Understanding The STATA Code for Plots With Two Modifying Variables

Below, I go through the STATA code for creating a plot that illustrates the marginal effect of some variable in an interaction model when there are two modifying variables. Specific lines of code are written in small font, whereas comments are written in larger font.

```
version 8.0
log using logfilename.log, replace
#delimit;
```

The first line of code says that we are using STATA 8.0. The second line opens a log file called 'logfilename.log' or whatever you want to call it. The third line simply tells STATA that all command lines are finished with a semi-colon.

```
* ****************************************************************** *;
* ****************************************************************** *;
* File-Name: singlemodifyingvariable.do                               *;
* Date: Month/Day/Year                                                *;
* Author: MRG                                                         *;
* Purpose: Do-file for creating figures showing the marginal effect of X on  *;
*          Y with two modifying variables where the dependent variable  *;
*          is continuous. The estimated model is:                       *;
*          Y = b0 + b1X + b2Z + b3W                                    *;
*                 + b4XZ + b5XW + b6ZW + b7XZW + epsilon                *;
* Input File: name_of_datafile.dta                                    *;
* Output File: logfilename.log                                        *;
* Data Output: None                                                   *;
* Previous file: None                                                 *;
* Machine: Office                                                     *;
* ****************************************************************** *;
* ****************************************************************** *;
```

This part of the code is not required for creating the figures. I use it so that I can keep track of what I am doing. I keep track of the do-file name, when I last edited it, who wrote it, why I wrote it, what dataset the do-file uses, what the logfile is called, what any new dataset created by the do-file is called, what the names of any previous do-files were called, and what machine I last ran the do-file on.

```
use h: name_of_datafile.dta;
```

This tells STATA the path for finding the dataset used by this do-file.

```
* ****************************************************************** *;
* Estimate Model: Y = b0 + b1X + b2Z + b3W                            *;
*                   + b4XZ + b5XW + b6ZW + b7XZW                      *;
*                   + b8Controls + epsilon                            *;
* ****************************************************************** *;

regress Y X Z W XZ XW ZW XZW Controls;
```

This is simply the estimation command. You replace the variable names Y, X, Z etc. with the names of your variables. 'Controls' simply refer to the other independent variables that are not involved in the interaction. You should enter the independent variables in this order. In other words,

the first independent variable should be the variable that you want to calculate the marginal effect of. The second independent variable should be the first modifying variable and the third independent variable should be the second modifying variable etc..

```
* ************************************************************ *;
* Generate the values of Z for which you want to calculate the  *;
* marginal effect (and standard errors) of X on Y. Z will be the X-axis  *;
* ************************************************************ *;

generate MV=((_n-1)/10);
```

This line of code is important because it indicates the values of the first modifying variable (Z) you are going to calculate the marginal effect at. It generates a variable called MV that is going to act as your primary modifying variable for the purpose of creating the plot. It starts at 0 and increases in increments of 0.1. Thus, it is a counter 0, 0.1, 0.2, 0.3 etc.. This indicates that you will be calculating the marginal effect of X across the modifying range of MV (or Z) in increments of 0.1. You may want to calculate the marginal effect of X for different increments in MV (or Z). To do this simply divide (_n-1) by a number different than 10. For example, you can divide by 100 to calculate the marginal effect of X across MV as MV increases in increments of 0.01. In other words, the analyst must choose the incremental increase in MV (or Z) for calculating the marginal effect of X. To some extent this decision will be influenced by the range of your actual modifying variable Z in the data. If Z ranges from 0 to 1, you may want to use increments of 0.01. But if Z ranges from 1 to 100 you may want to use increments of 1.

```
replace MV=. if _n >60;
```

This line of code simply says that you want to stop calculating the marginal effect of X when MV (or Z) is greater than 6. Why does the second line of code say _n >60 and not _n >6? The answer is that the MV variable is increasing in increments of 0.1. This means that when STATA's observation variable, which is denoted as _n, is at 60 then the generated modifying variable MV is at 6. How do you know when to stop calculating the marginal effect of X? This is a choice that is determined solely by the analyst. It is probably useful to calculate the marginal effect of X across the observed range of the actual modifying variable Z. You can get the observed range of Z by simply using the 'summarize' command on Z to get the min and max. In this example, the modifying variable runs from a minimum of 0 to a maximum of 6. It may be that you have a couple of outliers in Z. In this case, you may only want to show the marginal effect of X across the common values of Z. Like I said, this is a choice that the analyst must make.

```
* ************************************************************ *;
* Generate the values of W for which you want to calculate the  *;
* marginal effect (and standard errors) of X on Y.              *;
* ************************************************************ *;

scalar W0=0;

scalar W1=1;

scalar W2=2;

scalar W3=3;
```

