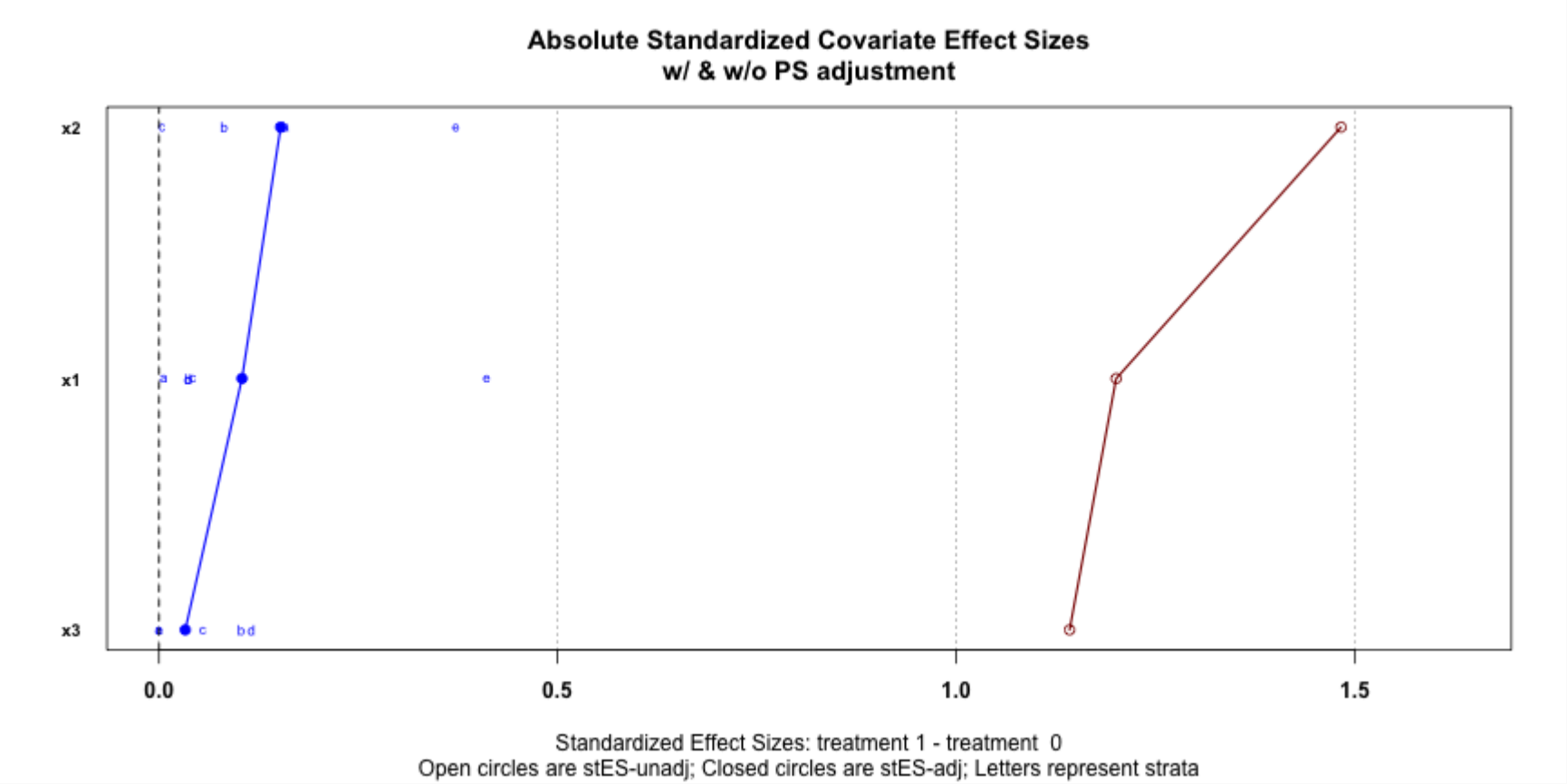
Distribution of Propensity Scores

```
ggplot(dat) +
    geom_histogram(data = dat[dat$treatment == 1,], aes(x = ps, y = after_stat(count)), bins = 50, fill = cols[2]) +
    geom_histogram(data = dat[dat$treatment == 0,], aes(x = ps, y = -after_stat(count)), bins = 50, fill = cols[1]) +
    geom_hline(yintercept = 0, lwd = 0.5) +    scale_y_continuous(label = abs)
```



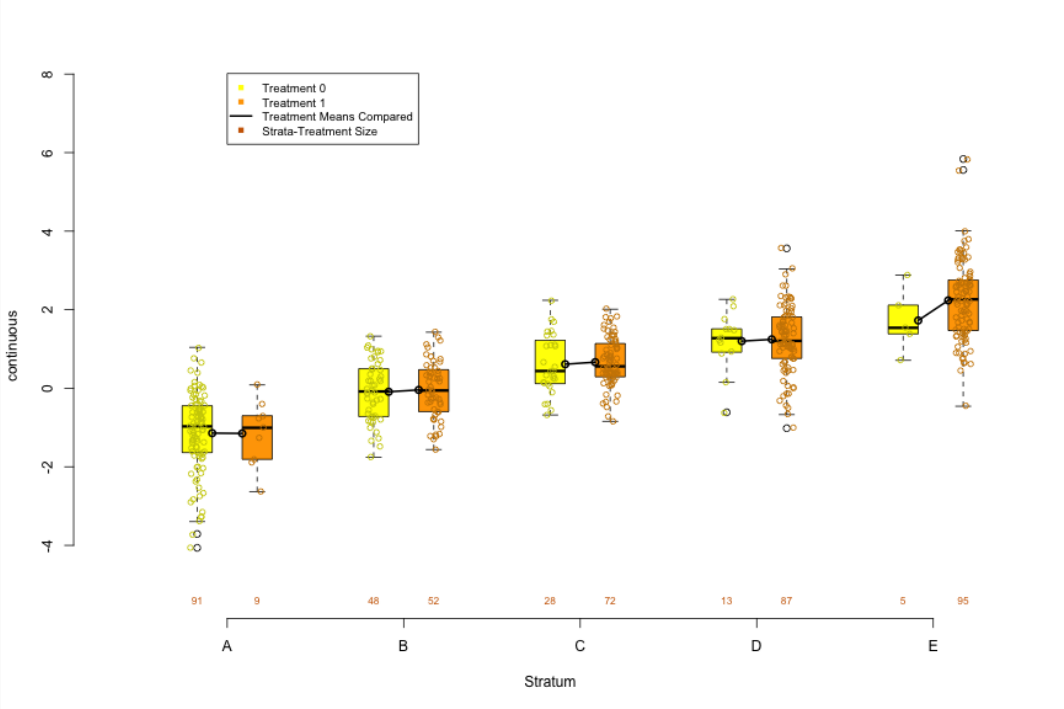
Check Balance: Multiple Covariates

```
PSAgraphics::cv.bal.psa(dat[,1:3], dat$treatment, dat$ps, strata = 5)
```

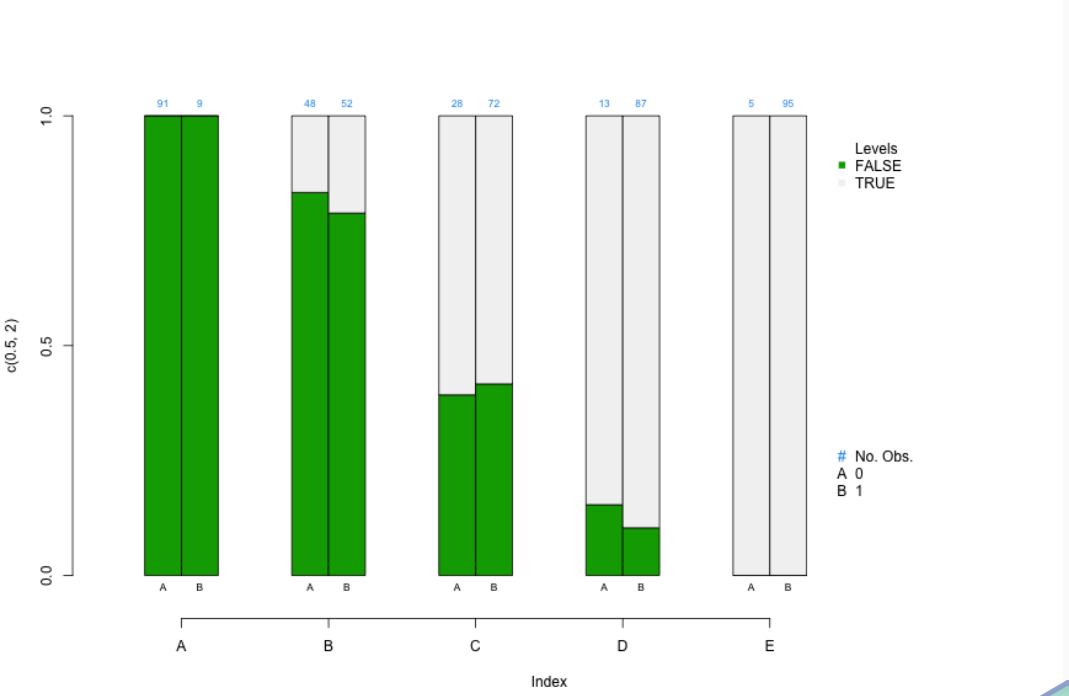


Check Balance: Single Covariate

```
PSAgraphics::box.psa(dat$x1,
                     dat$treatment,
                     dat$strata5)
```




```
PSAgraphics::cat.psa(dat$x3,
                     dat$treatment,
                     dat$strata5)
```

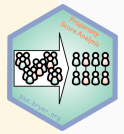


PS Weights for Understanding Treatment Effects

Given that the distribution of treatment and control observations across the propensity score range are not the same, there are a number of alternative estimates of treatment effect. We will explore three additional esimates in addition to the classic average treatment effect.

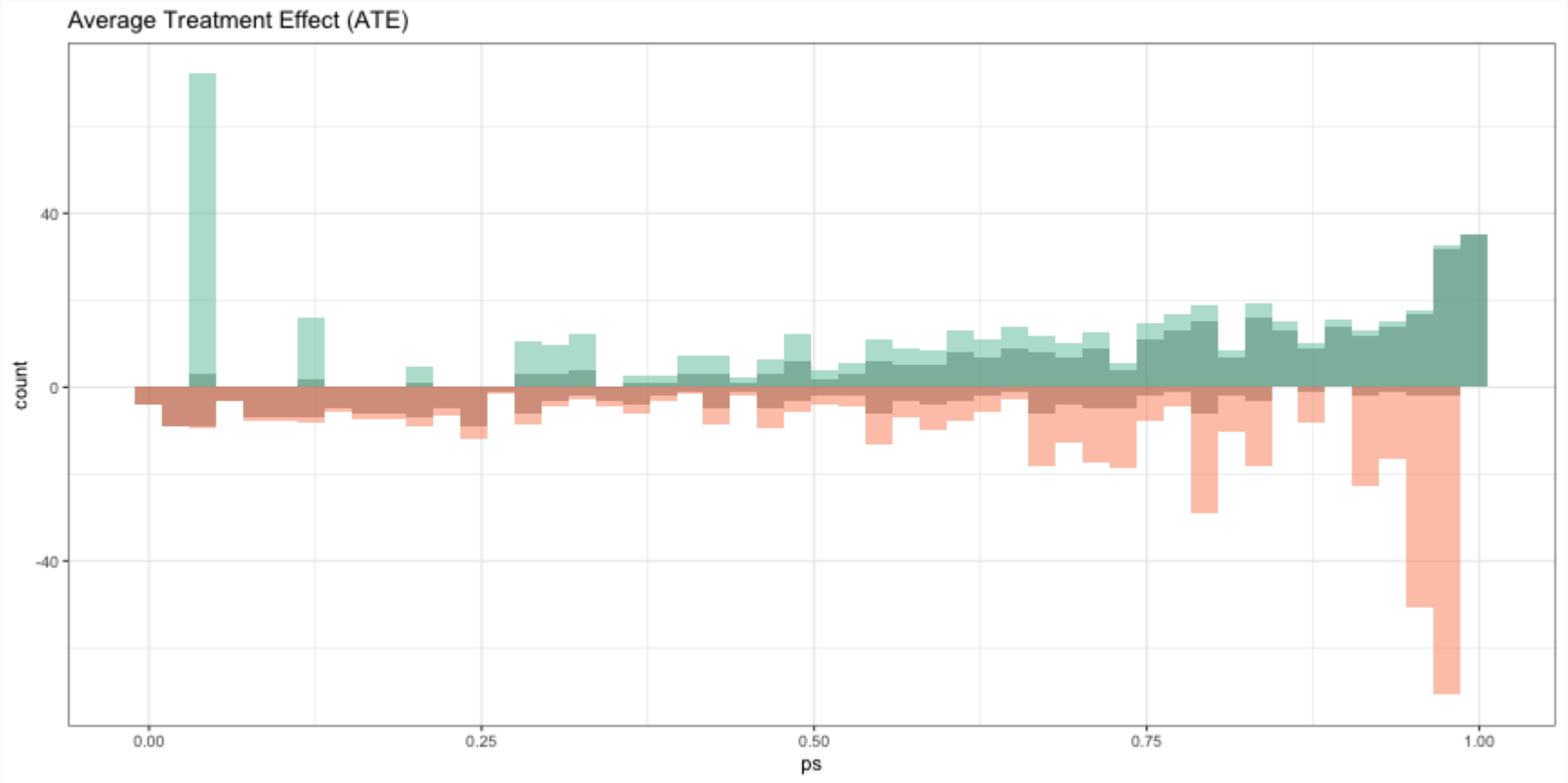
```
dat <- dat |> mutate(
    ate_weight = psa::calculate_ps_weights(treatment, ps, estimand = 'ATE'),
    att_weight = psa::calculate_ps_weights(treatment, ps, estimand = 'ATT'),
    atc_weight = psa::calculate_ps_weights(treatment, ps, estimand = 'ATC'),
    atm_weight = psa::calculate_ps_weights(treatment, ps, estimand = 'ATM')
)
dat |> head(n = 4)
```

```
## # A tibble: 4 × 11
##       x1    x2 x3    treatment outcome    ps strata5 ate_weight att_weight
##    <dbl> <dbl> <lgl>     <dbl>   <dbl> <dbl> <fct>        <dbl>      <dbl>
## 1 1.35  0.744 FALSE         0  1.46   0.725 C             3.63       2.63
## 2 0.149 1.55  TRUE          0 -0.924  0.790 D             4.76       3.76
## 3 2.47  2.39  TRUE          1 -0.0527 0.982 E             1.02       1
## 4 2.29  1.66  TRUE          1  1.05   0.922 D             1.08       1
## # ℹ 2 more variables: atc_weight <dbl>, atm_weight <dbl>
```



