CRSP 500/PQHS 500 - Spring 2018 – The **match1** Example

The match1.csv data file may be found in the Data and Code page on our web site

Using the **Matching** package in R to do Propensity Score Matching

We'll use a small, simulated example to demonstrate some of the tools available in the Matching package. The data can be found in the **match1.csv** data file on the course website. In total, there are 300 observations – the first five of which are shown below…

subject	exposed	age	female	race	comorbid	serumK	wbc	outcome
1	no	45	0	4	9	4.6	7750	139
2	yes	29	1	4	4	4.3	6200	117
3	no	61	1	1	6	4.1	6650	147
4	yes	58	1	3	1	4.7	9370	99
5	yes	33	1	4	7	4.4	9860	121

The subjects are identified by ID #s in the **subject** variable (1-300), and we have **exposure** status (78 "yes" and 222 "no") as well as each subject's **age, female, race** (4 categories, labeled 1, 2, 3 and 4), **# of comorbid** illnesses (out of a possible 9), **serumK** (serum potassium level), **wbc** (white blood cell count) and **outcome** (better results = higher outcome values).

We're going to fit a logistic regression model to predict propensity for exposure on the basis of the main effects of six covariates: **age, female, race** (treated as a factor), **comorbid** (treated as a count), **serumK,** and **wbc**

Then, we'll match exposed to unexposed patients using the Matching library, and look at [1] how effectively we balance the distributions of those covariates, and [2] obtain an average treatment effect on the treated (ATT) estimate for the causal effect of the **exposure** on **outcome** under each matching approach.

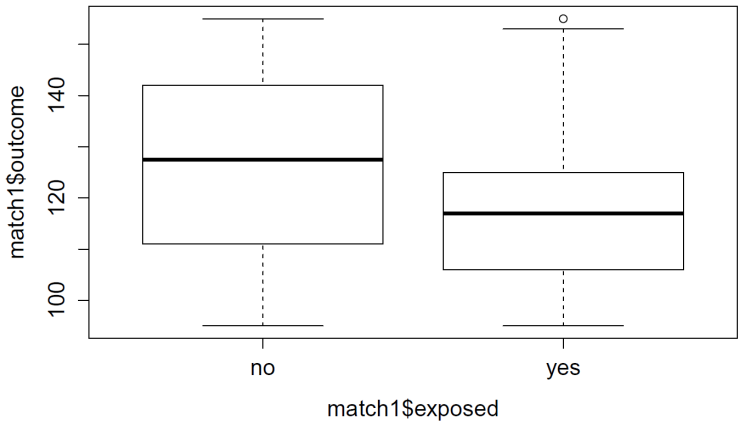
The matching approaches we'll demonstrate are:

1. 1:1 matching on the propensity score, without replacement. [We'll do this one in detail.]
2. 1:2 matching on the propensity score, without replacement.
3. 1:3 matching on the propensity score, without replacement.
4. 1:1 matching on the propensity score, with replacement.
5. 1:1 matching on the propensity score, with replacement, within groups defined by race.
6. 1:1 matching on the propensity score, with replacement, within a caliper
7. Genetic matching, which automatically finds balance by using a genetic search algorithm to determine the optimal weight for each covariate within the matching algorithm.

Results of walking through the R script, available on the course website as **match1script2017.R** follow…

To start, we'll evaluate the unadjusted estimate of the impact of exposure on outcome.

```
> plot(match1$outcome ~
match1$exposed)
```




```
> by(match1$outcome, match1$exposed, summary)
match1$exposed: no
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   95.0   111.0   127.5   126.3   142.0   155.0
------------------------------------------------------
match1$exposed: yes
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   95.0   106.0   117.0   117.6   125.0   155.0

> t.test(match1$outcome ~ match1$exposed)

Welch Two Sample t-test

data:  match1$outcome by match1$exposed
t = 4.1509, df = 153.278, p-value = 5.476e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:  4.551881 12.819567
sample estimates:
mean in group no mean in group yes
        126.2883          117.6026

> modelx <- lm(match1$outcome ~ match1$exposed)
> summary(modelx); confint(modelx)

Call: lm(formula = match1$outcome ~ match1$exposed)

Residuals:     Min      1Q  Median      3Q     Max
          -31.288 -13.774  -0.445  13.712  37.397

Coefficients:     Estimate Std. Error t value Pr(>|t|)
(Intercept)        126.288      1.140 110.817  < 2e-16 ***
match1$exposedyes   -8.686      2.235  -3.886 0.000125 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.98 on 298 degrees of freedom
Multiple R-squared: 0.04824,       Adjusted R-squared: 0.04504
F-statistic:  15.1 on 1 and 298 DF,  p-value: 0.0001255

                      2.5 %     97.5 %
(Intercept)       124.04558 128.531000
match1$exposedyes -13.08404  -4.287404
```

Next, we'll show a covariate-adjusted model for the impact of exposure on outcome, accounting for each of the variables that we'll wind up using in our propensity score model.

