Title

stata.com

nlogit — Nested logit regression

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 a random utility model (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 nested logit tree before calling `nlogit`.

Quick start

Create variables identifying alternatives at higher levels

For a three-level nesting structure, create `l2alt` identifying four alternatives at level two based on the variable `balt`, which identifies eight bottom-level alternatives

```
nlogitgen l2alt = balt(l2alt1: balt1 | balt2, l2alt2: balt3 | balt4,  ///
    l2alt3: balt5 | balt6, l2alt4: balt7 | balt8)
```

Same as above, defining `l2alt` from values of `balt` rather than from value labels

```
nlogitgen l2alt = balt(l2alt1: 1|2, l2alt2: 3|4, l2alt3: 5|6,  ///
    l2alt4: 7|8)
```

Create `talt` identifying top-level alternatives based on the four alternatives in `l2alt`

```
nlogitgen talt = l2alt(talt1: l2alt1 | l2alt2, talt2: l2alt3 | l2alt4)
```

Examine tree structure

Display three-level nesting structure

```
nlogittree balt l2alt talt
```

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

```
nlogittree balt l2alt talt, choice(chosen)
```

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

```
nlogittree balt l2alt talt, choice(chosen) case(casevar)
```

As above, and generate prob indicating problematic observations

```
nlogittree balt l2alt talt, choice(chosen) case(casevar) generate(prob)
```

Fit nested logit model

Three-level model with alternative-specific covariate x1, case-specific covariate x2 modeling top-level choice, and case-specific covariate x3 modeling level-two choice

```
nlogit chosen x1 || talt: x2 || l2alt: x3 || balt:, case(casevar)
```

As above, but do not estimate intercepts for bottom-level alternatives

```
nlogit chosen x1 || talt: x2 || l2alt: x3 || balt:, noconstant  ///
     case(casevar)
```

As above, but estimate intercepts for top-level alternatives

```
nlogit chosen x1 || talt: x2, estconst || l2alt: x3  ///
     || balt:, noconstant case(casevar)
```

Specify the base alternative at each level

```
nlogit chosen x1 || talt: x2, base(talt2) || l2alt: x3, base(l2alt4)  ///
     || balt:, base(balt8) case(casevar)
```

Use nonnormalized parameterization

```
nlogit chosen x1 || talt: x2 || l2alt: x3 || balt:, case(casevar)  ///
     nonnormalized
```

Menu

nlogit

Statistics > Choice models > Nested logit model > Nested logit model

nlogitgen

Statistics > Choice models > Nested logit model > Setup for nested logit model

nlogittree

Statistics > Choice models > Nested logit model > Display nested logit tree structure

Syntax

Nested logit regression

```
nlogit depvar [indepvars] [if] [in] [weight] [|| lev1_equation
    [|| lev2_equation ...]] || altvar: [byaltvarlist], case(varname) [nlogit_options]
```

where the syntax of *lev#_equation* is

```
altvar: [byaltvarlist] [, base(# | lbl) estconst]
```

Create variable based on specification of branches

```
nlogitgen newaltvar = altvar (branchlist) [, [no]log]
```

where *branchlist* is

```
branch, branch [, branch ...]
```

and *branch* is

```
[label:] alternative [| alternative [| alternative ...]]
```

Display tree structure

```
nlogittree altvarlist [if] [in] [weight] [, nlogittree_options]
```

nlogit_options	Description
Model	
* `case(`*varname*`)`	use *varname* to identify cases
`base(`*#* | *lbl*`)`	use the specified level or label of *altvar* as the base alternative for the bottom level
`noconstant`	suppress the constant terms for the bottom-level alternatives
`nonnormalized`	use the nonnormalized parameterization
`altwise`	use alternativewise deletion instead of casewise deletion
`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)`
`notree`	suppress display of tree-structure output; see also `nolabel` and `nobranches`
`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
Maximization	
maximize_options	control the maximization process; seldom used
`collinear`	keep collinear variables

* `case(`*varname*`)` is required.

nlogittree_options	Description
Main	
`choice(`*depvar*`)`	use *depvar* as the choice indicator variable
`case(`*varname*`)`	use *varname* to identify cases
`generate(`*newvar*`)`	create *newvar* to identify invalid observations
`nolabel`	suppress the value labels in tree-structure output
`nobranches`	suppress drawing branches in the tree-structure output

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.

