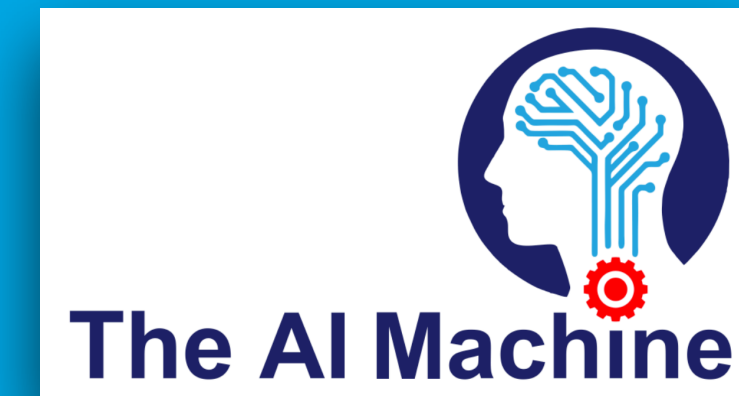


Certificate Program in Python for Finance —Algorithmic Trading, Computational Finance, and Asset Management

Introduction & Overview

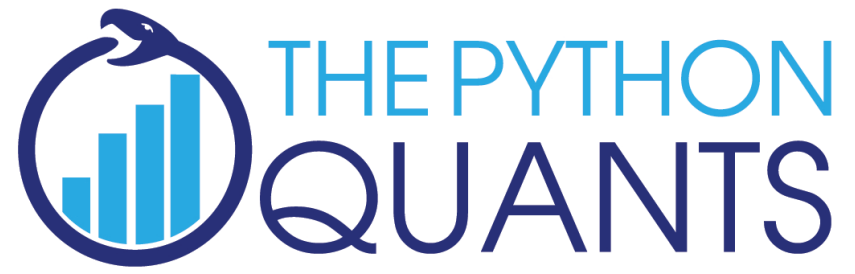
Dr. Yves J. Hilpisch



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



SERVICES

for financial institutions globally



EVENTS

for Python quants & algorithmic traders



TRAINING

about Python for finance
& algorithmic trading



CERTIFICATION

in cooperation with university



BOOKS

about Python and
finance



PLATFORM

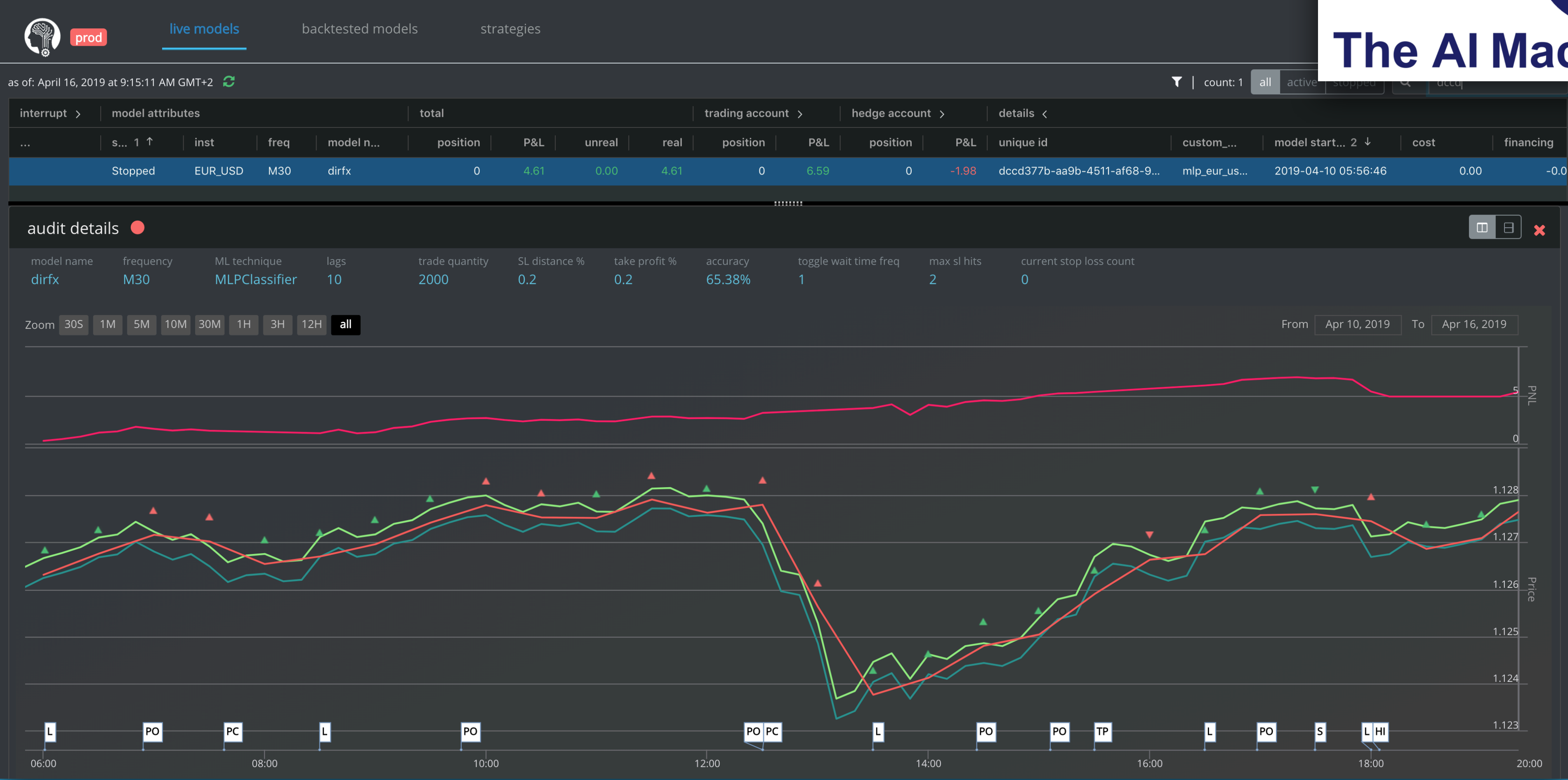
for browser-based
data analytics



OPEN SOURCE

Python library
for financial analytics





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.

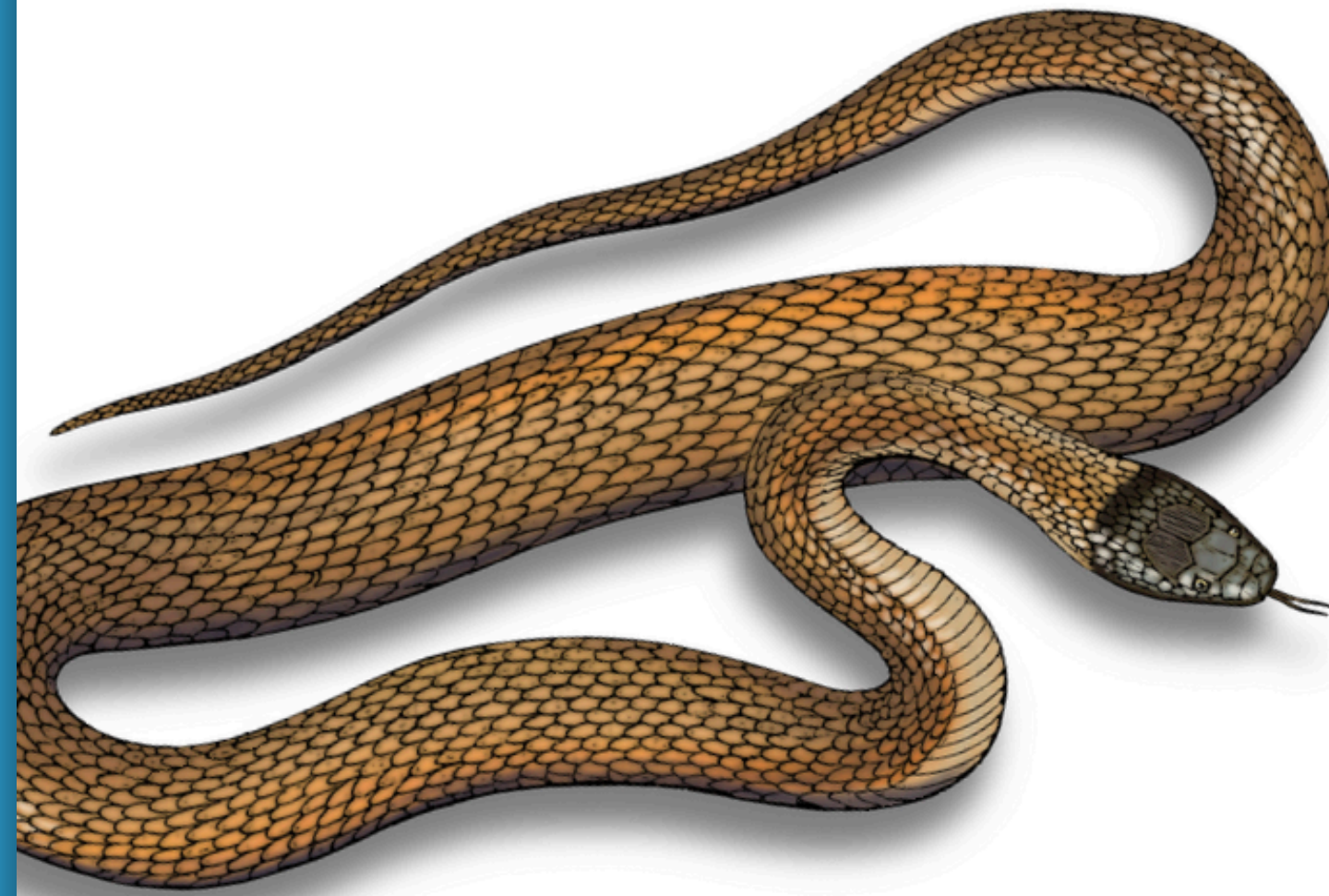
<http://hilpisch.com>

Financial Theory with Python

O'REILLY®

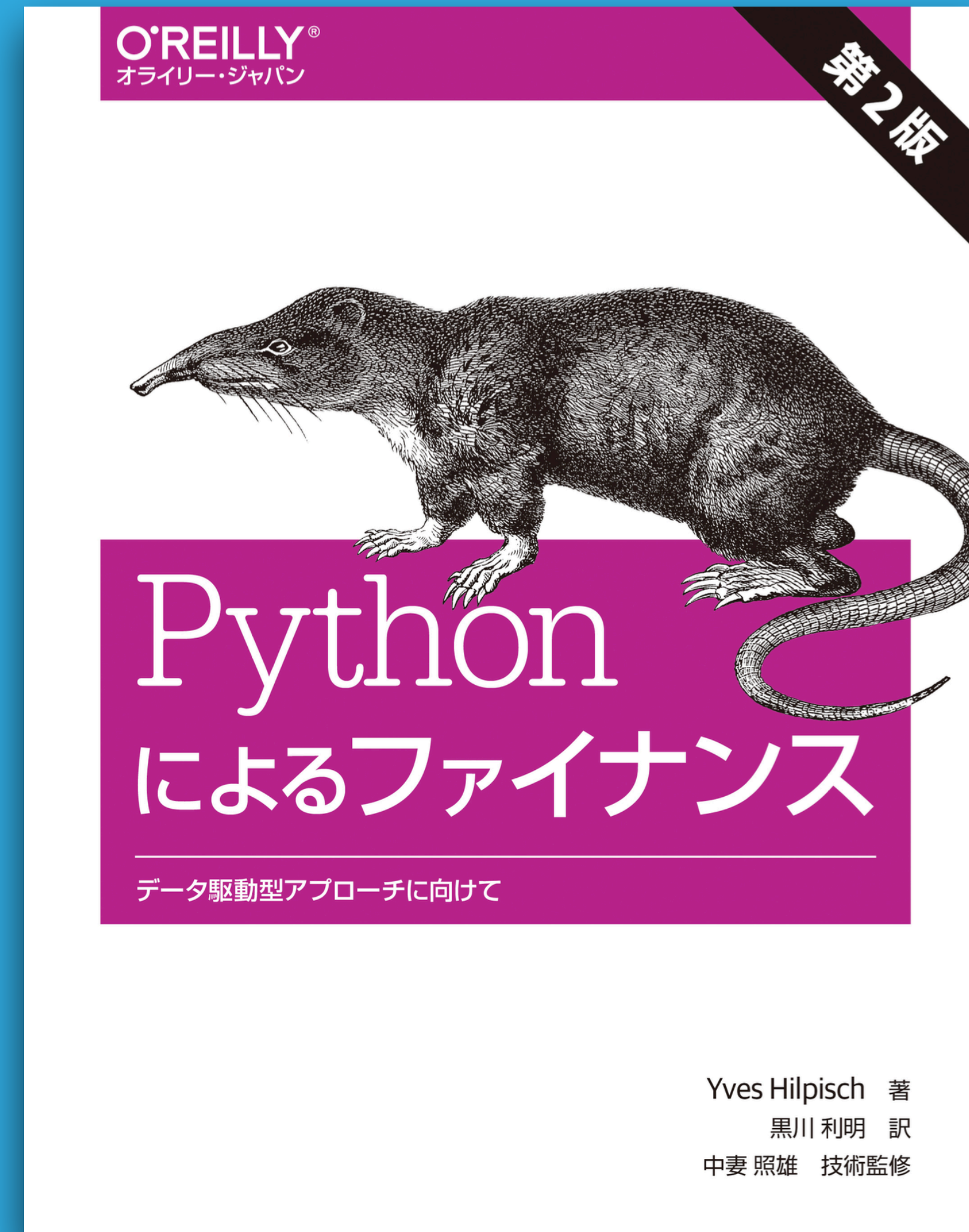
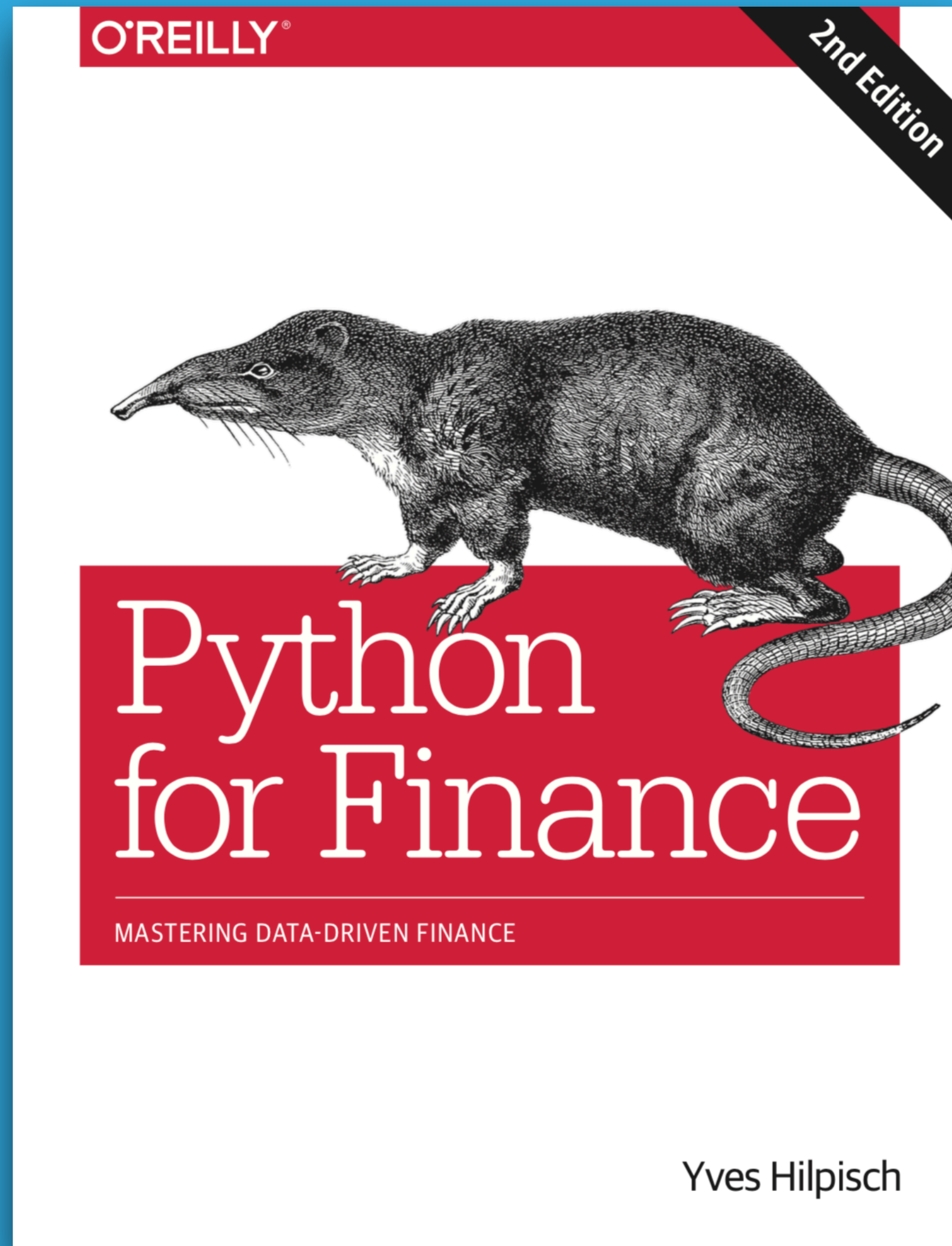
Financial Theory with Python

