

xtmlogit — Fixed-effects and random-effects multinomial logit models

Description

`xtmlogit` fits random-effects and conditional fixed-effects multinomial logit models for a categorical dependent variable with unordered outcomes. The actual values taken by the dependent variable are irrelevant.

Quick start

Random-effects model of `y` as a function of `x1`, `x2`, and indicators for levels of categorical variable `a` using `xtset` data

```
xtmlogit y x1 x2 i.a
```

As above, but report relative-risk ratios

```
xtmlogit y x1 x2 i.a, rrr
```

As above, but with all variances and covariances distinctly estimated

```
xtmlogit y x1 x2 i.a, rrr covariance(unstructured)
```

Conditional fixed-effects model

```
xtmlogit y x1 x2 i.a, fe
```

Random-effects model with cluster–robust standard errors for panels nested within `cvar`

```
xtmlogit y x1 x2 i.a, vce(cluster cvar)
```

Menu

Statistics > Longitudinal/panel data > Categorical outcomes > Multinomial logistic regression (FE, RE)

Syntax

Random-effects model

xtmlogit *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, re *RE_options*]

Conditional fixed-effects model

xtmlogit *depvar* [*indepvars*] [*if*] [*in*] [*weight*] , fe [*FE_options*]

RE_options	Description
Model	
noconstant	suppress constant term
re	use random-effects estimator; the default
baseoutcome(#)	value of *depvar* that will be the base outcome
constraints(*constraints*)	apply specified linear constraints
covariance(*vartype*)	variance–covariance structure of the random effects; default is covariance(independent)
SE/Robust	
vce(*vcetype*)	*vcetype* may be oim, robust, cluster *clustvar*, bootstrap, or jackknife
Reporting	
level(#)	set confidence level; default is level(95)
rrr	report relative-risk ratios
lrmodel	perform the likelihood-ratio model test instead of the default Wald test
nocnsreport	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Integration	
intmethod(*intmethod*)	integration method; *intmethod* may be mvaghermite (the default) or ghermite
intpoints(#)	use # quadrature points; default is intpoints(7)
Maximization	
maximize_options	control the maximization process; seldom used
startgrid(*numlist*)	improve starting values of the random-effects variance parameters by performing a grid search
collinear	keep collinear variables
coeflegend	display legend instead of statistics

vartype	Description
`independent`	distinct variances for each random effect and all covariances 0; the default
`shared`	one common random effect
`identity`	equal variances for random effects and all covariances 0
`exchangeable`	equal variances for random effects and one common pairwise covariance
`unstructured`	all variances and covariances to be distinctly estimated

FE_options	Description
Model	
`fe`	use fixed-effects estimator
`baseoutcome(#)`	value of *depvar* that will be the base outcome
`constraints(`*constraints*`)`	apply specified linear constraints
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `robust`, `cluster` *clustvar*, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`rrr`	report relative-risk ratios
`nodots`	suppress display of progress bar
`nocnsreport`	do not display constraints
display_options	control columns and column formats, row spacing, line width, display of omitted variables and base and empty cells, and factor-variable labeling
Permutations	
`rsample(#`[`, rseed(`$\#_s$`)`]`)`	draw sample of permuted outcome sequences at percentage #
`favor(speed`|`space)`	favor speed or space when generating permutations of outcome sequences; default is `favor(speed)`
`force`	force estimation to proceed even if the number of permutations exceeds 50 million
Maximization	
maximize_options	control the maximization process; seldom used
`collinear`	keep collinear variables
`coeflegend`	display legend instead of statistics

A panel variable must be specified; see **[XT] xtset**.

indepvars may contain factor variables and time-series operators; see **[U] 11.4.3 Factor variables** and **[U] 11.4.4 Time-series varlists**.

`bayes`, `by`, `collect`, `statsby`, and `svy` are allowed; see **[U] 11.1.10 Prefix commands**. For more details, see **[BAYES] bayes: xtmlogit**.

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

`fweight`s, `iweight`s, and `pweight`s are allowed; see **[U] 11.1.6 weight**. Weights must be constant within panel.

`startgrid()`, `collinear`, and `coeflegend` do not appear in the dialog box.

See **[U] 20 Estimation and postestimation commands** for more capabilities of estimation commands.

Options for RE model

Model

`noconstant`; see [R] **Estimation options**.

`re` requests the random-effects estimator. This is the default.

`baseoutcome(`*#*`)` specifies the value of *depvar* to be treated as the base outcome. The default is to choose the most frequent outcome.

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

`covariance(`*vartype*`)` specifies the structure of the covariance matrix for the random effects. A multinomial logit model with J outcomes can have up to $J-1$ random effects. *vartype* determines the structure that is assumed for the random effects and is one of the following: `independent`, `shared`, `identity`, `exchangeable`, or `unstructured`.

`covariance(independent)` estimates distinct variances for each of the $J-1$ random effects and all covariances are 0. This is the default.

`covariance(shared)` has one random effect that is common to all $J-1$ outcome equations. Because there is only one random effect, there is no covariance.

`covariance(identity)` estimates one common variance for all $J-1$ random effects and all covariances are 0.

`covariance(exchangeable)` estimates one common variance for all $J-1$ random effects and one common pairwise covariance.

`covariance(unstructured)` allows for all variances and covariances to be distinct. With $p = J-1$ random-effects terms, the unstructured covariance matrix will have $p(p+1)/2$ distinct parameters.

SE/Robust

`vce(`*vcetype*`)` specifies the type of standard error reported, which includes types that are derived from asymptotic theory (`oim`), 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 [XT] ***vce_options***.

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

If `vce(bootstrap)` or `vce(jackknife)` is specified, you must also specify `baseoutcome()`.

Reporting

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

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

`lrmodel`, `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(*intmethod*), intpoints(#); see [R] **Estimation options**.

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**. These options are seldom used.

The following options are available with xtmlogit but are not shown in the dialog box:

startgrid(*numlist*) performs a grid search to improve starting values of the random-effects parameters. By default, xtmlogit performs a grid search on startgrid(0.2 1).

collinear, coeflegend; see [R] **Estimation options**.

Options for FE model

Model

fe requests the fixed-effects estimator.

baseoutcome(#) specifies the value of *depvar* to be treated as the base outcome. The default is to choose the most frequent outcome.

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), 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 [XT] ***vce_options***.

Specifying vce(robust) is equivalent to specifying vce(cluster *panelvar*).

If the rsample() option is specified, the default is vce(robust) rather than vce(oim).

If vce(bootstrap) or vce(jackknife) is specified, you must also specify baseoutcome().

Reporting

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

rrr reports the estimated coefficients transformed to relative-risk ratios, that is, e^b rather than b. Standard errors and confidence intervals are transformed accordingly. This option affects how results are displayed, not how they are estimated. rrr may be specified at estimation or when replaying previously estimated results.

nodots suppresses the display of the dots that show the progress of permuting the observed outcomes.

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**.

Permutations

rsample(#[, rseed($\#_s$)]) specifies that a random subset be drawn from the set of all permutations of the observed sequence of outcomes for each panel. Optionally, a random-number seed, $\#_s$, can be specified to ensure reproducibility.

The size of the random subset is given as a percentage # of K_i, where K_i is the total number of permutations of the outcome sequence in the ith panel. The resulting subset is of size $L_i = \text{ceil}\{(\#/100)K_i\}$. The observed outcome sequence is also included for a total of $L_i + 1$ sequences. If rsample() is not specified, xtmlogit uses all K_i permutations in the conditional likelihood calculation.

Specifying rsample() requires setting a time variable with xtset so that the order of the observed outcome sequence is known.

If rsample() is specified, the default standard error type is vce(robust) rather than vce(oim).

favor(speed | space) instructs xtmlogit to favor either speed or space when generating the permutations of the outcome sequences. favor(speed) is the default. When favoring speed, the permuted sequences are generated once and stored in memory, thus increasing the speed of evaluating the likelihood. This speed increase can be seen when the number of observations per panel is relatively high. When favoring space, the permutations are generated repeatedly with each likelihood evaluation.

force forces estimation to proceed even if the total number of permutations ($\sum_i K_i$) exceeds 50 million. Without specification of force, the fixed-effects estimator issues an error message if the number of permutations exceeds 50 million. Estimation with this many permutations requires a considerable amount of memory and is computationally intensive.

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**. These options are seldom used.

The following options are available with xtmlogit but are not shown in the dialog box:

collinear, coeflegend; see [R] **Estimation options**.

Remarks and examples

stata.com

Remarks are presented under the following headings:

Introduction

xtmlogit fits random-effects and conditional fixed-effects multinomial logit (MNL) models. Whenever we refer to a fixed-effects model, we mean the conditional fixed-effects model.

Both the conditional fixed-effects and the random-effects estimators produce valid estimates in the presence of unobserved heterogeneity at the panel level. The fixed-effects estimator is described in Chamberlain (1980) and Pforr (2014). For a description of the random-effects estimator, see Hartzel, Agresti, and Caffo (2001). For an application of the fixed-effects estimator, see Börsch-Supan (1990); for an application of the random-effects estimator, see Grilli and Rampichini (2007).