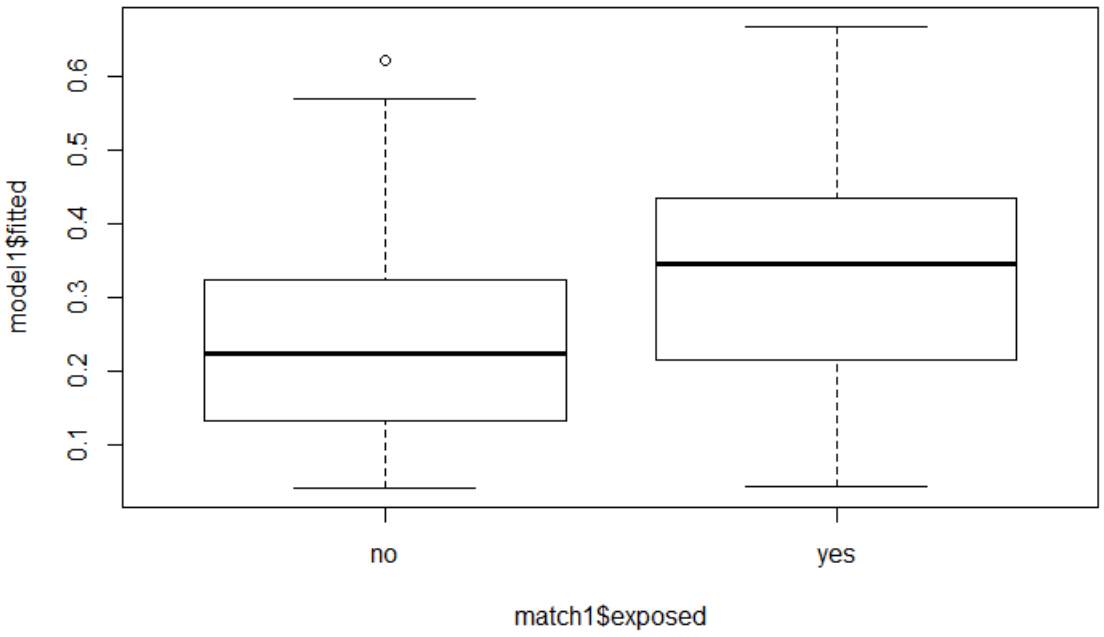
```
> modelxa <- lm(outcome ~ exposed + age + female + factor(race) +
comorbid + serumK + wbc, data=match1)

> summary(modelxa); round(confint(modelxa),2)

Call: lm(formula = outcome ~ exposed + age + female + factor(race) +
    comorbid + serumK + wbc, data = match1)

Residuals:     Min      1Q  Median      3Q     Max
          -32.623 -13.262   0.277  12.864  41.412

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    8.864e+01  1.172e+01   7.565 5.15e-13 ***
exposedyes    -1.015e+01  2.303e+00  -4.405 1.49e-05 ***
age            4.229e-02  7.089e-02   0.597    0.551
female        -1.183e+00  1.979e+00  -0.598    0.551
factor(race)2 -2.175e-01  2.775e+00  -0.078    0.938
factor(race)3 -1.003e+00  2.851e+00  -0.352    0.725
factor(race)4 -2.806e+00  2.871e+00  -0.977    0.329
comorbid       1.058e-01  3.729e-01   0.284    0.777
serumK         9.454e+00  2.333e+00   4.052 6.51e-05 ***
wbc           -6.925e-04  6.567e-04  -1.054    0.293
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 16.71 on 290 degrees of freedom
Multiple R-squared: 0.1031, Adjusted R-squared: 0.07527
F-statistic: 3.704 on 9 and 290 DF,  p-value: 0.0002044

                2.5 % 97.5 %
(Intercept)    65.58 111.70
exposedyes    -14.68  -5.61
age            -0.10   0.18
female         -5.08   2.71
factor(race)2  -5.68   5.24
factor(race)3  -6.61   4.61
factor(race)4  -8.46   2.84
comorbid       -0.63   0.84
serumK          4.86  14.04
wbc             0.00   0.00
```

Estimate	Point Estimate	SE	Lower 95	Upper 95
Unadjusted	-8.69	2.24	-13.08	-4.29
Covariate-adjusted (no PS)	-10.15	2.30	-14.68	-5.61

OK, now we'll fit the propensity score model to describe exposure on the basis of the covariates…

```
> model1 <- glm(exposed ~ age + female + factor(race) + comorbid +
serumK + wbc, family="binomial", data=match1)
> model1

Call: glm(formula = exposed ~ age + female + factor(race) + comorbid +
    serumK + wbc, family = "binomial", data = match1)

Coefficients:
  (Intercept)           age        female  factor(race)2
   -6.609e+00    -1.716e-02    -5.392e-01     4.577e-01
factor(race)3  factor(race)4      comorbid         serumK
    7.750e-01     7.274e-01     1.164e-01     1.040e+00
          wbc
    9.801e-05

Degrees of Freedom: 299 Total (i.e. Null);  291 Residual
Null Deviance:          343.8
Residual Deviance: 315.7   AIC: 333.7
```

```
> plot(model1$fitted ~ match1$exposed)
```




```
> library(Matching)
```

Save key data elements for Matching. Here, we'll match on the linear propensity scores.

```
> X <- model1$linear.predictors
> Tr <- as.logical(match1$exposed=="yes")
> Y <- match1$outcome
> ## 1:1 matching, without replacement
```

```
> try1 <- Match(Y, Tr, X, M=1, replace=FALSE)

> summary(try1)
Estimate...  -11.013
SE.........  2.3992
T-stat.....  -4.5902
p.val......  4.4288e-06

Original number of observations..............  300
Original number of treated obs...............  78
Matched number of observations...............  78
Matched number of observations  (unweighted).  78


> ## Get summary statistics to describe balance checks before and
after matching - including linear propensity scores

> mb1 <- MatchBalance(Tr ~ age + female + factor(race) + comorbid +
serumK + wbc + model1$linear.predictors, data=match1, match.out=try1,
nboots=500)

***** (V1) age *****
                         Before Matching          After Matching
mean treatment........     39.218              39.218
mean control..........     42.311               39.59
std mean diff.........    -21.555             -2.5911

mean raw eQQ diff.....     3.0256              1.3462
med  raw eQQ diff.....          3                   1
max  raw eQQ diff.....          8                   6

mean eCDF diff........   0.064983            0.028149
med  eCDF diff........   0.064276            0.025641
max  eCDF diff........     0.1438            0.089744

var ratio (Tr/Co).....     1.1125              1.0749
T-test p-value........   0.099433             0.87393
KS Bootstrap p-value..      0.122               0.842
KS Naive p-value......    0.18366             0.91194
KS Statistic..........     0.1438            0.089744
```

Let's break in here to look at the details of this particular variable's balance before and after matching.

```
                         Before Matching      After Matching
mean treatment........     39.218              39.218
mean control..........     42.311               39.59
std mean diff.........    -21.555             -2.5911
```

The results start with this table of means in the "treated" (exposed = yes, here) and "control" groups, along with one version of the standardized difference between the treated and control patients, specifically 100 times the mean difference between "treatment" and "control" patients, divided by the standard deviation of the "treatment" patients alone, rather than the pooled version shown in class.

```
                        Before Matching          After Matching
mean raw eQQ diff.....     3.0256              1.3462
med  raw eQQ diff.....          3                   1
max  raw eQQ diff.....          8                   6
```

This part of the table shows summary statistics for the empirical QQ plot – mean, median and maximum raw differences observed in that plot. These are on the scale of the age variable itself.

```
mean eCDF diff........   0.064983           0.028149
med  eCDF diff........   0.064276           0.025641
max  eCDF diff........     0.1438           0.089744
```

Next, the software shows summary statistics for the empirical cumulative distribution function plot – mean, median and maximum standardized differences observed in that plot.

```
var ratio (Tr/Co).....     1.1125            1.0749
T-test p-value........   0.099433           0.87393
KS Bootstrap p-value..      0.122             0.842
KS Naive p-value......    0.18366           0.91194
KS Statistic..........     0.1438           0.089744
```

