IS IT WORTH TO ATTEND SUMMER RESEARCH CAMP? A PROPENSITY SCORE MATCHING APPROACH TO EDUCATIONAL ATTAINMENT DETERMINANTS

Contents

0. Pre.

Change name and class to numeric:

```
data$sex<-as.numeric(data$SEX)
data$video<-as.numeric(data$video)
data$treat<-as.numeric(data$treat)
data$anx<-as.numeric(data$anx)
data$bor<-as.numeric(data$bor)
data$summer<-as.numeric(data$summer)
data$gpa<-as.numeric(data$HSGPA)
data$mat<-as.numeric(data$SATM)
data$income<-as.numeric(data$INCOME)
data$fg<-as.numeric(data$FIRSTGEN)
data$religion<-as.numeric(data$religion)
data$white<-as.numeric(data$white)
data$act<-as.numeric(data$ACTCOMP)

attach(data)
```

```
## The following objects are masked from data (pos = 3):
##
##     ACTCOMP, anx, bor, FIRSTGEN, HPW01_T2, HPW05, HPW09, HPW14, HPW15,
##     HSGPA, HSTYPE1, HSTYPE2, INCOME, religion, SATM, SEX, SUBJID,
##     summer, treat, video, white, YEAR
```

```r
class(gpa)
```

```
## [1] "numeric"
```

Vars.

Dependent variable: ACT composite (mean score at ACT test):

```r
summary(act)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00   23.00   27.00   26.37   30.00   36.00
```

```r
summary(gpa)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   6.000   7.000   6.867   8.000   8.000
```

The treatment here is going or not to a research camp during the summer:

```r
summary(summer)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.00000 0.00000 0.09881 0.00000 1.00000
```

Where 0=going and 1= not going.

Those students going to summer camp are assummed to be good students that perform above average in high school. This kind of camps are suppossed to be more than nice to improve student's knowlegde and performance.

1. Descriptive tables.

##. Values. ### Mean outcomes for treated/untreated population by bor and gender.

compute the means:

```r
mean<-mean(mat)

## Belong, male.
mean1<-mean(data.matrix(data[data$sex==1 & data$summer==1 & data$white==0,"mat"]))
mean2<-mean(data.matrix(data[data$sex==1 & data$summer==0 & data$white==0,"mat"]))
## Belong, female.
mean3<-mean(data.matrix(data[data$sex==2 & data$summer==1 & data$white==0,"mat"]))
mean4<-mean(data.matrix(data[data$sex==2 & data$summer==0 & data$white==0,"mat"]))
```

```r
## Belong, male.
mean5<-mean(data.matrix(data[data$sex==1 & data$summer==1 & data$white==1,"mat"]))
mean6<-mean(data.matrix(data[data$sex==1 & data$summer==0 & data$white==1,"mat"]))
## Belong, female
mean7<-mean(data.matrix(data[data$sex==2 & data$summer==1 & data$white==1,"mat"]))
mean8<-mean(data.matrix(data[data$sex==2 & data$summer==0 & data$white==1,"mat"]))
```

```r
d1<-mean1-mean2
d2<-mean3-mean4
d3<-mean5-mean6
d4<-mean7-mean8
```

Distrubution of treatment in population by FG and gender.

```r
p<-length(mat)

## First gen, male.
p1<-length(data.matrix(data[data$sex==1 & data$summer==1 & data$white==0,"mat"]))/p
p2<-length(data.matrix(data[data$sex==1 & data$summer==0 & data$white==0,"mat"]))/p
## First gen, female.
p3<-length(data.matrix(data[data$sex==2 & data$summer==1 & data$white==0,"mat"]))/p
p4<-length(data.matrix(data[data$sex==2 & data$summer==0 & data$white==0,"mat"]))/p

## No first gen, male.
p5<-length(data.matrix(data[data$sex==1 & data$summer==1 & data$white==1,"mat"]))/p
p6<-length(data.matrix(data[data$sex==1 & data$summer==0 & data$white==1,"mat"]))/p
## No first gen, female
p7<-length(data.matrix(data[data$sex==2 & data$summer==1 & data$white==1,"mat"]))/p
p8<-length(data.matrix(data[data$sex==2 & data$summer==0 & data$white==1,"mat"]))/p

t1<-sum(p1+p2)
t2<-sum(p3+p4)
t3<-sum(p5+p6)
t4<-sum(p7+p8)

sum(p1+p2+p3+p4+p5+p6+p7+p8)
```

```
## [1] 1
```

Tables.

a. Mean outcomes.

