Yuxing Yan

Python for Finance

Second Edition

Financial modeling and quantitative analysis explained

Packt

Python for Finance - Second Edition

This book uses Python as its computational tool. Since Python is free, any school or organization can download and use it.

This book is organized according to various finance subjects. In other words, the first edition focuses more on Python, while the second edition is truly trying to apply Python to finance.

The book starts by explaining topics exclusively related to Python. Then we deal with critical parts of Python, explaining concepts such as time value of money stock and bond evaluations, capital asset pricing model, multi-factor models, time series analysis, portfolio theory, options and futures.

This book will help us to learn or review the basics of quantitative finance and apply Python to solve various problems, such as estimating IBM's market risk, running a Fama-French 3-factor, 5-factor, or Fama-French-Carhart 4 factor model, estimating the VaR of a 5-stock portfolio, estimating the optimal portfolio, and constructing the efficient frontier for a 20-stock portfolio with real-world stock, and with Monte Carlo Simulation. Later, we will also learn how to replicate the famous Black-Scholes-Merton option model and how to price exotic options such as the average price call option.

Things you will learn:

- Become acquainted with Python in the first two chapters
- Run CAPM, Fama-French 3-factor, and Fama-French-Carhart 4-factor models
- Learn how to price a call, put, and several exotic options
- Understand Monte Carlo simulation, how to write a Python program to replicate the Black-Scholes-Merton options model, and how to price a few exotic options
- Understand the concept of volatility and how to test the hypothesis that volatility changes over the years
- Understand the ARCH and GARCH processes and how to write related Python programs

Packt>
www.packtpub.com

$ **49.99** US
£ **41.99** UK

Prices do not include local sales
Tax or VAT where applicable

ISBN 978-1-78712-569-8
54999
9 781787 125698

Python for Finance - Second Edition

Yuxing Yan

Python for Finance

Second Edition

Financial modeling and quantitative analysis explained

Yuxing Yan

Packt

BIRMINGHAM - MUMBAI

Python for Finance
Second Edition

First published: April 2014

Second edition: June 2017

Production reference: 1270617

Published by Packt Publishing Ltd.
Livery Place
35 Livery Street
Birmingham B3 2PB, UK.

ISBN 978-1-78712-569-8

www.packtpub.com

Credits

Author
Yuxing Yan

Reviewers
Dr. Param Jeet
Nabih Ibrahim Bawazir, M.Sc.
Joran Beasley

Commissioning Editor
Amey Varangaonkar

Acquisition Editor
Tushar Gupta

Content Development Editor
Amrita Noronha

Technical Editor
Akash Patel

Copy Editor
Safis Editing

Project Coordinator
Shweta H Birwatkar

Proofreader
Safis Editing

Indexer
Mariammal Chettiyar

Graphics
Tania Dutta

Production Coordinator
Nilesh Mohite

Cover Work
Nilesh Mohite

About the Author

Yuxing Yan graduated from McGill University with a PhD in finance. Over the years, he has been teaching various finance courses at eight universities: McGill University and Wilfrid Laurier University (in Canada), Nanyang Technological University (in Singapore), Loyola University of Maryland, UMUC, Hofstra University, University at Buffalo, and Canisius College (in the US).

His research and teaching areas include: market microstructure, open-source finance and financial data analytics. He has 22 publications including papers published in the Journal of Accounting and Finance, Journal of Banking and Finance, Journal of Empirical Finance, Real Estate Review, Pacific Basin Finance Journal, Applied Financial Economics, and Annals of Operations Research.

He is good at several computer languages, such as SAS, R, Python, Matlab, and C.

His four books are related to applying two pieces of open-source software to finance: *Python for Finance* (2014), *Python for Finance* (2nd ed., expected 2017), Python for Finance (Chinese version, expected 2017), and *Financial Modeling Using R* (2016).

