Certificate Program in Python for Finance —Algorithmic Trading, Computational Finance, and Asset Management

Introduction & Overview

Dr. Yves J. Hilpisch

THE PYTHON QUANTS

The AI Machine

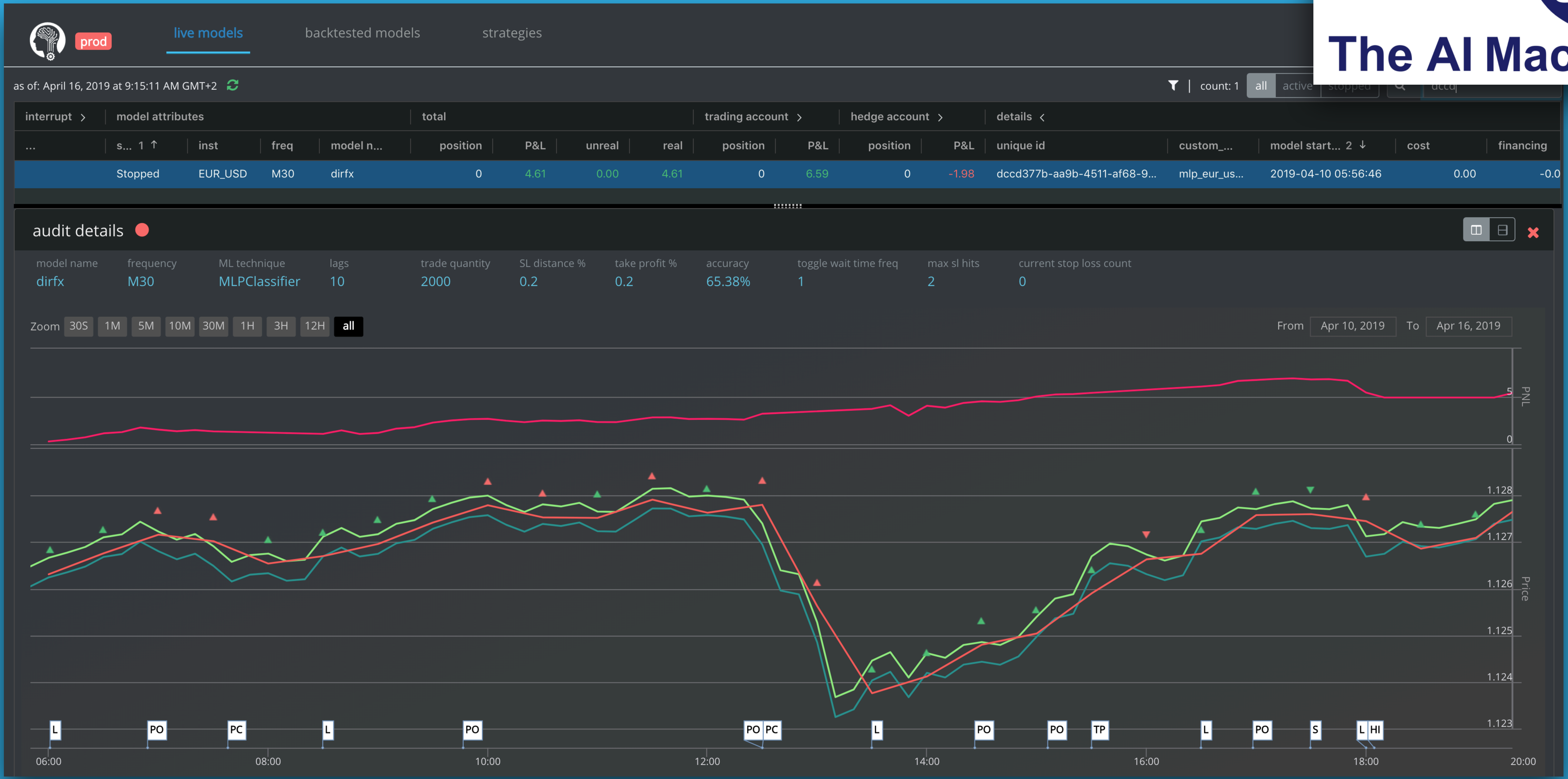
AGENDA

- Introduction
- The Program
- Quant Platform 2.0
- Mathematics Basics
- Python for Finance Basics
- Crypto Basics
- Finance with Python
- Tools and Skills
- Financial Packages
- Python for Financial Data Science
- Python for Excel
- Python for Databases
- Natural Language Processing
- Artificial Intelligence in Finance
- Reinforcement Learning for Finance
- Python for Algorithmic Trading
- Python for Computational Finance
- Python for Asset Management
- Case Studies & Demos
- Study Plans for the Programs
- Guiding Principles
- Reviews, Exercises & Test Projects
- User Forum (Technical Support)
- Discord Server (Realtime Chat)

Introduction

THE PYTHON QUANTS

http://tpq.io



The AI Machine

http://aimachine.io

Dr. Yves J. Hilpisch is the founder and CEO of **The Python Quants** (http://tpq.io), a group focusing on the use of open source technologies for financial data science, artificial intelligence, algorithmic trading, and computational finance. He is also the founder and CEO of **The AI Machine** (http://aimachine.io), a company focused on AI-powered algorithmic trading based on a proprietary strategy execution platform.

Yves has a Diploma in Business Administration, a Ph.D. in Mathematical Finance, and is Adjunct Professor for Computational Finance.

Yves is the author of six books (https://home.tpq.io/books):

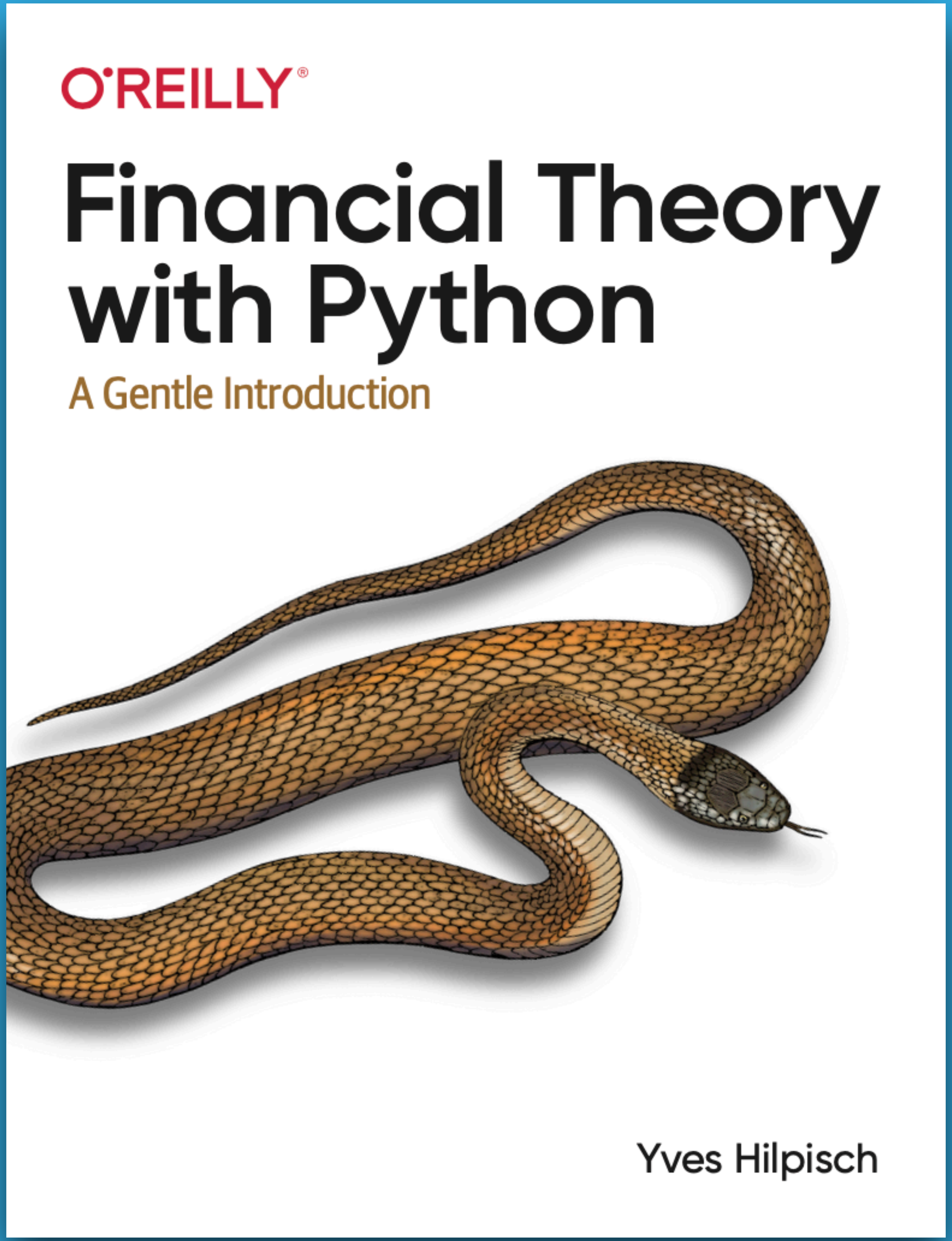
* Finance with Python (2021, O'Reilly)
* Artificial Intelligence in Finance (2020, O'Reilly)
* Python for Algorithmic Trading (2020, O'Reilly)
* Python for Finance (2018, 2nd ed., O'Reilly)
* Listed Volatility and Variance Derivatives (2017, Wiley Finance)
* Derivatives Analytics with Python (2015, Wiley Finance)

Yves is the director of the first online training program leading to **University Certificates in Python for Algorithmic Trading** (https://home.tpq.io/certificates/pyalgo), **Computational Finance** (https://home.tpq.io/certificates/compfin), and **Asset Management** (https://home.tpq.io/certificates/compfin). He also lectures on computational finance, machine learning, and algorithmic trading at the **CQF Program** (http://cqf.com).

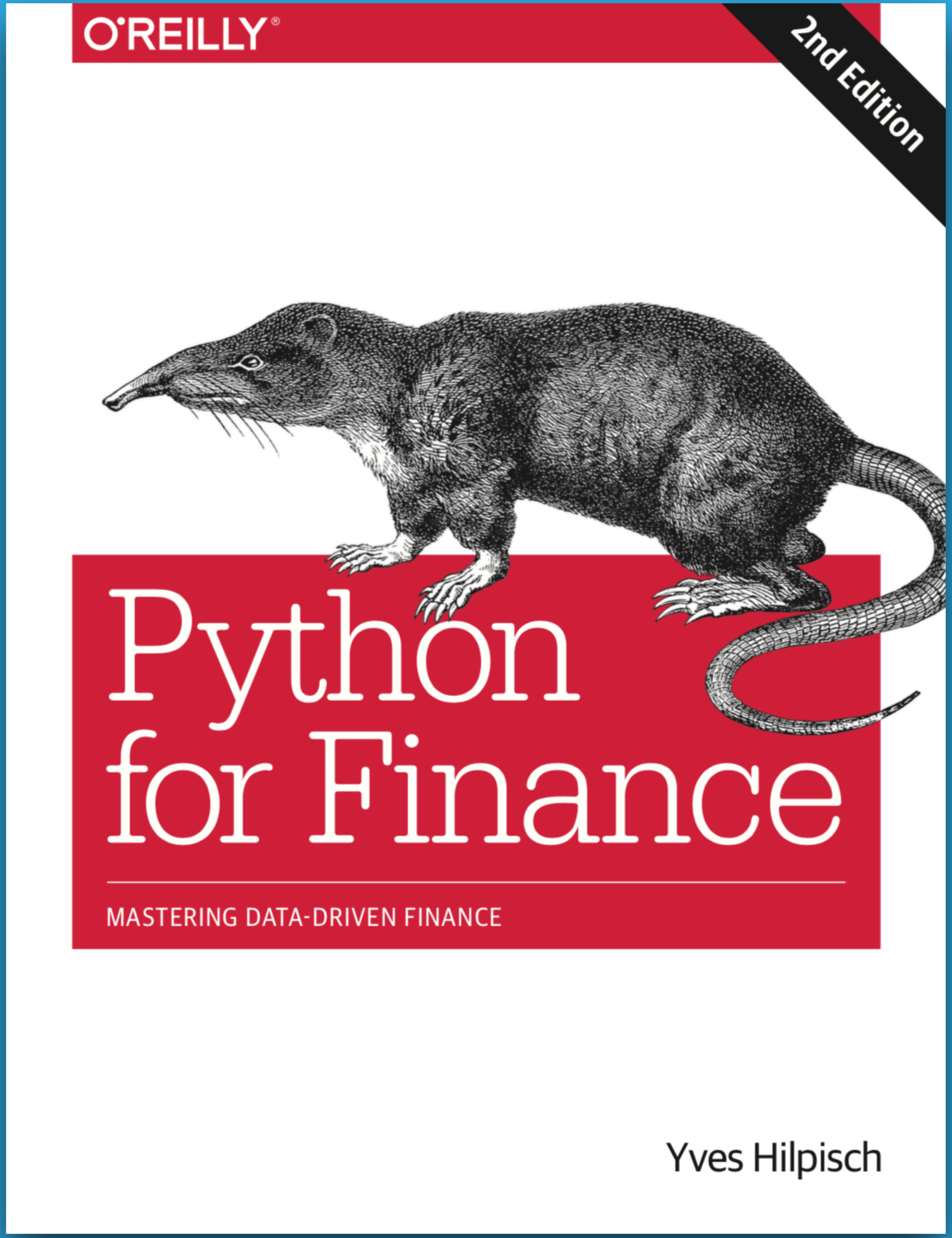
Yves is the originator of the financial analytics library **DX Analytics** (http://dx-analytics.com) and organizes Meetup group **events, conferences, and bootcamps** about Python, artificial intelligence and algorithmic trading in London (http://pqf.tpq.io), New York (http://aifat.tpq.io), Frankfurt, Berlin, and Paris. He has given **keynote speeches** at technology conferences in the United States, Europe, and Asia.

