Introduction

CPS600 Python Programming for Finance
Fall 2018

August 28, 2018

Welcome! What is this?

This is a course in Python, really.

Welcome! What is this?

This is a course in Python, really.

You should already know the finance concepts, since *since I will not expound on them*

Welcome! What is this?

This is a course in Python, really.

You should already know the finance concepts, since *since I will not expound on them*

We will get familiar with Python libraries useful in financial analytics and we will learn all of the programming techniques you'll need to make use of those libraries.

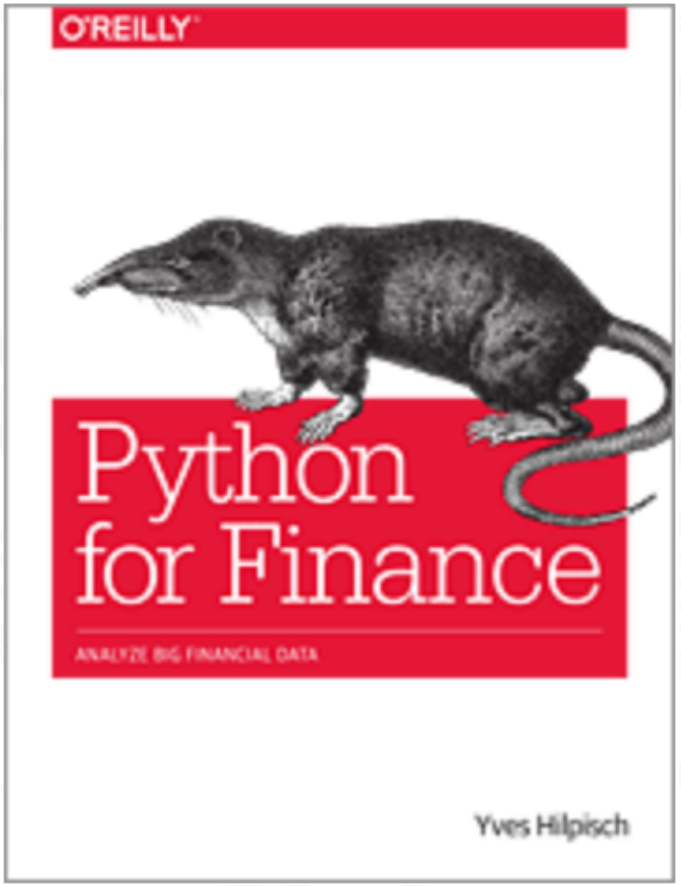
Administrivia

- Office: CST 4-230
- Office Hour: TBD
- Schedule an appointment *if you cannot get to office hours*.
- Email: mharriso@syr.edu
- Lab, Tuesday evening: optional

Administrivia Cont'd

- Text: *Python for Finance*, Yves Hilpisch
- Supporting Text: *Mastering Python for Finance*, James Ma Weiming
- additional Supporting Text: *Mastering Pandas for Finance*, Michael Heydt
- Python Reference: `https://docs.python.org/3/`
- Anaconda (our distribution of choice): `https://docs.anaconda.com/anaconda/`
- Conda (a package manager for Anaconda installations):`https://conda.io/docs/`
- Blackboard, henceforth "BB":`https://blackboard.syracuse.edu`

Main Text


O'REILLY
Python for Finance
ANALYZE BIG FINANCIAL DATA
Yves Hilpisch

More on the Course

- Introduce the most important language elements, tools & libraries that Python *as a technology platform* has to offer to finance
- Hands-on usage of Python, Numpy, Pandas, Scipy, Sympy, PyTables & others too
  - implement efficient analytics workflows

More on the Course

- Introduce the most important language elements, tools & libraries that Python *as a technology platform* has to offer to finance
- Hands-on usage of Python, Numpy, Pandas, Scipy, Sympy, PyTables & others too
  - implement efficient analytics workflows
  - perform analysis of financial time series

More on the Course

- Introduce the most important language elements, tools & libraries that Python *as a technology platform* has to offer to finance
- Hands-on usage of Python, Numpy, Pandas, Scipy, Sympy, PyTables & others too
  - implement efficient analytics workflows
  - perform analysis of financial time series
  - calculate options pricing using Black-Scholes models

More on the Course

- Introduce the most important language elements, tools & libraries that Python *as a technology platform* has to offer to finance
- Hands-on usage of Python, Numpy, Pandas, Scipy, Sympy, PyTables & others too
  - implement efficient analytics workflows
  - perform analysis of financial time series
  - calculate options pricing using Black-Scholes models
  - estimate volatility, perform portfolio and derivatives valuations

More on the Course

- Introduce the most important language elements, tools & libraries that Python *as a technology platform* has to offer to finance
- Hands-on usage of Python, Numpy, Pandas, Scipy, Sympy, PyTables & others too
  - implement efficient analytics workflows
  - perform analysis of financial time series
  - calculate options pricing using Black-Scholes models
  - estimate volatility, perform portfolio and derivatives valuations
  - calculate an option price using Monte Carlo simulations, etc.

