/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* change the name of the library \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

libname RISQtest 'h:\Documents\risq\risq-test'; run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* fill out this section \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

%let popsize=35455;

%let samsize=35455;

%let variablenum=3; /\*\*\*total number of variables in model (including interactions) \*\*/

%let variablenoint=2;/\*\*number of main effects variables in model \*\*/

%let variablestrat=1;/\*\* number of stratifying variables not in the model for partial indicators\*\*/

%let variableinter=2;/\*\* number of variables that are in interactions (do not repeat variables)\*\*/

/\*\* names of main effects variables \*\*/

%let var1='gender';

%let var2='agea';

/\*\* interactions \*\*\*/

%let var3= 'hhtype\*urb';

/\*\* variables in interaction (do not repeat names) \*\*/

%let vvar1='hhtype';

%let vvar2='urb';

/\*\* names of stratifying variable for partial indicator not in the model\*\*/

%let strat1='marstat';

PROC FORMAT;

VALUE hhtype

1="Single" 2="Couple no children" 3="Couple with children" 4="Single parent" 5="Other";

VALUE agea 1="12-14" 2="15-17" 3="18-19" 4="20-24" 5="25-29"

6="30-34" 7="35-39" 8="40-44" 9="45-49" 10="50-54" 11="55-59"

12="60-64" 13="65-69" 14="70-74" 15="75+" ;

value urb

1="very strong" 2="Strong" 3="Average" 4="Little" 5="Not";

value gender

1="male" 2="female";

value marstat

1="Not married" 2="Married" 3="Widowed" 4="Divorced";

run;

data RISQtest.att;

set RISQtest.RISQtest;

responsesamp1= respons;

/\*\*\*\*\* responsesamp1 is the indicator for response, 1=response, 0=non-response\*/

agea=age-2; /\*transformations on the data\*/

pi=&samsize/&popsize;

piinv=1/pi; /\* or define piinv= dweight if there are differential design weights in the file \*/

run;

/\*\*\*\*\*\*\*\*\*\*\*\* no need to go beyond this point \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

data rlll;

data stt;

%macro c;

%do po=1 %to &variablenum;

data astt;

set stt;

length t $20;

t= &&var&po;

output;

run;

data rlll;

set rlll astt ;

\* drop t;

%end;

%mend;

%c;

run;

data flll;

data rst;

%macro c;

%do pr=1 %to &variablestrat;

data hrts;

set rst;

length t $20;

t=&&strat&pr;

output;

run;

data flll;

set flll hrts;

%end;

%mend;

%c;

run;

data nlll;

data ntt;

%macro c;

%do po=1 %to &variablenoint;

data nstt;

set ntt;

length t $20;

t= &&var&po;

output;

run;

data nlll;

set nlll nstt ;

%end;

%do po=1 %to &variableinter;

data ostt;

set ntt;

length t $20;

t=&&vvar&po;

output;

run;

data nlll;

set nlll ostt;

%end;

%mend;

%c;

run;

%let jj=%eval(&variableinter+&variablenoint);

%let allv=%eval(&variableinter+&variablenoint+&variablestrat);

data cllla;

set nlll;

if t ne ' ';

row+1;

run;

data t;

set cllla;

drop row;

data listofpartialbs;

set t flll;

if t ne ' ';

row+1;

run;

data listofpartialws;

set cllla;

if t ne ' ';

run;

%macro fff;

%do po=1 %to &jj;

data clll&po;

set cllla;

if row=&po then delete;

run;

%end;

%mend fff;

%fff;

run;

data sx ;

set rlll (keep=t);

length allvars $10000;

retain allvars ' ';

set rlll end=eof;

allvars=trim(left(allvars))||' '||left(t);

if eof then call symput ('varlist', allvars);

run;

data sx ;

set nlll (keep=t);

length allvars $10000;

retain allvars ' ';

set nlll end=eof;

allvars=trim(left(allvars))||' '||left(t);

if eof then call symput ('varlistno', allvars);

run;

proc logistic data=RISQtest.att outest= betasamp covout outdesign=xdfull noprint ;

class &varlistno ;

model responsesamp1= &varlist

/ expb ;\* selection=b ;

output out= predsamp p=phatsamp lower=lcl upper=ucl ;

run;

data rindicator;

set predsamp;

ns=piinv;

os=1;

mmm=1;

rphatsamp=1-phatsamp;

run;

proc means data=rindicator mean noprint;

var rphatsamp;

output out=gaa mean(rphatsamp)=propmean;

data t;

set gaa;

call symput('propmean',propmean);

run;

data Risqtest.rindicator;

set rindicator;

run;

proc export data=Risqtest.rindicator outfile="h:\Documents\risq\checkcbs\rindicator.dta";

run;

data allwr ; data allbr ; data allwrr; run;

