



AgEcon SEARCH
RESEARCH IN AGRICULTURAL & APPLIED ECONOMICS

The World's Largest Open Access Agricultural & Applied Economics Digital Library

This document is discoverable and free to researchers across the globe due to the work of AgEcon Search.

Help ensure our sustainability.

Give to AgEcon Search

AgEcon Search

<http://ageconsearch.umn.edu>

aesearch@umn.edu

*Papers downloaded from **AgEcon Search** may be used for non-commercial purposes and personal study only. No other use, including posting to another Internet site, is permitted without permission from the copyright owner (not AgEcon Search), or as allowed under the provisions of Fair Use, U.S. Copyright Act, Title 17 U.S.C.*

No endorsement of AgEcon Search or its fundraising activities by the author(s) of the following work or their employer(s) is intended or implied.

Fitting garbage class mixed logit models in Stata

Marcel F. Jonker
Erasmus University Rotterdam
Rotterdam, The Netherlands
marcel@mfjonker.com

Abstract. In this article, I describe the `garbage_mixl` command, which fits the garbage class and standard panel mixed logit models in Stata.

Keywords: `st0753`, `garbage_mixl`, garbage class, `mixlogit`, mixed logit, random parameter logit

1 Introduction

Mixed logit (MIXL) models, also called random parameter logit models, are commonly used discrete choice models. Stata offers built-in support for fitting panel MIXL models using simulated maximum likelihood via the `cmxtmixlogit` command, and there are several community-contributed commands that fit different types of panel MIXL models using either simulated maximum likelihood (Hole 2007, 2015; Gu, Hole, and Knox 2013) or Bayesian Markov chain Monte Carlo (MCMC) methods (Baker 2014).

In this article, I describe the `garbage_mixl` command, which fits garbage class MIXL models in Stata.¹ The garbage class MIXL model (Jonker 2022) is an extension of the standard panel MIXL model that is primarily aimed at automatic screening and accounting for respondents with low data quality in discrete choice experiments. However, the `garbage_mixl` command can also fit standard panel MIXL models, and because of the specifics of the implementation, it provides a fast and reliable alternative to Stata's `cmxtmixlogit` command and the community-contributed estimation commands of Hole (2007) and Baker (2014), although with a few limitations (see section 3) and fewer customization options.

The article is organized as follows: section 2 provides a brief overview of the standard and garbage class MIXL model. Section 3 explains the estimation strategy. Section 4 describes the command's syntax and estimation options, and section 5 presents several examples. Section 6 provides a comparison between the `garbage_mixl` command and other estimation commands, and section 7 concludes.

1. The `garbage_mixl` command is compatible with all Windows machines.

2 The standard and garbage class MIXL model

The standard panel MIXL model assumes that there are N respondents that each complete T discrete choices, each from a set of J alternatives that are described by K explanatory variables. The explanatory variables can be summarized as

$$X_{itjk} \in \mathbb{R} \quad \text{for } i = 1, \dots, N; \quad t = 1, \dots, T; \quad j = 1, \dots, J; \quad k = 1, \dots, K$$

and the observed choices in the dependent variable

$$Y_{itj} \in \{0, 1\}$$

with entries equal to 1 for the alternative that was chosen and zero for all other alternatives in the choice tasks; that is,

$$(Y_{itj} = 1) \implies (Y_{itm} = 0, \quad \forall m \neq j)$$

Each respondent is presumed to have chosen the option that provides the highest utility, with the utility function specified as

$$U_{itj} = \sum_{k=1}^K \beta_{ik} \times X_{itjk} + \epsilon_{itj}$$

and with the error term ϵ assumed to be independently and identically Gumbel distributed. This implies that the probability of respondent i choosing alternative j in choice task t can be defined as

$$P_{itj} = \frac{\exp(\beta'_i \mathbf{X}_{itj})}{\sum_{m=1}^J \exp(\beta'_i \mathbf{X}_{itm})}$$

In the standard MIXL model, the mixing distribution of the respondents' β coefficients is multivariate normal (MVN) with mean vector μ and covariance matrix Σ ; that is,

$$f(\beta_i) \sim \text{MVN}(\mu, \Sigma)$$

which means that the likelihood contribution of each individual respondent is defined as

$$L_i = (2\pi)^{-\frac{1}{2}K} |\det(\Sigma)|^{-\frac{1}{2}} \int_{-\infty}^{\infty} \exp \left[-\frac{1}{2} \langle (\beta_i - \mu)' \Sigma^{-1} (\beta_i - \mu) \rangle + \sum_{t=1}^T \sum_{j=1}^J \{Y_{itj} \log(P_{itj})\} \right] d\beta_i \quad (1)$$

and with the overall (sample) likelihood function defined as the product of respondent-specific likelihoods over all respondents in the dataset; see, for example, Train (2009).

