The Stata Journal (2007)
7, Number 3, pp. 388–401

Fitting mixed logit models by using maximum simulated likelihood

Arne Risa Hole
National Primary Care Research and Development Centre
Centre for Health Economics
University of York
York, UK
ah522@york.ac.uk

Abstract. This article describes the `mixlogit` Stata command for fitting mixed logit models by using maximum simulated likelihood.

Keywords: st0133, mixlogit, mixlpred, mixlcov, mixed logit, maximum simulated likelihood

1 Introduction

In a recent issue of the *Stata Journal* devoted to maximum simulated likelihood estimation, Haan and Uhlendorff (2006) showed how to implement a multinomial logit model with unobserved heterogeneity in Stata. This article describes the `mixlogit` Stata command, which can be used to fit models of the type considered by Haan and Uhlendorff, as well as other types of mixed logit models (Train 2003).

The article is organized as follows: section 2 gives a brief overview of the mixed logit model, section 3 describes the `mixlogit` syntax and options, and section 4 presents some examples.

2 Mixed logit model

Per Revelt and Train (1998), we assume a sample of N respondents with the choice of J alternatives on T choice occasions. The utility that individual n derives from choosing alternative j on choice occasion t is given by $U_{njt} = \beta_n' x_{njt} + \varepsilon_{njt}$, where β_n is a vector of individual-specific coefficients, x_{njt} is a vector of observed attributes relating to individual n and alternative j on choice occasion t, and ε_{njt} is a random term that is assumed to be an independently and identically distributed extreme value. The density for β is denoted as $f(\beta|\theta)$, where θ are the parameters of the distribution. Conditional on knowing β_n, the probability of respondent n choosing alternative i on choice occasion t is given by

$$L_{nit}(\beta_n) = \frac{\exp(\beta_n' x_{nit})}{\sum_{j=1}^{J} \exp(\beta_n' x_{njt})}$$

which is the conditional logit formula (McFadden 1974). The probability of the observed sequence of choices conditional on knowing β_n is given by

$$S_n(\beta_n) = \prod_{t=1}^{T} L_{ni(n,t)t}(\beta_n)$$

where $i(n,t)$ denotes the alternative chosen by individual n on choice occasion t. The *unconditional* probability of the observed sequence of choices is the conditional probability integrated over the distribution of β:

$$P_n(\theta) = \int S_n(\beta) f(\beta|\theta) d\beta$$

The unconditional probability is thus a weighted average of a product of logit formulas evaluated at different values of β, with the weights given by the density f.

This specification is general because it allows fitting models with both *individual-specific* and *alternative-specific* explanatory variables. This is analogous to the way that the `clogit` command (see [R] **clogit**) can be used to fit multinomial logit models. In section 4, I show how `mixlogit` can fit various models, including the multinomial logit model with unobserved heterogeneity considered by Haan and Uhlendorff (2006).

The log likelihood for the model is given by $\mathrm{LL}(\theta) = \sum_{n=1}^{N} \ln P_n(\theta)$. This expression cannot be solved analytically, and it is therefore approximated using simulation methods (see Train 2003). The simulated log likelihood is given by

$$\mathrm{SLL}(\theta) = \sum_{n=1}^{N} \ln \left\{ \frac{1}{R} \sum_{r=1}^{R} S_n(\beta^r) \right\}$$

where R is the number of replications and β^r is the the rth draw from $f(\beta|\theta)$.

3 Commands

3.1 mixlogit

Syntax

`mixlogit` *depvar* [*indepvars*] [*if*] [*in*] , `group(`*varname*`)` `rand(`*varlist*`)`
[`id(`*varname*`)` `ln(`*#*`)` `corr` `nrep(`*#*`)` `burn(`*#*`)` `level(`*#*`)`
`constraints(`*numlist*`)` *maximize_options*]

Description

`mixlogit` is implemented as a `d0 ml` evaluator. The command allows correlated and uncorrelated normal and lognormal distributions for the coefficients. The pseudorandom draws used in the estimation process are generated using the Mata function `halton()` (Drukker and Gates 2006).

