```
*IN THIS CODE, X = EXPOSURE, {C1-C3} ARE CONFOUNDERS;
*NORMAL, CONSTANT VARIANCE;
proc genmod data=a;
   model x = c1 c2 c3 / maxiter=100;
   *variance of x is sse/df = valuedf;
   ods output Modelfit=ss(where=(criterion="Deviance"));
   output out=x xbeta=xb;*predicted mean is xb;
data ss;set ss;merg=1;
data x;set x;merg=1;
data a1;
   merge x ss;
   by merg;
   sd_x=sqrt(valuedf);*standard deviation;
   pdf_den = pdf("normal",x,xb,sd_x);*denominator density value;
   drop criterion df value valuedf;
run;
*t, CONSTANT VARIANCE;
proc genmod data=a;
   model  x = c1 c2 c3 / maxiter=100;
   output out=x stdresdev=e;
data a1;
   set x;
   pdf_den = pdf("t",e,1);*denominator density value with 1 df;
run;
*NORMAL, HETEROSCEDASTIC VARIANCE, TRUNCATED NORMAL, HETEROSCEDASTIC VARIANCE;
*QLIM CODE PROVIDED BY DR MIGUEL HERNAN;
proc qlim data=a;
   *TO FIT TRUNCATED NORMAL, UNCOMMENT BELOW;
    model x =  c1 c2 c3 ;* / truncated(lb=5 ub=30) ;
    hetero x ~  c1 c2 c3  / link = exp noconst;
    nloptions  tech=newrap maxiter=1000 maxfunc = 2000;
    output out=x predicted  errstd;
data a1;
   set x;
   pdf_den = pdf("normal",x,p_x,errstd_x);
   *above gives normal heteroscedastic;
   p_n = pdf("normal",x,p_x,errstd_x) ;
   p_lb = cdf("normal",5,p_x,errstd_x);
   p_ub = cdf("normal",30,p_x,errstd_x);
   pdf_den_trunc = p_n / (p_lb - p_ub);
   *above gives truncated normal heteroscedastic;
run;
*GAMMA;
proc genmod data=a;
   model  x = c1 c2 c3 / dist=gamma link=log maxiter=100;
   ods output ParameterEstimates=s(where=(Parameter="Scale"));
   output out=x p=xb;
data s;set s;merg=1;rename estimate=scale;
drop parameter df stderr lowerwaldcl upperwaldcl chisq probchisq;
data x; set x;merg=1;
data a1;
   merge x s;
   by merg;
   *determine Gamma shape parameter;
   lambda = xb/scale;
   *estimate denominator density;
   pdf_den = pdf("gamma",x,scale,lambda);
run;
*to obtain numerator density values for all of the above,
fit models without c1, c2, or c3 and repeat data step a1;
*QUANTILE BINNING;
proc rank data=a out=x ties=low groups=10;
var x;ranks x_r;
```

```
proc logistic data=x desc;
    model x_r = c1 c2 c3 / maxiter=100;
   output out=a1 predprobs=i;
run;quit;run;
data a1;
   set a1;
   array ii(*) ip_1-ip_10;
   *ip_1-ip_10 are predicted probabilities of being in category 1 to 10;
   do j=1 to 10;
      *set density to predicted probability of observed exposure category;
      if x_r = j then pdf_den=ii(j);
   end;
   drop j ip_1-ip_10;
run;quit;run;
*set numerator density value to 1/j;
*END;
```

N
N(1,99)
N(5,95)
N(10,90)
NHc
NHc(1,99)
NHc(5,95)
NHc(10,90)
NHh
NHh(1,99)
NHh(5,95)
NHh(10,90)
NHu
NHu(1,99)
NHu(5,95)
NHu(10,90)
t1
t3
t5
G
G(1,99)
G(5,95)
G(10,90)
QB10
QB15
QB20

(A)

N
N(1,99)
N(5,95)
N(10,90)
NHc
NHc(1,99)
NHc(5,95)
NHc(10,90)
NHh
NHh(1,99)
NHh(5,95)
NHh(10,90)
NHu
NHu(1,99)
NHu(5,95)
NHu(10,90)
t1
t3
t5
G
G(1,99)
G(5,95)
G(10,90)
QB10
QB15
QB20

(B)

−0.5
0
0.25
0.5
Bias

```
*************************************************************************;
"CONSTRUCTING INVERSE PROBABILITY WEIGHTS FOR A CONTINUOUS EXPOUSRE"
**AUTHOR: ASHLEY ISAAC NAIMI (ashley.naimi@mcgill.ca)
**VERSION: 3.3
**PURPOSE: GENERATE SIMULATED DATA AND ASSESS DIFFERENT METHODS TO CONSTRUCT IPW
*************************************************************************;
%inc "c:\ain\header.txt";
%inc "c:\ain\spline.txt";
proc printto log="Y:\Documents\Research\Papers\ConstructingWeights\20Jun13\ipw.sim2.log";run;
proc printto
print="Y:\Documents\Research\Papers\ConstructingWeights\20Jun13\ipw.sim2.lst";run;
*IMPORT QUEBEC BIRTH FILE DATA TO GENERATE SIMULATED DATA. KEEP ONLY MATERNAL AND PATERNAL
AGE AND PARITY;
proc import datafile="Y:\Documents\Research\Papers\ConstructingWeights\20Jun13\ipimoms.txt"
     out=moms
     dbms=dlm
     replace;
     delimiter='09'x;
run;
data moms;set moms;id = _N_;keep mage page parity id;run;
*SELECT 3000 IID RANDOM SAMPLES OF QUEBEC BIRTH FILE DATA TO GENERATE SIMULATED EXPOSURE AND
OUTCOME DATA;
proc surveyselect data=moms
                            out=a noprint
                            method=srs
                            reps=50
                            seed=675847
                            sampsize=1500;
run;
proc print data=a (obs=5);run;
*CREATE DISJOINT INDICATOR VARIALBES FOR PARITY;
data a;
       set a;
       array par(*) parity1-parity5;
       do pp = 1 to 5;
              par(pp) = (parity=pp);
       end;
run;
*GENERATE TWO SIMULATED EXPOSURES (x2a and x2c) AND OUTCOMES AS A FUNCTION OF QUEBEC BIRTH
FILE DATA COVARIATES;
data a;
       set a;
       call streaminit(789654);
                    r = rand("bernoulli",.5);
                    mu = .025*mage + .0025*page + .00125*mage*page
                           - .21*parity2 - .22*parity3 - .45*parity4 - .45*parity5;
                    x2a = rand("normal",15 + mu, 1.5);*r + exp(.95*rannor(987))*(1-r);*exp(mu
+ rand("normal",0,.25));*CONTAMINATED-NORMAL DISTRIBUTION: HOMOSCEDASTIC;
                    if x2a < 0 then x2a = 0;
                    jitter = rand("normal");
                    x2c = (rand("Poisson",mu) + jitter);*POISSON DISTRIBUTION:
HETEROSCEDASTIC WITH MEAN = VARIANCE;
                    if x2c < 0 then x2c = 0;
                    p2a = 1/(1+exp(-(-11.5 + log(1.25)*x2a + log(1.7)*sqrt(mage) +
log(1.5)*sqrt(page)
                                                                                +
log(0.75)*parity2 + log(0.8)*parity3 + log(0.85)*parity4 + log(0.9)*parity5)));
                    y2a = rand("bernoulli",p2a);
                    p2c = 1/(1+exp(-(-8.05 + log(1.25)*x2c + log(1.7)*sqrt(mage) +
log(1.5)*sqrt(page)
                                                                                +
log(0.75)*parity2 + log(0.8)*parity3 + log(0.85)*parity4 + log(0.9)*parity5)));
                    y2c = rand("bernoulli",p2c);
```

```
                    smage = sqrt(mage);
                    spage = sqrt(page);
                    logx2a = x2a;
                    llx2a = log(x2a);
                    logx2c = log(x2c+.001);
                    x2cind = (x2c>0);
                    pind = sum(of parity2-parity5);
                    cens=1;
                    magepage = mage*page;
rename replicate=h ;
label replicate = ' ' ;
run;
proc means data=a min max mean median q1 q3 var nmiss maxdec=3;
var x2a x2c logx2c y2a y2c mage page parity2-parity5;run;
*DETERMINE MARGINAL TRUE VALUE USING METHOD OF MARGINAL STANDARDIZATION: Localio AR, ET AL.