The MNL model is a popular method for modeling categorical outcome variables where the categories have no natural ordering. The MNL model is often used in the context of a random utility framework to analyze choices made by individuals. However, the MNL model can also be found used without an underlying utility theory, and the units of analysis do not necessarily have to be individuals or other decision-making entities. In what follows, however, we will refer to individuals for the sake of simplicity, and the set of choices each individual makes as a "panel".

Unlike in cross-sectional applications of the MNL model, in the context of panel and longitudinal data, we observe a sequence of outcomes for each individual in the dataset rather than just a single observation. Each individual sequence can be thought of as a process that depends on individual characteristics.

For example, if we were to analyze restaurant choices, vegetarians would consistently choose restaurants that offer vegetarian dishes, or health-oriented people would consistently avoid fast-food restaurants. In other words, the choices made by individuals are not independent over time because of underlying individual preferences or characteristics, which often remain unobserved in the data. The fixed- and random-effects MNL estimators discussed here offer a way to explicitly account for this unobserved heterogeneity by including an additional error term at the panel level. This panel-level error term is also known as a heterogeneity term and enters the model in addition to the error term that accounts for heterogeneity at the observation (time) level.

The unobserved-heterogeneity model for both the conditional fixed-effects as well as the random-effects estimator can be written in utility-maximization form as

$$U_{ijt} = \mathbf{x}_{it}\boldsymbol{\beta}_j + u_{ij} + \epsilon_{ijt}$$

Assuming we have a panel dataset with repeated observations from individuals, U_{ijt} is the utility of the ith individual toward outcome j at time t, with $i = 1, \dots, N$, $j = 1, \dots, J$, and $t = 1, \dots, T_i$. The observed component of utility is $\mathbf{x}_{it}\boldsymbol{\beta}_j$, where $\mathbf{x}_{it}$ is a row vector of covariates and $\boldsymbol{\beta}_j$ is a column vector of coefficients for outcome j. The unobserved part consists of error components u_{ij} and ϵ_{ijt}, where u_{ij} is the panel-level heterogeneity term and ϵ_{ijt} is an observation-level error term.

Assuming a type-1 extreme value distribution for ϵ_{ijt}, also known as a standard Gumbel distribution, gives rise to the MNL model

$$\Pr(y_{it} = m \,|\, \mathbf{x}_{it}, \boldsymbol{\beta}_j, u_{ij}) = \frac{\exp(\mathbf{x}_{it}\boldsymbol{\beta}_m + u_{im})}{\sum_{j=1}^{J} \exp(\mathbf{x}_{it}\boldsymbol{\beta}_j + u_{ij})}$$

For model identification, the above equation must be normalized with respect to a base category by setting both the elements in $\boldsymbol{\beta}_j$ as well as u_{ij} to zero for one of the categories of the outcome variable. If—without loss of generality—we let the base outcome be outcome 1, the probability that the ith individual chooses outcome m at time t is

$$\Pr(y_{it} = m \,|\, \mathbf{x}_{it}, \boldsymbol{\beta}_j, u_{ij}) = F(y_{it} = m, \mathbf{x}_{it}\boldsymbol{\beta}_j + u_{ij}) = \begin{cases} \dfrac{1}{1 + \sum\limits_{j=2}^{J} \exp(\mathbf{x}_{it}\boldsymbol{\beta}_j + u_{ij})} & \text{if } m = 1 \\ \dfrac{\exp(\mathbf{x}_{it}\boldsymbol{\beta}_m + u_{im})}{1 + \sum\limits_{j=2}^{J} \exp(\mathbf{x}_{it}\boldsymbol{\beta}_j + u_{ij})} & \text{if } m > 1 \end{cases}$$

Here $F(\cdot)$ is defined as the cumulative logistic distribution function.

The fixed-effects and random-effects estimators differ in their assumptions about the unobservables in $\mathbf{u}_i$ and also differ in their methods that the unobservables are accounted for with respect to estimating the coefficients in $\boldsymbol{\beta}_j$.

The random-effects estimator

The random-effects estimator requires an assumption about the distribution of u_{ij}, and the elements in $\mathbf{u}_i$ are assumed to be uncorrelated with the covariates in $\mathbf{x}_{it}$. The covariates $\mathbf{x}_{it}$ may contain constant terms as well as time-invariant predictor variables. Assuming a normal distribution for u_{ij}, the panel-level likelihood is

$$l_i = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \left\{ \prod_{t=1}^{T_i} F(y_{it} = m, \mathbf{x}_{it}\boldsymbol{\beta}_j + u_{ij}) \right\} \phi(\mathbf{u}_i, \boldsymbol{\Sigma}_u)\, d\mathbf{u}_i \tag{1}$$

where $\phi(\mathbf{u}_i, \boldsymbol{\Sigma}_u)$ is the probability density function of the normal distribution $\mathbf{u}_i \sim N(\mathbf{0}, \boldsymbol{\Sigma}_u)$. This integral of dimension $J-1$ has no closed-form solution and must be approximated numerically. By default, `xtmlogit` uses adaptive Gauss–Hermite quadrature to approximate this integral.

`xtmlogit` allows for imposing a variety of structures on $\boldsymbol{\Sigma}_u$. By default, `xtmlogit` estimates separate, independent variance components for each of the $J-1$ outcome equations. The option `covariance(shared)` estimates a single shared variance component for all $J-1$ outcome equations. The most general case is specified by the option `covariance(unstructured)`, which freely estimates all variances and covariances among the random effects instead of treating them as independent. Not imposing any structure on $\boldsymbol{\Sigma}_u$ can potentially yield more accurate results. However, this is also more computationally intensive, resulting in longer computation times.

The conditional fixed-effects estimator

The advantages of the conditional fixed-effects estimator are that elements in $\mathbf{u}_i$ can be correlated with the covariates in $\mathbf{x}_{it}$ and no distributional assumptions need to be imposed on u_{ij}. Unlike in linear fixed-effects models, the heterogeneity term u_{ij} of the logit model cannot be eliminated by taking deviations from the group mean. Moreover, it is also not feasible to account for the heterogeneity in u_{ij} by distinctly estimating an intercept for each panel because this leads to the incidental parameters problem, which renders the estimator of $\boldsymbol{\beta}_j$ inconsistent for a fixed T_i; see Andersen (1970) and Lancaster (2000). Instead, Chamberlain (1980) suggested the use of a sufficient statistic for the unobserved heterogeneity u_{ij}.

Let $\mathbf{Y}_i = (Y_{i1}, \ldots, Y_{iT_i})$ be the sequence of outcomes of the ith panel, and let $\mathbf{Y}_{it} = (Y_{i1t}, \ldots, Y_{iJt})$ be a vector with elements $Y_{ijt} = 1(i \text{ chooses } j \text{at } t)$ that indicate the chosen outcome of the ith panel at time t. The distribution of times that panel i chose each of the J alternatives over time points T_i is then the sufficient statistic $\Theta_i = \sum_{t=1}^{T_i} \mathbf{Y}_{it} = \mathbf{c}_i = (c_{i1}, \ldots, c_{iJ})$. In other words, the elements in $\mathbf{c}_i$ are sums of occurrences of each of the outcomes over time for the ith panel.

Conditioning on the sufficient statistic Θ_i, the probability of panel i having a sequence $\mathbf{Y}_i = \mathbf{s}_i$ that is consistent with $\mathbf{c}_i$ is

$$\begin{aligned}\Pr(\mathbf{Y}_i = \mathbf{s}_i \,|\, \Theta_i, \mathbf{u}_i, \mathbf{x}_i, \boldsymbol{\beta}) &= \Pr\{Y_{i1}, \ldots, Y_{iT_i} \,|\, \Psi(\mathbf{c}_i), \mathbf{u}_i, \mathbf{x}_i, \boldsymbol{\beta}\} \\ &= \frac{\exp\left(\sum\limits_{t=1}^{T_i}\sum\limits_{j=2}^{J} Y_{ijt}\mathbf{x}_{it}\boldsymbol{\beta}_j\right)}{\sum\limits_{\widetilde{Y}_{ijt}\in\Psi(\mathbf{c}_i)} \exp\left(\sum\limits_{t=1}^{T_i}\sum\limits_{j=2}^{J} \widetilde{Y}_{ijt}\mathbf{x}_{it}\boldsymbol{\beta}_j\right)}\end{aligned}$$

where $\Psi(\mathbf{c}_i)$ is the set of all permutations of individual i's observed sequence of outcomes that satisfy the condition $\sum_{t=1}^{T_i} \widetilde{\mathbf{Y}}_{it} = \mathbf{c}_i$. That is,

$$\Psi(\mathbf{c}_i) = \left\{ \widetilde{\mathbf{Y}}_i = (\widetilde{Y}_{i1}, \ldots, \widetilde{Y}_{iT_i}) \;\middle|\; \sum_{t=1}^{T_i} \widetilde{\mathbf{Y}}_{it} = \mathbf{c}_i \right\}$$

and $\widetilde{\mathbf{Y}}_{it} = (\widetilde{Y}_{i1t}, \ldots, \widetilde{Y}_{iJt})$ is a vector of indicators with respect to the permutations of the observed outcome sequence $\mathbf{Y}_i$. The log likelihood of panel i is then the natural logarithm of the above probability

$$\log l_i = \sum_{t=1}^{T_i}\sum_{j=2}^{J} Y_{ijt}\mathbf{x}_{it}\boldsymbol{\beta}_j - \log \sum_{\widetilde{Y}_{ijt}\in\Psi(\mathbf{c}_i)} \exp\left(\sum_{t=1}^{T_i}\sum_{j=2}^{J} \widetilde{Y}_{ijt}\mathbf{x}_{it}\boldsymbol{\beta}_j\right)$$

and the overall log likelihood is $\sum\limits_{i=1}^{N} \log l_i$.

