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

libname risq 'F:\Documents\risq\risq-test'; run;

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

%let popsize=3545500 ;

%let samsize=35455;

%let variablenum=2; /\*\*\*total number of variables in model (including interactions) \*\*/

%let variablenoint=1;/\*\*number of main effects variables in model \*\*/

%let variablestrat=1;/\*\* number of stratifying variables not in the model for unconditional partial R-indicator\*\*/

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

%let strat1='joba';

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

/\*\* names of variables in interactions\*\*/

%let vvar1='gender';

%let vvar2='urb';

/\*\* names of variables in original response model \*\*/

%let var1='agea';

%let var2='gender\*urb';

/\* names of all variables for partial R- indicators (in order of variables in original response model,

stratifying variables at the end\*/

%let xvar1=agea ;

%let xvar2= gender ;

%let xvar3= urb ;

%let xvar4=joba;

/\* number of categories of each variable\*/

%let nvar1=14;

%let nvar2=2;

%let nvar3=5;

%let nvar4=2;

PROC FORMAT;

VALUE agea

1="15-17"

2="18-19"

3="20-21"

4="22-24"

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";

value gender

1="male"

2="female" ;

value urb

1="very strong"

2="Strong"

3="Average"

4="Little"

5="Not";

value joba

1="unemployed"

2="employed";

run;

data att;

set risq.controlcbs;

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

responsesamp1= respons;

/\*define necessary transformations in the data\*/

agex=age-2;

if agex=1 or agex=2 then agea=1;

if agex ge 3 then agea=agex-1;

joba=job+1;

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

pi=&samsize/&popsize;

piinv=1/pi;

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

run;

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

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

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

data risq.a;

set att;

run;

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;

%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;

%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;

%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=risq.a outest= betasamp covout outdesign=xdfull noprint ;

class &varlistno ;

model responsesamp1= &varlist

/ expb ;

output out= predsamp p=phatsamp ;

run;

data rindicator;

set predsamp;

ns=piinv;

os=1;

mmm=1;

rphatsamp= 1-phatsamp;

run;

data predsampall;

set predsamp;

run;

data rindicatorall;

set rindicator;

run;

data betasampall;

set betasamp;

run;

data xdfullall;

set xdfull;

data rindicatora ;

set rindicator;

run;

data t;

set xdfullall;

drop responsesamp1;

proc contents noprint data=t out=metadata;

run;

Proc means data=metadata nway noprint ;

output out=df N=;

run;

data ddf;

set df;

g=\_freq\_;

call symput('nocat', g);

run;

%put &nocat;run;

%macro calcind;

proc means data=rindicatora nway sum mean var noprint ;

var rphatsamp ;

weight piinv;

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

run;

data uu;

set tt;

os=1;

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

keep HT sphatsamp mphatsamp os;

run;

data ll;

set uu;

call symput('htmean',ht);

run;

data cc;

merge rindicatora (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;

rf=rr`;

terma=rf\*zzz;

tee=zzz`;

termb=tee\*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);

cv=sqrt(srvar)/&htmean;

cv\_unadjusted=sqrt(svarb)/&htmean;

run;

data finalfilepart;

set t; set r;

avgprop=&htmean;

var\_cv =((variance\_r/4)/(avgprop\*avgprop)) + ( (srvar \*srvar ) / (&samsize\*(avgprop\*\*4))) ;

std\_cv =sqrt(var\_cv );

var\_cv\_unadjusted =((variance\_r/4)/(avgprop\*avgprop)) + ( (svarb \*svarb ) / (&samsize\*(avgprop\*\*4))) ;

std\_cv\_unadjusted =sqrt(var\_cv\_unadjusted);

keep r\_indicator r\_withbias variance\_r std\_r ci\_r bia sr\_ind srvar svarb avgprop cv cv\_unadjusted std\_cv std\_cv\_unadjusted;

run;

%mend calcind;