Journal of Clinical Epidemiology. 2007;
data check;
       set a;
       linear2anum = -11.5 + log(1.25)*17.984 + log(1.7)*sqrt(mage) + log(1.5)*sqrt(page) +
log(0.75)*parity2 + log(0.8)*parity3 + log(0.85)*parity4 + log(0.9)*parity5;
       p2anum = 1/(1+exp(-(linear2anum)));
       linear2aden = -11.5 + log(1.25)*16.984 + log(1.7)*sqrt(mage) + log(1.5)*sqrt(page) +
log(0.75)*parity2 + log(0.8)*parity3 + log(0.85)*parity4 + log(0.9)*parity5;
       p2aden = 1/(1+exp(-(linear2aden)));

       linear2cnum = -8.05 + log(1.25)*2.067 + log(1.7)*sqrt(mage) + log(1.5)*sqrt(page) +
log(0.75)*parity2 + log(0.8)*parity3 + log(0.85)*parity4 + log(0.9)*parity5;
       p2cnum = 1/(1+exp(-(linear2cnum)));
       linear2cden = -8.05 + log(1.25)*1.067 + log(1.7)*sqrt(mage) + log(1.5)*sqrt(page) +
log(0.75)*parity2 + log(0.8)*parity3 + log(0.85)*parity4 + log(0.9)*parity5;
       p2cden = 1/(1+exp(-(linear2cden)));
run;
proc means data=check noprint mean;
var p2anum p2aden;
var p2cnum p2cden;
output out=meancheck mean=p2anum p2aden p2cnum p2cden;
data meancheck;set meancheck;
truex2a = log((p2anum/(1-p2anum))/(p2aden/(1-p2aden)));truex2c = log((p2cnum/(1-
p2cnum))/(p2cden/(1-p2cden)));
merg = 1;
keep merg truex2a truex2c;
run;

*CONSTRUCTING IPWS;
*MODEL MARGINAL AND CONDITIONAL MEAN OF EXPOSURE TO CONSTRUCT IPW;

*FOR X2C MODEL ZERO-INFLATED PORTION;
ods select none;
proc logistic data=a;
       by h;
       title "Continuous Exposure Model: Numerator Zero Portion";
       model x2cind = / maxiter=100;
       output out=nx2cbin p=nx2cbin;
proc logistic data=a;
       by h;
       title "Continuous Exposure Model: Denominator Binary Portion";
       model x2cind = mage page mage*page parity2-parity5 / maxiter=100;
       output out=dx2cbin p=dx2cbin;

*NORMAL MODEL;
proc genmod data=a;
       by h;
       title "Continuous Exposure Model: Numerator";
```

```
       model logx2a = / maxiter=100;
       ods output Modelfit=ssnx2a(where=(criterion="Deviance"));
       output out=nx2aa xbeta=nxba stdresdev=nx2a;run;
data nx2a;
       merge nx2aa ssnx2a;
       by h;
       varnx2a=sqrt(valuedf);
       drop criterion df value valuedf;
run;
proc genmod data=a;
       by h;*where x2c>0;
       title "Continuous Exposure Model: Numerator";
       model logx2c = / maxiter=100;
       ods output Modelfit=ssnx2c(where=(criterion="Deviance"));
       output out=nx2cc xbeta=nxbc stdreschi=nx2c;
data nx2c;
       merge nx2cc ssnx2c;
       by h;
       varnx2c=sqrt(valuedf);
       drop criterion df value valuedf;
run;
proc genmod data=a;
       by h;
       title "Continuous Exposure Model: Denominator";
       model x2a = mage page mage*page parity2-parity5 / maxiter=100;
       ods output Modelfit=ssdx2a(where=(criterion="Deviance"));
       output out=dx2aa xbeta=dxba stdresdev=dx2a;
data dx2a;
       merge dx2aa ssdx2a;
       by h;
       vardx2a=sqrt(valuedf);
       drop criterion df value valuedf;
run;
proc genmod data=a;
       by h;*where x2c>0;
       title "Continuous Exposure Model: Denominator";
       model logx2c = mage page mage*page parity2-parity5 /  maxiter=100;
       ods output Modelfit=ssdx2c(where = (criterion = "Deviance"));
       output out=dx2cc xbeta=dxbc stdreschi=dx2c;run;
data dx2c;
       merge dx2cc ssnx2c;
       by h;
       vardx2c=sqrt(valuedf);
       drop criterion df value valuedf;
run;
ods select all;
proc plot data=dx2a;
title "Residual Plot A";
plot dx2a*dxba ;
proc plot data=dx2c;
title "Residual Plot C";
plot dx2c*dxbc ;
run;quit;run;

*NORMAL MODELS ACCOUNTING FOR HETEROSCEDASTICITY;
*METHOD IMPLEMENTED BY CERDA ET AL AS CITED IN MAIN TEXT;
ods select none;
proc reg data=a;by h;
       model x2a = mage page magepage parity2 parity3 parity4 parity5;
       output out=resid r=rr;
data resid;
       set resid;
       r2 = rr**2;
```

```
proc genmod data=resid;by h;
       model r2 = mage pind / dist=normal link=log noint;
       output out=x2avar p=x2avar;
proc reg data=a;by h;
       model logx2c = mage page magepage parity2 parity3 parity4 parity5;
       output out=resid r=rr;
data resid;
       set resid;
       r2 = rr**2;
proc genmod data=resid;by h;
       model r2 = mage pind / dist=normal link=log noint;
       output out=x2cvar p=x2cvar;*variance;
run;quit;run;

*METHOD IMPLEMENTED BY HERNAN AND ROBINS CAUSAL INFERENCE BOOK, CHAPTER 12 SECTION 4
(TRUNCATED NORMAL WITH HETEROSCEDASTIC VARIANCE);
proc qlim data=a ;
       by h;
    model x2a =   / truncated(lb=5  ub=30);
    hetero x2a ~    / link = exp noconst;
    nloptions  tech=newrap maxiter=1000 maxfunc = 2000;
    output out = numx2a_th predicted  errstd ;
data numx2a_th;set numx2a_th;rename p_x2a=Pn_x2a errstd_x2a=Errstdn_x2a;run;
proc qlim data=a ;
       by h;
    model x2a =  mage page magepage parity2 parity3 parity4 parity5 / truncated(lb=5  ub=30);
    hetero x2a ~  mage pind  / link = exp noconst;
    nloptions  tech=newrap maxiter=1000 maxfunc = 2000;
    output out = denx2a_th predicted  errstd ;
data denx2a_th;set denx2a_th;rename p_x2a=Pd_x2a errstd_x2a=Errstdd_x2a;
proc qlim data=a ;
       by h;
    model logx2c =   / truncated(lb=-7.5 ub=3.5);
    hetero logx2c ~    / link = exp noconst;
    nloptions  tech=newrap maxiter=1000 maxfunc = 2000;
    output out = numx2c_th predicted  errstd ;
data numx2c_th;set numx2c_th;rename p_logx2c=Pn_x2c errstd_logx2c=Errstdn_x2c;run;
proc qlim data=a ;
       by h;
    model logx2c =  mage page magepage parity2 parity3 parity4 parity5 / truncated(lb=-7.5
ub=3.5);
    hetero logx2c ~  mage pind  / link = exp noconst;
    nloptions tech=newrap maxiter=1000 maxfunc = 2000;
    output out = denx2c_th predicted  errstd ;
data denx2c_th;set denx2c_th;rename p_logx2c=Pd_x2c errstd_logx2c=Errstdd_x2c;
run;quit;run;

*USER WRITTEN CODE FOR HETERO NORMAL;
proc nlmixed data=a tech=NRRIDG maxiter=1000 maxfunc=2000;
       title "Numerator Model";
       by h;
       parms b0=0 sd0=1;
       bounds sd0 > 0;
       *MEAN;
       mu = b0;
       *CONSTANT VARIANCE;
       sigma_sq = sd0;
       sigma = sqrt(sigma_sq);
       model x2a ~ normal(mu,sigma_sq);
       predict mu  out=bbnx2amu;
       predict sigma out=bbnx2asig;
proc nlmixed data=a tech=NRRIDG maxiter=1000 maxfunc=2000;
