Solution for question 6

Week8

Lecture slides, week 8
week 8 (pdf)
Audio companion, week 8
parta partb

Matching Methods for Observational Data: Part II

Lecture topics
Computational Examples of Matching Methods
1. Ben Hansen's Nuclear Plants data
optmatch exs, nuclear plants, gender ascii version for some Ben Hansen matching exs using MatchIt/optmatch
Pair matching--nuclear plants data. 1:2 optimal pair matching using MatchIt and pairmatch in optmatch plus balance diagnostics.
2. Lalonde job training data
Lalonde NSW data. Subclassification/Stratification and Full matching.
Lalonde data class handout
Rogosa R-session (using R 3.3.3) 4/1/18 redo in R 3.4.4 (sparse)
2019 lalonde Matchit: full matching, balance with cobalt love.plot and bal.tab
2019 lalonde optmatch: fullmatch with outcome analysis
Legacy Stat209 Lab 4, Lalonde Data, is arranged in pieces
a. Lab4, exposition and commands
b. Lab 4, Rogosa R-session, Base (sections 1-3)
c. Lab 4, Rogosa R-session, additional matching exercises (incl secs 4-6)
d. Lab 4, Rogosa R-session: not done until ancova is run
3. Alternative (non-matching) propensity score analyses. Propensity score weighting: Inverse Probability of Treatment Weighting (IPTW).
`twang` package from RAND, tutorials and resources. Also, an exposition using the Lalonde data and another exposition

R Implementations and Resources
1. MatchIt provides a wrapper that can call optmatch or Sekhon's genetic matching
MatchIt: Nonparametric Preprocessing for Parametric Casual Inference Daniel Ho, Kosuke Imai, Gary King, Elizabeth Stuart
MatchIt vignette
JSS May 2011 exposition: MatchIt: Nonparametric Preprocessing for Parametric Causal Inference

2. Ben Hansen (local hero) optmatch manual R News Oct 2007
optmatch:fullmatch vignette optmatch another version another good tutorial optmatch Functions for Optimal Matching
Hansen presentation: Flexible, Optimal Matching for Comparative Studies Using the optmatch package
Additional exercises (checking balance) using the nuclearplants data (class handout ex) from Mark Fredrickson here
Optmatch application paper: Full matching in an observational study of coaching for the SAT.(Scholastic Assessment Test) *Journal of the American Statistical Association*; 9/1/2004; Hansen, Ben B.
Another optmatch example presentation: Attributing Effects to a Get-Out-The-Vote Campaign Using Full Matching and Randomization Inference Jake Bowers and Ben Hansen. Data archive and computing resources for the New Haven get-out-the-vote

3. Cobalt: Using cobalt with Other Preprocessing Packages Covariate Balance Tables and Plots: A Guide to the cobalt Package

4. R Package `PSAgraphics`: Vignette JSS PSAgraphics: An R Package to Support Propensity Score Analysis *Journal of Statistical Software* February 2009, Volume 29, Issue 6.

5. `Matching` package Multivariate and Propensity Score Matching Software for Causal Inference Jasjeet S. Sekhon

---

Costs of nuclear plants

A small comparative study from a classic text

`help.matchit` *HTML Help for Matchit Commands and Models*

Description

The `help.matchit` command launches html help for Matchit commands and supported methods. The full manual is available online at `http://gking.harvard.edu/matchit`.

Usage

```
help.matchit(object)
```

Arguments

`object`	a character string representing a Matchit command or model. `help.matchit("command")` will take you to an index of Matchit commands and `help.matchit("method")` will take you to a list of matching methods. The following inputs are currently available: `exact`, `nearest`, `subclass`, `full`, `optimal`.

Author(s)

Daniel Ho <⟨daniel.ho@yale.edu⟩>; Kosuke Imai <⟨kimai@princeton.edu⟩>; Gary King <⟨king@harvard.edu⟩>; Elizabeth Stuart<⟨stuart@stat.harvard.edu⟩>

See Also

The complete document is available online at `http://gking.harvard.edu/matchit`.

Lab 4 data for matching using MatchIt Is job training effective???

`lalonde` *Data from National Supported Work Demonstration and PSID, as analyzed by Dehejia and Wahba (1999).*

Description

This is a subsample of the data from the treated group in the National Supported Work Demonstration (NSW) and the comparison sample from the Current Population Survey (CPS). This data was previously analyzed extensively by Lalonde (1986) and Dehejia and Wahba (1999). The full dataset is available at `http://www.columbia.edu/~rd247/nswdata.html`.

Usage

```
data(lalonde)
```

From Lab 4

> dim(lalonde)□□[1] 614 10□□

> names(lalonde) "treat" "age" "educ" "black" "hispan" "married" "nodegree" "re74" "re75" "re78"

> attach(lalonde) > table(treat) treat□□ 0 1 □□429 185 □□

> lalonde[1:10,] treat age educ black hispan married nodegree re74 re75 re78

Format

614 actually

A data frame with 313 observations (185 treated, 429 control). There are 10 variables measured for each individual. "treat" is the treatment assignment (1=treated, 0=control). "age" is age in years. "educ" is education in number of years of schooling. "black" is an indicator for African-American (1=African-American, 0=not). "hispan" is an indicator for being of Hispanic origin (1=Hispanic, 0=not). "married" is an indicator for married (1=married, 0=not married). "nodegree" is an indicator for whether the individual has a high school degree (1=no degree, 0=degree). "re74" is income in 1974, in U.S. dollars. "re75" is income in 1975, in U.S. dollars. "re78" is income in 1978, in U.S. dollars.

Source

http://www.columbia.edu/~rd247/nswdata.html

References

Lalonde, R. (1986). Evaluating the econometric evaluations of training programs with experimental data. *American Economic Review* 76: 604-620. \

Dehejia, R.H. and Wahba, S. (1999). Causal Effects in Nonexperimental Studies: Re-Evaluating the Evaluation of Training Programs. *Journal of the American Statistical Association* 94: 1053-1062.

`match.data` *Output Matched Data Sets*

Description

`match.data` outputs matched data sets from `matchit()`.

Usage

```
match.data <- match.data(object, group="all", distance = "distance",
weights = "weights", subclass = "subclass")
```

Arguments

`object`	The output object from `matchit()`. This is a required input.
`group`	This argument specifies for which matched group the user wants to extract the data. Available options are `"all"` (all matched units), `"treat"` (matched units in the treatment group), and `"control"` (matched units in the control group). The default is `"all"`.
`distance`	This argument specifies the variable name used to store the distance measure. The default is `"distance"`.
`weights`	This argument specifies the variable name used to store the resulting weights from matching. The default is `"weights"`.
`subclass`	This argument specifies the variable name used to store the subclass indicator. The default is `"subclass"`.

Value

Returns a subset of the original data set sent to this-is-escaped-code{, with ju

The Lalonde Data

For all of our examples, we use data from the job training program analyzed in Lalonde (1986) and Dehejia & Wahba (1999). A subsample of the data consisting of the National Supported Work Demonstration (NSW) treated group and the comparison sample from the Population Survey of Income Dynamics (PSID) is included in MATCHIT, and the full dataset is available at `http://www.columbia.edu/~rd247/nswdata.html`.[5]

The variables in this dataset are in Table 1 below. One causal effect of interest is the impact that participation in the job training program, `treat==1`, had on real earnings in 1978, `re78`, for those that participated in the program, i.e., the average treatment effect on the treated (ATT):

(1)

where `re78(treat=1)` represents the potential outcome under the treatment of the job program, and `re78(treat=0)` under control. To be clear, note that the first term (inside the expectation) in Equation 1 is *observed*, whereas the second term is the *unobserved* counterfactual of real earnings if participants had not participated. The nature of causal inference is that one of the two terms in the difference will always be unobserved. The same expression of the ATT, in mathematical notation is:

(2)

Table 1: Description of Lalonde data

Name	Description
Outcome ( )	
`re78`	Real earnings (1978)
Treatment Indicator ( )	
`treat`	Treated in job training program from March 1975-June 1977 (1 if treated, 0 if not treated)
Pre-treatment Covariates ( )	
`age`	Age
`educ`	Years of education
`black`	Race black (1 if black, 0 otherwise)
`hispan`	Race hispanic (1 if Hispanic, 0 otherwise)
`married`	Marital status (1 if married, 0 otherwise)
`nodegree`	High school degree (1 if no degree, 0 otherwise)
`re74`	Real earnings (1974)
`re75`	Real earnings (1975)

Gary King 2005-03-09

Propensity Score Methods

- Rosenbaum and Rubin. "The Central Role of the Propensity Score in Observational Studies." Biometrika 1983.
- Observational study analogue of complete randomization
- The propensity score is the probability of treatment versus control as a function of observed covariates
  - Model the reasons for treatment versus control at the level of the decision makers
  - For example, logistic regression model to predict cigarette versus cigar/pipe smoking with age, education, income, etc. as predictors
- Then subclassify (or match) on the propensity score as if it were the only covariate, e.g., 5-10 subclasses
- If correctly done, this creates balance within each subclass on **ALL** covariates used to estimate the propensity score

Matching in Statistics: Cochran's School in the 1980s

- Propensity score
  - Close matches on multivariate **x** not needed if you can match closely on scalar $\phi(\mathbf{x})$ (Rosenbaum and Rubin, 1983, 1984).
  - Good to combine matching on **x** with matching on $\phi(\mathbf{x})$, privileging closeness on $\phi(\mathbf{x})$ (Rosenbaum and Rubin, 1985).
- Computerized matching $\rightarrow$ optimal matching (Rosenbaum, 1989)

Week 8 **Propensity Scores** | Stat 209

Let $z = 1, 0$ T/C $\underset{\sim}{x}$ vector of covariates

propensity score $e(\underset{\sim}{x}) = Pr(z=1 | \underset{\sim}{x})$ scalar $\hat{e}(\underset{\sim}{x})$

cond'l prob unit w vector $\underset{\sim}{x}$ observed cov. assigned to $T (z=1)$

Thm Balancing score $b(\underset{\sim}{x})$ s.t. conditional distrib of $\underset{\sim}{x}$ given $b(\underset{\sim}{x})$ same of treated and control units

$\underset{\sim}{x} \perp\!\!\!\perp z | b(\underset{\sim}{x})$. Coarsest (low dimen) balancing score is propensity score. $Pr(x, z|e) = Pr\{\underset{\sim}{x}|e) Pr(z|e)$

Thm (result) Approx 90% reduction in bias for subclassifying at quintiles of population propensity score. $B_I = E(f(\underset{\sim}{x})|z=1) - E(f(\underset{\sim}{x})|z=0)$, B_S after strat, treatm percent reduction in bias $100(1 - B_S/B_I) \approx 90\%$

Ros Rubin
1983 Biometrika
1984 JASA

(i) The propensity score is a balancing score.
(ii) Any score that is 'finer' than the propensity score is a balancing score; moreover, x is the finest balancing score and the propensity score is the coarsest.
(iii) If treatment assignment is strongly ignorable given x, then it is strongly ignorable given any balancing score
(iv) At any value of a balancing score, the difference between the treatment and control means is an unbiased estimate of the average treatment effect at that value of the balancing score if treatment assignment is strongly ignorable. Consequently, with strongly ignorable treatment assignment, pair matching on a balancing score, subclassification on a balancing score and covariance adjustment on a balancing score can all produce unbiased estimates of treatment effects.
(v) Using sample estimates of balancing scores can produce sample balance on x.

Applications: Rubin Breast Cancer, Love (RR '84) CAD, Love Aspirin, Hansen SAT coaching, Substance Abuse (UNC), Rosenbaum, Danish downers

Rubin Ann Int Medicine

BC -> T

Table 3: Estimated 5-year Survival Rates for Node-Negative Patients in SEER from Tables 5 and 7 in U.S. GAO Report (1994). AIM pub

Propensity Score

Subclass	Treatment	n	Estimate	n*	Estimate*
1	Breast Conservation	56	85.6%	54	88.8%
	Mastectomy	1,008	86.7%	966	90.5%
2	Breast Conservation	106	82.8%	102	86.0%
	Mastectomy	964	82.8%	917	87.7%
3	Breast Conservation	193	85.2%	184	89.4%
	Mastectomy	866	88.8%	841	91.4%
4	Breast Conservation	289	88.7%	279	92.0%
	Mastectomy	978	87.3%	742	91.5%
5	Breast Conservation	462	89.0%	453	90.7%
	Mastectomy	604	88.5%	589	90.7%

* omitting patients whose deaths were unrelated to cancer.

Lalonde data

Lab 4 stratification

```
> table(propbin, treat)
                 treat
propbin            0   1
  (0,0.0401]     122   1
  (0.0401,0.0872] 116   7
  (0.0872,0.27]  101  21
  (0.27,0.671]    53  71
  (0.671,1]       37  85
> tapply(re78, list(propbin, treat),mean)
                    0    1
(0,0.0401]      10467    0
(0.0401,0.0872]  5797 7919
(0.0872,0.27]    6043 9211
(0.27,0.671]     4977 5819
(0.671,1]        4666 6030
```

counts

means re78

LAB 4 excerpt

```
# now do the logistic regression that computes propensity scores (matching packages will do this for
> glm.lalonde = glm( treat ~ age + educ + black + hispan + married + nodegree + re74 + re75,
+ data = lalonde, family = binomial)
> propen = fitted(glm.lalonde)  # now we have the propensity scores, Lab script calls these propScore
> tapply(propen, treat, quantile) # look at overlap via 5-number summary (or side-by-side boxplots)
                                  not real good overlap, as noted in class handout
$`0`
     0%     25%     50%     75%    100%
0.00908 0.03888 0.07585 0.19514 0.78917

$`1`
     0%     25%     50%     75%    100%
0.02495 0.52646 0.65368 0.72660 0.85315

> # the common use of the propensity scores (backed by theory, class handout 2/26))
> # is to stratify by quintiles

> # the simple-minded way I do it is to use "cut", Lab script is fancier programming
> ?cut  # this is a simple function to create bins
> k = 1:4
> quantile(propen, k/5)
    20%     40%     60%     80%
0.04015 0.08721 0.26978 0.67085
>  propbin = cut(propen, c(0, .04015,.08721,.26978,.67085,1))


> table(propbin, treat) # either way you display it, we do not have good overlap in the bottom
                        two quintiles, lower estimated probability for being in treatment
                        for treatment cases
                 treat
propbin             0   1
  (0,0.0401]      122   1
  (0.0401,0.0872] 116   7
  (0.0872,0.27]   101  21
  (0.27,0.671]     53  71
  (0.671,1]        37  85


> tapply(re78, list(propbin, treat),mean)  # here are the mean diffs in re78 the outcome
                                           stratified by propensity quintile
# direction of mean diffs favors treatment, job training
                     0    1
(0,0.0401]       10467    0
(0.0401,0.0872]  5797 7919
(0.0872,0.27]    6043 9211
(0.27,0.671]     4977 5819
(0.671,1]        4666 6030


> t.test(re78[propbin == bins[5]] ~ treat[propbin == bins[5]]) # t-test for quintile 5

etc
```

Optmatch creator

Email Address
734-647-5456

Ben Hansen

Research Affiliate, Population Studies Center;
Assistant Professor, Statistics Department;
Faculty Associate, Survey Research Center

Ph.D., University of California, Berkeley
M.A., University of California, Berkeley

Ben Hansen's research interests include optimal matching, propensity-score adjustments for observational studies, quasiexperimental methods, and program assessment. In recent work, he investigates informed consent and perception of risk in survey participation; how to reduce disclosure risk; and how to increase security in the dissemination of human subjects data.

Funded Research:

Human Subjects Protection and Disclosure Risk Analysis (NICHD)

New Publications

Rogowski, Freedman, Schoeni. "Neighborhoods and Health of Elderly." PSC Research Report 06-600.

Geronimus, Hicken, Keene, & Bound. "Age Patterns of Allostatic Load Scores among Blacks and Whites." *AJPH*, 2006.

Farley and Haaga, eds. *The American People: Census 2000.*

Recent Publications

Journal Articles

Evans, S.E., Ben Hansen, P.B. Stark. "Minimax expected measure confidence sets for restricted location parameters." *Bernoulli,* 11:571-590. 2005.

Hansen, Ben. "Full Matching in an Observational Study of Coaching for the SAT." *Journal of the American Statistical Association,* 99:609-618. 2004.

Full matching with propensity scores. . .

IPTW

- relieves the analyst of the need to reject lots of control subjects in order to get comparable groups;
- can be accomplished with the help of my add-on package for R, optmatch;
- does not ward off problems due to lurking variables, *a.k.a. hidden bias*, or *unmeasured confounding*; but —
- in the absence of hidden bias, should reconstruct a "lurking experiment"; and
- offers greater promise of success at this than either multiple regression or matching with a xed number k of controls.

Oh, did I mention that there is a paper? Hansen, B.B. (2004), Full matching in an observational study of coaching for the SAT, *JASA* **99**, 609–618.

Package 'MatchIt'

February 22, 2017

Version 2.4-22

Date 2017-02-22

Title Nonparametric Preprocessing for Parametric Casual Inference

Author Daniel Ho <daniel.e.ho@gmail.com>,
Kosuke Imai <kimai@Princeton.Edu>,
Gary King <king@harvard.edu>,
Elizabeth Stuart <stuart@stat.harvard.edu>

Maintainer Kosuke Imai <kimai@Princeton.Edu>

Depends R (>= 2.6), MASS

Suggests cem, optmatch, Matching, nnet, rpart, mgcv, WhatIf, R.rsp

VignetteBuilder R.rsp

Description Selects matched samples of the original treated and control groups with similar covariate distributions -- can be used to match exactly on covariates, to match on propensity scores, or perform a variety of other matching procedures. The package also implements a series of recommendations offered in Ho, Imai, King, and Stuart (2007) <DOI:10.1093/pan/mpl013>.

LazyLoad yes

LazyData yes

License GPL (>= 2)

URL http://gking.harvard.edu/matchit

NeedsCompilation no

Repository CRAN

Date/Publication 2017-02-22 14:13:05

R topics documented:

> vignette(package = "MatchIt")
Vignettes in package 'MatchIt': linked
matchit MatchIt: Nonparametric Preprocessing for Parametric Causal Inference
(source, pdf)

Package 'optmatch'

July 31, 2015

Version 0.9-5

Date 2015-07-30

Title Functions for Optimal Matching

Description Provides routines for distance based bipartite matching to reduce covariate imbalance between treatment and control groups in observational studies. Routines are provided to generate distances from GLM models (propensity score matching) and formulas (Euclidean and Mahalanobis matching), stratified matching (exact matching), and calipers. Results of the fullmatch routine are guaranteed to provide minimum average within matched set distance.

Author Ben B. Hansen <ben.hansen@umich.edu>, Mark Fredrickson <mark.m.fredrickson@gmail.com>, Josh Buckner, Josh Errickson, and Peter Solenberger, with embedded Fortran code due to Dimitri P. Bertsekas <dimitrib@mit.edu> and Paul Tseng

Maintainer Mark M. Fredrickson <mark.m.fredrickson@gmail.com>

Depends R (>= 2.15.1), stats, methods, graphics, survival

LinkingTo Rcpp

Imports Rcpp, RItools, digest

Suggests boot, biglm, testthat, brglm, arm

License file LICENSE

URL http://www.r-project.org, https://github.com/markmfredrickson/optmatch

Collate 'DenseMatrix.R' 'InfinitySparseMatrix.R' 'Ops.optmatch.dlist.R' 'Optmatch.R' 'abs.optmatch.dlist.R' 'boxplotMethods.R' 'caliper.R' 'deprecated.R' 'distUnion.R' 'exactMatch.R' 'feasible.R' 'fill.NAs.R' 'fmatch.R' 'fullmatch.R' 'makedist.R' 'match_on.R' 'matched.R' 'matched.distances.R' 'matchfailed.R' 'max.controls.cap.R' 'mdist.R' 'min.controls.cap.R' 'pairmatch.R' 'print.optmatch.R' 'print.optmatch.dlist.R' 'relaxinfo.R' 'scores.R' 'stratumStructure.R' 'subDivStrat.R' 'summary.optmatch.R' 'utilities.R' 'zzz.R' 'zzzDistanceSpecification.R'

Stat 209 [redacted]

p. 2

matchit package:MatchIt R Documentation

redacted by drr

MatchIt: Matching Software for Causal Inference

Description: 'matchit' is the main command of the package _MatchIt_, which enables parametric models for causal inference to work better by selecting well-matched subsets of the original treated and control groups. MatchIt implements a wide range of sophisticated matching methods, Matched data sets created by MatchIt can be entered easily in Zelig (<URL: http://gking.harvard.edu/zelig>) for subsequent parametric analyses. Full documentation is available online at <URL: http://gking.harvard.edu/matchit>, and help for specific commands is available through 'help.matchit'.

