7. optmatch package, fullmatch, lalonde.
MatchIt uses the optmatch package fullmatch command for its "full" option, as used in the class example. Using the raw optmatch (without the matchit wrapper) allows additional specifications and controlls for the full or optimal matching.
For lalonde data try out optmatch fullmatching and compare results for subclasses and balance with the class example using optmatch through MatchIt.

Solution for Review Question 7

Week 2-- Matching Methods Part 2 (implementation); Potential Outcomes and Study Design

Lecture Topics Lecture 2 slide deck (companion audio)

```
1. Basic tools of multivariate matching (DOS, Secs 8.1-8.4)
2. Potential outcomes framework (DOS 2.2)
3. Fisher's sharp null; permutation test (DOS 2.3)
4. Various practical issues in matching (DOS, Chap 9)
```

Text Readings
Rosenbaum DOS: Chapter 2 (plus week1 items)
Additional Resources
From Donald B. Rubin
First section of Basic Concepts of Statistical Inference for Causal Effects in Experiments and Observational Studies Similar material Chaps 1 and 2 Causal Inference in Statistics, Social and Biomedical Sciences: An Introduction, Guido Imbens and Don Rubin linked on main page.

Computing Corner: Extended Data Analysis Examples
Lindner data, Percutaneous Coronary Intervention with 'evidence based medicine'.
Percutaneous coronary intervention (PCI), commonly known as coronary angioplasty or simply angioplasty, is a non-surgical procedure used to treat the stenotic (narrowed) coronary arteries of the heart found in coronary heart disease.
Lindner data in package `PSAgraphics` Use of Lindner data in Vignette JSS PSAgraphics: An R Package to Support Propensity Score Analysis *Journal of Statistical Software* February 2009, Volume 29, Issue 6. http://www.jstatsoft.org/

cc2 pdf slides Lindner example 2021 audio companion
Week 2 handout Rogosa R-session
Additional R-session Lindner fullmatch in optmatch and cobalt

Week 2 Review Questions
From Computing Corner
1. The JSS vignette for PSAgraphics (linked week 2 Computing Corner) does subclassification matching for Lindner data. Repeat their subclassification analyses and try out their balance displays and tests. They have some specialized functions. Compare with our basic approach.

Lindner data package `PSAgraphics` Vignette JSS outcome analysis, Rogosa session

2. The Week 2 presentation showed an alternative propensity score analysis -- analysis of covariance with propensity score as covariate. A rough analogy is to ancova vs blocking (where blocking is our subclassification, say quintiles). Try out the basic (here logistic regression) ancova approach for the lifepres dichotomous outcome

Solution for Review Question 2

From Lecture
3. Modify Fisher's Sharp Null to reflect the null hypothesis that the treatment adds five units to the outcome under control. Build a small simulation (e.g., 10 observations) and construct a table that summarizes the potential outcomes. Randomize using a fair coin flip to assign treatment or control for each observational unit. Use the permutation test to assess your data set using (i) Fisher's Sharp Null and (ii) the null hypothesis that the treatment adds five units to the outcome under control.

Solution for Review Question 3

4. Building off of RQ#3 above, sort your observations so they are in ascending order based on the outcome under control. Randomize two at a time: one fair coin flip now assigns either the first or second observation to treatment (and the other to control). A second fair coin flip assigns either the third or the fourth observation to treatment (and the other to control). This continues so on and so forth. Use the appropriate permutation test to assess your data set using (i) Fisher's Sharp Null and (ii) the null hypothesis that the treatment adds five units to the outcome under control. Contrast the results here with the results from RQ#3.

Solution for Review Question 4

From Computing Corner
5. Pair matching--nuclear plants data. See also week8,Stat209. Another (small) canonical matching example for optmatch expositions is the nuclear plants data from Cox and Snell text.
Data set is `nuclearplants` fromn `optmatch` optmatch manual Ben Hansen (local hero) exposition of nuclearplants example in R News Oct 2007
Additional exercises (checking balance) using the nuclearplants data from Mark Fredrickson here
Data cleaning gives 7 "treatment" and resevoir of 19 controls. Try out 1:2 optimal pair matching using MatchIt (see also stat209 exs) and compare with pairmatch in optmatch plus balance diagnostics.

nuclearplants using Matchit (Stat209 handout) optmatch for Review Question 5

Week 3-- Full matching, Inclusion and Exclusion, and Defining Treatment Effects

Lecture Topics Lecture 3 slide deck (companion audio)
optmatch example from lecture

```
1. Finish up: Basic tools of multivariate matching (DOS, Secs 8.1-8.4)
2. Various practical issues in matching (DOS, Chap 9)
3. Inverse probability weighting ( Robins & Hernan, Chap 2.4) -
```

```
# redo of week2 Lindner example using optmatch etc
# Rogosa R-session, warts and all

R version 3.5.3 (2019-03-11) -- "Great Truth"

> library(MatchIt)
> library(optmatch)
Loading required package: survival
The optmatch package has an academic license. Enter relaxinfo() for more information.
> library(cobalt)

> install.packages("PSAgraphics")

> library(PSAgraphics)
Loading required package: rpart

> data(lindner)

> head(lindner)
  lifepres cardbill abcix stent height female diabetic acutemi ejecfrac ves1proc
1      0.0    14301     1     0    163      1        1       0       56        1
2     11.6     3563     1     0    168      0        0       0       56        1
3     11.6     4694     1     0    188      0        0       0       50        1
4     11.6     7366     1     0    175      0        1       0       50        1
5     11.6     8247     1     0    168      1        0       0       55        1
6     11.6     8319     1     0    178      0        0       0       50        1
> covs = subset(lindner, select = -c(lifepres, cardbill, abcix ))
> head(covs)
  stent height female diabetic acutemi ejecfrac ves1proc
1     0    163      1        1       0       56        1
2     0    168      0        0       0       56        1
3     0    188      0        0       0       50        1
4     0    175      0        1       0       50        1
5     0    168      1        0       0       55        1
6     0    178      0        0       0       50        1

#start the optmatch process
> pfit = glm(f.build("abcix", covs), data = lindner, family = "binomial")
>  lindner$p.score = pfit$fitted.values #get the propensity score
>  boxplot(lindner$p.score ~ lindner$abcix) # propensity score
>  fm2 = fullmatch(abcix ~ p.score, data = lindner)
>  bal.tab(fm2, formula = f.build("abcix", covs), data = lindner)
Call
 fullmatch(x = abcix ~ p.score, data = lindner)

Balance Measures
            Type Diff.Adj
stent     Binary  -0.0185
height   Contin.  -0.0099
female    Binary   0.0347
diabetic  Binary  -0.0136
acutemi   Binary   0.0124
ejecfrac Contin.  -0.0570
ves1proc Contin.  -0.0020

Sample sizes
        Control Treated
All         298     698
Matched     298     698
> # gives you spectacular balance
> #optmatch balance much better on stent female diabetic

>  love.plot(bal.tab(fm2, formula = f.build("abcix", covs), data = lindner), threshold = .1)
```

