simulate_IV_-_Mar28.R

danny

2020-03-24

```
# Author: Gordon Burtch and Gautam Ray
# Course: MSBA 6440
# Session: Instrumental Variables
# Topic: Simulating Instruments
# Lecture 7

library(MASS)
library(stargazer)
```

```
##
## Please cite as:

##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.

##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```
library(AER)
```

```
## Warning: package 'AER' was built under R version 3.6.3

## Loading required package: car

## Warning: package 'car' was built under R version 3.6.3

## Loading required package: carData

## Loading required package: lmtest

## Warning: package 'lmtest' was built under R version 3.6.2

## Loading required package: zoo

## Warning: package 'zoo' was built under R version 3.6.2

##
## Attaching package: 'zoo'

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

## Loading required package: sandwich
```

```
## Warning: package 'sandwich' was built under R version 3.6.2
```

```
## Loading required package: survival
```

```
## Warning: package 'survival' was built under R version 3.6.3
```

```
library(lfe)
```

```
## Warning: package 'lfe' was built under R version 3.6.3
```

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

```
##
## Attaching package: 'lfe'
```

```
## The following object is masked from 'package:lmtest':
##
##     waldtest
```

```
# We will first simulate our treatment variable x, endogenous portion of x and its confounder, c
# Here, we refer to the endogenous variation in x as x*
# An easy way to make them confounded is to use the multivariate normal draw function, mvnorm.
xStarAndC <- mvrnorm(1000, c(20, 15), matrix(c(1, 0.5, 0.5, 1), 2, 2))
xStar <- xStarAndC[, 1]
c <- xStarAndC[, 2]

# If you are curious about syntax for mvnorm... ??MASS::mvnorm
# We pass it the number of obs to draw, the means of the two variables, and a covariance matrix.
# In this case, we simulated 1000 draws for two variables, mean 20 and 15, which are 50% correlated.
cov(xStar,c)
```

```
## [1] 0.5269345
```

```
# Now let's simulate our instrument, and make observed X a function of good variation (random stuff)
# and the bad variation, x*.
# By construction, z is a valid instrument for X now, because it is only correlated with the
# good variation, and it has no direct relationship on our eventual y (except through x).
z <- rnorm(1000)
x <- xStar + z

# Now let's simulate the data-generating process to recover y,
# a function of observed x, its confounder and an error term.
# Here, the true marginal effect of x on y is 2.
y <- 1 + 2*x + 10*c + rnorm(1000, 0, 0.5)

# Now lets check to make sure we have a problem of confounding.
cor(x, c)
```

```
## [1] 0.3881926
```

```r
cor(y, c)
```

```
## [1] 0.9731409
```

```r
# And let's check that the instrument is valid...
cor(x,z)
```

```
## [1] 0.6977486
```

```r
cor(c,z)
```

```
## [1] 0.01538544
```

```r
# Okay, let's run the 'true' regression first, controlling for the confounder.
ols_true <- lm(y ~ x + c)
