Lab 5: Matching and Weighting

2024-03-01

Matching: MatchIt

We will use Lalonde's data on the evaluation of the National Supported Work program to demonstrate MatchIt's capabilities.

```
pacman::p_load(tidyverse, MatchIt, broom, fixest, knitr, Matching, rgenoud)
data(lalonde)
lalonde <- lalonde %>% mutate(
     race = case_when(
          black == 1 ~ 'black',
          hisp == 1 ~ 'hispanic',
          TRUE ~ 'white'
     )
)
head(lalonde) %>% kable
```

age	educ	black	hisp	married	nodegr	re74	re75	re78	u74	u75	treat	race
37	11	1	0	1	1	0	0	9930.05	1	1	1	black
22	9	0	1	0	1	0	0	3595.89	1	1	1	hispanic
30	12	1	0	0	0	0	0	24909.50	1	1	1	black
27	11	1	0	0	1	0	0	7506.15	1	1	1	black
33	8	1	0	0	1	0	0	289.79	1	1	1	black
22	9	1	0	0	1	0	0	4056.49	1	1	1	black

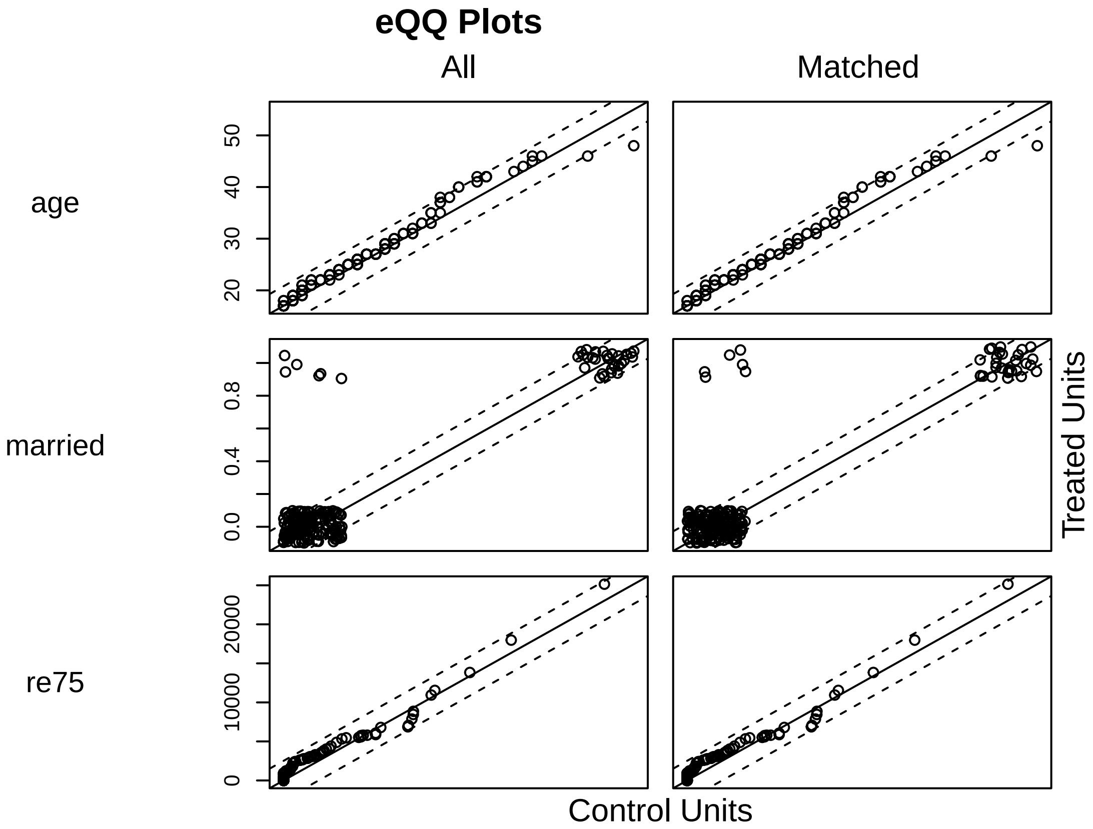
The statistical quantity of interest is the causal effect of the treatment (`treat`) on 1978 earnings (`re78`). The other variables are pre-treatment covariates. See `?lalonde` for more information on this dataset.

Before matching, it can be a good idea to view the initial imbalance in one's data that matching is attempting to eliminate. We can do this using the code below:

```
m.out0 <- matchit(
     treat ~ age + educ + race + married +
          nodegr + re74 + re75,
     data = lalonde,
     method = NULL, distance = "glm"
)
summary(m.out0)
```

```
##
## Call:
## matchit(formula = treat ~ age + educ + race + married + nodegr +
##     re74 + re75, data = lalonde, method = NULL, distance = "glm")
##
## Summary of Balance for All Data:
##              Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance            0.4377        0.4001          0.3935     1.0471    0.1117
## age                25.8162       25.0538          0.1066     1.0278    0.0254
## educ               10.3459       10.0885          0.1281     1.5513    0.0287
## raceblack           0.8432        0.8269          0.0449          .    0.0163
## racehispanic        0.0595        0.1077         -0.2040          .    0.0482
## racewhite           0.0973        0.0654          0.1077          .    0.0319
## married             0.1892        0.1538          0.0902          .    0.0353
## nodegr              0.7081        0.8346         -0.2783          .    0.1265
## re74             2095.5740     2107.0268         -0.0023     0.7381    0.0192
## re75             1532.0556     1266.9092          0.0824     1.0763    0.0508
##              eCDF Max
## distance       0.2140
## age            0.0652
## educ           0.1265
## raceblack      0.0163
## racehispanic   0.0482
## racewhite      0.0319
## married        0.0353
## nodegr         0.1265
## re74           0.0471
## re75           0.1075
##
## Sample Sizes:
##           Control Treated
## All           260     185
## Matched       260     185
## Unmatched       0       0
## Discarded       0       0
```

```
plot(m.out0,
     type = "qq", interactive = FALSE,
     which.xs = c("age", "married", "re75")
)
```



We can see severe imbalances as measured by the standardized mean differences (`Std. Mean Diff.`), variance ratios (`Var. Ratio`), and empirical cumulative density function (eCDF) statistics. Values of standardized mean differences and eCDF statistics close to zero and values of variance ratios close to one indicate good balance, and here many of them are far from their ideal values.

Now, matching can be performed. There are several different classes and methods of matching. You can use vignette("matching-methods") to know more.

