Applied Econometrics
Problem Set #8

Nathaniel Higgins
nhiggins@jhu.edu

1 Introduction

This problem set uses data from the National Supported Work Demonstration (NSW). This is the same data used by LaLonde (1986), Heckman and Hotz (1989), Dehejia and Wahba (1999, 2002) and Smith and Todd (2004).

2 Data

`nsw.dta`

The dataset contains the following variables:

Variable name	Description
sample	1 for the experimental sample (the union of the treatment and control groups) 2 for the Current Population Survey (CPS) comparison group 3 for the Panel Study of Income Dynamics (PSID) comparison group
treated	1 for the experimental treatment group 0 for the experimental control group . (missing) for everyone else
age	age in years
educ	years of schooling
black	1 if black, 0 otherwise
hisp	1 if hispanic, 0 otherwise
married	1 if married, 0 otherwise
nodegree	1 if no high school degree, 0 otherwise
re74	real earnings in 1974
re75	real earnings in 1975
re78	real earnings in 1978
dwincl	1 if included in the Dehejia and Wahba sample, 0 otherwise
early_ra	1 if included in the early random assignment sample of Smith and Todd (2004)

3 Software

We will install `psmatch2` in class together. If you are connected to the internet, this should work (all versions of Stata):
`ssc install psmatch2`
or
`ssc install psmatch2, replace`
or you can download the .ado file manually.

4 Preliminary

Before you start:

1. Drop the CPS comparison group. We will use only the PSID comparison group and the experimental sample.
   `drop if sample == 2`

2. Use just the experimental group to estimate the effect of being in the NSW treatment on earnings in 1978. Because the NSW used randomization, we can be confident that the NSW untreated group (`treated == 0`) represents a good control for the NSW treated group (`treated == 1`).
   ```
   * Create age-squared variable
   gen agesq = age*age
   regress re78 treated age agesq educ black hisp married nodegree re74
   re75
   ```
   Note that no individuals from the PSID sample are included in the regression since one of the independent variables (`treatment`) is equal to missing (`.`) and so all PSID observations are dropped from the regression.

   See the effect of the treatment? Now, the impact of the NSW treatment was estimated using a control group that was really, really good. That is, because we are looking at experimental data, we are confident that the control group is very similar to the treatment group.

   What if, instead, we were to estimate the effect of the NSW treatment on those that got treated (treatment == 1), using as a control group individuals whose data we gather from a representative survey of the U.S. population (PSID; sample == 3)? Let's do it in #3 below.

3. What we want to do is essentially pretend that an experiment didn't happen — we'll take the treated individuals from the NSW sample (`(sample == 1) & (treated == 1)`), and compare them to the untreated individuals from the rest of the population at large (`(sample == 3) & (treated == .)`).
   ```
   * Create a mock-treated variable
   ```

```
gen mytreated = .
replace mytreated = 0 if (sample == 3) & (treated == .)
replace mytreated = 1 if (sample == 1) & (treated == 1)
* Now re-run the regression, replacing treated with mytreated
regress re78 mytreated age agesq educ black hisp married nodegree re74
re75
```

What do you notice? What is the estimated impact of the NSW program on wages? Not a very good program, eh? Well, you probably realize by comparing the two estimates that we are seeing a substantial selection bias represented in the data. **In the rest of the problem set, we'll investigate the size of this bias by comparing those who were included in the NSW program (`sample == 1`) with those that were not (`sample == 3`).**

4. Estimate two sets of propensity scores (of the probability of being in the experimental sample) using a probit model (`d = 1` if individual is in NSW, regardless of treatment status, and `d = 0` if individual is in PSID). Recall that when we talk about "estimating propensity scores," we mean that we are generating predicted probabilities using an appropriate model. The first model should include the variables age, age squared, education, black, Hispanic, married, and no degree. Call these the "coarse" propensity scores. The second model should contain the variables in the first model plus earnings in 1974 and 1975. Call these the "rich" scores. Explain what is going on with the observations that are "completely determined"? [HINT — Type the following phrase in your google machine: `stata logit completely determined`. Note that for reasons which escape me, if you substitute the term `probit` for `logit` in the phrase above, you won't get what you're after. Frowny face.]

   ```
   gen d = 0
   replace d = 1 if sample == 1
   probit d age agesq educ black hisp married nodegree
   predict pscore_coarse, pr
   probit d age agesq educ black hisp married nodegree re74 re75
   predict pscore_rich, pr
   ```

   There are 135 "failures," i.e. control observations, that are perfectly predicted (propensity score exactly equals 0). This makes them pretty useless controls. These observations were absolutely guaranteed not to be selected for treatment in the NSW project. So they are crappy observations that we shouldn't use.

