Introduction to Propensity Score Analysis for Institutional Research

NEAIR 2013 Conference

Jason M. Bryer [1]
jason@bryer.org

[1] Excelsior College Albany, NY 12203
http://github.com/jbryer/IntroPSAwithR

November 11, 2013

Agenda

1. Randomized Experiments
2. Defining Propensity Scores
3. Different Methods of PSA
4. Tutoring Example
5. Bootstrapping
6. Matching of Non-Binary Treatments

Agenda

1. Randomized Experiments
2. Defining Propensity Scores
3. Different Methods of PSA
4. Tutoring Example
5. Bootstrapping
6. Matching of Non-Binary Treatments

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 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.

Rubin Causal Model[1]

- 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."

[1] See Rubin, 1974, 1977, 1978, 1980, and Holland, 1986

Agenda

1. Randomized Experiments
2. Defining Propensity Scores
3. Different Methods of PSA
4. Tutoring Example
5. Bootstrapping
6. Matching of Non-Binary Treatments

Propensity Score Analysis

Propensity score analysis (PSA) is a quasi-experimental design used to estimate causal effects in observational studies (i.e. studies where students are not randomized to treatment). PSA is conducted in two phases:

Phase I (Also referred to as the design phase) In phase one we are concerned with adjusting for selection bias. We model treatment placement using observed variables (see next slide). The propensity score is the probability of a student being in the treatment. With estimated propensity scores, clusters or matches are created for phase II.

Phase II With matches or clusters made in phase I, we compare the difference between matches or clusters on the outcome measure of interest.

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:

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

The Average treatment effect on the treated (ATT), is defined as:

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

Agenda

1 Randomized Experiments

2 Defining Propensity Scores

3 Different Methods of PSA

4 Tutoring Example

5 Bootstrapping

6 Matching of Non-Binary Treatments

Propensity score methods

Matching Each treatment unit is paired with a comparison unit based upon the pre-treatment covariates.

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.

Weighting Each observation is weighted by the inverse of the probability of being in that group.

$$\frac{1}{n}\sum_{i=1}^{n}\left(\frac{T_iY_i}{\pi(X_i)}-\frac{(1-T_i)Y_i}{1-\pi(X_i)}\right)$$

Steps for Implementing Matching Methods

Stuart and Rubin (2008) outline the following steps for matching, but the same approach can be used for stratification and weighting as well.

1. Choose the covariates to be used.
2. Define a distance measure (i.e. what constitutes similar).
3. Choose the matching algorithm.
4. Diagnose the matches (or strata) obtained (iterating through steps 2 and 3 as well).
5. Estimate the treatment effect using the matches (or strata) found in step 4.

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 quantiative covariates only).
- Matching with and without replacement.
- One-to-one or one-to-many matching.

Which matching method should you use?

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 quantiative covariates only).
- Matching with and without replacement.
- One-to-one or one-to-many matching.

Which matching method should you use?

Whichever one gives the best balance!

See Rosenbaum (2012), *Testing one hypothesis twice in observational studies*.

Agenda

1. Randomized Experiments
2. Defining Propensity Scores
3. Different Methods of PSA
4. Tutoring Example
5. Bootstrapping
6. Matching of Non-Binary Treatments

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.

Estimating Propensity Scores

```
> data(tutoring, package='TriMatch')
> tutoring$treatbool <- tutoring$treat != 'Control'
```

