

Full matching in a study of coaching for the SAT (condensed treatment)

Ben Hansen

July 23, 2013

This is a condensed, ever-so-slightly simplified session culminating in a full match discussed and recommended in my paper “Propensity score matching to extract latent experiments from nonexperimental data: A case study” (?). The longer, unsimplified session is available in a similar format.

The match this script leads to combines propensity and prognostic scores in a way that I’ve found reliable and effective to reduce bias while limiting the losses of sample size and precision that can sometimes accompany matching-based analysis.

```
> load("satcoach.RData")
> ###library(optmatch)
> library(optmatch)
> library(RIttools)
> library(xtable)
> library(splines)
> options(digits=3)
> sessionInfo()

R version 3.0.1 (2013-05-16)
Platform: x86_64-w64-mingw32/x64 (64-bit)

locale:
[1] LC_COLLATE=English_United States.1252
[2] LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252
[4] LC_NUMERIC=C
[5] LC_TIME=English_United States.1252

attached base packages:
[1] splines    stats      graphics  grDevices  utils      datasets  methods
[8] base

other attached packages:
[1] xtable_1.7-1  RIttools_0.1-11 optmatch_0.8-3

loaded via a namespace (and not attached):
[1] grid_3.0.1      lattice_0.20-15 SparseM_0.99     tools_3.0.1
```

1 Preliminaries to matching

1.1 the Race \times SES subclassification

The reasons to subclassify prior to matching are that large matching problems can take a lot of time, more time than several smaller matching problems taken together, at the same time that subclassifying

is more intuitive than propensity score matching.

As noted by ?, the matching algorithm `optmatch` is based on uses roughly $O(n^3 \log n)$ floating point operations, where n is the sample size. Comparing this estimate for the problem of matching without subclasses to that of matching withing Race \times SES subclasses gives:

```
> nrow(satcoach)

[1] 3994

> table(satcoach$ethn3levels, satcoach$dadsed)

      (0,5] (5,7] (7,9]  na
Asian    149   102   103   16
White   1026   794   786  273
URM      435   136   120   54

> matchflops <- function(n) n^3*log(n)
> sum(matchflops(table(satcoach$ethn3levels, satcoach$dadsed)))/
+ matchflops(nrow(satcoach))

[1] 0.0279
```

Preceding matching by the Race \times SES subclassification is the same as to match exactly (on propensity scores, Mahalanobis distances, prognostics scores or whatever) within Race \times SES subclasses, which seems a sensible thing to do.

The Race \times SES subclassification does improves balance, as the following calculations show. Matching will bring more satisfying improvements to balance, however, as will be seen below. Significance calculations here are permutation-based, as discussed in (?).

```
> xbs <- xtabs <- printtabs <- list()
> xbs$sc.big <-
+   xBalance(Coach ~
+     psatv + pmin(psatv,40) + pmax(psatv,60) +
+     psatm + pmin(psatm,40) + pmax(psatm,60) +
+     psat.NA +
+     presatv + pmin(presatv,400) + pmax(presatv,600) +
+     presatm + pmin(presatm,400) + pmax(presatm,600) +
+     presat.NA +
+     gpa + avgeng +
+     avgmth + avgnatsci + avgssci + nyrseng + nyrsfl + nyrsmath +
+     nyrsnatsci + nyrsssci +
+     parentsincome + dadsed + momsed +
+     full.1stlang + gender + ethn3levels +
+     Coll1mean +
+     Coll1mean.NA +
+     I23 + I24 + I25 + yearpref + deggoal + pubpripref,
+     strata=list(no.stratification=NULL, racebyeses=~ethn3levels+dadsed),
+     data=satcoach,
+     report=c("adj.mean.diffs", "chisquare.test"))
> ##xbs$sc.big$overall
> xbs$sc.big
```

	strata no.stratification stat	adj.diff	racebyeses adj.diff
vars			
psatv		-1.1e-01	-5.0e-01

