Lab 5: Appendix

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.

```
library("MatchIt")
library("broom")
library(estimatr)
```

```
## Warning: package 'estimatr' was built under R version 4.2.3
```

```
data("lalonde")
head(lalonde)
```

```
##      treat age educ   race married nodegree re74 re75       re78
## NSW1     1  37   11  black       1        1    0    0  9930.0460
## NSW2     1  22    9 hispan       0        1    0    0  3595.8940
## NSW3     1  30   12  black       0        0    0    0 24909.4500
## NSW4     1  27   11  black       0        1    0    0  7506.1460
## NSW5     1  33    8  black       0        1    0    0   289.7899
## NSW6     1  22    9  black       0        1    0    0  4056.4940
```

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. In particular, the analysis is concerned with the marginal, total effect of the treatment for those who actually received the treatment.

The planning phase of a matching analysis involves selecting the type of effect to be estimated, selecting the target population to which the treatment effect is to generalize, and selecting the covariates for which balance is required for an unbiased estimate of the treatment effect. After planning and prior to 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:

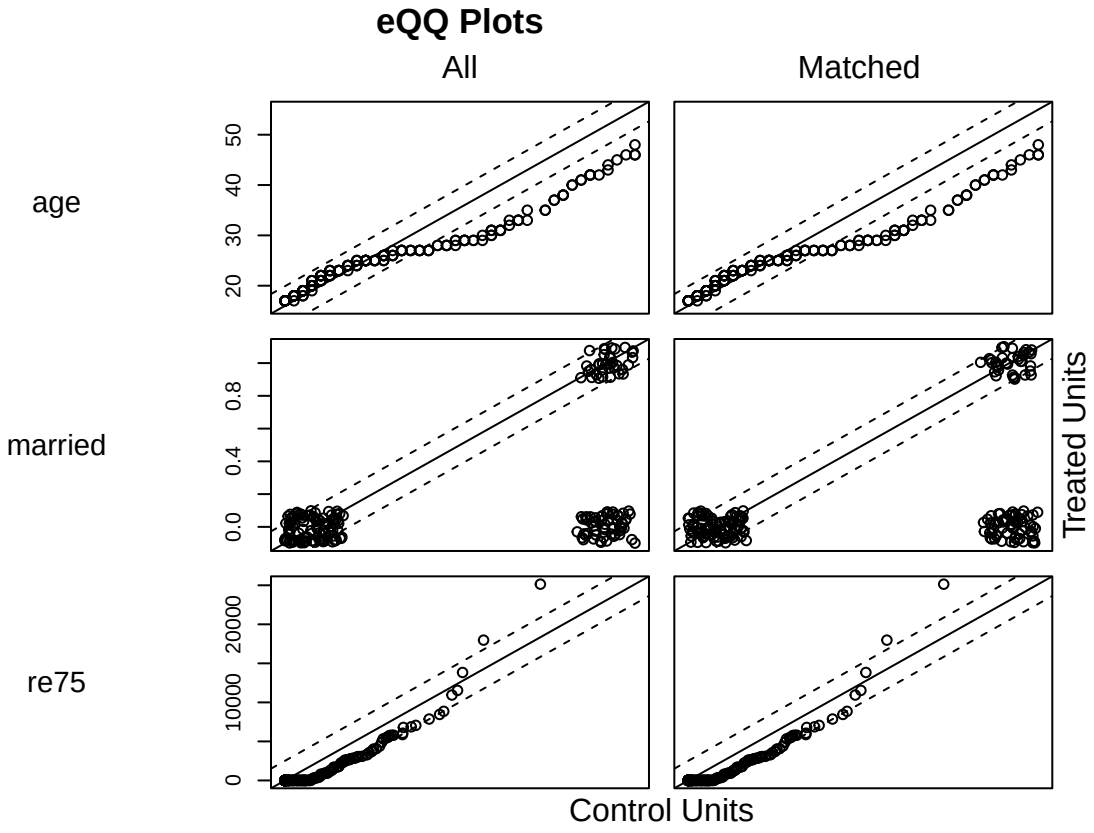
```
m.out0 <- matchit(treat ~ age + educ + race + married +
                    nodegree + re74 + re75, data = lalonde,
                  method = NULL, distance = "glm")
```

```
# Checking balance prior to matching
summary(m.out0)
```

```
##
## Call:
```

```
## matchit(formula = treat ~ age + educ + race + married + nodegree +
##     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          0.5774        0.1822          1.7941     0.9211    0.3774
## age              25.8162       28.0303         -0.3094     0.4400    0.0813
## educ             10.3459       10.2354          0.0550     0.4959    0.0347
## raceblack         0.8432        0.2028          1.7615          .    0.6404
## racehispan        0.0595        0.1422         -0.3498          .    0.0827
## racewhite         0.0973        0.6550         -1.8819          .    0.5577
## 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
## distance     0.6444
## age          0.1577
## educ         0.1114
## raceblack    0.6404
## racehispan   0.0827
## racewhite    0.5577
## married      0.3236
## nodegree     0.1114
## re74         0.4470
## re75         0.2876
##
## Sample Sizes:
##           Control Treated
## All           429     185
## Matched       429     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.

Now, we will perform 1:1 nearest neighbor (NN) matching on the propensity score, which is appropriate for estimating the ATT. 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.

