The Stata Journal (2016)
16, Number 2, pp. 511–516

Review of Christopher F. Baum's An Introduction to Stata Programming, Second Edition

Clyde Schechter
Albert Einstein College of Medicine
Bronx, NY

Abstract. In this article, I review *An Introduction to Stata Programming, Second Edition*, by Christopher F. Baum (2016 [Stata Press]).

Keywords: gn0070, book review, introduction, Stata, programming, isp2

1 Introduction

StataCorp has made initial adoption of its software easy through the graphical user interface (GUI), but that will only take one so far. The GUI provides limited capacity for automating analyses: there are occasional commands that the GUI cannot access. It also provides limited ability for users to customize their output. Anyone who uses Stata often will eventually encounter such limitations. After using the GUI, a Stata user may begin using the command interface; from there, it is a short step to saving and executing analyses in do-files, at which point the user is programming. For many Stata users, Stata is their first programming language. Even for those who might be familiar with other statistical packages, or who have some experience with a general-purpose computer language such as C, Python, or Java, Stata can seem difficult. At first glance, an expression like `‘"‘:word ‘=‘‘j’’+1’’ of ‘vlist’"’` can be intimidating. Stata's comprehensive manuals are intended as references and, except for the brief [GS] section, are not designed to teach Stata programming. But there are resources for learning Stata programming, in the form of both courses and books. In this article, I review a recent update of one such self-study book.

In the preface to *An Introduction to Stata Programming, Second Edition*, Baum proposes an ambitious agenda. Starting almost from square one, his book leads the reader through the basics of Stata, do-file programming, ado-file programming, and Mata programming in 395 pages. At first, I was skeptical of the possibility. However, after working with the book for several months, I have become a believer.

The book's structure is innovative. Although the first two chapters are a didactic presentation of Stata fundamentals, thereafter the chapters alternate between a didactic presentation of a topic and a "cookbook" chapter that presents a realistic (and occasionally real) problem and develops its solution. The solutions often begin with a simplified, basic approach, followed by enhancements that improve the code or add capabilities. As such, the flow of the "cookbook" chapters genuinely reflects the process

that an experienced Stata programmer would follow when solving real problems. While it is quite typical of programming textbooks to mix didactics with worked problems, examples that are this realistic and extensive, built up incrementally, are exceptional. The "cookbook" metaphor, though, understates the value of the presentation. None of my kitchen's tomes explain the rationale behind the recipes, nor do they review a recipe afterward and suggest ways to improve it!

The book's order of presentation, in addition to being logical, enables a reader to proceed through each chapter topic while starting and stopping at sections where his or her own goals are addressed. For example, while all regular Stata users must be able to write do-files, the ability to write ado-files is needed only by some. The first 10 chapters suffice for the former, and those readers can stop there. Similarly, an experienced do-file writer interested in developing new estimators for Stata could begin with chapter 11 and continue through the end of the book without missing anything.

2 Content

An Introduction to Stata Programming, Second Edition, is not a comprehensive reference for Stata or Mata commands. The book thoroughly covers the basic commands that regular Stata programmers need to use repeatedly. It discusses graphics but only lightly—there are other lengthy books devoted to Stata's pluripotential `graph` command. The book does not discuss Bayesian analysis and item response theory commands, introduced in Stata 14, nor does it discuss survival analysis or multilevel modeling. These omissions are noted not as criticisms but to emphasize that this book focuses on teaching Stata programming and is not a comprehensive resource for Stata users.

What, then, is in the book? Chapter 1 begins by explaining the difference between using Stata casually and programming in Stata. This chapter explains the reasons why a user might want to become a programmer. Chapter 2 covers the fundamentals: creating and organizing do-files, program files, and datasets and importing data from other formats. One of the most important parts of this book is the brief but content-rich section on programming style near the end of chapter 2. In teaching this, the book refrains from adopting specific variants of coding style that programmers often argue about. However, it still emphasizes that every programmer needs to adopt a set of rules and conventions to follow consistently.

Chapter 3 discusses the creation and transformation of variables, and it introduces by-groups and Stata's macros (which are string constants, not to be confused with parameterized blocks of code sometimes referred to as macros in other software). The centrality of these concepts in Stata programming cannot be overstated. The presentation in this chapter is complete but terse. The topics discussed truly come to life in chapter 4, where we see them used to solve real data management problems.

Chapters 5 and 6 provide the remaining essentials of data management and results presentation. When I teach data analysis, my first law is "Never trust anybody else's data," and my second is "Never trust your own data." These chapters introduce and develop the all-important `assert` command and its application to real-life data cleaning. They also include an exposition of the `reshape`, `append`, and `merge` commands and details on accessing the stored results of Stata's commands.