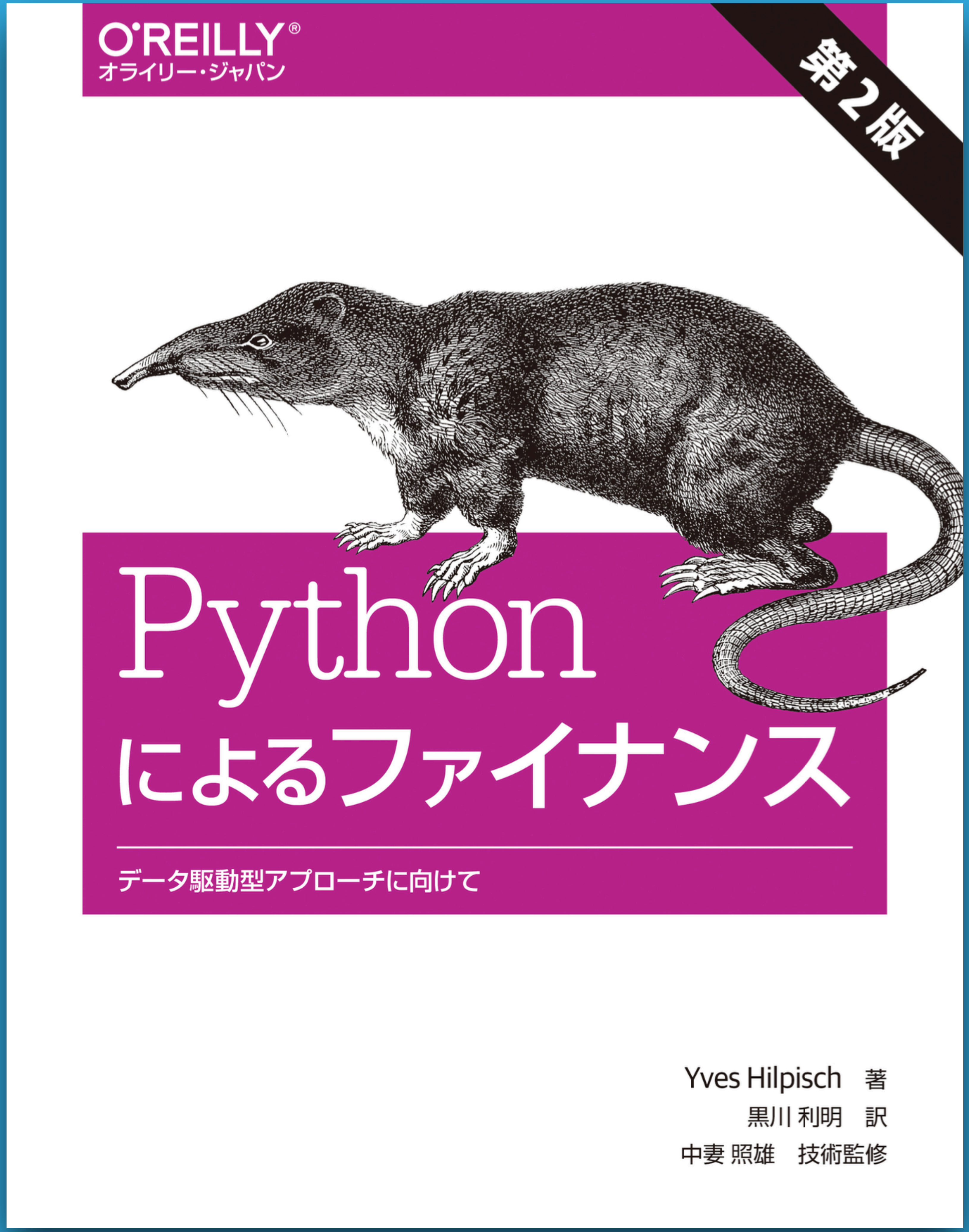
http://hilpisch.com

Financial Theory with Python



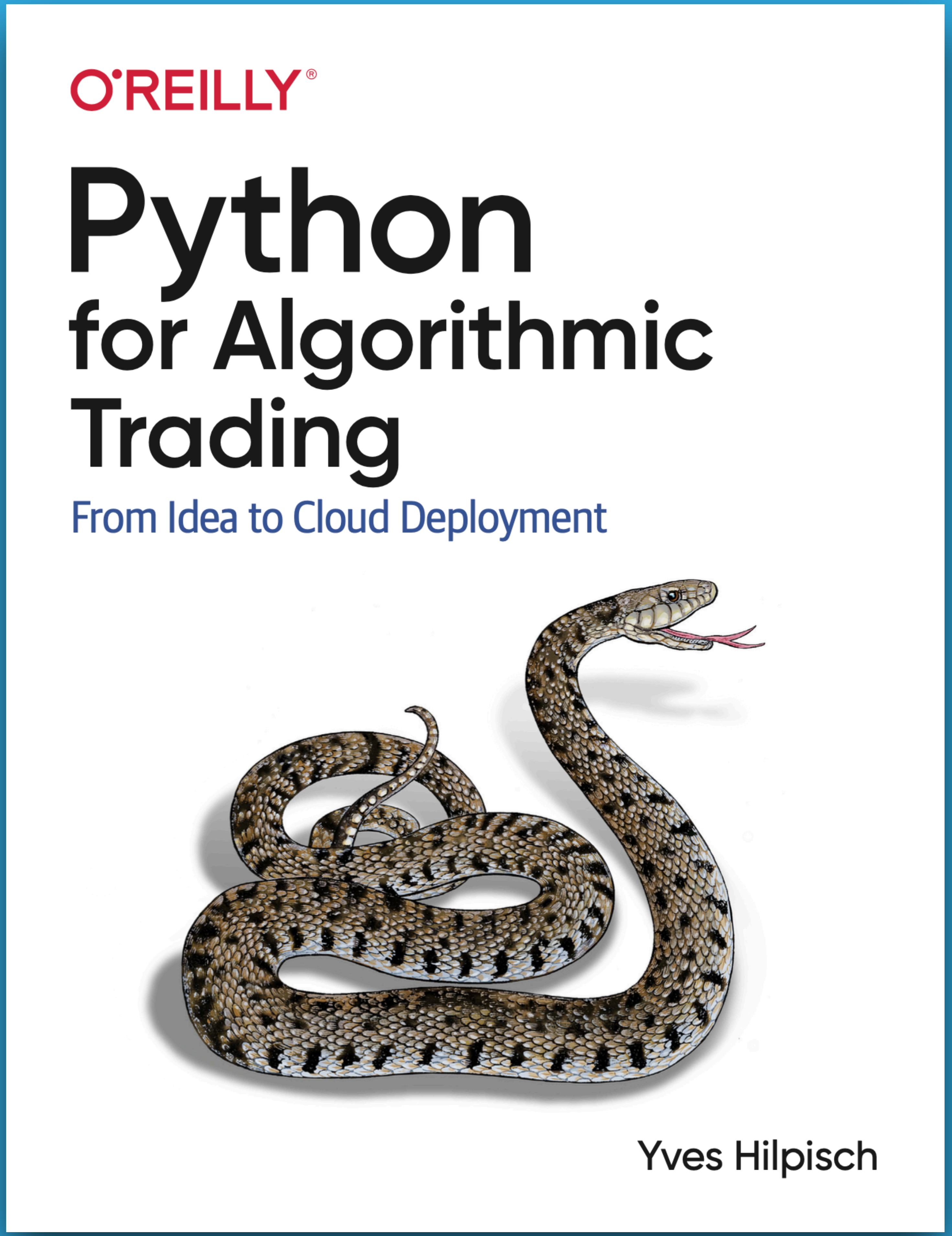
Python for Finance

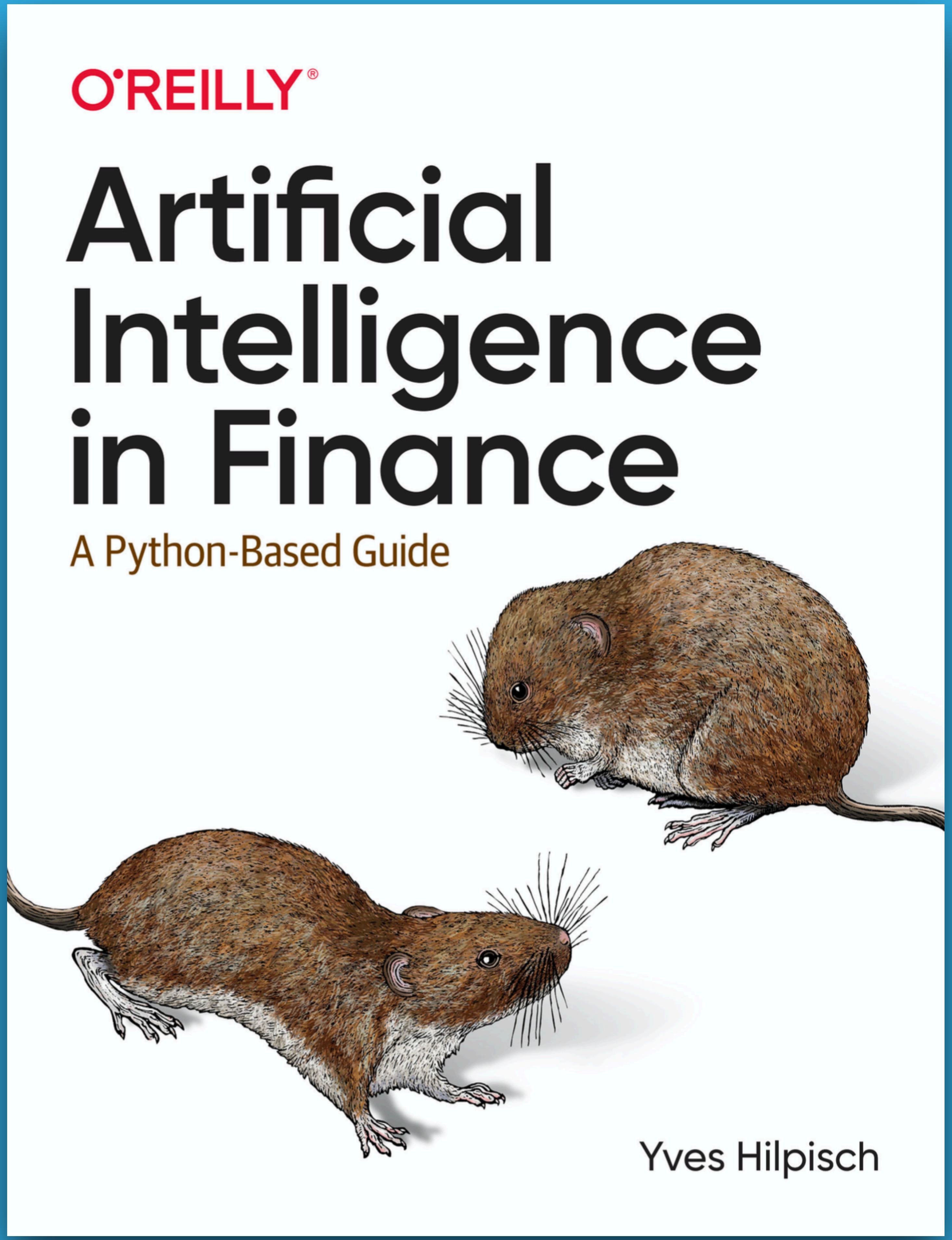




http://books.tpq.io

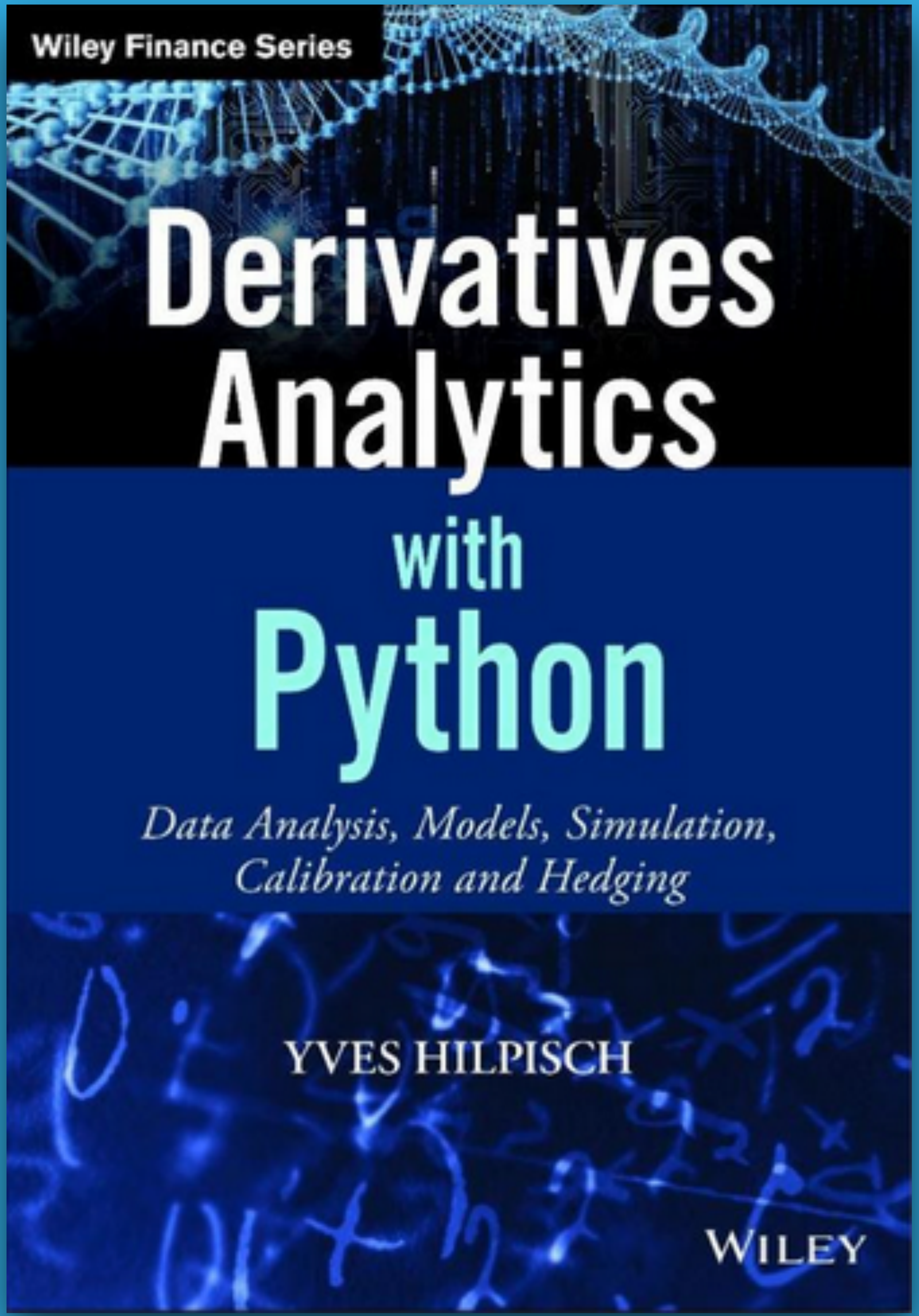
Python & AI for Finance & Trading

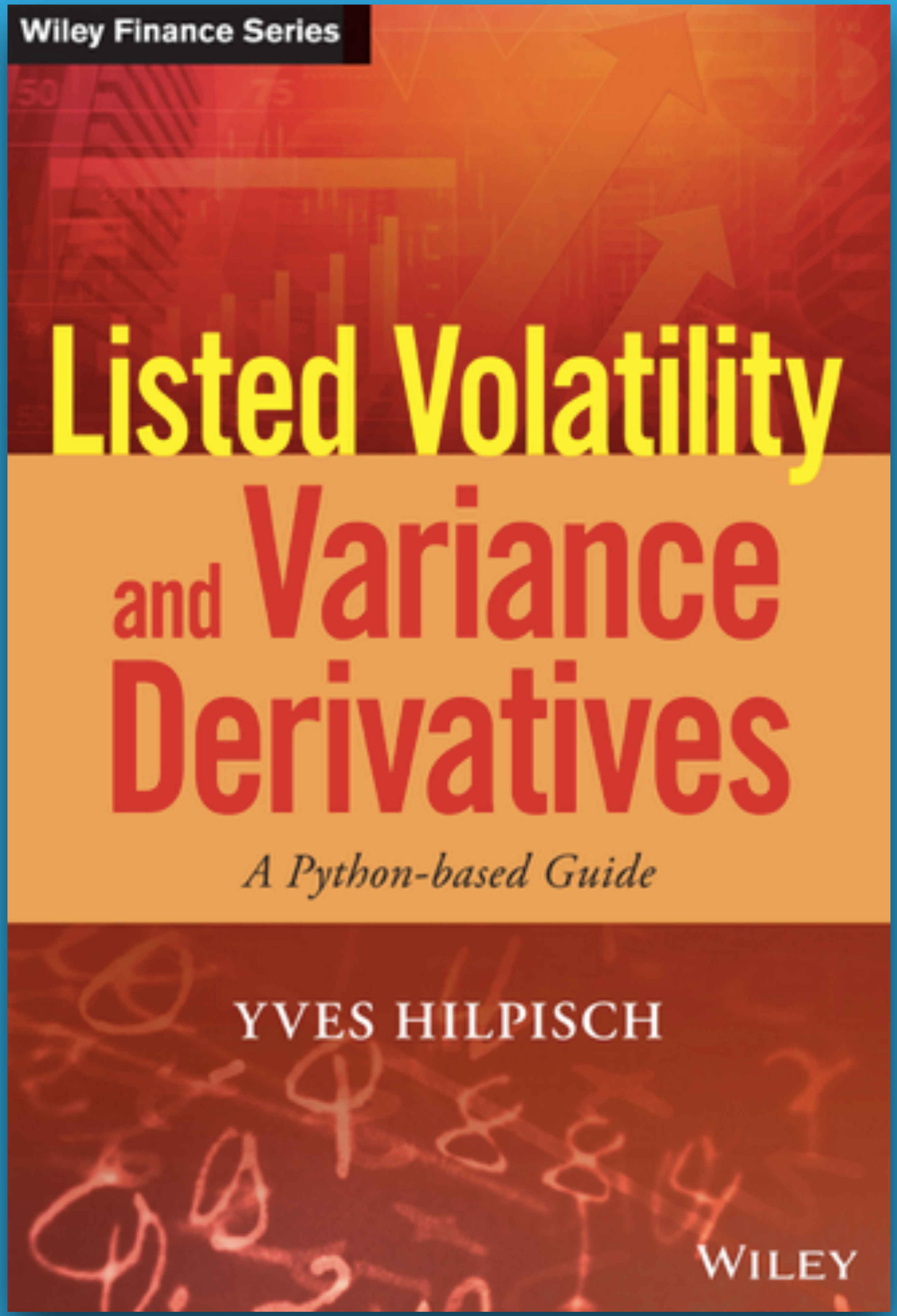




http://books.tpq.io

Quant Finance with Python





http://books.tpq.io

The Program

16 week program
(or self-paced)

250+ hours
of instruction

2,500+ pages
HTML/PDF

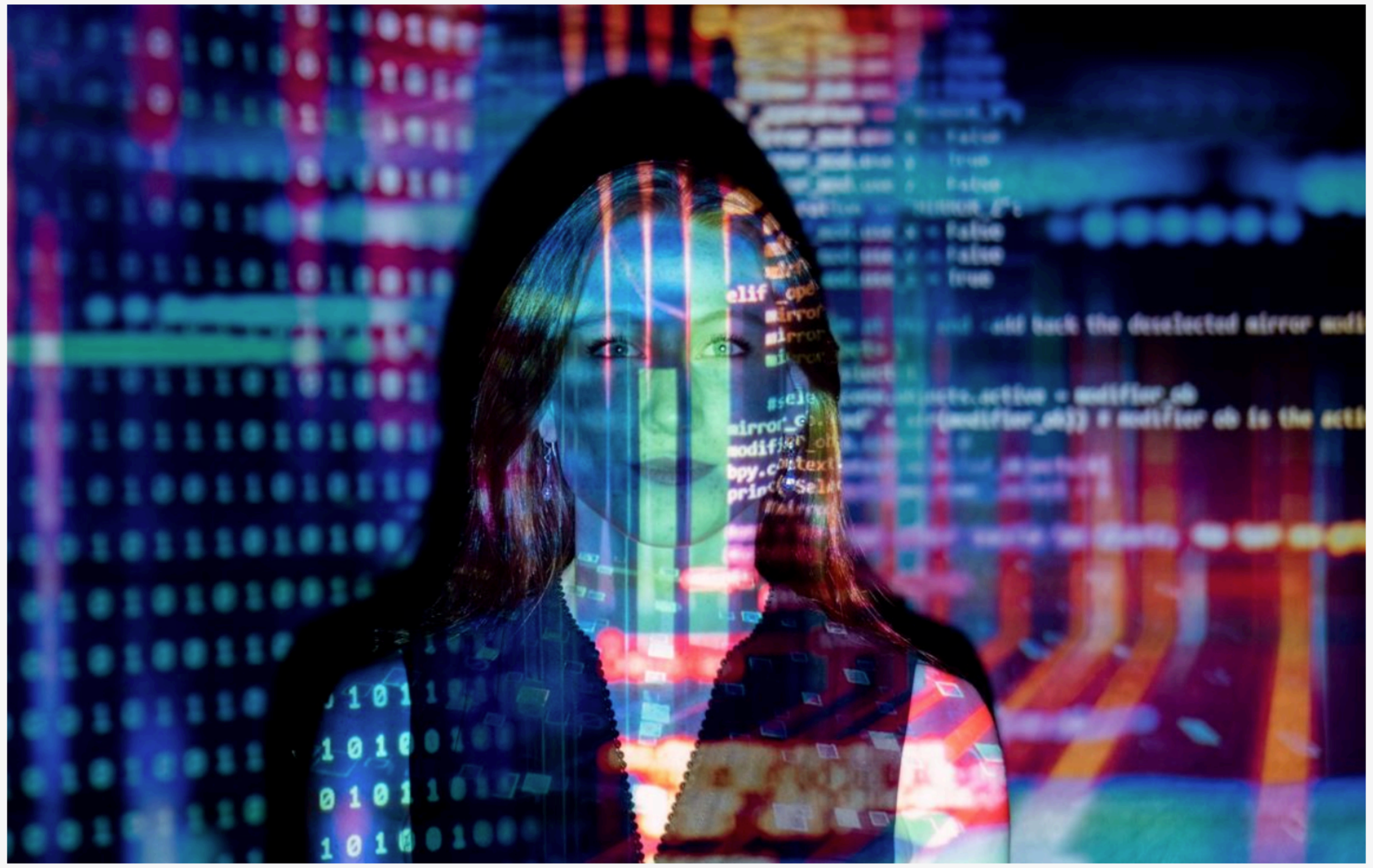
25,000+ lines
of code

250+ Jupyter
Notebooks

https://platinum.tpq.io

"If you want a banking job now, you need to code in Python"

by Mia Holmes 01 September 2021

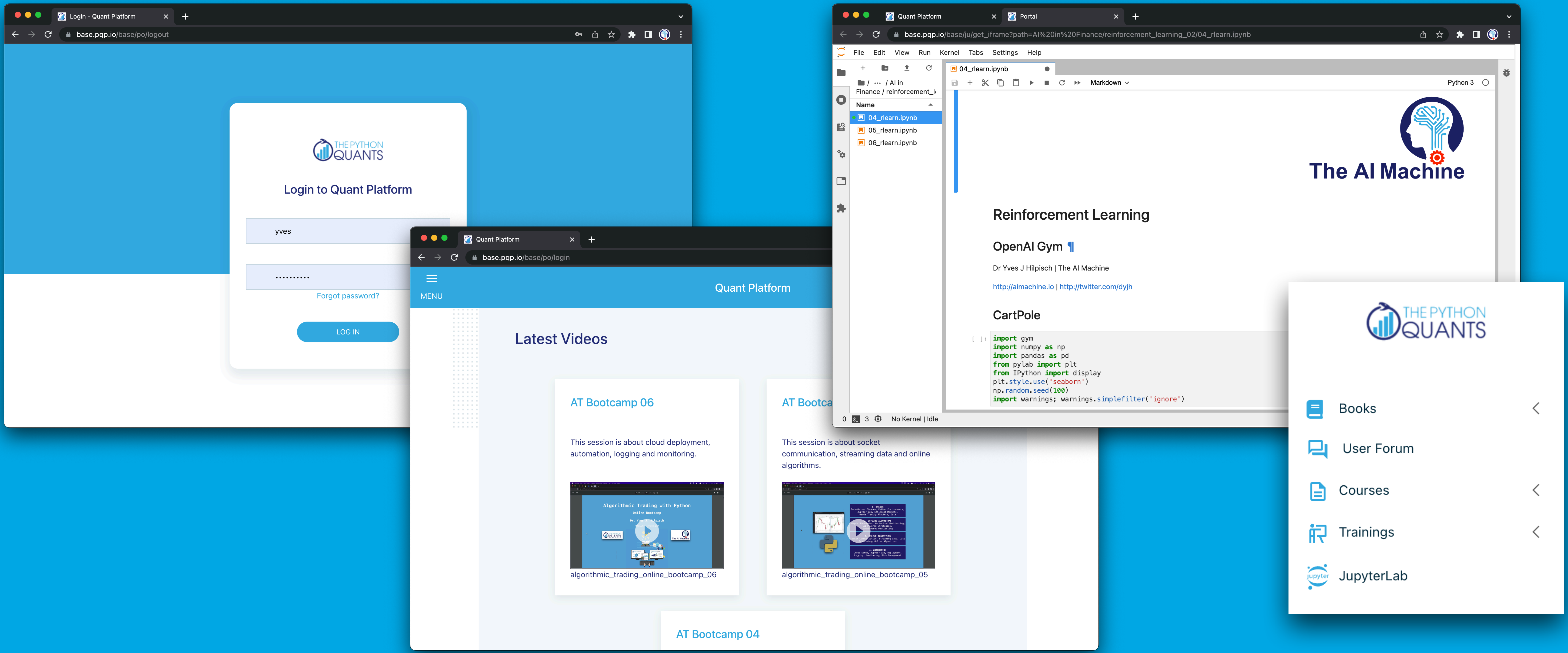


Therefore, if you're trying to get into banking now, Python is the skill that's needed. Python will get you a job across the banking industry - in anything from sales and trading to portfolio management or risk. It's the one skill that really differentiates applicants in the recruitment process.

Quant Platform 2.0

Quant Platform

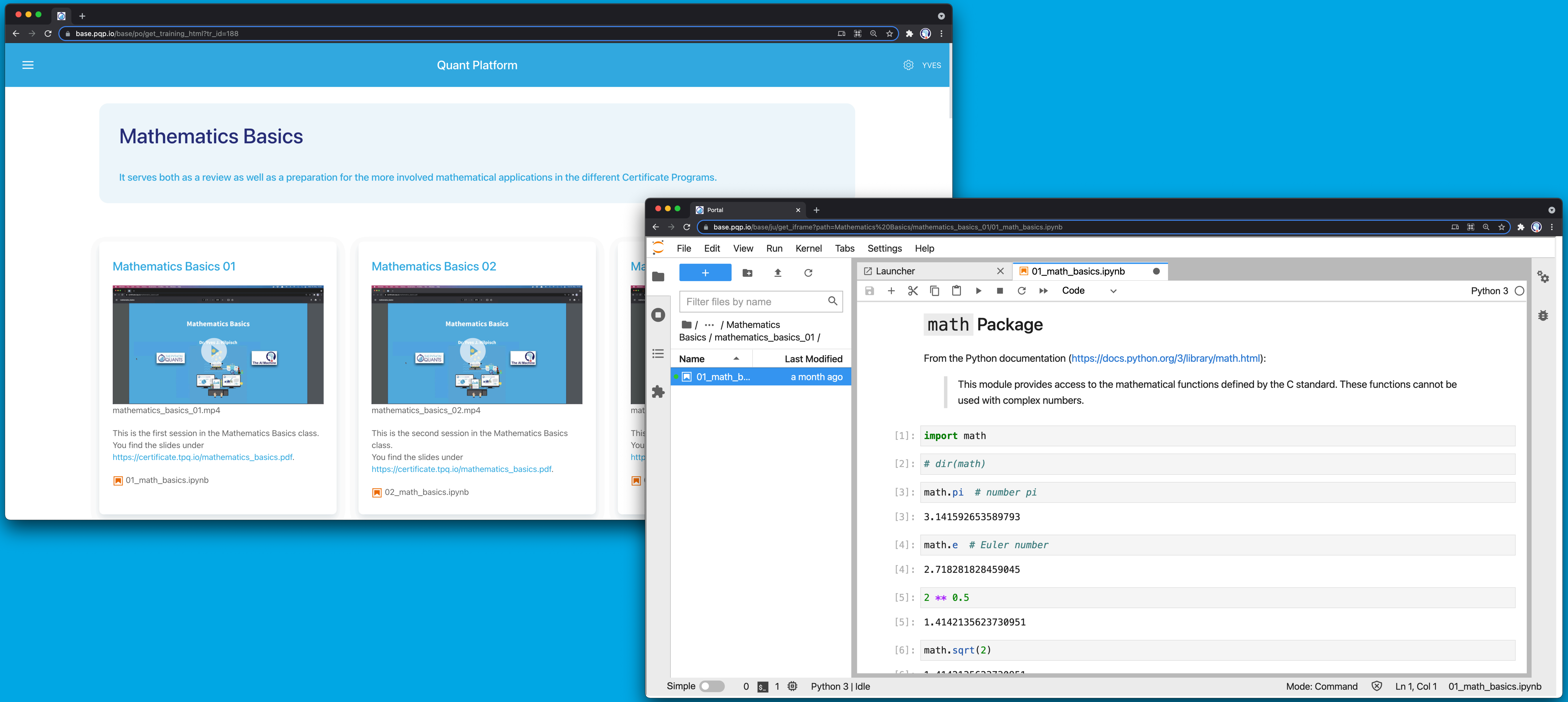
Your central content and code execution platform.



Mathematics Basics

Mathematics Basics

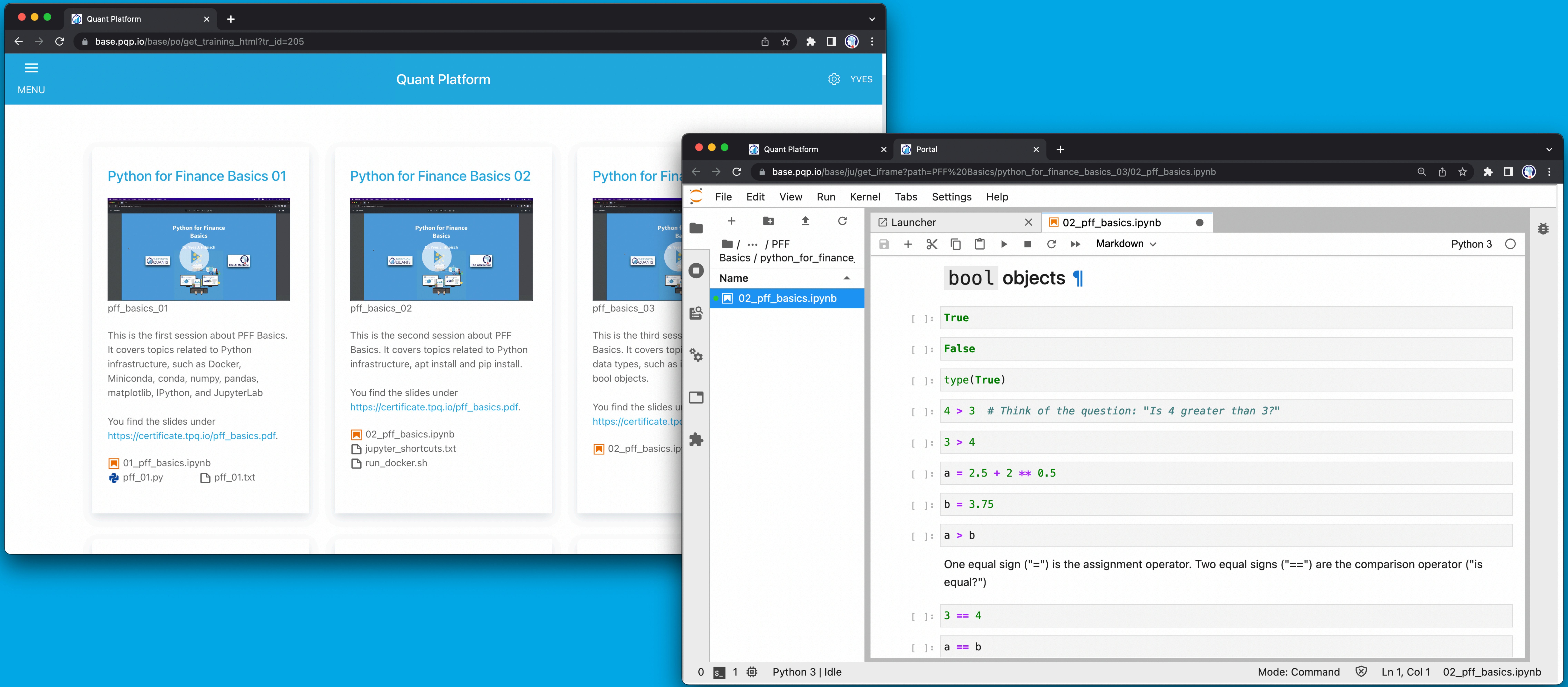
Reviewing fundamental mathematics concepts based on simple Python code.



Python for Fiance Basics

Python for Finance Basics

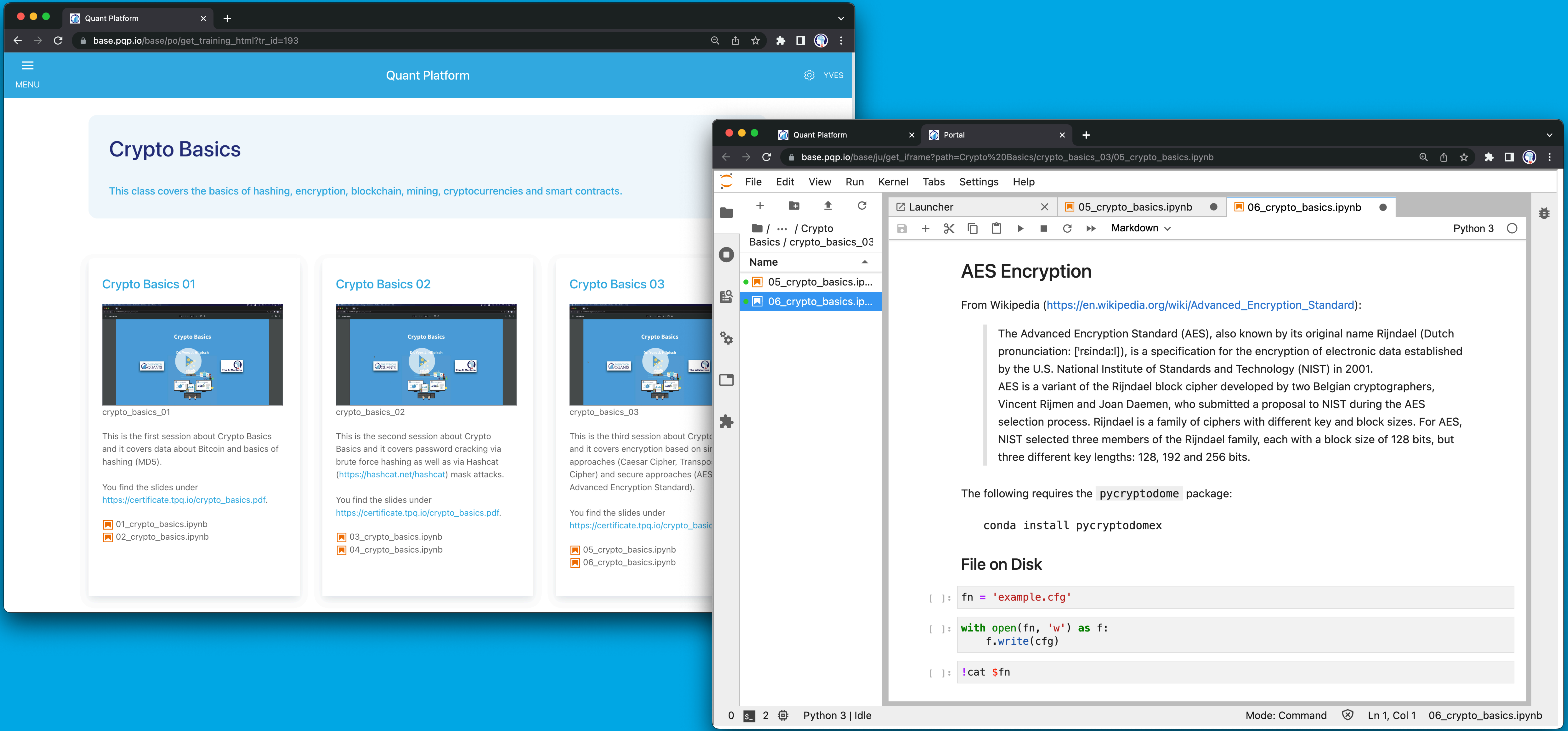
Introducing fundamental Python programming idioms and concepts.



Crypto Basics

Crypto Basics

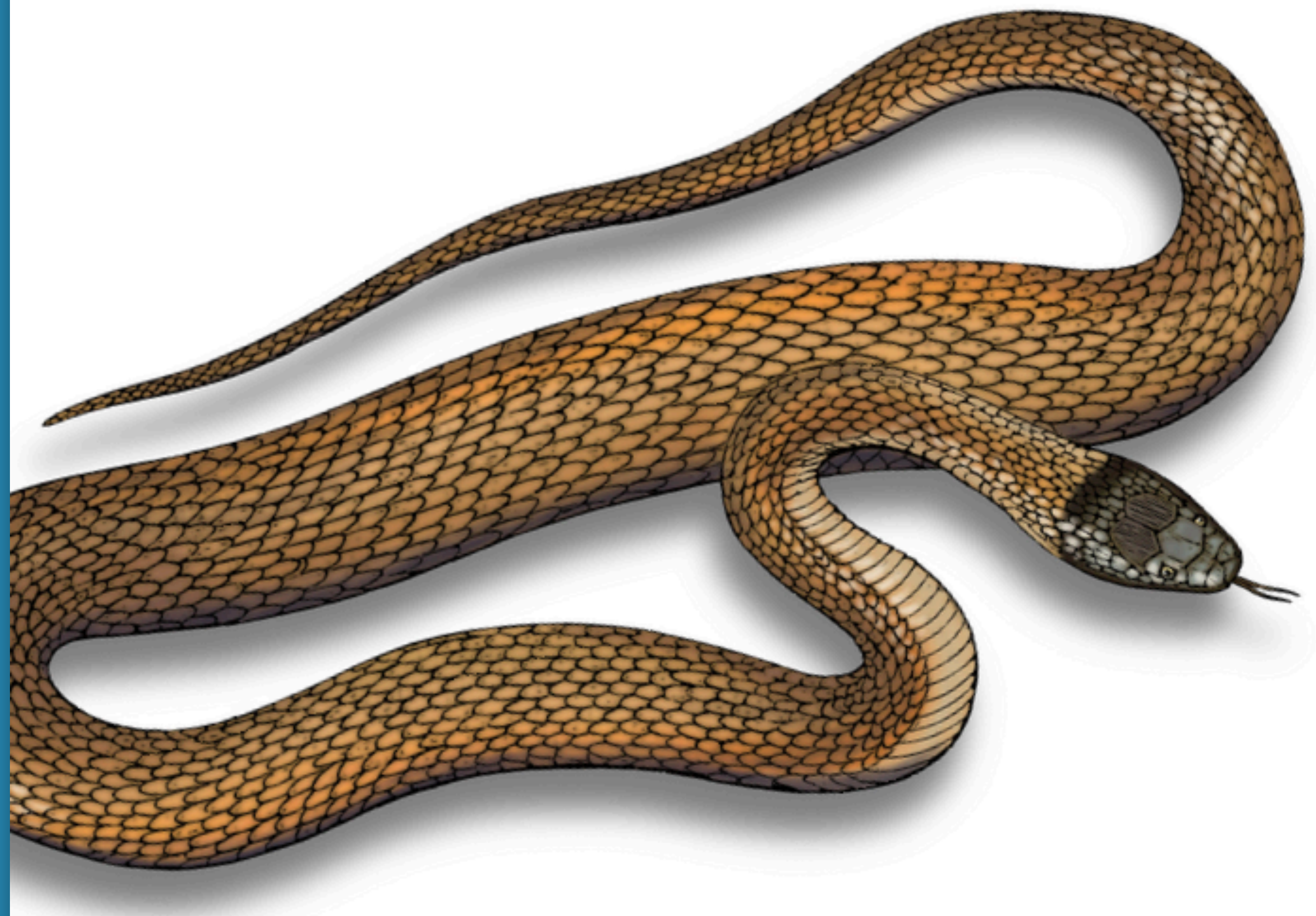
Mastering the basic building blocks of the crypto space.