       title "Denominator Model";
```

```
       by h;
       parms b0=0 b1=0 b2=0 b3=0 b4=0 b5=0 b6=0 b7=0 sigma_sq=1 sd1=1 sd2=1;
       bounds sigma_sq > 0, sd1>0, sd2>0;
       *MEAN;
       mu = b0 + b1*mage + b2*page + b3*mage*page + b4*parity2 + b5*parity3 + b6*parity4 +
b7*parity5;
       *HETEROSCEDASTIC VARIANCE FUNCTION;
       *See, e.g., Davidian and Carroll (1987) JASA Col 82 Issue 400 for justification of
equation;
       sigma_i_sq = log(sigma_sq) + sd1*mage + sd2*pind;
       sigma_i = sqrt(sigma_i_sq);
       model x2a ~ normal(mu,sigma_i_sq);
       predict mu  out=bbdx2amu;
       predict sigma_i out=bbdx2asig;
run;quit;run;
data bbnx2amu;set bbnx2amu;rename pred=nx2amu;run;
data bbnx2asig;set bbnx2asig;rename pred=nx2asig;run;
data bbdx2amu;set bbdx2amu;rename pred=dx2amu;run;
data bbdx2asig;set bbdx2asig;rename pred=dx2asig;run;
proc nlmixed data=a tech=nrridg maxiter=1000 maxfunc=2000;
       by h;
       parms b0=0 g0=0 sd0=1;
       bounds sd0 > 0;
       *CONTINUOUS MEAN PORTION;
       mu = b0;
       *ZERO INFLATED MEAN PORTION;
       pi = 1/(1+exp(-(g0)));
       *CONSTANT VARIANCE;
       sigma_sq = sd0;
       sigma = sqrt(sigma_sq);
       ll = (x2c=0)*(log(pi)) + (x2c>0)*((log(1-pi))+log(pdf('normal',logx2c,mu,sigma)));
       model logx2c ~ general(ll);
       predict mu  out=bbnx2cmu;
       predict pi out=bbnx2cpi;
       predict sigma out=bbnx2csig;run;
proc nlmixed data=a tech=nrridg maxiter=1000 maxfunc=2000;
       by h;
       parms b0=0 b1=0 b2=0 b3=0 b4=0 b5=0 b6=0 b7=0
             g0=0 g1=0  g2=0
             sigma_sq=1 sd1=0 sd2=0;
       bounds sigma_sq > 0;
       *CONTINUOUS MEAN PORTION;
       mu = b0 + b1*mage + b2*page + b3*mage*page + b4*parity2 + b5*parity3 + b6*parity4 +
b7*parity5;
       *ZERO INFLATED MEAN PORTION;
       pi = 1/(1+exp(-(g0 + g1*mage + g2*pind)));
       *HETEROSCEDASTIC VARIANCE;
       log_sigma_i_sq = log(sigma_sq) + sd1*mage + sd2*pind;
       sigma_i = sqrt(exp(log_sigma_i_sq));
       ll = (x2c=0)*log(pi) + (x2c>0)*((log(1-pi))+log(pdf('normal',logx2c,mu,sigma_i)));
       model logx2c ~ general(ll);
       predict mu  out=bbdx2cmu;
       predict pi out=bbdx2cpi;
       predict sigma_i out=bbdx2csig;
run;quit;run;
data bbnx2cmu;set bbnx2cmu;rename pred=nx2cmu;keep h id pred;run;
data bbnx2cpi;set bbnx2cpi;rename pred=nx2cpi;keep h id pred;run;
data bbnx2csig;set bbnx2csig;rename pred=nx2csig;keep h id pred;run;
data bbdx2cmu;set bbdx2cmu;rename pred=dx2cmu;keep h id pred;run;
data bbdx2cpi;set bbdx2cpi;rename pred=dx2cpi;keep h id pred;run;
data bbdx2csig;set bbdx2csig;rename pred=dx2csig;keep h id pred;run;

*GAMMA MODEL;
```

```
proc genmod data=a ;
       by h;
       title "Continuous Exposure Model: Numerator";
       model x2a = / link=log dist=gamma maxiter=100;
       ods output ParameterEstimates=ngx2aa1(where=(Parameter="Scale"));
       output out=ngx2aa2 p=ngx2a;run;
data ngx2a;
       merge ngx2aa1 ngx2aa2;
       by h;
       ngx2ascale = estimate;
       drop parameter df stderr lowerwaldcl upperwaldcl chisq probchisq;

proc genmod data=a ;
       where x2c > 0;
       by h;
       title "Continuous Exposure Model: Numerator Continuous Portion";
       model x2c = / link=log dist=gamma maxiter=100;
       ods output ParameterEstimates=ngx2cc1(where=(Parameter="Scale"));
       output out=ngx2cc2 p=ngx2c;
data ngx2c;
       merge ngx2cc1 ngx2cc2;
       by h;
       ngx2cscale = estimate;
       drop parameter df stderr lowerwaldcl upperwaldcl chisq probchisq;
proc genmod data=a ;
       by h;
       title "Continuous Exposure Model: Denominator";
       model x2a = mage page mage*page parity2-parity5 / link=log dist=gamma maxiter=100;
       ods output ParameterEstimates=dgx2aa1(where=(Parameter="Scale"));
       output out=dgx2aa2 p=dgx2a;
data dgx2a;
       merge dgx2aa1 dgx2aa2;
       by h;
       dgx2ascale = estimate;
       drop parameter df stderr lowerwaldcl upperwaldcl chisq probchisq;

proc genmod data=a ;
       where x2c > 0;
       by h;
       title "Continuous Exposure Model: Denominator Continuous Portion";
       model x2c = mage page mage*page parity2-parity5 / link=log dist=gamma maxiter=100;
       ods output ParameterEstimates=dgx2cc1(where=(Parameter="Scale"));
       output out=dgx2cc2 p=dgx2c;run;
data dgx2c;
       merge dgx2cc1 dgx2cc2;
       by h;
       dgx2cscale = estimate;
       drop parameter df stderr lowerwaldcl upperwaldcl chisq probchisq;
run;quit;run;

*CHECK IF CV IS CONSTANT ACROSS MU (GAMMA ASSUMPTION);
data dx2a;
       set dx2a;
       rr = dx2a**2;
proc genmod data=dx2a;
model rr = dxba / link=log dist=normal;
output out=cv p=sigma;
run;quit;run;
data cv;
       set cv;
       sd = sigma**(1/2);
       cv = sd/dxba;
ods select all;
```

```
proc plot data=cv;
plot cv*dxba;
run;quit;run;
ods select none;
proc reg data=cv;
model cv = dxba;
run;quit;run;

proc genmod data=a;
       where x2c>0;
       model x2c = mage page mage*page parity2-parity5 /  maxiter=100;
       output out=cv xbeta=dxbc stdreschi=dx2c;run;
data cv;
       set cv;
       rr = dx2c**2;
proc genmod data=cv;
where x2c>0;
model rr = dxbc / link=log dist=normal;
output out=cv p=sigma;
run;quit;run;
data cv;
       set cv;
       sd = sigma**(1/2);
       cv = sd/(dxbc);
run;
ods select all;
proc plot data=cv;
plot cv*dxbc;
run;quit;run;
proc reg data=cv;
model cv = dxbc;
run;quit;run;

*QUANTILE BINNING APPROACH;
*RANK CONTINUOUS EXPOSURES INTO CATEGORIES;
proc rank data=a out=cc(keep= h id x2a x2c x2arank10 x2crank10) ties=low groups=10;
by h;var x2a x2c;ranks x2arank10 x2crank10;run;
proc rank data=cc out=cc ties=low groups=15;
by h;var x2a x2c;ranks x2arank15 x2crank15;run;
proc rank data=cc out=cc ties=low groups=20;
by h;var x2a x2c;ranks x2arank20 x2crank20;run;
proc sort data=a;by h id;
proc sort data=cc;by h id;
data bb;merge a cc;by h id;
data bb;set bb;
array catrank(6) x2arank10 x2crank10 x2arank15 x2crank15 x2arank20 x2crank20;
do jj = 1 to 6;
       catrank(jj) = catrank(jj)+1;
end;
run;
*FIT CUMULATIVE LOGISTIC MODELS TO RANKED EXPOSURE VALUES;
proc logistic data=bb desc noprint;
       by h;
       title "Categorical Exposure Model: Denominator 10";