`collinear` does not appear in the dialog box.

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

`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)
```

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`, `noomitted`, `vsquish`, `noemptycells`, `baselevels`, `allbaselevels`, `nofvlabel`, `fvwrap(#)`, `fvwrapon(`*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. The default optimization technique is `technique(bfgs)`.

The following option is available with `nlogit` but is not shown in the dialog box:

`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.

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)
```

`log` and `nolog` specify whether to display the iteration log. The iteration log is displayed by default unless you used `set iterlog off` to suppress it; see `set iterlog` in [R] ***set iter***.

Specification and options for nlogittree

Main

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

stata.com

Remarks are presented under the following headings:

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. The nested logit model is a direct generalization of McFadden's choice model fit by `cmclogit`. You may want to read [CM] **Intro 5**, [CM] **Intro 8**, and [CM] **cmclogit** before continuing.

Although `nlogit` fits choice models, it is not a `cm` command, and you do not have to `cmset` your data. `nlogit` has its own data requirements for nested alternatives, which are detailed in the following examples.

By default, `nlogit` uses a RUM parameterization. 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 parameterization 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 (2015) 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: Families choosing a restaurant

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 Norteños, 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 https://www.stata-press.com/data/r16/restaurant
. describe
Contains data from https://www.stata-press.com/data/r16/restaurant.dta
  obs:         2,100
 vars:             8                          2 Dec 2018 15:07
-------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     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
```

```
. list family_id restaurant chosen kids rating distance in 1/21, sepby(fam)
> abbrev(10)
```

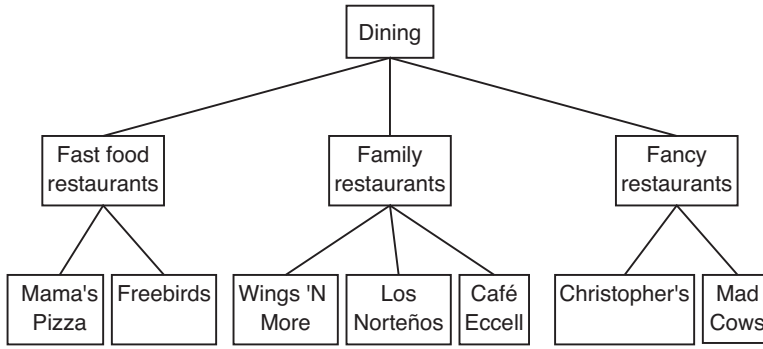
	family_id	restaurant	chosen	kids	rating	distance
1.	1	Freebirds	1	1	0	1.245553
2.	1	MamasPizza	0	1	1	2.82493
3.	1	CafeEccell	0	1	2	4.21293
4.	1	LosNortenos	0	1	3	4.167634
5.	1	WingsNmore	0	1	2	6.330531
6.	1	Christophers	0	1	4	10.19829
7.	1	MadCows	0	1	5	5.601388
8.	2	Freebirds	0	3	0	4.162657
9.	2	MamasPizza	0	3	1	2.865081
10.	2	CafeEccell	0	3	2	5.337799
11.	2	LosNortenos	1	3	3	4.282864
12.	2	WingsNmore	0	3	2	8.133914
13.	2	Christophers	0	3	4	8.664631
14.	2	MadCows	0	3	5	9.119597
15.	3	Freebirds	1	3	0	2.112586
16.	3	MamasPizza	0	3	1	2.215329
17.	3	CafeEccell	0	3	2	6.978715
18.	3	LosNortenos	0	3	3	5.117877
19.	3	WingsNmore	0	3	2	5.312941
20.	3	Christophers	0	3	4	9.551273
21.	3	MadCows	0	3	5	5.539806

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 `cmclogit` 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 or her attitude 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 nested logit 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 the 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 (2015, chap. 14). 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 nested logit tree.

▷ Example 2: Identifying the first-level set of alternatives

To run `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 `nlogitgen`.