Options

`group(`*varname*`)` is required and specifies a numeric identifier variable for the choice occasions.

`rand(`*varlist*`)` is required and specifies the independent variables whose coefficients are random. The random coefficients can be specified to be normally or lognormally distributed (see the `ln()` option). The variables immediately following the dependent variable in the syntax are specified to have fixed coefficients.

`id(`*varname*`)` specifies a numeric identifier variable for the decision makers. This option should be specified only when each individual performs several choices; i.e., the dataset is a panel.

`ln(`#`)` specifies that the last # variables in `rand()` have lognormally rather than normally distributed coefficients. The default is `ln(0)`.

`corr` specifies that the random coefficients are correlated. The default is that they are independent. When the `corr` option is specified, the estimated parameters are the means of the (fixed and random) coefficients plus the elements of the lower-triangular matrix $\mathbf{L}$, where the covariance matrix for the random coefficients is given by $\mathbf{V} = \mathbf{L}\mathbf{L}'$. The estimated parameters are reported in the following order: the means of the fixed coefficients, the means of the random coefficients, and the elements of the $\mathbf{L}$ matrix. The `mixlcov` command can be used postestimation to obtain the elements in the $\mathbf{V}$ matrix along with their standard errors.

If the `corr` option is not specified, the estimated parameters are the means of the fixed coefficients and the means and standard deviations of the random coefficients, reported in that order. The sign of the estimated standard deviations is irrelevant. Although in practice the estimates may be negative, interpret them as being positive.

The sequence of the parameters is important to bear in mind when specifying starting values.

`nrep(`#`)` specifies the number of Halton draws used for the simulation. The default is `nrep(50)`.

`burn(`#`)` specifies the number of initial sequence elements to drop when creating the Halton sequences. The default is `burn(15)`. Specifying this option helps reduce the correlation between the sequences in each dimension. Train (2003, 230) recommends that # should be at least as large as the prime number used to generate the sequences. If there are K random coefficients, `mixlogit` uses the first K primes to generate the Halton draws.

`level(`#`)`; see [R] **estimation options**.

`constraints(`*numlist*`)`; see [R] **estimation options**.

maximize_options: difficult, technique(*algorithm_spec*), iterate(#), trace, gradient, showstep, hessian, tolerance(#), ltolerance(#), gtolerance(#), nrtolerance(#), from(*init_specs*); see [R] **maximize**. technique(bhhh) is not allowed.

3.2 mixlpred

Syntax

mixlpred *newvarname* [*if*] [*in*] [, nrep(#) burn(#)]

Description

The command mixlpred can be used following mixlogit to obtain predicted probabilities. The predictions are available both in and out of sample; type mixlpred ... if e(sample) ... if predictions are wanted for the estimation sample only.

Options

nrep(#) specifies the number of Halton draws used for the simulation. The default is nrep(50).

burn(#) specifies the number of initial sequence elements to drop when creating the Halton sequences. The default is burn(15).

3.3 mixlcov

Syntax

mixlcov [, sd]

Description

The command mixlcov can be used following mixlogit to obtain the elements in the coefficient covariance matrix along with their standard errors. This command is relevant only when the coefficients are specified to be correlated; see the corr option above. mixlcov is a wrapper for nlcom (see [R] **nlcom**).

Option

sd reports the standard deviations of the correlated coefficients instead of the covariance matrix.

4 Examples

To show how the `mixlogit` command can fit mixed logit models with alternative-specific explanatory variables, we use part of the data from Huber and Train (2001) on households' choice of electricity supplier.[1] A sample of residential electricity customers were presented with four alternative electricity suppliers. The suppliers differed in the following characteristics: price per kilowatt-hour, length of contract, whether the company is local, and whether it is well known. Depending on the experiment, the price is either fixed or a variable rate that depends on the time of day or the season. The following explanatory variables enter the model:

- Price in cents per kilowatt-hour if fixed price, 0 if time-of-day or seasonal rates
- Contract length in years
- Whether company is local (0–1 dummy)
- Whether company is well known (0–1 dummy)
- Time-of-day rates (0–1 dummy)
- Seasonal rates (0–1 dummy)

The data setup for `mixlogit` is identical to that required by `clogit`. To give an impression of how the data are structured, I list the first 12 observations below. Each observation corresponds to an alternative, and the dependent variable `y` is 1 for the chosen alternative in each choice situation and 0 otherwise. `gid` identifies the alternatives in a choice situation, `pid` identifies the choice situations faced by a given individual, and the remaining variables are the alternative attributes described earlier. In the listed data, the same individual faces three choice situations.

