Title

mfx — Obtain marginal effects or elasticities after estimation

Syntax

`mfx` [`compute`] [*if*] [*in*] [, *options*]

`mfx replay` [, `level(#)`]

options	description
Model	
`predict(`*predict_option*`)`	calculate marginal effects (elasticities) for *predict_option*
`varlist(`*varlist*`)`	calculate marginal effects (elasticities) for *varlist*
`dydx`	calculate marginal effects; the default
`eyex`	calculate elasticities in the form of $\partial \log y / \partial \log x$
`dyex`	calculate elasticities in the form of $\partial y / \partial \log x$
`eydx`	calculate elasticities in the form of $\partial \log y / \partial x$
`nodiscrete`	treat dummy (indicator) variables as continuous
`nose`	do not calculate standard errors
Model 2	
`at(`*atlist*`)`	calculate marginal effects (elasticities) at these values
`noesample`	do not restrict calculation of means and medians to the estimation sample
`nowght`	ignore weights when calculating means and medians
Adv. model	
`nonlinear`	do not use the linear method
`force`	calculate marginal effects and standard errors when it would otherwise refuse to do so
Reporting	
`level(#)`	set confidence level; default is `level(95)`
`diagnostics(beta)`	report suitability of marginal-effect calculation
`diagnostics(vce)`	report suitability of standard-error calculation
`diagnostics(all)`	report all diagnostic information
`tracelvl(#)`	report increasing levels of detail during calculations

where *atlist* is *numlist* or *matname* or

[`mean` | `median` | `zero`] [*varname* = # [, *varname* = #] [...]]

where `mean` is the default.

Description

mfx numerically calculates the marginal effects or the elasticities and their standard errors after estimation. Exactly what mfx can calculate is determined by the previous estimation command and the predict(*predict_option*) option. The values at which the marginal effects or elasticities are to be evaluated is determined by the at(*atlist*) option. By default, mfx calculates the marginal effects or elasticities at the means of the independent variables by using the default prediction option associated with the previous estimation command.

Some disciplines use the term *partial effects*, rather than marginal effects, for what is computed by mfx.

mfx replay replays the results of the previous mfx computation.

Options

Model

predict(*predict_option*) specifies the function (that is, the form of y) for which to calculate the marginal effects or elasticities. The default is to use the default predict option of the preceding estimation command. To see which predict options are available, see help for that estimation command.

varlist(*varlist*) specifies the variables for which to calculate marginal effects (elasticities). The default is all variables.

dydx specifies that marginal effects be calculated. This is the default.

eyex specifies that elasticities be calculated in the form of $\partial \log y / \partial \log x$.

dyex specifies that elasticities be calculated in the form of $\partial y / \partial \log x$.

eydx specifies that elasticities be calculated in the form of $\partial \log y / \partial x$.

nodiscrete treats dummy variables as continuous. A dummy variable is one that takes on the value 0 or 1 in the estimation sample. If nodiscrete is not specified, the marginal effect of a dummy variable is calculated as the discrete change in y as the dummy variable changes from 0 to 1. This option is irrelevant to the computation of the elasticities because all dummy variables are treated as continuous when computing elasticities.

nose specifies that standard errors of the marginal effects (elasticities) not be computed.

Model 2

at(*atlist*) specifies the values at which the marginal effects (elasticities) are to be calculated. The default is to calculate at the means of the independent variables.

at(*numlist*) specifies that the marginal effects (elasticities) be calculated at *numlist*. For instance,

```
. use http://www.stata-press.com/data/r10/auto
. probit foreign mpg weight price
. mfx, predict(xb eq(#2)) at(200 3000 0.5)
```

computes the marginal effects for the second equation, setting disp $= 200$, weight $= 3000$, and foreign $= 0.5$.

The order of the values in the *numlist* is the same as the variables in the preceding estimation command, from left to right, without repetition. For instance,

```
. sureg (price disp weight) (mpg foreign disp)
. mfx, predict(xb) at(200 3000 0.5)
```

`at(`*matname*`)` specifies the points in a matrix format. The ordering of the variables is the same as that of *numlist*. For instance,

```
. probit foreign mpg weight price
. matrix A = (21, 3000, 6000)
. mfx, at(A)
```

`at(` [`mean` | `median` | `zero`] [*varname* = # [, *varname* = #] [...]] `)` specifies that the marginal effects (elasticities) be calculated at means, at medians of the independent variables, or at zeros. It also allows users to specify particular values for one or more independent variables, assuming that the rest are means, medians, or zeros.

```
. probit foreign mpg weight price
. mfx, at(mean mpg=30)
```

`at(`*varname* = # [, *varname* = #] [...] `)` specifies that the marginal effects or the elasticities be calculated at particular values for one or more independent variables, assuming that the rest are means.

```
. probit foreign mpg weight price
. mfx, at(mpg=30)
```

`noesample` affects `at(`*atlist*`)`, any offsets used in the preceding estimation, and the determination of dummy variables. It specifies that the whole dataset be considered instead of only those marked in the `e(sample)` defined by the previous estimation command.