       model x2arank10 = mage page mage*page parity2-parity5 / maxiter=100;
       output out=aa1 predprobs=i;
run;quit;run;
data dcatx2a10;
       set aa1;
       array ii(*) ip_1-ip_10;
       do kappa=1 to 10;
              if x2arank10 = kappa then dencata10=ii(kappa);
       end;
```

```
       drop kappa ip_1-ip_10;
run;
proc logistic data=bb desc noprint;
       by h;
       title "Categorical Exposure Model: Denominator 10";
       model x2crank10 = mage page mage*page parity2-parity5 / maxiter=100;
       output out=aa3 predprobs=i;
run;quit;run;
data dcatx2c10;
       set aa3;
       array ii(*) ip_1-ip_10;
       do kappa=1 to 10;
              if x2crank10 = kappa then dencatc10=ii(kappa);
       end;
       drop kappa ip_1-ip_10;
run;
proc logistic data=bb desc noprint;
       by h;
       title "Categorical Exposure Model: Denominator 15";
       model x2arank15 = mage page mage*page parity2-parity5 / maxiter=100;
       output out=aa1 predprobs=i;
run;quit;run;
data dcatx2a15;
       set aa1;
       array ii(*) ip_1-ip_15;
       do kappa=1 to 15;
              if x2arank15 = kappa then dencata15=ii(kappa);
       end;
       drop kappa ip_1-ip_15;
run;
proc logistic data=bb desc noprint;
       by h;
       title "Categorical Exposure Model: Denominator 15";
       model x2crank15 = mage page mage*page parity2-parity5 / maxiter=100;
       output out=aa3 predprobs=i;
run;quit;run;
data dcatx2c15;
       set aa3;
       array ii(*) ip_1-ip_15;
       do kappa=1 to 15;
              if x2crank15 = kappa then dencatc15=ii(kappa);
       end;
       drop kappa ip_1-ip_15;
run;
proc logistic data=bb desc noprint;
       by h;
       title "Categorical Exposure Model: Denominator 20";
       model x2arank20 = mage page mage*page parity2-parity5 / maxiter=100;
       output out=aa1 predprobs=i;
run;quit;run;
data dcatx2a20;
       set aa1;
       array ii(*) ip_1-ip_20;
       do kappa=1 to 20;
              if x2arank20 = kappa then dencata20=ii(kappa);
       end;
       drop kappa ip_1-ip_20;
run;
proc logistic data=bb desc noprint;
       by h;
       title "Categorical Exposure Model: Denominator 20";
       model x2crank20 = mage page mage*page parity2-parity5 / maxiter=100;
       output out=aa3 predprobs=i;
```

```
run;quit;run;
data dcatx2c20;
	set aa3;
	array ii(*) ip_1-ip_20;
	do kappa=1 to 20;
		if x2crank20 = kappa then dencatc20=ii(kappa);
	end;
	drop kappa ip_1-ip_20;
run;
*CREATE STABILIZED IPWs;
data b;
	merge a nx2cbin dx2cbin nx2a nx2c dx2a dx2c ngx2a ngx2c dgx2a dgx2c x2avar x2cvar
		numx2a_th denx2a_th numx2c_th denx2c_th dcatx2a10 dcatx2a15 dcatx2a20
dcatx2c10 dcatx2c15 dcatx2c20
		bbnx2amu bbnx2asig bbdx2amu bbdx2asig
		bbnx2cmu bbnx2cpi bbnx2csig bbdx2cmu bbdx2cpi bbdx2csig;
	by h id;

	*NORMAL;
	*x2a;
	numx2a=pdf("normal",x2a,nxba,varnx2a);
	denx2a=pdf("normal",x2a,dxba,vardx2a);

	*x2c;
	*ZERO INFLATED PORTION;
	if x2cind = 0 then do;
		numx2c=nx2cbin + (1-nx2cbin)*pdf("normal",0,nxbc,varnx2c);
		denx2c=dx2cbin + (1-dx2cbin)*pdf("normal",0,dxbc,vardx2c);
	end;
	*CONTINUOUS PORTION;
	if x2cind = 1 then do;
		numx2c=(1-nx2cbin)*pdf("normal",logx2c,nxbc,varnx2c);
		denx2c=(1-dx2cbin)*pdf("normal",logx2c,dxbc,vardx2c);
	end;

	*GAMMA;
	*x2a;
	*DETERMINE GAMMA SHAPE PARAMETER  (LAMBDA);
	ngx2alambda = ngx2a/ngx2ascale;
	dgx2alambda = dgx2a/dgx2ascale;

	numgx2a = pdf("gamma",x2a,ngx2ascale,ngx2alambda);
	dengx2a = pdf("gamma",x2a,dgx2ascale,dgx2alambda);

	*x2c;
	*DETERMINE GAMMA SHAPE PARAMETER (LAMBDA);
	*ZERO INFLATED PORTION;
	if x2cind = 0 then do;
		numgx2c = nx2cbin;*b/c gamma pdf defined as zero when x2c = 0, no continuous
portion;
		dengx2c = dx2cbin;
	end;
	*CONTINUOUS PORTION;
	if x2cind = 1 then do;
		ngx2clambda = ngx2c/ngx2cscale;
		dgx2clambda = dgx2c/dgx2cscale;
		numgx2c = (1-nx2cbin)*pdf("gamma",x2c,ngx2cscale,ngx2clambda);
		dengx2c = (1-dx2cbin)*pdf("gamma",x2c,dgx2cscale,dgx2clambda);
	end;

	*NORMAL HETEROSCEDASTIC: USING CERDA ET AL'S METHOD;
	*x2a;
	numx2ah=pdf("normal",x2a,nxba,varnx2a);
```

```
        denx2ah=pdf("normal",x2a,dxba,sqrt(x2avar));

        *x2c;
        *ZERO INFLATED PORTION;
        if x2cind = 0 then do;
               numx2ch=nx2cbin;* + (1-nx2cbin)*pdf("normal",0,nxbc,varnx2c);
               denx2ch=dx2cbin;* + (1-dx2cbin)*pdf("normal",0,dxbc,x2cvar);
        end;
        *CONTINUOUS PORTION;
        if x2cind = 1 then do;
               numx2ch=(1-nx2cbin)*pdf("normal",logx2c,nxbc,varnx2c);
               denx2ch=(1-dx2cbin)*pdf("normal",logx2c,dxbc,sqrt(x2cvar));
        end;

        *NORMAL HETEROSCEDASTIC: USING HERNAN & ROBINS' METHOD;
        *x2a;
        nx2a_th = pdf("normal",x2a,Pn_x2a,Errstdn_x2a);
        ubnx2a_th = cdf("normal",30, Pn_x2a, Errstdn_x2a);
        lbnx2a_th = cdf("normal",5, Pn_x2a, Errstdn_x2a) ;

        numx2a_th = nx2a_th / (ubnx2a_th - lbnx2a_th);

        dx2a_th = pdf("normal",x2a,Pd_x2a,Errstdd_x2a);
        ubdx2a_th = cdf("normal",30, Pd_x2a, Errstdd_x2a);
        lbdx2a_th = cdf("normal",5, Pd_x2a, Errstdd_x2a) ;

        denx2a_th = dx2a_th / (ubdx2a_th - lbdx2a_th);

        *x2c;
        *ZERO INFLATED PORTION;
        if x2cind = 0 then do;
               numx2c_th=nx2cbin + (1-
nx2cbin)*(pdf("normal",logx2c,pn_x2c,errstdn_x2c)/(cdf("normal",3.5,pd_x2c,errstdd_x2c)-
cdf("normal",-7.5,pn_x2c,errstdn_x2c)));
               denx2c_th=dx2cbin + (1-
dx2cbin)*(pdf("normal",logx2c,pd_x2c,errstdd_x2c)/(cdf("normal",3.5,pd_x2c,errstdd_x2c)-
cdf("normal",-7.5,pd_x2c,errstdd_x2c)));
        end;
        *CONTINUOUS PORTION;
        if x2cind = 1 then do;
               numx2c_th=(1-
nx2cbin)*(pdf("normal",logx2c,pn_x2c,errstdn_x2c)/(cdf("normal",3.5,pd_x2c,errstdd_x2c)-
cdf("normal",-7.5,pn_x2c,errstdn_x2c)));
               denx2c_th=(1-
dx2cbin)*(pdf("normal",logx2c,pd_x2c,errstdd_x2c)/(cdf("normal",3.5,pd_x2c,errstdd_x2c)-
cdf("normal",-7.5,pd_x2c,errstdd_x2c)));
        end;

        *NORMAL HETEROSCEDASTIC: USER WRITTEN;
        *x2a;
        numx2ahu=pdf("normal",x2a,nx2amu,nx2asig);
        denx2ahu=pdf("normal",x2a,dx2amu,dx2asig);

        *x2c;
        *ZERO PORTION;
        if x2c = 0 then do;
               numx2chu = nx2cpi;
               denx2chu = dx2cpi;
        end;