```
# 1:1 NN PS matching w/o replacement
m.out1 <- matchit(treat ~ age + educ + race + married +
                    nodegree + re74 + re75, data = lalonde,
                  method = "nearest", distance = "glm")
```

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.out1
```

```
## A matchit object
##  - method: 1:1 nearest neighbor matching without replacement
##  - distance: Propensity score
```

```
##               - estimated with logistic regression
##  - number of obs.: 614 (original), 370 (matched)
##  - target estimand: ATT
##  - covariates: age, educ, race, married, nodegree, re74, re75
```

The key components of the `m.out1` object are `weights` (the computed matching weights), `subclass` (matching pair membership), `distance` (the estimated propensity score), and `match.matrix` (which control units are matched to each treated unit).

```
# Checking balance after NN matching
summary(m.out1, un = FALSE)
```

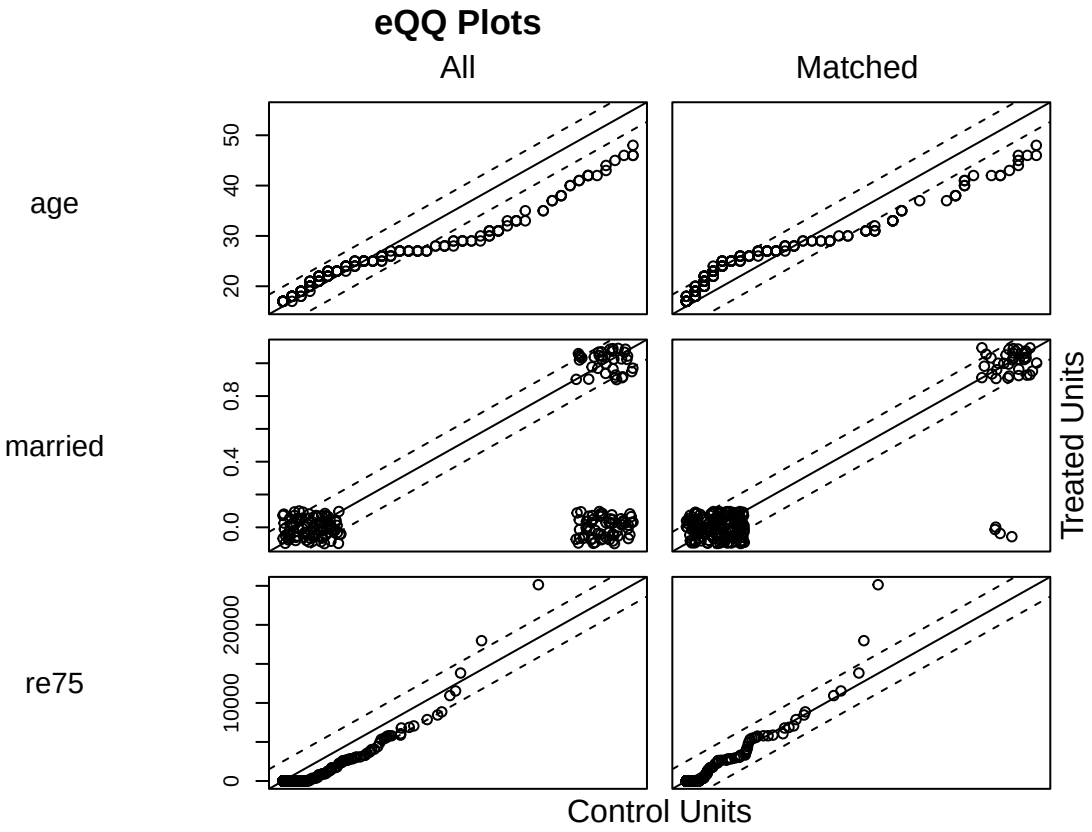
```
##
## Call:
## matchit(formula = treat ~ age + educ + race + married + nodegree +
##     re74 + re75, data = lalonde, method = "nearest", distance = "glm")
##
## Summary of Balance for Matched Data:
##            Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance          0.5774        0.3629          0.9739     0.7566    0.1321
## age              25.8162       25.3027          0.0718     0.4568    0.0847
## educ             10.3459       10.6054         -0.1290     0.5721    0.0239
## raceblack         0.8432        0.4703          1.0259          .    0.3730
## racehispan        0.0595        0.2162         -0.6629          .    0.1568
## racewhite         0.0973        0.3135         -0.7296          .    0.2162
## married           0.1892        0.2108         -0.0552          .    0.0216
## nodegree          0.7081        0.6378          0.1546          .    0.0703
## re74           2095.5737     2342.1076         -0.0505     1.3289    0.0469
## re75           1532.0553     1614.7451         -0.0257     1.4956    0.0452
##            eCDF Max Std. Pair Dist.
## distance     0.4216          0.9740
## age          0.2541          1.3938
## educ         0.0757          1.2474
## raceblack    0.3730          1.0259
## racehispan   0.1568          1.0743
## racewhite    0.2162          0.8390
## married      0.0216          0.8281
## nodegree     0.0703          1.0106
## re74         0.2757          0.7965
## re75         0.2054          0.7381
##
## Sample Sizes:
##           Control Treated
## All           429     185
## Matched       185     185
## Unmatched     244       0
## Discarded       0       0
```

To assess the quality of the resulting matches numerically, we can use the `summary()` function on `m.out1` as before. Here we set `un = FALSE` to suppress display of the balance before matching for brevity and because we already saw it. (Leaving it as `TRUE`, its default, would display balance both before and after matching.)