A Gentle Introduction



Yves Hilpisch

Python for Finance

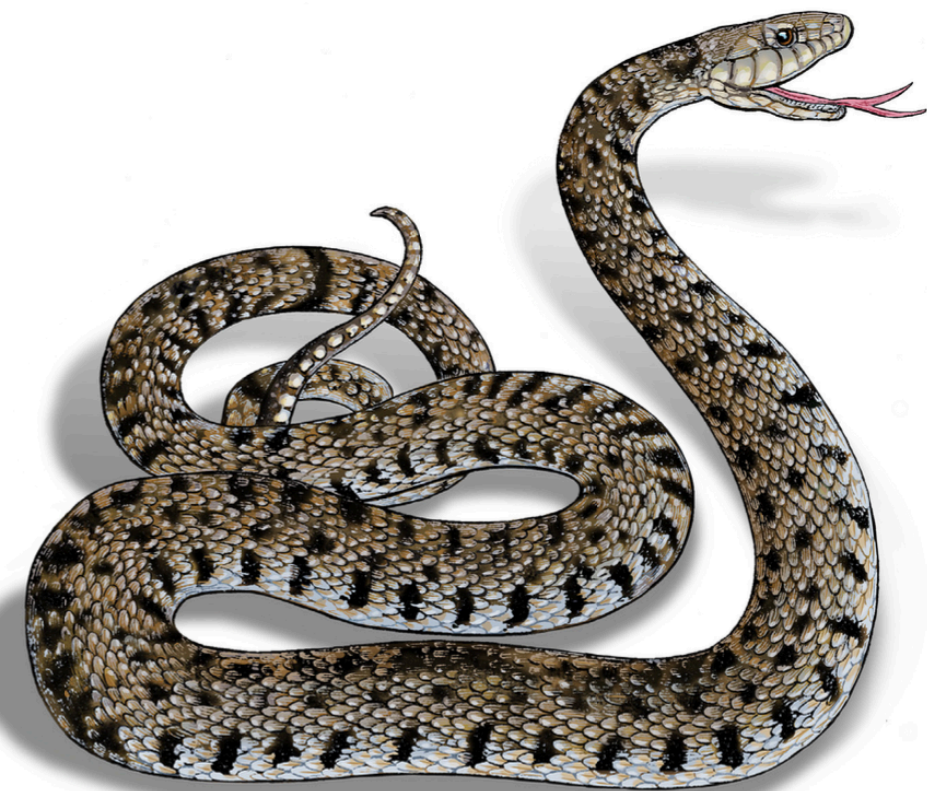


Python & AI for Finance & Trading

O'REILLY®

Python for Algorithmic Trading

From Idea to Cloud Deployment

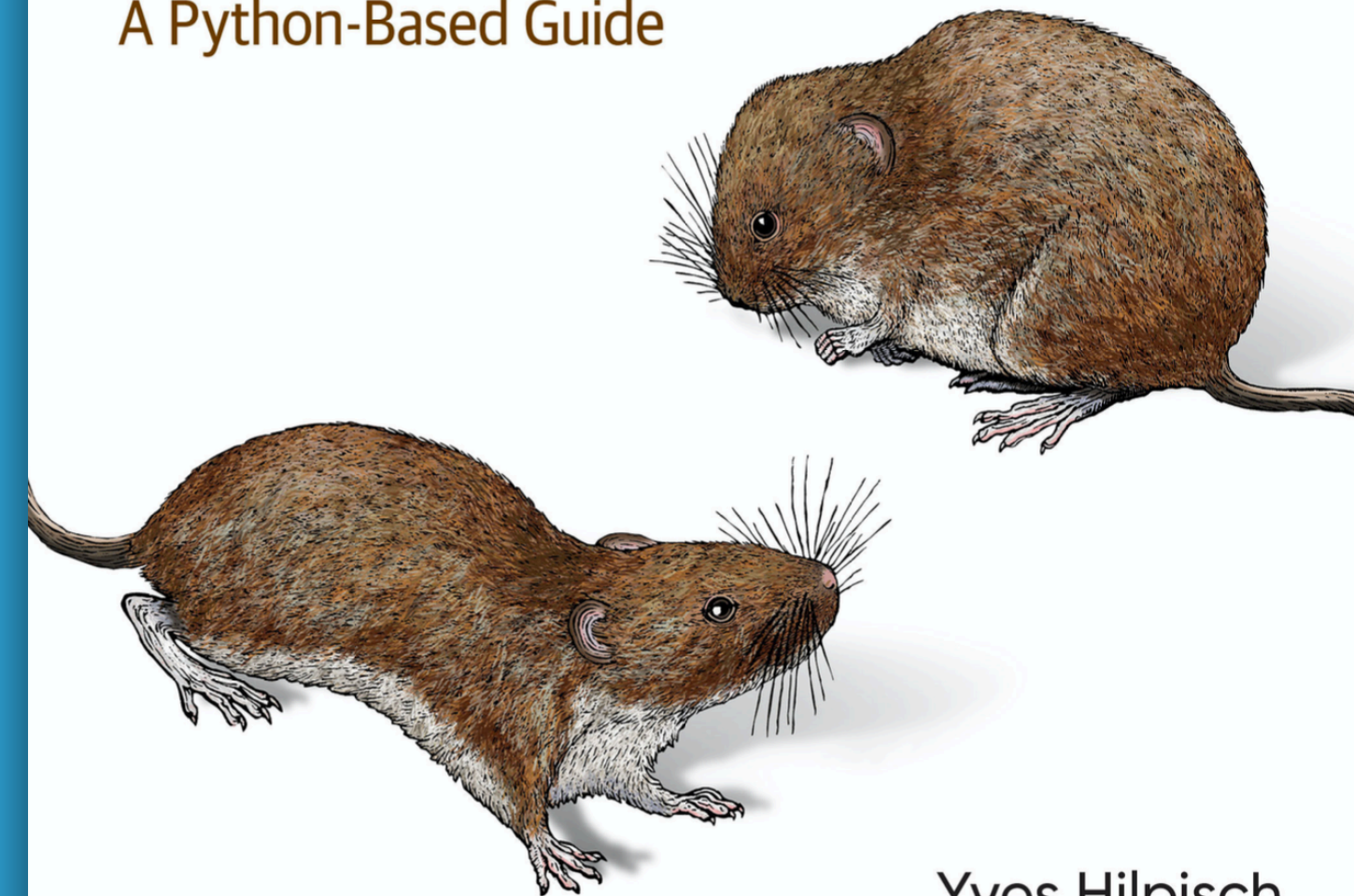


Yves Hilpisch

O'REILLY®

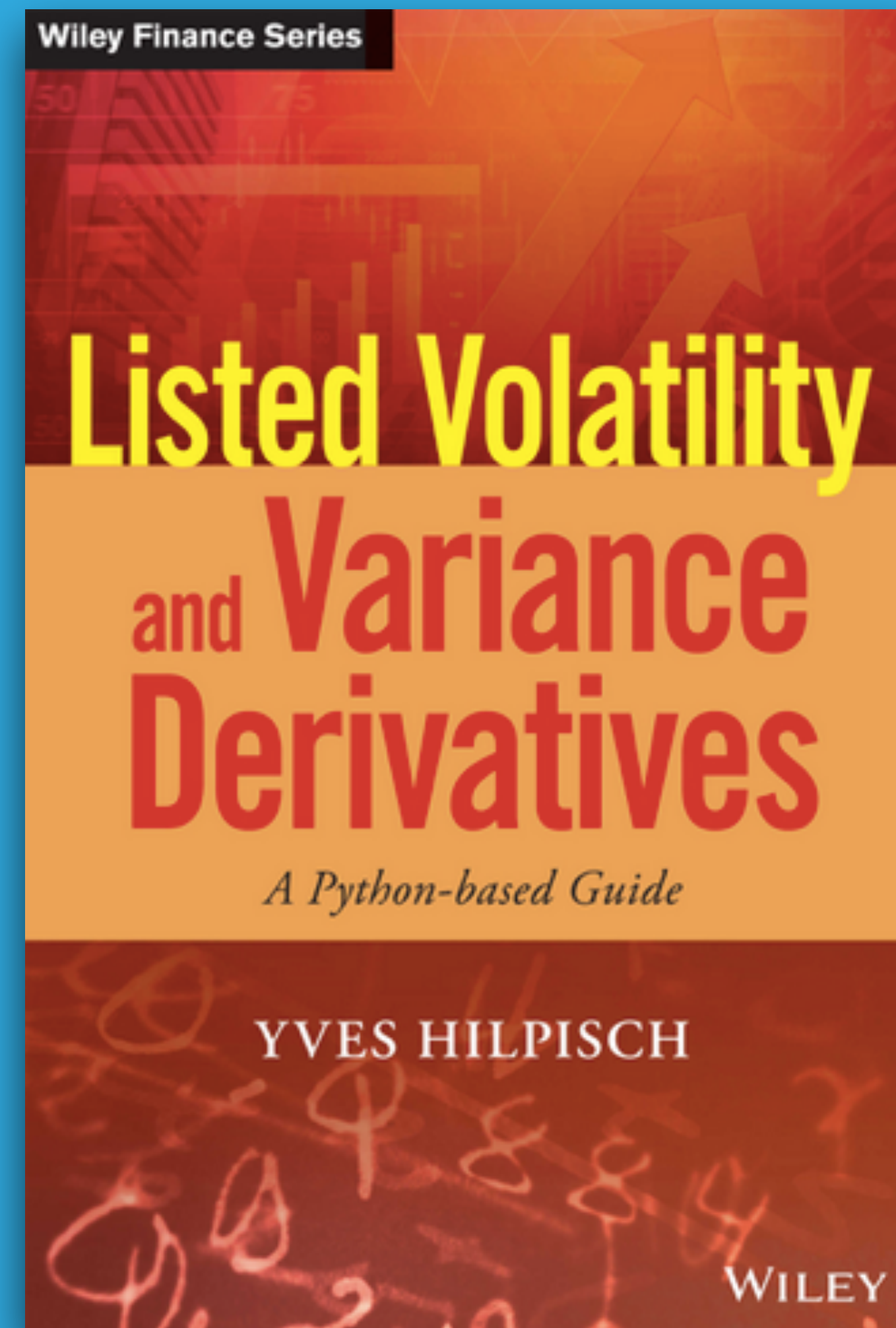
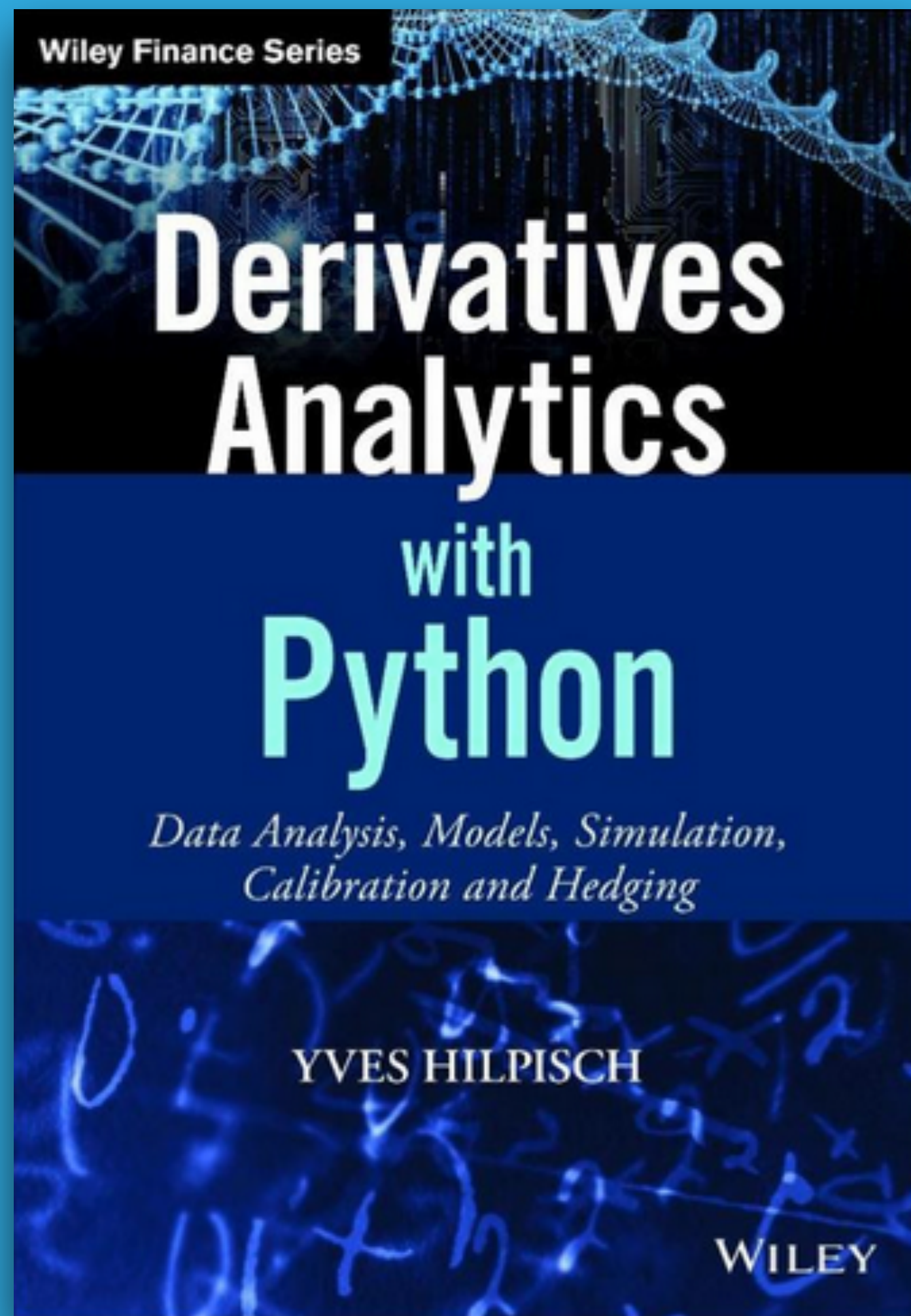
Artificial Intelligence in Finance

A Python-Based Guide



Yves Hilpisch

Quant Finance with Python



The Program

16 week program
(or self-paced)

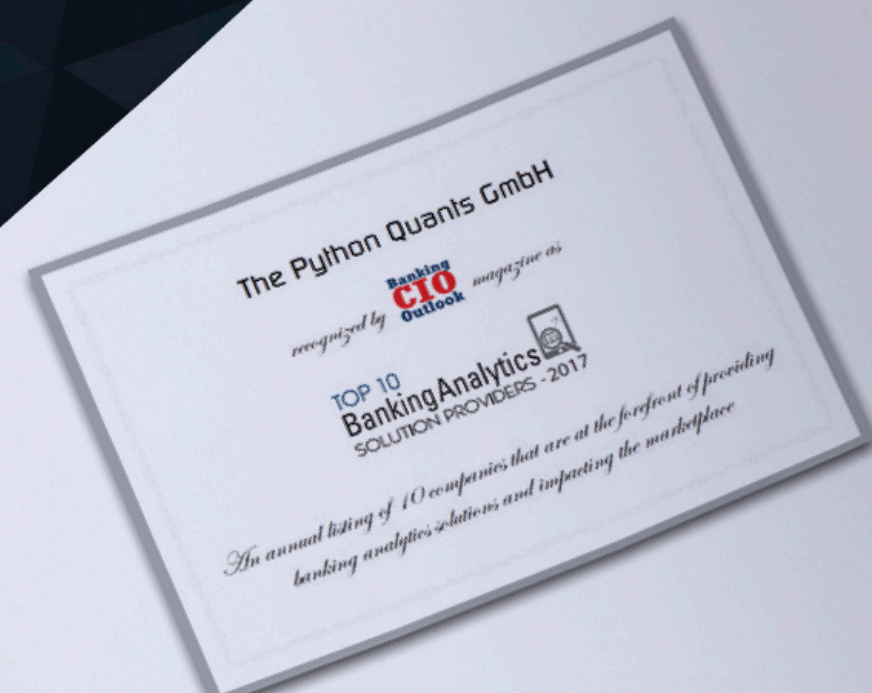
250+ hours
of instruction

PROGRAM DIRECTOR

Dr. Yves J. Hilpisch is founder and managing partner of The Python Quants (<http://tpq.io>), a group focusing on the use of open source technologies for financial data science, algorithmic trading and computational finance. He is the author of the books:

- Python for Finance (O'Reilly)
- Derivatives Analytics with Python (Wiley)
- Listed Volatility and Variance Derivatives (Wiley)

He has written the financial analytics library *DX Analytics* (<http://dx-analytics.com>) and organizes conferences and Meetup events about Python for finance and algorithmic trading in Frankfurt, London and New York. He has given keynote speeches at technology conferences in the United States, Europe and Asia.



UNIVERSITY CERTIFICATE IN PYTHON FOR ALGORITHMIC TRADING



The Python Quants GmbH
66333 Voelklingen
Germany
T/F +49 3212 112 91 94
<http://training.tpq.io>
training@tpq.io

April 2017

25,000+ lines
of code

250+ Jupyter
Notebooks

2,500+ pages
HTML/PDF

<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.

THE PYTHON QUANTS

Login to Quant Platform

yves

.....

Forgot password?

LOG IN

Quant Platform

Latest Videos

AT Bootcamp 06

This session is about cloud deployment, automation, logging and monitoring.

algorithmic_trading_online_bootcamp_06

AT Bootcamp 05

This session is about socket communication, streaming data and online algorithms.

algorithmic_trading_online_bootcamp_05

AT Bootcamp 04

Quant Platform

Portal

base.pqp.io/base/ju/get_iframe?path=AI%20in%20Finance/reinforcement_learning_02/04_rlearn.ipynb

File Edit View Run Kernel Tabs Settings Help

04_rlearn.ipynb

05_rlearn.ipynb

06_rlearn.ipynb

Python 3

THE AI MACHINE

Reinforcement Learning

OpenAI Gym

Dr Yves J Hilpisch | The AI Machine

<http://aimachine.io> | <http://twitter.com/dyjih>

CartPole

```
[ ]: import gym
import numpy as np
import pandas as pd
from pylab import plt
from IPython import display
plt.style.use('seaborn')
np.random.seed(100)
import warnings; warnings.simplefilter('ignore')
```

0 3 No Kernel | Idle

THE PYTHON QUANTS

Books

User Forum

Courses

Trainings

JupyterLab

Mathematics Basics

Mathematics Basics

Reviewing fundamental mathematics concepts based on simple Python code.

The image displays a web browser window and a Jupyter Notebook interface. The browser window shows the 'Quant Platform' with a 'Mathematics Basics' section. It includes a description: 'It serves both as a review as well as a preparation for the more involved mathematical applications in the different Certificate Programs.' Below this, there are two video thumbnails: 'Mathematics Basics 01' and 'Mathematics Basics 02'. Each thumbnail has a description and a link to a PDF file: https://certificate.tpq.io/mathematics_basics.pdf. The Jupyter Notebook interface shows a file explorer on the left with a table of files:

Name	Last Modified
01_math_b...	a month ago

The main area of the Jupyter Notebook displays the 'math Package' section, explaining that it provides access to mathematical functions defined by the C standard. It includes a list of code snippets:

```
[1]: import math
[2]: # dir(math)
[3]: math.pi # number pi
[3]: 3.141592653589793
[4]: math.e # Euler number
[4]: 2.718281828459045
[5]: 2 ** 0.5
[5]: 1.4142135623730951
[6]: math.sqrt(2)
```

The status bar at the bottom indicates 'Mode: Command', 'Ln 1, Col 1', and '01_math_basics.ipynb'.

Python for Fiance Basics

Python for Finance Basics

Introducing fundamental Python programming idioms and concepts.

The image displays a collage of three browser windows and a Jupyter Notebook interface, all related to the 'Python for Finance Basics' course.

The background window shows the 'Quant Platform' website. It features three course cards:

- Python for Finance Basics 01**: This is the first session about PFF Basics. It covers topics related to Python infrastructure, such as Docker, Miniconda, conda, numpy, pandas, matplotlib, IPython, and JupyterLab. You find the slides under https://certificate.tpq.io/pff_basics.pdf. Files: 01_pff_basics.ipynb, pff_01.py, pff_01.txt.
- Python for Finance Basics 02**: This is the second session about PFF Basics. It covers topics related to Python infrastructure, apt install and pip install. You find the slides under https://certificate.tpq.io/pff_basics.pdf. Files: 02_pff_basics.ipynb, jupyter_shortcuts.txt, run_docker.sh.
- Python for Finance Basics 03**: This is the third session about PFF Basics. It covers topics related to Python infrastructure, apt install and pip install. You find the slides under https://certificate.tpq.io/pff_basics.pdf. Files: 02_pff_basics.ipynb.

The foreground window shows a Jupyter Notebook titled '02_pff_basics.ipynb'. The code in the notebook is as follows:

```
bool objects ¶

[ ]: True

[ ]: False

[ ]: type(True)

[ ]: 4 > 3 # Think of the question: "Is 4 greater than 3?"

[ ]: 3 > 4

[ ]: a = 2.5 + 2 ** 0.5

[ ]: b = 3.75

[ ]: a > b

One equal sign ("=") is the assignment operator. Two equal signs ("==") are the comparison operator ("is equal?")

[ ]: 3 == 4

[ ]: a == b
```

Crypto Basics

Crypto Basics

Mastering the basic building blocks of the crypto space.

Quant Platform

base.pqp.io/base/po/get_training_html?tr_id=193

MENU


Quant Platform

YVES

Crypto Basics

This class covers the basics of hashing, encryption, blockchain, mining, cryptocurrencies and smart contracts.


Crypto Basics 01




crypto_basics_01


This is the first session about Crypto Basics and it covers data about Bitcoin and basics of hashing (MD5).

You find the slides under https://certificate.tpq.io/crypto_basics.pdf.

 01_crypto_basics.ipynb

 02_crypto_basics.ipynb


Crypto Basics 02




crypto_basics_02


This is the second session about Crypto Basics and it covers password cracking via brute force hashing as well as via Hashcat (<https://hashcat.net/hashcat>) mask attacks.

You find the slides under https://certificate.tpq.io/crypto_basics.pdf.

 03_crypto_basics.ipynb

 04_crypto_basics.ipynb


Crypto Basics 03




crypto_basics_03

This is the third session about Crypto Basics and it covers encryption based on simple approaches (Caesar Cipher, Transposition Cipher) and secure approaches (AES, Advanced Encryption Standard).

You find the slides under https://certificate.tpq.io/crypto_basics.pdf.

 05_crypto_basics.ipynb

 06_crypto_basics.ipynb

Quant Platform


Portal


base.pqp.io/base/ju/get_iframe?path=Crypto%20Basics/crypto_basics_03/05_crypto_basics.ipynb

File Edit View Run Kernel Tabs Settings Help

/ ... / Crypto Basics / crypto_basics_03

Name

 05_crypto_basics.ip...

 06_crypto_basics.ip...