Finally, we see the ratio of the variances in the treatment and control groups (we want this to be as close to 1 as possible) followed by two univariate tests – the t test and the bootstrap Kolmogorov-Smirnov (KS) test. Before matching, the software uses the independent-samples t test, afterwards, it uses a paired samples comparison. The author of the software, Jasjeet Sekhon, strongly prefers the bootstrap KS approach. The goal should be to improve the balance of each covariate substantially, pushing the p value for the KS as close to 1 as possible. When a variable is not continuous but instead a factor, the KS tests are not shown.

```
***** (V2) female *****
                        Before Matching          After Matching
mean treatment........    0.37179            0.37179
mean control..........    0.52252             0.4359
std mean diff.........    -30.988            -13.179

mean raw eQQ diff.....    0.15385           0.064103
med  raw eQQ diff.....          0                  0
max  raw eQQ diff.....          1                  1

mean eCDF diff........   0.075364           0.032051
med  eCDF diff........   0.075364           0.032051
max  eCDF diff........    0.15073           0.064103

var ratio (Tr/Co).....    0.94404            0.94987
T-test p-value........   0.020912             0.3534
```

```
***** (V3) factor(race)2 *****
                       Before Matching              After Matching
mean treatment........    0.24359                     0.24359
mean control..........    0.27928                     0.15385
std mean diff.........     -8.261                      20.773

mean raw eQQ diff.....   0.038462                    0.089744
med  raw eQQ diff.....          0                           0
max  raw eQQ diff.....          1                           1

mean eCDF diff........   0.017845                    0.044872
med  eCDF diff........   0.017845                    0.044872
max  eCDF diff........    0.03569                    0.089744

var ratio (Tr/Co).....    0.92311                      1.4154
T-test p-value........    0.53566                     0.10652


***** (V4) factor(race)3 *****
                       Before Matching              After Matching
mean treatment........    0.33333                     0.33333
mean control..........    0.22523                     0.35897
std mean diff.........     22.786                     -5.4043

mean raw eQQ diff.....    0.10256                    0.025641
med  raw eQQ diff.....          0                           0
max  raw eQQ diff.....          1                           1

mean eCDF diff........   0.054054                    0.012821
med  eCDF diff........   0.054054                    0.012821
max  eCDF diff........    0.10811                    0.025641

var ratio (Tr/Co).....     1.2842                     0.96571
T-test p-value........    0.07705                     0.67064


***** (V5) factor(race)4 *****
                       Before Matching              After Matching
mean treatment........    0.28205                     0.28205
mean control..........    0.24324                     0.35897
std mean diff.........     8.5686                     -16.984

mean raw eQQ diff.....   0.038462                    0.076923
med  raw eQQ diff.....          0                           0
max  raw eQQ diff.....          1                           1

mean eCDF diff........   0.019404                    0.038462
med  eCDF diff........   0.019404                    0.038462
max  eCDF diff........   0.038808                    0.076923

var ratio (Tr/Co).....     1.1093                        0.88
T-test p-value........    0.51076                     0.23877


***** (V6) comorbid *****
                       Before Matching              After Matching
mean treatment........      5.141                       5.141
mean control..........      4.455                       4.859
std mean diff.........     28.983                      11.915

mean raw eQQ diff.....    0.71795                     0.61538
med  raw eQQ diff.....          1                           1
max  raw eQQ diff.....          2                           2

mean eCDF diff........    0.07131                    0.061538
med  eCDF diff........    0.06878                    0.064103
max  eCDF diff........    0.15662                    0.089744

var ratio (Tr/Co).....     0.7639                     0.67897
T-test p-value........    0.03575                     0.48692
KS Bootstrap p-value..      0.046                       0.708
KS Naive p-value......    0.11781                     0.91194
```

```
KS Statistic.........    0.15662                    0.089744


***** (V7) serumK *****
                        Before Matching             After Matching
mean treatment........     4.6115                      4.6115
mean control.........      4.4329                       4.609
std mean diff.........      46.23                      0.6635

mean raw eQQ diff.....    0.18077                    0.053846
med  raw eQQ diff.....        0.2                         0.1
max  raw eQQ diff.....        0.4                         0.2

mean eCDF diff........    0.11166                    0.033654
med  eCDF diff........   0.091649                    0.019231
max  eCDF diff........    0.24948                    0.089744

var ratio (Tr/Co).....    0.79245                     0.84782
T-test p-value........ 0.00086801                     0.96248
KS Bootstrap p-value.. < 2.22e-16                       0.716
KS Naive p-value......  0.0015157                     0.91194
KS Statistic.........    0.24948                    0.089744


***** (V8) wbc *****
                        Before Matching             After Matching
mean treatment........     7324.7                      7324.7
mean control.........      7122.9                      7130.9
std mean diff.........     13.534                          13

mean raw eQQ diff.....      304.1                       315.13
med  raw eQQ diff.....        250                         230
max  raw eQQ diff.....        820                         830

mean eCDF diff........    0.04967                     0.05286
med  eCDF diff........   0.046604                    0.051282
max  eCDF diff........    0.12439                     0.11538

var ratio (Tr/Co).....    0.96525                      1.1107
T-test p-value........     0.3079                     0.41579
KS Bootstrap p-value..      0.274                       0.624
KS Naive p-value......    0.33359                     0.67676
KS Statistic.........     0.12439                     0.11538


***** (V9) model1$linear.predictors *****
                        Before Matching             After Matching
mean treatment........   -0.79585                    -0.79585
mean control.........     -1.3113                    -0.85653
std mean diff.........     65.903                       7.7588

mean raw eQQ diff.....    0.52784                    0.074026
med  raw eQQ diff.....    0.55819                     0.01097
max  raw eQQ diff.....    0.68572                     0.43502

mean eCDF diff........    0.19627                    0.029663
med  eCDF diff........    0.21864                    0.012821
max  eCDF diff........    0.32328                     0.12821

var ratio (Tr/Co).....     1.2096                      1.2304
T-test p-value........ 1.1099e-06                   0.0033366
KS Bootstrap p-value.. < 2.22e-16                       0.532
KS Naive p-value...... 1.1515e-05                      0.5431
KS Statistic.........    0.32328                     0.12821
```

At the end of the output, the software presents the smallest p value (indicating imbalance) before matching (and indicates the variable) and then this is repeated after matching.

```
Before Matching Minimum p.value: < 2.22e-16
Variable Name(s): serumK model1$linear.predictors  Number(s): 7 9
```

```
After Matching Minimum p.value: 0.0033366
Variable Name(s): model1$linear.predictors  Number(s): 9
```

Here, we started with the means of both serum potassium and the linear propensity score being so far separated that the p value comparing the distributions was essentially zero (when R says 2.2 x 10^{-16}, you should read that as "zero.")

