Title

mlogit — Multinomial (polytomous) logistic regression

Syntax

`mlogit` *depvar* [*indepvars*] [*if*] [*in*] [*weight*] [, *options*]

options	Description
Model	
`noconstant`	suppress constant term
`baseoutcome(#)`	value of *depvar* that will be the base outcome
`constraints(`*clist*`)`	apply specified linear constraints; *clist* has the form #[-#] [, #[-#] ...]
`collinear`	keep collinear variables
SE/Robust	
`vce(`*vcetype*`)`	*vcetype* may be `oim`, `robust`, `cluster` *clustvar*, `bootstrap`, or `jackknife`
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`rrr`	report relative-risk ratios
`nocnsreport`	do not display constraints
display_options	control 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
`coeflegend`	display legend instead of statistics

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

indepvars may contain time-series operators; see **[U] 11.4.4 Time-series varlists**.

`bootstrap`, `by`, `fp`, `jackknife`, `mfp`, `mi estimate`, `rolling`, `statsby`, and `svy` are allowed; see **[U] 11.1.10 Prefix commands**.

`vce(bootstrap)` and `vce(jackknife)` are not allowed with the `mi estimate` prefix; see **[MI] mi estimate**.

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

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

`fweight`s, `iweight`s, and `pweight`s are allowed; see **[U] 11.1.6 weight**.

`coeflegend` does not appear in the dialog box.

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

Menu

Statistics > Categorical outcomes > Multinomial logistic regression

Description

`mlogit` fits maximum-likelihood multinomial logit models, also known as polytomous logistic regression. You can define constraints to perform constrained estimation. Some people refer to conditional logistic regression as multinomial logit. If you are one of them, see [R] **clogit**.

See [R] **logistic** for a list of related estimation commands.

Options

Model

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

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

`constraints(`*clist*`)`, `collinear`; see [R] **estimation options**.

SE/Robust

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

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

Reporting

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

`rrr` reports the estimated coefficients transformed to relative-risk ratios, that is, e^b rather than b; see *Description of the model* below for an explanation of this concept. Standard errors and confidence intervals are similarly transformed. This option affects how results are displayed, not how they are estimated. `rrr` may be specified at estimation or when replaying previously estimated results.

`nocnsreport`; see [R] **estimation options**.

display_options: `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 following option is available with `mlogit` but is not shown in the dialog box:

`coeflegend`; see [R] **estimation options**.

Remarks and examples

Remarks are presented under the following headings:

`mlogit` fits maximum likelihood models with discrete dependent (left-hand-side) variables when the dependent variable takes on more than two outcomes and the outcomes have no natural ordering. If the dependent variable takes on only two outcomes, estimates are identical to those produced by `logistic` or `logit`; see [R] **logistic** or [R] **logit**. If the outcomes are ordered, see [R] **ologit**.

Description of the model

For an introduction to multinomial logit models, see Greene (2012, 763–766), Hosmer, Lemeshow, and Sturdivant (2013, 269–289), Long (1997, chap. 6), Long and Freese (2014, chap. 8), and Treiman (2009, 336–341). For a description emphasizing the difference in assumptions and data requirements for conditional and multinomial logit, see Davidson and MacKinnon (1993).

Consider the outcomes 1, 2, 3, $\ldots$, m recorded in y, and the explanatory variables X. Assume that there are $m = 3$ outcomes: "buy an American car", "buy a Japanese car", and "buy a European car". The values of y are then said to be "unordered". Even though the outcomes are coded 1, 2, and 3, the numerical values are arbitrary because $1 < 2 < 3$ does not imply that outcome 1 (buy American) is less than outcome 2 (buy Japanese) is less than outcome 3 (buy European). This unordered categorical property of y distinguishes the use of `mlogit` from `regress` (which is appropriate for a continuous dependent variable), from `ologit` (which is appropriate for ordered categorical data), and from `logit` (which is appropriate for two outcomes, which can be thought of as ordered).

In the multinomial logit model, you estimate a set of coefficients, $\beta^{(1)}$, $\beta^{(2)}$, and $\beta^{(3)}$, corresponding to each outcome:

$$\Pr(y=1) = \frac{e^{X\beta^{(1)}}}{e^{X\beta^{(1)}} + e^{X\beta^{(2)}} + e^{X\beta^{(3)}}}$$
$$\Pr(y=2) = \frac{e^{X\beta^{(2)}}}{e^{X\beta^{(1)}} + e^{X\beta^{(2)}} + e^{X\beta^{(3)}}}$$
$$\Pr(y=3) = \frac{e^{X\beta^{(3)}}}{e^{X\beta^{(1)}} + e^{X\beta^{(2)}} + e^{X\beta^{(3)}}}$$

The model, however, is unidentified in the sense that there is more than one solution to $\beta^{(1)}$, $\beta^{(2)}$, and $\beta^{(3)}$ that leads to the same probabilities for $y = 1$, $y = 2$, and $y = 3$. To identify the model, you arbitrarily set one of $\beta^{(1)}$, $\beta^{(2)}$, or $\beta^{(3)}$ to 0—it does not matter which. That is, if you arbitrarily set $\beta^{(1)} = 0$, the remaining coefficients $\beta^{(2)}$ and $\beta^{(3)}$ will measure the change relative to the $y = 1$ group. If you instead set $\beta^{(2)} = 0$, the remaining coefficients $\beta^{(1)}$ and $\beta^{(3)}$ will measure the change relative to the $y = 2$ group. The coefficients will differ because they have different interpretations, but the predicted probabilities for $y = 1$, 2, and 3 will still be the same. Thus either parameterization will be a solution to the same underlying model.

Setting $\beta^{(1)} = 0$, the equations become

$$\Pr(y=1) = \frac{1}{1+e^{X\beta^{(2)}}+e^{X\beta^{(3)}}}$$
$$\Pr(y=2) = \frac{e^{X\beta^{(2)}}}{1+e^{X\beta^{(2)}}+e^{X\beta^{(3)}}}$$
$$\Pr(y=3) = \frac{e^{X\beta^{(3)}}}{1+e^{X\beta^{(2)}}+e^{X\beta^{(3)}}}$$

The relative probability of $y = 2$ to the base outcome is

$$\frac{\Pr(y=2)}{\Pr(y=1)} = e^{X\beta^{(2)}}$$

Let's call this ratio the relative risk, and let's further assume that X and $\beta_k^{(2)}$ are vectors equal to $(x_1, x_2, \dots, x_k)$ and $(\beta_1^{(2)}, \beta_2^{(2)}, \dots, \beta_k^{(2)})'$, respectively. The ratio of the relative risk for a one-unit change in x_i is then

$$\frac{e^{\beta_1^{(2)}x_1+\cdots+\beta_i^{(2)}(x_i+1)+\cdots+\beta_k^{(2)}x_k}}{e^{\beta_1^{(2)}x_1+\cdots+\beta_i^{(2)}x_i+\cdots+\beta_k^{(2)}x_k}} = e^{\beta_i^{(2)}}$$

Thus the exponentiated value of a coefficient is the relative-risk ratio for a one-unit change in the corresponding variable (risk is measured as the risk of the outcome relative to the base outcome).