        *CONTINUOUS PORTION;
        if x2c > 0 then do;
               numx2chu = (1-nx2cpi)*pdf("normal",logx2c,nx2cmu,nx2csig);
               denx2chu = (1-dx2cpi)*pdf("normal",logx2c,dx2cmu,dx2csig);
```

```
        end;

        *T DISTRIBUTED;
        numt1x2a = pdf("T",nx2a,1);
        numt1x2c = pdf("T",nx2c,1);
        dent1x2a = pdf("T",dx2a,1);
        dent1x2c = pdf("T",dx2c,1);

        numt3x2a = pdf("T",nx2a,3);
        numt3x2c = pdf("T",nx2c,3);
        dent3x2a = pdf("T",dx2a,3);
        dent3x2c = pdf("T",dx2c,3);

        numt5x2a = pdf("T",nx2a,5);
        numt5x2c = pdf("T",nx2c,5);
        dent5x2a = pdf("T",dx2a,5);
        dent5x2c = pdf("T",dx2c,5);

        *STABILIZED WEIGHTS;
        *NORMAL;
        swx2a=numx2a/denx2a;
        swx2c=numx2c/denx2c;

        *HETEROSCEDASTIC NORMAL;
        swx2ah=numx2ah/denx2ah;
        swx2ch=numx2ch/denx2ch;

        *TRUNCATED HETEROSCEDASTIC NORMAL;
        swx2a_th = numx2a_th/denx2a_th;
        swx2c_th = numx2c_th/denx2c_th;

        *USER WRITTEN HETERO NORMAL;
        swx2ahu = numx2ahu/denx2ahu;
        swx2chu = numx2chu/denx2chu;

        *GAMMA;
        swgx2a=numgx2a/dengx2a;
        swgx2c=numgx2c/dengx2c;

        *T;
        swt1x2a=numt1x2a/dent1x2a;
        swt1x2c=numt1x2c/dent1x2c;

        swt3x2a=numt3x2a/dent3x2a;
        swt3x2c=numt3x2c/dent3x2c;

        swt5x2a=numt5x2a/dent5x2a;
        swt5x2c=numt5x2c/dent5x2c;

        *QUANTILE BINNING APPROACH;
        *BECAUSE QUANTILES ARE USED, NUMERATOR PROBABILITIES ARE SIMPLY 1/#QUANTILES;
        swcatx2a10 = (1/10)/dencata10;
        swcatx2a15 = (1/15)/dencata15;
        swcatx2a20 = (1/20)/dencata20;

        swcatx2c10 = (1/10)/dencatc10;
        swcatx2c15 = (1/15)/dencatc15;
        swcatx2c20 = (1/20)/dencatc20;

        merg=1;
run;
*FIND TRUNCATION VALUES;
proc univariate data=b noprint;by h;var swx2a swx2c swx2ah swx2ch swx2ahu swx2chu swx2a_th
```

```
swx2c_th swgx2a swgx2c;
output out=ptile p1 =swx2a1 swx2c1 swx2ah1 swx2ch1 swx2ahu1 swx2chu1 swx2a_th1 swx2c_th1
swgx2a1 swgx2c1
                          p99=swx2a99 swx2c99 swx2ah99 swx2ch99 swx2ahu99 swx2chu99 swx2a_th99
swx2c_th99 swgx2a99 swgx2c99
                            p5 =swx2a5 swx2c5 swx2ah5 swx2ch5 swx2ahu5 swx2chu5 swx2a_th5
swx2c_th5 swgx2a5 swgx2c5
                            p95=swx2a95 swx2c95 swx2ah95 swx2ch95 swx2ahu95 swx2chu95
swx2a_th95 swx2c_th95 swgx2a95 swgx2c95
                            p10=swx2a10 swx2c10 swx2ah10 swx2ch10 swx2ahu10 swx2chu10
swx2a_th10 swx2c_th10 swgx2a10 swgx2c10
                            p90=swx2a90 swx2c90 swx2ah90 swx2ch90 swx2ahu90 swx2chu90
swx2a_th90 swx2c_th90 swgx2a90 swgx2c90;
run;
data ptile;set ptile;merg=1;
*TRUNCATE STABILIZED WEIGHTS FROM NORMAL AND GAMMA MODELS;
data cc;merge b ptile;by h merg;
*NORMAL X2A AND X2C;
swx2a199 = swx2a;
if swx2a < swx2a1 then swx2a199 = swx2a1;
if swx2a > swx2a99 then swx2a199 = swx2a99;

swx2c199 = swx2c;
if swx2c < swx2c1 then swx2c199 = swx2c1;
if swx2c > swx2c99 then swx2c199 = swx2c99;

swx2a595 = swx2a;
if swx2a < swx2a5 then swx2a595 = swx2a5;
if swx2a > swx2a95 then swx2a595 = swx2a95;

swx2c595 = swx2c;
if swx2c < swx2c5 then swx2c595 = swx2c5;
if swx2c > swx2c95 then swx2c595 = swx2c95;

swx2a1090 = swx2a;
if swx2a < swx2a10 then swx2a1090 = swx2a10;
if swx2a > swx2a90 then swx2a1090 = swx2a90;

swx2c1090 = swx2c;
if swx2c < swx2c10 then swx2c1090 = swx2c10;
if swx2c > swx2c90 then swx2c1090 = swx2c90;

*NORMAL HETEROSCEDASTIC X2A AND X2C;
swx2ah199 = swx2ah;
if swx2ah < swx2ah1 then swx2ah199 = swx2ah1;
if swx2ah > swx2ah99 then swx2ah199 = swx2ah99;

swx2ch199 = swx2ch;
if swx2ch < swx2ch1 then swx2ch199 = swx2ch1;
if swx2ch > swx2ch99 then swx2ch199 = swx2ch99;

swx2ah595 = swx2ah;
if swx2ah < swx2ah5 then swx2ah595 = swx2ah5;
if swx2ah > swx2ah95 then swx2ah595 = swx2ah95;

swx2ch595 = swx2ch;
if swx2ch < swx2ch5 then swx2ch595 = swx2ch5;
if swx2ch > swx2ch95 then swx2ch595 = swx2ch95;

swx2ah1090 = swx2ah;
if swx2ah < swx2ah10 then swx2ah1090 = swx2ah10;
if swx2ah > swx2ah90 then swx2ah1090 = swx2ah90;
```

```
swx2ch1090 = swx2ch;
if swx2ch < swx2ch10 then swx2ch1090 = swx2ch10;
if swx2ch > swx2ch90 then swx2ch1090 = swx2ch90;

*TRUNCATED NORMAL HETEROSCEDASTIC X2A AND X2C;
swx2a_th199 = swx2a_th;
if swx2a_th < swx2a_th1 then swx2a_th199 = swx2a_th1;
if swx2a_th > swx2a_th99 then swx2a_th199 = swx2a_th99;

swx2c_th199 = swx2c_th;
if swx2c_th < swx2c_th1 then swx2c_th199 = swx2c_th1;
if swx2c_th > swx2c_th99 then swx2c_th199 = swx2c_th99;

swx2a_th595 = swx2a_th;
if swx2a_th < swx2a_th5 then swx2a_th595 = swx2a_th5;
if swx2a_th > swx2a_th95 then swx2a_th595 = swx2a_th95;

swx2c_th595 = swx2c_th;
if swx2c_th < swx2c_th5 then swx2c_th595 = swx2c_th5;
if swx2c_th > swx2c_th95 then swx2c_th595 = swx2c_th95;

swx2a_th1090 = swx2a_th;
if swx2a_th < swx2a_th10 then swx2a_th1090 = swx2a_th10;
if swx2a_th > swx2a_th90 then swx2a_th1090 = swx2a_th90;

swx2c_th1090 = swx2c_th;
if swx2c_th < swx2c_th10 then swx2c_th1090 = swx2c_th10;
if swx2c_th > swx2c_th90 then swx2c_th1090 = swx2c_th90;

*USER WRITTEN NORMAL HETEROSCEDASTIC X2A AND X2C;
swx2ahu199 = swx2ahu;
if swx2ahu < swx2ahu1 thuen swx2ahu199 = swx2ahu1;
if swx2ahu > swx2ahu99 thuen swx2ahu199 = swx2ahu99;

swx2chu199 = swx2chu;
if swx2chu < swx2chu1 then swx2chu199 = swx2chu1;
if swx2chu > swx2chu99 then swx2chu199 = swx2chu99;

swx2ahu595 = swx2ahu;
if swx2ahu < swx2ahu5 then swx2ahu595 = swx2ahu5;
if swx2ahu > swx2ahu95 then swx2ahu595 = swx2ahu95;

swx2chu595 = swx2chu;
if swx2chu < swx2chu5 then swx2chu595 = swx2chu5;
if swx2chu > swx2chu95 then swx2chu595 = swx2chu95;

swx2ahu1090 = swx2ahu;
if swx2ahu < swx2ahu10 then swx2ahu1090 = swx2ahu10;
if swx2ahu > swx2ahu90 then swx2ahu1090 = swx2ahu90;

swx2chu1090 = swx2chu;
if swx2chu < swx2chu10 then swx2chu1090 = swx2chu10;
if swx2chu > swx2chu90 then swx2chu1090 = swx2chu90;

*GAMMA X2A AND X2C;

swgx2a199 = swgx2a;
if swgx2a < swgx2a1 then swgx2a199 = swgx2a1;
if swgx2a > swgx2a99 then swgx2a199 = swgx2a99;

swgx2a595 = swgx2a;
if swgx2a < swgx2a5 then swgx2a595 = swgx2a5;
if swgx2a > swgx2a95 then swgx2a595 = swgx2a95;
```

```
swgx2a1090 = swgx2a;
if swgx2a < swgx2a10 then swgx2a1090 = swgx2a10;
