

cmxtmixlogit — Panel-data mixed logit choice model

Description	Quick start	Menu	Syntax
Options	Remarks and examples	Stored results	Methods and formulas
References	Also see		

Description

`cmxtmixlogit` fits a mixed logit model (MLM) to choice data in which decision makers make repeated choices, typically at different time periods. Data with repeated cases from the same units are also referred to as panel data. Panel-data mixed logit models use random coefficients to model the correlation of choices across alternatives. The random coefficients are on variables that vary across alternatives (and may vary over individuals and choice sets as well), known as alternative-specific variables.

The correlation of choices across alternatives relaxes the independence of irrelevant alternatives (IIA) assumption required by the conventional multinomial logit model, fit by `mlogit`, and the choice-model conditional logit model, fit by `cmclogit`.

In the context of panel-data applications, the MLM models the probability of selecting each alternative for each time period rather than modeling a single probability for selecting each alternative, as in the case of cross-sectional data. Mixed logit models for cross-sectional data are fit by `cmixlogit`.

Quick start

Mixed logit regression for panel data of y on x_1 , where the coefficients on x_1 are assumed random normal, using `cmset` data

```
cmxtmixlogit y, random(x1)
```

Same as above, and include levels of case-specific covariate a

```
cmxtmixlogit y, random(x1) casevars(i.a)
```

Same as above, and include lag of alternative-specific variable x_2 with triangular distributed random coefficients

```
cmxtmixlogit y, random(x1) random(L.x2, triangle) casevars(i.a)
```

Same as above, and include alternative-specific variable x_3 with fixed coefficient

```
cmxtmixlogit y x3, random(x1) random(L.x2, triangle) casevars(i.a)
```

Menu

Statistics > Choice models > Panel-data mixed logit model

Syntax

```
cmxtmixlogit depvar [indepvars] [if] [in] [weight] [, options]
```

depvar equal to 1 identifies the chosen alternatives, whereas a 0 indicates the alternatives that were not chosen. There can be only one chosen alternative for each case.

indepvars specifies the alternative-specific covariates that have fixed coefficients.

<i>options</i>	Description
Model	
<u>casevars</u> (<i>varlist</i>)	choice set-specific variables
<u>random</u> (<i>varlist</i> [, <i>distribution</i>])	specify variables that are to have random coefficients and the coefficients' distribution
<u>corrmetric</u> (<i>metric</i>)	correlation metric for correlated random coefficients
<u>basealternative</u> (# <i>lbl</i> <i>str</i>)	alternative used for normalizing location
<u>noconstant</u>	suppress the alternative-specific constant terms
<u>altwise</u>	use alternativewise deletion instead of casewise deletion
<u>constraints</u> (<i>constraints</i>)	apply specified linear constraints
SE/Robust	
<u>vce</u> (<i>vcetype</i>)	<i>vcetype</i> may be <u>oim</u> , <u>robust</u> , <u>cluster</u> <i>clustvar</i> , <u>opg</u> , <u>bootstrap</u> , or <u>jackknife</u>
Reporting	
<u>level</u> (#)	set confidence level; default is <u>level</u> (95)
<u>or</u>	report odds ratios and relative-risk ratios
<u>nocnsreport</u>	do not display constraints
<u>display_options</u>	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Integration	
<u>intmethod</u> (<i>seqspec</i>)	specify point set for Monte Carlo integration
<u>intpoints</u> (#)	specify number of points in each sequence
<u>intburn</u> (#)	specify starting index in the Hammersley or Halton sequence
<u>intseed</u> (#)	specify random-number seed for pseudo-random sequence
<u>favor</u> (<u>speed</u> <u>space</u>)	favor speed or space when generating integration points
Maximization	
<u>maximize_options</u>	control the maximization process; seldom used
<u>collinear</u>	keep collinear variables
<u>coeflegend</u>	display legend instead of statistics

<i>distribution</i>	Description
<code>normal</code>	Gaussian-distributed random coefficients; the default
<code>correlated</code>	correlated Gaussian-distributed random coefficients
<code>lnormal</code>	lognormal distributed random coefficients
<code>tnormal</code>	truncated normal distributed random coefficients
<code>uniform</code>	uniform distributed random coefficients
<code>triangle</code>	triangular distributed random coefficients

<i>metric</i>	Description
<code>correlation</code>	standard deviation and correlation; the default
<code>covariance</code>	variance and covariance
<code>cholesky</code>	Cholesky factor

seqspec is

```
seqtype [ , antithetics | mantithetics ]
```

<i>seqtype</i>	Description
<code>hammersley</code>	Hammersley point set; the default
<code>halton</code>	Halton point set
<code>random</code>	uniform pseudo-random point set

You must `cmset` your data before using `cmxtmixlogit`; see [CM] [cmset](#).

`indepvars` and `varlist` may contain factor variables and time-series operators; see [U] [11.4.3 Factor variables](#) and [U] [11.4.4 Time-series varlists](#).

`bootstrap`, `by`, `collect`, `jackknife`, `statsby`, and `svy` are allowed; see [U] [11.1.10 Prefix commands](#).

Weights are not allowed with the `bootstrap` prefix; see [R] [bootstrap](#).

`vce()` and weights are not allowed with the `svy` prefix; see [SVY] [svy](#).

`fweights`, `iwweights`, and `pweights` are allowed; see [U] [11.1.6 weight](#).

