SCRIPT `MOD5S3`: TREATMENT EFFECTS VIA MATCHING: JOB TRAINING APPLICATION

1. Load and describe data

This script uses the same data as `mod5s1`.

(1) train =1 if assigned to job training
(2) age =age in 1977
(3) educ =years of education
(4) black =1 if black
(5) hisp =1 if Hispanic
(6) married =1 if married
(7) nodegree =1 if no high school degree
(8) mosinex =No. months prior to Jan. 78 in experiment
(9) re74 =real earnings, 1974, $1000s
(10) re75 =real earnings, 1975, $1000s
(11) re78 =real earnings, 1978, $1000s
(12) unem74 =1 if unemployed all of 1974
(13) unem75 =1 if unemployed all of 1975
(14) unem78 =1 if unemployed all of 1978
(15) lre74 =log(re74); zero if re74 == 0
(16) lre75 =log(re75); zero if re75 == 0
(17) lre78 =log(re78); zero if re78 == 0
(18) agesq =age squared
(19) mostrn =months in training

```
R> load("c:/Klaus/AAEC5126/R/data/jtrain2.rda")
```

Data preparation

```
R> attach(data)
R> y<-as.matrix(re78)
R> n<-nrow(y)
R> w<-as.matrix(train)
R> #
R> f0<-find(w==0)
R> f1<-find(w==1)
R> #
R> y0<-as.matrix(y[f0])
R> y1<-as.matrix(y[f1])
R> n1<-nrow(y1)
R> n0<-nrow(y0)
```

```
R> #
R> X<-as.matrix(cbind(age,black,hisp,married,re74,re75,educ))
R> X1<-as.matrix(X[f1,])
R> X0<-as.matrix(X[f0,])
R> #we want to enforce exact matching on education, so place it last
R> k<-ncol(X)
R> #
R> M<-1 #min. number of matches per treated obs.
R> Me<-1 #number of variables that must match exactly
```

Find matches, store indices

```
R> y0hat=rep(n1,1)  # will collect counterfactual estimates
R> IMatch<-vector('list',n1) #collects matching info for each treatment obs
R> JMivec=rep(0,n1) #collects number of matches used for each treatment obs
R> KMlvec=rep(0,n)
R> #collects weighted counts for how often each control is used as match
R> # will be zero for treated obs's
R> #
R> Vi <- 1/as.matrix(diag(var(X))) #vector of inverted variances
R> pen<-c(rep(1,(k-Me)), rep(1000,Me))# penalties for variance terms
R> Vi<-Vi*pen  #penalize for exact matches
R> #
R> for (i in 1:n1) {
   xi<-as.matrix(X1[i,])
   int1<-(repmat(xi,n0,1)-X0)^2 #squared differences, k by n0
   int2<- repmat(Vi,n0,1)
   int<-as.matrix(sqrt(rowSums(int1*int2))) #n0 by 1 matching scores
   #
   Imat<-cbind(f0,y0,int)
   Imat<-Imat[order(int),] #sort in order of lowest to highest matching score
   int<-Imat[,3]
   #find >= M observations with lowest distance, allowing for ties
   g<-1 #counter for unique values - we need exactly M
   j<-1 #counts over observations
   #
   while (g<(M+1)){
     d<-int[j]-int[j+1]
     if (d!=0) g<-g+1
     j=j+1
   }
   #
   y0hat[i]=mean(Imat[1:(j-1),2])
   IMatch[[i]]=Imat[1:(j-1), ]
   JMivec[i]<-j-1
   #
   f<-Imat[1:(j-1),1] #set of indices for controls
```

```
  KMlvec[f]<-KMlvec[f]+(1/(j-1))
 }
```

Check for covariate balance (overlap)

```
R> chosen<-find(KMlvec>0)
R> #index vector for controls that were selected as a match at least once
R> X0m<-as.matrix(X[chosen,]) #covariate for matched controls
R> y0m<-as.matrix(y[chosen]) #outcome for matched controls
R> #
R> my1<-mean(y1)
R> my0<-mean(y0m)
R> sy1<-sd(y1)
R> sy0<-sd(y0m)
R> ndiffy<-(my1-my0)/sqrt(sy1^2 + sy0^2)
R> #
R> mX1<-colMeans(X1)
R> mX0<-colMeans(X0m)
R> sX1<-apply(X1,2,sd)
R> sX0<-apply(X0m,2,sd)
R> ndiffX<-as.vector((mX1-mX0)/sqrt(sX1^2 + sX0^2))
R> #
R> tt<-data.frame(col1=c("re78","age","black","hisp",
       "married","re74","re75","educ"),
 col2=c(ndiffy,ndiffX))
R> colnames(tt)<-c("variable","norm.diff")
```

Table 1. normalized differences treated vs. chosen controls

variable	norm.diff
re78	0.23
age	0.23
black	-0.02
hisp	-0.03
married	0.13
re74	0.14
re75	0.18
educ	0.07

All overlap scores are (well) below 0.25, so we can expect for ATE and ATT to be well-identified, using these explanatory variables and the matched control observations. Note: We could repeat the matching procedure chosing different distance metrics and / or different number of matches to further improve overlap. This is an active area of research in the matching literature. See, for example, Ho et al. (2007) and Diamond and Sekhon (2013).

