Instrumental Variables

Drew Dimmery drewd@nyu.edu

February 20, 2015

Debugging

- Rubber duck debugging:
  - Use when you can't figure out why your code doesn't work right
  - Find something inanimate to talk to
  - Explain what your code does, line by excruciating line
  - If you can't explain it, that's probably where the problem is.
  - This works ridiculously well.
  - You should also be able to tell your duck exactly what is stored in each variable at all times.
- Check individual elements of your code on small data such that you know what the right answer *should* be.

Introduce an Example

- We'll be working with data from a paper in the most recent issue of IO.
- Helfer, L.R. and E. Voeten. (2014) "International Courts as Agents of Legal Change: Evidence from LGBT Rights in Europe"
- The treatment we are interested in is the presence of absence of a ECtHR judgment.
- The outcome is the adoption of progressive LGBT policy.
- And there's a battery of controls, of course.
- Voeten has helpfully put all replication materials online.

Prepare example

```
require(foreign,quietly=TRUE)
d <- read.dta("replicationdataIOLGBT.dta")
```

```r
#Base specification
d$ecthrpos <- as.double(d$ecthrpos)-1
d.lm <- lm(policy~ecthrpos+pubsupport+ecthrcountry+lgbtlaws+cond+eumember0+euemploy+coemembe
d <- d[-d.lm$na.action,]
d$issue <- as.factor(d$issue)
d$ccode <- as.factor(d$ccode)
summary(d.lm)$coefficients[1:11,]
```

```
##                    Estimate   Std. Error    t value     Pr(>|t|)
## (Intercept)  -1.588605e+00 4.956355e-01 -3.2051890 1.360035e-03
## ecthrpos      6.500937e-02 1.056423e-02  6.1537237 8.289029e-10
## pubsupport    6.549488e-03 2.742967e-03  2.3877390 1.699714e-02
## ecthrcountry  1.297322e-01 3.583626e-02  3.6201389 2.979822e-04
## lgbtlaws      2.358238e-02 6.280655e-03  3.7547646 1.758966e-04
## cond          9.277344e-02 1.795954e-02  5.1656905 2.508722e-07
## eumember0    -8.586409e-03 8.497519e-03 -1.0104607 3.123339e-01
## euemploy      3.659200e-03 1.269275e-02  0.2882905 7.731389e-01
## coemembe      2.082823e-02 7.276808e-03  2.8622754 4.227313e-03
## lngdp        -7.522448e-07 4.501392e-07 -1.6711382 9.477027e-02
## year          8.019830e-04 2.522046e-04  3.1798904 1.484223e-03
```

Marginal Effects

- Blattman (2009) uses marginal effects "well" in the sense of causal inference.
- Use the builtin `predict` function; it will make your life easier.

. . .

```r
d.lm.interact <- lm(policy~ecthrpos*pubsupport+ecthrcountry+lgbtlaws+cond+eumember0+euemploy
frame0 <- frame1 <- model.frame(d.lm.interact)
frame0[,"ecthrpos"] <- 0
frame1[,"ecthrpos"] <- 1
meff <- mean(predict(d.lm.interact,newd=frame1) - predict(d.lm.interact,newd=frame0))
meff
```

```
## [1] 0.08197142
```

- Why might this be preferable to "setting things at their means/medians"?
- It's essentially integrating over the sample's distribution of observed characteristics.
- (And if the sample is a SRS from the population [or survey weights make it LOOK like it is], this will then get you the marginal effect on the population of interest)

Delta Method

- Note 1: We know that our vector of coefficients are asymptotically multivariate normal.
- Note 2: We can approximate the distribution of many (not just linear) functions of these coefficients using the delta method.
- Delta method says that you can approximate the distribution of $h(b_n)$ with $\bigtriangledown h(b)'\Omega \bigtriangledown h(b)$ Where Ω is the asymptotic variance of b.
- In practice, this means that we just need to be able to derive the function whose distribution we wish to approximate.

Trivial Example

- Maybe we're interested in the ratio of the coefficient on `ecthrpos` to that of `pubsupport`.
- Call it $\frac{b_2}{b_3}$. The gradient is $(\frac{1}{b_3}, \frac{b_2}{b_3^2})$
- Estimate this easily in R with:

. . .

```
grad<-c(1/coef(d.lm)[3],coef(d.lm)[2]/coef(d.lm)[3]^2)
grad
```

```
## pubsupport   ecthrpos
##   334.0251  8046.4669
```

```
se<-sqrt(t(grad)%*%vcov(d.lm)[2:3,2:3]%*%grad)
est<-coef(d.lm)[2]/coef(d.lm)[3]
c(estimate=est,std.error=se)
```

```
## estimate.ecthrpos         std.error
##          24.08941          35.32946
```

```
require(car,quietly=TRUE)
deltaMethod(d.lm,"ecthrpos/pubsupport")
```

```
##                     Estimate       SE
## ecthrpos/pubsupport 24.08941 35.54775
```

Linear Functions

- But for most "marginal effects", you don't need to use the delta method.
- Just remember your rules for variances.
- $\text{var}(aX + bY) = a^2\text{var}(X) + b^2\text{var}(Y) + 2ab\text{cov}(X, Y)$
- If you are just looking at changes with respect to a single variable, you can just multiply standard errors.
- That is, a change in a variable of 3 units means that the standard error for the marginal effect would be 3 times the standard error of the coefficient.
- This isn't what Clarify does, though.

