Detailed Explanation of Stata Code for a Marginal Effect Plot for X

Below, I go through the Stata code for creating a marginal effect plot for X for an interaction model with the following basic form:

$$Y = \beta_0 + \beta_1 X + \beta_2 Z + \beta_3 XZ + \epsilon. \tag{1}$$

```
version 11.0
#delimit ;
log using "C:\matt\publications\jop2\webpage\interaction1.log", replace;
set more off;
```

The first line simply indicates the version of Stata that is being used. The second line tells Stata that all command lines are finished with a semi-colon. The third line opens a log file and indicates where it will be saved. The fourth line tells Stata not to pause when running the do-file.

```
*     ************************************************************** *;
*     ************************************************************** *;
*       File-Name:      interaction1.do                               *;
*       Date:           1/6/2013                                      *;
*       Author:         MG                                            *;
*       Purpose:        Do-file to produce a marginal effect plot for   *;
*                       X when the interaction model is:              *;
*                       Y = b0 + b1X + b2Z + b3XZ + e                 *;
*       Input File:     alexseev.dta                                  *;
*       Output File:    interaction1.log                              *;
*       Data Output:    None                                          *;
*       Previous file:  None                                          *;
*       Machine:        laptop                                        *;
*     ************************************************************** *;
*     ************************************************************** *;
```

This part of the code is how I like to start all do-files and is not strictly necessary to create the sample marginal effect plot. It indicates the name of the do-file, the date on which it was written, the author, the purpose of the do-file, the data used with the do-file, the name of the associated log file, the name of any new data set that is created by the do-file, the name of any previous do-file or data set, and where the do-file was run.

```
use "C:\matt\publications\jop2\webpage\alexseev.dta", clear;
```

This tells Stata what data set to open and where to find it. It also clears any existing data in Stata 's memory.

```
set obs 10000;
```

This command sets the number of observations (rows) in the data set to 10,000. There is nothing special about this number. You will want to calculate the marginal effect of X at multiple values of Z. If you want to calculate the marginal effect at more values than you actually have in your original data set, then you will need to increase the number of observations in the data set. This is precisely what this line of code does.

```
regress xenovote slavicshare changenonslav slavicshare_changenonslav
        inc9903 eduhi02 unemp02 apt9200 vsall03 brdcont, cluster(region);
```

This line estimates the chosen model. In this case, `xenovote` is the dependent variable, Y, `slavicshare` is our X, `changenonslav` is our Z, `slavicshare_changenonslav` is our $X \times Z$, `inc9903 eduhi02 unemp02 apt9200 vsall03 brdcont` are control variables, and `cluster(region)` indicates the use of robust standard errors clustered by `region`.

```
matrix b=e(b);
matrix V=e(V);

scalar b1=b[1,1];
scalar b3=b[1,3];

scalar varb1=V[1,1];
scalar varb3=V[3,3];

scalar covb1b3=V[1,3];

scalar list b1 b3 varb1 varb3 covb1b3;
```

The first two lines grab the elements of the estimated coefficient vector and the variance-covariance matrix that are necessary to calculate the marginal effects and confidence intervals in the marginal effect plot. The estimated coefficient vector, `b`, is $1 \times k$, where k is the number of covariates including the constant term. The first element in `b`, `b[1,1]`, is β_1, the second element, `b[1,2]` is β_2, and so on. The last element in `b` is the coefficient on the constant term, β_0. The estimated variance-covariance matrix is `V`. The diagonal elements indicate the variances of the coefficients, while the off-diagonal elements indicate the covariances of the coefficients. For example, the element `V[1,1]` indicates the variance of β_1 and the element `V[1,3]` indicates the covariance between β_1 and β_3. The remaining lines of code create (and then list) scalars that are equal to the coefficients, variances, and covariances that are needed to calculate the desired marginal effects and standard errors. You can check the listed scalars with the regression output to make sure that you have done everything correctly.

```
*     ************************************************************  *;
*       Calculate data necessary for top marginal effect plot.      *;
*     ************************************************************  *;

generate MVZ=((_n-200)/100);

replace  MVZ=. if _n>1501;

gen conbx=b1+b3*MVZ if _n<1501;

gen consx=sqrt(varb1+varb3*(MVZ^2)+2*covb1b3*MVZ) if _n<1501;

gen ax=1.96*consx;

gen upperx=conbx+ax;

gen lowerx=conbx-ax;
```

These lines calculate everything that is necessary to produce the marginal effect plot for X, i.e., `slavicshare`. The first line is important and generates a variable `MVZ` that takes on all of the values of the modifying variable Z, `changenonslav`, for which you want to calculate the marginal effect of X; the more values, the

smoother your marginal effect line will be. _n in Stata indicates the observation number starting at 1. If your modifying variable starts at 1 and increases in increments of 1, then you could simply type generate `generate MVZ=_n;`. If your modifying variable starts at 0 and increases in increments of 1, then you would need to start at `_n-1`, and so you would type `generate MVZ=_n-1;`. If you have a continuous modifying variable that starts at 0 and you want to calculate the marginal effect of X at each 0.01 increment in the modifying variable Z, then you would type `generate MVZ=((_n-1)/100);`. The modifying variable Z in this specific example starts at -1.99, and I would like to calculate the marginal effect of X at each increment of 0.01 in the modifying variable Z: `generate MVZ=((_n-200)/100);`.

