An overview of `rbounds`: An `R` package for Rosenbaum bounds sensitivity analysis with matched data.

Luke Keele

August 20, 2010

1 Introduction

Matching has, in recent years, become the tool of choice for estimating treatment effects in observational data. Estimators for treatment effects based on matching have several advantages over regression based estimators. Matching greatly reduces dependency on functional form and allows for greater transparency in the modeling process. All matching estimators, however, do retain the strong assumption that observable covariates account for the selection process into the treatment and control conditions. Estimates of treatment effects based on matching are unbiased *if* there are no unobserved confounders and if all relevant covariates have been included in the matching model. Therefore, a common concern is that our adjustments via matching fail to account for some relevant covariate that was not measured.

One can, however, conduct a sensitivity analysis for the matching estimate. A sensitivity analysis is designed to provide a quantifiable increase in uncertainty when a key assumption is relaxed. One advantage of matching is that a transparent method exists for conducting sensitivity analyses. Rosenbaum (2002) has developed a method of sensitivity analysis to assess if one's estimated based on matching is robust to the possible presence of an unobserved confounder, the key assumption for matching based analyses. His sensitivity analysis for matched data provides a specific statement about the magnitude of hidden bias that would need to be present to explain the associations actually observed (Rosenbaum 2005). His method relies on the principle of randomization inference. In what follows, I provide brief overview of randomization inference with a focus on the Wilcoxon signed rank test. I then outline Rosenbaum's method of sensitivity analysis followed by a demonstration in `R` using the `rbounds` package.

2 Randomization Inference

Fisher developed the concept of randomization inference in 1935. Fisher considered randomization tests to be the ideal tests for experiments as he saw the random assignment of a treatment or treatments as the "reasoned basis for inference" (Fisher 1935). The quote refers to the fact that we can use randomization as our basis for the statistical test. These tests use permutations of the data based on the fact that we have randomized the treatment or treatments to conduct tests without sampling distributions. As with any statistical test of hypotheses, we need the following elements.

- data
- null hypothesis
- test statistic
- the distribution of the test statistic under the null hypothesis

I'll start with discussing these elements for the simplest test. The data will be comprised of a group of subjects where some form a control group and the rest form a single treatment group. Next is the null hypothesis, which is typically along the lines of the treatment has no effect. More formally, we could define this in the following way. First, the treatment effect is:

$$\Delta = L(T) - L(C)$$

where L is some measure of location. The usual null hypothesis is

$$H_0 : \Delta = 0.$$

This brings us to the test statistic. There are many possible test statistics, but here I focus on the test statistic from the Wilcoxon signed rank test. The signed rank test is the nonparametric alternative to the paired t-test. It is designed to assess comparisons in paired data, which clearly applies to matched data. The signed rank test statistic is calculated by taking the differences of the paired data. The absolute value of these differences are ranked. The test statistic is the sum of the ranks for the positive differences. The reason ranks are used is that they are resistant to outliers and as such the test is more robust than the usual parametric alternatives.

Finally, we need a distribution to compare this statistic with. This would be a distribution that holds if the null hypothesis were true. For standard parametric tests, this would be the t-distribution. Here we are going to exploit the fact that we have randomized the treatment and use permutations to form a null distribution. For permutation tests, permutation refers to any specified class of rearrangements or modifications of the data. The null hypothesis of the test specifies that permutations of the treatment and control group are all equally likely. Since we randomize the treatment, a subject is just as likely to be in the treatment group as in the control group. A another way to say this is that the distribution of the data under the null hypothesis satisfies exchangeability. Randomization makes the data exchangeable. That is, if the null hypothesis is true, then the estimated treatment effect will be the same even if we rearrange the subjects across the treatment and control groups. The distribution of the test statistic under the null hypothesis is computed by forming all of the permutations of the treatment group and calculating a test statistic for each of these permutations.

To calculate all the possible permutations of the subjects into control and treatment groups we use the binomial coefficient:

$$\begin{pmatrix} N \\ \\ n \end{pmatrix} =_N C_n = \frac{N!}{(N-n)!}/n! = \frac{N!}{(N-n)!n!}.$$

Our *exact* p-value is simply

$$p = \frac{\#(W)}{\begin{pmatrix} N \\ \\ n \end{pmatrix}}.$$

The numerator is simply the number of times the observed statistic occurs among all the possible permutations. Since all outcomes are equally likely, we can divide this sum by the total number of permutations.

Tests of this type are best understood with an example. In this example, I use a subset of the data from Rosenbaum (2002). In this data, children whose parents worked in a factory where they were exposed to lead were matched to children whose parents did not. The outcome is the level of blood in the children whose parents were exposed to lead. In the data, there are 33 treatment observations and 33 control observations. For computational ease, I use a subset of five of the treated observations and five of the control observations. The following R code calculates the Wilcoxon signed rank statistic based on the data which is printed below:

```
> y <- c(38,23,41,37,36)
> x <- c(16,18,24,19,11)
>
> t.1 <- c(1,1,1,1,1)
> t.0 <- c(0,0,0,0,0)
>
> #Observed Test Statistic
> tau.0 <- 0 #Difference in Groups Under the Null Hypothesis
> z <- (t.1 - t.0)*((y- tau.0)-x)
> sgn <- as.numeric((z>0))
> n <- length(y)
> r <- rank(abs(z))
> T <- sum(r*sgn)
> T
[1] 15
```

The calculation of the signed rank statistic is fairly simple. I obtain the differences between treated and control, and sum the ranks of the absolute differences when those differences are positive. This results in a test statistic of 15. I next derive the null distribution for the test. To derive that distribution, I simply take all the permutations of the data and apply the signed rank test statistic to each one. There are 32 ways to distribution the treatment within the five matched pairs. To form the null distribution, we take all 32 permutations and calculate our test statistic 32 times. Some of these permutations will produce a rank sum that will be identical to how the randomization actually turned out, many will not. To test our hypothesis, we simply need to sum how many times we find a signed rank statistic of 15 or greater out of the 32 possible outcomes. If 15 occurs rarely this tells us that it is unlikely that the treatment effect occurred due to chance. The following R code performs these operations. I also compare this code to the R function wilcox.exact, which calculates the signed rank test. Note that the exactRankTests library must be loaded in R to use this function.

```
> #Permuation of Treatments
> par.mat <- c()
>
>         for(y00 in 0:1){
+             for(y01 in 0:1){
+                  for(y10 in 0:1){
+                  for(y11 in 0:1){
+                  for(y22 in 0:1){
+                    par.mat <-  c(par.mat, c(y00, y01, y10, y11, y22))
+
+ }}}}}
>
> z.0 <- matrix(par.mat, 32, 5, byrow=TRUE)
> z.1 <- 1 - z.0
>
> #Permutation Distribution of Ranks
> p <- nrow(z.1)
> tau.0 <- 0
> T.star <- matrix(NA, 32, 1)
> for(i in 1:p){
+ z <- (z.1[i,] - z.0[i,])*((y - tau.0) - x)
+ r <- rank(abs(z))
+ sgn <- as.numeric((z>0))
+ T.star[i,] <- sum(r*sgn)
+ }
>
> #Exact p-value for sharp null
> (sum(T <= T.star))/p
[1] 0.03125
>
> wilcox.exact(y, x, paired=TRUE, exact=TRUE, alternative="greater")

Exact Wilcoxon signed rank test

data:  y and x
V = 15, p-value = 0.03125
alternative hypothesis: true mu is greater than 0
```