In addition, he is an expert on data, especially on financial databases. From 2003 to 2010, he worked at Wharton School as a consultant, helping researchers with their programs and data issues. In 2007, he published a book titled Financial Databases (with S.W. Zhu). This book is written in Chinese.

Currently, he is writing a new book called Financial Modeling Using Excel — in an R-Assisted Learning Environment. The phrase "R-Assisted" distinguishes it from other similar books related to Excel and financial modeling. New features include using a huge amount of public data related to economics, finance, and accounting; an efficient way to retrieve data: 3 seconds for each time series; a free financial calculator, showing 50 financial formulas instantly, 300 websites, 100 YouTube videos, 80 references, paperless for homework, midterms, and final exams; easy to extend for instructors; and especially, no need to learn R.

I would like to thank Ben Amoako-Adu, Brian Smith (who taught me the first two finance courses and offered unstinting support for many years after my graduation), George Athanassakos (one of his assignments "forced" me to learn C), and Jin-Chun Duan.

I would also like to thank Wei-Hung Mao, Jerome Detemple, Bill Sealey, Chris Jacobs, Mo Chaudhury, Summon Mazumdar (my former professors at McGill), and Lawrence Kryzanowski. (His wonderful teaching inspired me to concentrate on empirical finance and he edited my doctoral thesis word by word even though he was not my supervisor!). There is no doubt that my experience at Wharton has shaped my thinking and enhanced my skill sets. I thank Chris Schull and Michael Boldin for offering me the job; Mark Keintz, Dong Xu, Steven Crispi, and Dave Robinson, my former colleagues, who helped me greatly during my first two years at Wharton; and Eric Zhu, Paul Ratnaraj, Premal Vora, Shuguang Zhang, Michelle Duan, Nicholle Mcniece, Russ Ney, Robin Nussbaum-Gold, and Mireia Gine for all their help. In addition, I'd like to thank Shaobo Ji, Tong Yu, Shaoming Huang, Xing Zhang.

About the Reviewers

Dr. Param Jeet has a Ph.D. in mathematics from one of India's leading engineering institutes, IIT Madras. Dr. Param Jeet has a decade of experience in the data analytics industry. He started his career with Bank of America and since then worked with a few companies as a data scientist. He has also worked across domains such as capital market, education, telecommunication and healthcare. Dr. Param Jeet has expertise in Quantitative finance, Data analytics, machine learning, R, Python, Matlab, SQL, and big data technologies. He has also published a few research papers in reputed international journals, published and reviewed books, and has worked on *Learning Quantitative Finance with R*.

Nabih Ibrahim Bawazir, M.Sc. is a data scientist at an Indonesian financial technology start-up backed by Digital Alpha Group, Pte Ltd., Singapore. Most of his work is research on the development phase, from financial modeling to data-driven underwriting. Previously, he worked as actuary in CIGNA. He holds M.Sc in Financial Mathematics from Gadjah Mada University, Indonesia.

Joran Beasley received his degree in computer science from the University of Idaho. He works has been programming desktop applications in wxPython professionally for monitoring large scale sensor networks for use in agriculture for the last 7 years. He currently lives in Moscow Idaho, and works at Decagon Devices Inc. as a software engineer.

I would like to thank my wife Nicole, for putting up with my long hours hunched over a keyboard, and her constant support and help in raising our two wonderful children.

Customer Feedback

Thanks for purchasing this Packt book. At Packt, quality is at the heart of our editorial process. To help us improve, please leave us an honest review on this book's Amazon page at `https://www.amazon.com/dp/1787125696`. If you'd like to join our team of regular reviewers, you can e-mail us at `customerreviews@packtpub.com`. We award our regular reviewers with free eBooks and videos in exchange for their valuable feedback. Help us be relentless in improving our products!

Preface

It is our firm belief that an ambitious student major in finance should learn at least one computer language. The basic reason is that we have entered a so-called big data era. In finance, we have a huge amount of data, and most of it is publically available free of charge. To use such rich sources of data efficiently, we need a tool. Among many potential candidates, Python is one of the best choices.