But after matching, our *worst* balanced variable (by the relatively poor measure of a p value) is the linear propensity score. Looking at the output for that more closely, we see that the t test p value is highly significant, but neither of the Kolmogorov-Smirnov tests is problematic. I prefer the bootstrap.

```
***** (V9) model1$linear.predictors *****
                        Before Matching              After Matching
mean treatment........    -0.79585                     -0.79585
mean control..........     -1.3113                     -0.85653
std mean diff.........      65.903                       7.7588

T-test p-value........ 1.1099e-06                    0.0033366
KS Bootstrap p-value.. < 2.22e-16                        0.532
KS Naive p-value...... 1.1515e-05                       0.5431
KS Statistic..........    0.32328                      0.12821
```

In addition, the differences in means are much reduced from what they were – the mean in the linear propensity score group is -0.79585 – it may be easier to think about this in terms of the raw propensity score it represents – to find this, remember that the linear propensity score is the log odds ratio associated with the propensity score. To convert, you can use the equation below to turn the linear PS into the raw propensity score

Propensity score = exp(Linear PS) / (1 + exp(Linear PS))

So, in R, we can calculate the mean propensity score associated with the treatment group, as…

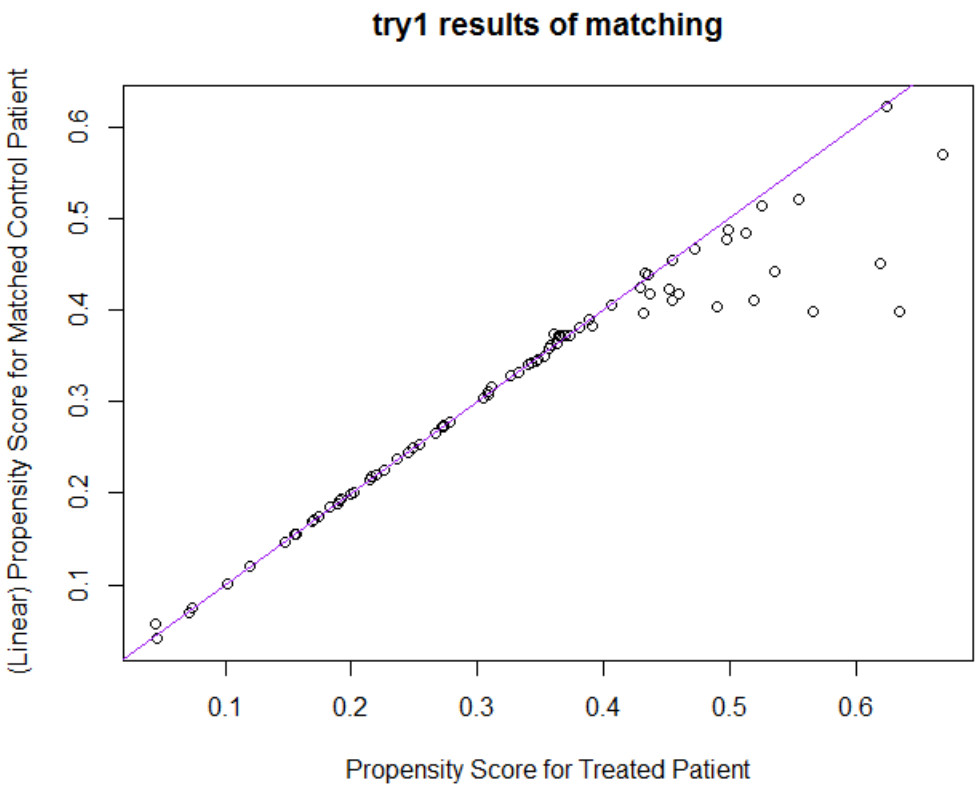
```
> exp(-0.79585)/(1+exp(-0.79585))
[1] 0.3109139
```

And for the control group, the mean propensity score after matching is…

```
> exp(-0.85653)/(1+exp(-0.85653))
[1] 0.2980648
```

So the mean propensity score in the treatment group is 0.31 after matching, as compared to the control group after matching, where the mean propensity score is 0.30. I think we're OK.

Let's get a plot to describe these matches. This isn't a great one, but it's a start.




```
> plot(model1$fitted[try1$index.control] ~
       model1$fitted[try1$index.treated], main="try1 results of
       matching", xlab="Propensity Score for Treated Patient",
       ylab="(Linear) Propensity Score for Matched Control Patient")
> abline(a=0,b=1, lwd=1, col="purple")
```

Note that the purple line here shows perfect matches on the propensity score. We can see that at the higher end of the propensity scores for treated patients, we start needing to match treated patients to controls with (in some cases) meaningfully lower propensity scores.

In the script, I also produce a plot comparing the linear propensity scores in this way. The plot's shape is essentially the same.

Next, we'll use the Matching output to look at the balance of the covariates, both before and after matching.

We'll look specifically at **standardized differences**, and then at **variance ratios**, for each covariate, in each case comparing results after matching to the results before matching in a table, and graphically. The graphs, as always, are the key thing.

To start, we'll get the standardized differences for each covariate, before and after matching, using the "pooled" standard deviation as the denominator, from the Matching output, and put these results into two vectors, called pre.szd (for the pre-matching standardized differences) and post.szd, with appropriate names from the covariates.

```
> covnames <- c("age", "female", "race = 2", "race = 3", "race = 4",
"comorbid", "serumK", "wbc", "linear ps")
> pre.szd <- NULL; post.szd <- NULL
> for(i in 1:length(covnames)) {
+ pre.szd[i] <- mb1$BeforeMatching[[i]]$sdiff.pooled
+ post.szd[i] <- mb1$AfterMatching[[i]]$sdiff.pooled
+ }
```

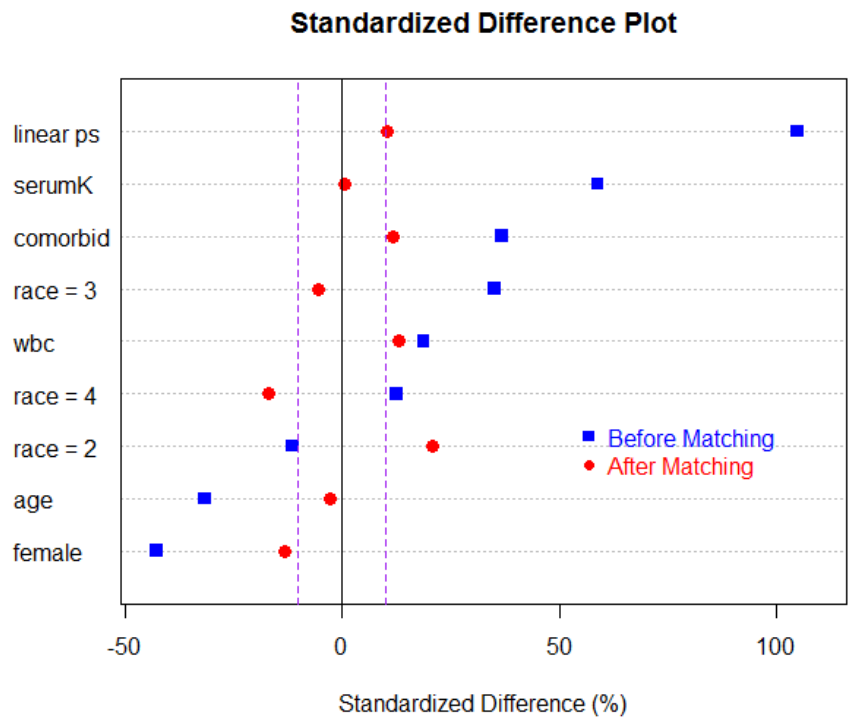
Next, we put the new szd variables and their names into a data frame, so we can build a table.

```
> temp <- data.frame(pre.szd, post.szd, row.names=covnames)
> print(temp, digits=3)
           pre.szd post.szd
age          -31.6   -2.591
female       -42.6  -13.179
race = 2     -11.3   20.773
race = 3      35.1   -5.404
race = 4      12.6  -16.984
comorbid      36.8   11.915
serumK        58.9    0.663
wbc           18.9   13.000
linear ps    104.8   10.446
```

OK, now we build the **standardized difference plot**.