The result is an exact *p*-value that tells us how likely we would observe this treated outcome due to chance. If this exact *p*-value is less than the usual 0.05 threshold, we would reject the null of no treatment effect. The `wilcox.exact` function also calculates the signed rank test along with the Hodges-Lehmann estimate of the additive treatment effect and confidence intervals for that treatment effect. Here, I use the complete data from Rosenbaum (2002).

```
> #Data:  Matched Data of Lead Blood Levels in Children
> trt <- c(38,23,41,18,37,36,23,62,31,34,24,14,21,17,16,20,15,10,45,39,22,35,49,48,
44,35,43,39,34,13,73,25,27)
> ctrl <- c(16,18,18,24,19,11,10,15,16,18,18,13,19,10,16,16,24,13,9,14,21,19,
7,18,19,12,11,22, 25,16,13,11,13)
>
> wilcox.exact(trt, ctrl, paired=TRUE, conf.int=TRUE, exact=TRUE)

Exact Wilcoxon signed rank test

data:  trt and ctrl
```

```
V = 499, p-value = 8.037e-07
alternative hypothesis: true mu is not equal to 0
95 percent confidence interval:
  9.5 21.5
sample estimates:
(pseudo)median
         15.25
```

We see that the difference is highly significant with lead levels being a little over 15 points higher among the children in the treatment group. I now turn to sensitivity analyses for matched data which are built on randomization inference.

3 Rosenbaum Sensitivity Analyses

Randomization inference is not generally valid with observational data. Without randomization of a treatment, we cannot assume the data points are exchangeable due to the probability of treatment being equal across the treated and control groups. As such we cannot permute the data to form the exact p-value. With matched data, however, we might say that this p-value is valid *if* there are no unobserved confounders. That is, if we have correctly matched the data, there should be no differences between the treated and control groups. That is if we have correctly matched on all the covariates that causes difference in the distribution of the treated and control groups, then the probability of treatment will be constant within those matched pairs just as if we had randomly assigned treatment within those pairs. If this is true randomization inference is valid with observational data. This is the basis on which Rosenbaum develops sensitivity tests for matched data (Rosenbaum 2002). First, I provide an outline of his model for an unobserved confounder.

Rosenbaum's method of sensitivity analysis relies on the sensitivity parameter Γ that measures the degree of departure from random assignment of treatment. Two subjects with the same observed characteristics may differ in the odds of receiving the treatment by at most a factor of Γ. In a randomized experiment, randomization of the treatment ensures that $\Gamma = 1$. In an observational study, if $\Gamma = 2$, and two subjects are identical on matched covariates then one might be twice as likely as the other to receive the treatment because they differ in terms of an unoberved covariate(Rosenbaum 2005). While values of Γ are unknown, we can try several values of Γ and see if the conclusions of the study change.

If after we have matched, the result is free of hidden bias from an unobserved confounder then the probability, π_j that unit j receives the treatment is only a function of the covariates $\mathbf{x}_j$ that describe unit j. There is hidden bias if two units with the same values on $\mathbf{x}$ have differing chances of receiving the treatment. More formally, we have hidden bias if $\mathbf{x}_j = \mathbf{x}_k$ but $\pi_j \neq \pi_k$ for some units j and k. In a sensitivity analysis, we ask how large the differences in π would need to be to change our basic inference. We do this through the use of the odds ratio. If π_j is the probability of treatment for unit j, then the odds that unit j receives the treatment is $\pi_j/(1-\pi_j)$. With the same being true for unit k. Suppose that we knew the odds ratio of units with the same values of $\mathbf{x}$ was at most:

$$\frac{1}{\Gamma} \leq \frac{\pi_j/(1-\pi_j)}{\pi_k/(1-\pi_k)} \leq \Gamma \tag{1}$$

for all j and k with $\mathbf{x}_j = \mathbf{x}_k$. If the value for Γ is one then then this implies that the odds ratio of treatment is the same and the study is free of hidden bias. If $\Gamma = 2$ then two units that have the same values of $\mathbf{x}$ could differ in their odds of receiving treatment by as much as as a factor of 2. Therefore one unit would be twice as likely to receive the treatment as the other unit. In a sensitivity analysis, we use Γ as a measure of the degree of departure from a study that is free of bias. One uses several different values of Γ to show how inferences might change if hidden bias were present.