%macro rrr;

%do po=1 %to &jj;

data sx ;

set clll&po (keep=t);

length allvars $10000;

retain allvars ' ';

set clll&po end=eof;

allvars=trim(left(allvars))||' '||left(t);

if eof then call symput ('varlist1', allvars);

run;

proc means sum mean data= rindicator nway noprint ;

var os rphatsamp ;

class &varlist1;

id mmm;

weight piinv;

output out=epw sum(os)=fbar&po mean(rphatsamp)=mrphat&po ;

run;

proc sort data=rindicator; by &varlist1; run;

data rindi ;

merge rindicator (in=v1) epw ;

by &varlist1;

if v1;

run;

proc means mean data=rindicator nway noprint ;

var rphatsamp ;

weight piinv;

output out=ephar mean(rphatsamp )=mrphatall;

data ephaar;

set ephar;

mmm=1;

data al;

merge epw (in=v1) ephaar;

by mmm;

if v1;

run;

data partialaan ;

set al;

p2Zka&po =sqrt(((fbar&po )/&popsize))\*(mrphat&po-mrphatall);

p1Zka&po =p2Zka&po\*p2Zka&po;

run;

proc means sum nway noprint ;

var p1Zka&po ;

output out=varr sum(p1Zka&po )=betweenvara;

run;

data allbr ;

set allbr varr ;

run;

data tall ;

set rindi ;

p3Zka&po=(1/(&popsize-1))\*piinv\*(rphatsamp -mrphat&po)\*\*2 ;

run;

proc means sum data=tall noprint nway ;

var p3Zka&po;

output out=tzr sum(p3Zka&po)=withinvara; run;

run;

data allwr ;

set allwr tzr ;

run;

data fmn;

set cllla;

if row ne &po then delete;

data afm;

set fmn;

call symput('hvar',t);

data sss;

set tall;

proc means sum data=sss nway noprint ;

var p3Zka&po;

class &hvar;

output out=tzrfina sum(p3Zka&po)=p3zk; run;

run;

data allwrr ;

set allwrr tzrfina ;

data RISQtest.partialp3&po;

set tzrfina ;

sqtp3zk=sqrt(p3zk);

drop \_type\_ \_freq\_;

%end;

%mend rrr;

%rrr;

run;

data allw;data allb; run;

%macro partial;

%do i=1 %to &variablenoint;

data r;

set rindicator;

length s $20;

s=&&var&i;

call symput('hvar',s);

proc means sum mean data= r nway noprint ;

var os rphatsamp ;

class &hvar;

weight piinv;

output out=ep&i sum(os)=fbar&i mean(rphatsamp)=mrphat&i ;

run;

proc sort data=r; by &hvar;

data rindicator;

merge r(in=v1) ep&i ;

by &hvar;

if v1;

run;

%end;

%let variab=%eval(&variablenoint+1);

%let endr=%eval(&variablenoint+&variableinter);

%do j=&variab %to &endr;

data r;

set rindicator;

%let i=%eval(&j-&variablenoint);

length s $20;

s=&&vvar&i;

call symput('hvar',s);

proc means sum mean data= r nway noprint ;

var os rphatsamp ;

class &hvar;

weight piinv;

output out=ep&j sum(os)=fbar&j mean(rphatsamp)=mrphat&j ;

run;

proc sort data=r; by &hvar;

data rindicator;

merge r(in=v1) ep&j ;

by &hvar;

if v1;

run;

%end;

%let variar=%eval(&variablenoint+1+&variableinter);

%let endd=%eval(&variablenoint+&variableinter+&variablestrat);

%do k=&variar %to &endd;

data r;

set rindicator;

%let t=%eval(&k-&variablenoint-&variableinter);

length s $20;

s=&&strat&t;

call symput('hvar',s);

proc means sum mean data= r nway noprint ;

var os rphatsamp ;

class &hvar;

weight piinv;

output out=ep&k sum(os)=fbar&k mean(rphatsamp)=mrphat&k ;

run;

proc sort data=r; by &hvar;

data rindicator;

merge r(in=v1) ep&k ;

by &hvar;

if v1;

run;

%end;

proc means mean data=rindicator nway noprint ;

var rphatsamp ;

weight piinv;

output out=epha mean(rphatsamp)=mrphatall;

data tk;

set epha;

call symput('phall',mrphatall);

run;

