

# Tutorial: Matching Methods for Causal Inference

Kwonsang Lee

2022-06-23

## Contents

<b>1 Introduction: Short summary of matching</b>	<b>1</b>
<b>2 NSW Experiment Data + Survey Data (CPS)</b>	<b>2</b>
2.1 Recall the NSW experiment data . . . . .	2
2.2 Compared to the survey data . . . . .	2
2.3 Estimating the propensity score $e(x)$ . . . . .	3
<b>3 Optimal Pair Matching</b>	<b>4</b>
3.1 Using the propensity score Only (or PSM) . . . . .	4
3.2 Using the rank-based Mahalanobis distance Within ps calipers . . . . .	6
3.3 Using the Mahalanobis distance within ps calipers with near-exact matching for black . . . . .	8
<b>4 When there are covariates that are hard to match on?</b>	<b>9</b>
4.1 Optimal full matching . . . . .	9
4.2 Separate optimal pair matching for black and non-black . . . . .	11
<b>5 Analysis Stage</b>	<b>12</b>
5.1 Using randomization inference . . . . .	12
5.2 Using an outcome model . . . . .	14
<b>6 Cardinality matching</b>	<b>16</b>
<b>7 A Bit More: separate matching + cardinality</b>	<b>17</b>
<b>8 Reference</b>	<b>21</b>

## 1 Introduction: Short summary of matching

The procedure for matching consists of design and analysis stages. During the design stage, the goal is to create matched pairs/sets so that matched subjects are sufficiently close enough in terms of their covariates. This stage consists of several tasks: (i) estimating the propensity score, (ii) choosing appropriate matching methods, and (iii) assessing covariate balance. We do not need the outcome  $Y$  in this stage. We use only the treatment indicator  $Z$  and covariates  $X$ . Doing so, transparency can be ensured. The analysis stage is quite simple. Based on the matched pairs/sets, we conduct hypothesis testing or estimate the treatment effects. Inferences can be made based on randomization inference as we covered, or we can use a weighted regression model with matching weights. Some practical issues are listed below:

- Variable selection for the propensity score model. See VanderWeele (2019).
- Nearest neighbor matching or optimal matching?
- Are there any covariates more important than the others? Do we need to match exactly on these?
- Use randomization inference or not?

The **MatchIt** package will be used. More details can be found at <https://kosukeimai.github.io/MatchIt/>. For general introductions to matching, see Stuart (2010).

## 2 NSW Experiment Data + Survey Data (CPS)

### 2.1 Recall the NSW experiment data

We examine the effect of the job training program on later earnings using the NSW experiment data. During this lecture, we used 185 treated and 185 control male subjects to evaluate the effect. Using Wilcoxon's signed rank statistic, we obtained the 95% confidence interval [\$391, \$2894] with the Hodges-Lehmann estimate \$1640.

```
library(DOS2)
## checking covariate balance between treated and control subjects.
summary.nsw.exp = aggregate(NSW[,3:10], list(NSW$z), FUN = mean)
colnames(summary.nsw.exp)[1] = "Treatment"
summary.nsw.exp

##   Treatment    age    edu    black    hisp    married    nodegree    re74
## 1         0 25.70270 10.19459 0.8540541 0.05405405 0.2000000 0.7675676 2009.448
## 2         1 25.81622 10.34595 0.8432432 0.05945946 0.1891892 0.7081081 2095.574
##      re75
## 1 1485.058
## 2 1532.055

## Wilcoxon's signed rank statistic
wilcox.test(NSW$re78[NSW$z==1] - NSW$re78[NSW$z==0], conf.int = T)

##
## Wilcoxon signed rank test with continuity correction
##
## data: NSW$re78[NSW$z == 1] - NSW$re78[NSW$z == 0]
## V = 9025, p-value = 0.009934
## alternative hypothesis: true location is not equal to 0
## 95 percent confidence interval:
##  391.359 2893.626
## sample estimates:
## (pseudo)median
##      1639.558

# wilcox.test(NSW$re78[NSW$z==1], NSW$re78[NSW$z==0], conf.int = T, paired = T)
```

### 2.2 Compared to the survey data

Now, we're going to examine the effect using the same 185 treated male subjects and other 429 controls. The 429 subjects are obtained from nonrandomized survey, CPS. Let's see how different they are compared to the treated subjects. Can we draw a similar conclusion as we did using the randomized experiment?

```
library(MatchIt)

lalonde$black = (lalonde$race == "black")
lalonde$hispan = (lalonde$race == "hispan")
summary.nsw.obs = aggregate(lalonde[, c(2,3,10,11,5,6,7,8)], list(lalonde$treat), FUN = mean)
colnames(summary.nsw.obs)[c(1,3)] = c("Treatment", "edu")
summary.nsw = rbind(summary.nsw.obs[1,], summary.nsw.exp)
summary.nsw[,1] = c("obs_control", "exp_control", "treated")
```

```
summary.nsw
```

```
##      Treatment      age      edu      black      hisp      married      nodegree
## 1 obs_control 28.03030 10.23543 0.2027972 0.14219114 0.5128205 0.5967366
## 2 exp_control 25.70270 10.19459 0.8540541 0.05405405 0.2000000 0.7675676
## 3      treated 25.81622 10.34595 0.8432432 0.05945946 0.1891892 0.7081081
##      re74      re75
## 1 5619.237 2466.484
## 2 2009.448 1485.058
## 3 2095.574 1532.055
```

Our goal is to find control subjects who are comparable to the treated subjects, and then, make inferences. The control reservoir is larger than the treated reservoir. We can start with optimal pair matching. To illustrate several matching techniques, we will use the **MatchIt** package.

### 2.3 Estimating the propensity score $e(x)$

We use the propensity score model proposed by Dehejia and Wahba (1999). It uses additional terms for age and education.