```
. 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
 type    N      restaurant    N   k
──────────────────────────────────────
 fast   600 ─┬─ Freebirds    300  12
             └─ MamasPizza   300  15
 family 900 ─┬─ CafeEccell   300  78
             ├─ LosNortenos  300  75
             └─ WingsNmore   300  69
 fancy  600 ─┬─ Christophers 300  27
             └─ MadCows      300  24
──────────────────────────────────────
                    total  2100 300
k = number of times alternative is chosen
N = number of observations at each level
```

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

◁

❑ Technical note

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

```
. use https://www.stata-press.com/data/r16/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
```

```
. 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

```
      case 1                          case 2
 type       restaurant           type       restaurant
──────────────────────          ──────────────────────
 fast   ─┬─ Freebirds           fast   ─┬─ Freebirds
         └─ MamasPizza                  └─ MamasPizza
 family ─┬─ CafeEccell          family ─┬─ LosNortenos
         ├─ LosNortenos                 └─ WingsNmore
         └─ WingsNmore
 fancy  ─┬─ Christophers        fancy  ── Christophers
         └─ MadCows
```

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 Norteños, and Wings 'N More are family restaurants; and Christopher's and Mad Cows are fancy restaurants. `nlogit` requires only that hierarchy be maintained for all cases.

Estimation

▷ Example 3: Fitting the model

With our `type` variable created that defines the three types of restaurants, we can now examine how the alternative-specific attributes (`cost`, `rating`, and `distance`) apply to the bottom alternative set (the seven restaurants) and how family-specific attributes (`income` and `kid`) apply to the alternative set at the first decision level (the three types of restaurants).

```
. use https://www.stata-press.com/data/r16/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
 type     N      restaurant     N   k
-----------------------------------------
 fast   600 ---- Freebirds     300  12
            `--- MamasPizza    300  15
 family 900 ---- CafeEccell    300  78
            |--- LosNortenos   300  75
            `--- WingsNmore    300  69
 fancy  600 ---- Christophers  300  27
            `--- MadCows       300  24
-----------------------------------------
                      total   2100 300
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                     Prob > chi2        =     0.0000
------------------------------------------------------------------------------
      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   .0090936     5.08   0.000     .0283595    .0640059
        kids |  -.3959413   .1220356    -3.24   0.001    -.6351267   -.1567559
-------------+----------------------------------------------------------------
dissimilarity parameters
-------------+----------------------------------------------------------------
/type        |
    fast_tau |   1.712878    1.48685                     -1.201295    4.627051
  family_tau |   2.505113   .9646351                       .614463    4.395763
   fancy_tau |   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 must specify the alternative set of parameters as 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 random utility maximization; 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 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: Testing the IIA assumption

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	.584642
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	.0564644
kidFast	-.2398757	.1063674	-2.26	0.024	-.448352	-.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
note: 222 groups (888 obs) dropped because of all positive or
      all negative outcomes.
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	.067852	-0.91	0.363	-.1946496	.0713254
rating	.1659001	.2832041	0.59	0.558	-.3891698	.72097
distance	-.244396	.0995056	-2.46	0.014	-.4394234	-.0493687
incFast	-.0737506	.0177444	-4.16	0.000	-.108529	-.0389721
kidFast	.4105386	.2137051	1.92	0.055	-.0083157	.8293928