To do this, we define the data frame of standardized differences, with appropriate names, sort by the pre-matching standardized differences, determine appropriate lower and upper limits for the plot, plot the data, add finishing touches, and then click on the plot to place a legend.

```
> temp <- data.frame(pre.szd, post.szd, row.names=covnames)
> tempsort <- temp[with(temp, order(pre.szd)), ]
> low <- min(min(pre.szd), min(post.szd), -0.1)
> high <- max(max(pre.szd), max(post.szd), 0.1)
> dotchart(tempsort$pre.szd, xlim=c(1.05*low, 1.05*high), pch="",
      labels=row.names(tempsort), main="Standardized Difference Plot",
      xlab="Standardized Difference (%)")
>    points(tempsort$pre.szd, seq(1:length(tempsort$pre.szd)), pch=15,
      col="blue", cex=1.2)
>    points(tempsort$post.szd, seq(1:length(tempsort$post.szd)),
      pch=19, col="red", cex=1.2)
>    abline(v=0, lty=1)
>    abline(v=10, lty=2, col="purple")
>    abline(v=-10, lty=2, col="purple")
>    legend(locator(1), legend = c("Before Matching", "After
      Matching"), col=c("blue", "red"), text.col=c("blue", "red"),
      bty="n", pch = c(15, 19))
```

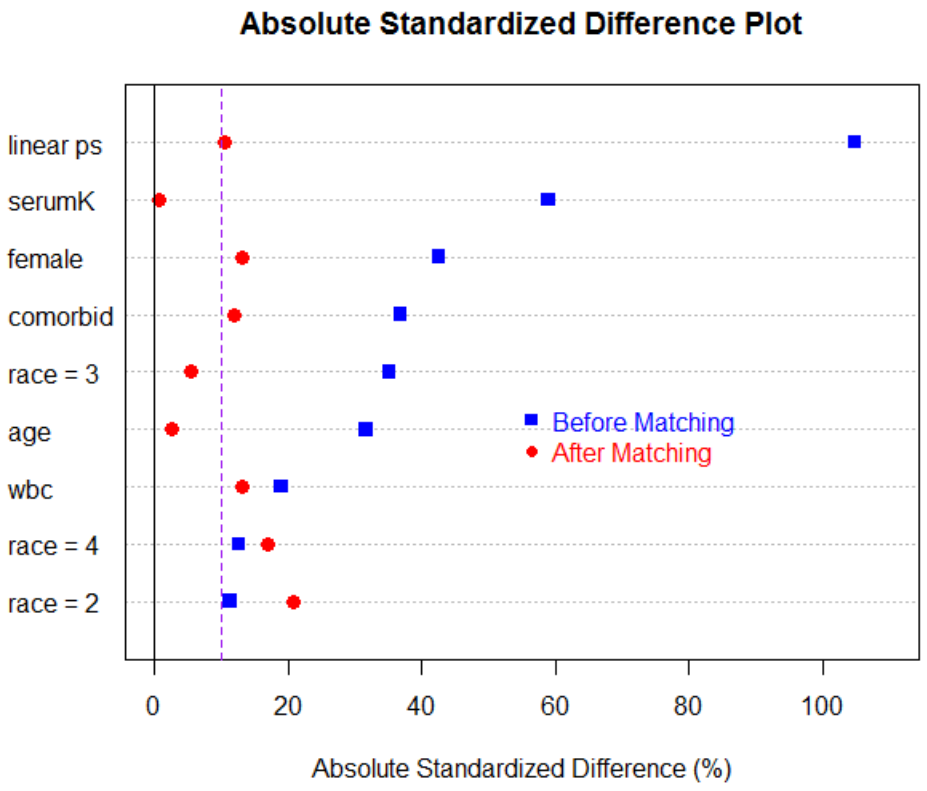


We see general improvement, with some concern about races 2 and 4, and wbc count, and perhaps comorbid illnesses. Still, the big problems (especially the propensity score overall) are much improved.

Next, we can use most of the same commands to build the **absolute standardized difference plot**.

Again, we define the data frame of standardized differences, with appropriate names (not necessary if we've already done it, but I'm assuming you may only want this plot), then sort by the absolute values of the pre-matching standardized differences, determine an appropriate upper limit for the plot (since 0 is always the appropriate lower limit), then plot, add finishing touches, and click to place a legend.

```
> temp <- data.frame(pre.szd, post.szd, row.names=covnames)
> tempsort <- temp[with(temp, order(abs(pre.szd))),]
> high <- max(max(abs(pre.szd)), max(abs(post.szd)), 0.1)
> dotchart(abs(tempsort$pre.szd), pch="", xlim=c(0, 1.05*high),
      labels=row.names(tempsort), main="Absolute Standardized
      Difference Plot", xlab="Absolute Standardized Difference (%)")
>    points(abs(tempsort$pre.szd), seq(1:length(tempsort$pre.szd)),
      pch=15, col="blue", cex=1.2)
>    points(abs(tempsort$post.szd), seq(1:length(tempsort$post.szd)),
      pch=19, col="red", cex=1.2)
>    abline(v=0, lty=1)
>    abline(v=10, lty=2, col="purple")
>    legend(locator(1), legend = c("Before Matching", "After
      Matching"), col=c("blue", "red"), text.col=c("blue", "red"),
      bty="n", pch = c(15, 19))
```



Either the absolute standardized differences or raw standardized differences plot should be done every time you assess the balance of propensity score matching – you don't need both, as you should draw the same conclusions regardless of which plot you draw. I usually use the absolute standardized differences version.

