1 Introduction

1.1 Is this book for me?

This book is for you if you have tried to learn Mata by reading the *Mata Reference Manual* and failed. You are not alone. Though the manual describes the parts of Mata, it never gets around to telling you what Mata is, what is special about Mata, what you might do with Mata, or even how Mata's parts fit together. This book does that.

This is an applied book. It will teach you the modern way to write programs, which is to say, it will teach you about structures, classes, and pointers. And the book will show you some programming techniques that may be new to you. In short, in this book, we are going to use Mata to write programs that are good enough that StataCorp could distribute them.

This book is for "serious programmers and those who want to be". Fifteen years ago, the subtitle would have referenced professional rather than serious programmers, and yet I would have written the same book. These days, the distinction is evaporating. I meet researchers who do not program for a living but are most certainly serious. And I meet the other kind, too.

A serious programmer is someone who has a serious interest in sharpening their programming skills and broadening their knowledge of programming tools. There is an easy test to determine whether you are serious. If I tell you that I know of a new technique for programming interrelated equations and your response is "Tell me about it," then you are serious.

Being serious is a matter of attitude, not current skill level or knowledge.

Still, I made assumptions in writing this book. I assumed that you have some experience with at least one programming language, be it Stata's ado, Python, Java, C++, Fortran, or any other language you care to mention. I also assumed that you already know that programs contain conditional statements and loops. If you need a first introduction to

programming, you could look at the introductory section of the Mata manual or at the Mata chapters in Baum's friendly text *An Introduction to Stata Programming* (2016).

The examples in this book are statistical and mathematical. Formulas are provided, but the formulas are of secondary importance. They just provide the examples of something for us to program.

In this book, I will show you a language aimed at programming statistical and data management applications that has all the usual features and some unique ones, too. And I will show you programming techniques that might be new to you.

As I said, being serious is a matter of attitude. New techniques and languages are continually being developed, and you need to learn them, just as I still learn them. I have been programming for 45 years as a professional. I have a lot of experience and knowledge, but I have not stopped learning new techniques. I may be a professional programmer, but more importantly, I am a serious one.

1.2 What is Mata?

Many Stata users would describe Mata as a matrix language. StataCorp itself markets Mata that way. Mata would be more accurately described, however, as an across-platform portable-code compiled programming language that happens to have matrix capabilities. Just as important as its matrix capabilities are Mata's structures, classes, and pointers.

We at StataCorp designed and wrote Mata to be the development language that we would use. Nowadays, we write most new features of Stata in Mata. Before Mata existed, we used C. Compared with C, Mata code is easier to write, less error prone, easier to debug, and easier to maintain.

It is important that Mata is compiled. Being compiled means that programs run fast. Stata's other programming language, ado, is interpreted. Interpreted languages are slow in comparison with compiled languages. Mata code runs 10–40 times faster than ado.

Mata looks a lot like C and C++. In *The C Programming Language*, Kernighan and Ritchie (1978) introduced what has become perhaps the most famous first program:

```
main()
{
        printf("hello, world\n") ;
}
```