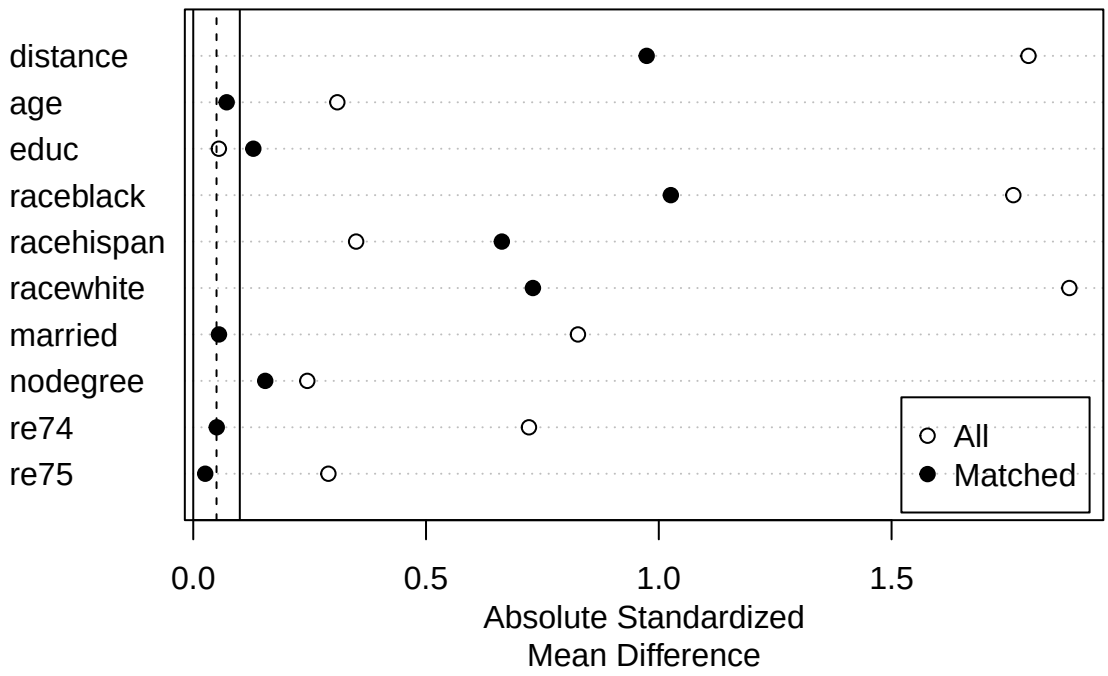
Although balance has improved for some covariates, in general balance is still quite poor, indicating that nearest neighbor propensity score matching is not sufficient for removing confounding in this dataset. The

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

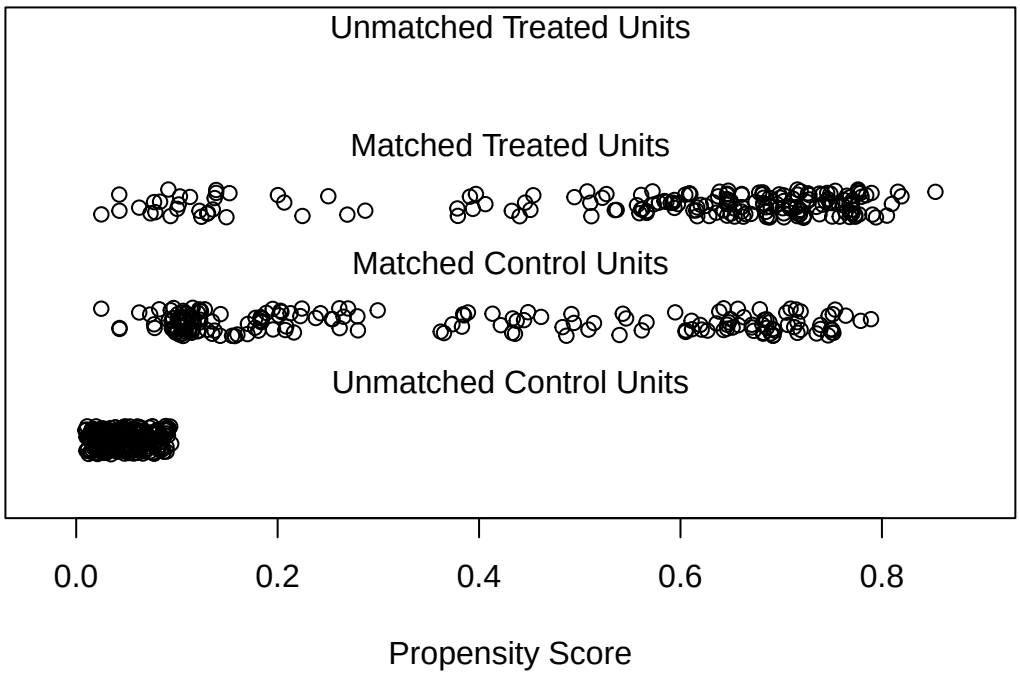
```
plot(m.out1, type = "qq", interactive = FALSE,
     which.xs = c("age", "married", "re75"))
```




```
plot(summary(m.out1))
```




```
plot(m.out1, type = "jitter", interactive = FALSE)
```



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.

```
# Exact Matching
m.out2 <- matchit(treat ~ age + educ + race + married +
                   nodegree + re74 + re75, data = lalonde,
                  method = "exact", distance = "glm")
```

```
m.out2
```