If statistical programming required a license, the core of the examination would be based on the first six chapters of this book. Anyone who masters these six chapters is well positioned to effectively and efficiently create properly organized and cleaned analytic datasets and develop coherent, tailored displays of analytic results.

Chapters 7 and 8 discuss the automation of repetitive calculations. The versatile `foreach` and `forvalues` commands receive an extensive treatment in these chapters—along with `by:`, these commands are the fundamental constructs for repetitive calculations in Stata. Their importance is reflected in the generous amount of space allocated to them. These chapters also cover specialized repetition commands, such as the `statsby:` and `rolling:` prefixes, as well as the still more specialized commands `permute`, `bootstrap`, `jackknife`, and `simulate`.

Chapters 9 and 10 introduce some less-used techniques, including the `postfile` suite, the `file` suite, `sersets`, and data characteristics. While these all belong in the toolkit of the advanced programmer, they are primarily useful for creating highly customized outputs, and most Stata programmers could function for long intervals without recourse to these.

The reader who learns the content of these first 10 chapters will be prepared to effectively analyze real datasets in Stata and present well-organized, concise output. The book would be well worth the cost and effort even if one stopped here.

Chapters 11 and 12 introduce the fundamentals of ado-file programming, including the `program`, `syntax`, `return`, and `marksample` commands. Ado-files contain Stata programs, which are blocks of code assigned a name by which they can be invoked and which usually accomplish some fairly general-purpose task. The programs written in ado-files, unlike the code written in do-files, are intended to be used with a variety of datasets. Thus, for example, ado-file programs generally do not reference specific variable names. They are written to work with "generic" data. These chapters also teach how to write programs to implement functions for commands such as `egen`, `nl`, and `ml`. The book, starting from chapter 11, becomes more difficult because of the inherently greater complexity of the content. Chapter 11 is also longer than all previous chapters, roughly twice the length of any other. Even moderately experienced do-file programmers should lightly review chapters 1–10 before starting here, because any gaps in earlier knowledge may cause problems. The worked examples in chapter 12 reflect the increased difficulty of the material. Although the first few examples are simple and straightforward, the difficulty soon increases, and, as befits the kinds of things that ordinarily warrant writing an ado-file program, the underlying statistics become more detailed. While there is nothing in the chapter that a well-trained statistician will find

intimidating, users with only an introductory-level statistics course behind them may find some of the content challenging.

The first 12 chapters expose the reader to the most important techniques used in Stata programming. One might consider printing out a "diploma" after mastering these chapters! Of course, to really consolidate this learning, one needs practice and experience.

Chapters 13 and 14 focus on StataCorp's other programming language, Mata. I put off learning Mata for several years, and I can tell you that I have yet to discover anything that can be done in Mata that cannot be somehow accomplished in Stata. But I have also seen on Statalist how Mata can sometimes be used to reduce a long Stata program to just a few lines of code. Mata programs can also be compiled and will execute faster than the corresponding Stata code. For professional statisticians who want to develop and implement their own models and estimators, Mata provides the advanced matrix-calculation and manipulation capabilities needed. Chapter 13 begins, appropriately, by discussing the interface between Stata and Mata and the use of Mata both in programs and interactively from the command line. The chapter then presents basic matrix operations followed by more advanced functions. It then explains how to write a Mata program and how to create libraries of Mata programs.

I found these two final chapters to be the "proof of the pudding". Although I have been writing Stata programs for over two decades and was already familiar with nearly everything in the first 12 chapters, I had never learned Mata. So I took the plunge with this book. I found the book's presentation of Mata to be clear and logical. After completing these final chapters, I felt capable of using Mata with reasonable facility, although I still need more practice before I can consider myself to have truly mastered it. But this book provided me with the necessary groundwork to move forward, and I now understand things that I could never quite grasp from using only the Mata user manual.

3 Strengths and limitations

A difficulty faced by anyone teaching Stata programming is how to address a professionally diverse audience. Baum is an economist, and most of the examples in this book use economic datasets and apply tools and techniques frequently used in economics and finance. While there are some examples based in other fields, they are the exception. Some substantive scientific background is provided with these examples, but readers who are completely unfamiliar with these disciplines may need to turn to other sources (a dictionary, search engine, or online reference materials) to clarify some vocabulary.