Exact Matching

With exact matching, a complete cross of the covariates is used to form subclasses defined by each combination of the covariate levels. Any subclass that doesn't contain both treated and control units is discarded, leaving only subclasses containing treatment and control units that are exactly equal on the included covariates. The benefits of exact matching are that confounding due to the covariates included is completely eliminated, regardless of the functional form of the treatment or outcome models. The problem is that typically many units will be discarded, sometimes dramatically reducing precision and changing the target population of inference.

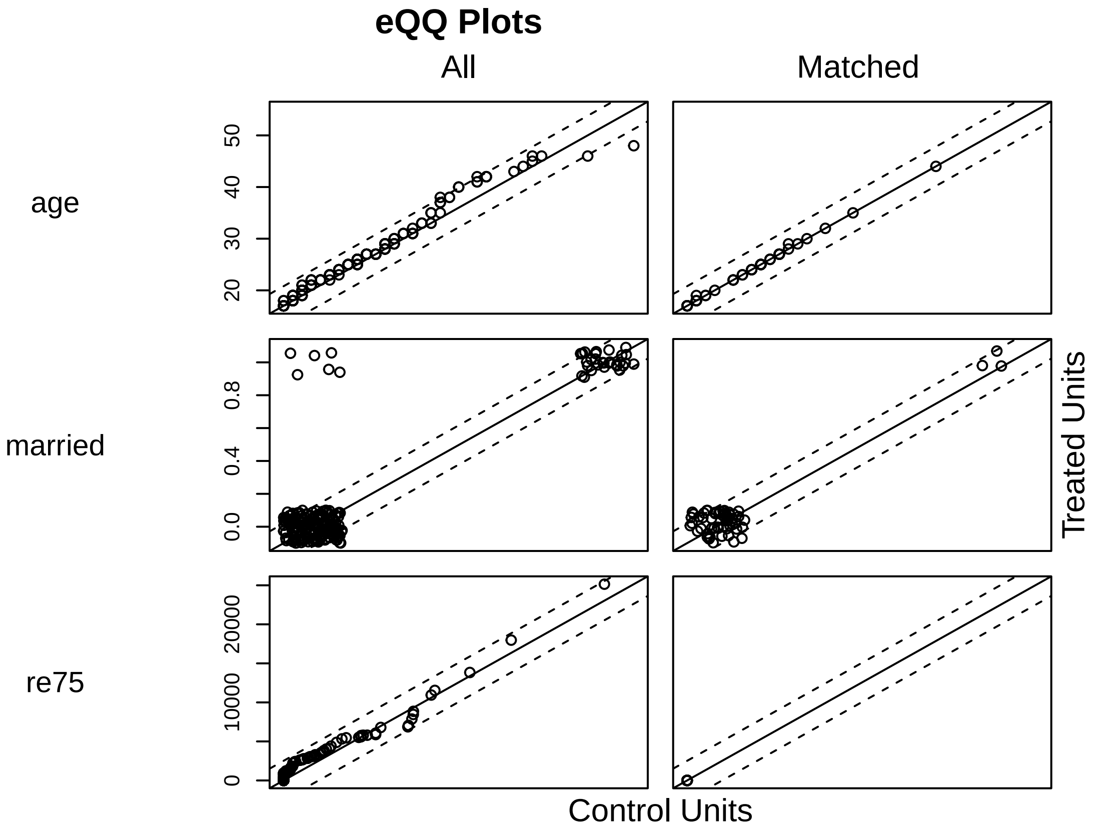
```r
# Exact Matching
m.exact <- matchit(
     treat ~ age + educ + race + married +
          nodegr + re74 + re75,
     data = lalonde,
     method = "exact", distance = "glm"
)
m.exact
```

```
## A ‘matchit‘ object
##  - method: Exact matching
##  - number of obs.: 445 (original), 129 (matched)
##  - target estimand: ATT
##  - covariates: age, educ, race, married, nodegr, re74, re75
```

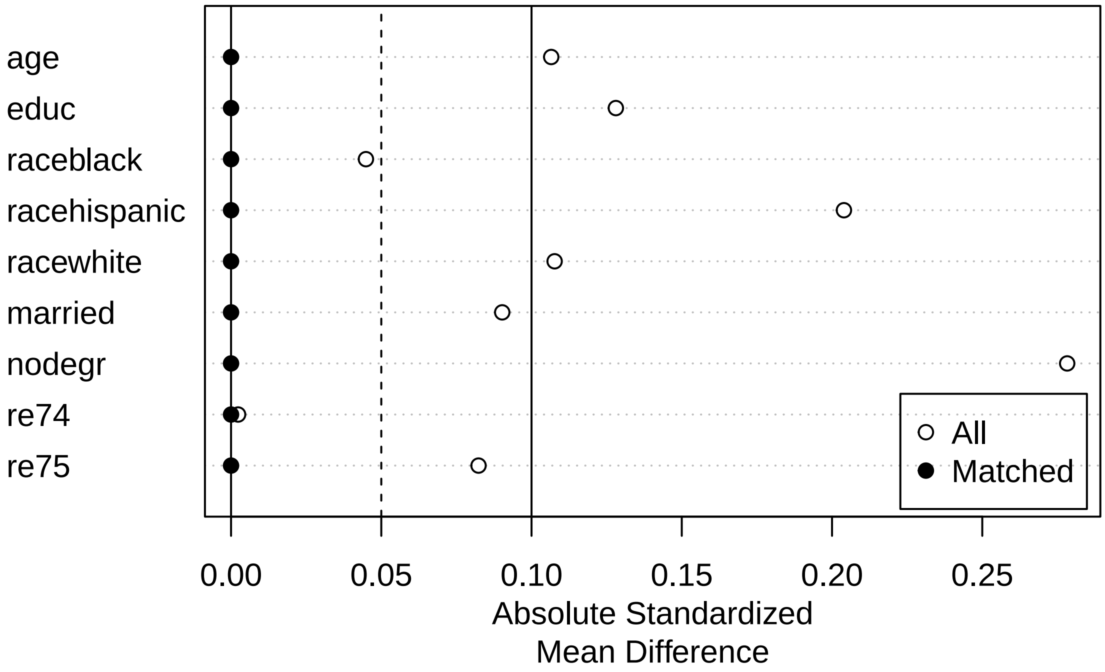
```
# un=F flag to exclude the pre-matched balance checks
summary(m.exact, un=F)
```

```
##
## Call:
## matchit(formula = treat ~ age + educ + race + married + nodegr +
##     re74 + re75, data = lalonde, method = "exact", distance = "glm")
##
## Summary of Balance for Matched Data:
##              Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## age                23.5818       23.5818               0     0.9963         0
## educ               10.4364       10.4364               0     0.9963         0
## raceblack           1.0000        1.0000               0          .         0
## racehispanic        0.0000        0.0000               0          .         0
## racewhite           0.0000        0.0000               0          .         0
## married             0.0545        0.0545               0          .         0
## nodegr              0.8000        0.8000               0          .         0
## re74                0.0000        0.0000               0          .         0
## re75                0.0000        0.0000               0          .         0
##              eCDF Max Std. Pair Dist.
## age                 0               0
## educ                0               0
## raceblack           0               0
## racehispanic        0               0
## racewhite           0               0
## married             0               0
## nodegr              0               0
## re74                0               0
## re75                0               0
##
## Sample Sizes:
##                Control Treated
## All               260.     185
## Matched (ESS)    45.9      55
## Matched          74.       55
## Unmatched       186.      130
## Discarded         0.        0
```

```
plot(m.exact,
     type = "qq", interactive = FALSE,
     which.xs = c("age", "married", "re75")
)
```



```r
plot(summary(m.exact))
```