Launcher

05_crypto_basics.ipynb

06_crypto_basics.ipynb

Python 3

AES Encryption

From Wikipedia (https://en.wikipedia.org/wiki/Advanced_Encryption_Standard):

The Advanced Encryption Standard (AES), also known by its original name Rijndael (Dutch pronunciation: [ˈreɪndɑːl]), is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001.

AES is a variant of the Rijndael block cipher developed by two Belgian cryptographers, Vincent Rijmen and Joan Daemen, who submitted a proposal to NIST during the AES selection process. Rijndael is a family of ciphers with different key and block sizes. For AES, NIST selected three members of the Rijndael family, each with a block size of 128 bits, but three different key lengths: 128, 192 and 256 bits.

The following requires the `pycryptodome` package:

```
conda install pycryptodome
```

File on Disk

```
[ ]: fn = 'example.cfg'

[ ]: with open(fn, 'w') as f:
    f.write(cfg)

[ ]: !cat $fn
```

0 \$ 2 Python 3 | Idle

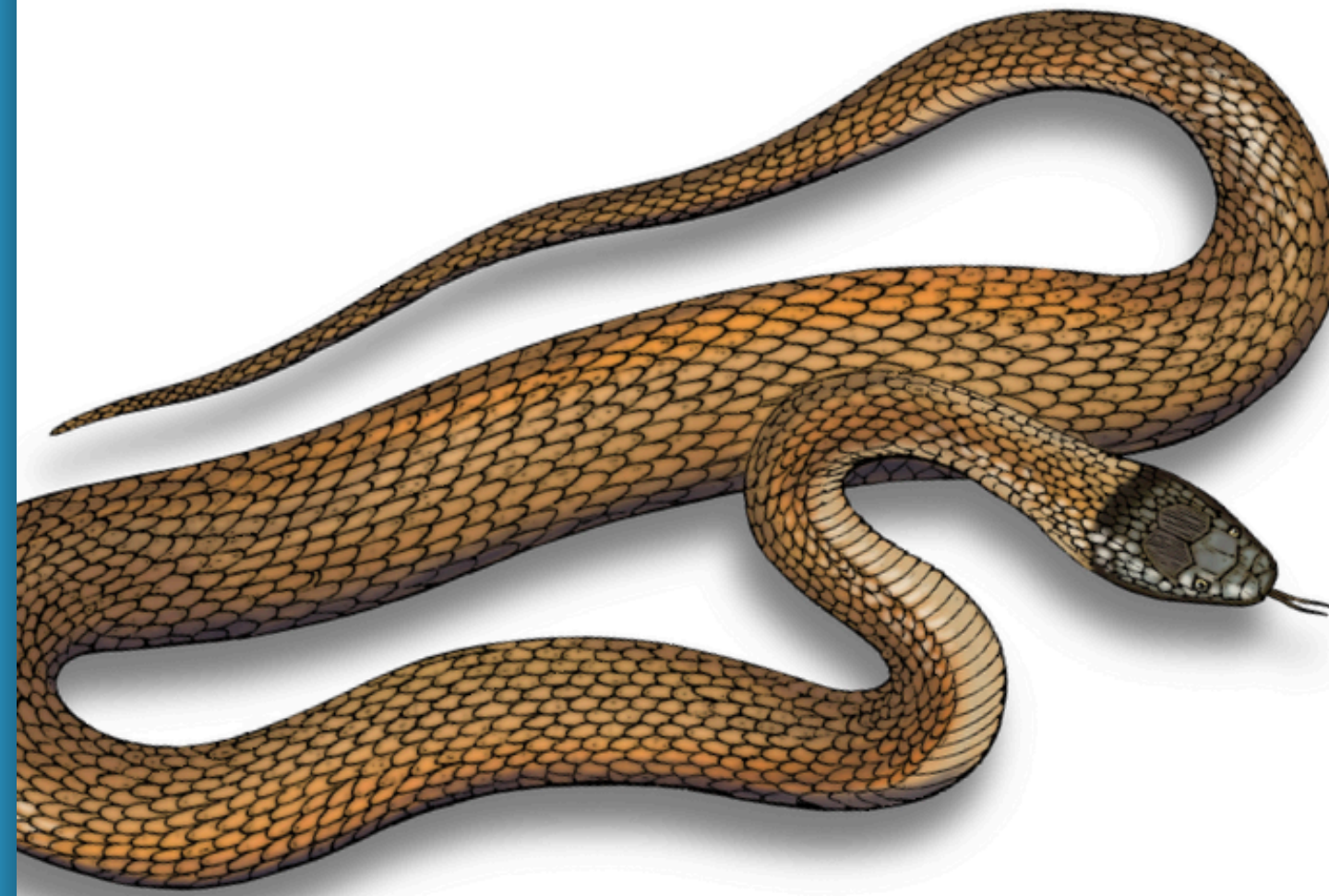
Mode: Command Ln 1, Col 1 06_crypto_basics.ipynb

Finance with Python

O'REILLY®

Financial Theory with Python

A Gentle Introduction

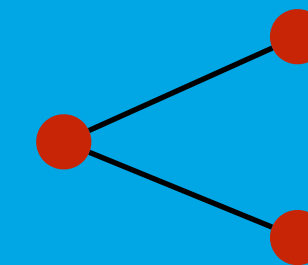


Yves Hilpisch

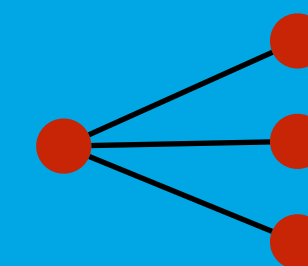
Finance with Python

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

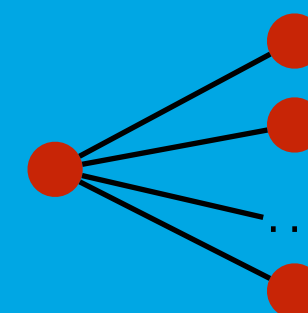
static two state economy
fundamental theorem of asset pricing



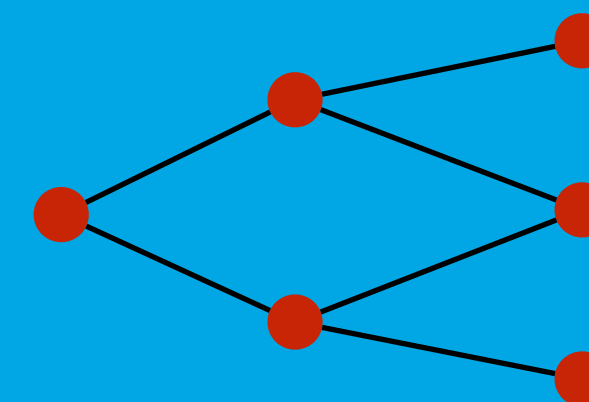
static three state economy
dealing with market incompleteness



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.

Table of Contents

Copyright

Preface

Why this Course?

Target Audience

Overview of the Course

Bibliography

1. Finance and Python

1.1. Introduction

1.2. A Brief History of Finance

1.3. A Four Languages World

1.4. The Approach of this Course

1.5. Getting Started with Python

1.6. Conclusions

1.7. Further Resources

2. Two State Economy

2.1. Introduction

2.2. Economy

2.3. Real Assets

2.4. Agents

2.5. Time

2.6. Money

2.7. Cash Flow

2.8. Return

2.9. Interest

2.10. Present Value

2.11. Net Present Value

2.12. Uncertainty

2. Two State Economy

“A fundamental question in finance is how the risk of an investment should affect its expected return.

— André Perold (2004)

2.1. Introduction

The analysis in this chapter is based on the most simple financial model that is still rich enough to introduce many important notions and concepts of finance: an economy with two relevant points in time and two uncertain future states only. It also allows to present some important results in the field, like the *Fundamental Theorems of Asset Pricing*.

The simple model chosen is a means to simplify the formal introduction of the sometimes rather abstract mathematical concepts and financial ideas by avoiding as many technicalities as possible. Once the simple model is understood, the transfer to more realistic financial models usually proves seamless.

This chapter covers mainly the following central topics from finance, mathematics

finance	mathematics
time	natural numbers \mathbb{N}
currency (money)	real numbers \mathbb{R}
cash flow	tuple
return, interest	real numbers \mathbb{R}
(net) present value	function

Quant Platform

Finance with Python

This training section contains the videos related to the Finance with Python course material.

Finance with Python 01

algo_cert_session_finpy_01.mp4

This module is about the topics covered in the Finance with Python course material.

Link to the slides

http://hilpisch.com/algo_cert_finance.pdf

01_finance_w_python_demo.ipynb

02_finance_w_python_demo.ipynb

03_finance_w_python_live.ipynb

algo_cert_finance.pdf

Finance with Python 02

algo_cert_session_finpy_02_cut.mp4

This module is about the topics covered in the Finance with Python course material.

Link to the slides

http://hilpisch.com/algo_cert_finance.pdf

04_finance_w_python_live.ipynb

neural_net_reg.pdf

Finance with Python 03

algo_cert_session_finpy_03.mp4

This module is about the topics covered in the Finance with Python course material.

Link to the slides

http://hilpisch.com/algo_cert_finance_new.pdf

05_finance_w_python_live.ipynb

base.pqp.io/course/finpy/finpy.html#two_state_economy

base.pqp.io/base/po/get_training_html?tr_id=102

base.pqp.io/base/fu/get_iframe

finpy_05.ipynb

Python 3

THE PYTHON QUANTS

Finance with Python

Chapter 05 — Static Economy

Numerical Examples

[1]: import numpy as np

[2]: np.random.seed(100)

[3]: I = 1000

[4]: S = np.random.normal(loc=100, scale=20, size=I)

[5]: S[:14].round(2)

[5]: array([65. , 106.85, 123.06, 94.95, 119.63, 110.28, 104.42, 78.6 , 96.21, 105.1 , 90.84, 108.7 , 88.33, 116.34])

[6]: S.mean()

[6]: 99.66455685312181

Mode: Command

Ln 1, Col 1

finpy_05.ipynb

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

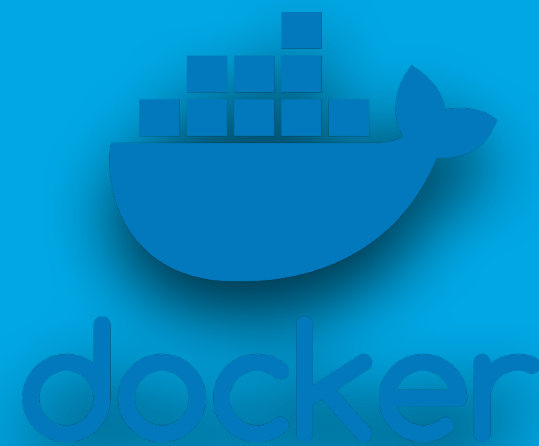
Tools and Skills

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

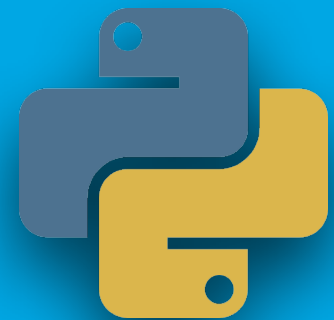


cloud instances (“virtual servers”) from a low as
5 USD per month, to really big instances

<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.

base.pqp.io/base/po/get_training_html?tr_id=113

Quant Platform YVES

Tools and Skills Special 01



tools_skills_special_01.mp4


This module is a special about IPython.

The slides are found under https://hilpisch.com/tools_skills_special.pdf

The Gist with further resources is found under http://bit.ly/tools_skills_special

No files attached

Tools and Skills Special 02



tools_skills_special_02.mp4


This module is a special about xonsh.

The slides are found under https://hilpisch.com/tools_skills_special.pdf

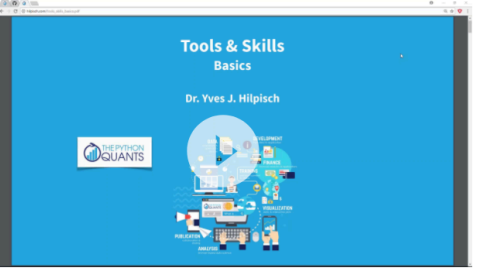
The Gist with further resources is found under http://bit.ly/tools_skills_special

No files attached

Tools and Skills 01



tools_skills_01_mac_linux.mp4



tools_skills_01_windows.mp4

This module covers the basics of setting up a Python environment.

The slides are found under http://hilpisch.com/tools_skills_basics.pdf

The Gist with selected resources is found under http://bit.ly/tools_skills_basics

No files attached

gist.github.com/yhilpisch/bda2479093216b299e59cf8c41bfa3e7

00_overview.md Raw

Python Tools & Skills


Basics

Dr. Yves J. Hilpisch | The Python Quants GmbH

Certificate Program, May 2018

(short link to this Gist: http://bit.ly/tools_skills_basics)

(short link to Gist for cloud files: http://bit.ly/tools_skills_cloud)



Slides

Python for Financial Data Science

O'REILLY®

2nd Edition



Python for Finance

MASTERING DATA-DRIVEN FINANCE

Yves Hilpisch

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

中妻 照雄 技術監修

黒川 利明 訳

Yves Hilpisch 著

Python for Financial Data Science

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

BASICS

Python Data
Types & Structures

Numerical Computing
with NumPy

Data Analysis
with pandas

Object Oriented
Programming

DATA SCIENCE

Visualization

Financial
Time Series

Input-Output
Operations

Performance
Python

MATHEMATICS

Mathematical
Tools

Stochastics

Statistics &
Machine Learning

Special:
Dates & Times

Python for Financial Data Science

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

Table of Contents

Python and Finance

1. Why Python for Finance

1.1. Python Programming Language

1.2. Technology in Finance

1.3. Python for Finance

1.4. Data-Driven and AI-First Finance

1.5. Conclusions

1.6. Further Resources

2. Python Infrastructure

2.1. Introduction

2.2. Conda as a Package Manager

2.3. Conda as a Virtual Environment Manager

2.4. Using Docker Containers

2.5. Using Cloud Instances

2.6. Conclusions

2.7. Further Resources

Mastering the Basics

3. Data Types and Structures

3.1. Introduction

3.2. Basic Data Types

3.3. Basic Data Structures

3.4. Conclusions

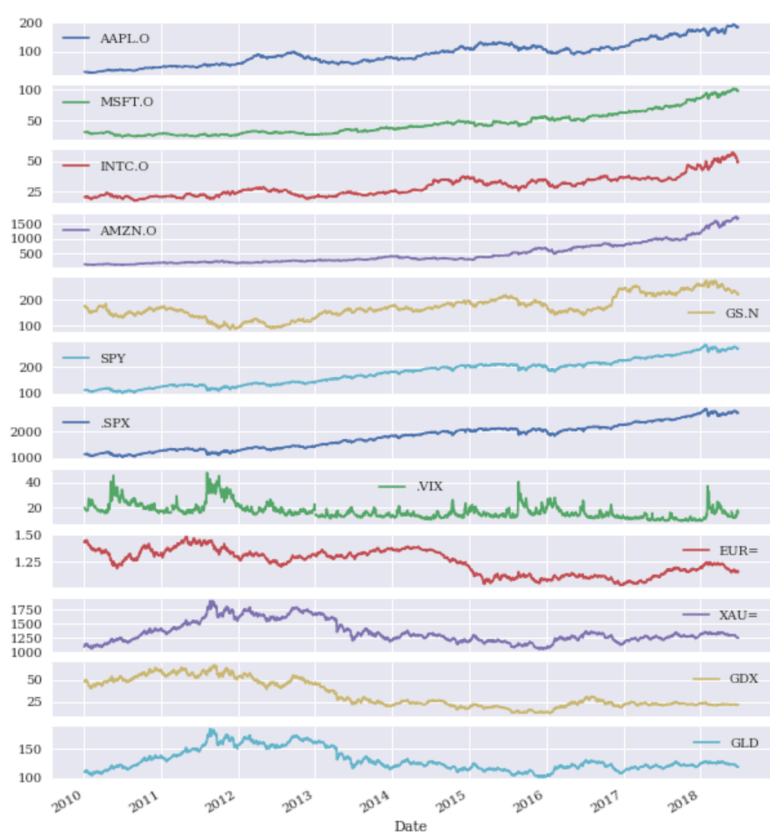
3.5. Further Resources

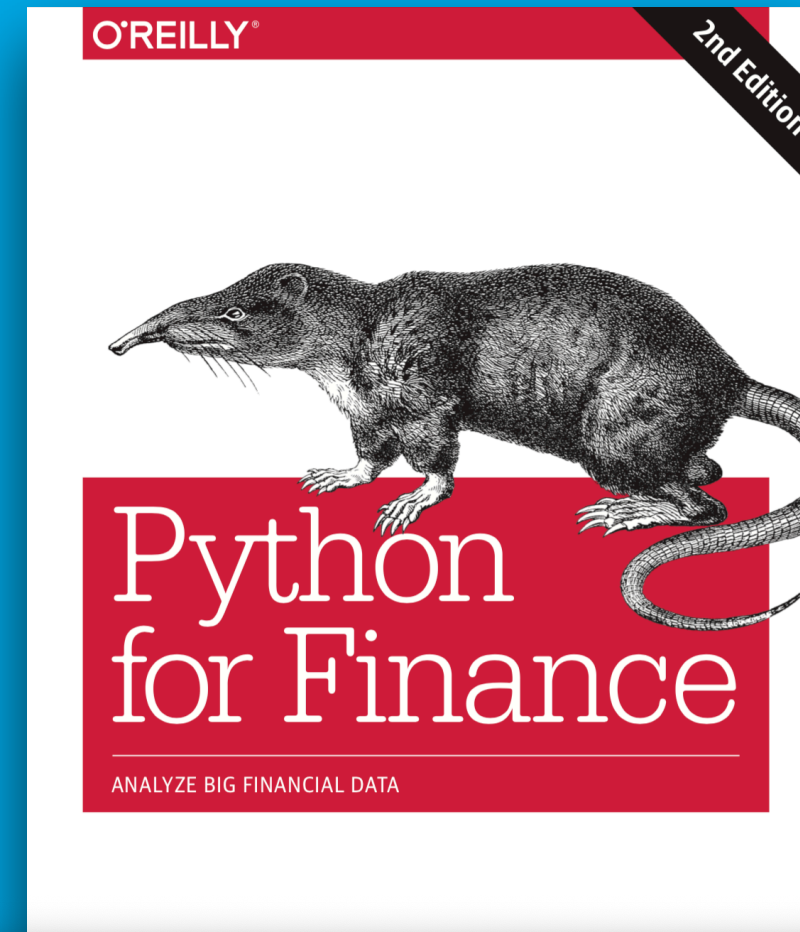
4. Numerical Computing with NumPy

4.1. Introduction

4.2. Arrays of Data

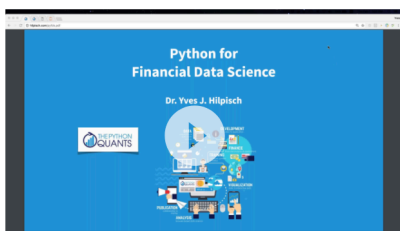
4.3. Regular NumPy Arrays





Quant Platform


Data Types and Structures 01





pyfds_session_01.mp4

This introductory session gives an overview and covers basic data types and structures in Python.

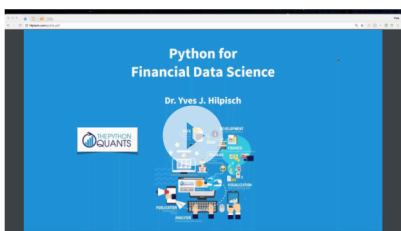
The slides are found under <http://hilpisch.com/pyfds.pdf>

 01_why_python.ipynb

 02_python_infrastructure.ipynb


 03_data_structures.ipynb


Data Types and Structures 02




pyfds_session_02.mp4


This session deepens the journey through data types and structures.