Because you have two modifying variables you have to choose specific values for the second modifying variable at which you want to calculate the marginal effect of X. In other words, you are going to calculate the marginal effect of X on Y for all values of the Z modifying variable but only some values of the W modifying variable. What values for W do you choose? Again, this is up to the analyst. You should choose values that are substantively meaningful. You might want to choose the mean of W as well as one standard deviation above and below the mean. Particular values of W may be of interest. In this example, I have choose to calculate the marginal effect of X on Y for

all values of Z when W=0, W=1, W=2, and W=3.

```
* ************************************************************** *;
* Grab elements of the coefficient and variance-covariance matrix *;
* that are required to calculate the marginal effect and standard *;
* errors.                                                         *;
* ************************************************************** *;

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

scalar b1=b[1,1];
scalar b2=b[1,2];
scalar b3=b[1,3];
scalar b4=b[1,4];
scalar b5=b[1,5];
scalar b6=b[1,6];
scalar b7=b[1,7];

scalar varb1=V[1,1];
scalar varb2=V[2,2];
scalar varb3=V[3,3];
scalar varb4=V[4,4];
scalar varb5=V[5,5];
scalar varb6=V[6,6];
scalar varb7=V[7,7];

scalar covb1b4=V[1,4];
scalar covb1b5=V[1,5];
scalar covb1b7=V[1,7];
scalar covb4b5=V[4,5];
scalar covb4b7=V[4,7];
scalar covb5b7=V[5,7];
```

These lines simply grab the elements of the coefficient matrix and the variance-covariance matrix that are necessary to calculate the marginal effects and the standard errors. The first two lines simply create two matrices. The first matrix is called 'b' and is the coefficient matrix. This matrix is $1 \times k$ where k is the number of independent variables including the constant term. The last element of the matrix is the constant term. The first element is the coefficient on the first independent variable, the second element is the coefficient on the second independent variable etc.. The second matrix is called 'V' and is the variance-covariance matrix. The diagonal elements of this matrix are the variances and the off-diagonals are the covariances. Thus, the [1,1] element is the variance of β_1. The [1,3] element is the covariance between β_1 and β_3. The remaining lines of code simply create scalars (or numbers) that are equal to the coefficients and variances that we need to calculate the marginal effects and standard errors. The exact formulas for the marginal effects and standard errors are given on the webpage.

```
scalar list b1 b2 b3 b4 b5 b6 b7 varb1 varb2 varb3 varb4 varb5 varb6 varb7 covb1b4 covb1b5 covb1b7 covb4b5
covb4b7 covb5b7;
```

This specific line is not required for the plots. It simply lists the coefficients, variances, and covariances that you have just grabbed. I do this so that I can check that I have grabbed the correct things etc. by comparing these numbers with the STATA output from the regression.

```
* ************************************************************ *;
* We want to calculate the marginal effect of X on Y for all   *;
* values of the first modifying variable (MV or Z) and for the specific *;
* values of the second modifying variable (W) that I have chosen *;
* ************************************************************ *;

gen conb0=b1+b4*MV+b5*W0+b7*(MV*W0) if _n < 60;

gen conb1=b1+b4*MV+b5*W1+b7*(MV*W1) if _n < 60;

gen conb2=b1+b4*MV+b5*W2+b7*(MV*W2) if _n < 60;

gen conb3=b1+b4*MV+b5*W3+b7*(MV*W3) if _n < 60;
```

