Journal of Statistical Software

June 2011, Volume 42, Issue 7. *http://www.jstatsoft.org/*

Multivariate and Propensity Score Matching Software with Automated Balance Optimization: The Matching Package for R

Jasjeet S. Sekhon
UC Berkeley

Abstract

Matching is an R package which provides functions for multivariate and propensity score matching and for finding optimal covariate balance based on a genetic search algorithm. A variety of univariate and multivariate metrics to determine if balance actually has been obtained are provided. The underlying matching algorithm is written in C++, makes extensive use of system BLAS and scales efficiently with dataset size. The genetic algorithm which finds optimal balance is parallelized and can make use of multiple CPUs or a cluster of computers. A large number of options are provided which control exactly how the matching is conducted and how balance is evaluated.

Keywords: propensity score matching, multivariate matching, genetic optimization, causal inference, R.

1. Introduction

The R (R Development Core Team 2011) package **Matching** implements a variety of algorithms for multivariate matching including propensity score, Mahalanobis, inverse variance and genetic matching (GenMatch). The last of these, genetic matching, is a method which automatically finds the set of matches which minimize the discrepancy between the distribution of potential confounders in the treated and control groups—i.e., covariate balance is maximized. The package enables a wide variety of matching options including matching with or without replacement, bias adjustment, different methods for handling ties, exact and caliper matching, and a method for the user to fine tune the matches via a general restriction matrix. Variance estimators include the usual Neyman standard errors (which condition on the matched data), Abadie and Imbens (2006) standard errors which account for the (asymp-

totic) variance induced by the matching procedure itself, and robust variances which do not assume a homogeneous causal effect. The **Matching** software package is available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/package=Matching`.

The package provides a set of functions to do the matching (`Match`) and to evaluate how good covariate balance is before and after matching (`MatchBalance`). The `GenMatch` function finds optimal balance using multivariate matching where a genetic search algorithm determines the weight each covariate is given. Balance is determined by examining cumulative probability distribution functions of a variety of standardized statistics. By default, these statistics include paired t tests, univariate and multivariate Kolmogorov-Smirnov (KS) tests. A variety of descriptive statistics based on empirical-QQ plots are also offered. The statistics are not used to conduct formal hypothesis tests, because no measure of balance is a monotonic function of bias in the estimand of interest and because we wish to maximize balance without limit. `GenMatch` can maximize balance based on a variety of pre-defined loss functions or any loss function the user may wish to provide.

In the next section I briefly offer some background material on both the Rubin causal model and matching methods. Section 3 provides an overview of the **Matching** package with examples. Section 4 concludes.

2. Background on matching

Matching has become an increasingly popular method of causal inference in many fields including statistics (Rubin 2006; Rosenbaum 2002), medicine (Christakis and Iwashyna 2003; Rubin 1997), economics (Abadie and Imbens 2006; Dehejia and Wahba 2002, 1999), political science (Bowers and Hansen 2005; Herron and Wand 2007; Imai 2005), sociology (Morgan and Harding 2006; Diprete and Engelhardt 2004; Winship and Morgan 1999; Smith 1997) and even law (Rubin 2001). There is, however, no consensus on how exactly matching ought to be done and how to measure the success of the matching procedure. A wide variety of matching procedures have been proposed, and **Matching** implements many of them.

When using matching methods to estimate causal effects, a central problem is deciding how best to perform the matching. Two common approaches are propensity score matching (Rosenbaum and Rubin 1983) and multivariate matching based on Mahalanobis distance (Cochran and Rubin 1973; Rubin 1979, 1980). Matching methods based on the propensity score (estimated by logistic regression), Mahalanobis distance or a combination of the two have appealing theoretical properties if covariates have ellipsoidal distributions—e.g., distributions such as the normal or t. If the covariates are so distributed, these methods (more generally affinely invariant matching methods[1]) have the property of "equal percent bias reduction" (EPBR) (Rubin 1976a,b; Rubin and Thomas 1992).[2] This property is formally defined in Appendix A. When this property holds, matching will reduce bias in all linear combinations of the covariates. If the EPBR property does not hold, then, in general, matching will increase the bias of some linear functions of the covariates even if all univariate means are closer in the matched data than the unmatched (Rubin 1976a). Unfortunately, the EPBR property

[1] Affine invariance means that the matching output is invariant to matching on X or an affine transformation of X.

[2] The EPBR results of Rubin and Thomas (1992) have been extended by Rubin and Stuart (2006) to the case of discriminant mixtures of proportional ellipsoidally symmetric (DMPES) distributions. This extension is important, but it is restricted to a limited set of mixtures.

rarely holds with actual data.

A significant shortcoming of common matching methods such as Mahalanobis distance and propensity score matching is that they may (and in practice, frequently do) make balance worse across measured potential confounders. These methods may make balance worse, in practice, even if covariates are distributed ellipsoidally because in a given finite sample there may be departures from an ellipsoidal distribution. Moreover, if covariates are neither ellipsoidally symmetric nor are mixtures of discriminant mixtures of proportional ellipsoidally symmetric (DMPES) distributions, propensity score matching has good theoretical properties if and only if the true propensity score model is known and the sample size is large.

These limitations often surprise applied researchers. Because of the limited theoretical properties for matching when the propensity score is not known, one approach is to algorithmically impose additional properties, and this is the approach used by genetic matching.

Diamond and Sekhon (2005) and Sekhon and Grieve (2011) propose a matching algorithm, genetic matching (GenMatch), that maximizes the balance of observed covariates between treated and control groups. GenMatch is a generalization of propensity score and Mahalanobis distance matching, and it has been used by a variety of researchers (e.g., Andam, Ferraro, Pfaff, Sanchez-Azofeifa, and Robalino 2008; Eggers and Hainmueller 2009; Gilligan and Sergenti 2008; Gordon 2009; Heinrich 2007; Hopkins 2010; Morgan and Harding 2006; Lenz and Ladd 2009; Raessler and Rubin 2005). The algorithm uses a genetic algorithm (Mebane, Jr. and Sekhon 2011; Sekhon and Mebane 1998) to optimize balance as much as possible given the data. The method is nonparametric and does not depend on knowing or estimating the propensity score, but the method is improved when a propensity score is incorporated.

The core motivation for all matching methods is the Rubin causal model which I discuss next followed by details on Mahalanobis, propensity score and genetic matching.

2.1. Rubin causal model

The Rubin causal model conceptualizes causal inference in terms of potential outcomes under treatment and control, only one of which is observed for each unit (Holland 1986; Splawa-Neyman 1923; Rubin 1974, 1978, 1990). A causal effect is defined as the difference between an observed outcome and its counterfactual.

Let Y_{i1} denote the potential outcome for unit i if the unit receives treatment, and let Y_{i0} denote the potential outcome for unit i in the control regime. The treatment effect for observation i is defined by $\tau_i = Y_{i1} - Y_{i0}$. Causal inference is a missing data problem because Y_{i1} and Y_{i0} are never both observed. Let T_i be a treatment indicator equal to 1 when i is in the treatment regime and 0 otherwise. The observed outcome for observation i is then $Y_i = T_i Y_{i1} + (1 - T_i) Y_{i0}$.

In principle, if assignment to treatment is randomized, causal inference is straightforward because the two groups are drawn from the same population by construction, and treatment assignment is independent of all baseline variables. As the sample size grows, observed and unobserved baseline variables are balanced across treatment and control groups with arbitrarily high probability, because treatment assignment is independent of Y_0 and Y_1—i.e., following the notation of Dawid (1979), $\{Y_{i0}, Y_{i1} \perp\!\!\!\perp T_i\}$. Hence, for $j = 0, 1$

$$E(Y_{ij} \mid T_i = 1) = E(Y_{ij} \mid T_i = 0) = E(Y_i \mid T_i = j)$$

Therefore, the average treatment effect (ATE) can be estimated by:

$$\begin{aligned} \tau &= E(Y_{i1} \mid T_i = 1) - E(Y_{i0} \mid T_i = 0) \\ &= E(Y_i \mid T_i = 1) - E(Y_i \mid T_i = 0) \end{aligned} \tag{1}$$

Equation 1 is estimable in an experimental setting because observations in treatment and control groups are exchangeable.[3] In the simplest experimental setup, individuals in both groups are equally likely to receive the treatment, and hence assignment to treatment will not be associated with the outcome. Even in an experimental setup, much can go wrong which requires statistical correction (e.g., Barnard, Frangakis, Hill, and Rubin 2003).

In an observational setting, covariates are almost never balanced across treatment and control groups because the two groups are not ordinarily drawn from the same population. Thus, a common quantity of interest is the average treatment effect for the treated (ATT):

$$\tau \mid (T = 1) = E(Y_{i1} \mid T_i = 1) - E(Y_{i0} \mid T_i = 1). \tag{2}$$

Equation 2 cannot be directly estimated because Y_{i0} is not observed for the treated. Progress can be made by assuming that selection into treatment depends on observable covariates X. Following Rosenbaum and Rubin (1983), one can assume that conditional on X, treatment assignment is unconfounded ($\{Y_0, Y_1 \perp\!\!\!\perp T\} \mid X$) and that there is overlap: $0 < Pr(T = 1 \mid X) < 1$. Together, unconfoundedness and overlap constitute a property known as strong ignorability of treatment assignment which is necessary for identifying the average treatment effect. Heckman, Ichimura, Smith, and Todd (1998) show that for ATT, the unconfoundedness assumption can be weakened to mean independence: $E\left(Y_{ij} \mid T_i, X_i\right) = E\left(Y_{ij} \mid X_i\right)$.[4] The overlap assumption for ATT only requires that the support of X for the treated be a subset of the support of X for control observations.

Then, following Rubin (1974, 1977) we obtain

$$E(Y_{ij} \mid X_i, T_i = 1) = E(Y_{ij} \mid X_i, T_i = 0) = E(Y_i \mid X_i, T_i = j). \tag{3}$$

By conditioning on observed covariates, X_i, treatment and control groups are exchangeable. The average treatment effect for the treated is estimated as

$$\tau \mid (T = 1) = E\left\{E(Y_i \mid X_i, T_i = 1) - E(Y_i \mid X_i, T_i = 0) \mid T_i = 1\right\}, \tag{4}$$

where the outer expectation is taken over the distribution of $X_i \mid (T_i = 1)$ which is the distribution of baseline variables in the treated group.

The most straightforward and nonparametric way to condition on X is to exactly match on the covariates. This is an old approach going back to at least Fechner (1966), the father of psychophysics. This approach fails in finite samples if the dimensionality of X is large or if X contains continuous covariates. Thus, in general, alternative methods must be used.

[3] It is standard practice to assume the Stable Unit Treatment Value assumption, also known as SUTVA (Holland 1986; Rubin 1978). SUTVA requires that the treatment status of any unit be independent of potential outcomes for all other units, and that treatment is defined identically for all units.

[4] Also see Abadie and Imbens (2006).

2.2. Mahalanobis and propensity score matching

The most common method of multivariate matching is based on Mahalanobis distance (Cochran and Rubin 1973; Rubin 1979, 1980). The Mahalanobis distance between any two column vectors is:

$$md(X_i, X_j) \quad = \left\{(X_i - X_j)^\top S^{-1}(X_i - X_j)\right\}^{\frac{1}{2}}$$

where S is the sample covariance matrix of X. To estimate ATT by matching with replacement, one matches each treated unit with the M closest control units, as defined by this distance measure, $md()$.[5] If X consists of more than one continuous variable, multivariate matching estimates contain a bias term which does not asymptotically go to zero at rate $\sqrt{n}$ (Abadie and Imbens 2006).

An alternative way to condition on X is to match on the probability of assignment to treatment, known as the propensity score.[6] As one's sample size grows large, matching on the propensity score produces balance on the vector of covariates X (Rosenbaum and Rubin 1983).

Let $e(X_i) \equiv Pr(T_i = 1 \mid X_i) = E(T_i \mid X_i)$, defining $e(X_i)$ to be the propensity score. Given $0 < Pr(T_i \mid X_i) < 1$ and $Pr(T_1, T_2, \cdots T_N \mid X_1, X_2, \cdots X_N) = \Pi_{i=1}^{N} e(X_i)^{T_i}(1 - e(X_i))^{(1-T_i)}$, Rosenbaum and Rubin (1983) prove that

$$\tau \mid (T = 1) \quad = \quad E\left\{E(Y_i \mid e(X_i), T_i = 1) - E(Y_i \mid e(X_i), T_i = 0) \mid T_i = 1\right\},$$

where the outer expectation is taken over the distribution of $e(X_i) \mid (T_i = 1)$. Since the propensity score is generally unknown, it must be estimated.

Propensity score matching involves matching each treated unit to the nearest control unit on the unidimensional metric of the propensity score vector. If the propensity score is estimated by logistic regression, as is typically the case, much is to be gained by matching not on the predicted probabilities (bounded between zero and one) but on the linear predictor $\hat{\mu} \equiv X\hat{\beta}$. Matching on the linear predictor avoids compression of propensity scores near zero and one. Moreover, the linear predictor is often more nearly normally distributed which is of some importance given the EPBR results if the propensity score is matched on along with other covariates.

Mahalanobis distance and propensity score matching can be combined in various ways (Rubin 2001; Rosenbaum and Rubin 1985). It is useful to combine the propensity score with Mahalanobis distance matching because propensity score matching is particularly good at minimizing the discrepancy along the propensity score and Mahalanobis distance is particularly good at minimizing the distance between individual coordinates of X (orthogonal to the propensity score) (Rosenbaum and Rubin 1985).

[5] Alternatively one can do optimal full matching (Hansen 2004; Hansen and Klopfer 2006; Rosenbaum 1989, 1991) instead of the 1-to-N matching with replacement which I focus on in this article. This decision is a separate one from the choice of a distance metric.

[6] The first estimator of treatment effects to be based on a weighted function of the probability of treatment was the Horvitz-Thompson statistic (Horvitz and Thompson 1952).