if swgx2a > swgx2a90 then swgx2a1090 = swgx2a90;

swgx2c199 = swgx2c;
if swgx2c < swgx2c1 then swgx2c199 = swgx2c1;
if swgx2c > swgx2c99 then swgx2c199 = swgx2c99;

swgx2c595 = swgx2c;
if swgx2c < swgx2c5 then swgx2c595 = swgx2c5;
if swgx2c > swgx2c95 then swgx2c595 = swgx2c95;

swgx2c1090 = swgx2c;
if swgx2c < swgx2c10 then swgx2c1090 = swgx2c10;
if swgx2c > swgx2c90 then swgx2c1090 = swgx2c90;

run;

data c;set cc;keep
       swx2a swx2c
       swx2ah swx2ch
       swx2ahu swx2chu
       swx2a_th swx2c_th

       swx2a199 swx2a595 swx2a1090
       swx2c199 swx2c595 swx2c1090
       swx2ah199 swx2ah595 swx2ah1090
       swx2ch199 swx2ch595 swx2ch1090
       swx2ahu199 swx2ahu595 swx2ahu1090
       swx2chu199 swx2chu595 swx2chu1090
       swx2a_th199 swx2a_th595 swx2a_th1090
       swx2c_th199 swx2c_th595 swx2c_th1090

       swgx2a swgx2c
       swgx2a199 swgx2a595 swgx2a1090
       swgx2c199 swgx2c595 swgx2c1090

       swt1x2a swt1x2c
       swt3x2a swt3x2c
       swt5x2a swt5x2c

       swcatx2a10 swcatx2c10
       swcatx2a15 swcatx2c15
       swcatx2a20 swcatx2c20

       x2a y2a x2c y2c h id ;run;
ods select all;
proc means data=c min max mean nmiss maxdec=2;
title "Distribution of Weights";
var
       swx2a swx2c
       swx2ah swx2ch
       swx2ahu swx2chu
       swx2a_th swx2c_th

       swx2a199 swx2a595 swx2a1090
       swx2c199 swx2c595 swx2c1090
       swx2ah199 swx2ah595 swx2ah1090
       swx2ch199 swx2ch595 swx2ch1090
       swx2ahu199 swx2ahu595 swx2ahu1090
       swx2chu199 swx2chu595 swx2chu1090
       swx2a_th199 swx2a_th595 swx2a_th1090
```

```
       swx2c_th199 swx2c_th595 swx2c_th1090

       swgx2a swgx2c
       swgx2a199 swgx2a595 swgx2a1090
       swgx2c199 swgx2c595 swgx2c1090

       swt1x2a swt1x2c
       swt3x2a swt3x2c
       swt5x2a swt5x2c

       swcatx2a10 swcatx2c10
       swcatx2a15 swcatx2c15
       swcatx2a20 swcatx2c20 ;
run;quit;run;
*FINAL MODELS;
ods select none;
title "x2a";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal";
model y2a = x2a / dist=bin maxiter=100;
weight swx2a;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta1(where=(parm = "x2a"));run;
data beta1;set beta1;length type $ 10;type = "N";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal (1,99)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2a199;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta2(where=(parm = "x2a"));run;
data beta2;set beta2;length type $ 10;type = "N(1,99)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal (5,95)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2a595;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta3(where=(parm = "x2a"));run;
data beta3;set beta3;length type $ 10;type = "N(5,95)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal (10,90)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2a1090;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4(where=(parm = "x2a"));run;
data beta4;set beta4;length type $ 10;type = "N(10,90)";

proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Cerda)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2ah;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a(where=(parm = "x2a"));run;
data beta4a;set beta4a;length type $ 10;type = "NHc";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Cerda)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2ah199;
```

```
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a1(where=(parm = "x2a"));run;
data beta4a1;set beta4a1;length type $ 10;type = "NHc(1,99)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Cerda)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2ah595;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a2(where=(parm = "x2a"));run;
data beta4a2;set beta4a2;length type $ 10;type = "NHc(5,95)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Cerda)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2ah1090;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a3(where=(parm = "x2a"));run;
data beta4a3;set beta4a3;length type $ 10;type = "NHc(10,90)";

proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Hernan)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2a_th;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a4(where=(parm = "x2a"));run;
data beta4a4;set beta4a4;length type $ 10;type = "NHh";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Hernan)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2a_th199;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a5(where=(parm = "x2a"));run;
data beta4a5;set beta4a5;length type $ 10;type = "NHh(1,99)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Hernan)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2a_th595;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a6(where=(parm = "x2a"));run;
data beta4a6;set beta4a6;length type $ 10;type = "NHh(5,95)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Hernan)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2a_th1090;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a7(where=(parm = "x2a"));run;
data beta4a7;set beta4a7;length type $ 10;type = "NHh(10,90)";

proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Naimi)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2ahu;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a8(where=(parm = "x2a"));run;
data beta4a8;set beta4a8;length type $ 10;type = "NHu";
proc genmod data=c desc;
by h;class id;
```

```
title2 "Standard Normal Heteroscedastic (Naimi)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2ahu199;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a9(where=(parm = "x2a"));run;
data beta4a9;set beta4a9;length type $ 10;type = "NHu(1,99)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Naimi)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2ahu595;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a10(where=(parm = "x2a"));run;
data beta4a10;set beta4a10;length type $ 10;type = "NHu(5,95)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Naimi)";
model y2a = x2a / dist=bin maxiter=100;
weight swx2ahu1090;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4a11(where=(parm = "x2a"));run;
data beta4a11;set beta4a11;length type $ 10;type = "NHu(10,90)";

proc genmod data=c desc;