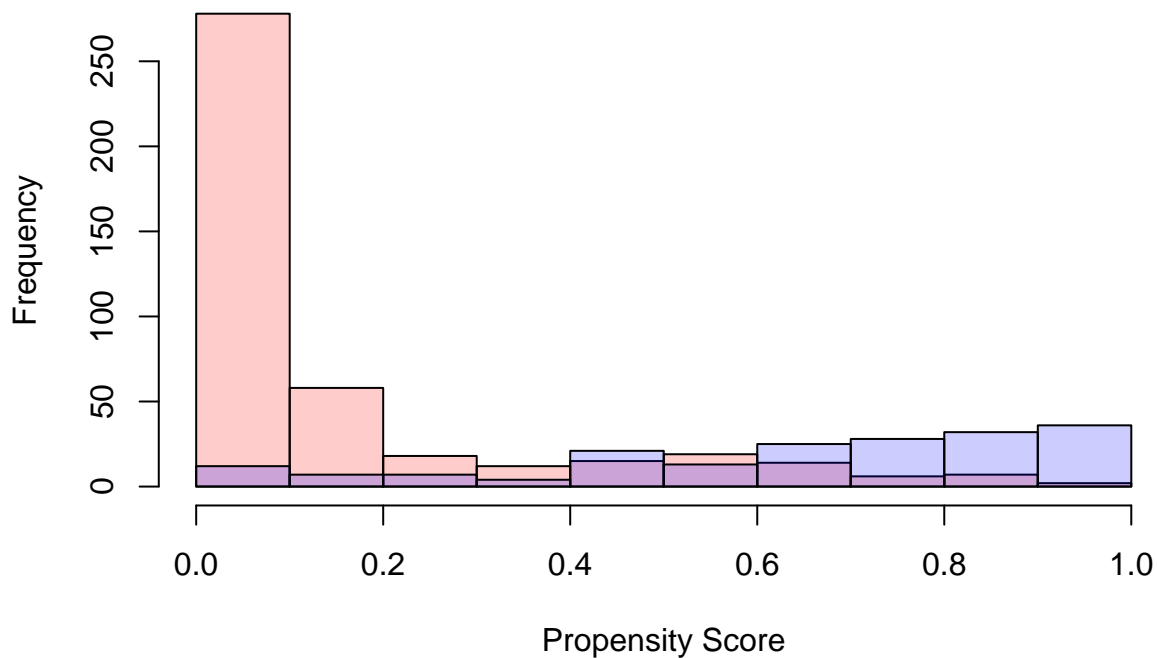
```
lalonde$age2 = (lalonde$age - mean(lalonde$age))^2
lalonde$age3 = (lalonde$age - mean(lalonde$age))^3
lalonde$educ2 = (lalonde$educ - mean(lalonde$educ))^2

## Propensity score estimation
ps.fit = glm(treat ~ age + age2 + age3 + educ + educ2 + black + hisp +
             married + nodegree + re74 + re75, family=binomial, x=T, data=lalonde)
summary(ps.fit)
```

```
##
## Call:
## glm(formula = treat ~ age + age2 + age3 + educ + educ2 + black +
##      hisp + married + nodegree + re74 + re75, family = binomial,
##      data = lalonde, x = T)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -2.4529  -0.4498  -0.2289   0.4115   2.6899
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.110e+00  1.432e+00  -1.473  0.14074
## age          7.431e-02  2.780e-02   2.673  0.00751 **
## age2        -1.731e-02  2.969e-03  -5.829 5.59e-09 ***
## age3         3.842e-04  1.506e-04   2.551  0.01073 *
## educ        -5.117e-02  9.194e-02  -0.557  0.57782
## educ2       -4.789e-02  1.608e-02  -2.977  0.00291 **
## blackTRUE    3.018e+00  3.058e-01   9.869 < 2e-16 ***
## hispTRUE     6.067e-01  4.617e-01   1.314  0.18876
## married     -1.497e+00  3.280e-01  -4.563 5.03e-06 ***
## nodegree     5.601e-01  4.229e-01   1.324  0.18540
## re74        -9.366e-05  3.202e-05  -2.925  0.00344 **
## re75         2.331e-05  4.814e-05   0.484  0.62818
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 751.49 on 613 degrees of freedom
## Residual deviance: 420.37 on 602 degrees of freedom
## AIC: 444.37
##
## Number of Fisher Scoring iterations: 6
est.ps = predict(ps.fit, type = "response")
hist(est.ps[lalonde$treat==0], col = rgb(1,0,0,0.2), xlab = "Propensity Score", main = "Treated (blue) vs. Control (red)", add = T)
hist(est.ps[lalonde$treat==1], col = rgb(0,0,1,0.2), add = T)
```

