Python for Financial Research

Dr. Vincent Grégoire
Department of Finance

March 2018

Why learn to code?

- It's fun
- Saves you time
- Produce better research
- Looks good on your CV
- Likely to help your career down the road

Why learn to code?

Automation is already here

"At its height back in 2000, the U.S. cash equities trading desk at Goldman Sachs's New York headquarters employed 600 traders, buying and selling stock on the orders of the investment bank's large clients. Today there are just two equity traders left.

Automated trading programs have taken over the rest of the work, supported by 200 computer engineers."

–MIT Technology Review, February 2017

Why Python?

- General purpose programming language
  - Websites (i.e. Youtube, Google, Instagram)
  - Games, full software
- Free, open source software
- Very large community
- Tons of online resources and textbooks
- Integrates nicely with R, C/C++ and other languages
- More general than domain-specific software:
  - Matlab
  - R
  - SAS
  - Stata
  - EViews
  - etc. . .

Why Python?

- What is it very good for?
  - Data manipulation
  - Visualization
  - Web scraping
  - Text analysis
  - Basic to somewhat advanced statistics and econometrics
  - Linear algebra
- What is it less good for
  - Very high-performance applications (additional libraries needed)
  - Advanced statistical analysis

Main scientific Python libraries

NumPy

The main package for numerical analysis in Python. Includes functions to deal with arrays of data, linear algebra, random number generation, and much more.

Main scientific Python libraries

NumPy

The main package for numerical analysis in Python. Includes functions to deal with arrays of data, linear algebra, random number generation, and much more.

pandas

Package for panel data analysis, built on top of NumPy. Great for importing/exporting, merging, cleaning and analysing data. Written by Wes McKinney while working at AQR (now at 2σ Investments), so very good for dealing with financial data.

Main scientific Python libraries

SciPy

A collection of packages that add a lot of nice features such as more advanced linear algebra (above NumPy), numerical integration and optimization.

Main scientific Python libraries

SciPy

A collection of packages that add a lot of nice features such as more advanced linear algebra (above NumPy), numerical integration and optimization.

statsmodels

Statistical analysis, including OLS regressions with support for fixed effects and clustering.

Main scientific Python libraries

SciPy

A collection of packages that add a lot of nice features such as more advanced linear algebra (above NumPy), numerical integration and optimization.

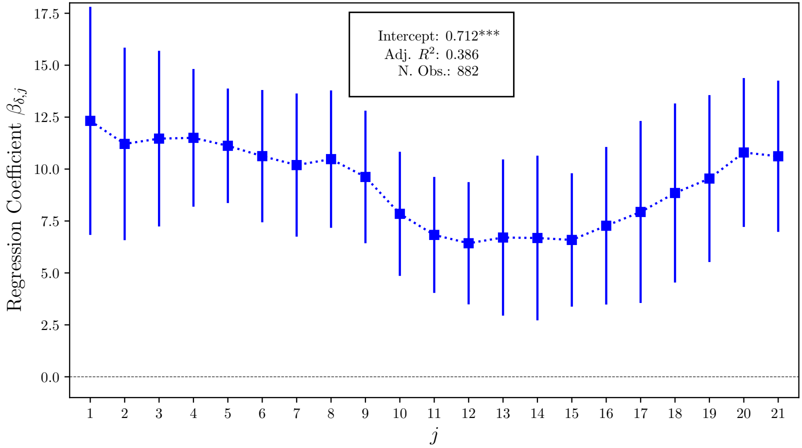
statsmodels

Statistical analysis, including OLS regressions with support for fixed effects and clustering.

matplotlib

Main package for visualization in Python. Can produce many type of graphs, highly customizable.

