For Python Quants Bootcamp

CQF | INSTITUTE

THE PYTHON QUANTS

Dr. Yves J. Hilpisch | @dyjh

London, 21. — 24. November 2017

Introduction

SERVICES
for financial institutions globally

EVENTS
for Python quants &
algorithmic traders

THE PYTHON QUANTS

TRAINING
about Python for finance
& algorithmic trading

CERTIFICATION
in cooperation with university

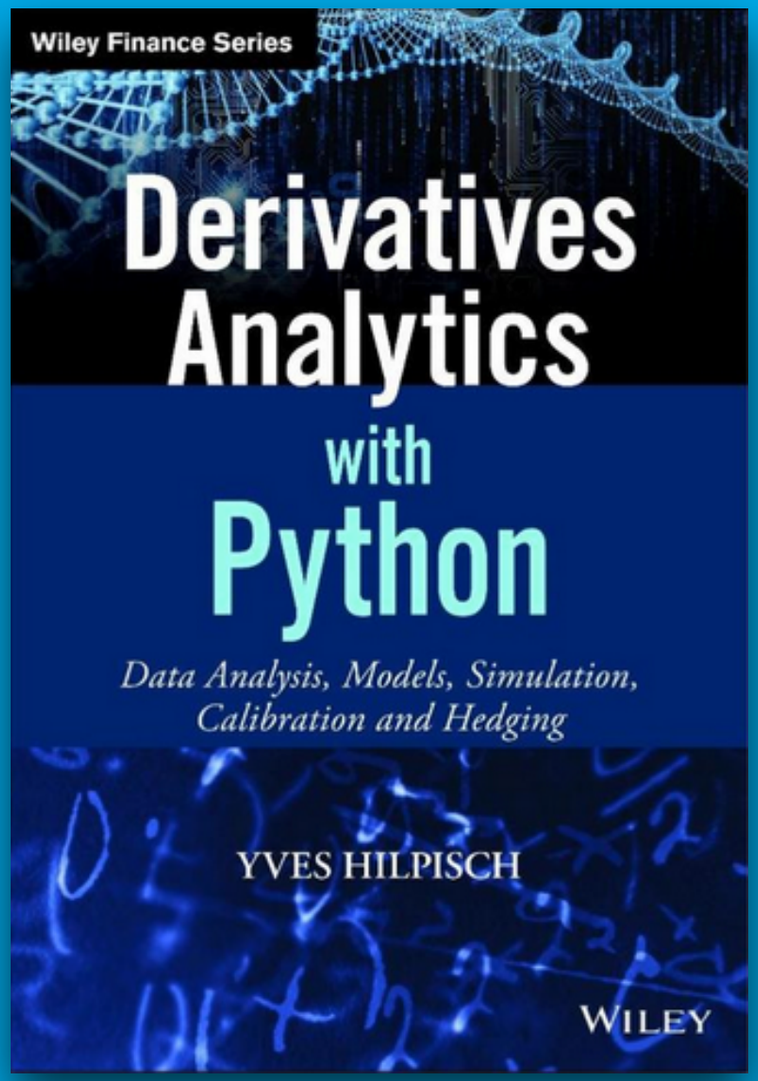
BOOKS
about Python and
finance

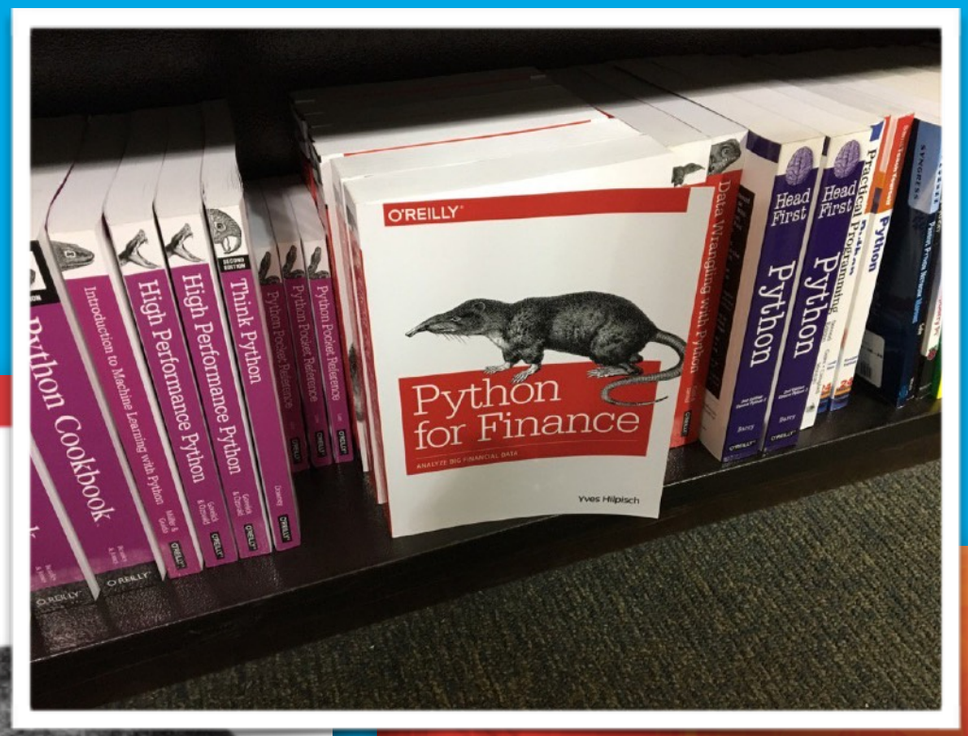
PLATFORM
for browser-based
data analytics

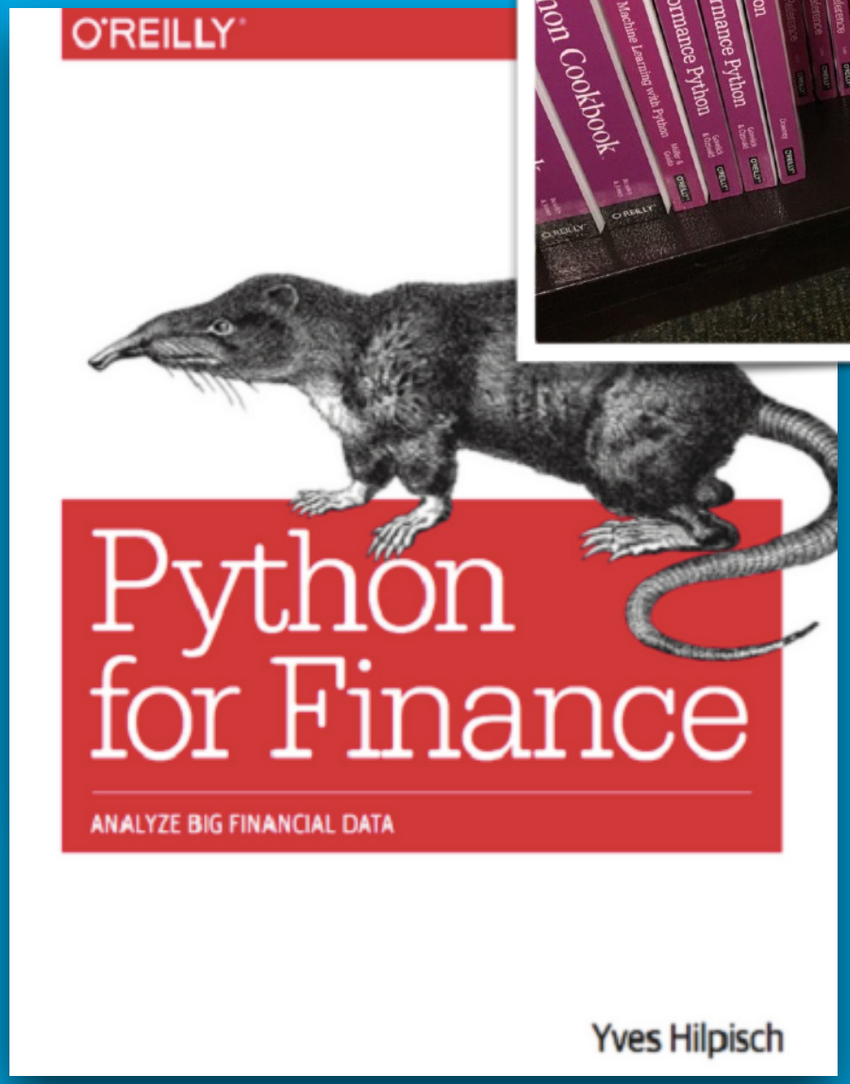
OPEN SOURCE
Python library
for financial analytics

