**Description**

*nlogit* performs full information maximum-likelihood estimation for nested logit models. These models relax the assumption of independently distributed errors and the independence of irrelevant alternatives inherent in conditional and multinomial logit models by clustering similar alternatives into nests.

By default, *nlogit* uses a parameterization that is consistent with random utility maximization (RUM). Before version 10 of Stata, a nonnormalized version of the nested logit model was fit, which you can request by specifying the *nonnormalized* option.

You must use *nlogitgen* to generate a new categorical variable to specify the branches of the decision tree before calling *nlogit*.

**Quick start**

Create variables identifying alternatives at higher levels

For a three-level nesting structure, create 12alt identifying four alternatives at level two based on the variable balt which identifies eight bottom-level alternatives

```
nlogitgen 12alt = balt(12alt1: balt1 | balt2, 12alt2: balt3 | balt4, ///
                      12alt3: balt5 | balt6, 12alt4: balt7 | balt8)
```

Same as above, defining 12alt from values of balt rather than from value labels

```
nlogitgen 12alt = balt(12alt1: 1|2, 12alt2: 3|4, 12alt3: 5|6, ///
                      12alt4: 7|8)
```

Create talt identifying top-level alternatives based on the four alternatives in 12alt

```
nlogitgen talt = 12alt(talt1: 12alt1 | 12alt2, talt2: ltalt3 | 12alt4)
```

Examine tree structure

Display three-level nesting structure

```
nlogittree balt 12alt talt
```

Also report choice frequencies of indicator chosen for each bottom-level alternative

```
nlogittree balt 12alt talt, choice(chosen)
```

Identify potentially problematic observations before fitting the model by specifying case identifier casevar

```
nlogittree balt 12alt talt, choice(chosen) case(casevar)
```
As above, and generate prob indicating problematic observations
\[ \text{nlogittree balt l2alt talt, choice(chosen) case(casevar) generate(prob)} \]

**Fit nested logit model**

Three-level model with alternative-specific covariate \( x_1 \), case-specific covariate \( x_2 \) modeling top-level choice, and case-specific covariate \( x_3 \) modeling level-two choice
\[ \text{nlogit chosen x1 || talt: x2 || 12alt: x3 || balt:, case(casevar)} \]

As above, but do not estimate intercepts for bottom-level alternatives
\[ \text{nlogit chosen x1 || talt: x2 || 12alt: x3 || balt:, noconstant ///} \]
\[ \text{case(casevar)} \]

As above, but estimate intercepts for top-level alternatives
\[ \text{nlogit chosen x1 || talt: x2, estconst || 12alt: x3 ///} \]
\[ || balt:, noconstant case(casevar) \]

Specify the base alternative at each level
\[ \text{nlogit chosen x1 || talt: x2, base(talt2) || 12alt: x3, base(12alt4) ///} \]
\[ || balt:, base(balt8) case(casevar) \]

Use nonnormalized parameterization
\[ \text{nlogit chosen x1 || talt: x2 || 12alt: x3 || balt:, case(casevar) ///} \]
\[ \text{nonnormalized} \]

**Menu**

**nlogit**
Statistics > Categorical outcomes > Nested logit regression

**nlogitgen**
Statistics > Categorical outcomes > Setup for nested logit regression

**nlogittree**
Statistics > Categorical outcomes > Display nested logit tree structure
Syntax

Nested logit regression

\[\text{nlogit} \, \text{depvar} \, [\text{indepvars}] \, [\text{if}] \, [\text{in}] \, [\text{weight}] \, [\llp \text{lev1}_\text{equation} \rlp \llp \text{lev2}_\text{equation} \ldots \rlp \llp \text{altvar} : [\text{byaltvarlist}] \], \text{case(varname)} \, [\text{nlogit_options}]\]

where the syntax of \(\text{lev#}_\text{equation}\) is

\[\text{altvar} : [\text{byaltvarlist}] \, [\llp , \text{base(#|lbl)} \llp \text{estconst} \rlp ]\]

Create variable based on specification of branches

\[\text{nlogitgen} \, \text{newaltvar} = \text{altvar} \, (\text{branchlist}) \, [\llp , \text{nolog} \rlp ]\]

where \(\text{branchlist}\) is

\[\text{branch} , \text{branch} \, [\llp , \text{branch} \ldots \rlp ]\]

and \(\text{branch}\) is

\[\llp [\text{label}:] \llp \text{alternative} \, [\llp , \text{alternative} \llp \text{alternative} \ldots \rlp ]\rlp \]

Display tree structure

\[\text{nlogittree} \, \text{altvarlist} \, [\text{if}] \, [\text{in}] \, [\text{weight}] \, [\llp , \text{nlogittree_options} \rlp ]\]
### nlogit_options

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Model</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>* case(varname)</td>
</tr>
<tr>
<td>base(#</td>
</tr>
<tr>
<td>noconstant</td>
</tr>
<tr>
<td>nonconstant</td>
</tr>
<tr>
<td>altwise</td>
</tr>
<tr>
<td>constraints(constraints)</td>
</tr>
<tr>
<td>collinear</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>SE/Robust</strong></td>
</tr>
<tr>
<td>vce(vcetype)</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Reporting</strong></td>
</tr>
<tr>
<td>level(#)</td>
</tr>
<tr>
<td>notree</td>
</tr>
<tr>
<td>nocnsreport</td>
</tr>
<tr>
<td>display_options</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>Maximization</strong></td>
</tr>
<tr>
<td>maximize_options</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>*case(varname) is required.</td>
</tr>
</tbody>
</table>

### nlogittree_options

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Main</strong></td>
</tr>
<tr>
<td>choice(depvar)</td>
</tr>
<tr>
<td>case(varname)</td>
</tr>
<tr>
<td>generate(newvar)</td>
</tr>
<tr>
<td>nolabel</td>
</tr>
<tr>
<td>nobranches</td>
</tr>
</tbody>
</table>

byaltvarlist may contain factor variables; see [U] 11.4.3 Factor variables.

bootstrap, by, fp, jackknife, and statsby are allowed; see [U] 11.1.10 Prefix commands.

Weights are not allowed with the bootstrap prefix; see [R] bootstrap.

fweights, iweights, and pweights are allowed with nlogit, and fweights are allowed with nlogittree; see [U] 11.1.6 weight. Weights for nlogit must be constant within case.