%calcind;

run;

proc print data= finalfilepart; run;

data bipartall;

set finalfilepart;

bipartall=bia/&popsize;

call symput('biasall',bipartall);

run;

data finalfileR\_ind;

set finalfilepart;

run;

/\* partial conditional R- indicator\*/

data allwr ; data allbr ; data allwrr; data varc\_level ;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 logistic data=risq.a outest= betasamp covout outdesign=xdfull noprint ;

class &varlist1 ;

model responsesamp1= &varlist1

/ expb ;

output out= predsamp p=phatsamp ;

run;

data rindicat ;

set predsamp;

ns=piinv;

os=1;

mmm=1;

rphatsamp= 1-phatsamp;

run;

data rindicatora;

set rindicat;

run;

%calcind;

data varx&po;

set finalfilepart;

varx=std\_r/2;

run;

proc means sum mean data= rindicatorall nway noprint ;

var ns rphatsamp ;

class &varlist1;

id mmm;

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

run;

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

data rindi&po ;

merge rindicatorall (in=v1) epw ;

by &varlist1;

if v1;

run;

proc means mean data=rindicatorall nway noprint ;

var rphatsamp ;

output out=ephar mean(rphatsamp )=mrphatall ;

run;

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&po ;

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

run;

proc means sum data=tall nway noprint ;

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);

run;

data sss;

set tall;

sss=1;

ttt=piinv;

run;

proc means sum var data=sss nway noprint ;

var p3Zka&po sss ttt ;

class &hvar;

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

sum(sss ttt)=ns ps; run;

run;

data allwrr ;

set allwrr tzrfina ;

data partialp3&po;

set tzrfina ;

sqtp3zk=sqrt(p3zk);

drop \_type\_ \_freq\_;

run;

data varc\_level;

set varc\_level varx&po;

run;

%end;

%mend rrr;

%rrr;

run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*variance p3 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

data gg;

set xdfullall;

drop responsesamp1;

run;

proc contents noprint data=gg out=metadata;

run;

Proc means data=metadata nway noprint ;

output out=df N=;

run;

data ddf;

set df;

g=\_freq\_;

call symput('noca', g);

run;

%put &noca;run;

data varfinalp3;run;

%macro varp3;

%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;

%do qo=1 %to &&nvar&po;

data gr;

set predsampall;

rphatsamp= phatsamp;

data trla;

set gr;

z1=rphatsamp\*(1-rphatsamp);

sortrow+1;

run;

data trl;

set trla;

keep z1;

run;

proc iml;

use trl ;

read all into z;

use gg;

read all into x;

f=z#x;

create terma from f;

append from f;

quit;run;

data l;

set trla; set terma;

run;

%let noc=&noca;

proc means mean data=l nway noprint ;

var col1-col&noc;

class &varlist1;

output out=kaa mean(col1-col&noc)=meanc1-meanc&noc;

run;

data aa;

set l;

run;

proc sort ; by &varlist1;

data bb;

set kaa;

drop \_type\_ \_freq\_;

proc sort; by &varlist1;

run;

data caa;

merge bb aa (in=v1);

by &varlist1 ;

if v1;

run;

proc sort data=caa; by sortrow; run;

data ll;

set caa;

array zz zz1-zz&noc;

array col col1-col&noc;

array meanc meanc1-meanc&noc;

do i=1 to &noc;

zz(i)=col(i)-meanc(i);

end;

run;

data trl;

set gr;

z1=rphatsamp ;

sortrow+1;

run;

proc means mean data=trl nway noprint;

var z1;

class &varlist1;

output out=ka mean(z1)=mz1;

run;

data aa;

set trl;

proc sort; by &varlist1;

data bb;

set ka ;

drop \_type\_ \_freq\_;

proc sort; by &varlist1;

run;

data caa1;

merge bb aa (in=v1);

by &varlist1 ;

if v1;

run;

proc sort data=caa1; by sortrow;

data llf;

set caa1;

yy=rphatsamp-mz1;

run;

data ab1;

set llf;

keep yy;

data ab2;

set ll;

keep zz1-zz&noc;\*&allv;

run;

data aaa ;

set betasampall;

if \_type\_='COV';

if intercept ne .;