1. You can download the dataset from Kenneth Train's web site as part of his excellent distance-learning course on discrete-choice methods (http://elsa.berkeley.edu/~train/).

```
. use traindata
. list in 1/12, sepby(gid)
```

	y	price	contract	local	wknown	tod	seasonal	gid	pid
1.	0	7	5	0	1	0	0	1	1
2.	0	9	1	1	0	0	0	1	1
3.	0	0	0	0	0	0	1	1	1
4.	1	0	5	0	1	1	0	1	1
5.	0	7	0	0	1	0	0	2	1
6.	0	9	5	0	1	0	0	2	1
7.	1	0	1	1	0	1	0	2	1
8.	0	0	5	0	0	0	1	2	1
9.	0	9	5	0	0	0	0	3	1
10.	0	7	1	0	1	0	0	3	1
11.	0	0	0	0	1	1	0	3	1
12.	1	0	0	1	0	0	1	3	1

We begin by fitting a model in which the coefficient for price is fixed and the remaining coefficients are normally distributed.[2] `mixlogit` uses the coefficients from a conditional logit model fitted using the same data as starting values for the means of the coefficients and sets the starting values for the standard deviations to 0.1. The model is fitted using 50 Halton draws. Whereas the accuracy of the results increases with the number of draws, so does the estimation time; the choice of draws therefore represents a tradeoff between the two. One possible strategy is to use a relatively small number of draws (say, 50) when doing the specification search and a larger number (say, 500) for the final model. Train (2003), Cappellari and Jenkins (2006), and Haan and Uhlendorff (2006) discuss the issue of accuracy in greater detail.

2. The fitted models have no alternative-specific constants. This is common practice when the data come from so-called unlabeled choice experiments, where the alternatives have no utility beyond the characteristics attributed to them in the experiment.

```
. global randvars "contract local wknown tod seasonal"
. mixlogit y price, rand($randvars) group(gid) id(pid) nrep(50)
Iteration 0:   log likelihood = -1320.2214  (not concave)
  (output omitted)
Iteration 8:   log likelihood = -1137.7962
Mixed logit model                                 Number of obs   =       4780
                                                  LR chi2(5)      =     437.18
Log likelihood = -1137.7962                       Prob > chi2     =     0.0000
```

y	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Mean						
price	-.8714238	.0587205	-14.84	0.000	-.9865138	-.7563338
contract	-.2337225	.0362325	-6.45	0.000	-.304737	-.162708
local	1.939449	.1736134	11.17	0.000	1.599173	2.279725
wknown	1.480568	.1427072	10.37	0.000	1.200867	1.760269
tod	-8.334529	.5066987	-16.45	0.000	-9.32764	-7.341418
seasonal	-8.449152	.5167853	-16.35	0.000	-9.462032	-7.436271
SD						
contract	.2959921	.0305113	9.70	0.000	.236191	.3557931
local	1.798179	.2129429	8.44	0.000	1.380819	2.21554
wknown	1.114257	.2248278	4.96	0.000	.6736025	1.554911
tod	1.560564	.1666314	9.37	0.000	1.233973	1.887156
seasonal	1.684004	.1799347	9.36	0.000	1.331338	2.036669

```
. *Save coefficients for later use
. matrix b = e(b)
```

On average, consumers prefer lower costs, shorter contract length, a local and well-known provider, and fixed rather than variable rates. Further, there is significant preference heterogeneity for all the attributes. From the magnitudes of the standard deviations relative to the mean coefficients, whereas practically all consumers prefer fixed to variable rates, 21% prefer longer contracts, 14% prefer a provider that is not local, and 9% prefer a provider that is not well known. These figures are given by $100 \times \Phi(-b_k/s_k)$, where Φ is the cumulative standard normal distribution and b_k and s_k are the mean and standard deviation, respectively, of the kth coefficient.