pmin(psatv, 40)	9.5e-02	1.8e-01
pmax(psatv, 60)	-2.1e-01 .	-3.7e-01 **
psatm	9.1e-01 *	1.2e-01
pmin(psatm, 40)	2.2e-01 *	2.0e-01 *
pmax(psatm, 60)	1.5e-01	-5.6e-02
psat.NATrue	4.6e-02 *	5.6e-02 *
presatv	-1.7e+00 .	-2.4e+00 *
pmin(presatv, 400)	-2.2e-01	-3.7e-01
pmax(presatv, 600)	-1.6e-01	-2.8e-01
presatm	9.4e-01	5.3e-01
pmin(presatm, 400)	1.8e-02	-1.8e-02
pmax(presatm, 600)	2.2e-01	2.1e-01
presat.NATrue	-5.9e-03	-5.7e-03
gpa(0,1]	-2.6e-02 *	-3.6e-02 **
gpa(1,3]	6.7e-02 **	5.3e-02 *
gpa(3,6]	-4.5e-02 .	-1.0e-02
gpa(6,12]	-2.2e-02	-9.0e-03
gpana	2.6e-02 *	2.3e-03
avgeng(0,1]	3.8e-02	1.8e-02
avgeng(1,5]	-6.5e-02 **	-2.1e-02
avgengna	2.7e-02 *	3.4e-03
avgmath(0,1]	2.5e-02	8.6e-03
avgmath(1,5]	-5.6e-02 *	-1.4e-02
avgmathna	3.1e-02 **	5.6e-03
avgnatsci(0,1]	3.6e-02	1.8e-02
avgnatsci(1,5]	-6.1e-02 *	-2.1e-02
avgnatscina	2.5e-02 *	2.9e-03
avgssci(0,1]	5.1e-02 *	2.9e-02
avgssci(1,5]	-7.3e-02 **	-2.9e-02
avgsscina	2.2e-02 .	-4.7e-04
nyrseng(0,4]	-2.6e-03	8.4e-03
nyrseng(4,6]	-1.4e-02	-9.4e-05
nyrsengna	1.6e-02	-8.3e-03
nyrsfl(0,4]	-1.6e-01 ***	-1.2e-01 ***
nyrsfl(4,6]	1.4e-01 ***	1.3e-01 ***
nyrsflna	1.7e-02	-4.6e-03
nyrsmath(0,4]	-9.0e-02 ***	-5.7e-02 **
nyrsmath(4,6]	6.4e-02 **	5.9e-02 **
nyrsmathna	2.5e-02 *	-2.6e-03
nyrsnatsci(0,4]	-7.0e-02 **	-3.0e-02
nyrsnatsci(4,6]	4.2e-02 .	2.7e-02
nyrsnatscina	2.8e-02 *	3.0e-03
nyrsssci(0,4]	-8.2e-02 ***	-6.7e-02 **
nyrsssci(4,6]	6.9e-02 **	7.8e-02 ***
nyrssscina	1.4e-02	-1.1e-02
parentsincome[1,6]	-1.3e-01 ***	-9.6e-02 ***
parentsincome(6,9]	-8.7e-02 ***	-4.8e-02 *
parentsincome(9,11]	-2.9e-02 .	-3.2e-02 .
parentsincome(11,13]	1.9e-01 ***	1.5e-01 ***
parentsincomea	5.9e-02 ***	3.1e-02 *
dadsed(0,5]	-2.0e-01 ***	-5.0e-17
dadsed(5,7]	-3.5e-02 .	-1.3e-16
dadsed(7,9]	1.9e-01 ***	-8.5e-17
dadsedna	4.1e-02 **	3.1e-17

```

momsed(0,5]                -2.0e-01 ***      -9.9e-02 ***
momsed(5,7]                2.0e-02          9.1e-03
momsed(7,9]                1.3e-01 ***      7.4e-02 ***
momsedna                   4.7e-02 ***      1.5e-02 *
full.1stlang1              -1.1e-01 ***      -7.7e-03
full.1stlang2              3.9e-02 **       6.5e-03
full.1stlang3              3.5e-02 **       -7.7e-03
full.1stlangna             3.5e-02 **       8.9e-03
genderB                    -1.1e-02          -3.1e-02
genderG                    1.1e-02          3.1e-02
ethn3levelsAsian           1.1e-01 ***      -6.6e-18
ethn3levelsWhite           -1.1e-01 ***      -4.8e-16
ethn3levelsURM             -5.2e-03          -6.5e-17
Coll1mean                  3.9e+01 ***      3.0e+01 ***
Coll1mean.NATrue           4.8e-03          1.0e-02
I230                       -4.3e-02 .       -3.1e-02
I231                       -7.3e-02 ***      -7.1e-02 ***
I232                       1.2e-01 ***      1.0e-01 ***
I240                       2.9e-02          3.5e-02 .
I241                       8.5e-02 ***      8.0e-02 ***
I242                       -4.6e-02 *       -5.0e-02 *
I243                       -6.8e-02 ***      -6.5e-02 ***
I250                       2.7e-02          3.3e-02
I251                       3.9e-02 .       2.7e-02
I252                       -6.6e-02 ***      -5.9e-02 ***
yearpref0                  -9.0e-03          -2.7e-02 *
yearpref1                  9.0e-03          2.7e-02 *
deggoal0                   -1.7e-02          -2.4e-02
deggoal1                   -1.4e-02 *       -9.2e-03
deggoal2                   -7.7e-02 ***      -5.1e-02 **
deggoal3                   1.1e-01 ***      8.4e-02 ***
pubpripref0                1.2e-01 ***      8.6e-02 ***
pubpripref1               -1.2e-01 ***      -8.6e-02 ***
---Overall Test---
                                chisquare df p.value
no.stratification          486 66 3.2e-65
racebyses                  287 61 5.3e-31
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