A few words for the second edition

For the second edition, we have reorganized the structure of the book by adding more chapters related to finance. This is recognition and response to the feedbacks from numerous readers. For the second edition, the first two chapters are exclusively devoted to Python. After that, all remaining chapters are associated with finance. Again, Python in this book is used as a tool to help readers learn and understand financial theories better. To meet the demand of using all types of data by various quantitative programs, business analytics programs and financial engineering programs, we add *Chapter 4, Sources of Data*. Because of this restructuring, this edition is more suitable for a one-semester course such as Quantitative Finance, Financial Analysis using Python and Business Analytics. Two finance professors, Premal P. Vora, at Penn State University, Sheng Xiao, at Westminister College, have adopted the first edition as their textbook. Hopefully, more finance, accounting professors would find the second edition is more suitable for their students, especially for those students from a financial engineering program, business analytics and other quantitative areas.

Why Python?

There are various reasons that Python should be used. Firstly, Python is free in terms of license. Python is available for all major operating systems, such as Windows, Linux/Unix, OS/2, Mac, and Amiga, among others. Being free has many benefits. When students graduate, they could apply what they have learned wherever they go. This is true for the financial community as well. In contrast, this is not true for SAS and MATLAB. Secondly, Python is powerful, flexible, and easy to learn. It is capable of solving almost all our financial and economic estimations. Thirdly, we could apply Python to big data. Dasgupta (2013) argues that R and Python are two of the most popular open source programming languages for data analysis. Fourthly, there are many useful modules in Python. Each model is developed for a special purpose. In this book, we focus on NumPy, SciPy, Matplotlib, Statsmodels, and Pandas modules.

A programming book written by a finance professor

There is no doubt that the majority of programming books are written by professors from computer science. It seems odd that a finance professor writes a programming book. It is understandable that the focus would be quite different. If an instructor from computer science were writing this book, naturally the focus would be Python, whereas the true focus should be finance. This should be obvious from the title of the book *Python for Finance*. This book intends to change the fact that many programming books serving the finance community have too much for the language itself and too little for finance. Another unique feature of the book is that it uses a huge amount public data related to economics, finance and accounting, see *Chapter 4, Sources of Data* for more details.

What this book covers

Chapter 1, Python Basics, offers a short introduction, and explains how to install Python, how to launch and quit Python, variable assignment, vector, matrix and Tuple, calling embedded functions, write your own functions, input data from an input file, simple data manipulations, output our data and results, and generate a Python dataset with an extension of pickle.

Chapter 2, Introduction to Python Modules, discusses the meaning of a module, how to import a module, show all functions contained in an imported module, adopt a short name for an imported module, compare between import math and from math import, delete an imported module, import just a few functions from a module, introduction to NumPy, SciPy, matplotlib, statsmodels, pandas and Pandas_reader, find out all built-in modules and all available (preinstalled) modules, how to find a specific uninstalled module.

Chapter 3, Time Value of Money, introduces and discusses various basic concepts and formulae associated with finance, such as present value of one future cash flow, present value of (growing) perpetuity, present and future value of annuity, perpetuity vs. perpetuity due, annuity vs. annuity due, relevant functions contained in SciPy and numpy.lib.financial submodule, a free financial calculator, written in Python, definition of NPV (Net Present Value) and its related rule, definition of IRR (Internal Rate of Return) and its related rule, Python graphical presentation of time value of money, and NPV profile.

Chapter 4, Sources of Data, discusses how to retrieve data from various public sources, such as Yahoo!Finance, Google finance, FRED (Federal Reserve Bank's Economics Data Library), Prof. French's Data Library, BLS (Bureau of Labor Statistics) and Census Bureau. In addition, it would discuss various methods to input data, such as files with formats of csv, txt, pkl, Matlab, SAS or Excel.