`nowght` affects only `at(`*atlist*`)` and offsets. It specifies that weights be ignored when calculating the means or medians for the *atlist* and when calculating the means for any offsets.

Adv. model

`nonlinear` specifies that y, the function to be calculated for the marginal effects or the elasticities, does not meet the linear-form restriction. By default, `mfx` assumes that y meets the linear-form restriction, unless one or more dependent variables are shared by multiple equations or the previous estimation command was `nl` (see *Using mfx after nl* below). For instance, predictions after

```
. heckman mpg price, sel(foreign=rep78)
```

meet the linear-form restriction, but those after

```
. heckman mpg price, sel(foreign=rep78 price)
```

do not. If y meets the linear-form restriction, specifying `nonlinear` should produce the same results as not specifying it. However, the nonlinear method is generally more time consuming. Most likely, you do not need to specify `nonlinear` after an official Stata command. For user-written commands, if you are not sure whether y is of linear form, specifying `nonlinear` is a safe choice.

`force` specifies that marginal effects and their standard errors be calculated when it would otherwise refuse to do so. Such cases arise, for instance, when the marginal effect is a function of a random quantity other than the coefficients of the model (e.g., a residual). If you specify this option, there is no guarantee that the resulting marginal effects and standard errors are correct.

Reporting

`level(`#`)` specifies the confidence level, as a percentage, for confidence intervals. The default is `level(95)` or as set by `set level`; see [U] **23.5 Specifying the width of confidence intervals**.

`diagnostics(`*diaglist*`)` asks `mfx` to display various diagnostic information.

`diagnostics(beta)` shows the information used to determine whether the prediction option is suitable for computing marginal effects.

`diagnostics(vce)` shows the information used to determine whether the prediction option is suitable for computing the standard errors of the marginal effects.

`diagnostics(all)` shows all the above diagnostic information.

`tracelvl(`*#*`)` shows increasing levels of detail during calculations. # may be 1, 2, 3, or 4. Level 1 shows the marginal effects and standard errors as they are computed, and which method, either linear or nonlinear, was used. Level 2 shows, in addition, the components of the matrix of partial derivatives needed for each standard error as they are computed. Level 3 shows counts of iterations in obtaining a suitable finite difference for each numerical derivative. Level 4 shows the values of these finite differences.

Remarks

Remarks are presented under the following headings:

Obtaining marginal effects after single-equation (SE) estimation
Obtaining marginal effects after multiple-equation (ME) estimation
Specifying the evaluation points
Obtaining three forms of elasticities
Using mfx after nl

Obtaining marginal effects after single-equation (SE) estimation

Before running `mfx`, type `help` *estimation_cmd* to see what can be predicted after the estimation and to see the default prediction.

▷ Example 1

We fit a logit model with the auto dataset:

```
. use http://www.stata-press.com/data/r10/auto
(1978 Automobile Data)

. logit foreign mpg price, nolog

Logistic regression                               Number of obs   =         74
                                                  LR chi2(2)      =      17.14
                                                  Prob > chi2     =     0.0002
Log likelihood = -36.462189                       Pseudo R2       =     0.1903

------------------------------------------------------------------------------
     foreign |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         mpg |   .2338353   .0671449     3.48   0.000     .1022338    .3654368
       price |    .000266   .0001166     2.28   0.022     .0000375    .0004945
       _cons |  -7.648111   2.043673    -3.74   0.000    -11.65364   -3.642586
------------------------------------------------------------------------------
```

To determine the marginal effects of `mpg` and `price` for the probability of a positive outcome at their mean values, we can issue the `mfx` command without the `predict` option because the default prediction after `logit` is the probability of a positive outcome. The calculation is requested at the mean values by default.

```
. mfx
Marginal effects after logit
      y  = Pr(foreign) (predict)
         = .26347633
```

variable	dy/dx	Std. Err.	z	P>|z|	[95%	C.I.]	X
mpg	.0453773	.0131	3.46	0.001	.019702	.071053	21.2973
price	.0000516	.00002	2.31	0.021	7.8e-06	.000095	6165.26

The first line of the output indicates that the marginal effects were calculated after a `logit` estimation. The second line of the output describes the form of y and the `predict` command that we would type to calculate y separately. The third line of the output gives the value of y given the values of X, which are displayed in the last column of the table.