The garbage class MIXL model is an extension of the standard panel MIXL model in which the structural part of the utility function ($\beta'_i \mathbf{X}_{itj}$) is multiplied with a class membership parameter ($\varphi_i \in [0, 1]$),

$$U_{itj} = \varphi_i \times \beta'_i \mathbf{X}_{itj} + \epsilon_{itj}$$

No other modifications are involved, and the interpretation of the class-membership parameter is also straightforward: φ_i represents the respondent-specific probability of being assigned to the standard MIXL utility specification. If φ_i equals 1, respondents are entirely assigned to the standard MIXL utility specification ($U_{itj} = \beta'_i \mathbf{X}_{itj} + \epsilon_{itj}$). If φ_i equals 0, there is no contribution from the structural part of the utility function, and respondents make random choices based on the error term ($U_{itj} = \epsilon_{itj}$). When aggregated, the sample average $\bar{\varphi} = \sum_{i=1}^N \varphi_i / N$ reflects the class share of the MIXL model and $(1 - \bar{\varphi})$ the garbage class share. The latter provides a readily available and objectively comparable estimate of the number of low-quality respondents in the dataset (Jonker 2022).

3 Estimation strategy

Because the MIXL likelihood function is not analytically tractable, the integral in (1) needs to be approximated. In Stata, MIXL model estimation is typically performed using simulated maximum-likelihood methods (see, for example, Train [2009]). Simulated maximum-likelihood estimation has a few practical disadvantages, including the risk of convergence in local optima. This necessitates that multiple MIXL model estimations be fit from different initial values. Additionally, the quality of pseudo-random draws may degrade in higher dimensions. For instance, standard Halton draws should not be used for models with more than 10–12 random parameters, even though some Stata commands allow up to 20 random parameters. In contrast, the implemented Bayesian estimation approach for the `garbage_mixl` command offers an attractive alternative designed to provide fast and robust results, particularly in higher dimensions. The default priors are defined as follows:

1. Normal priors with a mean of zero and standard deviation (SD) of 10 are assigned to the population mean parameters (μ).
2. A Huang and Wand (2013) scaled inverse-Wishart prior is assigned to the population covariance matrix (Σ), which results in marginal uniform distributions on the correlation parameters and half- t distributions with 2 degrees of freedom and SDs of 10 on the SDs of the random parameters.
3. `bernoulli(0.5)` priors are assigned to the MIXL class-selection parameters (φ_i).

Gibbs sampling is used to update the μ and Σ parameters, inverse transform sampling to update the φ_i parameters, and a combination of Metropolis-within-Gibbs (MWG) with antithetic updates (Bédard, Douc, and Moulin 2014) and a Metropolis-adjusted Langevin algorithm (MALA) (Marshall and Roberts 2012) to update the individual-level preference parameters (β_i). Both the MWG and the MALA samplers are tuned at increasing intervals during the burn-in phase, with diminishing adaptation and with target acceptance rates of 0.44 and 0.57, respectively.

To minimize the risk of nonconvergence and to ensure sufficiently accurate approximations of the posterior, the default and minimum number of MCMC draws of the

`garbage_mix1` command is set to 50,000 burn-in and 50,000 additional MCMC draws. This implies a minimum of 100,000 MCMC draws; in comparison, the `bayesmixedlogit` command (Baker 2014) has a default of 1,000 MCMC draws. In addition to the large burn-in phase, an MWG sampler with antithetic updates is used to reliably converge from the initial values to the posterior distribution during the initial 10,000 draws of the burn-in phase, updating only 1 parameter at a time. The MWG sampler is also used to create an initial estimate of the empirical variance–covariance matrix based on draws 4,000–10,000. The latter is used to transition to a more efficient and faster MALA sampler that updates the β_i parameters simultaneously but requires an estimate of the variance–covariance matrix to effectively explore the posterior. When one fits standard panel MIXL models, the MALA sampler is used for all respondents as the final sampling algorithm. However, in the garbage class MIXL model, there are insufficient burn-in draws to reliably tune the MALA sampler when respondents are predominantly assigned to the garbage class. Hence, for respondents who have a probability of MIXL class membership smaller than 0.5, MWG is used as the final sampling algorithm.

