Understanding The STATA Code for Plots With A Single Modifying Variable

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 is a single modifying variable. 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 a single modifying variable where the dependent     *;
*          variable is continuous. The estimated model is:            *;
*          Y = b0 + b1X + b2Z + b3XZ + 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 + b3XZ + b4Controls + epsilon *;
* ****************************************************************** *;

regress  Y  X  Z  XZ  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 modifying variable. The third independent variable should be the interaction term.

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

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

This line of code is important because it indicates the values of the modifying variable you are going to calculate the marginal effect at. It generates a variable called MV that is going to act as your 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.

```
* ************************************************************ *;
* 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 varb1=V[1,1];
scalar varb2=V[2,2];
scalar varb3=V[3,3];

scalar covb1b3=V[1,3];
scalar covb2b3=V[2,3];
```

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 varb1 varb2 varb3 covb1b3 covb2b3;
```

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.

```
* ************************************************************** *;
* Calculate the marginal effect of X on Y for all MV values of   *;
* the modifying variable Z.                                      *;
* ************************************************************** *;

gen conb=b1+b3*MV if _n < 60;
```

This line calculates the marginal effect (or conditional beta) of X for all values of the modifying variable MV so long as MV is less than 6. So, now we have the marginal effect of X for when MV=0, when MV=0.1, when MV=0.2 ..., when MV=5.9. The precise equation for the marginal effect 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.                 *;
* ************************************************************** *;

gen conse=sqrt(varb1+varb3*(MV^2)+2*covb1b3*MV) if _n < 60;
```

This line calculates the standard error (conditional standard error) for the marginal effect of X for all values of the modifying variable MV so long as MV is less than 6. So, now we have the conditional standard errors for the marginal effect of X for when MV=0, when MV=0.1, when MV=0.2 ..., when MV=5.9. The precise equation for the standard error is given on the webpage.

```
* ************************************************************** *;
* Generate upper and lower bounds of the confidence interval.    *;
* Specify the significance of the confidence interval.           *;
* ************************************************************** *;

gen a=1.96*conse;

gen upper=conb+a;

gen lower=conb-a;
```

These lines of code basically create the confidence intervals. You have to decide what the critical value of significance is for your dataset. For example, if you want 95% confidence intervals and you have more than 120 degrees of freedom you can use the critical value of 1.96 as I do here. If you want confidence intervals for a different level of significance or you have fewer degrees of freedom you will have to replace the 1.96 in the first line of the code here with the relevant critical value. Critical values are typically shown in tables at the back of most econometric books. The second and third lines basically create the upper and lower bounds of the confidence interval.

```
* ************************************************************ *;
* Graph the marginal effect of X on Y across the desired range of *;
* the modifying variable Z. Show the confidence interval.      *;
* ************************************************************ *;

graph twoway line conb MV, clwidth(medium) clcolor(blue) clcolor(black)
            || line upper MV, clpattern(dash) clwidth(thin) clcolor(black)
            || line lower MV, clpattern(dash) clwidth(thin) clcolor(black)
            || ,
              xlabel(0 1 2 3 4 5 6, labsize(2.5))
              ylabel(-4 -2 0 2 4, labsize(2.5))
              yscale(noline)
              xscale(noline)
              legend(col(1) order(1 2) label(1 "Marginal Effect of X")
                                       label(2 "95% Confidence Interval")
                                       label(3 " "))
              yline(0, lcolor(black))
              title("Marginal Effect of X on Y As Z Changes", size(4))
              subtitle(" " "Dependent Variable: Y" " ", size(3))
              xtitle( Z, size(3) )
              xsca(titlegap(2))
              ysca(titlegap(2))
              ytitle("Marginal Effect of X", size(3))
              scheme(s2mono) graphregion(fcolor(white));
```

These lines create the plot showing the marginal effect of X on Y across the observed range of the modifying variable Z with 95% confidence intervals. You can create the plot using fewer lines than this if you want. But this is what the code is doing: The first line plots the marginal effect (conb) against the modifying variable (MV). The rest of the line is simply formatting for what the plotted line should look like. The second line of code says plot the upper bound of the confidence interval (upper) against the modifying variable (MV) and make this line a dashed line. The third line is the same except that it says plot the lower bound of the confidence interval. The 'xlabel' and 'ylabel' commands are simply formatting commands for the x- and y-axis. My preference is not to have a line for the x- and y-axis and so I get rid of these with the 'yscale(noline)' and 'xscale(noline)' commands. The 'legend' command simply tells STATA what each of the lines that you plotted should be called. The 'title' commands are self-explanatory. The other lines are just more formatting commands and can be ignored. This graphing code works for STATA 8. You will have to change the code if you are using STATA 7.

```
* ************************************************************ *;
* 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 can determine the precise value of the modifying variable that the marginal effect of X becomes significant or positive (or negative) etc. by going into the STATA data editor, scroll to the right, and look at the variables that you have created. For example, 'conb' will show precisely when the marginal effect changes signs. You can look for the values of MV at which 'upper' and 'lower' have the same sign to determine when the marginal effect is significant. You can sort the data by the modifying variable to determine what percentage of the sample falls within any range of significance etc.