Factor Variables and Marginal Effects in Stata 11

Christopher F Baum

Boston College and DIW Berlin

January 2010

Using factor variables

One of the biggest innovations in Stata version 11 is the introduction of *factor variables*. Just as Stata's time series operators allow you to refer to lagged variables (`L.` or differenced variables (`D.`), the `i.` operator allows you to specify factor variables for any non-negative integer-valued variable in your dataset.

In the `auto.dta` dataset, where `rep78` takes on values 1...5, you could `list rep78 i.rep78`, or `summarize i.rep78`, or `regress mpg i.rep78`. Each one of those commands produces the appropriate indicator variables 'on-the-fly': not as permanent variables in your dataset, but available for the command.

For the `list` command, the variables will be named `1b.rep78`, `2.rep78` `...5.rep78`. The `b.` is the base level indicator, by default assigned to the smallest value. You can specify other base levels, such as the largest value, the most frequent value, or a particular value.

For the `summarize` command, only levels 2...5 will be shown; the base level is excluded from the list. Likewise, in a regression on `i.rep78`, the base level is the variable excluded from the regressor list to prevent perfect collinearity. The conditional mean of the excluded variable appears in the constant term.

Interaction effects

If this was the only feature of factor variables (being instantiated when called for) they would not be very useful. The real advantage of these variables is the ability to define `interaction effects` for both integer-valued and continuous variables. For instance, consider the indicator `foreign` in the `auto` dataset. We may use a new operator, `#`, to define an interaction:

```
regress mpg i.rep78 i.foreign i.rep78#i.foreign
```

All combinations of the two categorical variables will be defined, and included in the regression as appropriate (omitting base levels and cells with no observations).

In fact, we can specify this model more simply: rather than

```
regress mpg i.rep78 i.foreign i.rep78#i.foreign
```

we can use the *factorial interaction* operator, `##`:

```
regress mpg i.rep78##i.foreign
```

which will provide exactly the same regression, producing all first-level and second-level interactions. Interactions are not limited to pairs of variables; up to eight factor variables may be included.

Furthermore, factor variables may be interacted with continuous variables to produce analysis of covariance models. The continuous variables are signalled by the new `c.` operator:

```
regress mpg i.foreign i.foreign#c.displacement
```

which essentially estimates two regression lines: one for domestic cars, one for foreign cars. Again, the factorial operator could be used to estimate the same model:

```
regress mpg i.foreign##c.displacement
```

As we will see in discussing marginal effects, it is very advantageous to use this syntax to describe interactions, both among categorical variables and between categorical variables and continuous variables. Indeed, it is likewise useful to use the same syntax to describe squared (and cubed...) terms:

```
regress mpg i.foreign c.displacement c.displacement#c.displacement
```

In this model, we allow for an intercept shift for `foreign`, but constrain the slopes to be equal across foreign and domestic cars. However, by using this syntax, we may ask Stata to calculate the marginal effect $\partial mpg/\partial displacement$, taking account of the squared term as well, as Stata understands the mathematics of the specification in this explicit form.

Computing marginal effects

With the introduction of factor variables in Stata 11, a powerful new command has been added: `margins`, which supersedes earlier versions' `mfx` and `adjust` commands. Those commands remain available, but the new command has many advantages. Like those commands, `margins` is used after an estimation command.

In the simplest case, `margins` applied after a simple one-way ANOVA estimated with `regress i.rep78`, with `margins i.rep78`, merely displays the conditional means for each category of `rep78`.

```
. regress mpg i.rep78
```

Source	SS	df	MS
Model	549.415777	4	137.353944
Residual	1790.78712	64	27.9810488
Total	2340.2029	68	34.4147485

Number of obs	= 69
F(4, 64)	= 4.91
Prob > F	= 0.0016
R-squared	= 0.2348
Adj R-squared	= 0.1869
Root MSE	= 5.2897

mpg	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
rep78						
2	-1.875	4.181884	-0.45	0.655	-10.22927	6.479274
3	-1.566667	3.863059	-0.41	0.686	-9.284014	6.150681
4	.6666667	3.942718	0.17	0.866	-7.209818	8.543152
5	6.363636	4.066234	1.56	0.123	-1.759599	14.48687
_cons	21	3.740391	5.61	0.000	13.52771	28.47229

```
. margins i.rep78
Adjusted predictions                            Number of obs   =         69
Model VCE    : OLS
Expression   : Linear prediction, predict()
```

	Margin	Delta-method Std. Err.	z	P>|z|	[95% Conf.	Interval]
rep78						
1	21	3.740391	5.61	0.000	13.66897	28.33103
2	19.125	1.870195	10.23	0.000	15.45948	22.79052
3	19.43333	.9657648	20.12	0.000	17.54047	21.3262
4	21.66667	1.246797	17.38	0.000	19.22299	24.11034
5	27.36364	1.594908	17.16	0.000	24.23767	30.4896

We now estimate a model including both displacement and its square:

```
. regress mpg i.foreign c.displacement c.displacement#c.displacement
```

