Quant II Recitation

Antonella Bandiera aab639@nyu.edu

February, 2018

Homework

Question 1

- Most of you got the first part
- Some of you tried stuff that was a bit complicated (unnecessary)
  - You should just note the difference between N and n
  - If N increases, the estimator is still a single draw from that population, that is either $2Y_{i1}$ or $-2Y_{0i}$

Homework

- For question 2
- Consider a stratified estimator that controls for Z_i by
  (i) partitioning the sample by values of Z_i, then
  (ii) taking the difference in treated and control means within each of these strata, and then
  (iii) combining these stratum-specific estimates with a weighted average, where we weight each stratum contribution by the share of the P in each stratum

Notation and Setup

- So we consider the following two expectations:
  - $E[Y_i(1) - Y_i(0)|Z_i = 1]$ weighted by $p_Z = P(Z_i = 1)$
  - $E[Y_i(1) - Y_i(0)|Z_i = 0]$ weighted by $1 - p_Z$
- Then we want the weighted sum to be $E[Y_i(1) - Y_i(0)]$
- Homework: Is this possible?

Decompose to Principal Strata

- Within the stratum $Z = 1$, we have the following:
  - $p_{comp} = P(D_i(1) - D_i(0) = 1)$
  - $p_{NT} = P(D_i(1) = D_i(0) = 0)$
  - $p_{AT} = P(D_i(1) = D_i(0) = 1)$
  - $p_{def} = P(D_i(1) - D_i(0) = -1) = 0$
- And these probabilities are equal (in expectation) across strata defined by Z due to random assignment

Principal Strata TEs

- Each principal strata may have its own conditional average treatment effect
  - $\rho_{comp} = E[Y_i(1) - Y_i(0)|D_i(1) - D_i(0) = 1]$
  - $\rho_{NT} = E[Y_i(1) - Y_i(0)|D_i(1) = D_i(0) = 0]$
  - $\rho_{AT} = E[Y_i(1) - Y_i(0)|D_i(1) = D_i(0) = 1]$
  - $\rho_{def} = E[Y_i(1) - Y_i(0)|D_i(1) - D_i(0) = -1]$
- We don't assume anything about these effects.
- Also note that these are equal across strata in Z due to random assignment of Z
- More important, these effects assume counterfactual conditions in treatment that we don't observe

Counterfactuals and Principal Strata

- For instance, for never takers we don't observe counterfactual conditions:
  - $E[Y_i(D_i(1)) - Y_i(D_i(0))|D_i(1) = D_i(0) = 0]$
  - This observed quantity may be simplified: $E[Y_i(0) - Y_i(0)|D_i(1) = D_i(0) = 0]$
  - Which is equal to zero.
  - The same is true for always takers.

Complier TEs

- This is not the case for compliers, though.
  - $E[Y_i(D_i(1)) - Y_i(D_i(0))|D_i(1) - D_i(0) = 1]$
  - This can be similar simplified to:
  - $E[Y_i(1) - Y_i(0)|D_i(1) - D_i(0) = 1]$
- And we've assumed that there are no defiers.

Principal Strata TEs

- To get at out target, we must be able to recover $E[Y_i(1) - Y_i(0)|Z = 1]$ and vice versa within each strata. This would allow:
  - $E[Y_i(1) - Y_i(0)|Z = 0](1 - p_Z) + E[Y_i(1) - Y_i(0)|Z = 1]p_Z$
- But can we get that?
- NO!

What's in a strata?

- For $Z = 0$, and our three principal strata, we have:
  - Always Takers will be $D_i = 1$
  - Never takers will be $D_i = 0$
  - Compliers will be $D_i = 0$
- So we can decompose the difference in means is as follows:
  - $E[Y_i(1)|Z = 0] - E[Y_i(0)|Z = 0] = E[Y_i(1)|D_i(1) = D_i(0) = 1] - [E[Y_i(0)|D_i(1) - D_i(0) = 1]\frac{p_{comp}}{p_{NT} + p_{comp}} + E[Y_i(0)|D_i(1) = D_i(0) = 0]\frac{p_{NT}}{p_{NT} + p_{comp}}]$
- So the estimated treatment effect *within* the control strata is the difference between the always-takers and the sum of the outcomes for never-takers and compliers, weighted by their proportions
- The key point is that these counterfactuals are not the ones we want.
- Even if they were, we still wouldn't know what we were estimating without knowing proportions in each strata (which we wouldn't).

Finally...

- The last part of the proof shows that the weighted combination of the stratified estimators, $E[\hat{\rho}]$ is not equivalent to ρ
- What is equivalent to?