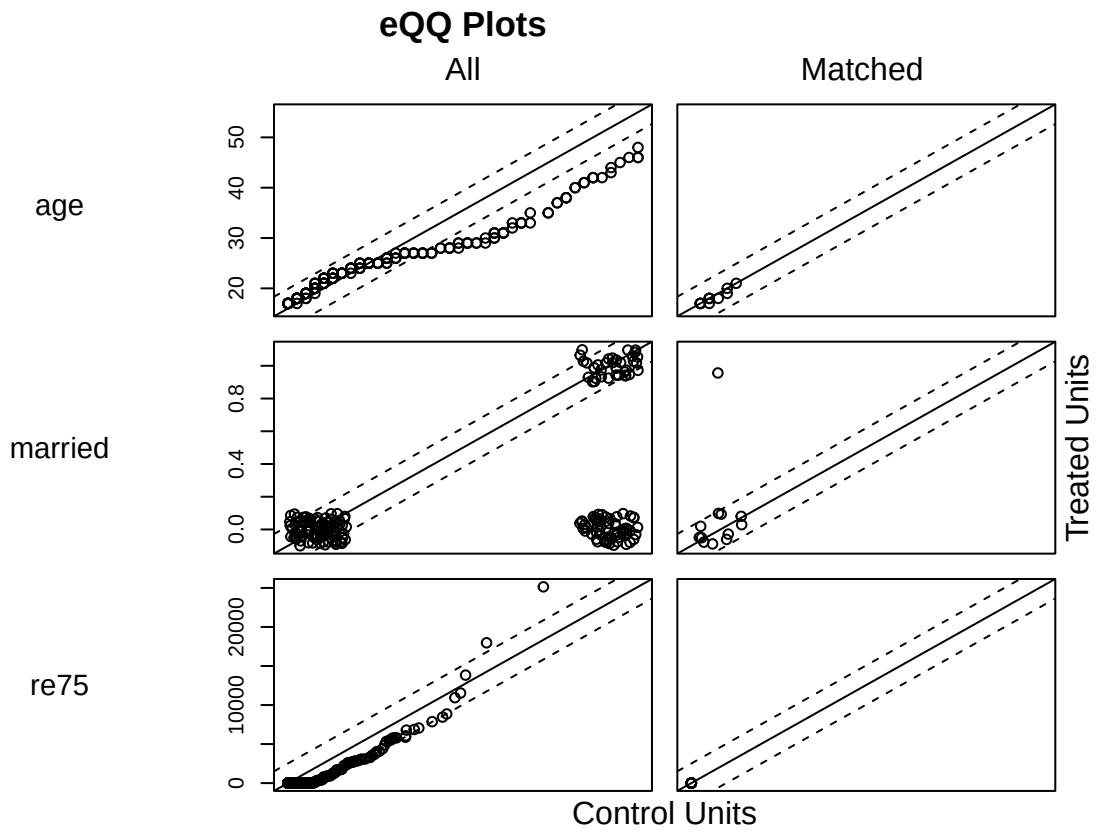
```
## A matchit object
##  - method: Exact matching
##  - number of obs.: 614 (original), 25 (matched)
##  - target estimand: ATT
##  - covariates: age, educ, race, married, nodegree, re74, re75
```

```
summary(m.out2)
```

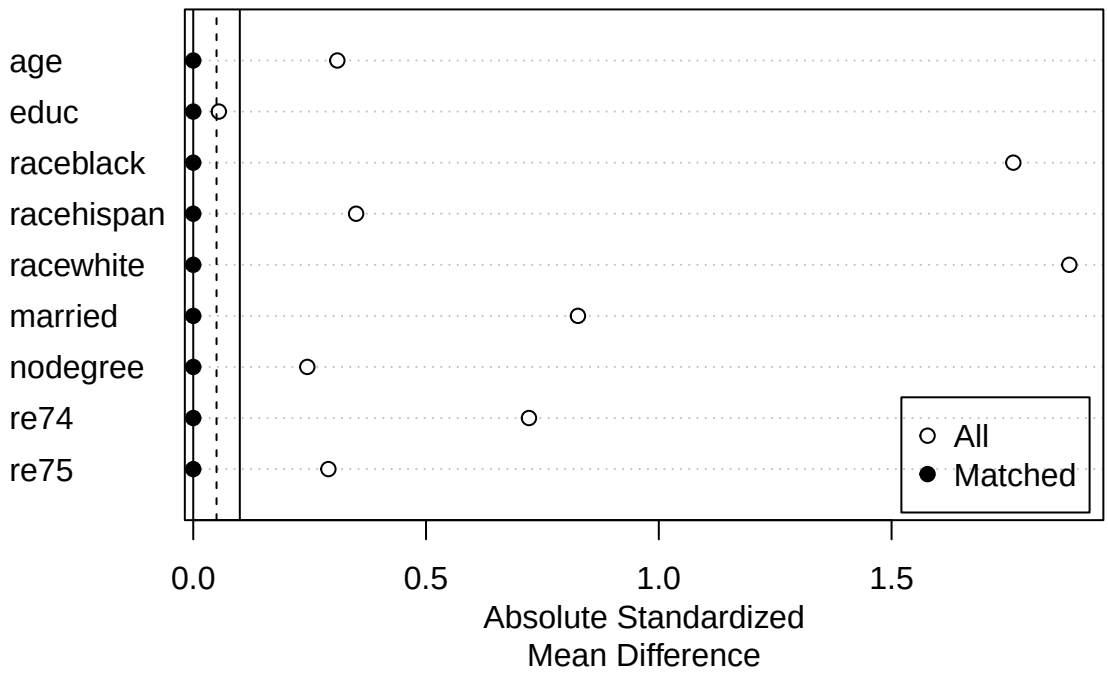
```
##
## Call:
```

```
## matchit(formula = treat ~ age + educ + race + married + nodegree +
##     re74 + re75, data = lalonde, method = "exact", distance = "glm")
##
## 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
## raceblack         0.8432        0.2028          1.7615          .    0.6404
## racehispan        0.0595        0.1422         -0.3498          .    0.0827
## racewhite         0.0973        0.6550         -1.8819          .    0.5577
## 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
## raceblack    0.6404
## racehispan   0.0827
## racewhite    0.5577
## 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              17.9231       17.9231               0     0.9712         0
## educ             10.1538       10.1538               0     0.9712         0
## raceblack         1.0000        1.0000               0          .         0
## racehispan        0.0000        0.0000               0          .         0
## racewhite         0.0000        0.0000               0          .         0
## married           0.0000        0.0000               0          .         0
## nodegree          0.8462        0.8462               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
## racehispan        0               0
## racewhite         0               0
## married           0               0
## nodegree          0               0
## re74              0               0
## re75              0               0
##
## Sample Sizes:
##               Control Treated
## All            429.       185
## Matched (ESS)    9.66      13
## Matched         12.        13
## Unmatched      417.       172
## Discarded        0.         0
```

```
plot(m.out2, type = "qq", interactive = FALSE,
     which.xs = c("age", "married", "re75"))
```




```
plot(summary(m.out2))
```



Matching using exact attribute for some variables

```
m.out3 <- matchit(treat ~ age + educ + race + nodegree +
                    married + re74 + re75, data = lalonde, replace = TRUE,
                    distance = "glm",
                    exact = ~ married + race)
```

```
m.out3
```

```
## A matchit object
##  - method: 1:1 nearest neighbor matching with replacement
##  - distance: Propensity score
##              - estimated with logistic regression
##  - number of obs.: 614 (original), 268 (matched)
##  - target estimand: ATT
##  - covariates: age, educ, race, nodegree, married, re74, re75
```

