```
@-----------------------------------------------------------------------------@
@ Ruth Judson, Feb 1996                                                       @
@ Model is Yit = GAMMA * Yit-1 + Xit * BETA + ETAi + EPSILONit                @
@       where Xit = RHO * Xit-1 + XIit                                        @
@       XI~N(0,SigXi), EPSILON~N(0,SigE)                                      @
@ Here SigE is normalized to 1, and RHO is always 0.5                         @
@ Xit is not constructed to be correlated with ETAi, but usually is, hence LSDV@
@ BETA is set to be 1-GAMMA so that the long-run multiplier is 1. GAMMA=0.2,0.8@
@ SigETA is set as MU*SigEP*(1-GAMMA) so that for MU=1, effect of EPS and ETA= @
@ SigS is defined as Var(RHS)-Var(error), the variance of the signal           @
@ SigS and other pars determine SigXi as in Eq 41.                            @
@ Here we also use AH to estimate gamma and compare bias/SE properties        @
@ Note that when the feasible Kiviet correction is used, a consistent est of  @
@ gamma, e.g. from AH, is needed                                              @
@ Kiv6si uses AHIV as starting value and skips GMM stuff since output can be  @
@ obtained from gmm8s.                                                        @
@-----------------------------------------------------------------------------@
new;
clear bols,bhat,bah,kcorr,eah,sigeah;

output file=kiv7chk.out reset; outwidth 200;

@ Set basic parameters: N T Gamma SigE Mu SigS Rho @
ncase=64; npar=7;
load parmat[ncase,npar]=parmat.asc;
load seeds[1100,3]=seeds.asc;

ndraw=1000;
nstart=50;
let mvals=2 5;
nest=4;
k=2;

/*
seed1=666;  "seed1=";;format /rds 1,0; seed1;
ss=rndns(1,1,seed1);  @ Set and then refresh seed for future draws @
*/

"KIV6GAU: Estimation of LSDV with one exog variable and one lagged dep var";
"Follows Kiviet 1995 and Arellano&Bond 1991";
"Second try at 10,000 draws";
"In this version, draw a fresh X vector each time";
"kiv7c checks for wild draws using rho=0.5";
format /rds 1,0; "NDraw=";;ndraw; "NStart=";;nstart;

" T \t N \tSgS\tMu \tPar\t Stat \t  OLS \t LSDV \tLSDVc \tA-HIV";
ic50=int(ndraw*0.5);
ic05=int(ndraw*0.05);
ic25=int(ndraw*0.25);
ic75=int(ndraw*0.75);
ic95=int(ndraw*0.95);
nstat=8;

let statnam=mean stdev median rmse pct05 pct25 pct75 pct95;

@ In outmat, save mean, RMSE, SD, median, 5th, 25th, 75th, 95th percentile @
icase=1; do until icase>ncase;
   outmatb=zeros(nstat,nest);
   outmatg=zeros(nstat,nest);
   pvec=parmat[icase,.]';
   n=pvec[1]; t=pvec[2]; gam=pvec[3]; sige=pvec[4];
   mu=pvec[5]; sigs=pvec[6]; rho=pvec[7];
   rho=0.5;
```

```
mvec=miss(zeros(n,1),0);
bvec=zeros(ndraw,nest);
gvec=zeros(ndraw,nest);

beta=1-gam;
sigeta=mu*sige*(1-gam);
sigxi2 = (1/beta^2)*(sigs - (gam^2/(1-gam^2))*sige^2)*
         (1 + ((gam+rho)^2/(1+gam*rho))*(gam*rho-1) - (gam*rho)^2);
sigxi=sqrt(sigxi2);

@ Form set matrices for docorr @
atmat=eye(t-1) - (1/(t-1))*ones(t-1,t-1);
qvec=ones(1,1)|zeros(k-1,1);

@ Form X here since it is not replicated every run                          @

time0=hsec;
screen on; output off;
idraw=1; do until idraw>ndraw;
   if idraw/100 == int(idraw/100); ".";; endif;
   gosub makex;
   gosub draw1;
   gosub doreg;
   gosub doah;
   gosub docorr;
   gvec[idraw,.]=bols[1]~bhat[1]~bhat[1]+kcorr[1]~bah[1];
   bvec[idraw,.]=bols[2]~bhat[2]~bhat[2]+kcorr[2]~bah[2];
idraw=idraw+1; endo;
screen on; output on;
print;
@=============================================================================