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

Specification and options for lev#_equation

altvar is a variable identifying alternatives at this level of the hierarchy.

byaltvarlist specifies the variables to be used to compute the by-alternative regression coefficients for that level. For each variable specified in the variable list, there will be one regression coefficient for each alternative of that level of the hierarchy. If the variable is constant across each alternative (a case-specific variable), the regression coefficient associated with the base alternative is not identifiable. These regression coefficients are labeled as (base) in the regression table. If the variable varies among the alternatives, a regression coefficient is estimated for each alternative.

base(# | lbl) can be specified in each level equation where it identifies the base alternative to be used at that level. The default is the alternative that has the highest frequency.

If vce(bootstrap) or vce(jackknife) is specified, you must specify the base alternative for each level that has a byaltvarlist or if the constants will be estimated. Doing so ensures that the same model is fit with each call to nlogit.

estconst applies to all the level equations except the bottom-level equation. Specifying estconst requests that constants for each alternative (except the base alternative) be estimated. By default, no constant is estimated at these levels. Constants can be estimated in only one level of the tree hierarchy. If you specify estconst for one of the level equations, you must specify noconstant for the bottom-level equation.

Options for nlogit

Model

case(varname) specifies the variable that identifies each case. case() is required.

base(# | lbl) can be specified in each level equation where it identifies the base alternative to be used at that level. The default is the alternative that has the highest frequency.

If vce(bootstrap) or vce(jackknife) is specified, you must specify the base alternative for each level that has a byaltvarlist or if the constants will be estimated. Doing so ensures that the same model is fit with each call to nlogit.

noconstant applies only to the equation defining the bottom level of the hierarchy. By default, constants are estimated for each alternative of altvar, less the base alternative. To suppress the constant terms for this level, specify noconstant. If you do not specify noconstant, you cannot specify estconst for the higher-level equations.

nonnormalized requests a nonnormalized parameterization of the model that does not scale the inclusive values by the degree of dissimilarity of the alternatives within each nest. Use this option to replicate results from older versions of Stata. The default is to use the RUM–consistent parameterization.

altwise specifies that alternativewise deletion be used when marking out observations because of missing values in your variables. The default is to use casewise deletion. This option does not apply to observations that are marked out by the if or in qualifier or the by prefix.
constraints(constRAINTS); see [R] estimation options.

The inclusive-valued/dissimilarity parameters are parameterized as ml ancillary parameters. They are labeled as [alternative_tau]_const, where alternative is one of the alternatives defining a branch in the tree. To constrain the inclusive-valued/dissimilarity parameter for alternative a1 to be, say, equal to alternative a2, you would use the following syntax:

    . constraint 1 [a1_tau]_cons = [a2_tau]_cons
    . nlogit ..., constraints(1)

collinear prevents collinear variables from being dropped. Use this option when you know that you have collinear variables and you are applying constraints() to handle the rank reduction. See [R] estimation options for details on using collinear with constraints().

nlogit will not allow you to specify an independent variable in more than one level equation. Specifying the collinear option will allow execution to proceed in this case, but it is your responsibility to ensure that the parameters are identified.

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 [R] vce_option.

If vce(robust) or vce(cluster clustvar) is specified, the likelihood-ratio test for the independence of irrelevant alternatives (IIA) is not computed.

Reporting

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

notree specifies that the tree structure of the nested logit model not be displayed. See also nolabel and nobranches below for when notree is not specified.

nocnsreport; see [R] estimation options.

display_options: noci, nopvalues, nonomitted, vsquish, noemptycells, baselevels, allbaselevels, nocflabel, fwrap(#), fvwrap(style), cformat(%fmt), pformat(%fmt), sformat(%fmt), and nolstretch; 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 technique(bhhh) option is not allowed.

Specification and options for nlogitgen

newaltvar and altvar are variables identifying alternatives at each level of the hierarchy.

label defines a label to associate with the branch. If no label is given, a numeric value is used.

alternative specifies an alternative, of altvar specified in the syntax, to be included in the branch. It is either a numeric value or the label associated with that value. An example of nlogitgen is

    . nlogitgen type = restaurant(fast: 1 | 2, > family: CafeEccell | LosNortenos | WingsNmore, fancy: 6 | 7)

nolog suppresses the display of the iteration log.
Specification and options for nlogittree

- `altvarlist` is a list of alternative variables that define the tree hierarchy. The first variable must define bottom-level alternatives, and the order continues to the variable defining the top-level alternatives.

- `choice(depvar)` defines the choice indicator variable and forces `nlogittree` to compute and display choice frequencies for each bottom-level alternative.

- `case(varname)` specifies the variable that identifies each case. When both `case()` and `choice()` are specified, `nlogittree` executes diagnostics on the tree structure and identifies observations that will cause `nlogit` to terminate execution or drop observations.

- `generate(newvar)` generates a new indicator variable, `newvar`, that is equal to 1 for invalid observations. This option requires that both `choice()` and `case()` are also specified.

- `nolabel` forces `nlogittree` to suppress value labels in tree-structure output.

- `nobranches` forces `nlogittree` to suppress drawing branches in the tree-structure output.

Remarks and examples

Remarks are presented under the following headings:

- Introduction
- Data setup and the tree structure
- Estimation
- Testing for the IIA
- Nonnormalized model

Introduction

`nlogit` performs full information maximum-likelihood estimation for nested logit models. These models relax the assumption of independently distributed errors and the IIA inherent in conditional and multinomial logit models by clustering similar alternatives into nests. Because the nested logit model is a direct generalization of the alternative-specific conditional logit model (also known as McFadden’s choice model), you may want to read [R] `asclogit` before continuing.

By default, `nlogit` uses a parameterization that is consistent with RUM. Before version 10 of Stata, a nonnormalized version of the nested logit model was fit, which you can request by specifying the `nonnormalized` option. We recommend using the RUM-consistent version of the model for new projects because it is based on a sound model of consumer behavior.

McFadden (1977, 1981) showed how this model can be derived from a rational choice framework. Amemiya (1985, chap. 9) contains a nice discussion of how this model can be derived under the assumption of utility maximization. Hensher, Rose, and Greene (2005) provide a lucid introduction to choice models including nested logit.

Throughout this entry, we consider a model of restaurant choice. We begin by introducing the data.