Covariate Adjustment in sampling

- Lin, Winston. (2013) "Agnostic Notes on Regression Adjustments to Experimental Data: Reexamining Freedman's Critique" *Annals of Applied Statistics*. 7(1):295-318.
- Lin shows that OLS adjustment (using a full set of treatment-covariate interactions) cannot hurt asymptotic precision
- We use interactions because we are adjusting for groups separately (especially important if groups are of unequal size)
- We recenter the covariate to have zero mean for ease of interpretation
- A computational shortcut is to regress Y on T, X, and $T \times (X - \bar{X})$, or equivalently to regress Y on T, $X - \bar{X}$, and $T \times (X - \bar{X})$, where $\bar{X}$ is the mean covariate value for the entire sample.
- Then the coefficient on T estimates the average treatment effect (ATE) for the entire sample.

Why do we do it?

- If we are doing an RCT is not necessary, ADJUSTING WILL NOT REDUCE BIAS
- But covariate adjustment will improve precision if the covariates are good predictors of the outcome
- Think about it, in large samples random assignment produces similar groups
- Still, by chance, some group might be better educated, have more income, be less politically active, etc.
- The ATE is subject to this sampling variability, which creates more noise (not bias!)
- Controlling for these covariates will improve precision
- Let's look at an example with a simulation that Drew put together
- Similar to the empirical example in Lin (2013)

Simulation

```
#Variables which govern the size of the simulation
# And the size of our causal effects
nclass <- 5
nstudent <- 25
Eff <- 5
EffSD <- 3
# Simulate data
set.seed(1977)
Yr1ClassType <- rep(c(1,0),nclass*nstudent)
Yr2ClassType <- sample(Yr1ClassType,replace=FALSE)
Yr1Score <- rnorm(2*nclass*nstudent,76+Yr1ClassType*5,9)
# Fixed margins randomization
Trt  <- sample(Yr1ClassType,replace=FALSE)
```

Simulation

```
#There is an independent effect of class type each year
# Variance is different across class types in year 2
CtlOutcome <- rnorm(2*nclass*nstudent,Yr1Score+
                      Yr2ClassType*3,9-Yr2ClassType*4)
# Treatment effect is random, but with expectation Eff
Yr2Obs <- CtlOutcome +
  Trt * rnorm(2*nclass*nstudent,Eff,EffSD)

summary(lm(Yr2Obs~Trt))$coefficients[2,]
```

```
##    Estimate  Std. Error     t value    Pr(>|t|)
## 4.307282238 1.558184168 2.764295984 0.006132827
```

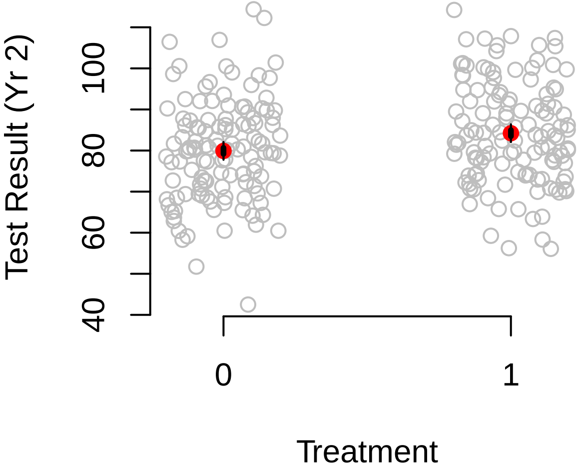
```
summary(lm(Yr2Obs~Trt+Yr1Score))$coefficients[2,]
```

```
##     Estimate   Std. Error      t value     Pr(>|t|)
## 3.5206647114 1.0064194358 3.4982081884 0.0005553714
```

Plot Data

```
pdf("scatter.pdf", 3.5,3.5)
plot(jitter(Trt),Yr2Obs,axes=F,xlab="Treatment",ylab="Test
axis(2)
axis(1,at=c(0,1))
# Calculate quantities for plotting CIs
mns <- tapply(Yr2Obs,Trt,mean)
# SEs could also be pulled from the linear models we fit a
ses <- tapply(Yr2Obs,Trt,function(x) sd(x)/sqrt(length(x))
points(c(0,1),mns,col="red",pch=19)
# Note the loop so that I only write this code once
for(tr in unique(Trt)) {
  for(q in c(.25,.025)) {
    upr<-mns[as.character(tr)]+qnorm(1-q)*ses[as.character(
    lwr <- mns[as.character(tr)]-qnorm(1-q)*ses[as.characte
    segments(tr,upr,tr,lwr,lwd=(-4/log(q)))
  }
}
dev.off()
```