To further assist and improve convergence of the MCMC samplers, relatively good initial values are obtained using an initial maximum-likelihood estimation. In this initial MIXL model optimization, the integral in (1) is approximated using a standard Laplace approximation (see, for example, Bleistein and Handelsman [1986, chap. 8] and Small [2010, chap. 6]). The Laplace approximation is an analytical (rather than sampling-based) approximation and allows for a MIXL model to be fit within seconds (see table A in the online supplemental). This provides relatively good initial values for the population mean (μ), covariance matrix (Σ), and all individual-level preference parameters (β_i), with all class-selection parameters (φ_i) initialized at 1 when fitting garbage class MIXL models. Because the Laplace MIXL estimates are used only as initial values and more precise initial estimates are superfluous, a tolerance of $1e-5$ on the L2 norm of the gradient is used as the stopping criterion, and the number of Broyden–Fletcher–Goldfarb–Shannon model optimization iterations is maximized at $25 + 2 \times K$.

To further improve computational efficiency, all estimations are implemented in C++ and linked to Stata via a plugin (<https://www.stata.com/plugins/>). This implies that the Stata graphical user interface freezes during `garbage_mix1` estimations, which is an inherent and unavoidable limitation of Stata’s C++ plugin functionality. Another limitation of the `garbage_mix1` command is that it requires an equal number of choice options per choice task in the dataset. This is not an inherent limitation but enforced to simplify the data transfer from Stata to the C++ plugin and to simplify the implementation of the likelihood and gradient calculations. Finally, because the MIXL class-selection parameters are identified based on the choice data of each individual respondent, garbage MIXL models tend to be sensitive to the priors that are assigned to the MIXL class-selection parameters. To make sure that there is enough information for each respondent and the class selection is not based entirely on the priors that are used, the `garbage_mix1` command requires at least as many choice tasks per respondent as random parameters when fitting garbage class MIXL models.

4 Syntax and options

4.1 Syntax

```
garbage_mixl devar [if] [in], rand(varlist) id(varname) group(varname)
    [burn(#) mcmc(#) bernoulli(#) nogarbage diag invwishart]
```

4.2 Options

`rand(varlist)` is required and specifies the independent variables. All independent variables' coefficients are assumed to be random; fixed coefficients are not supported.

`id(varname)` is required and specifies a numeric identifier variable for the decision makers.

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

`burn(#)` specifies the length of the burn-in period. The default and minimum value is `burn(50000)`.

`mcmc(#)` specifies the number of draws to be taken from the posterior distribution of the parameters. The default and minimum value is `mcmc(50000)`.

`bernoulli(#)` specifies the prior distribution for the individual-level MIXL class probabilities in the garbage class MIXL model. The default is `bernoulli(0.5)`. Note that the garbage class MIXL results tend to be sensitive to the Bernoulli prior, particularly with limited observations per decision maker. Therefore, the `garbage_mixl` command requires at least as many observations per decision maker as the number of random parameters specified.

`nogarbage` specifies that the `garbage_mixl` command fit a standard panel MIXL model.

`diag` specifies that a diagonal (uncorrelated) MIXL model is being fit, with half-normal (10) priors on the SDs of the normal mixing distributions.

`invwishart` replaces the default Huang and Wand (2013) scaled inverse-Wishart prior with a conventional inverse-Wishart prior on the variance-covariance matrix. The inverse-Wishart prior is specified with a unit scale matrix and degrees of freedom equal to the number of random parameters. This prior is more informative and more restrictive than the default prior, particularly when more random parameters are specified, and is seldom recommended.

5 Examples

To show how the `garbage_mix1` command fits standard and garbage class MIXL models, let's revisit the antibiotics example of Jonker (2022). This dataset can be downloaded from the *Stata Journal* website and contains 750 respondents who were recruited using a commercial survey sample provider. All respondents were asked to make 13 repeated choices between 2 types of antibiotics screening tests that were described using the following characteristics: 1) the speed in which the test results are available (fast versus slow), 2) the convenience of the diagnostic test (high versus low), and 3) the confidence in the test's results (very high versus high). Figure 1 shows an example choice task as shown to the respondents. The dependent variable y in the dataset describes the choices that were made, and the independent variables' speed, convenience, and confidence are dummy coded with the least favorable test characteristics selected as the omitted base-case levels.

If these were your only options and you had to make a choice, which diagnostic test would you prefer?

(1 of 13)

Test A	Test B
Speed: Slow (results next working day)	Speed: Fast (results after 12 minutes)
Convenience: High (take sample yourself, no discomfort)	Convenience: Low (sample taken by health care professional, possibly some discomfort)
Confidence: High (90% of results are correct)	Confidence: High (90% of results are correct)
<input type="button" value="Select"/>	<input type="button" value="Select"/>

Reminder: both tests provide information on [whether you need antibiotics or not](#), are made available at no additional cost, and are identical in all other aspects.