Estimating Propensity Scores

```
> tutoring.formu <- treatbool ~ Gender + Ethnicity + Military + ESL + EdMother +
    EdFather + Age + Employment + Income + Transfer + GPA
> glm1 <- glm(tutoring.formu, family=binomial, data=tutoring)
> summary(glm1)

Call:
glm(formula = tutoring.formu, family = binomial, data = tutoring)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-1.378  -0.702  -0.573  -0.419   2.343

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)    -1.03269    0.67963   -1.52  0.12864
GenderMALE     -0.86857    0.19051   -4.56  5.1e-06 ***
EthnicityOther -0.14237    0.27028   -0.53  0.59836
EthnicityWhite -0.08953    0.20874   -0.43  0.66800
MilitaryTRUE   -0.13876    0.21994   -0.63  0.52811
ESLTRUE         0.01065    0.31101    0.03  0.97268
EdMother       -0.05577    0.05961   -0.94  0.34949
EdFather        0.01982    0.05192    0.38  0.70270
Age             0.00814    0.00901    0.90  0.36636
Employment     -0.32764    0.11141   -2.94  0.00327 **
Income         -0.02275    0.03629   -0.63  0.53075
Transfer        0.01149    0.00340    3.38  0.00073 ***
GPA             0.08488    0.15693    0.54  0.58862
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1130.6  on 1141  degrees of freedom
Residual deviance: 1074.2  on 1129  degrees of freedom
AIC: 1100

Number of Fisher Scoring iterations: 4
```

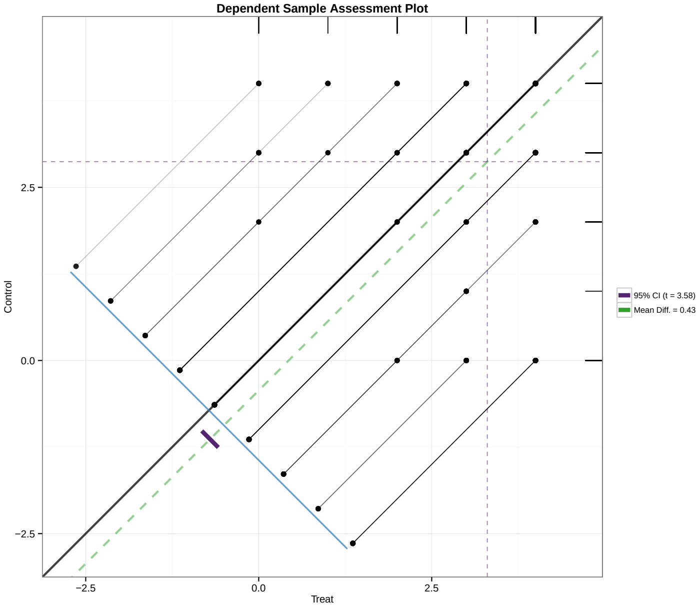
Estimating Propensity Scores

```
> ps <- fitted(glm1)  # Propensity scores
> Y  <- tutoring$Grade  # Dependent variable
> Tr <- tutoring$treatbool # Treatment indicator
> rr <- Match(Y=Y, Tr=Tr, X=ps, M=1, ties=FALSE)
> summary(rr) # The default estimate is ATT here

Estimate...  0.43304
SE.........  0.1207
T-stat.....  3.5877
p.val......  0.00033363

Original number of observations..............  1142
Original number of treated obs...............  224
Matched number of observations...............  224
Matched number of observations  (unweighted).  224
```

Visualizing Results

```
> matches <- data.frame(Treat=tutoring[rr$index.treated,'Grade'],
    Control=tutoring[rr$index.control,'Grade'])
> print(granovagg.ds(matches))
```

```
                              Summary Statistics
n                                         224.00
Treat mean                                  3.30
Control mean                                2.87
mean(D = Treat - Control)                   0.43
SD(D)                                       1.81
Effect Size                                 0.24
r(Treat, Control)                          -0.10
r(Treat + Control, D)                      -0.41
Lower 95% Confidence Interval               0.20
Upper 95% Confidence Interval               0.67
t (D-bar)                                   3.58
df.t                                      223.00
p-value (t-statistic)                       0.00
```