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

run;

data rdfull ;

set xdfullall;

drop responsesamp1;

run;

data lla;

set llf;

if &&xvar&po=&qo then delta=1; else delta=0;

keep delta;

run;

proc iml;

use aaa ;

read all into bet;

use rdfull;

read all into x1;

use lla;

read all into delta;

use rindicatorall;

read all var {piinv} into invpi;

use ab1;

read all into rhoavg;

use ab2;

read all into zzavg;

rr=rhoavg#delta;

zz=zzavg#delta;

zzz=invpi#zz;

terma=rr`\*zzz;

terman=terma/&popsize;

termb= zzz`\* zz ;

termbn=termb/&popsize;

first=terman\*bet;

second=first\*terman`;

one=second#4;

next=termbn\*bet;

tt=next\*next ;

uuu=2#trace(tt);

secondterm=one+uuu;

create secondt from secondterm;

append from secondterm;

quit;run;

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

data fr;

set rindi&po;

if &&xvar&po=&qo then delta=1; else delta=0;

a=piinv\* delta\*rphatsamp\*rphatsamp;

b=piinv\*delta\*rphatsamp;

c=piinv\*rphatsamp;

d=piinv;

e=piinv\*delta;

cou=1;

run;

proc means sum var data=fr nway noprint;

var a b c d e cou;

class &varlist1;

output out=ka sum(a b c d e )=suma sumb sumc sumd sume

sum(cou)=sc;

run;

data f;

set ka;

g=(1/&popsize)\*(suma-(2\*sumb\*sumc)/sumd+(sumc\*sumc\*sume)/(sumd\*sumd));

run;

data a;

set f;

drop \_type\_ \_freq\_;

proc sort; by &varlist1;

data b;

set fr;

proc sort; by &varlist1;

run;

data c;

merge b a (in=v1);

by &varlist1 ;

if v1;

run;

data try;

set c;

phi= (1/&popsize)\*(

(delta\*rphatsamp\*rphatsamp)-

(2\*sumc\*delta\*rphatsamp/sumd)+

(-2\*sumb/sumd+2\*sumc\*sume/(sumd\*sumd))\*rphatsamp+

(2\*sumb\*sumc/(sumd\*sumd)-2\*sumc\*sumc\*sume/(sumd\*sumd\*sumd))

+(sumc\*sumc\*delta/(sumd\*sumd)));

dphi= piinv\* phi;

count=1;

ccpop=piinv;

run;

proc means sum var data=try nway noprint ;

var ccpop count dphi ;

output out=v1 sum(ccpop count

dphi )=sumccpop sumcount

sdphi var(dphi)=vdphi;

run;

data uv;

set v1;

g= &samsize\*vdphi ;

proc means sum data=uv nway noprint;

var g ;

output out=h1 sum=;

run;

data t;

set partialp3&po;

if &&xvar&po=&qo;

call symput('val', sqtp3zk);run;

data ta;

set secondt;

call symput('secondorder', col1);run;

data hh1;

set h1;

second=&secondorder;

var1s=(1/4)\* (1/(&val\*&val))\* (g+&secondorder) ;

std1s=sqrt(var1s);

run;

data ft;

set hh1;

vstd1s =std1s;

secondor =second;

drop std1s second;

run;

data varfinalp3;

set varfinalp3 ft;

run;

%end;

%end;

%mend varp3;

%varp3;

run;

/\*unconditional partial R-indicator\*/

data allw;data allb; data varu\_level; run;

%macro partial;

