Instrumental Variables Simulation Homework

Kenny Mai

Objective

The goal of this exercise is to simulate data consistent with the assumptions of the IV estimaator we discussed in class (and described in the Angrist, Imbens, Rubin article posted on the Classes site). We will also evaluate the properties of different approaches to estimating the Complier Average Causal Effect.

Setting

To help conceptualize data that might be consistent with the IV assumptions, we will generate data from a hypothetical randomized encouragment design. In particular, imagine a study in which 1000 students entering an undergraduate degree program in the sciences in a major university were randomly assigned to one of two conditions. One group was encouraged via an email from the chair of their department to participate in a one week math boot camp just before the start of their first semester. Students in the other (not encouraged) group were also allowed to participate but received no special encouragement. In fact they would have had to discover on their own the existence of the program on the university website. The outcome variable is derived from the student test scores on the final exam for required math course for the sciences. In particular the Y variable that you will simulate below represents the *difference* between that score and the threshold for passing. Thus a negative value for a student reflects that the student did not pass.

Question 1. Simulate the data as god/goddess/supreme being of your choice.

In this section you will simulate data consistent with the assumptions. You will generate data for a sample of 1000 individuals.

(a) Simulate compliance status. Assume that 25% of individuals are compliers, 60% are never takers, and 15% are always takers. Generate D(0) and D(1) vectors to reflect this. You can also generate a vector indicating compliance type, C, if that is helpful to you.

```r
# Create C vector for compliance type
C <- c(rep("Complier",1000*0.25),rep("Never-taker",1000*0.6),rep("Always-taker",1000*0.15))
# Create D0's
D0 <- c(rep(0,1000*0.25),rep(0,1000*0.6),rep(1,1000*0.15))
# Create D1's
D1 <- c(rep(1,1000*0.25),rep(0,1000*0.6),rep(1,1000*0.15))
# cbind into data frame
compdat <- data.frame(cbind(C,D0,D1))
```

(b) Which compliance group has been omitted from consideration? What assumption does that imply?

Defiers, defined as D0 = 1 and D1 = 0 have been omitted from consideration, implying the assumption of monotonicity.

(c) Simulate the potential outcomes in a way that meets the following criteria:

(d) The exclusion restriction is satisfied.

(ii) The average effect of Z on Y for the compliers is 4.

(iii) The average Y(Z=0) for never takers is 0; The average Y(0) for compliers is 3; The average Y(Z=0) for always takers is 6.

(iv) The residual standard deviation is 1 for everyone in the sample (generated independently for each potential outcome).

```
# Initialize Y0
compdat$Y0 <- NA
# Adding noise with rnorm, generating values
compdat[compdat$C == "Never-taker",]$Y0 <- rnorm(1000*0.6,0,1)
compdat[compdat$C == "Complier",]$Y0 <- rnorm(1000*0.25,3,1)
compdat[compdat$C == "Always-taker",]$Y0 <- rnorm(1000*0.15,6,1)

# Initialize Y1
compdat$Y1 <- NA
# Adding noise with rnorm, generating values
compdat[compdat$C == "Never-taker",]$Y1 <- rnorm(1000*0.6,0,1)
compdat[compdat$C == "Complier",]$Y1 <- rnorm(1000*0.25,3+4,1)
compdat[compdat$C == "Always-taker",]$Y1 <- rnorm(1000*0.15,6,1)

# Quick sanity check
mean(compdat[compdat$C=="Never-taker",]$Y0)
```

```
## [1] -0.002812307
```

```
mean(compdat[compdat$C=="Complier",]$Y0)
```

```
## [1] 2.85566
```

```
mean(compdat[compdat$C=="Always-taker",]$Y0)
```

```
## [1] 5.950112
```

```
mean(compdat[compdat$C=="Never-taker",]$Y1)
```

```
## [1] 0.101127
```

```
mean(compdat[compdat$C=="Complier",]$Y1)
```

```
## [1] 7.015556
```

```
mean(compdat[compdat$C=="Always-taker",]$Y1)
```

```
## [1] 5.960076
```

(d) Calculate the SATE (average effect of Z on Y) for each of the compliance groups.

Exclusion restriction is assumed, so SATE for Never-takers and Always-takers are estimated to be close 0. Complier SATE is estimated in the following chunk:

```
# Difference in means conditioned on compliance type
compsate <- mean(compdat[compdat$C=="Complier",]$Y1) - mean(compdat[compdat$C=="Complier",]$Y0)
compsate
```

```
## [1] 4.159895
```

(e) What is another name for the SATE for the compliers?