Grading Breakdown

These are approximate and subject to change (likely to your benefit!)

- 40% Programming assignments/projects
- 30% Quizzes
- 20% Homework
- 10% Participation

- NO MIDTERM, NO FINAL

Course Content

Sensitive course materials will go on *Blackboard*. One of your responsibilities is to pay attention to what happens on Blackboard. If you have doubts about anying, then go check BB first.

Course Content

Sensitive course materials will go on *Blackboard*. One of your responsibilities is to pay attention to what happens on Blackboard. If you have doubts about anying, then go check BB first.

Everything else will go on GitHub:
`https://github.com/AmesHarrison/PythonFinance600`

Course Content

Sensitive course materials will go on *Blackboard*. One of your responsibilities is to pay attention to what happens on Blackboard. If you have doubts about anying, then go check BB first.

Everything else will go on GitHub:
`https://github.com/AmesHarrison/PythonFinance600`

Now is the time to dip your toes into *Git*.

Course Content

Sensitive course materials will go on *Blackboard*. One of your responsibilities is to pay attention to what happens on Blackboard. If you have doubts about anying, then go check BB first.

Everything else will go on GitHub:
`https://github.com/AmesHarrison/PythonFinance600`

Now is the time to dip your toes into *Git*.

No one is asking you to master it, and no one really understands it, so don't feel bad.

What is Python?

- Most popular programming language in the world (sort of - it's up there for sure)

What is Python?

- Most popular programming language in the world (sort of - it's up there for sure)
- Open source, from base distributions to packages in the broader *ecosystem*

What is Python?

- Most popular programming language in the world (sort of - it's up there for sure)
- Open source, from base distributions to packages in the broader *ecosystem*
- Used by Google, Dropbox, Netflix, YouTube, etc.

What is Python?

- Most popular programming language in the world (sort of - it's up there for sure)
- Open source, from base distributions to packages in the broader *ecosystem*
- Used by Google, Dropbox, Netflix, YouTube, etc.
- Modern, object oriented (you'll see!), high-level, multi-purpose (and multi-*paradigm* (again, you'll see!))

What is Python?

- Most popular programming language in the world (sort of - it's up there for sure)
- Open source, from base distributions to packages in the broader *ecosystem*
- Used by Google, Dropbox, Netflix, YouTube, etc.
- Modern, object oriented (you'll see!), high-level, multi-purpose (and multi-*paradigm* (again, you'll see!))
- Interpreted, eminently readable, but extensible with links to more efficient *compiled* languages such as C/C++

What is Python?

- Most popular programming language in the world (sort of - it's up there for sure)
- Open source, from base distributions to packages in the broader *ecosystem*
- Used by Google, Dropbox, Netflix, YouTube, etc.
- Modern, object oriented (you'll see!), high-level, multi-purpose (and multi-*paradigm* (again, you'll see!))
- Interpreted, eminently readable, but extensible with links to more efficient *compiled* languages such as C/C++
- At entry level, it is virtually *pseudocode*, which in turn is virtually plain language

What is Python?

- Most popular programming language in the world (sort of - it's up there for sure)
- Open source, from base distributions to packages in the broader *ecosystem*
- Used by Google, Dropbox, Netflix, YouTube, etc.
- Modern, object oriented (you'll see!), high-level, multi-purpose (and multi-*paradigm* (again, you'll see!))
- Interpreted, eminently readable, but extensible with links to more efficient *compiled* languages such as C/C++
- At entry level, it is virtually *pseudocode*, which in turn is virtually plain language
- Dynamic - no type declarations needed

What is Python?

- Most popular programming language in the world (sort of - it's up there for sure)
- Open source, from base distributions to packages in the broader *ecosystem*
- Used by Google, Dropbox, Netflix, YouTube, etc.
- Modern, object oriented (you'll see!), high-level, multi-purpose (and multi-*paradigm* (again, you'll see!))
- Interpreted, eminently readable, but extensible with links to more efficient *compiled* languages such as C/C++
- At entry level, it is virtually *pseudocode*, which in turn is virtually plain language
- Dynamic - no type declarations needed
- 'duck typing'

Python2 or Python3?

What is the difference? Look in the __future__ module.

- The **print** statement is now a *function*.

Python2 or Python3?

What is the difference? Look in the __future__ module.

- The **print** statement is now a *function*.
- Division now returns floats.

Python2 or Python3?

What is the difference? Look in the `__future__` module.

- The **print** statement is now a *function*.
- Division now returns floats.
- String literals, e.g. `"I am a string"`, are unicode by default.

Python2 or Python3?

What is the difference? Look in the __future__ module.

- The **print** statement is now a *function*.
- Division now returns floats.
- String literals, e.g. "I am a string", are unicode by default.
- A lot more *iterators* in 3, such as **range**.

Python2 or Python3?

What is the difference? Look in the `__future__` module.

- The **print** statement is now a *function*.
- Division now returns floats.
- String literals, e.g. `"I am a string"`, are unicode by default.
- A lot more *iterators* in 3, such as **range**.
- The **raw_input** is now **input**.