Figure 1. Example discrete choice task

5.1 Example 1

Let's start with fitting a standard panel MIXL model:

```
. use antibiotics
. garbage_mixl y, rand(speed convenience confidence) id(resp_id) group(group)
> nogarbage
The user interface will freeze during C++ model estimation; please do not close
> Stata!

Achieved MALA acceptance rates
min = 0.55
ave = 0.57
max = 0.60
```

Bayesian mixed logit		number of resp = 750				
		number of obs = 9750				
		MCMC burn-in = 50000				
		MCMC sample = 50000				
		Mean	Std.Dev.	MCSE	Median	[95% Cred. Interval]
Mean						
	speed	1.2254	.082619	.001182	1.22382	1.06788 1.39207
	convenience	1.83737	.111471	.001583	1.83553	1.62394 2.06094
	confidence	3.62273	.17944	.002509	3.61895	3.28248 3.98405
SD						
	speed	1.13157	.091489	.002267	1.13084	.954914 1.31343
	convenience	2.04984	.115009	.002209	2.04701	1.83199 2.28293
	confidence	3.37542	.17942	.003131	3.37025	3.04038 3.7383
Corr						
	β_1 - β_2	.23324	.076426	.001385	.23525	.077594 .375337
	β_1 - β_3	.43093	.066852	.000971	.43318	.294203 .55495
	β_2 - β_3	.425895	.048819	.00062	.427038	.326945 .518102

Note: Default priors are used for model parameters

As shown, the achieved MALA acceptance rates range from 0.55 to 0.60. Even though the performance of MCMC sampling algorithms is robust to deviations from optimal acceptance rates (see, for example, Roberts and Rosenthal [2001]), it is reassuring that the MALA samplers have achieved acceptance rates close to the target rate of 0.57. Similarly, it is reassuring to observe that the Monte Carlo standard errors (MCSE) expressed as a percentage of the SDs of the posterior samples are smaller than 5–10%. This is a simple rule of thumb to determine whether the posterior distributions are sufficiently accurately approximated.

The standard panel MIXL model output presents posterior summary statistics for the MIXL mean estimates, while the (co)variance estimates are separated into SD and correlation estimates. This approach is akin to Stata's `cmxtmixlogit` command, making it more convenient to interpret the degree of heterogeneity in and correlations among preferences for the explanatory variables. As indicated by the positive MIXL mean estimates, respondents on average prefer faster, more convenient, and more reliable tests. This has good face validity. There is also substantial variation in respondent-specific preferences

with the MIXL SD estimates being approximately as large as or larger than the MIXL mean estimates. This is somewhat unexpected because it suggests that a substantial number of respondents in the tail of the normal distributions actually prefer slower, less convenient, and less reliable diagnostics tests. In addition, the correlations between the preference parameters all have 95% credible intervals that are positive and do not comprise 0. This is entirely unexpected because it indicates that when respondents find one particular aspect of the test more important, they also find the other aspects of the test more important (and vice versa). Of course, these unexpected results could also be explained by not appropriately accounting for a subset of respondents who flatlined on one of the choice options or randomly clicked through the survey, which is a notorious problem with discrete choice experiment data obtained from commercial survey sample providers.

5.2 Example 2

As explained, the garbage class MIXL model intends to provide MIXL estimates that are corrected for the presence of respondents with poor data quality. If such respondents are included in the model estimations, one would expect the standard panel MIXL mean estimates to be biased toward zero and the MIXL SD estimates to overstate the true amount of variability in preferences in the MIXL class. Let's refit the model without the `nogarbage` option to see what happens:

```
. garbage_mixl y, rand(speed convenience confidence) id(resp_id) group(group)
The user interface will freeze during C++ model estimation; please do not close
> Stata!
```

```
Achieved MWG acceptance rates (n=243 resp)
```

```
min = 0.42
```

```
ave = 0.44
```

```
max = 0.46
```

```
Achieved MALA acceptance rates (n=507 resp)
```

```
garbage class = 0.57
```

```
MIXL class min = 0.57
```

```
ave = 0.59
```

```
max = 0.63
```

		Mean	Std.Dev.	MCSE	Median	[95% Cred. Interval]	
Bayesian garbage class mixed logit						number of resp =	750
						number of obs =	9750
						MCMC burn-in =	50000
						MCMC sample =	50000
	Mean						
	speed	3.074	.199707	.015449	3.07086	2.68734	3.46984
	convenience	5.0649	.349006	.026417	5.05412	4.39913	5.78365
	confidence	10.546	.704844	.061713	10.5345	9.2038	11.9731
	SD						
	speed	1.6416	.184716	.008732	1.63726	1.29921	2.01095
	convenience	4.06778	.338572	.022801	4.05106	3.43727	4.78173
	confidence	6.35555	.556382	.046813	6.33798	5.30431	7.50541
	Corr						
	β_1 - β_2	-.68414	.101021	.004215	-.688259	-.871383	-.472341
	β_1 - β_3	-.271483	.123048	.006804	-.270341	-.507598	-.032767
	β_2 - β_3	-.100439	.082623	.003097	-.100741	-.262969	.060548
	garbage class	.343424	.007357	.000354	.344	.329333	.357333