Usage: matchit(formula, data, method = "nearest", distance = "logit", distance.options = list(), discard = "none", reestimate = FALSE, ...)

Arguments: formula: This argument takes the usual syntax of R formula, 'treat ~ x1 + x2', where 'treat' is a binary treatment indicator and 'x1' and 'x2' are the pre-treatment covariates. Both the treatment indicator and pre-treatment covariates must be contained in the same data frame, which is specified as 'data' (see below). All of the usual R syntax for formula works. For example, 'x1:x2' represents the first order interaction term between 'x1' and 'x2', and 'I(x1^2)' represents the square term of 'x1'.

data: This argument specifies the data frame containing the variables called in 'formula'

method: This argument specifies a matching method. Currently, '"exact"' (exact matching), categorical vars '"full"' (full matching), Ben Hansen optmatch (gender equity) '"genetic"' (genetic matching), Sekhon fancy '"nearest"' (nearest neighbor matching), historical method '"optimal"' (optimal matching), and Ben Hansen optmatch, makes (2:1) '"subclass"' (subclassification) are available. The default is '"nearest"'. Note that within each of these matching methods, _MatchIt_ offers a variety of options. See <URL: http://gking.harvard.edu/matchit/docs/Inputs.html> for the complete list

References: Daniel Ho, Kosuke Imai, Gary King, and Elizabeth Stuart (2004) `Matching as Nonparametric Preprocessing for Improving Parametric Causal Inference,'' preprint available at <URL: http://gking.harvard.edu/files/abs/matchp-abs.shtml>

See Also: Please use 'help.matchit' to access the matchit reference manual. The complete document is available online at <URL: http://gking.harvard.edu/matchit>.

match.data package:MatchIt R Documentation

Output Matched Data Sets

get the list of matches

Description:

'match.data' outputs matched data sets from 'matchit()'.

Usage:

match.data <- match.data(object, group="all", distance = "distance", weights = "weights", subclass = "subclass")

Arguments:

object: The output object from {\tt matchit()}. This is an required input.

group: This argument specifies for which matched group the user wants to extract the data. Available options are '"all"' (all matched units), '"treat"' (matched units in the treatment group), and '"control"' (matched units in the control group). The default is '"all"'.

Value: Returns a subset of the original data set sent to 'matchit()', with just the matched units. The data set also contains the additional variables 'distance', 'weights', and 'subclass'. The variable 'distance' gives the estimated distance measure, and 'weights' gives the weights for each unit, generated in the matching procedure. The variable 'subclass' gives the subclass index for each unit (if applicable). See the <URL: http://gking.harvard.edu/matchit/> for the complete documentation and type 'demo(match.data)' at the R prompt to see a demonstration of the code. :

pdf or html manual

```
R version 3.4.4 (2018-03-15) -- "Someone to Lean On"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)



> ## multiple failures downloading 3.4.4 from Berkeley mirror 4/1/18
> install.packages("optmatch")
> install.packages("MatchIt")
> install.packages("lme4")
> install.packages("PSAgraphics")
> library(optmatch)
Loading required package: survival
The optmatch package has an academic license. Enter relaxinfo() for more information.
> library(MatchIt)
> library(lme4)
Loading required package: Matrix
> library(PSAgraphics)
Loading required package: rpart


> data(lalonde)
> dim(lalonde)
[1] 614  10
> attach(lalonde)
> ################## prelim compare groups on outcome measure
>  tapply(re78, treat, median)
       0        1
4975.505 4232.309
>
>  tapply(re78, treat, fivenum)
$`0`
[1]     0.0000   220.1813  4975.5050 11688.8200 25564.6700

$`1`
[1]     0.0000   485.2298  4232.3090  9642.9990 60307.9300

>
>  t.test(re78 ~ treat)

        Welch Two Sample t-test

data:  re78 by treat
t = 0.93773, df = 326.41, p-value = 0.3491
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -697.192 1967.244
sample estimates:
mean in group 0 mean in group 1
       6984.170        6349.144
```

Contents

Journal of Statistical Software

May 2011, Volume 42, Issue 8. *http://www.jstatsoft.org/*

MatchIt: Nonparametric Preprocessing for Parametric Causal Inference

Daniel E. Ho
Stanford Law School

Kosuke Imai
Princeton University

Gary King
Harvard University

Elizabeth A. Stuart
Johns Hopkins University

Abstract

MatchIt implements the suggestions of Ho, Imai, King, and Stuart (2007) for improving parametric statistical models by preprocessing data with nonparametric matching methods. **MatchIt** implements a wide range of sophisticated matching methods, making it possible to greatly reduce the dependence of causal inferences on hard-to-justify, but commonly made, statistical modeling assumptions. The software also easily fits into existing research practices since, after preprocessing data with **MatchIt**, researchers can use whatever parametric model they would have used without **MatchIt**, but produce inferences with substantially more robustness and less sensitivity to modeling assumptions. **MatchIt** is an R program, and also works seamlessly with **Zelig**.

Keywords: matching methods, causal inference, balance, preprocessing, R.

1. Introduction

1.1. What MatchIt does

MatchIt implements the suggestions of Ho, Imai, King, and Stuart (2007) for improving parametric statistical models and reducing model dependence by preprocessing data with semi-parametric and non-parametric matching methods. After appropriately preprocessing data with **MatchIt**, researchers can use whatever parametric model and software they would have used without **MatchIt**, without other modification, and produce inferences that are more robust and less sensitive to modeling assumptions. **MatchIt** reduces the dependence of

Chapter 3

User's Guide to MatchIt

3.1 Preprocessing via Matching

3.1.1 Quick Overview

The main command `matchit()` implements the matching procedures. A general syntax is:

```
> m.out <- matchit(treat ~ x1 + x2, data = mydata)
```

where `treat` is the dichotomous treatment variable, and `x1` and `x2` are pre-treatment covariates, all of which are contained in the data frame `mydata`. The dependent variable (or variables) may be included in `mydata` for convenience but is never used by MatchIt or included in the formula. This command creates the MatchIt object called `m.out`. Name the output object to see a quick summary of the results:

```
> m.out
```

3.1.2 Examples

To run any of the examples below, you first must load the library and and data:

```
> library(MatchIt)
> data(lalonde)
```

Our example data set is a subset of the job training program analyzed in Lalonde (1986) and Dehejia and Wahba (1999). MatchIt includes a subsample of the original data consisting of the National Supported Work Demonstration (NSW) treated group and the comparison sample from the Population Survey of Income Dynamics (PSID).[1] The variables in this data set include participation in the job training program (`treat`, which is equal to 1 if participated in the program, and 0 otherwise), age (`age`), years of education (`educ`), race

[1] This data set, `lalonde`, was created using NSWRE74_TREATED.TXT and CPS3_CONTROLS.TXT from http://www.columbia.edu/~rd247/nswdata.

(`black` which is equal to 1 if black, and 0 otherwise; `hispan` which is equal to 1 if hispanic, and 0 otherwise), marital status (`married`, which is equal to 1 if married, 0 otherwise), high school degree (`nodegree`, which is equal to 1 if no degree, 0 otherwise), 1974 real earnings (`re74`), 1975 real earnings (`re75`), and the main outcome variable, 1978 real earnings (`re78`).

3.1.2.1 Exact Matching

The simplest version of matching is exact. This technique matches *each* treated unit to *all* possible control units with exactly the same values on all the covariates, forming subclasses such that within each subclass all units (treatment and control) have the same covariate values. Exact matching is implemented in MatchIt using `method = "exact"`. Exact matching will be done on all covariates included on the right-hand side of the `formula` specified in the MatchIt call. There are no additional options for exact matching. (Exact restrictions on a subset of covariates can also be specified in nearest neighbor matching; see Section 3.1.2.3.) The following example can be run by typing `demo(exact)` at the R prompt,

```
> m.out <- matchit(treat ~ educ + black + hispan, data = lalonde,
                   method = "exact")
```

3.1.2.2 Subclassification

When there are many covariates (or some covariates can take a large number of values), finding sufficient exact matches will often be impossible. The goal of subclassification is to form subclasses, such that in each the distribution (rather than the exact values) of covariates for the treated and control groups are as similar as possible. Various subclassification schemes exist, including the one based on a scalar distance measure such as the propensity score estimated using the `distance` option (see Section 4.1.0.2.2). Subclassification is implemented in MatchIt using `method = "subclass"`.

The following example script can be run by typing `demo(subclass)` at the R prompt,

```
> m.out <- matchit(treat ~ re74 + re75 + educ + black + hispan + age,
                   data = lalonde, method = "subclass")
```

The above syntax forms 6 subclasses, which is the default number of subclasses, based on a distance measure (the propensity score) estimated using logistic regression. By default, each subclass will have approximately the same number of treated units.

Subclassification may also be used in conjunction with nearest neighbor matching described below, by leaving the default of `method = "nearest"` but adding the option `subclass`. When you choose this option, MatchIt selects matches using nearest neighbor matching, but after the nearest neighbor matches are chosen it places them into subclasses, and adds a variable to the output object indicating subclass membership.

3.1.2.3 Nearest Neighbor Matching

Nearest neighbor matching selects the r (default=1) best control matches for each individual in the treatment group (excluding those discarded using the `discard` option). Matching is done using a distance measure specified by the `distance` option (default=logit). Matches are chosen for each treated unit one at a time, with the order specified by the `m.order` command (default=largest to smallest). At each matching step we choose the control unit that is not yet matched but is closest to the treated unit on the distance measure.

Nearest neighbor matching is implemented in MatchIt using the `method = "nearest"` option. The following example script can be run by typing `demo(nearest)`:

```
> m.out <- matchit(treat ~ re74 + re75 + educ + black + hispan + age,
                   data = lalonde, method = "nearest")
```

3.1.2.4 Optimal Matching

The default nearest neighbor matching method in MatchIt is "greedy" matching, where the closest control match for each treated unit is chosen one at a time, without trying to minimize a global distance measure. In contrast, "optimal" matching finds the matched samples with the smallest average absolute distance across all the matched pairs. Gu and Rosenbaum (1993) find that greedy and optimal matching approaches generally choose the same sets of controls for the overall matched samples, but optimal matching does a better job of minimizing the distance within each pair. In addition, optimal matching can be helpful when there are not many appropriate control matches for the treated units.

Optimal matching is performed with MatchIt by setting `method = "optimal"`, which automatically loads an add-on package called `optmatch` (Hansen 2004). The following example can also be run by typing `demo(optimal)` at the R prompt. We conduct 2:1 optimal ratio matching based on the propensity score from the logistic regression.

```
> m.out <- matchit(treat ~ re74 + re75 + age + educ, data = lalonde,
                   method = "optimal", ratio = 2)
```

3.1.2.5 Full Matching

Full matching is a particular type of subclassification that forms the subclasses in an optimal way (Rosenbaum 2002; Hansen 2004). A fully matched sample is composed of matched sets, where each matched set contains one treated unit and one or more controls (or one control unit and one or more treated units). As with subclassification, the only units not placed into a subclass will be those discarded (if a `discard` option is specified) because they are outside the range of common support. Full matching is optimal in terms of minimizing a weighted average of the estimated distance measure between each treated subject and each control subject within each subclass.

Full matching can be performed with MatchIt by setting `method = "full"`. Just as with optimal matching, we use the `optmatch` package (Hansen 2004), which automatically

3.1.2.3 Nearest Neighbor Matching

Nearest neighbor matching selects the r (default=1) best control matches for each individual in the treatment group (excluding those discarded using the `discard` option). Matching is done using a distance measure specified by the `distance` option (default=logit). Matches are chosen for each treated unit one at a time, with the order specified by the `m.order` command (default=largest to smallest). At each matching step we choose the control unit that is not yet matched but is closest to the treated unit on the distance measure.

Nearest neighbor matching is implemented in MatchIt using the `method = "nearest"` option. The following example script can be run by typing `demo(nearest)`:

```
> m.out <- matchit(treat ~ re74 + re75 + educ + black + hispan + age,
                   data = lalonde, method = "nearest")
```

3.1.2.4 Optimal Matching

The default nearest neighbor matching method in MatchIt is "greedy" matching, where the closest control match for each treated unit is chosen one at a time, without trying to minimize a global distance measure. In contrast, "optimal" matching finds the matched samples with the smallest average absolute distance across all the matched pairs. Gu and Rosenbaum (1993) find that greedy and optimal matching approaches generally choose the same sets of controls for the overall matched samples, but optimal matching does a better job of minimizing the distance within each pair. In addition, optimal matching can be helpful when there are not many appropriate control matches for the treated units.

Optimal matching is performed with MatchIt by setting `method = "optimal"`, which automatically loads an add-on package called `optmatch` (Hansen 2004). The following example can also be run by typing `demo(optimal)` at the R prompt. We conduct 2:1 optimal ratio matching based on the propensity score from the logistic regression.

```
> m.out <- matchit(treat ~ re74 + re75 + age + educ, data = lalonde,
                   method = "optimal", ratio = 2)
```

3.1.2.5 Full Matching

Full matching is a particular type of subclassification that forms the subclasses in an optimal way (Rosenbaum 2002; Hansen 2004). A fully matched sample is composed of matched sets, where each matched set contains one treated unit and one or more controls (or one control unit and one or more treated units). As with subclassification, the only units not placed into a subclass will be those discarded (if a `discard` option is specified) because they are outside the range of common support. Full matching is optimal in terms of minimizing a weighted average of the estimated distance measure between each treated subject and each control subject within each subclass.

Full matching can be performed with MatchIt by setting `method = "full"`. Just as with optimal matching, we use the `optmatch` package (Hansen 2004), which automatically

loads when needed. The following example with full matching (using the default propensity score based on logistic regression) can also be run by typing `demo(full)` at the R prompt:

```
> m.out <- matchit(treat ~ age + educ + black + hispan + married +
                   nodegree + re74 + re75, data = lalonde, method = "full")
```

3.1.2.6 Genetic Matching

Genetic matching automates the process of finding a good matching solution (Diamond and Sekhon 2005). The idea is to use a genetic search algorithm to find a set of weights for each covariate such that the a version of optimal balance is achieved after matching. As currently implemented, matching is done with replacement using the matching method of Abadie and Imbens (2007) and balance is determined by two univariate tests, paired t-tests for dichotomous variables and a Kolmogorov-Smirnov test for multinomial and continuous variables, but these options can be changed.

Genetic matching can be performed with MatchIt by setting `method = "genetic"`, which automatically loads the `Matching` (**?**) package. The following example of genetic matching (using the estimated propensity score based on logistic regression as one of the covariates) can also be run by typing `demo(genetic)`:

```
> m.out <- matchit(treat ~ age + educ + black + hispan + married + nodegree +
                   re74 + re75, data = lalonde, method = "genetic")
```

3.1.2.7 Coarsened Exact Matching

Coarsened Exact Matching (CEM) is a Monotonoic Imbalance Bounding (MIB) matching method — which means that the balance between the treated and control groups is chosen by the user ex ante rather than discovered through the usual laborious process of checking after the fact and repeatedly reestimating, and so that adjusting the imbalance on one variable has no effect on the maximum imbalance of any other. CEM also strictly bounds through ex ante user choice both the degree of model dependence and the average treatment effect estimation error, eliminates the need for a separate procedure to restrict data to common empirical support, meets the congruence principle, is robust to measurement error, works well with multiple imputation methods for missing data, and is extremely fast computationally even with very large data sets. CEM also works well for multicategory treatments, determining blocks in experimental designs, and evaluating extreme counterfactuals (Iacus et al. 2008b).

CEM can be performed with MatchIt by setting `method = "cem"`, which automatically loads the `cem` package. The following examples of CEM (with automatic coarsening) can also be run by typing `demo(cem)`:

```
m.out <- matchit(treat ~ age + educ + black + hispan + married + nodegree
                 + re74 + re75, data = lalonde, method = "cem")
```

3.2 Checking Balance

3.2.1 Quick Overview

To check balance, use `summary(m.out)` for numerical summaries and `plot(m.out)` for graphical summaries.

3.2.2 Details

3.2.2.1 The `summary()` Command

The `summary()` command gives measures of the balance between the treated and control groups in the full (original) data set, and then in the matched data set. If the matching worked well, the measures of balance should be smaller in the matched data set (smaller values of the measures indicate better balance).

The `summary()` output for subclassification is the same as that for other types of matching, except that the balance statistics are shown separately for each subclass, and the overall balance in the matched samples is calculated by aggregating across the subclasses, where each subclass is weighted by the number of units in the subclass. For exact matching, the covariate values within each subclass are guaranteed to be the same, and so the measures of balance are not output for exact matching; only the sample sizes in each subclass are shown.

- **Balance statistics:** The statistics the `summary()` command provides include means, the original control group standard deviation (where applicable), mean differences, standardized mean differences, and (median, mean and maximum) Quantile-Quantile (Q-Q) plot differences. In addition, the `summary()` command will report (a) the matched call, (b) how many units were matched, unmatched, or discarded due to the `discard` option (described below), and (c) the percent improvement in balance for each of the balance measures, defined as $100((|a| - |b|)/|a|)$, where a is the balance before and b is the balance after matching. For each set of units (original and matched data sets, with weights used as appropriate in the matched data sets), the following statistics are provided:

    1. "Means Treated" and "Means Control" show the weighted means in the treated and control groups
    2. "SD Control" is the standard deviation calculated in the control group (where applicable)
    3. "Mean Diff" is the difference in means between the groups
    4. The final three columns of the summary output give summary statistics of a Q-Q plot (see below for more information on these plots). Those columns give the median, mean, and maximum distance between the two empirical quantile functions (treated and control groups). Values greater than 0 indicate deviations between the groups in some part of the empirical distributions. The plots of the

Costs of nuclear plants

A small comparative study from a classic text

Details

x is a formula of the form Z ~ X1 + X2, where Z is indicates treatment or control status, and X1 and X2 are variables can be converted to factors. Any additional arguments are passed to model.frame (e.g., a data argument containing Z, X1, and X2).

The the arguments scores and width must be passed together. The function will apply the caliper implied by the scores and the width while also adding in blocking factors.

Value

A factor grouping units, suitable for exactMatch.

---

nuclearplants *Nuclear Power Station Construction Data*

Description

The nuclearplants data frame has 32 rows and 11 columns.

The data relate to the construction of 32 light water reactor (LWR) plants constructed in the U.S.A in the late 1960's and early 1970's. The data was collected with the aim of predicting the cost of construction of further LWR plants. 6 of the power plants had partial turnkey guarantees and it is possible that, for these plants, some manufacturers' subsidies may be hidden in the quoted capital costs.

Usage

```
nuclearplants
```

Format

This data frame contains the following columns:

cost The capital cost of construction in millions of dollars adjusted to 1976 base.

date The date on which the construction permit was issued. The data are measured in years since January 1 1990 to the nearest month.

t1 The time between application for and issue of the construction permit.

t2 The time between issue of operating license and construction permit.

cap The net capacity of the power plant (MWe).

pr A binary variable where 1 indicates the prior existence of a LWR plant at the same site.

ne A binary variable where 1 indicates that the plant was constructed in the north-east region of the U.S.A.

ct A binary variable where 1 indicates the use of a cooling tower in the plant.

bw A binary variable where 1 indicates that the nuclear steam supply system was manufactured by Babcock-Wilcox.

cum.n The cumulative number of power plants constructed by each architect-engineer.

pt A binary variable where 1 indicates those plants with partial turnkey guarantees.

see handout

Existing site

	date	capacity
A	2.3	660
B	3.0	660
C	3.4	420
D	3.4	130
E	3.9	650
F	5.9	430
G	5.1	420

New site

	date	capacity
H	3.6	290
I	2.3	660
J	3.0	660
K	2.9	110
L	3.2	420
M	3.4	60
N	3.3	390
O	3.6	160
P	3.8	390
Q	3.4	130
R	3.9	650
S	3.9	450
T	3.4	380
U	4.5	440
V	4.2	690
W	3.8	510
X	4.7	390
Y	5.4	140
Z	6.1	730

“date” is date of construction, in years after 1965; “capacity” is net capacity of the power plant, in MWe above 400.