I didn't include this nice plot in pdf

```
> # outcome analysis log(cardbill)
> # look at subclasses
> summary(fm2)
Structure of matched sets:
5+:1  4:1  3:1  2:1  1:1  1:2  1:3  1:4 1:5+
  38   14   34   40  113   14    4    2    2
Effective Sample Size:  337.8
```

```
(equivalent number of matched pairs).

> stratumStructure(fm2)
17:1 16:1 15:1 13:1 12:1 11:1 10:1  9:1  8:1  7:1  6:1  5:1  4:1  3:1  2:1  1:1  1:2  1:3  1:4  1:5
   3    2    1    1    2    1    1    2    4    3    8   10   14   34   40  113   14    4    2    1
> # looks like 261 subclasses compared to 267 from MatchIt(full)
> # here we grab a factor giving us the subclass info
> # I call the augmented lindner dataset "Lmatched"
>
>  Lmatched = cbind(lindner, matches = fm2)
> head(Lmatched)
  lifepres cardbill abcix stent height female diabetic acutemi ejecfrac ves1proc   p.score matches
1      0.0    14301     1     0    163      1        1       0       56        1 0.4079170     1.1
2     11.6     3563     1     0    168      0        0       0       56        1 0.5784602   1.112
3     11.6     4694     1     0    188      0        0       0       50        1 0.5244469   1.223
4     11.6     7366     1     0    175      0        1       0       50        1 0.4727311   1.334
5     11.6     8247     1     0    168      1        0       0       55        1 0.4930466   1.445
6     11.6     8319     1     0    178      0        0       0       50        1 0.5625524   1.556
> str(Lmatched)
'data.frame':   996 obs. of  12 variables:
 $ lifepres: num  0 11.6 11.6 11.6 11.6 11.6 11.6 11.6 11.6 11.6 ...
 $ cardbill: int  14301 3563 4694 7366 8247 8319 8410 8517 8763 8823 ...
 $ abcix   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ stent   : int  0 0 0 0 0 0 0 0 0 0 ...
 $ height  : int  163 168 188 175 168 178 185 173 152 180 ...
 $ female  : int  1 0 0 0 1 0 0 1 1 0 ...
 $ diabetic: int  1 0 0 1 0 0 0 0 0 0 ...
 $ acutemi : int  0 0 0 0 0 0 0 0 0 0 ...
 $ ejecfrac: int  56 56 50 50 55 50 58 30 60 60 ...
 $ ves1proc: int  1 1 1 1 1 1 1 1 1 1 ...
 $ p.score : num  0.408 0.578 0.524 0.473 0.493 ...
 $ matches : Factor w/ 261 levels "1.1","1.10","1.101",..: 1 8 84 142 182 213 230 236 241 70 ...
  ..- attr(*, "exceedances")= Named num 0.565
  .. ..- attr(*, "names")= chr "1"
  ..- attr(*, "call")= language fullmatch(x = abcix ~ p.score, data = lindner)
  ..- attr(*, "contrast.group")= logi  TRUE TRUE TRUE TRUE TRUE TRUE ...
  ..- attr(*, "subproblem")= Factor w/ 1 level "1": 1 1 1 1 1 1 1 1 1 1 ...
  .. ..- attr(*, "names")= chr  "1" "2" "3" "4" ...
  ..- attr(*, "min.controls")= num 0
  ..- attr(*, "max.controls")= num Inf
  ..- attr(*, "omit.fraction")= num NA
  ..- attr(*, "hashed.distance")= chr "3fb101a69d79c551c6755deb7ec872dc"
> length(table(Lmatched$matches))
[1] 261
> table(Lmatched$matches) # lots of small subclasses, causes me lmer problems

  1.1  1.10 1.101 1.104 1.106 1.107 1.110 1.112 1.113 1.119  1.12 1.120 1.122 1.124 1.125 1.126 1.129
    2     2     2     2     4     2     2     8     3     2     4    11     2     6     2     2     3
1.136  1.14 1.143 1.144 1.147 1.148 1.149  1.15 1.150 1.152 1.153 1.154 1.155 1.157 1.158  1.16 1.161
    3     4     2     2     3     4     2     2     2     5     3     3     4     4     7     2     3
1.167 1.168 1.171 1.172 1.173 1.174 1.175 1.176 1.178 1.179  1.18 1.180 1.181 1.182 1.183 1.184 1.185
    2     2     2     3     2     5     2     5     2     3     4     4     4     5     4     7     2
1.191 1.192 1.193 1.194 1.196 1.197   1.2 1.200 1.201 1.204 1.206 1.209  1.21 1.212 1.213 1.215 1.216
    4     4     2     2     2     2     2     2     2     4     3     3     2     5     5     2     2
1.229 1.231 1.233 1.234 1.235 1.238  1.24 1.244 1.246  1.25 1.251 1.253 1.254 1.257  1.26 1.261 1.263
    3     2     2     2     2     2     6     3     2     3     2     2     2     2     2     2     3
1.275 1.279  1.28 1.285 1.287  1.29 1.292 1.298 1.299   1.3  1.30 1.300 1.301 1.302 1.303 1.304 1.305
    3     4     3     4     2     2     4     4    18     3     3     5     2     2     5     2     6
1.310 1.311 1.312 1.314 1.315 1.317 1.318 1.319  1.32 1.323 1.327 1.328 1.329  1.33 1.330 1.334 1.335
    3     4     3    17     6     7     8     4     4     6     6     2     2     2     3     3     2
1.343  1.35 1.351 1.353 1.354 1.357  1.36 1.361 1.363 1.364 1.368  1.37 1.377 1.379  1.38 1.385  1.39
    2     4     7     9     2     2     2     7     3     2     2     2     2     2     3     4     5
1.401 1.408  1.41 1.410 1.412 1.415 1.418 1.422 1.423  1.43 1.431 1.437  1.44 1.445 1.457  1.46 1.468
    3     2     4     4     2     3     3     2     4     5     2     3     2     5     2     3     3
1.474 1.477 1.478  1.48 1.480 1.483 1.485 1.486 1.490 1.495   1.5  1.50 1.503 1.512 1.517 1.519  1.52
   10     5     4     2    13    18    16    12     2    10     2     2    14     4     9     3     3
1.545  1.55 1.556 1.557 1.560 1.562  1.57 1.579  1.58  1.59 1.601  1.61 1.623  1.63 1.634  1.64  1.65
    2     4     2     3     4     7     3     2     2     3     2     3     2     3     3     2     2
1.669  1.67 1.673 1.676 1.677 1.679  1.68 1.686 1.687 1.688 1.690 1.692 1.695 1.697   1.7  1.70  1.73
    2     4     2     7     2     2     2     3     4     4     2     3     2     2     2     2     2
  1.8  1.81  1.83  1.84  1.88  1.89   1.9  1.90  1.94
```

```
    2     3     4     2     2     2     2     4     2
> library(lme4)
# lmer doesn't like me today
> optmatch_lmer = lmer(log(cardbill) ~ abcix +(abcix|matches), data = Lmatched)
boundary (singular) fit: see ?isSingular
> ?isSingular
starting httpd help server ... done
> isSingular(optmatch_lmer, tol = 1e-03)
[1] TRUE
> isSingular(optmatch_lmer, tol = 1e-02)
[1] TRUE
> isSingular(optmatch_lmer, tol = 1e-06)
[1] FALSE

# appease lmer with a control statement
> optmatch_lmer = lmer(log(cardbill) ~ abcix +(abcix|matches), control = lmerControl(check.conv.singu
> summary(optmatch_lmer)
Linear mixed model fit by REML ['lmerMod']
Formula: log(cardbill) ~ abcix + (abcix | matches)
   Data: Lmatched
Control: lmerControl(check.conv.singular = .makeCC(action = "message",      tol = 1e-06))

REML criterion at convergence: 1278.1

Scaled residuals:
    Min      1Q  Median      3Q     Max
-3.2461 -0.6354 -0.2558  0.4198  4.3152

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 matches  (Intercept) 0.08324  0.2885
          abcix       0.07030  0.2651   -1.00
 Residual             0.18820  0.4338
Number of obs: 996, groups:  matches, 261

Fixed effects:
            Estimate Std. Error t value
(Intercept)  9.40325    0.03132 300.201
abcix        0.17531    0.03471   5.051

Correlation of Fixed Effects:
      (Intr)
abcix -0.879
> # about the same as Matchit lmer

> confint(optmatch_lmer, oldNames = FALSE)
Computing profile confidence intervals ...
                                    2.5 %    97.5 %
sd_(Intercept)|matches         0.1914737 0.3702391
cor_abcix.(Intercept)|matches -1.0000000 1.0000000
sd_abcix|matches               0.1595840 0.3665216
sigma                          0.4093783 0.4568357
(Intercept)                    9.3418121 9.4649082
abcix                          0.1070127 0.2441981
There were 12 warnings (use warnings() to see them)
> # lmer is appeased, about same result; lmer from Matchit may well now have same issue
```

ascii R-session has complete code

Text Readings
Rosenbaum DOS: Chapters 8 and 9
Additional Resources
Smoking study (Prochaska et al 2016)
Dealing with limited overlap in estimation of average treatment effects (Crump et al 2009) (or see http://public.econ.duke.edu/~vjh3/working_papers/overlap.pdf)
Defining the Study Population for an Observational Study to Ensure Sufficient Overlap: A Tree Approach (Traskin & Small 2011)
CONSORT Statement (randomized trials)
STROBE Statement (observational studies)

Computing Corner resumes Week 4 with IPW methods

Week 3 Review Questions:
From Lecture
1. In this class we've shown you a couple of tools to assess the adequacy of a matched set - for example: Love plots, balance tables, standardized mean differences, and histogram plots of fitted propensity scores (or covariates). Why haven't we shown you a statistical test? That's weird, right? A ton of researchers fall for this, failing to see why assessing balance using a hypothesis test in an observational study is problematic. There are a couple of valid critiques; try articulating at least one such critique. (Hint: think about how we calculate the SMD vs a standard error.) Once you've given it a go, check out Section 6.6 of this paper (great paper!) for a couple of solutions to this question.

2. In section 6.7 of that same paper, the authors say their preferred tool for assessing balance is an empirical QQ plot. What's a QQ plot? Compare and contrast the use of QQ plots and a balance table. Neither of these tools in dominate, so what are the benefits and drawbacks to each?