Note: Default priors are used for model parameters

When a garbage MIXL model is being fit, the achieved acceptance rates for both the MWG and the MALA updates are reported separately. It is still reassuring to observe that the achieved acceptance rates are close to the target values of 0.44 and 0.57. The MCSE expressed as a percentage of the SDs is again between 5–10%. This should be sufficient for accurate results, but to verify that the accuracy of the posterior estimates improves with additional MCMC draws, let's rerun the model with 100,000 posterior draws before interpreting the results.

5.3 Example 3

```
. garbage_mixl y, rand(speed convenience confidence) id(resp_id) group(group)
> mcmc(100000)
The user interface will freeze during C++ model estimation; please do not close
> Stata!

Achieved MwG acceptance rates (n=242 resp)
min = 0.41
ave = 0.43
max = 0.45

Achieved MALA acceptance rates (n=508 resp)
garbage class = 0.57
MIXL class min = 0.55
      ave = 0.57
      max = 0.60
```

Bayesian garbage class mixed logit		number of resp = 750				
		number of obs = 9750				
		MCMC burn-in = 50000				
		MCMC sample = 100000				
		Mean	Std.Dev.	MCSE	Median	[95% Cred. Interval]
Mean						
	speed	3.05994	.211305	.006764	3.04963	2.67254 3.49173
	convenience	5.05092	.368091	.012091	5.03297	4.38113 5.83391
	confidence	10.5244	.742177	.027812	10.4757	9.20792 12.1336
SD						
	speed	1.63651	.199268	.008799	1.63402	1.25637 2.04053
	convenience	4.05656	.355563	.012215	4.03651	3.41999 4.81836
	confidence	6.32107	.563709	.026675	6.28175	5.32027 7.53663
Corr						
	β_1 - β_2	-.691453	.105089	.006391	-.698139	-.873565 -.467883
	β_1 - β_3	-.279289	.125751	.006943	-.279069	-.523331 -.036835
	β_2 - β_3	-.107222	.082121	.002711	-.107767	-.267296 .053768
	garbage class	.343758	.007442	.000136	.344	.329333 .358667

Note: Default priors are used for model parameters

As shown, the updated output is almost identical to the previous estimation results, but the MCSE are indeed smaller. Turning to the comparison between the garbage class and standard MIXL results, we should first notice that the size of the garbage class in the antibiotics dataset is approximately 34%. Accounting for respondents with random response patterns, the MIXL mean estimates become larger (that is, further away from zero, indicating higher average choice consistency), and the MIXL SD estimates relative to the means become smaller (implying less heterogeneity and, more importantly, a smaller proportion of respondents who prefer slower, less convenient, and less reliable diagnostic tests). Moreover, the signs of the mean posterior correlation estimates have switched and are now negative. This indicates that respondents in the MIXL class have stronger preferences for either the test's speed or the test's convenience and reliability. The reported negative correlations behaviorally make more sense than the positive

correlations as reported by the standard panel MIXL model. Of course, the positive correlations in the standard panel MIXL model are still correct in that they accurately reflect the effect of pooling respondents with random and informative response patterns in a single MIXL estimation.

5.4 Example 4

As explained in the estimation strategy section, the default `bernoulli(0.5)` prior implies an equal prior probability of being in the garbage or MIXL class. For each respondent, this accurately reflects the absence of prior information about the most likely class assignment. However, at the sample level, this also corresponds to the prior expectation that the size of the garbage class is approximately 50%. The latter is probably unintended because commercial survey samples commonly result in 10–25% low-quality respondents (not 50%). Fortunately, the Bayesian estimation strategy provides a convenient framework to incorporate such prior expectations in the model estimation. For example, let's see what happens when a `bernoulli(0.8)` prior is specified:

```
. garbage_mixed y, rand(speed convenience confidence) id(resp_id) group(group)
> mcmc(100000) bernoulli(0.8)
The user interface will freeze during C++ model estimation; please do not close
> Stata!

Achieved MwG acceptance rates (n=204 resp)
min = 0.42
ave = 0.44
max = 0.45

Achieved MALA acceptance rates (n=546 resp)
garbage class = 0.57
MIXL class min = 0.55
      ave = 0.57
      max = 0.59
```