To calculate the marginal effects at particular data points, say, `mpg` $= 20$, `price` $= 6000$, specify the `at()` option:

```
. mfx, at(mpg=20, price=6000)
Marginal effects after logit
      y  = Pr(foreign) (predict)
         =  .20176601
```

variable	dy/dx	Std. Err.	z	P>|z|	[95%	C.I.]	X
mpg	.0376607	.00961	3.92	0.000	.018834	.056488	20
price	.0000428	.00002	2.47	0.014	8.8e-06	.000077	6000

To calculate the marginal effects for the linear prediction (`xb`) instead of the probability, specify `predict(xb)`. The marginal effects for the linear prediction are the coefficients themselves.

```
. mfx, predict(xb)
Marginal effects after logit
      y  = Linear prediction (predict, xb)
         =-1.0279779
```

variable	dy/dx	Std. Err.	z	P>|z|	[95%	C.I.]	X
mpg	.2338353	.06714	3.48	0.000	.102234	.365437	21.2973
price	.000266	.00012	2.28	0.022	.000038	.000495	6165.26

If there is a dummy variable as an independent variable, `mfx` calculates the discrete change as the dummy variable changes from 0 to 1.

```
. generate goodrep = 0
. replace goodrep = 1 if rep > 3
(34 real changes made)
```

(*Continued on next page*)

```
. logit foreign mpg goodrep, nolog
Logistic regression                               Number of obs   =         74
                                                  LR chi2(2)      =      26.27
                                                  Prob > chi2     =     0.0000
Log likelihood = -31.898321                       Pseudo R2       =     0.2917
```

foreign	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]	
mpg	.1079219	.0565077	1.91	0.056	-.0028311	.2186749
goodrep	2.435068	.7128444	3.42	0.001	1.037918	3.832217
_cons	-4.689347	1.326547	-3.54	0.000	-7.28933	-2.089363

```
. mfx
Marginal effects after logit
      y  = Pr(foreign) (predict)
         = .21890034
```

variable	dy/dx	Std. Err.	z	P>|z|	[95% C.I.	]	X
mpg	.0184528	.01017	1.81	0.070	-.001475	.038381	21.2973
goodrep*	.4271707	.10432	4.09	0.000	.222712	.63163	.459459

(*) dy/dx is for discrete change of dummy variable from 0 to 1

If `nodiscrete` is specified, `mfx` treats the dummy variable as continuous.

```
. mfx, nodiscrete
Marginal effects after logit
      y  = Pr(foreign) (predict)
         = .21890034
```

variable	dy/dx	Std. Err.	z	P>|z|	[95% C.I.	]	X
mpg	.0184528	.01017	1.81	0.070	-.001475	.038381	21.2973
goodrep	.4163552	.10733	3.88	0.000	.205994	.626716	.459459

◁

❑ Technical Note

By default, `mfx` uses the estimation sample to determine which independent variables are dummies. A variable is declared a dummy if its only values in the estimation sample are zero or one. This determination may be affected by the option `noesample`. For example,

```
. replace rep78=rep78-3
(69 real changes made)
. logit foreign mpg rep78 if rep78==0|rep78==1, nolog
Logistic regression                               Number of obs   =         48
                                                  LR chi2(2)      =      21.88
                                                  Prob > chi2     =     0.0000
Log likelihood = -16.050909                       Pseudo R2       =     0.4053
```

foreign	Coef.	Std. Err.	z	P>|z|	[95% Conf. Interval]	
mpg	.347422	.120328	2.89	0.004	.1115834	.5832605
rep78	2.105463	.9194647	2.29	0.022	.3033449	3.90758
_cons	-9.683393	2.876752	-3.37	0.001	-15.32172	-4.045063

```
. mfx

Marginal effects after logit
      y  = Pr(foreign) (predict)
         =   .1357189
```

variable	dy/dx	Std. Err.	z	P>|z|	[95% C.I.]		X
mpg	.0407523	.01531	2.66	0.008	.010755	.07075	20.2708
rep78*	.3027042	.14694	2.06	0.039	.01471	.590699	.375

```
(*) dy/dx is for discrete change of dummy variable from 0 to 1
```

When `noesample` is specified, the value of `rep78` is considered for all observations in the dataset. Since observations with `rep78` not equal to zero or one do exist, `mfx` will conclude that it is not a dummy variable.

```
.table rep78
```

Repair Record 1978	Freq.
-2	2
-1	8
0	30
1	18
2	11

```
. mfx, noesample

Marginal effects after logit
      y  = Pr(foreign) (predict)
         =  .19312144
```

variable	dy/dx	Std. Err.	z	P>|z|	[95% C.I.]		X
mpg	.0541372	.02043	2.65	0.008	.014089	.094185	21.2973
rep78	.3280849	.14321	2.29	0.022	.047401	.608769	.405797

The qualifiers `if` and `in` have no effect on the determination of dummy variables.