Solution for Review Question 2

Week 4-- Models for Observational Studies

Lecture Topics Lecture 4 slide deck (companion audio)

`First model for observational studies (DOS, Sections 15.1 and 15.4; 3.1-3.3)`

Computing Corner: Extended Data Analysis Examples
Alternative propensity score analyses. Propensity score weighting: Inverse Probability of Treatment Weighting (IPTW). Treatment effect estimation without matching.
Primary sources:
Review paper: Moving towards best practice when using inverse probability of treatment weighting (IPTW) using the propensity score to estimate causal treatment effectsin observational studies Peter C. Austin and Elizabeth A. Stuart, Statistics in Medicine Statist. Med.2015,34 3661-36793661
A thorough R exposition using the Lalonde data A Practical Guide for Using Propensity Score Weighting in R Practical Assessment, Research & Evaluation, v20 n13 Jun 2015.

pdf slides cc4 2021 audio companion Rogosa R-session

Additional Resources:
`package bcaboot` intro vignette paper: The automatic construction of bootstrap confidence intervals
A Guide to Using WeightIt for Estimating Balancing Weights Noah Greifer
Propensity score techniques and the assessment of measured covariate balance to test causal associations in psychological research Valerie S. Harder, M.H.S., Ph.D., Elizabeth A. Stuart, Ph.D., and James C.Anthony, Ph.D. .R file (readable code showing matchit fullmatch and IPW) for paper Also Cox Regression, comparison with full matching (Elizabeth Stuart)

Week 4 Review Questions
From Computing Corner
1. Try out the ATE IPTW analysis (done in week4 computing corner) for the dichotomous outcome lifepres in the Lindner data. Compare with full matching results shown in class.

Solution for Review Question 1

2. Try an ATT IPTW analysis for log(cardbill) outcome in the Lindner data.

Solution for Review Question 2

From Lecture
3. The Wilcoxon signed rank test takes as its input a fixed number, designate this number *I*, of matched pairs. The Wilcoxon signed rank test is a permutation test with a specific test statistic. Let's explore the behavior of its statistic compared to the behavior of the average of the within-pair differences. You can use the sample code provided to simulate (i.e., simulation 1 here). Consider playing around with the sd in the data generating functions to see the impact in the histograms.
Question: what happens when we introduce **one** really 'weird' data point in our matched sets? Compare what happens to the distributions for mean(y_t - y_c |matched pairs) vs the Wilcoxon rank sign test. The solution is in the comments in simulation 3 in the link above.

Week 5-- Randomized Experiments and Design Sensitivity

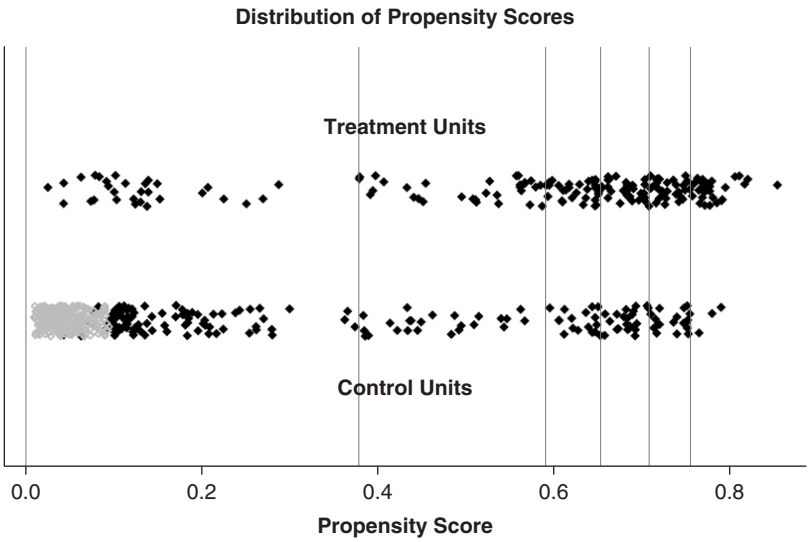


Figure 11.3 One-to-one nearest neighbor matching on the propensity score followed by subclassification. Black units were matched; gray units were unmatched. Subclasses indicated by vertical lines.

method, estimating the effect of SAT coaching, illustrating that, although the original treated and control groups had propensity score differences of 1.1 standard deviations, the matched sets from full matching differed by less than 2% of a standard deviation. To achieve efficiency gains, Hansen (2004) also describes a variation of full matching that restricts the ratio of the number of treated units to the number of control units in each matched set, a method also applied in Stuart and Green (in press).

The output from full matching is illustrated using the NSW data in Figure 11.4. Because it is not feasible to show the individual matched sets (in these data, 103 matched sets were created), the units are represented by their relative weights. All treated units receive a weight of 1 (and thus the symbols are all the same size). Control units in matched sets with many control units and few treated units receive small weight (e.g., the units with propensity scores close to 0), whereas control units in matched sets with few control units and many treated units (e.g., the units with propensity scores close to 0.8) receive large weight. The weighted treated and control group covariate distributions look very similar. As in simple subclassification, all control units within a matched set receive equal weight. However, because there are many more matched sets than with simple subclassification, the variation in the weights is much larger across matched sets.

Because subclassification and full matching place all available units into one of the subclasses, these methods may have particular appeal for researchers who are reluctant to discard some of the control units. However, these methods are not relevant for situations where the matching is being used to select units for follow-up or for situations where some units have essentially zero probability of receiving the other treatment.

Weighting Adjustments

Another method that uses all units is weighting, where observations are weighted by their inverse propensity score (Czajka, Hirabayashi, Little, & Rubin, 1992; Lunceford & Davidian, 2004; McCaffrey et al., 2004). Weighting can also be thought of as the limit of subclassification as the number of observations and the number of subclasses go to infinity. Weighting methods are based on Horvitz-Thompson estimation (Horvitz & Thompson, 1952), used frequently in sample surveys. A drawback of weighting adjustments is that, as with Horvitz-Thompson estimation, the sampling variance of resulting weighted estimators can be very large if the weights are extreme (if the propensity scores are close to 0 or 1). Thus, the subclassification or full matching approaches, which also use all units, may be more appealing because the resulting weights are less variable.

2 Statistical methods

In an observational study of the effect of treatment on outcomes, the propensity score is the probability of receiving the treatment of interest conditional on measured baseline covariates: $e = (Z = 1|X)$, where X denotes the vector of measured baseline covariates.[4] The propensity score is often estimated using a logistic regression model, with the propensity scores being the predicted probabilities generated by that model.

In constructing a full matching stratification, each subject is assigned to a matched set consisting of either one treated subject and at least one control subject or one control subject and at least one treated subject. Weights can be derived from the stratification imposed by the full matching. One set of weights permits estimation of the ATT, while a different set of weights permits estimation of the ATE. Weights that permit estimation of the ATT can be constructed as follows: treated subjects are assigned a weight of one, while each control subject has a weight proportional to the number of treated subjects in its matched set divided by the number of controls in the matched set.[21,22] The control group weights are scaled such that the sum of the control weights across all the matched sets is equal to the number of uniquely matched control subjects. Weights that permit estimation of the ATE can be constructed as follows: treated subjects are assigned a weight equal to the marginal probability of receiving the treatment in the overall sample multiplied by the number of subjects in the given subclass or stratum divided by the number of treated subjects in that subclass. Similarly, control subjects are assigned a weight equal to the marginal probability of receiving the control treatment in the overall sample multiplied by the number of subjects in the given subclass divided by the number of control subjects in that subclass.[23] Thus, if m and j denote the number of treated and control subjects in a given stratum, and q denotes the marginal probability of treatment in the overall sample, then the weights for treated and control subjects in the given stratum are $\frac{q(m+j)}{m}$ and $\frac{(1-q)(m+j)}{j}$, respectively.

When using IPTW to estimate the ATE, weights are computed that denote the probability of receiving the actual treatment that was received. If e denotes the estimated propensity score, then the original sample is weighted by the following weights: $\frac{Z}{e} + \frac{(1-Z)}{1-e}$ (i.e. treated subjects are assigned a weight equal to the reciprocal of the propensity score, while control subjects are assigned a weight equal to the reciprocal of one minus the propensity score).

or ifelse

Using either approach (full matching or IPTW) with survival outcomes, the hazard of the occurrence of the event of interest is regressed on an indicator variable denoting treatment status using a Cox proportional hazards model that incorporates the appropriate set of weights and that employs a robust variance estimator to account for the weights being estimated, rather than known with certainty.[19,24][–26] Furthermore, when using full matching, the clustering of subjects within strata was taken into account when estimating standard errors.

3 The relative performance of full matching and IPTW for estimating marginal hazard ratios with a correctly specified propensity score model

We conducted a series of Monte Carlo simulations to examine the relative performance of full matching and IPTW when estimating the effect of treatment on survival outcomes when the target estimand is the ATE. In this section, we restrict our attention to scenarios in which the propensity score model has been correctly specified. We considered a range of scenarios in terms of the extent of confounding. The methods' performances were assessed using the following criteria: (i) bias in estimating the true marginal log-hazard ratio; (ii) the mean squared error (MSE) of the estimated log-hazard ratio; and (iii) the empirical coverage rates of nominal 95% confidence intervals.

Practical Assessment, Research & Evaluation

A peer-reviewed electronic journal.

Volume 20, Number 13, June 2015 ISSN 1531-7714

A Practical Guide for Using Propensity Score Weighting in R

Antonio Olmos & Priyalatha Govindasamy
University of Denver

Propensity score weighting is one of the techniques used in controlling for selection biases in non-experimental studies. Propensity scores can be used as weights to account for selection assignment differences between treatment and comparison groups. One of the advantages of this approach is that all the individuals in the study can be used for the outcomes evaluation. In this paper, we demonstrate how to conduct propensity score weighting using R. The purpose is to provide a step-by-step guide to propensity score weighting implementation for practitioners. In addition to strengths, some limitations of propensity score weighting are discussed.