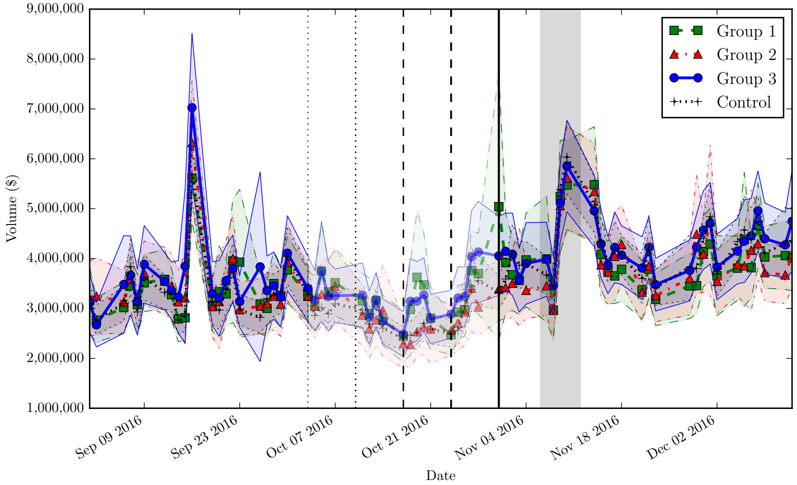
Sample Matplotlib plots



Source: Boguth, Gregoire and Martineau, *Shaping Expectations and Coordinating Attention: The Unintended Consequences of FOMC Press Conferences*, 2018.

Available at: `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2698477`

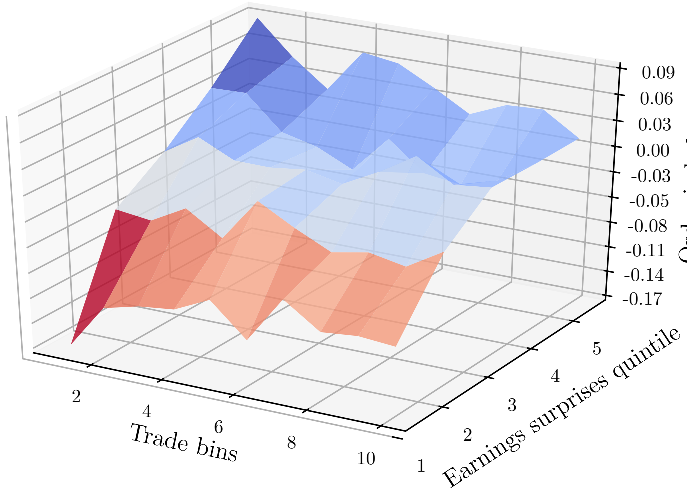
Sample Matplotlib plots



Source: Comerton-Forde, Gregoire and Zhong, *Inverted Fee Venues and Market Quality*, 2018.

Available at: `http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2939012`

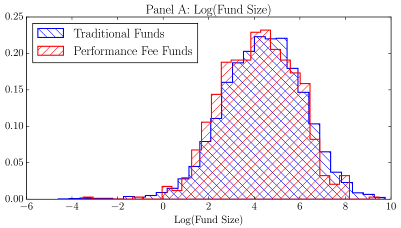
Sample Matplotlib plots

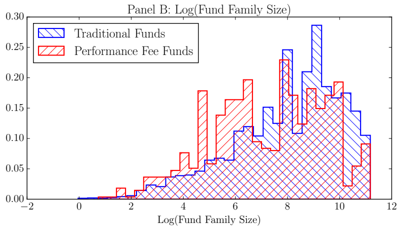


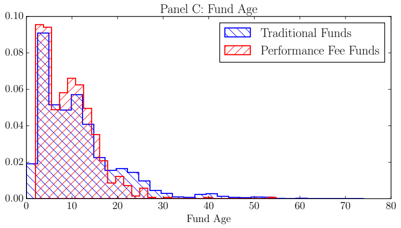
Source: Gregoire and Martineau, *How is Earnings News Transmitted to Stock Prices?*, 2018.

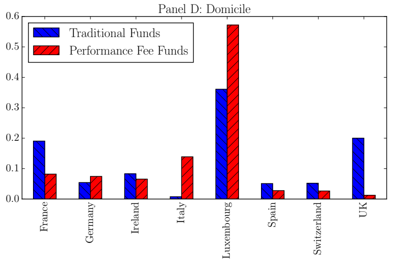
Available at: `https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3060094`

Sample Matplotlib plots









Source: Gregoire and Sotes-Paladino, *Double Bonus? Implicit incentives in Mutual Funds with Explicit Performance Fee*, 2017. (Work in progress)