%do k=1 %to &endd;

data ephaa;

set epha;

\_type\_=1;

data a ;

merge ep&k (in=v1) ephaa;

by \_type\_;

if v1;

run;

data RISQtest.partialp2&k;

set a ;

p2Zk&k =sqrt(((fbar&k )/&popsize))\*(mrphat&k-mrphatall);

p1Zk&k =p2Zk&k\*p2Zk&k;

run;

proc means sum nway noprint ;

var p1Zk&k ;

output out=va sum(p1Zk&k )=betweenvar;

run;

data allb;

set allb va ;

%end;

%do kk=1 %to &endd;

data tall;

set rindicator;

p3Zk&kk=(1/(&popsize-1))\*piinv\*(rphatsamp-mrphat&kk)\*\*2 ;

run;

proc means sum data=tall noprint nway ;

var p3Zk&kk;

output out=tz sum(p3Zk&kk)=withinvar; run;

run;

data allw;

set allw tz ;

%end;

%mend partial;

%partial;

run;

%macro calcind;

proc means data=rindicator nway sum mean noprint ;

var rphatsamp ;

weight piinv;

output out=tt sum(rphatsamp)=sphatsamp mean(rphatsamp)=mphatsamp ;

run;

data uu;

set tt;

os=1;

HT=(1/&popsize)\*sphatsamp;

keep HT sphatsamp mphatsamp os;

run;

data cc;

merge rindicator (in=v1) uu;

by os;

if v1;

run;

data dd;

set cc;

sr\_ind= piinv\* (rphatsamp-HT)\*\*2;

proc means sum data=dd nway noprint ;

var sr\_ind ;

output out= result sum=;

run;

proc means data=dd nway var sum noprint ;

var sr\_ind ;

output out= uuu var(sr\_ind )= vsr\_ind sum(sr\_ind)=ssr\_ind;

run;

data part1ofvar;

set uuu;

vvv =&samsize\*(vsr\_ind );

run;

data aa;

set betasamp;

if \_type\_='PARMS';

drop \_link\_ \_type\_ \_status\_ \_name\_ \_lnlike\_;

run;

data aaa ;

set betasamp;

if \_type\_='COV';

if intercept ne .;

drop \_link\_ \_type\_ \_status\_ \_name\_ \_lnlike\_;

run;

data rdfull ;

set xdfull;

drop responsesamp1;

run;

proc iml;

use aaa ;

read all into bet;

use aa ;

read all into parms ;

use rdfull;

read all into x1;

use rindicator;

read all var {piinv} into invpi;

ff=x1 \*parms`;

eff=exp(ff)/(1+exp(ff));

dr=1-eff;

gexp=dr#eff;

nr=nrow(eff);

avero=eff[+]/nr;

rr=eff-avero;

b=gexp#x1;

nb=nrow(b);

aveb=b[+,]/nb;

nff=ncol(x1);

create columnsd from nff;

append from nff;

c=j(nb,nff,1);

d=c#aveb;

zz=b-d;

zzz=invpi#zz;

terma=rr`\*zzz;

termb= zzz`\* zz ;

first=terma\*bet;

second=first\*terma`;

one=second#4;

next=termb\*bet;

tt=next\*next ;

uuu=2#trace(tt);

secondterm=one+uuu;

create secondt from secondterm;

append from secondterm;

c=gexp#gexp;

create apelet from c;

append from c;

va=x1\*bet;

create bpelet from va;

append from va;

create cpelet from x1 ;

append from x1;

create dpelet from invpi;

append from invpi;

quit;run;

data cc;

set columnsd;

call symput('numc',col1);

run;

%let numb=&numc;

run;

data gg;

set cpelet;

array acol acol1-acol&numb;

array col col1- col&numb;

do i=1 to &numb;

acol(i)=col(i);

end;

keep acol1-acol&numb;

run;

data hh;

set bpelet; set gg;

array acol acol1-acol&numb;

array col col1- col&numb;

h=0;

do i=1 to &numb;

h=h+acol(i)\*col(i);

end;

run;

data rrr;

set apelet;

cccc=col1;

keep cccc;

data qqq;

set dpelet;

dddd=col1;

keep dddd;

data ii;

set hh; set rrr; set qqq;

o=h\*cccc\*dddd;

proc means sum data=ii nway noprint;

var o;

output out=kl sum=;

run;

data g;

set kl;

bia=o ;

keep bia ;

run;

data r;