# ... and let's run the endogenous regression, ignoring the confounder.
ols_endog <- lm(y ~ x)
stargazer(ols_true,ols_endog,type="text",title="True vs. Endogenous Regression",column.labels = c("True
```

```
##
## True vs. Endogenous Regression
## ================================================================================
##                                        Dependent variable:
##                          ---------------------------------------------------
##                                                   y
##                                    True                       Endogenous
##                                    (1)                           (2)
## --------------------------------------------------------------------------------
## x                                1.995***                      4.770***
##                                  (0.013)                       (0.209)
##
## c                                9.989***
##                                  (0.018)
##
## Constant                         1.297***                      95.744***
##                                  (0.286)                       (4.183)
##
## --------------------------------------------------------------------------------
## Observations                      1,000                         1,000
## R2                                0.998                         0.343
## Adjusted R2                       0.998                         0.343
## Residual Std. Error         0.513 (df = 997)              9.222 (df = 998)
## F Statistic           245,276.000*** (df = 2; 997) 521.464*** (df = 1; 998)
## ================================================================================
## Note:                                              *p<0.1; **p<0.05; ***p<0.01
```

```r
# Okay, so let's start working toward IV reg. Let's do the first stage regression and use its predictio
xHat <- lm(x ~ z)$fitted.values
ols_corrected <- lm(y ~ xHat)
stargazer(ols_true,ols_endog,ols_corrected,type="text",title="True vs. Endogenous vs. Instrumented",colu
```

```
##
## True vs. Endogenous vs. Instrumented
## ================================================================================================
##                                              Dependent variable:
##                      ------------------------------------------------------------------------
##                                                      y
##                                 True                Endogenous              Manual
##                                 (1)                   (2)                    (3)
## ------------------------------------------------------------------------------------------------
## x                             1.995***              4.770***
##                               (0.013)               (0.209)
##
## c                             9.989***
##                               (0.018)
##
## xHat                                                                       2.149***
##                                                                            (0.363)
##
## Constant                      1.297***              95.744***             148.102***
##                               (0.286)               (4.183)                (7.262)
##
## ------------------------------------------------------------------------------------------------
## Observations                   1,000                 1,000                  1,000
## R2                             0.998                 0.343                  0.034
## Adjusted R2                    0.998                 0.343                  0.033
## Residual Std. Error       0.513 (df = 997)      9.222 (df = 998)       11.185 (df = 998)
## F Statistic         245,276.000*** (df = 2; 997) 521.464*** (df = 1; 998) 35.043*** (df = 1; 998)
## ================================================================================================
## Note:                                                              *p<0.1; **p<0.05; ***p<0.01
```