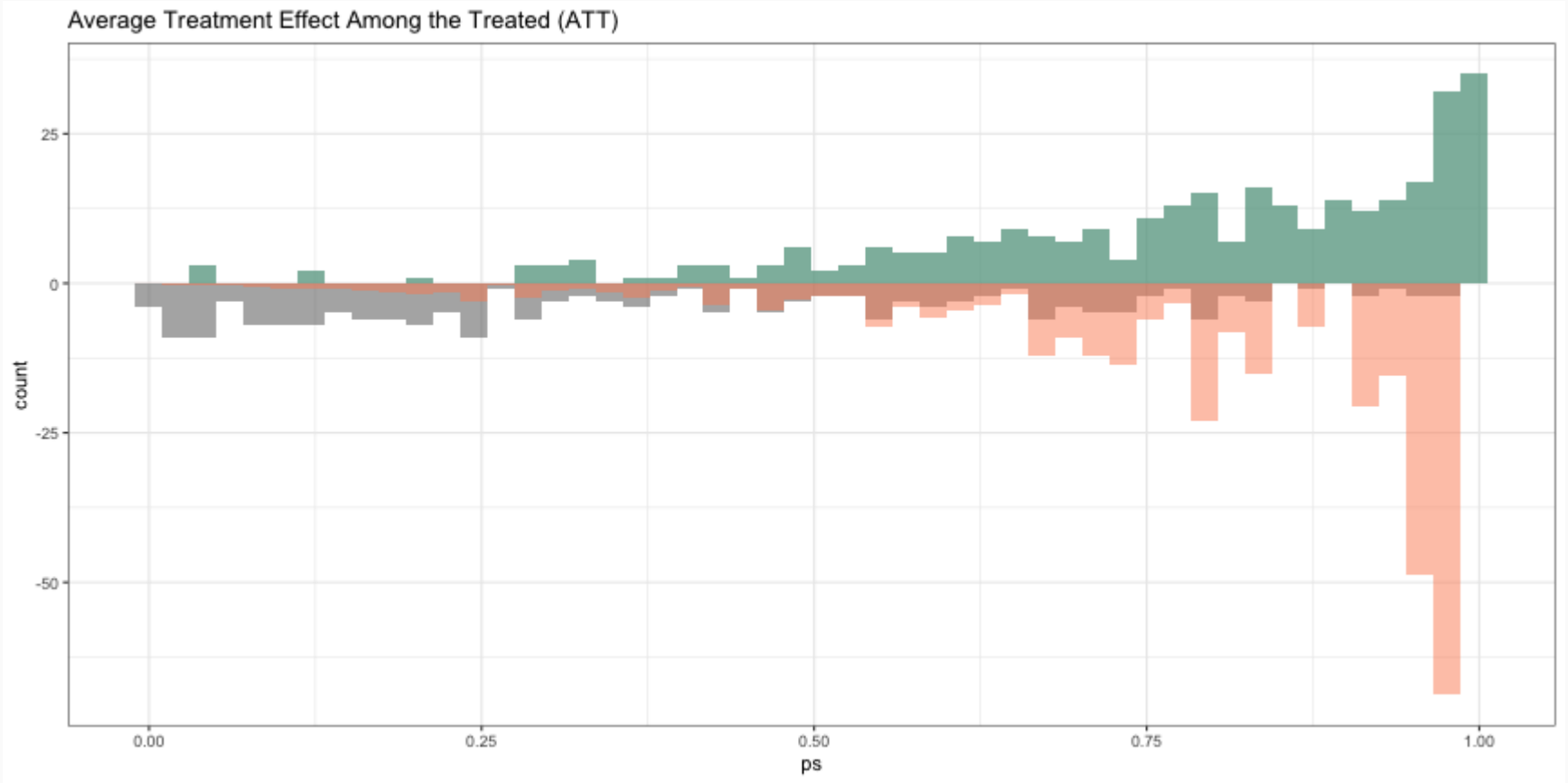
Average Treatment Effect (ATE)

$$ATE = E(Y_1 - Y_0|X) = E(Y_1|X) - E(Y_0|X)$$



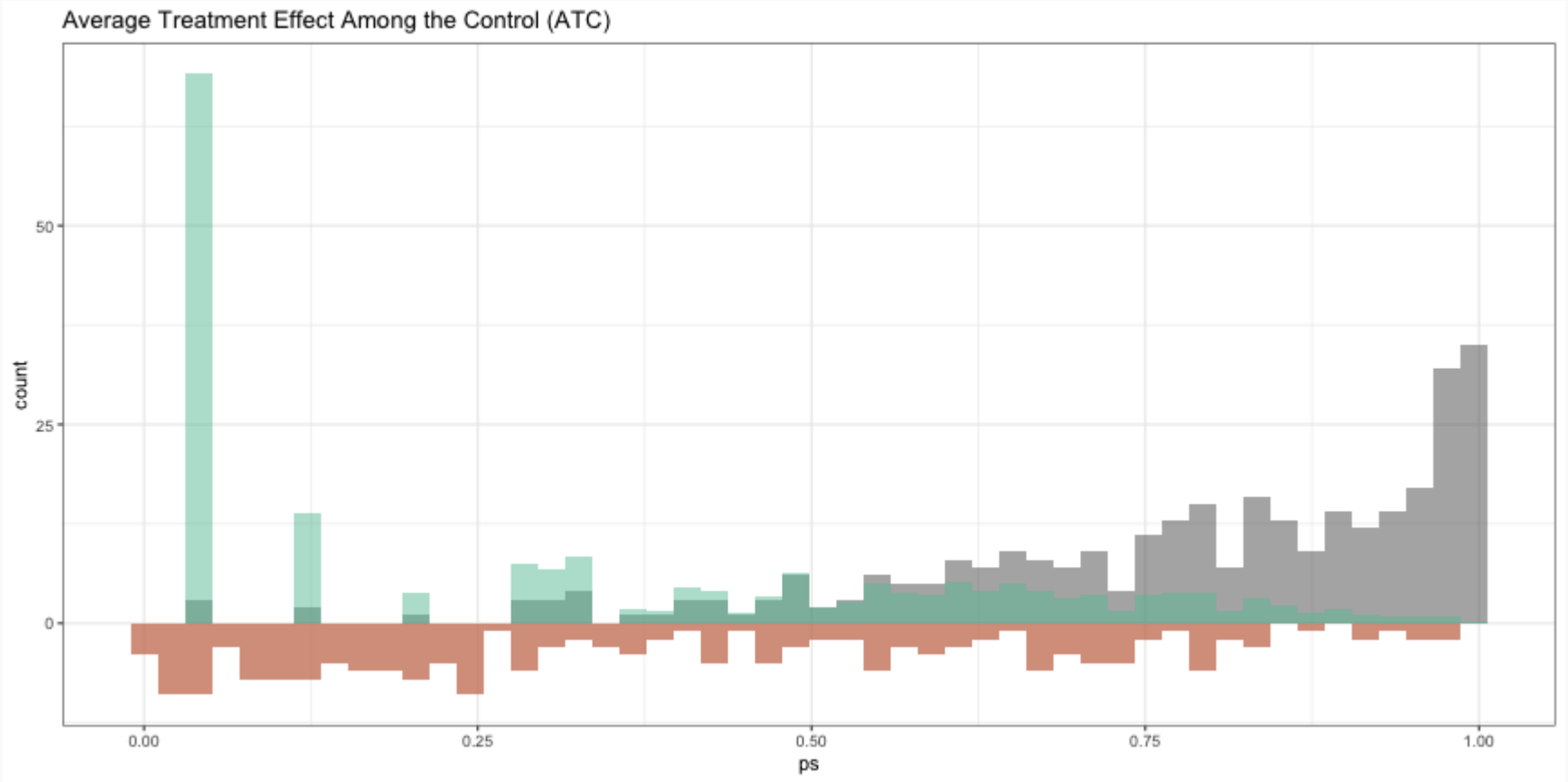
Average Treatment Effect Among the Treated (ATT)

$$ATT = E(Y_1 - Y_0|X, C = 1) = E(Y_1|X, C = 1) - E(Y_0|X, C = 1)$$



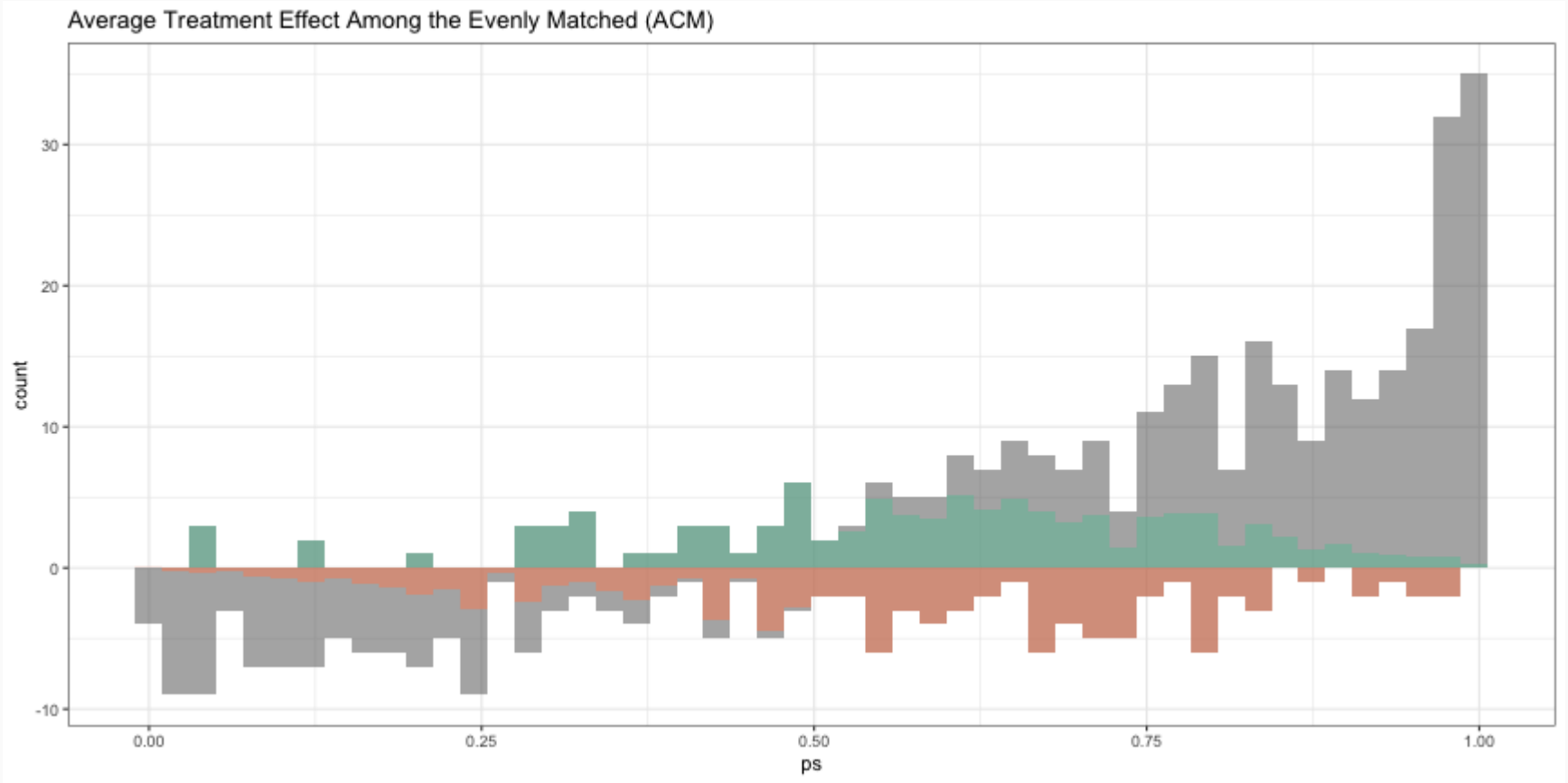
Average Treatment Effect Among the Control (ATC)

$$ATC = E(Y_1 - Y_0|X = 0) = E(Y_1|X = 0) - E(Y_0|X = 0)$$



Average Treatment Effect Among the Evenly Matched

$$ATM_d = E(Y_1 - Y_0 | M_d = 1)$$



Treatment Effects for Weighting

$$Treatment\ Effect = \frac{\sum Y_i Z_i w_i}{\sum Z_i w_i} - \frac{\sum Y_i (1 - Z_i) w_i}{\sum (1 - Z_i) w_i}$$

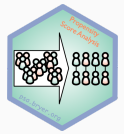
Where w is the weight (as defined in the following sections), Z_i is the treatment assignment such that $Z = 1$ is treatment and $Z = 0$ is control, and Y_i is the outcome

$$w_{ATE} = \frac{Z_i}{\pi_i} + \frac{1 - Z_i}{1 - \pi_i}$$

$$w_{ATC} = \frac{(1 - \pi_i) Z_i}{\pi_i} + \frac{(1 - e_i)(1 - Z_i)}{1 - \pi_i}$$

$$w_{ATT} = \frac{\pi_i Z_i}{\pi_i} + \frac{\pi_i (1 - Z_i)}{1 - \pi_i}$$

$$w_{ATM} = \frac{min\{\pi_i, 1 - \pi_i\}}{Z_i \pi_i (1 - Z_i)(1 - \pi_i)}$$



Treatment Effects

Average Treatment Effect

```
psa::treatment_effect(
    treatment = dat$treatment,
    outcome = dat$outcome,
    weights = dat$ate_weight)
```

```
## [1] 1.336979
```

```
lm(outcome ~ treatment,
   data = dat,
   weights = dat$ate_weight)
```

```
##
## Call:
## lm(formula = outcome ~ treatment, data = dat, weights =
##
## Coefficients:
## (Intercept)    treatment
##      -0.044        1.337
```

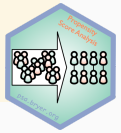
Average Treatment Effect Among the Treated

```
psa::treatment_effect(
    treatment = dat$treatment,
    outcome = dat$outcome,
    weights = dat$att_weight)
```

```
## [1] 1.447406
```

```
lm(outcome ~ treatment,
   data = dat,
   weights = dat$att_weight)
```

```
##
## Call:
## lm(formula = outcome ~ treatment, data = dat, weights =
##
## Coefficients:
## (Intercept)    treatment
##    -0.07002      1.44741
```



Treatment Effects (cont.)

Average Treatment Effect Among the Control

```
psa::treatment_effect(
    treatment = dat$treatment,
    outcome = dat$outcome,
    weights = dat$atc_weight)
```

```
## [1] 1.157861
```

```
lm(outcome ~ treatment,
   data = dat,
   weights = dat$atc_weight)
```

```
##
## Call:
## lm(formula = outcome ~ treatment, data = dat, weights =
##
## Coefficients:
## (Intercept)    treatment
##    0.002491     1.157861
```

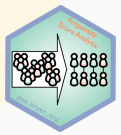
Average Treatment Effect Among the Evenly Matched

```
psa::treatment_effect(
    treatment = dat$treatment,
    outcome = dat$outcome,
    weights = dat$atm_weight)
```

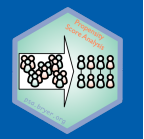
```
## [1] 1.370067
```

```
lm(outcome ~ treatment,
   data = dat,
   weights = dat$atm_weight)
```

```
##
## Call:
## lm(formula = outcome ~ treatment, data = dat, weights =
##
## Coefficients:
## (Intercept)    treatment
##    -0.02388      1.37007
```



Example: National Supported Work Demonstration

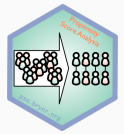


National Supported Work

The National Supported Work (NSW) Demonstration was a federally and privately funded randomized experiment done in the 1970s to estimate the effects of a job training program for disadvantaged workers.

- Participants were randomly selected to participate in the training program.
- Both groups were followed up to determine the effect of the training on wages.
- Analysis of the mean differences (unbiased given randomization), was approximately $800.

Lalonde (1986) used data from the Panel Survey of Income Dynamics (PSID) and the Current Population Survey (CPS) to investigate whether non-experimental methods would result in similar results to the randomized experiment. He found results ranging from $700 to $16,000.



National Supported Work (cont.)

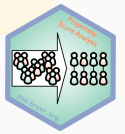
Dehejia and Wahba (1999) later used propensity score matching to analyze the data. The found that,

- Comparison groups selected by Lalonde were very dissimilar to the treated group.
- By restricting the comparison group to those that were similar to the treated group, they could replicate the original NSW results.
- Using the CPS data, the range of treatment effect was between $1,559 to $1,681. The experimental results for the sample sample was approximately $1,800.

The covariates available include: age, education level, high school degree, marital status, race, ethnicity, and earning sin 1974 and 1975.

Outcome of interest is earnings in 1978.

```
data(lalonde, package='Matching')
```



Estimating Propensity Scores

Estimate propensity scores using logistic regression.

```
lalonde.formu <- treat ~ age + educ + black + hisp +
    married + nodegr + re74 + re75
glm1 <- glm(lalonde.formu,
            data = lalonde,
            family = binomial(link = 'logit'))
```

Get the propensity scores:

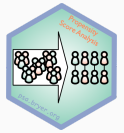
```
lalonde$ps <- fitted(glm1)
```

Define the stratification:

```
strata5 <- cut(lalonde$ps,
               quantile(lalonde$ps, seq(0, 1, 1/5)),
               include.lowest = TRUE,
               labels = letters[1:5])
```

