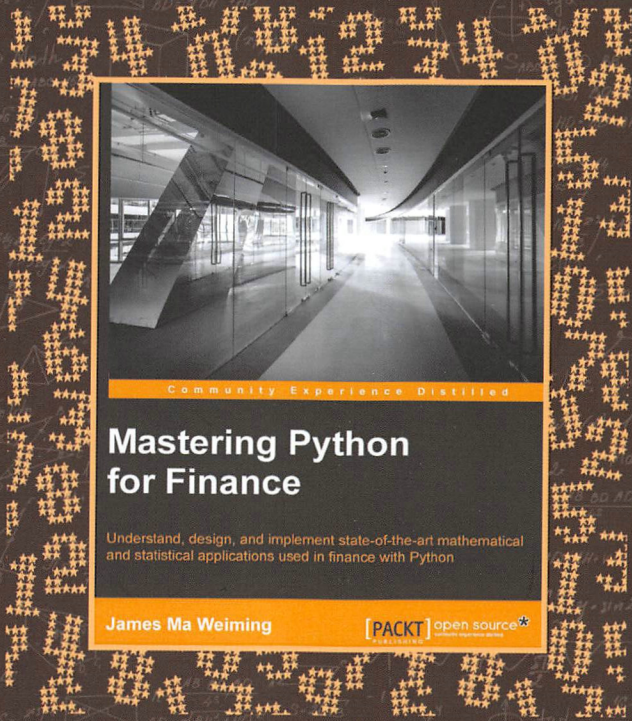




Python金融 数据分析

[新加坡] 马伟明 (James Ma Weiming) 著

张永翼 霍达 张彤 译



MASTERING PYTHON FOR FINANCE



机械工业出版社
China Machine Press



内容简介

本书系统阐述Python在金融领域的应用，不仅涵盖核心的金融理论及相关数学概念，还详细讲解如何应用Python求解经典的资产定价模型、解决金融中的线性和非线性问题、开发数值程序和利率模型，以及如何根据有限差分法定价来描绘含有期权的隐含波动率曲线等。

本书还介绍了Hadoop在大数据分析中的应用以及Python和Excel的融通，包括使用Python执行MapReduce操作，用NoSQL存储数据，在Python中构建一个组件对象模型服务器和客户端界面与Excel融通，以及在Excel中即时计算期权价格。

此外，通过学习本书，你将了解如何连接到代理API，检索市场数据，生成交易信号并向交易所发送订单，以及平均回报和趋势跟踪等交易策略的实施。另外，本书还将介绍风险管理、头寸跟踪和回测技术，以帮助你管理交易策略的实施效果。



华章 IT
HZBOOKS | Information Technology







数据科学与工程 技术丛书

MASTERING PYTHON
FOR FINANCE

Python 金融 数据分析

[新加坡] 马伟明 (James Ma Weiming) 著

张永冀 霍达 张彤 译



机械工业出版社
China Machine Press

试读结束 需要全本请在线购买: www.ertongbook.com



图书在版编目 (CIP) 数据

Python 金融数据分析 / (新加坡) 马伟明 (James Ma Weiming) 著; 张永翼, 霍达, 张彤译. —北京: 机械工业出版社, 2018.3

(数据科学与工程丛书)

书名原文: Mastering Python for Finance

ISBN 978-7-111-58998-3

I. P… II. ① 马… ② 张… ③ 霍… ④ 张… III. 软件工具—程序设计—应用—金融—分析 IV. F830.41-39

中国版本图书馆 CIP 数据核字 (2018) 第 041836 号

本书版权登记号: 图字 01-2015-7589

James Ma Weiming: *Mastering Python for Finance* (ISBN: 978-1-78439-451-6).

Copyright © 2015 Packt Publishing. First published in the English language under the title “Mastering Python for Finance”.

All rights reserved.

Chinese simplified language edition published by China Machine Press.

Copyright © 2018 by China Machine Press.