2.3. Genetic matching

The idea underlying the GenMatch algorithm is that if Mahalanobis distance is not optimal for achieving balance in a given dataset, one should be able to search over the space of distance metrics and find something better. One way of generalizing the Mahalanobis metric is to include an additional weight matrix:

$$d(X_i, X_j) \quad = \left\{ (X_i - X_j)^\top \left(S^{-1/2}\right)^\top W S^{-1/2} (X_i - X_j) \right\}^{\frac{1}{2}}$$

where W is a $k \times k$ positive definite weight matrix and $S^{1/2}$ is the Cholesky decomposition of S which is the variance-covariance matrix of X.[7]

Note that if one has a good propensity score model, one should include it as one of the covariates in `GenMatch`. If this is done, both propensity score matching and Mahalanobis matching can be considered special limiting cases of `GenMatch`. If the propensity score contains all of the relevant information in a given sample, the other variables will be given zero weight.[8] And `GenMatch` will converge to Mahalanobis distance if that proves to be the appropriate distance measure.

`GenMatch` is an affinely invariant matching algorithm that uses the distance measure $d()$, in which all elements of W are zero except down the main diagonal. The main diagonal consists of k parameters which must be chosen. Note that if each of these k parameters are set equal to 1, $d()$ is the same as Mahalanobis distance.

The choice of setting the non-diagonal elements of W to zero is made for reasons of computational power alone. The optimization problem grows exponentially with the number of free parameters. It is important that the problem be parameterized so as to limit the number of parameters which must be estimated.

This leaves the problem of how to choose the free elements of W. Many loss criteria recommend themselves, and `GenMatch` provides a number the user can choose from via the `fit.func` and `loss` options of `GenMatch`. By default, cumulative probability distribution functions of a variety of standardized statistics are used as balance metrics and are optimized without limit. The default standardized statistics are paired t tests and nonparametric KS tests.

The statistics are not used to conduct formal hypothesis tests, because no measure of balance is a monotonic function of bias in the estimand of interest and because we wish to maximize balance without limit. Descriptive measures of discrepancy generally ignore key information related to bias which is captured by probability distribution functions of standardized test statistics. For example, using several descriptive metrics, one is unable to recover reliably the experimental benchmark in a testbed dataset for matching estimators (Dehejia and Wahba 1999). And these metrics, unlike those based on optimized distribution functions, perform poorly in a series of Monte Carlo sampling experiments just as one would expect given their properties. For details see Sekhon (2006a).

By default, `GenMatch` attempts to minimize a measure of the maximum observed discrepancy between the matched treated and control covariates at every iteration of optimization. For a given set of matches resulting from a given W, the loss is defined as the minimum p value

[7] The Cholesky decomposition is parameterized such that $S = LL^\top$, $S^{1/2} = L$. In other words, L is a lower triangular matrix with positive diagonal elements.

[8] Technically, the other variables will be given weights just large enough to ensure that the weight matrix is positive definite.

observed across a series of standardized statistics. The user may specify exactly what tests are done via the `BalanceMatrix` option. Examples are offered in Section 3.

Conceptually, the algorithm attempts to minimize the largest observed covariate discrepancy at every step. This is accomplished by maximizing the smallest p value at each step.[9] Because `GenMatch` is minimizing the maximum discrepancy observed at each step, it is minimizing the infinity norm. This property holds even when, because of the distribution of X, the EPBR property does not hold. Therefore, if an analyst is concerned that matching may increase the bias in some linear combination of X even if the means are reduced, `GenMatch` allows the analyst to put in the loss function all of the linear combinations of X which may be of concern. Indeed, any nonlinear function of X can also be included in the loss function, which would ensure that bias in some nonlinear functions of X is not made inordinately large by matching.

The default `GenMatch` loss function does allow for imbalance in functions of X to worsen as long as the maximum discrepancy is reduced. Hence, it is important that the maximum discrepancy be small—i.e., that the smallest p value be large. p values conventionally understood to signal balance (e.g., 0.10), may be too low to produce reliable estimates. After GenMatch optimization, the p values from these balance tests cannot be interpreted as true probabilities because of standard pre-test problems, but they remain useful measures of balance. Also, we are interested in maximizing the balance in the current sample so a hypothesis test for balance is inappropriate.

The optimization problem described above is difficult and irregular, and the genetic algorithm implemented in the **rgenoud** package (Mebane, Jr. and Sekhon 2011) is used to conduct the optimization. Details of the algorithm are provided in Sekhon and Mebane (1998).

`GenMatch` is shown to have better properties than the usual alternative matching methods both when the EPBR property holds and when it does not (Sekhon 2006a; Diamond and Sekhon 2005). Even when the EPBR property holds and the mapping from X to Y is linear, GenMatch has better efficiency—i.e., lower mean square error (MSE)—in finite samples. When the EPBR property does not hold as it generally does not, GenMatch retains appealing properties and the differences in performance between GenMatch and the other matching methods can become substantial both in terms of bias and MSE reduction. In short, at the expense of computer time, GenMatch dominates the other matching methods in terms of MSE when assumptions required for EPBR hold and, even more so, when they do not.

GenMatch is able to retain good properties even when EPBR does not hold because a set of constraints is imposed by the loss function optimized by the genetic algorithm. The loss function depends on a large number of functions of covariate imbalance across matched treatment and control groups. Given these measures, `GenMatch` will optimize covariate balance.

3. Package overview and examples

The three main functions in the package are `Match`, `MatchBalance` and `GenMatch`. The first function, `Match`, performs multivariate and propensity score matching. It is intended to be

[9] More precisely lexical optimization will be done: all of the balance statistics will be sorted from the most discrepant to the least and weights will be picked which minimize the maximum discrepancy. If multiple sets of weights result in the same maximum discrepancy, then the second largest discrepancy is examined to choose the best weights. The processes continues iteratively until ties are broken.

used in conjunction with the `MatchBalance` function which checks if the results of `Match` have actually achieved balance on a set of covariates. `MatchBalance` can also be used before any matching to determine how balanced the raw data is. If one wants to do propensity score matching, one should estimate the propensity model before calling `Match`, and then send `Match` the propensity score to use. The `GenMatch` function can be used to *automatically find balance* by the use of a genetic search algorithm which determines the optimal weight to give each covariate.

Next, I present a set of propensity score (pscore) models which perform better as adjustments are made to them after the output of `MatchBalance` is examined. I then provide an example using `GenMatch`.

3.1. Propensity score matching example

In order to do propensity score matching, the work flow is to first estimate a propensity score using, for example, `glm` if one wants to estimate a propensity score using logistic regression. A number of alternative methods of estimating the propensity score, such as General Additive Models (GAMs), are possible. After the propensity score has been estimated, one calls `Match` to perform the matching and `MatchBalance` to examine how well the matching procedure did in producing balance. If the balance results printed by `MatchBalance` are not good enough, one would go back and change either the propensity score model or some parameter of how the matching is done—e.g., change from 1-to-3 matching to 1-to-1 matching.

The following example is adopted from the documentation of the `Match` function. The example uses the LaLonde (1986) experimental data which is based on a nationwide job training experiment. The observations are individuals, and the outcome of interest is real earnings in 1978. There are eight baseline variables age (`age`), years of education (`educ`), real earnings in 1974 (`re74`), real earnings in 1975 (`re75`), and a series of indicator variables. The indicator variables are black (`black`), Hispanic (`hisp`), married (`married`) and lack of a high school diploma (`nodegr`).

```
R> library("Matching")
R> data("lalonde")
R> attach(lalonde)
```

Save the outcome of interest in `Y` and the treatment indicator in `Tr`:

```
R> Y <- lalonde$re78
R> Tr <- lalonde$treat
```

We now estimate our first propensity score model:

```
R> glm1 <- glm(Tr ~ age + educ + black + hisp + married + nodegr +
+    re74 + re75, family = binomial, data = lalonde)
```

Let us do one-to-one matching with replacement using our preliminary propensity score model where the estimand is the average treatment effect on the treated (ATT):

```
R> rr1 <- Match(Y = Y, Tr = Tr, X = glm1$fitted)
```

None of the forgoing commands produce output. If we wanted to see the results from the call to `Match` which would display the estimate and its standard error we could do `summary(rr1)`, but it is best to wait until we have achieved satisfactory balance before looking at the estimates. To this end, `Match` does not even need to be provided with an outcome variable—i.e., `Y`—in order to work. Matches can be found and balance evaluated without knowledge of `Y`. Indeed, this is to be preferred so that the design stage of the observational study can be clearly separated from the estimation stage as is the case with experiments.

In the example above, the call to `glm` estimates a simple propensity score model and the syntax of this procedure is covered in the R documentation. Then a call to `Match` is made which relies heavily on the function's default behavior because only three options are explicitly provided: a vector (`Y`) containing the outcome variable, a vector (`Tr`) containing the treatment status of each observation—i.e., either a zero or one—and a matrix (`X`) containing the variables to be matched on, which in this case is simply the propensity score. By default `Match` does 1-to-1 matching with replacement and estimates ATT. The estimand is chosen via the `estimand` option, as in `estimand="ATE"` to estimate the average treatment effect. The ratio of treated to control observations is determined by the the `M` option and this ratio is by default set to 1. And whether matching should be done with replacement is controlled by the logical argument `replace` which defaults to `TRUE` for matching with replacement.

Ties are by default handled deterministically (Abadie and Imbens 2006) and this behavior is controlled by the `ties` option. By default `ties = TRUE`. If, for example, one treated observation matches more than one control observation, the matched dataset will include the multiple matched control observations and the matched data will be weighted to reflect the multiple matches. The sum of the weighted observations will still equal the original number of observations. If `ties = FALSE`, ties will be randomly broken. This in general is not a good idea because the variance of `Y` will be underestimated. But if the dataset is large and there are many ties between potential matches, setting `ties = FALSE` often results in significantly faster execution with negligible bias. Whether two potential matches are close enough to be considered tied, is controlled by the `distance.tolerance` option.

With these defaults, the command

```
R> m1 = Match(Y = Y, Tr = Tr, X = glm1$fitted)
```

is equivalent to

```
R> m1 = Match(Y = Y, Tr = Tr, X = glm1$fitted, estimand = "ATT",
+    M = 1, ties = TRUE, replace = TRUE)
```

We generally want to measure balance for more functions of the data than we include in our propensity score model. We can do this using the following call to the `MatchBalance` function. Note that the function is asked to measure balance for many more functions of the confounders than we included in the propensity score model.

```
R> MatchBalance(Tr ~ age + I(age^2) + educ + I(educ^2) + black + hisp +
+    married + nodegr + re74 + I(re74^2) + re75 + I(re75^2) + u74 + u75 +
+    I(re74 * re75) + I(age * nodegr) + I(educ * re74) + I(educ * re75),
+    match.out = rr1, nboots = 1000, data = lalonde)
```

The full output for this call to `MatchBalance` is presented in Appendix B. The formula used in the call to `MatchBalance` does *not* estimate any model. The formula is simply an efficient way to use the R modeling language to list the variables we wish to obtain univariate balance statistics on. The dependent variable in the formula is the treatment indicator.

The propensity score model is different from the balance statistics which are requested from `MatchBalance`. In general, one does **not** need to include all of the functions one wants to test balance on in the propensity score model. Indeed, doing so sometimes results in *worse* balance. Generally, one should request balance statistics on more higher-order terms and interactions than were included in the propensity score used to conduct the matching itself.

Aside from the formula, three additional arguments were given to the `MatchBalance` call. The `match.out` option is used to provide the output object from the previous call to `Match`. If this object is provided, `MatchBalance` will provide balance statistics for both before and after matching, otherwise balance statistics will only be provided for the unmatched raw dataset. The `nboots` option determines the number of bootstrap samples to be run. If zero, no bootstraps are done. Bootstrapping is highly recommended because the bootstrapped Kolmogorov-Smirnov test, unlike the standard test, provides correct coverage even when there are point masses in the distributions being compared (Abadie 2002). At least 500 `nboots` (preferably 1000) are recommended for publication quality p values. And finally, the `data` argument expects a data frame which contains all of the variables in the formula. If a data frame is not provided, the variables are obtained via lexical scoping.

For each term included into the modeling equation provided as the first argument to `MatchBalance`, detailed balance statistics are produced. Let's first consider the output for the `nodegr` variable. One could examine the long output from the call to `MatchBalance` above where `nodegr` is labeled as 'V8' because it was the eighth variable listed in the formula provided to `MatchBalance`. Alternatively, we could call `MatchBalance` with just `nodegr`:

```
R> MatchBalance(Tr ~ nodegr, match.out = rr1, nboots = 1000, data = lalonde)

***** (V1) nodegr *****
                       Before Matching        After Matching
mean treatment........     0.70811              0.70811
mean control..........     0.83462              0.76757
std mean diff.........     -27.751              -13.043

mean raw eQQ diff.....     0.12432             0.043605
med  raw eQQ diff.....           0                    0
max  raw eQQ diff.....           1                    1

mean eCDF diff........    0.063254             0.021802
med  eCDF diff........    0.063254             0.021802
max  eCDF diff........     0.12651             0.043605

var ratio (Tr/Co).....      1.4998               1.1585
T-test p-value........   0.0020368            0.0071385
```

There are two columns for each variable in the `MatchBalance` output. The first column containing the pre-matching balance statistics and the second one the post-matching statistics.

`nodegr` is an indicator variable for whether the individual in the worker training program has a high school diploma. For such variables, the Kolmogorov-Smirnov test results are not presented because they are the equivalent to the results from t tests.

Four different sets of balance statistics are provided for each variable. The first set consists of the means for the treatment and control groups. The second set contains summary statistics based on standardized empirical-QQ plots. The mean, median and maximum differences in the standardized empirical-QQ plots are provided. The third set of statistics consists of summary statistics from the raw empirical-QQ plots so they are on the scale of the variable in question. And the last set of statistics provides the variance ratio of treatment over control (which should equal 1 if there is perfect balance), and the t test of difference of means (the paired t test is provided post-matching). If they are calculated, the bootstrap Kolmogorov-Smirnov test results are also provided here.