set part1ofvar; set secondt;

variance\_r=(1/&popsize)\*(1/ssr\_ind)\*(vvv+col1);

std\_r=sqrt(variance\_r);

ci\_r=std\_r\*1.96;

run;

data t;

set result;set g ;

srvar=(1/(&popsize-1))\*((1+1/&samsize-1/&popsize)\*sr\_ind-bia);

r\_indicator=1-2\*sqrt(srvar);

svarb=(1/(&popsize-1)) \*(sr\_ind );

r\_withbias=1-2\*sqrt(svarb);

run;

data finalfilepart;

set t; set r;

keep r\_indicator r\_withbias variance\_r std\_r ci\_r bia sr\_ind srvar svarb;

run;

%mend calcind;

%calcind;

run;

data RISQtest.finalfileR\_ind;

set finalfilepart;

data tallw;

set allw;

if \_type\_ ne .;

data tallb;

set allb;

if \_type\_ ne .;

data tallwr;

set allwr;

if \_type\_ ne .;

data tallbr;

set allbr;

if \_type\_ ne .;

run;

data k;

set RISQtest.finalfileR\_ind;

bipart=bia/&popsize;

call symput('bias',bipart);

run;

data RISQtest.partialbetween;

set tallw; set tallb;

t=&bias;

m=withinvar+betweenvar;

sqrtbetween=sqrt(betweenvar);

s1=t\*withinvar/(withinvar+betweenvar);

s2=t\*betweenvar/(withinvar+betweenvar);

withinbiasadj=withinvar-s1;

betweenbiasadj=betweenvar-s2;

sqtwithinbiasadj =sqrt( withinbiasadj);

sqtbetweenbiasadj =sqrt( betweenbiasadj);

run;

data keepbiasbetween;

set RISQtest.partialbetween;

biasbetw=s2;

row+1;

keep biasbetw row;

run;

%macro hgt;

%do ji=1 %to &allv;

data a;

set keepbiasbetween;

if row=&ji;

call symput ('biasbet',biasbetw);

run;

data r;

set RISQtest.partialp2&ji;

mm=1;

proc means data=r noprint nway;

var p1zk&ji;

output out=oo sum(p1zk&ji)=sp1zk;

data ooo;

set oo;

mm=1;

keep mm sp1zk;

data ch;

merge r (in=v1) ooo;

by mm;

run;

data cch;

set ch;

s=&biasbet;

correct=s\*p1zk&ji/sp1zk;

p1zkbiasadj=p1zk&ji-correct;

if p2zk&ji lt 0 then sign=-1; else sign=1;

p2zkbiasadj=sqrt(p1zkbiasadj)\*sign;

run;

data RISQtest.partialp2&ji;

set cch;

drop mm;

run;

%end;

%mend hgt;

%hgt;

run;

data RISQtest.partialwithin;

set tallwr; set tallbr;

t=&bias;

m=withinvara+betweenvara;

sqrtwithina =sqrt(withinvara);

s1=t\*withinvara/(withinvara+betweenvara);

s2=t\*betweenvara/(withinvara+betweenvara);

withinabiasadj=withinvara-s1;

betweenabiasadj=betweenvara-s2;

mm= withinabiasadj+ betweenabiasadj;

sqtwithinabiasadj =sqrt( withinabiasadj);

sqtbetweenabiasadj =sqrt( betweenabiasadj);

run;

data keepbiaswithin;

set RISQtest.partialwithin;

biaswith=s1;

row+1;

keep biaswith row ;

run;

%macro hgt;

%do ji=1 %to &jj;

data a;

set keepbiaswithin;

if row=&ji;

call symput ('biaswit',biaswith);

run;

data r;

set RISQtest.partialp3&ji;

mm=1;

proc means data=r noprint nway;

var p3zk;

output out=oo sum(p3zk )=sp3zk ;

run;

data ooo;

set oo;

mm=1;

keep mm sp3zk ;

data ch;

merge r (in=v1) ooo;

by mm;

run;

data cch;

set ch;

s=&biaswit;

correct=s\*p3zk /sp3zk ;

p3zkbiasadj =p3zk -correct;

sqtp3zkbiasadj =sqrt(p3zkbiasadj ) ;

run;

data RISQtest.partialp3&ji;

set cch;

drop mm;

run;

%end;

%mend hgt;

%hgt;

run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\* outputs \*\*\*\*\*\*\*\*\*\*\*/

data u;

set RISQtest.finalfileR\_ind;

LB\_r=r\_indicator-ci\_r;