To illustrate the concept of permutations in this context, let us suppose we had a panel dataset with three observations per individual and an outcome variable with four categories, $j = 1, 2, 3, 4$. Let us further assume that for some individual in the dataset we observe the sequence $\mathbf{Y}_i = (3, 2, 3)$. This sequence has a total of three permutations, so the set of all permutations (which includes the original sequence) for this individual consists of $(2, 3, 3)$, $(3, 2, 3)$, and $(3, 3, 2)$. Notice that in all three permutations, outcome 3 occurs twice, and outcome 2 occurs once, just as in the original sequence.

Curse of dimensionality

Both the random-effects and fixed-effects estimators suffer from the curse of dimensionality. For the random-effects estimator, the curse is rooted in J, the number of outcomes, because the integral in (1) is a $J - 1$ dimensional integral unless one uses a common heterogeneity component for all outcomes. This means that the computation time can be high for more than just three or four outcomes. For example, if we had a dataset with six outcomes, we would have to approximate a five-dimensional integral. If we were to use the default seven quadrature integration points, which are integration points per dimension, we would end up with a total of $7^5 = 16807$ integration points, resulting in substantial computation time. If computation time becomes infeasible, one might consider using a single, shared variance component, if appropriate.

For the fixed-effects estimator, the curse of dimensionality is rooted mainly in T_i, the number of repeated observations and potentially in J. The problem is that the number of permutations in $\Psi(\mathbf{c}_i)$ grows exponentially with T_i and can become infeasibly large. The number of permutations of panel i's observed vector of outcomes is

$$K_i = \frac{T_i!}{c_{i1}! \cdots c_{ij}! \cdots c_{iJ}!}$$

For instance, suppose we observed an individual with 15 repeated observations in a dataset with 6 outcomes, $j = 1, 2, \ldots, 6$, with the sequence of outcomes $\mathbf{Y}_i = (3, 3, 3, 2, 4, 1, 1, 5, 4, 6, 6, 1, 1, 2, 4)$. Here $\sum_{t=1}^{T_i} Y_{i1t} = 4$, which is to say that outcome $Y_{it} = 1$ is observed 4 times, $\sum_{t=1}^{T_i} Y_{i2t} = 2$, and so on. The size of the set of permutations of this outcome vector is

$$K_i = \frac{15!}{4!\,2!\,3!\,3!\,1!\,2!} = 378{,}378{,}000$$

Notice that this number in the hundreds of millions is the size of the permutation set of just a single panel in the dataset, and clearly this number can quickly become infeasibly large.

A potential solution that can alleviate this problem to some degree is to use a random subset of permutations (D'Haultfœuille and Iaria 2016). The `rsample()` option can be used to specify the size of the random subset as a percentage of the full set of permutations. Realistically, however, the fixed-effects estimator is really feasible only with shorter panels where the number of repeated observations does not exceed $T_i = 9$ or $T_i = 10$, depending on J, the size of the dataset, and possibly other features of the data.

Examples

▷ Example 1: MNL model with random effects

We have a (fictitious) unbalanced panel dataset of 800 women aged 18 to 40 at the time of the first interview. We wish to estimate the effect of having children under the age of 18 in the household on the women's employment status. Specifically, we wish to find out whether women become more likely not to participate in the labor force in response to having children in the household. And if so, how much more unlikely is it?

The survey was repeated every two years, and the women were asked about their main employment status during the year preceding each of the interviews. The employment status response categories were employed (full time, part time, or self-employed), unemployed (job seeking), and out of the labor force. Here is an excerpt of the dataset, showing the employment history for three individuals:

```
. use https://www.stata-press.com/data/r17/estatus
(Fictional employment status data)
. list id year estatus hhchild age in 22/41, sepby(id) noobs
```

id	year	estatus	hhchild	age
5	2002	Employed	Yes	38
5	2004	Employed	No	40
5	2006	Employed	No	42
5	2008	Employed	No	44
5	2010	Out of labor force	No	46
5	2012	Out of labor force	No	48
5	2014	Unemployed	No	50
6	2002	Unemployed	Yes	31
6	2004	Employed	Yes	33
6	2006	Out of labor force	Yes	35
6	2008	Unemployed	Yes	37
6	2010	Out of labor force	Yes	39
6	2012	Unemployed	No	41
7	2002	Out of labor force	Yes	33
7	2004	Employed	Yes	35
7	2006	Employed	Yes	37
7	2008	Out of labor force	Yes	39
7	2010	Employed	No	41
7	2012	Employed	No	43
7	2014	Employed	No	45

The first person shown in the above excerpt (`id==5`) was observed between years 2002 and 2014. The variable `estatus` records the employment history over these years. In this case, the person has been employed between 2002 and 2008, was out of the labor force between 2010 and 2012, and was unemployed prior to the interview in 2014.

The variable `hhchild` records whether at least one child under the age of 18 was living with the surveyee in the same household at the time of the interview. Looking at the data of the first person in the above excerpt, we see that there was one or more children in the household in 2002, but no children in the household between 2004 and 2014. The variable `age` records the age of the women at each interview. In this case, the woman was observed between 38 and 50 years of age.

To inspect the distribution of employment status over the entire sample, we can use the `tabulate` command:

```
. tabulate estatus
```

Employment status	Freq.	Percent	Cum.
Out of labor force	1,682	35.33	35.33
Unemployed	703	14.77	50.09
Employed	2,376	49.91	100.00
Total	4,761	100.00	

We can see that in 35% of all observations, the interviewed women reported to be out of the labor force, 15% of the time the women were unemployed, and 50% of the time the women were employed.

As with other panel-data estimators, we first need to declare our dataset to be panel data by using the `xtset` command. Here we do not plan to use any lagged covariates, so it is sufficient to `xtset` our dataset with just the person identifier `id` and without a variable for time:

```
. xtset id

Panel variable: id (unbalanced)
```

We can now go ahead and fit our model using `xtmlogit`. We will also include a number of control variables: `age`, a person's annual household income at the time of interview (`hhincome`), whether a significant other was also living in the household at the time of interview (`hhsigno`), and whether the surveyee was the sole or primary breadwinner in her household at the time of interview (`bwinner`).