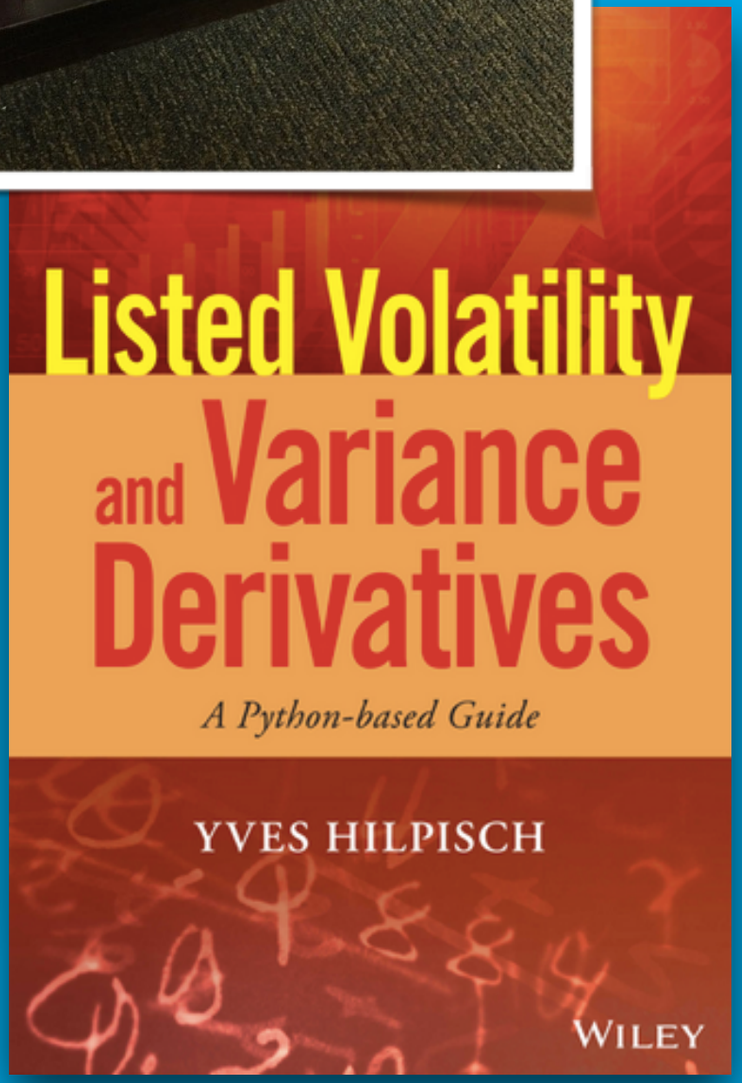
http://tpq.io

http://hilpisch.com





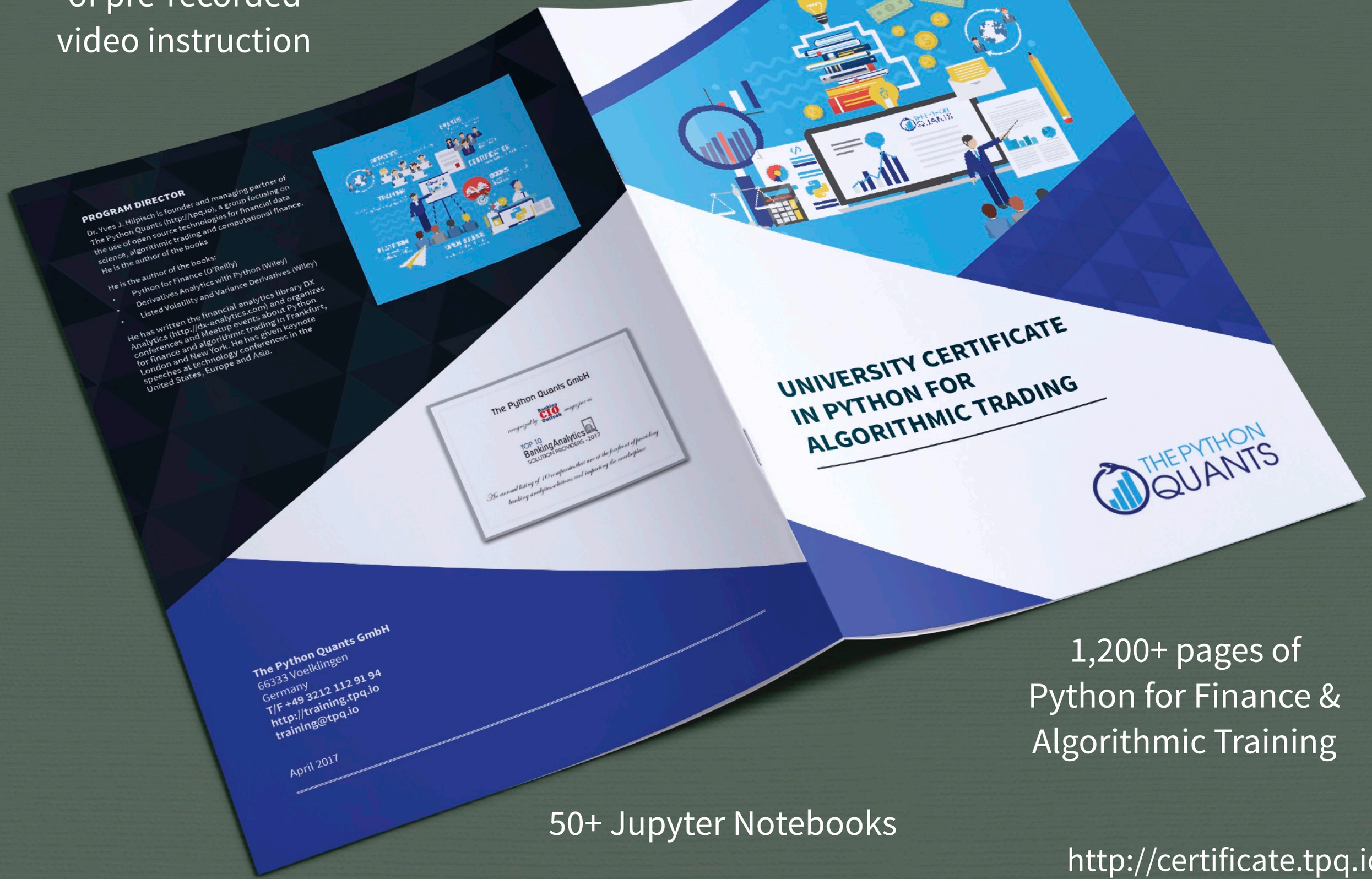




http://books.tpq.io

125+ hours
of pre-recorded
video instruction

5,000+ lines of code



1,200+ pages of
Python for Finance &
Algorithmic Training

50+ Jupyter Notebooks

http://certificate.tpq.io

THOMSON REUTERS

FitchLearning

CQF | INSTITUTE

htw saar

Hochschule für Technik und Wirtschaft des Saarlandes
University of Applied Sciences

