**data** a;

input prob cond$ v1 p1 va1 v2 p2 va2 v3 p3 va3 oc1-oc3;;

CARDS;

501 exp1 3 .45 0 2 .97 -20 1.5 .94 -20 .43 .50 .07

502 exp1 3 .45 0 1.6 .97 -20 1.5 .94 -20 .58 .36 .06

503 exp1 3 .45 0 1.2 .97 -20 1.5 .94 -20 .63 .14 .23

504 exp2 .6 1 0 1.6 .97 -20 1.5 .94 -20 .58 .36 .06

505 exp2 .6 1 0 1.2 .97 -20 1.5 .94 -20 .63 .14 .23

506 exp2 1.35 1 0 1.6 .97 -20 1.5 .94 -20 .58 .36 .06

507 exp2 1.35 1 0 1.2 .97 -20 1.5 .94 -20 .63 .14 .23

;

**data** a; set a; nsim=**2000**;

model='naive';\*'AOD';\*isaw2;\*full;\*naive; unip=**1**;

mopar='k46';

samples=**2**;

**data** a; set a; accu=**1**;

do kap= **8**;\*5;

do nu=**.9**;

do err=**0**;

do risk1=**2**;\*.8;

do wave2=**.5**;

do new1=**1.97**;

do psi1=**.59**;

do alfa1=**.06**;

do gama1=**.34**;

do teta1=**1.3**;

output;

end; end; end; end; end; end; end; end; end; end;

**data** a; set a;

wave1=**0**;

if model='full' then do; kap= **1**; wave1=**1**; samples=**1**; end;

if model='naive' then do; kap= **9**; wave1=**0**; samples=**1**; end;

if model='naive' and mopar='k46' then do; kap= **46**; wave1=**0**; samples=**1**; end;

array p{**3**} p1-p3;

array v{**3**} v1-v3;

array va{**3**} va1-va3;

array sum{**3**} sum1-sum3;

array rsum{**3**} rsum1-rsum3;

array rave{**3**} rave1-rave3;

array minn{**3**} minn1-minn3;

array maxxi{**3**} maxx1-maxx3;

array rssq{**3**} rssq1-rssq3;

array his{**3**, **100**} his1-his300;

array c{**4**,**3**} c1-c12;

array vv{**3**} vv1-vv3;

array stnd{**3**} std1-std3;

array uniq{**3**} uniq1-uniq3;

array nnu{**3**} nnu1-nnu3;

array ff{**3**} ff1-ff3;

array ss{**3**} ss1-ss3;

array qq{**3**} qq1-qq3;

array pp{**3**} pp1-pp3;

array saod{**3**} saod1-saod3;

nt=**100**;

nblk=**4**; blk=**1**; inblk=nt/nblk;

do blk=**1** to nblk; do i=**1** to **3**; c{blk,i}=**0**; end; end;

prep=**0**;

do sim=**1** to nsim;

condi=round(**.5**+ranuni(**0**)\***5**);

if condi=**1** then do; teta1=**1.34**; alfa1= **.06**; gama1=**.34**; psi1= **0.59**; new1= **1.97**; end;

if condi=**2** then do; teta1=**0.23**; alfa1= **.04**; gama1=**1.09**; psi1= **0.21**; new1= **1.33**; end;

if condi=**3** then do; teta1=**0.76**; alfa1= **.02**; gama1=**0.76**; psi1= **0.44**; new1= **0.63**; end;

if condi=**4** then do; teta1=**9.79**; alfa1= **.05**; gama1=**0.37**; psi1= **0.93**; new1= **0.94**; end;

if condi=**5** then do; teta1=**1.33**; alfa1= **.08**; gama1=**0.51**; psi1= **0.98**; new1= **1.13**; end;

blk=**1**;

do i=**1** to **3**; rave{i}=rannor(**0**)/**100000000**; rsum{i}=rannor(**0**)/**100000000**;

minn{i}=**1000**; stnd{i}=**0**; maxxi{i}=-**1000**;

