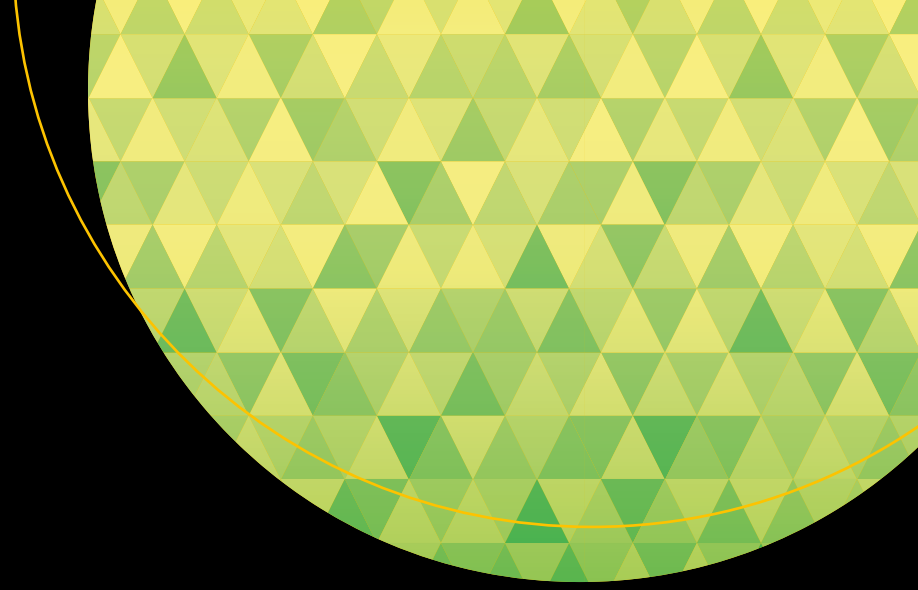

Automated Trading with R

Quantitative Research and Platform Development

Chris Conlan

Apress®

Automated Trading with R

Quantitative Research and Platform Development

Chris Conlan

Apress®

Automated Trading with R: Quantitative Research and Platform Development

Chris Conlan
Bethesda, Maryland
USA

ISBN-13 (pbk): 978-1-4842-2177-8 ISBN-13 (electronic): 978-1-4842-2178-5
DOI 10.1007/978-1-4842-2178-5

Library of Congress Control Number: 2016953336

Managing Director: Welmoed Spahr
Acquisitions Editor: Susan McDermott
Developmental Editor: Laura Berendson
Technical Reviewers: Stephen Nawara, Jeffery Holt
Editorial Board: Steve Anglin, Pramila Balen, Laura Berendson, Aaron Black, Louise Corrigan, Jonathan Gennick, Robert Hutchinson, Celestin Suresh John, Nikhil Karkal, James Markham, Susan McDermott, Matthew Moodie, Natalie Pao, Gwenan Spearing
Coordinating Editor: Rita Fernando
Copy Editor: Kim Wimpsett
Compositor: SPi Global
Indexer: SPi Global
Cover Image: Designed by Freepik

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springer.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a Delaware corporation.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales–eBook Licensing web page at www.apress.com/bulk-sales.

Any source code or other supplementary materials referenced by the author in this text is available to readers at www.apress.com. For detailed information about how to locate your book's source code, go to www.apress.com/source-code/.

Printed on acid-free paper

For my family.

Contents at a Glance

Contents

About the Author

Chris Conlan began his career as an independent data scientist specializing in trading algorithms. He attended the University of Virginia where he completed his undergraduate statistics coursework in three semesters. During his time at UVA, he secured initial fundraising for a privately held high-frequency forex group as president and chief trading strategist. He is currently managing the development of private technology companies in high-frequency forex, machine vision, and dynamic reporting.

About the Technical Reviewers

Dr. Stephen Nawara earned his PhD in pharmacology from Loyola University – Chicago. During the course of his dissertation, he gained five years of experience analyzing biomedical data. He currently works as a data scientist and R tutor. He specializes in applying high-performance computing and machine-learning techniques to automated portfolio management.

Professor Jeffrey Holt has served as the Program Director of the University of Virginia's MS in Data Science and chair of the Department of Statistics, where he is currently the director of the undergraduate program. He received his PhD in Mathematics from the University of Texas. His research concerns analyzing the effects of sampling methods in ecological studies. He teaches classes in machine learning, data manipulation, and mathematics for UVa undergraduate and graduate students.

Acknowledgments

I am grateful to Professor Jeffrey Holt for seeing this book through, from inception to completion. I offer my sincere appreciation to Professor Holt, Gretchen Martinet, and Paul Diver (of the Department of Statistics at the University of Virginia) whose dedicated teaching has inspired me to share my knowledge.

I am thankful to Dr. Stephen Nawara, a gifted programmer and fantastic business partner, for his extraordinary commitment to quality and clarity in his many revisions of this text.

Further, I would like to thank the R developer community and package contributors for donating their time and expertise to maintaining and extending the R language.

Lastly, I cannot thank my family enough for their continual love and support throughout the development of this text and my life as a whole.