CACE: Complier Average Causal Effect.

(f) Calculate the ITT using your simulated data.

```r
# Difference in means
mean(compdat$Y1 - compdat$Y0)
```

```
## [1] 1.103832
```

(g) Put D(0), D(1), Y(0), Y(1) into one dataset called dat.full. (You can also include a variable, C, indicating compliance group if you created one.)

```r
# Rename dataset
dat.full <- compdat
```

Question 2. Playing the role of the researcher to randomly assign treatments to observations.

Now switch to the role of the researcher. Pretend that you are running the experiment that we are examining for this assignment. Generate a binary indicator for the ignorable treatment *assignment* (as distinct from treatment receipt. . . . so this is Z, not D). Probability of receiving the treatment should be .5.

```r
# Use rbinom for treatment assignment
z = rbinom(1000,1,0.5)
# Quick check
table(z)
```

```
## z
##   0   1
## 503 497
```

Question 3. Back to playing god to understand which potential outcome manifests as an observed outcome.

Use dat.full to create a dataset that the researcher would actually get to see given the Z generated in Question 2. It should only have D, Z, and Y in it. Call it dat.obs.

```r
dat.res <- dat.full
# Assign treatment vector
dat.res$Z <- z
# Create single variables for D and Y conditioned on treatment variable
dat.res$D <- ifelse(dat.res$Z==1, dat.res$D1, dat.res$D0)
dat.res$Y <- ifelse(dat.res$Z==1, dat.res$Y1, dat.res$Y0)
# Build new data frame from old one
dat.res <- data.frame(D = dat.res$D, Z = dat.res$Z, Y = dat.res$Y)
```

Question 4. Estimate some quantities of interest as a researcher.

(a) *Estimate* the percent of compliers, never takers and always takers assuming that there are no defiers. Use only information in dat.obs.

```r
# Regression D~Z
summary(lm(D~Z, data=dat.res))
```

```
##
## Call:
## lm(formula = D ~ Z, data = dat.res)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.4044 -0.4044 -0.1491  0.5956  0.8509
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.14911    0.01912   7.798 1.58e-14 ***
## Z            0.25532    0.02712   9.414  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4288 on 998 degrees of freedom
## Multiple R-squared:  0.08155,    Adjusted R-squared:  0.08063
## F-statistic: 88.62 on 1 and 998 DF,  p-value: < 2.2e-16
```

```r
compliers <- summary(lm(D~Z, data=dat.res))$coefficients[2]
others <- 1-compliers
# Percentage of compliers
compliers
```

```
## [1] 0.2553212
```

```r
# Percentage of always takers and never takers combined
others
```

```
## [1] 0.7446788
```

(b) Estimate the naive regression estimate of the effect of the treatment on the outcome. Which estimand that we discussed in class is this equivalent to?

```r
# Regress Y~D
summary(lm(Y~D, data=dat.res))
```

```
##
## Call:
## lm(formula = Y ~ D, data = dat.res)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2129 -0.9334 -0.1959  0.7190  4.4088
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.50876    0.05054   10.07   <2e-16 ***
## D1           5.92359    0.09619   61.58   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.36 on 998 degrees of freedom
## Multiple R-squared:  0.7917, Adjusted R-squared:  0.7914
## F-statistic:  3792 on 1 and 998 DF,  p-value: < 2.2e-16
```

```r
naivetreat <- summary(lm(Y~D, data=dat.res))$coefficients[2]
naivetreat
```

```
## [1] 5.923595
```

The estimand this is equivalent to is the comparison between people who did receive treatment versus those who did not, in other words, the SATE.

(c) Estimate the intention-to-treat effect.

```r
# Regress Y~Z
summary(lm(Y~Z, data=dat.res))
```

```
##
## Call:
## lm(formula = Y ~ Z, data = dat.res)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.397 -2.399 -1.113  2.739  7.063
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.6010     0.1306  12.261  < 2e-16 ***
## Z             1.0919     0.1852   5.895 5.12e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.928 on 998 degrees of freedom
## Multiple R-squared:  0.03365,    Adjusted R-squared:  0.03268
## F-statistic: 34.75 on 1 and 998 DF,  p-value: 5.115e-09
```

```r
itt <- summary(lm(Y~Z, data=dat.res))$coefficients[2]
itt
```

```
## [1] 1.091891
```