UB\_r=r\_indicator+ci\_r;

propmean=&propmean;

r\_unadjusted=r\_withbias;

SE\_r=std\_r;

proc print data=u;

Title "R-Indicator";

var r\_indicator r\_unadjusted propmean SE\_r LB\_r UB\_r;

run;

data RISQtest.finalfiler\_ind;

set u;

run;

data c;

set RISQtest.partialbetween;

bias=t;

drop t;

run;

data a;

set c; set listofpartialbs;

Pu=sqtbetweenbiasadj;

Pu\_unadjusted=sqrtbetween;

drop \_type\_ \_freq\_ ;

proc print data=a;

Title "Variable Level Unconditional Partial Indicators";

var t Pu Pu\_unadjusted ;

run;

data RISQtest.partialbetween;

set a;

run;

data c;

set RISQtest.partialwithin;

bias=t;

drop t;

data a;

set c; set listofpartialws;

Pc=sqtwithinabiasadj;

Pc\_unadjusted=sqrtwithina;

drop \_type\_ \_freq\_ ;

proc print data=a;

Title "Variable Level Conditional Partial Indicators";

var t Pc Pc\_unadjusted ;

run;

data RISQtest.partialwithin;

set a;

run;

data g;

%macro pr;

%do i=1 %to &jj;

data tt;

set listofpartialws;

if row=&i;

run;

data uu;

set tt;

us=compress(t)||".";

call symput('hvar',t);

call symput('hhvar',us);

run;

data a;

set RISQtest.partialp3&i;

Pc= sqtp3zkbiasadj;

Pc\_unadjusted=sqtp3zk;

drop sp3zk s correct ;

run;

data g;

set g a;;

proc print data=a;

format &hvar &hhvar;

Title "Category Level Conditional Partial Indicators";

var &hvar Pc Pc\_unadjusted;

run;

%end;

%mend;

%pr;

run;

data RISQtest.partialp3all;

set g;

if p3zk ne .;

run;

data f;

%macro pr;

%do i=1 %to &allv;

data tt;

set listofpartialbs;

if row=&i;

run;

data uu;

set tt;

us=compress(t)||".";

call symput('hvar',t);

call symput('hhvar',us);

run;

data a;

set RISQtest.partialp2&i;

data aa;

set a;

p2zk=p2zk&i;

p1zk=p1zk&i;

Pu= p2zkbiasadj ;

Pu\_unadjusted= p2zk;

drop p2zk&i p1zk&i \_type\_ \_freq\_ fbar&i mrphat&i sp1zk s correct sign mrphatall;

data f;

set f aa;

proc print data=aa;

format &hvar &hhvar;

Title "Category Level Unconditional Partial Indicators";

var &hvar pu pu\_unadjusted;

run;

%end;

%mend;

%pr;

run;

data RISQtest.partialp2all;

set f;

if p1zk ne . ;

run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*\*\*\* confidence intervals for partial indicators \*\*\*\*/

/\*

%macro gba;

%do k=1 %to &allv;

data RISQtest.rrsta&k;

set RISQtest.partialp2&k;

p2zk=p2zk&k;

p1zk=p1zk&k;

drop p2zk&k p1zk&k;

run;

%end;

%mend gba;

%gba;

run;

%macro gb;

%do po=1 %to &jj;

data RISQtest.rrst&po;

set RISQtest.partialp3&po;

run;

%end;

%mend gb;

%gb;

run;

data RISQtest.prrst;

set RISQtest.partialbetween;

data RISQtest.prrsta;

set RISQtest.partialwithin;

run;

data RISQtest.final;

set RISQtest.finalfiler\_ind;

run;

%macro confid;

%do q= 1 %to 500;

data akl;

set RISQtest.att ;

proc surveyselect data=akl method = urs samprate = 1 outhits noprint

out=akla;

run;

proc logistic data=akla outest= betasamp covout outdesign=xdfull noprint ;

class &varlistno ;

model responsesamp1= &varlist

/ expb ;

output out= predsamp p=phatsamp lower=lcl upper=ucl ;

run;

data rindicator;

set predsamp;

ns=piinv;

os=1;

mmm=1;

rphatsamp=1-phatsamp;

run;

data allwr ; data allbr ; data allwrr; run;

%macro rrr;