The balance results make clear that `nodegr` is poorly balanced both before *and* after matching. Seventy-one percent of treatment observations have a high school diploma while seventy-seven percent of control observations do. And this difference is highly significant.

Next, let's consider another variable, `re74`, which is real earnings of participants in 1974:

```
R> MatchBalance(Tr ~ re74, match.out = rr1, nboots = 1000, data = lalonde)

***** (V1) re74 *****
                        Before Matching                 After Matching
mean treatment........     2095.6                          2095.6
mean control..........       2107                          2193.3
std mean diff.........   -0.23437                         -2.0004

mean raw eQQ diff.....     487.98                          869.16
med  raw eQQ diff.....          0                               0
max  raw eQQ diff.....       8413                           10305

mean eCDF diff........   0.019223                        0.054701
med  eCDF diff........     0.0158                        0.050872
max  eCDF diff........   0.047089                         0.12209

var ratio (Tr/Co).....     0.7381                          0.75054
T-test p-value........    0.98186                         0.84996
KS Bootstrap p-value..      0.559                       < 2.22e-16
KS Naive p-value......    0.97023                        0.011858
KS Statistic..........   0.047089                         0.12209
```

The balance of the `re74` variable has been made *worse* by matching. Before matching, treatment and control observations were only 11.4 dollars apart and this difference was not significant as judged by either a t test for difference of means or by the Kolmogorov-Smirnov test which tests for a significant difference across the entire distribution. After matching, the mean difference increases to almost 100 dollars, but it still not significant. Unfortunately, the mean, median and maximum differences in the empirical-QQ plots increase sharply. And consistent with this, the KS tests shows a large and significant difference between the distribution of control and treatment observations.

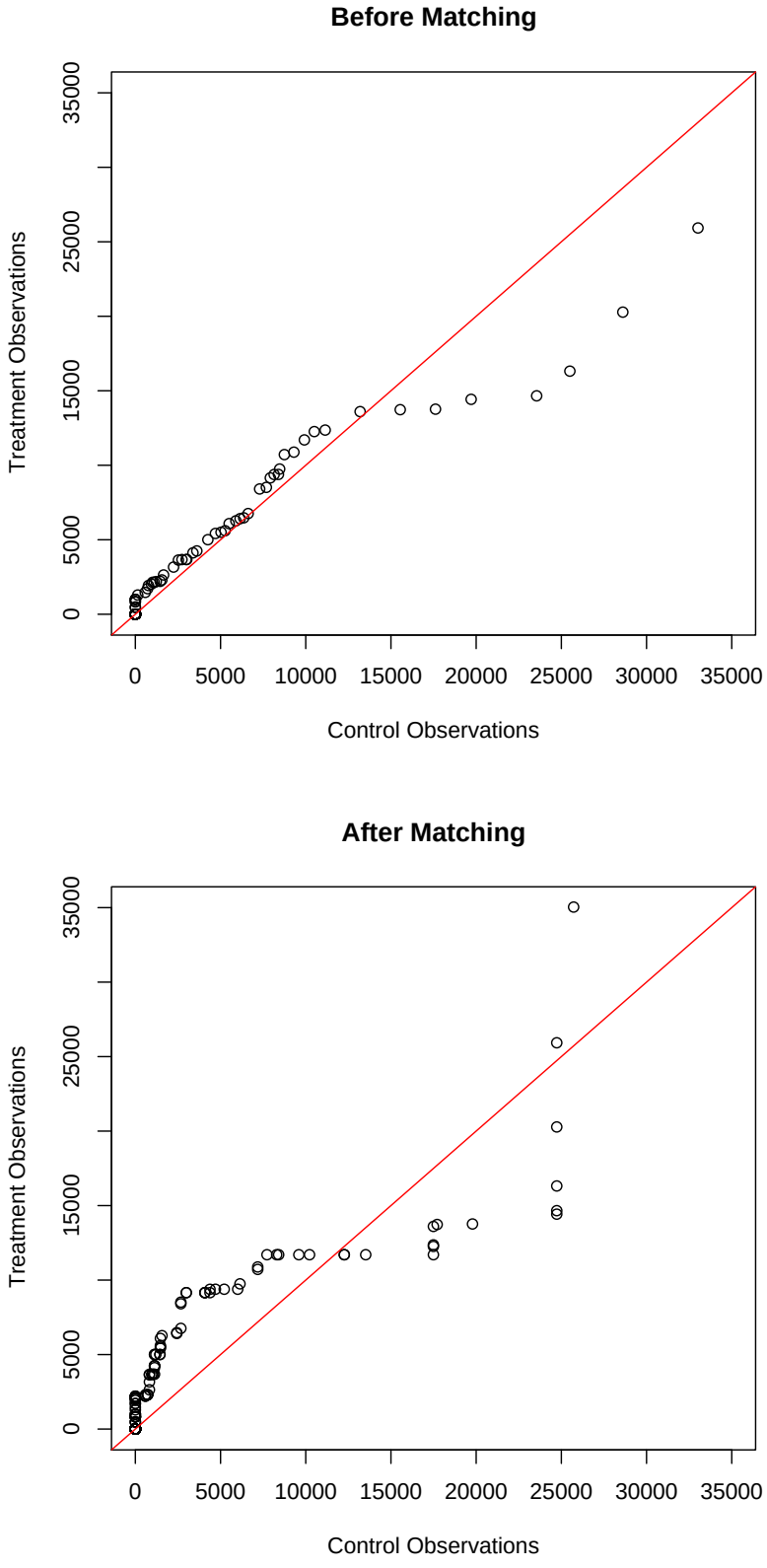

Figure 1: Empirical-QQ Plot of `re74` before and after pscore matching.

Figure 1 plots the empirical-QQ plot of this variable before and after matching, and it shows that balance has been made worse by matching. The after matching portion of this figure (without the captions) was generated by the following code:

```
R> qqplot(lalonde$re74[rr1$index.control], lalonde$re74[rr1$index.treated])
R> abline(coef = c(0, 1), col = 2)
```

The `index.control` and `index.treated` indices which are in the object returned by `Match` are vectors containing the observation numbers from the original dataset for the treated (control) observations in the matched dataset. Both indices together can be used to construct the matched dataset. The matched dataset is also returned in the `mdata` object—see the `Match` manual page for details.

This example shows that it is important to not simply look at differences of means. It is important to examine more general summaries of the distributions. Both the descriptive eQQ statistics and the KS test made clear that matching resulted in worse balance for this variable.

When faced with a propensity score which makes balance *worse*, it is sometimes possible to learn from the balance output and improve the propensity score. However, because the covariates are correlated with each other, it is difficult to know exactly how one should change the propensity score model. For example, the no highschool degree variable has significant imbalance both before and after matching. Should we interact it with other variables or do something else? It may be the case that we should not change the specification of `nodegr`, but instead change the specification of some other variable with which `nodegr` is correlated. In this example, that turns out to work.

Consider the following propensity score model proposed by Dehejia and Wahba (1999) to be used for the LaLonde data:

```
R> dw.pscore <- glm(Tr ~ age + I(age^2) + educ + I(educ^2) + black + hisp +
+    married + nodegr + re74 + I(re74^2) + re75 + I(re75^2) + u74 + u75,
+    family = binomial, data = lalonde)
R> rr.dw <- Match(Y = Y, Tr = Tr, X = dw.pscore$fitted)
```

This model adds second-order polynomials to the continuous variables we have: `age`, `educ`, `re74` and `re75`. And it adds indicator variables for whether income in 1974 and 1975 were zero: `u74`, `u75`. Note that this pscore model does not do anything different with `nodegr` than the previous one we used.

The Dehejia and Wahba model does, however, perform significantly better. See Appendix C for the full output for the following call to `MatchBalance`:

```
R> MatchBalance(Tr ~ age + I(age^2) + educ + I(educ^2) + black + hisp +
+    married + nodegr + re74 + I(re74^2) + re75 + I(re75^2) + u74 + u75 +
+    I(re74 * re75) + I(age * nodegr) + I(educ * re74) + I(educ * re75),
+    data = lalonde, match.out = rr.dw, nboots = 1000)
```

To focus, for example, on a few variable, consider the balance of `nodegr`, `re74` and `re74^2`:

```
R> MatchBalance(Tr ~ nodegr + re74 + I(re74^2), match.out = rr.dw,
+    nboots = 1000, data = lalonde)
```

```
***** (V1) nodegr *****
                        Before Matching                 After Matching
mean treatment........    0.70811                       0.70811
mean control..........    0.83462                       0.69189
std mean diff.........    -27.751                        3.5572

mean raw eQQ diff.....    0.12432                      0.014451
med  raw eQQ diff.....          0                             0
max  raw eQQ diff.....          1                             1

mean eCDF diff........   0.063254                     0.0072254
med  eCDF diff........   0.063254                     0.0072254
max  eCDF diff........    0.12651                      0.014451

var ratio (Tr/Co).....     1.4998                       0.96957
T-test p-value........  0.0020368                       0.49161


***** (V2) re74 *****
                        Before Matching                 After Matching
mean treatment........     2095.6                        2095.6
mean control..........       2107                        1624.3
std mean diff.........   -0.23437                        9.6439

mean raw eQQ diff.....     487.98                        467.33
med  raw eQQ diff.....          0                             0
max  raw eQQ diff.....       8413                         12410

mean eCDF diff........   0.019223                      0.019782
med  eCDF diff........     0.0158                      0.018786
max  eCDF diff........   0.047089                      0.046243

var ratio (Tr/Co).....     0.7381                        2.2663
T-test p-value........    0.98186                       0.22745
KS Bootstrap p-value..      0.569                         0.269
KS Naive p-value......    0.97023                        0.8532
KS Statistic..........   0.047089                      0.046243


***** (V3) I(re74^2) *****
                        Before Matching                 After Matching
mean treatment........   28141434                      28141434
mean control..........   36667413                      13117852
std mean diff.........    -7.4721                         13.167

mean raw eQQ diff.....   13311731                      10899373
med  raw eQQ diff.....          0                             0
```

```
max  raw eQQ diff.....  365146387                    616156569

mean eCDF diff........   0.019223                     0.019782
med  eCDF diff........     0.0158                     0.018786
max  eCDF diff........   0.047089                     0.046243

var ratio (Tr/Co).....    0.50382                      7.9006
T-test p-value........    0.51322                     0.08604
KS Bootstrap p-value..      0.569                       0.269
KS Naive p-value......    0.97023                      0.8532
KS Statistic..........   0.047089                     0.046243


Before Matching Minimum p.value: 0.0020368
Variable Name(s): nodegr  Number(s): 1

After Matching Minimum p.value: 0.08604
Variable Name(s): I(re74^2)  Number(s): 3
```

The balance of the `nodegr` variable has significantly improved from that of the unmatched dataset. The difference has been shrunk to the point that the remaining imbalance in this covariate is probably not a serious concern.

The balance in the income in 1974 is better than that produced by the previous pscore model, but it is still worse than balance in the unmatched data. The means of the `re74` variable across treatment and control groups and the standardized mean, median and maximum difference in the eQQ plots are *increased* by matching. Although the differences are not significant, they are if we examine the balance output for `re74^2`.

Note that the eQQ and KS test results are exactly the same for `re74` and `re74^2` as is to be expected because these non-parametric tests depend on the ranks of the observations rather than their precise values. However, the KS test is less sensitive to mean differences than the t test. It is more sensitive than the t test to differences in the distributions beyond the first two moments. In this case, the t test p value for `re74^2` is much lower than it is for `re74`: 0.086 versus 0.23.

Since the previous outcome is usually the most important confounder we need to worry about, the remaining imbalance in this variable is of serious concern. And it is further troubling that matching is making balance worse in this variable than doing nothing at all!

As this example hopefully demonstrates, moving back and forth from balance statistics to changing the matching model is a tedious process. Fortunately, as described in Section 2.3, the problem can be clearly posed as an optimization problem that can be algorithmically solved.

3.2. Genetic matching

The `GenMatch` function can be used for our example problem, and it greatly improves balance even over the Dehejia and Wahba (1999) propensity score model. `GenMatch` can be used with our without a propensity score model. In this example, we will not make use of any

propensity score model just to demonstrate that `GenMatch` can perform well even without a human providing such a model. However, in general, inclusion of a good propensity score model helps `GenMatch`.

```
R> X <- cbind(age, educ, black, hisp, married, nodegr, re74, re75, u74, u75)
R> BalanceMatrix <- cbind(age, I(age^2), educ, I(educ^2), black, hisp,
+    married, nodegr, re74, I(re74^2), re75, I(re75^2), u74, u75,
+    I(re74 * re75), I(age * nodegr), I(educ * re74), I(educ * re75))
R> gen1 <- GenMatch(Tr = Tr, X = X, BalanceMatrix = BalanceMatrix,
+    pop.size = 1000)
```

`GenMatch` takes four key arguments. The first two, `Tr` and `X`, are just the same as those of the `Match` function: the first is a vector for the treatment indicator and the second a matrix which contains the covariates which we wish to match on. The third key argument, `BalanceMatrix`, is a matrix containing the variables we wish to achieve balance on. This is by default equal to `X`, but it can be a matrix which contains more or less variables than `X` or variables which are transformed in various ways. It should generally contain the variables and the function of these variables that we wish to balance. In this example, I have made `BalanceMatrix` contain the same terms we had `MatchBalance` test balance for, and this, in general, is good practice. If you care about balance for a given function of the covariates, you should put it in `BalanceMatrix` just like how you should put it into the equation in `MatchBalance`.

The `pop.size` argument is important and greatly influences how long the function takes to run. This argument controls the population size used by the evolutionary algorithm–i.e., it is the number of individuals `genoud` uses to solve the optimization problem. This argument is also the number of random trail solutions which are tried at the beginning of the search process. The theorems proving that genetic algorithms find good solutions are asymptotic in population size. Therefore, it is important that this value not be small (Vose 1993; Nix and Vose 1992). On the other hand, computational time is finite so obvious trade-offs must be made.

