PSAboot: An R Package for Bootstrapping Propensity Score Analysis

2016 Atlantic Causal Inference Conference Lightning Talk

Jason Bryer, Ph.D.

Excelsior College
`http://jason.bryer.org/PSAboot`
jason@bryer.org

May 26, 2016

Why Bootstrap PSA?

Rosenbaum's (2012) paper, *Testing one hypothesis twice in observational studies*. The `PSAboot` will test the hypothesis M x m times, where M is the number of bootstrap samples and m is the number of different PSA methods.

Many observational studies often have many more control units then treatment units. As the ratio of treatment-to-control increases, the range of propensity scores (i.e. fitted values from a logistic regression) tends to shrink. It may be appropriate to randomly select control units to decrease this ratio. See `http://jason.bryer.org/multilevelPSA/psranges.html` for more information.

Bootstrapping Propensity Score Analysis

The `PSAboot` package/function will:

- Estimate the effects using the full dataset (i.e. the non-bootstrapped analysis).
- Draw *M* stratified bootstrap samples. Stratified on the treatment variable so that each bootstrap sample has the ratio of treatment to control units.
- For each bootstrap sample, estimate the effect for each method (default is five methods).
- Evaluate the balance for each method and bootstrap sample combination.
- Provide an overall pooled estimate across all bootstrap samples.

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.

Bootstrapping PSA

```
> tutoring.boot <- PSAboot(Tr=tutoring$treatbool,
                           Y=tutoring$Grade,
                           X=covs, seed=2112)

100 bootstrap samples using 5 methods.
Bootstrap sample sizes:
   Treated=224 (100%) with replacement.
   Control=918 (100%) with replacement.
```

PSAboot Parameters

Tr numeric (0 or 1) or logical vector of treatment indicators.

Y vector of outcome varaible.

X matrix or data frame of covariates used to estimate the propensity scores.

M number of bootstrap samples to generate (default is 100).

formu formula used for estimating propensity scores. The default is to use all covariates in X.

control.ratio the ratio of control units to sample relative to the treatment units.

control.sample.size the size of each bootstrap sample of control units.

control.replace whether to use replacement when sampling from control units.

treated.sample.size the size of each bootstrap sample of treatment units. The default uses all treatment units for each boostrap sample.

treated.replace whether to use replacement when sampling from treated units.

methods a named vector of functions for each PSA method to use.

seed random seed. Each iteration, i, will use a seed of seed + i.

parallel whether to run the bootstrap samples in parallel.

... other parameters passed to the PSA methods.

Default Methods

Stratification Uses quintiles on the propensity scores estimated using logistic regression.

ctree Estimates strata using conditional inference trees (see `ctree` in `party` package)

rpart Estimates strata using partition trees (see `rpart`)

Matching Matches using the `Matching` package. Propensity scores estimated using logistic regression.

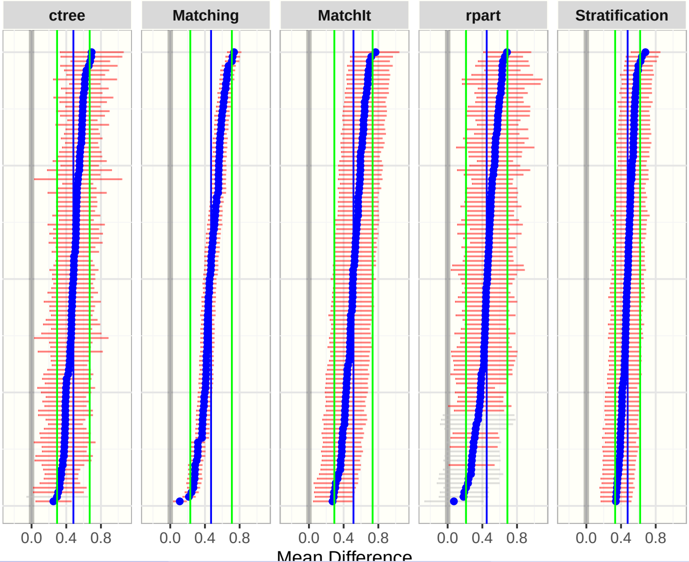
MatchIt Matches using the `MatchIt` package. Propensity scores estimated using logistic regression.