Optmatch: Flexible, Optimal Matching for Observational Studies

Ben B. Hansen

Observational studies compare subjects who received a specified treatment to others who did not, without controlling assignment to treatment and comparison groups. When the groups differ at baseline in ways that are relevant to the outcome, the study has to adjust for the differences. An old and particularly direct method of making these adjustments is to match treated subjects to controls who are similar in terms of their pretreatment characteristics, then conduct an outcome analysis conditioning upon the matched sets. Adjustments of this type enjoy properties of robustness (Rubin, 1979) and transparency not shared with purely model-based adjustments, such as covariance adjustment without matching or stratification; and with the introduction of propensity scores to matching (Rosenbaum and Rubin, 1985), the approach was shown to be more broadly applicable than was previously thought. Arguably, the reach of techniques based on matching now exceeds that of purely model-based adjustment (Hansen, 2004).

To achieve these benefits, matched adjustment requires the analyst to articulate a distinction between desirable and undesirable potential matches, and then to match treated and control subjects in such a way as to favor the more desirable pairings. Propensity scoring fits under the first of these tasks, as do the construction of Mahalanobis matching metrics (Rosenbaum and Rubin, 1985), prognostic scoring (Hansen, 2006b), and the distance metric optimization of Diamond and Sekhon (2006). The second task, matching itself, is less statistical in nature, but doing it well can substantially improve the power and robustness of matched inference (Hansen and Klopfer, 2006; Hansen, 2004). The main purpose of **optmatch** is to relieve the analyst of responsibility for this important, if potentially tedious, undertaking, freeing attention for other aspects of the analysis. Given discrepancies between each treatment and control subject that might potentially be matched, **optmatch** places them into non-overlapping matched sets, in the process solving the discrete optimization problems needed to make sums of matched discrepancies as small as possible; after this, the analysis can proceed using permutation inference (Rosenbaum, 2002; Hothorn et al., 2006; Bowers and Hansen, 2006), conditional inference (Breslow and Day, 1980; Cox and Snell, 1989; Hansen, 2004; Lumley and Therneau, 2006), approximately conditional inference (Pierce and Peters, 1992; Brazzale, 2005; Brazzale et al., 2006), or multilevel models (Smith, 1997; Raudenbush and Bryk, 2002; Gelman and Hill, 2006).

Optimal matching of two groups

To illustrate the meaning of optimal matching, consider Cox and Snell's (1981, p.81) study of costs of nuclear power. Of 26 light water reactor plants constructed in the U.S. between 1967 and 1972, seven had been built on the site of existing plants. The problem is to estimate the cost benefit (or penalty) of building on an existing site as opposed to a new one. A matched analysis seeks to adjust for background characteristics determinative of cost, such as the date of construction and the capacity of the plant, by linking similar refurbished and new plants: plants of about the same capacity and constructed at about the same time, for example. To highlight the analogy with intervention studies, I refer to existing-site plants as "treatments" and new-site plants as "controls."

Consider the problem of arranging the plants in disjoint triples, each containing one treatment and two controls, placing each treatment and 14 of the 19 controls into some matched triple or another. A straightforward way to create such a match is to move down the list of treatments, pairing each to the two most similar controls that have not yet been matched; this is *nearest-available matching*. Figure 1 shows the 26 plants, their capacities and dates of construction, and a 1 : 2 matching constructed in this way. First A was matched to I and J, then B to L and N, and so forth. This example is discussed by Rosenbaum (2002, ch.10).

Existing site

	date	capacity
A	2.3	660
B	3.0	660
C	3.4	420
D	3.4	130
E	3.9	650
F	5.9	430
G	5.1	420

New site

	date	capacity
H	3.6	290
I	2.3	660
J	3.0	660
K	2.9	110
L	3.2	420
M	3.4	60
N	3.3	390
O	3.6	160
P	3.8	390
Q	3.4	130
R	3.9	650
S	3.9	450
T	3.4	380
U	4.5	440
V	4.2	690
W	3.8	510
X	4.7	390
Y	5.4	140
Z	6.1	730

"date" is date of construction, in years after 1965; "capacity" is net capacity of the power plant, in MWe above 400.

Figure 1: 1:2 matching by a nearest-available algorithm.

How might this process be improved? To complete step i, the nearest-available algorithm requires

ISSN 1609-3631

Package 'optmatch'

July 31, 2015

Version 0.9-5

Date 2015-07-30

Title Functions for Optimal Matching

Description Provides routines for distance based bipartite matching to reduce covariate imbalance between treatment and control groups in observational studies. Routines are provided to generate distances from GLM models (propensity score matching) and formulas (Euclidean and Mahalanobis matching), stratified matching (exact matching), and calipers. Results of the fullmatch routine are guaranteed to provide minimum average within matched set distance.

Author Ben B. Hansen <ben.hansen@umich.edu>, Mark Fredrickson <mark.m.fredrickson@gmail.com>, Josh Buckner, Josh Errickson, and Peter Solenberger, with embedded Fortran code due to Dimitri P. Bertsekas <dimitrib@mit.edu> and Paul Tseng

Maintainer Mark M. Fredrickson <mark.m.fredrickson@gmail.com>

Depends R (>= 2.15.1), stats, methods, graphics, survival

LinkingTo Rcpp

Imports Rcpp, RItools, digest

Suggests boot, biglm, testthat, brglm, arm

License file LICENSE

URL http://www.r-project.org, https://github.com/markmfredrickson/optmatch

Collate 'DenseMatrix.R' 'InfinitySparseMatrix.R' 'Ops.optmatch.dlist.R' 'Optmatch.R' 'abs.optmatch.dlist.R' 'boxplotMethods.R' 'caliper.R' 'deprecated.R' 'distUnion.R' 'exactMatch.R' 'feasible.R' 'fill.NAs.R' 'fmatch.R' 'fullmatch.R' 'makedist.R' 'match_on.R' 'matched.R' 'matched.distances.R' 'matchfailed.R' 'max.controls.cap.R' 'mdist.R' 'min.controls.cap.R' 'pairmatch.R' 'print.optmatch.R' 'print.optmatch.dlist.R' 'relaxinfo.R' 'scores.R' 'stratumStructure.R' 'subDivStrat.R' 'summary.optmatch.R' 'utilities.R' 'zzz.R' 'zzzDistanceSpecification.R'

Details

x is a formula of the form Z ~ X1 + X2, where Z is indicates treatment or control status, and X1 and X2 are variables can be converted to factors. Any additional arguments are passed to model.frame (e.g., a data argument containing Z, X1, and X2).

The the arguments scores and width must be passed together. The function will apply the caliper implied by the scores and the width while also adding in blocking factors.

Value

A factor grouping units, suitable for exactMatch.

---

nuclearplants *Nuclear Power Station Construction Data*

Description

The nuclearplants data frame has 32 rows and 11 columns.

The data relate to the construction of 32 light water reactor (LWR) plants constructed in the U.S.A in the late 1960's and early 1970's. The data was collected with the aim of predicting the cost of construction of further LWR plants. 6 of the power plants had partial turnkey guarantees and it is possible that, for these plants, some manufacturers' subsidies may be hidden in the quoted capital costs.

Usage

```
nuclearplants
```

Format

This data frame contains the following columns:

cost The capital cost of construction in millions of dollars adjusted to 1976 base.

date The date on which the construction permit was issued. The data are measured in years since January 1 1990 to the nearest month.

t1 The time between application for and issue of the construction permit.

t2 The time between issue of operating license and construction permit.

cap The net capacity of the power plant (MWe).

pr A binary variable where 1 indicates the prior existence of a LWR plant at the same site.

ne A binary variable where 1 indicates that the plant was constructed in the north-east region of the U.S.A.

ct A binary variable where 1 indicates the use of a cooling tower in the plant.

bw A binary variable where 1 indicates that the nuclear steam supply system was manufactured by Babcock-Wilcox.

cum.n The cumulative number of power plants constructed by each architect-engineer.

pt A binary variable where 1 indicates those plants with partial turnkey guarantees.

Existing site		
	date	capacity
A	2.3	660
B	3.0	660
C	3.4	420
D	3.4	130
E	3.9	650
F	5.9	430
G	5.1	420

New site		
	date	capacity
H	3.6	290
I	2.3	660
J	3.0	660
K	2.9	110
L	3.2	420
M	3.4	60
N	3.3	390
O	3.6	160
P	3.8	390
Q	3.4	130
R	3.9	650
S	3.9	450
T	3.4	380
U	4.5	440
V	4.2	690
W	3.8	510
X	4.7	390
Y	5.4	140
Z	6.1	730

Example: 1:2 matching by a traditional, greedy algorithm.

"date" is date of construction, in years after 1965; "capacity" is net capacity of the power plant, in MWe above 400.

Existing site		
	date	capacity
A	2.3	660
B	3.0	660
C	3.4	420
D	3.4	130
E	3.9	650
F	5.9	430
G	5.1	420

New site		
	date	capacity
H	3.6	290
I	2.3	660
J	3.0	660
K	2.9	110
L	3.2	420
M	3.4	60
N	3.3	390
O	3.6	160
P	3.8	390
Q	3.4	130
R	3.9	650
S	3.9	450
T	3.4	380
U	4.5	440
V	4.2	690
W	3.8	510
X	4.7	390
Y	5.4	140
Z	6.1	730

Example: 1:2 matching by a traditional, greedy algorithm.

"date" is date of construction, in years after 1965; "capacity" is net capacity of the power plant, in MWe above 400.

Existing site		
	date	capacity
A	2.3	660
B	3.0	660
C	3.4	420
D	3.4	130
E	3.9	650
F	5.9	430
G	5.1	420

New site		
	date	capacity
H	3.6	290
I	2.3	660
J	3.0	660
K	2.9	110
L	3.2	420
M	3.4	60
N	3.3	390
O	3.6	160
P	3.8	390
Q	3.4	130
R	3.9	650
S	3.9	450
T	3.4	380
U	4.5	440
V	4.2	690
W	3.8	510
X	4.7	390
Y	5.4	140
Z	6.1	730

Example: 1:2 matching by a traditional, greedy algorithm.

"date" is date of construction, in years after 1965; "capacity" is net capacity of the power plant, in MWe above 400.

Existing site		
	date	capacity
A	2.3	660
B	3.0	660
C	3.4	420
D	3.4	130
E	3.9	650
F	5.9	430
G	5.1	420

New site		
	date	capacity
H	3.6	290
I	2.3	660
J	3.0	660
K	2.9	110
L	3.2	420
M	3.4	60
N	3.3	390
O	3.6	160
P	3.8	390
Q	3.4	130
R	3.9	650
S	3.9	450
T	3.4	380
U	4.5	440
V	4.2	690
W	3.8	510
X	4.7	390
Y	5.4	140
Z	6.1	730

Example: 1:2 matching by a traditional, greedy algorithm.

"date" is date of construction, in years after 1965; "capacity" is net capacity of the power plant, in MWe above 400.

Existing site		
	date	capacity
A	2.3	660
B	3.0	660
C	3.4	420
D	3.4	130
E	3.9	650
F	5.9	430
G	5.1	420

New site		
	date	capacity
H	3.6	290
I	2.3	660
J	3.0	660
K	2.9	110
L	3.2	420
M	3.4	60
N	3.3	390
O	3.6	160
P	3.8	390
Q	3.4	130
R	3.9	650
S	3.9	450
T	3.4	380
U	4.5	440
V	4.2	690
W	3.8	510
X	4.7	390
Y	5.4	140
Z	6.1	730

Example: 1:2 matching by a traditional, greedy algorithm.

"date" is date of construction, in years after 1965; "capacity" is net capacity of the power plant, in MWe above 400.

Existing site		
	date	capacity
A	2.3	660
B	3.0	660
C	3.4	420
D	3.4	130
E	3.9	650
F	5.9	430
G	5.1	420

New site		
	date	capacity
H	3.6	290
I	2.3	660
J	3.0	660
K	2.9	110
L	3.2	420
M	3.4	60
N	3.3	390
O	3.6	160
P	3.8	390
Q	3.4	130
R	3.9	650
S	3.9	450
T	3.4	380
U	4.5	440
V	4.2	690
W	3.8	510
X	4.7	390
Y	5.4	140
Z	6.1	730

Example: 1:2 matching by a traditional, greedy algorithm.

"date" is date of construction, in years after 1965; "capacity" is net capacity of the power plant, in MWe above 400.

Existing site		
	date	capacity
A	2.3	660
B	3.0	660
C	3.4	420
D	3.4	130
E	3.9	650
F	5.9	430
G	5.1	420

New site		
	date	capacity
H	3.6	290
I	2.3	660
J	3.0	660
K	2.9	110
L	3.2	420
M	3.4	60
N	3.3	390
O	3.6	160
P	3.8	390
Q	3.4	130
R	3.9	650
S	3.9	450
T	3.4	380
U	4.5	440
V	4.2	690
W	3.8	510
X	4.7	390
Y	5.4	140
Z	6.1	730

Example: 1:2 matching by a traditional, greedy algorithm.

"date" is date of construction, in years after 1965; "capacity" is net capacity of the power plant, in MWe above 400.

Existing site		
	date	capacity
A	2.3	660
B	3.0	660
C	3.4	420
D	3.4	130
E	3.9	650
F	5.9	430
G	5.1	420

New site		
	date	capacity
H	3.6	290
I	2.3	660
J	3.0	660
K	2.9	110
L	3.2	420
M	3.4	60
N	3.3	390
O	3.6	160
P	3.8	390
Q	3.4	130
R	3.9	650
S	3.9	450
T	3.4	380
U	4.5	440
V	4.2	690
W	3.8	510
X	4.7	390
Y	5.4	140
Z	6.1	730

Optimal vs. Greedy matching

By evaluating potential matches all together rather than sequentially, optimal matching (blue lines) reduces the sum of distances from 82 to 63.

(Match distance is to "optimal matching" as statistical model is to "maximum likelihood.")

after installing (Lab4) Week 8

Ben Hansen matching Exs using MatchIt wrapper
Stat 209

```
> library(MatchIt)
Loading required package: MASS
Loading required package: optmatch
Loading required package: Matching
##  MatchIt (Version 2.3-1, built: 2007-10-11)
##  Please refer to http://gking.harvard.edu/matchit for full documentation
> data(nuclear) # get from "boot"; library(boot)
> nuclear
     cost  date t1 t2  cap pr ne ct bw cum.n pt
1   460.1 68.58 14 46  687  0  1  0  0    14  0
2   453.0 67.33 10 73 1065  0  0  1  0     1  0
3   443.2 67.33 10 85 1065  1  0  1  0     1  0
4   652.3 68.00 11 67 1065  0  1  1  0    12  0
5   642.2 68.00 11 78 1065  1  1  1  0    12  0
.....................................................
24 608.8 70.08 19 58  821  1  0  0  0     3  0
29 284.9 67.83 12 63  886  0  0  0  1    11  1
32 270.7 67.83  7 80  886  1  0  0  1    11  1
```

> library(boot)
> data(nuclear)
insert

orig data set 'nuclearplants'

> library(optmatch)
> data(nuclearplants)

```
> bennuke = subset(nuclear, subset = pt == 0, select = c(date,cap,pr))
> bennuke$cap = bennuke$cap - 400 > bennuke$date = bennuke$date - 65
> bennuke
   date cap pr                  date cap pr
1  3.58 287  0               14 3.92 650  0
2  2.33 665  0               15 3.92 450  0
3  2.33 665  1               16 3.42 378  0
4  3.00 665  0               17 4.50 445  0
5  3.00 665  1               18 3.42 130  1
6  2.92 114  0               19 4.17 690  0
7  3.17 422  0               20 3.92 650  1
8  3.42  57  0               21 3.75 513  0
9  3.42 422  1               22 5.92 428  1
10 3.33 392  0               23 4.67 386  0
11 3.58 160  0               24 5.08 421  1
12 3.75 390  0               25 5.42 138  0
13 3.42 130  0               26 6.08 730  0
```

subset and transform to get BH cx

7 prior plants for matching

2 to 1 match

```
> mopt2= matchit(pr ~ date + cap, data = bennuke, method = "optimal", ratio = 2)
> summary(mopt2)
Percent Balance Improvement:
Sample sizes:           Control Treated
     All                   19       7
     Matched               14       7
     Unmatched              5       0
     Discarded              0       0
```

diagnostics, plots Lab4

see online for balance

lots of options full, matchit see Lab4, MatchIt docs p.2 handout

see Gender p.2

```
> #subclass gives you the matching
> mopt2.data = match.data(mopt2)   > mopt2.data
   date cap pr distance weights subclass
2  2.33 665  0   0.3409       1        1
3  2.33 665  1   0.3409       1        1
4  3.00 665  0   0.3513       1        2
5  3.00 665  1   0.3513       1        2
6  2.92 114  0   0.1514       1        4
7  3.17 422  0   0.2518       1        6
9  3.42 422  1   0.2551       1        3
10 3.33 392  0   0.2427       1        3
12 3.75 390  0   0.2473       1        3
13 3.42 130  0   0.1601       1        4
14 3.92 650  0   0.3590       1        2
15 3.92 450  0   0.2727       1        7
17 4.50 445  0   0.2787       1        6
18 3.42 130  1   0.1601       1        4
19 4.17 690  0   0.3817       1        5
20 3.92 650  1   0.3590       1        5
21 3.75 513  0   0.2960       1        1
22 5.92 428  1   0.2917       1        6
23 4.67 386  0   0.2578       1        7
24 5.08 421  1   0.2770       1        7
26 6.08 730  0   0.4328       1        5
> #this corresponds to hansen slide 45
```

create matching data, then list

subclass marks the matches

default distance measure is propensity score demonstrated in "more nukes"

Check Balance 2:1 match handout

```
>  mopt2= matchit(pr ~ date + cap, data = bennuke, method = "optimal", ratio = 2)
Installing package(s) into 'C:/Users/rag/Documents/R/win-library/2.14'
(as 'lib' is unspecified)
trying URL 'http://cran.cnr.Berkeley.edu/bin/windows/contrib/2.14/optmatch_0.7-1.zip'
Content type 'application/zip' length 330389 bytes (322 Kb)
opened URL
downloaded 322 Kb

package 'optmatch' successfully unpacked and MD5 sums checked

The downloaded packages are in
        C:\Users\rag\AppData\Local\Temp\RtmpWi648r\downloaded_packages
Loading required package: optmatch
> summary(mopt2)

Call:
matchit(formula = pr ~ date + cap, data = bennuke, method = "optimal",
    ratio = 2)

Summary of balance for all data:
propensity  Means Treated Means Control SD Control Mean Diff eQQ Med eQQ Mean  eQQ Max
score
distance          0.2907        0.2613     0.0832    0.0294  0.0398   0.0454    0.0852
date              3.8700        3.8079     0.8807    0.0621  0.1600   0.1529    0.5800
cap             483.0000      403.2632   214.1816   79.7368 65.0000 100.1429 283.0000


Summary of balance for matched data:
          Means Treated Means Control SD Control Mean Diff eQQ Med eQQ Mean  eQQ Max
distance          0.2907        0.2874     0.0799    0.0033   0.019   0.0263    0.0738
date              3.8700        3.7807     0.9177    0.0893   0.090   0.2129    0.9100
cap             483.0000      474.4286   194.3885    8.5714  30.000  42.8571 137.0000

Percent Balance Improvement:
          Mean Diff. eQQ Med eQQ Mean  eQQ Max
distance     88.8018 52.3558  42.1473  13.3177
date        -43.7651 43.7500 -39.2523 -56.8966
cap          89.2504 53.8462  57.2040  51.5901

Sample sizes:
           Control Treated
All             19       7
Matched         14       7
Unmatched        5       0
Discarded        0       0
```

more nuke matching Week 8 Stat 209

```
R version 2.8.1 (2008-12-22)
> install.packages("MatchIt")
> data(nuclear)
> bennuke = subset(nuclear, subset = pt == 0, select = c(date,cap,pr))
> bennuke$cap = bennuke$cap - 400
> bennuke$date = bennuke$date - 65
> attach(bennuke)
> mopt1= matchit(pr ~ date + cap, data = bennuke, method = "optimal")
> summary(mopt1)
Call:
matchit(formula = pr ~ date + cap, data = bennuke, method = "optimal")
Summary of balance for all data:
         Means Treated Means Control SD Control Mean Diff eQQ Med eQQ Mean eQQ Max
distance         0.291         0.261      0.083     0.029    0.04    0.045   0.085
date             3.870         3.808      0.881     0.062    0.16    0.153   0.580
cap            483.000       403.263    214.182    79.737   65.00  100.143 283.000
Summary of balance for matched data:
         Means Treated Means Control SD Control Mean Diff eQQ Med eQQ Mean eQQ Max
distance         0.291         0.292      0.070    -0.001       0    0.001   0.004
date             3.870         3.656      0.823     0.214       0    0.309   1.250
cap            483.000       493.429    195.677   -10.429       0   20.429  85.000
Percent Balance Improvement:
         Mean Diff. eQQ Med eQQ Mean eQQ Max
distance      95.81      100    97.29   94.91
date        -245.04      100  -101.87 -115.52
cap           86.92      100    79.60   69.97
Sample sizes:
          Control Treated
All            19       7
Matched         7       7
Unmatched      12       0
Discarded       0       0
```