by h;class id;
title2 "Gamma";
model y2a = x2a / dist=bin maxiter=100;
weight swgx2a;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4b(where=(parm = "x2a"));run;
data beta4b;set beta4b;length type $ 10;type = "G";
proc genmod data=c desc;
by h;class id;
title2 "Gamma";
model y2a = x2a / dist=bin maxiter=100;
weight swgx2a199;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4b1(where=(parm = "x2a"));run;
data beta4b1;set beta4b1;length type $ 10;type = "G(1,99)";
proc genmod data=c desc;
by h;class id;
title2 "Gamma";
model y2a = x2a / dist=bin maxiter=100;
weight swgx2a595;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4b2(where=(parm = "x2a"));run;
data beta4b2;set beta4b2;length type $ 10;type = "G(5,95)";
proc genmod data=c desc;
by h;class id;
title2 "Gamma";
model y2a = x2a / dist=bin maxiter=100;
weight swgx2a1090;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta4b3(where=(parm = "x2a"));run;
data beta4b3;set beta4b3;length type $ 10;type = "G(10,90)";

proc genmod data=c desc;
by h;class id;
title2 "t-Distribution (1df)";
model y2a = x2a / dist=bin maxiter=100;
weight swt1x2a;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta5(where=(parm = "x2a"));run;
```

```
data beta5;set beta5;length type $ 10;type = "t1";
proc genmod data=c desc;
by h;class id;
title2 "t-Distribution (3df)";
model y2a = x2a / dist=bin maxiter=100;
weight swt3x2a;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta6(where=(parm = "x2a"));run;
data beta6;set beta6;length type $ 10;type = "t3";
proc genmod data=c desc;
by h;class id;
title2 "t-Distribution (5df)";
model y2a = x2a / dist=bin maxiter=100;
weight swt5x2a;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta7(where=(parm = "x2a"));run;
data beta7;set beta7;length type $ 10;type = "t5";

proc genmod data=c desc;
by h;class id;
title2 "Quantile Binning (10)";
model y2a = x2a / dist=bin maxiter=100;
weight swcatx2a10;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta8(where=(parm = "x2a"));run;
data beta8;set beta8;length type $ 10;type = "QB10";
proc genmod data=c desc;
by h;class id;
title2 "Quantile Binning (15)";
model y2a = x2a / dist=bin maxiter=100;
weight swcatx2a15;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta9(where=(parm = "x2a"));run;
data beta9;set beta9;length type $ 10;type = "QB15";
proc genmod data=c desc;
by h;class id;
title2 "Quantile Binning (20)";
model y2a = x2a / dist=bin maxiter=100;
weight swcatx2a20;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta10(where=(parm = "x2a"));run;
data beta10;set beta10;length type $ 10;type = "QB20";
run;

title "x2c";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal";
model y2c = x2c / dist=bin maxiter=100;
weight swx2c;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta11(where=(parm = "x2c"));run;
data beta11;set beta11;length type $ 10;type = "N";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal (1,99)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2c199;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta12(where=(parm = "x2c"));run;
data beta12;set beta12;length type $ 10;type = "N(1,99)";
proc genmod data=c desc;
by h;class id;
```

```
title2 "Standard Normal (5,95)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2c595;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta13(where=(parm = "x2c"));run;
data beta13;set beta13;length type $ 10;type = "N(5,95)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal (10,90)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2c1090;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14(where=(parm = "x2c"));run;
data beta14;set beta14;length type $ 10;type = "N(10,90)";

proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Cerda)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2ch;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a(where=(parm = "x2c"));run;
data beta14a;set beta14a;length type $ 10;type = "NHc";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Cerda)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2ch199;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a1(where=(parm = "x2c"));run;
data beta14a1;set beta14a1;length type $ 10;type = "NHc(1,99)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Cerda)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2ch595;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a2(where=(parm = "x2c"));run;
data beta14a2;set beta14a2;length type $ 10;type = "NHc(5,95)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Cerda)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2ch1090;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a3(where=(parm = "x2c"));run;
data beta14a3;set beta14a3;length type $ 10;type = "NHc(10,90)";

proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Hernan)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2c_th;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a4(where=(parm = "x2c"));run;
data beta14a4;set beta14a4;length type $ 10;type = "NHh";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Hernan)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2c_th199;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a5(where=(parm = "x2c"));run;
```

```
data beta14a5;set beta14a5;length type $ 10;type = "NHh(1,99)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Hernan)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2c_th595;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a6(where=(parm = "x2c"));run;
data beta14a6;set beta14a6;length type $ 10;type = "NHh(5,95)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Hernan)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2c_th1090;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a7(where=(parm = "x2c"));run;
data beta14a7;set beta14a7;length type $ 10;type = "NHh(10,90)";

proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Naimi)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2chu;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a8(where=(parm = "x2c"));run;
data beta14a8;set beta14a8;length type $ 10;type = "NHu";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Naimi)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2chu199;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a9(where=(parm = "x2c"));run;