Compute uncorrected estimator & percentage of exact matches

```
R> ATT <-mean(y1-y0hat) #correct, same as Matlab
R> # of exact matches on education
R> ################################
R> exact<-rep(0,n1) #collects counts of exact matches for education
R> #
R> for (i in 1:n1){
     #Initial code: fi<-IMatch[[i]][,1]
     #Important note: With M=1, IMatch will usually only have one row,
     #but R doesn't understand that this is a matrix construct with one row,
     #and thus creates an error message when you call the first column of that "matrix."
     #BETTER:
     fiprep<-matrix(IMatch[[i]],ncol=3) # now it gets it,
     # a row with 3 columns (but multiple rows still OK)
     fi<-fiprep[,1]; #id's of matched observations

     int2<-X1[i,k]-X[fi,k] #difference in educ
     #
     fAll<-find(int2==0)
     exact[i]=length(fAll)
 }
R> pAll<-sum(exact)/sum(JMivec)
```

Compute consistent standard errors for uncorrected estimator

```
R> # compute sighat
R> ################################
R> sumterm=rep(0,n1)
R> for (i in 1:n1) {
     #outi=IMatch[[i]] #same problem as above
     outi<-matrix(IMatch[[i]],ncol=3)
     JMi<-nrow(outi) #number of obs's used for matching
     y0l<-outi[,2]
     int<-sum((y1[i]-y0l-ATT)^2) #y0l is JMi by 1, the other terms are scalars
     sumterm[i]<-(1/JMi)*int
 }
R> # compute variance
R> ################################
R> sighat<-(1/(2*n1))*sum(sumterm)
R> VarATT<-(sighat/n1^2)*(n1+sum(KMlvec^2))
R> seATT<-sqrt(VarATT) #correct, same as Matlab
R> tATT<-ATT/seATT
```

Compute corrected estimator

```
R> #
R> # run auxiliary regression
R> ############################
```

```
R> Xfull<-cbind(rep(1,n), X[,1:(k-Me)])
R> #add constant, drop variables that must match exactly
R> X1full=cbind(rep(1,n1), X1[,1:(k-Me)])
R> X0full=cbind(rep(1,n0), X0[,1:(k-Me)])
R> #
R> kfull<-ncol(Xfull)
R> #
R> fK<-find(KMlvec!=0) #use only matched obs
R> Kaux=KMlvec[fK]
R> yaux=sqrt(Kaux)*y[fK]
R> #weighting by (weighted) number of time an obs. was matched
R> Xaux<-repmat(sqrt(Kaux),1,kfull) * Xfull[fK,]
R> #
R> baux<-solve((t(Xaux)) %*% Xaux) %*% (t(Xaux) %*% yaux)
R> y1pred<-X1full %*% baux
R> #
R> # Compute estimator
R> ##########################
R> y0hat<-rep(0,n1) # will collect counterfactual estimates
R> for (i in 1:n1) {
     fiprep<-matrix(IMatch[[i]],ncol=3) #same correction as above
     fi<-fiprep[,1]  # id's of matched observations
     yl<-fiprep[,2]  # outcomes for matched controls
     y0lpred<-Xfull[fi,] %*% baux #predictions for matched controls
     y0hat[i]<- mean(yl-y0lpred+y1pred[i])
     #last element is a scalar, but R gets it
 }
R> #
R> ATTc<-mean(y1-y0hat) #same as Matlab
```

Compute consistent standard errors for corrected estimator

```
R> # compute sighat
R> ################################
R> sumterm=rep(0,n1)
R> for (i in 1:n1) {
     outi<-matrix(IMatch[[i]],ncol=3)
     JMi<-nrow(outi) #number of obs's used for matching
     y0l<-outi[,2]
     int<-sum((y1[i]-y0l-ATTc)^2) #y0l is JMi by 1, the other terms are scalars
     sumterm[i]<-(1/JMi)*int
 }
R> # compute variance
R> ################################
R> sighat<-(1/(2*n1))*sum(sumterm)
R> VarATTc<-(sighat/n1^2)*(n1+sum(KMlvec^2))
R> seATTc<-sqrt(VarATTc)
R> tATTc<-ATTc/seATTc
```

TABLE 2. Combined estimation results for ATT

estimator	estimate	s.e.	t-value
min. # matches	1.000	-	-
# vars, exact match	1.000	-	-
% exact matches	0.993	-	-
ATT, uncorrected	2.048	0.814	2.517
ATT, corrected	1.989	0.814	2.445

```
R> proc.time()-tic
   user  system elapsed
   1.53    0.21    1.77
```

REFERENCES

Diamond, A. and Sekhon, J. (2013). Genetic matching for estimating causal effects: A general multivariate matching method for achieving balance in observational studies, *The Review of Economics and Statistics* **95**: 932–945.

Ho, D., Kosuke, I., King, G. and Stuart, E. (2007). Matching as nonparametric preprocessing for reducing model dependence in parametric causal inference, *Policitcal Analysis* **15**: 199–236.