Before and After matching

optimal match 1:1

maybe get better balance, may be not compared to ratio=2

← only asked for 1:1 ratio=1

```
> mopt.data = match.data(mopt1)
> mopt.data
   date cap pr distance weights subclass
2  2.33 665  0   0.3409       1        1
3  2.33 665  1   0.3409       1        1
4  3.00 665  0   0.3513       1        2
5  3.00 665  1   0.3513       1        2
9  3.42 422  1   0.2551       1        3
13 3.42 130  0   0.1601       1        4
14 3.92 650  0   0.3590       1        5
17 4.50 445  0   0.2787       1        7
18 3.42 130  1   0.1601       1        4
20 3.92 650  1   0.3590       1        5
21 3.75 513  0   0.2960       1        6
22 5.92 428  1   0.2917       1        6
23 4.67 386  0   0.2578       1        3
24 5.08 421  1   0.2770       1        7
```

create a matched data set "subclass" tells us the matches

ID

default distance is propensity score

```
> glmnuke = glm(pr ~ date + cap, family = binomial, data = bennuke)
> propen = fitted(glmnuke)
> propen
     1      2      3      4      5      6      7      8      9     10     11     12     13
0.2089 0.3409 0.3409 0.3513 0.3513 0.1514 0.2518 0.1414 0.2551 0.2427 0.1699 0.2473 0.1601
    14     15     16     17     18     19     20     21     22     23     24     25     26
0.3590 0.2727 0.2387 0.2787 0.1601 0.3817 0.3590 0.2960 0.2917 0.2578 0.2770 0.1819 0.4328
```

create the propensity score by hand

⊗ in mopt.data

```
> attach(mopt.data)
> names(mopt.data)
[1] "date"     "cap"      "pr"       "distance" "weights"  "subclass"
> tapply(distance, list(pr,subclass), mean)
        1      2      3      4      5      6      7
0 0.3409 0.3513 0.2578 0.1601 0.3590 0.2960 0.2787
1 0.3409 0.3513 0.2551 0.1601 0.3590 0.2917 0.2770
> tapply(cap, list(pr,subclass), mean)
    1   2   3   4   5   6   7
0 665 665 386 130 650 513 445
1 665 665 422 130 650 428 421
> tapply(date, list(pr,subclass), mean)
     1 2    3    4    5    6    7
0 2.33 3 4.67 3.42 3.92 3.75 4.50
1 2.33 3 3.42 3.42 3.92 5.92 5.08
```

you can do your own comparisons of balance after matching

cf love Aspirin ex 10% effect size rule

```
> mnear1= matchit(pr ~ date + cap, data = bennuke, method = "nearest")
> summary(mnear1)
```

compare with nearest neighbor 1:1 matching

```
Call:
matchit(formula = pr ~ date + cap, data = bennuke, method = "nearest")

Summary of balance for all data:
         Means Treated Means Control SD Control Mean Diff eQQ Med eQQ Mean eQQ Max
distance         0.291         0.261      0.083     0.029    0.04    0.045   0.085
date             3.870         3.808      0.881     0.062    0.16    0.153   0.580
cap            483.000       403.263    214.182    79.737   65.00  100.143 283.000


Summary of balance for matched data:
         Means Treated Means Control SD Control Mean Diff eQQ Med eQQ Mean eQQ Max
distance         0.291         0.292      0.070    -0.001       0    0.001   0.004
date             3.870         3.656      0.823     0.214       0    0.309   1.250
cap            483.000       493.429    195.677   -10.429       0   20.429  85.000

Percent Balance Improvement:
         Mean Diff. eQQ Med eQQ Mean eQQ Max
distance      95.81      100    97.29   94.91
date        -245.04      100  -101.87 -115.52
cap           86.92      100    79.60   69.97

Sample sizes:
          Control Treated
All            19       7
Matched         7       7
Unmatched      12       0
Discarded       0       0

> mnear1.data = match.data(mnear1)
> mnear1.data
   date cap pr distance weights
2  2.33 665  0   0.3409       1
3  2.33 665  1   0.3409       1
4  3.00 665  0   0.3513       1
5  3.00 665  1   0.3513       1
9  3.42 422  1   0.2551       1
13 3.42 130  0   0.1601       1
14 3.92 650  0   0.3590       1
17 4.50 445  0   0.2787       1
18 3.42 130  1   0.1601       1
20 3.92 650  1   0.3590       1
21 3.75 513  0   0.2960       1
22 5.92 428  1   0.2917       1
23 4.67 386  0   0.2578       1
24 5.08 421  1   0.2770       1

>
```

matched data show same result as optimal for this small example.

Rubin An Int Med Cochrans ex, matching on Age

Table 1. Comparison of Mortality Rates for Three Smoking Groups in Three Databases*

Variable	Canadian Study			United Kingdom Study			United States Study		
	Nonsmokers	Cigarette Smokers	Cigar and Pipe Smokers	Nonsmokers	Cigarette Smokers	Cigar and Pipe Smokers	Nonsmokers	Cigarette Smokers	Cigar and Pipe Smokers
Mortality rates per 1000 person-years, %	20.2	20.5	35.5	11.3	14.1	20.7	13.5	13.5	17.4
Average age, *y*	54.9	50.5	65.9	49.1	49.8	55.7	57.0	53.2	59.7
Adjusted mortality rates using subclasses, %									
2 subclasses	20.2	26.4	24.0	11.3	12.7	13.6	13.5	16.4	14.9
3 subclasses	20.2	28.3	21.2	11.3	12.8	12.0	13.5	17.7	14.2
9–11 subclasses	20.2	29.5	19.8	11.3	14.8	11.0	13.5	21.2	13.7

* Adapted from Tables 1–3 in Cochran (2).

subclassification p757 Rubin for bias reduction

```
>  mfull= matchit(pr ~ date + cap, data = bennuke, method = "full")
> summary(mfull)
Call: matchit(formula = pr ~ date + cap, data = bennuke, method = "full")
Summary of balance for all data:
         Means Treated Means Control Mean Diff eQQ Med eQQ Mean  eQQ Max
distance        0.2907        0.2613    0.0294  0.0398   0.0454   0.0852
date            3.8700        3.8079    0.0621  0.1600   0.1529   0.5800
cap           483.0000      403.2632   79.7368 65.0000 100.1429 283.0000
Summary of balance for matched data:
         Means Treated Means Control Mean Diff eQQ Med eQQ Mean  eQQ Max
distance        0.2907        0.2939   -0.0032  0.0043   0.0113   0.0738
date            3.8700        3.6312    0.2388  0.1600   0.3107   1.4200
cap           483.0000      496.5905  -13.5905 23.0000  35.0232 261.0000
Percent Balance Improvement:
         Mean Diff. eQQ Med  eQQ Mean   eQQ Max
distance    89.2703 89.1190   75.2162   13.3177
date      -284.4471  0.0000 -103.2460 -144.8276
cap         82.9558 64.6154   65.0268    7.7739
Sample sizes:
          Control Treated
All            19       7
Matched        19       7
Unmatched       0       0
Discarded       0       0

> mfull.data = match.data(mfull)        > mfull.data
   date cap pr  distance   weights subclass
1  3.58 287  0 0.2089340 0.4523810        3
2  2.33 665  0 0.3408701 2.7142857        1
3  2.33 665  1 0.3408701 1.0000000        1
4  3.00 665  0 0.3513216 2.7142857        2
5  3.00 665  1 0.3513216 1.0000000        2
6  2.92 114  0 0.1513795 0.5428571        4
7  3.17 422  0 0.2518308 0.4523810        3
8  3.42  57  0 0.1414043 0.5428571        4
9  3.42 422  1 0.2550915 1.0000000        3
10 3.33 392  0 0.2426859 0.4523810        3
11 3.58 160  0 0.1699374 0.5428571        4
12 3.75 390  0 0.2472990 0.4523810        3
13 3.42 130  0 0.1601291 0.5428571        4
14 3.92 650  0 0.3589552 0.9047619        5
15 3.92 450  0 0.2726900 1.3571429        7
16 3.42 378  0 0.2386883 0.4523810        3
17 4.50 445  0 0.2786707 1.3571429        7
18 3.42 130  1 0.1601291 1.0000000        4
19 4.17 690  0 0.3816748 0.9047619        5
20 3.92 650  1 0.3589552 1.0000000        5
21 3.75 513  0 0.2960073 2.7142857        6
22 5.92 428  1 0.2916721 1.0000000        6
23 4.67 386  0 0.2577539 0.4523810        3
24 5.08 421  1 0.2770347 1.0000000        7
25 5.42 138  0 0.1819249 0.5428571        4
26 6.08 730  0 0.4327688 0.9047619        5
> attach(mfull.data)
> table(subclass)
subclass
1 2 3 4 5 6 7
2 2 7 6 4 2 3
> tapply(distance, list(pr,subclass), mean)
          1         2         3         4         5         6         7
0 0.3408701 0.3513216 0.2411986 0.1609550 0.3911329 0.2960073 0.2756804
1 0.3408701 0.3513216 0.2550915 0.1601291 0.3589552 0.2916721 0.2770347
> tapply(cap, list(pr,subclass), mean)         > tapply(date, list(pr,subclass), mean)
    1   2        3     4   5   6     7             1 2        3     4        5    6    7
0 665 665 375.8333 119.8 690 513 447.5         0 2.33 3 3.653333 3.752 4.723333 3.75 4.21
1 665 665 422.0000 130.0 650 428 421.0         1 2.33 3 3.420000 3.420 3.920000 5.92 5.08
```

Example # 2: Gender equity study for research scientists[1]

For HW

Women and men scientists are to be matched on grant funding.

Women		Men	
Subject	$\log_{10}$(Grant)	Subject	$\log_{10}$(Grant)
A	5.7	V	5.5
B	4.0	W	5.3
C	3.4	X	4.9
D	3.1	Y	4.9
		Z	3.9

[1] Discussed in Hansen and Klopfer (2006), Hansen (2004)

Week 8

p2 stat209) Ben Hansen, Gender Equity Example

```
> geneq  = read.table(file="D:\\drr08\\stat209\\week8\\genderdata", header = T)
> geneq
  log10Grant gender
1        5.7      W
2        4.0      W
3        3.4      W
4        3.1      W
5        5.5      M
6        5.3      M
7        4.9      M
8        4.9      M
9        3.9      M

# had some problems matchit; outcome has to be 1=W, 0=M or do a as.numeric
> #outcome has to be numeric categorical, not W/M

> mfullgen= matchit(gender ~ Grant, data = geneq, method = "full")
> summary(mfullgen)
Call:
matchit(formula = gender ~ Grant, data = geneq, method = "full")

Summary of balance for all data:
         Means Treated Means Control Mean Diff eQQ Med eQQ Mean eQQ Max
distance         0.576          0.339     0.237   0.275    0.252   0.421
Grant            4.050          4.900    -0.850   0.850    0.850   1.500

Summary of balance for matched data:
         Means Treated Means Control Mean Diff eQQ Med eQQ Mean eQQ Max
distance         0.576          0.532     0.045   0.129    0.121   0.463
Grant            4.050          4.213    -0.163   0.500    0.507   1.800

Percent Balance Improvement:
         Mean Diff. eQQ Med eQQ Mean eQQ Max
distance      81.18    53.10    51.96  -9.927
Grant         80.88    41.18    40.37 -20.000

Sample sizes:
          Control Treated
All             5       4
Matched         5       4
Unmatched       0       0
Discarded       0       0

> mfullgen.dat = match.data(mfullgen)
> mfullgen.dat
  Grant gender distance weights subclass
1   5.7      1   0.1561  1.0000        1
2   4.0      1   0.5903  1.0000        2
3   3.4      1   0.7484  1.0000        2
4   3.1      1   0.8103  1.0000        2
5   5.5      0   0.1907  0.3125        1
6   5.3      0   0.2307  0.3125        1
7   4.9      0   0.3271  0.3125        1
8   4.9      0   0.3271  0.3125        1
9   3.9      0   0.6192  3.7500        2
> #corresponds to slide, subclass
```

second Hansen talk ex. do in HW 8

see Hansen talk

An Optimal Full Matching for the Gender Equity Example

Women		Men	
Subject	$\log_{10}$(Grant)	Subject	$\log_{10}$(Grant)
A	5.7	V	5.5
B	4.0	W	5.3
C	3.4	X	4.9
D	3.1	Y	4.9
		Z	3.9

As compared to the optimal 1:(1 or 2) match, full matching:

- decreases the largest discrepancy from 1.5 to 0.8; and
- decreases the sum of discrepancies from 3.8 to 3.6.

In global terms, it gives a tighter match. In local terms, it gives a *much* tighter match.

`help.matchit` *HTML Help for Matchit Commands and Models*

Description

The `help.matchit` command launches html help for Matchit commands and supported methods. The full manual is available online at `http://gking.harvard.edu/matchit`.

Usage

```
help.matchit(object)
```

Arguments

`object`	a character string representing a Matchit command or model. `help.matchit("command")` will take you to an index of Matchit commands and `help.matchit("method")` will take you to a list of matching methods. The following inputs are currently available: `exact`, `nearest`, `subclass`, `full`, `optimal`.

Author(s)

Daniel Ho <⟨daniel.ho@yale.edu⟩>; Kosuke Imai <⟨kimai@princeton.edu⟩>; Gary King <⟨king@harvard.edu⟩>; Elizabeth Stuart<⟨stuart@stat.harvard.edu⟩>

See Also

The complete document is available online at `http://gking.harvard.edu/matchit`.

Lab 4 data for matching using MatchIt Is job training effective???

`lalonde` *Data from National Supported Work Demonstration and PSID, as analyzed by Dehejia and Wahba (1999).*

Description

This is a subsample of the data from the treated group in the National Supported Work Demonstration (NSW) and the comparison sample from the Current Population Survey (CPS). This data was previously analyzed extensively by Lalonde (1986) and Dehejia and Wahba (1999). The full dataset is available at `http://www.columbia.edu/~rd247/nswdata.html`.

Usage

```
data(lalonde)
```

From Lab 4

> dim(lalonde)□□[1] 614 10□□

> names(lalonde) "treat" "age" "educ" "black" "hispan" "married" "nodegree" "re74" "re75" "re78"

> attach(lalonde) > table(treat) treat□□ 0 1 □□429 185 □□

> lalonde[1:10,] treat age educ black hispan married nodegree re74 re75 re78

Format

614 actually

A data frame with 313 observations (185 treated, 429 control). There are 10 variables measured for each individual. "treat" is the treatment assignment (1=treated, 0=control). "age" is age in years. "educ" is education in number of years of schooling. "black" is an indicator for African-American (1=African-American, 0=not). "hispan" is an indicator for being of Hispanic origin (1=Hispanic, 0=not). "married" is an indicator for married (1=married, 0=not married). "nodegree" is an indicator for whether the individual has a high school degree (1=no degree, 0=degree). "re74" is income in 1974, in U.S. dollars. "re75" is income in 1975, in U.S. dollars. "re78" is income in 1978, in U.S. dollars.

Source

http://www.columbia.edu/~rd247/nswdata.html

References

Lalonde, R. (1986). Evaluating the econometric evaluations of training programs with experimental data. *American Economic Review* 76: 604-620. \

Dehejia, R.H. and Wahba, S. (1999). Causal Effects in Nonexperimental Studies: Re-Evaluating the Evaluation of Training Programs. *Journal of the American Statistical Association* 94: 1053-1062.

`match.data` *Output Matched Data Sets*

Description

`match.data` outputs matched data sets from `matchit()`.

Usage

```
match.data <- match.data(object, group="all", distance = "distance",
weights = "weights", subclass = "subclass")
```

Arguments

`object`	The output object from `matchit()`. This is a required input.
`group`	This argument specifies for which matched group the user wants to extract the data. Available options are `"all"` (all matched units), `"treat"` (matched units in the treatment group), and `"control"` (matched units in the control group). The default is `"all"`.
`distance`	This argument specifies the variable name used to store the distance measure. The default is `"distance"`.
`weights`	This argument specifies the variable name used to store the resulting weights from matching. The default is `"weights"`.
`subclass`	This argument specifies the variable name used to store the subclass indicator. The default is `"subclass"`.

Value

Returns a subset of the original data set sent to this-is-escaped-code{, with ju

The Lalonde Data

For all of our examples, we use data from the job training program analyzed in Lalonde (1986) and Dehejia & Wahba (1999). A subsample of the data consisting of the National Supported Work Demonstration (NSW) treated group and the comparison sample from the Population Survey of Income Dynamics (PSID) is included in MATCHIT, and the full dataset is available at `http://www.columbia.edu/~rd247/nswdata.html`.[5]

The variables in this dataset are in Table 1 below. One causal effect of interest is the impact that participation in the job training program, `treat==1`, had on real earnings in 1978, `re78`, for those that participated in the program, i.e., the average treatment effect on the treated (ATT):

(1)

where `re78(treat=1)` represents the potential outcome under the treatment of the job program, and `re78(treat=0)` under control. To be clear, note that the first term (inside the expectation) in Equation 1 is *observed*, whereas the second term is the *unobserved* counterfactual of real earnings if participants had not participated. The nature of causal inference is that one of the two terms in the difference will always be unobserved. The same expression of the ATT, in mathematical notation is:

(2)

Table 1: Description of Lalonde data

Name	Description
Outcome ( )	
`re78`	Real earnings (1978)
Treatment Indicator ( )	
`treat`	Treated in job training program from March 1975-June 1977 (1 if treated, 0 if not treated)
Pre-treatment Covariates ( )	
`age`	Age
`educ`	Years of education
`black`	Race black (1 if black, 0 otherwise)
`hispan`	Race hispanic (1 if Hispanic, 0 otherwise)
`married`	Marital status (1 if married, 0 otherwise)
`nodegree`	High school degree (1 if no degree, 0 otherwise)
`re74`	Real earnings (1974)
`re75`	Real earnings (1975)

Gary King 2005-03-09

```
R version 3.2.2 (2015-08-14) -- "Fire Safety"  #### Week 1 session. Lalonde data
# If you start from a relatively clean install, get MatchIt and optmatch
# some years order matters because of complication with license for optmatch algorithms this year appea

> install.packages("optmatch")
> library(optmatch)
> install.packages("MatchIt")
> library(MatchIt)

################################
> data(lalonde) # in MatchIt package
# get lalonde data from MatchIt, there are different versions in existence under this name
help(lalonde)  #produces
-----------------------------------
lalonde               package:MatchIt               R Documentation

Data from National Supported Work Demonstration and PSID, as analyzed by Dehejia and Wahba (1999).

Description:

     This is a subsample of the data from the treated group in the
     National Supported Work Demonstration (NSW) and the comparison
     sample from the Current Population Survey (CPS).   This data was
     previously analyzed extensively by Lalonde (1986) and Dehejia and
     Wahba (1999).   The full dataset is available at <URL:
     http://www.columbia.edu/~rd247/nswdata.html>. [note: broken link still in current documentation]

Usage:

     data(lalonde)

Format:

     A data frame with 313 [sic, 614] observations (185 treated, 429 control).
     There are 10 variables measured for each individual.  "treat" is
     the treatment assignment (1=treated, 0=control).   "age" is age in
     years.  "educ" is education in number of years of schooling.
     "black" is an indicator for African-American (1=African-American,
     0=not).  "hispan" is an indicator for being of Hispanic origin
     (1=Hispanic, 0=not).  "married" is an indicator for married
     (1=married, 0=not married).  "nodegree" is an indicator for
     whether the individual has a high school degree (1=no degree,
     0=degree). "re74" is income in 1974, in U.S. dollars.  "re75" is
     income in 1975, in U.S. dollars.  "re78" is income in 1978, in
     U.S. dollars.


References:

     Lalonde, R. (1986). Evaluating the econometric evaluations of
     training programs  with experimental data. American Economic
     Review 76: 604-620.

     Dehejia, R.H. and Wahba, S. (1999).  Causal Effects in
     Nonexperimental Studies:  Re-Evaluating the Evaluation  of
     Training Programs.  Journal of the American  Statistical
     Association 94: 1053-1062.
-----------------------------------------------
> dim(lalonde)
[1] 614  10
> attach(lalonde)
> table(treat)  # so these summaries synch with data description
treat
  0   1
429 185
```