Table 1. WHITE ACT by gender/treatment.

```r
FGmean<-matrix(c(mean1, mean2, d1, mean5, mean6, d2), ncol=3, byrow=T)

colnames(FGmean) <- c("Untreated","Treated","Difference")
rownames(FGmean) <- c("Male", "Female")
FGmean <- as.table(FGmean)
FGmean
```

```
##        Untreated   Treated Difference
## Male   652.99869 632.32822   20.67047
## Female 611.98187 605.39313   22.92794
```

Table 2. NON WHITE ACT by gender/treatment.

```r
NFmean<-matrix(c(mean5, mean6, d3, mean7, mean8, d4), ncol=3, byrow=T)

colnames(NFmean) <- c("Untreated","Treated","Total")
rownames(NFmean) <- c("Male", "Female")
```

```r
NFmean <- as.table(NFmean)
NFmean
```

```
##         Untreated    Treated      Total
## Male   611.981865 605.393131   6.588735
## Female 566.795026 560.355922   6.439103
```

b. Distribution of outcomes.

Table 1. WHITE ACT by gender/treatment.

```r
FGds<-matrix(c(p1, p2, t1, p5, p6, t3), ncol=3, byrow=T)

colnames(FGds) <- c("Treated","Non Treated","Difference")
rownames(FGds) <- c("Male", "Female")
FGds <- as.table(FGds)
FGds
```

```
##           Treated Non Treated Difference
## Male   0.03010119  0.30488260 0.33498379
## Female 0.01516848  0.08466451 0.09983299
```

Table 2. Non WHITE ACT by gender/treatment.

```r
NFds<-matrix(c(p5, p6, t3, p7, p8, t4), ncol=3, byrow=T)

colnames(NFds) <- c("Treated","Non Treated","Difference")
rownames(NFds) <- c("Male", "Female")
NFds <- as.table(NFds)
NFds
```

```
##           Treated Non Treated Difference
## Male   0.01516848  0.08466451 0.09983299
## Female 0.02290991  0.13685038 0.15976029
```

2. Difference in means: outcome variable.

First we standarise mat:

```r
smat<-(mat-mean(mat))/ sd(mat)
mean(smat)
```

```
## [1] 4.556088e-16
```

```r
sd(smat)
```

```
## [1] 1
```

Independent variable is going to summer camp:

```r
summary(summer)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.00000 0.00000 0.00000 0.09881 0.00000 1.00000
```

differences in means:

```r
library(dplyr)
```

```
##
```

```
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data %>%
  group_by(summer)  %>%
  summarise (n_students=n(),
             mean_smat= mean(smat),
             std_error=sd(smat)/ sqrt(n_students))
```

```
## # A tibble: 2 x 4
##   summer n_students mean_smat std_error
##    <dbl>      <int>     <dbl>     <dbl>
## 1      0      45866  4.56e-16   0.00467
## 2      1       5029  4.56e-16   0.0141
```

Non-std.

```
data %>%
  mutate(test = (mat - mean(mat)) / sd(mat)) %>% #this is how the math score is standardized
  group_by(summer) %>%
  summarise(mean_mat = mean(test))
```

```
## # A tibble: 2 x 2
##   summer mean_mat
##    <dbl>    <dbl>
## 1      0  -0.0130
## 2      1   0.119
```

The difference-in-means is statistically significant at conventional levels of confidence (as is also evident from the small standard error above):

```
with(data, t.test(smat ~ summer))
```

```
##
##  Welch Two Sample t-test
##
## data:  smat by summer
## t = -8.3761, df = 6021.5, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.1625796 -0.1009117
## sample estimates:
## mean in group 0 mean in group 1
##     -0.01301796      0.11872771
```