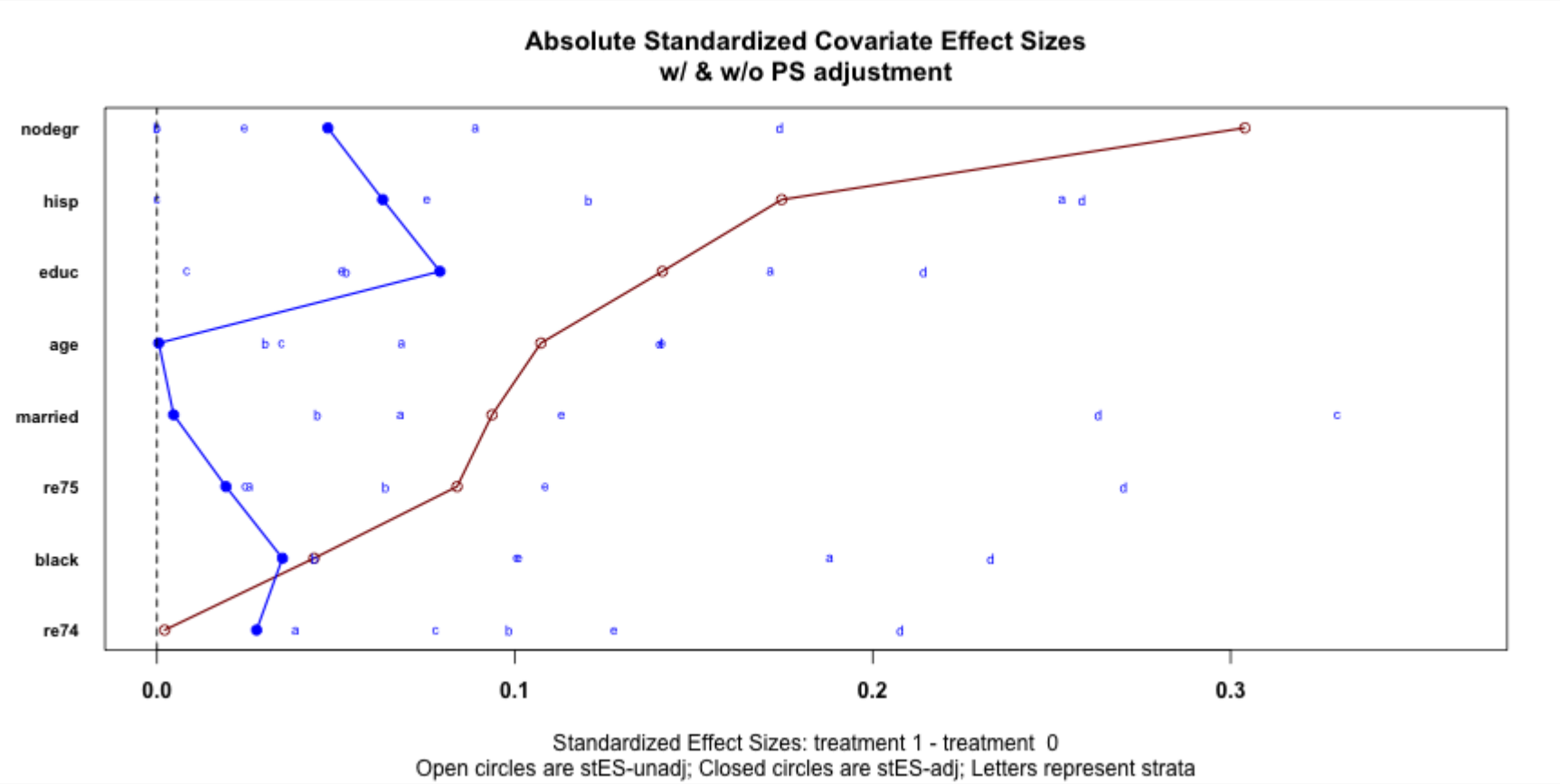
```
summary(glm1)
```

```
##
## Call:
## glm(formula = lalonde.formu, family = binomial(link = "logit"),
##     data = lalonde)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.178e+00  1.056e+00   1.115  0.26474
## age          4.698e-03  1.433e-02   0.328  0.74297
## educ        -7.124e-02  7.173e-02  -0.993  0.32061
## black       -2.247e-01  3.655e-01  -0.615  0.53874
## hisp        -8.528e-01  5.066e-01  -1.683  0.09228 .
## married      1.636e-01  2.769e-01   0.591  0.55463
## nodegr      -9.035e-01  3.135e-01  -2.882  0.00395 **
## re74        -3.161e-05  2.584e-05  -1.223  0.22122
## re75         6.161e-05  4.358e-05   1.414  0.15744
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 604.20  on 444  degrees of freedom
## Residual deviance: 587.22  on 436  degrees of freedom
## AIC: 605.22
##
## Number of Fisher Scoring iterations: 4
```



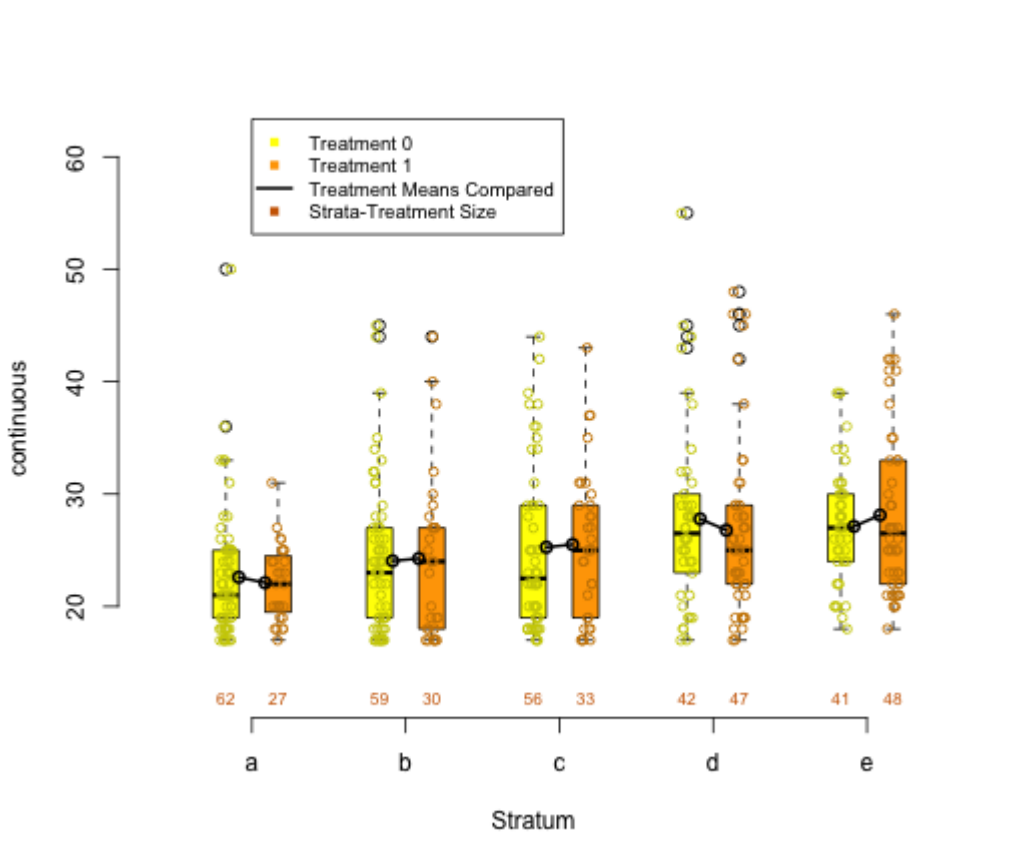
Checking Balance: Covariate Balance Plot

```
covars <- all.vars(lalonde.formu)
covars <- lalonde[,covars[2:length(covars)]]
cv.bal.psa(covars, lalonde$treat, lalonde$ps, strata = 5)
```

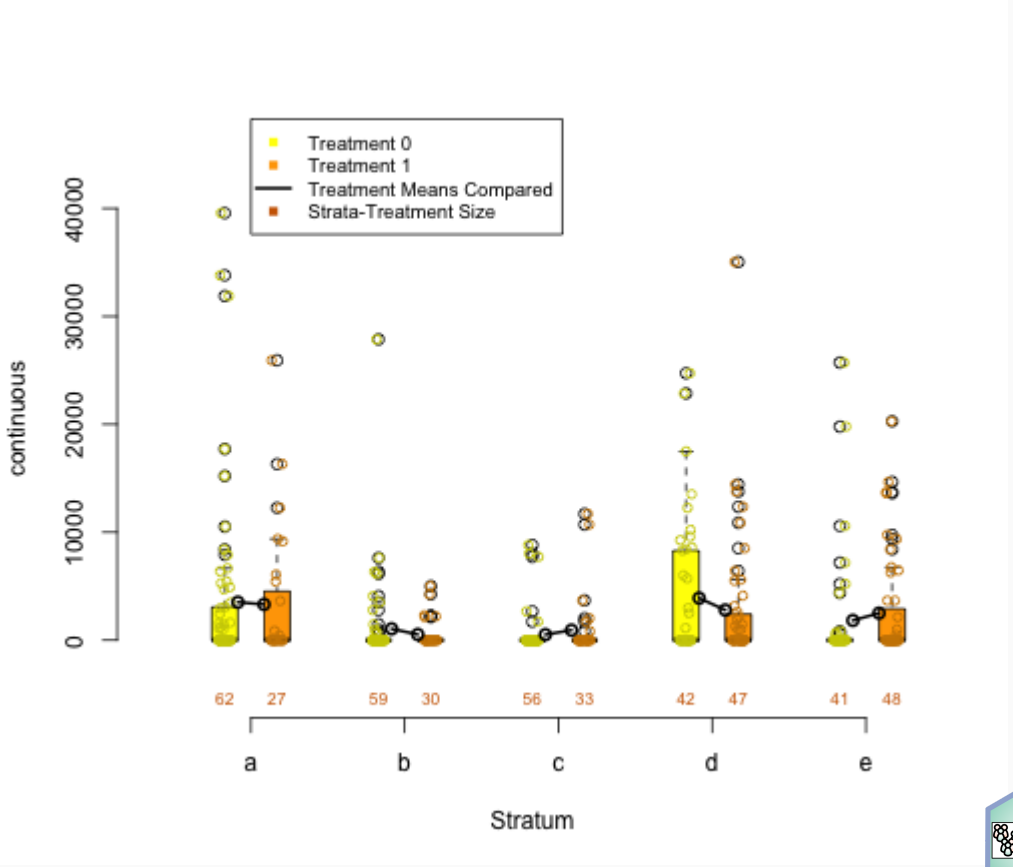


Checking Balance: Continuous Covariates

```
box.psa(lalonde$age, lalonde$treat, strata5)
```

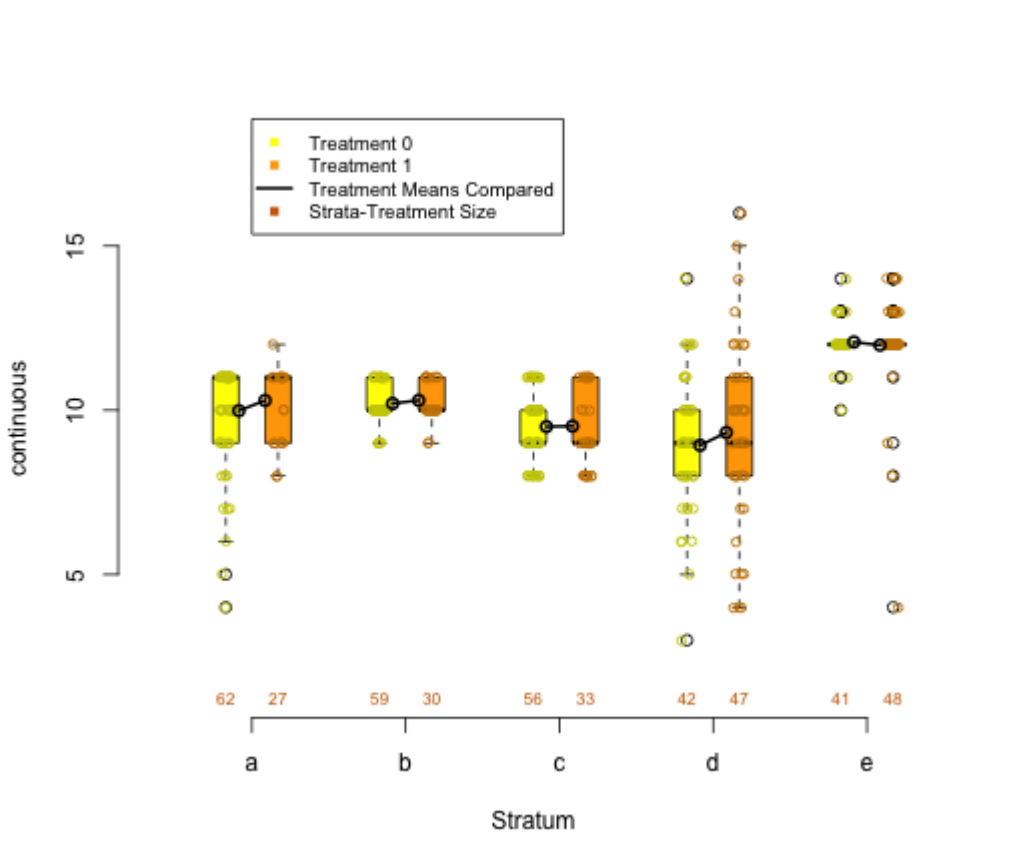



```
box.psa(lalonde$re74, lalonde$treat, strata5)
```

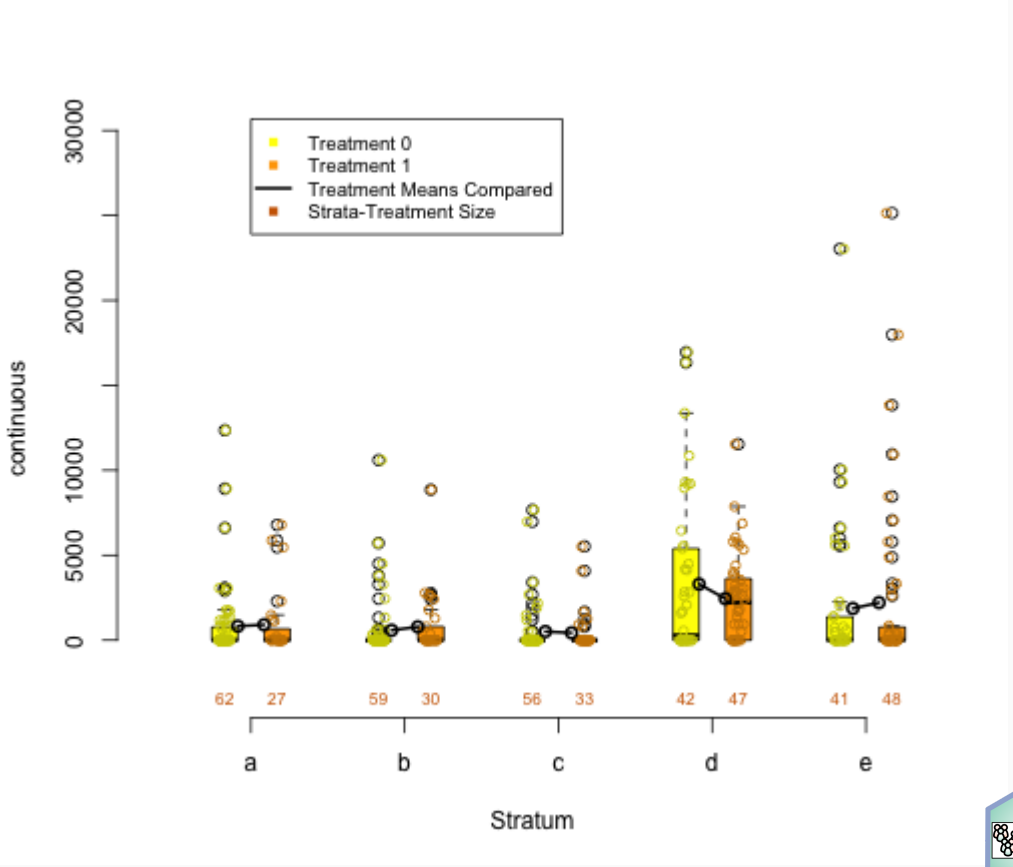


Checking Balance: Continuous Covariates (cont.)

```
box.psa(lalonde$educ, lalonde$treat, strata5)
```

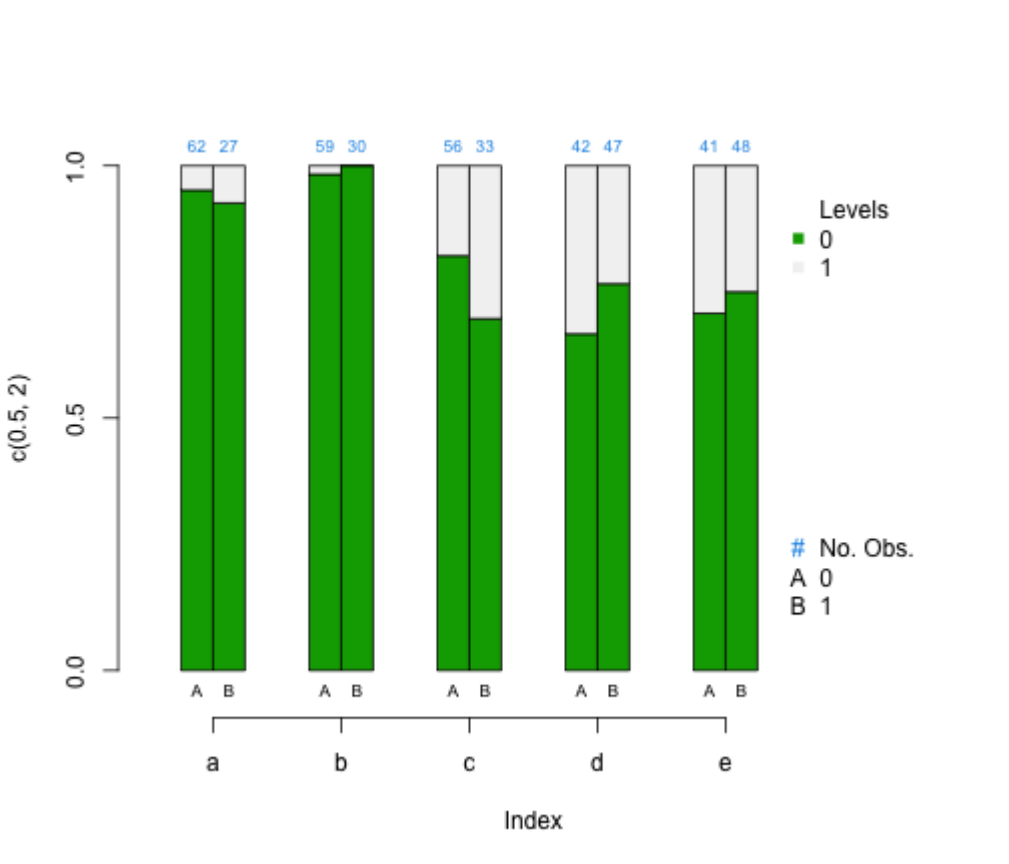



```
box.psa(lalonde$re75, lalonde$treat, strata5)
```