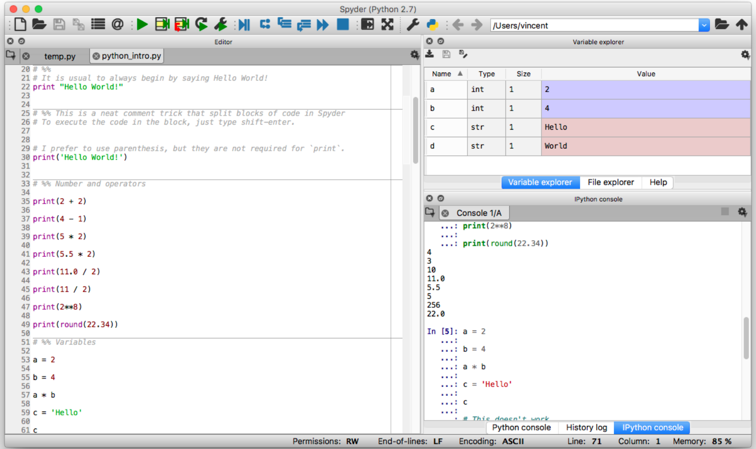
Additional useful libraries

- seaborn, plotly, Bokeh: other plotting libraries
- SymPy: symbolic algebra (think Mathematica or Maple)
- NetworkX: network analysis
- scikit-learn: machine learning
- NLTK, textblob: textual analysis
- Beautiful Soup, Scrapy, Selenium: webscraping and information extraction
- Plus many more! Just search for what you want to do, chances are there is a Python package for it.

Anaconda

- The easiest way to install Python with scientific packages is with the Anaconda distribution:
  - Mac OS, Windows, Linux
  - Python 2 or 3 (we use 3.6 in this tutorial)
  - `https://www.anaconda.com/download/`
- Includes hundreds of packages by default.
- Easily manage "environments" (sets of packages) for each project.
- Includes Spyder and Jupyter, and R as an option.

Spyder



```
# %%
# It is usual to always begin by saying Hello World!
print "Hello World!"


# %% This is a neat comment trick that split blocks of code in Spyder
# To execute the code in the block, just type shift-enter.


# I prefer to use parenthesis, but they are not required for `print`.
print('Hello World!')


# %% Number and operators

print(2 + 2)

print(4 - 1)

print(5 * 2)

print(5.5 * 2)

print(11.0 / 2)

print(11 / 2)

print(2**8)

print(round(22.34))

# %% Variables

a = 2

b = 4

a * b

c = 'Hello'

c
```

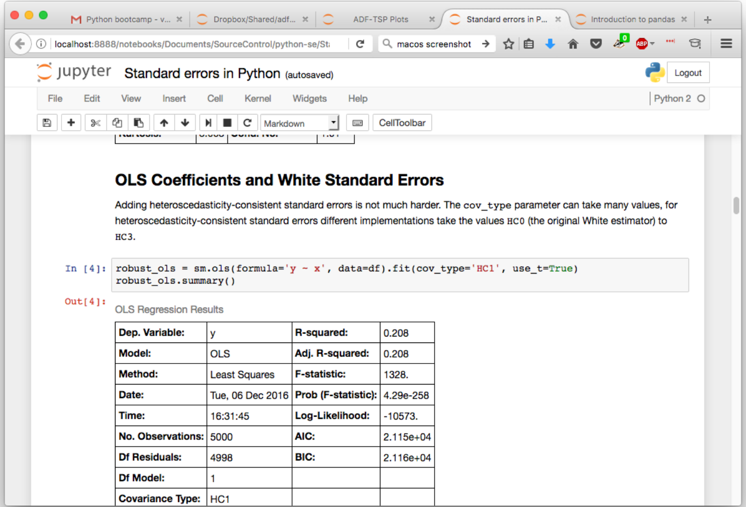
Name ▲	Type	Size	Value
a	int	1	2
b	int	1	4
c	str	1	Hello
d	str	1	World

```
   ...: print(2**8)
   ...:
   ...: print(round(22.34))
4
3
10
11.0
5.5
5
256
22.0

In [5]: a = 2
   ...:
   ...: b = 4
   ...:
   ...: a * b
   ...:
   ...: c = 'Hello'
   ...:
   ...: c
   ...:
   ...: # This doesn't work
```