Introduction

This book will cover the broad topic of *automated trading*, starting with mathematics and moving to computation and execution. You will gain unique insight into the mechanics and computational considerations taken in building a backtester, strategy optimizer, and fully functional trading platform.

The code examples in this text are derived from deliverables of real consulting and software development contracts. At the end of the book, we will bring the concepts together and build an automated trading platform from scratch. This book will give a prospective algorithm trader everything he needs except a trading account, including full source code.

Definitions

Trading strategies are predetermined sets of rules a trader uses to make trading decisions. Trading strategies use the following tools and techniques:

- *Manual execution* involves the trader placing his trades manually. This can be
  - Calling the brokerage
  - Placing an order through E*Trade, Tradestation, or other brokerage platforms
  - Pit trading
- *Computer automation* involves the trader authorizing a computer to place trades on his behalf. Many retail brokerage platforms and trading software have incorporated this functionality into their platforms, but they are typically very limited. Most brokerages have an API for more customized implementation through the trader's programming language of choice.
  - Tradestation Easy Language, Metatrader
  - Charles Schwab API
  - Black-box algorithms
- *Indicators* are functions of relevant data that inform the trader by interacting with rule sets.
  - MSI
  - Moving averages
  - Custom indicators

- *Rule sets* are logical filters of the indicator that trigger trading decisions. The indicator combined with the rule set comprises the trading strategy.
    - "Buy if the indicator rises above 80."
    - "Short if the indicator crosses two standard deviations below its mean."
    - "Cover short if the indicator crosses zero and the position is net short."

Strategy development is the art of building, testing, optimizing, and maintaining trading strategies. Major topics in strategy development include the following:

- *Backtesting* involves simulating past performance of a given strategy, often with specific parameters of interest. A backtest will yield the performance metric the developer aims to maximize. Backtests may be performed thousands or millions of times in order to optimize parameters in the strategy.
- *Strategy optimization* attempts to determine a strategy in the present that will maximize a performance metric in the future. Optimization methods make trade-offs between computation speed and search completeness.
    - Exhaustive search
    - Gradient methods
    - Genetic search
- *Performance metrics* can be any function of a return series or equity curve that the developer attempts to maximize.
    - Total return
    - Sharpe Ratio
    - Total Return to Max Drawdown Ratio
- *Parameter updating* is part of maintaining a strategy that utilizes real-time performance data to optimize performance. Traders use faster optimization methods and more local searches at this stage.

Scope of This Book

There are a lot of steps in turning a trading idea into a fully automated trading strategy. This book will discuss, from start to finish, the development process through R. With this discussion, this book will cover a broad range of topics in programming, high-performance computing, numerical optimization, finance, and networking.

There will be examples at every step, including full source code in Appendix A. This source code represents the total work product of the topics discussed in the book.

If you have brokerage accounts with the API clients covered in this text, you can plug in your username and password and start trading right away. Obviously, it is important that traders understand what is happening inside their scripts before they begin trading.

Programming in R

R is a language of choice for many data scientists and statisticians at every level. It has a large and rapidly growing community and more than 7,000 contributed packages as of the time of writing. Packages include software suites for data management, machine learning, graphics and plotting, and much more. Installing a new package takes a few seconds and opens up a ton of capabilities within R. If a trader wants to experiment with Lasso regression as an indicator, he can install the `glmnet` package and run Lasso regression with one line of code.

You are not required to have prior experience with R but will benefit from it. Most concepts will be discussed with complementary mathematics, so they can be read and learned without necessarily executing the code. Please see the book's website, `r.chrisconlan.com`, for instructions on downloading and installing R and RStudio.

High-Performance Computing

Any program that works can probably work even faster. In high-performance computing, we aim to minimize computation time by taking full advantage of a computer's resources in an organized fashion.

Most programs we run utilize only one core in our computers. Unless they are doing some very heavy lifting, this is probably best. When we write programs that do a lot of number crunching, we may benefit from distributing the load over multiple cores, known as *parallelizing*. We will see that some jobs are easy to parallelize, and some are not. We will also see that some jobs make huge speed improvements with parallelization, and others are made slower.

Sometimes programs might run very slowly because our computers run out of memory (RAM) and need to access memory on our hard drives (disk space). Storing and fetching information from the disk is a very slow process. We will see how memory management can lead to speed improvements by preventing our data from spilling out of RAM into disk.

Numerical Optimization

Some readers may recall finding the minimum or maximum of a function using basic calculus. This is known as *analytical optimization*. In analytical optimization, we analyze the mathematics to find a solution on paper.

Numerical optimization, on the other hand, involves using high-performance computing and search algorithms to estimate minima or maxima. Some of these algorithms will draw on calculus by estimating high-dimensional derivatives (or gradients), and others will search in an unguided grid-like fashion. We use these algorithms as opposed to calculus because we do not know the form of the performance function or its derivatives.

We will make our biggest speed improvements here by reducing the number of parameters in our trading strategy and selecting the best-suited algorithm to find the maximum of the performance function.