```r
# Note that the beta is correctly estimated but the standard errors are not if we use this approach.
# The ivreg package will calculate not only this beta, but the right standard errors.
ivreg <- ivreg(formula=y ~ x | z)
stargazer(ols_true,ols_endog,ols_corrected,ivreg,type="text",title="True vs. Endogenous vs. Manual vs. I
```

```
##
## True vs. Endogenous vs. Manual vs. Instrumented
## ================================================================================================
##                                              Dependent variable:
##                      ------------------------------------------------------------------------
##                                                      y
##                                                    OLS
##
##                                 True                Endogenous              Manual
##                                 (1)                   (2)                    (3)
## ------------------------------------------------------------------------------------------------
## x                             1.995***              4.770***
##                               (0.013)               (0.209)
##
## c                             9.989***
##                               (0.018)
##
## xHat                                                                       2.149***
##                                                                            (0.363)
```

```
##
## Constant                     1.297***                 95.744***              148.102***
##                              (0.286)                   (4.183)                (7.262)
##
## ---------------------------------------------------------------------------------------------------
## Observations                  1,000                     1,000                  1,000
## R2                            0.998                     0.343                  0.034
## Adjusted R2                   0.998                     0.343                  0.033
## Residual Std. Error       0.513 (df = 997)          9.222 (df = 998)       11.185 (df = 998)    9.9
## F Statistic         245,276.000*** (df = 2; 997) 521.464*** (df = 1; 998) 35.043*** (df = 1; 998)
## ===================================================================================================
## Note:                                                                          *p<0.1; **p<0
```

```
summary(ivreg,diagnostics=TRUE)
```

```
##
## Call:
## ivreg(formula = y ~ x | z)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -28.77064  -6.14892  -0.01766   6.18700  32.24975
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 148.1020     6.4430  22.987  < 2e-16 ***
## x             2.1494     0.3221   6.672 4.16e-11 ***
##
## Diagnostic tests:
##                  df1 df2 statistic p-value
## Weak instruments   1 998     946.9  <2e-16 ***
## Wu-Hausman         1 997     175.4  <2e-16 ***
## Sargan             0  NA        NA      NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.923 on 998 degrees of freedom
## Multiple R-Squared: 0.2396,  Adjusted R-squared: 0.2388
## Wald test: 44.52 on 1 and 998 DF,  p-value: 4.163e-11
```

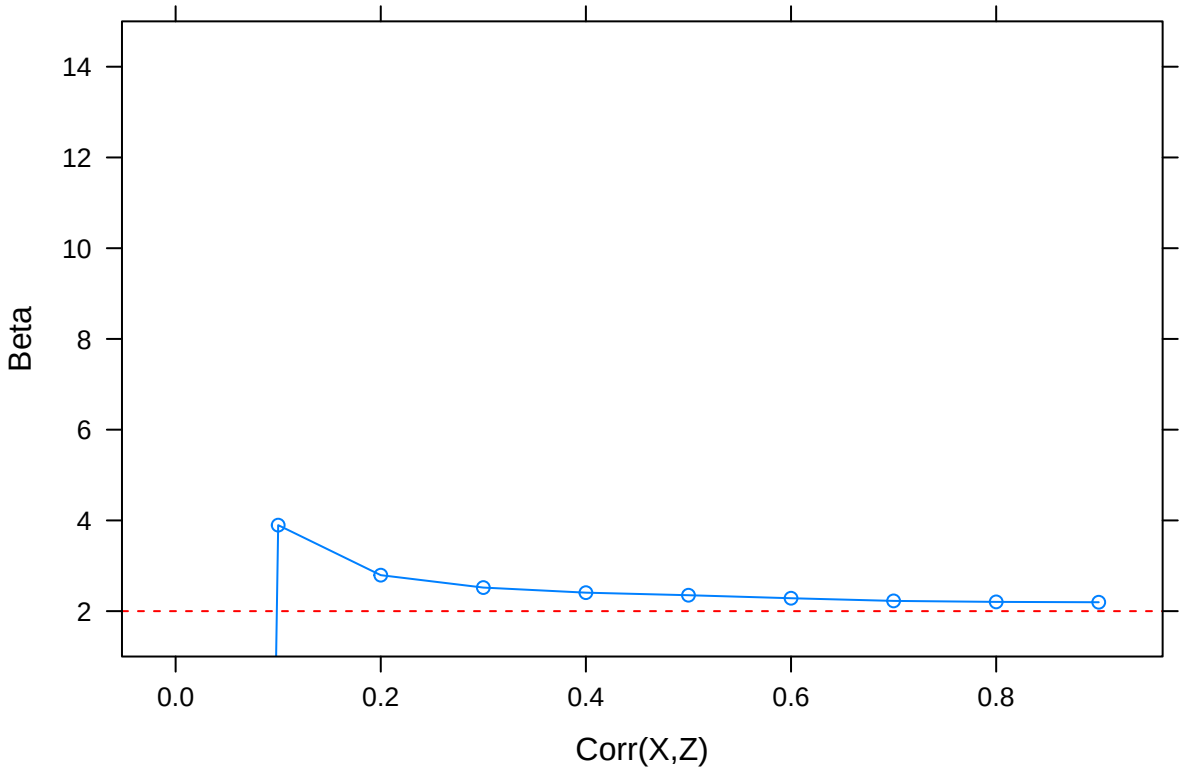
```
# If you have multiple endogenous and instrumental variables in a single regression, you can tell R exp
# This syntax means "remove" x and include z.
# Any variables not mentioned after the pipe instrument for themselves (perfect predictors).
#ivreg <- ivreg(formula=y ~ x | .-x + z)

# Okay, now let's see what happens if we use instruments that are too weak in their association with x.
x1 <- xStar + 0.9*z
x2 <- xStar + 0.8*z
x3 <- xStar + 0.7*z
x4 <- xStar + 0.6*z
x5 <- xStar + 0.5*z
x6 <- xStar + 0.4*z
x7 <- xStar + 0.3*z
```

```
x8 <- xStar + 0.2*z
x9 <- xStar + 0.1*z
x10 <- xStar + 0.01*z