`collinear` and `coeflegend` do not appear in the dialog box.

See [U] [20 Estimation and postestimation commands](#) for more capabilities of estimation commands.

Options

Model

`casevars`(*varlist*) specifies the case-specific variables that are constant for each case, that is, choice set. If there are a maximum of A alternatives, there will be $A - 1$ sets of coefficients associated with `casevars()`.

`random`(*varlist* [, *distribution*]) specifies the alternative-specific variables that are to have random coefficients and, optionally, the assumed distribution of the random coefficients. The default distribution is `normal`, meaning Gaussian-distributed random coefficients. *distribution* may also be `correlated`, `lnormal`, `tnormal`, `uniform`, or `triangle`. `random()` may be specified more than once to specify different sets of variables that correspond to different coefficient distributions.

`corrmetric(metric)` specifies the estimation metric for correlated random coefficients. `corrmetric(correlation)`, the default, estimates the standard deviations and correlations of the random coefficients. `corrmetric(covariance)` estimates variances and covariances, and `corrmetric(cholesky)` estimates Cholesky factors. `corrmetric()` is allowed only when `random(varlist, correlated)` is specified.

`basealternative(#|lbl|str)` sets the alternative that normalizes the level of utility. The base alternative may be specified as a number when the alternatives variable is numeric, as a label when it is numeric and has a [value label](#), or as a string when it is a string variable. The default is the alternative with the highest frequency of being chosen. This option is ignored if neither alternative-specific constants nor case-specific variables are specified.

`noconstant` suppresses the $A - 1$ alternative-specific constant terms.

`altwise` specifies that alternatively deletion be used when omitting observations because of missing values in your variables. The default is to use casewise deletion; that is, the entire group of observations making up a case is omitted if any missing values are encountered. This option does not apply to observations that are excluded by the `if` or `in` qualifier or the `by` prefix; these observations are always handled alternatively regardless of whether `altwise` is specified.

`constraints(constraints)`; see [\[R\] Estimation options](#).

SE/Robust

`vce(vcetype)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`oim`, `opg`), that are robust to some kinds of misspecification (`robust`), that allow for intragroup correlation (`cluster clustvar`), and that use bootstrap or jackknife methods (`bootstrap`, `jackknife`); see [\[R\] *vce* option](#).

Specifying `vce(robust)` is equivalent to specifying `vce(cluster caseid)`.

If specifying `vce(bootstrap)` or `vce(jackknife)`, you must also specify `basealternative()`.

Reporting

`level(#)`; see [\[R\] Estimation options](#).

`or` reports the estimated coefficients transformed to odds ratios for alternative-specific variables and relative-risk ratios for case-specific variables. That is, e^b rather than b is reported. Standard errors and confidence intervals are transformed accordingly. This option affects how results are displayed, not how they are estimated. `or` may be specified at estimation or when replaying previously estimated results.

`nocnsreport`; see [\[R\] Estimation options](#).

`display_options`: `noci`, `nopvalues`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(style)`, `cformat(%fmt)`, `pformat(%fmt)`, `sformat(%fmt)`, and `nolstretch`; see [\[R\] Estimation options](#).

Integration

`intmethod(seqtype [, antithetics|mantithetics])` specifies the method of generating the point sets used in the Monte Carlo integration. `intmethod(hammersley)`, the default, uses the Hammersley sequence; `intmethod(halton)` uses the Halton sequence; and `intmethod(random)` uses a sequence of uniform random numbers.

`antithetics` and `mantithetics` specify that a unidimensional antithetic sequence or a multidimensional antithetic sequence, respectively, be generated instead of the standard implementation of the requested `seqtype`. These methods improve the accuracy of the Monte Carlo integration at the cost of additional computation time; see [Methods and formulas](#).

`intpoints(#)` specifies the number of points to use in the Monte Carlo integration. The default number of points is a function of model complexity and integration method. If `intmethod(hammersley)` or `intmethod(halton)` is used, the default is $500 + \text{floor}[2.5\sqrt{N_c}\{\ln(r+5) + v\}]$, where N_c is the number of panels, r is the number of random coefficients in the model, and v is the number of variance parameters. If `intmethod(hammersley, mantithetics)` or `intmethod(halton, mantithetics)` is used, the number of integration points is $250 + \text{floor}[0.5\sqrt{N_c}\{\ln(r+5) + v\}]$. If `intmethod(random)` is used, the number of points is twice the number of points used by `intmethod(hammersley)` and `intmethod(halton)`. Larger values of `intpoints()` provide better approximations of the log likelihood at the cost of additional computation time.

`intburn(#)` specifies where in the Hammersley or Halton sequence to start, which helps reduce the correlation between the sequences of each dimension. The default is to discard the first n initial elements from each sequence, where n is the largest prime used to generate the sequences. This option may not be specified with `intmethod(random)`.

`intseed(#)` specifies the seed to use for generating uniform pseudo-random sequences. This option may be specified only with `intmethod(random)`. `#` must be an integer between 0 and $2^{31} - 1$. The default is to use the current seed value from Stata's uniform random-number generator; see [\[R\] set seed](#).

`favor(speed|space)` instructs `cmxtmixlogit` to favor either speed or space when generating the integration points. `favor(speed)` is the default. When favoring speed, the integration points are generated once and stored in memory, thus increasing the speed of evaluating the likelihood. This speed increase can be seen when there are many cases or when the user specifies many integration points in `intpoints(#)`. When favoring space, the integration points are generated repeatedly with each likelihood evaluation.

