eAppendix 1. Structural nested model SAS programs.

This appendix contains three SNM SAS programs: the log-linear SNM in eAppendix 1(a), the logistic SNM in eAppendix 1(b), and the linear SNM in eAppendix 1(c).

eAppendix 1(a). Log-linear SNM

```
*****************************************************************************
*******************************************************************
THIS IS THE LOG-LINEAR SNM
*This SAS program is based on the methodology developed and published in:
BA Brumback, Z He, M Prasad, MC Freeman, R Rheingans. Using structural-nested
models to estimate the effect of cluster-level adherence on individual-level
outcomes with a three-armed cluster-randomized trial. Statistics in medicine.
2014, 33 (9), 1490-1502

The methodology has been further discussed in:
S Helian, BA Brumback, MC Freeman, R Rheingans. Structural Nested Models for
Cluster-Randomized Trials. in Statistical Causal Inference and Their
Applications in Public Health Research, eds. H He, P Wu, DG Chen. Springer,
in press.

We acknowledge support from the National Science Foundation through grant NSF
SES-1115618 for this work.

The SAS program requires several variables:

Y – outcome variable
X – adherence variable denoting three adherence levels
X1 – indicator variable for X = 1
X2 – indicator variable for X = 2
W – previously created weight variable, which accounts for confounding by
individual-level baseline covariates
Z – randomization variable denoting three randomization arms
Z1 – indicator variable for Z = 1
Z2 – indicator variable for Z = 2
Stratum – variable denoting the strata
PSU – variable denoting the primary sampling units (e.g. schools)
Id – variable denoting a unique id for each participant in the study

***** The SAS program requires minor changes for:
studies with different numbers of PSU’s (it is now set up for 86 PSUs), or
studies with different numbers of strata (it is now set up for 2 strata).
The areas that would need to be changed are marked with comments.

***** The SAS program requires significant changes for:
a two armed trial (it is now set up for a three armed trial) which produces a
single adherence estimate (it now produces two estimates).
*****************************************************************************
*******************************************************************;

libname sim 'H:\IV analysis\loglinear';

*Rename the dataset sim.gooddata;
```

```
data sim.gooddata;
set weights;
run;

proc surveyfreq data=sim.gooddata;
tables Z*X / row cl;
weight W;
run;

*count the number of PSUs overall and the number of PSUs in each strata;
proc sort data=sim.gooddata;
by stratum psu;
run;
data PSU_level;
set sim.gooddata;
by stratum psu;
if first.psu;
keep stratum psu Z X ;
run;

Proc freq data = PSU_level;
tables Z X Z*X stratum/list missing;
run;

%macro simulate;

data sim.EYzeromat;
set _null_;
run;

data sim.etamat;
set _null_;
run;

data sim.squigiter;
set _null_;
run;

data sim.toliter;
set _null_;
run;

data sim.baddata;
set _null_;
run;

proc sort data=sim.gooddata;
by stratum psu;
run;

data psustrat;
set sim.gooddata;
by stratum psu;
if first.psu;
keep stratum psu Z X;
run;
```

```
proc freq data=psustrat;
tables Z*X;
run;

data sim.psustrat;
set psustrat;
keep stratum psu;
run;

proc surveymeans data=sim.gooddata varmethod=jk(outweights=sim.jkweights);
weight W;
strata stratum;
cluster psu;
run;

%do i=0 %to 86; /*Number of PSUs varies by study and so this number may need
to change.*/

%if &i=0 %then %do;
data sim.data0;
set sim.gooddata;
run;
%end;

%else %do;

data sim.data0;