Another way to think about this is in terms of an unobserved covariate. Characteristics that are not in $\mathbf{x}$ were not controlled for when we matched. We might call this unobserved covariate u. We can rewrite the model above in terms of this unobserved covariate. Let's say that unit j has observed covariates $\mathbf{x}$ and an unobserved covariate u_j, which nearly perfectly predicts the outcome. We can write a logistic regression model linking the odds of assignment to these two covariates:

$$\log\left(\frac{\pi_j}{(1-\pi_j)}\right) = \kappa(\mathbf{x}) + \gamma u_j \tag{2}$$

with a constraint on u_j of $0 \leq u_j \leq 1$. Here κ is an unknown function and γ is an unobserved parameter. If units j and k have the same values on $\mathbf{x}$ then $\mathbf{x}_j = \mathbf{x}_k$ and we can write the odds ratio of treatment for these two units as:

$$\frac{\pi_j/(1-\pi_j)}{\pi_k/(1-\pi_k)} = \exp\{\gamma(u_j - u + k)\}$$

Here two units with the same $\mathbf{x}$ values differ in their odds of treatment by a factor of γ and the difference in the unobserved covariate. Rosenbaum (2002) proves that the two statements of unobserved bias are the same. Therefore we can think if Γ as the size of log of the coefficient for the unobserved covariate u. The larger it is the more likely our inference will change due to the magnitude of the hidden bias.

The basic process for a sensitivity analysis is as follows. First one selects a series of values for Γ. One might, for example, use values of 1 to 6. I should note that a Γ value of 6 is very large and most findings in the social sciences are not robust hidden biases of this magnitude. Often one might need to use values between 1 and 2. These values are then used to adjust the finding. We can do two things. One we can do a sensitivity analysis on the p-values and see how the p-value increases for increasing values of Γ. We can also see how the magnitude of the treatment effect changes with an increasing Γ. Each sensitivity test is built on a specific randomization test for a type of outcome.

Rosenbaum (2002) has altered several different randomization tests to provide bounds based on the magnitude of Γ. For a binary outcomes, the sensitivity analysis is based on McNemar's test. For other outcomes, the sensitivity test is based on the Wilcoxon sign rank test and the Hodges-Lehmann point estimate for the sign rank test. I provide an example for binary data from Rosenbaum (2002) for a matched analysis of smoking. McNemar's statistic is T. It is based on a 2×2 cross-tab. In this example, the treatment indicator is whether a person was a smoker or not. The outcome is whether the person died of lung cancer or not. There are a 122 treated subjects in this study and 110 experience mortality due to lung cancer. McNemar's test statistic is simply the difference between the total number treated and the number of successes in the treated category. This statistic has a χ^2 distribution with a p-value associated with it. In this case it is highly significant at $p < .001$. This p-value is valid if the study is free from hidden bias from an unobserved confounder.

How will this p-value change in the presence of bias due to an unobserved confounder. That is, would we make the same inference if the odds of an unobserved confounder increase? To do this we need to calculate T^+ and T^-. These our the upper and lower bounds on the McNemar's test statistic based on values of Γ. We define $p^+ = \Gamma/(1+\Gamma)$ and $p^- = 1/(1+\Gamma)$. Using this we can calculate the upper and lower bounds on the p-value. The upper bound is:

$$\sum_{a=110}^{122} \binom{122}{a} (p^+)^a (1-p^+)^{122-a} \tag{3}$$

There is a similar lower bound but the lower bound is not that interesting since it is always lower than the observed p-value. If we set Γ to four and we plug that value into the formula above, we find that the upper bound on the p-value is 0.0019. For $\Gamma = 4$, one person in the matched pair may be four times more likely to smoke as the other due to different values on an unobserved covariate and the effect we observe here would still be significant. In Rosenbaum (2002) we see that a Γ value of 6 is required before we reach an upper bound that is above 0.05. This implies that to attribute a higher rate of death due from lung cancer due to an unobserved covariate rather than to smoking that unobserved covariate would need to produce a sixfold increase in the odds of smoking. This is a very high value of Γ. Most studies in the social sciences are not nearly this insensitive to hidden bias. Finally, I should note that these sensitivity analyses assume that matching has been done without replacement.