Maximization

`maximize_options`: `difficult`, `technique(algorithm_spec)`, `iterate(#)`, `[no]log`, `trace`, `gradient`, `showstep`, `hessian`, `showtolerance`, `tolerance(#)`, `ltolerance(#)`, `nrtolerance(#)`, `nonrtolerance`, and `from(init_specs)`; see [\[R\] Maximize](#).

Setting the optimization type to `technique(bhhh)` resets the default `vcetype` to `vce(opg)`.

The following options are available with `cmxtmixlogit` but are not shown in the dialog box: `collinear`, `coeflegend`; see [\[R\] Estimation options](#).

Remarks and examples

[stata.com](#)

`cmxtmixlogit` fits a mixed logit model to panel data. The mixed logit model is most frequently used to model the probability that an individual chooses one of several unordered alternatives. It is also known as the mixed multinomial logit model (McFadden and Train 2000), the random-parameters logit model (Cameron and Trivedi 2005), the logit kernel model (Ben-Akiva, Bolduc, and Walker 2001), or the hybrid logit model (Ben-Akiva et al. 1997). For an introduction to mixed logit models, see [\[CM\] cmmixlogit](#).

The panel-data mixed logit model is used in the case where a decision maker makes repeated choices over time, and it models the probability of selecting each alternative at each time point. See chapter 12 of Rabe-Hesketh and Skrondal (2022) for applications using Stata.

▷ Example 1: Panel-data mixed logit model with alternative- and case-specific covariates

We have created fictitious data for an individual's choice of transportation mode over multiple occasions. There are also data on his or her cost of transportation and travel time for each mode, as well as information about his or her income and age. We wish to examine how these factors affect the choice of transportation alternative.

`transport.dta` records information about the choices of primary transportation made by 500 individuals to their workplace in a metropolitan area. Each individual chose between four modes of transportation at three points in time. The alternatives are recorded in the variable `alt`.

The binary variable `choice` records the chosen alternative; `choice` is 1 for the chosen alternative and 0 otherwise. For each individual, we have one choice set per time, where `t` is the time variable. In the following, we refer to a choice set as a case. Here are the data for the first individual in the dataset:

```
. use https://www.stata-press.com/data/r18/transport
(Transportation choice data)
. list in 1/12, sepby(t)
```

	id	t	alt	choice	trcost	trtime	age	income	parttime
1.	1	1	Car	1	4.14	0.13	3.0	3	Full-time
2.	1	1	Public	0	4.74	0.42	3.0	3	Full-time
3.	1	1	Bicycle	0	2.76	0.36	3.0	3	Full-time
4.	1	1	Walk	0	0.92	0.13	3.0	3	Full-time
5.	1	2	Car	1	8.00	0.14	3.2	5	Full-time
6.	1	2	Public	0	3.14	0.12	3.2	5	Full-time
7.	1	2	Bicycle	0	2.56	0.18	3.2	5	Full-time
8.	1	2	Walk	0	0.64	0.39	3.2	5	Full-time
9.	1	3	Car	1	1.76	0.18	3.4	5	Part-time
10.	1	3	Public	0	2.25	0.50	3.4	5	Part-time
11.	1	3	Bicycle	0	0.92	1.05	3.4	5	Part-time
12.	1	3	Walk	0	0.58	0.59	3.4	5	Part-time

This person chose to use the car at all three points in time.

Travel time (`trtime`, measured in hours) and cost of travel (`trcost`, measured in \$) are alternative-specific variables that vary over alternatives, cases, and individuals. Variables `age` (measured in decades), `income` (annual income measured in \$10,000), and `parttime` (variable indicating a part-time job) are case-specific variables that vary both over cases and individuals but are constant across alternatives for a given individual and time.

Before we can fit the model, we will need to `cmset` the data. To specify that `id` identifies the individuals, that `t` identifies time, and that `alt` identifies the alternatives available to each individual, we type the following:

```
. cmset id t alt
note: case identifier _caseid generated from id and t.
note: panel by alternatives identifier _panelaltid generated from id and alt.

      Panel data: Panels id and time t
      Case ID variable: _caseid
      Alternatives variable: alt
      Panel by alternatives variable: _panelaltid (strongly balanced)
      Time variable: t, 1 to 3
      Delta: 1 unit

Note: Data have been xtset.
```

The output informs us that the panels are strongly balanced, which is to say that each individual made a choice in each of the same periods. The output also says that two new variables were created, `_caseid` and `_panelaltid`. See [example 2](#) in [\[CM\] cmset](#) for an explanation of the purpose of these new variables. You need not be concerned about them, just leave them in your dataset.

We can now use `cmxtmixlogit` to fit our model. We wish to estimate the effect of travel cost (`trcost`), travel time (`trtime`), income, and age on the choice of transportation mode. We assume that all individuals have the same preferences with respect to travel cost, but that preferences with respect to travel time are heterogeneous, and we model these heterogeneous preferences with a random coefficient for `trtime`.

We specify that `choice` contains the chosen alternatives and that `trcost` is included in the model with a fixed coefficient. Specifying `random(trtime)` includes `trtime` with a normally distributed random coefficient—the default distribution of random coefficients. Specifying `casevars(age income)` includes the case-specific variables `age` and `income` in the model with fixed coefficients.