Example 1

We have fictional data on 300 families and their choice of seven local restaurants. Freebirds and Mama’s Pizza are fast food restaurants; Café Eccell, Los Nortenos, and Wings ’N More are family restaurants; and Christopher’s and Mad Cows are fancy restaurants. We want to model the decision...
of where to eat as a function of household income (income, in thousands of dollars), the number of children in the household (kids), the rating of the restaurant according to a local restaurant guide (rating, coded 0–5), the average meal cost per person (cost), and the distance between the household and the restaurant (distance, in miles). income and kids are attributes of the family, rating is an attribute of the alternative (the restaurant), and cost and distance are attributes of the alternative as perceived by the families—that is, each family has its own cost and distance for each restaurant.

We begin by loading the data and listing some of the variables for the first three families:

```
. use http://www.stata-press.com/data/r14/restaurant
. describe

Contains data from http://www.stata-press.com/data/r14/restaurant.dta
obs: 2,100
vars: 8 2 Dec 2014 15:07
size: 33,600

     variable name storage display label      variable label
   ----------------------------------------
family_id     int %9.0g           family ID
restaurant    byte %12.0g  names     choices of restaurants
    income     int %9.0g           household income
cost          float %9.0g          average meal cost per person
    kids       byte %9.0g           number of kids in the household
   rating       byte %9.0g      ratings in local restaurant guide
distance      float %9.0g     distance between home and restaurant
   chosen       byte %9.0g            0 no 1 yes

Sorted by: family_id
```
Because each family chose among seven restaurants, there are 7 observations in the dataset for each family. The variable `chosen` is coded 0/1, with 1 indicating the chosen restaurant and 0 otherwise.

We could fit a conditional logit model to our data. Because `income` and `kids` are constant within each family, we would use the `asclogit` command instead of `clogit`. However, the conditional logit may be inappropriate. That model assumes that the random errors are independent, and as a result it forces the odds ratio of any two alternatives to be independent of the other alternatives, a property known as the IIA. We will discuss the IIA assumption in more detail later.

Assuming that unobserved shocks influencing a decision maker’s attitude toward one alternative have no effect on his attitudes toward the other alternatives may seem innocuous, but often this assumption is too restrictive. Suppose that when a family was deciding which restaurant to visit, they were pressed for time because of plans to attend a movie later. The unobserved shock (being in a hurry) would raise the likelihood that the family goes to either fast food restaurant (Freebirds or Mama’s Pizza). Similarly, another family might be choosing a restaurant to celebrate a birthday and therefore be inclined to attend a fancy restaurant (Christopher’s or Mad Cows).

Nested logit models relax the independence assumption and allow us to group alternatives for which unobserved shocks may have concomitant effects. Here we suspect that restaurants should be grouped by type (fast, family, or fancy). The tree structure of a family’s decision about where to eat might look like this:
At the bottom of the tree are the individual restaurants, indicating that there are some random shocks that affect a family’s decision to eat at each restaurant independently. Above the restaurants are the three types of restaurants, indicating that other random shocks affect the type of restaurant chosen. As is customary when drawing decision trees, at the top level is one box, representing the family making the decision.

We use the following terms to describe nested logit models.

- **level**, or decision level, is the level or stage at which a decision is made. The example above has only two levels. In the first level, a type of restaurant is chosen—fast food, family, or fancy—and in the second level, a specific restaurant is chosen.

- **bottom level** is the level where the final decision is made. In our example, this is when we choose a specific restaurant.

- **alternative set** is the set of all possible alternatives at any given decision level.

- **bottom alternative set** is the set of all possible alternatives at the bottom level. This concept is often referred to as the choice set in the economics-choice literature. In our example, the bottom alternative set is all seven of the specific restaurants.

- **alternative** is a specific alternative within an alternative set. In the first level of our example, “fast food” is an alternative. In the second or bottom level, “Mad Cows” is an alternative. Not all alternatives within an alternative set are available to someone making a choice at a specific stage, only those that are nested within all higher-level decisions.

- **chosen alternative** is the alternative from an alternative set that we observe someone having chosen.

**Technical note**

Although decision trees in nested logit analysis are often interpreted as implying that the highest-level decisions are made first, followed by decisions at lower levels, and finally the decision among alternatives at the bottom level, no such temporal ordering is implied. See Hensher, Rose, and Greene (2005, chap. 13). In our example, we are not assuming that families first choose whether to attend a fast, family, or fancy restaurant and then choose the particular restaurant; we assume merely that they choose one of the seven restaurants.
Data setup and the tree structure

To fit a nested logit model, you must first create a variable that defines the structure of your decision tree.

Example 2

To run \texttt{nlogit}, we need to generate a categorical variable that identifies the first-level set of alternatives: fast food, family restaurants, or fancy restaurants. We can do so easily by using \texttt{nlogitgen}.

\begin{verbatim}
.nlogitgen type = restaurant(fast: Freebirds | MamasPizza, > family: CafeEccell | LosNortenos| WingsNmore, fancy: Christophers | MadCows)
new variable type is generated with 3 groups
label list lb_type
lb_type:
   1 fast
   2 family
   3 fancy
.nlogittree restaurant type, choice(chosen)

tree structure specified for the nested logit model

<table>
<thead>
<tr>
<th>type</th>
<th>N</th>
<th>restaurant</th>
<th>N</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>fast</td>
<td>600</td>
<td>Freebirds</td>
<td>300</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MamasPizza</td>
<td>300</td>
<td>15</td>
</tr>
<tr>
<td>family</td>
<td>900</td>
<td>CafeEccell</td>
<td>300</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LosNortenos</td>
<td>300</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WingsNmore</td>
<td>300</td>
<td>69</td>
</tr>
<tr>
<td>fancy</td>
<td>600</td>
<td>Christophers</td>
<td>300</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MadCows</td>
<td>300</td>
<td>24</td>
</tr>
</tbody>
</table>

k = number of times alternative is chosen
N = number of observations at each level

The new categorical variable is \texttt{type}, which takes on value 1 (fast) if \texttt{restaurant} is Freebirds or Mama’s Pizza; value 2 (family) if \texttt{restaurant} is Café Eccell, Los Nortenos, or Wings ’N More; and value 3 (fancy) otherwise. \texttt{nlogittree} displays the tree structure.