data beta14a9;set beta14a9;length type $ 10;type = "NHu(1,99)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Naimi)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2chu595;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a10(where=(parm = "x2c"));run;
data beta14a10;set beta14a10;length type $ 10;type = "NHu(5,95)";
proc genmod data=c desc;
by h;class id;
title2 "Standard Normal Heteroscedastic (Naimi)";
model y2c = x2c / dist=bin maxiter=100;
weight swx2chu1090;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14a11(where=(parm = "x2c"));run;
data beta14a11;set beta14a11;length type $ 10;type = "NHu(10,90)";

proc genmod data=c desc;
by h;class id;
title2 "Gamma";
model y2c = x2c / dist=bin maxiter=100;
weight swgx2c;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14b(where=(parm = "x2c"));run;
data beta14b;set beta14b;length type $ 10;type = "G";
proc genmod data=c desc;
by h;class id;
title2 "Gamma(1,99)";
model y2c = x2c / dist=bin maxiter=100;
```

```
weight swgx2c199;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14c(where=(parm = "x2c"));run;
data beta14c;set beta14c;length type $ 10;type = "G(1,99)";
proc genmod data=c desc;
by h;class id;
title2 "Gamma(5,95)";
model y2c = x2c / dist=bin maxiter=100;
weight swgx2c595;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14d(where=(parm = "x2c"));run;
data beta14d;set beta14d;length type $ 10;type = "G(5,95)";
proc genmod data=c desc;
by h;class id;
title2 "Gamma(10,90)";
model y2c = x2c / dist=bin maxiter=100;
weight swgx2c1090;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta14e(where=(parm = "x2c"));run;
data beta14e;set beta14e;length type $ 10;type = "G(10,90)";

proc genmod data=c desc;
by h;class id;
title2 "t-Distribution (1df)";
model y2c = x2c / dist=bin maxiter=100;
weight swt1x2c;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta15(where=(parm = "x2c"));run;
data beta15;set beta15;length type $ 10;type = "t1";
proc genmod data=c desc;
by h;class id;
title2 "t-Distribution (3df)";
model y2c = x2c / dist=bin maxiter=100;
weight swt3x2c;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta16(where=(parm = "x2c"));run;
data beta16;set beta16;length type $ 10;type = "t3";
proc genmod data=c desc;
by h;class id;
title2 "t-Distribution (5df)";
model y2c = x2c / dist=bin maxiter=100;
weight swt5x2c;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta17(where=(parm = "x2c"));run;
data beta17;set beta17;length type $ 10;type = "t5";
proc genmod data=c desc;
by h;class id;
title2 "Quantile Binning (10)";
model y2c = x2c / dist=bin maxiter=100;
weight swcatx2c10;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta18(where=(parm = "x2c"));run;
data beta18;set beta18;length type $ 10;type = "QB10";
proc genmod data=c desc;
by h;class id;
title2 "Quantile Binning (15)";
model y2c = x2c / dist=bin maxiter=100;
weight swcatx2c15;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta19(where=(parm = "x2c"));run;
data beta19;set beta19;length type $ 10;type = "QB15";
proc genmod data=c desc;
by h;class id;
```

```
title2 "Quantile Binning (20)";
model y2c = x2c / dist=bin maxiter=100;
weight swcatx2c20;
repeated subject=id / type=ind;
ods output GEEEmpPEst=beta20(where=(parm = "x2c"));run;
data beta20;set beta20;length type $ 10;type = "QB20";
run;
title;title2;
ods select all;
*COMPUTE BIAS AND MSE;
data beta;
       set beta1-beta10 beta4a beta4a1-beta4a11 beta4b beta4b1-beta4b3 beta11-beta20 beta14a
beta14a1-beta14a11 beta14b beta14c beta14d beta14e;
       merg=1;
data bias;
       merge meancheck beta;
       by merg;
       if parm = "x2a" then do; bias = estimate - truex2a;  mse = (estimate -
truex2a)**2;end;
       if parm = "x2c" then do; bias = estimate - truex2c;  mse = (estimate -
truex2c)**2;end;

       if type="QB20" then typen=1;
       if type="QB15" then typen=2;
       if type="QB10" then typen=3;

       if type="G(10,90)" then typen=4;
       if type="G(5,95)" then typen=5;
       if type="G(1,99)" then typen=6;
       if type="G" then typen=7;

       if type="t5" then typen=8;
       if type="t3" then typen=9;
       if type="t1" then typen=10;

       if type="NHu(10,90)" then typen=11;
       if type="NHu(5,95)" then typen=12;
       if type="NHu(1,99)" then typen=13;
       if type="NHu" then typen=14;

       if type="NHh(10,90)" then typen=15;
       if type="NHh(5,95)" then typen=16;
       if type="NHh(1,99)" then typen=17;
       if type="NHh" then typen=18;

       if type="NHc(10,90)" then typen=19;
       if type="NHc(5,95)" then typen=20;
       if type="NHc(1,99)" then typen=21;
       if type="NHc" then typen=22;

       if type="N(10,90)" then typen=23;
       if type="N(5,95)" then typen=24;
       if type="N(1,99)" then typen=25;
       if type="N" then typen=26;

run;
proc format;
       value tpn 1 = "QB20"
                 2 = "QB15"
                 3 = "QB10"

                 4 = "G(10,90)"
                 5 = "G(5,95)"
```

```
                                6 = "G(1,99)"
                                7 = "G"

                                8 = "t5"
                                9 = "t3"
                                10 = "t1"

                                11 = "NHu(10,90)"
                                12 = "NHu(5,95)"
                                13 = "NHu(1,99)"
                                14 = "NHu"

                                15 = "NHh(10,90)"
                                16 = "NHh(5,95)"
                                17 = "NHh(1,99)"
                                18 = "NHh"

                                19 = "NHc(10,90)"
                                20 = "NHc(5,95)"
                                21 = "NHc(1,99)"
                                22 = "NHc"

                                23 = "N(10,90)"
                                24 = "N(5,95)"
                                25 = "N(1,99)"
                                26 = "N"
;
*RESULTS;
proc sort data=bias;by parm typen;run;
proc means data=bias mean maxdec=4;
class parm typen;
var bias mse;format typen tpn.;run;
proc univariate data=bias noprint;
class parm type;
var bias;histogram;
run;quit;run;
*END;
```