`GenMatch` has a large number of other options which are detailed in its help page. The options controlling features of the matching itself, such as whether to match with replacement, are the same as those of the `Match` function. But many other options are specific to `GenMatch` because they control the optimization process. The most important of these aside from `pop.size`, are `wait.generations` and `max.generations`.

In order to obtain balance statistics, we can simply do the following with the output object (`gen1`) returned by the call to `GenMatch` above:

```
R> mgen1 <- Match(Y = Y, Tr = Tr, X = X, Weight.matrix = gen1)

R> MatchBalance(Tr ~ age + I(age^2) + educ + I(educ^2) + black + hisp +
+    married + nodegr + re74 + I(re74^2) + re75 + I(re75^2) + u74 + u75 +
+    I(re74 * re75) + I(age * nodegr) + I(educ * re74) + I(educ * re75),
+    data = lalonde, match.out = mgen1, nboots = 1000)
```

The balance results from this `GenMatch` run are excellent. The full output from this call to `MatchBalance` is include in Appendix D. Note that `GenMatch` is a stochastic algorithm so your results may not be exactly the same.

The balance is now excellent for all variables. As shown in Appendix D, the *smallest p* value across all of the variables tested in `MatchBalance` is 0.408 (for `I(educ * re74))` compared with 0.086 for the Dehejia and Wahba propensity score model (for `re74^2`) and the pre-matching value of 0.002 (for `nodegr`).

As for our propensity score examples, the balance output for `nodegr`, `re74` and `re74^2` are presented for close examination:

```
R> MatchBalance(Tr ~ nodegr + re74 + I(re74^2), match.out = mgen1,
+    nboots = 1000, data = lalonde)

***** (V1) nodegr *****
                       Before Matching                 After Matching
mean treatment........    0.70811                      0.70811
mean control..........    0.83462                      0.70811
std mean diff.........    -27.751                            0

mean raw eQQ diff.....    0.12432                            0
med  raw eQQ diff.....          0                            0
max  raw eQQ diff.....          1                            0

mean eCDF diff........   0.063254                            0
med  eCDF diff........   0.063254                            0
max  eCDF diff........    0.12651                            0

var ratio (Tr/Co).....     1.4998                            1
T-test p-value........  0.0020368                            1


***** (V2) re74 *****
                       Before Matching                 After Matching
mean treatment........     2095.6                       2095.6
mean control..........       2107                       2017.2
std mean diff.........   -0.23437                       1.6031

mean raw eQQ diff.....     487.98                          174
med  raw eQQ diff.....          0                            0
max  raw eQQ diff.....       8413                       7175.7

mean eCDF diff........   0.019223                    0.0066891
med  eCDF diff........     0.0158                    0.0037313
max  eCDF diff........   0.047089                     0.029851

var ratio (Tr/Co).....     0.7381                       1.1519
T-test p-value........    0.98186                      0.51757
KS Bootstrap p-value..      0.578                        0.806
KS Naive p-value......    0.97023                      0.99976
KS Statistic..........   0.047089                     0.029851
```

```
***** (V3) I(re74^2) *****
                       Before Matching                 After Matching
mean treatment........   28141434                      28141434
mean control..........   36667413                      24686484
std mean diff.........    -7.4721                        3.0279

mean raw eQQ diff.....   13311731                       4823772
med  raw eQQ diff.....          0                             0
max  raw eQQ diff.....  365146387                     451383821

mean eCDF diff........   0.019223                      0.0066891
med  eCDF diff........     0.0158                      0.0037313
max  eCDF diff........   0.047089                       0.029851

var ratio (Tr/Co).....    0.50382                        1.5233
T-test p-value........    0.51322                        0.4652
KS Bootstrap p-value..      0.578                         0.806
KS Naive p-value......    0.97023                       0.99976
KS Statistic..........   0.047089                       0.029851


Before Matching Minimum p.value: 0.0020368
Variable Name(s): nodegr  Number(s): 1

After Matching Minimum p.value: 0.4652
Variable Name(s): I(re74^2)  Number(s): 3
```

The empirical-QQ plot for `re74`, as shown in Figure 2, now looks good, especially when compared with Figure 1. Balance is now improved, and not made worse, by matching.

Now that we have achieved excellent balance, we can examine our estimate of the treatment effect and its standard error. We can do this by simply running `summary` on the object returned by the `Match` function:

```
R> summary(mgen1)


Estimate...  1671.2
AI SE......  889.63
T-stat.....  1.8785
p.val......  0.060306

Original number of observations..............  445
Original number of treated obs...............  185
Matched number of observations...............  185
Matched number of observations  (unweighted).  268
```

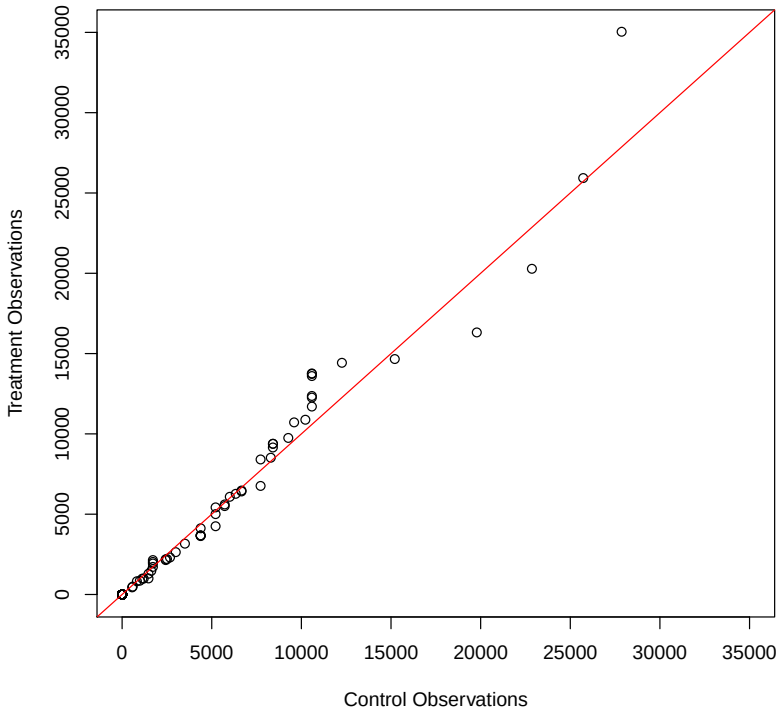

Figure 2: Empirical-QQ plot of `re74` using `GenMatch`.

The estimate of the treatment effect for the treated is \$1,671.20 with a standard error of 889.63. By default, the Abadie-Imbens (AI) standard error is printed (Abadie and Imbens 2006). In order to also obtain the usual Neyman standard error, one may call the `summary` function with the `full=TRUE` option.

The `summary` function also provides the number of observations in total (445), the number of treated observations (185), the number of matched pairs that were produced when the ties are properly weighted (185), and the number of matched pairs without using the weights which adjust for ties (268).

3.3. Parallel and cluster processing

`GenMatch` is a computationally intensive algorithm because it constructs matched datasets for each trail set of covariate weights. Fortunately, as with most genetic algorithms, the algorithm easily parallelizes. This functionality has been built directly in the **rgenoud** package and be readily accessed by `GenMatch`. The parallelization can be used for either multiple CPU computers or a cluster of computers, and makes use of R's **snow** (simple network of workstations) package (Rossini, Tierney, and Li 2007). Simulations to estimate how well the parallel algorithm scales with multiple CPUs are provided below. On a single computer with multiple CPUs, the proportion of time saved is almost linear in the number of CPUs if the dataset size is large. For a cluster of separate computers, the algorithm is significantly faster

for every extra node which is added, but the time savings are significantly less than linear. The exact amount of time saved depends on network latency and a host of other factors.

Two `GenMatch` options control the parallel processing: `cluster` and `balance`. The `cluster` option can either be an object of the 'cluster' class returned by one of the `makeCluster` commands in the **snow** package or a vector of machine names so that `GenMatch` can setup the cluster automatically via secure-shell (SSH). If it is the latter, the vector passed to the `cluster` option should look like the following:

```
R> c("localhost", "localhost", "musil", "musil", "deckard")
```

This vector would create a cluster with four nodes: two on the localhost another on 'deckard' and two on the machine named 'musil'. Two nodes on a given machine make sense if the machine has two or more chips/cores. `GenMatch` will setup a SOCK cluster by a call to `makeSOCKcluster`. This will require the user to type in her password for each node as the cluster is by default created via SSH. One can add on user names to the machine name if it differs from the current shell: `username@musil`. Other cluster types, such as PVM and MPI, which do not require passwords, can be created by directly calling `makeCluster`, and then passing the returned cluster object to `GenMatch`. For example, one can manually setup a cluster with a direct call to `makeCluster` as follows:

```
R> library("snow")
R> library("Matching")
R> data("lalonde")
R> attach(lalonde)
R> cl <- makeCluster(c("musil", "quetelet", "quetelet"), type = "SOCK")
R> X <- cbind(age, educ, black, hisp, married, nodegr, u74, u75, re75, re74)
R> genout <- GenMatch(Tr = treat, X = X, cluster = cl)
R> stopCluster(cl)
```

Note the `stopCluster(cl)` command which is needed because we setup the cluster output of `GenMatch`. So, we much manually shut the connections down.

The second `GenMatch` option which controls the behavior of parallel processing is the `balance` option. This is a logical flag which controls if load balancing is done across the cluster. Load balancing can result in better cluster utilization; however, increased communication can reduce performance. This options is best used if each individual call to `Match` takes at least several minutes to calculate or if the nodes in the cluster vary significantly in their performance.

Designing parallel software applications is difficult. A lot of work and trail-and-error has gone into writing the C++ functions which `GenMatch` relies upon to ensure that they are reliable and fast when run either serially or in parallel. Parallel execution is especially tricky because an algorithm which may be fast in serial mode can cause unexpected bottlenecks when run in parallel (such as a cache-bottleneck when executing SSE3 instructions via BLAS).

We now explore how well `GenMatch` scales with additional CPUs by using the following benchmark code:

```
R> library("Matching")
R> data("lalonde")
```

	1 CPU	2 CPUs	3 CPUs	4 CPUs
1780 Observations				
run time (seconds)	2557	1372	950	749
x CPU/1 CPU run time		0.54	0.37	0.29
1335 Observations				
run time (seconds)	826	475	317	255
x CPU/1 CPU run time		0.58	0.38	0.31
890 Observations				
run time (seconds)	532	338	233	193
x CPU/1 CPU run time		0.64	0.44	0.36

Table 1: Using multiple computer chips to run `GenMatch`.

```
R> attach(lalonde)
R> X <- cbind(age, educ, black, hisp, married, nodegr, u74, u75, re75, re74)
R> Xbig <- rbind(X, X, X, X)
R> Ybig <- c(treat, treat, treat, treat)
R> GenMatch(Tr = Ybig, X = Xbig, BalanceMatrix = Xbig, estimand = "ATE",
+    M = 1, pop.size = 1000, max.generations = 10, wait.generations = 1,
+    int.seed = 3818, unif.seed = 3527, cluster = c("localhost",
+    "localhost", "localhost", "localhost"))
```

This example makes use of four computer chips: note the four calls to `localhost`. The dataset is replicated four times (e.g., `Xbig` and `Ybig`) to obtain 1780 observations. And smaller datasets are created by not replicating the observations as often. The options `int.seed` and `unif.seed` set the random number seeds in order to ensure replication. These options are passed onto the `genoud` function from the **rgenoud** package. Please see that package for details.

Table 1 presents the average run times of this code as it is run on one to four CPUs and on various dataset sizes.[10]

`GenMatch` scales more efficiently across computer chips as the dataset size becomes larger. With 1780 observations, using four computer chips results in a run time which is 29% of the single chip run time. This is a good increase in performance given that if parallelization were as efficient as possible, using four chips would result in 25% of the run time as that of using a single chip. Of course, perfect parallelization is not possible given the overhead involved in setting up parallel computations.[11] Note that that scaling from one to two CPUs

[10]There is no significant difference in run times across different invocations of the same commands so no variance estimates are presented, just average run times. A four core Xeon processor (5150 @ 2.66GHz) computer running 64-bit Linux (Ubuntu Dapper) was used, and all extraneous daemons were shutdown.

[11]The efficiency of this parallelization is more impressive given that the test runs were run on a four core Xeon processor (5150) which is not really a four core chip. This chip actually consists of two (dual-core) Woodcrest chips. The two chips have no way to communicate directly with each other. All communications between them have to go through a shared front-side bus with the memory controller hub, or north bridge. And each chip independently accesses the cache coherency scheme.

is closer to the theoretical efficiency bound (1.08 = 0.54/0.5) then scaling from one to four chips (1.16 = 0.29/0.25). This may be due to the issue pointed out in Footnote 11.

With 890 observations, using four CPUs takes 36% of the run time as only using one CPU. This is a significantly smaller efficiency gain than that achieved with the dataset with 1780 observations.

It is clear from Table 1 that the computational time it takes to execute a matching algorithm does not increase linearly with sample size. The computational time increases as a polynomial of the sample size, and the average asymptotic order of the `Match` function is approximately $O(N^2)log(N)$.[12] The run times in Table 1 are generally consistent with this. Although the `Match` function increases in polynomial time, the problem which `GenMatch` attempts to solve (that of finding the best distance metric) increases exponentially in sample size, just like the traveling salesman problem. That is, the set of possible matched datasets grows exponentially with sample size.

4. Conclusion

The functions in **Matching** have many more options than can be reviewed in this brief paper. For additional details see the manual pages for the functions included in the R package. The **Matching** package includes four functions in addition to `Match`, `GenMatch`, and `MatchBalance`: `Matchby` (for large datasets), `qqstats` (descriptive eQQ statistics), `ks.boot` (bootstrap version of `ks.test`) and `balanceUV` (univariate balance statistics).