Technical note

We could also use values instead of value labels of \texttt{restaurant} in \texttt{nlogitgen}. Value labels are optional, and the default value labels for \texttt{type} are \texttt{type1}, \texttt{type2}, and \texttt{type3}. The vertical bar is also optional.

\begin{verbatim}
.use http://www.stata-press.com/data/r14/restaurant, clear
.nlogitgen type = restaurant(1 2, 3 4 5, 6 7)
new variable type is generated with 3 groups
label list lb_type
lb_type:
   1 type1
   2 type2
   3 type3
\end{verbatim}
. nlogittree

restaurant type

tree structure specified for the nested logit model

type N restaurant N

type1 600 Freebirds 300
MamasPizza 300

type2 900 CafeEccell 300
LosNortenos 300
WingsNmore 300

type3 600 Christophers 300
MadCows 300

total 2100

N = number of observations at each level

In our dataset, every family was able to choose among all seven restaurants. However, in other applications some decision makers may not have been able to choose among all possible alternatives. For example, two cases may have choice hierarchies of

<table>
<thead>
<tr>
<th>case 1</th>
<th>case 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>restaurant</td>
</tr>
<tr>
<td>fast</td>
<td>Freebirds</td>
</tr>
<tr>
<td></td>
<td>MamasPizza</td>
</tr>
<tr>
<td>family</td>
<td>CafeEccell</td>
</tr>
<tr>
<td></td>
<td>LosNortenos</td>
</tr>
<tr>
<td>fancy</td>
<td>Christophers</td>
</tr>
<tr>
<td></td>
<td>MadCows</td>
</tr>
</tbody>
</table>

where the second case does not have the restaurant alternatives Café Eccell or Mad Cows available to them. The only restriction is that the relationships between higher- and lower-level alternative sets be the same for all decision makers. In this two-level example, Freebirds and Mama’s Pizza are classified as fast food restaurants for both cases; Café Eccell, Los Nortenos, and Wings ’N More are family restaurants; and Christopher’s and Mad Cows are fancy restaurants. \texttt{nlogit} requires only that hierarchy be maintained for all cases.

\section*{Estimation}

\subsection*{Example 3}

With our \texttt{type} variable created that defines the three types of restaurants, we can now examine how the alternative-specific attributes (\texttt{cost}, \texttt{rating}, and \texttt{distance}) apply to the bottom alternative set (the seven restaurants) and how family-specific attributes (\texttt{income} and \texttt{kid}) apply to the alternative set at the first decision level (the three types of restaurants).
use http://www.stata-press.com/data/r14/restaurant, clear
qui nlogitgen type = restaurant(fast: Freebirds | MamasPizza,
  family: CafeEccell | LosNortenos | WingsNmore, fancy: Christophers | MadCows)
nlogit chosen cost rating distance || type: income kids, base(family) ||
  restaurant:, noconstant case(family_id)

Tree structure specified for the nested logit model

<table>
<thead>
<tr>
<th>type</th>
<th>N</th>
<th>restaurant</th>
<th>N</th>
<th>k</th>
</tr>
</thead>
<tbody>
<tr>
<td>fast</td>
<td>600</td>
<td>Freebirds</td>
<td>300</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MamasPizza</td>
<td>300</td>
<td>15</td>
</tr>
<tr>
<td>family</td>
<td>900</td>
<td>CafeEccell</td>
<td>300</td>
<td>78</td>
</tr>
<tr>
<td></td>
<td></td>
<td>LosNortenos</td>
<td>300</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WingsNmore</td>
<td>300</td>
<td>69</td>
</tr>
<tr>
<td>fancy</td>
<td>600</td>
<td>Christophers</td>
<td>300</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MadCows</td>
<td>300</td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>total</td>
<td>2100</td>
<td>300</td>
<td></td>
</tr>
</tbody>
</table>

k = number of times alternative is chosen
N = number of observations at each level
Iteration 0:  log likelihood = -541.93581
(output omitted)
Iteration 17: log likelihood = -485.47331

RUM-consistent nested logit regression
Number of obs = 2,100
Case variable: family_id
Number of cases = 300
Alternative variable: restaurant
Alts per case: min = 7
avg = 7.0
max = 7
Wald chi2(7) = 46.71
Log likelihood = -485.47331

| chosen          | Coef.  | Std. Err. | z     | P>|z| | [95% Conf. Interval] |
|-----------------|--------|-----------|-------|-----|----------------------|
| restaurant      |        |           |       |     |                      |
| cost            | -.1843847 | .0933975 | -1.97 | 0.048  | -.3674404 -.0013289 |
| rating          | .463694  | .3264935 | 1.42  | 0.156  | -.1762215 1.10361  |
| distance        | -.3797474 | .1003828 | -3.78 | 0.000  | -.5764941 -.1830007 |
| type equations  |        |           |       |     |                      |
| fast            |        |           |       |     |                      |
| income          | -.0266038 | .0117306 | -2.27 | 0.023  | -.0495952 -.0036123 |
| kids            | -.0872584 | .1385026 | -0.63 | 0.529  | -.3587184 .1842016 |
| family          |        |           |       |     |                      |
| income          | 0 (base)|         |       |     |                      |
| kids            | 0 (base)|         |       |     |                      |
| fancy           |        |           |       |     |                      |
| income          | .0461827 | .00909936 | 5.08  | 0.000  | .0283595 .0640059  |
| kids            | -.3959413 | .1220356 | -3.24 | 0.001  | -.6351267 -.1567559 |
| dissimilarity parameters |        |           |       |     |                      |
| type            |        |           |       |     |                      |
| /fastTau        | 1.712878 | 1.48865  |       |       | -1.201295 4.627051  |
| /familyTau      | 2.505113 | .9646351 |       |       | .614463 4.395763   |
| /fancyTau       | 4.099844 | 2.810123 |       |       | -1.407896 9.607583  |

LR test for IIA (tau=1): chi2(3) = 6.87  Prob > chi2 = 0.0762
First, let’s examine how we called nlogit. The delimiters (||) separate equations. The first equation specifies the dependent variable, chosen, and three alternative-specific variables, cost, rating, and distance. We refer to these variables as alternative-specific because they vary among the bottom-level alternatives, the restaurants. We obtain one parameter estimate for each variable. These estimates are listed in the equation subtable labeled restaurant.

For the second equation, we specify the type variable. It identifies the first-level alternatives, the restaurant types. Following the colon after type, we specify two case-specific variables, income and kids. Here we obtain a parameter estimate for each variable for each alternative at this level. That is why we call these variable lists by-alternative variables. Because income and kids do not vary within each case, to identify the model one alternative’s set of parameters must be set to zero. We specified the base(family) option with this equation to restrict the parameters for the family alternative.

The variable identifying the bottom-level alternatives, restaurant, is specified after the second equation delimiter. We do not specify any variables after the colon delimiter at this level. Had we specified variables here, we would have obtained an estimate for each variable in each equation. As we will see below, these variables parameterize the constant term in the utility equation for each bottom-level alternative. The noconstant option suppresses bottom-level alternative-specific constant terms.

