Options ls=80  ps=59  FormDlim='-'  NoDate  NoNumber;

/*  Estimate Power to discriminate proportional and independence relationships of amt of AI with exposure in a cluster sampling framework.  

This simulation generates lognormally distributed data that have a true proportional relationship with AI handled. Cluster effects are included in the lognormal model. For each set of simulated data the mixed linear model is fitted:

    Log Exposure = a + b*Log AI + ClusterEffect + MUEffect

ClusterEffect and MUEffects are normally distributed random effects with mean 0.

The simulation counts the number of times that b is significantly greater than 0. When expressed as a proportion of the number of simulations, this is just the power.

Because of the symmetry of the problem, the power to detect b>0 given true b=1 (proportionality) is the same as the power to detect b<1 given b=0 (independence).

The AI levels are assummed to be distributed in a logarithmic fashion over the MUs.

2 allocations of AI level to clusters are considered:
(1) maximum spread of AI levels between clusters
(2) maximum spread of AI levels within clusters
In theory, (2) should be more efficient since it reduces any confounding of ai level with cluster.

[ A 3rd configuaration is included which forces all AaiH levels in the same cluster to be identical. This configuration is not considered desireable, but was included out of curiosity. ]

--- L.R. Holden, 12/21/06
________________________________________________________ */

**** Change following macro parameters as desired ****;

*----> True parameters of lognormal distribution ----;

%Let gSD = 4;    ** residual geometric standard deviation **;
    ** Note: if independence is true, then this is the total GSD of exposure. If proportionality is true, this is the GSD of 'normalized exposures'. **;

%Let rho = 0.3;  ** intra-cluster correlation **;

*----> Experimental design parameters ----;

%Let NrLocs = 5;  ** assumed # clusters (e.g. locations) **;
%Let NrReps = 5;  ** assumed # sampling units/cluster **;
%Let NrTot = %Eval( &NrLocs * &NrReps );  ** NrTot = NrLocs*NrReps **;

*---> Amt of AI handled parameters ---;
%Let AIFold = 10;  ** AI Range = (LoAmt, LoAmt*AIFold) **;
%Let AIslope = 1;   ** LogExposure = AIint + AIslope*LogAI **;

*---> Simulation parameters ----;
%Let NrSims = 1000;
%Let Seed = 28718433; ** Fixed seed for reproducibility **;

Title1 "gSD=&gSD, rho=&rho, AI-Range=&AIFold.x, #Clusters=&NrLocs, #MUs/cluster=&NrReps";
Title2 "Percentiles based on &NrSims Simulations";

*---------------------------------------------------------------;

*---> Generate NrSims sets of data according to the study
design and specified lognormal distribution
i.e. X = exposure (=lognormal) and
Y = Ln exposure (=normal) ---;

Data Simmer;
    Retain Seed &Seed;
    Array LogAI{&NrTot} _temporary_;
    Array a1{&NrTot} _temporary_;
    Array a2{&NrTot} _temporary_;
    Array h{&NrTot} _temporary_;

*--> 1st calculate variance components for Y=ln(X) distribution from
macro variables ---;
    sT = log(&gSD);        ** true total residual sd Y **;
    sA = sqrt(&rho)*sT;    ** true cluster sd Y **;
    sE = sqrt(1-&rho)*sT;  ** true rep sd Y **;

*---> Specify a set of NrLocs*NrReps logarithmically spaced
AI levels to use for log-log regression ---;
    aiInt = 0;    ** intercept is arbitrary, just use 0 **;
    LoAmt = 1;    ** Lo AI amt is arbitrary, just use 1 **;
    aIslope = &aIslope;
    NrTot = &NrTot;
    HiAmt = LoAmt*AIFold;
    LogLo = Log(LoAmt);
    LogHi = Log(HiAmt);
    Delta = (LogHi - LogLo)/(NrTot-1);
    Do i=1 to NrTot;
        LogAI[i] = LogLo + (i-1)*Delta;
    End;

*---> 3 Allocations of AI over clusters:
    a1 = max between cluster ai differences
a2 = max within cluster ai diffs
a3 = no diffs within clusters (just for fun) ---;

j=0;
Do k=1 to &NrLocs;
   Do i=1 to &NrReps;
      j+1;
      a1(j) = j;
      a2(j) = k + (i-1)*&NrLocs;
   End;
End;

Epsilon = (LogHi - LogLo)/(&NrLocs-1);
j=0;
Do k=1 to &NrLocs;
   hh = LogLo + (k-1)*Epsilon;
   Do i=1 to &NrReps;
      j+1;
      H{j} = hh;
   End;
End;

*--> Next, Generate NrSims simulated data sets ---;
Do Sim = 1 to &NrSims;
   j=0;
   Do Loc = 1 to &NrLocs;

      *--> Generate cluster effect --;
      LocEff = RanNor(seed)*sA;

      Do Rep = 1 to &NrReps;

         *--> Generate 'rep' effect ---;
         RepEff = RanNor(seed)*sE;

         *--> Generate exposure values for 2 allocations ---;
         j+1;

         *--> AI handled on log scale ---;
         ay1 = LogAI{ a1[j] };
         ay2 = LogAI{ a2[j] };
         ay3 = H{j};

         *--> ai handled on natural scale ---;
         ax1 = exp(ay1);
         ax2 = exp(ay2);
         ax3 = exp(ay3);

         *--> Log exposures ---;
         Y1 = aiInt + aiSlope*ay1 + LocEff + RepEff;
         Y2 = aiInt + aiSlope*ay2 + LocEff + RepEff;
         Y3 = aiInt + aiSlope*ay3 + LocEff + RepEff;

         *--> exposures ---;
X1 = Exp(Y1); ** lognormal **;
X2 = Exp(Y2); ** lognormal **;
X3 = Exp(Y3); ** lognormal **;

Output;
Keep Sim Loc ax1 ax2 ax3 X1 X2 x3 ay1 ay2 ay3 Y1 Y2 Y3;

End; * Rep *
End; * Loc *
End; * Sim *
Run;

*--------------------------------------------------------------------------;
*---> Calculate Mixed Model Linear Regression ---;
*=======; ODS Listing Close; **--> turn off procedure output **;
*---> Regression on configuration 1 ----;

ODS Output estimates=est1;
Proc Mixed Data=Simmer;
   By Sim;
   Class Loc;
   Model Y1 = aY1 / ddfm=KenRog;
   Random Loc;
   Estimate 'Config1' aY1 1;
Run;

*---> Regression on configuration 2 ----;

ODS Output estimates=est2;
Proc Mixed Data=Simmer;
   By Sim;
   Class Loc;
   Model Y2 = aY2 / ddfm=KenRog;
   Random Loc;
   Estimate 'Config2' aY2 1;
Run;

*---> Regression on configuration 3 ----;

ODS Output estimates=est3;
Proc Mixed Data=Simmer;
   By Sim;
   Class Loc;
   Model Y3 = aY3 / ddfm=KenRog;
   Random Loc;
   Estimate 'Config3' aY3 1;
Run;

*======; ODS Listing; **--> turn procedure output back on **;

Data Rez;
Set est1 est2 est3;

*---> Convert 2-sided p-values to 1-sided ---;}
If estimate > 0 then p = Probt/2;
else p = 1-Probt/2;

*--> significance indicator ---;
Sig = (p<=0.05);
Run;

*----> Average Sig to get power proportion Results for all configs ---;
Proc Means Mean Data=Rez;
   Class Label;
   Var Sig;
Run;
endsas;