Finance with Python

Financial Theory with Python

A Gentle Introduction

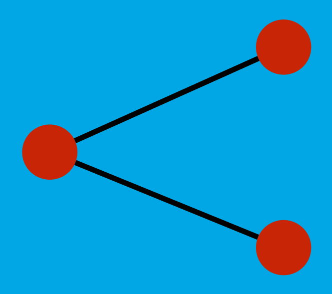


Yves Hilpisch

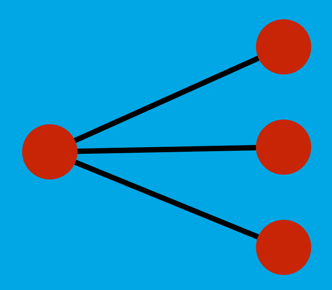
Finance with Python

A gentle introduction to Finance, Python and the combination of both.

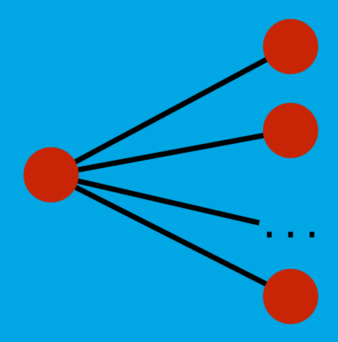
static two state economy
fundamental theorem of asset pricing



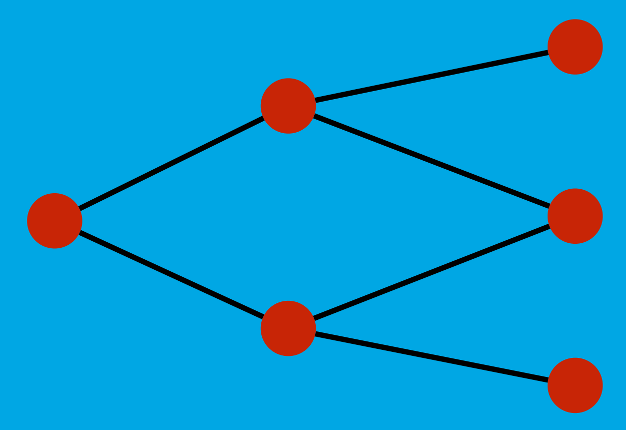
static three state economy
dealing with market incompleteness



static multi state economy
generalizing the state space

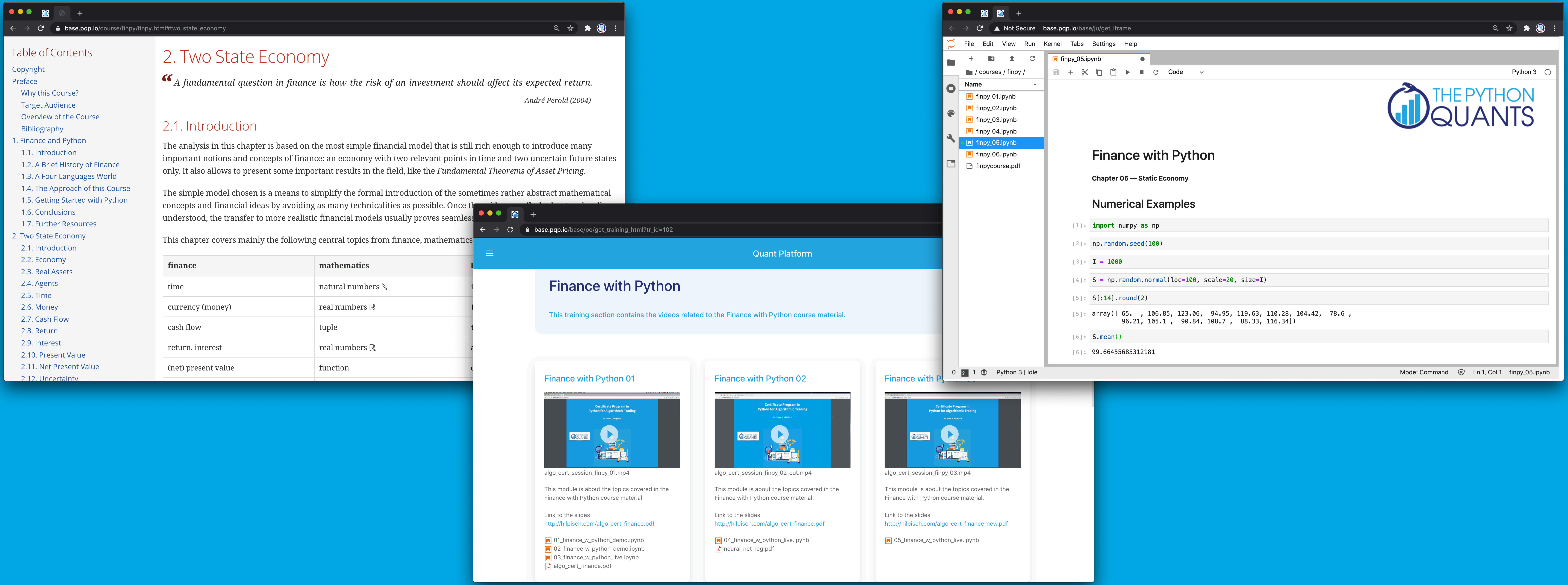


dynamic economies
modeling uncertainty over time



Finance with Python

A gentle introduction to Finance, Python and the combination of both.



Tools and Skills

Tools and Skills

Learning to use modern and proven tools for Python development and deployment.

BASICS	DEVELOPMENT	DISTRIBUTION
Python Installation & Environments on Mac & Linux on Windows	**Code Editing with Vim**	**Python Packaging**
Docker Usage **Jupyter Notebook** on Mac & Linux on Windows	**Screen + Vim + q (editing, logging, debugging)**	**Documentation**
Cloud Usage on Mac & Linux on Windows	**Doctest & Unittest**	**Code Hosting (eg Github)**
Basic Linux Tools & Shell Basics	**Git Version Control**	**Case Study**

Tools and Skills

Learning to use modern and proven tools for Python development and deployment.

cloud instances (“virtual servers”) from a low as 5 USD per month, to really big instances

https://m.do.co/c/fbe512dd3dac

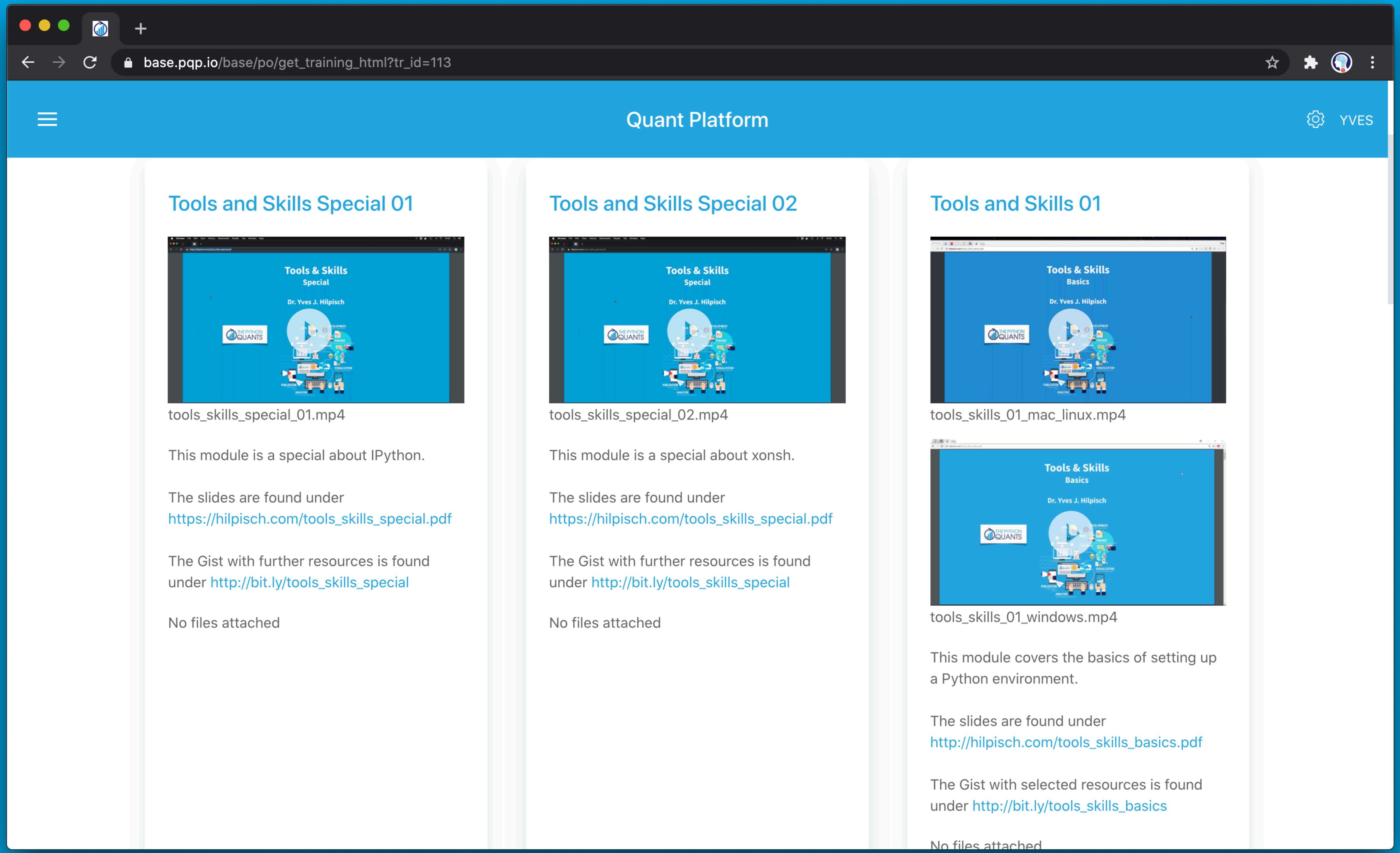
Docker containers to have separated OS run-times, packages and files
— on Mac OS, Linux & Windows

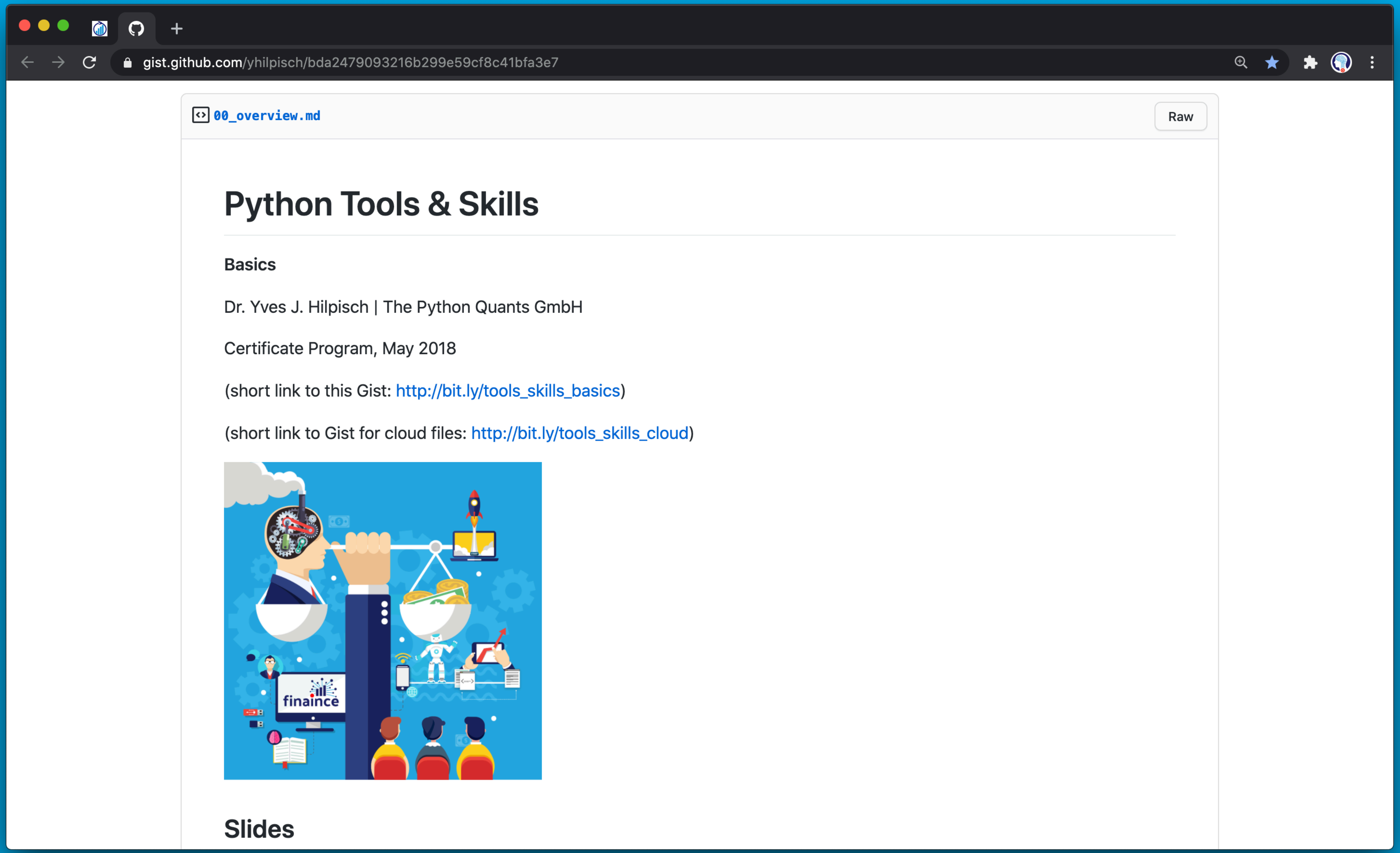
Python interpreter, packages, tools, environments
— via conda

Jupyter Notebooks, terminal, editor, IPython & Vim
— on every infrastructure

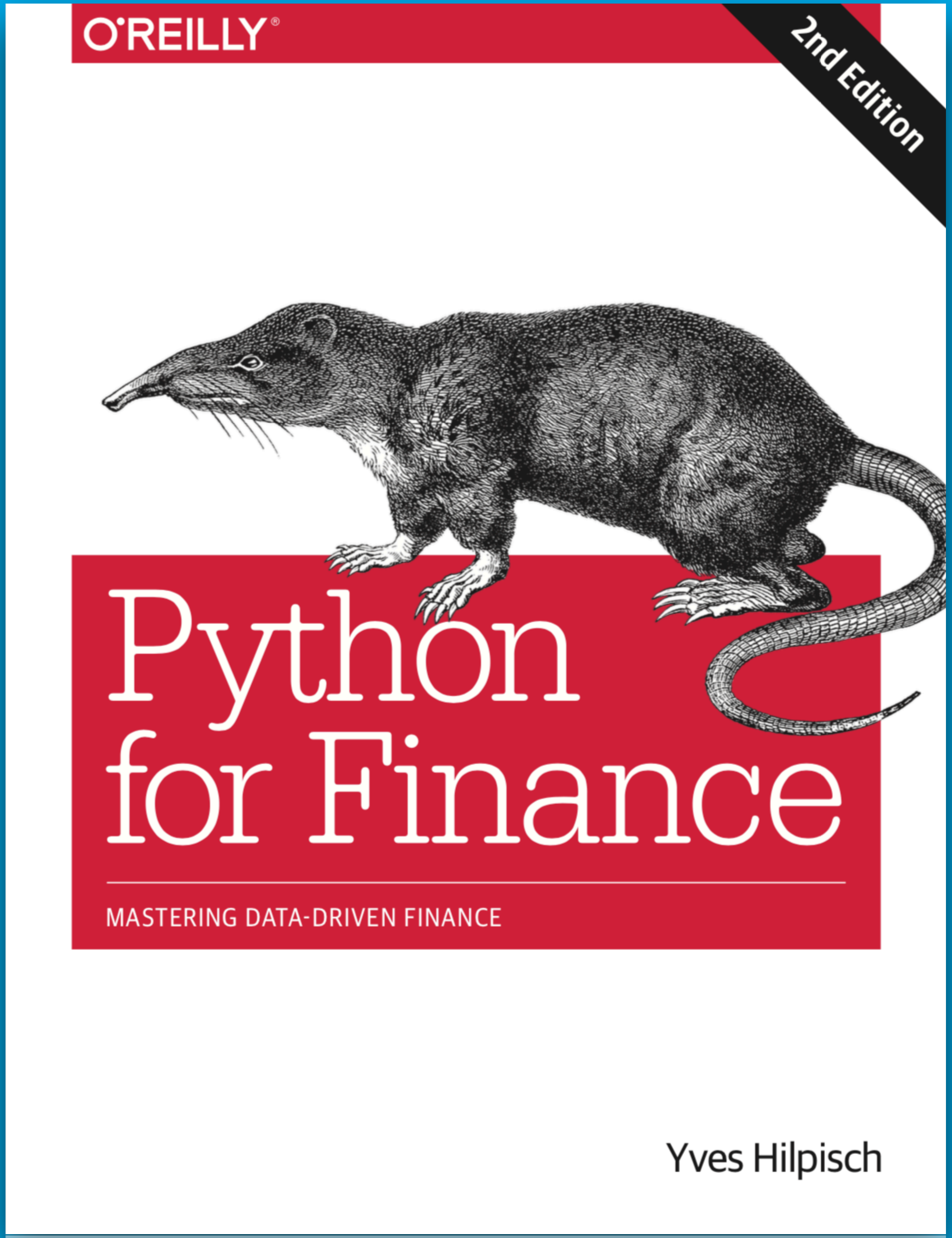
Tools and Skills

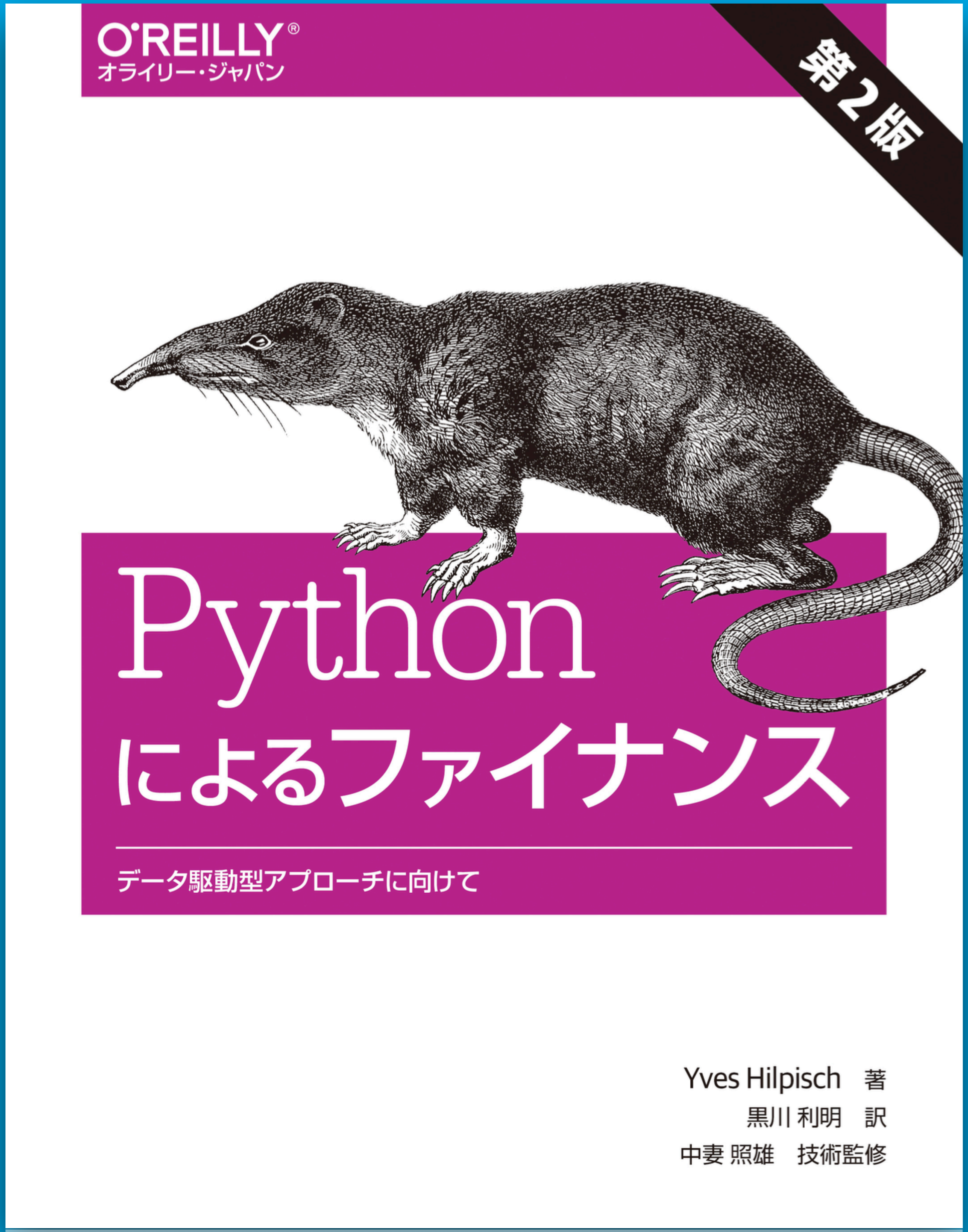
Learning to use modern and proven tools for Python development and deployment.





Python for Financial Data Science


O'REILLY
2nd Edition
Python
for Finance
MASTERING DATA-DRIVEN FINANCE
Yves Hilpisch


O'REILLY
オライリー・ジャパン
第2版
Python
によるファイナンス
データ駆動型アプローチに向けて
Yves Hilpisch 著
黒川 利明 訳
中妻 照雄 技術監修

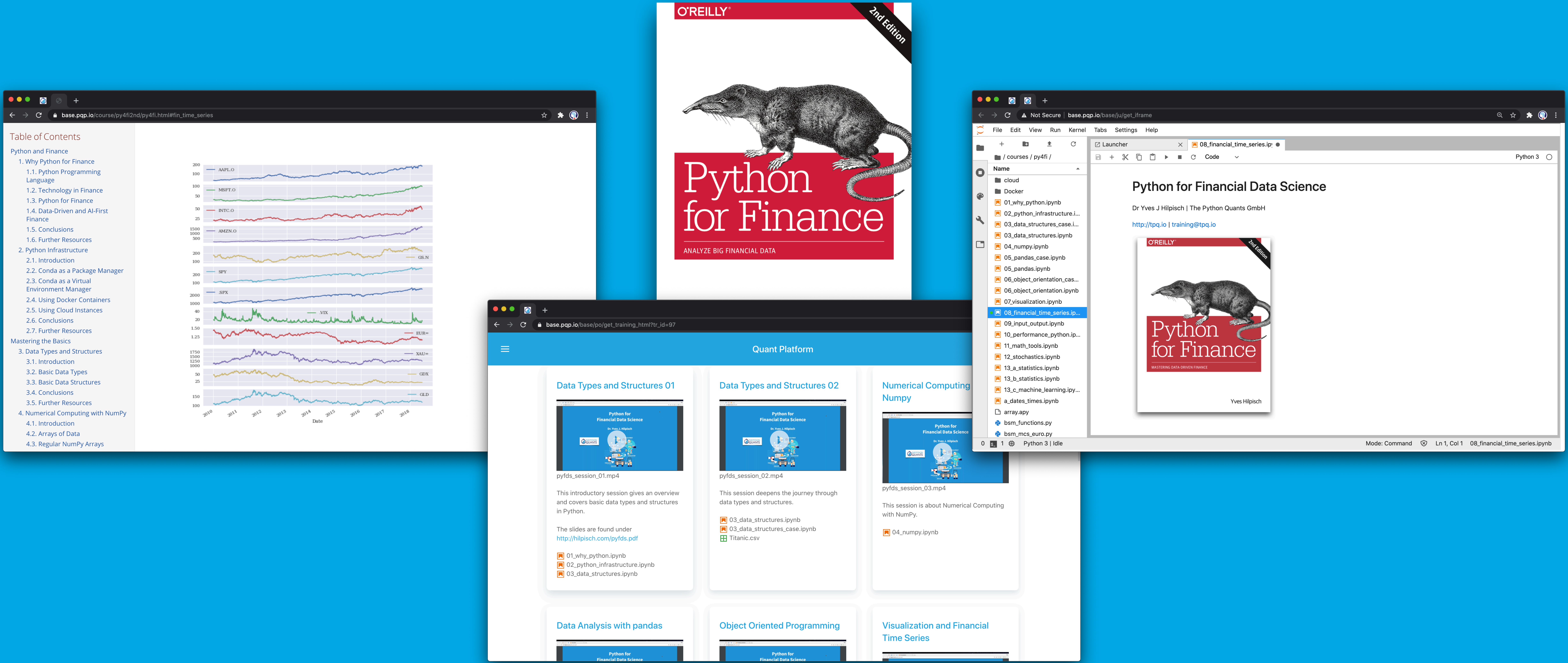
Python for Financial Data Science

Covering the most important Python idioms, techniques and packages for finance.

BASICS	DATA SCIENCE	MATHEMATICS
Python Data Types & Structures	Visualization	Mathematical Tools
Numerical Computing with NumPy	Financial Time Series	Stochastics
Data Analysis with pandas	Input-Output Operations	Statistics & Machine Learning
Object Oriented Programming	Performance Python	Special: Dates & Times

Python for Financial Data Science

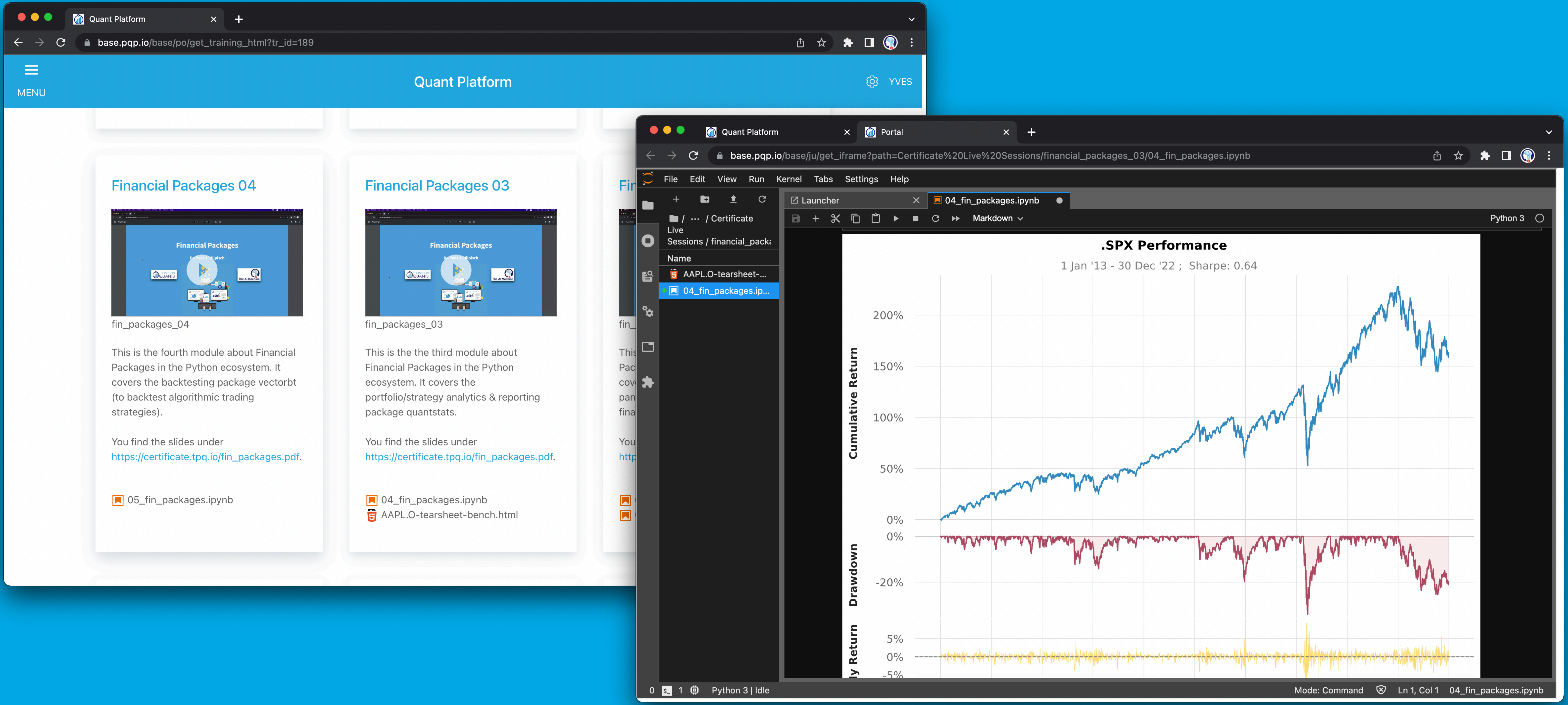
Covering the most important Python idioms, techniques and packages for finance.