Instrumental Variables

- $\rho = \frac{\text{Cov}(Y_i,Z_i)}{\text{Cov}(S_i,Z_i)} = \frac{\frac{\text{Cov}(Y_i,Z_i)}{\text{Var}(Z_i)}}{\frac{\text{Cov}(S_i,Z_i)}{\text{Var}(Z_i)}} = \frac{\text{Reduced form}}{\text{First stage}}$
- If we have a perfect instrument, this will be unbiased.
- But bias is a function of both violation of exclusion restriction and of strength of first stage.
- 2SLS has finite sample bias. (Cyrus showed this, but didn't dwell on it)
- In particular, it can be shown that this bias "is":
  $\frac{\sigma_{\eta\xi}}{\sigma_\xi^2}\frac{1}{F+1}$
  where η is the error in the structural model and ξ is the error in the first stage.
- With an irrelevant instrument ($F = 0$), the bias is equal to that of OLS (regression of Y on X).
- There are some bias corrections for this, we might talk about this next week.

Setup IV example

- For our example with IV, we will start with AJR (2001) - Colonial Origins of Comparative Development
- Treatment is average protection from expropriation
- Exogenous covariates are dummies for British/French colonial presence
- Instrument is settler mortality
- Outcome is log(GDP) in 1995

. . .

```
require(foreign,quietly=TRUE)
dat <- read.dta("maketable5.dta")
dat <- subset(dat, baseco==1)
```

Estimate IV via 2SLS

```r
require(AER,quietly=TRUE)
first <- lm(avexpr~logem4+f_brit+f_french,dat)
iv2sls<-ivreg(logpgp95~avexpr+f_brit+f_french,~logem4+f_brit+f_french,dat)
require(car)
linearHypothesis(first,"logem4",test="F")
```

```
## Linear hypothesis test
##
## Hypothesis:
## logem4 = 0
##
## Model 1: restricted model
## Model 2: avexpr ~ logem4 + f_brit + f_french
##
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1     61 116.983
## 2     60  94.013  1    22.969 14.659 0.0003101 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Examine First Stage

```r
summary(first)
```

```
##
## Call:
## lm(formula = avexpr ~ logem4 + f_brit + f_french, data = dat)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.98210 -0.86954  0.05616  0.86237  2.79411
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   8.7467     0.6904  12.669  < 2e-16 ***
## logem4       -0.5344     0.1396  -3.829  0.00031 ***
## f_brit        0.6293     0.3665   1.717  0.09109 .
## f_french      0.0474     0.4295   0.110  0.91249
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 1.252 on 60 degrees of freedom
## Multiple R-squared:  0.3081, Adjusted R-squared:  0.2736
## F-statistic: 8.908 on 3 and 60 DF,  p-value: 5.704e-05
```

Examine Output

```r
summary(iv2sls)
```

```
##
## Call:
## ivreg(formula = logpgp95 ~ avexpr + f_brit + f_french | logem4 +
##     f_brit + f_french, data = dat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2716 -0.7488  0.0728  0.7544  2.4004
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.3724     1.3880   0.989    0.327
## avexpr        1.0779     0.2176   4.953 6.28e-06 ***
## f_brit       -0.7777     0.3543  -2.195    0.032 *
## f_french     -0.1170     0.3548  -0.330    0.743
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.043 on 60 degrees of freedom
## Multiple R-Squared: 0.04833, Adjusted R-squared: 0.0007476
## Wald test: 10.07 on 3 and 60 DF,  p-value: 1.822e-05
```

Sensitivity Analysis

- Conley, Hansen and Rossi (2012)
- Suppose that the exclusion restriction does NOT hold, and there exists a direct effect from the instrument to the outcome.
- That is, the structural model is:
  $Y = X\beta + Z\gamma + \epsilon$
- If γ is zero, the exclusion restriction holds (we're in a structural framework)
- We can assume a particular value of γ, take $\tilde{Y} = Y - Z\gamma$ and estimate our model, gaining an estimate of β.
- This defines a sensitivity analysis on the exclusion restriction.

- Subject to an assumption about the support of γ, they suggest estimating in a grid over this domain, and then taking the union of the confidence intervals for each value of γ as the combined confidence interval (which will cover).

Sensitivity Analysis code

```
gamma <- seq(-1,1,.25)
ExclSens <- function(g) {
  newY <- dat$logpgp95 - g*dat$logem4
  coef(ivreg(newY~avexpr+f_brit+f_french,~logem4+f_brit+f_french,cbind(dat,newY)))[2]
}
sens.coefs <- sapply(gamma,ExclSens)
names(sens.coefs)<- round(gamma,3)
round(sens.coefs,3)
```