CEM

Coarsened exact matching (CEM) is a form of stratum matching that involves first coarsening the covariates by creating bins and then performing exact matching on the new coarsened versions of the covariates. The degree and method of coarsening can be controlled by the user to manage the trade-off between exact and approximate balancing.

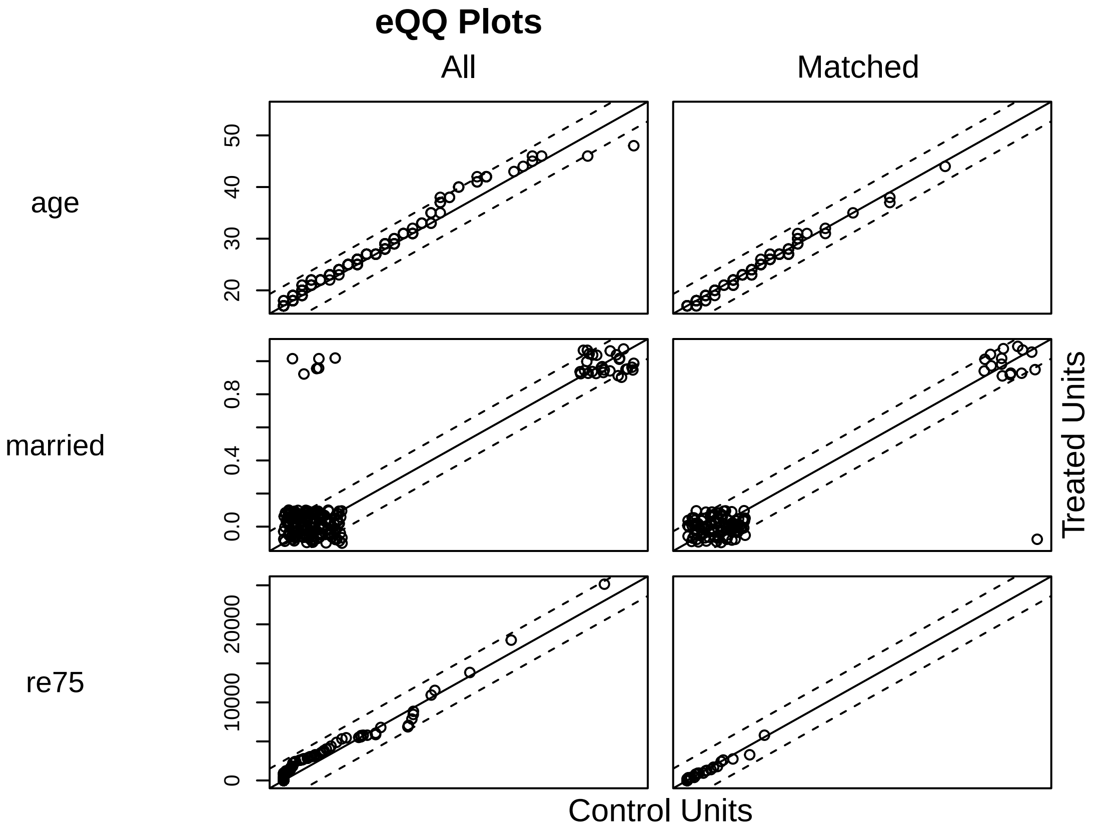
The default coarsening strategy uses the Sturges method for setting the bin size. See `?nclass.Sturges` for more information. You can set the cutpoints manually using the `cutpoints` argument. See `?method_cem` for more information.

```
m.cem <- matchit(
     treat ~ age + educ + race + married +
          nodegr + re74 + re75,
     data = lalonde,
     method = "cem"
)
summary(m.cem, un = FALSE)
```