Bayesian garbage class mixed logit		number of resp = 750				
		number of obs = 9750				
		MCMC burn-in = 50000				
		MCMC sample = 100000				
		Mean	Std.Dev.	MCSE	Median	[95% Cred. Interval]
Mean						
	speed	2.48552	.159642	.00788	2.48196	2.17981 2.8031
	convenience	3.90367	.261203	.010679	3.89665	3.41035 4.43326
	confidence	7.79861	.498313	.023663	7.78262	6.87025 8.81222
SD						
	speed	1.27895	.152155	.006394	1.27395	.996267 1.59192
	convenience	3.13892	.243284	.01061	3.13	2.69065 3.64387
	confidence	4.91009	.397105	.016468	4.89282	4.17732 5.73888
Corr						
	β_1 - β_2	-.700748	.111813	.005035	-.711817	-.886607 -.456652
	β_1 - β_3	-.253738	.120959	.003594	-.254727	-.488448 -.011089
	β_2 - β_3	-.014247	.076459	.001332	-.014042	-.165295 .134893
garbage class		.278868	.008596	.000321	.278667	.261333 .294667

Note: Bernoulli(0.8) priors are used for MIXL class selection parameters, default priors are used for other model parameters

As shown, specifying a `bernoulli(0.8)` prior reduces the size of the garbage class to approximately 28% and reduces the size of the MIXL mean estimates. In all other respects, the results remain similar to those obtained with the default `bernoulli(0.5)` prior: The mean estimates are still larger than those in the standard MIXL model; the MIXL SD estimates are still smaller than the MIXL means; and the correlation estimates are still negative as opposed to positive in the standard MIXL model.

5.5 Example 5

The size of the garbage class provides a readily available estimate of the number of low-quality respondents in a dataset. Nevertheless, there are situations where it is convenient to be able to manually identify respondents with low data quality, such as when performing sensitivity analyses. Thus, the posterior probability of each included

respondent belonging to the garbage class is made available via the `ereturn` matrix `pr_gar`. This matrix can, after performing a garbage class MIXL estimation, be inspected using the `matrix list` command:

```
. matrix list e(pr_gar)
e(pr_gar)[750,2]
      pr_gar  resp_id
r1      .88398      1
r2      .69107      2
r3      .00020      3
r4      .00089      4
r5      .99249      5
      (output omitted)
r750    .00011      750
```

The listed matrix corresponds to the garbage class MIXL model estimates from the previous example, and the provided class membership probabilities can be used to selectively exclude respondents from the dataset. For instance, let's observe what happens when respondents with garbage class membership probabilities ≥ 0.5 are removed from the sample and a standard panel MIXL model is fit on the subsample of respondents with acceptable data quality:

```
. local todrop
. forvalues i = 1/750 {
2.     if e(pr_gar)[`i',1]>= 0.5 {
3.         local id = e(pr_gar)[`i',2]
4.         local todrop `todrop' `id'
5.     }
6. }
. foreach id of local todrop {
2.     quietly drop if resp_id == `id'
3. }
```

```
. garbage_mixl y, rand(speed convenience confidence) id(resp_id) group(group)
> nogarbage
The user interface will freeze during C++ model estimation; please do not close
> Stata!

Achieved MALA acceptance rates
min = 0.55
ave = 0.57
max = 0.58
```

Bayesian mixed logit		number of resp = 540				
		number of obs = 7020				
		MCMC burn-in = 50000				
		MCMC sample = 50000				
		Mean	Std.Dev.	MCSE	Median	[95% Cred. Interval]
Mean						
	speed	2.60953	.143569	.008224	2.60447	2.34368 2.9059
	convenience	4.07655	.22953	.009973	4.07106	3.63958 4.5431
	confidence	8.08725	.403154	.015271	8.07446	7.33297 8.91465
SD						
	speed	1.39362	.138421	.008743	1.39099	1.1325 1.67663
	convenience	3.23828	.227796	.008453	3.23003	2.81223 3.70924
	confidence	4.94545	.352829	.011172	4.93254	4.28186 5.67404
Corr						
	β_1 - β_2	-.718868	.093876	.005309	-.724432	-.883555 -.521713
	β_1 - β_3	-.324602	.114345	.007036	-.326473	-.543816 -.09564
	β_2 - β_3	-.058956	.075735	.001137	-.059481	-.206796 .089862

Note: Default priors are used for model parameters

As shown, the standard panel MIXL estimates are based on a subset of 540 respondents. Consequently, 28% of respondents are excluded, which roughly corresponds to the size of the garbage class in example 4. The parameter estimates are also very similar to the garbage class MIXL estimates from example 4. Conversely, the estimates markedly differ from the standard panel MIXL estimates presented in example 1, which were based on the full sample of 750 respondents.