Resources

Slides

http://hilpisch.com/bootcamp.pdf

Gist

https://goo.gl/L8xZ8X

AI-First Finance

machine & deep learning

data

algorithms

hardware

optimization, training & learning

testing

validation

prediction (“self-driving car”)

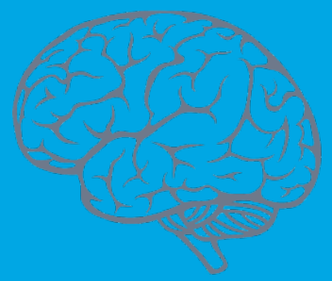
automation

trading (“money making machine”)

algorithmic trading

Humans

x



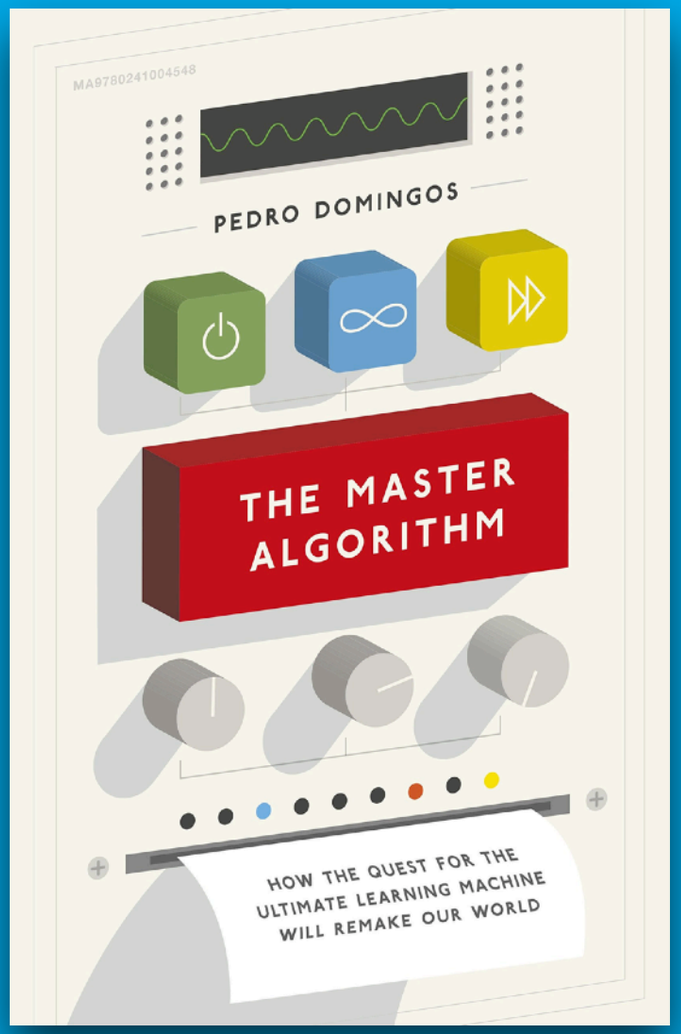
y

Algorithms

x

f(x)

y



"The grand aim of science is to cover the greatest number of experimental facts by logical deduction from the smallest number of hypotheses or axioms."
— Albert Einstein

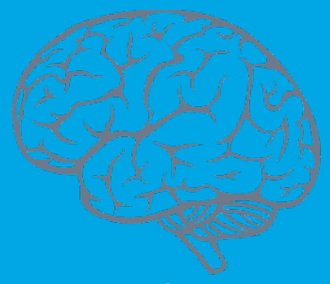
"Machine learning is the scientific method on steroids. It follows the same process of generating, testing, and discarding or refining hypotheses. But while a scientist may spend his or her whole life coming up with and testing a few hundred hypotheses, a machine-learning system can do the same in a second. Machine learning automates discovery. It's no surprise, then that it's revolutionizing science as much as it's revolutionizing business."