Fitting unconstrained models

▷ Example 1: A first example

We have data on the type of health insurance available to 616 psychologically depressed subjects in the United States (Tarlov et al. 1989; Wells et al. 1989). The insurance is categorized as either an indemnity plan (that is, regular fee-for-service insurance, which may have a deductible or coinsurance rate) or a prepaid plan (a fixed up-front payment allowing subsequent unlimited use as provided, for instance, by an HMO). The third possibility is that the subject has no insurance whatsoever. We wish to explore the demographic factors associated with each subject's insurance choice. One of the demographic factors in our data is the race of the participant, coded as white or nonwhite:

```
. use http://www.stata-press.com/data/r13/sysdsn1
(Health insurance data)

. tabulate insure nonwhite, chi2 col

+-------------------+
| Key               |
|-------------------|
|     frequency     |
| column percentage |
+-------------------+

           |       nonwhite
    insure |         0          1 |     Total
-----------+----------------------+----------
 Indemnity |       251         43 |       294 
           |     50.71      35.54 |     47.73 
-----------+----------------------+----------
   Prepaid |       208         69 |       277 
           |     42.02      57.02 |     44.97 
-----------+----------------------+----------
  Uninsure |        36          9 |        45 
           |      7.27       7.44 |      7.31 
-----------+----------------------+----------
     Total |       495        121 |       616 
           |    100.00     100.00 |    100.00 

          Pearson chi2(2) =   9.5599   Pr = 0.008
```

Although insure appears to take on the values Indemnity, Prepaid, and Uninsure, it actually takes on the values 1, 2, and 3. The words appear because we have associated a value label with the numeric variable insure; see [U] 12.6.3 Value labels.

When we fit a multinomial logit model, we can tell mlogit which outcome to use as the base outcome, or we can let mlogit choose. To fit a model of insure on nonwhite, letting mlogit choose the base outcome, we type

```
. mlogit insure nonwhite
Iteration 0:   log likelihood = -556.59502
Iteration 1:   log likelihood = -551.78935
Iteration 2:   log likelihood = -551.78348
Iteration 3:   log likelihood = -551.78348

Multinomial logistic regression                   Number of obs   =        616
                                                  LR chi2(2)      =       9.62
                                                  Prob > chi2     =     0.0081
Log likelihood = -551.78348                       Pseudo R2       =     0.0086

------------------------------------------------------------------------------
      insure |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
Indemnity    |  (base outcome)
-------------+----------------------------------------------------------------
Prepaid      |
    nonwhite |   .6608212   .2157321     3.06   0.002     .2379942    1.083648
       _cons |  -.1879149   .0937644    -2.00   0.045    -.3716896   -.0041401
-------------+----------------------------------------------------------------
Uninsure     |
    nonwhite |   .3779586    .407589     0.93   0.354    -.4209011    1.176818
       _cons |  -1.941934   .1782185   -10.90   0.000    -2.291236   -1.592632
------------------------------------------------------------------------------
```

mlogit chose the indemnity outcome as the base outcome and presented coefficients for the outcomes prepaid and uninsured. According to the model, the probability of prepaid for whites (nonwhite $= 0$) is

$$\Pr(\texttt{insure} = \texttt{Prepaid}) = \frac{e^{-.188}}{1 + e^{-.188} + e^{-1.942}} = 0.420$$

Similarly, for nonwhites, the probability of prepaid is

$$\Pr(\texttt{insure} = \texttt{Prepaid}) = \frac{e^{-.188+.661}}{1 + e^{-.188+.661} + e^{-1.942+.378}} = 0.570$$

These results agree with the column percentages presented by `tabulate` because the `mlogit` model is fully saturated. That is, there are enough terms in the model to fully explain the column percentage in each cell. The model chi-squared and the `tabulate` chi-squared are in almost perfect agreement; both test that the column percentages of `insure` are the same for both values of `nonwhite`.

◁

▷ Example 2: Specifying the base outcome

By specifying the `baseoutcome()` option, we can control which outcome of the dependent variable is treated as the base. Left to its own, `mlogit` chose to make outcome 1, indemnity, the base outcome. To make outcome 2, prepaid, the base, we would type

```
. mlogit insure nonwhite, base(2)
Iteration 0:   log likelihood = -556.59502
Iteration 1:   log likelihood = -551.78935
Iteration 2:   log likelihood = -551.78348
Iteration 3:   log likelihood = -551.78348
Multinomial logistic regression                   Number of obs   =        616
                                                  LR chi2(2)      =       9.62
                                                  Prob > chi2     =     0.0081
Log likelihood = -551.78348                       Pseudo R2       =     0.0086

------------------------------------------------------------------------------
      insure |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
Indemnity    |
    nonwhite |  -.6608212   .2157321    -3.06   0.002    -1.083648   -.2379942
       _cons |   .1879149   .0937644     2.00   0.045     .0041401    .3716896
-------------+----------------------------------------------------------------
Prepaid      |  (base outcome)
-------------+----------------------------------------------------------------
Uninsure     |
    nonwhite |  -.2828627   .3977302    -0.71   0.477      -1.0624    .4966742
       _cons |  -1.754019   .1805145    -9.72   0.000    -2.107821   -1.400217
------------------------------------------------------------------------------
```

The `baseoutcome()` option requires that we specify the numeric value of the outcome, so we could not type `base(Prepaid)`.

Although the coefficients now appear to be different, the summary statistics reported at the top are identical. With this parameterization, the probability of prepaid insurance for whites is

$$\Pr(\texttt{insure} = \texttt{Prepaid}) = \frac{1}{1 + e^{.188} + e^{-1.754}} = 0.420$$

This is the same answer we obtained previously.