本书中文简体字版由 Packt Publishing 授权机械工业出版社独家出版。未经出版者书面许可, 不得以任何方式复制或抄袭本书内容。

Python 金融数据分析

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 缪 杰

责任校对: 李秋荣

印 刷: 中国电影出版社印刷厂

版 次: 2018 年 3 月第 1 版第 1 次印刷

开 本: 185mm × 260mm 1/16

印 张: 15

书 号: ISBN 978-7-111-58998-3

定 价: 69.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzit@hzbook.com

版权所有 · 侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东



前 言

Python 已广泛应用于银行业、投资管理、保险业、房地产行业等金融领域，用于开发金融模型、管理风险和自动完成交易。许多大型金融机构依赖 Python 来搭建职位管理、资产定价、风险管理和交易系统等基础设施。

本书将介绍核心的金融理论，并给出它们的数学概念，以帮助读者更好地理解它们在实际中的应用价值。你将了解如何应用 Python 求解经典的资产定价模型，解决金融中的线性和非线性问题，开发数值程序和利率模型，以及如何根据有限差分法定价来描绘含有期权的隐含波动率曲线等。

随着高级计算技术的出现，我们必须要考虑如何存储和处理大量数据。而 Hadoop 是目前处理大数据的流行工具。因此本书将介绍 Hadoop 的工作原理及其与 Python 的集成，以获得金融数据的分析方法；以及如何利用 Python 实现 NoSQL 在存储非结构化数据中的应用。

目前许多公司开始向客户提供 API，以使用他们定制的交易软件进行交易。通过学习本书，你将了解如何连接到代理 API，检索市场数据，生成交易信号并向交易所发送指令，以及平均回报和趋势跟踪等交易策略的实施。另外，本书还将介绍风险管理、头寸跟踪和回溯测试技术，以帮助你管理交易策略的实施效果。

金融行业中，使用 Microsoft Excel 处理债券交易和后台业务是一种普遍现象。本书将介绍如何在 Python 中创建数字定价组件对象模型（COM）服务器，使你的电子表格能够即时计算和更新模型值。

本书的主要内容

第 1 章探讨了 Python 在金融领域的适用性，引入 IPython 作为可视化数据和执行科学计算的有效工具。



IV

第2章介绍了使用 Python 求解线性方程组的方法，执行整数规划，以及将矩阵应用于投资组合分析的线性优化。

第3章讨论了使用 Python 构建金融非线性模型以及术根方法。

第4章探讨了如何使用二叉树模型、二叉树网格和有限差分法等进行期权估值。

第5章讨论了收益率曲线的引导过程，涵盖一些利用 Python 实现的衍生品利率的短期定价模型。

第6章讨论了波动率指数，对欧洲斯托克 50 指数波动率数据进行分析，并使用子指数的期权价格复制主要指数。

第7章介绍了 Hadoop 在大数据分析中的应用，如何使用 Python 执行 MapReduce 操作，以及如何使用 NoSQL 存储数据。

第8章探讨了逐步开发均值回归算法交易和趋势跟踪算法交易策略，以及交易系统风险管理等。

第9章讨论如何设计并实现一个事件驱动的回溯测试系统，帮助你把握模拟交易策略的性能。

第10章介绍了如何在 Python 中构建一个组件对象模型服务器和客户端界面与 Excel 融通，以及如何在 Excel 中即时计算期权价格。

学习本书的软硬件支持

学习本书需要安装如下软件：

□ 操作系统：

- 能使用 Python 2.7 或更高版本的操作系统
- 第10章需要 Windows XP 或更高版本的操作系统
- 第7章需要有至少 4GB RAM 的 64 位主机操作系统

□ 本书将使用以下 Python 包的 Python、SciPy、pandas、IPython 和 Matplotlib 模块：

- Continuum Analytics 的 Anaconda 2.1 或更高版本：<https://store.continuum.io/cshop/anaconda/>
- Enthought 的 Canopy 1.5 或更高版本：<https://store.enthought.com/downloads/>