rssq{i}=**0**; uniq{i}=**0**; nnu{i}=**0**;

qq{i}=**0**;

end;

wave = wave2\*ranuni(**0**);

kapa = round(**.5**+ranuni(**0**)\*kap);

nui= ranuni(**0**)\*nu; if nu>**1** then nui=**1**;

risk1i= ranuni(**0**)\*risk1; if risk1>**2** then risk1i=**1**;

erri=ranuni(**0**)\*err;

last=round(**.5**+ranuni(**0**)\***3**);

safe=round(**.5**+ranuni(**0**)\***3**);

new= new1\*(**1**-unip)+unip\*new1\***2**\*ranuni(**0**);

psi=psi1\*(**1**-unip)+unip\*psi1\***2**\*ranuni(**0**);

alfa=alfa1\*(**1**-unip)+unip\*alfa1\***2**\*ranuni(**0**);

gama=gama1\*(**1**-unip)+unip\*gama1\***2**\*ranuni(**0**);

teta=teta1\*(**1**-unip)+unip\*teta1\***2**\*ranuni(**0**);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

do t=**1** to nt;

if model='AOD' then do;

sumdom=**0**;

maxq=max(qq1,qq2,qq3);

do i=**1** to **3**; sumdom=sumdom+exp((qq{i}-maxq)\*teta); end;

do i=**1** to **3**; pp{i}= exp((qq{i}-maxq)\*teta)/sumdom; end;

rnt=ranuni(**0**);

dd2=**1**; if rnt>pp1 then dd2=**2**; if rnt>pp1+pp2 and cond ne '2alt' then dd2=**3**;

end;\*\*of AOD;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*ISAW2\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

if model='isaw2' then do;

\*\*\*\*\*inertia;

rndnu=ranuni(**0**);

if rndnu>nui then dd2=last;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*in not in an inertia mode\*\*\*\*\*\*\*\*\*\*\*\*\*;

if rndnu=<nui then do;

\*\*finding safe based on a small sample \*\*\*\*\*\*\*;

minna=**10000000**;

safe=round(**.5**+ranuni(**0**)\***3**);

if ranuni(**0**)<risk1i and t>**1** then do i=**1** to **3**;

tmp=stnd{i}+rannor(**0**)/**10000000000**;

if tmp<minna then do; minna=tmp; safe=i; end;

end;

\*\*first sample;

do i=**1** to **3**; sum{i}=**0**; end;

do j=**1** to kapa; rnd1=round(**.5**+ranuni(**0**)\*(t-**1**));

do i=**1** to **3**; if i ne safe then do;

if t>**1** then sum{i}=sum{i}+his{i,rnd1}/kapa;

if t=**1** then sum{i}=sum{i}+rannor(**0**)/**100**;

end; end; \*of i;

end; \*\*of kapa;

minna=-**10000**;

do i=**1** to **3**; if i ne safe then do;

if wave1\*rave{i}+(**1**-wave1)\*sum{i}>minna then do; minna=wave1\*rave{i}+(**1**-wave1)\*sum{i}; ifc=i; end;

vvc=sum{ifc};

end; end;

\*\*second sample;

do i=**1** to **3**; sum{i}=**0**; end;

do j=**1** to kapa; rnd2=round(**.5**+ranuni(**0**)\*(t-**1**)); ia=min(safe,ifc); ib=max(safe,ifc);

do i=ia, ib;

if t>**1** then do; sum{i}=sum{i}+his{i,rnd2}/kapa; ; end;

if t=**1** then do; sum{i}=sum{i}+rannor(**0**)/**10000**; end;

end; \*of i;

end; \*\*of kapa;

dd2=ifc;

if wave\*rave{ifc}+(**1**-wave)\*mean(vvc,sum{ifc})<wave\*rave{safe}+(**1**-wave)\*sum{safe}+rannor(**0**)/**10000000** then dd2=safe; \*end;

end;\*\*of not in inertia;

end; \*\*isaw2;