Summary

```
> summary(tutoring.boot)
Stratification Results:
   Complete estimate = 0.482
   Complete CI = [0.3, 0.665]
   Bootstrap pooled estimate = 0.476
   Bootstrap weighted pooled estimate = 0.475
   Bootstrap pooled CI = [0.332, 0.62]
   100% of bootstrap samples have confidence intervals that do not s
      100% positive.
      0% negative.
ctree Results:
   Complete estimate = 0.458
   Complete CI = [0.177, 0.739]
   Bootstrap pooled estimate = 0.482
   Bootstrap weighted pooled estimate = 0.477
   Bootstrap pooled CI = [0.294, 0.67]
   99% of bootstrap samples have confidence intervals that do not sp
      99% positive.
      0% negative.
```

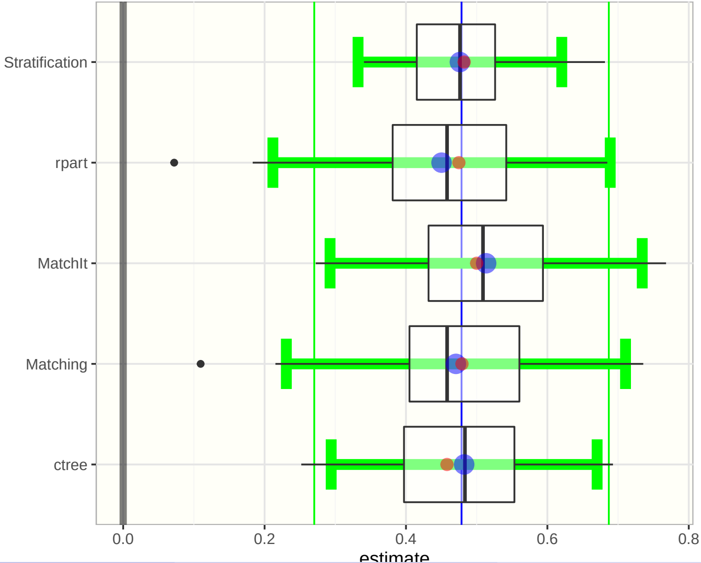
Plotting

```
> plot(tutoring.boot)
```



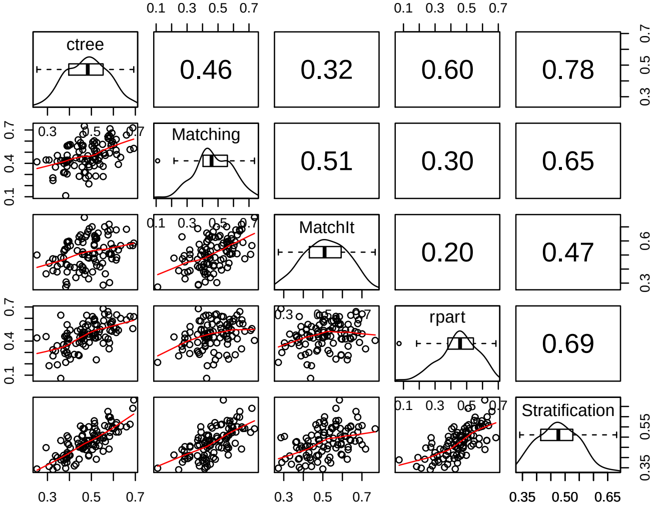
Boxplot

```
> boxplot(tutoring.boot)
```



Matrix Plot

```
> matrixplot(tutoring.boot)
```



Checking Balance

The estimates are only as good as the balance achieved!

```
> tutoring.bal <- balance(tutoring.boot)
> tutoring.bal
```

```
Unadjusted balance: 0.117875835338968
               Complete Bootstrap
Stratification    0.029     0.038
ctree             0.044     0.069
rpart             0.078     0.087
Matching          0.045     0.067
MatchIt           0.051     0.058
```