```
. hausman . fullset
```

	Coefficients			
	(b)	(B)	(b-B)	sqrt(diag(V_b-V_B))
	.	fullset	Difference	S.E.
cost	-.0616621	-.1367799	.0751178	.0576092
rating	.1659001	.3066622	-.1407621	.2451308
distance	-.244396	-.1977505	-.0466456	.0876173
incFast	-.0737506	-.0390183	-.0347323	.015049
kidFast	.4105386	-.2398757	.6504143	.1853533

```
                b = consistent under Ho and Ha; obtained from clogit
     B = inconsistent under Ha, efficient under Ho; obtained from clogit
Test:  Ho:  difference in coefficients not systematic
           chi2(5) = (b-B)'[(V_b-V_B)^(-1)](b-B)
                   =       10.70
         Prob>chi2 =      0.0577
         (V_b-V_B is not positive definite)
```

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 change 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 (2018, 837–839). Here we outline the differences between the nonnormalized model and the random utility parameterization of the model. Our discussion follows Heiss (2002) and assumes the nested logit 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

$$U_{ij} = V_{ij} + \epsilon_{ij} = \alpha_j + \mathbf{x}_{ij}\boldsymbol{\beta}_j + \mathbf{z}_i\boldsymbol{\gamma}_j + \epsilon_{ij}$$

where V_{ij} is the deterministic part of utility and ϵ_{ij} is the random part. $\mathbf{x}_{ij}$ are alternative-specific variables and $\mathbf{z}_i$ 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 ρ_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 mth 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,

$$\text{IV}_m = \ln \sum_{j \in B_m} \exp(V_k/\tau_m) \tag{1}$$

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

$$\Pr_j = \frac{\exp\{V_j/\tau(j)\}}{\exp\{\text{IV}(j)\}} \frac{\exp\{\tau(j)\text{IV}(j)\}}{\sum_m \exp(\tau_m \text{IV}_m)}$$

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

$$\widetilde{V}_{i,j} = \widetilde{\alpha}_j + \mathbf{x}_{i,j}\widetilde{\boldsymbol{\beta}}_j + \mathbf{z}_i\widetilde{\boldsymbol{\gamma}}_j$$

and corresponding inclusive values

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

The probability of choosing alternative j is

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

Equations (1) and (2) represent the key difference between the random utility and the nonnormalized models. By scaling the V_{ij} within each nest, the RUM parameterization 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, but adding a constant to each $\widetilde{V}_{ij}$ 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 consistent with a random utility parameterization 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 coefficients that are consistent with a RUM.

❑ 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 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 constraint, set the dissimilarity parameter to 1. See the description of `constraints()` in *Options* for details on setting constraints on the dissimilarity parameters.

❑

Stored results

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

Scalars

`e(N)`	number of observations
`e(N_case)`	number of cases
`e(N_ic)`	N for Bayesian information criterion (BIC)
`e(N_clust)`	number of clusters