8 cmxtmixlogit — Panel-data mixed logit choice model

```

. cmxtmixlogit choice trcost, random(trtime) casevars(age income)
Fitting fixed parameter model:
Fitting full model:
Iteration 0: Log simulated-likelihood = -1025.707 (not concave)
Iteration 1: Log simulated-likelihood = -1014.2513
Iteration 2: Log simulated-likelihood = -1006.2212
Iteration 3: Log simulated-likelihood = -1005.9904
Iteration 4: Log simulated-likelihood = -1005.9899
Iteration 5: Log simulated-likelihood = -1005.9899
Mixed logit choice model          Number of obs      =      6,000
                                Number of cases     =      1,500
Panel variable: id                Number of panels   =       500
Time variable: t                  Cases per panel:  min =        3
                                avg =       3.0
                                max =        3
Alternatives variable: alt        Alts per case:    min =        4
                                avg =       4.0
                                max =        4
Integration sequence:             Hammersley
Integration points:               594
Log simulated-likelihood = -1005.9899
                                Wald chi2(8)        =      432.68
                                Prob > chi2         =      0.0000

```

choice	Coefficient	Std. err.	z	P> z	[95% conf. interval]	
alt						
trcost	-.8388216	.0438587	-19.13	0.000	-.9247829	-.7528602
trtime	-1.508756	.2641554	-5.71	0.000	-2.026492	-.9910212
/Normal						
sd(trtime)	1.945596	.2594145			1.498161	2.526661
Car (base alternative)						
Public						
age	.1538915	.0672638	2.29	0.022	.0220569	.2857261
income	-.3815444	.0347459	-10.98	0.000	-.4496451	-.3134437
_cons	-.5756547	.3515763	-1.64	0.102	-1.264732	.1134222
Bicycle						
age	.20638	.0847655	2.43	0.015	.0402426	.3725174
income	-.5225054	.0463235	-11.28	0.000	-.6132978	-.4317131
_cons	-1.137393	.4461318	-2.55	0.011	-2.011795	-.2629909
Walk						
age	.3097417	.1069941	2.89	0.004	.1000372	.5194463
income	-.9016697	.0686042	-13.14	0.000	-1.036132	-.7672078
_cons	-.4183279	.5607111	-0.75	0.456	-1.517302	.6806458

The output header shows that our model was fit using data on 500 panels. Each panel is an individual in our example. Each panel/person made a choice in each of 3 time periods, so there are 1,500 cases. Because every case has 4 alternatives, there are 6,000 observations in total.

The estimate of the fixed coefficient on `trcost` is -0.84 . The estimated mean of the random coefficients on `trtime` is -1.51 . The estimated standard deviation of these random coefficients is 1.95 , and there is significant heterogeneity in these coefficients over the individuals in the population.

Each case-specific variable has different coefficients for each alternative. These coefficients are interpreted similarly to conventional multinomial logit. For example, that the estimated coefficients on `age` for the `Public` and `Bicycle` alternatives are positive when `Car` is the base alternative implies that an increase in age makes it more likely that someone will choose public transportation or a bicycle over a car. Similarly, an increase in income makes it less likely that someone will choose public transportation or a bicycle over a car.

After you have fit what you consider your final model, you should run the same model again, but this time setting `intpoints(#)`, the number of integration points in the simulated likelihood to a larger number. In this example, we see from the header that the default number of points was 594. We would run our model again using, say, 1000 points and see by how much the coefficient or covariance parameter estimates change. If changes are small compared with standard errors, we can have confidence in the numerical soundness of the simulation used to compute the likelihood. See [Setting the number of integration points](#) in [CM] [Intro 5](#) for more information.

◀

▷ Example 2: Expected choice probabilities

The choice probabilities are a nonlinear function of the estimated parameters, which complicates parameter interpretation. We can, however, use `margins` to estimate the effects of variables.

We wish to estimate expected choice probabilities for specified income levels. We estimate average probabilities of each alternative when every individual has an income of \$30,000 and when every individual has an income of \$80,000. The observed values are used for all other variables.

```
. margins, at(income=(3 8))
Predictive margins                                Number of obs = 6,000
Model VCE: OIM
Expression: Pr(alt), predict()
1._at: income = 3
2._at: income = 8
```

	Delta-method		z	P> z	[95% conf. interval]	
	Margin	std. err.				
._outcome#_at						
Car#1	.3331611	.0196734	16.93	0.000	.294602	.3717203
Car#2	.7009364	.0109829	63.82	0.000	.6794103	.7224626
Public#1	.2210964	.0184285	12.00	0.000	.1849772	.2572156
Public#2	.1801593	.0091811	19.62	0.000	.1621646	.1981539
Bicycle#1	.1676081	.0181511	9.23	0.000	.1320325	.2031837
Bicycle#2	.0862692	.0080813	10.68	0.000	.0704302	.1021081
Walk#1	.2781343	.0243791	11.41	0.000	.2303521	.3259166
Walk#2	.0326352	.0058942	5.54	0.000	.0210827	.0441877

If every individual had an income of \$30,000, 33% of them would choose a car. On the other hand, if everybody had an income of \$80,000, 70% would choose a car. The results for `Walk` are reversed. If every individual had an income of \$30,000, 28% of them would choose to walk. If every individual had an income of \$80,000, only 3% of them would choose to walk.