For variance ratios, we will again produce a table and a graph. To start, we'll get the treatment / control variance ratios for each covariate, before and after matching, from the Matching output, and put these results into two vectors, called pre.vratio (for the pre-matching variance ratios) and post.vratio, with appropriate names from the covariates, as we did with the standardized differences.

```
> covnames <- c("age", "female", "race = 2", "race = 3", "race = 4",
"comorbid", "serumK", "wbc", "lin ps")
> pre.vratio <- NULL; post.vratio <- NULL
> for(i in 1:length(covnames)) {
+   pre.vratio[i] <- mb1$BeforeMatching[[i]]$var.ratio
+   post.vratio[i] <- mb1$AfterMatching[[i]]$var.ratio
+ }
```

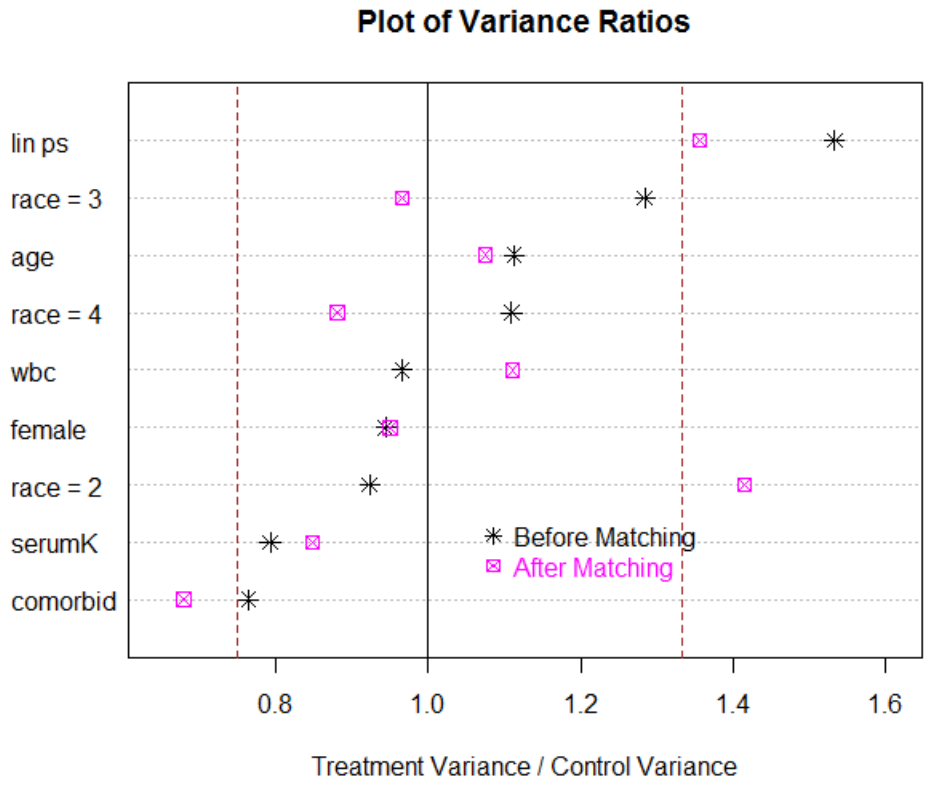
Here's the resulting table:

```
> temp <- data.frame(pre.vratio, post.vratio, row.names=covnames)
> print(temp, digits=2)
         pre.vratio post.vratio
age            1.11        1.07
female         0.94        0.95
race = 2       0.92        1.42
race = 3       1.28        0.97
race = 4       1.11        0.88
comorbid       0.76        0.68
serumK         0.79        0.85
wbc            0.97        1.11
lin ps         1.53        1.36
```

And here's the script for the **variance ratios** plot, most of which just involves small tweaks to what we did for the standardized difference plots, but now with vertical lines drawn at 1, and at 3/4 and 4/3 …

```
> temp <- data.frame(pre.vratio, post.vratio, row.names=covnames)
> tempsort <- temp[with(temp, order(pre.vratio)), ]
> low <- min(min(pre.vratio), min(post.vratio))
> high <- max(max(pre.vratio), max(post.vratio))

> dotchart(tempsort$pre.vratio, xlim=c(0.95*low, 1.05*high), pch="",
      labels=row.names(tempsort), main="Plot of Variance Ratios",
      xlab="Treatment Variance / Control Variance")
> points(tempsort$pre.vratio, seq(1:length(tempsort$pre.vratio)),
      pch=8, col="black", cex=1.2)
> points(tempsort$post.vratio, seq(1:length(tempsort$post.vratio)),
      pch=7, col="magenta", cex=1.2)
> abline(v=1, lty=1)
> abline(v=3/4, lty=2, col="brown")
> abline(v=4/3, lty=2, col="brown")
> legend(locator(1), legend = c("Before Matching", "After Matching"),
      col=c("black", "magenta"), text.col=c("black", "magenta"), pch =
      c(8, 7))
```

The resulting plot is shown at the top of the next page.



You can see I've changed the color scheme and plotting symbols a bit, to try to help us distinguish the two types of plots. Here, we see that most variances are close to 1, both before and after matching, though we might look a bit more closely at race 2 and at comorbid.

What is the difference between the ATT and ATE estimands?

You'll see in the R script that I provide code to obtain both an ATE estimate and an ATT estimate here.

The **average treatment effect on the treated** (ATT) = E[Y(treated) – Y(control) | Z = 1], is the expected gain in outcome due to treatment for the population of people who were actually treated. Most of the time, this is the estimand we focus on when doing propensity score matching where we match a control patient (from a pool of such patients) to each treated patient.

The **average treatment effect** (ATE) = E[Y(treated) – Y(control)], is the expected gain in outcome due to treatment for a randomly selected member of the entire population of interest.

Here, the estimates turn out to be different (the ATT is -11.0 and the ATE is -11.5), and really, in most clinical studies with heterogeneous treatment effects, they will be even more substantially different, as they're answering different questions. Again, we'll stick with ATT for propensity matching, mostly.

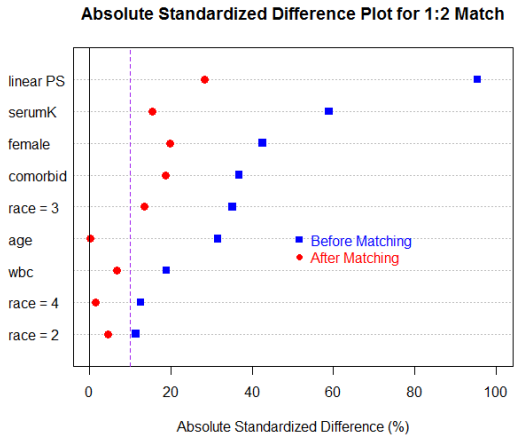
Now, we'll move on to several other approaches for matching, specifically:

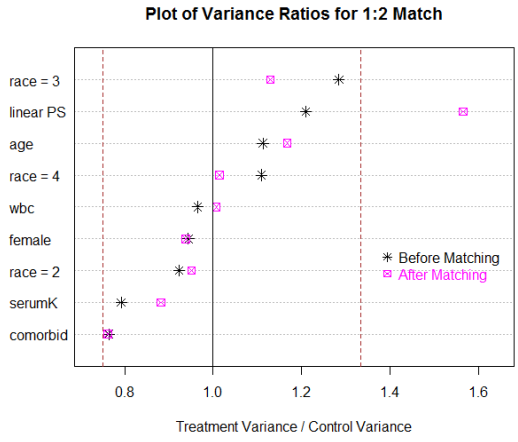
1:2 matching, without replacement

```
try2 <- Match(Y, Tr, X, M=2, replace=FALSE)

summary(try2)
```

```
Estimate...  -10.545
SE.........  2.2626
T-stat.....  -4.6606
p.val......  3.1533e-06

Original number of observations..............  300
Original number of treated obs...............  78
Matched number of observations...............  78
Matched number of observations  (unweighted).  156
```

```
mb2 <- MatchBalance(Tr ~ age + female + factor(race) + comorbid +
serumK + wbc, data=match1, match.out=try2, nboots=500)
```





1:3 matching, without replacement

```
try3 <- Match(Y, Tr, X, M=3, replace=FALSE)

summary(try3)

mb3 <- MatchBalance(Tr ~ age + female + factor(race) + comorbid +
serumK + wbc, data=match1, match.out=try3, nboots=500)
```

I'll skip these 1:3 match results here. You can get a more complete set if you run the script.

