```
@------------------------------------------------------------------------------@
@ GMMTest.GAU: Check diff between 'full' GMM, limited GMM, and iterated GMM    @
@ 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                                               @
@ Kiv3 does GMM as well as AH and LSDV                                         @
@ Kiv4 does limited GMM--max of 10 instruments                                 @
@ Kiv5 does correction using GMM1 as initial consistent estimates              @
@ Note that iterative GMM might be better                                      @
@ Kiv6 uses iterative restricted GMM to get initial cons. estimates            @
@ GMMTest compares various forms of GMM--restricted, iterative, GMM1, GMM2     @
@ GMM8s used fixed seed                                                        @
@ GMM9 uses past and future values for instruments                             @
@      Numbers of instruments are 1,3,7,9                                      @
@------------------------------------------------------------------------------@
new;
time0=hsec;
clear bgmm13,bgmm15,bgmm17,bgmm23,bgmm25,bgmm27,m;

output file=gmm9chk.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;

ncase=rows(parmat);
npar=cols(parmat);
ndraw=1000;
nstart=50;
@ let mvals= 3  5  7;                      @
let mvals= 3    7;                       @ Max # of inst (m=t-2 is full GMM)@
k=2;
@ nest=2*rows(mvals);@
nest=rows(mvals);

"GMMTest: Estimation of GMM restricted to 1, 3 ,5, and 7 instruments";
format /rds 1,0; "NDraw=";;ndraw; "NStart=";;nstart;;"Sige=1";
@ " T \t N \tSgS\tMu \tPar\t Stat \tGMM13 \tGMM15 \tGMM17 \tGMM23 \tGMM25 \tGMM27";@
@ " T \t N \tSgS\tMu \tPar\t Stat \tGMM13 \tGMM17";@
ic50=int(ndraw*0.5);
ic05=int(ndraw*0.05)+1;
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;

icase=1; do until icase>ncase;
   "Working on case ";;format /rdn 1,0; icase;
   outmatg=zeros(nstat,nest);
   outmatb=zeros(nstat,nest);
   time0=hsec;
   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                          @
   screen on; output off;
   idraw=1; do until idraw>ndraw;
@  if idraw/100 == int(idraw/100); "*";; else; ".";; endif;@
   if idraw/100 == int(idraw/100); "*";; endif;
      gosub makex;
      gosub draw1;
      gosub dogmm;
@       gvec[idraw,.]=bgmm13[1]~bgmm15[1]~bgmm17[1]~bgmm23[1]~bgmm25[1]~bgmm27[1];@
@       bvec[idraw,.]=bgmm13[2]~bgmm15[2]~bgmm17[2]~bgmm23[2]~bgmm25[2]~bgmm27[2];@
       gvec[idraw,.]=bgmm13[1]~bgmm17[1];
       bvec[idraw,.]=bgmm13[2]~bgmm17[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]=ghold[ic50]-gam;   outmatb[3,iest]=bhold[ic50]-beta;
           outmatg[4,iest]=sqrt(meanc((gbias.*gbias)));
           outmatb[4,iest]=sqrt(meanc((bbias.*bbias)));
           outmatg[5,iest]=ghold[ic05]-gam;   outmatb[5,iest]=bhold[ic05]-beta;
           outmatg[6,iest]=ghold[ic25]-gam;   outmatb[6,iest]=bhold[ic25]-beta;
           outmatg[7,iest]=ghold[ic75]-gam;   outmatb[7,iest]=bhold[ic75]-beta;
           outmatg[8,iest]=ghold[ic95]-gam;   outmatb[8,iest]=bhold[ic95]-beta;
   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";
   endcase:
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;
@-------------------------------------------------------------------------------@
@ DoGMM                                                                         @
@ Do Arellano-Bond GMM1 and GMM2                                                @
@ Do GMM1 and GMM2 for m=3 and m=5                                              @
@-------------------------------------------------------------------------------@
dogmm:
yreg=dy; xreg=dylag~dxvec;
xx=packr(yreg~xreg);
yreg=xx[.,1]; xreg=xx[.,2:cols(xx)];
@ output on; format /rds 10,6;  "Y=";;yvec; "X=";xvec;@
@-------------------------------------------------------------------------------@
@ Now do restricted GMM @
@-------------------------------------------------------------------------------@