In the above example, we averaged the choice probabilities over all time periods. This works well for time-invariant effects, but we could be interested in having our predictions evaluated for certain points in time. In the following example, we estimate the same averaged predictions as in the previous example but only for the first time period. We do this by specifying `subpop(if t==1)`.

```
. margins, at(income=(3 8)) subpop(if t==1)
Predictive margins                                Number of obs   = 6,000
Model VCE: OIM                                    Subpop. no. obs = 2,000
Expression: Pr(alt), predict()
1._at: income = 3
2._at: income = 8
```

	Delta-method				[95% conf. interval]	
	Margin	std. err.	z	P> z		
<code>_outcome#_at</code>						
Car#1	.3280809	.0193673	16.94	0.000	.2901217	.3660401
Car#2	.6910464	.0110758	62.39	0.000	.6693381	.7127546
Public#1	.2292844	.0188376	12.17	0.000	.1923633	.2662054
Public#2	.1892283	.0094331	20.06	0.000	.1707398	.2077168
Bicycle#1	.1723129	.0184627	9.33	0.000	.1361265	.2084992
Bicycle#2	.0880065	.0083066	10.59	0.000	.0717258	.1042872
Walk#1	.2703219	.0238268	11.35	0.000	.2236222	.3170215
Walk#2	.0317188	.005727	5.54	0.000	.0204942	.0429434

Here the results are relatively similar to the ones we obtained by averaging over all time periods.

◀

► Example 3: Contrasts of marginal predictions

The [previous example](#) estimated averages for two different scenarios at one point in time. Based on this model, and assuming we have a random or otherwise representative sample, we can interpret the difference in these average probabilities as the effect of having an income of \$80,000 instead of an income of \$30,000. We can estimate this difference using the `contrast()` option. We request this for each time point separately by including the `over(t)` option.

```

. margins, at(income=(3 8)) contrast(at(r) nowald) over(t)
Contrasts of predictive margins                                Number of obs = 6,000
Model VCE: OIM
Expression: Pr(alt), predict()
Over:      t
1._at: 1.t
           income = 3
1._at: 2.t
           income = 3
1._at: 3.t
           income = 3
2._at: 1.t
           income = 8
2._at: 2.t
           income = 8
2._at: 3.t
           income = 8

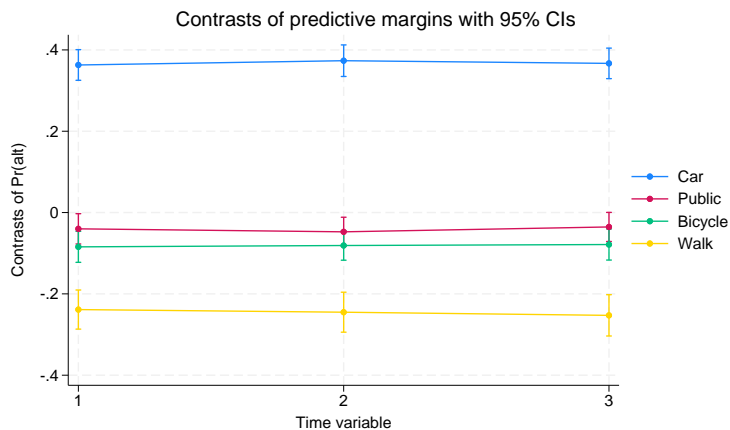
```

	Delta-method		
	Contrast	std. err.	[95% conf. interval]
_at@_outcome#t			
(2 vs 1) Car#1	.3629654	.0192268	.3252817 .4006492
(2 vs 1) Car#2	.3734991	.0197989	.334694 .4123042
(2 vs 1) Car#3	.3668613	.0191453	.3293372 .4043854
(2 vs 1) Public#1	-.040056	.0189899	-.0772756 -.0028365
(2 vs 1) Public#2	-.0473747	.0183016	-.0832451 -.0115042
(2 vs 1) Public#3	-.0353807	.0182474	-.071145 .0003835
(2 vs 1) Bicycle#1	-.0843064	.0194643	-.1224557 -.046157
(2 vs 1) Bicycle#2	-.0810573	.018425	-.1171697 -.0449449
(2 vs 1) Bicycle#3	-.0786532	.0194817	-.1168366 -.0404699
(2 vs 1) Walk#1	-.2386031	.0245353	-.2866913 -.1905148
(2 vs 1) Walk#2	-.2450671	.0250718	-.2942069 -.1959274
(2 vs 1) Walk#3	-.2528274	.0259127	-.3036154 -.2020394

For example, at the first time point, we estimate that the effect of making \$80,000 instead of \$30,000 is an increase of 0.36 in the probability of choosing to travel by car.

We can also plot our results by using `marginsplot`.

```
. marginsplot
Variables that uniquely identify margins: t _outcome
```



Here we have the differences in probabilities on the y axis and time on the x axis. Differences over time are shown for each of the alternatives. We see that there is no substantial change over time in the effect of making \$80,000 instead of \$30,000.