Financial Markets

x

y

Finance History



f(•)

f(x) ≠ y

“brain driven”

AI in Finance = finaince

x

m(•, a, b)

finaince

m(x, a*, b*) ≈ y

“data driven”

Markets & Agents

Algorithms

x

x

m(x, a*, b*)

y

y

Why Python for AI-First Finance?

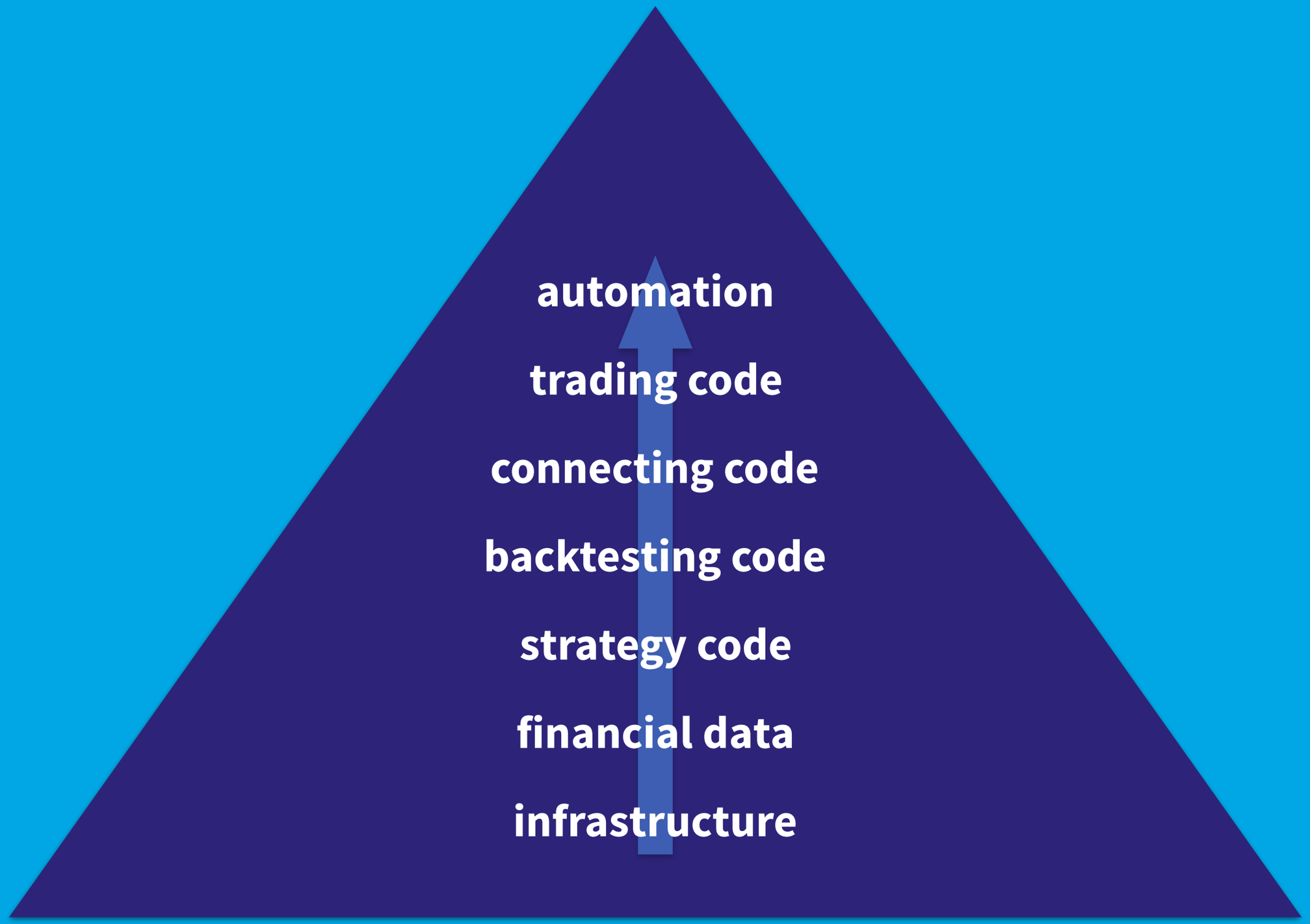
MACHINE LEARNING &
AI-FIRST FINANCE NEED …

… access to lots of historical, granular data sets

… access to real-time (“streaming”) data

… flexible algorithms that can be efficiently trained

… powerful soft- and hardware


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

PYTHON'S BENEFITS ...	... COMPARED TO
1. open source software	vendor developed & maintained
2. general purpose language	domain specific language
3. multi-paradigm language	single-paradigm languages
4. powerful ecosystem of packages	weak ecosystems
5. leading in data science	just good in finance or single area
6. first class citizen in AI	no access to AI world
7. core technology in finance	just a "somehow used" technology
8. supported by many players	emphasized by selected players
9. strong and open communities	vendor driven and/or small communities
10. books, resources, trainings	vendor and/or few specialized resources

Program

DAY 1	DAY 2	DAY 3	DAY 4
introduction infrastructure	data science case with CSV, NumPy, SQL	financial data with pandas, data API with Flask	stock market prediction with deep learning
first steps data structures first algorithm	data science case with pandas & classification algos from machine learning	vectorized backtesting of trading algorithms	streaming data & visualization
finance in a complete 2-state economy	NumPy for efficient numerical computations	stock market prediction with regression & ML	algorithmic trading with Oanda