```

So the stratification helps, mostly, but observed balance is still far worse than it would have been under random assignment.

1.2 Estimating the propensity score

Here is an estimated propensity score. To somewhat relax the assumption of linearity of the propensity score in the 5 measurement variables, we expand each of them in natural splines.

```

> ppty <- list()
> ppty$fmla <- Coach ~ ns(psatv,df=4) + ns(psatm,df=4) + ns(presatv, df=4) +
+ ns(presatm, df=4) + ns(Coll1mean, df=4) + I23 +
+ I24 + I25 + parentsincome + dadsed + momsed + clpsed.ethn +
+ full.1stlang + yearpref + deggoal + pubpripref + gpa + avgeng +
+ avgmth + avgnatsci + avgssci + nyrseng + nyrsfl + nyrmth +
+ nyrsnatsci + nyrsosci + gender + ethn3levels +

```

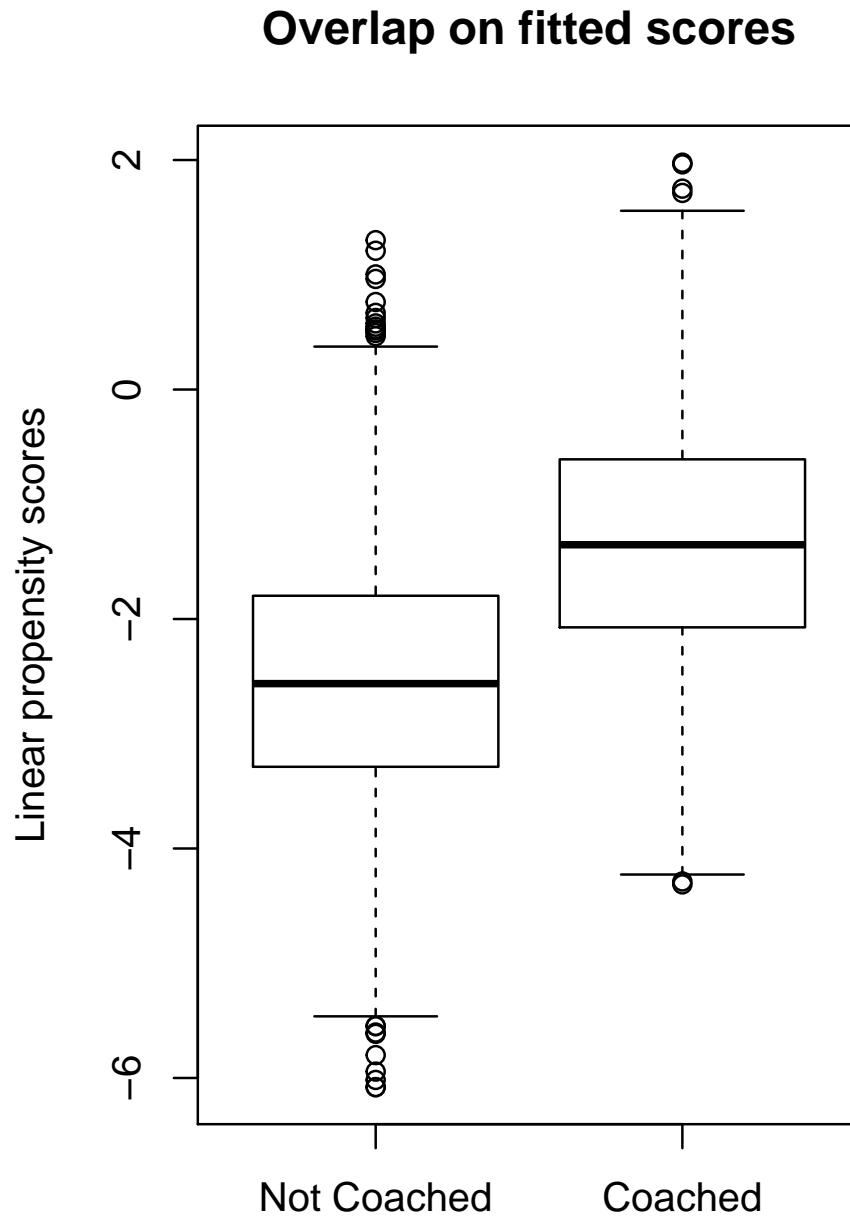
```

+   psat.NA + presat.NA + Collmean.NA
> ppty$model <- glm(ppty$fmla, data=satcoach, family=binomial)
> ppty$score <- ppty$model$linear.predictors
> all.equal(names(ppty$score), row.names(satcoach))

[1] TRUE

> boxplot(ppty$model, names=c("Not Coached", "Coached"),
+         ylab="Linear propensity scores", xlab="", varwidth = TRUE)