These lines calculate the marginal effect (or conditional beta) of X for all values of the first modifying variable MV (or Z) for each of the desired levels of the second modifying variable W so long as MV is less than 6. The precise equation for the marginal effects is given on the webpage.

```
* ************************************************************ *;
* Calculate the standard errors for the marginal effect of X on Y *;
* for all MV values of the modifying variable Z. Do this for the *;
* case when W=0, when W=1, when W=2, and when W=3.             *;
* ************************************************************ *;

gen conse0=sqrt(varb1
                + varb4*(MV^2) + varb5*(W0^2) + varb7*(MV^2)*(W0^2)
                + 2*MV*covb1b4 + 2*W0*covb1b5 + 2*MV*W0*covb1b7 + 2*MV*W0*covb4b5
                + 2*W0*(MV^2)*covb4b7 + 2*(W0^2)*MV*covb5b7 if _n < 60;

gen conse1=sqrt(varb1
                + varb4*(MV^2) + varb5*(W1^2) + varb7*(MV^2)*(W1^2)
                + 2*MV*covb1b4 + 2*W1*covb1b5 + 2*MV*W1*covb1b7 + 2*MV*W1*covb4b5
                + 2*W1*(MV^2)*covb4b7 + 2*(W1^2)*MV*covb5b7 if _n < 60;

gen conse2=sqrt(varb1
                + varb4*(MV^2) + varb5*(W2^2) + varb7*(MV^2)*(W2^2)
                + 2*MV*covb1b4 + 2*W2*covb1b5 + 2*MV*W2*covb1b7 + 2*MV*W2*covb4b5
                + 2*W2*(MV^2)*covb4b7 + 2*(W2^2)*MV*covb5b7 if _n < 60;

gen conse3=sqrt(varb1
                + varb4*(MV^2) + varb5*(W3^2) + varb7*(MV^2)*(W3^2)
                + 2*MV*covb1b4 + 2*W3*covb1b5 + 2*MV*W3*covb1b7 + 2*MV*W3*covb4b5
                + 2*W3*(MV^2)*covb4b7 + 2*(W3^2)*MV*covb5b7 if _n < 60;
```

These lines calculate the standard error (conditional standard error) for the marginal effect of X for all values of the first modifying variable MV (or Z) for each of the desired levels of the second modifying variable W, so long as MV is less than 6. The precise equation for the standard errors is given on the webpage.

```
* ************************************************************ *;
* Create t statistics                                          *;
* ************************************************************ *;

gen t0=conb0/conse0;

gen t1=conb1/conse1;

gen t2=conb2/conse2;
```

```
gen t3=conb3/conse3;
```

These lines calculate t-statistics for the marginal effects we have calculated. We will use these to determine when the marginal effect is statistically significant at the desired level of significance.

```
* ************************************************************** *;
* Generate a 'shadow' variable that is equal to the marginal     *;
* marginal effects that have already been calculated. These will *;
* be helpful in placing the stars in the figure as you will see. *;
* ************************************************************** *;

gen consb0=conb0;

gen consb1=conb1;

gen consb2=conb2;

gen consb3=conb3;
```

What do these lines do? Well, we are going to place stars when the marginal effect is statistically significant. These lines just create what I call 'shadow' marginal effects. Their use will become obvious in a moment.

```
* ************************************************************** *;
* Make the 'shadow' variable be missing if the t score is not    *;
* larger than the critical level of significance that you want.  *;
* ************************************************************** *;

replace consb0 = . if abs(t0)< 1.96;

replace consb1 = . if abs(t1)< 1.96;

replace consb2 = . if abs(t2)< 1.96;

replace consb3 = . if abs(t3)< 1.96;
```

Now I am replacing the 'shadow' marginal effects as missing if they are not statistically significant at the 95% level. In other words, make them missing if the t-statistic is not larger than the critical value for statistical significance. In this example, I am assuming the critical value is 1.96 for 95% statistical significance. You will have to change this critical value if you have fewer degrees of freedom than 120 or if you want a different level of statistical significance. So, I am left with only statistically significant 'shadow' marginal effects.

```
* ************************************************************** *;
* Generate a string variable called str1 that is designated with a star.  *;
* ************************************************************** *;

generate str1 txt"*";
```