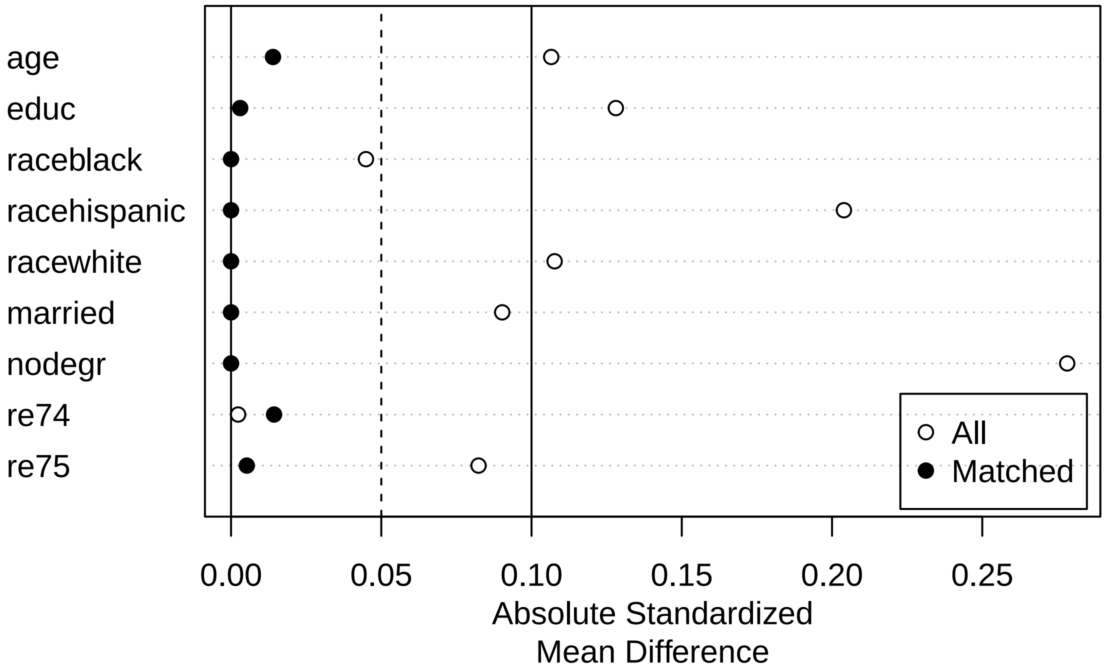
```
##
## Call:
## matchit(formula = treat ~ age + educ + race + married + nodegr +
##     re74 + re75, data = lalonde, method = "cem")
##
## Summary of Balance for Matched Data:
```

```
##                Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## age                  23.6147       23.7145         -0.0140     1.0171    0.0085
## educ                 10.3853       10.3914         -0.0030     1.0143    0.0004
## raceblack             0.9174        0.9174          0.0000          .    0.0000
## racehispanic          0.0183        0.0183         -0.0000          .    0.0000
## racewhite             0.0642        0.0642         -0.0000          .    0.0000
## married               0.1376        0.1376         -0.0000          .    0.0000
## nodegr                0.7523        0.7523          0.0000          .    0.0000
## re74                475.9838      406.0326          0.0143     0.9226    0.0114
## re75                323.3285      340.2165         -0.0052     0.8582    0.0087
##                eCDF Max Std. Pair Dist.
## age              0.0428          0.1474
## educ             0.0061          0.0088
## raceblack        0.0000          0.0000
## racehispanic     0.0000          0.0000
## racewhite        0.0000          0.0000
## married          0.0000          0.0000
## nodegr           0.0000          0.0000
## re74             0.0430          0.0816
## re75             0.0386          0.0969
##
## Sample Sizes:
##                 Control Treated
## All              260.       185
## Matched (ESS)    104.52     109
## Matched          156.       109
## Unmatched        104.        76
## Discarded          0.         0
```

```
plot(m.cem,
     type = "qq", interactive = FALSE,
     which.xs = c("age", "married", "re75")
)
```



```r
plot(summary(m.cem))
```



Propensity Score Matching

Next, we will perform 1:1 nearest neighbor (NN) matching on the propensity score. One by one, each treated unit is paired with an available control unit that has the closest propensity score to it. Any remaining control units are left unmatched and excluded from further analysis.

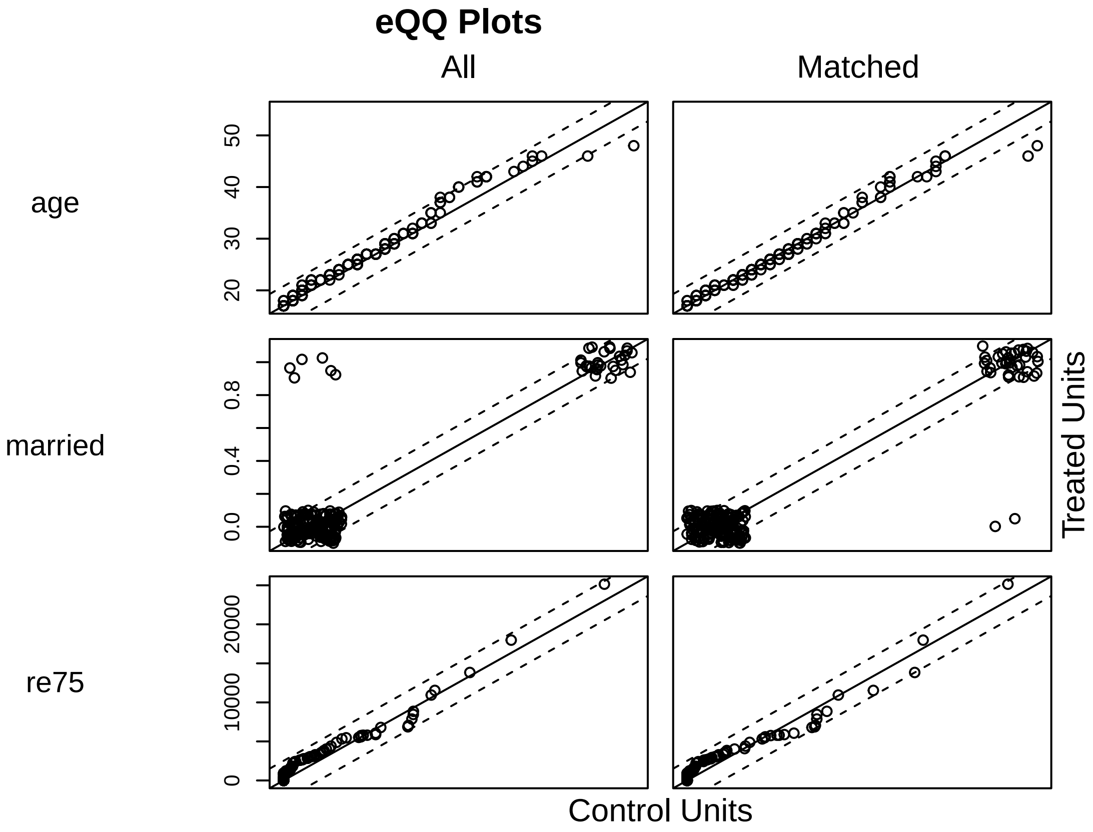
We use the same syntax as before, but this time specify `method = "nearest"` to implement nearest neighbor matching, again using a logistic regression propensity score. Many other arguments are available for tuning the matching method and method of propensity score estimation.

```
m.pscore.nn <- matchit(
     treat ~ age + educ + race + married +
          nodegr + re74 + re75,
     data = lalonde,
     method = "nearest", distance = "glm"
)
summary(m.pscore.nn, un = FALSE)
```