6 Comparisons

To conclude the description of the `garbage_mixl` command, tables 1 and 2 provide a comparison of the standard and garbage MIXL model estimates as produced by the `garbage_mixl` command with estimation results obtained from the general-purpose JAGS software (Plummer 2007). In addition, table 1 provides comparisons with the other panel MIXL estimators that are available in Stata, that is, the default `cmxtmixlogit` as well as community-contributed `mixlogit` (Hole 2007) and `bayesmixedlogit` (Baker 2014) commands. All estimations were performed using Stata/MP 16.1 for Windows (x86-64) on a PC with an AMD 3950 CPU with default priors and default estimation settings for a panel MIXL model with correlated random parameters. Two changes to

the default estimation settings were made—one for the `mixlogit` command, for which the default 50 Halton draws were deemed very inadequate (1,000 Halton draws were used instead), and one for the `bayesmixedlogit` command, for which the same number of draws as in JAGS and the `garbage_mix1` command was specified. Note that JAGS model code is provided in the online supplemental and that MCMC summary statistics for JAGS and `bayesmixedlogit` were created using the R-CODA package (Plummer et al. 2006).

As shown, all commands provide similar panel MIXL estimates with the main difference being the required run time of the command (table 1). In addition, the command `garbage_mix1` and JAGS software provide nearly identical results that are within Monte Carlo sampling precision (tables 1 and 2). This is as expected given the identical priors, large burn-in, and many posterior samples and is used as a test case to confirm that the `garbage_mix1` command provides correct results.

Table 1. Standard (panel) MIXL estimates

	garbage_mixl (50k burn + 50k MCMC)		JAGS* (50k burn + 50k MCMC)		cmxmixlogit (695 Hammerley draws)		mixlogit (1,000 Halton draws)		bayesmixedlogit* (50k burn + 50k MCMC)				
	Mean	Std. dev.	MCSE	Mean	Std. dev.	Coef.	Std. err.	Coef.	Std. err.	Mean	Std. dev.	MCSE	
Speed (mean)	1.23	0.08	0.0011	1.22	0.08	0.0026	1.25	0.08	1.23	0.08	1.22	0.08	0.0030
Convenience (mean)	1.84	0.11	0.0015	1.83	0.11	0.0023	1.86	0.11	1.84	0.11	1.83	0.11	0.0030
Confidence (mean)	3.62	0.18	0.0025	3.62	0.18	0.0056	3.65	0.19	3.62	0.18	3.60	0.18	0.0056
Speed (SD)	1.13	0.09	0.0023	1.12	0.09	0.0051	1.13	0.09	1.12	0.09	1.11	0.10	0.0051
Convenience (SD)	2.04	0.12	0.0022	2.04	0.12	0.0042	2.03	0.11	2.04	0.11	2.02	0.11	0.0039
Confidence (SD)	3.37	0.19	0.0031	3.37	0.18	0.0065	3.47	0.19	3.40	0.18	3.35	0.18	0.0066
Estimation time		44 sec			12 min 0 sec		31 min 32 sec		10 min 34 sec		1 day 19 hours		

NOTE: * Summary statistics of the MCMC draws created using the R-CODA package.

Table 2. Garbage class MIXL estimates

	garbage_mixl			JAGS		
	(50k burn + 100k MCMC)			(50k burn + 100k MCMC)		
	Coef.	Std. dev.	MCSE	Coef.	Std. dev.	MCSE
Speed (mean)	3.06	0.21	0.0068	3.07	0.22	0.0181
Convenience (mean)	5.05	0.37	0.0121	5.04	0.37	0.0245
Confidence (mean)	10.5	0.74	0.0278	10.5	0.75	0.0567
Speed (SD)	1.64	0.20	0.0088	1.65	0.19	0.0133
Convenience (SD)	4.06	0.36	0.0122	4.01	0.35	0.0216
Confidence (SD)	6.32	0.56	0.0267	6.33	0.56	0.0309
Garbage class share	0.34	0.01	0.0001	0.34	0.01	0.0003
Estimation time	1 min 56 sec			24 min 31 sec		

7 Conclusions

The garbage class MIXL model was designed to produce standard panel MIXL preference estimates that are automatically corrected for the impact of survey respondents with low-quality response patterns, such as flatlining and random responses, and provide an objective measure of data quality in discrete choice experiments. As shown in the provided examples, the inclusion of garbage respondents in standard panel MIXL estimations can have a substantial impact on the MIXL model estimates and result in substantively different conclusions about respondents' preferences in a discrete choice experiment. Thus, the `garbage_mixl` estimation command presents a valuable addition to the already available MIXL model estimation commands in Stata.