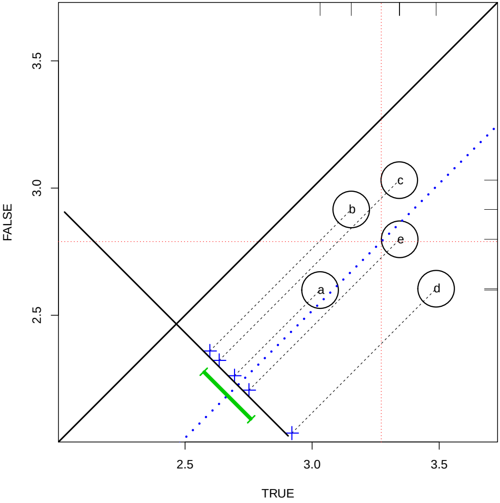

Dependent Sample Assessment Plot
Control
Treat
2.5
0.0
−2.5
−2.5
0.0
2.5
95% CI (t = 3.58)
Mean Diff. = 0.43

Stratification (5 Strata)

```
> strata <- cut(ps, quantile(ps, seq(0, 1, 1/5)), include.lowest=TRUE, labels=letters[1:5])
> circ.psa(tutoring$Grade, tutoring$treatbool, strata, revc=TRUE)
```

```
$summary.strata
  n.FALSE n.TRUE means.FALSE means.TRUE
a     197     32         2.6        3.0
b     202     26         2.9        3.2
c     193     35         3.0        3.3
d     187     41         2.6        3.5
e     139     90         2.8        3.3

$wtd.Mn.TRUE
[1] 3.3

$wtd.Mn.FALSE
[1] 2.8

$ATE
[1] -0.48

$se.wtd
[1] 0.093

$approx.t
[1] -5.2

$df
[1] 1132

$CI.95
[1] -0.67 -0.30
```

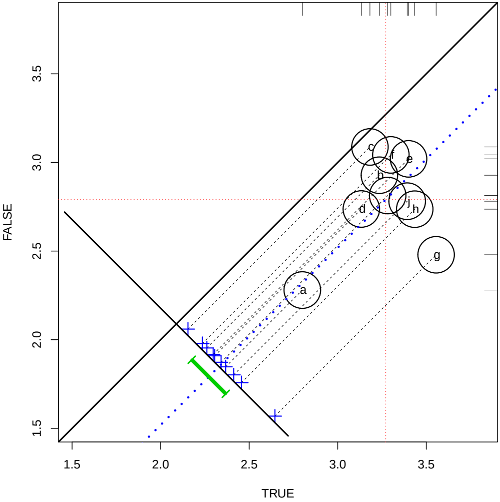
Stratification (5 Strata)



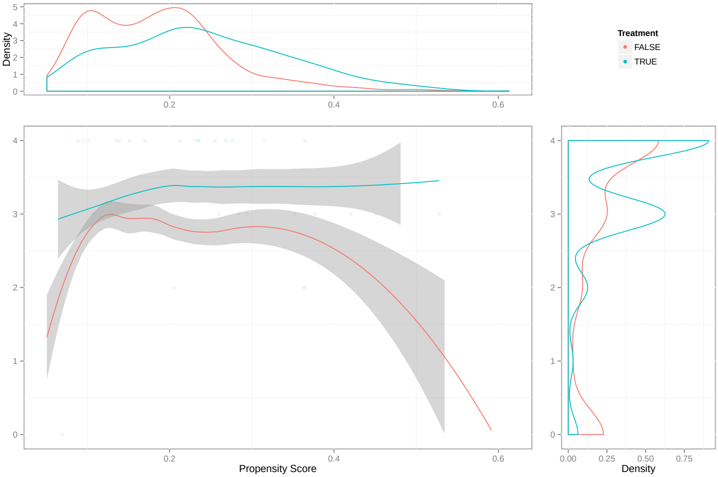
Stratification (10 Strata)

```
> strata10 <- cut(ps, quantile(ps, seq(0, 1, 1/10)), include.lowest=TRUE, labels=letters[1:10])
> circ.psa(tutoring$Grade, tutoring$treatbool, strata10, revc=TRUE)

$summary.strata
  n.FALSE n.TRUE means.FALSE means.TRUE
a     100     15         2.3        2.8
b      97     17         2.9        3.2
c     103     11         3.1        3.2
d      99     15         2.7        3.1
e      99     15         3.0        3.4
f      94     20         3.0        3.3
g      96     18         2.5        3.6
h      91     23         2.7        3.4
i      75     39         2.8        3.3
j      64     51         2.8        3.4

$wtd.Mn.TRUE
[1] 3.3

$wtd.Mn.FALSE
[1] 2.8

$ATE
[1] -0.48

$se.wtd
[1] 0.095

$approx.t
[1] -5.1

$df
[1] 1122

$CI.95
[1] -0.67 -0.29
```

