

The lp-mfx package, version 1.0

Allin Cottrell

April, 2019

1 Introduction

This package computes marginal effects for a certain set of limited dependent variable estimators, namely binary logit and probit, ordered logit and probit, multinomial logit and logistic regression. See the `gretl` commands `logit`, `probit` and `logistic` for details on these. It can be called from the **Analysis** menu in a GUI window displaying a supported model (see section 5), or via scripting (see section 4).

The effects computed are derivatives with respect to the regressors (or the outcome of a discrete change from 0 to 1 in the case of 0/1 dummy regressors). For binary logit and probit the numerator or target of these derivatives is the estimated probability that the dependent variable equals 1; in the ordered and multinomial cases it is the estimated probability of each discrete outcome in turn; and for logistic regression it is the predicted value of the untransformed dependent variable. In most cases (details below) standard errors for these effects are available, computed via the delta method, along with z -scores and P -values for the null hypothesis of no effect.

2 Modes of marginal effects

Three modes are supported as follows:

1. Marginal effects at the means of the regressors.
2. Average marginal effects.
3. Marginal effects at each observation in the sample range.

The first of these modes is fast, since only one calculation is required for each regressor/outcome pair. Average marginal effects are more complex since a calculation must be performed for each observation used in estimation. If the number of observations is large this can take a while; the process can be speeded up by choosing not to produce standard errors for the effects.

Per-observation marginal effects are available only when the dependent variable is “unitary” (either binary or logistic). Standard errors are not produced in this case.

3 Primary and non-primary regressors

Besides detecting, and giving special treatment to, 0/1 dummies among the regressors, `lp-mfx` attempts to identify two common cases of “non-primary” regressors, namely

- Regressor x_j is the square of regressor x_i (quadratic specification).
- Regressor x_k is the product of regressors x_i and x_j (interaction term).

If the model specification is, to take a simple binary probit example,

$$P(y = 1 | x, \beta) = \Phi(\beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + \beta_3 x_2 + \beta_4 x_3 + \beta_5 x_2 x_3) = \Phi(x\beta)$$

then the marginal effects for the primary regressors x_1 , x_2 and x_3 are

$$\begin{aligned}\partial P / \partial x_1 &= \phi(x\beta) \cdot (\beta_1 + 2\beta_2 x_1) \\ \partial P / \partial x_2 &= \phi(x\beta) \cdot (\beta_3 + \beta_5 x_3) \\ \partial P / \partial x_3 &= \phi(x\beta) \cdot (\beta_4 + \beta_5 x_2)\end{aligned}$$

while the non-primary terms, x_1^2 and $x_2 x_3$, do not possess marginal effects of their own.

To ensure that such non-primary terms are detected correctly, it is advisable to create them within gretl; any of the following commands will give the series in question labels that make them identifiable:

```
series x1sq = x1*x1
series x1p2 = x^2
square x1 # produces sq_x1
series interact = x2*x3
series iact = x2 * x3
```

The names given to the series don't matter, only the expressions used to generate them; and in the expressions spaces or their absence don't matter.

If you are using a third-party dataset that includes non-primary terms that are not identified as such by descriptive labels, as an alternative to recreating the relevant series you may use the utility function `lp_mfx_fixlist()` supplied by `lp-mfx`. This takes a named list argument. Each series in the list is checked, by brute-force comparison of data values, and if it is the square of another series in the list, or the product of any other two series, this fact is recorded in its label. As an example of usage, suppose we are using as regressors series `x1` to `x5`; then we might do

```
list X = x1 x2 x3 x4 x5
lp_mfx_fixlist(X)
probit y const X
```

after which we can be confident that `lp-mfx` will deal correctly with any squares or interactions among the regressors in `X`.

Note that more elaborate cases of non-primary regressors are not handled automatically at present (for example, cubes or higher powers of a regressor, or interaction terms that are the products of three or more variables).

4 Scripting basics

After issuing the command `include lp-mfx.gfn` the following primary function is available:

```
bundle lp_mfx (const bundle mod,
               int mode[1:3:1],
               bool skip_se[0])
```

The `mod` argument (the only required one) can be obtained after estimation of a suitable model via the `$model` accessor, as in

```
logit y 0 x1 x2 x3
bundle mod = $model
```

The optional second argument—an integer value 1, 2 or 3—can be used to select the mode of the marginal effects, as set out in section 2. The default is 1, giving effects at the means.

There's also a variant of this function which allows you to supply a specific vector of values of the independent variables at which marginal effects should be evaluated:

```
bundle lp_mfx_atx (const bundle mod,
                  const matrix x,
                  bool skip_se[0])
```

The argument `x` must be a row vector of length equal to that of the model's parameter vector (including the constant, if present).

The optional third argument to `lp_mfx` and `lp_mfx_atx` can be used to suppress the computation of standard errors.