%do po=1 %to &jj;

data sx ;

set clll&po (keep=t);

length allvars $10000;

retain allvars ' ';

set clll&po end=eof;

allvars=trim(left(allvars))||' '||left(t);

if eof then call symput ('varlist1', allvars);

run;

proc means sum mean data= rindicator nway noprint ;

var os rphatsamp ;

class &varlist1;

id mmm;

weight piinv;

output out=epw sum(os)=fbar&po mean(rphatsamp)=mrphat&po ;

run;

proc sort data=rindicator; by &varlist1; run;

data rindi ;

merge rindicator (in=v1) epw ;

by &varlist1;

if v1;

run;

proc means mean data=rindicator nway noprint ;

var rphatsamp ;

output out=ephar mean(rphatsamp )=mrphatall;

data ephaar;

set ephar;

mmm=1;

data al;

merge epw (in=v1) ephaar;

by mmm;

if v1;

run;

data partialaan ;

set al;

p2Zka&po =sqrt(((fbar&po )/&popsize))\*(mrphat&po-mrphatall);

p1Zka&po =p2Zka&po\*p2Zka&po;

run;

proc means sum nway noprint ;

var p1Zka&po ;

output out=varr sum(p1Zka&po )=betweenvara;

run;

data allbr ;

set allbr varr ;

run;

data tall ;

set rindi ;

p3Zka&po=(1/(&popsize-1))\*piinv\*(rphatsamp -mrphat&po)\*\*2 ;

run;

proc means sum data=tall noprint nway ;

var p3Zka&po;

output out=tzr sum(p3Zka&po)=withinvara; run;

run;

data allwr ;

set allwr tzr ;

run;

data sss;

set tall;

s=&&var&po;

call symput('hvar',s);

proc means sum data=sss nway noprint ;

var p3Zka&po;

class &hvar;

output out=tzrfina sum(p3Zka&po)=p3zk; run;

run;

data allwrr ;

set allwrr tzrfina ;

data sta;

set tzrfina ;

sqtp3zk&q=sqrt(p3zk);

p3zk&q=p3zk;

keep &hvar sqtp3zk&q p3zk&q;

data RISQtest.rrst&po;

merge RISQtest.rrst&po sta ;

by &hvar;

run;

%end;

%mend rrr;

%rrr;

run;

data allw;data allb; run;

%macro partial;

%do i=1 %to &variablenoint;

data r;

set rindicator;

length s $20;

s=&&var&i;

call symput('hvar',s);

proc means sum mean data= r nway noprint ;

var os rphatsamp ;

class &hvar;

weight piinv;

output out=ep&i sum(os)=fbar&i mean(rphatsamp)=mrphat&i ;

run;

proc sort data=r; by &hvar;

data rindicator;

merge r(in=v1) ep&i ;

by &hvar;

if v1;

run;

%end;

%let variab=%eval(&variablenoint+1);

%let endr=%eval(&variablenoint+&variableinter);

%do j=&variab %to &endr;

data r;

set rindicator;

%let i=%eval(&j-&variablenoint);

length s $20;

s=&&vvar&i;

call symput('hvar',s);

proc means sum mean data= r nway noprint ;

var os rphatsamp ;

class &hvar;

weight piinv;

output out=ep&j sum(os)=fbar&j mean(rphatsamp)=mrphat&j ;

run;

proc sort data=r; by &hvar;

data rindicator;

merge r(in=v1) ep&j ;

by &hvar;

if v1;

run;

%end;

%let variar=%eval(&variablenoint+1+&variableinter);

%let endd=%eval(&variablenoint+&variableinter+&variablestrat);

%do k=&variar %to &endd;

data r;

set rindicator;

%let t=%eval(&k-&variablenoint-&variableinter);

length s $20;

s=&&strat&t;

call symput('hvar',s);

proc means sum mean data= r nway noprint ;

var os rphatsamp ;

class &hvar;

weight piinv;

output out=ep&k sum(os)=fbar&k mean(rphatsamp)=mrphat&k ;

run;

proc sort data=r; by &hvar;

data rindicator;

merge r(in=v1) ep&k ;

by &hvar;

if v1;

run;

%end;

proc means mean data=rindicator nway noprint ;

var rphatsamp ;

output out=epha mean(rphatsamp)=mrphatall;

data tk;

set epha;

call symput('phall',mrphatall);

run;

%do k=1 %to &endd;

data ephaa;

set epha;

\_type\_=1;

data a ;

merge ep&k (in=v1) ephaa;

by \_type\_;

if v1;

run;

data bcd;

set a ;

p2Zk =sqrt(((fbar&k )/&popsize))\*(mrphat&k-mrphatall);

p1Zk =p2Zk \*p2Zk ;

run;

proc means sum nway noprint ;

var p1Zk ;

output out=va sum(p1Zk )=betweenvar;

run;

data ccd ;

set bcd ;

p2Zk&q=p2zk ;

p1zk&q=p1zk ;

keep p2zk&q p1zk&q ;

run;

data RISQtest.rrsta&k;

set RISQtest.rrsta&k ; set ccd ;

run;

data allb;

set allb va ;