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

################## prelim compare groups on outcome measure
> tapply(re78, treat, median)
       0        1
4975.505 4232.309
> tapply(re78, treat, fivenum)
$`0`
[1]     0.0000   220.1813  4975.5050 11688.8200 25564.6700
$`1`
[1]     0.0000   485.2298  4232.3090  9642.9990 60307.9300

> t.test(re78 ~ treat)
        Welch Two Sample t-test
data:  re78 by treat
t = 0.93773, df = 326.41, p-value = 0.3491
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -697.192 1967.244
sample estimates:
mean in group 0 mean in group 1
       6984.170        6349.144

> wilcox.test(re78 ~ treat, conf.int = TRUE)
        Wilcoxon rank sum test with continuity correction
data:  re78 by treat
W = 41840, p-value = 0.2818
alternative hypothesis: true location shift is not equal to 0
95 percent confidence interval:
 -4.664401e-05  1.053159e+03
sample estimates:
difference in location
          5.053114e-05

> #####But wait, some say "we are never done until the ancova is run" see Fish
> # as we see the social science, life science practice is to put in the treatment variable  and
> # a whole bunch of other variables to "control" for self-selection, nonequivalence etc.
> #  equivalent to analysis of covariance by whatever name
>  ancova.lalonde = lm( re78 ~ treat + age + educ + black + hispan + married + nodegree + re74 + re75)
>  summary(ancova.lalonde)
Call:
lm(formula = re78 ~ treat + age + educ + black + hispan + married +
    nodegree + re74 + re75)

Residuals:
   Min     1Q Median     3Q    Max
-13595  -4894  -1662   3929  54570

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.651e+01  2.437e+03   0.027   0.9782
treat        1.548e+03  7.813e+02   1.982   0.0480 *
age          1.298e+01  3.249e+01   0.399   0.6897
educ         4.039e+02  1.589e+02   2.542   0.0113 *
black       -1.241e+03  7.688e+02  -1.614   0.1071
hispan       4.989e+02  9.419e+02   0.530   0.5966
married      4.066e+02  6.955e+02   0.585   0.5590
nodegree     2.598e+02  8.474e+02   0.307   0.7593
re74         2.964e-01  5.827e-02   5.086 4.89e-07 ***
```

Standard Analysis (ancova)

OUTCOME ~ TREATMENT + (binary, contin)

CONFOUNDERS (controls)

see FISH (in the news) example

```
re75         2.315e-01  1.046e-01   2.213   0.0273 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 6948 on 604 degrees of freedom
Multiple R-squared:  0.1478,    Adjusted R-squared:  0.1351
F-statistic: 11.64 on 9 and 604 DF,  p-value: < 2.2e-16

> # so treatment is significantly helpful ??
```

First by hand, then by algorithm

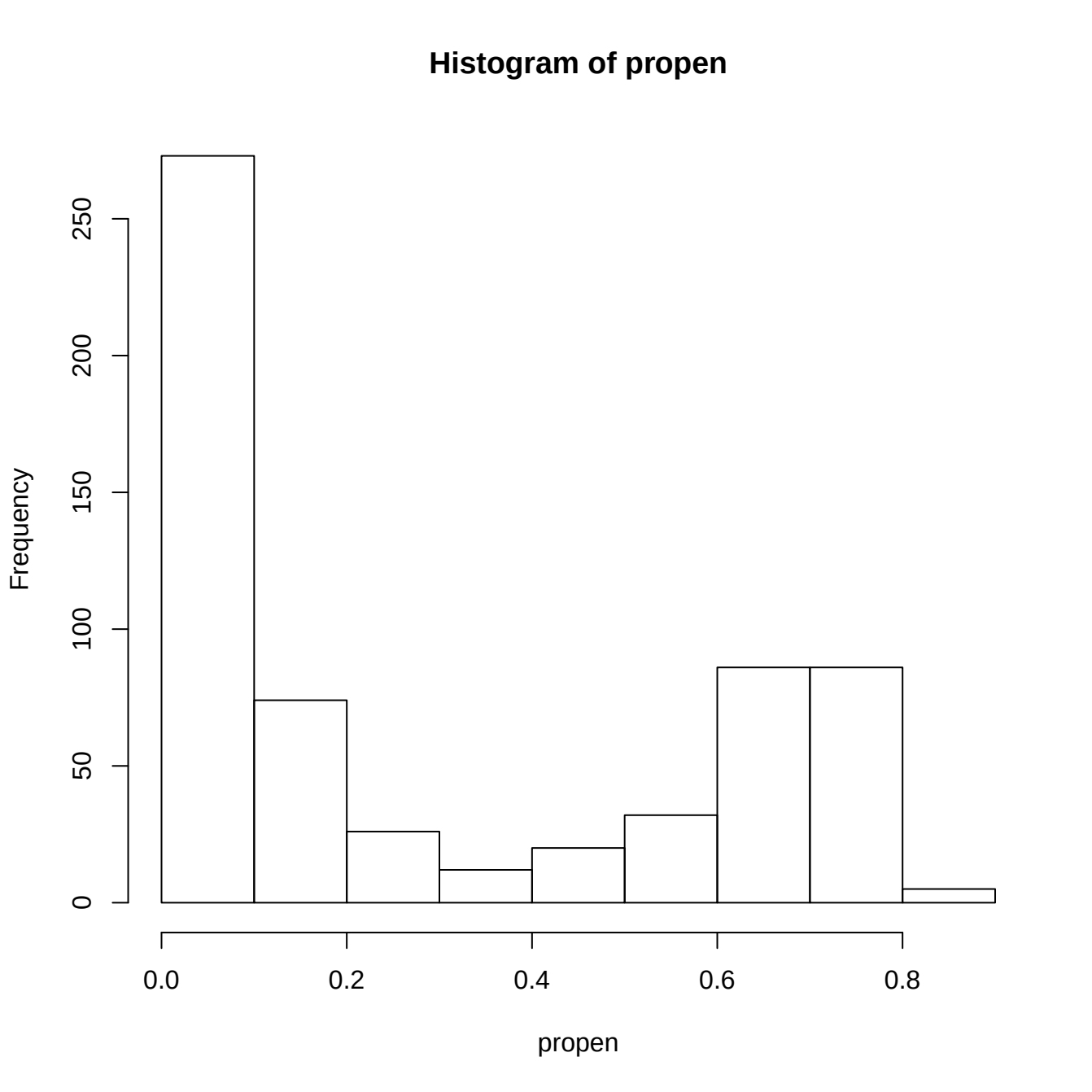
```
########### Begin matching analysis; Quintile Subclassification with Propensity Scores
## original Rosenbaum-Rubin, cardiac; Rubin breast cancer

> # now do the logistic regression that computes propensity scores (matching packages will do this for
>  glm.p = glm( treat ~ age + educ + black + hispan + married + nodegree + re74 + re75, data = lalonde,
>  summary(glm.p)
Call:
glm(formula = treat ~ age + educ + black + hispan + married +
    nodegree + re74 + re75, family = binomial, data = lalonde)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7645  -0.4736  -0.2862   0.7508   2.7169

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.729e+00  1.017e+00  -4.649 3.33e-06 ***
age          1.578e-02  1.358e-02   1.162  0.24521
educ         1.613e-01  6.513e-02   2.477  0.01325 *
black        3.065e+00  2.865e-01  10.699  < 2e-16 ***
hispan       9.836e-01  4.257e-01   2.311  0.02084 *
married     -8.321e-01  2.903e-01  -2.866  0.00415 **
nodegree     7.073e-01  3.377e-01   2.095  0.03620 *
re74        -7.178e-05  2.875e-05  -2.497  0.01253 *
re75         5.345e-05  4.635e-05   1.153  0.24884
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
(Dispersion parameter for binomial family taken to be 1)
    Null deviance: 751.49  on 613  degrees of freedom
Residual deviance: 487.84  on 605  degrees of freedom
AIC: 505.84
Number of Fisher Scoring iterations: 5

> propen = fitted(glm.p)  # now we have the propensity scores
> quantile(propen) # overall distrib
         0%          25%          50%          75%         100%
0.009080193 0.048536484 0.120676493 0.638715991 0.853152844
> tapply(propen, treat, quantile) # look at overlap via 5-number summary (or side-by-side boxplots) not
$`0`
         0%          25%          50%          75%         100%
0.009080193 0.038880745 0.075849106 0.195135746 0.789172834
$`1`
        0%         25%         50%         75%        100%
0.02495179 0.52646352 0.65368426 0.72659995 0.85315284

> # as we are fitting prob(treat = 1) fits for those in treatment group will be larger, we need good ov
> boxplot(propen ~ treat) #gives side-by-side boxplots, you can add labels, not wonderful overlap
> detach(lalonde)
> lalonde$propen = propen
> attach(lalonde)

#### looking at overlap, histograms
>  p1 = propen[treat == 1]
>  length(p1)
[1] 185
>  p0 = propen[treat == 0]
```

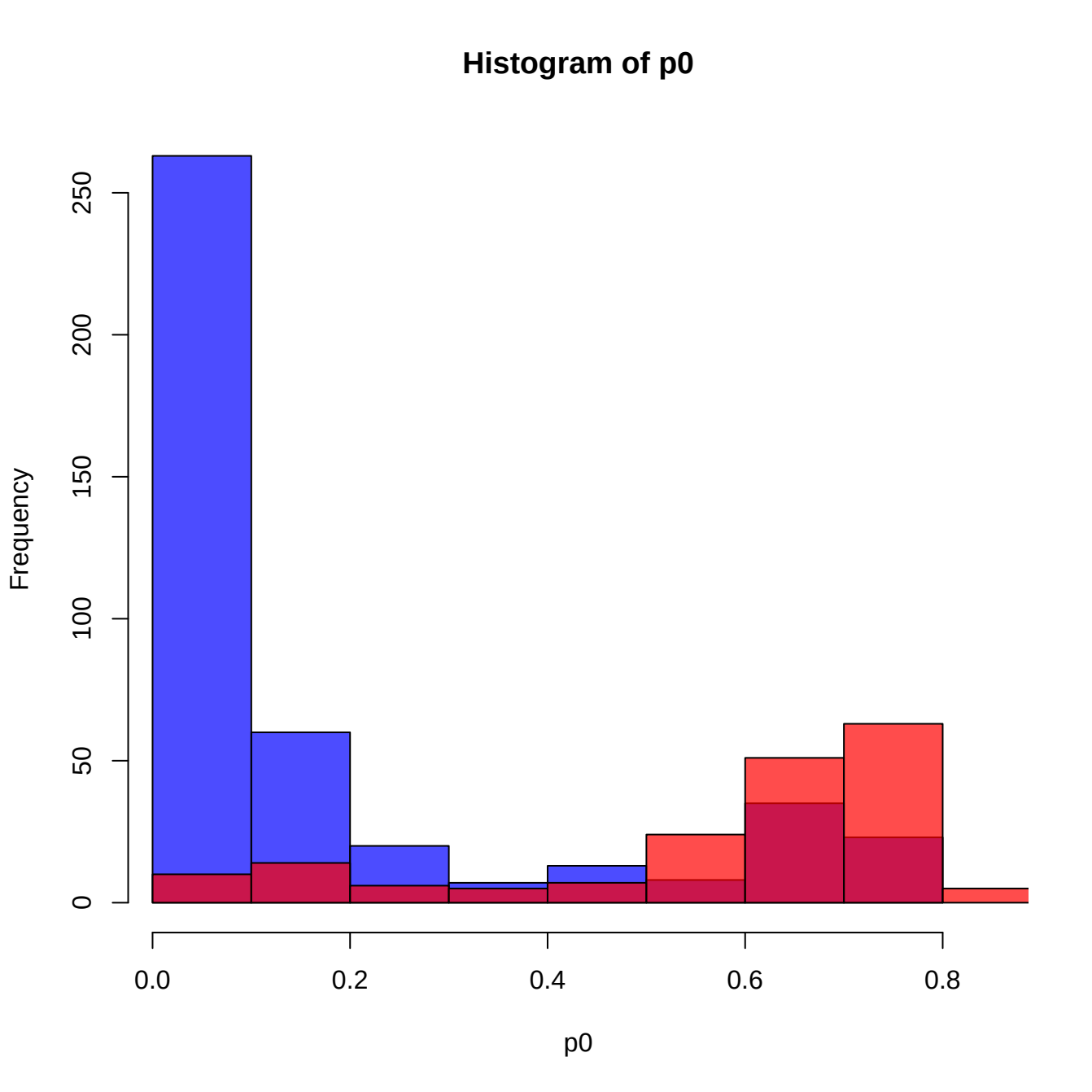
LAB 4 excerpt

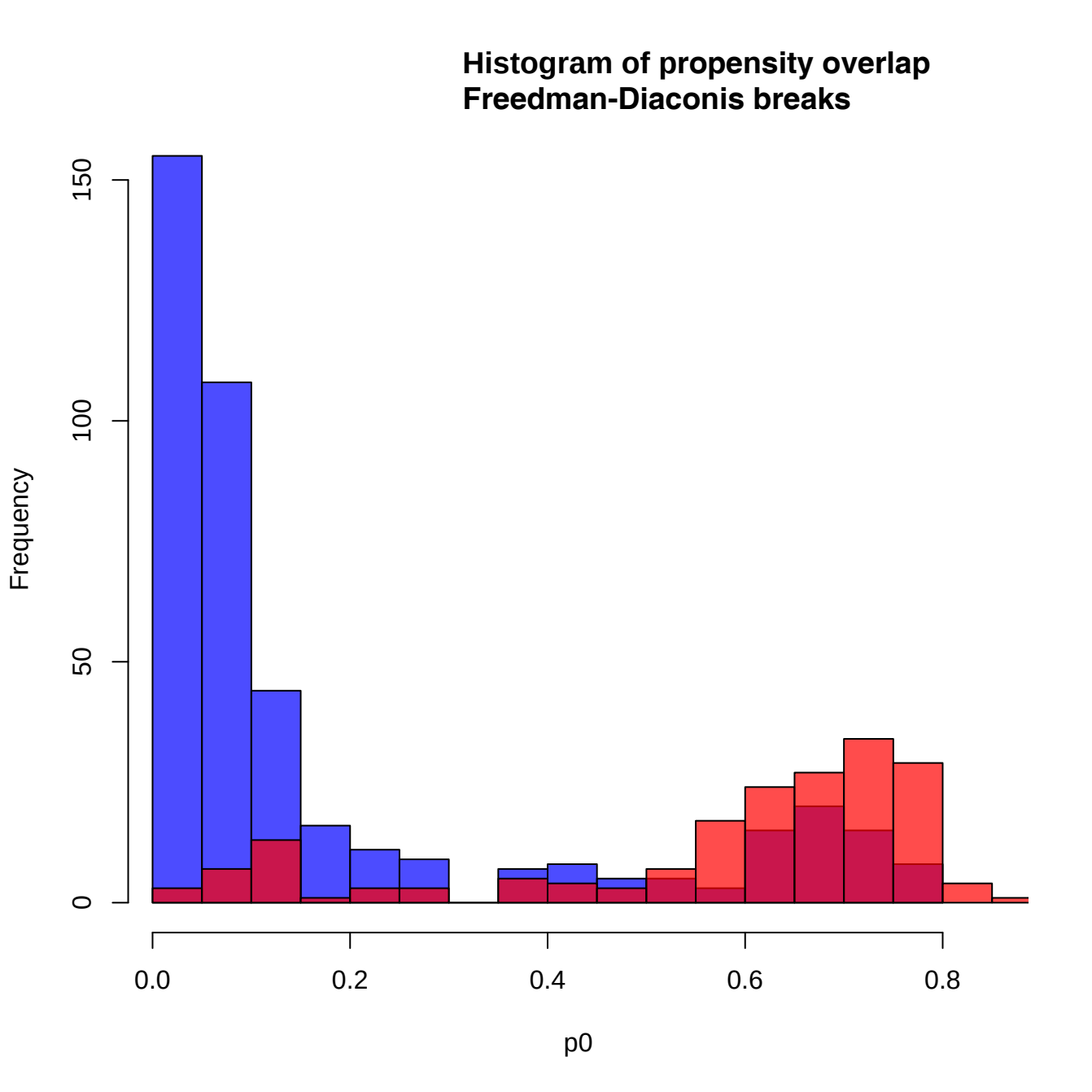
```
# now do the logistic regression that computes propensity scores (matching packages will do this for
> glm.lalonde = glm( treat ~ age + educ + black + hispan + married + nodegree + re74 + re75,
+ data = lalonde, family = binomial)
> propen = fitted(glm.lalonde)  # now we have the propensity scores, Lab script calls these propScore
> tapply(propen, treat, quantile) # look at overlap via 5-number summary (or side-by-side boxplots)
                                  not real good overlap, as noted in class handout
$`0`
     0%     25%     50%     75%    100%
0.00908 0.03888 0.07585 0.19514 0.78917

$`1`
     0%     25%     50%     75%    100%
0.02495 0.52646 0.65368 0.72660 0.85315

> # the common use of the propensity scores (backed by theory, class handout 2/26))
> # is to stratify by quintiles

> # the simple-minded way I do it is to use "cut", Lab script is fancier programming
> ?cut  # this is a simple function to create bins
> k = 1:4
> quantile(propen, k/5)
    20%     40%     60%     80%
0.04015 0.08721 0.26978 0.67085
>  propbin = cut(propen, c(0, .04015,.08721,.26978,.67085,1))


> table(propbin, treat) # either way you display it, we do not have good overlap in the bottom
                        two quintiles, lower estimated probability for being in treatment
                        for treatment cases
                 treat
propbin             0   1
  (0,0.0401]      122   1
  (0.0401,0.0872] 116   7
  (0.0872,0.27]   101  21
  (0.27,0.671]     53  71
  (0.671,1]        37  85


> tapply(re78, list(propbin, treat),mean)  # here are the mean diffs in re78 the outcome
                                           stratified by propensity quintile
# direction of mean diffs favors treatment, job training
                     0    1
(0,0.0401]       10467    0
(0.0401,0.0872]  5797 7919
(0.0872,0.27]    6043 9211
(0.27,0.671]     4977 5819
(0.671,1]        4666 6030