Near the bottom of the output are the dissimilarity parameters, which measure the degree of correlation of random shocks within each of the three types of restaurants. Dissimilarity parameters greater than one imply that the model is inconsistent with RUM; Hensher, Rose, and Greene (2005, sec. 13.6) discuss this in detail. We will ignore the fact that all our dissimilarity parameters exceed one.

The conditional logit model is a special case of nested logit in which all the dissimilarity parameters are equal to one. At the bottom of the output, we find a likelihood-ratio test of this hypothesis. Here we have mixed evidence of the null hypothesis that all the parameters are one. Equivalently, the property known as the IIA imposed by the conditional logit model holds if and only if all dissimilarity parameters are equal to one. We discuss the IIA in more detail now.

Testing for the IIA

The IIA is a property of the multinomial and conditional logit models that forces the odds of choosing one alternative over another to be independent of the other alternatives. For simplicity, suppose that a family was choosing only between Freebirds and Mama’s Pizza, and the family was equally likely to choose either of the restaurants. The probability of going to each restaurant is 50%. Now suppose that Bill’s Burritos opens up next door to Freebirds, which is also a burrito restaurant. If the IIA holds, then the probability of going to each restaurant must now be 33.33% so that the family remains equally likely to go to Mama’s Pizza or Freebirds.

The IIA may sometimes be a plausible assumption. However, a more likely scenario would be for the probability of going to Mama’s Pizza to remain at 50% and the probabilities of going to Freebirds and Bill’s Burritos to be 25% each, because the two restaurants are next door to each other and serve the same food. Nested logit analysis would allow us to relax the IIA assumption of conditional logit. We could group Bill’s Burritos and Freebirds into one nest that encompasses all burrito restaurants and create a second nest for pizzerias.

The IIA is a consequence of assuming that the errors are independent and identically distributed (i.i.d.). Because the errors are i.i.d., they cannot contain any alternative-specific unobserved information, and therefore adding a new alternative cannot affect the relationship between a pair of existing alternatives.
In the previous example, we saw that a joint test that the dissimilarity parameters were equal to one is one way to test for IIA. However, that test required us to specify a decision tree for the nested logit model, and different specifications could lead to conflicting results of the test. Hausman and McFadden (1984) suggest that if part of the choice set truly is irrelevant with respect to the other alternatives, omitting that subset from the conditional logit model will not lead to inconsistent estimates. Therefore, Hausman’s (1978) specification test can be used to test for IIA, and this test will not be sensitive to the tree structure we specify for a nested logit model.

**Example 4**

We want to test the IIA for the subset of family restaurants against the alternatives of fast food and fancy restaurants. To do so, we need to use Stata’s `hausman` command; see [R] hausman.

We first run the estimation on the full bottom alternative set, store the results by using `estimates store`, and then run the estimation on the bottom alternative set, excluding the alternatives of family restaurants. We then run the `hausman` test.

```
. generate incFast = (type == 1) * income
. generate incFancy = (type == 3) * income
. generate kidFast = (type == 1) * kids
. generate kidFancy = (type == 3) * kids
. clogit chosen cost rating distance incFast incFancy kidFast kidFancy, group(family_id) nolog
   Conditional (fixed-effects) logistic regression
   Number of obs = 2,100
   LR chi2(7) = 189.73
   Prob > chi2 = 0.0000
   Log likelihood = -488.90834  Pseudo R2 = 0.1625

   chosen        Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
   ------------- -------- -------- -------- -------- ---------------------
      cost    -.1367799   .0358479   -3.82   0.000    -.2070404   -.0665193
     rating     .3066622   .1418291     2.16   0.031     .0286823    0.084642
     distance  -.1977505   .0471653   -4.19   0.000    -.2901927   -.1053082
    incFast     -.0390183   .0094018   -4.15   0.000    -.0574455   -.0205911
    incFancy     .0407053   .0080405     5.06   0.000     .0249462    0.0564644
    kidFast  -.2398757   .1063674   -2.26   0.024    -.4483520   -.0313994
    kidFancy   -.3893862   .1143797   -3.40   0.001    -.6135662   -.1652061

. estimates store fullset
. clogit chosen cost rating distance incFast kidFast if type != 2, group(family_id) nolog
   Conditional (fixed-effects) logistic regression
   Number of obs = 312
   LR chi2(5) = 44.35
   Prob > chi2 = 0.0000
   Log likelihood = -85.955324  Pseudo R2 = 0.2051

   chosen        Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
   ------------- -------- -------- -------- -------- ---------------------
      cost    -.0616621   .0678525   -0.91   0.363    -.1946496    .0713254
     rating     .1659001   .2832041     0.59   0.558    -.3891698    .720977
     distance  -.2443962   .0995056   -2.46   0.014    -.4394234   -.0493687
    incFast  -.0737506   .0177444   -4.16   0.000    -.1085290   -.0389721
    kidFast    .4105386   .2137051     1.92   0.055    -.0083157    .8293928
```

Note: 222 groups (888 obs) dropped because of all positive or all negative outcomes.
Similar to our findings in example 3, the results of the test of the IIA are mixed. We cannot reject the IIA at the commonly used 5% significance level, but we could at the 10% level. Substantively, a significant test result suggests that the odds of going to one of the fancy restaurants versus going to one of the fast food restaurants changes if we include the family restaurants in the alternative set and that a nested logit specification may be warranted.

Nonnormalized model

Previous versions of Stata fit a nonnormalized nested logit model that is available via the *nonnormalized* option. The nonnormalized version is presented in, for example, Greene (2012, 768–770). Here we outline the differences between the RUM-consistent and nonnormalized models. Our discussion follows Heiss (2002) and assumes the decision tree has two levels, with $M$ alternatives at the upper level and a total of $J$ alternatives at the bottom level.

In a RUM framework, by consuming alternative $j$, decision maker $i$ obtains utility

$$U_{ij} = V_{ij} + \epsilon_{ij} = \alpha_j + x_{ij}\beta_j + z_i\gamma_j + \epsilon_{ij}$$

where $V_{ij}$ is the deterministic part of utility and $\epsilon_{ij}$ is the random part. $x_{ij}$ are alternative-specific variables and $z_i$ are case-specific variables. The set of errors $\epsilon_{i1}, \ldots, \epsilon_{iJ}$ are assumed to follow the generalized extreme-value (GEV) distribution, which is a generalization of the type 1 extreme-value distribution that allows for alternatives within nests of the tree structure to be correlated. Let $\rho_m$ denote the correlation in nest $m$, and define the dissimilarity parameter $\tau_m = \sqrt{1 - \rho_m}$. $\tau_m = 0$ implies that the alternatives in nest $m$ are perfectly correlated, whereas $\tau_m = 1$ implies independence.