◀

► Example 4: Marginal predictions with alternative-specific variables

We now estimate the effect of an alternative-specific covariate. Changing the value on one alternative-specific covariate affects the choice probabilities of all the alternatives. For example, an increase in the cost of car travel changes the probabilities of choosing Car, Public, Bicycle, or Walk. We estimate expected probabilities of choosing each outcome under two scenarios. The first scenario is the as-is scenario: all variables are as observed. In the second scenario, the cost of car travel increases by 25%. We do this by specifying the option `alternative(Car)` to tell `margins` that we wish to apply our `at()` scenarios for the alternative-specific variable `trcost` to the Car alternative.

```
. margins, alternative(Car) at(trcost = generate(trcost))
> at(trcost = generate(1.25*trcost)) subpop(if t==1)

Predictive margins                                Number of obs   = 6,000
Model VCE: OIM                                    Subpop. no. obs = 2,000

Expression: Pr(alt), predict()
Alternative: Car

1._at: trcost =      trcost
2._at: trcost = 1.25*trcost
```

	Delta-method				
	Margin	std. err.	z	P> z	[95% conf. interval]
_outcome#_at					
Car#1	.5439062	.0113994	47.71	0.000	.5215638 .5662486
Car#2	.4405694	.0101017	43.61	0.000	.4207704 .4603683
Public#1	.2010082	.0104382	19.26	0.000	.1805497 .2214668
Public#2	.2548516	.0117988	21.60	0.000	.2317264 .2779769
Bicycle#1	.1255662	.0095539	13.14	0.000	.1068409 .1442914
Bicycle#2	.1566796	.0110237	14.21	0.000	.1350736 .1782856
Walk#1	.1295194	.0101536	12.76	0.000	.1096187 .1494201
Walk#2	.1478994	.0110109	13.43	0.000	.1263185 .1694803

The average probability of choosing car travel falls from 0.54 to 0.44 when we move from the current situation to one in which the cost of car travel increases by 25%. About half of those that would stop selecting car travel would move to public transportation; the expected probability of selecting public transportation increases from 0.20 to 0.25.



► Example 5: Average marginal effects

A marginal effect is the change in the probability of choosing a certain alternative as a function of an infinitesimal change in a covariate at a given point. In other words, it is the slope of the choice probability with respect to a covariate at a given point of the covariate. The average marginal effect is the average of the marginal effects over all the observations.

As in Example 4, a change to one alternative-specific covariate affects the choice probabilities of all alternatives. In our example, there are four marginal effects for the cost of Car, four marginal effects for the cost of Public, four marginal effects for the cost of Bicycle, and four marginal effects for the cost of Walk.

In the context of discrete choice models with alternative-specific variables, we distinguish between direct and indirect marginal effects. The marginal effects of an alternative-specific variable on its own choice probability is called a direct marginal effect. The marginal effects of an alternative-specific variable on the choice probabilities of the other alternatives are called indirect marginal effects.

We now estimate the direct and the indirect marginal effects of travel time for Car.

```
. margins, dydx(trtime) alternative(Car)
Average marginal effects                                Number of obs = 6,000
Model VCE: OIM
Expression: Pr(alt), predict()
Alternative: Car
dy/dx wrt: trtime
```

	Delta-method		z	P> z	[95% conf. interval]	
	dy/dx	std. err.				
trtime						
_outcome						
Car	-.1581844	.0269102	-5.88	0.000	-.2109275	-.1054414
Public	.1055447	.0171745	6.15	0.000	.0718834	.139206
Bicycle	.0374872	.0073318	5.11	0.000	.0231171	.0518573
Walk	.0151526	.0043034	3.52	0.000	.006718	.0235871

The estimate of the direct average marginal effect is around -0.16 , suggesting a decrease in the probability of choosing a car if car travel time increases. The estimated indirect marginal effects are all positive, with the effect on Public being larger than those on Bicycle and on Walk.

◀

□ Technical note

Although `cmxtmixlogit` follows a random-effects approach, the model does not actually contain a separately identifiable variance component beyond the error components of the random coefficients. Rather, the extra variance due to panel random effects is contained in the error components associated with the random coefficients. No fixed-effects estimator is available for the mixed logit model.

□

Stored results

`cmxtmixlogit` stores the following in `e()`:

Scalars

<code>e(N)</code>	number of observations
<code>e(N_case)</code>	number of cases
<code>e(N_panel)</code>	number of panels
<code>e(N_ic)</code>	N for Bayesian information criterion (BIC)
<code>e(N_clust)</code>	number of clusters
<code>e(k)</code>	number of parameters
<code>e(k_alt)</code>	number of alternatives
<code>e(k_casevars)</code>	number of case-specific variables
<code>e(k_eq)</code>	number of equations in <code>e(b)</code>
<code>e(k_eq_model)</code>	number of equations in overall model test
<code>e(df_m)</code>	model degrees of freedom
<code>e(df_c)</code>	degrees of freedom, comparison test
<code>e(ll)</code>	log simulated-likelihood
<code>e(ll_c)</code>	log likelihood, comparison model
<code>e(const)</code>	constant indicator
<code>e(intpoints)</code>	number of raw integration points
<code>e(lsequence)</code>	length of each integration sequence
<code>e(intburn)</code>	starting sequence index

e(chi2)	χ^2
e(chi2_c)	χ^2 , comparison test
e(p)	<i>p</i> -value for model test
e(p_c)	<i>p</i> -value for comparison test
e(alt_min)	minimum number of alternatives
e(alt_avg)	average number of alternatives
e(alt_max)	maximum number of alternatives
e(t_avg)	average number of time points per panel
e(t_min)	minimum number of time points per panel
e(t_max)	maximum number of time points per panel
e(rank)	rank of $e(V)$
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