> t.test(re78[propbin == bins[5]] ~ treat[propbin == bins[5]]) # t-test for quintile 5

etc
```

0.0
0.2
0.4
0.6
0.8
0
1

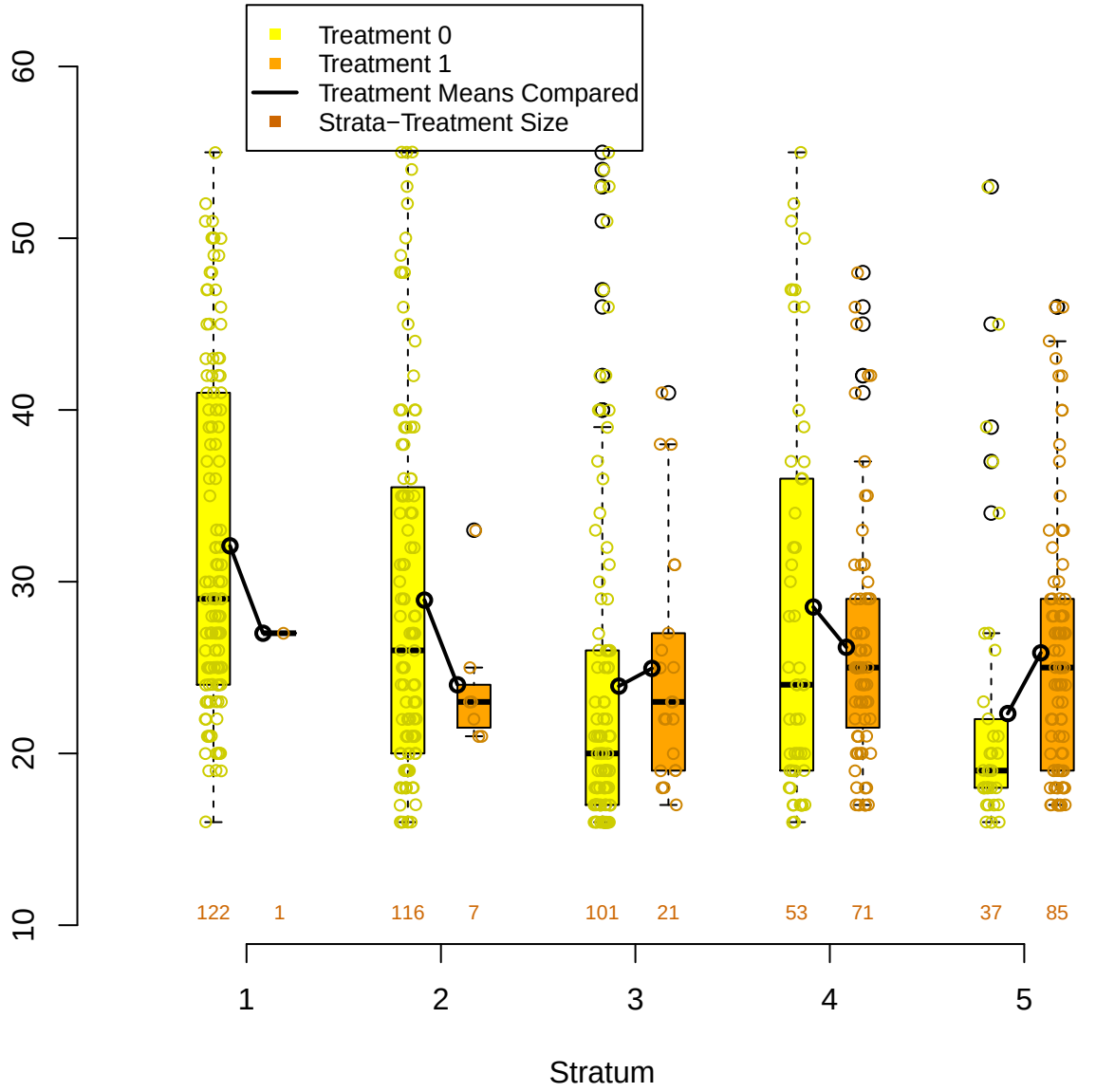
```
>  length(p0)
[1] 429
>  fivenum(p1)
    NSW124     NSW156      NSW50     NSW119     NSW178 
0.02495179 0.52646352 0.65368426 0.72659995 0.85315284 
>  fivenum(p0)
    PSID296     PSID347     PSID221     PSID334     PSID118 
0.009080193 0.038880745 0.075849106 0.195135746 0.789172834 
>  hist(p0,col=rgb(0,0,1,0.7),xlim=range(c(p0,p1)))
>  hist(p1,col=rgb(1,0,0,0.7),add=T)
>  # superimposed propensity histograms, like Ben Hansen SAT, contol is blue, treatment is red, overlap
> hist(p0, breaks = "FD", col=rgb(0,0,1,0.7),xlim=range(c(p0,p1)))
> hist(p1, breaks = "FD", col=rgb(1,0,0,0.7),add=T)

### make quintiles of propensity distribution
>  pbin = cut(propen, quantile(propen, seq(0, 1, 1/5)), include.lowest = TRUE, labels = FALSE)
> detach(lalonde)
> lalonde$bins = pbin
> attach(lalonde)
>  table(pbin, treat)
    treat
pbin   0   1
   1 122   1
   2 116   7
   3 101  21
   4  53  71
   5  37  85

##### examples of checking balance (more to come)
> tapply(age, list(bins, treat), median)
   0  1
1 29 27
2 26 23
3 20 23
4 24 25
5 19 25

> ### install.packages("PSAgraphics")
>  library(PSAgraphics)
>  box.psa(age, treat, bins)

#################################### examine outcome by strata
> tapply(re78, list(bins, treat),mean)  # here are the mean diffs in re78 (the outcome) stratified by p
          0        1
1 10467.064    0.000
2  5796.548 7919.316
3  6043.316 9210.726
4  4977.401 5819.143
5  4666.221 6030.258
> # direction of mean diffs favors treatment, job training

> # contrast that with the comparison ignoring any concerns about self-selection (selection bias), effe
>  tapply(re78, treat, mean)
       0        1 
6984.170 6349.144 

> ##### can do t-tests by subclassification (strata)
> # e.g. for the 3 upper quintiles is the mean difference significant? since we are doing 3 of these be
> ##  we won't find any evidence for the effectiveness of job training looking at each of the subclasse
>
> ##### lmer, a better way to do the t-tests  #######################
> library(lme4)
Loading required package: Matrix
> propen.lmer = lmer(re78 ~ treat + (1 + treat|bins), data = lalonde)
> summary(propen.lmer)
Linear mixed model fit by REML ['lmerMod']
```


Histogram of propen
Frequency
0
50
100
150
200
250
0.0
0.2
0.4
0.6
0.8
propen


Histogram of p0
Frequency
0
50
100
150
200
250
0.0
0.2
0.4
0.6
0.8
p0


Histogram of propensity overlap
Freedman-Diaconis breaks
Frequency
0
50
100
150
0.0
0.2
0.4
0.6
0.8
p0

from PSAgraphics, in Lab4



```
>  length(p0)
[1] 429
>  fivenum(p1)
    NSW124     NSW156      NSW50     NSW119     NSW178
0.02495179 0.52646352 0.65368426 0.72659995 0.85315284
>  fivenum(p0)
    PSID296     PSID347     PSID221     PSID334     PSID118
0.009080193 0.038880745 0.075849106 0.195135746 0.789172834
>  hist(p0,col=rgb(0,0,1,0.7),xlim=range(c(p0,p1)))
>  hist(p1,col=rgb(1,0,0,0.7),add=T)
>  # superimposed propensity histograms, like Ben Hansen SAT, contol is blue, treatment is red, overlap
> hist(p0, breaks = "FD", col=rgb(0,0,1,0.7),xlim=range(c(p0,p1)))
> hist(p1, breaks = "FD", col=rgb(1,0,0,0.7),add=T)

### make quintiles of propensity distribution
>  pbin = cut(propen, quantile(propen, seq(0, 1, 1/5)), include.lowest = TRUE, labels = FALSE)
> detach(lalonde)
> lalonde$bins = pbin
> attach(lalonde)
>  table(pbin, treat)
    treat
pbin   0   1
   1 122   1
   2 116   7
   3 101  21
   4  53  71
   5  37  85

##### examples of checking balance (more to come)
> tapply(age, list(bins, treat), median)
   0  1
1 29 27
2 26 23
3 20 23
4 24 25
5 19 25

> ### install.packages("PSAgraphics")
>  library(PSAgraphics)
>  box.psa(age, treat, bins)

#################################### examine outcome by strata
> tapply(re78, list(bins, treat),mean)  # here are the mean diffs in re78 (the outcome) stratified by p
         0        1
1 10467.064    0.000
2  5796.548 7919.316
3  6043.316 9210.726
4  4977.401 5819.143
5  4666.221 6030.258
> # direction of mean diffs favors treatment, job training

> # contrast that with the comparison ignoring any concerns about self-selection (selection bias), effe
>  tapply(re78, treat, mean)
       0        1
6984.170 6349.144

> ##### can do t-tests by subclassification (strata)
> # e.g. for the 3 upper quintiles is the mean difference significant? since we are doing 3 of these be
> ##  we won't find any evidence for the effectiveness of job training looking at each of the subclasse
>
> ##### lmer, a better way to do the t-tests  ########################
> library(lme4)
Loading required package: Matrix
> propen.lmer = lmer(re78 ~ treat + (1 + treat|bins), data = lalonde)
> summary(propen.lmer)
Linear mixed model fit by REML ['lmerMod']
```

```
Formula: re78 ~ treat + (1 + treat | bins)
   Data: lalonde

REML criterion at convergence: 12637.1

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.3976 -0.7541 -0.2878  0.5408  7.4535

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 bins     (Intercept)  5208943 2282
          treat        2069963 1439     -1.00
 Residual             52597981 7252
Number of obs: 614, groups:  bins, 5

Fixed effects:
            Estimate Std. Error t value
(Intercept)   6434.2     1090.2   5.902
treat          385.7      950.8   0.406

Correlation of Fixed Effects:
      (Intr)
treat -0.795

# so here we have an overall estimate of the effect of the treat on re78 of positive $386, but
# far from significant. Much smaller point estimate than in some of the individual strata

> confint(propen.lmer) # bombs
> confint(propen.lmer, method = "boot", nsim = 1000, boot.type = "perc")
Computing bootstrap confidence intervals ...

                  2.5 %   97.5 %
.sig01        414.81230 4084.578
.sig02         -1.00000    1.000
.sig03         54.74858 3644.981
.sigma       6846.49101 7654.434
(Intercept)  4432.91940 8695.198
treat       -1681.75647 2565.802

  some bootstrap runs failed (7/1000)
```

```
########################## Full Matching (Hansen, via Rosenbaum, using MatchIt)

>  m2full.out = matchit(treat ~ re74 + re75 + educ + black + hispan + age + married + nodegree, data =
Warning message:
```

```
#let's try Ben's full matching with all the vars; sould also compare with propensity in part
> m2fullvars.out = matchit(treat ~ re74 + re75 + educ + black + hispan + age + married + nod
+ data = lalonde, method = "full")
> m2fullvars.out
Call:
matchit(formula = treat ~ re74 + re75 + educ + black + hispan +
    age + married + nodegree, data = lalonde, method = "full")

Sample sizes:
          Control Treated
All           429     185
Matched       429     185
Unmatched       0       0
Discarded       0       0

> summary(m2fullvars.out)
Call:
matchit(formula = treat ~ re74 + re75 + educ + black + hispan +
    age + married + nodegree, data = lalonde, method = "full")

Summary of balance for all data:
         Means Treated Means Control Mean Diff  eQQ Med eQQ Mean  eQQ Max
distance         0.577         0.182     0.395    0.518    0.396    0.597
re74          2095.574      5619.237 -3523.663 2425.572 3620.924 9216.500
re75          1532.055      2466.484  -934.429  981.097 1060.658 6795.010
educ            10.346        10.235     0.111    1.000    0.703    4.000
black            0.843         0.203     0.640    1.000    0.643    1.000
hispan           0.059         0.142    -0.083    0.000    0.081    1.000
age             25.816        28.030    -2.214    1.000    3.265   10.000
married          0.189         0.513    -0.324    0.000    0.324    1.000
nodegree         0.708         0.597     0.111    0.000    0.114    1.000

Summary of balance for matched data:
         Means Treated Means Control Mean Diff eQQ Med eQQ Mean   eQQ Max
distance         0.577         0.576     0.001   0.003    0.006     0.087
re74          2095.574      2434.869  -339.295 311.523  659.367 13121.750
re75          1532.055      1577.728   -45.672 205.887  468.549 12746.050
educ            10.346        10.442    -0.096   0.000    0.392     4.000
black            0.843         0.835     0.009   0.000    0.000     1.000
hispan           0.059         0.061    -0.001   0.000    0.002     1.000
age             25.816        24.707     1.110   3.000    3.141     9.000
married          0.189         0.131     0.058   0.000    0.044     1.000
nodegree         0.708         0.695     0.013   0.000    0.011     1.000

Percent Balance Improvement:
         Mean Diff. eQQ Med eQQ Mean eQQ Max
distance      99.64   99.50   98.493   85.49
re74          90.37   87.16   81.790  -42.37
re75          95.11   79.02   55.825  -87.58
educ          13.08  100.00   44.158    0.00
black         98.66  100.00   99.938    0.00
hispan        98.26    0.00   98.027    0.00
age           49.88 -200.00    3.788   10.00
married       82.06    0.00   86.557    0.00
nodegree      88.53    0.00   90.133    0.00

Sample sizes:
          Control Treated
All           429     185
Matched       429     185
```

```
Unmatched       0       0
Discarded       0       0

#### do the outcome re78 analysis
> m2fullvars.out = matchit(treat ~ re74 + re75 + educ + black + hispan + age + married + nod

> m2full.dat = match.data(m2fullvars.out)
> head(m2full.dat)
     treat age educ black hispan married nodegree re74 re75       re78  distance weights sub
NSW1     1  37   11     1      0       1        1    0    0  9930.0460 0.6387699       1
NSW2     1  22    9     0      1       0        1    0    0  3595.8940 0.2246342       1
NSW3     1  30   12     1      0       0        0    0    0 24909.4500 0.6782439       1
NSW4     1  27   11     1      0       0        1    0    0  7506.1460 0.7763241       1
NSW5     1  33    8     1      0       0        1    0    0   289.7899 0.7016387       1
NSW6     1  22    9     1      0       0        1    0    0  4056.4940 0.6990699       1

> library(lme4)
> mfull.lmer = lmer(re78 ~ treat + (1 + treat|subclass), data = m2full.dat) # like for the q
> summary(mfull.lmer)
Linear mixed model fit by REML ['lmerMod']
Formula: re78 ~ treat + (1 + treat | subclass)
   Data: m2full.dat

REML criterion at convergence: 12633.5
Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.5267 -0.7497 -0.2851  0.5165  7.4616

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 subclass (Intercept)  4027897 2007
          treat        3810081 1952     -0.89
 Residual             51103906 7149
Number of obs: 614, groups:  subclass, 104

Fixed effects:
            Estimate Std. Error t value
(Intercept)   5862.9      507.8  11.546
treat          504.5      736.2   0.685    ## about the same as seen in base section 384 (95

Correlation of Fixed Effects:
      (Intr)
treat -0.679
> confint(mfull.lmer)              # this took a while
Computing profile confidence intervals ...
                2.5 %   97.5 %
.sig01      1216.8647 3011.968
.sig02        -1.0000    1.000
.sig03         0.0000      Inf
.sigma      6740.8624 7581.414
(Intercept) 4807.1941 6873.722
treat       -985.7685 1977.973
There were 50 or more warnings (use warnings() to see the first 50)
```

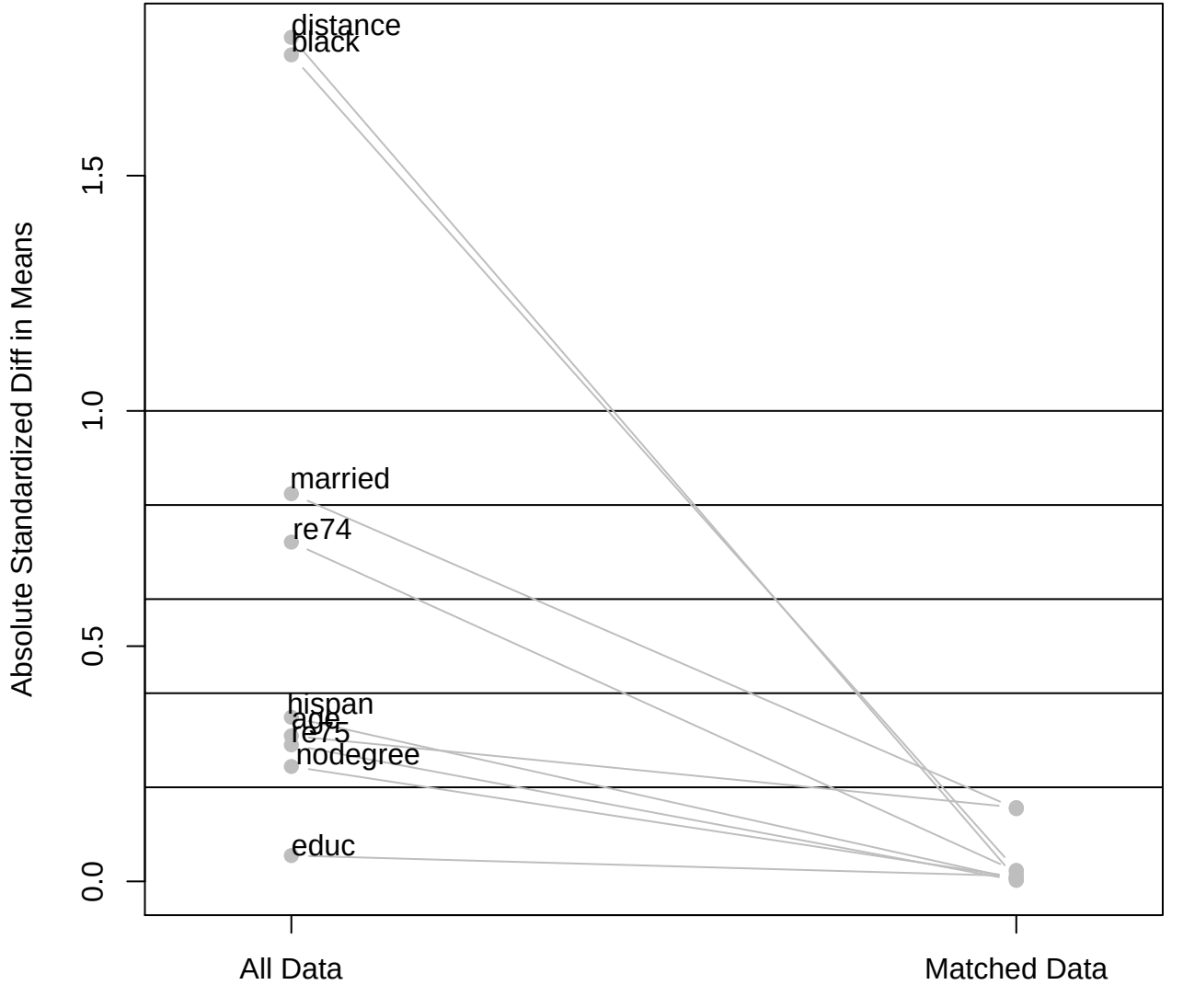
```
Formula: re78 ~ treat + (1 + treat | bins)
   Data: lalonde

REML criterion at convergence: 12637.1

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.3976 -0.7541 -0.2878  0.5408  7.4535

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 bins     (Intercept)  5208943 2282
          treat        2069963 1439     -1.00
 Residual             52597981 7252
Number of obs: 614, groups:  bins, 5

Fixed effects:
            Estimate Std. Error t value
(Intercept)   6434.2     1090.2   5.902
treat          385.7      950.8   0.406

Correlation of Fixed Effects:
      (Intr)
treat -0.795

# so here we have an overall estimate of the effect of the treat on re78 of positive $386, but
# far from significant. Much smaller point estimate than in some of the individual strata

> confint(propen.lmer) # bombs
> confint(propen.lmer, method = "boot", nsim = 1000, boot.type = "perc")
Computing bootstrap confidence intervals ...

                  2.5 %   97.5 %
.sig01        414.81230 4084.578
.sig02         -1.00000    1.000
.sig03         54.74858 3644.981
.sigma       6846.49101 7654.434
(Intercept)  4432.91940 8695.198
treat       -1681.75647 2565.802

  some bootstrap runs failed (7/1000)
```

```
########################## Full Matching (Hansen, via Rosenbaum, using MatchIt)

>  m2full.out = matchit(treat ~ re74 + re75 + educ + black + hispan + age + married + nodegree, data =
Warning message:
In fullmatch(d, ...) :
  Without 'data' argument the order of the match is not guaranteed
    to be the same as your original data.