# Now let's simulate the data-generating process to recover y,
# a function of observed x, its confounder and an error term.
# Here, the true marginal effect of x on y is 2.
y1 <- 1 + 2*x1 + 10*c + rnorm(1000, 0, 0.5)
y2 <- 1 + 2*x2 + 10*c + rnorm(1000, 0, 0.5)
y3 <- 1 + 2*x3 + 10*c + rnorm(1000, 0, 0.5)
y4 <- 1 + 2*x4 + 10*c + rnorm(1000, 0, 0.5)
y5 <- 1 + 2*x5 + 10*c + rnorm(1000, 0, 0.5)
y6 <- 1 + 2*x6 + 10*c + rnorm(1000, 0, 0.5)
y7 <- 1 + 2*x7 + 10*c + rnorm(1000, 0, 0.5)
y8 <- 1 + 2*x8 + 10*c + rnorm(1000, 0, 0.5)
y9 <- 1 + 2*x9 + 10*c + rnorm(1000, 0, 0.5)
y10 <- 1 + 2*x10 + 10*c + rnorm(1000, 0, 0.5)

ivreg_weak1 <- ivreg(formula=y1 ~ x1 | z)
ivreg_weak2 <- ivreg(formula=y2 ~ x2 | z)
ivreg_weak3 <- ivreg(formula=y3 ~ x3 | z)
ivreg_weak4 <- ivreg(formula=y4 ~ x4 | z)
ivreg_weak5 <- ivreg(formula=y5 ~ x5 | z)
ivreg_weak6 <- ivreg(formula=y6 ~ x6 | z)
ivreg_weak7 <- ivreg(formula=y7 ~ x7 | z)
ivreg_weak8 <- ivreg(formula=y8 ~ x8 | z)
ivreg_weak9 <- ivreg(formula=y9 ~ x9 | z)
ivreg_weak10 <- ivreg(formula=y10 ~ x10 | z)

# The weaker our instrument, the less accurate our final estimate of X's effect becomes.
stargazer(ivreg_weak1,ivreg_weak2,ivreg_weak3,ivreg_weak4,ivreg_weak5,ivreg_weak6,ivreg_weak7,ivreg_weak
```

```
##
## ==========================================================================================================
##                                                                               Dependent variable:
##                                  -------------------------------------------------------------------------
##                                      y1          y2          y3          y4          y5          y6
##                                     (1)         (2)         (3)         (4)         (5)         (6)
## ----------------------------------------------------------------------------------------------------------
## x1                                2.196***
##                                   (0.358)
##
## x2                                            2.204***
##                                               (0.403)
##
## x3                                                        2.226***
##                                                           (0.460)
##
## x4                                                                    2.285***
##                                                                       (0.539)
##
## x5                                                                                2.351***
##                                                                                   (0.645)
```

```
##
## x6                                                                                                  2.407***
##                                                                                                     (0.810)
##
## x7                                                                                                                2.5
##                                                                                                                   (1
##
## x8
##
##
## x9
##
##
## x10
##
##
## Constant                          147.108*** 146.993*** 146.565*** 145.332*** 144.004*** 142.891*** 140
##                                    (7.157)    (8.061)    (9.201)    (10.785)   (12.891)   (16.205)   (2
##
## ---------------------------------------------------------------------------------------------------------
## Observations                         1,000      1,000      1,000      1,000      1,000      1,000      1
## R2                                   0.239      0.235      0.233      0.234      0.237      0.238      0
## Adjusted R2                          0.239      0.234      0.233      0.233      0.236      0.237      0
## Residual Std. Error (df = 998)       9.909      9.906      9.873      9.894      9.817      9.815      9
## =========================================================================================================
## Note:
```

```r
# Let's plot it for interests sake...
# First we pull out all the beta estimates, and we make a vector of the correlations we used (strength
betas <- rep(NA,10)
for (i in 1:10){
  betas[i] <- get(paste0('ivreg_weak',i))$coefficients[2]
}
weakness <- c(.9,.8,.7,.6,.5,.4,.3,.2,.1,.01)

# Now let's plot our recovered betas, against their strength, and include a ref line for true value of
library(lattice)
p <- xyplot(betas~weakness,xlab="Corr(X,Z)",ylab="Beta",ylim=c(1,15),type="b")
update(p, panel=function(...){
  panel.xyplot(...)