The second line determines the last value of the modifying variable Z for which you want to calculate the marginal effect of X. Given how it has been constructed, `MVZ` $= 13$ when `_n` $= 1501$. Since the highest observed value of Z in my data is 13, I replace `MVZ` as `.`, or missing, for all values of `_n>1501`.

The third line calculates the marginal effect of X on Y, `conbx`, for the desired values of the modifying variable Z. The marginal effect of X for an interactive model like that shown in Eq. 1 is:

$$\frac{\partial Y}{\partial X} = \beta_1 X + \beta_3 Z. \tag{2}$$

The third line calculates this marginal effect of X for all values `MVZ`, so long as `MVZ` is less than 13.

The fourth line calculates the standard error for the marginal effect of X on Y, `consx`, for the desired values of the modifying variables of Z. The standard error for the marginal effect of X for an interactive model like that shown in Eq. 1 is:

$$\sqrt{\text{var}(\beta_1) + Z^2\text{var}(\beta_3) + 2Z\text{cov}(\beta_1\beta_3)}. \tag{3}$$

The fourth line calculates the standard error for the marginal effect of X for all values `MVZ`, so long as `MVZ` is less than 13.

The last three lines use the standard errors and marginal effects to create two-tailed 95% confidence intervals. The critical value for the t statistic if you have more than 120 degrees of freedom is 1.96. If you have fewer degrees of freedom or want to calculate confidence intervals for a different level of significance, then you will have to change the critical value for the t statistic. `lowerx` and `upperx` capture the lower and upper value, respectively, for the two-tailed 95% confidence interval around the marginal effect of X.

```
gen where=-0.045;

gen pipe = "|";

egen tag_nonslav = tag(changenonslav);
```

These lines create the necessary information if you want to include a rug plot in the marginal effect plot. Whether you want to include a rug plot will likely depend on the sample size; they are less useful when the sample size is large. Unfortunately, Stata does not have a command for producing a rug plot, and so you have to produce it manually as is the case here. The first line indicates *where* the rug plot will be located relative to the horizontal axis in the marginal effect plot. You will have to play around with this number so

that it is appropriate for your particular marginal effect plot; the value associated with `where` is related to the scale used in the marginal effect plot. The second line creates the vertical symbol that will be used in the rug plots. The last line prevents Stata from having to overwrite symbols for observations that have the same value. For more information on how to construct rug plots in Stata , see here.

```
gen yline=0;
```

This will create a horizontal line at 0 on the vertical axis. This line will enable us to easily see when the marginal effect is greater or smaller than 0.

```
graph twoway hist changenonslav, width(0.5) percent color(gs14) yaxis(2)
        ||   scatter where changenonslav if tag_nonslav,
             plotr(m(b 4)) ms(none) mlabcolor(gs5) mlabel(pipe) mlabpos(6) legend(off)
        ||   line conbx MVZ, clpattern(solid) clwidth(medium) clcolor(black) yaxis(1)
        ||   line upperx MVZ, clpattern(dash) clwidth(thin) clcolor(black)
        ||   line lowerx MVZ, clpattern(dash) clwidth(thin) clcolor(black)
        ||   line yline MVZ, clwidth(thin) clcolor(black) clpattern(solid)
        ||   ,
             xlabel(-2 0 2 4 6 8 10 12, nogrid labsize(2))
             ylabel(-0.05 0 .05 .1 .15 .2 .25, axis(1) nogrid labsize(2))
             ylabel(0 2 4 6 8 10 12 14 16, axis(2) nogrid labsize(2))
             yscale(noline alt)
             yscale(noline alt axis(2))
             xscale(noline)
             legend(off)
             xtitle("" , size(2.5)  )
             ytitle("" , axis(2) size(2.5))
             xsca(titlegap(2))
             ysca(titlegap(2))
             scheme(s2mono) graphregion(fcolor(white) ilcolor(white) lcolor(white));
```

This produces the marginal effect plot for X. The first part of the command,

```
graph twoway hist changenonslav, width(0.5) percent color(gs14) yaxis(2)
```

overlays a histogram on the marginal effect plot, sets the width of the bars, indicates that we want the bars to represent the *percent* of observations, sets the color of the bars, and indicates that the second vertical axis on the right will be associated with the histogram. The second part of the command,

```
scatter where changenonslav if tag_nonslav,
plotr(m(b 4)) ms(none) mlabcolor(gs5) mlabel(pipe) mlabpos(6) legend(off)
```

overlays the rug plot. The third part of the command,

```
line conbx   MVZ, clpattern(solid) clwidth(medium) clcolor(black) yaxis(1)
```

plots the marginal effect of X on Y across the observed values of Z, and associates it with the first vertical axis on the left. The fourth part of the command,

```
line upperx  MVZ, clpattern(dash) clwidth(thin) clcolor(black)
```

plots the upper bound of the confidence interval. The fifth part of the command,

```
line lowerx  MVZ, clpattern(dash) clwidth(thin) clcolor(black)
```

plots the lower bound of the confidence interval. The sixth part of the command,

```
line yline  MVZ,  clwidth(thin) clcolor(black) clpattern(solid)
```

plots the horizontal line at 0. The remaining lines serve various purposes, such as indicating how the axes should look.