The inclusive value for the $m$th nest corresponds to the expected value of the utility that decision maker $i$ obtains by consuming an alternative in nest $m$. Denote this value by $IV_m$:

$$IV_m = \ln \sum_{j \in B_m} \exp(V_k/\tau_m)$$

where $B_m$ denotes the set of alternatives in nest $m$. Given the inclusive values, we can show that the probability that random-utility–maximizing decision maker $i$ chooses alternative $j$ is

$$Pr_j = \frac{\exp\{V_j/\tau(j)\} \exp\{\tau(j)IV(j)\}}{\sum_m \exp(\tau_mIV_m)}$$
where $\tau(j)$ and $\text{IV}(j)$ are the dissimilarity parameter and inclusive value for the nest in which alternative $j$ lies.

In contrast, for the nonnormalized model, we have a latent variable

$$\tilde{V}_{i,j} = \tilde{\alpha}_j + x_{i,j}\tilde{\beta}_j + z_i\tilde{\gamma}_j$$

and corresponding inclusive values

$$\tilde{\text{IV}}_m = \ln \sum_{j \in B_m} \exp(\tilde{V}_k) \quad (2)$$

The probability of choosing alternative $j$ is

$$\Pr_j = \frac{\exp(\tilde{V}_j) \exp\{\tau(j)\tilde{\text{IV}}(j)\}}{\exp\{\tilde{\text{IV}}(j)\} \sum_m \exp(\tau_m \tilde{\text{IV}}_m)}$$

Equations (1) and (2) represent the key difference between the RUM-consistent and nonnormalized models. By scaling the $V_{ij}$ within each nest, the RUM-consistent model allows utilities to be compared across nests. Without the rescaling, utilities can be compared only for goods within the same nest. Moreover, adding a constant to each $V_{ij}$ for consumer $i$ will not affect the probabilities of the RUM-consistent model, but adding a constant to each $\tilde{V}_{ij}$ will affect the probabilities from the nonnormalized model. Decisions based on utility maximization can depend only on utility differences and not the scale or zero point of the utility function because utility is an ordinal concept, so the nonnormalized model cannot be consistent with utility maximization.

Heiss (2002) showed that the nonnormalized model can be RUM consistent in the special case where all the variables are specified in the bottom-level equation. Then multiplying the nonnormalized coefficients by the respective dissimilarity parameters results in the RUM-consistent coefficients.

**Technical note**

Degenerate nests occur when there is only one alternative in a branch of the tree hierarchy. The associated dissimilarity parameter of the RUM model is not defined. The inclusive-valued parameter of the nonnormalized model will be identifiable if there are alternative-specific variables specified in (1) of the model specification (the `indepvars` in the model syntax). Numerically, you can skirt the issue of nonidentifiable/undefined parameters by setting constraints on them. For the RUM model constraint, set the dissimilarity parameter to 1. See the description of `constraints()` in `Options` for details on setting constraints on the dissimilarity parameters.
nlogit stores the following in `e()`:

Scalars
- `e(N)`: number of observations
- `e(N_case)`: number of cases
- `e(k_eq)`: number of equations in `e(b)`
- `e(k_eq_model)`: number of equations in overall model test
- `e(k_alt)`: number of alternatives for bottom level
- `e(k_altj)`: number of alternatives for `j`th level
- `e(k_indvars)`: number of independent variables
- `e(k_ind2vars)`: number of by-alternative variables for bottom level
- `e(k_ind2varsj)`: number of by-alternative variables for `j`th level
- `e(df_m)`: model degrees of freedom
- `e(df_c)`: `clogit` model degrees of freedom
- `e(ll)`: log likelihood
- `e(ll_c)`: `clogit` model log likelihood
- `e(N_clust)`: number of clusters
- `e(chi2)`: \( \chi^2 \)
- `e(chi2_c)`: likelihood-ratio test for IIA
- `e(p)`: \( p \)-value for model Wald test
- `e(p_c)`: \( p \)-value for IIA test
- `e(i_base)`: base index for bottom level
- `e(i_basej)`: base index for `j`th level
- `e(levels)`: number of levels
- `e(alt_min)`: minimum number of alternatives
- `e(alt_avg)`: average number of alternatives
- `e(alt_max)`: maximum number of alternatives
- `e(const)`: constant indicator for bottom level
- `e(constj)`: constant indicator for `j`th level
- `e(rum)`: 1 if RUM model, 0 otherwise
- `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)` : `nllogit`
- `e(cmdline)` : command as typed
- `e(depvar)` : name of dependent variable
- `e(indvars)` : name of independent variables
- `e(ind2vars)` : by-alternative variables for bottom level
- `e(ind2vars_j)` : by-alternative variables for jth level
- `e(case)` : variable defining cases
- `e(altvar)` : alternative variable for bottom level
- `e(altvar_j)` : alternative variable for jth level
- `e(alteqs)` : equation names for bottom level
- `e(alteqs_j)` : equation names for jth level
- `e(alt_i)` : ith alternative for bottom level
- `e(alt_j_i)` : ith alternative for jth level
- `e(wtype)` : weight type
- `e(wexp)` : weight expression
- `e(title)` : title in estimation output
- `e(clustvar)` : name of cluster variable
- `e(chi2type)` : Wald, type of model χ² test
- `e(vce)` : vcetype specified in vce()
- `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(user)` : name of likelihood-evaluator program
- `e(technique)` : maximization technique
- `e(datasignature)` : the checksum
- `e(datasignaturevars)` : variables used in calculation of checksum
- `e(properties)` : `b V`
- `e(estat_cmd)` : program used to implement estat
- `e(predict)` : program used to implement predict
- `e(marginsnotok)` : predictions disallowed by 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(k_altern)` : number of alternatives at each level
- `e(k_branch)` : number of branches at each alternative of jth level
- `e(stats)` : alternative statistics for bottom level
- `e(stats_j)` : alternative statistics for jth level
- `e(altidx)` : alternative indices for jth level
- `e(alt_ind2vars)` : indicators for bottom level estimated by-alternative variable—`e(k_alt) × e(k_ind2vars)`
- `e(alt_ind2vars_j)` : indicators for jth level estimated by-alternative variable—`e(k_alt_j) × e(k_ind2vars_j)`
- `e(i(log))` : 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