  panel.abline(h=2,lty=2,col="red")
} )
```

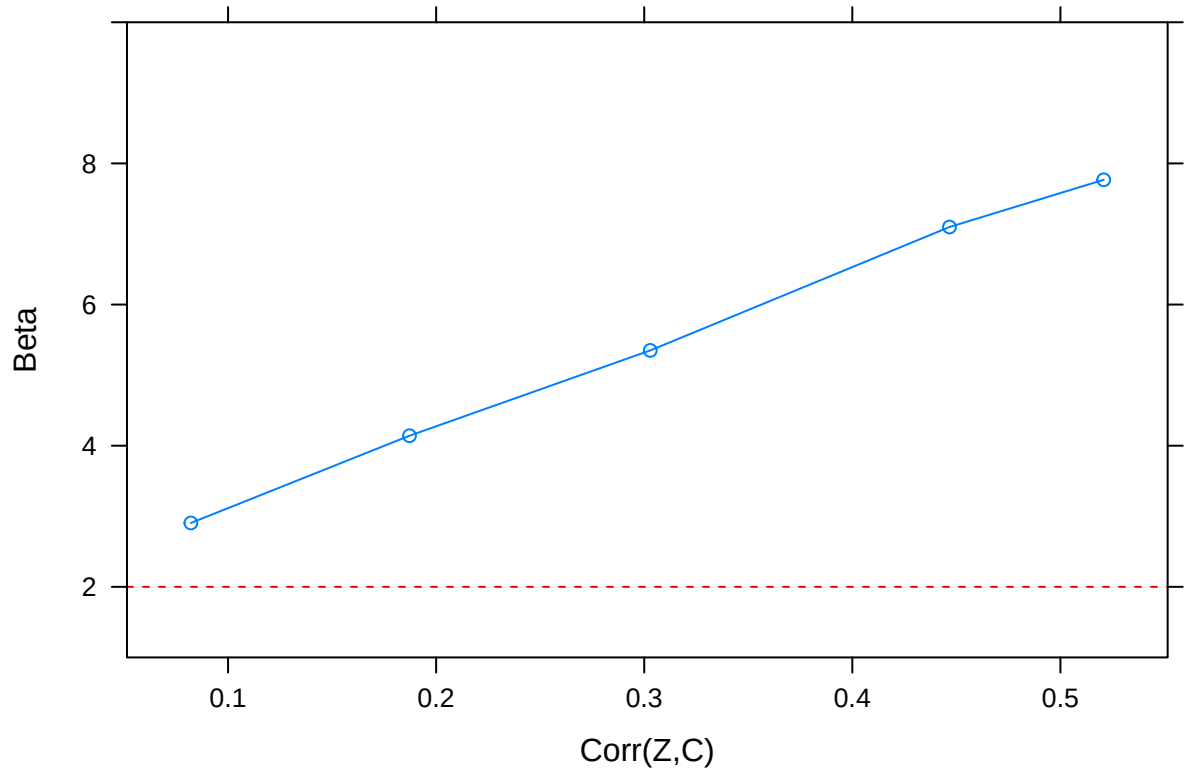



```
# Okay, now let's see what happens as we violate exclusion
# That is, as we allow z to be correlated to an increasing degree with the confounders in the error ter
# To make this work, we now need to draw all three variables from a joint distribution (good x, z and c
x_C_Z_1 <- mvrnorm(1000, c(20, 15, 10), matrix(c(1,0.5,0.9,0.5,1,.1,.9,.1,1), 3, 3))
x_C_Z_2 <- mvrnorm(1000, c(20, 15, 10), matrix(c(1,0.5,0.9,0.5,1,.2,.9,.2,1), 3, 3))
x_C_Z_3 <- mvrnorm(1000, c(20, 15, 10), matrix(c(1,0.5,0.9,0.5,1,.3,.9,.3,1), 3, 3))
x_C_Z_4 <- mvrnorm(1000, c(20, 15, 10), matrix(c(1,0.5,0.9,0.5,1,.4,.9,.4,1), 3, 3))
x_C_Z_5 <- mvrnorm(1000, c(20, 15, 10), matrix(c(1,0.5,0.9,0.5,1,.5,.9,.5,1), 3, 3))

x11 <- x_C_Z_1[, 1]
x12 <- x_C_Z_2[, 1]
x13 <- x_C_Z_3[, 1]
x14 <- x_C_Z_4[, 1]
x15 <- x_C_Z_5[, 1]
c1 <- x_C_Z_1[, 2]
c2 <- x_C_Z_2[, 2]
c3 <- x_C_Z_3[, 2]
c4 <- x_C_Z_4[, 2]
c5 <- x_C_Z_5[, 2]
z1 <- x_C_Z_1[, 3]
z2 <- x_C_Z_2[, 3]
z3 <- x_C_Z_3[, 3]
z4 <- x_C_Z_4[, 3]
z5 <- x_C_Z_5[, 3]