```



2 Full matching on the propensity score

To match within Race×SES subclasses, we only need to calculate discrepancies on the propensity score (and/or on whatever else we may decided to match on) within these subclasses.

First we create a block of possible matches.

```
> ppty$blocks <- exactMatch(Coach ~ ethn3levels + dadsed,data=satcoach)
```

This incidentally reduces the burden of the problem on memory: matching within subclasses involves calculating and storing 12 distance matrices, with 230000 entries in total, whereas matching without subclasses requires us to store and calculate one large 500×3494 matrix, with 1750000 entries.

2.1 Propensity score full matching

Plain-vanilla full matching on the propensity score (within Race×SES subclasses).

```
> #fm.ppty <- fullmatch(ppty$model, within=ppty$blocks, data=satcoach)
> #will replace following 2 lines
> ppty$dist<-match_on(ppty$model, within=ppty$blocks, data=satcoach)
> fm.ppty<-fullmatch(ppty$dist,data=satcoach)
> summary(fm.ppty,ppty$model)
```

Structure of matched sets:

```
5+:1  4:1  3:1  2:1  1:1  1:2  1:3  1:4  1:5+
      3    3    4   15  101   49   42   41  193
```

Effective Sample Size: 680

(equivalent number of matched pairs).

```
sum(matched.distances)=328
```

(within 2.22 of optimum).

Percentiles of matched distances:

```
0%   50%   95%  100%
0.000 0.026 0.404 2.380
```

Balance test overall result:

```
chisquare df p.value
      16.4  69      1
```

```
> stratumStructure(fm.ppty, min.controls=0.2, max.controls=5)
```

```
5+:1  4:1  3:1  2:1  1:1  1:2  1:3  1:4  1:5+
      3    3    4   15  101   49   42   41  193
```

```
attr("comparable.num.matched.pairs")
```

```
[1] 680
```

Balance looks good, but there are outliers in terms of matched distances on the propensity score. This is something to avoid (?). Calipers are the natural remedy:

```
> #fm.ppty.clpr050 <- fullmatch(ppty$model, within=ppty$dist, data=satcoach, caliper=0.5)
> ppty$dist.clpr050 <-match_on(ppty$model, within=ppty$blocks, caliper=0.5, data=satcoach)
> fm.ppty.clpr050 <- fullmatch(ppty$dist.clpr050, data=satcoach)
> summary(fm.ppty.clpr050,ppty$model)
```

Matching failed in subclasses containing 150 of 3994 observations.

Reporting on subclasses where matching worked. (Enter ?matchfailed for more info.)

Structure of matched sets:

```
5+:1  4:1  3:1  2:1  1:1  1:2  1:3  1:4  1:5+
      2    4    3   15   97   51   39   44  193
```

Effective Sample Size: 676
(equivalent number of matched pairs).

```
sum(matched.distances)=199
(within 2.23 of optimum).
Percentiles of matched distances:
      0%    50%    95%   100%
0.0000 0.0239 0.2330 0.4990
Balance test overall result:
      chisquare df p.value
      11.3    69      1

> stratumStructure(fm.ppty.clpr050,min.controls=0.2, max.controls=5)

      1:0 5+:1  4:1  3:1  2:1  1:1  1:2  1:3  1:4 1:5+  0:1
      10   2   4   3  15  97  51  39  44 193 140
attr(,"comparable.num.matched.pairs")
[1] 676
```

Note that the addition of this caliper has caused some treatment group members to go unmatched — these are the “1:0” matched sets noted in the summary. (`fullmatch(<...>)` and `pairmatch(<...>, remove.unmatchables=T)` automatically remove subjects without permissible matches before they begin to match. That’s what’s happened to these treatment group members, as well as to the controls falling in 0:1 “matched sets.”)