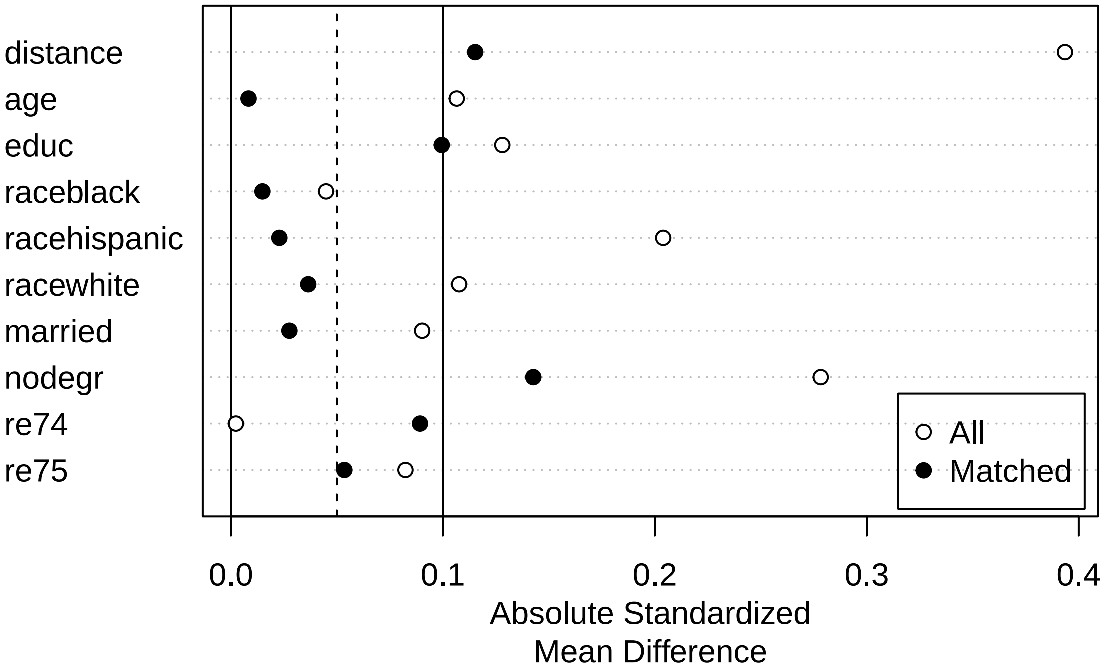
```
##
## Call:
## matchit(formula = treat ~ age + educ + race + married + nodegr +
##     re74 + re75, data = lalonde, method = "nearest", distance = "glm")
##
## Summary of Balance for Matched Data:
##              Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance            0.4377        0.4267          0.1152     1.0987    0.0298
## age                25.8162       25.8757         -0.0083     0.9551    0.0121
## educ               10.3459       10.1459          0.0995     1.3748    0.0220
## raceblack           0.8432        0.8486         -0.0149          .    0.0054
## racehispanic        0.0595        0.0649         -0.0229          .    0.0054
## racewhite           0.0973        0.0865          0.0365          .    0.0108
## married             0.1892        0.2000         -0.0276          .    0.0108
## nodegr              0.7081        0.7730         -0.1427          .    0.0649
## re74             2095.5740     1659.5326          0.0892     1.1752    0.0351
## re75             1532.0556     1359.6980          0.0535     0.9270    0.0502
##              eCDF Max Std. Pair Dist.
## distance       0.1027          0.1302
## age            0.0432          0.8711
## educ           0.0757          0.6533
## raceblack      0.0054          0.4906
## racehispanic   0.0054          0.2057
## racewhite      0.0108          0.4377
## married        0.0108          0.7177
## nodegr         0.0649          0.2378
## re74           0.0865          0.6297
## re75           0.1081          0.7019
##
## Sample Sizes:
##           Control Treated
## All           260     185
## Matched       185     185
## Unmatched      75       0
## Discarded       0       0
```

Although balance has improved for some covariates, in general balance is still quite poor, indicating that nearest neighbor propensity score matching is not sufficient for removing confounding in this dataset. The final column, `Std. Pair Diff`, displays the average absolute within-pair difference of each covariate. When these values are small, better balance is typically achieved and estimated effects are more robust to misspecification of the outcome model

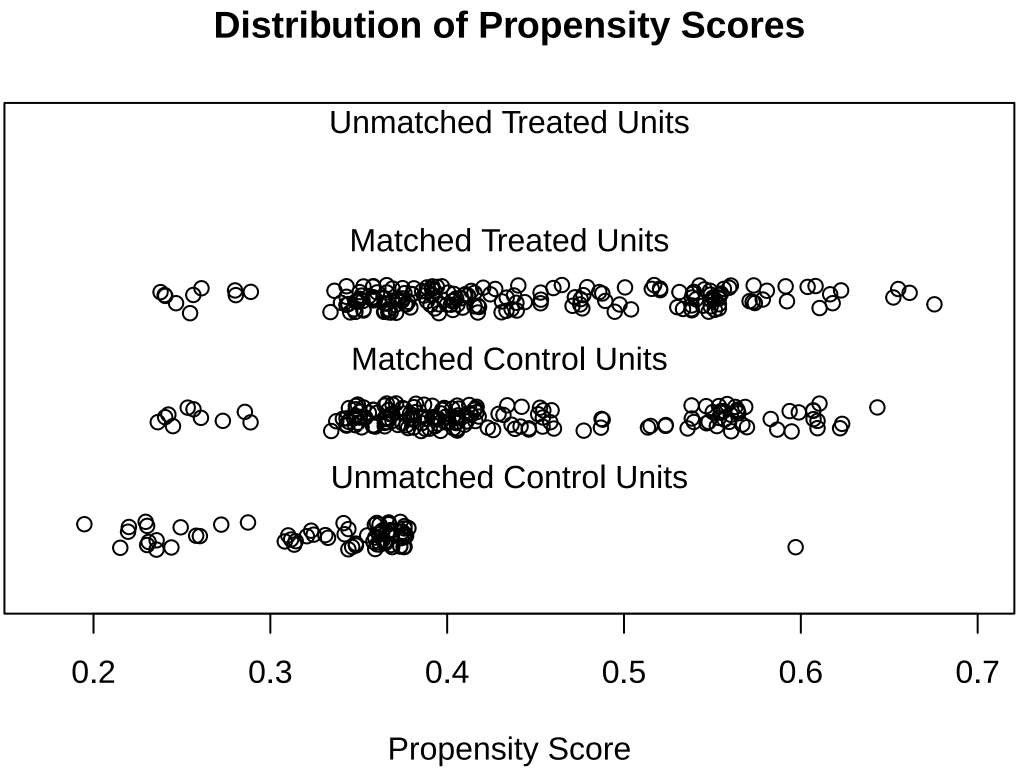
```r
plot(m.pscore.nn,
     type = "qq", interactive = FALSE,
     which.xs = c("age", "married", "re75")
)
```



```
plot(summary(m.pscore.nn))
```



```r
plot(m.pscore.nn, type = "jitter", interactive = FALSE)
```