`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_alt`*j*`)`	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_ind2vars`*j*`)`	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(chi2)`	χ^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_base`*j*`)`	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(const`*j*`)`	constant indicator for *j*th level
`e(rum)`	`1` if RUM, `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)`	`nlogit`
`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 *j*th level
`e(case)`	variable defining cases
`e(altvar)`	alternative variable for bottom level
`e(altvar`*j*`)`	alternative variable for *j*th level
`e(alteqs)`	equation names for bottom level
`e(alteqs`*j*`)`	equation names for *j*th level
`e(alt`*i*`)`	*i*th alternative for bottom level
`e(alt`*j*`_`*i*`)`	*i*th alternative for *j*th level
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(marktype)`	`casewise` or `altwise`, type of markout
`e(key_N_ic)`	`cases`, key for N for Bayesian information criterion (BIC)
`e(title)`	title in estimation output
`e(clustvar)`	name of cluster variable
`e(chi2type)`	`Wald`, type of model χ^2 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*j*)	number of branches at each alternative of *j*th level
e(stats)	alternative statistics for bottom level
e(stats*j*)	alternative statistics for *j*th level
e(altidx*j*)	alternative indices for *j*th level
e(alt_ind2vars)	indicators for bottom level estimated by-alternative variable—e(k_alt)×e(k_ind2vars)
e(alt_ind2vars*j*)	indicators for *j*th level estimated by-alternative variable—e(k_alt*j*)×e(k_ind2vars*j*)
e(ilog)	iteration log (up to 20 iterations)
e(gradient)	gradient vector
e(V)	variance–covariance matrix of the estimators
e(V_modelbased)	model-based variance
Functions	
e(sample)	marks estimation sample

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

Matrices	
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

Methods and formulas are presented under the following headings:

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 $\mathbf{z}_t$ and $\mathbf{x}_{tj}$, 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 $U_{tj} = \mathbf{z}_t\boldsymbol{\alpha}_t + \mathbf{x}_{tj}\boldsymbol{\beta}_j + \epsilon_{tj} = \boldsymbol{\eta}_{tj} + \epsilon_{tj}$, where $\boldsymbol{\alpha}_t$ and $\boldsymbol{\beta}_j$ are column vectors and the ϵ_{tj} are random disturbances. When the $\mathbf{x}_{tj}$ are alternative specific, we can drop the indices from $\boldsymbol{\beta}$, where we estimate one coefficient for each alternative in R_t, $t \in T$. These variables are specified in the first equation of the 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 restaurant type $t \in T$. For the RUM parameterization of `nlogit`, the conditional distribution of ϵ_{tj} given choice of restaurant type t is a multivariate version of Gumbel's extreme-value distribution,

$$F_{R|T}(\boldsymbol{\epsilon} \,|\, t) = \exp\left[-\left\{\sum_{m \in R_t} \exp(\epsilon_{tm}/\tau_t)\right\}^{\tau_t}\right] \tag{3}$$

where it has been shown that the ϵ_{tj}, $j \in R_t$, are exchangeable with correlation $1-\tau_t^2$, for $\tau_t \in (0,1]$ (Kotz and Nadarajah 2000). For example, the probability of choosing Christopher's, $j = 6$ given type $t = 3$, is