3 Full matching with prognostic and propensity scores

Here’s another way to manage the tradeoff between bias reduction and effective sample size, one that can be operationally simpler and might offer some additional bias protection as well: matching one propensity score within calipers of another, constructing the first propensity score in such a way as to keep to a minimum separation between coached and uncoached kids while constructing the second in such a way as to promote balance on as many covariates as possible. A natural way to get less-separation propensity scores is to permit in them only covariates or functions of the covariate that are most predictive of the response.

3.1 A focused propensity score

Extracting summaries of the part of the covariate other than pretests, as it relates to pretest score. Because these summaries are inherently one-dimensional, because OLS is not necessarily the best approach, and because we’re skipping any diagnostics, they are potentially rather crude. I have set to ‘NA’ early SAT scores or PSAT scores from test sittings after a student was coached, as well as prior test scores of uncoached students from sittings close in time to their sitting for the posttest, as discussed in (?, § 1.2). This helps to ensure that the pretest genuinely anteceded the treatment, and improves motivation of models in which assignment to treatment occurs strictly after (what we are considering to be) pretests.

```
> pg.mod <- list()
> (pg.mod$m.fmla <- update.formula(ppty$fmla, postsatmath~.-ethn3levels))

postsatmath ~ ns(psatv, df = 4) + ns(psatm, df = 4) + ns(presatv,
df = 4) + ns(presatm, df = 4) + ns(Coll1mean, df = 4) + I23 +
I24 + I25 + parentsincome + dadsged + momsged + clpsed.ethn +
full.1stlang + yearpref + deggoal + pubpripref + gpa + avgeng +
avgmath + avgnatsci + avgssci + nyrseng + nyrsfl + nyrsmath +
nyrsnatsci + nyrsnsci + gender + psat.NA + presat.NA + Coll1mean.NA
```

```

> (pg.mod$v.fmla <- update.formula(ppty$fmla, postsatverb~.-ethn3levels))

postsatverb ~ ns(psatv, df = 4) + ns(psatm, df = 4) + ns(presatv,
  df = 4) + ns(presatm, df = 4) + ns(Coll1mean, df = 4) + I23 +
  I24 + I25 + parentsincome + dadsed + momsed + clpsed.ethn +
  full.1stlang + yearpref + deggoal + pubpripref + gpa + avgeng +
  avgmth + avgnatsci + avgssci + nyrseng + nyrsfl + nyrsmath +
  nyrsnatsci + nyrsosci + gender + psat.NA + presat.NA + Coll1mean.NA

> ###pg.mod$basicfmla <- ~ Coll1mean + I23 +
> ### I24 + I25 + parentsincome + dadsed + momsed + clpsed.ethn +
> ### full.1stlang + yearpref + deggoal + pubpripref + gpa + avgeng +
> ### avgmth + avgnatsci + avgssci + nyrseng + nyrsfl + nyrsmath +
> ### nyrsnatsci + nyrsosci + gender + Coll1mean.NA
>
> pg.mod$v <-
+ lm(pg.mod$v.fmla, #update.formula(pg.mod$basicfmla, psatv~.),
+ data=satcoach, subset=!Coach#!psat.NA
+ )
> pg.mod$m <-
+ lm(pg.mod$m.fmla, #update.formula(pg.mod$basicfmla, psatm~.),
+ data=satcoach, subset=!Coach#!psat.NA
+ )
> pg.mod$scorenames <-
+ c(psatv, psatm, presatv, presatm, psat.NA, presat.NA)
> pg.mod$scores <-
+ data.frame(satcoach[c(Coach, pg.mod$scorenames)],
+            pg.v=predict(pg.mod$v, satcoach),
+            pg.m=predict(pg.mod$m, satcoach),
+            check.rows=T)
> pg.ppty <- list()
> pg.ppty$fmla <- Coach ~ psatv + psatm + presatv + presatm +
+ psat.NA + presat.NA + pg.v + pg.m
> (pg.ppty$model <-
+ glm(pg.ppty$fmla,
+ data=pg.mod$scores, family=binomial) )

Call: glm(formula = pg.ppty$fmla, family = binomial, data = pg.mod$scores)

```

Coefficients:

(Intercept)	psatv	psatm	presatv	presatm
0.10274	-0.06651	-0.01100	-0.01103	0.00339
psat.NATrue	presat.NATrue	pg.v	pg.m	
0.68817	0.55884	0.00559	0.00359	

Degrees of Freedom: 3993 Total (i.e. Null); 3985 Residual

Null Deviance: 3010

Residual Deviance: 2950 AIC: 2970

Here's how the ordinary propensity and the prognostic propensity differ in terms of how they separate treatment and control groups.

```

> pg.ppty$score <- pg.ppty$model$linear.predictor
> all.equal(names(pg.ppty$score), row.names(satcoach))