%do i=1 %to &variablenoint;

data radd;

set rindicatorall;

run;

data s;

length s $20;

s=&&var&i;

call symput('hvar',s);

run;

proc logistic data=risq.a outest= betasamp covout outdesign=xdfull noprint ;

class &hvar ;

model responsesamp1= &hvar

/ expb ;

output out= predsamp p=phatsamp;

run;

data rindicat ;

set predsamp;

ns=piinv;

os=1;

mmm=1;

rphatsamp=1- phatsamp;

run;

data rindicatora;

set rindicat;

run;

%calcind;

data varc&i;

set finalfilepart;

varc= std\_r/2;

run;

proc means sum mean data= radd nway noprint ;

var ns rphatsamp ;

class &hvar;

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

run;

proc sort data=radd; by &hvar;

data rindicatorall;

merge radd (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 radd;

set rindicatorall;

run;

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

data s;

length s $20;

s=&&vvar&i;

call symput('hvar',s);

run;

proc logistic data=risq.a outest= betasamp covout outdesign=xdfull noprint ;

class &hvar ;

model responsesamp1= &hvar

/ expb ;\* selection=b ;

output out= predsamp p=phatsamp ;

run;

data rindicat ;

set predsamp;

ns=piinv;

os=1;

mmm=1;

rphatsamp= 1-phatsamp;

run;

data rindicatora;

set rindicat;

run;

%calcind;

data varc&j;

set finalfilepart;

varc= std\_r/2;run;

proc means sum mean data= radd nway noprint ;

var ns rphatsamp ;

class &hvar;

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

run;

proc sort data=radd; by &hvar;

data rindicatorall;

merge radd (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 radd;

set rindicatorall;

run;

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

data s;

length s $20;

s=&&strat&t;

call symput('hvar',s);

run;

proc logistic data=risq.a outest= betasamp covout outdesign=xdfull noprint ;

class &hvar ;

model responsesamp1= &hvar

/ expb ;\* selection=b ;

output out= predsamp p=phatsamp ;

run;

data rindicat ;

set predsamp;

ns=piinv;

os=1;

mmm=1;

rphatsamp= 1-phatsamp;

run;

data rindicatora;

set rindicat;

run;

%calcind;

data varc&k;

set finalfilepart;

varc= std\_r/2;run;

proc means sum mean data= radd nway noprint ;

var ns rphatsamp ;

class &hvar;

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

run;

proc sort data=radd; by &hvar;

data rindicatorall;

merge radd (in=v1) ep&k ;

by &hvar;

if v1;

run;

%end;

proc means mean data=rindicatorall 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 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 ;

id mrphatall;

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

run;

data allb;

set allb va ;

%end;

%do kk=1 %to &endd;

data tall;

set rindicatorall;

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 ;

data varu\_level;

set varu\_level varc&kk;

run;

%end;

%mend partial;

%partial;

run;

proc means mean data=rindicatorall noprint;

var rphatsamp;

output out=jk mean(rphatsamp)=mall;

run;

data jjk;

set jk;

call symput('mall',mall);

run;

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

/\*\*\*\*\*\*\*\*\*\*\*\*\* variance of unconditional partial indicator \*\*\*\*\*/

data varfinalp2; run;

%macro varp2;

%do po=1 %to &endd;

%do qo=1 %to &&nvar&po;

data f;

set rindicatorall;

if &&xvar&po=&qo then call symput ('mhat',mrphat&po);

if &&xvar&po=&qo then call symput ('fb',fbar&po);

run;

data g;

set f;

if &&xvar&po=&qo then delta=1; else delta=0;

phi1=piinv\*rphatsamp\*delta/&fb;

phi2new=(piinv\*rphatsamp\*(1-delta))/(&popsize-&fb);

run;

proc means var sum data=g nway noprint ;

var phi1 phi2new;

output out=h var( phi1 phi2new)=

vphi1 vphi2new

sum( phi1 phi2new)= sphi1 sphi2new;

run;

data hh;

set h;

v1=vphi1\*&samsize;

v2new=vphi2new\*&samsize;

run;

data ta;

set partialp2&po;

if &&xvar&po=&qo;

call symput('val', p2zk&po);

run;

data vrl;

set hh;

varfinal =((1-(&fb/&popsize))\*\*2)\* v1+((1-(&fb/&popsize))\*\*2)\*v2new;

sefinal = sqrt(&fb/&popsize)\*sqrt(varfinal );

run;

data varfinalp2;

set varfinalp2 vrl;