The use of propensity scores is becoming part of the evaluation landscape (Guo & Fraser, 2015). Rosenbaum and Rubin (1983) introduced the concept of propensity score analysis to address selection bias when random assignment is not feasible. As defined by Rosenbaum and Rubin, a propensity score is the conditional probability of assignment to a treatment condition given a set of observed covariates: e = p(z=i|X). When propensity scores are used, the resulting groups will have similar characteristics to those created through random assignment. Most of the applications related to propensity scores are in matching (Thoemmes & Kim, 2011). Recently, Randolph, Fable, Manuel, and Balloun (2014) in this journal, described in detail how to conduct propensity score matching using R.

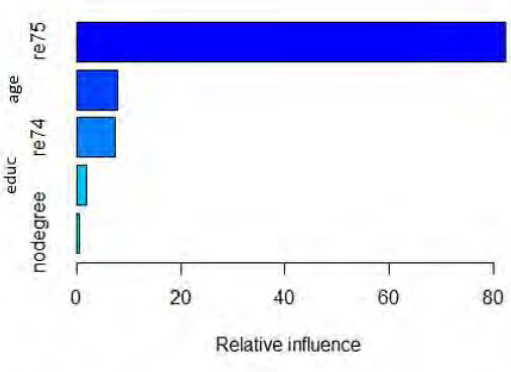
A potential drawback of propensity scores when used for matching is that a very large number of subjects may be needed, especially in the control group (Guo & Fraser, 2015). And, dependent on the specific matching technique, the use of a caliper, and the number of subjects being matched to every subject in the control group (1 or more), a large number of those subjects in the control group may not be used (see Randolph et. al, 2014 for more information about propensity score matching). Given that in evaluation settings, data collection is costly for both treatment and control subjects, techniques that may be able to use all the subjects in the study pool should be preferred to techniques that discard substantial amounts of data. Propensity scores can also be used as weights in a linear model such as regression or ANOVA, so all the subjects in the control and treatment group can be used for this application.

This article will illustrate how to use propensity scores as weights in a weighted regression using R. Program evaluators can benefit tremendously from the ability to use propensity scores to create treatment and control groups that are matched in every way except for the intervention. This is especially appealing when this ability to match individuals will not mean sacrificing individuals who cannot be matched. In that sense, using propensity scores as weights represents a very powerful combination.

Using Propensity Scores as weights in a weighted regression

The idea behind the use of propensity scores as weights is to control the influence of participants by weighting their responses based on their propensity scores (also known as reweighting, McCaffrey, Ridgeway & Morrall, 2004). The key of this analysis is the creation of weights based on propensity scores.

may be more effective for the estimation of propensity scores. Some evidence seems to suggest that generalized boosted models may outperform logistic regression (McCaffrey et al, 2004), but this is an area that needs more research. Readers are encouraged to describe in detail the procedure they followed for their development of propensity scores, as well as to scan the literature for new developments in this area.




```
##                var    rel.inf
## re75          re75 82.3429390
## age            age  7.9577202
## re74          re74  7.3572500
## educ          educ  1.9110088
## nodegree nodegree  0.4310819

#--- Attaching propensity score to the lalonde dataset
>lalonde$gpsvalue <- predict(gps, type = "response")
```

Figure 5. Relative influence of the variables in the generalized boosted model and code to bind the predicted scores to the datafile.

Note: the subcommand type = "response" saves the predicted value from the Generalized Boosted Model so it can be used as the propensity score.

4. Weight estimation using propensity scores

In order to use propensity scores in a weighted regression, the propensity scores ($\hat{e}(x)$) need to be transformed so they can be used as weights in a linear regression. The weight estimates for the **ATE** are estimated as follows: for the individuals in the treatment group:

$$\omega = \frac{1}{\hat{e}(x)} \quad (1)$$

And for the individuals in the control group:

$$\omega = \frac{1}{1 - \hat{e}(x)} \quad (2)$$

As explained earlier, every statistical software package that runs regression has routines for weighted regression. Figures 6 and 7 shows the code needed to transform the propensity scores according to equations 1 and 2. These equations can be used with propensity scores calculated using either logistic regression (Figure 6) or Generalized Boosted Models (Figure 7). The code used to compute and save weights to the dataset follows.

```
a. Estimation and storing weights using propensity score estimated using
logistic regression

#---Weights for ATE: We define the weights using :
## for the treatment group: 1/lalonde$psvalue
## for the control group: 1/(1-lalonde$psvalue)

#---Attaching weight to the lalonde dataset
>lalonde$weight.ATE <- ifelse(lalonde$treat == 1, 1/lalonde$psvalue, 1/(1-
lalonde$psvalue))
```

Figure 6. Code to transform the propensity scores to weights estimated using logistic regression.

```
b. Estimation and storing weights using propensity score estimated using
Generalized Boosted Model

# Weights for ATE: We define the weights using :
## for the treatment group: 1/lalonde$gpsvalue
## for the control group: 1/(1-lalonde$gpsvalue)


>lalonde$gweight.ATE <- ifelse(lalonde$treat == 1, 1/lalonde$gpsvalue, 1/
(1-lalonde$gpsvalue))
```

Figure 7. Code to transform the propensity scores to weights estimated using the Generalized Boosted Model

5. Balance analysis after implementing propensity scores

The ultimate purpose of using propensity scores is to balance the treatment/comparison groups on the observed covariates. This purpose does not change when using the propensity scores as weights in a weighted regression. To assess the success of the propensity scores as weights in a weighted regression for removing selection bias, a new set of tests to check the balance should be performed. Figures 8 and 9 present the results of tests similar to those presented in Figure 2. However, now weighted linear and generalized linear regressions are performed using the computed propensity scores as weights. Figure 8 presents the balance assessment for a continuous (re74) and a categorical (nodegree) variable using propensity scores weights computed using logistic regression. Figure 9 presents the balance for the same variables using propensity scores weights using the Generalized Boosted Model.

Figures 8 and 9 show that to run the balance test after creating the propensity scores the subcommand "**weights** = (**weight**.**ATE**)" for logistic regression, or

estimate of the average treatment effect. However, in an observational study, we have that, in general, $E[Y(1)|Z = 1] \neq E[Y(1)]$. Thus, in an observational study simply comparing outcomes between the two treatment groups does not necessarily yield an unbiased estimate of the average treatment effect.

2.2. *The propensity score and inverse probability of treatment weighting*

As previously discussed, let Z denote treatment assignment (Z = 1 denoting treatment; Z = 0 denoting absence of treatment), and let **X** denote a vector of observed baseline covariates. The propensity score is defined as $e = P(Z = 1|\mathbf{X})$: the probability of a subject receiving the treatment of interest conditional on their observed baseline covariates [1]. The inverse probability of treatment weight is defined as $w = \frac{Z}{e} + \frac{1-Z}{1-e}$. Each subject's weight is equal to the inverse of the probability of receiving the treatment that the subject received [4].

Lunceford and Davidian provide a review of methods for estimating treatment effects that use weighting by the inverse of the probability of treatment [14]. If Y denotes an outcome variable, the average treatment effect (ATE) can be estimated by $\frac{1}{n}\sum_{i=1}^{n}\frac{Z_iY_i}{e_i} - \frac{1}{n}\sum_{i=1}^{n}\frac{(1-Z_i)Y_i}{1-e_i}$, where n denotes the number of subjects. An alternative estimator of the ATE is $\left(\sum_{i=1}^{n}\frac{Z_i}{e_i}\right)^{-1}\sum_{i=1}^{n}\frac{Z_iY_i}{e_i} - \left(\sum_{i=1}^{n}\frac{1-Z_i}{1-e_i}\right)^{-1}\sum_{i=1}^{n}\frac{(1-Z_i)Y_i}{1-e_i}$ [14]. When the propensity score model is correctly specified, both estimators are consistent estimators of the true treatment effect [14]. However, Lunceford and Davidian found that in empirical studies, in general, the variance of the former estimator is greater than that of the latter estimator [14].

Joffe *et al*. describe how weighting by the inverse probability of treatment results in an artificial population in which baseline covariates are independent of treatment status [15]. Furthermore, Joffe *et al*. describe how regression models can be combined with weighting by the inverse probability of treatment to estimate causal treatment effects. While weighting by the inverse probability of treatment allows the comparison of expectations and distributions between treated and control subjects, methods that account for the weighting must be used in estimating variances and significance levels [14, 15]. For instance, Joffe *et al*. suggest that a robust, sandwich-type variance estimator be used to account for the fact that the weights are estimated, rather than known with certainty. Other alternatives to variance estimation include bootstrap-based methods.