◁

▷ Example 3: Displaying relative-risk ratios

By specifying `rrr`, which we can do at estimation time or when we redisplay results, we see the model in terms of relative-risk ratios:

```
. mlogit, rrr
Multinomial logistic regression                 Number of obs     =        616
                                                LR chi2(2)        =       9.62
                                                Prob > chi2       =     0.0081
Log likelihood = -551.78348                     Pseudo R2         =     0.0086
```

insure	RRR	Std. Err.	z	P>|z|	[95% Conf.	Interval]
Indemnity						
nonwhite	.516427	.1114099	-3.06	0.002	.3383588	.7882073
_cons	1.206731	.1131483	2.00	0.045	1.004149	1.450183
Prepaid	(base outcome)					
Uninsure						
nonwhite	.7536233	.2997387	-0.71	0.477	.3456255	1.643247
_cons	.1730769	.0312429	-9.72	0.000	.1215024	.2465434

Looked at this way, the relative risk of choosing an indemnity over a prepaid plan is 0.516 for nonwhites relative to whites.

To illustrate, from the output and discussions of examples 1 and 2 we find that

$$\Pr\left(\texttt{insure} = \texttt{Indemnity} \mid \text{white}\right) = \frac{1}{1+e^{-.188}+e^{-1.942}} = 0.507$$

and thus the relative risk of choosing indemnity over prepaid (for whites) is

$$\frac{\Pr\left(\texttt{insure} = \texttt{Indemnity} \mid \text{white}\right)}{\Pr\left(\texttt{insure} = \texttt{Prepaid} \mid \text{white}\right)} = \frac{0.507}{0.420} = 1.207$$

For nonwhites,

$$\Pr\left(\texttt{insure} = \texttt{Indemnity} \mid \text{not white}\right) = \frac{1}{1+e^{-.188+.661}+e^{-1.942+.378}} = 0.355$$

and thus the relative risk of choosing indemnity over prepaid (for nonwhites) is

$$\frac{\Pr\left(\texttt{insure} = \texttt{Indemnity} \mid \text{not white}\right)}{\Pr\left(\texttt{insure} = \texttt{Prepaid} \mid \text{not white}\right)} = \frac{0.355}{0.570} = 0.623$$

The ratio of these two relative risks, hence the name “relative-risk ratio”, is $0.623/1.207 = 0.516$, as given in the output under the heading “RRR”.

◁

❑ Technical note

In models where only two categories are considered, the `mlogit` model reduces to standard `logit`. Consequently the exponentiated regression coefficients, labeled as `RRR` within `mlogit`, are equal to the odds ratios as given when the `or` option is specified under `logit`; see [R] **logit**.

As such, always referring to `mlogit`'s exponentiated coefficients as odds ratios may be tempting. However, the discussion in example 3 demonstrates that doing so would be incorrect. In general `mlogit` models, the exponentiated coefficients are ratios of relative risks, not ratios of odds.

❑

▷ Example 4: Model with continuous and multiple categorical variables

One of the advantages of `mlogit` over `tabulate` is that we can include continuous variables and multiple categorical variables in the model. In examining the data on insurance choice, we decide that we want to control for age, gender, and site of study (the study was conducted in three sites):

```
. mlogit insure age male nonwhite i.site
Iteration 0:   log likelihood = -555.85446
Iteration 1:   log likelihood = -534.67443
Iteration 2:   log likelihood = -534.36284
Iteration 3:   log likelihood = -534.36165
Iteration 4:   log likelihood = -534.36165
Multinomial logistic regression                   Number of obs   =        615
                                                  LR chi2(10)     =      42.99
                                                  Prob > chi2     =     0.0000
Log likelihood = -534.36165                       Pseudo R2       =     0.0387

------------------------------------------------------------------------------
      insure |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
Indemnity    |  (base outcome)
-------------+----------------------------------------------------------------
Prepaid      |
         age |   -.011745   .0061946    -1.90   0.058    -.0238862    .0003962
        male |   .5616934   .2027465     2.77   0.006     .1643175    .9590693
    nonwhite |   .9747768   .2363213     4.12   0.000     .5115955    1.437958
             |
        site |
          2  |   .1130359   .2101903     0.54   0.591    -.2989296    .5250013
          3  |  -.5879879   .2279351    -2.58   0.010    -1.034733   -.1412433
             |
       _cons |   .2697127   .3284422     0.82   0.412    -.3740222    .9134476
-------------+----------------------------------------------------------------
Uninsure     |
         age |  -.0077961   .0114418    -0.68   0.496    -.0302217    .0146294
        male |   .4518496   .3674867     1.23   0.219     -.268411     1.17211
    nonwhite |   .2170589   .4256361     0.51   0.610    -.6171725     1.05129
             |
        site |
          2  |  -1.211563   .4705127    -2.57   0.010    -2.133751   -.2893747
          3  |  -.2078123   .3662926    -0.57   0.570    -.9257327     .510108
             |
       _cons |  -1.286943   .5923219    -2.17   0.030    -2.447872   -.1260134
------------------------------------------------------------------------------
```

These results suggest that the inclination of nonwhites to choose prepaid care is even stronger than it was without controlling. We also see that subjects in site 2 are less likely to be uninsured.

◁

Fitting constrained models

mlogit can fit models with subsets of coefficients constrained to be zero, with subsets of coefficients constrained to be equal both within and across equations, and with subsets of coefficients arbitrarily constrained to equal linear combinations of other estimated coefficients.

Before fitting a constrained model, you define the constraints with the constraint command; see [R] **constraint**. Once the constraints are defined, you estimate using mlogit, specifying the constraint() option. Typing constraint(4) would use the constraint you previously saved as 4. Typing constraint(1,4,6) would use the previously stored constraints 1, 4, and 6. Typing constraint(1-4,6) would use the previously stored constraints 1, 2, 3, 4, and 6.

Sometimes you will not be able to specify the constraints without knowing the omitted outcome. In such cases, assume that the omitted outcome is whatever outcome is convenient for you, and include the baseoutcome() option when you specify the mlogit command.

▷ Example 5: Specifying constraints to test hypotheses

We can use constraints to test hypotheses, among other things. In our insurance-choice model, let's test the hypothesis that there is no distinction between having indemnity insurance and being uninsured. Indemnity-style insurance was the omitted outcome, so we type

```
. test [Uninsure]

 ( 1)  [Uninsure]age = 0
 ( 2)  [Uninsure]male = 0
 ( 3)  [Uninsure]nonwhite = 0
 ( 4)  [Uninsure]1b.site = 0
 ( 5)  [Uninsure]2.site = 0
 ( 6)  [Uninsure]3.site = 0
       Constraint 4 dropped

           chi2(  5) =    9.31
         Prob > chi2 =    0.0973
```

If indemnity had not been the omitted outcome, we would have typed test [Uninsure=Indemnity].

The results produced by test are an approximation based on the estimated covariance matrix of the coefficients. Because the probability of being uninsured is low, the log likelihood may be nonlinear for the uninsured. Conventional statistical wisdom is not to trust the asymptotic answer under these circumstances but to perform a likelihood-ratio test instead.