Permissions: RW   End-of-lines: LF   Encoding: ASCII   Line: 71   Column: 1   Memory: 85 %

Jupyter (IPython notebook)



OLS Coefficients and White Standard Errors

Adding heteroscedasticity-consistent standard errors is not much harder. The `cov_type` parameter can take many values, for heteroscedasticity-consistent standard errors different implementations take the values `HC0` (the original White estimator) to `HC3`.

```
In [4]: robust_ols = sm.ols(formula='y ~ x', data=df).fit(cov_type='HC1', use_t=True)
        robust_ols.summary()
```

Out[4]:

OLS Regression Results

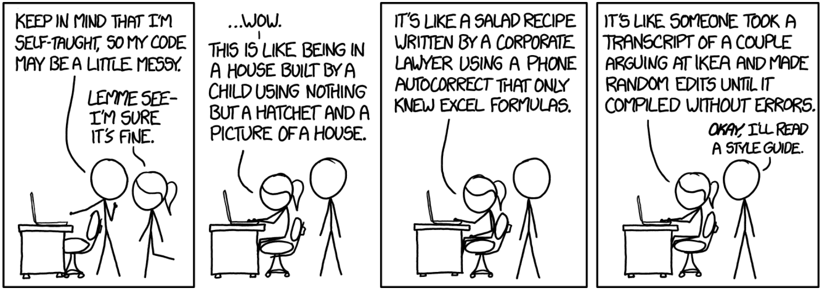
Dep. Variable:	y	**R-squared:**	0.208
Model:	OLS	**Adj. R-squared:**	0.208
Method:	Least Squares	**F-statistic:**	1328.
Date:	Tue, 06 Dec 2016	**Prob (F-statistic):**	4.29e-258
Time:	16:31:45	**Log-Likelihood:**	-10573.
No. Observations:	5000	**AIC:**	2.115e+04
Df Residuals:	4998	**BIC:**	2.116e+04
Df Model:	1		
Covariance Type:	HC1		

Keep you code clean

- Write code that is easy to read, and document it well.
- Will save you time in the long run, easier to spot errors.
- Also easier to get feedback and help!
- Python style guide is called PEP8.
  - `https://www.python.org/dev/peps/pep-0008/`

Keep you code clean

- Write code that is easy to read, and document it well.
- Will save you time in the long run, easier to spot errors.
- Also easier to get feedback and help!
- Python style guide is called PEP8.
    - `https://www.python.org/dev/peps/pep-0008/`


KEEP IN MIND THAT I'M SELF-TAUGHT, SO MY CODE MAY BE A LITTLE MESSY.
LEMME SEE– I'M SURE IT'S FINE.
...WOW.
THIS IS LIKE BEING IN A HOUSE BUILT BY A CHILD USING NOTHING BUT A HATCHET AND A PICTURE OF A HOUSE.
IT'S LIKE A SALAD RECIPE WRITTEN BY A CORPORATE LAWYER USING A PHONE AUTOCORRECT THAT ONLY KNEW EXCEL FORMULAS.
IT'S LIKE SOMEONE TOOK A TRANSCRIPT OF A COUPLE ARGUING AT IKEA AND MADE RANDOM EDITS UNTIL IT COMPILED WITHOUT ERRORS.
OKAY, I'LL READ A STYLE GUIDE.

Credit: `http://xkcd.com`

Online resources

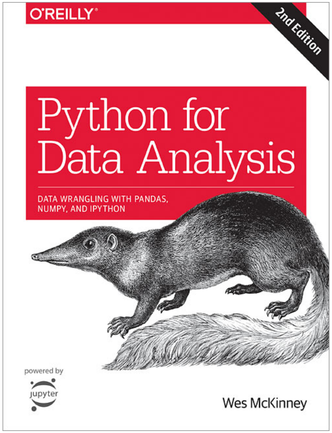
- Slides and code from this workshop `https://github.com/vgreg/python-finance-unimelb2018`
- Python documentation `https://docs.python.org/3/`
  - Tutorial: `https://docs.python.org/3/tutorial/index.html`
- LearnPython.org `http://www.learnpython.org/`
- Package documentation
  - pandas: `http://pandas.pydata.org/pandas-docs/stable/`
  - matplotlib: `http://matplotlib.org/contents.html`
  - numpy: `https://docs.scipy.org/doc/numpy/index.html`

Online resources

- Using Python with WRDS `https://wrds-www.wharton.upenn.edu/pages/support/wrds-cloud/holding/python-programming-wrds/`
- Kaggle `https://www.kaggle.com/`
- Quantopian lectures `https://www.quantopian.com/lectures`
- Stack Overflow `http://stackoverflow.com/`
- Google / DuckDuckGo / Bing (?)

