ECOMAss2

NathanAung

April 18, 2017

```
knitr::opts_chunk$set(echo = TRUE)
set.seed(42)
library(R6) #using R6 oop system
library(AER)
```

```
## Warning: package 'AER' was built under R version 3.3.2

## Loading required package: car

## Warning: package 'car' was built under R version 3.3.2

## Loading required package: lmtest

## Warning: package 'lmtest' was built under R version 3.3.2

## 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
```

```
library(stargazer)
```

```
## Warning: package 'stargazer' was built under R version 3.3.2

##
## Please cite as:

##  Hlavac, Marek (2015). stargazer: Well-Formatted Regression and Summary Statistics Tables.

##  R package version 5.2. http://CRAN.R-project.org/package=stargazer
```

Q1

Define two classes that generates results:

```
#Class that generates all combinations in a vector
regMCloop = R6Class(
  public = list(
    nrange=NULL,
    sigmarange=NULL,
    Urange=NULL,
    a1range=NULL,
    a2range=NULL,
    results = NULL,
    initialize = function(nrange,sigmarange,Urange,a2range=0,a1range=1)
    {
      self$nrange = nrange
      self$sigmarange = sigmarange
      self$Urange = Urange
      self$a2range = a2range
      self$a1range = a1range
      self$results = vector("list",length(nrange)*length(sigmarange)*length(Urange)*length(a2range))#li
    },

    loop = function( a3=1, b1=1, b2=1, b3=0,IV=FALSE)
    {
      i=1#elements of result vector
      for(n in self$nrange)
      {
        for(sigma in self$sigmarange)
        {
          for(dist in self$Urange)
          {
            for(a2 in self$a2range)
            {
              for(a1 in self$a1range)
              self$results[[i]] = regMClass$new(ns=n,sigma=sigma,dist=dist, a2=a2, a3=a3, b1=b1, b2=b2,
              i=i+1#next result
            }
          }
        }
      }
    }


  )
)
```

```
regMClass = R6Class(
  public = list(
    bias = NULL,
    sd = NULL,
    CovRate=NULL,
    CovLength=NULL,
    ns = NULL,
    sigma= NULL,
    dist= NULL,
    a2 =NULL,


    initialize = function(ns = 20, n=1000, b1=1, b2=1, b3=0, a1=1, a2=0, a3=1, sigma=1,dist="norm", IV=
    {
      intervalrange =vector(mode="numeric",length=0)
      b1list = vector(mode="numeric",length=0)
      inInterval=vector(mode="logical",length=0)#declare vectors so that append method works
      for(j in 1:n)
      {
        #__init__
        Z1 = rnorm(n=ns,mean=0,sd=1)
        X2 = rnorm(n=ns,mean=0,sd=1)
        X3 = rnorm(n=ns,mean=0,sd=1)
        V = rnorm(n=ns,mean=0,sd=1)

        if (dist == "norm")
        {
          U = rnorm(n=ns,mean=0,sd=1)
        }
        else
        {
          U = rlnorm(n=ns,mean=0,sd=1)
        }
        #Generate X1 and Y
        X1 = a1*Z1 + a2*X2 + a3*X3 +V
        Y = b1*X1 + b2*X2 +b3*X3 + sigma*U
        #Estimates
        if(IV==FALSE){
        hat = lm(Y ~ X1+X2)
        }

        else{
        hat = ivreg(Y~X1+X2|X2+Z1)
        }
        b1list=append(b1list,hat$coefficients["X1"])
        #In confidence interval?
        interval = confint(hat,parm = "X1",interval="confidence")
        inInterval = append(inInterval,b1<interval[2]&b1>interval[1])
        intervalrange = append(intervalrange,interval[2]-interval[1])
      }
      self$ns = ns # so we can sort by values used
      self$sigma = sigma
      self$dist = dist
      self$a2 = a2
      self$bias = mean(b1list)-1
      self$sd = sd(b1list)
      self$CovRate = mean(inInterval)
      self$CovLength = mean(intervalrange)
    }
  )
)
```

We use a nested loop to generate all combinations needed and store them in a vector of results. These results are then stored within an object. It is possible that a recursive function approach could have been more elegant but for the number of combinations here, nested loop should suffice.

```
dataQ1 = regMCloop$new(nrange=c(20,200,2000),sigmarange=c(1,2),Urange=c("norm","lognorm"))
dataQ1$loop()
```

Here we create an instance of the class and use the loop methods to generate the results. This does take some time(~20s) as we are looping over quite a large number of iterations.