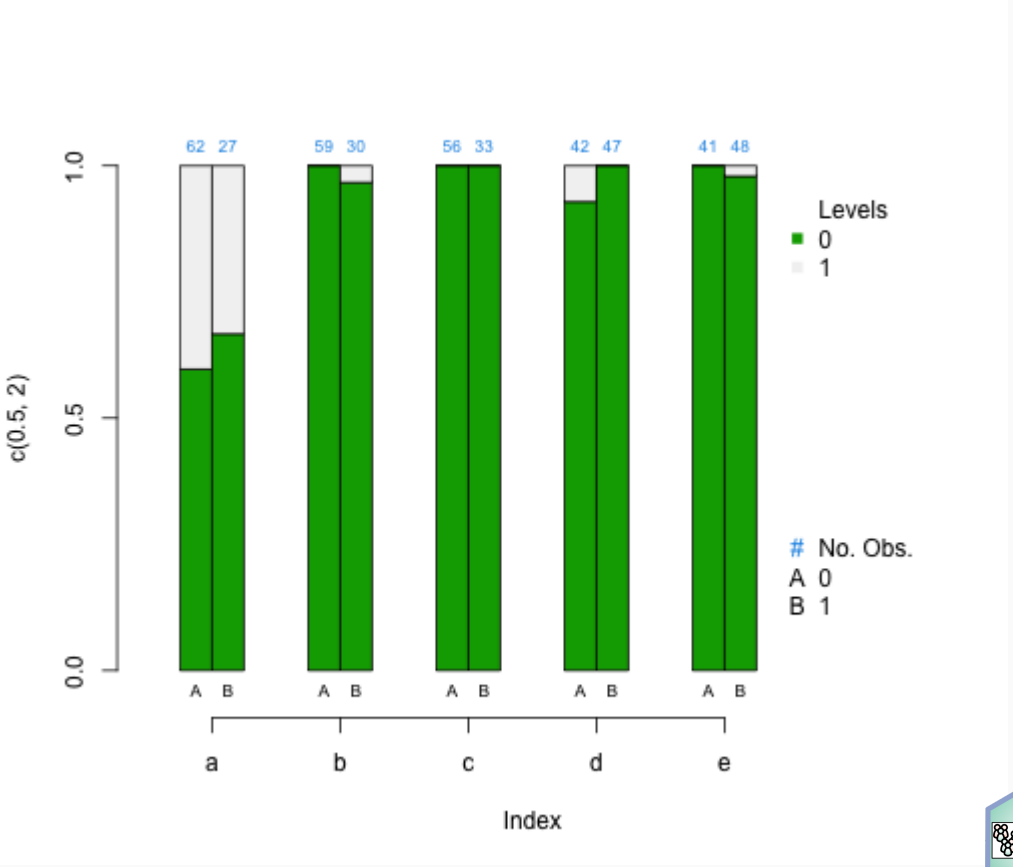


Checking Balance: Categorical Covariates

```
cat.psa(lalonde$married, lalonde$treat, strata5)
```

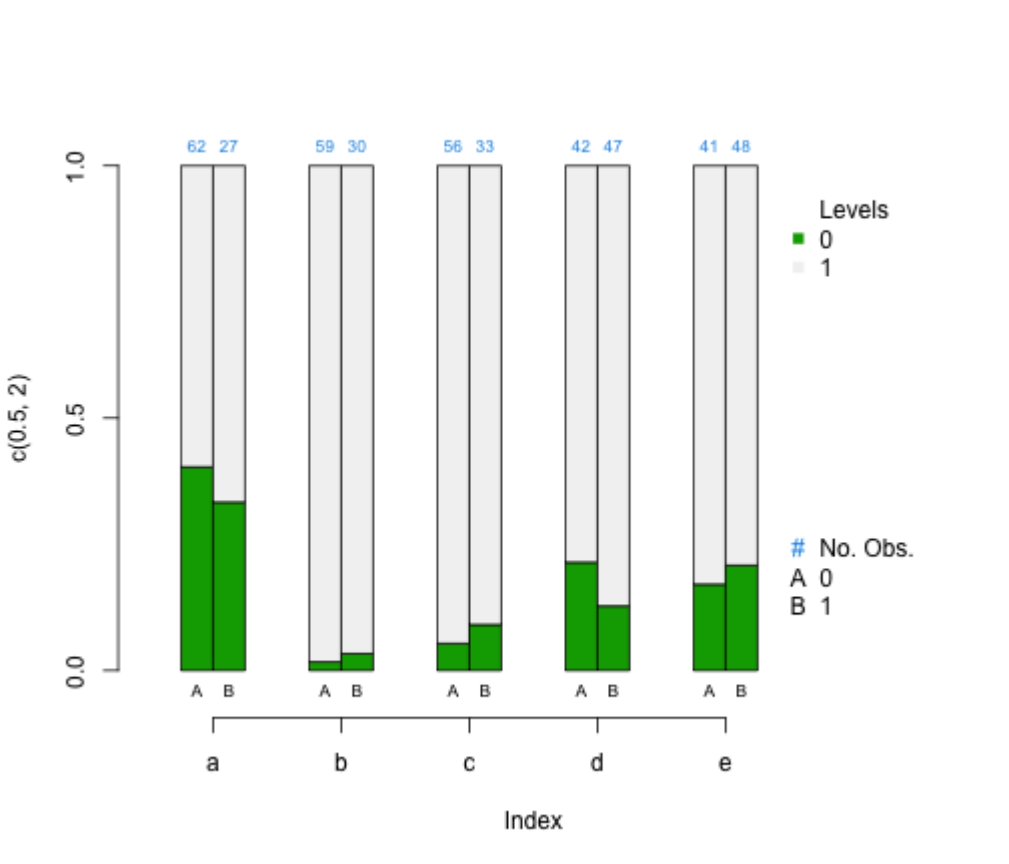



```
cat.psa(lalonde$hisp, lalonde$treat, strata5)
```

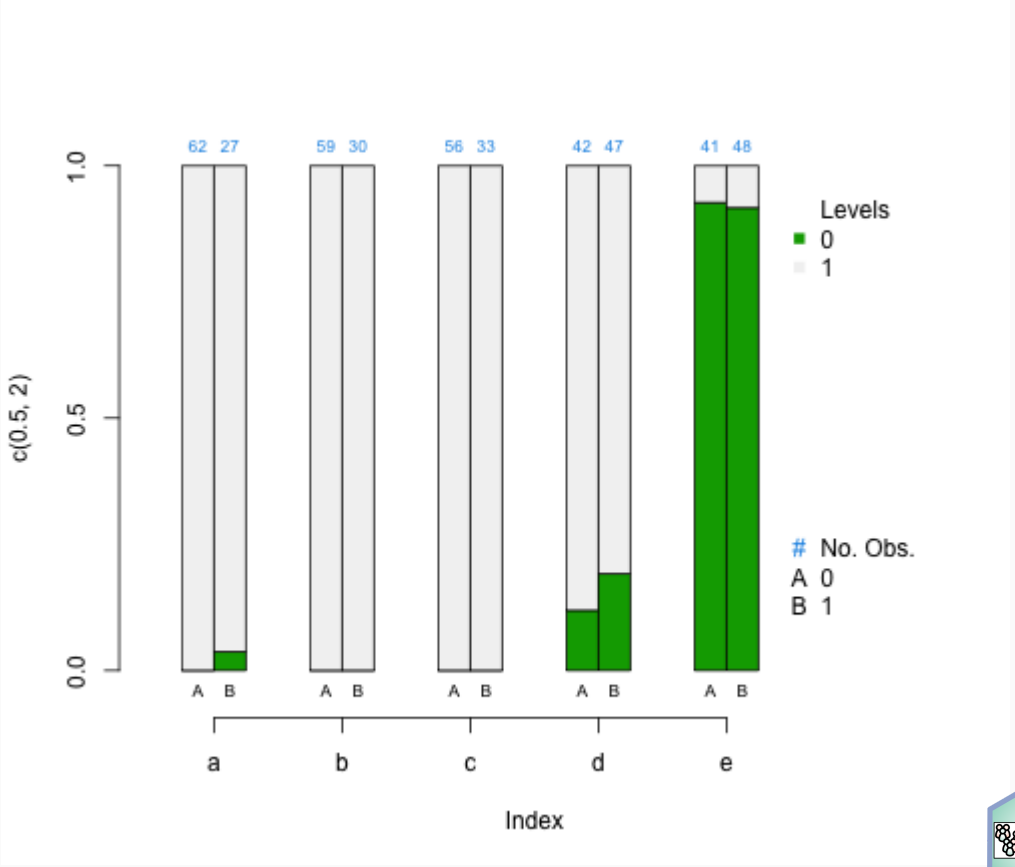


Checking Balance: Categorical Covariates (cont.)

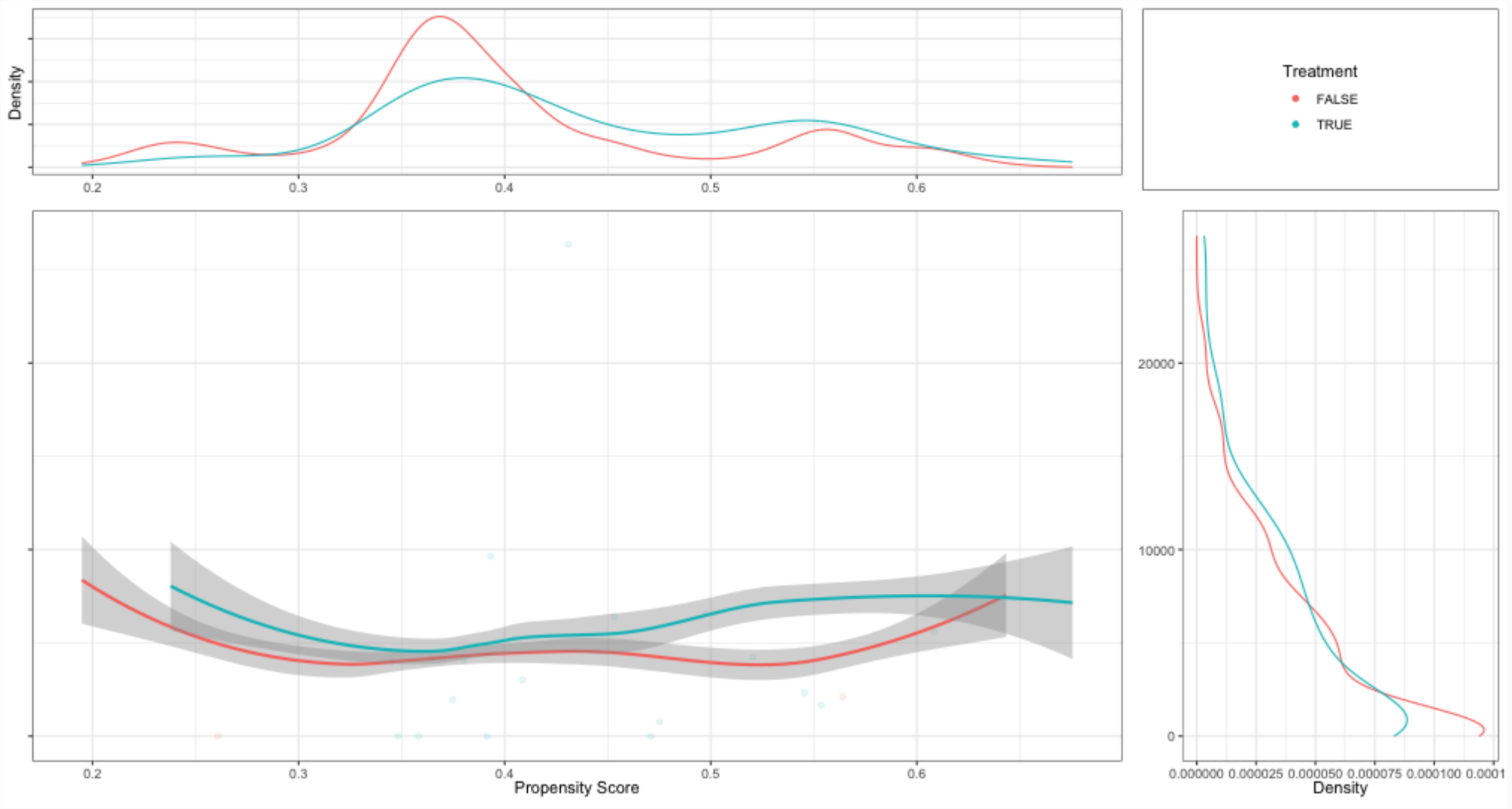
```
cat.psa(lalonde$black, lalonde$treat, strata5)
```




```
cat.psa(lalonde$nodegr, lalonde$treat, strata5)
```

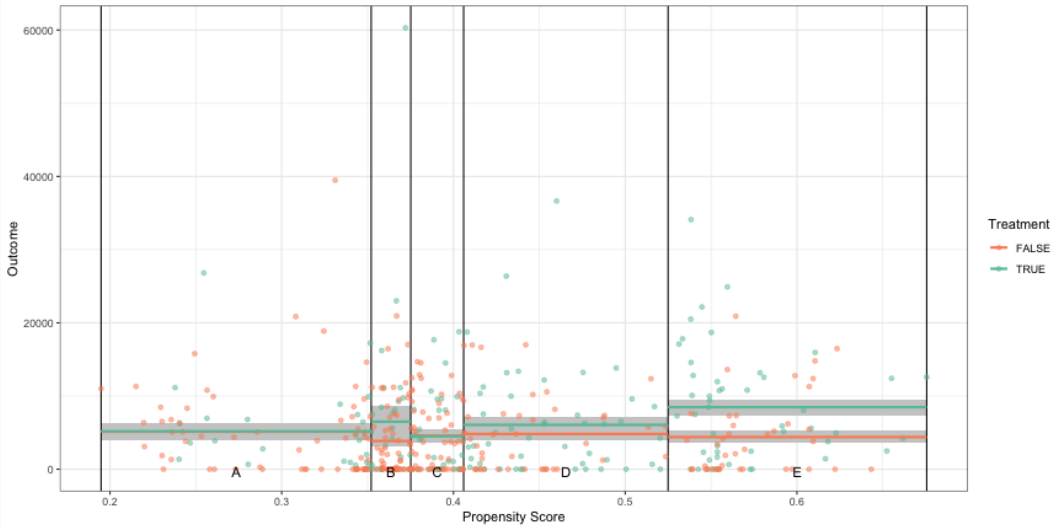


Loess Regression

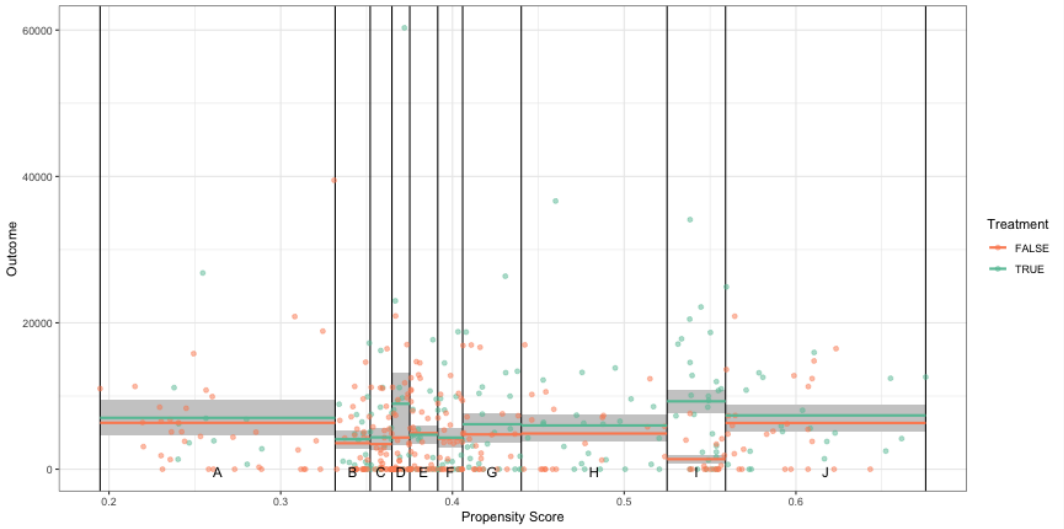


Stratification

```
psa::stratification_plot(ps = psadf$ps,
                         treatment = psadf$Tr,
                         outcome = psadf$Y,
                         n_strata = 5)
```