□ 其他必需的 Python 模块：

- Statsmodels, 见 <http://statsmodels.sourceforge.net/>
 - PuLP (第2章), 见 <https://github.com/coin-or/pulp>
 - lxml (第6章), 见 <http://lxml.de/>
 - PyMongo 2.7 (第7章), 见 <https://pypi.python.org/pypi/pymongo/>
 - IbPy (第8章), 见 <https://github.com/blampe/IbPy>
 - oandapy (第8章), 见 <https://github.com/oanda/oandapy>
 - python-requests (第8章), 见 <https://pypi.python.org/pypi/requests/>
 - PyWin32 (第10章), 见 <http://sourceforge.net/projects/pywin32/files/>
- 可选 Python 模块:
- 使用 pip 6.0 自动安装 Python 包, 见 <https://pypi.python.org/pypi/pip>
- 需要的软件:
- Mozilla Firefox, 见 <https://www.mozilla.org/en-US/firefox/new/>
 - MongoDB 2.6 (第7章), 见 <http://www.mongodb.org/downloads>
 - VirtualBox 4.3 (第7章), 见 <https://www.virtualbox.org/wiki/Downloads>
 - Cloudera QuickStart VM with CDH (Cloudera Distribution Including Apache Hadoop) (第7章), 见 http://www.cloudera.com/content/cloudera/en/downloads/quickstart_vms.html
 - Interactive Brokers (IB) Trader Workstation (TWS) (第8章), 见 <https://www.interactivebrokers.com/en/index.php?f=1537>
 - 使用 Oracle Java 7 运行 IB TWS 和 OANDA fxTrade 平台 (第8章)
 - Microsoft Office Excel 2010 或更高版本, 并使用宏 (第10章)

本书的读者对象

本书面向开发金融应用程序的学生和程序员, 提供金融服务的顾问, 金融分析师以及想要利用 Python 在数据可视化、交互式分析和科学计算方面的优势进行财务分析的人员。对此, 你需要掌握一定的 Python 基础知识和金融概念, 在学习每一章的技术内容之前, 本书会为初学者介绍相关的背景资料。

目 录

前言

第 1 章 Python 在金融中的应用

应用	1
1.1 Python 适合我吗	2
1.1.1 免费 + 开源	2
1.1.2 高级、强大、灵活的 编程语言	2
1.1.3 丰富的标准库	3
1.2 面向对象编程与函数式 编程	3
1.2.1 面向对象式方法	3
1.2.2 函数式方法	4
1.2.3 我该使用哪种方法	4
1.3 我该使用哪个版本的 Python	5
1.4 IPython 简介	5
1.4.1 安装 IPython	5
1.4.2 使用 pip	6
1.4.3 IPython Notebook	6
1.4.4 Notebook 单元格	8
1.4.5 IPython Notebook 简单 的练习	9
1.4.6 Notebook 与金融	14
1.5 总结	15

第 2 章 金融中的线性问题

2.1 资本资产定价模型与证券 市场线	17
2.2 套利定价模型	20
2.3 因子模型的多元线性回归	20
2.4 线性最优化	21
2.4.1 安装 PuLP	22
2.4.2 一个简单的线性优化 问题	22
2.4.3 线性规划的结果	24
2.4.4 整数规划	24
2.5 使用矩阵解线性方程组	27
2.6 LU 分解	29
2.7 Cholesky 分解	30
2.8 QR 分解	31
2.9 总结	36

第 3 章 非线性与金融

3.1 非线性建模	37
3.2 非线性模型举例	38
3.2.1 隐含波动率模型	38
3.2.2 马尔可夫机制转换模型	39
3.2.3 门限自回归模型	40
3.2.4 平滑转换模型	41
3.3 非线性模型求根算法概述	41