@

@ In new format with fixed seed, print as follows
@

@ For each set of parameters, estimator results across
@

@ Under mean for each estimate, SE, median, 5th, 25th, 75th, 95th percentile
@

@ Outmat holds results for only one case at a time, then is cleared
@

@ Have to loop twice: once to fill outmat by col, then once to print by row
@

@=============================================================================
@

iest=1; do until iest>nest;
        ghold=sortc(gvec[.,iest],1);       bhold=sortc(bvec[.,iest],1);
        gbias=ghold-gam;                   bbias=bhold-beta;
        outmatg[1,iest]=meanc(gbias);      outmatb[1,iest]=meanc(bbias);
        outmatg[2,iest]=stdc(ghold);       outmatb[2,iest]=stdc(bhold);
        outmatg[3,iest]=gbias[ic50];       outmatb[3,iest]=bbias[ic50];
        outmatg[4,iest]=sqrt(meanc((gbias).^2));
        outmatb[4,iest]=sqrt(meanc((bbias).^2));
        outmatg[5,iest]=gbias[ic05];       outmatb[5,iest]=bbias[ic05];
        outmatg[6,iest]=gbias[ic25];       outmatb[6,iest]=bbias[ic25];
        outmatg[7,iest]=gbias[ic75];       outmatb[7,iest]=bbias[ic75];
        outmatg[8,iest]=gbias[ic95];       outmatb[8,iest]=bbias[ic95];
```

```
   iest=iest+1; endo; print;

   "Gamma results, case ";; format /rdt 3,0; icase;
   t;;n;;sigs;;mu;;format /rdt 3,1; gam;;
   istat=1; do until istat>nstat;
            if istat>1; "   \t   \t   \t   \t   \t";; endif;
            format /rdt 6,6; $statnam[istat];; format /rdt 6,3;
            outmatg[istat,.];

   istat=istat+1; endo;

   "Beta results, case ";; format /rdt 3,0; icase;
   t;;n;;sigs;;mu;;format /rdt 3,1; beta;;
   istat=1; do until istat>nstat;
            if istat>1; "   \t   \t   \t   \t   \t";; endif;
            format /rdt 6,6; $statnam[istat];; format /rdt 6,3;
            outmatb[istat,.];

   istat=istat+1; endo;
   "Time to run ";; format /rds 1,0; ndraw;; "draws=";;
   format /rdn 8,2; (hsec-time0)/6000;; " minutes";
icase=icase+1; endo;

stop;
end;
@=============================================================================@
@ Subroutines:                                                                @
@  MakeX creates the X variable                                               @
@  DRAW1 creates the data                                                     @
@  DoReg does the regression                                                  @
@  DoAH does Anderson-Hsiao IV estimation (consistent, but big SEs)           @
@  DoCorr calculates the correction                                           @
@-----------------------------------------------------------------------------@
@ MAKEX                                                                       @
@-----------------------------------------------------------------------------@
makex:

ss=seeds[idraw,1];
ximat=sigxi*rndns(t+nstart,n,ss);

x=zeros(t+nstart,n);
x[1,.]=ximat[1,.];
ii=2; do until ii>t+nstart;
      x[ii,.]=rho*x[ii-1,.] + ximat[ii,.];
ii=ii+1; endo;

dx=mvec'|(x[2:t+nstart,.]-x[1:t+nstart-1,.]);
x=x[1:t+nstart,.];

return;
end;

@-----------------------------------------------------------------------------@
@ DRAW1                                                                       @
@-----------------------------------------------------------------------------@
draw1:


ss=seeds[idraw,2];
eta=sigeta*rndns(n,1,ss);

ss=seeds[idraw,3];
epsmat=sige*rndns(t+nstart,n,ss);
```

```
y=zeros(t+nstart,n);