Stratification (10 Strata)

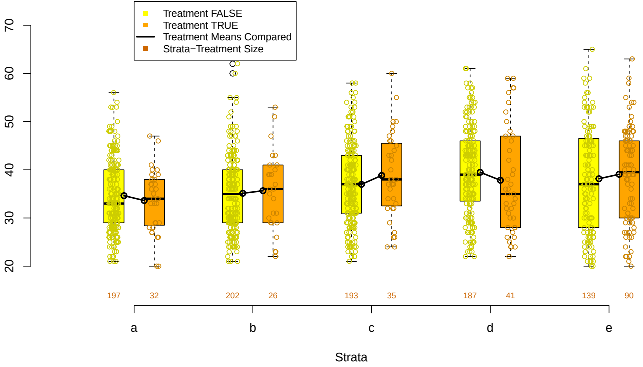


Loess Regression



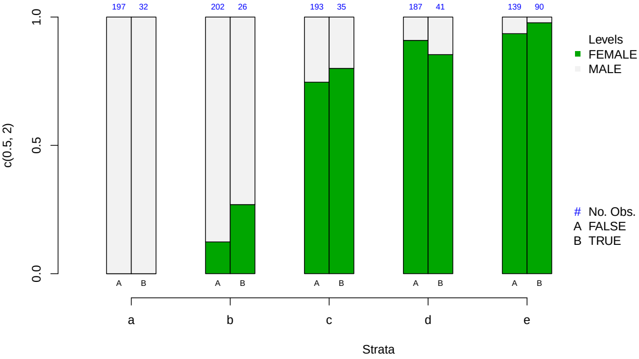
Checking Balance: Continuous Covariates

```
> box.psa(tutoring$Age, tutoring$treatbool, strata, xlab="Strata",
   balance=FALSE)
```



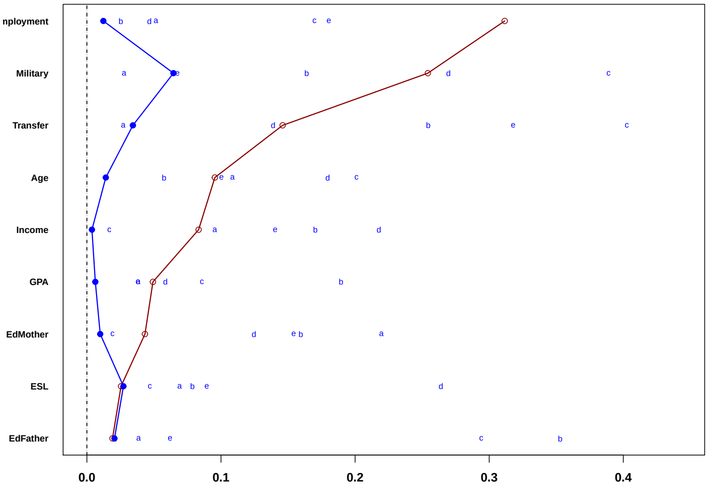
Checking Balance: Categorical Covariates

```
> cat.psa(tutoring$Gender, tutoring$treatbool, strata, xlab='Strata
   balance=FALSE)
```



Checking Balance: Covariate Balance Plot

Absolute Standardized Covariate Effect Sizes
w/ & w/o PS adjustment



Standardized Effect Sizes: treatment TRUE − treatment FALSE
Open circles are stES−unadj; Closed circles are stES−adj; Letters represent strata

Agenda

1 Randomized Experiments

2 Defining Propensity Scores

3 Different Methods of PSA

4 Tutoring Example

5 Bootstrapping

6 Matching of Non-Binary Treatments

Bootstrapping

- Bootstrapping (Efron, 1979) is a resampling technique used to estimate sample statistics.
- It works by drawing many random samples of the same size from all the observed data, generally with replacement.
- With a large enough sample, the distribution of the sample statistic across all bootstrapped samples will be normal (Central Limit Theorem).
- A condifdence intreval around the true sample statistic can be estimated from all the bootstrap samples.