\*\*one sample;

if model='naive' or model='full' then do;

do i=**1** to **3**; sum{i}=**0**; end;

do j=**1** to kapa; rnd12=round(**.5**+ranuni(**0**)\*(t-**1**));

do i=**1** to **3**;

if t>**1** then do; sum{i}=sum{i}+his{i,rnd12}/kapa; ; end;

if t=**1** then do; sum{i}=sum{i}+rannor(**0**)/**10000**; end;

end; \*\*of i;

end; \*\*of j;

maxx=-**100**;

do i=**1** to **3**;

if (**1**-wave1)\*sum{i}+wave1\*rave{i}>maxx then do;

maxx=(**1**-wave1)\*sum{i}+wave1\*rave{i}; dd2=i;

end;

end;

end; \*\*full or naive;

\*end;\* of new dec;

rep=**0**; if dd2=last then rep=**1**;

last=dd2;

\*end;\*\*\*\*\*end of naive or full ;

if ranuni(**0**)<erri then dd2=round(**.5**+ranuni(**0**)\***3**);

\*\*\*stat\*\*\*\*\*\*\*\*\*\*\*\*\*;

if t>**0** then do i=**1** to **3**; if dd2=i then c{blk,i}=c{blk,i}+**1**/(nt\*nsim/nblk); end;

if t>**1** then prep=prep+rep/(nt\*nsim);

\*\*\*\*\*;

\*\*\*\*\*\*\*\*\*\*\*\*\*outcomes\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*;

ranuni1=ranuni(**0**);

ranuni2=ranuni(**0**);

ranuni3=ranuni2;

vv1=v1; if ranuni1>p1 then vv1=va1;

vv2=v2; if ranuni2>p2 then vv2=va2;

vv3=v3; if ranuni3>p3 then vv3=va3;

\*\*\*AOD stat;

if model='AOD' then do;

sumff=**0**;

do i=**1** to **3**;

if vv{i}>=**0** then ff{i}=vv{i}\*\*gama;

if vv{i}<**0** then ff{i}=-(abs(vv{i}))\*\*gama;

sumff=sumff+abs(ff{i})/**3**;

end;

ss1=(exp(-psi\*(abs(ff1-ff2)))+ exp(-psi\*(abs(ff1-ff3))))/**2**;

ss2=(exp(-psi\*(abs(ff1-ff2)))+ exp(-psi\*(abs(ff2-ff3))))/**2**;

ss3=(exp(-psi\*(abs(ff3-ff2)))+ exp(-psi\*(abs(ff1-ff3))))/**2**;

do i=**1** to **3**;

saod{i}=ff{i}-new\*ss{i}\*sumff;

qq{i}=qq{i}+alfa\*(saod{i}-qq{i});

end;

end;\*\* of aod;

\*\*\*ISAW2 naive full memory;

if model='isaw2' or model='full' or model='naive' then do;

do i=**1** to **3**; his{i,t}=vv{i}; rsum{i}=rsum{i}+vv{i};end;

do i=**1** to **3**; rave{i}=rsum{i}/t ;end;

do i=**1** to **3**; if vv{i}<minn{i} then minn{i}=vv{i} ;end;

do i=**1** to **3**; rssq{i}=rssq{i}+(rave{i}-vv{i})\*\***2**; end;

do i=**1** to **3**; stnd {i}=((rssq{i})\*\***.5**)/t;\*-mean(minn{i},rave{i}); end;

end; \*\*of isaw2 full naive;

do i=**1** to **3**; if vv{i}>maxxi{i} then maxxi{i}=vv{i} ;end;

if (t/inblk) =round(t/inblk) then blk=blk+**1**;

end; \*\*of t;

end;\*\*\*of sim;

sumc=c1+c2+c3;

msd=((c1-oc1)\*\***2**+(c2-oc2)\*\***2**+(c3-oc3)\*\***2**)/**3**;

zz=**.**;

ev1=v1\*p1+(**1**-p1)\*va1;

ev2=v2\*p2+(**1**-p2)\*va2;

ev3=v3\*p3+(**1**-p3)\*va3;

drop his1-his100;