Financial Packages

Financial Packages

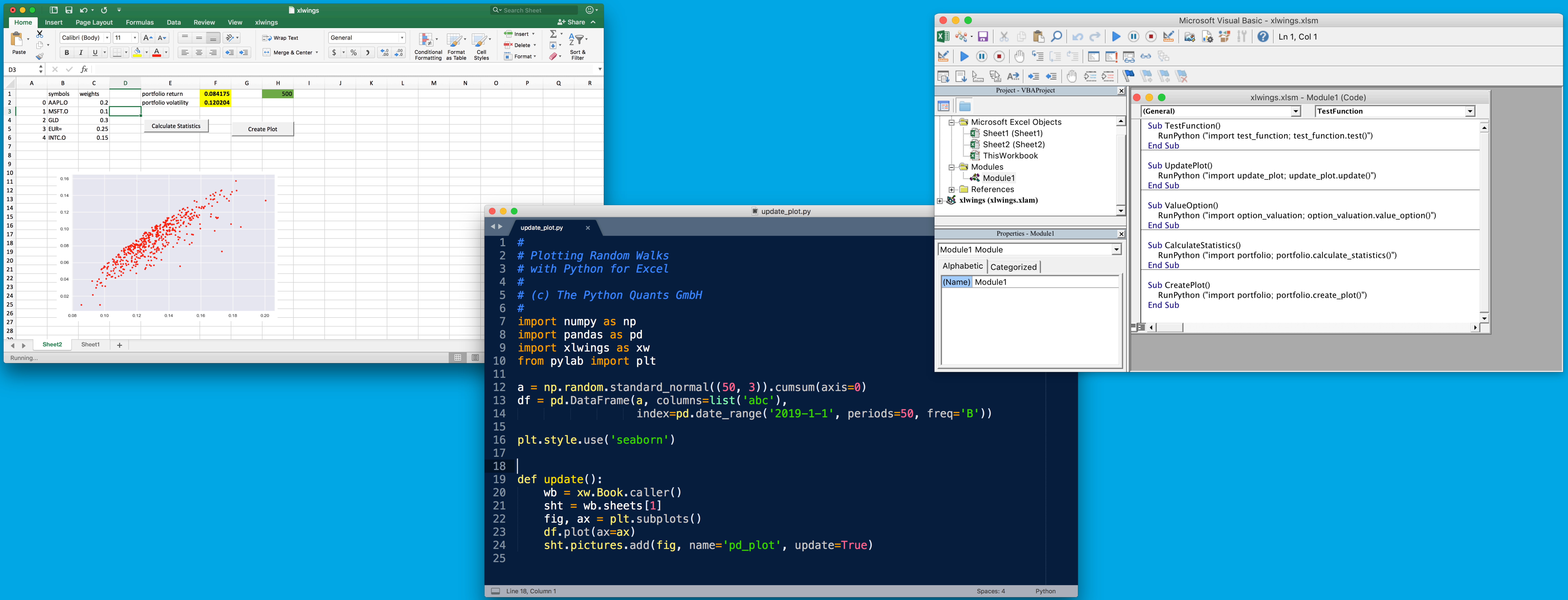
Efficient and powerful Python packages for finance.



Python for Excel

Python for Excel

Combining Excel with the analytics power of Python.



Python for Databases

Python for Databases

Making use of database technologies to efficiently manage (large) financial data sets.

Introduction & Motivation

SQL & NoSQL

- **MySQL**
- **SQLite**
- **MongoDB**

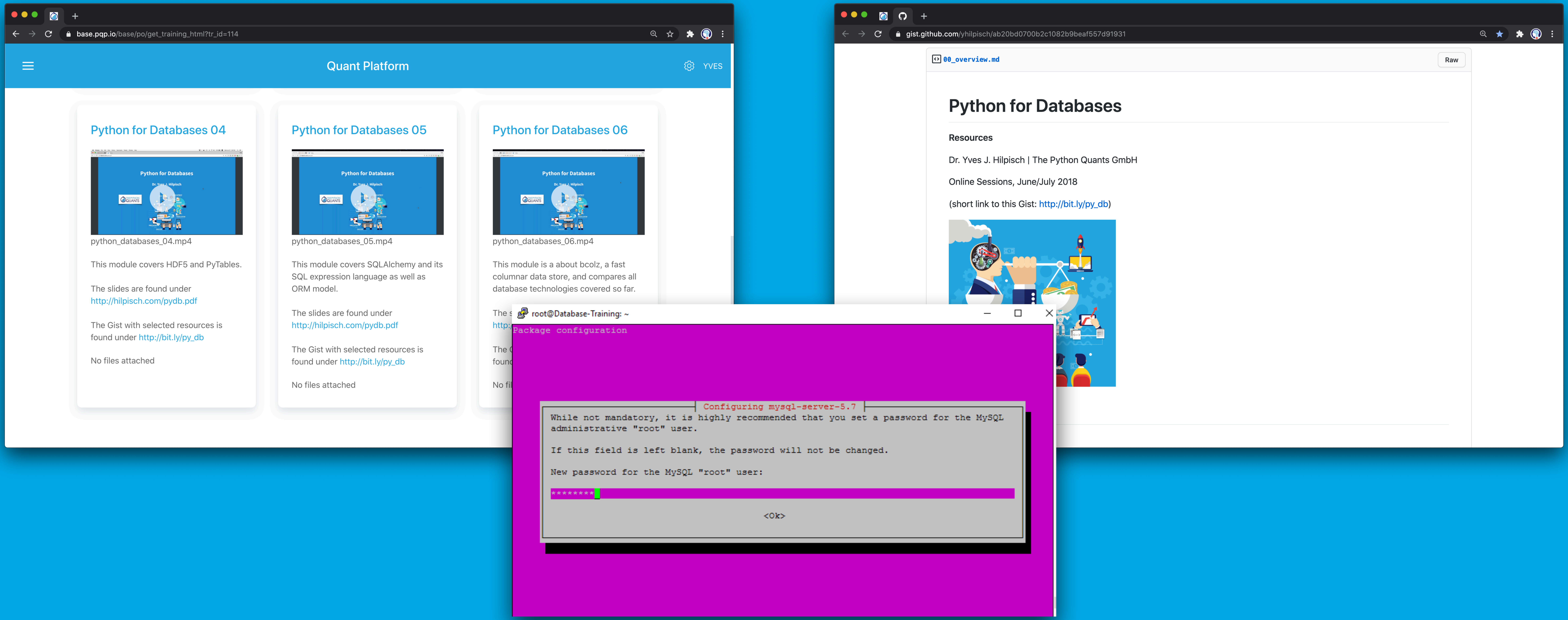
PYTHON

- **PyTables**
- **SQLAlchemy**
- **bcolz**

Comparison

Python for Databases

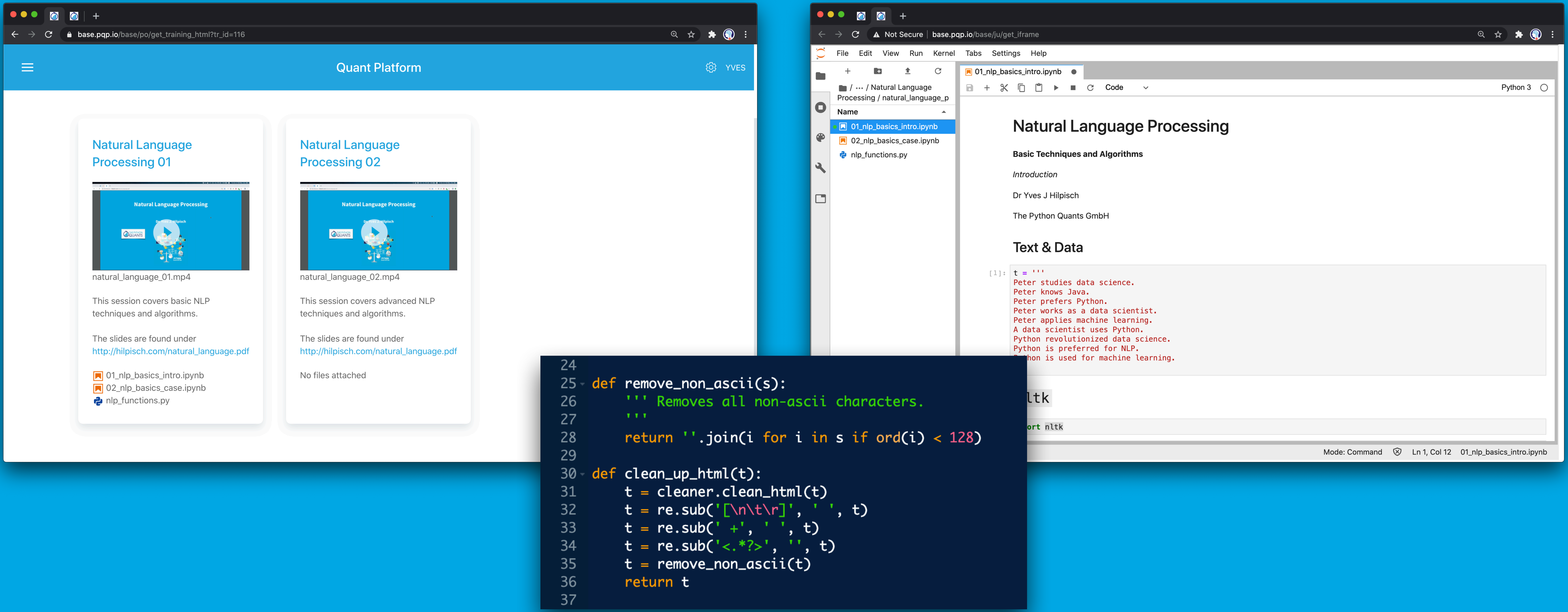
Making use of database technologies to efficiently manage (large) financial data sets.



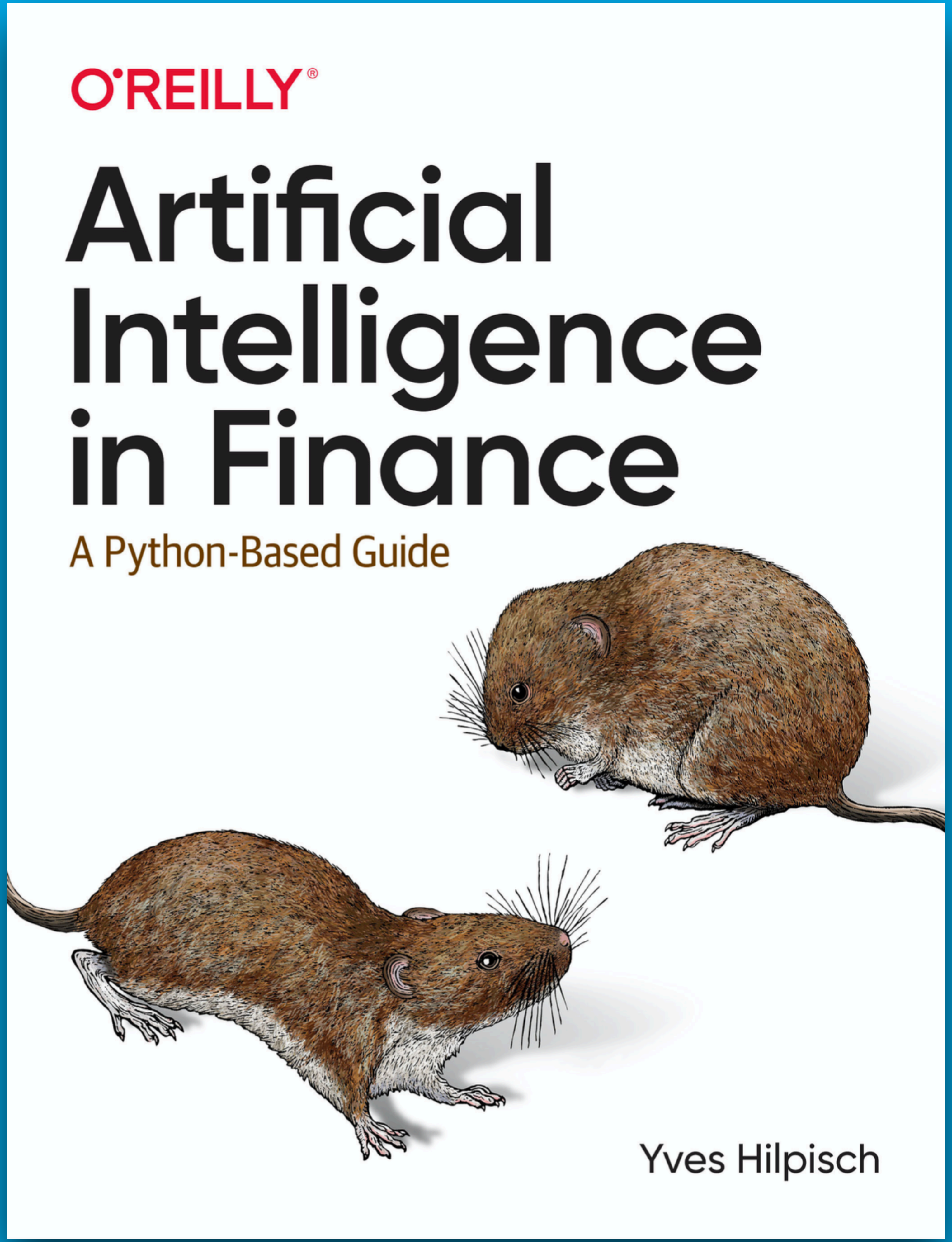
Natural Language Processing

Natural Language Processing

Being able to process unstructured data sources such as news, texts, etc. at scale with Python.



Artificial Intelligence in Finance



CHAPTER 2

Superintelligence

> This is the first time that a computer program has defeated a human professional player in the full-sized game of Go, a feat previously thought to be at least a decade away.
>
> —Silver et al. (2016)

There are multiple definitions for the term *technological singularity*. Its use dates back at least to the article by Vinge (1993), which the author provocatively begins like this:

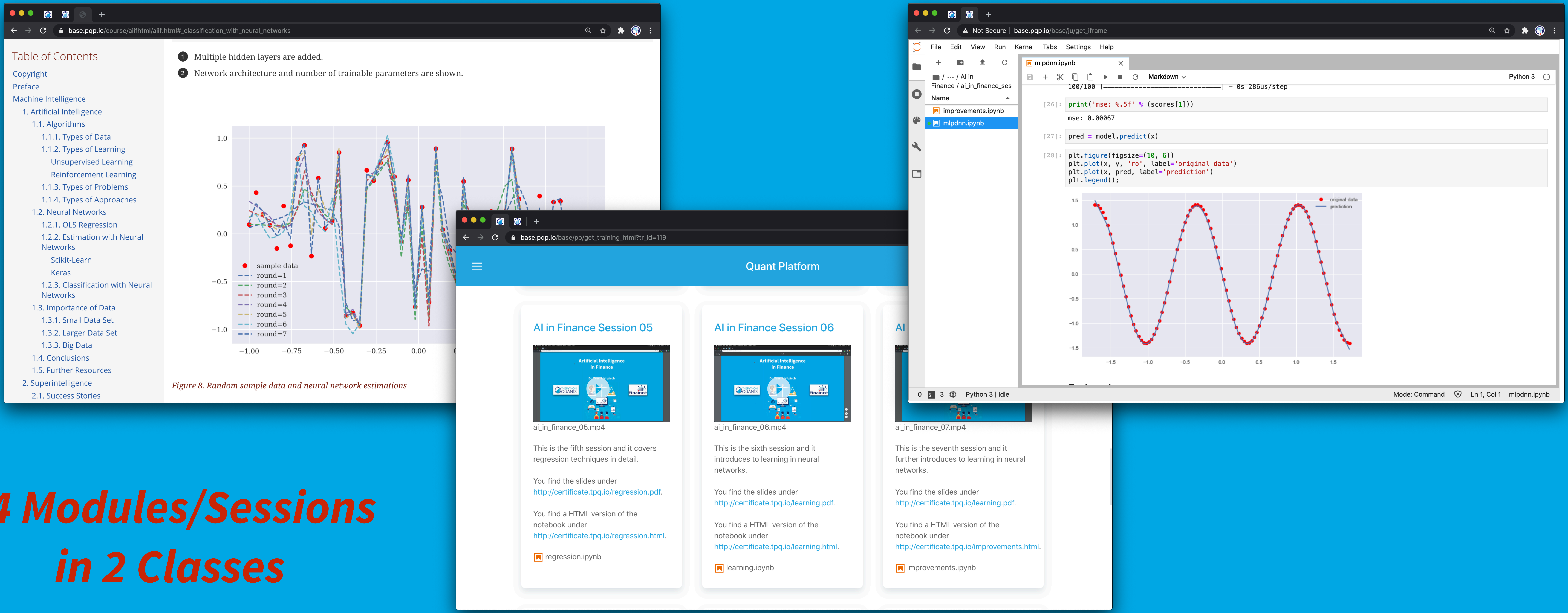
> Within thirty years, we will have the technological means to create superhuman intelligence. Shortly after, the human era will be ended.
>
> —Vinge (1993)

For the purposes of this chapter and book, *technological singularity* refers to a point in time at which machines can achieve superhuman intelligence or superintelligence — this is mostly in line with the original idea of Vinge (1993). The idea and concept was further popularized by the widely read and cited book Kurzweil (2005). Barrat (2013) has a wealth of historical and anecdotal information around the topic. Shanahan (2015) provides an informal introduction and overview of its central aspects. The expression *technological singularity* itself has its origin in a the concept of a *singularity* in physics. It refers to the center of a black hole where mass is highly concentrated, gravitation becomes infinite and traditional laws of physics break down. The beginning of the universe, the so-called Big Bang, is also referred to as a singularity.

Although the general ideas and concepts of the technological singularity and of superintelligences might not have an obvious and direct relationship to AI applied to finance, a better understanding of their background, related problems, and potential consequences are beneficial. The insights gained are important in a narrower context as well, such as AI in finance, and that they help in guiding the discussion about how AI might reshape finance in the near and long term.

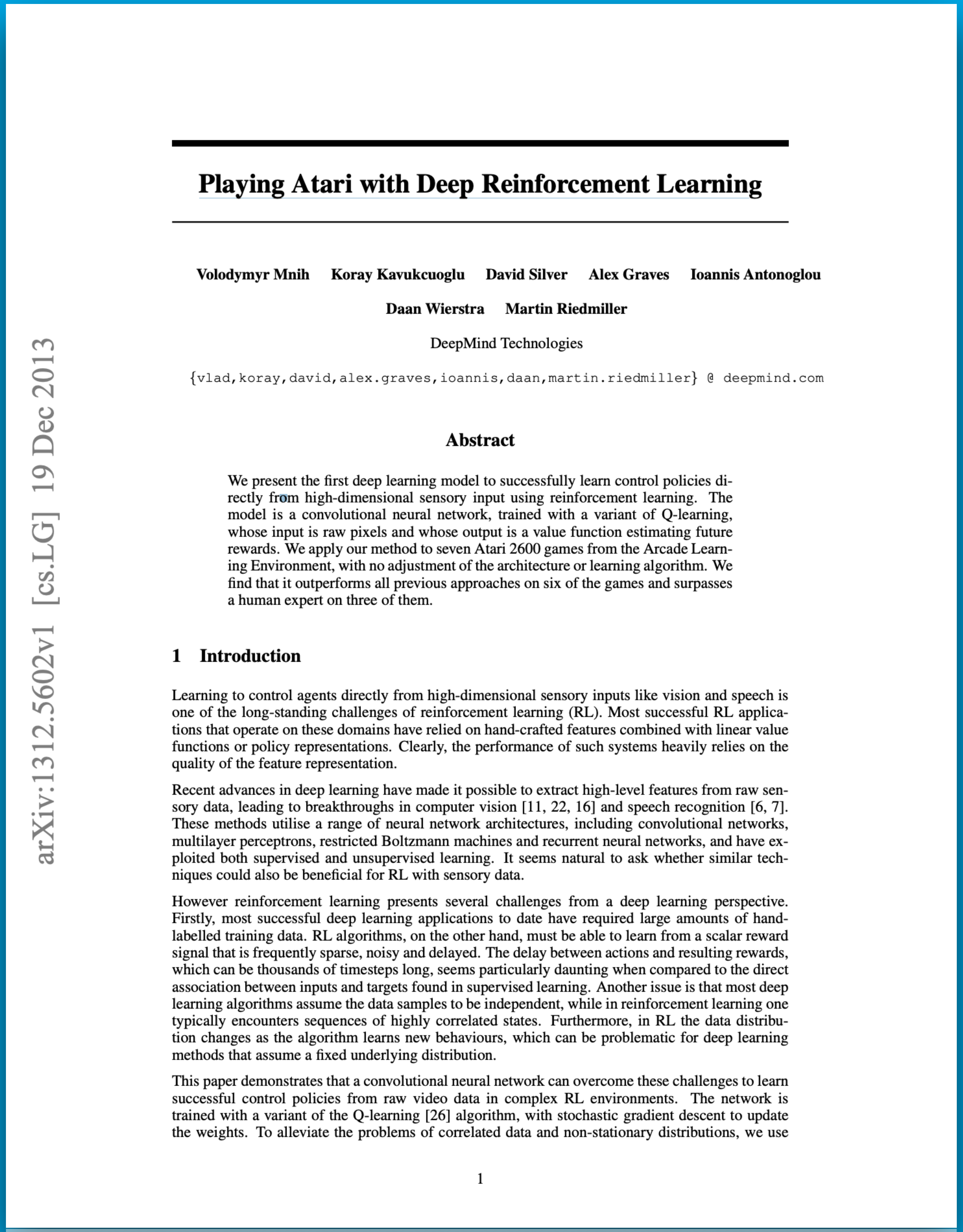
Artificial Intelligence in Finance

Replacing normative finance by data-driven, AI-first finance



24 Modules/Sessions in 2 Classes

Reinforcement Learning for Finance



arXiv:1312.5602v1 [cs.LG] 19 Dec 2013

Playing Atari with Deep Reinforcement Learning

Volodymyr Mnih **Koray Kavukcuoglu** **David Silver** **Alex Graves** **Ioannis Antonoglou**

Daan Wierstra **Martin Riedmiller**

DeepMind Technologies

{vlad,koray,david,alex.graves,ioannis,daan,martin.riedmiller} @ deepmind.com

Abstract

We present the first deep learning model to successfully learn control policies directly from high-dimensional sensory input using reinforcement learning. The model is a convolutional neural network, trained with a variant of Q-learning, whose input is raw pixels and whose output is a value function estimating future rewards. We apply our method to seven Atari 2600 games from the Arcade Learning Environment, with no adjustment of the architecture or learning algorithm. We find that it outperforms all previous approaches on six of the games and surpasses a human expert on three of them.