3. Difference in means: pre-treatment covariates.

Let's calculate the mean for each covariate by the treatment status:

```
ecls_cov <- c('sex', 'income', 'fg', 'religion', 'bor', 'white', 'gpa')
data %>%
```

```
  group_by(summer) %>%
  select(one_of(ecls_cov)) %>%
  summarise_all(funs(mean(., na.rm = T)))
```

```
## Adding missing grouping variables: `summer`

## Warning: funs() is soft deprecated as of dplyr 0.8.0
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once per session.

## # A tibble: 2 x 8
##   summer   sex income    fg religion   bor white   gpa
##    <dbl> <dbl>  <dbl> <dbl>    <dbl> <dbl> <dbl> <dbl>
## 1      0  1.57   20.8  1.11     1.21 0.442 0.246  6.86
## 2      1  1.54   20.5  1.12     1.21 0.411 0.385  6.97
```

4. Propensity score estimation.

We estimate the propensity score by running a logit model (probit also works) where the outcome variable is a binary variable indicating treatment status. What covariates should you include? For the matching to give you a causal estimate in the end, you need to include any covariate that is related to both the treatment assignment and potential outcomes. I choose just a few covariates below—they are unlikely to capture all covariates that should be included. You'll be asked to come up with a potentially better model on your own later.

```
ecls_cov
```

```
## [1] "sex"      "income"   "fg"       "religion" "bor"      "white"    "gpa"
```

```
m_ps <- glm(summer ~ sex + income+fg+religion+bor+white+gpa,
            family = binomial(), data = data)
summary(m_ps)
```

```
##
## Call:
## glm(formula = summer ~ sex + income + fg + religion + bor + white +
##     gpa, family = binomial(), data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.6725  -0.4647  -0.4211  -0.3893   2.5311
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.090557   0.141004 -21.918  < 2e-16 ***
## sex         -0.168904   0.030378  -5.560  2.7e-08 ***
## income       0.006699   0.002433   2.753  0.00591 **
```

```
## fg           0.011749   0.048520   0.242  0.80867
## religion     0.037954   0.036612   1.037  0.29990
## bor         -0.093737   0.030500  -3.073  0.00212 **
## white        0.719607   0.032652  22.039  < 2e-16 ***
## gpa          0.109958   0.012706   8.654  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 32824  on 50894  degrees of freedom
## Residual deviance: 32274  on 50887  degrees of freedom
## AIC: 32290
##
## Number of Fisher Scoring iterations: 5
```

```
prs_df <- data.frame(pr_score = predict(m_ps, type = "response"),
                     summer = m_ps$model$summer)
head(prs_df)
```

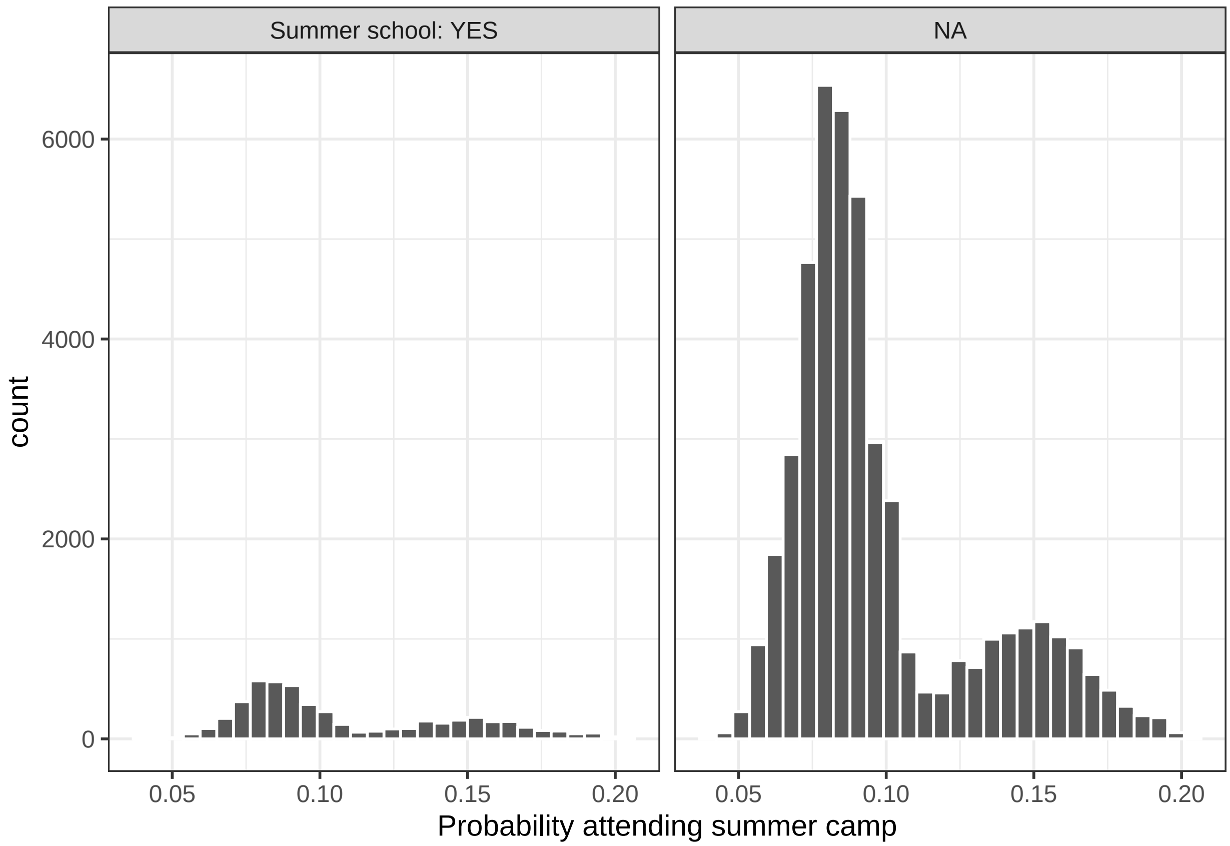
```
##     pr_score summer
## 1 0.16004599      0
## 2 0.06464719      0
## 3 0.12692770      0
## 4 0.07153383      0
## 5 0.15996666      0
## 6 0.14243286      0
```