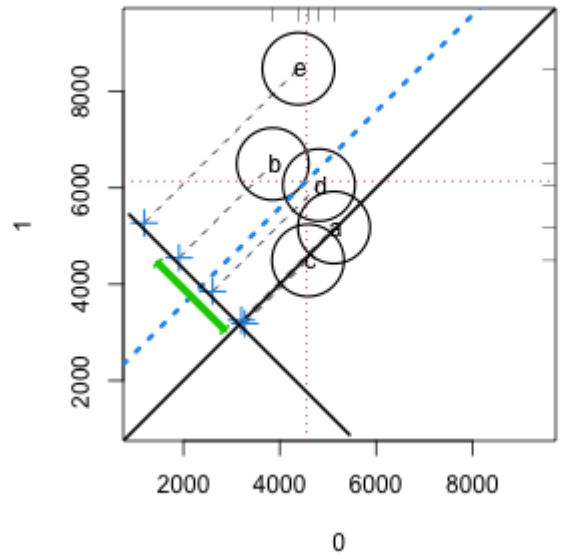



```
psa::stratification_plot(ps = psadf$ps,
                         treatment = psadf$Tr,
                         outcome = psadf$Y,
                         n_strata = 10)
```

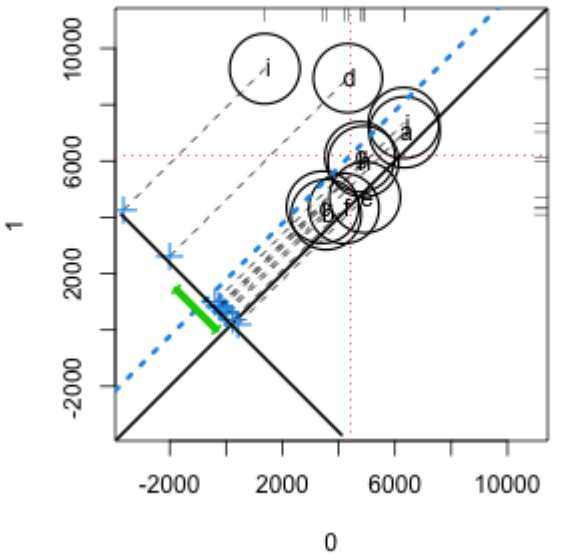


Stratification (cont.)

```
strata5 <- cut(lalonde$ps,
               quantile(lalonde$ps, seq(0, 1, 1/5)),
               include.lowest = TRUE,
               labels = letters[1:5])
circ.psa(lalonde$re78, lalonde$treat, strata5)
```



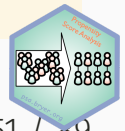

```
strata10 <- cut(lalonde$ps,
                quantile(lalonde$ps, seq(0, 1, 1/10)),
                include.lowest = TRUE,
                labels = letters[1:10])
circ.psa(lalonde$re78, lalonde$treat, strata10)
```



Stratification (cont.)

```
## $summary.strata
##   n.0 n.1  means.0  means.1
## a  62  27 5126.493 5178.073
## b  59  30 3855.200 6496.695
## c  56  33 4586.869 4495.076
## d  42  47 4814.028 6059.232
## e  41  48 4387.692 8474.201
##
## $wtd.Mn.0
## [1] 4554.056
##
## $wtd.Mn.1
## [1] 6140.655
##
## $ATE
## [1] 1586.599
##
## $se.wtd
## [1] 693.5067
##
## $approx.t
## [1] 2.287792
##
## $df
## [1] 435
##
## $CI.95
## [1]  223.5584 2949.6395
```

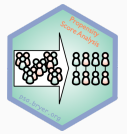
```
## $summary.strata
##   n.0 n.1  means.0  means.1
## a  35  10 6339.437 7019.962
## b  27  17 3554.157 4094.609
## c  31  16 3430.148 4356.532
## d  28  14 4325.792 8942.596
## e  30  15 4932.648 4710.588
## f  26  18 4187.895 4315.483
## g  22  22 4755.015 6148.795
## h  20  25 4878.944 5980.416
## i  16  28 1375.014 9276.448
## j  25  20 6315.806 7351.056
##
## $wtd.Mn.0
## [1] 4414.111
##
## $wtd.Mn.1
## [1] 6195.262
##
## $ATE
## [1] 1781.151
##
## $se.wtd
## [1] 710.5964
##
## $approx.t
## [1] 2.506559
##
## $df
## [1] 425
##
## $CI.95
## [1]  384.4306 3177.8724
```



Matching

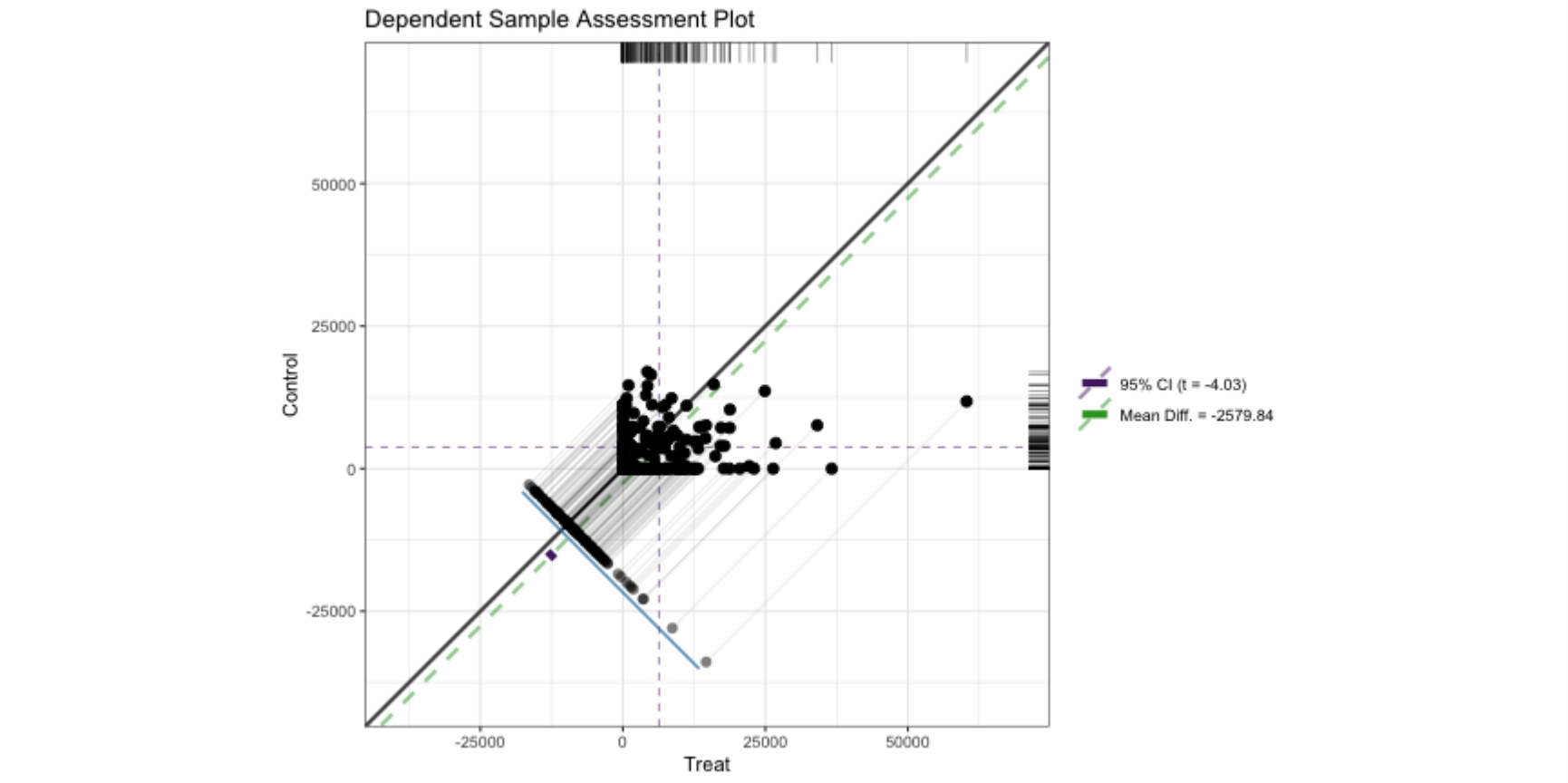
```
rr <- Match(Y = lalonde$re78,
            Tr = lalonde$treat,
            X = lalonde$ps,
            M = 1,
            estimand = 'ATT',
            ties = FALSE)
summary(rr)
```

```
##
## Estimate...  2579.8
## SE.........  637.69
## T-stat.....  4.0456
## p.val......  5.2189e-05
##
## Original number of observations..............  445
## Original number of treated obs...............  185
## Matched number of observations...............  185
## Matched number of observations  (unweighted).  185
```



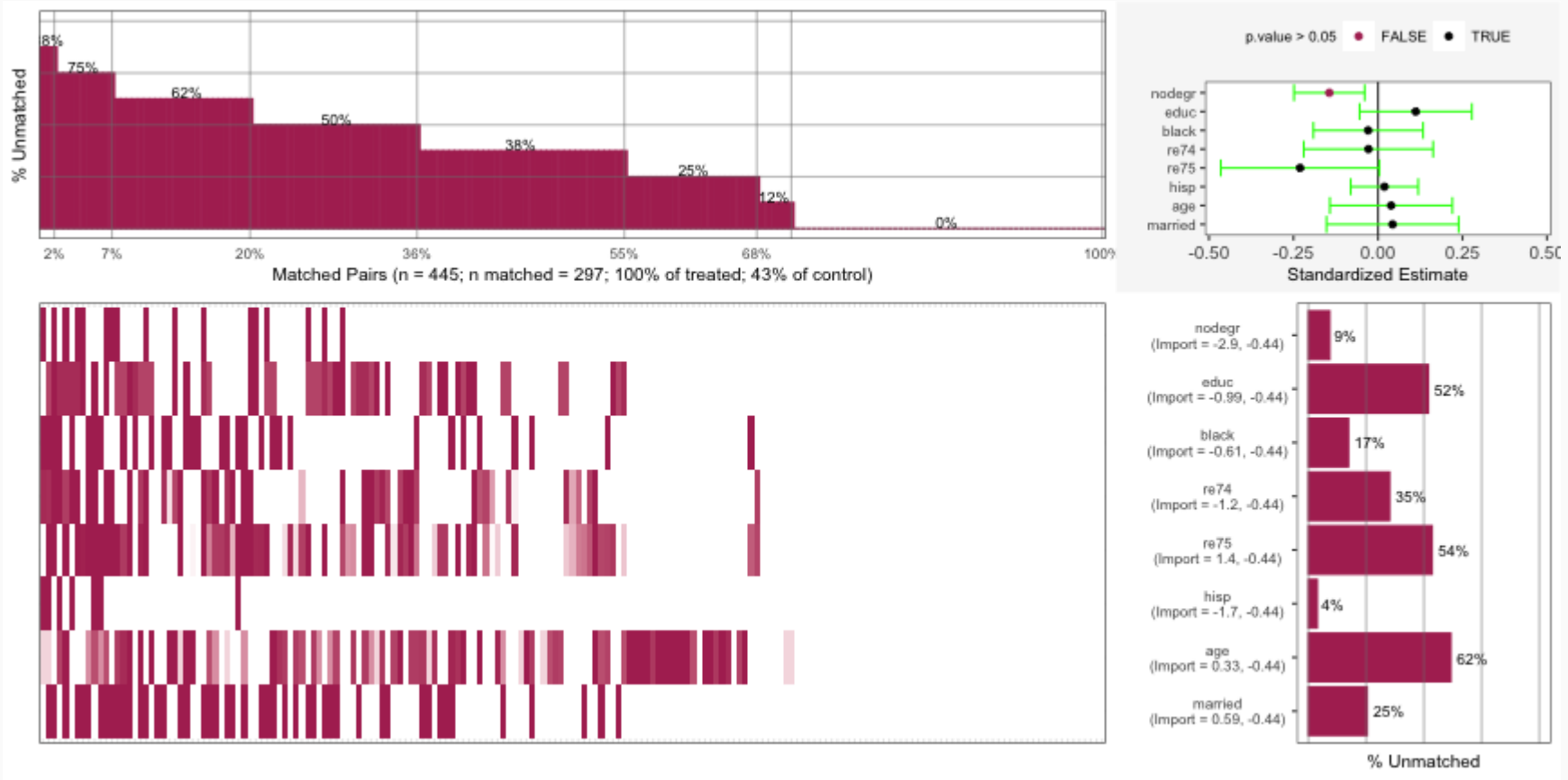
Visualizing Matching Results

```
matches <- data.frame(Treat = lalonde[rr$index.treated,'re78'],
                      Control = lalonde[rr$index.control,'re78'])
granovagg.ds(matches[,c('Control','Treat')], xlab = 'Treat', ylab = 'Control')
```



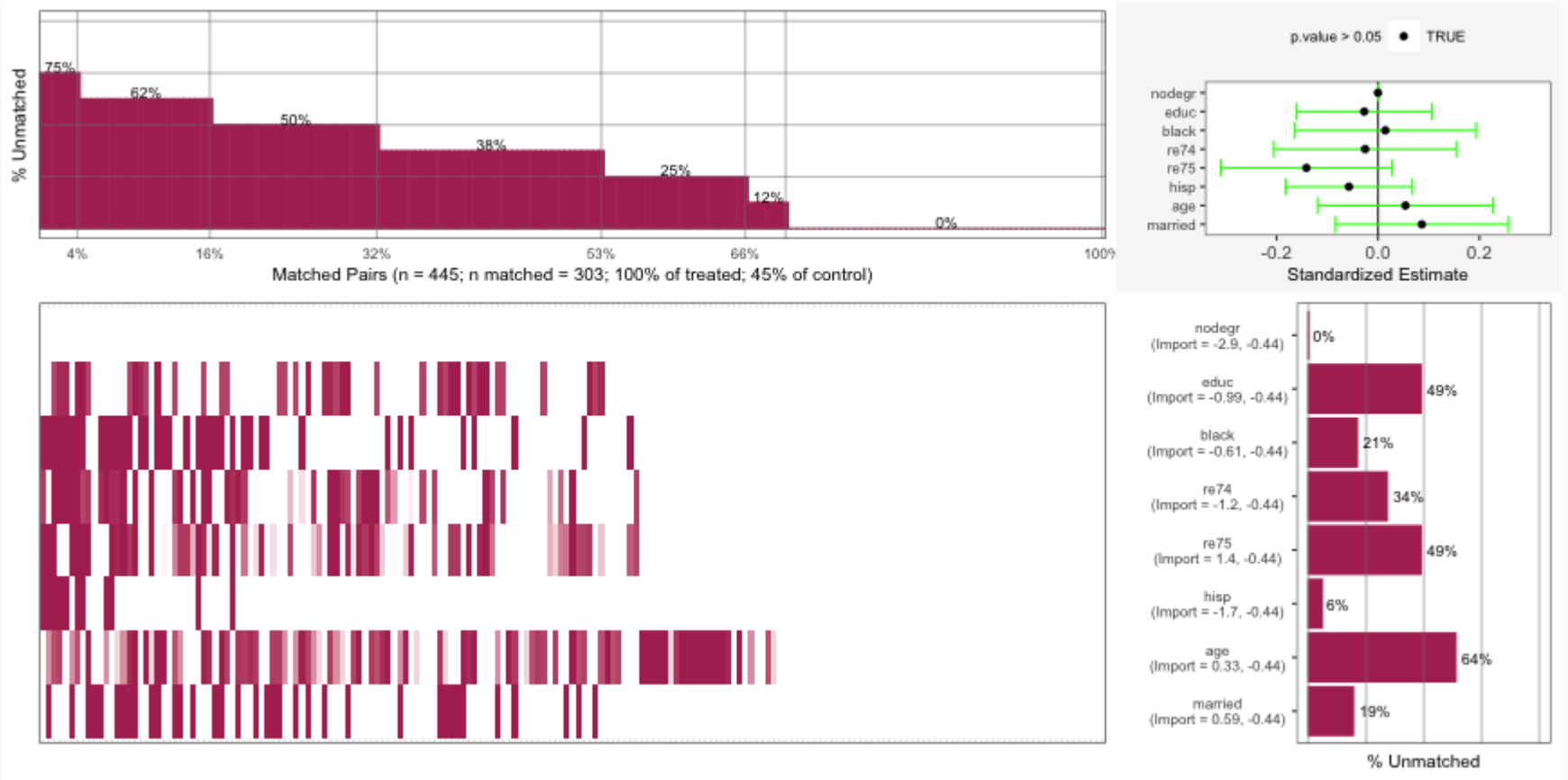
Balance for Matching

```
psa::MatchBalance(df = lalonde, formu = lalonde.formu,
                  formu.Y = update.formula(lalonde.formu, re78 ~ .),
                  M = 1, estimand = 'ATT', ties = FALSE) |> plot()
```

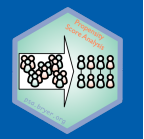


Balance for Matching (cont.)

```
psa::MatchBalance(df = lalonde, formu = lalonde.formu,
                  formu.Y = update.formula(lalonde.formu, re78 ~ .),
                exact.covs = c('nodegr'),
                  M = 1, estimand = 'ATT', ties = FALSE) |> plot()
```



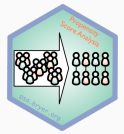
Sensitivity Analysis



Sensitivity Analysis

- An observational study is free of hidden bias if the propensity scores for each subject depend only on the observed covariates.
- That is, the *p*-value is valid *if* there are no unobserved confounders.
- However, there are very likely covariates that would better model treatment. These introduce hidden bias.
- Hidden bias exists if two subjects have the same covariates, but different propensity scores.