To use Stata's lrtest (likelihood-ratio test) command, we must fit both the unconstrained and constrained models. The unconstrained model is the one we have previously fit. Following the instruction in [R] **lrtest**, we first store the unconstrained model results:

```
. estimates store unconstrained
```

To fit the constrained model, we must refit our model with all the coefficients except the constant set to 0 in the Uninsure equation. We define the constraint and then refit:

```
. constraint 1 [Uninsure]
. mlogit insure age male nonwhite i.site, constraints(1)
Iteration 0:   log likelihood = -555.85446
Iteration 1:   log likelihood = -539.80523
Iteration 2:   log likelihood = -539.75644
Iteration 3:   log likelihood = -539.75643
Multinomial logistic regression                   Number of obs   =        615
                                                  Wald chi2(5)    =      29.70
Log likelihood = -539.75643                       Prob > chi2     =     0.0000
 ( 1)  [Uninsure]o.age = 0
 ( 2)  [Uninsure]o.male = 0
 ( 3)  [Uninsure]o.nonwhite = 0
 ( 4)  [Uninsure]2o.site = 0
 ( 5)  [Uninsure]3o.site = 0

      insure        Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]

Indemnity     (base outcome)

Prepaid
         age    -.0107025   .0060039    -1.78   0.075    -.0224699    .0010649
        male     .4963616   .1939683     2.56   0.010     .1161907    .8765324
    nonwhite     .9421369   .2252094     4.18   0.000     .5007346    1.383539

        site
          2      .2530912   .2029465     1.25   0.212    -.1446767    .6508591
          3     -.5521773   .2187237    -2.52   0.012    -.9808678   -.1234869

       _cons     .1792752   .3171372     0.57   0.572    -.4423023    .8008527

Uninsure
         age            0  (omitted)
        male            0  (omitted)
    nonwhite            0  (omitted)

        site
          2             0  (omitted)
          3             0  (omitted)

       _cons     -1.87351   .1601099   -11.70   0.000     -2.18732     -1.5597
```

We can now perform the likelihood-ratio test:

```
. lrtest unconstrained .
Likelihood-ratio test                                 LR chi2(5)  =     10.79
(Assumption: . nested in unconstrained)               Prob > chi2 =    0.0557
```

The likelihood-ratio chi-squared is 10.79 with 5 degrees of freedom—just slightly greater than the magic $p = 0.05$ level—so we should not call this difference significant. ◁

❑ Technical note

In certain circumstances, you should fit a multinomial logit model with conditional logit; see [R] **clogit**. With substantial data manipulation, `clogit` can handle the same class of models with some interesting additions. For example, if we had available the price and deductible of the most competitive insurance plan of each type, `mlogit` could not use this information, but `clogit` could. ❑

Stored results

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

Scalars

`e(N)`	number of observations
`e(N_cd)`	number of completely determined observations
`e(k_out)`	number of outcomes
`e(k)`	number of parameters
`e(k_eq)`	number of equations in `e(b)`
`e(k_eq_model)`	number of equations in overall model test
`e(k_dv)`	number of dependent variables
`e(df_m)`	model degrees of freedom
`e(r2_p)`	pseudo-R-squared
`e(ll)`	log likelihood
`e(ll_0)`	log likelihood, constant-only model
`e(N_clust)`	number of clusters
`e(chi2)`	χ^2
`e(p)`	significance
`e(k_eq_base)`	equation number of the base outcome
`e(baseout)`	the value of *depvar* to be treated as the base outcome
`e(ibaseout)`	index of the base outcome
`e(rank)`	rank of `e(V)`
`e(ic)`	number of iterations
`e(rc)`	return code
`e(converged)`	`1` if converged, `0` otherwise

Macros

`e(cmd)`	`mlogit`
`e(cmdline)`	command as typed
`e(depvar)`	name of dependent variable
`e(wtype)`	weight type
`e(wexp)`	weight expression
`e(title)`	title in estimation output
`e(clustvar)`	name of cluster variable
`e(chi2type)`	`Wald` or `LR`; type of model χ^2 test
`e(vce)`	*vcetype* specified in `vce()`
`e(vcetype)`	title used to label Std. Err.
`e(eqnames)`	names of equations
`e(baselab)`	value label corresponding to base outcome
`e(opt)`	type of optimization
`e(which)`	`max` or `min`; whether optimizer is to perform maximization or minimization
`e(ml_method)`	type of `ml` method
`e(user)`	name of likelihood-evaluator program
`e(technique)`	maximization technique
`e(properties)`	`b V`
`e(predict)`	program used to implement `predict`
`e(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(out)`	outcome values
`e(Cns)`	constraints matrix
`e(ilog)`	iteration log (up to 20 iterations)
`e(gradient)`	gradient vector
`e(V)`	variance–covariance matrix of the estimators
`e(V_modelbased)`	model-based variance

Functions

`e(sample)`	marks estimation sample

Methods and formulas

The multinomial logit model is described in Greene (2012, 763–766).

Suppose that there are k categorical outcomes and—without loss of generality—let the base outcome be 1. The probability that the response for the jth observation is equal to the ith outcome is

$$p_{ij} = \Pr(y_j = i) = \begin{cases} \dfrac{1}{1 + \sum_{m=2}^{k} \exp(\mathbf{x}_j \boldsymbol{\beta}_m)}, & \text{if} \quad i = 1 \\ \dfrac{\exp(\mathbf{x}_j \boldsymbol{\beta}_i)}{1 + \sum_{m=2}^{k} \exp(\mathbf{x}_j \boldsymbol{\beta}_m)}, & \text{if} \quad i > 1 \end{cases}$$

where $\mathbf{x}_j$ is the row vector of observed values of the independent variables for the jth observation and $\boldsymbol{\beta}_m$ is the coefficient vector for outcome m. The log pseudolikelihood is

$$\ln L = \sum_j w_j \sum_{i=1}^{k} I_i(y_j) \ln p_{ik}$$

where w_j is an optional weight and

$$I_i(y_j) = \begin{cases} 1, \text{ if } y_j = i \\ 0, \text{ otherwise} \end{cases}$$

Newton–Raphson maximum likelihood is used; see [R] **maximize**.

For constrained equations, the set of constraints is orthogonalized, and a subset of maximizable parameters is selected. For example, a parameter that is constrained to zero is not a maximizable parameter. If two parameters are constrained to be equal to each other, only one is a maximizable parameter.