5. Region of common support.

After estimating the propensity score, it is useful to plot histograms of the estimated propensity scores by treatment status:

```
labs <- paste("Summer school:", c("YES", "NA"))
library(ggplot2)
prs_df %>%
  mutate(summer = ifelse(summer == 0, labs[0], labs[1])) %>%
  ggplot(aes(x = pr_score)) +
  geom_histogram(color = "white") +
  facet_wrap(~summer) +
  xlab("Probability attending summer camp") +
  theme_bw()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



6. Matching algorithm

The method we use below is to find pairs of observations that have very similar propensity scores, but that differ in their treatment status. We use the package MatchIt for this. This package estimates the propensity score in the background and then matches observations based on the method of choice ("nearest" in this case).

```
ecls_nomiss <- data %>%  # MatchIt does not allow missing values
  select(mat, summer, one_of(ecls_cov)) %>%
  na.omit()

library(MatchIt)
mod_match <- matchit(summer ~ sex + income+fg+religion+bor+white+gpa,
                     method = "nearest", data = ecls_nomiss)
```

To create a dataframe containing only the matched observations, use the match.data() function:

```
dta_m <- match.data(mod_match)
dim(dta_m)
```

```
## [1] 10058    11
```

7. Examining covariate balance in the matched sample

```
dta_m %>%
  group_by(summer) %>%
```

```r
  select(one_of(ecls_cov)) %>%
  summarise_all(funs(mean))
```

```
## Adding missing grouping variables: `summer`

## # A tibble: 2 x 8
##   summer   sex income    fg religion   bor white   gpa
##    <dbl> <dbl>  <dbl> <dbl>    <dbl> <dbl> <dbl> <dbl>
## 1      0  1.54   20.5  1.12     1.21 0.412 0.384  6.98
## 2      1  1.54   20.5  1.12     1.21 0.411 0.385  6.97
```

You can test this more formally using t-tests. Ideally, we should not be able to reject the null hypothesis of no mean difference for each covariate:

```r
lapply(ecls_cov, function(v) {
    t.test(dta_m[, v] ~ dta_m$summer)
})
```

```
## [[1]]
##
##  Welch Two Sample t-test
##
## data:  dta_m[, v] by dta_m$summer
## t = 0.18012, df = 10056, p-value = 0.8571
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.01768611  0.02126535
## sample estimates:
## mean in group 0 mean in group 1
##        1.543647        1.541857
##
##
## [[2]]
##
##  Welch Two Sample t-test
##
## data:  dta_m[, v] by dta_m$summer
## t = -0.08146, df = 10056, p-value = 0.9351
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.2940429  0.2705790
## sample estimates:
## mean in group 0 mean in group 1
##        20.53251        20.54424
##
##
## [[3]]
##
##  Welch Two Sample t-test
##
## data:  dta_m[, v] by dta_m$summer
## t = -0.36417, df = 10055, p-value = 0.7157
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.01523009  0.01045777
## sample estimates:
```