finance in an incomplete economy	mean-variance portfolio theory with pandas & SciPy	object oriented programming	deployment & automation

“In building a house, there is the problem of the selection of wood.
It is essential that the carpenter’s aim be to carry equipment that will cut well and,
when he has time, to sharpen that equipment.”

Miyamoto Musashi (The Book of Five Rings)

“Any fool can write code that a computer can understand.
Good programmers write code that humans can understand.”

Martin Fowler

“In fact, I'm a huge proponent of designing your code around the data,
rather than the other way around, …”

Linus Torvalds

“Dataism says that the universe consists of data flows, and the value of any phenomenon or entity is determined by its contribution to data processing. … Dataism thereby collapses the barrier between animals [humans] and machines, and expects electronic algorithms to eventually decipher and outperform biochemical algorithms”

Yuval Noah Harari (Homo Deus)

Interactive Style Throughout

"Making mistakes together."

vim

```
#
# Simple Tick Data Server with
# ZeroMQ
#
import zmq
import time
import random

context = zmq.Context()
socket = context.socket(zmq.PUB)
socket.bind('tcp://0.0.0.0:5555')

AAPL = 100.

while True:
    AAPL += random.gauss(0, 1) * 0.5
    msg = 'AAPL %s' % AAPL
```

tick_server.py [+]

```
#
# Simple Tick Data Client with
# ZeroMQ
#
import zmq
import datetime

context = zmq.Context()
socket = context.socket(zmq.SUB)
socket.connect('tcp://0.0.0.0:5555')
socket.setsockopt_string(zmq.SUBSCRIBE, 'AAPL')

while True:
    msg = socket.recv_string()
    t = datetime.datetime.now()
    print('%s | %s' % (t, msg))
```