set sim.jkweights;
W=RepWt_&i;
run;

%end;


proc genmod data=sim.data0;
model Y = X1 X2 Z1 Z2 X1*Z1 X1*Z2 X2*Z1 X2*Z2 / dist=poisson link=log;
weight W;
output out=sim.xbeta xbeta=linpred;
run;

data sim.xbeta;
set sim.xbeta;
keep id linpred;
run;

proc sort data=sim.xbeta;
by id;
run;

proc sort data=sim.data0;
by id;
run;

data sim.data0;
merge sim.data0 sim.xbeta;
by id;
run;
```

```
data sim.iv;
set sim.data0;
squig0=0.3320108649; /*Program runs faster with a better initial value.*/
squig1=-0.917864789; /*Program runs faster with a better initial value.*/
squig2=-0.333768258; /*Program runs faster with a better initial value.*/
run;

%macro iv;

%let k= 0;
%do %until(&tol<0.001);
%let k= &k+1;

data sim.iv;
set sim.iv;
lp=linpred-X1*squig1 - X2*squig2;
explp=exp(lp);
Ystar = explp*(1+ X1*squig1 + X2*squig2);
Xstar1 = X1*explp;
Xstar2 = X2*explp;
U1=W*(explp-squig0);
U2=W*Z1*(explp-squig0);
U3=W*Z2*(explp-squig0);
U1star=W*(Ystar-Xstar1*squig1-Xstar2*squig2-squig0);
U2star=W*Z1*(Ystar-Xstar1*squig1-Xstar2*squig2-squig0);
U3star=W*Z2*(Ystar-Xstar1*squig1-Xstar2*squig2-squig0);
run;

proc syslin data=sim.iv 2sls;
endogenous Xstar1 Xstar2;
instruments Z1 Z2;
model Ystar =Xstar1 Xstar2;
weight W;
ods output ParameterEstimates=sim.squig;
run;

data sim.check1;
set sim.squig;
run;

data sim.squig;
set sim.squig;
keep Estimate;
run;

proc transpose data=sim.squig out=sim.squig;
run;

data sim.check2;
set sim.squig;
run;

data sim.iv;
set sim.iv;
drop squig0 squig1 squig2;
one=1;
```

```
run;

data sim.squig;
set sim.squig;
squig0=COL1;
squig1=COL2;
squig2=COL3;
one=1;
keep one squig0 squig1 squig2;
run;

data sim.squigiter;
set sim.squigiter sim.squig;
run;

data sim.iv;
merge sim.iv sim.squig;
by one;
run;

data sim.tol;
set sim.iv;
by one;
if first.one then do;
sumU1=0;
sumU2=0;
sumU3=0;
sumU1star=0;
sumU2star=0;
sumU3star=0;
end;
sumU1+U1;
sumU2+U2;
sumU3+U3;
sumU1star+U1star;
sumU2star+U2star;
sumU3star+U3star;
if last.one then output;
keep sumU1 sumU2 sumU3 sumU1star sumU2star sumU3star;
run;

data sim.tol;
set sim.tol;
tol = sumU1*sumU1 + sumU2*sumU2 + sumU3*sumU3;
call symput('tol',left(round(tol,.0001)));
keep tol sumU1 sumU2 sumU3 sumU1star sumU2star sumU3star;
run;

data sim.toliter;
set sim.toliter sim.tol;
run;

%if &k = 50 %then %do;
%let tol=0;
data sim.baddata;
set sim.baddata sim.data0;
run;
```

```
%end;

%end;

%mend;

%let tol=1;

%iv;

data sim.iv;
set sim.iv;
Yzero = explp;
run;

proc reg data=sim.iv;
model Yzero = X1 X2;
weight W;
ods output ParameterEstimates=sim.EYzero;
run;

proc reg data=sim.iv;
model Y = X1 X2;
weight W;
ods output ParameterEstimates=sim.eta;
run;

data sim.EYzero;
set sim.EYzero;
keep Estimate;
run;

data sim.eta;
set sim.eta;
keep Estimate;
run;

proc transpose data=sim.EYzero out=sim.EYzero;
run;

data sim.EYzero;
set sim.EYzero;
EYzero0=COL1;
EYzero1=COL2;
EYzero2=COL3;
EYzero1=EYzero0+EYzero1;
EYzero2=EYzero0+EYzero2;
keep EYzero1 EYzero2;
run;

proc transpose data=sim.eta out=sim.eta;
run;

data sim.eta;
set sim.eta;
eta0=COL1;
eta1=COL2;
```

```
eta2=COL3;
eta1=eta0+eta1;
eta2=eta0+eta2;
keep eta1 eta2;
run;

data sim.EYzeromat;
set sim.EYzeromat sim.EYzero;
run;

data sim.etamat;
set sim.etamat sim.eta;
run;

%end;

%mend;

%simulate;



*jackknife variance for log-linear SNM for the RD;
data eta;
set sim.etamat;
if _n_ > 1;
run;

data yzero;
set sim.EYzeromat;
if _n_ > 1;
run;

proc means data=eta;
run;

proc means data=yzero;
run;

data eta0;
set sim.etamat;
if _n_=1;
eta10=eta1;
eta20=eta2;
one=1;
keep eta10 eta20 one;
run;

data yzero0;
set sim.EYzeromat;
if _n_=1;