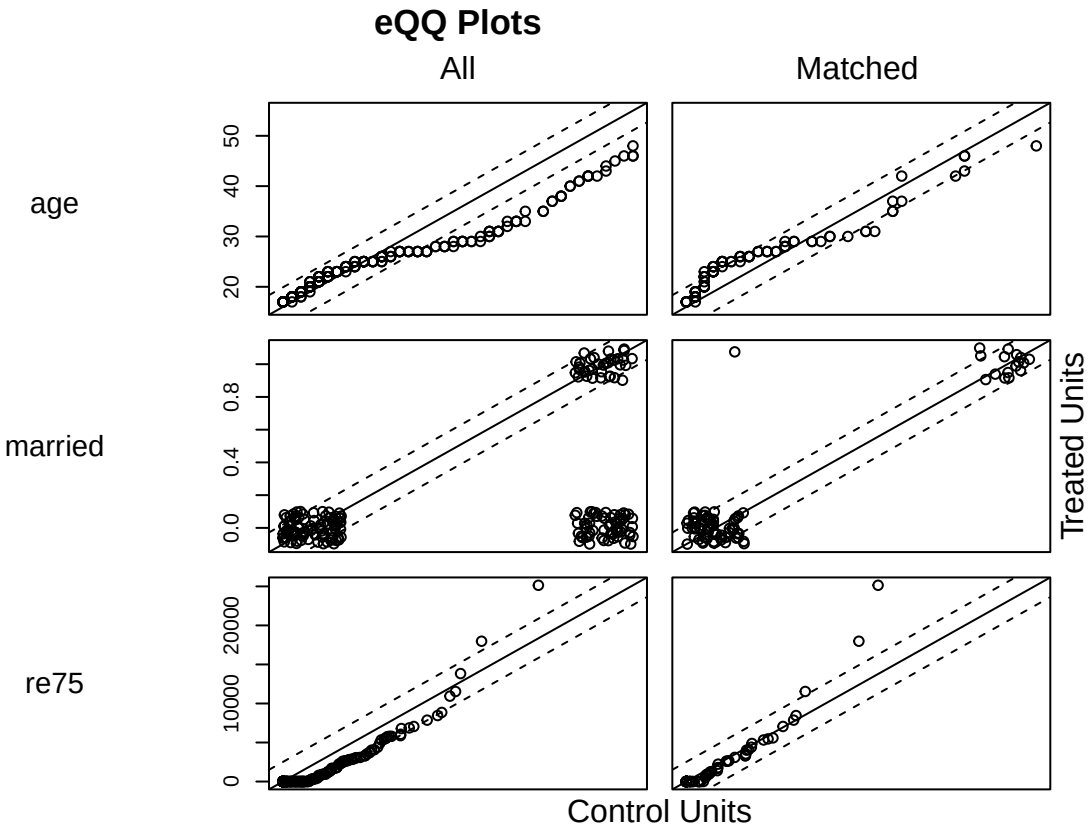
```
summary(m.out3, un = TRUE)
```

```
##
## Call:
## matchit(formula = treat ~ age + educ + race + nodegree + married +
##     re74 + re75, data = lalonde, distance = "glm", exact = ~married +
##     race, replace = TRUE)
##
```

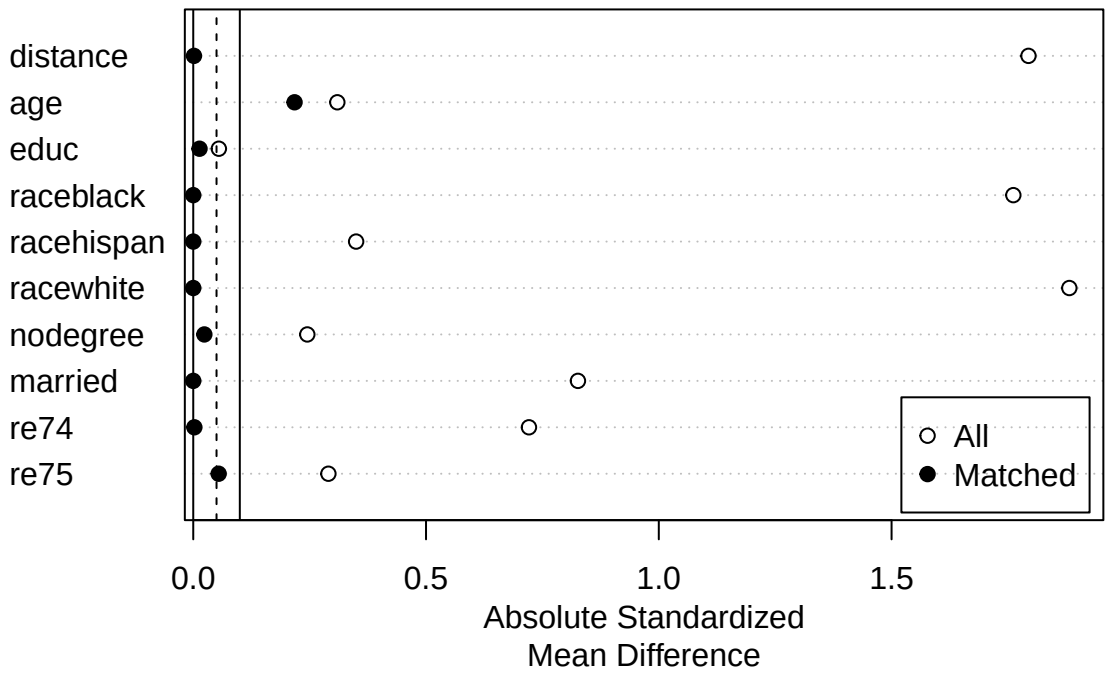
```
## Summary of Balance for All Data:
##            Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance          0.5774        0.1822          1.7941     0.9211    0.3774
## age              25.8162       28.0303         -0.3094     0.4400    0.0813
## educ             10.3459       10.2354          0.0550     0.4959    0.0347
## raceblack         0.8432        0.2028          1.7615          .    0.6404
## racehispan        0.0595        0.1422         -0.3498          .    0.0827
## racewhite         0.0973        0.6550         -1.8819          .    0.5577
## nodegree          0.7081        0.5967          0.2450          .    0.1114
## married           0.1892        0.5128         -0.8263          .    0.3236
## re74           2095.5737     5619.2365         -0.7211     0.5181    0.2248
## re75           1532.0553     2466.4844         -0.2903     0.9563    0.1342
##            eCDF Max
## distance     0.6444
## age          0.1577
## educ         0.1114
## raceblack    0.6404
## racehispan   0.0827
## racewhite    0.5577
## nodegree     0.1114
## married      0.3236
## re74         0.4470
## re75         0.2876
##
## Summary of Balance for Matched Data:
##            Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## distance          0.5774        0.5771          0.0015     0.9947    0.0053
## age              25.8162       24.2595          0.2176     0.6151    0.0730
## educ             10.3459       10.3730         -0.0134     0.6207    0.0151
## raceblack         0.8432        0.8432          0.0000          .    0.0000
## racehispan        0.0595        0.0595         -0.0000          .    0.0000
## racewhite         0.0973        0.0973         -0.0000          .    0.0000
## nodegree          0.7081        0.7189         -0.0238          .    0.0108
## married           0.1892        0.1892         -0.0000          .    0.0000
## re74           2095.5737     2084.3104          0.0023     1.1683    0.0409
## re75           1532.0553     1707.6677         -0.0546     1.5525    0.0615
##            eCDF Max Std. Pair Dist.
## distance     0.0541          0.0294
## age          0.3027          1.0833
## educ         0.0378          1.0189
## raceblack    0.0000          0.0000
## racehispan   0.0000          0.0000
## racewhite    0.0000          0.0000
## nodegree     0.0108          0.8798
## married      0.0000          0.0000
## re74         0.2216          0.4847
## re75         0.2000          0.7177
##
## Sample Sizes:
##               Control Treated
## All            429.       185
## Matched (ESS)   43.49     185
## Matched         83.       185
## Unmatched      346.         0
```

```
## Discarded        0.          0
```

```
plot(m.out3, type = "qq", interactive = FALSE,
     which.xs = c("age", "married", "re75"))
```




```
plot(summary(m.out3))
```