 03_data_structures.ipynb

 03_data_structures_case.ipynb


 Titanic.csv

Numerical Computing Numpy

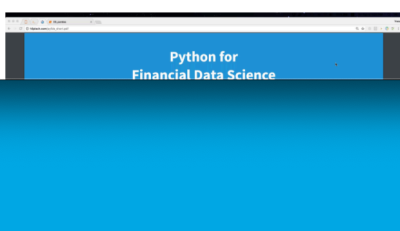


pyfds_session_03.mp4

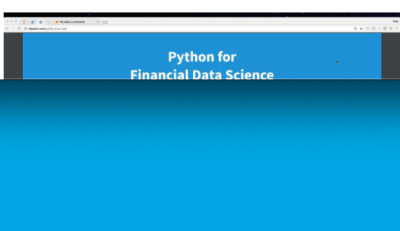
This session is about Numerical Computing with NumPy.

 04_numpy.ipynb

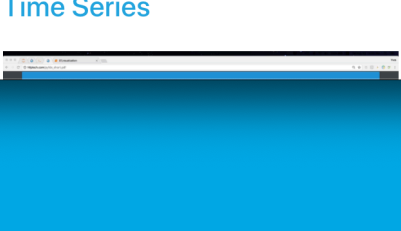
Data Analysis with pandas



Object Oriented Programming



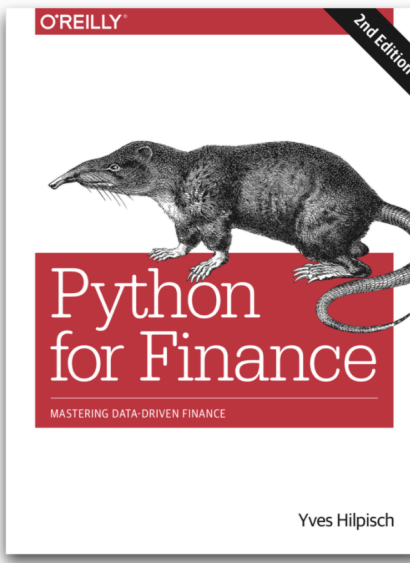
Visualization and Financial Time Series



Python for Financial Data Science

Dr Yves J Hilpisch | The Python Quants GmbH

<http://tpq.io> | training@tpq.io



Yves Hilpisch

01_why_python.ipynb

02_python_infrastructure.i...

03_data_structures_case.i...

03_data_structures.ipynb

04_numpy.ipynb

05_pandas_case.ipynb

05_pandas.ipynb

06_object_orientation_cas...

06_object_orientation.ipynb

07visualization.ipynb

08_financial_time_series.ip...

09_input_output.ipynb

10_performance_python.ip...

11_math_tools.ipynb

12_stochastics.ipynb

13_a_statistics.ipynb

13_b_statistics.ipynb

13_c_machine_learning.ipy...

a_dates_times.ipynb

array.apy

bsm_functions.py

bsm_mcs_euro.py

Mode: Command Ln 1, Col 1 08_financial_time_series.ipynb

Financial Packages

Financial Packages

Efficient and powerful Python packages for finance.

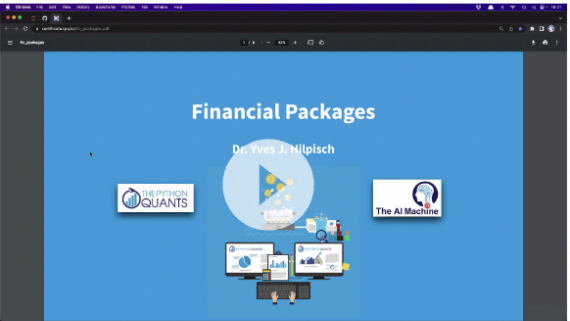
Quant Platform

base.pqp.io/base/po/get_training_html?tr_id=189

MENU

YVES


Financial Packages 04



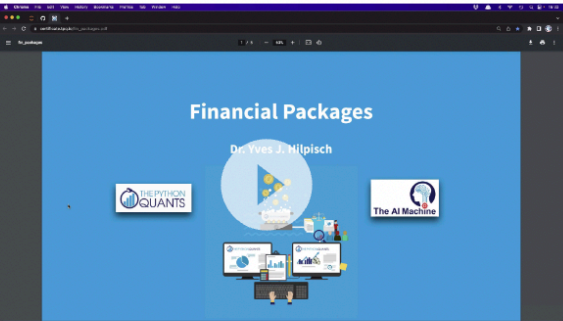
fin_packages_04

This is the fourth module about Financial Packages in the Python ecosystem. It covers the backtesting package vectorbt (to backtest algorithmic trading strategies).

You find the slides under https://certificate.tpq.io/fin_packages.pdf.

 05_fin_packages.ipynb



Financial Packages 03



fin_packages_03

This is the the third module about Financial Packages in the Python ecosystem. It covers the portfolio/strategy analytics & reporting package quantstats.

You find the slides under https://certificate.tpq.io/fin_packages.pdf.

 04_fin_packages.ipynb
 AAPL.O-tearsheet-bench.html

Quant Platform

base.pqp.io/base/ju/get_iframe?path=Certificate%20Live%20Sessions/financial_packages_03/04_fin_packages.ipynb

File Edit View Run Kernel Tabs Settings Help

Launcher

04_fin_packages.ipynb

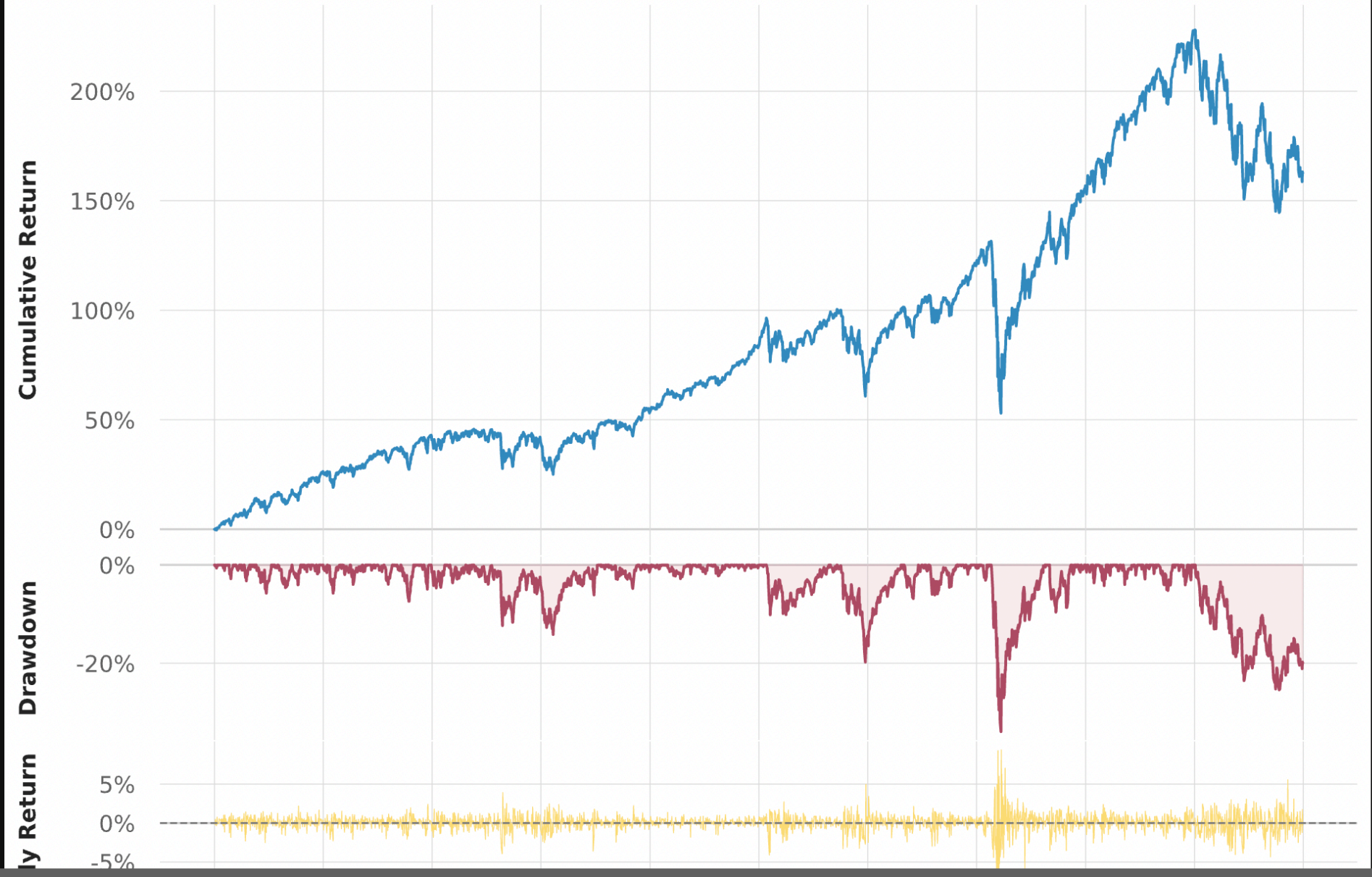
Python 3

Cumulative Return

Drawdown

Volatility Return

1 Jan '13 - 30 Dec '22 ; Sharpe: 0.64

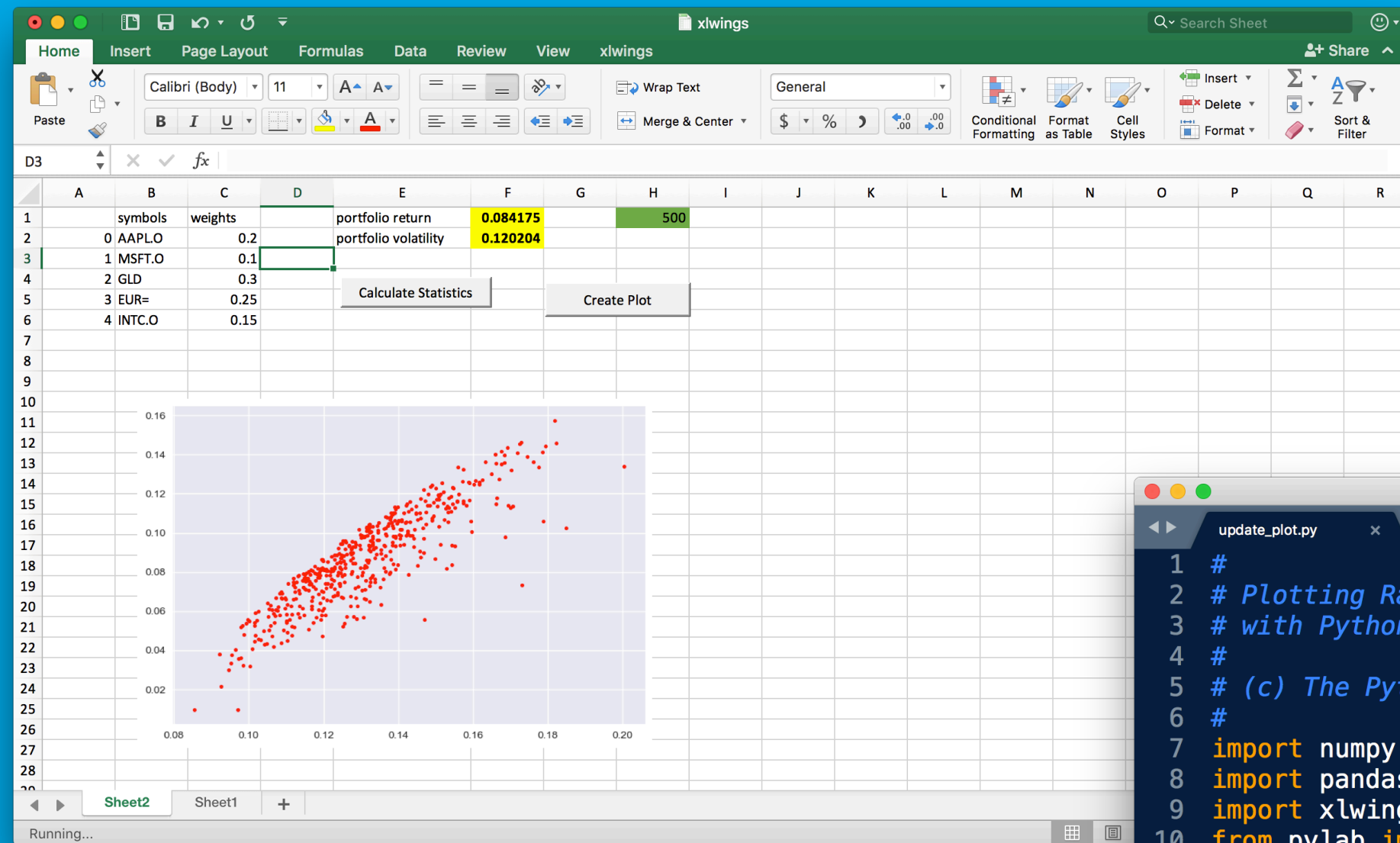


0 1 Python 3 | Idle Mode: Command Ln 1, Col 1 04_fin_packages.ipynb

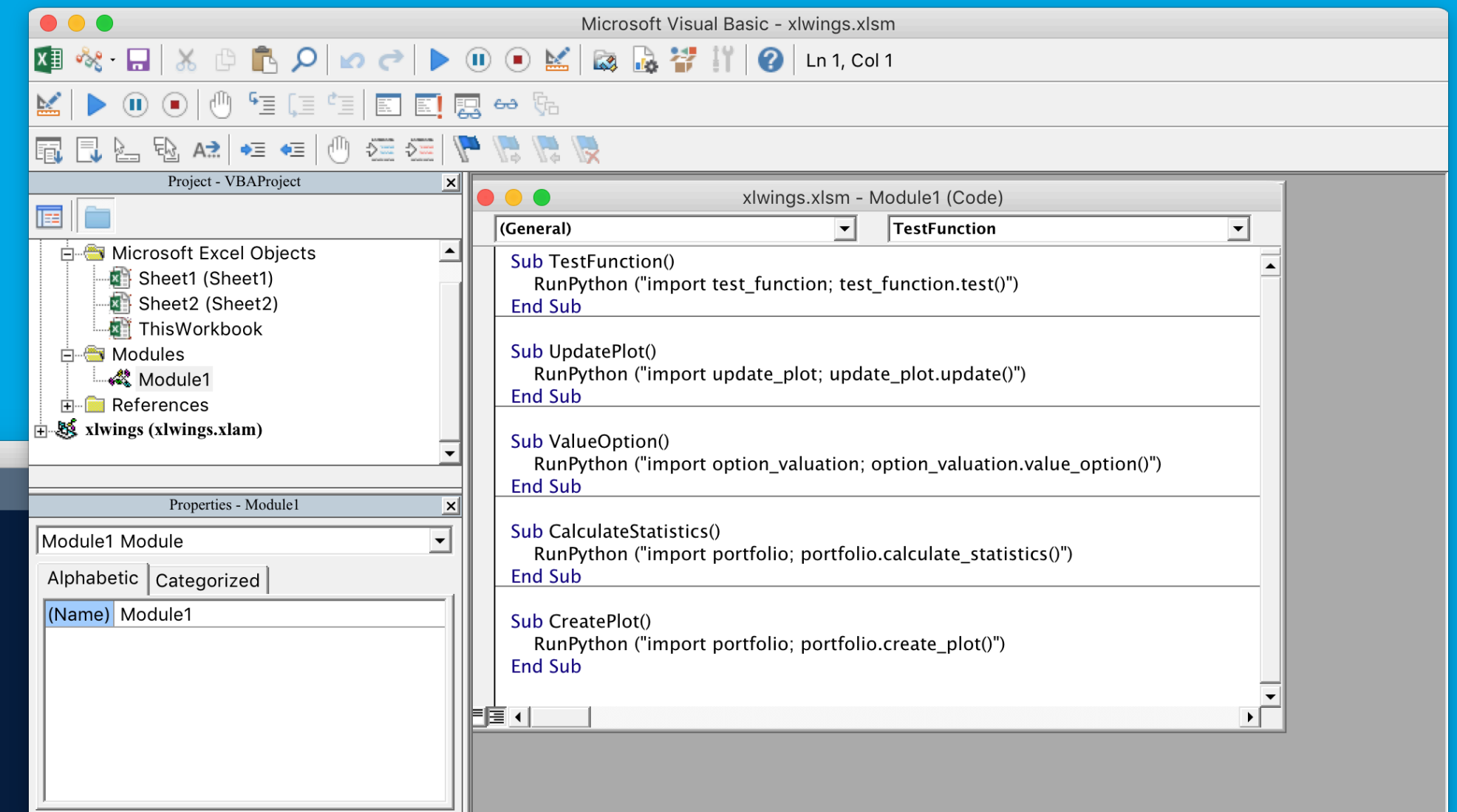
Python for Excel

Python for Excel

Combining Excel with the analytics power of Python.



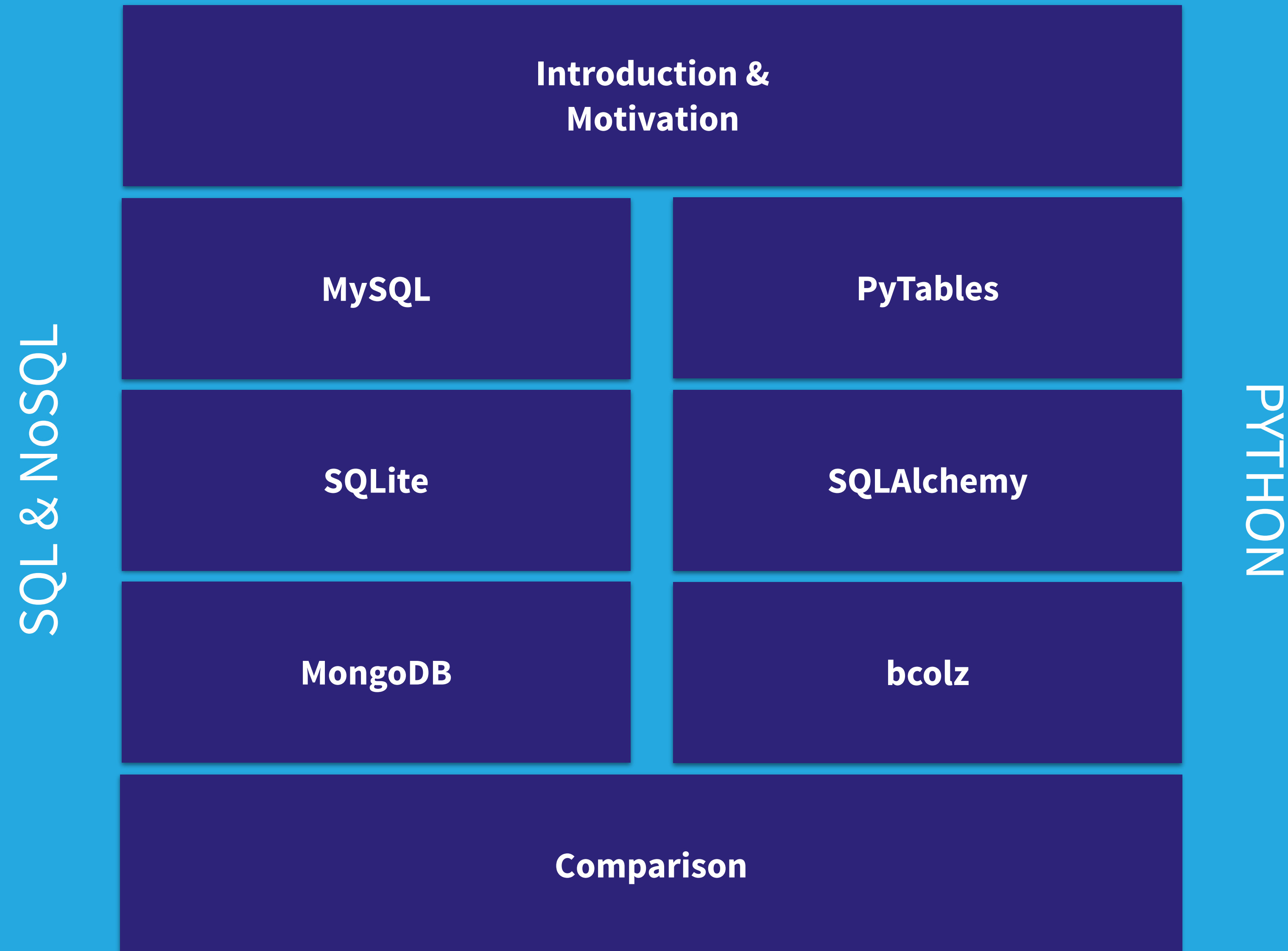
```
update_plot.py
1 #
2 # Plotting Random Walks
3 # with Python for Excel
4 #
5 # (c) The Python Quants GmbH
6 #
7 import numpy as np
8 import pandas as pd
9 import xlwings as xw
10 from pylab import plt
11
12 a = np.random.standard_normal((50, 3)).cumsum(axis=0)
13 df = pd.DataFrame(a, columns=list('abc'),
14                  index=pd.date_range('2019-1-1', periods=50, freq='B'))
15
16 plt.style.use('seaborn')
17
18
19 def update():
20     wb = xw.Book.caller()
21     sht = wb.sheets[1]
22     fig, ax = plt.subplots()
23     df.plot(ax=ax)
24     sht.pictures.add(fig, name='pd_plot', update=True)
25
```



Python for Databases

Python for Databases

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



Python for Databases

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

The image displays a collage of three browser windows and a terminal window, illustrating the resources and setup for the 'Python for Databases' training.