%end;

%end;

%mend varp2;

%varp2;

run;

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* bias adjustments \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

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 partialbetween;

set tallw; set tallb;

t=&biasall;

sqrtbetween=sqrt(betweenvar);

sams= &samsize;

ss1=t\*withinvar/(withinvar+betweenvar);

ss2=t\*betweenvar/(withinvar+betweenvar);

betweenbiasadj =betweenvar-ss2;

sqtbetweenbiasadj =sqrt( betweenbiasadj);

run;

data partialwithin;

set tallwr; set tallbr;

t=&biasall;

sqrtwithina =sqrt(withinvara);

sams= &samsize;

ss1=t\*withinvara/(withinvara+betweenvara);

ss2=t\*betweenvara/(withinvara+betweenvara);

withinabiasadj=withinvara-ss1;

sqtwithinabiasadj =sqrt( withinabiasadj);

run;

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

data u1;

set varfinalp2;

if \_type\_ ne .;

SE\_uncond\_cat=sefinal;

keep SE\_uncond\_cat;

run;

data u2;

set varfinalp3;

if \_type\_ ne .;

SE\_cond\_cat =vstd1s;

keep SE\_cond\_cat;

run;

data u3;

set varu\_level;

if bia ne .;

SE\_uncond\_var=varc ;

srvar\_uncond=srvar;

svarb\_uncond=svarb;

\*keep SE\_uncond\_var srvar\_uncond svarb\_uncond ;

keep SE\_uncond\_var;

run;

data u4;

set varc\_level;

if bia ne .;

SE\_cond\_var =varx;

srvar\_cond=srvar;

svarb\_cond=svarb;

\*keep SE\_cond\_var srvar\_cond svarb\_cond;

keep SE\_cond\_var ;

run;

data u5;

set finalfileR\_ind;

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

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

variance\_prop=svarb;

variance\_prop\_adj=srvar;

StdErr\_r=std\_r;

CV\_prop\_adj=cv ;

StdErr\_CV\_adj=std\_cv;

CV\_prop\_unadj=cv\_unadjusted ;

StdErr\_CV\_unadj=std\_cv\_unadjusted;

keep R\_indicator R\_withbias StdErr\_r variance\_prop variance\_prop\_adj lb\_r ub\_r CV\_prop\_adj StdErr\_CV\_adj CV\_prop\_unadj StdErr\_CV\_unadj

avgprop;

run;

data kkk;

set u5;

call symput('srvar',variance\_prop\_adj);

run;

%put &srvar;

data kkk;

set u5;

call symput('svarb',variance\_prop);

run;

%put &svarb;

data f;

set listofpartialbs;

nnn=t;

drop t;

data u6;

set partialbetween;set f;

uncond\_var=betweenvar;

sqrt\_uncond\_var=sqrtbetween;

uncond\_var\_adj=betweenbiasadj;

sqrt\_uncond\_var\_adj=sqtbetweenbiasadj;

namevaru=nnn;

keep uncond\_var sqrt\_uncond\_var uncond\_var\_adj sqrt\_uncond\_var\_adj namevaru ;

run;

data ff;

set listofpartialws;

nnn=t;

drop t;

data u7;

set partialwithin; set ff;

cond\_var=withinvara;

sqrt\_cond\_var=sqrtwithina;

cond\_var\_adj=withinabiasadj;

sqrt\_cond\_var\_adj=sqtwithinabiasadj;

namevaru=nnn;

keep cond\_var sqrt\_cond\_var cond\_var\_adj sqrt\_cond\_var\_adj namevaru;

run;