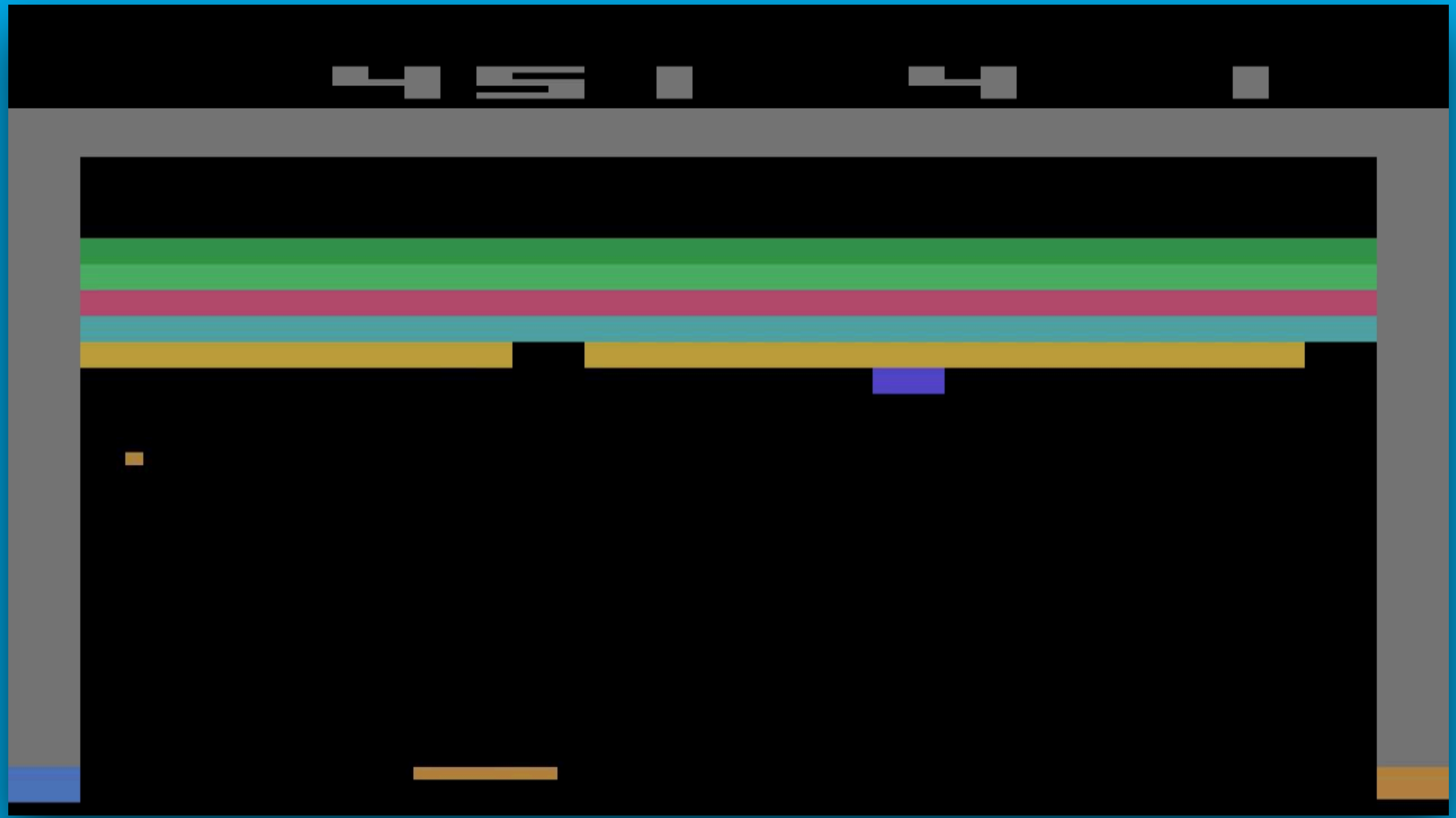
1 Introduction

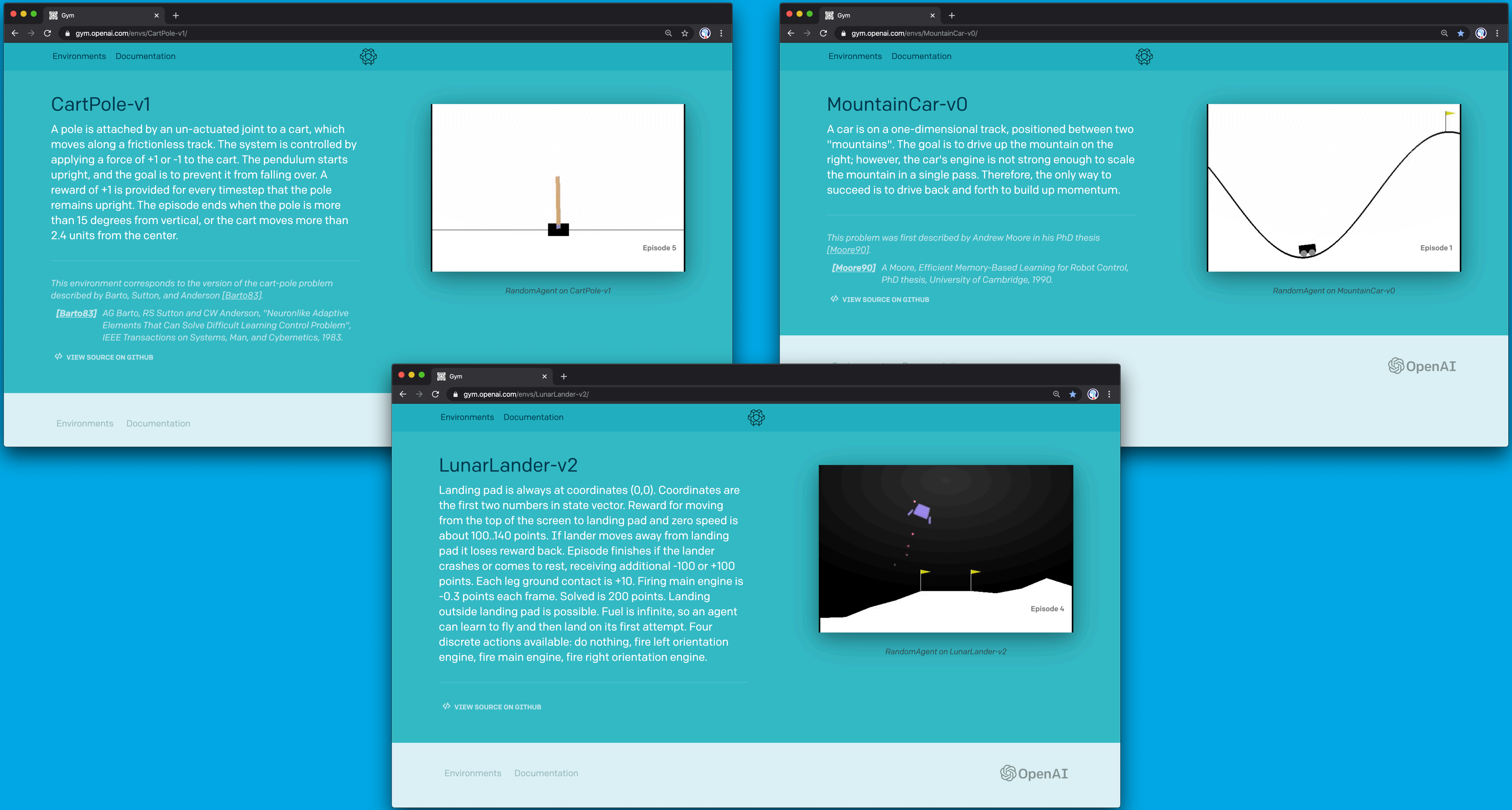
Learning to control agents directly from high-dimensional sensory inputs like vision and speech is one of the long-standing challenges of reinforcement learning (RL). Most successful RL applications that operate on these domains have relied on hand-crafted features combined with linear value functions or policy representations. Clearly, the performance of such systems heavily relies on the quality of the feature representation.

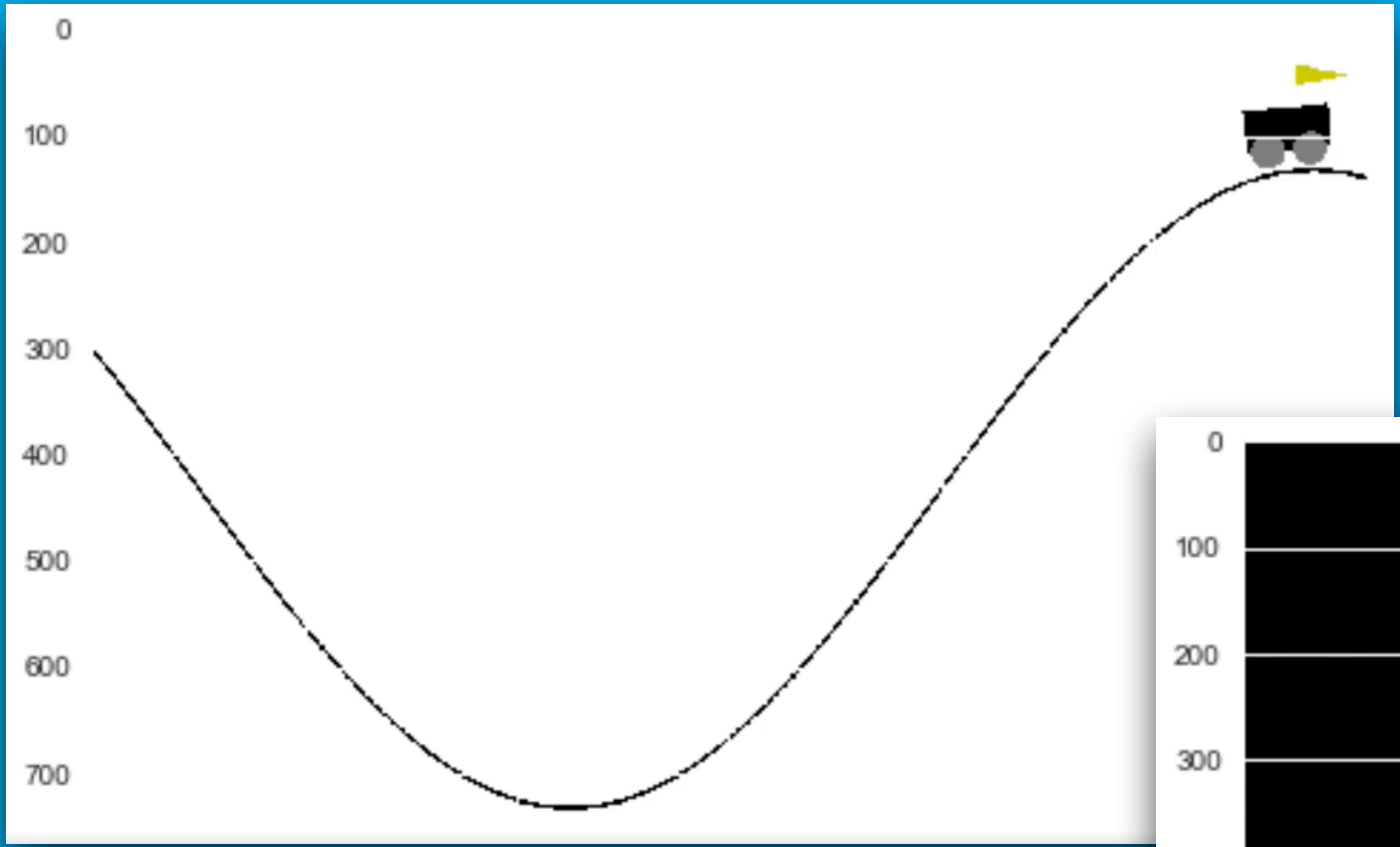
Recent advances in deep learning have made it possible to extract high-level features from raw sensory data, leading to breakthroughs in computer vision [11, 22, 16] and speech recognition [6, 7]. These methods utilise a range of neural network architectures, including convolutional networks, multilayer perceptrons, restricted Boltzmann machines and recurrent neural networks, and have exploited both supervised and unsupervised learning. It seems natural to ask whether similar techniques could also be beneficial for RL with sensory data.

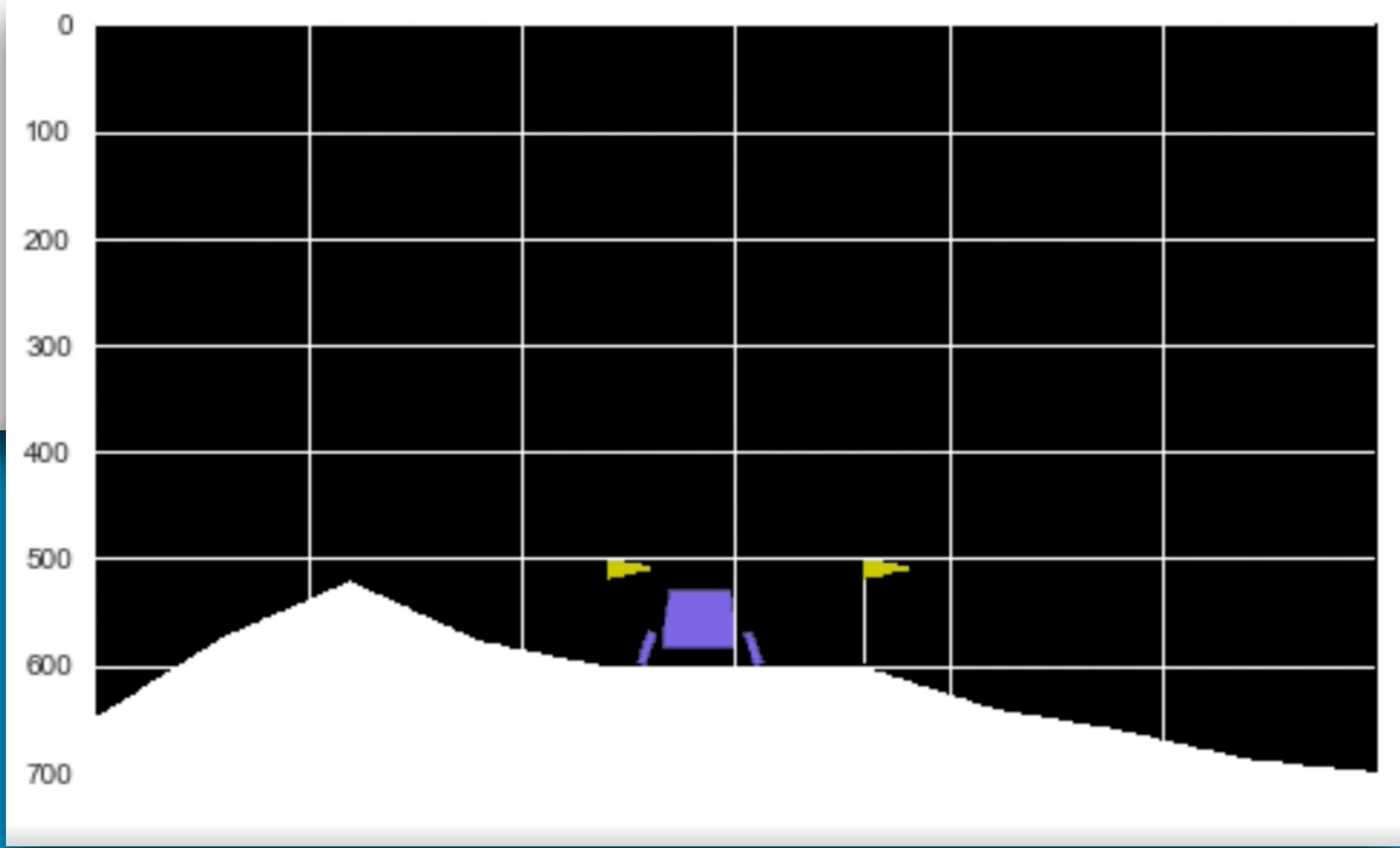
However reinforcement learning presents several challenges from a deep learning perspective. Firstly, most successful deep learning applications to date have required large amounts of hand-labelled training data. RL algorithms, on the other hand, must be able to learn from a scalar reward signal that is frequently sparse, noisy and delayed. The delay between actions and resulting rewards, which can be thousands of timesteps long, seems particularly daunting when compared to the direct association between inputs and targets found in supervised learning. Another issue is that most deep learning algorithms assume the data samples to be independent, while in reinforcement learning one typically encounters sequences of highly correlated states. Furthermore, in RL the data distribution changes as the algorithm learns new behaviours, which can be problematic for deep learning methods that assume a fixed underlying distribution.

This paper demonstrates that a convolutional neural network can overcome these challenges to learn successful control policies from raw video data in complex RL environments. The network is trained with a variant of the Q-learning [26] algorithm, with stochastic gradient descent to update the weights. To alleviate the problems of correlated data and non-stationary distributions, we use



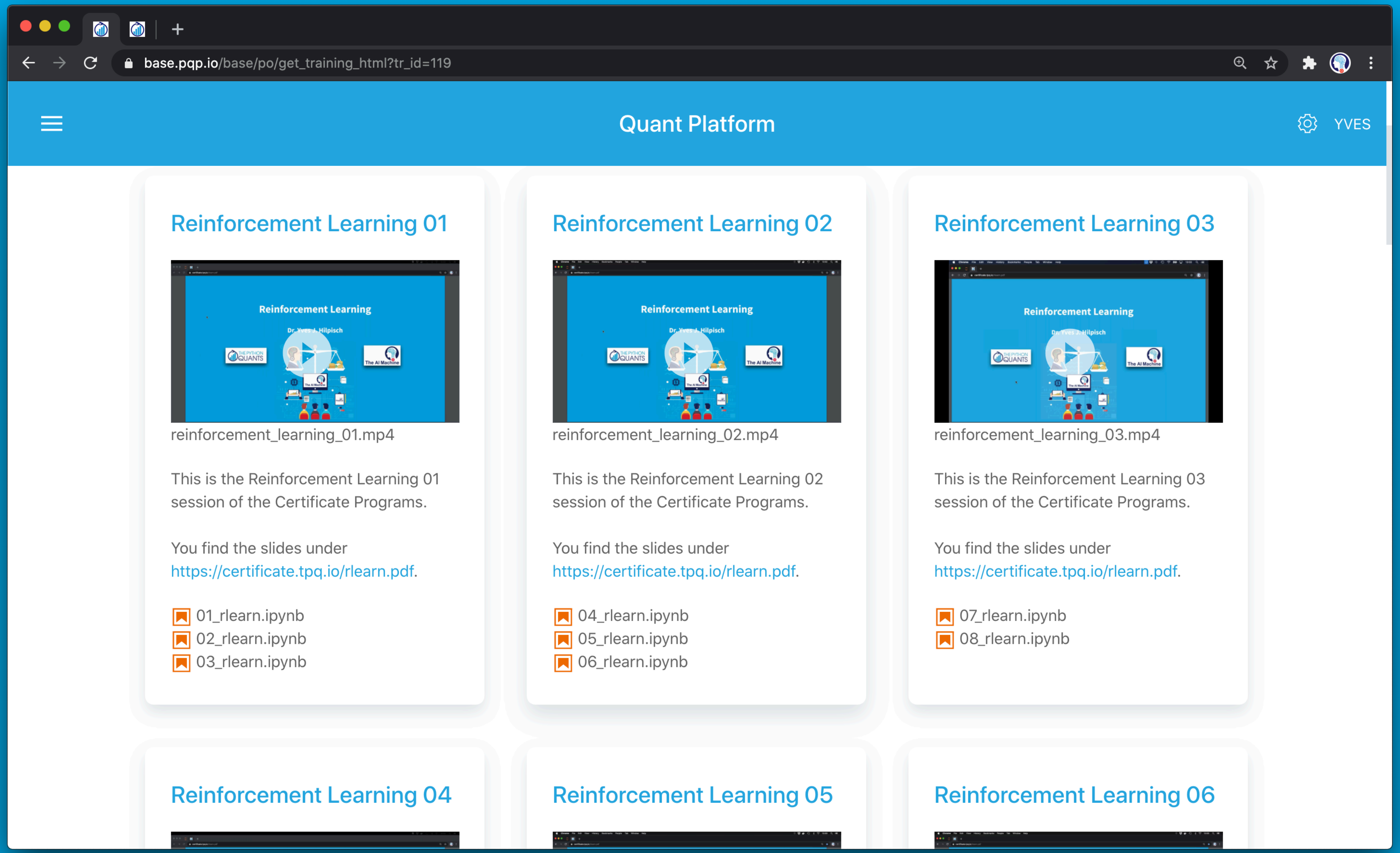

Gym
gym.openai.com/envs/CartPole-v1/
Environments Documentation
CartPole-v1
A pole is attached by an un-actuated joint to a cart, which moves along a frictionless track. The system is controlled by applying a force of +1 or -1 to the cart. The pendulum starts upright, and the goal is to prevent it from falling over. A reward of +1 is provided for every timestep that the pole remains upright. The episode ends when the pole is more than 15 degrees from vertical, or the cart moves more than 2.4 units from the center.
This environment corresponds to the version of the cart-pole problem described by Barto, Sutton, and Anderson [Barto83].
[Barto83] AG Barto, RS Sutton and CW Anderson, "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problem", IEEE Transactions on Systems, Man, and Cybernetics, 1983.
VIEW SOURCE ON GITHUB
Episode 5
RandomAgent on CartPole-v1
Environments Documentation
Gym
gym.openai.com/envs/MountainCar-v0/
Environments Documentation
MountainCar-v0
A car is on a one-dimensional track, positioned between two "mountains". The goal is to drive up the mountain on the right; however, the car's engine is not strong enough to scale the mountain in a single pass. Therefore, the only way to succeed is to drive back and forth to build up momentum.
This problem was first described by Andrew Moore in his PhD thesis [Moore90].
[Moore90] A Moore, Efficient Memory-Based Learning for Robot Control, PhD thesis, University of Cambridge, 1990.
VIEW SOURCE ON GITHUB
Episode 1
RandomAgent on MountainCar-v0
OpenAI
Gym
gym.openai.com/envs/LunarLander-v2/
Environments Documentation
LunarLander-v2
Landing pad is always at coordinates (0,0). Coordinates are the first two numbers in state vector. Reward for moving from the top of the screen to landing pad and zero speed is about 100..140 points. If lander moves away from landing pad it loses reward back. Episode finishes if the lander crashes or comes to rest, receiving additional -100 or +100 points. Each leg ground contact is +10. Firing main engine is -0.3 points each frame. Solved is 200 points. Landing outside landing pad is possible. Fuel is infinite, so an agent can learn to fly and then land on its first attempt. Four discrete actions available: do nothing, fire left orientation engine, fire main engine, fire right orientation engine.
VIEW SOURCE ON GITHUB
Episode 4
RandomAgent on LunarLander-v2
Environments Documentation
OpenAI