❑

Obtaining marginal effects after multiple-equation (ME) estimation

If you have not read the discussion above on using `mfx` after SE estimations, please do so. As a general introduction to the ME models, the following examples will demonstrate `mfx` after `heckman` and `mlogit`.

(*Continued on next page*)

▷ Example 2

```
. use http://www.stata-press.com/data/r10/auto, clear
(1978 Automobile Data)

. heckman mpg weight length, sel(foreign = displacement) nolog
Heckman selection model                         Number of obs      =        74
(regression model with sample selection)        Censored obs       =        52
                                                Uncensored obs     =        22

                                                Wald chi2(2)       =      7.27
Log likelihood = -87.58426                      Prob > chi2        =    0.0264

------------------------------------------------------------------------------
             |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
mpg          |
      weight |  -.0039923   .0071948    -0.55   0.579    -.0180939    .0101092
      length |  -.1202545   .2093074    -0.57   0.566    -.5304895    .2899805
       _cons |   56.72567   21.68463     2.62   0.009     14.22458    99.22676
-------------+----------------------------------------------------------------
foreign      |
displacement |  -.0250297   .0067241    -3.72   0.000    -.0382088   -.0118506
       _cons |   3.223625   .8757406     3.68   0.000     1.507205    4.940045
-------------+----------------------------------------------------------------
     /athrho |  -.9840858   .8112212    -1.21   0.225     -2.57405    .6058785
    /lnsigma |   1.724306   .2794524     6.17   0.000     1.176589    2.272022
-------------+----------------------------------------------------------------
         rho |  -.7548292    .349014                     -.9884463    .5412193
       sigma |   5.608626   1.567344                      3.243293    9.698997
      lambda |  -4.233555   3.022645                     -10.15783    1.690721
------------------------------------------------------------------------------
LR test of indep. eqns. (rho = 0):   chi2(1) =     1.37   Prob > chi2 = 0.2413
------------------------------------------------------------------------------
```

heckman estimated two equations, mpg and foreign; see [R] **heckman**. Two of the prediction options after heckman are the expected value of the dependent variable and the probability of being observed. To obtain the marginal effects of all the independent variables for the expected value of the dependent variable, we specify predict(yexpected) with mfx.

```
. mfx, predict(yexpected)

Marginal effects after heckman
      y  = E(mpg*|Pr(foreign)) (predict, yexpected)
         = .56522778
------------------------------------------------------------------------------
variable |      dy/dx    Std. Err.     z    P>|z|  [    95% C.I.   ]      X
---------+--------------------------------------------------------------------
  weight |  -.0001725      .00041   -0.42   0.675  -.000979  .000634   3019.46
  length |  -.0051953      .01002   -0.52   0.604   -.02483   .01444   187.932
displa~t |  -.0340055      .02793   -1.22   0.223  -.088739  .020728   197.297
------------------------------------------------------------------------------
```

To calculate the marginal effects for the probability of being observed, we specify predict(psel) with mfx. Since only the independent variables in equation foreign affect the probability of being observed, some of the marginal effects will be zero. Using the option varlist(*varlist*) with mfx will restrict the calculation of marginal effects to the independent variables in the *varlist*.

```
. mfx, predict(psel)

Marginal effects after heckman
      y  = Pr(foreign) (predict, psel)
         =  .04320292
```

variable	dy/dx	Std. Err.	z	P>|z|	[95% C.I.	]	X
weight	0	0	.	.	0	0	3019.46
length	0	0	.	.	0	0	187.932
displa~t	-.0022958	.00153	-1.50	0.133	-.005287	.000696	197.297

```
. mfx, predict(psel) varlist(displacement)

Marginal effects after heckman
      y  = Pr(foreign) (predict, psel)
         =  .04320292
```

variable	dy/dx	Std. Err.	z	P>|z|	[95% C.I.	]	X
displa~t	-.0022958	.00153	-1.50	0.133	-.005287	.000696	197.297

◁

▷ Example 3

`predict` after `mlogit`, unlike most other estimation commands, can predict multiple new variables by issuing `predict` only once; see [R] **mlogit**. To calculate the marginal effects for the probability of more than one outcome, we run `mfx` separately for each outcome.

```
. mlogit rep78 mpg, nolog

Multinomial logistic regression                   Number of obs   =         69
                                                  LR chi2(4)      =      15.88
                                                  Prob > chi2     =     0.0032
Log likelihood = -85.752375                       Pseudo R2       =     0.0847
```

rep78	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
1						
mpg	.0708122	.1471461	0.48	0.630	-.2175888	.3592132
_cons	-4.137144	3.15707	-1.31	0.190	-10.32489	2.050599
2						
mpg	-.0164251	.0926724	-0.18	0.859	-.1980597	.1652095
_cons	-1.005118	1.822129	-0.55	0.581	-4.576426	2.566189
4						
mpg	.0958626	.0633329	1.51	0.130	-.0282676	.2199927
_cons	-2.474187	1.341131	-1.84	0.065	-5.102756	.154381
5						
mpg	.2477469	.0764076	3.24	0.001	.0979908	.397503
_cons	-6.653164	1.841793	-3.61	0.000	-10.26301	-3.043316