3.4	增量法	42	4.6.4	奇异障碍期权定价	83
3.5	二分法	44	4.6.5	美式期权定价的有限 差分	85
3.6	牛顿迭代法	46	4.7	隐含波动率模型	88
3.7	割线法	47	4.8	总结	91
3.8	求根法的结合使用	49	第5章 利率及其衍生工具		92
3.9	利用 SciPy 求解	49	5.1	固定收益证券	93
3.9.1	SciPy 求根标量函数	49	5.2	收益率曲线	93
3.9.2	通用非线性求解器	50	5.3	无息债券	94
3.10	总结	52	5.4	自助法构建收益率曲线	95
第4章 利用数值方法为衍生 品定价		53	5.5	远期利率	98
4.1	什么是期权	54	5.6	计算到期收益率	100
4.2	二叉树期权定价模型	54	5.7	计算债券定价	101
4.2.1	欧式期权定价	54	5.8	久期	101
4.2.2	编写 StockOption 类	56	5.9	凸度	102
4.2.3	编写 BinomialEuropean- Option 类	57	5.10	短期利率模型	103
4.2.4	利用 BinomialTreeOption 类给美式期权定价	59	5.10.1	Vasicek 模型	104
4.2.5	Cox-Ross-Rubinstein 模型	61	5.10.2	Cox-Ingersoll-Ross 模型	105
4.2.6	Leisen-Reimer 模型	62	5.10.3	Rendleman and Bartter 模型	106
4.3	希腊值	64	5.10.4	Brennan and Schwartz 模型	107
4.4	三叉树期权定价模型	67	5.11	债券期权	108
4.5	期权定价中的 Lattice 方法	70	5.11.1	可赎回债券	109
4.5.1	二叉树网格	70	5.11.2	可回售债券	109
4.5.2	编写 BinomialCRR- Option 类	70	5.11.3	可转换债券	109
4.5.3	三叉树网格	72	5.11.4	优先股	110
4.6	有限差分法	73	5.12	可赎回债券定价	110
4.6.1	显式方法	75	5.12.1	Vasicek 模型定价 无息债券	110
4.6.2	隐式方法	78	5.12.2	提前行权定价	112
4.6.3	Crank-Nicolson 方法	81	5.12.3	有限差分策略 迭代法	114

5.12.4 可赎回债券定价的 其他影响因素	120	7.2.1 HDFS	148
5.13 总结	121	7.2.2 YARN	149
第6章 利用 Python 分析欧 洲斯托克 50 指数波 动率	122	7.2.3 MapReduce	149
6.1 波动率指数衍生品	123	7.3 大数据工具对我来说 实用吗	149
6.1.1 STOXX 与欧洲期货 交易所	123	7.4 获取 Apache Hadoop	150
6.1.2 EURO STOXX 50 指数	123	7.4.1 从 Cloudera 获取 QuickStart VM	150
6.1.3 VSTOXX	123	7.4.2 获取 VirtualBox	150
6.1.4 VIX	124	7.4.3 在 VirtualBox 上运行 Cloudera VM	150
6.2 获取 EUROX STOXX 50 指数和 VSTOXX 数据	124	7.5 Hadoop 中的字计数程序	153
6.3 数据合并	128	7.5.1 下载示例数据	153
6.4 SX5E 与 V2TX 的财务 分析	129	7.5.2 map 程序	153
6.5 SX5E 与 V2TX 的相关性	132	7.5.3 reduce 程序	154
6.6 计算 VSTOXX 子指数	134	7.5.4 测试脚本	155
6.6.1 获取 OESX 数据	134	7.5.5 在 Hadoop 上运行 MapReduce	156
6.6.2 计算 VSTOXX 子指数的 公式	136	7.5.6 使用 Hue 浏览 HDFS	158
6.6.3 VSTOXX 子指数值的 实现	137	7.6 Hadoop 的金融实践	158
6.6.4 分析结果	142	7.6.1 从 Yahoo! Finance 获取 IBM 股票价格	159
6.7 计算 VSTOXX 主指数	143	7.6.2 修改 map 程序	159
6.8 总结	145	7.6.3 使用 IBM 股票价格 测试 map 程序	160
第7章 大数据分析	147	7.6.4 运行 MapReduce 计算 日内价格变化	160
7.1 什么是大数据	148	7.6.5 分析 MapReduce 结果	162
7.2 Hadoop	148	7.7 NoSQL 简介	162
		7.7.1 获取 MongoDB	163