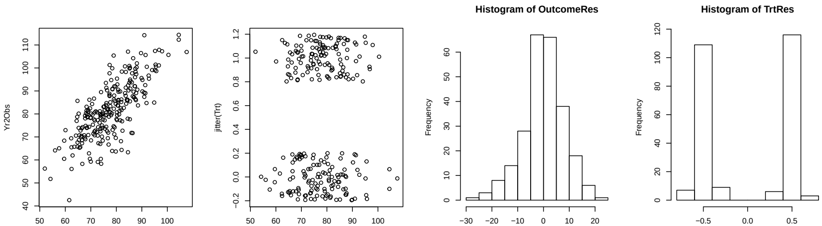
Plot Data



Partial Regression

- A way to make this plot a little bit more friendly is to remove some variance from the outcome
- First some classic diagnostics

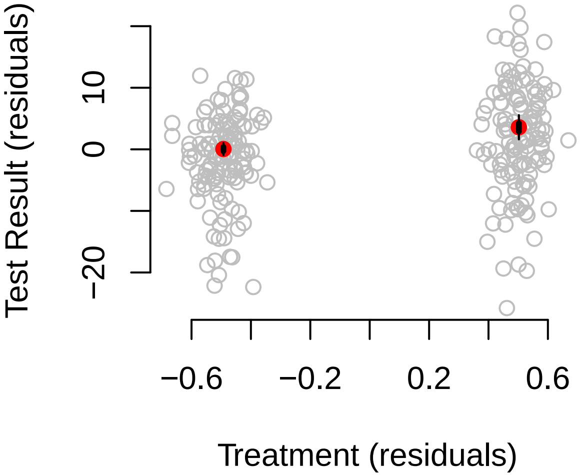
```
OutcomeRes <- residuals(lm(Yr2Obs~Yr1Score+0))
TrtRes <- residuals(lm(Trt~Yr1Score+0))
# Diagnostics
par(mfrow=c(1,4))
plot(Yr1Score,Yr2Obs)
plot(Yr1Score,jitter(Trt))
hist(OutcomeRes)
hist(TrtRes)
```



Residualized Plot

```
pdf("residualized.pdf", 3.5,3.5)
par(mfrow=c(1,1))
plot(jitter(TrtRes),OutcomeRes,axes=F,xlab="Treatment (res
axis(2)
axis(1)
# Pull information from the new bivariate model
mns<-coef(lm(OutcomeRes~TrtRes))
ses<-summary(lm(OutcomeRes~TrtRes))$coefficients[,2]
TrtResMns<-tapply(TrtRes,Trt,mean)
names(ses)<-names(mns)<-names(TrtResMns)
points(TrtResMns,mns,col="red",pch=19)
for(tr in names(TrtResMns)) {
  for(q in c(.25,.025)) {
    upr<-mns[tr]+qnorm(1-q)*ses[tr]
    lwr <- mns[tr]-qnorm(1-q)*ses[tr]
    segments(TrtResMns[tr],upr,TrtResMns[tr],lwr,lwd=(-4/lo
  }
}
```

Residualized Plot



Coefficient Plots

- We only really care about one causal parameter at a time, usually.
- When we're in the causal inference mindset, we very rarely want to see long lists of covariates that are causally meaningless.
- In general, it is often worthwhile to make simple plots of your coefficients.
- I'll put some example code on the next page.
- It assumes a couple vectors exist:

Coefficient Plots

- We only really care about one causal parameter at a time, usually.
- When we're in the causal inference mindset, we very rarely want to see long lists of covariates that are causally meaningless.
- In general, it is often worthwhile to make simple plots of your coefficients.
- I'll put some example code on the next page.
- It assumes a couple vectors exist:

```
ests <- c(coef(lm(Yr2Obs~Trt))[2],coef(lm(Yr2Obs~Trt+Yr1Sco
ses <- c(summary(lm(Yr2Obs~Trt))$coefficients[2,2],summary(
var.names <- c("Unadjusted","Adjusted")
```