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

data g;run;

%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 partialp3&i;

run;

data g;

set g a;

%end;

%mend pr;

%pr;

run;

data u8;

set g;

if p3zk ne .;

cond\_cat=p3zk;

sqrt\_cond\_cat=sqtp3zk;

sampsize=ns;

popsize=ps;

drop p3zk sqtp3zk ns ps ;

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 aa;

set partialp2&i;

popsize=fbar&i;

avg\_propensity\_cat=mrphat&i;

avg\_propensity=mrphatall;

uncond\_cat=p1zk&i;

sqrt\_uncond\_cat=p2zk&i;

drop fbar&i mrphat&i p1zk&i p2zk&i mrphatall;

run;

data f;

set f aa;

%end;

%mend pr;

%pr;

run;

data u9;

set f;

if \_type\_ ne .;

drop \_type\_ \_freq\_;

run;

data final1;

set u6; set u3;

row+1;

data final3;

set u7; set u4;

row+1;

data tem;

merge final1 (in=v1) final3 (in=v2);

by row;

if v1;

run;

data ffinal1 ;

set tem;

cvu=sqrt\_uncond\_var/&htmean;

cvc=sqrt\_cond\_var/&htmean;

cvu\_adj=sqrt\_uncond\_var\_adj/&htmean;

cvc\_adj=sqrt\_cond\_var\_adj/&htmean;

avgpr=&htmean;

variance\_cvu=((se\_uncond\_var\*se\_uncond\_var)/(avgpr\*avgpr))+ ((sqrt\_uncond\_var\*sqrt\_uncond\_var\*&srvar\*&srvar)/(&samsize\*(avgpr\*\*4)));

std\_cvu=sqrt(variance\_cvu);

variance\_cvc=((se\_cond\_var\*se\_cond\_var)/(avgpr\*avgpr))+ ((sqrt\_cond\_var\*sqrt\_cond\_var\*&srvar\*&srvar)/(&samsize\*(avgpr\*\*4)));

std\_cvc=sqrt(variance\_cvc);

drop row avgpr variance\_cvu variance\_cvc;

run;

data final2;

set u9; set u1;

row+1;

data final4;

set u8; set u2;

row+1;

data tem;

merge final2 (in=v1) final4 (in=v2);

by row;

if v1;

run;

data ffinal2 ;

set tem;

cvu\_category=sqrt\_uncond\_cat/&htmean;

cvc\_category=sqrt\_cond\_cat/&htmean;

avgpr=&htmean;

variance\_cvu\_category=((se\_uncond\_cat\*se\_uncond\_cat)/(avgpr\*avgpr))+ ((sqrt\_uncond\_cat\*sqrt\_uncond\_cat\*&srvar\*&srvar)/(&samsize\*(avgpr\*\*4)));

std\_cvu\_category=sqrt(variance\_cvu\_category);

variance\_cvc\_category=((se\_cond\_cat\*se\_cond\_cat)/(avgpr\*avgpr))+ ((sqrt\_cond\_cat\*sqrt\_cond\_cat\*&srvar\*&srvar)/(&samsize\*(avgpr\*\*4)));

std\_cvc\_category=sqrt(variance\_cvc\_category);

drop row avgpr variance\_cvu\_category variance\_cvc\_category;

run;

/\*\*\*\*\* final output file - names and directory of output can be changed here \*\*\*\*\*\*\*\*\*\*/

data risq.final\_output\_ex2;

set u5 ffinal1 ffinal2 ;

run;

PROC EXPORT DATA= RISQ.FINAL\_OUTPUT\_EX2

OUTFILE= "F:\Documents\risq\risq-test\finaloutputex2.csv"

DBMS=CSV REPLACE;

PUTNAMES=YES;

RUN;