(rep78==3 is the base outcome)

```
. mfx, predict(outcome(1))

Marginal effects after mlogit
      y  = Pr(rep78==1) (predict, outcome(1))
         =  .03233059
```

variable	dy/dx	Std. Err.	z	P>|z|	[95%	C.I.]	X
mpg	.0004712	.0045	0.10	0.917	-.00835	.009292	21.2899

◁

Specifying the evaluation points

By default, `mfx` evaluates the marginal effects at the means of the independent variables. To evaluate elsewhere, you would specify the option `at()` with `mfx`. This option allows several different syntaxes.

▷ Example 4

Using the *numlist* and *matname* syntax, we must specify the evaluation points in the same order as the variables in the preceding estimation command, that is, from left to right, without repetition.

```
. sureg (price disp weight) (mpg foreign disp)

Seemingly unrelated regression
```

Equation	Obs	Parms	RMSE	"R-sq"	chi2	P
price	74	2	2466.937	0.2909	29.98	0.0000
mpg	74	2	4.061588	0.5004	74.03	0.0000

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
price						
displacement	2.660349	7.043538	0.38	0.706	-11.14473	16.46543
weight	1.747666	.8322723	2.10	0.036	.1164423	3.37889
_cons	363.3701	1441.681	0.25	0.801	-2462.273	3189.014
mpg						
foreign	-.6825672	1.30813	-0.52	0.602	-3.246455	1.881321
displacement	-.0465529	.0065556	-7.10	0.000	-.0594017	-.033704
_cons	30.68498	1.632312	18.80	0.000	27.4857	33.88425

```
. mfx, predict(xb) at(200 3000 0.5)

Marginal effects after sureg
      y  = Linear prediction (predict, xb)
         =  6138.4383
```

variable	dy/dx	Std. Err.	z	P>|z|	[95%	C.I.]	X
displa~t	2.660349	7.04354	0.38	0.706	-11.1447	16.4654	200
weight	1.747666	.83227	2.10	0.036	.116442	3.37889	3000
foreign*	0	0	.	.	0	0	.5

(*) dy/dx is for discrete change of dummy variable from 0 to 1

◁

❑ Technical Note

When using the *numlist* or *matname* syntax together with `varlist()`, you must specify values for all the independent variables, not just those for which marginal effects will be calculated. These values are used in the estimation of the marginal effects, and although they don't display in the output of `mfx`, they are included in the saved results.

```
. probit foreign mpg price, nolog
Probit regression                                 Number of obs   =         74
                                                  LR chi2(2)      =      17.53
                                                  Prob > chi2     =     0.0002
Log likelihood = -36.266068                       Pseudo R2       =     0.1947

------------------------------------------------------------------------------
     foreign |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         mpg |   .1404876   .0373595     3.76   0.000     .0672644    .2137108
       price |   .0001571   .0000641     2.45   0.014     .0000315    .0002827
       _cons |  -4.592058   1.115907    -4.12   0.000    -6.779195   -2.404921
------------------------------------------------------------------------------

. capture noisily mfx, at(6000) varlist(price)
numlist too short in at()
. mfx, at(20 6000) varlist(price)
Marginal effects after probit
      y  = Pr(foreign) (predict)
         =   .2005512
------------------------------------------------------------------------------
variable |      dy/dx    Std. Err.     z    P>|z|  [    95% C.I.   ]      X
---------+--------------------------------------------------------------------
   price |   .0000441      .00002    2.62   0.009   .000011  .000077      6000
------------------------------------------------------------------------------

. matrix list e(Xmfx_X)
e(Xmfx_X)[1,2]
      mpg  price
r1     20   6000
```

❑

Obtaining three forms of elasticities

`mfx` can also be used to obtain all three forms of elasticities.

option	elasticity
eyex	$\partial \log y / \partial \log x$
dyex	$\partial y / \partial \log x$
eydx	$\partial \log y / \partial x$