## Methods and formulas

Methods and formulas are presented under the following headings:

- **Two-level nested logit model**
- **Three-level nested logit model**
Two-level nested logit model

Consider our two-level nested logit model for restaurant choice. We define \( T = \{1, 2, 3\} \) to be the set of indices denoting the three restaurant types and \( R_1 = \{1, 2\}, R_2 = \{3, 4, 5\}, \) and \( R_3 = \{6, 7\} \) to be the set of indices representing each restaurant within type \( t \in T \). Let \( C_1 \) and \( C_2 \) be the random variables that represent the choices made for the first level, restaurant type, and second level, restaurant, of the hierarchy, where we observe the choices \( C_1 = t, t \in T, \) and \( C_2 = j, j \in R_t \). Let \( z_t \) and \( x_{tj} \), for \( t \in T \) and \( j \in R_t \), refer to the row vectors of explanatory variables for the first-level alternatives and bottom-level alternatives for one case, respectively. We write the utilities (latent variables) as \( U_{tj} = z_t \alpha_t + x_{tj} \beta_j + \epsilon_{tj} = \eta_{tj} + \epsilon_{tj} \), where \( \alpha_t \) and \( \beta_j \) are column vectors and the \( \epsilon_{tj} \) are random disturbances. When the \( x_{tj} \) are alternative specific, we can drop the indices from \( \beta \), where we estimate one coefficient for each alternative in \( R_t \), \( t \in T \). These variables are specified in the first equation of the \texttt{nlogit} syntax (see example 3).

When the random-utility framework is used to describe the choice behavior, the alternative that is chosen is the alternative that has the highest utility. Assume for our restaurant example that we choose \( \epsilon \) where it has been shown that the \( \epsilon \) reduces to the multinomial logistic function.

For the logistic function in (4), we scale the linear predictors by the dissimilarity parameters. Another formulation of the conditional probability of choosing alternative \( j \in R_t \) given choice \( t \in T \) is the logistic function without this normalization:
\[
\Pr(C_2 = j \mid C_1 = t) = \frac{\exp(x_{tj}\beta_j)}{\sum_{m \in R_t} \exp(x_{tm}\beta_m)}
\]

and this is what is used in nlogit’s nonnormalized parameterization.

Amemiya (1985) defines the general form for the joint distribution of the \( \epsilon \)'s as

\[
F_{T,R}(\epsilon) = \exp\left\{ -\sum_{k \in T} \theta_k \left( \sum_{m \in R_k} \exp(-\epsilon_{km}/\tau_k) \right)^{\tau_k} \right\}
\]

from which the probability of choice \( t, t \in T \) can be derived as

\[
\Pr(C_1 = t) = \frac{\theta_t \left\{ \sum_{m \in R_t} \exp(\eta_{tm}/\tau_t) \right\}^{\tau_t}}{\sum_{k \in T} \theta_k \left\{ \sum_{m \in R_k} \exp(\eta_{km}/\tau_k) \right\}^{\tau_k}}
\]

(5)

nlogit sets \( \theta_t = 1 \). Noting that

\[
\left\{ \sum_{m \in R_t} \exp(\eta_{tm}/\tau_t) \right\}^{\tau_t} = \left\{ \sum_{m \in R_t} \exp\left( \frac{z_t \alpha_t + x_{tm}\beta_m}{\tau_t} \right) \right\}^{\tau_t}
\]

\[
= \exp(z_t \alpha_t) \left\{ \sum_{m \in R_t} \exp\left( \frac{x_{tm}\beta_m}{\tau_t} \right) \right\}^{\tau_t}
\]

\[
= \exp(z_t \alpha_t + \tau_t I_t)
\]

we define the inclusive values \( I_t \) as

\[
I_t = \ln\left\{ \sum_{m \in R_t} \exp(x_{tm}\beta_m/\tau_t) \right\}
\]

and we can view

\[
\exp(\tau_t I_t) = \left\{ \sum_{m \in R_t} \exp(x_{tm}\beta_m)^{1/\tau_t} \right\}^{\tau_t}
\]

as a weighted average of the \( \exp(x_{tm}\beta_m) \), for \( m \in R_t \). For the nlogit RUM parameterization, we can express (5) as

\[
\Pr(C_1 = t) = \frac{\exp(z_t \alpha_t + \tau_t I_t)}{\sum_{k \in T} \exp(z_k \alpha_k + \tau_k I_k)}
\]

Next we define inclusive values for the nonnormalized model to be

\[
\tilde{I}_t = \ln\left\{ \sum_{m \in R_t} \exp(x_{tm}\beta_m) \right\}
\]

and we express \( \Pr(C_1 = t) \) as

\[
\Pr(C_1 = t) = \frac{\exp(z_t \alpha_t + \tau_t \tilde{I}_t)}{\sum_{k \in T} \exp(z_k \alpha_k + \tau_k \tilde{I}_k)}
\]

(6)
Equation (5) is consistent with (6) only when \( \eta_{ij} = x_{ij} \beta_j \), so in general the \text{nlogit} nonnormalized model is not consistent with the RUM model.

Now assume that we have \( N \) cases where we add a third subscript, \( i \), to denote case \( i, i = 1, \ldots, N \). Denote \( y_{itj} \) to be a binary variable indicating the choice made by case \( i \) so that for each \( i \) only one \( y_{itj} \) is 1 and the rest are 0 for all \( t \in T \) and \( j \in R_t \). The log likelihood for the two-level RUM-consistent model is

\[
\log \ell = \sum_{i=1}^{N} \sum_{k \in T} \sum_{m \in R_k} y_{itj} \log \left\{ \Pr(C_{i1} = k) \Pr(C_{i2} = m | C_{i1} = k) \right\} \\
= \sum_{i=1}^{N} \sum_{k \in T} \sum_{m \in R_k} y_{itj} \left[ z_{ik} \alpha_k + \tau_k I_{ik} - \log \left\{ \sum_{l \in T} \exp(z_{il} \alpha_l + \tau_l I_{il}) \right\} \right] + \frac{x_{itj} \beta_m}{\tau_k} - \log \left\{ \sum_{l \in R_k} \exp(x_{ikl} \beta_l / \tau_k) \right\}
\]

The likelihood for the nonnormalized model has a similar form, replacing \( I \) with \( \tilde{I} \) and by not scaling \( x_{ikj} \beta_j \) by \( \tau_k \).