Hybrid: Exact and Propensity Score Matching

The MatchIt package also allows us to use a hybrid technique: we can specify certain covariates to match on exactly and use nearest neighbor matches to do the rest of the work.

```
m.exact.subset <- matchit(
     treat ~ age + educ + race + nodegr +
          married + re74 + re75,
     data = lalonde, replace = TRUE,
     distance = "glm",
     exact = ~ married + race
)
```

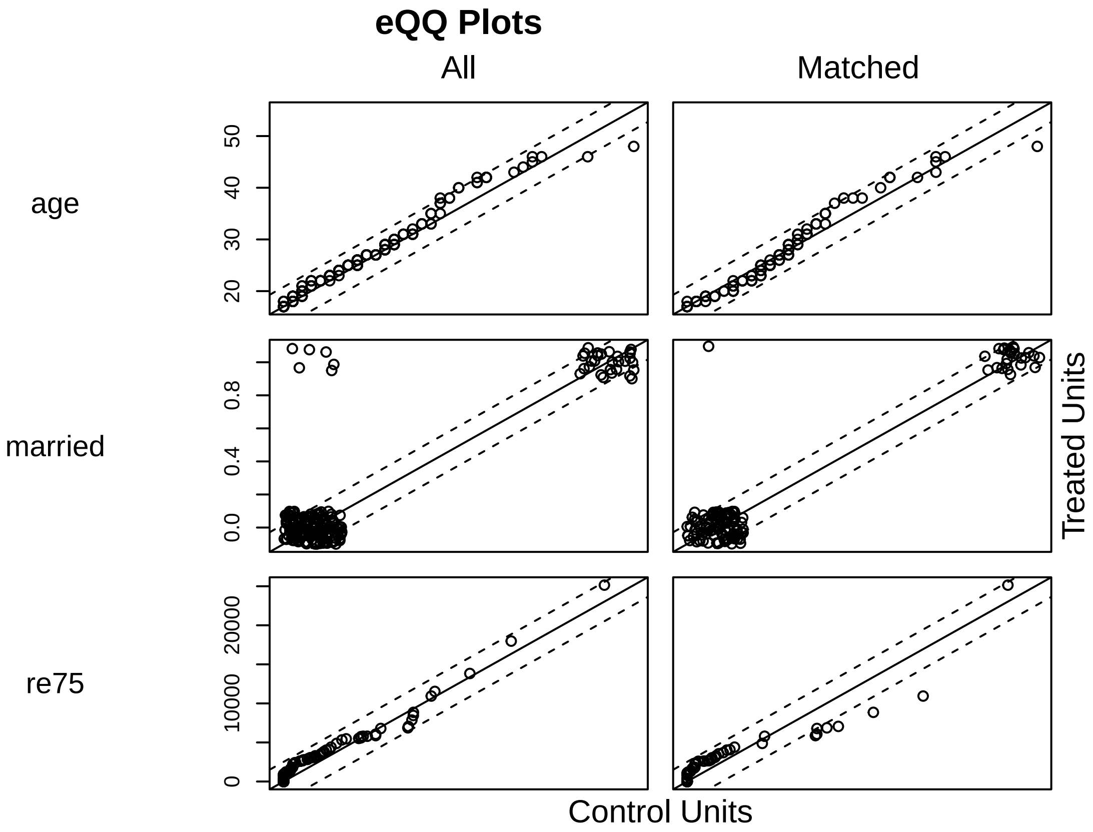
```
m.exact.subset
```

```
## A ‘matchit‘ object
##  - method: 1:1 nearest neighbor matching with replacement
##  - distance: Propensity score
##              - estimated with logistic regression
##  - number of obs.: 445 (original), 301 (matched)
##  - target estimand: ATT
##  - covariates: age, educ, race, nodegr, married, re74, re75
```

