set matsize 10000

capture log close

log using Appendix, text replace

/\* Replicate results in Table 1 \*/

use "AF\_data\_standard.dta", clear

\* List of explanatory variables including ASC

global xlist1 asc\_t stroke\_risk\_d bleed\_risk\_d antidote bloodtest dosefreq ///

drug\_int cost bleed\_risk\_d\_antidote block\_asc\_t

\* List of explanatory variables excluding ASC

global xlist2 stroke\_risk\_d bleed\_risk\_d antidote bloodtest dosefreq ///

drug\_int cost bleed\_risk\_d\_antidote block\_asc\_t

\* Estimate MNL model

clogit c1 $xlist1, group(set\_id\_n) vce(cluster id)

estimates store MNL

\* Estimate MXL model with normally distributed intercept

mixlogit c1 $xlist2, group(set\_id\_n) id(id) rand(asc\_t) robust

estimates store MXL

\* Estimate SH model

gmnl c1 $xlist1, group(set\_id\_n) id(id) vce(robust)

\* Estimate G-MNL model with normally distributed intercept

gmnl c1 $xlist2, group(set\_id\_n) id(id) rand(asc\_t) gamma(0) vce(robust)

\* Estimate ROL model using rologit without exploding data

gen rankchoice=1

replace rankchoice=2 if c2==1

replace rankchoice=3 if c1==1

rologit rankchoice $xlist1, group(set\_id\_n) vce(cluster id)

\* Estimate ROL model using clogit after exploding data

use "AF\_data\_exploded.dta", clear

clogit c1 $xlist1, group(set\_id\_n) vce(cluster id)

\* Estimate MROL model

mixlogit c1 $xlist2, group(set\_id\_n) id(id) rand(asc\_t) robust

estimates store MROL

/\* Replicate results in Table 2 \*/

\* Calculate WTP estimates using MNL results

estimates restore MNL

nlcom (stroke:-\_b[stroke\_risk\_d]/\_b[cost]) ///

(bleed\_noant:-\_b[bleed\_risk\_d]/\_b[cost]) ///

(bleed\_ant:-(\_b[bleed\_risk\_d]+\_b[bleed\_risk\_d\_antidote])/\_b[cost])

\* Calculate WTP estimates using MXL results

estimates restore MXL

nlcom (stroke:-\_b[stroke\_risk\_d]/\_b[cost]) ///

(bleed\_noant:-\_b[bleed\_risk\_d]/\_b[cost]) ///

(bleed\_ant:-(\_b[bleed\_risk\_d]+\_b[bleed\_risk\_d\_antidote])/\_b[cost])

\* Calculate WTP estimates using MROL results

estimates restore MROL

nlcom (stroke:-\_b[stroke\_risk\_d]/\_b[cost]) ///

(bleed\_noant:-\_b[bleed\_risk\_d]/\_b[cost]) ///

(bleed\_ant:-(\_b[bleed\_risk\_d]+\_b[bleed\_risk\_d\_antidote])/\_b[cost])

/\* Replicate results in Table 3 \*/

use "AF\_data\_standard.dta", clear

\* Create new observations for predictive analysis

expand 2 if set\_id\_n <104, gen(newobs)

replace stroke\_risk\_d=0 if newobs==1

replace bleed\_risk\_d=0 if newobs==1

replace antidote=0 if newobs==1

replace bloodtest=0 if newobs==1

replace dosefreq=0 if newobs==1

replace drug\_int=0 if newobs==1

replace cost=0 if newobs==1

replace bleed\_risk\_d\_antidote=0 if newobs==1

replace block\_asc\_t=0 if newobs==1

replace cost=50 if (newobs==1 & alt==2 & set\_id\_n==102)

replace stroke\_risk\_d=1 if newobs==1 & alt==2 & set\_id\_n==103

\* MNL predictions

estimates restore MNL

predict mnlpred if newobs==1

list c1 alt mnlpred if newobs==1, sep(3) ab(16)