We use the variable `estatus` as our dependent variable, and `hhchild` is our independent variable of interest. Because `hhchild`, `hhsigno`, and `bwinner` are binary variables, we specify them as factor variables.

```
. xtmlogit estatus i.hhchild age hhincome i.hhsigno i.bwinner
Fitting comparison model ...
Refining starting values:
Grid node 0:   log likelihood = -4483.1721
Grid node 1:   log likelihood = -4516.6753
Fitting full model:
Iteration 0:   log likelihood = -4483.1721
Iteration 1:   log likelihood = -4474.3849
Iteration 2:   log likelihood = -4468.9353
Iteration 3:   log likelihood = -4468.8415
Iteration 4:   log likelihood = -4468.8413
Random-effects multinomial logistic regression      Number of obs     =  4,761
Group variable: id                                  Number of groups =    800
Random effects u_i ~ Gaussian                       Obs per group:
                                                                 min =      5
                                                                 avg =    6.0
                                                                 max =      7
Integration method: mvaghermite                     Integration pts. =      7
                                                    Wald chi2(10)     = 239.26
Log likelihood = -4468.8413                         Prob > chi2       = 0.0000

------------------------------------------------------------------------------
     estatus | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
Out_of_lab~e |
     hhchild |
        Yes  |   .4628125   .0962758     4.81   0.000     .2741154    .6515096
         age |   -.004825   .0066428    -0.73   0.468    -.0178446    .0081946
    hhincome |  -.0046922    .001839    -2.55   0.011    -.0082965   -.0010879
             |
     hhsigno |
        Yes  |   .4967056   .0946442     5.25   0.000     .3112063    .6822049
             |
     bwinner |
        Yes  |  -.4740919   .0727992    -6.51   0.000    -.6167756   -.3314082
       _cons |  -.4787579   .2845139    -1.68   0.092    -1.036395    .0788792
-------------+----------------------------------------------------------------
Unemployed   |
     hhchild |
        Yes  |  -.0401989    .119596    -0.34   0.737    -.2746027    .1942049
         age |   .0042644   .0081818     0.52   0.602    -.0117716    .0203004
    hhincome |  -.0308468   .0026529   -11.63   0.000    -.0360463   -.0256473
             |
     hhsigno |
        Yes  |      .0968   .1192659     0.81   0.417    -.1369568    .3305568
             |
     bwinner |
        Yes  |  -.2252587   .0951984    -2.37   0.018    -.4118441   -.0386733
       _cons |  -.0953821   .3508736    -0.27   0.786    -.7830817    .5923175
-------------+----------------------------------------------------------------
Employed     |  (base outcome)
-------------+----------------------------------------------------------------
      var(u1)|   .8587807   .1090216                      .6696113    1.101392
      var(u2)|   .7370366   .1388917                      .5094287    1.066338
------------------------------------------------------------------------------
LR test vs. multinomial logit: chi2(2) = 225.31           Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

Looking at the table header, we can find some useful information about the model we just fit. For example, we can see that the estimation sample consists of 4,761 observations from 800 groups (800 individuals in this case), with between 5 and 7 observations per group. The model test right above the table on the right is a joint test of all model coefficients except the constants.

Looking at the output table itself, we see the results for all $J-1$ equations. Because we have three outcome categories, we see the coefficient estimates for two of the outcomes, while employment is our base outcome. Here using employment as the base makes sense given our research question, and we would have chosen this as a base if we had to specify it explicitly. In this case, however, employment was chosen automatically because it is the most frequent category in our dataset, which is what `xtmlogit` defaults to. If we had wanted to specify a different category as the base, we would have used the `baseoutcome()` option.

Below the model coefficient estimates, we find the estimated variances of the random effects. In this case, we have two estimates that correspond to the nonbase equations. By default, `xtmlogit` assumes that the random effects are uncorrelated across the equations. We will see in the next example how to use the `covariance()` option to specify a different covariance structure. Here we can see that there is some considerable variance of the panel-level unobservables. The lower bound of the 95% confidence interval is not close to zero relative to their estimated standard errors. This observation is confirmed by the likelihood-ratio test shown beneath the table, which is a test of our model against the MNL model without random effects.

Let us get back to our initial research question: what is the effect of having children under the age of 18 in the household on employment status? The interpretation of the coefficients is the same as in a conventional cross-sectional MNL model, except that, in the random-effects case, they are to be interpreted as conditional on the random effects, while they naturally have a population-average interpretation in the cross-sectional case. Either way, the coefficients are difficult to interpret. They can be thought of as the natural logarithm of a double ratio: the logarithm of the relative risk, relative to the base category. Realistically, only the sign of these coefficients can be interpreted usefully. Looking at the results, we can see that the coefficient of `hhchild` in the first equation (out of labor force) is around 0.46. Thus, we can say that women with children under 18 in the household are more likely not to participate in the labor force than women with no young children in the household, relative to being employed full time.

A more informative way to interpret the results would be to use relative-risk ratios (RRRs) instead of log relative-risk ratios by exponentiating the coefficients. That is, instead of β_j, we use $\exp(\beta_j)$ to interpret the results. With `xtmlogit`, we can use the `rrr` option for that purpose. This option can be used at the time of estimation or when replaying results. Here we use it as a replay option:

```
. xtmlogit, rrr
Random-effects multinomial logistic regression      Number of obs     =  4,761
Group variable: id                                  Number of groups  =    800
Random effects u_i ~ Gaussian                       Obs per group:
                                                                  min =      5
                                                                  avg =    6.0
                                                                  max =      7
Integration method: mvaghermite                     Integration pts.  =      7
                                                    Wald chi2(10)     = 239.26
Log likelihood = -4468.8413                         Prob > chi2       = 0.0000

------------------------------------------------------------------------------
     estatus |        RRR   Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
Out_of_lab~e |
     hhchild |
        Yes  |   1.588535   .1529375     4.81   0.000     1.315367    1.918435
         age |   .9951866   .0066108    -0.73   0.468     .9823137    1.008228
    hhincome |   .9953188   .0018303    -2.55   0.011     .9917379    .9989127
             |
     hhsigno |
        Yes  |   1.643299   .1555288     5.25   0.000     1.365071    1.978235
             |
     bwinner |
        Yes  |   .6224501   .0453138    -6.51   0.000     .5396818    .7179121
       _cons |   .6195525   .1762713    -1.68   0.092     .3547312    1.082074
-------------+----------------------------------------------------------------
Unemployed   |
     hhchild |
        Yes  |   .9605983   .1148837    -0.34   0.737     .7598739    1.214345
         age |   1.004274   .0082168     0.52   0.602     .9882974    1.020508
    hhincome |   .9696241   .0025723   -11.63   0.000     .9645956    .9746788
             |
     hhsigno |
        Yes  |    1.10164   .1313881     0.81   0.417     .8720079    1.391743
             |
     bwinner |
        Yes  |   .7983097   .0759978    -2.37   0.018     .6624275    .9620649
       _cons |   .9090255   .3189531    -0.27   0.786     .4569955    1.808174
-------------+----------------------------------------------------------------
Employed     |  (base outcome)
-------------+----------------------------------------------------------------
     var(u1) |   .8587807   .1090216                      .6696113    1.101392
     var(u2) |   .7370366   .1388917                      .5094287    1.066338
------------------------------------------------------------------------------
Note: Estimates are transformed only in the first 3 equations to
      relative-risk ratios.
Note: _cons estimates baseline relative risk (conditional on zero random
      effects).
LR test vs. multinomial logit: chi2(2) = 225.31            Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

Looking at the RRRs of `hhchild` in the out-of-labor-force equation, which is around 1.6, we can say that the relative risk of being out of the labor force for women having at least one child under the age of 18 in the household versus having no children under 18 in the household is 1.6 times as large as the relative risk in the case of employment. While this provides a little bit more information, it still does not provide a very intuitive way to interpret our results. It would be easier if we could just see the actual risks for each of the outcomes with respect to the `hhchild` variable and then also the risk differences rather than risk ratios. To that end, we can use `margins`:

```
. margins hhchild

Predictive margins                                     Number of obs = 4,761
Model VCE: OIM

1._predict: Pr(estatus==Out_of_labor_force), predict(pr outcome(1))
2._predict: Pr(estatus==Unemployed), predict(pr outcome(2))
3._predict: Pr(estatus==Employed), predict(pr outcome(3))

------------------------------------------------------------------------------
             |            Delta-method
             |     Margin   std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
   _predict# |
     hhchild |
        1#No |   .3021986   .0131047    23.06   0.000     .2765138    .3278834
       1#Yes |   .3912783   .0119865    32.64   0.000     .3677852    .4147714
        2#No |   .1630791   .0101239    16.11   0.000     .1432367    .1829216
       2#Yes |    .139782   .0079417    17.60   0.000     .1242167    .1553474
        3#No |   .5347223   .0136504    39.17   0.000      .507968    .5614766
       3#Yes |   .4689397   .0116018    40.42   0.000     .4462006    .4916787
------------------------------------------------------------------------------
```

By default, `margins` uses predicted probabilities that account for the random effects. The probabilities are obtained by integrating out the random effects such that their averages can be used to make population-average inferences. Starting with our third outcome, employment, we can see that the averaged probability of being employed full time is around 0.47 in the presence of children under the age of 18 in the household, whereas this probability is around 0.53 in the absence of young children. Thus, women have a higher chance of being employed full time if they have no young children living with them in the same household.

We can further quantify the difference in chance by calculating the risk difference, which here is around 0.07. Using a percentage scale rather than probability scale, we can say that the chance of being employed is higher by about 7 percentage points if no young child is in the household. Looking at the other outcome of interest, we can see that the chance of being out of the labor force is about 39% in the presence of young children in the household and around 30% otherwise, resulting in a risk difference of around 9 percentage points.

We could also compute these risk differences directly by using the contrast operator `r.`:

```
. margins r.hhchild

Contrasts of predictive margins                          Number of obs = 4,761
Model VCE: OIM

1._predict: Pr(estatus==Out_of_labor_force), predict(pr outcome(1))
2._predict: Pr(estatus==Unemployed), predict(pr outcome(2))
3._predict: Pr(estatus==Employed), predict(pr outcome(3))
```

	df	chi2	P>chi2
hhchild@_predict			
(Yes vs No) 1	1	26.36	0.0000
(Yes vs No) 2	1	3.28	0.0700
(Yes vs No) 3	1	13.33	0.0003
Joint	2	26.40	0.0000

	Contrast	Delta-method std. err.	[95% conf.	interval]
hhchild@_predict				
(Yes vs No) 1	.0890797	.0173496	.0550752	.1230843
(Yes vs No) 2	-.0232971	.0128562	-.0484948	.0019005
(Yes vs No) 3	-.0657826	.0180195	-.1011001	-.0304651

We can see that the results match the differences from the previous `margins` call. The predicted probabilities underlying the `margins` analysis are also the default predictions of `predict` after `xtmlogit, re`.