$X_a = X_b$ but $\pi_a \neq \pi_b$ for some a and b.



Sensitivity Analysis

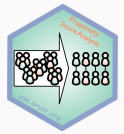
Each person in the treatment is matched to exactly one person in the control. The odds of being in the treatment for persons a and b are:

$O_a = \frac{\pi_a}{1-\pi_a}$ and $O_b = \frac{\pi_b}{1-\pi_b}$

The ratio of these odds, Γ, measures the bias after matching.

$$\Gamma = \frac{O_a}{O_b} = \frac{\pi_a/(1-\pi_a)}{\pi_b/(1-\pi_b)}$$

This is the ratio of the odds the treated unit being in the treatment group to the matched control unit being in the treatment group.



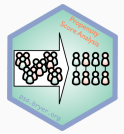
Sensitivity Analysis

Sensitivity analysis tests whether the results hold for various ranges of Γ. That is, we test how large the differences in π (i.e. propensity scores) would have to be to change our basic inference. Let p_a and p_b be the probability of each unit of the matched pair being treated, conditional on exactly one being treated. For example:

- If $\Gamma = 1$, the treatment and control unit within each pair has the same value of treatment assignment ($p_a = 0.5$ and $p_b = 0.5$).
- If $\frac{1}{2} \leq \Gamma \leq 2$, no unit can be more than twice as likely as its match to get treated ($0.33 \leq p_a, p_b \leq 0.66$).
- If $\frac{1}{3} \leq \Gamma \leq 3$, no unit can be more than three times as likely as its match to get treated ($0.25 \leq p_a, p_b \leq 0.75$)

To get the bounds:

$$\frac{1}{\Gamma + 1} \leq p_a, p_b \leq \frac{\Gamma}{\Gamma + 1}$$

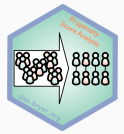


Wilcoxon Signed Rank Test

- Drop pairs where the matches have the same outcome.
- Calculate the difference in outcomes within each pair.
- Rank the pairs from smallest absolute difference to largest absolute difference (i.e. the smallest = 1).
- Take the sum of the ranks where the treated unit had the higher outcome.

$$W = \left| \sum_{1}^{N_r} sgn(x_{T,i} - x_{C,i}) \cdot R_i \right|$$

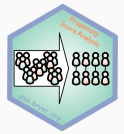
Where N is the number of ranked pairs; R_i is the rank for pair r; $x_{T,i}$ and $x_{C,i}$ are the outcomes for the i^{th} treated and control pair, respectively.



Sensitivity Analysis

The process for sensitivity analysis:

- Select a series of values for Γ. For social science research, values between 1 and 2 is an appropriate start.
- For each Γ, estimate the *p*-values to see how the *p*-values increase for larger values of Γ.
- For binary outcomes, use McNemar's test, for all others use Wilcoxon sign rank test and the Hodges-Lehmann point estimate. See Keele (2010) for more information.

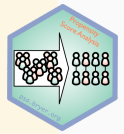


Sensitivity Analysis

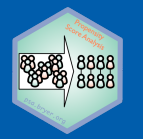
Children of parents who had worked in a factory where lead was used in making batteries were matched by age, exposure to traffic, and neighborhood with children whose parents did not work in lead-related industries. Whole blood was assessed for lead content yielding measurements in mg/dl

```
require(rbounds)
psens(lalonde$re78[rr$index.treated],
      lalonde$re78[rr$index.control],
      Gamma = 2, GammaInc = 0.1)
```

```
##
##  Rosenbaum Sensitivity Test for Wilcoxon Signed Rank P-Value
##
## Unconfounded estimate ....  2e-04
##
##  Gamma Lower bound Upper bound
##    1.0       2e-04      0.0002
##    1.1       0e+00      0.0016
##    1.2       0e+00      0.0069
##    1.3       0e+00      0.0215
##    1.4       0e+00      0.0527
```



Bootstrapping Propensity Score Analysis

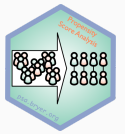


Boostrapping

- Bootstrapping was first introduced by Efron (1979) in *Bootstrap Methods: Another Look at the Jackknife*.
- Estimates confidence of statistics by resampling *with* replacement.
- The *bootstrap sample* provides an estimate of the sampling distribution.

For PSA, sensitivity analysis is only well defined for matched samples.

Rosenbaum (2012) suggested that one way to test for sensitivity of model selection is to test the null hypothesis twice.



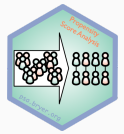
Bootstrapping Propensity Score Analysis

The `PSAboot` implements bootstrapping for propensity score analysis.

1. A stratified bootstrap sample is drawn to ensure the ratio of treatment-to-control observations is the same (i.e. sampling with replacement is done for the treatment and control observations is done separately). Note that the `control.ratio` and `treat.ratio` parameters allow for under sampling in the case of imbalanced data.

2. For each bootstrap sample balance statistics and treatment effects are estimated using each method (five by default).

3. Overall treatment effect with confidence interval is estimated from the bootstrap samples.

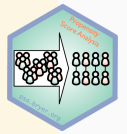


PSA Bootstrapping Example

```
library(PSAboot)
psaboot <- PSAboot(Tr = lalonde$treat,
                   Y = lalonde$re78,
                   X = lalonde,
                   formu = lalonde.formu)
```

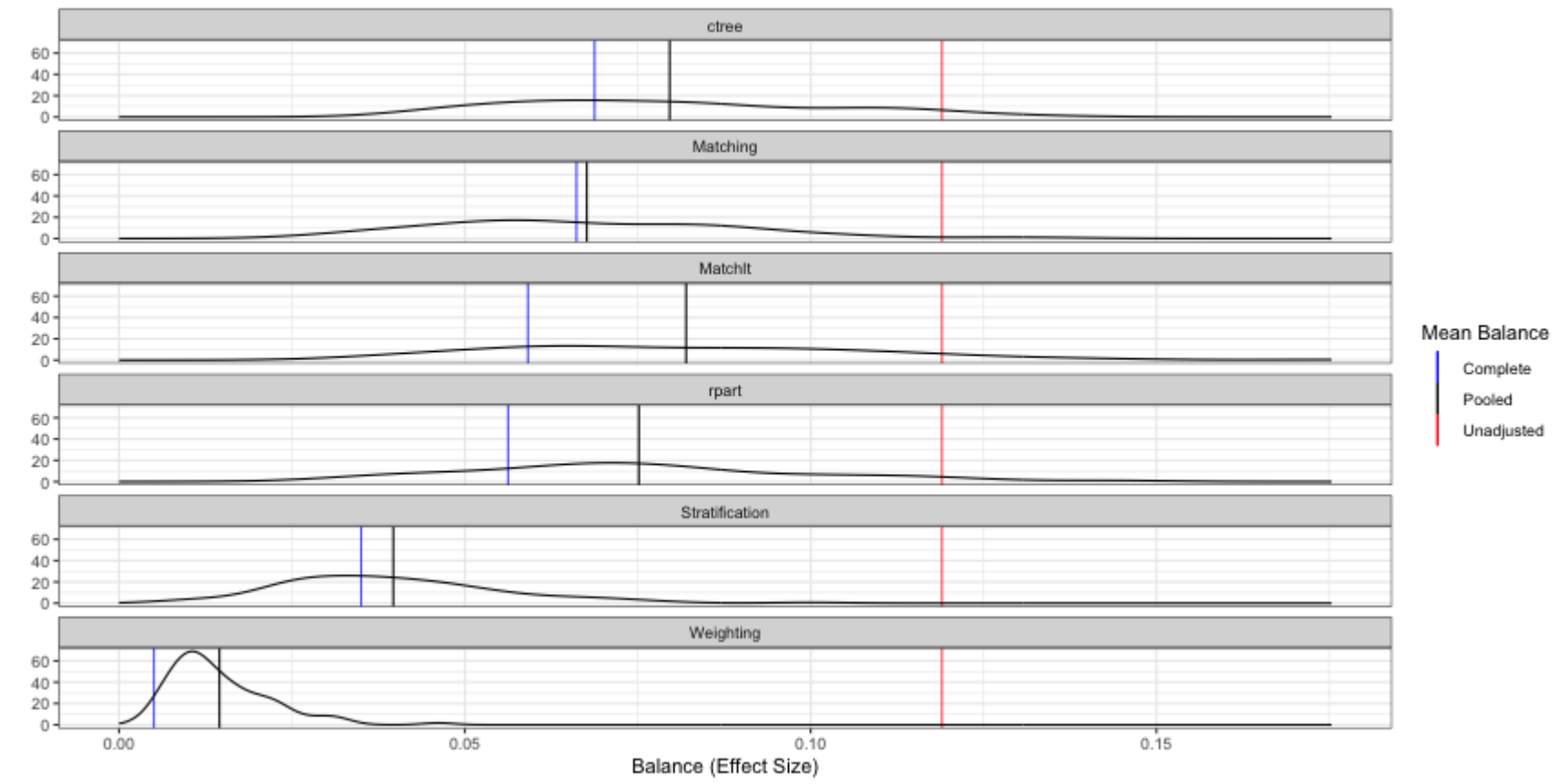
```
summary(psaboot)
```

```
## Stratification Results:
##    Complete estimate = 1587
##    Complete CI = [224, 2950]
##    Bootstrap pooled estimate = 1610
##    Bootstrap weighted pooled estimate = 1558
##    Bootstrap pooled CI = [234, 2985]
##    68% of bootstrap samples have confidence intervals that do not span zero.
##       68% positive.
##       0% negative.
## ctree Results:
##    Complete estimate = 1598
##    Complete CI = [-6.62, 3203]
##    Bootstrap pooled estimate = 1654
##    Bootstrap weighted pooled estimate = 1643
##    Bootstrap pooled CI = [228, 3080]
```



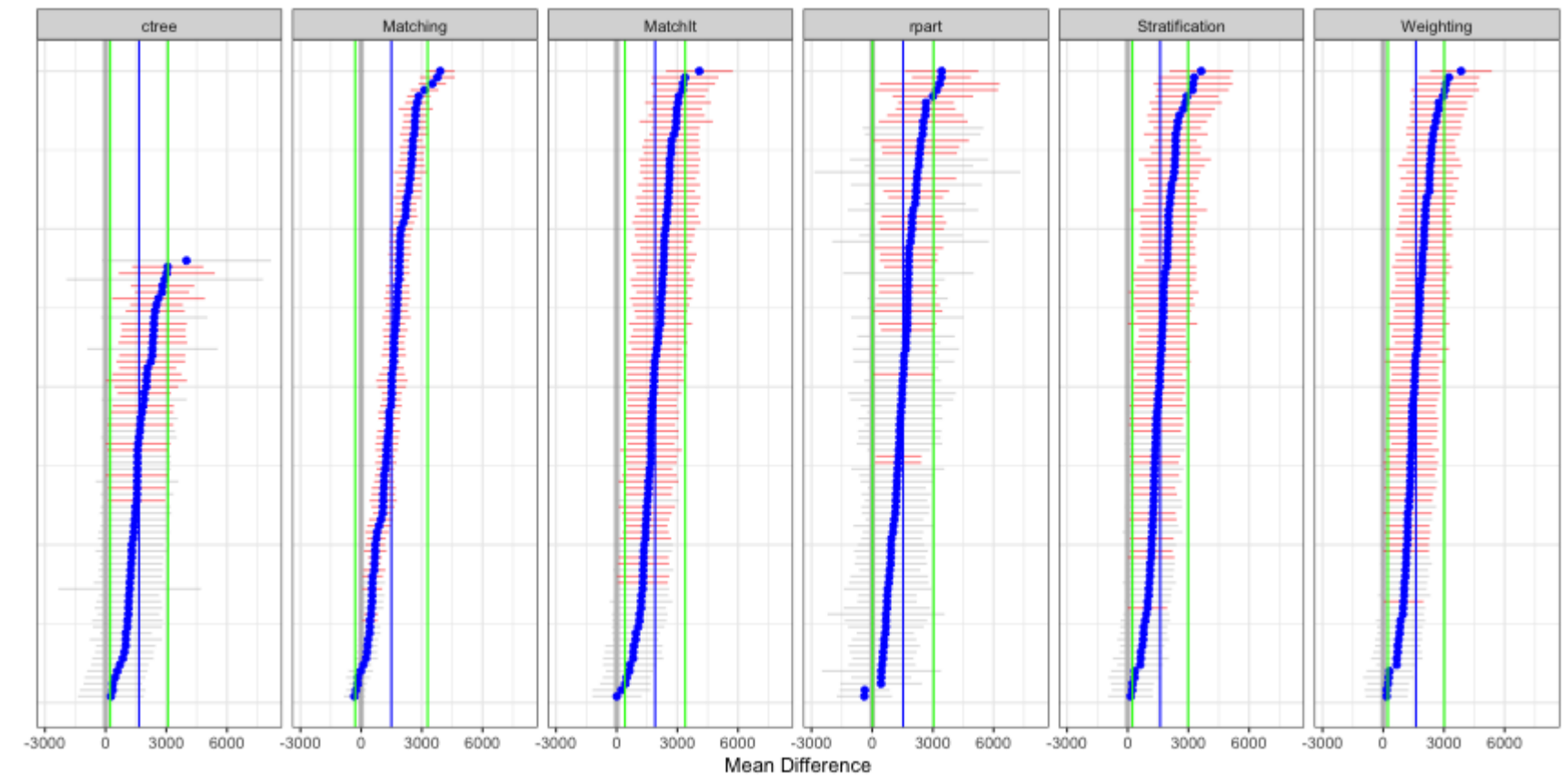
Checking Balance

```
psaboot_bal <- balance(psaboot)
plot(psaboot_bal)
```



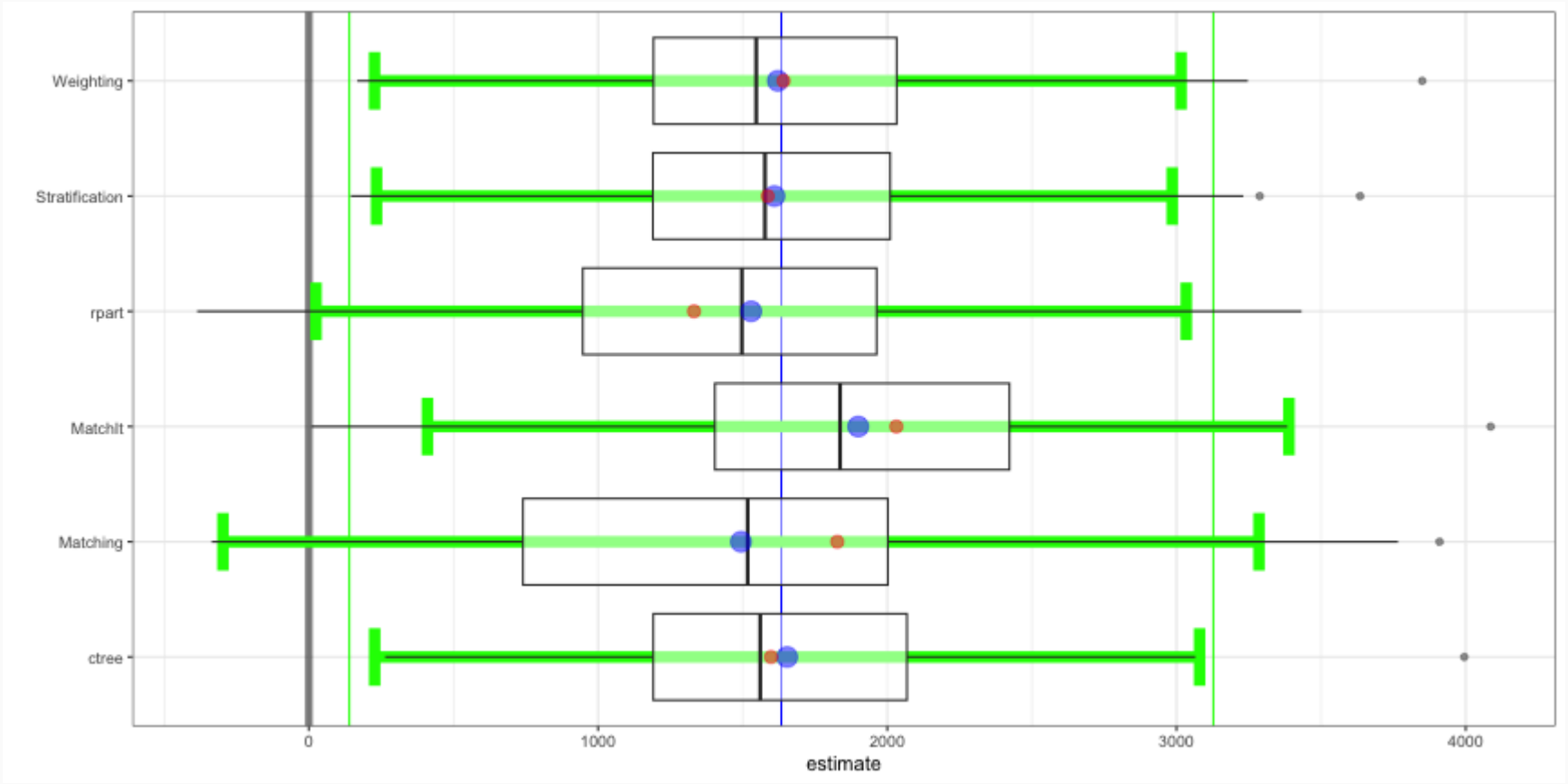
Plotting all Boostrap Samples

```
plot(psaboot)
```