Quant Platform (Left Window): The website shows three video modules:

- Python for Databases 04:** Covers HDF5 and PyTables. Slides are found under <http://hilpisch.com/pydb.pdf>. The Gist with selected resources is found under http://bit.ly/py_db. No files attached.
- Python for Databases 05:** Covers SQLAlchemy and its SQL expression language as well as ORM model. Slides are found under <http://hilpisch.com/pydb.pdf>. The Gist with selected resources is found under http://bit.ly/py_db. No files attached.
- Python for Databases 06:** This module is about bcolz, a fast columnar data store, and compares all database technologies covered so far. Slides are found under <http://hilpisch.com/pydb.pdf>. The Gist with selected resources is found under http://bit.ly/py_db. No files attached.

Python for Databases (Middle Window): The Gist page shows the title 'Python for Databases' and the author 'Dr. Yves J. Hilpisch | The Python Quants GmbH'. It mentions 'Online Sessions, June/July 2018' and provides a short link to the Gist: http://bit.ly/py_db. An illustration of a person's head with gears and a rocket is shown below the text.

Terminal Window (Bottom): The terminal shows the command prompt 'root@Database-Training: ~' and the command 'Package configuration'. A dialog box titled 'Configuring mysql-server-5.7' is displayed, prompting the user to set a password for the MySQL administrative 'root' user. The dialog box text includes: 'While not mandatory, it is highly recommended that you set a password for the MySQL administrative "root" user.', 'If this field is left blank, the password will not be changed.', and 'New password for the MySQL "root" user:'. A password field with a green cursor is visible, and the '<Ok>' button is at the bottom.

Natural Language Processing

Natural Language Processing

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

The image displays three screenshots related to Natural Language Processing (NLP) and Python.

Left Screenshot: Quant Platform

The Quant Platform website shows two video thumbnails for "Natural Language Processing 01" and "Natural Language Processing 02".

- Natural Language Processing 01:** This session covers basic NLP techniques and algorithms. The slides are found under http://hilpisch.com/natural_language.pdf. Files attached: 01_nlp_basics_intro.ipynb, 02_nlp_basics_case.ipynb, nlp_functions.py.
- Natural Language Processing 02:** This session covers advanced NLP techniques and algorithms. The slides are found under http://hilpisch.com/natural_language.pdf. No files attached.

Middle Screenshot: Jupyter Notebook

The Jupyter Notebook interface shows a file explorer on the left with the following files:

- 01_nlp_basics_intro.ipynb
- 02_nlp_basics_case.ipynb
- nlp_functions.py

Right Screenshot: Jupyter Notebook Content

The Jupyter Notebook content shows the title "Natural Language Processing" and the subtitle "Basic Techniques and Algorithms". The introduction text reads:

Introduction

Dr Yves J Hilpisch

The Python Quants GmbH

Text & Data

```
[1]: t = '''
Peter studies data science.
Peter knows Java.
Peter prefers Python.
Peter works as a data scientist.
Peter applies machine learning.
A data scientist uses Python.
Python revolutionized data science.
Python is preferred for NLP.
Python is used for machine learning.
```

Code Snippet:

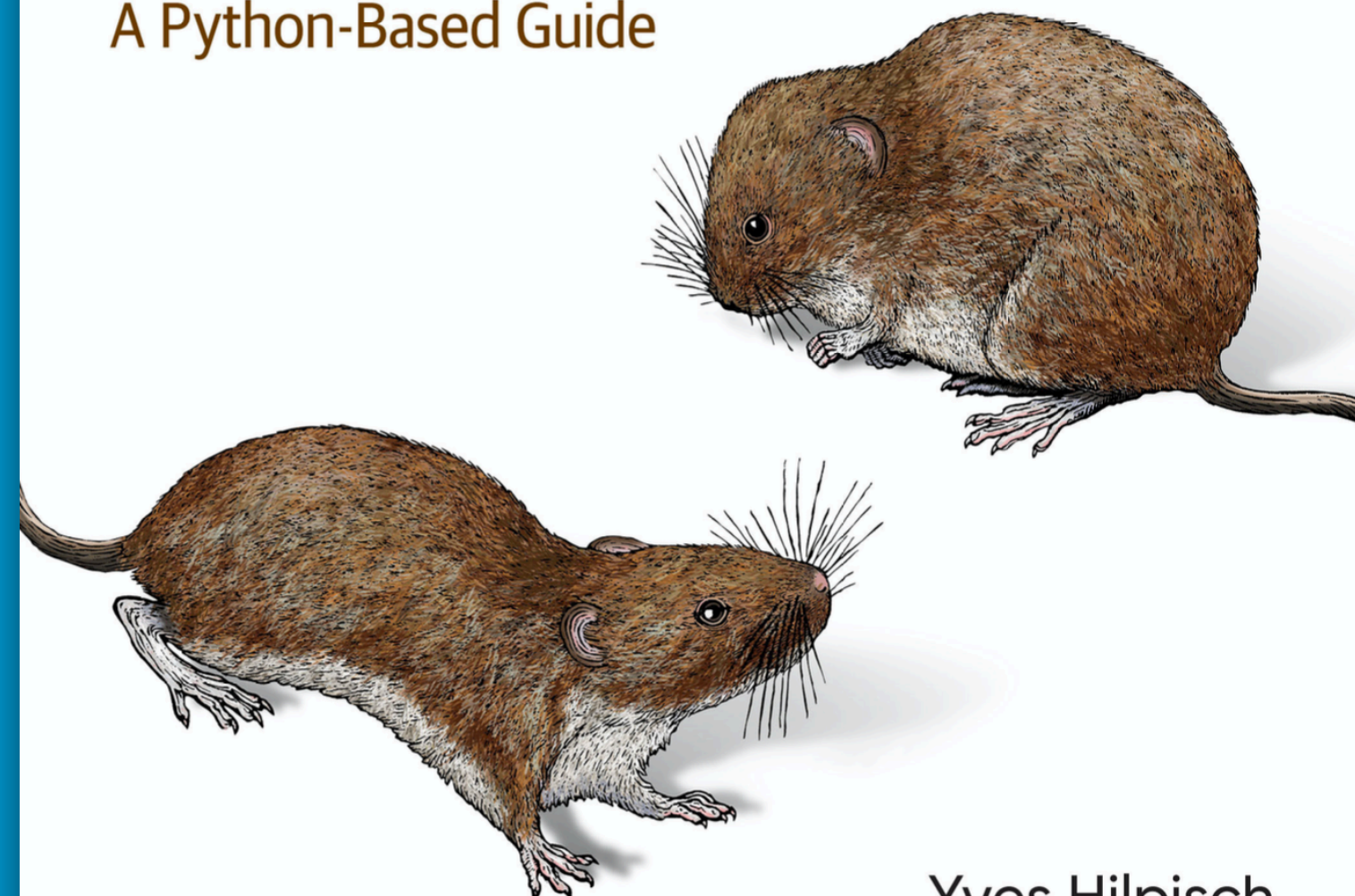
```
24
25 def remove_non_ascii(s):
26     ''' Removes all non-ascii characters.
27     '''
28     return ''.join(i for i in s if ord(i) < 128)
29
30 def clean_up_html(t):
31     t = cleaner.clean_html(t)
32     t = re.sub('[\n\t\r]', ' ', t)
33     t = re.sub(' +', ' ', t)
34     t = re.sub('<.*?>', '', t)
35     t = remove_non_ascii(t)
36     return t
37
```

Artificial Intelligence in Finance

O'REILLY®

Artificial Intelligence in Finance

A Python-Based Guide



Yves Hilpisch

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

Table of Contents

Copyright

Preface

Machine Intelligence

1. Artificial Intelligence

1.1. Algorithms

1.1.1. Types of Data

1.1.2. Types of Learning

Unsupervised Learning

Reinforcement Learning

1.1.3. Types of Problems

1.1.4. Types of Approaches

1.2. Neural Networks

1.2.1. OLS Regression

1.2.2. Estimation with Neural Networks

Scikit-Learn

Keras

1.2.3. Classification with Neural Networks

1.3. Importance of Data

1.3.1. Small Data Set

1.3.2. Larger Data Set

1.3.3. Big Data

1.4. Conclusions

1.5. Further Resources

2. Superintelligence

2.1. Success Stories

1 Multiple hidden layers are added.

2 Network architecture and number of trainable parameters are shown.

Figure 8. Random sample data and neural network estimations

Quant Platform

AI in Finance Session 05

ai_in_finance_05.mp4

This is the fifth session and it covers regression techniques in detail.

You find the slides under <http://certificate.tpq.io/regression.pdf>.

You find a HTML version of the notebook under <http://certificate.tpq.io/regression.html>.

regression.ipynb

AI in Finance Session 06

ai_in_finance_06.mp4

This is the sixth session and it introduces to learning in neural networks.

You find the slides under <http://certificate.tpq.io/learning.pdf>.

You find a HTML version of the notebook under <http://certificate.tpq.io/learning.html>.

learning.ipynb

AI in Finance Session 07

ai_in_finance_07.mp4

This is the seventh session and it further introduces to learning in neural networks.

You find the slides under <http://certificate.tpq.io/improvements.pdf>.

You find a HTML version of the notebook under <http://certificate.tpq.io/improvements.html>.

improvements.ipynb

File Edit View Run Kernel Tabs Settings Help

mlpdnn.ipynb

100/100 [=====] - 0s 286us/step

[26]: print('mse: %.5f' % (scores[1]))

mse: 0.00067

[27]: pred = model.predict(x)

[28]: plt.figure(figsize=(10, 6))

plt.plot(x, y, 'ro', label='original data')

plt.plot(x, pred, label='prediction')

plt.legend()

0 3 Python 3 | Idle

Mode: Command Ln 1, Col 1 mlpdnn.ipynb

24 Modules/Sessions
in 2 Classes

Reinforcement Learning for Finance

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.

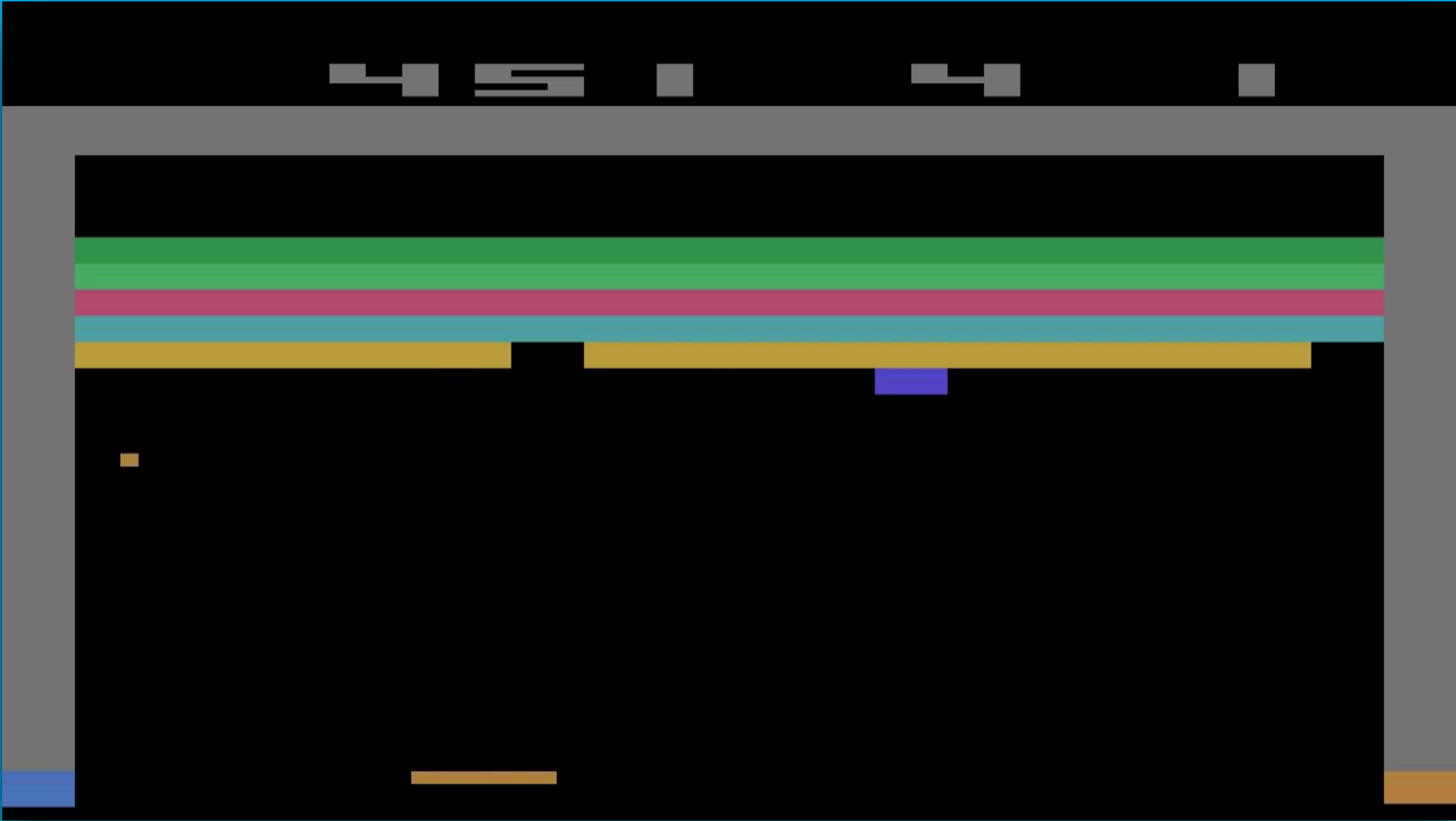
1 Introduction

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

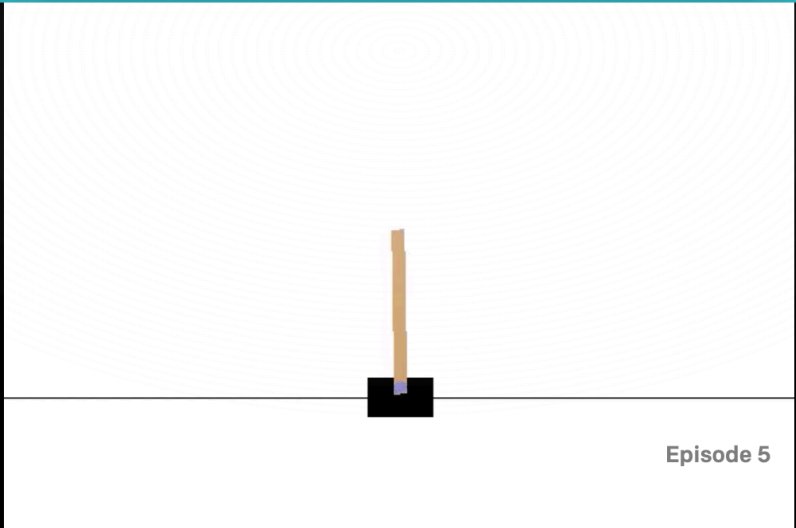
CartPole-v1

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](#)



RandomAgent on CartPole-v1

Environments Documentation

OpenAI

Gym

gym.openai.com/envs/MountainCar-v0/

Environments Documentation

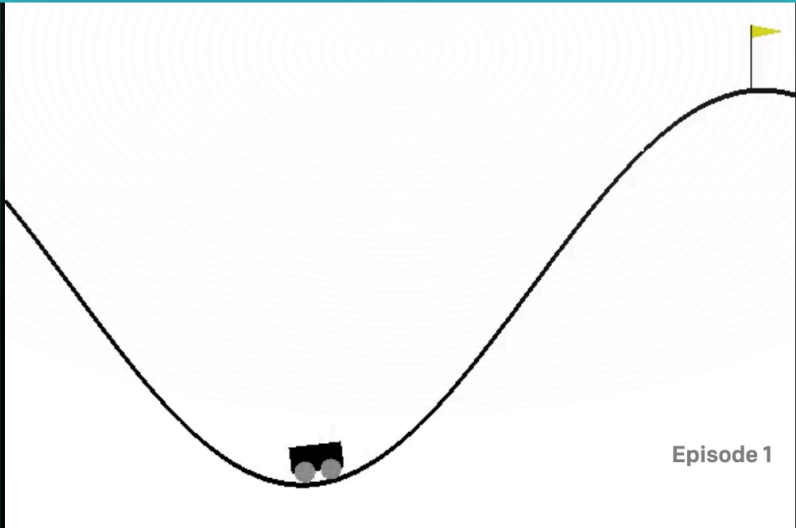
MountainCar-v0

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](#)



RandomAgent on MountainCar-v0

Environments Documentation

OpenAI

Gym

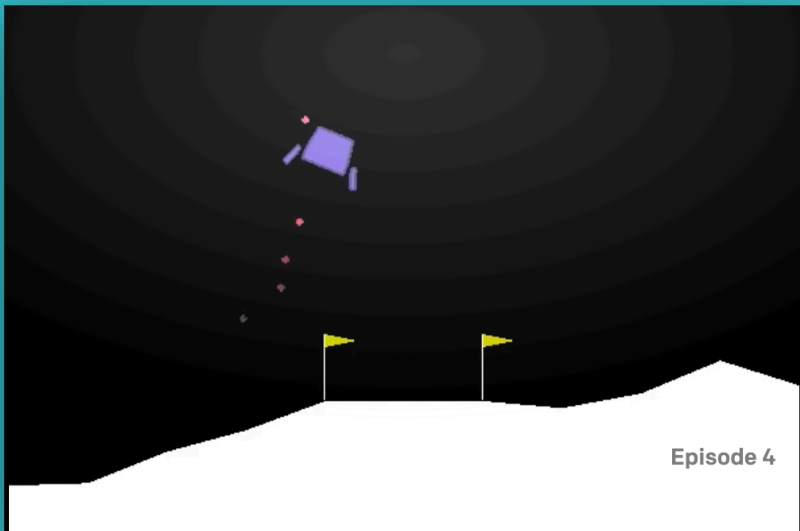
gym.openai.com/envs/LunarLander-v2/

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.