◁

▷ Example 2: Covariance structure of the random effects

As mentioned in the previous example, `xtmlogit` by default uses an independent covariance structure for the random effects, which is to say that the random effects for each of the $J-1$ equations are assumed to be uncorrelated. A more general case here would be to not impose any structure on the random effects and freely estimate the covariances among the random effects rather than assuming that the covariances are zero. To fit our model with an unstructured covariance matrix, we use the option `covariance(unstructured)`:

```
. xtmlogit estatus i.hhchild age hhincome i.hhsigno i.bwinner,
> covariance(unstructured)
```

(output omitted)

```
Random-effects multinomial logistic regression      Number of obs    =  4,761
Group variable: id                                  Number of groups =    800

Random effects u_i ~ Gaussian                       Obs per group:
                                                                 min =      5
                                                                 avg =    6.0
                                                                 max =      7

Integration method: mvaghermite                     Integration pts. =      7

                                                    Wald chi2(10)    = 242.93
Log likelihood = -4438.2887                         Prob > chi2      = 0.0000

------------------------------------------------------------------------------
     estatus | Coefficient  Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
Out_of_lab~e |
     hhchild |
        Yes  |   .4924799   .1002988     4.91   0.000      .295898    .6890619
         age |   -.004219   .0070064    -0.60   0.547    -.0179513    .0095133
    hhincome |   -.006046    .001992    -3.04   0.002    -.0099503   -.0021417
             |
     hhsigno |
        Yes  |   .5036976   .0966982     5.21   0.000     .3141726    .6932225
             |
     bwinner |
        Yes  |   -.489057   .0745454    -6.56   0.000    -.6351632   -.3429507
       _cons |  -.3930378    .298386    -1.32   0.188    -.9778636     .191788
-------------+----------------------------------------------------------------
Unemployed   |
     hhchild |
        Yes  |   .0399687   .1238417     0.32   0.747    -.2027565    .2826939
         age |   .0045538   .0085081     0.54   0.592    -.0121219    .0212294
    hhincome |  -.0315377   .0027426   -11.50   0.000    -.0369131   -.0261624
             |
     hhsigno |
        Yes  |   .1495817   .1214242     1.23   0.218    -.0884053    .3875687
             |
     bwinner |
        Yes  |  -.2552257   .0968165    -2.64   0.008    -.4449826   -.0654689
       _cons |  -.0417024   .3633406    -0.11   0.909    -.7538368     .670432
-------------+----------------------------------------------------------------
Employed     |  (base outcome)
-------------+----------------------------------------------------------------
     var(u1) |   1.132081   .1331468                       .899012    1.425572
     var(u2) |   1.102612   .1698422                      .8152803     1.49121
-------------+----------------------------------------------------------------
   cov(u1,u2)|   .7871916   .1222148     6.44   0.000      .547655    1.026728
------------------------------------------------------------------------------
LR test vs. multinomial logit: chi2(3) = 286.41           Prob > chi2 = 0.0000
Note: LR test is conservative and provided only for reference.
```

At the bottom of the table, we can see the additional estimate for the covariance among the random effects. When we look at the estimate relative to its standard error, or at the corresponding test result, it looks as though the random effects are correlated considerably. To get a better idea of how strongly the random effects are correlated, we might want to look at standard deviations and correlations, rather than variances and covariances. We can do that by using the `estat sd` postestimation command:

```
. estat sd
```

estatus	Coefficient	Std. err.	z	P>|z|	[95% conf.	interval]
sd(u1)	1.063993	.0625694			.9481624	1.193973
sd(u2)	1.050053	.0808731			.9029287	1.221151
corr(u1,u2)	.7045801	.0632646	11.14	0.000	.5581225	.8084624

The results of `estat sd` show that the correlation between the random effects, `u1` and `u2`, is around 0.7, which appears rather substantial. If we had more than one estimated covariance and wanted to test the inclusion of covariance estimates as a whole, we could perform a joint test on the covariances against zero using the `test` command. Testing covariances against zero is straightforward because they are not bounded, unlike the variances. Because here we have only a single covariance estimate, we can simply take the test result reported by `xtmlogit`. The results show that we can reject the hypothesis of the covariance being zero.

Alternatively, we could perform a likelihood-ratio test here because the model with independent covariance structure is a special case of the model with no structure imposed. We will fit the model with uncorrelated random effects again, store the results, and use the `lrtest` command to perform the likelihood-ratio test:

```
. estimates store unstr
. xtmlogit estatus i.hhchild age hhincome i.hhsigno i.bwinner, baseoutcome(3)
  (output omitted )
. estimates store indep
. lrtest unstr indep
Likelihood-ratio test
Assumption: indep nested within unstr
 LR chi2(1) =  61.11
Prob > chi2 = 0.0000
```

The conclusion here is the same as before: the model with no structure imposed on the random effects covariance matrix appears to be preferable. However, if we compare the results with those from our previous model, we can see that the model with unstructured covariance matrix would not necessarily lead to substantially different conclusions, judging by the differences in relative-risk ratios between the two models. This becomes even more apparent if we were to look at the differences in the averaged marginal probabilities. For example, the difference between having and not having a child in the household with respect to not participating in the labor force was 0.089 on the probability scale in the previous example with independent covariance structure. If we were to compute this risk difference again for the unstructured model, we would find a difference of 0.092 with a similar standard error.

As an aside, notice that when we refit the model with uncorrelated random effects, we specified the option `baseoutcome(3)`. We would not have to do this because we already knew that `xtmlogit` would choose the third outcome as base, but we did so anyway to point out that it is good practice to be explicit about this in this context. It is important that the models that are compared with a likelihood-ratio test use the same base outcome. This is because, unlike in a conventional cross-sectional MNL model, the likelihood solution differs with different base outcomes because the modeling of random effects depends on what category is selected as the reference category.

◁

▷ Example 3: MNL model with conditional fixed effects

We will now use the conditional fixed-effects estimator instead of the random-effects estimator to fit our model. To do so, all we need to do is to specify the `fe` option of `xtmlogit`. However, because we have seen that the results are easier to interpret with relative-risk ratios, we will specify the `rrr` option right away:

```
. xtmlogit estatus i.hhchild age hhincome i.hhsigno i.bwinner, fe rrr
note: 80 groups (451 obs) omitted because of no variation in the outcome
      variable over time.

Computing initial values ...

Setting up 26,168 permutations:
....10%....20%....30%....40%....50%....60%....70%....80%....90%....100%

Fitting full model:

Iteration 0:   log likelihood = -2136.5919
Iteration 1:   log likelihood = -2136.2728
Iteration 2:   log likelihood = -2136.2728

Fixed-effects multinomial logistic regression        Number of obs     =  4,310
Group variable: id                                   Number of groups =    720

                                                     Obs per group:
                                                                  min =      5
                                                                  avg =    6.0
                                                                  max =      7

                                                     LR chi2(10)       = 103.29
Log likelihood = -2136.2728                          Prob > chi2       = 0.0000

------------------------------------------------------------------------------
     estatus |        RRR   Std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
Out_of_lab~e |
     hhchild |
        Yes  |   1.800717   .2266555     4.67   0.000     1.407036    2.304549
         age |   .9996159   .0147684    -0.03   0.979     .9710854    1.028985
    hhincome |   .9878698   .0087391    -1.38   0.168     .9708891    1.005148
             |
     hhsigno |
        Yes  |   1.663632    .166548     5.08   0.000     1.367233    2.024287
             |
     bwinner |
        Yes  |   .6277743   .0491447    -5.95   0.000     .5384781    .7318786
-------------+----------------------------------------------------------------
Unemployed   |
     hhchild |
        Yes  |   1.177757   .1930267     1.00   0.318     .8541801    1.623911
         age |   1.006356   .0195273     0.33   0.744     .9688014    1.045366
    hhincome |   .9706959   .0116513    -2.48   0.013     .9481262    .9938029
             |
     hhsigno |
        Yes  |   1.124478   .1463356     0.90   0.367     .8713222    1.451187
             |
     bwinner |
        Yes  |   .7795833   .0802992    -2.42   0.016      .637069    .9539784
-------------+----------------------------------------------------------------
Employed     |  (base outcome)
------------------------------------------------------------------------------
```

Starting with the table header, we can see that our estimation sample consists of 4,310 observations from 720 women. We saw earlier that we had 800 women in our dataset, so why do we now have only 720? The answer to this question is given in the note near the top of the output, which lets us know that 451 observations from 80 groups were dropped for the analysis because in these cases, there is no variation in the outcome variable over time. Technically, these observations could have been kept in the estimation sample, but with no variation in the outcome, these observations would not contribute anything to the likelihood, so they can as well be excluded. Looking at the relative-risk ratios, we see the results are fairly similar to our random-effects estimates. We observe an RRR for our variable of interest `hhchild` of around 1.8 for the out-of-labor-force category. The interpretation of the RRRs here is the same as with the random-effects model from the earlier examples.