A difficulty that can arise when using the weights described previously is that treated subjects with a very low propensity score can result in a very large weight. Similarly, a control subject with a propensity score close to one can result in a very large weight. Such weights can increase the variability of the estimated treatment effect [16]. An alternative to the conventional weights described previously is to use stabilized weights: $w = \frac{Z\Pr(Z=1)}{e} + \frac{(1-Z)\Pr(Z=0)}{1-e}$ [16]. $\Pr(Z = 1)$ and $\Pr(Z = 0)$ denote the marginal probability of treatment and control in the overall sample. Another alternative to address the problems that can arise with very large weights is to use trimmed or truncated weights, in which weights that exceed a specified threshold are each set to that threshold [16, 17]. The threshold is often based on quantiles of the distribution of the weights (e.g., the 1st and 99th percentiles).

The weights described previously $\left(w_{\text{ATE}} = \frac{Z}{e} + \frac{1-Z}{1-e}\right)$ permit estimation of the ATE. However, a different set of weights permit estimation of the average treatment effect in the treated (ATT): $w_{\text{ATT}} = Z + \frac{e(1-Z)}{1-e}$ [18]. These weights are obtained by multiplying the conventional weights by e, so that treated subjects receive a weight of one. Thus, the treated sample is being used as the reference population to which the treated and control samples are being standardized. While the current article is focused on the use of the ATE weights, the balance diagnostics discussed are equally applicable to situations in which the ATT weights are employed.

2.3. *Variable selection for the propensity score model*

The propensity score is defined as the probability of treatment selection conditional on measured baseline covariates. A natural question that arises is what variables should be included in the propensity score model. A reasonable suggestion would be to include those variables that influence the treatment selection process. A different answer can be obtained by remembering the primary property of the propensity score: that it is a balancing score [1]. Thus, conditioning on the propensity score permits one to balance the distribution of measured baseline covariates between treated and control subjects. Accordingly, Rosenbaum suggests that one ask 'which covariates do you wish to balance by matching on the propensity score?' [19] (page 356). The goal of propensity score analyses should be to induce balance in measured baseline

Oxford Journals Science & Mathematics Biometrika Volume 96 Issue 1 Pp. 187-199.

Dealing with limited overlap in estimation of average treatment effects

Richard K. Crump

Department of Economics, University of California, Berkeley, California 94720, U.S.A.
crump{at}econ.berkeley.edu

V. Joseph Hotz

Department of Economics, Duke University, Durham, North Carolina 27708, U.S.A.
hotz{at}econ.duke.edu

Guido W. Imbens

Department of Economics, Harvard University, Cambridge, Massachusetts 02138, U.S.A.
imbens{at}harvard.edu

Oscar A. Mitnik

Department of Economics, University of Miami, Coral Gables, Florida 33124, U.S.A.
omitnik{at}miami.edu

Received June 1, 2007.
Revision received June 1, 2008.

Abstract

Estimation of average treatment effects under unconfounded or ignorable treatment assignment is often hampered by lack of overlap in the covariate distributions between treatment groups. This lack of overlap can lead to imprecise estimates, and can make commonly used estimators sensitive to the choice of specification. In such cases researchers have often used ad hoc methods for trimming the sample. We develop a systematic approach to addressing lack of overlap. We characterize optimal subsamples for which the average treatment effect can be estimated most precisely. Under some conditions, the optimal selection rules depend solely on the propensity score. For a wide range of distributions, a good approximation to the optimal rule is provided by the simple rule of thumb to discard all units with estimated propensity scores outside the range [0.1,0.9].

Key words Average treatment effect Causality Ignorable treatment assignment Overlap Propensity score Treatment effect heterogeneity Unconfoundedness

