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

/* ------------------------------------------------------------
     Relative Accuracy of Distribution Estimates assuming
     cluster sampling from a lognormal distribution:

       am, gm, 75th, 90th, 95th percentiles considered

   Determine relatives errors for estimated lognormal distribution
       with sample variance based on
    (1) Simple random sampling assumption
    (2) cluster random sampling assumption

  [ Note: the relative accuracy some estimates, such as GM under (1),
        do have an analytical solution.  But all estimates were
        included in the simulation for consistency ]

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

 /* change the following macro-variables as needed */

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

%Let gM = 1;  ** gM doesnt matter, since only relative accuracy
considered **;

%Let gSD = 4;     ** geometric standard deviation **;
%Let rho = 0.3;  ** intra-class (intra-cluster) correlation (ICC) **;

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

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

   *---> Simulation parameters ----;

%Let NrSims = 10000;
%Let Seed = 28718433; ** Fixed seed for reproducibility **;

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

Title1 "gSD=&gSD, rho=&rho, #Clusters=&NrLocs, #SMTs/cluster=&NrReps";
Title2 "Percentiles based on &NrSims Simulations";

/* ------------------------------------------------------------
     Compute all true values of lognormal distribution
           parameters of interest:
           X ~ lognormal
           Y = log(X)
  ---------------------------------------------------------- */

Data TruVals;

  tYm = log(&gM);   ** true mean Y **;
  tYs = log(&gSD);  ** true total sd Y **;
```

```
  tXgm = exp(tYm);
  tXam = exp(tYm + 0.5*tYs**2);

  tZ75 = Probit(0.75);
  tZ90 = Probit(0.90);
  tZ95 = Probit(0.95);
  tX75 = exp(tYm + tZ75*tYs);
  tX90 = exp(tYm + tZ90*tYs);
  tX95 = exp(tYm + tZ95*tYs);

  Output;
  Drop tYm tYs;
Run;

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

  *---> Generate NrSims sets of data accoring to the study
         design and specified lognormal distribution ---;

Data Simmer;
   Retain Seed &Seed;

    *--> 1st calculate variance components for Y=ln(X) distribution
from macro variables ---;

 mT = log(&gM);          ** true mean Y **;
 sT = log(&gSD);         ** true total sd Y **;
 sA = sqrt(&rho)*sT;     ** true cluster sd Y **;
 sE = sqrt(1-&rho)*sT;   ** true rep sd Y **;

   *--> Next, Generate NrSims simulated data sets ---;

  Do Sim = 1 to &NrSims;

          *--> Generate cluster effects --;
     Do Loc = 1 to &NrLocs;
         LocEff = RanNor(seed)*sA;

               *---> Generate 'rep' effects ---;
         Do Rep = 1 to &NrReps;
            RepEff = RanNor(seed)*sE;

               *---> Generate exposure values ---;

            Y = mT + LocEff + RepEff;  ** normal **;
            X = Exp(Y);  ** lognormal **;
            Output;
            Keep Sim Loc X Y;

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

*----------------------------------------------------------------;
   *---> Calculate Simple Random Sample estimates ---;
```

```
Title3 "95% Bounds for Accuracy: Simple Random Sampling Estimates";

Proc Means noprint data=simmer;
   By Sim;
   Var X Y;
   Output Out=RezSRS (drop=_type_ _freq_)
       Mean = Xam Yam
       Std = Xs Ys;
Run;

Data RezSRS;
   If _n_=1 then Set TruVals;
   Set RezSRS;

    *---> Compute sample estimates and ratios to true values --;
  Xgm = exp(Yam);
  X75 = exp(Yam + tZ75*Ys);
  X90 = exp(Yam + tZ90*Ys);
  X95 = exp(Yam + tZ95*Ys);
  rXam = Xam/tXam;
  rXgm = Xgm/tXgm;
  rX75 = X75/tX75;
  rX90 = X90/tX90;
  rX95 = X95/tX95;
  Keep Sim rXam rXgm rX75 rX90 rX95;
Run;
;