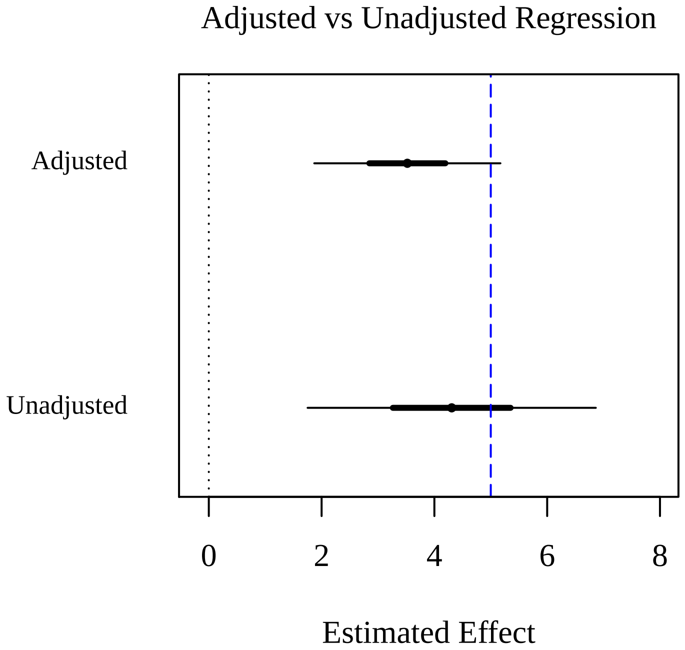
Coefficient Plot Code

```
par(
  family = "serif",
  oma = c(0,0,0,0),
  mar = c(5,10,4,2)
)

plot(NULL,
  xlim = c(-0.2, 8),
  ylim = c(.7, length(ests) + .3),
  axes = F, xlab = NA, ylab = NA)

for (i in 1:length(ests)) {
  points(ests[i], i, pch = 19, cex = .5)
  lines(c(ests[i] + 1.64*ses[i], ests[i] - 1.64*ses[i]), c(
  lines(c(ests[i] + .67*ses[i], ests[i] - .67*ses[i]), c(i,
  text(-1.1, i, var.names[i], xpd = T, cex = .8,pos=2)
}
```

Plotted



Adjusted vs Unadjusted Regression
Adjusted
Unadjusted
0
2
4
6
8
Estimated Effect

Last example

- This example posted in the egap Methods Guide based on an experiment by Gin'e and Mansuri on female voting behavior in Pakistan
- This is an information treatment: authors randomized info to women in Pakistan and study effect on turnout behavior, independence of their candidate choice and political knowledge
- They used a baseline survey to recover some covariates, we will simulate them: woman owns ID card, has formal schooling, age, and access to TV
- Potential outcomes include a measure of the extent to which a woman's choice of candidate was independent of the opinions of the men in her family

Last Example

- The potential outcomes are correlated with all four covariates, and the built-in “true” treatment effect on the independence measure here is 1.
- We will provide some evidence that the ATE is not biased if we don’t adjust, but that it is less precise

Simulation II

```
rm(list=ls())

set.seed(20140714)
N <- 2000
N.treated <- 1000
Replications <- 10000

true.treatment.effect <- 1

# Create pre-treatment covariates
owns.id.card <- rbinom(n = N, size = 1, prob = .18)
has.formal.schooling <- rbinom(n = N, size = 1, prob = .6)
age <- round(rnorm(n = N, mean = 37, sd = 16))
age[age<18] <- 18
age[age>65] <- 65
TV.access <- rbinom(n = N, size = 1, prob = .7)
epsilon <- rnorm(n = N, mean = 0, sd = 2)
```

Simulation II

```
# Create potential outcomes correlated
#with pre-treatment covariates
Y0 <- round(owns.id.card + 2*has.formal.schooling +
            3*TV.access + log(age) + epsilon)
Y1 <- Y0 + true.treatment.effect

# Assign treatment repeatedly
Z.mat <- replicate(Replications,
                   ifelse(1:N %in%
                          sample(1:N, N.treated), 1, 0))

# %in% is a logical vector which indicates whether a
#match was located for vector1 in vector2
```

Simulation II

```
# Generate observed outcomes
Y.mat <- Y1 * Z.mat + Y0 * (1 - Z.mat)

diff.in.means <- function(Y, Z) {
  coef(lm(Y ~ Z))[2]
}
ols.adjust <- function(Y, Z) {
  coef(lm(Y ~ Z + owns.id.card +
            has.formal.schooling + age + TV.access))[2]
}
unadjusted.estimates <- rep(NA, Replications)
adjusted.estimates   <- rep(NA, Replications)
```

Simulation II

```
for (i in 1:Replications) {
  unadjusted.estimates[i]  =
    diff.in.means(Y.mat[,i], Z.mat[,i])
  adjusted.estimates[i]    =
    ols.adjust(Y.mat[,i], Z.mat[,i])
}
# Estimated variability
sd.of.unadj <- sd(unadjusted.estimates)
sd.of.unadj
```

```
## [1] 0.121097
```

```
sd.of.adj   <- sd(adjusted.estimates)
sd.of.adj
```

```
## [1] 0.09288266
```

Simulation II

```r
# Estimated bias of each estimator
mean(unadjusted.estimates) - true.treatment.effect
```

```
## [1] 0.000753
```

```r
mean(adjusted.estimates) - true.treatment.effect
```

```
## [1] 0.0002864253
```

```r
# Confidence interval for the bias
1.96 * sd.of.unadj / sqrt(Replications)
```

```
## [1] 0.002373501
```

```r
1.96 * sd.of.adj   / sqrt(Replications)
```

```
## [1] 0.0018205
```