APPENDIX SIX

SAS CODE FOR THE SIMULATION OF THE COMMON VARIATES MODEL

```
*Programming code for simulating 5000 datasets of size 4000 and then
 estimating the CVA/time model.  P=3 is the number of variables, t=3 is the
 number of occasions, and m=4 is the number of groups.  tn=4000 is the total
 sample size and gn=1000 is the sample size in each group.;

%let P=3; %let t=3; %let m=4;
%let count=5000;  %let tn=4000;   %let gn=1000;


*The code generates a dataset, then estimates both the common variate model
 and the unique variate model.  The global do loop invokes several macros
 repeatedly.  Hence the macros are discussed first.;


*fititu and fititc are macros that calculate the fit of the estimates for
 the unique variates and the common variates models, respectively.  These
 fits are used in the calculation of the maximum likelihood test statistic.;

%macro fititu;
fit=j(9,9,0);v1=b[1:3,];v2=b[4:6,];v3=b[7:9];
  %do g=1 %to &m;
    %do s=1 %to &t;
      res&g&s=x&g&s-e[&g,&s]*v&s;
    %end;
    res&g=res&g.1//res&g.2//res&g.3;
    res&g.res=res&g*res&g`;
    fit=&gn*res&g.res+fit;
  %end;
%mend;

%macro fititc;
fit=j(9,9,0);     v=b[1:3,];
  %do g=1 %to &m;
    %do s=1 %to &t;
      res&g&s=x&g&s-e[&g,&s]*v;
    %end;
```

```
    res&g=res&g.1//res&g.2//res&g.3;
    res&g.res=res&g*res&g`;
    fit=&gn*res&g.res+fit;
  %end;
%mend;
```

*The macros makmat and xbar set up the data matrices that are used in the algorithm. Makmat inverts S (the sample covariance) and then breaks it into smaller submatrices. xbar sets up the group means.;

```
%macro makmat;
  Si=inv(S);
  %do i=1 %to &t;
    %do j=1 %to &t;
      a=&i*&t-(&t-1); k=&i*&t;   g=&j*&t-(&t-1); d=&j*&t;
      Si&i&j=Si[a:k,g:d];
    %end;
  %end;
%mend;

%macro xbar;
%do g=1 %to 4;
  %do s=1 %to 3;
     j1=&s*3-2; j2=&s*3;
     x&g&s= fava[&g,j1:j2]`;
  %end;
%end;
%mend;
```

*The macros normalF and normalu evaluate the normal equations for the common variates and the unique variates models respectively.;

```
%macro normalF;  v=B[1:&p,];  lm=B[L,1];  e=j(&m,&t,0);
  %do g=1 %to &m;
    j1=&P+(&g-1)*&t+1;  j2=&P+&g*&t;
    e[&g,]=B[j1:j2,]`;
  %end;
  F&w=j(L,1,0); wr=j(&P,1,0);
  %do g=1 %to &m;
    %do q=1 %to &t;
      %do s=1 %to &t;
      wn=e[&g,&q]*e[&g,&s]*Si&q&s*v - e[&g,&q]*Si&q&s*x&g&s +lm*v;
      wr=wr+wn;
      %end;
    %end;
  %end;
  D=j(&m,&t,0);
  %do g=1 %to &m;
    %do q=1 %to &t;
       %do s=1 %to &t;
         D[&g,&q]=D[&g,&q]+(0.5)*e[&g,&s]*v`*Si&q&s*v+
                    (0.5)*e[&g,&s]*v`*Si&s&q*v - v`*Si&q&s*x&g&s;
       %end;
    %end;
  %end;
  norm=ssq(v)-1;
  F&w=wr//D[1,]`//D[2,]`//D[3,]`//D[4,]`//norm;
```

```
%mend;

%macro normalu; nv=&p*&t; v=B[1:nv,];  e=j(&m,&t,0);
  %do r=1 %to &t;
    j1=&p*&r-&p+1;      j2=&p*&r;
    v&r=v[j1:j2,];      j3=&p*&t + &t*&m + &r;
    lm&r=B[j3,1];
  %end;
  %do g=1 %to &m;
    j1=(&p-1)*&t + &g*&t + 1;  j2=(&p-1)*&t + &g*&t + &t;
    e[&g,]=B[j1:j2,]`;
  %end;
  F&w=j(L,1,0);
  %do r=1 %to &t;      w&r=j(&P,1,0);
    %do g=1 %to &m;
      %do s=1 %to &t;
      wn=-e[&g,&r]*e[&g,&s]*Si&r&s*v&s + e[&g,&r]*Si&r&s*x&g&s +
        e[&g,&r]*e[&g,&s]*v&r`*Si&r&s*v&s*v&r-e[&g,&r]*v&r`*Si&r&s*x&g&s*v&r;
      w&r=w&r+wn;
      %end;
    %end;