▷ Example 5

We fit a regression model with the auto dataset. The marginal effects for the predicted value y after a `regress` are the same as the coefficients. To obtain the elasticities of form $\partial \log y / \partial \log x$, we specify the `eyex` option:

```
. regress mpg weight length

      Source |       SS       df       MS              Number of obs =      74
-------------+------------------------------           F(  2,    71) =   69.34
       Model |  1616.08062     2  808.040312           Prob > F      =  0.0000
    Residual |  827.378835    71   11.653223           R-squared     =  0.6614
-------------+------------------------------           Adj R-squared =  0.6519
       Total |  2443.45946    73  33.4720474           Root MSE      =  3.4137

------------------------------------------------------------------------------
         mpg |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
      weight |  -.0038515    .001586    -2.43   0.018    -.0070138   -.0006891
      length |  -.0795935   .0553577    -1.44   0.155    -.1899736    .0307867
       _cons |   47.88487    6.08787     7.87   0.000       35.746    60.02374
------------------------------------------------------------------------------

. mfx, eyex

Elasticities after regress
      y  = Fitted values (predict)
         = 21.297297
------------------------------------------------------------------------------
variable |      ey/ex    Std. Err.     z    P>|z|  [    95% C.I.   ]      X
---------+--------------------------------------------------------------------
  weight |  -.5460497      .22509   -2.43   0.015  -.987208 -.104891   3019.46
  length |  -.7023518      .48867   -1.44   0.151  -1.66012  .255414   187.932
------------------------------------------------------------------------------
```

The first line of the output indicates that the elasticities were calculated after a `regress` estimation. The title of the second column of the table gives the form of the elasticities, $\partial \log y / \partial \log x$, the percent change in y for a 1% change in x.

If the independent variables have been log-transformed already, we will want the elasticities of the form $\partial \log y / \partial x$ instead.

```
. generate lnweight = ln(weight)

. generate lnlength = ln(length)

. regress mpg lnweight lnlength
      Source |       SS       df       MS              Number of obs =      74
-------------+------------------------------           F(  2,    71) =   74.00
       Model |  1651.28916     2  825.644581           Prob > F      =  0.0000
    Residual |  792.170298    71  11.1573281           R-squared     =  0.6758
-------------+------------------------------           Adj R-squared =  0.6667
       Total |  2443.45946    73  33.4720474           Root MSE      =  3.3403

------------------------------------------------------------------------------
         mpg |      Coef.   Std. Err.      t    P>|t|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    lnweight |   -13.5974   4.692504    -2.90   0.005    -22.95398   -4.240811
    lnlength |  -9.816726   10.40316    -0.94   0.349    -30.56005    10.92659
       _cons |   181.1196   22.18429     8.16   0.000     136.8853    225.3538
------------------------------------------------------------------------------

. mfx, eydx

Elasticities after regress
      y  = Fitted values (predict)
         = 21.297297
------------------------------------------------------------------------------
variable |      ey/dx    Std. Err.     z    P>|z|  [    95% C.I.   ]      X
---------+--------------------------------------------------------------------
lnweight |  -.6384565      .22064   -2.89   0.004   -1.0709 -.206009   7.97875
lnlength |  -.4609376      .48855   -0.94   0.345  -1.41847  .496594   5.22904
------------------------------------------------------------------------------
```

Although the interpretation is the same, the results for `eyex` and `eydx` differ since we are fitting different models.

If the dependent variable were log-transformed, we would specify `dyex` instead. ◁

Using mfx after nl

You must specify the independent variables by using the `variables()` option when using the interactive version of `nl` to obtain marginal effects. Otherwise, `mfx` has no way of distinguishing the independent variables from the parameters of your model and will therefore exit with an error message.

Instead of typing

```
. nl (mpg = {b0} + {b1}*gear^{b2=1})
```

type

```
. nl (mpg = {b0} + {b1}*gear^{b2=1}), variables(gear)
```

If you use the programmed substitutable expression or function evaluator program versions of `nl`, you do not need to use the `variables()` option.

Saved Results

In addition to the `e()` results from the preceding estimation, `mfx` saves the following in `e()`:

Scalars	
`e(Xmfx_y)`	value of y given X
`e(Xmfx_off)`	value of mean of the offset variable or log of the exposure variable
`e(Xmfx_off#)`	value of mean of the offset variable for equation #
Macros	
`e(Xmfx_type)`	`dydx`, `eyex`, `eydx` or `dyex`
`e(Xmfx_discrete)`	`discrete` or `nodiscrete`
`e(Xmfx_cmd)`	`mfx`
`e(Xmfx_label_p)`	label for prediction in output
`e(Xmfx_predict)`	*predict_option* specified in `predict()`
`e(Xmfx_dummy)`	corresponding to independent variables; `1` means dummy, `0` means continuous
`e(Xmfx_variables)`	corresponding to independent variables; `1` means marginal effect calculated, `0` otherwise
`e(Xmfx_method)`	`linear` or `nonlinear`
Matrices	
`e(Xmfx_dydx)`	marginal effects
`e(Xmfx_se_dydx)`	standard errors of the marginal effects
`e(Xmfx_eyex)`	elasticities of form `eyex`
`e(Xmfx_se_eyex)`	standard errors of elasticities of form `eyex`
`e(Xmfx_eydx)`	elasticities of form `eydx`
`e(Xmfx_se_eydx)`	standard errors of elasticities of form `eydx`
`e(Xmfx_dyex)`	elasticities of form `dyex`
`e(Xmfx_se_dyex)`	standard errors of elasticities of form `dyex`
`e(Xmfx_X)`	values around which marginal effects (elasticities) were estimated

Methods and Formulas

mfx is implemented as an ado-file.