```

```
[1] TRUE
```



```

> par(mfrow=c(1,2))
> boxplot(ppty$model, names=c("Not coached", "Coached"),
+ ylab="Ordinary propensity scores", xlab="", varwidth = TRUE, ylim=range(ppty$score))
> boxplot(pg.ppty$model, names=c("Not coached", "Coached"),
+ ylab="Focused propensity scores", xlab="", varwidth = TRUE,
+ ylim=range(ppty$score))
> par(mfrow=c(1,1))
> lm1 <- lm(ppty$score~Coach, data=satcoach)
> (ppty$meansep <- coef(lm1)[Coach]/sqrt(mean(residuals(lm1)^2)))

```

Coach

1.11

```

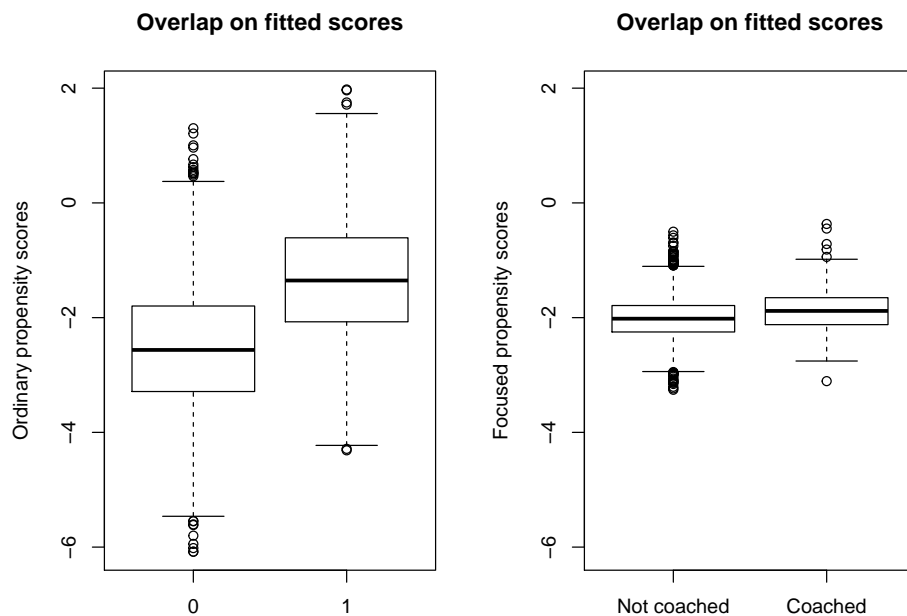
> lm1 <- lm(pg.ppty$score~Coach, data=satcoach)
> (pg.ppty$meansep <- coef(lm1)[Coach]/sqrt(mean(residuals(lm1)^2)))

```

Coach

0.376

```
> rm(lm1)
```



3.2 Full matching on propensity and prognostic propensity scores

Distance on the prognostic score (to begin with, without calipers). Note the complaint from the distance-making functions about the stratifying variables not being in the propensity specification. Given that they are included in the ordinary propensity score, and we'll be putting in a caliper on that, it's nothing to worry about.

```

> pg.ppty$blocks <- exactMatch(Coach ~ ethn3levels + dadsed, data=satcoach)
> pg.ppty$dist <- match_on(pg.ppty$model, within=pg.ppty$blocks, data=satcoach)
> pg.ppty$dist.with.ppty.calipers <- caliper(pg.ppty$dist, width=0.5)
> fm.pg.ppty.ppty.clpr <- fullmatch(pg.ppty$dist.with.ppty.calipers, data=satcoach)

```

```
> #fm.pgpty.pptyclpr <- fullmatch(pg.ppty$model, within=pg.ppty$blocks, data=satcoach, caliper=0.5)
> #previous will replace pg.ppty$dist and pg.ppty$dist.with.ppty.calipers and fm.pgpty.pptyclpr
>
```

3.3 Assessing the match

Summary, including summary of balance on all variables contributing to the (ordinary) propensity score.

```
> summary(fm.pgpty.pptyclpr, ppty$model)
```

```
Matching failed in subclasses containing 131 of 3994 observations.
Reporting on subclasses where matching worked. (Enter ?matchfailed for more info.)
Structure of matched sets:
  1:1  1:2  1:3  1:4 1:5+
 384  17  14   7  73
Effective Sample Size: 575
(equivalent number of matched pairs).
```

```
sum(matched.distances)=0
(within 0.488 of optimum).
Percentiles of matched distances:
  0%  50%  95% 100%
   0    0    0    0
Balance test overall result:
  chisquare df  p.value
    196 69 3.87e-14
```

```
> stratumStructure(fm.pgpty.pptyclpr,min.controls=0.2,max.controls=5)

  1:0  1:1  1:2  1:3  1:4 1:5+  0:1
    5 384  17  14   7  73 126
attr(,"comparable.num.matched.pairs")
[1] 575
```