Table 1:

	ndf	sigma	Distribution	Bias	SD	CovRate	CovLength
1	20	1	Normal	0.005	0.148	0.949	0.592
2	20	1	LogNormal	-0.012	0.301	0.956	1.100
3	20	2	Normal	0.001	0.287	0.958	1.153
4	20	2	LogNormal	0.016	0.621	0.946	2.228
5	200	1	Normal	-0.003	0.039	0.957	0.162
6	200	1	LogNormal	0.004	0.090	0.954	0.337
7	200	2	Normal	-0.001	0.085	0.948	0.325
8	200	2	LogNormal	0.009	0.182	0.939	0.666
9	$2,000$	1	Normal	0.001	0.013	0.946	0.051
10	$2,000$	1	LogNormal	0.001	0.027	0.953	0.109
11	$2,000$	2	Normal	0.001	0.026	0.944	0.101
12	$2,000$	2	LogNormal	0.001	0.056	0.950	0.218

a)

In all combinations, |bias| is unanimously within ± 0.005 of 0. Considering that even under all combinations in which the number of samples is 20 that this is present, these results strongly indicate that the estimator is or very close to unbiased. It is impossible to say definitively due to the stochastic nature of the simulation, however the weak law of large numbers and central limit theorem point to this conclusion. This seems appropriate considering the theoretical aspects of the estimator. (more explanation needed here?)

b)

For each combination, of σ and distribution, we vary the number of samples and observe the results. In nearly all cases, bias decreases monotonically throughout. In the few cases where it does not, bias is very close to zero and is still within the same order of magnitude of the previous estimate. This is to be expected even with a consistent estimator as the estimator becomes closer and closer to the true value of the variable. Perhaps with a higher number of simulations, this error would be reduced. Further, testing across a larger number of sample sizes may have graphically indicated the monotonicity of the bias as sample size increased. With only three data points, there could be any number of small changes in between.

c)

Coverage rate is consistently within 0.15 of the expected 95% coverage. Coverage rates between normal and log normal distributions appear to be inconclusive. Similarly, σ appears to have limited effect on coverage rate. The number of samples also inconclusive. One could of course perform an anova test to show these and reveal possible interaction effects.

d)

For all combinations, log normal generates much large confidence intervals than its normal counterpart.

This can be seen by examining the variance of normal and log normal distributions. With standard deviation σ, the variance for normal and lognormal respectively is:

$$\sigma^2$$

$$(e^{\sigma^2} - 1)(e^{2\mu+\sigma^2})$$

One can verify that for all values the variance will be greater for the log normal case . Therefore the estimated standard deviation will be much larger for the distribution and thus parameter estimates will have a larger confidence interval for the same coverage rate.

This can be verified again as for unitary σ, the coverage rate is far smaller than their $\sigma = 2$ counterparts.

Sample size reduces interval length drastically for all combinations. 2a)

```
dataQ2a = regMCloop$new(nrange=c(20,200,2000),sigmarange=c(1,2),Urange=c("norm","lognorm"))
dataQ2a$loop(b3=1,a3=1)
```

Table 2:

	ndf	sigma	Distribution	Bias	SD	CovRate	CovLength
1	20	1	Normal	0.328	0.192	0.594	0.778
2	20	1	LogNormal	0.341	0.325	0.730	1.217
3	20	2	Normal	0.319	0.309	0.813	1.279
4	20	2	LogNormal	0.327	0.612	0.887	2.285
5	200	1	Normal	0.331	0.055	0	0.209
6	200	1	LogNormal	0.333	0.091	0.071	0.363
7	200	2	Normal	0.335	0.090	0.035	0.350
8	200	2	LogNormal	0.333	0.169	0.486	0.683
9	2, 000	1	Normal	0.333	0.016	0	0.065
10	2, 000	1	LogNormal	0.335	0.030	0	0.116
11	2, 000	2	Normal	0.334	0.028	0	0.109
12	2, 000	2	LogNormal	0.335	0.057	0	0.222

Calculate bias here: It is clear that omitted variable bias is present here with an apparent consistent bias of ~0.33. We can calculate the theoretical bias here:

$$y = X_1\beta_1 + X_2\beta_2 + X_3\beta_3 + U$$

$$X_1 = \alpha_1 Z_1 + \alpha_3 X_3 + V_i$$

OLS estimates:

$$E(\hat{\beta}_1) = E\frac{\text{Cov}(X_1, y)}{\text{Var}(X_1)}$$

$$E(\hat{\beta}_1) = E\frac{\text{Cov}(X_1, X_1\beta_1 + X_2\beta_2 + X_3\beta_3 + U)}{\text{Var}(Z_1 + X_3 + U)}$$

$$E(\hat{\beta}_1) = E\frac{\text{Cov}(X_1, X_1\beta_1) + \text{Cov}(X_1, X_2\beta_2) + \text{Cov}(X_1, X_3\beta_3) + \text{Cov}(X_1, U)}{\text{Var}(Z_1 + X_3 + U)}$$