Bootstrapping

- Bootstrapping (Efron, 1979) is a resampling technique used to estimate sample statistics.
- It works by drawing many random samples of the same size from all the observed data, generally with replacement.
- With a large enough sample, the distribution of the sample statistic across all bootstrapped samples will be normal (Central Limit Theorem).
- A condifdence intreval around the true sample statistic can be estimated from all the bootstrap samples.

Bootstrapping can address a number potential issues that arrise in propensity score analysis, namely:

- Shrinkage in the range of propensity scores (i.e. fitted values of logistic regression) due to a large ratio of control-to-treated values.
- Issues in bias reduction to due to outliers.
- Matching issues with tied, or close-to-tied, values.
- Difficulty comparing multiple propensity score methods.

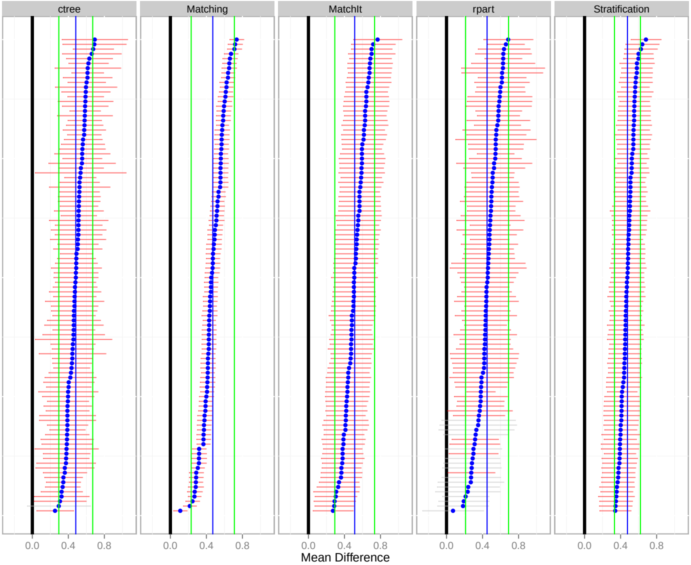
Bootstrapping

```
> table(tutoring$treatbool)

FALSE  TRUE
  918   224

> X <- tutoring[,all.vars(tutoring.formu)]
> X <- X[,-1] # Remove the treatment indicator
> Tr <- tutoring$treatbool
> Y <- tutoring$Grade

> tutoring.boot <- PSAboot(Tr=Tr, Y=Y, X=X, seed=2112,
     control.sample.size=918, control.replace=TRUE,
     treated.sample.size=224, treated.replace=TRUE)
```