Let $\mathbf{r}$ be the vector of maximizable parameters. $\mathbf{r}$ is physically a subset of the solution parameters, $\mathbf{b}$. A matrix, $\mathbf{T}$, and a vector, $\mathbf{m}$, are defined as

$$\mathbf{b} = \mathbf{T}\mathbf{r} + \mathbf{m}$$

so that

$$\frac{\partial f}{\partial \mathbf{b}} = \frac{\partial f}{\partial \mathbf{r}}\mathbf{T}'$$

$$\frac{\partial^2 f}{\partial \mathbf{b}^2} = \mathbf{T}\frac{\partial^2 f}{\partial \mathbf{r}^2}\mathbf{T}'$$

$\mathbf{T}$ consists of a block form in which one part is a permutation of the identity matrix and the other part describes how to calculate the constrained parameters from the maximizable parameters.

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*.

`mlogit` also supports estimation with survey data. For details on VCEs with survey data, see [SVY] **variance estimation**.

References

Davidson, R., and J. G. MacKinnon. 1993. *Estimation and Inference in Econometrics*. New York: Oxford University Press.

Freese, J., and J. S. Long. 2000. sg155: Tests for the multinomial logit model. *Stata Technical Bulletin* 58: 19–25. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, pp. 247–255. College Station, TX: Stata Press.

Greene, W. H. 2012. *Econometric Analysis*. 7th ed. Upper Saddle River, NJ: Prentice Hall.

Haan, P., and A. Uhlendorff. 2006. Estimation of multinomial logit models with unobserved heterogeneity using maximum simulated likelihood. *Stata Journal* 6: 229–245.

Hamilton, L. C. 1993. sqv8: Interpreting multinomial logistic regression. *Stata Technical Bulletin* 13: 24–28. Reprinted in *Stata Technical Bulletin Reprints*, vol. 3, pp. 176–181. College Station, TX: Stata Press.

——. 2013. *Statistics with Stata: Updated for Version 12*. 8th ed. Boston: Brooks/Cole.

Hendrickx, J. 2000. sbe37: Special restrictions in multinomial logistic regression. *Stata Technical Bulletin* 56: 18–26. Reprinted in *Stata Technical Bulletin Reprints*, vol. 10, pp. 93–103. College Station, TX: Stata Press.

Hole, A. R. 2007. Fitting mixed logit models by using maximum simulated likelihood. *Stata Journal* 7: 388–401.

Hosmer, D. W., Jr., S. A. Lemeshow, and R. X. Sturdivant. 2013. *Applied Logistic Regression*. 3rd ed. Hoboken, NJ: Wiley.

Kleinbaum, D. G., and M. Klein. 2010. *Logistic Regression: A Self-Learning Text*. 3rd ed. New York: Springer.

Long, J. S. 1997. *Regression Models for Categorical and Limited Dependent Variables*. Thousand Oaks, CA: Sage.

Long, J. S., and J. Freese. 2014. *Regression Models for Categorical Dependent Variables Using Stata*. 3rd ed. College Station, TX: Stata Press.

Tarlov, A. R., J. E. Ware, Jr., S. Greenfield, E. C. Nelson, E. Perrin, and M. Zubkoff. 1989. The medical outcomes study. An application of methods for monitoring the results of medical care. *Journal of the American Medical Association* 262: 925–930.

Treiman, D. J. 2009. *Quantitative Data Analysis: Doing Social Research to Test Ideas*. San Francisco: Jossey-Bass.

Wells, K. B., R. D. Hays, M. A. Burnam, W. H. Rogers, S. Greenfield, and J. E. Ware, Jr. 1989. Detection of depressive disorder for patients receiving prepaid or fee-for-service care. Results from the Medical Outcomes Survey. *Journal of the American Medical Association* 262: 3298–3302.

Xu, J., and J. S. Long. 2005. Confidence intervals for predicted outcomes in regression models for categorical outcomes. *Stata Journal* 5: 537–559.

Also see

[R] **mlogit postestimation** — Postestimation tools for mlogit

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

[R] **logistic** — Logistic regression, reporting odds ratios

[R] **logit** — Logistic regression, reporting coefficients

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

[R] **nlogit** — Nested logit regression

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

[R] **rologit** — Rank-ordered logistic regression

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

[MI] **estimation** — Estimation commands for use with mi estimate

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

[U] 20 Estimation and postestimation commands

Title

mlogit postestimation — Postestimation tools for mlogit

Description

The following postestimation commands are available after `mlogit`:

Command	Description
`contrast`	contrasts and ANOVA-style joint tests of estimates
`estat ic`	Akaike's and Schwarz's Bayesian information criteria (AIC and BIC)
`estat summarize`	summary statistics for the estimation sample
`estat vce`	variance–covariance matrix of the estimators (VCE)
`estat` (svy)	postestimation statistics for survey data
`estimates`	cataloging estimation results
`forecast`[1]	dynamic forecasts and simulations
`hausman`	Hausman's specification test
`lincom`	point estimates, standard errors, testing, and inference for linear combinations of coefficients
`lrtest`[2]	likelihood-ratio test
`margins`	marginal means, predictive margins, marginal effects, and average marginal effects
`marginsplot`	graph the results from margins (profile plots, interaction plots, etc.)
`nlcom`	point estimates, standard errors, testing, and inference for nonlinear combinations of coefficients
`predict`	predictions, residuals, influence statistics, and other diagnostic measures
`predictnl`	point estimates, standard errors, testing, and inference for generalized predictions
`pwcompare`	pairwise comparisons of estimates
`suest`	seemingly unrelated estimation
`test`	Wald tests of simple and composite linear hypotheses
`testnl`	Wald tests of nonlinear hypotheses

[1] `forecast` is not appropriate with `mi` or `svy` estimation results.
[2] `lrtest` is not appropriate with `svy` estimation results.

Syntax for predict

predict [*type*] { *stub** | *newvar* | *newvarlist* } [*if*] [*in*] [, *statistic* outcome(*outcome*)]

predict [*type*] { *stub** | *newvarlist* } [*if*] [*in*], scores

statistic	Description
Main	
pr	probability of a positive outcome; the default
xb	linear prediction
stdp	standard error of the linear prediction
stddp	standard error of the difference in two linear predictions

If you do not specify outcome(), pr (with one new variable specified), xb, and stdp assume outcome(#1). You must specify outcome() with the stddp option.

You specify one or k new variables with pr, where k is the number of outcomes.

You specify one new variable with xb, stdp, and stddp.

These statistics are available both in and out of sample; type predict ... if e(sample) ... if wanted only for the estimation sample.

Menu for predict

Statistics > Postestimation > Predictions, residuals, etc.