Online ISSN 1464-3510 - Print ISSN 0006-3444

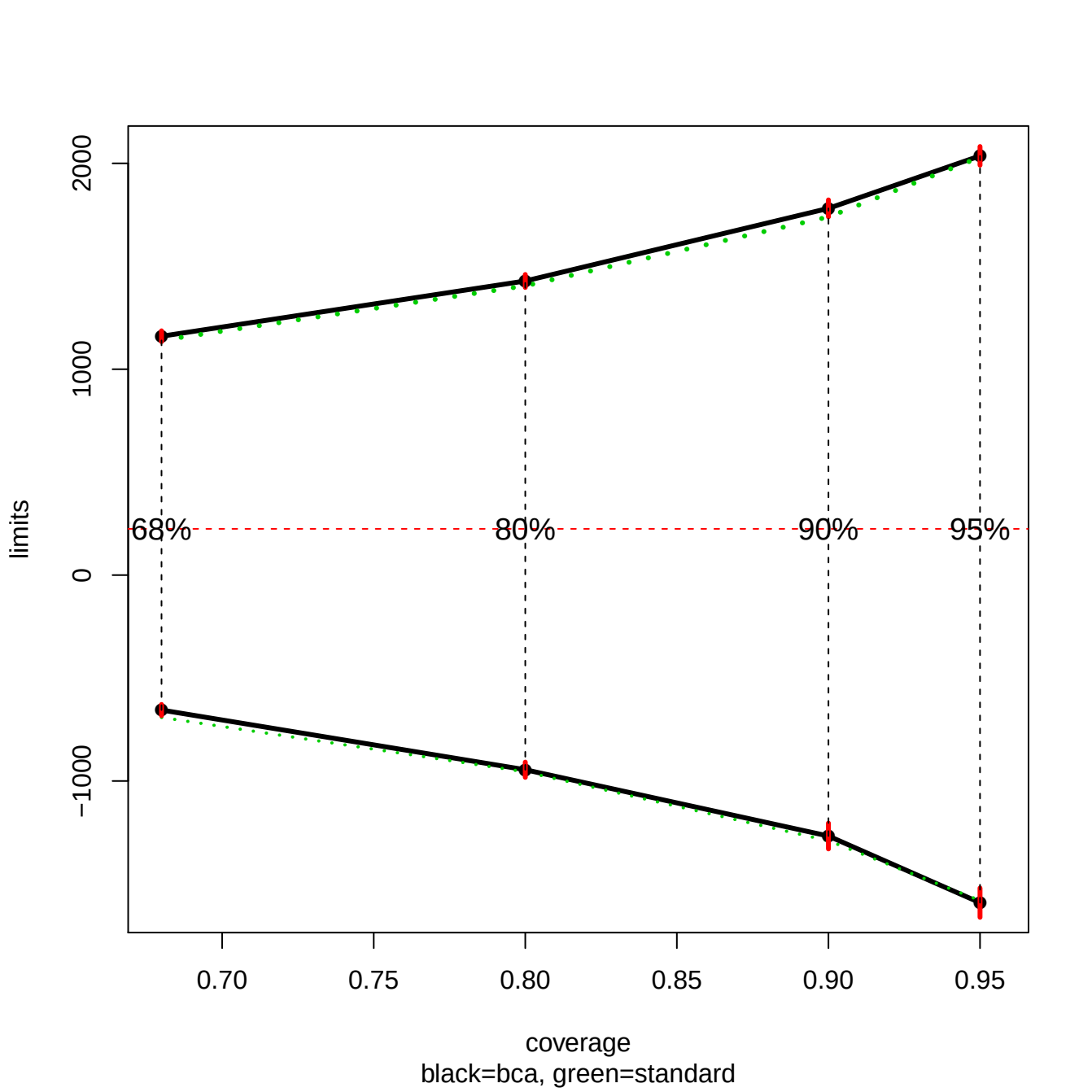
```
> set.seed(1) # for tutorial match
> data(lalonde) # like week1 ComCo
```

```
> # Details IPTW for later
> # Weights for ATT are 1 for the treatment cases and p/(1-p) for the control cases.
> # Weights for ATE are 1/p for the treatment cases and 1/(1-p) for the control cases
```

```
#### CC_4, IPW
R version 3.5.3 (2019-03-11) -- "Great Truth"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)


> library(MatchIt)
> data(lalonde)
> head(lalonde)
     treat age educ black hispan married nodegree re74 re75       re78
NSW1     1  37   11     1      0       1        1    0    0  9930.0460
NSW2     1  22    9     0      1       0        1    0    0  3595.8940
NSW3     1  30   12     1      0       0        0    0    0 24909.4500
NSW4     1  27   11     1      0       0        1    0    0  7506.1460
NSW5     1  33    8     1      0       0        1    0    0   289.7899
NSW6     1  22    9     1      0       0        1    0    0  4056.4940
> ps.lalonde = fitted(glm(treat ~ age + educ + black + hispan + nodegree + + married + re74 + re75, family = binomial, data = lalonde)
> lalonde$ps = ps.lalonde
> head(lalonde)
     treat age educ black hispan married nodegree re74 re75       re78        ps
NSW1     1  37   11     1      0       1        1    0    0  9930.0460 0.6387699
NSW2     1  22    9     0      1       0        1    0    0  3595.8940 0.2246342
NSW3     1  30   12     1      0       0        0    0    0 24909.4500 0.6782439
NSW4     1  27   11     1      0       0        1    0    0  7506.1460 0.7763241
NSW5     1  33    8     1      0       0        1    0    0   289.7899 0.7016387
NSW6     1  22    9     1      0       0        1    0    0  4056.4940 0.6990699
> fivenum(lalonde$ps)
[1] 0.009080193 0.048502044 0.120676493 0.638769933 0.853152844
> # did at least that much right ....
>  # Details IPTW
>  # Weights for ATT are 1 for the treatment cases and p/(1-p) for the control cases.
>  # Weights for ATE are 1/p for the treatment cases and 1/(1-p) for the control cases
> #implement ATE as
> lalonde$wATE = ifelse(lalonde$treat ==1, 1/lalonde$ps, 1/(1 - lalonde$ps))
> head(lalonde)
     treat age educ black hispan married nodegree re74 re75       re78        ps     wATE
NSW1     1  37   11     1      0       1        1    0    0  9930.0460 0.6387699 1.565509
NSW2     1  22    9     0      1       0        1    0    0  3595.8940 0.2246342 4.451681
NSW3     1  30   12     1      0       0        0    0    0 24909.4500 0.6782439 1.474396
NSW4     1  27   11     1      0       0        1    0    0  7506.1460 0.7763241 1.288122
NSW5     1  33    8     1      0       0        1    0    0   289.7899 0.7016387 1.425235
NSW6     1  22    9     1      0       0        1    0    0  4056.4940 0.6990699 1.430472
> tail(lalonde)
        treat age educ black hispan married nodegree re74 re75       re78         ps     wATE
PSID424     0  25   14     0      0       0        0    0    0     0.0000 0.11145877 1.125440
PSID425     0  18   11     0      0       0        1    0    0 10150.5000 0.12314388 1.140438
PSID426     0  24    1     0      1       1        1    0    0 19464.6100 0.03456038 1.035798
PSID427     0  21   18     0      0       0        0    0    0     0.0000 0.18335100 1.224516
PSID428     0  32    5     1      0       1        1    0    0   187.6713 0.38303231 1.620830
PSID429     0  16    9     0      0       0        1    0    0  1495.4590 0.08971192 1.098553
# wATE correct
```

```
> lmATE = lm(re78 ~ treat, data = lalonde, weights = (wATE)) # weighted OLS
> summary(lmATE)

Call:
lm(formula = re78 ~ treat, data = lalonde, weights = (wATE))

Weighted Residuals:
   Min     1Q Median     3Q    Max
-42083  -6606  -2284   4979  77674

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   6422.8      397.4  16.161   <2e-16 ***
treat          224.7      577.7   0.389    0.697
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9864 on 612 degrees of freedom
Multiple R-squared:  0.0002471, Adjusted R-squared:  -0.001386
F-statistic: 0.1513 on 1 and 612 DF,  p-value: 0.6975

> #note unweighted t-test cc1, control was 635 higher on outcome
> confint(lmATE)
                2.5 %   97.5 %
(Intercept) 5642.353 7203.325
treat       -909.755 1359.108

# Believed weight_lm gives too small s.e. (surveyreg somewhat better)
# often bootstrap is used to give morre accurate s.e. and CI

## Introducing the new bcaboot (efron)

> library(bcaboot)

> lmATE$coef
(Intercept)       treat
  6422.8390    224.6763
> lmATE$coef[2]
   treat
224.6763


> rfun <- function(lalonde) {
+
+ y <- lalonde$re78
+ x <- lalonde$treat
+ lm(y ~ x, weights = (lalonde$wATE))$coef[2]  }
> result <- bcajack(x = lalonde, B = 1000, func = rfun, verbose = FALSE)
> result$lims
             bca    jacksd        std   pct
0.025 -1576.1512 112.30007 -1490.4244 0.016
0.05  -1335.2714  55.62231 -1214.6816 0.034
```

```
0.1    -985.4604  73.58843  -896.7678 0.075
0.16   -738.9990  38.10547  -645.5414 0.127
0.5     164.8410  30.00823   224.6763 0.448
0.84   1005.0343  57.87255  1094.8940 0.803
0.9    1227.8668  68.61720  1346.1204 0.871
0.95   1522.3414  65.42471  1664.0342 0.931
0.975  1829.2775 122.19899  1939.7770 0.962
> result <- bcajack(x = lalonde, B = 4000, func = rfun, verbose = FALSE)
> # maybe 20secs
> result$lims
             bca   jacksd        std     pct
0.025 -1615.6117 48.84414 -1556.9294 0.01875
0.05  -1350.7139 47.30824 -1270.4944 0.04025
0.1    -952.9389 50.47428  -940.2531 0.08525
0.16   -703.2580 26.67179  -679.2851 0.14125
0.5     195.0399 21.76850   224.6763 0.47350
0.84   1082.7824 42.89119  1128.6378 0.82025
0.9    1343.2181 48.85178  1389.6057 0.88400
0.95   1671.2320 37.45937  1719.8470 0.93900
0.975  1963.3432 38.55807  2006.2820 0.96750
> result <- bcajack(x = lalonde, B = 4000, func = rfun, verbose = FALSE)
> result$lims
             bca   jacksd        std     pct
0.025 -1591.1985 71.10970 -1580.5113 0.02275
0.05  -1267.3800 62.66232 -1290.2849 0.04750
0.1    -945.4222 37.05866  -955.6724 0.09825
0.16   -655.4565 27.13271  -691.2502 0.15975
0.5     228.1372 13.50145   224.6763 0.50425
0.84   1160.1825 26.36629  1140.6028 0.83975
0.9    1428.5868 31.45005  1405.0250 0.89825
0.95   1781.2698 41.08429  1739.6375 0.94750
0.975  2036.2541 45.38391  2029.8639 0.97275
> bcaplot(result) # see plot
> #bca doesn't do that much here
> ###############################################################################
> # perfunctory balance checks
> # by wOLS O&G fig8; or A&S sec 4 by weighted means
> bal_re74 = lm(re74 ~ treat, weights = (wATE), data = lalonde)
> summary(bal_re74)

Call:
lm(formula = re74 ~ treat, data = lalonde, weights = (wATE))

Weighted Residuals:
   Min     1Q Median     3Q    Max
-14181  -4678  -3338   3816  83133

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4552.7      336.3  13.539  < 2e-16 ***
treat        -1620.6      488.8  -3.316 0.000969 ***
---
```


limits
2000
1000
0
−1000
68%
80%
90%
95%
0.70
0.75
0.80
0.85
0.90
0.95
coverage
black=bca, green=standard

While it was not the focus of the review, we noted that several sets of authors incorrectly defined the weights as the reciprocal of the propensity score, rather the reciprocal of the probability of receiving the treatment that was actually received.

4. IPTW diagnostics

In this section, we describe both quantitative and qualitative methods for assessing balance in observed baseline covariates between treated and control subjects in a sample weighted by the inverse probability of treatment. We also describe methods for assessing the validity of the positivity assumption.

4.1. Balance diagnostics

In this sub-section, we consider diagnostics for assessing the balance of baseline covariates between treated and control subjects in a sample weighted by the inverse probability of treatment. As noted previously, the objective of IPTW analyses is to create a weighted sample in which the distribution of either the confounding variables or the prognostically important covariates is the same between treated and control subjects.

4.1.1. Comparison of means and proportions of baseline variables. The first quantitative method compares the means of observed baseline covariates between treated and control subjects in the weighted sample. For a continuous variable, let $\bar{x}_{\text{treatment}}$ and $\bar{x}_{\text{control}}$ denote the sample mean of X in treated and control subjects, respectively, while $s^2_{\text{treatment}}$ and s^2_{control} denote the sample variance of X in treated and control subjects, respectively. Similarly, for a dichotomous variable, $\hat{p}_{\text{treatment}}$ and $\hat{p}_{\text{control}}$ denote the sample prevalence of the variable in treated and control subjects, respectively. In an unweighted sample, the standardized difference is defined as

$$d = 100 \times \frac{(\bar{x}_{\text{treatment}} - \bar{x}_{\text{control}})}{\sqrt{\frac{s^2_{\text{treatment}} + s^2_{\text{control}}}{2}}} \tag{1}$$

for continuous variables. The standardized difference was developed for comparing continuous variables; however, it can justifiably be used for comparing dichotomous variables [38]. For dichotomous variables the standardized difference is defined as

$$d = 100 \times \frac{(\hat{p}_{\text{treatment}} - \hat{p}_{\text{control}})}{\sqrt{\frac{\hat{p}_{\text{treatment}}(1-\hat{p}_{\text{treatment}}) + \hat{p}_{\text{control}}(1-\hat{p}_{\text{control}})}{2}}} \tag{2}$$

The standardized difference compares the difference in means in units of the pooled standard deviation [39]. Unlike *t*-tests and other statistical tests of hypothesis, the standardized difference is not influenced by sample size. Thus, the use of the standardized difference can be used to compare balance in measured variables between treated and control subjects in the same sample when different weights are assigned to the same subjects. Furthermore, it allows for the comparison of the relative balance of variables measured in different units (e.g., age in years vs. systolic blood pressure in mm Hg) by calculating each on the standard deviation scale.