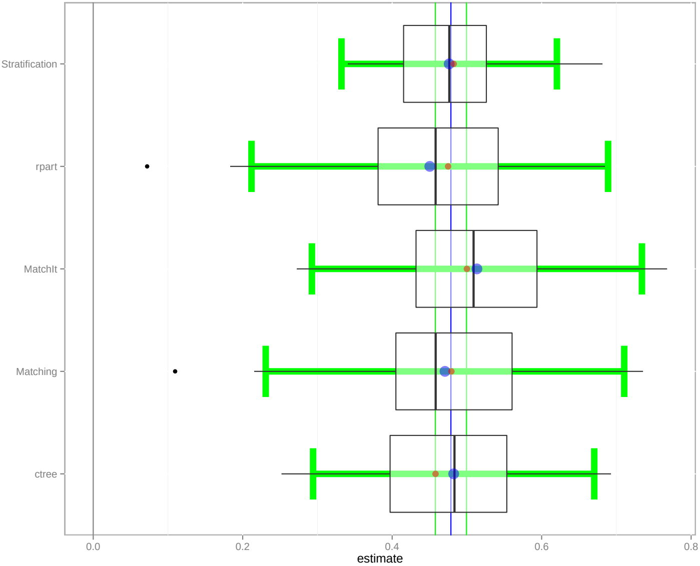
Summary of Bootstrap Samples

```
> summary(tutoring.boot)
Stratification Results:
   Complete estimate = 0.482
   Complete CI = [0.3, 0.665]
   Bootstrap pooled estimate = 0.476
   Bootstrap pooled CI = [0.332, 0.62]
   100% of bootstrap samples have confidence intervals that do not span zero.
      100% positive.
      0% negative.
ctree Results:
   Complete estimate = 0.458
   Complete CI = [0.177, 0.739]
   Bootstrap pooled estimate = 0.482
   Bootstrap pooled CI = [0.294, 0.67]
   99% of bootstrap samples have confidence intervals that do not span zero.
      99% positive.
      0% negative.
rpart Results:
   Complete estimate = 0.475
   Complete CI = [0.165, 0.784]
   Bootstrap pooled estimate = 0.45
   Bootstrap pooled CI = [0.212, 0.689]
   84% of bootstrap samples have confidence intervals that do not span zero.
      84% positive.
      0% negative.
Matching Results:
   Complete estimate = 0.479
   Complete CI = [0.388, 0.571]
   Bootstrap pooled estimate = 0.471
   Bootstrap pooled CI = [0.231, 0.711]
   100% of bootstrap samples have confidence intervals that do not span zero.
      100% positive.
      0% negative.
MatchIt Results:
   Complete estimate = 0.5
   Complete CI = [0.253, 0.747]
   Bootstrap pooled estimate = 0.513
   Bootstrap pooled CI = [0.293, 0.734]
   100% of bootstrap samples have confidence intervals that do not span zero.
      100% positive.
      0% negative.
```

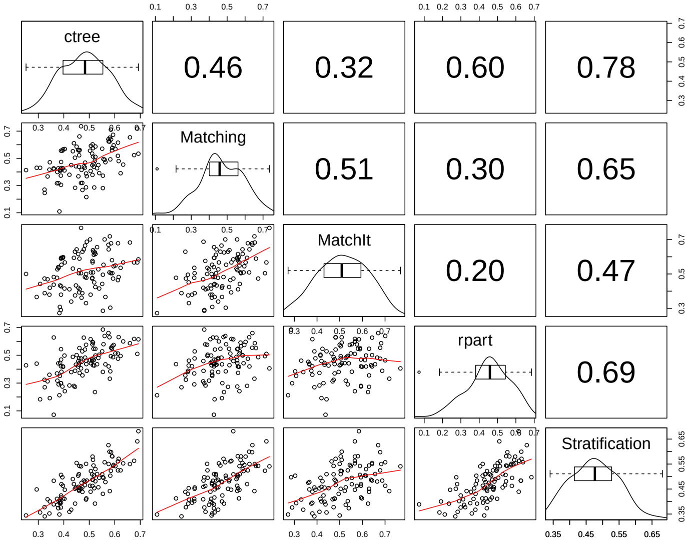
Visualizing Bootstrap Samples



Bootstrap Boxplot



Bootstrap Matrix Plot



Agenda

1. Randomized Experiments
2. Defining Propensity Scores
3. Different Methods of PSA
4. Tutoring Example
5. Bootstrapping
6. Matching of Non-Binary Treatments