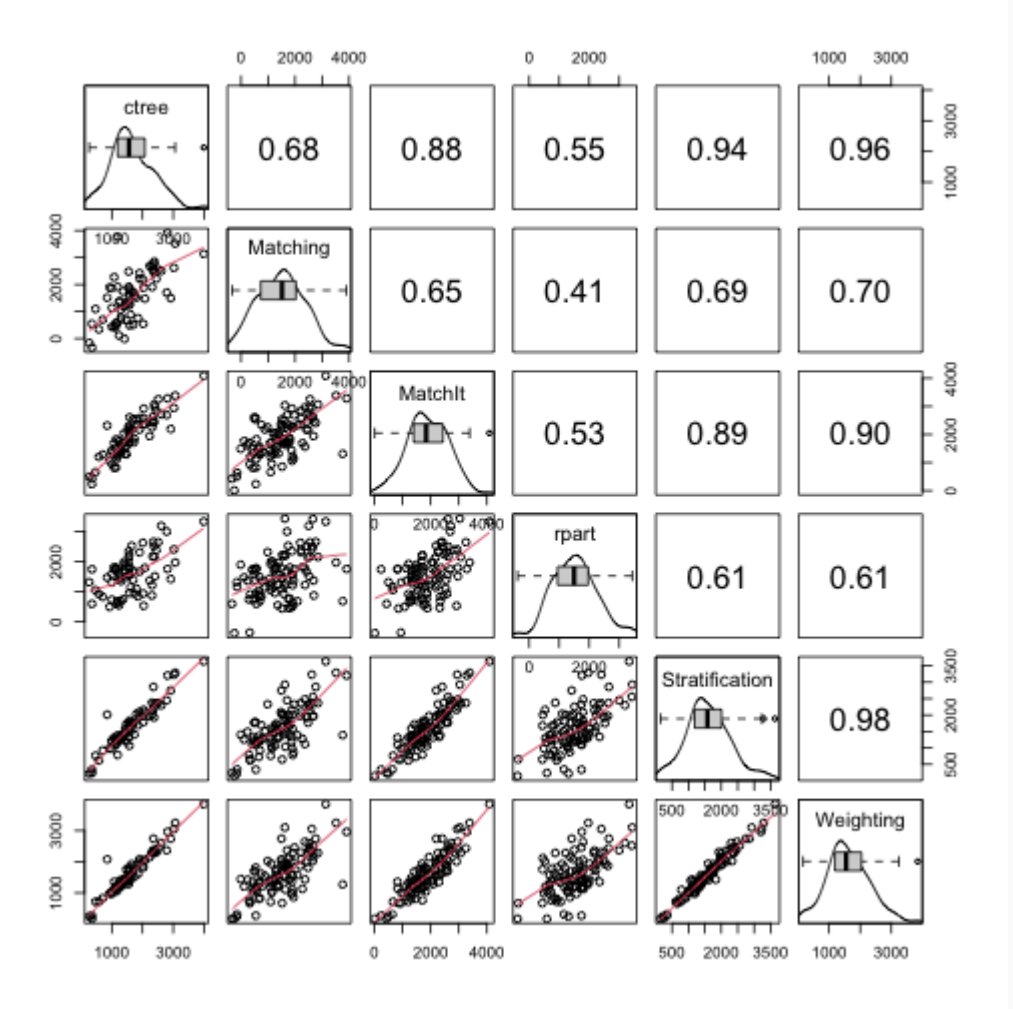
Boxplot of Boostrapping PSA

```
boxplot(psaboot)
```

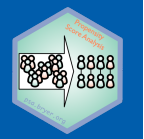


Comparing Across PSA Methods

```
matrixplot(psaboot)
```

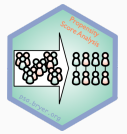


Matching of Non-Binary Treatments



Matching of Non-Binary Treatments

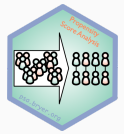
- The `TriMatch` package provides functions for finding matched triplets.
- Estimates propensity scores for three separate logistic regression models (one for each pair of groups, that is, treat1-to-control, treat2-to-control, and treat1-to-treat2).
- Finds matched triplets that minimize the total distance (i.e. sum of the standardized distance between propensity scores within the three models). within a caliper.
- Provides multiple methods for determining which matched triplets are retained:
- Optimal which attempts to retain all treatment units.
- Full which retains all matched triplets within the specified caliper (.25 by default as suggested by Rosenbaum).
- Analog of the one-to-many for matched triplets. Specify how many times each treat1 and treat2 unit can be matched.
- Unique which allows each unit to be matched once, and only once.
- Functions for conducting repeated measures ANOVA and Freidman Ranksum Tests are provided.



Example: Tutoring

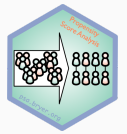
Students can opt to utilize tutoring services to supplement math courses. Of those who used tutoring services, approximately 58% of students used the tutoring service once, whereas the remaining 42% used it more than once. Outcome of interest is course grade.

- **Military** Active military status.
- **Income** Income level.
- **Employment** Employment level.
- **NativeEnglish** Is English their native language
- **EdLevelMother** Education level of their mother.
- **EdLevelFather** Education level of their father.
- **Ethnicity** American Indian or Alaska Native, Asian, Black or African American, Hispanic, Native Hawaiian or Other Pacific Islander, Two or more races, Unknown, White
- **Gender** Male, Female
- **Age** Age at course start.
- **GPA** Student GPA at the beginning of the course.



New Student Outreach: Covariates

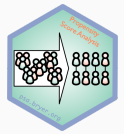
Newly enrolled students received outreach contacts until they registered for a course or six months have passed, whichever came first. Outreach was conducted by two academic advisors and a comparison group was drawn from students who enrolled prior to the start of the outreach program. Outcome of interest is number of credits attempted within the first seven months of enrollment.



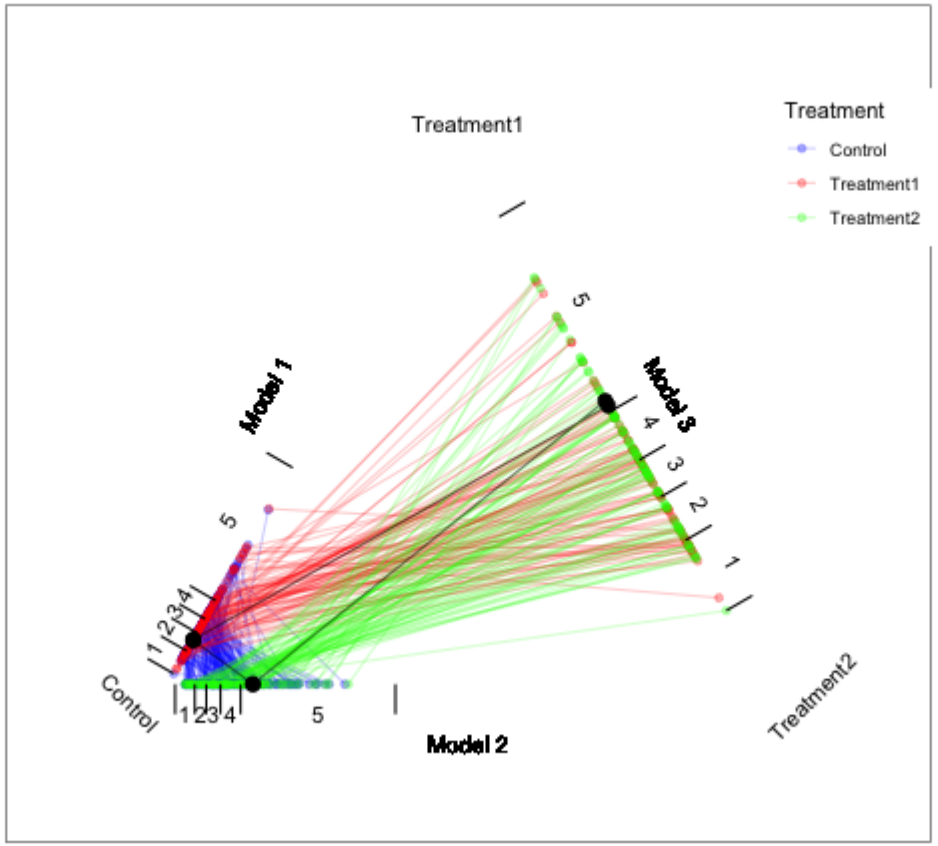
PSA for Non-Binary Treatments

The `TriMatch` algorithm works as follows:

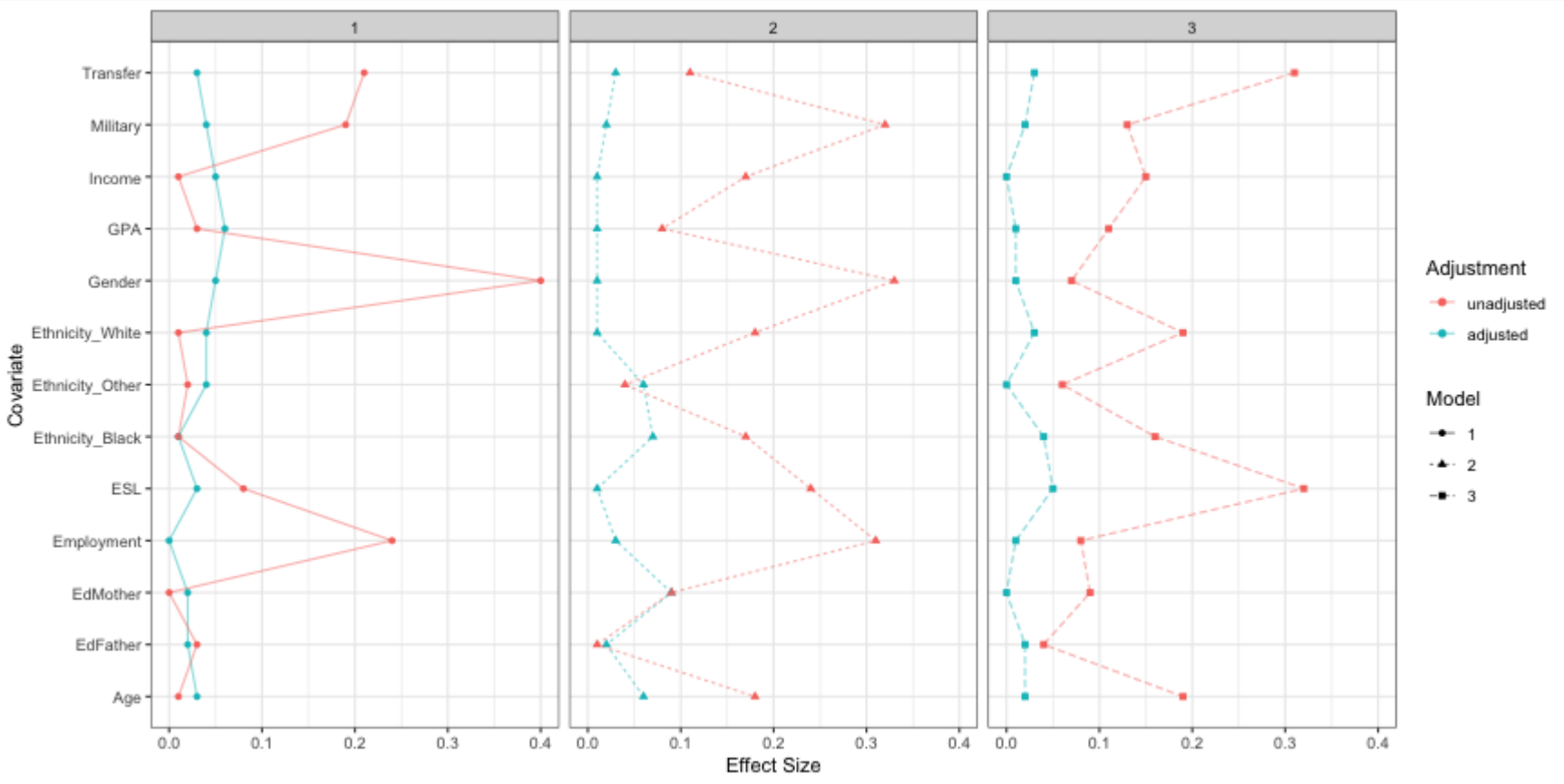
1. Estimate three separate propensity score models for each pair of groups (i.e. Control-to-Treat1, Control-to-Treat2, Treat1-to-Treat2).
2. Determine the matching order. The default is to start with the largest of two treatments, then the other treatment, followed by the control.
3. For each unit in group 1, find all units from group 2 within a certain threshold (i.e. difference between PSs is within a specified caliper).
4. For each unit in group 2, find all units from group 3 within a certain threshold.
5. Calculate the distance (difference) between each unit 3 found and the original unit 1. Eliminate candidates that exceed the caliper.
6. Calculate a total distance (sum of the three distances) and retain the smallest unique *M* group 1 units (by default *M*=2)



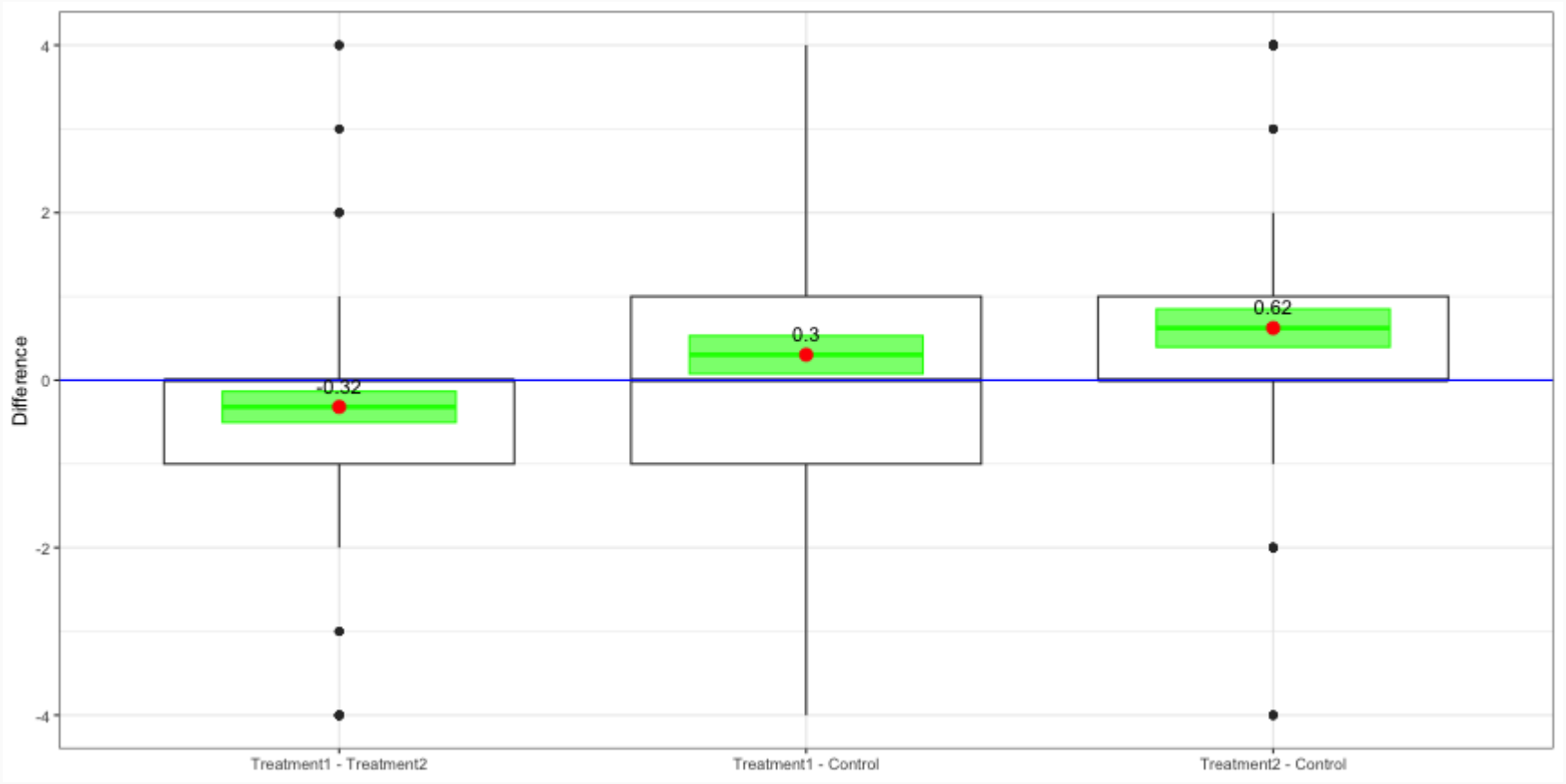
Matching Triplets



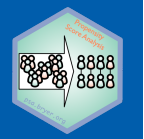
Checking Balance



Results


Difference
4
2
0
-2
-4
-0.32
0.3
0.62
Treatment1 - Treatment2
Treatment1 - Control
Treatment2 - Control

Multilevel Propensity Score Analysis



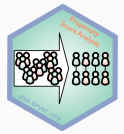
Multilevel PSA

The use of PSA for clustered, or multilevel data, has been limited (Thoemmes \& Felix, 2011). Bryer and Pruzek (2012, 2013) have introduced an approach to analyzing multilevel or clustered data using stratification methods and implemented in the `multilevelPSA` R package.