The sample means, sample variances, and sample prevalences in formulas (1) and (2) are unweighted estimates. However, each sample estimate can be replaced by its weighted equivalent. The weighted mean is defined as $\bar{x}_{\text{weight}} = \frac{\sum w_i x_i}{\sum w_i}$, while the weighted sample variance is defined as $s^2_{\text{weight}} = \frac{\sum w_i}{(\sum w_i)^2 - \sum w_i^2} \sum w_i (x_i - \bar{x}_{\text{weight}})^2$, where w_i is the weight assigned to the i-*th* subject. In our context, the weight is the inverse probability of treatment received, as defined in Section 2. The use of standardized differences allows researchers to quantitatively compare balance in measured baseline covariates between treated and control subjects in the sample weighted by the inverse probability of treatment.

4.1.2. Comparison of interactions and higher-order moments of continuous variables. The methods described in the previous section allow one to compare means and prevalences of continuous and dichotomous variables, respectively, between treated and control subjects in the weighted sample. However, one desires to balance not only means and prevalences but also other characteristics of the distribution. In particular, higher-order moments and interactions between variables should be similar between

"(**weight.gATE**)" for the Generalized Boosted Model need to be added to the regression model. Figures 8 and 9 also show that although there was some improvement in the balance, there might still be some bias, since the coefficient for the variable treat is statistically significant.

```
## 1. Propensity score weights: Logistic regression

#--- re74 (a continuous variable)----------------------------------------
>aft.re74.ATE <- lm(re74 ~ treat, data=data, weights = (weight.ATE))
>summary(aft.re74.ATE)

## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4616.8      574.7   8.034 4.88e-15 ***
## treat         3910.1      787.2   4.967 8.82e-07 ***
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#--- nodegree (a dichotomous variable)-----------------------------------
>aft.nodegree.ATE <- glm(nodegree ~ treat, data=data, family = binomial(),
 weights = (weight.ATE))
>summary(aft.nodegree.ATE)

## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.51160    0.08353   6.125 9.08e-10 ***
## treat       -0.32914    0.11294  -2.914  0.00357 **
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 8. Balance checking using propensity score weights computed using logistic regression

```
## 2. Propensity score weights: Generalized Boosted Model

#--- re74 (a continuous variable)----------------------------------------
>aft.re74.gATE <- lm(re74 ~ treat, data=lalonde, weights = (gweight.ATE))
>summary(aft.re74.gATE)

## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5618.6      320.6  17.523  < 2e-16 ***
## treat        -3522.0      497.0  -7.087 3.8e-12 ***
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

#--- nodegree (a dichotomous variable)-----------------------------------
>aft.nodegree.gATE <- glm(nodegree ~ treat, data=lalonde, family = binomia
l(), weights = (gweight.ATE))
>summary(aft.nodegree.gATE)

## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.3918     0.0777   5.043 4.57e-07 ***
## treat         0.4943     0.1261   3.921 8.81e-05 ***
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 9. Balance checking using propensity score weights computed using the Generalized Boosted Model

6. Outcomes analysis using propensity scores as weights in a weighted regression

The final step in the analysis is to run the outcomes model using the propensity scores as weights. In this example the outcomes model includes re78 as the outcome variable and treat, black, hispanic and married as independent variables. Figures 10 and 11 present the code to run a weighted regression to estimate ATE scores using weights computed through logistic regression (Figure 10) or the Generalized Boosted Model (Figure 11).

```
#---1.Weighted regression analysis using propensity score weights from
logistic regression

>model.ATE <- lm(re78 ~ treat + black + hispan + married, data = lalonde,
weights=(weight.ATE))
>summary(model.ATE)

## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2645.8      769.6   3.438 0.000626 ***
## treat         3684.0     1003.8   3.670 0.000264 ***
## black         1068.6     1076.7   0.992 0.321372
## hispan        -232.6     1389.7  -0.167 0.867133
## married       8018.0      787.4  10.183  < 2e-16 ***
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Figure 10. Weighted regression using propensity score weights computed through logistic regression

```
#---2. Weighted regression analysis using propensity score as weights from
 Generalized Boosted Model.

>model.gATE <- lm(re78 ~ treat + black + hispan + married, data = lalonde,
 weights=(gweight.ATE))
>summary(model.gATE)

## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6318.6      601.8  10.499  < 2e-16 ***
## treat         1241.8      801.4   1.549  0.12179
## black        -1892.0      831.0  -2.277  0.02315 *
## hispan        -423.6     1032.0  -0.410  0.68161
## married       2163.1      665.2   3.252  0.00121 **
```

Figure 11. Weighted regression using propensity score weights computed through the Generalized Boosted Model

A comparison of the results of the analysis presented in Figures 10 and 11 to the results of the analysis presented in Figure 1 shows that the effect of the treatment was statistically significant after the groups are balanced using propensity scores. However, this was only the case when the propensity scores were estimated using logistic regression (Figure 10). There was no statistically significant treatment effect after balancing the groups using propensity scores estimated using the Generalized Boosted Model (Figure 11). The discrepancy in the results is a clear indication that the model fit achieved by both techniques is different. It also highlights the fact that more research is needed to determine which technique may yield results that are more accurate. To date, the authors are not aware of any formal comparison between the goodness of fit for these two techniques. Readers are encouraged to search the literature for new research that may help clarify this difference.

Conclusion

Propensity score analysis is a technique that has proven useful for evaluators trying to assess the outcomes of quasi-experiments and observational studies. A drawback associated with propensity scores

```
0.1    -985.4604  73.58843  -896.7678 0.075
0.16   -738.9990  38.10547  -645.5414 0.127
0.5     164.8410  30.00823   224.6763 0.448
0.84   1005.0343  57.87255  1094.8940 0.803
0.9    1227.8668  68.61720  1346.1204 0.871
0.95   1522.3414  65.42471  1664.0342 0.931
0.975  1829.2775 122.19899  1939.7770 0.962
> result <- bcajack(x = lalonde, B = 4000, func = rfun, verbose = FALSE)
> # maybe 20secs
> result$lims
             bca   jacksd        std     pct
0.025 -1615.6117 48.84414 -1556.9294 0.01875
0.05  -1350.7139 47.30824 -1270.4944 0.04025
0.1    -952.9389 50.47428  -940.2531 0.08525
0.16   -703.2580 26.67179  -679.2851 0.14125
0.5     195.0399 21.76850   224.6763 0.47350
0.84   1082.7824 42.89119  1128.6378 0.82025
0.9    1343.2181 48.85178  1389.6057 0.88400
0.95   1671.2320 37.45937  1719.8470 0.93900
0.975  1963.3432 38.55807  2006.2820 0.96750
> result <- bcajack(x = lalonde, B = 4000, func = rfun, verbose = FALSE)
> result$lims
             bca   jacksd        std     pct
0.025 -1591.1985 71.10970 -1580.5113 0.02275
0.05  -1267.3800 62.66232 -1290.2849 0.04750
0.1    -945.4222 37.05866  -955.6724 0.09825
0.16   -655.4565 27.13271  -691.2502 0.15975
0.5     228.1372 13.50145   224.6763 0.50425
0.84   1160.1825 26.36629  1140.6028 0.83975
0.9    1428.5868 31.45005  1405.0250 0.89825
0.95   1781.2698 41.08429  1739.6375 0.94750
0.975  2036.2541 45.38391  2029.8639 0.97275
> bcaplot(result) # see plot
> #bca doesn't do that much here
> ##############################################################################
> # perfunctory balance checks
> # by wOLS O&G fig8; or A&S sec 4 by weighted means
> bal_re74 = lm(re74 ~ treat, weights = (wATE), data = lalonde)
> summary(bal_re74)

Call:
lm(formula = re74 ~ treat, data = lalonde, weights = (wATE))

Weighted Residuals:
   Min     1Q Median     3Q    Max
-14181  -4678  -3338   3816  83133

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   4552.7      336.3  13.539  < 2e-16 ***
treat        -1620.6      488.8  -3.316 0.000969 ***
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8346 on 612 degrees of freedom
Multiple R-squared:  0.01765,   Adjusted R-squared:  0.01604
F-statistic: 10.99 on 1 and 612 DF,  p-value: 0.0009686

> # treat signif predictor of IPW weighted re74, is that bad?
> 3.316/sqrt(612)
[1] 0.1340414
> # a little greater than .10

> # try a dichtomous var... nodegree
> bal_nod = glm(nodegree ~ treat, weights = (wATE), family = binomial, data = lalonde)
Warning message:
In eval(family$initialize) : non-integer #successes in a binomial glm!
> summary(bal_nod)

Call:
glm(formula = nodegree ~ treat, family = binomial, data = lalonde,
    weights = (wATE))

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-8.2276  -1.4397   0.9936   1.2273   5.1259

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.51063    0.08322   6.136 8.48e-10 ***
treat       -0.22779    0.11957  -1.905   0.0568 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1575.2  on 613  degrees of freedom
Residual deviance: 1571.6  on 612  degrees of freedom
AIC: 1538.6

Number of Fisher Scoring iterations: 5

> # note O&G get -.33 for log-odds, but they use a subset of the lalonde vars
> # so in log-odds metric
> stand_diff = .2278/(sqrt(612)*.1196)
> stand_diff
[1] 0.07699222
> #not bad
> # twang gives pretty balance info for IPW comparisons
>
################################################################################
also show lindner, cursory, here pick up from wek2 session
propensity score picked off from matchit 'distance', could directly compute
> ################## WEEK 4 Addendum, Lindner data  IPTW analyses
```