Python2 or Python3?

What is the difference? Look in the `__future__` module.

- The **print** statement is now a *function*.
- Division now returns floats.
- String literals, e.g. `"I am a string"`, are unicode by default.
- A lot more *iterators* in 3, such as **range**.
- The **raw_input** is now **input**.

Python2 or Python3?

What is the difference? Look in the `__future__` module.

- The **print** statement is now a *function*.
- Division now returns floats.
- String literals, e.g. `"I am a string"`, are unicode by default.
- A lot more *iterators* in 3, such as **range**.
- The **raw_input** is now **input**.

See here: `https://sebastianraschka.com/Articles/2014_python_2_3_key_diff.html`

Why Python for Finance?

Python is especially well-suited to address the analytics needs of the financial industry because of its *ecosystem* or *frameworks*. It lets non-CS types build powerful, consistent end-to-end production pipelines.

An Ambitious Syllabus

Week 1 Introduction [Ch 1 & Ch 2]

- Installing & running Python, plus various tools
- Python basics
  - Variables, names, objects & builtins
  - Types and classes, methods
  - Data Structures: lists, tuples, dictionaries, sets
  - The 'Scientific Stack', i.e. numpy,scipy, pytables , pandas, matplotlib , etc

Week 2 Python Crash Course [Ch 4]

- Functions
- Logic & control flow
- Types & classes
- Coding constructs, e.g. comprehensions, decorators, generators & more
- Numpy: array and *vectorization*

An Ambitious Syllabus

Week 3 IO Operations [Ch 7 & Ch 13]

- Modules, packages & programs, objects & classes
- Data representation - unicode, bytes, bytearrays
- Basic IO, IO with pandas and with pytables
- File IO - text and binaries
- Structured text - CSV, XML, HTML, JSON
- Structured Binary Files - spreadsheets, data as Excel files
- Relational databases

Week 4 Examples [Ch 3 & Ch 8]

- Example: **implied volatilities**
- Example: **Monte Carlo Simulation** (python, vectorization, graphics)
- Example: Technical Analysis - implement backtesting & trend-based investment strategy
- Others

An Ambitious Syllabus

Week 5 Data Viz & Web [Ch 5 & Ch 14]

- Data visualization
  - 2D graphics with matplotlib, 3D graphics & animation with pandas
  - Financial plots, graphs, visualizations
- The Web
  - Web clients & Web servers in Python
  - Web services & Automations - XML, JSON, Crawl & Scrape
  - Web plotting - static, interactive & real-time
  - Rapid web apps (traders chat room, data modelling, styling templating
  - Web services - financial model & its implementation
- Basic IO, IO with pandas and with pytables
- File IO - text and binaries
- Structured text - CSV, XML, HTML, JSON
- Structured Binary Files - spreadsheets, data as Excel files
- Relational databases

An Ambitious Syllabus

Week 6 Math Tools I [Ch 9 & Ch 13]

- Math & stats: mathematical functions, complex numbers, rationals, matrix manipulations
- numpy - working with arrays, array math, linear algebra
- scipy , sklearn , ipython , pandas
- Excel interaction - reading/writing workbooks
- Scripting Excel with Python

An Ambitious Syllabus

Week 7 Financial Time Series [Ch 6]

- pandas& DataFrame class
- Financial Data
- Regression Analysis
- High-Frequency Data

Week 8 Math Tools II [Ch 9]

- Approximation - regression, interpolation
- Convex optimization (*global*, *local* & *constrained*)
- Integration
- Symbolic computation (equations, integration, differentiation)

An Ambitious Syllabus

Week 9 Stochastic Processes [Ch 10]

- Random numbers
- Simulation - random variables, stochastic processes, variance reduction
- Valuation - European, American
- Risk measures

Week 10 Stats [Ch 11]

- Normality tests - benchmarks, real-world data
- Portfolio optimization - efficient frontier, capital market line
- Principal Components Analysis (PCA) - DAX index & its 30 stocks, PCA index
- Bayesian regression (example, real data)

An Ambitious Syllabus

Week 11 Simulation [Ch 15]

- Random number generation & generic simulation class
- Geometric Brownian motion (use case)
- Jump diffusion (use case)
- Square-root diffusion (use case)

Week 12 Derivatives Valuation [Ch 17]

- Generic Valuation Class
- European Exercise (use case)
- American Exercise (least-squares Monte Carlo, use case)

An Ambitious Syllabus

Week 13 Portfolio Valuation & Volatility Options [Ch 18 & Ch 19]

- Derivatives Portfolios (use case)
- Derivatives Positions (use case)
- The VSTOXX Data Index, Futures data, Options data
- Model Calibration Market Data, Option Modelling, Calibration
- American options on the VSTOXX - modelling option positions, options portfolio

Week 14 Examples

Today

- Getting Python
- IDEs - *Integrated Development Environments*
  - Shell
  - IDLE
  - Spyder
  - Ipython
  - Jupyter
- Running Python programs
- Interactive Python sessions
- Editing Python scripts
- Anaconda & conda
- PyPi & pip