A great deal of effort has been made in order to ensure that the matching functions are as fast as possible. The computationally intensive functions are written in C++ which make extensive use of the BLAS libraries, and `GenMatch` can be used with multiple computers, CPUs or cores to perform parallel computations. The C++ functions have been written so that the GNU `g++` compiler does a good job of optimizing them. Indeed, compiling the **Matching** package with the Intel C++ compiler does not result in faster code. This is unusual with floating point code, and is the result of carefully writing code so that the GNU compiler is able to optimize it aggressively. Moreover, the `Matchby` function has been tuned to work well with large datasets.

After intensive benchmarking and instrumenting the code, it was determined that performance on OS X was seriously limited because the default OS X memory allocator is not as efficient as Lea (2000)'s `malloc` given the frequent memory allocations made by the matching code. The matching algorithm was rewritten in order to be more efficient with memory on all platforms, and on OS X, **Matching** is compiled against Lea's `malloc` which is something more packages for R may wish to do. For details see Sekhon (2006b).

The literature on matching methods is developing quickly with new innovations being made by a variety of researchers in fields ranging from economics, epidemiology and political science to sociology and statistics. Hence, new options are being added frequently.

[12] The precise asymptotic order is difficult to calculate because assumptions have to be made about various features of the data such as the proportion of ties.

References

Abadie A (2002). "Bootstrap Tests for Distributional Treatment Effect in Instrumental Variable Models." *Journal of the American Statistical Association*, **97**(457), 284–292.

Abadie A, Imbens G (2006). "Large Sample Properties of Matching Estimators for Average Treatment Effects." *Econometrica*, **74**, 235–267.

Andam KS, Ferraro PJ, Pfaff A, Sanchez-Azofeifa GA, Robalino JA (2008). "Measuring the Effectiveness of Protected Area Networks in Reducing Deforestation." *Proceedings of the National Academy of Sciences*, **105**(42), 16089–16094.

Barnard J, Frangakis CE, Hill JL, Rubin DB (2003). "Principal Stratification Approach to Broken Randomized Experiments: A Case Study of School Choice Vouchers in New York City." *Journal of the American Statistical Association*, **98**(462), 299–323.

Bowers J, Hansen B (2005). "Attributing Effects to A Cluster Randomized Get-Out-The-Vote Campaign." *Technical Report 448*, Statistics Department, University of Michigan. URL http://www-personal.umich.edu/~jwbowers/PAPERS/bowershansen2006-10TechReport.pdf.

Christakis NA, Iwashyna TI (2003). "The Health Impact of Health Care on Families: A matched cohort study of hospice use by decedents and mortality outcomes in surviving, widowed spouses." *Social Science & Medicine*, **57**(3), 465–475.

Cochran WG, Rubin DB (1973). "Controlling Bias in Observational Studies: A Review." *Sankhya A*, **35**, 417–446.

Dawid AP (1979). "Conditional Independence in Statistical Theory." *Journal of the Royal Statistical Society B*, **41**(1), 1–31.

Dehejia R, Wahba S (1999). "Causal Effects in Non-Experimental Studies: Re-Evaluating the Evaluation of Training Programs." *Journal of the American Statistical Association*, **94**(448), 1053–1062.

Dehejia RH, Wahba S (2002). "Propensity Score Matching Methods for Nonexperimental Causal Studies." *Review of Economics and Statistics*, **84**(1), 151–161.

Diamond A, Sekhon JS (2005). "Genetic Matching for Estimating Causal Effects: A General Multivariate Matching Method for Achieving Balance in Observational Studies." *Technical report*, Department of Political Science, UC Berkeley. URL http://sekhon.berkeley.edu/papers/GenMatch.pdf.

Diprete TA, Engelhardt H (2004). "Estimating Causal Effects with Matching Methods in the Presence and Absence of Bias Cancellation." *Sociological Methods & Research*, **32**(4), 501–528.

Eggers A, Hainmueller J (2009). "The Value of Political Power: Estimating Returns to Office in Post-War British Politics." *American Political Science Review*, **103**(4), 513–533.

Fechner GT (1966). *Elements of Psychophysics, Volume 1*. Rinehart & Winston, New York. Translated by Helmut E. Adler and edited by D. H. Howes and E. G. Boring.

Gilligan MJ, Sergenti EJ (2008). "Do UN Interventions Cause Peace? Using Matching to Improve Causal Inference." *Quarterly Journal of Political Science*, **3**, 89–122.

Gordon SC (2009). "Assessing Partisan Bias in Federal Public Corruption Prosecutions." *American Political Science Review*, **103**(4), 534–554.

Hansen BB (2004). "Full Matching in an Observational Study of Coaching for the SAT." *Journal of the American Statistical Association*, **99**, 609–618.

Hansen BB, Klopfer SO (2006). "Optimal Full Matching and Related Designs via Network Flows." *Journal of Computational and Graphical Statistics*, **15**, 609–627.

Heckman JJ, Ichimura H, Smith J, Todd P (1998). "Characterizing Selection Bias Using Experimental Data." *Econometrica*, **66**(5), 1017–1098.

Heinrich CJ (2007). "Demand and Supply-Side Determinants of Conditional Cash Transfer Program Effectiveness." *World Development*, **35**(1), 121–143.

Herron M, Wand J (2007). "Assessing Partisan Bias in Voting Technology: The Case of the 2004 New Hampshire Recount." *Electoral Studies*, **26**(2), 247–261.

Holland PW (1986). "Statistics and Causal Inference." *Journal of the American Statistical Association*, **81**(396), 945–960.

Hopkins DJ (2010). "Politicized Places: Explaining Where and When Immigrants Provoke Local Opposition." *American Political Science Review*, **104**(1), 40–60.

Horvitz DG, Thompson DJ (1952). "A Generalization of Sampling without Replacement from a Finite Universe." *Journal of the American Statistical Association*, **47**, 663–685.

Imai K (2005). "Do Get-Out-The-Vote Calls Reduce Turnout? The Importance of Statistical Methods for Field Experiments." *American Political Science Review*, **99**(2), 283–300.

LaLonde R (1986). "Evaluating the Econometric Evaluations of Training Programs with Experimental Data." *American Economic Review*, **76**, 604–20.

Lea D (2000). "A Memory Allocator." URL http://gee.cs.oswego.edu/dl/html/malloc.html.

Lenz GS, Ladd JM (2009). "Exploiting a Rare Communication Shift to Document the Persuasive Power of the News Media." *American Journal of Political Science*, **53**(2), 394–410.

Mebane, Jr WR, Sekhon JS (2011). "Genetic Optimization Using Derivatives: The **rgenoud** Package for R." *Journal of Statistical Software*, **42**(11), 1–26. URL http://www.jstatsoft.org/v42/i11/.

Morgan SL, Harding DJ (2006). "Matching Estimators of Causal Effects: Prospects and Pitfalls in Theory and Practice." *Sociological Methods & Research*, **35**(1), 3–60.

Nix AE, Vose MD (1992). "Modeling Genetic Algorithms with Markov Chains." *Annals of Mathematics and Artificial Intelligence*, **5**, 79–88.

Raessler S, Rubin D (2005). "Complications When Using Nonrandomized Job Training Data to Draw Causal Inferences." In *Proceedings of the International Statistical Institute*. URL http://www.websm.org/uploadi/editor/1132827726ISI.Susie.doc.

R Development Core Team (2011). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Rosenbaum PR (1989). "Optimal Matching for Observational Studies." *Journal of the American Statistical Association*, **84**(408), 1024–1032.

Rosenbaum PR (1991). "A Characterization of Optimal Designs for Observational Studies." *Journal of the Royal Statistical Society B*, **53**(3), 597–610.

Rosenbaum PR (2002). *Observational Studies*. 2nd edition. Springer-Verlag, New York.

Rosenbaum PR, Rubin DB (1983). "The Central Role of the Propensity Score in Observational Studies for Causal Effects." *Biometrika*, **70**(1), 41–55.

Rosenbaum PR, Rubin DB (1985). "Constructing a Control Group Using Multivariate Matched Sampling Methods That Incorporate the Propensity Score." *The American Statistician*, **39**(1), 33–38.

Rossini AJ, Tierney L, Li N (2007). "Simple Parallel Statistical Computing in R." *Journal of Computational and Graphical Statistics*, **16**(2), 399–420.

Rubin D, Stuart EA (2006). "Affinely Invariant Matching Methods with Discriminant Mixtures of Proportional Ellipsoidally Symmetric Distributions." *The Annals of Statistics*, **34**(4), 1814–1826.

Rubin DB (1974). "Estimating Causal Effects of Treatments in Randomized and Nonrandomized Studies." *Journal of Educational Psychology*, **66**, 688–701.

Rubin DB (1976a). "Multivariate Matching Methods That are Equal Percent Bias Reducing, I: Some Examples." *Biometrics*, **32**(1), 109–120.

Rubin DB (1976b). "Multivariate Matching Methods That are Equal Percent Bias Reducing, II: Maximums on Bias Reduction for Fixed Sample Sizes." *Biometrics*, **32**(1), 121–132.

Rubin DB (1977). "Assignment to a Treatment Group on the Basis of a Covariate." *Journal of Educational Statistics*, **2**, 1–26.

Rubin DB (1978). "Bayesian Inference for Causal Effects: The Role of Randomization." *The Annals of Statistics*, **6**(1), 34–58.

Rubin DB (1979). "Using Multivariate Sampling and Regression Adjustment to Control Bias in Observational Studies." *Journal of the American Statistical Association*, **74**, 318–328.

Rubin DB (1980). "Bias Reduction Using Mahalanobis-Metric Matching." *Biometrics*, **36**(2), 293–298.

Rubin DB (1990). "Comment: Neyman (1923) and Causal Inference in Experiments and Observational Studies." *Statistical Science*, **5**(4), 472–480.

Rubin DB (1997). "Estimating Causal Effects from Large Data Sets Using Propensity Scores." *Annals of Internal Medicine*, **127**(8S), 757–763.

Rubin DB (2001). "Using Propensity Scores to Help Design Observational Studies: Application to the Tobacco Litigation." *Health Services & Outcomes Research Methodology*, **2**(1), 169–188.

Rubin DB (2006). *Matched Sampling for Causal Effects*. Cambridge University Press, New York.

Rubin DB, Thomas N (1992). "Affinely Invariant Matching Methods with Ellipsoidal Distributions." *The Annals of Statistics*, **20**(2), 1079–1093.

Sekhon JS (2006a). "Alternative Balance Metrics for Bias Reduction in Matching Methods for Causal Inference." *Technical report*, Department of Political Science, UC Berkeley. URL http://sekhon.berkeley.edu/papers/SekhonBalanceMetrics.pdf.

Sekhon JS (2006b). "The Art of Benchmarking: Evaluating the Performance of R on Linux and OS X." *The Political Methodologist*, **14**(1).

Sekhon JS, Grieve R (2011). "A Non-Parametric Matching Method for Bias Adjustment with Applications to Economic Evaluations." *Health Economics*. Forthcoming.

Sekhon JS, Mebane Jr WR (1998). "Genetic Optimization Using Derivatives: Theory and Application to Nonlinear Models." *Political Analysis*, **7**, 189–203.

Smith HL (1997). "Matching with Multiple Controls to Estimate Treatment Effects in Observational Studies." *Sociological Methodology*, **27**, 305–353.

Splawa-Neyman J (1923). "On the Application of Probability Theory to Agricultural Experiments. Essay on Principles. Section 9." *Statistical Science*, **5**(4), 465–472. Trans. Dorota M. Dabrowska and Terence P. Speed.

Vose MD (1993). "Modeling Simple Genetic Algorithms." In LD Whitley (ed.), *Foundations of Genetic Algorithms 2*. Morgan Kaufmann, San Mateo, CA.

Winship C, Morgan S (1999). "The Estimation of Causal Effects from Observational Data." *Annual Review of Sociology*, **25**, 659–707.

A. Equal percent bias reduction (EPBR)

Affinely invariant matching methods, such as Mahalanobis metric matching and propensity score matching (if the propensity score is estimated by logistic regression), are equal percent bias reducing if all of the covariates used have ellipsoidal distributions (Rubin and Thomas 1992)—e.g., distributions such as the normal or t—or if the covariates are mixtures of proportional ellipsoidally symmetric (DMPES) distributions (Rubin and Stuart 2006).[13]

To formally define EPBR, let Z be the expected value of X in the matched control group. Then, as outlined in Rubin (1976a), a matching procedure is EPBR if

$$E(X \mid T = 1) - Z \quad = \gamma \{E(X \mid T = 1) - E(X \mid T = 0)\}$$

for a scalar $0 \leq \gamma \leq 1$. In other words, we say that a matching method is EPBR for X when the percent reduction in the biases of each of the matching variables is the same. One obtains the same percent reduction in bias for any linear function of X if and only if the matching method is EPBR for X. Moreover, if a matching method is not EPBR for X, the bias for some linear function of X is increased even if all univariate covariate means are closer in the matched data than the unmatched (Rubin 1976a).

Even if the covariates have elliptic distributions, in finite samples they may not. Then Mahalanobis distance may not be optimal because the matrix used to scale the distances, the covariance matrix of X, can be improved upon.

The EPBR property itself is limited and in a given substantive problem it may not be desirable. This can arise if it is known that one covariate has a large nonlinear relationship with the outcome while another does not—e.g., $Y = X_1^4 + X_2$, where $X_1 > 1$. In such a case, reducing bias in X_1 will be more important than X_2.