An unfortunate side effect of the fixed-effects estimator is that we cannot make predictions that account for the panel-level unobservables. That is because we do not estimate the unobservables explicitly. Therefore, unfortunately, we also cannot perform useful marginal analyses using the `margins` command.

◁

▷ Example 4: Fixed effects estimation with random permutation sampling

As noted earlier, if the number of repeated observations, T_i, becomes larger than, say, $T_i = 10$, the set of permutations can become very large, resulting in computations that may become infeasibly intensive. In that case, a potential solution could be to take a random sample of the set of permutations. This can be done by using the `rsample()` option, which allows one to specify the size of the sample as a percentage of the full set of permutations. Here we will fit the same model as in the previous example, except that we take a 10% random sample of the permutations.

Before we do so, however, note that in this case we have to `xtset` the data with a time variable. The reason for this is that we have to determine the observed sequence of outcomes that has to be included in the set of permutations that we use in the denominator of the formula. This is not necessary without sampling, because the full set of permutations always includes the observed sequence. Without determining the observed sequence, estimation results would randomly depend on the sort order of the data. To specify drawing a 10% random sample and to also set a random-number seed for reproducibility, we just add the option `rsample(10, rseed(123))`.

```
. xtset id year

Panel variable: id (unbalanced)
 Time variable: year, 2002 to 2014, but with gaps
         Delta: 1 unit

. xtmlogit estatus i.hhchild age hhincome i.hhsigno i.bwinner, fe rrr
> rsample(10, rseed(123))
note: option vce() set to vce(robust) because of permutation sampling.
note: 80 groups (451 obs) omitted because of no variation in the outcome
      variable over time.

Computing initial values ...

Setting up 3,495 permutations:
....10%....20%....30%....40%....50%....60%....70%....80%....90%....100%

Fitting full model:

Iteration 0:   log pseudolikelihood = -908.26163
Iteration 1:   log pseudolikelihood =  -906.4585
Iteration 2:   log pseudolikelihood = -906.45801
Iteration 3:   log pseudolikelihood = -906.45801

Fixed-effects multinomial logistic regression       Number of obs    =  4,310
Group variable: id                                  Number of groups =    720

                                                    Obs per group:
                                                                 min =      5
                                                                 avg =    6.0
                                                                 max =      7

                                                    Wald chi2(10)    =  72.91
Log pseudolikelihood = -906.45801                   Prob > chi2      = 0.0000

                                    (Std. err. adjusted for 720 clusters in id)
------------------------------------------------------------------------------
             |               Robust
     estatus |        RRR   std. err.      z    P>|z|     [95% conf. interval]
-------------+----------------------------------------------------------------
Out_of_lab~e |
     hhchild |
        Yes  |   1.790876   .2663706     3.92   0.000     1.338011    2.397017
         age |    .994506   .0167663    -0.33   0.744     .9621816    1.027916
    hhincome |   .9858517   .0099036    -1.42   0.156     .9666309    1.005455
             |
     hhsigno |
        Yes  |   1.559166   .1891864     3.66   0.000     1.229162    1.977769
             |
     bwinner |
        Yes  |   .6304536   .0616622    -4.72   0.000     .5204757    .7636702
-------------+----------------------------------------------------------------
Unemployed   |
     hhchild |
        Yes  |   1.186982   .2173595     0.94   0.349     .8290349    1.699479
         age |   .9953453   .0215995    -0.21   0.830     .9538986    1.038593
    hhincome |   .9661192   .0127244    -2.62   0.009     .9414989    .9913833
             |
     hhsigno |
        Yes  |   .9267669   .1294269    -0.54   0.586     .7048498    1.218553
             |
     bwinner |
        Yes  |   .7490293    .088281    -2.45   0.014     .5945326    .9436738
-------------+----------------------------------------------------------------
Employed     |  (base outcome)
------------------------------------------------------------------------------
```

Looking at the output, we can see that the results are very close to the results from the previous example, with standard errors being slightly larger, as we would expect. Notice also that, by default, `xtmlogit` computes cluster–robust standard errors in this case because the likelihood function is not the true likelihood because a term that is the sum over all permutations is replaced by a sum over a sample of the permutations.

◁

▷ Example 5: Choosing between fixed- and random-effects models

It can be challenging to decide a priori whether to use the fixed- or the random-effects estimator. If the assumptions of the random-effects estimator hold, then both the random and fixed-effects estimators are consistent. However, in that case, the random-effects estimator is more efficient and thus preferable. On the other hand, if the assumptions of the random-effects estimator do not hold, it becomes inconsistent, and we should use the fixed-effects estimator. In that sense, one could get the idea that we should err on the side of caution and always use the fixed-effects estimator. However, the random-effects estimator has a couple of practical advantages beyond efficiency.

For example, if we were interested in estimating the effect of a variable that is constant over time (for all observations in the dataset), then we could include that variable in the random-effects model, but not in the fixed-effects model. With the fixed-effects model, variables that are constant over time are absorbed into the fixed effects. Also, with the random-effects estimator, we can predict probabilities that account for panel-level unobservables and that lend themselves to a population-average interpretation when we use `margins`—something we cannot do with the fixed-effects estimator because the unobservables are not estimated.

A possible solution to this dilemma is to use a Hausman test. In our context here, our null hypothesis (H_0) is that the panel-level unobservables are uncorrelated with the covariates in the model, while the alternative hypothesis (H_a) is that the unobservables are correlated with the covariates. The fixed-effects estimator is consistent under both H_0 and H_a, while the random-effects estimator is inconsistent under H_a but efficient under H_0. To apply the Hausman test, we first fit both the fixed- and random-effects models, store their results, and then use the `hausman` command:

```
. xtmlogit estatus i.hhchild age hhincome i.hhsigno i.bwinner, fe
  (output omitted)
. estimates store FE
. xtmlogit estatus i.hhchild age hhincome i.hhsigno i.bwinner
  (output omitted)
. estimates store RE
. hausman FE RE, alleqs

                    ---- Coefficients ----
             |      (b)          (B)            (b-B)     sqrt(diag(V_b-V_B))
             |       FE           RE         Difference       Std. err.
-------------+----------------------------------------------------------------
Out_of_lab~e |
   1.hhchild |   .5881852     .4628125        .1253727        .0810809
         age |  -.0003842    -.004825         .0044408        .0131965
    hhincome |  -.0122043    -.0046922       -.0075122        .0086532
   1.hhsigno |   .5090034     .4967056        .0122977        .0326296
   1.bwinner |  -.4655745    -.4740919        .0085173        .0287868
-------------+----------------------------------------------------------------
Unemployed   |
   1.hhchild |    .163612    -.0401989         .203811        .1120618
         age |   .0063355     .0042644        .0020711        .0175947
    hhincome |   -.029742    -.0308468        .0011048        .0117062
   1.hhsigno |   .1173192        .0968        .0205192        .0520686
   1.bwinner |  -.2489958    -.2252587       -.0237371        .0393297
------------------------------------------------------------------------------
                          b = Consistent under H0 and Ha; obtained from xtmlogit.
           B = Inconsistent under Ha, efficient under H0; obtained from xtmlogit.

Test of H0: Difference in coefficients not systematic

   chi2(10) = (b-B)'[(V_b-V_B)^(-1)](b-B)
            =   8.05
Prob > chi2 = 0.6238
```

Notice that we specified `hausman` such that we gave it the results of the estimator that is consistent under both H_0 and H_a first (the fixed-effects estimator) and the results of the estimator that is efficient under H_0 second (the random-effects estimator). We also specified the `alleqs` option to apply the test to all equations present in both models. The result of the test, $\chi^2 = 8.05$ with $\mathrm{df} = 10$ yielding $p = 0.62$, suggests that we do not reject H_0. In other words, here we may proceed with the random-effects estimator.

◁

❑ Technical note

The random-effects model is calculated using quadrature, which is an approximation whose accuracy depends partially on the number of integration points used. We can use the `quadchk` command to see if changing the number of integration points affects the results. If the results change, the quadrature approximation is not accurate given the number of integration points. Try increasing the number of integration points using the `intpoints()` option, and run `quadchk` again. Do not attempt to interpret the results of estimates when the coefficients reported by `quadchk` differ substantially. See [XT] **quadchk** for details and [XT] **xtprobit** for an example.

Because the `xtmlogit` likelihood function is calculated by Gauss–Hermite quadrature, on large problems computations can be slow. Computation time is roughly proportional to the number of points used for the quadrature.