After an estimation, mfx calculates marginal effects (elasticities) and their standard errors. A marginal effect of a continuous independent variable x is the partial derivative, with respect to x, of the prediction function f specified in mfx's predict option; see Greene (2003, 668) for more information about marginal effects. If no prediction function is specified, the default prediction for the preceding estimation command is used. This derivative is evaluated at the values of the independent variables specified in the at() option of mfx or, if none is specified, at the default values that are the means of the independent variables. If there were any offsets in the preceding estimation, the derivative is evaluated at the means of the offset variables.

For a dummy variable—one that takes only the values zero or one in the estimation sample—a difference rather than a derivative is computed. The difference is the value of the prediction function at one, minus its value at zero.

For a continuous variable, the derivative is calculated numerically, which means that it approximates the derivative by using the following formula with an appropriately small h,

$$\frac{\partial y}{\partial x_j} = \lim_{h\to 0} \frac{f(x_1,\ldots,x_j+h,\ldots,x_p,\beta_0,\ldots,\beta_p) - f(x_1,\ldots,x_j,\ldots,x_p,\beta_0,\ldots,\beta_p)}{h} \qquad (1)$$

The delta method is used to estimate the variance of the marginal effect (Greene 2003, 70). The marginal effect is a function of only the coefficients of the model since all other variables are held constant at the values at which the marginal effect is sought.

$$\mathrm{Var}\left(\frac{\partial y}{\partial x_j}\right) = \mathbf{D}_j'\mathbf{V}\mathbf{D}_j$$

where $\mathbf{V}$ is the variance–covariance matrix from the estimation and $\mathbf{D}_j$ is the column vector whose kth entry is the partial derivative of the marginal effect of x_j, with respect to the coefficient of the kth independent variable:

$$(\mathbf{D}_j)_k = \frac{\partial}{\partial\beta_k}\,\frac{\partial y}{\partial x_j}$$

Thus to compute one standard error, the derivative of the marginal effect is computed with respect to each coefficient in the model.

Computing the derivative of a function f with respect to a variable x can be time consuming because an iterative algorithm must be used to find an appropriately small change in x for use in (1). The command mfx avoids this type of iteration as much as possible. If the independent variables and coefficients appear in the formula for the prediction function f only in the sum

$$\mathbf{x}\boldsymbol{\beta} = \beta_0 + \mathbf{x}\boldsymbol{\beta}_x = \beta_0 + \sum_{j=1}^{p} x_j\beta_j$$

a marked simplification in the computation of the marginal effects and their standard errors can be made.

An example of a prediction that satisfies this condition is the predicted probability of success following logistic:

$$f(x_1,\ldots x_p,\beta_0,\ldots,\beta_p) = \frac{\exp(\mathbf{x}\boldsymbol{\beta})}{1+\exp(\mathbf{x}\boldsymbol{\beta})}$$

An example when this condition is not satisfied is the predicted hazard ratio following `streg` without the option `noconstant`:

$$f(x_1, \ldots x_p, \beta_0, \ldots, \beta_p) = \exp\left(\mathbf{x}\boldsymbol{\beta}_x\right)$$

The constant β_0 is missing from the sum.

For this condition to be satisfied after multiple-equation estimation, say, an estimation with two equations, the variables and coefficients can only appear as part of the two sums: that of the first equation $\mathbf{x}\boldsymbol{\beta}$ and that of the second equation $\mathbf{z}\boldsymbol{\gamma}$. If the same variable appears in both equations, even the linear predictor for the first equation does not satisfy this condition: if $x_{j_0} = z_{k_0}$ for some j_0 and k_0, then

$$\mathbf{x}\boldsymbol{\beta} = \sum_{j=0}^{p} x_j\beta_j = z_{k_0}\beta_{j_0} + \sum_{j=0}^{j_0-1} x_j\beta_j + \sum_{j=j_0+1}^{p} x_j\beta_j$$

so z_{k_0} is appearing, but not as part of the sum $\mathbf{z}\boldsymbol{\gamma} = \sum_{k=0}^{q} z_k\gamma_k$.

If this condition is satisfied, the linear-form restriction has been met, and the linear method, to be described below, is used to estimate the marginal effects and their standard errors. If not, the usual method, described above, is used and is called the nonlinear method. Following a multiple-equation estimation with any independent variables common to more than one equation, the nonlinear method will always be used.