```
## mean in group 0 mean in group 1
##        1.121893        1.124279
##
##
## [[4]]
##
##  Welch Two Sample t-test
##
## data:  dta_m[, v] by dta_m$summer
## t = -0.097411, df = 10056, p-value = 0.9224
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.01680086  0.01521009
## sample estimates:
## mean in group 0 mean in group 1
##        1.212567        1.213362
##
##
## [[5]]
##
##  Welch Two Sample t-test
##
## data:  dta_m[, v] by dta_m$summer
## t = 0.10129, df = 10056, p-value = 0.9193
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.01824735  0.02023581
## sample estimates:
## mean in group 0 mean in group 1
##       0.4124080       0.4114138
##
##
## [[6]]
##
##  Welch Two Sample t-test
##
## data:  dta_m[, v] by dta_m$summer
## t = -0.14345, df = 10056, p-value = 0.8859
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.02041213  0.01762828
## sample estimates:
## mean in group 0 mean in group 1
##       0.3839730       0.3853649
##
##
## [[7]]
##
##  Welch Two Sample t-test
##
## data:  dta_m[, v] by dta_m$summer
## t = 0.57396, df = 10051, p-value = 0.566
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
```

```
##  -0.03361792  0.06145645
## sample estimates:
## mean in group 0 mean in group 1
##        6.981706        6.967787
```

No difference of means now.

8. Estimating treatment effects

Estimating the treatment effect is simple once we have a matched sample that we are happy with. We can use a t-test:

```
with(dta_m, t.test(mat ~ summer))
```

```
##
##  Welch Two Sample t-test
##
## data:  mat by summer
## t = -6.3973, df = 10042, p-value = 1.652e-10
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -16.512394  -8.766588
## sample estimates:
## mean in group 0 mean in group 1
##        603.2977        615.9372
```

8.1. simple ols

Or we can use OLS with or without covariates:

```
lm_treat1 <- lm(mat ~ summer, data = dta_m)
summary(lm_treat1)
```

```
##
## Call:
## lm(formula = mat ~ summer, data = dta_m)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -415.94  -65.94    6.70   74.06  196.70
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  603.298      1.397 431.828  < 2e-16 ***
## summer        12.639      1.976   6.397 1.65e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 99.07 on 10056 degrees of freedom
## Multiple R-squared:  0.004053,   Adjusted R-squared:  0.003954
## F-statistic: 40.92 on 1 and 10056 DF,  p-value: 1.652e-10
```

Clear effect of tratment!

8.2. Multiple ols.

```r
lm_treat2 <- lm(mat ~ summer + sex + income+fg+religion+bor+white+gpa, data = dta_m)
summary(lm_treat2)
```

```
##
## Call:
## lm(formula = mat ~ summer + sex + income + fg + religion + bor +
##     white + gpa, data = dta_m)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -410.73  -53.78    2.10   55.32  418.02
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 395.0763     7.7029  51.289  < 2e-16 ***
## summer       13.0608     1.6257   8.034 1.05e-15 ***
## sex         -41.3255     1.6626 -24.855  < 2e-16 ***
## income        2.6815     0.1267  21.164  < 2e-16 ***
## fg          -30.5865     2.6734 -11.441  < 2e-16 ***
## religion     23.3760     1.9950  11.717  < 2e-16 ***
## bor          16.8785     1.6665  10.128  < 2e-16 ***
## white       -11.5740     1.7925  -6.457 1.12e-10 ***
## gpa          31.5693     0.6821  46.285  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 81.52 on 10049 degrees of freedom
## Multiple R-squared:  0.3263, Adjusted R-squared:  0.3257
## F-statistic: 608.3 on 8 and 10049 DF,  p-value: < 2.2e-16
```

Conclusion: clar effect of going to a summer camp, even when comparing between those students with similar characteristics.