  *---> Determine a 95% probability interval for relative accuracies --
-;

Proc Univariate Noprint data=rezsrs;
   Var rXam rXgm rX75 rX90 rX95;
   Output Out=StatzSRS
       pctlpts = 2.5 97.5
       pctlpre = rXam rXgm rX75 rX90 rX95
       pctlname = _025 _975;
Run;

Data srsneat;
   Set StatzSRS;
   Parameter='rXam'; Lo=rXam_025; Hi=rXam_975; Output;
   Parameter='rXgm'; Lo=rXgm_025; Hi=rXgm_975; Output;
   Parameter='rX75'; Lo=rX75_025; Hi=rX75_975; Output;
   Parameter='rX90'; Lo=rX90_025; Hi=rX90_975; Output;
   Parameter='rX95'; Lo=rX95_025; Hi=rX95_975; Output;
Run;
Data srsneat;
   Set srsneat;
   MaxFold = max(Hi,1/Lo);  ** maximum of upper & lower bounds **;
Run;

Proc print noobs data=srsneat;
  Var Parameter Lo Hi MaxFold;
Run;
```

```
*----------------------------------------------------------------;
   *---> Calculate Cluster Sampling estimates ---;

Title3 "95% Bounds for Accuracy: Cluster Random Sampling Estimates";

*======; ODS Listing Close;  **--> Turn off procedure output;

  *---> Estimate variance components within each simulated sample ---;

ODS Output Estimates=VCout;
Proc Varcomp Data=Simmer Method=ML;
  By Sim;
  Class Loc;
  Model Y = Loc;
Run;

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

Proc Transpose Data=VCout
     Out=Trout (drop=_name_ rename=(Var_Loc_ = vA  Var_Error_=vE));
  By Sim;
  Var Estimate;
  ID VarComp;
Run;

  *--> Compute arithmetic means of Y and X
        (can use SRS estimates since cluster sizes are equal) --;

Proc Means Noprint Data=Simmer;
   By Sim;
   Var Y X;
  Output Out=mout (drop=_type_ _freq_ )
        mean=Yam Xam;
Run;

Data RezClus;
  Merge Mout Trout; By Sim;
Run;

Data RezClus;
   If _n_=1 then Set TruVals;
   Set RezClus;

    *---> Compute sample estimates and ratios to true values --;

  Ys = sqrt( vA + vE); ** sample estimate of total SD **;
  Xgm = exp(Yam);
  X75 = exp(Yam + tZ75*Ys);
  X90 = exp(Yam + tZ90*Ys);
  X95 = exp(Yam + tZ95*Ys);
  rXam = Xam/tXam;
  rXgm = Xgm/tXgm;
  rX75 = X75/tX75;
  rX90 = X90/tX90;
  rX95 = X95/tX95;
  Keep Sim rXam rXgm rX75 rX90 rX95;
Run;
```

```
;

Proc Univariate Noprint data=RezClus;
   Var rXam rXgm rX75 rX90 rX95;
   Output Out=StatzClus
       pctlpts = 2.5 97.5
       pctlpre = rXam rXgm rX75 rX90 rX95
       pctlname = _025 _975;
Run;

Data ClusNeat;
   Set StatzClus;
   Parameter='rXam'; Lo=rXam_025; Hi=rXam_975; Output;
   Parameter='rXgm'; Lo=rXgm_025; Hi=rXgm_975; Output;
   Parameter='rX75'; Lo=rX75_025; Hi=rX75_975; Output;
   Parameter='rX90'; Lo=rX90_025; Hi=rX90_975; Output;
   Parameter='rX95'; Lo=rX95_025; Hi=rX95_975; Output;
Run;
Data Clusneat;
   Set Clusneat;
   MaxFold = max(Hi,1/Lo);
Run;

Proc print noobs data=Clusneat;
  Var Parameter Lo Hi MaxFold;
Run;
```