```
> set.seed(1) # for tutorial match
> data(lalonde) # like week1 ComCo
```

```
> # Details IPTW for later
> # Weights for ATT are 1 for the treatment cases and p/(1-p) for the control cases.
> # Weights for ATE are 1/p for the treatment cases and 1/(1-p) for the control cases
```

```
> ################ WEEK 3 Addendum, Lindner data  IPTW analyses

> ## now try IPTW (see eval reference), 'alternative to full matching"
> ## we have propensity scores as 'distance' in m2full.dat
> detach(m2full.dat)
```

no trimming done here, good overlap in Lindner

```
> # weights for estimating ATE (according to lore)
> # for abcix = 1, 1/(m2full.dat$distance)
> # for abcix = 0, 1/(1 - m2full.dat$distance)
```

propensity score is 'distance' in matchit; could also compute from a fresh logistic regression. Will redo a 3.3.3 version for posting

```
> m2full.dat$weight.ATE =
                ifelse(m2full.dat$abcix == 1, 1/m2full.dat$distance, 1/(1 - m2full.dat$distance))
> head(m2full.dat)
  lifepres cardbill abcix stent height female diabetic acutemi ejecfrac ves1proc  distance weights subc
1      0.0    14301     1     0    163      1        1       0       56         1 0.4079170        1
2     11.6     3563     1     0    168      0        0       0       56         1 0.5784602        1
3     11.6     4694     1     0    188      0        0       0       50         1 0.5244469        1
4     11.6     7366     1     0    175      0        1       0       50         1 0.4727311        1
5     11.6     8247     1     0    168      1        0       0       55         1 0.4930466        1
6     11.6     8319     1     0    178      0        0       0       50         1 0.5625524        1
> tail(m2full.dat) # yes there are non-drug subjects with high propensity (see boxplot)
    lifepres cardbill abcix stent height female diabetic acutemi ejecfrac ves1proc  distance  weights s
991     11.6    15176     0     1    180      0        0       0       55         3 0.9038603 3.842407
992     11.6    15736     0     0    170      0        0       1       51         1 0.8262407 3.842407
993     11.6    19547     0     1    170      0        0       1       55         2 0.9444637 5.123209
994     11.6    36834     0     0    183      0        0       0       35         3 0.8718633 4.696275
995     11.6    46124     0     0    175      0        0       0       45         4 0.9342004 1.280802
996     11.6    47732     0     1    185      0        0       0       40         2 0.8355348 2.561605
```

```
# outcome IPTW analaysis
> lm.IPTW = lm(log(cardbill) ~ abcix, data = m2full.dat, weights = (weight.ATE))
> summary(lm.IPTW)
Call: lm(formula = log(cardbill) ~ abcix, data = m2full.dat, weights = (weight.ATE))
Weighted Residuals:
    Min      1Q  Median      3Q     Max
-3.2034 -0.4174 -0.1548  0.2535  6.2000
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.47026    0.02245 421.787  < 2e-16 ***
abcix        0.10322    0.03184   3.242  0.00123 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7122 on 994 degrees of freedom
Multiple R-squared:  0.01046,   Adjusted R-squared:  0.009466
F-statistic: 10.51 on 1 and 994 DF,  p-value: 0.001228

> confint(lm.IPTW)
                 2.5 %    97.5 %
(Intercept) 9.42620153 9.5143218
abcix       0.04073608 0.1657008
> # a little smaller effect on cardbill than say lmer, full matching
```

could also use regression tools from survey package. RQ?.x

```
> lm2.IPTW = lm(log(cardbill) ~ abcix + acutemi + ejecfrac, data = m2full.dat, weights = (weight.ATE))
> summary(lm2.IPTW)
Call: lm(formula = log(cardbill) ~ abcix + acutemi + ejecfrac, data = m2full.dat,
    weights = (weight.ATE))
Weighted Residuals:
    Min      1Q  Median      3Q     Max
-3.1474 -0.4037 -0.1484  0.2401  6.1726

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.979448   0.081983 121.726  < 2e-16 ***
abcix        0.103107   0.031224   3.302 0.000994 ***
acutemi     -0.061709   0.044696  -1.381 0.167703
ejecfrac    -0.009819   0.001533  -6.406  2.3e-10 ***
```

```
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6984 on 992 degrees of freedom
Multiple R-squared:  0.05032,   Adjusted R-squared:  0.04745
F-statistic: 17.52 on 3 and 992 DF,  p-value: 4.34e-11

> # adding some of the covariates leaves unchanged
> # exercise try out this analysis with lifepres outcome

> lm2.IPTW = lm(log(cardbill) ~ abcix + acutemi + ejecfrac + ves1proc,
                                              data = m2full.dat, weights = (weight.ATE))
> summary(lm2.IPTW)
Call: lm(formula = log(cardbill) ~ abcix + acutemi + ejecfrac + ves1proc,
    data = m2full.dat, weights = (weight.ATE))
Weighted Residuals:
    Min      1Q  Median      3Q     Max
-3.0136 -0.3896 -0.1455  0.2404  4.8563

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  9.647987   0.087654 110.069  < 2e-16 ***
abcix        0.111768   0.030117   3.711 0.000218 ***
acutemi     -0.099223   0.043301  -2.291 0.022146 *
ejecfrac    -0.008693   0.001483  -5.861 6.27e-09 ***
ves1proc     0.195657   0.022378   8.743  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6733 on 991 degrees of freedom
Multiple R-squared:  0.1183,    Adjusted R-squared:  0.1148
F-statistic: 33.25 on 4 and 991 DF,  p-value: < 2.2e-16
> # nasty ves1proc is spelled with a numeral 1

> # to come, propensity estimated by boosted regression (as in twang)
# and partition trees (rpart, cf PSAgraphics)
```

Package 'twang'

February 20, 2015

Version 1.4-9.3

Date 2015-01-30

Title Toolkit for Weighting and Analysis of Nonequivalent Groups

Author Greg Ridgeway, Dan McCaffrey, Andrew Morral, Beth Ann, Lane Burgette <burgette@rand.org>

Maintainer Lane Burgette <burgette@rand.org>

Depends R (>= 2.10), gbm (>= 1.5-3), survey, xtable, lattice, latticeExtra

Description This package offers functions for propensity score estimating and weighting, nonresponse weighting, and diagnosis of the weights. This package was originally developed by Drs. Ridgeway, McCaffrey, and Morral. Burgette, Griffin and McCaffrey updated the package during 2011-2015.

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2015-02-02 15:32:02

Repository/R-Forge/Project twang

Repository/R-Forge/Revision 27

Repository/R-Forge/DateTimeStamp 2015-01-30 14:00:12

R topics documented:

Toolkit for Weighting and Analysis of Nonequivalent Groups: A tutorial for the `twang` package

Greg Ridgeway, Dan McCaffrey, Andrew Morral, Lane Burgette and Beth Ann Griffin*
RAND

January 30, 2015

1 Introduction

The Toolkit for Weighting and Analysis of Nonequivalent Groups, `twang`, contains a set of functions and procedures to support causal modeling of observational data through the estimation and evaluation of propensity scores and associated weights. This package was developed in 2004. After extensive use, it received a major update in 2012. This tutorial provides an introduction to `twang` and demonstrates its use through illustrative examples.

The foundation to the methods supported by `twang` is the propensity score. The propensity score is the probability that a particular case would be assigned or exposed to a treatment condition. Rosenbaum & Rubin (1983) showed that knowing the propensity score is sufficient to separate the effect of a treatment on an outcome from observed confounding factors that influence both treatment assignment and outcomes, provided the necessary conditions hold. The propensity score has the balancing property that given the propensity score the distribution of features for the treatment cases is the same as that for the control cases. While the treatment selection probabilities are generally not known, good estimates of them can be effective at diminishing or eliminating confounds between pretreatment group differences and treatment outcomes in the estimation of treatment effects.

There are now numerous propensity scoring methods in the literature. They differ in how they estimate the propensity score (e.g. logistic regression, CART), the target estimand (e.g. treatment effect on the treated, population treatment effect), and how they utilize the resulting estimated propensity scores (e.g. stratification, matching, weighting, doubly robust estimators). We originally developed the `twang` package with a particular process in mind, namely, generalized boosted regression to estimate the propensity scores and weighting of the comparison cases to estimate the average treatment effect on the treated (ATT). However, we have updated the package to also meaningfully handle the case where interest lies in using the population weights (e.g., weighting of comparison and treatment cases to estimate the population average treatment effect, ATE). The main workhorse of `twang` is the `ps()` function which implements generalized boosted regression modeling to estimate the propensity scores. However, the framework of the package is flexible enough to allow the user to use propensity score estimates from other methods and to assess the usefulness of those estimates for ensuring equivalence (or "balance") in the pretreatment covariate distributions of treatment and control groups using tools from the

*The twang package and this tutorial were developed under NIDA grants R01 DA017507 and R01 DA015697-03

Week 5-- Randomized Experiments and Design Sensitivity