A likelihood-ratio test for the joint significance of the standard deviations is reported in the upper-right corner of the table. The associated p-value is small, implying rejection of the null hypothesis that all the standard deviations are equal to zero.

Restricting the sign of the coefficients to be either positive or negative for all individuals may sometimes be desirable. If so, the lognormal distribution provides an alternative to the normal distribution. Whereas specifying a coefficient to be lognormally distributed implies that it is positive for all individuals, negative coefficients can be accommodated by entering the attribute multiplied by -1 in the model. The following example demonstrates this by specifying the price coefficient to be lognormally distributed:

```
. gen mprice=-1*price
. global lnrandv "contract local wknown tod seasonal mprice"
. mixlogit y, rand($lnrandv) group(gid) id(pid) ln(1) nrep(50)
Iteration 0:   log likelihood = -1277.6348  (not concave)
  (output omitted)
Iteration 7:   log likelihood = -1130.7054
Mixed logit model                                 Number of obs   =       4780
                                                  LR chi2(6)      =     451.36
Log likelihood = -1130.7054                       Prob > chi2     =     0.0000
```

y	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Mean						
contract	-.2464903	.0357441	-6.90	0.000	-.3165473	-.1764332
local	2.19609	.2192702	10.02	0.000	1.766328	2.625852
wknown	1.47136	.1279781	11.50	0.000	1.220528	1.722193
tod	-8.604945	.5067256	-16.98	0.000	-9.598109	-7.611781
seasonal	-8.903156	.5259955	-16.93	0.000	-9.934089	-7.872224
mprice	-.0695898	.0681756	-1.02	0.307	-.2032115	.0640319
SD						
contract	.2791737	.0294739	9.47	0.000	.221406	.3369415
local	1.656503	.2948766	5.62	0.000	1.078556	2.234451
wknown	.673231	.1638918	4.11	0.000	.352009	.9944531
tod	.8999244	.2082437	4.32	0.000	.4917742	1.308075
seasonal	1.102238	.2370826	4.65	0.000	.6375645	1.566911
mprice	.2367957	.0256924	9.22	0.000	.1864395	.287152

The estimated price parameters in the above model are the mean (b_p) and standard deviation (s_p) of the natural logarithm of the price coefficient. The median, mean, and standard deviation of the coefficient itself are given by $\exp(b_p)$, $\exp(b_p + s_p^2/2)$, and $\exp(b_p + s_p^2/2) \times \sqrt{\exp(s_p^2) - 1}$, respectively (Train 2003). The standard errors of the mean, median, and standard deviation of the coefficient can be conveniently calculated using `nlcom`:

```
. nlcom (mean_price: -1*exp([Mean]_b[mprice]+0.5*[SD]_b[mprice]^2))
>         (med_price: -1*exp([Mean]_b[mprice]))
>         (sd_price: exp([Mean]_b[mprice]+0.5*[SD]_b[mprice]^2)
>                 * sqrt(exp([SD]_b[mprice]^2)-1))
  mean_price:  -1*exp([Mean]_b[mprice]+0.5*[SD]_b[mprice]^2)
   med_price:  -1*exp([Mean]_b[mprice])
    sd_price:  exp([Mean]_b[mprice]+0.5*[SD]_b[mprice]^2)
> * sqrt(exp([SD]_b[mprice]^2)-1)
```

y	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
mean_price	-.9592978	.0634784	-15.11	0.000	-1.083713	-.8348824
med_price	-.9327763	.0635926	-14.67	0.000	-1.057415	-.8081372
sd_price	.2303795	.0258277	8.92	0.000	.1797582	.2810008

The mean and median estimates have been multiplied by −1 to undo the sign change introduced in the estimation process.