    w&r=w&r+lm&r*v&r;
  %end;
  D=j(&m,&t,0);
  %do g=1 %to &m;
    %do r=1 %to &t;
       %do s=1 %to &P;
         D[&g,&r]=D[&g,&r] + e[&g,&s]*v&r`*Si&r&s*v&s - v&r`*Si&r&s*x&g&s;
       %end;
    %end;
  %end;
  norm=j(&t,1,0);
  %do s=1 %to &t;
    norm[&s,1]=ssq(v&s)-1;
  %end;
  F&w=w1//w2//w3//D[1,]`//D[2,]`//D[3,]`//D[4,]`//norm;
%mend;



proc iml;
BigBc=j(&count,15,0);BigBu=j(&count,23,0);
devmatc=j(&count,1,0);  devmatu=j(&count,1,0);   llcm=j(&count,5,0);
llum=j(&count,5,0); testy=j(&count,2,0);       llc=j(1,5,0); llu=j(1,5,0);
test=j(1,2,0);

do ttt=1 to &count;



*First a data set is created.  W is the covariance matrix of the data, v is
 the common canonical variate, and e is the matrix of group scores.  seed is
 is a 4000 by 9 matrix of zeros which serve as seeds for the random number
 generator.  The random numbers follow the normal distribution.  The matrix
 of random numbers, data, is treated as a matrix of residuals.;

  seed=j(&tn,9,0);  data=normal(seed);

W={  4.8   2.1   1.0   2.4  1.05 0.5  1.2   0.525 0.25,
```

```
     2.1   3.3   1.4   1.05 1.65 0.7  0.525 0.825 0.35,
     1.0   1.4   2.9   0.5  0.7  1.45 0.25  0.35  0.725,
     2.4   1.05  0.5   4.8  2.1  1.0  2.4   1.05  0.5,
     1.05  1.65  0.7   2.1  3.3  1.4  1.05  1.65  0.7,
     0.5   0.7   1.45  1.0  1.4  2.9  0.5   0.7   1.45,
     1.2   0.525 0.25  2.4  1.05 0.5  4.8   2.1   1.0,
     0.525 0.825 0.35  1.05 1.65 0.7  2.1   3.3   1.4,
     0.25  0.35  0.725 0.5  0.7  1.45 1.0   1.4   2.9};

  call svd(a,b,c,W);
  tdata=data*(diag(b)##0.5)*c`;
rhalf=(0.5)##(0.5);
v={0.5, 0.5, 0};    v[3,1]=rhalf;

                     e={1.0 0.0 1.0,
                        0.5 1.0 -.5,
                        -.5 0.0 0.5,
                        -1  -1  -1 };


*The subsequent statements center the data and create a matrix of group
 means and a within-groups error matrix.;

u1=e[1,]`@v;
u2=e[2,]`@v;
u3=e[3,]`@v;
u4=e[4,]`@v;
    u=u1||u2||u3||u4;
d1=j(&gn,1,1)@u1`;
d2=j(&gn,1,1)@u2`;
d3=j(&gn,1,1)@u3`;
d4=j(&gn,1,1)@u4`;
d=d1//d2//d3//d4;   bt=d`*d;
f=d+tdata;

ftot=j(1,9,0);
do i=1 to &tn;
  ftot=ftot+f[i,];
end;
fmean=ftot/&tn;
m=j(&tn,1,1);
 fmeans=m@fmean;
fad=f-fmeans;
sstot=fad`*fad;

f1tot=j(1,9,0);  f2tot=j(1,9,0);  f3tot=j(1,9,0);  f4tot=j(1,9,0);
f1=f[1:1000,]; f2=f[1001:2000,]; f3=f[2001:3000,]; f4=f[3001:&tn,];
do i=1 to &gn;
  f1tot=f1tot+f1[i,];
  f2tot=f2tot+f2[i,];
  f3tot=f3tot+f3[i,];
  f4tot=f4tot+f4[i,];
end;
f1ave=f1tot/&gn;
f2ave=f2tot/&gn;
f3ave=f3tot/&gn;
f4ave=f4tot/&gn;
ftot=(f1ave+f2ave+f3ave+f4ave)/4;
f1ava=f1ave-ftot; f2ava=f2ave-ftot; f3ava=f3ave-ftot; f4ava=f4ave-ftot;
fava=f1ava//f2ava//f3ava//f4ava;      *print fava;
```

```
bhat=(fava`*fava)*&gn;

f1ad=j(&gn,9,0); f2ad=j(&gn,9,0); f3ad=j(&gn,9,0); f4ad=j(&gn,9,0);
do i=1 to &gn;
  f1ad[i,]=f1[i,]-f1ave;
  f2ad[i,]=f2[i,]-f2ave;
  f3ad[i,]=f3[i,]-f3ave;
  f4ad[i,]=f4[i,]-f4ave;
end;
ss1=f1ad`*f1ad; ss2=f2ad`*f2ad; ss3=f3ad`*f3ad; ss4=f4ad`*f4ad;
sst=ss1 + ss2 + ss3 + ss4;
sstotal=bhat + sst;







%xbar

within=sst/&tn; S=within;

%let P=3; %let t=3; %let m=4;  L=&P+&t*&m+1;

%makmat


*The following set of lines is the code that finds the maximum likelihood
 estimate for the common variate model by solving the normal equations.