**Treated (blue) vs. Control (red)**



### 3 Optimal Pair Matching

#### 3.1 Using the propensity score Only (or PSM)

We use the estimated propensity score  $e(x)$  and proceed the optimal matching procedure. To do so, we need to specify the distance matrix between treated and control subjects. We use the absolute difference, i.e.,  $|e(x_t) - e(x_c)|$ . Other distances can be used; e.g., squared difference  $(e(x_t) - e(x_c))^2$ .

```
library(optmatch)
ps.dist = match_on(est.ps, z=lalonde$treat)

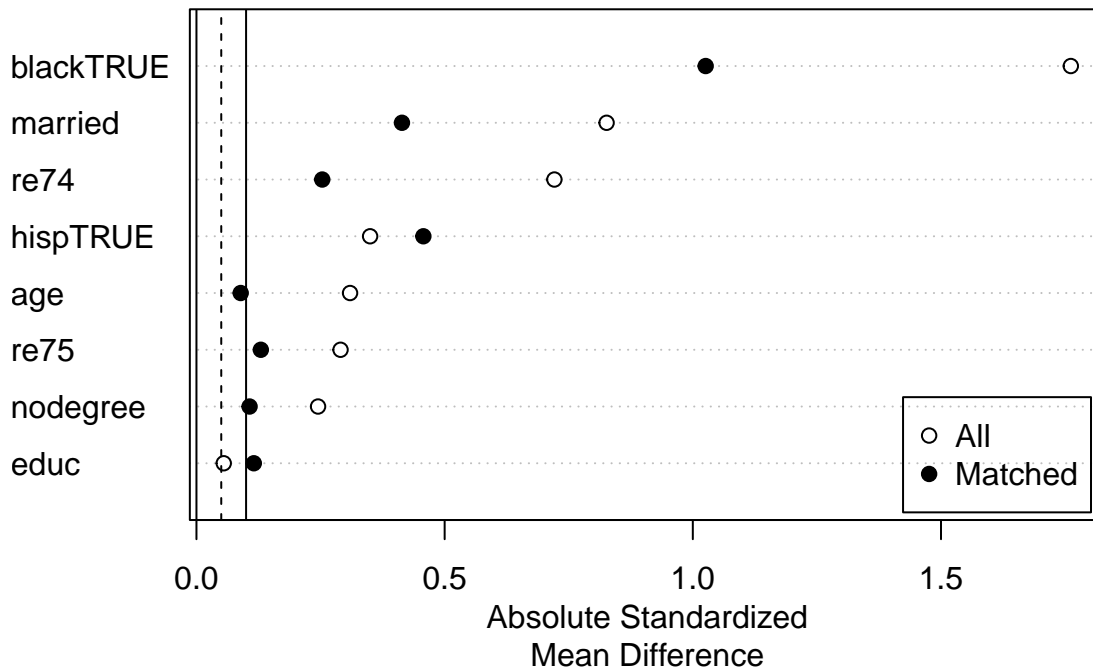
psm.out = matchit(treat ~ age + educ + black + hisp + married + nodegree + re74 + re75,
                  data = lalonde, method= "optimal", distance = ps.dist)
# If you don't have a specific ps model in mind, then you could set the distance automatically.
# psm.out = matchit(treat ~ age + educ + black + hisp + married + nodegree + re74 + re75,
#                   data = lalonde, method= "optimal", distance = "glm")
summary(psm.out)
```

```

##
## Call:
## matchit(formula = treat ~ age + educ + black + hisp + married +
##         nodegree + re74 + re75, data = lalonde, method = "optimal",
##         distance = ps.dist)
##
## Summary of Balance for All Data:
##           Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## age           25.8162      28.0303      -0.3094      0.4400      0.0813
## educ           10.3459      10.2354       0.0550      0.4959      0.0347
## blackTRUE       0.8432       0.2028       1.7615       .           0.6404
## hispTRUE        0.0595       0.1422      -0.3498       .           0.0827
## married         0.1892       0.5128      -0.8263       .           0.3236
## nodegree        0.7081       0.5967       0.2450       .           0.1114
## re74            2095.5737     5619.2365     -0.7211      0.5181      0.2248
## re75            1532.0553     2466.4844     -0.2903      0.9563      0.1342
##           eCDF Max
## age           0.1577
## educ           0.1114
## blackTRUE     0.6404
## hispTRUE      0.0827
## married       0.3236
## nodegree      0.1114
## re74          0.4470
## re75          0.2876
##
##
## Summary of Balance for Matched Data:
##           Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## age           25.8162      26.4541     -0.0891      0.6723      0.0392
## educ           10.3459      10.1135      0.1156      0.6279      0.0265
## blackTRUE       0.8432       0.4703       1.0259       .           0.3730
## hispTRUE        0.0595       0.1676     -0.4572       .           0.1081
## married         0.1892       0.3514     -0.4140       .           0.1622
## nodegree        0.7081       0.6595      0.1070       .           0.0486
## re74            2095.5737     3333.9943     -0.2534      1.1565      0.1232
## re75            1532.0553     1949.6322     -0.1297      1.1638      0.0708
##           eCDF Max Std. Pair Dist.
## age           0.0973           1.1272
## educ           0.0919           1.2340
## blackTRUE     0.3730           1.2043
## hispTRUE      0.1081           0.7772
## married       0.1622           0.9937
## nodegree      0.0486           1.0106
## re74          0.3946           0.8468
## re75          0.1946           0.8166
##
## Sample Sizes:
##           Control Treated
## All           429      185
## Matched       185      185
## Unmatched     244       0
## Discarded      0       0

```

```
plot(summary(psm.out), var.order = "unmatched")
```



The covariate balance table is shown before and after matching. We need to closely look at the column of **Std. Mean Diff**, standardized mean difference (SMD). A rule of thumb is making the SMD smaller than 0.1. We say 0.2 is acceptable, but 0.1 is preferable. As shown in the table, the balance is quite bad. Some covariates achieve good balance, but black is not balanced at all.

### 3.2 Using the rank-based Mahalanobis distance Within ps calipers

Let's try the rank-based Mahalanobis distance within propensity score (ps) caliper. There are several options to compute this distance, but we recommend to use the `match_on` function in the `optmatch` package or the `smahal` function in the `DOS2` package.

A shortcut without computing the distance is to use the distance option `mahalanobis` or `robust_mahalanobis` in the `matchit` function in the `MatchIt` package. However, in this setting, ps caliper cannot be specified.

```
mc.out = matchit(treat ~ age + educ + black + hisp + married + nodegree + re74 + re75,
                 data = lalonde, method= "optimal", distance = "robust_mahalanobis")
summary(mc.out, un = F)
```

```
##
## Call:
## matchit(formula = treat ~ age + educ + black + hisp + married +
##         nodegree + re74 + re75, data = lalonde, method = "optimal",
##         distance = "robust_mahalanobis")
##
## Summary of Balance for Matched Data:
##           Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## age           25.8162      26.9622      -0.1602      0.4640  0.0768
## educ           10.3459      10.3351       0.0054      0.5622  0.0239
## blackTRUE       0.8432       0.4595       1.0556      .      0.3838
## hispTRUE        0.0595       0.0649      -0.0229      .      0.0054
## married         0.1892       0.3189      -0.3312      .      0.1297
## nodegree        0.7081       0.6649       0.0951      .      0.0432
```

```
## re74          2095.5737    2731.7476         -0.1302    1.1238    0.0734
## re75          1532.0553    1713.2806         -0.0563    1.2924    0.0478
##              eCDF Max Std. Pair Dist.
## age           0.1514         0.6029
## educ          0.0703         0.3280
## blackTRUE     0.3838         1.0556
## hispTRUE      0.0054         0.0686
## married       0.1297         0.3312
## nodegree      0.0432         0.0951
## re74          0.3351         0.3647
## re75          0.1838         0.3801
##
## Sample Sizes:
##              Control Treated
## All           429      185
## Matched       185      185
## Unmatched     244       0
## Discarded     0        0
```