\* Confidence intervals

matrix b = e(b)

matrix V = e(V)

global coeflist1 b\_asc\_t b\_stroke\_risk\_d b\_bleed\_risk\_d b\_antidote b\_bloodtest ///

b\_dosefreq b\_drug\_int b\_cost b\_bleed\_risk\_d\_antidote b\_block\_asc\_t

// Generate 1000 simulated coefficient vectors using the estimated betas and their variance

preserve

quietly drawnorm $coeflist1, means(b) cov(V) n(1000) seed(12345) clear

quietly mkmat $coeflist1, matrix(bsim)

restore

set seed 12345

forvalues i = 1(1)1000 {

matrix brep = bsim[`i',1..colsof(bsim)]

// Generate predictions using the simulated coefficients

quietly clogit c1 $xlist1 if newobs==0, group(set\_id\_n) from(brep, copy) iter(0)

quietly predict mnlpred`i' if newobs==1

}

egen mnlpred\_ci\_lower = rowpctile(mnlpred1-mnlpred1000), p(2.5)

egen mnlpred\_ci\_upper = rowpctile(mnlpred1-mnlpred1000), p(97.5)

drop mnlpred1-mnlpred1000

list c1 alt mnlpred mnlpred\_ci\_lower mnlpred\_ci\_upper if newobs==1, sep(3) ab(16)

\* MXL predictions

estimates restore MXL

mixlpred mxlpred if newobs==1

list c1 alt mnlpred if newobs==1, sep(3) ab(16)

\* Confidence intervals - takes several minutes to run

matrix b = e(b)

matrix V = e(V)

global coeflist2 b\_stroke\_risk\_d b\_bleed\_risk\_d b\_antidote b\_bloodtest b\_dosefreq ///

b\_drug\_int b\_cost b\_bleed\_risk\_d\_antidote b\_block\_asc\_t b\_asc\_t sd\_asc\_t

// Generate 1000 simulated coefficient vectors using the estimated betas and their variance

preserve

quietly drawnorm $coeflist2, means(b) cov(V) n(1000) seed(12345) clear

quietly mkmat $coeflist2, matrix(bsim)

restore

forvalues i = 1(1)1000 {

matrix brep = bsim[`i',1..colsof(bsim)]

// Generate predictions using the simulated coefficients

quietly mixlogit c1 $xlist2 if newobs==0, group(set\_id\_n) id(id) rand(asc\_t) from(brep, copy) iter(0) nrep(1)

quietly mixlpred mxlpred`i' if newobs==1

}

egen mxlpred\_ci\_lower = rowpctile(mxlpred1-mxlpred1000), p(2.5)

egen mxlpred\_ci\_upper = rowpctile(mxlpred1-mxlpred1000), p(97.5)

drop mxlpred1-mxlpred1000

list c1 alt mxlpred mxlpred\_ci\_lower mxlpred\_ci\_upper if newobs==1, sep(3) ab(16)

\* MROL predictions

estimates restore MROL

mixlpred mrolpred if newobs==1

list c1 alt mrolpred if newobs==1, sep(3) ab(16)

\* Confidence intervals - takes several minutes to run

matrix b = e(b)

matrix V = e(V)

global coeflist2 b\_stroke\_risk\_d b\_bleed\_risk\_d b\_antidote b\_bloodtest b\_dosefreq ///

b\_drug\_int b\_cost b\_bleed\_risk\_d\_antidote b\_block\_asc\_t b\_asc\_t sd\_asc\_t

// Generate 1000 simulated coefficient vectors using the estimated betas and their variance

preserve

quietly drawnorm $coeflist2, means(b) cov(V) n(1000) seed(12345) clear

quietly mkmat $coeflist2, matrix(bsim)

restore

forvalues i = 1(1)1000 {

matrix brep = bsim[`i',1..colsof(bsim)]

// Generate predictions using the simulated coefficients

// Note: exploding data is not necessary here as no estimation is carried out

quietly mixlogit c1 $xlist2 if newobs==0, group(set\_id\_n) id(id) rand(asc\_t) from(brep, copy) iter(0) nrep(1)

quietly mixlpred mrolpred`i' if newobs==1

}

egen mrolpred\_ci\_lower = rowpctile(mrolpred1-mrolpred1000), p(2.5)

egen mrolpred\_ci\_upper = rowpctile(mrolpred1-mrolpred1000), p(97.5)

drop mrolpred1-mrolpred1000

list c1 alt mrolpred mrolpred\_ci\_lower mrolpred\_ci\_upper if newobs==1, sep(3) ab(16)

capture log close

/\* Replicate results in Table 5 \*/

\*\*\* NLOGIT code \*\*\*

\* The NLOGIT variables are named as follows:

\* stroke = stroke\_risk\_d

\* bleed = bleed\_risk\_d

\* adote = antidote

\* btest = bloodtest

\* dfreq = dosefreq

\* d\_int = drug\_int

\* b\_asc\_t = block\_asc\_t

\* Note that, unlike Stata, variable names in NLOGIT are not case sensitive

RPLOGIT ; Lhs = C1 ; Choices = A,B,C

; Model: U(A,B,C) = asc\_t \* ASC\_T + stroke \* STROKE +

bleed \* BLEED + adote \* ADOTE + btest \* BTEST + dfreq\* DFREQ +

d\_int \* D\_INT + cost \* COST + b\_asc\_t \* B\_ASC\_T

; Fcn = asc\_t(n)

; Halton ; Pds = 16 ; Pts = 500 $

\*\*\* Stata code \*\*\*

use "AF\_data\_standard.dta", clear

mixlogit c1 stroke\_risk\_d bleed\_risk\_d antidote bloodtest dosefreq ///

drug\_int cost block\_asc\_t, group(set\_id\_n) id(id) rand(asc\_t) nrep(500)

\*\*\* Biogeme code \*\*\*

\* Note that the dataset needs to be in wide form to be used by Biogeme.

\* For each attribute there are now three variables, one for each alternative.

\* For example the stroke variable in long form becomes stroke1, stroke2

\* and stroke3 in wide form. Otherwise the variable names are the same as

\* for NLOGIT (see above).

\* The dependent variable "choice" is coded 1, 2 or 3, indicating

\* the chosen alternative

\* av1, av2, and av3 are all constants equal to 1, which specify that all

\* respondents could choose between the full set of three alternatives

[ModelDescription]

"MXL model"

[Choice]

choice

[Beta]

// Name Value LowerBound UpperBound status (0=variable, 1=fixed)

asc\_t 0.964 -10000 10000 0

stroke 0.626 -10000 10000 0

bleed 0.509 -10000 10000 0

adote 0.545 -10000 10000 0

btest -0.0720 -10000 10000 0

dfreq -0.0756 -10000 10000 0

d\_int -0.288 -10000 10000 0

cost -0.0113 -10000 10000 0

b\_asc\_t -0.605 -10000 10000 0

sd\_asc\_t 0.1 -10000 10000 0

[Utilities]

// Id Name Avail linear-in-parameter expression (beta1\*x1 + beta2\*x2 + ... )

1 Alt1 av1 asc\_t [ sd\_asc\_t ] \* asc\_t1 + stroke \* stroke1 +

bleed \* bleed1 + adote \* adote1 + btest \* btest1 + dfreq \* dfreq1 +

d\_int \* d\_int1 + cost \* cost1 + b\_asc\_t \* b\_asc\_t1

2 Alt2 av2 asc\_t [ sd\_asc\_t ] \* asc\_t2 + stroke \* stroke2 +

bleed \* bleed2 + adote \* adote2 + btest \* btest2 + dfreq \* dfreq2 +

d\_int \* d\_int2 + cost \* cost2 + b\_asc\_t \* b\_asc\_t2

3 Alt3 av3 asc\_t [ sd\_asc\_t ] \* asc\_t3 + stroke \* stroke3 +

bleed \* bleed3 + adote \* adote3 + btest \* btest3 + dfreq \* dfreq3 +

d\_int \* d\_int3 + cost \* cost3 + b\_asc\_t \* b\_asc\_t3

[Draws]

500

[PanelData]

id

asc\_t\_sd\_asc\_t

[Model]

// Currently, only $MNL (multinomial logit), $NL (nested logit), $CNL

// (cross-nested logit) and $NGEV (Network GEV model) are valid keywords

//

$MNL