%end;

%do kk=1 %to &endd;

data tall;

set rindicator;

p3Zk&kk=(1/(&popsize-1))\*piinv\*(rphatsamp-mrphat&kk)\*\*2 ;

run;

proc means sum data=tall noprint nway ;

var p3Zk&kk;

output out=tz sum(p3Zk&kk)=withinvar; run;

run;

data allw;

set allw tz ;

%end;

%mend partial;

%partial;

run;

%calcind;

run;

data RISQtest.final;

set RISQtest.final finalfilepart;

run;

data tallw;

set allw;

if \_type\_ ne .;

data tallb;

set allb;

if \_type\_ ne .;

data tallwr;

set allwr;

if \_type\_ ne .;

data tallbr;

set allbr;

if \_type\_ ne .;

run;

data k;

set finalfilepart;

bipart=bia/&popsize;

call symput('bias',bipart);

data partialr12;

set tallw; set tallb;

t=&bias;

m=withinvar+betweenvar;

sqrtbetween=sqrt(betweenvar);

s1=t\*withinvar/(withinvar+betweenvar);

s2=t\*betweenvar/(withinvar+betweenvar);

withinbiasadj=withinvar-s1;

betweenbiasadj&q=betweenvar-s2;

sqtwithinbiasadj =sqrt( withinbiasadj);

sqtbetweenbiasadj&q =sqrt( betweenbiasadj&q);

run;

data par12;

set partialr12;

keep betweenbiasadj&q sqtbetweenbiasadj&q;

run;

data RISQtest.prrst;

set RISQtest.prrst; set par12;

run;

data keepbiasbetween;

set partialr12;

biasbetw=s2;

row+1;

keep biasbetw row;

run;

%macro hgt;

%do ji=1 %to &allv;

data a;

set keepbiasbetween;

if row=&ji;

call symput ('biasbet',biasbetw);

run;

data r;

set RISQtest.rrsta&ji;

mm=1;

proc means data=r noprint nway;

var p1zk&q;

output out=oo sum(p1zk&q)=sp1zk ;

data ooo;

set oo;

mm=1;

keep mm sp1zk ;

data ch;

merge r (in=v1) ooo;

by mm;

run;

data cch;

set ch;

s=&biasbet;

correct=s\*p1zk&q/sp1zk ;

p1zkbiasadj&q=p1zk&q-correct;

if p2zk&q lt 0 then sign=-1; else sign=1;

p2zkbiasadj&q=sqrt(p1zkbiasadj&q)\*sign;

run;

data RISQtest.rrsta&ji;

set cch;

drop mm;

run;

%end;

%mend hgt;

%hgt;

run;

data partialr3;

set tallwr; set tallbr;

t=&bias;

m=withinvara+betweenvara;

sqrtwithina =sqrt(withinvara);

s1=t\*withinvara/(withinvara+betweenvara);

s2=t\*betweenvara/(withinvara+betweenvara);

withinabiasadj&q=withinvara-s1;

betweenabiasadj=betweenvara-s2;

mm= withinabiasadj&q+ betweenabiasadj;

sqtwithinabiasadj&q=sqrt( withinabiasadj&q);

sqtbetweenabiasadj =sqrt( betweenabiasadj);

run;

data par12a;

set partialr3;

keep withinabiasadj&q sqtwithinabiasadj&q;

data RISQtest.prrsta;

set RISQtest.prrsta; set par12a;

run;

data keepbiaswithin;

set partialr3;

biaswith=s1;

row+1;

keep biaswith row ;

run;

%macro hgt;

%do ji=1 %to &jj;

data a;

set keepbiaswithin;

if row=&ji;

call symput ('biaswit',biaswith);

run;

data r;

set RISQtest.rrst&ji;

mm=1;

proc means data=r noprint nway;

var p3zk&q;

output out=oo sum(p3zk&q )=sp3zk ;

run;

data ooo;

set oo;

mm=1;

keep mm sp3zk ;

data ch;

merge r (in=v1) ooo;

by mm;

run;

data cch;

set ch;

s=&biaswit;

correct=s\*p3zk&q /sp3zk ;

p3zkbiasadj&q =p3zk&q -correct;

sqtp3zkbiasadj&q =sqrt(p3zkbiasadj&q ) ;

run;

data RISQtest.rrst&ji;

set cch;

drop mm;

run;