eyzero10=eyzero1;
eyzero20=eyzero2;
one=1;
keep eyzero10 eyzero20 one;
run;
```

```
proc freq data = PSU_level;
tables stratum /list missing;
run;

data sim.all;
merge eta yzero sim.psustrat;
rd1 = (eta1-eyzero1);
rd2 = (eta2-eyzero2);
if stratum=1 then mult=(48-1)/48; /*(Number of PSUs within stratum h minus
1)/(Number of PSUs within stratum h)*/
if stratum=2 then mult=(38-1)/38; /*(Number of PSUs within stratum h minus
1)/(Number of PSUs within stratum h)*/
/*if stratum=3 then mult=(number_in_stratum-1)/number_in_stratum;*/ /*And so
on, if there are more strata*/
one=1;
run;

data sim.all;
merge sim.all eta0 yzero0;
by one;
rd10=(eta10-eyzero10);
rd20=(eta20-eyzero20);
var1=mult*(rd1-rd10)**2;
var2=mult*(rd2-rd20)**2;
run;

proc univariate data=sim.all;
var var1 var2;
output out=sim.variances sum=var1 var2;
run;

data sim.variances;
merge eta0 yzero0 sim.variances;
rd10=eta10-eyzero10;
rd20=eta20-eyzero20;
sd10=sqrt(var1);
sd20=sqrt(var2);
lcl1=rd10-1.96*sd10;
ucl1=rd10+1.96*sd10;
lcl2=rd20-1.96*sd20;
ucl2=rd20+1.96*sd20;
keep eta10 eta20 eyzero10 eyzero20 rd10 lcl1 ucl1 rd20 lcl2 ucl2;
run;

proc print data=sim.variances;run;




*jackknife variance for log-linear SNM for the RR;
data eta;
set sim.etamat;
if _n_ > 1;
run;

data yzero;
```

```
set sim.EYzeromat;
if _n_ > 1;
run;

proc means data=eta;
run;

proc means data=yzero;
run;

data eta0;
set sim.etamat;
if _n_=1;
eta10=eta1;
eta20=eta2;
one=1;
keep eta10 eta20 one;
run;

data yzero0;
set sim.EYzeromat;
if _n_=1;
eyzero10=eyzero1;
eyzero20=eyzero2;
one=1;
keep eyzero10 eyzero20 one;
run;


proc freq data = PSU_level;
tables stratum /list missing;
run;

data sim.all;
merge eta yzero sim.psustrat;
lrr1 = log(eta1/eyzero1);
lrr2 = log(eta2/eyzero2);
if stratum=1 then mult=(48-1)/48; /*(Number of PSUs within stratum h minus
1)/(Number of PSUs within stratum h)*/
if stratum=2 then mult=(38-1)/38; /*(Number of PSUs within stratum h minus
1)/(Number of PSUs within stratum h)*/
/*if stratum=3 then mult=(number_in_stratum-1)/number_in_stratum;*/ /*And so
on, if there are more strata*/
one=1;
run;

data sim.all;
merge sim.all eta0 yzero0;
by one;
lrr10=log(eta10/eyzero10);
lrr20=log(eta20/eyzero20);
var1=mult*(lrr1-lrr10)**2;
var2=mult*(lrr2-lrr20)**2;
run;

proc univariate data=sim.all;
var var1 var2;
```

```
output out=sim.variances sum=var1 var2;
run;

data sim.variances;
merge eta0 yzero0 sim.variances;
rr10=eta10/eyzero10;
rr20=eta20/eyzero20;
lrr10=log(rr10);
lrr20=log(rr20);
sd10=sqrt(var1);
sd20=sqrt(var2);
llcl1=lrr10-1.96*sd10;
lucl1=lrr10+1.96*sd10;
llcl2=lrr20-1.96*sd20;
lucl2=lrr20+1.96*sd20;
lcl1=exp(llcl1);
ucl1=exp(lucl1);
lcl2=exp(llcl2);
ucl2=exp(lucl2);
keep eta10 eta20 eyzero10 eyzero20 rr10 lcl1 ucl1 rr20 lcl2 ucl2;
run;

proc print data=sim.variances;run;
```

eAppendix 1(b). Logistic SNM

```
****************************************************************************
******************************************************************
THIS IS THE LOGISTIC SNM
*This SAS program is based on the methodology developed and published in:
BA Brumback, Z He, M Prasad, MC Freeman, R Rheingans. Using structural-nested
models to estimate the effect of cluster-level adherence on individual-level
outcomes with a three-armed cluster-randomized trial. Statistics in medicine.
2014, 33 (9), 1490-1502

The methodology has been further discussed in:
S Helian, BA Brumback, MC Freeman, R Rheingans. Structural Nested Models for
Cluster-Randomized Trials. in Statistical Causal Inference and Their
Applications in Public Health Research, eds. H He, P Wu, DG Chen. Springer,
in press.