e(cmd)	cmxtmixlogit
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(caseid)	name of case ID variable
e(timevar)	name of time variable
e(altvar)	name of alternatives variable
e(alteqs)	alternative equation names
e(alt#)	alternative labels
e(base)	base alternative
e(corrmetric)	correlation metric for correlated random coefficients
e(wtype)	weight type
e(wexp)	weight expression
e(marktype)	casewise or altwise, type of markout
e(key_N_ic)	cases, key for <i>N</i> for Bayesian information criterion (BIC)
e(title)	title in estimation output
e(clustvar)	name of cluster variable
e(chi2type)	type of χ^2
e(vce)	<i>vctype</i> specified in <i>vce()</i>
e(vcetype)	title used to label Std. err.
e(opt)	type of optimization
e(which)	max or min; whether optimizer is to perform maximization or minimization
e(ml_method)	type of ml method
e(intmethod)	technique used to generate sequences
e(sequence)	type of sequences
e(mc_rngstate)	random-number state used
e(user)	name of likelihood-evaluator program
e(technique)	maximization technique
e(properties)	b V
e(predict)	program used to implement predict
e(marginsok)	predictions allowed by margins
e(marginsnotok)	predictions disallowed by margins
e(asbalanced)	factor variables <i>fvset</i> as <i>asbalanced</i>
e(asobserved)	factor variables <i>fvset</i> as <i>asobserved</i>

Matrices

e(b)	coefficient vector
e(altvals)	alternative values
e(Cns)	constraints matrix
e(ilog)	iteration log (up to 20 iterations)
e(gradient)	gradient vector
e(V)	variance-covariance matrix of the estimators
e(V_modelbased)	model-based variance

Functions

e(sample)	marks estimation sample
-----------	-------------------------

In addition to the above, the following is stored in `r()`:

Matrices	
<code>r(table)</code>	matrix containing the coefficients with their standard errors, test statistics, p -values, and confidence intervals

Note that results stored in `r()` are updated when the command is replayed and will be replaced when any `r-class` command is run after the estimation command.

Methods and formulas

For the mixed logit model, the utility that individual i receives from alternative a at time t , denoted by U_{iat} , is

$$U_{iat} = \mathbf{x}_{iat}\boldsymbol{\beta}_i + \mathbf{w}_{iat}\boldsymbol{\alpha} + \mathbf{z}_{it}\boldsymbol{\delta}_a + \epsilon_{iat} \quad a = 1, 2, \dots, A$$

$\boldsymbol{\beta}_i$ are random coefficients that vary over individuals in the population, and \mathbf{x}_{iat} is a vector of alternative-specific variables. $\boldsymbol{\alpha}$ are fixed coefficients on \mathbf{w}_{iat} , a vector of alternative-specific variables. $\boldsymbol{\delta}_a$ are fixed, alternative-specific coefficients on \mathbf{z}_{it} , a vector of case-specific variables. ϵ_{iat} is a random term that follows a type I extreme value distribution. `cmxtmixlogit` estimates the fixed coefficients $\boldsymbol{\alpha}$ and $\boldsymbol{\delta}_a$ and the parameters of $f(\boldsymbol{\beta})$, the distribution of the random coefficients.

`cmxtmixlogit` estimates the parameters of the mixed logit model by maximum simulated likelihood (MSL). The probability that case i chooses alternative a at time t , conditional on the random parameter $\boldsymbol{\beta}_i$, is

$$P_{iat}(\boldsymbol{\beta}) = \frac{e^{\mathbf{x}_{iat}\boldsymbol{\beta}_i + \mathbf{w}_{iat}\boldsymbol{\alpha} + \mathbf{z}_{it}\boldsymbol{\delta}_a}}{\sum_{a=1}^A e^{\mathbf{x}_{iat}\boldsymbol{\beta}_i + \mathbf{w}_{iat}\boldsymbol{\alpha} + \mathbf{z}_{it}\boldsymbol{\delta}_a}}$$

We get the unconditional choice probability, P_{iat} , by integrating over the mixing distribution $f(\boldsymbol{\beta})$:

$$P_{iat} = \int P_{iat}(\boldsymbol{\beta})f(\boldsymbol{\beta})d\boldsymbol{\beta} \quad (1)$$

This integral of dimension d , where d equals the number of random parameters, is approximated by simulation. The simulated likelihood for the i th case is

$$L_i = \prod_{t=1}^T \sum_{a=1}^A d_{iat} \hat{P}_{iat}$$

where d_{iat} is an indicator that takes on the value 1 for the chosen alternative at time t and 0 otherwise. The overall log simulated-likelihood is then $\sum_{i=1}^N \ln(L_i)$. \hat{P}_{iat} are the simulated probabilities

$$\hat{P}_{iat} = \frac{1}{M} \sum_{m=1}^M P_{iat}(\boldsymbol{\beta}^m) \quad (2)$$

where $\boldsymbol{\beta}^m$ are the random parameters drawn from $f(\boldsymbol{\beta})$ and M is the number of random draws. Equation (2) is the computation used to approximate the probabilities in (1).

Computation of \hat{P}_{iat} is based on integration sequences where each point of the sequence is a draw from density $f(\boldsymbol{\beta})$. The underlying uniform sequences are either pseudo-random draws from the uniform density or deterministic sequences such as a Halton sequence. Using deterministic sequences leads to better coverage of the probability space and lower variance of the simulator and thus having a smaller approximation error than pseudo-random sequences, given the same number of draws. `cmxtmixlogit` supports pseudo-random, Halton, and Hammersley sequences; see [Drukker and Gates \(2006\)](#) for details.