%end;

%mend hgt;

%hgt;

run;

%end;

%mend confid;

%confid;

run;

\*/

/\*

data h;

set RISQtest.final;

keep r\_indicator;

proc univariate loccount noprint;

var r\_indicator;

output out= vrind mean=meanrind var=varrind std =stdrind pctlpre=rindce\_ci\_ pctlpts=2.5, 97.5;

run;

data g;

set RISQtest.finalfiler\_ind; set vrind;

run;

data RISQtest.finalfiler\_ind;

set g;

run;

data a; data b; run;

%macro ppp;

%do pp=1 %to &jj;

data hb&pp;

set RISQtest.rrst&pp;

keep sqtp3zk1-sqtp3zk1000 ;

data hba&pp;

set RISQtest.rrst&pp;

keep sqtp3zkbiasadj1-sqtp3zkbiasadj1000 ;

data a;

set a hb&pp;

data b;

set b hba&pp;

run;

%end;

%mend ppp;

%ppp;

run;

data aa;

set a;

if sqtp3zk1 ne .;

data aaa;

set aa;

row+1;

run;

proc transpose data= aaa out=t; by row;run;

proc univariate loccount noprint;

var col1;

by row;

output out= vpartialwp3 mean=meanp3 var=varp3 std =stdp3 pctlpre=p3ce\_ci\_ pctlpts=2.5, 97.5;

run;

data bb;

set b;

if sqtp3zkbiasadj1 ne .;

data bbb;

set bb;

row+1;

run;

proc transpose data= bbb out=t; by row;run;

proc univariate loccount noprint;

var col1;

by row;

output out= vpartialwp3b mean=meanp3adj var=varp3adj std =stdp3adj pctlpre=p3adjce\_ci\_ pctlpts=2.5, 97.5;

run;

data f;

set RISQtest.partialp3all; set vpartialwp3; set vpartialwp3b;

run;

data RISQtest.partialp3all;

set f;

run;

data b; data c; run;

%macro ppp;

%do pp=1 %to &allv;

data h&pp;

set RISQtest.rrsta&pp;

keep p2zk1-p2zk1000 ;

data ha&pp;

set RISQtest.rrsta&pp;

keep p2zkbiasadj1-p2zkbiasadj1000 ;

data b;

set b h&pp;

data c;

set c ha&pp;

run;

%end;

%mend ppp;

%ppp;

run;

data bb;

set b;

if p2zk1 ne .;

data bbb;

set bb;

row+1;

run;

proc transpose data=bbb out=t; by row;run;

proc univariate loccount noprint;

var col1;

by row;

output out= vpartialbp2 mean=meanp2 var=varp2 std=stdp2 pctlpre=p2ce\_ci\_ pctlpts=2.5, 97.5;

run;

data cc;

set c;

if p2zkbiasadj1 ne .;

data ccc;

set cc;

row+1;

run;

proc transpose data=ccc out=t; by row;run;

proc univariate loccount noprint;

var col1;

by row;

output out= vpartialbp2b mean=meanp2adj var=varp2adj std=stdp2adj pctlpre=p2adjce\_ci\_ pctlpts=2.5, 97.5;

run;

data f;

set RISQtest.partialp2all; set vpartialbp2; set vpartialbp2b;

run;

data RISQtest.partialp2all;

set f;

run;

data g1;

set RISQtest.prrst;

row+1;

keep sqtbetweenbiasadj1-sqtbetweenbiasadj1000 row;

run;

proc transpose data= g1 out=t;by row;run;

proc univariate loccount noprint;

var col1;

by row;

output out= vbetween mean=meanbetween var=varbetween std=stdbetween pctlpre=betweence\_ci\_ pctlpts=2.5, 97.5;

run;

data g;

set RISQtest.partialbetween; set vbetween;

run;

data RISQtest.partialbetween;

set g;

run;

data g2;

set RISQtest.prrsta;

row+1;

keep sqtwithinabiasadj1-sqtwithinabiasadj1000 row;

run;

proc transpose data= g2 out=t;by row;run;

proc univariate loccount noprint;

var col1;

by row;

output out=vwithin mean=meanwithin var=varwithin std=stdwithin pctlpre=withince\_ci\_ pctlpts=2.5, 97.5;

run;

data g;

set RISQtest.partialwithin; set vwithin;

run;

data RISQtest.partialwithin;

set g;

run;

\*/