We acknowledge support from the National Science Foundation through grant NSF
SES-1115618 for this work.

The SAS program requires several variables:

Y – outcome variable
X – adherence variable denoting three adherence levels
X1 – indicator variable for X = 1
X2 – indicator variable for X = 2
W – previously created weight variable, which accounts for confounding by
individual-level baseline covariates
Z – randomization variable denoting three randomization arms
Z1 – indicator variable for Z = 1
Z2 – indicator variable for Z = 2
Stratum – variable denoting the strata
PSU – variable denoting the primary sampling units (e.g. schools)
Id – variable denoting a unique id for each participant in the study

***** The SAS program requires minor changes for:
studies with different numbers of PSU’s (it is now set up for 86 PSUs), or
studies with different numbers of strata (it is now set up for 2 strata).
The areas that would need to be changed are marked with comments.

***** The SAS program requires significant changes for:
a two armed trial (it is now set up for a three armed trial) which produces a
single adherence estimate (it now produces two estimates).
The areas that would need to be changed are marked with comments.
****************************************************************************
******************************************************************;

libname sim 'H:\IV analysis\logistic';

*Rename the dataset sim.gooddata;
data sim.gooddata;
set weights;
run;

proc surveyfreq data=sim.gooddata;
```

```
tables Z*X / row cl;
weight W;
run;

*count the number of PSUs overall and the number of PSUs in each strata;
proc sort data=sim.gooddata;
by stratum psu;
run;
data PSU_level;
set sim.gooddata;
by stratum psu;
if first.psu;
keep stratum psu Z X ;
run;

Proc freq data = PSU_level;
tables Z X Z*X stratum/list missing;
run;

%macro simulate;

data sim.EYzeromat;
set _null_;
run;

data sim.etamat;
set _null_;
run;

data sim.squigiter;
set _null_;
run;

data sim.toliter;
set _null_;
run;

data sim.baddata;
set _null_;
run;

proc sort data=sim.gooddata;
by stratum psu;
run;

data psustrat;
set sim.gooddata;
by stratum psu;
if first.psu;
keep stratum psu Z X;
run;

proc freq data=psustrat;
tables Z*X;
run;

data sim.psustrat;
```

```
set psustrat;
keep stratum psu;
run;

proc surveymeans data=sim.gooddata varmethod=jk(outweights=sim.jkweights);
weight W;
strata stratum;
cluster psu;
run;

%do i=0 %to 86; /*Number of PSUs varies by study and so this number may need
to change.*/

%if &i=0 %then %do;
data sim.data0;
set sim.gooddata;
run;
%end;

%else %do;

data sim.data0;
set sim.jkweights;
W=RepWt_&i;
run;

%end;

proc genmod data=sim.data0 descending;
model Y = X1 X2 Z1 Z2 X1*Z1 X1*Z2 X2*Z1 X2*Z2 / dist=bin link=logit; /*If I
make a macro it would need to change the dist the link and to add
descending*/
weight W;
output out=sim.xbeta xbeta=linpred;
run;

data sim.xbeta;
set sim.xbeta;
keep id linpred;
run;

proc sort data=sim.xbeta;
by id;
run;

proc sort data=sim.data0;
by id;
run;

data sim.data0;
merge sim.data0 sim.xbeta;
by id;
run;

data sim.iv;
set sim.data0;
squig0=0.3318753373;  /*Program runs faster with a better initial value.*/
```

```
squig1=-1.347449395;  /*Program runs faster with a better initial value.*/
squig2=-0.474860432;  /*Program runs faster with a better initial value.*/
run;

%macro iv;

%let k= 0;
%do %until(&tol<0.001);
%let k= &k+1;

data sim.iv;
set sim.iv;
lp=linpred-X1*squig1 - X2*squig2;
expitlp=exp(lp)/(1+exp(lp));
Ystar = expitlp + (X1*squig1 + X2*squig2)*expitlp*(1-expitlp);
Xstar1 = X1*expitlp*(1-expitlp);