$$E(\hat{\beta}_1) = E\frac{\beta_1 Var(X_1) + \beta_2 0 + \text{Cov}(Z_1 + X_3 + U, X_3\beta_3) + 0}{\text{Var}(Z_1 + X_3 + U)}$$

$$E(\hat{\beta}_1) = E(\beta_1 + \frac{\text{Cov}(Z_1 + X_3 + U, X_3\beta_3)}{\text{Var}(Z_1 + X_3 + U)})$$

$$E(\hat{\beta_1}) = E(\beta_1 + \frac{\beta_3 \mathrm{Cov}(Z_1, X_3) + \beta_3 \mathrm{Cov}(X_3, X_3) + \mathrm{Cov}(U, X_3)}{1+1+1})$$
$$E(\hat{\beta_1}) = \beta_1 + \frac{1}{3}$$
$$\text{Bias} = \frac{1}{3}$$

Standard deviation is also uniformly higher compared to a) and as such confidence intervals are larger. Coverage rate is especially poor here indicating a severe deviation from the requirements and assumptions of linear regression.

Whilst X_3 determines X_1 in both 1) and 2a), it does not determine Y in 1). As such, it does not fulfill the requirements for an omitted variable bias as endogeniety does not occur:

1) The omitted variable must be a determinant of the dependant variable(ie. regression coefficient is non-zero)
2) Omitted variable is correlated with the explanatory variable.

Whilst condition 2) is present through both questions, condition 1 only holds for question 2a).

2b)

```r
dataQ2b = regMCloop$new(nrange=c(20,200,2000),sigmarange=c(1,2),Urange=c("norm","lognorm"))
dataQ2b$loop(b3=1,a3=0)
```

Table 3:

	ndf	sigma	Distribution	Bias	SD	CovRate	CovLength
1	20	1	Normal	-0.007	0.247	0.954	1.025
2	20	1	LogNormal	-0.005	0.421	0.954	1.572
3	20	2	Normal	0.001	0.392	0.955	1.636
4	20	2	LogNormal	0.006	0.810	0.952	2.803
5	200	1	Normal	-0.003	0.073	0.949	0.281
6	200	1	LogNormal	0.004	0.116	0.963	0.461
7	200	2	Normal	0.001	0.117	0.943	0.442
8	200	2	LogNormal	0.001	0.218	0.952	0.861
9	2,000	1	Normal	0.001	0.022	0.955	0.088
10	2,000	1	LogNormal	0.0003	0.036	0.951	0.147
11	2,000	2	Normal	-0.001	0.035	0.956	0.139
12	2,000	2	LogNormal	-0.002	0.068	0.956	0.273

As expected bias disappears as condition 1 is removed. The another anomalies can be explained as follows: Insert math here

3a) Yes, it satisfies the two conditions required for an IV.

1) Z is not correlated with Y other than through X_1
2) Z is correlated with X_1

3b)

```r
dataQ3 = regMCloop$new(nrange=c(20,200,2000),sigmarange=c(1,2),Urange=c("norm","lognorm"))
dataQ3$loop(b3=1,IV=TRUE)
```

Asymptotic convergence is slower but bias is reduced eventually. Standard deviations are higher due to higher presence of multicollinearity.

We can also prove that the IV estimate has higher standard deviation than the OLS one.

Table 4:

	ndf	sigma	Distribution	Bias	SD	CovRate	CovLength
1	20	1	Normal	-0.195	1.908	0.949	7.389
2	20	1	LogNormal	-0.212	3.076	0.961	12.309
3	20	2	Normal	-0.138	1.731	0.961	6.757
4	20	2	LogNormal	-0.089	2.283	0.964	7.319
5	200	1	Normal	-0.004	0.105	0.945	0.403
6	200	1	LogNormal	-0.010	0.171	0.950	0.662
7	200	2	Normal	0.005	0.163	0.956	0.633
8	200	2	LogNormal	-0.016	0.317	0.950	1.209
9	$2,000$	1	Normal	-0.0005	0.032	0.944	0.124
10	$2,000$	1	LogNormal	0.001	0.056	0.940	0.209
11	$2,000$	2	Normal	0.0003	0.049	0.954	0.196
12	$2,000$	2	LogNormal	0.003	0.101	0.950	0.387

4)

```
dataQ4 = regMCloop$new(nrange=200,sigmarange=1,Urange="normal",a1range=seq(0.8,0,by=-0.1))
dataQ4$loop(b3=1)
```

As the correlation between Z and X decreases, Z becomes takes on more and more of the characteristics of a weak instrument until it becomes an invalid instrument at 0. Bias increases accordingly, and the convergence of the estimator is slower.