Instead, we can compute the distance first, and add caliper to it. We can specify the caliper width. The default setting is 0.2. In this example, width=0.2 is too tight to solve the matching problem.

```
smahal.dist = match_on(treat ~ age + educ + black + hisp + married + nodegree + re74 + re75,
                      data = lalonde, method="rank_mahalanobis")

### caliper width = 0.2 -> doesn't work
smahal.dist2 = smahal.dist + caliper(ps.dist, width = 0.2)
mc.out2 = matchit(treat ~ age + educ + black + hisp + married + nodegree + re74 + re75,
                 data = lalonde, method= "optimal", distance = as.matrix(smahal.dist2))

## Error: No matches were found.

### Caliper width = 1
smahal.dist3 = smahal.dist + caliper(ps.dist, width = 1)
mc.out3 = matchit(treat ~ age + educ + black + hisp + married + nodegree + re74 + re75,
                 data = lalonde, method= "optimal", distance = as.matrix(smahal.dist3))
summary(mc.out3, un = F)
```

```
##
## Call:
## matchit(formula = treat ~ age + educ + black + hisp + married +
##         nodegree + re74 + re75, data = lalonde, method = "optimal",
##         distance = as.matrix(smahal.dist3))
##
## Summary of Balance for Matched Data:
##              Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## age           25.8162      26.9622         -0.1602    0.4640    0.0768
## educ          10.3459      10.3351         0.0054    0.5622    0.0239
## blackTRUE     0.8432       0.4595         1.0556     .    0.3838
## hispTRUE      0.0595       0.0649        -0.0229     .    0.0054
## married       0.1892       0.3189        -0.3312     .    0.1297
## nodegree      0.7081       0.6649         0.0951     .    0.0432
## re74          2095.5737    2731.7476        -0.1302    1.1238    0.0734
## re75          1532.0553    1713.2806        -0.0563    1.2924    0.0478
##              eCDF Max Std. Pair Dist.
## age           0.1514         0.6029
```

```
## educ      0.0703      0.3280
## blackTRUE 0.3838      1.0556
## hispTRUE  0.0054      0.0686
## married   0.1297      0.3312
## nodegree  0.0432      0.0951
## re74      0.3351      0.3647
## re75      0.1838      0.3801
##
## Sample Sizes:
##           Control Treated
## All           429      185
## Matched       185      185
## Unmatched     244       0
## Discarded      0       0
```

Still no significant improvement in covariate balance.

### 3.3 Using the Mahalanobis distance within ps calipers with near-exact matching for black

We can directly add a penalty for black. If two subjects have different values of black, then their distance gets larger. The penalty values is  $10 \times$  maximum distance.

```
smahal.dist.black = addalmostexact(as.matrix(smahal.dist), lalonde$treat, lalonde$black, mult = 10)
mc.black.out = matchit(treat ~ age + educ + black + hisp + married + nodegree + re74 + re75,
                       data = lalonde, method= "optimal",
                       distance = as.matrix(smahal.dist.black))
summary(mc.black.out, un = F)
```

```
##
## Call:
## matchit(formula = treat ~ age + educ + black + hisp + married +
##         nodegree + re74 + re75, data = lalonde, method = "optimal",
##         distance = as.matrix(smahal.dist.black))
##
## Summary of Balance for Matched Data:
##           Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## age           25.8162      26.9297      -0.1556      0.4656      0.0759
## educ           10.3459      10.3730      -0.0134      0.5770      0.0225
## blackTRUE       0.8432       0.4703       1.0259         .      0.3730
## hispTRUE        0.0595       0.0649      -0.0229         .      0.0054
## married         0.1892       0.3243      -0.3450         .      0.1351
## nodegree        0.7081       0.6595       0.1070         .      0.0486
## re74           2095.5737     2875.0812     -0.1595      1.0613      0.0819
## re75           1532.0553     1743.9677     -0.0658      1.2681      0.0498
##           eCDF Max Std. Pair Dist.
## age           0.1514      0.5953
## educ           0.0649      0.3253
## blackTRUE      0.3730      1.0259
## hispTRUE       0.0054      0.0686
## married        0.1351      0.3450
## nodegree       0.0486      0.1070
## re74           0.3459      0.3833
## re75           0.1838      0.3867
```



```
##
## Sample Sizes:
##           Control Treated
## All           429     185
## Matched       185     185
## Unmatched     244       0
## Discarded      0       0
```

No significant change in matching quality.

## 4 When there are covariates that are hard to match on?

In such cases, we need to look at the variables that are difficult to match on. The most difficult variable is black.

```
table(lalonde$treat, lalonde$black)
```

```
##
##   FALSE TRUE
##  0   342   87
##  1    29  156
```

There is no problem in matching for non-black people. There are 29 treated and 342 control subjects. The control reservoir is much larger than the treated one. However, among black people, there are 156 treated subjects (> 87 control subjects). In this case, there are two options we can take: (1) optimal full matching, and (2) two smaller matching problems for black and non-black separately.

### 4.1 Optimal full matching

The optimal full matching method consists of 1:k matched sets & k:1 matched sets (various  $k$ ). The value of  $k$  can be very large. To avoid this situation, we usually set an option, “max.controls.” As shown in the matching output, the balance is now acceptable.

```
m.full.out = matchit(treat ~ age + educ + married + black + hisp + nodegree + re74 + re75,
                    data= lalonde, method = "full",
                    distance = "glm", max.controls = 30)
summary(m.full.out, un = F)
```

```
##
## Call:
## matchit(formula = treat ~ age + educ + married + black + hisp +
##         nodegree + re74 + re75, data = lalonde, method = "full",
##         distance = "glm", max.controls = 30)
##
## Summary of Balance for Matched Data:
##           Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance           0.5774           0.5756           0.0082           0.9784           0.0075
## age                25.8162           24.7162           0.1537           0.4989           0.0805
## educ               10.3459           10.3732           -0.0135           0.5601           0.0218
## married            0.1892            0.1387            0.1289              .           0.0505
## blackTRUE          0.8432            0.8347            0.0236              .           0.0086
## hispTRUE           0.0595            0.0575            0.0081              .           0.0019
## nodegree           0.7081            0.7032            0.0108              .           0.0049
## re74               2095.5737          2444.7294          -0.0715           1.0359           0.0441
## re75               1532.0553          1589.2944          -0.0178           1.8906           0.0672
##
##           eCDF Max Std. Pair Dist.
```

```
## distance    0.0486      0.0469
## age         0.3203      1.2982
## educ        0.0581      1.1970
## married     0.0505      0.7976
## blackTRUE   0.0086      0.0378
## hispTRUE    0.0019      0.5649
## nodegree    0.0049      1.0198
## re74        0.2469      1.0090
## re75        0.2364      0.8407
```

```
##
```

```
## Sample Sizes:
```

```
##           Control Treated
## All           429.      185
## Matched (ESS)  53.05    185
## Matched       429.      185
## Unmatched     0.        0
## Discarded     0.        0
```