One of the major benefits of the `garbage_mixl` command is its computational efficiency and robust performance, particularly for MIXL models with more random parameters. Another attractive feature is the implemented Laplace approximation optimization, which produces accurate initial values for the MCMC update algorithms within a few seconds. Moreover, the `garbage_mixl` command can fit standard panel MIXL models without a garbage class, which is convenient because the command is substantially faster than other estimation commands that can fit panel MIXL models in Stata.

The increased performance does come at a cost: The command requires an equal number of choice options per choice task, has fewer customization options, and, being based on Stata's C++ plugin functionality, the user interface cannot provide real-time information on the progress of the model estimation. In addition, to guarantee that there is enough information for each respondent and ensure that the respondent-specific garbage class-selection parameters are not entirely dictated by the Bernoulli priors, the `garbage_mixl` command requires at least as many choice tasks per respondent as random parameters when fitting a garbage class MIXL model. This restriction is not imposed for standard panel MIXL model estimations.

8 Acknowledgments

I thank Prof. Dr. Bas Donkers and Prof. Dr. Martyn Plummer for their valuable comments and suggestions on the implementation of the Bayesian sampling algorithms. In addition, I want to express my gratitude to Dr. Piers Lawrence for his assistance with the implementation of the analytical derivatives used in the Laplace approximations.

9 Programs and supplemental material

To install the software files as they existed at the time of publication of this article, type

```
. net sj 24-3
. net install st0753      (to install program files, if available)
. net get st0753         (to install ancillary files, if available)
```

10 References

- Baker, M. J. 2014. Adaptive Markov chain Monte Carlo sampling and estimation in Mata. *Stata Journal* 14: 623–661. <https://doi.org/10.1177/1536867X1401400309>.
- Bédard, M., R. Douc, and E. Moulin. 2014. Scaling analysis of delayed rejection MCMC methods. *Methodology and Computing in Applied Probability* 16: 811–838. <https://doi.org/10.1007/s11009-013-9326-y>.
- Bleistein, N., and R. A. Handelsman. 1986. *Asymptotic Expansions of Integrals*. New York: Dover.
- Gu, Y., A. R. Hole, and S. Knox. 2013. Fitting the generalized multinomial logit model in Stata. *Stata Journal* 13: 382–397. <https://doi.org/10.1177/1536867X1301300213>.
- Hole, A. R. 2007. Fitting mixed logit models by using maximum simulated likelihood. *Stata Journal* 7: 388–401. <https://doi.org/10.1177/1536867X0700700306>.
- . 2015. mixlogitwtp: Stata module to estimate mixed logit models in WTP space. Statistical Software Components S458037, Department of Economics, Boston College. <https://ideas.repec.org/c/boc/bocode/s458037.html>.
- Huang, A., and M. P. Wand. 2013. Simple marginally noninformative prior distributions for covariance matrices. *Bayesian Analysis* 8: 439–452. <https://doi.org/10.1214/13-BA815>.
- Jonker, M. F. 2022. The garbage class mixed logit model: Accounting for low-quality response patterns in discrete choice experiments. *Value in Health* 25: 1871–1877. <https://doi.org/10.1016/j.jval.2022.07.013>.
- Marshall, T., and G. O. Roberts. 2012. An adaptive approach to Langevin MCMC. *Statistics and Computing* 22: 1041–1057. <https://doi.org/10.1007/s11222-011-9276-6>.

- Plummer, M. 2007. JAGS. <http://mcmc-jags.sourceforge.net/>.
- Plummer, M., N. Best, K. Cowles, and K. Vines. 2006. CODA: Convergence diagnosis and output analysis for MCMC. *R News* 6: 7–11.
- Roberts, G. O., and J. S. Rosenthal. 2001. Optimal scaling for various Metropolis-Hastings algorithms. *Statistical Science* 16: 351–367. <https://doi.org/10.1214/ss/1015346320>.
- Small, C. G. 2010. *Expansions and Asymptotics for Statistics*. Boca Raton, FL: Chapman and Hall/CRC. <https://doi.org/10.1201/9781420011029>.
- Train, K. E. 2009. *Discrete Choice Methods with Simulation*. 2nd ed. Cambridge: Cambridge University Press. <https://doi.org/10.1017/CBO9780511805271>.

About the author

Marcel F. Jonker received his PhD in public health and epidemiology and is currently affiliated with the Erasmus School of Health Policy & Management and the Erasmus Choice Modelling Centre of the Erasmus University Rotterdam, The Netherlands.