im=1; do until im>rows(mvals); m=mvals[im];

xcol=m*(t-2) - (1/8)*(m^2 - 1);
if m>6; if t==5; xcol=14; else; xcol=xcol-(1/2)*(m-6)*(m-5); endif; endif;

if t>m+2; ycol=(1/2)*m*(m+1) + m*(t-m-2); else; ycol=(1/2)*(t-2)*(t-1); endif;

zcol=xcol+ycol;

amat=zeros(zcol,zcol);                    @ First-stage weighting matrix  @
hmat=2*eye(t-2);                          @ Amat is sum over n z'hz inv.  @
xprimez=zeros(cols(xreg),zcol);
zprimey=zeros(zcol,1);

@ The loop below fills zmat @
in=1; do until in>n;
      ix1=(in-1)*t+1; ix2=in*t;
      in1=(in-1)*(t-2)+1; in2=in*(t-2);
      zzkount=0;
      zmat=zeros(t-2,zcol);
      it=1; do until it>t-2;
            if it<=m;
               zmat[it,zzkount+1:zzkount+it]=yvec[ix1:ix1+it-1]';
               zzkount=zzkount+it;
               if it<(1/2)*(m-3); iz1=ix1;  else; iz1=ix1+it+1-(1/2)*(m-1); endif;
               if t-it<(1/2)*(m+3); iz2=ix2; else; iz2=ix1+it+1+(1/2)*(m-1); endif;
               iz=iz2-iz1+1;
               zmat[it,zzkount+1:zzkount+iz]=xvec[iz1:iz2]';
               zzkount=zzkount+iz;
            else;
               zmat[it,zzkount+1:zzkount+m]=yvec[ix1+it-m:ix1+it-1]';
               zzkount=zzkount+m;
               if it<(1/2)*(m-3); iz1=ix1; else; iz1=ix1+it+1-(1/2)*(m-1); endif;
               if t-it<(1/2)*(m+3); iz2=ix2; else; iz2=ix1+it+1+(1/2)*(m-1); endif;
               iz=iz2-iz1+1;
               zmat[it,zzkount+1:zzkount+iz]=xvec[iz1:iz2]';
               zzkount=zzkount+iz;
            endif;
            if in==1 and it>1; hmat[it,it-1]=-1; hmat[it-1,it]=-1; endif;
      it=it+1; endo;
      amat=amat + zmat'*hmat*zmat;
      xprimez=xprimez + xreg[in1:in2,.]'zmat;
      zprimey=zprimey + zmat'yreg[in1:in2];
```

```
@       if in==1; format /rds 5,2; "Zmat for in=1 is";zmat; endif;@
      clear zmat;
in=in+1; endo;
amat=inv(amat/n);

bgmm1=inv(xprimez*amat*xprimez');
bgmm1=bgmm1*(xprimez*amat*zprimey);
@ format /rds 8,4; "For M=";;m;;", GMM1=";;bgmm1';;@
nugmm1=yreg-xreg*bgmm1;
if m==3; bgmm13=bgmm1; endif;
if m==5; bgmm15=bgmm1; endif;
if m==7; bgmm17=bgmm1; endif;
/*
@--------------------------------------------------------------------------------@
@ Restricted GMM2          @
@--------------------------------------------------------------------------------@
@ Calculate weighting matrix @
amat=zeros(zcol,zcol);
xprimez=zeros(cols(xreg),zcol);
zprimey=zeros(zcol,1);
in=1; do until in>n;
      ix1=(in-1)*t+1; ix2=in*t;
      in1=(in-1)*(t-2) + 1; in2=in*(t-2);
      zzkount=0;
      zmat=zeros(t-2,zcol);
      it=1; do until it>t-2;
            if it<=m;
               zmat[it,zzkount+1:zzkount+it]=yvec[ix1:ix1+it-1]';
               zzkount=zzkount+it;
               if it<(1/2)*(m-3); iz1=ix1;  else; iz1=ix1+it+1-(1/2)*(m-1); endif;
               if t-it<(1/2)*(m+3); iz2=ix2; else; iz2=ix1+it+1+(1/2)*(m-1); endif;
               iz=iz2-iz1+1;
               zmat[it,zzkount+1:zzkount+iz]=xvec[iz1:iz2]';
               zzkount=zzkount+iz;
            else;
               zmat[it,zzkount+1:zzkount+m]=yvec[ix1+it-m:ix1+it-1]';
               zzkount=zzkount+m;
               if it<(1/2)*(m-3); iz1=ix1; else; iz1=ix1+it+1-(1/2)*(m-1); endif;
               if t-it<(1/2)*(m+3); iz2=ix2; else; iz2=ix1+it+1+(1/2)*(m-1); endif;
               iz=iz2-iz1+1;
               zmat[it,zzkount+1:zzkount+iz]=xvec[iz1:iz2]';
               zzkount=zzkount+iz;
            endif;
      it=it+1; endo;
      nui=nugmm1[in1:in2];
      amat=amat + zmat'nui*nui'zmat;
      xprimez=xprimez + xreg[in1:in2,.]'zmat;
      zprimey=zprimey + zmat'yreg[in1:in2];
      clear zmat;
in=in+1; endo;

amat=inv(amat/n);

zaz=zmat*amat*zmat';
bgmm2=inv(xprimez*amat*xprimez');
bgmm2=bgmm2*(xprimez*amat*zprimey);
@ "GMM2=";;bgmm2'; @
nugmm2=yreg-xreg*bgmm2;
if m==3; bgmm23=bgmm2; endif;
if m==5; bgmm25=bgmm2; endif;
if m==7; bgmm27=bgmm2; endif;
*/
im=im+1; endo;
```

```
return;
end;
```