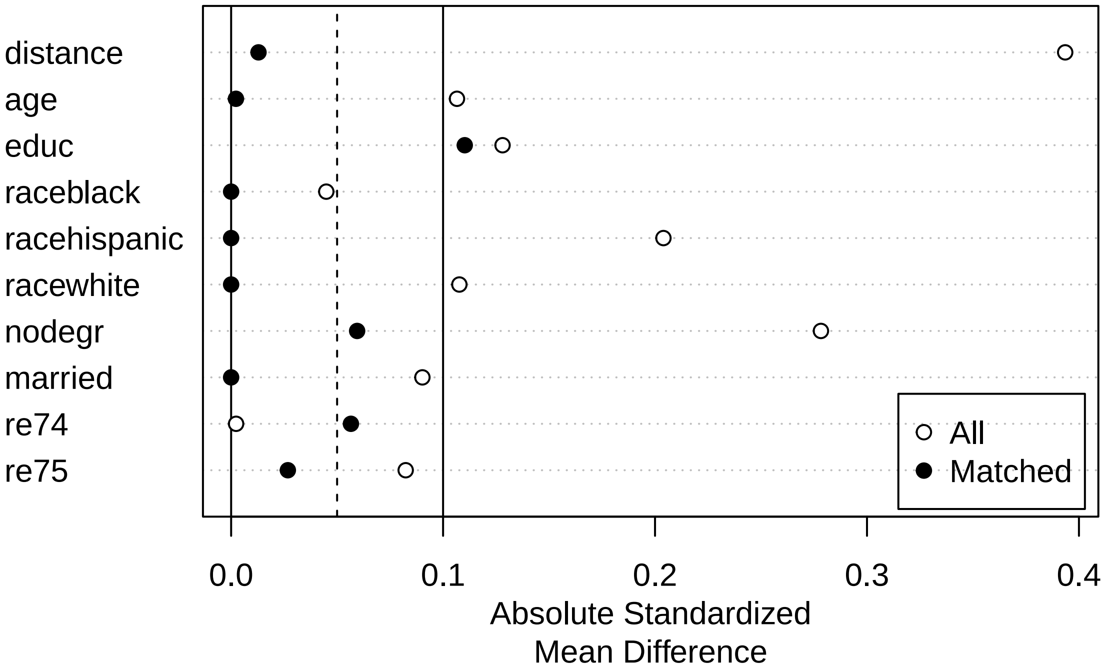
```r
summary(m.exact.subset, un = TRUE)
```

```
##
## Call:
## matchit(formula = treat ~ age + educ + race + nodegr + married +
##     re74 + re75, data = lalonde, distance = "glm", exact = ~married +
##     race, replace = TRUE)
##
## Summary of Balance for All Data:
##              Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance            0.4377        0.4001          0.3935     1.0471    0.1117
## age                25.8162       25.0538          0.1066     1.0278    0.0254
## educ               10.3459       10.0885          0.1281     1.5513    0.0287
## raceblack           0.8432        0.8269          0.0449          .    0.0163
## racehispanic        0.0595        0.1077         -0.2040          .    0.0482
## racewhite           0.0973        0.0654          0.1077          .    0.0319
## nodegr              0.7081        0.8346         -0.2783          .    0.1265
## married             0.1892        0.1538          0.0902          .    0.0353
## re74             2095.5740     2107.0268         -0.0023     0.7381    0.0192
## re75             1532.0556     1266.9092          0.0824     1.0763    0.0508
##              eCDF Max
## distance       0.2140
## age            0.0652
## educ           0.1265
## raceblack      0.0163
## racehispanic   0.0482
## racewhite      0.0319
## nodegr         0.1265
## married        0.0353
## re74           0.0471
## re75           0.1075
##
## Summary of Balance for Matched Data:
##              Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance            0.4377        0.4365          0.0129     1.0207    0.0058
## age                25.8162       25.8000          0.0023     1.2293    0.0294
## educ               10.3459       10.1243          0.1102     0.9515    0.0197
## raceblack           0.8432        0.8432         -0.0000          .    0.0000
## racehispanic        0.0595        0.0595         -0.0000          .    0.0000
## racewhite           0.0973        0.0973         -0.0000          .    0.0000
## nodegr              0.7081        0.7351         -0.0594          .    0.0270
## married             0.1892        0.1892         -0.0000          .    0.0000
## re74             2095.5740     1819.4181          0.0565     0.8989    0.0392
## re75             1532.0556     1445.8911          0.0268     0.6650    0.0607
##              eCDF Max Std. Pair Dist.
## distance       0.0270          0.0464
## age            0.0919          0.6792
## educ           0.0703          0.4974
## raceblack      0.0000          0.0000
## racehispanic   0.0000          0.0000
## racewhite      0.0000          0.0000
## nodegr         0.0270          0.1546
## married        0.0000          0.0000
## re74            0.1081          0.6037
## re75            0.1243          0.6588
##
## Sample Sizes:
##               Control Treated
## All            260.       185
## Matched (ESS)   79.78     185
## Matched        116.       185
## Unmatched      144.         0
## Discarded        0.         0
```

```
plot(m.exact.subset,
     type = "qq", interactive = FALSE,
     which.xs = c("age", "married", "re75")
)
```



```
plot(summary(m.exact.subset))
```



Genetic/MD Nearest neighbor matching

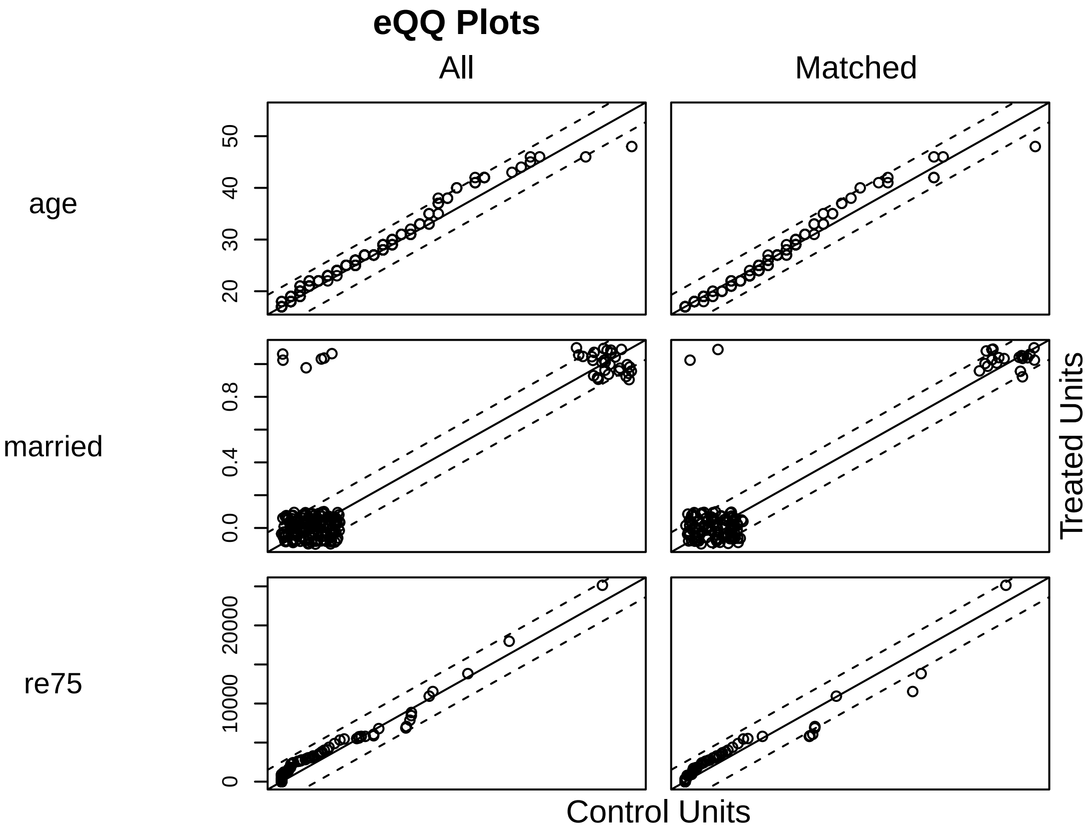
Instead of nearest neighbor matching on the propensity score, we can also use a genetic algorithm based on the Mahalanobis distance.