>  summary(m2full.out)
Call:
matchit(formula = treat ~ re74 + re75 + educ + black + hispan +
    age + married + nodegree, data = lalonde, method = "full")

Summary of balance for all data:
         Means Treated Means Control  Mean Diff   eQQ Med  eQQ Mean   eQQ Max
distance        0.5774        0.1822     0.3952    0.5176    0.3955    0.5966
re74         2095.5737     5619.2365 -3523.6628 2425.5720 3620.9240 9216.5000
re75         1532.0553     2466.4844  -934.4291  981.0968 1060.6582 6795.0100
educ           10.3459       10.2354     0.1105    1.0000    0.7027    4.0000
black           0.8432        0.2028     0.6404    1.0000    0.6432    1.0000
hispan          0.0595        0.1422    -0.0827    0.0000    0.0811    1.0000
age            25.8162       28.0303    -2.2141    1.0000    3.2649   10.0000
```

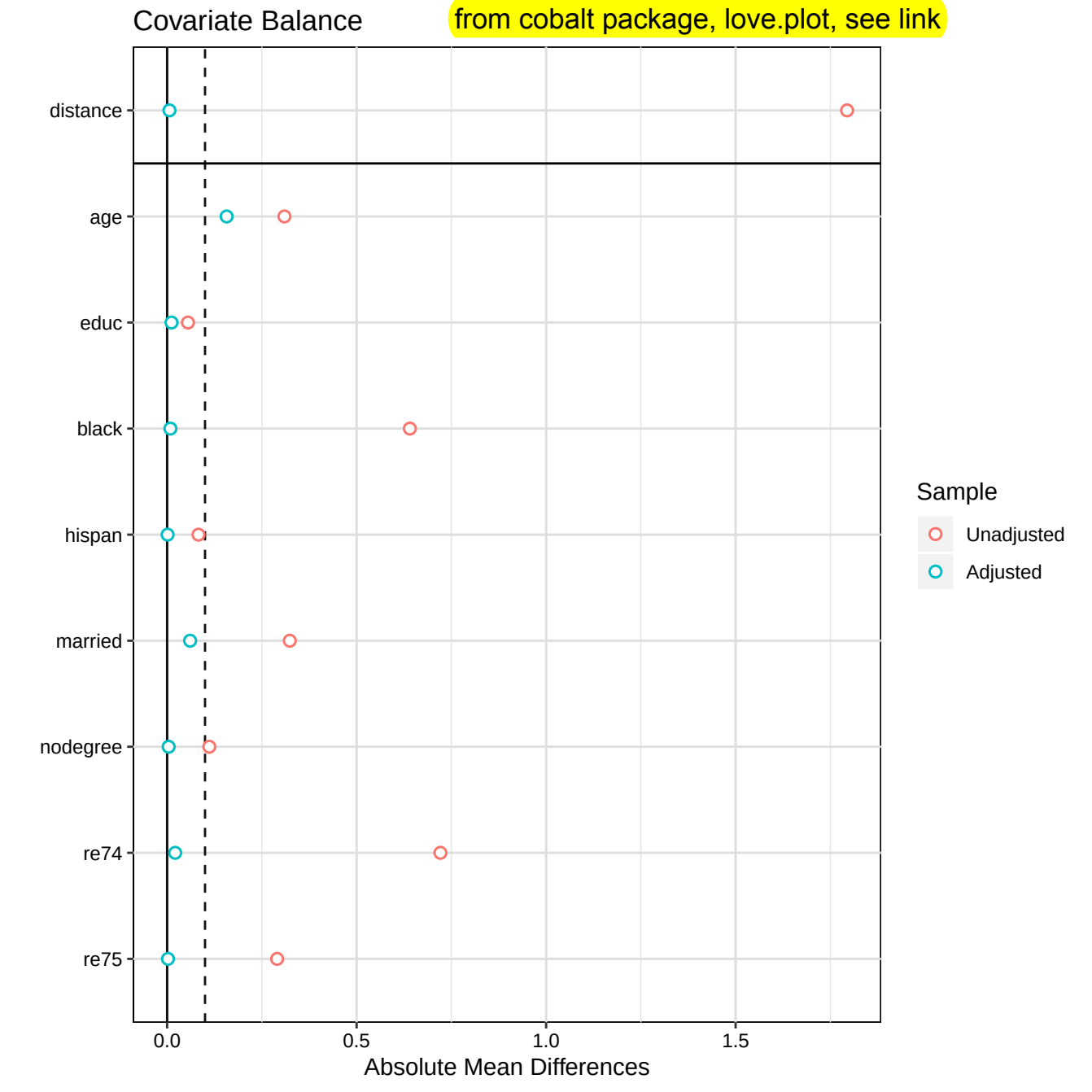
```
married         0.1892        0.5128    -0.3236    0.0000    0.3243    1.0000
nodegree        0.7081        0.5967     0.1114    0.0000    0.1135    1.0000


Summary of balance for matched data:
         Means Treated Means Control Mean Diff  eQQ Med eQQ Mean   eQQ Max
distance        0.5774         0.5761    0.0013   0.0026   0.0066     0.096
re74         2095.5737      2199.7126 -104.1390  72.6510 512.7210 13121.750
re75         1532.0553      1524.8362    7.2191 209.6655 460.5643 12746.050
educ           10.3459        10.3227    0.0233   0.0000   0.4596     4.000
black           0.8432         0.8347    0.0086   0.0000   0.0020     1.000
hispan          0.0595         0.0583    0.0012   0.0000   0.0012     1.000
age            25.8162        24.6928    1.1235   3.0000   3.3100     9.000
married         0.1892         0.1285    0.0607   0.0000   0.0544     1.000
nodegree        0.7081         0.7040    0.0041   0.0000   0.0028     1.000

Percent Balance Improvement:
         Mean Diff.   eQQ Med eQQ Mean  eQQ Max
distance    99.6662   99.5001  98.3388  83.9052
re74        97.0446   97.0048  85.8401 -42.3724
re75        99.2274   78.6295  56.5775 -87.5796
educ        78.9494  100.0000  34.5954   0.0000
black       98.6582  100.0000  99.6891   0.0000
hispan      98.5858    0.0000  98.5200   0.0000
age         49.2583 -200.0000  -1.3825  10.0000
married     81.2495    0.0000  83.2267   0.0000
nodegree    96.3435    0.0000  97.5333   0.0000

Sample sizes:
          Control Treated
All           429     185
Matched       429     185
Unmatched       0       0
Discarded       0       0
```

```
> summary(m2full.out, standardize = T)
Call:
matchit(formula = treat ~ re74 + re75 + educ + black + hispan +
    age + married + nodegree, data = lalonde, method = "full")

Summary of balance for all data:
         Means Treated Means Control Std. Mean Diff. eCDF Med eCDF Mean eCDF Max
distance        0.5774        0.1822          1.7941   0.3964    0.3774   0.6444
re74         2095.5737     5619.2365         -0.7211   0.2335    0.2248   0.4470
re75         1532.0553     2466.4844         -0.2903   0.1355    0.1342   0.2876
educ           10.3459       10.2354          0.0550   0.0228    0.0347   0.1114
black           0.8432        0.2028          1.7568   0.3202    0.3202   0.6404
hispan          0.0595        0.1422         -0.3489   0.0414    0.0414   0.0827
age            25.8162       28.0303         -0.3094   0.0827    0.0813   0.1577
married         0.1892        0.5128         -0.8241   0.1618    0.1618   0.3236
nodegree        0.7081        0.5967          0.2443   0.0557    0.0557   0.1114


Summary of balance for matched data:
         Means Treated Means Control Std. Mean Diff. eCDF Med eCDF Mean eCDF Max
distance        0.5774        0.5761          0.0060   0.0060    0.0085   0.0596
re74         2095.5737     2199.7126         -0.0213   0.0160    0.0476   0.2268
re75         1532.0553     1524.8362          0.0022   0.0348    0.0693   0.2324
educ           10.3459       10.3227          0.0116   0.0286    0.0275   0.0568
black           0.8432        0.8347          0.0236   0.0104    0.0104   0.0208
hispan          0.0595        0.0583          0.0049   0.0036    0.0036   0.0072
age            25.8162       24.6928          0.1570   0.0416    0.0857   0.3436
married         0.1892        0.1285          0.1545   0.0366    0.0366   0.0732
nodegree        0.7081        0.7040          0.0089   0.0008    0.0008   0.0016
```

```
Percent Balance Improvement:
          Std. Mean Diff. eCDF Med eCDF Mean  eCDF Max
distance          99.6662  98.4863   97.7452   90.7506
re74              97.0446  93.1488   78.8321   49.2658
re75              99.2274  74.3198   48.3597   19.2062
educ              78.9494 -25.6137   20.8722   48.9995
black             98.6582  96.7523   96.7523   96.7523
hispan            98.5858  91.2972   91.2972   91.2972
age               49.2583  49.7246   -5.3122 -117.8448
married           81.2495  77.3817   77.3817   77.3817
nodegree          96.3435  98.5634   98.5634   98.5634

Sample sizes:
           Control Treated
All            429     185
Matched        429     185
Unmatched        0       0
Discarded        0       0

>  plot(summary(m2full.out, standardize = T))
[1] "To identify the variables, use first mouse button; to stop, use second."
warning: nearest point already identified
warning: nearest point already identified
warning: nearest point already identified
warning: nearest point already identified
warning: nearest point already identified
warning: nearest point already identified
integer(0)
>    setwd("D:\\drr16\\somgen290\\week1\\")
>  plot(m2full.out)
Waiting to confirm page change...
Waiting to confirm page change...
> # gives you QQ plots for each var

> detach(lalonde)
>  m2full.dat = match.data(m2full.out)  # obtain results from the full matching
>  head(m2full.dat)
     treat age educ black hispan married nodegree re74 re75       re78    propen bins  distance weights
NSW1     1  37   11     1      0       1        1    0    0  9930.0460 0.6387699    4 0.6387699       1
NSW2     1  22    9     0      1       0        1    0    0  3595.8940 0.2246342    3 0.2246342       1
NSW3     1  30   12     1      0       0        0    0    0 24909.4500 0.6782439    5 0.6782439       1
NSW4     1  27   11     1      0       0        1    0    0  7506.1460 0.7763241    5 0.7763241       1
NSW5     1  33    8     1      0       0        1    0    0   289.7899 0.7016387    5 0.7016387       1
NSW6     1  22    9     1      0       0        1    0    0  4056.4940 0.6990699    5 0.6990699       1

> dim(m2full.dat)
[1] 614  15
> head(m2full.dat)
     treat age educ black hispan married nodegree re74 re75       re78    propen bins  distance weights
NSW1     1  37   11     1      0       1        1    0    0  9930.0460 0.6387699    4 0.6387699       1
NSW2     1  22    9     0      1       0        1    0    0  3595.8940 0.2246342    3 0.2246342       1
NSW3     1  30   12     1      0       0        0    0    0 24909.4500 0.6782439    5 0.6782439       1
NSW4     1  27   11     1      0       0        1    0    0  7506.1460 0.7763241    5 0.7763241       1
NSW5     1  33    8     1      0       0        1    0    0   289.7899 0.7016387    5 0.7016387       1
NSW6     1  22    9     1      0       0        1    0    0  4056.4940 0.6990699    5 0.6990699       1
> attach(m2full.dat)

>  # so you can see match.data appends 3 colums "distance" "weights"  "subclass" to the original data s
>  table(m2full.dat$subclass) #the 104 subclasses have various sizes

  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26
  2  13   2   7   3   5   3   2   4   2   8   3   2   2   9   4   2   9   6  14   3   2   2   6   3   4
 36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61
 14   5   3   3   2   6   2   5   3   2  10   2   4   8   3   2  14   7   2  14   2   2   4  40   2   2
 71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96
```



Other Examples: (propen success) SAT coaching ASPIRIN


Covariate Balance
from cobalt package, love.plot, see link
distance
age
educ
black
hispan
married
nodegree
re74
re75
0.0
0.5
1.0
1.5
Absolute Mean Differences
Sample
Unadjusted
Adjusted

```
R version 3.5.3 (2019-03-11) -- "Great Truth"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
# clean start 2019
> install.packages("optmatch")
> install.packages("MatchIt")
also installing the dependency ‘Matching’

> install.packages("cobalt")
also installing the dependencies ‘ggstance’, ‘backports’

> library(optmatch)
Loading required package: survival
The optmatch package has an academic license. Enter relaxinfo() for more information.
> library(MatchIt)
> data(lalonde)
> dim(lalonde)
[1] 614  10
> library(cobalt)

> head(lalonde)
     treat age educ black hispan married nodegree re74 re75       re78
NSW1     1  37   11     1      0       1        1    0    0  9930.0460
NSW2     1  22    9     0      1       0        1    0    0  3595.8940
NSW3     1  30   12     1      0       0        0    0    0 24909.4500
NSW4     1  27   11     1      0       0        1    0    0  7506.1460
NSW5     1  33    8     1      0       0        1    0    0   289.7899
NSW6     1  22    9     1      0       0        1    0    0  4056.4940
> #pick out matching vars (not treat or outcome)
> covs <- subset(lalonde, select = -c(treat, re78))
# try ?f.build from cobalt
> m2full.out = matchit(f.build("treat", covs), data = lalonde, method = "full")
Warning message:
In optmatch::fullmatch(d, ...) :
  Without 'data' argument the order of the match is not guaranteed
    to be the same as your original data.
> summary(m2full.out, standardize = T) # as we saw before

Call:
matchit(formula = f.build("treat", covs), data = lalonde, method = "full")

Summary of balance for all data:
          Means Treated Means Control Std. Mean Diff. eCDF Med eCDF Mean eCDF Max
distance         0.5774        0.1822          1.7941   0.3964    0.3774   0.6444
age             25.8162       28.0303         -0.3094   0.0827    0.0813   0.1577
educ            10.3459       10.2354          0.0550   0.0228    0.0347   0.1114
black            0.8432        0.2028          1.7568   0.3202    0.3202   0.6404
hispan           0.0595        0.1422         -0.3489   0.0414    0.0414   0.0827
married          0.1892        0.5128         -0.8241   0.1618    0.1618   0.3236
nodegree         0.7081        0.5967          0.2443   0.0557    0.0557   0.1114
re74          2095.5737     5619.2365         -0.7211   0.2335    0.2248   0.4470
re75          1532.0553     2466.4844         -0.2903   0.1355    0.1342   0.2876


Summary of balance for matched data:
          Means Treated Means Control Std. Mean Diff. eCDF Med eCDF Mean eCDF Max
distance         0.5774        0.5761          0.0060   0.0120    0.0132   0.0484
age             25.8162       24.6928          0.1570   0.0610    0.0883   0.3192
educ            10.3459       10.3227          0.0116   0.0152    0.0238   0.0624
black            0.8432        0.8347          0.0236   0.0142    0.0142   0.0284
hispan           0.0595        0.0583          0.0049   0.0016    0.0016   0.0032
married          0.1892        0.1285          0.1545   0.0318    0.0318   0.0636
```

```
nodegree        0.7081        0.7040          0.0089   0.0008    0.0008   0.0016
re74         2095.5737     2199.7126         -0.0213   0.0140    0.0449   0.2288
re75         1532.0553     1524.8362          0.0022   0.0284    0.0573   0.1964

Percent Balance Improvement:
         Std. Mean Diff. eCDF Med eCDF Mean  eCDF Max
distance         99.6662  96.9725   96.5015   92.4887
age              49.2583  26.2789   -8.5456 -102.3750
educ             78.9494  33.2403   31.4865   43.9713
black            98.6582  95.5656   95.5656   95.5656
hispan           98.5858  96.1321   96.1321   96.1321
married          81.2495  80.3480   80.3480   80.3480
nodegree         96.3435  98.5634   98.5634   98.5634
re74             97.0446  94.0052   80.0291   48.8184
re75             99.2274  79.0426   57.2989   31.7216

Sample sizes:
          Control Treated
All           429     185
Matched       429     185
Unmatched       0       0
Discarded       0       0

> # same as prior exs
# MatchIt has a plot-- here's one that you may prefer
> love.plot(bal.tab(m2full.out), threshold = .1)
# the other main cobalt function, works same as MatchIt
> bal.tab(m2full.out)
Call
 matchit(formula = f.build("treat", covs), data = lalonde, method = "full")

Balance Measures
             Type Diff.Adj
distance Distance   0.0060
age       Contin.   0.1570
educ      Contin.   0.0116
black      Binary   0.0086
hispan     Binary   0.0012
married    Binary   0.0607
nodegree   Binary   0.0041
re74      Contin.  -0.0213
re75      Contin.   0.0022

Effective sample sizes
           Control Treated
Unadjusted 429.000     185
Adjusted    53.329     185
> # same as matchit summary()
>
```

```
Unmatched       0       0
Discarded       0       0

#### do the outcome re78 analysis
> m2fullvars.out = matchit(treat ~ re74 + re75 + educ + black + hispan + age + married + nod

> m2full.dat = match.data(m2fullvars.out)
> head(m2full.dat)
     treat age educ black hispan married nodegree re74 re75       re78  distance weights sub
NSW1     1  37   11     1      0       1        1    0    0  9930.0460 0.6387699       1
NSW2     1  22    9     0      1       0        1    0    0  3595.8940 0.2246342       1
NSW3     1  30   12     1      0       0        0    0    0 24909.4500 0.6782439       1
NSW4     1  27   11     1      0       0        1    0    0  7506.1460 0.7763241       1
NSW5     1  33    8     1      0       0        1    0    0   289.7899 0.7016387       1
NSW6     1  22    9     1      0       0        1    0    0  4056.4940 0.6990699       1

> library(lme4)
> mfull.lmer = lmer(re78 ~ treat + (1 + treat|subclass), data = m2full.dat) # like for the q
> summary(mfull.lmer)
Linear mixed model fit by REML ['lmerMod']
Formula: re78 ~ treat + (1 + treat | subclass)
   Data: m2full.dat

REML criterion at convergence: 12633.5
Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.5267 -0.7497 -0.2851  0.5165  7.4616

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 subclass (Intercept)  4027897 2007
          treat        3810081 1952     -0.89
 Residual             51103906 7149
Number of obs: 614, groups:  subclass, 104

Fixed effects:
            Estimate Std. Error t value
(Intercept)   5862.9      507.8  11.546
treat          504.5      736.2   0.685    ## about the same as seen in base section 384 (95

Correlation of Fixed Effects:
      (Intr)
treat -0.679
> confint(mfull.lmer)              # this took a while
Computing profile confidence intervals ...
                2.5 %   97.5 %
.sig01      1216.8647 3011.968
.sig02        -1.0000    1.000
.sig03         0.0000      Inf
.sigma      6740.8624 7581.414
(Intercept) 4807.1941 6873.722
treat       -985.7685 1977.973
There were 50 or more warnings (use warnings() to see the first 50)
```

Week 1 Computing Corner

Stat 266
CHPR 290

```
> data(lalonde) # in MatchIt package, help(lalonde)
> dim(lalonde)  > attach(lalonde)
[1] 614  10
> table(treat)
treat
  0   1
429 185
> head(lalonde)
     treat age educ black hispan married nodegree re74 re75       re78
NSW1     1  37   11     1      0       1        1    0    0  9930.0460
NSW2     1  22    9     0      1       0        1    0    0  3595.8940
NSW3     1  30   12     1      0       0        0    0    0 24909.4500
NSW4     1  27   11     1      0       0        1    0    0  7506.1460
NSW5     1  33    8     1      0       0        1    0    0   289.7899
NSW6     1  22    9     1      0       0        1    0    0  4056.4940

################## prelim compare groups on outcome measure
> tapply(re78, treat, median)
       0        1
4975.505 4232.309
> t.test(re78 ~ treat)
        Welch Two Sample t-test
data:  re78 by treat
t = 0.93773, df = 326.41, p-value = 0.3491
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:   -697.192 1967.244
sample estimates: mean in group 0 mean in group 1
                        6984.170        6349.144

> #####But wait, some say "we are never done until the ancova is run" see Fish
> # as we see the social science, life science practice is to put in the treatment variable  and
> # a whole bunch of other variables to "control" for self-selection, nonequivalence etc.
> #  equivalent to analysis of covariance by whatever name
>  ancova.lalonde = lm( re78 ~ treat + age + educ + black + hispan + married + nodegree + re74 + re75)
>  summary(ancova.lalonde)
Call: lm(formula = re78 ~ treat + age + educ + black + hispan + married +  nodegree + re74 + re75)
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.651e+01  2.437e+03   0.027   0.9782
treat        1.548e+03  7.813e+02   1.982   0.0480 *
age          1.298e+01  3.249e+01   0.399   0.6897
educ         4.039e+02  1.589e+02   2.542   0.0113 *
black       -1.241e+03  7.688e+02  -1.614   0.1071
hispan       4.989e+02  9.419e+02   0.530   0.5966
married      4.066e+02  6.955e+02   0.585   0.5590
nodegree     2.598e+02  8.474e+02   0.307   0.7593
re74         2.964e-01  5.827e-02   5.086 4.89e-07 ***
re75         2.315e-01  1.046e-01   2.213   0.0273 *
---
> # so treatment is significantly helpful ??

