Book Reviews

MARK LAMOURINE AND RIK FARROW

Python for Data Analysis

Wes McKinney
O'Reilly Media, 2013, 447 pages
ISBN 978-1-449-31979-3
Reviewed by Mark Lamourine

Python has become a popular programming language for numeric processing. This is largely due to the creation of NumPy, a binary module that provides the performance of a compiled program with the ease of use and fast development cycle of an interpreted language.

NumPy was just the start. A range of modules have been added on top to provide even more specialized functions or different techniques for accessing and manipulating numerical data. Wes McKinney is actively developing an extension called "pandas" (the name is not capitalized). According to the introduction on the pandas section of pydata.org (http://pandas.pydata.org), "pandas is a Python package providing fast, flexible, and expressive data structures designed to make working with 'relational' or 'labeled' data both easy and intuitive." That's important because NumPy by itself wasn't designed to work with complex data structures composed of mixed data in the columns.

In *Python for Data Analysis*, McKinney offers a primer for data analysis with Python, using iPython for interaction, and NumPy and pandas for numerical manipulation and data access.

McKinney opens with a whirlwind tour of simple examples to showcase the capabilities of Python, iPython, NumPy and pandas, along with matplotlib and SciPy. He then returns to the beginning to add depth to each topic. His treatment of iPython for interactive development is one of the best I've seen. I learned several things I hadn't known before. The chapter on NumPy basics is similarly informative.

When he gets to using pandas I must say I began to lose the thread. The examples are clear and effective in showing the creation of and operations on the data structures, but I don't have the background in data analysis to follow why they are important. I can only assume that someone with more experience would know when and why to use them. Operations for loading and storing data are clear to me, but data transformations, aggregation, and group operations are outside my scope. The section on generating graphics with matplotlib was also in my range, but when McKinney uses it to display financial and time-series data, I appreciate the images but not necessarily their significance.

In the final section, McKinney returns to advanced features of NumPy, including some internals and optimizations.

Python for Data Analysis is not going to be a good place for someone new to numerical analysis. For readers already familiar with the concepts but who need to learn to do their work using Python, it's going to be a great resource.

An Introduction to Programming in Go

Caleb Doxsey
CreateSpace Independent Publishing Platform, 2012, 161 pages
ISBN 978-1478355823
Reviewed by Mark Lamourine

Google (in the persons of Robert Griesemer, Rob Pike, Ken Thompson) began development of Go in 2007, announced it publicly in 2009, and released version 1.0 in 2012. Go is distributed under the BSD license. At about the same time, according to the copyright notice, Caleb Doxsey was releasing *An Introduction to Programming in Go* on the Web (www.golang-book.com/books/intro), in PDF, and in hard copy. The copyright page notes that portions are used under the Creative Commons 3.0 Attribution license, originally published by Google, but I can't find any indications of which parts.

In this slim volume Doxsey provides a brief look into all of the major features of the Go language and of the development environment: installation, compilation, syntax and language features, a model for testing, software packages, and a survey of the core standard libraries. Each chapter closes with a set of exercises that should be easy for an experienced developer, and an appropriate challenge for a classroom student.

The hard copy clearly shows its heritage as a Web document. The typography and layout almost feel like a large-text book, perhaps typeset originally with LaTeX. The book was published using the CreateSpace Independent Publishing Platform, a service offered by Amazon. I've reviewed other books created using alternative publishers, and I'm encouraged that I can buy books like this that might never have made it to print under a traditional imprint. Amazon might not be to some people's liking but it is one choice for an unsigned author.

The independent publishing origin doesn't detract at all from the book's utility as a primer for new Go programmers. This is yet another case where I appreciate a paper book even when the complete text is available online. The Web version shows a copyright date of 2015, so it's likely that Doxsey is continuing to make updates, though a quick scan didn't show anything obvious. Go recently released version 1.5 and that includes several backward-compatible syntax changes that could make the examples simpler or clearer.

An Introduction to Programming in Go is by no means a complete, in-depth tour of the Go language, but as a first look it is a good choice.

The Go Programming Language

Alan Donovan and Brian Kernighan
Addison-Wesley Professional Computing Series, 2015, 400 pages
ISBN 978-0134190440
Reviewed by Rik Farrow

I still have Kernighan and Ritchie's *The C Programming Language* sitting on the bookshelf within reach of my desk. What made that book so useful was its clarity, brevity, and a wealth of examples that were actually relevant to a lot of the tasks one might have to handle in C.

Alan Donovan and Brian Kernighan have written an even better book. Like the original "white book," they include relevant examples all through the book's 400 pages. What's much better is that we now have the Internet, and all of the examples can be built using `go build`, or downloaded for editing, and that's very useful when it comes time to try out the exercises.

I don't think anyone can learn a programming language without actually working with it, and the path Donovan and Kernighan provide is a reasonable one. The first chapter, "Tutorial," covers examples for things you can learn in the next seven chapters. That means you can't possibly understand all that's covered in Chapter one, but you get an overview of what will be described eventually in greater depth.

And there is a lot of depth. I had no idea how different Go was, although I had some inklings. One of the things I liked about the book is that the authors casually repeat key points—for example, that variables, functions, and types that you want to export between packages must begin with a capital letter. This, and other subtleties, can easily trip up someone new to a programming language, and I found the reminders helpful.

One-fifth of the book is devoted to just two topics: interfaces and goroutines. Interfaces are contracts, promises in the strongly typed Go, that provide much of the flexibility that has been exposed in the earlier six chapters. And goroutines, along with channels for communicating between goroutines, provide the primary, but not the only, method for writing concurrent applications. The ninth chapter covers more traditional techniques for concurrency using shared variables.

I enjoyed reading this book, which is not something I can say about most technical books. The style is always direct and understandable, yet at the same time, I felt like I was reading for an advanced class in programming, what I might have encountered in my third year of university. You can learn about more than just Go by reading this book. If you have any interest in learning Go, I strongly recommend that you own this book.

Thanks to Our USENIX and LISA SIG Supporters

USENIX Patrons
Facebook Google Microsoft Research NetApp VMware

USENIX Benefactors
Hewlett-Packard *Linux Pro Magazine* Symantec

USENIX and LISA SIG Partners
Booking.com Cambridge Computer Can Stock Photo Fotosearch Google

USENIX Partners
Cisco-Meraki EMC Huawei

Open Access Publishing Partner
PeerJ

usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION