R Notebook

Notebook explaining the Job Training program analysis for ECON3210. Written by Kyler Blackburn.

```r
library(readr)
job_training <- read_csv("C:/Users/Kai/Coding/UNSW/ECON3210/econ3210/data/job_training.csv")
```

```
## New names:
## Rows: 19204 Columns: 12
## -- Column specification
## -------------------------------------------------------- Delimiter: "," dbl
## (12): ...1, treated, age, educ, black, hisp, married, nodegree, urban, f...
## i Use `spec()` to retrieve the full column specification for this data. i
## Specify the column types or set `show_col_types = FALSE` to quiet this message.
## * `` -> `...1`
```

We need to do some data manipulation to convert this dataframe into a shape that's useful for *DiD*.

```r
#creating DiD dataframe
newdf <- subset(job_training, select = -c(re14, re18))
newdf <- rbind(newdf, newdf)
newdf$tgroup <- ifelse(newdf$treated == 1,1,0)
re18.df <- data.frame(realIncome = job_training$re18, period = c(rep(1,nrow(job_training))))
re14.df <- data.frame(realIncome = job_training$re14, period = c(rep(0,nrow(job_training))))
newdf <- cbind(newdf,rbind(re14.df,re18.df))
newdf$treated[newdf$period == 0] <- 0

dim(newdf)
```

```
## [1] 38408    13
```

```r
head(newdf,5)
```

```
##   ...1 treated age educ black hisp married nodegree urban fsize tgroup
## 1    1       0  42   16     0    0       1        0     1     2      0
## 2    2       0  20   13     0    0       0        0     1     1      0
## 3    3       0  37   12     0    0       1        0     1     4      0
## 4    4       0  48   12     0    0       1        0     1     3      0
## 5    5       0  51   12     0    0       1        0     0     2      0
##   realIncome period
## 1      0.000      0
## 2   3317.468      0
## 3  22781.855      0
## 4  20839.355      0
## 5  21575.178      0
```

Let's start with a fixed-effects OLS DiD regression, without weights, just controls interacted with period.

```
library(fixest)
#first estimation
feols(realIncome ~ tgroup:period | factor(...1) + period[age, educ, black, hisp, married, nodegree, urba
```

```
## OLS estimation, Dep. Var.: realIncome
## Observations: 38,408
## Fixed-effects: factor(...1): 19,204,  period: 2
## Varying slopes: age (period: 2),  educ (period: 2),  black (period: 2),  hisp (period: 2),  married
## Standard-errors: IID
##                Estimate Std. Error   t value Pr(>|t|)
## tgroup:period -58.6718    489.823 -0.119782  0.90466
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 4,001.7     Adj. R2: 0.714589
##                 Within R2: 7.478e-7
```

It seems like this result above is statistically insignificant. Let's continue improving the model.

Use the `ebal` library to calculate entropy-balanced weights. Re-estimate the FEOLS as before but with weights.

```
#ebalance
library(ebal)
```

```
## ##
## ## ebal Package: Implements Entropy Balancing.

## ## See http://www.stanford.edu/~jhain/ for additional information.
```

```
X <- subset(newdf, select = -c(...1,treated,realIncome,period,tgroup))
eb <- ebalance(Treatment = newdf$tgroup, X = X)
```

```
## Converged within tolerance
```

```
newdf$weights <- 1
newdf$weights[newdf$tgroup==0] <- eb$w
#estimate with ebalance
feols(realIncome ~ tgroup:period | factor(...1) + period[age, educ, black, hisp, married, nodegree, urba
```

```
## OLS estimation, Dep. Var.: realIncome
## Observations: 38,408
## Weights: newdf$weights
## Fixed-effects: factor(...1): 19,204,  period: 2
## Varying slopes: age (period: 2),  educ (period: 2),  black (period: 2),  hisp (period: 2),  married
## Standard-errors: IID
##                Estimate Std. Error t value   Pr(>|t|)
## tgroup:period  463.193    109.161 4.24322 2.2137e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 664.8     Adj. R2: 0.465101
##               Within R2: 9.376e-4
```

Our result is now statistically significant! We're not done: let's try a DML approach. First, create the new target and predictors: * We are calculating ΔY for our target. * Let's create up to the third power of continuous features.

```
#running double ML
job_training$incChange <- job_training$re18 - job_training$re14

job_training$age2 <- (job_training$age)^2
job_training$age3 <- (job_training$age)^3
job_training$educ2 <- (job_training$educ)^2
job_training$educ3 <- (job_training$educ)^3
job_training$fsize2 <- (job_training$fsize)^2
job_training$fsize3 <- (job_training$fsize)^3

newdf$age2 <- (newdf$age)^2
newdf$age3 <- (newdf$age)^3
newdf$educ2 <- (newdf$educ)^2
newdf$educ3 <- (newdf$educ)^3
newdf$fsize2 <- (newdf$fsize)^2
newdf$fsize3 <- (newdf$fsize)^3
```