The argument `pop.size` must be set and is a hyper-parameter to the genetic algorithm

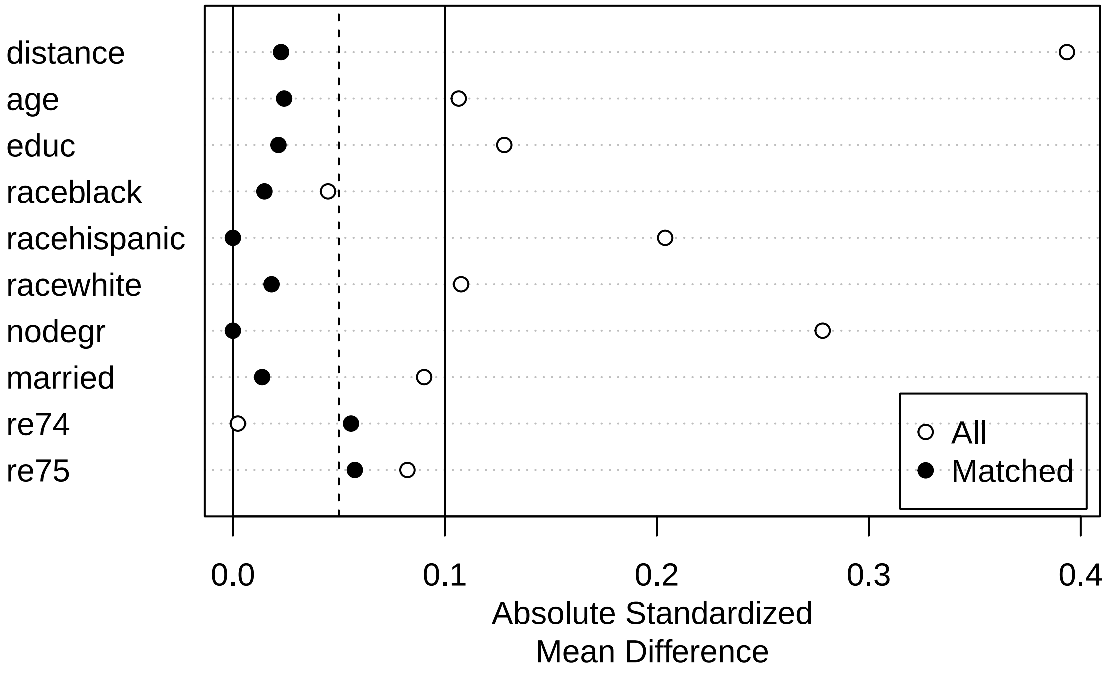
```
m.genetic.nn <- matchit(
     treat ~ age + educ + race + nodegr +
          married + re74 + re75,
     data = lalonde, replace = TRUE,
     method = "genetic",
     pop.size = 150
)
summary(m.genetic.nn, un = TRUE)
```

```
##
## Call:
## matchit(formula = treat ~ age + educ + race + nodegr + married +
##     re74 + re75, data = lalonde, method = "genetic", replace = TRUE,
##     pop.size = 150)
##
## Summary of Balance for All Data:
##                Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance              0.4377        0.4001          0.3935     1.0471    0.1117
## age                25.8162       25.0538          0.1066     1.0278    0.0254
## educ               10.3459       10.0885          0.1281     1.5513    0.0287
## raceblack           0.8432        0.8269          0.0449          .    0.0163
## racehispanic        0.0595        0.1077         -0.2040          .    0.0482
## racewhite           0.0973        0.0654          0.1077          .    0.0319
## nodegr              0.7081        0.8346         -0.2783          .    0.1265
## married             0.1892        0.1538          0.0902          .    0.0353
## re74             2095.5740     2107.0268         -0.0023     0.7381    0.0192
## re75             1532.0556     1266.9092          0.0824     1.0763    0.0508
##              eCDF Max
## distance       0.2140
## age            0.0652
## educ           0.1265
## raceblack      0.0163
## racehispanic   0.0482
## racewhite      0.0319
## nodegr         0.1265
## married        0.0353
## re74           0.0471
## re75           0.1075
##
## Summary of Balance for Matched Data:
##              Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance            0.4377        0.4355          0.0227     0.9980    0.0254
## age                25.8162       25.6432          0.0242     1.1640    0.0207
## educ               10.3459       10.3892         -0.0215     1.2938    0.0131
## raceblack           0.8432        0.8486         -0.0149          .    0.0054
## racehispanic        0.0595        0.0595         -0.0000          .    0.0000
## racewhite           0.0973        0.0919          0.0182          .    0.0054
## nodegr              0.7081        0.7081         -0.0000          .    0.0000
## married             0.1892        0.1838          0.0138          .    0.0054
## re74             2095.5740     1823.2203          0.0557     1.2560    0.0149
## re75             1532.0556     1346.8279          0.0575     0.9450    0.0474
##              eCDF Max Std. Pair Dist.
## distance       0.0811          0.2062
## age            0.0595          0.1571
## educ           0.0270          0.2097
## raceblack      0.0054          0.0149
## racehispanic   0.0000          0.0000
## racewhite      0.0054          0.0182
## nodegr         0.0000          0.0000
## married        0.0054          0.0138
## re74           0.0486          0.4809
## re75           0.0973          0.3834
##
## Sample Sizes:
##               Control Treated
## All             260.       185
## Matched (ESS)   94.81     185
## Matched         124.       185
## Unmatched       136.         0
## Discarded         0.         0
```

```
plot(m.genetic.nn,
     type = "qq", interactive = FALSE,
     which.xs = c("age", "married", "re75")
)
```



```
plot(summary(m.genetic.nn))
```



Comparison of Methods

```
fmla <- as.formula("re78 ~ treat + age + educ + race + married + nodegr + re74 + re75")
etable(
     list(
          feols(fmla, data=match.data(m.cem)),
          feols(fmla, data=match.data(m.exact.subset)),
          feols(fmla, data=match.data(m.pscore.nn)),
          feols(fmla, data=match.data(m.genetic.nn)),
          feols(fmla, data=lalonde)
     ),
     keep='treat',
     fitstat=c('n')
)
```

```
##                            model 1           model 2            model 3
## Dependent Var.:               re78              re78               re78
##
## treat           1,572.4. (865.4) 1,958.1* (798.7) 1,887.8** (670.2)
## ________________ _________________ _________________ __________________
## S.E. type                      IID               IID                IID
## Observations                   265               301                370
##
##                            model 4            model 5
## Dependent Var.:             re78              re78
##
## treat           1,625.9* (825.6) 1,676.3** (638.7)
## _______________ ________________ _________________
## S.E. type                    IID               IID
## Observations                 309               445
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```