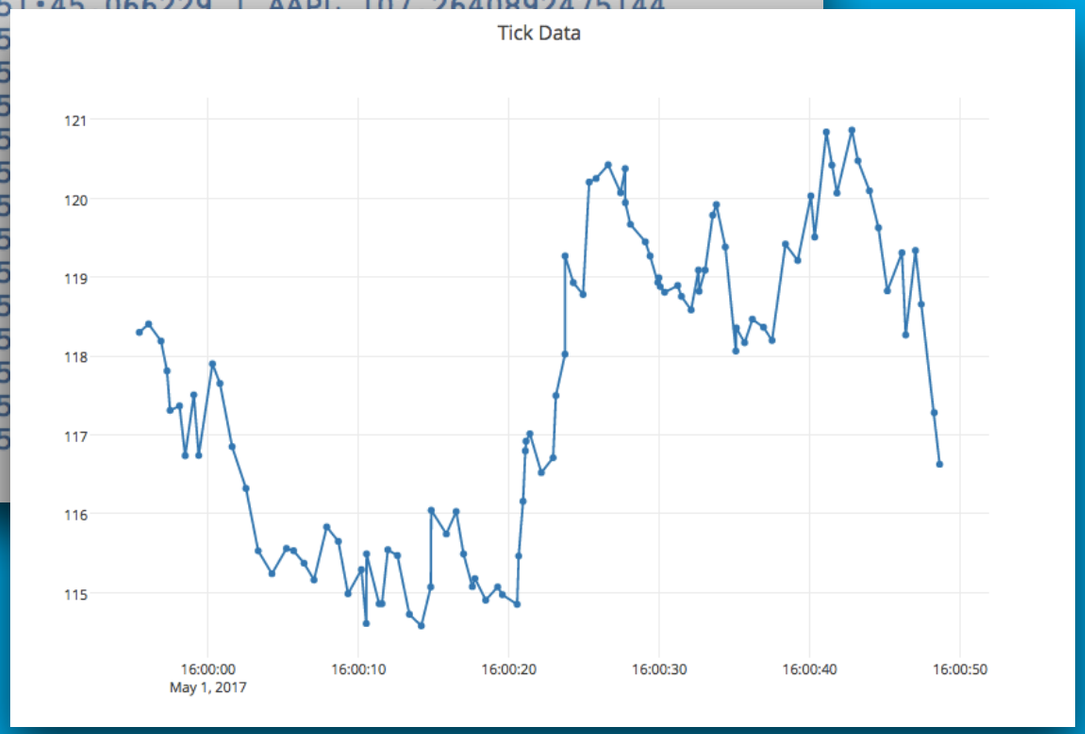
tick_client.py

IPython: live/data (python3.6)

```
AAPL 107.15636235397254
AAPL 107.18612019583905
AAPL 107.4983187955743
AAPL 107.2640892475144
AAPL 107.68358829560407
AAPL 106.9232056802307
AAPL 106.55017297488794
AAPL 105.97708319698597
AAPL 106.00856053822193
AAPL 105.37221723045396
AAPL 105.09251644774177
AAPL 104.9267694947986
AAPL 105.03306681222703
AAPL 105.1223727550806
AAPL 105.29880694705703
AAPL 105.438670667864
AAPL 105.60426198517378
```

root@pythonquants02: ~ (python3.6)

```
2017-05-01 23:51:44.010545 | AAPL 106.94730057503057
2017-05-01 23:51:44.184665 | AAPL 107.15636235397254
2017-05-01 23:51:44.663153 | AAPL 107.18612019583905
2017-05-01 23:51:44.707051 | AAPL 107.4983187955743
2017-05-01 23:51:45.066229 | AAPL 107.2640892475144
2017-05-01 23:5
2017-05-01 23:5
2017-05-01 23:5
2017-05-01 23:5
2017-05-01 23:5
2017-05-01 23:5
2017-05-01 23:5
2017-05-01 23:5
2017-05-01 23:5
2017-05-01 23:5
2017-05-01 23:5
2017-05-01 23:5
```



The Python Quants GmbH

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