We begin our description of the linear method with the easiest case, a single-equation estimation. Using the chain rule, we can write the marginal effect as

$$\frac{\partial y}{\partial x_j} = \frac{dy}{d(\mathbf{x}\boldsymbol{\beta})}\,\frac{\partial(\mathbf{x}\boldsymbol{\beta})}{\partial x_j} = \frac{dy}{d(\mathbf{x}\boldsymbol{\beta})}\;\beta_j$$

The same derivative, $dy/d(\mathbf{x}\boldsymbol{\beta})$, is used for every x_j. To calculate it, we use the same formula in reverse. Since it doesn't matter which variable is used, we use the first one, x_1:

$$\frac{dy}{d(\mathbf{x}\boldsymbol{\beta})} = \frac{1}{\beta_1}\,\frac{\partial y}{\partial x_1}$$

Therefore, only one derivative $\partial y/\partial x_1$ needs to be calculated by the usual nonlinear method, and all marginal effects are then obtained by multiplying the derivative by the appropriate coefficient. Thus the linear method is generally much faster than the nonlinear method.

To compute the standard errors, we need the second derivatives. If j is not equal to k, using the chain rule we have

$$\frac{\partial}{\partial\beta_k}\,\frac{\partial y}{\partial x_j} = \beta_j\;x_k\;\frac{d}{d(\mathbf{x}\boldsymbol{\beta})}\,\frac{dy}{d(\mathbf{x}\boldsymbol{\beta})} \tag{2}$$

If j is equal to k, using the product rule we have

$$\frac{\partial}{\partial\beta_j}\,\frac{\partial y}{\partial x_j} = \beta_j\;x_j\;\frac{d}{d(\mathbf{x}\boldsymbol{\beta})}\,\frac{dy}{d(\mathbf{x}\boldsymbol{\beta})}\;+\;\frac{dy}{d(\mathbf{x}\boldsymbol{\beta})} \tag{3}$$

We obtain the second derivative, again using the chain rule,

$$\frac{d}{d(\mathbf{x}\boldsymbol{\beta})}\,\frac{dy}{d(\mathbf{x}\boldsymbol{\beta})} = \frac{1}{\beta_1^2}\,\frac{\partial^2 y}{\partial x_1^2}$$

Now we turn to multiple equations. The linear method will be used only when there are no variables in common between the equations. For marginal effects, the formulas developed above apply in each equation separately. For standard errors, we will consider the case of two equations. Suppose that x_j and x_k are both in the first equation and j is not equal to k. Then $\partial/\partial\beta_k(\partial y/\partial x_j)$ is calculated as in (2) above, since x_j does not appear in the second equation, making $\partial(\mathbf{z}\boldsymbol{\gamma})/\partial x_j = 0$, and β_k is in equation one, making $\partial(\mathbf{z}\boldsymbol{\gamma})/\partial\beta_k = 0$. If x_j and x_k are both in the first equation and j is equal to k, we use the product rule and obtain the same as (3) above. Now suppose that x_j is in equation one and z_k is in equation two. Then

$$\frac{\partial}{\partial\gamma_k}\,\frac{\partial y}{\partial x_j} = \beta_j\ z_k\ \frac{d}{d(\mathbf{z}\boldsymbol{\gamma})}\ \frac{dy}{d(\mathbf{x}\boldsymbol{\beta})}$$

The second derivative is calculated by $d/d(\mathbf{z}\boldsymbol{\gamma})\{dy/d(\mathbf{x}\boldsymbol{\beta})\} = 1/(\gamma_1\beta_1)\{\partial^2 y/(\partial z_1 \partial x_1)\}$.

For multiple equations, it is possible to have an equation that is a constant only, such as an ancillary parameter. Then it is not possible to obtain $d^2y/d(\mathbf{z}\boldsymbol{\gamma})d(\mathbf{x}\boldsymbol{\beta})$ by converting to derivatives with respect to the independent variables, so it is evaluated directly. For example, if (2) had only a constant term so that $\mathbf{z}\boldsymbol{\gamma} = \gamma_0$, then $d^2y/d(\mathbf{z}\boldsymbol{\gamma})d(\mathbf{x}\boldsymbol{\beta}) = 1/\beta_1\ \partial/\partial x_1(dy/d\gamma_0)$.

References

Bartus, T. 2005. Estimation of marginal effects using margeff. *Stata Journal* 5: 309–329.

Baum, C. F. 2006. *An Introduction to Modern Econometrics Using Stata*. College Station, TX: Stata Press.

Greene, W. H. 2003. *Econometric Analysis*. 5th ed. Upper Saddle River, NJ: Prentice Hall.

Also See

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

[R] **predict** — Obtain predictions, residuals, etc., after estimation