Finance

When building a backtesting algorithm, we must estimate the impact of many real-world financial phenomena to make sure we produce accurate estimates of strategy performance. We will discuss various estimation methods for commissions, margin, slippage, and others in order to produce accurate performance projections in backtesting.

We will address questions like the best time of day to trade, how to find the optimal trading frequency given account constraints, and which risk model validation metrics to use.

Networking

Data providers supply data to all sorts of players in the financial world in real time. Brokerages take messages from clients and execute orders on their behalf. How do traders get their data? And how do brokers get their messages?

To get the data, we will send computer-generated messages to data providers, and they will respond with the data we request. These computer-generated messages work with the providers through an application programming interface (API). With an API, our computers can talk to their computers in a predefined language they understand. It may be through a very long URL or a form of formatted message.

To give brokerages our orders, we will do the same. Most platform-based brokerages have APIs by which traders can program computers to trade on their behalf. Brokerages sometimes require different request and message formats to add security. We will discuss various file transfer and message transfer formats and why certain services use them.

Material Overview

This book will be broken into three major parts. Part I will further clarify the objectives and goals of the book and discuss some interesting analytic problems in strategy trading. Part II will focus on developing the core functionality of the platform. This is where the majority of R programming happens. Part III brings the platform into a production environment by extending and scheduling the platform built in Part II. It will also discuss how our platform measures up to the competition and where to go next to further your education and/or career in strategy development.

Part I: Problem Scope

- *Chapter 1, "Fundamentals of Automated Trading"*: We will continue defining the problem scope of automated trading by mathematically defining the equity curve and return series. We will introduce some popular risk-return metrics and explore their characteristics on simulated equity curves and the S&P 500.

Part II: Building the Platform

- *Chapter 2, "Networking Part I"*: We begin by fetching, storing, and loading the data we will use for analysis and trading throughout the book. We will use URL-based APIs and MySQL-style APIs to build an ASCII database of .csv files of stock data. We will discuss efficient updating, storage, and loading into memory for analysis.
- *Chapter 3, "Data Preparation"*: Here we take the data loaded in Chapter 2 and apply a handful of use-specific cleaning methods. We discuss these methods and generate additional data for use in analysis in later chapters.
- *Chapter 4, "Indicators"*: We discuss the theory and usage of indicators in trading strategies. We introduce the concept of information latency and compute a handful of indicators as examples. You will grow very comfortable with apply-style functions that are the cornerstone of time-series computations in R.
- *Chapter 5, "Rule Sets"*: We discuss the theory and usage of rule sets in trading strategies. We introduce and standardize important terminology for discussing and programming rule sets. We give a lot of attention to which types of indicators work well with which types of rule sets.

- *Chapter 6, "High-Performance Computing"*: This chapter serves as a broad introduction to high-performance computing and a specific guide on high-performance computing in R. This will extend your familiarity with apply-style functions to multicore computing.
- *Chapter 7, "Simulation and Backtesting"*: We will use our combined knowledge thus far to generate simulated trade results from our data, indicators, and rule sets with high-performance methods from Chapter 6.
- *Chapter 8, "Optimization"*: This chapter places Chapter 7 inside a for loop to discover optimal parameters for trading strategies. We spend a lot of time discussing optimal methods for parameter discovery.
- *Chapter 9, "Networking Part II"*: This chapter covers a handful of popular brokerages and how to send orders to them through API calls.

Part III: Production Trading

- *Chapter 10, "Organizing and Automating Scripts"* : We establish CRON jobs in both UNIX and Windows to run your trading strategies automatically on a schedule.
- *Chapter 11, "Looking Forward"*: We discuss the challenges that large-scale funds and high-frequency funds face, what program languages they may use, and generally how to advance a career in automated trading.

Learning Resources

- *Setting up R and RStudio*: `r.chrisconlan.com`
- *Community discussion*: `r.chrisconlan.com`

Risk Disclosure

Apress Media LLC and the author warn there is a high level of risk associated with automated trading in any asset class, and it may not be suitable for all investors. Automation can work against you, as well as to your advantage. Before deciding to invest in automated trading, you should carefully consider your investment objectives, level of experience, and risk appetite. The possibility exists that you could sustain a loss of some or all of your initial investment, and therefore you should not invest money that you cannot afford to lose. There are risks associated with the use of online deal execution and trading systems including but not limited to software and hardware failure and Internet disconnection. You should be aware of all the risks associated with automated trading and consult with an independent financial advisor if you have any doubts.

Apress Media LLC and the author shall not be responsible for any loss arising from any investment based on any recommendation, forecast, or other information provided. Apress Media LLC and the author will not accept liability for any loss or damage, including without limitation to any loss of profit that may arise directly or indirectly from use of or reliance on such information.

The materials printed in this book are solely for informational purposes. No offer or solicitation to buy or sell financial assets, trading advice, or strategy is made, given, or in any manner endorsed by Apress Media LLC and the author. You are fully responsible for any investment or trading decisions you make, and such decisions should be based solely on your evaluation of your financial circumstances, investment/trading objectives, risk tolerance, and liquidity needs.