7.7.2	创建数据目录并运行 MongoDB	163	8.6.2	处理事件	183
7.7.3	获取 PyMongo	164	8.6.3	实现均值回归算法	184
7.7.4	运行测试连接	164	8.6.4	跟踪头寸	186
7.7.5	获取数据库	165	8.7	使用 OANDA API 进行外汇交易	187
7.7.6	获取集合	165	8.7.1	什么是 REST	188
7.7.7	插入文档	166	8.7.2	设置 OANDA 账户	188
7.7.8	获取单个文档	166	8.7.3	OANDA API 使用方法	191
7.7.9	删除文档	167	8.7.4	获取 oandapy——OANDA REST API 包装器	191
7.7.10	批量插入文档	167	8.7.5	获取并解析汇率数据	191
7.7.11	统计集合文档	168	8.7.6	发送指令	192
7.7.12	查找文档	168	8.8	构建趋势跟踪外汇交易平台	193
7.7.13	文档排序	168	8.8.1	设置主程序	193
7.7.14	结论	169	8.8.2	处理事件	194
7.8	总结	169	8.8.3	实现趋势跟踪算法	195
8.1	什么是算法交易	172	8.8.4	跟踪头寸	195
8.2	带有公共 API 的交易平台列表	173	8.9	风险价值模型	197
8.3	有没有最好的编程语言	173	8.10	总结	199
8.4	系统功能	173	第 9 章	回溯测试	201
8.5	通过 Interactive Brokers 和 IbPy 进行算法交易	174	9.1	回溯测试概述	201
8.5.1	获取 Interactive Brokers 的 Trader WorkStation	174	9.1.1	回溯测试的缺陷	202
8.5.2	获取 IbPy——IB API 包装器	177	9.1.2	事件驱动回溯测试系统	202
8.5.3	指令路由机制	177	9.2	设计并实施回溯测试系统	204
8.6	构建均值回归算法交易系统	181	9.2.1	TickData 类	204
8.6.1	设置主程序	181	9.2.2	MarketData 类	204
			9.2.3	MarketDataSource 类	205

9.2.4	Order 类	206	9.5	总结	217
9.2.5	Position 类	206	第 10 章 Python 与 Excel 的		
9.2.6	Strategy 类	207	融通		
9.2.7	MeanReverting Strategy 类	207	10.1	COM 概述	220
9.2.8	Backtester 类	209	10.2	Excel 与金融	220
9.2.9	运行回溯测试系统	211	10.3	构建 COM 服务器	220
9.2.10	改进回溯测试系统	213	10.3.1	先决条件	221
9.3	回溯测试模型的 10 个注意 事项	213	10.3.2	获取 pythoncom 模块	221
9.3.1	模型的资源限制	214	10.3.3	构建 Black-Scholes 模型 COM 服务器	221
9.3.2	模型评价标准	214	10.3.4	注册和注销 COM 服务器	222
9.3.3	估计回溯测试参数的 质量	214	10.3.5	构建 Cox-Ross- Rubinstein 模型 COM 服务器	223
9.3.4	应对模型风险	214	10.3.6	构建二叉网格模型 COM 服务器	223
9.3.5	样本数据回测	214	10.4	在 Excel 中构建 COM 客户端	224
9.3.6	解决回溯测试的常见 缺陷	214	10.4.1	设置 VBA 代码	225
9.3.7	常识错误	215	10.4.2	设置单元格	227
9.3.8	理解模型环境	215	10.5	COM 的其他功能	228
9.3.9	数据准确性	215	10.6	总结	229
9.3.10	数据挖掘	216			
9.4	回溯测试中的算法选择	216			
9.4.1	k-均值聚类算法	216			
9.4.2	KNN 机器学习算法	216			
9.4.3	分类回归树分析	217			
9.4.4	2k 析因设计	217			
9.4.5	遗传算法	217			

第 1 章

Python 在金融中的应用

本章将探讨 Python 作为金融编程语言的实用性。时至今日，Python 已经在银行业、投资管理和保险业等金融领域取得广泛应用，甚至帮助房地产行业开发金融建模、风险管理和交易的程序。为了更好地体验 Python 的强大功能，本书推荐使用 IPython Notebook，它能够更有效地实现数据可视化，将科学计算过程呈现给最终用户。