 The algorithm is a Gauss-Newton.;

  delta=0.000001; epsilon=1;   a=0;   r=10;   h=-2; con=0.01;   dev=0;
  B={.5,0.5,.707,1.0,0.0,1.0,0.5,1.0,-.5,-.5, 0.0, 0.5, -1, -1, -1,0 };
  bold=B;
  do until(alpha<0.000001);
    a=a+1;
    Hessian=j(L,L,0);
    rold=r;
    %let w=1;
    %NormalF
    do co=1 to 16;
      B=bold;     devold=dev;
      B[co,1]=bold[co,1]+delta;
      %let w=2;
      %NormalF
      Hessian[,co]=(F2-F1)/delta;
    end;
    B=bold-epsilon*inv(Hessian+I(16)*con)*F1;
    %let w=3;
    %NormalF
    bdiff=b-bold;  dev=sum(abs(f3));
    *print  a h epsilon f3 f1 dev B;
    sumabsd=sum(abs(f3)-abs(f1));
    ralpha=0;
    if sumabsd<0 then do; epsilon=1;  bold=b; h=-2; con=0.01; devold=dev;end;
    else do;
      b=bold;         dev=devold;
      if epsilon=1 then epsilon=0.1;
      else do;
        h=h+1;
```

```
        con=r**h; epsilon=1;
      end;
    end;
    alpha=abs(sum(f3));            if h=3 then alpha=0;
  end;


*The following lines of code generate the maximum likelihood test
 statistics.;

%fititc
n={&tn};
Sh=within + fit*(1/&tn);  iw=inv(within);
invSh=inv(Sh);

lrscapx=trace(iw*fit);
lrs=n*log(det(Sh))+n*trace(invSh*within)+trace(invSh*fit);
lrs1=n*log(det(Sh)); lrs2=trace(invSh*within); lrs3=trace(invSh*fit);
llc[,1]=lrs1;llc[,2]=lrs2;llc[,3]=lrs3;llc[,4]=lrs;llc[,5]=lrscapx;

BigBc[ttt,]=B[1:15,]`;   devmatc[ttt,]=dev;

Lu=&t*(&P+&m+1);



*The following set of lines is the code that finds the maximum likelihood
 estimate for the unique variate model by solving the normal equations.;

  delta=0.000001; epsilon=1;   a=0;   r=10;   h=-2; con=0.01;   dev=0;
  balt=b; b=j(24,1,0);
  B[1:3,]=Balt[1:3,]; B[4:6,]=Balt[1:3,]; B[7:9,]=Balt[1:3,];
  B[10:21,]=Balt[4:15,]; B[22:24,]={0, 0, 0};
  bold=B;
  do until(alpha<0.000001);
    a=a+1;
    Hessian=j(Lu,Lu,0);
    rold=r;
    %let w=1;
    %Normalu
    do co=1 to 24;
      B=bold;     devold=dev;
      B[co,1]=bold[co,1]+delta;
      %let w=2;
      %Normalu
      Hessian[,co]=(F2-F1)/delta;
    end;
    B=bold-epsilon*inv(Hessian+I(24)*con)*F1;
    %let w=3;
    %Normalu
    bdiff=b-bold;  dev=sum(abs(f3));
    sumabsd=sum(abs(f3)-abs(f1));
    ralpha=0;
    if sumabsd<0 then do; epsilon=1;  bold=b; h=-2; con=0.01; devold=dev;end;
    else do;
      b=bold;         dev=devold;
      if epsilon=1 then epsilon=0.1;
      else do;
        h=h+1;
        con=r**h; epsilon=1;
```

```
      end;
    end;
    alpha=abs(sum(f3));             if h=3 then alpha=0;
  end;


*The following lines of code generate the maximum likelihood test
 statistics.;

%fititu
n={&tn};
Sh=within + fit*(1/&tn);  iw=inv(within);
invSh=inv(Sh);

lrsuapx=trace(iw*fit);
lrus=n*log(det(Sh))+n*trace(invSh*within)+trace(invSh*fit);
lrsu1=n*log(det(Sh)); lrsu2=trace(invSh*within); lrsu3=trace(invSh*fit);
llu[,1]=lrsu1;llu[,2]=lrsu2;llu[,3]=lrsu3;llu[,4]=lrus;llu[,5]=lrsuapx;
test[,1]=lrs-lrus;
test[,2]=lrscapx-lrsuapx;

BigBu[ttt,]=B[1:23,]`; devmatu[ttt,]=dev;
   llum[ttt,1:5]=llu; llcm[ttt,1:5]=llc;  testy[ttt,1:2]=test;
end;

libname wat 'c:\prg'; reset storage=wat.simk5k;
store bigbc bigbu llum llcm testy devmatc devmatu;
quit iml;
```