- Exact and partially exact matching methods implicitly adjust for clustering. That is, the covariates chosen to exactly match are, in essence, clustering variables.
- Exact matching only applies to phase I of PSA. How are the clusters related to outcome of interest.

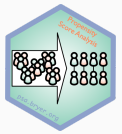
The `multilevelPSA` uses stratification methods (e.g. quintiles, classification trees) by:

- Estimate separate propensity scores for each cluster.
- Identify strata within each cluster (e.g. leaves of classification trees, quintiles).
- Estimate ATE (or ATT) within each cluster.
- Aggregate estimated ATE to provide an overall ATE estimate.
- Several functions to summarize and visualize results and check balance.



The Programme of International Student Assessment

- International assessment conducted by the Organization for Economic Co-operation and Development (OECD).
- Assesses students towards the end of secondary school (approximately 15-year-old children) in math, reading, and science.
- Collects a robust set of background information from students, parents, teachers, and schools.
- Assess both private and public school students in many countries.
- We will use PISA to estimate the effects of private school attendance on PISA outcomes.

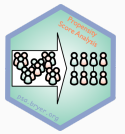


Phase I of Multilevel PSA

The `multilevelPSA` provides two functions, `mlpsa.ctree` and `mlpsa.logistic`, that will estimate propensity scores using classification trees and logistic regression, respectively. Since logistic regression requires a complete dataset (i.e. no missing values), we will use classification trees in this example.

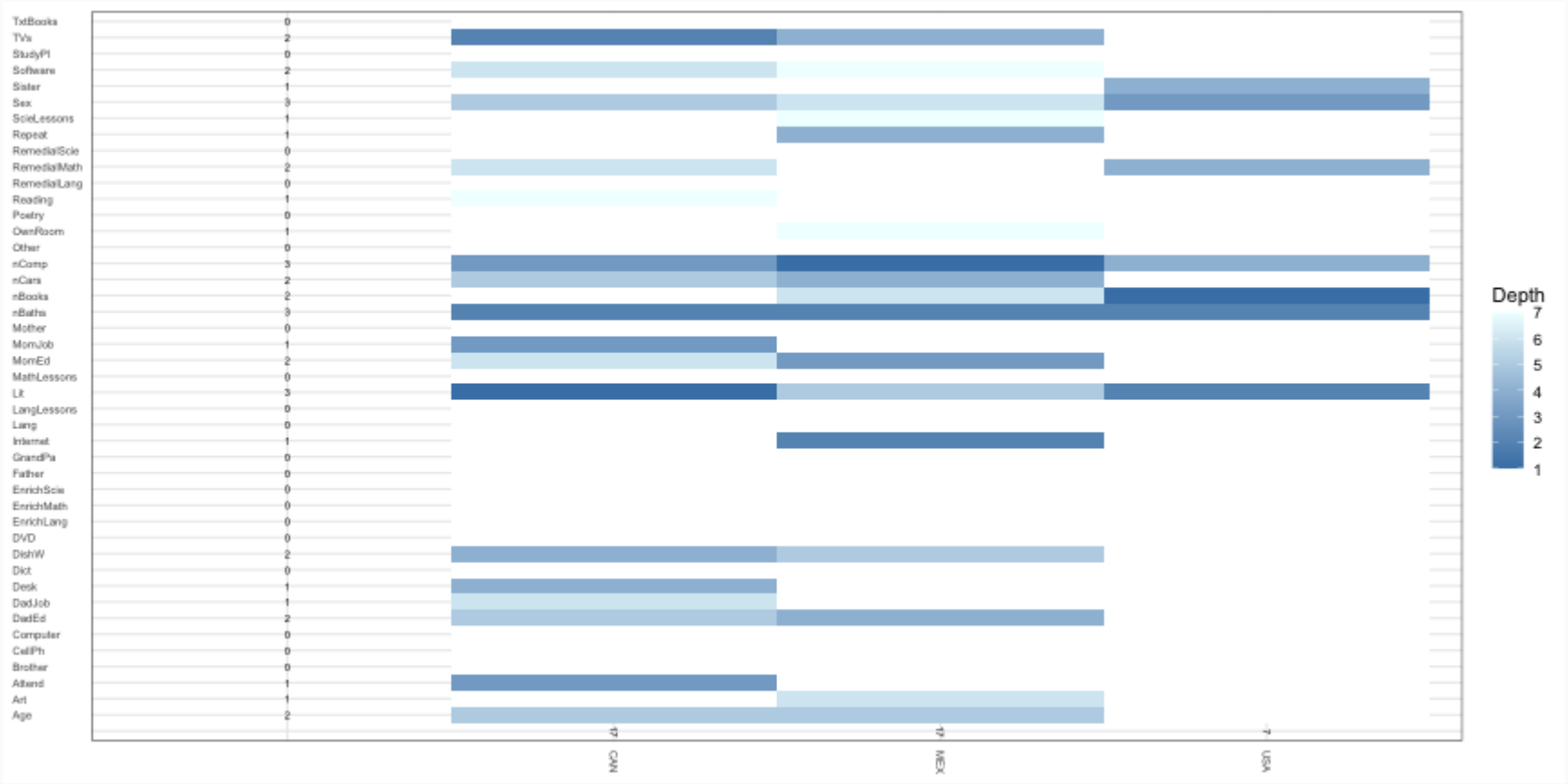
```
data(pisana)
data(pisa.colnames)
data(pisa.psa.cols)
student = pisana
mlctree = mlpsa.ctree(student[,c('CNT','PUBPRIV',pisa.psa.cols)],
                      formula=PUBPRIV ~ ., level2='CNT')
student.party = getStrata(mlctree, student, level2='CNT')
student.party$mathscore = apply(
student.party[,paste0('PV', 1:5, 'MATH')], 1, sum) / 5
```

To assess what covariates were used in each tree model, as well as the relative importance, we can create a heat map of covariate usage by level.



Covariate Heat Map

```
tree.plot(mlctree, level2Col=student$CNT, colLabels=pisa.colnames[,c('Variable','ShortDesc')])
```



Phase II of Multilevel PSA

The `mlpsa` function will compare the outcome of interest.

```
results.psa.math = mlpsa(response=student.party$mathscore,
                         treatment=student.party$PUBPRIV, strata=student.party$strata,
                         level2=student.party$CNT, minN=5)
results.psa.math$overall.wtd
```

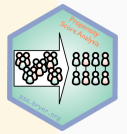
```
## [1] -28.02406
```

```
results.psa.math$overall.ci
```

```
## [1] -31.30261 -24.74552
```

```
results.psa.math$level2.summary[,c('level2','Private','Private.n',
'Public','Public.n','diffwtd','ci.min','ci.max')]
```

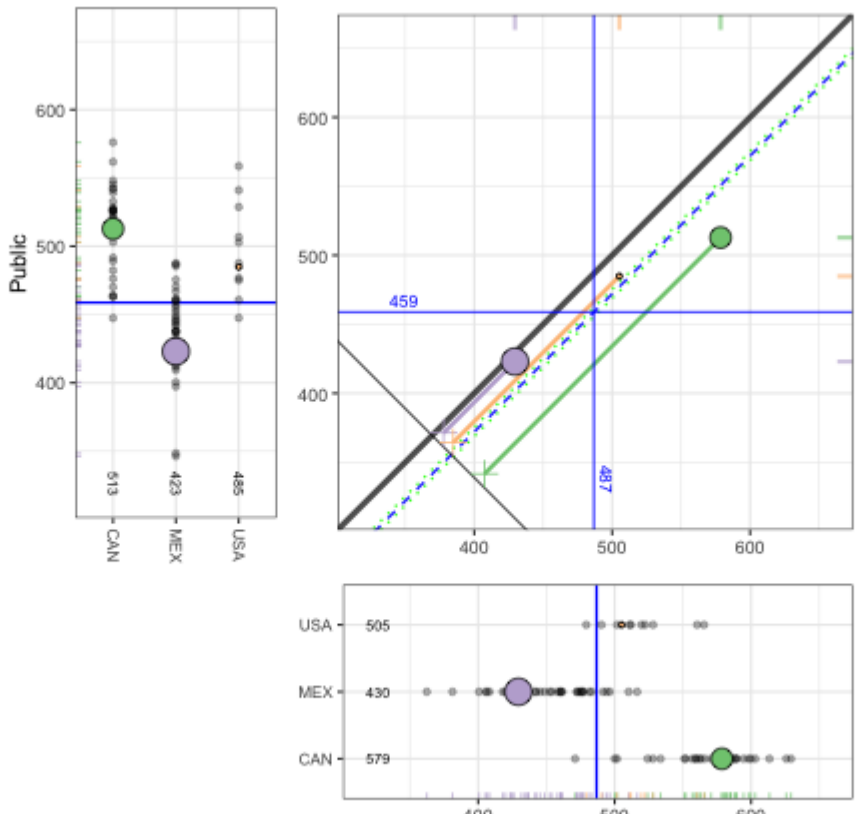
```
##   level2  Private Private.n   Public Public.n    diffwtd    ci.min     ci.max
## 1    CAN 578.6262      1625 512.7997    21093 -65.826528 -72.08031 -59.572751
## 2    MEX 429.5247      4044 422.9746    34090  -6.550102 -10.04346  -3.056743
## 3    USA 505.2189       345 484.8212     4888 -20.397746 -32.03916  -8.756334
```



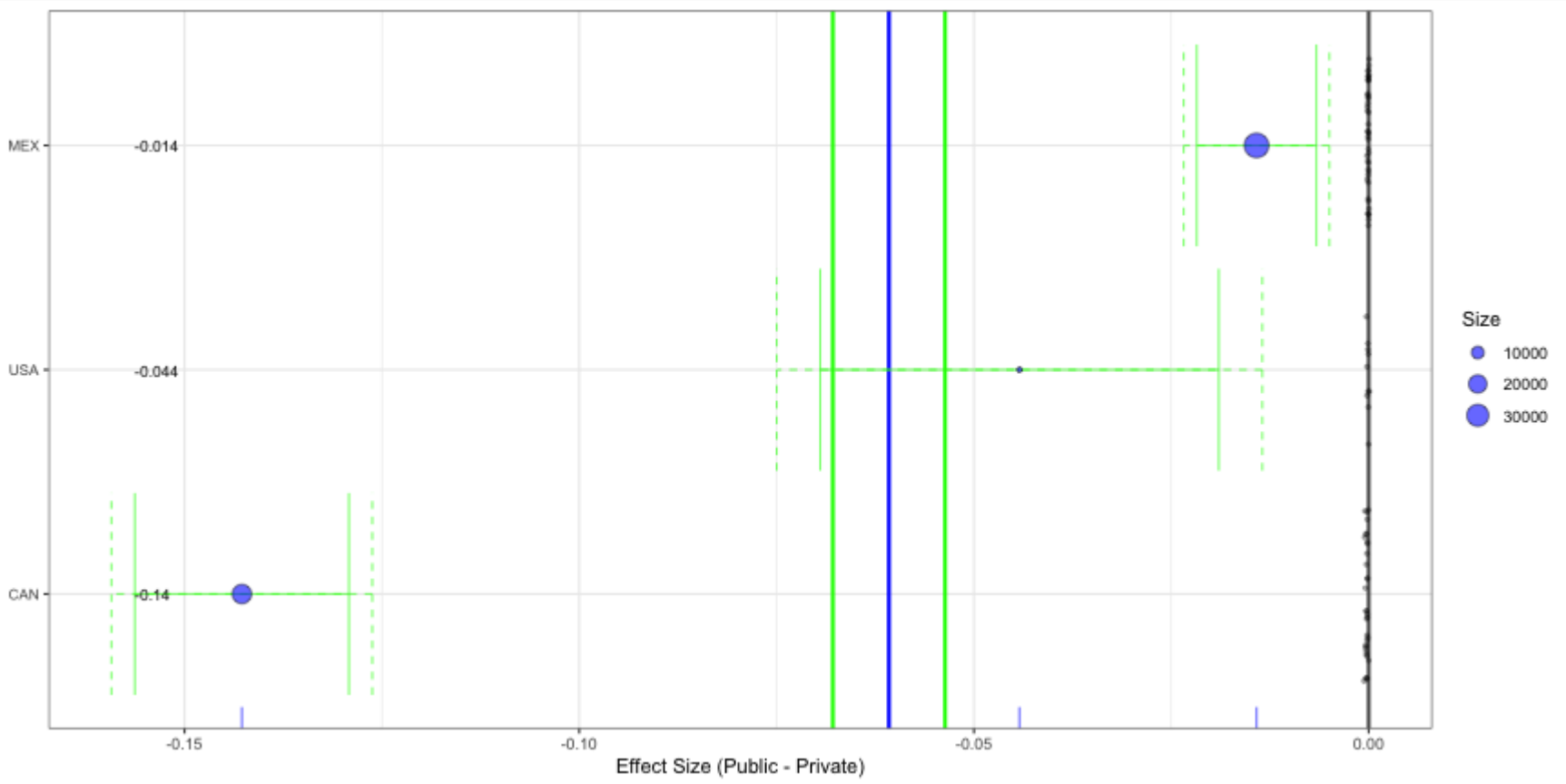
Multilevel PSA Assessment Plot

The multilevel PSA assessment plot is an extension of the `circ.psa` plot in `PSAgraphics` introduced by Helmreich and Pruzek (2009).

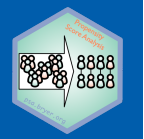
```
plot(results.psa.math)
```



Multilevel PSA Difference Plot

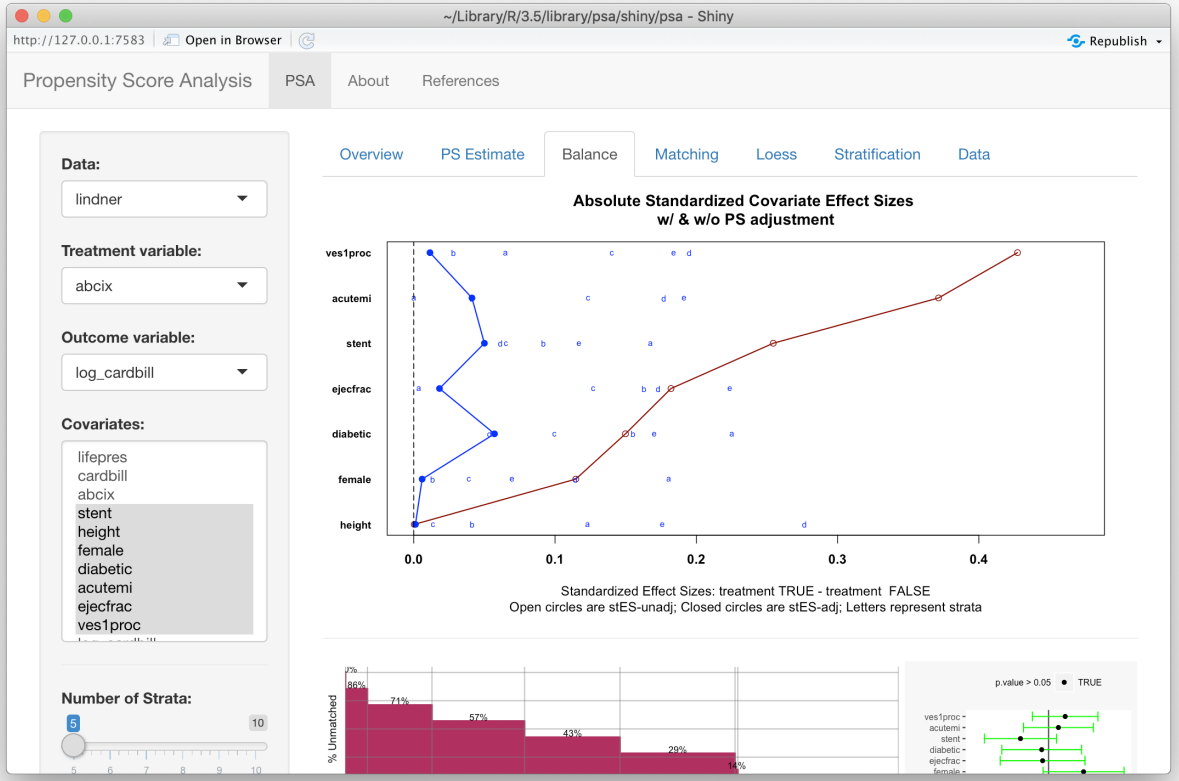


Shiny Application



Shiny Application

```
psa::psa_shiny()
```



Thank You!

jason.bryer@cuny.edu
@jbryer
@jbryer@vis.social
psa.bryer.org
github.com/jbryer/psa