To create the `XML` feature matrix object, we add all these features (as well as logarithms), as well as second degree interactions of all variables.

```
XML <-      (model.matrix(~ -1 + age + age2 + age3 + log(age) + educ + educ2 + educ3 + log(educ) + fsize
XML.long <- (model.matrix(~ -1 + age + age2 + age3 + log(age) + educ + educ2 + educ3 + log(educ) + fsize

head(XML, 1)
```

```
##   age age2  age3 log(age) educ educ2 educ3 log(educ) fsize log(fsize) fsize2
## 1  42 1764 74088  3.73767   16   256  4096  2.772589     2  0.6931472      4
##   fsize3 black hisp married nodegree urban age:fsize educ:fsize fsize:black
## 1      8     0    0       1        0     1        84         32           0
##   fsize:hisp fsize:married fsize:nodegree fsize:urban age:educ age:black
## 1          0             2              0           2      672         0
##   age:hisp age:married age:nodegree age:urban educ:black educ:hisp educ:married
## 1        0          42            0        42          0         0           16
##   educ:nodegree educ:urban black:hisp black:married black:nodegree black:urban
## 1             0         16          0             0              0           0
##   hisp:married hisp:nodegree hisp:urban married:nodegree married:urban
## 1            0             0          0                0             1
##   nodegree:urban
## 1              0
```

Using Lasso, let's do automatic feature selection on both the treatment and ΔY.

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loaded glmnet 4.1-7
```

```r
DPred <- cv.glmnet(XML,job_training$treated)
coef(DPred)
```

```
## 46 x 1 sparse Matrix of class "dgCMatrix"
##                              s1
## (Intercept)       1.455764e-02
## age               .
## age2              .
## age3              .
## log(age)          .
## educ              .
## educ2             .
## educ3             .
## log(educ)         .
## fsize             .
## log(fsize)        .
## fsize2            .
## fsize3            .
## black             1.003857e-03
## hisp              .
## married          -8.787944e-03
## nodegree          .
## urban             .
## age:fsize         .
## educ:fsize        .
## fsize:black       .
## fsize:hisp        .
## fsize:married     .
## fsize:nodegree    .
## fsize:urban       .
## age:educ          .
## age:black         .
## age:hisp          .
## age:married      -7.953855e-05
## age:nodegree      .
## age:urban         .
## educ:black        5.038707e-03
## educ:hisp         .
## educ:married      .
## educ:nodegree     .
## educ:urban        .
## black:hisp        .
## black:married    -3.945450e-02
## black:nodegree    7.592038e-02
## black:urban       .
## hisp:married      .
## hisp:nodegree     .
## hisp:urban        .
## married:nodegree  .
## married:urban     .
## nodegree:urban    .
```

```r
YPred <- cv.glmnet(XML,job_training$incChange)
coef(YPred)
```

```
## 46 x 1 sparse Matrix of class "dgCMatrix"
##                          s1
## (Intercept)        8947.237
## age                    .
## age2                   .
## age3                   .
## log(age)          -2188.912
## educ                   .
## educ2                  .
## educ3                  .
## log(educ)              .
## fsize                  .
## log(fsize)             .
## fsize2                 .
## fsize3                 .
## black                  .
## hisp                   .
## married                .
## nodegree               .
## urban                  .
## age:fsize              .
## educ:fsize             .
## fsize:black            .
## fsize:hisp             .
## fsize:married          .
## fsize:nodegree         .
## fsize:urban            .
## age:educ               .
## age:black              .
## age:hisp               .
## age:married            .
## age:nodegree           .
## age:urban              .
## educ:black             .
## educ:hisp              .
## educ:married           .
## educ:nodegree          .
## educ:urban             .
## black:hisp             .
## black:married          .
## black:nodegree         .
## black:urban            .
## hisp:married           .
## hisp:nodegree          .
## hisp:urban             .
## married:nodegree       .
## married:urban          .
## nodegree:urban         .
```

Let's take these selected features and include them in this next model. We recalculate `ebal` weights based on these selected covariates.

```r
DML.DF <- subset(as.data.frame(XML.long),select = c(1,4,13,15,28,31,37,38))
ebML <- ebalance(Treatment = newdf$tgroup, X = DML.DF)
```

```
## Converged within tolerance
```

```r
newdf$weightsML <- 1
newdf$weightsML[newdf$tgroup==0] <- ebML$w
```

Finally, let's try running this FEOLS without and with `ebal` weights. The second regression (with weights), is statistically significant!

```r
#double ML
feols(realIncome ~ tgroup:period | factor(...1) + period[age, log(age), black, married, age:married, edu
```

```
## OLS estimation, Dep. Var.: realIncome
## Observations: 38,408
## Fixed-effects: factor(...1): 19,204,  period: 2
## Varying slopes: age (period: 2),  log(age) (period: 2),  black (period: 2),  married (period: 2),  ag
## Standard-errors: IID
##               Estimate Std. Error  t value Pr(>|t|)
## tgroup:period  408.072    500.495 0.815337  0.41489
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 3,995.2     Adj. R2: 0.715504
##                 Within R2: 3.465e-5
```

```r
#double ML + ebalance weights
feols(realIncome ~ tgroup:period | factor(...1) + period[age, log(age), black, married, age:married, edu
```

```
## OLS estimation, Dep. Var.: realIncome
## Observations: 38,408
## Weights: newdf$weightsML
## Fixed-effects: factor(...1): 19,204,  period: 2
## Varying slopes: age (period: 2),  log(age) (period: 2),  black (period: 2),  married (period: 2),  ag
## Standard-errors: IID
##               Estimate Std. Error t value  Pr(>|t|)
## tgroup:period  587.586    111.129 5.28744 1.254e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 676.8     Adj. R2: 0.475967
##               Within R2: 0.001455
```