Using a higher M in (2) will produce a better approximation to the probabilities in (1), at the cost of increased computation time. M is a function of the number of raw integration points q , which may be specified using the `intpoints()` option. In the default method, $M = q$ is the number of draws used in the approximation in (2). In addition to the default method, `cmxtmixlogit` supports methods in which the draws are symmetric around a midpoint, known as unidimensional and multidimensional antithetic draws. These antithetic methods produce a better approximation to the probabilities in (1), at the cost of additional computation time; see Train (2009, sec. 9.3.1). For unidimensional antithetics, $M = 2q$ draws are used. For multidimensional antithetics on a problem with d random coefficients, $M = 2^d q$ draws are used.

Random coefficients with mean μ and scale parameter σ are simulated as follows:

$$\begin{aligned}\beta_{\text{normal}}^m &= \mu + \sigma\eta_i & \eta_i &\sim N(0, 1) \\ \beta_{\text{lognormal}}^m &= \exp(\mu + \sigma\eta_i) & \eta_i &\sim N(0, 1) \\ \beta_{\text{truncated_normal}}^m &= \mu + \sigma\eta_i & \eta_i &\sim \text{TN}(0, 1, -1.96, 1.96) \\ \beta_{\text{uniform}}^m &= \mu + \sigma\eta_i & \eta_i &\sim U(-1, 1) \\ \beta_{\text{triangular}}^m &= \mu + \sigma\eta_i & \eta_i &\sim \Delta(-1, 1)\end{aligned}$$

where $N(\mu, \sigma^2)$ is the normal distribution, $\text{TN}(\mu, \sigma^2, a, b)$ is the truncated normal distribution with lower truncation point a and upper truncation point b , $U(a, b)$ is uniform over $[a, b]$, and $\Delta(a, b)$ is the triangular distribution over $[a, b]$.

Correlated random parameters drawn from the multivariate normal distribution are generated as $\beta_{\text{MVN}}^m = \mathbf{M} + \mathbf{L}\eta_i$, where \mathbf{M} is a vector of means, $\eta_i \sim N(\mathbf{0}, \mathbf{I})$, and \mathbf{L} is the Cholesky factor such that the variance-covariance matrix $\mathbf{V} = \mathbf{L}\mathbf{L}'$.

This command supports the clustered version of the Huber/White/sandwich estimator of the variance using `vce(robust)` and `vce(cluster clustvar)`. See [P] `_robust`, particularly *Maximum likelihood estimators* and *Methods and formulas*. Specifying `vce(robust)` is equivalent to specifying `vce(cluster panelvar)`, where `panelvar` is the variable that identifies the panels.

References

- Ben-Akiva, M., D. Bolduc, and J. Walker. 2001. Specification, identification, and estimation of the logit kernel (or continuous mixed logit) model. Manuscript, University of California, Berkeley. <http://eml.berkeley.edu/reprints/misc/multinomial2.pdf>.
- Ben-Akiva, M., D. L. McFadden, M. Abe, U. Böckenholt, D. Bolduc, D. Gopinath, T. Morikawa, V. Ramaswamy, V. Rao, D. Revelt, and D. Steinberg. 1997. Modeling methods for discrete choice analysis. *Marketing Letters* 8: 273–286. <https://doi.org/10.1023/A:1007956429024>.
- Brownstone, D., and K. E. Train. 1998. Forecasting new product penetration with flexible substitution patterns. *Journal of Econometrics* 89: 109–129. [https://doi.org/10.1016/S0304-4076\(98\)00057-8](https://doi.org/10.1016/S0304-4076(98)00057-8).
- Cameron, A. C., and P. K. Trivedi. 2005. *Microeconometrics: Methods and Applications*. New York: Cambridge University Press.
- Drukker, D. M., and R. Gates. 2006. Generating Halton sequences using Mata. *Stata Journal* 6: 214–228.
- McFadden, D. L., and K. E. Train. 2000. Mixed MNL models for discrete response. *Journal of Applied Econometrics* 15: 447–470. [https://doi.org/10.1002/1099-1255\(200009/10\)15:5<447::AID-JAE570>3.0.CO;2-1](https://doi.org/10.1002/1099-1255(200009/10)15:5<447::AID-JAE570>3.0.CO;2-1).
- Rabe-Hesketh, S., and A. Skrondal. 2022. *Multilevel and Longitudinal Modeling Using Stata*. 4th ed. College Station, TX: Stata Press.
- Train, K. E. 2009. *Discrete Choice Methods with Simulation*. 2nd ed. New York: Cambridge University Press.

Also see

[CM] **cmxtmixlogit postestimation** — Postestimation tools for cmxtmixlogit

[CM] **cmmixlogit** — Mixed logit choice model

[CM] **cmset** — Declare data to be choice model data

[CM] **margins** — Adjusted predictions, predictive margins, and marginal effects

[SVY] **svy estimation** — Estimation commands for survey data

[U] **20 Estimation and postestimation commands**

Stata, Stata Press, and Mata are registered trademarks of StataCorp LLC. Stata and Stata Press are registered trademarks with the World Intellectual Property Organization of the United Nations. StataNow and NetCourseNow are trademarks of StataCorp LLC. Other brand and product names are registered trademarks or trademarks of their respective companies. Copyright © 1985–2023 StataCorp LLC, College Station, TX, USA. All rights reserved.



For suggested citations, see the FAQ on [citing Stata documentation](#).