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. When doing CEM, there are three mains steps:

1. Coarsen the data to reduce the level of granularity. This means binning numerical values and/or grouping categorical values.
2. Apply an exact matching on the coarsened data to find comparable control and treatment groups. This means finding all combinations of the covariates that have at least one control and one treatment record and keep records that belong to the combinations and drop the rest. Each combination is referred to as *stratum*.
3. Estimate the causal impact using the matched data.

Let's assume we wanted to understand the causal impact of `treat` on `re78` using this method

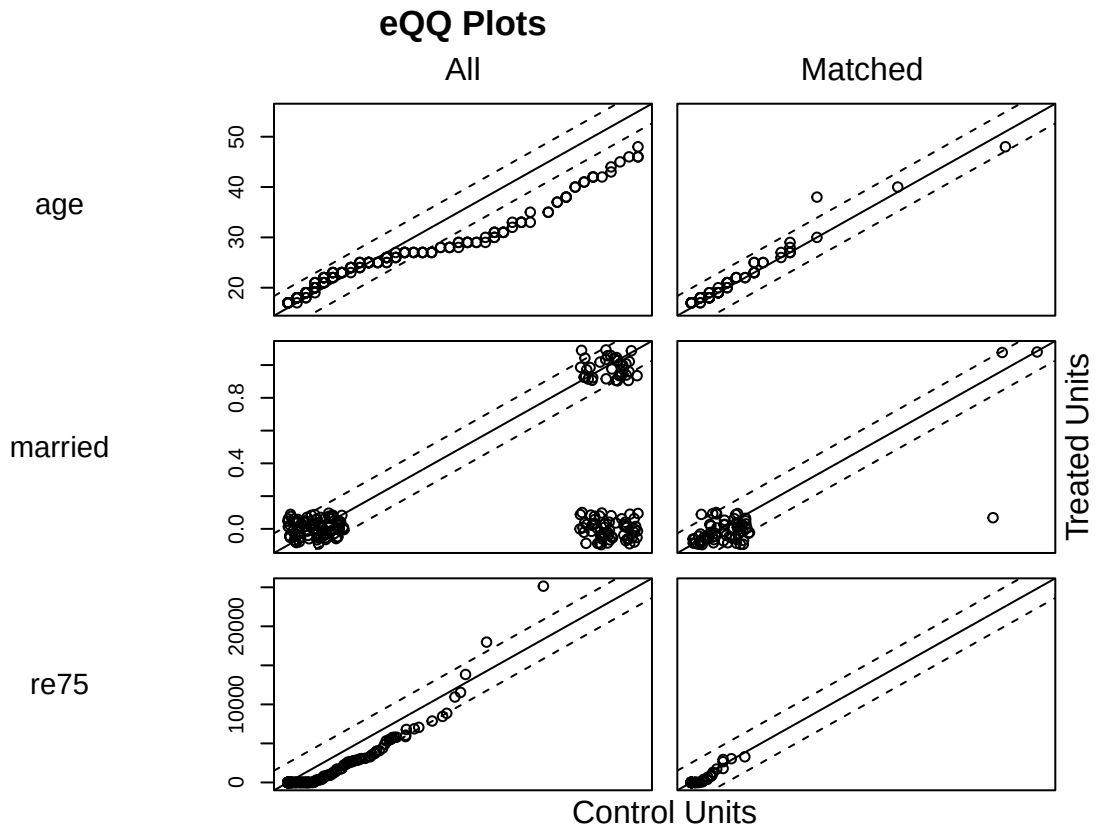
```
m.out4 <- matchit(treat ~ age + educ + race + married +
                  nodegree + re74 + re75, data = lalonde,
                    method = 'cem', estimand = 'ATE')

summary(m.out4, un=FALSE)
```