Options for predict

Main

pr, the default, calculates the probability of each of the categories of the dependent variable or the probability of the level specified in outcome(*outcome*). If you specify the outcome(*outcome*) option, you need to specify only one new variable; otherwise, you must specify a new variable for each category of the dependent variable.

xb calculates the linear prediction. You must also specify the outcome(*outcome*) option.

stdp calculates the standard error of the linear prediction. You must also specify the outcome(*outcome*) option.

stddp calculates the standard error of the difference in two linear predictions. You must specify the outcome(*outcome*) option, and here you specify the two particular outcomes of interest inside the parentheses, for example, predict sed, stddp outcome(1,3).

outcome(*outcome*) specifies the outcome for which the statistic is to be calculated. equation() is a synonym for outcome(): it does not matter which you use. outcome() or equation() can be specified using

> #1, #2, ..., where #1 means the first category of the dependent variable, #2 means the second category, etc.;
>
> the values of the dependent variable; or
>
> the value labels of the dependent variable if they exist.

`scores` calculates equation-level score variables. The number of score variables created will be one less than the number of outcomes in the model. If the number of outcomes in the model were k, then

the first new variable will contain $\partial \ln L/\partial(\mathbf{x}_j\boldsymbol{\beta}_1)$;

the second new variable will contain $\partial \ln L/\partial(\mathbf{x}_j\boldsymbol{\beta}_2)$;

...

the $(k-1)$th new variable will contain $\partial \ln L/\partial(\mathbf{x}_j\boldsymbol{\beta}_{k-1})$.

Remarks and examples

Remarks are presented under the following headings:

Obtaining predicted values
Calculating marginal effects
Testing hypotheses about coefficients

Obtaining predicted values

▷ Example 1: Obtaining predicted probabilities

After estimation, we can use `predict` to obtain predicted probabilities, index values, and standard errors of the index, or differences in the index. For instance, in example 4 of [R] **mlogit**, we fit a model of insurance choice on various characteristics. We can obtain the predicted probabilities for outcome 1 by typing

```
. use http://www.stata-press.com/data/r13/sysdsn1
(Health insurance data)
. mlogit insure age i.male i.nonwhite i.site
  (output omitted)
. predict p1 if e(sample), outcome(1)
(option pr assumed; predicted probability)
(29 missing values generated)
. summarize p1
```

Variable	Obs	Mean	Std. Dev.	Min	Max
p1	615	.4764228	.1032279	.1698142	.71939

We added the `i.` prefix to the `male`, `nonwhite`, and `site` variables to explicitly identify them as factor variables. That makes no difference in the estimated results, but we will take advantage of it in later examples. We also included `if e(sample)` to restrict the calculation to the estimation sample. In example 4 of [R] **mlogit**, the multinomial logit model was fit on 615 observations, so there must be missing values in our dataset.

Although we typed `outcome(1)`, specifying `1` for the indemnity outcome, we could have typed `outcome(Indemnity)`. For instance, to obtain the probabilities for prepaid, we could type

```
. predict p2 if e(sample), outcome(Prepaid)
(option pr assumed; predicted probability)
(29 missing values generated)
. summarize p2
```

Variable	Obs	Mean	Std. Dev.	Min	Max
p2	615	.4504065	.1125962	.1964103	.7885724

We must specify the label exactly as it appears in the underlying value label (or how it appears in the mlogit output), including capitalization.

Here we have used predict to obtain probabilities for the same sample on which we estimated. That is not necessary. We could use another dataset that had the independent variables defined (in our example, age, male, nonwhite, and site) and use predict to obtain predicted probabilities; here, we would not specify if e(sample).

◁

▷ Example 2: Obtaining index values

predict can also be used to obtain the index values—the $\sum x_i \widehat{\beta}_i^{(k)}$—as well as the probabilities:

```
. predict idx1, outcome(Indemnity) xb
(1 missing value generated)

. summarize idx1

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
        idx1 |       643           0           0          0          0
```

The indemnity outcome was our base outcome—the outcome for which all the coefficients were set to 0—so the index is always 0. For the prepaid and uninsured outcomes, we type

```
. predict idx2, outcome(Prepaid) xb
(1 missing value generated)

. predict idx3, outcome(Uninsure) xb
(1 missing value generated)

. summarize idx2 idx3

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
        idx2 |       643   -.0566113     .4962973  -1.298198   1.700719
        idx3 |       643   -1.980747     .6018139  -3.112741  -.8258458
```

We can obtain the standard error of the index by specifying the stdp option:

```
. predict se2, outcome(Prepaid) stdp
(1 missing value generated)

. list p2 idx2 se2 in 1/5

     +----------------------------------+
     |       p2        idx2         se2 |
     |----------------------------------|
  1. | .3709022   -.4831167    .2437772 |
  2. | .4977667     .055111    .1694686 |
  3. | .4113073   -.1712106    .1793498 |
  4. | .5424927    .3788345    .2513701 |
  5. |        .   -.0925817    .1452616 |
     +----------------------------------+
```

We obtained the probability, p2, in the previous example.

Finally, `predict` can calculate the standard error of the difference in the index values between two outcomes with the `stddp` option:

```
. predict se_2_3, outcome(Prepaid,Uninsure) stddp
(1 missing value generated)

. list idx2 idx3 se_2_3 in 1/5

     +-----------------------------------+
     |       idx2        idx3     se_2_3 |
     |-----------------------------------|
  1. |  -.4831167   -3.073253   .5469354 |
  2. |    .055111   -2.715986   .4331918 |
  3. |  -.1712106   -1.579621   .3053815 |
  4. |   .3788345   -1.462007   .4492552 |
  5. |  -.0925817   -2.814022   .4024784 |
     +-----------------------------------+
```

In the first observation, the difference in the indexes is $-0.483 - (-3.073) = 2.59$. The standard error of that difference is 0.547.

◁

▷ Example 3: Interpreting results using predictive margins

It is more difficult to interpret the results from `mlogit` than those from `clogit` or `logit` because there are multiple equations. For example, suppose that one of the independent variables in our model takes on the values 0 and 1, and we are attempting to understand the effect of this variable. Assume that the coefficient on this variable for the second outcome, $\beta^{(2)}$, is positive. We might then be tempted to reason that the probability of the second outcome is higher if the variable is 1 rather than 0. Most of the time, that will be true, but occasionally we will be surprised. The probability of some other outcome could increase even more (say, $\beta^{(3)} > \beta^{(2)}$), and thus the probability of outcome 2 would actually fall relative to that outcome. We can use `predict` to help interpret such results.

