SCRIPT MOD4S1B: ENDOGENEITY AND TSLS

INSTRUCTOR: KLAUS MOELTNER

Load and Prepare Data

This example is based on Greene's consumption regression in Example 5.3 (5th edition). As shown in the lecture notes, in this case the correlation between **X** and the error term arises due to endogeneity.

```
R> data<- read.table('c:/Klaus/AAEC5126/R/data/consumption.txt', sep="\t", header=FALSE)
R> #
R> #assign variable names
R> names(data)[1]<-"year"
R> names(data)[2]<-"quarter"
R> names(data)[3]<-"realgdp"
R> names(data)[4]<-"realcons"
R> names(data)[5]<-"realinv"
R> names(data)[6]<-"realgov"
R> names(data)[7]<-"realdpi"
R> names(data)[8]<-"cpi"
R> names(data)[9]<-"m1"
R> names(data)[10]<-"tbill"
R> names(data)[11]<-"unemp"
R> names(data)[12]<-"pop"
R> names(data)[13]<-"infl"
R> names(data)[14]<-"realint"
R> #
R> save(data, file = "c:/Klaus/AAEC5126/R/data/consumption.rda")
R> attach(data)
```

Variable definitions:

```
% Contents of data
%%%%%%%%%%%%%%%%%%%%%%
% 1     Year = Date
% 2     Qtr = Quarter
% 3     Realgdp = Real GDP ($billion)
% 4     Realcons = Real consumption expenditures ($billion)
% 5     Realinvs = Real investment by private sector ($billion)
% 6     Realgovt = Real government expenditures ($billion)
% 7     Realdpi = Real disposable personal income ($billion)
% 8     CPI_U = Consumer price index
% 9     M1 = Nominal money stock
% 10   Tbilrate = Quarterly average of month end 90 day t bill rate
% 11   Unemp = Unemployment rate
% 12   pop = Population, million
% 13   Infl = Rate of inflation (first observation is missing and set to zero)
% 14   Realint = Ex post real interest rate = Tbilrate - Infl.
```

```
%       (First observation missing and set to zero)
```

Simple OLS

```
R> # Define variables
R> n<-nrow(data)
R> y<-realcons[2:n]
R> ylag<-realcons[1:(n-1)]
R> dpi<-realdpi[2:n]
R> dpilag<-realdpi[1:(n-1)]
R> n<-length(y)  #IMPORTANT - re-define n!
R> #
R> X<-cbind(rep(1,n),dpi)
R> k<-ncol(X)
R> #
R> bols<-solve((t(X)) %*% X) %*% (t(X) %*% y)# compute OLS estimator
R> e<-y-X%*%bols # Get residuals.
R> SSR<-(t(e)%*%e)#sum of squared residuals - should be minimized
R> s2<-(t(e)%*%e)/(n-k) #get the regression error (estimated variance of "eps").
R> s2ols<-s2 #for Hausman test below
R> Vb<-s2[1,1]*solve((t(X))%*%X) # get the estimated VCOV matrix of bols
R> se=sqrt(diag(Vb)) # get the standard erros for your coefficients;
R> tval=bols/se # get your t-values.
R> #
R> tt<-data.frame(col1=c("constant","dpi"),
               col2=bols,
               col3=se,
               col4=tval)
R> colnames(tt)<-c("variable","estimate","s.e.","t")
```

Table 1. OLS output

variable	estimate	s.e.	t
constant	-81.24659	14.42952	-5.63058
dpi	0.92188	0.00390	236.58551

TSLS

We will instrument dpi by lagged dpi and lagged consumption.

```
R> # Build instrument matrix
R> Z<-cbind(rep(1,n),dpilag,ylag)
R> Xhat<-Z %*% solve(t(Z) %*% Z) %*% t(Z) %*% X
R> k<-ncol(Xhat)  #Don't forget to update k!
R> #
R> btsls<-solve((t(Xhat)) %*% Xhat) %*% (t(Xhat) %*% y)# compute OLS estimator
R> e<-y-X%*%btsls # careful - don't use Xhat here!
R> SSR<-(t(e)%*%e)#sum of squared residuals - should be minimized
R> s2<-(t(e)%*%e)/n #no need to correct for k in this case
```

```
R> Vb<-s2[1,1]*solve((t(Xhat))%*%Xhat) # get the estimated VCOV matrix of bols
R> se=sqrt(diag(Vb)) # get the standard erros for your coefficients;
R> tval=btsls/se # get your t-values.
R> #
R> tt<-data.frame(col1=c("constant","dpi"),
                 col2=btsls,
                 col3=se,
                 col4=tval)
R> colnames(tt)<-c("variable","estimate","s.e.","t")
```

Table 2. TSLS output

variable	estimate	s.e.	t
constant	-81.95259	14.36040	-5.70685
dpi	0.92209	0.00388	237.77128

Hausman Test

```
R> d<-btsls-bols
R> W<-solve(t(Xhat) %*% Xhat)- solve(t(X) %*% X)
R> H<-(t(d) %*% pseudoinverse(W) %*% d)/s2ols[1,1]   #Note use of OLS s2
R> J<-1
R> pval=1-pchisq(H,J)
```

The Hausman test statistic is 8.4814. The corresponding p-value is 0.0036.

Wu Test

```
R> # Step 1: regress dpi on Z and capture predicted values
R> dpihat<- Z %*% solve(t(Z) %*% Z) %*% t(Z) %*% dpi
R> #
R> # Step 2: add predicted values to original regression
R> X<-cbind(rep(1,n),dpi,dpihat)
R> k<-ncol(X)
R> bwu<-solve((t(X)) %*% X) %*% (t(X) %*% y)# compute OLS estimator
R> e<-y-X%*%bwu # Get residuals.
R> s2<-(t(e)%*%e)/(n-k) #get the regression error (estimated variance of "eps").
R> Vb<-s2[1,1]*solve((t(X))%*%X) # get the estimated VCOV matrix of bols
R> #
R> # Step 3: Perform F-test
R> Rmat<-matrix(c(0, 0, 1),nrow=1)
R> q<- 0
R> J<-nrow(Rmat)
R> b<-bwu
R> Fstat<-(1/J)* t(Rmat %*% b-q) %*% solve(Rmat%*%Vb%*%t(Rmat))%*%(Rmat%*%b-q)
R> pval<-1-pf(Fstat,J,n-k)
```

The Wu test statistic is 8.811. The corresponding p-value is 0.0034.

```
R> proc.time()-tic
   user  system elapsed
   0.15    0.06    0.25
```