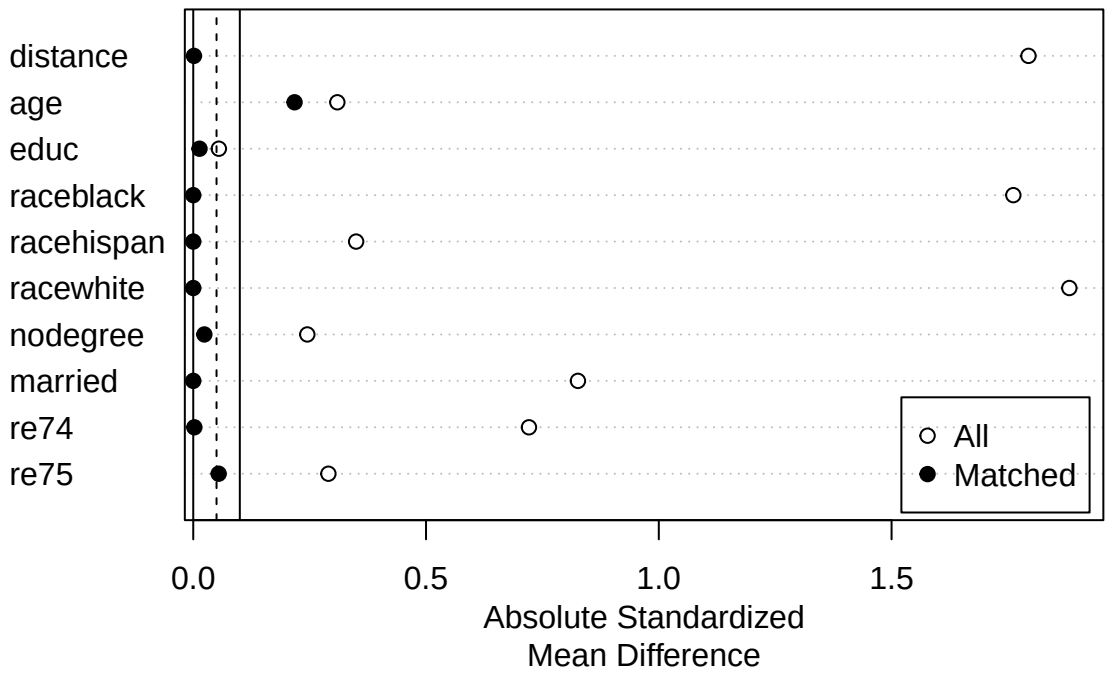
```
##
## Call:
## matchit(formula = treat ~ age + educ + race + married + nodegree +
```

```
##     re74 + re75, data = lalonde, method = "cem", estimand = "ATE")
##
## Summary of Balance for Matched Data:
##            Means Treated Means Control Std. Mean Diff. Var. Ratio eCDF Mean
## age              20.9760       20.5292          0.0488     0.8710    0.0149
## educ             10.1711       10.1298          0.0167     0.9033    0.0080
## raceblack         0.7429        0.7429         -0.0000          .    0.0000
## racehispan        0.0357        0.0357          0.0000          .    0.0000
## racewhite         0.2214        0.2214          0.0000          .    0.0000
## married           0.0500        0.0500          0.0000          .    0.0000
## nodegree          0.7000        0.7000          0.0000          .    0.0000
## re74            447.6595      744.7416         -0.0502     1.3572    0.0548
## re75            348.5106      520.1616         -0.0527     0.8601    0.0423
##            eCDF Max Std. Pair Dist.
## age          0.1401          0.1073
## educ         0.0629          0.1587
## raceblack    0.0000          0.0000
## racehispan   0.0000          0.0000
## racewhite    0.0000          0.0000
## married      0.0000          0.0000
## nodegree     0.0000          0.0000
## re74         0.3572          0.0800
## re75         0.1936          0.1617
##
## Sample Sizes:
##               Control Treated
## All            429.    185.
## Matched (ESS)   63.78   46.42
## Matched         75.     65.
## Unmatched      354.    120.
## Discarded        0.      0.
```

```
plot(m.out4, type = "qq", interactive = FALSE,
     which.xs = c("age", "married", "re75"))
```




```
plot(summary(m.out3))
```




```
m.data1 <- match.data(m.out1)

head(m.data1)
```

```
##      treat age educ   race married nodegree re74 re75       re78  distance
## NSW1     1  37   11  black       1        1    0    0  9930.0460 0.6387699
## NSW2     1  22    9 hispan       0        1    0    0  3595.8940 0.2246342
## NSW3     1  30   12  black       0        0    0    0 24909.4500 0.6782439
## NSW4     1  27   11  black       0        1    0    0  7506.1460 0.7763241
## NSW5     1  33    8  black       0        1    0    0   289.7899 0.7016387
## NSW6     1  22    9  black       0        1    0    0  4056.4940 0.6990699
##      weights subclass
## NSW1       1        1
## NSW2       1       98
## NSW3       1      109
## NSW4       1      120
## NSW5       1      131
## NSW6       1      142
```

```
tidy(lm_robust(re78 ~ treat + age + educ + race + married + nodegree +
             re74 + re75, data = m.data1))
```

```
##          term      estimate    std.error  statistic     p.value      conf.low
## 1 (Intercept) -2581.6442386 3322.5932441 -0.7769968 0.437670920 -9115.7745769
## 2       treat  1344.9356054  756.2623146  1.7783983 0.076182344  -142.3113040
```

```
## 3          age     7.8035414   42.6294067  0.1830554 0.854857630   -76.0304038
## 4         educ   602.2031926  214.7715980  2.8039238 0.005322128   179.8386384
## 5   racehispan  1533.4786344 1030.9876087  1.4873880 0.137787687  -494.0362967
## 6    racewhite   469.4336863  890.8725128  0.5269370 0.598561880 -1282.5343355
## 7      married  -158.2545481  957.3235529 -0.1653094 0.868793277 -2040.9035547
## 8     nodegree   923.2840337 1143.5093227  0.8074128 0.419961853 -1325.5133389
## 9         re74     0.0263618    0.1737832  0.1516936 0.879513539    -0.3153959
## 10        re75     0.2206775    0.1682732  1.3114241 0.190550314    -0.1102444
##       conf.high  df outcome
## 1  3952.4860997 360    re78
## 2  2832.1825148 360    re78
## 3    91.6374865 360    re78
## 4  1024.5677468 360    re78
## 5  3560.9935655 360    re78
## 6  2221.4017082 360    re78
## 7  1724.3944586 360    re78
## 8  3172.0814063 360    re78
## 9     0.3681195 360    re78
## 10    0.5515995 360    re78
```

```
m.data3 <- match.data(m.out3)

head(m.data3)
```

```