1:1 matching, with replacement

```
try4 <- Match(Y, Tr, X, M=1)

summary(try4)

mb4 <- MatchBalance(Tr ~ age + female + factor(race) + comorbid +
serumK + wbc, data=match1, match.out=try4, nboots=500)
```

I'll skip these 1:1 with replacement results here. You can get a more complete set if you run the script.

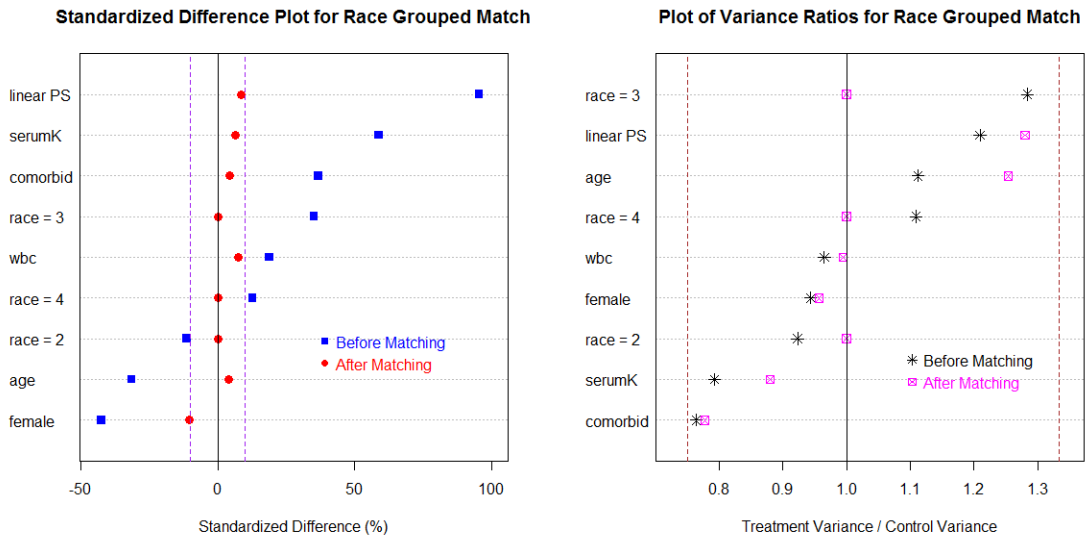
Grouped Propensity Score Matching - separate into subgroups defined by a factor, like race

This leads to an exact match on race, at the possible expense of a poorer match on other variables.

```
try5 <- Matchby(Y, Tr, X, by=match1$race, M=1, replace=FALSE)

summary(try5)
```

```
Estimate...  -11.808
SE.........  2.3292
T-stat.....  -5.0695
p.val......  3.9893e-07

Original number of observations..............  300
Original number of treated obs...............  78
Matched number of observations...............  78
Matched number of observations  (unweighted).  78
```

```
mb5 <- MatchBalance(Tr ~ age + female + factor(race) + comorbid +
serumK + wbc, data=match1, match.out=try5, nboots=500)
```



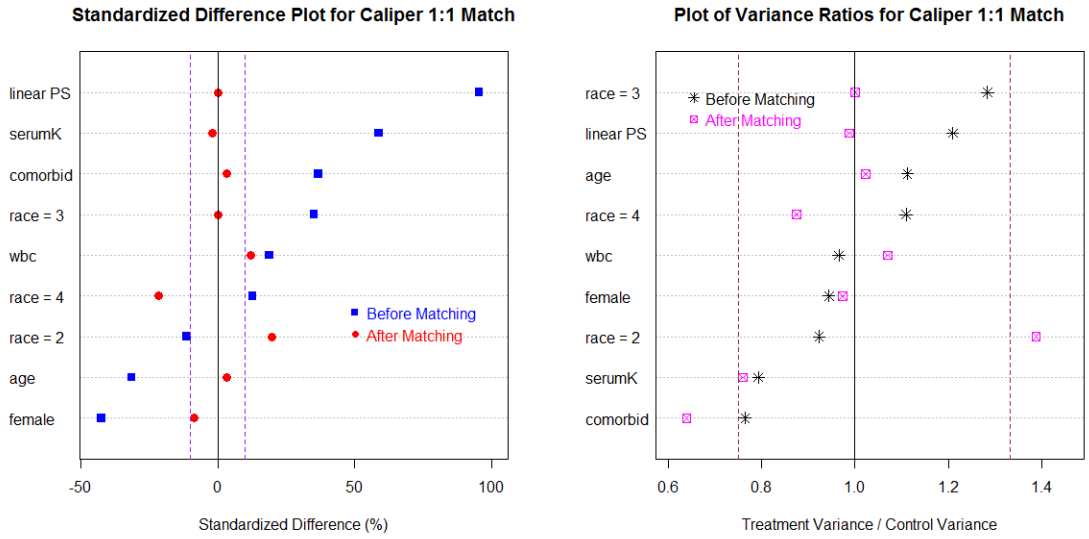
Caliper Propensity Score Matching

Here, we drop all matches not equal to or within 0.25 standard deviations for each of the covariates. We can set separate calipers for each variable, too. This generally changes what we're estimating.

```
try6 <- Match(Y, Tr, X, M=1, replace=FALSE, caliper=0.25)

summary(try6)
```

```
Estimate...  -10.814
SE.........  2.6603
T-stat.....  -4.065
p.val......  4.8024e-05

Original number of observations..............  300
Original number of treated obs...............  78
Matched number of observations...............  70
Matched number of observations  (unweighted).  70

Caliper (SDs)........................................   0.25
Number of obs dropped by 'exact' or 'caliper'  8
```

```
mb6 <- MatchBalance(Tr ~ age + female + factor(race) + comorbid +
serumK + wbc, data=match1, match.out=try6, nboots=500)
```



GenMatch – Genetic Search Matching to do 1:1 Matching Without Replacement

The Matching library also provides a tool to use a genetic search algorithm (and not the propensity score) to make decisions about the most appropriate way to match patients so as to balance the covariates of interest. This is called optimal balance using multivariate matching. Balance is determined by the same set of comparisons as are provided in the Matching library function.

```
> library(rgenoud)
> attach(match1)
```