# What are we doing here? Making versions of z that are increasingly correlated with c.
```

```r
# Let's store those correlations for our plot later.
exclusion <- rep(NA,5)
for (i in 1:5){
  exclusion[i] <- cor(get(paste0("c",i)),get(paste0("z",i)))
}

# Okay, now let's simulate our Y's
y11 <- 1 + 2*x11 + 10*c1 + rnorm(1000, 0, 0.5)
y12 <- 1 + 2*x12 + 10*c2 + rnorm(1000, 0, 0.5)
y13 <- 1 + 2*x13 + 10*c3 + rnorm(1000, 0, 0.5)
y14 <- 1 + 2*x14 + 10*c4 + rnorm(1000, 0, 0.5)
y15 <- 1 + 2*x15 + 10*c5 + rnorm(1000, 0, 0.5)

ivreg_endog1 <- ivreg(formula=y11 ~ x11 | z1)
ivreg_endog2 <- ivreg(formula=y12 ~ x12 | z2)
ivreg_endog3 <- ivreg(formula=y13 ~ x13 | z3)
ivreg_endog4 <- ivreg(formula=y14 ~ x14 | z4)
ivreg_endog5 <- ivreg(formula=y15 ~ x15 | z5)

stargazer(ivreg_endog1,ivreg_endog2,ivreg_endog3,ivreg_endog4,ivreg_endog5,type="text")
```

```
##
## ================================================================================
##                                          Dependent variable:
##                                 ---------------------------------------------
##                                    y11        y12        y13        y14        y15
##                                    (1)        (2)        (3)        (4)        (5)
## --------------------------------------------------------------------------------
## x11                              2.905***
##                                  (0.341)
##
## x12                                        4.141***
##                                            (0.328)
##
## x13                                                   5.350***
##                                                       (0.310)
##
## x14                                                              7.099***
##                                                                  (0.302)
##
## x15                                                                         7.768***
##                                                                             (0.304)
##
## Constant                        132.862*** 108.121*** 83.880*** 49.297*** 36.087***
##                                  (6.839)    (6.563)    (6.215)   (6.038)   (6.089)
##
## --------------------------------------------------------------------------------
## Observations                       1,000      1,000      1,000     1,000     1,000
## R2                                 0.250      0.319      0.369     0.444     0.398
## Adjusted R2                        0.249      0.318      0.369     0.443     0.397
## Residual Std. Error (df = 998)     9.623      9.150      8.796     8.484     8.701
## ================================================================================
## Note:                                                  *p<0.1; **p<0.05; ***p<0.01
```

```r
# Let's plot it again...
# As you can see, as Z becomes less "excluded" we see it yields worse and worse estimates of X's margin
# Effect on Y.
betas <- rep(NA,5)
for (i in 1:5){
  betas[i] <- get(paste0('ivreg_endog',i))$coefficients[2]
}
p <- xyplot(betas~exclusion,xlab="Corr(Z,C)",ylab="Beta",ylim=c(1,10),type="b")
update(p, panel=function(...){
  panel.xyplot(...)
  panel.abline(h=2,lty=2,col="red")
} )
```




```r
# Okay let's do a real example here...
# This dataset is state-level data on cigarette sales, prices
# and taxes. Taxes are used as an instrument for prices here.
data("CigarettesSW")
sales <- lm(log(packs) ~ log(price) + year + state, data=CigarettesSW)
sales_iv <- ivreg(log(packs) ~ log(price) + year + state | .-log(price) + tax, data = CigarettesSW)
stargazer(sales,sales_iv,title="OLS vs. IV",type="text",column.labels = c("OLS","IV"),omit=c("state","y
```

```
##
## OLS vs. IV
## ========================================================================
##                                        Dependent variable:
```

```
##                                 -----------------------------------
##                                             log(packs)
##                                       OLS           instrumental
##                                                       variable
##                                       OLS               IV
##                                       (1)               (2)
## ------------------------------------------------------------------
## log(price)                         -1.085***         -1.380***
##                                     (0.151)           (0.192)
##
## Constant                           9.763***          11.108***
##                                     (0.689)           (0.879)
##
## ------------------------------------------------------------------
## Observations                          96                96
## R2                                   0.966             0.963
## Adjusted R2                          0.929             0.923
## Residual Std. Error (df = 46)        0.065             0.068
## F Statistic                 26.306*** (df = 49; 46)
## ==================================================================
## Note:                                *p<0.1; **p<0.05; ***p<0.01
```

```r
# Setting "diagnostics = TRUE" let's us assess a hausman test, weak IV stats and overidentifying tests
summary(sales_iv,diagnostics=TRUE)
```

```
##
## Call:
## ivreg(formula = log(packs) ~ log(price) + year + state | . -
##     log(price) + tax, data = CigarettesSW)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -1.068e-01 -3.755e-02 -1.776e-15  3.755e-02  1.068e-01
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.107617   0.878778  12.640  < 2e-16 ***