##      treat age educ   race married nodegree re74 re75       re78  distance
## NSW1     1  37   11  black       1        1    0    0  9930.0460 0.6387699
## NSW2     1  22    9 hispan       0        1    0    0  3595.8940 0.2246342
## NSW3     1  30   12  black       0        0    0    0 24909.4500 0.6782439
## NSW4     1  27   11  black       0        1    0    0  7506.1460 0.7763241
## NSW5     1  33    8  black       0        1    0    0   289.7899 0.7016387
## NSW6     1  22    9  black       0        1    0    0  4056.4940 0.6990699
##      weights
## NSW1       1
## NSW2       1
## NSW3       1
## NSW4       1
## NSW5       1
## NSW6       1
```

```
tidy(lm_robust(re78 ~ treat + age + educ + race + married + nodegree +
            re74 + re75, data = m.data3))
```

```
##           term      estimate    std.error  statistic    p.value      conf.low
## 1  (Intercept) -3.103791e+03 3741.5981142 -0.8295361 0.40756797 -1.047175e+04
## 2        treat  1.707324e+03  878.6513824  1.9431190 0.05308943 -2.291732e+01
## 3          age  4.240302e+01   59.2108221  0.7161364 0.47455484 -7.419501e+01
## 4         educ  5.819425e+02  228.6622111  2.5449877 0.01151122  1.316606e+02
## 5   racehispan  8.549743e+02 1560.7121802  0.5478104 0.58429573 -2.218382e+03
## 6    racewhite  1.230372e+03 1098.3499485  1.1202005 0.26367003 -9.325001e+02
## 7      married -2.959291e+02 1199.9900841 -0.2466096 0.80540649 -2.658951e+03
## 8     nodegree  2.514433e+02 1362.6903698  0.1845198 0.85375068 -2.431968e+03
## 9         re74  4.688075e-02    0.1936282  0.2421174 0.80888151 -3.344121e-01
## 10        re75  1.562123e-01    0.1785333  0.8749756 0.38240096 -1.953558e-01
```

```
##       conf.high  df outcome
## 1  4264.1694981 258    re78
## 2  3437.5657210 258    re78
## 3   159.0010553 258    re78
## 4  1032.2244551 258    re78
## 5  3928.3309119 258    re78
## 6  3393.2443964 258    re78
## 7  2067.0930784 258    re78
## 8  2934.8550688 258    re78
## 9     0.4281736 258    re78
## 10    0.5077804 258    re78
```

```
m.data4 <- match.data(m.out4)

head(m.data4)
```

```
##       treat age educ   race married nodegree re74 re75     re78   weights
## NSW2      1  22    9 hispan       0        1    0    0 3595.894 0.9285714
## NSW4      1  27   11  black       0        1    0    0 7506.146 0.5571429
## NSW6      1  22    9  black       0        1    0    0 4056.494 0.9285714
## NSW7      1  23   12  black       0        0    0    0    0.000 0.6706349
## NSW9      1  22   16  black       0        0    0    0 2164.022 0.9285714
## NSW11     1  19    9  black       0        1    0    0 8173.908 1.0059524
##       subclass
## NSW2        22
## NSW4        25
## NSW6        15
## NSW7        17
## NSW9        20
## NSW11        6
```

```
tidy(lm_robust(re78 ~ treat + age + educ + race + married + nodegree +
            re74 + re75, data = m.data4))
```

```
##           term      estimate    std.error  statistic   p.value      conf.low
## 1  (Intercept) -9679.4067810 10443.409483 -0.9268436 0.3557253 -30340.443469
## 2        treat  1415.5379180  1156.677890  1.2237961 0.2232426   -872.810955
## 3          age   213.1326465   168.674191  1.2635759 0.2086435   -120.569067
## 4         educ   757.0689098   594.597854  1.2732453 0.2052031   -419.271833
## 5   racehispan  1627.2174349  2307.366131  0.7052272 0.4819305  -2937.630507
## 6    racewhite   867.0397820  1231.371874  0.7041250 0.4826143  -1569.082206
## 7      married -3334.9337139  2174.426980 -1.5337069 0.1275319  -7636.777444
## 8     nodegree  1949.3865230  2697.505926  0.7226626 0.4711853  -3387.306345
## 9         re74    -0.2131667     0.585376 -0.3641535 0.7163350     -1.371263
## 10        re75     2.4298735     1.070638  2.2695574 0.0248812      0.311745
##       conf.high  df outcome
## 1  1.098163e+04 130    re78
## 2  3.703887e+03 130    re78
## 3  5.468344e+02 130    re78
## 4  1.933410e+03 130    re78
## 5  6.192065e+03 130    re78
## 6  3.303162e+03 130    re78
## 7  9.669100e+02 130    re78
```

```
## 8  7.286079e+03 130    re78
## 9  9.449297e-01 130    re78
## 10 4.548002e+00 130    re78
```