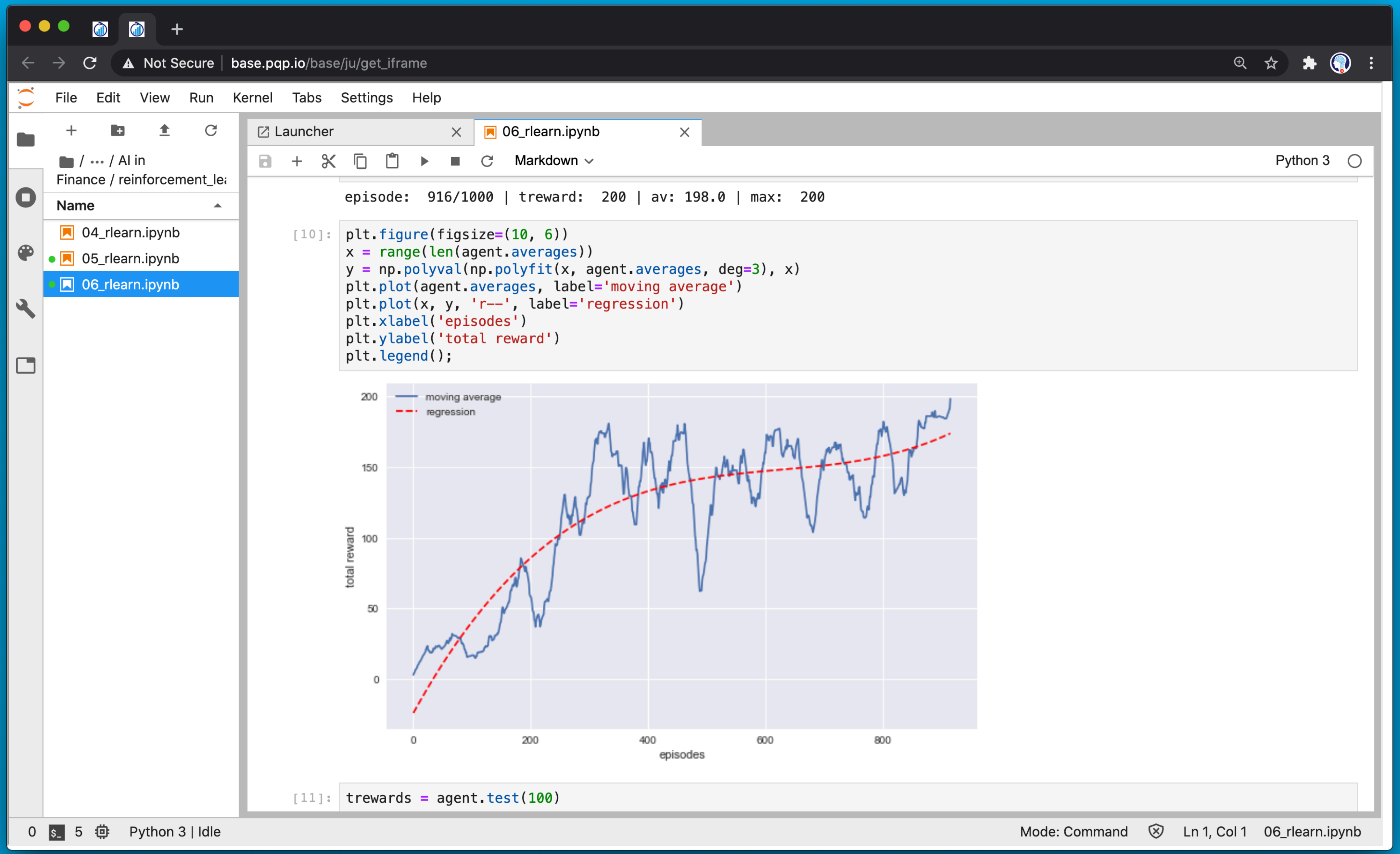

0
100
200
300
400
500
600
700


0
100
200
300
400
500
600
700

Reinforcement Learning for Finance

From playing games to trading in the financial markets



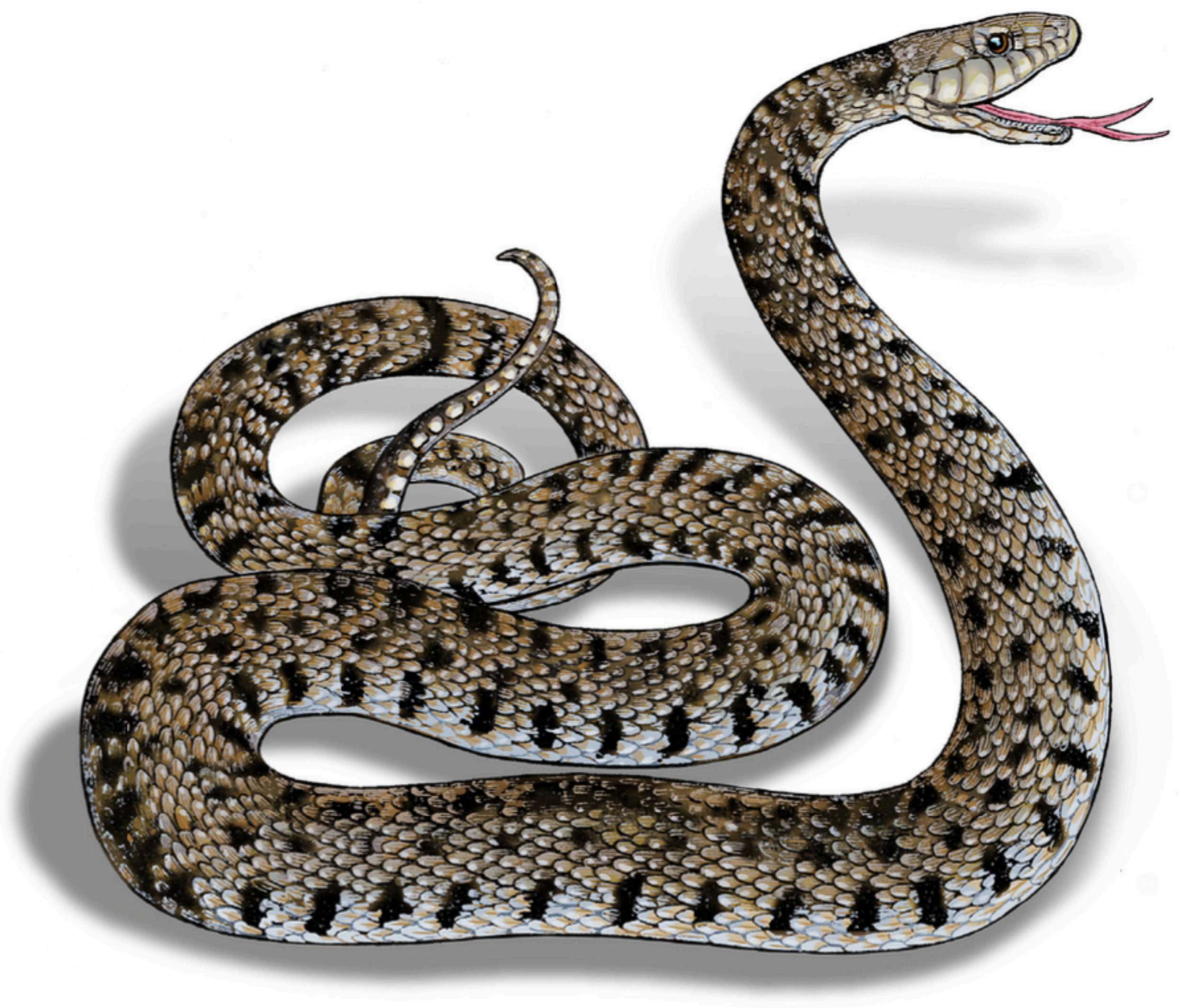


Python for Algorithmic Trading

O'REILLY

Python for Algorithmic Trading

From Idea to Cloud Deployment



Yves Hilpisch

FOUNDATIONS

- Finance with Python 01–02
- Finance with Python 03–04
- Financial Data Science

STRATEGIES

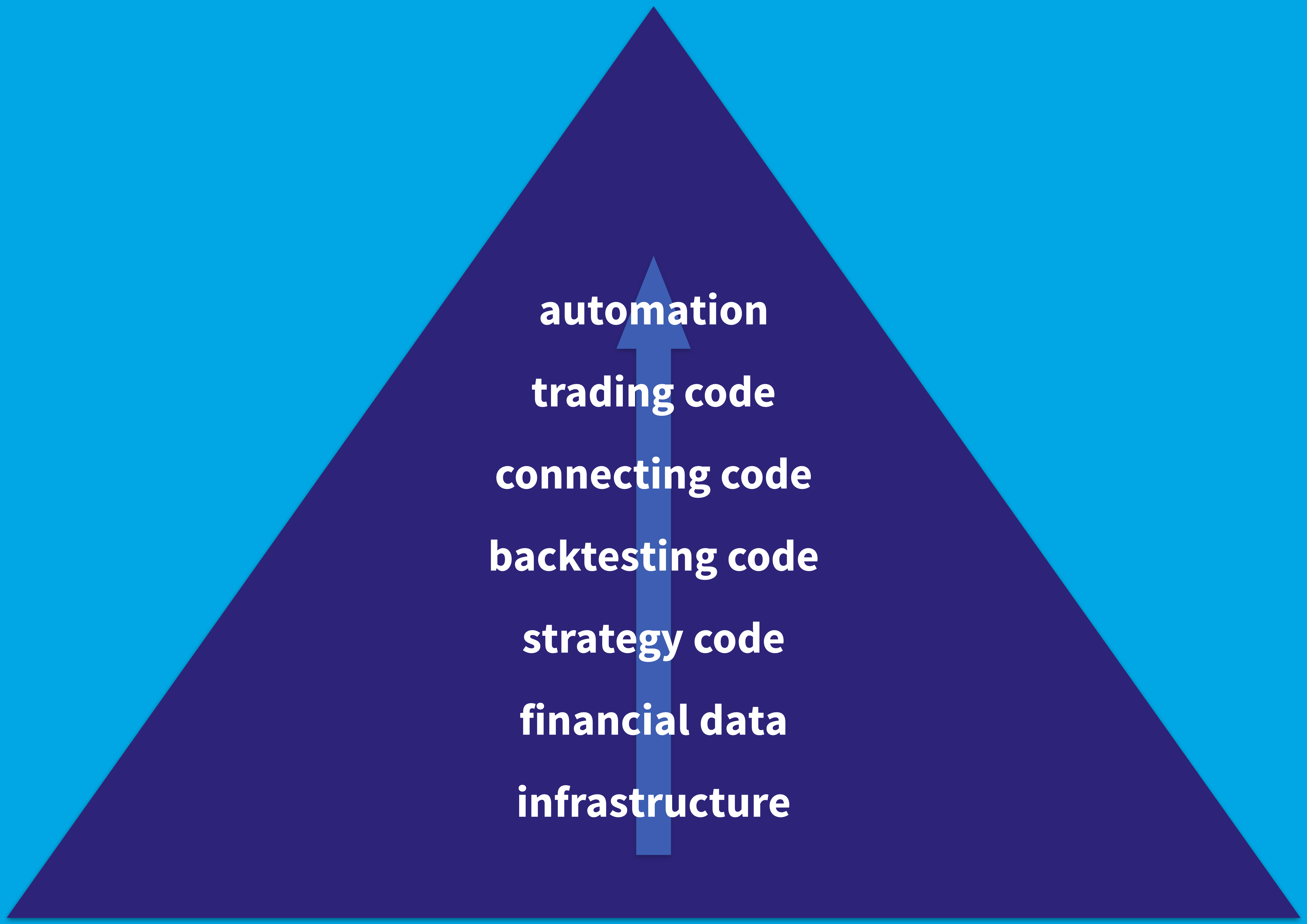
- Vectorized Backtesting
- Prediction-Based Trading
- Event-Based Backtesting

ALGO TRADING

- Real-Time Data & Streaming
- **Oanda** Trading Platform
- **FXCM** Trading Platform
- **Interactive Brokers** Trading Platform
- **Gemini** Trading Platform
- Automation & Review

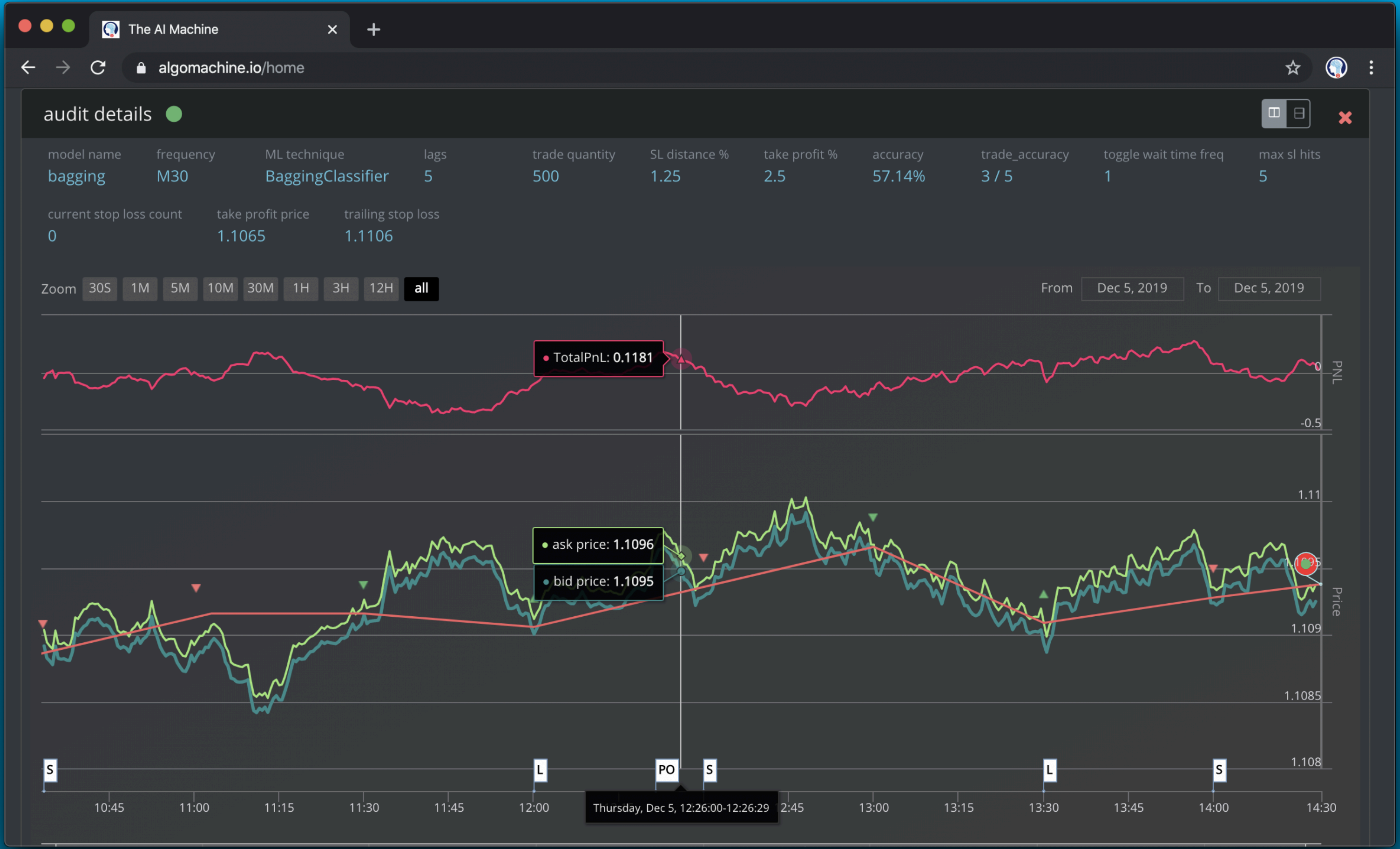
EXPERIENCE

- Practice Module 1 (“own strategy”)
- Practice Module 2 (“deployment”)
- Final Project (“research paper”)

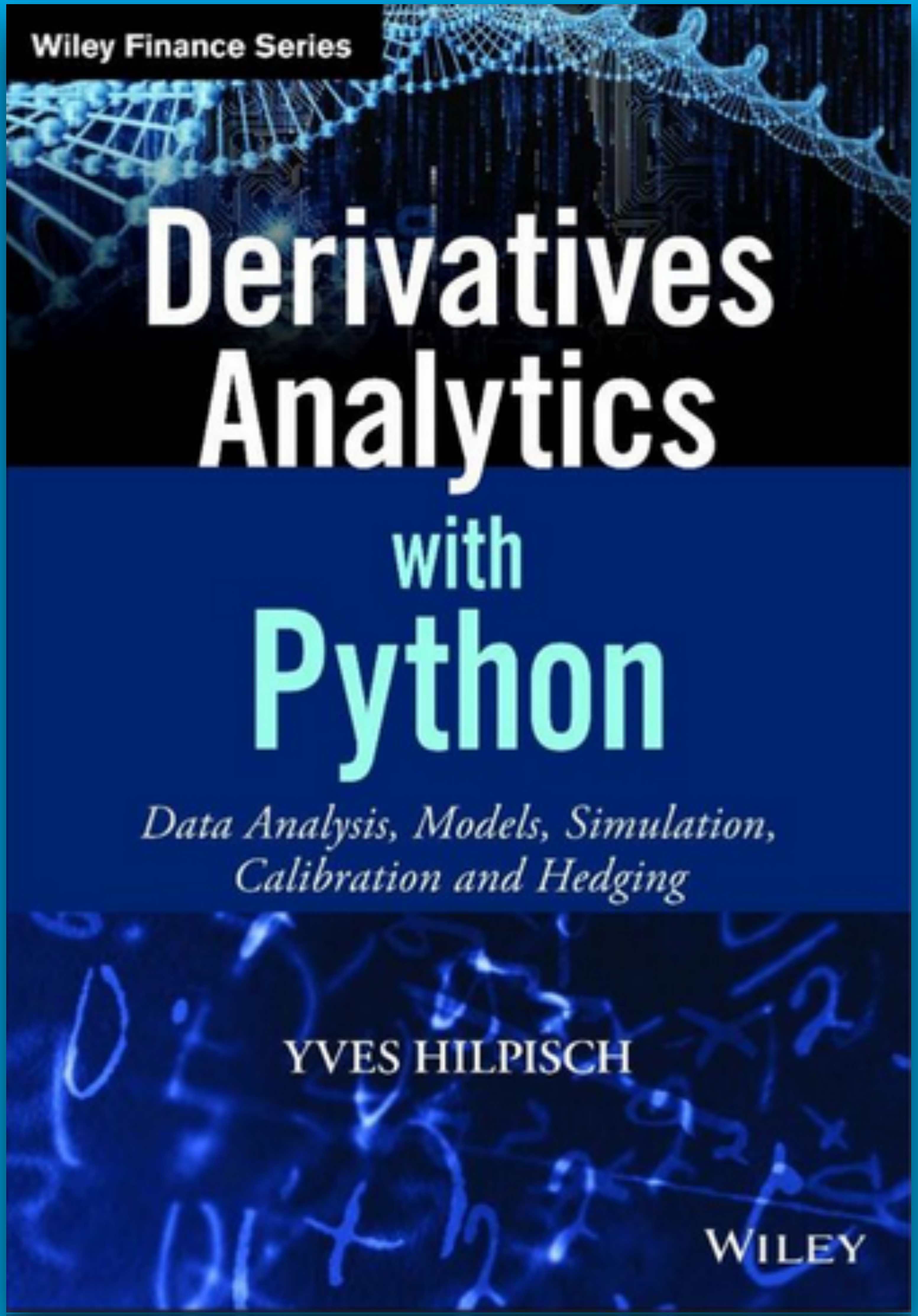

automation
trading code
connecting code
backtesting code
strategy code
financial data
infrastructure

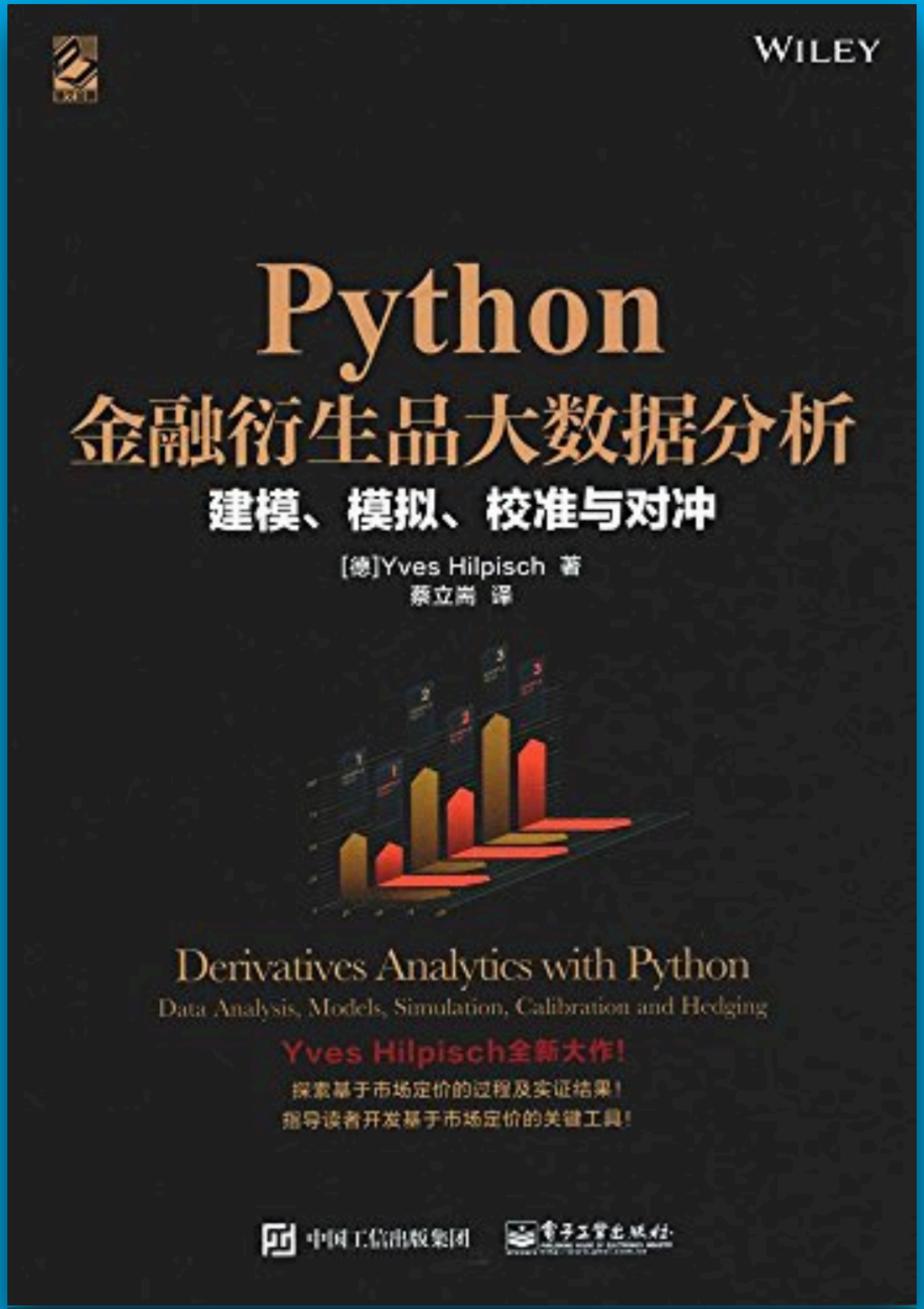
Master Class

The AI Machine



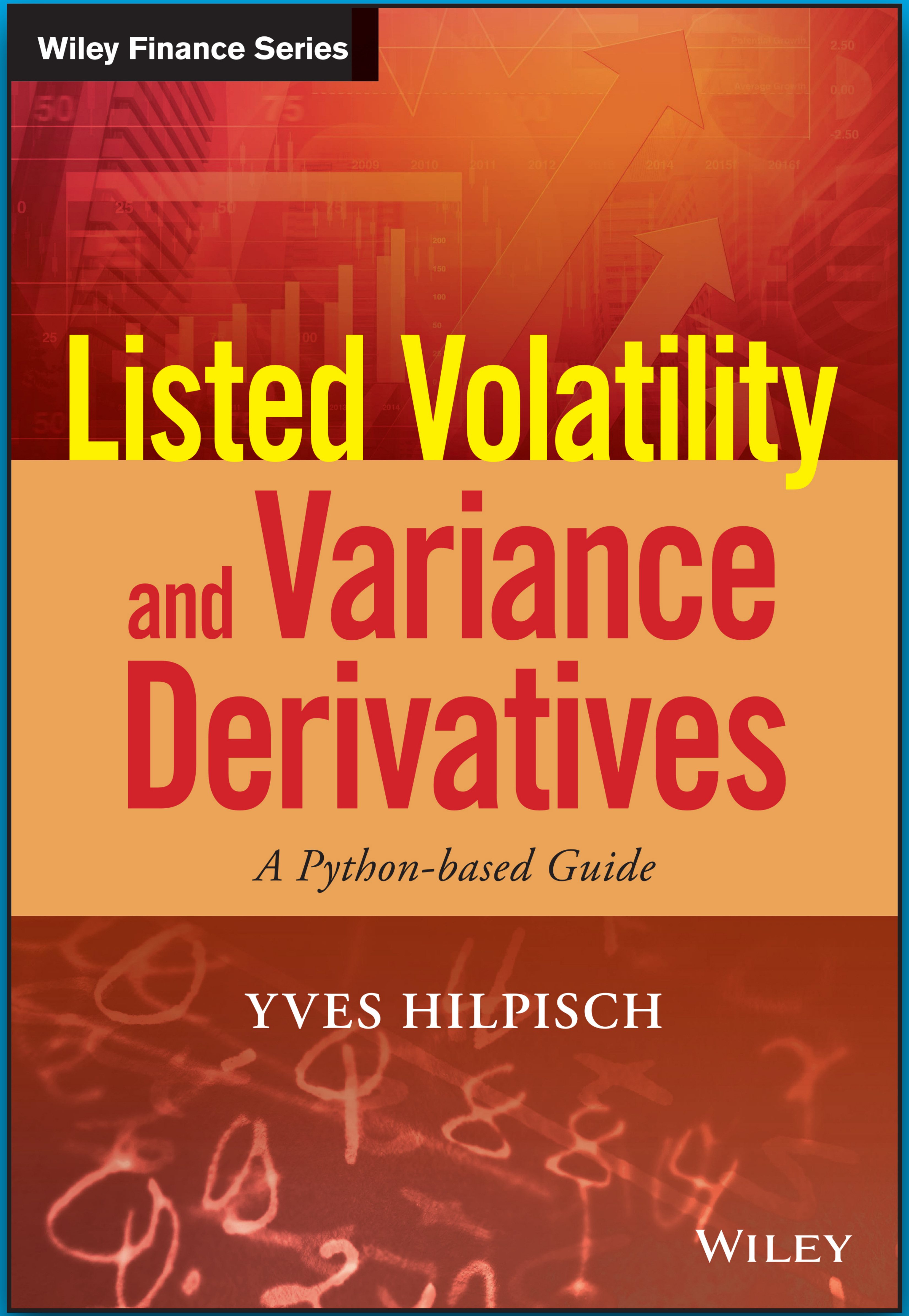
Python for Computational Finance


Wiley Finance Series
Derivatives Analytics with Python
Data Analysis, Models, Simulation, Calibration and Hedging
YVES HILPISCH
WILEY


WILEY
Python
金融衍生品大数据分析
建模、模拟、校准与对冲
[德]Yves Hilpisch 著
蔡立耑 译
Derivatives Analytics with Python
Data Analysis, Models, Simulation, Calibration and Hedging
Yves Hilpisch全新大作!
探索基于市场定价的过程及实证结果!
指导读者开发基于市场定价的关键工具!
中国工信出版集团
電子工業出版社

Computational Finance

BASICS	ADVANCED	APPLICATIONS
Market-Based Valuation	Fourier Pricing Applications	Calibration
Complete Market Models	American Options	Hedging
Risk-Neutral Valuation	General Market Model	Review
Fourier Pricing Theory	Monte Carlo Simulation	Practice


Wiley Finance Series
Listed Volatility
and Variance
Derivatives
A Python-based Guide
YVES HILPISCH
WILEY

Listed Volatility & Variance Derivatives

Trading Volatility & Variance as an Asset Class

INTRODUCTION	VOLATILITY	VARIANCE	DX Analytics	ADDON
Derivatives, Volatility and Variance	Data Analysis & Strategies	Realized Variance & Variance Swaps	An Overview	Introduction to Python
Model-Free Replication of Variance	VSTOXX Index	Variance Futures at Eurex	Square-Root Diffusion	Volatility: Terms of the VSTOXX and its Derivatives
	Valuing Volatility Derivatives		Square-Root Jump Diffusion	Variance: Trading & Settlement
	Advanced Modelling of the VSTOXX Index			

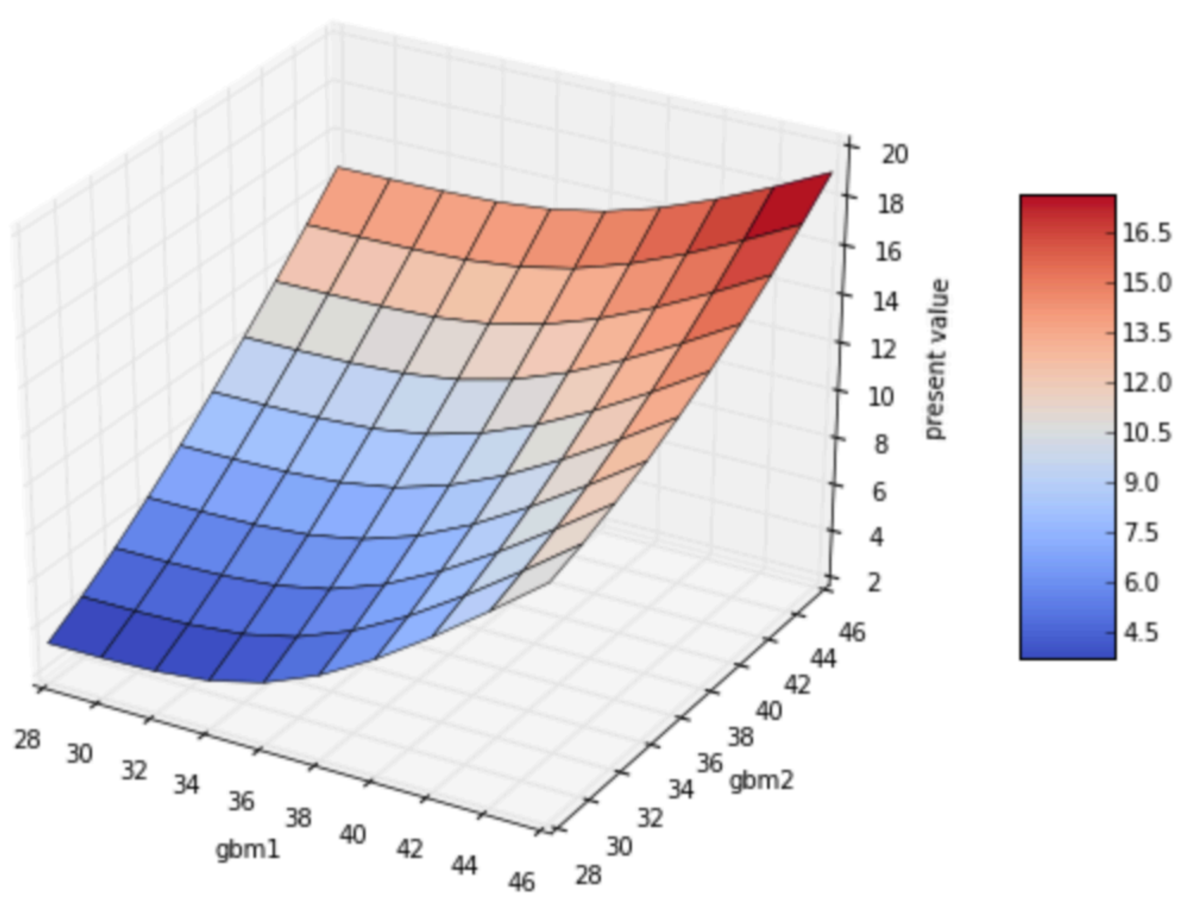
DX Analytics

DX Analytics is a **Python-based financial analytics library** which allows the modeling of rather complex derivatives instruments and portfolios. Make sure to fully understand what you are using this Python package for and how to apply it. Please also read the license text and disclaimer.

Basic Philosophy

DX Analytics is a Python-based financial analytics library that mainly implements what is sometimes called the **global valuation of (complex portfolios of) derivatives instruments** (cf. http://www.riskcare.com/files/7314/0360/6145/LowResRiskcare_Risk_0510_2.pdf). The major characteristic of this approach is the **non-redundant modeling** of all components needed for the valuation (e.g. risk factors) and the **consistent simulation and valuation** of all relevant portfolio components (e.g. correlated risk factors, multi-risk derivatives and portfolios themselves).

With DX Analytics you can, for instance, model and risk manage multi-risk derivatives instruments (e.g. American maximum call option) and generate 3-dimensional **present value surfaces** like this one:



THE PYTHON QUANTS

Navigation

Quick search

1. Quickstart

This brief first part illustrates—without much explanation—the usage of the DX Analytics library. It models two risk factors, two derivatives instruments and values these in a portfolio context.

[1]:
```
import dx
import datetime as dt
import pandas as pd
from pylab import plt
plt.style.use('seaborn')
```

1.1. Risk Factor Models ¶

The first step is to define a **model for the risk-neutral discounting**.