Chapter 5, Bond and Stock Valuation, introduces interest rate and its related concepts, such as APR (Annual Percentage Rate), EAR (Effective Annual Rate), compounding frequency, how to convert one effective rate to another one, the term structure of interest rate, how to estimate the selling price of a regular bond, how to use the so-called discount dividend model to estimate the price of a stock and so on.

Chapter 6, Capital Asset Pricing Model, shows how to download data from Yahoo!Finance in order to run a linear regression for CAPM, rolling beta, several Python programs to estimate beta for multiple stocks, adjusted beta and portfolio beat estimation, two beta adjustment methods by Scholes and Williams (1977) Dimson (1979).

Chapter 7, Multifactor Models and Performance Measures, shows how to extend the single-factor model, described in *Chapter 6, Capital Asset Pricing Model*, to multifactor and complex models such as the Fama-French three-factor model, the Fama-French-Carhart four-factor model, and the Fama-French five-factor model, and performance measures such as the Sharpe ratio, Treynor ratios, Sortino ratio, and Jensen's alpha.

Chapter 8, Time-Series Analysis, shows how to design a good date variable, merge datasets by this date variable, normal distribution, normality tests, term structure of interest rate, 52-week high and low trading strategy, return estimation, convert daily returns to monthly or annual returns, T-test, F-test, Durbin-Watson test for autocorrelation, Fama-MacBeth regression, Roll (1984) spread, Amihud's (2002) illiquidity, Pastor and Stambaugh's (2003) liquidity measure, January effect, weekday effect, retrieving high-frequency data from Google Finance and from Prof. Hasbrouck's TORQ database (Trade, Order, Report and Quotation) and introduction to CRSP (Center for Research in Security Prices) database.

Chapter 9, Portfolio Theory, discusses mean and risk estimation of a 2-stock portfolio, N-stock portfolio, correlation vs. diversification effect, how to generate a return matrix, generating an optimal portfolio based on the Sharpe ratio, the Treynor ratio and the Sortinor ratio; how to construct an efficient frontier; Modigliani and Modigliani performance measure (M2 measure); and how to estimate portfolio returns using value-weighted and equal-weighed methodologies.

Chapter 10, Options and Futures, discusses payoff and profit/loss functions for calls and puts and their graphical representations, European versus American options; normal distribution; standard normal distribution; cumulative normal distribution; the famous Black-Scholes-Merton option model with/without dividend; various trading strategies and their visual presentations, such as covered call, straddle, butterfly, and calendar spread; Greeks; the put-call parity and its graphical representation; a graphical representation of a one-step and a two-step binomial tree model; how to use the binomial tree method to price both European and American options; and implied volatility, volatility smile, and skewness.

Chapter 11, Value at Risk, first reviews the density and cumulative functions of a normal distribution, then discusses the first method to estimate VaR based on the normality assumption, conversion from one day risk to n-day risk, one-day VaR to n-day VaR, normality tests, impact of skewness and kurtosis, modifying the VaR measure by including both skewness and kurtosis, the second method to estimate VaR based on historical returns, how to link two methods by using Monte Carlo simulation, back testing, and stress testing.

Chapter 12, Monte Carlo Simulation, discusses how to estimate the π value by using Monte Carlo simulation; simulating stock price movement with a lognormal distribution; constructing efficient portfolios and an efficient frontier; replicating the Black-Scholes-Merton option model by simulation; pricing several exotic options, such as lookback options with floating strikes; bootstrapping with/without replacements; long term expected return forecast and a related efficiency, quasi Monte Carlo simulation, and Sobol sequence.

Chapter 13, Credit Risk Analysis, discusses Moody's, Standard & Poor's, and Fitch's credit ratings, credit spread, 1-year and 5-year migration matrices, term structure of interest rate, Altman's Z-score to predict corporate bankruptcy, the KMV model to estimate total assets and its volatility, default probability and distance to default, and credit default swap.