❑

Stored results

xtmlogit, re stores the following in e():

Scalars

e(N)	number of observations
e(N_g)	number of groups
e(k)	number of parameters
e(k_out)	number of outcomes
e(k_eq)	number of equations in e(b)
e(k_eq_model)	number of equations in overall model test
e(k_eq_base)	equation number of the base outcome
e(baseout)	the value of *depvar* to be treated as the base outcome
e(ibaseout)	index of the base outcome
e(k_dv)	number of dependent variables
e(df_m)	model degrees of freedom
e(ll)	log likelihood
e(ll_0)	log likelihood, constant-only model
e(ll_c)	log likelihood, comparison model
e(chi2)	χ^2
e(chi2_c)	χ^2 for comparison test
e(p)	p-value for model test
e(p_c)	p-value for comparison test
e(df_c)	comparison test degrees of freedom
e(N_clust)	number of clusters
e(n_quad)	number of quadrature points
e(g_min)	smallest group size
e(g_avg)	average group size
e(g_max)	largest group size
e(rank)	rank of e(V)
e(rank0)	rank of e(V) for constant-only model
e(ic)	number of iterations
e(rc)	return code
e(converged)	1 if converged, 0 otherwise

Macros

e(cmd)	gsem
e(cmd2)	xtmlogit
e(cmdline)	command as typed
e(depvar)	name of dependent variable
e(wtype)	weight type
e(wexp)	weight expression
e(covariance)	random-effects covariance structure
e(ivar)	variable denoting groups
e(model)	re
e(title)	title in estimation output
e(distrib)	Gaussian; the distribution of the random effect
e(clustvar)	name of cluster variable
e(eqnames)	names of equations
e(baselab)	value label corresponding to base outcome
e(chi2type)	Wald or LR; type of model χ^2 test
e(vce)	*vcetype* specified in vce()
e(vcetype)	title used to label Std. err.
e(intmethod)	integration method
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(user)	name of likelihood-evaluator program
e(technique)	maximization technique
e(properties)	b V
e(predict)	program used to implement predict
e(estat_cmd)	program used to implement estat
e(marginsok)	predictions allowed by margins
e(marginsnotok)	predictions disallowed by margins

`e(marginsdefault)`	default `predict()` specification for `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`

Matrices

`e(b)`	coefficient vector
`e(Cns)`	constraints matrix
`e(out)`	outcome values
`e(ilog)`	iteration log
`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

`r(table)`	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.

`xtmlogit, fe` stores the following in `e()`:

Scalars

`e(N)`	number of observations
`e(N_g)`	number of groups
`e(N_drop)`	number of observations dropped because of no variation in outcome
`e(N_group_drop)`	number of groups dropped because of no variation in outcome
`e(k)`	number of parameters
`e(k_out)`	number of outcomes
`e(k_eq)`	number of equations in `e(b)`
`e(k_eq_model)`	number of equations in overall model test
`e(k_eq_base)`	equation number of the base outcome
`e(baseout)`	the value of *depvar* to be treated as the base outcome
`e(ibaseout)`	index of the base outcome
`e(nperm)`	number of permutations
`e(nperm_sampled)`	number of sampled permutations
`e(rsample)`	`1` if permutation sampling, `0` otherwise
`e(rssize)`	percentage of sampled permutations
`e(k_dv)`	number of dependent variables
`e(df_m)`	model degrees of freedom
`e(ll)`	log likelihood
`e(ll_0)`	log likelihood, baseline model
`e(chi2)`	χ^2
`e(p)`	p-value for model test
`e(N_clust)`	number of clusters
`e(g_min)`	smallest group size
`e(g_avg)`	average group size
`e(g_max)`	largest group size
`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)`	`xtmlogit`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(wtype)`	weight type
`e(wexp)`	weight expression

`e(ivar)`	variable denoting groups
`e(model)`	`fe`
`e(title)`	title in estimation output
`e(clustvar)`	name of cluster variable
`e(eqnames)`	names of equations
`e(baselab)`	value label corresponding to base outcome
`e(chi2type)`	`Wald` or `LR`; type of model χ^2 test
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. err.
`e(rs_rngstate)`	random-number state used
`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(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(marginsdefault)`	default `predict()` specification for `margins`
`e(asbalanced)`	factor variables `fvset` as `asbalanced`
`e(asobserved)`	factor variables `fvset` as `asobserved`
Matrices	
`e(b)`	coefficient vector
`e(Cns)`	constraints matrix
`e(out)`	outcome values
`e(ilog)`	iteration log
`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	
`r(table)`	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 panels $i = 1, \ldots, N$ with outcomes $j = 1, \ldots, J$ observed at times $t = 1, \ldots, T_i$, the model with unobserved heterogeneity at the panel level is

$$U_{ijt} = \mathbf{x}_{it}\boldsymbol{\beta}_j + u_{ij} + \epsilon_{ijt} \tag{2}$$

The variable U_{ijt} measures the utility of the ith panel toward outcome j at time t and is the sum of observed and unobserved components. The observed part of U_{ijt} consists of $\mathbf{x}_{it}$, a row vector of observed covariates of the ith panel at time t, and $\boldsymbol{\beta}_j$, a column vector of coefficients for the jth outcome. The vector of covariates is the same for each outcome, and the covariates do not vary over the outcomes for a given panel at a given time point. The unobserved part of U_{ijt} consists of u_{ij} and ϵ_{ijt}, where u_{ij} is a panel-level unobserved heterogeneity term and ϵ_{ijt} is the observation-level error term. For model identification, (2) must be normalized with respect to a base category.

Assuming a type-1 extreme value distribution for ϵ_{ijt} gives rise to the MNL model

$$\Pr(y_{it} = m \,|\, \mathbf{x}_{it}, \boldsymbol{\beta}, u_{ij}) = F(y_{it} = m, \mathbf{x}_{it}\boldsymbol{\beta}_j + u_{ij}) = \frac{\exp(\mathbf{x}_{it}\boldsymbol{\beta}_m + u_{im})}{\sum_{j=1}^{J} \exp(\mathbf{x}_{it}\boldsymbol{\beta}_j + u_{ij})}$$

For normalization, $\boldsymbol{\beta}_j$ and u_{ij} are set to zero for $j = b$, where b is the base outcome.

The random-effects estimator of `xtmlogit` assumes that $\mathbf{u}_i$ is distributed $\mathbf{u}_i \sim N(\mathbf{0}, \boldsymbol{\Sigma}_u)$. The likelihood for the ith panel is

$$\begin{aligned} l_i &= \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \left\{ \prod_{t=1}^{T_i} F(y_{it} = m, \mathbf{x}_{it}\boldsymbol{\beta}_j + u_{ij}) \right\} \phi(\mathbf{u}_i, \boldsymbol{\Sigma}_u)\, d\mathbf{u}_i \\ &\equiv \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f(y_{it} = m, \eta_{ijt})\, d\mathbf{u}_i \end{aligned}$$

where ϕ is the probability density function of the normal distribution and $\eta_{ijt} = \mathbf{x}_{it}\boldsymbol{\beta}_j + u_{ij}$. This integral of dimension $J-1$ must be approximated numerically because it has no closed-form solution.

In the univariate case, the integral of a function multiplied by the kernel of the standard normal distribution can be approximated using Gauss–Hermite quadrature. For q-point Gauss–Hermite quadrature, let the abscissa and weight pairs be denoted by (a_k^*, w_k^*), $k = 1, \dots, q$. The Gauss–Hermite quadrature approximation is then

$$\int_{-\infty}^{\infty} f(x) \exp(-x^2)\, dx \approx \sum_{k=1}^{q} w_k^* f(a_k^*)$$

Using the standard normal distribution yields the approximation

$$\int_{-\infty}^{\infty} f(x)\phi(x)\, dx \approx \sum_{k=1}^{q} w_k f(a_k)$$

where $a_k = \sqrt{2}a_k^*$ and $w_k = w_k^*/\sqrt{\pi}$.

We can use a change-of-variables technique to transform the multivariate integral into a set of nested univariate integrals. Each univariate integral can then be evaluated using Gauss–Hermite quadrature. Let $\boldsymbol{v}$ be a random vector whose elements are independently standard normal, and let $\mathbf{L}$ be the Cholesky decomposition of $\boldsymbol{\Sigma}_u$; that is, $\boldsymbol{\Sigma}_u = \mathbf{L}\mathbf{L}'$. In the distribution, we have that $\mathbf{u}_i \approx \mathbf{L}\boldsymbol{v}$, and the linear predictions vector as a function of $\boldsymbol{v}$ is

$$\widetilde{\eta}_{ijt} = \mathbf{x}_{it}\boldsymbol{\beta}_j + \mathbf{L}\boldsymbol{v}$$

so the likelihood for a given panel is

$$l_i = (2\pi)^{-r/2} \int_{-\infty}^{\infty} \dots \int_{-\infty}^{\infty} \exp\left\{ \log f(\mathbf{y}_i, \boldsymbol{\eta}_i) - \frac{1}{2}\sum_{k=1}^{r} v_k^2 \right\} dv_1 \dots dv_r$$

Consider an r-dimensional quadrature grid, $r = J-1$, containing q quadrature points in each dimension. Let the vector of abscissas $\boldsymbol{a}_{\mathbf{k}} = (a_{k_1}, \dots, a_{k_r})'$ be a point in this grid, and let $\boldsymbol{w}_{\mathbf{k}} = (w_{k_1}, \dots, w_{k_r})'$ be the vector of corresponding weights. The Gauss–Hermite quadrature approximation to the likelihood for a given panel is

$$\dot{l}_i = \sum_{k_1=1}^{q} \dots \sum_{k_r=1}^{q} \left[\exp\left\{ \sum_{t=1}^{T_i} \log f(y_{it} = m, \widetilde{\eta}_{ijt\mathbf{k}}) \right\} \prod_{s=1}^{r} w_{k_s} \right]$$

where

$$\widetilde{\eta}_{ijt\mathbf{k}} = \mathbf{x}_{it}\boldsymbol{\beta}_j + \mathbf{L}\boldsymbol{a}_\mathbf{k}$$