The next example demonstrates how `mixlogit` can fit a model with correlated normally distributed coefficients. Here the `from()` option is used to specify the starting values, which are taken from the model with uncorrelated normal coefficients. The final 15 coefficients are the elements of the lower-triangular matrix $\mathbf{L}$, where the covariance matrix for the random coefficients is given by $\mathbf{V} = \mathbf{LL}'$ (the $\mathbf{L}$ matrix is the Cholesky factorization of the covariance matrix $\mathbf{V}$).

```
. *Starting values
. matrix b = b[1,1..7],0,0,0,0,b[1,8],0,0,0,b[1,9],0,0,b[1,10],0,b[1,11]
. mixlogit y price, rand($randvars) group(gid) id(pid) nrep(50) corr
> from(b, copy)
Iteration 0:   log likelihood = -1137.7962  (not concave)
  (output omitted)
Iteration 11:  log likelihood = -1060.8267
Mixed logit model                                 Number of obs   =       4780
                                                  LR chi2(15)     =     591.12
Log likelihood = -1060.8267                       Prob > chi2     =     0.0000

------------------------------------------------------------------------------
           y |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
       price |  -.8886558   .0604113   -14.71   0.000     -1.00706   -.7702517
    contract |  -.2283449   .0354989    -6.43   0.000    -.2979216   -.1587683
       local |   2.526601   .2448635    10.32   0.000     2.046677    3.006524
      wknown |   1.994449   .1883359    10.59   0.000     1.625318    2.363581
         tod |  -8.680891   .5628236   -15.42   0.000    -9.784005   -7.577777
    seasonal |  -8.480598   .5405829   -15.69   0.000    -9.540121   -7.421075
-------------+----------------------------------------------------------------
        /l11 |   .3242159   .0327134     9.91   0.000     .2600988     .388333
        /l21 |   .5076903   .1918852     2.65   0.008     .1316022    .8837785
        /l31 |   .5164185   .1574542     3.28   0.001     .2078139    .8250231
        /l41 |  -.5622626   .2119886    -2.65   0.008    -.9777527   -.1467725
        /l51 |   .2008204    .193612     1.04   0.300    -.1786521    .5802928
        /l22 |   2.638329   .2709843     9.74   0.000      2.10721    3.169449
        /l32 |    1.69457   .2366775     7.16   0.000      1.23069    2.158449
        /l42 |   .5041138   .2377615     2.12   0.034     .0381099    .9701178
        /l52 |   .6190068   .2024403     3.06   0.002     .2222311    1.015782
        /l33 |   .4146707   .1683532     2.46   0.014     .0847044     .744637
        /l43 |    1.13526   .2551698     4.45   0.000     .6351367    1.635384
        /l53 |   .3854603   .2379867     1.62   0.105     -.080985    .8519056
        /l44 |   2.003161   .2427176     8.25   0.000     1.527443    2.478879
        /l54 |   1.346629   .2146771     6.27   0.000     .9258694    1.767388
        /l55 |    1.57518   .1856905     8.48   0.000     1.211233    1.939127
------------------------------------------------------------------------------
```

The joint significance of the off-diagonal elements of the covariance matrix can be tested using a likelihood-ratio test. The test statistic, which is chi-squared distributed with 10 degrees of freedom under the null of uncorrelated coefficients, is given by $2 \times (1{,}137.7962 - 1{,}060.8267) = 153.939$, implying rejection of the null hypothesis.

The covariance matrix and standard deviations of the random coefficients can conveniently be calculated using `mixlcov`:

```
. mixlcov
```
(*output omitted*)