Let me emphasize that this book does not purport to teach statistics, and it will not be helpful if that is your need. Furthermore, it is not organized as a reference for experienced Stata programmers. Although the text is well indexed and it is easy to locate information on certain commands, it can be difficult to find details on specific programming techniques without already knowing where to look. Moreover, there are

large swaths of Stata commands never mentioned in the book. In short, this book will not supplant Stata's online help files and user manuals.

The material in chapter 11 is more difficult than that in the earlier chapters. If the reader has mastered all chapters before 11, then he or she should be able to proceed without trouble. But for a reader with only minimal experience writing do-files and wanting only to quickly learn how to write a short special-purpose program, starting with this chapter may prove too difficult.

The layout and size of the book are convenient. The font is easy on the eyes, and the book fits easily into a backpack, briefcase, or medium-sized handbag. Users will find the typographic conventions in the book to be quite familiar from the Stata user manuals.

Baum is also the curator of Boston College's Statistical Software Components Stata code repository. That position enables him to introduce the reader to many user-written tools that make performing important tasks easier and quicker. I learned several new commands this way while reading the book for this review. The book also discusses the pros and cons of using prepackaged solutions that achieve generality by including many user-specified options as well as the pros and cons of writing your own narrowly tailored code from the basics. Any commands used in the book that are not part of official Stata are available from the Statistical Software Components archive. The datasets used in the book and the example do- and ado-files are all available on the Stata Press website.

One aspect of this book that I particularly like is its emphasis on programming style. The bare-minimum requirement of any program is that it must produce correct results. Beyond that, programs often require periodic extension or revision, so it is important that other programmers, and even the original programmer, can easily return to a program and understand how it works. Commenting, code formatting, and following certain conventions greatly enhance the human readability of code and make it easier to maintain. This is a lesson that many programmers (in any language) learn the hard way, often after having a bad experience using a program they wrote several months earlier and having little recall of how it works. The importance of programming style is often overlooked in introductory programming courses, but programming style is not just for advanced programmers. Ideally, good coding style should be practiced starting with the very first line of code that one ever writes. It is gratifying that this point is made early in this book, specifically near the end of chapter 2 for do-files and again in chapter 11 for ado-files, and that the code examples all adhere to the best practices.

I was also pleased by the book's focus on data validation. Data analysis sometimes results in unpleasant surprises, such as unexpected error messages from `xtset` about duplicate observations or regression results that cannot possibly be right and are due to incorrectly entered data (such as data claiming a mother is 15 years younger than her daughter or that an adult weighs 5 kg). Worse still are the results that are not obviously wrong and that you learn about much later when others complain that your results cannot be reproduced. The generous use of `assert`ions in do-files that create analytic datasets, taught in chapter 5, is the best way to prevent these errors.

I would have liked to see more emphasis on program correctness in both the didactics and examples of chapters 11 and 12. Although the book discusses and illustrates validation scripts, little is said about designing programs to catch user errors. A program should not only produce correct answers when given valid inputs but also identify when its inputs are not valid and fail gracefully with an informative error message. While the `syntax` command does a thorough validation of variable lists and option types, it is still the programmer's responsibility to range check the values of variables and arguments and to verify any assumptions needed for the algorithm to work correctly with the data. This type of checking for program correctness deserves more emphasis and illustration.

The book does not include any unworked exercises for the reader to attempt on his or her own. A learner whose work or school environment provides problems to work will not be disadvantaged by this. However, for a learner not in such an environment who wants to expand his or her skills, this will be a limitation. Instructors considering using this book as a text to accompany a course on Stata will need a supplementary workbook with problems.

4 Conclusion

An Introduction to Stata Programming, Second Edition, is a well-written and superbly organized book. It is suitable for a Stata user at any level who wants to learn Stata programming or improve already acquired programming skills. The reader can advance from neophyte to expert in Stata (and Mata) programming in useful but manageable increments by studying this book one or two chapters at a time. The extensively developed examples of realistic programming problems enliven the didactic presentation. The methods taught in this book can be applied to cutting-edge statistical work, but the book's presentation and most of the examples assume readers have only the statistical knowledge commonly possessed by graduate students and early-career data analysts. The book will be especially appealing to learners who have a background in finance or economics.

5 Reference

Baum, C. F. 2016. *An Introduction to Stata Programming*. 2nd ed. College Station, TX: Stata Press.

About the author

Clyde Schechter is a professor of family and social medicine at the Albert Einstein College of Medicine, where he works as an epidemiologist and focuses on clinical and health services research projects involving many specialties and disciplines. He has used Stata since version 4, is an active participant in the Statalist forum, and is an occasional contributor to the *Stata Journal*.