5. Use the `summarize, detail` command to examine the distributions of estimated propensity scores for the experimental (NSW) and comparison group (PSID) samples (for the 1/0 variable you created to run the probit models in the previous problem). What do the descriptive statistics suggest about the common support condition in these data? What do they suggest about the comparability of the PSID comparison group?

```
save nswunsorted.dta, replace
sort d
by d:  sum pscore_coarse, detail
by d:  sum pscore_rich, detail
clear
use nswunsorted.dta
```

or

```
preserve
sort d
by d:  sum pscore_coarse, detail
by d:  sum pscore_rich, detail
restore
```

Also try:

```
scatter pscore_coarse id if d==0, color("red") || ///
scatter pscore_coarse id if d==1, ///
color("blue") legend( label(1 "d=0") label(2 "d=1") )
```

The point of this exercise is to notice that the distribution of the control (psid) and "treated" (nsw) groups do not overlap much. The 90th percentile of the psid group has the same propensity score as the 10th percentile of the nsw group.

6. Construct histograms of the estimated propensity scores for the combined experimental treatment and control groups and for the PSID comparison group. Using a command such as: `histogram phat, start(0.0) width(0.05) by(d, col(1))` will make it easy to compare the histograms, where phat is the estimated propensity score and where d = 1 for the experimental sample and d = 0 for the comparison group sample. You can type `help histogram` in Stata to find more about the histogram command; in much earlier versions of Stata you will need to use the graph command (I hope this doesn't affect anybody). What do the histograms suggest about the common support condition in these data? What do they suggest about the comparability of the PSID comparison group?

   ```
   histogram pscore_coarse, start(0.0) width(0.5) by(d, col(1))
   histogram pscore_rich, start(0.0) width(0.5) by(d, col(1))
   ```

   These graphs reveal in figures what the previous question demonstrated in numbers.

7. Drop the experimental treatment group. We are no longer interested in the effect of the NSW program, but rather we are interested in estimating the selection bias that exists (selecting in to the NSW). For the remainder of the problem set, you will construct bias estimates using the experimental control group (`(sample == 1) (treated == 0)`) and the PSID comparison group (`(sample == 3)  (treated == .)`). Construct bias estimates for both sets of estimated propensity scores

using single nearest neighbour matching *without* replacement. Impose the common support condition using the min-max version described in class. This can be accomplished using the `common` option to `psmatch2`. How many observations are dropped by imposing the common support condition? Which observations get dropped? [Hint — `psmatch2` produces a variable called `_support` that will be useful here.] Do the rich scores that contain pre-treatment earnings perform better (i.e., result in lower bias estimates) than the coarse scores that do not?

```
drop if treated == 1
psmatch2 d, outcome(re78) pscore(pscore_coarse) noreplacement common
* psmatch2 d, outcome(re78) pscore(pscore_coarse) noreplacement common
ties
sum pscore_coarse if treated == 0 & _support == 0, detail
sum pscore_coarse if treated == 0 & _support == 1, detail
psmatch2 d, outcome(re78) pscore(pscore_rich) noreplacement common
psmatch2 d, outcome(re78) pscore(pscore_rich) noreplacement common ties
sum pscore_rich if treated == 0 & _support == 0, detail
sum pscore_rich if treated == 0 & _support == 1, detail
```

27 observations are dropped by imposing the common support condition. Those control group (psid) observations with the highest propensity scores are being dropped. The bias estimates are indeed lower using the rich scores (about -$9,700 vs. about -$5,600).

5 Problems

1. Repeat the previous problem using single nearest neighbour matching *with* replacement. How much does allowing comparison group observations to be reused in the matching change the estimates in this context? Explain what you think is going on.

2. Generate bootstrap standard errors for the estimates obtained using the rich propensity scores in the preceding problem (just the immediately preceding problem using nearest neighbor matching with replacement). You should use Stata's `bs` command for this task; there is an example of how to set up the command in the help file for psmatch2 (`help psmatch2`). First use 10 replications then use 100 replications. What did you find? What is the difference between the results with 10 replications and 100 replications?

3. Create propensity score matching estimates using the rich propensity scores and kernel matching with a Gaussian (normal) kernel and bandwidths of 0.02, 0.2 and 2. Describe the resulting impact estimates. How do the estimates change as the bandwidth increases? How do the estimates differ from the single nearest neighbor matching with replacement estimates?

4. Repeat the previous problem using local linear matching rather than kernel matching. How do the estimates change as the bandwidth increases? How do the estimates differ from the single nearest neighbor matching with replacement estimates and the kernel matching estimates obtained in the previous problem?