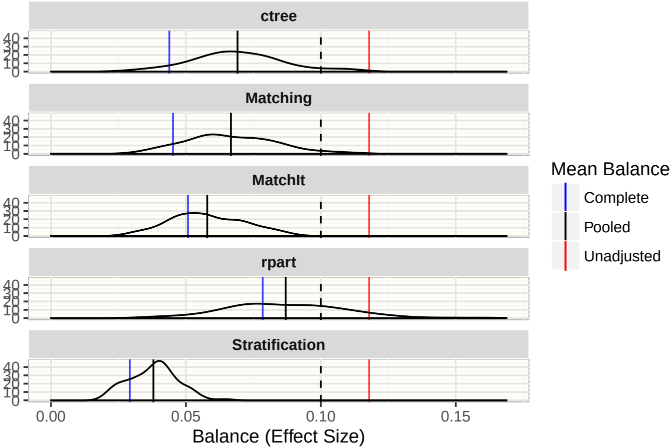
The `balance` function will calculate the standardized effect sizes for each covariate after adjustment. The `pool.fun` allows you to define how the balance statistics are combined. It defaults to `mean`, but other options include `q25`, `q75`, `median` or `max`.

```
> ls(tutoring.bal)
```

```
[1] "balances"   "complete"   "pool.fun"   "pooled"
[5] "unadjusted"
```

Checking Balance: Density Plots

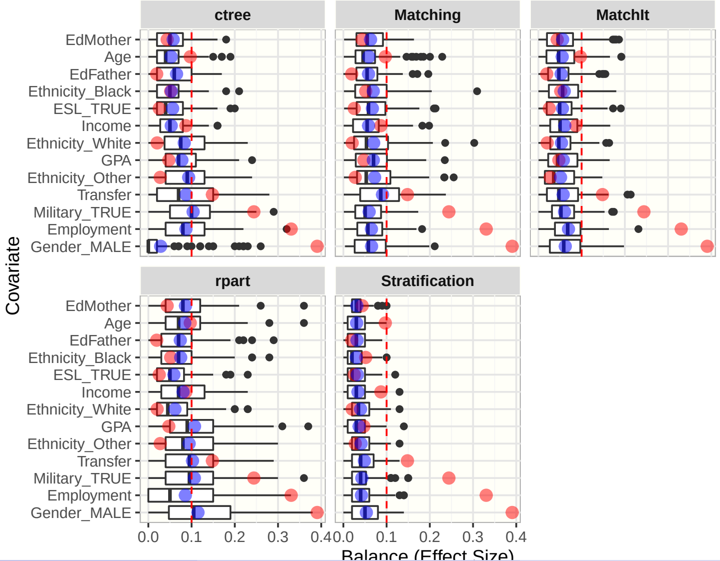
```
> plot(tutoring.bal) + geom_vline(xintercept=.1, linetype=2)
```



Red line is unadjusted balance; Blue line is the non-bootstrap balance; Black line is the pooled bootstrap balance.

Checking Balance: Boxplots

```
> boxplot(tutoring.bal) + geom_hline(yintercept=.1, color='red', lin
```



Extending PSAboot for Other Methods

Define a function with the following parameters: Tr (vector of treatment indicators), Y (vector outcome measure), X (data frame of covariates), X.trans (numeric matrix with non-numeric variables dummy coded), formu (the formula used for estimating propensity scores, ... (other parameters passed from the user).

```
> boot.matching.1to3 <- function(Tr, Y, X, X.trans, formu, ...) {
          return(boot.matching(Tr=Tr, Y=Y, X=X,
                 X.trans=X.trans,
                 formu=formu, M=3, ...))
  }
```

Extending PSAboot for Other Methods

Define a function with the following parameters: Tr (vector of treatment indicators), Y (vector outcome measure), X (data frame of covariates), X.trans (numeric matrix with non-numeric variables dummy coded), formu (the formula used for estimating propensity scores, ... (other parameters passed from the user).

```
> boot.matching.1to3 <- function(Tr, Y, X, X.trans, formu, ...) {
          return(boot.matching(Tr=Tr, Y=Y, X=X,
                 X.trans=X.trans,
                 formu=formu, M=3, ...))
  }
```

```
> tutoring.boot <- PSAboot(Tr=tutoring$treatbool,
                           Y=tutoring$Grade,
                           X=covs,
              methods=c("Matcing-1-to-3"=boot.matching.1to3,
                        getPSAbootMethods()))
```

The getPSAbootMethods() function returns a vector of the five default functions. Note that the name of each element in methods will be the name used in the figures.

Getting More Information

- Package Vignette
  `vignette("PSAboot")`
- Lalonde Demo
  `demo("PSAbootLalonde")`
- Tutoring Demo
  `demo("PSAbootTutoring")`
- Programme of International Student Assessment Demo
  `demo("PSAbootPISA")`

Thank You

Jason Bryer, Ph.D. (jason@bryer.org)
http://jason.bryer.org/PSAboot
http://github.com/jbryer/PSAboot