Plot of balance on variables contributing to focused/prognostic propensity score.

```
> (xb <-
+ xBalance(Coach ~ pg.v + pg.m + psat.NA + psatv + psatm +
+           presat.NA + presatv + presatm,
+           strata=list(racebySES=
+             ~interaction(satcoach$dadsed,satcoach$ethn3levels),
+             matched=~fm.pgpty.pptyclpr),
+           report=c("std.diffs", "p.values", "chisquare.test"),
+           data=pg.mod$scores)
+ )
```

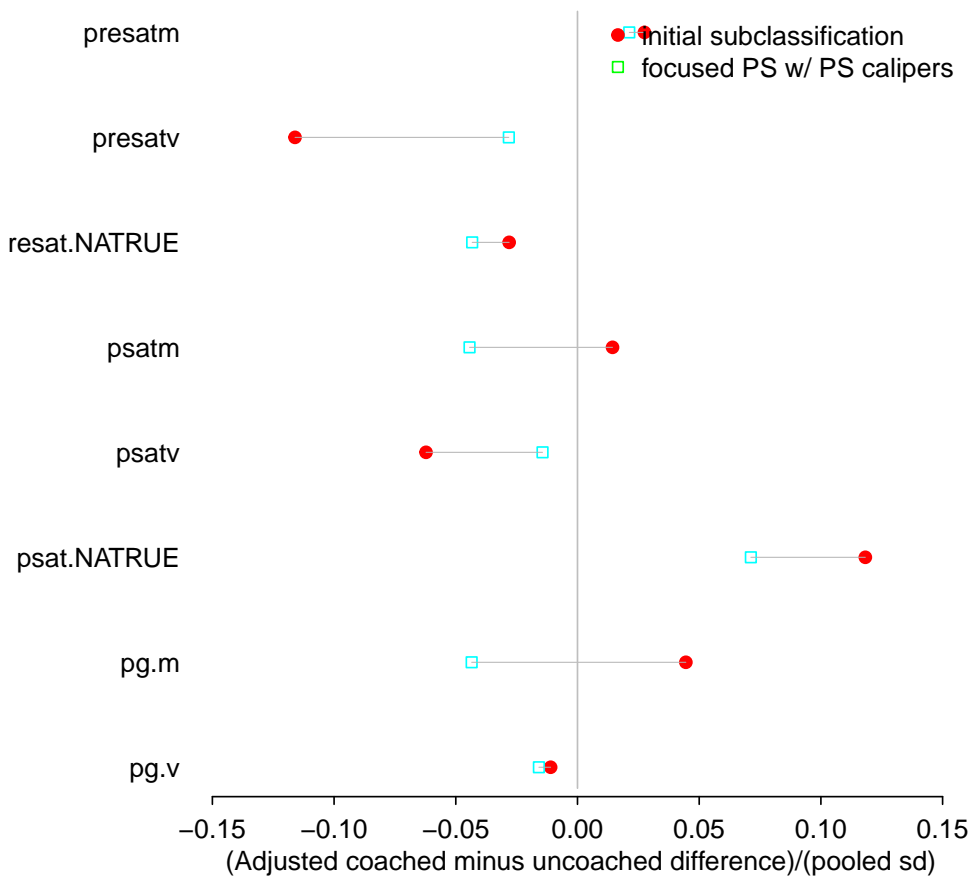
	strata racebySES stat	std.diff	matched std.diff
vars			
pg.v		-0.011	-0.016
pg.m		0.045	-0.043
psat.NATRUE		0.118 *	0.071
psatv		-0.062	-0.014
psatm		0.014	-0.044
presat.NATRUE		-0.028	-0.043
presatv		-0.116 *	-0.028

```

presatm          0.028          0.021
---Overall Test---
      chisquare df p.value
racebySES      47  8 1.4e-07
matched        21  8 7.5e-03
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

> plot(xb, legend=FALSE,
+       thexlab="(Adjusted coached minus uncoached difference)/(pooled sd)")
> legend(x=topright,
+       legend=c("initial subclassification",
+               "focused PS w/ PS calipers"),
+       col=rainbow(3), pch=c(19,22,23), bty=n )
>