Continuing with our previously fit insurance-choice model, we wish to describe the model's predictions by race. For this purpose, we can use the method of predictive margins (also known as recycled predictions), in which we vary characteristics of interest across the whole dataset and average the predictions. That is, we have data on both whites and nonwhites, and our individuals have other characteristics as well. We will first pretend that all the people in our data are white but hold their other characteristics constant. We then calculate the probabilities of each outcome. Next we will pretend that all the people in our data are nonwhite, still holding their other characteristics constant. Again we calculate the probabilities of each outcome. The difference in those two sets of calculated probabilities, then, is the difference due to race, holding other characteristics constant.

```
. gen byte nonwhold=nonwhite               // save real race

. replace nonwhite=0                       // make everyone white
(126 real changes made)

. predict wpind, outcome(Indemnity)        // predict probabilities
(option pr assumed; predicted probability)
(1 missing value generated)

. predict wpp, outcome(Prepaid)
(option pr assumed; predicted probability)
(1 missing value generated)

. predict wpnoi, outcome(Uninsure)
(option pr assumed; predicted probability)
(1 missing value generated)

. replace nonwhite=1                       // make everyone nonwhite
(644 real changes made)
```

```
. predict nwpind, outcome(Indemnity)
(option pr assumed; predicted probability)
(1 missing value generated)
. predict nwpp, outcome(Prepaid)
(option pr assumed; predicted probability)
(1 missing value generated)
. predict nwpnoi, outcome(Uninsure)
(option pr assumed; predicted probability)
(1 missing value generated)
. replace nonwhite=nonwhold              // restore real race
(518 real changes made)
. summarize wp* nwp*, sep(3)
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
       wpind |       643    .5141673    .0872679   .3092903     .71939
         wpp |       643    .4082052    .0993286   .1964103   .6502247
       wpnoi |       643    .0776275    .0360283   .0273596   .1302816
-------------+--------------------------------------------------------
      nwpind |       643    .3112809    .0817693   .1511329    .535021
        nwpp |       643     .630078    .0979976   .3871782   .8278881
      nwpnoi |       643    .0586411    .0287185   .0209648   .0933874
```

In example 1 of [R] **mlogit**, we presented a cross-tabulation of insurance type and race. Those values were unadjusted. The means reported above are the values adjusted for age, sex, and site. Combining the results gives

	Unadjusted		Adjusted	
	white	nonwhite	white	nonwhite
Indemnity	0.51	0.36	0.51	0.31
Prepaid	0.42	0.57	0.41	0.63
Uninsured	0.07	0.07	0.08	0.06

We find, for instance, after adjusting for age, sex, and site, that although 57% of nonwhites in our data had prepaid plans, 63% of nonwhites chose prepaid plans.

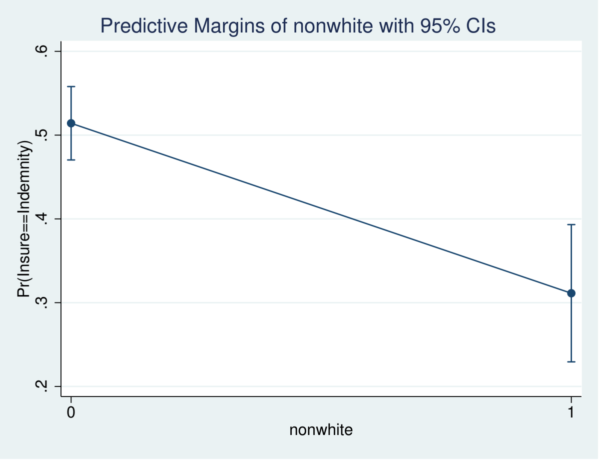
Computing predictive margins by hand was instructive, but we can compute these values more easily using the `margins` command (see [R] **margins**). The two margins for the indemnity outcome can be estimated by typing

```
. margins nonwhite, predict(outcome(Indemnity)) noesample
Predictive margins                                Number of obs   =        643
Model VCE    : OIM
Expression   : Pr(insure==Indemnity), predict(outcome(Indemnity))
------------------------------------------------------------------------------
             |            Delta-method
             |     Margin   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    nonwhite |
          0  |   .5141673   .0223485    23.01   0.000      .470365    .5579695
          1  |   .3112809   .0418049     7.45   0.000     .2293448     .393217
------------------------------------------------------------------------------
```

`margins` also estimates the standard errors and confidence intervals of the margins. By default, `margins` uses only the estimation sample. We added the `noesample` option so that `margins` would use the entire sample and produce results comparable to our earlier analysis.

We can use `marginsplot` to graph the results from `margins`:

```
. marginsplot
  Variables that uniquely identify margins: nonwhite
```



The margins for the other two outcomes can be computed by typing

```
. margins nonwhite, predict(outcome(Prepaid)) noesample
  (output omitted)
. margins nonwhite, predict(outcome(Uninsure)) noesample
  (output omitted)
```

◁

❑ Technical note

You can use `predict` to classify predicted values and compare them with the observed outcomes to interpret a multinomial logit model. This is a variation on the notions of sensitivity and specificity for logistic regression. Here we will classify indemnity and prepaid as definitely predicting indemnity, definitely predicting prepaid, and ambiguous.

```
. predict indem, outcome(Indemnity) index            // obtain indexes
(1 missing value generated)
. predict prepaid, outcome(Prepaid) index
(1 missing value generated)
. gen diff = prepaid-indem                           // obtain difference
(1 missing value generated)
. predict sediff, outcome(Indemnity,Prepaid) stddp   // & its standard error
(1 missing value generated)
. gen type = 1 if diff/sediff < -1.96                // definitely indemnity
(504 missing values generated)
. replace type = 3 if diff/sediff > 1.96             // definitely prepaid
(100 real changes made)
. replace type = 2 if type>=. & diff/sediff < .      // ambiguous
(404 real changes made)
. label def type 1 "Def Ind" 2 "Ambiguous" 3 "Def Prep"
. label values type type                             // label results
```

```
. tabulate insure type

           |               type
    insure |   Def Ind  Ambiguous   Def Prep |     Total
-----------+---------------------------------+----------
 Indemnity |        78        183         33 |       294
   Prepaid |        44        177         56 |       277
  Uninsure |        12         28          5 |        45
-----------+---------------------------------+----------
     Total |       134        388         94 |       616
```

We can see that the predictive power of this model is modest. There are many misclassifications in both directions, though there are more correctly classified observations than misclassified observations.

Also the uninsured look overwhelmingly as though they might have come from the indemnity system rather than from the prepaid system.

❑

Calculating marginal effects

▷ Example 4

We have already noted that the coefficients from multinomial logit can be difficult to interpret because they are relative to the base outcome. Another way to evaluate the effect of covariates is to examine the marginal effect of changing their values on the probability of observing an outcome.