```
##     -1  -0.75   -0.5  -0.25      0   0.25    0.5   0.75      1
## -0.793 -0.326  0.142  0.610  1.078  1.546  2.013  2.481  2.949
```

More IV Stuff

- We're going to be looking at Ananat (2011) in AEJ
- This study looks at the effect of racial segregation on economic outcomes.
- Outcome: Poverty rate & Inequality (Gini index)
- Treatment: Segregation
- Instrument: "railroad division index"
- Main covariate of note: railroad length in a town
- I'm dichotomizing treatment and instrument for simplicity.
- And my outcomes are for the Black subsample

. . .

```
require(foreign)
d<-read.dta("aej_maindata.dta")
d$herf_b<-with(d,ifelse(herf >= quantile(herf,.5),1,0))
d$dism1990_b<-with(d,ifelse(dism1990 >= quantile(dism1990,.5),1,0))
first.stage <- lm(dism1990~herf+lenper,d)
first.stage.b <- lm(dism1990_b~herf_b+lenper,d)
require(AER)
gini.iv <- ivreg(lngini_b~dism1990+lenper,~herf+lenper,d)
gini.iv.b <- ivreg(lngini_b~dism1990_b+lenper,~herf_b+lenper,d)
pov.iv <- ivreg(povrate_b~dism1990+lenper,~herf+lenper,d)
pov.iv.b <- ivreg(povrate_b~dism1990_b+lenper,~herf_b+lenper,d)
```

Base Results

```r
round(summary(first.stage)$coefficients[2,],3)
```

```
##   Estimate Std. Error    t value   Pr(>|t|)
##      0.357      0.081      4.395      0.000
```

```r
round(summary(first.stage.b)$coefficients[2,],3)
```

```
##   Estimate Std. Error    t value   Pr(>|t|)
##      0.372      0.083      4.481      0.000
```

```r
round(summary(gini.iv)$coefficients[2,],3)
```

```
##   Estimate Std. Error    t value   Pr(>|t|)
##      0.875      0.302      2.895      0.005
```

```r
round(summary(gini.iv.b)$coefficients[2,],3)
```

```
##   Estimate Std. Error    t value   Pr(>|t|)
##      0.211      0.081      2.615      0.010
```

```r
round(summary(pov.iv)$coefficients[2,],3)
```

```
##   Estimate Std. Error    t value   Pr(>|t|)
##      0.258      0.144      1.798      0.075
```

```r
round(summary(pov.iv.b)$coefficients[2,],3)
```

```
##   Estimate Std. Error    t value   Pr(>|t|)
##      0.059      0.039      1.543      0.125
```

Abadie's κ

- Recall from the lecture that we can use a weighting scheme to calculate statistics on the compliant population.
- $E[g(Y,D,X)|D_1 > D_0] = \frac{1}{p(D_1>D_0)} E[\kappa g(Y,D,X)]$
- $\kappa = 1 - \frac{D_i(1-Z_i)}{p(Z_i=0|X)} - \frac{(1-D_i)Z_i}{p(Z_i=1|X)}$
- $E[\kappa|X] = E[D_1 - D_0|X] = E[D|X, Z=1] - E[D|X, Z=0]$
- Take $w_i = \frac{\kappa_i}{E[D_1 - D_0|X_i]}$

- Use this in calculating any interesting statistics (means, variance, etc)
- This let's you explore the units composing your LATE.

. . .

```
getKappaWt<-function(D,Z) {
  pz <- mean(Z)
  pcomp <- mean(D[Z==1]) - mean(D[Z==0])
  if(pcomp < 0) stop("Assuming p(D|Z) > .5")
  kappa <- 1 - D*(1-Z)/(1-pz) - (1-D)*Z/pz
  # Note that pcomp = mean(kappa)
  kappa / pcomp
}
w <- with(d,getKappaWt(D=dism1990_b,Z=herf_b))
varlist <- c("closeness","area1910","ctyliterate1920","hsdrop_b","manshr","ctymanuf_wkrs192(
samp.stats<-sapply(varlist,function(v) mean(d[,v],na.rm=TRUE))
comp.stats<-sapply(varlist,function(v) weighted.mean(d[,v],w,na.rm=TRUE))
```

Examine Complier Statistics

```
summary(w)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  -2.511  -2.429   2.470   1.000   2.470   2.470
```

```
rbind(sample=samp.stats,compliers=comp.stats)
```

```
##           closeness area1910 ctyliterate1920  hsdrop_b    manshr
## sample    -362.4348 14626.43       0.9585012 0.2516300 0.1891766
## compliers -299.1428 18012.56       0.9514523 0.2423754 0.2109807
##           ctymanuf_wkrs1920   ngov62
## sample            0.4618666 55.55072
## compliers         0.4266065 83.65072
```