B. Full balance output for the first propensity score

Attached is the full output from `MatchBalance` for the first propensity score model we estimated:

```
R> glm1 <- glm(Tr ~ age + educ + black + hisp + married + nodegr +
+    re74 + re75, family = binomial, data = lalonde)
R> rr1 <- Match(Y = Y, Tr = Tr, X = glm1$fitted)
R> MatchBalance(Tr ~ age + I(age^2) + educ + I(educ^2) + black + hisp +
+    married + nodegr + re74 + I(re74^2) + re75 + I(re75^2) + u74 + u75 +
+    I(re74 * re75) + I(age * nodegr) + I(educ * re74) + I(educ * re75),
+    match.out = rr1, nboots = 1000, data = lalonde)

***** (V1) age *****
                       Before Matching                 After Matching
mean treatment........     25.816                        25.816
```

[13]Note that DMPES defines a limited set of mixtures. In particular, countably infinite mixtures of ellipsoidal distributions where: (1) all inner products are proportional and (2) where the centers of each constituent ellipsoidal distribution are such that all best linear discriminants between any two components are also proportional.

```
mean control.........     25.054                    25.692
std mean diff........     10.655                    1.7342

mean raw eQQ diff.....    0.94054                   0.73837
med  raw eQQ diff.....          1                         0
max  raw eQQ diff.....          7                         9

mean eCDF diff........   0.025364                  0.021893
med  eCDF diff........   0.022193                  0.020349
max  eCDF diff........   0.065177                  0.061047

var ratio (Tr/Co).....     1.0278                    1.083
T-test p-value........    0.26594                   0.84975
KS Bootstrap p-value..      0.518                     0.362
KS Naive p-value......     0.7481                   0.54314
KS Statistic..........   0.065177                  0.061047


***** (V2) I(age^2) *****
                       Before Matching                After Matching
mean treatment........     717.39                    717.39
mean control.........     677.32                     707.1
std mean diff........     9.2937                    2.3873

mean raw eQQ diff.....     56.076                    46.901
med  raw eQQ diff.....         43                         0
max  raw eQQ diff.....        721                       909

mean eCDF diff........   0.025364                  0.021893
med  eCDF diff........   0.022193                  0.020349
max  eCDF diff........   0.065177                  0.061047

var ratio (Tr/Co).....     1.0115                    1.0072
T-test p-value........    0.33337                   0.80409
KS Bootstrap p-value..      0.518                     0.362
KS Naive p-value......     0.7481                   0.54314
KS Statistic..........   0.065177                  0.061047


***** (V3) educ *****
                       Before Matching                After Matching
mean treatment........     10.346                    10.346
mean control.........     10.088                    10.146
std mean diff........     12.806                    9.9664

mean raw eQQ diff.....    0.40541                   0.23256
med  raw eQQ diff.....          0                         0
```

```
max  raw eQQ diff.....          2                        2

mean eCDF diff........   0.028698                 0.016611
med  eCDF diff........   0.012682                 0.010174
max  eCDF diff........    0.12651                 0.061047

var ratio (Tr/Co).....     1.5513                   1.2344
T-test p-value........    0.15017                   0.1842
KS Bootstrap p-value..      0.009                    0.193
KS Naive p-value......   0.062873                  0.54314
KS Statistic..........    0.12651                 0.061047


***** (V4) I(educ^2) *****
                       Before Matching                After Matching
mean treatment........     111.06                   111.06
mean control..........     104.37                   106.19
std mean diff.........     17.012                    12.39

mean raw eQQ diff.....     8.7189                   4.7384
med  raw eQQ diff.....          0                        0
max  raw eQQ diff.....         60                       60

mean eCDF diff........   0.028698                 0.016611
med  eCDF diff........   0.012682                 0.010174
max  eCDF diff........    0.12651                 0.061047

var ratio (Tr/Co).....     1.6625                   1.2999
T-test p-value........   0.053676                 0.080965
KS Bootstrap p-value..      0.009                    0.193
KS Naive p-value......   0.062873                  0.54314
KS Statistic..........    0.12651                 0.061047


***** (V5) black *****
                       Before Matching                After Matching
mean treatment........    0.84324                  0.84324
mean control..........    0.82692                  0.86847
std mean diff.........     4.4767                  -6.9194

mean raw eQQ diff.....   0.016216                 0.026163
med  raw eQQ diff.....          0                        0
max  raw eQQ diff.....          1                        1

mean eCDF diff........  0.0081601                 0.013081
med  eCDF diff........  0.0081601                 0.013081
max  eCDF diff........    0.01632                 0.026163
```

```
var ratio (Tr/Co).....    0.92503                    1.1572
T-test p-value........    0.64736                   0.40214


***** (V6) hisp *****
                        Before Matching                After Matching
mean treatment........   0.059459                  0.059459
mean control..........    0.10769                   0.04955
std mean diff.........    -20.341                    4.1792

mean raw eQQ diff.....   0.048649                  0.011628
med  raw eQQ diff.....          0                         0
max  raw eQQ diff.....          1                         1

mean eCDF diff........   0.024116                  0.005814
med  eCDF diff........   0.024116                  0.005814
max  eCDF diff........   0.048233                  0.011628

var ratio (Tr/Co).....    0.58288                    1.1875
T-test p-value........   0.064043                   0.46063


***** (V7) married *****
                        Before Matching                After Matching
mean treatment........    0.18919                   0.18919
mean control..........    0.15385                   0.18423
std mean diff.........     8.9995                    1.2617

mean raw eQQ diff.....   0.037838                  0.026163
med  raw eQQ diff.....          0                         0
max  raw eQQ diff.....          1                         1

mean eCDF diff........   0.017672                  0.013081
med  eCDF diff........   0.017672                  0.013081
max  eCDF diff........   0.035343                  0.026163

var ratio (Tr/Co).....     1.1802                    1.0207
T-test p-value........    0.33425                   0.89497


***** (V8) nodegr *****
                        Before Matching                After Matching
mean treatment........    0.70811                   0.70811
mean control..........    0.83462                   0.76757
std mean diff.........    -27.751                   -13.043
```

```
mean raw eQQ diff.....    0.12432                    0.043605
med  raw eQQ diff.....          0                           0
max  raw eQQ diff.....          1                           1

mean eCDF diff........   0.063254                   0.021802
med  eCDF diff........   0.063254                   0.021802
max  eCDF diff........    0.12651                   0.043605

var ratio (Tr/Co).....     1.4998                     1.1585
T-test p-value........  0.0020368                  0.0071385


***** (V9) re74 *****
                       Before Matching                After Matching
mean treatment........     2095.6                     2095.6
mean control..........       2107                     2193.3
std mean diff.........   -0.23437                    -2.0004

mean raw eQQ diff.....     487.98                     869.16
med  raw eQQ diff.....          0                          0
max  raw eQQ diff.....       8413                      10305

mean eCDF diff........   0.019223                   0.054701
med  eCDF diff........     0.0158                   0.050872
max  eCDF diff........   0.047089                    0.12209

var ratio (Tr/Co).....     0.7381                     0.75054
T-test p-value........    0.98186                    0.84996
KS Bootstrap p-value..      0.594                      0.001
KS Naive p-value......    0.97023                   0.011858
KS Statistic..........   0.047089                    0.12209


***** (V10) I(re74^2) *****
                       Before Matching                After Matching
mean treatment........   28141434                   28141434
mean control..........   36667413                   36454686
std mean diff.........    -7.4721                    -7.2857

mean raw eQQ diff.....   13311731                   14189969
med  raw eQQ diff.....          0                          0
max  raw eQQ diff.....  365146387                  566243911

mean eCDF diff........   0.019223                   0.054701
med  eCDF diff........     0.0158                   0.050872
max  eCDF diff........   0.047089                    0.12209
```

```
var ratio (Tr/Co).....    0.50382                    0.85502
T-test p-value........    0.51322                    0.49446
KS Bootstrap p-value..      0.594                      0.001
KS Naive p-value......    0.97023                   0.011858
KS Statistic..........   0.047089                    0.12209


***** (V11) re75 *****
                        Before Matching                After Matching
mean treatment........     1532.1                     1532.1
mean control..........     1266.9                     2179.9
std mean diff.........     8.2363                    -20.125

mean raw eQQ diff.....     367.61                     590.34
med  raw eQQ diff.....          0                          0
max  raw eQQ diff.....     2110.2                     8092.9

mean eCDF diff........   0.050834                   0.050338
med  eCDF diff........   0.061954                   0.049419
max  eCDF diff........    0.10748                   0.098837

var ratio (Tr/Co).....     1.0763                    0.56563
T-test p-value........    0.38527                   0.079002
KS Bootstrap p-value..      0.051                      0.017
KS Naive p-value......    0.16449                   0.069435
KS Statistic..........    0.10748                   0.098837


***** (V12) I(re75^2) *****
                        Before Matching                After Matching
mean treatment........   12654753                   12654753
mean control..........   11196530                   22975211
std mean diff.........     2.6024                     -18.418

mean raw eQQ diff.....    2840830                    7689340
med  raw eQQ diff.....          0                          0
max  raw eQQ diff.....  101657197                  208799779

mean eCDF diff........   0.050834                   0.050338
med  eCDF diff........   0.061954                   0.049419
max  eCDF diff........    0.10748                   0.098837

var ratio (Tr/Co).....     1.4609                    0.68801
T-test p-value........    0.77178                    0.10936
KS Bootstrap p-value..      0.051                      0.017
KS Naive p-value......    0.16449                   0.069435
KS Statistic..........    0.10748                   0.098837
```

```
***** (V13) u74 *****
                        Before Matching                 After Matching
mean treatment........    0.70811                        0.70811
mean control..........       0.75                        0.72027
std mean diff.........    -9.1895                        -2.6679

mean raw eQQ diff.....   0.037838                       0.081395
med  raw eQQ diff.....          0                              0
max  raw eQQ diff.....          1                              1

mean eCDF diff........   0.020946                       0.040698
med  eCDF diff........   0.020946                       0.040698
max  eCDF diff........   0.041892                       0.081395

var ratio (Tr/Co).....     1.1041                         1.0259
T-test p-value........    0.33033                        0.76177


***** (V14) u75 *****
                        Before Matching                 After Matching
mean treatment........        0.6                            0.6
mean control..........    0.68462                        0.60459
std mean diff.........    -17.225                       -0.93533

mean raw eQQ diff.....   0.081081                       0.075581
med  raw eQQ diff.....          0                              0
max  raw eQQ diff.....          1                              1

mean eCDF diff........   0.042308                       0.037791
med  eCDF diff........   0.042308                       0.037791
max  eCDF diff........   0.084615                       0.075581

var ratio (Tr/Co).....     1.1133                         1.0039
T-test p-value........   0.068031                        0.91711


***** (V15) I(re74 * re75) *****
                        Before Matching                 After Matching
mean treatment........   13118591                       13118591
mean control..........   14530303                       25001164
std mean diff.........    -2.7799                        -23.399

mean raw eQQ diff.....    3278733                        8171759
med  raw eQQ diff.....          0                              0
max  raw eQQ diff.....  188160151                      243080836
```

```
mean eCDF diff........   0.022723                    0.04676
med  eCDF diff........   0.014449                   0.046512
max  eCDF diff........   0.061019                    0.09593

var ratio (Tr/Co).....    0.69439                    0.33337
T-test p-value........    0.79058                    0.11452
KS Bootstrap p-value..      0.324                      0.005
KS Naive p-value......    0.81575                   0.084363
KS Statistic..........   0.061019                    0.09593


***** (V16) I(age * nodegr) *****
                        Before Matching                 After Matching
mean treatment........     17.968                      17.968
mean control..........     20.608                      19.591
std mean diff.........    -20.144                     -12.388

mean raw eQQ diff.....     2.7189                      1.3866
med  raw eQQ diff.....          1                           0
max  raw eQQ diff.....         18                          17

mean eCDF diff........   0.020386                    0.019732
med  eCDF diff........  0.0061331                    0.011628
max  eCDF diff........    0.12651                    0.072674

var ratio (Tr/Co).....     1.3301                      1.0752
T-test p-value........   0.027633                    0.069335
KS Bootstrap p-value..      0.031                       0.195
KS Naive p-value......   0.062873                     0.32369
KS Statistic..........    0.12651                    0.072674


***** (V17) I(educ * re74) *****
                        Before Matching                 After Matching
mean treatment........      22899                       22899
mean control..........      21067                       21812
std mean diff.........      3.191                      1.8935

mean raw eQQ diff.....     4775.1                      9105.7
med  raw eQQ diff.....          0                           0
max  raw eQQ diff.....     173996                      233352

mean eCDF diff........   0.018141                    0.057045
med  eCDF diff........   0.015281                    0.049419
max  eCDF diff........    0.04553                     0.11919
```

```
var ratio (Tr/Co).....     1.1152                        1.06
T-test p-value........    0.73471                     0.84458
KS Bootstrap p-value..      0.612                       0.001
KS Naive p-value......    0.97849                    0.015094
KS Statistic..........    0.04553                     0.11919


***** (V18) I(educ * re75) *****
                        Before Matching                 After Matching
mean treatment........      15881                         15881
mean control..........      12981                         21895
std mean diff.........     8.5349                       -17.702

mean raw eQQ diff.....     3760.4                        5727.7
med  raw eQQ diff.....          0                             0
max  raw eQQ diff.....      46244                         71480

mean eCDF diff........   0.050006                      0.051959
med  eCDF diff........   0.064293                      0.043605
max  eCDF diff........     0.1052                      0.098837

var ratio (Tr/Co).....     1.1901                       0.64031
T-test p-value........    0.35903                       0.10655
KS Bootstrap p-value..      0.062                         0.015
KS Naive p-value......    0.18269                      0.069435
KS Statistic..........     0.1052                      0.098837


Before Matching Minimum p.value: 0.0020368
Variable Name(s): nodegr  Number(s): 8

After Matching Minimum p.value: 0.001
Variable Name(s): re74 I(re74^2) I(educ * re74)  Number(s): 9 10 17
```