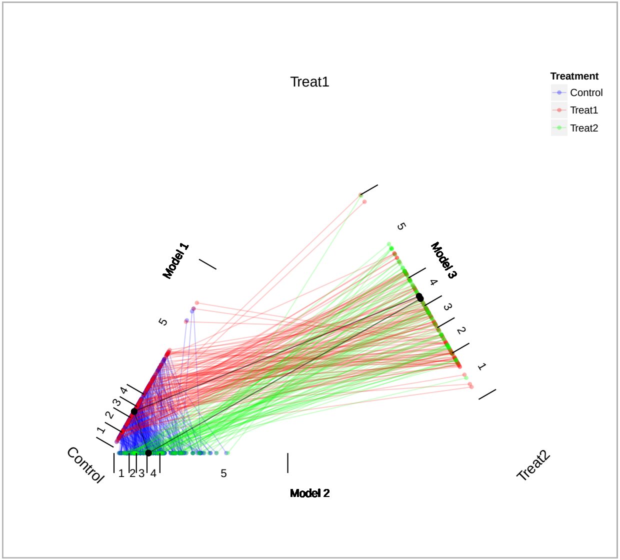
Matching of Non-Binary Treatments

- 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 modesl). 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.

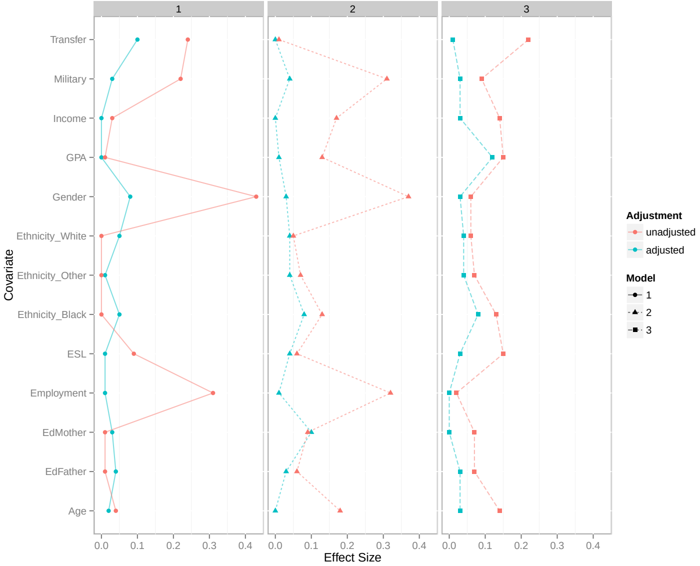
PSA for Non-Binary Treatments

- 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)

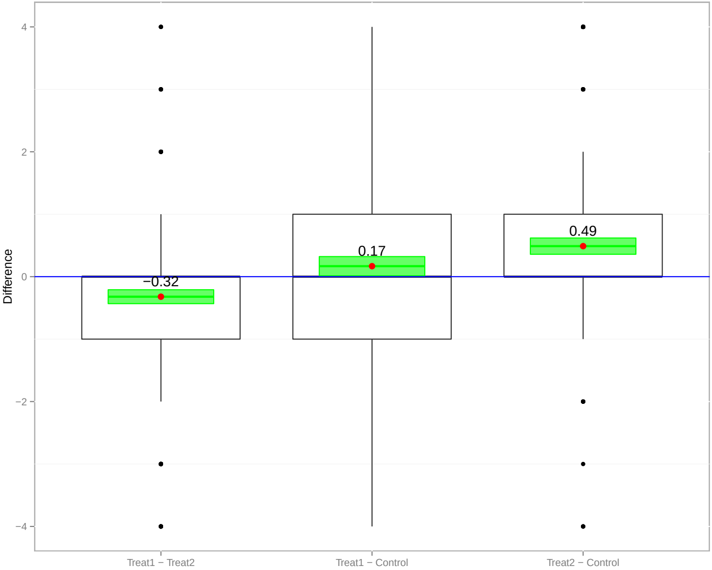
Matching Triplets



Checking Balance



Results



Difference
4
2
0
−2
−4
−0.32
0.17
0.49
Treat1 − Treat2
Treat1 − Control
Treat2 − Control

Thank You

Jason Bryer (jason@bryer.org)
http://www.bryer.org