Xstar2 = X2*expitlp*(1-expitlp);
U1=W*(expitlp-squig0);
U2=W*Z1*(expitlp-squig0);
U3=W*Z2*(expitlp-squig0);
U1star=W*(Ystar-Xstar1*squig1-Xstar2*squig2-squig0);
U2star=W*Z1*(Ystar-Xstar1*squig1-Xstar2*squig2-squig0);
U3star=W*Z2*(Ystar-Xstar1*squig1-Xstar2*squig2-squig0);
run;

proc syslin data=sim.iv 2sls;
endogenous Xstar1 Xstar2;
instruments Z1 Z2;
model Ystar =Xstar1 Xstar2;
weight W;
ods output ParameterEstimates=sim.squig;
run;

data sim.check1;
set sim.squig;
run;

data sim.squig;
set sim.squig;
keep Estimate;
run;

proc transpose data=sim.squig out=sim.squig;
run;

data sim.check2;
set sim.squig;
run;

data sim.iv;
set sim.iv;
drop squig0 squig1 squig2;
one=1;
run;

data sim.squig;
set sim.squig;
```

```
squig0=COL1;
squig1=COL2;
squig2=COL3;
one=1;
keep one squig0 squig1 squig2;
run;

data sim.squigiter;
set sim.squigiter sim.squig;
run;

data sim.iv;
merge sim.iv sim.squig;
by one;
run;

data sim.tol;
set sim.iv;
by one;
if first.one then do;
sumU1=0;
sumU2=0;
sumU3=0;
sumU1star=0;
sumU2star=0;
sumU3star=0;
end;
sumU1+U1;
sumU2+U2;
sumU3+U3;
sumU1star+U1star;
sumU2star+U2star;
sumU3star+U3star;
if last.one then output;
keep sumU1 sumU2 sumU3 sumU1star sumU2star sumU3star;
run;

data sim.tol;
set sim.tol;
tol = sumU1*sumU1 + sumU2*sumU2 + sumU3*sumU3;
call symput('tol',left(round(tol,.0001)));
keep tol sumU1 sumU2 sumU3 sumU1star sumU2star sumU3star;
run;

data sim.toliter;
set sim.toliter sim.tol;
run;

%if &k = 50 %then %do;
%let tol=0;
data sim.baddata;
set sim.baddata sim.data0;
run;
%end;

%end;
```

```
%mend;

%let tol=1;

%iv;

data sim.iv;
set sim.iv;
Yzero = expitlp;
run;

proc reg data=sim.iv;
model Yzero = X1 X2;
weight W;
ods output ParameterEstimates=sim.EYzero;
run;

proc reg data=sim.iv;
model Y = X1 X2;
weight W;
ods output ParameterEstimates=sim.eta;
run;

data sim.EYzero;
set sim.EYzero;
keep Estimate;
run;

data sim.eta;
set sim.eta;
keep Estimate;
run;

proc transpose data=sim.EYzero out=sim.EYzero;
run;

data sim.EYzero;
set sim.EYzero;
EYzero0=COL1;
EYzero1=COL2;
EYzero2=COL3;
EYzero1=EYzero0+EYzero1;
EYzero2=EYzero0+EYzero2;
keep EYzero1 EYzero2;
run;

proc transpose data=sim.eta out=sim.eta;
run;

data sim.eta;
set sim.eta;
eta0=COL1;
eta1=COL2;
eta2=COL3;
eta1=eta0+eta1;
eta2=eta0+eta2;
keep eta1 eta2;
```

```
run;

data sim.EYzeromat;
set sim.EYzeromat sim.EYzero;
run;

data sim.etamat;
set sim.etamat sim.eta;
run;

%end;

%mend;

%simulate;





*jackknife variance for logistic SNM for the RD;
data eta;
set sim.etamat;
if _n_ > 1;
run;

data yzero;
set sim.EYzeromat;
if _n_ > 1;
run;

proc means data=eta;
run;

proc means data=yzero;
run;

data eta0;
set sim.etamat;
if _n_=1;
eta10=eta1;
eta20=eta2;
one=1;
keep eta10 eta20 one;
run;

data yzero0;
set sim.EYzeromat;
if _n_=1;
eyzero10=eyzero1;
eyzero20=eyzero2;
one=1;
keep eyzero10 eyzero20 one;
run;


proc freq data = PSU_level;
```

```
tables stratum /list missing;
run;

data sim.all;
merge eta yzero sim.psustrat;
rd1 = (eta1-eyzero1);
rd2 = (eta2-eyzero2);
if stratum=1 then mult=(48-1)/48; /*(Number of PSUs within stratum h minus
1)/(Number of PSUs within stratum h)*/
if stratum=2 then mult=(38-1)/38; /*(Number of PSUs within stratum h minus
1)/(Number of PSUs within stratum h)*/
/*if stratum=3 then mult=(number_in_stratum-1)/number_in_stratum;*/ /*And so
on, if there are more strata*/
one=1;
run;

data sim.all;
merge sim.all eta0 yzero0;
by one;
rd10=(eta10-eyzero10);
rd20=(eta20-eyzero20);
var1=mult*(rd1-rd10)**2;
var2=mult*(rd2-rd20)**2;
run;

proc univariate data=sim.all;
var var1 var2;
output out=sim.variances sum=var1 var2;
run;

data sim.variances;
merge eta0 yzero0 sim.variances;