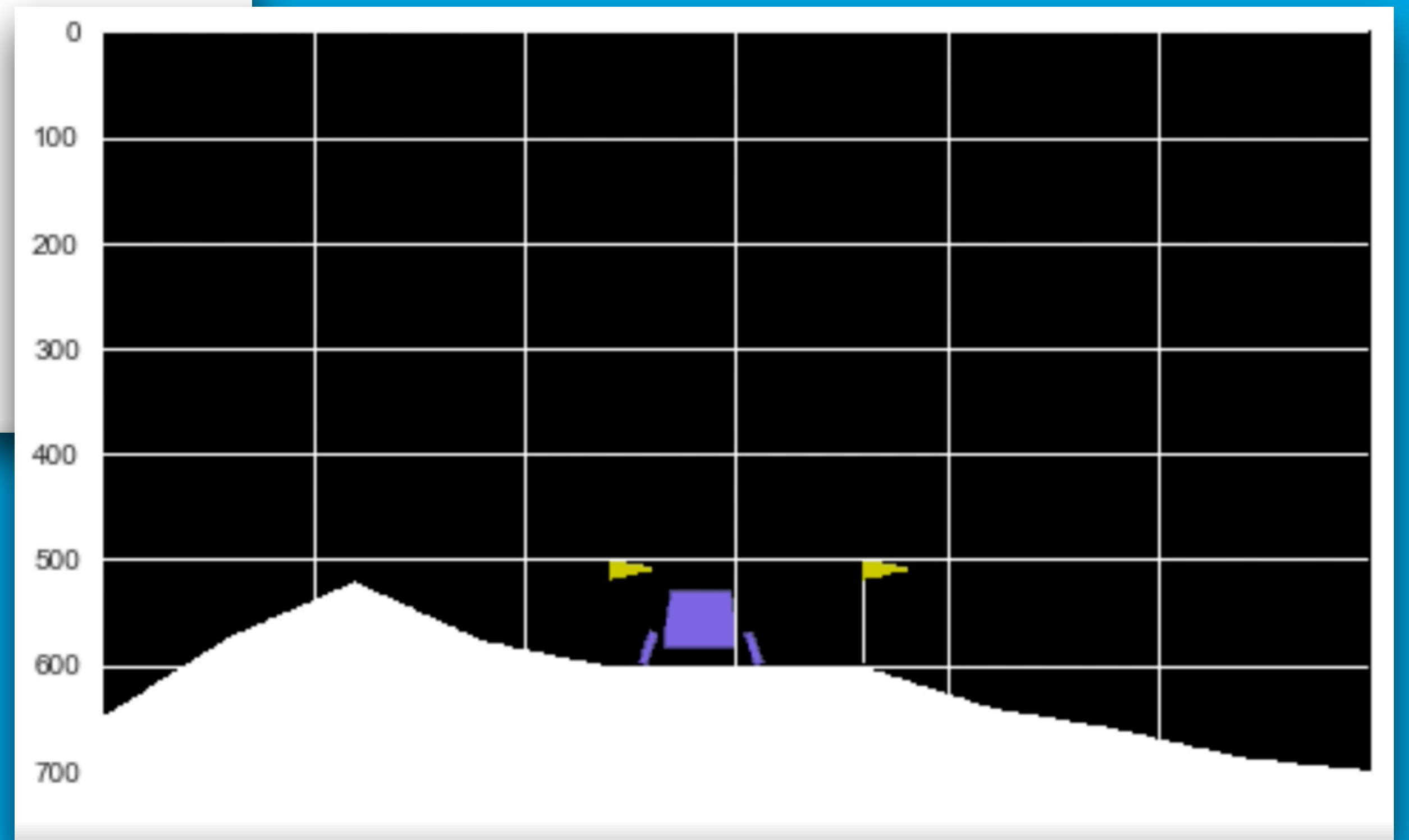
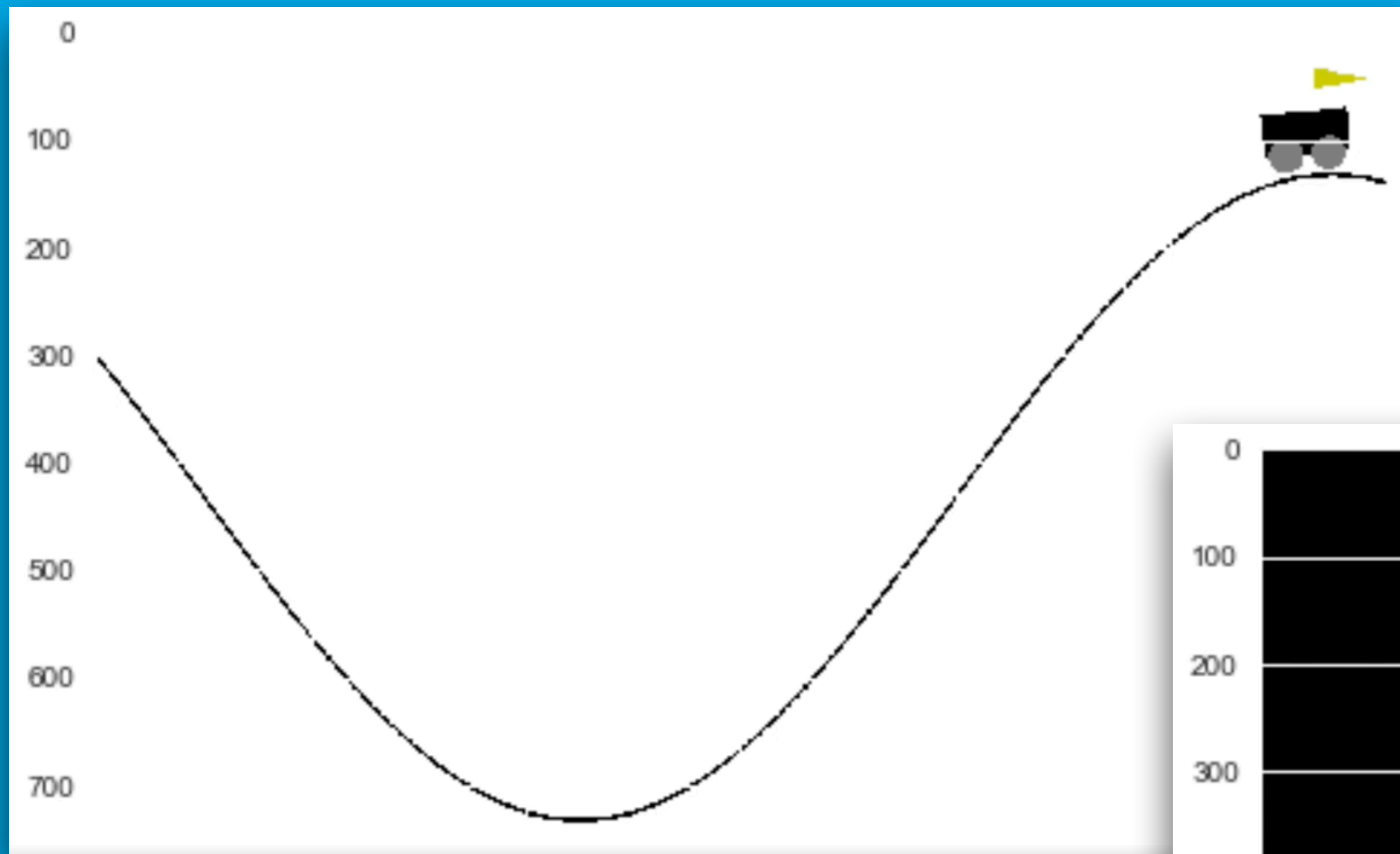
[VIEW SOURCE ON GITHUB](#)



RandomAgent on LunarLander-v2

Environments Documentation

OpenAI



Reinforcement Learning for Finance

From playing games to trading in the financial markets

Quant Platform

Reinforcement Learning 01

reinforcement_learning_01.mp4

This is the Reinforcement Learning 01 session of the Certificate Programs.

You find the slides under <https://certificate.tpq.io/rlearn.pdf>.

- 01_rlearn.ipynb
- 02_rlearn.ipynb
- 03_rlearn.ipynb

Reinforcement Learning 02

reinforcement_learning_02.mp4

This is the Reinforcement Learning 02 session of the Certificate Programs.

You find the slides under <https://certificate.tpq.io/rlearn.pdf>.

- 04_rlearn.ipynb
- 05_rlearn.ipynb
- 06_rlearn.ipynb

Reinforcement Learning 03

reinforcement_learning_03.mp4

This is the Reinforcement Learning 03 session of the Certificate Programs.

You find the slides under <https://certificate.tpq.io/rlearn.pdf>.

- 07_rlearn.ipynb
- 08_rlearn.ipynb

Reinforcement Learning 04

Reinforcement Learning 05

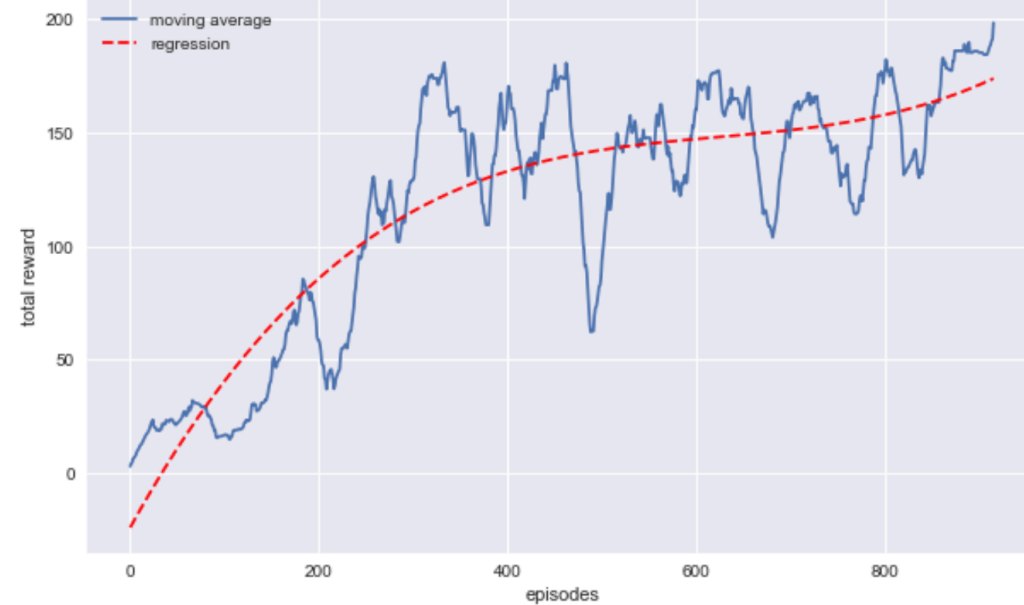
Reinforcement Learning 06

File Edit View Run Kernel Tabs Settings Help

06_rlearn.ipynb

episode: 916/1000 | treward: 200 | av: 198.0 | max: 200

```
[10]: plt.figure(figsize=(10, 6))
x = range(len(agent.averages))
y = np.polyval(np.polyfit(x, agent.averages, deg=3), x)
plt.plot(agent.averages, label='moving average')
plt.plot(x, y, 'r--', label='regression')
plt.xlabel('episodes')
plt.ylabel('total reward')
plt.legend();
```



```
[11]: trewards = agent.test(100)
```

Python 3 | Idle

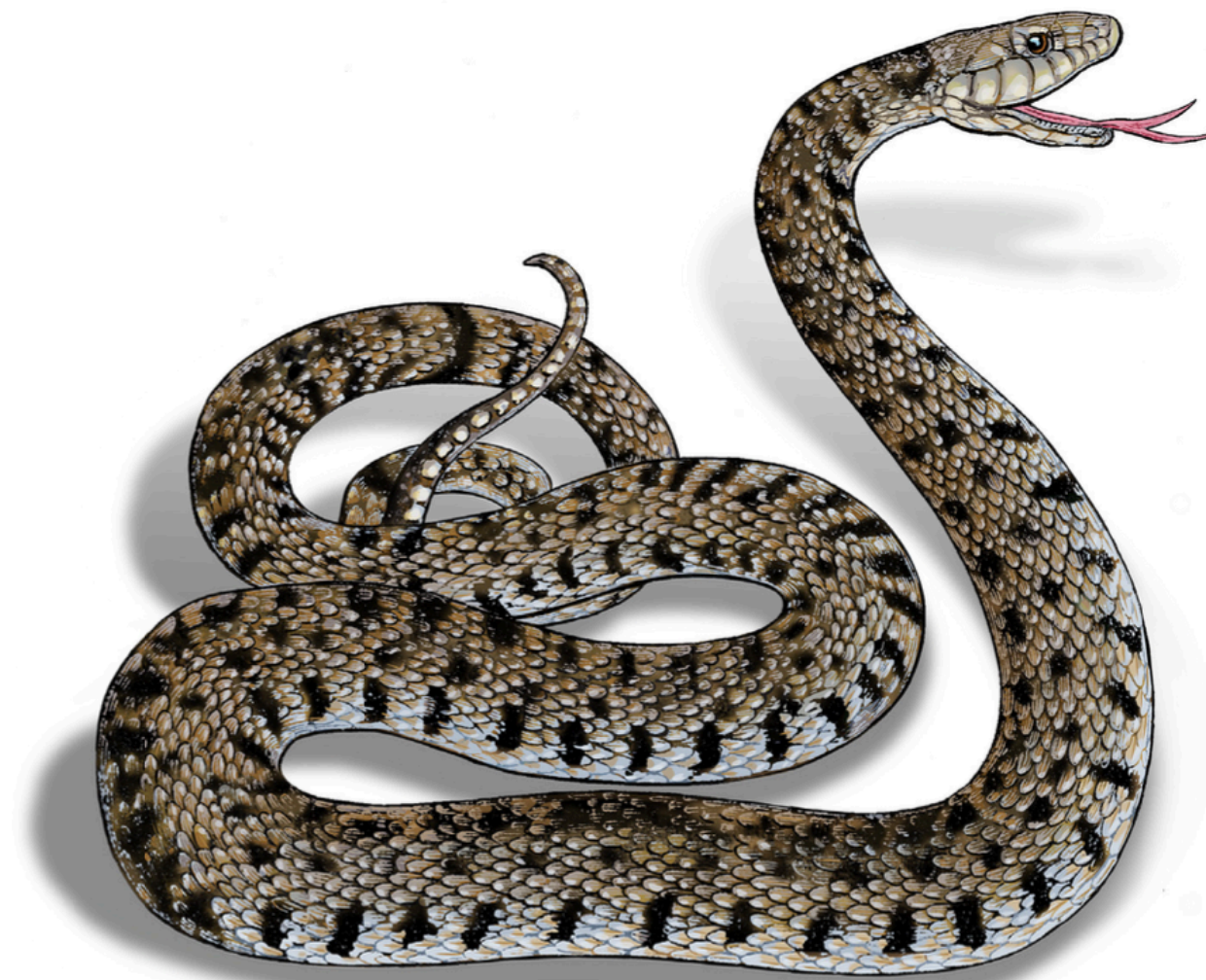
Mode: Command Ln 1, Col 1 06_rlearn.ipynb

Python for Algorithmic Trading

O'REILLY®

Python for Algorithmic Trading

From Idea to Cloud Deployment



Yves Hilpisch

Yves Hilpisch

FOUNDATIONS

STRATEGIES

ALGO TRADING

EXPERIENCE

Finance with Python
01–02

Vectorized
Backtesting

Real-Time Data &
Streaming

Oanda
Trading Platform

Practice Module 1
("own strategy")

Finance with Python
03–04

Prediction-Based
Trading

FXCM
Trading Platform

Interactive Brokers
Trading Platform

Practice Module 2
("deployment")

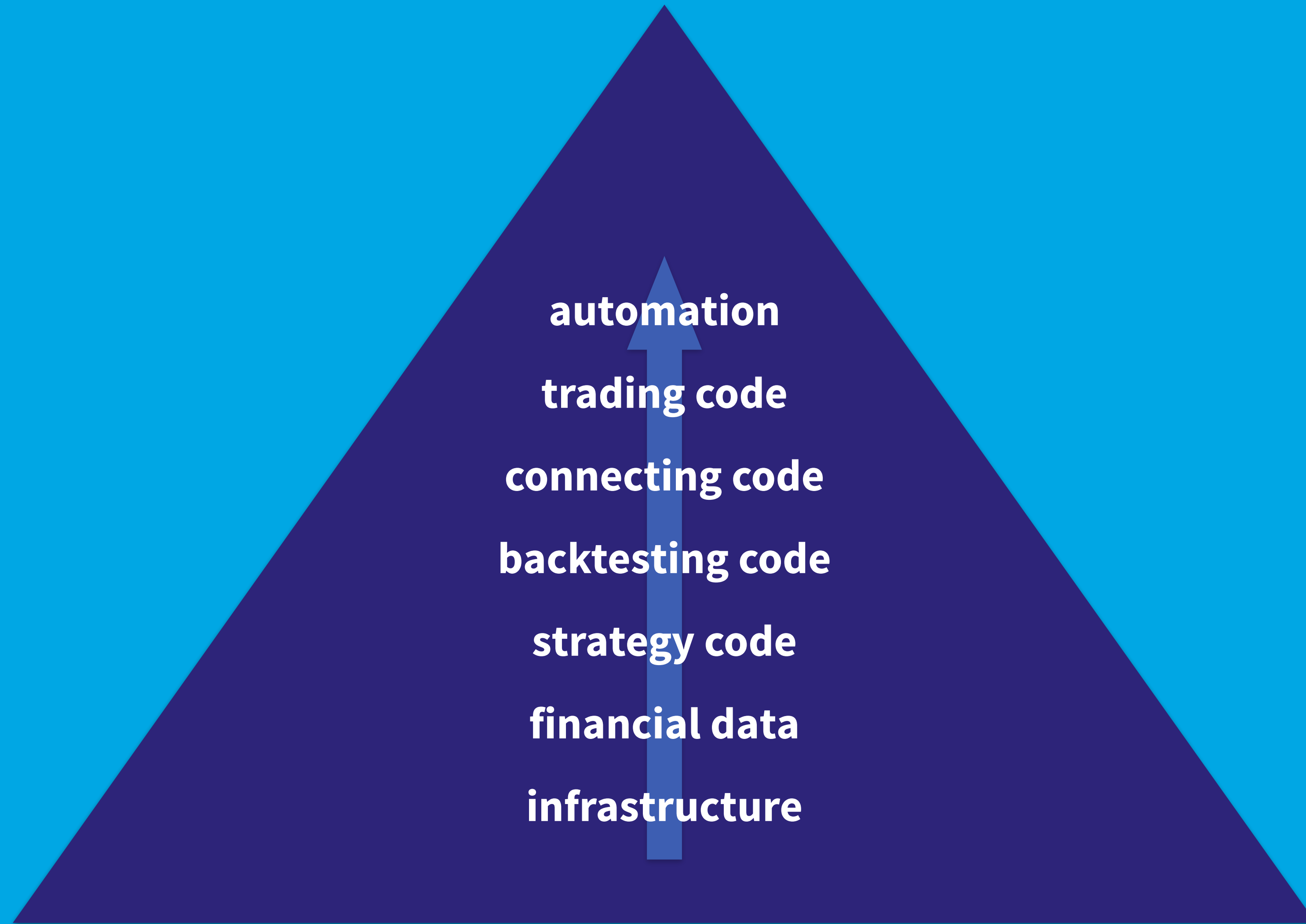
Financial
Data Science

Event-Based
Backtesting

Gemini
Trading Platform

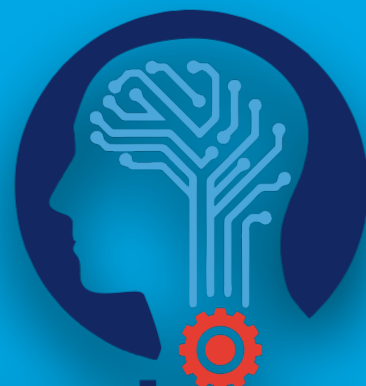
Automation &
Review

Final Project
("research paper")