$$\begin{aligned}
\Pr(C_2 = 6 \,|\, C_1 = 3) &= \Pr\left(U_{36} - U_{37} > 0\right) \\
&= \Pr\left(\epsilon_{37} \le \epsilon_{36} + \eta_{36} - \eta_{37}\right) \\
&= \int_{-\infty}^{\infty} \left\{ \int_{-\infty}^{\epsilon_{36}+\eta_{36}-\eta_{37}} f_{R|T}\left(\epsilon_{36}, \epsilon_{37}\right) d\epsilon_{37} \right\} d\epsilon_{36}
\end{aligned}$$

where $f = \dfrac{\partial F}{\partial \epsilon_{36} \partial \epsilon_{37}}$ is the joint density function of $\boldsymbol{\epsilon}$ given t. U_{37} is the utility of eating at Mad Cows, the other fancy ($t = 3$) restaurant. Amemiya (1985) demonstrates that this integral evaluates to the logistic function

$$\begin{aligned}
\Pr(C_2 = 6 \,|\, C_1 = 3) &= \frac{\exp(\boldsymbol{\eta}_{36}/\tau_3)}{\exp(\boldsymbol{\eta}_{36}/\tau_3) + \exp(\boldsymbol{\eta}_{37}/\tau_3)} \\
&= \frac{\exp(\mathbf{x}_{36}\boldsymbol{\beta}_6/\tau_3)}{\exp(\mathbf{x}_{36}\boldsymbol{\beta}_6/\tau_3) + \exp(\mathbf{x}_{37}\boldsymbol{\beta}_7/\tau_3)}
\end{aligned}$$

and in general

$$\Pr(C_2 = j \,|\, C_1 = t) = \frac{\exp(\mathbf{x}_{tj}\boldsymbol{\beta}_j/\tau_t)}{\sum_{m \in R_t} \exp(\mathbf{x}_{tm}\boldsymbol{\beta}_m/\tau_t)} \tag{4}$$

Letting $\tau_t = 1$ in (3) reduces to the product of independent extreme-value distributions, and (4) 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,

$$\Pr(C_2 = j \,|\, C_1 = t) = \frac{\exp(\mathbf{x}_{tj}\boldsymbol{\beta}_j)}{\sum_{m \in R_t} \exp(\mathbf{x}_{tm}\boldsymbol{\beta}_m)}$$

and this is what is used in `nlogit`'s nonnormalized parameterization.

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

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

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

$$\Pr(C_1 = t) = \frac{\theta_t\left\{\sum_{m\in R_t}\exp(\boldsymbol{\eta}_{tm}/\tau_t)\right\}^{\tau_t}}{\sum_{k\in T}\theta_k\left\{\sum_{m\in R_k}\exp(\boldsymbol{\eta}_{km}/\tau_k)\right\}^{\tau_k}} \tag{5}$$

`nlogit` sets $\theta_t = 1$. Noting that

$$\begin{aligned}\left\{\sum_{m\in R_t}\exp(\boldsymbol{\eta}_{tm}/\tau_t)\right\}^{\tau_t} &= \left\{\sum_{m\in R_t}\exp\left(\frac{\mathbf{z}_t\boldsymbol{\alpha}_t + \mathbf{x}_{tm}\boldsymbol{\beta}_m}{\tau_t}\right)\right\}^{\tau_t}\\ &= \exp(\mathbf{z}_t\boldsymbol{\alpha}_t)\left\{\sum_{m\in R_t}\exp\left(\mathbf{x}_{tm}\boldsymbol{\beta}_m/\tau_t\right)\right\}^{\tau_t}\\ &= \exp(\mathbf{z}_t\boldsymbol{\alpha}_t + \tau_t I_t)\end{aligned}$$

we define the inclusive values I_t as

$$I_t = \ln\left\{\sum_{m\in R_t}\exp(\mathbf{x}_{tm}\boldsymbol{\beta}_m/\tau_t)\right\}$$

and we can view

$$\exp(\tau_t I_t) = \left\{\sum_{m\in R_t}\exp(x_{tm}\boldsymbol{\beta}_m)^{1/\tau_t}\right\}^{\tau_t}$$

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

$$\Pr(C_1 = t) = \frac{\exp(\mathbf{z}_t\boldsymbol{\alpha}_t + \tau_t I_t)}{\sum_{k\in T}\exp(\mathbf{z}_k\boldsymbol{\alpha}_k + \tau_k I_k)}$$

Next, we define inclusive values for the nonnormalized model to be

$$\widetilde{I}_t = \ln\left\{\sum_{m\in R_t}\exp(\mathbf{x}_{tm}\boldsymbol{\beta}_m)\right\}$$

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