We can try a different link (probit) for the propensity score model. Balance is better, as determined by the lower SMD.

```
m.full.probit.out = matchit(treat ~ age + educ + married + black + hisp + nodegree + re74 + re75,
                           data= lalonde, method = "full",
                           distance = "glm", link = "probit")
summary(m.full.probit.out, un = F)
```

```
##
```

```
## Call:
```

```
## matchit(formula = treat ~ age + educ + married + black + hisp +
##         nodegree + re74 + re75, data = lalonde, method = "full",
##         distance = "glm", link = "probit")
##
```

```
## Summary of Balance for Matched Data:
```

	Means Treated	Means Control	Std. Mean Diff.	Var. Ratio	eCDF Mean
## distance	0.5773	0.5766	0.0034	0.9957	0.0042
## age	25.8162	25.3355	0.0672	0.4776	0.0810
## educ	10.3459	10.4480	-0.0508	0.6013	0.0232
## married	0.1892	0.1623	0.0687	.	0.0269
## blackTRUE	0.8432	0.8389	0.0119	.	0.0043
## hispTRUE	0.0595	0.0484	0.0469	.	0.0111
## nodegree	0.7081	0.6767	0.0692	.	0.0315
## re74	2095.5737	2062.0642	0.0069	1.3524	0.0292
## re75	1532.0553	1619.7359	-0.0272	1.4936	0.0569

```
##           eCDF Max Std. Pair Dist.
```

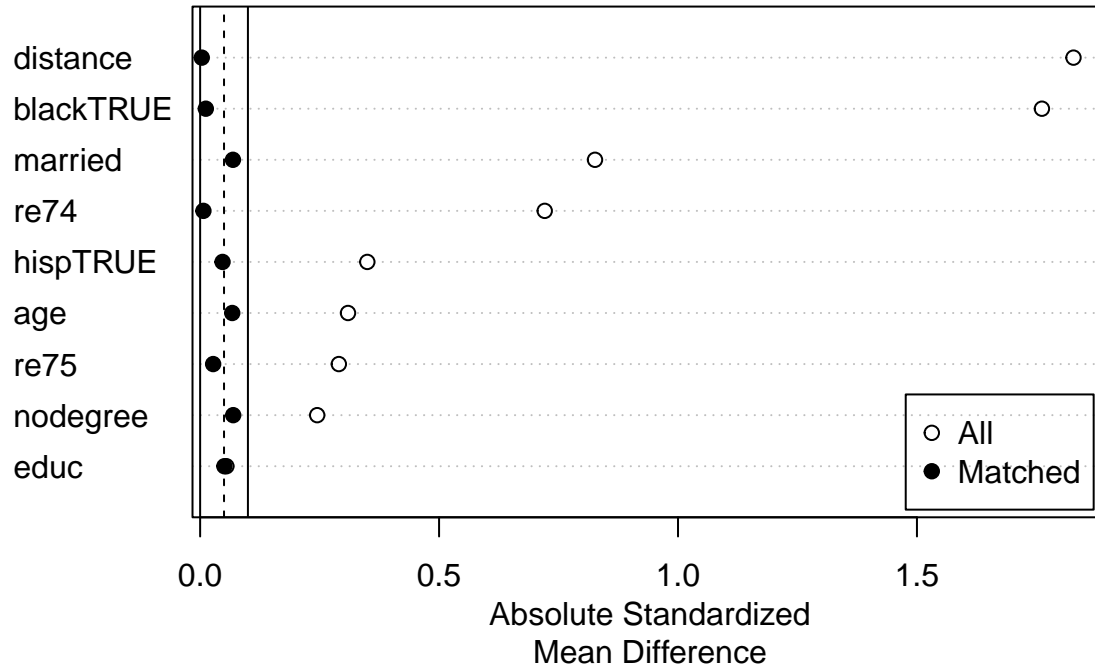
	eCDF Max	Std. Pair Dist.
## distance	0.0486	0.0198
## age	0.2859	1.2683
## educ	0.0597	1.2042
## married	0.0269	0.4725
## blackTRUE	0.0043	0.0162
## hispTRUE	0.0111	0.4911
## nodegree	0.0315	0.9439
## re74	0.1983	0.8411
## re75	0.2213	0.8228

```
##
```

```
## Sample Sizes:
```

```
##           Control Treated
## All           429.      185
## Matched (ESS)  51.37   185
## Matched       429.      185
## Unmatched      0.       0
## Discarded     0.       0
```

```
plot(summary(m.full.probit.out), var.order = "unmatched")
```



## 4.2 Separate optimal pair matching for black and non-black

In the full matching method, there is no thrown away subject. However, using all subjects is not always the best way for matching. There could be a loss in precision due to the weights. Subjects may be weighted in such a way that they contribute less to the sample than would unweighted subjects. **Effective sample size (ESS)** of the full matching weighted sample may be lower than even that of 1:1 pair matching.

An alternative method is to remove subjects that are not comparable. We can match exactly on black.

```
## matching separate
m.exact.out = matchit(treat ~ age + educ + black + hisp + married + nodegree + re74 + re75,
                      data = lalonde, method = "optimal",
                      distance = "robust_mahalanobis", exact = ~black)

summary(m.exact.out, un = F)
```

```
##
## Call:
## matchit(formula = treat ~ age + educ + black + hisp + married +
##         nodegree + re74 + re75, data = lalonde, method = "optimal",
##         distance = "robust_mahalanobis", exact = ~black)
##
## Summary of Balance for Matched Data:
##           Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## age           24.7414         26.1466         -0.1964         0.4534         0.0666
```

```

## educ          10.2672      10.2069      0.0300      0.6052      0.0231
## blackTRUE     0.7500      0.7500      0.0000      .           0.0000
## hispTRUE      0.0948      0.0862      0.0365      .           0.0086
## married       0.2328      0.2672     -0.0880      .           0.0345
## nodegree      0.6724      0.6293      0.0948      .           0.0431
## re74          2803.2334     2735.4405     0.0139      1.3688     0.0262
## re75          1914.6811     1722.1051     0.0598      1.5669     0.0180
##              eCDF Max Std. Pair Dist.
## age           0.1207      0.5313
## educ          0.0603      0.3044
## blackTRUE     0.0000      0.0000
## hispTRUE      0.0086      0.0365
## married       0.0345      0.0880
## nodegree      0.0431      0.0948
## re74          0.1724      0.3061
## re75          0.0517      0.4172
##
## Sample Sizes:
##              Control Treated
## All           429      185
## Matched       116      116
## Unmatched     313      69
## Discarded      0       0

```

## 5 Analysis Stage

### 5.1 Using randomization inference

Optimal full matching produces matched sets with variable set sizes. Wilcoxon's signed rank test cannot be applicable. We instead use Huber's M-statistic. Randomization inference using this statistic is easily done with the `senfm` function in the `sensitivityfull` package.