rd10=eta10-eyzero10;
rd20=eta20-eyzero20;
sd10=sqrt(var1);
sd20=sqrt(var2);
lcl1=rd10-1.96*sd10;
ucl1=rd10+1.96*sd10;
lcl2=rd20-1.96*sd20;
ucl2=rd20+1.96*sd20;
keep eta10 eta20 eyzero10 eyzero20 rd10 lcl1 ucl1 rd20 lcl2 ucl2;
run;

proc print data=sim.variances;run;



*jackknife variance for logistic SNM for the RR;
data eta;
set sim.etamat;
if _n_ > 1;
run;

data yzero;
set sim.EYzeromat;
if _n_ > 1;
run;
```

```
proc means data=eta;
run;

proc means data=yzero;
run;

data eta0;
set sim.etamat;
if _n_=1;
eta10=eta1;
eta20=eta2;
one=1;
keep eta10 eta20 one;
run;

data yzero0;
set sim.EYzeromat;
if _n_=1;
eyzero10=eyzero1;
eyzero20=eyzero2;
one=1;
keep eyzero10 eyzero20 one;
run;


proc freq data = PSU_level;
tables stratum /list missing;
run;

data sim.all;
merge eta yzero sim.psustrat;
lrr1 = log(eta1/eyzero1);
lrr2 = log(eta2/eyzero2);
if stratum=1 then mult=(48-1)/48; /*(Number of PSUs within stratum h minus
1)/(Number of PSUs within stratum h)*/
if stratum=2 then mult=(38-1)/38; /*(Number of PSUs within stratum h minus
1)/(Number of PSUs within stratum h)*/
/*if stratum=3 then mult=(number_in_stratum-1)/number_in_stratum;*/ /*And so
on, if there are more strata*/
one=1;
run;

data sim.all;
merge sim.all eta0 yzero0;
by one;
lrr10=log(eta10/eyzero10);
lrr20=log(eta20/eyzero20);
var1=mult*(lrr1-lrr10)**2;
var2=mult*(lrr2-lrr20)**2;
run;

proc univariate data=sim.all;
var var1 var2;
output out=sim.variances sum=var1 var2;
run;
```

```
data sim.variances;
merge eta0 yzero0 sim.variances;
rr10=eta10/eyzero10;
rr20=eta20/eyzero20;
lrr10=log(rr10);
lrr20=log(rr20);
sd10=sqrt(var1);
sd20=sqrt(var2);
llcl1=lrr10-1.96*sd10;
lucl1=lrr10+1.96*sd10;
llcl2=lrr20-1.96*sd20;
lucl2=lrr20+1.96*sd20;
lcl1=exp(llcl1);
ucl1=exp(lucl1);
lcl2=exp(llcl2);
ucl2=exp(lucl2);
keep eta10 eta20 eyzero10 eyzero20 rr10 lcl1 ucl1 rr20 lcl2 ucl2;
run;

proc print data=sim.variances;run;
```

eAppendix 1(c). Linear SNM

```
*************************************************************************
****************************************************************
THIS IS THE LINEAR SNM
*This SAS program is based on the methodology developed and published in:
BA Brumback, Z He, M Prasad, MC Freeman, R Rheingans. Using structural-nested
models to estimate the effect of cluster-level adherence on individual-level
outcomes with a three-armed cluster-randomized trial. Statistics in medicine.
2014, 33 (9), 1490-1502

The methodology has been further discussed in:
S Helian, BA Brumback, MC Freeman, R Rheingans. Structural Nested Models for
Cluster-Randomized Trials. in Statistical Causal Inference and Their
Applications in Public Health Research, eds. H He, P Wu, DG Chen. Springer,
in press.

We acknowledge support from the National Science Foundation through grant NSF
SES-1115618 for this work.


The SAS program requires several variables:

Y – outcome variable
X – adherence variable denoting three adherence levels
X1 – indicator variable for X = 1
X2 – indicator variable for X = 2
W – previously created weight variable, which accounts for confounding by
individual-level baseline covariates
Z – randomization variable denoting three randomization arms
Z1 – indicator variable for Z = 1
Z2 – indicator variable for Z = 2
Stratum – variable denoting the strata
PSU – variable denoting the primary sampling units (e.g. schools)
Id – variable denoting a unique id for each participant in the study

***** The SAS program requires minor changes for:
studies with different numbers of PSU’s (it is now set up for 86 PSUs), or
studies with different numbers of strata (it is now set up for 2 strata).