Since I want to use a star to indicate statistical significance (to mirror how we represent statistical significance in traditional tables of results) we need to create a string variable that is a star. This is what this line does.

```
* ************************************************************ *;
* Graph the marginal effect of X on Y across the whole range of the first   *;
* modifying variable Z and for the desired values of the second modifying   *;
* variable W.                                                               *;
* ************************************************************ *;

graph twoway line conb0 MV, clpattern(solid) clwidth(thin)
             || scatter consb0 MV, mlabel(txt) msymbol(i) mlabsize(vsmall) mlabgap(-1.0) mlabposition(11)
             || line conb1 MV, clpattern(solid) clwidth(thin)
             || scatter consb1 MV, mlabel(txt) msymbol(i) mlabsize(vsmall) mlabgap(-1.0) mlabposition(11)
             || line conb2 MV, clpattern(solid) clwidth(thin)
             || scatter consb2 MV, mlabel(txt) msymbol(i) mlabsize(vsmall) mlabgap(-1.0) mlabposition(11)
             || line conb3 MV, clpattern(solid) clwidth(thin)
             || scatter consb3 MV, mlabel(txt) msymbol(i) mlabsize(vsmall) mlabgap(-1.0) mlabposition(11)
             || ,
               xlabel(0 1 2 3, labsize(2.5))
               ylabel(-0.5 0 0.5 1 1.5, labsize(2.5))
               yscale(noline)
               xscale(noline)
               legend(off)
               yline(0, lcolor(black))
               title(Marginal Effect of X on Y As Z and W Change, size(4))
               subtitle(" " "Dependent Variable: Y" " " " ", size(3))
               xtitle(Z, size(3))
               ytitle(Marginal Effect of X, size(3))
               xsca(titlegap(2))
               ysca(titlegap(4))
               text(0.5 3.3 "W=3" justification(left) size(2.5))
               text(0.5 3.3 "W=2", justification(left) size(2.5))
               text(0.5 3.3 "W=1", justification(left) size(2.5))
               text(0.5 3.3 "W=0", justification(left) size(2.5))
               text(1.5 1 "* indicates significance at the 95% level", justification(left) size(2.5))
               scheme(s2mono) graphregion(fcolor(white))
               graphregion(margin(r=28));
```

These lines of code plot the marginal effect of X on Y when W=0, when W=1, when W=2, and when W=3. They also indicate statistical significance at the 95% level by the use of stars. A basic plot can be generated using fewer lines of code, but this is what the code is doing: First, we plot the marginal effect of X on Y when W=0 (conb0) against MV (or Z). This is a line. The next line then plots the 'shadow' marginal effects of X on Y when W=0 that are significant against MV (or Z) - But instead of plotting the marginal effect, we use a star (this is the mlabel(txt) command). STATA would place the star directly on top of the marginal effect line unless we told it to place the star just above the line - we do this with the 'mlabgap' and 'mlabposition' commands. Thus, the first two lines of code plot the marginal effect of X on Y when W=0 and places stars just above this line when the marginal effect is significant. The next six lines of code repeat this for when W=1, W=2, and W=3. This is the basic plot. The 'xlabel' and 'ylabel' commands basically format the axes. I turn the legend off because it is better to provide legends yourself. The 'title' commands are self-explanatory. Thus far, we have a plot for four lines for the various marginal effects (and stars above them when relevant). The lines of code beginning with 'text' basically allow you to place legends next to each marginal effect line so that the reader can see the scenario that is being captured. The analyst will have to fiddle with this code to know exactly where to place these legends. Finally, the last couple of lines of code do some more formatting for the whole plot.

```
* ************************************************************ *;
* Figure can be saved in a variety of formats.                 *;
* ************************************************************ *;

graph export h: figure1.eps, replace;

translate @Graph h: figure1.wmf;
```

There are several ways of exporting the plot so that you can work on it or import it into your document. I tend to use one of these commands. The first command exports the plot as an encapsulated postscript (eps) file. These can be imported directly into any LaTeX document. The second command exports the plot in wmf format. You can insert the plot into a word document simply by inserting it is a picture file.

```
* ************************************************************ *;
* THE END                                                      *;
* ************************************************************ *;

exit;
```

Finally, you will note that the plot indicates statistical significance through the use of stars. I do this because I find that plots involving two modifying variables can become cluttered if you use confidence intervals. However, you can use the code from the example with a single modifying variable to see how to create confidence intervals. It is certainly possible to show confidence intervals with multiple modifying variables.