The `margins` command can be used for this too. We can estimate the marginal effect of each covariate on the probability of observing the first outcome—indemnity insurance—by typing

```
. margins, dydx(*) predict(outcome(Indemnity))
Average marginal effects                        Number of obs     =        615
Model VCE    : OIM

Expression   : Pr(insure==Indemnity), predict(outcome(Indemnity))
dy/dx w.r.t. : age 1.male 1.nonwhite 2.site 3.site

------------------------------------------------------------------------------
             |            Delta-method
             |      dy/dx   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         age |   .0026655    .001399     1.91   0.057    -.0000765    .0054074
      1.male |  -.1295734   .0450945    -2.87   0.004    -.2179571   -.0411898
  1.nonwhite |  -.2032404   .0482554    -4.21   0.000    -.2978192   -.1086616
             |
        site |
          2  |   .0070995   .0479993     0.15   0.882    -.0869775    .1011765
          3  |   .1216165   .0505833     2.40   0.016      .022475     .220758
------------------------------------------------------------------------------
Note: dy/dx for factor levels is the discrete change from the base level.
```

By default, `margins` estimates the average marginal effect over the estimation sample, and that is what we see above. Being male decreases the average probability of having indemnity insurance by 0.130. We also see, from the note at the bottom of the table, that the marginal effect was computed as a discrete change in the probability of being male rather than female. That is why we made `male` a factor variable when fitting the model.

The `dydx(*)` option requested that `margins` estimate the marginal effect for each regressor, `dydx(age)` would have produced estimates only for the effect of `age`. `margins` has many options for controlling how the marginal effect is computed, including the ability to average over subgroups or to compute estimates for specified values of the regressors; see [R] **margins**.

We could evaluate the marginal effects on the other two outcomes by typing

```
. margins, dydx(*) predict(outcome(Prepaid))
  (output omitted )
. margins, dydx(*) predict(outcome(Uninsure))
  (output omitted )
```

Testing hypotheses about coefficients

Example 5

`test` tests hypotheses about the coefficients just as after any estimation command; see [R] **test**. Note, however, `test`'s syntax for dealing with multiple-equation models. Because `test` bases its results on the estimated covariance matrix, we might prefer a likelihood-ratio test; see example 5 in [R] **mlogit** for an example of `lrtest`.

If we simply list variables after the `test` command, we are testing that the corresponding coefficients are zero across all equations:

```
. test 2.site 3.site

 ( 1)  [Indemnity]2o.site = 0
 ( 2)  [Prepaid]2.site = 0
 ( 3)  [Uninsure]2.site = 0
 ( 4)  [Indemnity]3o.site = 0
 ( 5)  [Prepaid]3.site = 0
 ( 6)  [Uninsure]3.site = 0
       Constraint 1 dropped
       Constraint 4 dropped

           chi2(  4) =   19.74
         Prob > chi2 =    0.0006
```

We can test that all the coefficients (except the constant) in an equation are zero by simply typing the outcome in square brackets:

```
. test [Uninsure]

 ( 1)  [Uninsure]age = 0
 ( 2)  [Uninsure]0b.male = 0
 ( 3)  [Uninsure]1.male = 0
 ( 4)  [Uninsure]0b.nonwhite = 0
 ( 5)  [Uninsure]1.nonwhite = 0
 ( 6)  [Uninsure]1b.site = 0
 ( 7)  [Uninsure]2.site = 0
 ( 8)  [Uninsure]3.site = 0
       Constraint 2 dropped
       Constraint 4 dropped
       Constraint 6 dropped

           chi2(  5) =    9.31
         Prob > chi2 =    0.0973
```

We specify the outcome just as we do with `predict`; we can specify the label if the outcome variable is labeled, or we can specify the numeric value of the outcome. We would have obtained the same test as above if we had typed `test [3]` because 3 is the value of `insure` for the outcome uninsured.

We can combine the two syntaxes. To test that the coefficients on the site variables are 0 in the equation corresponding to the outcome prepaid, we can type

```
. test [Prepaid]: 2.site 3.site

 ( 1)  [Prepaid]2.site = 0
 ( 2)  [Prepaid]3.site = 0

           chi2(  2) =   10.78
         Prob > chi2 =    0.0046
```

We specified the outcome and then followed that with a colon and the variables we wanted to test.

We can also test that coefficients are equal across equations. To test that all coefficients except the constant are equal for the prepaid and uninsured outcomes, we can type

```
. test [Prepaid=Uninsure]

 ( 1)  [Prepaid]age - [Uninsure]age = 0
 ( 2)  [Prepaid]0b.male - [Uninsure]0b.male = 0
 ( 3)  [Prepaid]1.male - [Uninsure]1.male = 0
 ( 4)  [Prepaid]0b.nonwhite - [Uninsure]0b.nonwhite = 0
 ( 5)  [Prepaid]1.nonwhite - [Uninsure]1.nonwhite = 0
 ( 6)  [Prepaid]1b.site - [Uninsure]1b.site = 0
 ( 7)  [Prepaid]2.site - [Uninsure]2.site = 0
 ( 8)  [Prepaid]3.site - [Uninsure]3.site = 0
       Constraint 2 dropped
       Constraint 4 dropped
       Constraint 6 dropped

           chi2(  5) =   13.80
         Prob > chi2 =    0.0169
```

To test that only the site variables are equal, we can type

```
. test [Prepaid=Uninsure]: 2.site 3.site

 ( 1)  [Prepaid]2.site - [Uninsure]2.site = 0
 ( 2)  [Prepaid]3.site - [Uninsure]3.site = 0

           chi2(  2) =   12.68
         Prob > chi2 =    0.0018
```

Finally, we can test any arbitrary constraint by simply entering the equation and specifying the coefficients as described in [U] **13.5 Accessing coefficients and standard errors**. The following hypothesis is senseless but illustrates the point:

```
. test ([Prepaid]age+[Uninsure]2.site)/2 = 2-[Uninsure]1.nonwhite

 ( 1)  .5*[Prepaid]age + [Uninsure]1.nonwhite + .5*[Uninsure]2.site = 2

           chi2(  1) =   22.45
         Prob > chi2 =    0.0000
```

See [R] **test** for more information about `test`. The information there about combining hypotheses across `test` commands (the `accumulate` option) also applies after `mlogit`.

◁

Reference

Fagerland, M. W., and D. W. Hosmer, Jr. 2012. A generalized HosmerLemeshow goodness-of-fit test for multinomial logistic regression models. *Stata Journal* 12: 447–453.

Also see

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

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