In the case of adaptive Gauss–Hermite quadrature, the likelihood is approximated with

$$\ddot{l}_i = \sum_{k_1=1}^{q} \cdots \sum_{k_r=1}^{q} \left[\exp\left\{ \sum_{t=1}^{T_i} \log f(y_{it} = m, \check{\eta}_{ijt\mathbf{k}}) \right\} \prod_{s=1}^{r} \omega_{k_s} \right]$$

where

$$\check{\eta}_{ijt\mathbf{k}} = \mathbf{x}_{it}\boldsymbol{\beta}_j + \mathbf{L}\boldsymbol{\alpha}_\mathbf{k}$$

and $\boldsymbol{\alpha}_\mathbf{k}$ and the ω_{k_s} are the adaptive versions of the abscissas and weights after an orthogonalizing transformation, which eliminates posterior covariances between the latent variables. $\boldsymbol{\alpha}_\mathbf{k}$ and the ω_{k_s} are functions of $\boldsymbol{a}_\mathbf{k}$ and $\boldsymbol{w}_\mathbf{k}$ as well as the posterior mean and the posterior variance of $\boldsymbol{v}$.

The fixed-effects estimator follows the derivations in Chamberlain (1980) and Pforr (2014). Let $\mathbf{Y}_i = (Y_{i1}, \ldots, Y_{iT_i})$ be the sequence of outcomes of the ith panel, and let $\mathbf{Y}_{it} = (Y_{i1t}, \ldots, Y_{iJt})$ be a vector with elements $Y_{ijt} = 1(i \text{ chooses } j \text{at } t)$ that indicate the chosen outcome of the ith panel at time t.

The distribution of times that panel i chose each of the J alternatives over time points T_i is the sufficient statistic $\Theta_i = \sum_{t=1}^{T_i} \mathbf{Y}_{it} = \mathbf{c}_i = (c_{i1}, \ldots, c_{iJ})$. Conditioning on the sufficient statistic Θ_i, we find the probability of panel i choosing a sequence $\mathbf{Y}_i = \mathbf{s}_i$ that is consistent with $\mathbf{c}_i$ is

$$\begin{aligned} \Pr(\mathbf{Y}_i = \mathbf{s}_i \,|\, \Theta_i, \mathbf{u}_i, \mathbf{x}_i, \boldsymbol{\beta}) &= \Pr\{Y_{i1}, \ldots, Y_{iT_i} \,|\, \Psi(\mathbf{c}_i), \mathbf{u}_i, \mathbf{x}_i, \boldsymbol{\beta}\} \\ &= \frac{\exp\left(\sum_{t=1}^{T_i} \sum_{j=1, j\neq b}^{J} Y_{ijt}\mathbf{x}_{it}\boldsymbol{\beta}_j \right)}{\sum_{\widetilde{Y}_{ijt}\in\Psi(\mathbf{c}_i)} \exp\left(\sum_{t=1}^{T_i} \sum_{j=1, j\neq b}^{J} \widetilde{Y}_{ijt}\mathbf{x}_{it}\boldsymbol{\beta}_j \right)} \end{aligned}$$

where $\Psi(\mathbf{c}_i)$ is the set of all permutations of individual i's observed sequence of outcomes that satisfy the condition $\sum_{t=1}^{T_i} \widetilde{\mathbf{Y}}_{it} = \mathbf{c}_i$. That is,

$$\Psi(\mathbf{c}_i) = \left\{ \widetilde{\mathbf{Y}}_i = (\widetilde{Y}_{i1}, \ldots, \widetilde{Y}_{iT_i}) \;\middle|\; \sum_{t=1}^{T_i} \widetilde{\mathbf{Y}}_{it} = \mathbf{c}_i \right\}$$

and $\widetilde{\mathbf{Y}}_{it} = (\widetilde{Y}_{i1t}, \ldots, \widetilde{Y}_{iJt})$ is a vector of indicators with respect to the permutations of the observed outcome sequence $\mathbf{Y}_i$. The log likelihood of panel i is then the natural logarithm of the above probability

$$\log l_i = \sum_{t=1}^{T_i} \sum_{j=1, j\neq b}^{J} Y_{ijt}\mathbf{x}_{it}\boldsymbol{\beta}_j - \log \sum_{\widetilde{Y}_{ijt}\in\Psi(\mathbf{c}_i)} \exp\left(\sum_{t=1}^{T_i} \sum_{j=1, j\neq b}^{J} \widetilde{Y}_{ijt}\mathbf{x}_{it}\boldsymbol{\beta}_j \right)$$

and the overall log likelihood is $\sum_{i=1}^{N} \log l_i$.

Consistent, albeit less efficient, estimates of the parameters in $\boldsymbol{\beta}_j$ can be obtained by taking a random sample of the elements in $\Psi(\mathbf{c}_i)$. The total number of permutations in $\Psi(\mathbf{c}_i)$ is

$$K_i = \frac{T_i!}{c_{i1}! \cdots c_{ij}! \cdots c_{iJ}!}$$

Let $\ddot{\Psi}(\mathbf{c}_i)$ be a random subset of $\Psi(\mathbf{c}_i)$. $\ddot{\Psi}(\mathbf{c}_i)$ consists of $L_i + 1$ elements, where L_i elements are randomly drawn without replacement and equal probability from the set $\Psi(\mathbf{c}_i)$ that has the observed sequence of outcomes removed, and then the observed sequence is added such that $\ddot{\Psi}(\mathbf{c}_i)$ always contains the observed outcome sequence. The log likelihood with sampled permutations is

$$\log l_i = \sum_{t=1}^{T_i} \sum_{j=1, j\neq b}^{J} Y_{ijt}\mathbf{x}_{it}\boldsymbol{\beta}_j - \log \sum_{\widetilde{Y}_{ijt} \in \ddot{\Psi}(\mathbf{c}_i)} \exp\left(\sum_{t=1}^{T_i} \sum_{j=1, j\neq b}^{J} \widetilde{Y}_{ijt}\mathbf{x}_{it}\boldsymbol{\beta}_j \right)$$

The above is a consistent estimator of $\boldsymbol{\beta}_j$ but is less efficient compared with using the full set of permutations $\Psi(\mathbf{c}_i)$ because of the added Monte Carlo variance. The smaller the size of the sample relative to K_i, the number of all permutations, the less efficient it becomes. The permutation sampling implemented in `xtmlogit` follows the approach outlined in D'Haultfœuille and Iaria (2016).

References

Andersen, E. B. 1970. Asymptotic properties of conditional maximum likelihood estimators. *Journal of the Royal Statistical Society, Series B* 32: 283–301. https://doi.org/10.1111/j.2517-6161.1970.tb00842.x.

Börsch-Supan, A. 1990. Panel data analysis of housing choices. *Regional Science and Urban Economics* 20: 65–82. https://doi.org/10.1016/0166-0462(90)90025-X.

Chamberlain, G. 1980. Analysis of covariance with qualitative data. *Review of Economic Studies* 47: 225–238. https://doi.org/10.2307/2297110.

D'Haultfœuille, X., and A. Iaria. 2016. A convenient method for the estimation of the multinomial logit model with fixed effects. *Economics Letters* 141: 77–79. https://doi.org/10.1016/j.econlet.2016.02.002.

Grilli, L., and C. Rampichini. 2007. A multilevel multinomial logit model for the analysis of graduates' skills. *Statistical Methods and Applications* 16: 381–393. https://doi.org/10.1007/s10260-006-0039-z.

Hartzel, J., A. Agresti, and B. S. Caffo. 2001. Multinomial logit random effects models. *Statistical Modelling* 1: 81–102. https://doi.org/10.1177/1471082X0100100201.

Lancaster, T. 2000. The incidental parameter problem since 1948. *Journal of Econometrics* 95: 391–413. https://doi.org/10.1016/S0304-4076(99)00044-5.

Pforr, K. 2014. femlogit—Implementation of the multinomial logit model with fixed effects. *Stata Journal* 14: 847–862.

Also see

[XT] **xtmlogit postestimation** — Postestimation tools for xtmlogit

[XT] **quadchk** — Check sensitivity of quadrature approximation

[XT] **xtlogit** — Fixed-effects, random-effects, and population-averaged logit models

[XT] **xtset** — Declare data to be panel data

[BAYES] **bayes: xtmlogit** — Bayesian random-effects multinomial logit model

[R] **clogit** — Conditional (fixed-effects) logistic regression

[R] **mlogit** — Multinomial (polytomous) logistic regression

[R] **mprobit** — Multinomial probit regression

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

[U] 20 Estimation and postestimation commands