y[1,.]=epsmat[1,.] + x[1,.] + eta';
ii=2; do until ii>t+nstart;
      y[ii,.]=gam*y[ii-1,.] + beta*x[ii,.] + epsmat[ii,.] + eta';
ii=ii+1; endo;

dy=mvec'|(y[nstart+2:t+nstart,.]-y[nstart+1:t+nstart-1,.]);
dylag=vec(mvec'|dy[1:t-1,.]);
dy=vec(dy);

ylag=mvec'|y[nstart+1:t+nstart-1,.];
ylag2=vec(mvec'|ylag[1:t-1,.]);
ylag=vec(ylag);

dx=mvec'|(x[nstart+2:t+nstart,.]-x[nstart+1:t+nstart-1,.]);
yvec=vec(y[nstart+1:t+nstart,.]);
dxvec=vec(dx);
xvec=vec(x[nstart+1:t+nstart,.]);

return;
end;
@-----------------------------------------------------------------------------@
@ DOREG                                                                       @
@ For now, check bias of uncorrected LSDV                                     @
@-----------------------------------------------------------------------------@
doreg:

@ Do OLS @
yreg=yvec; xreg=ylag~xvec;
k=cols(xreg);
nobs=rows(xreg);
xx=packr(yreg~xreg~ones(nobs,1));
yreg=xx[.,1]; xreg=xx[.,2:k+2];
xxi=invpd(xreg'xreg);
bols=xxi*(xreg'yreg);

@ Do LSDV @
yreg=yvec; xreg=ylag~xvec;
xx=packr(yreg~xreg);
yreg=xx[.,1]; xreg=xx[.,2:k+1];
ylsdv=yreg; xlsdv=xreg;

ii=1; do until ii>n;
      ii1=(ii-1)*(t-1)+1; ii2=ii*(t-1);
      yreg[ii1:ii2]=yreg[ii1:ii2]-meanc(packr(yreg[ii1:ii2]));
      xreg[ii1:ii2,.]=xreg[ii1:ii2,.]-meanc(packr(xreg[ii1:ii2,.]))';
ii=ii+1; endo;
xxi=invpd(xreg'xreg);
bhat=xxi*(xreg'yreg);

ylsdvdm=yreg; xlsdvdm=xreg;
return;
end;
@-----------------------------------------------------------------------------@
@ The correction consists of 3 terms, and is on p. 64 (Thm 1) in Kiviet 1995   @
@ Use initial consistent estimates from AHIV                                  @
@-----------------------------------------------------------------------------@
@ NB: Many of these vectors and matrices can be defined/calculated less often  @
@ Problems: WBar def, values to use for bias calculation @
@-----------------------------------------------------------------------------@
docorr:
```

```
cmat=zeros(t-1,t-1);
i=2; do until i>t-1;
     j=1; do until j>i-1;
          cmat[i,j]=bah[1]^(i-j-1);
     j=j+1; endo;
i=i+1; endo;
ici=sumc(sumc(cmat));
trcac=sumc(diag(cmat'atmat*cmat));
trcacac=sumc(diag(cmat'*atmat*cmat*atmat*cmat));
atc=atmat*cmat;

awtilde=zeros(n*(t-1),1);
i=1; do until i>n;
     i1=(i-1)*(t-1)+1; i2=i*(t-1);
     awtilde[i1:i2]=atc*eah[i1:i2];
i=i+1; endo;
awbar=(xlsdvdm[.,1]-awtilde)~xlsdvdm[.,2:k];
waw=awbar'awbar;

dbar=waw;
dbar[1,1]=dbar[1,1] + sigeah*n*trcac;
dbarinv=invpd(dbar);
corr1=(n/(t-1))*ici*(2*qvec - waw*dbarinv*qvec);

wacaw=zeros(k,k);
i=1; do until i>n;
     i1=(i-1)*(t-1)+1; i2=i*(t-1);
     wacaw=wacaw + awbar[i1:i2,.]'cmat*awbar[i1:i2,.];
i=i+1; endo;
wacawd=wacaw*dbarinv;

corr2=sumc(diag(wacawd))*qvec + wacawd*qvec;
corr3=(sigeah*n*qvec'*dbarinv*qvec) * ((-n/(t-1))*ici*trcac + 2*trcacac)*qvec;

kcorr=sigeah*dbarinv*(corr1+corr2+corr3);

return;
end;

@-----------------------------------------------------------------------------@
@ DoAH                                                                        @
@ Do Anderson-Hsiao IV to get consistent estimate of gamma                    @
@-----------------------------------------------------------------------------@
doah:
yreg=dy; xreg=dylag~dxvec; zreg=ylag2~dxvec;
xx=packr(yreg~xreg~zreg);
yreg=xx[.,1]; xreg=xx[.,2:3]; zreg=xx[.,4:5];
xxi=inv(zreg'xreg);
bah=xxi*(zreg'yreg);

eah=ylsdvdm-xlsdvdm*bah;
sigeah=(eah'eah)/(rows(ylsdvdm)-cols(xlsdvdm)-n);
return;
end;
```