Source	SS	df	MS		
Model	1416.01205	3	472.004018	Number of obs	= 74
Residual	1027.44741	70	14.6778201	F(3, 70)	= 32.16
				Prob > F	= 0.0000
				R-squared	= 0.5795
				Adj R-squared	= 0.5615
Total	2443.45946	73	33.4720474	Root MSE	= 3.8312

mpg	Coef.	Std. Err.	t	P>|t|	[95% Conf.	Interval]
1.foreign	-2.88953	1.361911	-2.12	0.037	-5.605776	-.1732833
displacement	-.1482539	.0286111	-5.18	0.000	-.2053169	-.0911908
c. displacement# c. displacement	.0002116	.0000583	3.63	0.001	.0000953	.0003279
_cons	41.40935	3.307231	12.52	0.000	34.81328	48.00541

`margins` can then properly evaluate the regression function for domestic and foreign cars at selected levels of `displacement`:

```
. margins i.foreign, at(displacement=(100 300))
Adjusted predictions                              Number of obs   =         74
Model VCE    : OLS
Expression   : Linear prediction, predict()
1._at        : displacement    =         100
2._at        : displacement    =         300
```

	Margin	Delta-method Std. Err.	z	P>|z|	[95% Conf.	Interval]
_at#foreign						
1 0	28.69991	1.216418	23.59	0.000	26.31578	31.08405
1 1	25.81038	.8317634	31.03	0.000	24.18016	27.44061
2 0	15.97674	.7014015	22.78	0.000	14.60201	17.35146
2 1	13.08721	1.624284	8.06	0.000	9.903668	16.27074

In earlier versions of Stata, calculation of marginal effects in this model required some programming due to the nonlinear term `displacement`. Using `margins, dydx`, that is now simple. Furthermore, and most importantly, the default behavior of `margins` is to calculate average marginal effects (AMEs) rather than marginal effects at the average (MAE) or at some other point in the space of the regressors. In Stata 10, the user-written command `margeff` (Tamas Bartus, on the SSC Archive) was required to compute AMEs.

Current practice favors the use of AMEs: the computation of each observation's marginal effect with respect to an explanatory factor, averaged over the estimation sample, to the computation of MAEs (which reflect an average individual: e.g. a family with 2.3 children).

We illustrate by computing average marginal effects (AMEs) for the prior regression:

```
. margins, dydx(foreign displacement)
Average marginal effects                        Number of obs   =         74
Model VCE    : OLS
Expression   : Linear prediction, predict()
dy/dx w.r.t. : 1.foreign displacement
```

	dy/dx	Delta-method Std. Err.	z	P>|z|	[95% Conf.	Interval]
1.foreign	-2.88953	1.361911	-2.12	0.034	-5.558827	-.2202327
displacement	-.0647596	.007902	-8.20	0.000	-.0802473	-.049272

Note: dy/dx for factor levels is the discrete change from the base level.

Alternatively, we may compute elasticities or semi-elasticities:

```
. margins, eyex(displacement) at(displacement=(100(100)400))
Average marginal effects                        Number of obs   =         74
Model VCE    : OLS
Expression   : Linear prediction, predict()
ey/ex w.r.t. : displacement
1._at        : displacement    =         100
2._at        : displacement    =         200
3._at        : displacement    =         300
4._at        : displacement    =         400
```

	ey/ex	Delta-method Std. Err.	z	P>|z|	[95% Conf.	Interval]
displacement _at						
1	-.3813974	.0537804	-7.09	0.000	-.486805	-.2759898
2	-.6603459	.0952119	-6.94	0.000	-.8469578	-.473734
3	-.4261477	.193751	-2.20	0.028	-.8058926	-.0464028
4	.5613844	.4817784	1.17	0.244	-.3828839	1.505653

Consider a model where we specify a factorial interaction between categorical and continuous covariates:

```
regress mpg i.foreign i.rep78##c.displacement
```

In this specification, each level of `rep78` has its own intercept and slope, whereas `foreign` only shifts the intercept term.

We may compute elasticities or semi-elasticities with the `over` option of `margins` for all combinations of `foreign` and `rep78`:

```
. margins, eyex(displacement) over(foreign rep78)
Average marginal effects                        Number of obs   =         69
Model VCE    : OLS
Expression   : Linear prediction, predict()
ey/ex w.r.t. : displacement
over         : foreign rep78
```

	Delta-method ey/ex	Std. Err.	z	P>|z|	[95% Conf.	Interval]
displacement						
foreign# rep78						
0 1	-.7171875	.5342	-1.34	0.179	-1.7642	.3298253
0 2	-.5953046	.219885	-2.71	0.007	-1.026271	-.1643379
0 3	-.4620597	.0999242	-4.62	0.000	-.6579077	-.2662118
0 4	-.6327362	.1647866	-3.84	0.000	-.955712	-.3097604
0 5	-.8726071	.0983042	-8.88	0.000	-1.06528	-.6799345
1 3	-.128192	.0228214	-5.62	0.000	-.1729213	-.0834628
1 4	-.1851193	.0380458	-4.87	0.000	-.2596876	-.110551
1 5	-1.689962	.3125979	-5.41	0.000	-2.302642	-1.077281

The `margins` command has many other capabilities which we will not discuss here. Perusal of the Stata 11 reference manual article on `margins` would be useful to explore its additional features.