C. Dehejia and Wahba model full balance output

Attached is the full output from `MatchBalance` using one of Dehejia and Wahba's propensity score models:

```
R> dw.pscore <- glm(Tr ~ age + I(age^2) + educ + I(educ^2) + black + hisp +
+    married + nodegr + re74 + I(re74^2) + re75 + I(re75^2) + u74 + u75,
+    family = binomial, data = lalonde)
R> rr.dw <- Match(Y = Y, Tr = Tr, X = dw.pscore$fitted)
R> MatchBalance(Tr ~ age + I(age^2) + educ + I(educ^2) + black + hisp +
+    married + nodegr + re74 + I(re74^2) + re75 + I(re75^2) + u74 + u75 +
```

```
+    I(re74 * re75) + I(age * nodegr) + I(educ * re74) + I(educ * re75),
+    data = lalonde, match.out = rr.dw, nboots = 1000)

***** (V1) age *****
                        Before Matching                 After Matching
mean treatment........     25.816                       25.816
mean control..........     25.054                       25.006
std mean diff.........     10.655                       11.317

mean raw eQQ diff.....    0.94054                      0.41618
med  raw eQQ diff.....          1                            0
max  raw eQQ diff.....          7                            9

mean eCDF diff........   0.025364                     0.010597
med  eCDF diff........   0.022193                    0.0086705
max  eCDF diff........   0.065177                     0.049133

var ratio (Tr/Co).....     1.0278                       1.0662
T-test p-value........    0.26594                      0.23472
KS Bootstrap p-value..      0.523                        0.556
KS Naive p-value......     0.7481                      0.79781
KS Statistic..........   0.065177                     0.049133


***** (V2) I(age^2) *****
                        Before Matching                 After Matching
mean treatment........     717.39                       717.39
mean control..........     677.32                       673.08
std mean diff.........     9.2937                       10.275

mean raw eQQ diff.....     56.076                       28.948
med  raw eQQ diff.....         43                            0
max  raw eQQ diff.....        721                          909

mean eCDF diff........   0.025364                     0.010597
med  eCDF diff........   0.022193                    0.0086705
max  eCDF diff........   0.065177                     0.049133

var ratio (Tr/Co).....     1.0115                      0.91516
T-test p-value........    0.33337                      0.31819
KS Bootstrap p-value..      0.523                        0.556
KS Naive p-value......     0.7481                      0.79781
KS Statistic..........   0.065177                     0.049133


***** (V3) educ *****
                        Before Matching                 After Matching
```

```
mean treatment........     10.346                    10.346
mean control..........     10.088                     10.48
std mean diff.........     12.806                   -6.6749

mean raw eQQ diff.....    0.40541                   0.16185
med  raw eQQ diff.....          0                         0
max  raw eQQ diff.....          2                         2

mean eCDF diff........   0.028698                  0.011561
med  eCDF diff........   0.012682                 0.0086705
max  eCDF diff........    0.12651                  0.052023

var ratio (Tr/Co).....     1.5513                    1.1917
T-test p-value........    0.15017                   0.45021
KS Bootstrap p-value..      0.011                     0.307
KS Naive p-value......   0.062873                   0.73726
KS Statistic..........    0.12651                  0.052023


***** (V4) I(educ^2) *****
                        Before Matching                 After Matching
mean treatment........     111.06                    111.06
mean control..........     104.37                    113.21
std mean diff.........     17.012                   -5.466

mean raw eQQ diff.....     8.7189                    3.1098
med  raw eQQ diff.....          0                         0
max  raw eQQ diff.....         60                        60

mean eCDF diff........   0.028698                  0.011561
med  eCDF diff........   0.012682                 0.0086705
max  eCDF diff........    0.12651                  0.052023

var ratio (Tr/Co).....     1.6625                    1.2716
T-test p-value........   0.053676                   0.51046
KS Bootstrap p-value..      0.011                     0.307
KS Naive p-value......   0.062873                   0.73726
KS Statistic..........    0.12651                  0.052023


***** (V5) black *****
                        Before Matching                 After Matching
mean treatment........    0.84324                   0.84324
mean control..........    0.82692                   0.85946
std mean diff.........     4.4767                   -4.4482

mean raw eQQ diff.....   0.016216                  0.0086705
```

```
med  raw eQQ diff.....          0                        0
max  raw eQQ diff.....          1                        1

mean eCDF diff........  0.0081601                0.0043353
med  eCDF diff........  0.0081601                0.0043353
max  eCDF diff........    0.01632                0.0086705

var ratio (Tr/Co).....    0.92503                   1.0943
T-test p-value........    0.64736                  0.57783


***** (V6) hisp *****
                        Before Matching               After Matching
mean treatment........   0.059459                0.059459
mean control..........    0.10769                0.048649
std mean diff.........    -20.341                  4.5591

mean raw eQQ diff.....   0.048649                0.0057803
med  raw eQQ diff.....          0                        0
max  raw eQQ diff.....          1                        1

mean eCDF diff........   0.024116                0.0028902
med  eCDF diff........   0.024116                0.0028902
max  eCDF diff........   0.048233                0.0057803

var ratio (Tr/Co).....    0.58288                   1.2083
T-test p-value........   0.064043                  0.41443


***** (V7) married *****
                        Before Matching               After Matching
mean treatment........    0.18919                 0.18919
mean control..........    0.15385                 0.16667
std mean diff.........     8.9995                   5.735

mean raw eQQ diff.....   0.037838                0.017341
med  raw eQQ diff.....          0                        0
max  raw eQQ diff.....          1                        1

mean eCDF diff........   0.017672                0.0086705
med  eCDF diff........   0.017672                0.0086705
max  eCDF diff........   0.035343                 0.017341

var ratio (Tr/Co).....     1.1802                   1.1045
T-test p-value........    0.33425                  0.46741
```

```
***** (V8) nodegr *****
                        Before Matching                 After Matching
mean treatment........    0.70811                       0.70811
mean control..........    0.83462                       0.69189
std mean diff.........    -27.751                        3.5572

mean raw eQQ diff.....    0.12432                      0.014451
med  raw eQQ diff.....          0                             0
max  raw eQQ diff.....          1                             1

mean eCDF diff........   0.063254                     0.0072254
med  eCDF diff........   0.063254                     0.0072254
max  eCDF diff........    0.12651                      0.014451

var ratio (Tr/Co).....     1.4998                       0.96957
T-test p-value........  0.0020368                       0.49161


***** (V9) re74 *****
                        Before Matching                 After Matching
mean treatment........     2095.6                        2095.6
mean control..........       2107                        1624.3
std mean diff.........   -0.23437                        9.6439

mean raw eQQ diff.....     487.98                        467.33
med  raw eQQ diff.....          0                             0
max  raw eQQ diff.....       8413                         12410

mean eCDF diff........   0.019223                      0.019782
med  eCDF diff........     0.0158                      0.018786
max  eCDF diff........   0.047089                      0.046243

var ratio (Tr/Co).....     0.7381                        2.2663
T-test p-value........    0.98186                       0.22745
KS Bootstrap p-value..      0.549                         0.263
KS Naive p-value......    0.97023                        0.8532
KS Statistic..........   0.047089                      0.046243


***** (V10) I(re74^2) *****
                        Before Matching                 After Matching
mean treatment........   28141434                      28141434
mean control..........   36667413                      13117852
std mean diff.........    -7.4721                         13.167

mean raw eQQ diff.....   13311731                      10899373
med  raw eQQ diff.....          0                             0
```

```
max  raw eQQ diff.....  365146387                  616156569

mean eCDF diff........   0.019223                   0.019782
med  eCDF diff........     0.0158                   0.018786
max  eCDF diff........   0.047089                   0.046243

var ratio (Tr/Co).....    0.50382                    7.9006
T-test p-value........    0.51322                   0.08604
KS Bootstrap p-value..      0.549                     0.263
KS Naive p-value......    0.97023                    0.8532
KS Statistic..........   0.047089                   0.046243


***** (V11) re75 *****
                      Before Matching                After Matching
mean treatment........     1532.1                     1532.1
mean control..........     1266.9                     1297.6
std mean diff.........     8.2363                     7.2827

mean raw eQQ diff.....     367.61                     211.42
med  raw eQQ diff.....          0                          0
max  raw eQQ diff.....     2110.2                     8195.6

mean eCDF diff........   0.050834                   0.023047
med  eCDF diff........   0.061954                   0.023121
max  eCDF diff........    0.10748                   0.057803

var ratio (Tr/Co).....     1.0763                     1.4291
T-test p-value........    0.38527                   0.33324
KS Bootstrap p-value..      0.039                      0.152
KS Naive p-value......    0.16449                    0.60988
KS Statistic..........    0.10748                   0.057803


***** (V12) I(re75^2) *****
                      Before Matching                After Matching
mean treatment........   12654753                   12654753
mean control..........   11196530                    8896263
std mean diff.........     2.6024                     6.7076

mean raw eQQ diff.....    2840830                    2887443
med  raw eQQ diff.....          0                          0
max  raw eQQ diff.....  101657197                  344942969

mean eCDF diff........   0.050834                   0.023047
med  eCDF diff........   0.061954                   0.023121
max  eCDF diff........    0.10748                   0.057803
```

```
var ratio (Tr/Co).....     1.4609                    3.559
T-test p-value........    0.77178                  0.37741
KS Bootstrap p-value..      0.039                    0.152
KS Naive p-value......    0.16449                  0.60988
KS Statistic..........    0.10748                 0.057803


***** (V13) u74 *****
                        Before Matching                After Matching
mean treatment........    0.70811                  0.70811
mean control..........       0.75                  0.68458
std mean diff.........    -9.1895                   5.1608

mean raw eQQ diff.....   0.037838                 0.017341
med  raw eQQ diff.....          0                        0
max  raw eQQ diff.....          1                        1

mean eCDF diff........   0.020946                0.0086705
med  eCDF diff........   0.020946                0.0086705
max  eCDF diff........   0.041892                 0.017341

var ratio (Tr/Co).....     1.1041                  0.95721
T-test p-value........    0.33033                  0.52298


***** (V14) u75 *****
                        Before Matching                After Matching
mean treatment........        0.6                      0.6
mean control..........    0.68462                  0.62072
std mean diff.........    -17.225                  -4.2182

mean raw eQQ diff.....   0.081081                 0.031792
med  raw eQQ diff.....          0                        0
max  raw eQQ diff.....          1                        1

mean eCDF diff........   0.042308                 0.015896
med  eCDF diff........   0.042308                 0.015896
max  eCDF diff........   0.084615                 0.031792

var ratio (Tr/Co).....     1.1133                   1.0194
T-test p-value........   0.068031                  0.46507


***** (V15) I(re74 * re75) *****
                        Before Matching                After Matching
mean treatment........   13118591                 13118591
```

```
mean control.........   14530303                    8958064
std mean diff........    -2.7799                     8.1928

mean raw eQQ diff.....    3278733                    3085879
med  raw eQQ diff.....          0                          0
max  raw eQQ diff.....  188160151                  211819713

mean eCDF diff........   0.022723                   0.014519
med  eCDF diff........   0.014449                   0.014451
max  eCDF diff........   0.061019                   0.037572

var ratio (Tr/Co).....    0.69439                    2.7882
T-test p-value........    0.79058                    0.30299
KS Bootstrap p-value..      0.275                      0.386
KS Naive p-value......    0.81575                    0.96754
KS Statistic..........   0.061019                   0.037572


***** (V16) I(age * nodegr) *****
                       Before Matching                After Matching
mean treatment........     17.968                     17.968
mean control.........     20.608                     17.294
std mean diff........    -20.144                     5.1366

mean raw eQQ diff.....     2.7189                    0.60405
med  raw eQQ diff.....          1                          0
max  raw eQQ diff.....         18                         17

mean eCDF diff........   0.020386                  0.0090105
med  eCDF diff........  0.0061331                  0.0072254
max  eCDF diff........    0.12651                   0.037572

var ratio (Tr/Co).....     1.3301                    0.98044
T-test p-value........   0.027633                   0.48453
KS Bootstrap p-value..       0.03                      0.819
KS Naive p-value......   0.062873                   0.96754
KS Statistic..........    0.12651                   0.037572


***** (V17) I(educ * re74) *****
                       Before Matching                After Matching
mean treatment........      22899                      22899
mean control.........      21067                      17069
std mean diff........      3.191                      10.157

mean raw eQQ diff.....     4775.1                     5443.8
med  raw eQQ diff.....          0                          0
```

```
max  raw eQQ diff.....     173996                      267977

mean eCDF diff........   0.018141                    0.016409
med  eCDF diff........   0.015281                    0.014451
max  eCDF diff........    0.04553                    0.049133

var ratio (Tr/Co).....     1.1152                      2.9191
T-test p-value........    0.73471                     0.18059
KS Bootstrap p-value..      0.585                       0.219
KS Naive p-value......    0.97849                     0.79781
KS Statistic..........    0.04553                    0.049133


***** (V18) I(educ * re75) *****
                        Before Matching                 After Matching
mean treatment........      15881                       15881
mean control..........      12981                       13051
std mean diff.........     8.5349                      8.3267

mean raw eQQ diff.....     3760.4                      2235.4
med  raw eQQ diff.....          0                           0
max  raw eQQ diff.....      46244                      124045

mean eCDF diff........   0.050006                    0.022441
med  eCDF diff........   0.064293                    0.020231
max  eCDF diff........     0.1052                    0.057803

var ratio (Tr/Co).....     1.1901                      1.6746
T-test p-value........    0.35903                     0.25369
KS Bootstrap p-value..      0.042                       0.157
KS Naive p-value......    0.18269                     0.60988
KS Statistic..........     0.1052                    0.057803


Before Matching Minimum p.value: 0.0020368
Variable Name(s): nodegr  Number(s): 8

After Matching Minimum p.value: 0.08604
Variable Name(s): I(re74^2)  Number(s): 10
```