We need to make a table for outcome first. Each row represents a matched set. The first column is always the treated outcomes.

```

## analysis

m.full = match.data(m.full.probit.out)
subclass = as.numeric(m.full$subclass)
## outcome matrix
y.fm = matrix(NA, nrow = length(unique(subclass)), ncol = max(table(subclass)))
treat1.vec = rep(NA, length(unique(subclass)))
for(i in 1:length(unique(subclass))){
  temp.indicator = (subclass == i)
  if(sum(m.full$treat[temp.indicator]) == 1){
    y.fm[i,1] = m.full$re78[temp.indicator==1 & m.full$treat==1]
    y.fm[i,(2:sum(temp.indicator))] = m.full$re78[temp.indicator==1 & m.full$treat ==0]
    treat1.vec[i]= TRUE
  }else{
    y.fm[i,1] = m.full$re78[temp.indicator==1 & m.full$treat==0]
    y.fm[i,(2:sum(temp.indicator))] = m.full$re78[temp.indicator==1 & m.full$treat==1]
    treat1.vec[i]= FALSE
  }
}
}

```

Computing the paired outcome table for matched pairs is easier. Randomization inference can be implemented by using the `senmw` function in the `sensitivitymw` package.

```
m.exact = match.data(m.exact.out)
y.pair = matrix(NA, nrow = length(unique(m.exact$subclass)), ncol = 2)
for(i in 1:length(unique(m.exact$subclass))){
  y.pair[i,1] = m.exact$re78[m.exact$subclass == i & m.exact$treat == 1]
  y.pair[i,2] = m.exact$re78[m.exact$subclass == i & m.exact$treat == 0]
}
```

```
library(sensitivityfull)
senfm(y.fm, treated1 = treat1.vec)
```

```
## $pval
## [1] 0.04161583
##
## $deviate
## [1] 1.732235
##
## $statistic
## [1] 4.226839
##
## $expectation
## [1] 3.365737e-16
##
## $variance
## [1] 5.954119
```

```
senfmCI(y.fm, treated1 = treat1.vec)
```

```
## $PointEstimates
## [1] 1384.781 1384.781
##
## $ConfidenceInterval
## [1] -183.1355 2968.8862
##
## $description
##           Coverage           Gamma Confidence Interval
##           "0.95"           "1"           "Two-sided"
```

```
library(sensitivitymw)
senmw(y.pair)
```

```
## $pval
## [1] 0.07078286
##
## $deviate
## [1] 1.469985
##
## $statistic
## [1] 3.845654
##
## $expectation
## [1] 0
##
## $variance
```

```
## [1] 6.844069
senmwCI(y.pair, one.sided = F)
```

```
## $PointEstimate
## minimum maximum
## 1250 1250
##
## $Confidence.Interval
## minimum maximum
## -450 3000
```

## 5.2 Using an outcome model

We can run a simple regression of the outcome on the treatment with matching weights. Using cluster-robust standard errors is recommended. Also, using pair membership as the clustering variable is recommended.

```
library("lmtest") # coeftest
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
## as.Date, as.Date.numeric
```

```
library("sandwich") # vcovCL
```

```
fit.full = lm(re78 ~ treat, data = m.full, weights = weights)
coeftest(fit.full, vcov. = vcovCL, cluster = ~subclass)
```

```
##
## t test of coefficients:
##
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4384.97    597.26  7.3418 6.744e-13 ***
## treat       1964.17    701.34  2.8006 0.005262 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

A natural question to ask is whether we can include covariates in the outcome model. One may ask, why use matching at all if you are going to model the outcome with covariates anyway? Matching reduces the dependence of the effect estimate on correct specification of the outcome model, and this is the central idea of Ho et al. (2007).

However, searching for the best outcome model by trying many models is discouraged. Doing so can invalidate results and yield a conclusion that fails to replicate. We recommend only including the same terms included in the propensity score model unless there is a strong *a priori* and justifiable reason to model the outcome differently.

```
fit.full.with.cov = lm(re78 ~ treat + age + educ + black + hisp +
                      married + nodegree + re74 + re75,
                      data = m.full, weights = weights)
coeftest(fit.full.with.cov, vcov. = vcovCL, cluster = ~subclass)
```

```
##
## t test of coefficients:
```

```
##
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.2479e+03  3.2382e+03  1.3118  0.19009
## treat        2.0915e+03  6.7833e+02  3.0834  0.00214 **
## age         -3.5164e+01  3.6614e+01 -0.9604  0.33724
## educ        2.1371e+02  2.0154e+02  1.0604  0.28939
## blackTRUE   -1.5550e+03  8.2045e+02 -1.8953  0.05853 .
## hispTRUE    -5.2880e+02  1.5323e+03 -0.3451  0.73014
## married     7.4987e+02  1.0759e+03  0.6970  0.48609
## nodegree    -1.3684e+03  1.2127e+03 -1.1284  0.25960
## re74        -1.9924e-02  1.4178e-01 -0.1405  0.88830
## re75         5.9919e-01  1.3929e-01  4.3017  1.977e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

For matched pairs, we can use

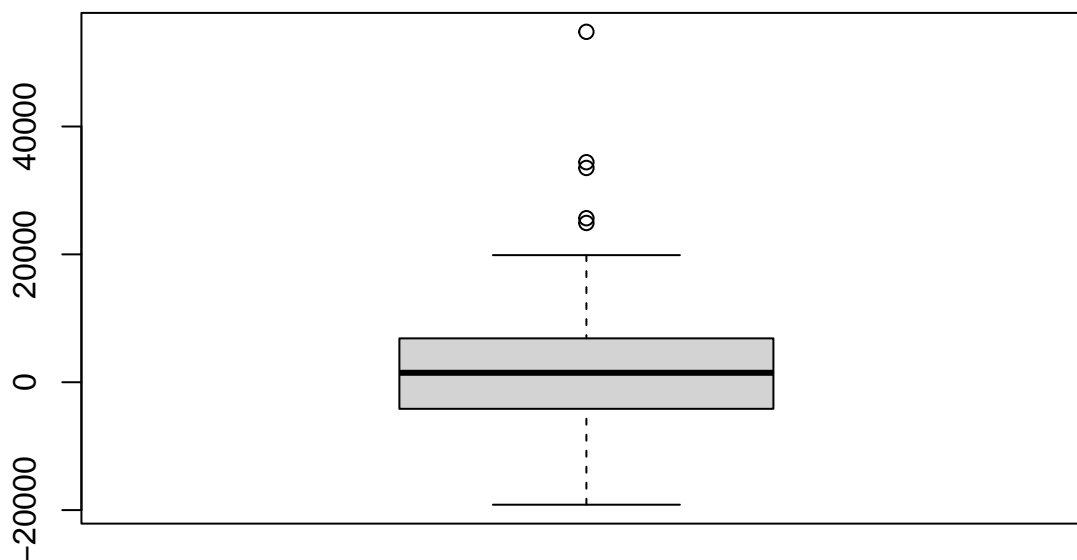
```
fit.pair = lm(re78 ~ treat, data = m.exact, weights = weights)
coefstest(fit.pair, vcov. = vcovCL, cluster = ~subclass)
```

```
##
## t test of coefficients:
##
##           Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5310.0     605.1  8.7754 3.909e-16 ***
## treat         1905.7     1011.1  1.8847  0.06073 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Oh, wait! Why are these regression estimates so different from the estimates based on randomization inference?