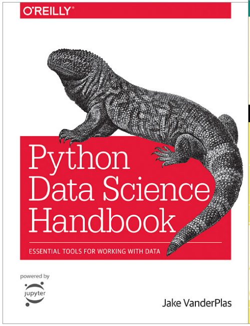
Books

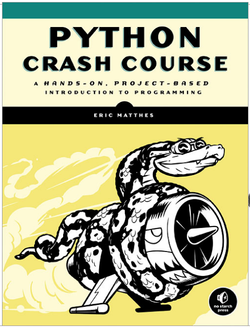
If there's one book you need to get, it's this one.

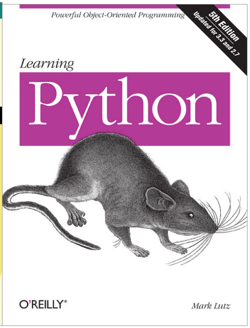


Note: Make sure to get the second edition (2017).

Books


O'REILLY
Python Data Science Handbook
ESSENTIAL TOOLS FOR WORKING WITH DATA
powered by
jupyter
Jake VanderPlas


PYTHON CRASH COURSE
A HANDS-ON, PROJECT-BASED INTRODUCTION TO PROGRAMMING
ERIC MATTHES
no starch press


Powerful Object-Oriented Programming
5th Edition
Updated for 3.3 and 2.7
Learning
Python
O'REILLY
Mark Lutz

Other (free, open source) tools for research

- Markdown: Easy markup language that can be used in Jupyter notebooks. Visual Studio Code is a great text editor for Markdown and Python (now distributed by Anaconda.)
  - `http://daringfireball.net/projects/markdown/`
  - `https://code.visualstudio.com/`

- LaTeX: Typesetting system to produce nice report papers (and presentations such as this one). You can export pandas datasets directly to LaTeX tables, and import in LaTeX the PDF figures created in Matplotlib.
  - `https://www.tug.org/begin.html`

Other (free, open source) tools for research

- R: Statistical computing, more advanced functions than currently in Python. You can install R through Anaconda, and call R code from within Python. (Microsoft's version is free as well)
  - `https://www.r-project.org/`
  - `https://mran.microsoft.com/`
- git: Version control, think "track changes" for your code.
  - `https://git-scm.com/doc`
  - `https://github.com/`

Outline

Four blocks of three hours:

1. Introduction to Python programming

We will discuss what is Python and you will learn the basic structure of the language. You will also learn our way around the programming environment, including the two main editors for scientific Python, Spyder and Jupyter.

Outline

Four blocks of three hours:

1. Introduction to Python programming

We will discuss what is Python and you will learn the basic structure of the language. You will also learn our way around the programming environment, including the two main editors for scientific Python, Spyder and Jupyter.

2. Introduction to data analysis using pandas, matplotlib

You will learn how to import, export and transform data using pandas, the panel data package for Python. You will also learn how to explore the data by generating summary statistics and plotting graphs using matplotlib.

Outline

3. More data analysis using pandas and statsmodels

You will learn more advanced features of Python and pandas, including dealing with timestamps and estimating measures from daily and intraday data. You will also learn how to estimate OLS and panel regressions using statsmodels.

Outline

3. More data analysis using pandas and statsmodels

You will learn more advanced features of Python and pandas, including dealing with timestamps and estimating measures from daily and intraday data. You will also learn how to estimate OLS and panel regressions using statsmodels.

4. Other topics

In this block, you will be introduced briefly to other Python packages that can be helpful for research. We will look at an example of web scraping with textual analysis.

Let's get started!