4 The rbounds package

The rbounds package performs Rosenbaum's method of sensitivity analyses for matched data. The package contains functions for sensitivity analyses with matched data for binary and continuous or ordered outcomes. A separate function implements sensitivity analyses when there are multiple control observations matched to the treated units. I provide an demonstration using the Lalonde data on job training and the Matching() package in R. First, I load the rbounds library. Notice that the Matching() package is automatically loaded with rbounds. I also load the Lalonde data as contained in Matching().

```
> library(rbounds)
Loading required package: Matching
Loading required package: rgenoud
##  rgenoud (Version 5.4-7, Build Date: 2007-01-04)
##  See http://sekhon.berkeley.edu/rgenoud for additional documentation.
Loading required package: MASS
##
##  Matching (Version 4.6-2, Build Date: 2008/06/26)
##  See http://sekhon.berkeley.edu/matching for additional documentation.
##  Please cite software as:
##   Jasjeet S. Sekhon. Forthcoming. ''Multivariate and Propensity Score Matching
##   Software with Automated Balance Optimization: The Matching package for R.''
##   Journal of Statistical Software.
##
>
> data(lalonde)
> attach(lalonde)
```

In this example, I use the genetic matching algorithm in the Matching package. The genetic component refers to the search algorithm used which is based on an evolutionary algorithm developed by Sekhon and Mebane (1998). Genetic matching specifically creates matches that optimize balance (Sekhon and Diamond 2005; Sekhon 2007). The outcome in this study are wages and the treatment is participation in a job training program. See Lalonde (1986) for details. I match subjects on age, education, race, marital status, an indicator for no high school degree, earning for the two years before the training program and indicators if they were unemployed in the two years before the training program. I omit an assessment of balance, but the Genmatch function balances the data with little trouble (Sekhon and Diamond 2005).

```
> Y  <- lalonde$re78   #the outcome of interest
> Tr <- lalonde$treat #the treatment of interest
>
> #Now With GenMatch
>
> X  <- cbind(age, educ, black, hisp, married, nodegr, re74, re75, u74, u75)
>
> BalanceMat <- 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*75))
>
> #Genetic Weights
> gen1 <- GenMatch(Tr=Tr, X=X, BalanceMat=BalanceMat, pop.size=50,
+                  data.type.int=FALSE, print=0, replace=FALSE)
>
> #Match
> mgen1 <- Match(Y=Y, Tr=Tr, X=X, Weight.matrix=gen1, replace=FALSE)
> summary(mgen1)

Estimate...  1767.7
AI SE......  830.85
T-stat.....  2.1275
p.val......  0.033377

Original number of observations.............  445
Original number of treated obs..............  185
Matched number of observations..............  185
Matched number of observations  (unweighted).  270
```

The output thus far is that of a standard matching analysis. We see here that after matching, the estimated treatment effect is $1767. This implies that subjects who participated in the job training program had earnings that were higher than those who did not by over $1700. This estimate is quite close to the experimental benchmark and based on the Abadie-Imbens standard is statistically significant at conventional levels. This analysis assumes that we have matched on all relevant characteristics and that there is not an unobserved confounder that may account for this difference across the treatment and control groups. A sensitivity analysis allows us to assess how reasonable this assumption may be. I, next, use the functions `psens` and `hlsens` to calculate Rosenbaum bounds for both the p-value and the estimated treatment effect. In the analysis, I set the maximum value for Γ to 1.5 with increments of 0.1. These values are a good starting place for many data sets in the social sciences. The output from the two functions is below.

```
> #Sensitivity Tests
> psens(mgen1, Gamma=1.5, GammaInc=.1)
Rosenbaum Sensitivity Test for Wilcoxon Signed Rank P-Value
     Gamma L. Bound P-Value U. Bound P-Value
[1,]   1.0           0.0346           0.0346
[2,]   1.1           0.0062           0.1271
[3,]   1.2           0.0009           0.3000
[4,]   1.3           0.0001           0.5164
[5,]   1.4           0.0000           0.7139
[6,]   1.5           0.0000           0.8539

Note: Gamma is Log Odds of Differential Assignment To Treatment
Due to Unobserved Factors
>
> hlsens(mgen1, Gamma=1.5, GammaInc=.1, .1)
Rosenbaum Sensitivity Test for Hodges-Lehmann Point Estimate
     Gamma L. Bound HL Est. U. Bound HL Est.
[1,]   1.0      1194.000000           1194.0
[2,]   1.1       560.780000           1231.2
[3,]   1.2       274.080000           1598.4
[4,]   1.3        -0.015006           1944.9
[5,]   1.4      -113.220000           2218.3
[6,]   1.5      -333.220000           2424.8

Note: Gamma is Log Odds of Differential Assignment To
Treatment Due To Unobserved Factors
```

