

A Quick and Dirty Guide to Econometric Software

Myles J. Callan

February 9, 2006

Abstract

This aim of this guide is to get the reader to use the software as quickly as possible—the logic being that there is no substitute for practice in the world of econometric software. However, practice only leads to expertise if the tools are being used correctly to begin with (the paradox of trying to learn anything¹), so that the focus is on providing a clear intuition as to how to use the software (which will not be altogether technically exact or comprehensive). The programs that we will begin with are: Matlab and Eviews.

1 The Big Picture

Matlab and Eviews are significantly different applications to provide an interesting contrast (that, I think, makes learning them easier when done together—you can tell me what you think when you have learned how to use them). Eviews was developed as a **Menu** driven application (like Word or Excel), to which **scripting (or programming)** capabilities were added later. Matlab’s development was the direct opposite: It was designed as a scripting application, and menu driven (mini-)applications were created later by users.

You are going to begin with the scripting features of the software.² This is a technically more challenging skill to learn, but it also more rewarding in terms of understanding the applications.

¹If you put the ladder against the wrong wall, it doesn’t matter how quickly you climb

²Using applications that allow scripting/programming is usually the most efficient way to perform specialized tasks such as data analysis. This is because menu driving applications require that for each action you must move the mouse to the required place, selecting the required option, and so on: The more actions required, the more clicks required. You will see that a script/program is a labor saving device. Also, when tasks get more specialized the actions required are more convoluted. It tends to be easier to write the commands out rather than selecting them from menus (that is, it is easier to remember the command names than to remember the location of the commands on a menu). For example, this review is written in \LaTeX . This is a simple scripting language that is a substitute for using a word processor. If you imagine trying to write :

$$A^{tr}(x) = (8\pi)^{\frac{1}{2}} \sum_K \left[e_1(q_K^{(1)} \cos(K \cdot x) + q_K^{(3)} \sin(K \cdot x)) + e_2(q_K^{(2)} \cos(K \cdot x) + q_K^{(4)} \sin(K \cdot x)) \right] \quad (1)$$

in Word, you first of all would need to know whether it is possible (it is, but it is not as nicely “typeset”/formatted), then you would have to learn where all of the formatting options are. Whereas, if you write these types of formulas constantly, learning the \LaTeX code would involve memorizing less (as the “commands” are usually named intuitively, unlike the position of some of the options on menus). Although initially a \LaTeX file may look a little messy, it has relatively straightforward structure.

There are 5 important features that you need to be able to understand to make the application do what you want it to do. These are:

1. Comments — allow you to add sufficient comments (text that is ignored by the application) to your programs to enable you to understand what they do in 2 or 3 years time.
2. “Internal Rule Book”—understanding that there is a “internal rule book” which is written using the logic of the software developer ³
3. Functions vs. Scripts — very simply: functions add to the internal rule book of the application, whereas scripts simply tell the application what to do.
4. Data in — how do I input the data that I want the application to analyse.
5. Results out — how do I save the results of my analysis, and, equally importantly, how do I get the results in the format that will be useful for me?

In each case, learning by doing is the quickest way to learn. To ensure that you have the ladder again the right wall, the following provides a few notes on each feature. These notes in combination with the tutorial and the programs supplied should allow you to quickly get started using Eviews and Matlab.

2 Comments

Most people writing a guide to programming languages would include a note on “Comments” in a footnote. I, however, view comments, in the long term development of learning how to program, as fundamental. Organization is the key to learning how to program. In the beginning, the last program that you wrote will form the basis for the next program that you write. As you learn how to use the application to analyse data in different ways, the programs that you write will become more and more specialized (and longer), so that you will quickly have a library of programs. If you have not commented them, or archived them in a location that makes them easy to find, then the next time that you are analysing a dataset in a way that you last did 3 years ago you may have to start your program from scratch. This is a very inefficient way of proceeding, especially when it takes only a few minutes to add comments that will describe the program.

3 “Internal Rule Book”

The main comment that I can make here is: Learn by doing. Eviews and Matlab are two applications that allow you to, amongst other things, analyze data.⁴ For **any** type of statistical analysis that

³This is **the** most important feature to understand when learning how to write scripts/programs. There are no intellectual brownie-points for knowing that the developer of Eviews decided that “” denotes a comment, whereas the developer of Matlab use the “%”. There are intellectual brownie-points for understanding that comments are a vital part of writing a script, so that there must be **some** method of signifying them. For example, in **L^AT_EX** the symbol is also the “%”, whereas in **HTML** comments are denoted, curiously, by “<!-” — the latter being the perfect of example of “how was I supposed to know that?”

⁴To be precise, Eviews, short for Econometric Views, is primarily for data analysis, whereas Matlab, short for matrix laboratory, is useful for any form of analysis that uses matrices

you can think of, these applications will have some command in the internal rule book that allows you to perform that analysis. The more that you use the program, the more of the commands that you remember (unless you have a photographic memory and enjoy reading manuals).

4 Functions vs. Scripts

This is where the two programs differ most significantly: Matlab is designed to allow you to write functions (i.e. add to the internal rule book), whereas Eviews does not have that capability.

Both, obviously, have powerful scripting languages.

Another very important difference is that you can perform all your statistical analysis in Eviews without ever having to write a script. This is because it was initially written as a menu driven program (the rationale for writing scripts is efficiency, although the gain in efficiency when doing projects in an introductory course in econometrics is negligible—in fact, it is probably quicker to use the menus). With Matlab, you will not get anywhere without using the programming commands, although you can use them in the command window (and not have to write a script, but you will have to type the same commands in a script or on the window). So, writing a script is more efficient because the script can be used again.

5 Data in

Unless you are simulating data (your first assignment), you will have to input the data that you are analysing. In Eviews, it is easier to do this using the GUI (i.e. menus). In Matlab it is easier to do this using the scripting language. If you use the scripting language in either program, you must tell the application certain key pieces of information about your data: the format it is in (i.e. is it a text file, is it an excel file), the size of the dataset (number of observations and number of variables), ... are the most common facts that the applications need to know to accurately load the data.

6 Results out

The reason for using these applications is to get information about the data that you input. You will want to save this information in some way: either saving it to your disk, or printing a hardcopy. The sample programs teach you how to do this.

7 Conclusion

I hope that it will be very quickly obvious to you that you cannot, and will not be expected to, learn these programs overnight. In fact to successfully and expertly use them usually takes years. Take your time and enjoy learning how to use them, by the time you are writing your dissertation they will be child's play to you!