The areas that would need to be changed are marked with comments.

***** The SAS program requires significant changes for:
a two armed trial (it is now set up for a three armed trial) which produces a
single adherence estimate (it now produces two estimates).
The areas that would need to be changed are marked with comments.
*************************************************************************
****************************************************************;

libname sim 'H:\IV analysis\linear';


*Rename the dataset sim.gooddata;
data sim.gooddata;
set weights;
run;
```

```
proc surveyfreq data=sim.gooddata;
tables Z*X / row cl;
weight W;
run;

*count the number of PSUs overall and the number of PSUs in each strata;
proc sort data=sim.gooddata;
by stratum psu;
run;
data PSU_level;
set sim.gooddata;
by stratum psu;
if first.psu;
keep stratum psu Z X ;
run;

Proc freq data = PSU_level;
tables Z X Z*X stratum/list missing;
run;


%macro simulate;

data sim.EYzeromat;
set _null_;
run;

data sim.etamat;
set _null_;
run;

proc sort data=sim.gooddata;
by stratum psu;
run;

data psustrat;
set sim.gooddata;
by stratum psu;
if first.psu;
keep stratum psu Z X;
run;

proc freq data=psustrat;
tables Z*X;
run;

data sim.psustrat;
set psustrat;
keep stratum psu;
run;

proc surveymeans data=sim.gooddata varmethod=jk(outweights=sim.jkweights);
weight W;
strata stratum;
cluster psu;
run;
```

```
%do i=0 %to 86; /*Number of PSUs varies by study and so this number may need
to change.*/

%if &i=0 %then %do;
data sim.data0;
set sim.gooddata;
run;
%end;

%else %do;

data sim.data0;
set sim.jkweights;
W=RepWt_&i;
run;

%end;

proc genmod data=sim.data0;
model Y = X1 X2 Z1 Z2 X1*Z1 X1*Z2 X2*Z1 X2*Z2;
weight W;
output out=sim.xbeta xbeta=linpred;
run;

data sim.xbeta;
set sim.xbeta;
keep id linpred;
run;

proc sort data=sim.xbeta;
by id;
run;

proc sort data=sim.data0;
by id;
run;

data sim.data0;
merge sim.data0 sim.xbeta;
by id;
run;

%macro iv;

proc syslin data=sim.data0 2sls;
endogenous X1 X2;
instruments Z1 Z2;
model Y =X1 X2;
weight W;
ods output ParameterEstimates=sim.squig;
run;

data sim.check1;
set sim.squig;
run;
```

```
data sim.squig;
set sim.squig;
keep Estimate;
run;

proc transpose data=sim.squig out=sim.squig;
run;

data sim.check2;
set sim.squig;
run;

data sim.squig;
set sim.squig;
squig0=COL1;
squig1=COL2;
squig2=COL3;
one=1;
keep one squig0 squig1 squig2;
run;

data sim.iv;
set sim.data0;
one=1;
run;

data sim.iv;
merge sim.iv sim.squig;
by one;
run;

%mend;

%let tol=1;

%iv;

data sim.iv;
set sim.iv;
Yzero = linpred-X1*squig1-X2*squig2;
run;

proc reg data=sim.iv;
model Yzero = X1 X2;
weight W;
ods output ParameterEstimates=sim.EYzero;
run;

proc reg data=sim.iv;
model Y = X1 X2;
weight W;
ods output ParameterEstimates=sim.eta;
run;

data sim.EYzero;
set sim.EYzero;
keep Estimate;
```

```
run;

data sim.eta;
set sim.eta;
keep Estimate;
run;

proc transpose data=sim.EYzero out=sim.EYzero;
run;

data sim.EYzero;
set sim.EYzero;
EYzero0=COL1;
EYzero1=COL2;
EYzero2=COL3;
EYzero1=EYzero0+EYzero1;
EYzero2=EYzero0+EYzero2;
keep EYzero1 EYzero2;
run;

proc transpose data=sim.eta out=sim.eta;
run;

data sim.eta;
set sim.eta;
eta0=COL1;
eta1=COL2;
eta2=COL3;
eta1=eta0+eta1;
eta2=eta0+eta2;
keep eta1 eta2;