The return value from these functions is a bundle containing (primarily) one or more matrices holding the marginal effects. The results can be displayed in tabular form via the function `lp_mfx_print`, which takes as its single argument a "pointer to" the bundle:

```
bundle mf = lp_mfx($model)
lp_mfx_print(&mf)
```

5 GUI access

On installing the `lp_mfx` package you should get the choice of letting it attach to the **Analysis** menu in gretl model windows under the item **Marginal effects**. In that case when you estimate a suitable model via the graphical interface you can view the marginal effects by clicking on this item.

6 Examples

We confine ourselves below to datasets supplied with gretl, so it should be possible to follow along if you wish. Here's a simple binary probit example, using the `mroz87` dataset which is to do with women's labor force participation:

```
include lp-mfx.gfn
open mroz87.gdt
probit LFP 0 KL6 K618 WA WE
bundle b = lp_mfx($model, 1)
lp_mfx_print(&b)
bundle b = lp_mfx($model, 2)
lp_mfx_print(&b)
```

which produces (partial output):

```
Binary probit marginal effects at means
LFP = 1, Pr = 0.5733
```

	dp/dx	s.e.	z	pval	xbar
KL6	-0.34753	0.044200	-7.8627	3.7585e-15	0.23772
K618	-0.021842	0.015719	-1.3895	0.16468	1.3533
WA	-0.015009	0.0029216	-5.1370	2.7910e-07	42.538
WE	0.047075	0.0086964	5.4132	6.1910e-08	12.287

```
Binary probit average marginal effects
```

	dp/dx	s.e.	z	pval
KL6	-0.31238	0.034676	-9.0084	2.0914e-19
K618	-0.019633	0.014082	-1.3941	0.16328
WA	-0.013490	0.0024938	-5.4097	6.3138e-08
WE	0.042314	0.0073701	5.7412	9.3981e-09

In this case we see that the marginal effects at the regressor means and the average marginal effects are not very different. Both show that women are less likely to participate in the labor force (LFP) when

they have children under the age of 6 (KL6), and when they are older (WA), but more likely to participate when their education level (WE) is higher. The example is simple in that all the regressors are “primary” and none is a dummy variable.

As a variant on the above, we might ask what the marginal effects look like for a 30-year old woman with 15 years of education and no children. The additional command is

```
matrix x = {1, 0, 0, 30, 15}
bundle b = lp_mfx_atx($model, x)
lp_mfx_print(&b)
```

and the output is

```
Binary probit marginal effects at specified x
LFP = 1, Pr = 0.8991
```

	dp/dx	s.e.	z	pval	xval
KL6	-0.15658	0.026928	-5.8148	6.0712e-09	0.0000
K618	-0.0098410	0.0061266	-1.6063	0.10821	0.0000
WA	-0.0067621	0.00084402	-8.0118	1.1300e-15	30.000
WE	0.021210	0.0047280	4.4860	7.2563e-06	15.000

A second example uses the `keane` dataset. The dependent variable, `status`, has three unordered values: 1 indicates that the respondent (a young man) is in school, 2 that he is at home, and 3 that he is in work. The `lp-mfx` output shows marginal effects on the probability of each outcome but here we focus on the “in school” outcome. The script

```
include lp-mfx.gfn
open keane.gdt -q
smpl (year==87) --restrict
logit status 0 educ exper expersq black --multinomial
bundle b = lp_mfx($model)
lp_mfx_print(&b)
bundle b = lp_mfx($model, 2)
lp_mfx_print(&b)
```

produces (in part)

```
Multinomial logit marginal effects at means
note: dp/dx based on discrete change for black
```

```
Outcome 1: (status = 1, Pr = 0.0212)
```

	dp/dx	s.e.	z	pval	xbar
educ	0.0073312	0.0015143	4.8412	1.2907e-06	12.549
exper	-0.0054125	0.0019287	-2.8063	0.0050119	3.4403
black	-0.0073531	0.0055632	-1.3217	0.18625	0.37973

```
Multinomial logit average marginal effects
note: dp/dx based on discrete change for black
```

```
Outcome 1: (status = 1)
```

	dp/dx	s.e.	z	pval
educ	0.017379	0.0029057	5.9808	2.2201e-09
exper	-0.019438	0.0042472	-4.5766	4.7269e-06
black	-0.017991	0.011646	-1.5449	0.12238

In this case the average marginal effects are larger and more sharply estimated than the effects at the means. Note that the regressor `expersq`, the square of `exper` (years of work experience), is correctly

excluded from the list of marginal effects: the effect shown for `exper` takes account of the quadratic specification.

Anyone interested in a comparison with **Stata** (and in possession of a copy of said software) can append the following to the basic `gretl` script

```
foreign language=stata --send-data
  quietly mlogit status educ exper c.exper#c.exper i.black
  margins, dydx(educ exper black) atmeans predict(outcome(1))
  margins, dydx(educ exper black) predict(outcome(1))
end foreign
```

to reveal that the answers are the same as those shown above. Note that special syntax is required on the `mlogit` command line to get the square of `exper` recognized as such, and to get `black` treated as a 0/1 dummy rather than as continuous. Also the `margins` command produces average marginal effects by default but the `atmeans` option can be used to get the effects at the means.