########### Begin matching analysis; Quintile Subclassification with Propensity Scores
## original Rosenbaum-Rubin, cardiac; Rubin breast cancer

> # now do the logistic regression that computes propensity scores
  # matching packages will do this for you with propen as distance measure
>  glm.p = glm( treat ~ age + educ + black + hispan + married + nodegree + re74 + re75,
                                                  data = lalonde, family = binomial)

>  summary(glm.p)
Call: glm(formula = treat ~ age + educ + black + hispan + married +
    nodegree + re74 + re75, family = binomial, data = lalonde)
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.729e+00  1.017e+00  -4.649 3.33e-06 ***
age          1.578e-02  1.358e-02   1.162  0.24521
educ         1.613e-01  6.513e-02   2.477  0.01325 *
black        3.065e+00  2.865e-01  10.699  < 2e-16 ***
```

training (treatment)

outcome

control has higher wages (re78)

First approach, vintage

fit from logistic regression

Week 1 p2
Stat 266

```
hispan       9.836e-01  4.257e-01   2.311  0.02084 *
married     -8.321e-01  2.903e-01  -2.866  0.00415 **
nodegree     7.073e-01  3.377e-01   2.095  0.03620 *
re74        -7.178e-05  2.875e-05  -2.497  0.01253 *
re75         5.345e-05  4.635e-05   1.153  0.24884
---

> propen = fitted(glm.p)  # now we have the propensity scores

> quantile(propen) # overall distrib
         0%          25%          50%          75%         100%
0.009080193 0.048536484 0.120676493 0.638715991 0.853152844
 # look at overlap via 5-number summary (or side-by-side boxplots) not good overlap,
> tapply(propen, treat, quantile)
$`0`
         0%          25%          50%          75%         100%
0.009080193 0.038880745 0.075849106 0.195135746 0.789172834
$`1`
        0%         25%         50%         75%        100%
0.02495179 0.52646352 0.65368426 0.72659995 0.85315284

> # as we are fitting prob(treat = 1) fits for those in treatment group will be larger,
   # we need good overlap for matching purposes
> detach(lalonde) > lalonde$propen = propen  > attach(lalonde)
> boxplot(propen ~ treat) #gives side-by-side boxplots, you can add labels, not wonderful overlap
```

see pictures

```
#### looking at overlap, histograms
>  p1 = propen[treat == 1]  >  p0 = propen[treat == 0]
>  hist(p0,col=rgb(0,0,1,0.7),xlim=range(c(p0,p1)))
>  hist(p1,col=rgb(1,0,0,0.7),add=T)  # superimposed propensity histograms, like Ben Hansen SAT,
   control is blue, treatment is red, overlap close to perfect Stanford Cardinal red
> hist(p0, breaks = "FD", col=rgb(0,0,1,0.7),xlim=range(c(p0,p1)))
> hist(p1, breaks = "FD", col=rgb(1,0,0,0.7),add=T) # Freedman-Diaconis breakpoints

### make quintiles of propensity distribution to to subclassification/strata matching
>  pbin = cut(propen, quantile(propen, seq(0, 1, 1/5)), include.lowest = TRUE, labels = FALSE)
> detach(lalonde) > lalonde$bins = pbin > attach(lalonde)
>  table(pbin, treat)  #each bin of size 122,123
    treat
pbin   0   1
   1 122   1
   2 116   7
   3 101  21
   4  53  71
   5  37  85
```

a pbin classification for each subject

```
##### examples of checking balance (more to come)
> tapply(age, list(bins, treat), median)
   0  1
1 29 27
2 26 23
3 20 23
4 24 25
5 19 25
```

not great see picture

```
> ### install.packages("PSAgraphics") >  library(PSAgraphics)
>  box.psa(age, treat, bins) # see picture
################################### examine outcome re78 by strata
> tapply(re78, list(bins, treat),mean)  # mean diffs in re78 stratified by propensity quintile
          0        1
1 10467.064    0.000
2  5796.548 7919.316
3  6043.316 9210.726
4  4977.401 5819.143
5  4666.221 6030.258
> # direction of mean diffs favors treatment, job training

> # contrast that with the comparison ignoring any concerns about self-selection (selection bias),
```

```
     effect in the other direction, but not significant
>  tapply(re78, treat, mean)
       0        1
6984.170 6349.144

> ##### can do t-tests by subclassification (strata) e.g. for the 3 upper quintiles
> ##### lmer, a better way to do the t-tests  ########################
> library(lme4)
> propen.lmer = lmer(re78 ~ treat + (1 + treat|bins), data = lalonde)
> summary(propen.lmer)
Linear mixed model fit by REML ['lmerMod']
Formula: re78 ~ treat + (1 + treat | bins)   Data: lalonde
Random effects:
 Groups   Name        Variance Std.Dev. Corr
 bins     (Intercept)  5208943 2282
          treat        2069963 1439     -1.00
 Residual             52597981 7252
Number of obs: 614, groups:  bins, 5

Fixed effects:
            Estimate Std. Error t value
(Intercept)   6434.2     1090.2   5.902
treat          385.7      950.8   0.406

# so here we have an overall estimate of the effect of the treat on re78 of positive $386, but
# far from significant. Much smaller point estimate than in some of the individual strata

> confint(propen.lmer) # bombs
> confint(propen.lmer, method = "boot", nsim = 1000, boot.type = "perc")
Computing bootstrap confidence intervals ...
                 2.5 %   97.5 %
.sig01        414.81230 4084.578
.sig02         -1.00000    1.000
.sig03         54.74858 3644.981
.sigma       6846.49101 7654.434
(Intercept)  4432.91940 8695.198
treat       -1681.75647 2565.802  some bootstrap runs failed (7/1000)
```

second, another approach

```
############################ Full Matching (Hansen, via Rosenbaum, using MatchIt)

>  m2full.out = matchit(treat ~ re74 + re75 + educ + black + hispan + age + married + nodegree,
                                                                  data = lalonde, method = "full")
>  summary(m2full.out)
Call: matchit(formula = treat ~ re74 + re75 + educ + black + hispan +
    age + married + nodegree, data = lalonde, method = "full")
Summary of balance for all data:
         Means Treated Means Control  Mean Diff   eQQ Med  eQQ Mean   eQQ Max
distance        0.5774        0.1822     0.3952    0.5176     0.3955    0.5966
re74         2095.5737     5619.2365 -3523.6628 2425.5720 3620.9240 9216.5000
re75         1532.0553     2466.4844  -934.4291  981.0968 1060.6582 6795.0100
educ           10.3459       10.2354     0.1105    1.0000     0.7027    4.0000
black           0.8432        0.2028     0.6404    1.0000     0.6432    1.0000
hispan          0.0595        0.1422    -0.0827    0.0000     0.0811    1.0000
age            25.8162       28.0303    -2.2141    1.0000     3.2649   10.0000
married         0.1892        0.5128    -0.3236    0.0000     0.3243    1.0000
nodegree        0.7081        0.5967     0.1114    0.0000     0.1135    1.0000
Summary of balance for matched data:
         Means Treated Means Control Mean Diff  eQQ Med eQQ Mean   eQQ Max
distance        0.5774        0.5761    0.0013   0.0026   0.0066     0.096
re74         2095.5737     2199.7126 -104.1390  72.6510 512.7210 13121.750
re75         1532.0553     1524.8362    7.2191 209.6655 460.5643 12746.050
educ           10.3459       10.3227    0.0233   0.0000   0.4596     4.000
black           0.8432        0.8347    0.0086   0.0000   0.0020     1.000
hispan          0.0595        0.0583    0.0012   0.0000   0.0012     1.000
age            25.8162       24.6928    1.1235   3.0000   3.3100     9.000
married         0.1892        0.1285    0.0607   0.0000   0.0544     1.000
```

```
nodegree        0.7081        0.7040    0.0041   0.0000   0.0028     1.000
Percent Balance Improvement:
          Mean Diff.   eQQ Med eQQ Mean  eQQ Max
distance     99.6662   99.5001  98.3388  83.9052
re74         97.0446   97.0048  85.8401 -42.3724
re75         99.2274   78.6295  56.5775 -87.5796
educ         78.9494  100.0000  34.5954   0.0000
black        98.6582  100.0000  99.6891   0.0000
hispan       98.5858    0.0000  98.5200   0.0000
age          49.2583 -200.0000  -1.3825  10.0000
married      81.2495    0.0000  83.2267   0.0000
nodegree     96.3435    0.0000  97.5333   0.0000

Sample sizes:
          Control Treated   # uses all cases, as do 'inferior' IPTW methods
All           429     185
Matched       429     185
Unmatched       0       0
Discarded       0       0
```

(twang)

alternative optimal 2:1 or 1:1
see RQ 4.1

```
>  summary(m2full.out, standardize = T)
>  plot(summary(m2full.out, standardize = T))  # see picture. 10% criteria
>  plot(m2full.out) > # gives you QQ plots for each var

> detach(lalonde)
>  m2full.dat = match.data(m2full.out)  # obtain results from the full matching
> dim(m2full.dat)
[1] 614  15
> head(m2full.dat) > attach(m2full.dat)
```

get matching data

```
>  # so you can see match.data appends 3 colums "distance" "weights"  "subclass" to the original data s
>  table(m2full.dat$subclass) #the 104 subclasses have various sizes
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26
  2  13   2   7   3   5   3   2   4   2   8   3   2   2   9   4   2   9   6  14   3   2   2   6   3   4
 36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61
 14   5   3   3   2   6   2   5   3   2  10   2   4   8   3   2  14   7   2  14   2   2   4  40   2   2
 71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96
  2   3  70   2   5   6   2   2  13   2   2   2   2   2   7   3   2   2   3   2   2   2   2   3   6   4

###### outcome comparison over the (matched) subclasses # like for the quintiles
> mfull.lmer = lmer(re78 ~ treat + (1 + treat|subclass), data = m2full.dat)
> summary(mfull.lmer)
Linear mixed model fit by REML ['lmerMod']
Formula: re78 ~ treat + (1 + treat | subclass)
   Data: m2full.dat
Number of obs: 614, groups:  subclass, 104

Fixed effects:
            Estimate Std. Error t value
(Intercept)   5862.9      507.8  11.546
treat          504.5      736.2   0.685 ## about the same as seen in base section 384 (952)
```

analog to paired t-test

```
> confint(mfull.lmer)
Computing profile confidence intervals ...
                2.5 %   97.5 %
.sig01      1216.8647 3011.968
.sig02        -1.0000    1.000
.sig03         0.0000      Inf
.sigma      6740.8624 7581.414
(Intercept) 4807.1941 6873.722
treat       -985.7685 1977.973
There were 50 or more warnings (use warnings() to see the first 50)
>
```

a little tighter CI

```
# outcome analysis: optmatch fullmatch, lalonde data
# see week 1 RQ7 for balance checks

R version 3.5.3 (2019-03-11) -- "Great Truth"
Copyright (C) 2019 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)
> library(optmatch)

> data(lalonde)
> dim(lalonde)
[1] 614  10
> library(cobalt)

Attaching package: 'cobalt'

The following object is masked _by_ '.GlobalEnv':

    lalonde

The following object is masked from 'package:MatchIt':

    lalonde

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

> #####
> ###now try optmatch

> # cobalt vignette does it this way
> covs <- subset(lalonde, select = -c(treat, re78))
> pfit = glm(f.build("treat", covs), data = lalonde, family = "binomial")
> lalonde$p.score = pfit$fitted.values #get the propensity score
> boxplot(lalonde$p.score ~ lalonde$treat) # propensity score
> fm2 = fullmatch(treat ~ p.score, data = lalonde)
> bal.tab(fm2, formula = f.build("treat", covs), data = lalonde)
Call
 fullmatch(x = treat ~ p.score, data = lalonde)

Balance Measures
            Type Diff.Adj
age      Contin.  0.1494
educ     Contin. -0.0364
black     Binary  0.0086
hispan    Binary -0.0013
married   Binary  0.0579
nodegree  Binary  0.0092
re74     Contin. -0.0636
re75     Contin. -0.0124

Sample sizes
        Control Treated
All         429     185
Matched     429     185

> love.plot(bal.tab(fm2, formula = f.build("treat", covs), data = lalonde))
> # close, but not quite the same balance table as MatchIt(full); 102 subgroups, 104 from Matchit(full)

> summary(fm2)
Structure of matched sets:
5+:1  4:1  3:1  2:1  1:1  1:2  1:3  1:4 1:5+
   8    2    6   16   39    7    4    5   15
Effective Sample Size:  137.4
(equivalent number of matched pairs).

> # looks like we got 102 subclasses here, 104 from MatchIt(full)
> stratumStructure(fm2)
12:1  9:1  8:1  7:1  6:1  5:1  4:1  3:1  2:1  1:1  1:2  1:3  1:4  1:5  1:6  1:7  1:8  1:9 1:12 1:13
   1    1    1    1    1    3    2    6   16   39    7    4    5    1    1    1    1    1    3    3
1:20 1:41 1:51 1:90
   1    1    1    1

## what we need for outcome analysis is to add the subclass info for each unit to the lalonde dataset
```

```
## In MatchIt matched.data does this for us
# here we grab a factor giving us the subclass info
# I call the augmented lalonde dataset "matched"

> matched = cbind(lalonde, matches = fm2)
> head(matched)
     treat age educ black hispan married nodegree re74 re75       re78   p.score matches
NSW1     1  37   11     1      0       1        1    0    0  9930.0460 0.6387699     1.1
NSW2     1  22    9     0      1       0        1    0    0  3595.8940 0.2246342    1.98
NSW3     1  30   12     1      0       0        0    0    0 24909.4500 0.6782439   1.109
NSW4     1  27   11     1      0       0        1    0    0  7506.1460 0.7763241   1.120
NSW5     1  33    8     1      0       0        1    0    0   289.7899 0.7016387   1.131
NSW6     1  22    9     1      0       0        1    0    0  4056.4940 0.6990699   1.142
> matched[1:20,]
      treat age educ black hispan married nodegree re74 re75       re78    p.score matches
NSW1      1  37   11     1      0       1        1    0    0  9930.0460 0.63876993     1.1
NSW2      1  22    9     0      1       0        1    0    0  3595.8940 0.22463424    1.98
NSW3      1  30   12     1      0       0        0    0    0 24909.4500 0.67824388   1.109
NSW4      1  27   11     1      0       0        1    0    0  7506.1460 0.77632408   1.120
NSW5      1  33    8     1      0       0        1    0    0   289.7899 0.70163874   1.131
NSW6      1  22    9     1      0       0        1    0    0  4056.4940 0.69906990   1.142
NSW7      1  23   12     1      0       0        0    0    0     0.0000 0.65368426   1.153
NSW8      1  32   11     1      0       0        1    0    0  8472.1580 0.78972311   1.164
NSW9      1  22   16     1      0       0        0    0    0  2164.0220 0.77983825   1.120
NSW10     1  33   12     0      0       1        0    0    0 12418.0700 0.04292461     1.2
NSW11     1  19    9     1      0       0        1    0    0  8173.9080 0.68901996    1.13
NSW12     1  21   13     1      0       0        0    0    0 17094.6400 0.68244400    1.24
NSW13     1  18    8     1      0       0        1    0    0     0.0000 0.64986767    1.35
NSW14     1  27   10     1      0       1        1    0    0 18739.9300 0.56241073    1.46
NSW15     1  17    7     1      0       0        1    0    0  3023.8790 0.60858629    1.57
NSW16     1  19   10     1      0       0        1    0    0  3228.5030 0.72249036    1.68
NSW17     1  27   13     1      0       0        0    0    0 14581.8600 0.70259562   1.131
NSW18     1  23   10     1      0       0        1    0    0  7693.4000 0.73496416    1.90
NSW19     1  40   12     1      0       0        0    0    0 10804.3200 0.71166489    1.97
NSW20     1  26   12     1      0       0        0    0    0 10747.3500 0.66431981    1.99
> table(matched$matches)

  1.1 1.100 1.101 1.102 1.107 1.108 1.109  1.11 1.113 1.114 1.118 1.119 1.120 1.121 1.122 1.123
    3    13     2     7     5     3     2     2     8     3     2     2     9     3     3    10
1.124 1.125 1.126 1.129  1.13 1.131 1.132 1.133 1.134 1.135 1.137 1.138 1.139  1.14 1.140 1.141
    5    14     4     2     2     6     2     3     4     3     2     3     4     4     4     3
1.142 1.143 1.145 1.148 1.151 1.152 1.153 1.154 1.155 1.157  1.16 1.160 1.162 1.164 1.166 1.167
    2     2     3     2    13     5     2     3     2     6     5     3     2    10     2     3
1.168  1.17 1.170 1.172 1.174 1.176 1.182 1.183 1.185  1.19   1.2  1.20  1.22  1.23  1.24  1.26
    8     3     3    14     4     2    14     2     3     2    21    42     2     3     2     2
 1.28  1.29   1.3  1.30  1.31  1.34  1.35  1.37  1.40  1.43  1.44  1.46  1.47  1.49  1.52  1.53
    2    91     5     2     2    13     3     3     3    52     2     5     6     2    13     2
 1.57  1.60  1.63  1.68   1.7  1.71  1.72  1.75  1.76  1.82  1.85  1.87  1.89  1.90  1.91  1.92
    4     2     2     7     3     2     2     3     2     2     2     3     9     4     2     4
 1.93  1.94  1.96  1.97  1.98  1.99
    2     5     2     2     4     6
> length(table(matched$matches))
[1] 102

> library(lme4)
Loading required package: Matrix

> str(matched)
'data.frame':   614 obs. of  13 variables:
 $ treat   : int  1 1 1 1 1 1 1 1 1 1 ...
 $ age     : int  37 22 30 27 33 22 23 32 22 33 ...
 $ educ    : int  11 9 12 11 8 9 12 11 16 12 ...
 $ black   : int  1 0 1 1 1 1 1 1 1 0 ...
 $ hispan  : int  0 1 0 0 0 0 0 0 0 0 ...
 $ married : int  1 0 0 0 0 0 0 0 0 1 ...
 $ nodegree: int  1 1 0 1 1 1 0 1 0 0 ...
 $ re74    : num  0 0 0 0 0 0 0 0 0 0 ...
 $ re75    : num  0 0 0 0 0 0 0 0 0 0 ...
 $ re78    : num  9930 3596 24909 7506 290 ...
 $ p.score : num  0.639 0.225 0.678 0.776 0.702 ...
 $ matches : Factor w/ 102 levels "1.1","1.100",..: 1 101 7 13 22 33 39 46 13 59 ...

# so now we can use the factor matches just like we used subclass from MatchIt
# lmer isn't that numerically happy, but we get about the same result

> optmatch_lmer2 = lmer(re78 ~ treat + (1 + treat|matches), data = matched)
Warning message:
In checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv,  :
  Model failed to converge with max|grad| = 0.00776755 (tol = 0.002, component 1)
```

```
> summary(optmatch_lmer2)
Linear mixed model fit by REML ['lmerMod']
Formula: re78 ~ treat + (1 + treat | matches)
   Data: matched

REML criterion at convergence: 12634.7

Scaled residuals:
    Min      1Q  Median      3Q     Max
-1.5112 -0.7597 -0.2716  0.5129  7.4641

Random effects:
 Groups   Name        Variance Std.Dev. Corr
 matches  (Intercept)  3476596 1865
          treat        7026358 2651     -1.00
 Residual             51450478 7173
Number of obs: 614, groups:  matches, 102

Fixed effects:
            Estimate Std. Error t value
(Intercept)   5875.6      491.7  11.950
treat          464.6      743.7   0.625

Correlation of Fixed Effects:
      (Intr)
treat -0.691
convergence code: 0
Model failed to converge with max|grad| = 0.00776755 (tol = 0.002, component 1)
```

Software (R software with no guarantees)

Two R Packages for Sensitivity Analysis in Observational Studies

sensitivitymv (R package at cran)

sensitivitymw (Rpackage at cran)

"A new u-statistic..." Biometrics 2011 R-Session (Supplement 2): txt.document

Match Functions from Design of Observational Studies R workspace

Selected Data Sets from Design of Observational Studies R workspace

Appendix 3.9 from Design of Observational Studies R workspace

Software supplement to "Imposing minimax constraints..." pdf aamatch package local files zip tar.gz

Suggested R Packages for Matching

Ben Hanson's optmatch (at cran)

Sam Pimentel's rcbalance (at cran)

Bo Lu, Robert Greevy, Xinyi Xu and Cole Beck's nbpMatching (at cran)

Dan Yang's finebalance package (archived but working at cran)

Jose Zubizarreta's mipmatch (requires special installation)

Adaptive sensitivity analysis

Dylan Small's SensitivityCaseControl (at cran) including adaptive.noether.brown