[2]:
```
r = dx.constant_short_rate('r', 0.01)
```

We then define a **market environment** containing the major parameter specifications needed,

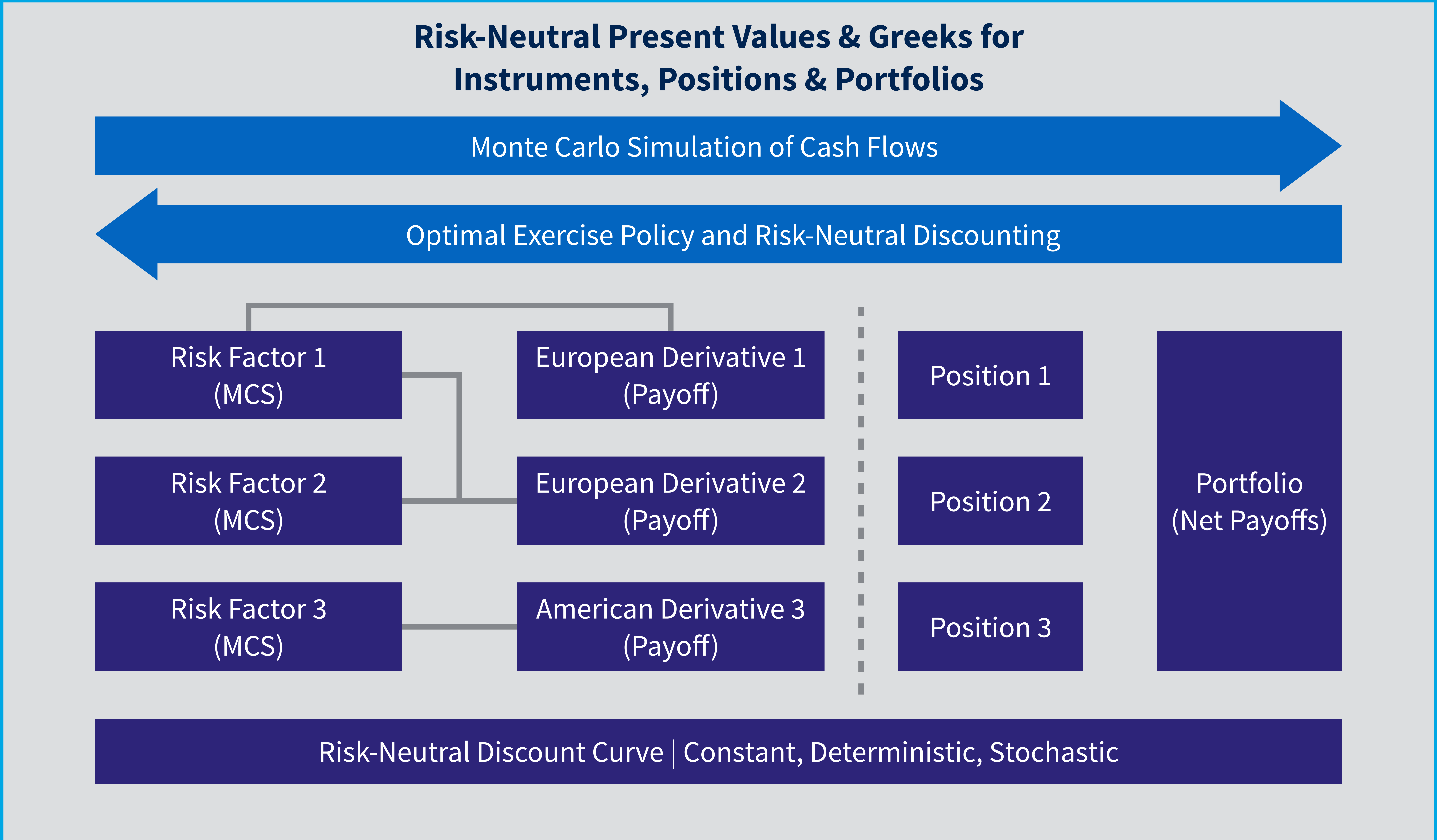
[3]:
```
me_1 = dx.market_environment('me', dt.datetime(2016, 1, 1))
```

[4]:
```
me_1.add_constant('initial_value', 100.)
  # starting value of simulated processes
me_1.add_constant('volatility', 0.2)
  # volatiltiy factor
me_1.add_constant('final_date', dt.datetime(2017, 6, 30))
  # horizon for simulation
me_1.add_constant('currency', 'EUR')
  # currency of instrument
me_1.add_constant('frequency', 'W')
  # frequency for discretization
me_1.add_constant('paths', 10000)
  # number of paths
me_1.add_curve('discount_curve', r)
  # number of paths
```

Next, the model object for the first risk factor, based on the geometric Brownian motion (Black-Scholes-Merton (1973) model).

DX Analytics

BASICS	ADVANCED	SPECIAL
Quick Start	Fourier Pricing	Multi-Risk Derivatives
Framework & Model Simulation	American Valuation	Implied Volatilities & Calibration
European Valuation	Stochastic Short Rates	Hedging
Derivatives Portfolios	Derivatives Portfolios	Complex Portfolios


Risk-Neutral Present Values & Greeks for
Instruments, Positions & Portfolios
Monte Carlo Simulation of Cash Flows
Optimal Exercise Policy and Risk-Neutral Discounting
Risk Factor 1 (MCS)
Risk Factor 2 (MCS)
Risk Factor 3 (MCS)
European Derivative 1 (Payoff)
European Derivative 2 (Payoff)
American Derivative 3 (Payoff)
Position 1
Position 2
Position 3
Portfolio (Net Payoffs)
Risk-Neutral Discount Curve | Constant, Deterministic, Stochastic


hedging
market valuation
calibration
Fourier pricing
simulation
models
financial data
infrastructure

Python for Asset Management

Python for Asset Management

Basics of Risk and Return in Finance

Mean-Variance Portfolio Theory

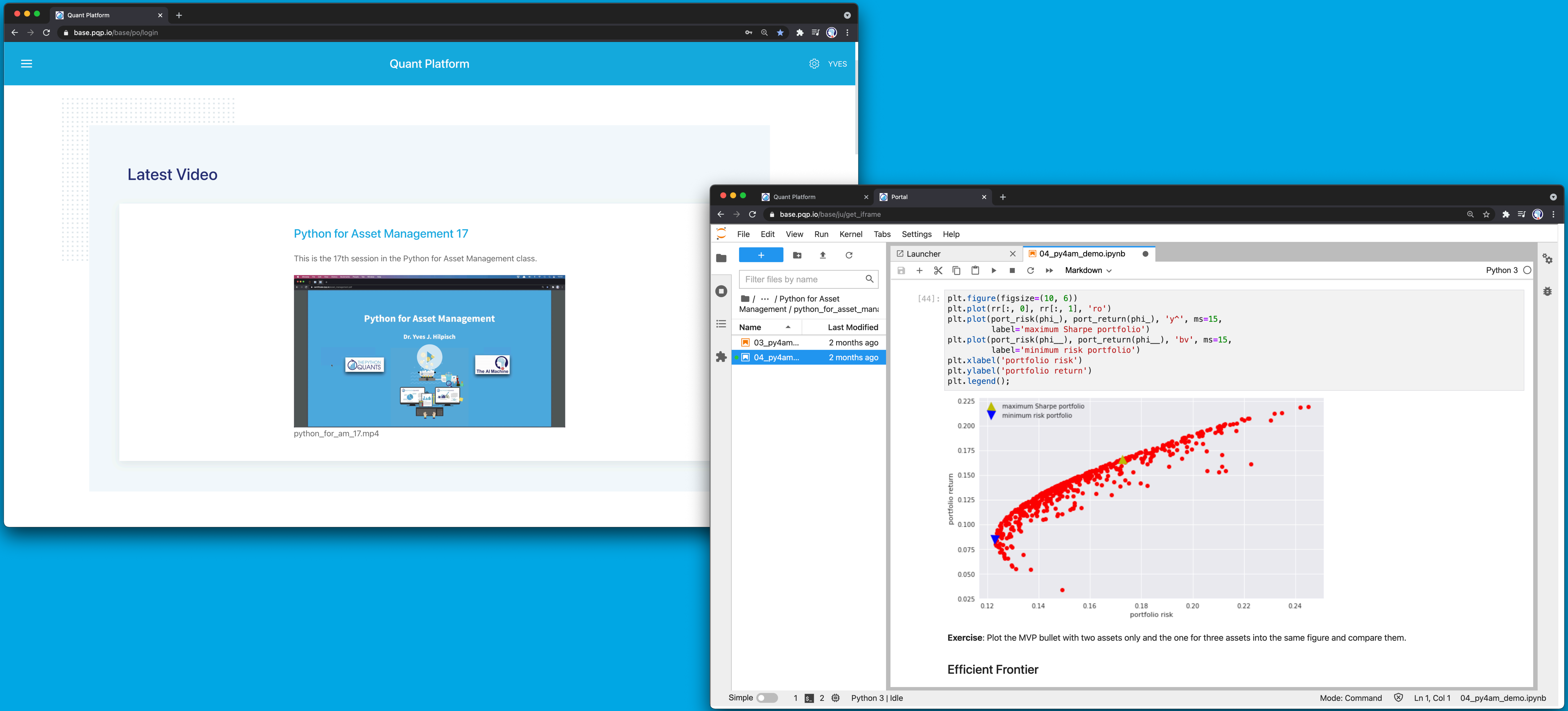
Capital Market Theory
(CAPM, APT)

Alternatives to MVP Theory
(e.g. Risk Budgeting/Parity)

AI and Machine Learning for
Asset Management

Python for Asset Management

From traditional Mean-Variance Portfolio Theory to AI for Asset Management



PORTFOLIO SELECTION*

Harry Markowitz
The Rand Corporation

The process of selecting a portfolio may be divided into two stages. The first stage starts with observation and experience and ends with beliefs about the future performances of available securities. The second stage starts with the relevant beliefs about future performances and ends with the choice of portfolio. This paper is concerned with the second stage. We first consider the rule that the investor does (or should) maximize discounted expected, or anticipated, returns. This rule is rejected both as a hypothesis to explain, and as a maximum to guide investment behavior. We next consider the rule that the investor does (or should) consider expected return a desirable thing *and* variance of return an undesirable thing. This rule has many sound points, both as a maxim for, and hypothesis about, investment behavior. We illustrate geometrically relations between beliefs and choice of portfolio according to the "expected returns—variance of returns" rule.

One type of rule concerning choice of portfolio is that the investor does (or should) maximize the discounted (or capitalized) value of future returns.[1] Since the future is not known with certainty, it must be "expected" or "anticipated" returns which we discount. Variations of this type of rule can be suggested. Following Hicks, we could let "anticipated" returns include an allowance for risk.[2] Or, we could let the rate at which we capitalize the returns from particular securities vary with risk.

The hypothesis (or maxim) that the investor does (or should) maximize discounted return must be rejected. If we ignore market imperfections the foregoing rule never implies that there is a diversified portfolio which is preferable to all non-diversified portfolios. Diversification is both observed and sensible; a rule of behavior which does not imply the superiority of diversification must be rejected both as a hypothesis and as a maxim.

* This paper is based on work done by the author while at the Cowles Commission for Research in Economics and with the financial assistance of the Social Science Research Council. It will be reprinted as Cowles Commission Paper, New Series, No. 60.

1. See, for example, J. B. Williams, *The Theory of Investment Value* (Cambridge, Mass.: Harvard University Press, 1938), pp. 55–75.

2. J. R. Hicks, *Value and Capital* (New York: Oxford University Press, 1939), p. 126. Hicks applies the rule to a firm rather than a portfolio.

For fixed probability beliefs (μ_i, σ_{ij}) the investor has a choice of various combinations of E and V depending on his choice of portfolio $X_1, \ldots, X_N$. Suppose that the set of all obtainable (E, V) combinations were as in Figure 1. The E-V rule states that the investor would (or should) want to select one of those portfolios which give rise to the (E, V) combinations indicated as efficient in the figure; i.e., those with minimum V for given E or more and maximum E for given V or less.

There are techniques by which we can compute the set of efficient portfolios and efficient (E, V) combinations associated with given μ_i

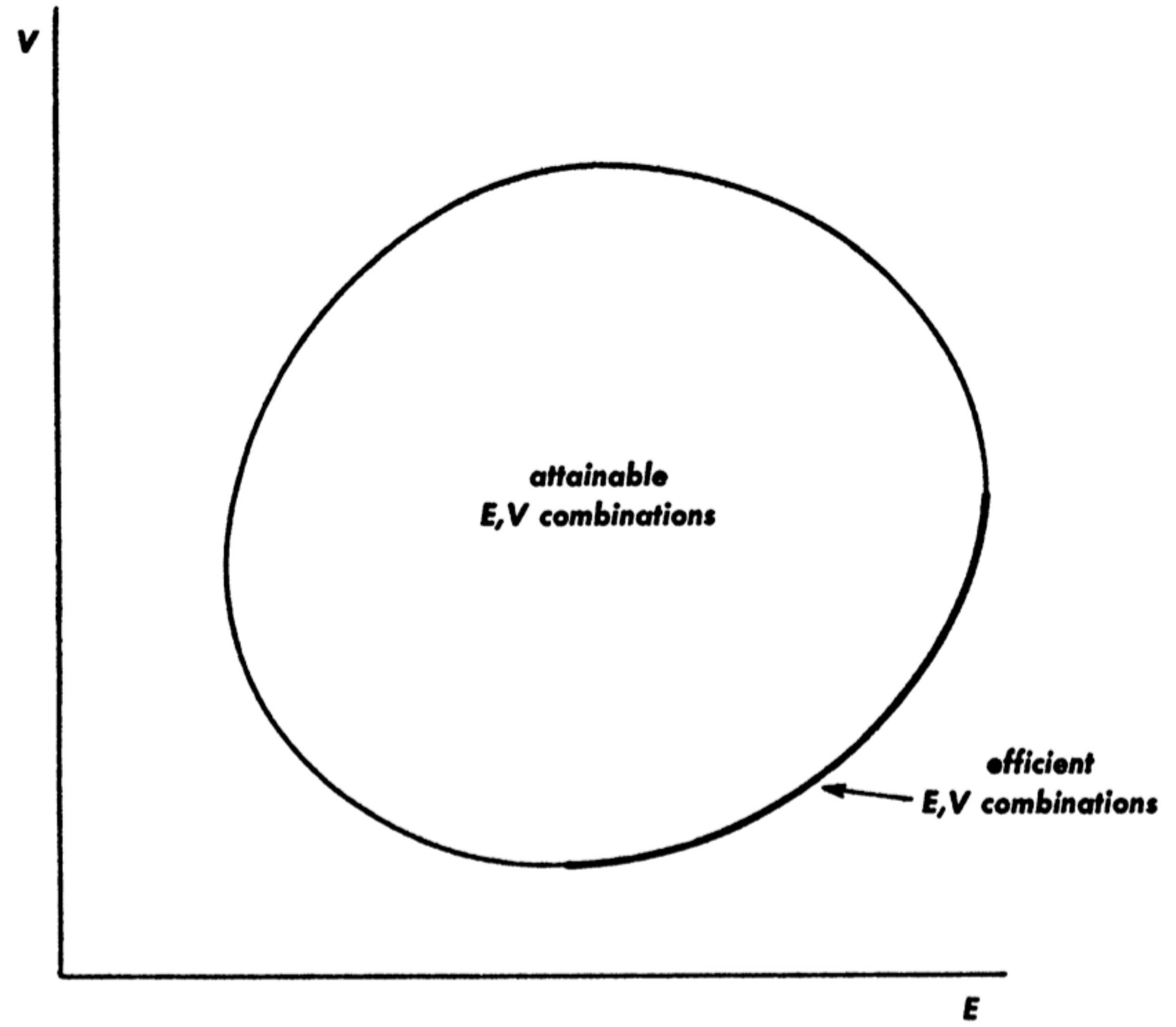


Fig. 1

and σ_{ij}. We will not present these techniques here. We will, however, illustrate geometrically the nature of the efficient surfaces for cases in which N (the number of available securities) is small.

The calculation of efficient surfaces might possibly be of practical use. Perhaps there are ways, by combining statistical techniques and the judgment of experts, to form reasonable probability beliefs (μ_i, σ_{ij}). We could use these beliefs to compute the attainable efficient combinations of (E, V). The investor, being informed of what (E, V) combinations were attainable, could state which he desired. We could then find the portfolio which gave this desired combination.

Two conditions—at least—must be satisfied before it would be practical to use efficient surfaces in the manner described above. First, the investor must desire to act according to the E-V maxim. Second, we must be able to arrive at reasonable μ_i and σ_{ij}. We will return to these matters later.

Let us consider the case of three securities. In the three security case our model reduces to

$$1)\quad E = \sum_{i=1}^{3} X_i \mu_i$$

$$2)\quad V = \sum_{i=1}^{3} \sum_{j=1}^{3} X_i X_j \sigma_{ij}$$

$$3)\quad \sum_{i=1}^{3} X_i = 1$$

$$4)\quad X_i \geqslant 0 \quad \text{for} \quad i = 1, 2, 3 .$$

From (3) we get

$$3')\quad X_3 = 1 - X_1 - X_2$$

If we substitute (3′) in equation (1) and (2) we get E and V as functions of X_1 and X_2. For example we find

$$1')\quad E = \mu_3 + X_1(\mu_1 - \mu_3) + X_2(\mu_2 - \mu_3)$$

The exact formulas are not too important here (that of V is given below).[8] We can simply write

$$a)\quad E = E(X_1, X_2)$$

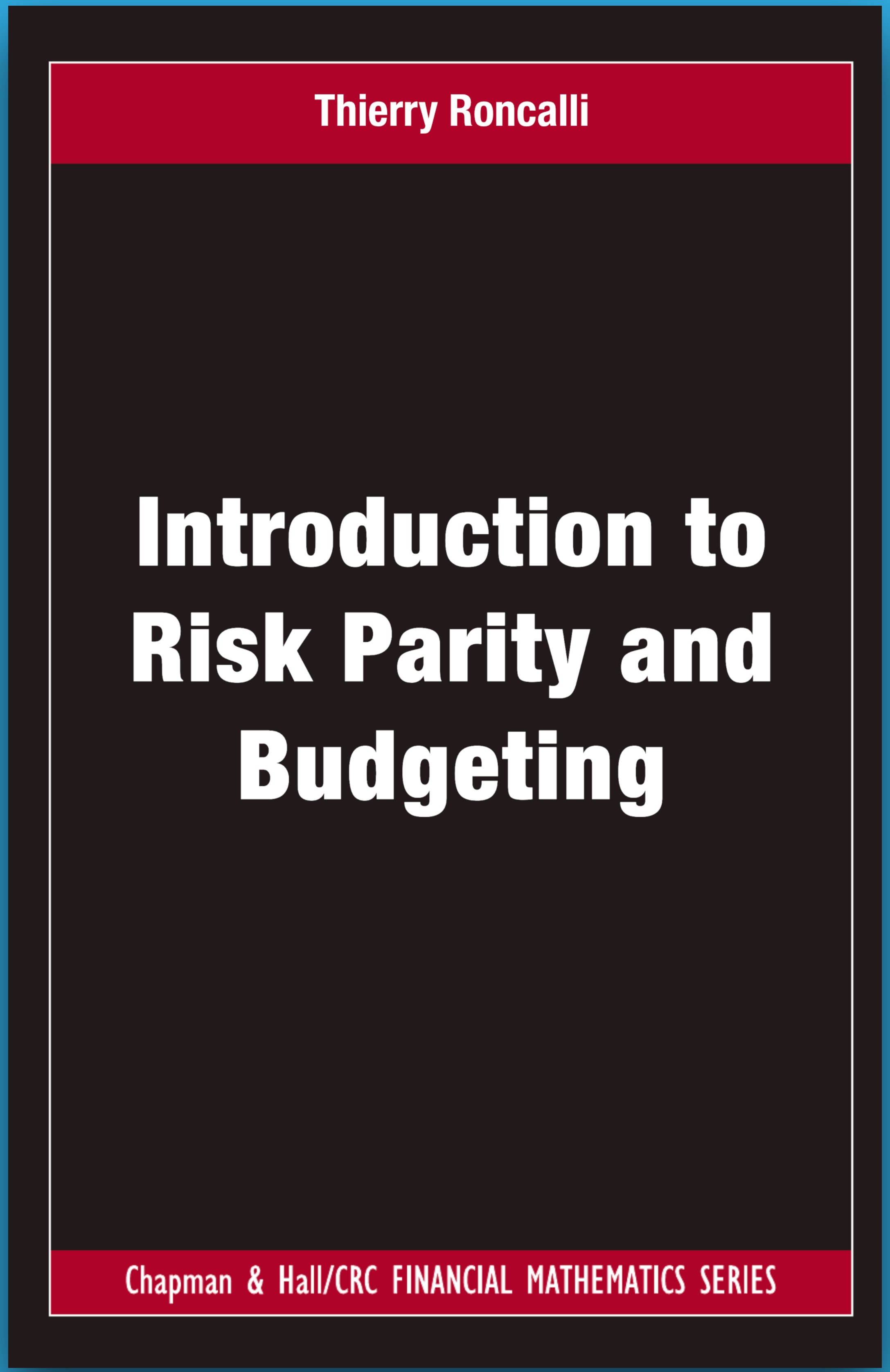
$$b)\quad V = V(X_1, X_2)$$

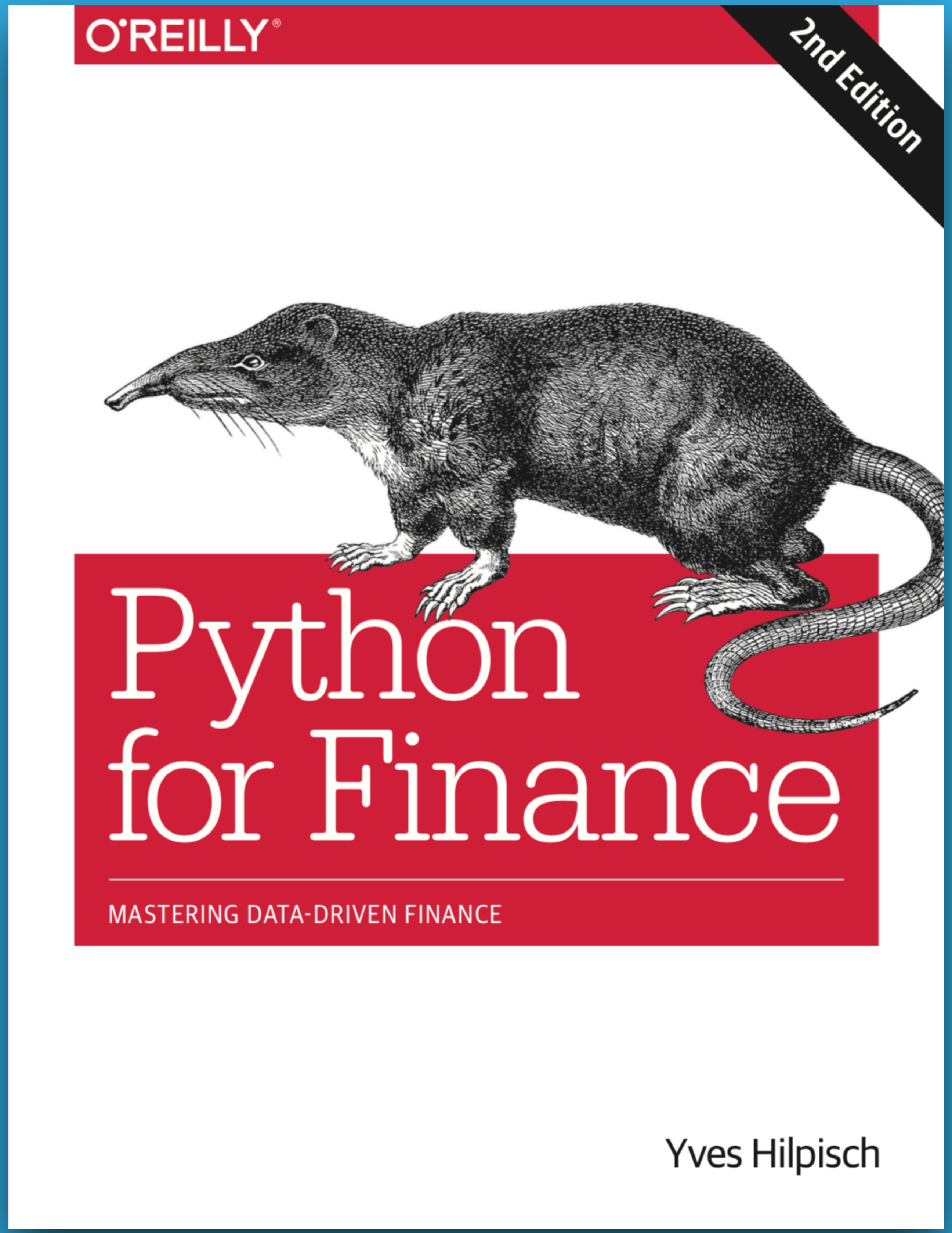
$$c)\quad X_1 \geqslant 0,\ X_2 \geqslant 0,\ 1 - X_1 - X_2 \geqslant 0$$

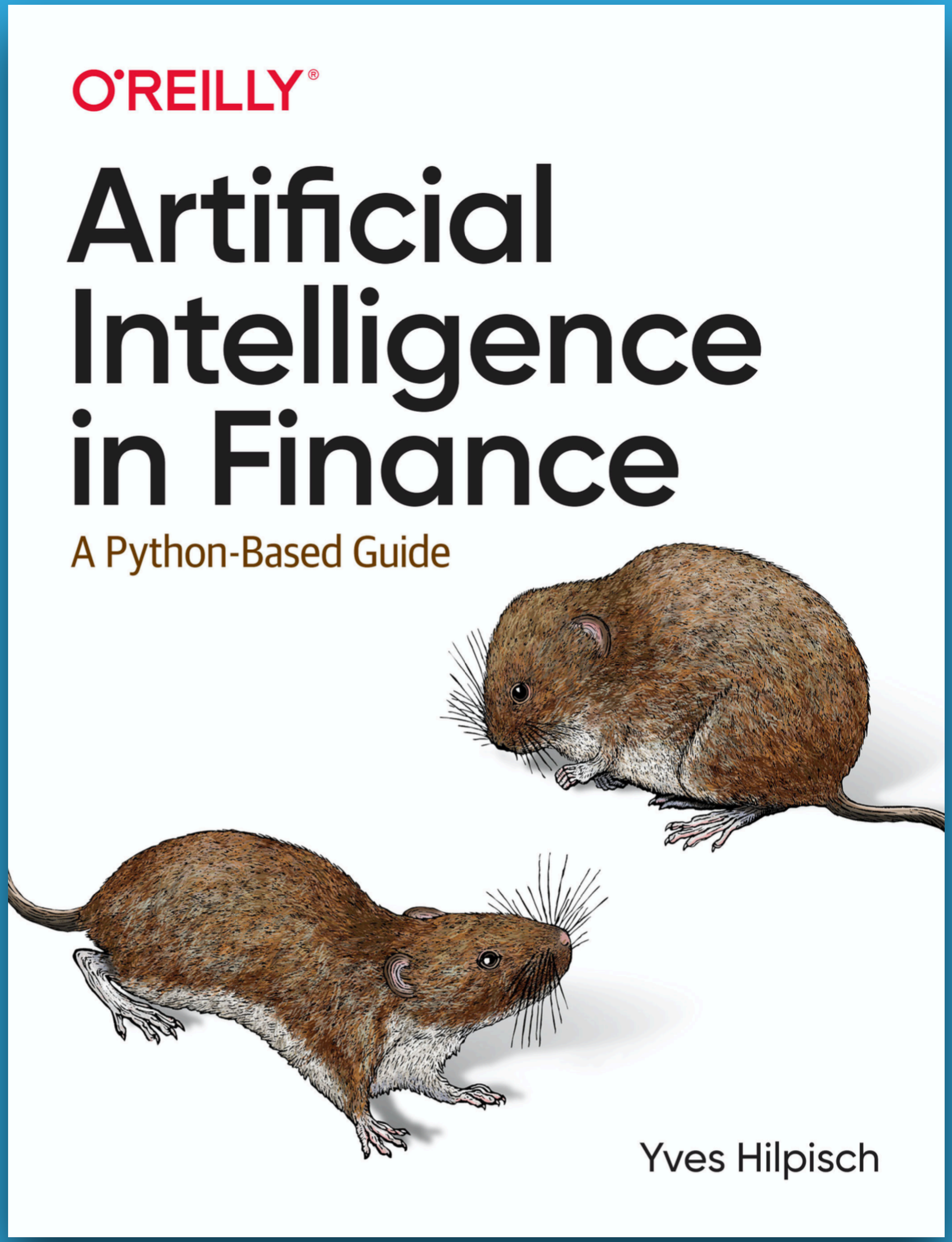
By using relations (a), (b), (c), we can work with two dimensional geometry.