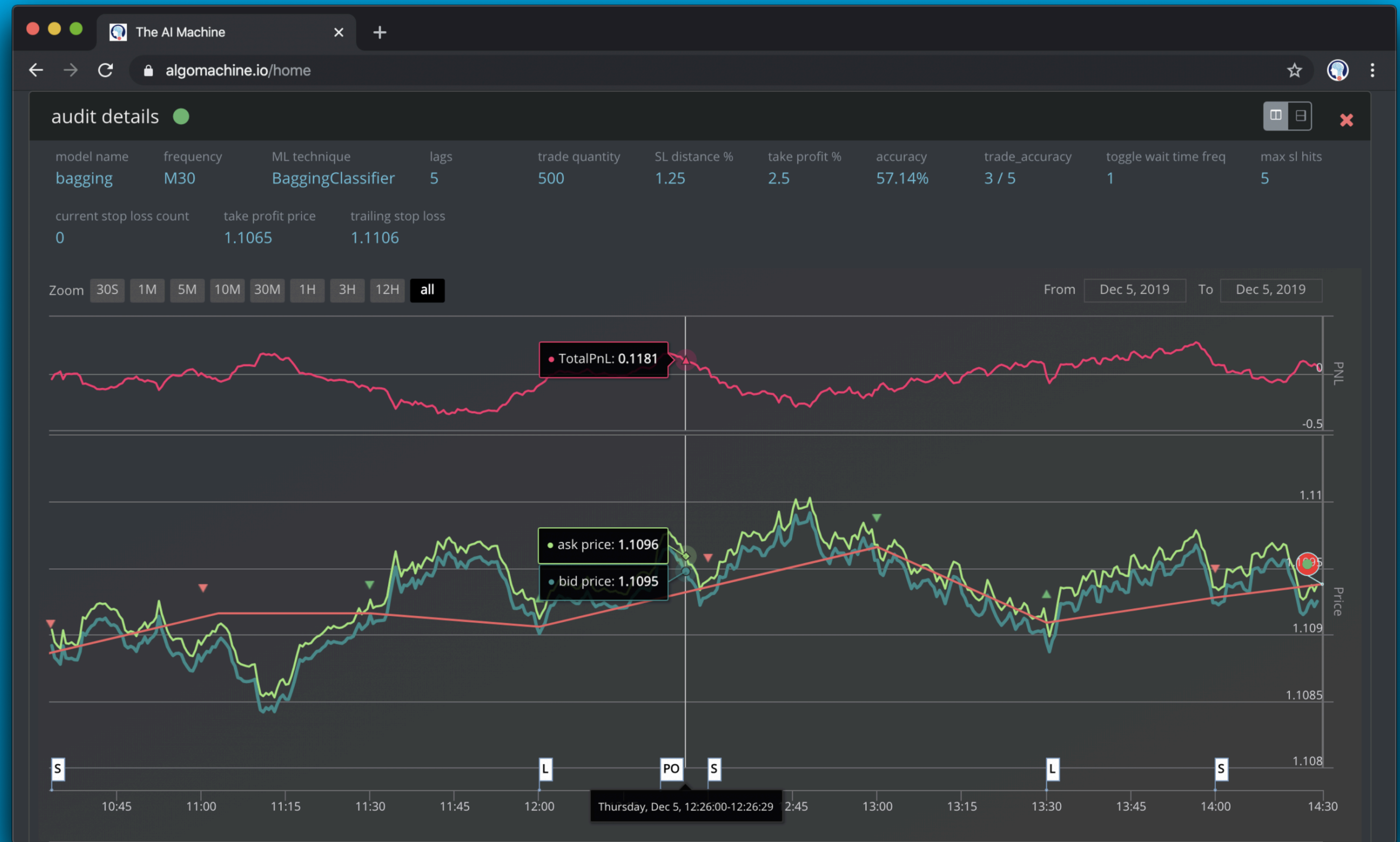




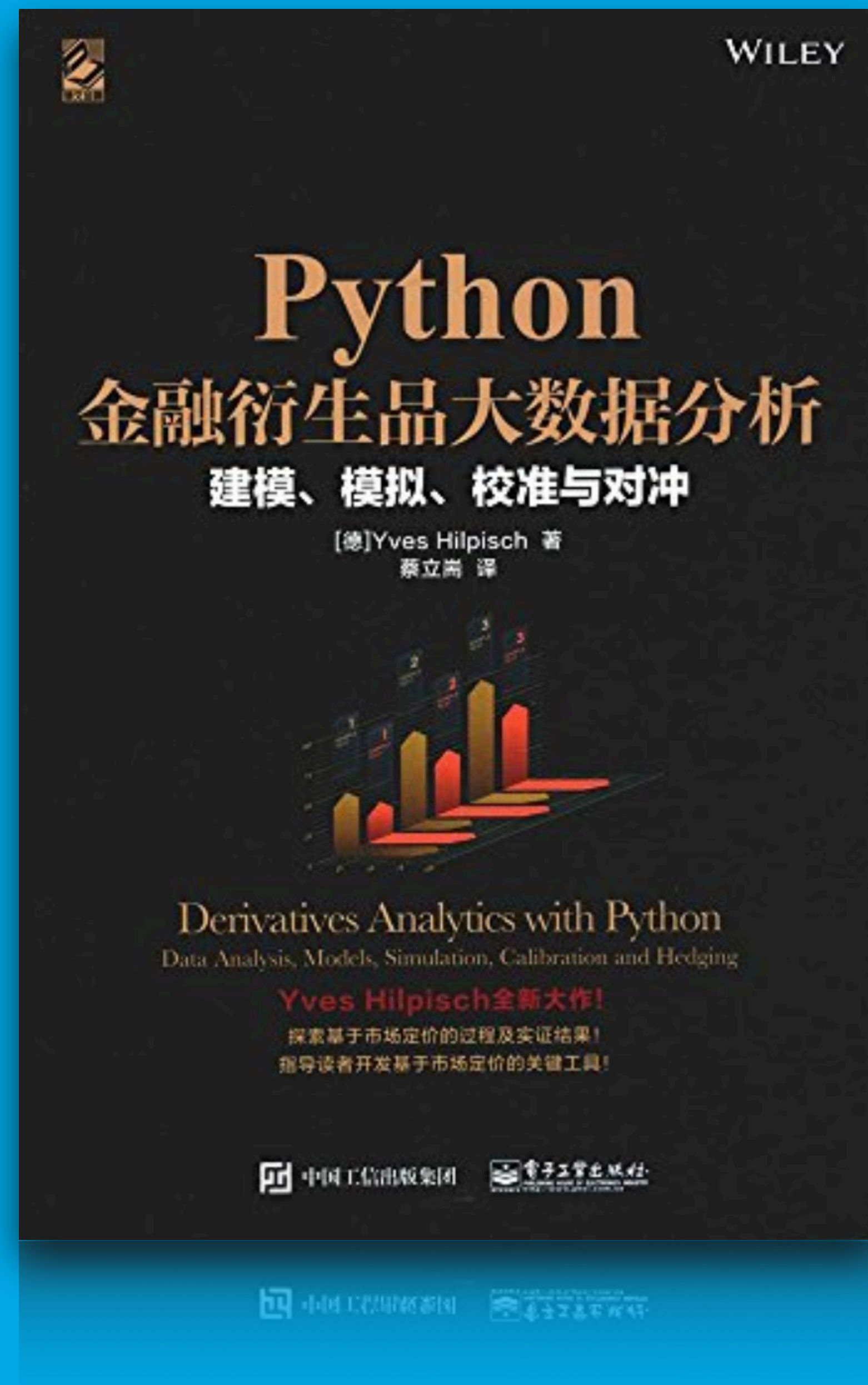
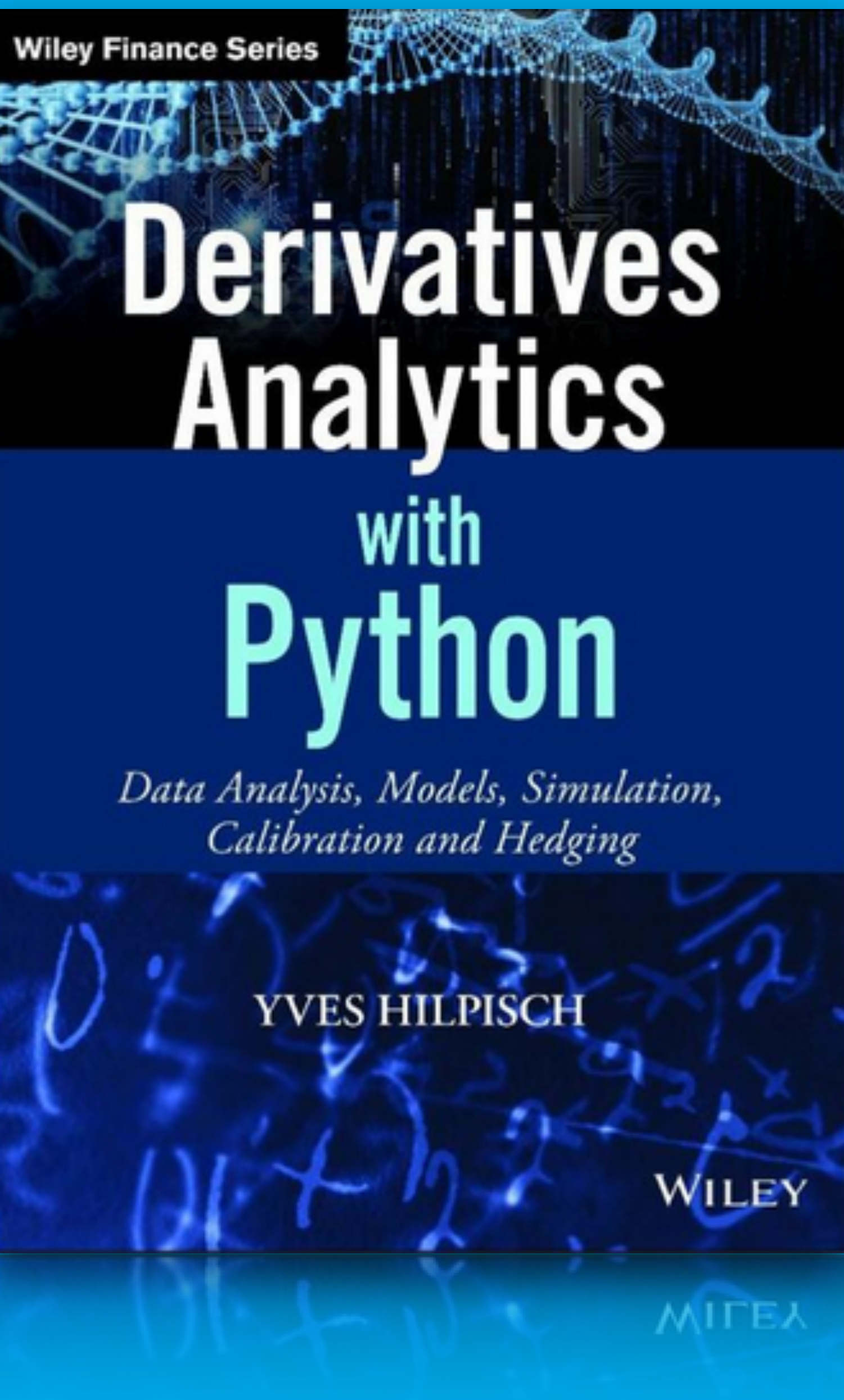
Master Class



The AI Machine



Python for Computational Finance



Computational Finance

BASICS

Market-Based
Valuation

Complete Market
Models

Risk-Neutral
Valuation

Fourier Pricing
Theory

ADVANCED

Fourier Pricing
Applications

American Options

General Market
Model

Monte Carlo
Simulation

APPLICATIONS

Calibration

Hedging

Review

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

Derivatives, Volatility
and Variance

Model-Free
Replication of Variance

VOLATILITY

Data Analysis &
Strategies

VSTOXX Index

Valuing Volatility
Derivatives

Advanced Modelling of
the VSTOXX Index

VARIANCE

Realized Variance &
Variance Swaps

Variance Futures
at Eurex

DX Analytics

An Overview

Square-Root
Diffusion

Square-Root
Jump Diffusion

ADDON

Introduction to
Python

Volatility:
Terms of the VSTOXX
and its Derivatives

Variance:
Trading & Settlement

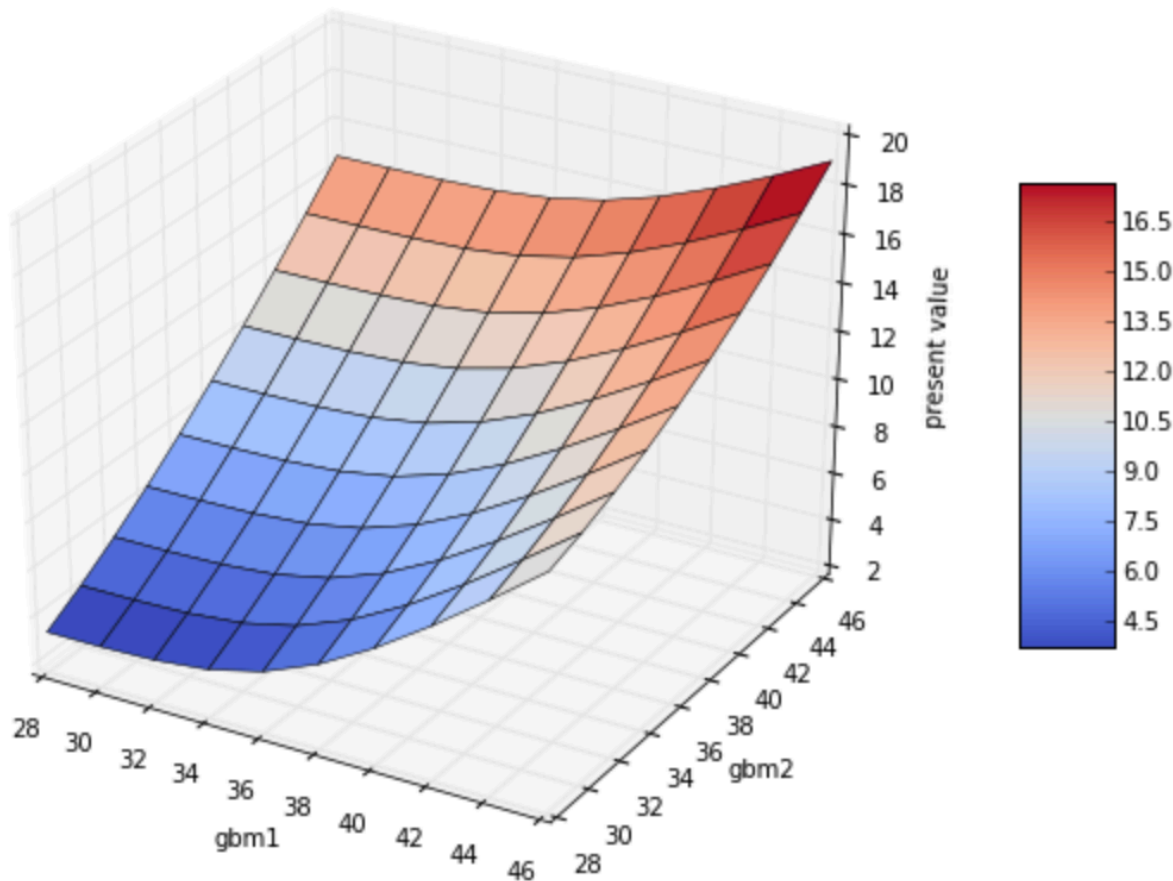
DX Analytics

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:



Navigation

- 1. Quickstart
 - 1.1. Risk Factor Models
 - 1.2. Valuation Models
 - 1.3. Excursion: SABR Model
 - 1.4. Options Portfolio
- 2. Framework Classes and Functions
- 3. Model Classes
- 4. Single-Risk Derivatives Valuation
- 5. Multi-Risk Derivatives Valuation
- 6. Multi-Risk Derivatives Portfolios
- 7. Parallel Valuation of Large Portfolios
- 8. Derivatives Portfolio Risk Statistics
- 9. Fourier-based Option Pricing
- 10. Implied Volatilities and Model Calibration
- 11. Interest Rate Swaps
- 12. Mean-Variance Portfolio Class
- 13. Stochastic Short Rates
- 14. Quite Complex Portfolios