y	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
v11	.1051159	.0212124	4.96	0.000	.0635404	.1466915
v21	.1646013	.0664459	2.48	0.013	.0343696	.2948329
v31	.1674311	.055532	3.02	0.003	.0585903	.2762718
v41	-.1822945	.0772516	-2.36	0.018	-.3337048	-.0308841
v51	.0651091	.0622506	1.05	0.296	-.0568998	.1871181
v22	7.218532	1.40776	5.13	0.000	4.459373	9.977691
v32	4.733013	1.031262	4.59	0.000	2.711778	6.754249
v42	1.044563	.6297305	1.66	0.097	-.1896861	2.278812
v52	1.735098	.5491026	3.16	0.002	.658877	2.81132
v33	3.310206	.8129714	4.07	0.000	1.716811	4.903601
v43	1.034652	.4864574	2.13	0.033	.0812134	1.988091
v53	1.312496	.3707537	3.54	0.000	.5858326	2.03916
v44	5.871741	1.390635	4.22	0.000	3.146145	8.597336
v54	3.334249	.8074509	4.13	0.000	1.751674	4.916823
v55	4.866679	.9491078	5.13	0.000	3.006462	6.726896

```
. mixlcov, sd
```
(*output omitted*)

y	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
contract	.3242159	.0327134	9.91	0.000	.2600988	.388333
local	2.686733	.2619837	10.26	0.000	2.173254	3.200211
wknown	1.819397	.2234178	8.14	0.000	1.381506	2.257288
tod	2.423167	.2869458	8.44	0.000	1.860764	2.985571
seasonal	2.206055	.2151143	10.26	0.000	1.784439	2.627671

To show how the **mixlogit** command can fit a multinomial logit model with unobserved heterogeneity, we use the data from Haan and Uhlendorff (2006) on teachers' ratings of pupils' behavior. The first step is to rearrange the data so that they are in the form required by **mixlogit**. This is analogous to the example in Long and Freese (2006), section 7.2.4, which shows how **clogit** can fit a multinomial logit model. I list the first 4 observations in the dataset below:

```
. use jspmix, clear
. list scy3 id tby sex in 1/4
```

	scy3	id	tby	sex
1.	1	280	1	0
2.	1	281	2	1
3.	1	282	1	0
4.	1	283	1	1

The next step is to expand the data. Because there are three alternatives (low, medium, and high performance), we create three duplicate records with the `expand 3` command. Then we create variable `alt`, which identifies the alternatives and is used to generate alternative-specific constants, as well as interactions with the gender variable:

```
. expand 3
(2626 observations created)
. by id, sort: gen alt = _n
. gen mid = (alt == 2)
. gen low = (alt == 3)
. gen sex_mid = sex*mid
. gen sex_low = sex*low
```

Finally, we generate the new dependent variable `choice` that equals 1 if `tby == alt` and 0 otherwise:

```
. gen choice = (tby == alt)
```

The observations corresponding to the first four records in the original dataset are below:

```
. sort scy3 id alt
. list scy3 id choice mid low sex_mid sex_low in 1/12, sepby(id)
```

	scy3	id	choice	mid	low	sex_mid	sex_low
1.	1	280	1	0	0	0	0
2.	1	280	0	1	0	0	0
3.	1	280	0	0	1	0	0
4.	1	281	0	0	0	0	0
5.	1	281	1	1	0	1	0
6.	1	281	0	0	1	0	1
7.	1	282	1	0	0	0	0
8.	1	282	0	1	0	0	0
9.	1	282	0	0	1	0	0
10.	1	283	1	0	0	0	0
11.	1	283	0	1	0	1	0
12.	1	283	0	0	1	0	1

To replicate the results from Haan and Uhlendorff (2006), we begin by fitting a model with random but uncorrelated intercepts:

```
. mixlogit choice sex_mid sex_low, group(id) id(scy3) rand(mid low) nrep(50)
Iteration 0:   log likelihood = -1329.3862  (not concave)
  (output omitted)
Iteration 4:   log likelihood = -1315.5573
Mixed logit model                                 Number of obs   =       3939
                                                  LR chi2(2)      =      32.73
Log likelihood = -1315.5573                       Prob > chi2     =     0.0000
```

choice	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Mean						
sex_mid	.4797341	.1419879	3.38	0.001	.201443	.7580252
sex_low	1.019557	.1699843	6.00	0.000	.6863943	1.352721
mid	.531875	.1143518	4.65	0.000	.3077496	.7560004
low	-.6773663	.1503376	-4.51	0.000	-.9720225	-.3827101
SD						
mid	.514833	.1095759	4.70	0.000	.3000681	.7295979
low	.5778384	.1126083	5.13	0.000	.3571303	.7985466

```
. matrix b = e(b)
```