本章讨论以下主题：

- Python 相较于其他金融编程语言的优势
- Python 在金融应用中的功能
- Python 的编程和实现范式：面向对象设计（object-oriented design）和函数式设计（functional design）
- IPython 概述
- IPython 和 IPython Notebook 入门
- 创建与存储 Notebook 文档
- 导出多种格式 Notebook 文档
- Notebook 文档用户界面
- 在 Notebook 文档中插入 Markdown 语言
- 在 Notebook 文档中利用 Python 实现运算
- 在 Notebook 文档中创建函数图
- 在 Notebook 文档中显示数学方程式的多种方式
- 在 Notebook 文档中插入图像和视频文件
- 在 Notebook 文档中使用 HTML 和 pandas DataFrame

1.1 Python 适合我吗

当前，金融 IT 工程师能够通过多种计算机语言实现丰富的软件解决方案，如 C、Java、R 和 MATLAB。但每种语言是为实现特定任务开发的，各语言内部工作原理、行为、句法规则和性能差异都会影响用户的使用结果。

本书将重点介绍 Python 在金融计量和分析中的应用。作为原用于科学运算的计算机语言，Python 借助 AQR 资产管理开发的 Pandas 开源软件库，为金融数据管理和量化分析提供了高效平台。

大型金融机构也利用 Python 建立基础系统架构。比如美国银行“石英”项目通过 Python 实现头寸位管理、定价和风险管理功能；摩根大通利用 Python 的兼容性和灵活性，融合 C++ 和 Java，建立跨市场的风险管理和交易系统——“雅典娜”。

本书将介绍借助 Python 开发金融应用的基本内容，包括投资组合最优化、数值定价、交互式分析、利用 Hadoop 进行大数据投资决策等。

使用 Python 开发金融应用的优势体现在以下几个方面。

1.1.1 免费 + 开源

Python 许可证以及互联网在线 Python 社区、文档均可免费使用，任何人都能根据实际情况浏览并修改免费的开放源码。

Python 可安装于所有主流操作系统，如 Windows、UNIX、OS/2、Mac 等。它的开放性创造了前所未有的新机遇，任何人都有机会增强现有模块或创建新模块。高级用户可实现跨语言平台编程。借助适合的库，Python 解释器可集成 C、C++、Fortran、Lisp、PHP、Lua 等不同编程语言。

1.1.2 高级、强大、灵活的编程语言

用户借助 Python 这种高级通用的编程语言，可以快速开发原型程序，避免内存管理等低层次机制构造问题。

Python 支持不同的编程和实现范式，如面向对象式、函数式和过程式编程。基于这种灵活性，它在运行包含多个可变参数的复杂数学模型时尤为实用。

1.1.3 丰富的标准库

NumPy、SciPy、matplotlib、statsmodels 和 pandas 是定量分析和数据管理经常使用的模块。

其他库可以拓展 Python 的功能，例如，gnuplot 软件包能够将 Python 变为数据可视化工具，以可视化数学函数和数据交互；基于 Tk 的 GUI 工具（如 Tkinter）可将 Python 脚本变为 GUI 程序。

IPython 是目前广泛使用的 Python 图形界面，最初用于处理 Python 数据，以其高效的交互式计算工具提供并行与分布式计算。IPython Notebook 是 IPython 的 Web 浏览器版本，可以共享代码、文本、数学表达式、图像和其他丰富的媒体资料。IPython 设计的初衷是让科学家使用 Python 和数据。

1.2 面向对象编程与函数式编程

如果你是一名金融程序员，你编写的程序可能用来处理价值成千上万美元的交易，必须保证零出错。金融模型和软件系统越来越复杂，优秀的软件设计尤为重要。出于易读性的考虑，编写 Python 代码可以选择面向对象式或函数式方法。

1.2.1 面向对象式方法

随着对程序清晰、快速和灵活性需求的增加，保持代码可读性、可管理性愈发重要。一种常见方法是面向对象式编程构建软件系统。如以下例子所示，将 Greeting 定义为一个类：

```
class Greeting(object):  
  
    def __init__(self, my_greeting):  
        self.my_greeting = my_greeting  
  
    def say_hello(self, name):  
        print "%s %s" % (self.my_greeting, name)
```



下载示例代码

凡在 <http://www.packtpub.com> 官网购买书籍的客户，可登录账号，自行下载书中配套代码文件。若在其他渠道购买本书，可访问 <http://www.packtpub.com/support> 并注册，我们将把代码文件直接发送到您邮箱。