The attainable set of portfolios consists of all portfolios which satisfy constraints (c) and (3′) (or equivalently (3) and (4)). The attainable combinations of X_1, X_2 are represented by the triangle $\overline{abc}$ in Figure 2. Any point to the left of the X_2 axis is not attainable because it violates the condition that $X_1 \geqslant 0$. Any point below the X_1 axis is not attainable because it violates the condition that $X_2 \geqslant 0$. Any

8. $V = X_1^2(\sigma_{11} - 2\sigma_{13} + \sigma_{33}) + X_2^2(\sigma_{22} - 2\sigma_{23} + \sigma_{33}) + 2X_1X_2(\sigma_{12} - \sigma_{13} - \sigma_{23} + \sigma_{33}) + 2X_1(\sigma_{13} - \sigma_{33}) + 2X_2(\sigma_{23} - \sigma_{33}) + \sigma_{33}$


Thierry Roncalli
Introduction to Risk Parity and Budgeting
Chapman & Hall/CRC FINANCIAL MATHEMATICS SERIES


O'REILLY
2nd Edition
Python for Finance
MASTERING DATA-DRIVEN FINANCE
Yves Hilpisch


O'REILLY
Artificial Intelligence in Finance
A Python-Based Guide
Yves Hilpisch

CASE STUDIES & DEMOS

1. Financial Packages
2. Model Calibration
3. Market Prediction
4. Oanda Trading Platform

Link to Gist: http://bit.ly/cert_intro

Study Plans for the Programs

TPQ Certificate Program (16 weeks)

W	Asset Management	Algorithmic Trading	Computational Finance	Tools & Skills	Financial Data Science	AI in Finance (Book)	Optional	Live Sessions
1	**Finance with Python 01, 02** Financial Theory with Python Chs 01-03			Tools & Skills 01 **(Python Environments)**	**Data Types & Data Structures 01** Python for Finance Chs 01-02	**AI in Finance 01** plus Ch 01	AI in Finance 01 (see ***AI in Finance*** class) PyExcel 01	**Intro & Overview (08.05.23)** **Linux 01 (10.05.23)** **Linux 02 (11.05.23)**
2	**Finance with Python 03, 04** Financial Theory with Python Ch 04			Tools & Skills 02 **(Docker Usage, Jupyter)**	**Data Types & Data Structures 02** Python for Finance Ch 03	**AI in Finance 02** plus Ch 02	AI in Finance 02 PyExcel 02	
3	**Finance with Python 05, 06** Financial Theory with Python Chs 05-06			Tools & Skills 03 **(Cloud Usage, Jupyter)**	**Numerical Computing with NumPy** Python for Finance Ch 04	**AI in Finance 03** plus Ch 03	AI in Finance 03 PyExcel 03 OOP 01	
4	**Python for Asset Management 01** **Python for Asset Management 02**	**Vectorized Backtesting** PyAlgo Chs 01-04	**Market Based Valuation** **DX Quick Start** **DAWP Chs 01-03**	Tools & Skills 01 Special **(IPy)** Tools & Skills 02 Special **(xonsh)**	**Data Analysis with pandas** Python for Finance Ch 05	**AI in Finance 04** plus Ch 04	AI in Finance 04 RL for Finance 01 PyExcel 04 OOP 02	
5	**Python for Asset Management 03** **Python for Asset Management 04**	**Predicting Market Movements** PyAlgo Ch 05	**Complete Market Models DX Frame & Simulation** **DAWP Ch 05**	Tools & Skills 04 **(Vim Code Editor)**	**Object Oriented Programming** Python for Finance Ch 06	**AI in Finance 05** plus Ch 05	AI in Finance 05 RL for Finance 02 PYDB 01 OOP 03	
6	**Python for Asset Management 05** **Python for Asset Management 06**	**Event-based Backtesting** PyAlgo Ch 06	**Risk-Neutral Valuation** **DX European Valuation** DAWP Ch 04 ***LVVD 01***	Tools & Skills 05 **(Screen + Vim + q)**	**Visualization & Financial Time Series** Python for Finance Chs 07-08	**AI in Finance 06** plus Ch 06	AI in Finance 06 RL for Finance 03 PYDB 02	
7	**Python for Asset Management 07** **Python for Asset Management 08**	**Real-Time Data Handling & Viz** PyAlgo Ch 07	**Fourier Pricing** **DX Fourier Pricing** DAWP Ch 06 ***LVVD 02***	Tools & Skills 06 **(Doctest & Unittest)**	**Input-Output Operations** Python for Finance Ch 09	**AI in Finance 07** plus Ch 07	AI in Finance 07 RL for Finance 04 PYDB 03	

https://bit.ly/tpq_4m

Quant Platform | TPQ Certificate Programs (16 v | TPQ Platinum Package SELF-P

docs.google.com/document/d/1DKzlzir_q_BTtpVCJBbKkH0olFl1o3ekG5XK_82szzw/edit

TPQ Platinum Package SELF-PACED (1 yr)

File Edit View Tools Help

Request edit access | Share

Study Plan for the 1st Trimester | Focus: *Asset Management*

Week	Asset Management	Data Science	AI in Finance (Book)	Basics	Tutorial
1	**Finance with Python 01, 02** Finance with Python Chs 01-03	Tools & Skills 01 **(Python Environments)**	AI in Finance 01 plus Ch 01	Math Basics 01 Math Basics 02	-
2	**Finance with Python 03, 04** Finance with Python Ch 04	Tools & Skills 02 **(Docker Usage, Jupyter)**	AI in Finance 02 plus Ch 02	Math Basics 03 Math Basics 04	Tutorial 01
3	**Finance with Python 05, 06** Finance with Python Chs 05-06	Tools & Skills 03 **(Cloud Usage, Jupyter)**	AI in Finance 03 plus Ch 03	Math Basics 05 Math Basics 06	Tutorial 02 Review 01 (PyAM)
4	**Python for Asset Management 01** **Python for Asset Management 02**	Tools & Skills S 01 **(IPy)** Tools & Skills S 02 **(xonsh)**	AI in Finance 04 plus Ch 04	Math Basics 07 Math Basics 08	Tutorial 03
5	**Python for Asset Management 03** **Python for Asset Management 04**	Tools & Skills 04 **(Vim Code Editor)**	AI in Finance 05 plus Ch 05	Math Basics 09 Math Basics 10	Tutorial 04
6	**Python for Asset Management 05** **Python for Asset Management 06**	Tools & Skills 05 **(Screen + Vim + q)**	AI in Finance 06 plus Ch 06	Math Basics 11 Math Basics 12	Tutorial 05 Review 02 (PyAM)
7	**Python for Asset Management 07** **Python for Asset Management 08**	Tools & Skills 06 **(Doctest & Unittest)**	AI in Finance 07 plus Ch 07	Math Basics 13 Math Basics 14	Tutorial 06
8	**Python for Asset Management 09** **Python for Asset Management 10**	Tools & Skills 07 **(Git Version Control)**	AI in Finance 08 plus Ch 08	Math Basics 15 Math Basics 16	Tutorial 07
9	**Python for Asset Management 11** **Python for Asset Management 12**	Tools & Skills 08 **(Python Packaging)**	AI in Finance 09 plus Ch 09	Math Basics 17 Math Basics 18	Tutorial 08 Review 03 (PyAM)
10	**Python for Asset Management 13** **Python for Asset Management 14**	Tools & Skills 09 **(Documentation)**	AI in Finance 10 plus Ch 10	Math Basics 19 Math Basics 20	Tutorial 09
11	**Python for Asset Management 15** **Python for Asset Management 16**	Tools & Skills 10 **(Code Hosting/ Case)**	AI in Finance 11 plus Ch 11	Math Basics 21 Math Basics 22	Tutorial 10
12	**Python for Asset Management 17**	Tools & Skills S 03 **(Sublime)**	AI in Finance 12 plus Chs 12-14	Math Basics 23 Math Basics 24	Review 04 (PyAM)

PLUS 4 WEEKS REVIEW & STUDY PHASE

https://bit.ly/tpq_oy

Guiding Principles

Guiding Principles

Python First

Coding and implementation are the focus, rather than theory or practical considerations.

Specific

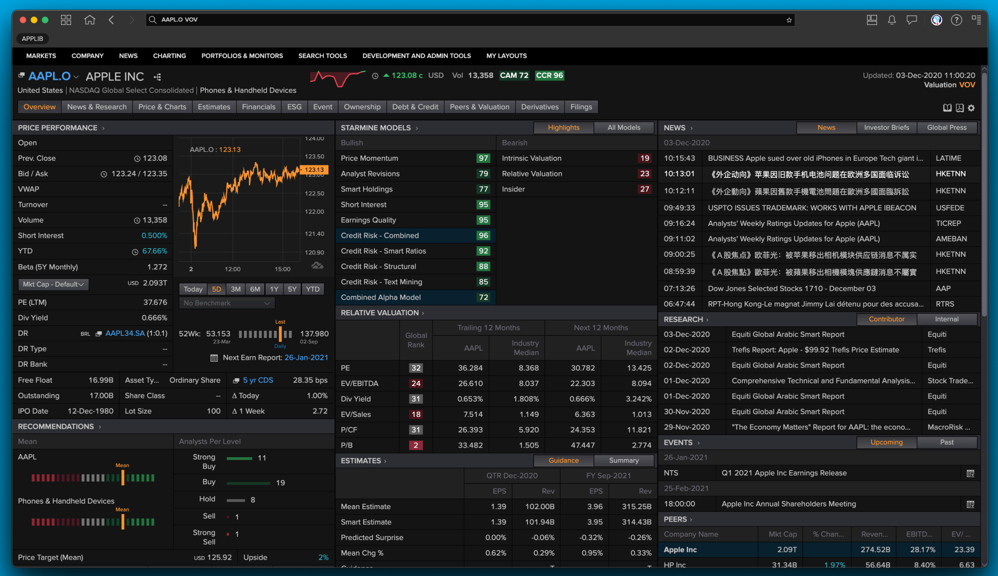
Algorithms used & examples shown are specific in nature and not meant to provide an exhaustive overview.

```
In [60]: P  1
Out[60]: array([0.4, 0.6])

In [61]: S0 = 10  2

In [62]: S1 = np.array((20, 5))  3

In [63]: np.dot(P, S1)  4
Out[63]: 11.0
```

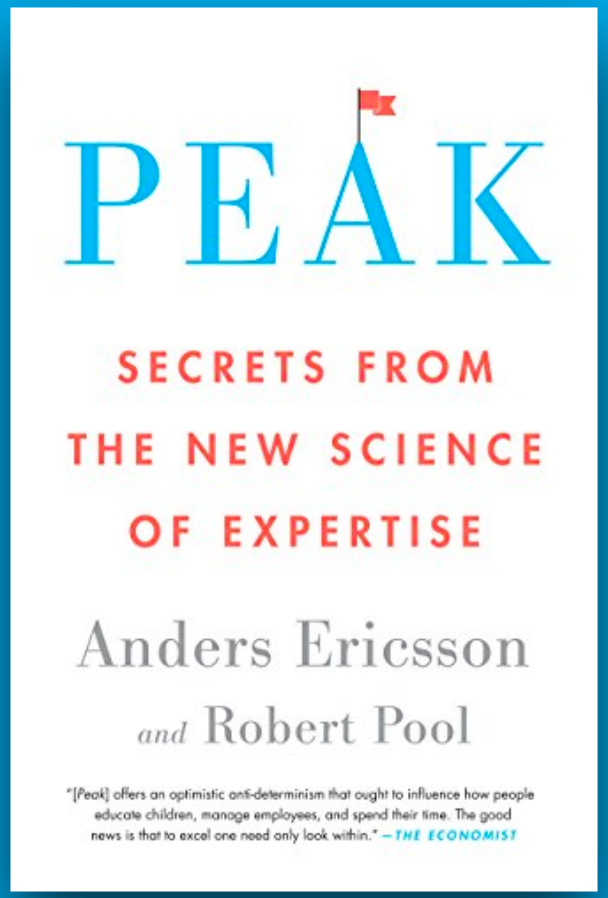


Reproducible

All examples are based on static data sets to allow for reproducibility of results.

Practical

To acquire coding and other practical skills is the main goal. Therefore <u>skill acquisition</u> is indispensable.



Might also be noteworthy …

Different Levels

The classes and sessions will be easy for some and difficult for others. Delegates might be beginners or experts.

MacOS

Most classes are done on MacOS and should be straightforward to implement on Linux. With small adjustments also on Windows.

Sometimes Opportunistic

The classes and sessions sometimes do not provide a comprehensive, systematic treatment of the topics touched upon.

Delayed Rewards

Delegates should not only expect “immediate rewards”. Often, it’s better to follow closely, to digest, and revisit the materials later.

Review Questions,
Exercises & Test Projects

THE PYTHON QUANTS

Certificate Program in Python for Algorithmic Trading

Review Questions
Weeks 01, 02, & 03

With regard to the the topics covered in the first three weeks, you might review the materials based on the following questions. The review questions focus mainly on the **big picture**.

- **Python in General**
  - Which data types did you learn about?
  - What basic data structures provides Python you with?
  - What is the basic syntax of a "for loop"?
  - What is the syntax of a list comprehension? (make an example)
  - Why is OOP a useful programming paradigm?

- **Finance with Python**
  - Name three approaches to price a (European) option in a complete market model.
  - Why is (freedom of) arbitrage such a strong argument in the context of financial pricing?
  - What is a martingale measure and how can it be used to price options?
  - What is the expected utility approach all about and how does it model decision making under uncertainty?
  - What problems arise in the context of an incomplete market (model)?
  - Explain the difference between a static and a dynamic financial model (for pricing purposes).

- **Tools & Skills**
  - Which dual role does the tool conda play?
  - Why does it make sense to work with Python environments?
  - How do you create an environment with conda?
  - How do you install Python packages to such an environment?
  - How do you delete a Python environment?
  - Why is Docker a helpful, platform-independent technology?
  - How does it help you with managing Python installations?
  - What are the benefits of a cloud-based Python installation compared to a local one?
  - Why, do you think, a cloud infrastructure is indispensable when running algorithmic trading strategies in automated fashion?

- **Financial Data Science**
  - What are typical real world problems that you face with regard to (financial) data sets?
  - What are basic approaches to process CSV files with Python?
  - Why is NumPy such a helpful package for numerical computing and financial analytics in particular?
  - Explain the benefits of specialized data structures (eg ndarray object) as compared to more general ones (eg list object).
  - What are vectorized operations (with NumPy) and what are the benefits of this programming paradigm?
  - How do you generate random numbers with NumPy?

- **AI in Finance**
  - What do people understand under the Technological Singularity (TS)?
  - What basic paths consider researchers possible to reach the TS?
  - Do you personally believe a TS is possible and why or why not?
  - What do people understand under the Financial Singularity and how might it be achieved technologically?
  - What is meant by data-driven finance and which technological advances drive it?

How does it enable AI-first finance and what do you understand when you hear this expression?
What is the difference between a normative and a positive approach to finance?
How does it relate to the history of financial economics and to an AI-first future of finance?
Name three financial theories that might be considered elegant but not realistic (given empirical support).

Finance with Python

Arbitrage Pricing

Assume two traded assets are given, a risky one (stock) and a risk-less one (bond), with prices $S_0 = 10, B_0 = 1$. Their future payoffs are:

$$S_1 = \begin{pmatrix} 15 \\ 5 \end{pmatrix}$$

$$B_1 = \begin{pmatrix} 1.01 \\ 1.01 \end{pmatrix}$$

A European put option is introduced to the market with payoff:

$$P_1 = \begin{pmatrix} 0 \\ 3 \end{pmatrix}$$

Questions:

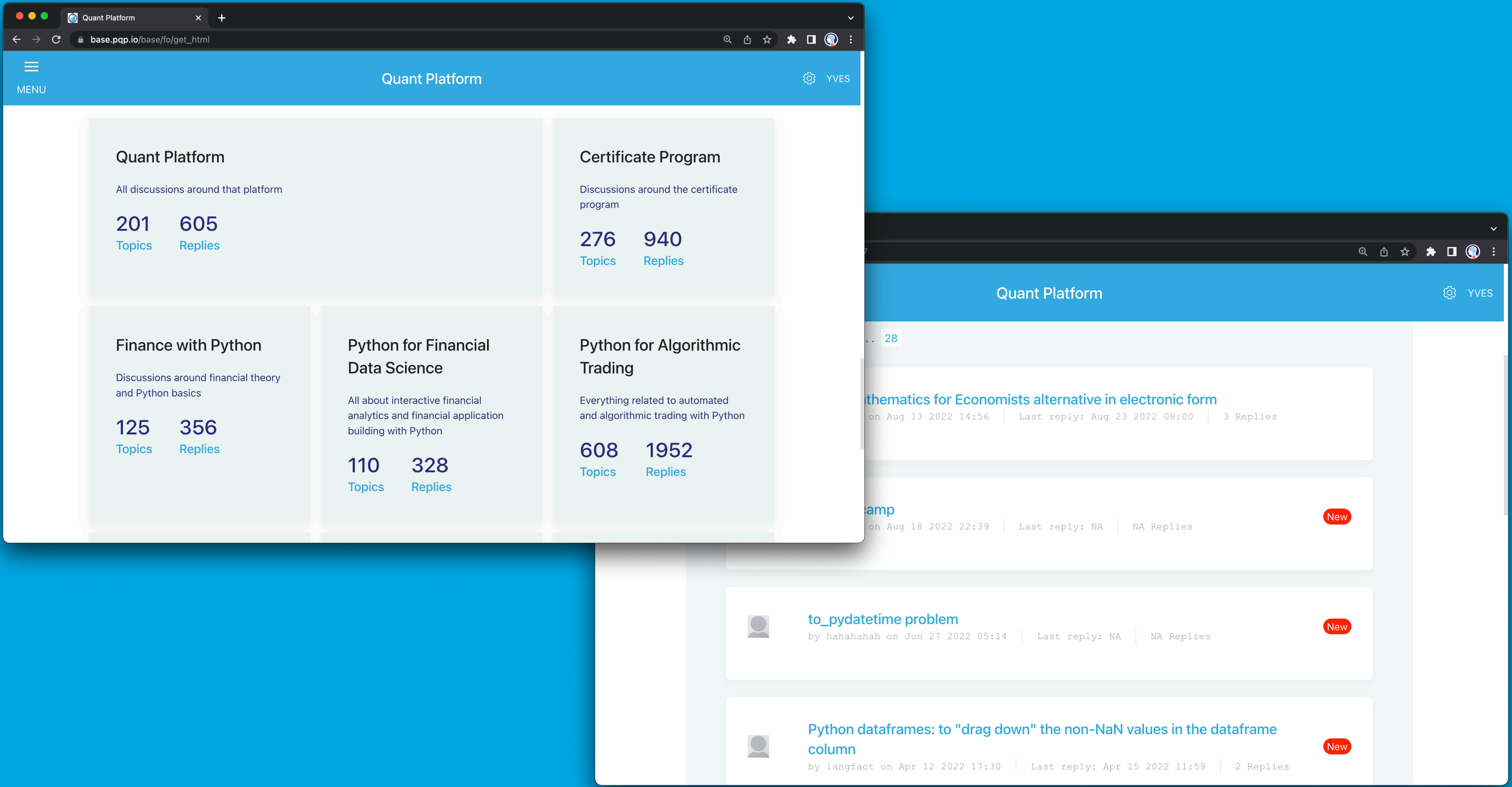
- What is the strike price of the put option?
- Which portfolio replicates the put option payoff perfectly?
- Is there a difference when using OLS regression for the replication?
- How can a learning algorithm be used to replicate the payoff?

Test Project

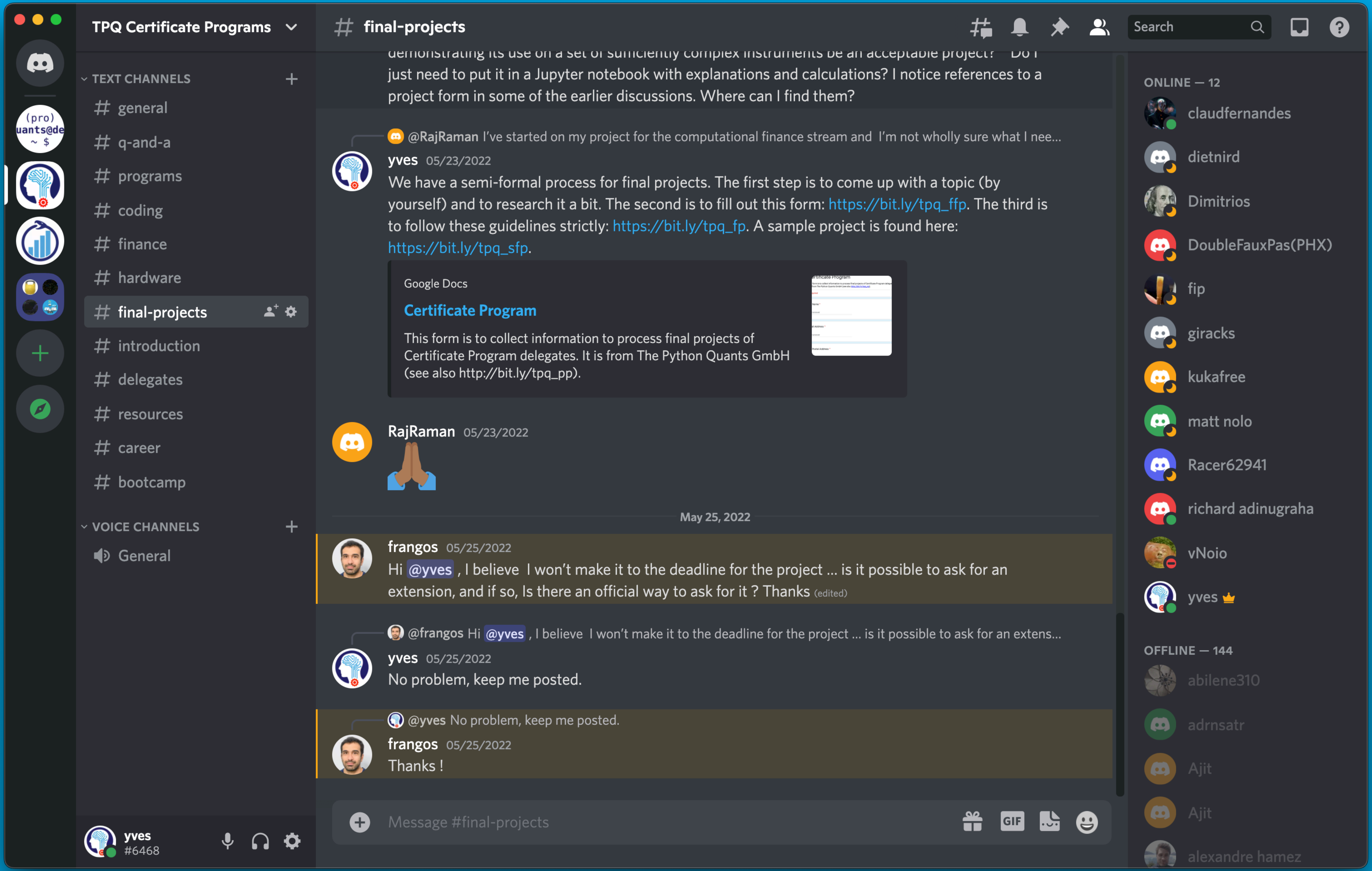
Implement the following game:

- a random number is drawn between 1 and 100
- the user is asked to provide a guess for the number
- it is checked whether the number is too low, too high or exactly right
- the user is provided with feedback about the result
- the game ends when the user has come up with the correct number
- the user is informed about the number of guesses needed

User Forum


Quant Platform
base.pqp.io/base/fo/get_html
MENU
Quant Platform
YVES
Quant Platform
All discussions around that platform
201 Topics
605 Replies
Certificate Program
Discussions around the certificate program
276 Topics
940 Replies
Finance with Python
Discussions around financial theory and Python basics
125 Topics
356 Replies
Python for Financial Data Science
All about interactive financial analytics and financial application building with Python
110 Topics
328 Replies
Python for Algorithmic Trading
Everything related to automated and algorithmic trading with Python
608 Topics
1952 Replies
Quant Platform
YVES
28
thematics for Economists alternative in electronic form
on Aug 13 2022 14:56
Last reply: Aug 23 2022 08:00
3 Replies
amp
on Aug 18 2022 22:39
Last reply: NA
NA Replies
New
to_pydatetime problem
by hahahahah on Jun 27 2022 05:14
Last reply: NA
NA Replies
New
Python dataframes: to "drag down" the non-NaN values in the dataframe column
by langfact on Apr 12 2022 17:30
Last reply: Apr 15 2022 11:59
2 Replies
New

Discord Server


TPQ Certificate Programs
TEXT CHANNELS
general
q-and-a
programs
coding
finance
hardware
final-projects
introduction
delegates
resources
career
bootcamp
VOICE CHANNELS
General
yves
#6468
final-projects
Search
demonstrating its use on a set of sufficiently complex instruments be an acceptable project? Do I just need to put it in a Jupyter notebook with explanations and calculations? I notice references to a project form in some of the earlier discussions. Where can I find them?
@RajRaman I've started on my project for the computational finance stream and I'm not wholly sure what I nee...
yves 05/23/2022
We have a semi-formal process for final projects. The first step is to come up with a topic (by yourself) and to research it a bit. The second is to fill out this form: https://bit.ly/tpq_ffp. The third is to follow these guidelines strictly: https://bit.ly/tpq_fp. A sample project is found here: https://bit.ly/tpq_sfp.
Google Docs
Certificate Program
This form is to collect information to process final projects of Certificate Program delegates. It is from The Python Quants GmbH (see also http://bit.ly/tpq_pp).
RajRaman 05/23/2022
May 25, 2022
frangos 05/25/2022
Hi @yves , I believe I won't make it to the deadline for the project ... is it possible to ask for an extension, and if so, Is there an official way to ask for it ? Thanks (edited)
@frangos Hi @yves , I believe I won't make it to the deadline for the project ... is it possible to ask for an extens...
yves 05/25/2022
No problem, keep me posted.
@yves No problem, keep me posted.
frangos 05/25/2022
Thanks !
Message #final-projects
GIF
ONLINE — 12
claudfernandes
dietnird
Dimitrios
DoubleFauxPas(PHX)
fip
giracks
kukafree
matt nolo
Racer62941
richard adinugraha
vNoio
yves
OFFLINE — 144
abilene310
adrnsatr
Ajit
Ajit
alexandre hamez

The Python Quants GmbH

Dr. Yves J. Hilpisch
+49 3212 112 9194
http://tpq.io | training@tpq.io
@dyjh

THE PYTHON QUANTS