The next step is to use the coefficients from the above model as starting values for the final model specification with correlated intercepts:

```
. matrix b = b[1,1..5],0,b[1,6]
. mixlogit choice sex_mid sex_low, group(id) id(scy3) rand(mid low) corr
> nrep(5 0) from(b, copy)
Iteration 0:   log likelihood = -1315.5573
  (output omitted)
Iteration 5:   log likelihood = -1300.1117
Mixed logit model                                 Number of obs   =       3939
                                                  LR chi2(3)      =      63.62
Log likelihood = -1300.1117                       Prob > chi2     =     0.0000
```

choice	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
sex_mid	.5494836	.1456751	3.77	0.000	.2639657	.8350015
sex_low	1.101967	.1747535	6.31	0.000	.7594559	1.444477
mid	.6278598	.1425238	4.41	0.000	.3485182	.9072013
low	-.5204487	.1806557	-2.88	0.004	-.8745274	-.16637
/l11	.7321527	.119431	6.13	0.000	.4980721	.9662332
/l21	.8096981	.1564731	5.17	0.000	.5030165	1.11638
/l22	-.346577	.1106231	-3.13	0.002	-.5633942	-.1297597

The results are similar, but not identical, to those reported by Haan and Uhlendorff. The Halton draws are generated differently in the two applications: whereas Haan and Uhlendorff base their draws on primes 7 and 11, `mixlogit` uses primes 2 and 3 (see Drukker and Gates [2006] for a description of how Halton draws are generated). Simulation-based estimators will generally produce slightly different results unless the draws are generated in the same way.

As before, the covariance matrix and standard deviations of the random coefficients can conveniently be calculated using `mixlcov`:

```
. mixlcov
  (output omitted)
```

choice	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
v11	.5360475	.1748835	3.07	0.002	.1932821	.8788129
v21	.5928226	.1889485	3.14	0.002	.2224904	.9631548
v22	.7757266	.2540111	3.05	0.002	.2778739	1.273579

```
. mixlcov, sd
  (output omitted)
```

choice	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
mid	.7321527	.119431	6.13	0.000	.4980721	.9662332
low	.8807534	.1442011	6.11	0.000	.5981245	1.163382

5 Acknowledgments

I thank Kenneth Train for helpful comments and for giving me permission to use his data. Thanks also to Peter Haan for providing me with the data on pupil behavior, Peter Sivey for helping to test the code, and an anonymous referee for helpful comments and suggestions for how to improve the manuscript.

National Primary Care Research and Development Centre receives funding from the Department of Health. The views expressed are not necessarily those of the funders.

6 References

Cappellari, L., and S. P. Jenkins. 2006. Calculation of multivariate normal probabilities by simulation, with applications to maximum simulated likelihood estimation. *Stata Journal* 6: 156–189.

Drukker, D. M., and R. Gates. 2006. Generating Halton sequences using Mata. *Stata Journal* 6: 214–228.

Haan, P., and A. Uhlendorff. 2006. Estimation of multinomial logit models with unobserved heterogeneity using maximum simulated likelihood. *Stata Journal* 6: 229–245.

Huber, J., and K. Train. 2001. On the simularity of classical and Bayesian estimates of individual mean partworths. *Marketing Letters* 12: 259–269.

Long, J. S., and J. Freese. 2006. *Regression Models for Categorical Dependent Variables Using Stata*. 2nd ed. College Station, TX: Stata Press.

McFadden, D. 1974. Conditional logit analysis of qualitative choice behavior. In *Frontiers in Econometrics*, ed. P. Zerembka, 105–142. New York: Academic Press.

Revelt, D., and K. Train. 1998. Mixed logit with repeated choices: Households' choices of appliance efficiency level. *Review of Economics and Statistics* 80: 647–657.

Train, K. 2003. *Discrete Choice Methods with Simulation*. Cambridge: Cambridge University Press.

About the author

Arne Risa Hole is a research fellow at the Centre for Health Economics, York, UK.