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

Quick Start

Framework &
Model Simulation

European Valuation

Derivatives
Portfolios

ADVANCED

Fourier Pricing

American Valuation

Stochastic
Short Rates

Derivatives
Portfolios

SPECIAL

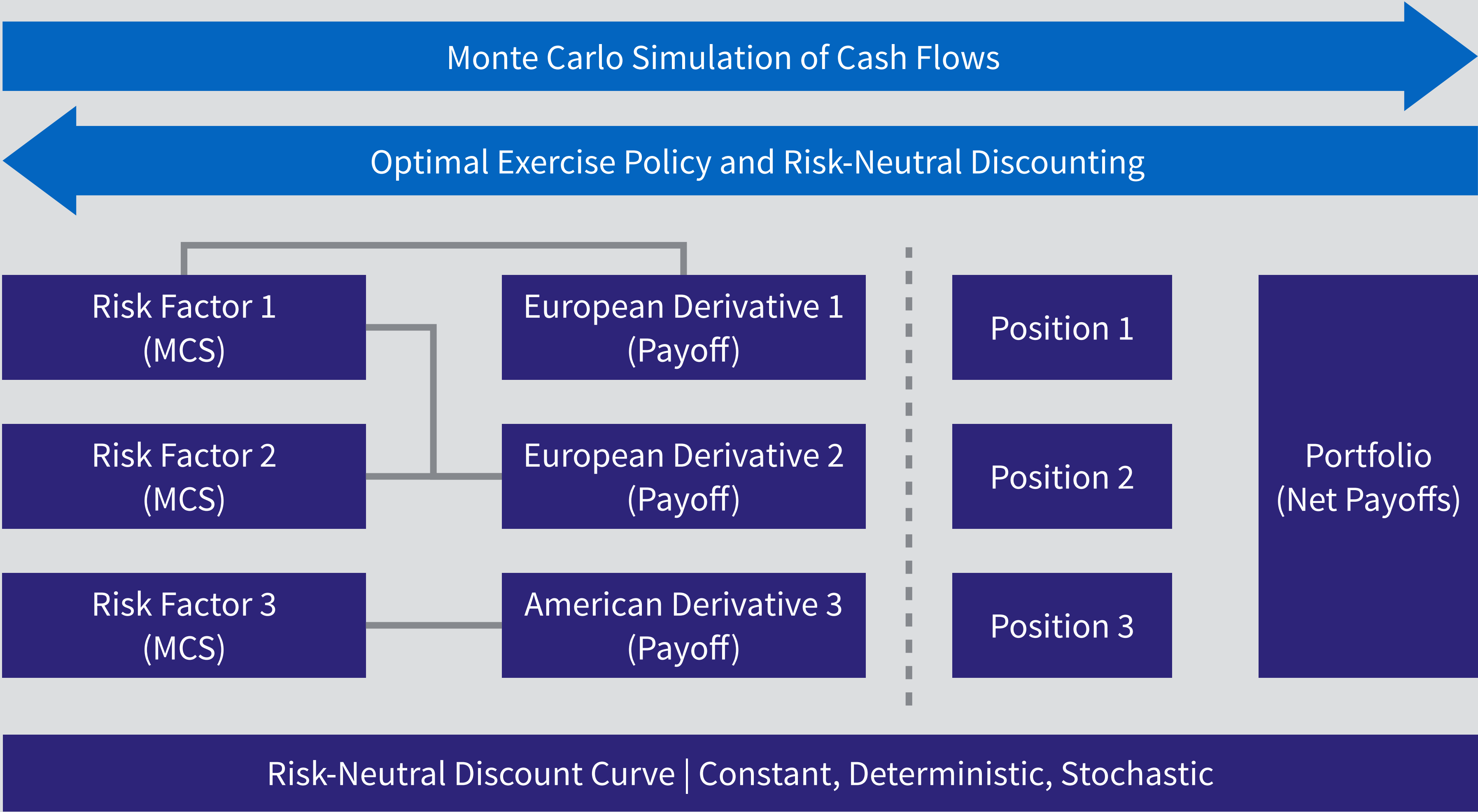
Multi-Risk
Derivatives

Implied Volatilities
& Calibration

Hedging

Complex
Portfolios

**Risk-Neutral Present Values & Greeks for
Instruments, Positions & Portfolios**





Python for Asset Management

Python for Asset Management

Basics of Risk and Return in Finance

Mean-Variance Portfolio Theory

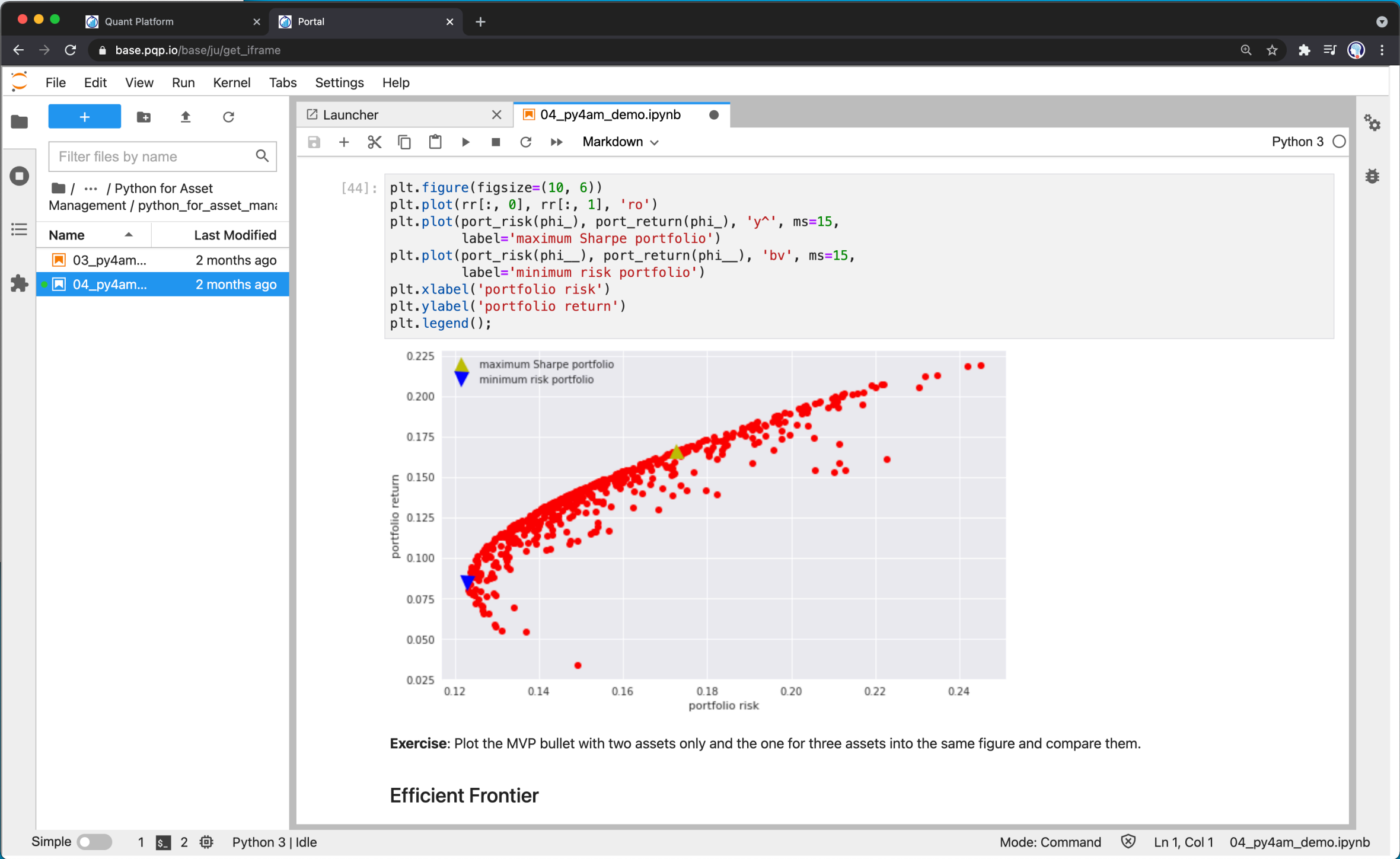
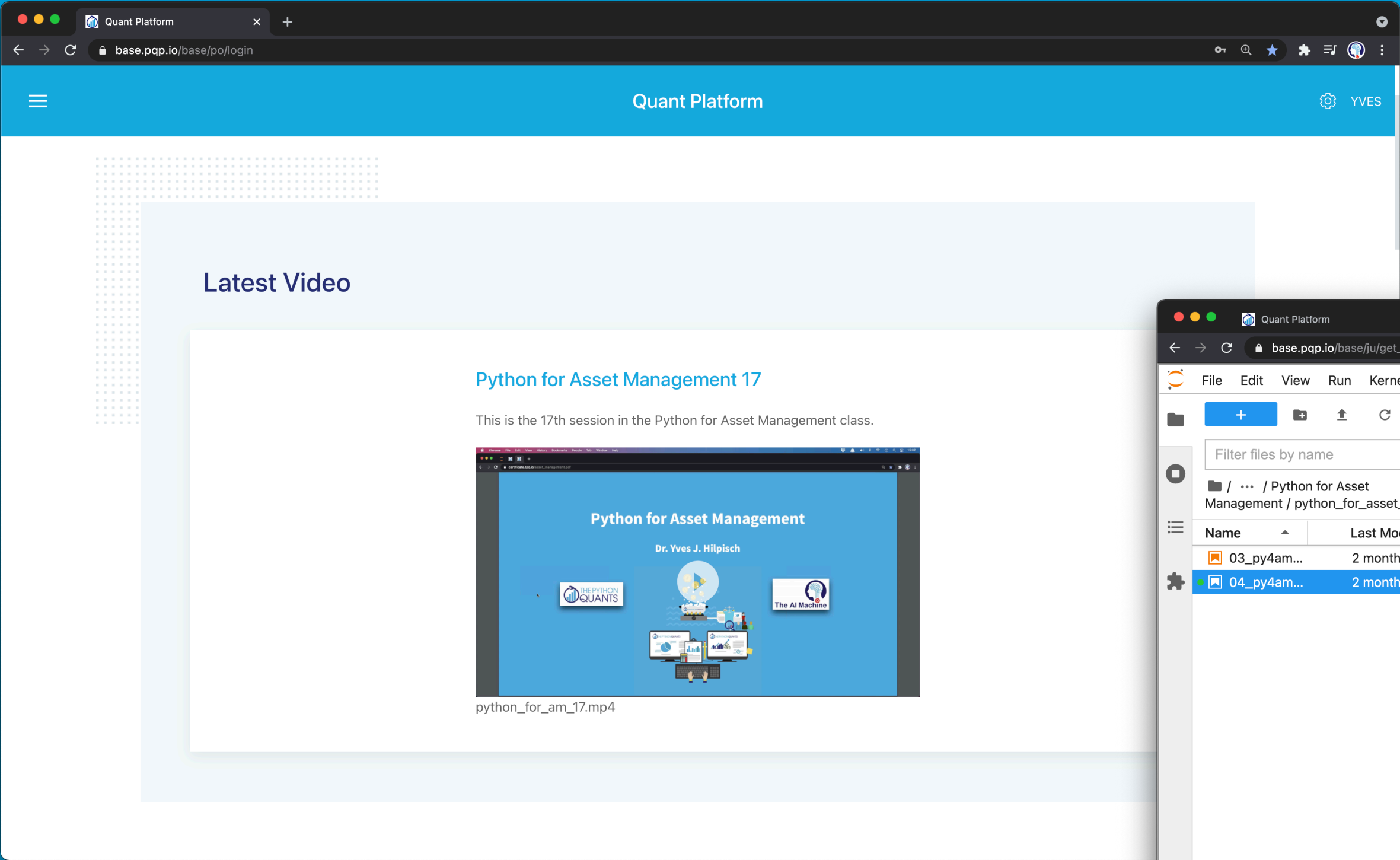
Capital Market Theory
(CAPM, APT)

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, \dots, 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

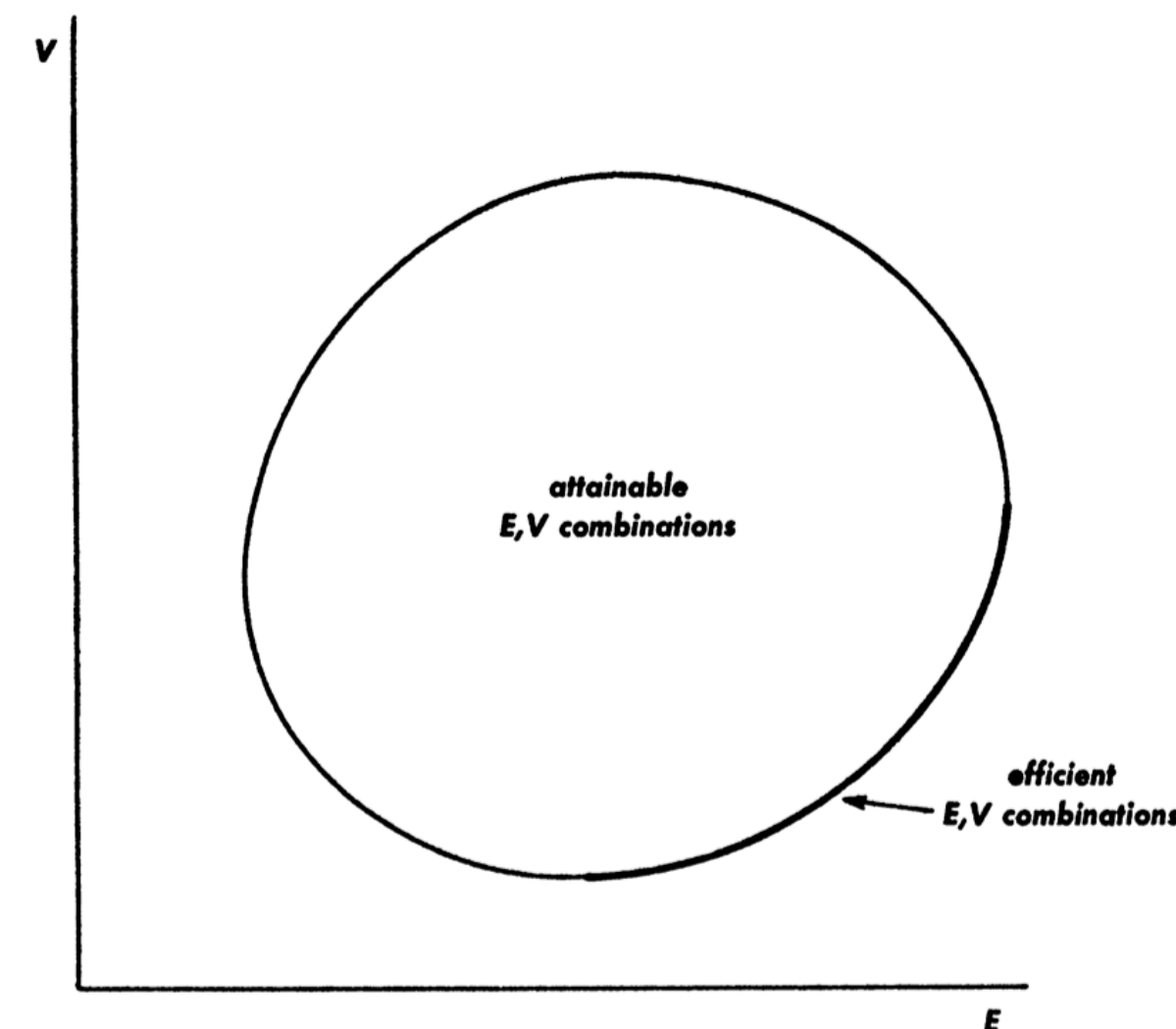


FIG. 1

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

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

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

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

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

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

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

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

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

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 \bar{abc} in Figure 2. Any point to the left of the X_2 axis is not attainable because it violates the condition that $X_1 \geq 0$. Any point below the X_1 axis is not attainable because it violates the condition that $X_2 \geq 0$. Any

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

Thierry Roncalli

Introduction to Risk Parity and Budgeting

Chapman & Hall/CRC FINANCIAL MATHEMATICS SERIES

O'REILLY®

2nd Edition



Python for Finance

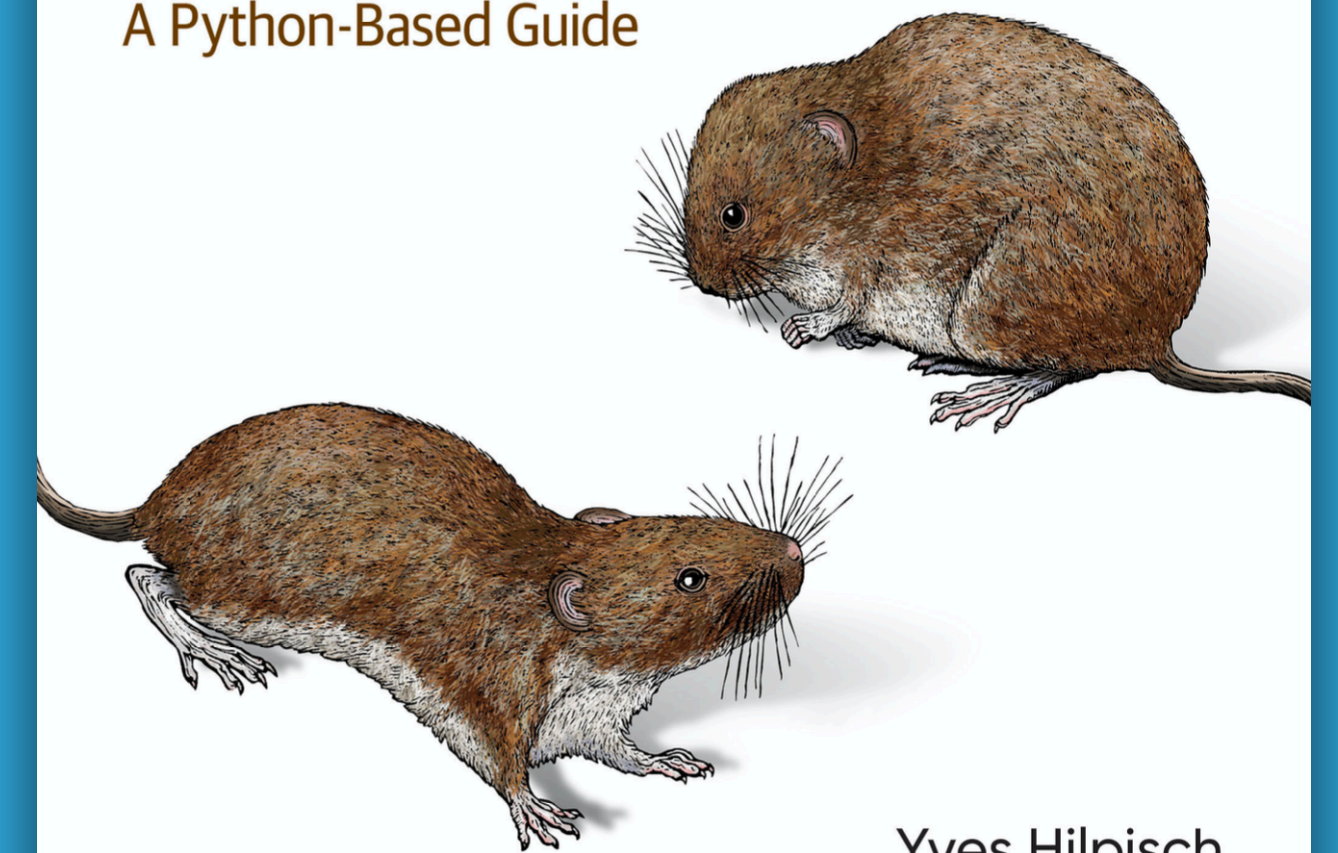
MASTERING DATA-DRIVEN FINANCE

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)

File Edit View Insert Format Tools Extensions Help

100% Normal text Robot... 13 B I U A

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28

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	

The Python Quants GmbH | Certificate Program (16 weeks)

page 2 of 3

training@tpq.io | as of 08. May 2023

https://bit.ly/tpq_4m

Quant Platform

TPQ Certificate Programs (16 v x

TPQ Platinum Package SELF-P x

docs.google.com/document/d/1DKzIzir_q_BTtpVCJBbKkH0oIF1o3ekG5XK_82szzw/edit

TPQ Platinum Package SELF-PACED (1 yr)

Request edit access

Share

File Edit View Tools Help

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

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

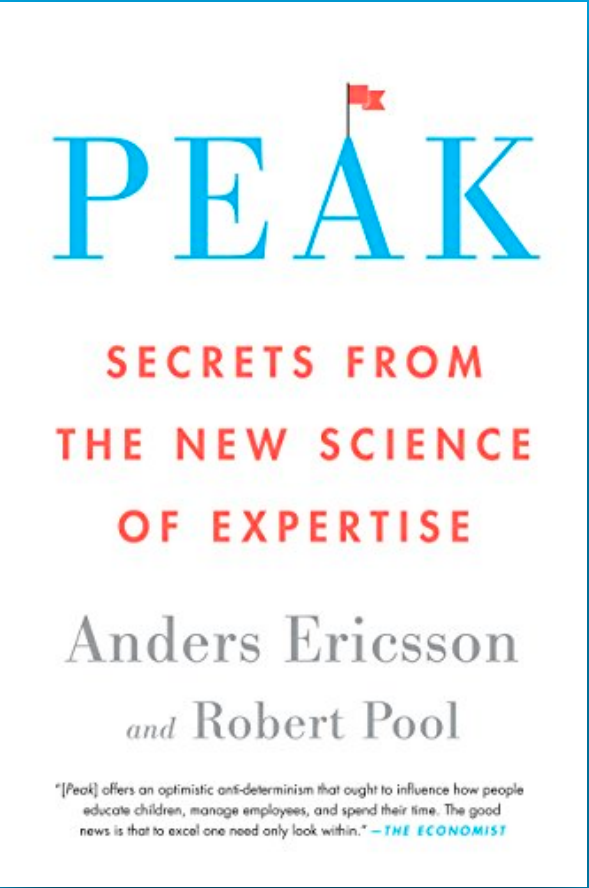
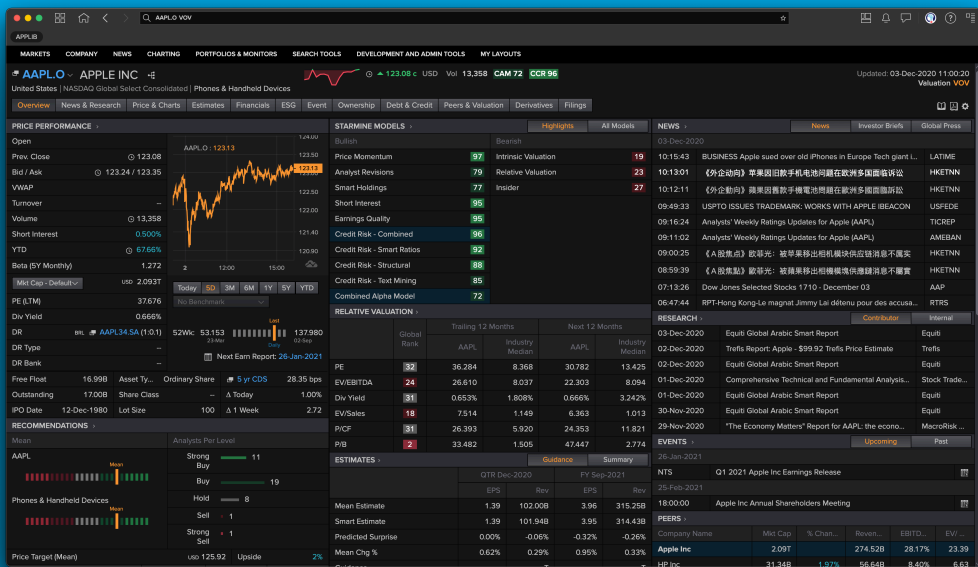
In [63]: np.dot(P, S1) 4
Out[63]: 11.0
```

Reproducible

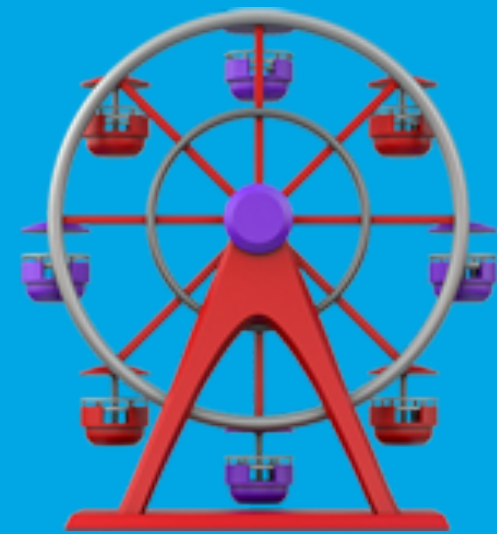
All examples are based on static data sets to allow for reproducibility of results.

Practical

To acquire coding and other practical skills is the main goal. Therefore skill acquisition 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



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?

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

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

- the user is informed about the number of guesses needed
- the game ends when the user has come up with the correct number

User Forum

Quant Platform

MENU

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

Mathematics for Economists alternative in electronic form

on Aug 13 2022 14:56 | Last reply: Aug 23 2022 08:00 | 3 Replies

camp

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

final-projects

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 need...

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

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