```
boxplot(y.pair[,1] - y.pair[,2], main = "Treated-Minus-Control Differences")
```

### Treated-Minus-Control Differences



## 6 Cardinality matching

```
solver = "glpk"
m.card.out = matchit(treat ~ age + educ + black + hisp + married + nodegree + re74 + re75,
                    data = lalonde, method = "cardinality", tols = 0.01,
                    solver = "glpk", exact = ~black)
# m.card.out = matchit(treat ~ age + educ + black + hisp + married + nodegree + re74 + re75,
#                      data = lalonde, method = "cardinality", tols = 0.1,
#                      solver = "glpk")
summary(m.card.out, un = F)

##
## Call:
## matchit(formula = treat ~ age + educ + black + hisp + married +
##         nodegree + re74 + re75, data = lalonde, method = "cardinality",
##         exact = ~black, tols = 0.01, solver = "glpk")
##
## Summary of Balance for Matched Data:
##           Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## age           25.8017      25.7931          0.0012      0.4481  0.0851
## educ           10.2069      10.1983          0.0043      0.5636  0.0231
## blackTRUE       0.7500       0.7500          0.0000          .  0.0000
## hispTRUE        0.0948       0.0948          0.0000          .  0.0000
## married         0.2672       0.2672          0.0000          .  0.0000
## nodegree        0.6379       0.6379          0.0000          .  0.0000
## re74            2747.3458     2784.6924         -0.0076      1.1033  0.0328
## re75            1834.4821     1817.5940          0.0052      1.3637  0.0234
##           eCDF Max
## age           0.2414
## educ           0.0517
## blackTRUE     0.0000
## hispTRUE      0.0000
## married       0.0000
## nodegree      0.0000
## re74          0.1638
## re75          0.0690
##
## Sample Sizes:
##           Control Treated
## All           429      185
## Matched       116      116
## Unmatched     313       69
## Discarded      0       0

## re-match to improve balance within pairs
m.card.re = matchit(treat ~ age + educ + black + hisp + married + nodegree + re74 + re75,
                   data = lalonde, method = "optimal", distance = "mahalanobis",
                   discard = m.card.out$weights == 0)

m.card = match.data(m.card.re)

## analysis
y.pair.card = matrix(NA, nrow = length(unique(m.card$subclass)), ncol = 2)
for(i in 1:length(unique(m.card$subclass))){
```



```

y.pair.card[i,1] = m.card$re78[m.card$subclass == i & m.card$treat == 1]
y.pair.card[i,2] = m.card$re78[m.card$subclass == i & m.card$treat == 0]
}
senmwCI(y.pair.card, one.sided = F)

```

```

## $PointEstimate
## minimum maximum
##      660      660
##
## $Confidence.Interval
## minimum maximum
##     -890     2220

```

```

coefstest(lm(re78 ~ treat, data = m.card, weights = weights),
          vcov. = vcovCL, cluster = ~subclass) # Similar?

```

```

##
## t test of coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  5583.36    608.85   9.1703  <2e-16 ***
## treat         938.54    837.82   1.1202   0.2638
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

## 7 A Bit More: separate matching + cardinality

```

## Black
m.black.out = matchit(treat ~ age + educ + married + nodegree + re74 + re75,
                     data = subset(lalonde, race == "black"),
                     method = "cardinality", tols = 0.01, solver = "glpk")
summary(m.black.out, un = F)

```

```

##
## Call:
## matchit(formula = treat ~ age + educ + married + nodegree + re74 +
##         re75, data = subset(lalonde, race == "black"), method = "cardinality",
##         tols = 0.01, solver = "glpk")
##
## Summary of Balance for Matched Data:
##           Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## age           26.1379      26.0690      0.0094      0.4436  0.0907
## educ           10.1034      10.0920      0.0056      0.6288  0.0194
## married         0.2874         0.2874      0.0000          .  0.0000
## nodegree        0.6437         0.6437      0.0000          .  0.0000
## re74           3068.8587     3117.0584     -0.0094      1.3726  0.0402
## re75           1860.2966     1834.4220      0.0078      1.6167  0.0322
##           eCDF Max
## age           0.2644
## educ           0.0460
## married        0.0000
## nodegree       0.0000
## re74           0.2299
## re75           0.1034