(d) Estimate the CACE by dividing the ITT estimate by the percent of compliers in the sample.

```r
# Calculate CACE
cace <- itt/compliers
cace
```

```
## [1] 4.276539
```

(e) Estimate the CACE by performing two stage least squares on your own (that is without using an IV function in the R package AER).

```r
# Step 1, regress D~Z, save fitted values
s1 <- lm(D~Z, data=dat.res)
dhat <- fitted.values(s1)

# Step 2, regress Y ~ fitted values
s2 <- lm(dat.res$Y ~ dhat)
summary(s2)$coefficients[2]
```

```
## [1] 4.276539
```

(f) Provide an estimate of CACE and its standard error using the ivreg command in the AER package.

```r
# Load AER
library(AER)
```

```
## Loading required package: car

## Loading required package: carData

## Loading required package: lmtest

## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

## Loading required package: survival
```

```r
# Regress Y~D conditional on treatment
ivfitted <- ivreg(Y~D|Z, data=dat.res)
summary(ivfitted)$coefficients[2]
```

```
## [1] 4.276539
```

```r
summary(ivfitted)$coefficients[4]
```

```
## [1] 0.3831266
```

Question 5. Back to god mode: sampling distribution

Simulate a sampling distribution (with 10000 draws) for the estimator used in (4f). This will be simplified if you create a function from your simulation steps in Questions 1, 2, and 3. Is the estimator unbiased? Also report the standard deviation of the sampling distribution and compare to the standard error from your original dataset in (4f).

```r
# Doesn't look like I'll need to use this again, so no function
# Set simulation size
N <- 10000
# Initialize results vector
caceresult <- rep(NA, N)

# Begin loop
for(i in 1:N){
  z = rbinom(1000,1,0.5)
  dat.res <- dat.full
  dat.res$Z <- z
  dat.res$D <- ifelse(dat.res$Z==1, dat.res$D1, dat.res$D0)
  dat.res$Y <- ifelse(dat.res$Z==1, dat.res$Y1, dat.res$Y0)
  dat.res <- data.frame(D = dat.res$D, Z = dat.res$Z, Y = dat.res$Y)
  ivfitted <- ivreg(Y~D|Z, data=dat.res)
  caceresult[i] <- summary(ivfitted)$coefficients[2]
}

# Return values
mean(caceresult)
```

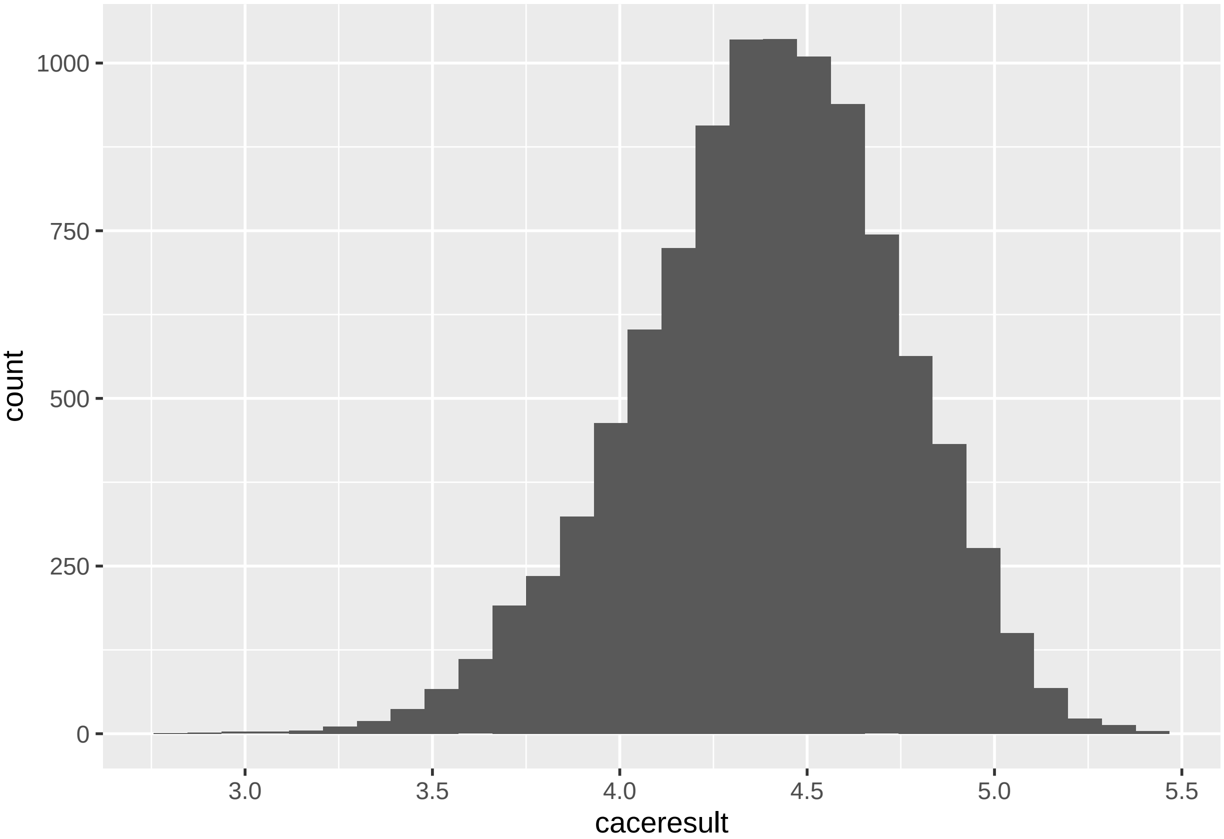
```
## [1] 4.389336
```

```r
sd(caceresult)
```

```
## [1] 0.3497342
```

```r
# Visual check for bias
library(ggplot2)
ggplot() +
  geom_histogram(data = as.data.frame(caceresult), aes(x = caceresult))
```