Chapter 14, Exotic Options, first compares European and American options we learned about in *Chapter 9, Portfolio Theory* with Bermudan options, then discusses methods to price simple chooser options; shout, rainbow, and binary options; the average price option; barrier options such as the up-and-in option and the up-and-out option; and barrier options such as down-and-in and down-and-out options.

Chapter 15, Volatility, Implied Volatility, ARCH, and GARCH, focuses on two issues: volatility measures and ARCH/GARCH.

Small-program oriented

Based on the author's teaching experience at seven schools, McGill and Wilfrid Laurier University (in Canada), NTU (in Singapore), and Loyola University, Maryland, UMUC, Hofstra University, and Canisius College (in the United States), and his eight-year consulting experience at Wharton School, he knows that many finance students like small programs that solve one specific task. Most programming books offer just a few complete and complex programs. The number of programs is far too less than enough few. There are two side effects to such an approach. First, finance students are drowned in programming details, get intimidated, and eventually lose interest in learning a computer language. Second, they don't learn how to apply what they just learned, such as running a capital asset pricing model (CAPM) to estimate IBM's beta from 1990 to 2013. This book offers about 300 complete Python programs around many finance topics.

Using real-world data

Another shortcoming of the majority of books for programming is that they use hypothetical data. In this book, we use real-world data for various financial topics. For example, instead of showing how to run CAPM to estimate the beta (market risk), I show you how to estimate IBM's, Apple's, or Walmart's betas. Rather than just presenting formulae that shows you how to estimate a portfolio's return and risk, the Python programs are given to download real-world data, form various portfolios, and then estimate their returns and risk, including Value at Risk (VaR). When I was a doctoral student, I learned the basic concept of volatility smiles. However, until writing this book, I had a chance to download real-world data to draw IBM's volatility smile.

What you need for this book

Here, we use several concrete examples to show what a reader could achieve after going through this book carefully.

First, after reading the first two chapters, a reader/student should be able to use Python to calculate the present value, future value, present value of annuity, IRR (internal rate of return), and many other financial formulae. In other words, we could use Python as a free ordinary calculator to solve many finance problems. Second, after the first three chapters, a reader/student or a finance instructor could build a free financial calculator, that is, combine a few dozen small Python programs into a big Python program. This big program behaves just like any other module written by others. Third, readers learn how to write Python programs to download and process financial data from various public data sources, such as Yahoo! Finance, Google Finance, Federal Reserve Data Library, and Prof. French's Data Library.

Fourth, readers will understand basic concepts associated with modules, which are packages written by experts, other users, or us, for specific purposes. Fifth, after understanding the Matplotlib module, readers can produce various graphs. For instance, readers could use graphs to demonstrate payoff/profit outcomes based on various trading strategies by combining the underlying stocks and options. Sixth, readers will be able to download IBM's daily price, the S&P 500 index price, and data from Yahoo! Finance and estimate its market risk (beta) by applying CAPM. They will also be able to form a portfolio with different securities, such as risk-free assets, bonds, and stocks. Then, they can optimize their portfolios by applying Markowitz's mean-variance model. In addition, readers will know how to estimate the VaR of their portfolios.

Seventh, a reader should be able to price European and American options by applying both the Black-Scholes-Merton option model for European options only, and the Monte Carlo simulation for both European and American options. Last but not least, readers will learn several ways to measure volatility. In particular, they will learn how to use AutoRegressive Conditional Heteroskedasticity (ARCH) and Generalized AutoRegressive Conditional Heteroskedasticity (GARCH) models.

Who this book is for

If you are a graduate student majoring in finance, especially studying computational finance, financial modeling, financial engineering, or business analytics, this book will benefit you greatly. Here are two examples: Prof. Premal P. Vora at Penn State University has used this book for his course titled *Data Science in Finance*, and Prof. Sheng Xiao at Westminister College has done so for his course titled *Financial Analytics*. If you are a professional, you could learn Python and use it in many financial projects. If you are an individual investor, you could benefit from reading this book as well.