```

```

##
## Sample Sizes:
##           Control Treated
## All           87    156
## Matched       87     87
## Unmatched     0     69
## Discarded     0     0

m.black.re = matchit(treat ~ age + educ + black + hisp + married + nodegree + re74 + re75,
                    data = subset(lalonde, race == "black"),
                    method = "optimal", distance = "mahalanobis",
                    discard = m.black.out$weights == 0)
m.b = match.data(m.black.re)

y.black = matrix(NA, nrow = length(unique(m.b$subclass)), ncol = 2)
for(i in 1:length(unique(m.b$subclass))){
  y.black[i,1] = m.b$re78[m.b$subclass == i & m.b$treat == 1]
  y.black[i,2] = m.b$re78[m.b$subclass == i & m.b$treat == 0]
}
senmwCI(y.black, one.sided = F)

## $PointEstimate
## minimum maximum
##      890      890
##
## $Confidence.Interval
## minimum maximum
##     -950     2760

## Non-black
m.nb.out1 = matchit(treat ~ age + educ + hisp + married + nodegree + re74 + re75,
                   data = subset(lalonde, race != "black"),
                   method = "cardinality", tols = 0.01, solver = "glpk", ratio = 3)
m.nb.out2 = matchit(treat ~ age + educ + hisp + married + nodegree + re74 + re75,
                   data = subset(lalonde, race != "black"),
                   method = "cardinality", tols = 0.1, solver = "glpk", ratio = 5)
summary(m.nb.out1, un = F)

##
## Call:
## matchit(formula = treat ~ age + educ + hisp + married + nodegree +
##         re74 + re75, data = subset(lalonde, race != "black"), method = "cardinality",
##         ratio = 3, tols = 0.01, solver = "glpk")
##
## Summary of Balance for Matched Data:
##           Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## age           24.9310      24.9195      0.0018      0.3852      0.0802
## educ           10.5172      10.5057      0.0066      0.4118      0.0333
## hispTRUE        0.3793        0.3793      -0.0000      .           0.0000
## married         0.2069        0.2069      -0.0000      .           0.0000
## nodegree        0.6207        0.6207      -0.0000      .           0.0000
## re74            1775.8300     1785.8829     -0.0029      0.9649      0.0559
## re75            1754.3996     1738.9384      0.0055      0.8699      0.0654
##           eCDF Max
## age           0.2759

```

```

## educ      0.0920
## hispTRUE  0.0000
## married   0.0000
## nodegree  0.0000
## re74      0.2184
## re75      0.2069
##
## Sample Sizes:
##           Control Treated
## All           342      29
## Matched       87      29
## Unmatched     255      0
## Discarded      0      0
summary(m.nb.out2, un = F)

##
## Call:
## matchit(formula = treat ~ age + educ + hisp + married + nodegree +
##          re74 + re75, data = subset(lalonde, race != "black"), method = "cardinality",
##          ratio = 5, tols = 0.1, solver = "glpk")
##
## Summary of Balance for Matched Data:
##           Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## age           24.9310      24.5034          0.0672    0.4062    0.0755
## educ           10.5172      10.3448          0.0988    0.3241    0.0446
## hispTRUE        0.3793        0.3310          0.0995      .    0.0483
## married         0.2069        0.2414         -0.0851      .    0.0345
## nodegree        0.6207        0.5862          0.0711      .    0.0345
## re74           1775.8300     2083.7951         -0.0887    0.9999    0.0647
## re75           1754.3996     1700.4114          0.0192    1.7056    0.0863
##           eCDF Max
## age           0.2759
## educ           0.1103
## hispTRUE      0.0483
## married       0.0345
## nodegree      0.0345
## re74          0.3379
## re75          0.2483
##
## Sample Sizes:
##           Control Treated
## All           342      29
## Matched       145      29
## Unmatched     197      0
## Discarded      0      0

m.nb.re1 = matchit(treat ~ age + educ + black + hisp + married + nodegree + re74 + re75,
                  data = subset(lalonde, race != "black"),
                  method = "optimal", distance = "mahalanobis",
                  discard = m.nb.out1$weights == 0, ratio = 3)
m.nb.re2 = matchit(treat ~ age + educ + black + hisp + married + nodegree + re74 + re75,
                  data = subset(lalonde, race != "black"),
                  method = "optimal", distance = "mahalanobis",
                  discard = m.nb.out2$weights == 0, ratio = 5)

```

```

m.nb1 = match.data(m.nb.re1)
m.nb2 = match.data(m.nb.re2)

y.nb1 = matrix(NA, nrow = length(unique(m.nb1$subclass)), ncol = 4)
for(i in 1:length(unique(m.nb1$subclass))){
  y.nb1[i,1] = m.nb1$re78[m.nb1$subclass == i & m.nb1$treat == 1]
  y.nb1[i,2:4] = m.nb1$re78[m.nb1$subclass == i & m.nb1$treat == 0]
}
senmwCI(y.nb1, one.sided = F)

## $PointEstimate
## minimum maximum
## 1190 1190
##
## $Confidence.Interval
## minimum maximum
## -1610 3980

y.nb2 = matrix(NA, nrow = length(unique(m.nb2$subclass)), ncol = 6)
for(i in 1:length(unique(m.nb2$subclass))){
  y.nb2[i,1] = m.nb2$re78[m.nb2$subclass == i & m.nb2$treat == 1]
  y.nb2[i,2:6] = m.nb2$re78[m.nb2$subclass == i & m.nb2$treat == 0]
}
senmwCI(y.nb2, one.sided = F)

## $PointEstimate
## minimum maximum
## 1620 1620
##
## $Confidence.Interval
## minimum maximum
## -770 4030

y.card.tot1 = rbind(cbind(y.black, rep(NA,87),rep(NA,87)), y.nb1)
y.card.tot2 = rbind(cbind(y.black, rep(NA,87),rep(NA,87),rep(NA,87),rep(NA,87)), y.nb2)
senfmCI(y.card.tot1, treated1 = rep(TRUE, 116))

## $PointEstimates
## [1] 982.1014 982.1014
##
## $ConfidenceInterval
## [1] -582.8645 2534.8365
##
## $description
## Coverage Gamma Confidence Interval
## "0.95" "1" "Two-sided"

senfmCI(y.card.tot2, treated1 = rep(TRUE, 116))

## $PointEstimates
## [1] 1148.722 1148.722
##
## $ConfidenceInterval
## [1] -301.7941 2622.4135
##

```

```
## $description
##           Coverage           Gamma Confidence Interval
##           "0.95"           "1"           "Two-sided"
```

## 8 Reference

- Dehejia, R.H., and Wahba,S. (1999). Causal Effects in Nonexperimental Studies: Reevaluating the Evaluation of Training Programs, *Journal of the American Statistical Association*, 94(448), 1053–1062.
- Ho, D.E., Imai, K., King, G., and Stuart, E.A. (2007). Matching as Nonparametric Preprocessing for Reducing Model Dependence in Parametric Causal Inference, *Political Analysis*, 15(3), 199-236.
- Ho, D.E., Imai, K., King, G., and Stuart, E.A. (2011). MatchIt: Nonparametric Preprocessing for Parametric Causal Inference. *Journal of Statistical Software*, 42(8), 1–28.
- Stuart, E.A. (2010). Matching Methods for Causal Inference: A Review and a Look Forward, *Statistical Science*, 25(1), 1–21.
- VanderWeele T.J. (2019). Principles of Confounder Selection, *European Journal of Epidemiology*, 34(3), 211–219.