D. Genetic matching full balance output

Attached is the full `MatchBalance` output from genetic matching:

```
R> X <- cbind(age, educ, black, hisp, married, nodegr, re74, re75, u74, u75)
R> BalanceMatrix <- cbind(age, I(age^2), educ, I(educ^2), black, hisp,
```

```
+    married, nodegr, re74, I(re74^2), re75, I(re75^2), u74, u75,
+    I(re74 * re75), I(age * nodegr), I(educ * re74), I(educ * re75))
R> gen1 <- GenMatch(Tr = Tr, X = X, BalanceMatrix = BalanceMatrix,
+    pop.size = 1000)

R> mgen1 <- Match(Y = Y, Tr = Tr, X = X, Weight.matrix = gen1)
R> MatchBalance(Tr ~ age + I(age^2) + educ + I(educ^2) + black + hisp +
+    married + nodegr + re74 + I(re74^2) + re75 + I(re75^2) + u74 + u75 +
+    I(re74 * re75) + I(age * nodegr) + I(educ * re74) + I(educ * re75),
+    data = lalonde, match.out = mgen1, nboots = 1000)

***** (V1) age *****
                        Before Matching                 After Matching
mean treatment........     25.816                       25.816
mean control.........      25.054                       25.648
std mean diff.........     10.655                       2.3545

mean raw eQQ diff.....    0.94054                      0.45149
med  raw eQQ diff.....          1                            0
max  raw eQQ diff.....          7                            9

mean eCDF diff........   0.025364                     0.012638
med  eCDF diff........   0.022193                     0.011194
max  eCDF diff........   0.065177                     0.037313

var ratio (Tr/Co).....     1.0278                       1.0303
T-test p-value........    0.26594                      0.47904
KS Bootstrap p-value..      0.522                        0.931
KS Naive p-value......     0.7481                       0.9922
KS Statistic..........   0.065177                     0.037313


***** (V2) I(age^2) *****
                        Before Matching                 After Matching
mean treatment........     717.39                       717.39
mean control.........      677.32                       707.23
std mean diff.........     9.2937                       2.3581

mean raw eQQ diff.....     56.076                       30.422
med  raw eQQ diff.....         43                            0
max  raw eQQ diff.....        721                          909

mean eCDF diff........   0.025364                     0.012638
med  eCDF diff........   0.022193                     0.011194
max  eCDF diff........   0.065177                     0.037313

var ratio (Tr/Co).....     1.0115                      0.97264
```

```
T-test p-value........    0.33337                    0.51616
KS Bootstrap p-value..      0.522                      0.931
KS Naive p-value......     0.7481                     0.9922
KS Statistic..........   0.065177                   0.037313


***** (V3) educ *****
                        Before Matching                After Matching
mean treatment........     10.346                     10.346
mean control..........     10.088                     10.351
std mean diff.........     12.806                    -0.26884

mean raw eQQ diff.....    0.40541                   0.078358
med  raw eQQ diff.....          0                          0
max  raw eQQ diff.....          2                          2

mean eCDF diff........   0.028698                   0.005597
med  eCDF diff........   0.012682                   0.005597
max  eCDF diff........    0.12651                   0.014925

var ratio (Tr/Co).....     1.5513                     1.1302
T-test p-value........    0.15017                    0.86959
KS Bootstrap p-value..      0.016                      0.999
KS Naive p-value......   0.062873                          1
KS Statistic..........    0.12651                   0.014925


***** (V4) I(educ^2) *****
                        Before Matching                After Matching
mean treatment........     111.06                     111.06
mean control..........     104.37                     110.71
std mean diff.........     17.012                    0.89394

mean raw eQQ diff.....     8.7189                     1.5187
med  raw eQQ diff.....          0                          0
max  raw eQQ diff.....         60                         60

mean eCDF diff........   0.028698                   0.005597
med  eCDF diff........   0.012682                   0.005597
max  eCDF diff........    0.12651                   0.014925

var ratio (Tr/Co).....     1.6625                     1.1964
T-test p-value........   0.053676                    0.61968
KS Bootstrap p-value..      0.016                      0.999
KS Naive p-value......   0.062873                          1
KS Statistic..........    0.12651                   0.014925
```

```
***** (V5) black *****
                       Before Matching                 After Matching
mean treatment........    0.84324                      0.84324
mean control..........    0.82692                      0.85405
std mean diff.........     4.4767                      -2.9655

mean raw eQQ diff.....   0.016216                     0.0074627
med  raw eQQ diff.....          0                             0
max  raw eQQ diff.....          1                             1

mean eCDF diff........  0.0081601                     0.0037313
med  eCDF diff........  0.0081601                     0.0037313
max  eCDF diff........    0.01632                     0.0074627

var ratio (Tr/Co).....    0.92503                       1.0605
T-test p-value........    0.64736                       0.4798


***** (V6) hisp *****
                       Before Matching                 After Matching
mean treatment........   0.059459                      0.059459
mean control..........    0.10769                      0.054054
std mean diff.........    -20.341                        2.2796

mean raw eQQ diff.....   0.048649                     0.0037313
med  raw eQQ diff.....          0                             0
max  raw eQQ diff.....          1                             1

mean eCDF diff........   0.024116                     0.0018657
med  eCDF diff........   0.024116                     0.0018657
max  eCDF diff........   0.048233                     0.0037313

var ratio (Tr/Co).....    0.58288                       1.0937
T-test p-value........   0.064043                      0.65507


***** (V7) married *****
                       Before Matching                 After Matching
mean treatment........    0.18919                      0.18919
mean control..........    0.15385                      0.17838
std mean diff.........     8.9995                       2.7528

mean raw eQQ diff.....   0.037838                     0.011194
med  raw eQQ diff.....          0                             0
max  raw eQQ diff.....          1                             1
```

```
mean eCDF diff........   0.017672                    0.005597
med  eCDF diff........   0.017672                    0.005597
max  eCDF diff........   0.035343                    0.011194

var ratio (Tr/Co).....     1.1802                      1.0467
T-test p-value........    0.33425                      0.7459


***** (V8) nodegr *****
                       Before Matching                 After Matching
mean treatment........    0.70811                     0.70811
mean control..........    0.83462                     0.70811
std mean diff.........    -27.751                           0

mean raw eQQ diff.....    0.12432                           0
med  raw eQQ diff.....          0                           0
max  raw eQQ diff.....          1                           0

mean eCDF diff........   0.063254                           0
med  eCDF diff........   0.063254                           0
max  eCDF diff........    0.12651                           0

var ratio (Tr/Co).....     1.4998                           1
T-test p-value........  0.0020368                           1


***** (V9) re74 *****
                       Before Matching                 After Matching
mean treatment........     2095.6                      2095.6
mean control..........       2107                      2017.2
std mean diff.........   -0.23437                      1.6031

mean raw eQQ diff.....     487.98                         174
med  raw eQQ diff.....          0                           0
max  raw eQQ diff.....       8413                      7175.7

mean eCDF diff........   0.019223                   0.0066891
med  eCDF diff........     0.0158                   0.0037313
max  eCDF diff........   0.047089                    0.029851

var ratio (Tr/Co).....     0.7381                      1.1519
T-test p-value........    0.98186                     0.51757
KS Bootstrap p-value..      0.587                       0.805
KS Naive p-value......    0.97023                     0.99976
KS Statistic..........   0.047089                    0.029851
```

```
***** (V10) I(re74^2) *****
                        Before Matching                 After Matching
mean treatment........   28141434                       28141434
mean control..........   36667413                       24686484
std mean diff.........    -7.4721                         3.0279

mean raw eQQ diff.....   13311731                        4823772
med  raw eQQ diff.....          0                              0
max  raw eQQ diff.....  365146387                      451383821

mean eCDF diff........   0.019223                       0.0066891
med  eCDF diff........     0.0158                       0.0037313
max  eCDF diff........   0.047089                        0.029851

var ratio (Tr/Co).....    0.50382                         1.5233
T-test p-value........    0.51322                         0.4652
KS Bootstrap p-value..      0.587                          0.805
KS Naive p-value......    0.97023                        0.99976
KS Statistic..........   0.047089                        0.029851


***** (V11) re75 *****
                        Before Matching                 After Matching
mean treatment........     1532.1                         1532.1
mean control..........     1266.9                           1483
std mean diff.........     8.2363                         1.5234

mean raw eQQ diff.....     367.61                         167.61
med  raw eQQ diff.....          0                              0
max  raw eQQ diff.....     2110.2                         2973.4

mean eCDF diff........   0.050834                       0.016297
med  eCDF diff........   0.061954                       0.014925
max  eCDF diff........    0.10748                       0.044776

var ratio (Tr/Co).....     1.0763                         1.0405
T-test p-value........    0.38527                        0.63214
KS Bootstrap p-value..      0.043                          0.563
KS Naive p-value......    0.16449                          0.951
KS Statistic..........    0.10748                       0.044776


***** (V12) I(re75^2) *****
                        Before Matching                 After Matching
mean treatment........   12654753                       12654753
mean control..........   11196530                       12106082
std mean diff.........     2.6024                        0.97918
```

```
mean raw eQQ diff.....    2840830                    2124586
med  raw eQQ diff.....          0                          0
max  raw eQQ diff.....  101657197                  101657197

mean eCDF diff........   0.050834                   0.016297
med  eCDF diff........   0.061954                   0.014925
max  eCDF diff........    0.10748                   0.044776

var ratio (Tr/Co).....     1.4609                     1.3641
T-test p-value........    0.77178                    0.69082
KS Bootstrap p-value..      0.043                      0.563
KS Naive p-value......    0.16449                      0.951
KS Statistic..........    0.10748                   0.044776


***** (V13) u74 *****
                       Before Matching                After Matching
mean treatment........    0.70811                    0.70811
mean control..........       0.75                    0.70811
std mean diff.........    -9.1895                          0

mean raw eQQ diff.....   0.037838                          0
med  raw eQQ diff.....          0                          0
max  raw eQQ diff.....          1                          0

mean eCDF diff........   0.020946                          0
med  eCDF diff........   0.020946                          0
max  eCDF diff........   0.041892                          0

var ratio (Tr/Co).....     1.1041                          1
T-test p-value........    0.33033                          1


***** (V14) u75 *****
                       Before Matching                After Matching
mean treatment........        0.6                        0.6
mean control..........    0.68462                    0.61622
std mean diff.........    -17.225                    -3.3012

mean raw eQQ diff.....   0.081081                   0.018657
med  raw eQQ diff.....          0                          0
max  raw eQQ diff.....          1                          1

mean eCDF diff........   0.042308                  0.0093284
med  eCDF diff........   0.042308                  0.0093284
max  eCDF diff........   0.084615                   0.018657
```

```
var ratio (Tr/Co).....     1.1133                    1.0148
T-test p-value........   0.068031                   0.46714


***** (V15) I(re74 * re75) *****
                        Before Matching                 After Matching
mean treatment........   13118591                  13118591
mean control..........   14530303                  12984123
std mean diff.........    -2.7799                   0.26479

mean raw eQQ diff.....    3278733                   2854107
med  raw eQQ diff.....          0                         0
max  raw eQQ diff.....  188160151                 188160151

mean eCDF diff........   0.022723                  0.0078223
med  eCDF diff........   0.014449                  0.0074627
max  eCDF diff........   0.061019                   0.026119

var ratio (Tr/Co).....    0.69439                   0.91092
T-test p-value........    0.79058                   0.95773
KS Bootstrap p-value..      0.292                     0.872
KS Naive p-value......    0.81575                   0.99999
KS Statistic..........   0.061019                   0.026119


***** (V16) I(age * nodegr) *****
                        Before Matching                 After Matching
mean treatment........     17.968                    17.968
mean control..........     20.608                    17.864
std mean diff.........    -20.144                   0.79047

mean raw eQQ diff.....     2.7189                   0.27985
med  raw eQQ diff.....          1                         0
max  raw eQQ diff.....         18                         9

mean eCDF diff........   0.020386                  0.0073423
med  eCDF diff........  0.0061331                  0.0037313
max  eCDF diff........    0.12651                   0.041045

var ratio (Tr/Co).....     1.3301                   0.98897
T-test p-value........   0.027633                   0.60664
KS Bootstrap p-value..      0.041                     0.828
KS Naive p-value......   0.062873                   0.97767
KS Statistic..........    0.12651                   0.041045
```

```
***** (V17) I(educ * re74) *****
                        Before Matching                 After Matching
mean treatment........      22899                          22899
mean control..........      21067                          21615
std mean diff.........      3.191                          2.236

mean raw eQQ diff.....     4775.1                         2863.8
med  raw eQQ diff.....          0                              0
max  raw eQQ diff.....     173996                         211917

mean eCDF diff........   0.018141                       0.0075061
med  eCDF diff........   0.015281                       0.0074627
max  eCDF diff........    0.04553                        0.026119

var ratio (Tr/Co).....     1.1152                         1.3621
T-test p-value........    0.73471                        0.40827
KS Bootstrap p-value..      0.632                          0.876
KS Naive p-value......    0.97849                        0.99999
KS Statistic..........    0.04553                       0.026119


***** (V18) I(educ * re75) *****
                        Before Matching                 After Matching
mean treatment........      15881                          15881
mean control..........      12981                          15521
std mean diff.........     8.5349                         1.0583

mean raw eQQ diff.....     3760.4                         1949.2
med  raw eQQ diff.....          0                              0
max  raw eQQ diff.....      46244                          46244

mean eCDF diff........   0.050006                        0.01618
med  eCDF diff........   0.064293                       0.014925
max  eCDF diff........     0.1052                       0.041045

var ratio (Tr/Co).....     1.1901                         1.0948
T-test p-value........    0.35903                        0.76596
KS Bootstrap p-value..      0.052                          0.652
KS Naive p-value......    0.18269                        0.97767
KS Statistic..........     0.1052                       0.041045


Before Matching Minimum p.value: 0.0020368
Variable Name(s): nodegr  Number(s): 8

After Matching Minimum p.value: 0.40827
Variable Name(s): I(educ * re74)  Number(s): 17
```

Affiliation:

Jasjeet S. Sekhon
Department of Political Science
210 Barrows Hall #1950
UC Berkeley
Berkeley, CA 94720-1950, United States of America
E-mail: sekhon@berkeley.edu
URL: http://sekhon.berkeley.edu

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 42, Issue 7 *Submitted:* 2007-02-07
June 2011 *Accepted:* 2007-06-18