```



4 Outcome Analysis

4.1 Matched permutation tests without covariance adjustment

Here is a formula that will be used to adjust the outcome vector for various hypothesized treatment effects.

```
> verbfmla <-
+ paste("Coach ~", paste(paste("I(postsatverb -", -15:35, "*Coach)"), collapse="+"))
> verbfmla <- as.formula(verbfmla)
```

p-values, a Hodges-Lehmann point estimate and confidence intervals.

```
> pvals.verb <- xBalance(verbfmla, strata=fm.pgpty.pptyclpr, data=satcoach,
+                         report=c("p.values"))$results[,p,1]
> names(pvals.verb) <- -15:35
> (-15:35)[pvals.verb==max(pvals.verb)]

[1] 0

> range((-15:35)[pvals.verb>=.05])
```

```
[1] -11 12
```

Similarly for math effects:

```
> mathfmla <-
+ paste("Coach ~", paste(paste("I(postsatmath -", -5:45, "*Coach)"), collapse="+"))
> mathfmla <- as.formula(mathfmla)
> pvals.math <- xBalance(mathfmla, strata=fm.pgpty.pptyclpr, data=satcoach,
+                         report=c("p.values"))$results[,p,1]
> names(pvals.math) <- -5:45
> (-5:45)[pvals.math==max(pvals.math)]

[1] 19

> range((-5:45)[pvals.math>=.05])

[1] 8 29

>
```

4.2 Matched permutation tests with Rosenbaum-type covariance adjustment

Here is a data frame consisting of sat-verbal responses adjusted, first, for one of 51 hypothesised treatment effects; and, second, adjusted for covariance with pretests. From the perspective of any of the 51 null hypotheses, after the two adjustments residuals ought to be permutable within matched sets, and a permutation test is appropriate. The RIttools function `xBalance()` is used to provide Normal approximations to these permutation tests.

Because we first introduce offsets for hypothesized treatment effects and then perform covariance adjustment, on the entire data set, I call this ? type covariance adjustment.

```
> fmla.lhs <- ~ psatv + psatm + psat.NA + presatv + presatm + presat.NA #+
> #fm.pgpty.pptyclpr
>
> adj.vresp <-
+ lapply(-15:35, function(tau) {
+ fmla <- as.formula(paste("I(postsatverb -", tau, "*Coach)~.", collapse=""))
+ fmla <- update.formula(fmla.lhs,
+                         fmla)
+ residuals(lm(fmla, data=satcoach))
+ })
> adj.vresp <- as.data.frame(adj.vresp)
> names(adj.vresp) <- paste(e, 1:length(adj.vresp), sep="")
```

Now to compute and extract the confidence interval.

```
> xbfmla.rhs <- paste(e, 1:length(adj.vresp), sep="")
> xbfmla.rhs <- as.formula(paste("~", paste(xbfmla.rhs, collapse="+")))
> adj.vresp$Coach <- satcoach$Coach#[matched(fm.pgpty.pptyclpr)]
> pvals.verb <-
+   xBalance(update.formula(xbfmla.rhs, Coach ~.),
+             strata=fm.pgpty.pptyclpr,#[matched(fm.pgpty.pptyclpr)],
+             data=adj.vresp,
+             report=c("p.values"))$results[,p,1]
> names(pvals.verb) <- -15:35
> (-15:35)[pvals.verb==max(pvals.verb)]

[1] 4

> range((-15:35)[pvals.verb>=.05])

[1] -4 11

>
```

Now to do the same for math.

```
> adj.mresp <-
+ lapply(-5:45, function(tau) {
+ fmla <- as.formula(paste("I(postsatmath ~", tau, "*Coach)~.", collapse=""))
+ fmla <- update.formula(fmla.lhs,
+                       fmla)
+ residuals(lm(fmla, data=satcoach))
+ })
> adj.mresp <- as.data.frame(adj.mresp)
> names(adj.mresp) <- paste(e, 1:length(adj.mresp), sep="")
> adj.mresp$Coach <- satcoach$Coach#[matched(fm.pgpty.pptyclpr)]
> pvals.math <-
+   xBalance(update.formula(xbfmla.rhs, Coach ~.),
+             strata=fm.pgpty.pptyclpr,#[matched(fm.pgpty.pptyclpr)],
+             data=adj.mresp,
+             report=c("p.values"))$results[,p,1]
> names(pvals.math) <- -5:45
> (-5:45)[pvals.math==max(pvals.math)]

[1] 23

> range((-5:45)[pvals.math>=.05])

[1] 15 30

>
```