$$\Pr(C_1 = t) = \frac{\exp(\mathbf{z}_t\boldsymbol{\alpha}_t + \tau_t\widetilde{I}_t)}{\sum_{k\in T}\exp(\mathbf{z}_k\boldsymbol{\alpha}_k + \tau_k\widetilde{I}_k)} \tag{6}$$

Equation (5) is consistent with (6) only when $\boldsymbol{\eta}_{ij} = \mathbf{x}_{ij}\boldsymbol{\beta}_j$, so in general the `nlogit` nonnormalized model is not consistent with the RUM.

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 is

$$\begin{aligned}\log\ell &= \sum_{i=1}^{N}\sum_{k\in T}\sum_{m\in R_k} y_{itj}\log\left\{\Pr(C_{i1}=k)\Pr(C_{i2}=m|C_{i1}=k)\right\}\\ &= \sum_{i=1}^{N}\sum_{k\in T}\sum_{m\in R_k} y_{itj}\left[\mathbf{z}_{ik}\boldsymbol{\alpha}_k + \tau_k I_{ik} - \log\left\{\sum_{l\in T}\exp(\mathbf{z}_{il}\boldsymbol{\alpha}_l + \tau_l I_{il})\right\} + \right.\\ &\qquad \left. \mathbf{x}_{itj}\boldsymbol{\beta}_m/\tau_k - \log\left\{\sum_{l\in R_k}\exp(\mathbf{x}_{ikl}\boldsymbol{\beta}_l/\tau_k)\right\}\right]\end{aligned}$$

The likelihood for the nonnormalized model has a similar form, replacing I with $\widetilde{I}$ and by not scaling $\mathbf{x}_{ikj}\boldsymbol{\beta}_j$ by τ_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 ϵ_{tjk} be Gumbel's multivariate extreme value of the form

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

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

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

$$\Pr(C_2 = j \,|\, C_1 = t) = \frac{\left\{\sum_{k\in R_j}\exp(\eta_{tjk}/\tau_j)\right\}^{\tau_j/\upsilon_t}}{\sum_{l\in S_t}\left\{\sum_{k\in R_l}\exp(\eta_{tlk}/\tau_l)\right\}^{\tau_l/\upsilon_t}}$$

$$\Pr(C_1 = t) = \frac{\left[\sum_{j\in S_t}\left\{\sum_{k\in R_j}\exp(\eta_{tjk}/\tau_j)\right\}^{\tau_j/\upsilon_t}\right]^{\upsilon_t}}{\sum_{l\in T}\left[\sum_{j\in S_l}\left\{\sum_{k\in R_j}\exp(\eta_{ljk}/\tau_j)\right\}^{\tau_j/\upsilon_l}\right]^{\upsilon_l}}$$

Assume that we can decompose the linear predictor as $\boldsymbol{\eta}_{tjk} = \mathbf{z}_t\boldsymbol{\alpha}_t + \mathbf{u}_{tj}\boldsymbol{\gamma}_j + \mathbf{x}_{tjk}\boldsymbol{\beta}_k$. Here $\mathbf{z}_t$, $\mathbf{u}_{tj}$, and $\mathbf{x}_{tjk}$ are the row vectors of explanatory variables for the first, second, and bottom levels of the hierarchy, respectively, and $\boldsymbol{\alpha}_t$, $\boldsymbol{\gamma}_j$, and $\boldsymbol{\beta}_k$ 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

$$I_{tj} = \log \sum_{k \in R_j} \exp(\mathbf{x}_{tjk}\boldsymbol{\beta}_k/\tau_j)$$

$$J_t = \log \sum_{j \in S_t} \exp(\mathbf{u}_{tj}\boldsymbol{\gamma}_j/\upsilon_t + \frac{\tau_j}{\upsilon_t} I_{tj})$$

and rewrite the probabilities

$$\Pr(C_3 = k \,|\, C_1 = t, C_2 = j) = \frac{\exp(\mathbf{x}_{tjk}\boldsymbol{\beta}_k/\tau_j)}{\sum_{l \in R_j} \exp(\mathbf{x}_{tjl}\boldsymbol{\beta}_l/\tau_j)}$$

$$\Pr(C_2 = j \,|\, C_1 = t) = \frac{\exp(\mathbf{u}_{tj}\boldsymbol{\gamma}_j/\upsilon_t + \frac{\tau_j}{\upsilon_t} I_{tj})}{\sum_{l \in S_t} \exp(\mathbf{u}_{tl}\boldsymbol{\gamma}_l/\upsilon_t + \frac{\tau_l}{\upsilon_t} I_{tl})}$$

$$\Pr(C_1 = t) = \frac{\exp(\mathbf{z}_t\boldsymbol{\alpha}_t + \upsilon_t J_t)}{\sum_{l \in T} \exp(\mathbf{z}_l\boldsymbol{\alpha}_l + \upsilon_l J_l)}$$

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 `nlogit` RUM is