The function `psens` provides Rosenbaum's bounds for the p-values from Wilcoxon's signed rank test. When $\Gamma = 1$, we see the p-value is quite close to that estimated in the matching analysis. The proper interpretation of this p-value is that it holds assuming there is no hidden bias due to an unobserved confounder. This p-value will not always directly match that from the matching analysis as the two may differ in the presence of outliers. We see that for even a small increase of 0.1 in Γ the p-value increases to 0.127, which is above the usual 0.05 threshold. That is even if the odds of one person being in the job training program are only 1.1 times higher because of different values on an unobserved covariate u despite being identical on the matched covariates, our inference changes. This suggests that even a small unobserved difference in a covariate would change our inference. Next, let's look at the Hodges-Lehmann point estimate.

The function `hlsens` provides Rosenbaum's bounds for the additive effect due to treatment. This can be roughly interpreted as the difference in medians across treatment and control groups, though they are not the same estimate. See Hollander and Wolfe (1999) for more details on the Hodges-Lehmman point estimate for the sign rank test. We see here that the median difference in wages if there is no hidden bias is \$1194. As we might expect, the median shift is smaller than the mean shift estimated above. We see that for $\Gamma = 1.2$ this might be as high as \$1231 or as low as \$560. The estimate is slightly more robust as it requires a Γ value of 1.3 before the upper and lower bounds bracket zero. The general conclusion then is that while it would appear the the job training program had a positive treatment effect, the finding is sensitive to possible hidden bias due to an unobserved confounder.

Sensitivity analysis should be an important component in any quantitative study. For matching estimators, a key assumption is that the selection process is accounted for by observable covariates. Rosenbaum's method of sensitivity analysis provides analysts with a method to assess how robust their findings are to hidden bias due to an unobserved confounder. The `rbounds` package provides analysts with a convenient set of software tools for performing sensitivity tests with matched data in `R`.

References

Fisher, Ronald A. 1935. *The Design of Experiments.* London: Oliver and Boyd.

Hollander, Myles, and Douglas A. Wolfe. 1999. *Nonparametric Statistical Methods.* 2nd ed. New York, NY: John Wiley and Sons.

Lalonde, R. 1986. "Evaluating the Econometric Evaluations of Training Programs." *American Economic Review* 76 (September): 604-620.

Rosenbaum, Paul R. 2002. *Observational Studies.* 2nd ed. New York, NY: Springer.

Rosenbaum, Paul R. 2005. "Observational Study." In *Encyclopedia of Statistics in Behavioral Science*, ed. Brian S. Everitt and David C. Howell. Vol. 3 John Wiley and Sons.

Sekhon, Jasjeet S. 2007. "Multivariate and Propensity Score Matching Software with Automated Balance Optimization: The Matching Package For R." *Journal of Statistical Software* Forthcoming.

Sekhon, Jasjeet S., and Alexis Diamond. 2005. "Genetic Matching for Estimating Causal Effects." Presented at the Annual Meeting of the Political Methodology, Tallahassee, FL.

Sekhon, Jasjeet S., and Walter R. Mebane. 1998. "Genetic Optimization Using Derivative: Theory and Application to Nonlinear Models." *Political Analysis* 7: 189-213.