We separately identify

- the treatment indicator (must be a logical variable),
- the variable(s) we want to match on (in X1 here) – that could just be the propensity score, or alternatively the actual observed covariates, and
- the variables we wish to achieve balance on (in BalanceMat here) – in default, this is the same set as those we want to match on, but it could be a larger or smaller group

```
> Tr <- as.logical(match1$exposed=="yes")
> X1 <- cbind(age,female,factor(race),comorbid,serumK,wbc)
> BalanceMat <- cbind(age,female,factor(race),comorbid,serumK,wbc)
> BalanceMat
      age female   comorbid serumK   wbc
  [1,]  45      0 4        9    4.6  7750
  [2,]  29      1 4        4    4.3  6200
  [3,]  61      1 1        6    4.1  6650
  [4,]  58      1 3        1    4.7  9370
  [5,]  33      1 4        7    4.4  9860
Etc.
```

Now, we'll use the GenMatch command to find the "optimal" matching approach for an ATT estimate using 1:1 matching without replacement, in these data…

```
> genout <- GenMatch(Tr=Tr, X=X1, BalanceMatrix=BalanceMat,
estimand="ATT", M=1, replace=FALSE)
```

Next, we specify our outcome variable:

```
> Y <- match1$outcome
```

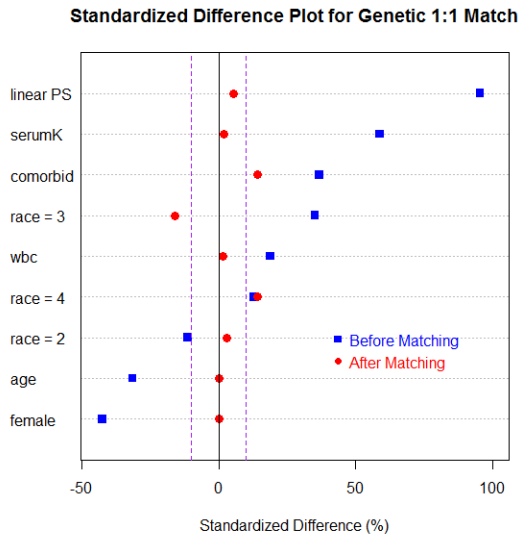
and develop a causal ATT estimate for the effect of our treatment on the outcome.

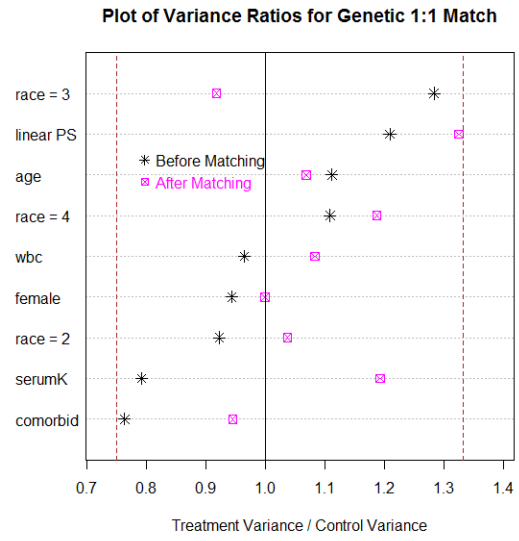
```
> trygen <- Match(Y=Y, Tr=Tr, X=X1, estimand="ATT",
Weight.matrix=genout)
```

```
> summary(trygen)

Estimate...  -11.397
AI SE......  2.7632
T-stat.....  -4.1247
p.val......  3.7127e-05

Original number of observations..............  300
Original number of treated obs...............  78
Matched number of observations...............  78
Matched number of observations  (unweighted).  79
```

To see the assessment of balance on the variables of interest, we can use `MatchBalance` again…

```
> mbgen <- MatchBalance(Tr ~ age + female + factor(race) + comorbid +
serumK + wbc, data=match1, match.out=trygen, nboots=500)

***** (V1) age *****
                    Before Matching        After Matching
mean treatment........     39.218              39.218
mean control.........      42.311              39.218
std mean diff.........    -21.555                   0

Etc.

Before Matching Minimum p.value: < 2.22e-16
Variable Name(s): serumK  Number(s): 7

After Matching Minimum p.value: 0.16429
Variable Name(s): factor(race)4  Number(s): 5
```





Comparison of Balance Achieved (something you might actually do)

Try	Approach	Standardized Differences After Matching (Low, High)	Variance Ratios After Matching (Low, High)	Smallest P Value After Matching
1	ATT 1:1 PS matching, without replacement	(-17, 21)	(0.68, 1.42)	0.003 (linear PS)
2	ATT 1:2 PS matching, without replacement	(-20, 28)	(0.76, 1.57)	< 0.001 (linear PS)
3	ATT 1:3 PS matching, without replacement	(-30, 61)	(0.78, 1.28)	< 0.001 (serum K and linear PS)
4	ATT 1:1 PS matching, with replacement	(-14, 36)	(0.63, 1.30)	0.035 (linear PS)
5	ATT 1:1 PS matching, without replacement with exact matching on race	(-13, 10)	(0.77, 1.30)	0.003 (linear PS)
6	ATT 1:1 PS matching, without replacement with a caliper of 0.25 SD	(-19, 17)	(0.63, 1.27)	0.182 (comorbid)
gen	ATT Optimal 1:1 matching based on genetic search algorithm weights	(-19, 17)	(0.91, 1.24)	0.056 (race 4)

Which of these seven matching strategies appear most worthy of additional consideration in this case?

Comparison of Outcome Estimates (this complete table is something you'd rarely, if ever, do – instead, you'd pick a good matching approach based on the information you obtain on covariate balance, then look just at that one)

Approach	Point Estimate	SE	Lower 95% CI	Upper 95% CI
Unadjusted	-8.69	2.24	-13.08	-4.29
Covariate adjusted (no PS)	-10.15	2.30	-14.68	-5.61
ATT 1:1 PS matching, without replacement	-11.01	2.40	-15.81	-6.21
ATE 1:1 PS matching, without replacement	-11.52	1.60	-14.72	-8.32
ATT 1:2 PS matching, without replacement	-10.55	2.26	-15.07	-6.03
ATT 1:3 PS matching, without replacement	-8.32	2.45	-13.22	-3.42
ATT 1:1 PS matching, with replacement	-12.82	3.14	-19.10	-6.54
ATT 1:1 PS matching, without replacement with exact matching on race	-11.60	2.34	-16.28	-6.92
ATT 1:1 PS matching, without replacement with a caliper of 0.25 SD	-10.74	2.70	-16.14	-5.34
ATT Optimal 1:1 matching based on genetic search algorithm weights	-11.40	2.76	-16.92	-5.88