**proc** **print** round;

**proc** **sort** out=b; by wave1 wave2 kap nu risk1;

**proc** **means** noprint; by wave1 wave2 kap nu risk1;

var msd;

output out=o mean=msd;

**proc** **sort**; by msd;

**proc** **print** data=o;

**run**;

**data** a; set a;

array c{**4**,**3**} c1-c12;

array bs{**3**,**4**} saf1-saf4 lr1-lr4 hr1-hr4;

array ac{**4**} acc1-acc4;

do b=**1** to **4**; do s=**1** to **3**;

bs{s,b}=c{b,s};

end; end;

do b=**1** to **4**;

ac{b}=**100**\*(**.03**\*bs{**2**,b}+**.06**\*bs{**3**,b});

end;

**proc** **sort** data=a; by cond prob;

**proc** **print** data =a round;

var cond prob cond v1 p1 va1 v2 p2 va2 v3 p3 va3 safe c1-c3 ev1-ev3 zz oc1-oc3 prep rep msd;

**proc** **print** data =a round;

var v1 p1 va1 v2 p2 va2 v3 p3 va3 saf1-saf4 lr1-lr4 hr1-hr4 acc1-acc4;

**run**;

**data** o1; input prob mcond tax oc1 oc2 oc3 oc4 oc5 oc6 oc7 oc8 oc9 oc10 oc11 oc12;

cards;

501 . 0.0 0.38259 0.40047 0.44329 0.49271 0.51294 0.54259 0.50682 0.45459 0.10447 0.05694 0.04988 0.05271

502 . 0.4 0.46965 0.55388 0.61976 0.68753 0.41647 0.40235 0.33459 0.27059 0.11388 0.04376 0.04565 0.04188

503 . 0.8 0.52329 0.63341 0.66447 0.71341 0.14071 0.14024 0.14918 0.12329 0.33600 0.22635 0.18635 0.16329

;

**data** o1; set o1;

array pr{**4**,**3**} pp1-pp12;

array oo{**3**,**4**} oc1-oc12;

array tpr{**12**} pp1-pp12;

array too{**12**} oc1-oc12;

msd=**0**;

do j=**1** to **4**; do k=**1** to **3**; pr{j,k}=oo{k,j};end; end;

do i=**1** to **12**; too{i}=tpr{i}; end;

keep prob mcond tax oc1 oc2 oc3 oc4 oc5 oc6 oc7 oc8 oc9 oc10 oc11 oc12;

**data** o2; input prob mcond tax oc1 oc2 oc3 oc4 oc5 oc6 oc7 oc8 oc9 oc10 oc11 oc12;

cards;

504 0.60 0.4 0.14698 0.67442 0.17860 0.23163 0.64279 0.12558 0.25860 0.64047 0.10093 0.34000 0.57674 0.08326

505 0.60 0.8 0.12977 0.21070 0.65953 0.22744 0.24512 0.52744 0.30047 0.21442 0.48512 0.38093 0.16605 0.45302

506 1.35 0.4 0.20533 0.69280 0.10187 0.35467 0.60267 0.04267 0.41867 0.53493 0.04640 0.50400 0.44320 0.05280

507 1.35 0.8 0.27733 0.15307 0.56960 0.49653 0.12267 0.38080 0.56800 0.10347 0.32853 0.60533 0.10773 0.28693

;

**data** o; set o1 o2;

**data** m; merge a o; by prob;

array cc{**4**,**3**} c1-c12;

array oo{**4**,**3**} oc1-oc12;

msd=**0**;

do j=**1** to **4**; do k=**1** to **3**;

msd=msd+(**1**/**12**)\***1**\*(oo{j,k}-cc{j,k})\*\***2**;

end; end;

**proc** **print**;

var prob tax oc1-oc12 c1-c12 msd;

**proc** **means**; by cond; var msd;

**run**;