### Three-level nested logit model

Here we define a three-level nested logit model that can be generalized to the four-level and higher models. As before, let the integer set \( T \) be the indices for the first level of choices. Let sets \( S_t, t \in T \), be mutually exclusive sets of integers representing the choices of the second level of the hierarchy. Finally, let \( R_j, j \in S_t \), be the bottom-level choices. Let \( U_{tjk} = \eta_{tjk} + \epsilon_{tjk}, k \in R_j \), and the distribution of \( \epsilon_{tjk} \) be Gumbel’s multivariate extreme value of the form

\[
F(\epsilon) = \exp \left( -\sum_{t \in T} \left[ \sum_{j \in S_t} \left\{ \sum_{k \in R_j} \exp(-\eta_{tjk} / \tau_j) \right\}^{\tau_j / \upsilon_{tj}} \right]^{\upsilon_{tj}} \right)
\]

Let \( C_1, C_2, \) and \( C_3 \) represent the choice random variables for levels 1, 2, and the bottom, respectively. Then the set of conditional probabilities is

\[
\Pr(C_3 = k | C_1 = t, C_2 = j) = \frac{\exp(\eta_{tjk} / \tau_j)}{\sum_{l \in R_j} \exp(\eta_{tlk} / \tau_l)}
\]

\[
\Pr(C_2 = j | C_1 = t) = \frac{\left\{ \sum_{k \in R_j} \exp(\eta_{tjk} / \tau_j) \right\}^{\tau_j / \upsilon_{tj}}}{\sum_{l \in S_t} \left\{ \sum_{k \in R_l} \exp(\eta_{tlk} / \tau_l) \right\}^{\tau_l / \upsilon_{tl}}}
\]

\[
\Pr(C_1 = t) = \frac{\left[ \sum_{j \in S_t} \left\{ \sum_{k \in R_j} \exp(\eta_{tjk} / \tau_j) \right\}^{\tau_j / \upsilon_{tj}} \right]^{\upsilon_{tj}}}{\sum_{l \in T} \left[ \sum_{j \in S_l} \left\{ \sum_{k \in R_j} \exp(\eta_{tjk} / \tau_j) \right\}^{\tau_j / \upsilon_{tl}} \right]^{\upsilon_{tl}}}
\]
Assume that we can decompose the linear predictor as \( \eta_{tjk} = z_{it} \alpha_t + u_{tj} \gamma_j + x_{tjk} \beta_k \). Here \( z_{it} \), \( u_{tj} \), and \( x_{tjk} \) are the row vectors of explanatory variables for the first, second, and bottom levels of the hierarchy, respectively, and \( \alpha_t \), \( \gamma_j \), and \( \beta_k \) are the corresponding column vectors of regression coefficients for \( t \in T \), \( j \in S_t \), and \( k \in R_j \). We then can define the inclusive values for the first and second levels as

\[
I_{tj} = \log \sum_{k \in R_j} \exp(x_{tjk} \beta_k / \tau_j)
\]

\[
J_t = \log \sum_{j \in S_t} \exp(u_{tj} \gamma_j / v_t + \tau_j I_{tj})
\]

and rewrite the probabilities

\[
Pr(C_3 = k \mid C_1 = t, C_2 = j) = \frac{\exp(x_{tjk} \beta_k / \tau_j)}{\sum_{l \in R_j} \exp(x_{tlj} \beta_l / \tau_j)}
\]

\[
Pr(C_2 = j \mid C_1 = t) = \frac{\exp(u_{tj} \gamma_j / v_t + \tau_j I_{tj})}{\sum_{l \in S_t} \exp(u_{tlj} \gamma_l / v_t + \tau_l I_{tl})}
\]

\[
Pr(C_1 = t) = \frac{\exp(z_{it} \alpha_t + v_t I_t)}{\sum_{l \in T} \exp(z_{il} \alpha_l + v_l I_l)}
\]

We add a fourth index, \( i \), for case and define the indicator variable \( y_{itjk} \), \( i = 1, \ldots, N \), to indicate the choice made by case \( i \), \( t \in T \), \( j \in S_t \), and \( k \in R_j \). The log likelihood for the \texttt{nlogit} RUM-consistent model is

\[
\ell = \sum_{i=1}^{N} \sum_{t \in T} \sum_{j \in S_t} \sum_{k \in R_j} y_{itjk} \left\{ z_{it} \alpha_t + v_t J_{it} - \log \left( \sum_{m \in T} z_{im} \alpha_m + v_m J_{im} \right) + u_{itj} \gamma_j / v_t + \tau_j I_{tj} - \log \left( \sum_{m \in S_t} u_{itm} \gamma_m / v_t + \tau_m I_{itm} \right) + x_{itjk} \beta_k / \tau_k - \sum_{m \in R_t} \exp(x_{itjm} \beta_m / \tau_k) \right\}
\]

and for the nonnormalized \texttt{nlogit} model the log likelihood is

\[
\ell = \sum_{i=1}^{N} \sum_{t \in T} \sum_{j \in S_t} \sum_{k \in R_j} y_{itjk} \left\{ z_{it} \alpha_t + v_t J_{it} - \log \left( \sum_{m \in T} z_{im} \alpha_m + v_m J_{im} \right) + u_{itj} \gamma_j + \tau_j I_{tj} - \log \left( \sum_{m \in S_t} u_{itm} \gamma_m + \tau_m I_{itm} \right) + x_{itjk} \beta_k - \sum_{m \in R_t} \exp(x_{itjm} \beta_m) \right\}
\]

Extending the model to more than three levels is straightforward, albeit notationally cumbersome.

This command supports the Huber/White/sandwich estimator of the variance and its clustered version using \texttt{vce(robust)} and \texttt{vce(cluster clustvar)}, respectively. See \texttt{[P] _robust}, particularly \textit{Maximum likelihood estimators} and \textit{Methods and formulas}. 

\texttt{nlogit} — Nested logit regression

23
References


Also see

[R] nlogit postestimation — Postestimation tools for nlogit

[R] asclogit — Alternative-specific conditional logit (McFadden’s choice) model

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

[R] mlogit — Multinomial (polytomous) logistic regression

[R] ologit — Ordered logistic regression

[R] rologit — Rank-ordered logistic regression

[R] slogit — Stereotype logistic regression

[U] 20 Estimation and postestimation commands