$$\ell = \sum_{i=1}^{N} \sum_{t \in T} \sum_{j \in S_t} \sum_{k \in R_j} y_{itjk} \left\{ \mathbf{z}_{it}\boldsymbol{\alpha}_t + \upsilon_t J_{it} - \log \left(\sum_{m \in T} \mathbf{z}_{im}\boldsymbol{\alpha}_m + \upsilon_m J_{im} \right) + \right.$$
$$\mathbf{u}_{itj}\boldsymbol{\gamma}_j/\upsilon_t + \frac{\tau_j}{\upsilon_t} I_{itj} - \log \left(\sum_{m \in S_t} \mathbf{u}_{itm}\boldsymbol{\gamma}_m/\upsilon_t + \frac{\tau_m}{\upsilon_t} I_{itm} \right) +$$
$$\left. \mathbf{x}_{itjk}\boldsymbol{\beta}_k/\tau_k - \sum_{m \in R_t} \exp(\mathbf{x}_{itjm}\boldsymbol{\beta}_m/\tau_k) \right\}$$

and for the nonnormalized `nlogit` model, the log likelihood is

$$\ell = \sum_{i=1}^{N} \sum_{t \in T} \sum_{j \in S_t} \sum_{k \in R_j} y_{itjk} \left\{ \mathbf{z}_{it}\boldsymbol{\alpha}_t + \upsilon_t J_{it} - \log \left(\sum_{m \in T} \mathbf{z}_{im}\boldsymbol{\alpha}_m + \upsilon_m J_{im} \right) + \right.$$
$$\mathbf{u}_{itj}\boldsymbol{\gamma}_j + \tau_j I_{itj} - \log \left(\sum_{m \in S_t} \mathbf{u}_{itm}\boldsymbol{\gamma}_m + \tau_m I_{itm} \right) +$$
$$\left. \mathbf{x}_{itjk}\boldsymbol{\beta}_k - \sum_{m \in R_t} \exp(\mathbf{x}_{itjm}\boldsymbol{\beta}_m) \right\}$$

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 `vce(robust)` and `vce(cluster` *clustvar*`)`, respectively. See [P] **_robust**, particularly *Maximum likelihood estimators* and *Methods and formulas*.

References

Amemiya, T. 1985. *Advanced Econometrics*. Cambridge, MA: Harvard University Press.

Greene, W. H. 2018. *Econometric Analysis*. 8th ed. New York: Pearson.

Hausman, J. A. 1978. Specification tests in econometrics. *Econometrica* 46: 1251–1271.

Hausman, J. A., and D. L. McFadden. 1984. Specification tests for the multinomial logit model. *Econometrica* 52: 1219–1240.

Heiss, F. 2002. Structural choice analysis with nested logit models. *Stata Journal* 2: 227–252.

Hensher, D. A., J. M. Rose, and W. H. Greene. 2005. *Applied Choice Analysis: A Primer*. New York: Cambridge University Press.

——. 2015. *Applied Choice Analysis*. 2nd ed. Cambridge: Cambridge University Press.

Kotz, S., and S. Nadarajah. 2000. *Extreme Value Distributions: Theory and Applications*. London: Imperial College Press.

Maddala, G. S. 1983. *Limited-Dependent and Qualitative Variables in Econometrics*. Cambridge: Cambridge University Press.

McFadden, D. L. 1977. Quantitative methods for analyzing travel behaviour of individuals: Some recent developments. Working paper 474, Cowles Foundation. http://cowles.yale.edu/cfdp-474.

——. 1981. Econometric models of probabilistic choice. In *Structural Analysis of Discrete Data with Econometric Applications*, ed. C. F. Manski and D. L. McFadden, 198–272. Cambridge, MA: MIT Press.

Also see

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

[CM] **cmclogit** — Conditional logit (McFadden's) choice model

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

[CM] **cmmprobit** — Multinomial probit choice model

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

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

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

[R] **ologit** — Ordered logistic regression

[R] **slogit** — Stereotype logistic regression

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