run;

data sim.EYzeromat;
set sim.EYzeromat sim.EYzero;
run;

data sim.etamat;
set sim.etamat sim.eta;
run;

%end;

%mend;

%simulate;



*jackknife variance for linear SNM for the RD;
data eta;
set sim.etamat;
if _n_ > 1;
run;

data yzero;
```

```
set sim.EYzeromat;
if _n_ > 1;
run;

proc means data=eta;
run;

proc means data=yzero;
run;

data eta0;
set sim.etamat;
if _n_=1;
eta10=eta1;
eta20=eta2;
one=1;
keep eta10 eta20 one;
run;

data yzero0;
set sim.EYzeromat;
if _n_=1;
eyzero10=eyzero1;
eyzero20=eyzero2;
one=1;
keep eyzero10 eyzero20 one;
run;


proc freq data = PSU_level;
tables stratum /list missing;
run;

data sim.all;
merge eta yzero sim.psustrat;
rd1 = (eta1-eyzero1);
rd2 = (eta2-eyzero2);
if stratum=1 then mult=(48-1)/48; /*(Number of PSUs within stratum h minus
1)/(Number of PSUs within stratum h)*/
if stratum=2 then mult=(38-1)/38; /*(Number of PSUs within stratum h minus
1)/(Number of PSUs within stratum h)*/
/*if stratum=3 then mult=(number_in_stratum-1)/number_in_stratum;*/ /*And so
on, if there are more strata*/
one=1;
run;

data sim.all;
merge sim.all eta0 yzero0;
by one;
rd10=(eta10-eyzero10);
rd20=(eta20-eyzero20);
var1=mult*(rd1-rd10)**2;
var2=mult*(rd2-rd20)**2;
run;

proc univariate data=sim.all;
var var1 var2;
```

```
output out=sim.variances sum=var1 var2;
run;

data sim.variances;
merge eta0 yzero0 sim.variances;
rd10=eta10-eyzero10;
rd20=eta20-eyzero20;
sd10=sqrt(var1);
sd20=sqrt(var2);
lcl1=rd10-1.96*sd10;
ucl1=rd10+1.96*sd10;
lcl2=rd20-1.96*sd20;
ucl2=rd20+1.96*sd20;
keep eta10 eta20 eyzero10 eyzero20 rd10 lcl1 ucl1 rd20 lcl2 ucl2;
run;

proc print data=sim.variances;run;



*jackknife variance for linear SNM for the RR;
data eta;
set sim.etamat;
if _n_ > 1;
run;

data yzero;
set sim.EYzeromat;
if _n_ > 1;
run;

proc means data=eta;
run;

proc means data=yzero;
run;

data eta0;
set sim.etamat;
if _n_=1;
eta10=eta1;
eta20=eta2;
one=1;
keep eta10 eta20 one;
run;

data yzero0;
set sim.EYzeromat;
if _n_=1;
eyzero10=eyzero1;
eyzero20=eyzero2;
one=1;
keep eyzero10 eyzero20 one;
run;


proc freq data = PSU_level;
```

```
tables stratum /list missing;
run;

data sim.all;
merge eta yzero sim.psustrat;
lrr1 = log(eta1/eyzero1);
lrr2 = log(eta2/eyzero2);
if stratum=1 then mult=(48-1)/48; /*(Number of PSUs within stratum h minus
1)/(Number of PSUs within stratum h)*/
if stratum=2 then mult=(38-1)/38; /*(Number of PSUs within stratum h minus
1)/(Number of PSUs within stratum h)*/
/*if stratum=3 then mult=(number_in_stratum-1)/number_in_stratum;*/ /*And so
on, if there are more strata*/
one=1;
run;

data sim.all;
merge sim.all eta0 yzero0;
by one;
lrr10=log(eta10/eyzero10);
lrr20=log(eta20/eyzero20);
var1=mult*(lrr1-lrr10)**2;
var2=mult*(lrr2-lrr20)**2;
run;

proc univariate data=sim.all;
var var1 var2;
output out=sim.variances sum=var1 var2;
run;

data sim.variances;
merge eta0 yzero0 sim.variances;
rr10=eta10/eyzero10;
rr20=eta20/eyzero20;
lrr10=log(rr10);
lrr20=log(rr20);
sd10=sqrt(var1);
sd20=sqrt(var2);
llcl1=lrr10-1.96*sd10;
lucl1=lrr10+1.96*sd10;
llcl2=lrr20-1.96*sd20;
lucl2=lrr20+1.96*sd20;
lcl1=exp(llcl1);
ucl1=exp(lucl1);
lcl2=exp(llcl2);
ucl2=exp(lucl2);
keep eta10 eta20 eyzero10 eyzero20 rr10 lcl1 ucl1 rr20 lcl2 ucl2;
run;

proc print data=sim.variances;run;
```