## log(price)  -1.379516   0.192287  -7.174 5.00e-09 ***
## year1995     0.528352   0.109568   4.822 1.59e-05 ***
## stateAR      0.162404   0.068269   2.379 0.021573 *
## stateAZ     -0.026116   0.073058  -0.357 0.722376
## stateCA     -0.128743   0.075062  -1.715 0.093046 .
## stateCO     -0.134011   0.067687  -1.980 0.053721 .
## stateCT      0.224606   0.085641   2.623 0.011793 *
## stateDE      0.242414   0.067798   3.576 0.000835 ***
## stateFL      0.183090   0.073189   2.502 0.015984 *
## stateGA     -0.017651   0.067915  -0.260 0.796100
## stateIA      0.069703   0.069904   0.997 0.323916
## stateID     -0.119940   0.068841  -1.742 0.088142 .
## stateIL      0.101860   0.071709   1.420 0.162213
## stateIN      0.166653   0.068135   2.446 0.018333 *
## stateKS     -0.016880   0.067964  -0.248 0.804951
## stateKY      0.334593   0.071542   4.677 2.58e-05 ***
## stateLA      0.145811   0.068569   2.126 0.038860 *
```

```
## stateMA      0.140108   0.078244   1.791 0.079931 .
## stateMD     -0.075703   0.067854  -1.116 0.270358
## stateME      0.237828   0.072314   3.289 0.001934 **
## stateMI      0.246150   0.080069   3.074 0.003544 **
## stateMN      0.186660   0.079611   2.345 0.023416 *
## stateMO      0.127731   0.067728   1.886 0.065628 .
## stateMS      0.090566   0.068273   1.327 0.191217
## stateMT     -0.158804   0.067765  -2.343 0.023485 *
## stateNC      0.071140   0.071434   0.996 0.324515
## stateND     -0.003623   0.071412  -0.051 0.959758
## stateNE     -0.001406   0.069445  -0.020 0.983938
## stateNH      0.471720   0.067662   6.972 1.00e-08 ***
## stateNJ      0.110496   0.074559   1.482 0.145162
## stateNM     -0.281978   0.068497  -4.117 0.000158 ***
## stateNV      0.320003   0.076739   4.170 0.000133 ***
## stateNY      0.106181   0.078329   1.356 0.181853
## stateOH      0.114078   0.067700   1.685 0.098754 .
## stateOK      0.124564   0.067928   1.834 0.073162 .
## stateOR      0.057357   0.068822   0.833 0.408925
## statePA      0.095985   0.069629   1.379 0.174709
## stateRI      0.253760   0.074983   3.384 0.001468 **
## stateSC     -0.029784   0.069276  -0.430 0.669253
## stateSD     -0.058327   0.067647  -0.862 0.393040
## stateTN      0.184855   0.067821   2.726 0.009048 **
## stateTX      0.020787   0.072590   0.286 0.775883
## stateUT     -0.483985   0.070562  -6.859 1.48e-08 ***
## stateVA      0.050808   0.067873   0.749 0.457924
## stateVT      0.269097   0.068215   3.945 0.000271 ***
## stateWA      0.135134   0.092001   1.469 0.148687
## stateWI      0.161820   0.075760   2.136 0.038038 *
## stateWV      0.129671   0.068524   1.892 0.064749 .
## stateWY      0.044147   0.068171   0.648 0.520465
##
## Diagnostic tests:
##                  df1 df2 statistic  p-value
## Weak instruments   1  46    91.451 1.65e-12 ***
## Wu-Hausman         1  45     8.905  0.00458 **
## Sargan             0  NA        NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06764 on 46 degrees of freedom
## Multiple R-Squared: 0.9627,  Adjusted R-squared: 0.9229
## Wald test: 24.36 on 49 and 46 DF,  p-value: < 2.2e-16
```