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



The estimate from the simulated data is close to 4, so we know that the CACE estimator used in 4f is unbiased. The simulated standard deviation is also very close to the standard error in 4f.

Question 6. Back to god mode: sampling distribution, assumptions

(a) Describe the assumptions required to obtain and unbiased estimate of the treatment effect, as described in AIR. We have generated data that satisfy these assumptions. Suppose instead you were handed data from the study described above. Comment on the plausibility of each of the required assumptions in that setting.

The required assumptions are:

Ignorability: the instrument variable must be randomly assigned. In other words, the encouragement email must be sent to random students. This is plausible, as it is easy and practical to implement.

Exclusion restriction: the always-takers and never takers test scores remain the same, regardless of whether or not they received the encouragement email. This is plausible.

Monotonicity: there are no students who would have gone to the camp if they hadn't received an email, and there are not students who wouldn't have gone had they received one. This is plausible, there doesn't seem to be a reason why students who were excited to go to the math camp.

Non-zero correlation: receiving an encouraging email has no effect on a student's decision to go to the boot camp. This is plausible, but there is the possiblity that students ignore any and all emails.

SUTVA: a student going to the boot camp has no effect on the test score of other students. Not plausible, the students go to the same school and interact with each other regarding academics. Spillover effects are possible since students aren't forced to isolate from each other.

(b) Suppose that the data generating process above included a covariate that predicted both Z and Y. Which of the assumptions described in (a) would that change and how?

Being able to predict both Z and Y violates the ignorability assumption on the instrument. Z should be assumed to be randomly assigned in order to be uncorrelated to any other observed covariates in each student.

(c) Suppose that the directions for Q1.c.iii was amended as follows " (iii) The average Y(0) for never takers is 0; The average Y(0) for compliers is 3; The average Y(0) for always takers is 6. The average Y(1) for never takers is 2." Which of the assumptions described in (a) would that violate?

Exclusion would be violated. The outcomes for students would would never go to the camp regardless of the encouragement would not remain the same. There would be a difference of 2 instead of 0.

(d) Redo one of the commands from Question 1 (just provide the code – you don't have to run it) such that the monotonicy assumption is violated.

```
C <- c(rep("Complier",1000*0.25),rep("Never-taker",1000*0.6),rep("Always-taker",1000*0.15))
D0 <- c(rep(0,1000*0.25),rep(0,1000*0.6),rep(0,1000*0.15)) # Changed last input in c()
D1 <- c(rep(1,1000*0.25),rep(1,1000*0.6),rep(1,1000*0.15)) # Changed 2nd input in c()
compdat <- data.frame(cbind(C,D0,D1))
```

(e) How could we alter the study design to preclude the existence of always takers? Would this be ethical?

A survey could be sent out asking students their attitudes towards attending a math boot camp if offerred. Students who respond stating they would definitely attend the boot camp could be labeled as always-takers and precluded. This would be ethically questionable, as it would seem unfair for students who are always-takers because they feel they struggle in math would be left out of the boot camp simply because they acknowledge that they might need help.